



## **WinSyslog 7.3**

© 2007 Adiscon GmbH

# Table of Contents

<b>Part I Introduction</b>	<b>4</b>
1 About WinSyslog .....	4
2 Features .....	4
3 Components .....	8
Core Components .....	8
Add-On Components .....	9
How these components work together .....	10
4 System Requirements .....	12
<b>Part II Getting Started</b>	<b>13</b>
1 Setup .....	13
2 Creating an Initial Configuration .....	14
3 Installing phpLogCon .....	14
4 Obtaining a Printable Manual .....	14
5 Export Settings .....	15
<b>Part III Step-by-Step Guides</b>	<b>16</b>
<b>Part IV Using Interactive Syslog Server</b>	<b>17</b>
1 Launching the Interactive Syslog Server .....	17
2 The Interactive Logging .....	18
3 Interactive Syslog Server Options .....	19
<b>Part V Configuring WinSyslog</b>	<b>22</b>
1 License Options .....	25
2 General Options .....	26
3 Services .....	32
Understanding Services .....	32
Syslog Server .....	33
SETP Server .....	35
Heartbeat .....	38
SNMP Trap Receiver Service .....	39
MonitorWare Echo Reply .....	41
4 Filter Conditions .....	41
Filter Conditions .....	41
Filter Conditions - Brushup .....	44
Global Conditions .....	45
Operators .....	46
Filters .....	46
General .....	48

Date/Time .....	50
InformationUnit Type .....	51
Syslog .....	52
SNMP Traps .....	53
Custom Property .....	55
Store Filter Results .....	56
<b>5 Actions .....</b>	<b>56</b>
Understanding Actions .....	56
Resolve Hostname Action .....	56
File Options .....	57
Database Options .....	63
OLEDB Database Action .....	67
Event Log options .....	70
Mail Options .....	72
Forward Syslog Options .....	77
Forward SETP Options .....	79
Net Send .....	81
Start Program .....	82
Play Sound .....	84
Send to Communications Port .....	85
Set Status .....	88
Set Property .....	89
Call RuleSet .....	90
Discard .....	91
Post-Process Event .....	91
<b>Part VI Getting Help .....</b>	<b>100</b>
<b>Part VII WinSyslog Concepts .....</b>	<b>103</b>
<b>Part VIII Purchasing WinSyslog .....</b>	<b>103</b>
<b>Part IX Reference .....</b>	<b>104</b>
<b>1 Comparison of properties Available in MonitorWare Agent, EventReporter and WinSyslog</b>	
<b>2 Event Properties .....</b>	<b>105</b>
<b>Accessing Properties .....</b>	<b>105</b>
Property.....	106
FromPos.....	106
ToPos .....	107
Options.....	108
Examples.....	109
<b>System Properties .....</b>	<b>110</b>
<b>Custom Properties .....</b>	<b>111</b>
<b>Event-Specific Properties .....</b>	<b>111</b>
Standard Properties.....	112
Windows Event Log Properties.....	114
Syslog Message Properties .....	114
Disk Space Monitor.....	114
File Monitor.....	114
Windows Service Monitor.....	115
Ping Probe.....	115
Port Probe.....	116

Database Monitor.....	116
Serial Monitor.....	116
MonitorWare Echo Request.....	116
FTP Probe.....	117
IMAP Probe.....	117
NNTP Probe.....	117
SMTP Probe.....	117
POP3 Probe.....	117
HTTP Probe.....	117
<b>3 Complex Filter Conditions .....</b>	<b>117</b>
<b>4 WinSyslog Shortcut Keys .....</b>	<b>121</b>
<b>5 Version Comparison .....</b>	<b>122</b>
<b>Part X Copyrights .....</b>	<b>122</b>
<b>Part XI Glossary of Terms .....</b>	<b>122</b>
1 EventReporter .....	122
2 Millisecond .....	123
3 Monitor Ware Line of Products .....	123
4 Resource ID .....	123
5 SETP .....	124
6 SMTP .....	124
7 Syslog Facility .....	125
8 TCP .....	125
9 UDP .....	125
10 Upgrade Insurance .....	125
11 UTC .....	126
<b>Index .....</b>	<b>0</b>

# 1 Introduction

## 1.1 About WinSyslog

**WinSyslog is an enhanced syslog server for Windows. It serves the same purpose as a Unix Syslog daemon. It is an integrated, modular and distributed solution for system management.**

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

[Syslog](#) is a standard protocol for centralized reporting of system events. Its roots are in the UNIX environment, but most modern devices (e. g. Cisco routers) use the Syslog protocol. They report important events, operating parameters and even debug messages via Syslog. Unfortunately Microsoft Windows does not include a Syslog server (a Syslog server is called "Syslog daemon" or - short - Syslogd under UNIX).

Adiscon's [WinSyslog](#) fills this gap. Prior to version 3.0, WinSyslog was known under the name of "NTSLog". WinSyslog is the first and original Syslog server available on the Windows platform. Its initial version was created in 1996 just to receive Cisco routers status messages. The product has been continuously developed during the past years. Version 3 represented a major stepping stone. That was the main reason we decided to rename the product.

WinSyslog can also be used in conjunction with Adiscon's [MonitorWare Agent](#), [EventReporter](#) and [ActiveLogger](#) products to build a totally centralized Windows event log monitoring tool. More information on centrally monitoring Windows NT/2000/XP/2002 can be found at [www.monitorware.com](http://www.monitorware.com)

Most customers use WinSyslog to gather events reported from Syslog enabled devices (routers, switches, firewalls and printers to name a few) and store them persistently on their Windows system. WinSyslog can display Syslog messages interactively on-screen but also store them in flat ASCII files, ODBC databases or the Windows event log. The product runs as a reliable background service and needs no operator intervention once it is configured and running. As a service, it can start up automatically during Windows boot.

The improvised services and rules introduced in version 4 allow very flexible configuration of WinSyslog. WinSyslog detects conditions like string matches in the incoming messages and can actively act on them. For example, an email message can be send if a high priority message is detected. There can also be multiple Syslog servers running at the same time, each one listening to different ports.

## 1.2 Features

### Centralized Logging

This is the key feature. WinSyslog gathers all Syslog messages send from different sources and stores them locally on the Windows system. Event source can be any Syslog enabled device. Today, virtually all devices can use Syslog. Prominent examples are Cisco routers.

### **Ease of Use**

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

### **Powerful Actions**

Each message received is processed by WinSyslog's powerful and extremely flexible rule engine. Each rule defines which actions to carry out (e. g. send an email message or store event log to a database) when the message matches the rule's filter condition. Among others, filter conditions are string matches inside the message or Syslog facility or priority. There are an unlimited number of filter conditions and actions per rule available.

### **Interactive Server**

Use the Interactive Syslog Server to interactively display messages as they arrive. Message buffer size is configurable and only limited by the amount of memory installed in the machine.

### **Send Syslog Test Message**

WinSyslog client comes with "Send Syslog Test Message" facility. It can be accessed via the "Tools" menu. This option enables to check if syslog messages being sent properly to the destination or not. Please note that the "Send Syslog Test Message" sends UDP syslog, only! It does not at all send RFC 3195, nor syslog/tcp!

### **Freeware Mode**

We care for the home user! WinSyslog can operate as freeware in so-called "freeware mode" without a valid license. It supports a scrolling interactive display of the 60 most current messages for an unlimited time. This feature is most commonly requested for home environments. And: even our free copies come with Adiscon's great support!

### **Standards Compatible**

WinSyslog is compatible with the Syslog [RFC 3164](#). It operates as an original sender (device), server and relay. All specified operation modes are supported. Non-RFC compliance can be configured by the administrator to fine-tune WinSyslog to the local environment (e.g. timestamps can be taken from the local system instead of the reporting device in case the device clocks are unreliable).

### **WinSyslog Web Access**

---

Never need to look at plain text files! WinSyslog comes with a fully functional ASP application that will display the contents of WinSyslog generated database entries. The ASP pages are in full source code and can easily be customized.

### **Syslog Hierarchy**

WinSyslog supports cascaded configurations most commonly found in larger organizations. In a cascaded configuration, there are local WinSyslog instances running at department or site level which report important events to a central WinSyslog in the headquarter. There is no limit on the number of levels in a cascaded system.

### **Email Notifications**

WinSyslog emails receive events based on the user defined rule set. Email notifications can be sent to any standard Internet email address, which allows forwarding not only to typical email clients but also pager and cellular phones. The email subject line is fully customizable and can be set to include the original message. That way, pagers can receive full event information.

### **Store Messages Persistently**

The WinSyslog server process stores all messages persistently. It helps to audit and review important system events later on without any hard effort. Messages can be written to flat ASCII files, ODBC data sources and the Windows event log.

### **Multiple Instances**

WinSyslog supports running multiple Syslog servers on the same machine. Each instance can listen to a different Syslog port, either via TCP or UDP and can be bound to a different rule set for execution.

### **Full Logging**

WinSyslog logs the received Syslog message together with it's priority and facility code as well as the sender's system IP address and date. It is also able to log abnormally formatted packages (without or with invalid priority / facility), so no message is lost.

### **Robustness**

WinSyslog is written to perform robust even under unusual circumstances. Its reliability has been proven at customers sites since 1996.

### **Minimal Resource Usage**

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

### **Firewall Support**

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

### **NT Service**

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

### **Full Windows 2000, 2003 and XP Support**

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

### **Multi-Language Client**

The WinSyslog 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 XML based localization technology. We ask customers interested in an additional language for a little help with the translation work (roughly 1 hour of work). Adiscon will than happily create a new version. This service is free!

### **Friendly and Customizable User Interface**

New Skinning feature has been added to the WinSyslog Client. By default 5 new fresh skins are installed and can be selected. These skins can be colorized with Hue, Saturation and RGB colors. [Click to see.](#)

New Cloning feature added to the WinSyslog 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 WinSyslog Client.

The WinSyslog 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.

## 1.3 Components

### 1.3.1 Core Components

#### WinSyslog Configuration Client

The WinSyslog Configuration Client - called "the Client" - is used to configure all components and features of the WinSyslog Service. 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.

#### WinSyslog Service

The WinSyslog Service - called "[the service](#)" - runs as a Windows service and carries out the actual work.

The service is the only component that needs to be installed on a monitored system. The WinSyslog service is called the product "engine". As such, we call systems with only the service installed "[engine-only](#)" installations.

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

#### x64 Build

This is the first Version which introduces WinSyslog on the x64 platform. 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 WinSyslog!**

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 an 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 WinSyslog.
3. Install WinSyslog by using the x64 Edition of the setup.
4. Import your old settings from the registry or xml file.

### 1.3.2 Add-On Components

#### **Interactive Syslog Server**

The Interactive Syslog Server is a Windows GUI application receiving and displaying Syslog events. It is a Syslog server in it's own right. Typically, it is used in conjunction with the WinSyslog service, but it can also be used as a stand-alone Syslog server.

The Interactive Syslog Server replaces the Interactive display from the pre 4.0 release WinSyslog Client. It was brought into a separate program because there was some confusion about the interactive display in the past.

#### **PHPLogCon**

PHPLogCon is a convenient facility to access MonitorWare gathered events over the web. All major browsers are supported.

PHPLogCon is an easy to use solution for browsing Syslog messages, Windows event log data and other network events over the web. PHPLogCon enables the system administrator to quickly and easily review his central log repository. It provides views typically used on log data. It integrates with web resources for easy analysis of data found in the logs.

Mainly it helps to have quick overview over current system activity and accessing the log data while not being able to access the administrator workstation (e.g. being on the road or roaming through the enterprise). While originally initiated to work in conjunction with Adiscon's MonitorWare product line, it can easily be modified to work with other solutions as well.

PHPLogCon is included in the MonitorWare Agent install set. It gets copied onto machine but not installed. For installation of PHPLogCon, refer to the installation instructions in the doc folder of PHPLogCon or see the online manual at [http://www.phplogcon.com/module-Static\\_Docs-view-f-index.htm.phtml](http://www.phplogcon.com/module-Static_Docs-view-f-index.htm.phtml)

#### **MonitorWare Console**

MonitorWare Console facilitates the Network Administrators to gather valuable information about their networks and offers them strong analytical abilities with which they can examine their network proficiently against countless problems including security breaches. Using the Views and Reporting Modules of MonitorWare Console, you can find the problematic areas in your network very efficiently and promptly. As a network administrator, you would not only like to find the problems but also their solutions. MonitorWare Console's Knowledge Base Module is exactly meant for this

purpose. In short, MonitorWare Console is a very powerful tool that facilitates the Network Administrators to scrutinize their networks from tip to toe and give an in-depth perspective about what's going on in their system.

For further details please visit the MonitorWare Console website at [www.mwconsole.com](http://www.mwconsole.com)

### 1.3.3 How these components work together

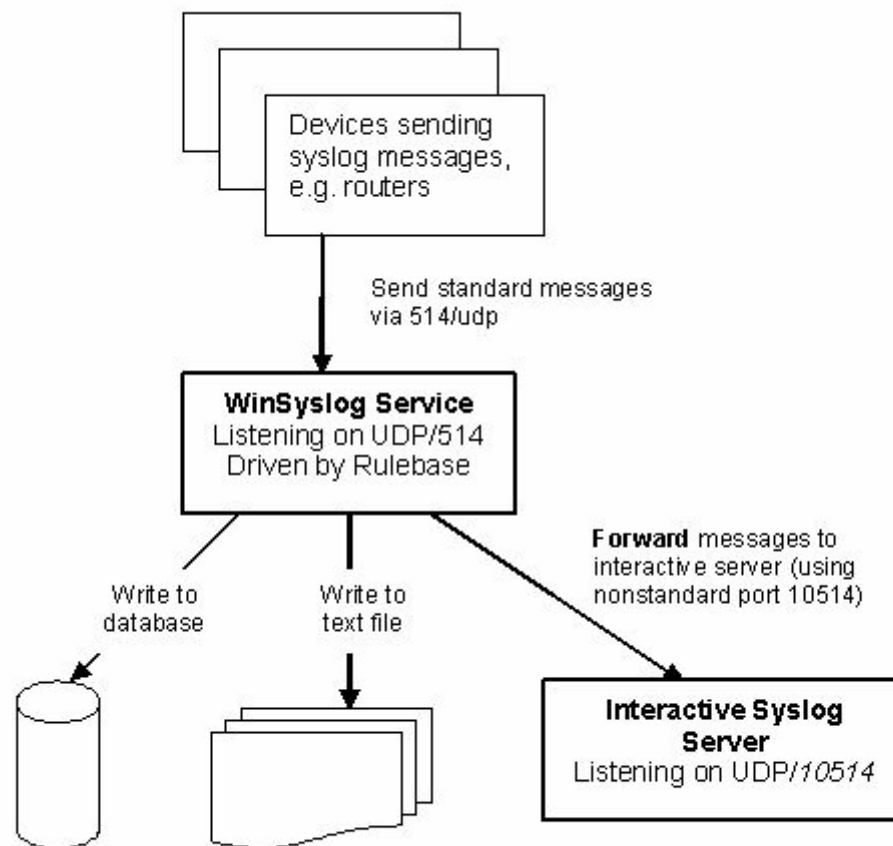
#### How these components work together?

All four components work closely together. The core component is the WinSyslog Service, continuously running in the background. WinSyslog Configuration Client creates the service configuration. This is the only task performed with the Configuration Client. Consequently, the Configuration Client does not need to be run continuously.

Once the service is configured, it operates in the background and performs the configured duties. Most importantly, this includes receiving Syslog messages, processing them via the rule base and storing them e.g. to a database, text file or creating alerts.

The WinSyslog service itself does not have any interactive component. If Syslog messages should be displayed with a Windows GUI, the Interactive Syslog Server is needed. That server is implemented as a lightweight Syslog server. So itself is a full Syslog server with limited capabilities but interactive message display. It performs its work only while it is running. To view Syslog messages interactively, the WinSyslog service forwards them to the Interactive server. By default, this is done via the non-standard port 10514 over UDP. As such, both Syslog servers (the service as well as the interactive one) can run on a single machine without conflicts.

The message flow can be seen in this diagram:



In a typical configuration, the Syslog devices (for example routers or switches) send standard Syslog messages via port 514 to the WinSyslog service. The service receives these messages and processes them as configured in the rule base. In our example, there are three actions configured for all incoming messages: writing them to a database, to a text file as well as forwarding them to the Interactive Syslog Server.

By default, messages are forwarded to the local (127.0.0.1) Interactive Server via port 10514. The Interactive Server in turn listens to that port and receives the forwarded Syslog messages from the server.

In UNIX-speak, the WinSyslog Service acts as a receiver as well as a Syslog relay. The Interactive Syslog Server is just a receiver (and can never relay).

In fact, we have a cascaded Syslog server configuration here. Please note that the Interactive Server is able to display the original message origin's address as the message source because it honors a custom extension to the Syslog protocol that enables this functionality.

The Configuration Client is only needed to create the service configuration. Once this is done, it need not to be used and as such is not part of the message flow.

PHPLogCon is only needed if accessing Syslog messages over the web is desired. It is a convenient facility to access MonitorWare gathered events over the web. All major browsers are supported. PHPLogCon is included in the WinSyslog install set. It gets copied onto machine but not installed. For installation of PHPLogCon, refer to the installation instructions in the doc folder of PHPLogCon or see the online manual at <http://www.phplogcon.com/en/Manual/current/index.htm>. Please email [support@adiscon.com](mailto:support@adiscon.com), if you want some more help in this regard.

Please keep in mind that the above example is just an example - there are numerous ways to configure WinSyslog and its components to suit every specific need. But we hope this sample clarifies how the WinSyslog components work together.

## 1.4 System Requirements

The WinSyslog Service has minimal system requirements. The actual minimum requirements depend on the type of installation. If the Client is installed, they are higher. The service has very minimal requirements, enabling it to run on a large variety of machines - even highly utilized ones.

### Client

- The **Client and Interactive Syslog Server** can be installed on Windows NT 4.0 SP6 and above. This includes Windows 2000 (SP3 minimum due the requirements of Windows Installer 3.0), Windows XP and the Windows 2003 servers. The operating system variant (Workstation, Server ...) is irrelevant.
- The Client uses XML technology. Unfortunately, operating system XML support is only available if at least Internet Explorer 4.01 SP1 is installed.
- The Client requires roughly 6 MB RAM in addition to the operating system minimum requirements. It also needs around 10 MB of disk space.
- The Client is available for Intel based systems, only.

### Service

- The **service** has fewer requirements. Most importantly, it does not need Internet Explorer to be installed on the system.
- It works under the same operating system versions.
- Additionally, it should perform well under NT 3.51, but as we have not yet received any request for supporting this operating system version, no tests have been conducted yet. This will be done upon request.
- The service also by design supports the Compaq/Digital APHA processor, but again has not been ported yet due to missing demand. If you are in need of such a version, please contact Adiscon at [support@adiscon.com](mailto:support@adiscon.com).
- At runtime, the base service requires 4 MB of main memory and less than 1 MB of disk space. However, the actual resources used by the service largely depend on the services configured.
- If the service shall just receive a few syslog messages per second, a performance impact is barely noticeable, if at all visible.
- If the WinSyslog service is receiving hundreds of messages per second, it will need

much more resources. Even then, the actual load is depending on the actions carried out. Storing the messages to text files is much less performance intense than writing them to a database table - especially if the database engine is located on the same machine. As such, there is no single guideline for hardware sizing. It needs to be adapted to the expected workload. We have created an article on [performance optimization for syslog server operations](#), which you may want to read.

- Please note, however, that the service is specifically optimized to handle high throughput including message bursts (for example received via syslog).
- If you expect high volume burst and carry out time consuming actions (for example database writes), we highly recommend adding additional memory to the machine. Even 64 MB additional memory will do nicely. A typical Syslog message (including overhead) will take roughly 1.5 KB. With 64 MB, you can buffer up to 50,000 messages in 64 MB.
- WinSyslog is capable of storing such bursts temporarily in memory even if the machine would otherwise be too slow to process the messages.

### PHPLogCon

- PHPLogCon requires Microsoft Internet Information Server (IIS) version 4 or higher to be present on the machine where PHPLogCon is to be installed but it is not mandatory, we recommend to go on with Apache in conjunction with PHP.

## 2 Getting Started

**WinSyslog** can be used for simple as well as complex scenarios.

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

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

### 2.1 Setup

***Setup is quick and easy. The WinSyslog Service uses a standard setup wizard.***

We highly recommend visiting our [Online Seminars](#) - to access the online seminars on WinSyslog as well as other members of this product family.

**Please note that these are not marketing videos but actually technically-packed presentations that will help you getting started quickly and efficiently.**

[Installing WinSyslog](#) is simple and easy. A standard setup program installs the application. WinSyslog is part of Adiscon's MonitorWare line of products.

A number of different [Download Versions](#) of the product is available. The install set (the ZIP file you downloaded) contains a standard setup program and its necessary

helper files. Please unzip the archive to any directory you like. This can be a local drive, a removable one or a remote share on a file server. A Win32 Unzip program can be found at [www.winzip.com](http://www.winzip.com).

After unzipping, simply double-click "setup.exe" (this is the setup program) and follow the onscreen instructions.

**Please note that you might have downloaded the setup.exe file directly. This is depending from where you download the install set. In this case simply run it to setup the product.**

## 2.2 Creating an Inital Configuration

Once WinSyslog is installed, a working configuration needs to be created. The reason is that WinSyslog does not perform any work without being instructed to do so. To create some basic work, the following needs to be done:

- **Create a simple rule set** - The most basic rule set includes no criteria, which means all incoming messages will match. To get started, we recommend using just a single "[Write to File](#)" action which will write the incoming messages to the local disk.
- **Create at least one syslog listener** - Be sure to associate the created rule set with the "[Syslog Listener](#)".
- **Start the WinSyslog service**

Your system is now ready to acced and store incoming messages.

## 2.3 Installing phpLogCon

[phpLogCon](#) is a convenient facility to access MonitorWare gathered events over the web. All major browsers are supported. phpLogCon is included in the MonitorWare Agent install set. It gets copied onto machine but not installed.

For installation of phpLogCon, refer to the installation instructions in the doc folder of phpLogCon or see the online manual at [http://www.phplogcon.com/module-Static\\_Docs-view-f-index.htm.phtml](http://www.phplogcon.com/module-Static_Docs-view-f-index.htm.phtml). Please email [support@adiscon.com](mailto:support@adiscon.com), if you want some more help in this regard.

## 2.4 Obtaining a Printable Manual

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

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

The version on the web might also include some new additions, as we post manual

changes frequently – including new samples and as soon as they become available. Past manual versions are also available for those customers in need of it.

## 2.5 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. This is done via standard Windows registry files. 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.

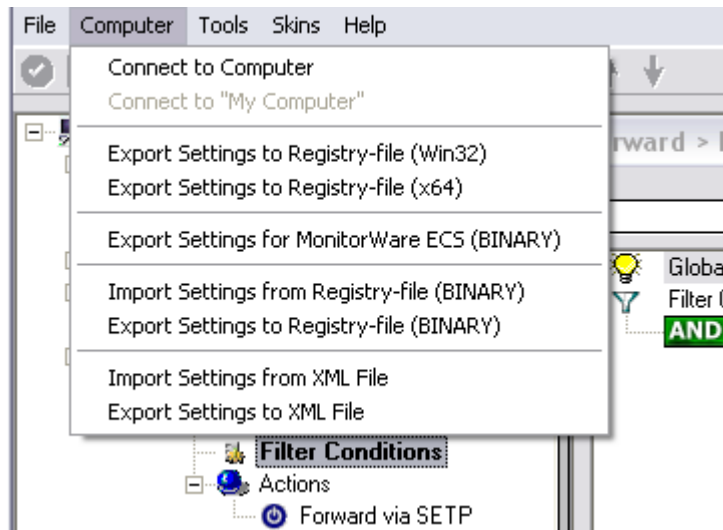


Figure1: Export Settings to a file

To use it, please do the following:

1. Go to "Computer Menu"
2. Choose "Export Settings to Registry-File" (be sure **NOT** to select a binary format - they are only for special purposes. You can also **NOT** review binary files for security-relevant data.) Please also note that you can export in Win32 or x64 format so please choose the right one for your system.
3. Save this registry 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 1 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.

### Fully XML Import & Export of Settings

It is now possible to save the whole configuration as XML. You can edit this XML, duplicate Services, Rules or Actions and reimport the Settings. This is very useful to sort and order large configurations.

### 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**

- [Database Logging with MSSQL](#)
- [How do I apply filters in MonitorWare Agent, WinSyslog and EventReporter?](#)
- [How To Setup MonitorWare Agent/ WinSyslog/ EventReporter](#)
- [How To setup php-syslog-ng with MonitorWare Products?](#)

#### **Services**

- [How To create a simple Syslog Server](#)
- [How To setup SETP Server Service](#)
- [Forwarding NT event logs to a Syslog server](#)
- [Forwarding NT event logs to an SETP server](#)

#### **Actions**

- [How To setup an SETP Action](#)

- [Creating a rule set for database logging](#)

### Centralized Monitoring

- [How To setup Windows centralized Monitoring \(EventReporter 7.x, WinSyslog 6.x and Monilog 2.x\)](#)
- [How To setup PIX centralized Monitoring \(WinSyslog 6.x, MonitorWare Agent 3.x & MonitorWare Console 2.x\)](#)
- [How To setup Windows centralized Monitoring \(EventReporter 7.x & WinSyslog 6.x\)](#)

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 Using Interactive Syslog Server

***With interactive Syslog Server is easy to immediately display Syslog messages.***

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

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

The Interactive Syslog Server replaces the Realtime Display from older WinSyslog Client version. It is a very helpful application to verify that the WinSyslog Service is running and working correctly. You can configure a Syslog Server. WinSyslog default with one Forward Syslog Action that forwards Syslog messages to the local machine on port 10514. The Interactive Server is configured to run on port 10514 by default. That means that after installing WinSyslog, you can directly use the Interactive Syslog Server to display Syslog messages.

### 4.1 Launching the Interactive Syslog Server

To run the Interactive Syslog Server, click the "Interactive Syslog Server" icon present in the Programs Folder -> WinSyslog located in the Start menu.

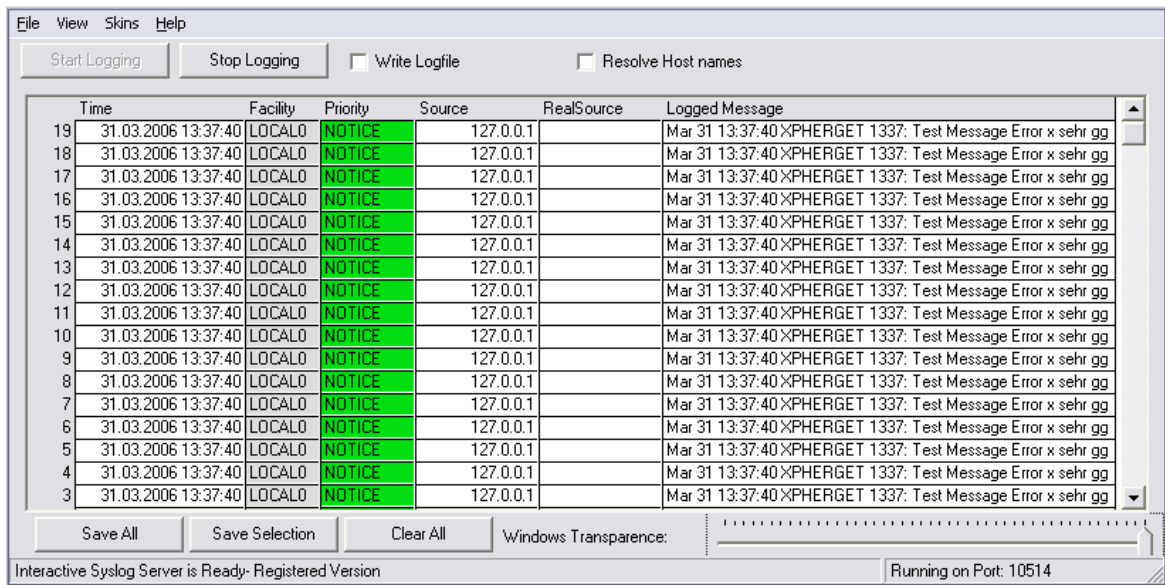
It can also be launched from the command prompt:

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

## 4.2 The Interactive Logging

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

Interactive Syslog Server is also supported under Windows 9x and Windows Me systems. The service does work on these platforms.



*Interactive Syslog Server - Interactive Logging*

### Start / Stop Logging Buttons

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

### Write Logfile

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

### Resolve Host Names

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

### **Save All**

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

### **Save Selection**

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

### **Clear All**

It erases all the messages from real-time display.

### **Windows Transparency**

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

### **Detail View**

If you double click on any message then it's detail view is opened. You can now go up and down through the received syslog messages. You can also copy a whole message into the clipboard like in the Event Viewer!

## **4.3 Interactive Syslog Server Options**

There are three options in the main bar of the Interactive Syslog Server. These are:

- 1.** File
- 2.** View
- 3.** Skin
- 4.** Help

### **File Menu**

You can access the File Menu either by clicking on the "**File**" in the main menu bar or by pressing "**Alt + F**". Once you access the File Menu you would see the following Options:

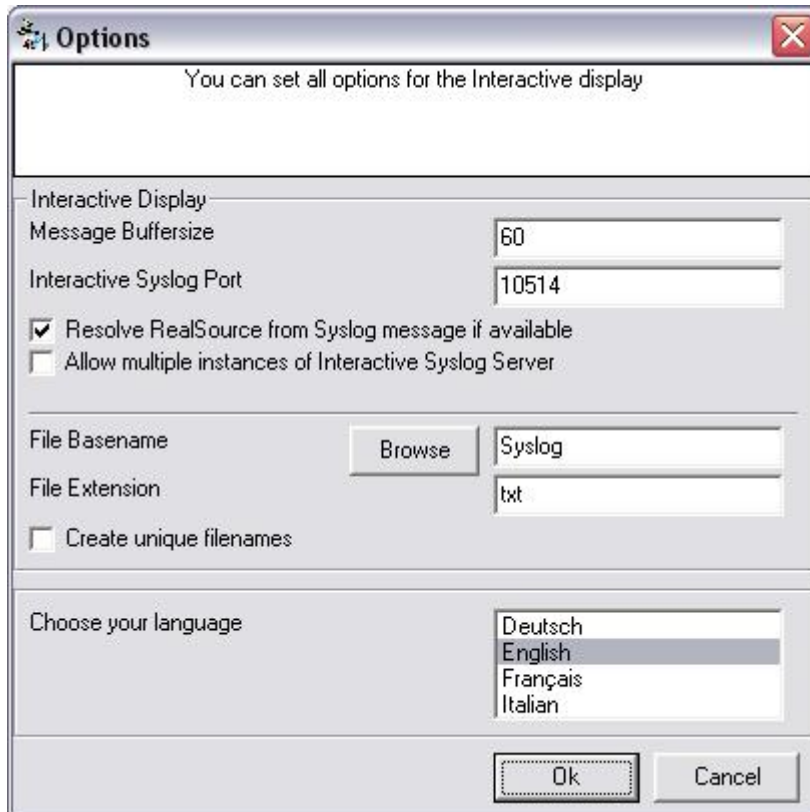
- 1.** Options

2. Run Configuration Client
3. Quit

Each of these options are described below.

## Options

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



*Interactive Syslog Server options*

### Message Buffersize

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

### Interactive Syslog Port

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

### Resolve RealSource

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

### **Allow multiple instances**

If enabled, it allows you to run the InteractiveSyslogServer in more than one instance.

### **File Basename**

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

### **File Extension**

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

### **Create unique filenames**

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

### **Language**

The Interactive Syslog Server is multilingual by design. Select the user interface language here. Languages are set on a per user basis. They can be switched instantly without the need to restart!

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

### **Run Configuration Client**

When this option is checked it starts the MonitorWare Agent Client.

### **Quit**

Closes the Interactive Syslog Server.

### **Skin Menu**

You can access the View Menu either by clicking on the "**Skin**" in the main menu bar or by pressing "**Alt + S**". This option allows you to choose the skin of your Interactive Syslog Server. By default 5 new fresh skins are installed and can be selected. These skins can be colored with Hue, Saturation and RGB colors.

### View Menu

You can access the View Menu either by clicking on the "**View**" in the main menu bar or by pressing "**Alt + V**". Currently only one option is available i.e. "**Always on Top**". If you check this option then the Interactive Syslog Server application always remains on the top of all other applications.

### Help Menu

You can access the Help Menu either by clicking on the "**Help**" in the main menu bar or by pressing "**Alt + H**". The Help menu provides a single option i.e. "**About**". If you check this option it displays information about Interactive Syslog Server client version.

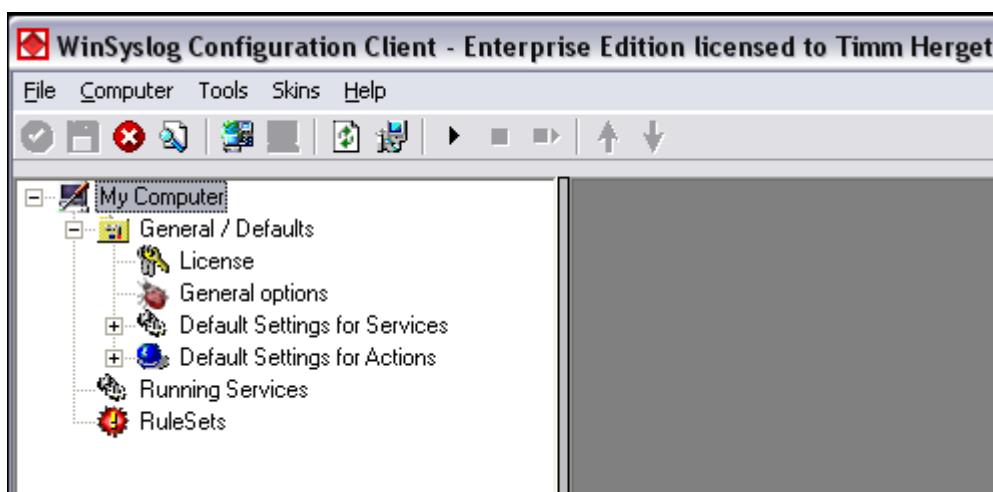
## 5 Configuring WinSyslog

*WinSyslog is easy to use and powerful.*

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

The most important part of WinSyslog - the 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 WinSyslog configuration Client application. It is used to configure the service settings.

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



### *WinSyslog 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 WinSyslog 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.

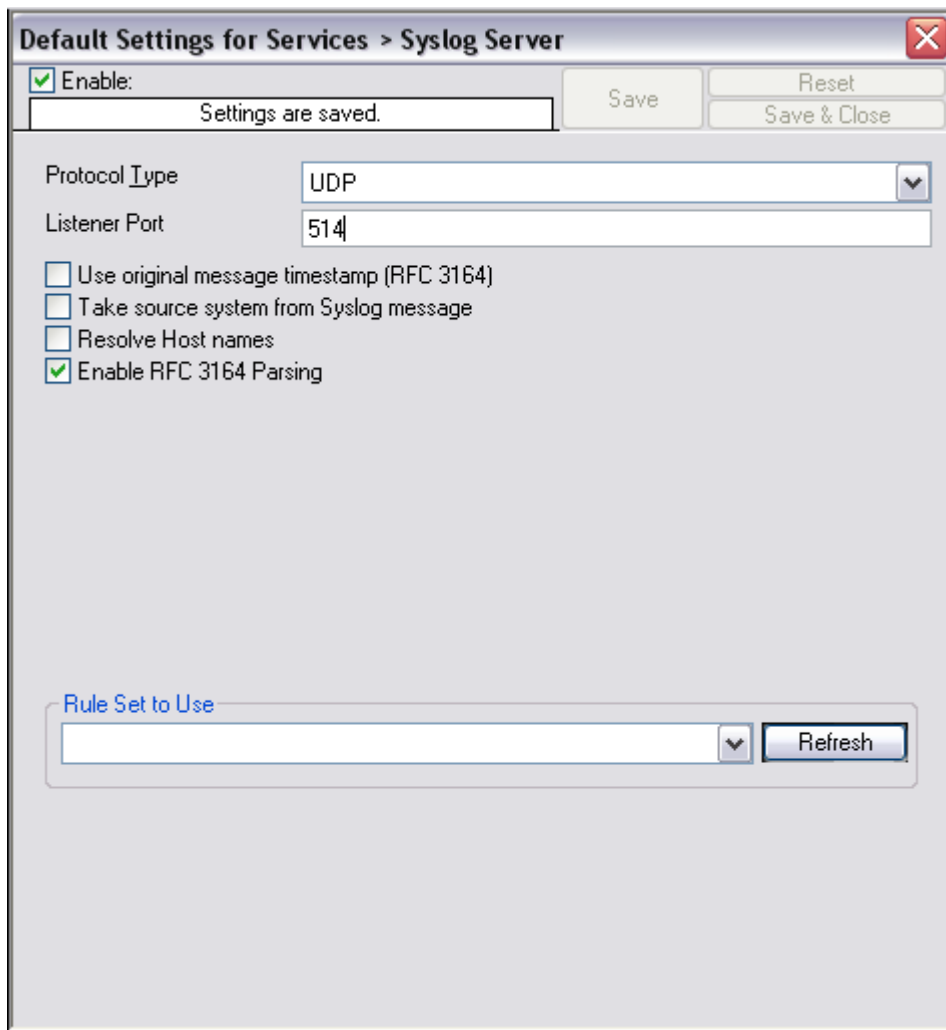
The tree view has three top-level elements: **General / Defaults**, **Running Services** and **RuleSets**.

Under **General / Defaults**, 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, which reduces 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.** Typically, there can be multiple instances of the same service running, as long as their configuration parameters do not conflict. For example the syslog service: there can be multiple syslog servers on a given system as long as they listen to different ports. Consequently, there can be multiple instances of the syslog service be created. For example, there could be three of them: two listen to the default port of 514, but one with TCP and one with UDP and a third one listens to UDP, port 10514. All three coexist and run at the same time. If these three services are listening to the same port then an error message is logged into Windows Event log that more than one instance of Syslog Server is running. After which WinSyslog wouldn't be able to perform the desired action.

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

The service definition looks like this:



*WinSyslog Configuration Client - Service Definition View*

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

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

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

The tree view's last main element is **RuleSets**. 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.

## 5.1 License Options

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

License

Settings are saved. Save Reset Save & Close ?

License

Registration Name

Registration Number

Adiscon IT-Solutions GmbH

More Information at [www.winsyslog.com](http://www.winsyslog.com)

[Click here to join the WinSyslog MailingList](#)

*License Option Parameters*

### Registration Name

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

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

## Registration Number

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

## 5.2 General Options

### General Options Tab

The General Options available on this form are explained below:

General options

Settings are saved. Save Reset Save & Close ?

General Options | Engine specific | Debug Options

ProcessPriority: Normal QueueLimit: 200000

CustomerID: 0 SystemID: 0

Location of your MIBS: D:\Program Files\MonitorWare\Agent\mibs Browse

Default Timevalues are based on:  Localtime  Universal Coordinated Time (UTC/GMT)

Protect Service against Shutdown

Log Warnings into the Windows Application Eventlog

Special Unicode Conversion for Japanese Systems

[Click here to join the MonitorWare Agent MailingList](#)

Figure1: General Options

### Process Priority

Configurable Process Priority to fine-tune application behavior.

### Queue Limit

The application 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.

### **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 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.

### **SystemID**

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

### **Location of your MIBS**

Click the Browse button to search for your MIBS location or enter the path manually.

### **Default TimeValues based on UTC or Local Time**

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

### **Protect Service against Shutdown**

The service keeps an in-memory queue of yet unprocessed events. When the service is stopped, this in-memory queue is drained. Not yet processed events are lost in such a case. If you click "Protect Service against Shutdown", it ensures that all events are processed before the service is stopped. Please note, however, that this may cause the service to look hanging. This is especially the case if there is a large in-memory queue.

### **Log Warnings into the Windows Application Eventlog**

If this option is enabled it will, as the name already says, log Program Warnings into

the Windows EventLog.

## Special Unicode Conversion for Japanese Systems

On Japanese Systems, the character handling is different and if you experience problems with the encoding of received messages, kindly turn on this new Option.

## Engine specific Options Tab

The Engine specific Options are explained below:

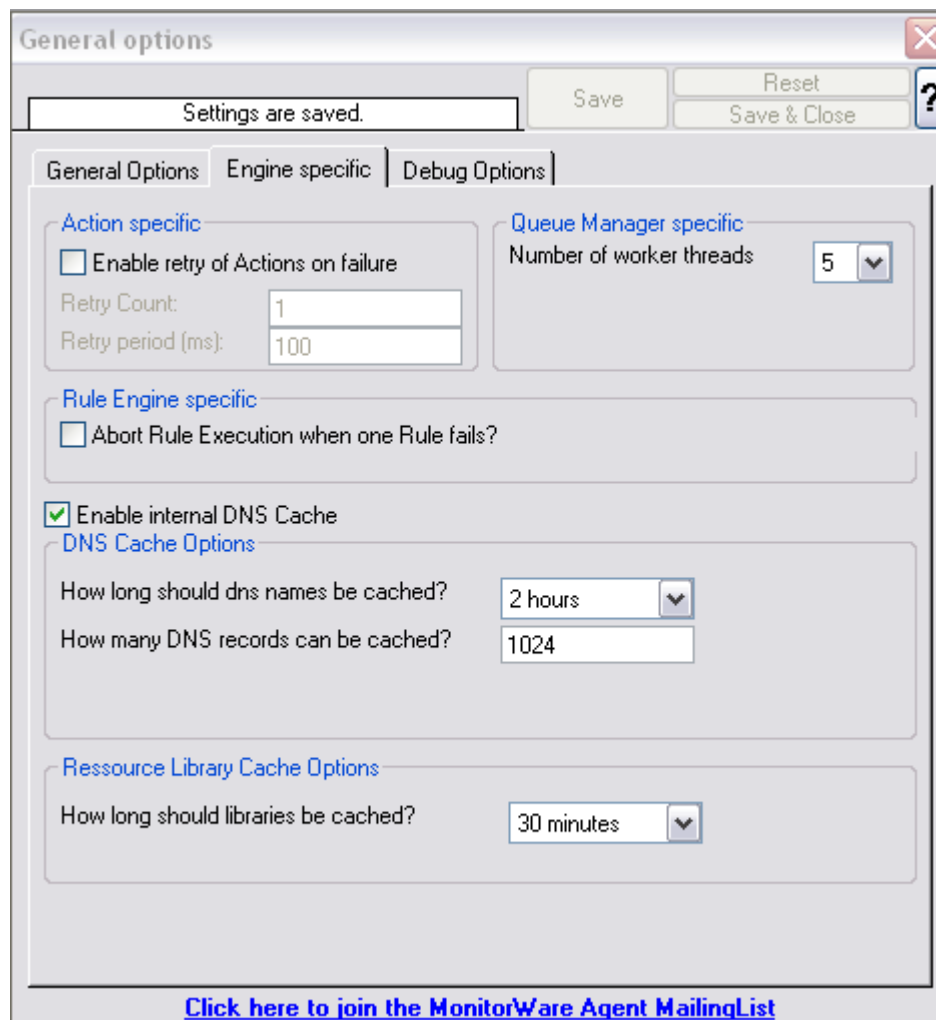


Figure2: Engine specific Options

## Action specific

### **Enable retry of Actions on failure**

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.

### **Queue Manager specific**

#### **Number of worker threads**

Defines the number of worker background threads that MWAgent uses to process it's queue.

### **Rule Engine specific**

#### **Abort Rule Execution when one Rule fails?**

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**

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 reverse 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-existing 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 overwritten with the newly requested one.

#### **How long should DNS names be cached?**

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?**

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.

## **Ressource Library Cache Options**

### **How long should libraries be cached?**

This feature will be mainly useful for EventLog Monitor. For events with the same reoccurring 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.

## **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.**

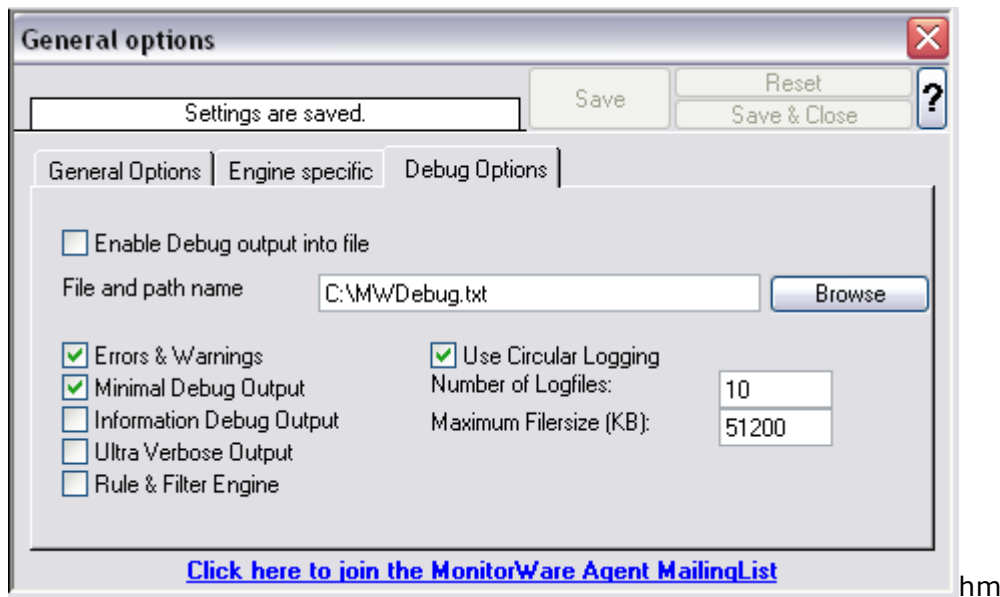


Figure3: Debug Options

### Enable Debug output into file

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

### File and path name

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

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

**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

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

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 or disable this feature.

---

### **Join the MonitorWare Agent MailingList**

Adiscon offers a free notification mailing list. The list provides information on important product enhancements and discovered bugs. Click the text to subscribe to this list. Please note that you can unsubscribe at any time and the list is not used for marketing mails.

---

## **5.3 Services**

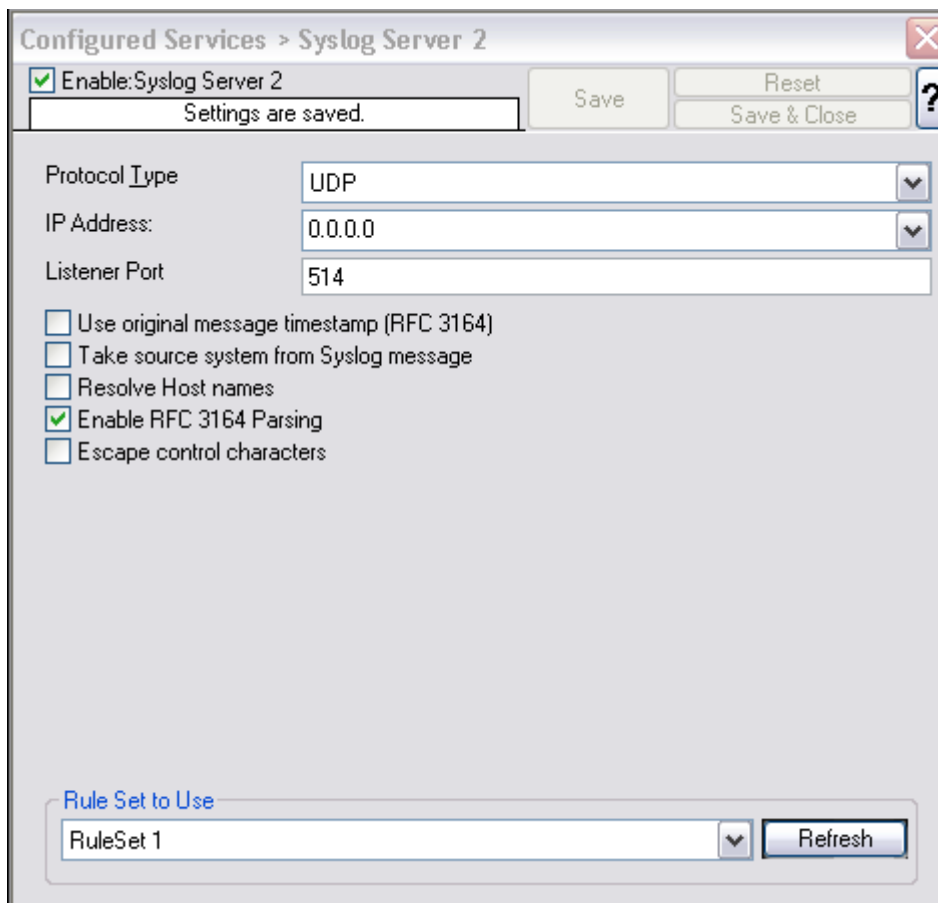
### **5.3.1 Understanding Services**

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.

### 5.3.2 Syslog Server

Configures a Syslog Server service. It can be set to listen to any valid port. UDP and TCP communication is supported.



The screenshot shows a configuration window titled "Configured Services > Syslog Server 2". At the top, there is a checkbox labeled "Enable: Syslog Server 2" which is checked. To the right of this checkbox are buttons for "Save", "Reset", and "Save & Close". Below the checkbox is a status bar that says "Settings are saved." and a help icon. The main configuration area contains the following fields and options:

- Protocol Type:** A dropdown menu set to "UDP".
- IP Address:** A dropdown menu set to "0.0.0.0".
- Listener Port:** A text input field containing "514".
- Options:** A list of checkboxes:
  - Use original message timestamp (RFC 3164)
  - Take source system from Syslog message
  - Resolve Host names
  - Enable RFC 3164 Parsing
  - Escape control characters
- Rule Set to Use:** A dropdown menu set to "RuleSet 1" with a "Refresh" button next to it.

*Syslog Server Properties*

#### Protocol Type

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

#### IP Address

The Syslog Server can now be bound to a specific IP 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.

#### Listener Port

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

### **Use Original Message Timestamp**

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

### **Take source system from Syslog message**

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

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

### **Resolve Hostnames**

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

**Please note that this setting does have any effect if the "Take source system from Syslog message" setting is checked. In this case, the message is always taken from the Syslog message itself.**

### **Enable RFC 3164 Parsing**

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

### **Escape Control Characters**

Control characters are special characters. They are used e.g. for tabulation, generating beeps and other non-printable uses. Typically, syslog messages should not contain control characters. If they do, control characters could eventually affect your logging. However, it might also be that control characters are needed.

With this setting, you can specify how control characters received should be handled.

When checked, control characters are replaced by a 5-byte sequence with the ASCII character ID. For example, a beep is the ASCII BEL character. BEL is assigned the numerical code 7. So if a BEL is received, it would be converted to "<007>" inside your syslog message. When the box is left unchecked, no conversion takes place.

In any case, ASCII NULs are converted to "<000>" to prevent security issues in the log files.

Please note: if you used double-byte character sets, control character escaping can cause your message to become clobbered. So be sure to leave it unchecked in that case.

### **Default Ruleset Name**

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

### **Please Note**

Updated the OpenSSL components and libraries with the latest Version openssl-0.9.8d.

## **5.3.3 SETP Server**

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

Configured Services > SETP Server 1

Enable: SETP Server 1

Settings are saved.

Save Reset Save & Close

Listener Port: 5432

IP Address: 0.0.0.0

Options

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

Use zlib Compression to compress the data.

Session Timeout: 30 seconds

Notify Sender about Rule Action Errors?

Rule Set to Use

Refresh

*SETP Server Properties*

### Listener Port

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

### IP Address

The SETP server service can be bound to a specific IP 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.

### Enable SSL/TLS

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

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

Options

Under this group box, you can see three more options as discussed below:

### **Use zLib Compression to compress the data**

When enabled, MonitorWare Agent decompresses the zLib compressed data sent by the SETP senders. It is still be able to receive normal data. zLib compression is useful to reduce traffic in WAN environments.

### **Session Timeout**

It controls how long a session is to be opened from the server side.

### **Notify Sender about Rule Action Errors?**

Enable this option to communicate the outcome of an action back to the the sender of the SETP message.

This communicates back the status of actions carried out on the receiver to the sender of the event. In essence, the sender system will know if the action failed or succeeded on the remote machine. It can then act exactly like the action was carried out on the local machine. The exact handling of failure states is depending on the event source.

An example: you have a machine running an EventLog Monitor and sending these events via SETP, and on the other side have all incoming events written into a database. If the database would be offline and the events not being written into it, the SETP server would return as the last message that the action failed (as long as this option is enabled) and generate a error event with ID 1005 (and generate a Success Event with ID 1012 if successful again). The sender would then halt and retry sending the event. This is because SETP is built somehow like TCP which ensures data transfer, but additionally can return a status to the sender if the following action was successful.

This happens because the event log monitor (as well as the file monitor and others) is a restartable event source. It uses the outcome of actions to decide if the action is to be retried in another run of the same source. Other event sources have different behavior. The syslog server, for example, does not retry failed actions. This is due to the lossy nature of syslog, in which losing syslog messages is explicitly permitted (and favourable over taking up too many system resources by trying to buffer them).

**Please Note:** If you enable this feature, older MonitorWare Agent Versions (4.2.x and below, as well as WinSyslog 7.2.x and EventReporter 8.2.x and below) may have trouble sending data over SETP once a Rule Exception occurs! If you intend to use this feature, make sure all MonitorWare Agent Installations are at least Version 4.3.x (This applies for WinSyslog 7.3.x and EventReporter 8.3.x as well).

### **Default Ruleset Name**

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

### 5.3.4 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.

The screenshot shows the 'Heartbeat Properties' dialog box. At the top, there is a checked checkbox for 'Enable: Heartbeat'. To the right are buttons for 'Save', 'Reset', and 'Save & Close'. Below this is a status bar that says 'Settings are saved.' The main area is divided into sections. The first section is 'Message that is send during each heartbeat' with a text area containing 'I am still running gg'. The second section is 'Heartbeat clock (SleepTime)' with a dropdown menu set to '1 minute'. The third section is 'General Values' with four fields: 'Syslog Facility' set to 'LOCAL0 (16)', 'Syslog Priority' set to 'INFO (6)', 'Resource ID' (empty), and 'Syslog Tag Value' set to 'MwHeartbeat'. The final section is 'Rule Set to Use' with a dropdown menu set to 'Defaults' and a 'Refresh' button.

*Heartbeat Properties*

#### Message to Send

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

#### Sleep Time

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

#### Syslog Facility

The Syslog facility to be assigned to events created by the heartbeat service. Most

useful if the message shall be forwarded to a syslog server.

### **Syslog Priority**

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

### **Syslog Tag Value**

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

### **Resource ID**

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

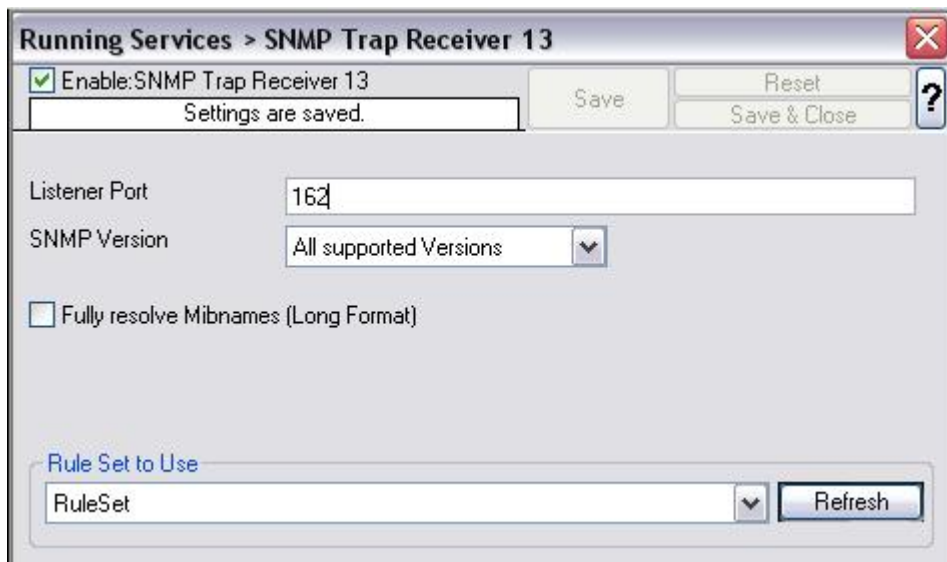
### **Default Ruleset Name**

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

## **5.3.5 SNMP Trap Receiver Service**

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

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



*SNMP Trap Receiver Properties*

### **Listener Port**

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

### **SNMP Version**

Can be used to restrict the SNMP versions. The available values are:

1. All Supported Versions (i.e. SNMP Version 1 and SNMP Version 2c only)
2. SNMP Version 1 only
3. SNMP Version 2c only

### **Fully Resolve Mibnames (Long Format)**

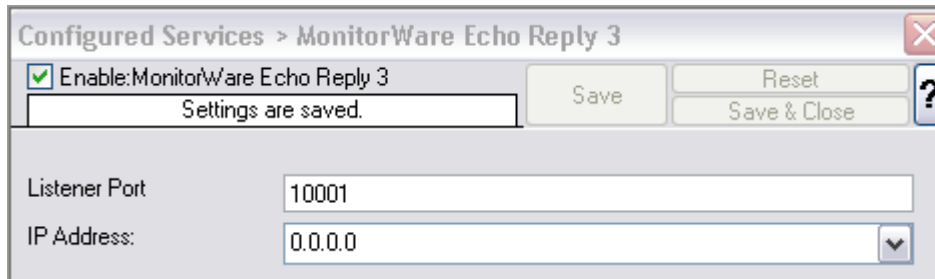
This Option fully resolves the Mibnames like in the Client Mibbrowser Application.

### **Rule Set to Use**

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

### 5.3.6 MonitorWare Echo Reply

The Echo Reply service is used on each of the installed WinSyslog. A central agent running the MonitorWare Agent is using the echo request and instructs to poll each of the other WinSyslog 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 WinSyslog service is done.



*MonitorWare Echo Reply Properties*

#### Listener Port

Specify the listener port here.

#### IP Address

The MonitorWare Echo Reply service can be bound to a specific IP 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.

## 5.4 Filter Conditions

### 5.4.1 Filter Conditions

Filter conditions specify **when** to apply a rule. If the filter condition evaluates to true, the rule containing those conditions is treated as matching and the actions specified in that rule 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](mailto:support@adiscon.com), and we look into it!

## 5.4.2 Filter Conditions - Brushup

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

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

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

**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.

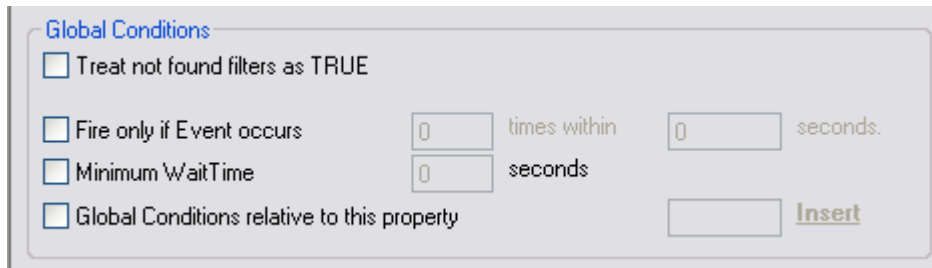
**NOT** – Negates a value. Example: NOT A: If A is true, the outcome is false and vice versa. There can only be a single operand for a NOT operation.

**TRUE** – Returns true.

**FALSE** – Returns false.

### 5.4.3 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.



Global Conditions

Treat not found filters as TRUE

Fire only if Event occurs  times within  seconds.

Minimum WaitTime  seconds

Global Conditions relative to this property

*Filter Form - General 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.

## **5.4.4 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.

## **5.4.5 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 =" and "Range Match".

**Number**

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

**Boolean**

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

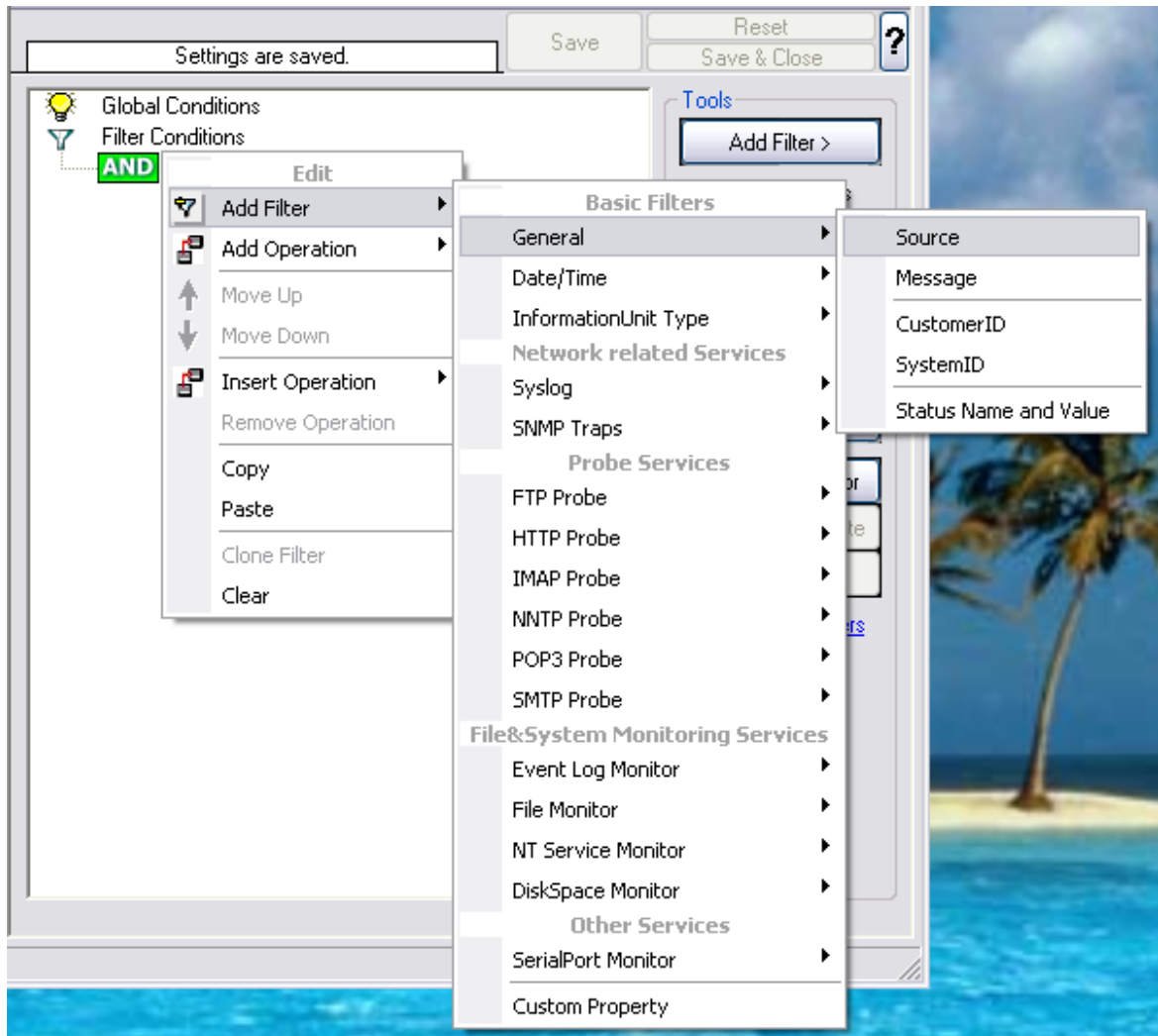
**Time**

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

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

### 5.4.6 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.

#### 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 operation. 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 receive 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.

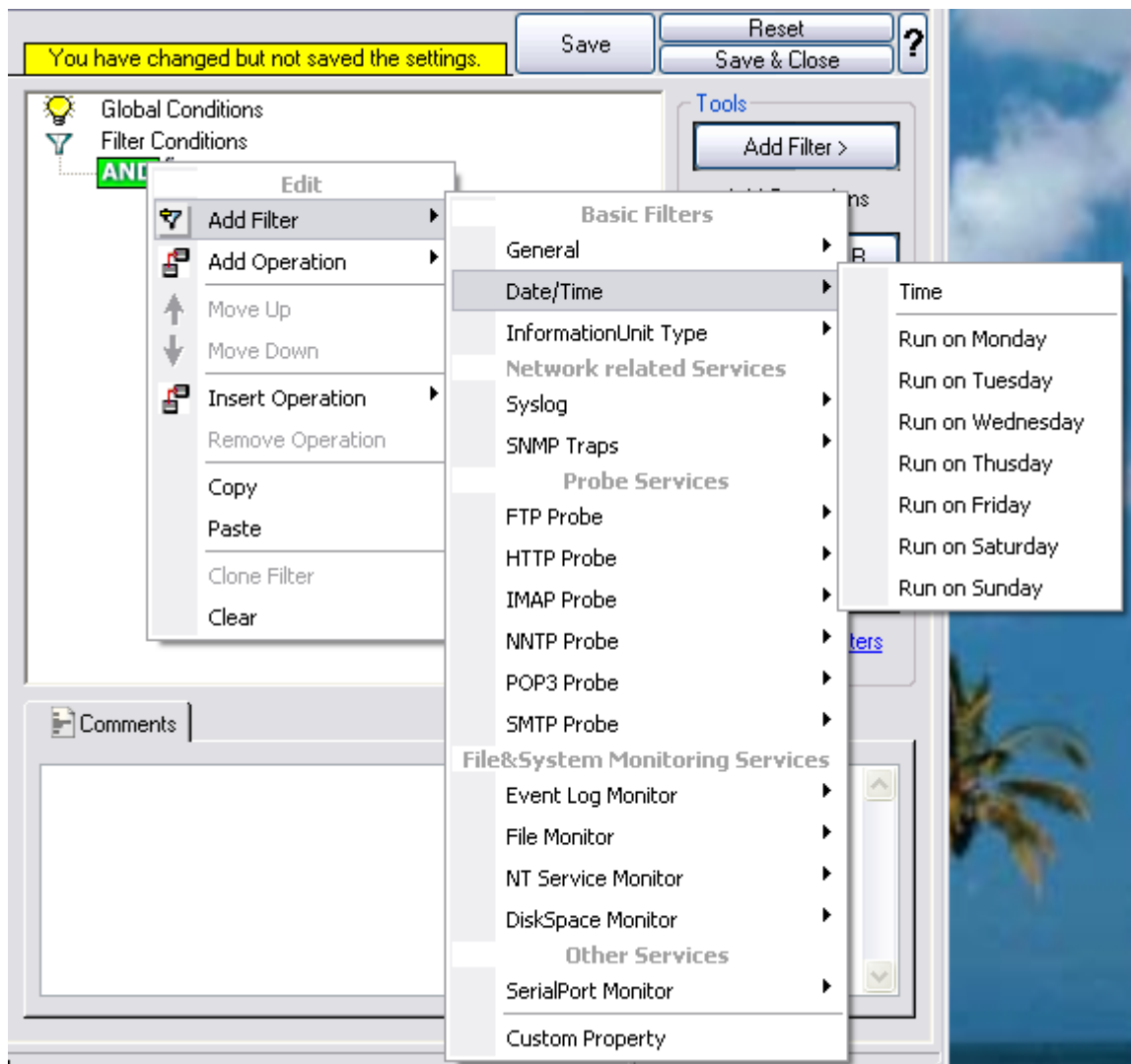
SystemID (Type=Number).

### **Status Name and Value**

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

### 5.4.7 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).

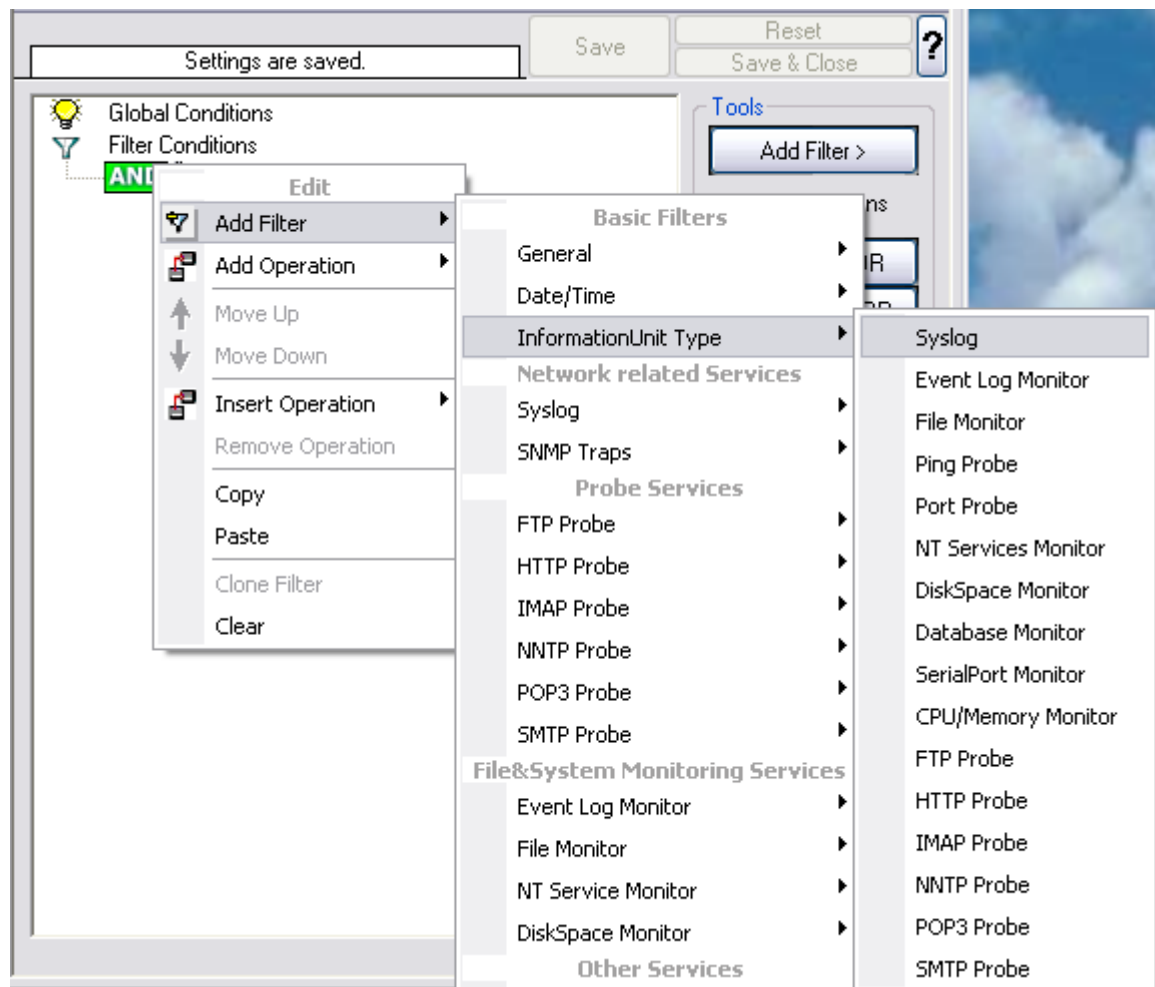
## Weekdays

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)

### 5.4.8 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).



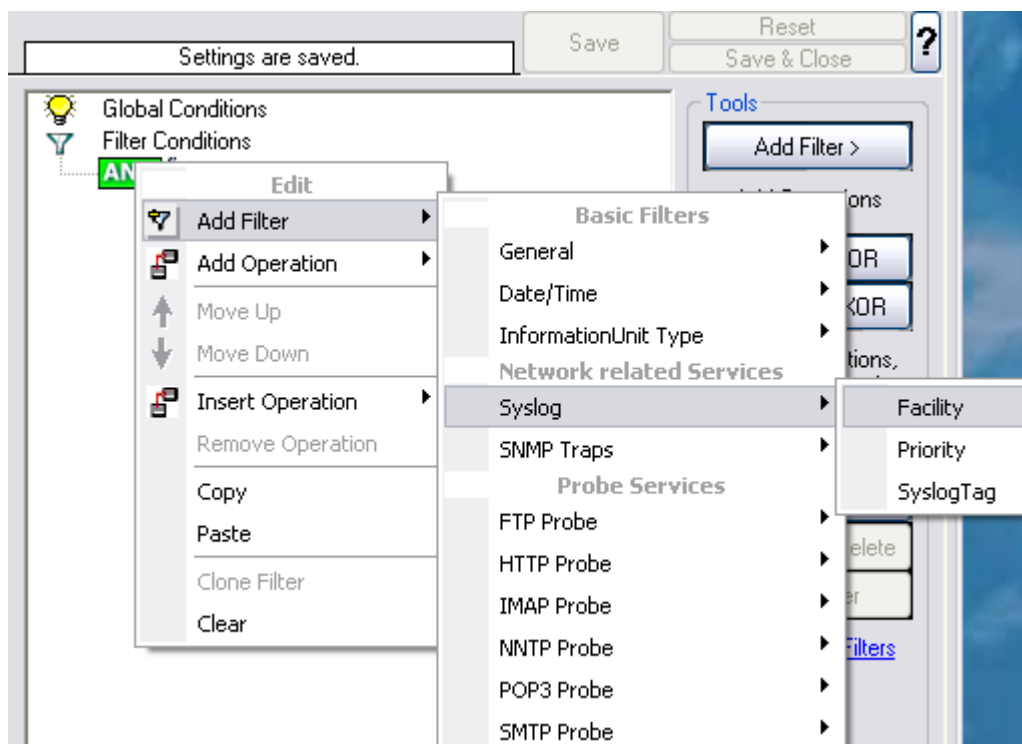
*Filter Conditions - InformationUnit Type*

The following filters are available:

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

### 5.4.9 Syslog

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



*Filter Conditions - Syslog*

#### **Syslog Facility**

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

This filter is of type number.

### **Syslog Priority**

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

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

This filter is of type number.

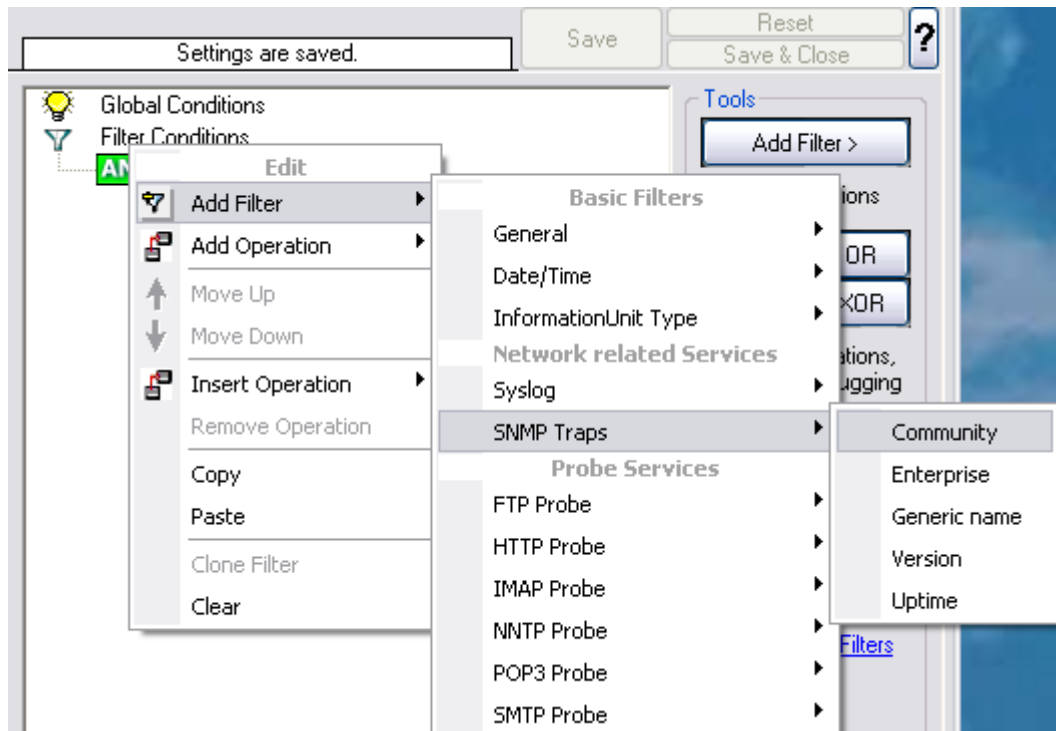
### **Syslog Tag**

This filter is of type string.

## **5.4.10 SNMP Traps**

Using SNMP Traps, MonitorWare Agent 3.0 now can be used to manage and monitor all sorts of equipment including computers, routers, wiring hubs, toasters and jukeboxes.

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



*Filter Conditions - SNMP Traps*

### **Community**

It corresponds to the respective SNMP entity.

This filter is of type string.

### **Enterprise**

It corresponds to the respective SNMP entity.

This filter is of type string.

### **Generic name**

It corresponds to the respective SNMP entity.

This filter is of type string.

### **Version**

It corresponds to the respective SNMP entity.

This filter is of type number.

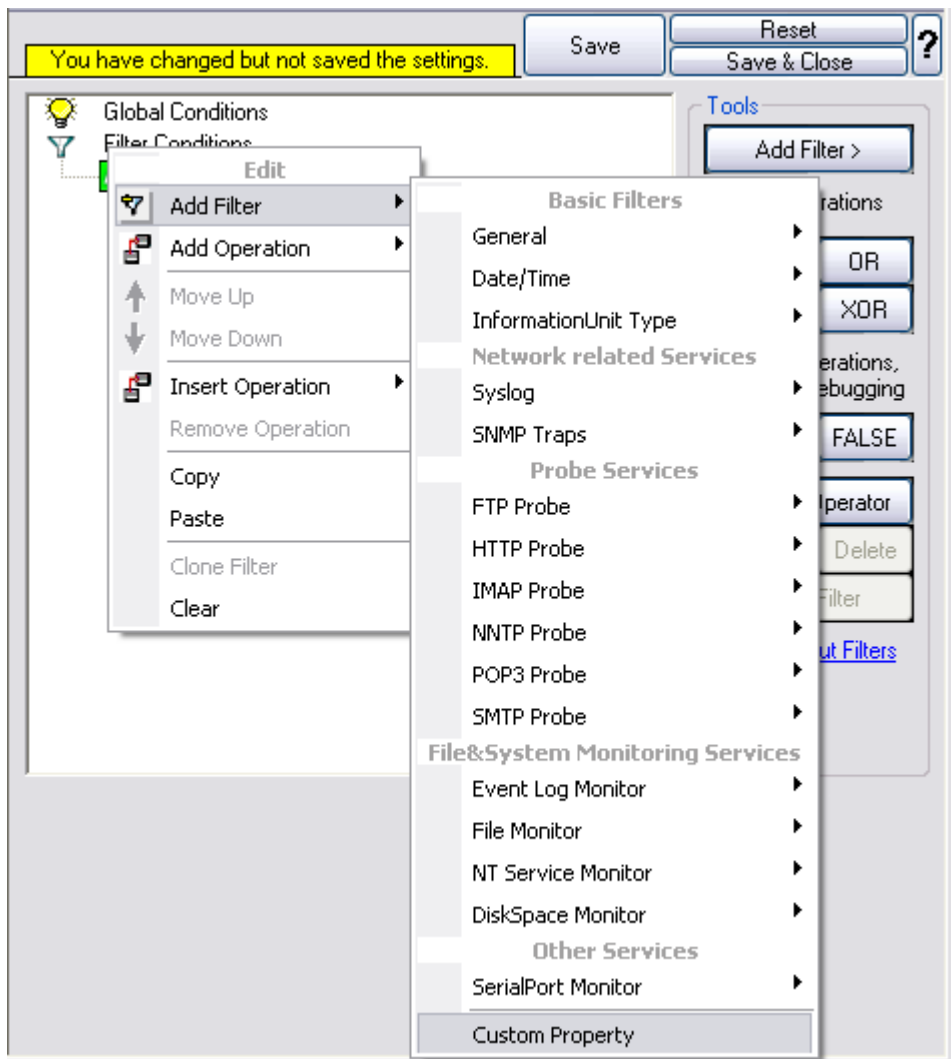
## Uptime

It corresponds to the respective SNMP entity.

This filter is of type string.

### 5.4.11 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.

#### 5.4.12 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.

## 5.5 Actions

### 5.5.1 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.**

### 5.5.2 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:

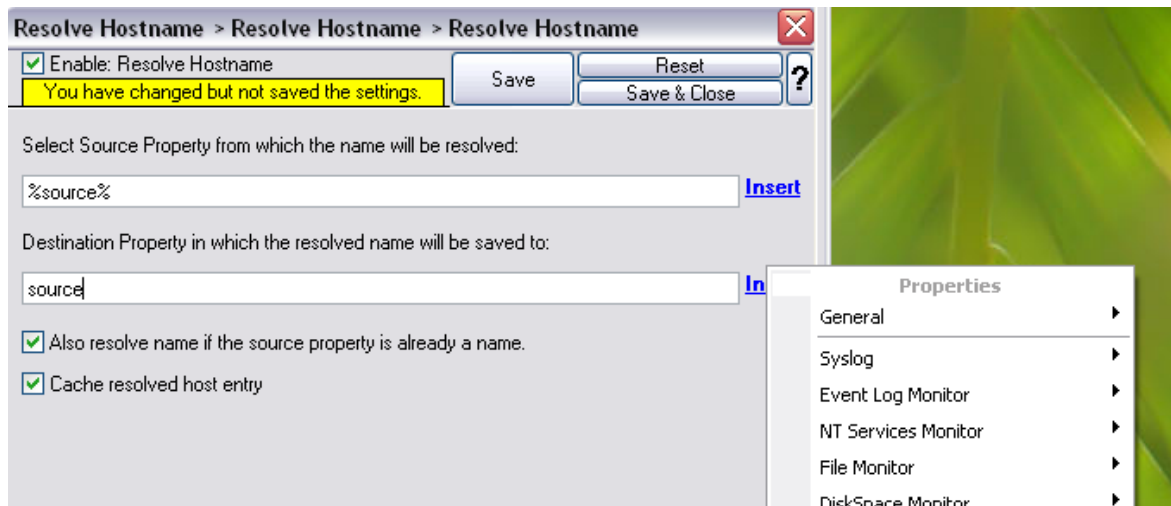


Figure1: Resolve Hostname Action with opened up "Insert" Menu.

### Select Source Property from which the name will be resolved:

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:

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.

Activates the feature that the name will also be resolved if there is already a source property with that name.

### Cache resolved host entry

If activated this will, as it says, cache the resolved host entry.

## 5.5.3 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 NT Event Log in this case). We recommend copying the file when accessing it at runtime - or use notepad.exe, which does not place file-locks on the files it opens.

**The filename is build as follows:**

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

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

The screenshot shows a configuration dialog for WinSyslog. At the top, there is a status bar that says "Settings are saved." and buttons for "Save", "Reset", and "Save & Close". A "Configure for..." link is visible in the top right. The main area is divided into two sections: "Filename related options" and "General file options".

**Filename related options:**

- Enable Property replacements in Filename (with a "Browse" button)
- File Path Name: C:\temp (with an "Insert" button)
- File Base Name: MonitorWare (with an "Insert" button)
- File Extension: log
- File format: Adiscon (dropdown menu)
- Custom Line Format: %msg%%\$CRLF% (with an "Insert" button and scroll arrows)

**General file options:**

- Create unique filenames
- Use Circular Logging
- Include Source in Filename
- Number of Logfiles: 10
- Use UTC in Filename
- Maximum Filesize (KB): 4096
- Segment files when the following filesize is reached (KB): 4096
- Use XML to Report
- Use UTC for Timestamps
- Include Date and Time
- Include Date and Time reported by Device
- Include Syslog Facility
- Include Source
- Include Syslog Priority
- Include RAW Message
- Include Message

*File Logging Options*

**Configure For ...**

If you want to generate the reports on log files using [Monilog](#) or [MonitorWare Console](#), then its absolutely necessary that the log files are in a specific format. This option allows you to configure the file logging format for Monilog and MonitorWare Console.

If the log file entries are not in the correct format for MonitorWare Console (for PIX or

Windows Reports), then it writes error messages for first 50 lines in Windows event log and ignores them for the generation of report, resulting in a generation of empty report.

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

Following three options are available:

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

### **Configure for MonitorWare Console PIX Reports**

This option changes the file logging format of MonitorWare Agent to the correct format expected by MonitorWare Console for PIX report generation.

### **Configure for MonitorWare Console Windows Reports**

This option changes the file logging format of MonitorWare Agent to the correct format expected by MonitorWare Console for Windows report generation.

### **Configure for Monilog**

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

### **Enable Property replacements in Filename**

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

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

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

If your source is 10.0.0.1, that 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 property 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.**

### **File Path Name**

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

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

**Event properties are described in the property replacer section.**

### File Base Name

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

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

### File Extension

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

### File Format

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

The "Raw Syslog message" format writes raw Syslog format to the log file. That is, each line contains the Syslog message as of 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.**

### Custom Line Format

Custom 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%".

**Event properties are described in the property replacer section.**

### **Create unique Filenames**

If checked, MonitorWare Agent 3.0 creates a unique file name for each day. This is done by adding the current date to the base name (as can be seen above).

If left unchecked, the date is not added and as such, there 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**

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**

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.**

### **Use Circular Logging**

When enabled log files are created and over written in a cycle.

### **Number of Log files**

Once the last logfile is reached, circular logging begins and over write the first log file again.

### **Maximum File size**

Max filesize of a log file, once this size is reached a new logfile is created.

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

Files are segmented when the defined file size is reached. The file name will have a sequence number appended (\_1 to \_n).

### **General file options**

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

#### **Use XML to Report**

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

#### **Use UTC for Timestamps**

Please see the definition of UTC above at "Use UTC in Filename". This setting is very similar. If checked, all time stamps 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>**

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 MonitorWare Agent 3.0 – totally unmodified, received it. This might be useful if a third party application is expecting raw Syslog entries. The message itself is just that part of the Syslog message that is being parsed as message that is without e.g. host information or a tag value. Please note that we recommend selecting only one of these options, as otherwise two message fields 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.

#### 5.5.4 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 for Adiscon [MonitorWare Console](#) product as well as the 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.

RuleSet 1 > Write To OLEDB Database > Write To Database 2

Enable: Write To Database 2  
Settings are saved. Save Reset Save & Close ?

DSN: [ ] Data Sources (ODBC) Create Database  
 User-ID: [ ] Password: [ ]  
 Table Name: SystemEvents  Enable Encryption  
 SQL Statement Type: INSERT Statement  
 Output Encoding: System Default  
 Connection Timeout: 60 seconds

Detail data logging  
 Enable Detail Property Logging  
 Detail data TableName: SystemEventsProperties  
 Maximum value length (Bytes): 512

Insert Delete  
 Fieldname: Facility  
 Fieldtype: int  
 Fieldcontent: syslogfacility **Insert**

Fieldname	Fieldtype	Fieldcontent
<b>Facility</b>	<b>int</b>	<b>syslogfacility</b>
Priority	int	syslogpriority
FromHost	varchar	source
Message	text	%msg%
ReceivedAt	DateTime UTC	timegenerated
DeviceReportedTime	DateTime UTC	timereported
CustomerID	int	CustomerID
SystemID	int	SystemID
SysLogTag	varchar	syslogtag

### Database Logging Options

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 (like MonitorWare Console) need the database contents as specified. As such, malfunctions may occur if you modify the database assignments and then use these tools.

The "**fieldname**" is the database column name. It can be any field inside the table. The provided names are those that Adiscon's schema uses - you can add your own if you have a need for this. "**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. Finally, the "**Fieldcontent**" is the event property. For a complete list of supported properties, see **Event properties**.

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. You can move rows up and down by using the arrow keys. Moving them up and down is cosmetic - it will not affect the write to database action.

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%".

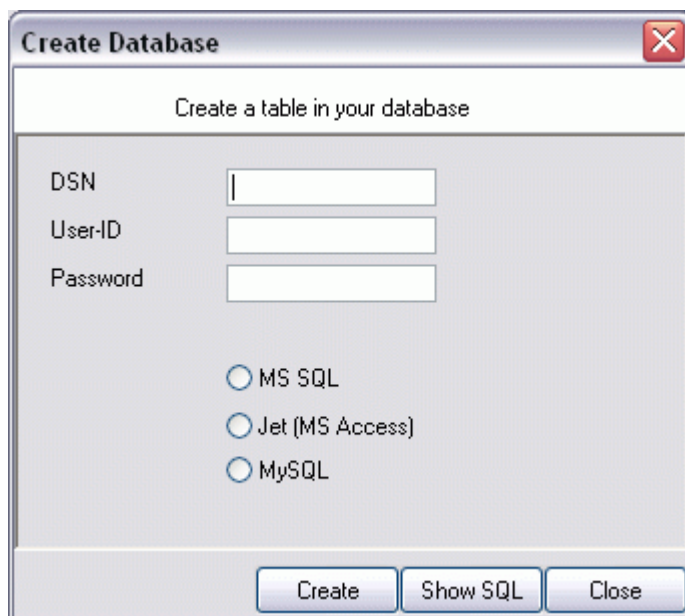
The rest of this section describes the labelled paramters.

### Data Sources (ODBC)

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).

### Create Database

If you click on this button, it opens a form as shown below:



*Create Database Form*

In this form, you have to provide your DSN, User-ID, Password and select your underlying database. After this you have to click Create button to create the table in your database. You can also click Show SQL button to see the SQL query that is to be executed. Close button is to close the form.

### DSN

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

and removed.

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

### User-ID

The User-ID used to connect to the database. It is dependant on the database system used if it 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

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

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.

### Table Name

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

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

### SQL Statement Type

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**

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.

### **Connection Timeout**

Defines the Timeout for the connection

### **Enable Detail Property Logging**

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.

### **Connection Retry**

If a connection is broken, MWAgent gracefully shutdowns the DB Connection and tries to reopen the Connection with the next Actioncall.

## **5.5.5 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 [MonitorWare Console](#) product as well as the 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.

Fieldname	Fieldtype	Fieldcontent
<b>Facility</b>	<b>int</b>	<b>syslogfacility</b>
Priority	int	syslogpriority
FromHost	varchar	source
Message	text	%msg%
ReceivedAt	DateTime UTC	timegenerated
DeviceReportedTime	DateTime UTC	timereported
CustomerID	int	CustomerID
SystemID	int	SystemID

#### OLEDB Database Action Options

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. You only need to keep in mind that Adiscon analysis products (like MonitorWare Console) need the database contents as specified. As such, malfunctions may occur if you modify the database assignments and then use these tools.

The **"fieldname"** is the database column name. It can be any field inside the table. The provided names are those that Adiscon's schema uses - you can add your own if you have a need for this. **"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. Finally, the "**Fieldcontent**" is the event property. For a complete list of supported properties, see **Event properties**.

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. You can move rows up and down by using the arrow keys. Moving them up and down is cosmetic - it will not affect the write to database action.

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.

### **Configure Data Source**

If you click on this button, it starts the OLEDB administrator of the operating system where you can add, edit or remove a data source(s).

### **Verify Database Access**

This button verifies if your indicated data source works fine.

### **Main Table Name**

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

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

### **SQL Statement Type**

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**

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.

## Connection Timeout

Defines the Timeout for the connection

## Enable Detail Property Logging

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.

## Connection Retry

If a connection is broken, MWAgent gracefully shutdowns the DB Connection and tries to reopen the Connection with the next Actioncall.

### 5.5.6 Event Log options

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

The screenshot shows a configuration window for event log options. At the top, there is a checked checkbox labeled "Enable: Write To EventLog". Below it, a status bar displays "Settings are saved." and buttons for "Save", "Reset", and "Save & Close". A help icon (?) is also present. The main area contains several fields:

- Replace Event Log Source
- Custom EventSource: %source%
- EventType: AUDIT\_SUCCESS (dropdown menu)
- EventID: 10000
- Message to Log: %msg%

There are "Insert" buttons next to the Custom EventSource and Message to Log fields.

## Event Logging Options

### Replace Event Log Source

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

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

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

### Custom Event Source

EventSource is now fully configurable with all possibilities the property engine gives you. **Please note that content of this field can be configured. Event properties are described in the [property replacer section](#).**

### EventType

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

### EventID

The ID to be used when writing to the event log. Different IDs can be used to provide other processes with a consistent interface to specific messages. WinSyslog does not restrict the IDs that can be used. However, if an ID is written that is not registered with the operating system, Windows event viewer places an error message pointing this out before the actual message text. To avoid this text, event IDs 10,000 to 10,100 have been registered with the OS. We highly recommend that these IDs 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

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.

### 5.5.7 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.

RuleSet 1 > File Logging > Forward via Email 2

Enable: Forward via Email 2

Settings are saved. Save Reset Save & Close ?

Mailservers

Mailserver 127.0.0.1 Port 25

Enable Backup Server, if first Mailserver fails

Backup Mailserver 127.0.0.1 Port 25

Sender sender@example.com

Recipient receiver@example.com

Use legacy subject line processing [Insert](#)

Subject Email for you

Mail Priority Normal Priority

[Insert](#)

Mail Message Format

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

Session Timeout (0 - 4000 ms) 0

Output Encoding System Default

Use SMTP Authentication

SMTP Username

SMTP Password

Include message / event in email body

Use XML to Report

*Forward Email Properties*

#### Mailserver

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

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

### Backup Mailserver

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.

### Port

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.

### Sender

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

### Recipient

The recipient emails are addressed to. To send a message to multiple recipients, enter all recipient's email addresses in this field. Separate addresses 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.

### Subject

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**

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.

### **Use legacy subject line processing**

This checkbox specifies which type of subject line processing will be done. If it is checked, the old-style processing using single character replacement sequences is applied. If it is left unchecked, the far more 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 received 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 characters from the message text (position 1 to 15).

### **Mail Message Format**

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

### **Session Timeout**

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

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

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

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

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

### **Use SMTP Authentication**

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

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

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

We recommend using authentication if it is available. Even when the current server configuration allows unauthenticated relay, this potentially will change in the future

(as the SPAM problem grows). If you already use authentication, such a server configuration change will not affect you. Otherwise, it will disrupt mail service.

### **Include message / event in email body**

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

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

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

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

### **Use XML to Report**

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

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

## 5.5.8 Forward Syslog Options

This dialog controls Syslog forwarding options.

RuleSet 1 > Forward Syslog > Forward Syslog

Enable: Forward Syslog

Settings are saved. Save Reset Save & Close ?

Syslog Server

Syslog Port 514

Protocol Type UDP

Use zLib Compression to compress the data.

Compression Level Best Compression

Process message while relaying

Message Format %msg% Insert

Output Encoding System Default

Add Syslog Source when forwarding to other Syslog servers

Use XML to Report

Forward as MWAagent XML representation code

*Forward Syslog Properties*

### Syslog Server

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

### Syslog Port

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

## Protocol Type

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

## Use zLib Compression to compress the data

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.

## Process Message while relaying

This setting indicates if the message should be processed while it is being forwarded. Processing is necessary if you need to reformat the message, e.g. by using the property replacer. If you just intend to forward the message "as is" to another syslog server, uncheck this box. That will guarantee the message is not reformatted and it also offers the best possible performance. In many cases, leaving the box unchecked is a good idea.

## Message Format

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

**Please note that the message content of the Message field can be configured. Event properties are described in the [property replacer section](#).**

## Output Encoding

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

## Add Syslog Source

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

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

## Use XML to Report

If 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**

MonitorWare supports a specific XML-Representation of the event. If it is checked, that XML representation is used. It provides additional information (like informationunit type, original source system, reception time & many more) but is harder to read by a human. At the same time, it is obviously easier to parse. **Please note that this option is only "experimental" and is not an official standard.**

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

### **Note on Using Syslog Compression**

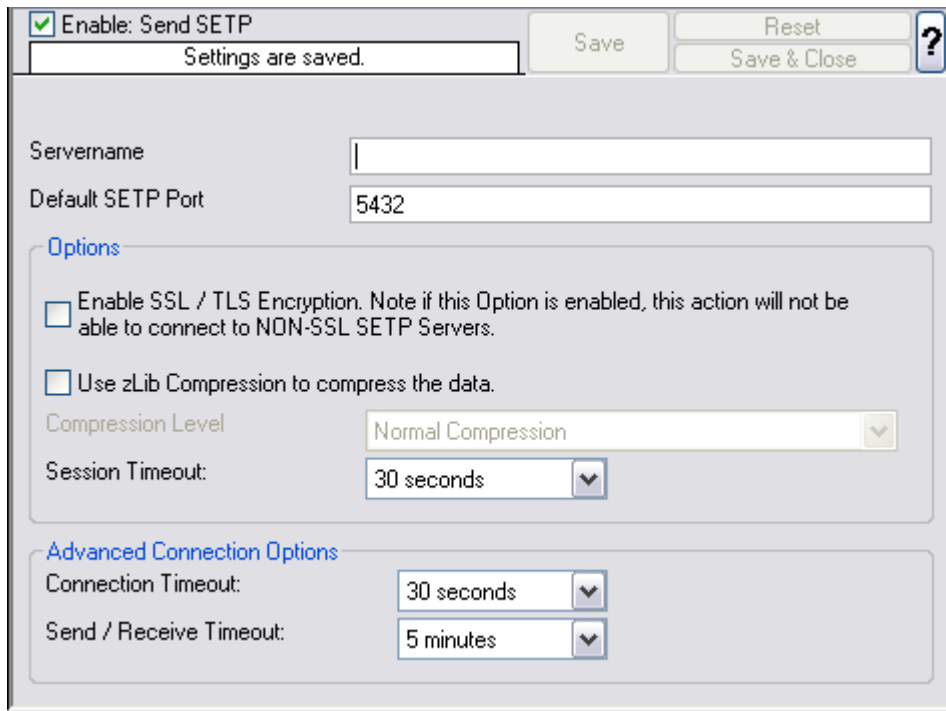
Compressing syslog messages is an **experimental** feature. There is only a very limited set of receivers who is able to understand that format. Turning on compression can save valuable bandwidth in low-bandwidth environments. Depending on the message, the saving can be anything from no saving at all 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 compression 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.

### **5.5.9 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

*Send SETP Dialog*

### Servername

The MonitorWare Agent 3.0 sends SETP to the server / listener under this name.

### Default SETP Port

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

**Please note: The SETP port configured here must match the port configured at the listener side (i.e. MonitorWare Agent 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 NT Event Log.**

### Options

Under this group box, you can see different options as discussed below:

#### Enable SSL/TLS

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

#### Use zLib Compression to compress the data

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

### Compression level

Higher level results in better compression but slower performance.

### Session Timeout

The maximum time a session to a SETP server is to be kept open.

### Advanced Connection Options

In this group box, you can find the options discussed below:

#### Connection Timeout

Maximum time a connection can take to connect or disconnect.

#### Send / Receive Timeout

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.

## 5.5.10 Net Send

This dialog controls the net send options.

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

The screenshot shows a dialog box titled "Net Send Dialog". At the top left, there is a checked checkbox labeled "Enable: Net Send". To its right is a yellow warning box with the text "You have changed but not saved the settings.". Further right are four buttons: "Save", "Reset", "Save & Close", and a help icon (?). Below these is a "Target" label and a text box containing "ADMINWRK". Underneath is a "Message to send" label and a large text area containing "Test Message: Error 1337 occurred.". To the right of the text area is an "Insert" button.

*Net Send Dialog*

## Target

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

## Message to Send

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

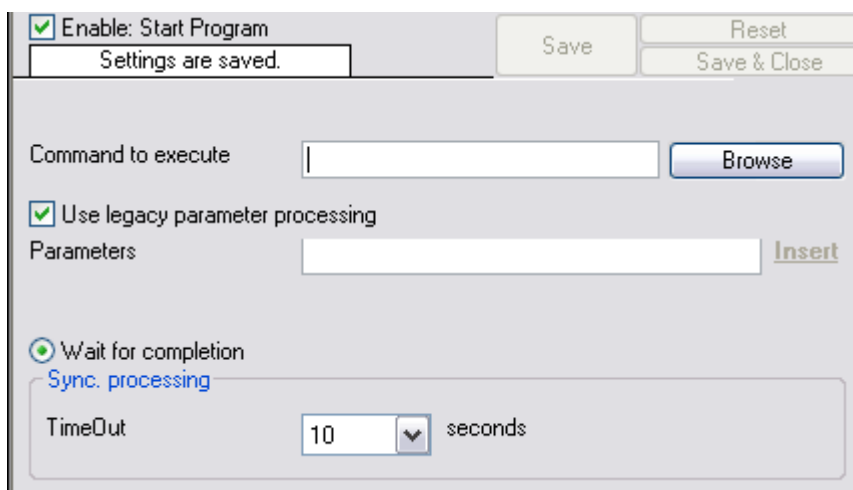
**Please note that the message content of the Message to send field can now be configured. Event properties are described in the [property replacer section](#).**

### 5.5.11 Start Program

This dialog controls the start program options.

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

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



*Start Program Dialog*

## Command to execute

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

When enabled, old style parameter processing is used. Otherwise all properties can be used.

### Parameters

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

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

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

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

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

### Time Out

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.

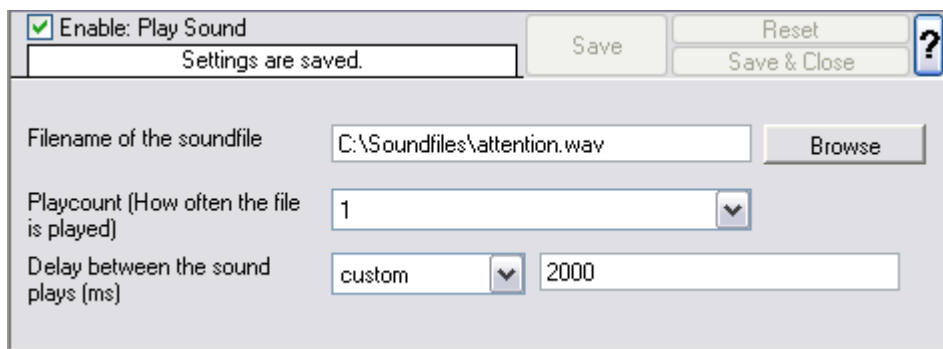
The external program should only run for a limited amount of time. If it would block for some reason, the agent would be prevented from carrying out any further processing. As such, a timeout value must be specified. If the program still runs after the configured timeout, the rule engine cancels it,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

### 5.5.12 Play Sound

This action allows you to play a sound file.



*Play Sound Dialog*

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

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

#### Filename of the Soundfile

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

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

#### Playcount

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

**Please note: Playing sounds is performance intense and MonitorWare Agent 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

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

### 5.5.13 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.

Enable: Send to Communications Port  
 Settings are saved. Save Reset Save & Close

Timeout limit: 1 minute

To which Port do you want the message to send? and Settings\\All Users\\Desktop\\\*.pdf

**Port Settings**

Bits per second	57600
Data bits	8
Parity	NO PARITY
Stop bits	1 stop bit
DTR Control Flow	DTR_CONTROL_DISABLE
RTS Control Flow	RTS_CONTROL_DISABLE

Message to send:  Insert

*Send to Communications Port Options*

### Timeout Limit

The maximum time allowed for the device to accept the message. If the message could not be send within that period, the action is aborted. Depending on the device,

it may be left in an unstable state.

### Port to Send To

Specify the port to which your device is being attached. Typically, this should be one of the COMx: ports. The listbox shows all ports that 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

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

### Bits per Seconds

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

### Databits

Databits defines that how many bits you want to send and receive to the communication port.

### Parity

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

1. Even
2. Mark
3. No parity

4. Odd
5. Space

### Stop bits

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

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

### DTR Control Flow

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

1. DTR\_CONTROL\_DISABLE - Disables the DTR line when the device is opened and leaves it disabled.
2. DTR\_CONTROL\_ENABLE - Enables the DTR line when the device is opened and leaves it on.
3. DTR\_CONTROL\_HANDSHAKE - Enables DTR handshaking.

### RTS Control Flow

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

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

### Message to Send

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

**Please note that the message content of the Message field can now be configured. Event properties are described in the [property replacer section](#).**

### 5.5.14 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*

#### Property Name

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

#### Set Property Type

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

#### Insert

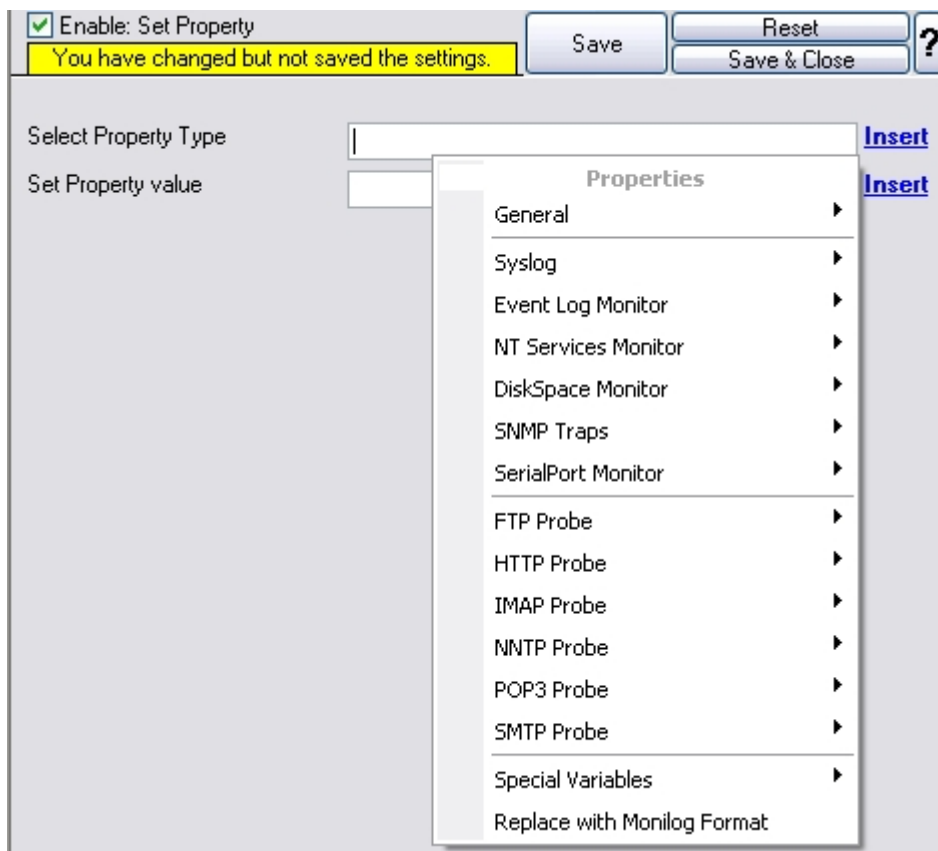
Click here to get a list of predefined variables/values to insert.

### 5.5.15 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

Select the property type to be changed. The list box contains all properties that can be changed. By default it is set to nothing.

#### Set Property Value

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 used in the filter conditions and actions.

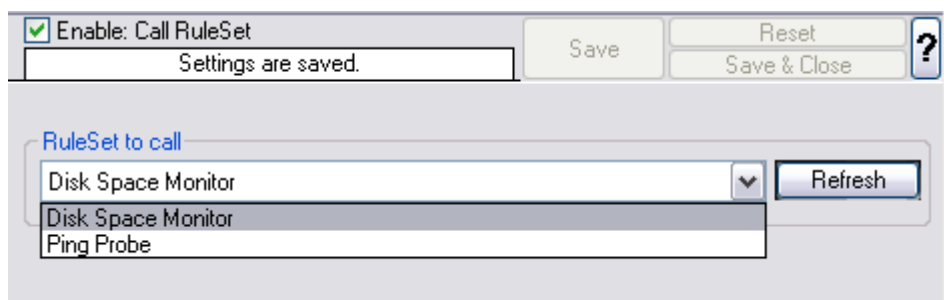
### 5.5.16 Call RuleSet

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 goes 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**

Select the Ruleset to be called.

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

### 5.5.17 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.

[Click here](#) to see an example about the Discard action.

### 5.5.18 Post-Process Event

The post process action allows you to re-parse a message after it has been processed e.g. **Tab Delimited** format.

Such re-parsing is useful if you either have a non-standard Syslog format or if you would like to extract specific properties from the message.

The post process action takes the received message and parses it according to a parse map. The parse map specifies which properties of which type are present at which position in the message. If the message actually matches the parse map, all properties are extracted and are set as part of the event. If the parse map does not match the message, parsing stops at the first-non matching entry.



## Property

The property name that is to be parsed. The list box is pre-populated with standard and event properties. However, you can add any property name you like. If you create your own properties, we highly recommend prefixing their name with "u-" so that there will be no duplicates with standard properties. Adiscon will never prefix any properties with "u-". For example, if you would like to create a custom property "MyProperty", we highly suggest that you use the property name "u-MyProperty" instead.

The property name "Filler" is reserved. Any values assigned to the Filler-property will be discarded. This is the way to get rid of fill-characters that you do not really need.

## Type

This is the format that will be parsed from the message. For example, an integer type will parse one integer from the message while a word type will parse the next word.

## Value

Some types need an additional value. If that is needed, you can provide it here.

## Message Preview

This is a read-only box. It shows a hypothetical message that would match the configured parsing rules.

## Parsing log messages

This article describes how to parse log message via "Post-Process". It illustrates the logic behind Post-Process action.

### Get relevant information from logs

Log files contain a lot of information. In most cases only a small part of the log message is of actual interest. Extracting relevant information is often difficult. Due to a variety of different log formats a generic parser covering all formats is not available.

Good examples are firewalls. Cisco PIX and Fortigate firewalls both use syslog for logging.. But the content of their respective log messages are very different. Therefore a method is needed to parse the logs in a generic way. Here Post-Process action of Adiscon's MonitorWare comes into play.

### Tool kit for parsing

Post-Process action provides an editor for creating a log format template. A template consists of as many rules as necessary to parse out the relevant information.

### Determine necessary information

In order to parse out information it is vital to know the exact structure of the message. Identifying the position of each relevant item is essential. Assuming for auditing purposes the following items are needed:

*Timestamp | Source IP-Address | SyslogTag | MessageID | Username | Status | Additional Information*

A sample message looks like:

*Mar 29 08:30:00 172.16.0.1 %Access-User: 12345: rule=monitor-user-login user=Bob status=denied msg=User does not exist*

In order to extract the information let us examine each item within the message. Splitting the message makes it easier to explain. So here we go.

Pos = Position of the character.

\*p = Points to the position the parser stands after parsing the rule.

Log = Message subdivided into its characters.

Pro = Property. In the term of Adiscon a property is the name of the item which is parsed out.

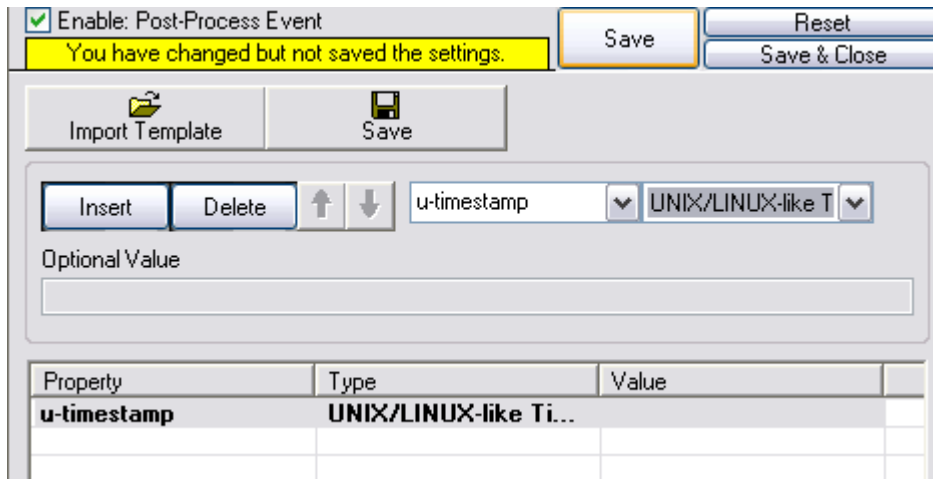
Pos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Log	M	a	r		2	9		0	8	:	3	0	:	0	0		1	7	2	.
*p	*																			
Pro																				

Note that at beginning of the parse process the parser's pointer points to the first character. Each parse type starts parsing at the current position of the pointer.

### Parsing out a Timestamp

The first identified item is a so called Unix/Timestamp. It has always a length of 15 characters. 'UNIX/LINUX-like Timestamp' parse type exactly covers the requirement to parse this item. Therefore insert a rule and select 'UNIX/LINUX-like Timestamp' type. This rule parses out the timestamp and moves the pointer to the next character after the timestamp. Name the property 'u-timestamp' [\[1\]](#).

**Note:** There is a second timestamp-type, the **ISO-like-timestamp**. It has the format **2006-07-24 13:37:00**.



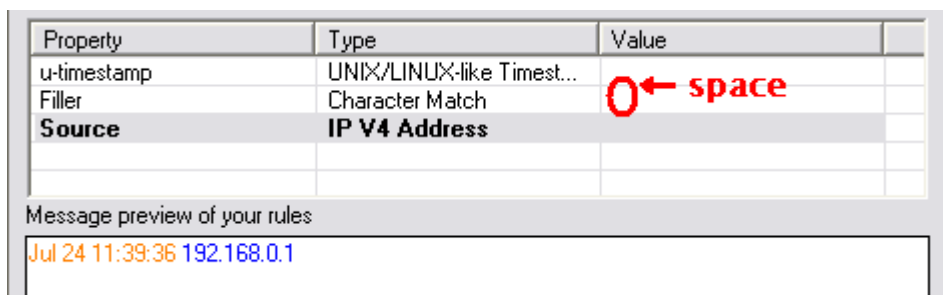
Post-Process Editor: Inserted a 'UNIX/LINUX like timestamp' rule

Pos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Log	M	a	r		2	9		0	8	:	3	0	:	0	0		1	7	2	.
*p																*				
Pro	u-timestamp																			

### Get the IP-Address

Next item is the IP address. Note that after the timestamp follows a space and then the IP address. Therefore insert a 'Character Match' rule with a space as value. Select the 'Filler' [2] property for this rule. 'Character Match' requires a user defined value. This parse type compares the given value with the character at the current position of the message. The character has to be identical with the given value otherwise the parse process will fail. After applying this parse type the parse pointer is moved to the position immediately after the given value. In our sample this is the start position of the IP Address (Pos 17).

After that the address can be obtained. Place in a 'IP V4 Address' type. This type parses out a valid IP regardless of its length. No need to take care about the characters. Select 'Source' property or name it to whatever you prefer. The parser will automatically move the pointer to the position next to the address.



Note the value of 'Character Match' rule is a space.

Pos	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
Log	0		1	7	2	.	1	6	.	0	.	1		%	A	c	c	e	e	s
*p													*							
Pro	Filler	Source																		

**Obtain the syslogtag**

Behind the IP it is a blank followed by a percent sign. The percent indicates that the syslogtag is following. To move the pointer to the syslogtag position once again a 'Character Match' rule is necessary. It has to match the space (actual position of the pointer) and the percent sign. This content is not needed therefore assign it to the 'Filler' property.

A colon is immediately behind the syslogtag. So all characters between the percent sign and the colon are needed. The 'UpTo' type can do this job. Insert an 'UpTo' rule. As value enter ':' (without the quotes) and select the syslogtag property. Note that after parsing the pointer stands on the first character of the 'UpTo' value.

Property	Type	Value
u-timestamp	UNIX/LINUX-like Timest...	
Filler	Character Match	
Source	IP V4 Address	
Filler	Character Match	%
<b>syslogtag</b>	<b>UpTo</b>	:

Message preview of your rules

```
Jul 24 11:45:13 192.168.0.1 %:
```

Pos	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
Log	1		&	A	c	c	e	s	s	-	U	s	e	r	:		1	2	3	4
*p														*						
Pro	Filler	syslogtag																		

Important: It points to the colon not to the blank.

**Take the MessageID**

The next interesting item is the MessageID. Move the pointer to start position of the MessageID part. Again, do this by using a 'Character Match' rule. Keep in mind that the pointer points to the colon. Behind the colon is a space and then the MessageID starts. Thus, the value of the rule has to be ':'.

MessageID consist of numbers only. For numeric parsing the 'Integer' parse type exist. This type captures all characters until a non-numeric character appears. The pointer is moved behind the number. Note that numeric values with decimal dots can not be parsed with this type (because they are not integers). This means trying to

parse 1.1 results in 1, because the dot is a non-numeric value.

Property	Type	Value
u-timestamp	UNIX/LINUX-like Timest...	
Filler	Character Match	
Source	IP V4 Address	
Filler	Character Match	%
syslogtag	UpTo	:
Source	Character Match	← space
<b>u-messageid</b>	<b>Integer</b>	

Message preview of your rules

```
Jul 24 12:19:39 192.168.0.1 %: 12345
```

Pos	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
Log	r	:		1	2	3	4	5	:		r	u	l	e	=	m	o	n	i	t
*p									*											
Pro				u-messageid																

### Find the username and status

Looking at the remainder of the message indicates that the username is not immediately after syslogtag. Thankfully though, the username always starts with 'user='. Consequently the 'UpTo' type can be used to identify the username. To get the start position of the username we have to use 'UpTo' together with 'Character Match'. Remember that 'UpTo' points to the first character of the given value. For this reason the 'Character Match' rule is necessary.

After locating the start position of the username 'Word' parse type can be used. 'Word' parses as long as a space sign is found. Enter 'u-username' as property.

Property	Type	Value
u-timestamp	UNIX/LINUX-like Timest...	
Filler	Character Match	
Source	IP V4 Address	
Filler	Character Match	%
syslogtag	UpTo	:
Source	Character Match	:
u-messageid	Integer	
Filler	UpTo	user=
Filler	Character Match	user=
<b>u-username</b>	<b>Word</b>	

Message preview of your rules

```
Jul 24 12:23:53 192.168.0.1 %: 12345user=user=aWord
```

Pos	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89
Log	i	n		u	s	e	r	=	B	o	b		s	t	a	t	u	s	=	d
*p	Filler		Filler				u-username			*										
Pro																				

**Notice:** After parsing a word the pointer stands on the space behind the parsed word.

The steps to get the status are very similar to the previous one.

Property	Type	Value
u-timestamp	UNIX/LINUX-like Timest...	
Filler	Character Match	
Source	IP V4 Address	
Filler	Character Match	%
syslogtag	UpTo	:
Source	Character Match	:
u-messageid	Integer	
Filler	UpTo	user=
Filler	Character Match	user=
u-username	Word	
Filler	UpTo	status=
Filler	Character Match	status=
<b>u-status</b>	<b>Word</b>	

Message preview of your rules

```
Jul 24 12:27:38 192.168.0.1 %: 12345user=user=aWordstatus=status=aWord
```

### The last rule - Additional Information

One item of interest is left. The last part of the message contains additional information. It starts after 'msg='. So the combination of 'UpTo' and 'Character Match' is used to go to the right position. All characters after 'msg=' until the end of the message are interesting. For this purpose the 'Rest of Message' parse type is available. It stores all characters from the current position until the end of the message. This also means that this rule can only be used once in a template and is always the last rule.

Enable: Post-Process Event  
 Settings are saved.

Optional Value

Property	Type	Value
u-timestamp	UNIX/LINUX-like Timest...	
Filler	Character Match	
Source	IP V4 Address	
Filler	Character Match	%
syslogtag	UpTo	:
Source	Character Match	:
u-messageid	Integer	
Filler	UpTo	user=
Filler	Character Match	user=
u-username	Word	
Filler	UpTo	status=
Filler	Character Match	status=
u-status	Word	
Filler	UpTo	msg=
Filler	Character Match	msg=
<b>msg</b>	<b>Rest of Message</b>	

Message preview of your rules

```

Jul 24 12:29:03 192.168.0.1 %: 12345user=user=aWordstatus=status=aWordmsg=msg=$R$E
$M$A$I$N$D$E$R$$$M$S$G$$$$$$$$$$$$$$$$
    
```

Complete parse template.

**What happens if the parser fails?**

If a rule does not match processing stops at this point. This means all properties of rules which were processed successfully until the non-matching rule occurs are available.

Let's assume the fourth rule of the following sample does not match.

Property	Type	Value
u-timestamp	UNIX/LINUX-like Timest...	
Filler	Character Match	
Source	IP V4 Address	
Filler	Character Match	%
syslogtag	UpTo	:
Source	Character Match	:
u-messageid	Integer	

rule does not match

these rules are never reached

The first three rules were processed successfully. Therefore u-timestamp and Source are available. But syslogtag and u-messageid are always empty due to the parser never process this rules.

The Post-Process template which was created in this article is available for [download](#). If you have further question on Post-Process, please contact our [support](#).

[1] Using the "u-" prefix is recommended to differentiate between MonitorWare-defined properties and user defined one. It is not required, but often of great aid. A common trap is that future versions of MonitorWare may use property names that a user has also used. MonitorWare will never use any name starting with "u-", so the prefix also guards against such a scenario.

[2] Filler is a predefined property which acts as a bin for unwanted characters. Essentially, the data is simply discarded.

**Please Note:** There's also a StepByStep Guide available which describes how the PostProcessAction works, you can find it [here](#).

## 6 Getting Help

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

Do you need help with the WinSyslog Service or WinSyslog in general? Do you need an important question answered? No problem, there is lots of help available!

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 [www.winsyslog.com/en/FAQ](http://www.winsyslog.com/en/FAQ)

The FAQ area is continuously being updated.

### 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 to 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 not to 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 aggressive. 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 disturbed. 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.

### WinSyslog Web Site

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

### Support Forum

Share questions and answers with your peers! The forum is also monitored by Adiscon support staff. To access the forum, point your browser at <http://forum.adiscon.com/forum,4.html>.

### Email

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

**Please note:** we have increasingly often problems with too-aggressive 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 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](mailto:info@adiscon.com) for further details.**

## Fax

Please direct your faxes to

**+49-9349-928820**

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

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

## Software Maintenance

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

To learn more about UpgradeInsurance, please visit

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

## Non-Technical Questions

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

## Product Updates

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

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

## 7 WinSyslog Concepts

*Learn what WinSyslog is made for and made of.*

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

WinSyslog operates on a set of elements. These are

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

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

## 8 Purchasing WinSyslog

*If you would like to use WinSyslog's advanced features, you can purchase your own copy.*

### The License

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

### Which Edition is for Me?

Information on all available WinSyslog editions can be found on the web at the following URL. This includes a feature and price comparison.

<http://www.winsyslog.com/common/en/products/winsyslog5-editions.asp>

### Pricing & Ordering

Please visit <http://www.winsyslog.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](mailto:info@adiscon.com).

## 9 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 WinSyslog Service](#)
- [Support for Mass Rollouts](#)
- [Version History](#)
- [Formats \(XML and Database\)](#)
- Property Replacer

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

### 9.1 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	<a href="#">MonitorWare Agent</a>	<a href="#">WinSyslog</a>
Standard Property	Yes	Yes
MonitorWare Echo Reply		Yes
Windows Event Log Properties	Yes	
Syslog Message Properties	Yes	Yes
Disk Space Monitor	Yes	
File Monitor	Yes	
Windows Service Monitor	Yes	
Ping Probe	Yes	
Port Probe	Yes	
Database Monitor	Yes	
Serial Port Monitor	Yes	
MonitorWare Echo Request	Yes	
System Properties	Yes	Yes
Custom Properties	Yes	Yes

## 9.2 Event Properties

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

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

### 9.2.1 Accessing 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 access the properties: they can not only be accessed as one full property. They can also be accessed as substrings and even be reformatted. As such, the property replacer provides a specific 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 lengthy property. The options allow to specify some additional formatting.

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%`

#### 9.2.1.1 Property

This is the name of the property to be replaced. It can be any property that a given event possesses. 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.

#### 9.2.1.2 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 occurrence 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 position 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.

#### **9.2.1.3 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 search 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 position (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 positions 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 occurrence 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 position 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:/:ocured/$/%
```

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.

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

```
%msg:/error/$:/ocured/%
```

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

```
%msg:/error /:/ ocured/$/%
```

### Default

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

#### 9.2.1.4 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:

<b>lowercase</b>	All characters in the resulting property extract will be converted to lower case.
<b>uppercase</b>	All characters in the resulting property extract will be converted to upper case.
<b>uxTimeStamp</b>	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.
<b>escapecc</b>	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).
<b>spacecc</b>	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.
<b>convgeruml</b>	Converts German Umlaut characters to their official replacement sequence (e.g. "ö" --> "oe")
<b>localtime</b>	Now you can print the Time with localtime format by using %variable:::localtime%

\* = 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!

### 9.2.1.5 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).**

## **9.2.2 System Properties**

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

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

### 9.2.3 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.

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

In order to make the product really generally useful, some few properties have been

---

defined in a generic way and are guaranteed 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.

#### 9.2.4.1 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.

<b>msgPropertyDescribed</b>	A human-readable representation of the message text. While this is generally 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.
<b>source</b>	The source system the message originated from. This can be in various representations (e.g. IP address or DNS name) depending on configuration settings.
<b>syslogpriority</b>	The severity of a syslog message. For non-syslog messages, this should be a close approximation to what a syslog severity code means.
<b>syslogfacility</b>	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.
<b>syslogtag</b>	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.
<b>resource</b>	A user-assigned numerical value. Does not have any specific meaning. Primarily intended for quick filtering.
<b>CustomerID</b>	A user-assigned numerical value. Does not have any specific meaning. Primarily intended for quick filtering.
<b>SystemID</b>	A user-assigned numerical value. Does not have any specific meaning. Primarily intended for quick filtering.
<b>timereported</b>	<p>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.</p> <p>For Windows Event Log events, timereported contains the timestamp from the event log record.</p>
<b>timegenerated</b>	The time the event was recorded by the service. If messages are forwarded via SETP, this timestamp remains intact.
<b>importance</b>	Reserved for future use.
<b>iut</b>	<p>Indicates the type of the event. Possible values are:</p> <ul style="list-style-type: none"> <li>1- syslog message</li> <li>2- heartbeat</li> <li>3- Windows Event Log Entry</li> <li>4- SNMP trap message</li> <li>5- file monitor</li> <li>8- ping probe</li> <li>9- port probe</li> <li>10- Windows service monitor</li> <li>11- disk space monitor</li> <li>12- database monitor</li> <li>13- serial device monitor</li> </ul>
<b>iusers</b>	Version of the event record (info unit). This is a monitorware internal version identifier.

#### 9.2.4.2 Windows Event Log Properties

<b>id</b>	Windows Event ID
<b>severity</b>	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)
<b>severityid</b>	The severity encoded as a numerical entity (like in Windows API)
<b>sourceproc</b>	The process that wrote the event record (called "source" in Windows event viewer).
<b>category</b>	The category ID from the Windows event log record. This is a numerical value. The actual value is depending on the event source.
<b>user</b>	The user name that was recorded in the Windows event log. This is "N\A" if no user was recorded.
<b>NTEventLogType</b>	The name of the Windows event log this event is from (for example "System" or "Security").
<b>bdata</b>	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 <b>not</b> to be present. This is because the binary data is seldomly used and very performance-intensive.

#### 9.2.4.3 Syslog Message Properties

<b>rawmessage</b>	The message as it was received from the wire (unparsed).
-------------------	--

#### 9.2.4.4 Disk Space Monitor

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

#### 9.2.4.5 File Monitor

<b>genericfilename</b>	The configured generic name of the file being reported.
------------------------	---

### Special IIS LogFile Properties

The Logfile Fields in IIS Logfiles are customizable, 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:

<b>p-date</b>	The Date on which the Event occurs
<b>p-time</b>	The Time on which the Event occurs
<b>p-c-ip</b>	The IP Adress of the User which accessed
<b>p-cs-username</b>	The Username of the User which accessed
<b>p-s-ip</b>	The Server IP
<b>p-s-port</b>	The Server Port
<b>p-cs-method</b>	The Client-Server Method (POST,GET)
<b>p-cs-uri-stem</b>	The accessed File including its path

#### 9.2.4.6 Windows Service Monitor

<b>sourceproc</b>	The name of the service whoms status is being reported (from the Windows service registry).
-------------------	---

#### 9.2.4.7 Ping Probe

<b>echostatus</b>	Status returned for the echo request
<b>roundtriptime</b>	Round trip time for the ping packet (if successful)

#### 9.2.4.8 Port Probe

<b>responsestatus</b>	The status of the probe.
<b>responsemsg</b>	The response message received (if any)

#### 9.2.4.9 Database Monitor

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

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

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

```
select name, street, zip, city from addresses
```

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

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

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

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

#### 9.2.4.10 Serial Monitor

<b>portname</b>	The name of the port that the data originated from (typical examples are COM1, COM2). The actual name is taken from the configuration settings (case is also taken from there).
-----------------	---

#### 9.2.4.11 MonitorWare Echo Request

<b>responsestatus</b>	<p>The status of the echo request. Possible values:  0 - request failed (probed system not alive)  1 - request succeeded</p> <p>If the request failed, additional information can be found in the <i>msg</i> <a href="#">standard property</a>.</p>
-----------------------	---

**9.2.4.12 FTP Probe**

<b>ftpstatus</b>	The status of the connection.
<b>ftprespmsg</b>	The response of the connection.

**9.2.4.13 IMAP Probe**

<b>imapstatus</b>	The status of the connection.
<b>imaprespmsg</b>	The response of the connection.

**9.2.4.14 NNTP Probe**

<b>nntpstatus</b>	The status of the connection.
<b>nntprespmsg</b>	The response of the connection.

**9.2.4.15 SMTP Probe**

<b>smtpstatus</b>	The status of the connection.
<b>smtprespmsg</b>	The response of the connection.

**9.2.4.16 POP3 Probe**

<b>pop3status</b>	The status of the connection.
<b>pop3respmsg</b>	The response of the connection.

**9.2.4.17 HTTP Probe**

<b>httpstatus</b>	The status of the connection.
<b>httprespmsg</b>	The response of the connection.

## 9.3 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 Microsoft Network Monitor. So thankfully, many administrators are already used to this type of Interface. If you are not familiar with it, however, it looks a bit confusing at first. In this chapter, we are providing some samples of how boolean expressions can be brought into the tree.

### Example 1

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

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

```
If (msg = "DUPADDRESS") Or (msg = "SPANTREE") Or (msg = "DUPLICATE_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, not 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:

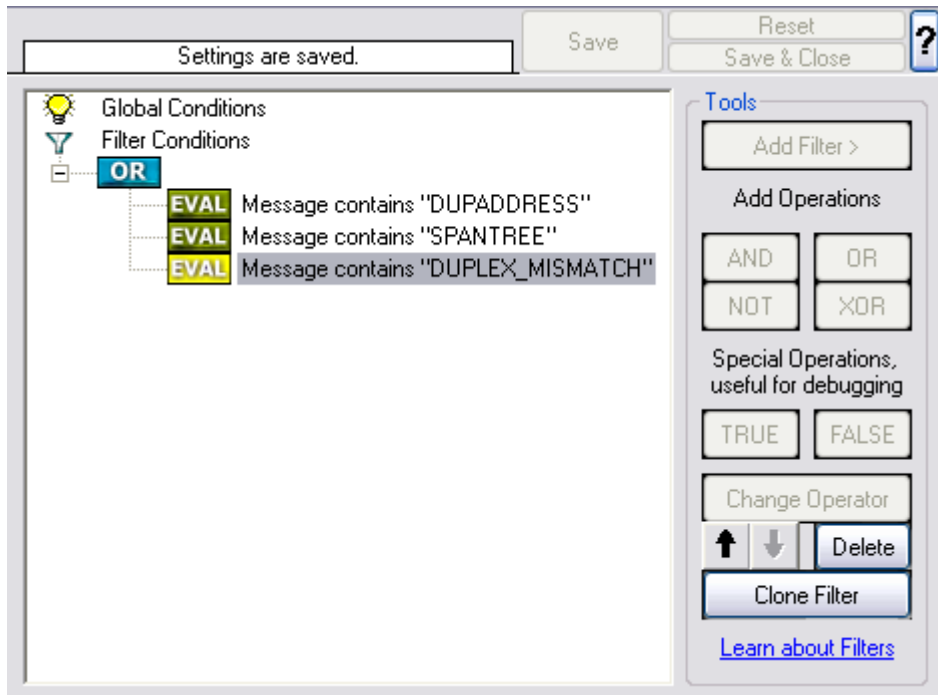


Figure 1 - Example 1

## Example 2

Example 2 is very similar to example 1. Again, the message content is to be checked for three strings. 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 = "DUPADDRESS") And (msg = "SPANTREE") And (msg = "DUPLICATE_MISMATCH) then
    execute action(s)
end if
```

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

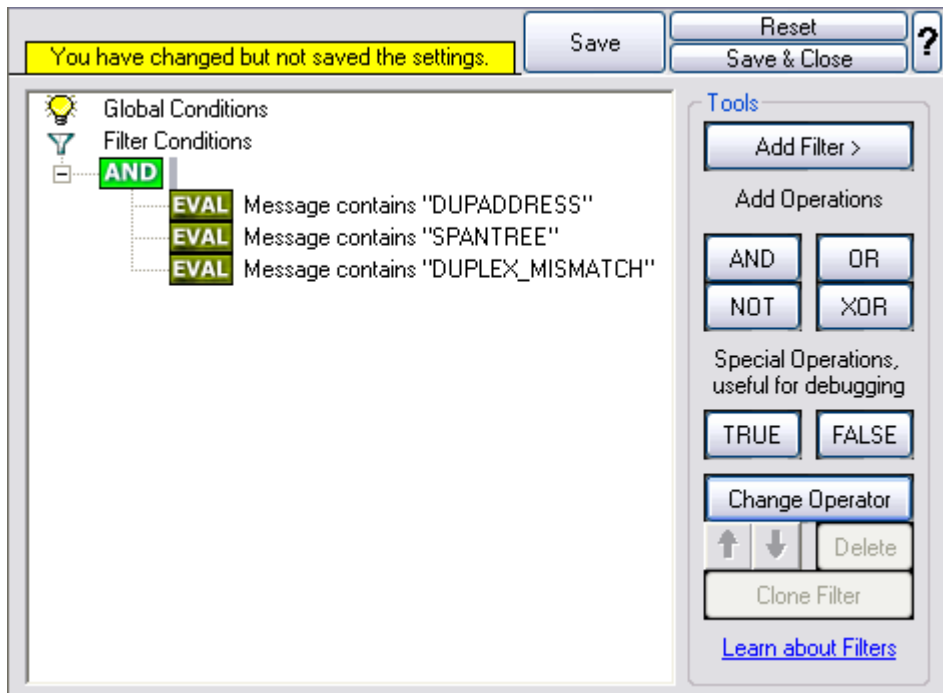


Figure 2 - Example 2

### 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 specific subnet.

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

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

```
If ((sourceSys = "192.0.2")
    And
    ((msg = "DUPADDRESS") Or (msg = "SPANTREE") Or (msg = "DUPLEX_MISMATCH"))
    ) then
    execute action(s)
end if
```

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

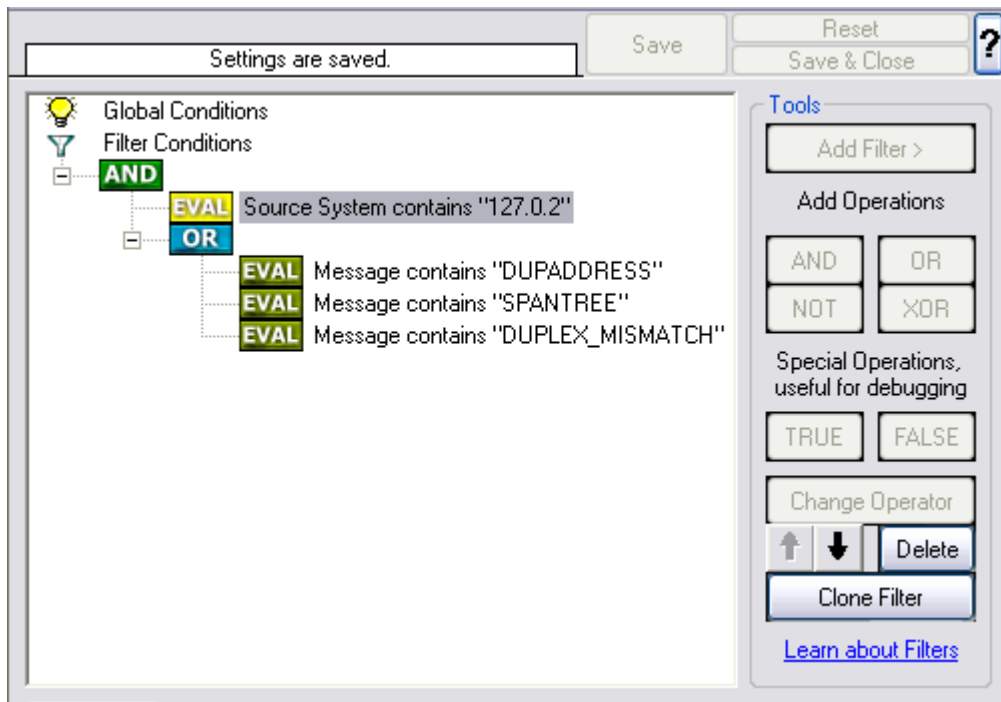


Figure 3 - Example 3

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](#)

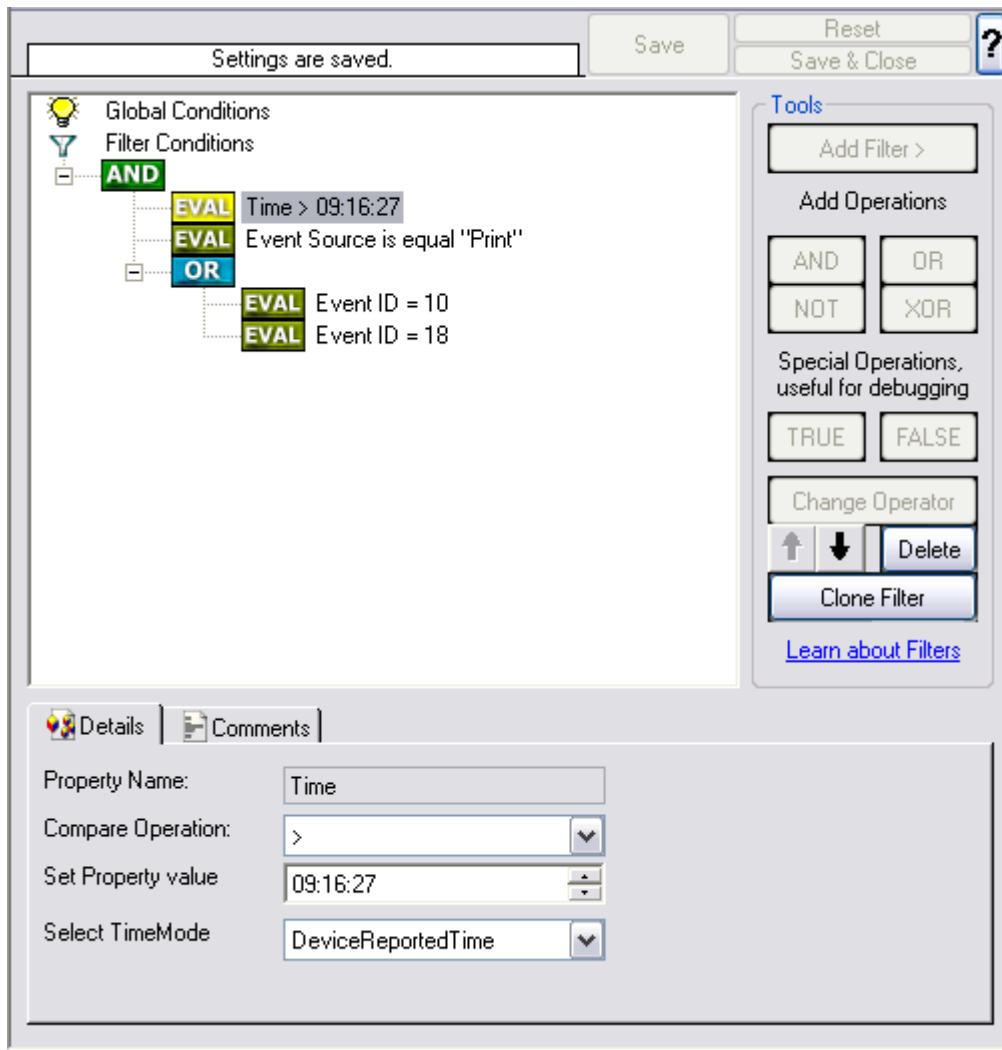
### Example 4

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:



## 9.4 WinSyslog Shortcut Keys

Use shortcut keys as an alternative to the mouse when working in WinSyslog Client. Keyboard shortcuts may also make it easier for you to interact with WinSyslog. All these shortcuts 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.**

## 9.5 Version Comparison

WinSyslog comes in different versions. Some of them are feature-richer than 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 which Version provides which services, actions and other features.

## 10 Copyrights

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

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

**Openssl-0.9.8a:** <http://www.adiscon.org/3rdparty/openssl-0.9.8a.tar.gz>

**Net-SNMP-5.2.1:** <http://www.adiscon.org/3rdparty/net-snmp-5.2.1.tar.gz>

**Liblogging:** <http://www.adiscon.org/3rdparty/liblogging.zip>

**VB6**

**NeoCaption:** [http://www.adiscon.org/3rdparty/VB6\\_NeoCaption\\_Full\\_Source.zip](http://www.adiscon.org/3rdparty/VB6_NeoCaption_Full_Source.zip)

**Note:** 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.

## 11 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.

### 11.1 EventReporter

[EventReporter](#) is [Adiscon's](#) solution to forward Windows NT/2000/XP 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.

## 11.2 Millisecond

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

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

[Click here](#) for more Information about Milliseconds.

## 11.3 Monitor Ware Line of Products

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

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

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

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

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

[Click here](#) for more Information about the MonitorWare Line of Products.

## 11.4 Resource ID

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

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

In [MonitorWare Agent](#) 1.0 and [WinSyslog](#) 4.0 support for Resource IDs is limited. The field is present and can be persisted to the database or stored in XML files, but besides this there is no value in it.

Later releases of the MonitorWare Line of Products will much broader support the

Resource ID.

[Click here](#) for more Information about the Resource ID:

## 11.5 SETP

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

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

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

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

[Click here](#) for more Information about SETP.

## 11.6 SMTP

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

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

SMTP is highly standardized. As such, a standard email client can work with all SMTP compliant servers. In the public Internet, almost all providers offer SMTP compliant

mail servers for their customer's use.

[Click here](#) for more Information about SMTP.

## 11.7 Syslog Facility

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

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

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

[Click here](#) for more Information about Syslog Facility.

## 11.8 TCP

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

## 11.9 UDP

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

[Click here](#) for more Information about UDP.

## 11.10 Upgrade Insurance

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

[Click here](#) for more Information about Upgrade Insurance.

## 11.11 UTC

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

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

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

[Click here](#) for More Information about UTC.