WinSyslog

Table of Contents

1 Introduction

1.1 About WinSyslog

WinSyslog 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 router 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

Most customers use WinSyslog to gather events reported from syslog enabled devices (routers, switches, firewalls and printers to name a view) 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 improvide servcies and rule introduced in version 4 allows 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 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. email message or store 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.

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 a 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 received 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. So later auditing and review of important system events is possible without 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 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 will be lost.

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.

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).

Runs on large Variety of NT Systems

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

Multi-Language Client

The WinSyslog client comes with multiple languages ready to go. Out of the box, English, French , and German 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!

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.

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.

WinSyslog Web Access

WinSyslog Web Access allows to access the WinSyslog database over the web. Syslog data can be filtered and viewed in any browser. Web access is an optional component that can be installed at any time. It can also be used to view real-time data if the Interactive Syslog Server should not be used.

1.3.3 How these components work together

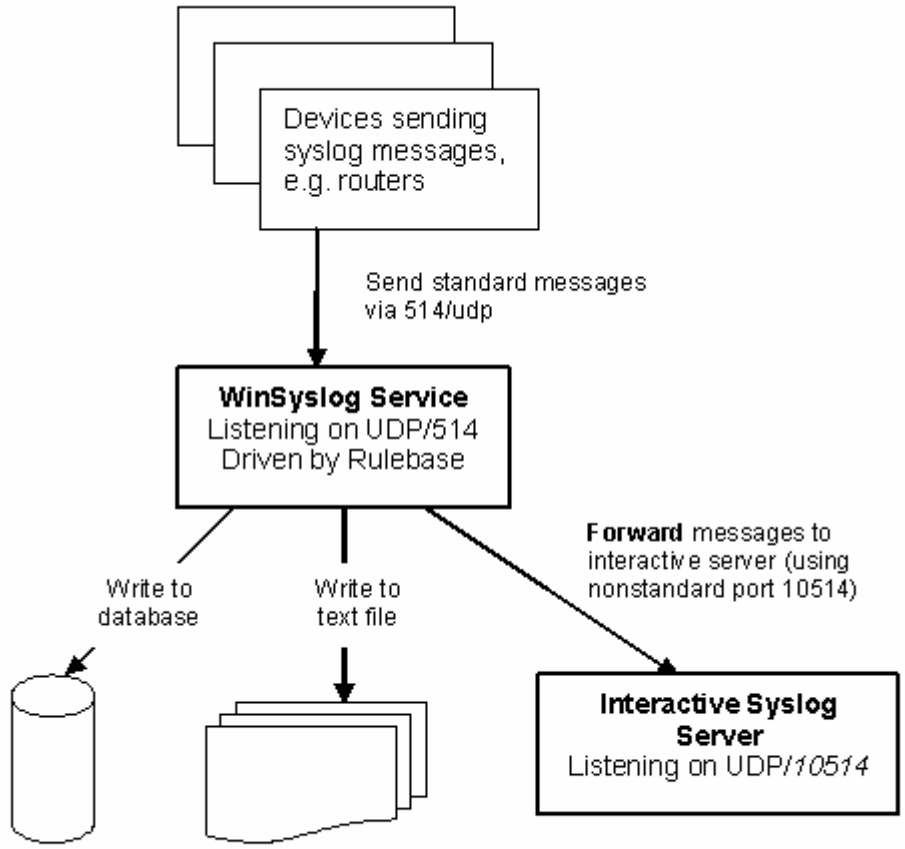
How these components work together

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 it's 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).

So 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.

WinSyslog Web Access is only needed if accessing syslog messages over the web is desired. Thus it is an optional component and is not installed by default. For Web Access to operate correctly, the service must be configured to store incoming messages into a database. This is not done by default and needs to be configured in the service configuration client. Web Access is fully optional, so there is no need to install it. No other component is depending on the presence of Web Access.

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.

The **client and Interactive Syslog Server** can be installed on Windows NT 4.0 SP6 and above. This includes Windows 2000, 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.

The **service** has fewer requirements. Most importantly, it does not need Internet Explorer to be installed on the system. It works under the same operating system versions. Additionally, it should perform well under NT 3.51, but as we have not yet received any request for supporting this operating system version, no tests have been conducted yet. This will be done upon request. The service also by design supports the Compaq/Digital APHA processor, but again has not been ported yet due to missing demand. If you are in need of such a version, please contact Adiscon at support@adiscon.com.

At runtime, the base service requires 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.

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 . WinSyslog is capable of storing such bursts temporarily in memory even if the machine would otherwise be too slow to process the messages.

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

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.

WinSyslog is part of Adiscon's MonitorWare line of products. We highly recommend visiting

http://www.monitorware.com/Common/en/SeminarsOnline/

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 simply and easy. A standard setup program installs the application.

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.

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.

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.

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 critaria, 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 that syslog listener

- **Start the WinSyslog service**

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

2.3 Installing WinSyslog Web Access

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

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

In future releases, WinSyslog Web Access will be enhanced to support web based configuration. Visit www.winsyslog.com to learn if a new version is already available.

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 not include all information that might be relevant to the situation.

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.

- Creating a simple syslog server
- How to configure a syslog server
- Forwarding NT Event Logs to a syslog server

There may be additional step-by-step guides available on our web site. Please visit

http://www.winsyslog.com/Common/en/stepbystep/

to see the most current list.

4 Using Interactive Syslog Server

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

In this chapter, you will learn how to work and configure the Interactive Syslog Server.

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. WinSyslog is configured by 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 will directly be able using 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 WinSyslog program folder located in the Start menu.

It can also be launched from the command prompt:

- Open a Command Prompt window

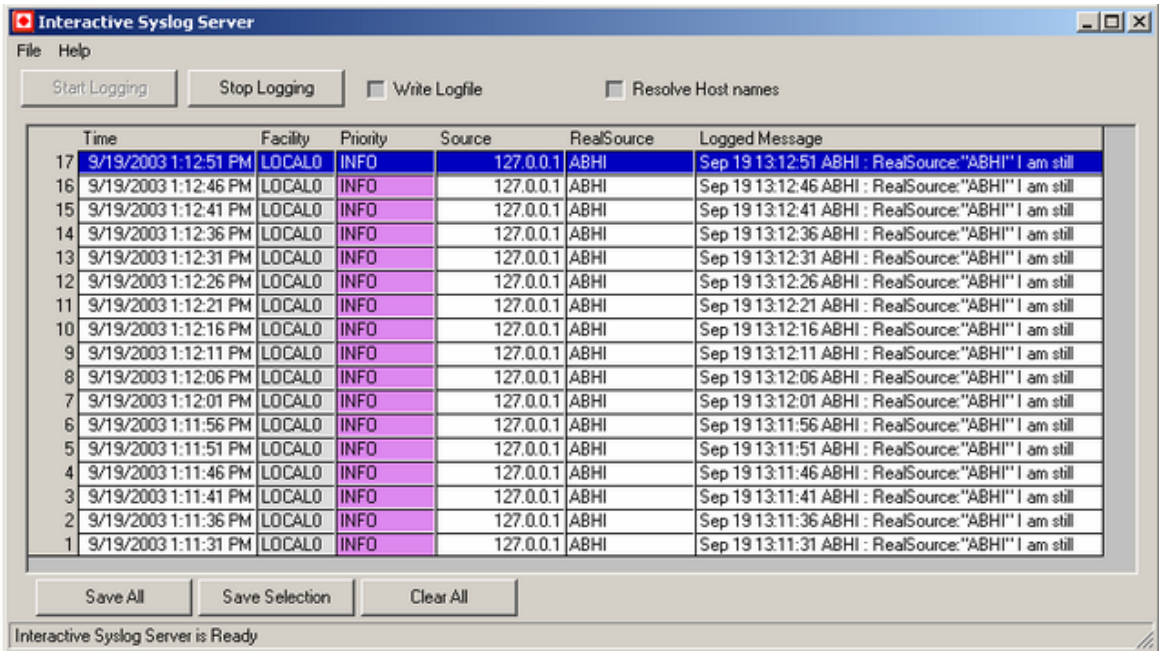
- Change to the drive and directory where the WinSyslog software is installed (default: "\Program Files\WinSyslog")

- Type " InteractiveSyslogServer.exe " and hit enter.

4.2 The Interactive Logging

Interactive Logging enables the client to log syslog messages itself. So it can work without the service. However, by default the service is required to run. This is done to prevent conflicts between the interactive server and the background service. If you do not have a good reason to do so, we strongly recommend using this default setup.

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



Start / Stop Logging Buttons

These buttons start and stop Interactive logging. Once started, the client will log all incoming messages until logging is stopped by the user. Messages are written to a circular buffer. That means if the maximum buffer size is reached, new messages will be stored, but older messages will be removed from the buffer. This allows the client to run for extended periods of time without taking up to 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

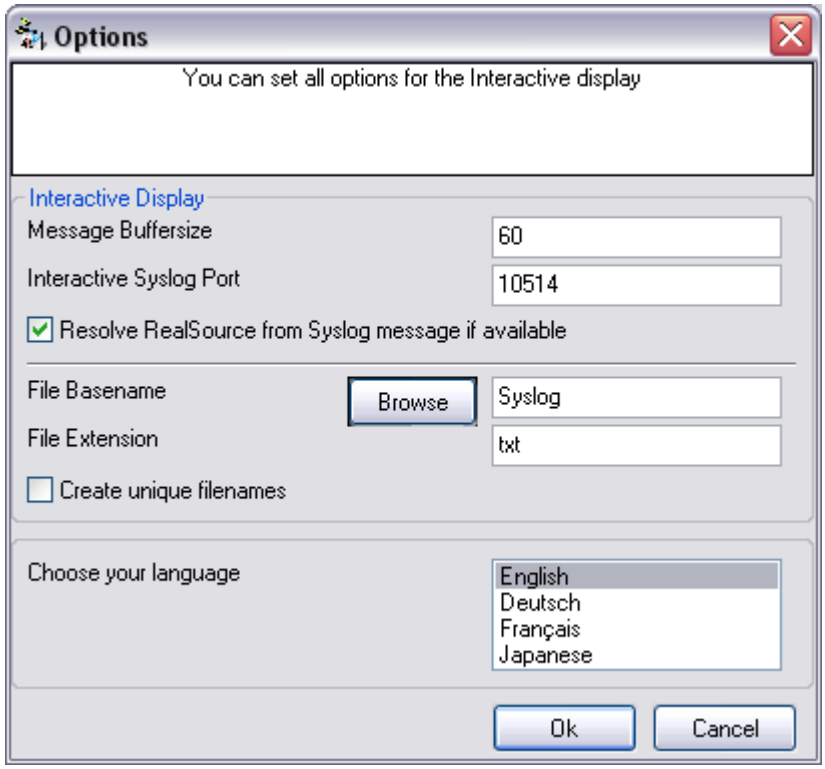
Also saves a comma-delimited file. However, only messages selected (highlighted) will be written to the file.

Clear All

Erases all messages from real-time display.

4.3 Interactive Syslog Server Options

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



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. 0 is default from system services database. Most installations can leave it at 10514.

File Basename

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

File Extension

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

Create unique filenames

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

Language

The Interactive Syslog Server is multilingual by desing. 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 made available. Please check www.winsyslog.com from time to time. If you are interested in other languages and volunteer to provide translation services, please email info@adiscon.com. We will gladly help.

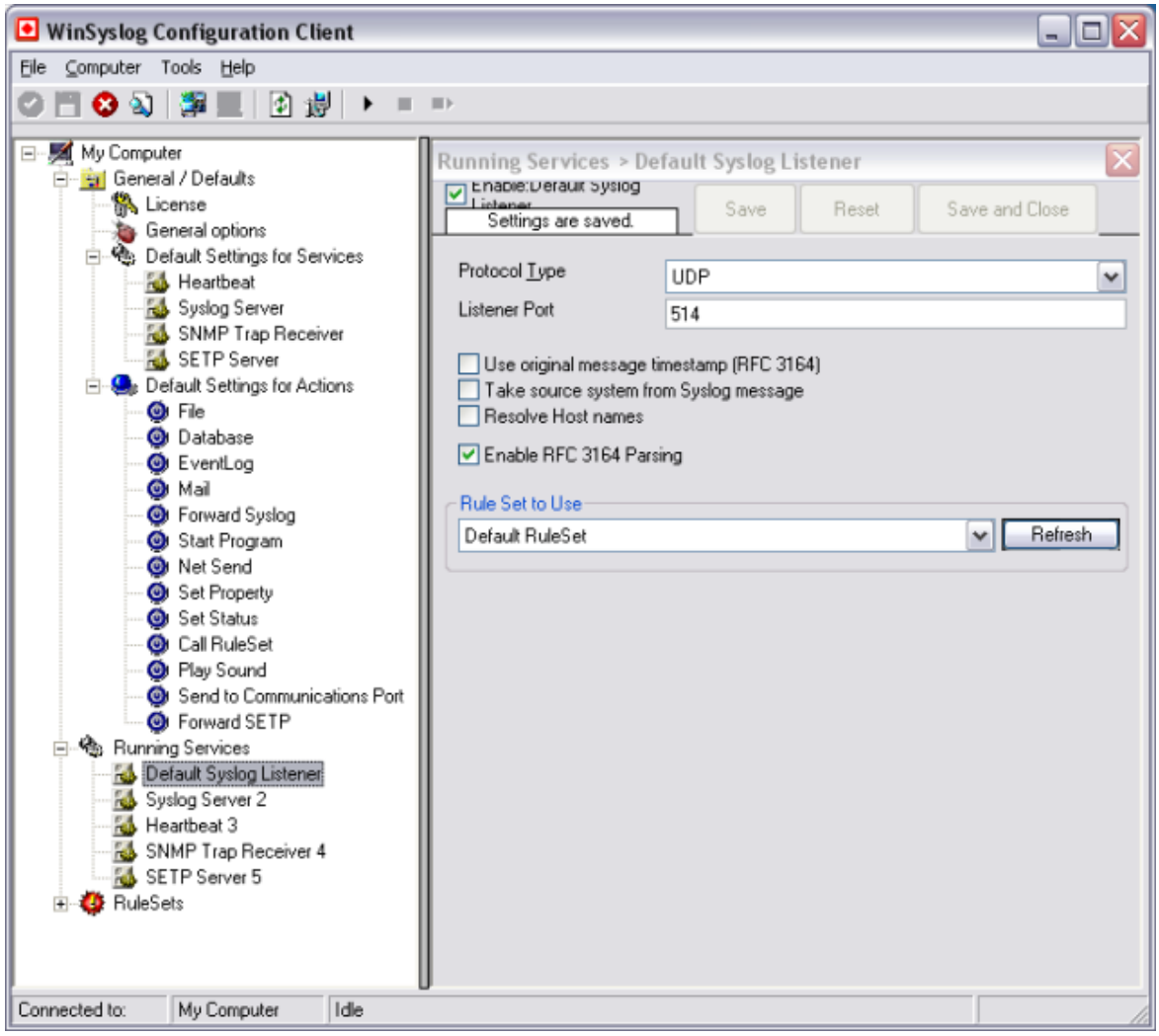
5 Configuring WinSyslog

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

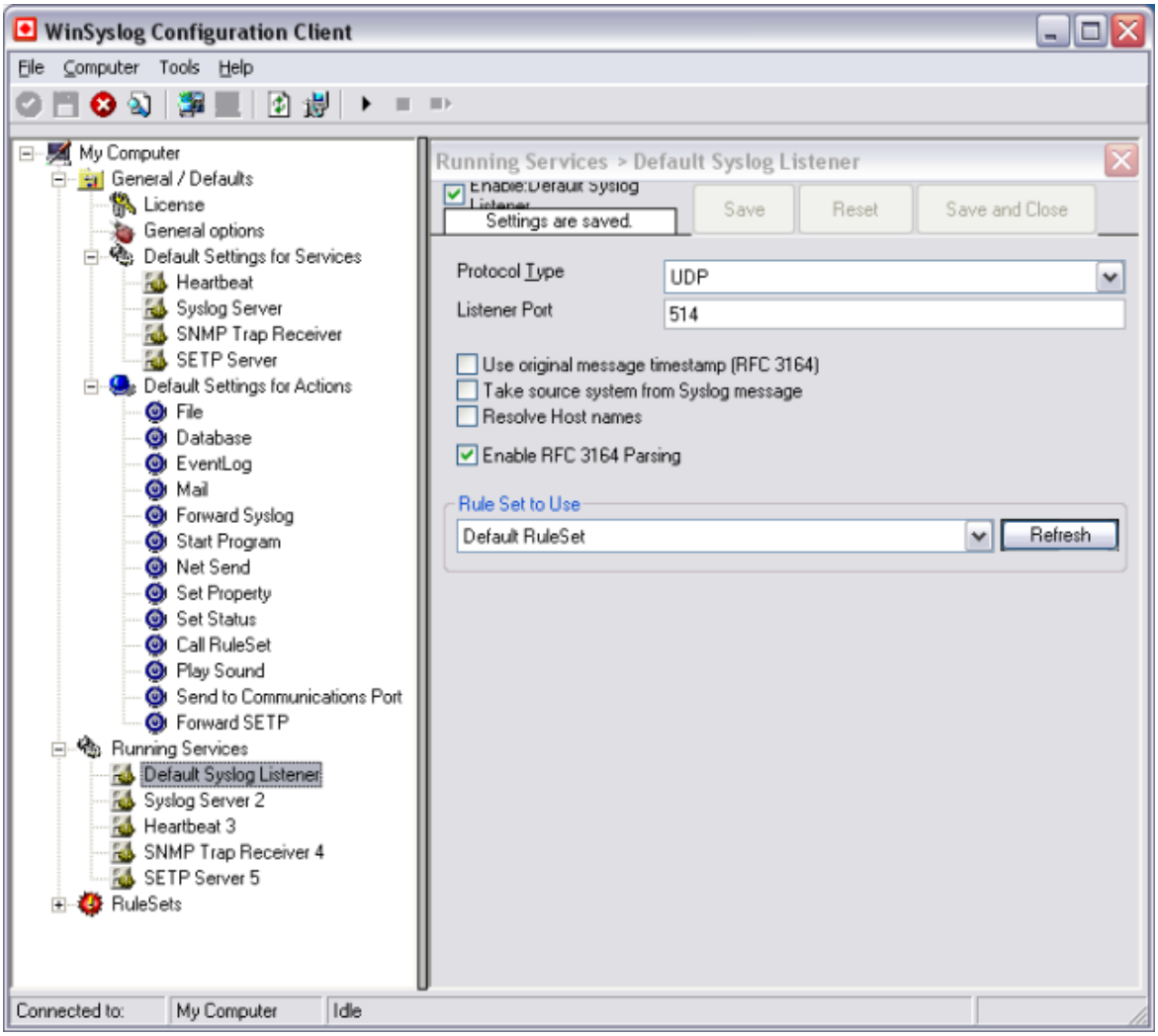
The configuration client ("the client") has two elements. On the left hand side is a tree view that allows you to select the various elements of the WinSyslog system. On the right hand side are parameters specific to the element selected in the tree view. In the sample above, the right hand side displays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

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

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

The tree view's **Services** area lists all configured services as well as their parameters. There is exactly one service entry for each service created. Please note that there can be as many instances of a specific service type as your application requires. In the above example, there are two instances of the syslog listener, each one listening to a separate port. Theoretically, you can run a few hundred services in a single service instance. However, both from a usage scenario point of view as well as in regard to 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. WinSyslog does not restrict this number. If there is a need for a large number of services and the hardware is capable of managing all this tasks, there is nothing in WinSyslog 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 be 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 "Services". Then select "Add Service" and the respective service type from the pop up menu. Then follow the wizard. To delete an existing service, right click it and select "Delete Service". This will remove the service and its configuration irrecoverable. To temporarily "remove" a service, simply disable it in the property sheet.

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

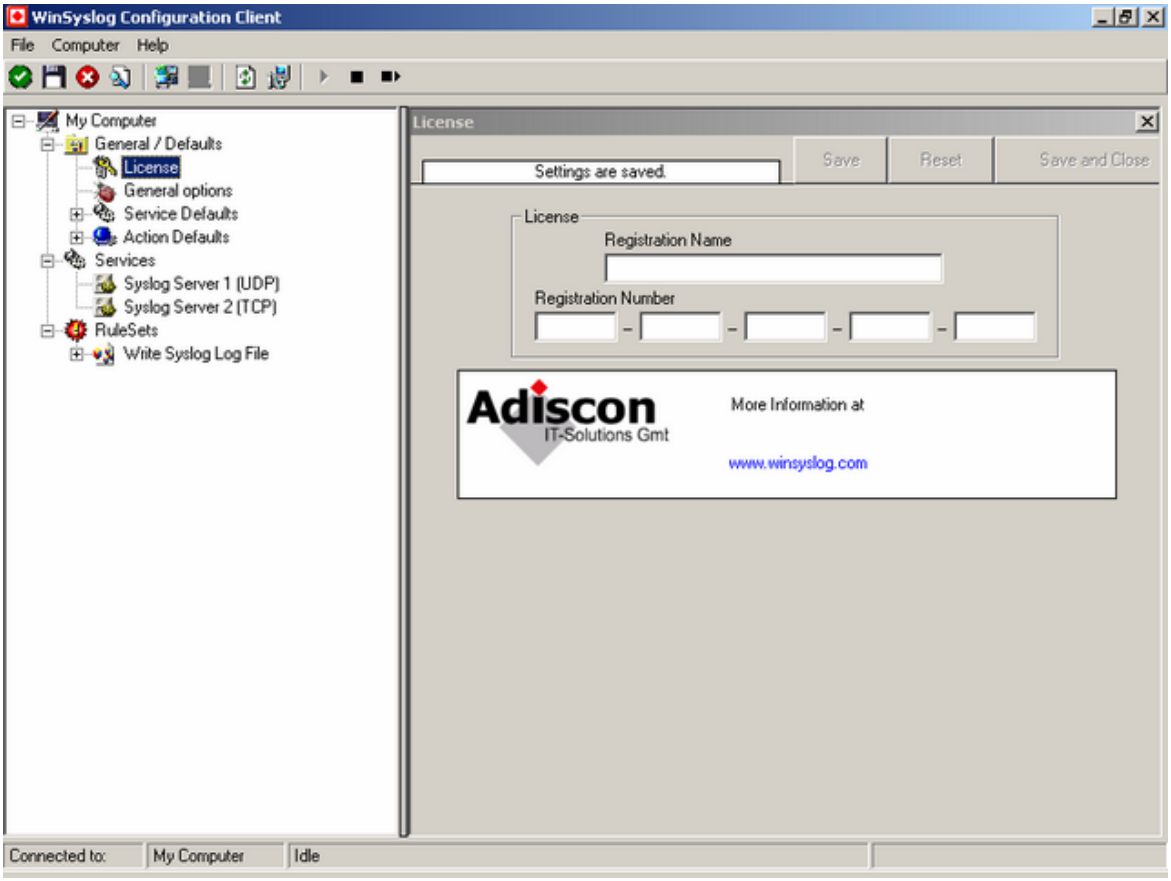
Beneath each rule set are the individual rules. As described in "Rules", a rule's position in the list is vitally important. Rules at the top of the rule set are executed before those further down. To move a rule up or down, simply right click it and select "move up" or "move down" from the pop up menu.

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

The following sections describe each element's properties.

5.1 License Options

This tab can be used to enter the WinSyslog license after purchase. It activates the professional version's advanced features.



License Option Parameters

Registration Name

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

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

Registration Number

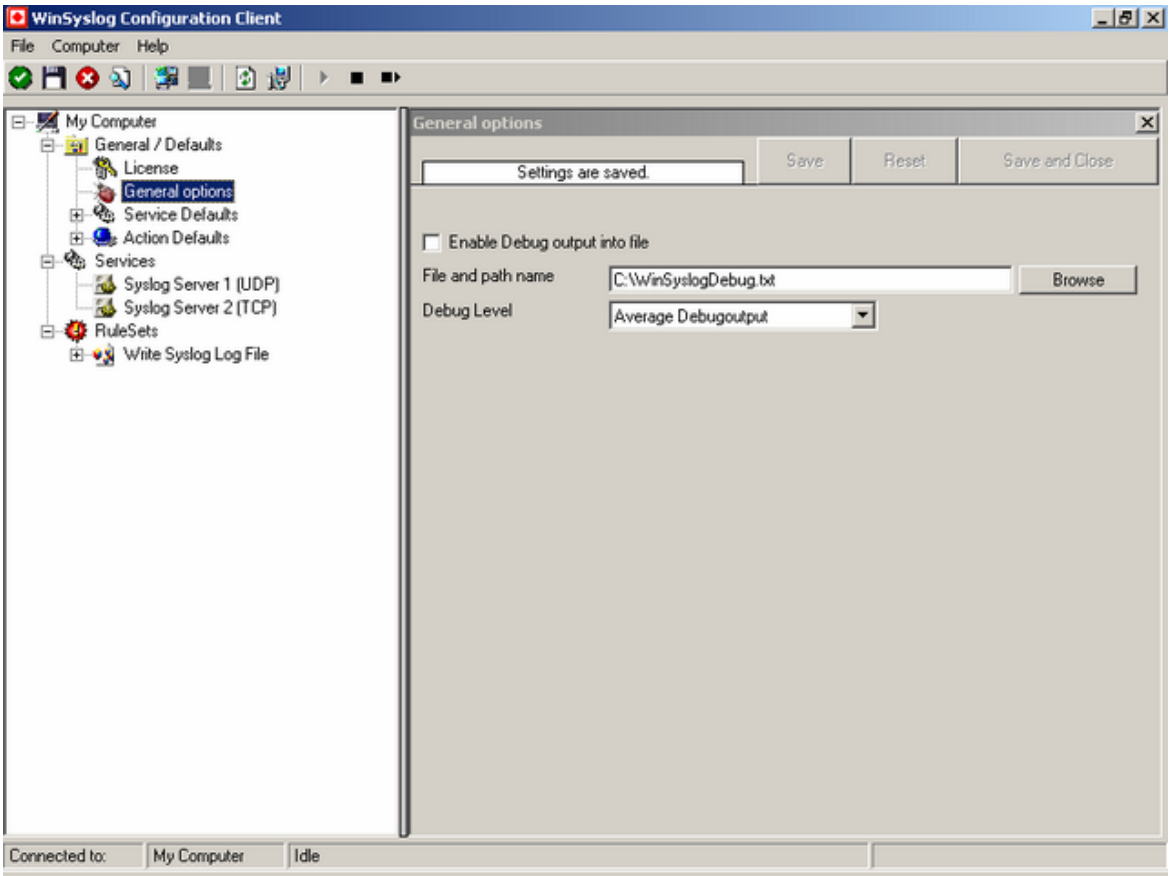
Adiscon provides this number. It is valid for a specific registration name. Be sure to enter the correct registration number. The client will detect invalid registration numbers and report and corresponding error.

5.2 General Options

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

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

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



Debug OptionsParameters

Enable Debug output into file

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

File and path name

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

Debug Level

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

5.3 Services

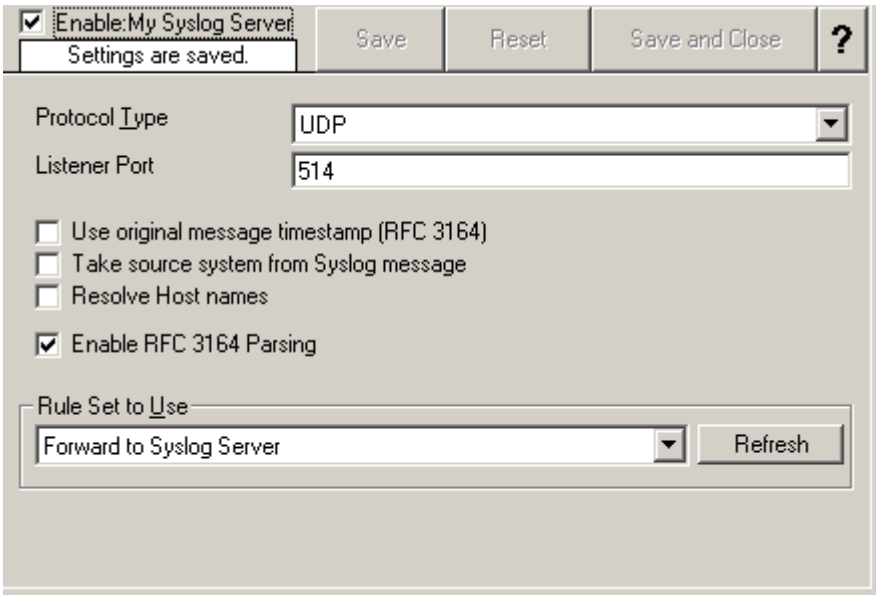
5.3.1 Understanding Services

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

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

5.3.2 Syslog Server

Configures a Syslog server service.



Protocol Type

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

Listener Port

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

Use Original Message Timestamp

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

Take source system from Syslog message

If this box is checked, the name or IP address of the source system is retrieved from the Syslog message itself (according to RFC 3164). If left unchecked, it is generated based on the address, the message was received from.
Please note that there are many devices, which do NOT generate RFC 3164 compliant messages. If you check this option here, you might see a very strange value as the event source!

Resolve Hostnames

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

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

Enable RFC 3164 Parsing

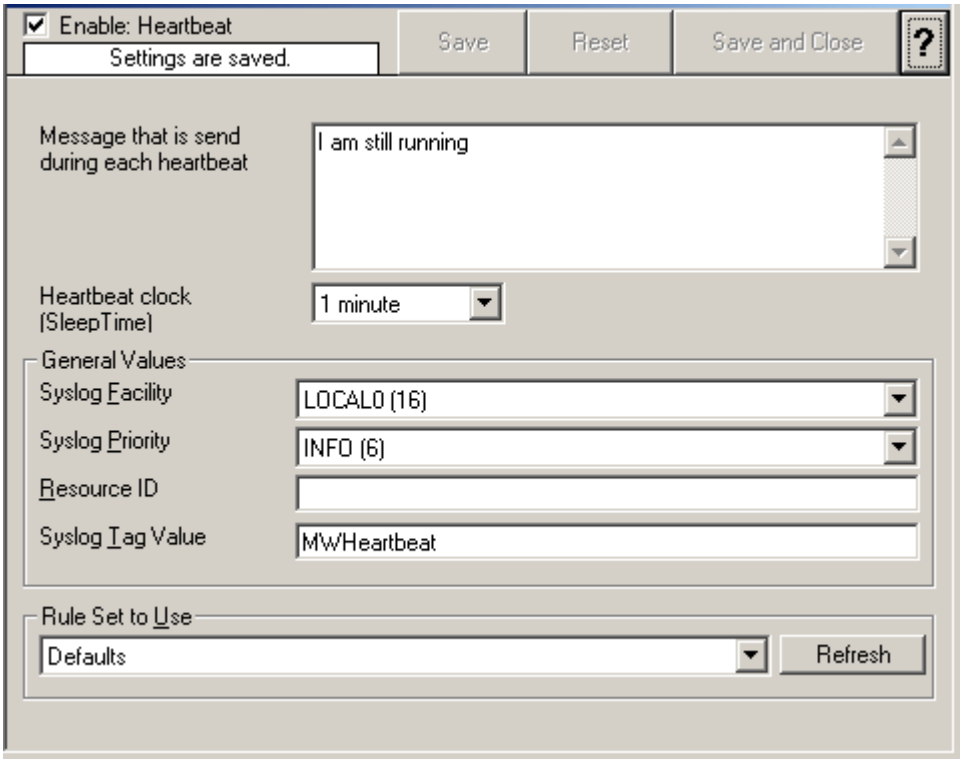
If this box is checked, RFC 3164 compliant message parsing is enabled. If unchecked, "traditional" Adiscon message parsing is selected. If you experience trouble with the sender host name or the timestamp, we suggest that you turn of RFC 3164 compliant message parsing. Many existing devices do not fully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

Default Ruleset Name

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

5.3.3 Heartbeat

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



Message to Send

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

Sleep Time

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

Syslog Facility

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

Syslog Priority

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

Syslog Tag Value

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

Resource ID

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

Default Ruleset Name

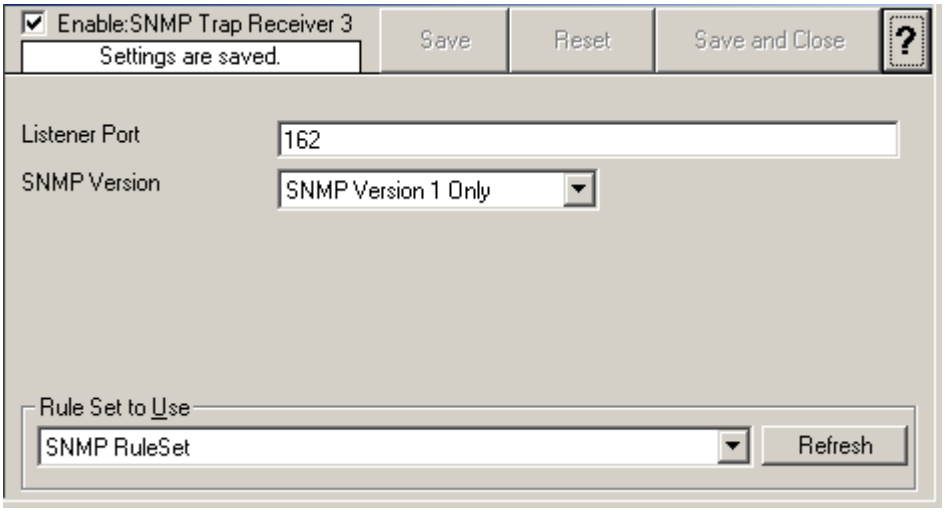
Name of the rule set to be used for this service. The rule set name must be valid.

5.3.4 SNMP Trap Receiver Service

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

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

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



Listener Port

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

SNMP Version

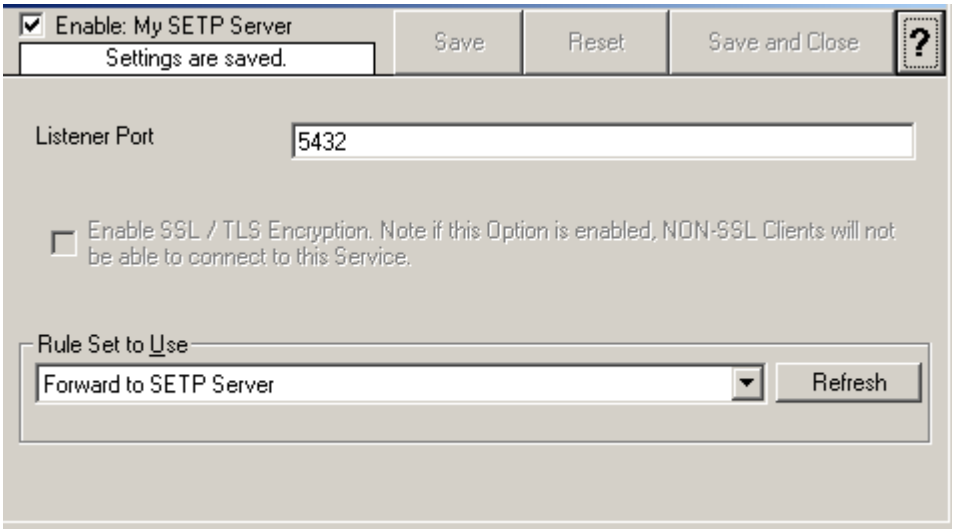
Can be used to restrict the SNMP versions.

Rule Set to Use

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

5.3.5 SETP Server

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



SETP Server Properties

Listener Port

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

Enable SSL/TLS

If this option is enabled then this action will be able to connect to SSL/TLS SETP servers. Please make sure that you want this option to be enabled.
Please note: If this option is enabled, this action will not be able to connect to NON-SSL SETP Servers.

Default Ruleset Name

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

5.4 Filter Conditions

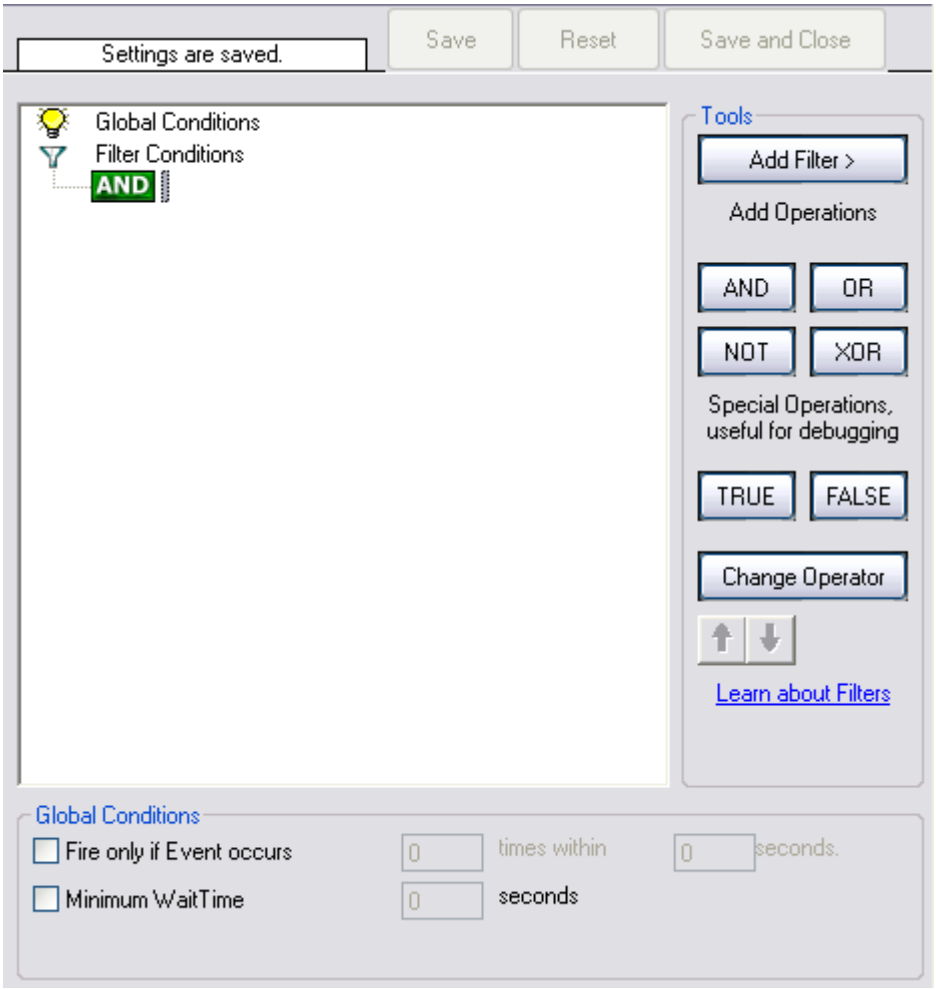
5.4.1 Filter Conditions

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 will be carried out.

Filter conditions can be as complex as needed. Full support for Boolean operations

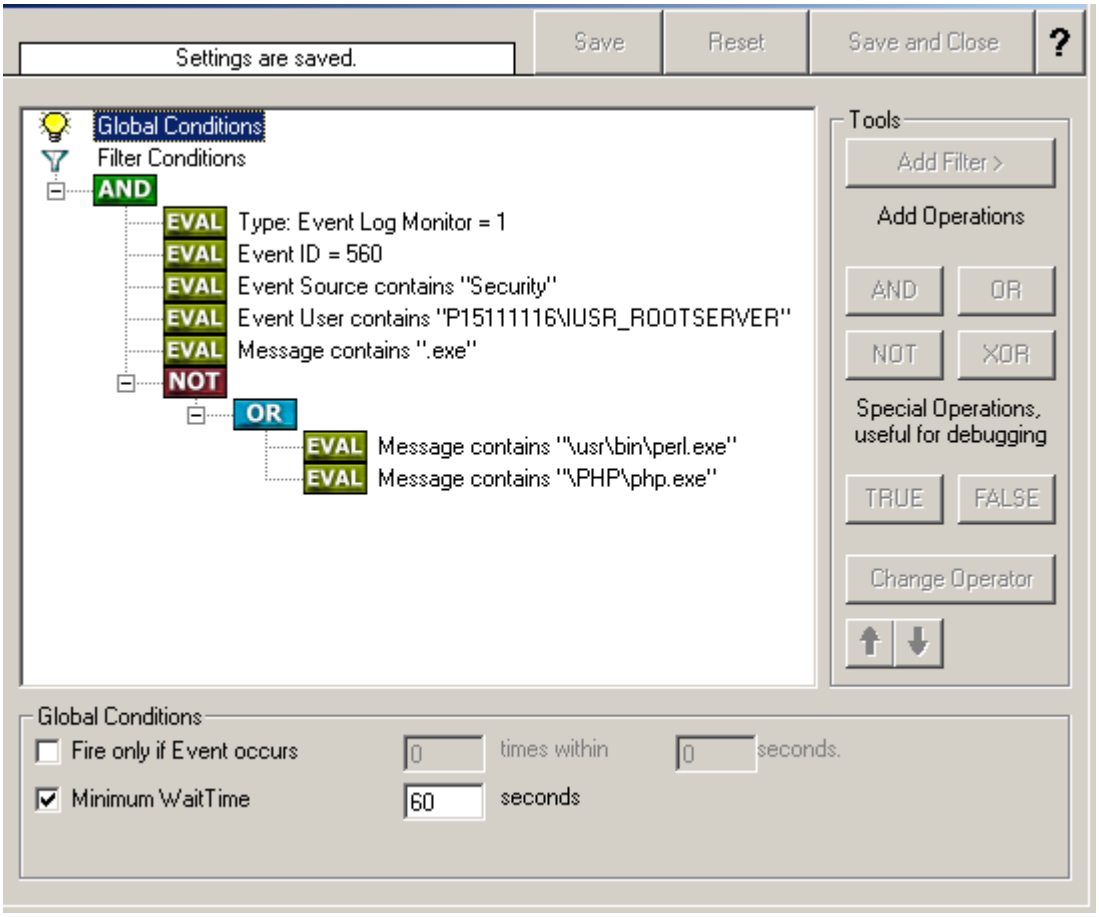
and nesting of conditions is supported.

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



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

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



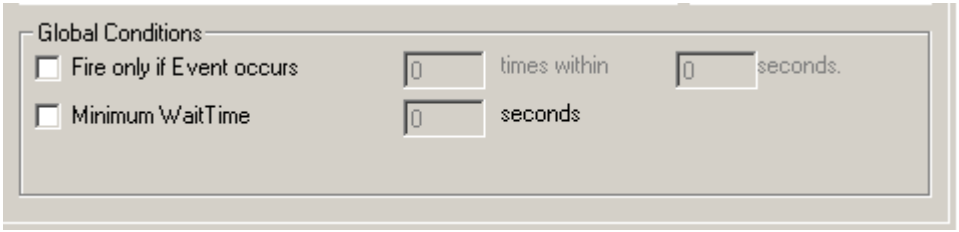
This filter condition is part of an intrusion detection rule set. Here Windows file system auditing is used to detect a potentially successful intrusion via Internet information server. This is done by enabling auditing on all executable files. Internet Information Server will access them under the IUSR_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

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

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

5.4.2 Global Conditions

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



Fire only if Event occurs

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

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

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

Minimum Wait Time

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

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

5.4.3 Operators

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

AND

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

OR

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

NOT

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

XOR

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

TRUE

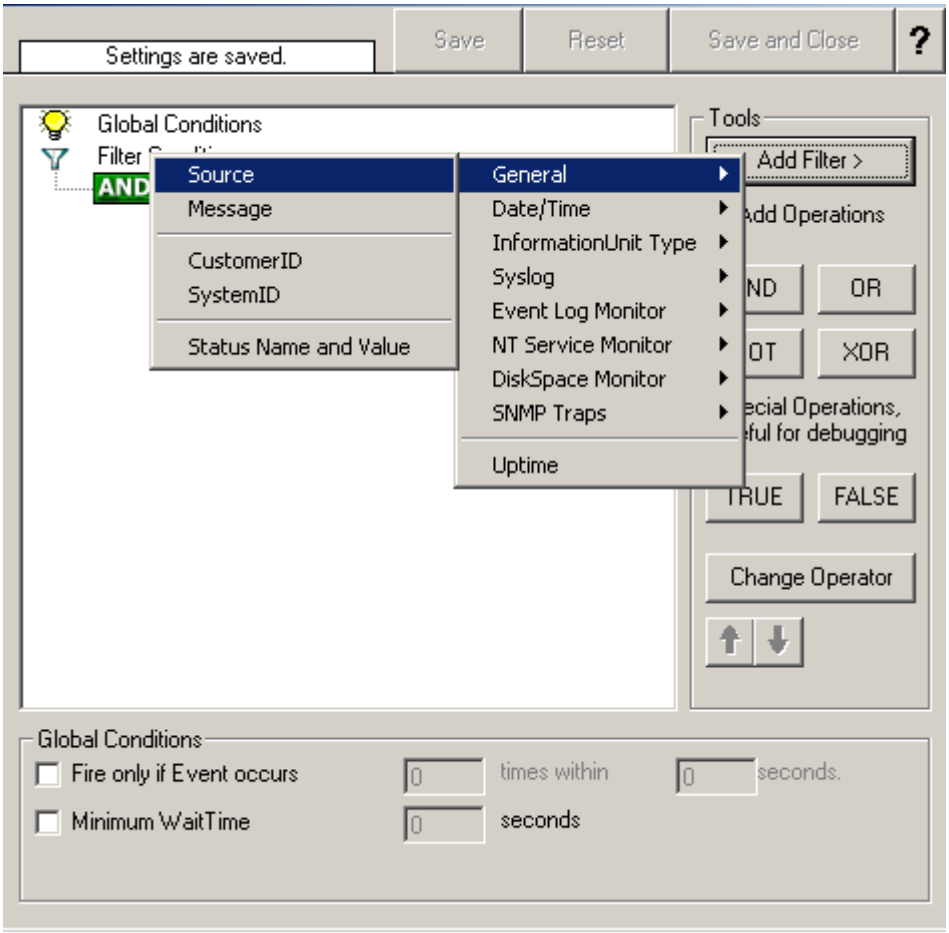
Useful for debugging, will just return TRUE.

FALSE

Useful for debugging as well, will return FALSE.

5.4.4 General

These are non-event log specific settings.



Source System

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

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

Message Content

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

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

This filter is of type string.

CustomerID

CustomerID (Type=Number).

SystemID

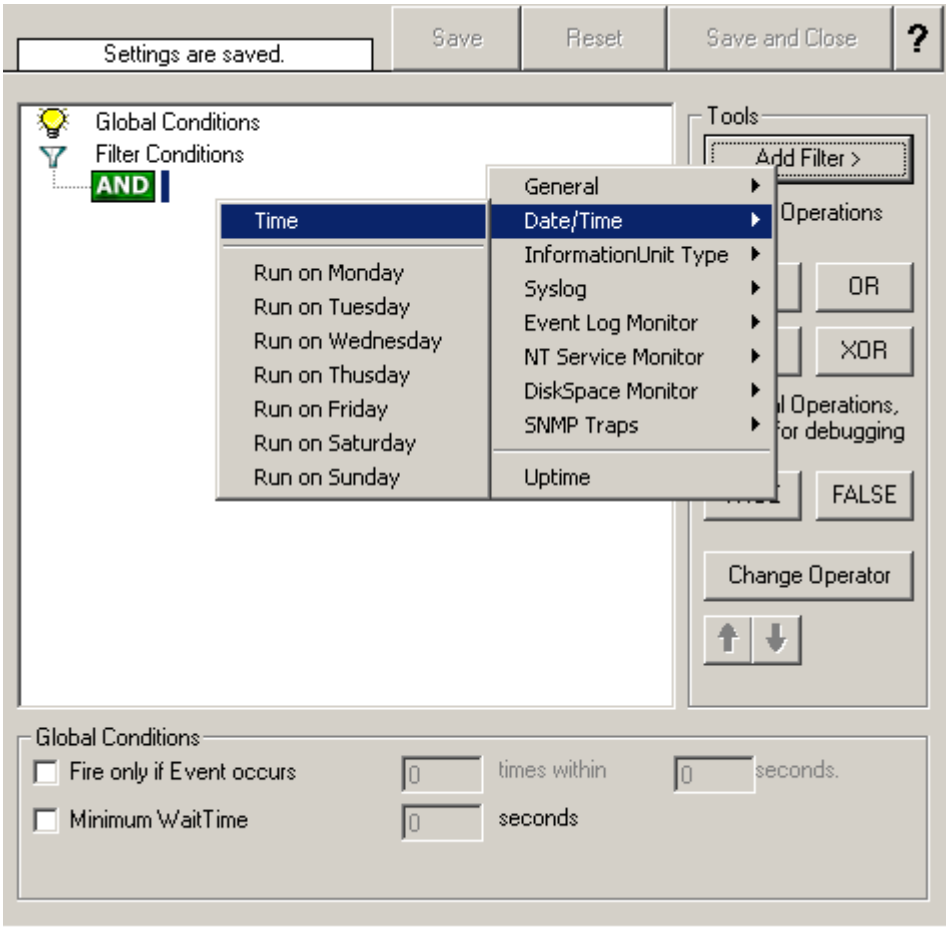
SystemID (Type=Number).

Status Name and Value

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

5.4.5 Date/Time

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



The following filters are available in detail:

Start time (Type=Time)

End Time (Type=Time)

Run on Monday (Type=Boolean)

Run on Tuesday (Type=Boolean)

Run on Wednesday (Type=Boolean)

Run on Thursday (Type=Boolean)

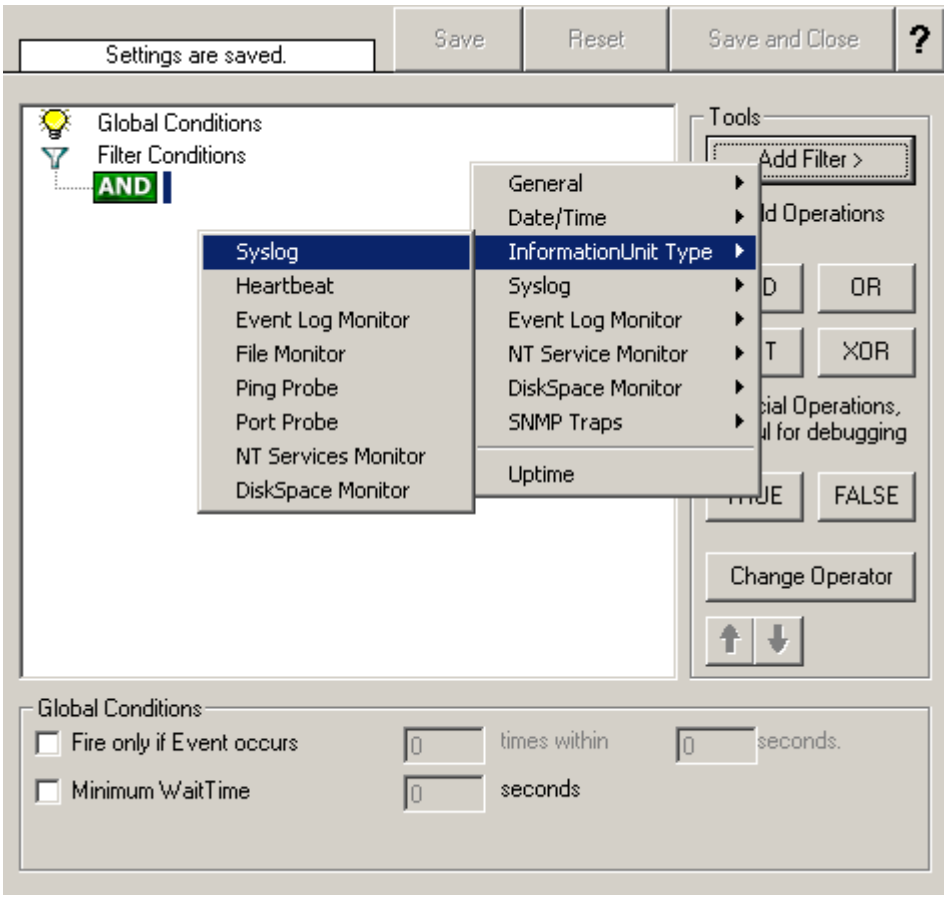
Run on Friday (Type=Boolean)

Run on Saturday (Type=Boolean)

Run on Sunday (Type=Boolean)

5.4.6 InformationUnit Type

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



Syslog (Type=Boolean)

Heartbeat (Type=Boolean)

Event Log Monitor (Type=Boolean)

File Monitor (Type=Boolean)

Ping Probe (Type=Boolean)

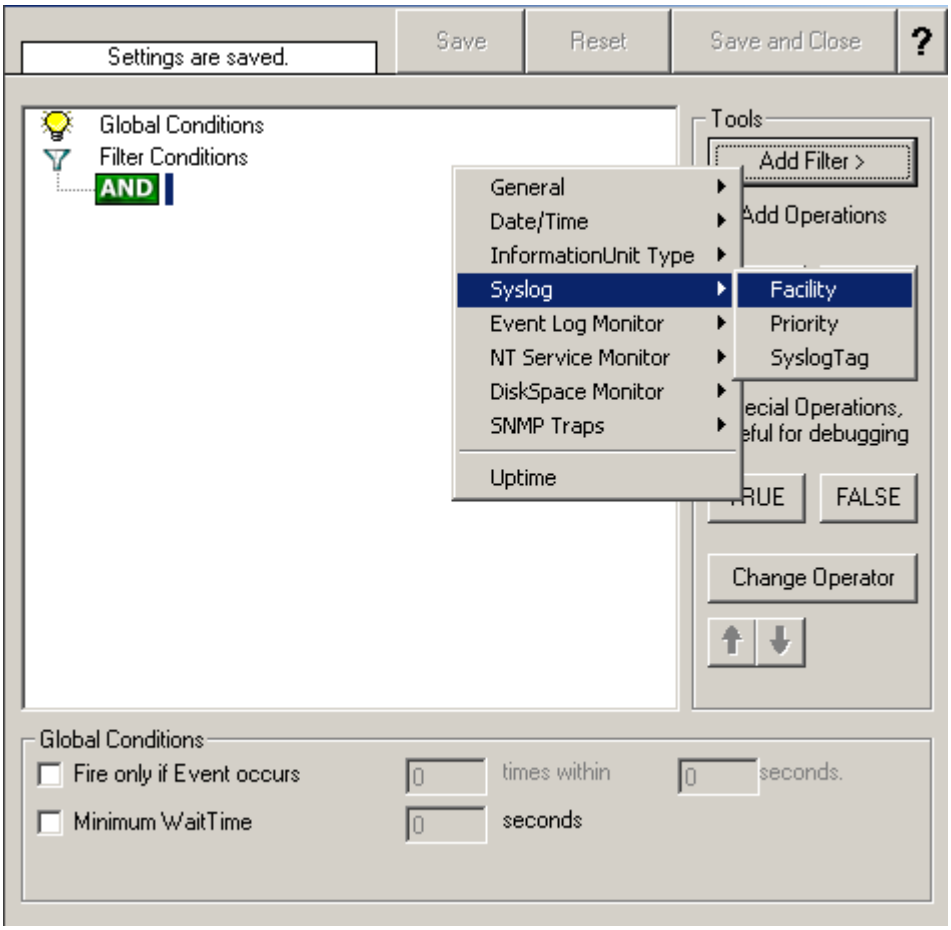
Port Probe (Type=Boolean)

NT Services Monitor (Type=Boolean)

Disk Space Monitor (Type=Boolean)

5.4.7 Syslog

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



Syslog Facility

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

This filter is of type number.

Syslog Priority

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

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

This filter is of type number.

Syslog Tag

This filter is of type string.

5.5 Actions

5.5.1 Understanding Actions

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

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

5.5.2 File Options

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

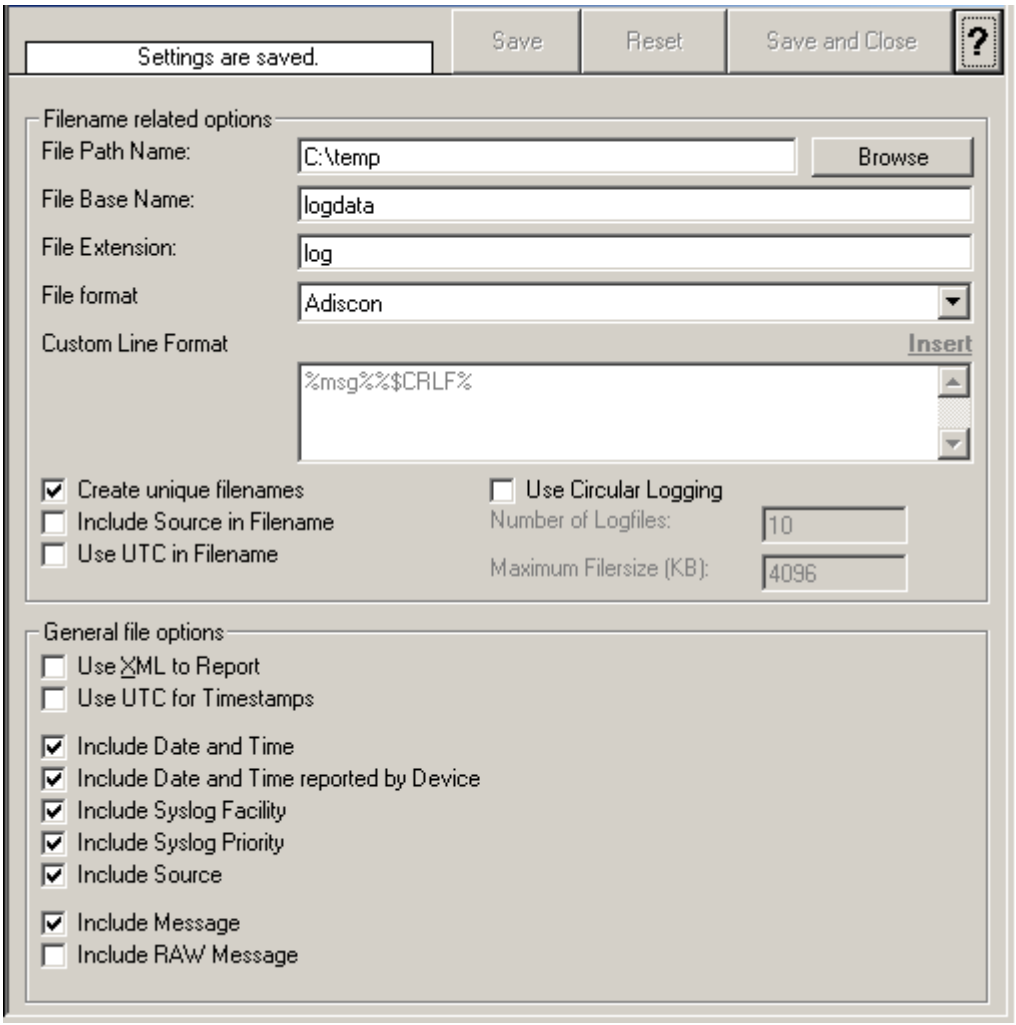
File logging is used to write text files of received messages. One file per day is written. New entries are appended to the end of the file.

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

The filename is build as follows:

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

With the parameters in brackets being configured via the dialog.



File Logging Options

Create unique Filenames

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

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

Use UTC in Filename

This works together with the "Create unique Filenames" setting. If unique names are to be created, the "Use UTC in Filename" selects if the file name is generated based on universal coordinated time (UTC) or on local time. UTC was formerly referred to as "GMT" and is the basis of the time zone system. For example, New York, USA is 5 hours behind UTC. Therefore, if it is 12 noon in New York, the UTC time is 5pm.

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

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

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

File Path Name

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

File Base Name

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

File Extension

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

File Format

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

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

The "WebTrends Syslog compatible" mimics the format that WebTrends applications expect. Please note that we only mimic the log file format. It is still the job of the reporting device (most notable firewall) to generate the correct WebTrends WELF format. The "WebTrends" format is supported because many customers would like to use MonitorWare Agent 2.0 enhanced features while still having the ability to work with WebTrends.

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

Include Source in Filename

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

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

Use XML to Report

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

Use UTC for Timestamps

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

Include <Fieldname>

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

Please note the difference between the "Date and Time" and "Date and Time reported

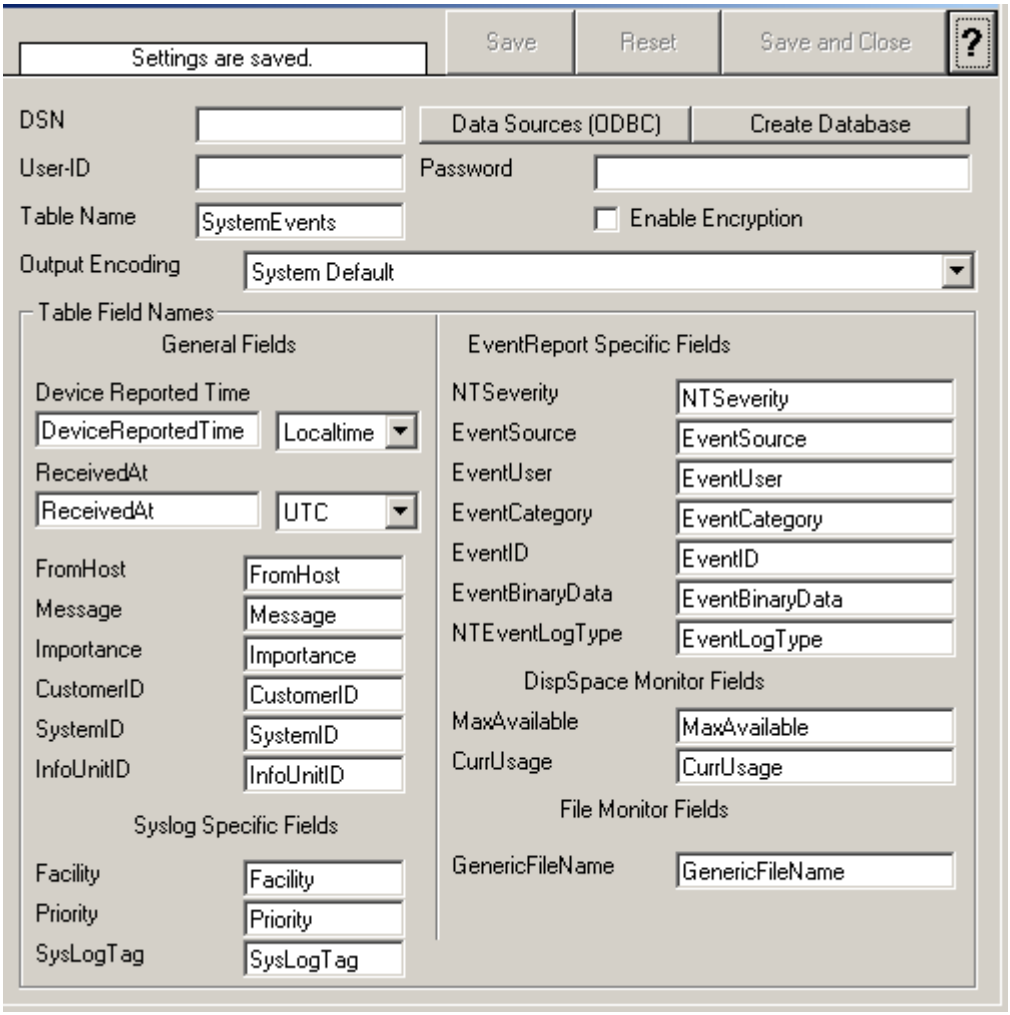
by Device". Both are timestamps. Either both are written in local time or UTC based on the "Use UTC for Timestamps" check box. However, "Date and Time" is MonitorWare Agent 2.0 received the time the message. Therefore, it always is a consistent value.

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

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

5.5.3 Database Options

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



Database Logging Options

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

DSN

This is the name of the system data source (DSN - data source name) to be used when connecting to the database. Create this in ODBC manager (can be found in control panel under Windows NT). Press the "Data Sources (ODBC)" button to start the operating system ODBC Administrator where data sources can be added, edited and removed.
Important: The DSN must be a system DSN, not a user or file DSN. The DSN must be

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

User-ID

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

Password

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

Enable Encryption

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

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

Table Name

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

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

Table Field Names

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

present. Otherwise, the ODBC writer will fail.

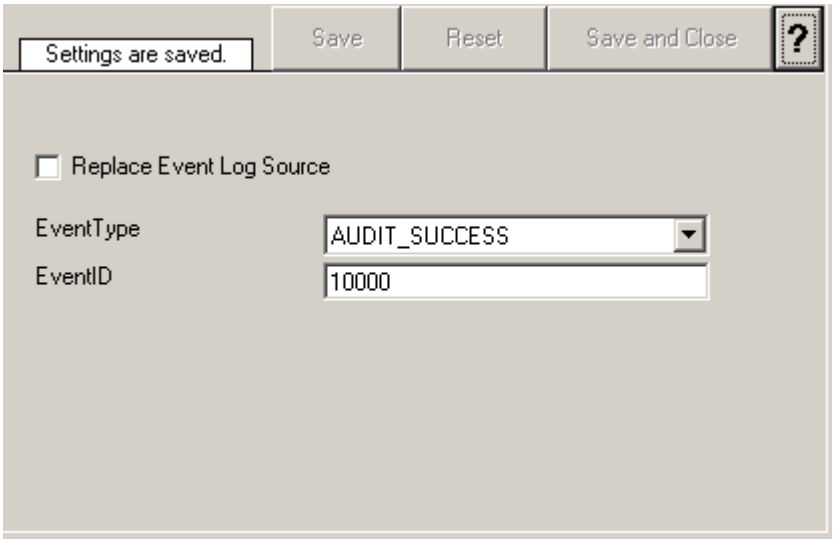
Please note that the default field names must be used when other members of the MonitorWare family (like the web interface or the MonitorWare Console) should work with the database. This customization option is meant for those customers that use third-party or custom software.
Important

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

The database conforms to the Common MonitorWare Database Format.

5.5.4 Event Log options

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



Event Logging Options

Replace Event Log Source

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

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

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

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

EventType

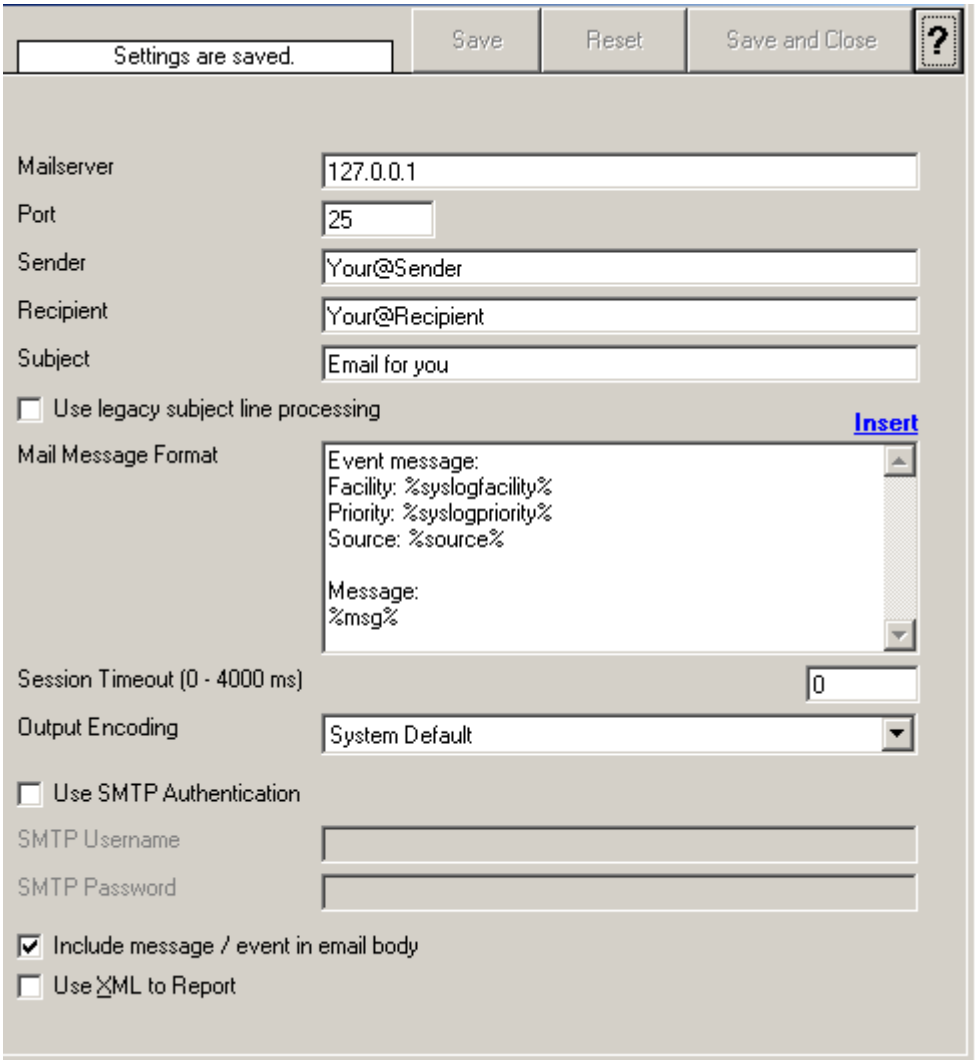
The type – or severity – this log entry is written with. Select from the available Windows system values.

EventID

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

5.5.5 Mail Options

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



Forward Email Properties

Mailserver

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

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

Port

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

Sender

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

Recipient

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

Subject

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

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

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

Use legacy subject line processing

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

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

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

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

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

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

Mail Message Format

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

Session Timeout

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

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

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

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

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

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

Use SMTP Authentication

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

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

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

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

Include message / event in email body

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

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

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

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

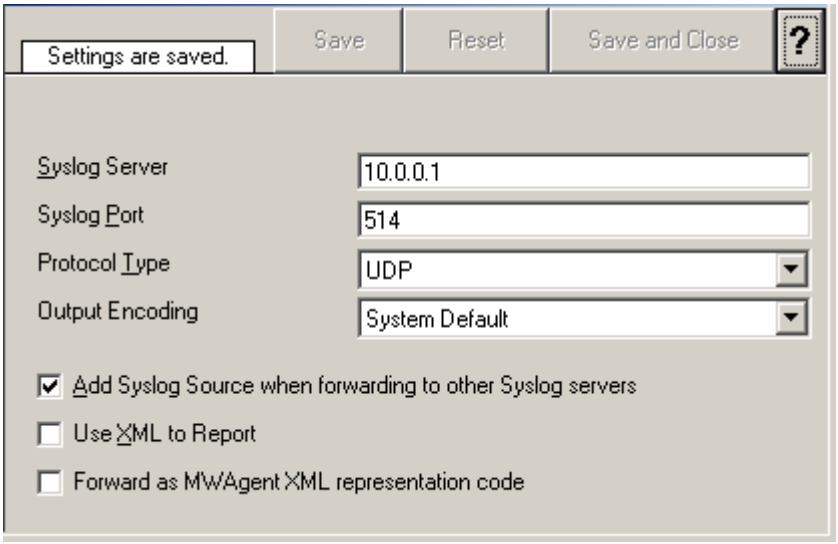
Use XML to Report

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.

5.5.6 Forward Syslog Options

This dialog controls Syslog forwarding options.



Forward Syslog Properties

Syslog Server

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

Syslog Port

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

Protocol Type

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

Output Encoding

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

Add Syslog Source

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

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

Use XML to Report

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

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

Forward as MW Agent XML Representation Code

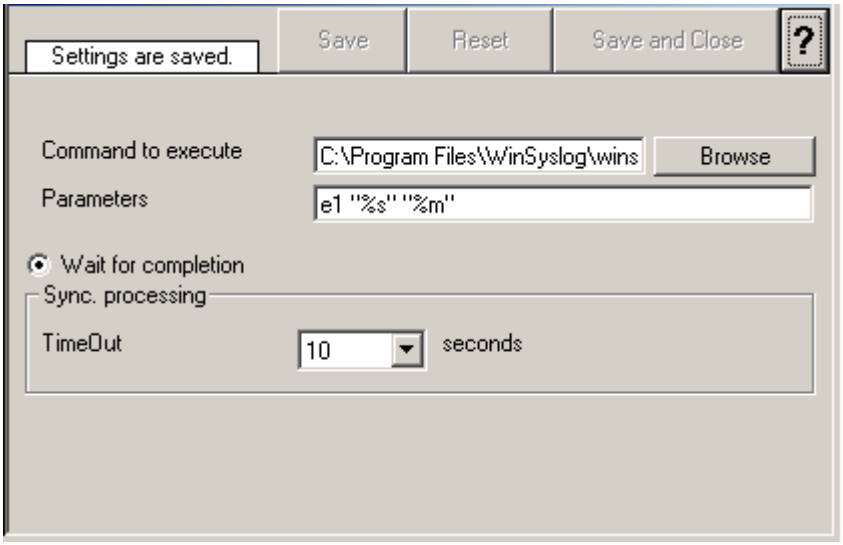
MonitorWare supports a specific XML-Representation of the event. If it is checked, that XML representation is used. It provides additional information (like InformationUnit Type, original source system, reception time & many more) but is harder to read by a human. At the same time, it is obviously easier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

5.5.7 Start Program

This dialog controls the start program options.

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

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



Start Program Dialog

Program to execute

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

Parameters

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

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

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

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

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

Time Out

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

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

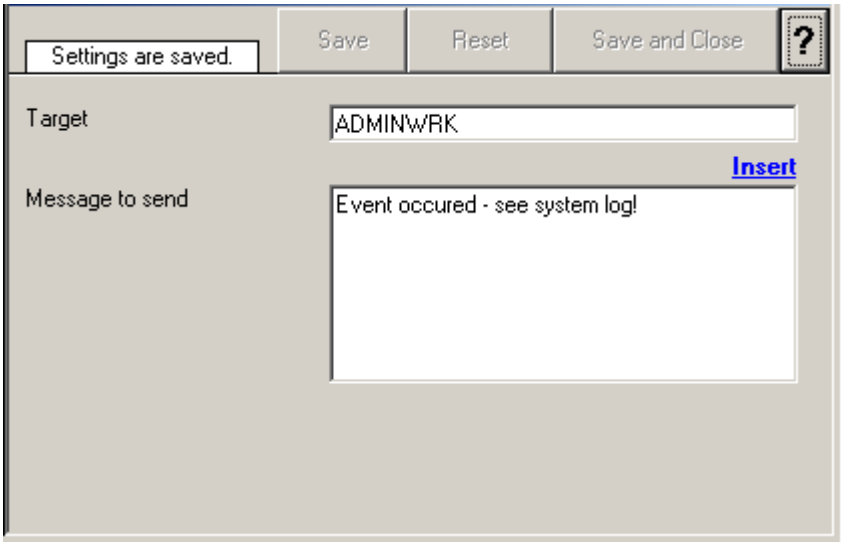
Important: Even though the timeout value can be as high as 30 seconds, we strongly recommend limiting the run time of external program to below 5 seconds. Otherwise, they could affect the overall performance too much. If the average run time is 5 seconds, the default timeout of 10 seconds ensures that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

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

5.5.8 Net Send

This dialog controls the net send options.

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



Net Send Dialog

Target

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

Message to Send

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

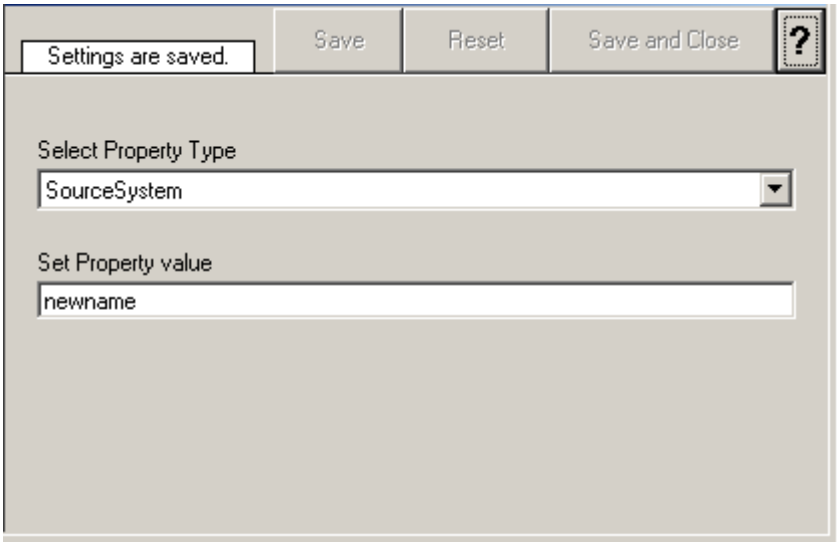
5.5.9 Set Property

This dialog controls the set property options.

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

equally named devices.

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



Set Property Dialog

Select Property Type

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

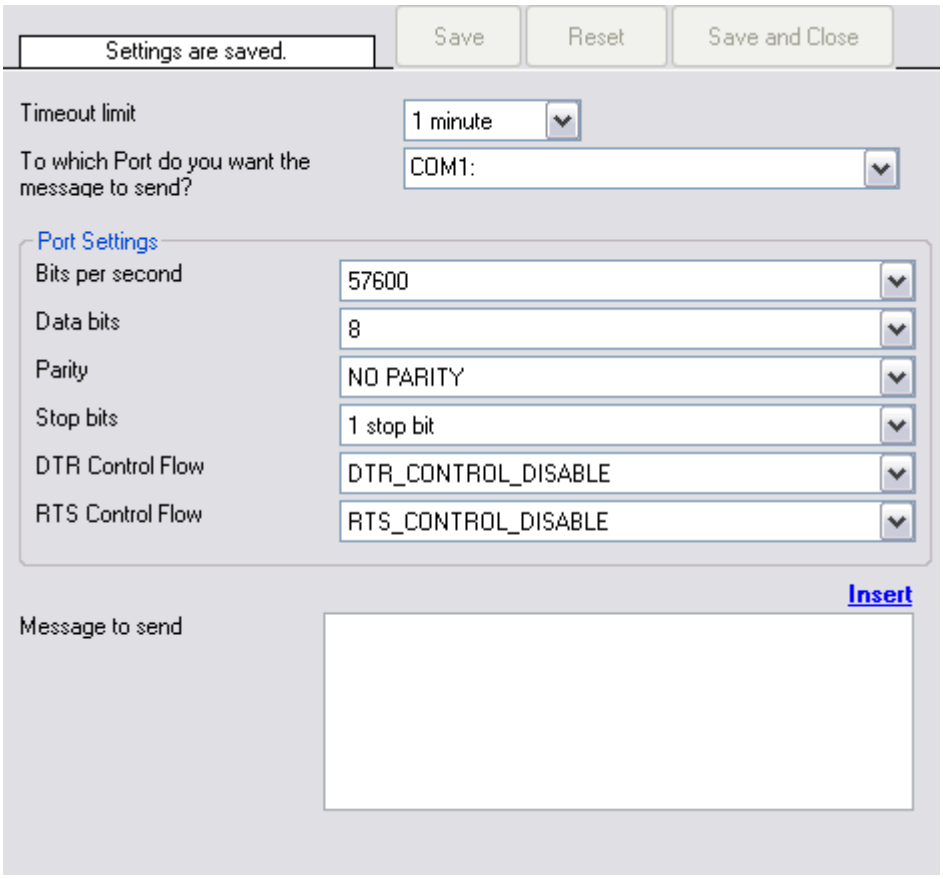
Set Property Value

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

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

5.5.10 Send to Communications Port

This action allows you to send a string to an attached communications device.



Send to Communications Port Options

Timeout Limit

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

Port to Send To

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

Port Settings

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

Message to Send

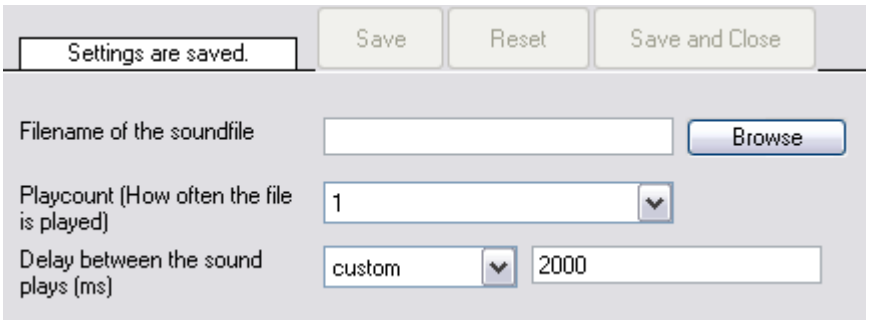
This is the message that is to be send to the device. You can enter text plainly and you can also include all properties from the current event. For example, if you have a serial audit printer and you

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.

Event properties are described in the property replacer section.

5.5.11 Play Sound

This action allows you to play a sound file.



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

Filename of the Soundfile

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

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

Playcount

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

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

Delay between Plays

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

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

The FAQ area is continously 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 to not receive responses from us, it is a very good idea to check the web interface.** Unfortunately, anti-SPAM measures are being setup more and more agressive. We are noticing an increasing number of replies that simply do not make it to your mailbox, because some SPAM filter considered it to be SPAM and removed it. Also, it may happen that your support question actually did not get past our own SPAM filter. We try very hard to avoid this. If we discard mail,

we send a notification of this, so you should at least have an indication that your mail did not reach us. Using the customer support system via its own web interface removes all SPAM troubles. So we highly recommend doing this if communication otherwise seems to be distrubed. In this case, please remember that notification emails may also get lost, so it is a good idea to check your ticket for status updates from time to time.

WinSyslog Web Site

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

Support Forum

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

Email

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

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

Online Seminars

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

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

Phone

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

Fax

Please direct your faxes to

+49-9349-928820

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

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

Software Maintenance

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

To learn more about UpgradeInsurance, please visit

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

Non-Technical Questions

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

Product Updates

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

Please visit www.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

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

Which Edition is for Me?

Information on all available WinSyslog editions can be found on the web at

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

This includes a feature and price comparison.

How to order

Using the Online Processing System

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

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

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

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

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

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

for details.

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

Direct your orders to:

Adiscon GmbH
Franz-Marc-Strasse 144
50374 Erftstadt
Germany

Fax: +49-2235-985032
Phone +49-9349-928820

email: order@Adiscon.com

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

Placing a Purchase Order

If you would like to order via purchase order, please visit http://www.adiscon.com/Common/en/OrderByPO.asp to get the details.

9 Reference

Here is the list of the references that we had used. We hope that it will help you to understand our product better.

- The WinSyslog Service
- Support for Mass Rollouts
- Formats
- Version History
- ICMP Codes
- Property Replacer

9.1 Formats

Database Format

WinSyslog stores and expects data in the "MonitorWare Common Database Format". All members of the MonitorWare line of products understand this format.

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

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

Microsoft SQL Server this is the "text" data type.

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

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

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

Database Samples

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

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

JET (MS Access) Sample

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

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

Microsoft SQL Server Sample

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

```
CREATE TABLE.SystemEvents
(
      ID int IDENTITY (1, 1) NOT NULL,
      CustomerID bigint,
      ReceivedAt datetime NULL,
      DeviceReportedTime datetime NULL,
```

```
	Facility smallint NULL,
	Priority smallint NULL,
	FromHost nvarchar (60) NULL,
	Message text,
	NTSeverity int NULL,
	Importance int NULL,
	EventSource nvarchar (60),
	EventUser nvarchar (60) NULL,
	EventCategory int NULL,
	EventID int NULL,
	EventBinaryData text NULL,
	MaxAvailable int NULL,
	CurrUsage int NULL,
	MinUsage int NULL,
	MaxUsage int NULL,
	InfoUnitID int NULL ,
	SysLogTag varchar(60),
	EventLogType varchar(60),
	GenericFileName VarChar(60),
	SystemID int NULL
)

CREATE TABLE SystemEventsProperties
(
	ID int IDENTITY (1, 1) NOT NULL ,
	SystemEventID int NULL ,
	ParamName varchar (255) NULL ,
	ParamValue text NULL
)
```

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

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

9.2 Property Replacer

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

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

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

%property:fromPos:toPos%

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

Property

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

A property is either an always-present property, an event specific property, a dynamic property or a system property.

FromPos

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

ToPos

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

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

Examples

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

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

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

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

9.2.1 System Properties

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

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

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

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

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

Other trademarks mentioned belong to their respective owners. They are solely used for reference purposes.

11 Glossary of Terms

The Glossary of Terms is also available on the Web:

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

11.1 EventReporter

EventReporter is Adiscon's solution to forward Windows NT/2000/XP event log entries to central system.
These central systems can be either WinSyslog's, other syslog daemons (e.g. on UNIX) or MonitorWare Agents. EventReporter is part of Adiscon's MonitorWare line of products

More Information about EventReporter:
http://www.adiscon.com/Common/en/glossary/eventreporter.asp

11.2 Millisecond

A millisecond is a thousand of a second. It is abbreviated as "ms". As such, 500ms mean half a second.
Inside the MonitorWare line of products, many timers are expressed in milliseconds as a fine control over the services and actions is provided to the administrator.

More Information about Milliseconds:
http://www.adiscon.com/Common/en/glossary/Millisecond.asp

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

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

- liblogging (www.liblogging.org)

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

More Information about the MonitorWare Line of Products:
http://www.adiscon.com/Common/en/glossary/MonitorWare-Line-of-Products.asp

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.

More Information about the Resource ID:
http://www.adiscon.com/Common/en/glossary/Resource-ID.asp

11.5 SETP

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

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

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

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

More Information about SETP:
http://www.adiscon.com/Common/en/glossary/SETP.asp

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.

More Information about SMTP:
http://www.adiscon.com/Common/en/glossary/SMTP.asp

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. Tradidionally, under UNIX, there are facilities like KERN (the OS kernel itself), LPD (the line printer daemon) and so on. There are also the LOCAL_0 to LOCAL_7 facilities, which were traditionally reserved for administrator and application use.

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

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

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

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.

More Information about TCP:
http://www.adiscon.com/Common/en/glossary/TCP.asp

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.

More Information about UDP:
http://www.adiscon.com/Common/en/glossary/UDP.asp

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.

More Information about Upgrade Insurance:
http://www.adiscon.com/Common/en/products/upgrade-insurance-details.asp

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 centrel location is to consolidate events from senders in multiple time zones.

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

More Information about UTC:
http://www.adiscon.com/Common/en/glossary/UTC.asp