rsyslog Windows Agent Configuration Documentation

version 26.10

Adiscon GmbH

October 07, 2026

Contents

Manual **1**

- Getting Started 1
  - Installation 1
  - System Requirements 2
  - Understand the Components 3
  - Collect and Forward Windows Events 3
  - Creating an Initial Configuration 4
  - Process and Filter 5
  - Store and Forward 6
  - Operate and Troubleshoot 6
- Tutorials 7
  - Tutorial: Forward Windows Events to rsyslog 7
  - Tutorial: Forward Windows Events via TLS to rsyslog 8
  - Tutorial: Forward Windows Events via RELP to rsyslog 10
  - Tutorial: Export the Configuration and Create a Debug Log 10
- Configuration 11
  - Core concepts 12
    - rsyslog Windows Agent Services 13
    - Information Units 14
    - Rules 15
    - The Rule Engine 18
    - rsyslog Windows Agent Filter Conditions 26
    - rsyslog Windows Agent Actions 27
  - Configure rsyslog Windows Agent 27
  - Multiple RuleSets - Rules - Actions 28
  - Client Options 29
  - Client Tools 31
  - Using File based configuration 36
  - General Options 41
    - License 42
    - General 43
    - Debug 47
    - Engine 49
    - Queue Manager 54
  - Services 55
    - Heartbeat 57
    - MonitorWare Echo Reply 59
    - Syslog server 60
    - Event Log Monitor V1 70
    - Event Log Monitor V2 82

File Monitor 91
Filter Conditions 101
Global Conditions 103
Date Conditions 104
Operators 105
Filters 106
General 107
Date/Time 109
InformationUnit Type 110
Syslog 111
Event Log Monitor 112
Event Log Monitor V2 114
File Monitor 116
Custom Property 117
Extended Number Property 118
Extended IP Property 119
File Exists 121
Store Filter Results 122
Actions 122
Send RELP 123
Syslog Forwarding 128
Send DTLS 141
Call RuleSet 143
Compute Status Variable 144
Discard 145
Normalize Event 146
Resolve Hostname Action 147
Message Ringbuffer Action 148
Set Property 150
Set Status 151
FAQ 151
Are rsyslog Windows Agent products affected by recent OpenSSL CVEs? 152
Troubleshooting the Start Program action in rsyslog 152
High-load configuration reload stalls in rsyslog Windows Agent 154
How to Export rsyslog Windows Agent Settings for a Support Call 156
Why did my rsyslog Windows Agent update report success when the files were not upgraded correctly? 156
How to resolve performance issues on high-load systems? 164
Is MariaDB supported by the ODBC action? 167
How Do I Prepare rsyslog Windows Agent for Repeatable Deployment? 168
Why did my rsyslog Windows Agent update report success when the files were not upgraded correctly? 169

How Can I Copy an rsyslog Windows Agent Configuration to Another System? 176
How Can I Obtain a Printable Version of the rsyslog Windows Agent Manual? 177
What do CA PEM, Certificate PEM, and Key PEM mean for TLS input services? 177
How Do I Enter rsyslog Windows Agent License Information? 178
Why did product versions jump to 26.07? 178
What happens when the client opens a configuration with unsupported blocks? 179
Do the configuration clients require .NET Framework, or is .NET Core or .NET 5+ enough? 179
Sales 180
How do I contact Adiscon sales? 181
What should I include in a quote request? 181
What happens after I open a sales ticket? 182
How do purchase orders and billing requests work? 183
Licensing and ordering 184
Can I use an older product version with my license? 185
Air-gapped environments 186
Offline installation and activation 186
Online verification after activation 187
Perpetual licenses and UpgradeInsurance 187
UpgradeInsurance 188
Reference 189
rsyslog Windows Agent Shortcut Keys 189
Command Line Switches 189
Edition Comparison 190
Event Properties 191
Accessing Properties 192
System Properties 197
Custom Properties 198
Event-Specific Properties 199
Complex Filter Conditions 206
Connect to Computer 209
System Error Codes 210
User Interface changes 210
Glossary 212
FTP 213
HTTP 214
IETF 215
IMAP 216
Information Units 217
IPv6 218
Millisecond 219
NNTP 220
POP3 221

RELP 222
Resource ID 223
RFC 3164 224
RFC 3195 225
RFC 5424 226
Rules 227
The Rule Engine 230
SETP 238
SMTP 239
SNMP 240
Syslog Facility 241
TCP 242
UDP 243
UTC 244
rsyslog Windows Agent product Event ID reference 244
rsyslog Windows Agent Event ID troubleshooting procedures 245
Known issues and errata 253
Copyrights 253

About rsyslog Windows Agent

rsyslog Windows Agent collects Windows-originated events and forwards them to rsyslog and other downstream receivers. It is built for Windows systems that need reliable event forwarding, local filtering, and integration with Linux- or rsyslog-centered logging environments.

The product runs on the Windows systems that generate or relay events. It can collect Windows Event Log data, monitor text-based log files, receive syslog messages, and then process those events through rules before forwarding them.

rsyslog Windows Agent is especially useful when you want Windows systems to feed a central rsyslog deployment with better structure and control than a simple one-shot forwarder can provide. It supports syslog forwarding over UDP and TCP, secure forwarding with TLS, and RELP for scenarios where reliable transport matters.

The product is centered on two main runtime roles:

- the **rsyslog Windows Agent service**, which collects and forwards events in the background
- the **rsyslog Windows Agent Configuration Client**, which defines services, rulesets, filters, and forwarding actions

Inside the product, data enters through configured **services**. In plain language, you can read those services as the configured inputs that collect or receive events before rulesets and actions process them.

This manual covers installation, first-time setup, configuration, tutorials, FAQ content, and reference material for rsyslog Windows Agent.

Manual

Getting Started

Start here to understand what rsyslog Windows Agent is, how its main components work together, how Windows events move through it, and how to build a first working forwarding configuration.

Installation

Use this page to install rsyslog Windows Agent and prepare the system for initial configuration.

Before you begin

Make sure that:

- the target system runs a supported version of Windows
- you have local administrative rights for the installation
- the system can install Microsoft .NET Framework 4.7.2 or a newer .NET Framework 4.x release for the Configuration Client if it is not yet present
- you know whether this system will run the full product or an engine-only deployment
- if this is an engine-only target, the Configuration Client is not required on that system and the .NET Framework requirement applies where the client is installed and used

Installation steps

1. Download the current installer from the rsyslog Windows Agent download page.
2. Run the installer with administrative rights.
3. Select the components you want to install.
    - The **rsyslog Windows Agent service** is the runtime component.
    - The **rsyslog Windows Agent Configuration Client** is the administrative UI.
4. Finish the installer.
5. Start the rsyslog Windows Agent Configuration Client.
6. Create or review the initial configuration.
7. Apply the configuration so the service uses the current settings.

What gets installed

A standard installation includes:

- the rsyslog Windows Agent service
- the rsyslog Windows Agent Configuration Client
- supporting program files and libraries
- local help content shipped with the product

What to do next

After installation, continue with:

- Understand the Components
- Collect and Forward Windows Events
- Creating an Initial Configuration

If an in-place update reports success but the client, service, or selected features still show older files, see Why did my rsyslog Windows Agent update report success when the files were not upgraded correctly? before attempting another repair.

System Requirements

The actual requirements depend on whether you install only the background service or the full product including the Configuration Client.

Supported Windows platforms

Current rsyslog Windows Agent releases are intended for modern Windows versions, including Windows 10, Windows 11, Windows Server 2016, Windows Server 2019, Windows Server 2022, Windows Server 2025, and newer compatible releases.

Client

- The Configuration Client supports both 32-bit and 64-bit Windows environments.
- The Configuration Client requires Microsoft .NET Framework 4.7.2 or a newer .NET Framework 4.x release, such as .NET Framework 4.8 or 4.8.1.
- .NET Framework 4.x and .NET Core / .NET 5+ are different runtime families, so .NET Core and .NET 5+ do not satisfy this requirement.
- The installer can add the required .NET Framework components if they are not yet present.
- The client itself requires only modest memory and disk space beyond normal Windows requirements.

This requirement applies to systems where the Configuration Client is installed. An engine-only target does not require the Configuration Client.

Service

- The service has lower overhead than the full UI installation.
- Runtime requirements depend mostly on which services are enabled and how much data the system processes.
- Forwarding large event volumes, using reliable queues, or buffering temporary outages requires more memory and disk than a basic event-forwarding setup.
- If the agent only monitors local Windows Event Logs and forwards selected events, system impact is typically low.

Sizing guidance

Resource needs vary with:

- event volume
- number of enabled services
- filter complexity

- forwarding protocol and queue settings
- outage tolerance and disk queue sizing

If you expect sustained high event volume or long queue retention windows, allocate additional memory and disk space on the monitored system.

Understand the Components

rsyslog Windows Agent is easiest to understand if you separate its main components by role.

The two main components

1. **rsyslog Windows Agent Service** This is the runtime component. It runs in the background as a Windows service, runs the configured input services, evaluates rules, and forwards matching data.
2. **rsyslog Windows Agent Configuration Client** This is the administrative tool. You use it to define input services, rulesets, filters, and actions. Changes are made in the Configuration Client and then applied so the running service can use the new configuration.

How they fit together

- The **rsyslog Windows Agent service** performs the actual collection, filtering, and forwarding work through the configured input services.
- The **configuration client** defines what those input services, rulesets, and actions should do.
- The runtime service continues using the currently applied configuration until you save or apply changes from the client.

For the current split between remote administration and browser-based review, see **`../shared/faq/remote-administration-and-browser-based-review`**.

Optional downstream tools

rsyslog Windows Agent commonly forwards events to a central rsyslog server. In some environments, stored data may later be reviewed with tools such as Adiscon LogAnalyzer, but those are downstream systems, not core product components.

What to read next

- To understand the main event path, continue with Collect and Forward Windows Events.
- To build a first working setup, continue with Creating an Initial Configuration.
- To understand rule processing, see Process and Filter.

Collect and Forward Windows Events

rsyslog Windows Agent is primarily used to collect Windows-originated events through configured input services and forward them to rsyslog or another downstream receiver.

What the product collects

The most common sources are:

- Windows Event Log channels
- text-based application log files
- incoming syslog messages when the agent is used as a relay

What the product usually does next

After collection, rsyslog Windows Agent processes the events through a ruleset and then forwards the matching data to another system, most often a central rsyslog server.

Where to configure it

- Services is the entry point for event collection.
- Event Log Monitor V2 is the preferred service for modern Windows Event Log channels.
- File Monitor is available when you need to watch text-based log files.
- Syslog server is available when the agent should relay incoming syslog.
- If you run multiple input services, see `../shared/faq/listener-binding-rules` before reusing a transport, IP address, and port combination.

Quick verification

- Configure one collection input service.
- Bind it to a ruleset with a visible forwarding action.
- Apply the configuration and confirm that new events reach the receiver.

Creating an Initial Configuration

Use this page to build the first working rsyslog Windows Agent configuration: collect Windows Event Log events and forward them to an rsyslog receiver.

In this manual, **input** is the plain-language concept, while the configured object is a **service**.

Goal

At the end of this procedure, rsyslog Windows Agent will:

- monitor one or more Windows event logs
- process matching events through a ruleset
- forward them to a remote rsyslog server

Prerequisites

- rsyslog Windows Agent is installed.
- You can open the rsyslog Windows Agent Configuration Client.
- The rsyslog Windows Agent service is installed on the system.
- You know the destination host, port, and transport mode expected by the receiver.

Steps

1. Create a ruleset.
    - In the rsyslog Windows Agent Configuration Client, create a new ruleset.
    - Leave filtering simple for the first test so that visible events can match.
2. Add one forwarding action to that ruleset.
    - Inside the ruleset, add a Forward Syslog action.
    - Enter the receiver host and port.
    - Choose the transport mode expected by the receiver.
3. Create one event collection service.
    - Under Services, add an Event Log Monitor V2 service.
    - Bind that input service to the ruleset you created.
    - Select at least one Windows event log or channel to monitor.
4. Save and apply the configuration.
    - Apply or save the changes in the Configuration Client so the input service can use them.
    - Until you apply the changes, the running service continues to use the previous configuration.
5. Start or restart the rsyslog Windows Agent service if required.

How to verify

1. Trigger or wait for a Windows event that should be visible.
2. Confirm that the receiver gets the forwarded event.
3. If nothing arrives, check:
    - the rsyslog Windows Agent service is running
    - the event collection service is enabled
    - the input service is bound to the correct ruleset
    - the forwarding action is inside that ruleset
    - the receiver host, port, and transport mode are correct

Expected result

If the configuration is correct, rsyslog Windows Agent reads Windows Event Log data and forwards matching events to the configured receiver.

Next step

- To refine matching behavior, continue with Process and Filter.
- To learn about forwarding options, continue with Store and Forward.

Process and Filter

rsyslog Windows Agent uses a rules engine to decide what to do with each collected event: forward it, enrich it, route it differently, or drop it.

Where to configure

- Configure rsyslog Windows Agent explains the tree view and how services, rulesets, rules, filters, and actions fit together.
- Filter Conditions decide which events match a rule.
- Actions define what happens for matching events.

Recommended setup path

1. Start with one collection service bound to one ruleset.
2. Add one simple forwarding action so results are easy to verify.
3. Add filter conditions to narrow down the events you care about.
    - Start with event source, event ID, severity, or log name.
    - Add message-content filters only after the basic event path works.
4. Add further actions only after the rule matches exactly what you intend.

Things that commonly trip people up

- Rule order matters: rules are evaluated top-to-bottom inside a ruleset.
- The service-to-ruleset binding decides which ruleset sees a collected event.
- Defaults are templates. They do not process anything until you create an actual service or action instance.

Next steps

- Learn the underlying model in Core concepts.
- For the detailed tree structure, see Multiple RuleSets - Rules - Actions.

Store and Forward

Once rsyslog Windows Agent collects an event and it matches a rule, actions can forward it to downstream systems or alter the processing flow internally.

Where to configure

- Actions is the entry point for forwarding and internal processing behavior.

Common forwarding targets

- Syslog: Forward Syslog
- RELP: Send RELP
- TLS-protected syslog: use the TLS options inside Forward Syslog
- DTLS: Send DTLS

Common internal actions

- Call RuleSet
- Normalize Event
- Resolve Hostname
- Set Property
- Set Status
- Discard

Related analysis component

- For reviewing stored logs later, Adiscon LogAnalyzer works with file- and database-based storage.
- Users who prefer a simpler standalone tool for direct log file review may also consider the third-party log viewer Retrospective by centeractive.

Recommended setup path

1. Start with one forwarding target and verify end-to-end delivery.
2. Prefer reliable transports when message loss is not acceptable.
3. Add internal actions only after the forwarding path behaves as expected.
4. If you need outage tolerance, configure disk-backed queues where the action supports it.

If stored data is reviewed later in a browser, treat Adiscon LogAnalyzer as a separate downstream component. For the current split between service administration and browser-based review, see `../shared/faq/remote-administration-and-browser-based-review`.

Operate and Troubleshoot

After initial setup, most operational work is validating event intake, tuning rules, and diagnosing why forwarding succeeded, failed, or behaved differently than expected.

Quick checklist

1. Confirm the rsyslog Windows Agent service is running.
2. Confirm the collection service is enabled and bound to the intended ruleset.
3. Confirm your rule order and filters match what you expect.
4. Check the forwarding action host, port, and transport settings.

Useful diagnostics

- Export the current configuration and collect a debug log when investigating problems. See Tutorial: Export the Configuration and Create a Debug Log.
- If forwarding to rsyslog fails, verify connectivity, protocol mode, TLS or certificate settings, and receiver-side listener configuration.

Where to look next

- Input issues: Services
- Matching issues: Filter Conditions
- Output issues: Actions
- Common operational questions: FAQ

Tutorials

Use this section for concrete rsyslog Windows Agent tasks. Each tutorial is self-contained and focuses on one outcome.

Tutorial: Forward Windows Events to rsyslog

Use this tutorial when rsyslog Windows Agent should relay selected Windows events to a remote rsyslog server.

Goal

At the end of this procedure, rsyslog Windows Agent will forward matching Windows event log entries to a remote rsyslog receiver.

Prerequisites

- The destination host name or IP address
- The target port and transport mode expected by the receiver
- A ruleset that receives events from an Event Log Monitor service

Steps

1. Create or choose the ruleset whose events should be forwarded.
2. Add a Forward Syslog action to that ruleset.
3. Configure the remote target.
    - Enter the destination host name or IP address.
    - Set the destination port.
    - Choose the transport mode that matches the receiver.
4. Keep protocol choices conservative.
    - Use UDP only if message loss is acceptable.
    - Prefer TCP-based modes when reliable delivery matters.
5. Save and apply the configuration.
6. Restart the rsyslog Windows Agent service if required.

Verification

1. Trigger an event that matches the ruleset.
2. Confirm that the remote rsyslog server receives it.
3. If forwarding fails, verify host, port, protocol mode, and firewall rules on both sides.

Next step

If forwarding works, continue with:

- Tutorial: Forward Windows Events via TLS to rsyslog
- Syslog Forwarding

Tutorial: Forward Windows Events via TLS to rsyslog

Use this tutorial when rsyslog Windows Agent should forward selected Windows Event Log records to an rsyslog receiver over encrypted syslog transport.

Goal

At the end of this procedure, rsyslog Windows Agent will forward matching Windows events to an rsyslog server over TCP with TLS enabled.

Prerequisites

- The rsyslog receiver host name or IP address
- The TCP port used by the rsyslog TLS listener
- The framing mode expected by the receiver
- The CA certificate or client certificate files required by the receiver
- A ruleset that receives events from an Event Log Monitor service

Configure the rsyslog receiver

Configure the rsyslog server first so that it accepts TLS-protected syslog over TCP. The examples below use RainerScript syntax and follow the local rsyslog documentation under *../rsyslog2/doc/source/*.

Minimal TLS listener with anonymous authentication:

```
global(
  defaultNetstreamDriver="gtls"
  defaultNetstreamDriverCAFile="/etc/rsyslog.d/certs/ca.pem"
  defaultNetstreamDriverCertFile="/etc/rsyslog.d/certs/server-cert.pem"
  defaultNetstreamDriverKeyFile="/etc/rsyslog.d/certs/server-key.pem"
)

module(
  load="imtcp"
  streamDriver.name="gtls"
  streamDriver.mode="1"
  streamDriver.authMode="anon"
)

input(
  type="imtcp"
  port="6514"
)
```

Stricter listener with certificate validation and client name matching:

```
global(
  defaultNetstreamDriver="gtls"
  defaultNetstreamDriverCAFile="/etc/rsyslog.d/certs/ca.pem"
  defaultNetstreamDriverCertFile="/etc/rsyslog.d/certs/server-cert.pem"
  defaultNetstreamDriverKeyFile="/etc/rsyslog.d/certs/server-key.pem"
)

module(
  load="imtcp"
  streamDriver.name="gtls"
  streamDriver.mode="1"
  streamDriver.authMode="x509/name"
)
```

```
input(
  type="imtcp"
  port="6514"
  permittedPeer=["windows-agent01.example.net"]
)
```

Use *x509/name* when the receiver should validate the client certificate and restrict accepted senders to the permitted certificate names. Use *anon* only when that weaker trust model is acceptable in your environment. In both cases, restart rsyslog after changing the listener configuration.

Steps

1. Create or choose the ruleset whose events should be forwarded.
2. Add a Forward Syslog action to that ruleset.
3. Configure the target host and port.
   - Enter the rsyslog server host name or IP address.
   - Set the TCP port used by the rsyslog TLS listener.
4. Select a TCP-based syslog transport mode.
   - Prefer **TCP (octet-count based framing)** when the receiver supports it.
   - Otherwise use the TCP framing mode that matches the rsyslog listener.
5. Open the TLS settings for the action and enable **SSL / TLS Encryption**.
6. Select the TLS mode that matches the receiver configuration.
   - Use the default anonymous mode only if the receiver is configured for it.
   - Use certificate-based mode when the receiver expects CA validation or a client certificate.
7. If the receiver requires certificate-based trust, provide the matching files.
   - Select the common CA PEM file used to validate the receiver.
   - If mutual authentication is required, also select the client certificate PEM and key PEM files.
8. Keep protocol settings modern.
   - Leave **SSL v3** disabled.
   - Leave **TLS v1.0** disabled.
   - Use **TLS v1.2** or **TLS v1.3** when the receiver supports them.
9. Save and apply the configuration.
10. Restart the rsyslog Windows Agent service if required in your environment.

Verification

1. Trigger an event that matches the ruleset.
2. Confirm that the rsyslog receiver accepts the TLS connection and receives the forwarded event.
3. If forwarding fails, check:
   - target host and port
   - rsyslog *imtcp* listener configuration
   - TCP framing mode
   - CA, certificate, and key files
   - TLS version compatibility
   - *permittedPeer* entries on the rsyslog side when *x509/name* is used
   - firewall rules between rsyslog Windows Agent and rsyslog

Next step

If forwarding works, continue with:

- Syslog Forwarding
- Store and Forward

Tutorial: Forward Windows Events via RELP to rsyslog

Use this tutorial when rsyslog Windows Agent should forward selected Windows Event Log records to an rsyslog receiver with RELP.

Goal

At the end of this procedure, rsyslog Windows Agent will forward matching Windows events to an rsyslog server through RELP.

Prerequisites

- The rsyslog receiver host name or IP address
- The RELP port used by the receiver
- Any TLS or certificate files required by the receiver
- A ruleset that receives events from an Event Log Monitor service

This tutorial assumes the rsyslog side is already configured to accept RELP.

Steps

1. Create or choose the ruleset whose events should be forwarded.
2. Add a Send RELP action to that ruleset.
3. Configure the target host and RELP port.
4. Keep the default RELP port unless the receiver uses a different one.
5. If the receiver expects TLS, enable SSL / TLS encryption and provide the matching CA or client certificate files.
6. If reliable delivery across temporary outages matters, enable disk-backed queues for the action.
7. Save and apply the configuration.
8. Restart the rsyslog Windows Agent service if required.

Verification

1. Trigger an event that matches the ruleset.
2. Confirm that the rsyslog receiver accepts the RELP connection and receives the event.
3. If forwarding fails, check host, port, TLS settings, and receiver-side RELP listener configuration.

Next step

If RELP forwarding works, continue with:

- Send RELP
- Store and Forward

Tutorial: Export the Configuration and Create a Debug Log

Use this tutorial when you need rsyslog Windows Agent configuration data and a debug log for troubleshooting or for a support case.

Goal

At the end of this procedure, you will have:

- a text-based export of the rsyslog Windows Agent configuration
- a debug log file that captures service behavior

Prerequisites

- Access to the rsyslog Windows Agent Configuration Client
- Permission to restart the rsyslog Windows Agent service if needed
- A writable location for the exported settings and debug log

Steps

1. Export the configuration.
    - Open the rsyslog Windows Agent Configuration Client.
    - Go to **Computer -> Export Settings to Registry File**.
    - Export the settings as a text-based registry file.
2. Save the export to a known location.
    - Use a descriptive file name such as `rsyslogwa-config.reg`.
    - Do not use the binary export format.
3. Enable debug logging.
    - Open Debug under **General**.
    - Enable debug output into file.
    - Set a full path and file name for the debug log.
    - Start with **Errors & Warnings** and **Minimum Debug Output** unless support asks for a higher level.
4. Restart the rsyslog Windows Agent service if required.
5. Reproduce the problem or run the test scenario.
6. Disable debug logging again after capture.
    - Debug logging increases system load and should not stay enabled during normal operation.
7. Package the files for review or support.
    - Compress the exported registry file and the debug log file into one ZIP archive.

Verification

1. Confirm that the registry export file exists and is readable as text.
2. Confirm that the debug log file exists and contains recent entries from the test period.
3. If the debug log stays empty, verify that the service was restarted after enabling debug output.

Next step

If you need to send the data to Adiscon, continue with:

- How to Export rsyslog Windows Agent Settings for a Support Call
- How do I contact Adiscon sales?

Configuration

Use this section to configure how rsyslog Windows Agent collects Windows events, processes them through rules, and forwards the results.

If you are new to the product, start with `Getting Started` for the first working setup and return here for the detailed configuration pages.

In this manual, **input** is the clearest plain-language concept for anything that collects or receives events, while **service** remains the operational term for the configured rsyslog Windows Agent object.

Recommended setup path

1. Define input services under Services and bind each service to a ruleset.

2. Build processing under Filter Conditions.
3. Add forwarding or internal processing under Actions.
4. Verify end-to-end delivery with a simple forwarding action before refining the filters.

Core concepts

Use this section to understand how rsyslog Windows Agent thinks about collected Windows events and how configuration objects interact.

Concept map

rsyslog Windows Agent processing follows this model:

1. An **input service** collects or receives an event.
2. The collected data becomes an **information unit** inside the product.
3. The **rule engine** evaluates the event against **rules** and **filter conditions**.
4. Matching **actions** forward, transform, or discard the event.

In plain language, you can read this as:

```
input service -> ruleset -> action
```

Canonical concept pages

rsyslog Windows Agent Services

Services are the input objects that collect or receive events and pass them to rulesets.

Key idea

Each active service instance is bound to a ruleset. That binding decides which processing logic sees the collected event.

Common service types

- **Heartbeat** generates status events that show the agent is still alive.
- **MonitorWare Echo Reply** responds to echo requests in monitored environments.
- **Syslog Server** receives syslog messages on the configured port.
- **Event Log Monitor** reads Windows Event Log channels and creates internal events from them.
- **File Monitor** watches text-based logs and turns new lines into events.

Important distinction

Service defaults are templates. They are not active services and do not collect anything until you create a concrete service instance.

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).

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.

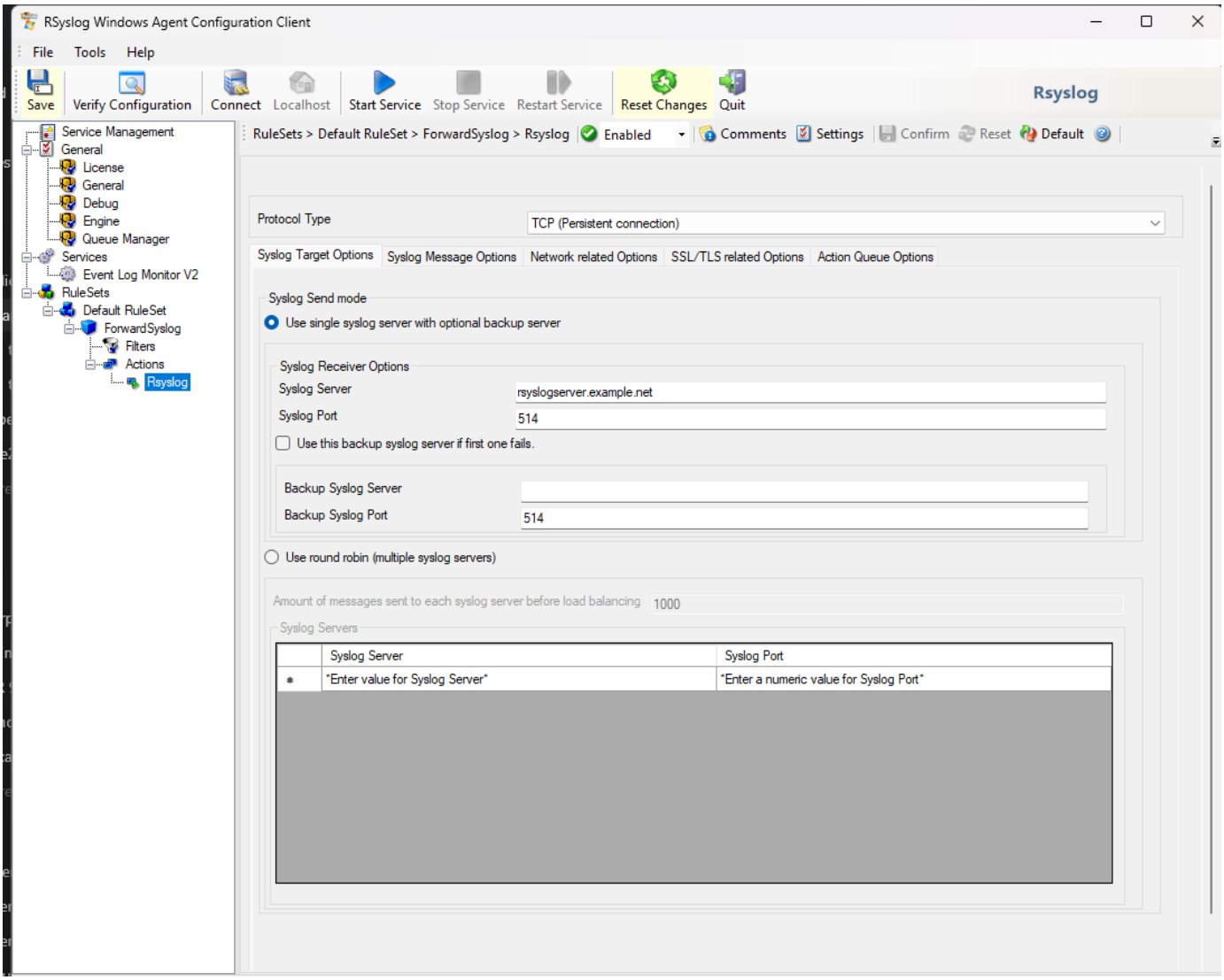
The rule base consists of the rules as configured in the client. The rule engine is the process carrying out the rules. A rule base can contain no, one or an unlimited number or rules. However, if there is no rule at all defined, no action will ever be carried out by the agent. Consequently, the client 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 the 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:



rsyslog Windows Agent sample rule base

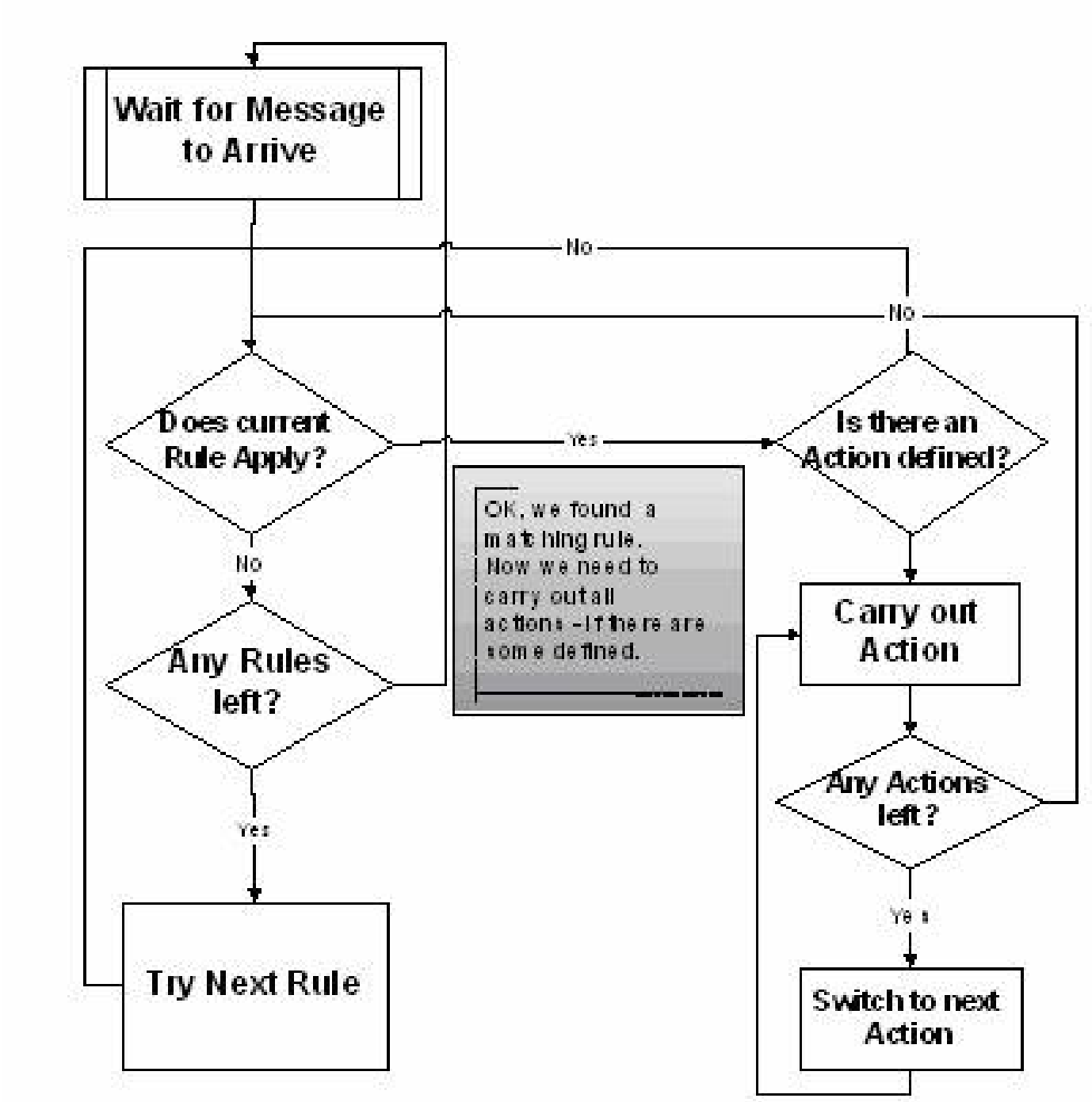
Now that we know the elements, how are rules being processed. It is easy. Rules are strictly processed from top to bottom, or from number one to the last one. For each rule the filter conditions are checked to see if they match. If they do, all associated actions are carried out. Then, the rule engine 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:

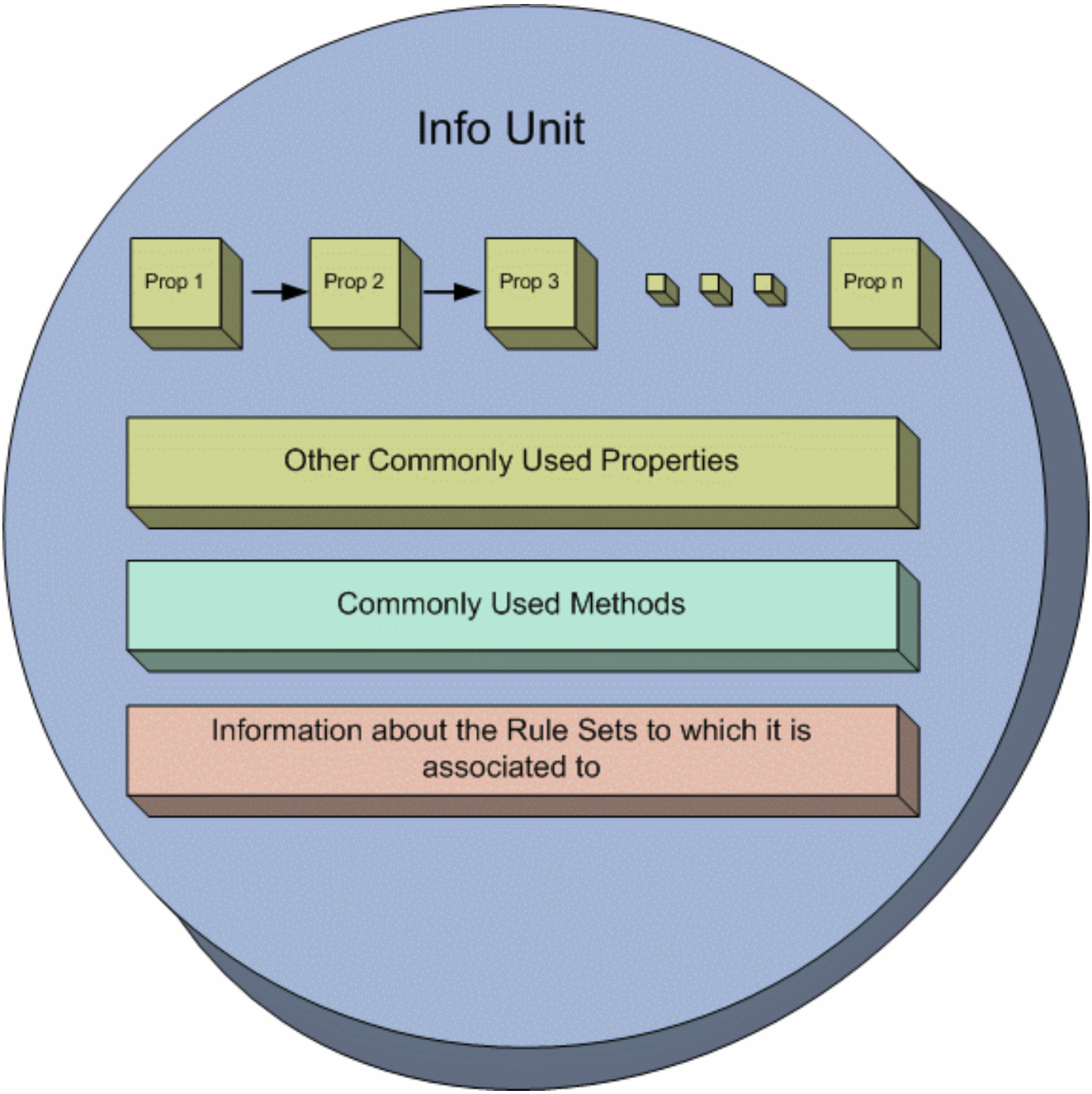


Figure 1: Conceptual Diagram of an Info Unit

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

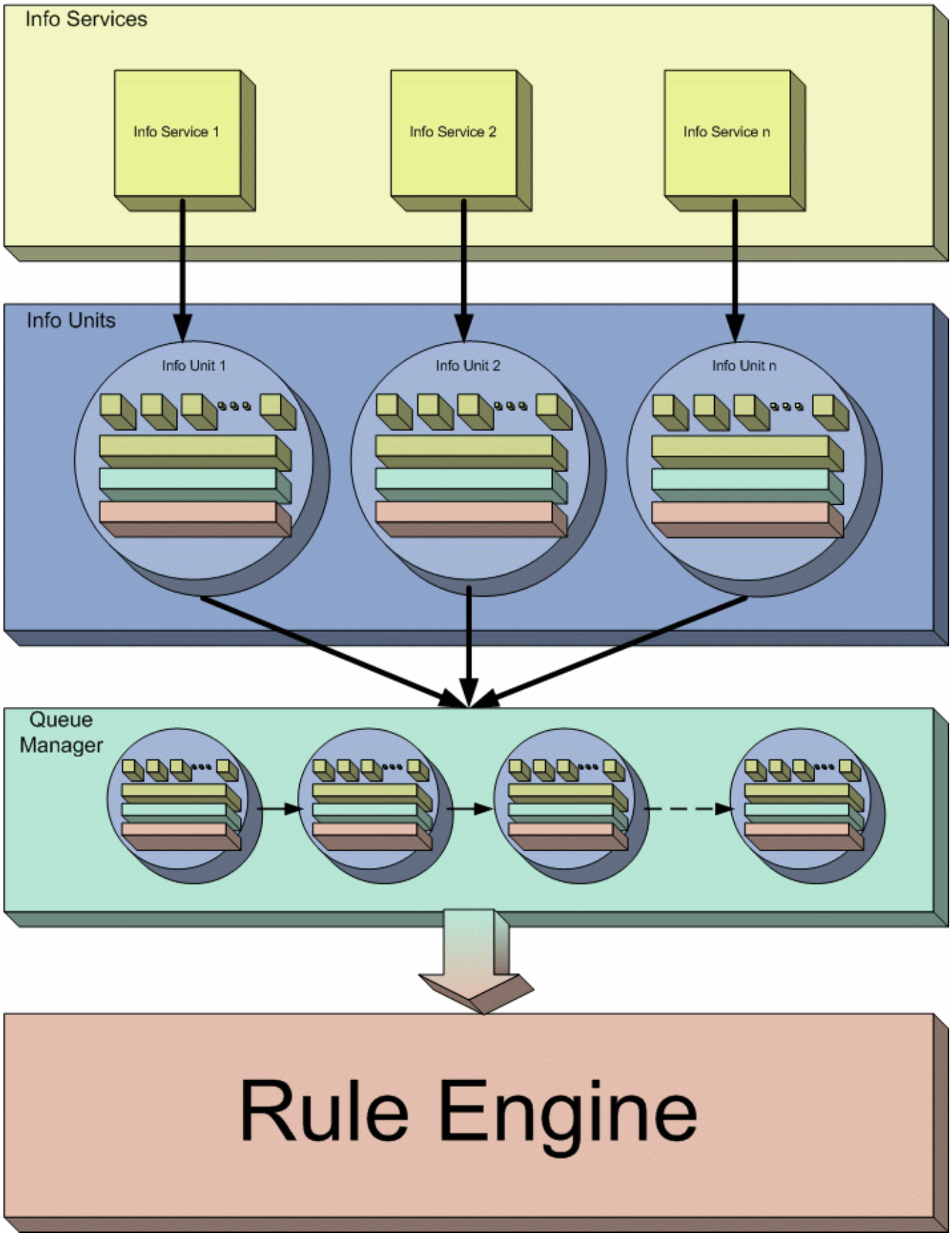


Figure 2: Overall Process

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:

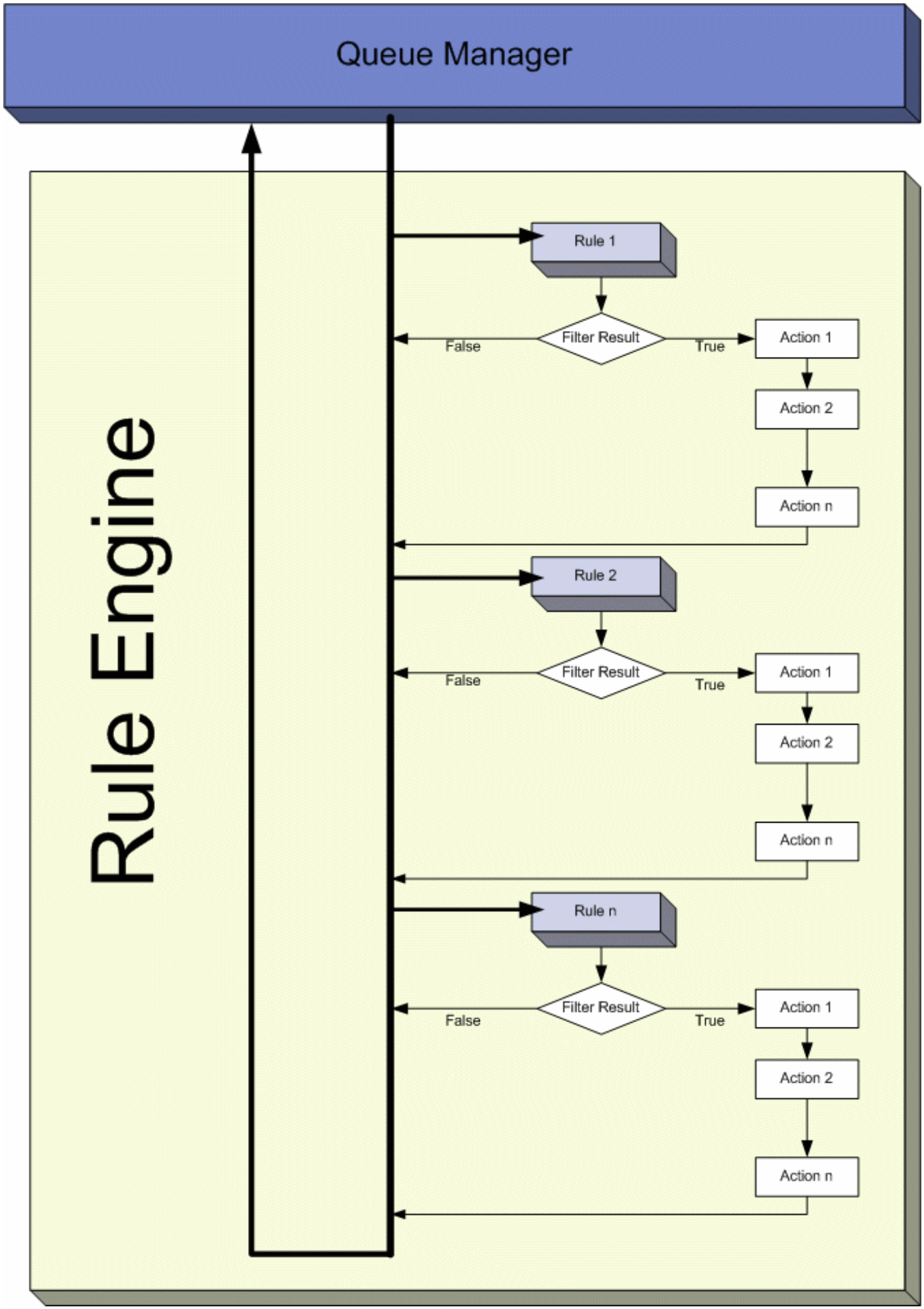

Queue Manager
Rule Engine
Rule 1
Filter Result
False
True
Action 1
Action 2
Action n
Rule 2
Filter Result
False
True
Action 1
Action 2
Action n
Rule n
Filter Result
False
True
Action 1
Action 2
Action n

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.

rsyslog Windows Agent Filter Conditions

Filter conditions decide whether a rule matches an event.

What they are for

Use filter conditions to separate the events you care about from the ones you do not want to forward, enrich, or route further.

Typical filter categories

- **Global Conditions** such as minimum wait time or event occurrence count
- **General Conditions** such as source system or message content
- **Date / Time** constraints for business-hour or schedule-based logic
- **Information Unit Type** to separate event log, syslog, or file-based data
- **Syslog** properties such as facility or priority
- **Event Log Monitor** properties such as event ID, event source, severity, category, or user
- **Custom Property** conditions for enriched or generated values

Recommended approach

Start simple. First verify that the event reaches the intended ruleset. Then add the smallest set of filters needed to narrow the match.

rsyslog Windows Agent Actions

Actions define what happens after an event matches a rule.

Forwarding actions

The main forwarding actions in rsyslog Windows Agent are:

- **Forward Syslog** for standard syslog delivery over UDP or TCP
- **Send RELP** for reliable event transport to a RELP-enabled receiver
- **Send DTLS** for packet-based transport with encryption where that is required

Internal actions

Internal actions modify or redirect processing inside the agent itself:

- **Call RuleSet** transfers processing to another ruleset
- **Normalize Event** rewrites selected event content
- **Resolve Hostname** resolves a name or address from one property and stores the result in another property
- **Set Property** and **Set Status** add or modify internal values
- **Discard** stops further processing for the current event

Why order matters

Actions run in the order you place them inside a rule. That order affects both event content and delivery behavior.

Why this matters

Understanding these concepts helps you:

- design rulesets with predictable behavior
- avoid duplicate or conflicting processing paths
- choose the right transport and action type for each receiver
- troubleshoot why an event did or did not match a rule

Configure rsyslog Windows Agent

Use this page to understand how the rsyslog Windows Agent Configuration Client is organized and how changes reach the running service.

How configuration works

The rsyslog Windows Agent service runs in the background once it is installed and configured. The Configuration Client is the administrative tool used to define services, rulesets, filters, and actions.

Changes are made in the Configuration Client and then applied so the running service can use the new configuration. Until you apply the changes, the service continues using the previous active configuration.

How the tree is organized

The tree view has three main areas:

- **General / Defaults** for product-wide settings and default templates
- **Running Services** for active inputs such as Event Log Monitor, File Monitor, or Syslog Server
- **RuleSets** for the processing logic that evaluates filters and actions

Defaults vs. active instances

Default objects are templates. They provide pre-filled values for new services or actions, but they do not collect, process, or forward anything by themselves. Actual work only happens in the active service and action instances you create under **Running Services** and the selected rulesets.

How data moves through the product

1. A configured **service** collects an event.
2. That service sends the event into its assigned **ruleset**.
3. Each **rule** evaluates its filter conditions in order.
4. Matching **actions** forward, enrich, route, or discard the event.

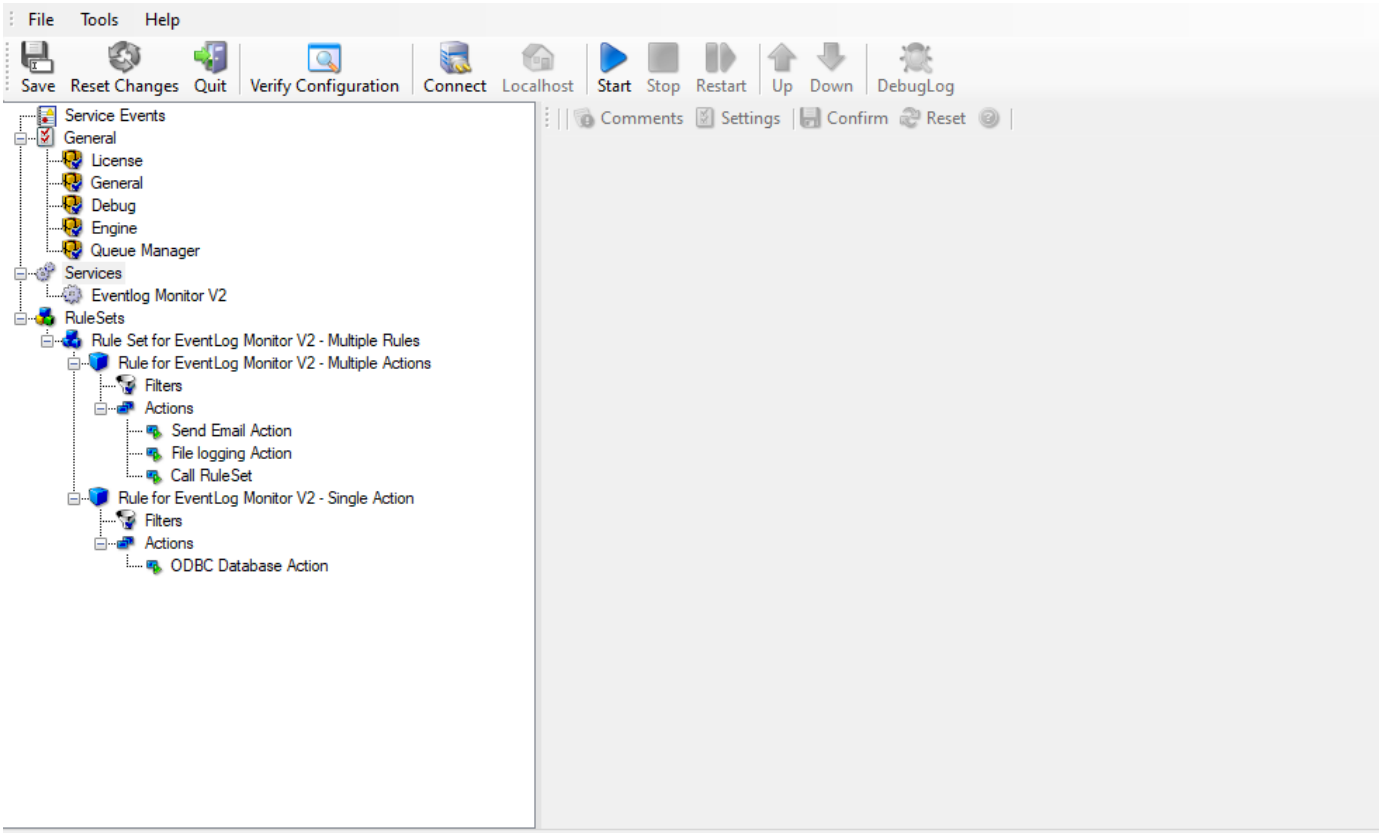
Common editing tasks

- To add an input, right-click **Running Services** and create the service you need.
- To add processing logic, create or edit a ruleset under **RuleSets**.
- To change output behavior, edit the actions inside the relevant rule.
- To enable new settings for the service, save or apply the configuration after editing.

What to read next

- User Interface changes
- Core concepts
- Services
- Filter Conditions
- Actions

Multiple RuleSets - Rules - Actions



- Multiple RuleSets, Rules and Actions*

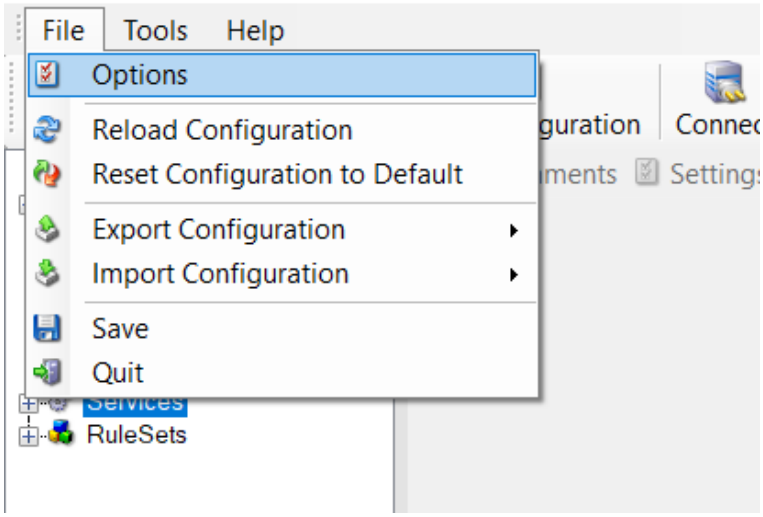
With rsyslog Windows Agent as many “RuleSets”, “Rules” and “Actions” as necessary can be defined.

You can create a separate “RuleSet” for each Service used, or just one “RuleSet” for all services. RuleSets can also be created to use them with the “callruleset action”.

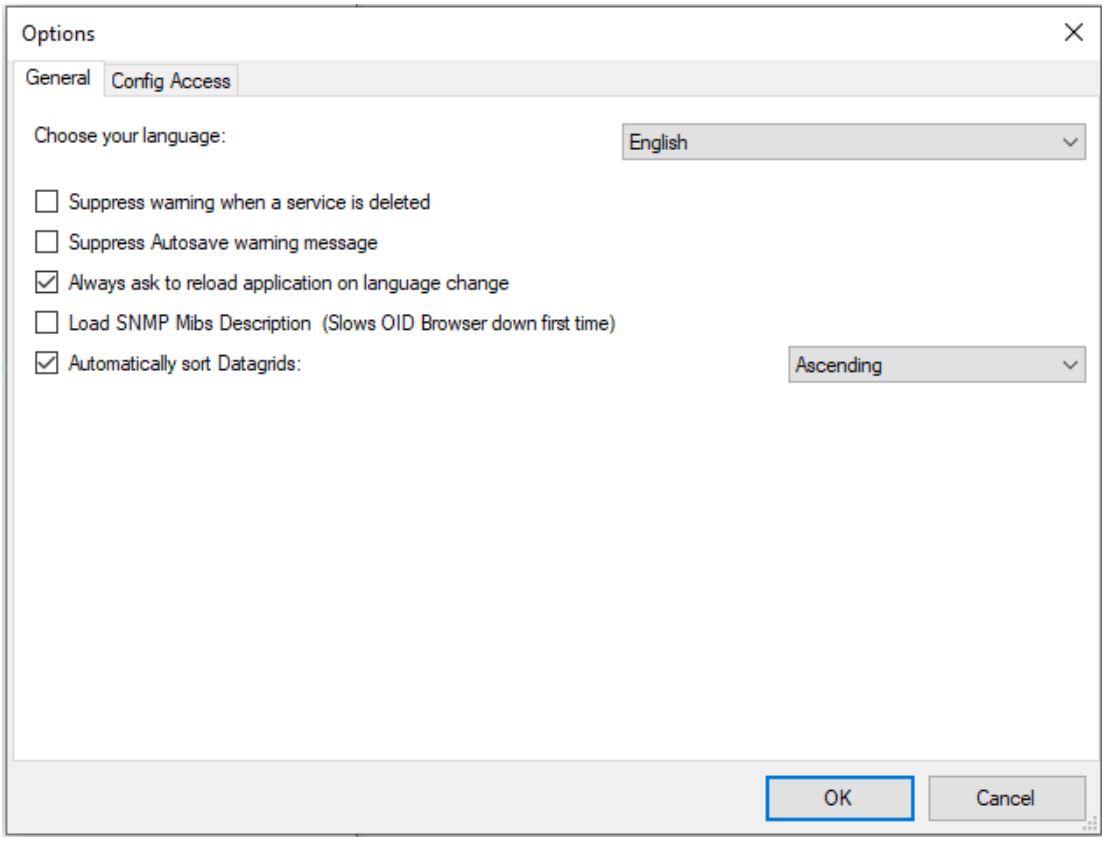
RuleSets contain one or multiple rules. All actions and processing carried out is defined by the rules. A rule consists of filter conditions and 1 to multiple actions. All actions that have to meet the same filter conditions can be combined in the same rule.

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

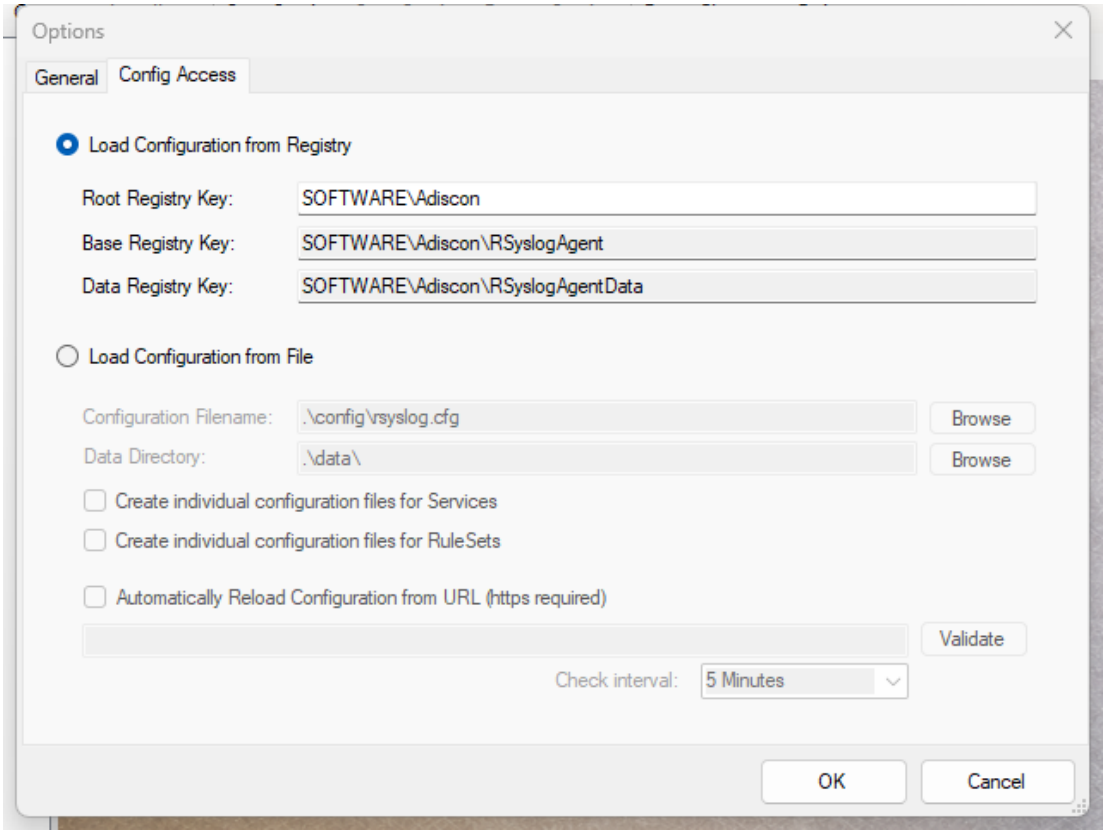
When you change the language, a popup will ask you to reload the configuration client to properly apply the changes and load with the set language.

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.



rsyslog Windows Agent 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 the saved within the Parameters key of the Service.

Load Configuration from File

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

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

Create individual configuration files for Services

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

Create individual configuration files for RuleSets

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

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 then 5 minutes.

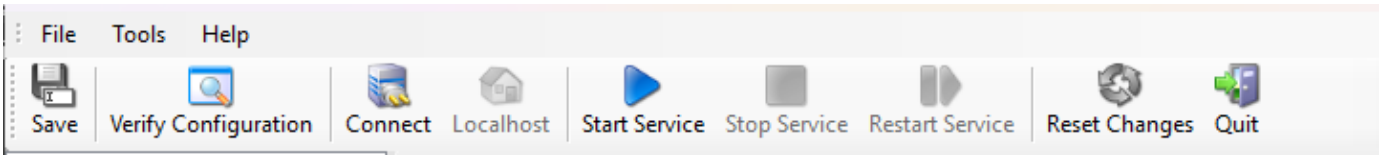
Client Tools

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.

Service control

From the **main window** toolbar: **Start**, **Stop**, **Restart**, **Kill**, and **Open Windows Services**. **Kill** ends an orphan process when the service manager state is inconsistent. When the service is paused during configuration reload, stop and restart may be unavailable until reload completes.

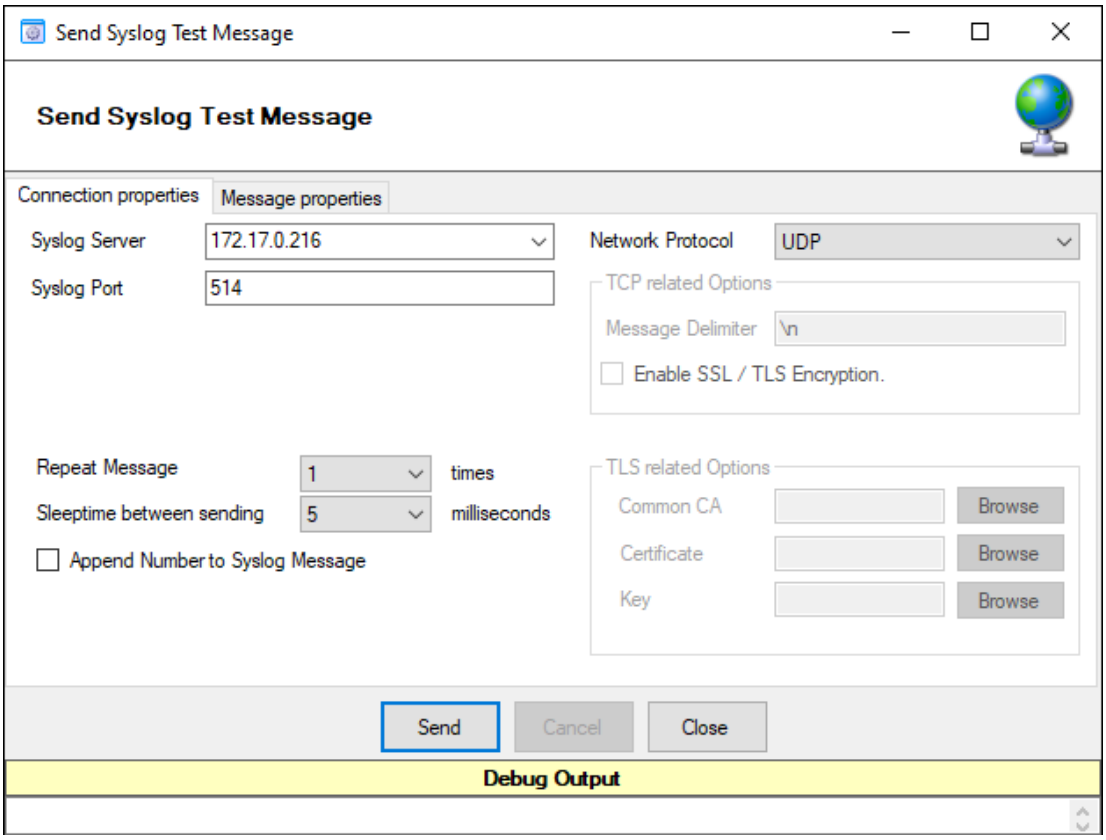
The property window toolbar (shown below) provides **Start**, **Stop**, **Restart**, and **Reset Changes** while editing a service or action. **Kill** and **Open Windows Services** are on the main window toolbar only.



See User Interface changes for **Reset to default** (selected item vs entire configuration), YAML import/export, and the MonitorWare Agent User Interface.

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

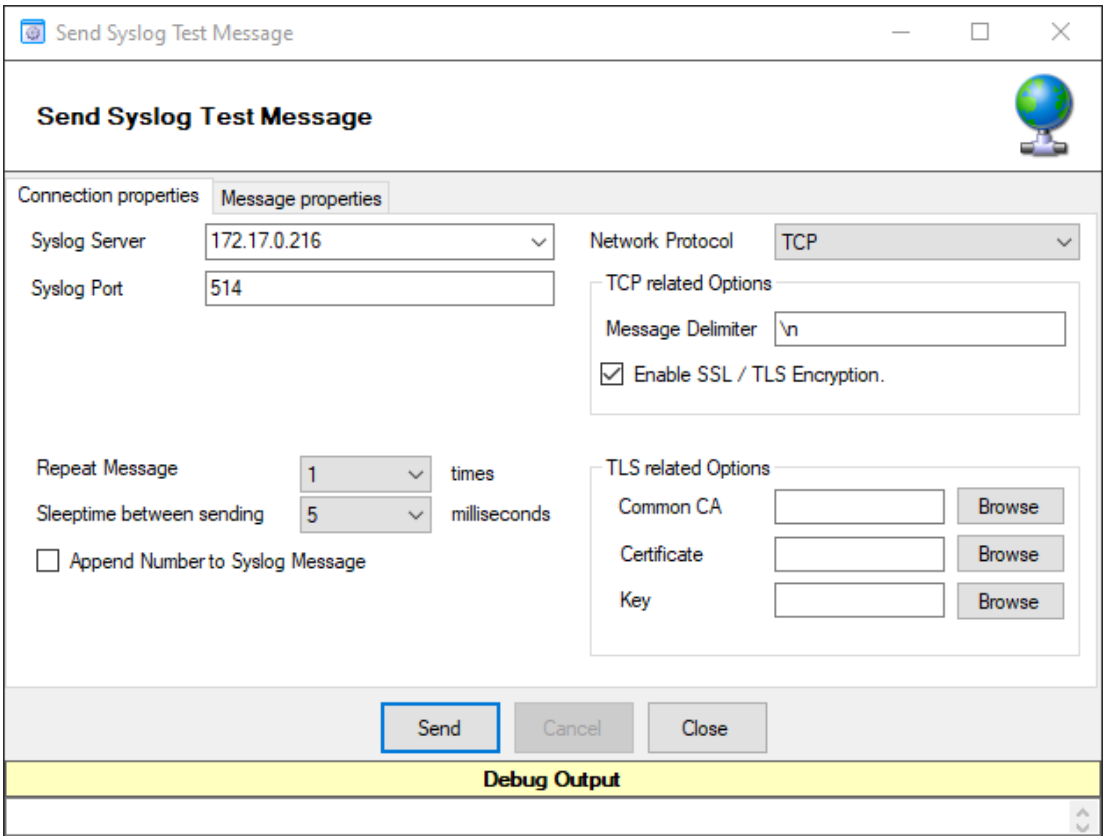
When using TCP, you can use 0ms. For UDP we recommend 1-5ms as sleeptime between sending syslog messages. Otherwise package loss can happen.

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.



- 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)

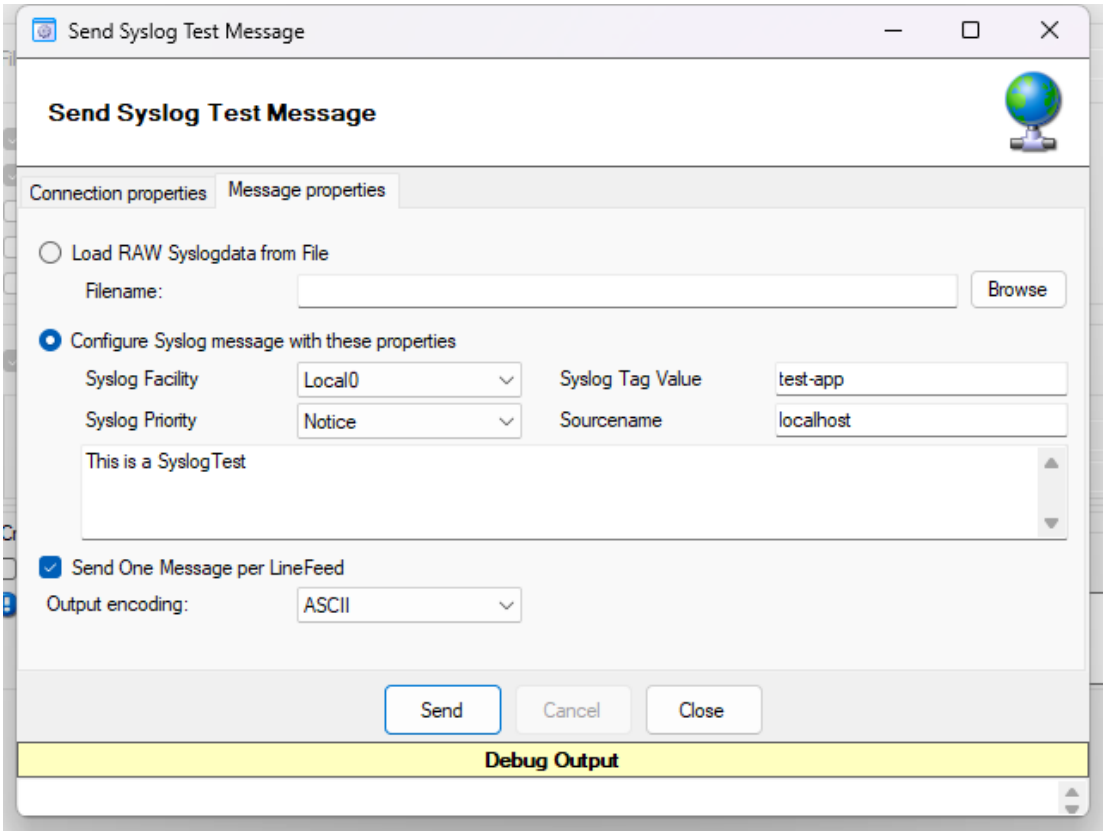
Check this option to enable the TLS related Options.

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).



rsyslog Windows Agent Syslog Test 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

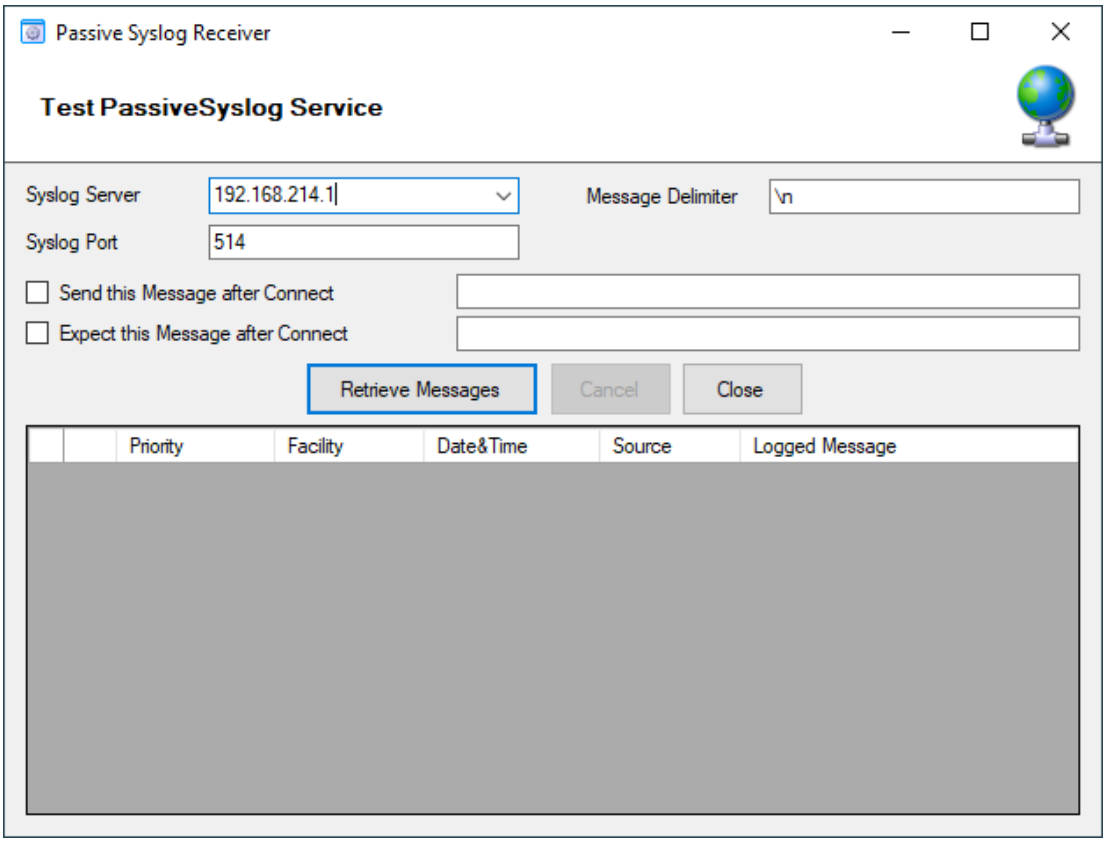
Check if your syslogdata contains multiple syslog messages divided by line feeds

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.



- 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

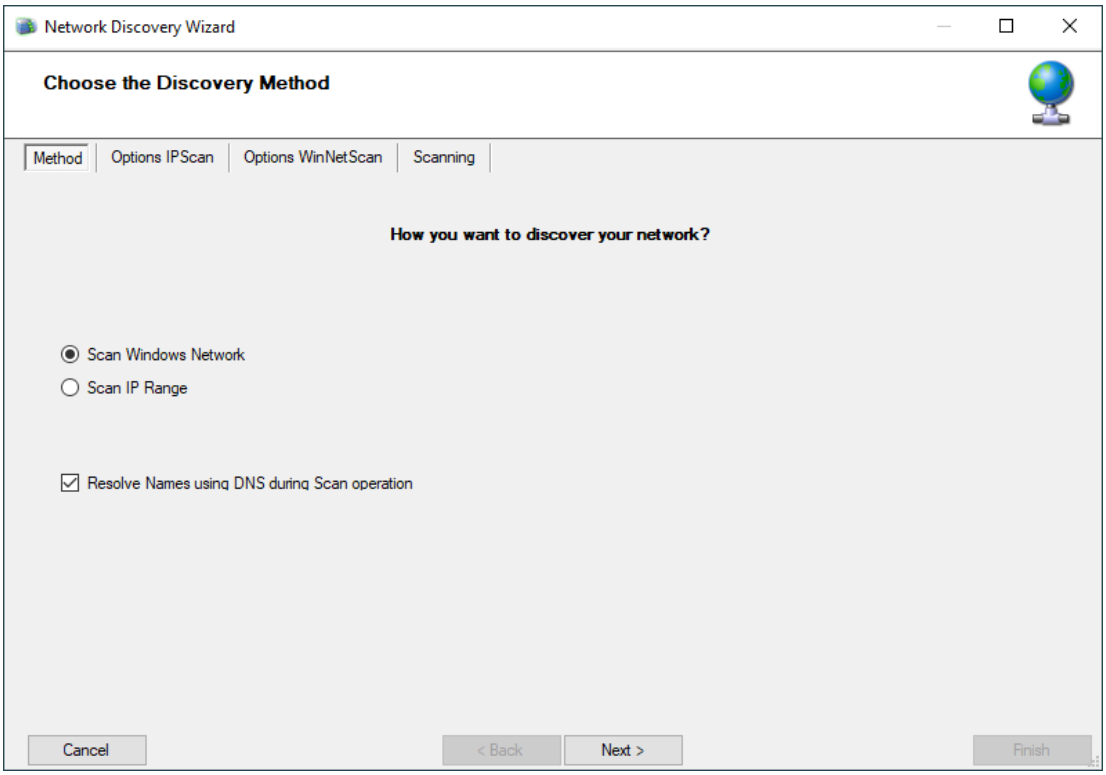
If required, configure a custom message that is send to the server after connect.

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.



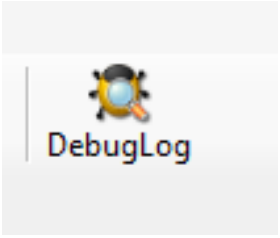
- Network Discovery - Choose the discovery Method*

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 quiet simple. In the following description, all the configuration options will be explained in detail.

Adiscon Configuration format explained

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

Variables

All variables start with a dollar ($). Name and Value of a variable are separated by the FIRST space character. Everything else behind the first space will be considered as the Value. A 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
               $PropertyValueType NOTNEEDED
               $CompareOperation EQUAL
               $nOptionalValue 0
               $nSaveIntoProperty 0
               $szSaveIntoPropertyName FilterMatch
       }
    }
}
```

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

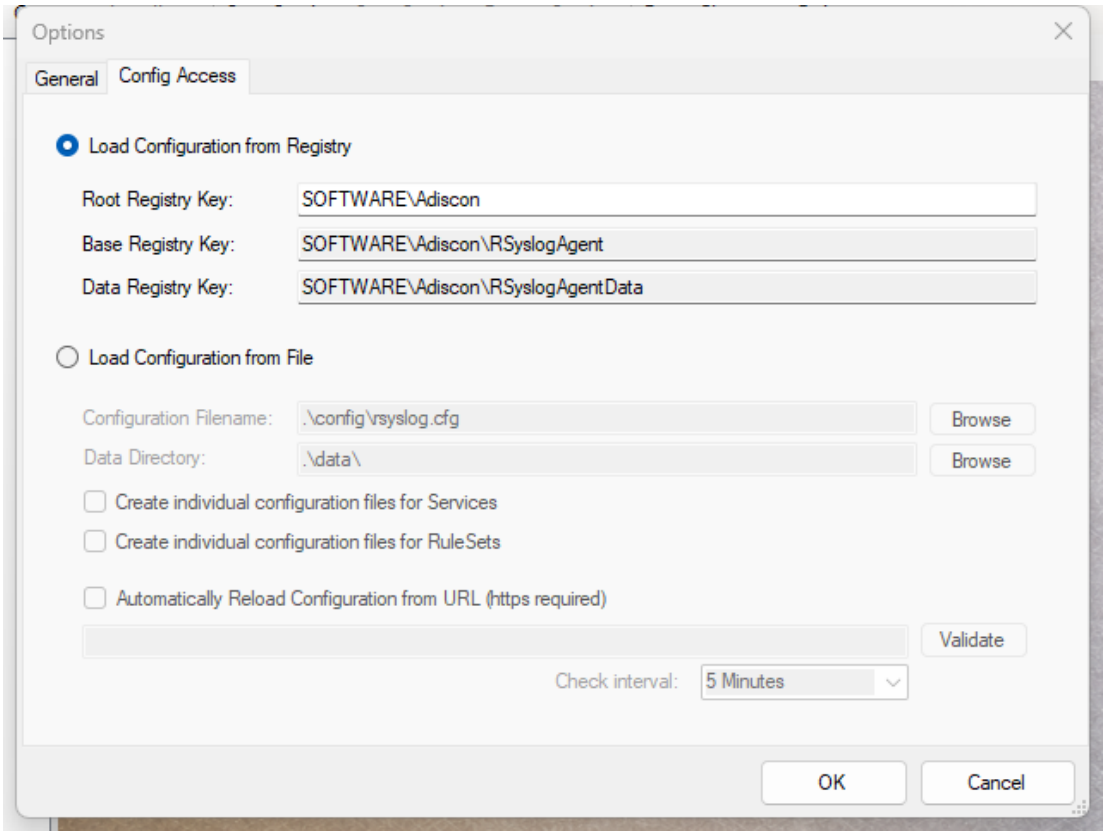
Within a Rule you can have Actions. The brackets start and end an action block. All variables in an action block between the brackets will be loaded into the action configuration. The name attribute specifies the service display name. The type attribute contains the action type ID. It can be one of the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

```
1000 = ODBC Database
1001 = Send Syslog
1008 = Net Send (legacy; removed from 26.07 configuration client)
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
1032 = HTTP REST Output
1033 = Message Ringbuffer
```

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.



rsyslog Windows Agent file-based configuration options

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

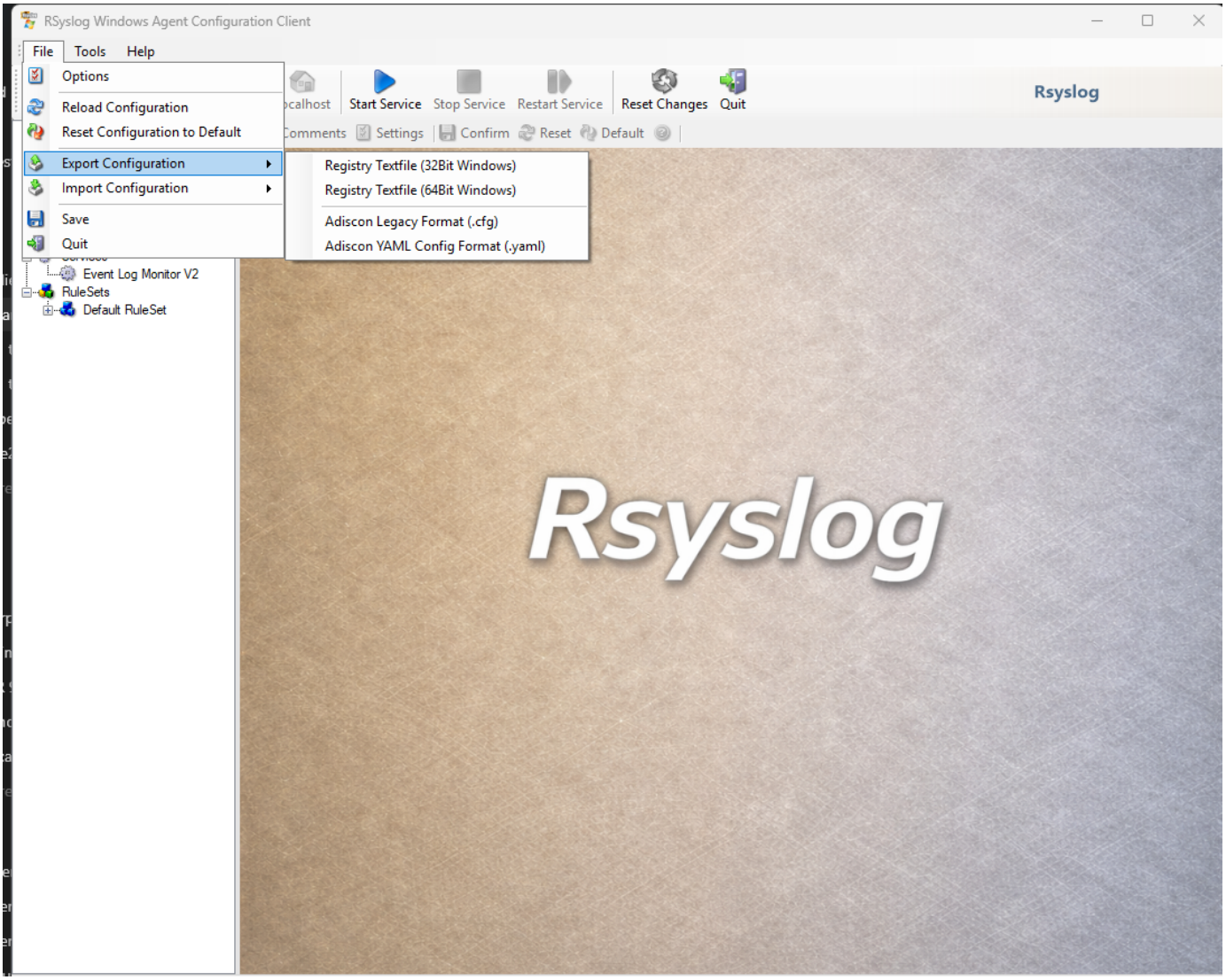
When enabled, the configuration client will create separated configuration files for each configured ruleset. The main configuration file will then use the includeconfig statement to include all these configuration files by using a pattern. When deleting a ruleset, its configuration file will be deleted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

Adiscon YAML configuration format

The configuration client can import and export **Adiscon YAML Config Format** (`.yaml`) files in addition to the legacy `.cfg` format (**Adiscon Legacy Format** in the File menu).

New file-based deployments should use YAML configuration. The legacy `.cfg` Adiscon Config Format remains supported for compatibility with existing installations, test fixtures, support workflows, and external tooling, but it is not required for new Adiscon product deployments.

Import / export: Use **File** → import or export and select the YAML format. Select legacy CFG only when your workflow requires that format.



rsyslog Windows Agent File menu with YAML import and export commands

Unsupported blocks: If you open a configuration that contains blocks this client version does not understand (for example newer service features), the client skips them and may show a warning. Saving from an older client can drop those sections. See What happens when the client opens a configuration with unsupported blocks?.

License file path in general options:

```
general(name="License") {
 $szLicenseV2Path C:\\ProgramData\\Adiscon\\MonitorWare\\license.alic
}
```

Send RELP window size example:

```
action(type="1027" name="Forward RELP") {
 $nSelpWindowSize 128
 ...
}
```

HTTP REST Output example:

```
action(type="1032" name="REST Out") {
 $szProfile
 $szServer api.example.com
 $szPath /v1/events
 $bUseTLS true
 $szMethod POST
 ...
}
```

General Options

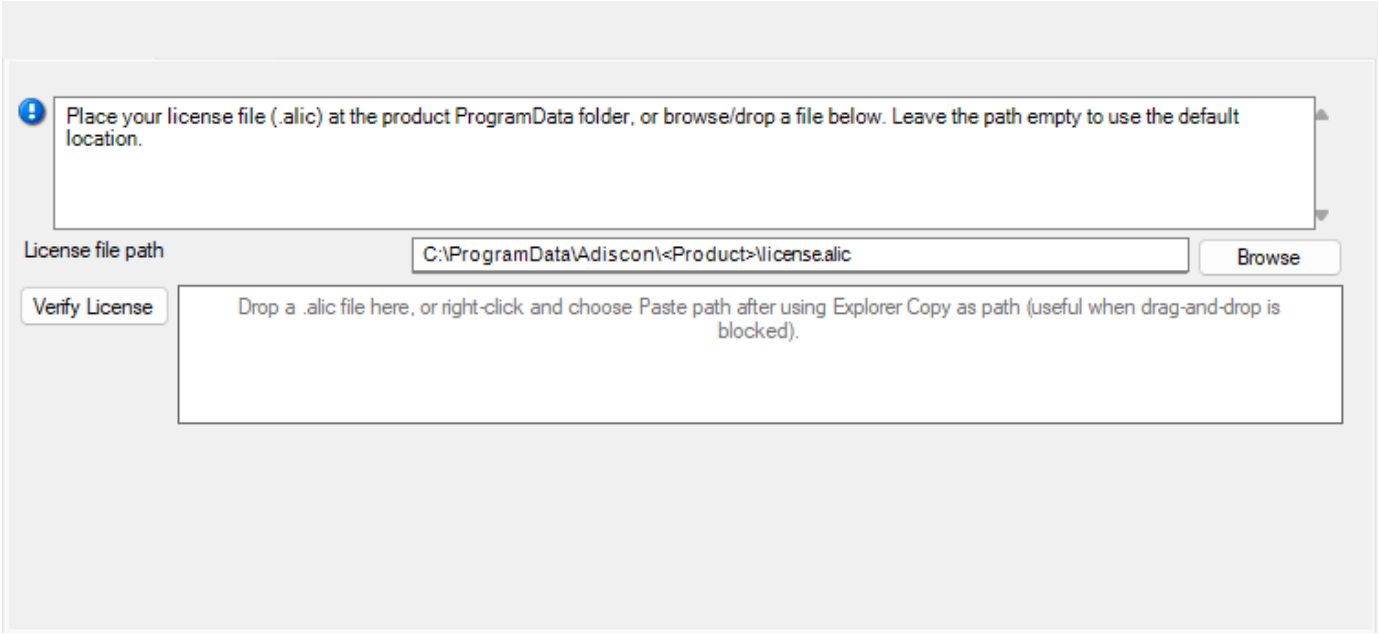
In this chapter, you find the general option settings.

License

Licensing options are on **General** -> **License** in the configuration client. Current builds use a `license.alic` license file.

License File

Use this page to apply `license.alic`.



File Configuration field:

szLicenseV2Path

Description:

Optional path to the license file. When empty, the service uses the default location:

```
%ProgramData%\Adiscon\RSyslogAgent\license.alic
```

You can browse for the file, drag-and-drop it onto the client, or paste a file path. The license is validated and applied when you save the configuration.

Current configuration clients use the license file path only. They do not read or save legacy registration-name or numeric license-key configuration values. Update an existing configuration to use `szLicenseV2Path` and a `license.alic` file.

The main window status bar shows license status (for example, the licensed organization). A version mismatch between the license file and the installed service may show a warning banner until you apply a matching license.

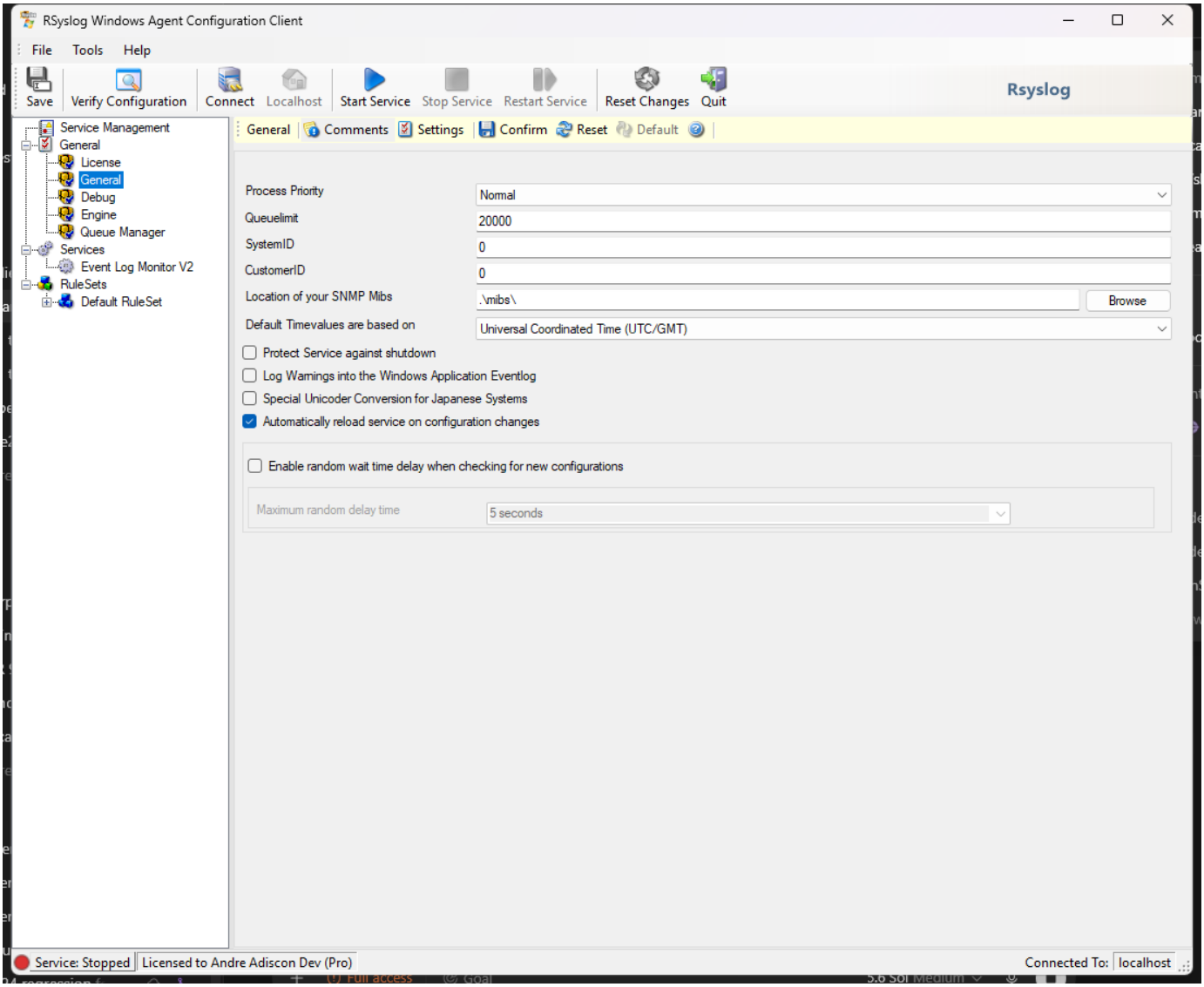
Verify License

Use this control to verify that the configured license is valid before or after you save.

If a newly issued file reports that it is not active yet, wait until the activation time shown by the client and then verify it again.

General

The General Options available on this form are explained below:



rsyslog Windows Agent General Options

Process Priority

File Configuration field:

nProcessPriority

Description

Configurable Process Priority to fine-tune application behavior.

Queuelimit

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 enqueuing 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). 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 local time output, see **`FAQ: default time values 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.**

Shutdown protection timeout

This capability is available starting with **26.08**.

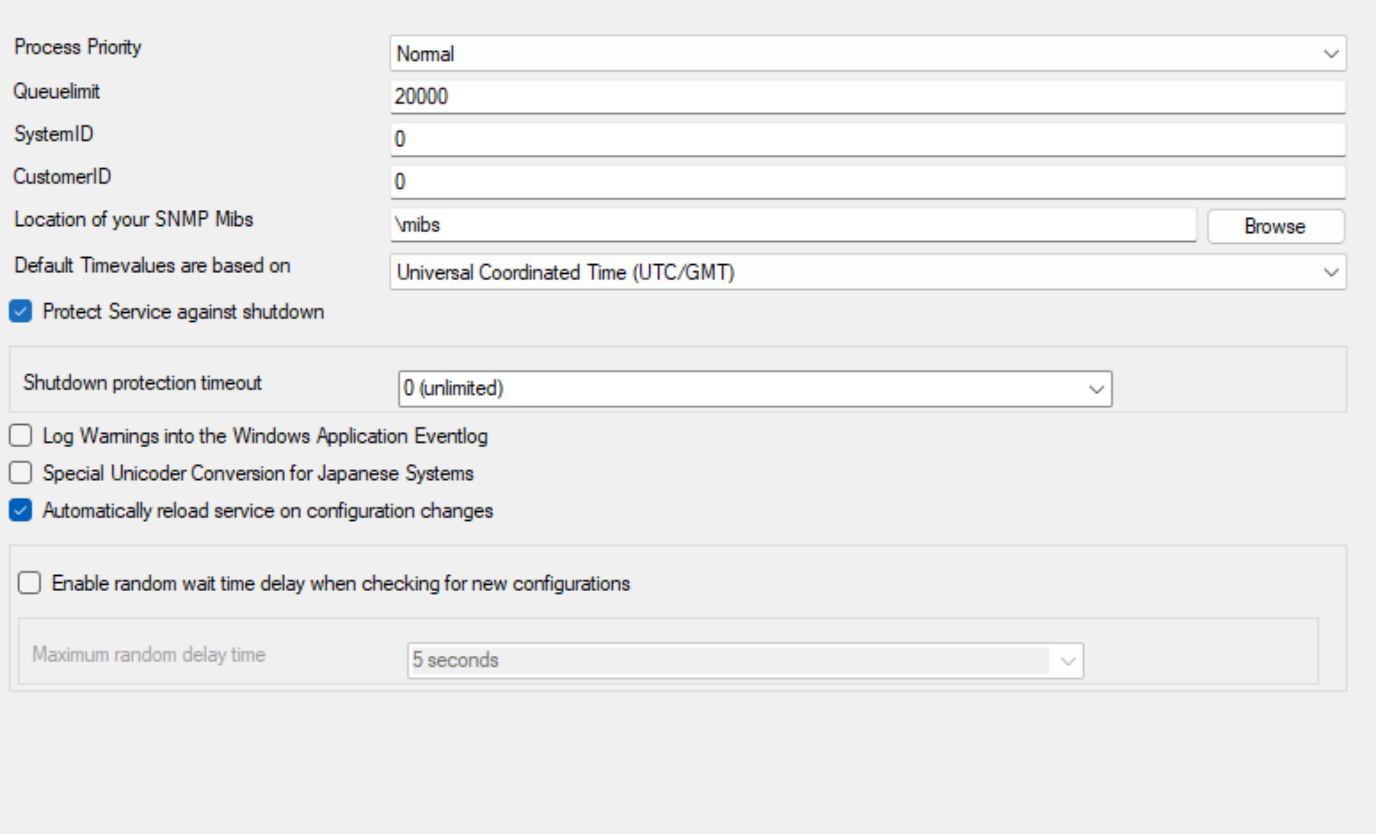
File Configuration field:

nShutdownProtectionTimeout

Description

This setting applies only when **Protect Service against shutdown** is enabled. Enter the maximum time in milliseconds for the complete protected shutdown. Set it to `0` to wait without a deadline for cooperative queue processing to finish.

A positive timeout bounds shutdown time, but messages that are still in memory or have not been stored durably can be lost when that deadline is reached.



Shutdown protection enabled with the timeout set to 0 (unlimited).

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 its 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.

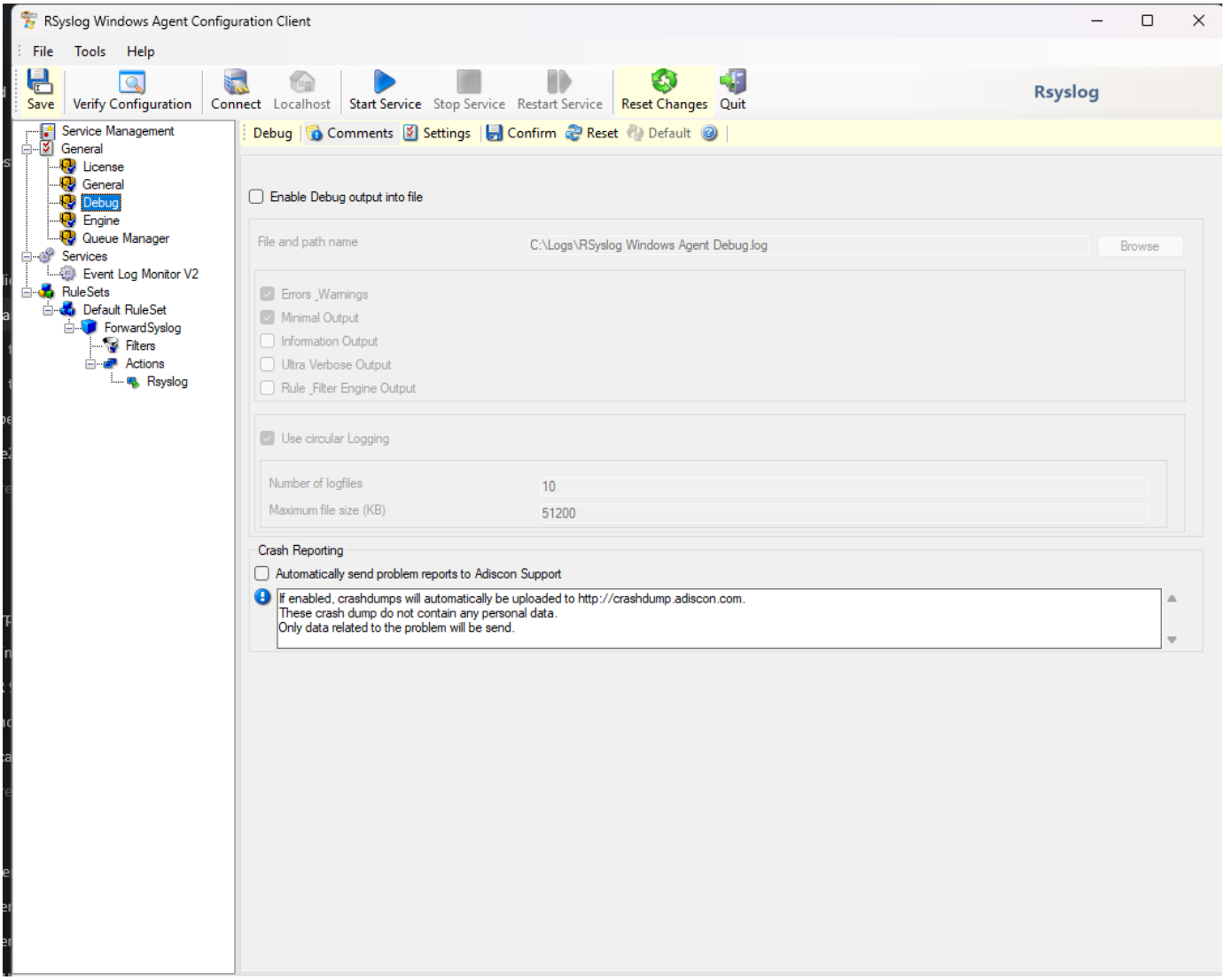
Debug

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.



rsyslog Windows Agent Debug options

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:

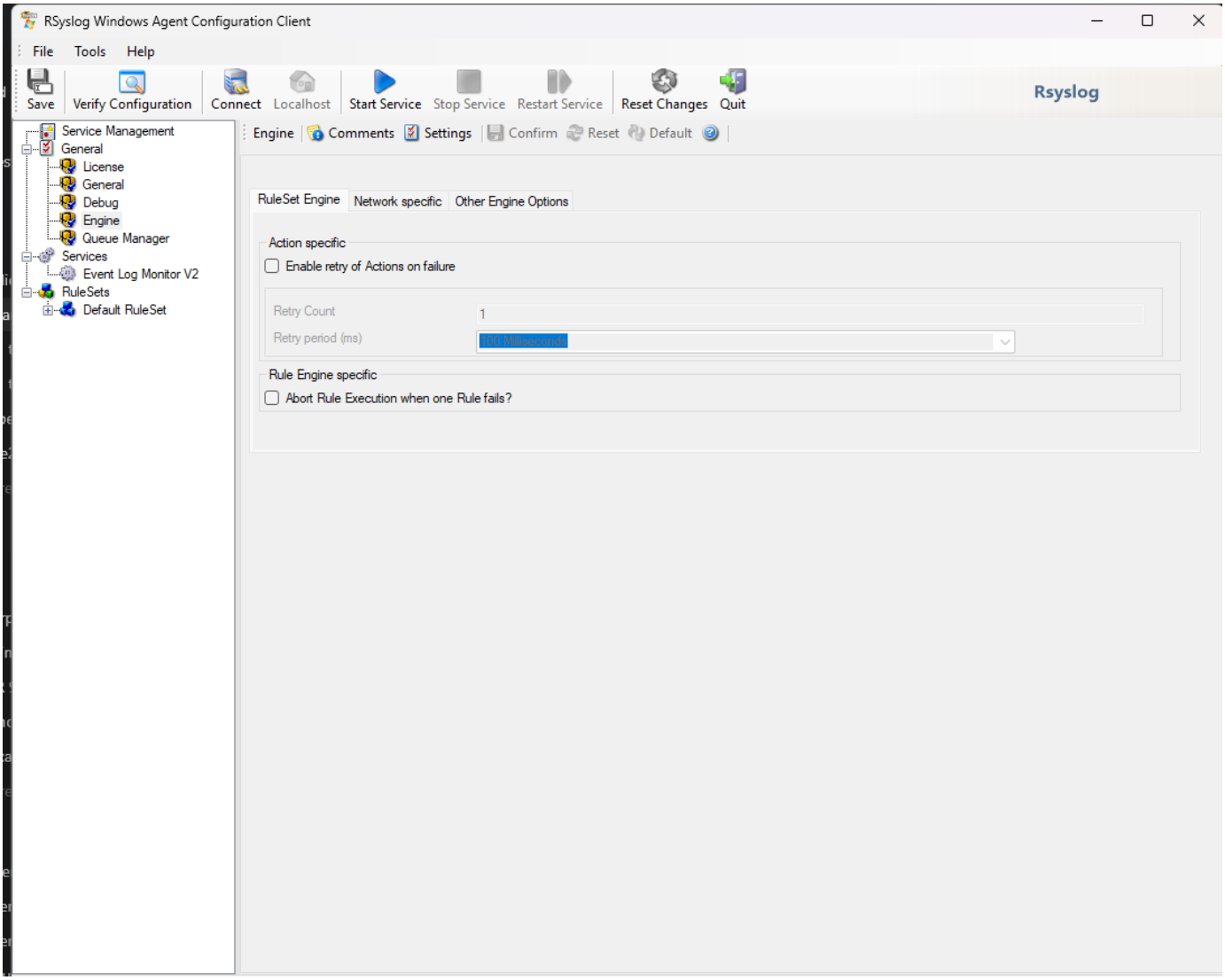
nReportCrash

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:



rsyslog Windows Agent 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:

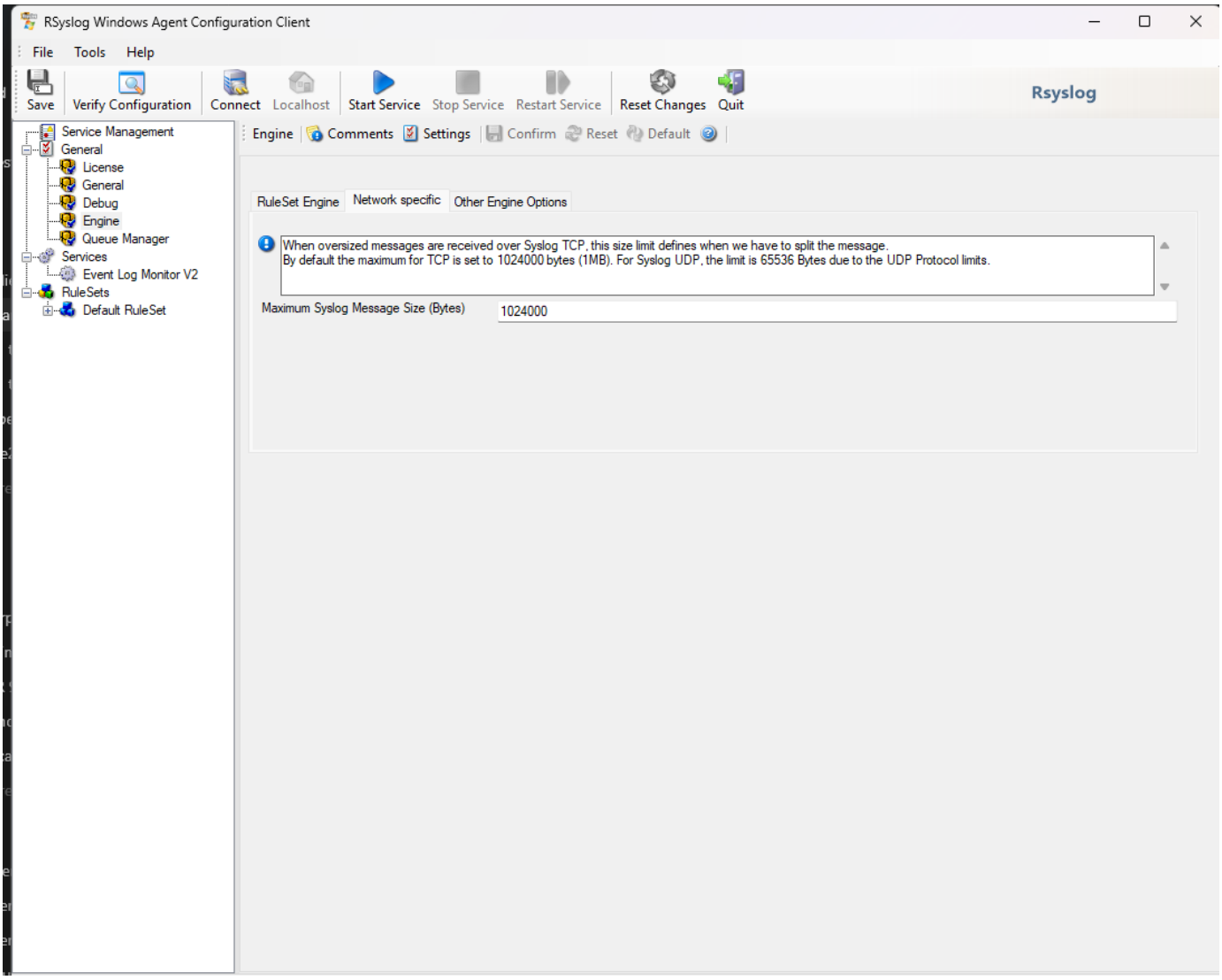
bAbortRuleOnFailure

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.



rsyslog Windows Agent 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:

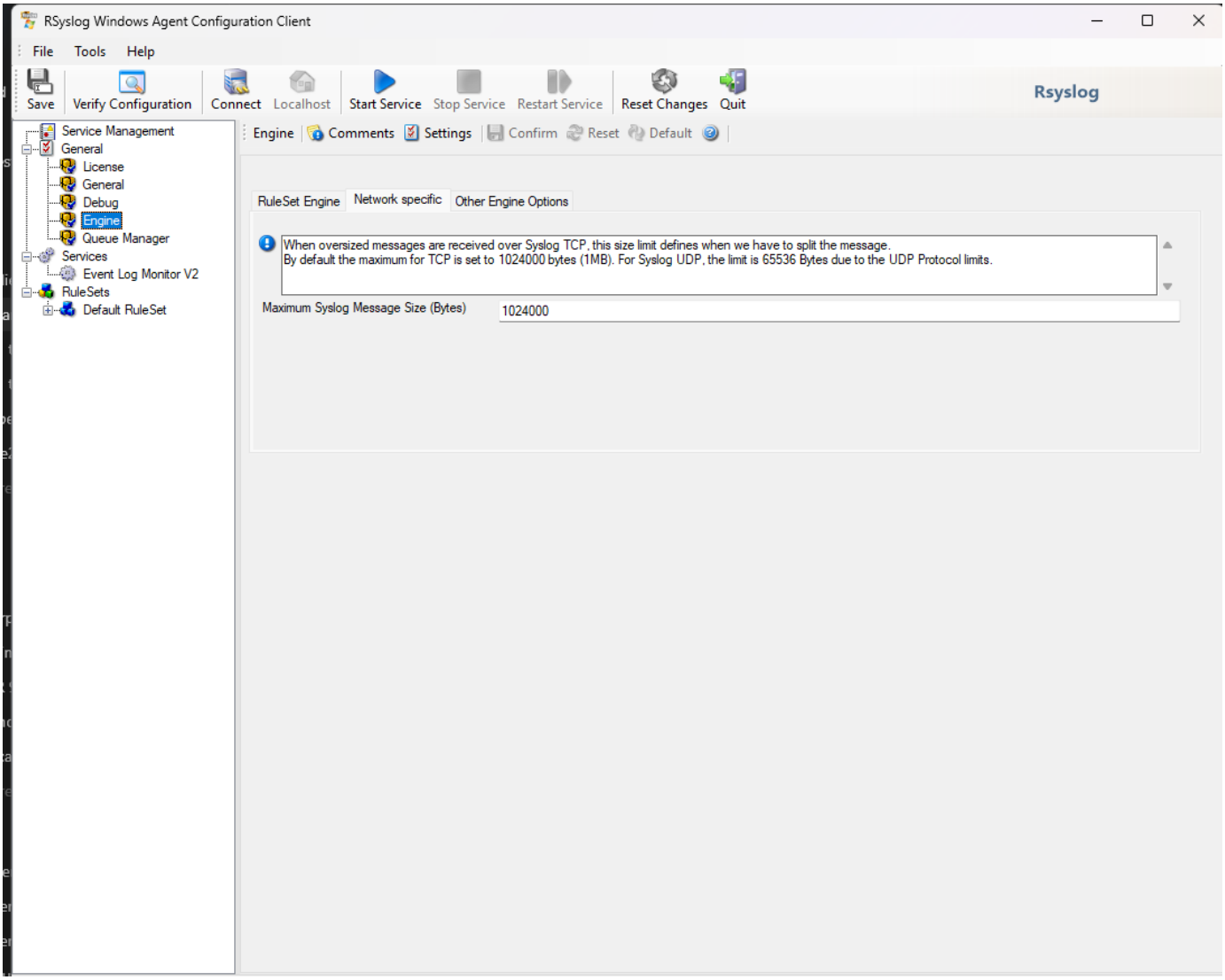
nLogRotateWorkerStopWaitTimeout

Description

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

If a queued or already active detached rotation does not finish within this bounded wait time, the unfinished detached work is saved and resumed on the next startup instead of extending shutdown indefinitely. For the full guarantee, see **`../shared/faq/log-rotation-guarantees`**.

Network specific Options



rsyslog Windows Agent Network specific Options tab

Maximum Syslog Message Size (Bytes)

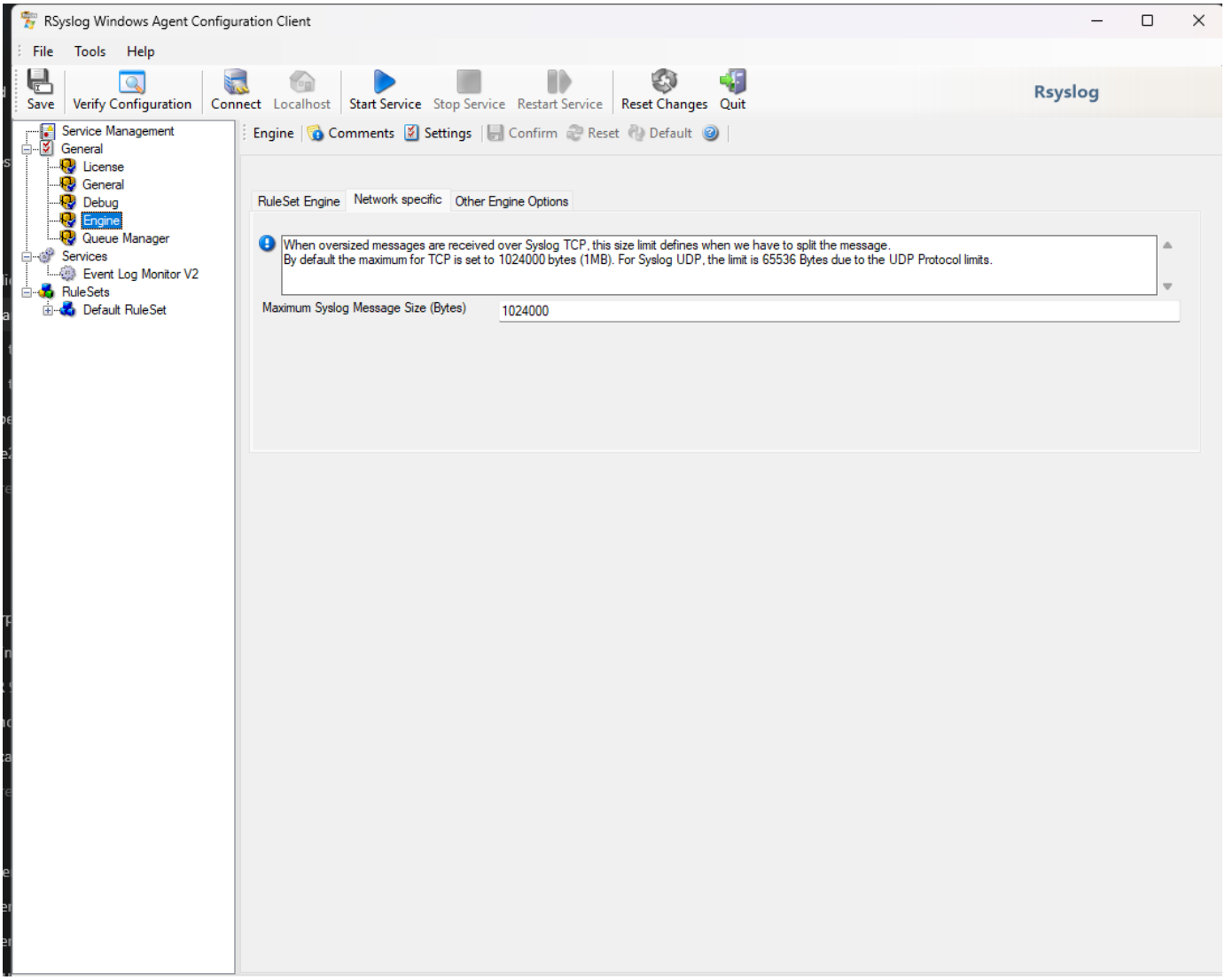
File Configuration field:

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



rsyslog Windows Agent 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?

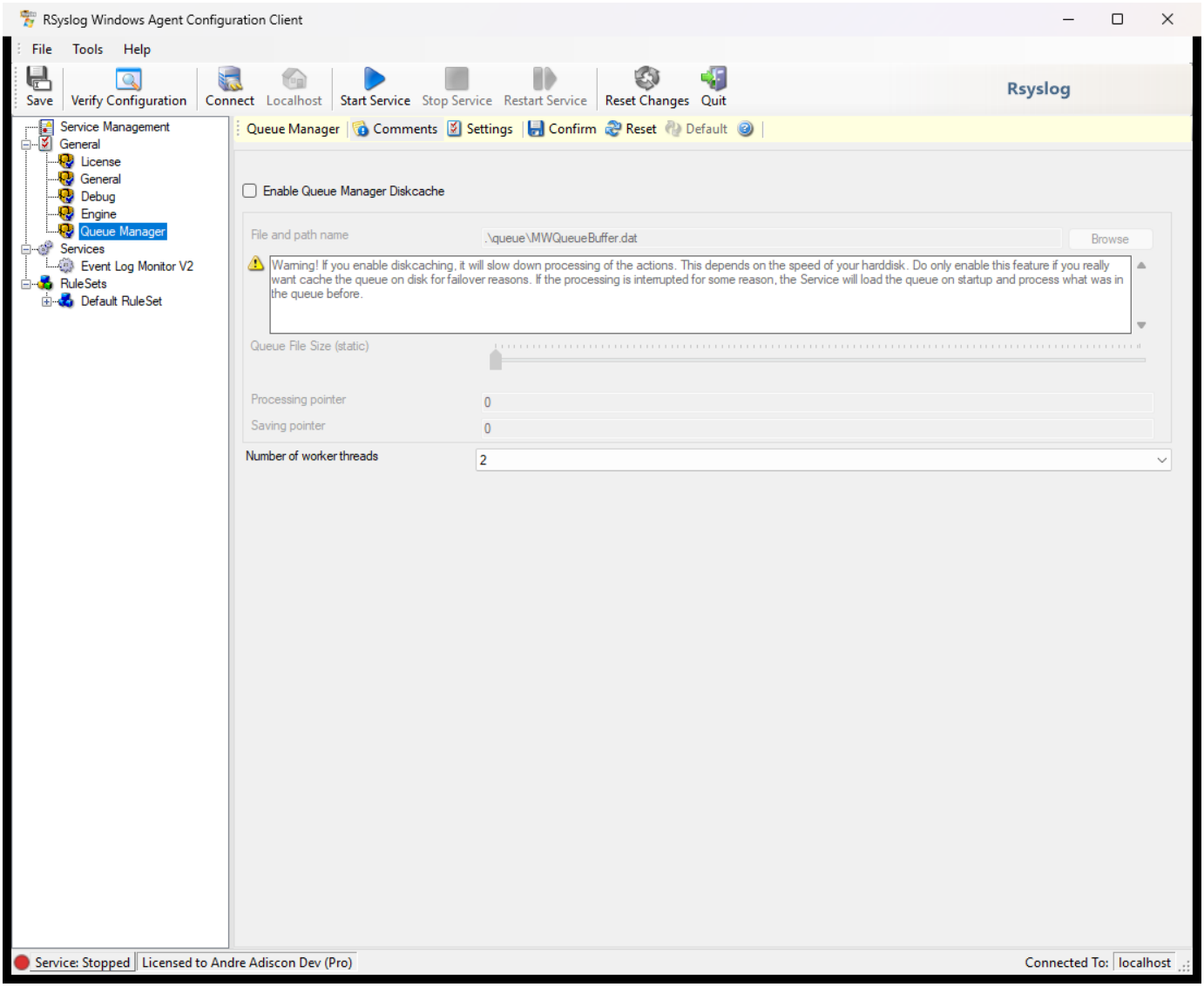
File Configuration field:

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



rsyslog Windows Agent Queue Manager options

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.

Services

Services are the configured input side of rsyslog Windows Agent. They collect or receive events and send them into the assigned ruleset.

In this manual, **input** is the clearest plain-language concept, while **service** remains the operational term for the configured rsyslog Windows Agent object.

You must define at least one input service. Without a service, the product does not collect or receive any event data.

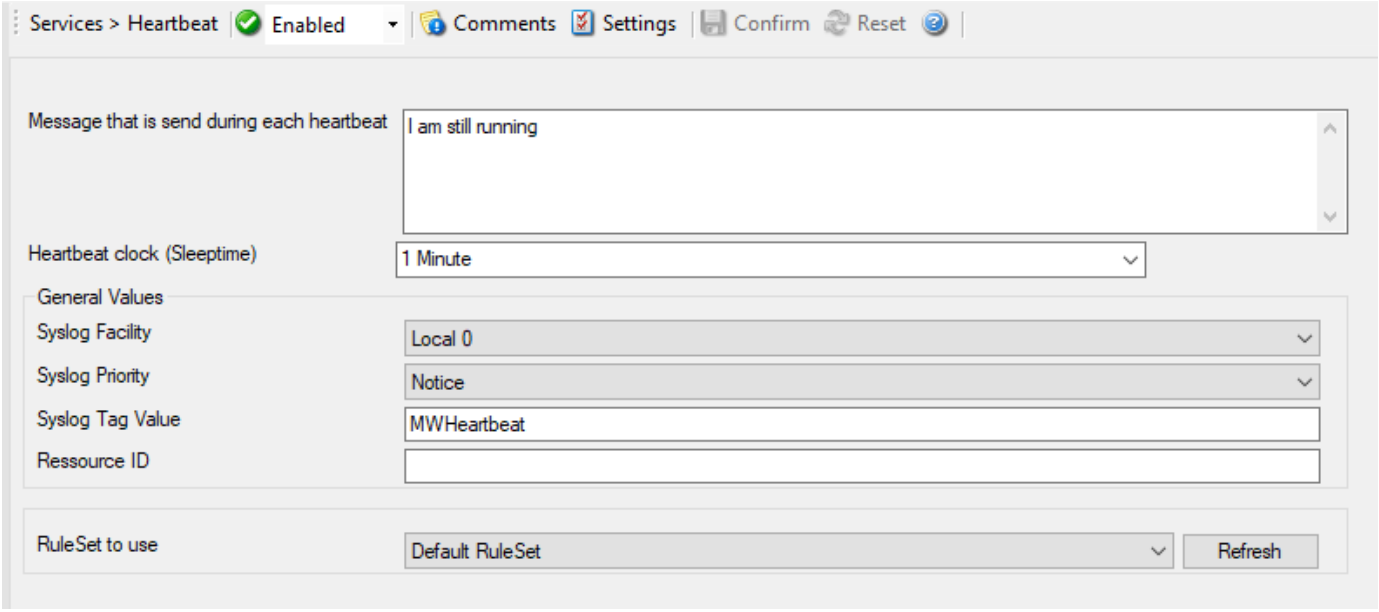
Defaults vs. active services

Service defaults are templates. They provide starting values for new services, but they do not collect data by themselves. Actual processing only happens in the active service instances you create under **Running Services**.

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:

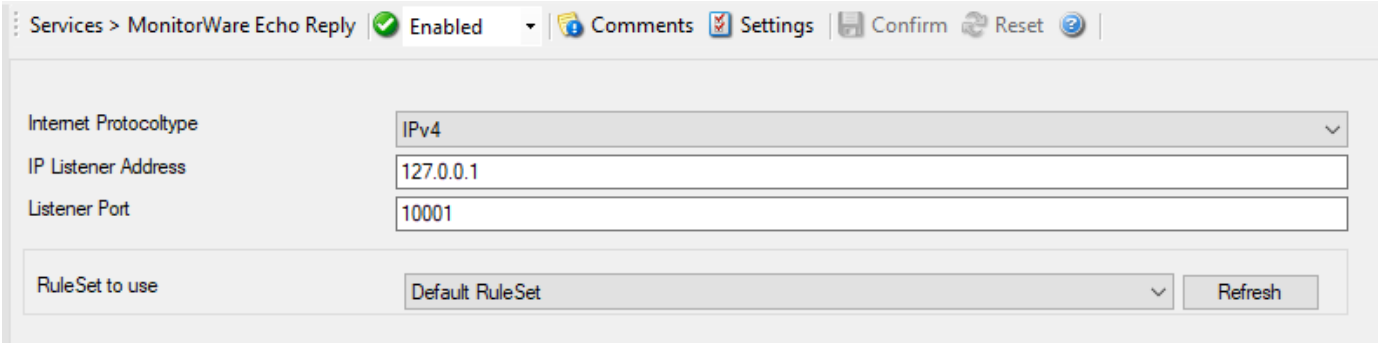
szRuleSetName

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.



- 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:

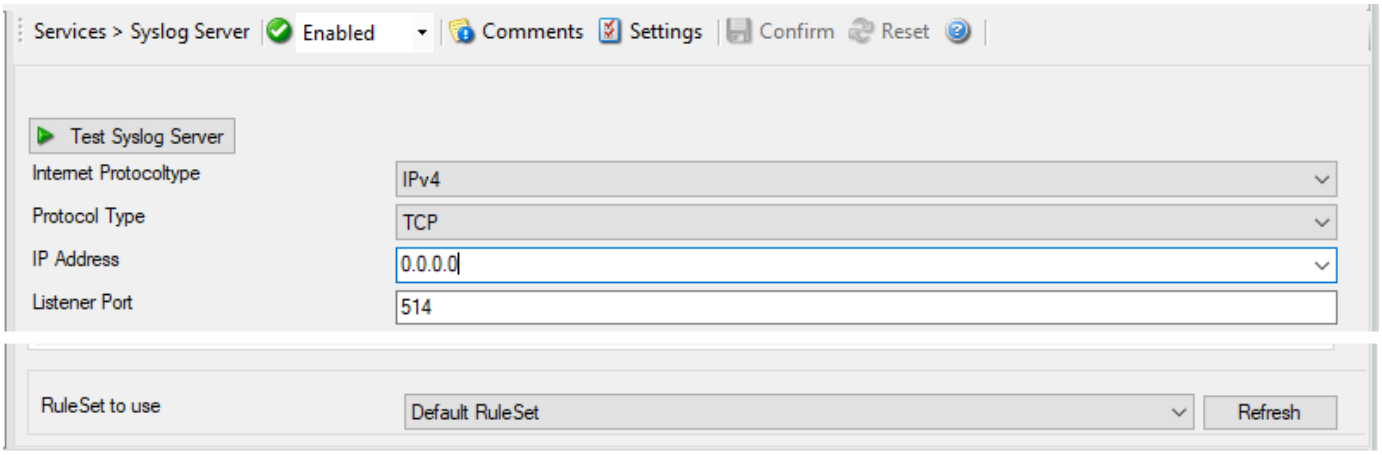
Name of the ruleset to be used for this service. The RuleSet name must be a valid RuleSet.

Network services

Syslog server

Configures the `Syslog server` service. In practice, this is the WinSyslog or MonitorWare Agent input service that receives incoming syslog messages. 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.



- Service - Syslog server Global Properties*

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:

Syslog messages can be received via udp, tcp or rfc 3195 RAW. One service instance can only use one of the protocols at a time. Typically, syslog messages are received via UDP protocol, which is the default. The `Syslog server` service 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` service 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 input services 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` service 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

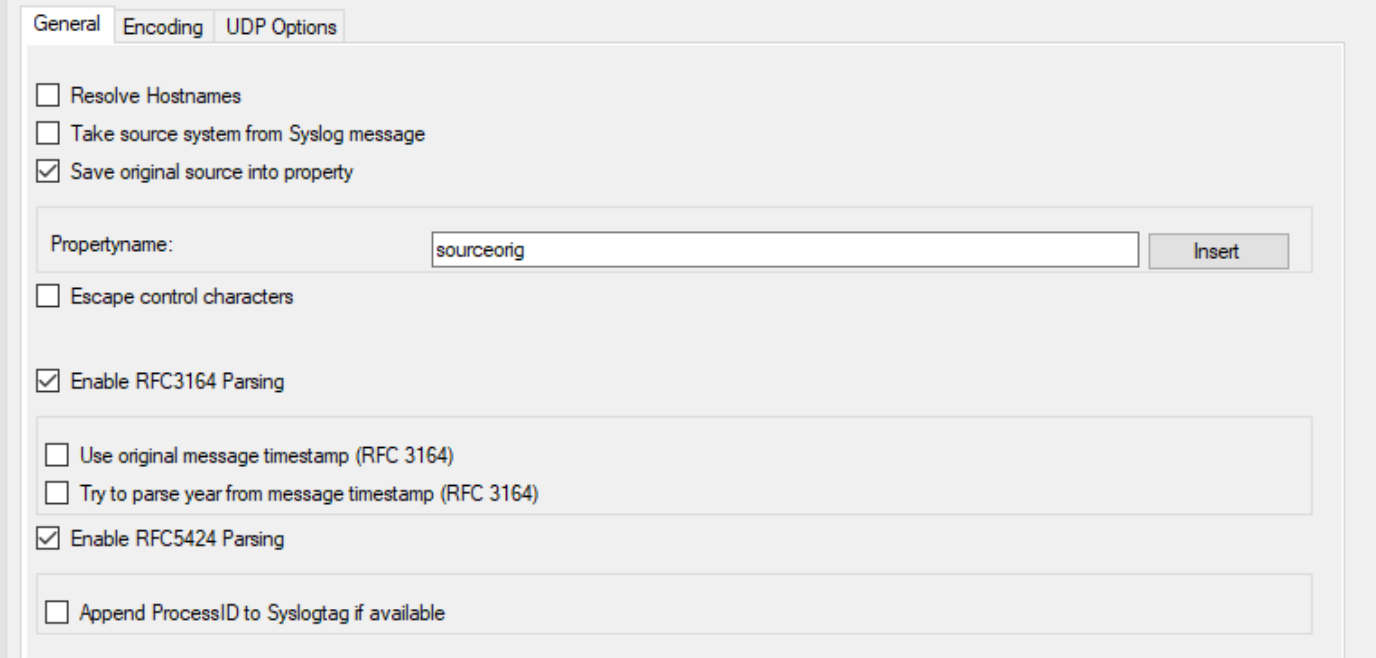
File Configuration field:

szRuleSetName

Description:

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

General Options



- 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 reserve 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

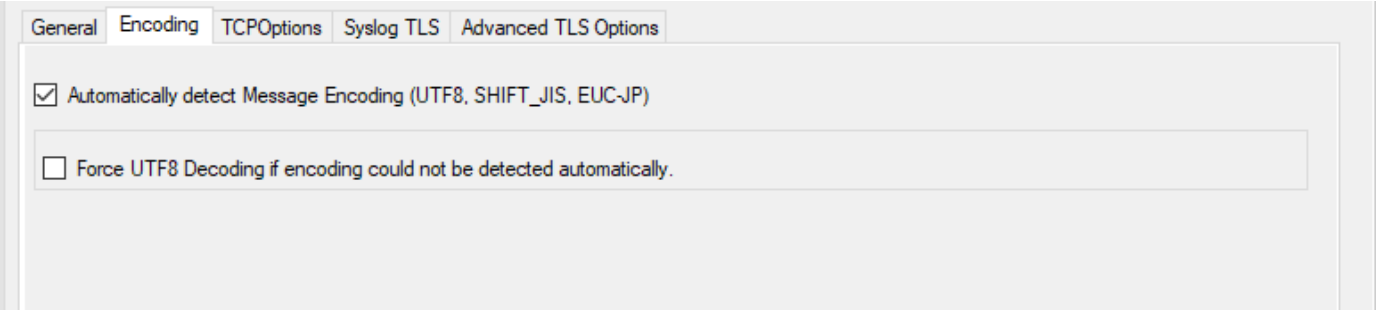
File Configuration field:

nRFC5424AddProcID2SyslogTag

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



- 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

- Service - Syslog server UDP Options Tab*

Enable receiving from a UDP Multicast Group

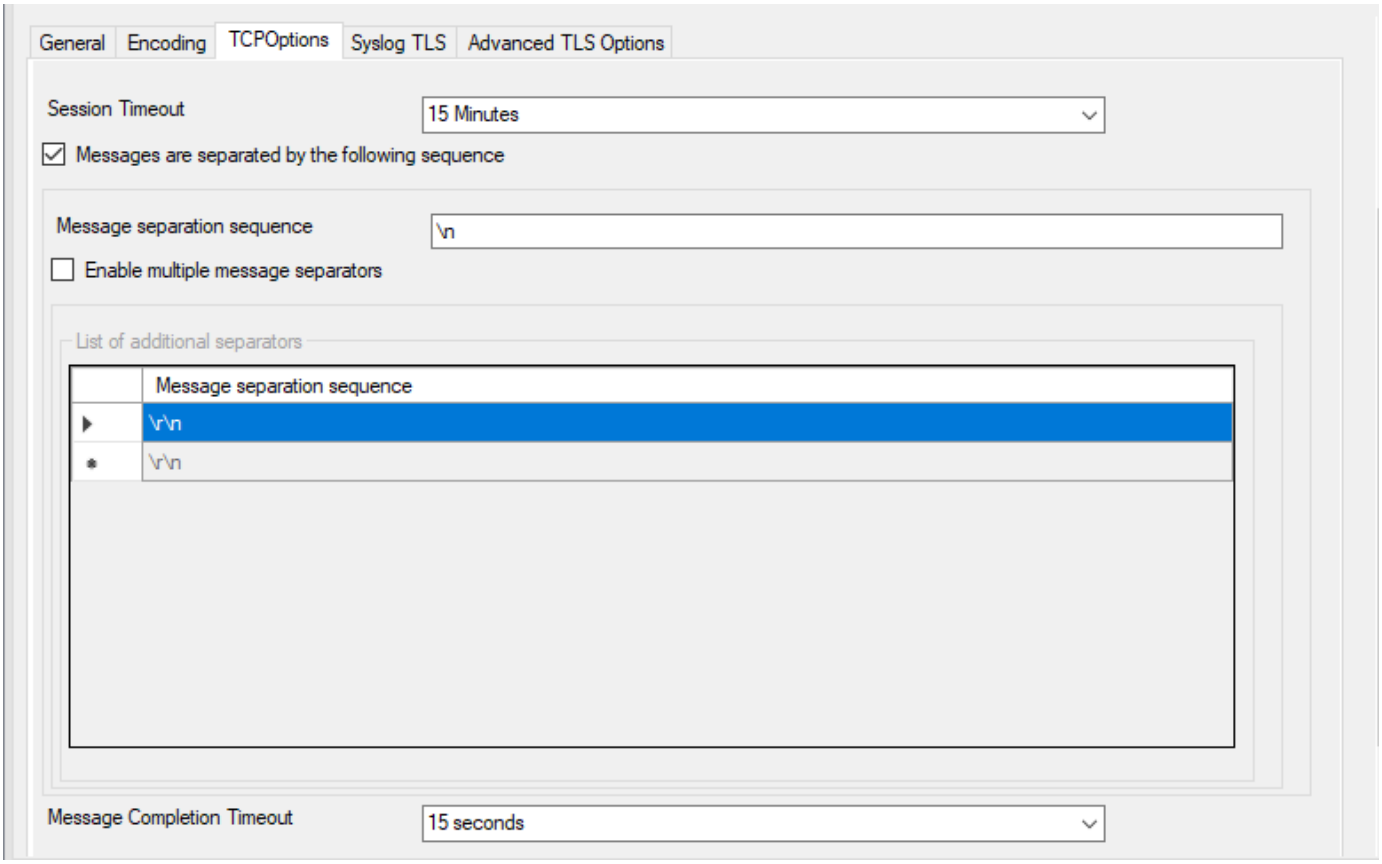
File Configuration field:

nEnableMultiCastGroup

Description:

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

TCP Specific Options



- 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.

Maximum concurrent TCP sessions

File Configuration field:

nTCPMaxSessions

Description:

Limits how many TCP sessions this listener accepts at the same time. The default is `1024`. A value of `0` disables the cap. For internet-facing or untrusted networks, keep a finite limit so stalled connections cannot consume an unbounded number of receiver threads.

TLS handshake timeout

File Configuration field:

nTLSHandshakeTimeout

Description:

Sets the timeout, in milliseconds, used while a new TLS connection completes its handshake. The default is `30000`. This setting applies before the normal TCP session timeout and helps disconnect clients that open a TLS socket but do not complete the handshake.

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 “rn” 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

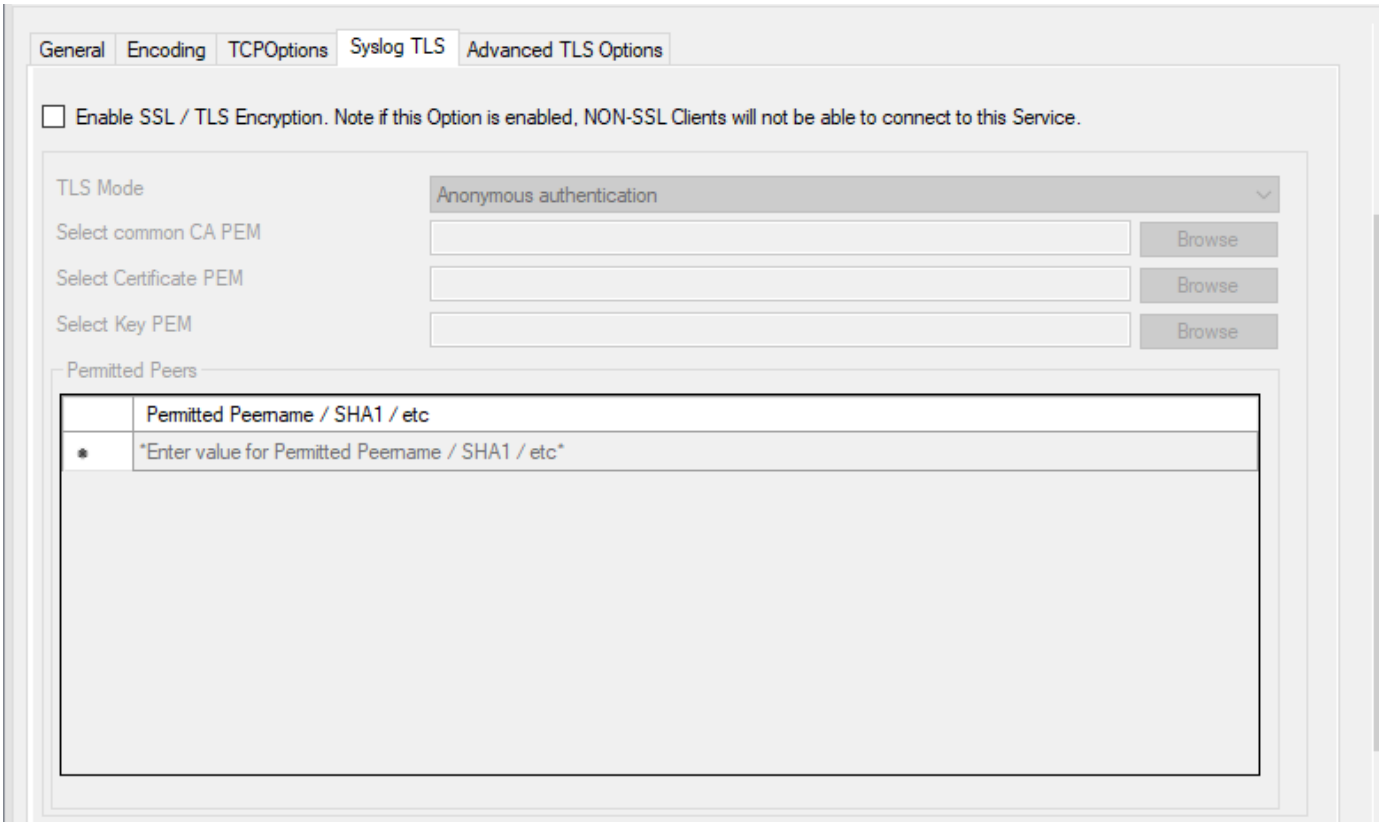
File Configuration field:

nTimeOutMsg

Description:

Here you can set the time that is allowed to complete a message. Is the time 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



- 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. This provides encryption without peer authentication. Use it only when that is intentional for the deployment; use one of the x509 modes when peer authenticity matters.

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. Newer hardened configurations can disable the legacy subject substring match and rely on certificate name checks instead.

x509/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. SHA-1 fingerprint labels remain supported for RFC 5425 interoperability. Hardened configurations can disable legacy substring matching and use exact normalized SHA-1 or SHA-256 fingerprints.

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 CA certificate or CA bundle used to validate certificates presented by connecting clients. If you use a CA chain, include the intermediate CA certificates first and the root CA certificate last.

Select Certificate PEM

File Configuration field:

szTLSCertFile

Description:

Select the server certificate in PEM format. This is the certificate the Syslog server presents to connecting clients. If needed, append the intermediate CA certificates after the server certificate so clients can validate the chain.

Select Key PEM

File Configuration field:

szTLSKeyFile

Description:

Select the private key in PEM format that matches the server certificate. Passphrase-protected private keys are not supported.

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`.

TLS hardening compatibility options

File Configuration fields:

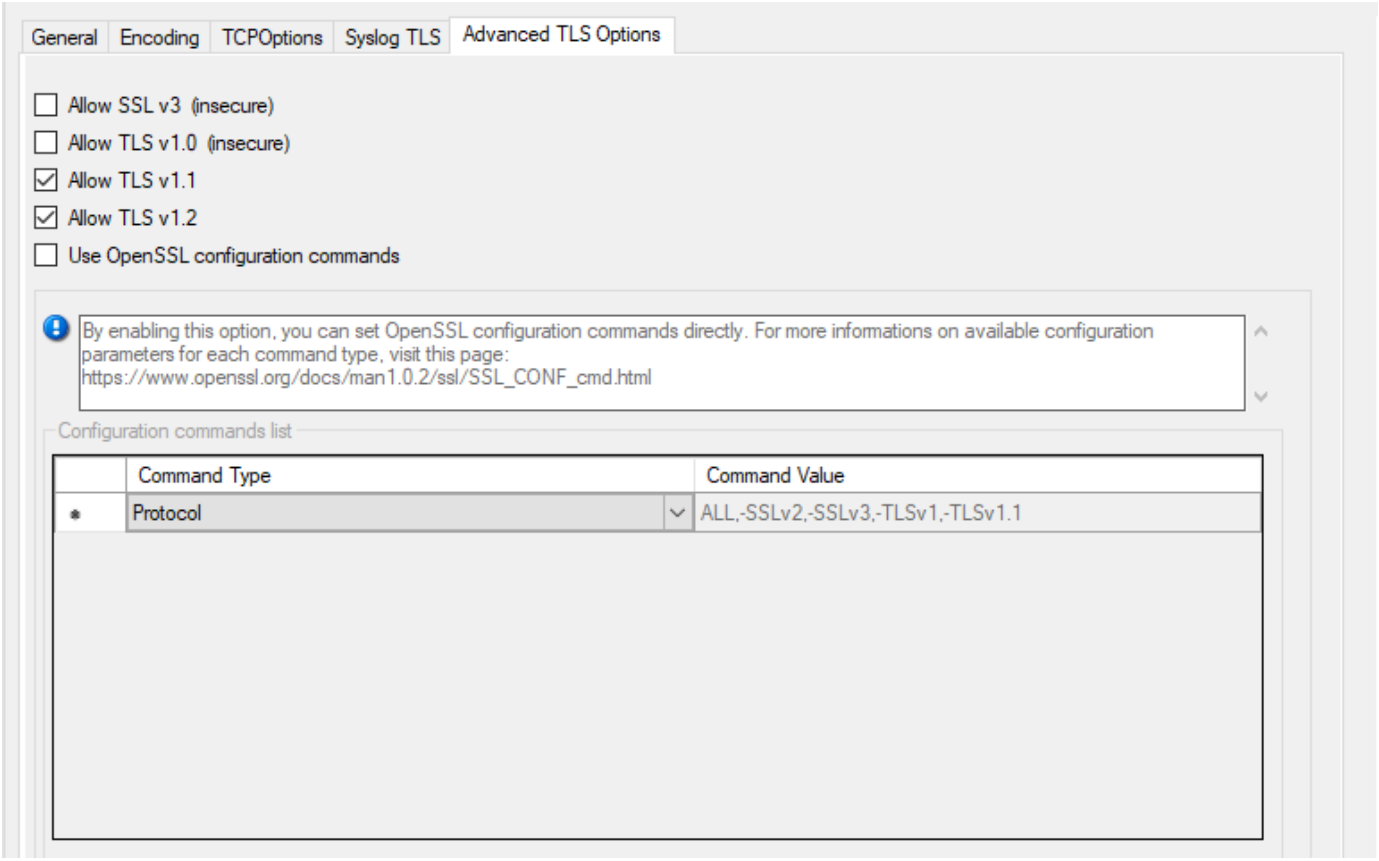
nTLSAllowExpiredCerts, nTLSLegacyPeerNameMatch, nTLSLegacyFingerprintMatch

Description:

These compatibility options control stricter certificate handling for authenticated TLS modes. The defaults preserve legacy behavior: `nTLSAllowExpiredCerts=1` accepts expired client certificates, `nTLSLegacyPeerNameMatch=1` allows the historic subject substring match for x509/name mode, and `nTLSLegacyFingerprintMatch=1` allows the historic SHA-1 fingerprint substring match.

Recommended hardened values are `0` for all three settings. With those values, expired client certificates are rejected, x509/name mode uses certificate name matching, and x509/fingerprint mode requires exact normalized SHA-1 or SHA-256 fingerprints.

Advanced TLS



- 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.

Fail on TLS configuration command error

File Configuration field:

nTLSFailOnConfigCommandError

Description:

Controls whether the listener fails to start TLS when an OpenSSL configuration command cannot be applied. The default is `0`, which keeps compatibility by logging the error and continuing. Set this value to `1` for hardened deployments where a weaker-than-intended TLS configuration should stop the listener instead of being accepted.

Windows and file monitoring

Event Log Monitor V1

This dialog configures the Windows Event Log Monitor service.

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

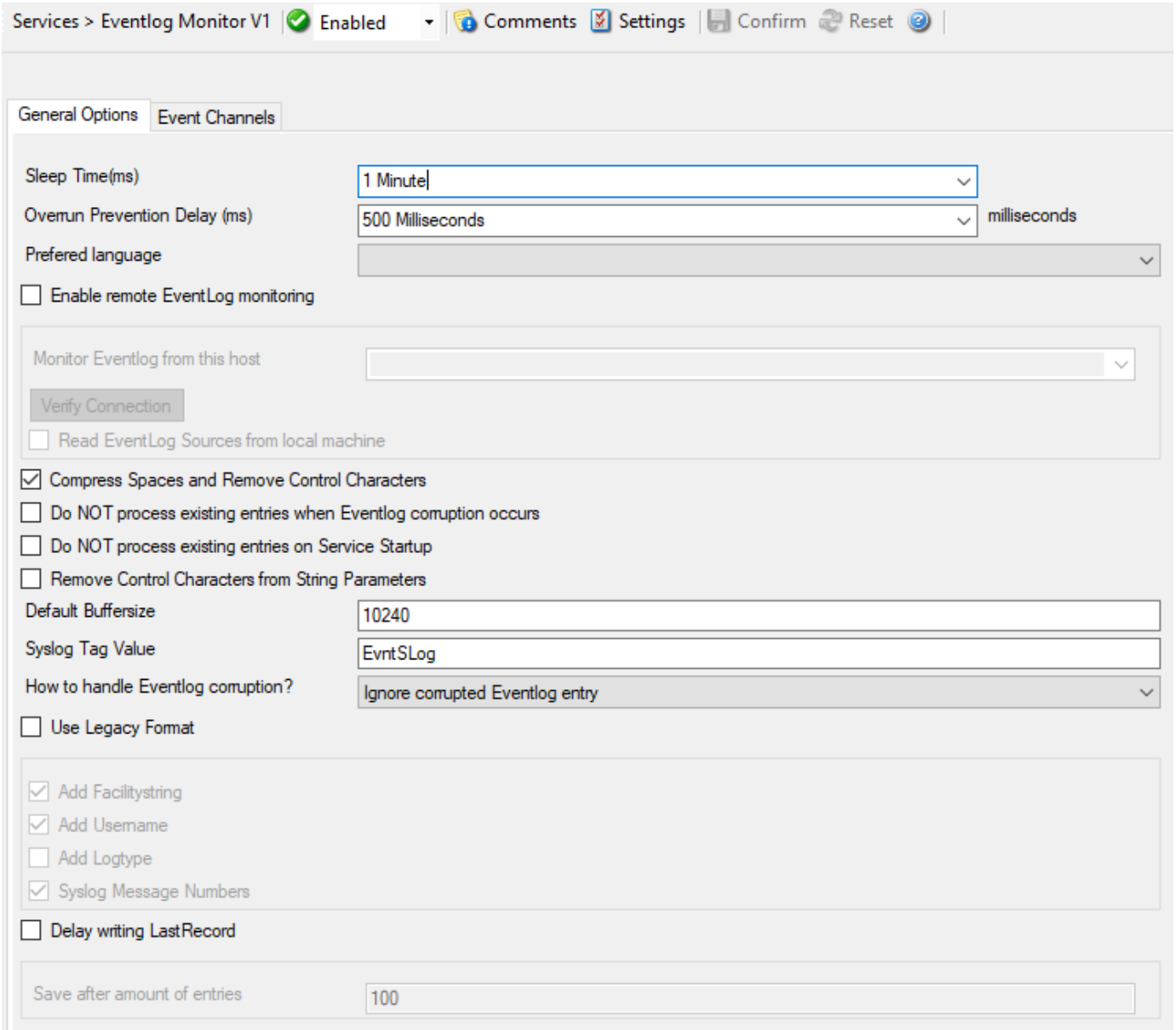
Newer Windows versions come with a considerably changed event logging system. In theory, the Event Log Monitor works with them, too. However, we know of some incompatibilities. For best results, we recommend using the event log monitor v2 service, which was specifically written for Windows Vista and newer. The Event Log Monitor described here is applicable for legacy Windows systems, and XP (where the new event logging system is not available). The Client will automatically detect and load available Event Log types during the first startup of the Event Log Monitor.

Event Log Monitor V1

- Windows XP
- Windows 2003

Event Log Monitor V2

- all modern Windows versions (Windows 10, 11, Server 2016, 2019, 2022, and newer).



- Service - Event Log Monitor V1*

General Options Tab

Sleep Time(ms)

File Configuration field:

nSleepTime

Description:

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

We suggest a value of 60,000 milliseconds for the “Sleep Time”. With that setting, the Event Log Monitor checks for new events every 60 seconds. Larger periods can be specified for occasionally connected systems or if email delivery with few emails per day is intended.

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

Overrun Prevention Delay (ms)

File Configuration field:

nPreventOverrunDelay

Description:

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

If run at a value of zero, the Event Log Monitor service generates events as fast as the machine permits. We have seen scenarios where routers and receivers are not able to keep up with this rate, resulting in packet loss. In addition, the CPU of the reporting machine is run at 100% - which is not a problem because the service runs at a low priority. However, with even a 1-millisecond delay, there is no noticeable CPU activity even when large bursts of events are forwarded. At one millisecond, the service can still generate 1000 events per second.

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

Preferred language

File Configuration field:

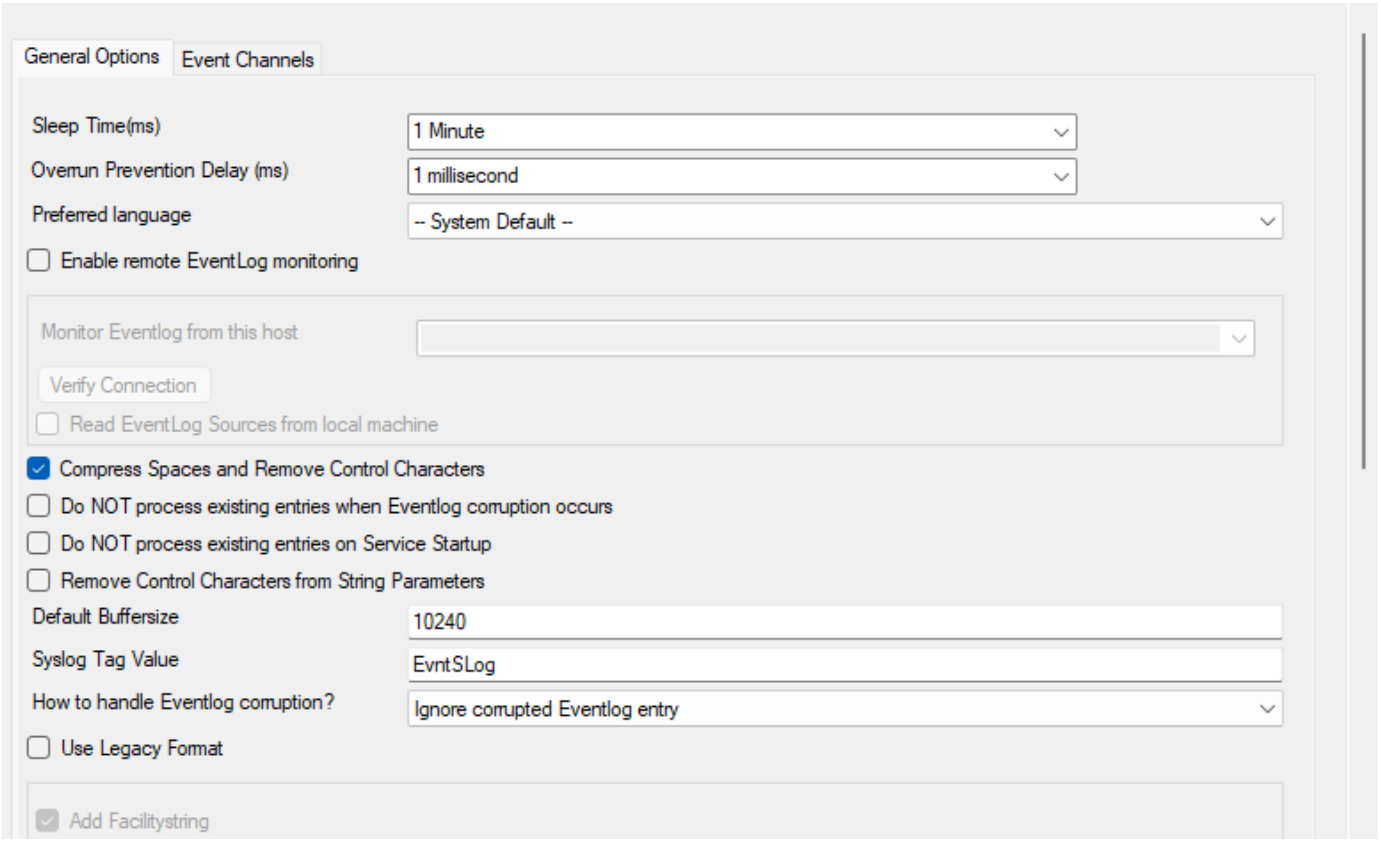
nLanguageLCID

Description:

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

System Default is available as an explicit selection starting with **26.08**.

System Default is a valid selection and is stored as `0`.



Event Log Monitor V1 preferred language.

Enable remote Event Log monitoring

File Configuration field:

nEnabledRemote

Description:

If enabled, the Event Log Monitor will read and process the EventLog from a remote machine. Use the verify button to make sure that the network connection is working correctly.

Please make sure that the machine, which you are going to monitor, does have File and Print Services enabled, and is accessible by this machine.

This is important as the Event Log Service will read the message libraries on the remote machine by using the default administrative shares. For this reason, the Service must be configured to run with a user who has administrative privileges/permissions on the local and remote machine. If File and Print services remain disabled, the local message libraries will used automatically instead. Note that you may experience a lot of missing message libraries in this case.

Additionally you have an option to read the Event Log Sources from the local machine. If enabled, the local message libraries will be used instead of the remote machines ones. Sometimes local Event Sources are more reliable or are required for third-party EventLog implementations.

Compress Spaces and Remove Control Characters

File Configuration field:

nCompressControlChars

Description:

This option allows you to control the control character removal and space compression. If checked, control characters (e.g. CR, LF, TAB - non printable characters in general) are removed. Also multiple spaces are compressed to a single one. By default this is checked. We recommend keeping it checked for most cases as it provides better display.

Please note that it should be unchecked if events should be forwarded via email. And it MUST be turned off if double-byte character sets are being processed (e.g. Japanese).

Do NOT process existing entries when Event Log corruption occurs

File Configuration field:

nDoNotProcessLastRecord

Description:

When this option is checked, it prevents reprocessing of the whole Windows Event Log when the log becomes corrupted or truncated. This helps avoid re-reading all existing entries after such an event.

Do NOT process existing entries on Service Startup

File Configuration field:

szSyslogTagValue

Description:

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

Remove Control Characters from String Parameters

File Configuration field:

nRemoveControlCharsFromParams

Description:

Enable this option to remove control characters like carriage return or line feeds from parameter strings and category names in Windows Events.

Default Buffersize

File Configuration field:

nDefaultBuffer

Description:

The default Buffersize is 10k. This value will be increased or decreased dynamically if necessary. If you want to use third-party applications like NetApp you must increase the Buffersize manually (minimum 65k), because dynamic adjusting is not possible with them.

SyslogTag Value

File Configuration field:

szSyslogTagValue

Description:

The SyslogTag Value determines the SyslogTag that is used when forwarding Events via syslog. This is useful, if you want to determine later, what kind of syslog message this is, perhaps because you log Event Logs and syslog into the same database.

How to handle Eventlog corruption

File Configuration field:

nEventLogCor

- 0 = Retry processing from beginning
- 1 = Ignore corrupted Eventlog entry
- 2 = Clear all events from Eventlog

Description:

Sometimes it can occur that Event Log messages are corrupted and cannot be read correctly. This usually happens if someone tampered with the Event Log or if you are processing the Eventlog for the first time. In cases like this, you can automatically handle the situation with this option. You have the following options:

- Retry processing Event Log from the beginning: in this case the complete Eventlog will be processed again.
- Ignore corrupted Event Log entry (default): the affected Eventlog entry will be ignored and processing will continue.
- Clear all Events from the Event Log: the Event Log will be cleared completely and new Events hopefully don't get corrupted before they are processed.

Use Legacy Format

File Configuration field:

bUseLegacyFormat

Description:

This option enhances compatibility to scripts and products working with previous versions of EventReporter. The legacy format contains all Windows Event Log properties within the message itself.

The new format includes the plain text message only. The additional information fields (like event ID or event source) are part of the XML formatted event data. If the new format is used, we highly recommend sending or storing information in XML format. This is an option in each of the action properties (of those actions that support it – the write to database option for example always stores the fields separated, so there is no specific option to do so).

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

Add Facility String

File Configuration field:

bAddFacilityString

Description:

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

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

Add Username

File Configuration field:

nAddUserName

Description:

If checked, the Windows user that generated the event log entry is transmitted. If unchecked, this information is not forwarded. This is a configurable option for customers who have written parsing scripts for a previous format which did not contain Usernames. This option must also be unchecked if MoniLog is used.

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

Add Logtype

File Configuration field:

nAddLogType

Description:

If checked, then log types e.g. system, security etc. is prepended to the generated message.

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

Syslog Message Numbers

File Configuration field:

bShowSyslogMsgNbr

Description:

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

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

Delay writing LastRecord

File Configuration field:

nEnableLastRecordDelay

Description:

Enables the LastRecord writeback delay to the configured properties below.

Save after amount of entries

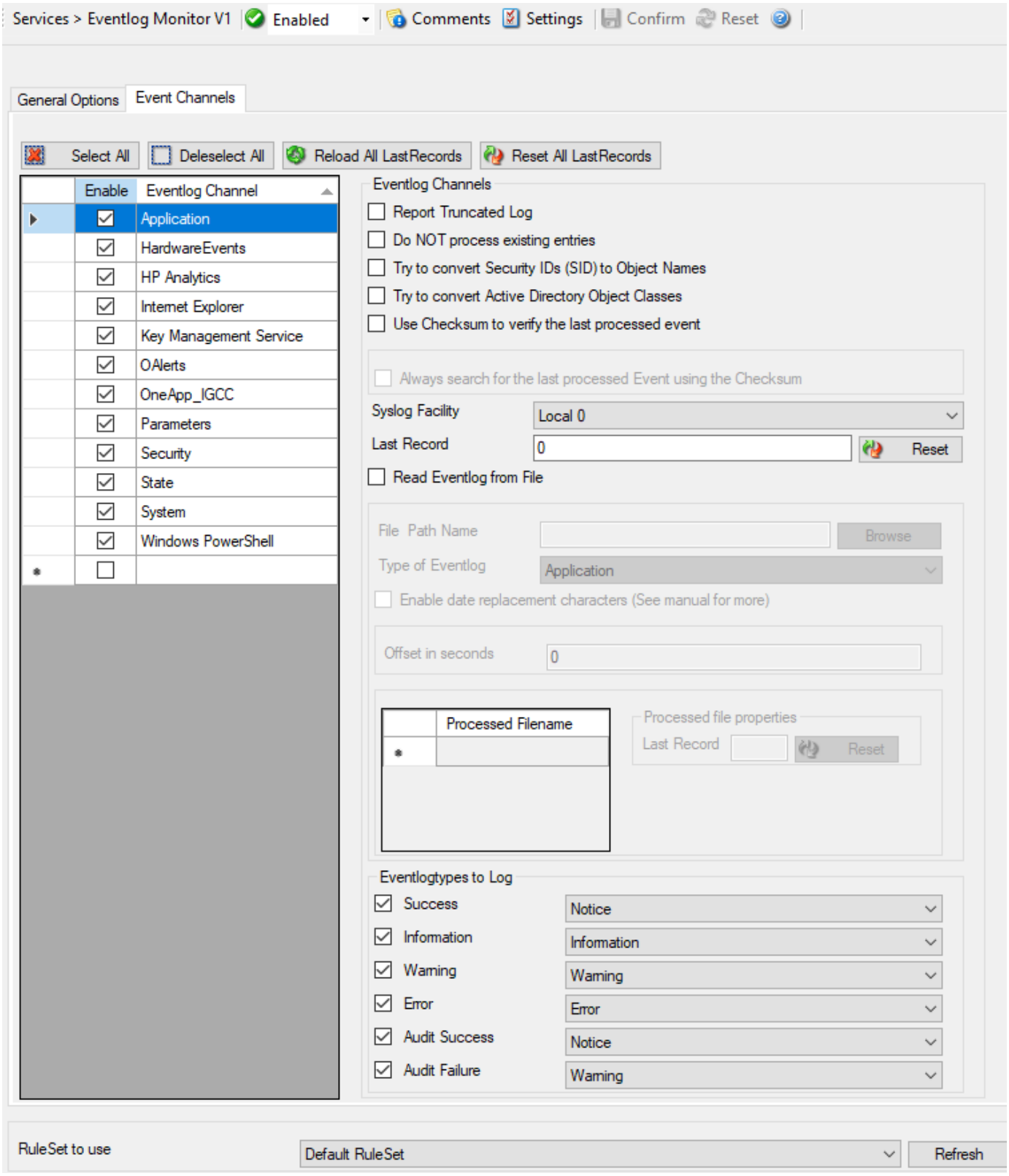
File Configuration field:

nLastRecordDelayCount

Description:

The LastRecord will be written after the amount of processed event log entries that are specified here.

Event Channels Tab



- Service - Event Log Monitor V1 Channels Tab*

Event Log Channels

They are basically a list of the different log types. The corresponding log is only be processed if the respective “Enable” checkbox is checked. The parameters are common to all logs and are explained only once.

Report Truncated Log

File Configuration field:

bReportTruncatedLog

Description:

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

The service detects event log truncation. If “Report Truncated Log” is checked, it generates a separate message stating the truncation. This option is most useful in environments where truncation is not expected and as such might be an indication of system compromise.

If you regularly truncate the Windows Event Logs as part of your day-to-day operation, we suggest you turn this option off. In this case, we also recommend using a short sleep period (for example 10,000 which is 10 seconds) to avoid losing log entries.

Do NOT process existing entries

File Configuration field:

nNoExistingEntries

Description:

If you do not want to get a dump of an existing specific Windows Event Log then use this option. When MonitorWare Agent / EventReporter are restarted it will start processing after that last entry and do not look for the previous entries.

Try to convert Security IDs (SID) to Object Names

File Configuration field:

nTryConvSIDtoObj

Description:

With this option you can convert Security ID’s (SIDs) to object names. You can enable this feature in the advanced configuration of each event log type in the Event Log Monitor service. Simple check the “Try to convert Security IDs (SID) to Object Names” option.

Note that only the Security event log has this feature enabled by default. For all other event log types this feature is disabled by default.

Try to convert Active Directory Object Classes

File Configuration field:

nTryConvertDsClasses

Description:

With this option you can convert ActiveDirectory Schema GUID’s from Security Events on Domain Controllers to object names. For example Event 565, which usually has a lot of these Schema GUID’s! The GUID’s are internally cached to speed up EventLog processing operations.

Note that only the Security event log has this feature enabled by default. For all other event log types this feature is disabled by default.

Use Checksum to verify the last processed event

File Configuration field:

nEventUseChecksum

Description:

A checksum of the last processed Event will be stored along with the LastRecord of an event log. This checksum is checked during each iteration. If the checksum does not match , we consider the EventLog has

been altered, cleared, or something else happened. In this case the EventLog is being reprocessed from the beginning.

Please note: This option will prevent you from modifying the LastRecord value. If you do, the whole EventLog will be reprocessed! Please note that this behavior is by design and cannot be avoided. So we recommend to use this feature only if you intend to double check if the LastRecord value is valid.

Always search for the last processed Event using this Checksum

File Configuration field:

nEventScanLastEventByChecksum

Description:

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

Syslog Facility

File Configuration field:

nFacility

Description:

The syslog facility to map information units stemming from this log to. Most useful if the message is to forward to a Syslog daemon.

Last Record

File Configuration field:

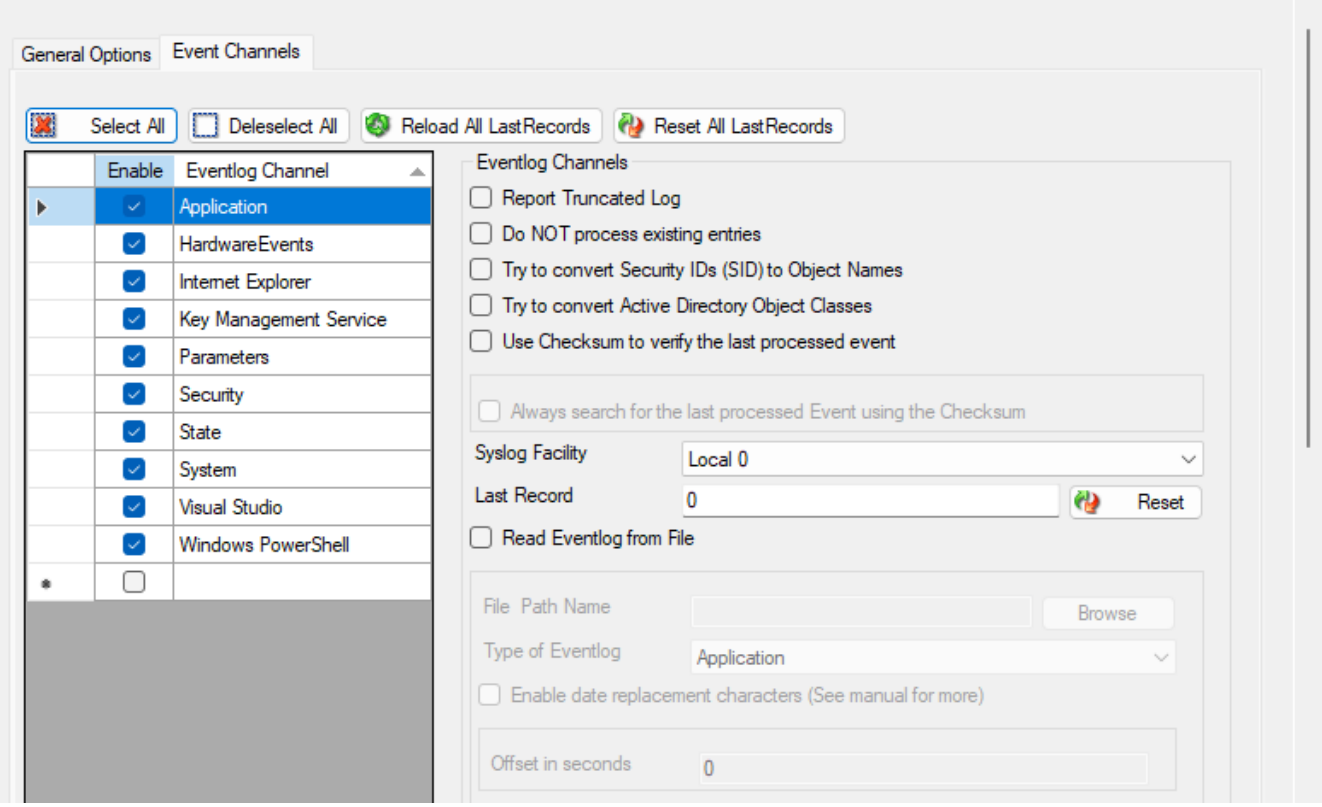
nLastRecord

Description:

Windows Event Log records are numbered serially, starting at one. The service records the last record processed. This textbox allows you to override this value.

Automatic refresh of the persisted bookmark values when you open this service property window is available starting with **26.08**.

When you open this service property window, the configuration client refreshes the persisted bookmark values from the service. It does not replace unsaved edits in the open form.



Event Log Monitor V1 Event Channels tab with the Last Record bookmark.

Use it with caution!

If you would like a complete dump of a specific Windows Event Log, reset the “Last Record” to zero with the reset button. If you missed some events, simply reset it to some lower value than currently set. It is possible to set “Last Record” to a higher value. This suspends event reporting until that record has been created. We strongly discourage to use this feature unless definitely needed.

Read Eventlog from File

File Configuration field:

nReadFromFile

Description:

When enabled, the Eventlog is read from a file instead from the system.

File Path Name

File Configuration field:

szLogFileName

Description:

It defines that which file to be read, only available when “Read Eventlog From File” is enabled.

Type of Eventlog

File Configuration field:

szLogType

Description:

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

Enable date replacement characters

File Configuration field:

nEnableDateReplacements

description:

Allow the use of dynamic files/paths when using evt files. The same replacement characters as in the FileMonitor apply to this feature. A configured filename may look like this: `C:\temp\evt_%Y%m%d.evt` and would be replaced with `C:\temp\evt_20130101.evt`.

To support changing log file names, there are replacement characters available within the file name. These are:

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

Offset in seconds

File Configuration field:

nEnableDateReplacementsOffset

Description:

When “Enable date replacement characters” is enabled, the current date will be used to generate the filenames. However in certain cases, there is a need to generate filenames with past or future dates. Negative values will generate past filenames, while positive values will generate filenames in the future. For example if you want to generate filenames from yesterday (24 hours back), use -84600 as offset.

Event Types to Log

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

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

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.

Note

If you intend to make the Event ID part of the actual Syslog message while forwarding to a Syslog server, adjust the Event Log Monitor output format or the Forward Syslog action message format accordingly.

The Event Log Monitor caches messages libraries. This greatly speeds up processing, but causes memory consumption for the cached libraries. By default, libraries are cached for 30 minutes. If memory consumption is too high, you may consider to lower the cache period. The cache is global to all event log monitors. As such, its size must be changed in the Engine specific Options Tab. Here you find the Resource Library Cache Options

Event Log Monitor V2

This dialog configures the Windows Event Log Monitor V2 service.

Eventlog Monitor V2

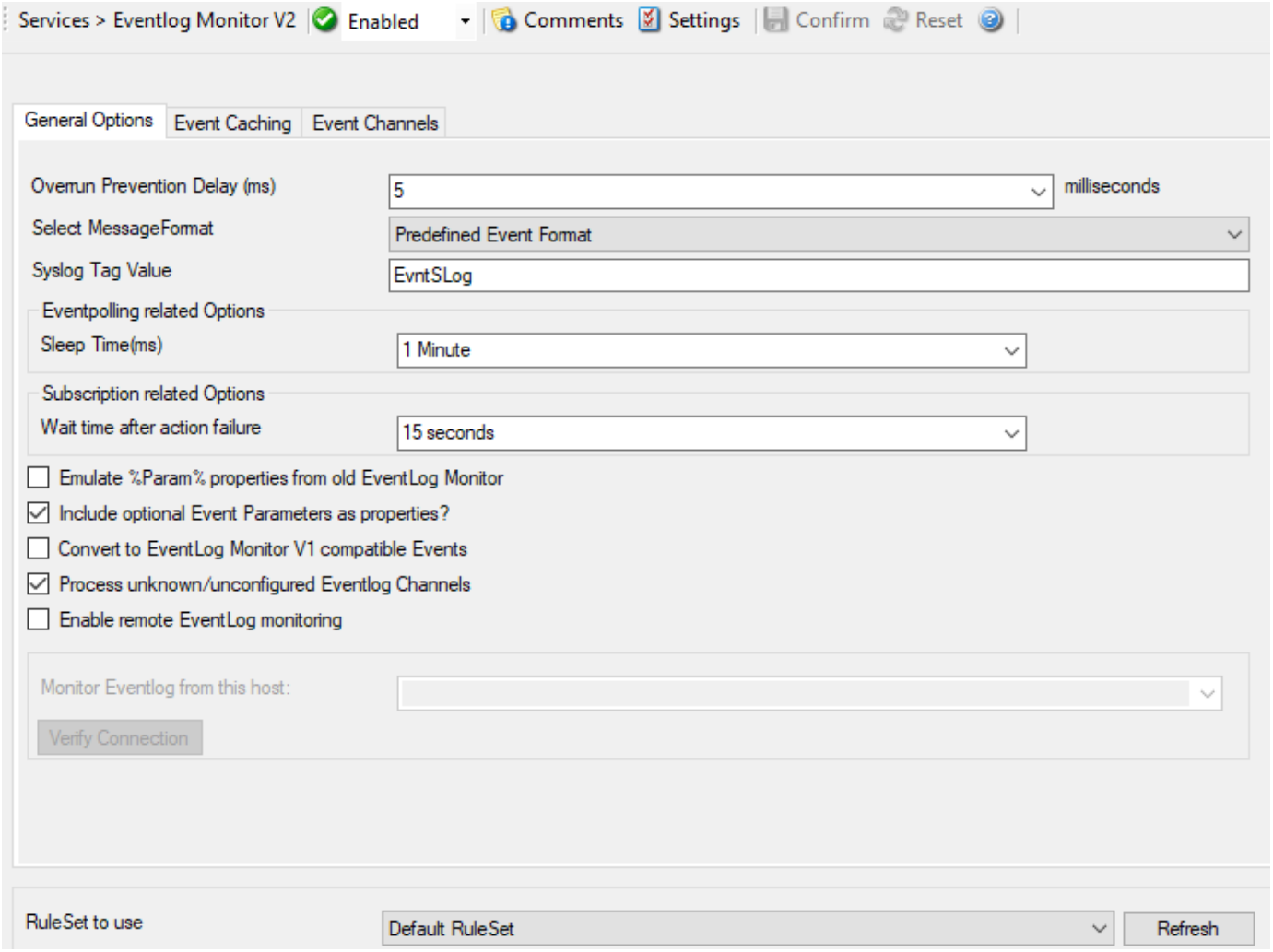
- all modern Windows versions (Windows 10, 11, Server 2016, 2019, 2022, and newer).

Eventlog Monitor

- Windows XP
- Windows 2003

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

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



- Service - Event Log Monitor V2 - General Options*

General Options Tab

Overrun Prevention Delay (ms)

File Configuration field:

nPreventOverrunDelay

Description:

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

If run at a value of zero, the Event Log Monitor service generates events as fast as the machine permits. We have seen scenarios where routers and receivers are not able to keep up with this rate, resulting in packet loss. In addition, the CPU of the reporting machine is run at 100% - which is not a problem because the service runs at a low priority. However, with even a 1-millisecond delay, there is no noticeable CPU activity even when large bursts of events are forwarded. At one millisecond, the service can still generate 1000 events per second.

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

Select Message Format

File Configuration field:

nFormatType

- 0 = XML Format
- 1 = Predefined EventFormat

Description:

With this option you can choose whether the Events will be extracted in “Raw XML Format” or in the “Predefined Event Format”.

The XML format is the exact representation of the XML Stream returned by the EventLog System. **Please note that it only contains EventLog data and not a formatted message.**

The “Predefined Event Format” is what the Event in the event viewer looks like.

Copy Format into Property

If enabled, a second message format can be stored into a custom property.

Select Message Format

File Configuration field:

nCopyFormatIntoProperty

Description

Select which message format should be stored into the custom property.

Store into Property

File Configuration field:

szCopyFormatIntoProperty

Description

The custom message property, for the “Copy Format into Property” Option.

SyslogTag Value

File Configuration field:

szSyslogTagValue

Description:

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

Eventpolling related Option: Sleep Time(ms)

File Configuration field:

nSleepTime

Description:

As said in the overview, this controls iteration over the configured channels. The value is specified in milliseconds.

For channels configured to use Polling Mode, the “Sleep Time” parameter specifies how often they are processed. Note that when multiple channels are set to polling mode, they are processed one after another. So there is a somewhat larger delay in processing than given by the “Sleep Time” parameter. The total frequency depends on how busy all polling channels are.

For channels configured to subscription mode, the “Sleep Time” interval will only influence how often a potential reset of “Last Record” is checked. Other than that, it has no effect on the event delivery rate.

We suggest a value of 60,000 milliseconds for the “Sleep Time”. With that setting, the Event Log Monitor checks for new events every 60 seconds. Larger periods can be specified for occasionally connected systems or if email delivery with few emails per day is intended.

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

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

Subscription related Option: Wait time after action failure

File Configuration field:

nSubscriptionSleepTime

Description:

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

Emulate %Param% properties from old EventLog Monitor

File Configuration field:

nEmulateParameters

Description:

This option emulates the %Param% properties, which were often used in the old EventLog Monitor. The new EventLog implementation does not support them in the same way anymore. The Event Log Monitor V2 is still able to provide parameters in the “old style” format, what means that log analysis scripts can receive a consistent stream of data for both new style and old style Windows events.

Include optional Event Parameters as properties?

File Configuration field:

nIncludeEventParameters

Description:

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

Convert to EventLog Monitor V1

File Configuration field:

nConvertToEventLogMonitorV1

Description:

This option maps the EventID's from the Security EventLog back to V1 (Windows 2000/2003). The internal InforUnitID is also changed to V1. This option helps postprocessing EventLog V1 and V2 events equally.

Process unknown/unconfigured Eventlog Channels

File Configuration field:

nEnableUnknownChannels

Description:

If enabled, unconfigured Eventlog Channels (Those found on the system, but not listed in the Eventlog Channels list) will automatically be processed.

Enable remote EventLog monitoring

File Configuration field:

nEnabledRemote

Description:

If enabled, EventLog Monitor will connect to a remote machine to process the EventLog. Please note that the RPC Service needs to be installed on the remote machine, and the Service has to be configured with a network user that has sufficient access rights.

Monitor Eventlog from this host

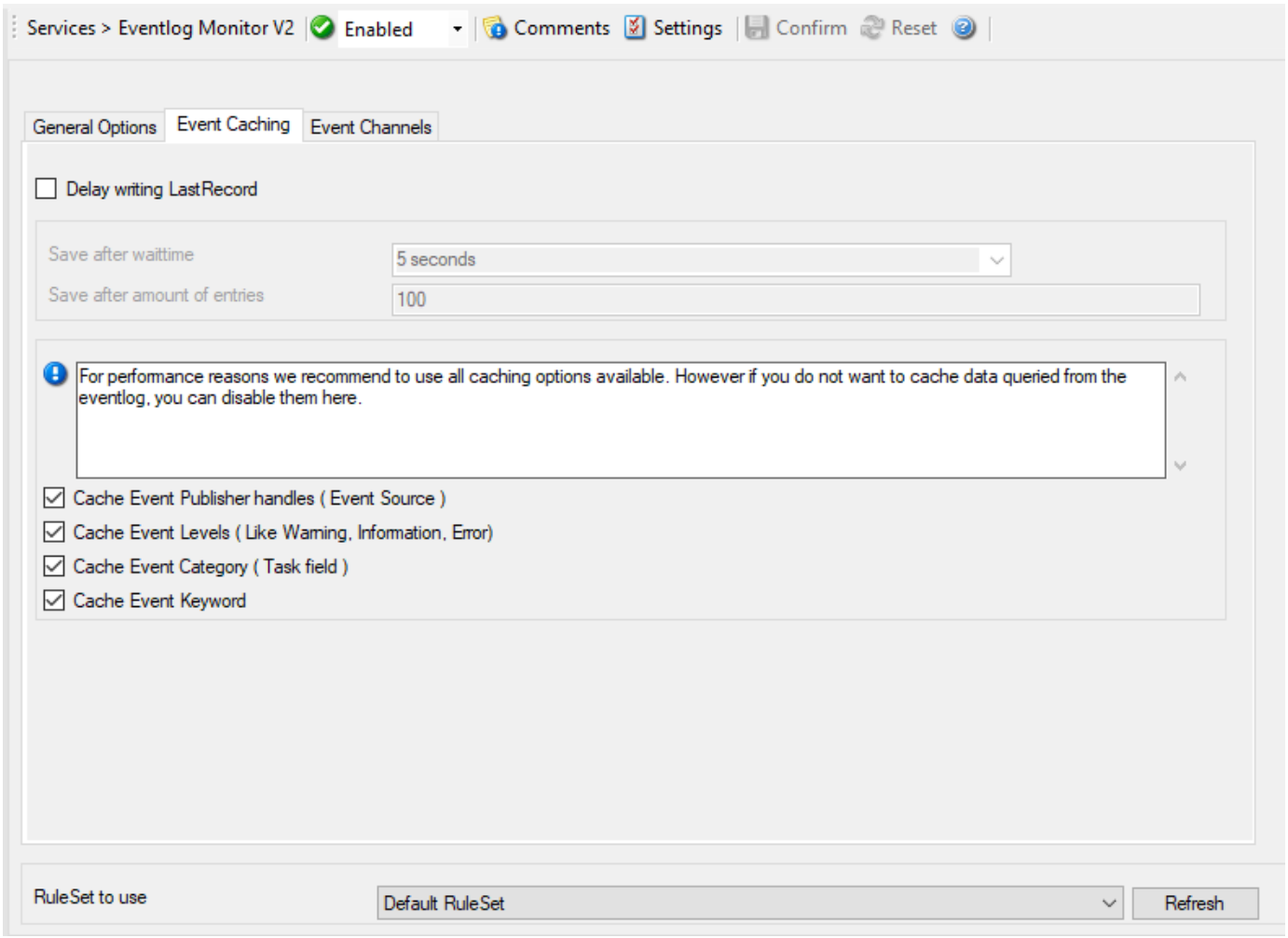
File Configuration field:

szServerName

Description:

The hostname of the remote server to be monitored by EventLog Monitor.

Event Caching Tab



- Service - Event Log Monitor V2 - Event Caching*

Delay writing LastRecord

File Configuration field:

nEnableLastRecordDelay

Description:

Enables the LastRecord writeback delay to the configured properties below.

Save after waittime

File Configuration field:

nLastRecordDelayTime

Description:

Regardless of the amount of processed event log entries, the lastrecord value will delayed for this waittime period.

Save after amount of entries

File Configuration field:

nLastRecordDelayCount

Description:

Regardless of the configured waittime period, the LastRecord will be written after the amount of processed event log entries that are specified here.

Cache Event Publisher handles (Event Source)

File Configuration field:

nCacheEventPublisher

Description:

All publisher sources will be kept open once loaded until the application / service is stopped. This increases processing speed for events from same sources.

Cache Event Levels (Like Warning, Information, Error)

File Configuration field:

nCacheEventLevel

Description:

If enabled the textual representations for event levels will be cached.

Cache Event Category (Task field)

File Configuration field:

nCacheEventCategory

Description:

If enabled, all textual representations for event categories will be cached.

Cache Event Keyword

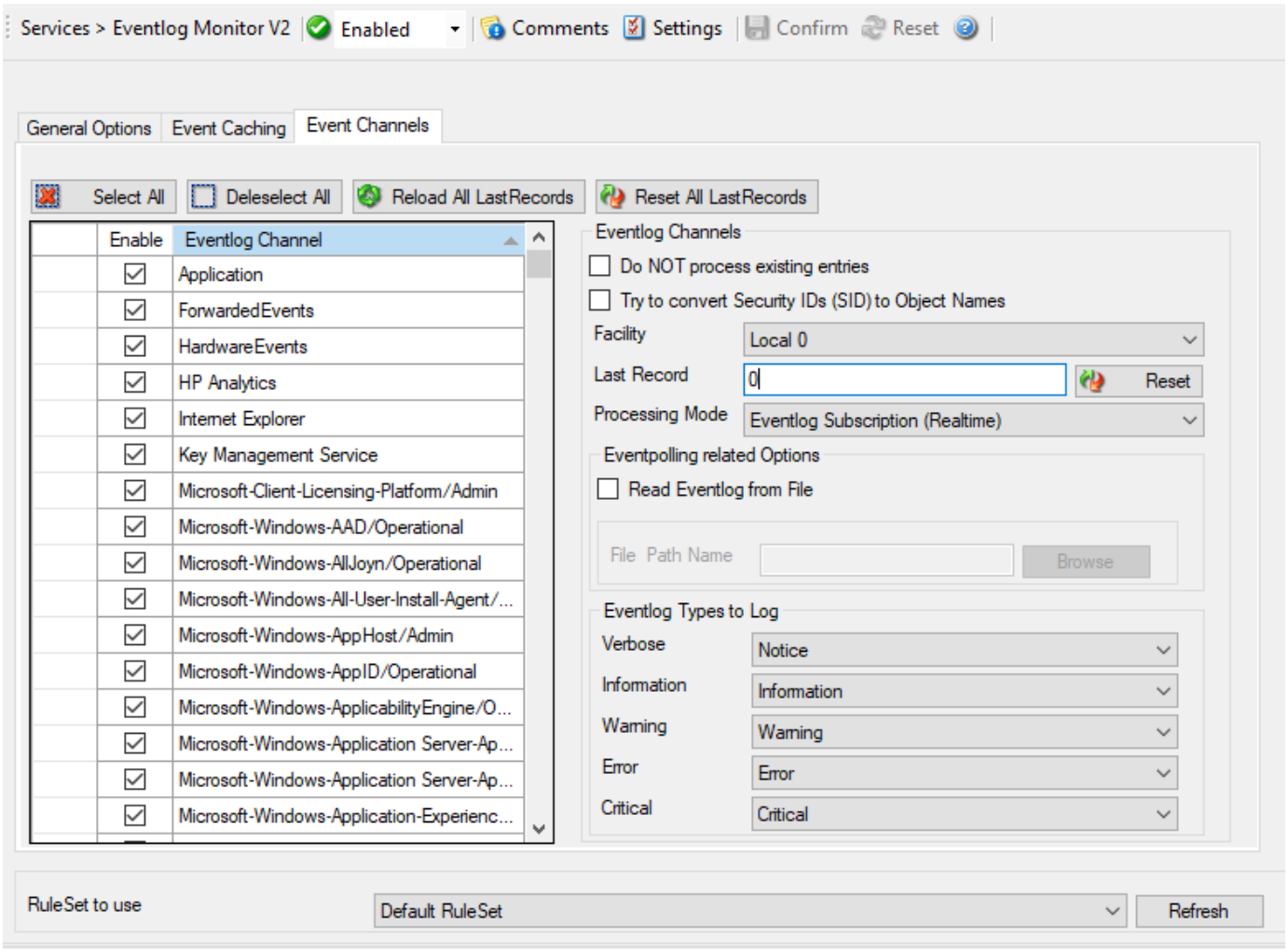
File Configuration field:

nCacheEventKeyword

Description:

If enabled, all textual representations for event keywords will be cached.

Event Channels Tab



- Service - Event Log Monitor V2 Event Channels*

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

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

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

Do NOT process existing entries

File Configuration field:

nNoExistingEntries

Description:

If you do not want to get a dump of an existing specific Windows Event Log then use this option. When MonitorWare Agent / EventReporter are restarted it will start processing after that last entry and do not look for the previous entries.

Try to convert Security IDs (SID) to Object Names

File Configuration field:

nTryConvSIDtoObj

Description:

With this option you can convert Security ID's (SIDs) to object names. You can enable this feature in the advanced configuration of each event log type in the Event Log Monitor service. Simple check the "Try to convert Security IDs (SID) to Object Names" option.

Note that only the Security event log has this feature enabled by default. For all other event log types this feature is disabled by default.

Facility

File Configuration field:

nFacility

Description:

The syslog facility to map information units stemming from this log to. Most useful if the message is to forward to a Syslog daemon.

Last Record

File Configuration field:

szXMLBookmark

Description:

Windows Event Log records are numbered serially, starting at one. The service records the last record processed. This textbox allows you to override this value. **Use it with caution!**

If you would like a complete dump of a specific Windows Event Log, reset the "Last Record" to zero. If you missed some events, simply reset it to some lower value than currently set. It is possible to set "Last Record" to a higher value. This suspends event reporting until that record has been created. We strongly discourage to use this feature unless definitely needed.

Processing Mode

File Configuration field:

nApiReadMode

- 0 = Subscription Readmode (Real-time)
- 1 = Polling Readmode (Sleeptime)

Description:

There are two processing modes available, first the default processing mode is "EventLog Subscription" which processes Events in real time. This means events are send to MonitorWare Agent by the OS as they happen, there is no delay at all. The other processing mode called "Eventlog Polling" and is similar to the method used in the old EventLog Monitor. The EventLog is checked and processed periodically controlled by the sleeptime. However using the polling method, you enable the "Read EventLog From File" option.

Eventpolling related Options

Read Eventlog from File

File Configuration field:

nReadFromFile

Description:

When enabled, the Eventlog is read from a file instead from the system.

File Path Name

File Configuration field:

szLogFileName

Description:

It defines which is file to be read, only available when “Read Eventlog From File” is enabled.

Event Types to Log

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

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

RuleSet to use

File Configuration field:

szRuleSetName

Description:

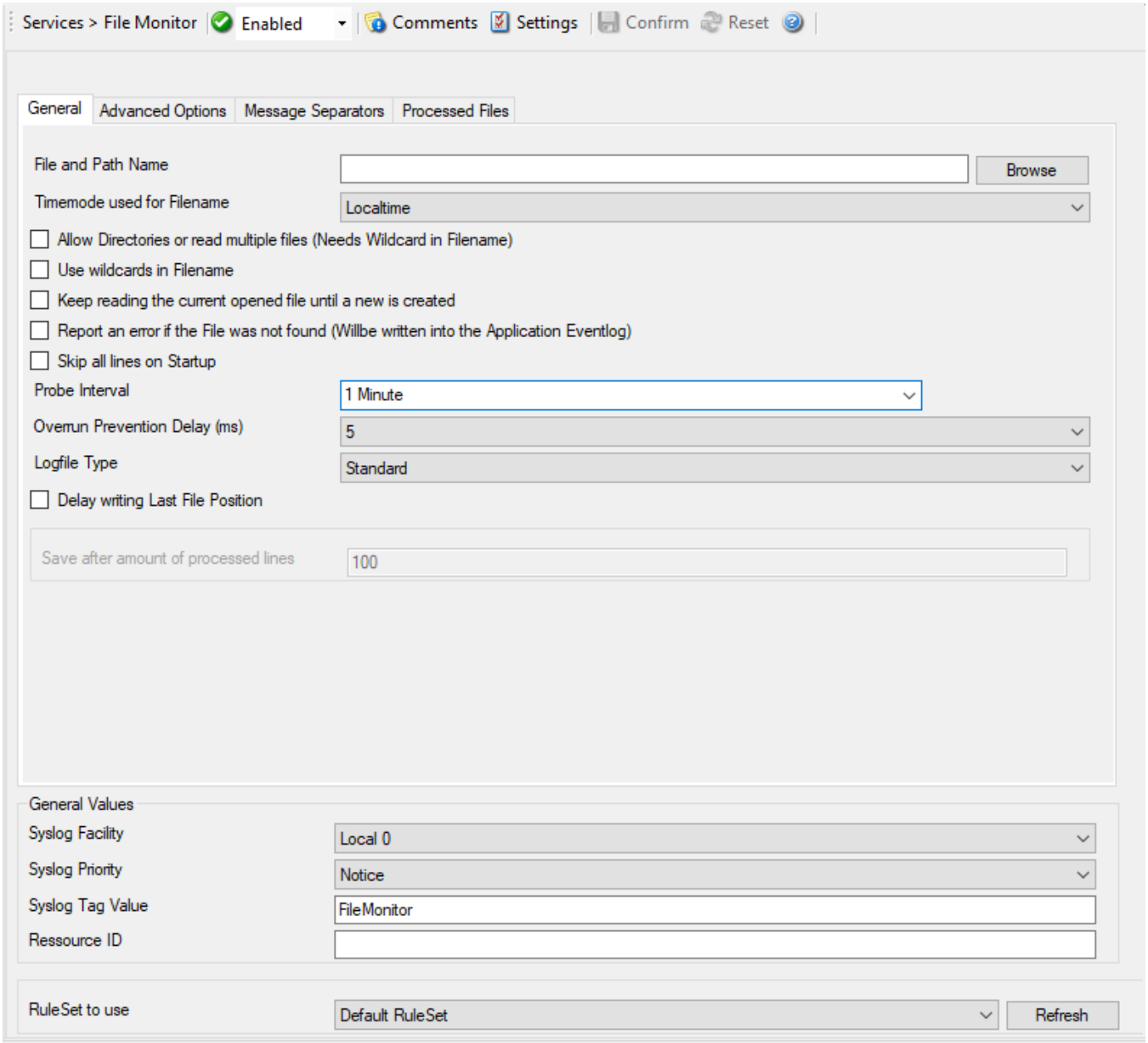
Name of the ruleset to be used for this service. The RuleSet name must be a valid RuleSet.

File Monitor

The file monitor monitors the content of a text file just as the event monitor monitors the Windows Event Log. Its purpose is to gather vital information that is stored in system text files. Many applications do not write events to the event log but to a text file. This is also the case with many Microsoft applications (for example the WINS log).

The file monitor can also gather Internet Information Server (Windows' web server) log files. This is very useful for monitoring web activity and detecting attacks.

General Tab



- Service - File Monitor General*

File and Path name

File Configuration field:

szPathAndFileName

Description:

Here, you must type the name of the file to be monitored. To select a file from any specified location, press the browse button. Once a complete file name is specified, exactly that file is monitored.

The file name is never changed automatically. However, many systems generate changing log files. For example, Internet Information Server (IIS) can be configured to change the log file every day. Therefore, each day's log file has a different name.

To support changing log file names, there are replacement characters available within the file name. These are:

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

Please note: the replacement characters are case sensitive!

For example, daily Internet Information Server log files are named "exyymmdd.log", with yy being the 2 digit year, mm the month, and dd the day of month. To generate the same name with file monitor, use the following name "ex%y%m%d.log".

Please note that there is no replacement character for the monthly week number (1st week, 2nd week). As such, the weekly log file setting of IIS is not supported.

On network paths: The File Monitor can also work on network storages. There are two ways of monitoring files in a network path.

1. Direct the service 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 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.

TimeMode used for Filename

File Configuration field:

nTimeMode

Description:

Select the time mode used for the log file to be monitored with this drop-down list. Available options are:

1. Local Time: log file is monitored based on local time.
2. utc: log file is monitored based on universal coordinated time. UTC was formerly referred to as "GMT" (Greenwich Mean Time) and is the basis of the international time zone system.

Allow Directories or Read Multiple Files

This capability is available starting with **26.08**.

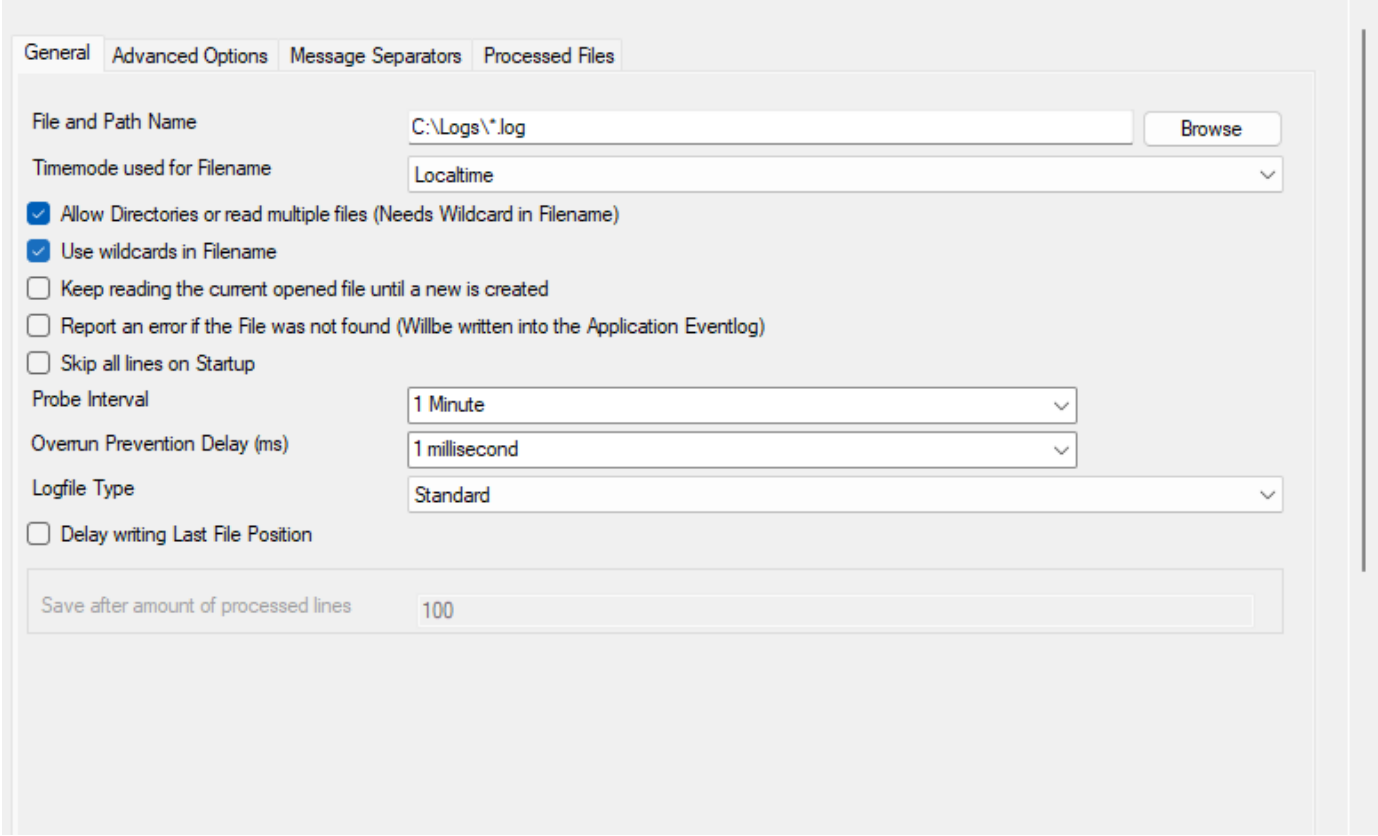
File Configuration field:

nMultipleFiles

Description:

This option reads a set of matching files. It supports wildcards in directory names and in the filename. For example, `C:\logs\tenant*\app\*.log` and `C:\logs\foo*\bar*\*.log` first expand matching directories, then apply the filename pattern inside them. Each matching file keeps separate read state.

Use this option for multiple files or wildcard directories. Directory matches are recursive only; similarly named files are not treated as directories.



File Monitor configured for multiple files and filename wildcards.

Use wildcards in Filename

File Configuration field:

nUseWildcard

Description:

This option allows you use `*` as a random character sequence in the filename for a single-file configuration.

Please note: with this single-file option, wildcards are limited to the filename, not the file path. To use wildcard directories, enable **Allow Directories or Read Multiple Files** instead.

Keep reading the current opened file until a new is created

File Configuration field:

nKeepReadingUntilNew

Description:

This has been added to define if the Service shall continuously read an open logfile until a new file (depending on the configured filename) is available. This Options is helpful for such cases where you do not know when a new logfile is generated and the old one is closed.

Report an error if the File was not found (Will be written into the Application Eventlog)

File Configuration field:

nReportErrorOnFNF

Description:

As the name says, if this setting is enabled, an error is reported in the Windows Eventlog if the file was not found.

Skip all lines on Startup

File Configuration field:

nSkipLinesEnabled

Description:

If this option is enabled, the File Monitor will skip all new lines of a logfile during startup. This will work in single-file mode as well as multi-file mode.

Probe Interval

File Configuration field:

nSleepTime

Description:

This interval is in milliseconds. After the specified interval the file monitor checks the file for new records.

We recommend a value of 60000 milliseconds for the “Check Interval”. With that setting, the file monitor checks for new records every 60 seconds. Larger periods can be specified for occasionally connected systems or if email delivery with few emails per day is intended.

Very security-aware environments might use a shorter interval. The MonitorWare Agent 3.0 is specifically designed to limit the burden on the monitored system. As such, resource usage is typically low, even with frequently run file monitor checks. However, we recommend not running the file monitor more often than once a second.

Overrun Prevention Delay (ms)

File Configuration field:

nPreventOverrunDelay

Description:

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

If run at a value of zero, the Service will process events as fast as the machine permits. We have seen scenarios where routers and receivers are not able to keep up with this rate, resulting in packet loss. In addition, the CPU of the reporting machine may be higher which is not a problem for most common multi-core machines in these days. However, with even a 1-millisecond delay, there is no noticeable CPU activity even when large bursts of events are forwarded. At one millisecond, the Service can still generate about 500- 1000 events per second.

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

If you need real-time event reporting, we recommend to set the value to 0.

Logfile Type

File Configuration field:

nLogFileType

- 0 = Standard (Default)
- 1 = W3CLog

Description:

Select the type of the log file to be monitored with this drop-down list. Available options are:

- Standard - a standard text log file
- W3C Web Server logfile - log files in the W3C web server compliant format.

Delay writing Last File Position

File Configuration field:

nEnableLastRecordDelay

Description:

Enables a writeback delay for the Last File Position. When processing large logfiles, we would recommend to enable this feature.

Save after amount of processed lines

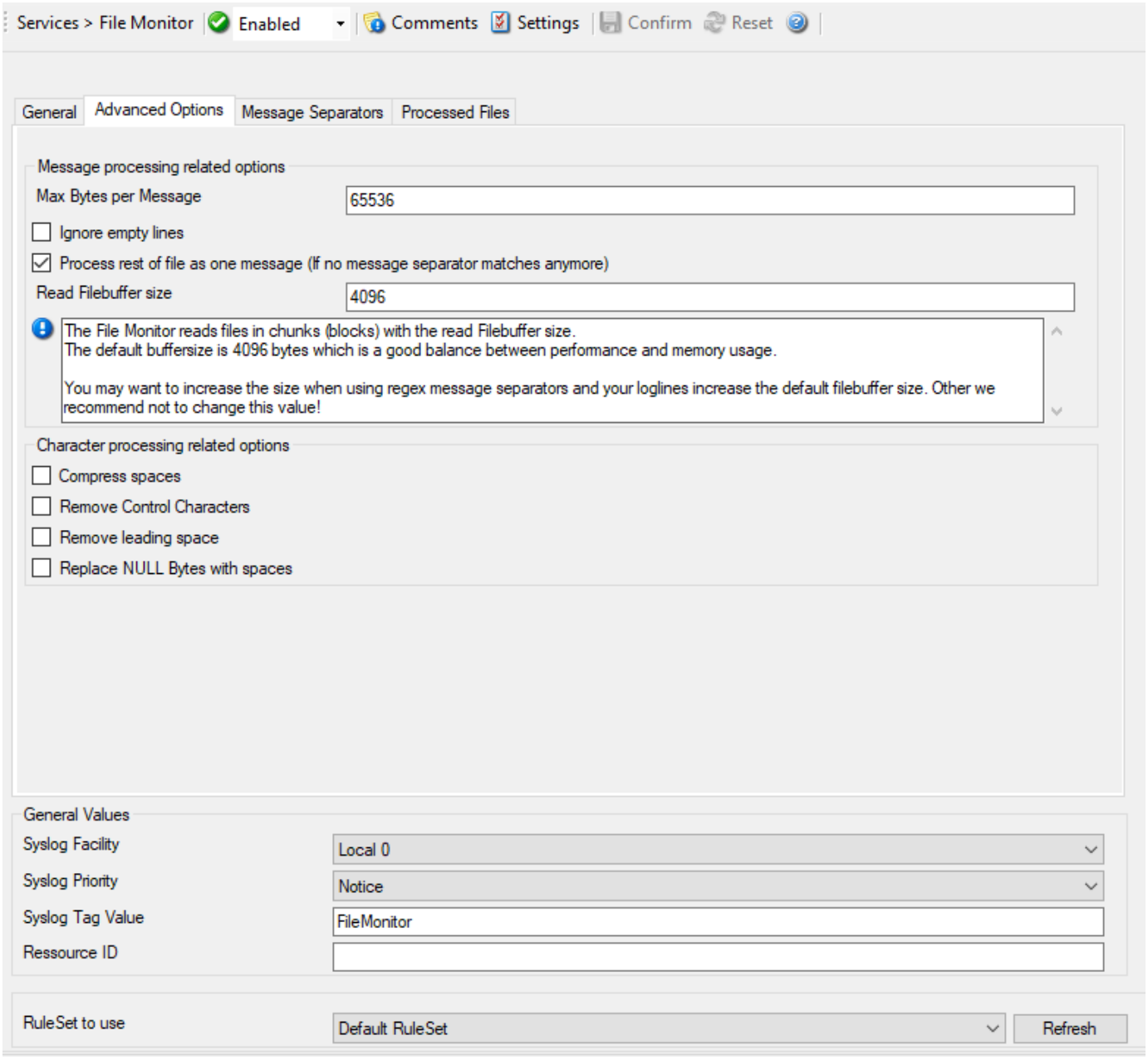
File Configuration field:

nLastRecordDelayCount

Description:

When Delay writing Last File Position is enabled, the Last File Position will be written after the amount of processed lines that are specified here.

Advanced Options Tab



- Service - File Monitor Advanced Options*

Message processing related options

Max Bytes per Message

File Configuration field:

nMaxBytesPerMessage

Description:

Maximum value of bytes that the file monitor reads per line. If a message is larger then this value, the message splits into multiple parts.

Ignore empty lines

File Configuration field:

nIgnoreEmptyLines

Description:

As the name already says, this option discards empty lines within the logfile.

Process rest of file as one message (If no message separator matches anymore)

File Configuration field:

nProcessRestOfFile

Description:

When enabled, file monitor will consider the rest of a file as single message if no other message separator matches anymore.

Read Filebuffer size

File Configuration field:

nReadFileBuffer

Description:

File Monitor reads files in chunks (blocks) with the read Filebuffer size. You can define the size of these chunks here, the default buffersize is 4096 which is a good balance between performance and memory usage. Please note that when using regex message separators, you may need to increase the Filebuffer size to the maximum possible length of a single logline. Other we recommend not to change this value!

Character processing related options

Compress spaces

File Configuration field:

nCompressSpaces

Description:

This option compresses sequences of spaces found inside the message to a single one.

Remove Control Characters

File Configuration field:

nRemoveControlChars

Description:

Removes control characters like CR and LF(carriage return and line feed).

Remove leading space

File Configuration field:

nRemoveLeadingSpaces

Description:

If there are any leading spaces in the file, this option removes them.

Replace NULL Bytes with spaces

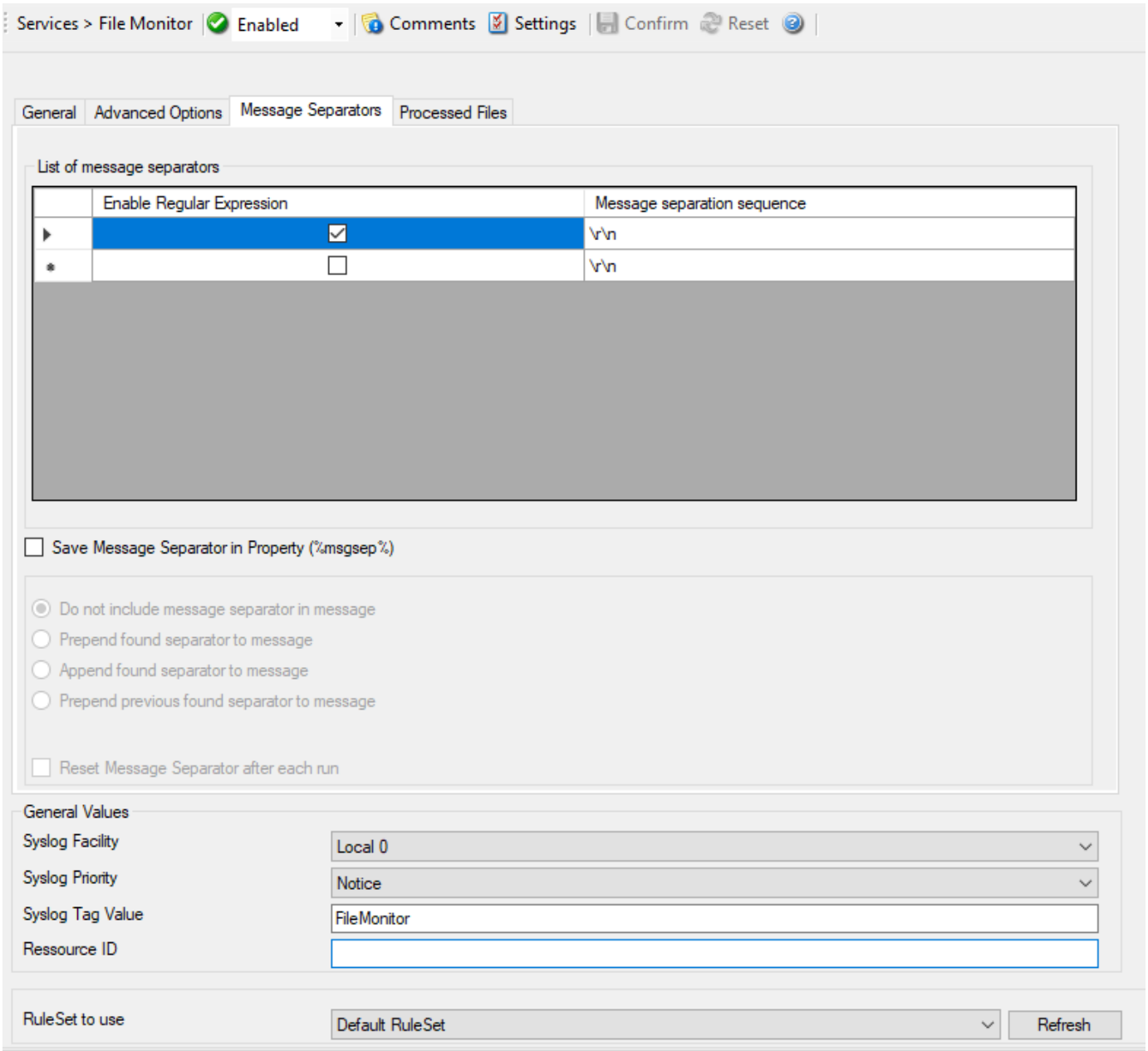
File Configuration field:

nReplaceNullBytes

Description:

If this option is enabled, the FileMonitor will replace NULL Bytes within files with spaces. These spaces again can be compressed with the “Compress spaces” options.

Message Separators Tab



- Service - File Monitor Message Separators*

Enable Regular Expression

File Configuration field:

bIsRegex_[n]

Description:

When enabled, the Message separation sequence will be handled as regular expression. Everything known in the regular expression syntax can be used to define a matching pattern.

Here are some regular expressions samples:

Regular Expression: `[0-9]{4,4}-[0-9]{1,2}-[0-9]{1,2} [0-9]{1,2}:[0-9]{1,2}:[0-9]{1,2}` Matches typical Date like `2015-11-20 12:11:01`

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

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

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

Message separation sequence

File Configuration field:

szMsgSep_[n]

Description:

Defines a customizable separation sequence by which the file monitor splits messages.

If using multiple separation sequences, the comparison operation will be held as an OR operation. That means, that either this value or another value has to be true, so a message can be split. This is especially important for logfiles with different log formats.

To date, the following control characters are available:

- r -> carriage return
- n -> line feed

Save Message Separator in Property (%msgsep%)

If this option is enabled, the current and last used message separator will be saved into the properties `%msgsep%` and `%msgseplast%`.

When using regular expression as message separators, it might be needed to append or prepend the message separator to the message itself. For these cases, we have multiple way to do this:

Do not include message separator in message

File Configuration field:

nPropertyMsgSep = 0

Description:

Message Separator will not be included in the message

Prepend found separator to message

File Configuration field:

nPropertyMsgSep = 1

Description:

When enabled, the current found message separator will be prepended to the message.

Append found separator to message

File Configuration field:

nPropertyMsgSep = 2

Description:

When enabled, the current found message separator will be appended to the message.

Prepend previous found separator to message

File Configuration field:

nPropertyMsgSep = 3

Description:

When enabled, the previous found message separator will be prepended to the message.

Reset Message Separator after each run

File Configuration field:

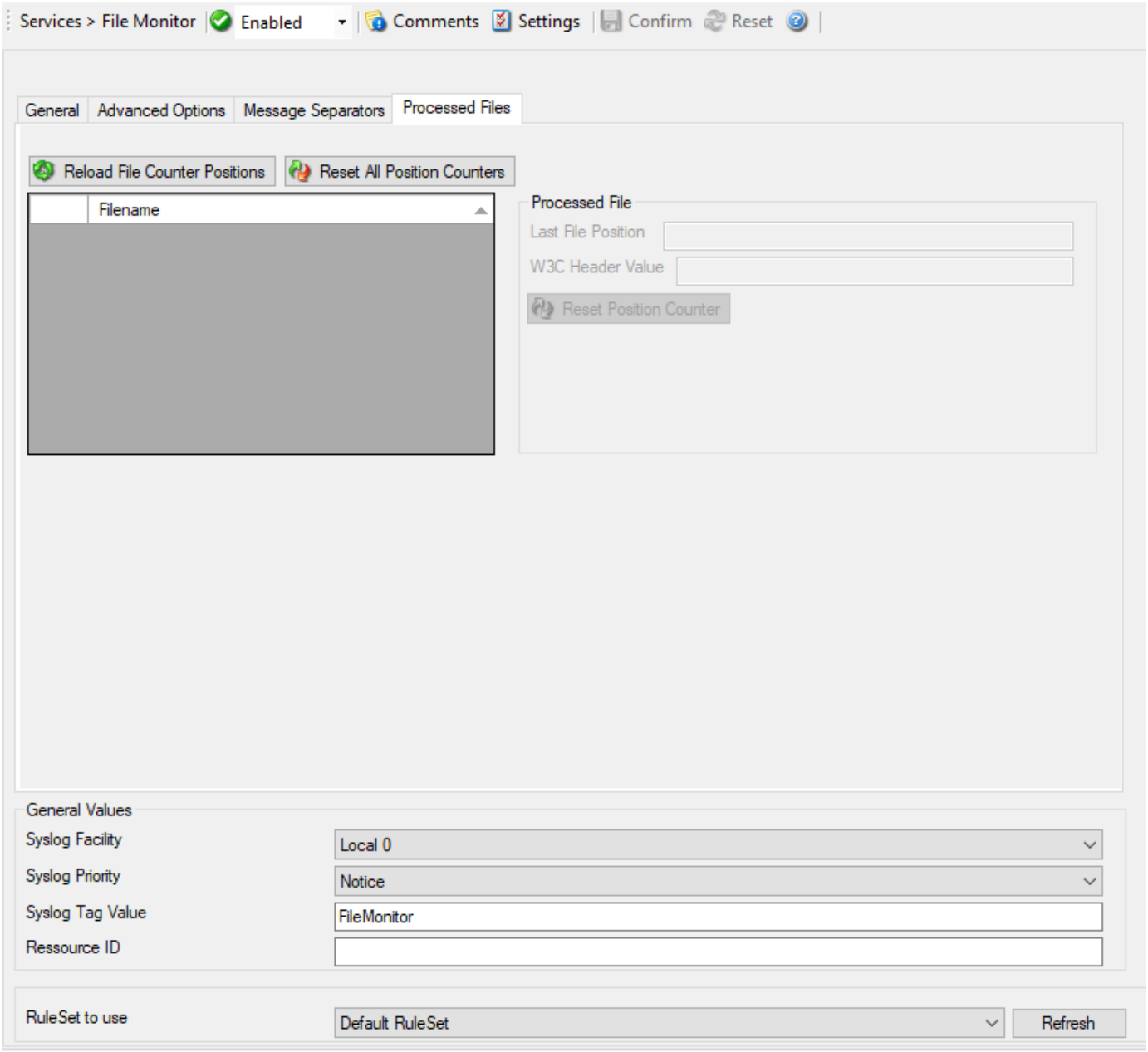
nPropertyMsgSepReset

Description:

If enabled, the message separator values will be reset after the File Monitor has finished a run (reached the end of a file).

Processed Files Tab

On this tab you see all files in a data grid that file monitor has yet processed. Select a file within this grid to see its data values like the last file position.



- Service - File Monitor Processed Files*

Last File Position

File Configuration field:

nLastPos [Data Value]

Description: This is the last file (Byte) position that the File Monitor processed. Lower or reset the Position Counter if you want to reprocess the file partially or even fully.

The data values can be controlled using the Reload File Counter Positions and Reset All Position Counters buttons, or simply by editing the Last File Position field. When saving applied changes to the data values, the service will reload them during next check interval and reprocess files if necessary.

W3C Header Value

File Configuration field:

szW3CHeader [Data Value]

Description:

If logfile type is set to “W3C Webserver Logfile”, this value will be filled with a list of field types taken from the W3C Field header.

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.

Filter Conditions

Filter conditions specify **when** a rule should run. If the filter condition evaluates to true, the actions in that rule are executed.

In practice, filters let you decide which Windows events should be forwarded, which should be routed differently, and which should be ignored.

How to use them well

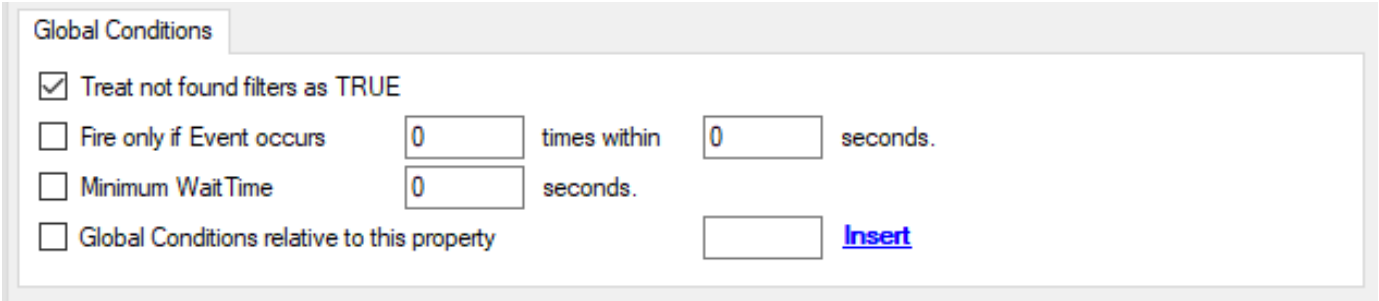
- Start with broad filters so the event path is easy to verify.

- Narrow down with event source, event ID, severity, or log name.
- Add message-content filters only after the basic event path works.
- Remember that rule order matters just as much as the filters themselves.

Detailed filter references

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

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.

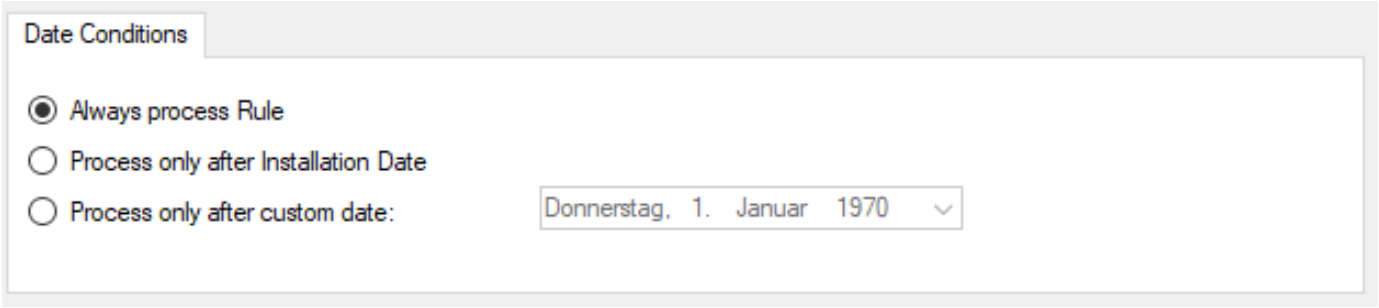
Global Conditions relative to this property

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

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

number:

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

boolean:

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

time:

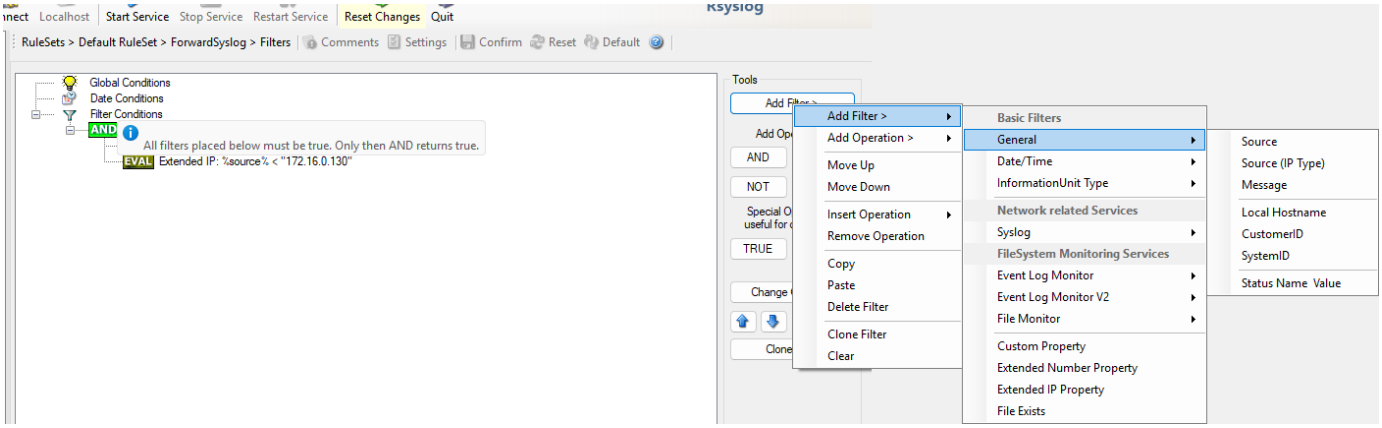
can be compared with another time but only with “=”

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.



rsyslog Windows Agent 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 is contains a valid IP Address or hostname. The IP Filter can filter against hostnames and IP Addresses, hostnames are automatically resolved using the internal DNSCache (for obvious performance reasons).

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

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

Message Content

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

The content search can be limited to a region within the message. To do so, select a starting and ending position within the string by choosing the **“contains within range”** compare 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).

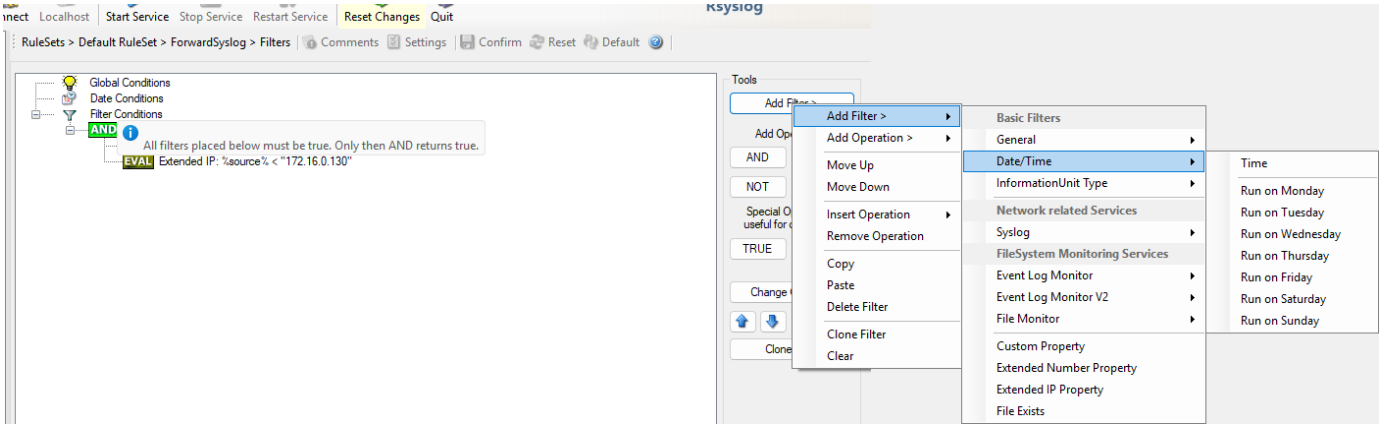
Status Name and Value

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.



rsyslog Windows Agent 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).

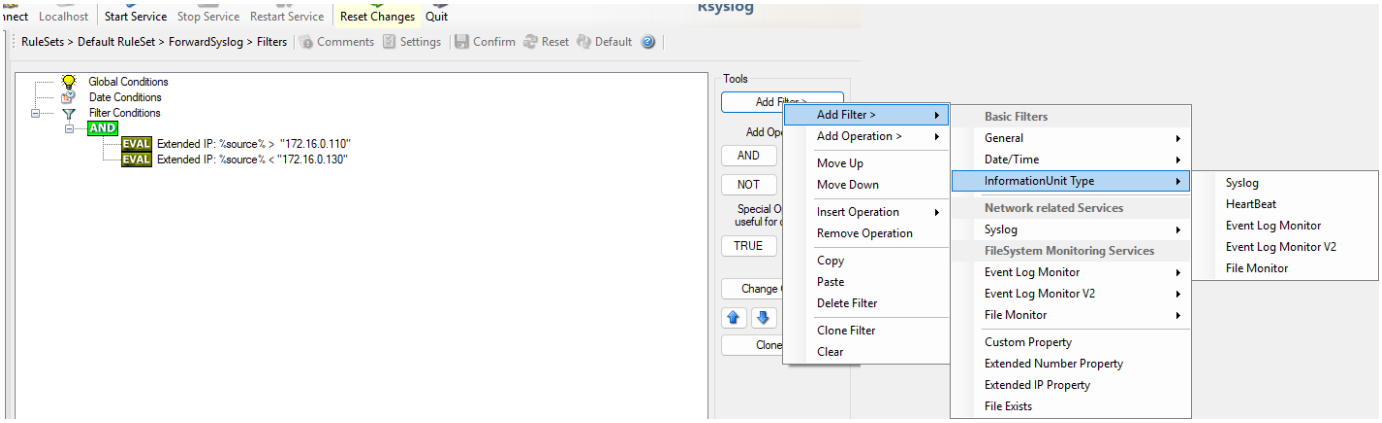
Weekdays

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).



rsyslog Windows Agent Filter Conditions - InformationUnit Type

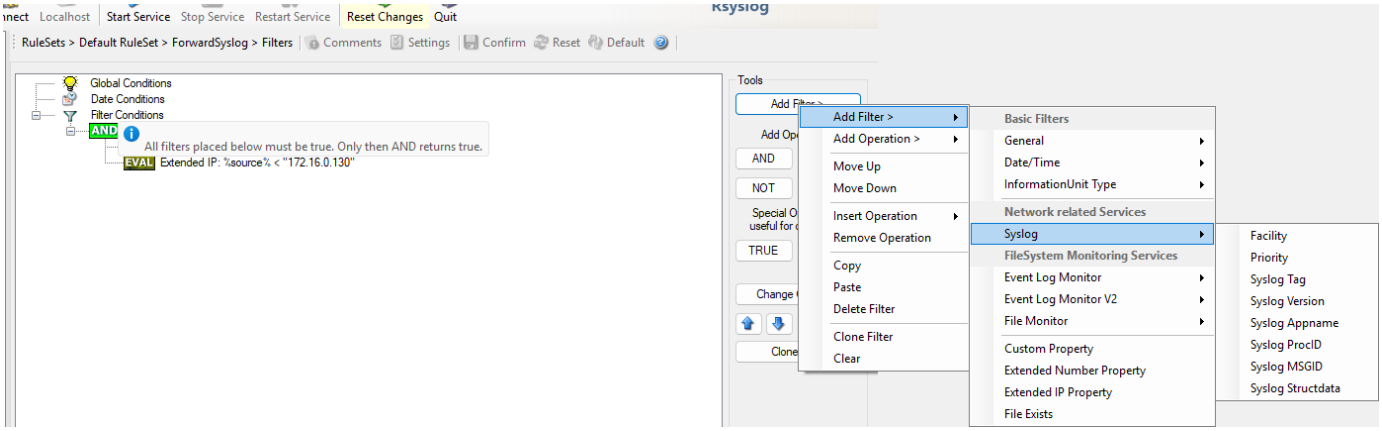
The following filters are available:

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.



rsyslog Windows Agent 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.

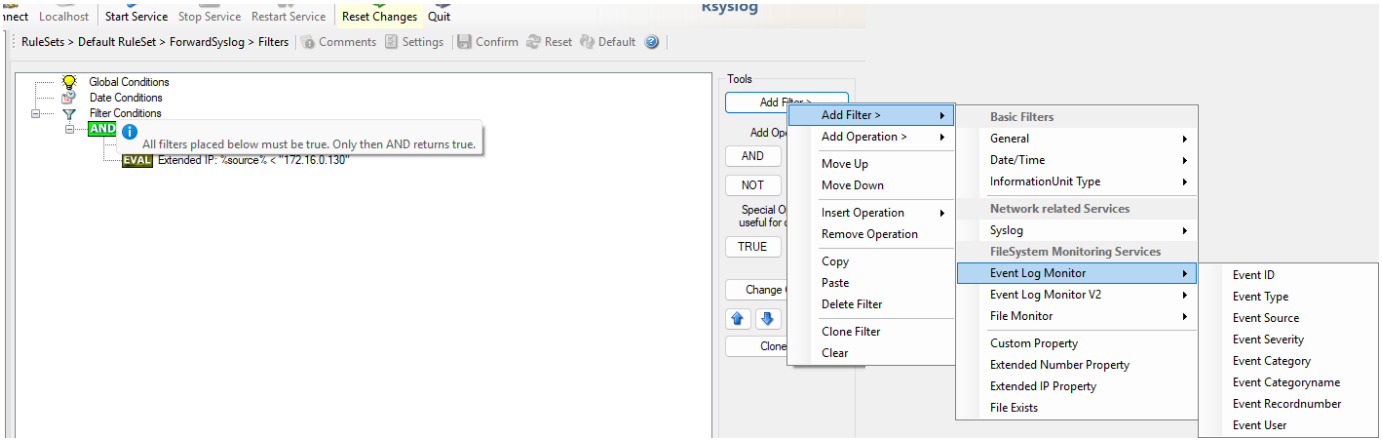
Syslog Tag

This filter is of type string.

Windows and file monitoring filters

Event Log Monitor

Event Log Monitor specific filters are grouped here.



rsyslog Windows Agent 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

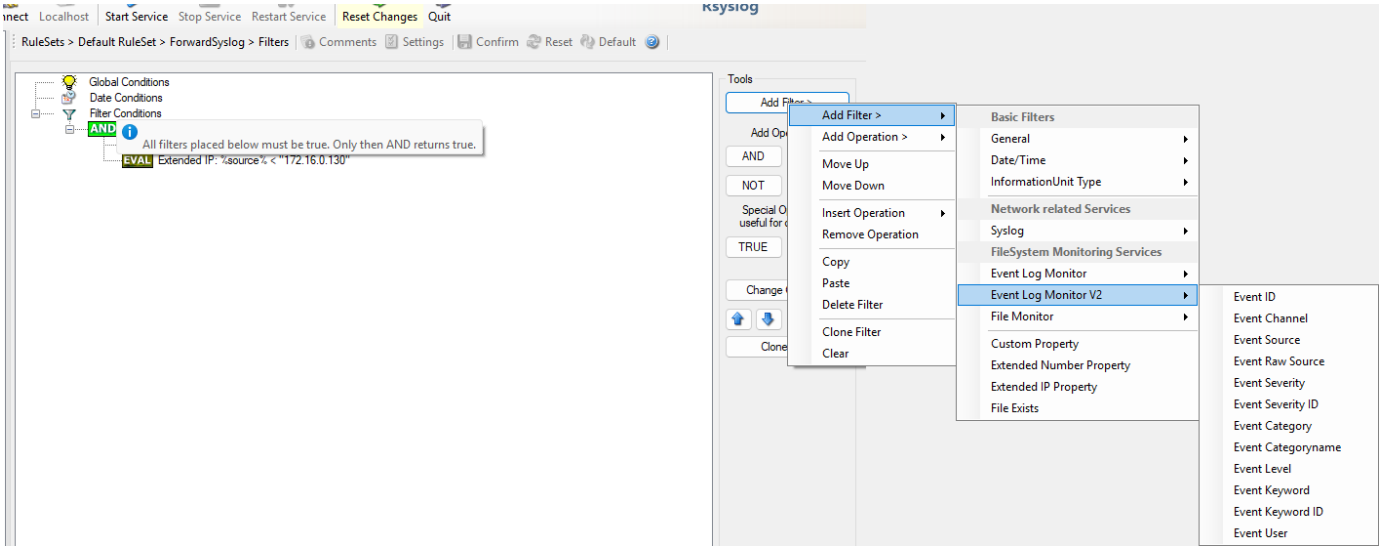
This is the event log user as specified in the Windows Event Log. If enabled, the event must have the configured event user or the rule will not match. Since it is a string value there must be an exact match. Please note that this value is case-sensitive.

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.



rsyslog Windows Agent 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

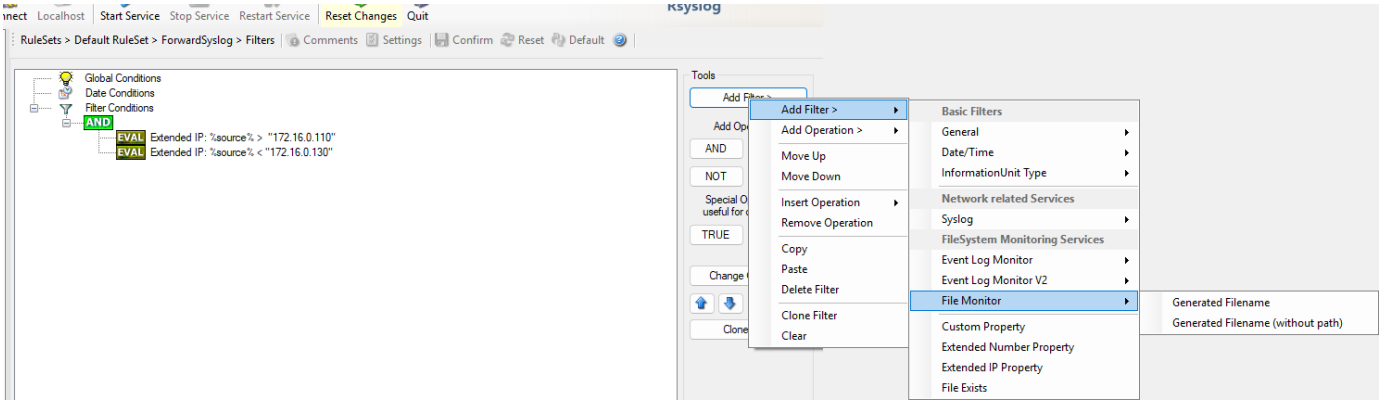
This is the internal keyword ID as string. If enabled, the event must have the configured event keyword ID or the rule will not match. There must be an exact match. Please note that this value is case-sensitive.

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.



rsyslog Windows Agent Filter Conditions - File Monitor

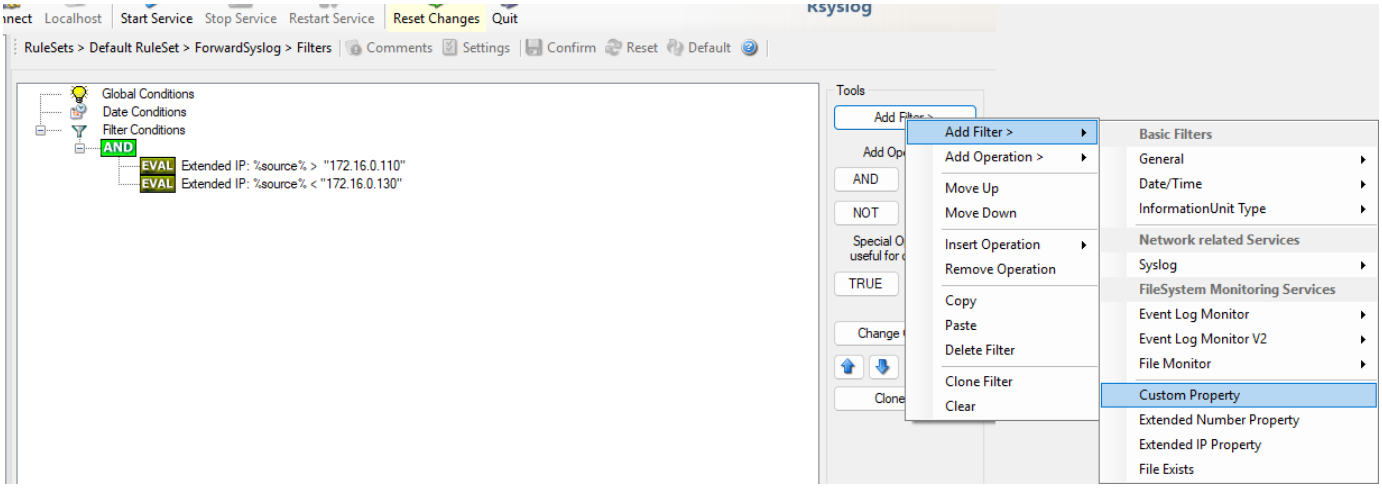
Generated Filename

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.



rsyslog Windows Agent Filter Conditions - Custom Property

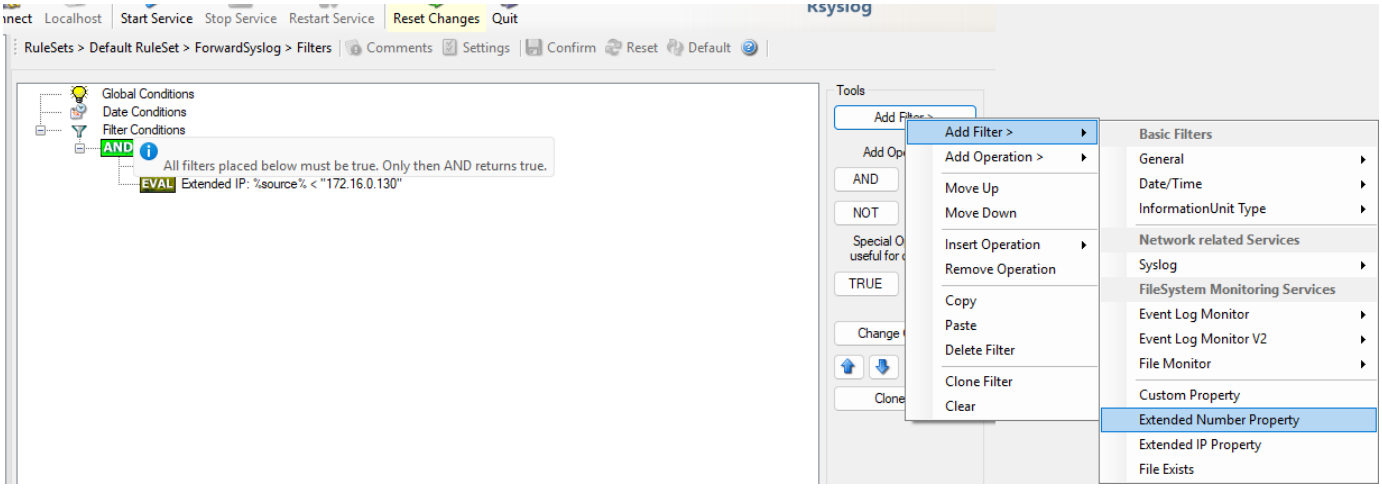
Custom Property

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.



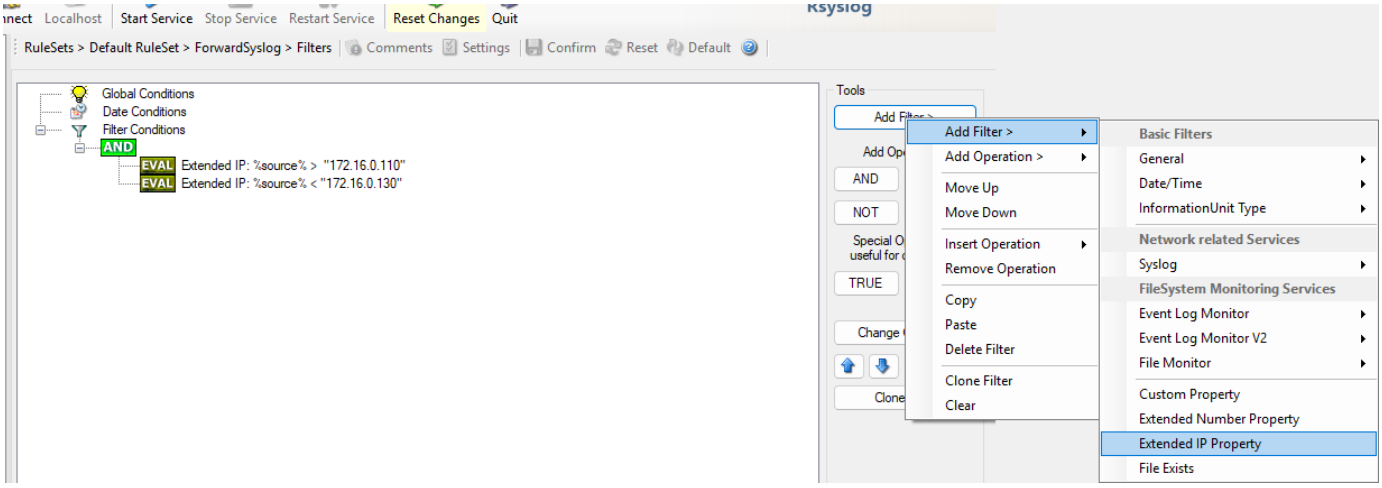
rsyslog Windows Agent Filter Conditions - Extended Number Property

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



rsyslog Windows Agent Filter Conditions - Extended IP Property

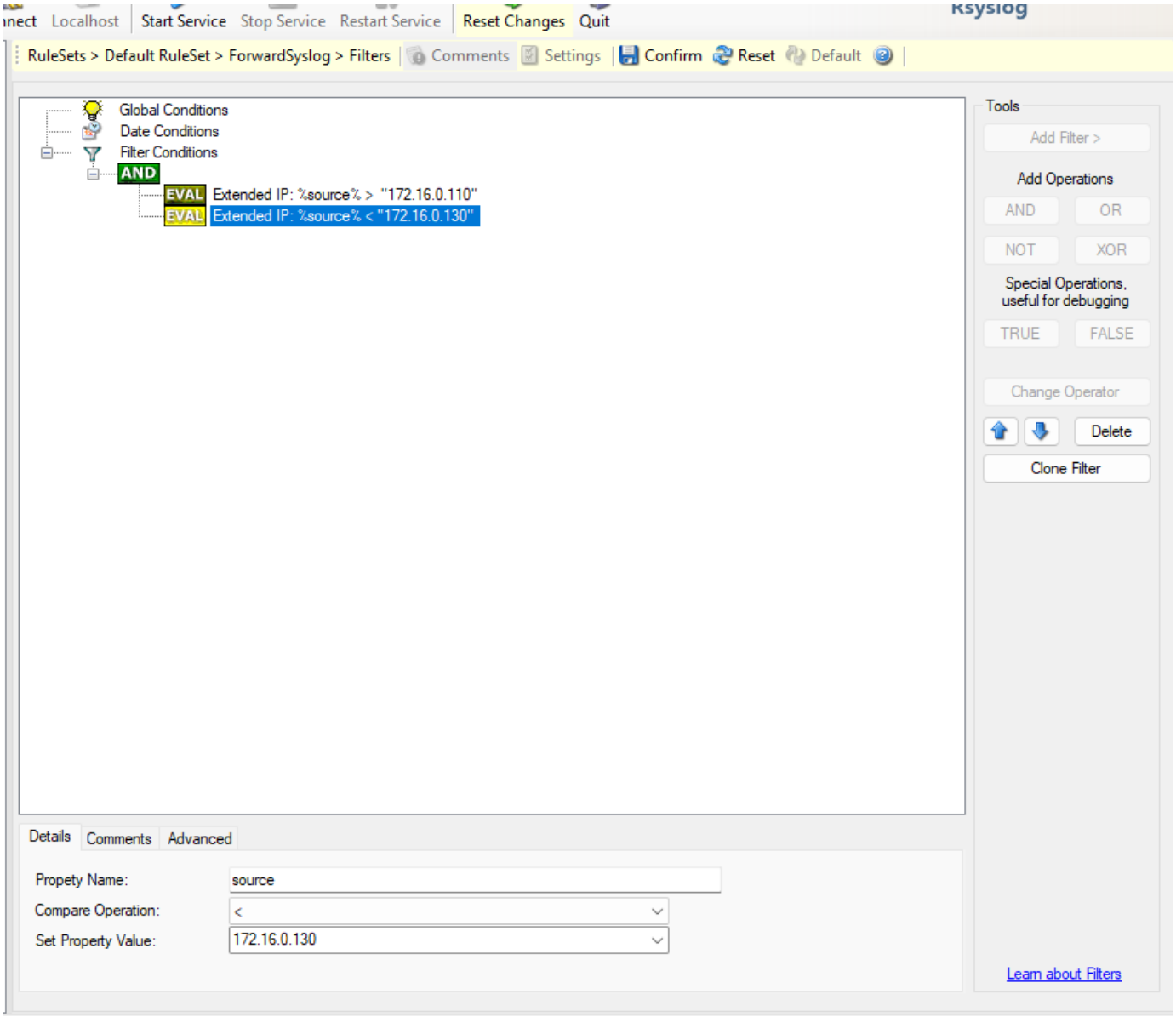
Extended IP Property filter settings

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.

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:

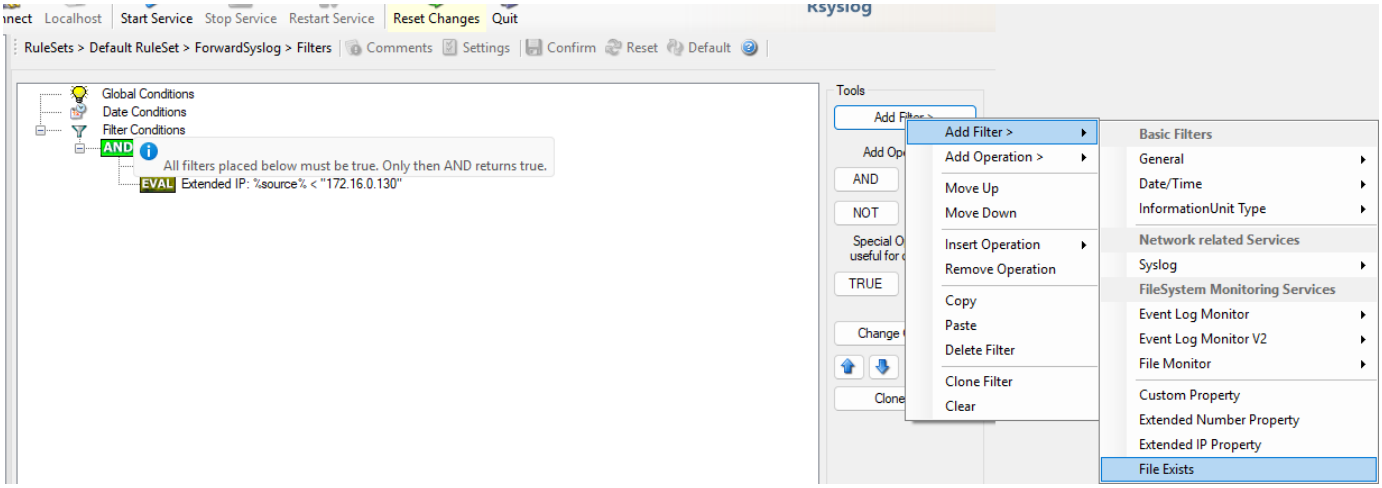


rsyslog Windows Agent Filter Conditions - Filtering for an IP Range

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.



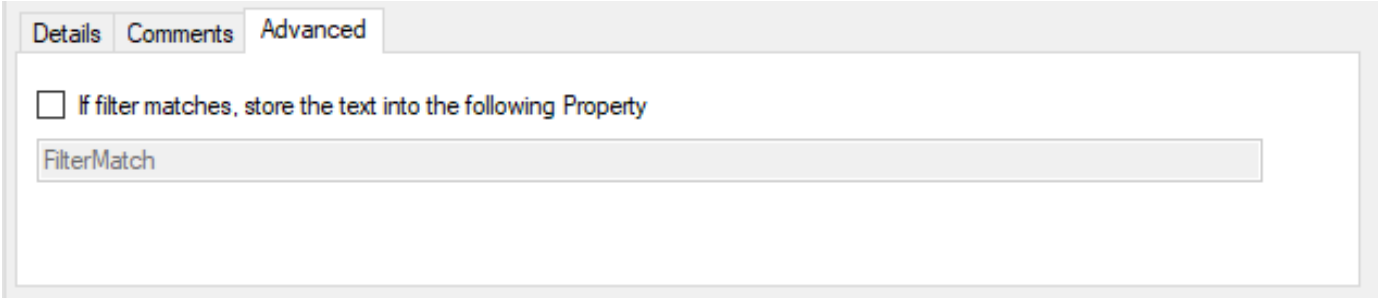
rsyslog Windows Agent Filter Conditions - File Exists

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

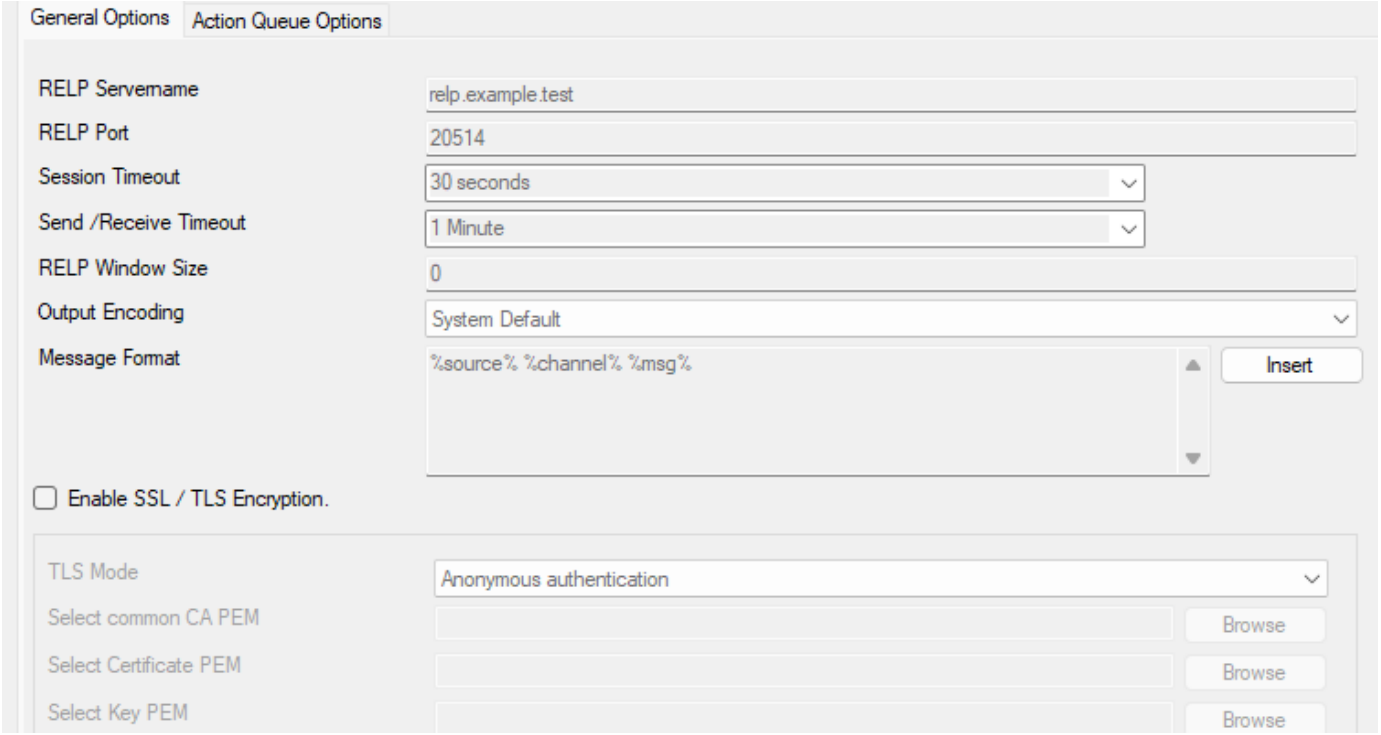
Actions define what rsyslog Windows Agent does with an event after it matches a rule.
There can be multiple actions for each rule. Actions are processed in the order you configure them.

Forwarding actions

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).

RELP Window Size

File Configuration field:

nSelpWindowSize

Description:

Configures the RELP send window size (0–4096). When greater than `1`, the value is passed to the RELP client library to control how many messages can be in flight before acknowledgment. Use `0` or `1` for default librelp behavior. Tune this when you need higher throughput on reliable links or stricter flow control on constrained links.

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

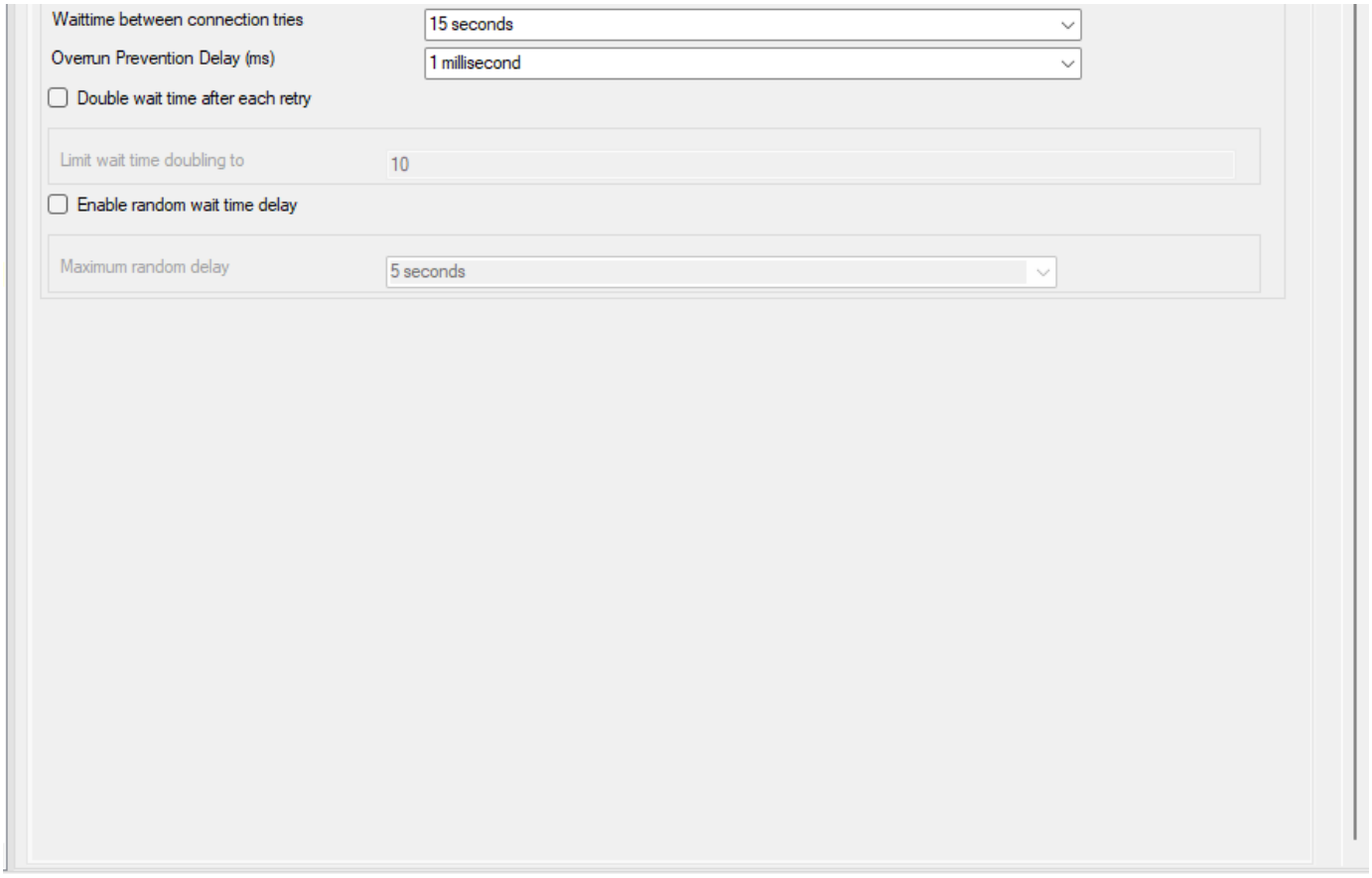
File Configuration field:

szTLSKeyFile

Description:

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

Action Queue Options



rsyslog Windows Agent - Send RELP Action Queue Options

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 are split when they reach this size in bytes. The default is 10485760 (10 MB). The maximum per file is 2147483648 bytes (2 GB). The total diskqueue size is limited by free disk space.

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 many times the wait time is doubled after a failed connection try. This is a count, not seconds. The range is 1 to 100, and the default is 10.

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

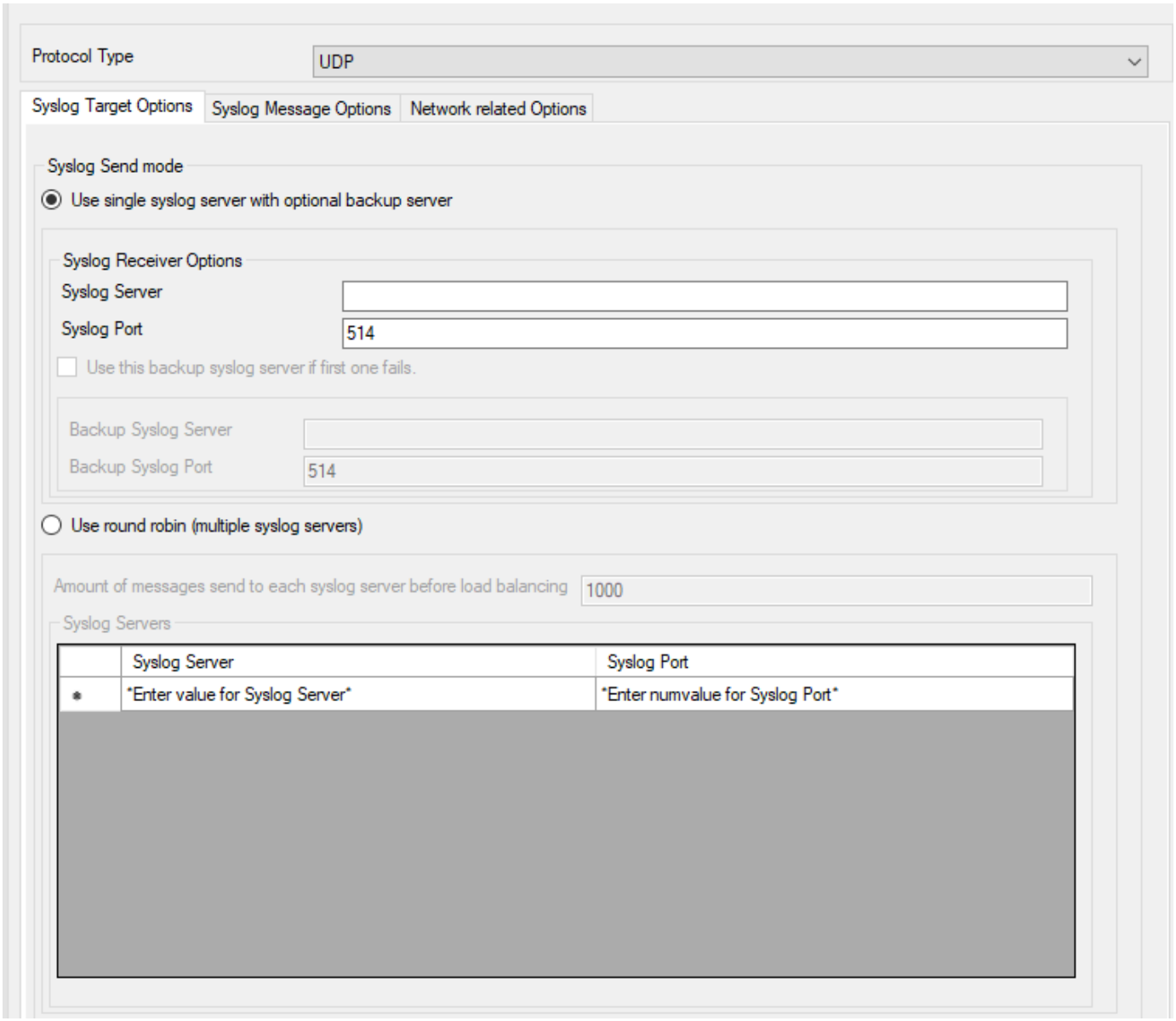
There are various ways to transmit syslog messages. In general,they can be sent via UDP, TCP, or RFC 3195 RAW. Typically, syslog messages are received via UDP protocol, which is the default. UDP is understood by almost all servers, but doesn't guarantee transport. In plain words, this means that syslog 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 receive any importance in practice. Servers are hard to find. As one of the very few, Adiscon products support RFC 3195 also in the server implementations. Due to limited deployment, however, RFC 3195 is very little 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



- 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:

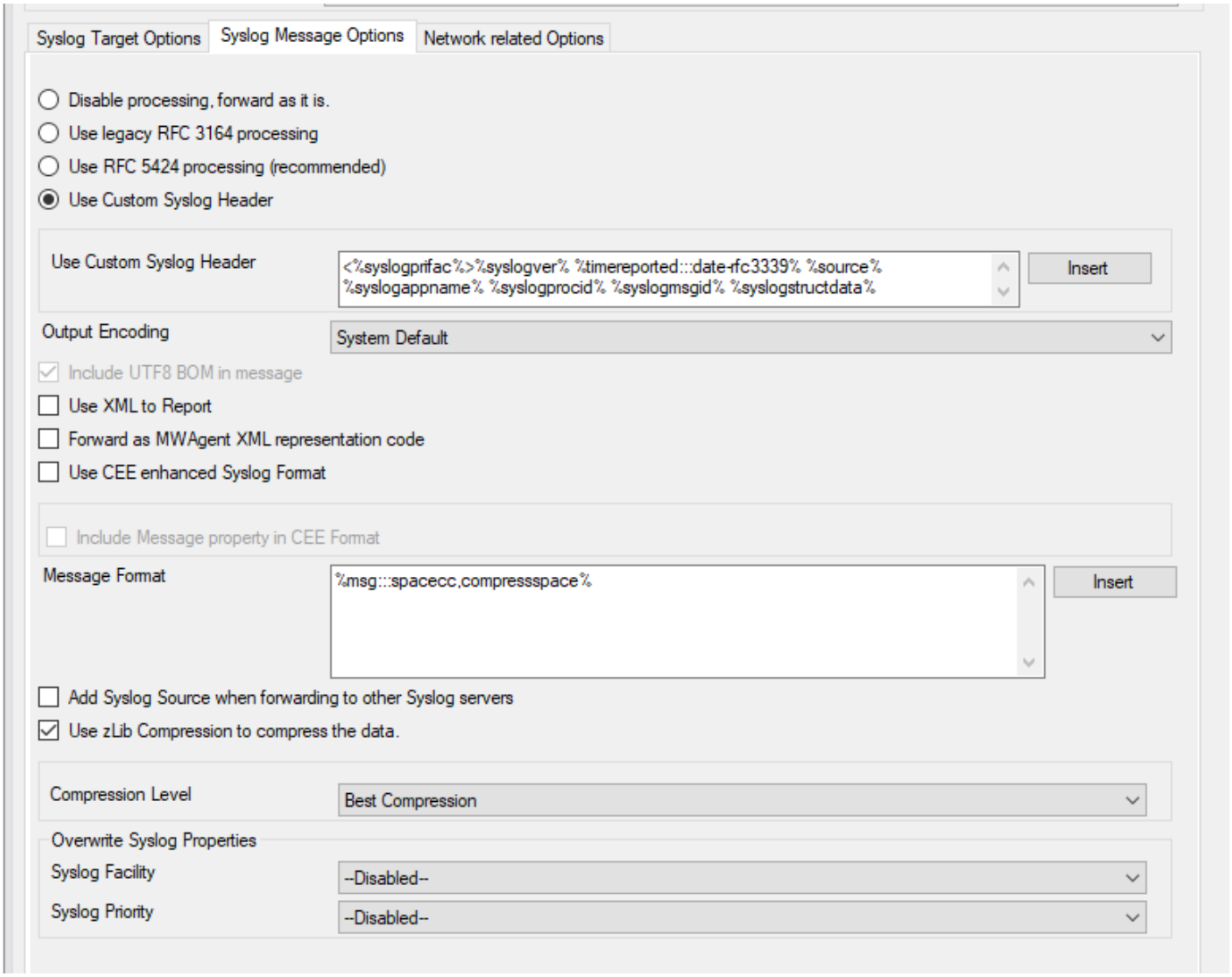
nSyslogPort_[n]

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 MWAgent 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-Wi
ndows-Security-Auditing", "id": "4648", "categoryid": "12544", "category": "12544", "keyw
```

```
ordid": "0x8020000000000000", "user": "N\A", "SubjectUserSid": "S-1-5-11-222222222-33333333
3-4444444444-5555", "SubjectUserName": "User", "SubjectDomainName": "DOMAIN", "SubjectLogon
Id": "0x5efdd", "LogonGuid": "{00000000-0000-0000-0000-000000000000}", "TargetUserName": "A
dministrator", "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 the Event ID part of the actual syslog message while forwarding to a syslog server. To do that, adjust the forwarded message format so it includes the event ID property in the custom message template.

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

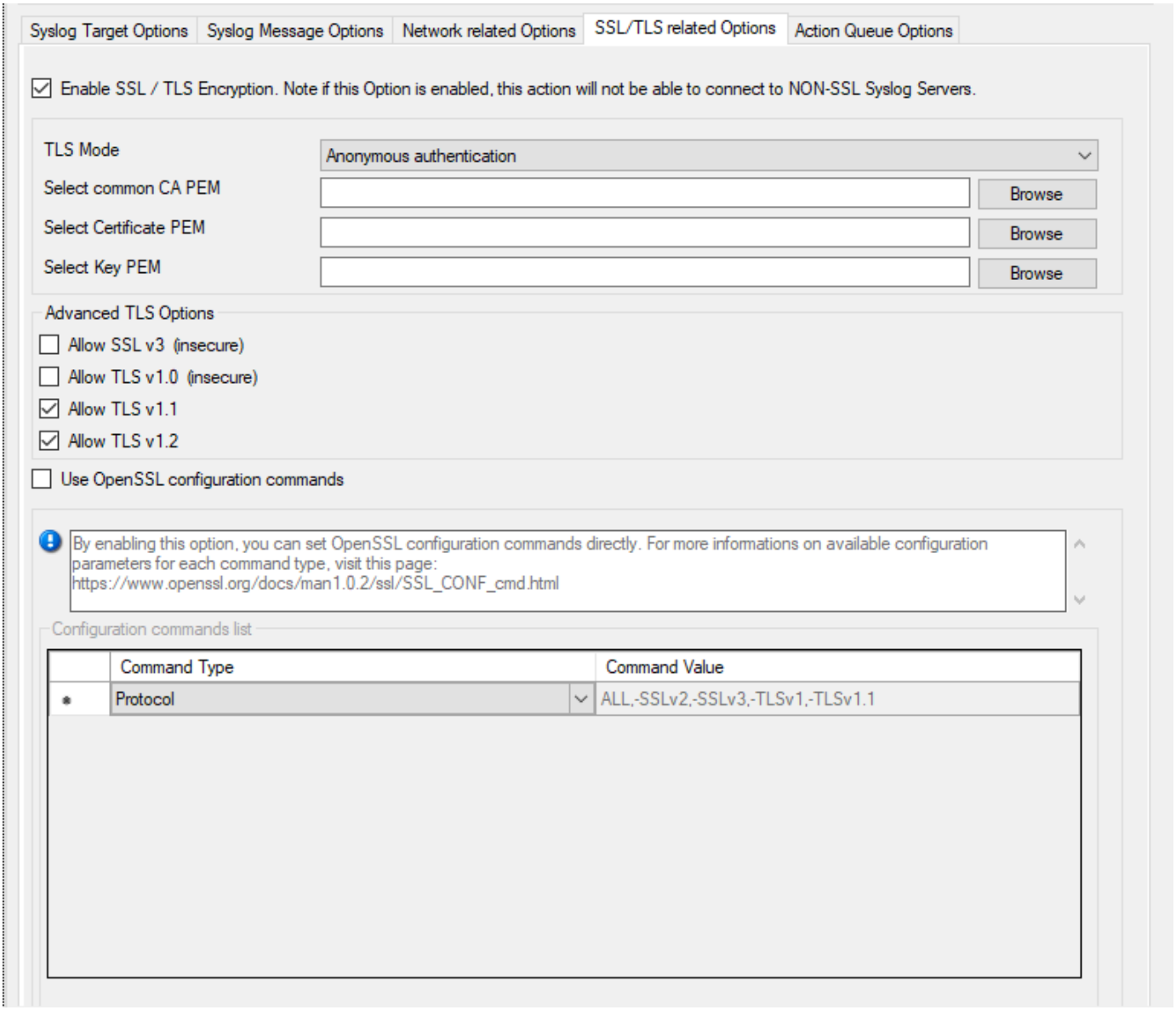
File Configuration field:

nSyslogPriority

Description:

When configured, will overwrite the Syslog Priority with the configured value.

SSL/TLS related Options



- 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

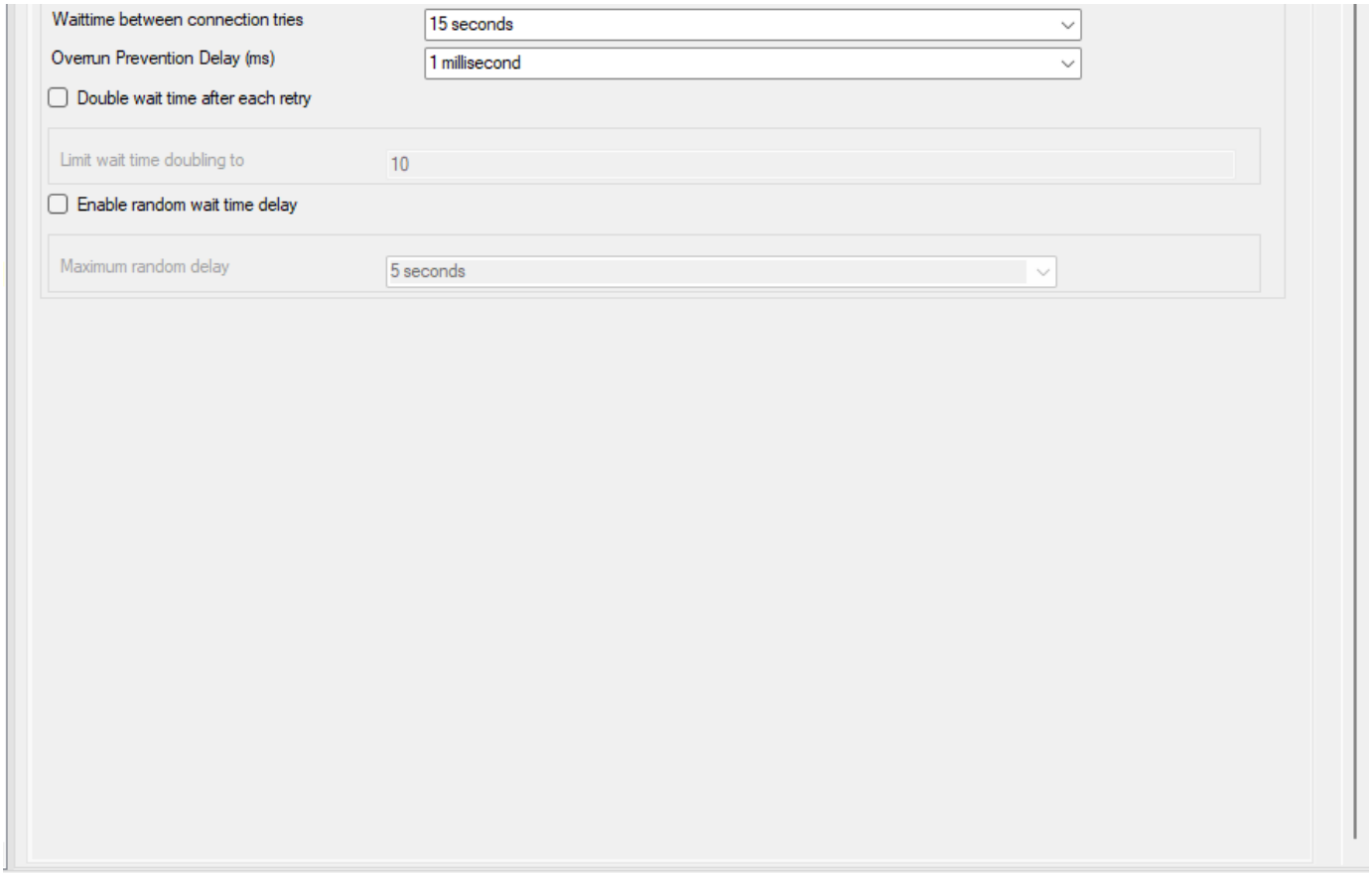
File Configuration field:

nTimeoutValue

Description:

Timeout value for TCP persistent and octet-count based framing connections.

Action Queue Options



rsyslog Windows Agent - Forward Syslog Action Queue Options

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 are split when they reach this size in bytes. The default is 10485760 (10 MB). The maximum per file is 2147483648 bytes (2 GB). The total diskqueue size is limited by free disk space.

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 many times the wait time is doubled after a failed connection try. This is a count, not seconds. The range is 1 to 100, and the default is 10.

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:

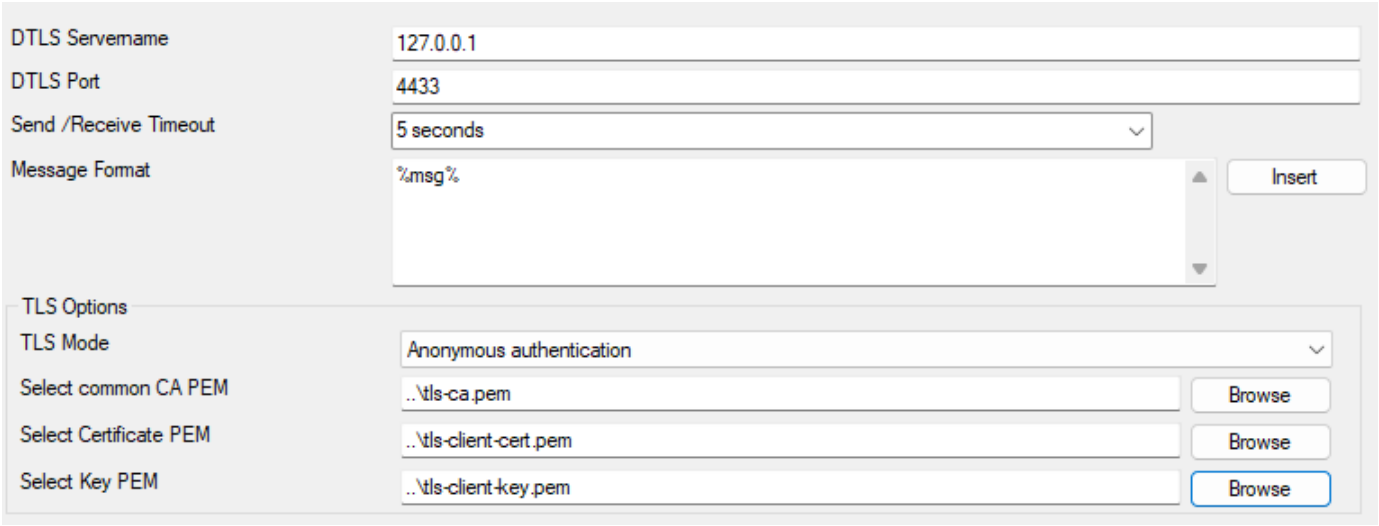
szSpoofedIPAddress

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.

Send DTLS

This action sends messages securely using Datagram Transport Layer Security (DTLS). Use it when the receiving system expects DTLS transport for forwarded messages.



- Action - Send DTLS Configuration*

DTLS Servername

File Configuration field:

szDTLSServer

Description:

The hostname or IP address of the DTLS server to which messages should be sent. You can use an IPv4, IPv6 address, or a hostname resolving to one of these.

DTLS Port

File Configuration field:

nDTLSSendPort

Description:

The port number on the DTLS server where messages are sent. Typically, this port is 4433.

Send/Receive Timeout

File Configuration field:

nSendTimeOut

Description:

Specifies the time in seconds to wait for a response from the DTLS server before timing out. For instance, set this value to “5 seconds” for a 5-second timeout.

Message Format

File Configuration field:

szMessage

Description:

Defines the format of the message being sent. Use placeholders like “%msg%” to define the dynamic content of the message. Multi-line messages are supported.

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

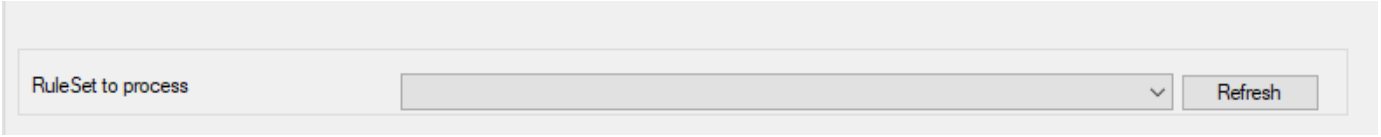
Call RuleSet

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.



- 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 then "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.

- 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.

Use the Discard action when matching events should be ignored and no later actions in the same rule chain should run for them.

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.
- XLM 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:

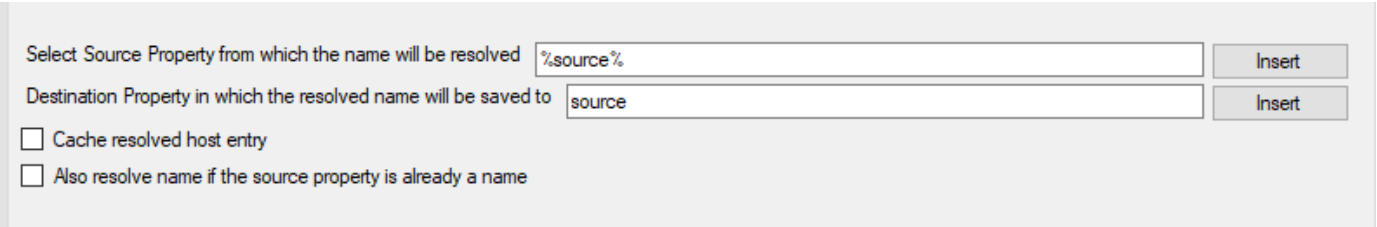
szOutputProperty

Description:

The property where the normalized format is saved to.

Resolve Hostname Action

The **Resolve Hostname** action resolves a host name or address from one event property and stores the result in another property. Because it is an action, you can place it in the ruleset where the resolved value is needed.



- 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:

nResolveIfName

Description:

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

Cache resolved host entry

File Configuration field:

nCacheNameEntry

Description:

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

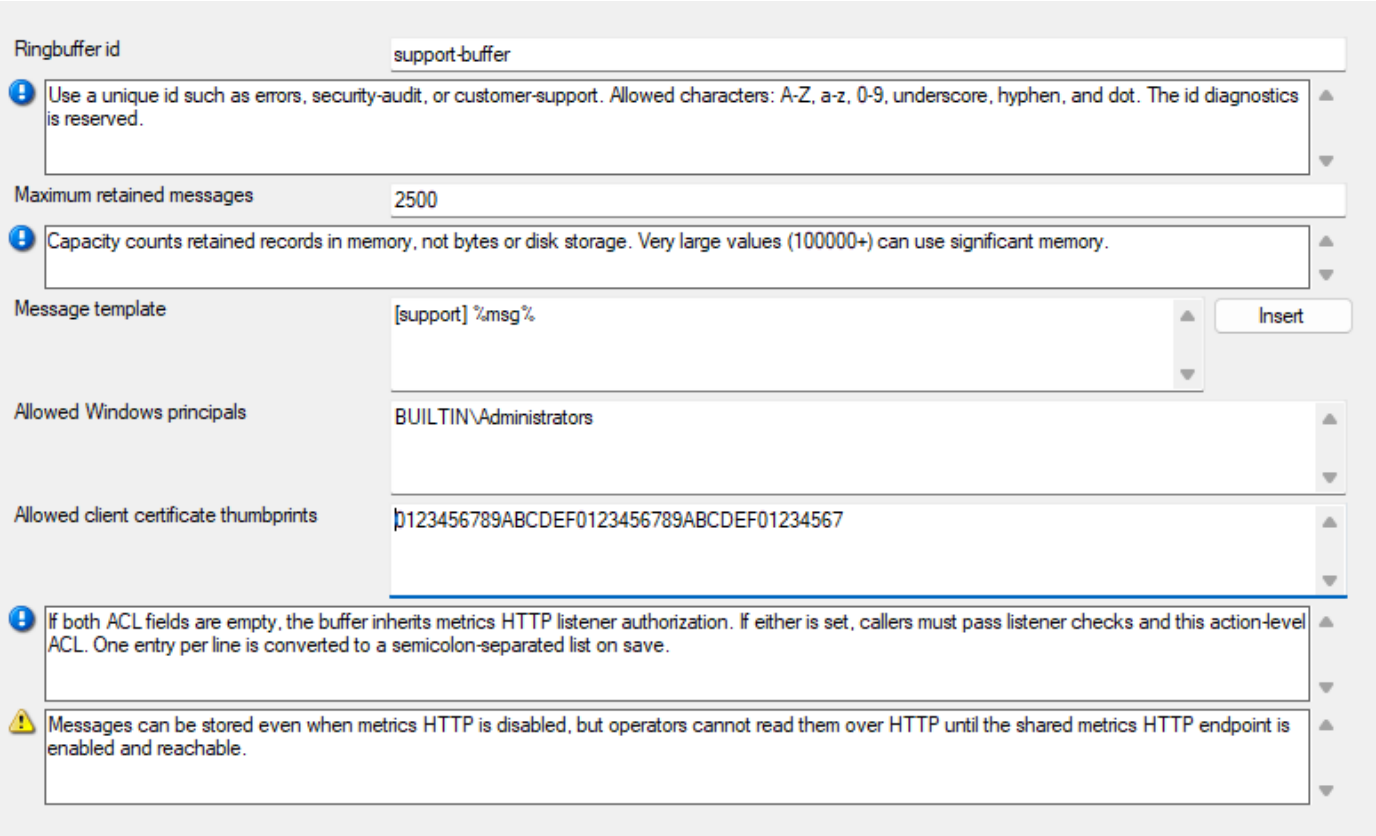
Message Ringbuffer Action

The **Message Ringbuffer** action keeps recent copies of processed messages in a named, in-memory buffer. Use it for bounded troubleshooting or an approved integration that needs recent messages. It does not replace archival logging, audit retention, or a disk-backed queue.

The action is available starting with **26.08** in MonitorWare Agent, WinSyslog, EventReporter, and rsyslog Windows Agent builds when the installed edition includes it.

Messages are retained only while the service and this action configuration remain active. When the buffer is full, the oldest retained messages are overwritten. Its contents are lost when the service restarts, when a configuration reload recreates the action, or when the action is removed.

Remote reads use the secure metrics HTTP listener. Enable that listener through the file-based `general(name="Metrics")` settings; see **`metrics-http-listener`**. Current configuration clients preserve the Metrics block but do not provide a Metrics editor.



Message Ringbuffer action properties.

Ringbuffer ID

File Configuration field:

szRingBufferId

Description:

Enter a unique ID for this buffer. Use one to 64 letters, digits, periods, underscores, or hyphens. The ID identifies the configured buffer, so do not reuse it for another Message Ringbuffer action. The ID `diagnostics` is reserved and cannot be used.

Maximum retained messages

File Configuration field:

nRingBufferMaxMessages

Description:

Set the number of messages to retain, from 2 through 1,000,000. This is a record count, not a storage size. Choose a bounded value after considering the average message size and the available memory. Larger buffers keep more history but use more memory.

Message template

File Configuration field:

szMessageTemplate

Description:

Define the text stored for each message. The default is `%msg%`. Use the property insertion control to include the event properties needed by the intended troubleshooting or integration workflow.

Allowed Windows principals

File Configuration field:

szRingBufferAllowedWindowsPrincipals

Description:

Optionally list the Windows users, groups, aliases, or SID strings that may read this buffer. Separate entries with commas, semicolons, or line breaks. Listener authorization is inherited only when both action ACL fields are empty. If either this field or **Allowed client certificate thumbprints** is populated, the caller must still pass the metrics HTTP listener checks and must also match at least one populated action ACL list.

Allowed client certificate thumbprints

File Configuration field:

szRingBufferAllowedClientCertificateThumbprints

Description:

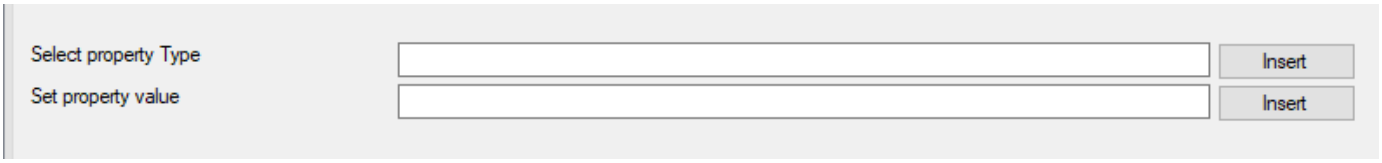
Optionally list SHA-1 client certificate thumbprints that may read this buffer. Do not enter SHA-256 fingerprints; this field matches SHA-1 only. Use this only when the deployment requires client-certificate authentication on the configured secure metrics listener. Listener authorization is inherited only when both action ACL fields are empty. If either this field or **Allowed Windows users, groups, aliases, or SID strings** is populated, the caller must still pass the metrics HTTP listener checks and must also match at least one populated action ACL list.

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!



- 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.

Difference from Set Status

Set Property changes data on the current message. The updated value is then used by later filters and actions for that same message only.

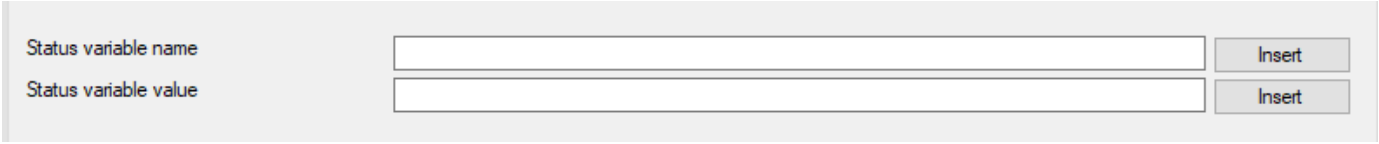
Use *Set Property* when you need to rewrite, normalize, or enrich message content. Use Set Status instead when you need a global value that persists across multiple messages.

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.

Difference from Set Property

Set Status and *Set Property* look similar, but they solve different problems:

- A **property** belongs to the current message and exists only while that message is processed.
- A **status variable** is global and remains available across multiple messages until another action changes it.

Use *Set Status* when you need shared state across messages, for example a counter, workflow state, or a flag that later filters can evaluate.

FAQ

Use this section for direct answers to common rsyslog Windows Agent questions. The entries below are grouped by user intent so you can scan by task instead of by product component.

Operation and troubleshooting

Are rsyslog Windows Agent products affected by recent OpenSSL CVEs?

Question

Are rsyslog Windows Agent products affected by recent OpenSSL CVEs? Which OpenSSL version do the products use, and are the vulnerable components used?

Problem

Customers may see OpenSSL security advisories (e.g., multiple CVEs from OpenSSL 3.6, 3.5, 3.4, 3.3, 3.0, or 1.1.1/1.0.2 branches) and need to know:

- Whether rsyslog Windows Agent is affected by specific CVEs
- Which OpenSSL version is shipped with rsyslog Windows Agent
- Whether the vulnerable code paths or components are used

Symptoms

- Security or compliance teams request a formal assessment of OpenSSL CVEs for rsyslog Windows Agent
- Scans or reports may flag rsyslog Windows Agent due to bundled OpenSSL
- No observable runtime failure; this is a security/compliance assessment topic

Solution

rsyslog Windows Agent **26.07** bundles an updated OpenSSL library as part of the platform refresh. OpenSSL advisories list affected version ranges per CVE. Many CVEs affect only certain release branches (for example 3.6, 3.5, 3.4, 3.3, 3.0, 1.1.1, 1.0.2) and do **not** include every minor line shipped with Adiscon products.

If rsyslog Windows Agent ships OpenSSL from a branch that is not in the affected set for a given CVE, the product is not vulnerable to that CVE regardless of whether the vulnerable API exists in the code base.

Information:

- OpenSSL versions are embedded into the product statically without dependencies on system-installed versions
- The product uses its own bundled OpenSSL library, independent of any OpenSSL installation on the system
- This means system OpenSSL updates do not affect the product, and conversely, the product’s OpenSSL does not affect system security

Important Notes:

- OpenSSL version information for your specific build can be obtained from Adiscon Support
- Adiscon monitors security advisories and provides updates as necessary
- For the most current information, consult the rsyslog Windows Agent release notes or contact Support

Notes

Troubleshooting the Start Program action in rsyslog

This article explains common issues with the Start Program action in rsyslog and provides solutions to resolve them.

Background

The Start Program action allows rsyslog to execute external programs or scripts when specific syslog message conditions are met. However, there are several common issues that can prevent this action from working correctly on Linux/Unix systems.

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 system log - The external program works when run manually from command line

Root Cause: rsyslog 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: `/usr/bin/curl google.com > /tmp/temp.txt`
2. **Verify executable location** - Check if the program exists in the specified path: `which curl` - Ensure the program has execute permissions: `ls -la /usr/bin/curl` - Test the command manually from terminal
3. **Check PATH environment variable** - rsyslog 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 system log - Program works when run manually but not through rsyslog

Root Cause: rsyslog runs as a system service with different permissions than your user account.

Solutions:

1. **Check file permissions** - Ensure the executable has execute permissions: `chmod +x /path/to/script` - Check if rsyslog user can access the file: `sudo -u rsyslog /path/to/script`
2. **Check directory permissions** - Ensure rsyslog can access the directory containing the script - Check parent directory permissions: `ls -la /path/to/`
3. **Use appropriate file locations** - Store scripts in system directories like `/usr/local/bin/` - Avoid user home directories unless properly configured

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 rsyslog executes the program may be different from expected.

Solutions:

1. **Use absolute paths for all file references** - Instead of: `> temp.txt` - Use: `> /tmp/temp.txt`
2. **Set working directory in scripts** - Add `cd /your/working/directory` at the beginning of scripts - Use `pushd` and `popd` for temporary directory changes

Issue 4: Environment variable problems

Symptoms: - Program runs but cannot access required environment variables - Scripts fail due to missing environment settings

Root Cause: rsyslog service may not have access to the same environment variables as your user session.

Solutions:

1. **Set environment variables in scripts** - Add `export VAR=value` at the beginning of scripts - Use absolute paths instead of relying on environment variables
2. **Use system-wide environment files** - Configure environment variables in `/etc/environment` - Use `/etc/profile.d/` for custom environment settings

Issue 5: Shell and interpreter problems

Symptoms: - Scripts fail to execute - Wrong interpreter is used for scripts

Root Cause: rsyslog may not use the same shell or interpreter as your user session.

Solutions:

1. **Use proper shebang lines** - Add `#!/bin/bash` for bash scripts - Add `#!/usr/bin/python3` for Python scripts - Ensure the interpreter path is correct
2. **Test with explicit interpreter** - Use `/bin/bash /path/to/script` instead of just `/path/to/script` - Verify the interpreter exists and is executable

Troubleshooting Steps

1. **Check system logs** - View rsyslog logs: `journalctl -u rsyslog` - Check system log: `tail -f /var/log/syslog` - Look for error messages related to your script

2. **Test with simple commands first** - Start with a basic script that creates a file - Example: `echo "Test" > /tmp/test.txt`
3. **Verify the command works manually** - Run the exact same command that rsyslog should execute - Test as the rsyslog user: `sudo -u rsyslog /path/to/script`
4. **Check rsyslog configuration** - Verify the action is properly configured in `/etc/rsyslog.conf` - Ensure the rule is being triggered correctly
5. **Test script execution** - Make script executable: `chmod +x /path/to/script` - Test with different users: `sudo -u rsyslog /path/to/script`

Example Working Configuration

Here's an example of a properly configured Start Program action for rsyslog:

rsyslog configuration:

```
# Log all messages to file and execute script
*.* /var/log/messages
*.* ^/usr/local/bin/process-syslog.sh
```

Script content (/usr/local/bin/process-syslog.sh):

```
#!/bin/bash
# Process syslog message
echo "$(date): $@" >> /var/log/processed-messages.log
```

Key points: - Full path to script - Proper shebang line - Absolute paths for output files - Script is executable and accessible by rsyslog user

Additional Tips

- **Security considerations:** Be careful with scripts that process syslog data
- **Performance:** Keep external programs lightweight to avoid impacting rsyslog performance
- **Error handling:** Add proper error handling to your scripts
- **Logging:** Add logging to your scripts to help troubleshoot issues
- **Testing:** Always test scripts thoroughly before deploying in production

If you continue to experience issues after following these steps, please contact Adiscon support with: - rsyslog version - Linux distribution and version - Exact command being executed - Any error messages from system logs - Results of manual command testing - Script permissions and ownership

High-load configuration reload stalls in rsyslog Windows Agent

This article explains why rsyslog Windows Agent configuration reloads can stall under heavy load and how to stabilize reload behavior.

Problem

During periods of high message volume, rsyslog Windows Agent detects configuration changes but may not complete the reload process. The service can appear to hang during reloads and stop requests can time out.

Symptoms

- Configuration changes are detected, but Event ID 126 ("Configuration reload successfully done") is not logged in the Windows Application Event Log
- Service stop operations timeout with "Could not stop the service within 20 seconds" errors
- Reloads complete successfully during low-load periods but fail during high-load periods
- Noticeable time gaps between detection of a configuration change and reload completion

Root Cause

A configuration reload requires the service to pause message processing, drain in-flight work, reload configuration data, reinitialize rules and actions, and then resume processing. Under high message volume, draining in-flight work takes much longer when worker threads are insufficient, which stretches the reload window and can prevent completion.

Solution

Option 1: Increase worker threads

1. Open the rsyslog Windows Agent Configuration Client.
2. Navigate to **General Options > Queue Manager**.
3. Set **Number of worker threads** to at least half the CPU core count.
    - Example: For an 8-core system, set at least 4 worker threads.
    - Example: For a 16-core system, set at least 8 worker threads.
4. Save the configuration and allow rsyslog Windows Agent to reload.

Option 2: Upgrade to the latest build

Upgrade to the latest rsyslog Windows Agent build to pick up fixes that address configuration reload and log rotation behavior under load.

Option 3: Reduce the Queue Limit when Action Queue is enabled

If you use Action Queue on database actions, review **General Options > General > Queue Limit** and keep it around 100,000 to 200,000 when configured higher. This reduces queue management overhead while Action Queue continues to buffer output.

Option 4: Disable automatic reload in production

If configuration changes are infrequent, clear **Automatically reload service on configuration changes** in **General Options > General** and perform manual service restarts during maintenance windows.

Option 5: Split configuration into multiple services and rulesets

A single high-volume service with a single ruleset forces every message to be evaluated against all rules. Split the configuration so each service uses a smaller ruleset.

Implementation steps:

1. Create additional services for the highest-volume inputs.
2. Create matching rulesets and move rules to the appropriate ruleset.
3. Update data sources to use the new services.
4. Test each service and ruleset combination.
5. Migrate one source at a time.
6. Keep the original configuration as a backup until migration is complete.

Best Practices

- Set worker threads to at least half the CPU core count.
- Use Action Queue for database actions instead of expanding the main queue.
- Plan reloads during low-traffic periods or scheduled maintenance windows.
- Monitor Event ID 126 to confirm reload completion.

Verification

1. Check the Windows Application Event Log for Event ID 126 after configuration changes.
2. Confirm rsyslog Windows Agent service stop operations complete within the timeout period.

3. Monitor worker thread utilization and queue depth during peak load.
4. Verify reload completion times are consistent under normal conditions.

How to Export rsyslog Windows Agent Settings for a Support Call

Question

How do I export the rsyslog Windows Agent configuration so I can send it to Adiscon support?

Answer

Open the rsyslog Windows Agent Configuration Client, export the settings as a text-based registry file, compress the exported file, and submit it through the Support Portal.

Details

Do **not** use the binary registry export format. Use the text-based export so support can inspect the configuration.

Action path

1. Open the rsyslog Windows Agent Configuration Client.
2. Go to **Computer -> Export Settings to Registry File**.
3. Save the file to a known location.
4. Compress the exported `.reg` file.
5. Send the ZIP archive through the support portal.

Related information

- Tutorial: Export the Configuration and Create a Debug Log

Why did my rsyslog Windows Agent update report success when the files were not upgraded correctly?

Question

Why can an in-place rsyslog Windows Agent update report success while the installed service, client, or selected features still contain files from the older release?

Answer

This can happen when Windows Installer retains an incomplete feature state while processing a cross-generation update. The setup wizard can finish and report success even though a later maintenance operation still sees features as advertised, unavailable, or otherwise not fully installed. The product configuration is normally not deleted; the problem is that the expected program files were not refreshed.

Details

This behavior has been confirmed for direct rsyslog Windows Agent updates from the 6.x generation to the 8.x generation. Other direct source versions can show the same symptoms. The source version alone does not prove that an update path is safe or affected.

The underlying mechanism is consistent with Windows Installer feature-state handling. A newer setup package can be accepted as an update without changing the installed product identity, but Windows Installer still evaluates the feature and component state recorded for the previous installation. A change between product generations can leave a feature only partially selected or advertised. `ADDLOCAL=ALL` explicitly requests all features locally, while `REINSTALL=ALL` only refreshes features that Windows Installer considers installed. This is why a two-stage repair is useful for this symptom.

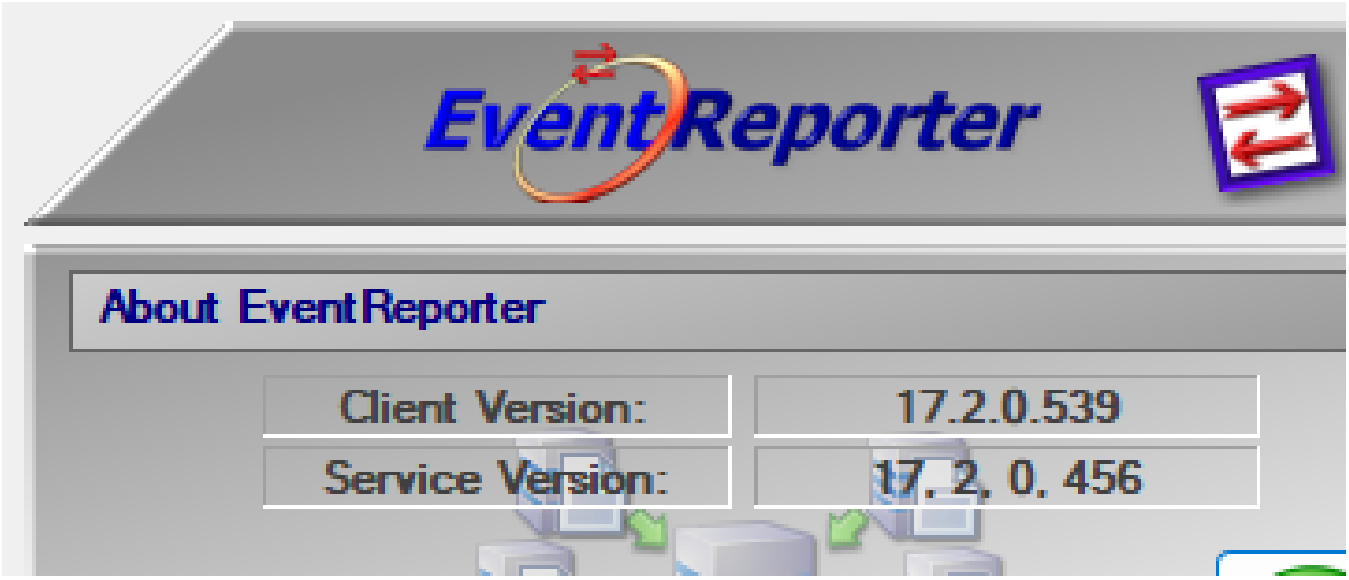
Look for this combination:

- the setup wizard displays a successful completion page
- the rsyslog Windows Agent About dialog or file properties still show an older client or service version
- Modify or Custom Setup shows an unexpected state such as `0 of 1 subfeatures selected` or `0 KB`
- a normal repair does not replace the older files
- the installer requests a reboot because a service or file is still in use

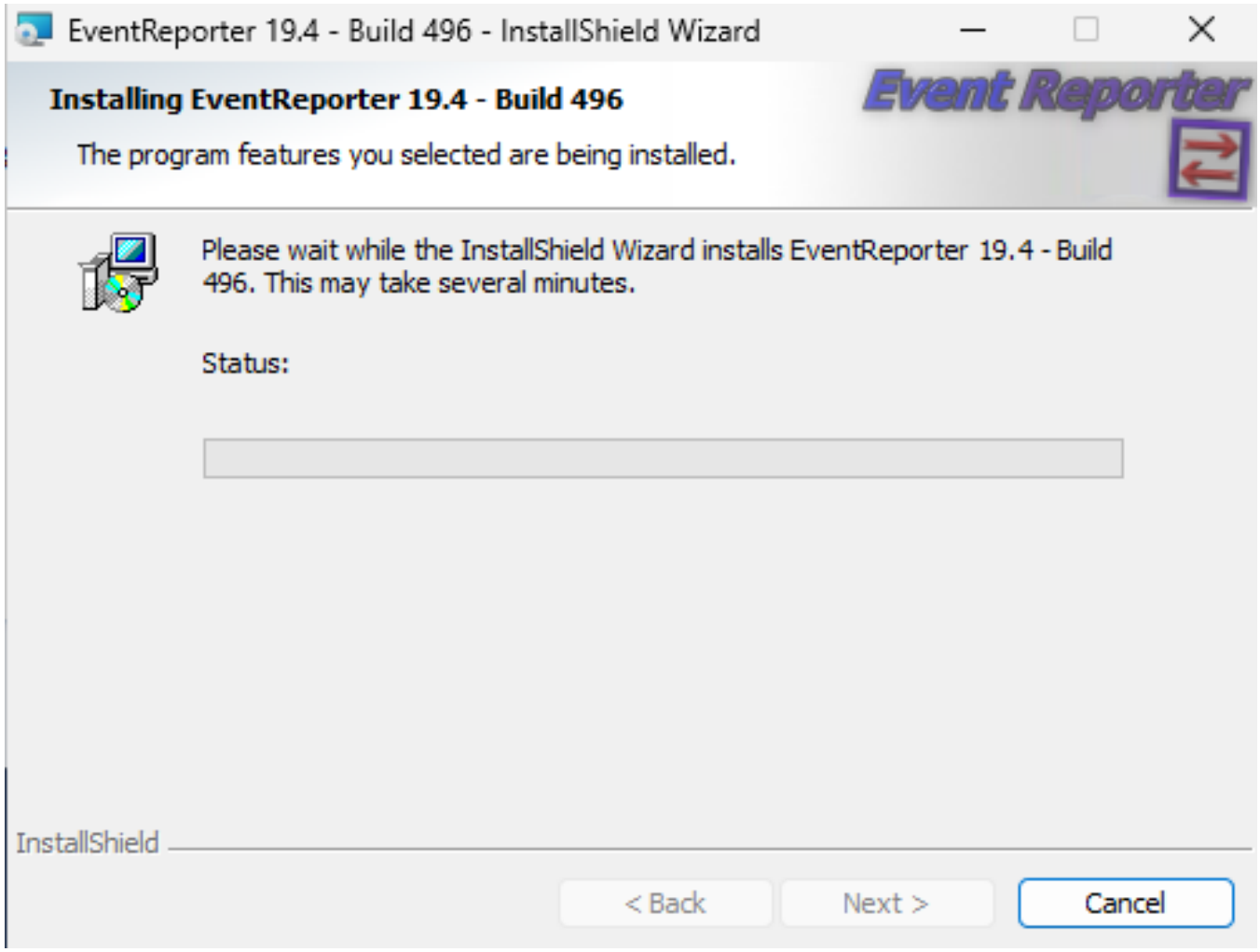
orphan:

Representative setup evidence

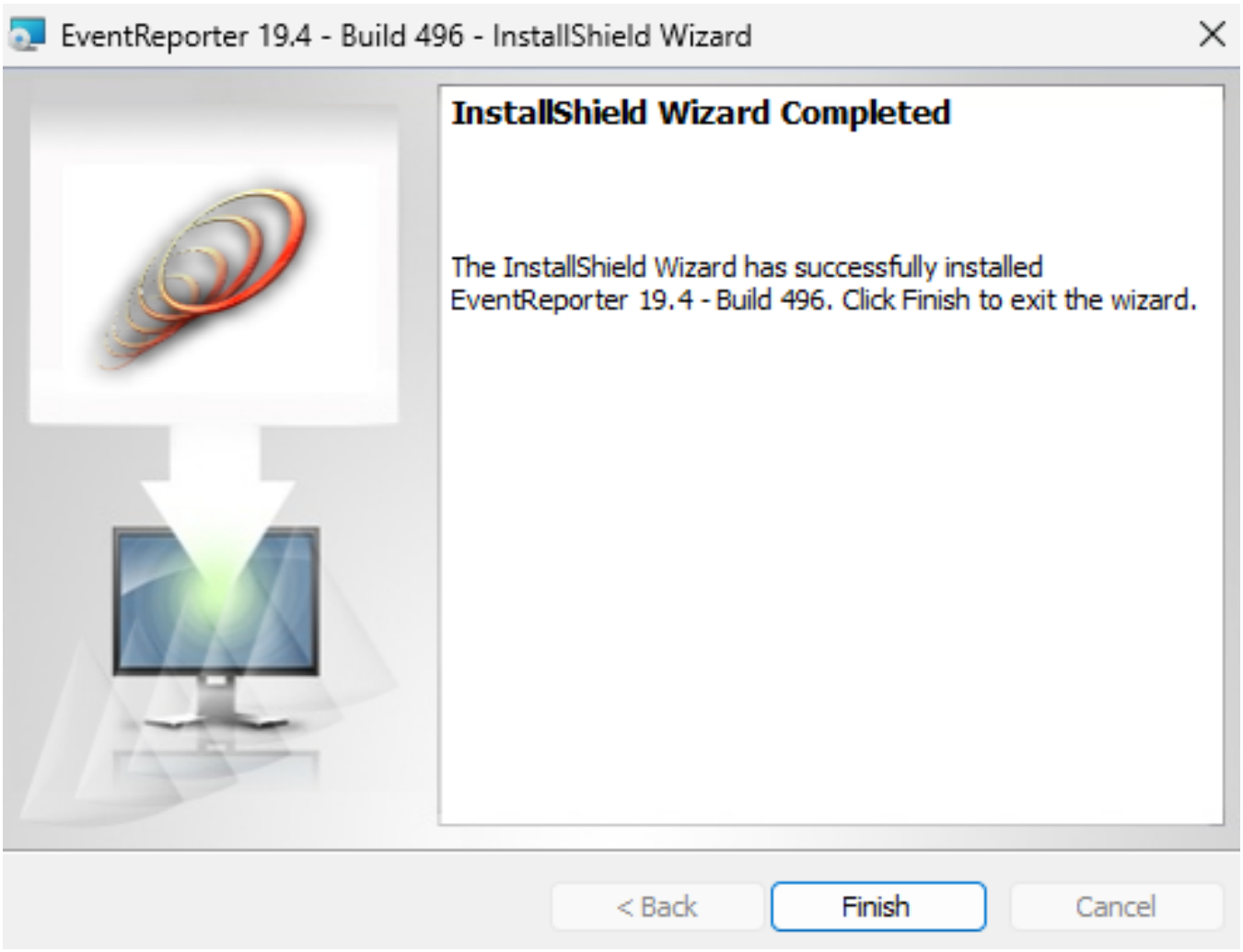
The screenshots below come from one affected setup run and are representative of the visual clues described in this article. Product names, component labels, and version numbers can differ in other product installations.



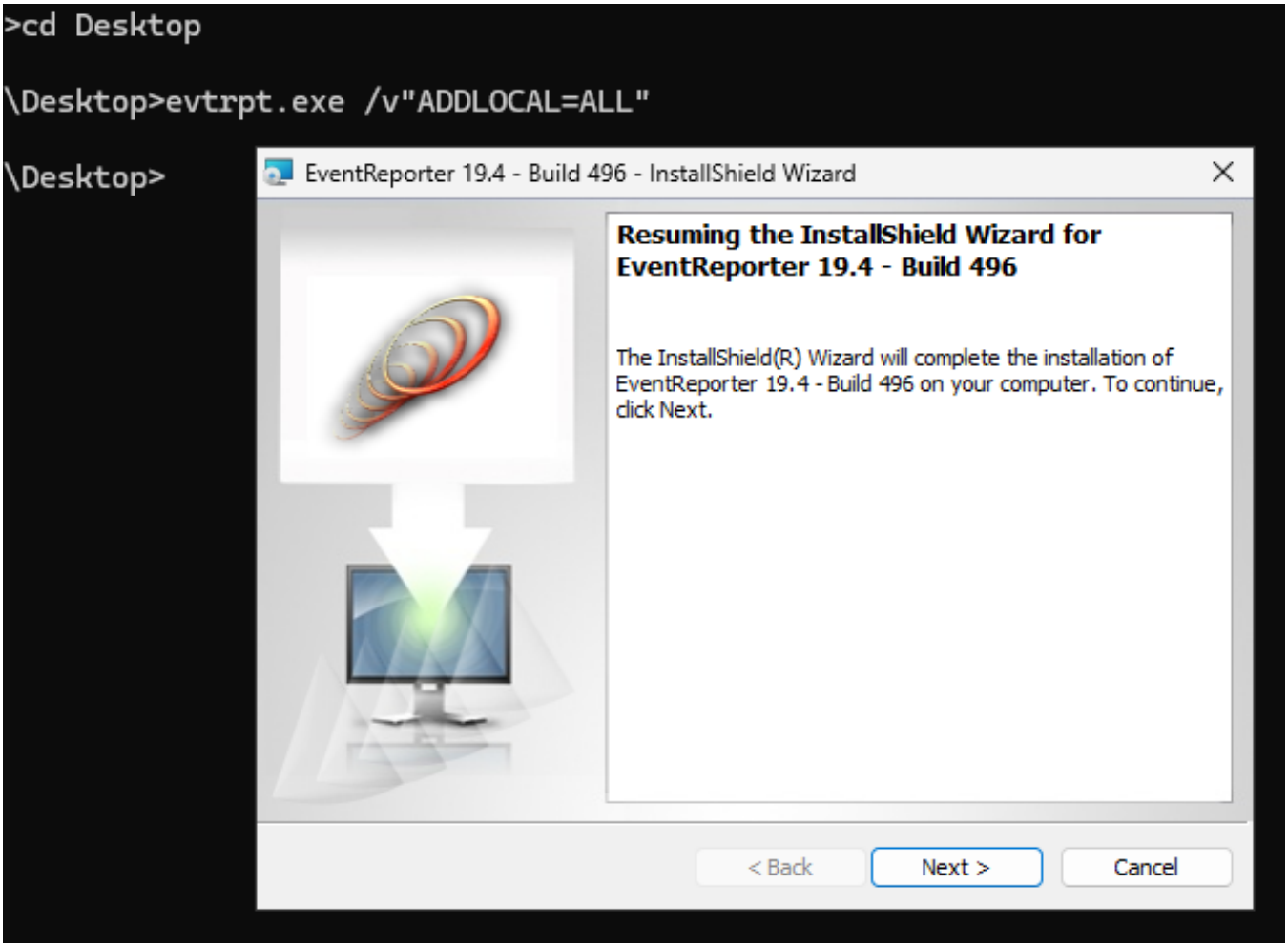
The installed product reports the older client and service versions before the update.



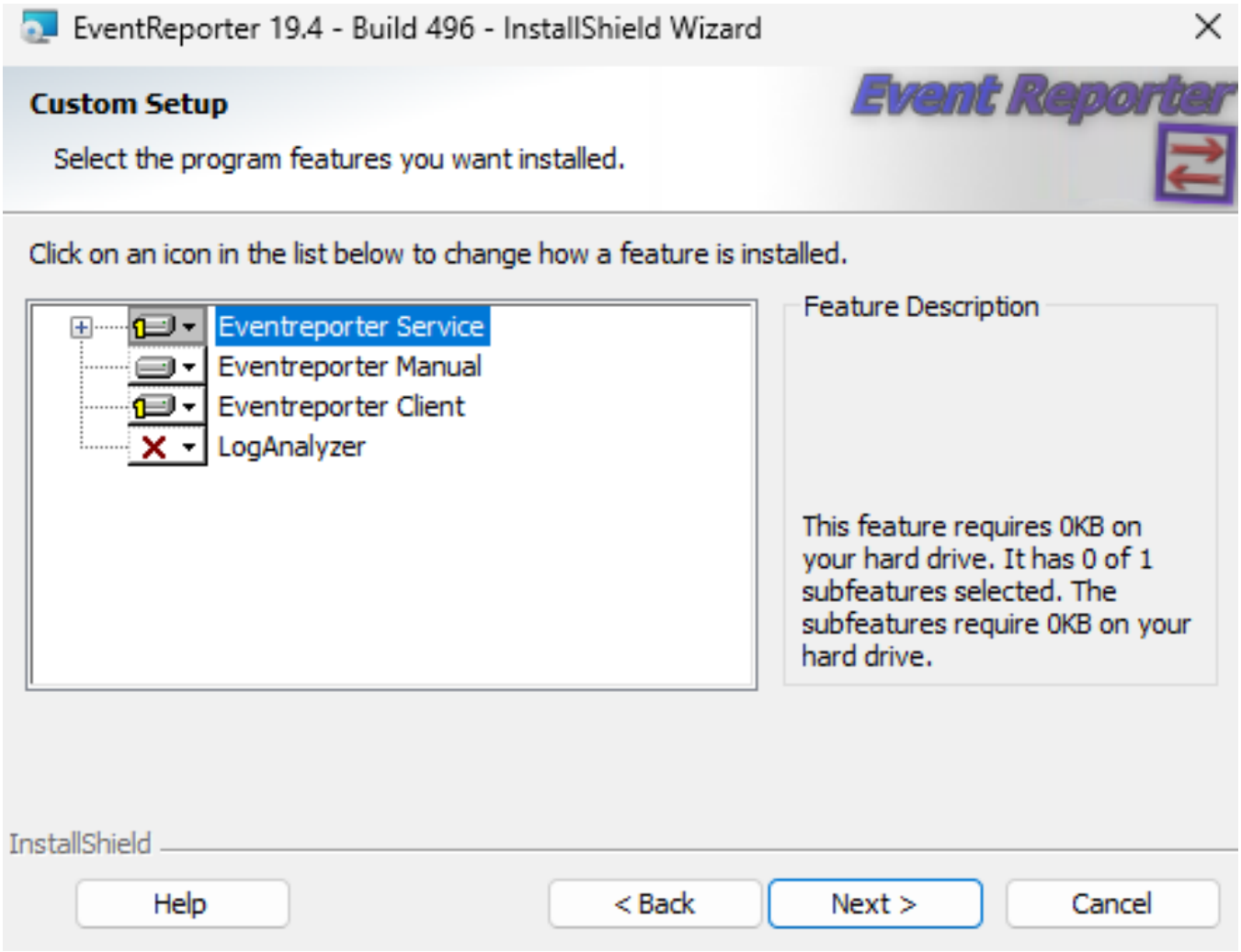
The setup wizard can display a normal installation-progress page.



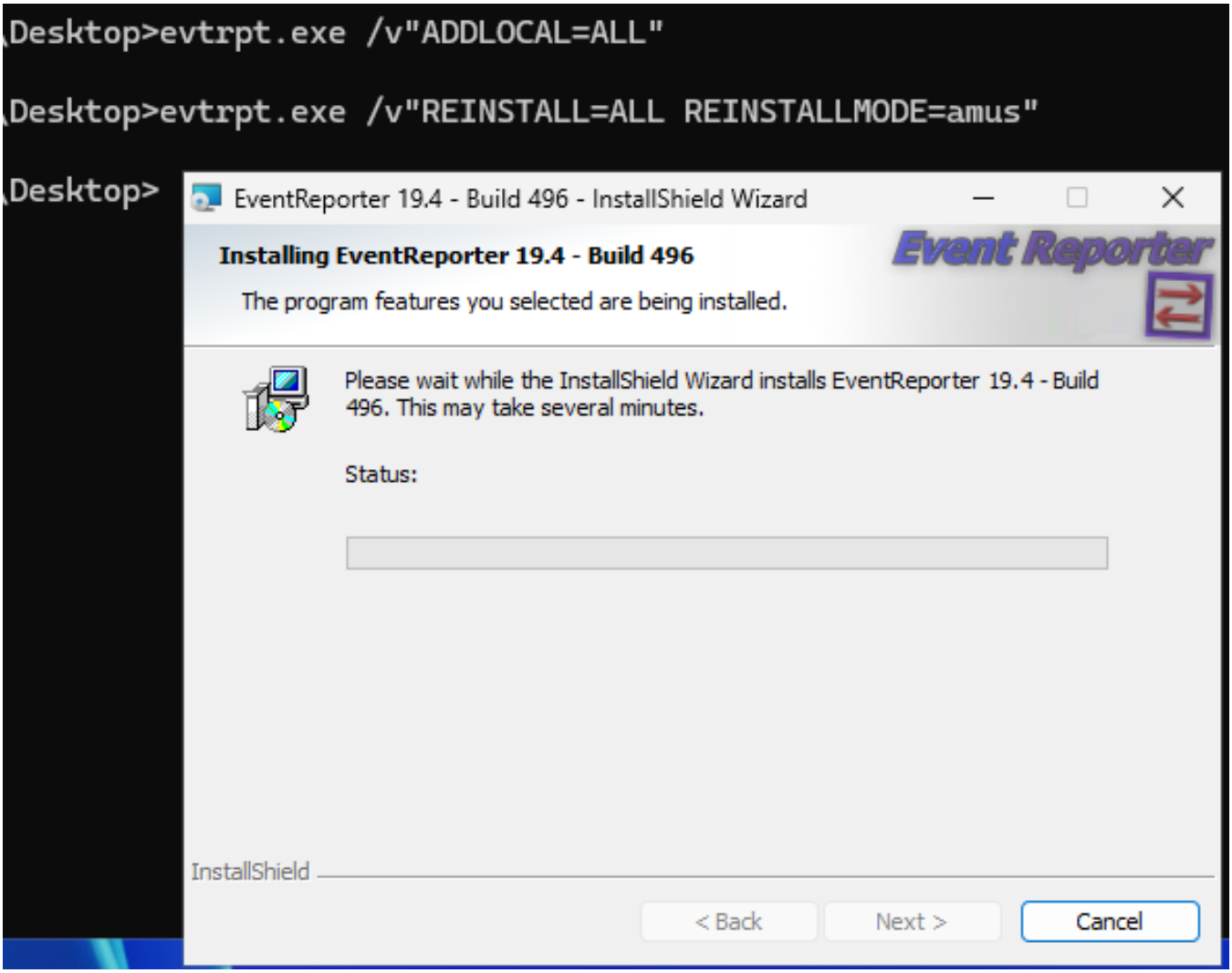
The wizard can report that the installation completed successfully.



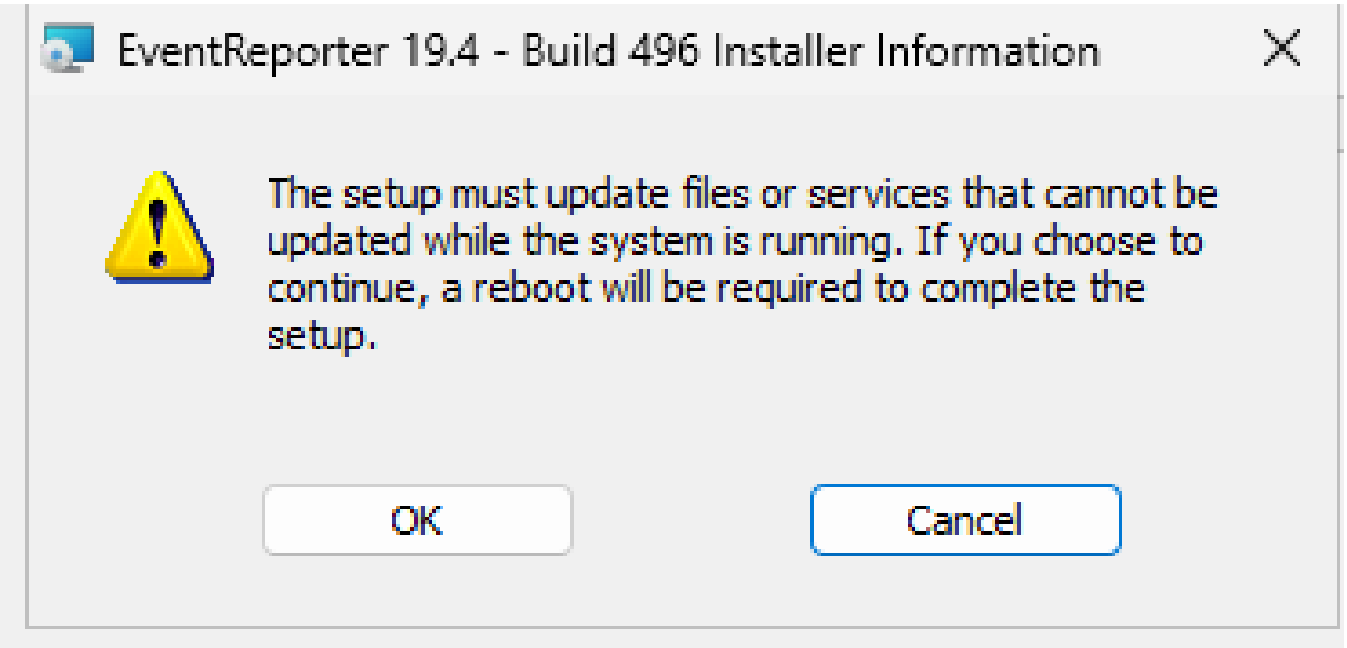
A later maintenance action can resume the InstallShield wizard.



Custom Setup can show an unexpected feature state, such as zero selected subfeatures or zero required disk space.

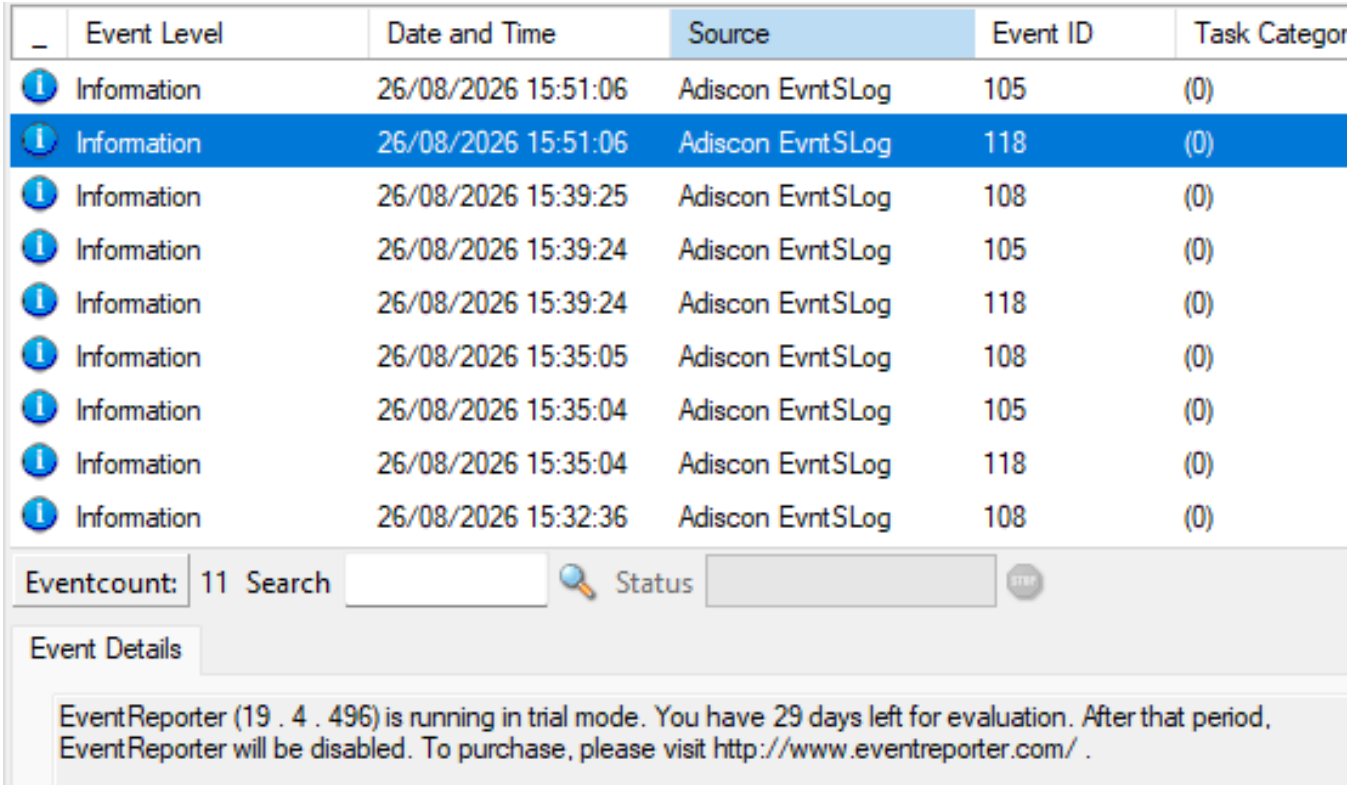


A second maintenance run can reinstall the selected product files.



Files or services that remain in use can cause the installer to request a reboot.

After a successful file refresh, the client and service report the newer versions.



Verify the running product after the installer finishes.

Action path

1. Export or back up the rsyslog Windows Agent configuration before changing the installation.

2. Close the rsyslog Windows Agent Configuration Client and stop the rsyslog Windows Agent service manually before each setup run. Stopping the service first can avoid a reboot request caused by locked service files, but it cannot avoid every reboot request.
3. If a staged update is practical, use a tested intermediate release, such as 6.x to 7.x to 8.x, and verify the client, service, features, and configuration after each hop. Do not assume that an adjacent version path is safe without verification.
4. If the installation remains inconsistent, use the clean-install procedure for rsyslog Windows Agent after exporting the configuration.
5. If you are already working from the folder that contains the installer, run the following as two separate elevated setup runs.

   First run: mark every setup feature as locally installed.

   ```
   rsyslogwa.exe /v"ADDLOCAL=ALL /L*v %TEMP%\rsyslogwa-addlocal.log"
   ```

   Second run: force a refresh of the installed files and registration data.

   ```
   rsyslogwa.exe /v"REINSTALL=ALL REINSTALLMODE=amus /L*v %TEMP%\rsyslogwa-reinstall.log"
   ```

   Let the wizard finish after each command. Do not combine the two property sets into one command for this recovery path. If the first command opens a resume page, continue through the wizard. After the second run, check Modify or Custom Setup, then verify the About dialog and the file versions before starting the service.

Related information

- **`err-2026-002-rsyslogwa-cross-generation-installer-upgrade`**
- Installation
- Microsoft Windows Installer ADDLOCAL property
- Microsoft Windows Installer REINSTALL property
- Microsoft Windows Installer REINSTALLMODE property
- Microsoft Windows Installer system reboots

How to resolve performance issues on high-load systems?

This article explains how to resolve performance issues such as slow configuration reloads, extended service start/stop times, queue buildup, and potential timeouts during service operations on systems with high message volume.

Applies To

- WinSyslog
- MonitorWare Agent
- rsyslog Windows Agent

Problem

On systems with high message volume, these products may experience performance issues such as slow configuration reloads, extended service start/stop times, queue buildup, and potential timeouts during service operations. These issues are more likely to occur when the system is processing many messages with insufficient worker threads.

Symptoms

- Configuration reloads take longer than expected or appear to hang
- Service restart operations timeout with “Could not stop the service within 20 seconds” error
- Extended service start and stop times, especially when debug logging is enabled
- Queue buildup during high message volume periods

- Slow filter and action processing during peak load
- System appears unresponsive during configuration changes

Root Cause

Under high message load, these products need sufficient parallel processing capacity to handle incoming messages efficiently. When worker threads are too few relative to the CPU cores and message volume:

- Filters and actions cannot run in parallel effectively
- Queues drain slowly, causing message buildup
- Configuration reload windows stretch because in-flight work drains slowly
- Service start/stop operations take longer because the service must wait for in-flight work to complete

The default worker thread count may be insufficient for high-load systems with multiple CPU cores.

Solution

Primary Recommendation: Increase Worker Threads

The most important setting for high-load systems is the Worker Threads configuration:

For WinSyslog:

1. Open the WinSyslog Configuration Client
2. Navigate to **General Options > Queue Manager** section
3. Set **Number of worker threads** to at least **half the CPU core count**
    - Example: For an 8-core system, set to at least 4 worker threads
    - Example: For a 16-core system, set to at least 8 worker threads
4. Save the configuration
    - If automatic configuration reload is enabled, the service will reload automatically
    - If automatic configuration reload is disabled, manually restart the service

For MonitorWare Agent:

1. Open the MonitorWare Agent Configuration Client
2. Navigate to **General Options > Queue Manager** section
3. Set **Number of worker threads** to at least **half the CPU core count**
    - Example: For an 8-core system, set to at least 4 worker threads
    - Example: For a 16-core system, set to at least 8 worker threads
4. Save the configuration
    - If automatic configuration reload is enabled, the service will reload automatically
    - If automatic configuration reload is disabled, manually restart the service

For rsyslog Windows Agent:

1. Open the product configuration interface
2. Navigate to the **Queue Manager** or equivalent section (location may vary by product)
3. Set **Number of worker threads** to at least **half the CPU core count**
    - Example: For an 8-core system, set to at least 4 worker threads
    - Example: For a 16-core system, set to at least 8 worker threads
4. Save the configuration
    - If automatic configuration reload is enabled, the service will reload automatically
    - If automatic configuration reload is disabled, manually restart the service

Why this helps:

- Allows filters and actions to run in parallel
- Drains queues faster during high message volume
- Reduces configuration reload window duration under load

- Improves overall system responsiveness during peak periods

Additional Recommendations for High-Load Systems

Test Configuration Changes on Test System First

For busy production systems, avoid making large configuration changes directly:

1. Perform larger configuration changes on a test system first
2. Verify the changes work correctly on the test system
3. Import the verified configuration into the production system
4. If automatic configuration reload is enabled, the service will reload automatically after saving
5. If automatic configuration reload is disabled, manually restart the service from the Configuration Client or Windows Services Management Console
6. Monitor system performance after the change

Use Windows Services Management Console for Service Operations

If automatic configuration reload is disabled and you need to restart the service, or if the configuration client service restart fails or times out:

1. Open Windows Services Management Console (services.msc)
2. Locate the product service (WinSyslog, MonitorWare Agent, or rsyslog Windows Agent service)
3. Right-click and select **Restart** (or **Stop** then **Start**)
4. The Services Console provides more reliable service control under high load

Note: If automatic configuration reload is enabled, manual service restart is typically not needed after configuration changes.

Allow Sufficient Time for Operations

On high-load systems, allow adequate time for:

- Configuration reloads to complete
- Service restarts to finish
- Queue processing to catch up after changes

Avoid making multiple rapid configuration changes in succession.

Best Practices

- **Set Worker Threads to at least half the CPU core count** - This is the most critical setting for high-load systems
- **Test configuration changes on a test system first** - Reduces risk of issues on production systems
- **Monitor system performance** - Watch for queue buildup, slow processing, or extended operation times
- **Avoid debug logging on production systems** - Debug logging significantly increases processing overhead and extends operation times
- **Plan configuration changes during lower-traffic periods** - When possible, schedule major changes during maintenance windows
- **Monitor Event ID 126 (WinSyslog)** - Verify that configuration reloads complete successfully (“Configuration reload successfully done”)

Verification

To verify that worker thread settings are appropriate:

1. Check the current Worker Threads setting in QueueManager section
2. Determine your system’s CPU core count
3. Verify that Worker Threads is set to at least half the CPU core count
4. Monitor system performance during peak message volume:
    - Check for queue buildup
    - Monitor service responsiveness
    - Verify configuration reloads complete in reasonable time
    - Check for timeout errors during service operations

If performance issues persist after adjusting worker threads, consider:

- Further increasing worker threads (up to the CPU core count)
- Reviewing filter and action complexity

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.

Modern Deployment Recommendations

For current MariaDB deployments, we recommend the following:

1. **Use a current MariaDB server and connector** - Use currently supported MariaDB server releases - Use a current MariaDB Connector/ODBC package from the official source
2. **Use a dedicated database account** - Create a dedicated user for the Adiscon product - Grant only the required privileges on the target database/schema
3. **Enable secure transport for remote database connections** - Use TLS between the Adiscon host and MariaDB server when traffic crosses networks - Configure certificate settings in the DSN/driver according to your security policy
4. **Use UTF-8 consistently** - Prefer UTF-8/`utf8mb4` settings for server, database, and connector - This prevents character conversion issues in international log messages

5. **Validate end-to-end before production rollout** - Use “Verify Database” in the ODBC action - Insert sample messages and verify they are written and readable as expected

Common Modern Troubleshooting Checks

If connection tests fail, verify:

- Driver architecture matches the product runtime (32-bit vs 64-bit)
- Host, port, database name, and credentials in the DSN are correct
- MariaDB user authentication method is supported by the installed connector
- TLS requirements (if enabled) match server and connector configuration
- Firewall rules allow database traffic

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.

Deployment and administration

How Do I Prepare rsyslog Windows Agent for Repeatable Deployment?

Answer

Use a reference system to build and test the configuration, export that configuration, and then deploy the service and settings consistently to the target systems.

Details

Repeatable deployment is a deployment task, not a first-time setup task. The goal is to prepare one known-good configuration and reuse it safely across multiple Windows systems.

A typical repeatable deployment uses:

- one reference installation for testing
- exported configuration data for reuse
- installer options or deployment tooling that match your environment
- a validation step on a small pilot group before broader deployment

Action path

1. Install and configure rsyslog Windows Agent on a reference system.
2. Verify that event collection and forwarding work as expected.
3. Export the configuration from the reference system.
4. Decide whether the deployment needs the full product or only the service on the target systems.
5. Deploy the software and import the validated configuration on a pilot group.
6. Confirm that pilot systems forward the expected events.
7. Deploy to the remaining systems.

Related information

- How Can I Copy an rsyslog Windows Agent Configuration to Another System?
- Installation

Why did my rsyslog Windows Agent update report success when the files were not upgraded correctly?

Question

Why can an in-place rsyslog Windows Agent update report success while the installed service, client, or selected features still contain files from the older release?

Answer

This can happen when Windows Installer retains an incomplete feature state while processing a cross-generation update. The setup wizard can finish and report success even though a later maintenance operation still sees features as advertised, unavailable, or otherwise not fully installed. The product configuration is normally not deleted; the problem is that the expected program files were not refreshed.

Details

This behavior has been confirmed for direct rsyslog Windows Agent updates from the 6.x generation to the 8.x generation. Other direct source versions can show the same symptoms. The source version alone does not prove that an update path is safe or affected.

The underlying mechanism is consistent with Windows Installer feature-state handling. A newer setup package can be accepted as an update without changing the installed product identity, but Windows Installer still evaluates the feature and component state recorded for the previous installation. A change between product generations can leave a feature only partially selected or advertised. `ADDLOCAL=ALL` explicitly requests all features locally, while `REINSTALL=ALL` only refreshes features that Windows Installer considers installed. This is why a two-stage repair is useful for this symptom.

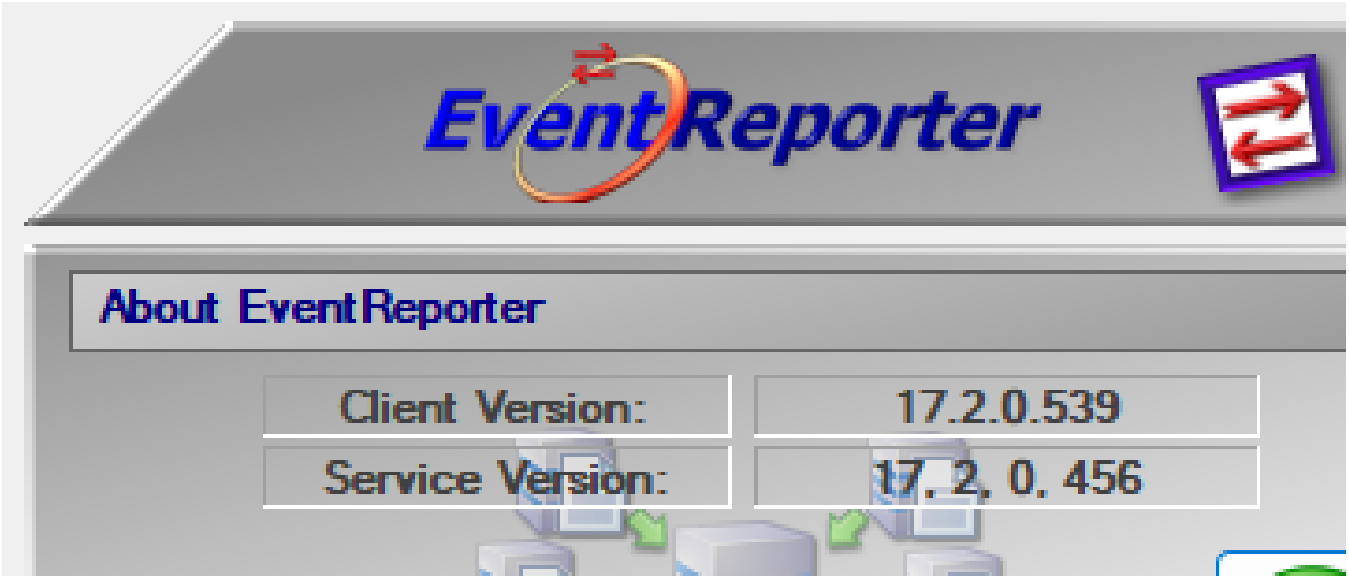
Look for this combination:

- the setup wizard displays a successful completion page
- the rsyslog Windows Agent About dialog or file properties still show an older client or service version
- Modify or Custom Setup shows an unexpected state such as `0 of 1 subfeatures selected` or `0 KB`
- a normal repair does not replace the older files
- the installer requests a reboot because a service or file is still in use

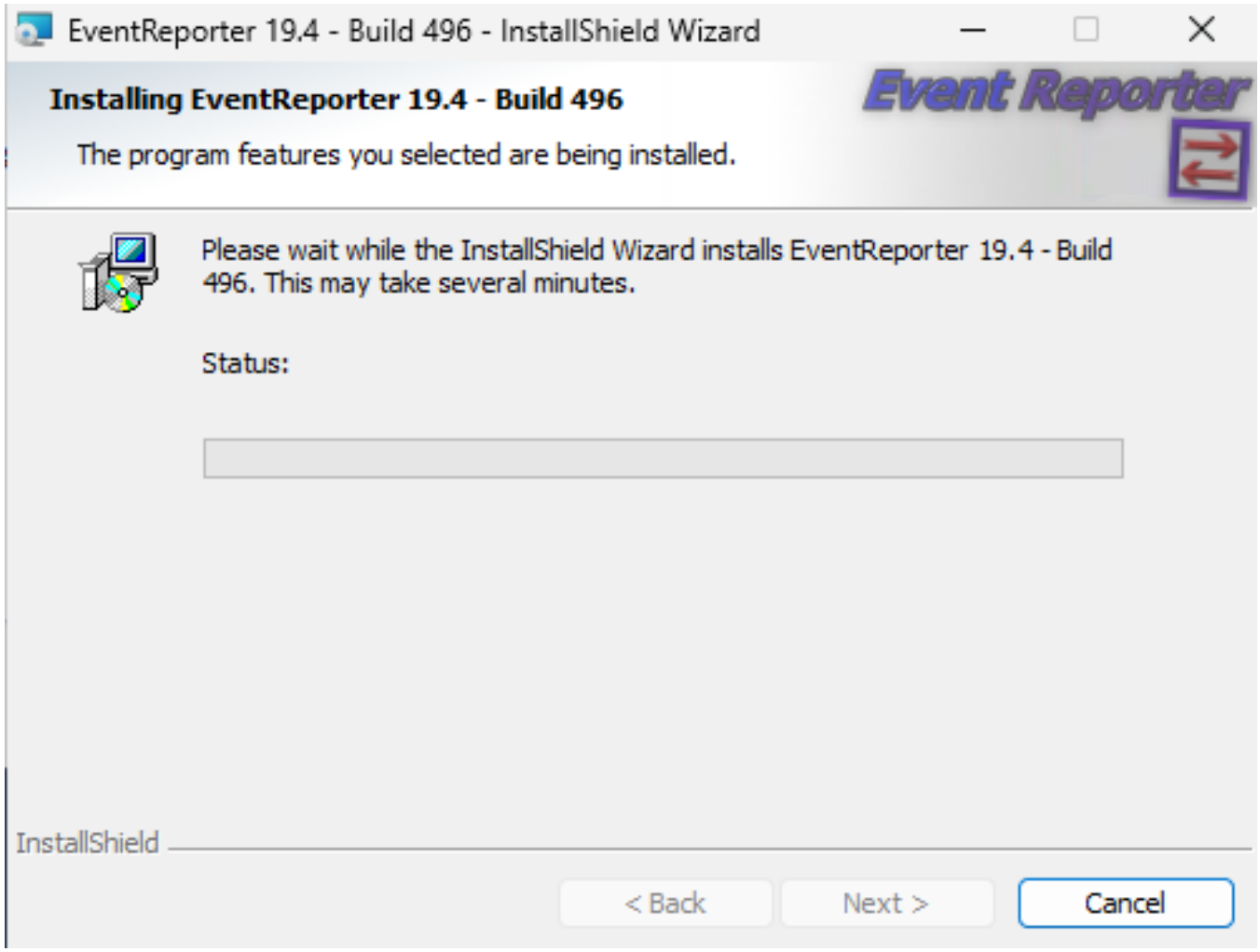
orphan:

Representative setup evidence

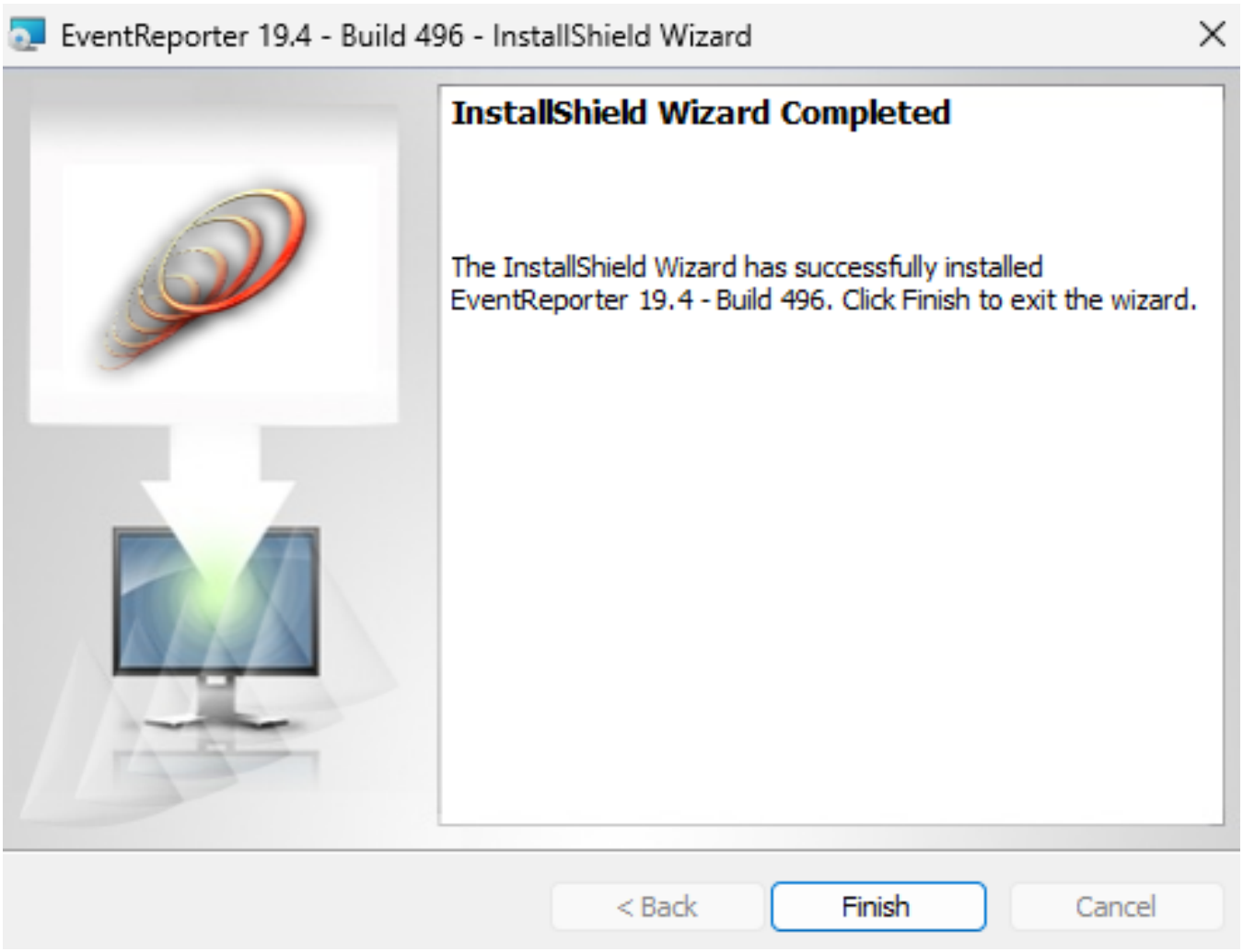
The screenshots below come from one affected setup run and are representative of the visual clues described in this article. Product names, component labels, and version numbers can differ in other product installations.



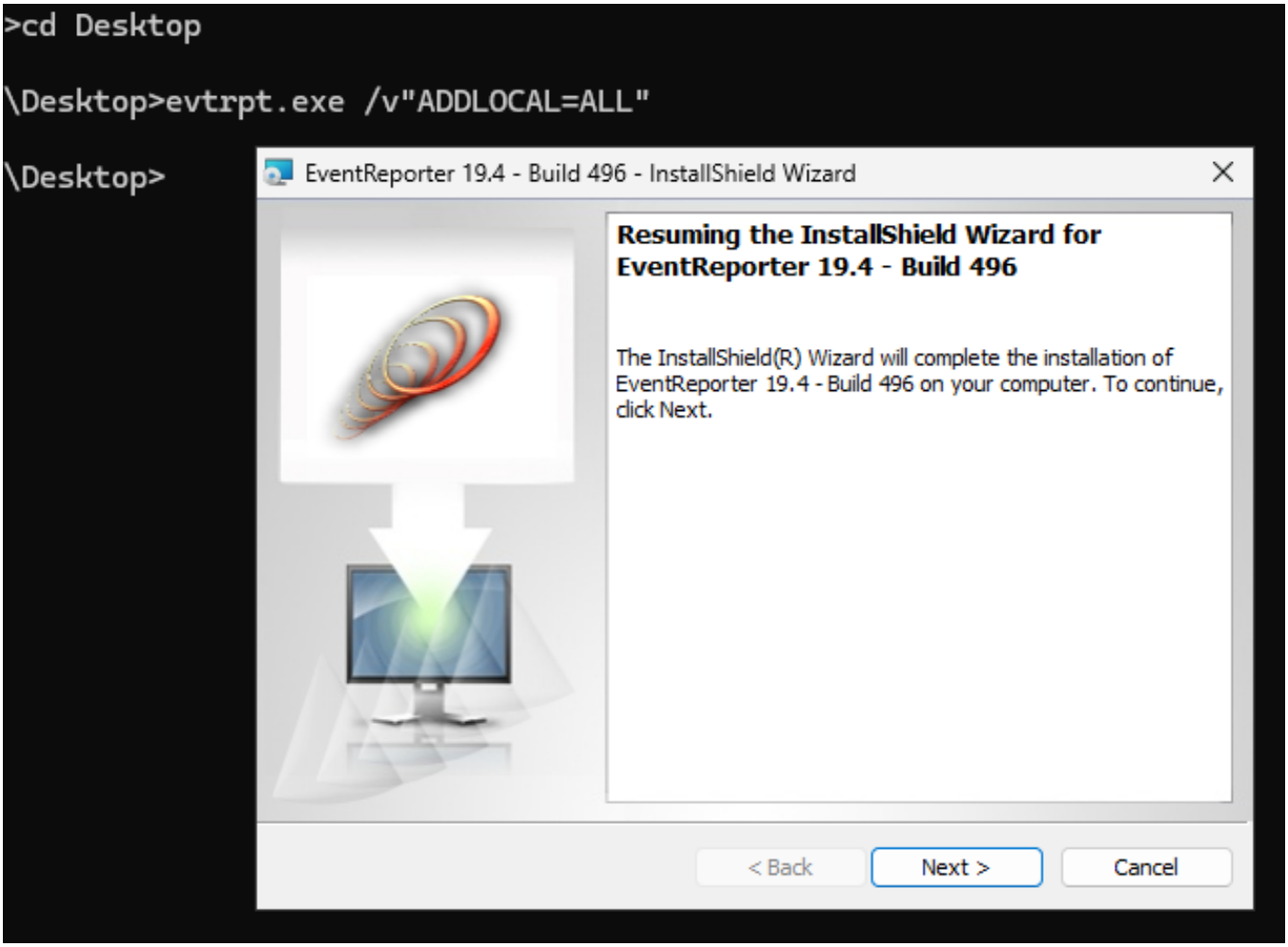
The installed product reports the older client and service versions before the update.



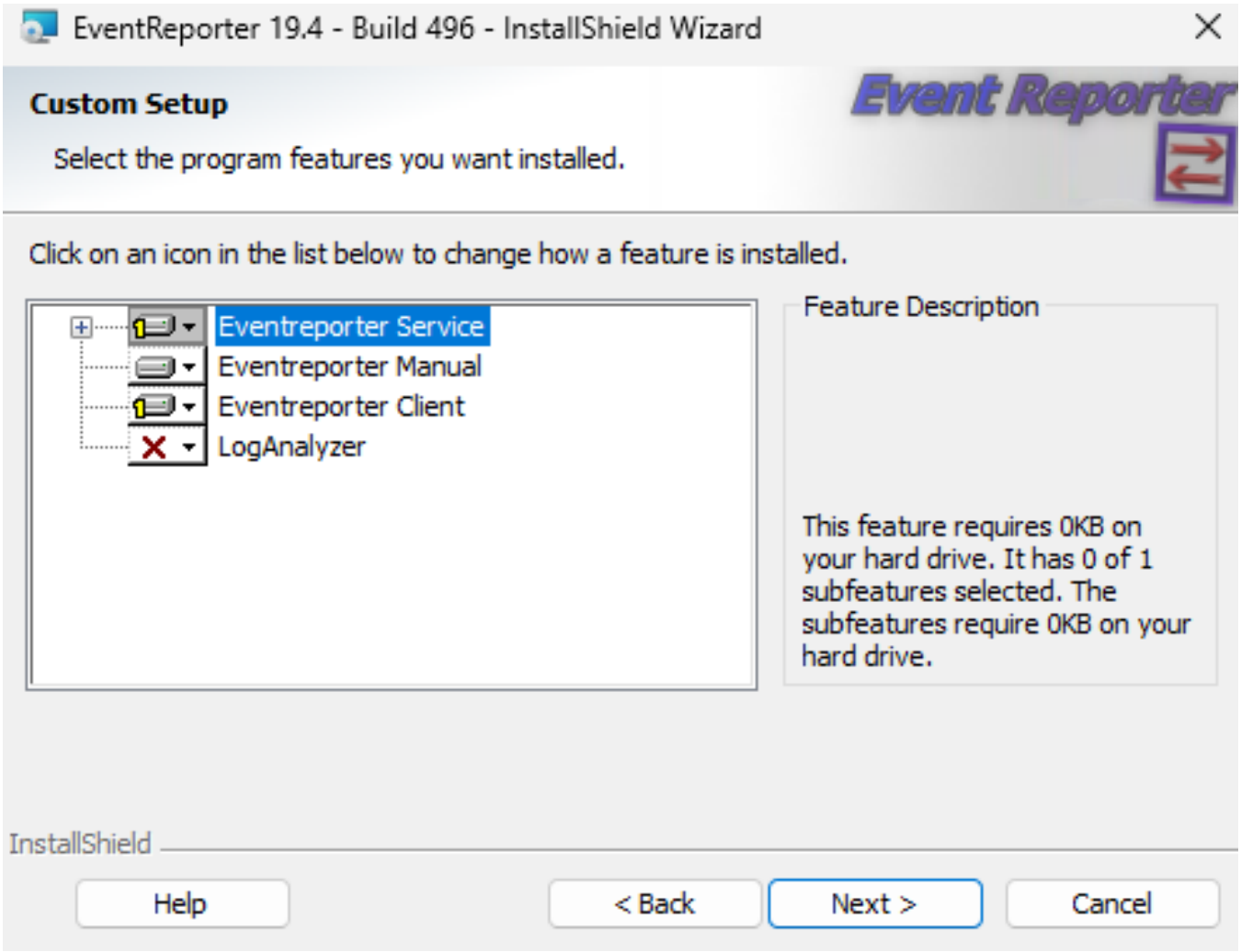
The setup wizard can display a normal installation-progress page.



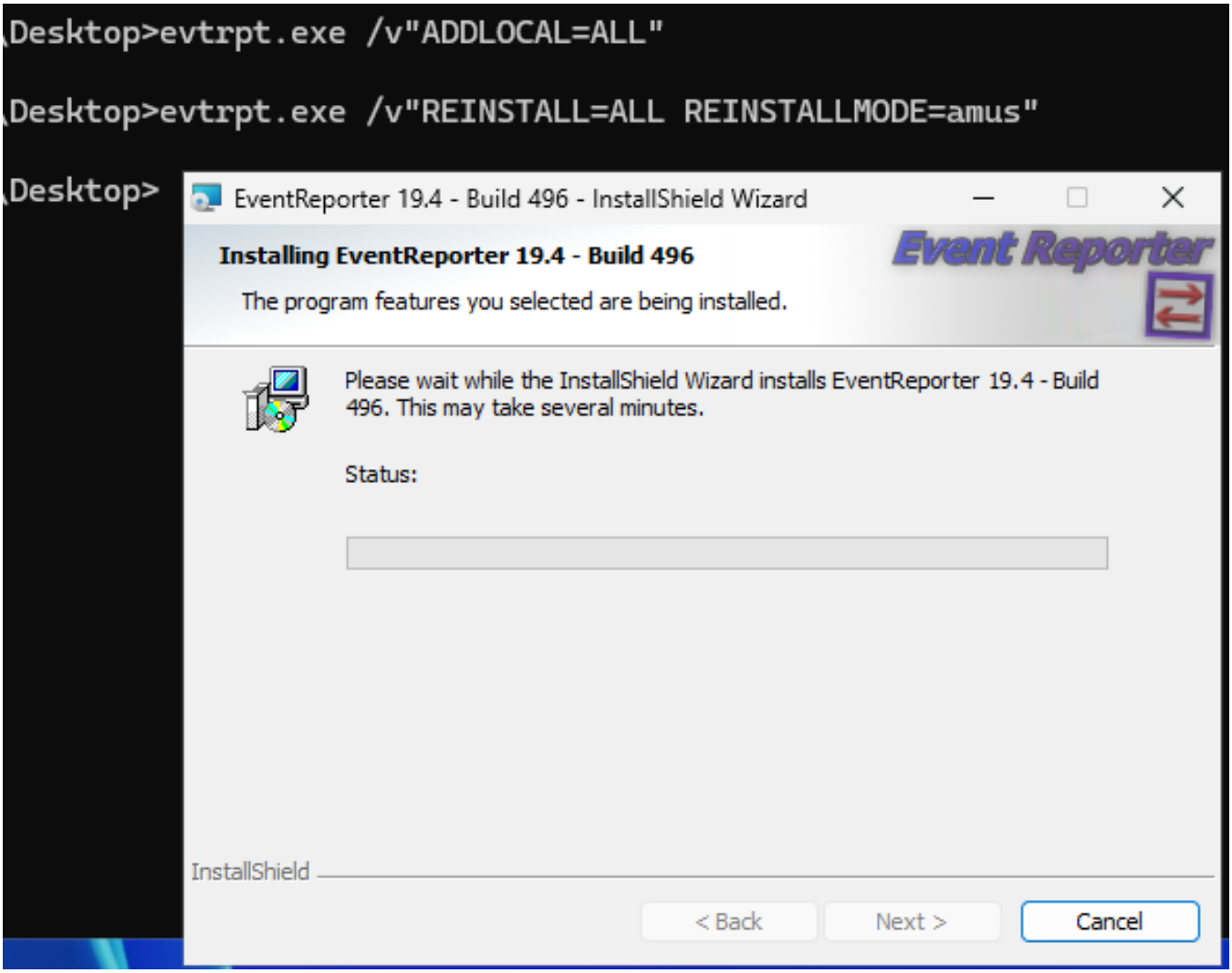
The wizard can report that the installation completed successfully.



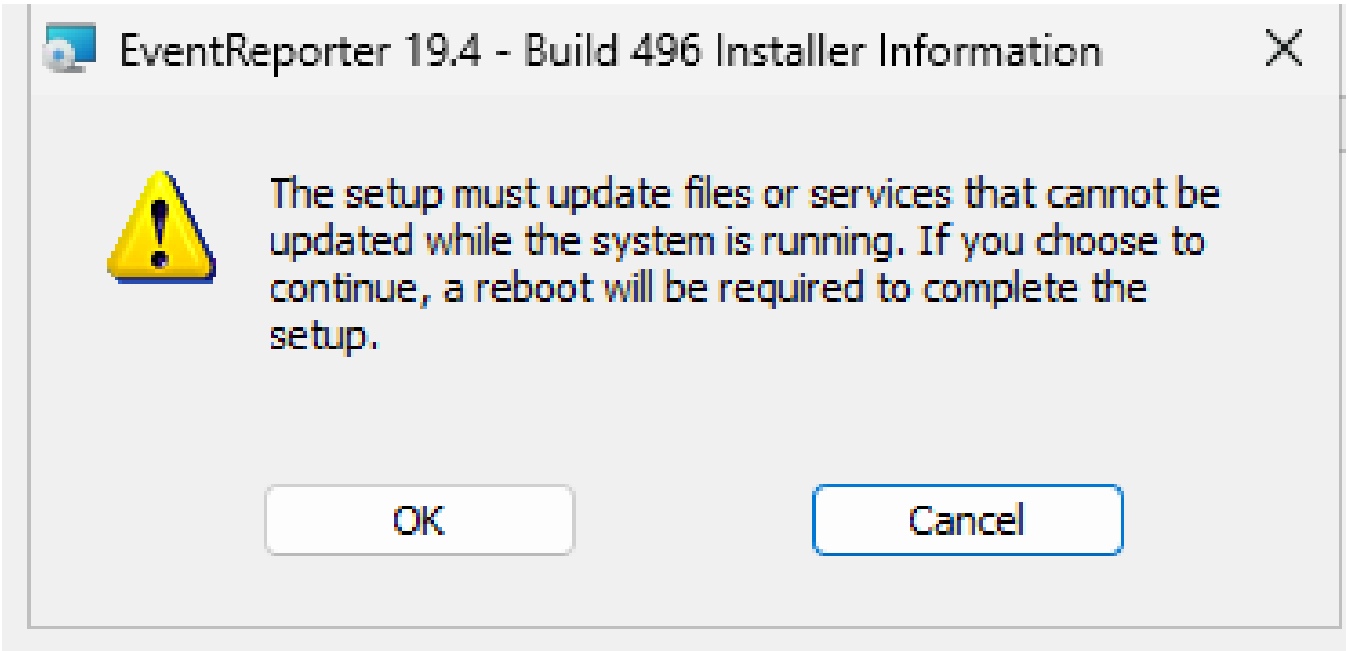
A later maintenance action can resume the InstallShield wizard.



Custom Setup can show an unexpected feature state, such as zero selected subfeatures or zero required disk space.

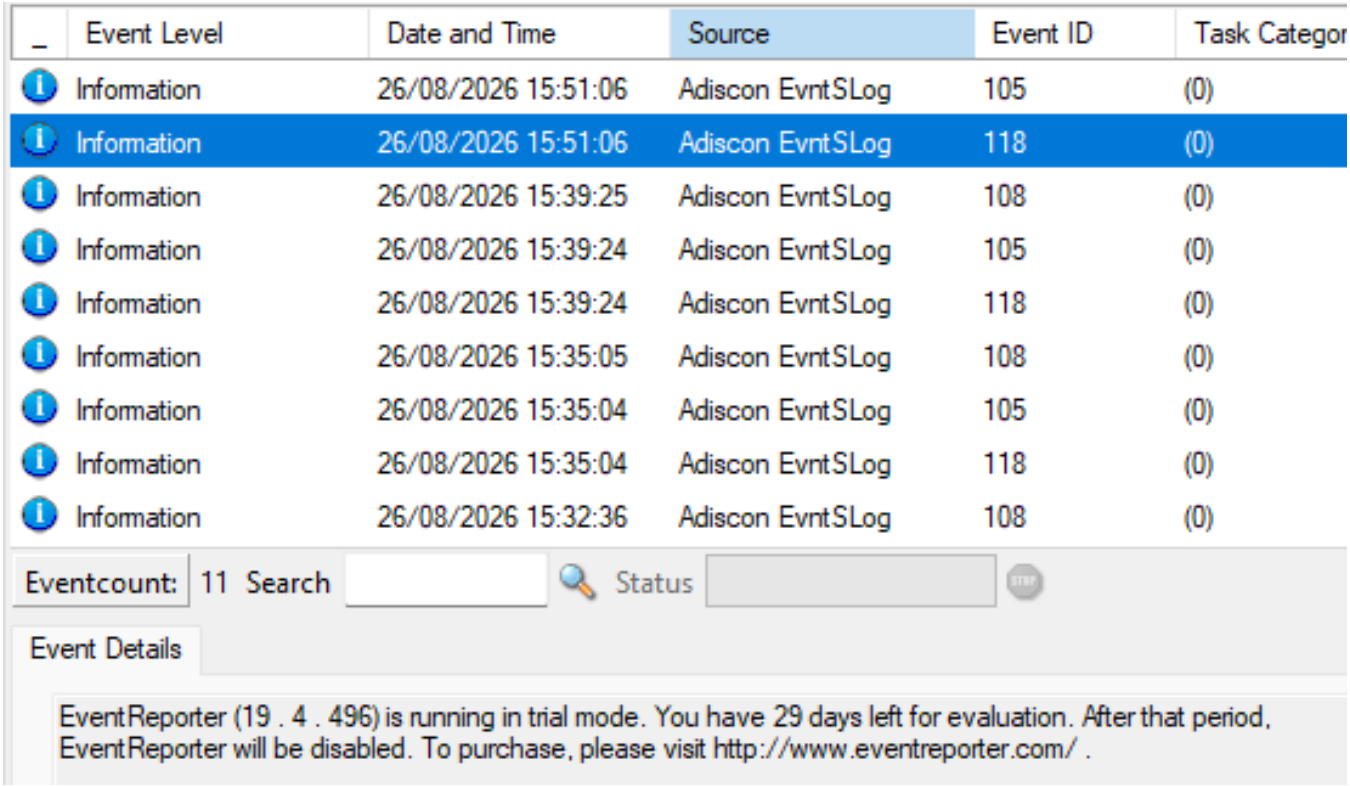


A second maintenance run can reinstall the selected product files.



Files or services that remain in use can cause the installer to request a reboot.

After a successful file refresh, the client and service report the newer versions.



Verify the running product after the installer finishes.

Action path

1. Export or back up the rsyslog Windows Agent configuration before changing the installation.

2. Close the rsyslog Windows Agent Configuration Client and stop the rsyslog Windows Agent service manually before each setup run. Stopping the service first can avoid a reboot request caused by locked service files, but it cannot avoid every reboot request.
3. If a staged update is practical, use a tested intermediate release, such as 6.x to 7.x to 8.x, and verify the client, service, features, and configuration after each hop. Do not assume that an adjacent version path is safe without verification.
4. If the installation remains inconsistent, use the clean-install procedure for rsyslog Windows Agent after exporting the configuration.
5. If you are already working from the folder that contains the installer, run the following as two separate elevated setup runs.

   First run: mark every setup feature as locally installed.

   ```
   rsyslogwa.exe /v"ADDLOCAL=ALL /L*v %TEMP%\rsyslogwa-addlocal.log"
   ```

   Second run: force a refresh of the installed files and registration data.

   ```
   rsyslogwa.exe /v"REINSTALL=ALL REINSTALLMODE=amus /L*v %TEMP%\rsyslogwa-reinstall.log"
   ```

   Let the wizard finish after each command. Do not combine the two property sets into one command for this recovery path. If the first command opens a resume page, continue through the wizard. After the second run, check Modify or Custom Setup, then verify the About dialog and the file versions before starting the service.

Related information

- **`err-2026-002-rsyslogwa-cross-generation-installer-upgrade`**
- Installation
- Microsoft Windows Installer ADDLOCAL property
- Microsoft Windows Installer REINSTALL property
- Microsoft Windows Installer REINSTALLMODE property
- Microsoft Windows Installer system reboots

How Can I Copy an rsyslog Windows Agent Configuration to Another System?

Answer

Export the configuration from the source system, import it on the target system, then review host-specific settings before applying it.

Details

The safest method is to export the configuration as a text-based registry file from the Configuration Client and import it on the destination system.

Before applying the imported configuration, review:

- target host names or IP addresses
- TLS certificate paths
- service account or path differences
- local file paths used by debug logs or queues

Action path

1. Export the configuration from the source system.
2. Transfer the exported file to the destination system.
3. Open the rsyslog Windows Agent Configuration Client on the destination system.
4. Import the configuration file.
5. Review environment-specific settings.

6. Save and apply the configuration.
7. Restart the service if required.

Related information

- How to Export rsyslog Windows Agent Settings for a Support Call
- Tutorial: Export the Configuration and Create a Debug Log

How Can I Obtain a Printable Version of the rsyslog Windows Agent Manual?

Answer

You can use the online rsyslog Windows Agent manual in your browser or download its printable PDF version from the manual site:

rsyslog Windows Agent manual

Details

The online manual is usually the best choice for browsing, searching, and following links between related topics. If you need a printable copy for offline review or internal distribution, use the PDF version available from the manual site.

If rsyslog Windows Agent is already installed, local help is also included with the installation, so an additional download is often unnecessary.

Related information

- Installation

Protocols and integration

What do CA PEM, Certificate PEM, and Key PEM mean for TLS input services?

Question

What do the `CA PEM`, `Certificate PEM`, and `Key PEM` fields mean for TLS-enabled input services such as `RELP Listener` or `SETP Server`?

Answer

For TLS-enabled input services:

- `CA PEM` is the CA bundle used to validate peer certificates.
- `Certificate PEM` is the certificate presented by the receiving service.
- `Key PEM` is the private key that matches that certificate.

The private key must be in PEM format and must not be protected by a passphrase.

Details

The `CA PEM` field is used when certificate validation is enabled for incoming TLS connections. The receiving service uses the certificates in this file to validate certificates presented by connecting clients.

The `Certificate PEM` field should contain the certificate that the product presents to connecting clients. If intermediate CA certificates are required so that clients can build the chain, include the service certificate first and then append the intermediate CA certificates in order toward the root CA.

The `Key PEM` field must contain the private key that belongs to that certificate. Use an unencrypted PEM key file. Passphrase-protected private keys are not supported in these fields.

If client certificates are issued through a CA chain, the `CA PEM` file can contain multiple CA certificates. Include the intermediate CA certificates first, followed by the root CA certificate.

Action path

1. Decide whether the input service should validate client certificates.
2. Prepare a PEM CA bundle for `CA PEM` if certificate validation is used.
3. Prepare a PEM certificate file for `Certificate PEM`. Include the intermediate chain if clients need it to validate the certificate.
4. Prepare the matching PEM private key for `Key PEM` and remove any passphrase protection before using it.
5. Load the files into the TLS configuration fields and test the connection.

Related information

See the product-specific TLS service reference page for the exact UI location of these fields.

Licensing and purchasing

How Do I Enter rsyslog Windows Agent License Information?

Answer

Apply the `license.alic` file through **General** -> **License** in the configuration client, save, and restart the service.

Details

Current rsyslog Windows Agent builds require a `license.alic` file from Adiscon.

Default file location:

`%ProgramData%\Adiscon\RSyslogAgent\license.alic`

Use the default location unless Adiscon support or your deployment process requires another path.

Action path

1. Obtain `license.alic` from Adiscon for your edition.
2. Open the rsyslog Windows Agent Configuration Client.
3. Expand **General** and select **License**.
4. Browse for `license.alic`, drag-and-drop the file, or paste the file path.
5. Save the configuration and restart the service.

Related information

- Installation
- How do I contact Adiscon sales?

Why did product versions jump to 26.07?

Question

Why is the current version `26.07` when older product versions used numbers such as 18, 19, 15, or 8?

Answer

Adiscon changed to calendar-based version numbers. In `26.07`, `26` means 2026 and `07` means July. There are no missing intermediate product releases.

Details

The latest old-style version families were:

- **WinSyslog**: `18.x`
- **EventReporter**: `19.x`
- **MonitorWare Agent**: `15.x`
- **rsyslog Windows Agent**: `8.x`

The new `YY.MM` format starts with `26.07`.

Related information

- See the product website for version history and release notes.

What happens when the client opens a configuration with unsupported blocks?

Question

The configuration client shows a warning about unsupported configuration items when I open a file. What does that mean?

Answer

The client skipped configuration blocks that this client version does not understand. Saving the configuration may remove those unsupported sections.

Details

Newer service or file formats can introduce configuration blocks that a configuration client cannot edit. The client loads what it supports and displays a one-time warning.

Current configuration clients preserve an existing `general(name="Metrics")` block when they load and save CFG, YAML, or registry configuration. They do not provide a Metrics editor. Other unknown blocks can still be skipped and may be lost when the configuration is saved.

If you save without upgrading the client, unsupported blocks may be dropped from the saved file.

Action path

1. Note which blocks the warning mentions.
2. If you need to edit those settings, upgrade to a configuration client that supports them, or edit the YAML or legacy `.cfg` file with care in a text editor.
3. Back up the configuration before saving from an older client.

Related information

- See the file-based configuration chapter in the WinSyslog, EventReporter, MonitorWare Agent, or rsyslog Windows Agent manual.

Platform and compatibility

Do the configuration clients require .NET Framework, or is .NET Core or .NET 5+ enough?

Question

Can the Windows configuration clients run with only the newer .NET runtime installed, such as .NET Core, .NET 8, or .NET 10?

Answer

No. The Windows configuration clients require **Microsoft .NET Framework 4.7.2 or a newer .NET Framework 4.x release**.

`.NET Framework 4.x` is a Windows-only runtime family. `.NET Core` and `.NET 5+` are a different cross-platform runtime family.

`.NET 5+` continues the `.NET Core` line under a new name. It is not a newer version of `.NET Framework 4.x`.

Installing only `.NET Core` or `.NET 5+` does not satisfy a `.NET Framework` requirement.

Details

For these products, the requirement applies to the **Windows configuration client**. The background service is a separate component.

If you deploy only the background service on a target system, the Configuration Client is not required on that system. In that case, this `.NET Framework` requirement applies where the Configuration Client is installed and used, not to the service-only target.

The following satisfy this requirement:

- `.NET Framework 4.7.2`
- `.NET Framework 4.8`
- `.NET Framework 4.8.1`

The following do not satisfy this requirement on their own:

- `.NET Core 3.x`
- `.NET 5`
- `.NET 6`
- `.NET 8`
- `.NET 10`

Action path

1. Check whether the system has `.NET Framework 4.7.2` or a newer `.NET Framework 4.x` release installed.
2. If it is missing, install `.NET Framework` separately or allow the product installer to add the required Framework components.
3. If the system only has `.NET Core` or `.NET 5+`, do not assume that the configuration client can run.

Related information

- Microsoft documentation on .NET Framework versions and dependencies: https://learn.microsoft.com/en-us/dotnet/framework/install/versions-and-dependencies
- Microsoft documentation on installing .NET on Windows: https://learn.microsoft.com/en-us/dotnet/core/install/windows

Sales

This section answers sales and licensing questions for Adiscon products in a question-focused format.

Use this section for commercial topics (quotes, orders, licenses, renewals, and purchase orders). For technical troubleshooting and product support, use the FAQ and troubleshooting sections in the relevant product manual.

Quick actions

- I need to contact sales: How do I contact Adiscon sales?.
- I need to request a quote: What should I include in a quote request?.
- I opened a ticket and need to check delivery/notification behavior: What happens after I open a sales ticket?.
- I need help with purchase orders and billing data: How do purchase orders and billing requests work?.
- I need product licensing and ordering links: Licensing and ordering.
- I need to use an older product version with my license: Can I use an older product version with my license?.

- I need to confirm offline or air-gapped licensing behavior: Air-gapped environments.
- I need UpgradeInsurance policy details: UpgradeInsurance.

How do I contact Adiscon sales?

Short answer

Use the Customer Service System as the primary channel; email to `sales@adiscon.com` is also accepted and creates tickets.

Question

How do I contact Adiscon sales for quotes, ordering, licensing, or renewals?

Answer

Use the Customer Service System. This is the primary contact path for sales questions.

Details

- You can also send email to `sales@adiscon.com`.
- Emails to `sales@adiscon.com` create tickets automatically.
- The ticket portal is recommended because email delivery can be unreliable in both directions (customer to Adiscon and Adiscon to customer).
- If you do not see a response, this is usually a delivery or filtering issue, not an intentionally unanswered request.
- Choosing a sales category helps speed up handling. If a ticket is opened in the wrong category, it is reassigned internally.
- For language handling, see **`../contact-language-policy`**.

Technical support routing

For technical incidents, troubleshooting, and product behavior issues, use the support category in the same system or the FAQ and troubleshooting sections in your product manual.

Next action

Open a sales ticket in the Customer Service System or send email to `sales@adiscon.com` (emails create tickets).

What should I include in a quote request?

Short answer

You can start with a short request, and sales will guide you through missing details; providing more context upfront can speed up the first quote.

Question

What information should I provide to get an accurate sales quote quickly?

Answer

You can start with a short message like “I want to buy your tool.” Sales will follow up and guide you through any missing details.

Details

If you want a faster first quote

- Product name
- Edition (if known) or note that you need help selecting one
- Number of licenses or systems/devices to be covered
- Billing country
- Commercial contact email

Optional but speeds finalization

- Deployment type (for example production, test, or mixed)
- License term and UpgradeInsurance requirements
- Preferred purchase method (online order or purchase order)
- Company legal name and billing contact details
- Target purchase timeline
- Existing license, renewal, or expansion context

EU VAT note

Customers in EU member states are normally charged VAT. If your organization is eligible for VAT exemption, provide your official VAT ID during ordering or invoicing so the billing team can apply the exemption.

Optional copy/paste template

If you already have details, this template can speed up quote processing:

```
Product:
Edition (or "help me choose"):
Scope (licenses/systems/devices):
Deployment type (production/test/mixed):
License term / UpgradeInsurance needs:
Billing country:
Company legal name:
Purchase method (online order/PO):
Target purchase date:
Existing license context (if any):
VAT ID (EU, if applicable):
Contact email:
```

Next action

Open a sales ticket in the Customer Service System or send email to `sales@adiscon.com` (emails create tickets).

What happens after I open a sales ticket?

Short answer

Your ticket is confirmed by email and then handled in the same ticket thread, with first response usually within one business day.

Question

What should I expect after opening a sales ticket?

Answer

Adiscon sales reviews and responds in the same ticket thread, which is the primary place to track status and replies.

Normal flow

1. You open a ticket (via portal or email).
2. An automatic confirmation email is sent (if a valid address is available).

3. A sales agent replies in the same ticket thread.
4. Follow-up questions and updates continue in that same thread.

Details

Notification behavior

When a ticket is opened, an automatic response is sent by email to the registered address (if available).

If no automatic response arrives

Do the following:

- Spam or junk folders
- Mail security gateways
- Internal email policy filters

Then sign in to the Customer Service System to confirm whether your ticket was created.

Avoid duplicate tickets

If a ticket already exists, continue in that ticket instead of opening a duplicate request. Duplicate tickets can slow handling and split context.

When replying by email, keep the same subject/thread whenever possible so updates remain attached to the original ticket.

Next action

Check and continue the existing ticket in the Customer Service System. If no ticket exists, open a new one in the portal or send email to `sales@adiscon.com`.

How do purchase orders and billing requests work?

Short answer

Use the Customer Service System for PO processing, billing changes, and invoice requests; for enterprise procurement flows, describe your process and Adiscon will adapt as much as possible.

Question

How do I handle purchase orders, invoicing, and billing-data updates?

Answer

Use the Customer Service System for purchase order processing and billing requests. Include complete company and billing details to avoid order delays.

Details

Typical flow

1. Open a sales ticket in the Customer Service System.
2. Share PO and billing details in that ticket.
3. Sales confirms details and clarifies open points.
4. Order and invoice handling continues in the same ticket thread.

For larger organizations with their own procurement flow, open a sales ticket and describe your process. Adiscon will adapt as much as possible.

Purchase orders

Helpful information to include at the start:

- PO number (if already issued)
- Company legal entity
- Billing address
- Billing contact
- Product and scope

For formal purchase order requirements and additional details, see: Adiscon purchase orders.

Billing and invoice requests

For invoice details, billing data updates, tax/VAT information, and related order handling, contact sales via the Customer Service System.

Use the Customer Service System for billing-data changes both before and after order placement.

Security and language

- Prefer the ticket portal (including attachments) over plain email when sending sensitive billing data.
- For language handling, see **`../contact-language-policy`**.

Common requests

The sales team can also handle:

- Invoice copy and invoice re-send requests
- Billing address/contact changes
- VAT/tax-related billing updates

Next action

Open a sales ticket in the Customer Service System or send email to `sales@adiscon.com` (emails create tickets).

Licensing and ordering

Short answer

Use product ordering pages for direct purchase or open a sales ticket for edition guidance and licensing clarification.

Question

How do I order Adiscon products, and where do I clarify licensing questions?

Answer

Order through the product-specific ordering pages or via purchase order. If edition or licensing details are unclear, open a sales ticket first.

Details

Trial period

WinSyslog, EventReporter, MonitorWare Agent, and rsyslog Windows Agent provide 30-day trial functionality after installation. After that period, a valid license is required for full-feature operation.

License agreement (EULA)

The End User License Agreement (EULA) is shown during setup. If you need a copy or have licensing questions, contact the Customer Service System.

Product-specific links

Product	Pricing and ordering
WinSyslog	WinSyslog ordering page
EventReporter	EventReporter ordering page
MonitorWare Agent	MonitorWare Agent ordering page
rsyslog Windows Agent	rsyslog Windows Agent ordering page

Choosing an edition

If you are unsure which edition fits your requirements, open a sales ticket in the Customer Service System. A short request is enough to start.

Product-specific licensing details

Licensing and counting rules can be product-specific. For now, use the sales ticket process to clarify product-specific licensing details. These details will be documented in dedicated product-specific sales pages.

Offline and licensing behavior

For common licensing and deployment questions, see these canonical answers:

- Air-gapped environments
- Offline installation and activation
- Online verification after activation
- Perpetual licenses and UpgradeInsurance
- Can I use an older product version with my license?

Purchase methods

You can place orders through each product ordering page or by purchase order. If you need a formal quote before placing the order, open a sales ticket in the Customer Service System.

Maintenance and renewal

For maintenance and renewal policy details, see UpgradeInsurance.

Purchase orders and billing

For purchase order handling and billing details, see How do purchase orders and billing requests work?.

Next action

Open a sales ticket in the Customer Service System or send email to `sales@adiscon.com` (emails create tickets).

Can I use an older product version with my license?

Question

Can I use an older Adiscon product version with my current license?

Answer

Yes. Your license can cover older product versions.

If the older product version does not support license files, contact Adiscon support or sales and request a free license key for that product version.

Details

Current products use a `license.alic` license file. Older products that do not support license files use a registration name and license key instead. Tell Adiscon which product and version you need to run so the license can be issued in the right format.

Related information

- How do I contact Adiscon sales?
- Licensing and ordering

Air-gapped environments

Short answer

Adiscon products can be deployed and operated in fully closed environments without internet access.

Question

Can Adiscon products be used in completely closed or air-gapped environments?

Answer

Yes. Adiscon products support deployment and operation in fully closed networks.

Details

Products can be installed and operated in environments where internet access is not available or not permitted. They do not require outbound internet access for normal runtime operation in such environments.

If you also need details about license activation without internet access, see Offline installation and activation.

Action path

If your procurement or security process requires a written confirmation for a specific environment, open a sales ticket in the Customer Service System.

Related information

- Offline installation and activation
- Online verification after activation

Offline installation and activation

Short answer

Installation and license activation do not require internet access.

Question

Do Adiscon product installation and license activation require internet access?

Answer

No. Installation and license activation can be completed without an internet connection.

Details

Adiscon supports installation and license activation in offline environments. This allows products to be deployed in restricted networks where direct internet connectivity is not available.

This question is separate from whether the licensed product later requires online verification. For that, see Online verification after activation.

Action path

If you need product delivery or license handling guidance for an offline rollout, open a sales ticket in the Customer Service System.

Related information

- Air-gapped environments
- Online verification after activation

Online verification after activation

Short answer

After activation, Adiscon product licenses do not require ongoing internet connectivity, online heartbeats, or “phone home” checks to remain valid.

Question

Do Adiscon product licenses require online verification after activation?

Answer

No. After activation, the license remains valid without periodic online verification.

Details

Once the product is installed and the license has been applied, no ongoing outbound connection to Adiscon systems is required to keep the license active. There is no periodic online verification, heartbeat, or “phone home” mechanism needed to maintain licensed status.

This question is separate from whether installation and activation themselves require internet access. For that, see Offline installation and activation.

Action path

If your security or compliance process requires explicit confirmation about offline license behavior, open a sales ticket in the Customer Service System.

Related information

- Air-gapped environments
- Offline installation and activation
- Perpetual licenses and UpgradeInsurance

Perpetual licenses and UpgradeInsurance

Short answer

Adiscon product licenses are perpetual. UpgradeInsurance is a separate, time-limited service that is strongly recommended for access to version-line upgrades.

Question

Are Adiscon product licenses perpetual, and how does that differ from UpgradeInsurance?

Answer

Adiscon product licenses are perpetual. UpgradeInsurance is an additional, time-limited service for upgrade and maintenance benefits.

Details

A perpetual license allows continued use of the licensed product version after activation. It does not depend on periodic online verification to remain valid.

UpgradeInsurance is separate from the base license. It is a time-limited additional service that is strongly recommended if you want access to future product version lines and related maintenance benefits.

When UpgradeInsurance expires, the licensed product version does not stop working. What changes is entitlement to later version-line upgrades and the additional benefits covered by active UpgradeInsurance.

License files: Current product versions use a `license.alic` file from Adiscon.

For detailed renewal and reinstatement policy, see UpgradeInsurance.

Action path

If you need confirmation for purchasing, renewal, or compliance review, open a sales ticket in the Customer Service System.

Related information

- UpgradeInsurance
- Can I use an older product version with my license?
- Online verification after activation

UpgradeInsurance

Short answer

UpgradeInsurance provides version-line upgrade rights and priority support while active; if coverage has lapsed, reinstatement is possible with back-dated start rules.

Question

What is UpgradeInsurance, and how do renewal and reinstatement work?

Answer

UpgradeInsurance is Adiscon’s maintenance plan.

Details

What it includes

- Free version-line upgrades during active coverage
- Priority support handling

Coverage period

UpgradeInsurance is available for periods between 1 and 5 years.

Coverage continuity and reinstatement

- UpgradeInsurance should be kept active through timely renewal.
- If coverage lapses, reinstatement is possible.

- If UpgradeInsurance was not purchased with the original license, it can still be added later.
- Reinstatement is back-dated to the previous coverage end date or, if it was never purchased, to the original license purchase date.

Difference from the base license

UpgradeInsurance is separate from the base product license. For the distinction between perpetual licenses and this time-limited service, see Perpetual licenses and UpgradeInsurance.

Examples

1. If coverage ended on June 30 and renewal is requested in September, reinstatement starts from June 30.
2. If no UpgradeInsurance was purchased with the initial order, later activation is back-dated to the original license purchase date.

How to request renewal or reinstatement

Open a sales ticket in the Customer Service System or send email to `sales@adiscon.com` (emails create tickets).

Next action

Open a sales ticket in the Customer Service System or send email to `sales@adiscon.com` (emails create tickets).

Reference

Use this section for lookup material, not for first-time setup. It collects the pages you typically need while operating, troubleshooting, or fine-tuning an existing rsyslog Windows Agent deployment.

Use `Getting Started` for first-time setup, `Tutorials` for task guidance, and `Configuration` for the main product setup pages.

Product controls and local operation

rsyslog Windows Agent Shortcut Keys

Use shortcut keys as an alternative to the mouse when working in rsyslog Windows Agent.

Keyboard shortcuts may also make it easier for you to interact with rsyslog Windows Agent. All these shortkeys 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 rsyslog Windows Agent from the command line. To use these switches you need administrative rights.

- `-h` Show command line help
- `-v` Show version information and whether the service is installed
- `-i` Install service
- `-u` Remove (uninstall) service
- `-i <CustomServiceName>` Install service with a custom service name
- `-u <CustomServiceName>` Uninstall a service with a custom service name
- `-r` Run as console application

- `-r -o` Run once as console application

If you install the service, you can start and stop it with commands such as `net start`, `net stop`, `sc start`, `sc stop`, or PowerShell (`Start-Service` / `Stop-Service`). 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.

Direct foreground configuration

This capability is available starting with **26.08**.

Foreground runs can use an ephemeral configuration without installing a service or creating a service `Parameters` key:

- `-r -fileconfig <path> [-datadir <path>]`
- `-r -accessmode file -fileconfig <path> [-datadir <path>]`
- `-r -accessmode registry [-regpath <path>]`

Direct options take precedence for that foreground process only. File mode requires `-fileconfig`; `-datadir` cannot be used alone or with explicit registry mode, and `-regpath` is available only with explicit registry mode. The existing `-i` installation configuration remains persistent and unchanged.

The first direct foreground run registers the canonical Windows Event Log provider using the executable message file. This does not create a service. Provider creation or an update can require one elevated first run; the command reports that requirement before startup.

Custom service name examples:

- `rsyslogd.exe -i CustomServiceName`
- `rsyslogd.exe -u CustomServiceName`

You can import configuration files via the command line as well. Use YAML (`.yaml`) for new imports. The legacy Adiscon Config Format (`.cfg`) remains supported for compatibility with existing files and support workflows. The syntax is quite easy: execute the rsyslog Windows Agent configuration client and append the name of the configuration file.

Sample:

RsyslogConfigClient.exe example.yaml

or

RsyslogConfigClient.exe “example.yaml”

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` parameter has to be appended. This will look like this:

```
RsyslogConfigClient.exe "example.yaml" /f
```

Edition Comparison

rsyslog Windows Agent is available in multiple editions. The manual describes the full feature set, while edition-specific licensing determines which features are available in a given deployment.

For Event Log Monitor deployments, licensing follows the same model as EventReporter. A license is required for each source system whose Windows Event Logs are collected or forwarded. This applies to physical systems and virtual machines alike.

Use the official edition comparison page to confirm which services, actions, and capabilities are included in the edition you plan to deploy:

rsyslog Windows Agent edition comparison

License file: Current builds use a `license.alic` file.

Technical lookup material

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%`

minlength=N

Pads the transformed output on the right with spaces until it contains at least `N` characters. For example, `%msg:1:3:minlength=10%` emits the first three message characters followed by seven spaces. Options compose: `%msg:1:3:uppercase,minlength=10%`.

* = 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:

`%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%,%syslogprio rity%,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.

nteventlogtype

The name of the Windows Event Log this event is from (for example “System” or “Security”).

channel

The channel property for event log entries, for classic Event logs they match the `%nteventlogtype%` property, for new event logs, they match the “Event Channel”.

sourceraw

This contains the full internal name of the event source for new event logs, for classic event logs it contains the same value as in %sourceproc%.

level

Textual representation of the 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

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:

```
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  = "DUPADDRES") OR (msg = "SPANTREE") OR (msg = "DUPLEX_MISMATCH) then
    execute action(s)
end if
```

Please note: in the example, we have abbreviated “message” to just “msg”. Also note that for brevity reasons we use the equals (“=”) comparison operator, 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:

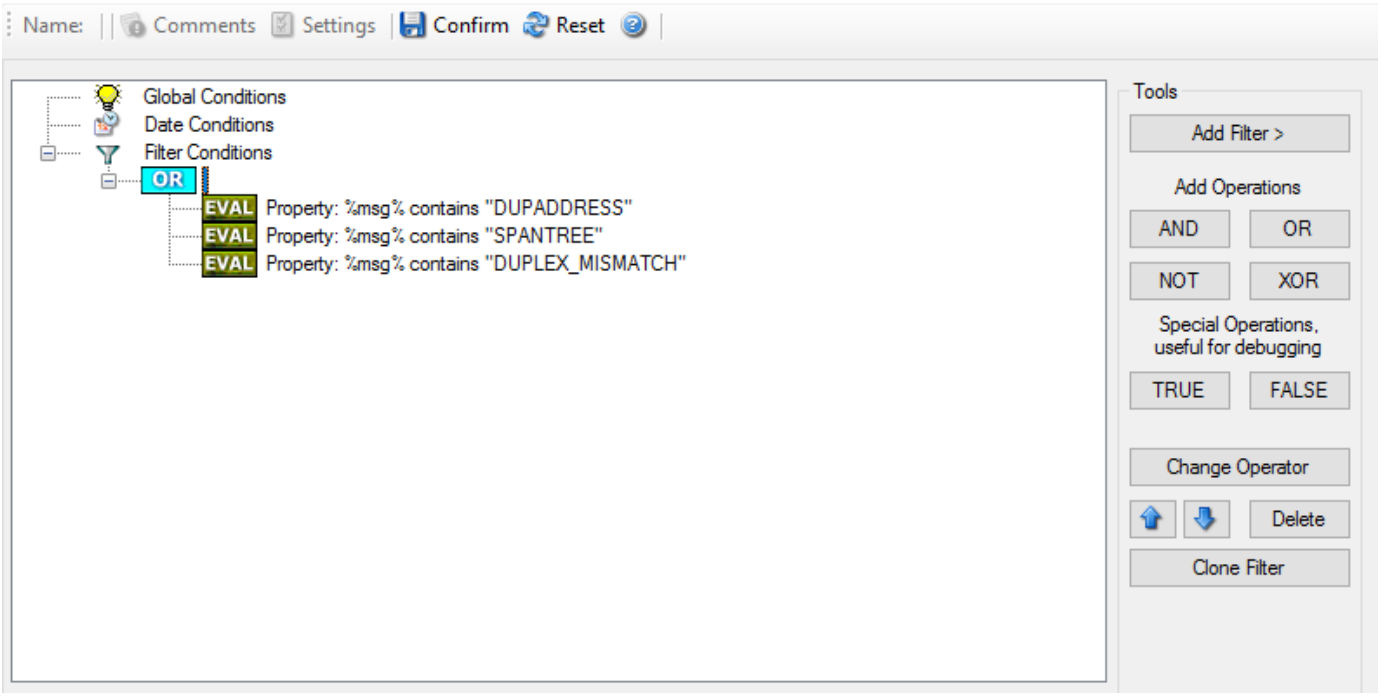


Figure 1 - Example 1

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  = "DUPADDRES") AND (msg = "SPANTREE") AND (msg = "DUPLEX_MISMATCH) then
    execute action(s)
end if
```

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

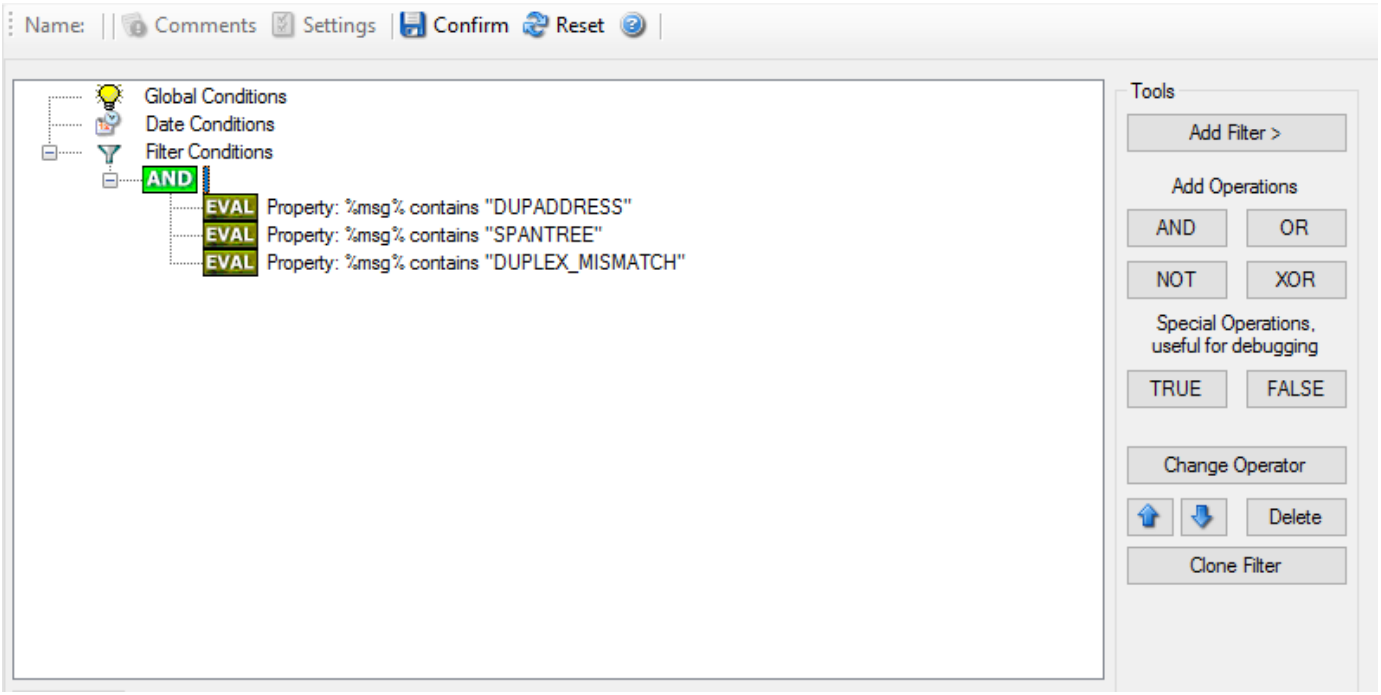


Figure 2 - Example 2

Example 3

This example is a bit more complex version of example 1. Again, the same message text filtering is done, that is if any one of the provided substrings is present, the filter eventually evaluates to true. To do so, the source system must also contain the string “192.0.2”, which can be used to filter on a device from a 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:

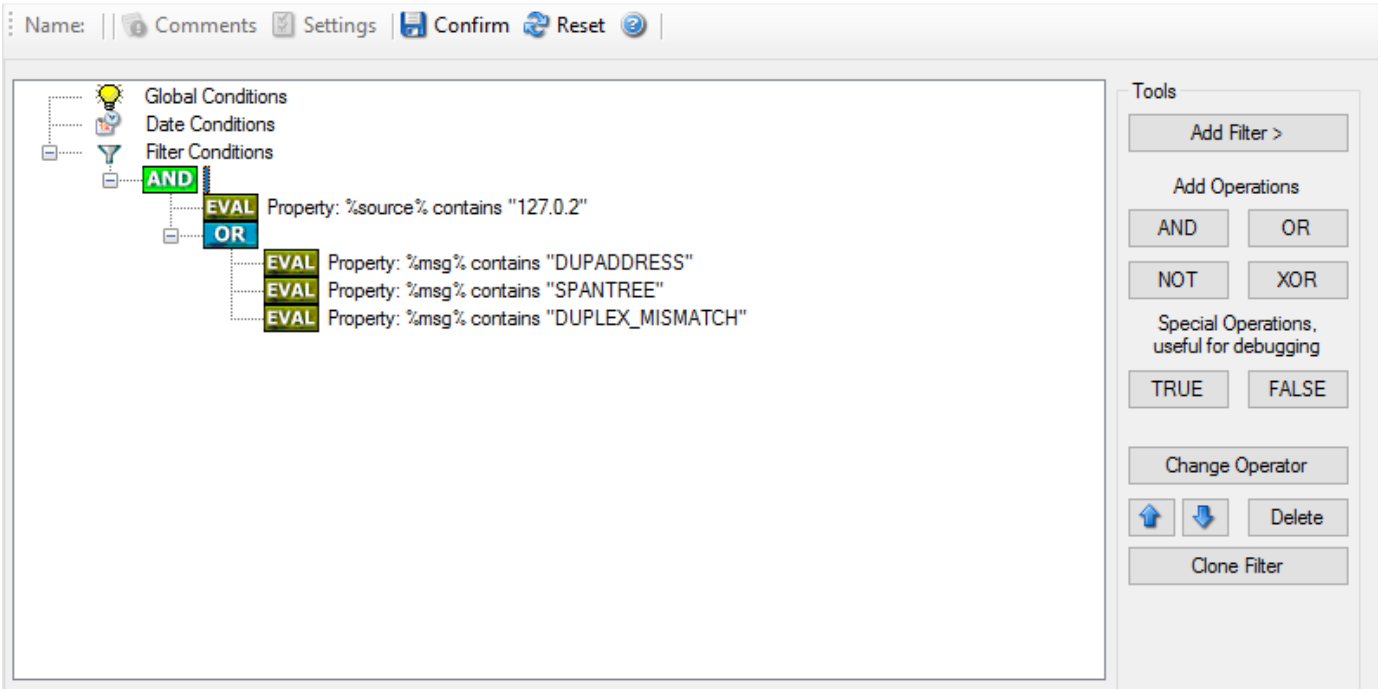


Figure 3 - Example 3

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

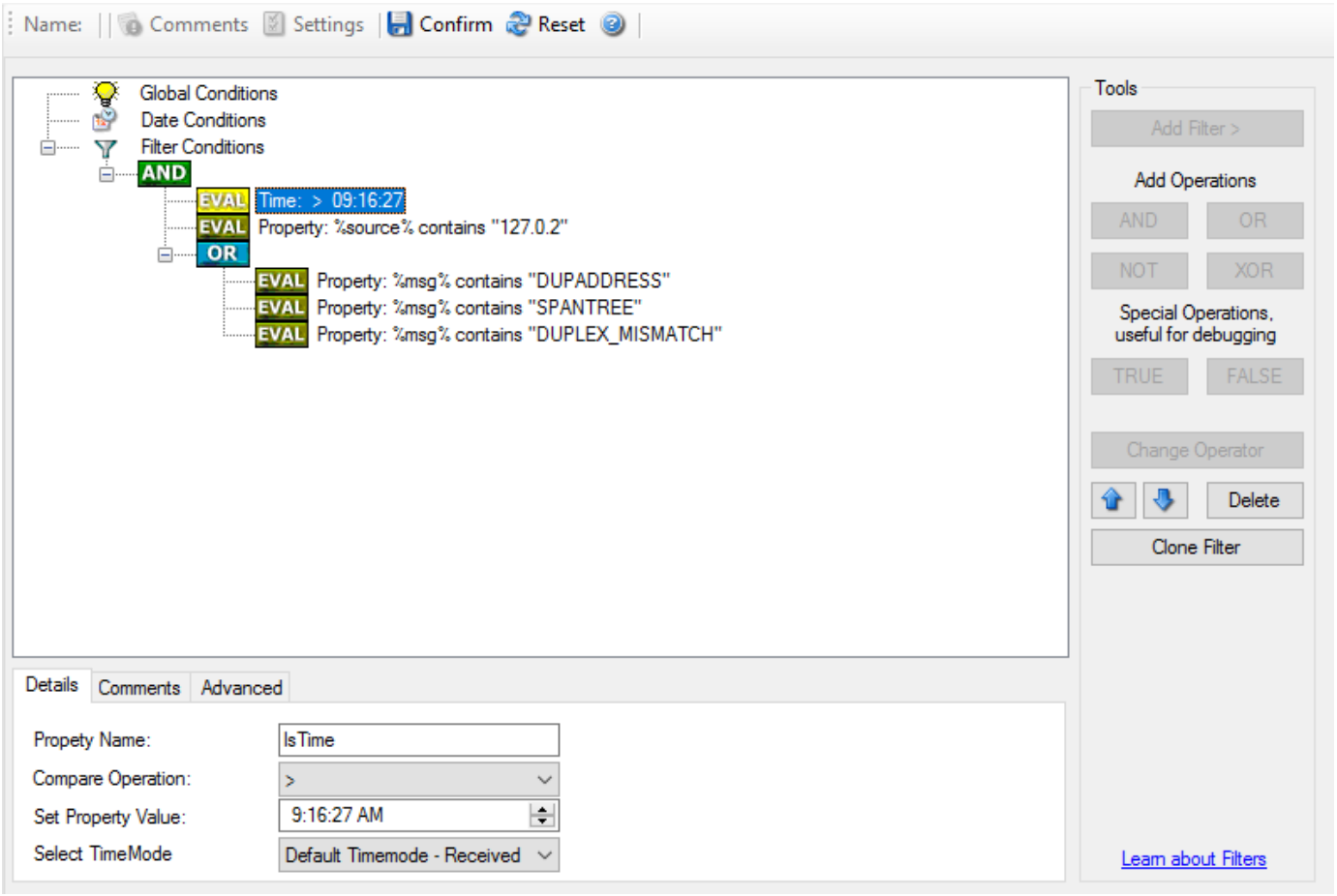
Example 4

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.

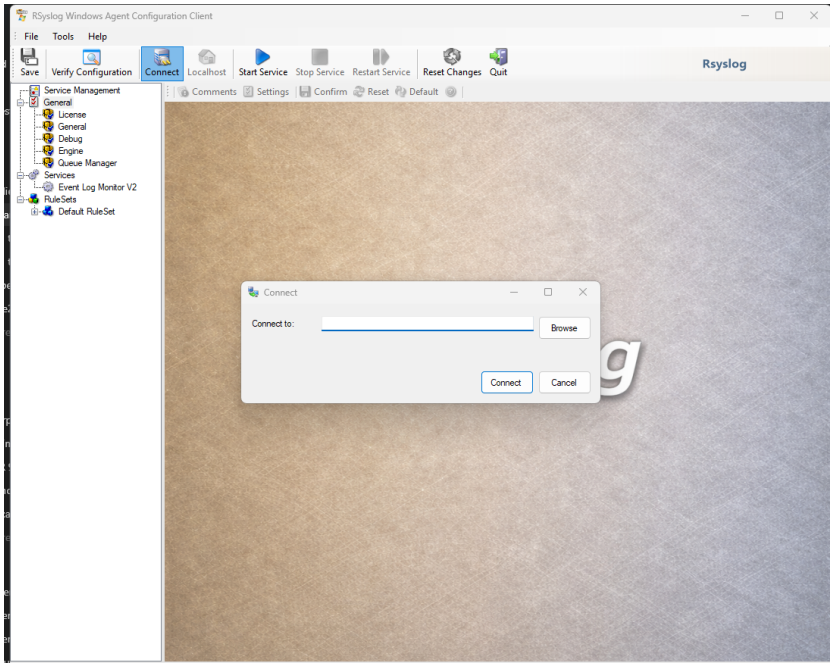
In pseudo-code, the filter could be written like this:

```
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:

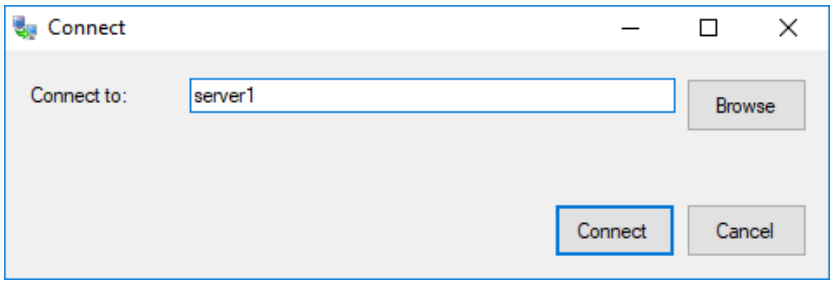


Connect to Computer



rsyslog Windows Agent Connect to Computer window

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.

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.

User Interface changes

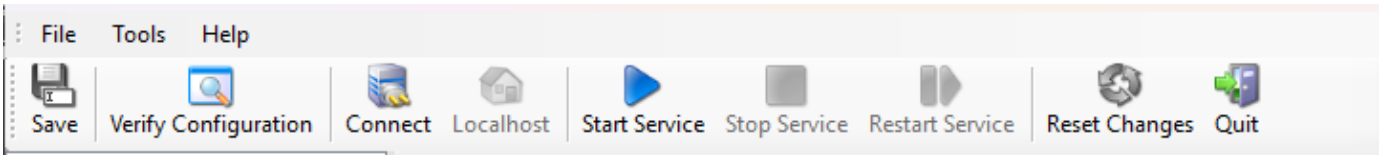
The **26.07** installer includes the established configuration client and the new user interface for this product:

- **Established configuration client** - the WinForms client documented in this manual and the primary reference for setup steps and screenshots.
- **rsyslog Windows Agent User Interface** - the newer WinUI-based interface (`adiscon-client-ng`) available alongside the established configuration client.

The established configuration client supports license file application and full configuration editing. See your product website's **Upgrade** article for the customer-facing description of the product-specific user interface.

Property window toolbar

When you open a service or action property window, the toolbar provides **Save**, **Verify Configuration**, **Connect**, **Start**, **Stop**, **Restart**, **Reset Changes**, and **Quit**. **Reset Changes** discards unsaved edits on the open form only; it does not restore factory defaults for the selected item.

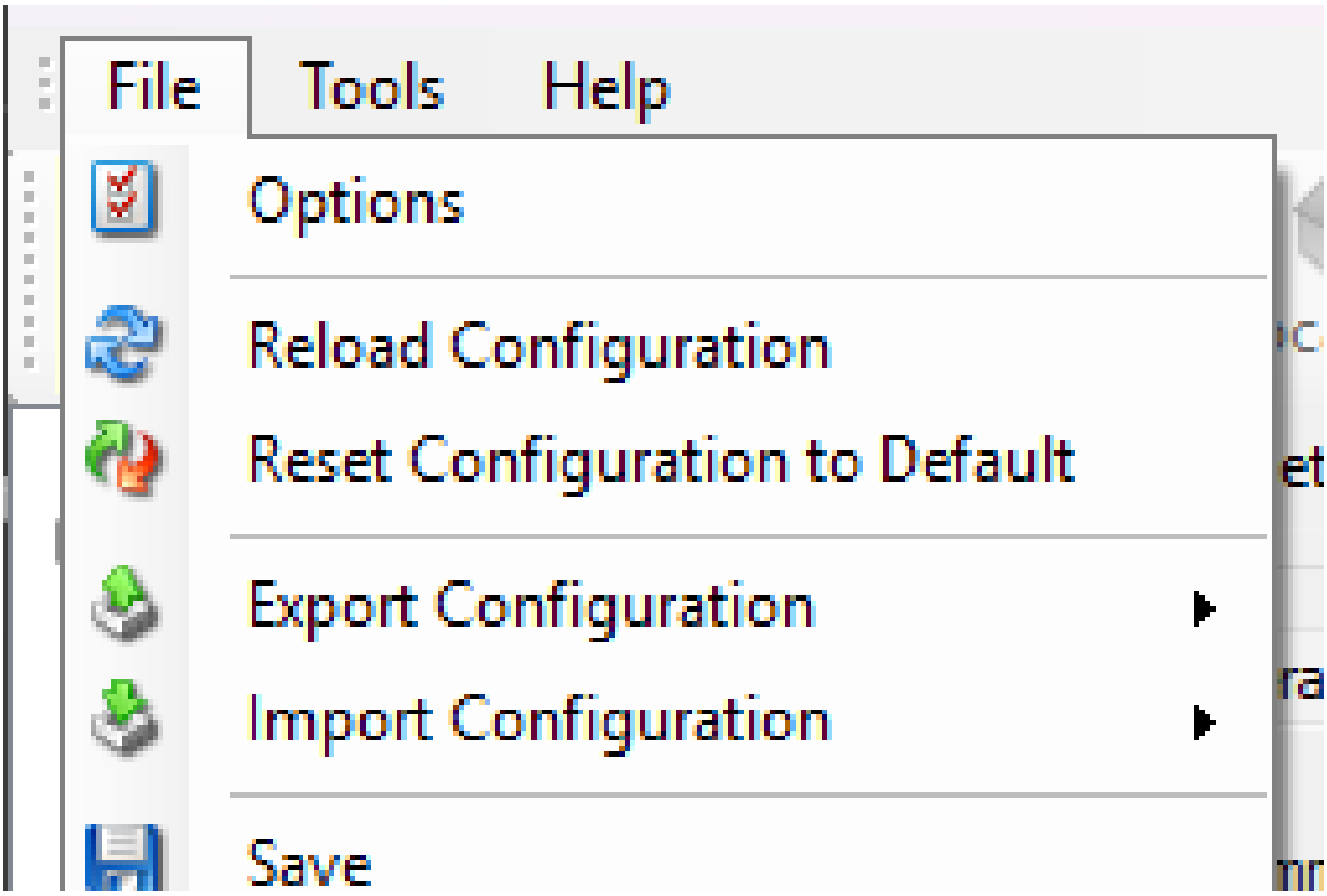


The **main window** (rules tree) has a separate toolbar with **Start**, **Stop**, **Restart**, **Kill**, and **Open Windows Services** for the background service. **Kill** ends an orphan process when the service manager state is inconsistent. **Open Windows Services** opens `services.msc` for Windows Service management.

When the service is **paused** during configuration reload, stop and restart may be disabled until reload completes.

Reset configuration to default

The **File** menu provides **Reset Configuration to Default**, which resets the whole deployed configuration:



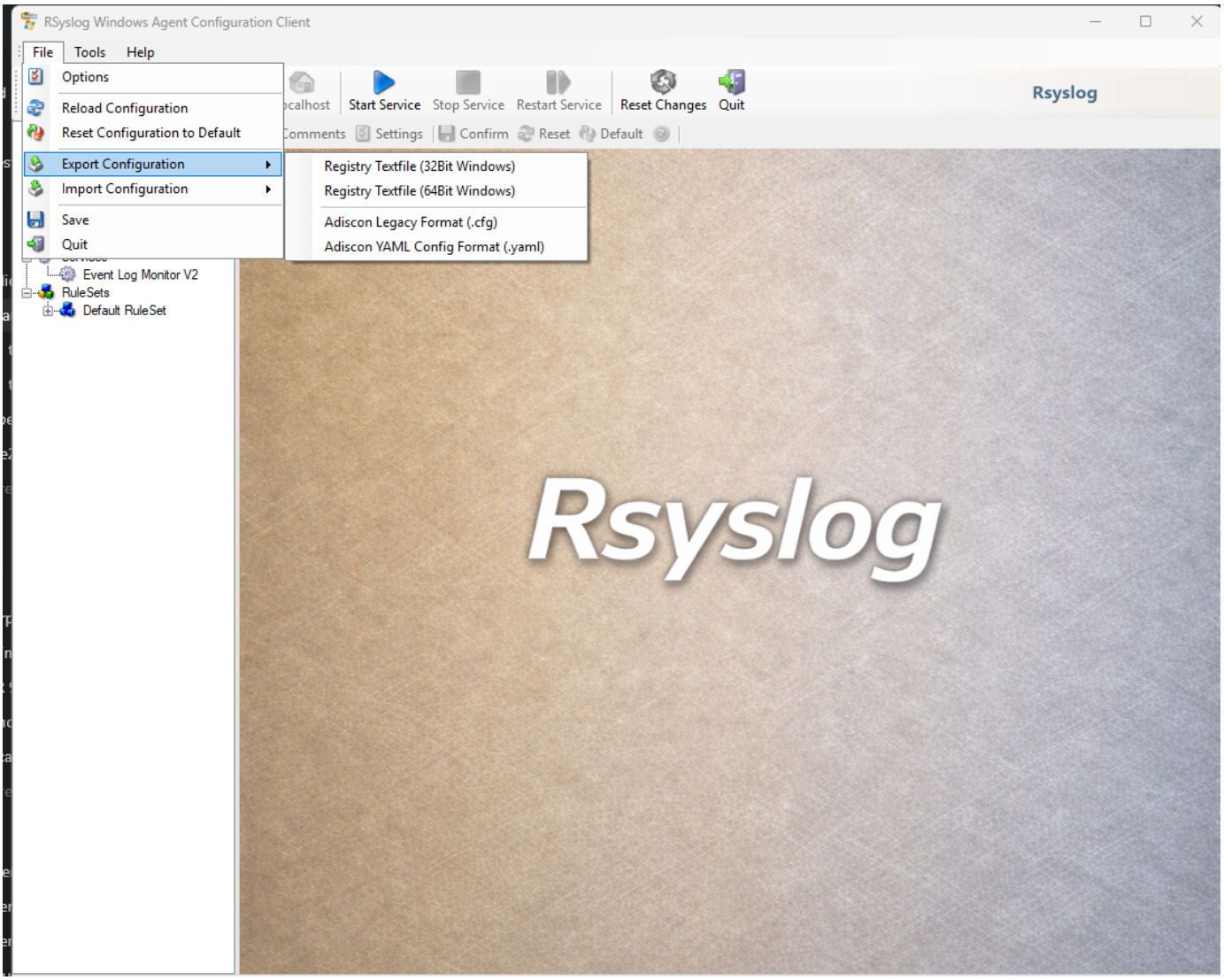
This is separate from **Reset Changes** on an open property form. It is also separate from the context menu or toolbar **Reset** command for a selected service or action, which restores only that selected item to factory defaults.

Event Viewer

After start, stop, or configuration reload, service-management events are prepended in the Event Viewer so recent control actions appear at the top.

File formats

The **File** menu supports:



rsyslog Windows Agent File menu with YAML import and export commands

- **Adiscon YAML Config Format** (`.yaml`) - import and export
- **Adiscon Legacy Format** (`.cfg`) - legacy text configuration

See the file-based configuration chapter in your product manual and What happens when the client opens a configuration with unsupported blocks?.

Related information

- Why did product versions jump to 26.07?

Glossary

Use this section to look up terms that appear throughout the rsyslog Windows Agent manual.

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.

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.

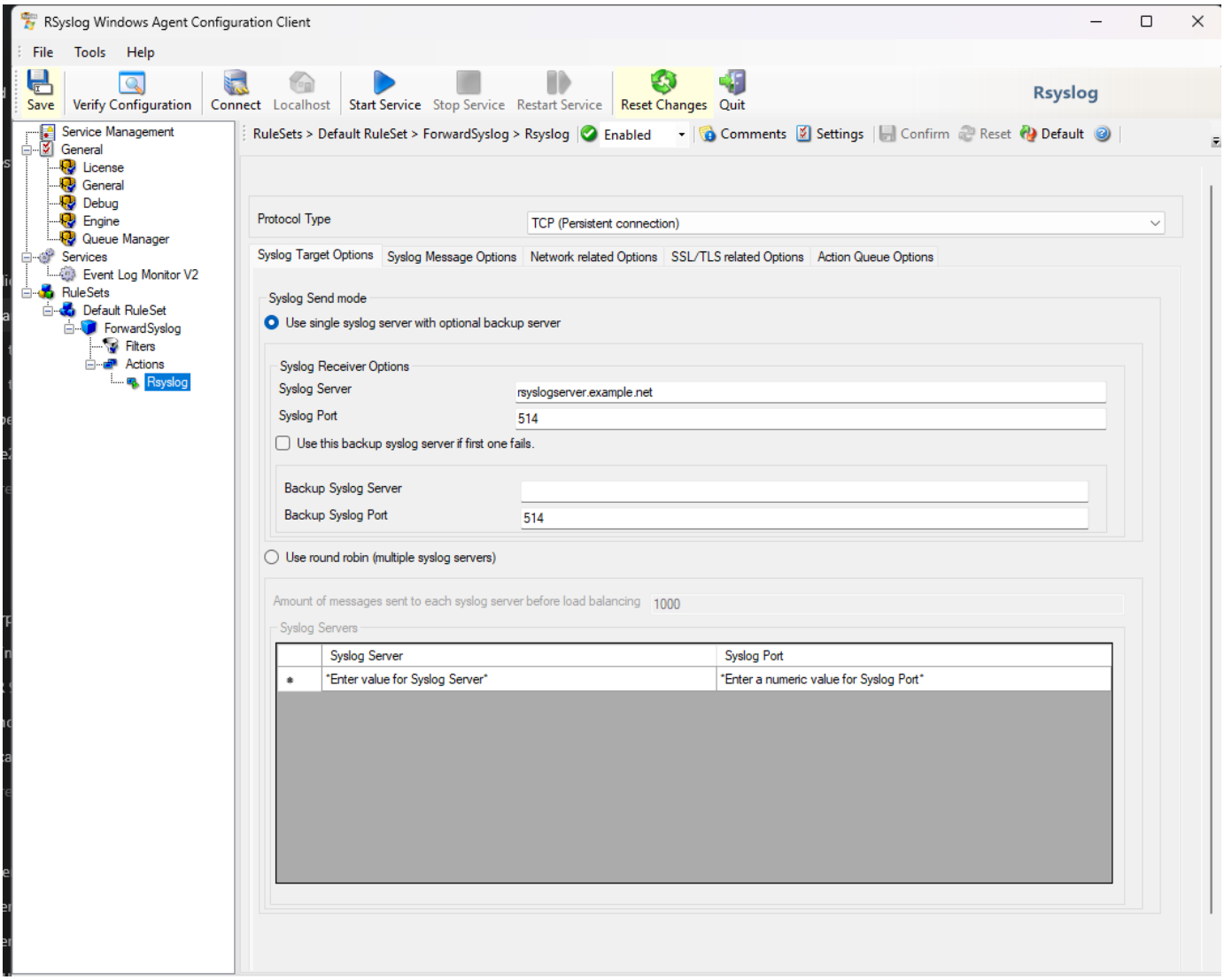
The rule base consists of the rules as configured in the client. The rule engine is the process carrying out the rules. A rule base can contain no, one or an unlimited number or rules. However, if there is no rule at all defined, no action will ever be carried out by the agent. Consequently, the client 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 the 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:



rsyslog Windows Agent sample rule base

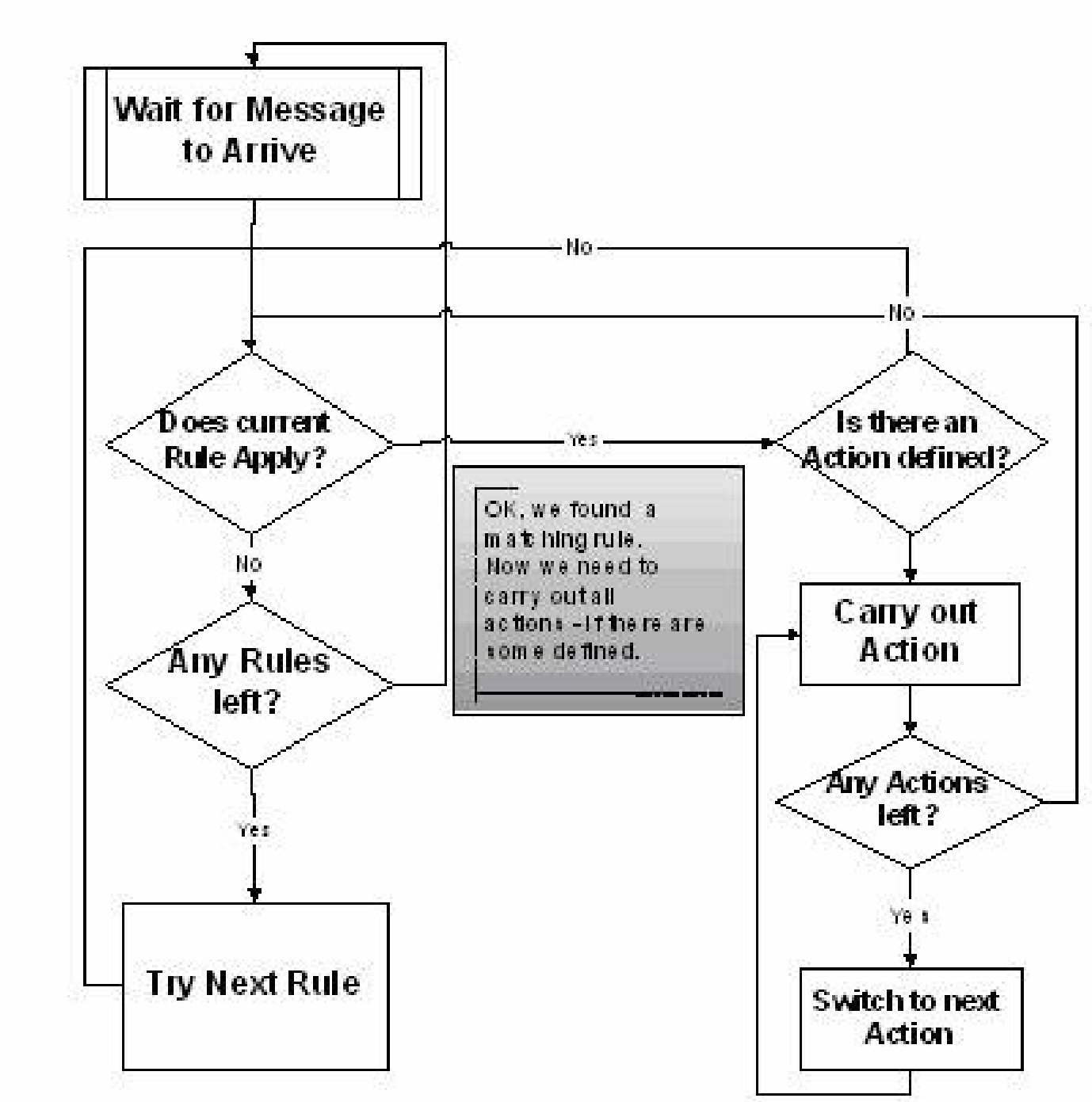
Now that we know the elements, how are rules being processed. It is easy. Rules are strictly processed from top to bottom, or from number one to the last one. For each rule the filter conditions are checked to see if they match. If they do, all associated actions are carried out. Then, the rule engine 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:

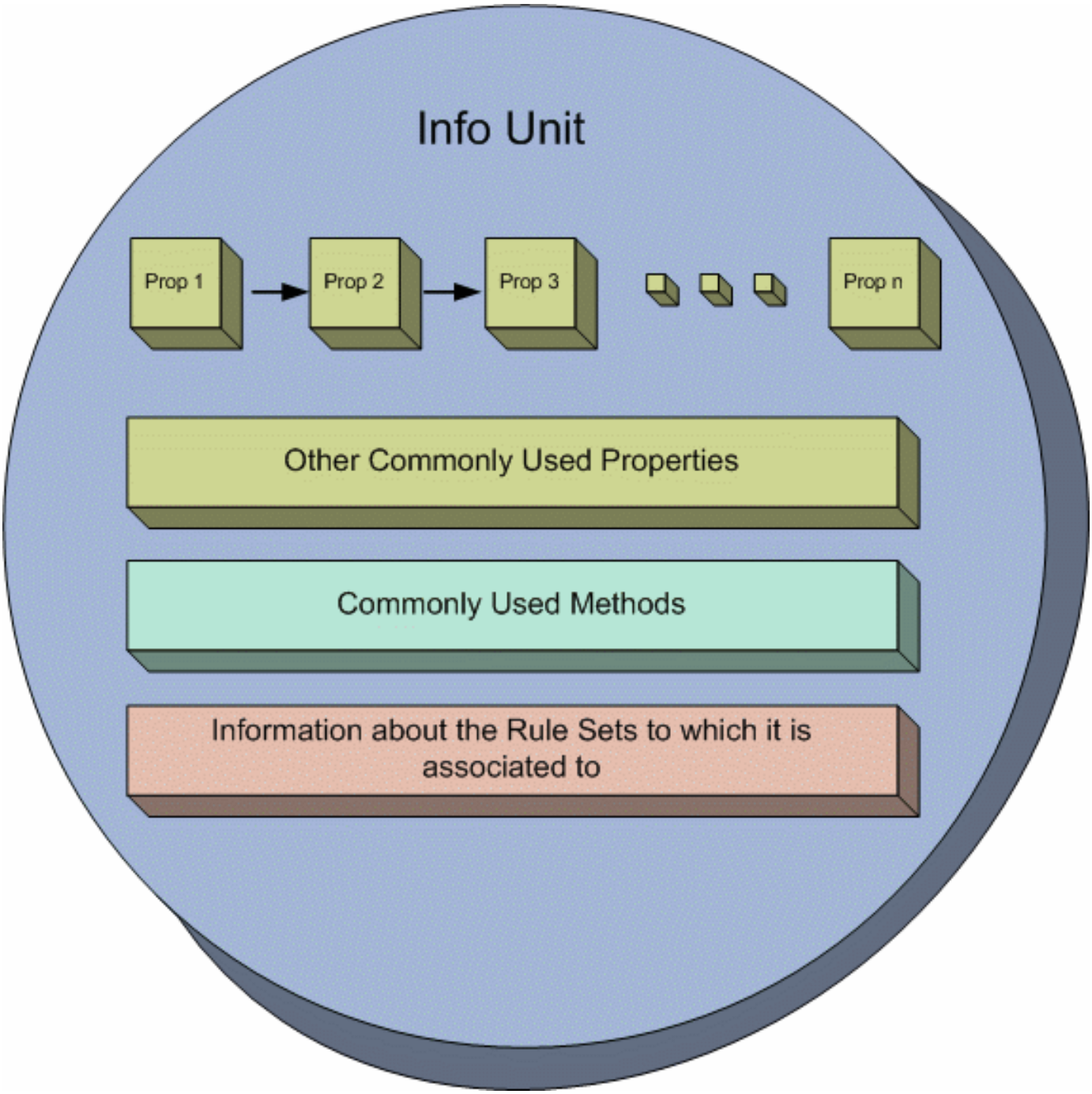


Figure 1: Conceptual Diagram of an Info Unit

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

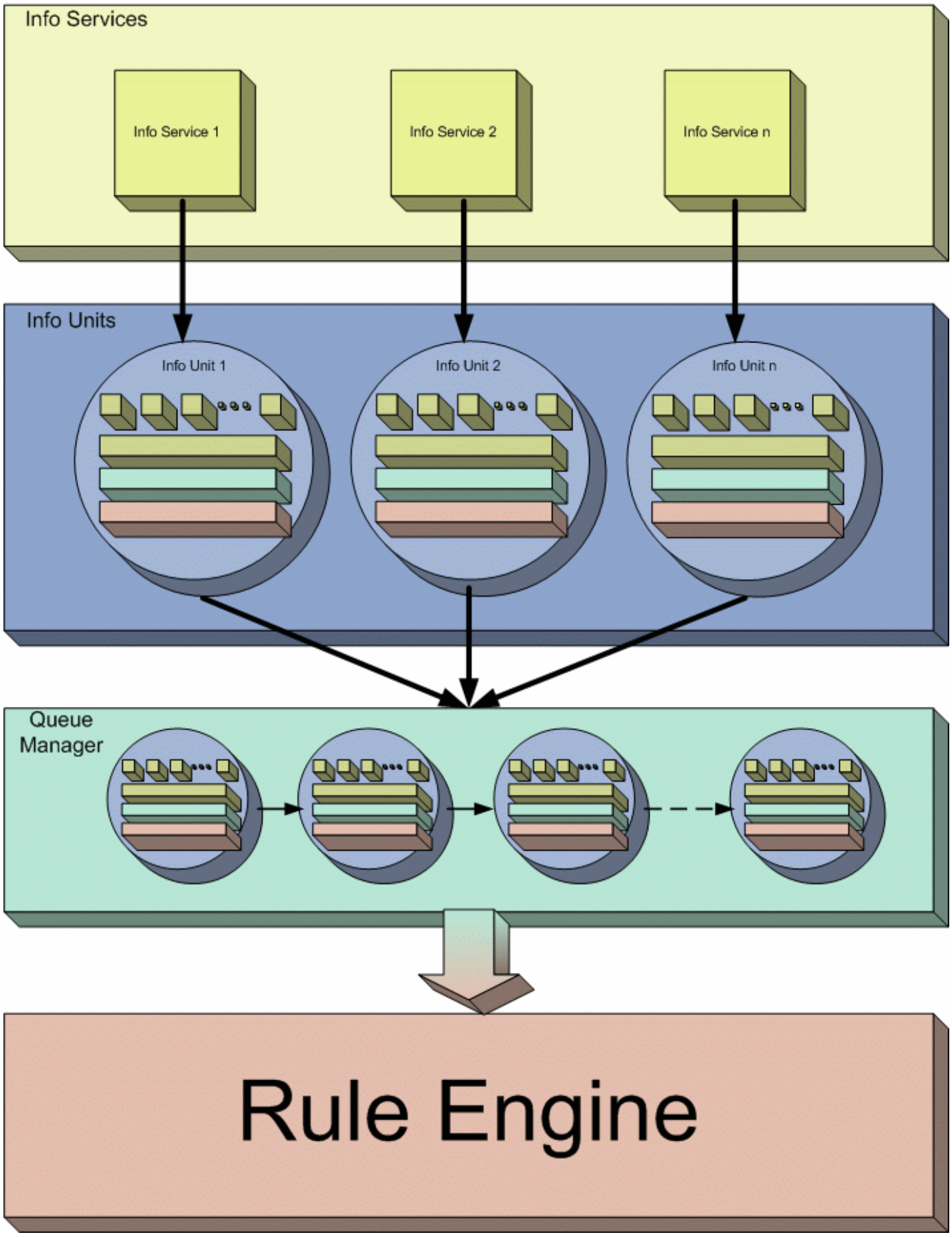


Figure 2: Overall Process

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:

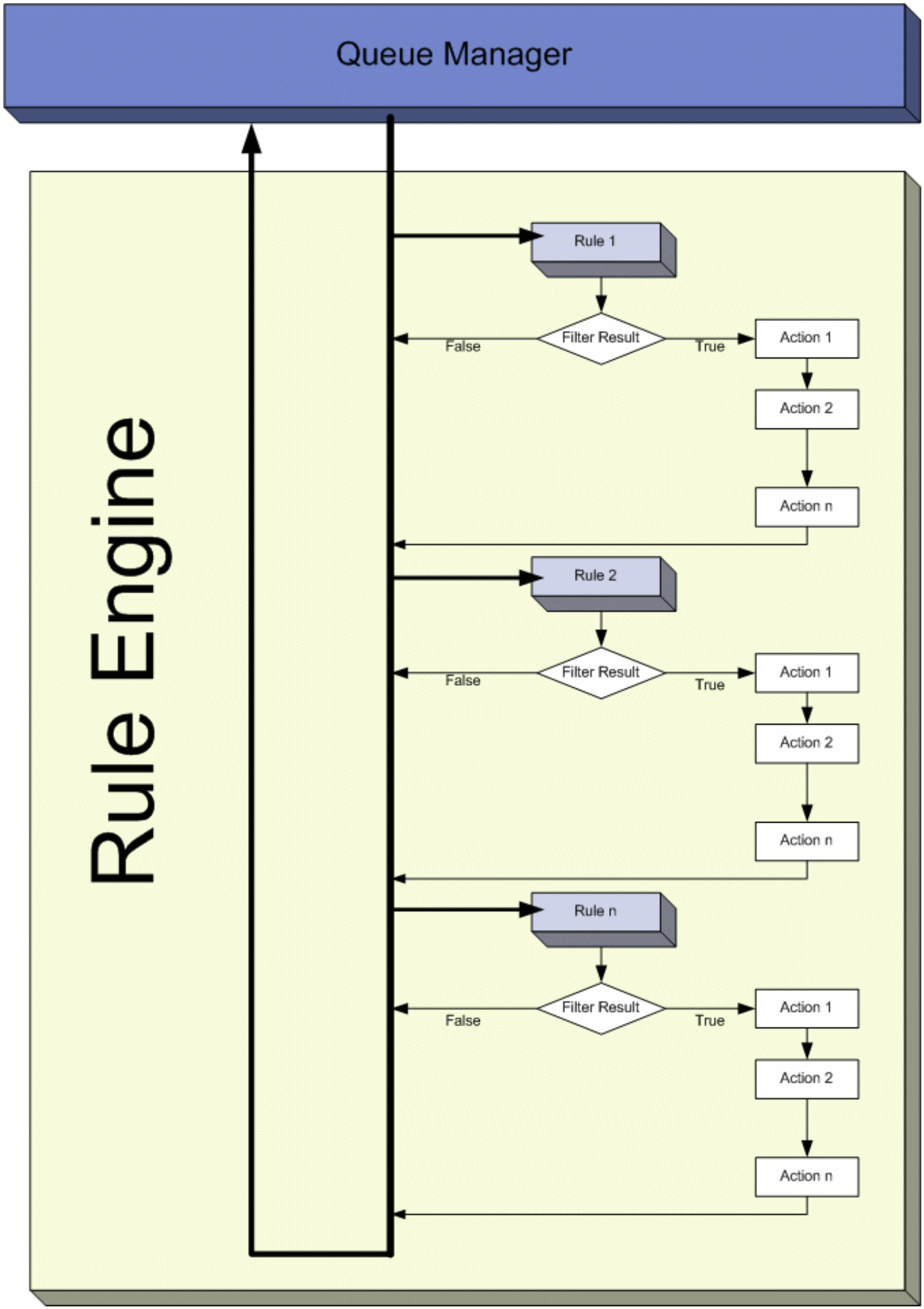

Queue Manager
Rule Engine
Rule 1
Filter Result
False
True
Action 1
Action 2
Action n
Rule 2
Filter Result
False
True
Action 1
Action 2
Action n
Rule n
Filter Result
False
True
Action 1
Action 2
Action n

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.

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

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.

rsyslog Windows Agent product Event ID reference

Use this appendix to look up Windows Application Event Log entries generated by rsyslog Windows Agent. Each Event ID links to its meaning, immediate checks, exact procedures, and escalation evidence.

Scope

This reference covers product service diagnostics emitted by current supported builds. It does not cover Event IDs collected from other Windows applications or IDs chosen by an administrator in the Write to Windows Event Log action.

rsyslog Windows Agent Event ID troubleshooting procedures

Use these procedures from the applicable Event ID page. Each procedure includes commands, expected results, failure branches, recovery verification, and evidence collection.

- **`Collect an Event ID and neighboring product events`** — Preserve the complete event and the product events immediately before and after it.
- **`Collect evidence for an escalation-only runtime event`** — Capture a bounded reproducible support package without unsafe generic repair.
- **`Diagnose an action backlog or disk queue`** — Identify why queued work is not draining while preserving data.
- **`Diagnose log rotation and retention`** — Verify trigger, names, handles, destination access, and retention.
- **`Export configuration and collect a bounded debug log`** — Create a text configuration export and time-bounded debug capture, then disable debugging.
- **`Interpret a Windows or Winsock error code`** — Translate a numeric code without losing its operation or subsystem context.
- **`Repair the permitted-senders configuration`** — Authorize an intended sender without weakening the sender restriction.
- **`Resolve a destination and test its TCP port`** — Verify DNS, selected address, routing, and TCP establishment.
- **`Test an ODBC connection in the product context`** — Verify ODBC provider, architecture, authentication, connectivity, and a minimal query.
- **`Test an OLE DB connection in the product context`** — Verify OLE DB provider, architecture, authentication, connectivity, and a minimal query.
- **`Validate configuration and reload it safely`** — Back up, inspect, correct, and test the exact invalid configuration object.
- **`Verify a monitored remote service`** — Confirm resolution, transport, protocol, credentials, expected response, and timing.
- **`Verify a program or Windows-service control action`** — Check target, arguments, working directory, account rights, and positive result.
- **`Verify a UDP path without assuming delivery`** — Confirm receiver binding, firewall policy, and positive receipt for a paced sample.
- **`Verify an SNMP trap sender and receiver path`** — Confirm transport, endpoint, SNMP version, security/community, OIDs, and receipt.
- **`Verify Event Log channel access and bookmark state`** — Confirm channel existence, enablement, account access, and collection position.
- **`Verify File Monitor source state and encoding`** — Check source path, sharing, encoding, line endings, and read position.
- **`Verify file paths, permissions, and free space`** — Check expansion, existence, ACLs, service-account context, and storage.
- **`Verify listener binding and Windows Firewall rules`** — Confirm effective address, port, transport, owning process, and inbound policy.
- **`Verify Microsoft Message Queuing availability and access`** — Confirm feature, service, queue path, transaction mode, and send rights.
- **`Verify product license and feature entitlement state`** — Confirm product, version, validity, edition, and required feature without exposing license data.
- **`Verify sender, receiver, and queued-message recovery`** — Prove end-to-end recovery and backlog drainage.
- **`Verify service state, dependencies, and service account`** — Confirm service state, start mode, dependencies, account, and SCM errors.
- **`Verify SMTP connectivity and mail delivery`** — Separate DNS, TCP, TLS, authentication, relay, recipient, and downstream delivery.
- **`Verify TLS certificates, private keys, and permitted peers`** — Identify CA, certificate, private-key, trust-chain, and permitted-peer failures without exposing private key material.

Event ID index

Event ID	Meaning	Severity	Component
`100`	The service was installed	Information	Windows service lifecycle

101	The service was removed	Information	Windows service lifecycle
102	The service could not be removed	Error	Windows service lifecycle
103	The service control handler could not be installed	Error	Windows service lifecycle
104	The service initialization process failed	Error	Windows service lifecycle
105	The service was started	Information	Windows service lifecycle
106	The service received an unsupported control request	Error	Windows service lifecycle
108	The service was stopped	Information	Windows service lifecycle
118	Product license status reminder	Information	Licensing
11000	SETP connection closed by the receiver	Warning	SETP sender
11001	SETP TLS session closed	Warning	SETP sender
11002	SETP delivery delayed in the action cache	Warning	SETP sender
11003	SETP delivery failed	Error	SETP sender
11004	SETP session restored	Information	SETP sender
11005	Configured feature is not licensed	Warning	Licensing
11006	Call Ruleset action could not find its target ruleset	Error	Call Ruleset action
11007	Action configuration: required timer could not be created	Warning	Action configuration
11008	Action configuration: configured action feature is unavailable	Warning	Action configuration
11009	A queued action message could not be delivered	Warning	Action disk queue
11010	ODBC database action: database operation failed	Error	ODBC database action
11011	ODBC database action: database operation failed	Error	ODBC database action
11012	OLE DB action: database operation failed	Error	OLE DB action
11013	OLE DB action: OLE DB operation failed	Error	OLE DB action
11014	Action processing: action initialization failed	Error	Action processing
11015	Microsoft Message Queuing action: Microsoft Message Queuing operation failed	Error	Microsoft Message Queuing action
11016	An SNMP trap variable contains an invalid OID	Warning	Send SNMP Trap action
11017	Forward Syslog action: action initialization failed	Error	Forward Syslog action
11018	A queued Forward Syslog message could not be delivered	Warning	Forward Syslog disk queue
11019	Syslog Queue action has no queue name	Error	Syslog Queue action
11020	Control Windows Service action could not open Service Control Manager	Error	Control Windows Service action
11021	Action processing: configured action feature is unavailable	Warning	Action processing
11022	Action processing: action type is missing from configuration	Error	Action processing
11023	Action processing: invalid action type in configuration	Error	Action processing
11024	Configured action type is not recognized	Error	Rule engine action
11025	Action processing: invalid action type in configuration	Error	Action processing
11026	Configured action failed after its retries	Error	Rule engine action
11027	Configured action raised an unknown exception at its retry limit	Error	Rule engine action
11028	Configured action raised an unknown exception below its retry limit	Error	Rule engine action
11029	ODBC database action: database operation failed	Error	ODBC database action

11030	ODBC database action: database operation failed	Error	ODBC database action
11031	OLE DB action: database operation failed	Error	OLE DB action
11032	OLE DB action: OLE DB operation failed	Error	OLE DB action
11033	Post Process action ignored an unsupported parse type	Error	Post Process action
11034	SETP sender: TLS client initialization failed	Error	SETP sender
11035	SETP sender: SETP session initialization failed	Error	SETP sender
11036	RELP delivery failed and the message was queued	Warning	Send RELP action
11037	Send to Communications Port action could not write a byte	Error	Send to Communications Port action
11038	SNMP trap transmission failed	Error	Send SNMP Trap action
11039	Forward Syslog delivery failed and the message was queued	Warning	Forward Syslog action
11040	Forward Syslog delivery failed without queue protection	Error	Forward Syslog action
11041	Start Program action received an unexpected wait result	Error	Start Program action
11042	Write to File action has an invalid output format	Error	Write to File action
11043	A sender connection was refused by the permitted-senders limit	Error	Client connection licensing
11044	A sender connection exceeded the licensed client limit	Error	Client connection licensing
11045	File-based configuration: file-based configuration could not be loaded	Error	File-based configuration
11046	File-based configuration: file-based configuration could not be loaded	Error	File-based configuration
11047	Configuration: invalid host or IP address	Error	Configuration
11048	Network and TLS transport: unknown runtime error	Warning	Network and TLS transport
11049	TLS CA certificate file could not be loaded	Warning	TLS certificate loading
11050	TLS certificate or private key could not be loaded	Warning	TLS certificate loading
11051	TLS client certificate verification failed	Warning	TLS client certificate verification
11052	TLS client certificate identity is unavailable	Warning	TLS client certificate verification
11053	TLS client certificate does not match a permitted peer	Warning	TLS peer authorization
11054	TLS certificate-name authentication has no permitted peers	Warning	TLS peer authorization
11055	TLS client certificate fingerprint could not be computed	Warning	TLS peer authorization
11056	TLS certificate-fingerprint authentication has no permitted peers	Warning	TLS peer authorization
11057	Network and TLS transport: TLS server initialization failed	Warning	Network and TLS transport
11058	Filter engine: regular expression initialization failed	Error	Filter engine
11059	Monitoring service: service component initialization failed	Error	Monitoring service
11060	MonitorWare Echo service: MonitorWare Echo operation raised an exception	Warning	MonitorWare Echo service
11061	MonitorWare Echo service: MonitorWare Echo operation raised an exception	Error	MonitorWare Echo service
11062	MonitorWare Echo service: MonitorWare Echo operation raised an exception	Error	MonitorWare Echo service

11063	MonitorWare Echo service: MonitorWare Echo operation raised an unknown exception	Error	MonitorWare Echo service
11064	CPU Monitor service: CPU monitor raised an exception	Error	CPU Monitor service
11065	CPU Monitor service: CPU monitor raised an unknown exception	Error	CPU Monitor service
11066	Database Monitor service: database operation failed	Error	Database Monitor service
11067	Database Monitor service: database operation failed	Error	Database Monitor service
11068	Database Monitor service: database monitor ID field warning	Warning	Database Monitor service
11069	Database Monitor service: database operation failed	Error	Database Monitor service
11070	Database Monitor service: database operation failed	Error	Database Monitor service
11071	Disk Space Monitor could not complete a disk-space check	Error	Disk Space Monitor service
11072	DTLS listener: listener could not bind to its configured address	Error	DTLS listener
11073	DTLS listener stopped after an unknown exception	Error	DTLS listener
11074	File Monitor service: file monitoring warning	Warning	File Monitor service
11075	File Monitor service: configured file could not be opened	Error	File Monitor service
11076	File Monitor service: configured file could not be opened	Error	File Monitor service
11077	File Monitor service: file monitoring operation failed	Error	File Monitor service
11078	File Monitor service: file monitoring operation failed	Error	File Monitor service
11079	File Monitor service: configured file could not be opened	Error	File Monitor service
11080	File Monitor service: configured file could not be opened	Error	File Monitor service
11081	File Monitor service: file monitoring operation failed	Error	File Monitor service
11082	FTP Probe service: probe operation raised an exception	Error	FTP Probe service
11083	FTP Probe service: probe operation raised an unknown exception	Error	FTP Probe service
11084	Heartbeat service: heartbeat operation raised an exception	Error	Heartbeat service
11085	Heartbeat service: heartbeat operation raised an unknown exception	Error	Heartbeat service
11086	HTTP Probe service: probe operation raised an exception	Error	HTTP Probe service
11087	HTTP Probe service: probe operation raised an unknown exception	Error	HTTP Probe service
11088	IMAP Probe service: probe operation raised an exception	Error	IMAP Probe service
11089	IMAP Probe service: probe operation raised an unknown exception	Error	IMAP Probe service
11090	SETP listener stopped after a protocol server error	Error	SETP listener
11091	SETP receiver: SETP operation raised an exception	Warning	SETP receiver
11092	SETP receiver: SETP operation raised an exception	Error	SETP receiver

11093	SETP receiver: SETP operation raised an exception	Error	SETP receiver
11094	SETP receiver: SETP operation raised an unknown exception	Error	SETP receiver
11095	NNTP Probe service: probe operation raised an exception	Error	NNTP Probe service
11096	NNTP Probe service: probe operation raised an unknown exception	Error	NNTP Probe service
11097	Event Log Monitor service: Windows Event Log operation failed	Error	Event Log Monitor service
11098	Event Log Monitor service: Windows Event Log could not be opened	Error	Event Log Monitor service
11099	Event Log Monitor service: unknown runtime error	Error	Event Log Monitor service
11100	A corrupted Windows Event Log record was skipped	Warning	Event Log Monitor service
11101	Event reporter service	Error	Event Log Monitor service
11102	Event Log Monitor service: Windows Event Log record was unreadable	Error	Event Log Monitor service
11103	Event Log Monitor service: Windows Event Log clear target already exists	Error	Event Log Monitor service
11104	Event Log Monitor service: Windows Event Log could not be cleared	Error	Event Log Monitor service
11105	Event Log Monitor service: Windows Event Log clear retry succeeded	Information	Event Log Monitor service
11106	Windows Event Log cleared and saved	Information	Event Log Monitor service
11107	Windows Event Log cleared	Information	Event Log Monitor service
11108	Event Log Monitor service: Windows Event Log operation failed	Error	Event Log Monitor service
11109	Event Log Monitor service: Windows Event Log operation failed	Error	Event Log Monitor service
11110	Event Log Monitor service: licensed Event Log Monitor limit reached	Error	Event Log Monitor service
11111	Windows Service Monitor: Windows Service Monitor raised an exception	Error	Windows Service Monitor
11112	Syslog listener: service component initialization failed	Error	Syslog listener
11113	Syslog listener: service component initialization failed	Error	Syslog listener
11114	Syslog listener: listener could not bind to its configured address	Warning	Syslog listener
11115	Syslog listener: listener could not bind to its configured address	Error	Syslog listener
11116	Passive Syslog listener stopped after an unknown exception	Error	Passive Syslog listener
11117	Ping Probe service stopped after a reported runtime error	Error	Ping Probe service
11118	Ping Probe service stopped after an unknown exception	Error	Ping Probe service
11119	POP3 Probe service: probe operation raised an exception	Error	POP3 Probe service
11120	POP3 Probe service: probe operation raised an unknown exception	Error	POP3 Probe service
11121	RELP listener: listener could not bind to its configured address	Error	RELP listener
11122	RELP listener stopped after an unknown exception	Error	RELP listener

11123	Serial Port Monitor: serial port operation failed	Error	Serial Port Monitor
11124	Serial Port Monitor: serial port operation raised an exception	Error	Serial Port Monitor
11125	Serial Port Monitor: serial port operation raised an unknown exception	Error	Serial Port Monitor
11126	SMTP listener: SMTP listener raised an exception	Warning	SMTP listener
11127	SMTP listener: SMTP listener raised an exception	Error	SMTP listener
11128	SMTP listener: SMTP listener raised an exception	Error	SMTP listener
11129	SMTP listener: SMTP listener raised an unknown exception	Error	SMTP listener
11130	SMTP Probe service: probe operation raised an exception	Error	SMTP Probe service
11131	SMTP Probe service: probe operation raised an unknown exception	Error	SMTP Probe service
11132	SNMP Monitor service: SNMP operation raised an exception	Warning	SNMP Monitor service
11133	SNMP Monitor service: SNMP operation raised an exception	Error	SNMP Monitor service
11134	SNMP Monitor service: SNMP operation raised an unknown exception	Error	SNMP Monitor service
11135	SNMP Monitor service: SNMP operation raised an exception	Error	SNMP Monitor service
11136	SNMP Monitor service: SNMP operation raised an unknown exception	Error	SNMP Monitor service
11137	SNMP Trap Receiver: SNMP operation raised an exception	Warning	SNMP Trap Receiver
11138	SNMP Trap Receiver: SNMP operation raised an exception	Error	SNMP Trap Receiver
11139	SNMP Trap Receiver: SNMP operation raised an exception	Error	SNMP Trap Receiver
11140	SNMP Trap Receiver: SNMP operation raised an unknown exception	Error	SNMP Trap Receiver
11141	Syslog listener: BEEP listener option could not be applied	Error	Syslog listener
11142	Syslog listener: BEEP listener initialization failed	Error	Syslog listener
11143	Syslog listener: BEEP listener port could not be applied	Error	Syslog listener
11144	Syslog listener: listener could not bind to its configured address	Warning	Syslog listener
11145	Syslog listener: listener could not bind to its configured address	Error	Syslog listener
11146	Syslog listener stopped after an unknown exception	Error	Syslog listener
11147	Event Log Monitor V2 service: required timer could not be created	Warning	Event Log Monitor V2 service
11148	Event Log Monitor could not create its saved bookmark	Error	Event Log Monitor V2 service
11149	Event Log Monitor V2 service: general runtime error	Warning	Event Log Monitor V2 service
11150	Event Log Monitor V2 service: Windows Event Log operation failed	Error	Event Log Monitor V2 service
11151	Event Log Monitor V2 service: Windows Event Log operation failed	Error	Event Log Monitor V2 service
11152	A queued event has an unsupported information-unit type	Error	Event reconstruction

11153	Existing archive file was renamed before log rotation	Warning	Log rotation
11154	Log rotation was deferred because an archive destination could not be renamed	Warning	Log rotation
11155	Cross-volume archive copy succeeded but source cleanup failed	Warning	Log rotation
11156	Rotated log could not be moved to its archive destination	Warning	Log rotation
11157	Existing archive could not be renamed to prevent overwrite	Warning	Log rotation
11158	Empty log-rotation queue file could not be deleted	Warning	Log rotation queue
11159	Corrupt log-rotation queue file could not be quarantined	Warning	Log rotation queue
11160	Corrupt log-rotation queue file was quarantined	Warning	Log rotation queue
11161	Configuration reload kept the existing log-rotation workers	Warning	Log rotation
11162	Log-rotation jobs cannot be persisted because the data directory is unset	Warning	Log rotation queue
11163	Persisted log-rotation queue file could not be read	Warning	Log rotation queue
11164	Email action: invalid SMTP recipient address	Error	Email action
11165	Email action: SMTP recipient address is missing	Error	Email action
11166	OLE DB action: OLE DB operation failed	Error	OLE DB action
11167	Runtime resource management: memory reserve is in use	Warning	Runtime resource management
11168	Runtime resource management: memory allocation failed	Error	Runtime resource management
11169	Individual Ping Probe stopped after an unknown exception	Error	Ping Probe
11170	Queue worker received an unexpected wake-up	Error	Main message queue
11171	Queue manager: unknown runtime error	Error	Queue manager
11172	Queue manager: unknown runtime error	Error	Queue manager
11173	An event failed while the main queue applied its rules	Error	Main message queue
11174	Queue manager: queue capacity condition	Information	Queue manager
11175	Queue manager: queue capacity condition	Warning	Queue manager
11177	Queue manager: queue capacity condition	Error	Queue manager
11178	Queue manager: disk queue cache operation failed	Error	Queue manager
11179	Queue manager: disk queue cache operation failed	Error	Queue manager
11180	Queue manager: disk queue cache operation failed	Error	Queue manager
11181	Queue manager: disk queue cache operation failed	Error	Queue manager
11182	Queue manager: disk queue cache operation failed	Error	Queue manager
11183	Configuration reload was skipped because the log-rotation scheduler did not stop	Warning	Configuration reload
11184	Service configuration: memory reserve could not be allocated	Warning	Service configuration
11185	No product input service is configured	Error	Service configuration
11186	The trial expired and the service was disabled	Error	Licensing
11187	A configured input service type is not recognized	Error	Service configuration
11188	Service configuration: configured service feature is unavailable	Warning	Service configuration
11189	Input service could not find its assigned ruleset	Error	Service configuration

11190	Service configuration: configured feature is unavailable	Warning	Service configuration
11191	Service configuration: log rotation is not licensed	Information	Service configuration
11192	No configured product input service could be started	Error	Service startup
11193	Service configuration: service configuration raised an unknown exception	Warning	Service configuration
11194	Product runtime: message compression failed	Warning	Product runtime
11195	Disk queue: runtime data error	Error	Disk queue
11196	Disk queue: runtime data error	Error	Disk queue
11197	Disk queue: runtime data error	Error	Disk queue
11198	TCP and TLS listener: TCP listener fatal error	Error	TCP and TLS listener
11199	TCP and TLS listener: TLS server initialization failed	Warning	TCP and TLS listener
11200	TCP and TLS listener: TCP listener fatal error	Error	TCP and TLS listener
11201	TCP and TLS listener: TCP listener fatal error	Error	TCP and TLS listener
11202	TCP listener could not assign an accepted socket	Error	TCP and TLS listener
11211	Product runtime: configured advanced feature is unavailable	Error	Product runtime
11212	System exception while compiling an NT Event Log expression	Error	Event Log Monitor service
11213	System exception while compiling an NT Event Log category	Error	Event Log Monitor service
11214	System exception in the queue worker thread	Error	Queue manager
11215	A debug error was forwarded to the Windows Event Log	Warning	Diagnostic forwarding
11216	Control Windows Service action could not open the target service	Error	Control Windows Service action
11217	Control Windows Service action could not start the target service	Error	Control Windows Service action
11218	Control Windows Service action could not stop the target service	Error	Control Windows Service action
11219	Control Windows Service action could not pause the target service	Error	Control Windows Service action
11220	Control Windows Service action could not continue the target service	Error	Control Windows Service action
11221	Target service did not reach the requested state before timeout	Error	Control Windows Service action
11222	Consumed log-rotation queue file could not be deleted	Warning	Log rotation queue
11223	Protected service shutdown deadline expired	Error	Service shutdown
11224	Configuration reload shutdown incomplete	Error	Service shutdown
11225	RELP listener shutdown wake fallback	Warning	RELP listener
11226	SNMP listener shutdown wake fallback	Warning	SNMP trap listener
11227	Event rejected after shutdown intake closed	Warning	Main message queue
11228	Action disk queue flush failed during shutdown	Error	Action persistence
11229	Message Ringbuffer action is not registered	Error	Message Ringbuffer action

Known issues and errata

Known issues and errata

This section documents confirmed product behavior that can affect operation or the interpretation of collected data. Each notice identifies the affected products and service versions, the practical impact, available workarounds, and the planned or released correction.

Current errata

- ERR-2026-002: rsyslog Windows Agent update reports success but program files are not upgraded correctly

External reference

- Version History
- rsyslog Windows Agent 26.07 release notes

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

C

Command Line Switches
Complex Filter Conditions
Connect to Computer
Customer Properties

D

Database Monitor

E

Edition Comparison
Event Properties
Event-Specific Properties

F

Filter Conditions
FromPos
FTP

H

HTTP

I

IETF
IMAP
Information Units
Installation
IPv6

M

Millisecond

N

NNTP

O

Options

P

POP3
Property

R

RELP
Resource ID
RFC 3164
RFC 3195
RFC 5424
rsyslog Windows Agent Shortcut Keys
Rule Engine
Rules

S

SETP
SMTP
SNMP
Standard Properties
Syslog Facility
Syslog Message Properties
System Properties
System Requirements

T

TCP
ToPos

U

UDP
UTC

W

Windows Event Log Properties
Windows Event Log V2 Properties