

WinSyslog Configuration Documentation

version 18.0

Adiscon GmbH

November 12, 2025

Contents

About WinSyslog	1
Manual	3
Introduction	5
Features	5
Send Syslog Test Message	5
IPv6	6
Multi-Language Client	7
Friendly and Customizable User Interface	7
Components	7
Add-on Components	8
InterActive SyslogViewer	8
How these components work together	9
System Requirements	11
InterActive SyslogViewer	11
Product Tour	13
Syslog server	13
Heartbeat	14
SNMP Trap Receiver	14
SETP Server	15
Write to File	15
Write to Database	16
Write to Event Log	17
Forward via eMail	18
Net Send	19
Play Sound	19
Syslog Support	19
Forward via SETP	20
Powerful Event Processing	21
Send Syslog Test Message	22
Set Status	23
Set Property	23
Send to Communication Port	24
Post Processing	24
Start Program	25
Friendly and Customizable User Interface	25
Multi-Language Client	26
Other Miscellaneous Features	26
Getting Started	29
Installation	29
Information for a Mass Rollout	29
Creating an Initial Configuration	30
Obtaining a Printable Manual	30
Organizing with RuleSets, Rules, and Actions	31
Step-by-Step Guides	33

Installations and Configurations	33
services	34
Actions	34
Centralized Monitoring	35
InterActive SyslogViewer	37
InterActive SyslogViewer	37
Features	37
Options & Configuration	37
General Options	39
Configuring	49
Configuring WinSyslog	49
Client Options	50
Client Tools	53
Using File based configuration	58
General Options	61
Services	74
Heartbeat	74
Syslog Facility	75
Resource ID	75
SETP Server	79
SNMP Trap Receiver	81
Syslog server	83
General Options	85
Filter Conditions	92
Actions	115
Post Processing	133
Net Send	143
General Options	156
Syslog Facility	167
Post Processing	177
Set Property	187
Set Status	188
Play Sound	189
Start Program	189
Getting Help	193
Frequently Asked Questions	193
Customer Service System	193
Phone	193
WinSyslog Web Site	193
Software Maintenance	194
Non-Technical Questions	194
Product Updates	194
Concepts	195
Purchasing	197
Articles	199
Difference between Set Status - Set Property Action	199

How can I use a second sound card with the Play Sound Action?	199
Default Timevalues Setting in EventReporter/MonitorWare Agent/WinSyslog explained	200
FAQ	201
Why are Logfiles sometimes not rotated in WinSyslog 17.5 or lower?	201
Is WinSyslog v18+ supported on Windows Server IoT 2025?	202
Troubleshooting the Start Program action in WinSyslog	202
Is MariaDB supported by the ODBC action?	204
Recommended Palo Alto Firewall Syslog Configuration	205
References	209
Comparison of properties	209
Event Properties	209
Complex Filter Conditions	222
WinSyslog Shortcut Keys	226
Command Line Switches	226
Edition Comparison	227
Connect to Computer	227
Registry Paths	228
System Error Codes	228
Information for a Mass Rollout	228
Glossary of Terms	231
Actions	231
Write to File	231
Write to Database	231
Forward via Email	231
Forward via SETP	231
Net Send	231
Start Program	231
Set Status	232
Set Property	232
EventReporter	232
Filter Conditions	232
FTP	234
HTTP	235
IETF	235
IMAP	235
Information Units	235
IPv6	235
Millisecond	236
Monitor Ware Line of Products	236
NNTP	236
POP3	236
RELP	236
Resource ID	236
RFC 3164	237
RFC 3195	237
RFC 5424	237

Rules	237
The Rule Engine	240
Actions	242
WinSyslog - Services	248
SETP	249
SMTP	249
SNMP	249
Syslog Facility	250
TCP	250
UDP	250
Upgrade Insurance	250
UTC	250
Copyrights	251
Index	253

About WinSyslog

WinSyslog is a powerful and professional-grade syslog server for Microsoft Windows. It is designed for demanding enterprise environments and provides the same core functionality as a Unix syslog daemon, but with extended capabilities, modular design, and seamless Windows integration.

WinSyslog is actively maintained and continuously enhanced. With over two decades of development, it is one of the most mature, robust, and feature-rich syslog solutions available for the Windows platform today.

Network administrators rely on WinSyslog to monitor critical infrastructure in real time and receive alerts the moment important events occur.

Syslog is a well-established protocol for centralized logging of system events. While it originated in the UNIX world, today it is used by nearly all networked devices—including routers, switches, firewalls, and printers—to report events, status updates, and diagnostics.

Microsoft Windows does not include a native syslog server (known as “syslogd” on UNIX). This is where Adiscon’s **WinSyslog** steps in. Developed since 1996, it was the first syslog server for Windows and remains a trusted solution for system and network administrators worldwide.

WinSyslog is developed by the same experienced team behind the industry-leading **Rsyslog** project, the de facto syslog standard in Linux. This shared expertise ensures deep protocol understanding and consistent implementation across platforms.

The product has evolved significantly since its early days. Originally released as “NTSLog”, it became “WinSyslog” starting with version 3 to reflect its expanded capabilities. Each release has introduced meaningful enhancements, with version 4 adding rule-based processing and modular services that allow for extremely flexible setups.

WinSyslog can operate as a standalone solution or be combined with other Adiscon tools—such as **MonitorWare Agent** and **EventReporter**—to form a comprehensive, centralized event monitoring and alerting system for Windows-based infrastructures.

Typical use cases include:

- Logging events from syslog-capable devices such as routers, switches, and printers
- Long-term storage of logs in text files, ODBC databases, or the Windows Event Log
- Real-time display of messages and automatic notification (e.g., via email) on critical events
- Running multiple concurrent syslog listeners on different ports

WinSyslog runs as a stable, low-maintenance Windows service that starts automatically with the system. Once configured, it operates reliably in the background, requiring no manual intervention.

With decades of field-tested experience, WinSyslog delivers unmatched reliability and versatility for professionals who demand robust, scalable, and secure event logging on Windows systems.

To learn more about the full suite of MonitorWare products, visit: <https://www.adiscon.com/products>

Manual

Introduction

In this first chapter we describe the features, components and system requirements.

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 set up and customize. We also support full documentation and support for large-scale unattended installations.

Powerful Actions

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

Interactive Server

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

Send Syslog Test Message

WinSyslog client comes with "Send Syslog Test Message" facility. It can be accessed via the "Tools" menu. This option enables to check if syslog messages being sent properly to the destination or not.

Please note that the "Send Syslog Test Message" sends udp syslog, only! It does not at all send rfc 3195, or syslog/tcp!

Freeware Mode

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

Standards Compatible

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

WinSyslog Web Access

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

Syslog Hierarchy

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

Email Notifications

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

Store Messages Persistently

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

Multiple Instances

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

Full Logging

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

Robustness

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

Minimal Resource Usage

WinSyslog has no noticeable impact on system resources. It was specifically written with minimal resource usage in mind. In typical scenarios, its 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.

Windows Service

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

IPv6

Support for IPv6 is available in all network related facilities of the engine. All network related actions will automatically detect ipv6 and IPv4 target addresses if configured. You can also use DNS resolution to resolve valid IPv6 addresses. Network related Services can either use IPv4 or IPv6 as internet protocol. In order to support both protocols, you will need to create two services. The only exception is the RELP Listener, which uses IPv4 and IPv6 automatically if available.

Multi-Language Client

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

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

Friendly and Customizable User Interface

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

Move up and Move down function has been added for actions in the WinSyslog Client.

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

Multiple RuleSets - Rules - Actions

With WinSyslog as many "RuleSets", "Rules" and "Actions" as necessary can be defined.

multiple rulesets - rules - actions

Handling for low-memory cases

MWAgent allocates some emergency memory on startup. If the system memory limit is reached, it releases the emergency memory and locks the queue. That means not more items can be queued, this prevents a crash of the Agent and the queue is still being processed. Many other positions in the code have been hardened against out-of-memory scenarios.

Runs on a large Variety of Windows Systems

Windows 2019/2016/2012/10/8/2008 (R2)/7/Vista/2008/2003/2003 (R2)/XP/2000; Workstation or Server – MonitorWare Agent runs on all of them.

Support for End-of-Life operating systems is only partially available. Only a minimal service installation may be possible. More details: information for a mass rollout

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 an Windows Service and coordinates all log processing and forwarding activity at the monitored system (server or workstation).

The service is the only component that needs to be installed on a monitored system. The WinSyslog service is called the product "engine". As such, we call systems with only the service installed the 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. The Client can also be used to control service instances

x64 Build

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

- ODBC Database Action fully runs on x64 now. Please note that there are currently very few ODBC drivers for x64 available!
- Configuration Registry Access, a DWORD Value will now be saved as QWORD into the registry. However the Configuration Client and Win32 Service Build can handle these data type and convert these values automatically into DWORD if needed. The Configuration Client will remain a win32 application. Only the Service has been ported to the x64 platform.

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

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

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

Add-on Components

Adiscon offers several optional components as free downloads.

All optional components seamlessly integrate with the MonitorWare common database format.

database format.

InterActive SyslogViewer

The [InterActive SyslogViewer](#) is a Windows GUI application that receives and displays Syslog events. It functions as a standalone Syslog server. Typically, you use it in conjunction with a Syslog Forward Rule within the service, but it can also operate independently.

Although not a core component, the MonitorWare Agent installation set includes it.

For more information, see the [InterActive SyslogViewer online manual](#).

Adiscon LogAnalyzer

[Adiscon LogAnalyzer](#) provides a convenient web-based interface to access events gathered by MonitorWare. It supports all major browsers.

Adiscon LogAnalyzer offers an easy-to-use solution for Browse Syslog messages, Windows Event Log data, and other network events via the web. It enables system administrators to quickly and easily review their central log repository. It provides commonly used views for log data and integrates with web resources for simple analysis of data found in the logs.

Its primary benefit is offering a quick overview of current system activity and allowing access to log data even when you cannot access the administrator workstation (e.g., when traveling or moving through the enterprise). While initially designed to work with Adiscon's MonitorWare product line, you can easily modify it to integrate with other solutions.

Adiscon LogAnalyzer is included in the MonitorWare Agent installation set; the installer copies it to the machine but does not automatically install it. For installation instructions, refer to the documentation in the Adiscon LogAnalyzer's doc folder or see the [online manual](#).

Tools Available from the Tools Folder

Logger

Adiscon **Logger** is a command-line tool for Windows that functions like the UNIX `logger` command. This re-written tool offers enhanced functionality while supporting all popular UNIX options. It also provides reliable syslog transport via [RFC 3195](#) and plain TCP, a feature found in other Adiscon products and tools like `syslog-ng`. Additionally, Logger includes options specifically for the Windows environment.

For more details, visit: [An UNIX-like logger for Windows](#).

LogZip

Adiscon **LogZip** is a command-line tool for Windows that zips log files. Its primary purpose is to collect log files and store them in a specified ZIP archive. You can easily integrate LogZip with the Windows Task Scheduler, allowing you to automatically archive and move unneeded log files to a different location. This keeps only recent log files available for review.

Note

LogZip requires the Microsoft .NET Framework to run. Ensure you have it installed.

For more details, visit: [LogZip - Archiving tool for Windows](#).

LogDeleter

Adiscon **LogDeleter** is a database log deleter and backup tool for Windows.

[LogDeleter](#) can delete database records older than a specified number of days and offers the option to back up data first. It operates via an ODBC database connection. For example, you can run it through a Scheduled Task once a week to back up and delete old records.

LogDeleter is part of [Adiscon's MonitorWare solution](#).

LogViewer

The **LogViewer** tool efficiently processes very large log files. While other tools often struggle with files larger than 100 MB, Adiscon's LogViewer handles files exceeding 1 GB with ease. It performs almost as quickly with 5 GB files as it does with 5 MB files.

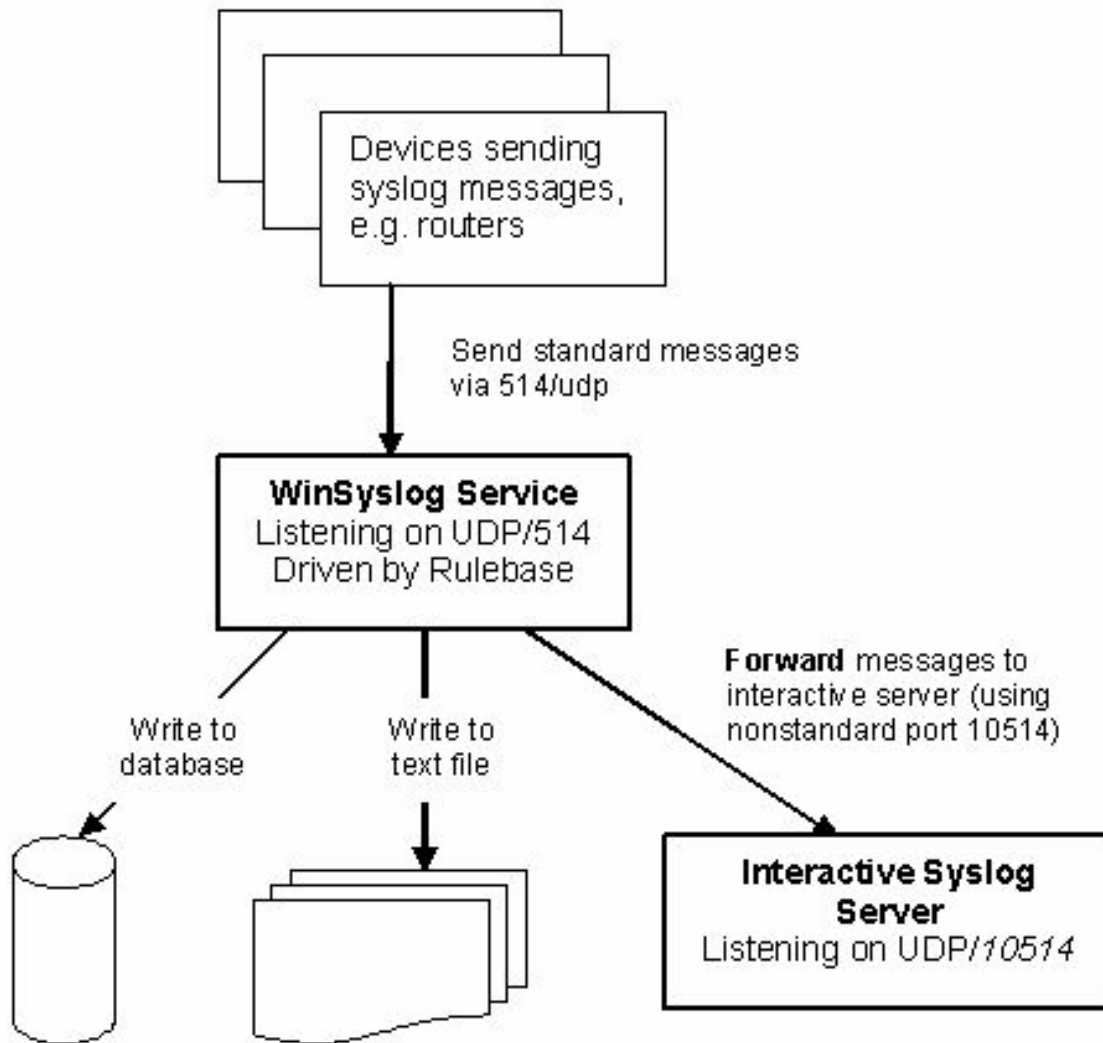
A special highlighting option assists in reviewing logs. You can define rules to highlight any keyword or phrase, associating terms with specific colors. This feature proves particularly useful when searching for specific errors or other log entries.

How these components work together

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

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

The message flow can be seen in this diagram:



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

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

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

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

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

Adiscon LogAnalyzer is only needed if accessing Syslog messages over the web is desired. It is a convenient facility to access MonitorWare gathered events over the web. All major browsers are supported. Adiscon LogAnalyzer is included in the WinSyslog install set. It gets copied onto machine but not installed. For installation of Adiscon LogAnalyzer, refer to the installation instructions in the doc folder of Adiscon LogAnalyzer or see the online manual at <https://loganalyzer.adiscon.com/doc/manual.html>. Please contact Adiscon via the [Customer Service System](#), if you want some more help in this regard.

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

System Requirements

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

Client

- The client can be installed on Windows 10, Windows 11, and Windows Server 2016/2019/2022. The operating system variant (Workstation, Server ...) is irrelevant. Note: For legacy systems (Windows XP, Server 2003), older versions are available - contact Adiscon for details.
- The client is suited for 32bit and 64bit operating systems. It runs automatically on each platform in 32Bit or 64Bit mode.
- The client uses Microsoft .Net Framework technology. The Installer will automatically install the necessary .Net Framework components before installation. A network connection maybe required in order to download additional components.
- The client requires roughly 8 MB RAM in addition to the operating system minimum requirements. It also needs around 5 MB of disk space.

Service

- The service has fewer requirements.
- It works under the same operating system versions.
- At runtime, the base service requires 5 MB of main memory and less than 5 MB of disk space. However, the actual resources used by the agent largely depend on the services configured.
- If the WinSyslog Service acts as a central Syslog server receiving hundreds of messages per second, it needs many more resources. Even then, the actual load is depending on the actions carried out. Storing the messages to text files is much less performance intense than writing them to a database table – especially if the database engine is located on the same machine. As such, there is no single guideline for hardware sizing. It needs to be adapted to the expected workload. We have created an article on [performance optimization for Syslog server operations](#), which you may want to read.
- Please note however, that the service is specifically optimized to handle high throughput including message bursts (for example received via syslog).
- If you expect high volume burst and carry out time consuming actions (for example database writes), we highly recommend adding additional memory to the machine. Please note that the 32Bit Service is limited to 2GB of usable memory. The 64Bit version does not have any limit. A typical Syslog message (including overhead) takes roughly 4-8 KB. With 1024 MB, you can buffer up to 100,000-200,000 messages in 1024 MB.
- Please note WinSyslog is capable of storing such bursts temporarily in memory even if the machine would otherwise be too slow to process the messages.

InterActive SyslogViewer

- SyslogViewer can be installed on Windows 10, Windows 11, and Windows Server 2016/2019/2022.
- SyslogViewer runs in 32Bit mode only. When accessing ODBC data sources, make sure to configure System DSN in the 32Bit ODBC Admin.
- SyslogViewer uses Microsoft .Net Framework technology. The Installer will automatically install the necessary .Net Framework components before installation. A network connection maybe required in order to download additional components.

- SyslogViewer requires roughly 15 MB RAM in addition to the operating system minimum requirements. It also needs around 2 MB of disk space.

Adiscon LogAnalyzer

- Adiscon LogAnalyzer requires a local web server installed. Microsoft Internet Information Server (IIS) or Apache with PHP support is recommended as Adiscon LogAnalyzer is a PHP based application. Adiscon LogAnalyzer is an optional component and not mandatory. We recommend using it with modern PHP versions (PHP 8.0 or higher) on Apache or IIS with current XAMPP, WAMP, or similar web server packages.

Product Tour

WinSyslog - Quick Tour

Services

Syslog server

This is a full-featured Syslog server, including support for syslog via TCP and RFC 3195. MonitorWare Agent helps to Configure a Syslog server service. It can be set to listen to any valid port. UDP and TCP communication is supported.

tion Client

Verify Configuration | Connect | Localhost | Start | Stop | Restart | Up | Down | DebugLog

Services > Syslog Server | Enabled | Comments | Settings | Confirm | Reset | ?

Test Syslog Server

Internet Protocoltype: IPv4

Protocol Type: UDP

IP Address: 0.0.0.0

Listener Port: 514

General | Encoding | UDP Options

☐ Resolve Hostnames

☐ Take source system from Syslog message

☐ Save original source into property

Propertyname: sourceorig Insert

☐ Escape control characters

☒ Enable RFC3164 Parsing

☐ Use original message timestamp (RFC 3164)

☒ Enable RFC5424 Parsing

☐ Append ProcessID to Syslogtag if available

RuleSet to use: Default RuleSet Refresh

- Syslog server*

Further details can be found here: [syslog server](#).

Heartbeat

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

The screenshot shows the 'Heartbeat' configuration page. At the top, there's a breadcrumb 'Services > Heartbeat' followed by a status 'Enabled' with a green checkmark. Action buttons include 'Comments', 'Settings' (checked), 'Confirm', 'Reset', and a help icon. The main configuration area includes:

- 'Message that is send during each heartbeat': A text area containing 'I am still running'.
- 'Heartbeat clock (Sleeptime)': A dropdown menu set to '1 Minute'.
- 'General Values' section:
 - 'Syslog Facility': 'Local 0' (dropdown)
 - 'Syslog Priority': 'Notice' (dropdown)
 - 'Syslog Tag Value': 'MWHeartbeat' (text input)
 - 'Ressource ID': An empty text input field.
- 'RuleSet to use': 'Default RuleSet' (dropdown) with a 'Refresh' button.

- Heartbeat*

Further details can be found here: [heartbeat](#).

SNMP Trap Receiver

SNMP Trap Receiver service allows 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. MonitorWare Agent supports decoding of MID values and also supports forwarding SNMP traps via other protocols, for example syslog.

The screenshot shows the 'SNMP Trap Receiver' configuration page. At the top, there's a breadcrumb 'Services > SNMP Trap Receiver' followed by a status 'Enabled' with a green checkmark. Action buttons include 'Comments', 'Settings' (checked), 'Confirm', 'Reset', and a help icon. The main configuration area includes:

- 'Internet Protocoltype': 'IPv4' (dropdown)
- 'Protocol Type': 'UDP' (dropdown)
- 'Listener Port': '162' (text input)
- 'SNMP Version': 'All supported Versions' (dropdown)
- Four checkboxes for output formatting:
 - ☐ Fully resolve Mibnames (Long Format)
 - ☐ Use short Format (Last Portion only)
 - ☐ Append MIB Description after Mibname (Attention, can be a lot of information!)
 - ☐ Compress Outputformat (Remove spaces/quotations)
- 'RuleSet to use': 'Default RuleSet' (dropdown) with a 'Refresh' button.

- SNMP Trap Receiver*

Further details can be found here: [snmp trap receiver](#).

SETP Server

MonitorWare Agent configures a SETP Server Service. A SETP Server is used inside the MonitorWare line of products to ensure reliable delivery of events from other systems. There are only few configuration options, as SETP takes the original message from the sender and uses the exact settings that the sender was configured for. No alteration occurs at the SETP server side; as such, no values need to be configured for the message format. SETP traffic can optionally be SSL-protected.

Services > SETP Server Enabled Comments Settings Confirm Reset ?

Internet Protocoltype: IPv4

Listener Port: 5432

Listener IP Address: 0.0.0.0

Session Timeout: 30 seconds

Options

- ☐ Enable SSL / TLS Encryption. Note if this Option is enabled, NON-SSL Clients will not be able to connect to this Service.
- ☐ Use zlib Compression to compress the data.
- ☐ Notify Sender about Rule Action Errors?

RuleSet to use: Default RuleSet Refresh

- SETP Server*

Further details can be found here: [setp server](#).

data collection

Write to File

All incoming events – no matter what source they came from – can be stored persistently. Options include archiving in databases as well as log files. 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.

RuleSets > Default RuleSet > Default Rule > File Logging Enabled Comments Settings Confirm Reset

Filename related options | File format | Post Processing

☐ Enable Property replacements in Filename

File Path Name Browse Insert

File Base Name Insert

File Extension

☒ Continuous Logging

☒ Create unique filenames

☐ Include Source in Filename

☐ Use UTC in Filename

☐ Segment files when the following filesize is reached (KB)

Segment Filesize (KB)

☐ Circular Logging

Number of Logfiles

Maximum Filesize (KB)

☐ Clear logfile instead of deleting (File will be reused)

File Handling Options

Output Encoding

Timeout until unused filehandles are closed

☐ Explicitly update create and modified file timestamp

- Write to File*

Further details can be found here: [write to file](#).

Write to Database

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.

Connection Options

Action Queue Options

SQL Options

Configure DSN

Verify Database

Create Database

DSN

User-ID

Password

SQL Connection Timeout

1 Minute

Enable Password encryption

SQL Options

Table Name

Statement Type

Output Encoding

SystemEvents

INSERT

System Default

Insert NULLValue if string is empty

Enable Detail Property Logging

Detaildata Tablename

Maximum value length (Bytes):

SystemEventsProperties

512

Datafields

	Fieldname	Fieldtype	Fieldcontent
	CurrUsage	int	cumusage
	CustomerID	int	CustomerID
	DeviceReportedTime	DateTime UTC	timereported
	EventBinaryData	text	%bdata%
	EventCategory	int	category
	EventID	int	id
	EventLogType	varchar	NTEventLogType
	EventSource	varchar	sourceproc
	EventUser	varchar	user

- Write to Database*

Further details can be found here: [write to database](#).

Write to Event Log

Allows any event (e.g. syslog, SNMP trap, protocol probes) to be written to the Windows Event Log. It is used to configure the logging to the Windows or XP event log. It is primarily included for legacy purposes.

RuleSets > Default RuleSet > Default Rule > EventLog Enabled Comments Settings Confirm Reset

☒ Use logsource from service
☐ Replace Event Log Source

Custom Eventlog Source: Insert

☐ Enable custom Eventlog Channel

Custom Eventlog Channel: Insert

Use Custom Eventlog Type: INFORMATION

Event ID:

Message to log: Insert

- Write to Event Log*

Further details can be found here: [event log](#).

alerting

Forward via eMail

Events of any kind can be forwarded via eMail. This is most often used for alerting. Together with your cell phone's provider eMail to messaging functionality, you can often send events directly to your cell phone. You can use this feature to receive eMail messages in your mail boxes.

RuleSets > Default RuleSet > Default Rule > Send Email Enabled Comments Settings Confirm Reset

Mail Server Options Mail Format Options

Mailserver:

Mailserver port:

☐ Enable Backup Server, used if first Mailserver fails

Backup Mailserver:

Backup Mailserver port:

☐ Use SMTP Authentication

SMTP Username:

SMTP Password:

Session Timeout: 0 (disabled)

☐ Use a secure connection (SSL) to the mail server
☐ Use STARTTLS SMTP Extension
☐ Use UTC Time in Date-Header

- Forward via eMail*

Here is an example how to receive **email notifications** when certain events happen.

Further details can be found here: [action send email](#).

Net Send

This helps to send short alert messages to recipient machine via Windows Net Send facility. Great for alerting logged-on administrators.

RuleSets > Default RuleSet > Default Rule > Net Send | Enabled | Comments | Settings | Confirm | Reset |

Target Machine

Message to send

- Net Send*

Here is an example how to receive **notifications via net send**.

Further details can be found here: [net send](#).

Play Sound

This action allows you to play a sound file.

RuleSets > Default RuleSet > Default Rule > Play Sound | Enabled | Comments | Settings | Confirm | Reset |

Filename of the soundfile

Playcount (How often the file is played)

Delay between the sound plays (ms) milliseconds

- Play Sound*

Further details can be found here: [play sound](#).

miscellaneous

Syslog Support

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

RuleSets > Default RuleSet > ForwardSyslog > Syslog Forwarding ✓ Enabled Comments Settings Confirm Reset

Protocol Type UDP

Syslog Target Options Syslog Message Options UDP related Options

Syslog Send mode

☒ Use single syslog server with optional backup server

Syslog Receiver Options

Syslog Server

Syslog Port

☐ Use this backup syslog server if first one fails.

Backup Syslog Server

Backup Syslog Port

☐ Use round robin (multiple syslog servers)

Amount of messages send to each syslog server before load balancing

Syslog Servers

	Syslog Server	Syslog Port
*	*Enter value for Syslog Server*	*Enter numvalue for Syslog Port*

Further details can be found here: [syslog forwarding](#).

Forward via SETP

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

RuleSets > Default RuleSet > ForwardSyslog > Send SETP ✓ Enabled 🗉 Comments ⚙️ Settings 💾 Confirm ↺ Reset ❓

Servename

Default SETP Port

☐ Enable SSL / TLS Encryption. Note if this Option is enabled, this action will not be able to connect to NON-SSL SETP Servers.

☐ Use zLib Compression to compress the data

Compression Level Best Compression

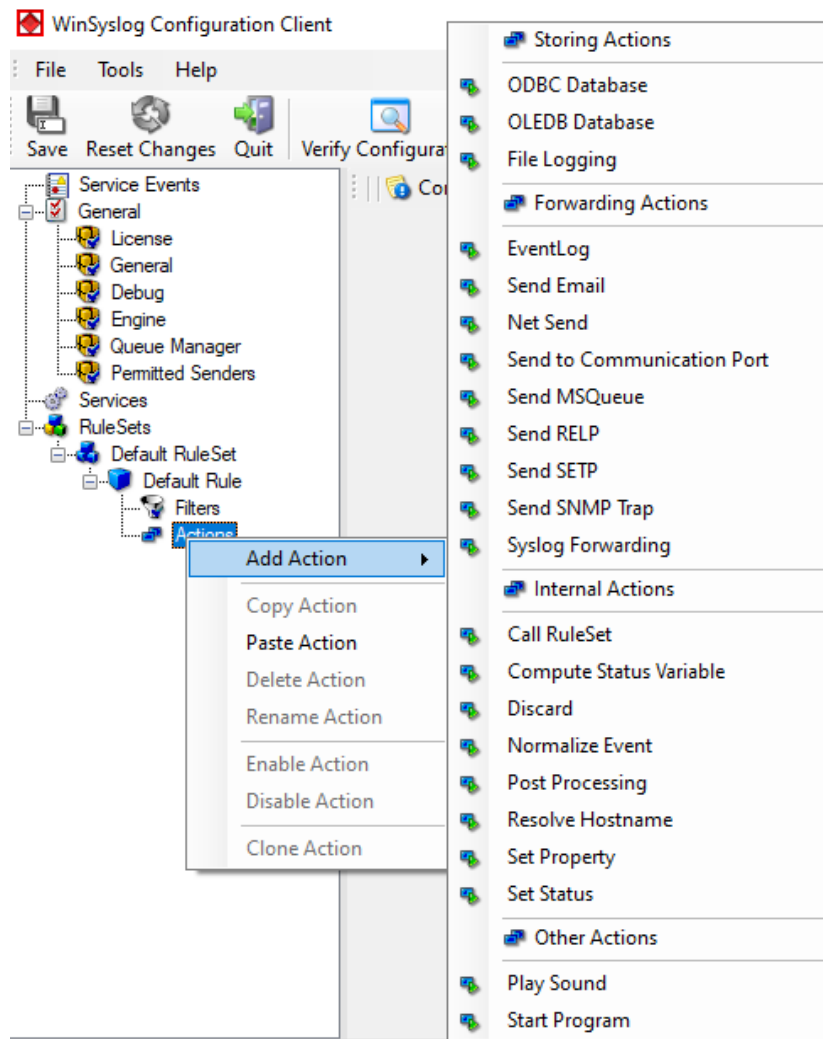
Timeout Options

Session Timeout	<input type="text" value="30 seconds"/>
Connection Timeout	<input type="text" value="30 seconds"/>
Send / Receive Timeout	<input type="text" value="5 Minutes"/>

Further details can be found here: [forward via setp](#).

Powerful Event Processing

WinSyslog has a powerful and flexible rule engine that processes all events based on a configured set of actions. An unlimited number of rules and actions allows tailoring to the specific needs.

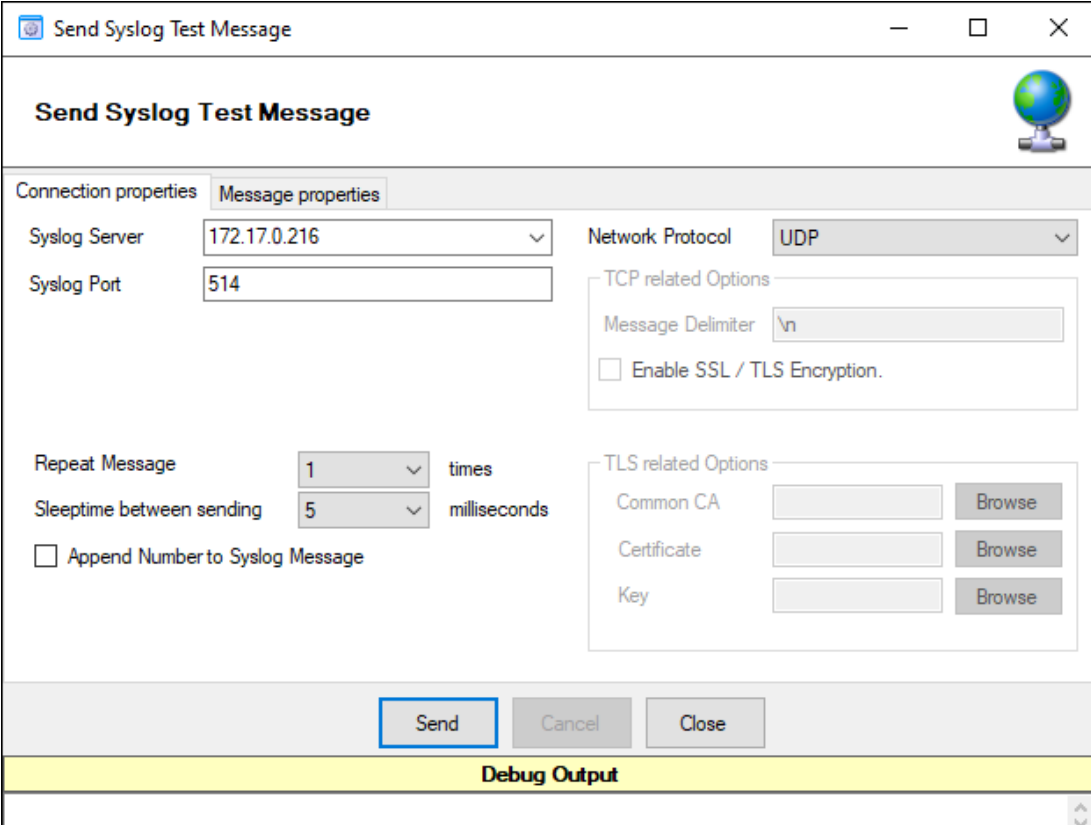


- Powerful Event Processing*

Further details can be found here: [actions](#).

Send Syslog Test Message

The MonitorWare Agent client comes with Send Syslog Test Message. This option enables to check if syslog messages being sent properly to destination or not.



Send Syslog Test Message

Connection properties | **Message properties**

Syslog Server: 172.17.0.216
 Syslog Port: 514

Network Protocol: UDP

TCP related Options
 Message Delimiter: \n
☐ Enable SSL / TLS Encryption.

Repeat Message: 1 times
 Sleep time between sending: 5 milliseconds
☐ Append Number to Syslog Message

TLS related Options
 Common CA: [Browse]
 Certificate: [Browse]
 Key: [Browse]

Send Cancel Close

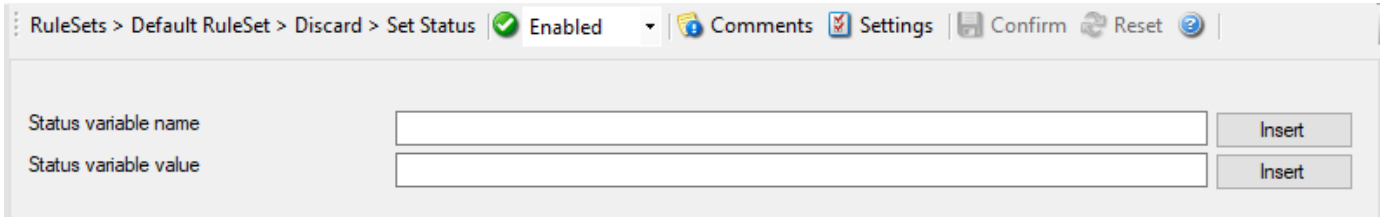
Debug Output

- Syslog Test Message*

Further details can be found here: client tools.

Set Status

Each information unit has certain properties e.g. EventID, Priority, Facility etc. You can create a new property and assign any valid desired value as well as filter to it. This is great for very demanding situations where complex rule sets are needed.



RuleSets > Default RuleSet > Discard > Set Status Enabled Comments Settings Confirm Reset ?

Status variable name: [] Insert

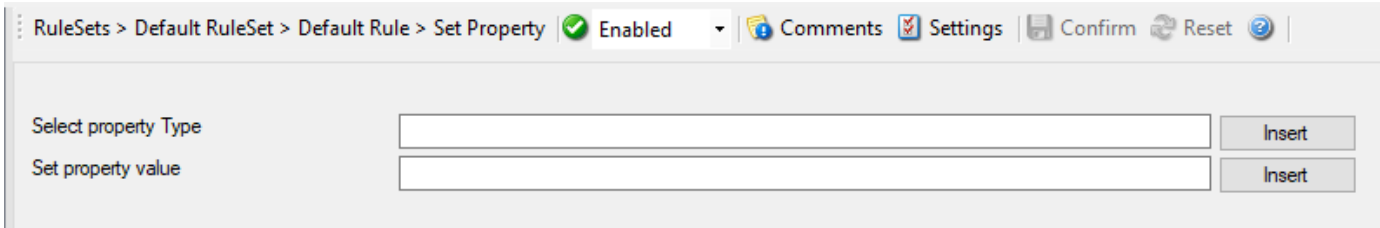
Status variable value: [] Insert

- Set Status*

Further details can be found here: set status.

Set Property

With the "Set Property", 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 servers.



RuleSets > Default RuleSet > Default Rule > Set Property Enabled Comments Settings Confirm Reset ?

Select property Type: [] Insert

Set property value: [] Insert

Further details can be found here: set property.

Send to Communication Port

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

RuleSets > Default RuleSet > Default Rule > Send to Communication Port ✓ Enabled 📄 Comments ⚙️ Settings 💾 Confirm

Timeout Limit: 1 Minute

Send message to this communication port: COM1:

Port Settings

Bits per second: 57600

Data bits: 8

Parity: No Parity

Stop bits: 1 Stop bit

DTR Control Flow: DTR Control Disable

RTS Control Flow: RTS Control Disable

Message to send: %msg%

Insert

- Send to Communications Port*

Further details can be found here: [send to communications port](#).

Post Processing

The Post Processing Action allows to re-parse a message after it has been processed e.g. Tab Delimited format. Such re-parsing is useful if you either have a non-standard event format or if you would like to extract specific properties from the message.

RuleSets > Default RuleSet > Default Rule > Post Processing ✓ Enabled 📄 Comments ⚙️ Settings 💾 Confirm 🔄 Reset ❓

Import Rules Export Rules

Property List

	Property Name	Type	Value
*	*Enter value for Property Name*	Character Match	*Enter value for Value*

Message Preview of your rules

- Post Processing*

Further details can be found here: [post processing](#).

Start Program

With this, 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.

The screenshot shows the 'Start Program' configuration window. It has a 'Command to execute' text box with a 'Browse' button. Below it is a checked checkbox for 'Use legacy parameter processing'. Under this is a 'Command Parameters' text box with an 'Insert' button. There is a radio button selected for 'Synchronous Processing (Wait for Completion)'. At the bottom, there is a 'Sync Timeout' dropdown menu currently set to '10 seconds'.

- Start Program*

Further details can be found here: [start program](#).

Friendly and Customizable User Interface

The Cloning feature helps to clone a Ruleset, a Rule, an Action, or a Service with one mouse click. It includes a Move up and Move down function for Actions in the Client.

With the MonitorWare Agent as many RuleSets, Rules and Actions as necessary can be defined.

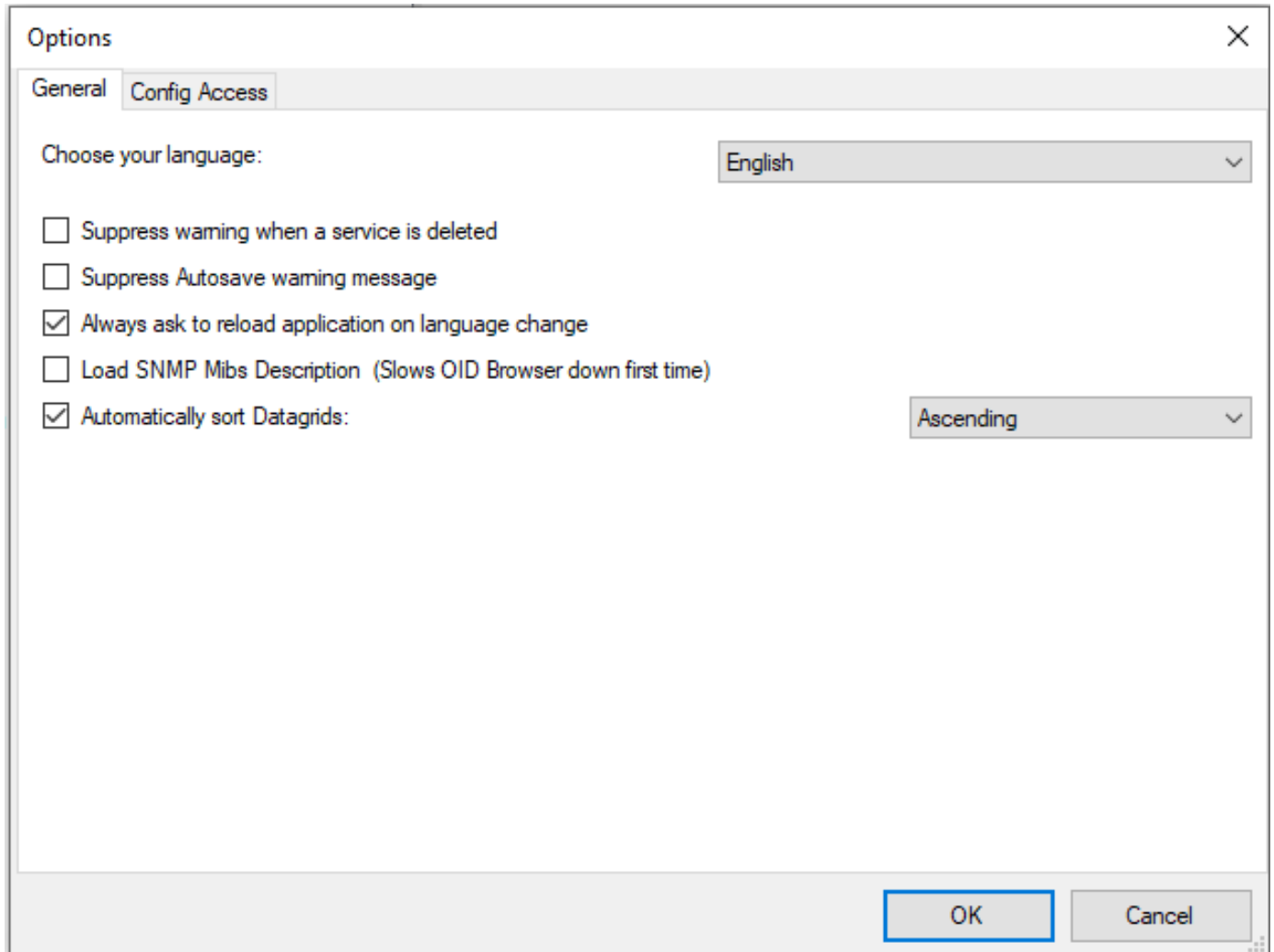
The screenshot shows the MonitorWare Agent configuration window. The left sidebar contains a tree view with 'Service Events' (General, License, Debug, Engine, Queue Manager) and 'Services' (Eventlog Monitor V1, RuleSets, Default RuleSet). The main area has tabs for 'General', 'Comments', 'Settings', 'Confirm', 'Reset', and a help icon. The 'General' tab is active, showing fields for 'Process Priority' (Normal), 'QueueLimit' (20000), 'SystemID' (0), 'CustomerID' (0), 'Location of your SNMP Mibs' (C:\Program Files (x86)\MonitorWare\Agent\mibs), and 'Default Timevalues are based on' (Universal Coordinated Time (UTC/GMT)). There are several checkboxes: 'Protect Service against shutdown', 'Log Warnings into the Windows Application Eventlog', 'Special Unicode Conversion for Japanese Systems', 'Automatically reload service on configuration changes' (checked), and 'Enable random wait time delay when checking for new configurations' (unchecked). Below the last checkbox is a 'Maximum random delay time' dropdown set to '5 seconds'.

- Friendly and Customizable User Interface*

Multi-Language Client

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

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



Other Miscellaneous Features

Full Logging

WinSyslog logs the received syslog message together with its 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.

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.

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.

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).

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.

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.

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 ruleset for execution.

Ease of Use

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

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, its 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 service. It can be controlled via the control panel services applet or the computer management MMC.

Full Modern Windows Support

WinSyslog provides comprehensive support for all current Windows operating systems, taking advantage of modern Windows features and security enhancements.

Runs on Wide Variety of Windows Systems

WinSyslog runs on all current Windows versions including Windows 10, Windows 11, Windows Server 2016, 2019, 2022, and newer versions. Both Workstation and Server editions are fully supported. Legacy support for Windows XP and Server 2003 is available in older product versions only.

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 set up and configure.

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

Installation

Installing the product is straightforward because a familiar Windows setup program guides you through the process.

Multiple download variants are available so you can pick the package that matches your environment; always install the most recent release to benefit from the latest fixes and improvements.

Each installer automatically creates a Windows Firewall exception for the service process during setup, ensuring the service can communicate right away without additional manual steps.

Installation is quick and easy. The WinSyslog Service uses a standard installation wizard.

We highly recommend visiting our [Online Seminars](#) to access the online seminars on WinSyslog as well as other members of this product family. These are not marketing videos but technically packed presentations that help you get started quickly and efficiently.

The installation walkthrough at [Installing WinSyslog](#) is a helpful reference, and WinSyslog is part of [Adiscon's MonitorWare line of products](#).

WinSyslog installers are available from the [Download Versions](#) page. Launch the setup by double-clicking `wnsyslog.exe` and following the onscreen instructions.

Information for a Mass Rollout

A mass rollout in this context means deploying an Adiscon client (for example MonitorWare Agent, EventReporter, or WinSyslog) to more than a handful of systems in an automated way. The aim is to invest the upfront effort needed to create a consistent "master" configuration once and then reuse it for every target machine. For guidance on differentiating between initial and update rollouts, see the MWAgent FAQ section.

Preparing the Baseline

1. Install the product on a single master system and configure it exactly as desired. Verify that the configuration works before continuing.
2. Export the configuration to a **registry file** via the configuration client (Computer → Export Settings).
3. Gather the files required for an engine-only installation:
 - For MonitorWare Agent 8.1 and newer this is typically `mwagent.exe` and `mwagent.pem`.
 - Older releases may also require the Visual C++ runtime and OpenSSL helper files (`Microsoft.VC90.CRT.manifest`, `libeay32.dll`, `ssleay32.dll`, `msvcm90.dll`, `msvcp90.dll`, `msvcr90.dll`).

Automated Rollout Example

Once the master system is prepared, copy the required files to a network share or removable media and automate the rollout with a script similar to the following:

```
copy \\server\share\mwagent.exe C:\some-local-dir
copy \\server\share\mwagent.pem C:\some-local-dir
cd C:\some-local-dir
mwagent -i
regedit /s \\server\share\configParams.reg
net start "AdisconMonitorWareAgent"
```

`configParams.reg` represents the registry export taken from the master system. Because the rollout ships only the engine files, this approach works well for DMZ environments where RPC or file sharing cannot be opened.

Note

`mwagent -i` (or the equivalent command-line switch for other Adiscon products) only registers the Windows service. It assumes the binaries already exist in the current directory, so copy the files before running the command.

Branch Office Rollouts

For branch offices or semi-automated deployments, distribute the prepared package and have the local administrator perform the following steps:

1. Create a directory on the target computer and copy the provided files into it.
2. Run `mwagent -i` from that directory to register the service.
3. Import the exported configuration by double-clicking the `.reg` file (or by running `regedit /s` from an elevated command prompt).
4. Start the Windows service via `net start` or the Services management console. Restarting the entire machine is not required.

Important

The directory that hosts the engine files **is** the installation directory. Deleting it removes the binaries and effectively uninstalls the product.

Updating Existing Rollouts

To upgrade an engine-only installation, update the master system first, export the revised configuration, and distribute the refreshed files using the same process. Uninstallation is unnecessary as long as you overwrite the files in place, but always stop the Windows service before copying the new binaries. For a walkthrough focused on update scenarios, refer to the MWAgent FAQ section.

Creating an Initial 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 ruleset** - The most basic ruleset includes no criteria, which means all incoming messages will match. To get started, we recommend using just a single **“Write to File”** action which will write the incoming messages to the local disk.
- **Create at least one syslog listener** - Be sure to associate the created ruleset with the **“Syslog Listener”**.
- **Start the WinSyslog service**
Your system is now ready to accept and store incoming messages.

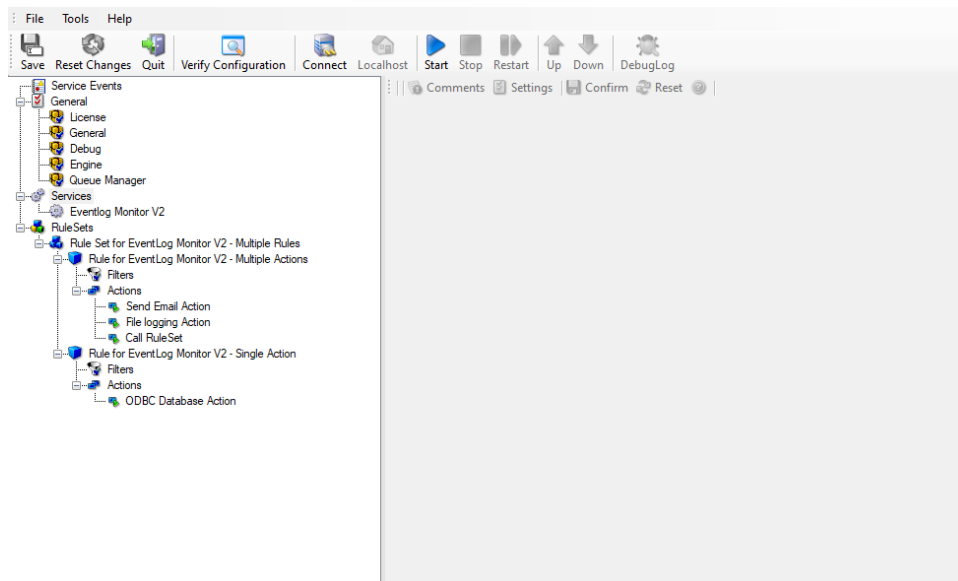
Obtaining a Printable Manual

A printable version of the manual can be obtained at <https://www.WinSyslog.com/help/manual/>.

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

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

Organizing with RuleSets, Rules, and Actions



WinSyslog gives you incredible flexibility to manage your log data. You can set up as many **RuleSets**, **Rules**, and **Actions** as you need to process your logs exactly the way you want.

RuleSets are like folders that help you group your rules. They make it easy to keep your log processing organized. For example, you could:

- Create a separate RuleSet for each service (like your firewall or web server). This keeps your configuration tidy because everything related to a specific service is in one place.
- Or, you might use a single RuleSet for all services if your logs require a more general kind of processing.
- You can also design RuleSets specifically to be called from within other RuleSets. This is done using the “callruleset action” and is great for reusing logic or structuring complex workflows.

Every RuleSet contains one or more **Rules**. These rules are at the heart of your log processing. They determine precisely what should happen with a log message. Think of a Rule as an “if-then” statement. It has two main parts:

1. **Filter Conditions:** These are the criteria a log message must meet for the rule to apply. For example, “Is it an error message?” or “Did it come from a specific IP address?” You can learn more about them in filter conditions.
2. **Actions:** These are the tasks that will be performed if the filter conditions are met. This could be writing the message to a file, sending it to a database, dispatching an email, and much more. A rule can include one or more actions. If several actions share the same filter conditions, you can conveniently combine them within a single rule.

Essentially, with WinSyslog, you use RuleSets to choose the processing area, Rules to define which messages you’re interested in, and Actions to specify what WinSyslog should do with them. This structure not only makes your configuration clearer but also helps you find issues faster and implement changes more easily. Our decades of experience in log processing ensure that Adiscon provides reliable solutions for businesses worldwide, handling even the most complex logging scenarios.

Step-by-Step Guides

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

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

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

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

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

Installations and Configurations

How to enter the license information

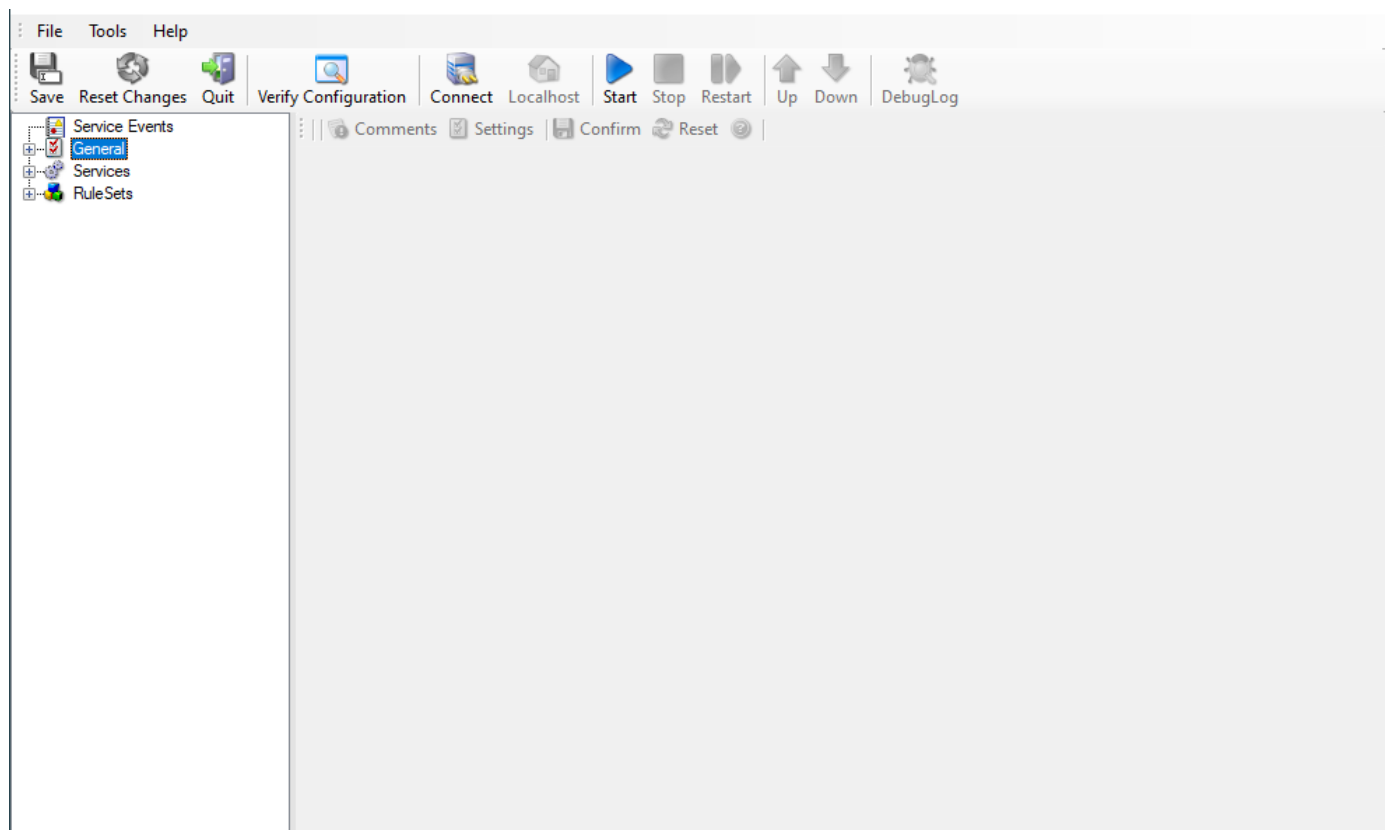
Article created 2021-10-05 by adiscon team

This article describes how to enter the license information you received via mail by buying one of our products.

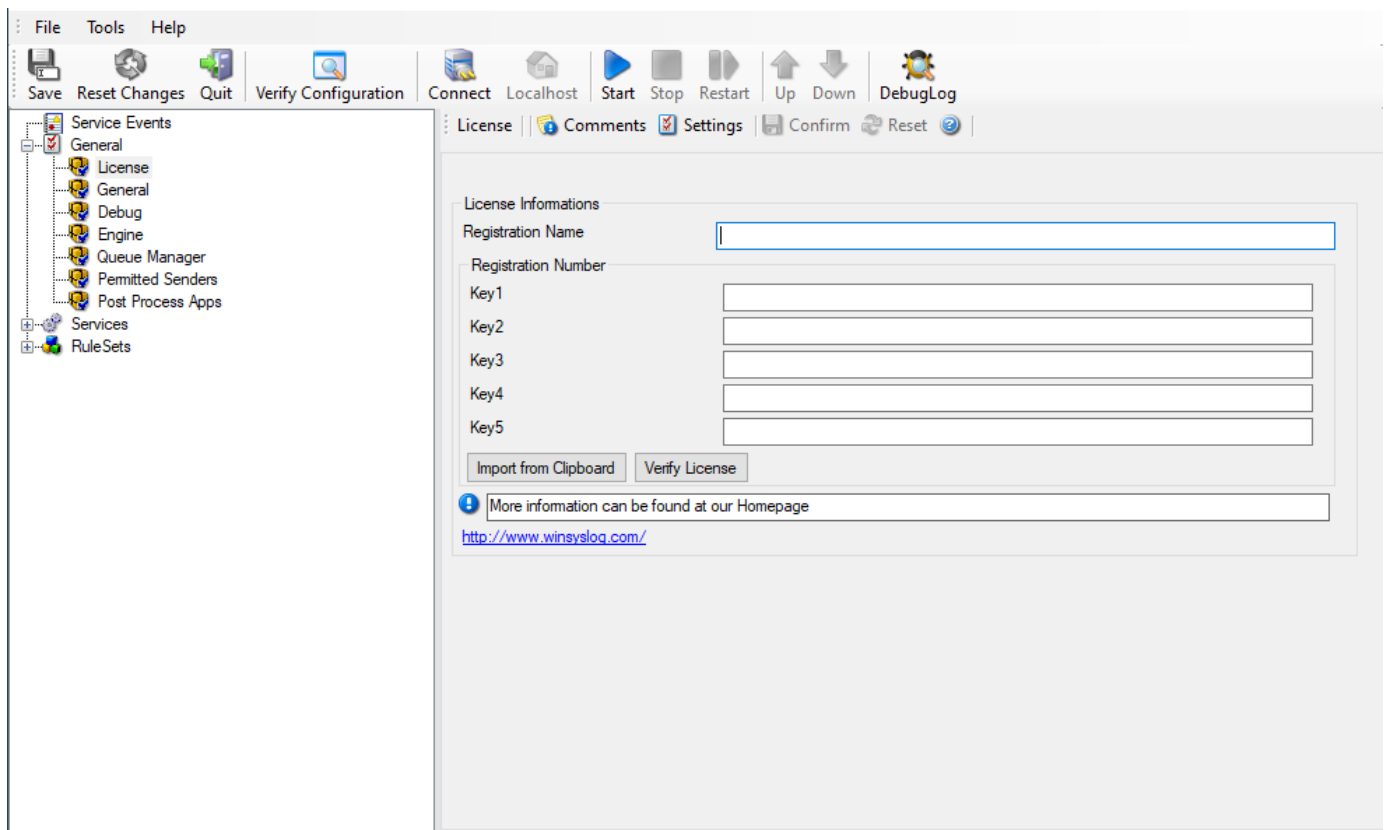
The Article is applicable to EventReporter, MonitorWare Agent, WinSyslog and Rsyslog WindowsAgent.

The license screen can be found on the left side of the client under the item General. Applying the license is very straightforward with only a few steps. After purchase, you will receive an email from us that contains the license name and key.

Under General - on the left side of the Configuration Client - you will find the menu entry: License.



If you click on it, you will find the license screen on the right.



The easiest way is to copy and paste the license name without quotation marks into the field “Registration Name” because it is case sensitive and must be entered exactly as given. Leading and trailing spaces are also part of the registration name. Be careful not to enter any.

Copy the full license key and use the button “Import from Clipboard” to paste it into the key fields. The client detects invalid registration numbers and reports the corresponding error.

Save the configuration and restart the service.

This is all that will be required to apply the license.

- [how do i export the configuration and create a debug file?](#)
- [how do i enter the license information from the product delivery email?](#)
- [monitorware agent database formats](#)
- [database logging with mssql](#)
- [how do i apply filters in monitorware agent, winsyslog, and eventreporter?](#)
- [how to setup monitorware agent/ winsyslog/ eventreporter](#)
- [how to setup php-syslog-ng with monitorware products?](#)

services

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

Actions

- [How To setup the Forward via Syslog Action](#)
- [How To setup an SETP Action](#)

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

Centralized Monitoring

- [How To setup PIX centralized Monitoring \(WinSyslog 8.x, MonitorWare Agent 5.x & MonitorWare Console 3.x\)](#)
- [How To setup Windows centralized Monitoring \(EventReporter 9.x & WinSyslog 8.x\)](#)
- [How To setup Windows centralized Monitoring \(EventReporter 8.x, WinSyslog 7.x, and Monilog 2.x\)](#)

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

InterActive SyslogViewer

InterActive SyslogViewer is a tool that lets you review your syslog data very easy. It is a separate syslog server, that simply displays all incoming data. By this you can see directly what is happening.

InterActive SyslogViewer

Features

Fast and Easy syslog Viewing

The SyslogViewer allows you to directly view and review syslog messages. Therefore you can react much better on occurring problems or check if everything is OK.

Review stored logs from a database

You can as well directly review log entries in a database. Simply enter the login details and that's it. You can then review your logs and even filter the view. That helps you to find the important data in an easy way.

Export selected data

You can export selected data for further manual processing, like sending an email to your colleague for informing them about what is happening.

Requirements

Any Windows-Windows based operating system like Windows XP, Vista, 7, 8, 10, 2003, 2008, 2012, 2016, 2019.

You need at least .NET 2.0 framework installed in order to run Adiscon's Syslog Viewer.

Hardware requirements: - 32MB RAM

Options & Configuration

InterActive SyslogViewer is an add-on to the MonitorWare Agent and WinSyslog. **Please note that it is a utility program, with a primary focus on real-time troubleshooting.****

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

Launching InterActive SyslogViewer

To run the InterActive SyslogViewer, click the "SyslogViewer" icon present in the Programs Folder -> MonitorWare Agent/WinSyslog located in the Start menu.

It can also be launched from the command prompt:

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

Available Command Line parameters are:

```
/? => Show Options
/autolisten => Start Syslog server automatically
/port=10514 => Overwrites the configured port
/windowpos 0,0,512,800 => Sets default window positions
```

Using InterActive SyslogViewer

InterActive SyslogViewer is an add-on to MonitorWare Agent and WinSyslog.

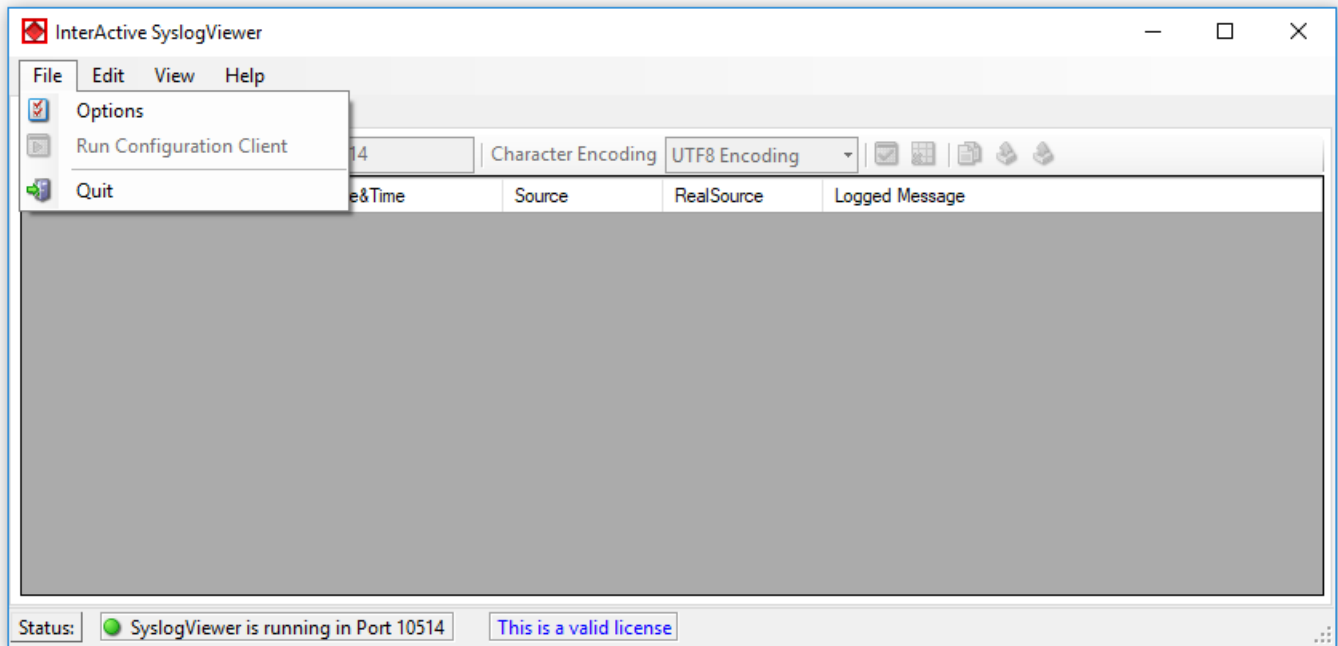
Please note that it is a utility program with a primary focus on real-time troubleshooting.

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

Options & Menus

Please find more information about the different menus and options in the respective sub-category.

File Menu



- InterActive SyslogViewer - File Menu*

Options

This will open the Options dialog. Please see the sub-chapters for more details on this.

General Options

The screenshot shows the 'Options' dialog box with the 'General Options' tab selected. The dialog has three tabs: 'General Options', 'Notifications & Questions', and 'License'. The 'General Options' tab contains several checkboxes and two input fields. The checkboxes are: 'Enable resolving Syslog RealSource (Adiscon specific)' (checked), 'Allow multiple SyslogViewer instances' (unchecked), 'Enable Trayicon (Hide Main Window on Close)' (unchecked), 'Start syslog viewer on launch' (checked), 'Show control characters (linefeeds) in datagrid view' (checked), 'Switch between two colors for each row in the Datagrid' (checked), 'Display Syslog message counter in Datagrid' (checked), and 'Show extra read/unread checkbox in Datagrid' (unchecked). The 'Max Messages in the Datagrid:' field is set to '10000'. The 'Select UI Language:' dropdown menu is open, showing 'Deutsch', 'English' (selected), and 'Japanese'. At the bottom are 'OK' and 'Cancel' buttons.

- InterActive SyslogViewer - General Options Tab*

Enable Resolving Syslog RealSource (Adiscon specific)

With this option enabled, you can see the real source in multiply forwarded messages. That means, you can see the system that forwarded the message and the system where the message originates from.

Allow multiple SyslogViewer instances

You can have multiple instances of the InterActive SyslogViewer by activating this option. This allows you to have multiple forwarding servers sending on different ports and receive their messages separately.

Enable Trayicon (Hide Main Windows on Close)

Enable this to have a tray icon. This enables a soft-close. InterActive SyslogViewer will stay active, but the window will be completely hidden except the tray icon. By double-clicking on the icon, the window will show again.

Start SyslogViewer on launch

Enable this to start the Syslog server directly when starting InterActive SyslogViewer.

Show control characters (line feeds) in data grid view

When enabled, you will see control characters like line feeds in the data grid as well.

Switch between two colors for each row in the data grid

To have a better overview over the syslog data, activate this option.

Display Syslog message counter in the data grid

You can enable a counter by checking the box here. It will count further, even if the maximum of messages is already exceeded.

Show extra read/unread checkbox in the data grid

If enabled, an additional checkbox is added for each record in the data grid that can be marked as checked.

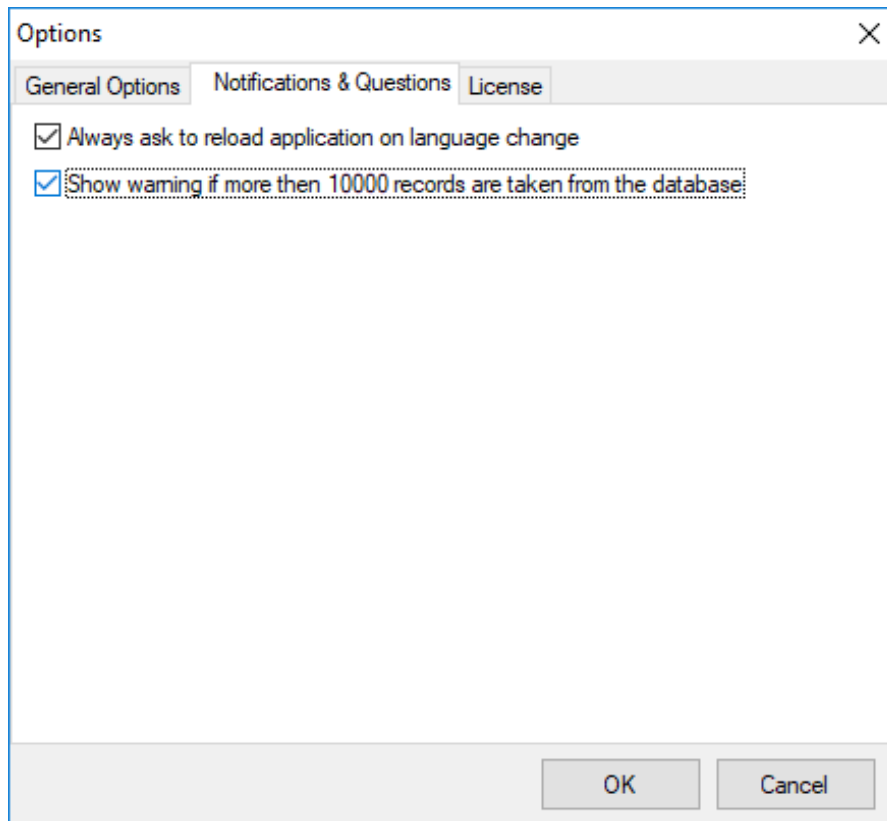
Max Messages in the data grid

Here you can adjust the maximum messages that will be available in the data grid. By increasing this value, you can store more messages for direct review. Please note, that increasing the maximum number of messages will have a severe impact on your memory.

Select UI Language

Here you can choose your favorite language for the InterActive SyslogViewer. By default it is English. You can choose German or Japanese as well.

Notifications & Questions Tab



- InterActive SyslogViewer - Notifications & Questions Tab*

Always ask to reload application on language change

While the box is checked, InterActive SyslogViewer will ask to reload the application on a language change. This is, because the language file can only be loaded while starting the application and not while it is running.

Show warning if more than 10000 records are taken from the database

By activating this option, you will be warned, if the records in the database are just too much. This is to prevent the machine from receiving too much load. Polling lots of messages from a database can have a severe impact on the performance of the machine.

License Tab

Options

General Options Notifications & Questions **License**

Add from Clipboard Add Edit Delete

Licensee Name	Key 1	Key 2	Key 3	Key 4	Key 5

Trial expired - click here enter your licensee

Click the link below to purchase the Interactive SyslogViewer

www.monitorware.com

OK Cancel

- InterActive SyslogViewer - License Tab*

Here you can insert the license. You have several options:

Add from Clipboard

This will insert the license you have currently on your clipboard.

Add

This button is to manually add a license manually. A new window will open, which shows you the form for entering the license information. This consists of a license name and five blocks of numbers.

Edit

Once a license is entered, it can be changed afterwards. This is done with this button. Mark the license you want to edit and click the button. A window will open which looks just like when adding a license, but the marked license details are inserted already. You can edit every field separately.

Delete

If a license is not needed anymore, you can delete it from the license screen. Mark the license and hit the button. The license will be deleted directly.

Please note, that the screen will give you additional information. You have

an overview of the licenses used and if not entered correctly it will show how long your trial period still is.

run configuration client

this option will open the configuration client of MonitorWare Agent/WinSyslog. here you can do detail configuration of the service.

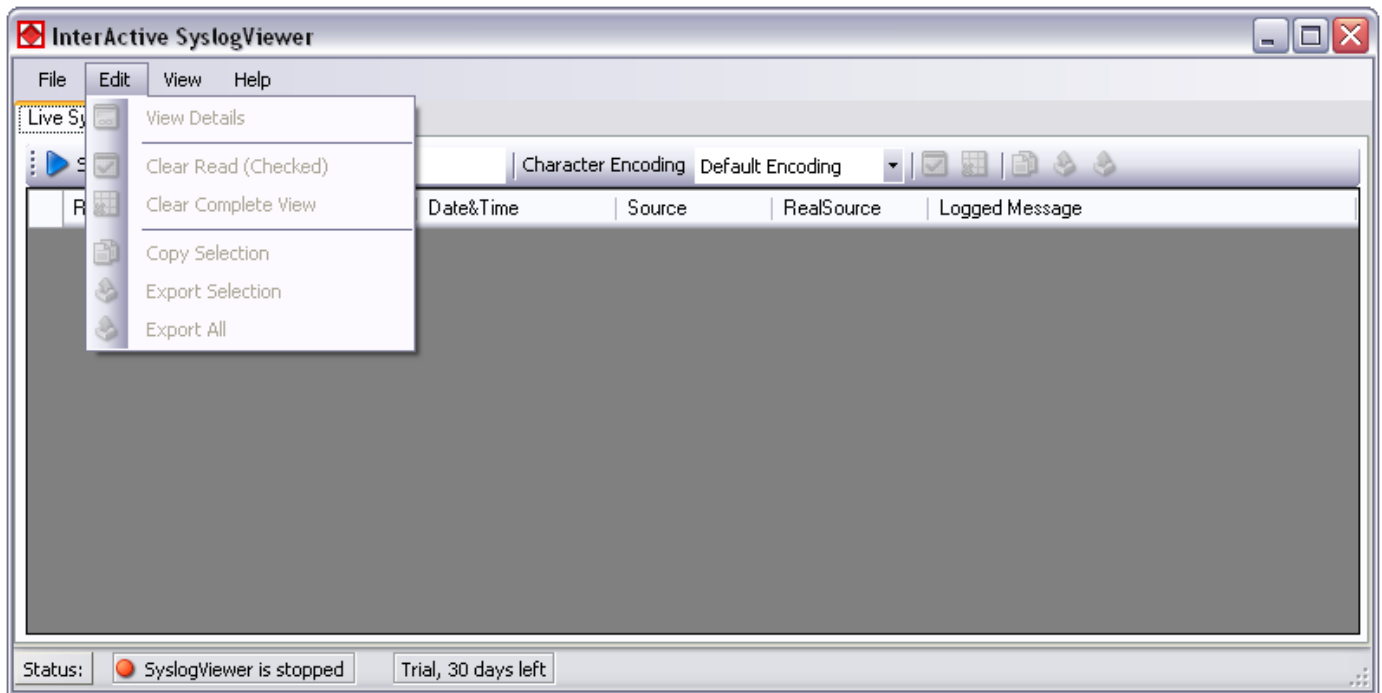
minimize to tray

this will minimize the interactive syslogviewer window and remove it from the taskbar. you can open it again by double-clicking on the icon in the system tray.

quit

by clicking here, interactive syslogviewer stops receiving data and it will close the application.

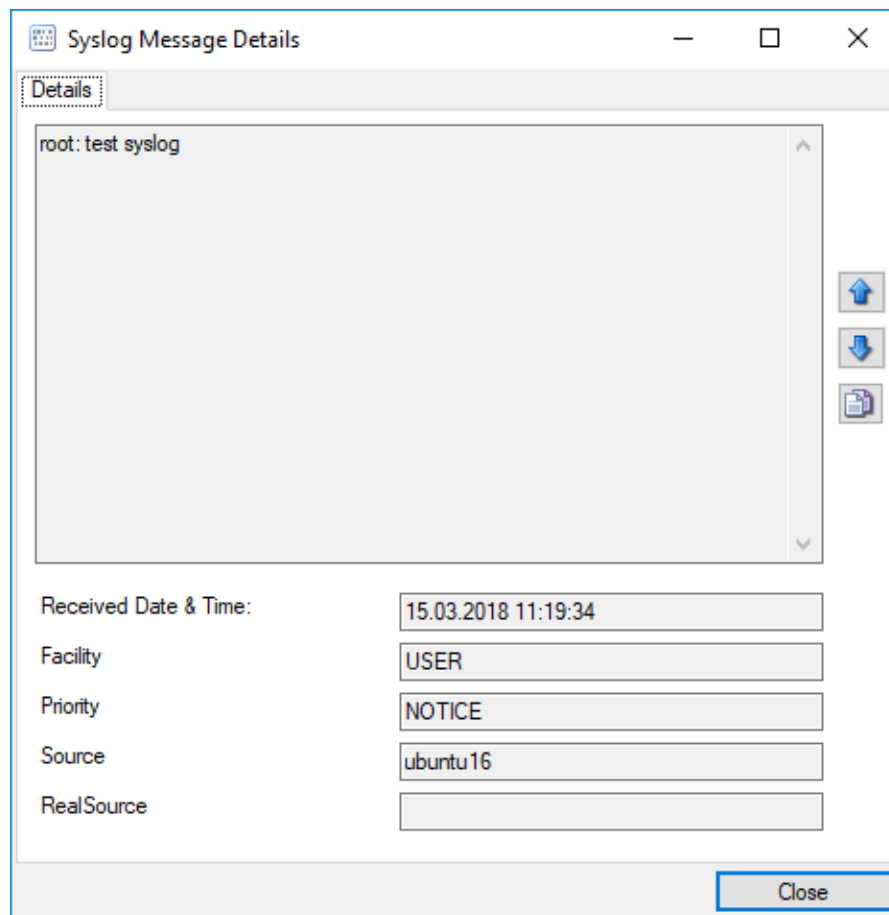
Edit Menu



- InterActive SyslogViewer - Edit Menu*

View Details

When using this option, another window will open up, which shows the details of this event in a more readable view. This could look like this:



- InterActive SyslogViewer Syslog - Message Details*

Clear Read (Checked)

By activating this, you can clear the checkboxes of the items your marked as read.

Clear Complete View

This option will clear the screen and remove all received data from the view.

Copy Selection

Having selected one or multiple entries, you can copy them using this function.

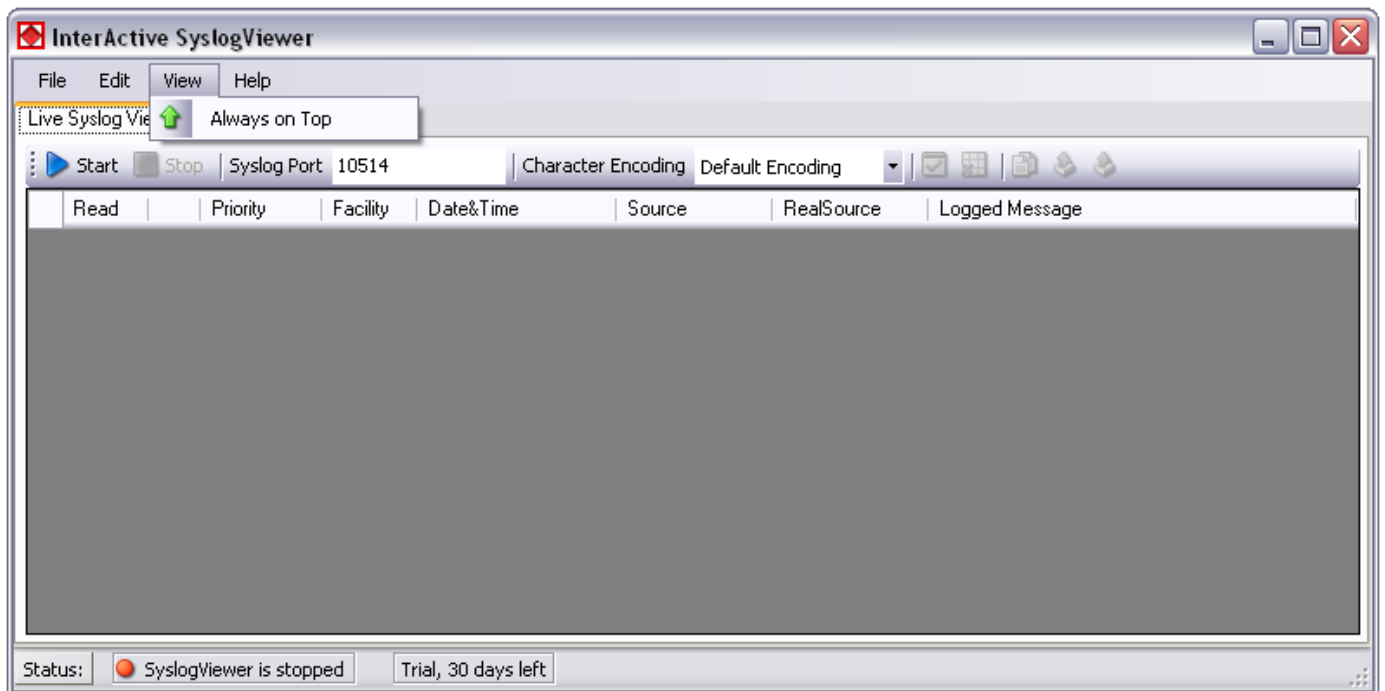
Export Selection

Instead of copying you can extract the selected data into a text file.

Export All

Or you directly export all the data that is currently in the list.

View Menu

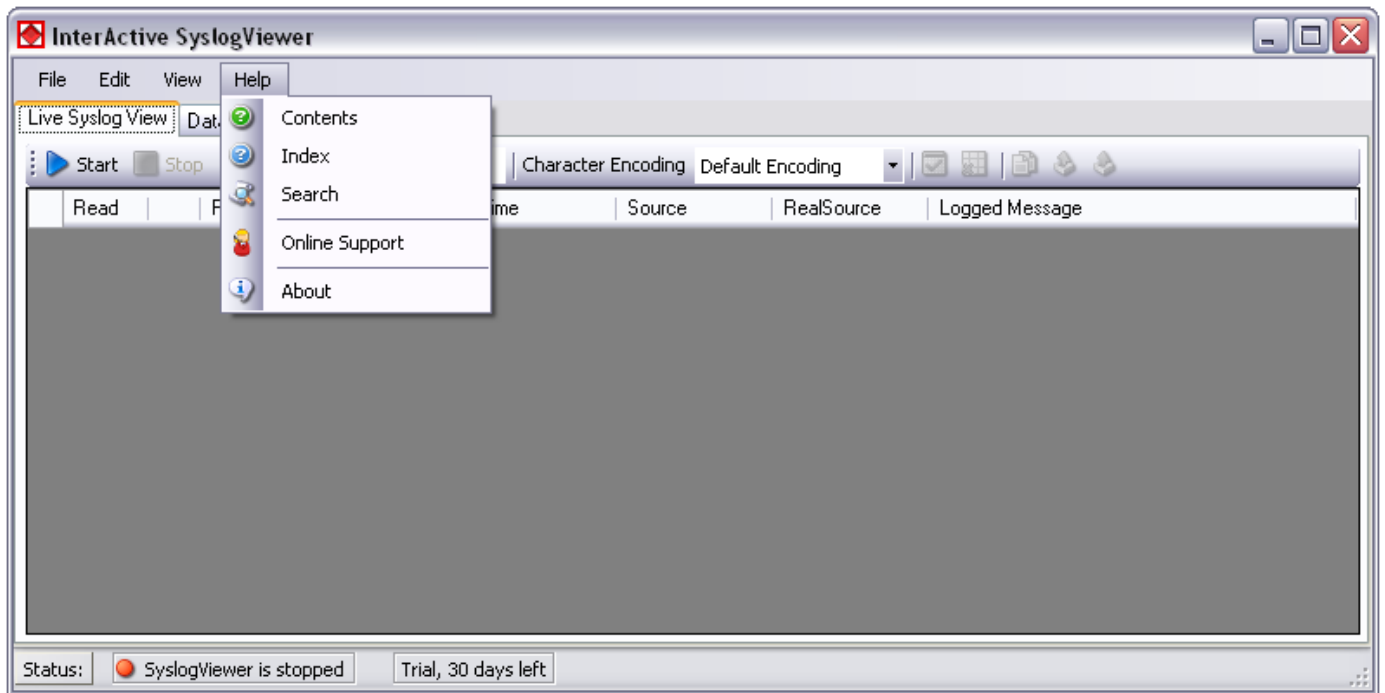


- InterActive SyslogViewer Syslog - View Menu*

Always on Top

This option is very self-explanatory. While activated, the InterActive SyslogViewer window will stay on top of all other applications, so you will have all incoming log data directly in your point of view.

Help Menu



- InterActive SyslogViewer Syslog - Help Menu*

Contents

Show the manual.

Index

Show the manual index.

Search

Search the manual.

Online Support

By clicking here, a browser window will open and you will be directed to our support website.

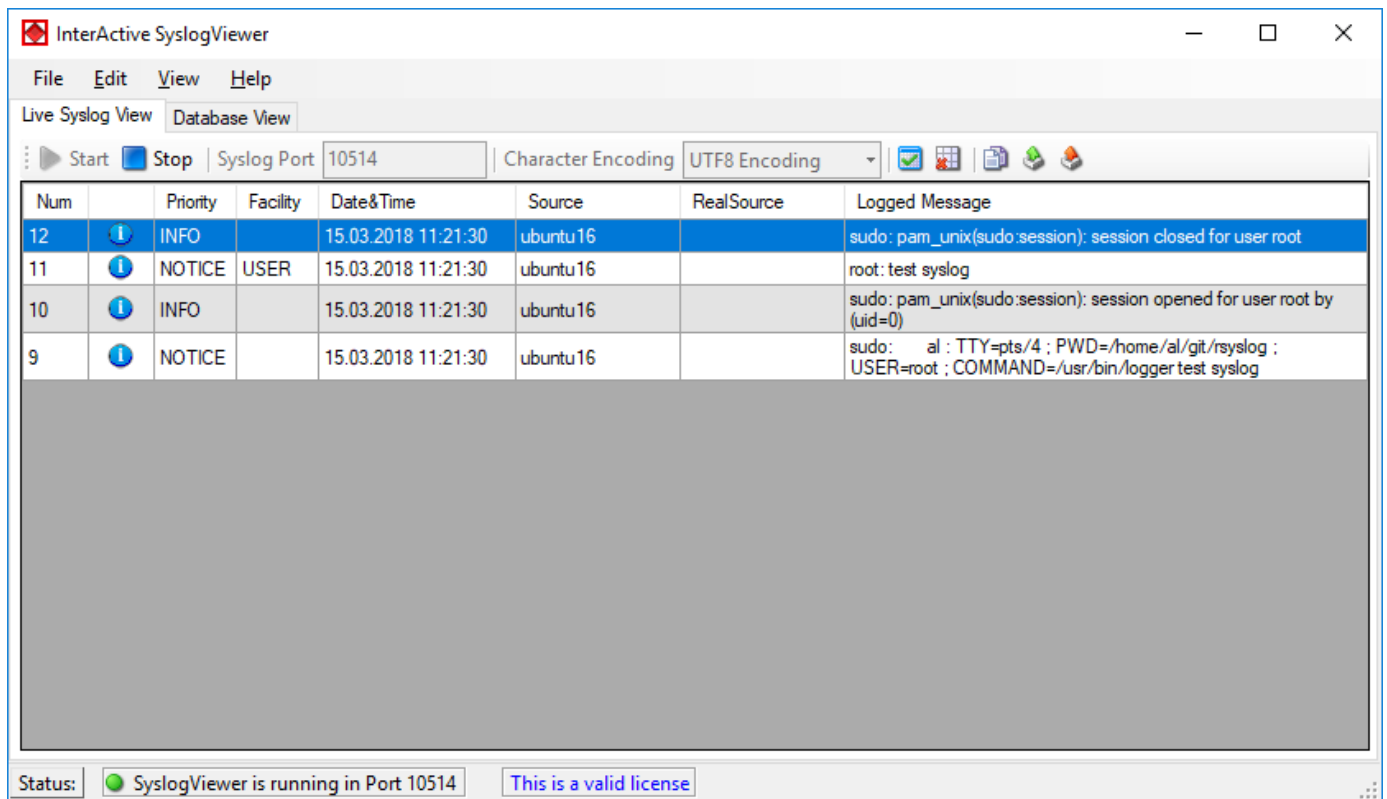
About

The About-window will give you additional information to the tool, like the program version.

Live Syslog View

Primarily, the InterActive SyslogViewer is used for viewing current syslog traffic. All messages are shown in a list with the most important information. These are the Priority, Facility, Date&Time, Source, RealSource and the Message. At the beginning of each line you can see the number of the logged event and a checkbox, for you to track if a message has been read.

The status bar at the bottom of the screen shows you, if the SyslogViewer is running or stopped and how much time you have left for the trial or your licensing status.



- InterActive SyslogViewer Syslog - Live Syslog View*

The toolbar provides you with direct access to the most important functions. These are described here:

Start

With the start button, you start the receiving service. Now the InterActive SyslogViewer will receive and display all incoming messages. If messages were sent before starting the service, they will be dropped.

Stop

Here you can stop the receiving server.

Syslog Port

Here you can define the syslog port where the Viewer should receive the syslog messages.

Character Encoding

Here you can define how characters will be decoded. You can choose from Default Encoding (depending on OS), ASCII, Unicode, UTF8, or UTF32.

Clear checked

With this button, you can clear all the checkboxes in front of the messages.

Clear View

By clicking on this button, all data will be deleted from your data grid.

Copy Selection

This helps you copying the selected messages.

Export Selection

You can export the selected data directly by using this button.

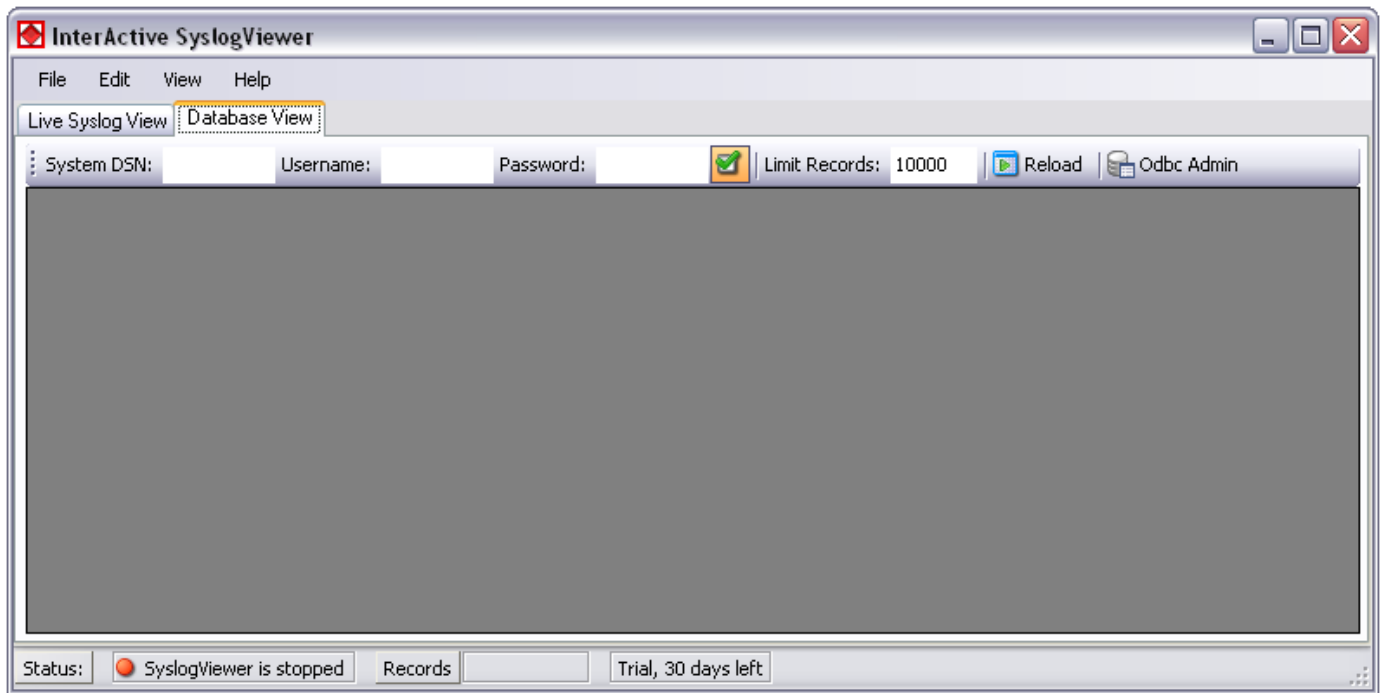
Export All

Export the complete data that is in the data grid.

Database View

Another feature is the possibility to review log messages which are stored in a database.

The status bar at the bottom of the screen shows you, if the SyslogViewer is running or stopped, how many records are currently shown and how much time you have left for the trial or your licensing status.



- InterActive SyslogViewer Syslog - Database View*

The toolbar in this case is for entering the login information for the database.

System DSN

Specify the System DSN of your database here.

Username

The username for the database.

Password

The appropriate password for the database.

Store Username and Password

With the checkbox you can tell the InterActive SyslogViewer to keep the username and password or not. This is to make usage easier for you.

Limit Records

This limits the maximum of the shown records. The default value is 10000. If changed, this can have a enormous impact on your machine.

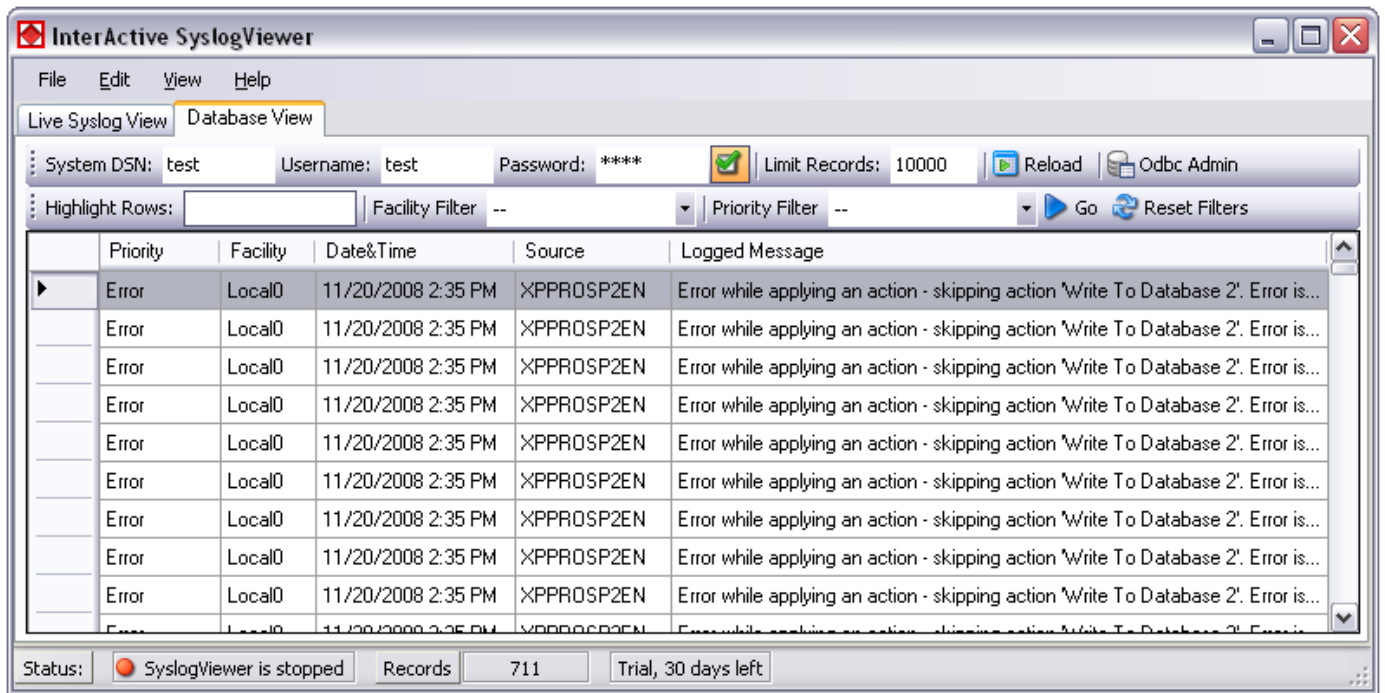
Reload

This button is to reload the database. This is needed to view if there are new log messages in the database.

Odbc Admin

This button opens the Administration Panel for ODBC Data Source connections

Once a database connection is successfully established, you can see another toolbar with the filter options:



- InterActive SyslogViewer Syslog - Active Database View*

Highlight Rows

You can enter a keyword into the field, the rows containing this keyword will be highlighted. You can then find the messages much easier,

Facility Filter

Allows you to only show messages with a certain facility. You can use the drop-down menu to specify the facility.

Priority Filter

Allows you to only show messages with a certain priority. You can use the drop-down menu to specify the priority.

Go

With this button, you apply the filter settings to the current view. Depending on the filter settings you chose you will see either colored lines and/or only the lines from the category you wish to see.

Reset Filters

Resets the filter settings and returns you to the default view of your database.

Configuring

WinSyslog is easy to use and is powerful.

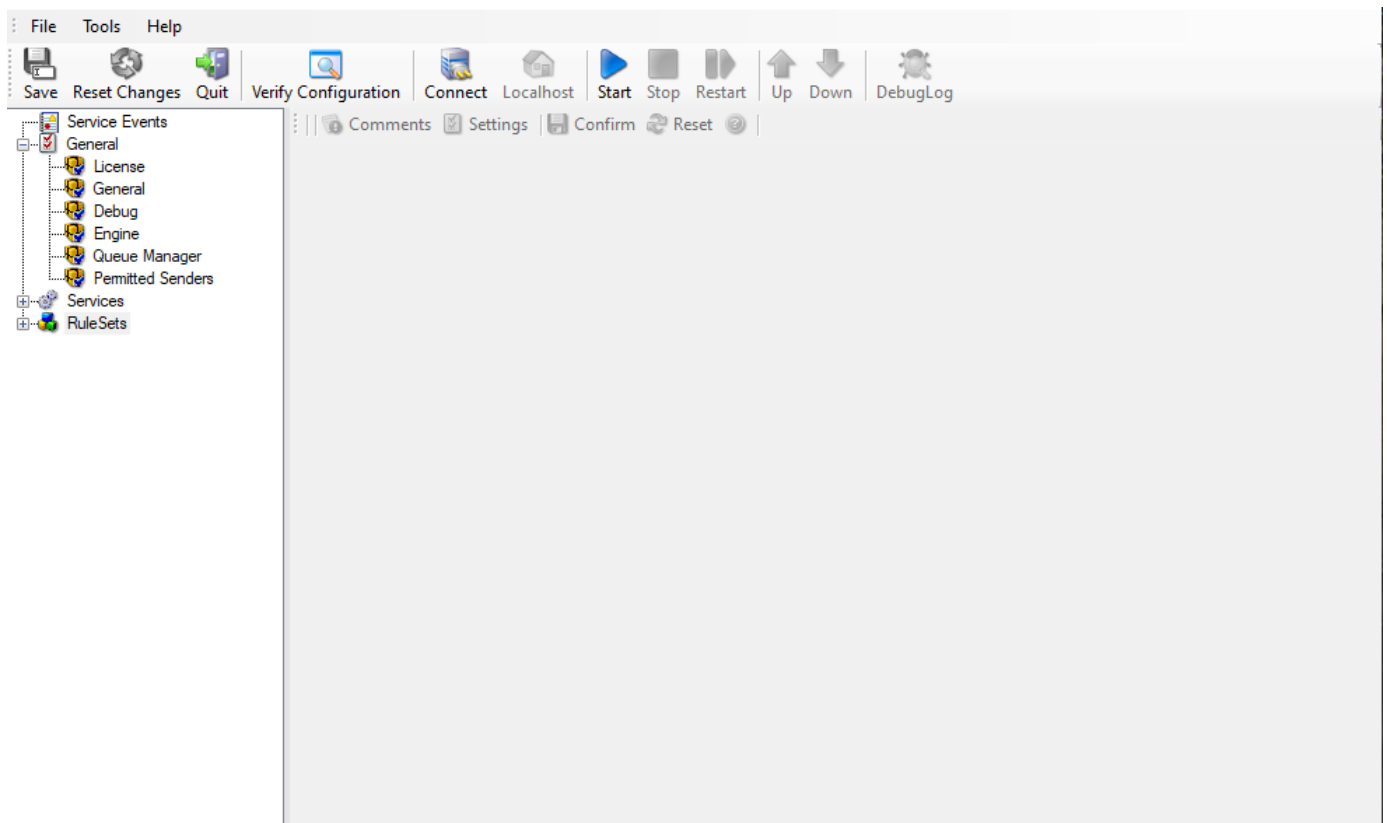
In this chapter, you will learn how to configure the EventReporter Service.

Configuring WinSyslog

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:



- Configuration Client*

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

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

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

The tree view's Running Services area lists all configured services as well as their parameters. There is exactly one service entry for each service created. Please note that there can be as many instances of a specific service type as your application requires. Typically, there can be multiple instances of the same service running, as long as their configuration parameters do not conflict. For example the syslog service: there can be multiple syslog servers on a given system as long as they listen to different ports. Consequently, there can be multiple instances of the syslog service be created. For example, there could be three of them: two listen to the default port of 514, but one with TCP and one with UDP, and a third one listens to UDP, port 10514. All three coexist and run at the same time. If these

three services are listening to the same port then an error message is logged into Windows Event Log that more than one instance of Syslog server is running. After which WinSyslog wouldn't be able to perform the desired action.

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

The 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 not run, but the configuration data can still be present. That way, it is easy to temporarily disable a service without deleting it.

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

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

The tree view's last main element is RuleSets. Here, all rulesets are configured. Directly beneath "Rules" are the individual rulesets. 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).

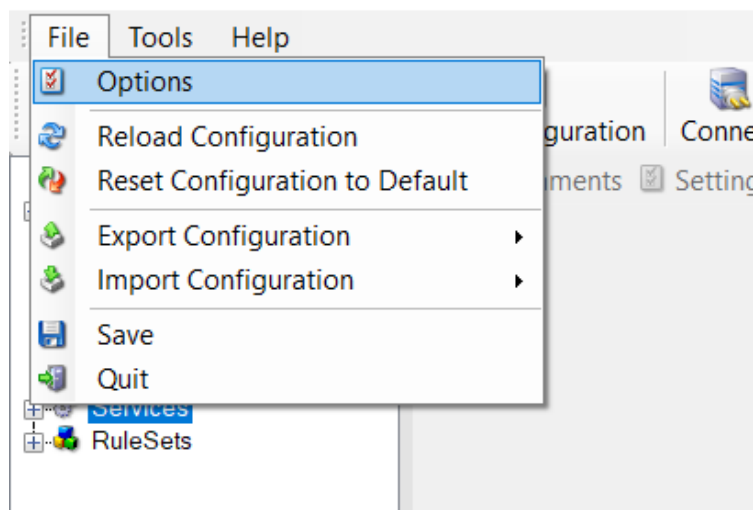
Beneath each ruleset are the individual rules. As described in Rules , a rule's position in the list is vitally important. rules at the top of the rule set are executed before those further down. To move a rule up or down, simply right click it and select "move up" or "move down" from the pop up menu.

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.

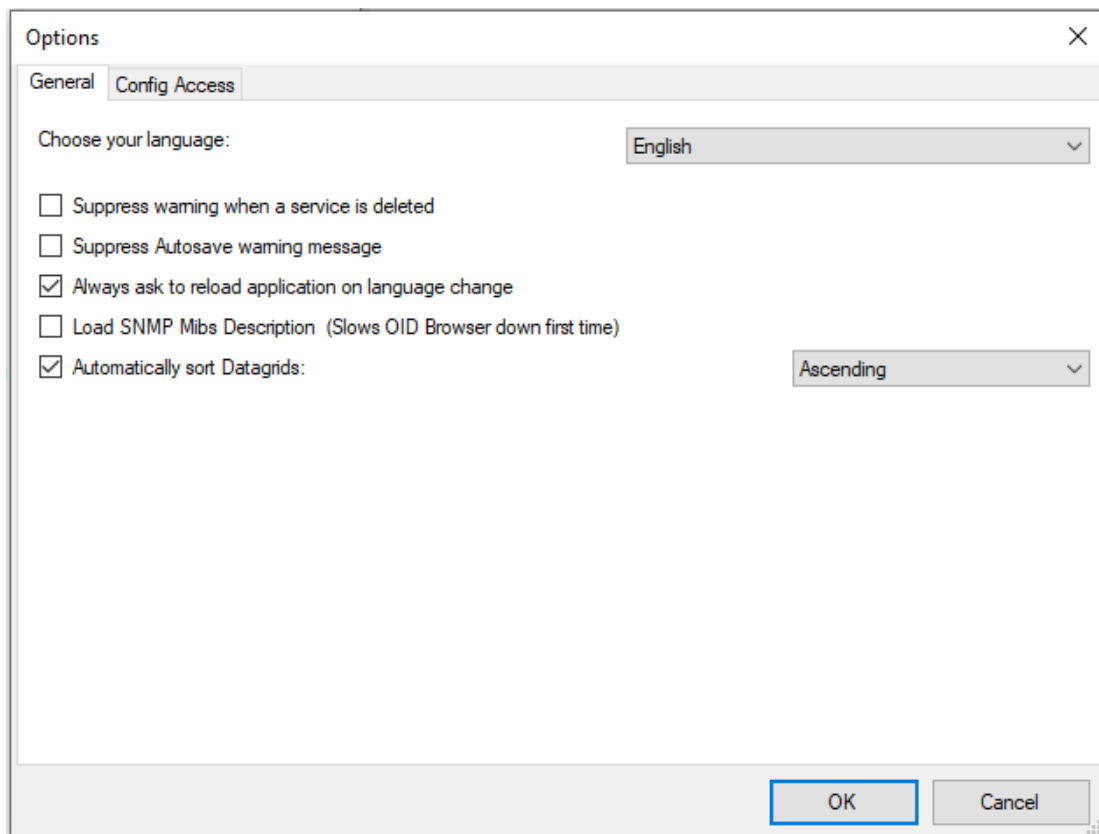
The following sections describe each element's properties.

Client Options

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



- Client Options*



- General Tab*

Choose your language

You can choose a language pack. "English" is the default and suggested language.

Suppress warning when a service is deleted

If this option is checked you will not get a warning when you try to delete a service and there is no other service that uses the connected ruleset.

Suppress autosave warning message

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

Always ask to reload application after language change

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

Load SNMP Mibs Description (Slows OID Browser down first time)

If enabled, load SNMP Descriptions from MIB files (Client starts a little slower on startup).

Automatically sort Datagrids

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

Options

General Config Access

☐ Load Configuration from Registry

Root Registry Key: SOFTWARE\Adiscon\MonitorWare

Base Registry Key: SOFTWARE\Adiscon\MonitorWare\Agent

Data Registry Key: SOFTWARE\Adiscon\MonitorWare\AgentData

☒ Load Configuration from File

Configuration Filename: C:\Program Files (x86)\MonitorWare\Agent\mwagent.cfg Browse

Data Directory: C:\Program Files (x86)\MonitorWare\Agent\ Browse

☒ Create individual configuration files for Services

☒ Create individual configuration files for RuleSets

☐ Automatically Reload Configuration from URL (https required)

Validate

Check interval: 5 Minutes

OK Cancel

- Config Access Tab*

Load Configuration from Registry

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

Load Configuration from File

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

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

Create individual configuration files for Services

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

Create individual configuration files for RuleSets

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

Automatically Reload Configuration from URL (https required)

Only possible if File Configuration Mode is used.

If enabled, the configuration will be reloaded from a remote https location. Please note that a valid SSL certificate is required, or if custom certificates are used they have to be imported on the local machine properly.

If the remote configuration file can be downloaded from the configured location and differs from the current configuration, it will be installed automatically and the service will reload itself.

Check interval

Specifies how often the service will check for remote configuration files. Please keep in mind that the configuration needs to be downloaded each time from the remote https url for comparison with the local one. We do not recommend to use a value lower than 5 minutes.

Client Tools

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

Syslog Test Message

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

- Syslog Test Message Connection properties - UDP*

Syslog server

The hostname or ip address of the target Syslog server.

Syslog Port

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

Repeat Message

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

Sleeptime between sending

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

Append Number to Syslog Message

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

Network Protocol

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

Send Syslog Test Message

Connection properties | Message properties

Syslog Server: 172.17.0.216
 Syslog Port: 514
 Network Protocol: TCP

Repeat Message: 1 times
 Sleeptime between sending: 5 milliseconds
☐ Append Number to Syslog Message

TCP related Options
 Message Delimiter: \n
☒ Enable SSL / TLS Encryption.

TLS related Options
 Common CA: [Browse]
 Certificate: [Browse]
 Key: [Browse]

Send Cancel Close

Debug Output

- Syslog Test Message Connection properties - TCP*

Message Delimiter (TCP related Options)

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

Enable SSL/TLS Encryption (TCP related Options)

Check this option to enable the TLS related Options.

TLS related Options (TCP related Options)

Select common CA: Select the certificate from the common Certificate Authority (CA), the syslog receiver should use the same CA.

Select Certificate: Select the client certificate (PEM Format).

Select Key: Select the keyfile for the client certificate (PEM Format).

- Syslog Test Message Message properties*

Load RAW Syslogdata from File

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

Configure Syslog message with these properties

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

Send one Message per LineFeed

Check if your syslogdata contains multiple syslog messages divided by line feeds

Output encoding

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

Passive Syslog Receiver

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

Passive Syslog Receiver

Test PassiveSyslog Service

Syslog Server: 192.168.214.1

Syslog Port: 514

Message Delimiter: \n

☐ Send this Message after Connect

☐ Expect this Message after Connect

Retrieve Messages Cancel Close

Priority	Facility	Date&Time	Source	Logged Message

- Test Passive Syslog Service*

Syslog server

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

Syslog Port

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

Message Delimiter

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

Send this Message after Connect

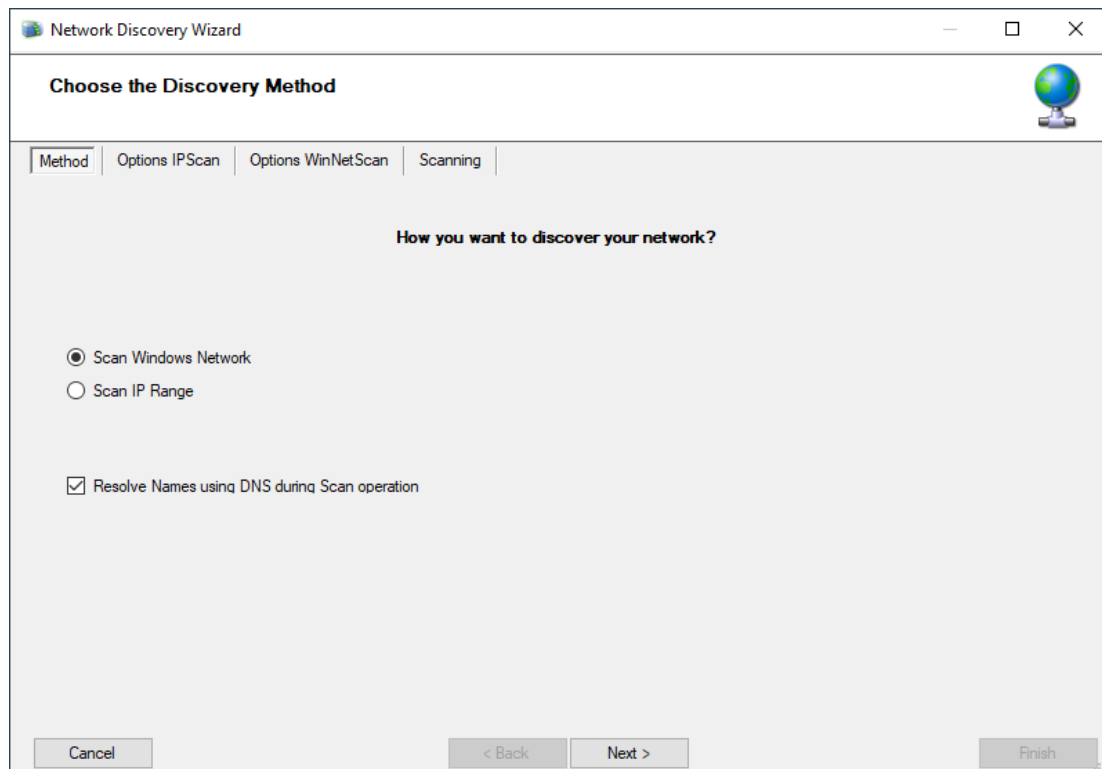
If required, configure a custom message that is send to the server after connect.

Expect this Message after Connect

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

Network Discovery

Opens up a Wizard that will help you discover devices in your local network. Once the wizard has scanned your network, it will show Windows compatible devices it has found. Please note that this will require Windows Management Instrumentation (WMI) access to the remote machines which may be disabled in Windows Firewalls by default.



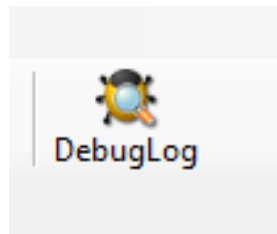
- Network Discovery - Choose the discovery Method*

Kill Service

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

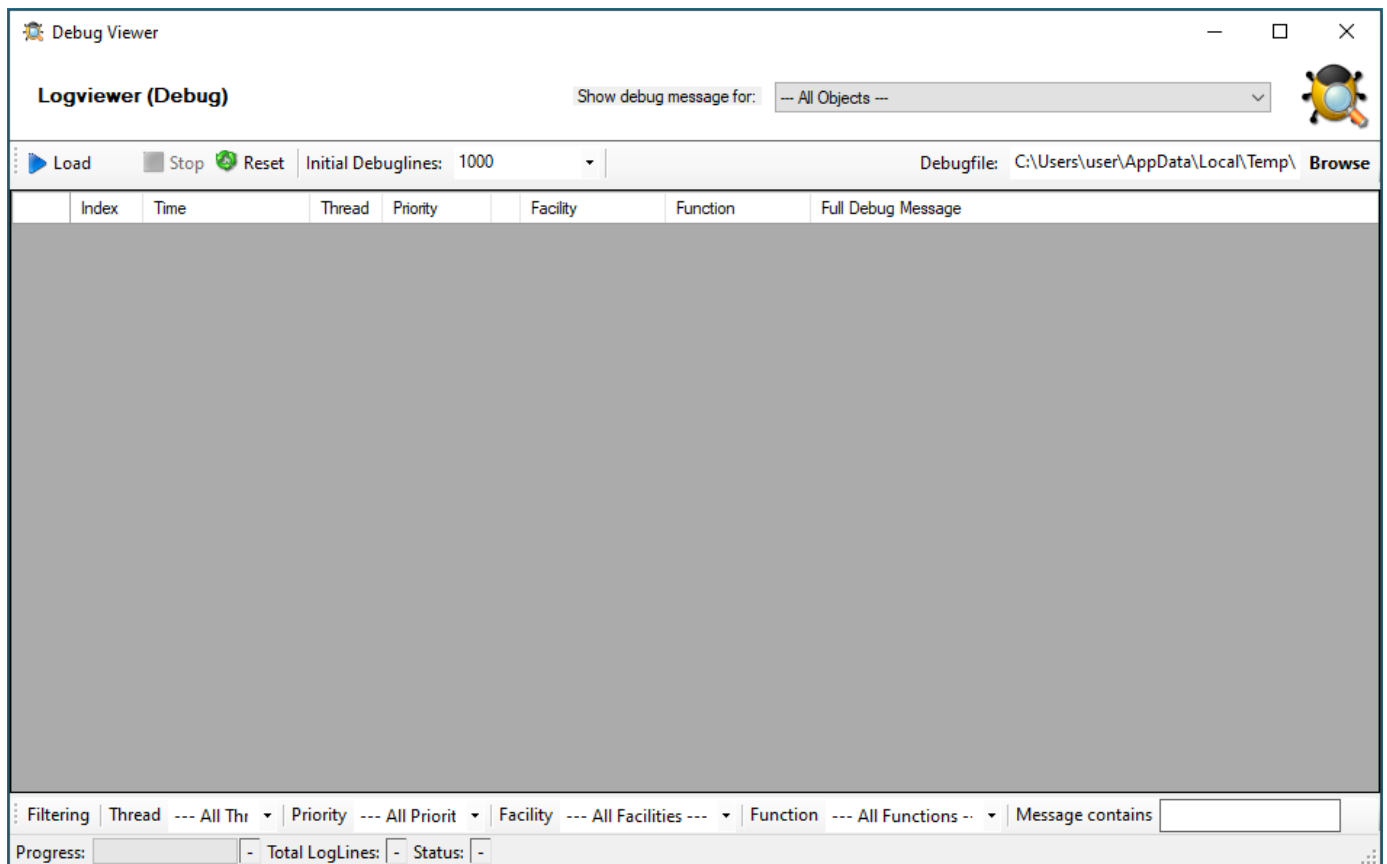
DebugLog

The **DebugLog** Button will be available if Debug Logging is enabled in your Debug Options



- DebugLog*

When clicked, a new Logviewer window will be opened. The Debug Logviewer can load, parse, and analyze debug log-files from the service.



- Logviewer (Debug)*

Debugfile

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

Load

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

Stop

Stop continuous loading of the debug log.

Reset

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

Init Debuglines

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

Show debug messages for

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

Filtering (bottom bar)

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

Using File based configuration

Working with File based Configurations

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

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

Adiscon Configuration format explained

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

Variables

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

Comments

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

File Includes

Sample

```
includeconfig my-subconfigfiles-*.cfg*
```

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

General Options

Sample

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

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

Services

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

Sample Service configuration:

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

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

```
1      = Syslog  
2      = Heartbeat  
3      = EventLog Monitor V1 (Win 2000 / XP / 2003 )  
4      = SNMP Trap Listener  
5      = File Monitor  
8      = Ping Probe  
9      = Port Probe  
10     = NTService Monitor  
11     = Diskspace Monitor  
12     = Database Monitor  
13     = Serialport Monitor  
14     = CPU Monitor  
16     = MonitorWare Echo Request
```

```

17      = SMTP Probe
18      = FTP Probe
19      = POP3 Probe
20      = IMAP Probe
21      = IMAP Probe
22      = NNTP Probe
23      = EventLog Monitor V2 (Win VISTA/7/2008 or higher)
24      = SMTP Listener
25      = SNMP Monitor
26      = RELP Listener
27      = Passive Syslog Listener
1999998 = MonitorWare Echo Reply
1999999 = SETP Listener

```

RuleSets

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

Sample

```

ruleset(name="[name]" expanded="[on/off]") {
  rule(name="[name]" expanded="[on/off]" actionexpanded="[on/off]"
    ThreatNotFoundFilters="[on/off]" GlobalCondProperty="[on/off]"
    GlobalCondPropertyString="" ProcessRuleMode="[0/1/2]"
    ProcessRuleDate="[uxtimestamp]") {
    action(type="[ID]" name="[name]") {
      $var1 Value1
      $var2 Value2
      ...
    }
    filter(nTabSelection="0") {
      $nOperationType AND
      $PropertyType NOTNEEDED
      $PropertyValuePropertyType NOTNEEDED
      $CompareOperation EQUAL
      $nOptionalValue 0
      $nSaveIntoProperty 0
      $szSaveIntoPropertyName FilterMatch
    }
  }
}

```

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

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

```

1000 = ODBC Database
1001 = Send Syslog
1008 = Net Send
1009 = Start Program
1011 = Send SETP
1012 = Set Property
1013 = Set Status
1014 = Call RuleSet
1015 = Post Process
1016 = Play Sound
1017 = Send to Communication Port

```

1021 = Send SNMP
 1022 = Control NT Service
 1023 = Compute Status Variable
 1024 = HTTP Request
 1025 = OleDb Database
 1026 = Resolve Hostname
 1027 = Send RELP
 1028 = Send MS Queue
 1029 = Normalize Event
 1030 = Syslog Queue

How to enable file based configuration?

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

The screenshot shows the 'Options' dialog box with the 'Config Access' tab selected. The 'Load Configuration from File' radio button is chosen. The 'Configuration Filename' field contains 'C:\Program Files (x86)\MonitorWare\Agent\mwagent.cfg' and the 'Data Directory' field contains 'C:\Program Files (x86)\MonitorWare\Agent\'. Both fields have 'Browse' buttons next to them. The 'Create individual configuration files for Services' and 'Create individual configuration files for RuleSets' checkboxes are checked. The 'Automatically Reload Configuration from URL (https required)' checkbox is unchecked. There is a 'Validate' button next to an empty text field. The 'Check interval' is set to '5 Minutes' with a dropdown arrow. At the bottom are 'OK' and 'Cancel' buttons.

- Client Options Configure File Based Configuration*

Create individual configuration files for Services

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

Create individual configuration files for RuleSets

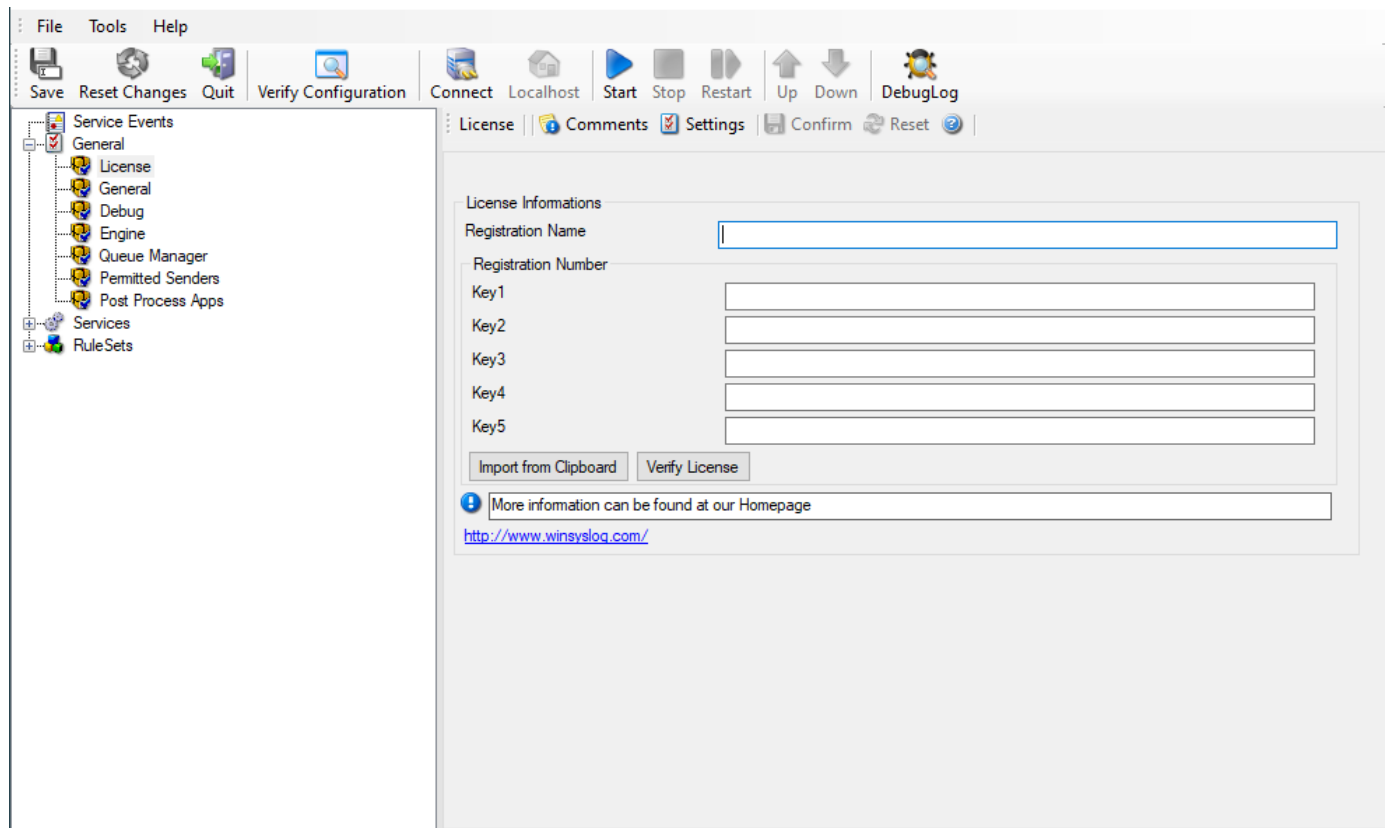
When enabled, the configuration client will create separated configuration files for each configured ruleset. The main configuration file will then use the includeconfig statement to include all these configuration files by using a pattern. When deleting a ruleset, its configuration file will be deleted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

General Options

In this chapter, you find the general option settings.

License

After the purchase, the licensing information can be entered here.



Registration Name

File Configuration field:

szlicense

Description

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

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

Registration number

File Configuration field:

nLicenseKey1, nLicenseKey2, nLicenseKey3, nLicenseKey4, nLicenseKey5

Description

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

Import from Clipboard

If the key has been copied to the clipboard it can be imported with this button.

Verify License

Here it can be verified if the license is valid.

General

The General Options available on this form are explained below:

Process Priority

File Configuration field:

nProcessPriority

Description

Configurable Process Priority to fine-tune application behavior.

Queue Limit

File Configuration field:

nQueueLimit

Description

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

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

SystemID

File Configuration field:

nSystemID

Description

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

CustomerID

File Configuration field:

nCustomerID

Description

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

Location of your SNMP MIBs

File Configuration field:

szMIBSPath

Description

Click the Browse button to search for your MIBs location or enter the path manually. The Client and Service will read all files from this directory automatically on startup.

Default Timevalues are based on

File Configuration field:

nTimeMode

Description

The general options of each product (EventReporter, MonitorWare Agent and WinSyslog) contain a setting for the "Default Timevalues are based on". This setting can be set to Localtime and UTC (Universal Coordinated Time) which is default. This setting has an effect on:

- Send Email Action: The date in the email header is affected
- Start Program Action: Time parameters in the command line are affected
- Write File Action: Time properties in the file name are affected
- Filter Engine: If you filter by weekday or time fields, localtime does affect the filter result

For information about "How can I get localtime output" please see default timevalues setting in EventReporter/MonitorWare Agent/WinSyslog explained.

Protect Service against shutdown

File Configuration field:

nProtectAgainstShutdown

Description

When enabled, the Agent will not stop processing the internal queue when it is stopped. **Please note that it will remain in the stopping state then.**

Log Warnings into the Windows Application Eventlog

File Configuration field:

nEnableEventlogWarnings

Description

The Service will also log Warnings into the Windows Application Eventlog, and so be more verbose for troubleshooting. Default is disabled.

Special Unicoder Conversion for Japanese Systems

File Configuration field:

nJapanStringHandling

Description

This is a historical option for older multibyte systems from the time when UTF8 was not known yet. If enabled, whenever text is being converted from 16 Bit wide character to 8 Bit character, the conversion is done with bit masking in order to avoid broken encoding. **For today modern systems, we do NOT recommend to enable this option.**

Automatically reload service on configuration changes

File Configuration field:

nEnableAutoConfigReload

Description

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

Enable random wait time delay when checking for new configurations

File Configuration field:

bAutoReloadRandomDelay

Description

When enabled, a random delay (with the configured maximum) will be added between new configuration checks.

Maximum random delay time

File Configuration field:

nAutoReloadDelayTime

Description

The maximum for this random delay is 24 hours. The random delay has no affect on the service control anymore.

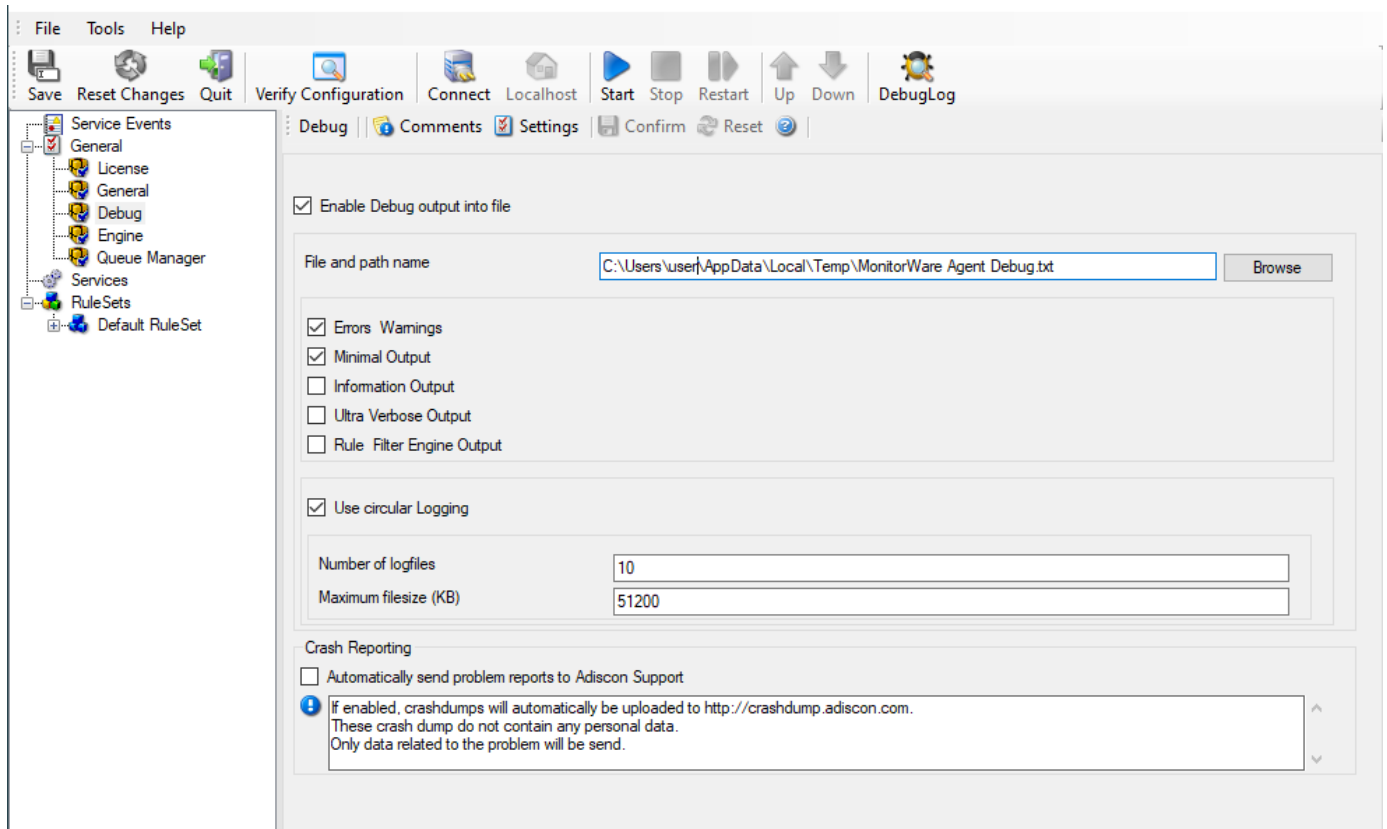
Debug

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

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

Note

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



Enable Debug output into file

File Configuration field:

nEnableDebugOutput

Description

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

File and path name

File Configuration field:

szDebugFileName

Description

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

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

Debug Levels

File Configuration field:

nDebugErrors, nDebugMini, nDebugInternal, nDebugUltra, nDebugRuleEngine

Description

These checkboxes control the amount of debug information being written. We highly recommend only selecting "Errors & Warnings" as well as "Minimum Debug Output" unless otherwise instructed by Adiscon support.

Use circular Logging

File Configuration field:

nCircularLogging

Description

Support for circular debug logging has been added as the debuglog can increase and increase over time. This will avoid an accidental overload of the hard disk. Of course you can also customize the amount of files used and their size or disable this feature.

Automatically send problem reports to Adiscon Support**File Configuration field:**

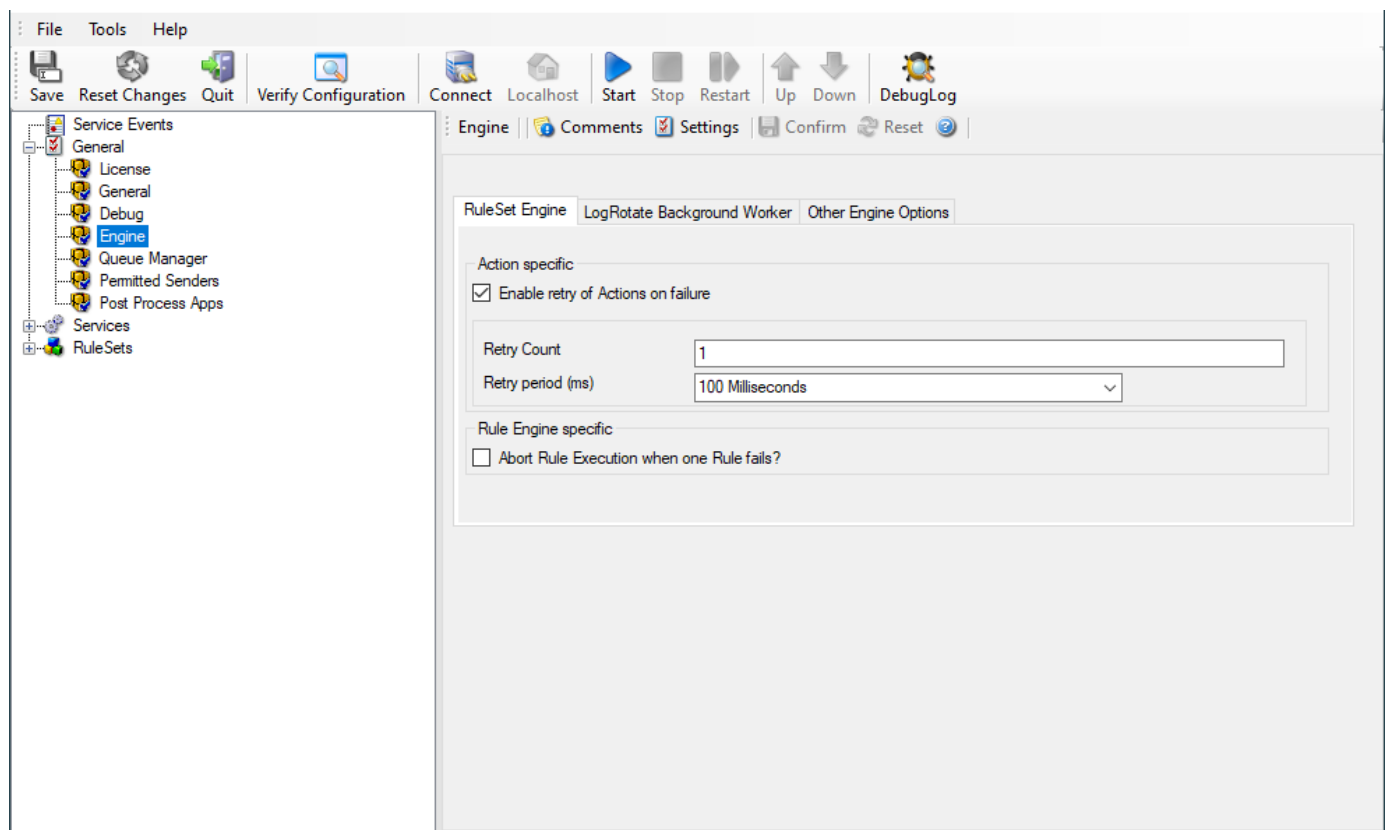
nReportCrash

Description

If enabled, problem reports will automatically be uploaded to <http://crashdump.adiscon.com>. A problem report is generated if the service internally stops working for some unknown reason. The reports are small dumpfiles which do not contain any personal data and will help us find and fix the problem. Also the dumpfiles are very small and do not exceed 256 Kbyte. In most cases only 32Kbyte data is send.

Engine

The Engine specific Options are explained below:



- RuleSet Engine Tab*

Action specific**Enable retry of Actions on failure****File Configuration field:**

nEnableRetry

Description

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

Rule Engine specific

Abort Rule Execution when one Rule fails?

File Configuration field:

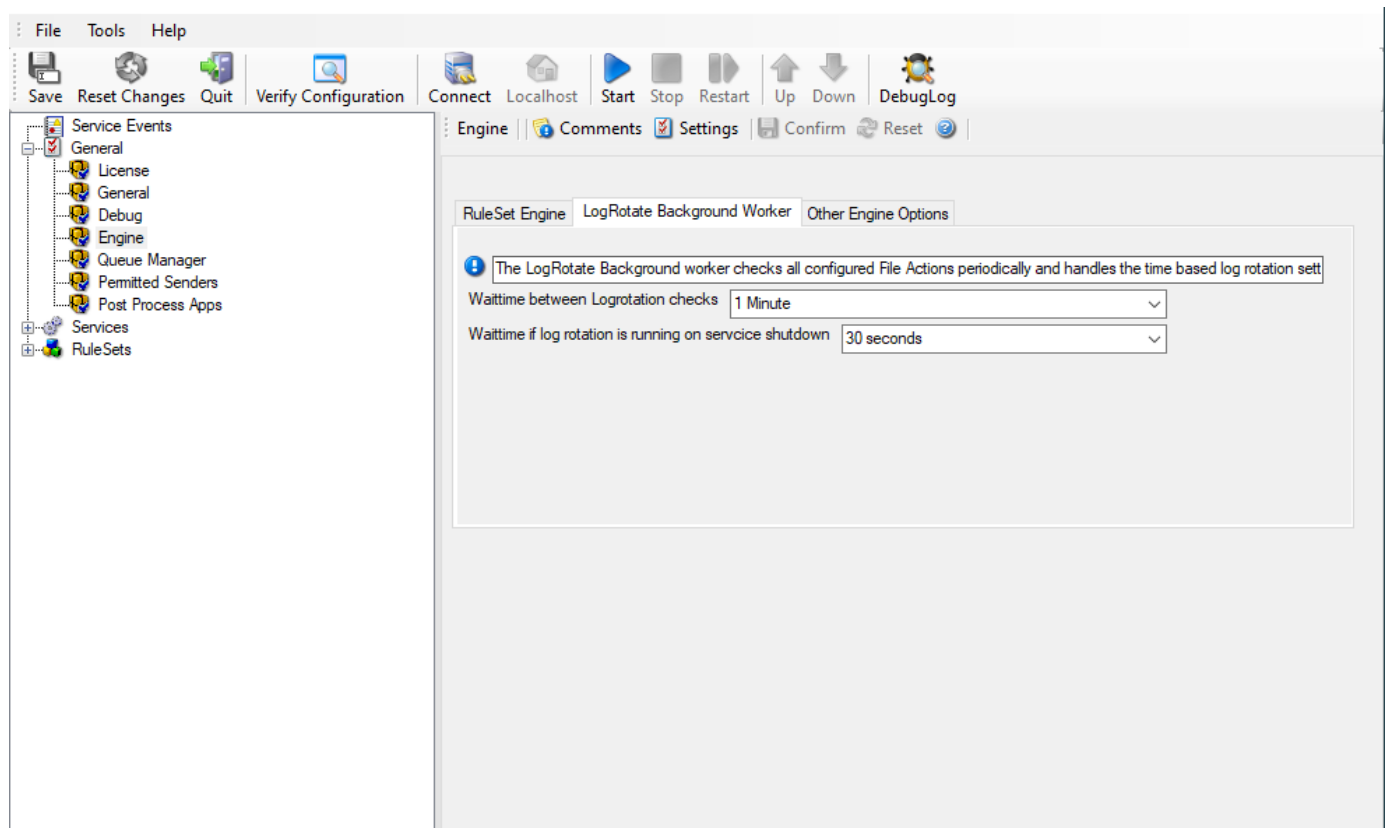
bAbortRuleOnFailure

Description

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

LogRotate Background Worker

The LogRotate Background worker checks all configured File Actions periodically and handles the time based log rotation settings, if enabled.



- LogRotate Background Worker Tab*

Wait time between Logrotation checks

File Configuration field:

nLogRotateWorkerSleepTime

Description

Defines how often the logrotate background worker thread checks all configured actions to see if any logfiles need to be rotated based on time related rotate conditions.

Wait time if log rotation is running on service shutdown

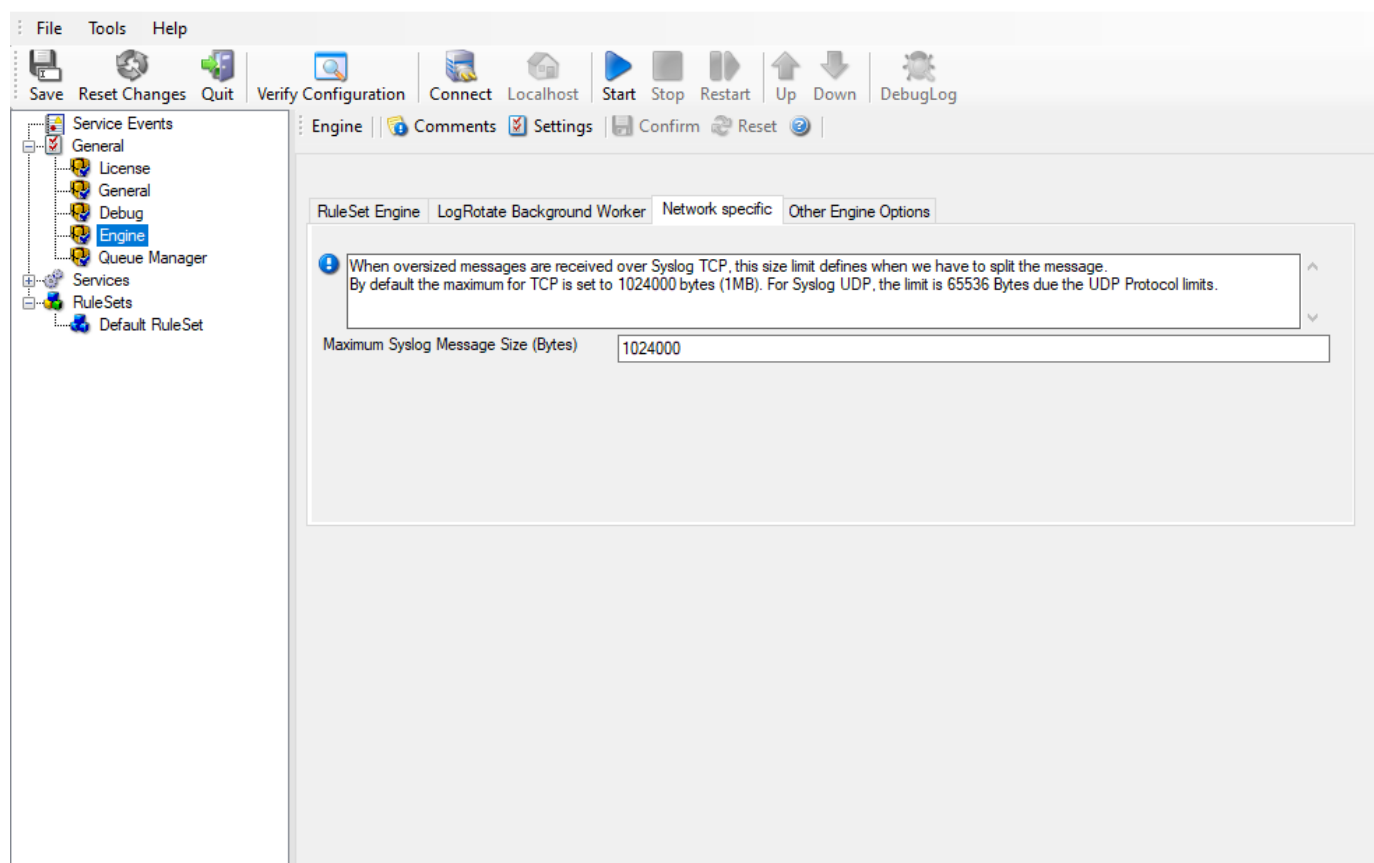
File Configuration field:

nLogRotateWorkerStopWaitTimeout

Description

When service is being shutdown, this defines how much time the logrotate background worker thread has left to finish its log rotations before a forceful termination.

Network specific Options



- Network specific Options Tab*

Maximum Syslog Message Size (Bytes)

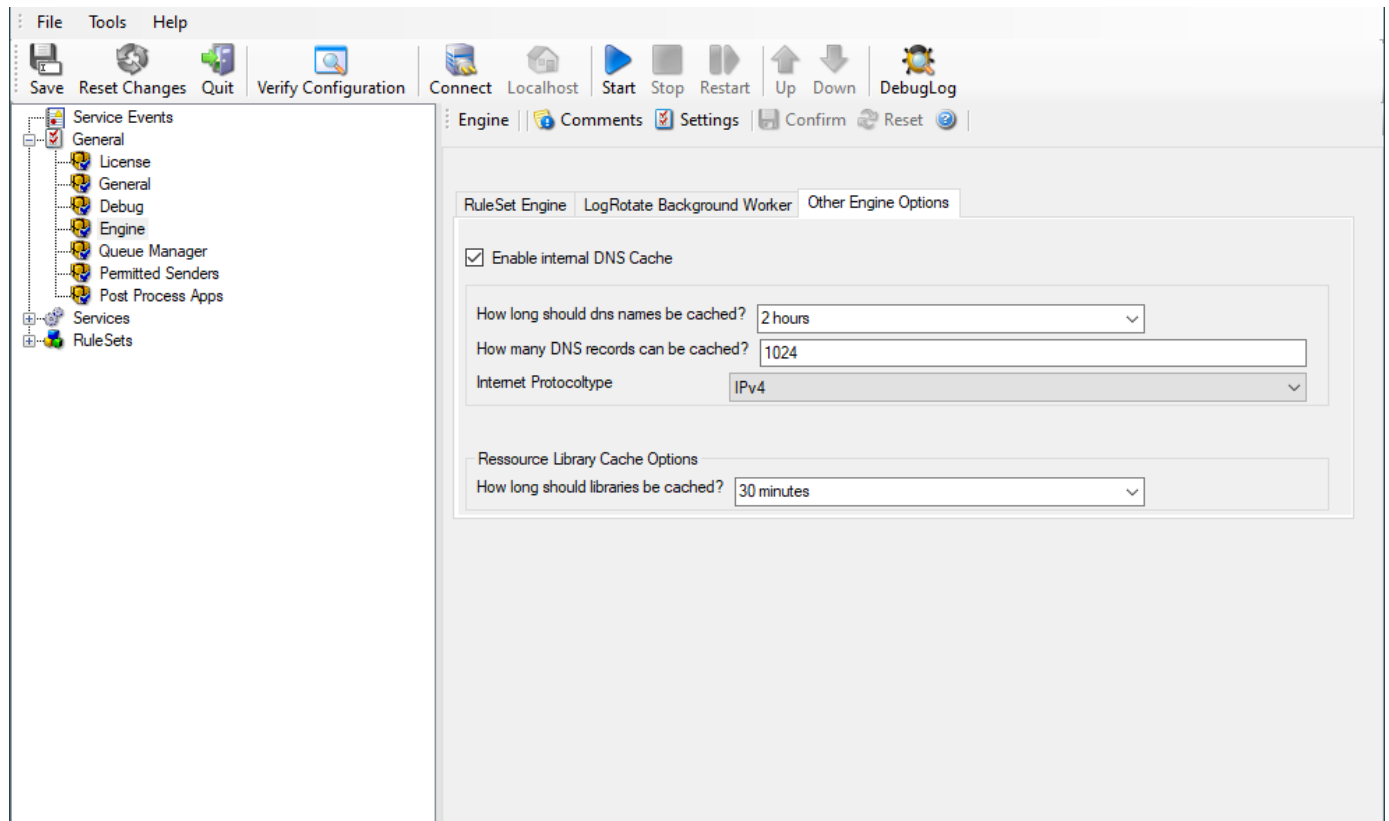
File Configuration field:

nSyslogMaxMessageSize

Description:

Configurable message size limit for Syslog TCP messages. The default is 1MB which is far more as defined in Syslog RFC's. If a syslog message exceeds the size limit, it will be split into multiple messages.

Other Engine Options



- Other Engine Options Tab*

Enable internal DNS Cache

File Configuration field:

nEnableDNSCache

Description

The DNS cache is used for reverse DNS lookups. A reverse lookup is used to translate an IP address into a computer name. This can be done via the resolve hostname action. For each lookup, DNS needs to be queried. This operation is somewhat costly (in terms of performance). Thus, lookup results are cached. Whenever a lookup needs to be performed, the system first checks if the result is already in the local cache. Only if not, the actual DNS query is performed and the result then stored to the cache. This greatly speeds up reverse host name lookups.

However, computer names and IP addresses can change. If they do, the owner updates DNS to reflect the change. If we would cache entries forever, the new name would never be known (because the entry would be in the cache and thus no DNS lookup would be done). To reduce this problem, cache records expire. Once expired, the record is considered to be non-existing in the cache and thus a new lookup is done.

Also, cache records take up system memory. If you have a very large number of senders who you need to resolve, more memory than you would like could be allocated to the cache. To solve this issue, a limit on the maximum number of cache records can be set. If that limit is hit, no new cache record is allocated. Instead, the least recently used record is overwritten with the newly requested one.

How long should DNS names be cached?

File Configuration field:

nDNSCacheTime

Description

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

How many DNS records can be cached?

File Configuration field:

nDNSCacheLimit

Description

This is the maximum number of DNS records that can be cached. The system allocates only as many memory, as there are records required. So if you have a high limit but only few sending host names to resolve, the cache will remain small. However, if you have a very large number of host names to resolve, it might be useful to place an upper limit on the cache size. But this comes at the cost of more frequent DNS queries. You can calculate about 1 to 2 KBytes per cache record.

Internet Protocoltype

File Configuration field:

nDNSInetProtocol

Description

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

Resource Library Cache Options

How long should libraries be cached?

File Configuration field:

nLibCacheTimeOut

Description

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

Queue Manager

Queue Manager
Comments
Settings
Confirm
Reset

☐ Enable Queue Manager Diskcache

File and path name
C:\Program Files (x86)\MonitorWare\Agent\MWQueueBuffer.dat
Browse

Warning! If you enable diskcaching, it will slow down processing of the actions. This depends on the speed of your harrdisk. Do only enable this feature if you really want cache the queue on disk for failover reasons. If the processing is interrupted for some reason, the Service will load the queue on startup and process what was in the queue before.

Queue File Size (static)

Processing pointer
0

Saving pointer
0

Number of worker threads
2

- Queue Manager*

Enable Queue Manager DiskCache

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

Warning

Only use this feature if you really need to!

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

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

Enable Queue Manager Diskcache

File Configuration field:

nEnableRingBuffer

Description

Enable the disk based queue manager. Please read the description about the Queue Manager DiskCache first!

File and Pathname

File Configuration field:

szRingBufferFile

Description

As everywhere else, you can define here, where the queue file should be stored.

Queue File Size

File Configuration field:

nRingBufferSize

Description

With this slider, the queue size can be set from 1 MB to 2048 MB.

Processing pointer

File Configuration field:

nProcessingLow

Description

Points to the current processing position within the queue file.

Saving pointer

File Configuration field:

nSavingLow

Description

Points to the last processed position within the queue file.

Queue Manager specific

Number of worker threads

File Configuration field:

nWorkerThreads

Description

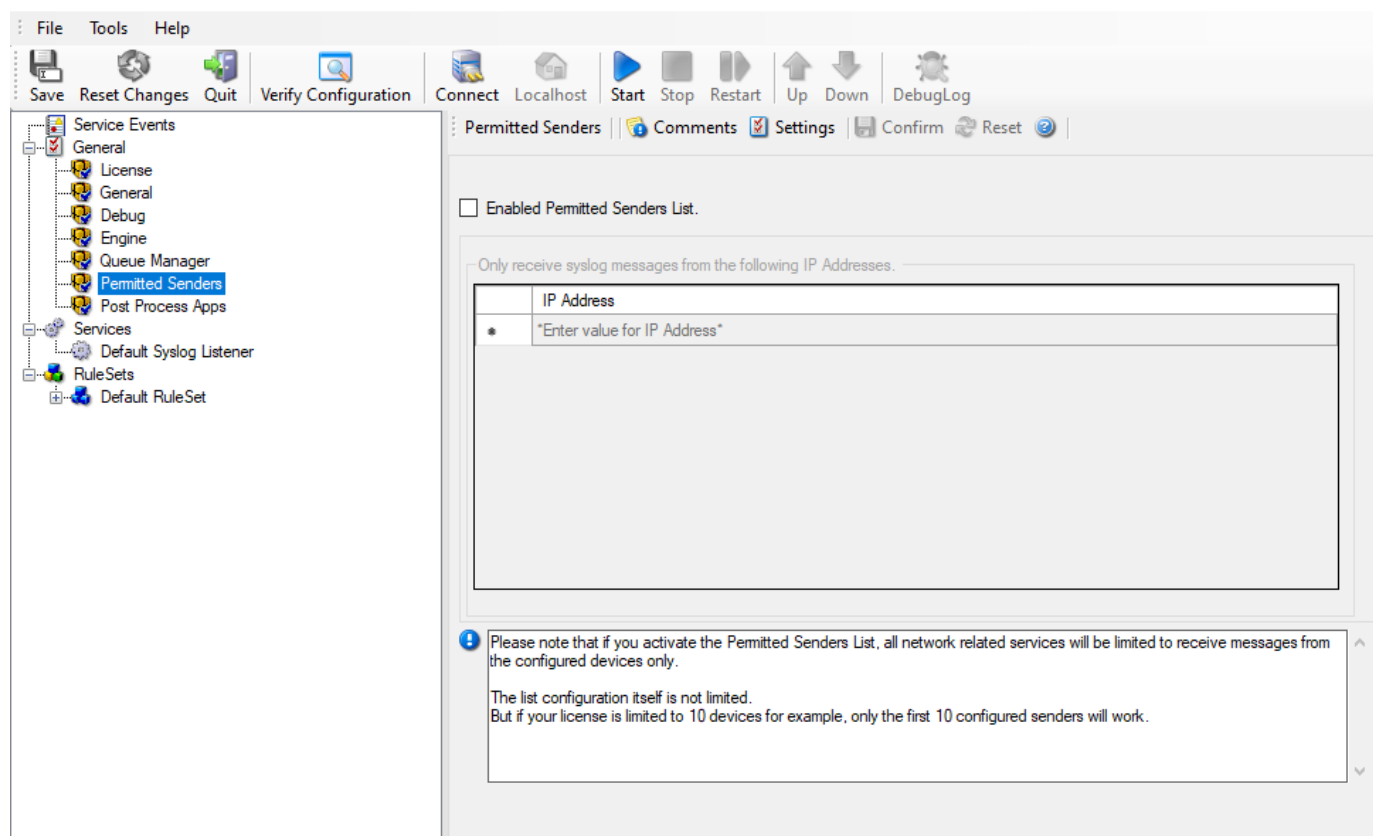
Defines the number of worker background threads that the core engine uses to process its queue.

Permitted Senders

Please note that if you activate the Permitted Senders List, all network related services will be limited to receive messages from the configured devices only.**

The list configuration itself is not limited.

But if your license is limited to 10 devices for example, only the first 10 configured senders will work.**



Enable permitted Senders List

File Configuration field:

nEnablePermittedSenders

Description

If this option is enabled, all network related services will be limited to receive messages only from the configured IP Addresses. Please note the list is also limited to your license limit. For example if your license allows 10 devices, only the first 10 configured senders will be allowed.

Only receive syslog messages from the following IP Addresses

File Configuration field:

szIP_[n]

Description

This list contains all sender IP Addresses which are allowed to send data to network related services. You can either configure IPv4 or IPv6 Addresses here.

Services

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

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

Added a test mode for Services, currently EventLog Monitor V1 & V2 and File Monitor are supported. When enabling the testmode for a certain service, it will process it's Events/Files over and over again. So only use this setting for testing purposes.

Basic Services

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 be assumed that the sender is either in trouble or already stopped running.

- Service - Heartbeat*

Message that is send during each heartbeat

File Configuration field:

szMessage

Description:

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

Heartbeat clock (Sleeptime)

File Configuration field:

nSleepTime

Description:

This is the interval, in milliseconds, that the heartbeat service generates information units in. Please note that the receiving side should be tolerant. The interval specified here is the minimum time between packets. Under

heavy load, the interval might be slightly longer. It is good practice to allow twice this interval before the service is considered suspect by the system monitoring the services health.

General Values (Common settings for most services)

Syslog Facility

File Configuration field:

nSyslogFacility

Description:

The syslog facility to be assigned to events created by this service. Most useful if the message is to forward to a Syslog server.

Syslog Priority

File Configuration field:

nSyslogPriority

Description:

The Syslog priority to be assigned to events created by this service. Most useful if the message is to forward to a Syslog server.

Syslog Tag Value

File Configuration field:

szSyslogTagValue

Description:

The Syslog tag value to be assigned to events created by this service. Most useful if the message is to forward to a Syslog server.

Resource ID

File Configuration field:

szResource

Description:

The resource id to be assigned to events created by this service. Most useful if the message is to forward to a Syslog server.

RuleSet to Use

File Configuration field:

szRuleSetName

Description:

Name of the ruleset to be used for this service. The RuleSet name must be a valid RuleSet.

MonitorWare Echo Reply

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

Services > MonitorWare Echo Reply ✓ Enabled 🗉 Comments ⚙ Settings 📄 Confirm 🔄 Reset ?

Internet Protocoltype IPv4

IP Listener Address 127.0.0.1

Listener Port 10001

RuleSet to use Default RuleSet Refresh

- Service - MonitorWare Echo Reply*

Internet Protocoltype

File Configuration field:

nInetType

Description:

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

IP Listener Address

File Configuration field:

szMyIPAddress

Description:

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

Listener Port

File Configuration field:

nListenPort

Description:

Specify the listener port here.

RuleSet to Use

File Configuration field:

szRuleSetName

Description:

Name of the ruleset to be used for this service. The RuleSet name must be a valid RuleSet.

network services

RELP Listener

The relp listener support the new reliable event logging protocol (RELP), which enables a more reliable transmission of messages than plain tcp syslog protocol. The service permits to accept messages from senders who themselves support RELP.

Other than that it is using a different communications protocol, the RELP listener is functionally equivalent to the syslog listener. The RELP Listener will automatically listen on all available IP Addresses which includes IPv4 and IPv6. This is due the librelp implementation method.

Services > RELP Listener Enabled Comments Settings Confirm Reset ?

Internet Protocoltype IPv4

Listener Port 20514

Session Timeout 30 seconds

☐ Enable SSL / TLS Encryption.

TLS Mode Anonymous authentication

Select common CA PEM Browse

Select Certificate PEM Browse

Select Key PEM Browse

Permitted Peers

	Permitted Peename / SHA1 / etc
*	*Enter value for Permitted Peename / SHA1 / etc*

RuleSet to use Default RuleSet Refresh

- Service - RELP Listener*

Internet Protocoltype

File Configuration field:

nInetType

Description:

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

Listener Port

File Configuration field:

nListenPort

Description:

The port the RELP Listener listens on. The typical (standard) value is 20514. This should be changed only if there is a definite need for it. Such a need typically arises from security concerns. If the port is changed, all reporting devices must also be configured to use the non-standard port.

Session Timeout

File Configuration field:

nTimeOutSession

Description:

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

Enable SSL / TLS Encryption

File Configuration field:

nUseSSL

Description:

This option enables SSL / TLS encryption for your RELP Server. Please note, that with this option enabled, the server only accepts SSL / TLS enabled senders.

TLS Mode

File Configuration field:

nTLSMode

Description:

The TLS mode can be set to the following:

Anonymous authentication Default option, which means any client certificate will be accepted, or even none.

x509/name (certificate validation and name authentication) When this mode is selected, the subject within the client certificate will be checked against the permitted peers list. This means the RELP Server will only accept the secured connection if it finds the permitted peer in the subject.

509/fingerprint (certificate fingerprint authentication) This mode creates a SHA1 Fingerprint from the client certificate it receives, and compares it to fingerprints from the permitted peers list. You can use the debuglog to see fingerprints of client certificates which were not permitted.

Select common CA PEM

File Configuration field:

szTLSCAFile

Description:

Select the certificate from the common Certificate Authority (CA), the RELP receiver should use the same CA.

Select Certificate PEM

File Configuration field:

szTLSCertFile

Description:

Select the client certificate (PEM Format).

Select Key PEM

File Configuration field:

szTLSKeyFile

Description:

Select the keyfile for the client certificate (PEM Format).

Permitted Peers

Permitted Peername / SHA1 / etc

File Configuration field:

szIP_[n]

Description:

This list contains all permitted peers. If x509/name is used, this can contain parts of the client certificate subject. For example if you have CN = secure.syslog.msg in the certificate subject, you can add "secure.syslog.msg" as permitted peer. When using x509/fingerprint, this list holds a list of permitted SHA1 fingerprints. The fingerprints can either be generated with OpenSSL Tools, or grabbed from the debug logfile. The format is like described in RFC 5425, for example: SHA1 : 2C : CA : F9 : 19 : B8 : F5 : 6C : 37 : BF : 30 : 59 : 64 : D5 : 9A : 8A : B2 : 79 : 9D : 77 : A0.

RuleSet to Use

File Configuration field:

szRuleSetName

Description:

Name of the ruleset to be used for this service. The RuleSet name must be a valid RuleSet.

SETP Server

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

- Service - SETP Server*

Internet Protocoltype

File Configuration field:

nlnetType

Description:

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

Listener Port

File Configuration field:

nListenPort

Description:

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

Listener IP Address

File Configuration field:

szMyIPAddress

Description:

The SETP server service can be bound to a specific IP Address. You can either use an IPv4, an IPv6 Address, or a Hostname that resolves to an IPv4 or IPv6 Address. This feature is useful for multihome environments

where you want to run different Syslog Servers on different IP Addresses. Please note that the default IP Address 0.0.0.0 means ANY IP Address.

Session Timeout

File Configuration field:

nTimeOutSession

Description:

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

Options

Enable SSL/TLS

Note: if this Option is enabled, NON-SSL Clients will not be able to connect to this Service.

File Configuration field:

nUseSSL

Description:

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

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

Use zLib Compression to compress the data

File Configuration field:

nZlibComp

Description:

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

Notify Sender about Rule Action Errors?

File Configuration field:

bIndicateErrorToOrigin

Description:

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

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

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

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

Please Note: If you enable this feature, older MonitorWare Agent Versions (4.2.x and below, as well as WinSyslog 7.2.x and EventReporter 8.2.x and below) may have trouble sending data over SETP once a Rule

Exception occurs! If you intend to use this feature, make sure all MonitorWare Agent Installations are at least Version 4.3.x (This applies for WinSyslog 7.3.x and EventReporter 8.3.x as well).

RuleSet to use

File Configuration field:

szRuleSetName

Description:

Name of the ruleset to be used for this service. The RuleSet name can be chosen from a drop-down list where you find your RuleSets.

SNMP Trap Receiver

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.

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

- Service - SNMP Trap Receiver*

Internet Protocoltype

File Configuration field:

nInetType

Description:

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

Protocol Type

File Configuration field:

nProtocolType

Description:

You can select to listen on UDP or TCP protocol for SNMP Traps.

Listener Port**File Configuration field:**

nPort

Description:

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**File Configuration field:**

nSnmVersion

- -1 = All Supported Versions
- 0 = SNMP Version 1 only
- 1 = SNMP Version 2c only

Description:

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

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

Fully resolve MIB names (long format)**File Configuration field:**

nResolveLongMibNames

Description:

This option fully resolves the MIB names like in the client MIB browser application.

Use short format (last portion only)**File Configuration field:**

nResolveMibNamesShort

Description:

Fully resolved MIB names including their tree can become very long and unreadable. Use this option to shorten them to the last portion of the full MIB name.

Append MIB description after MIB name**File Configuration field:**

nAddMibDescriptionToMsg

Description:

Append the MIB description after the MIB name. **Attention, this can be a lot of information.**

Compress output format (remove spaces/quotations)**File Configuration field:**

nCompressOutputFormat

Description:

When enabled the output format will be reduced to a minimum and comma separated. Here is a sample output:

```
source=127.0.0.1, community=public, version=Ver2,
iso.3.6.1.2.1.1.3.0=Timeticks: (3493305159) 404 days, 7:37:31.59,
iso.3.6.1.6.3.1.1.4.1.0=OID: iso.3.6.1.4.1.19406.1.2.2,
iso.3.6.1.4.1.19406.1.1.1.7=This is a SyslogTest
```


General Values (Common settings for most services)

RuleSet to Use

File Configuration field:

szRuleSetName

Description:

Name of the ruleset to be used for this service. The RuleSet name must be a valid RuleSet.

Please Note:

Managing incoming Traps works the same way as with a Syslog server for example.

Incoming Traps will be forwarded to the corresponding Ruleset and pass by rule after rule. There it can be filtered for general information like the "Community", the "Version" or "Value" for example. Finally it will be processed by an action, which you can select to your needs. The SNMP Agent service will co-exist peacefully next to the Windows SNMP Agent and will not hinder it in its functionality. The Windows SNMP Agent listens to port 161, while MonitorWare Agent and WinSyslog listen to port 162.**

For internal processing, the variables of incoming SNMP messages will be added to a new property. Those properties will be named %snmp_var_x% with the x being a number starting with 1. You can use these custom properties for filtering and everywhere where you can use or print properties. For example, you can create a "send mail"-action. Here you can specify complete freely how the message will look like. You can use a introductory text and then let it show the error message in some context. This could look like this:

```
Hello Admin,
the following error occurred
%snmp_var_5%
Please take care at once.
Very urgent!
```

The result will be, that the 5th property of the snmp trap will be inserted into the message text.

Syslog server

Configures a Syslog server service. Multiple protocols (IPv4/IPv6 and UDP/TCP) can be configured and are supported.

When configuring Syslog Services, the functionality can be checked using the Test Syslog server button. It will open the Syslog Test Message function from the configuration client.

Services > Syslog Server | ☒ Enabled | Comments | Settings | Confirm | Reset |

Test Syslog Server

Internet Protocoltype: IPv4

Protocol Type: TCP

IP Address: 0.0.0.0

Listener Port: 514

RuleSet to use: Default RuleSet | Refresh

- Service - Syslog server Global Properties*

Internet Protocoltype

File Configuration field:

nlNetType

Description:

Select the desired protocol type. IPv4 and ipv6 are available. The IPv6 protocol needs to be properly installed in order to be used. **Note that one Service can only handle IPv4 or IPv6, so if you want to use both protocols, you will need to create two separate services.**

Protocol Type

File Configuration field:

nProtocolType

Description:

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 RFC 3195 RAW standard. Depending on which protocol type you choose, you get different option tabs. General and encoding are the same for everyone.

IP Address

File Configuration field:

szMyIPAddress

Description:

The Syslog server can now be bound to a specific IP address. You can either use an IPv4, an IPv6 Address, or a Hostname that resolves to an IPv4 or IPv6 Address. This feature is useful for multihome environments where you want to run different Syslog Servers on different IP Addresses. Please note that the default IP address 0.0.0.0 means ANY IP Address.

Listener Port

File Configuration field:

nListenPort

Description:

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.

RuleSet to use

File Configuration field:

szRuleSetName

Description:

Name of the ruleset to be used for this service. The RuleSet name must be a valid RuleSet.

General Options

The screenshot shows the 'General' configuration tab for a Syslog server. It includes the following options:

- ☐ Resolve Hostnames
- ☐ Take source system from Syslog message
- ☒ Save original source into property
- Propertyname:
- ☐ Escape control characters
- ☒ Enable RFC3164 Parsing
- ☐ Use original message timestamp (RFC 3164)
- ☐ Try to parse year from message timestamp (RFC 3164)
- ☒ Enable RFC5424 Parsing
- ☐ Append ProcessID to Syslogtag if available

- Service - Syslog server General Tab*

Resolve Hostnames

File Configuration field:

nResolveNames

Description:

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

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

Take source system from Syslog message

File Configuration field:

nTakeSourceSysFromSyslogMsg

Description:

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!

Save original source into property

File Configuration field:

nSaveSourceIntoProperty

Descriptions:

When this options is enabled, the original network source will be stored into the custom defined property (%sourceorig% by default). In case the original network source is needed for filtering for example.

Escape Control Characters

File Configuration field:

nEscapeControlCharacters

Description:

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

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

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

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

Enable RFC3164 Parsing

File Configuration field:

nRFC3164Parsing

Description:

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

Use Original Message Timestamp

File Configuration field:

nParseSyslogDate

Description:

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

Try to parse year from message timestamp (RFC3164)

File Configuration field:

nRFC3164DetectYear

Description:

If enabled, the service will try to detect a Year after the usual RFC3164 Date Header.

Enable RFC5424 Parsing

File Configuration field:

nRFC5424Parsing

Description:

If this box is checked, rfc 5424 compliant message parsing is enabled for Syslog RFC5424 Header detection and decoding. This also involves new usable Syslog properties. If unchecked, "traditional" Adiscon message parsing is selected. If you experience trouble with the sender host name or the timestamp, we suggest that you turn off RFC 5424 compliant message parsing. Many existing devices do not fully comply with RFC 5424 and this can cause those issues.

Append ProcessID to SyslogTag if available

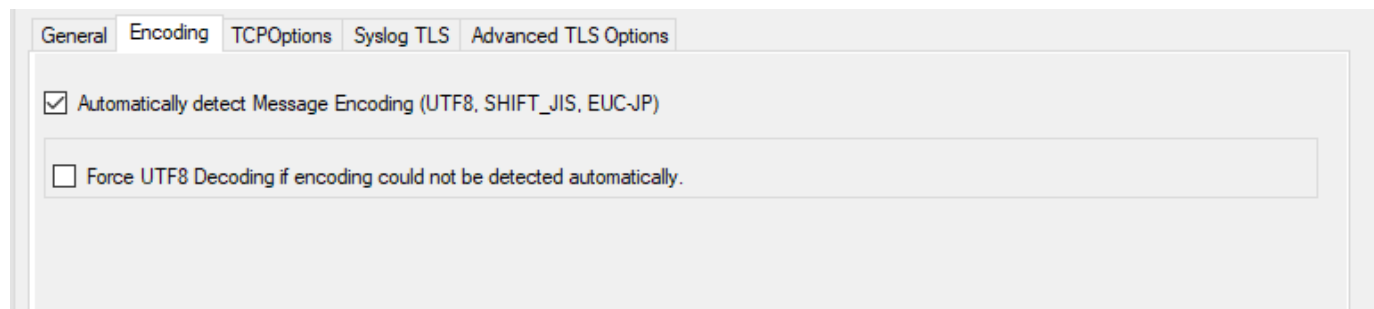
File Configuration field:

nRFC5424AddProclD2SyslogTag

Description:

This option is related to RFC5424 header parsing and was default in previous versions. However the default now is off in order to separate the Syslogtag from the ProcessID.

Encoding Options



The screenshot shows the 'Encoding' tab of the Syslog server configuration. It has five tabs: 'General', 'Encoding', 'TCPOptions', 'Syslog TLS', and 'Advanced TLS Options'. The 'Encoding' tab is active. It contains two checkboxes: 'Automatically detect Message Encoding (UTF8, SHIFT_JIS, EUC-JP)' which is checked, and 'Force UTF8 Decoding if encoding could not be detected automatically.' which is unchecked.

- Service - Syslog server Encoding Tab*

Automatically detect Message Encoding (UTF8, SHIFT_JIS, EUCJP)

File Configuration field:

nTryDetectMessageEncoding

Description:

If enabled, the message will be checked for different encodings. This is important if you have syslog messages with multibyte characters. Once an encoding is detected, it will automatically be converted into UTF16 internally.

Force UTF8 Decoding

File Configuration field:

nForceUTF8Decoding

Description:

This option forces UTF8 Decoding of all incoming messages. This is also useful for syslog messages encoded in UTF8 but missing the BOM within the Syslog message.

UDP Options



The screenshot shows the 'UDP Options' tab of the Syslog server configuration. It has three tabs: 'General', 'Encoding', and 'UDP Options'. The 'UDP Options' tab is active. It contains a checkbox 'Enable receiving from a UDP Multicast Group' which is unchecked. Below it is a text field labeled 'Multicast Address' with the value '224.0.0.1' and a dropdown arrow on the right.

- Service - Syslog server UDP Options Tab*

Enable receiving from a UDP Multicast Group

File Configuration field:

nEnableMultiCastGroup

Description:

This option supports receiving Syslog messages via multicast IP Addresses like 224.0.0.1 for example.

TCP Specific Options

General Encoding **TCPOptions** Syslog TLS Advanced TLS Options

Session Timeout 15 Minutes

☒ Messages are separated by the following sequence

Message separation sequence \n

☐ Enable multiple message separators

List of additional separators

	Message separation sequence
▶	\r\n
*	\r\n

Message Completion Timeout 15 seconds

- Service - Syslog server TCP Options Tab*

Session Timeout

File Configuration field:

nTimeOutSession

Description:

One of the TCP-specific options is the session timeout. This value declares, how long a TCP session may be kept open, after the last package of data has been sent. You can by default set values between 1 second and 1 day or you can use a custom value with a maximum of 2147483646 milliseconds. If you wish to disable the session timeout, you can use a custom value of 0 milliseconds to disable it.

Messages are separated by the following sequence

File Configuration field:

szMsgSep_[n]

Description:

If this option is checked, you can use multiple messages in the same transmission and the following options are enabled: Message separation sequence and Message Completion Timeout.

Message separation sequence

File Configuration field:

nEnableTCPMsgSep

Description:

Determines, how you want to separate the messages. By default “\r\n” is the value for this, as most times a message ends with a carriage return and/or a line feed. But, you can choose your own separation sequence here as well.

Enable multiple message separators**File Configuration field:**

nEnableMultiTCPMsgSep

Description:

If you choose the checkbox you can use more than one message separator.

Message Completion Timeout**File Configuration field:**

nTimeOutMsg

Description:

Here you can set the time that is allowed to complete a message. If the time is exceeded, but the message not yet completed, the rest will be treated as a new message. The counter is reset each time a new message begins. You can choose from multiple values between 1 second and 1 day, or choose a custom value in milliseconds (0 = disable, maximum = 2147483646)

Syslog TLS

General Encoding TCPOptions **Syslog TLS** Advanced TLS Options

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

TLS Mode: Anonymous authentication

Select common CA PEM: Browse

Select Certificate PEM: Browse

Select Key PEM: Browse

Permitted Peers

	Permitted Peername / SHA1 / etc
*	*Enter value for Permitted Peername / SHA1 / etc*

- Service - Syslog server Syslog TLS Tab*

Enable SSL / TLS Encryption**File Configuration field:**

nUseSSL

Description:

This option enables SSL / TLS encryption for your Syslog server. Please note, that with this option enabled, the server only accepts SSL / TLS enabled senders.

TLS Mode**File Configuration field:**

nTLSMode

Description:

The TLS mode can be set to the following:

Anonymous authentication Default option, which means any client certificate will be accepted, or even none.

x509/name (certificate validation and name authentication) When this mode is selected, the subject within the client certificate will be checked against the permitted peers list. This means the Syslog server will only accept the secured connection if it finds the permitted peer in the subject.

509/fingerprint (certificate fingerprint authentication) This mode creates a SHA1 Fingerprint from the client certificate it receives, and compares it to fingerprints from the permitted peers list. You can use the debuglog to see fingerprints of client certificates which were not permitted.

x509/certvalid (certificate validation only) A Syslog Sender is accepted when the client certificate is valid. No further checks are done.

Select common CA PEM**File Configuration field:**

szTLSCAFile

Description:

Select the certificate from the common Certificate Authority (CA), the syslog receiver should use the same CA.

Select Certificate PEM**File Configuration field:**

szTLSCertFile

Description:

Select the client certificate (PEM Format).

Select Key PEM**File Configuration field:**

szTLSKeyFile

Description:

Select the keyfile for the client certificate (PEM Format).

Permitted Peers**Permitted Peername / SHA1 / etc.****File Configuration field:**

szIP_[n]


Description:

This list contains all permitted peers. If x509/name is used, this can contain parts of the client certificate subject. For example if you have CN = secure.syslog.msg in the certificate subject, you can add "secure.syslog.msg" as permitted peer. When using x509/fingerprint, this list holds a list of permitted SHA1 fingerprints. The fingerprints can either be generated with OpenSSL Tools or grabbed from the debug logfile. The format is like described in RFC 5425, for example: SHA1:2C:CA:F9:19:B8:F5:6C:37:BF:30:59:64:D5:9A:8A:B2:79:9D:77:A0.

Advanced TLS

General
Encoding
TCPOptions
Syslog TLS
Advanced TLS Options

☐ Allow SSL v3 (insecure)
☐ Allow TLS v1.0 (insecure)
☒ Allow TLS v1.1
☒ Allow TLS v1.2
☐ Use OpenSSL configuration commands

 By enabling this option, you can set OpenSSL configuration commands directly. For more informations on available configuration parameters for each command type, visit this page:
https://www.openssl.org/docs/man1.0.2/ssl/SSL_CONF_cmd.html

Configuration commands list

	Command Type	Command Value
*	Protocol	ALL,-SSLv2,-SSLv3,-TLSv1,-TLSv1.1

- Service - Syslog server Advanced TLS Options Tab*

Allow SSL v3

File Configuration field:

nTLSAllowSSLv3

Description:

This option enables insecure protocol method SSLv3. We recommend NOT enabling this option as SSLv3 is considered broken.

Allow SSL v1.0

File Configuration field:

nTLSAllowTLS10

Description:

This option enables insecure protocol method TLSv1. We recommend NOT enabling this option as TLSv1 is considered broken.

Allow SSL v1.1

File Configuration field:

nTLSAllowTLS11

Description:

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

Allow SSL v1.2

File Configuration field:

nTLSAllowTLS12

Description:

This option enables protocol method TLS1.2 which is enabled by default.

Use OpenSSL configuration commands

File Configuration field:

nTLSUseConfigurationCommands

Description:

By enabling this option, you can set OpenSSL configuration commands directly. For more information's on available configuration parameters for each command type, visit this page: https://www.openssl.org/docs/man1.0.2/ssl/SSL_CONF_cmd.html

We allow to the set the following OpenSSL configuration commands in the configuration commands list.

- CipherSuite: This sets the available ciphers for TLS >= v1.3. For TLS < v1.3 use Ciphers instead. Note: setting this option will OVERWRITE the internal default CipherSuite.
- Ciphers: This sets the available ciphers for TLS < v1.3. For TLS >= v1.3 use CipherSuite instead. Setting this option will OVERWRITE the internal default cipher list.
- CipherString: Sets the allowed/disallowed used Ciphers. Setting this value will OVERWRITE the internal default ciphers.
- SignatureAlgorithms: This sets the supported signature algorithms for TLS v1.2.
- Curves: This sets the supported elliptic curves.
- Protocol: Sets the supported versions of the SSL or TLS protocol. This will OVERWRITE the Allow SSL options from above!
- Options: The value argument is a comma separated list of various flags to set.

Allow TLS v1.3

File Configuration field:

nTLSAllowTLS13

Description:

This option enables protocol method TLS1.3 which provides enhanced security and performance.

When setting advanced configuration commands, we highly recommend to enable

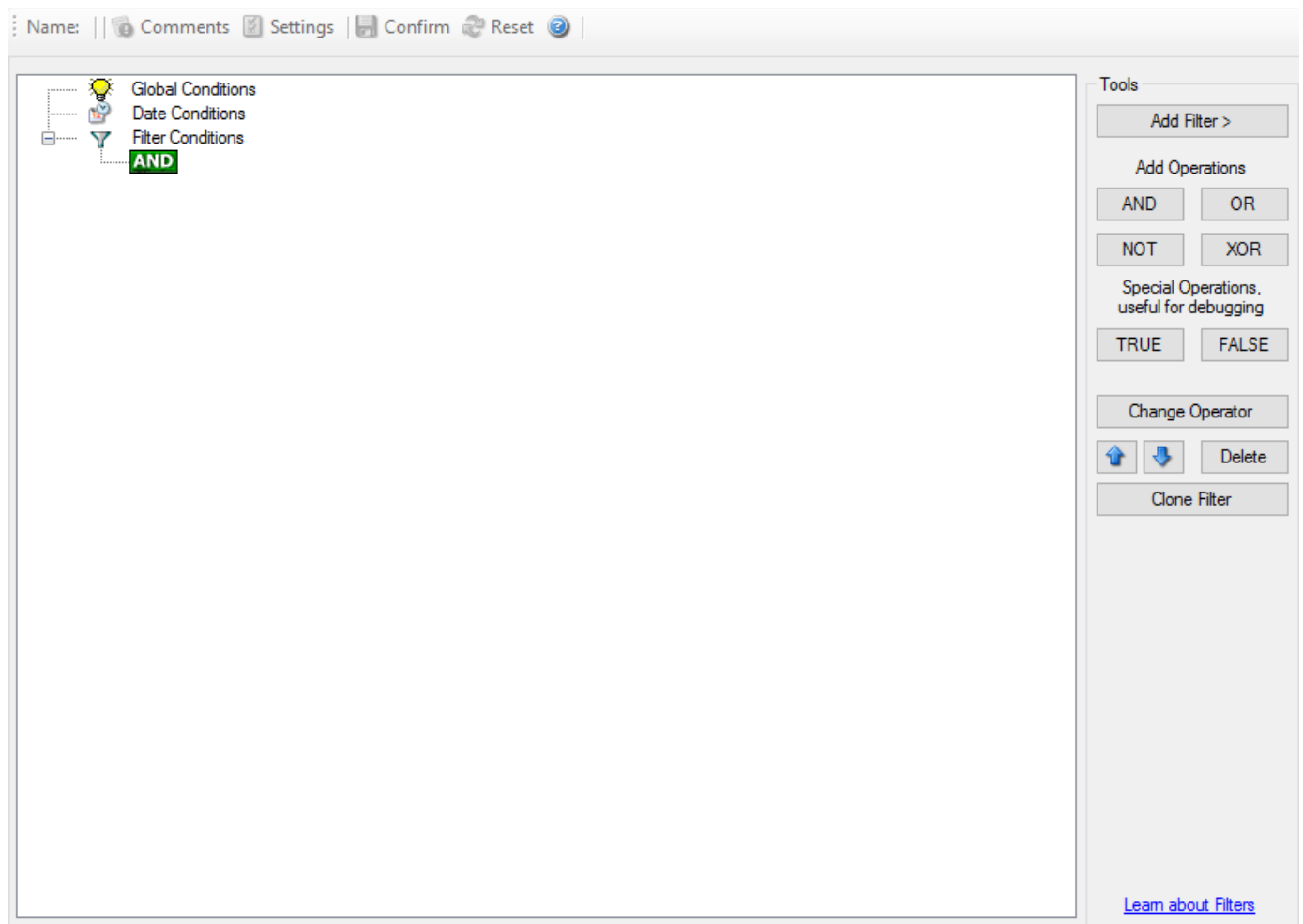
debug logging and review it after changes have been made. An error will be logged in the debug logfile if a configuration command cannot be processed successfully.

Filter Conditions

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

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

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



Filter Conditions - Figure 1

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:

Name: | Comments | Settings | Confirm | Reset | ?

Global Conditions

- Date Conditions
- Filter Conditions
 - AND
 - EVAL Property: %IsEventlogMonitor% = "1"
 - EVAL Property: %id% = 560
 - EVAL Property: %sourceproc% contains "Security"
 - EVAL Property: %user% contains "P15111116\IUSR_ROOTSERVER"
 - EVAL Property: %msg% contains ".exe"
 - NOT
 - OR
 - EVAL Property: %msg% contains "\\usr\\bin\\perl.exe"
 - EVAL Property: %msg% contains "\\PHP\\php.exe"

Tools

Add Filter >

Add Operations

AND OR

NOT XOR

Special Operations, useful for debugging

TRUE FALSE

Change Operator

↑ ↓ Delete

Clone Filter

[Learn about Filters](#)

Global Conditions

☒ Threat not found filters as TRUE

☐ Fire only if Event occurs times within seconds.

☐ Minimum Wait Time seconds.

☐ Global Conditions relative to this property [Insert](#)

Filter Conditions - Figure 2

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

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

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

In order to avoid too frequent alerts, we also have specified a minimum wait time of 60 seconds. Therefore, the filter condition evaluates as "true" at most every 60 seconds, even if all other conditions are true.**Note:** If you want to know more about complex filter conditions you can click on the "Learn about Filters" link.

String comparison in Filter Conditions are "Case Sensitive"! For example, if the Source System name is "ws01" and you had written "WS01" while applying the

filter, then this filter condition would***NEVER*** evaluate to True! Please double check before proceeding further!

If you are not still sure about what to do, you can drop a word about your requirements to <https://ticket.adiscon.com>, and we look into it!

Global Conditions

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

Global Conditions

☒ Treat not found filters as TRUE

☐ Fire only if Event occurs

times within

seconds.

☐ Minimum Wait Time

seconds.

☐ Global Conditions relative to this property

[Insert](#)

- Global Conditions*

Treat not found Filters as TRUE

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

Fire only if Event occurs

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

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

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

Minimum Wait Time

This filter condition can be used to prevent rules from firing too often. For example, a rule might be created to check the status of a port probe event. The port probe probes an smtp server. If the event is fired and the rule detects it, it spawns a process that tries to restart the service. This process takes some time. Maybe the SMTP gateway need some more time to fully start up so that the port probe might fail again while the problem is already taken care of. The port probe as such generates an additional event.

Setting a minimum wait time prevents this second port probe event to fire again if it is – let’s say – within 5 minutes from the original one. In this case, the minimum wait time is not yet reached and as such, the rule does not match. If, however, the same event is generated 5 hours later (with the mail gateway failing again), the rule once again fires and corrective actions are taken.

Global Conditions relative to this property

This feature enables you to control the Global Conditions based on a property.

For example take the source of a message as property. In this case, the Minimum WaitTime for example would be applied individual on each message source.

Date Conditions

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

Date Conditions

☒ Always process Rule
☐ Process only after Installation Date
☐ Process only after custom date:

- Date Conditions*

Always process Rule

No date filter will be applied

Process only after Installation Date

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

Process only after custom date

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

Operators

In general, operators describes how filter conditions are linked together. The following operators can be used. Boolean operators like “AND” or “OR” can be used to create complex filter conditions.

AND:

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

OR:

If one or both of the filters placed below is true, OR returns TRUE.

NOT:

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

XOR:

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

TRUE:

Useful for debugging, just returns TRUE.

FALSE:

Useful for debugging as well, returns FALSE.

Filters

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

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

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

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

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

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

String:

Can be compared to another String with "=", "Not =", "Range Match" or through

REGEX Compare Operation

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

Here are some regular expressions samples:

Regular Expression:

```
[0-9]{4,4}-[0-9]{1,2}-[0-9]{1,2} [0-9]{1,2}:[0-9]{1,2}:[0-9]{1,2}
```

Matches typical Date like 2018-11-20 12:11:01

Regular Expression: `\n[0-9]{4,4}`

Matches Linefeed and 4-digit number.

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

More samples and details about the Regular Expression Syntax can be found here:

[https://msdn.microsoft.com/en-us/library/bb982727\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/bb982727(v=vs.90).aspx)

number:

can be compared with another number with "=", "not =", "<" and ">"

boolean:

can be compared to either true or false with "=" and "not ="

time:

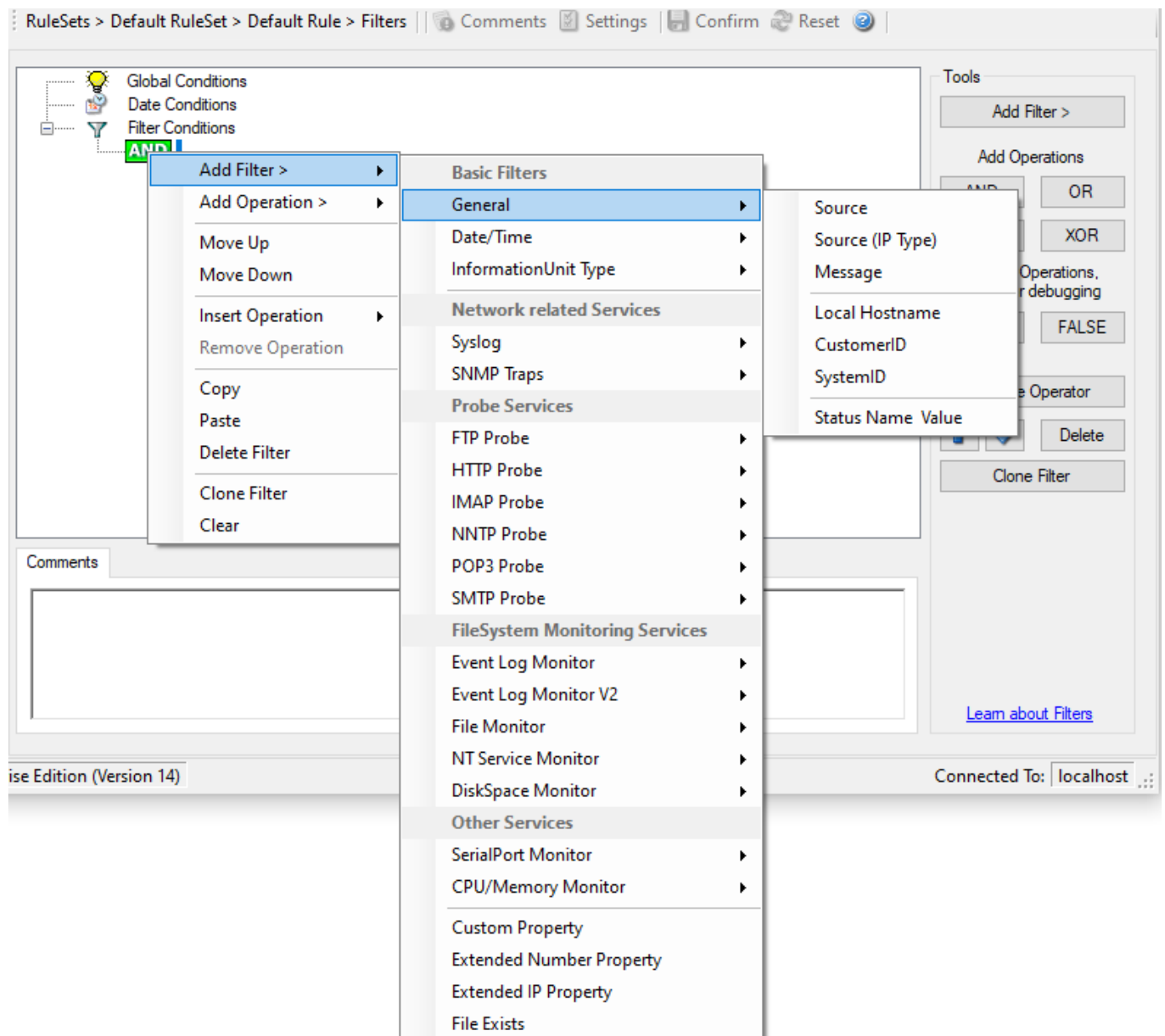
can be compared with another time but only with "="

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

basic filters

General

These are non-event log specific settings.



- Filter Conditions - General*

Source

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

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

Source System (IP)

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

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

Please see the description for extended ip property for more information on how to use this property.

Message Content

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

The content search can be limited to a region within the message. To do so, select a starting and ending position within the string by choosing the “**contains within range**” compare operation. This can be done by specifying the start range and end range into the respective boxes.**Please note that you can enter the character position you desire in these fields. The default “Start Range” and “End Range” are set to 0.**

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

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

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

This filter is of type string.

CustomerID

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

CustomerID (Type=Number).

SystemID

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

SystemID (Type=Number).

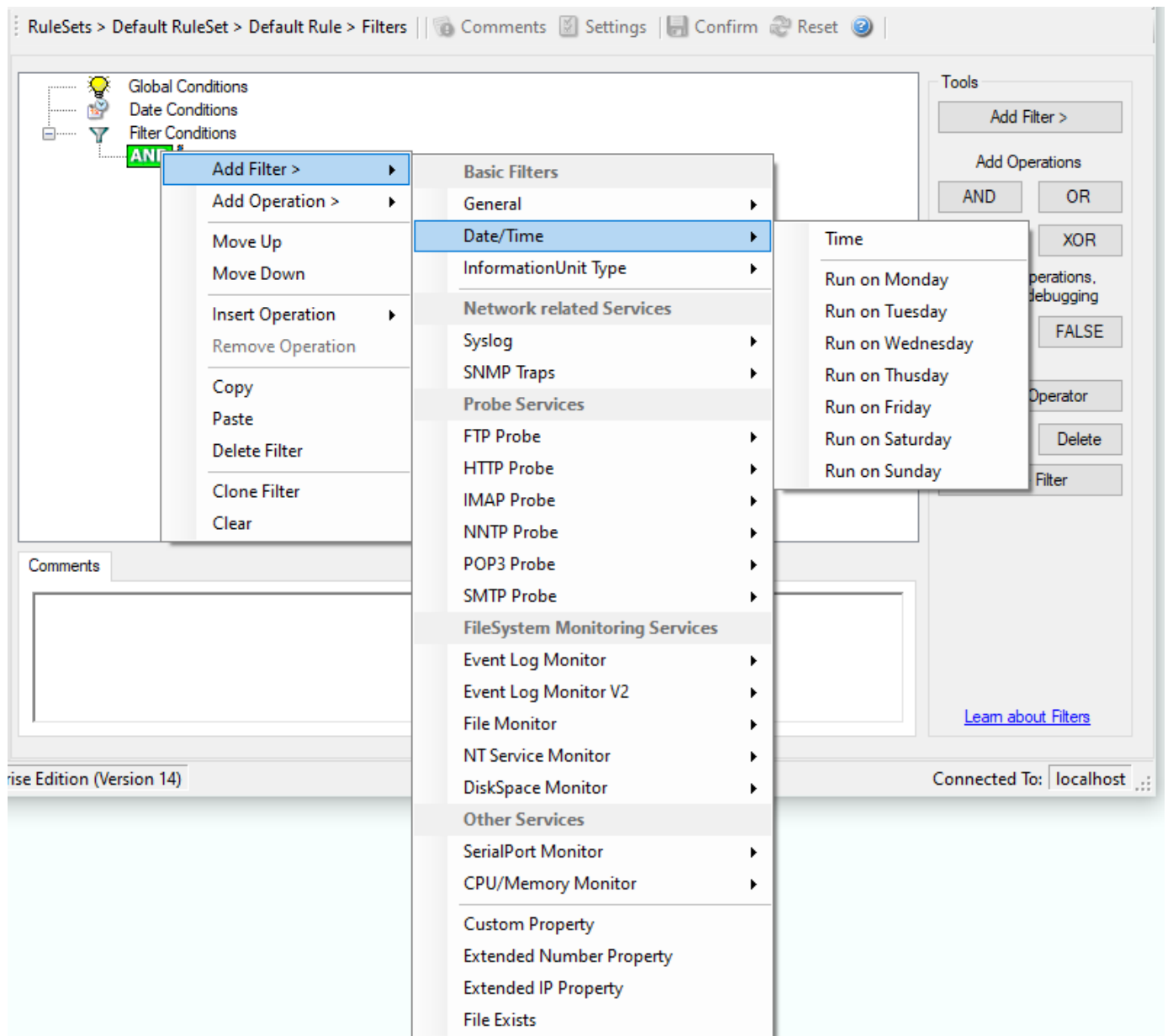
Status Name and Value

These filter type corresponds to set status action .

Status Name and Value (Type=String)

Date/Time

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



- Filter Conditions - Date/Time*

Time

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

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

Weekdays

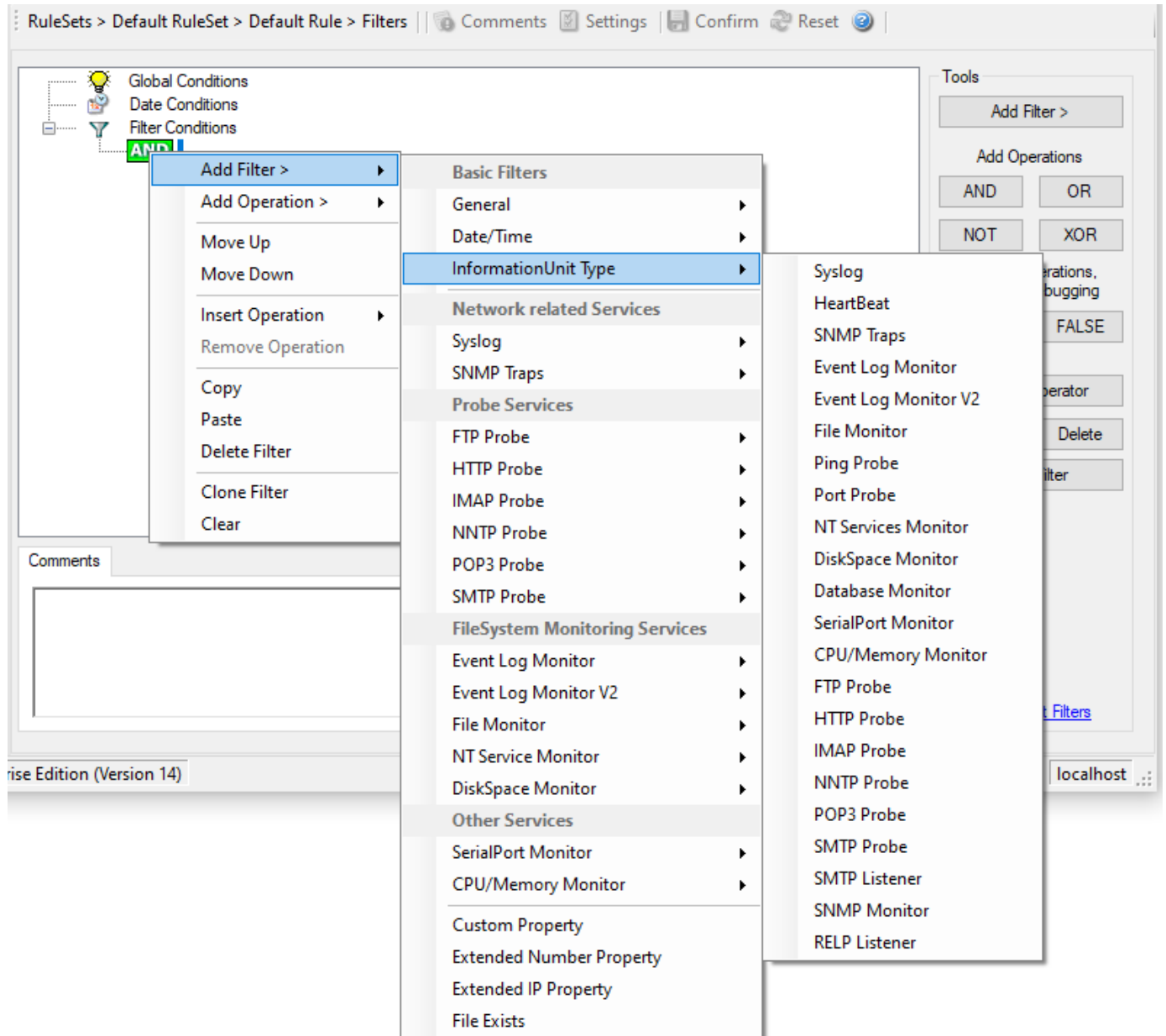
This is closely equivalent to the time filter condition, except that it is applied on a per-day basis. So it can be used to detect for example events occurring on weekends and act differently on them. The following filters are available:

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

6. Run on Saturday (Type=Boolean)
7. Run on Sunday (Type=Boolean)

InformationUnit Type

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



• Filter Conditions - InformationUnit Type*

The following filters are available:

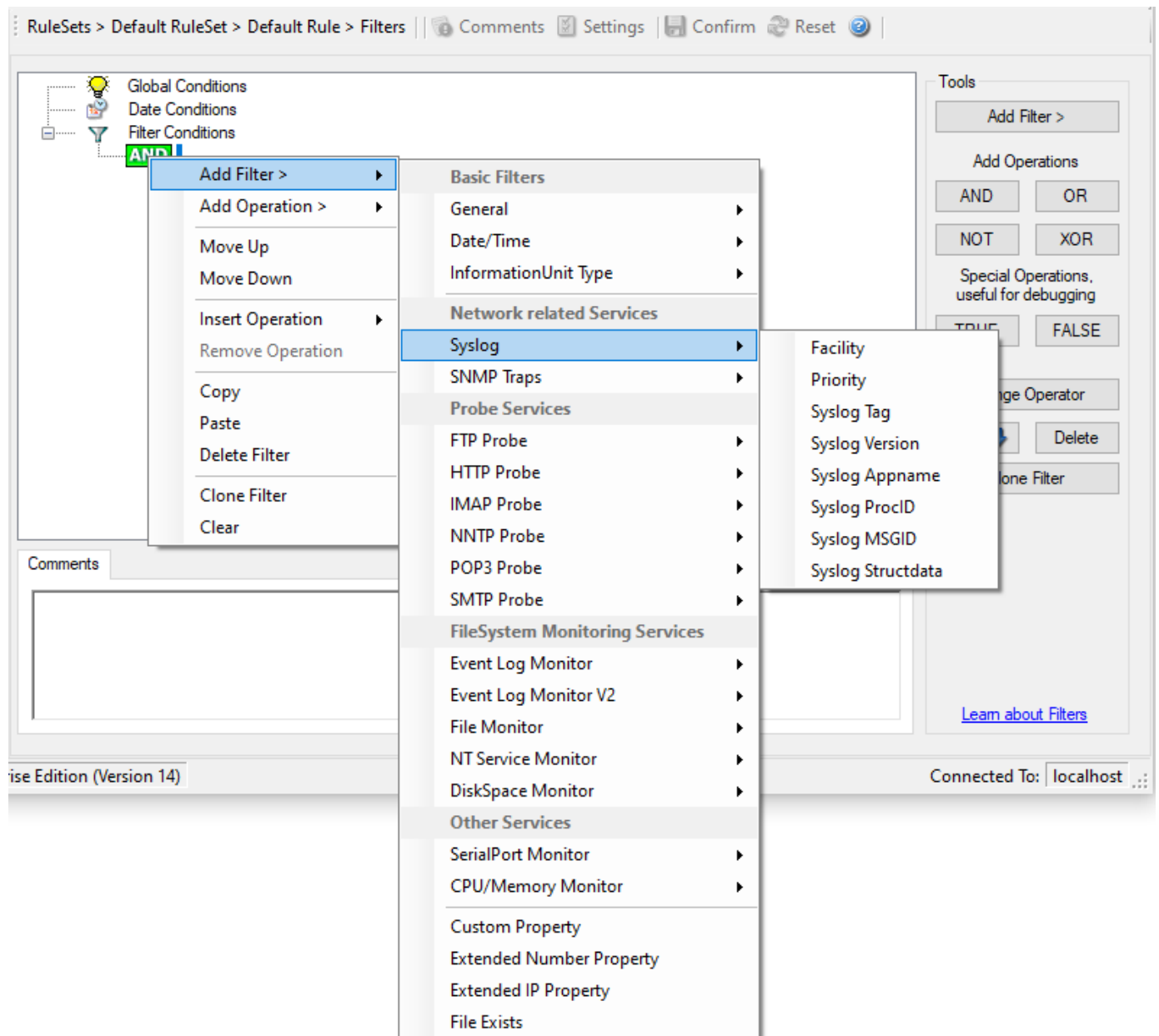
1. Syslog (Type=Boolean)
2. Heartbeat (Type=Boolean)
3. SNMP Traps (Type=Boolean)
4. Event Log Monitor (Type=Boolean)
5. File Monitor (Type=Boolean)
6. Ping Probe (Type=Boolean)

- 7 . Port Probe (Type=Boolean)
- 8 . NT Services Monitor (Type=Boolean)
- 9 . Disk Space Monitor (Type=Boolean)
- 10 Database Monitor (Type=Boolean)
- .
- 11 Serial Port Monitor (Type=Boolean)
- .
- 12 CPU/Memory Monitor (Type=Boolean)
- .
- 13 FTP Probe (Type=Boolean)
- .
- 14 HTTP Probe (Type=Boolean)
- .
- 15 IMAP Probe (Type=Boolean)
- .
- 16 NNTP Probe (Type=Boolean)
- .
- 17 POP3 Probe (Type=Boolean)
- .
- 18 SMTP Probe (Type=Boolean)
- .

network related filters

Syslog

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



- Filter Conditions - Syslog*

Syslog Facility

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

This filter is of type number.

Syslog Priority

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

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

This filter is of type number.

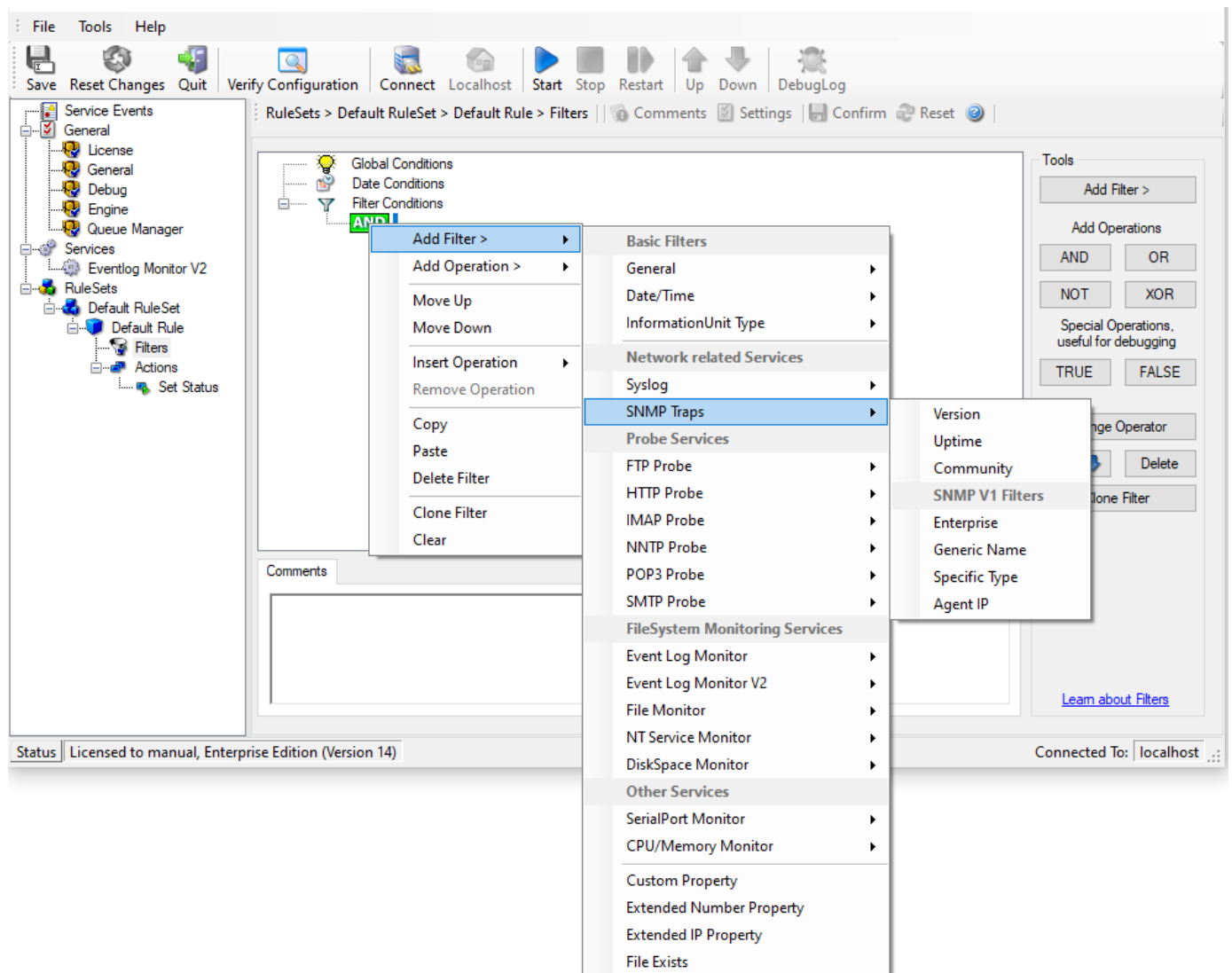
Syslog Tag

This filter is of type string.

SNMP Traps

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

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



- Filter Conditions - SNMP Traps*

Community

It corresponds to the respective SNMP entity.

This filter is of type string.

Enterprise

It corresponds to the respective SNMP entity.

This filter is of type string.

Generic name

It corresponds to the respective SNMP entity.

This filter is of type string.

Version

It corresponds to the respective SNMP entity.

This filter is of type number.

Uptime

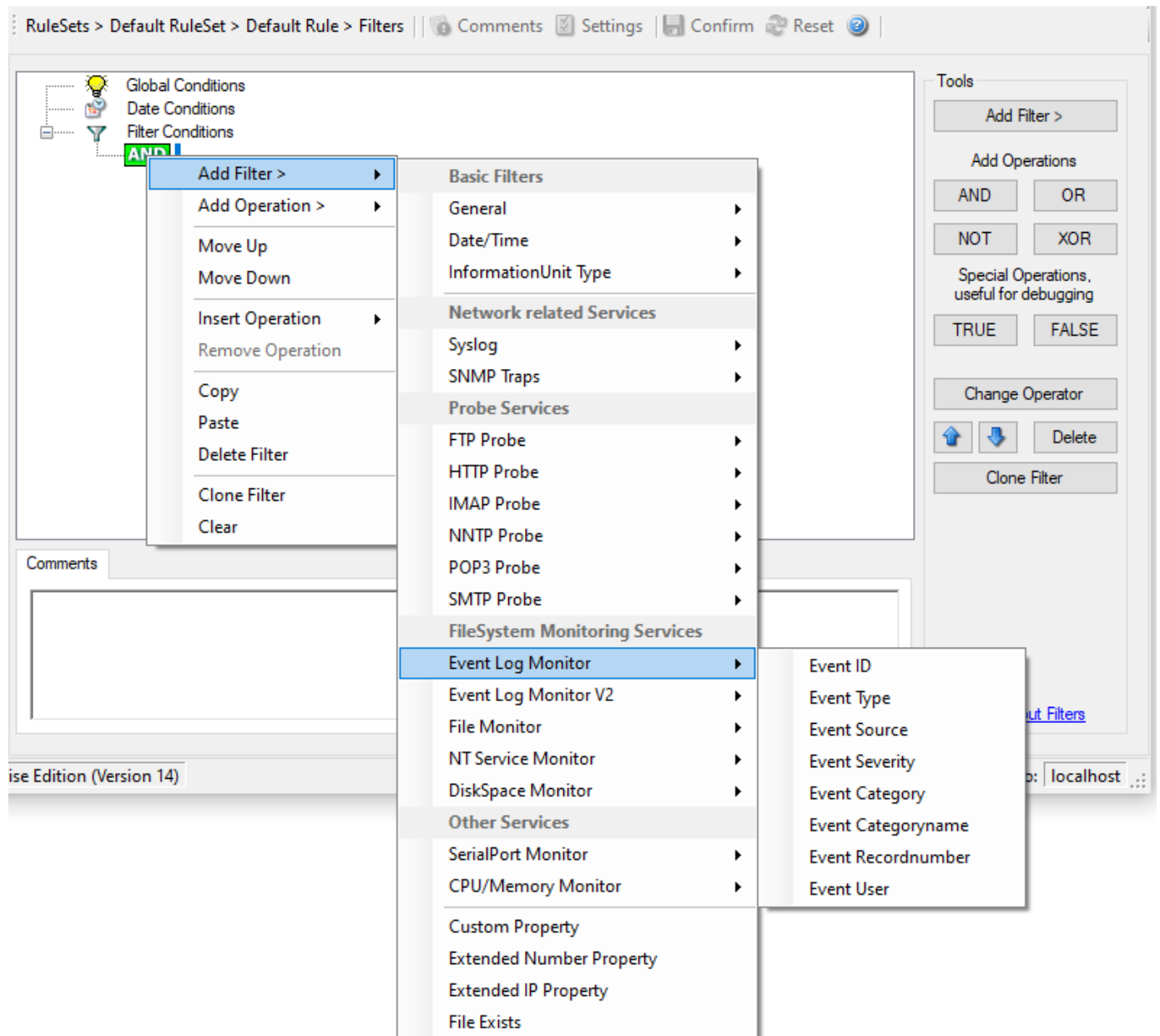
It corresponds to the respective SNMP entity.

This filter is of type string.

filesystem monitoring filters

Event Log Monitor

Event Log Monitor specific filters are grouped here.



- Filter Conditions - Event Log Monitor V1*

Event ID

This is the event log ID as specified in the Windows Event Log. If enabled, the event must have the configured event ID or the rule will not match. This is an integer value.

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

This filter is of type number.

Event Type

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

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

This filter is of type string.

Event Source

This is the event log source as specified in the Windows Event Log. If enabled, the event must have the configured event source or the rule will not match. This is a string value. There must be an exact match. Please note that this value is case-sensitive.

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

This filter is of type string.

Event Severity

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

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

This filter is of type number.

Event Category

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

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

This filter is of type number.

Event Categoryname

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

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

This filter is of type string.

Event Recordnumber

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

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

This filter is of type number.

Event User

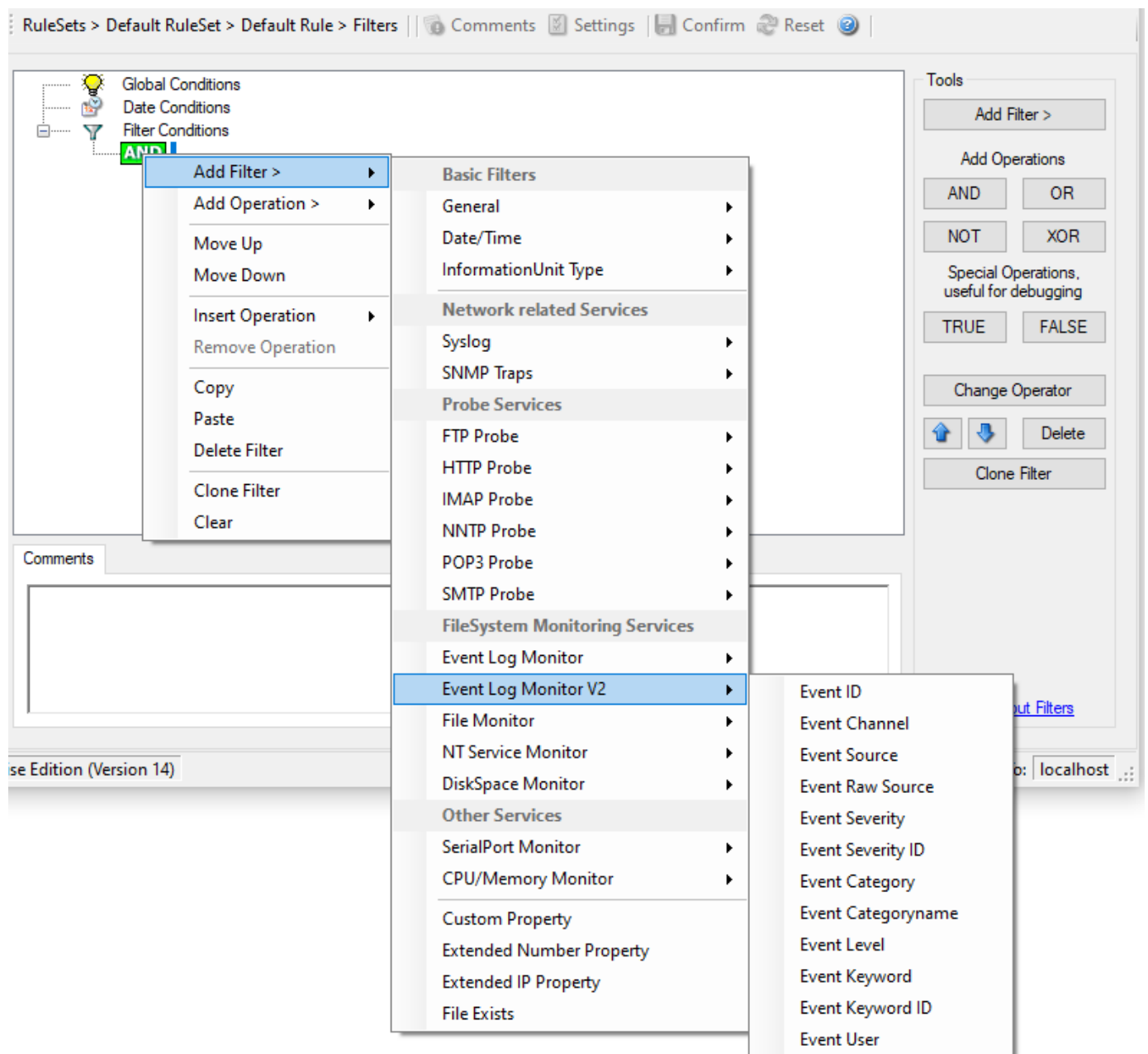
This is the event log user as specified in the Windows Event Log. If enabled, the event must have the configured event user or the rule will not match. Since it is a string value there must be an exact match. Please note that this value is case-sensitive.

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

This filter is of type string.

Event Log Monitor V2

Event Log Monitor V2 specific filters are grouped here.



- Filter Conditions - Event Log Monitor V2*

Event Channel

The channel property for event log entries, for classic Event logs they match the %nteventlogtype% property, for new event logs, they match the "Event Channel". If enabled, the event must have the configured event type or the rule will not match. There must be an exact match. Please note that this value is case-sensitive.

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

This filter is of type string.

Event Raw Source

This contains the full internal name of the event source for new event logs, for classic event logs it contains the same value as in %sourceproc%. If enabled, the event must have the configured event source or the rule will not match. This is a string value. There must be an exact match. Please note that this value is case-sensitive.

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

This filter is of type string.

Event SeverityID

This is the internal ID of the event log level as number. This is a integer value. There must be an exact match. Please note that this value is case-sensitive.

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

This filter is of type number.

Event Level

This is a textual representation of the event log level (which is stored as number in %severityid%). This property is automatically localized by the system. If enabled, the event must have the configured level or the rule will not match. There must be an exact match. Please note that this value is case-sensitive.

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

This filter is of type string.

Event Keyword

This is a textual representation of the event keyword. This property is automatically localized by the system. If enabled, the event must have the configured event keyword or the rule will not match. There must be an exact match. Please note that this value is case-sensitive.

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

This filter is of type string.

Event KeywordID

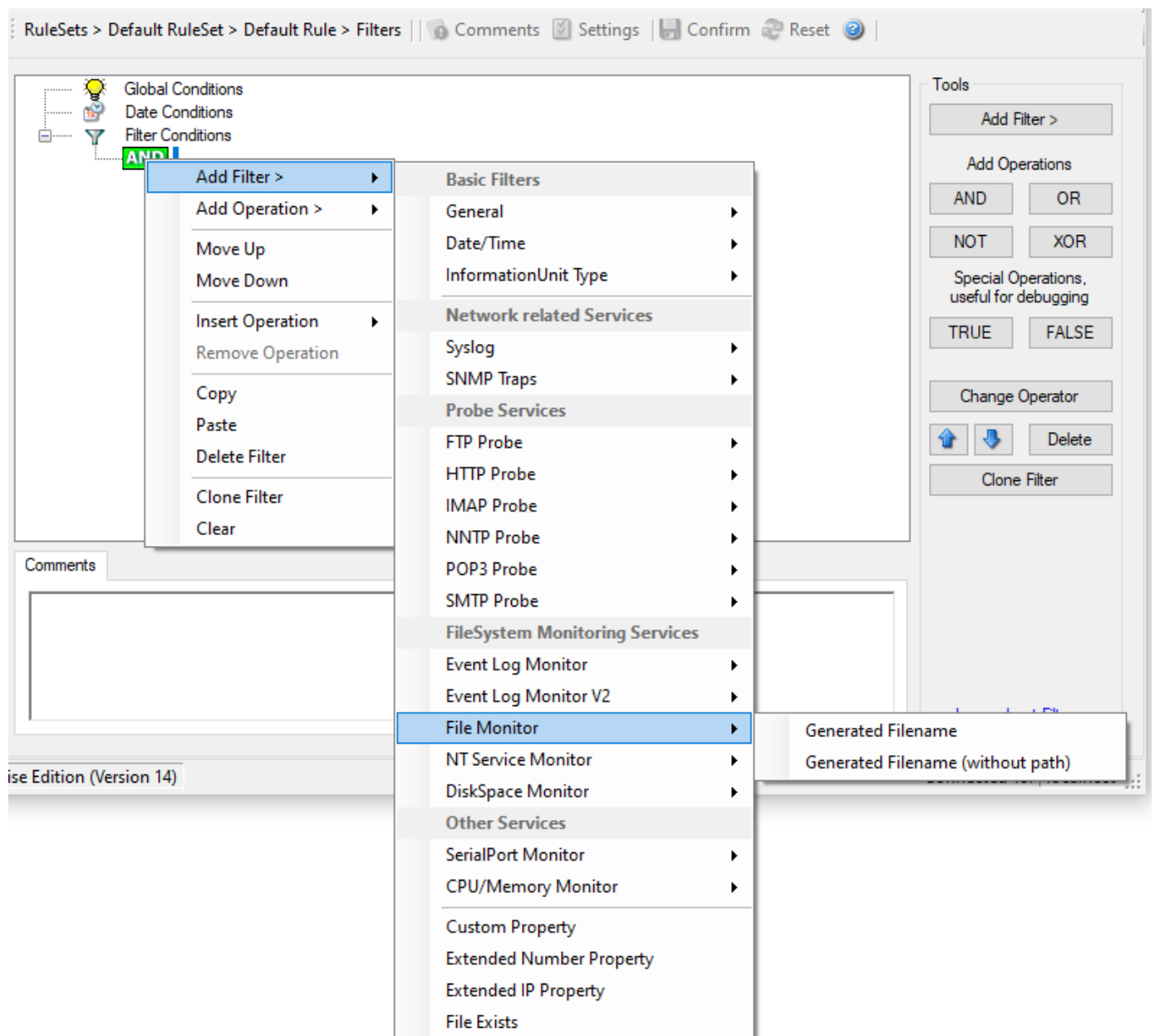
This is the internal keyword ID as string. If enabled, the event must have the configured event keyword ID or the rule will not match. There must be an exact match. Please note that this value is case-sensitive.

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

This filter is of type string.

File Monitor

File Monitor specific filter is described here.



- Filter Conditions - File Monitor*

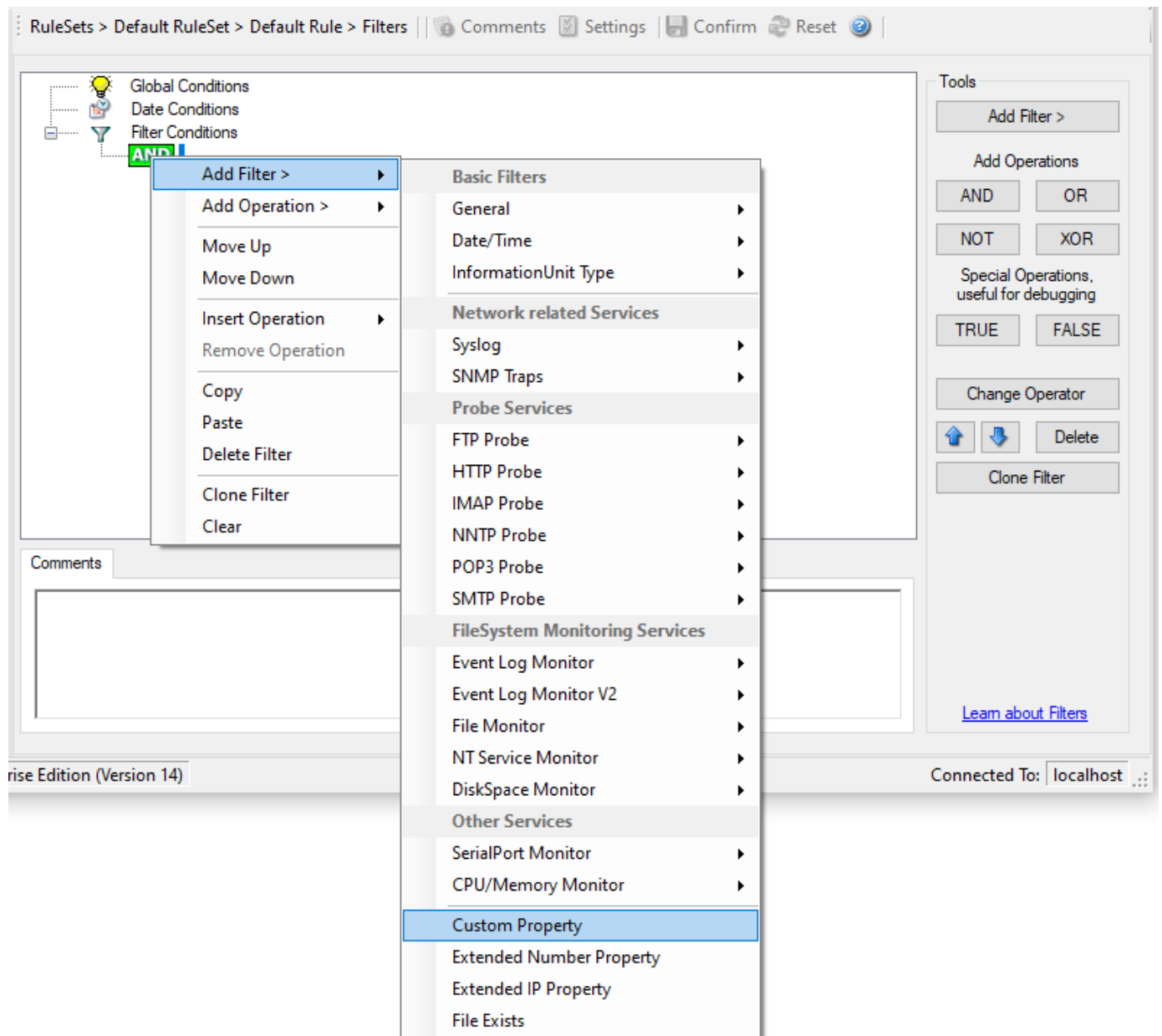
Generated Filename

The configured generic name of the file being reported. Filter has to match exactly to work.

custom properties

Custom Property

Custom Property specific filter is described here.



- Filter Conditions - Custom Property*

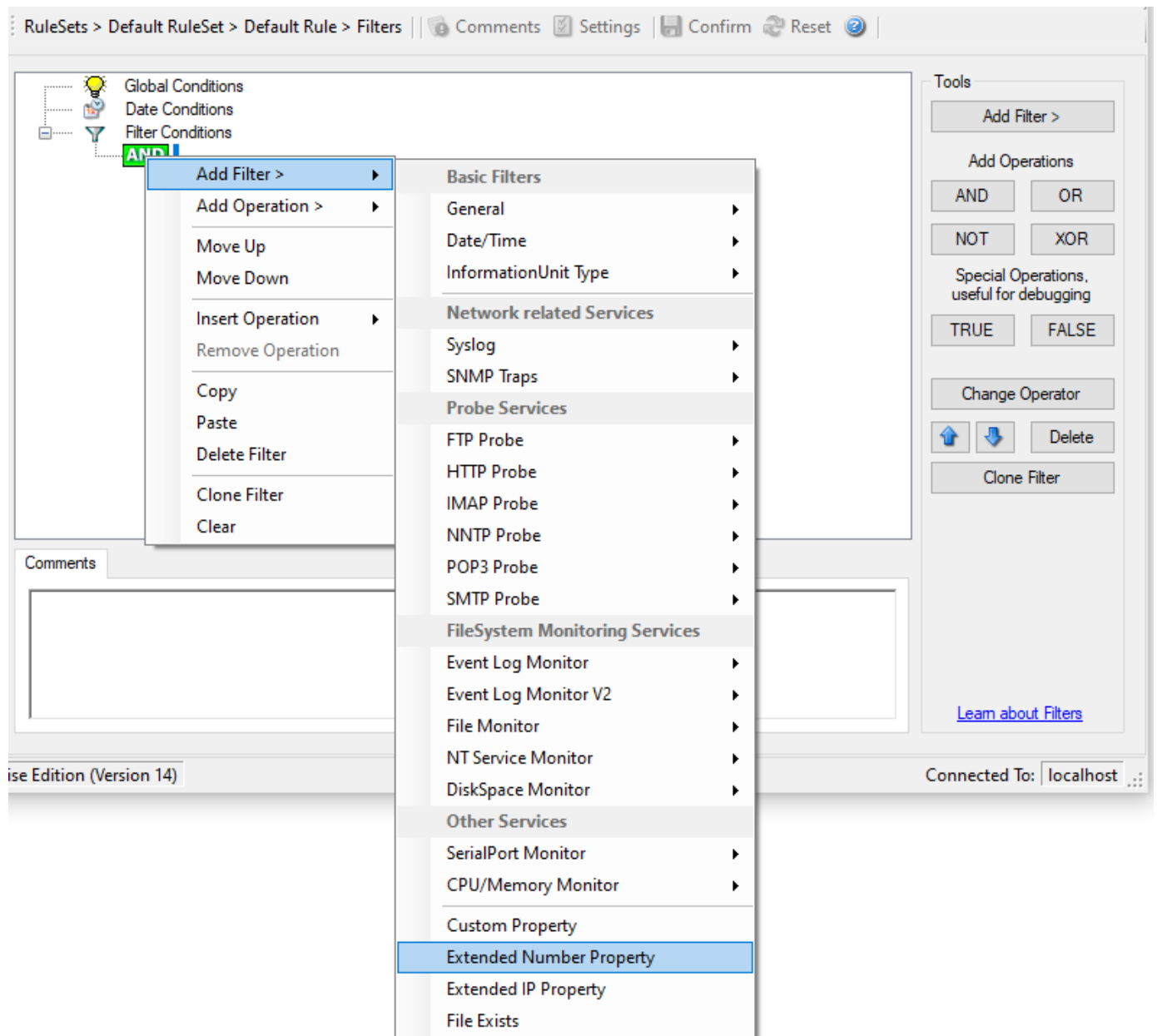
Custom Property

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

This filter is of type string.

Extended Number Property

Extended Number Property specific filter is described here.



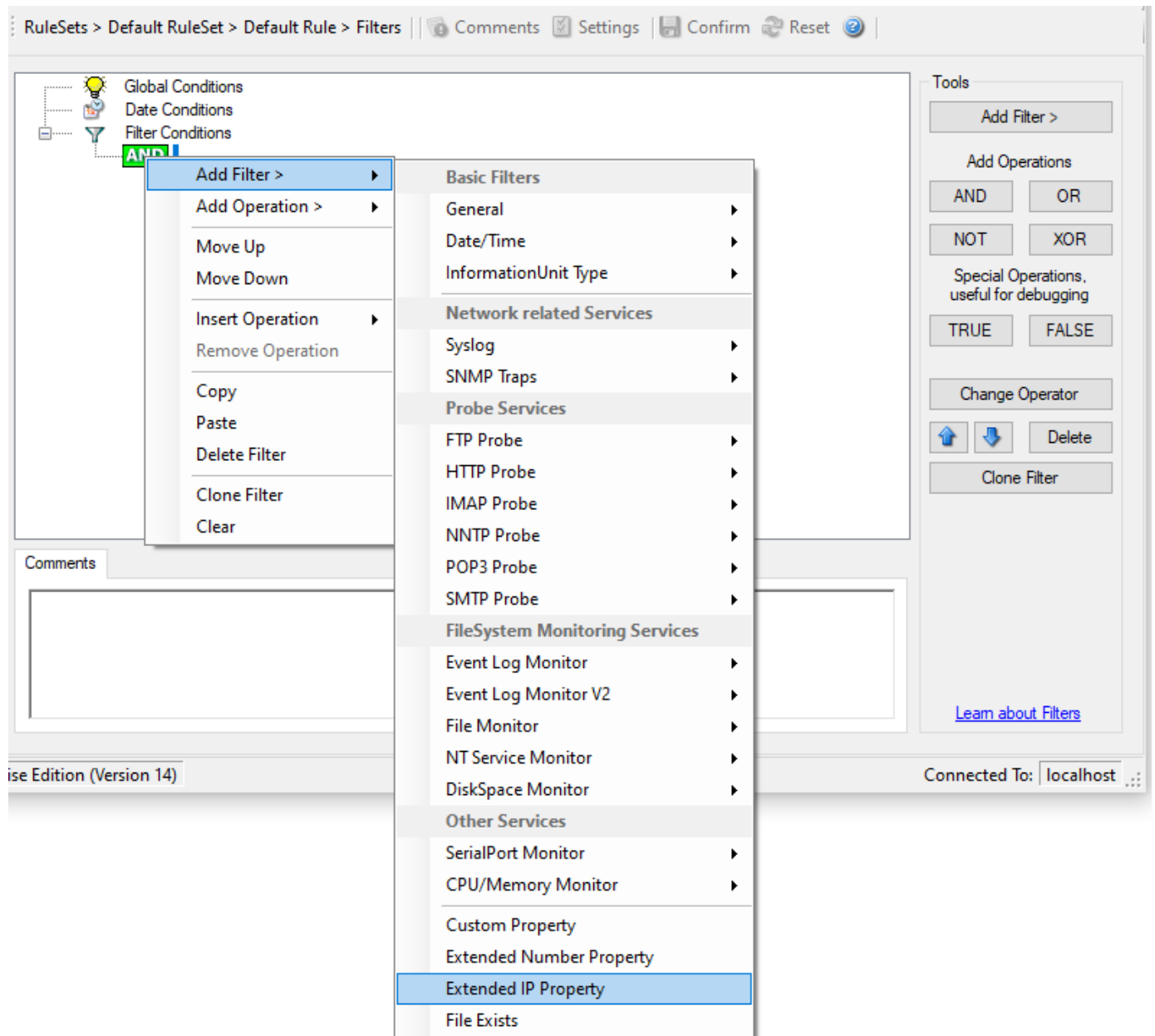
- Filter Conditions - Extended Number Property*

Extended Number Property

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

This filter is of type numeric.

Extended IP Property



- Filter Conditions - Extended IP Property*

Extended IP Property filter settings

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

Available compare operations for the IP Filter Type are:

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

Configuring

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

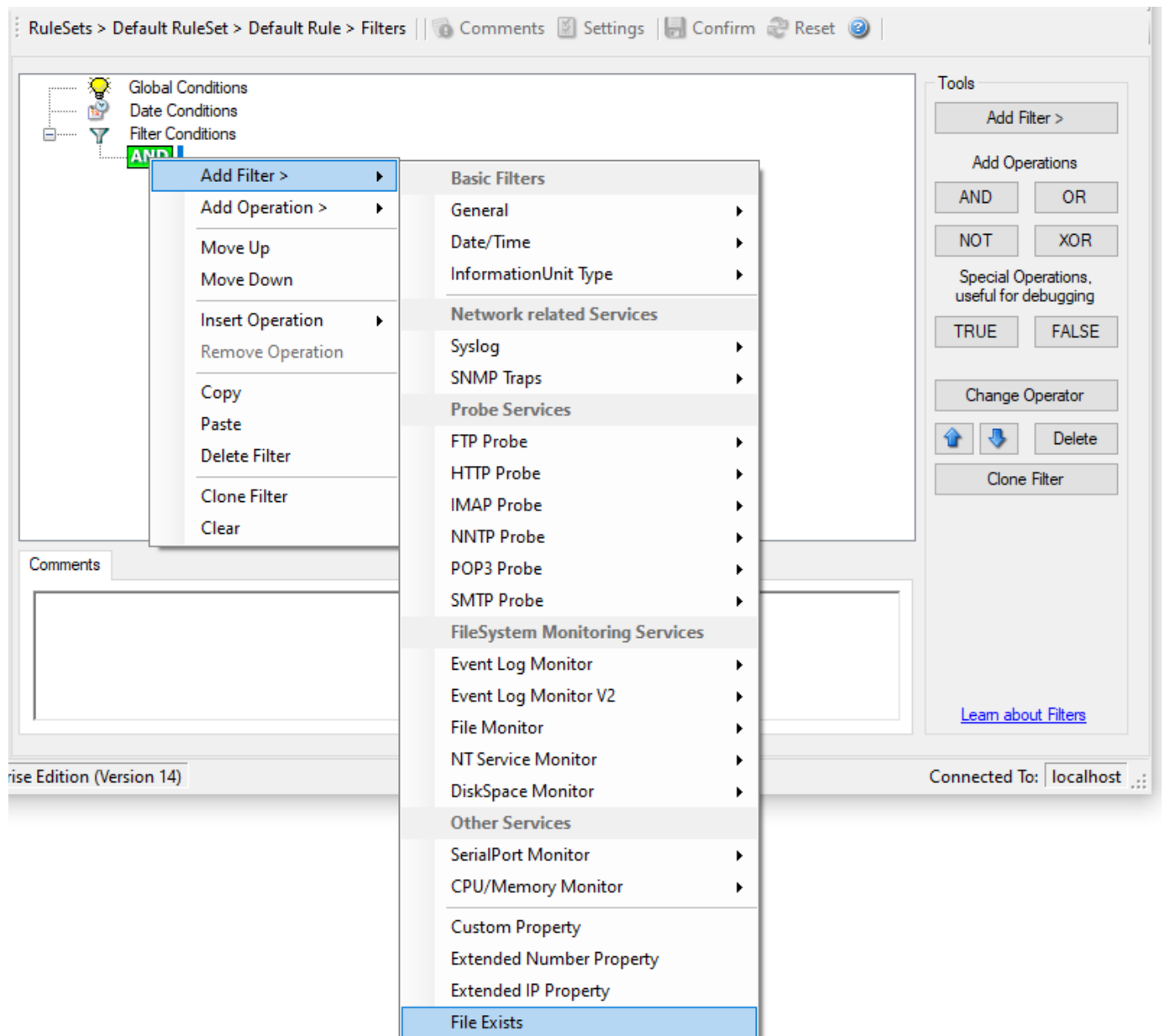
The screenshot shows the 'RuleSets > Syslog FW > Syslog UDP > Filters' configuration page. The main area displays a filter tree with 'Global Conditions', 'Date Conditions', and 'Filter Conditions'. Under 'Filter Conditions', there is an 'AND' operator connecting two 'EVAL' conditions: 'Extended IP: %source% > "172.16.0.110"' and 'Extended IP: %source% < "172.16.0.130"'. The right sidebar contains 'Tools' for adding filters and operations (AND, OR, NOT, XOR), special operations (TRUE, FALSE), and buttons for changing operators, deleting, and cloning filters. At the bottom, the 'Details' tab shows the selected filter's properties: Property Name: source, Compare Operation: >, and Set Property Value: 172.16.0.110.

- Filter Conditions - Filtering for an IP Range*

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

File Exists

Filter setting by string.



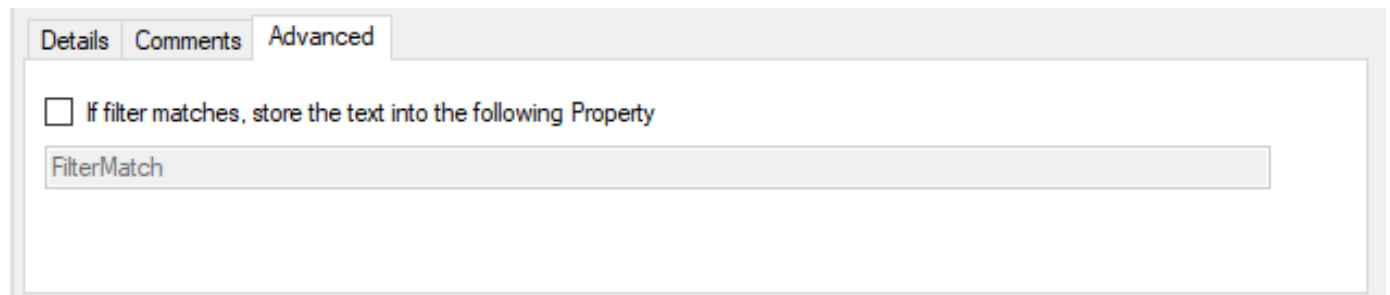
- Filter Conditions - File Exists*

File Exists

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

Store Filter Results

How to store Filter Results is described here.



- Filter Conditions - Store Filter Results*

Store Filter Results

If a filter matches, you can now store the result of the match into a custom property.

This custom property can be used in Actions later.

Actions

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

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

Storing Actions

ODBC Database Options

Use database logging to store messages into a database.

Database logging allows writing incoming events directly to any ODBC - compliant database (virtually any database system currently available for the Windows operating system supports ODBC). We support any database system that provides OLEDB or ODBC drivers. This includes Microsoft JET databases (as used by Microsoft Access), Microsoft SQL Server, MySQL, Oracle, PostgreSQL, Sybase, and many other database systems.

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

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

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

Connection Options

RuleSets > Default RuleSet > Default Rule > ODBC Database Enabled Comments Settings Confirm Reset ?

Connection Options

Configure DSN Verify Database Create Database

DSN

User-ID

Password ☒ Enable Password encryption

SQL Connection Timeout

SQL Options

Table Name

Statement Type

Output Encoding

☒ Insert NULLValue if string is empty

☒ Enable Detail Property Logging

Detaildata Tablename

Maximum value length (Bytes):

- Action - ODBC Database Connection*

Configure DSN

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

Note: The DSN must be a System DSN. **Verify Database** The configuration client will attempt to establish a database connection to your configured ODBC System DSN.

Create Database

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

DNS

File Configuration field:

szODBCDsn

Description:

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

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

User-ID

File Configuration field:

szODBCUid

Description:

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

Password

File Configuration field:

szODBCPwd

Description:

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

Enable Encryption

File Configuration field:

nODBCEnCryption

Description:

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

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

SQL Connection Timeout

File Configuration field:

nSQLConnectionTimeOut

Description:

Defines the Timeout for the connection.

SQL Options**Table Name****File Configuration field:**

szTableName

Description:

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

SQL Statement Type**File Configuration field:**

nSQLStatementType

Description:

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

Output Encoding**File Configuration field:**

nOutputEncoding

Description:

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

Insert NULL Value if string is empty**File Configuration field:**

nSQLConnectionTimeOut

Description:

This option inserts a NULL value, if a property is empty.

Enable Detail Property Logging**File Configuration field:**

nPropertiesTable

Description:

This option logs event properties other than the standard properties to the SystemEventProperties table. A single event can potentially have multiple properties, so selecting this option can result in multiple writes. With Syslog data, however, there are seldom any additional properties. They most often occur when you use the "Post Process" action to define your own properties. Additional properties are typically found in SETP received data originating from an Event Log Monitor, file monitor, or database monitor (plus other monitors, but these are the most prominent ones).

For example, with Event Log data received via SETP, these properties contain the actually Windows event properties and the event data. Please note that this does not apply to event log messages received via Syslog, because they are no native events but rather Syslog data.

Please make sure you actually need this before activating it. As a side note: some of the MonitorWare Console reports may need detail logging.

Detaildata Tablename**File Configuration field:**

szPropertiesTableName

Description:

Tablename for Detail Property Logging

Maximum value length (Bytes)**File Configuration field:**

nMaxValueLength

Description:

Maximum length in bytes for values stored in Detaildata table.

Datafields

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

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

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

Datafields			
	Fieldname	Fieldtype	Fieldcontent
	CurrUsage	int	▼ curusage
	CustomerID	int	▼ CustomerID
	DeviceReportedTime	DateTime UTC	▼ timereported
	EventBinaryData	text	▼ %bdata%
	EventCategory	int	▼ category
	EventID	int	▼ id
	EventLogType	varchar	▼ NTEventLogType
	EventSource	varchar	▼ sourceproc
	EventUser	varchar	▼ user

- Action - ODBC Database Datafields*

Fieldname**File Configuration field:**

szFieldName_[n]

Description:

The Fieldname is the database column name. It can be any field inside the table.

Fieldtype**File Configuration field:**

nFieldType_[n]

- 1 = varchar
- 2 = int
- 3 = text
- 4 = DateTime

Description:

Fieldtype is the data type of the database column. It must reflect the column type selected in the database. It must also be consistent in type with the actual property that must be stored. For example, an integer type property like the syslogpriority can be stored in a varchar column. A string data type like the syslogtag can - for obvious reasons - not be stored in an integer column.

Fieldcontent**File Configuration field:**

szFieldContent_[n]

Description:

Finally, the Fieldcontent is the event property. For a complete list of supported properties, see event properties

Action Queue Options

- Action - Send RELP Action Queue*

Use Diskqueue if connection to Syslog server fails**File Configuration field:**

nUseDiscQueue

Description:

Enable diskqueueing syslog messages after unexpected connection loss.

Split files if this size is reached**File Configuration field:**

nDiskQueueMaxFileSize

Description:

Files will be split until they reach the configured size in bytes. The maximum support file size is 10485760 bytes.

Diskqueue Directory**File Configuration field:**

szDiskQueueDirectory

Description:

The directory where the queue files will be generated in. The queuefiles will be generated with a dynamic UUID bound to the action configuration.

Waittime between connection tries**File Configuration fields:**

nDiskCacheWait

Description:

The minimum waittime until the Syslog Action retries to establish a connection to the Syslog server after failure.

Overrun Prevention Delay (ms)

File Configuration field:

nPreventOverrunDelay

Description:

When the Action is processing syslog cache files, an overrun prevention delay can be added to avoid flooding the target Syslog server.

Double wait time after each retry

File Configuration field:

bCacheWaittimeDoubling

Description:

If enabled, the configured waittime is doubled after each try.

Limit wait time doubling to

File Configuration field:

nCacheWaittimeDoublingTimes

Description:

How often the waittime is doubled after a failed connection try.

Enable random wait time delay

File Configuration field:

bCacheRandomDelay

Description:

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

Maximum random delay

File Configuration field:

nCacheRandomDelayTime

Description:

Maximum random delay time that will be added to the configured waittime if Enable random wait time delay is enabled.

OLEDB Database Action

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

This Action allows writing incoming events directly to any OLEDB - compliant database.

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

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

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

Connection Options

RuleSets > Default RuleSet > Default Rule > OLEDB Database Enabled Comments Settings Confirm Reset ?

Connection Options

Configure OLEDB Connection Verify Database Create Database

SQL Connection Timeout: 1 Minute

Provider:

Data Source:

Location:

Data Catalog:

Username:

Password: ☒ Encrypt password

SQL Options

Table Name: SystemEvents

Statement Type: CALL (MSSQLStored Procedure)

Output Encoding: System Default

☒ Enable Detail Property Logging

Detaildata Tablename: SystemEventsProperties

Maximum value length (Bytes): 512

- Action - OLEDB Database Connection*

Configure OLEDB Connection

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

Verify Database

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

Create Database

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

SQL Connection Timeout

File Configuration field:

nSQLConnectionTimeOut

Description:

Defines the Timeout for the connection

Provider

File Configuration field:

szProvider

Description:

OleDb Provider like SQL Server Client (SQLNCLI11.1). Should be filled automatically with Configure OLEDB Connection button.

Data Source

File Configuration field:

szDataSource

Description:

Data source is most often the server name or IP address like SERVERNAME\$SQLEXPRESS for example. Should be filled automatically with Configure OLEDB Connection button.

Location

File Configuration field:

szLocation

Description:

OLEDB Location. Should be filled automatically with Configure OLEDB Connection button.

Data Catalog

File Configuration field:

szDataCatalog

Description:

Is the database name in most cases. Should be filled automatically with Configure OLEDB Connection button.

Username

File Configuration field:

szUsername

Description:

Username used for authentication. Should be filled automatically with Configure OLEDB Connection button.

Password

File Configuration field:

szPassword

Description:

Password used for authentication. Should be filled automatically with Configure OLEDB Connection button.

Encrypt password

File Configuration field:

szPassword

Description:

Password used for authentication. Should be filled automatically with Configure OLEDB Connection button.

Table Name

File Configuration field:

szTableName

Description:

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

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

Statement Type**File Configuration field:**

nSQLStatementType

Description:

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

Output Encoding**File Configuration field:**

nOutputEncoding

Description:

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

Enable Detail Property Logging**File Configuration field:**

nPropertiesTable

Description:

This option logs event properties other than the standard properties to the SystemEventProperties table. A single event can potentially have multiple properties, so selecting this option can result in multiple writes. With Syslog data, however, there are seldom any additional properties. They most often occur when you use the "Post Process" action to define your own properties. Additional properties are typically found in SETP received data originating from an Event Log Monitor, file monitor, or database monitor (plus other monitors, but these are the most prominent ones).

For example, with Event Log data received via SETP, these properties contain the actually Windows event properties and the event data. Please note that this does not apply to event log messages received via Syslog, because they are no native events but rather Syslog data.

Please make sure you actually need this before activating it. As a side note, some of the MonitorWare Console reports may need detail logging.

Detailldata Tablename**File Configuration field:**

szPropertiesTableName

Description:

Tablename for Detail Property Logging

Maximum value length (Bytes)**File Configuration field:**

nMaxValueLength

Description:

Maximum length in bytes for values stored in Detailldata table.

Datafields

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

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

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

Datafields			
	Fieldname	Fieldtype	Fieldcontent
	CumUsage	int	cumusage
	CustomerID	int	CustomerID
	DeviceReportedTime	Date Time UTC	timereported
	EventBinaryData	text	%bdata%
	EventCategory	int	category
	EventID	int	id
	EventLogType	varchar	NTEventLogType
	EventSource	varchar	sourceproc
	EventUser	varchar	user

- Action - OLEDB Database Datafields*

Fieldname

File Configuration field:

szFieldName_[n]

Description:

The Fieldname is the database column name. It can be any field inside the table.

Fieldtype

File Configuration field:

nFieldType_[n]

- 1 = varchar
- 2 = int
- 3 = text
- 4 = DateTime

Description:

Fieldtype is the data type of the database column. It must reflect the column type selected in the database. It must also be consistent in type with the actual property that must be stored. For example, an integer type property like the syslogpriority can be stored in a varchar column. A string data type like the syslogtag can - for obvious reasons - not be stored in an integer column.

Fieldcontent

File Configuration field:

szFieldContent_[n]

Description:

Finally, the Fieldcontent is the event property. For a complete list of supported properties, see event properties

Action Queue Options

Connection Options
Action Queue Options

☐ Use Diskqueue if connection to Syslog Server fails

Split files if this size is reached
10485760

Diskqueue Directory
D:\cvsroot\adiscon-client\MWAgent\bin\Release
Browse

Waittime between connection tries
15 seconds

Overrun Prevention Delay (ms)
1
milliseconds

☐ Double wait time after each retry

Limit wait time doubling to
10

☐ Enable random wait time delay

Maximum random delay
5 seconds

- Action - Send RELP Action Queue*

Use Diskqueue if connection to Syslog server fails

File Configuration field:

nUseDiscQueue

Description:

Enable diskqueueing syslog messages after unexpected connection loss.

Split files if this size is reached

File Configuration field:

nDiskQueueMaxFileSize

Description:

Files will be split until they reach the configured size in bytes. The maximum support file size is 10485760 bytes.

Diskqueue Directory

File Configuration field:

szDiskQueueDirectory

Description:

The directory where the queue files will be generated in. The queuefiles will be generated with a dynamic UUID bound to the action configuration.

Waittime between connection tries

File Configuration fields:

nDiskCacheWait

Description:

The minimum waittime until the Syslog Action retries to establish a connection to the Syslog server after failure.

Overrun Prevention Delay (ms)

File Configuration field:

nPreventOverrunDelay

Description:

When the Action is processing syslog cache files, an overrun prevention delay can be added to avoid flooding the target Syslog server.

Double wait time after each retry

File Configuration field:

bCacheWaittimeDoubling

Description:

If enabled, the configured waittime is doubled after each try.

Limit wait time doubling to

File Configuration field:

nCacheWaittimeDoublingTimes

Description:

How often the waittime is doubled after a failed connection try.

Enable random wait time delay

File Configuration field:

bCacheRandomDelay

Description:

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

Maximum random delay

File Configuration field:

nCacheRandomDelayTime

Description:

Maximum random delay time that will be added to the configured waittime if Enable random wait time delay is enabled.

File Logging Options

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

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

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

The filename is build as follows:

`<FilePathName><FileName>-year-month-day.<FileExtension>`

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

RuleSets > Default RuleSet > Default Rule > File Logging Enabled Comments Settings Confirm Reset

Filename related options | File format | Post Processing

☐ Enable Property replacements in Filename

File Path Name Browse Insert

File Base Name Insert

File Extension

☒ Continuous Logging

☒ Create unique filenames

☐ Include Source in Filename

☐ Use UTC in Filename

☐ Segment files when the following filesize is reached (KB)

Segment Filesize (KB)

☐ Circular Logging

Number of Logfiles

Maximum Filesize (KB)

☐ Clear logfile instead of deleting (File will be reused)

File Handling Options

Output Encoding

Timeout until unused filehandles are closed

☐ Explicitly update create and modified file timestamp

- Action - File Logging Filename related*

Enable Property replacements in Filename

File Configuration field:

nEnablePropertyFileName

Description:

By activating this option, you can use properties within the file or pathname like %source% and all the others. For example: File Path Name can be F:\syslogs\%source% File Base Name can be IIS-%source%

If your source is 10.0.0.1, that writes the following file: F:\syslogs\10.0.0.1\IIS-10.0.0.1.log

The path f:\syslogs\10.0.0.1 was generated because the source property was used inside the path.

Please Note that you can use ANY property inside the path and base name. event properties are described in the property replacer section.

File Path Name

File Configuration field:

szFilePath

Description:

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

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

event properties are described in the property replacer section.

On network paths: The File Logging action can also work on network storages. There are two ways of storing log files in a network path.

1. Direct the action to a full UNC path. In this case, make sure the system account with which the service is running is able to access the network path or the service will fail to access with a permission error. Sample path: `\Hostname\folder1\folder2\`
2. Map the UNC path to a local drive letter in Windows. In this case, the path will look like a regular local path, but actually points to a network location. This requires a workaround, which is to run a scheduled task at system startup under Local System and perform a net use specifying the user and password of the share. Else, the service will not be able to access the mapped UNC path, because the mapping usually happens for interactive sessions only.

File Base Name

File Configuration field:

`szFileBaseName`

Description:

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

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

File Extension

File Configuration fields:

`szFileExtension`

Description:

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

Continuous Logging

Description

When enabled log files will not be overwritten, there is a single file with consistent file name. See below checkboxes to choose in which cases a new file should be created.

Create unique Filenames

File Configuration field:

`nUniqueFileName`

Description:

If checked, a unique file name is created for each day. This is done by adding the current date to the base name.

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

Include Source in Filename

File Configuration field:

`nIncludeSourceInFilename`

Description:

This works together with the "Create unique Filenames" setting. If checked, the file name generation explained above is modified. The source of the Syslog message is automatically added to the file name.

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

Use UTC in Filename**File Configuration field:**

nUseUTCInFileName

Description:

This works together with the “Create unique Filenames” setting. If unique names are to be created then select the “Use UTC in Filename” option, in this case the file name is generated on the basis of universal 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 7 pm New York time. If it were unchecked, the rollover would occur exactly at midnight New York time (5 am UTC).

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

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

Segment files when the following file size is reached (KB)**File Configuration field:**

nSegmentFileEnable

Description:

Files are segmented if the defined file size: Segment Filesize (KB) is reached. A sequence number is appended to the file name: _1 to _n.

Circular Logging**File Configuration field:**

nCircularLogging

Description:

If enabled, log files are created and overwritten in a cycle.

Number of Log Files**File Configuration field:**

nNumberOfLogfiles

Description:

Once the last log file is reached, circular logging begins and overwrites the first log file again. If set to 0, log files will not be rotated but can still be processed by Rotate Post Processing (for example compression or backup) along with the Rotate Conditions.

Maximum Filesize (KB)**File Configuration field:**

nMaxFileSize

Description:

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

Clear logfile instead of deleting (File will be reused)**File Configuration field:**

nReUseFile

Description:

This option causes the File Action to truncate the log file instead of deleting and recreating it.

File Handling Options

Output Encoding

File Configuration field:

nOutputEncoding

Description:

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

Timeout until unused filehandles are closed

File Configuration field:

nCleanFileHandlesTimeout

Description:

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

Explicitly update create and modified file Timestamp

File Configuration field:

nEnableUpdateFileTime

Description:

If the checkbox is not selected the operating system updates the timestamps for creating and modifying files. In cases where the filesystem does not do this reliably, the checkbox can be selected. Now the service itself updates the timestamps for creating and modifying files.

File Format

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

RuleSets > Default RuleSet > Default Rule > File Logging Enabled Comments Settings Confirm Reset

Filename related options | **File format** | Post Processing

☒ Adiscon

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

☐ Raw Syslog message
☐ Webtrends syslog compatible
☐ Custom format

Custom Line Format

%msg%%\$CRLF%

Insert

- Action - File Logging File Format*

Adiscon

Note

Any other format besides “Adiscon Default” are fixed formats. As such, if it is selected, all other formatting options do not apply and consequently are turned off.

The following options are possible:

Use XML to Report

File Configuration field:

nUseXMLtoReport

Description:

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

Use UTC for Timestamps

File Configuration field:

nUseUTCForTimestamps

Description:

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

Include <Fieldname>

File Configuration field:

- nFileDateTime
- nFileFacility
- nFilePriority
- nFileDateTimeReported
- nFileSource
- nIncludeMessage
- nIncludeRAWMessage

Description:

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

Please note the difference between the “Date and Time” and “Date and Time reported by Device”. Both are timestamps. Either both are written in local time or utc based on the “Use UTC for Timestamps” check box. However, “Date and Time” is the time when MonitorWare Agent received the message. Therefore, it is always a consistent value.

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

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

Raw Syslog message

The “Raw Syslog message” format writes raw Syslog format to the log file. That is, each line contains the Syslog message as of RFC 3164. No specific field processing or information adding is done. Some third party applications require that format.

Webtrends syslog compatible

The “WebTrends Syslog compatible” mimics the format that WebTrends applications expect. Please note that we only mimic the log file format. It is still the job of the reporting device (most notable firewall) to generate the correct WebTrends WELF format. The “WebTrends” format is supported because many customers would like to use MonitorWare Agent 3.0 enhanced features while still having the ability to work with WebTrends.

Custom format

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

Custom Line Format

File Configuration field:

szLineFormat

Description:

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

Post Processing

Filename related options
File format
Post Processing

☒ Enable Log Rotation

Max waittime for log rotation
15 seconds

Maximum number of rotated logfiles to keep
7

Rotate Conditions

☐ Rotate each time a file is closed

☒ Do not rotate files on shutdown

☐ Rotate if this filesize limit is being reached:

Filesize limit (KB)
4096

☐ Enable time based rotation

Rotate logfiles older than:
24 hours

☒ Enable rotation by time of the day

Rotate files at this time (hour:minute)
00:00

☒ Monday

☒ Tuesday

☒ Wednesday

☒ Thursday

☒ Friday

☒ Saturday

☒ Sunday

Rotate PostProcessing

☒ Compress file after log rotation

Compression Format
ZIP (.zip) Compression

Compression Level
Normal Compression

☒ Move file after log rotation

Target directory
C:\backup
Browse
Insert

- Action - File Logging Post Processing*

Enable Log Rotation

File Configuration field:

nCircularLogging

Description:

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

Maximum wait time for log rotation

File Configuration field:

nLogRotateMaxWait

Description:

Maximum Wait time when log rotation is processed within the Queue Engine.

Maximum number of rotated log files to keep

File Configuration field:

nNumberOfLogfiles

Description:

Once the last log file is reached, circular logging begins and overwrites the first log file again. If set to 0, log files will not be rotated but can still be processed by Rotate Post Processing (for example compression or backup) along with the Rotate Conditions.

Rotate Conditions

Rotate each time a file is closed

File Configuration field:

nLogRotateOnClose

Description:

When a file is closed (Timeout for example), log rotation will be done.

Do not rotate files on Shutdown

File Configuration field:

nLogDoNotRotateOnShutdown

Description:

Do not rotate log files if service is stopped even with “Rotate each time a file is closed” enabled.

Rotate if this filesize limit is being reached

File Configuration field:

nLogRotateOnSizeLimit

Description:

Enable log rotation if a configured file size is reached.

Filesize limit (KB)

File Configuration field:

nLogRotateSizeLimit

Description:

The actual file size in KB for “Rotate if this filesize limit is being reached”.

Enable time based rotation

File Configuration field:

nLogEnableRotateTimeout

Description:

Enable time based log rotation.

Rotate log files older than

File Configuration field:

nLogRotateTimeout

Description:

Sets the maximum file age before a logfile is being rotated when “Enable time based rotation” is enabled.

Enable rotation by time of the day

File Configuration field:

nLogEnableRotateTimeOfDay

Description:

Rotate this file at this time (hour:minute) and the checked day/days.

Rotate PostProcessing

Compress File After log rotation

File Configuration field:

nLogZipAfterRotate

Description:

Enable file compression after log rotation.

Compression Format

File Configuration field:

nLogZipAfterRotateFormat

Description:

It is possible to compress to ZIP or GZIP format.

Compression Level

File Configuration field:

nLogZipCompressionLevel

Description:

There are different levels that can be selected:

- Best Speed
- Low Compression
- Normal Compression
- Best Compression

Move file after log rotation

File Configuration field:

nLogMoveAfterRotate

Description:

Move logfile after rotation & compression.

Target directory

File Configuration field:

szLogMoveAfterRotatePath

Description:

Location where to move the logfile after rotation & compression.

forwarding actions

Event Log Options

This tab is used to configure the logging to the Windows Event Log. It is primarily included for legacy purposes.

- Action - EventLog*

Use logsource from service

File Configuration field:

bUseCustomEventLog = 0

Description:

Takes the service name as logsource for the log entry. This option is enabled by default.

Replace Event Log Source

File Configuration field:

bUseCustomEventLog = 1

Description:

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

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

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

Custom Event Log Source

File Configuration field:

szCustomSource

Description:

EventSource is now fully configurable with all possibilities the property engine gives you. Please note that content of this field can be configured. event properties are described in the property replacer section.

Enable custom Eventlog Channel

File Configuration field:

bUseCustomEventLog

Description:

If enabled, a custom event log channel will be used instead of application.

Custom Eventlog Channel

File Configuration field:

szCustomEventLog

Description:

The custom Eventlog channel to be used instead of application. Will be automatically created if the channel does not exist.

Use Custom Eventlog Type

File Configuration field:

nEventType

- 0 = EVENTLOG_SUCCESS (Information event)
- 1 = EVENTLOG_ERROR_TYPE (Error event)
- 2 = EVENTLOG_WARNING_TYPE (Warning event)
- 4 = EVENTLOG_INFORMATION_TYPE (Information event)
- 8 = EVENTLOG_AUDIT_SUCCESS (Success Audit event)
- 16 = EVENTLOG_AUDIT_FAILURE (Failure Audit event)

Description:

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

EventID

File Configuration field:

nEventID

Description:

The ID to be used when writing to the event log. Different IDs can be used to provide other processes with a consistent interface to specific messages. WinSyslog does not restrict the IDs that can be used. However, if an ID is written that is not registered with the operating system, Windows event viewer places an error message pointing this out before the actual message text. To avoid this text, event IDs 10,000 to 10,100 have been registered with the OS. We highly recommend that these IDs should be used for all custom messages. IDs below 10,000 should not be used as they might potentially interfere with events generated by MonitorWare Agent 3.0 itself.

Message to Log

File Configuration field:

szMessagecontent

Description:

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

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

Please note that the message content of the message field can be configured. event properties are described in the property replacer section.

Send Email

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

Mail Server Options

RuleSets > Default RuleSet > Default Rule > Send Email Enabled Comments Settings Confirm Reset ?

Mail Server Options Mail Format Options

Mailserver

Mailserver port

☐ Enable Backup Server, used if first Mailserver fails

Backup Mailserver

Backup Mailserver port

☐ Use SMTP Authentication

SMTP Username

SMTP Password

Session Timeout

☐ Use a secure connection (SSL) to the mail server

☐ Use STARTTLS SMTP Extension

☐ Use UTC Time in Date-Header

- Action - Send Email - Mail Server Options*

Mailserver

File Configuration field:

szSMTPServer

Description:

This is the Name or IP address of the mail server to be used for forwarding messages. You can either use an IPv4, an IPv6 Address, or a Hostname that resolves to an IPv4 or IPv6 Address. Please note that this server must be able to relay messages if the recipient is not hosted at this server. Be sure to contact your mail server's administrator if in doubt on this issue.

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

Mailserver port

File Configuration field:

nSMTPPort

Description:

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

Enable Backup Server, used if first Mailserver fails

File Configuration field:

nEnableBackupServer

Description:

When enabled, you can configure a second Mailserver that will be used if the regular Mailserver is not available/accessible.

Backup Mailserver

File Configuration field:

szSMTPServerBackup

Description:

In case that the connection to the main configured mail server cannot be established, the backup mail server is tried. Note that an error is only generated, if the connection to the backup server fails as well.

Backup Mailserver port

File Configuration field:

nSMTPPortBackup

Description:

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

Use SMTP Authentication

File Configuration field:

nUseSMTPAuth

Description:

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

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

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

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

Session Timeout

File Configuration field:

nTimeoutValue

Description:

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

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

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

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

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

Use a secure connection (SSL) to the mail server

File Configuration field:

nUseSSL

Description:

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

Use STARTTLS SMTP Extension

File Configuration field:

nUseUTCTimeStamp

Description:

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

Use UTC Time in Date-Header

File Configuration field:

nUseUTCTimeStamp

Description:

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

Mail Format Options

RuleSets > Default RuleSet > Default Rule > Send Email Enabled Comments Settings Confirm Reset ?

Mail Server Options **Mail Format Options**

Sender Emailaddress:

Recipient Emailaddress:

☐ Use legacy subject line processing

Subject: Insert

Mail Priority: Normal Priority ▼

Mail Message Format:

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

Insert

Output Encoding: System Default ▼

☐ Use XML to Report

- Action - Send Email - Mail Format Options*

Sender email address

File Configuration field:

szSMTPSender

Description:

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 email address

File Configuration field:

szSMTPRecipient

Description:

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

Use legacy subject line processing

File Configuration field:

nUseLegacySubjectProcessing

Description:

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

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

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

`%f` Numeric facility code of the received message

`%p` Numeric priority code of the received message

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

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

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

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

Subject

File Configuration field:

szSMTPSubject

Description:

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

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

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

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

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

Mail Priority

File Configuration field:

nMailPriority

- 0 = low
- 1 = Default
- 2 = High

Description:

Here you can adjust the priority with which the mail will be sent. You can choose between “low”, “normal”, and “high” priority. With this you can give your setup some complexity, being able to send some events as “important” and others with less importance.

Mail Message Format

File Configuration field:

szSMTPBody

Description:

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

Output Encoding

File Configuration field:

nOutputEncoding

Description:

Determines the character encoding mode.

Use XML to Report

File Configuration field:

nUseXMLtoReport

Description:

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

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

Net Send

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”.

- Action - Net Send*

Target Machine

File Configuration field:

szTarget

Description:

This is the Windows user name of the intended recipient, a NETBIOS machine name, or even an IP address (in the form of 10.1.1.1). You can either use an IPv4, an IPv6 Address or a Hostname that resolves to an IPv4 or IPv6 Address.

Message to send

File Configuration field:

szMessage

Description:

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

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

Send to Communications Port

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

RuleSets > Default RuleSet > Default Rule > Send to Communication Port Enabled Comments Settings Confirm

Timeout Limit: 1 Minute

Send message to this communication port: COM1:

Port Settings

Bits per second: 57600

Data bits: 8

Parity: No Parity

Stop bits: 1 Stop bit

DTR Control Flow: DTR Control Disable

RTS Control Flow: RTS Control Disable

Message to send: %msg%

Insert

- Action - Send to Communication Port*

Timeout Limit

File Configuration field:

nTimeOutLimit

Description:

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

Send message to this communication port

File Configuration field:

szPortName

Description:

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

1. MSFAX
2. COM1
3. COM2
4. COM3
5. COM4
6. FILE
7. LPT1
8. LPT2
9. LPT3
10. AVMISDN1
-
11. AVMISDN2
-

- 12 AVMISDN3
- .
- 13 AVMISDN4
- .
- 14 AVMISDN5
- .
- 15 AVMISDN6
- .
- 16 AVMISDN7
- .
- 17 AVMISDN8
- .
- 18 AVMISDN9
- .

Port Settings

File Configuration field:

szPortSettings

Description:

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

Bits per Seconds

File Configuration field:

nBps

Description:

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

Databits

File Configuration field:

nDatabits

Description:

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

Parity

File Configuration field:

nParity

Description:

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

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

Stop bits

File Configuration field:

nStopbits

Description:

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

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

DTR Control Flow

File Configuration field:

nDtsControl

Description:

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

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

RTS Control Flow

File Configuration field:

nRtsControl

Description:

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

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

Message to Send

File Configuration field:

szMessage

Description:

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

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

Send MSQueue

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

RuleSets > Default RuleSet > Default Rule > Send MSQueue Enabled Comments Settings Confirm Reset

Server Computename/IP	<input type="text" value="localhost"/>	
Queue name	<input type="text"/>	
Queue Priority	<input type="text" value="3"/>	
Queue Message Label	<input type="text" value="Message"/>	<input type="button" value="Insert"/>
Queue Message Body	<input type="text" value="%msg%"/>	<input type="button" value="Insert"/>

- Action - Send MSQueue*

Server Computename/IP

File Configuration field:

szComputerName

Description:

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

Queue name

File Configuration field:

szQueueName

Description:

Specify the Queue name into which you want to write.

Queue Priority

File Configuration field:

nMessagePriority

Description:

Configure or set the priority property here.

Queue Message Label

File Configuration field:

szQueueLabel

Description:

Sets the Label text of a queue item.

Queue Message Body

File Configuration field:

szQueueBody

Description:

The text here will be set to the body of a queue item.

Send RELP

This action is roughly equivalent to the “send syslog” action, except that it utilizes the new reliable event logging protocol (RELP) for message transmission. It can only be used together with a RELP-enabled receiver but then provides enhance reliability in the communications process.

Note that for reliable operation where messages should be preserved over a service shutdown, queue cache mode must be activated. This is because RELP guards only the transmission channel, but not local processing.

- Action - Send RELP General*

RELP Servername

File Configuration field:

szSelpSendServer

Description:

This is the name or IP address of the system to which RELP messages should be sent to. You can either use an IPv4, an IPv6 Address, or a Hostname that resolves to an IPv4 or IPv6 Address.

RELP Port

File Configuration field:

nSelpSendPort

Description:

The remote port on the RELP server to report to. If in doubt, please leave it at the default value of 20514, which is typically the RELP port. Different values are only required for special setups, for example in security sensitive areas.

Instead of the port number, a service name can be used. If so, that name is looked up via the socket service database functions.

Session Timeout

File Configuration field:

nTimeOutSession

Description:

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

Send / Receive Timeout

File Configuration field:

nSendTimeOut

Description:

The maximum time a server waits for a response of a remote server. When the timeout expires without receiving a response, the connection is broken and (based on rule settings) being reestablished. This can be a useful option if the remote system drops connections for whatever reason AND the sender system is not notified about this (which, for example, can happen due to some firewall configurations).

Output Encoding

File Configuration field:

nOutputEncoding

Description:

Allows you to specify the character encoding for messages sent to the RELP server. The default setting is "System Default", which uses the system's default character encoding. Other common options include UTF-8, ASCII, and other standard encodings.

Message format

File Configuration field:

szMessage

Description:

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

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

Enable SSL / TLS Encryption

File Configuration field:

nUseSSL

Description:

If this option is enabled, the action will use SSL/TLS encryption for secure communication with the RELP server. When disabled, messages will be sent unencrypted. Note that if this option is enabled, the action will not be able to talk to a NON-SSL secured server.

TLS Mode

File Configuration field:

nTLSMode

Description:

Anonymous Authentication Default option. This means that a default certificate will be used.

Use Certificate If this option is enable, you can specify your own certificate. For further authentication solutions, you will need to create your own certificates using OpenSSL Tools for example.

Select common CA PEM

File Configuration field:

szTLSCAFile

Description:

Select the certificate from the common Certificate Authority (CA). The RELP Receiver should use the same CA.

Select Certificate PEM

File Configuration field:

szTLSCertFile

Description:

Select the client certificate (PEM Format).

Select Key PEM**File Configuration field:**

szTLSKeyFile

Description:

Select the keyfile for the client certificate (PEM Format).

Action Queue Options

RuleSets > Default RuleSet > Default Rule > Send RELP Enabled Comments Settings Confirm Reset ?

General Options **Action Queue Options**

☐ Use Diskqueue if connection to Syslog Server fails

Split files if this size is reached

szDiskQueueDirectory

Waittime between connection tries

Overrun Prevention Delay (ms) milliseconds

☐ Double wait time after each retry

Limit wait time doubling to

☐ Enable random wait time delay

Maximum random delay

- Action - Send RELP Action Queue*

Use Diskqueue if connection to Syslog server fails**File Configuration field:**

nUseDiscQueue

Description:

Enable diskqueuing syslog messages after unexpected connection loss.

Split files if this size is reached**File Configuration field:**

nDiskQueueMaxFileSize

Description:

Files will be split until they reach the configured size in bytes. The maximum support file size is 10485760 bytes.

Diskqueue Directory**File Configuration field:**

szDiskQueueDirectory

Description:

The directory where the queue files will be generated in. The queuefiles will be generated with a dynamic UUID bound to the action configuration.

Waittime between connection tries

File Configuration fields:

nDiskCacheWait

Description:

The minimum waittime until the Syslog Action retries to establish a connection to the Syslog server after failure.

Overrun Prevention Delay (ms)

File Configuration field:

nPreventOverrunDelay

Description:

When the Action is processing syslog cache files, an overrun prevention delay can be added to avoid flooding the target Syslog server.

Double wait time after each retry

File Configuration field:

bCacheWaittimeDoubling

Description:

If enabled, the configured waittime is doubled after each try.

Limit wait time doubling to

File Configuration field:

nCacheWaittimeDoublingTimes

Description:

How often the waittime is doubled after a failed connection try.

Enable random wait time delay

File Configuration field:

bCacheRandomDelay

Description:

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

Maximum random delay

File Configuration field:

nCacheRandomDelayTime

Description:

Maximum random delay time that will be added to the configured waittime if Enable random wait time delay is enabled.

Send SETP

With the "Send SETP" action, messages can be sent to a SETP server.

RuleSets > Default RuleSet > Default Rule > Send SETP Enabled Comments Settings Confirm Reset

General Options | Action Queue Options

Servename

Default SETP Port

☐ Enable SSL / TLS Encryption. Note if this Option is enabled, this action will not be able to connect to NON-SSL SETP Servers.

☐ Use zlib Compression to compress the data

Compression Level Best Compression

Timeout Options

Session Timeout

Connection Timeout

Send / Receive Timeout

- Action - Send SETP General*

Servename

File Configuration field:

szServer

Description:

The MonitorWare Agent sends setp to the server/listener under this name. You can either use an IPv4, an IPv6 Address, or a Hostname that resolves to an IPv4 or IPv6 Address.

Default SETP Port

File Configuration field:

nMIAPSendPort

Description:

The Send setp sends outgoing requests on this port. The default value is 5432. Set the port to 0 to use the system-supplied default value (which defaults to 5432 if not modified by a system administrator).

Instead of the port number, a service name can be used. If so, that name is looked up via the socket service database functions. The lookup is for protocol TCP.

Please Note: The SETP port configured here must match the port configured at the listener side (i.e. MonitorWare Agent or WinSyslog Enterprise edition). If they do not match, a Send SETP session cannot be initiated. The rule engine will log this to the Windows Event Log.

Enable SSL / TLS Encryption

File Configuration field:

nUseSSL

Description:

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

Use zlib Compression to compress the data

File Configuration field:

nZlibComp

Description:

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

Compression Level

File Configuration field:

nCompLevel

- 1 = Best Speed
- 3 = Low Compression
- 6 = Normal Compression
- 9 = Best Compression

Description:

Higher level results in better compression but slower performance.

Session Timeout

File Configuration field:

nTimeOutSession

Description:

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

Connection Timeout

File Configuration field:

nConnectTimeOut

Description:

Maximum time a connection can take to connect or disconnect.

Send / Receive Timeout

File Configuration field:

nSendRecvTimeOut

Description:

When sending or receiving data, this timeout applies.

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

Action Queue Options

RuleSets > Default RuleSet > Default Rule > Send SETP Enabled Comments Settings Confirm Reset ?

General Options Action Queue Options

☒ Use Diskqueue if connection to Syslog Server fails

Split files if this size is reached

szDiskQueueDirectory Browse

Waittime between connection tries

Overrun Prevention Delay (ms) milliseconds

☒ Double wait time after each retry

Limit wait time doubling to

☒ Enable random wait time delay

Maximum random delay

- Action - Send SETP Action Queue*

Use Diskqueue if connection to Syslog server fails

File Configuration field:

nUseDiscQueue

Description:

Enable diskqueuing syslog messages after unexpected connection loss.

Split files if this size is reached

File Configuration field:

nDiskQueueMaxFileSize

Description:

Files will be split until they reach the configured size in bytes. The maximum support file size is 10485760 bytes.

Diskqueue Directory

File Configuration field:

szDiskQueueDirectory

Description:

The directory where the queue files will be generated in. The queuefiles will be generated with a dynamic UUID bound to the action configuration.

Waittime between connection tries

File Configuration fields:

nDiskCacheWait

Description:

The minimum waittime until the Syslog Action retries to establish a connection to the Syslog server after failure.

Overrun Prevention Delay (ms)

File Configuration field:

nPreventOverrunDelay

Description:

When the Action is processing syslog cache files, an overrun prevention delay can be added to avoid flooding the target Syslog server.

Double wait time after each retry

File Configuration field:

bCacheWaittimeDoubling

Description:

If enabled, the configured waittime is doubled after each try.

Limit wait time doubling to

File Configuration field:

nCacheWaittimeDoublingTimes

Description:

How often the waittime is doubled after a failed connection try.

Enable random wait time delay

File Configuration field:

bCacheRandomDelay

Description:

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

Maximum random delay

File Configuration field:

nCacheRandomDelayTime

Description:

Maximum random delay time that will be added to the configured waittime if Enable random wait time delay is enabled.

Send SNMP Trap

General Options

RuleSets > Default RuleSet > Default Rule > Send SNMP Trap Enabled Comments Settings Confirm Reset ?

SNMP Options Action Queue Options

General SNMP Options

Internet Protocoltype IPv4

Protocol Type UDP

SNMP Server 127.0.0.1

SNMP Port 162

Community public

Output Encoding System Default

☐ SNMP Version 1 Only

Enterprise OID .1.3.6.1.4.1.3.1.1 Browse

Generic Name 0 - Cold Start

Specific Type 0

Agent IP Address %source%

☒ SNMP Version 2c Only

Trap OID .1.3.6.1.4.1.19406.1.2.2 Browse

SNMP Variables

	Variable OID	Variable Type	Variable Value
▶	.1.3.6.1.4.1.19406.1.1.1.7	Octet String	%msg%
*	*Enter value for Variable OID*	Octet String	*Enter value for Variable Value*

- Action - Send SNMP Trap Options*

SNMP Version

You can choose between SNMP Version 1 Only and SNMP Version 2c Only. Both options have different SNMP related configuration properties which need to be configured.

Internet Protocoltype

File Configuration field:

nlnetType

Description:

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

Protocol Type

File Configuration field:

nProtocolType

Description:

You can select to listen on UDP or TCP protocol for SNMP Traps.

SNMP Server

File Configuration field:

szAgent

Description:

Specify the agent that has to receive the SNMP trap. Please note that specifying a host name can cause the SMTP probe to fail if DNS name resolution fails (for example due to a failing DNS server). To avoid this, specify an IP address. You can either use an IPv4, an IPv6 Address or a Hostname that resolves to an IPv4 or IPv6 Address.

SNMP Port

File Configuration field:

nPort

Description:

This port is to be probed. Please see your server's reference for the actual value to use. For example, mail servers typically listen to port 25 and web servers to 80.

Community

File Configuration field:

szCommunity

Description:

Specify the SNMP community to which the messages belong too.

SNMP V1 Specific Parameters

Under this group box you can see the parameters related to SNMP version 1.

Enterprise OID

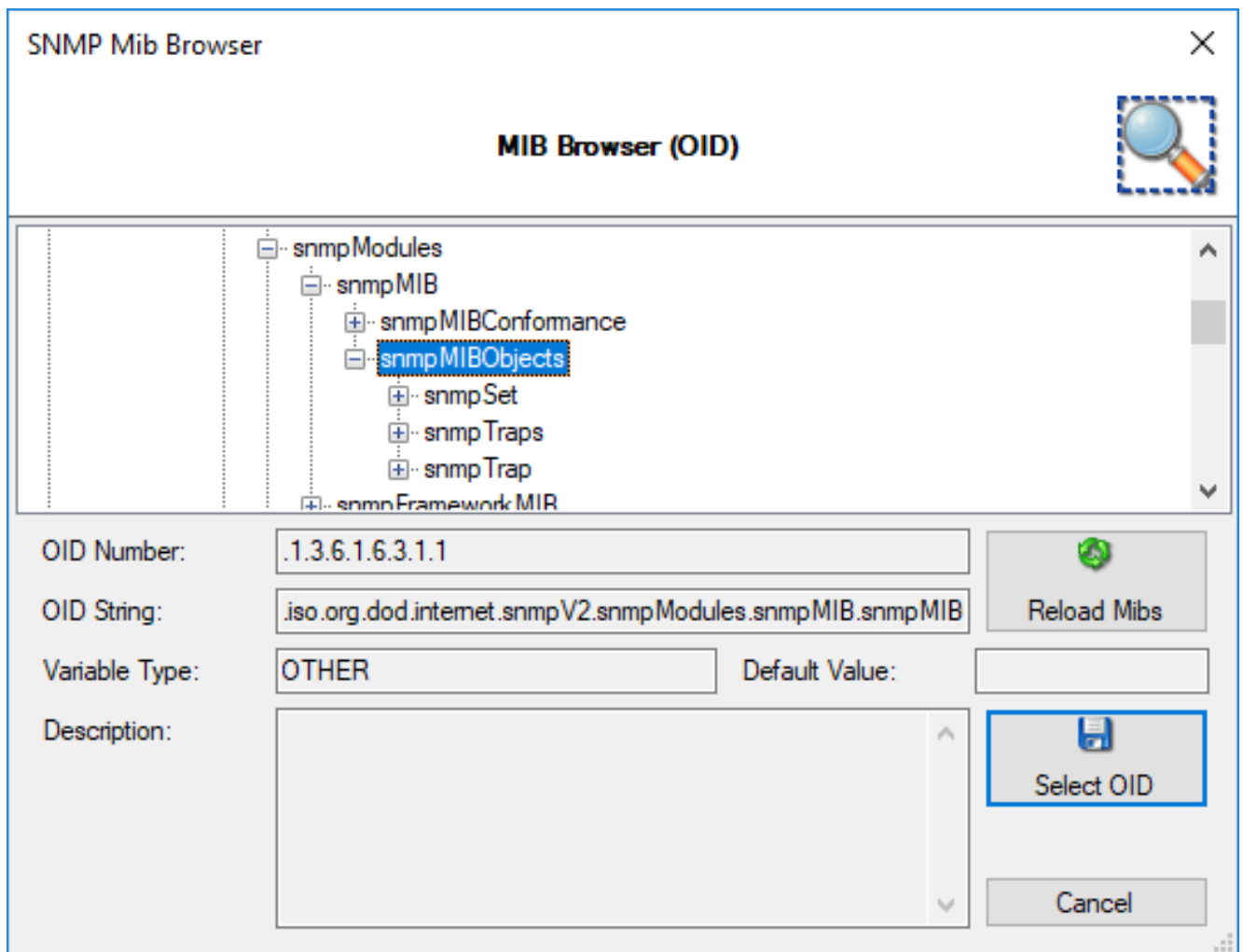
File Configuration field:

nInetType

Description:

It is also an optional value which can be used to specify a UserAgent that is send in the HTTP header.

Specify the enterprise object ID here. You can use Browse option to select your OID. If you click the Browse link, the screen similar to shown below is appeared:



MIB Browser

You can select your MIB here.

Generic Name

File Configuration field:

nGenericName

- 0 - Cold Start
- 1 - Warm Start
- 2 - Link Down
- 3 - Link Up
- 4 - Authentication Failure
- 5 - EGP Neighbor Loss
- 6 - Enterprise Specific

Description:

You can specify the generic name of the trap which can be one of these: coldStart(0), warmStart(1), linkDown(2), linkUp(3), authenticationFailure(4), egpNeighborLoss(5), or enterpriseSpecific(6).

Specific Type

File Configuration field:

nSpecificType

Description:

You can define an additional code for the trap. It is also an Integer value.

Agent IP Address

File Configuration field:

szAgentIP

Description:

The SNMPv1 Agent Address field can be set to other IP Addresses here. Hostnames will automatically be resolved if possible. By default we are using the %source% property.

SNMP Variables

These are the variables to send in the SNMP Trap. If you know the trap codes, you can enter them manually, otherwise use the built-in SNMP MIB Browser.

Variable OID

File Configuration field:

szVariableOID_[n]

Description:

OID of the SNMP Trap. Use the built-in SNMP MIB Browser for a list of known and available OIDs.

Variable Type

File Configuration field:

nVariableType_[n]

- 1 = TYPE_OBJID
- 2 = TYPE_OCTETSTR
- 3 = TYPE_INTEGER
- 5 = TYPE_IPADDR
- 6 = TYPE_COUNTER
- 7 = TYPE_GAUGE
- 8 = TYPE_TIMETICKS
- 12 = TYPE_BITSTRING
- 14 = TYPE_UIINTEGER
- 15 = TYPE_UNSIGNED32
- 16 = TYPE_INTEGER32

Description:

The variable type of the variable, usually OCTETSTRING or INTEGER. Depending on this type, the Variable value needs to be formatted correctly (Like for the type IPADDR).

Variable Value

File Configuration field:

szVariableValue_[n]

Description:

The value of the Variable. It needs to be formatted depending on the variable type.

Please Note:

The “Send SNMP Trap”-Action is capable of sending all kinds of Traps. You can choose the whole variety of the MonitorWare Products’ Properties as a value for the messages. With that, you can send SNMP Traps to the Windows internal SNMP Agent or any other device that is able to receive SNMP Traps. Of course you

have full enterprise support, too. This gives you the possibility to involve every machine on your network into your security plan or whatever purpose it should serve.

Action Queue Options

RuleSets > Default RuleSet > Default Rule > Send SNMP Trap Enabled Comments Settings Confirm Reset ?

SNMP Options **Action Queue Options**

☐ Use Diskqueue if connection to Syslog Server fails

Split files if this size is reached

szDiskQueueDirectory Browse

Waittime between connection tries

Overrun Prevention Delay (ms) milliseconds

☐ Double wait time after each retry

Limit wait time doubling to

☐ Enable random wait time delay

Maximum random delay

- Action - Send SNMPT Trap Action Queue*

Use Diskqueue if connection to Syslog server fails

File Configuration field:

nUseDiscQueue

Description:

Enable diskqueuing syslog messages after unexpected connection loss.

Split files if this size is reached

File Configuration field:

nDiskQueueMaxFileSize

Description:

Files will be split until they reach the configured size in bytes. The maximum support file size is 10485760 bytes.

Diskqueue Directory

File Configuration field:

szDiskQueueDirectory

Description:

The directory where the queue files will be generated in. The queuefiles will be generated with a dynamic UUID bound to the action configuration.

Waittime between connection tries

File Configuration fields:

nDiskCacheWait

Description:

The minimum waittime until the Syslog Action retries to establish a connection to the Syslog server after failure.

Overflow Prevention Delay (ms)**File Configuration field:**

nPreventOverflowDelay

Description:

When the Action is processing syslog cache files, an overflow prevention delay can be added to avoid flooding the target Syslog server.

Double wait time after each retry**File Configuration field:**

bCacheWaittimeDoubling

Description:

If enabled, the configured waittime is doubled after each try.

Limit wait time doubling to**File Configuration field:**

nCacheWaittimeDoublingTimes

Description:

How often the waittime is doubled after a failed connection try.

Enable random wait time delay**File Configuration field:**

bCacheRandomDelay

Description:

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

Maximum random delay**File Configuration field:**

nCacheRandomDelayTime

Description:

Maximum random delay time that will be added to the configured waittime if Enable random wait time delay is enabled.

Syslog Forwarding**Protocol Type**

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

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

TCP mode comes in three flavors. This stems back to the fact that transmission of syslog messages via plain TCP is not yet officially standardized (and it is doubtful if it ever will be). However, it is the most relevant and most widely implemented reliable transmission mode for syslog. It is a kind of unwritten industry standard. We support three

different transmission modes offering the greatest compatibility with all existing implementations. The mode “TCP (one message per connection)” is a compatibility mode for Adiscon servers that are older than roughly June 2006. It may also be required for some other vendors. We recommend not to use this setting, except when needed. “TCP (persistent connection)” sends multiple messages over a single connection, which is held open for an extended period of time. This mode is compatible with almost all implementations and offers good performance. Some issues may occur if control characters are present in the syslog message, which typically should not happen. The mode “TCP (octet-count based framing)” implements algorithms of an IETF standard RFC 6587. It also uses a persistent connection. This mode is reliable and also deals with embedded control characters very well. This standard is now widely supported by modern syslog receivers and implementations.

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

Syslog Target Options

Protocol Type
UDP

Syslog Target Options
Syslog Message Options
Network related Options

Syslog Send mode

☒ Use single syslog server with optional backup server

Syslog Receiver Options

Syslog Server

Syslog Port

514

☐ Use this backup syslog server if first one fails.

Backup Syslog Server

Backup Syslog Port

514

☐ Use round robin (multiple syslog servers)

Amount of messages send to each syslog server before load balancing

1000

Syslog Servers

	Syslog Server	Syslog Port
*	*Enter value for Syslog Server*	*Enter numvalue for Syslog Port*

- Action - Forward Syslog Target Options*

Syslog Send mode

File Configuration field:

nSendMode

Description

The Sendmode has been added since 2018 into all products supporting the forward syslog action. There are two options available.

Use single Syslog server with optional backup server This is the classic syslog send mode which uses a primary Syslog server and a secondary backup Syslog server if configured.

Use round robin (multiple syslog servers) This new method allows you to configure multiple targets that will be used one by one after a configured amount of messages has been sent to each target.

Syslog server (Syslog Send mode)**File Configuration field:**

szSyslogSendServer

Description:

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

Syslog Port (Syslog Send mode)**File Configuration field:**

nSyslogSendPort

Description:

The remote port on the Syslog server to report to. If in doubt, please leave it at the default value of 514, which is typically the Syslog port. Different values are only required for special setups, for example in security sensitive areas. Set the port to 0 to use the system-supplied default value (which defaults to 514 on almost all systems).

Instead of the port number, a service name can be used. If so, that name is looked up via the socket service database functions.

Use this backup Syslog server if first one fails**File Configuration field:**

nEnableBackupServer

Description:

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

Use round robin (multiple Syslog server)**Amount of messages send to each Syslog server before load balancing****File Configuration field:**

nRoundRobinMsgCount

Description:

When using round robin mode, this is the amount of messages to be sent to each configured Syslog server.

Syslog server (Round robin mode)**File Configuration field:**

szSyslogServer_[n]

Description:

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

Syslog Port (Round robin mode)

File Configuration field:

nSyslogPort_[n]

Description:

The remote port on the Syslog server to report to. If in doubt, please leave it at the default value of 514, which is typically the Syslog port. Different values are only required for special setups, for example in security sensitive areas. Set the port to 0 to use the system-supplied default value (which defaults to 514 on almost all systems).

Instead of the port number, a service name can be used. If so, that name is looked up via the socket service database functions.

Syslog Message Options

- Action - Forward Syslog - Message Options*

Syslog processing

File Configuration field:

bProcessDuringRelay

- 0 = Disable processing, forward as it is
- 1 = RFC3164 Header - Use legacy RFC 3164 processing
- 2 = RFC5424 Header - Use RFC 5424 processing (recommended)
- 3 = Custom Syslog Header

Description:

With this settings you can assign how your syslog messages will be processed.

For processing syslog you can choose out of four different options. You can use rfc3164 or RFC5424 (recommended) which is the current syslog standard, you are able to customize the syslog header or you do not process your syslog and forwards it as it is.

Use Custom Syslog Header

File Configuration field:

szCustomSyslogHeader

Description:

In this field you can specify the contents of your syslog header. This option is only available when you choose "Use Custom Syslog Header" in the Syslog Processing menu. The contents can be either a fixed message part which you can write into the field yourself or you use properties as dynamic content. By default the Header field is filled with the content of the RFC 5424 header.

Please note that the header content of the Header field can be configured. event properties are described in the property replacer section.

Output Encoding

File Configuration field:

nOutputEncoding

Description:

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

Include UTF8 BOM in message

File Configuration field:

nProtocolType

Description:

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

Use XML to Report

File Configuration field:

bReportInXML

Description:

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

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

Forward as MonitorWare Agent XML Representation Code

File Configuration field:

nForwardIUT

Description:

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

Use CEE enhanced Syslog Format

File Configuration field:

nReportInJSON

Description:

If enabled, the CEE enhanced Syslog format will be used. All useful properties will be included in a JSON Stream. The message itself can be included as well, see the "Include message property in CEE Format" option. Here is a sample how the format looks like for a security Eventlog message:

```
@cee: {"source": "machine.local", "nteventlogtype": "Security", "sourceproc": "Microsoft-Windows-Security-Auditing", "id": "4648", "categoryid": "12544", "category": "12544", "keywordid": "0x8020000000000000", "user": "N\\A", "SubjectUserSid": "S-1-5-11-22222222-33333333-44444444-5555", "SubjectUserName": "User", "SubjectDomainName": "DOMAIN", "SubjectLogonId": "0x5efdd", "LogonGuid": "{00000000-0000-0000-0000-000000000000}", "TargetUserName": "Administrator", "TargetDomainName": "DOMAIN", "TargetLogonGuid": "{00000000-0000-0000-0000-000000000000}", "TargetServerName": "servername", "TargetInfo": "servername", "ProcessId": "0x76c", "ProcessName": "C:\\Windows\\System32\\spoolsv.exe", "IpAddress": "-", "IpPort": "-", "catname": "Logon", "keyword": "Audit Success", "level": "Information", }
```

Additionally to this format you can set: Include message property in CEE Format.

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

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

Include message property in CEE Format

Description

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

Message Format

File Configuration field:

szMessageFormat

Description:

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

Add Syslog Source when forwarding to other Syslog servers

File Configuration field:

nSyslogInsertSource

Description:

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

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

Use zLib Compression to compress the data

File Configuration field:

nUseCompression

Description:

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

Compression Level

File Configuration field:

nCompressionLevel

- 1 = Best Speed
- 3 = Low Compression
- 6 = Normal Compression
- 9 = Best Compression (default)

Description:

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

Note on Using Syslog Compression

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

Please note that compression over TCP connections requires a special transfer mode. This mode uses OpenSSL TLS Implementation 3.x for secure transmission. TLS compression is not implemented; instead, the system uses standard OpenSSL compression mechanisms.

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

Overwrite Syslog Properties

Syslog Facility

File Configuration field:

nSyslogFacility

Description:

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

Syslog Priority

File Configuration field:

nSyslogPriority

Description:

When configured, will overwrite the Syslog Priority with the configured value.

SSL/TLS related Options

[Syslog Target Options](#)
[Syslog Message Options](#)
[Network related Options](#)
[SSL/TLS related Options](#)
[Action Queue Options](#)

☒ Enable SSL / TLS Encryption. Note if this Option is enabled, this action will not be able to connect to NON-SSL Syslog Servers.

TLS Mode: Anonymous authentication

Select common CA PEM: [Browse](#)


Select Certificate PEM: [Browse](#)

Select Key PEM: [Browse](#)

Advanced TLS Options

☐ Allow SSL v3 (insecure)
☐ Allow TLS v1.0 (insecure)
☒ Allow TLS v1.1
☒ Allow TLS v1.2

☐ Use OpenSSL configuration commands

 By enabling this option, you can set OpenSSL configuration commands directly. For more informations on available configuration parameters for each command type, visit this page: https://www.openssl.org/docs/man1.0.2/ssl/SSL_CONF_cmd.html

Configuration commands list

	Command Type	Command Value
*	Protocol	ALL,-SSLv2,-SSLv3,-TLSv1,-TLSv1.1

- Action - Forward Syslog SSL/TLS related Options*

Enable SSL / TLS Encryption

File Configuration field:

nUseSSL

Description:

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

TLS Mode

File Configuration field:

nTLSMode

Description:

Anonymous Authentication

Default option. This means that a default certificate will be used.

Use Certificate

If this option is enable, you can specify your own certificate. For further authentication solutions, you will need to create your own certificates using OpenSSL Tools for example.

Select common CA PEM

File Configuration field:

szTLSCAFile

Description:

Select the certificate from the common Certificate Authority (CA). The syslog receiver should use the same CA.

Select Certificate PEM

File Configuration field:

szTLSCertFile

Description:

Select the client certificate (PEM Format).

Select Key PEM

File Configuration field:

szTLSKeyFile

Description:

Select the keyfile for the client certificate (PEM Format).

Allow SSL v3

File Configuration field:

nTLSAllowSSLv3

Description:

This option enables insecure protocol method SSLv3. We recommend NOT enabling this option as SSLv3 is considered broken.

Allow SSL v1.0

File Configuration field:

nTLSAllowTLS10

Description:

This option enables insecure protocol method TLSv1. We recommend NOT enabling this option as TLSv1 is considered broken.

Allow SSL v1.1

File Configuration field:

nTLSAllowTLS11

Description:

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

Allow SSL v1.2

File Configuration field:

nTLSAllowTLS12

Description:

This option enables protocol method TLS1.2 which is enabled by default.

Allow TLS v1.3

File Configuration field:

nTLSAllowTLS13

Description:

This option enables protocol method TLS1.3 which provides enhanced security and performance.

Use OpenSSL configuration commands

File Configuration field:

nTLSUseConfigurationCommands

Description:

By enabling this option, you can set OpenSSL configuration commands directly. For more information's on available configuration parameters for each command type, visit this page:

https://www.openssl.org/docs/man1.0.2/ssl/SSL_CONF_cmd.html

We allow to the set the following OpenSSL configuration commands in the configuration commands list:

- CipherString: Sets the allowed/disallowed used Ciphers. Setting this value will OVERWRITE the internal default ciphers.
- SignatureAlgorithms: This sets the supported signature algorithms for TLS v1.2.
- Curves: This sets the supported elliptic curves.
- Protocol: Sets the supported versions of the SSL or TLS protocol. This will OVERWRITE the Allow SSL options from above!
- Options: The value argument is a comma separated list of various flags to set.

When setting advanced configuration commands, we highly recommend to enable debug logging and review it after changes have been made. An error will be logged in the debug logfile if a configuration command cannot be processed successfully.

TCP related Options

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

Please Note: This option is not available for UDP or RFC 3195.

Session Timeout

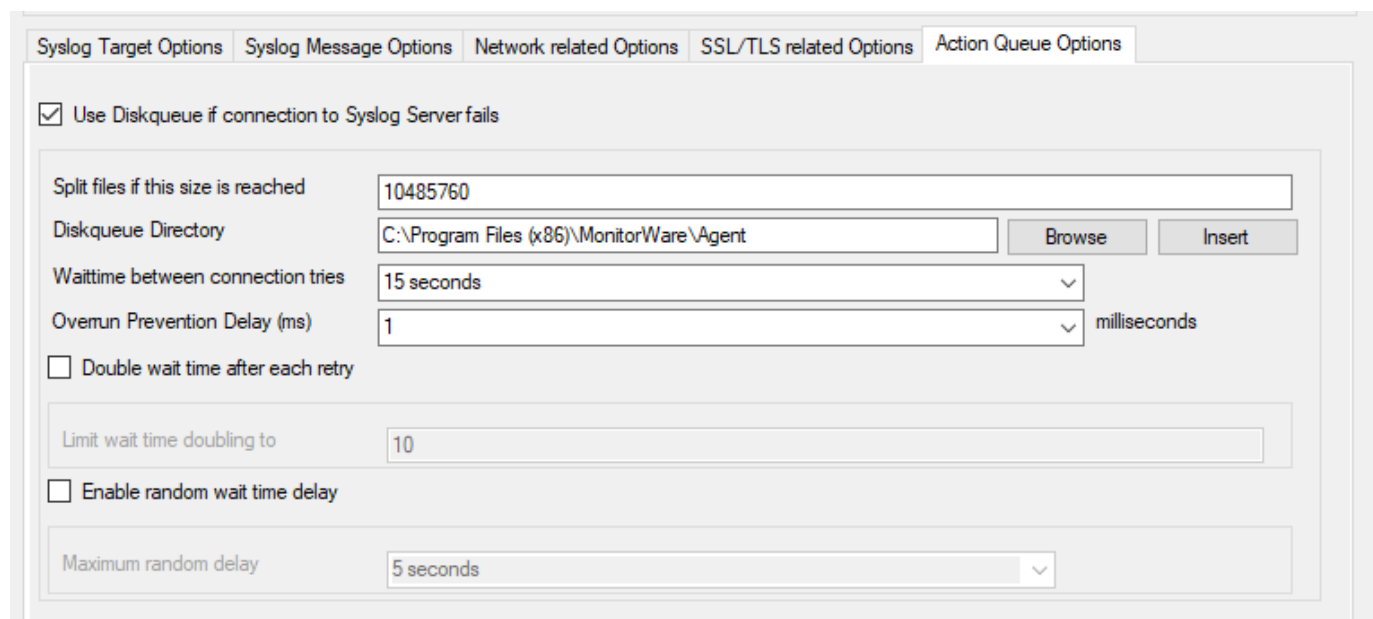
File Configuration field:

nTimeoutValue

Description:

Timeout value for TCP persistent and octet-count based framing connections.

Action Queue Options



☒ Use Diskqueue if connection to Syslog Server fails

Split files if this size is reached: 10485760

Diskqueue Directory: C:\Program Files (x86)\MonitorWare\Agent Browse Insert

Waittime between connection tries: 15 seconds

Overrun Prevention Delay (ms): 1 milliseconds

☐ Double wait time after each retry

Limit wait time doubling to: 10

☐ Enable random wait time delay

Maximum random delay: 5 seconds

- Action - Forward Syslog Action Queue*

Use Diskqueue if connection to Syslog server fails

File Configuration field:

nUseDiscQueue

Description:

Enable diskqueueing syslog messages after unexpected connection loss.

Split files if this size is reached

File Configuration field:

nDiskQueueMaxFileSize

Description:

Files will be split until they reach the configured size in bytes. The maximum support file size is 10485760 bytes.

Diskqueue Directory

File Configuration field:

szDiskQueueDirectory

Description:

The directory where the queue files will be generated in. The queuefiles will be generated with a dynamic UUID bound to the action configuration.

Waittime between connection tries

File Configuration fields:

nDiskCacheWait

Description:

The minimum waittime until the Syslog Action retries to establish a connection to the Syslog server after failure.

Overrun Prevention Delay (ms)

File Configuration field:

nPreventOverrunDelay

Description:

When the Action is processing syslog cache files, an overrun prevention delay can be added to avoid flooding the target Syslog server.

Double wait time after each retry

File Configuration field:

bCacheWaittimeDoubling

Description:

If enabled, the configured waittime is doubled after each try.

Limit wait time doubling to

File Configuration field:

nCacheWaittimeDoublingTimes

Description:

How often the waittime is doubled after a failed connection try.

Enable random wait time delay

File Configuration field:

bCacheRandomDelay

Description:

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

Maximum random delay

File Configuration field:

nCacheRandomDelayTime

Description:

Maximum random delay time that will be added to the configured waittime if Enable random wait time delay is enabled.

UDP related Options

Enable IP Spoofing for the UDP Protocol

File Configuration field:

nSpoofIPAddress

Description:

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

Fixed IP or single property

File Configuration field:

szSpoofedIPAddress

Description:

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

DTLS Servename	<input type="text" value="127.0.0.1"/>
DTLS Port	<input type="text" value="4433"/>
Send /Receive Timeout	<input type="text" value="5 seconds"/>
Message Format	<div><input type="text" value="%msg%"/><div>▲▼</div></div> <div>Insert</div>
TLS Options	
TLS Mode	<input type="text" value="Anonymous authentication"/>
Select common CA PEM	<div><input type="text" value="..\tls-ca.pem"/><div>Browse</div></div>
Select Certificate PEM	<div><input type="text" value="..\tls-client-cert.pem"/><div>Browse</div></div>
Select Key PEM	<div><input type="text" value="..\tls-client-key.pem"/><div>Browse</div></div>

- DTLS Servername

DTLS Port

Send/Receive Timeout	
Send Timeout	10000
Receive Timeout	10000

Message Format

173

TLS Options

TLS Mode

File Configuration field:

nTLSMode

Description:

Specifies the authentication method used. Options include “Anonymous authentication” or other modes requiring certificates.

Select common CA PEM

File Configuration field:

szTLSCAFile

Description:

Path to the CA certificate file (e.g., *tls-ca.pem*).

Select Certificate PEM

File Configuration field:

szTLSCertFile

Description:

Path to the client certificate file (e.g., *tls-client-cert.pem*).

Select Key PEM

File Configuration field:

szTLSKeyFile

Description:

Path to the private key file for the client (e.g., *tls-client-key.pem*).

internal actions

Call RuleSet

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

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

Note: there is no limit on including the rules which means that a rule

that has been included in another rule may contain another rule in it which might contain another rule in it and so on.

RuleSets > Default RuleSet > Default Rule > Call RuleSet Enabled Comments Settings Confirm Reset ?

RuleSet to process Refresh

- Action - Call RuleSet*

Ruleset to Call**File Configuration field:**

szRuleSet

Description:

Select the Ruleset to be called.

Note: Call RuleSet stays disabled until you have more than “One” RuleSet!**Compute Status Variable**

An internal action used to compute a status variable. This is needed for RuleSets which operate on a counter basis. This dialog controls the compute status variable options.

RuleSets > Default RuleSet > Default Rule > Compute Status Variable Enabled Comments Settings Confirm

Status variable

Operation type

☒ Increment Value (+)

☐ Decrement Value (-)

Operation value

- Action - Compute Status Variable*

Status variable**File Configuration field:**

szStatusVar

Description:

Name of the unique status variable.

Operation Type**Increment Value****File Configuration field:**

nCalcType = 1

Description:

It increments the value by the operation value.

Decrement Value**File Configuration field:**

nCalcType = 2

Description:

It decrements the value by the operation value.

Operation value**File Configuration field:**

nChangeVal

Description:

The operation value that is to be used.

Discard

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

A sample how to ignore Events with the Discard Action can be found here: [ignoring events](#).

Normalize Event

Parameters can be normalized and converted into XML, CSV, and JSON formats. The normalization result is stored into an internal property which can be used for filtering decisions as well as for output actions.

The action uses liblognorm (<http://www.liblognorm.com/files/manual/index.html>) which is also used by Rsyslog. Rulebases created for liblognorm can easily be used and adapted.

- Action - Normalize Event*

Normalize Parameter

File Configuration field:

szMessage

Description:

Specifies the property that you want to normalize, by default this is the %msg% property.

Select Rulebase File

File Configuration field:

szRulebase

Description:

The text file that contains the rulebase definitions (see liblognorm documentation for more).

Lognorm Output Format

File Configuration field:

nOutputFormat

- 0 = DISABLED
- 1 = JSON
- 2 = XML
- 3 = CSV

Description:

- Disabled: No additional output format.
- JSON Format: Creates a string formatted in JSON which is stored in the output property.
- XML Format: Creates an XML formatted string which is stored in the output property.
- CSV Format: Creates a CSV (Comma separated values) string which is stored in the output property.

Output Property

File Configuration field:

szOutputProperty

Description:

The property where the normalized format is saved to.

Post Processing

The Post Processing action allows you to re-parse a message after it has been processed e.g. Tab Delimited format. Such re-parsing is useful if you either have a non-standard Syslog format or if you would like to extract specific properties from the message.

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

RuleSets > Default RuleSet > Default Rule > Post Processing Enabled Comments Settings Confirm Reset

Import Rules Export Rules

Property List

	Property Name	Type	Value
*	*Enter value for Property Name*	Character Match	*Enter value for Value*

Message Preview of your rules

- Action - Post Processing*

Templates

Parse maps can be quite complex. In order to facilitate exchange for parse maps, they can be persisted to XML files. Adiscon also plans to provide parse maps for some common devices.

We know that creating a parse map is often not a trivial task. If you are in doubt how to proceed, please contact us via the [Customer Service System](#) - we will happily assist you with your needs. In this case, you will probably receive a parse map file that you can import here.

The Parse Map Editor

In this dialog, you can edit only in the text boxes above the data grid. When you select an entry in the grid, its values are updated in the textboxes. Any edits made there will automatically be reflected to the grid. Pressing Insert or Delete will create a new entry or delete the currently selected one.

Property Name**File Configuration field:**

szProperty_[n]

Description:

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

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

Type**File Configuration field:**

nSyntax_[n]

- Integer = 101
- IPV4Addr = 102
- CharMatch = 201
- RestOfMessage = 202
- Word = 203
- UpTo = 204
- TimeStampISO = 301
- TimeStampUNIX = 302

Description

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

Value**File Configuration field:**

szParsValue_[n]

Description:

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

Message Preview

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

Parsing log messages

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

Get relevant information from logs

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

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

Tool kit for parsing

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

Determine necessary information

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

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

A sample message looks like:

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

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

```
Pos = Position of the character
*p  = Points to the position the parser stands after parsing the*
      rule
Log = Message subdivided into its characters.
Pro = Property. In the term of Adiscon a property is the name of
      the item which is parsed out.mk:
      @MSITStore:C:\PROGRA~2\MONITO~1\Agent\manual\MONITO~1.CHM::/
```

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

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

Parsing out a Timestamp

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

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

RuleSets > Default RuleSet > File Action > Post Processing Enabled Comments Settings Confirm Reset

Import Rules Export Rules

Property List

	Property Name	Type	Value
	u-timestamp	UNIX/LINUX-like Timestamp	*Enter value for Value*
»	*Enter value for Property Name*	Character Match	*Enter value for Value*

Message Preview of your rules

```
40 23 09:40:56
```

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

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

Get the IP-Address

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

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

RuleSets > Default RuleSet > File Action > Post Processing Enabled Comments Settings Confirm Reset

Import Rules Export Rules

Property List

	Property Name	Type	Value
	u-timestamp	UNIX/LINUX-like Timestamp	*Enter value for Value*
	Filler	Character Match	Space
	Source	IP V4 Address	*Enter value for Value*
▶▶	*Enter value for Property Name*	Character Match	*Enter value for Value*

Message Preview of your rules

```
42 23 09:42:01 192.168.0.1
```

- Post-Process Editor: Note the value of 'Character Match' rule is a Space*

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

Obtain the syslogtag

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

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

RuleSets > Default RuleSet > File Action > Post Processing Enabled Comments Settings Confirm Reset

Import RulesExport Rules

Property List

	Property Name	Type	Value
	u-timestamp	UNIX/LINUX-like Timestamp	*Enter value for Value*
	Filler	Character Match	
	Source	IP V4 Address	*Enter value for Value*
	Filler	Character Match	%
	syslogtag	UpTo	:
»*	*Enter value for Property Name*	Character Match	*Enter value for Value*

Message Preview of your rules

43 23 09:43:49 192.168.0.1 %:

• Post-Process Editor*

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

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

Take the MessageID

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

MessageID consist of numbers only. For numeric parsing the 'Integer' parse type exist. This type captures all characters until a non-numeric character appears. The pointer is moved behind the number. Note that numeric values with decimal dots cannot be parsed with this type (because they are not integers). This means trying to parse 1.1 results in 1, because the dot is a non-numeric value.

RuleSets > Default RuleSet > File Action > Post Processing Enabled Comments Settings Confirm Reset

Import Rules Export Rules

Property List

	Property Name	Type	Value
	u-timestamp	UNIX/LINUX-like Timestamp	*Enter value for Value*
	Filler	Character Match	
	Source	IP V4 Address	*Enter value for Value*
	Filler	Character Match	%
	syslogtag	UpTo	:
	Source	Character Match	:
	u-messageid	Integer	*Enter value for Value*
▶▶	*Enter value for Property Name*	Character Match	*Enter value for Value*

Message Preview of your rules

```
45 23 09:45:46 192.168.0.1 %:: 12345
```

• Post-Process Editor*

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

Find the username and status

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

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

RuleSets > Default RuleSet > File Action > Post Processing Enabled Comments Settings Confirm Reset

Property List

Property Name	Type	Value
Source	IP V4 Address	*Enter value for Value*
Filler	Character Match	%
syslogtag	UpTo	:
Source	Character Match	:
u-messageid	Integer	*Enter value for Value*
Filler	UpTo	user=
Filler	Character Match	user=
u-username	Single Word	*Enter value for Value*
* * "Enter value for Property Name"	Character Match	*Enter value for Value*

Message Preview of your rules

```
47 23 09:47:16 192.168.0.1 %:: 12345user=user=aWord
```

- Post-Process Editor*

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

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

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

RuleSets > Default RuleSet > File Action > Post Processing Enabled Comments Settings Confirm Reset

Import Rules Export Rules

Property List

	Property Name	Type	Value
	Source	Character Match	:
	u-messageid	Integer	*Enter value for Value*
	Filler	UpTo	user=
	Filler	Character Match	user=
	u-username	Single Word	*Enter value for Value*
	Filler	UpTo	status=
	Filler	Character Match	status=
	u-status	Single Word	*Enter value for Value*
▶▶	*Enter value for Property Name*	Character Match	*Enter value for Value*

Message Preview of your rules

```
48 23 09:48:31 192.168.0.1 %:: 12345user=user=aWordstatus=status=aWord
```

The last rule - Additional Information

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

RuleSets > Default RuleSet > File Action > Post Processing Enabled Comments Settings Confirm Reset

Import Rules Export Rules

Property List

	Property Name	Type	Value
	Filler	Character Match	user=
	u-username	Single Word	*Enter value for Value*
	Filler	UpTo	status=
	Filler	Character Match	status=
	u-status	Single Word	*Enter value for Value*
	Filler	UpTo	msg=
	Filler	Character Match	msg=
	msg	Rest of Message	*Enter value for Value*
▶▶	*Enter value for Property Name*	Character Match	*Enter value for Value*

Message Preview of your rules

```
49 23 09:49:56 192.168.0.1 %:: 12345user=user=aWordstatus=status=aWordmsg=msg=$R$E$M$A$N$D$E$R$$$M$$$G$$$$$$$$$$$$$$$$
```

- Complete parse template.*

What happens if the parser fails?

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

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

RuleSets > Default RuleSet > File Action > Post Processing Enabled Comments Settings Confirm Reset

Import Rules Export Rules

Property List

Property Name	Type	Value
u-timestamp	UNIX/LINUX-like Timestamp	*Enter value for Value*
Filler	Character Match	
Source	IP V4 Address	*Enter value for Value*
Filler	Character Match	% Rule does not match
syslogtag	Up To	:
Source	Character Match	:
u-messageid	Integer	*Enter value for Value*
Filler	Up To	user=
Filler	Character Match	user=

Message Preview of your rules

```
49 23 09:49:56 192.168.0.1 %:: 12345user=user=aWordstatus=status=aWordmsg=msg=$R$E$M$A$I$N$D$E$R$$$M$$$G$$$$$$$$$$$$$$$$$
```

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







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


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


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

Resolve Hostname Action

Many Customers asked for resolve hostname options in different services. This feature has now been implemented as an action. An action can be used with every service, and it does not delay the work of a service.

RuleSets > Default RuleSet > Default Rule > Resolve Hostname  Enabled  Comments  Settings  Confirm  Reset 

Select Source Property from which the name will be resolved  Insert

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

☐ Cache resolved host entry

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

- Action - Resolve Hostname*

Select Source Property from which the name will be resolved**File Configuration field:**

szSourcePropertyName

Description:

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

Destination Property in which the resolved name will be saved to**File Configuration field:**

szDestinationPropertyName

Description:

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

Also resolve name if the source property is already a name**File Configuration field:**

nResolvelfName

Description:

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

Cache resolved host entry**File Configuration field:**

nCacheNameEntry

Description:

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

Set Property

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

This dialog controls the set property options. With the “Set Property” action, some properties of the incoming message can be modified. This is especially useful if an administrator would like to e.g. rename two equally named devices.

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

RuleSets > Default RuleSet > Default Rule > Set Property Enabled Comments Settings Confirm Reset

Select property Type

Set property value

- Action - Set Property*

Select Property Type**File Configuration field:**

szPropertyType

Description:

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

Please note that the field content can be configured with event properties are described in the property replacer section.

Set Property Value

File Configuration field:

szPropertyValue

Description:

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

Please note that the field content can be configured with event properties are described in the property replacer section.

Set Status

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

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

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

- Action - Set Status*

Status Variable Name

File Configuration field:

szPropertyName

Description:

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

Please note that the field content can be configured with event properties are described in the property replacer section.

Status Variable Value

File Configuration field:

szPropertyValue

Description:

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

Please note that the field content can be configured with event properties are described in the property replacer section.

other actions

Play Sound

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

- Action - Play Sound*

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.

There is a work around if you want to use play sound action for a second sound card!

Filename of the Soundfile

File Configuration field:

szFilename

Description:

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

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

Playcount

File Configuration field:

nCount

Description:

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

Delay between Plays

File Configuration field:

nDelay

Description:

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

Start Program

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

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

The screenshot shows a configuration interface with the following elements:

- A text field labeled "Command to execute" with a "Browse" button to its right.
- A checked checkbox labeled "Use legacy parameter processing".
- A text field labeled "Command Parameters" with an "Insert" button to its right.
- A radio button labeled "Synchronous Processing (Wait for Completion)".
- A dropdown menu labeled "Sync Timeout" with "10 seconds" selected.

- Action - Start Program*

Command to execute

File Configuration field:

szCommand

Description:

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

Use legacy parameter processing

File Configuration field:

nUseLegacyProcessing

Description:

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

Parameters

File Configuration field:

szParameters

Description:

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

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

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

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

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

Sync Timeout

File Configuration field:

nSyncTimeOut

Description:

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

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

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

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

Getting Help

The WinSyslog Service is very reliable. Here's how to get help if you encounter problems.

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 support in the local language. Please check with them.**

Frequently Asked Questions

For a current list of Frequently Asked Questions (FAQ), please visit <https://www.WinSyslog.com/category/faq/general-questions/>. The FAQ area is continuously being updated.

Customer Service System

Our Customer Service System is available at <https://ticket.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 User name or email and Password when you open it. If you do not have registered yet, you can simply click the "register now" button to do so. If you choose the continue "as guest" button you can also open a ticket without registering first.

Why using the Customer Service 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 Service 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. However, there are some situations where the support system should be used:

- Email notifications do NOT include attachments!** If we provide an attachment, you must login into the ticket in order to obtain this. For your convenience, each email notification contains an active link that allows you to login immediately.

- If you seem to not receive responses from us, it is a very good idea to check the web interface. Unfortunately, anti-SPAM measures are being setup more and more aggressive.** We are noticing an increasing number of replies that simply do not make it to your mailbox, because some SPAM filter considered it to be SPAM and removed it. Also, it may happen that your support question actually did not get past our own SPAM filter. We try very hard to avoid this. If we discard mail, we send a notification of this, so you should at least have an indication that your mail did not reach us. Using the customer support system via its own web interface removes all SPAM troubles. So we highly recommend doing this if communication otherwise seems to be disturbed. In this case, please remember that notification emails may also get lost, so it is a good idea to check your ticket for status updates from time to time.

Email

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

Please note: we have increasingly seen problems with too-aggressive SPAM filtering, resulting in loss of our replies. If you do not receive a response from us within two working days, we highly recommend re-submitting your support call via the** [Customer Service System](#).

Phone

Phone support is limited to those who purchased support incidents. If you are interested in doing so, please contact us via the [Customer Service System](#) for further details.

WinSyslog Web Site

Visit the support area at <https://www.WinSyslog.com/help/support/> for further information. If for any reason that URL ever becomes invalid, please visit <https://www.adiscon.com> for general information.

Software Maintenance

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

[Click here](#) to learn more about UpgradeInsurance.

Non-Technical Questions

For non-technical questions please contact us via [the Customer Service System](#).

Product Updates

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

Please visit <https://www.WinSyslog.com/> for information about new and updated products.

Concepts

WinSyslog offers advanced monitoring capabilities. It cannot 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. MonitorWare Agent has multiple and very powerful capabilities. This enables very quick configuration of highly efficient and comprehensive systems. On the other hand, the concepts must be fully understood to make such complex systems really work

Purchasing

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

The License

The end user license agreement (EULA) is displayed during setup. If you need to receive a copy of the license agreement, please contact us via the [Customer Service System](#). The license agreement of the current version can be also found on the web site of the product: <https://www.WinSyslog.com/WinSyslog-eula/>

Which Edition is for Me?

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

<https://www.WinSyslog.com/product-info/edition-comparison/>

Pricing & Ordering

Please visit <https://www.WinSyslog.com/order-and-pricing/order-now/> to obtain pricing information. This form can also be used for placing an order online. If you would like to place a purchase order, please visit <https://www.adiscon.com/purchase-orders/> to obtain details.

If you would like to receive assistance with your order or need a quote, please contact us via the [Customer Service System](#).

Articles

Here you find articles about the WinSyslog Products:

Difference between Set Status - Set Property Action

The difference is, that a property is a part of the message object, while the status is a global object. That means that a property will change with every message, e.g. the timestamp or msg property. So properties are the actual different values of a message. Values are gone once the message is fully processed.

A status on the other hand is global and stays the same across all messages if not altered. A status variable can be filtered, but its value cannot be emitted in a message.

A setup for using a custom status object could be like this: Message "x" gets processed every now and then. Every time the message comes through the status variable "y" can be increased by 1. Once the status reaches a value of "n" a special message with properties from message "x" can be sent via email and the status variable "y" will be reset.

So, while setup looks similar, these are actually very different in their concept.

1. Initializing 'status': This is usually done when needed in the processing flow with "Set Status".
2. How to set the status value again to initial value if I want to count up from the start: If you have a condition that waits for a specific status value to be reached, you can simply use another "Set Status" action after the output action.
3. Simple question: Status value should be numeral value? You can use numerical as well as alphabetical values for a status. But, I suppose it is best to only use something like boolean-like words when using alphabetical values, like on/off or yes/no. this is mostly because of logical understanding. Numerical values should be used for counters of course.

How can I use a second sound card with the Play Sound Action?

I have got a second sound card on my machine, how can I use it with the Play Sound Action?

PlaySounds action plays a sound on the local machine. It is possible to play wave files and some other "system" supported sound files. This does "NOT" include mp3 files. As MonitorWare Agent is usually running "as a" System service, there are some things which needed to be noted!

On machines with more than ONE sound card, the MonitorWare Agent Service will take the "first active installed soundcard as output device regardless what is configured".

If there is a need to play the sound on another sound card instead of the first active installed one, then there are two workarounds:

1. Specify a "User Account" for the Service which has a local profile where the sound card you want to use configured as primary playback device.
2. Run the MonitorWare Agent Service in console mode using the "-r" switch under a user account which has the sound card you want to use configured as primary playback device.

By following the above mentioned work around, you would be able to use second sound card (even x sound cards where x is user configurable) with the Play Sound Action.

The following things are user configurable in the Play Sound Action

Filename of the Soundfile - A full path and filename to the wave file which will be played. If the sound file specified here cannot be found or is not a valid wave file, a simple system beep will be played.

Playcount: Default is 1 - can be configured up to 100 times.

Delay between the sound plays - Only useful if the sound is played more than once. Between each play, MonitorWare Agent will wait for this time until it plays the sound again.

Note: A prior running sound will be aborted when this action is executed.

Default Timevalues Setting in EventReporter/MonitorWare Agent/WinSyslog explained

- Created: 2008-01-24 Andre Lorbach
- Updated: 2020-10-05 adisconteam

The general options of each product (EventReporter, MonitorWare Agent and WinSyslog) contain a setting for the "Default Timevalues". This setting can be set to Localtime and UTC (Universal Coordinated Time) which is default.

If you switch this setting to Localtime, you may wonder why output timevalues still are in UTC.

Internally we need to calculate with UTC time. This is needed in order to maintain the time values if they are send via Syslog or SETP. If we wouldn't do this, this could result to unexpected time differences.

So where does this setting have an effect?

- Send Email Action: The date in the email header is affected
- Start Program Action: Time parameters in the command line are affected
- Write File Action: Time properties in the file name are affected
- Filter Engine: If you filter by weekday or time fields, localtime does affect the filter result

But how can I get localtime output?

We added two additional options into the property engine which can be applied on time based values for this purpose.

Property Option: **localtime** = converts the output of the timestamp into localtime:

Sample: %variable:::localtime%

Property Option: **uxLocalTimeStamp** = same output as uxTimeStamp, but localtime is used

Sample: %variable:::uxLocalTimeStamp%

further articles you find on adiscon.com :

- [articles](#)

and on [WinSyslog](#) :

- [WinSyslog Articles](#)

FAQ

Here you find FAQ about the WinSyslog Products:

Why are Logfiles sometimes not rotated in WinSyslog 17.5 or lower?

This article explains why log files may sometimes not be rotated as expected in WinSyslog versions 17.5 and earlier, and provides solutions for this issue.

Background

In WinSyslog versions 17.5 and earlier, there is a feature called “Automatic File Handle Cleanup” that can interfere with log file rotation under certain circumstances. This feature was redesigned in WinSyslog version 18.1 to resolve these issues.

The Problem

Users may experience inconsistent log file rotation behavior where:

- Some log files rotate successfully every day as scheduled
- Some log files rotate only partially (not every day)
- Some log files never rotate at all

This typically occurs when log rotation is scheduled at specific times (e.g., at 0:00 every day).

Root Cause

The issue is related to WinSyslog’s Automatic File Handle Cleanup feature, which by default:

- Closes inactive log files after 2 hours of no activity
- Frees up memory and system resources
- Prevents resource exhaustion when many log files are being monitored

At the time of scheduled rotation, if a log file has been inactive for more than 2 hours, WinSyslog may have already closed its file handle. When the rotation process runs, it cannot rotate a file that is no longer actively opened by WinSyslog.

Note: This behavior is similar to how your computer closes unused programs to maintain system stability.

Affected Versions

This issue affects WinSyslog versions 17.5 and earlier. The log rotation logic was redesigned starting with WinSyslog version 18.1, which resolves these limitations.

Solutions

Recommended Solution: Upgrade WinSyslog

The most effective solution is to upgrade to WinSyslog version 18.1 or later, where the log rotation logic has been redesigned to handle these scenarios properly.

Alternative Solution: Adjust File Handle Cleanup Interval

If upgrading is not immediately possible:

1. Open the WinSyslog configuration
2. Navigate to the service settings
3. Increase the “Automatic File Handle Cleanup” interval from the default 2 hours to 24 hours (or longer, depending on your environment)
4. This adjustment will reduce the likelihood of missing log rotations

Important: The longer cleanup interval may increase memory usage, so monitor your system's resource utilization accordingly.

Is WinSyslog v18+ supported on Windows Server IoT 2025?

Overview

This FAQ answers whether WinSyslog v18+ is supported on Windows Server IoT 2025 and outlines current guidance, functional status, and considerations for deployments, including Server Core.

Support Status

Official support:

- Windows Server IoT 2025 is not yet explicitly listed in the WinSyslog v18+ platform matrix.

Functional status:

- WinSyslog v18+ is known to function properly on Windows Server IoT 2025 (including Server Core) based on internal testing and field feedback.

Guidance for Server Core Deployments

Windows Server IoT 2025 Server Core does not provide a graphical user interface. For headless deployments, we recommend configuring WinSyslog using Adiscon Config Files (*.cfg), a portable, file-based configuration format.

Recommended workflow:

1. Create the configuration on a GUI-enabled machine - Install WinSyslog and open the Configuration Client - Configure rules, services, and actions as required - Export the configuration as Adiscon Config Files (*.cfg)
2. Transfer the configuration to Server Core - Copy the exported .cfg to the Server Core system (e.g., via PowerShell Remoting or SMB)
3. Enable File Config Mode and set paths via registry

Registry path: `HKEY_LOCAL_MACHINE\SOFTWARE\Adiscon\WinSyslog\Settings`

Required values:

- `szFileConfig` (REG_SZ): Example `c:\configs\winsyslog\central-server.cfg`
- `szDataDirectory` (REG_SZ): Example `c:\configs\winsyslog\`
- `iAccessMode` (REG_DWORD): 1 (enables file config mode)

Important

When running in file config mode, ensure the service account has read access to the configuration file and write access to the data directory.

Notes

Troubleshooting the Start Program action in WinSyslog

This article explains common issues with the Start Program action in WinSyslog and provides solutions to resolve them.

Background

The Start Program action allows WinSyslog to execute external programs, batch files, or scripts when specific conditions are met. However, there are several common issues that can prevent this action from working correctly.

Common Issues and Solutions

Issue 1: Program not found or path problems

Symptoms: - The Start Program action appears to run but nothing happens - No error messages in the Windows Event Log - The external program works when run manually from command line

Root Cause: WinSyslog may not be able to locate the executable file due to path issues or missing dependencies.

Solutions:

1. **Use absolute paths for all executables** - Instead of: `curl google.com > temp.txt` - Use: `C:\curl\curl-win\bin\curl.exe google.com > C:\temp\temp.txt`
2. **Verify executable location** - Check if the program exists in the specified path - Ensure all required DLL files are present - Test the command manually from Windows Command Prompt
3. **Check Windows PATH environment variable** - WinSyslog may not have access to the same PATH as your user session - Use full paths instead of relying on PATH resolution

Issue 2: Permission problems

Symptoms: - No error messages in Event Log - Program works when run manually but not through WinSyslog

Root Cause: WinSyslog runs as a Windows service with different permissions than your user account.

Solutions:

1. **Store files in accessible locations** - Avoid system folders like `C:\Windows\System32` - Use generic folders like `C:\temp` or `C:\scripts` - Ensure WinSyslog service has read/execute permissions
2. **Check file permissions** - Right-click on the executable file - Go to Properties > Security - Ensure "SYSTEM" and "SERVICE" accounts have execute permissions

Issue 3: Working directory problems

Symptoms: - Program runs but cannot find input/output files - Relative paths in scripts don't work

Root Cause: The working directory when WinSyslog executes the program may be different from expected.

Solutions:

1. **Use absolute paths for all file references** - Instead of: `> temp.txt` - Use: `> C:\temp\temp.txt`
2. **Set working directory in batch files** - Add `cd /d C:\your\working\directory` at the beginning of batch files

Troubleshooting Steps

1. **Check Windows Event Log** - Open Event Viewer (type "Event Viewer" in Windows search) - Navigate to Windows Logs > Application - Look for WinSyslog-related error events
2. **Test with simple commands first** - Start with a basic batch file that creates a text file - Example: `echo Test > C:\temp\test.txt`
3. **Verify the command works manually** - Open Command Prompt as Administrator - Run the exact same command that WinSyslog should execute - Ensure it works from the command line first
4. **Check WinSyslog service account** - Verify which account WinSyslog is running under - Ensure that account has necessary permissions

Example Working Configuration

Here's an example of a properly configured Start Program action:

Command to execute: `C:\curl\curl-win\bin\curl.exe`

Parameters: `google.com > C:\temp\response.txt`

Key points: - Full path to `curl.exe` - Absolute path for output file - No reliance on PATH environment variable - Output directory exists and is writable

Additional Tips

- **Timeout settings:** Keep external programs under 5 seconds runtime for best performance
- **Error handling:** Consider adding error checking to your batch files
- **Logging:** Add logging to your scripts to help troubleshoot issues
- **Testing:** Always test Start Program actions in a development environment first

If you continue to experience issues after following these steps, please contact Adiscon support with: - WinSyslog version - Windows version - Exact command being executed - Any error messages from Event Log - Results of manual command testing

Is MariaDB supported by the ODBC action?

This article explains MariaDB support in ODBC database actions.

Question

Is MariaDB supported by the ODBC action?

Answer

Yes, MariaDB is fully supported by the ODBC action and can be used as a direct replacement for MySQL.

Background

MariaDB is a free and open-source alternative to MySQL. It is a fork of MySQL, initiated by the original MySQL developers after Oracle acquired Sun Microsystems (the former owner of MySQL). MariaDB was designed to be binary-compatible with MySQL, which generally makes switching from MySQL to MariaDB very easy.

Key characteristics of MariaDB:

- **Open Source:** MariaDB is consistently Open Source under a license that guarantees free use and further development
- **Binary Compatibility:** Designed to be binary-compatible with MySQL, making migration straightforward
- **Independent Development:** Continuous, independent development separate from MySQL
- **Performance:** Often preferred as an alternative due to sometimes better performance characteristics

Configuration

To use MariaDB with the ODBC action:

1. **Install MariaDB ODBC Driver:** - Download and install the [MariaDB Connector/ODBC driver](#) from the official MariaDB website - Ensure you install the correct version (32-bit or 64-bit) to match your Adiscon product installation
2. **Configure System DSN:** - Open the ODBC Data Source Administrator (use the 32-bit version if your product runs in 32-bit mode) - Create a new System DSN - Select the MariaDB ODBC driver - Configure the connection settings (server, database, credentials)
3. **Configure Database Action:** - In your Adiscon product configuration, select the ODBC Database action - Choose the MariaDB System DSN you created - Test the connection using the "Verify Database" button - Create the database tables if needed using the "Create Database" button

Note: The configuration process is identical to configuring MySQL, as MariaDB uses MySQL-compatible drivers and protocols.

Additional Information

For more information about database actions, see the ODBC Database Options documentation in your product's manual.

For MariaDB-specific information, visit the [official MariaDB website](#).

Recommended Palo Alto Firewall Syslog Configuration

This article provides configuration recommendations for Palo Alto firewalls to ensure consistent and reliable syslog message parsing by your syslog server.

Question

What is the recommended syslog format configuration for Palo Alto firewalls when sending logs to a syslog server?

Answer

We recommend configuring Palo Alto firewalls to use IETF RFC 5424 syslog format instead of BSD

RFC 3164

format. The IETF format provides a structured, unambiguous message format that ensures consistent parsing regardless of Palo Alto firmware version or spacing differences in log messages.

Why Use IETF (RFC 5424) Format?

IETF format is recommended over BSD

RFC 3164

format for the following reasons:

1. **Structured format:** IETF format includes a required APP-NAME field that eliminates parsing ambiguity
2. **Consistent parsing:** The structured format ensures your syslog server parses messages consistently regardless of: * Palo Alto firmware version * Spacing differences in log messages * Future firmware updates that may change message formatting
3. **Better compatibility:** IETF format is the modern syslog standard and provides better support for SIEM systems and log analysis tools
4. **Prevents parsing issues:** BSD format relies on heuristics that can be affected by spacing changes, potentially causing fields like `version=` to be parsed incorrectly or missing from output

Note: If you're experiencing issues where the `version=` field is missing from syslog output after a Palo Alto upgrade, this is typically caused by BSD format parsing ambiguity due to spacing changes. Switching to IETF format resolves this issue.

Configuration Steps

Step 1: Access Syslog Server Profile

1. Log in to the Palo Alto Networks firewall web interface
2. Navigate to: **Device > Server Profiles > Syslog**
Reference: [Palo Alto Documentation - Configure Syslog Monitoring](#)
3. Either: * Edit an existing syslog server profile, or * Click **Add** to create a new profile

Step 2: Configure Syslog Server Settings

For each syslog server in the profile:

1. **Name:** Enter a unique name for the server (if creating new)
2. **Syslog Server:** Enter the IP address or FQDN of your syslog server
3. **Transport:** * **Important:** IETF format typically uses TCP or SSL (TLS) * Select **TCP** or **SSL** (not UDP) * If using SSL, ensure TLSv1.2 is supported
Reference: [Palo Alto Documentation - Configure Syslog Server Profile](#)

4. **Port:** Enter the port number (default TCP syslog port is 514, but verify with your syslog server configuration)
5. **Format:** Select **IETF** (this is the key setting)
Reference: [Palo Alto Documentation - Configure Syslog Server Profile](#)
6. **Facility:** Select the appropriate syslog facility value (default is LOG_USER)

Step 3: Verify The Syslog Service Supports RFC 5424

Before applying the changes, ensure:

1. The Syslog Service supports RFC 5424 format: Verify that RFC 5424 parsing is enabled
Ensure RFC 5424 parsing is enabled in the Syslog Server service configuration.

Step 4: Commit Configuration

1. Click **OK** to save the syslog server profile
2. Commit the configuration
3. Review the commit and click **Commit** again to confirm
Reference: [Palo Alto Documentation - Commit Changes](#)

Step 5: Verify Configuration

After committing:

1. Check syslog messages on your syslog server
2. Verify the format: Messages should now appear in IETF format:

```
<14>1 2025-10-30T13:13:04.000Z e26secgw02 paloalto - - [meta version="11.2.6"] version=11.2.6|subtype=general|...
```
3. Verify APP-NAME field: The `paloalto` field (APP-NAME) should be present and consistently parsed by your syslog server
4. Verify output format: Syslog server output should now consistently include the `version=` prefix

Expected Results

After configuring IETF format, you should see:

- **Consistent message format:** Messages appear in structured IETF format with the APP-NAME field (`paloalto`) consistently parsed
- **Reliable field extraction:** All fields, including `version=`, are reliably extracted regardless of Palo Alto firmware version
- **Future-proof configuration:** The structured format ensures consistent behavior even after firmware upgrades
- **Better log analysis:** The structured format provides better support for SIEM systems and log analysis tools

Benefits Summary

Using IETF (RFC 5424) format provides:

- **Eliminates parsing ambiguity:** The structured format with required APP-NAME field ensures consistent parsing
- **Prevents version-related issues:** Spacing changes in firmware updates won't affect message parsing
- **Industry standard:** IETF format is the modern syslog standard recommended for enterprise environments
- **Better integration:** Improved compatibility with SIEM systems, log analysis tools, and centralized logging solutions

Technical Reference

- RFC 3164 (BSD)
- RFC 5424 (IETF)
- [Palo Alto Documentation - Configure Syslog Monitoring](#)
- [Palo Alto Documentation - Syslog Field Descriptions](#)
- [Palo Alto Documentation - Use Syslog for Monitoring](#)

Additional Information

For more information about syslog server configuration and RFC 5424 support, see the Syslog Server documentation in your product's manual.

further FAQ you find on adiscon.com :

- [FAQ](#)

and on [WinSyslog](#) :

- WinSyslog FAQ: [general questions](#) and [configuration](#)

References

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

- [Formats \(XML and Database\)](#)
- [Version History](#)
- [The WinSyslog Service](#)

Comparison of properties

Available in MonitorWare Agent, EventReporter and WinSyslog

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

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

Event Properties

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

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

Accessing Properties

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

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

```
%property:fromPos:toPos:options%
```

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

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

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

Property

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

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

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

FromPos

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

Fixed Character position

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

Search Pattern

A search pattern is specified as follows:

`/<search-pattern>/<options>`

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

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

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

Default Value

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

ToPos

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

Absolute Position

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

Relative Position

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

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

Search Pattern

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

A search pattern is specified as follows:

`<search-pattern>/<options>`

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

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

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

Search Example

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

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

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

```
%msg:/error/$:/occurred/%
```

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

```
%msg:/error /:/ occurred/$/%
```

Default

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

Options

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

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

lowercase

All characters in the resulting property extract will be converted to lower case.

uppercase

All characters in the resulting property extract will be converted to upper case.

uxTimeStamp

This is a special switch for date conversions. It only works if the extracted property value is an ISO-like timestamp (`YYYY-MM-DD HH:MM:SS`). If so, it will be converted to a Unix-like `ctime()` timestamp. If the extracted property value is not an ISO-like timestamp, no conversion happens.

uxLocalTimeStamp

This is the same as `uxTimeStamp`, but with local time instead of GMT.

date-rfc3339

This option is for replacing the normal date format with the date format from RFC3339.

date-rfc3164

This option is for replacing the normal date format with the date format from RFC3164.

date-rfc3164strict

Does the same as date-rfc3164 but when the date is below 10, two spaces will be added between Month and day (Which is defined in rfc3164).

escapecc

Control characters* in property are replaced by the sequence ##hex-val##, where* hex-val is the hexadecimal value of the control character (at least two digits, may be more).

spacecc

Control characters* in the property are replaced by spaces. This option is most* useful when a message contains control characters (e.g. a Windows Event Log Message) and should be written to a log file.

compressspace

Compresses multiple consecutive space characters into a single one. The result is a string where all words are separated by just single spaces. To also compress control characters, use the compressspace and spacecc options together (e.g. ``%msg:::spacecc,compressspace%``). Please note that space compression happens on the final substring. So if you use the FromPos and ToPos capabilities the substring is extracted first and then the space compression applied. For example, you may have the msg string "1 2". There are two space between 1 and 2. Thus, the property replacer expression: ``%msg:1:3:compressspace%``

will lead to "1 " ('1' followed by two spaces). If you intend to receive "1 2" ``('1' followed by one space, followed by '2')``, you need to use ``%msg:1:4:compressspace%``

or

```
%msg:1:/2/$:compressspace%
```

In the second case, the exact length of the uncompressed string is not known, thus a search is used in topos to obtain it. The result is then space-compressed.

compsp

Exactly the same as compressspace, just an abbreviated form for those that like it brief.

csv

For example %variable:::csv%. This option will create a valid CSV string. For example a string like this: this is a "test"! becomes this "this is a ""test""!" where quotes are replaced with double quotes.

cef

Convert string content into valid McAfee CEF Format. This means that =`` will be replaced with=`` and \ will be replaced with \. **convgermuml**

Converts German Umlaut characters to their official replacement sequence (e.g. "ö" → "oe")

localtime

Now you can print the Time with localtime format by using ``%variable:::localtime%``

nomatchblank

If this is used, the Property Replacer will return an empty string if the frompos or topos is not found.

replacepercent

This option replaces all % occurrences with a double %, which is needed for the property replacer engine in case that a string is reprocessed. This is needed because the percent sign is a special character for the property replacer. Once the property is processed, the double ``%`` become automatically one ``%``. **toipv4address**

Property string will be converted into IPv4 Address format if possible.

toipv6address

Property string will be converted into IPv6 Address format if possible.

crlftovbar

Does the same as `date-rfc3164` but when the date is below 10, two spaces will be added between Month and day (Which is defined in `rfc3164`).

removecc

Removes all control characters from 0x00 to 0x1F

replacechar

Replaces a single character with another single character.

How ASCII characters are being handled:

Sample: `%msg:$x:$y:replacechar%`

Broken down:

`%msg:$``<-` Tells property replacer that a character is being expected (At the moment only for REPLACECHAR Option). ``x``<-` The character to search for ```:`

`$``<-` Tells property replacer that a character is being expected (At the moment only for REPLACECHAR Option). ``y``<-` The character to replace with ```:`

`replacechar%`

How special characters are handled?

Sample: `%msg:$\n:$|:replacechar%`

`%msg:$ <-` Tells property replacer that a character is being expected (At the moment only for REPLACECHAR Option). `\n <-` The character to search for special character, possible values: t for tab,

n for newline,

v for verticaltab,

f for formfeed,

r for carriage return

for an actual backslash.``:

`$``<-` Tells property replacer that a character is being expected (At the moment only for REPLACECHAR Option). ``|``<-` The character to replace with ```:`

`replacechar%`

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

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

Simple Examples

A good example for this is the email subject line, which has severe length constraints. If you would like to have only the first 40 characters of the actual message text in the subject, you could use the replacer: `"%msg:1:40%".` If you know the first 10 characters of the message are meaningless for you but you would like to see the full rest of the message (no matter how long it may be), you can use a sequence like `"%msg:11%".`

If you would just like to see the plain message from beginning to end, you can simply omit frompos and topos: `"%msg".` Of course, all of these sample not only work with the “msg” property, but also with all others like “facility”, or “priority”, or W3C-log header extracted property names.

More complex Examples

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

If you would like to have the first 40 characters in front of the string “- aborted” (including that string):

`%msg:/- aborted/$:-40%`

If you would like to receive everything starting from (and including) “Log.”:

`%msg:/Log/%`

If you would like to have everything between the string “FROM” and “TO” including NONE of the both searchstrings:

References

```
%msg:/FROM/$:/TO/%
```

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

```
%msg:::lowercase%
```

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

```
%msg:50:::lowercase%
```

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

```
%datereceived:::uxTimeStamp%
```

Please see also the focused sample in the topos description.

A real world Sample

We use the following template to generate output suitable as input for MoniLog:

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

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

System Properties

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

\$CRLF

A Windows newline sequence consisting in the characters CR and LF. If you just need CR, you can use %\$CRLF:1:1% and if you need use LF you can use %\$CRLF:2:2%

\$TAB

An US-ASCII horizontal tab (HT, 0x09) character

\$HT

same as \$TAB

\$CR

A single US-ASCII CR character (shortcut for %\$CRLF:1:1%) **\$LF**

A single US-ASCII LF character (shortcut for ``%\$CRLF:2:2%``) **\$xNN**

A single character, whose value (in hexadecimal) is given by NN. NN must be two hexadecimal digits - a leading zero must be used if a value below 16 is to be represented. The value 0 (%x00) is invalid and - if specified - replaced by the “?” character.

As an example, \$CR could also be expressed as %\$x0d%.

Please note that only one character can be represented. If you need to specify multiple characters, you need multiple \$xNN sequences. An example may be \$CRLF which could also be specified as %\$x0d%\$x0a% (but not as %\$x0d0a%).

\$NOW

Contains the current date and time in the format:YYYY-MM-DD HH.MM.SS

Please note that the time parts are delimited by ‘.’ instead of ‘:’. This makes the generated name directly suitable for file name generation.

If you need just parts of the timestamp, please use the property replacer’s substring functionality to obtain the desired part. Use ``%\$NOW:1:4% to get the year,``

%\$NOW:6:7% to get the month,

...

%\$NOW:1:10% to get the full datestamp,

`;%$NOW:12:20%` to get the full timestamp

\$NEWUUID

Creates a new UUID (Universally Unique Identifiers), a unique 128-bit integer represented as a 32 digit hexadecimal number.

Custom Properties

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

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

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

Event-Specific Properties

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

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

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

Standard Properties

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

msgPropertyDescribed

A human-readable representation of the message text. While this is generally available, the exact contents largely depends on the source of the information. For example, for a file monitor it contains the file line and for a syslog message it contains the parsed part of the syslog message.

source

The source system the message originated from. This can be in various representations (e.g. IP address or DNS name) depending on configuration settings.

localhostname

On service startup it is automatically set to the local system computer name. It is read only and can be used if source property is not usable. E.g. if the Source property cannot be translated to IP format because the event log entry was recorded with an old computer name that no longer exists.

resource

A user-assigned numerical value. Does not have any specific meaning. Primarily intended for quick filtering.

CustomerID

A user-assigned numerical value. Does not have any specific meaning. Primarily intended for quick filtering.

SystemID

A user-assigned numerical value. Does not have any specific meaning. Primarily intended for quick filtering.

timereported

The time the originator tells us when this message was reported. For example, for syslog this is the timestamp from the syslog message (if not configured otherwise). Please note that timereported eventually is incorrect or inconsistent with local system time - as it depends on external devices, which may not be properly synchronized.

For Windows Event Log events, timereported contains the timestamp from the event log record.

timegenerated

The time the event was recorded by the service. If messages are forwarded via SETP, this timestamp remains intact.

importance

Reserved for future use.

iut

Indicates the type of the event. Possible values are:

- 1- syslog message
- 2- heartbeat
- 3- Windows Event Log Entry
- 4- SNMP trap message
- 5- file monitor
- 8- ping probe
- 9- port probe
- 10- Windows service monitor
- 11- disk space monitor
- 12- database monitor
- 13- serial device monitor

iuvers

Version of the event record (info unit). This is a monitorware internal version identifier.

Windows Event Log Properties

id

Windows Event ID

severity

severity as indicated in the event log. This is represented in string form. Possible values are:

- [INF] - informational
- [AUS] - Audit Success
- [AUF] - Audit failure
- [WRN] - Warning
- [ERR] - Error
- [NON] - Success (called "NON" for historical reasons)

severityid

The severity encoded as a numerical entity (like in Windows API)

sourceproc

The process that wrote the event record (called "source" in Windows event viewer).

category

The category ID from the Windows Event Log record. This is a numerical value. The actual value is depending on the event source.

catname

The category name from the Windows Event Log record. This is a string value. The actual value is depending on the event source. This value is a textual representation from the Category ID. This property could contain line feeds, which can be removed by activating the option “Remove Control Characters from String Parameters” in the advanced options of the EventLog Monitor Service.

user

The user name that was recorded in the Windows Event Log. This is “NA” if no user was recorded.

NTEventLogType

The name of the Windows Event Log this event is from (for example “System” or “Security”).

bdata

Windows Event Log records sometimes contain binary data. The Event Log Monitor service can be set to include this binary data into the event, if it is present. If it is configured to do so, the binary data is put into the “bdata” property. Every byte of binary data is represented by two hexadecimal characters.

Please note that it is likely for bdata not to be present. This is because the binary data is seldom used and very performance-intense. (%id%) - “%msg%”%\$CRLF%

Windows Event Log V2 Properties

id

Windows Event ID

severity

severity as indicated in the event log. This is represented in string form. Possible values are:

```
[INF] - informational
[AUS] - Audit Success
[AUF] - Audit failure
[WRN] - Warning
[ERR] - Error
[NON] - Success (called "NON" for historical reasons)
```

severityid

The severity encoded as a numerical entity (like in Windows API)

sourceproc

The process that wrote the event record (called “source” in Windows event viewer).

category

The category ID from the Windows Event Log record. This is a numerical value. The actual value is depending on the event source.

catname

The category name from the Windows Event Log record. This is a string value. The actual value is depending on the event source. This value is a textual representation from the Category ID. This property could contain line feeds, which can be removed by activating the option “Remove Control Characters from String Parameters” in the advanced options of the EventLog Monitor Service.

user

The user name that was recorded in the Windows Event Log. This is “NA” if no user was recorded.

ntheventlogtype

The name of the Windows Event Log this event is from (for example “System” or “Security”).

channel

The channel property for event log entries, for classic Event logs they match the %ntheventlogtype% property, for new event logs, they match the “Event Channel”.

sourceraw

This contains the full internal name of the event source for new event logs, for classic event logs it contains the same value as in %sourceproc%.

level

Textual representation of the event log level (which is stored as a number in %severityid%). This property is automatically localized by the system.

categoryid

Internal category id as number.

keyword

Textual representation of the event keyword. This property is automatically localized by the system.

user_sid

If available, contains the raw SID of the username (%user%) property.

recordnum

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

Syslog Message Properties

rawsyslogmsg

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

syslogfacility

The facility of a syslog message. For non-syslog messages, the value is provided based on configuration. In essence, this is simply an integer value that can be used for quick filtering inside your rules.

syslogfacility_text

The facility of a syslog message. This property is automatically created by using the syslogfacility properly and set to these values: "Kernel", "User", "Mail", "Daemons", "Auth", "Syslog", "Lpr", "News", "UUCP", "Cron", "System0", "System1", "System2", "System3", "System4", "System5", "Local0", "Local1", "Local2", "Local3", "Local4", "Local5", "Local6", "Local7"

syslogpriority

The severity of a syslog message. For non-syslog messages, this should be a close approximation to what a syslog severity code means.

syslogpriority_text

The severity of a syslog message. This property is automatically created by using the syslogpriority properly and set to these values: "Emergency", "Alert", "Critical", "Error", "Warning", "Notice", "Informational", "Debug"

syslogtag

The syslog tag value, a short string. For non-syslog messages, this is provided based on configuration. In most cases, this is used for filtering.

syslogver

Contains the syslog version number which will be one or higher if a rfc 5424 valid message has been received, or 0 otherwise

syslogappname

Contains the appname header field, only available if the Syslog message was in rfc 5424 format. Otherwise, this field will be emulated by the %syslogtag% property

syslogprocid

Contains the procid header field, only set if the Syslog message was in rfc 5424 format.

syslogmsgid

Contains the msgid header field, only set if the Syslog message was in rfc 5424 format.

syslogstructdata

Contains the structdata header field (in raw format), only set if the Syslog message was in rfc 5424 format.

syslogprifac

Contains combined syslog facility and priority useful to build your own custom syslog headers

Disk Space Monitor

currusage

The currently used disk space.

maxavailable

The overall capacity of the (logical) disk drive.

CPU/Memory Monitor

wmi_type

This variable is a string and can be one of the following variables: cpu_usage, mem_virtual_usage, mem_physical_usage, mem_total_usage.

cpu_number

Number of the current checked CPU.

cpu_load

The workload of the CPU as number, can be 0 to 100.

mem_virtual_load

How much virtual memory is used (MB).

mem_virtual_max

How much virtual memory is max available (MB).

mem_virtual_free

How much virtual memory is free (MB).

mem_physical_load

How much physical memory is used (MB).

mem_physical_max

How much physical memory is max available (MB).

mem_physical_free

How much physical memory is free (MB).

mem_total_load

How much total(Virtual+Physical) memory is used (MB).

mem_total_max

How much total(Virtual+Physical) memory is max available (MB).

mem_total_free

How much total(Virtual+Physical) memory is free (MB).

File Monitor

genericfilename

References

The configured generic name of the file being reported.

generatedbasefilename

Contains the generated file name without the full path.

Special IIS LogFile Properties

The Logfile Fields in IIS Logfiles are customizable, so there is no hardcoded command for their use.

The property-name depends on its name in the logfile. For example we take this Logfile:

```
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2005-10-27 14:15:25
#Fields: date time c-ip cs-username s-ip s-port cs-method cs-uri-stem
cs-uri-query sc-status cs(User-Agent)
2005-10-27 14:15:16 127.0.0.1 - 192.168.0.1 443 POST /eCommerce/asdf.php
2005-10-27 14:15:16 127.0.0.1 - 192.168.0.1 443 POST /eCommerce/asdf.php
2005-10-27 14:15:16 127.0.0.2 - 192.168.0.1 443 POST /eCommerce/asdf.php
2005-10-27 14:15:16 127.0.0.2 - 192.168.0.1 443 POST /eCommerce/asdf.php
```

As you can see, in our sample the fields are named: date, time, c-ip, cs-username, s-ip, and so on.

To use them as a Property inside our MonitorWareProducts, just use the names from your Logfile and add a "p-" before it:

p-date

The Date on which the Event occurs

p-time

The Time on which the Event occurs

p-c-ip

The IP address of the User which accessed

p-cs-username

The Username of the User which accessed

p-s-ip

The Server IP

p-s-port

The Server Port

p-cs-method

The Client-Server Method (POST,GET)

p-cs-uri-stem

The accessed File including its path

Windows Service Monitor

sourceproc

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

Ping Probe

echostatus

Status returned for the echo request

The status value can be one of the following:

References

```
0 = IP_SUCCESS
11002 = IP_DEST_NET_UNREACHABLE
11003 = IP_DEST_HOST_UNREACHABLE
11010 = IP_REQ_TIMED_OUT
11013 = IP_TTL_EXPIRED_TRANSIT
11016 = IP_SOURCE_QUENCH
11018 = IP_BAD_DESTINATION
```

roundtriptime

Round trip time for the ping packet (if successful)

Port Probe

responsestatus

The status of the probe.

responsemsg

The response message received (if any)

Database Monitor

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

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

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

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

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

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

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

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

Serial Monitor

portname

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

MonitorWare Echo Request

responsestatus

The status of the echo request. Possible values:

```
0 - request failed (probed system not alive)
1 - request succeeded
```

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

FTP Probe

ftpstatus

The status of the connection.

ftprespmsg

The response of the connection.

IMAP Probe

imapstatus

The status of the connection.

imaprespmsg

The response of the connection.

NNTP Probe

nntpstatus

The status of the connection.

nntprespmsg

The response of the connection.

SMTP Probe

smtpstatus

The status of the connection.

smtprespmsg

The response of the connection.

POP3 Probe

pop3status

The status of the connection.

pop3respmsg

The response of the connection.

HTTP Probe

httpstatus

The status of the connection.

httprespmsg

The response of the connection.

Complex Filter Conditions

The rule engine uses complex filter conditions.

Powerful boolean operations can be used to build filters as complex as needed. A boolean expression tree is graphically created. The configuration program is modeled after Microsoft Network Monitor. So thankfully, many administrators are already used to this type of Interface. If you are not familiar with it, however, it looks a bit confusing at first. In this chapter, we are providing some samples of how boolean expressions can be brought into the tree.

Example 1

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

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

```
If (msg = "DUPADDRESS") OR (msg = "SPANTREE") OR (msg = "DUPLEX_MISMATCH) then
    execute action(s)
end if
```

Please note: in the example, we have abbreviated “message” to just “msg”. Also note that for brevity reasons we use the equals (“=”) comparison operator, not the contains. The difference between the equals and the contains operator is that with “contains”, the string must just be part of the message.

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

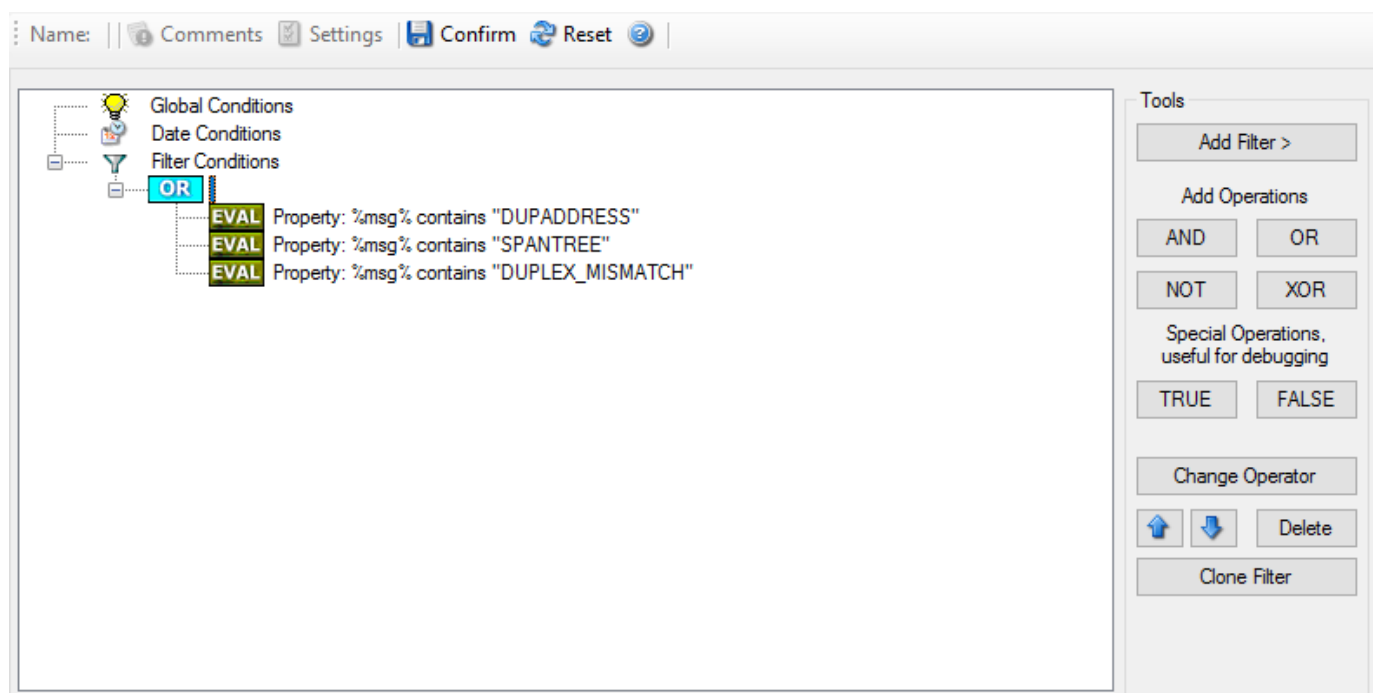


Figure 1 - Example 1

Example 2

Example 2 is very similar to example 1. Again, the message content is to be checked for three string. This time, all of these strings must be present in order for the boolean tree to evaluate to false.

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

```
If (msg = "DUPADDRESS") AND (msg = "SPANTREE") AND (msg = "DUPLEX_MISMATCH) then
    execute action(s)
end if
```

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

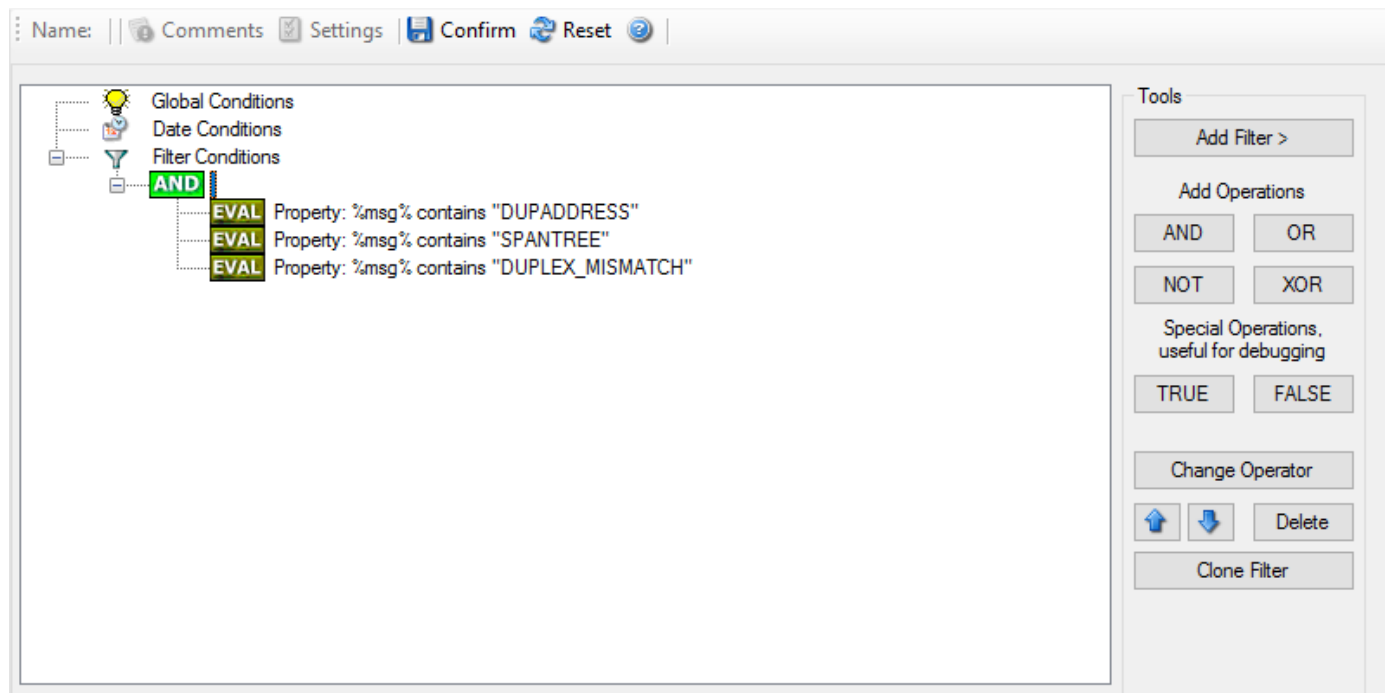


Figure 2 - Example 2

Example 3

This example is a bit more complex version of example 1. Again, the same message text filtering is done, that is if any one of the provided substrings is present, the filter eventually evaluates to true. To do so, the source system must also contain the string "192.0.2", which can be used to filter on a device from a specific subnet.

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

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

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

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

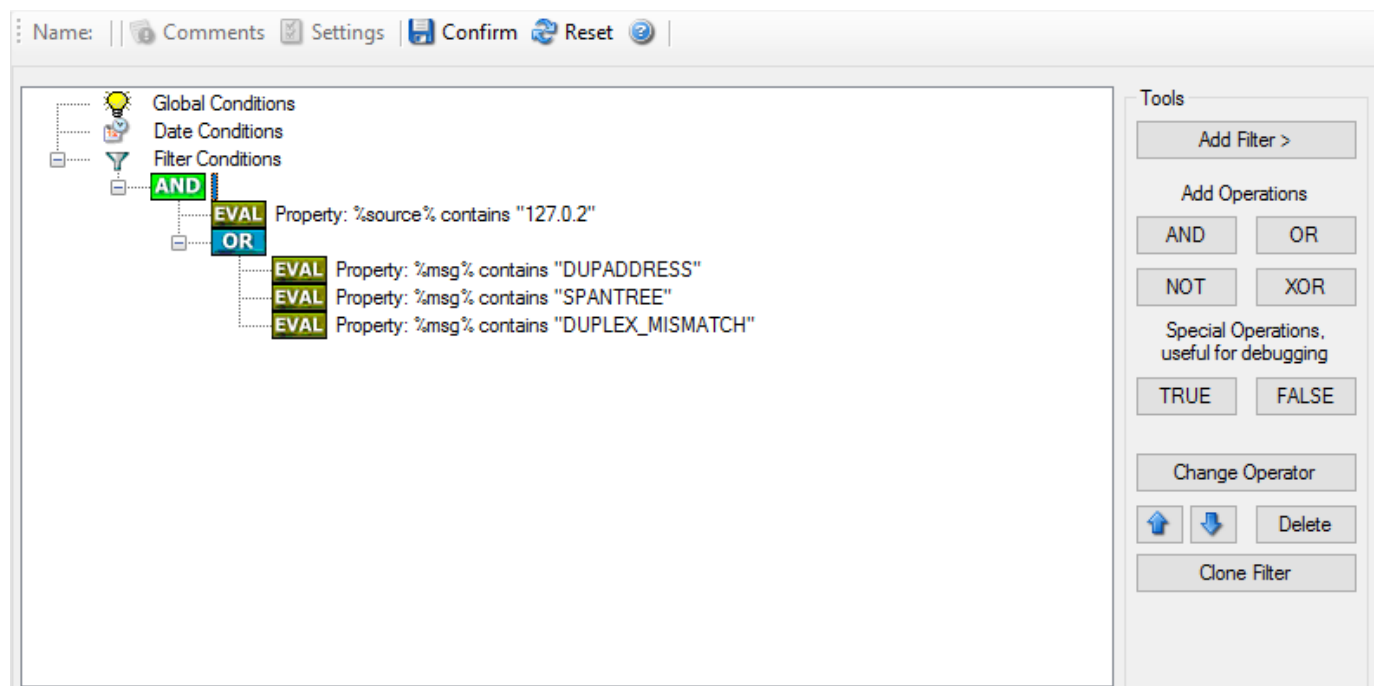


Figure 3 - Example 3

As a side note, you may want to use a range check instead of a simple include for the source system. With a range string check, you can specify that the string must be within a specified column range, in this case obviously at the beginning of the source system IP address.

Real-World Examples

To see some real-world examples of where boolean conditions inside filtering are used, please visit these web links:

- [Detecting Password Attacks under Windows](#)

Example 4

In this example, the report is to be filtered in such a way that it shows information only in the case, if the time is greater then certain time with certain event source and one of two event ID's.

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

```
If (DeviceReportedTime is greater than {9:16:27} AND EventSource is equal to {Print} AND [EventID is equal to {10} OR EventID is equal to {18}])
```

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

The screenshot shows the WinSyslog Filter Conditions editor. At the top, there is a toolbar with buttons for Name, Comments, Settings, Confirm, Reset, and a help icon. The main workspace displays a hierarchical tree of filter conditions. The tree starts with 'Global Conditions' (lightbulb icon), followed by 'Date Conditions' (calendar icon), and 'Filter Conditions' (funnel icon). Under 'Filter Conditions', there is an 'AND' operator (green box) which contains three 'EVAL' (yellow box) conditions: 'Time: > 09:16:27', 'Property: %source% contains "127.0.2"', and an 'OR' operator (blue box). The 'OR' operator contains three 'EVAL' conditions: 'Property: %msg% contains "DUPADDRESS"', 'Property: %msg% contains "SPANTREE"', and 'Property: %msg% contains "DUPLEX_MISMATCH"'. On the right side, there is a 'Tools' panel with buttons for 'Add Filter >', 'Add Operations' (AND, OR, NOT, XOR), 'Special Operations, useful for debugging' (TRUE, FALSE), 'Change Operator', 'Delete', and 'Clone Filter'. At the bottom, there is a 'Details' tab (selected) showing fields for 'Property Name' (IsTime), 'Compare Operation' (>), 'Set Property Value' (9:16:27 AM), and 'Select TimeMode' (Default TimeMode - Received). A link 'Learn about Filters' is at the bottom right.

WinSyslog Shortcut Keys

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

CTRL+S = Save

CTRL+X = Cut

CTRL+C = Copy

CTRL+V = Paste

CTRL+Z = Undo

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

Command Line Switches

There are several command line switches available for using the agent via the command line. To use the agent via the command line you need administrative rights.

- h Shows command line help
- v Shows version information and whether or not the service is installed
- i Install service
- u Remove (uninstall) service
- i Install service with a custom servicename "CustomServiceName"
- u Uninstall a service with a custom servicename "CustomServiceName"
- r Run as console application

- `r -o` Run ONCE as console application

If you install the service, you can start and stop the service with the “net start” and “net stop” commands. By using the “-r” switch, you run it only on the command line. When you close the command line, the program will stop working.

The “-v” switch gives you information about the version of the service.

You can import Adiscon Config Format (cfg) configuration files via the command line as well. The syntax is quite easy. Simply execute the configuration client and append the name of the configuration file. This could look like this:

Sample for MonitorWare Agent:

```
mwclient.exe example.cfg
```

Sample for EventReporter:

```
CFGEvntSLog.exe example.cfg
```

Sample for WinSyslog:

```
WINSyslogClient.exe example.cfg
```

Sample for Rsyslog Windows Agent:

```
RsyslogConfigClient.exe example.cfg
```

or

Sample for MonitorWare Agent:

```
mwclient.exe "example.cfg"
```

Sample for EventReporter:

```
CFGEvntSLog.exe "example.cfg"
```

Sample for WinSyslog:

```
WINSyslogClient.exe "example.cfg"
```

Sample for Rsyslog Windows Agent:

```
RsyslogConfigClient.exe "example.cfg"
```

After this is executed, you will see the splash screen of the configuration client and then the import dialogue, which you have to confirm manually.

For doing a silent import, the “/f” (without the quotes) parameter has to be appended. This will look like this:

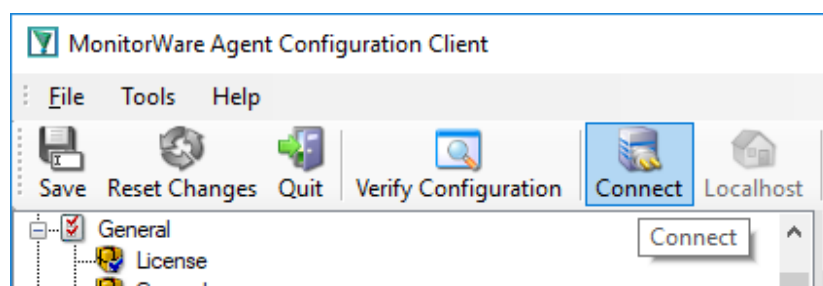
```
mwclient.exe "example.cfg" /f
```

In this case, the filename of the configuration has to be used with the quotes.

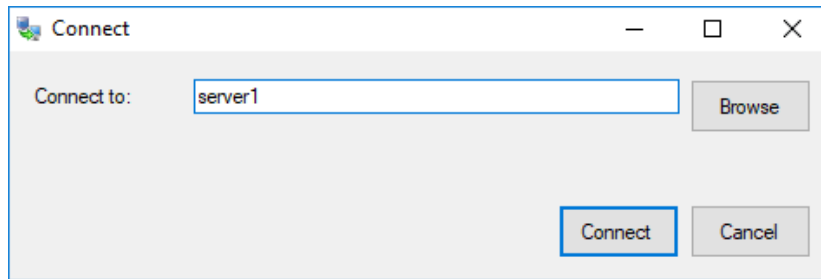
Edition Comparison

WinSyslog comes in different versions. Some of them are more feature-rich than others. The manual covers description about the full feature set. In order to remove confusion we have created a Product Comparison Sheet which identifies the differences between different available versions. [Click here](#) to see which services, actions and other features are provided by each version.

Connect to Computer



Click the Connect button in order to access another machine remotely. A window will open up.



Here you can enter the name of the machine you want to configure remotely. You can either directly enter the name into the textfield or you use the Browse button to see a list of available machines in the network. The click on the Connect button, the configuration client will verify access to the remote machine. If the verification is successful, you will be able to proceed with the remote access. Otherwise an error message will be shown.

Please Note: For remote configurations, you must ensure, that the remote machine is accessible by network and has access rights for the current logged on local user.

Registry Paths

Here are some more details regarding registry paths.

Since 64bit Windows re-routes all registry keys for 32bit programs automatically (HKEY_LOCAL_MACHINE\SOFTWARE\ to HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node), Adiscon decided in the course of development that the 32bit as well as 64bit registry keys of the service should use the ``HKEY_LOCAL_MACHINESOFTWAREWow6432Node`` subkey.

But for your case, this is rather problematic since you need a registry file for 32bit and for 64bit systems. Thus we decided to use a workaround, namely to use the parameters key in the service area. In the services subkey of the registry is no automatic mapping for Win32/64 applications and thus you can use the same registry file for all systems.

System Error Codes

In most of the cases where you will get system error codes, see this list for their meanings: [System Error Codes](#)

If you cannot find them there, please contact us via the [Customer Service System](#).

Information for a Mass Rollout

A mass rollout in this context means deploying an Adiscon client (for example MonitorWare Agent, EventReporter, or WinSyslog) to more than a handful of systems in an automated way. The aim is to invest the upfront effort needed to create a consistent “master” configuration once and then reuse it for every target machine. For guidance on differentiating between initial and update rollouts, see the MWAgent FAQ section.

Preparing the Baseline

1. Install the product on a single master system and configure it exactly as desired. Verify that the configuration works before continuing.
2. Export the configuration to a **registry file** via the configuration client (Computer → Export Settings).
3. Gather the files required for an engine-only installation:
 - For MonitorWare Agent 8.1 and newer this is typically `mwagent.exe` and `mwagent.pem`.
 - Older releases may also require the Visual C++ runtime and OpenSSL helper files (Microsoft.VC90.CRT.manifest, libeay32.dll, ssleay32.dll, msvcm90.dll, msvcp90.dll, msucr90.dll).

Automated Rollout Example

Once the master system is prepared, copy the required files to a network share or removable media and automate the rollout with a script similar to the following:

```
copy \\server\share\mwagent.exe C:\some-local-dir
copy \\server\share\mwagent.pem C:\some-local-dir
cd C:\some-local-dir
mwagent -i
regedit /s \\server\share\configParams.reg
net start "AdisconMonitorWareAgent"
```

`configParams.reg` represents the registry export taken from the master system. Because the rollout ships only the engine files, this approach works well for DMZ environments where RPC or file sharing cannot be opened.

Note

`mwagent -i` (or the equivalent command-line switch for other Adiscon products) only registers the Windows service. It assumes the binaries already exist in the current directory, so copy the files before running the command.

Branch Office Rollouts

For branch offices or semi-automated deployments, distribute the prepared package and have the local administrator perform the following steps:

1. Create a directory on the target computer and copy the provided files into it.
2. Run `mwagent -i` from that directory to register the service.
3. Import the exported configuration by double-clicking the `.reg` file (or by running `regedit /s` from an elevated command prompt).
4. Start the Windows service via `net start` or the Services management console. Restarting the entire machine is not required.

Important

The directory that hosts the engine files **is** the installation directory. Deleting it removes the binaries and effectively uninstalls the product.

Updating Existing Rollouts

To upgrade an engine-only installation, update the master system first, export the revised configuration, and distribute the refreshed files using the same process. Uninstallation is unnecessary as long as you overwrite the files in place, but always stop the Windows service before copying the new binaries. For a walkthrough focused on update scenarios, refer to the MWAgent FAQ section.

Glossary of Terms

Unfortunately, in the IT world terms are not necessarily used the same way by all people.

To clarify what we are talking about, we have created a glossary of terms as we at Adiscon understand them. Of course, we try hard to stay with the mainstream definitions.

This list includes both general IT terms as well as Adiscon-specific terms. We hope this glossary of terms will be useful for all interested parties.

Actions

Actions tell the Product (i.e. MonitorWare Agent or EventReporter or WinSyslog or any of the combinations) 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. These actions are described in the following section.

Write to File

The message is written to a plain text log file.

Write to Database

The message will be written to the specified ODBC database. This database format will be used by the MonitorWare Console that becomes available later. Therefore, if you intend to use the console, we recommend adding at least one rule that persists data to the database.

Write to EventLog

The message will be written to the application event log. Please note that the agent intentionally does not try to make the message look like it was generated on the local system. This could be very confusing. Instead, it is written inside the message part with standard values for event source and type.

Forward via Email

The message will be forwarded via email. Please note that each message will generate one email message. Messages are not combined to fit into a single mail. The Send Mail Action includes a timeout feature (`m_nTimeoutValue`) that provides control over message delivery timing.

Forward via Syslog

The message will be forwarded to a syslog daemon. UDP and TCP forwarding is supported.

Forward via SETP

The message will be forwarded via the custom SETP protocol. This is typically used in environments where data from different agents will be consolidated in a central place. SETP allows to transfer all InformationUnits exactly as they are. As such, the central repository can store an exact picture of the whole network.

Net Send

The message will be forwarded via the Windows “net send” functionality. Please note that the Windows function is not very reliable and requires the user to be logged in. As such, we recommend using “Net Send” only in combination with other actions.

Start Program

The message will be passed to an external process. The command line is specified in the action modifier.

Play Sound Action

This action allows you to play a sound file.

Send to Communications Port

This action allows you to send a string to an attached communication device, that is it sends a message through a Serial Port. It can send any message to a configured Serial or Printer port.

Set Status

This action allows you to create new properties of your own choice in the incoming messages. There is an internal Status List within the product which you can use for more complex filtering. You can set property over the Set Status action and you can add filter for them. They are more or less helpers for building complex rule constructions.

Set Property

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.

Call RuleSet

This Action simply calls another RuleSet in some existing RuleSet. When this Action is encountered, the Rule Engine leaves the normal flow and go to the called RuleSet (which may contain many rules as well). It executes all the rules that have been defined in that called RuleSet. After the execution of all of them, it will return to its point from where it left the original flow.

Discard

Please see the rules description below for a complete discussion. Effectively, the message will be discarded and any further processing of this information unit be stopped as soon as a “Discard” action is found.

Post-Process Event Action

The post process action allows you to re-parse a message after it has been processed e.g. Tab Delimited format. Such re-parsing is useful if you either have a non-standard syslog format or if you would like to extract specific properties from the message.

EventReporter

[EventReporter](#) is Adiscon’s solution to forward Windows Event Log entries from all supported Windows versions to a central system.

These central systems can be either [WinSyslog](#) , other Syslog daemons (e.g. on UNIX), or [MonitorWare Agent](#). EventReporter is part of adiscon’s monitorware line of products.

Filter Conditions

Filter conditions are used inside the rule engine. They help to decide when a rule is to be carried out. Filter conditions are considered to match of the outcome if the configured comparison operation is “TRUE”. Available filter conditions are listed down below:

- Global Conditions
- General Conditions
- Date / Time
- InformationUnit Type
- Syslog
- SNMP Traps

- Custom Property

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. These are:

- Treat not found Filters as TRUE*

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

- Fire only if Event occurs* - This is kind of the opposite of the “Minimum Wait

Time”. Here, multiple events must come in before a rule fires. For example 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 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.

- Minimum Wait Time* - This filter condition can be used to prevent rules from

firing too often. For example, a rule might be created to check the status of a port probe event. The port probe probes an smtp server. If the event is fired and the rule detects it, it will spawn a process that tries to restart the service. This process will take some time. Maybe the SMTP gateway need some more time to fully start up so that the port probe might fail again while the problem is already taken care of. The port probe as such will generate an additional event. Setting a minimum wait time will prevent this second port probe event to fire again if it is – let us 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.

Date Conditions

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

General Filter Conditions

This set includes filters which are related to Non-Event Log specific settings. These are:

- Source System* - This is the system a message is originated from. It can be used to check for authorized systems to pass messages to the MonitorWare Agent.
- Message Content* - The message content filter condition is very powerful. It evaluates to true if the specified content is found anywhere in the message. As there is implicit wildcarding, there is no need to specify extra wildcards.
- CustomerID* - CustomerID is provided for customer ease. For example if someone monitors his customer’s server, he can store different CustomerIDs in each agent. This is user configurable.
- SystemID* - SystemID is of type integer and is to be used by our customer. In addition, it is user configurable.
- Status Name and Value* - These filter type corresponds to set status action.

Date / Time

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

- Time* - This filter condition is used to check the period in which an event

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

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

Information Unit Type

This is based on the type of service that generated the information unit. So with this setting rules can be created that act only on e.g. syslog messages or NT event reports.

Syslog

Syslog related filters are grouped here:

- **Syslog Facility*** - For syslog information units, this is the actual syslog facility. If that filter condition is used on non-syslog originated information units, it will be a value mapped on a best effort basis to a syslog facility.
- **Syslog Priority*** - For syslog information units, this is the actual syslog priority. If that filter condition is used on non-syslog originated information units, it will be a value mapped on a best effort basis to a syslog priority.
- **Syslog Tag*** - The syslog tag value, is a short string. This is provided for non-syslog messages based on configuration. In most cases, this is used for filtering.

SNMP Traps

Using SNMP Traps MonitorWare Agent can be used to manage and monitor all sorts of equipment including computers, routers, wiring hubs etc. A trap is generated when the device feels it should do so and it contains the information that the device feels should be transmitted. Related filters are grouped here:

- **Community*** - It corresponds to the respective SNMP entity.
- **Enterprise*** - It corresponds to the respective SNMP entity.
- **Generic name*** - It corresponds to the respective SNMP entity.
- **Version*** - It corresponds to the respective SNMP entity.
- **Uptime*** - It corresponds to the respective SNMP entity.

Custom Property

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

FTP

FTP stands for File Transfer Protocol. FTP is the best means for moving large files across the Internet. FTP is a client/server protocol that enables a user with an FTP client to log on to a remote machine, navigate the file system of that remote machine, and upload and download files from that machine.

There are two basic types of FTP on the Internet anonymous ftp and private ftp. With anonymous ftp, one logs in as user anonymous, giving one's email address as a password. With private FTP, one logs in with the username and password one has established on that particular system. You are logged into your home directory, with all the file permissions you would normally have there.

HTTP

HyperText Transfer Protocol, the underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what action Web servers and browsers should take in response to various commands. For example, when you enter a URL in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page.

IETF

The IETF is an important Internet standards body. **IETF** is a short name for “Internet Engineering Task Force”. The IETF is responsible for the creation of RFCs. Unlike other, formal standards bodies it is loosely organized. There is no specific membership to the IETF, anyone (knowledgeable) can become an IETF member just by participating on the IETF discussion mailing lists.

The IETF itself provides a good overview over itself at <https://www.ietf.org/about/mission/>.

IMAP

Internet Message Access Protocol. A protocol allowing a client to access and manipulate electronic mail messages on a server. It permits manipulation of remote message folders (mailboxes), in a way that is functionally equivalent to local mailboxes.

IMAP includes operations for creating, deleting, and renaming mailboxes; checking for new messages; permanently removing messages; searching; and selective fetching of message attributes, texts, and portions thereof. It does not specify a means of posting mail; this function is handled by a mail transfer protocol such as SMTP.

Information Units

Information units contain the data gathered by the services. As soon as a service detects a reportable event, it creates a new information unit. The information unit contains a textual representation of the event (for example a syslog message) as well as information about the event itself. For example, it contains the system that the event was originated from and the date and time it was received.

Which data is contained in the information unit depends on its type. However, there are a number of common data elements present in all information units. Most of these elements can be used as filter conditions in the rule engine. Information unit specific data elements are not eligible as filter conditions. However, there are data elements (properties) which are defined to be present in all information units even though they seem to be specific to a service type. One example is syslog priority. These values are present in each information unit type simply because priority is a good abstraction for other types, too. Such generally available properties are mapped if they are not directly supported by the service type. In the example, an Event Log Monitor maps the event log severity to the syslog priority.

There is a direct one-to-one relation between service type and information unit type. Each service type has its own information unit type.

Inside the rule base, the information unit type itself can be used as a filter condition. This facilitates creating rules that check information unit type specific properties only if they originated from the specific service type (e.g. check syslog priority only if the information unit was generated by a Syslog server).

IPv6

Adiscon Products officially support IPv6. The IPv6 support was introduced with the following versions:

- MonitorWare Agent 8.0
- WinSyslog 11.0
- EventReporter 12.0

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

Millisecond

A millisecond is a thousand of a second. It is abbreviated as “ms”. As such, 500ms mean half a second.

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

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:

- [EventReporter](#)
- [MonitorWare Agent](#)
- [WinSyslog](#)
- [Rsyslog Windows Agent](#)

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

- [Liblogging](#)

New products are continuously being added - please be sure to check <https://www.adiscon.com/products/> from time to time for update

NNTP

NNTP stands for Network News Transport Protocol. This protocol is used by client and server software to carry USENET postings back and forth over a TCP/IP network.

When you are using any of the common software like modern web browsers or newsreaders, you are taking benefit of the NNTP connection to participate in newsgroups.

POP3

POP3 (Post Office Protocol 3) is the most recent version of a standard protocol for receiving e-mail. POP3 is a client/server protocol in which e-mail is received and held for you by your Internet server. Periodically, you (or your client e-mail receiver) check your mail-box on the server and download any mail. It is also built into modern web browsers and email clients.

RELP

RELP is the “Reliable Event Logging Protocol”. It assures that no message is lost in transit, not even when connections breaks and a peer becomes unavailable. The current version of the RELP protocol has a minimal window of opportunity for message duplication after a session has been broken due to network problems. In this case, a few messages may be duplicated (a problem that also exists with plain tcp syslog).

RELP addresses many shortcomings of the traditional plain tcp syslog protocol. For some insight into that, please have a look at <https://rainer.gerhards.net/2008/04/on-unreliability-of-plain-tcp-syslog.html>. Please note that RELP is currently a proprietary protocol. So the number of interoperable implementations is limited.

Note that for reliable operation where messages should be preserved over a service shutdown, queue cache mode must be activated.

Resource ID

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

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

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

RFC 3164

RFC 3164 is a [IETF](#) document. It describes how [syslog](#) messages have been seen in traditional implementations. RFC 3164 is not a standard but rather a descriptive (“informational” in IETF terms) document. It does not demand a specific behavior but rather documents what has been seen. Some existing implementations of real-world syslog use different formats.

RFC 3164 is just the first step towards a newer and better syslog standard. A standard already produced by this working group is rfc 3195, which describes how syslog can be sent reliably over a tcp connection.

Adiscon supports RFC 3164 messages. There are a number of switches in each product to take care of those implementation that do it slightly different.

The formal specification for RFC 3164 can be found in the [IETF RFC](#) repository.

RFC 3195

RFC 3195 is an [IETF](#) standard. It specifies how [syslog](#) messages can reliably be transmitted via a tcp connection. RFC 3195 optionally allows for message encryption and authentication of sender and receiver. However, it has not receive any importance in practice. Servers are hard to find.

adiscon's monitorware line of products implement the core RFC 3195 protocol (actually, [Adiscon was the first one to do this on the Windows platform](#)). Under UNIX [Rsyslog](#) and [SDSC syslog](#) are known to support RFC 3195. Our [liblogging](#) project enables your own applications to “talk” 3195.

The formal specification for RFC 3195 can be found in the [IETF RFC repository](#) .

During its creation, RFC 3195 was known as “syslog-reliable”. Many people still use this name to refer to it.

RFC 5424

RFC 5424 is a [IETF](#) document.

This document describes the syslog protocol, which is used to convey event notification messages. This protocol utilizes a layered architecture, which allows the use of any number of transport protocols for transmission of [syslog](#) messages. It also provides a message format that allows vendor-specific extensions to be provided in a structured way.

A standard already produced by this working group is rfc 3195, which describes how syslog can be sent reliably over a tcp connection.

Adiscon supports RFC 5424 messages. There are a number of switches in each product to take care of those implementation that do it slightly different.

The formal specification for RFC 5424 can be found in the [IETF RFC](#) repository.

Rules

Rules are the workhorse of the MonitorWare Agent. All actions and processing carried out is configured by the rules defined. Rules are configured by the client and processed by the so-called “rule engine” inside the MonitorWare Agent service.

You might already know something similar to the MonitorWare Agent rule engine. Rule engines and rule bases are an extremely powerful tool and in widespread use in the industry. Examples of rule bases can be found at Checkpoint's Firewall One Firewall Rule Base or Cisco Routing filter - just to name a few.

The rule base consists of the rules as configured in the client. The rule engine is the process carrying out the rules. A rule base can contain no, one or an unlimited number of rules. However, if there is no rule at all defined, no action will ever be carried out by the agent. Consequently, the client will issue a warning message in this case.

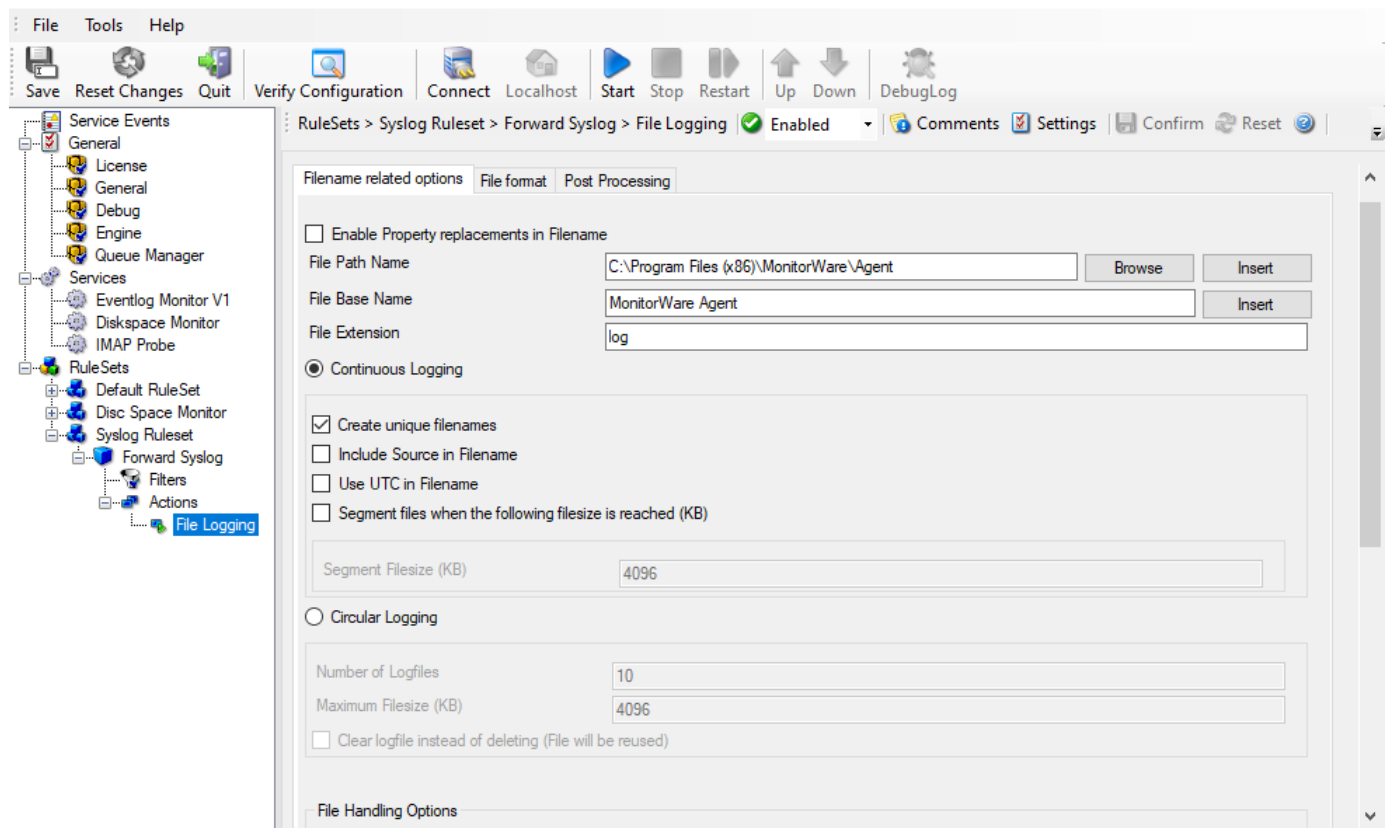
A rule has a description, associated match conditions, and actions. The match conditions are called “filter conditions”. These specify when a rule is to be carried out. Again, there can be no, one, or many filter conditions for a

single rule. If there are no filter conditions, the rule will always match. This is useful in many cases. If there is more than one filter condition, all filter conditions need to match in order for the rule to match (logical AND).

Actions associated with a rule specify what to do when the associated rule matches (and only the associated rule). Actions carry out the actual processing of messages. For example, actions include logging a message to a flat file or database, sending it via email or forwarding it to syslog daemon or another MonitorWare Agent. There can be no, one, or an unlimited number of actions associated with a rule. However, if no action is associated, the rule will not have any effect. Consequently, the client will issue a warning when writing the rule base. Rules without actions can be useful to temporarily disable a rule with complex filter condition. If there are multiple actions, they are not guaranteed to be carried out in any specific order. If you definitely need an action to be carried out before another one, you currently need to define two rules.

Actions can be modified with action modifiers. These are the strings attached to a specific action. Action modifiers allow customizing a specific behavior of this action. It modifies only this action and only this one, other actions of the same type are not affected - regardless if they appear in the same rule or a very different one. The use of the action modifier depends on the type of action. For example, with syslog forwarding it is the host the syslog message is to be forwarded to. With ODBC database logging it is the DSN and so on. If there is no action modifier, the values configured in the client's configuration tabs will be used. They are also used for all values that cannot be modified via the action modifier (e.g. the SMTP server address for email forwarding).

Below find a screenshot of a rule base with a number of rules, filter conditions and action modifiers:



Sample Rule Base

Now that we know the elements, how are rules being processed. It is easy. Rules are strictly processed from top to bottom, or from number one to the last one. For each rule the filter conditions are checked to see if they match. If they do, all associated actions are carried out. Then, the rule engine advances to the next configured rule. Once again, it checks if it matches and - if it does - carries out the actions associated with that rule. Then the next rule is

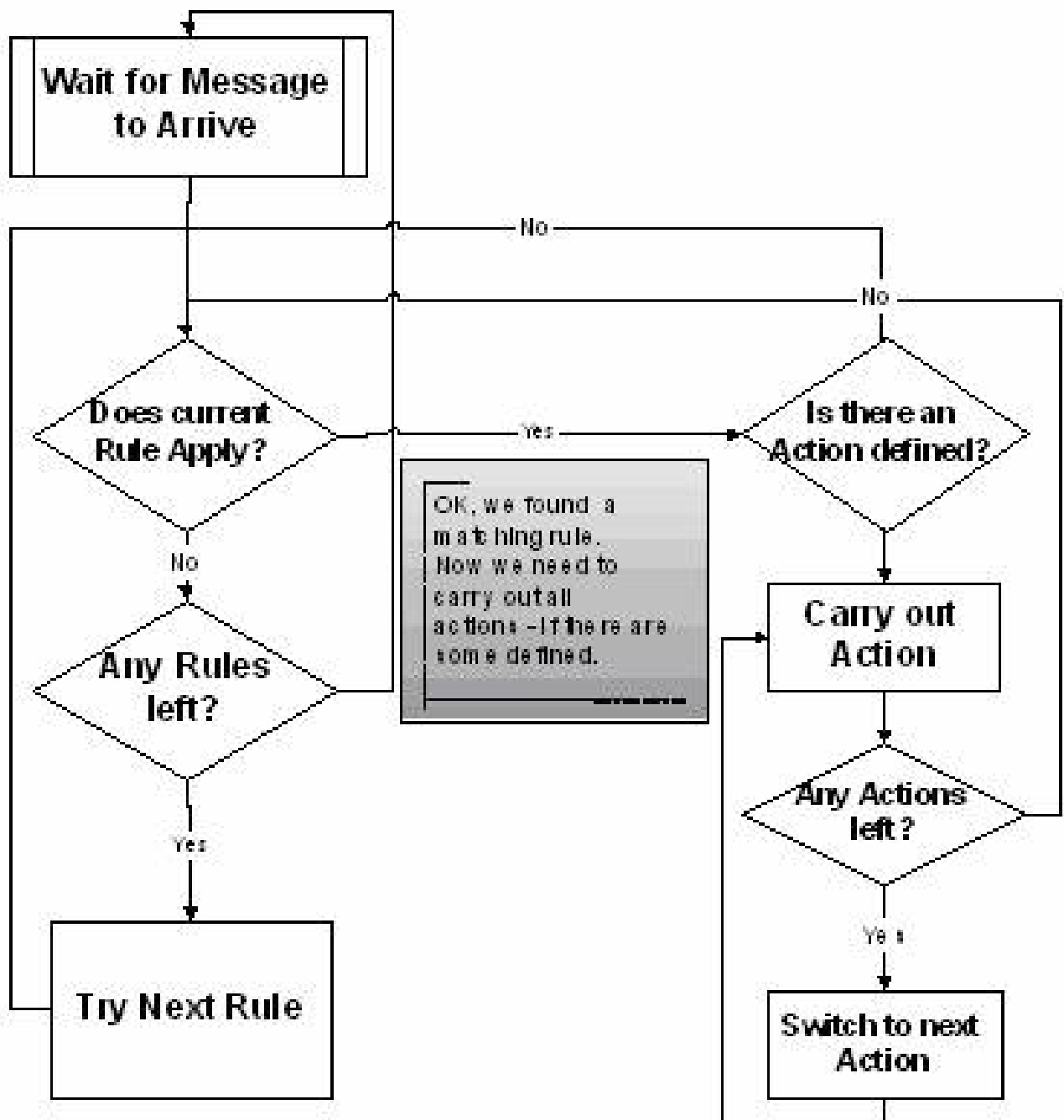
processed and so on. The rule engine stops when there are no more rules to be evaluated. It also stops if a rule contains a "discard" action.

The "discard action" is a very special and powerful action. It does not actually carry out any processing. In fact, it disables all further processing for a message as soon as it is found by the rule engine. So what is the discard action good for? It is used to handle common situations where a number of well know messages - unimportant messages - should be filtered out so that the other rules do not need to take care of these messages. In many other products using rules bases, this is called the "block rule". Please note that with Adiscon's rule engine, there can be multiple block rules at multiple layers of the rule base giving you additional flexibility.

One last thing to mention: the rule base is applied to every message arriving at the MonitorWare Agent. By design, there is no way to modify the behavior of the rule base for the next message to be arrived. This ensures an always consistent processing of incoming messages. However, there can be multiple rule bases. Each rule base is associated with a service. Only the rule base associated with the service generating the message will be processed.

While building and testing your rule base, please keep in mind that the MonitorWare Agent service needs to be restarted to load a modified rule base. The reason is that the service does not re-read the rule base to save system resources.

There is an online seminar available on the rule engine and its processing. If you are interested in a more in-depth view, you might want to visit it at rule engine.



Rule Engine Flowchart

For those interested in more in-depth information on how the rule engine works, this flowchart might be helpful:

There is an online seminar available on the rule engine and its processing. If you are interested in a more in-depth view, you might want to visit it at rule engine.

The Rule Engine

Overview

This paper explains you the Rule Engine that is employed in some of the MonitorWare Line of Products namely MonitorWare Agent, WinSyslog, and EventReporter 6.0 (and higher)

What is the Rule Engine

Rule Engine is actually an engine present in the above mentioned MonitorWare Line of Products using which you can define certain filters and the actions that are to be carried out if the defined filter condition matches with the real time condition.

Rule Engine revolves around four basic concepts:

- Information Unit
- Information Services
- RuleSets
- Queue Manager

In order to understand the complete Rule Engine, you need to understand the above mentioned four concepts. The details of these are written below

1. Information Unit (Info Unit)

“Information Unit” or “Info Unit”, as we call it, is the basic building block of Rule Engine. Info Unit is basically an object that contains all the information about a specific event which includes:

- Message
- Which application generated this event
- When this event was generated
- Syslog Facility
- Syslog Priority
- Info Unit Type (it tells which Info Service has generated this Info Unit)
- etc

The following figure will give you an idea about an Info Unit:

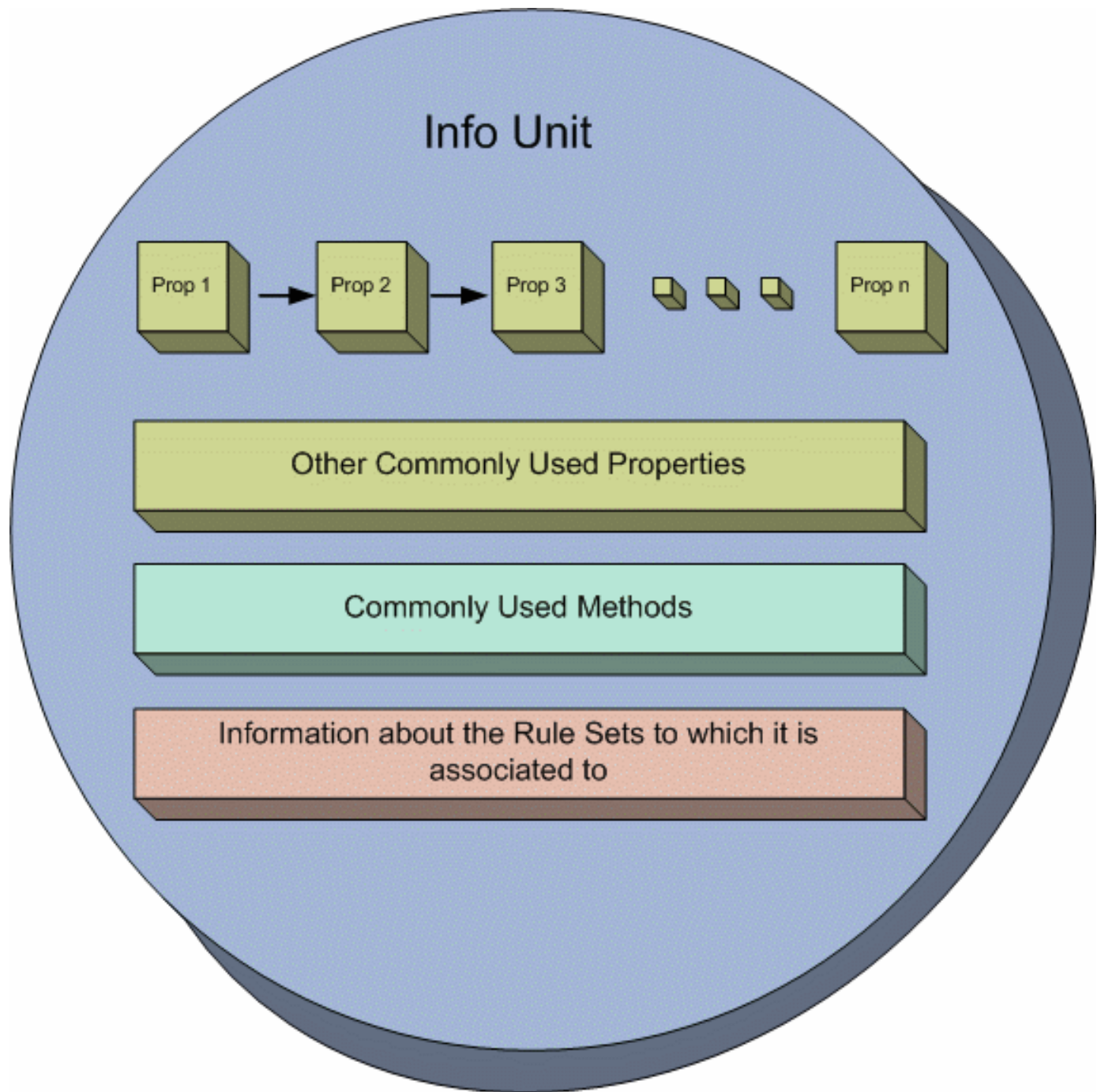


Figure 1: Conceptual Diagram of an Info Unit

As the figure illustrates, an Info Unit contains most of the properties (mentioned above in bullets) in the form of a list. In addition to this list, it also has some commonly used properties separately stored in it (for efficiency reasons). Apart from the properties, an Info Unit also has some methods which allow it to write it or to construct itself etc. Information about the RuleSets (will be explained in the coming sections) is also contained within each Info Unit so that it exactly knows which rules will be applied on it.

2. Information Services (Info Service)

“Information Services” or “Info Services”, as we call them, generate Info Units. Each Information Service will generate its own Info Units. The important thing to note over here is that each Info Unit has the same format but can have different properties and rulesets associated with it. For example, if an architect makes a building plan then it becomes a template. Now he can use this template to construct as many buildings as he likes but each one can have different properties (they can differ in color scheme, window styles etc). Exactly in the similar way, an Info Unit is actually a template from which each Info Service makes a specific object of Info Unit that might differ in properties from another Info Unit object.

Examples of Info Services

There can be a number of different examples on Info Services. Following are some of the examples:

1. Syslog server

It receives the messages that are forwarded to it and for each message (or event) it generates an Info Unit out of it.

2. Event Log Monitor

It picks up the events from the Window's Event Log and for each event it constructs an Info Unit.

3. Ping Probe

It pings a specified device and if doesn't find a response from the other side, it generates an Info Unit with desired information.

Important Note

One thing to note about Info Services is that there can a number of different Services running on the same machine. You can even run the different instances of the same Info Service (but with different properties naturally). In either case, each Info Service will generate its own Info Unit.

3. RuleSets

As the name suggests, a RuleSet is a set of Rules. A "Rule" consists of the following two things

- Filter Condition
- Actions

You might have noticed that the point 1 written above is singular and point 2 is plural which clearly means that you can define only one Filter condition for one rule but can define as many actions as you like. The filter condition can however contain as many Boolean operators as you like.

Filter Condition

Filter Condition is a combination of different Boolean operators which will evaluate to a Boolean answer. In simple words, the result of a filter condition can either be True or False.

Actions

Actions are all those events which are fired when a filter condition evaluates to a True value. As mentioned above, a Rule can have more than one actions associated with it which means that if a filter condition evaluates to a true value then all of the actions associated with that rule will execute. If the filter condition, on the other hand, evaluates to a false value then all of the defined actions will be skipped.

Note that other than normal actions, there are three special kinds of Actions that are worth mentioning here:

- Discard Action (Explained Later)
- Include Action (Explained Later)
- Actions that can alter the contents of Info Units permanently

4. Queue Manager

Queue Manager simply maintains a queue of all of the Info Units that have been forwarded to it by different Info Services.

Overall Picture

This section will explain you that how the different components are related to each other and how does the whole process work. The picture shown below gives an idea about how things are working. As you can see that we have four different stages through which the events are processed.

Info Services picks up the events and convert them into Info Unit. Note that each Info Service has its own Info Unit. These Info Units are passed to the Queue Manager. The job of the Queue Manager, as mentioned above and as

clear from the diagram, is to simply make a queue of these Info Units that it has received from various Info Services. The Rule Engine picks up the Info Units from this Queue Manager, applies the rules on these Info Units (as mentioned above, each Info Unit has the information about which rules should be applied on it) and if necessary carries on the actions. The rule engine keeps on repeating this process while there are some Info Units present in the Queue.

Manager's Queue

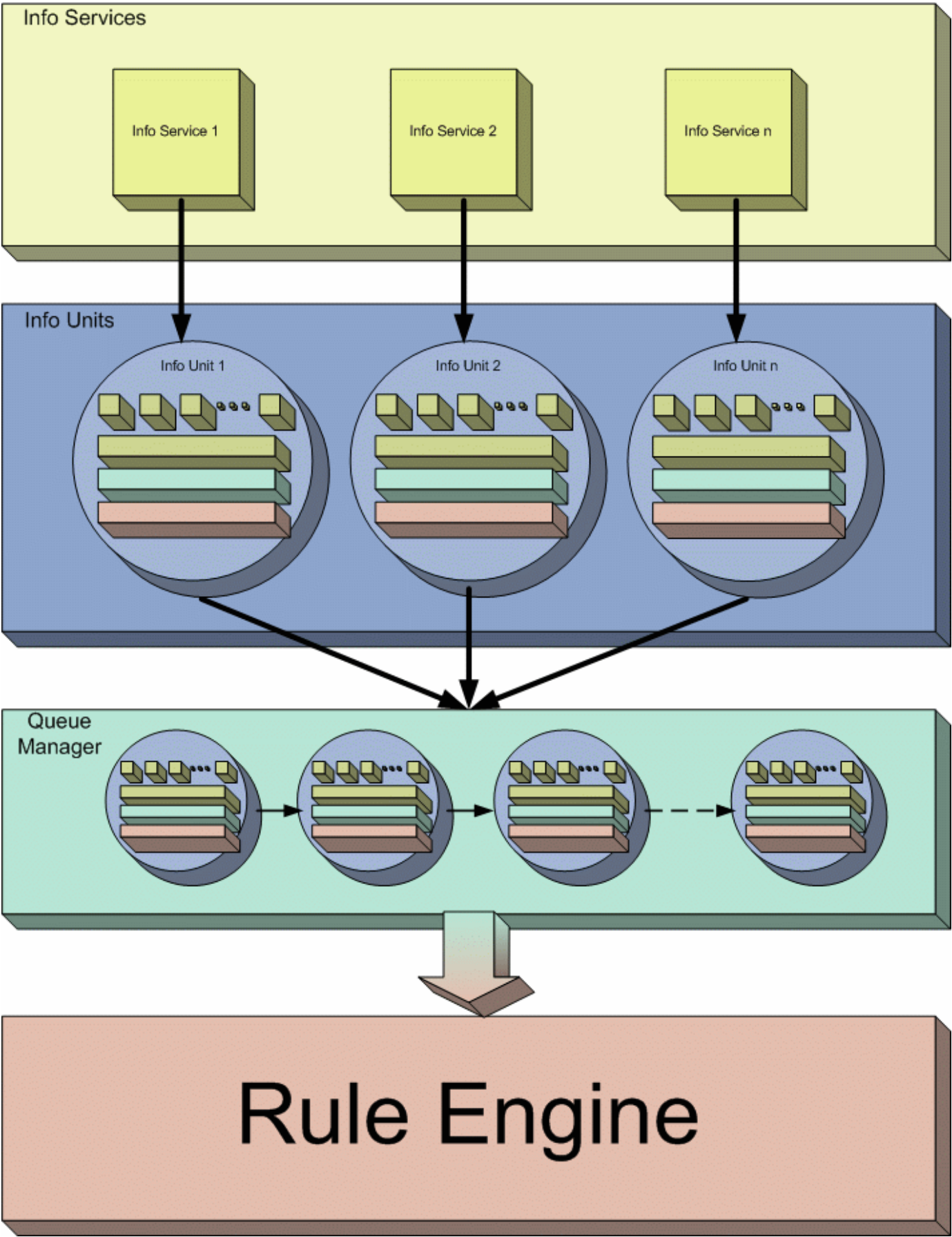


Figure 2: Overall Process

How Does the Rule Engine Work

Having explained the overall picture of the whole process, let's specifically talk about Rule Engine. The following figure explains it in detail:

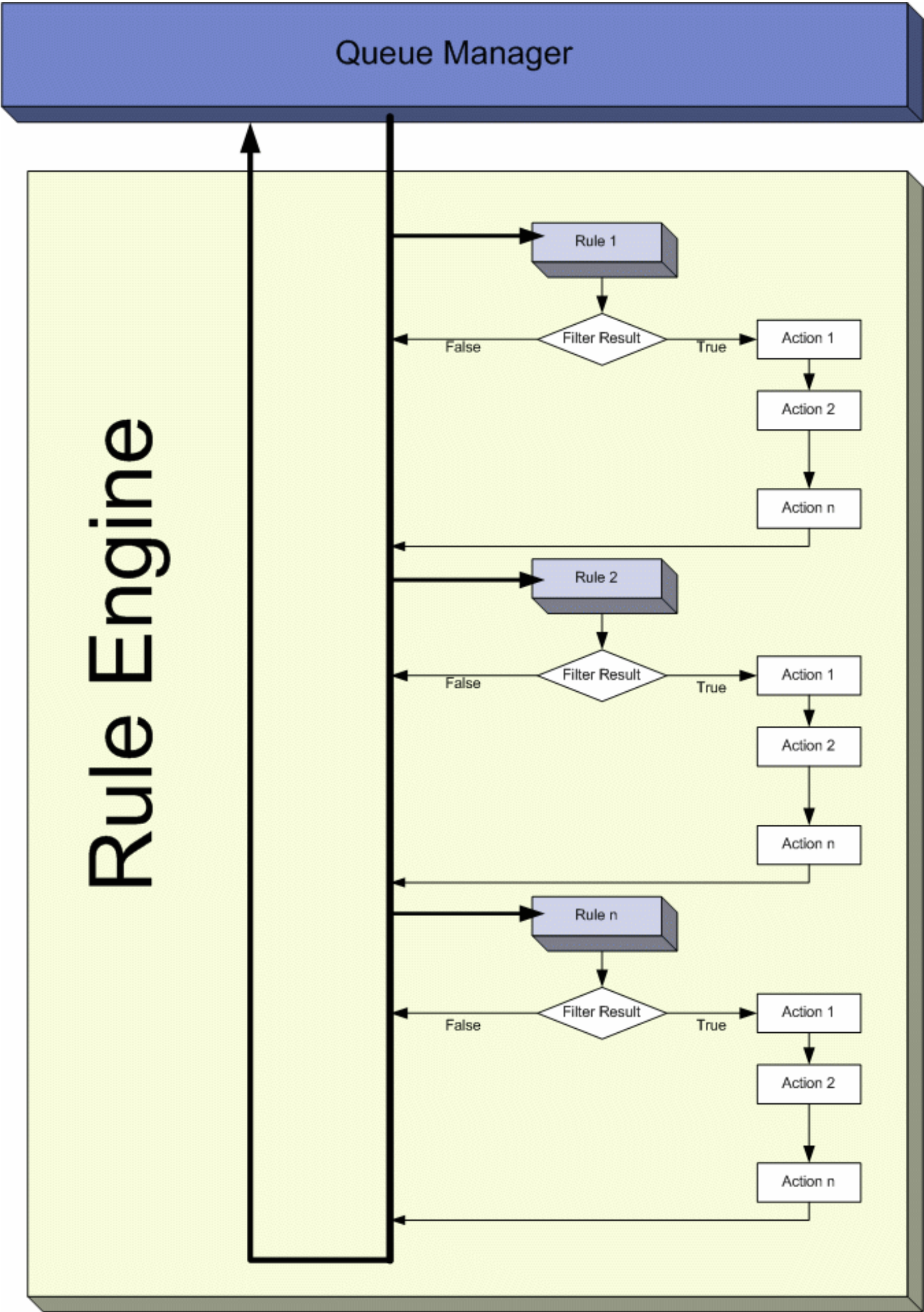


Figure 3: Working of Rule Engine

As you can see in the figure, the Rule Engine picks up Info Units one by one from the Queue Manager. Since each Info Unit has the information about its Rule sets, it will apply the rules on it in the same order in which they were defined. As you can see above, it will pick up the first Rule and evaluates its Filter Condition. If that Filter Condition is evaluated to false, all the actions associated with that rule will be skipped and it will pick up the second Rule. If the Filter Condition, on the other hand, evaluates to True, it will execute all the actions that are associated with this Rule in the order in which they were defined. After the execution of all these actions, it will pick up the next rule in the current ruleset. Once all the rules have been executed, the current Info Unit (that was handed over to Rule Engine by the Queue Manager) will be destroyed and the Rule Engine will go to the Queue Manager to pick up the next Info Unit if there exists one.

The above picture has been drawn for normal flow of executions. There can be 2 conditions when the flow will not follow the diagram shown above. These conditions arise in response to 2 special kinds of actions that are called Discard Action and Include Action.

Discard Action

A Discard Action immediately destroys the current Info Unit and any action of any Rule that has been defined after the Discard Action will not be executed at all. Let's take a simple example to clarify it further.

Let's say that Action 2 of Rule 1 in the picture above is a Discard Action. If the Filter Result of Rule 1 is evaluated to true, then Action 1 will be executed. As Action 2 is a Discard Action, immediately the current Info Unit will be destroyed (which means that now the Rule Engine will skip all the Rules and all the actions associated with them) and the Rule Engine will go back to the Queue Manager to pick up the next Info Unit in the Queue.

Include Action

An Include Action simply includes another RuleSet in some existing RuleSet. When this Action is encountered, the Rule Engine leaves the normal flow and go to the included ruleset (which may contain many rules as well). It executes all the rules that have been defined in that included RuleSet. After the execution of all of them, it will return to its point from where it left the original flow. Let's take an example to clarify it a little further.

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

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

Suggestions for Defining Complex RuleSets

While defining a complex RuleSet, it might be a good idea to follow the stages defined below.

Edit Stage # - Actions Stage 0 - Discard unwanted events Stage 1 - Post Process Stage 2 - Discard unwanted events Stage 3 - All Actions Stage 4 - Individual Actions

As mentioned above, the rules and actions will be executed in the order in which you will define them. So it's very important that you define the actions in a way such that you achieve the desired results as well as achieve them with efficiency. For example, if you haven't defined any filter which we call as No Filter (it always evaluates to true) and if the first action that you have defined is the Discard Action, then there is no meaning of defining any action after this first action because the first action will always be executed and it will always discard the complete Info Unit.

Here is the explanation of the above mentioned stages.

Stage 0

In this stage, you can discard those events that you are not interested in. You can use the Discard Action explained above to discard the events.

Stage 1

In this stage, we recommend to Post Process the incoming Info Units. Once the Info Unit has been handed over to the Rule Engine from the Queue Manager, you can actually change the contents of the Info Units to make them more meaningful.

Stage 2

In this stage, you might want to again discard those events that you are not interested in. Simply use the Discard Action.

Stage 3

In this stage, you will apply the actions that will apply to all of the Info Units coming (to be more specific, you will apply those rules over here for which you have selected “No Filter” as the filter condition.

Stage 4

In this stage, you will create the rules for which you have specific filter conditions.

To sum it up, we recommend doing most generic things first and least generic things later or in other words, do the generic things first and the specific things later. Note that this section suggests only the typical scenario but it can vary from depending upon the needs. For example, you might want to perform some actions on some specific events after stage 1 and before stage 2.

WinSyslog - Services

Services inside the WinSyslog gather the data that is processed by rules. Each service type reflects a specific set of code inside WinSyslog. For example, a Syslog Service represents an instance of a Syslog server and an NT Event Log Monitor Service represents an instance of an NT Log Monitor (periodically pulling out log information).

Typically, there can be multiple instances of the same service running, as long as their configuration parameters do not conflict. For example the syslog service: there can be multiple syslog servers on a given system as long as they listen to different ports. Consequently, there can be multiple instances of the syslog service be created. For example, there could be three of them: two listen to the default port of 514, but one with TCP and one with UDP, and a third one listens to UDP, port 10514. All three coexist and run at the same time.

The following services are supported:

Heartbeat

This service generates a special information type. Its primary purpose is to notify a receiving system that WinSyslog, set for heart beating is still alive. So the receiving system can be configured to raise alarms (or corrective actions) if it does not receive heartbeats from WinSyslog.

MonitorWare Echo Reply

A central agent running the MonitorWare Agent is using the echo request and instructs to poll each of the other WinSyslog services. When the request is not carried out successfully, an alert is generated. The MonitorWare echo protocol

RELP Listener

The RELP listener supports the new reliable event logging protocol (RELP), which enables a more reliable transmission of messages than plain tcp syslog protocol. The service permits to accept messages from senders who themselves support RELP. The RELP Listener will automatically listen on all available IP Addresses which includes IPv4 and IPv6. This is due the librelp implementation method.

Apart from the fact that a different communication protocol is used, the RELP listener corresponds functionally to the syslog listener. The RELP listener automatically monitors all available IP addresses, including IPv4 and IPv6. This is due to the Librelp implementation method.

SETP server

Implements an SETP Server. It is used for reliable receiving event notifications.

SNMP Trap Receiver

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.

Syslog server

Implements a Syslog server. It can be set to listen to any valid port. UDP and TCP communication is supported.

Event Log Monitor

Monitors Windows event logs. As soon as new events are detected, these are forwarded to MonitorWare processing. This service is similar to the Adiscon EventReporter functionality.

Associated rulesets

Each instance of a service has an associated ruleset. This allows easy creation of customized rulesets on a per service basis. Of course, all services can also operate on a common ruleset.

All services are executed as multiple threads inside the MonitorWare Agent. From the operating point of view, there is only one system service called the "MonitorWare Agent". If the service configuration of the MonitorWare Agent is modified, the MonitorWare system service needs to be restarted in order to activate the new configuration. Later releases will have some options to automate this task.

SETP

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

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

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

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

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 cannot 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.

SNMP

SNMP stands for Simple Network Management Protocol. A set of standards for communication with devices connected to a TCP/IP network, like routers, hubs and switches. A device is said to be SNMP compatible if it can be monitored and/or controlled using SNMP messages.

SNMP messages are known as PDU's - Protocol Data Units. Devices that are SNMP compatible contain SNMP 'agent' software to receive, send, and act upon SNMP messages. Software for managing devices via SNMP are available for every kind of commonly used computer and are often bundled along with the device they are designed to manage. Some SNMP software is designed to handle a wide variety of devices.

Syslog Facility

Syslog Facility is one information field associated with a syslog message. It is defined by the [Syslog](#) protocol. It is meant to provide a very rough clue from what part of a system the message originated from. Traditionally, under UNIX, there are facilities like KERN (the OS kernel itself), LPD (the line printer daemon), and so on. There are also the *LOCAL_0* to *LOCAL_7* facilities, which were traditionally reserved for administrator and application use.

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

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

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.

Here you find information about Performance [Tests and Results](#)

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.

Here you find information about Performance [Tests and Results](#)

Upgrade Insurance

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

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

UTC

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

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

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

Copyrights

This documentation as well as the actual MonitorWare Agent product is copyrighted by Adiscon GmbH, Germany. To learn more about other Adiscon products, please visit <https://www.adiscon.com/en/products>.

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

Openssl-3.2.1:	https://www.openssl.org/source/openssl-3.2.1.tar.gz	Liblogging	0.7.1:
	https://github.com/Rsyslog/liblogging/archive/refs/tags/v0.7.1.tar.gz	Librelp	1.11.0:
	https://github.com/Rsyslog/librelp/archive/refs/tags/v1.10.0.tar.gz		Libfastjson-0.99.8:
	https://github.com/Rsyslog/libfastjson/archive/refs/tags/v0.99.8.tar.gz		

Liblognorm 0.3.5 <https://github.com/Rsyslog/liblognorm/archive/refs/tags/v0.3.5.tar.gz>

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

Other mentioned trademarks are for reference only. They belong to their respective owners.

- **genindex**

Index

A

[Accessing Properties](#)
[Add-on Components](#)

C

[Command Line Switches](#)
[Comparison of Properties](#)
[Complex Filter Conditions](#)
[Components](#)
[Connect to Computer](#)
[Customer Properties](#)

D

[Database Monitor](#)

E

[Event Properties](#)
[Event-Specific Properties](#)
[EventReporter](#)

F

[FromPos](#)
[FTP](#)

H

[HTTP](#)

I

[IETF](#)
[IMAP](#)
[Information Units](#)
[Installation](#)
[IPv6](#)

M

[Millisecond](#)
[MonitorWare Line of Products](#)

N

[NNTP](#)

O

[Options](#)

P

[POP3](#)
[Property](#)

R

[Registry Paths](#)
[RELP](#)
[Resource ID](#)
RFC
 [RFC 3195](#)
[RFC 3164](#)
[RFC 3195](#)
[RFC 5424](#)
[Rule Engine](#)
[Rules](#)

S

[SETP](#)
[SMTP](#)
[SNMP](#)
[Standard Properties](#)
[Syslog Facility](#)
[Syslog Message Properties](#)
[System Properties](#)
[System Requirements](#)

T

[TCP](#)
[ToPos](#)

U

[UDP](#)
[UpgradeInsurance](#)
[UTC](#)

W

[Windows Event Log Properties](#)
[Windows Event Log V2 Properties](#)