WinSyslog 5.0

User Manual

By Adiscon

Contents

About Adiscon WinSyslog 5.0

WinSyslog is an integrated, modular and distributed solution for system management. Network administrators can continuously monitor their systems and receive alarms as soon as important events occur.

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

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

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

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

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

Features

Centralized Logging

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

Ease of Use

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

Powerful Actions

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

Interactive Server

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

Freeware Mode

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

Standards Compatible

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

WinSyslog Web Access

Never need to look at plain text files! WinSyslog comes with a fully functional ASP application that will display the contents of WinSyslog generated database entries.

The ASP pages are in full source code and can easily be customized.

Syslog Hierarchy

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

Email Notifications

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

Store Messages Persistently

The WinSyslog server process stores all messages persistently. So later auditing and review of important system events is possible without effort. Messages can be written to flat ASCII files, ODBC data sources and the Windows event log.

Multiple Instances

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

Full Logging

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

Full Windows 2000, 2003 and XP Support

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

Robustness

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

Minimal Resource Usage

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

Firewall Support

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

NT Service

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

Runs on large Variety of NT Systems

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

Multi-Language Client

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

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

Components

WinSyslog Configuration Client

The WinSyslog Configuration Client – called “the client” - is used to configure all components and features of the WinSyslog Service. The client can also be used to create a configuration profile on a base system. That profile can later be distributed to a large number of target systems.

WinSyslog Service

The WinSyslog Service – called “the service” - runs as a Windows service and carries out the actual work.

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

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

Interactive Syslog Server

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

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

WinSyslog Web Access

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

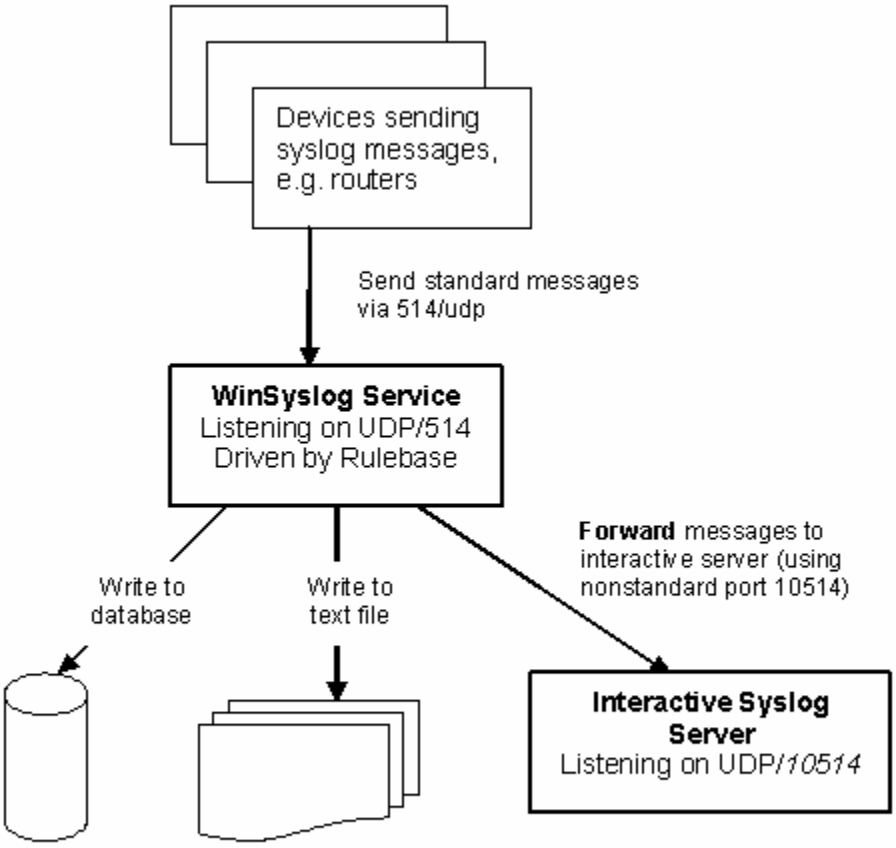
How these components work together

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

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

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

The message flow can be seen in this diagram:



In a typical configuration, the syslog devices (for example routers or switches) send standard syslog messages via port 514 to the WinSyslog service. The service receives these messages and processes them as configured in the rule base. In our example, there are three actions configured for all incoming messages: writing them to a database, to a text file as well as forwarding them to the Interactive Syslog Server. By default, messages are forwarded to the local (127.0.0.1) Interactive Server via port 10514. The Interactive Server in turn listens to that port and receives the forwarded syslog messages from the server.

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

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

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

WinSyslog Web Access is only needed if accessing syslog messages over the web is desired. Thus it is an optional component and is not installed by default. For Web Access to operate correctly, the service must be configured to store incoming messages into a database. This is not done by default and needs to be configured in

the service configuration client. Web Access is fully optional, so there is no need to install it. No other component is depending on the presence of Web Access.

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

System Requirements

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

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

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

At runtime, the base service requires 4 MB of main memory and less than 1 MB of disk space. However, the actual resources used by the service largely depend on the services configured.

If the service shall just receive a few syslog messages per second, a performance impact is barely noticeable, if at all visible. If the WinSyslog service is receiving hundreds of messages per second, it will need much more resources. Even then, the actual load is depending on the actions carried out. Storing the messages to text files is much less performance intense than writing them to a database table – especially if the database engine is located on the same machine. As such, there is no single guideline for hardware sizing. It needs to be adapted to the expected workload.

Please note, however, that the service is specifically optimized to handle high throughput including message bursts (for example received via syslog). If you expect high volume burst and carry out time consuming actions (for example database writes), we highly recommend adding additional memory to the machine[1]. WinSyslog is capable of storing such bursts temporarily in memory even if the machine would otherwise be too slow to process the messages.

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

[1] Even 64 MB additional memory will do nicely. A typical syslog message (including overhead) will take roughly 1,5 KB. With 64 MB, you can buffer up to 50,000 messages in 64 MB.

Getting Started

Setup

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

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

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

to access the online seminars on WinSyslog as well as other members of this product family. Please note that these are not marketing videos but actually technically-packed presentations that will help you getting started quickly and efficiently.

Installing WinSyslog is simply and easy. A standard setup program installs the application.

The install set (the ZIP file you downloaded) contains a standard setup program and its necessary helper files. Please unzip the archive to any directory you like. This can be a local drive, a removable one or a remote share on a file server. A Win32 Unzip program can be found at www.winzip.com.

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

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

Creating an Initial Configuration

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

- **create a simple rule set**

  The most basic rule set includes no critaria, which means all incoming messages will match. To get started, we recommend using just a single "write to file" action which will write the incoming messages to the local disk.

- **create at least one syslog listener**

Be sure to associate the created rule set with that syslog listener

- **start the WinSyslog service**

Your system is now ready to acced and store incoming messages. To unleash the full power of WinSyslog, be sure to read “WinSyslog Concepts” on page 50.

Installing WinSyslog Web Access

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

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

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

Step-by-Step Guides

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

In order to save you download time, we have just included one of the step-by-step guides into the manual. All others are available online at

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

Please visit this hyperlink to view what is available – chances are good there are some matching your desired scenario.

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

The step-by-step guides do not include all information that might be relevant to the situation. For details on the configuration properties, please see "Configuring WinSyslog" on page 26.

In the step-by-step guides, we assume the product is already successfully installed but no configuration has been done. If it is not installed, please do so first. Information on installing can be found in "Setup" on page 7.

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

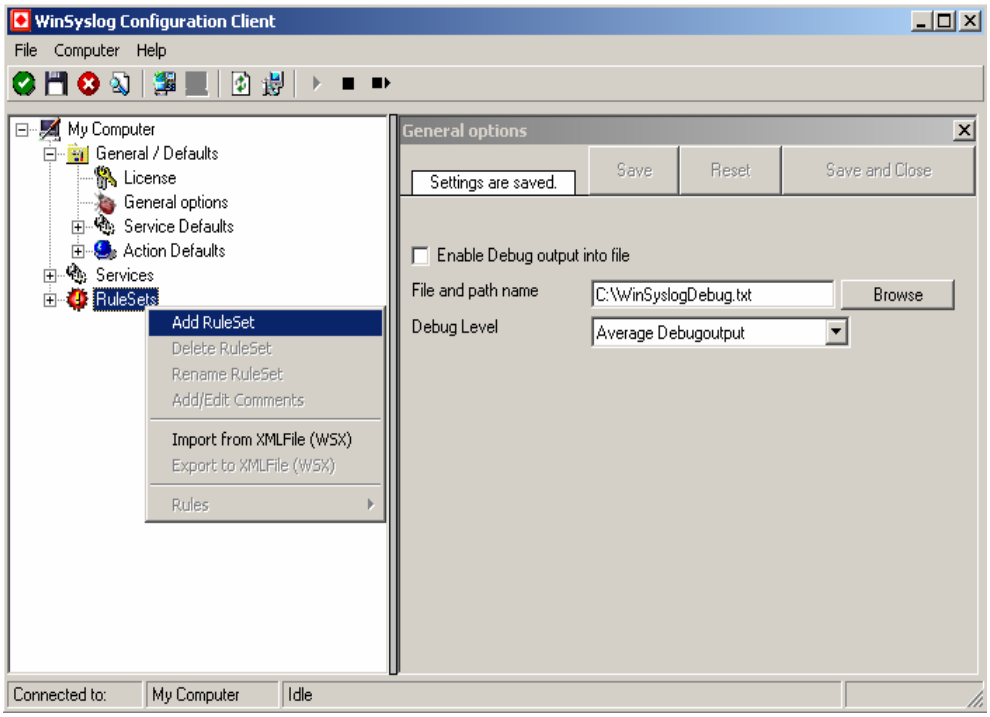
Creating a simple Syslog Server

In this scenario, a simple syslog server will be created. No other services are configured. The syslog server will operate as a standard syslog server on the default port of 514/UDP. All incoming data will be written to a single text file.

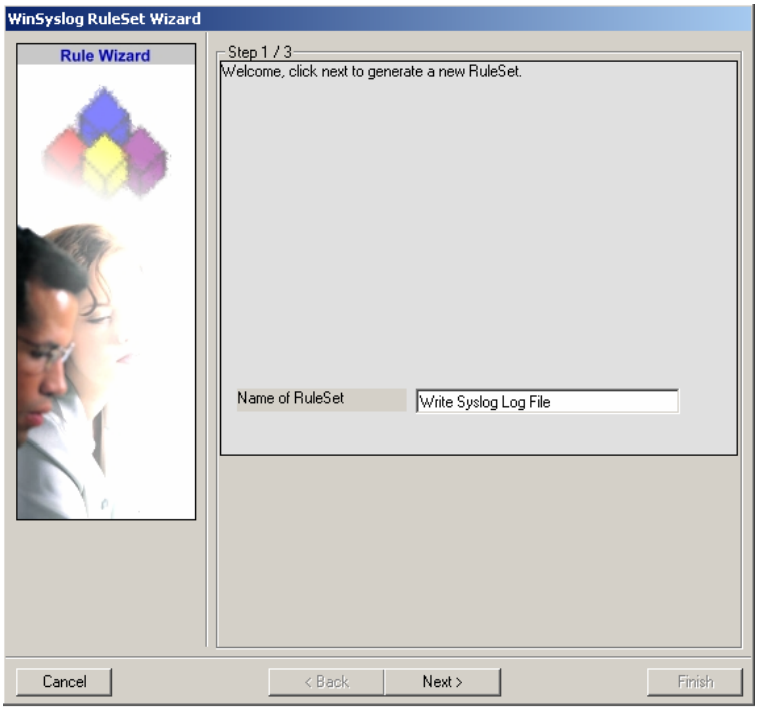
Step 1 – Defining a Rule Set for File Logging

The rule set specifies what action to carry out. You might be tempted to define the service first, but starting with the rule set makes things easier as it already is present when the service will be defined later and needs to be bound to a rule set.

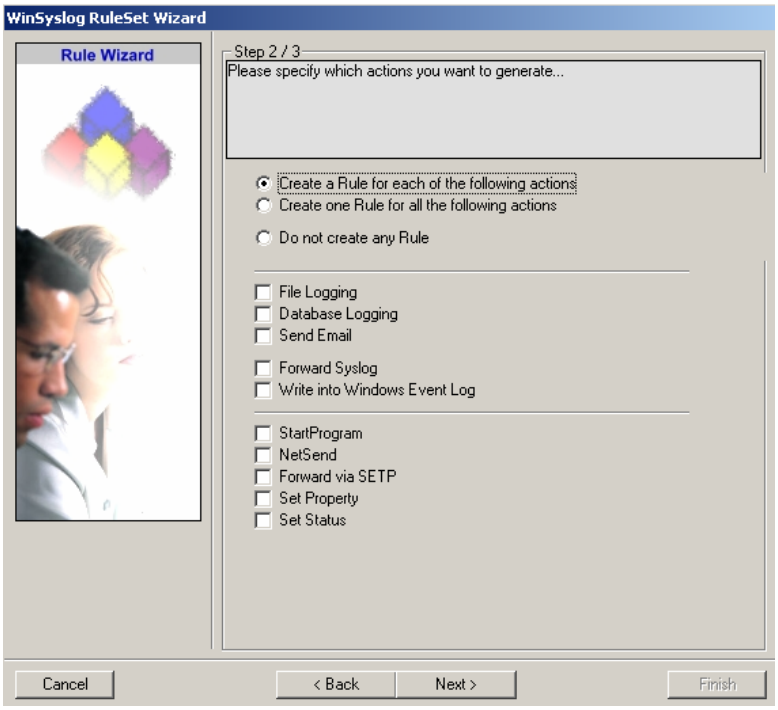
To define a new rule set, right click “Rules”. A pop up menu will appear. Select “Add Rule Set” from this menu. On screen, it looks as follows:



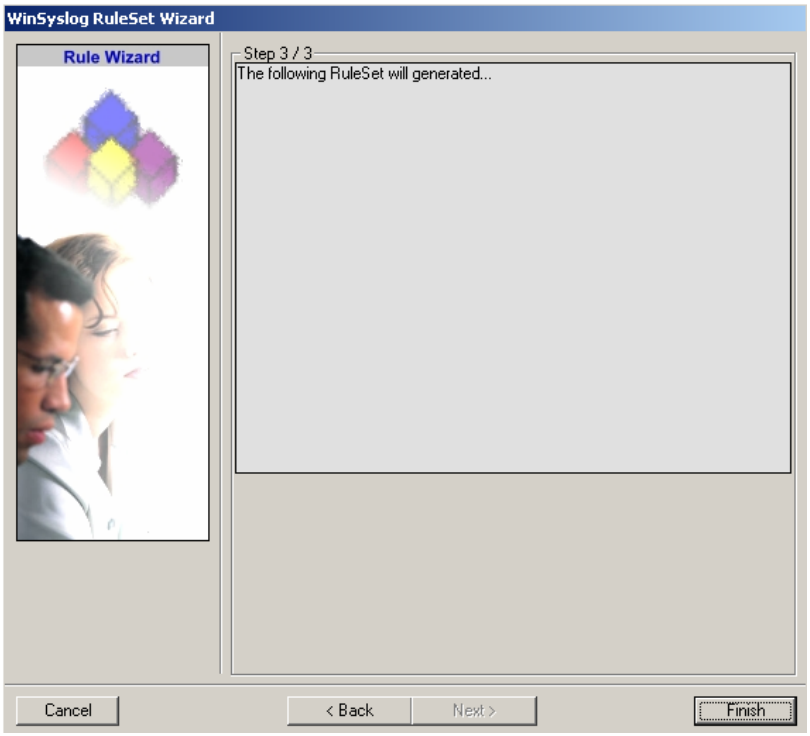
Then, a wizard starts. Change the name of the rule set to whatever name you like. We will use “Write Syslog Log File” in this example. The screen looks as follows:



Click “Next”. A new wizard page appears:

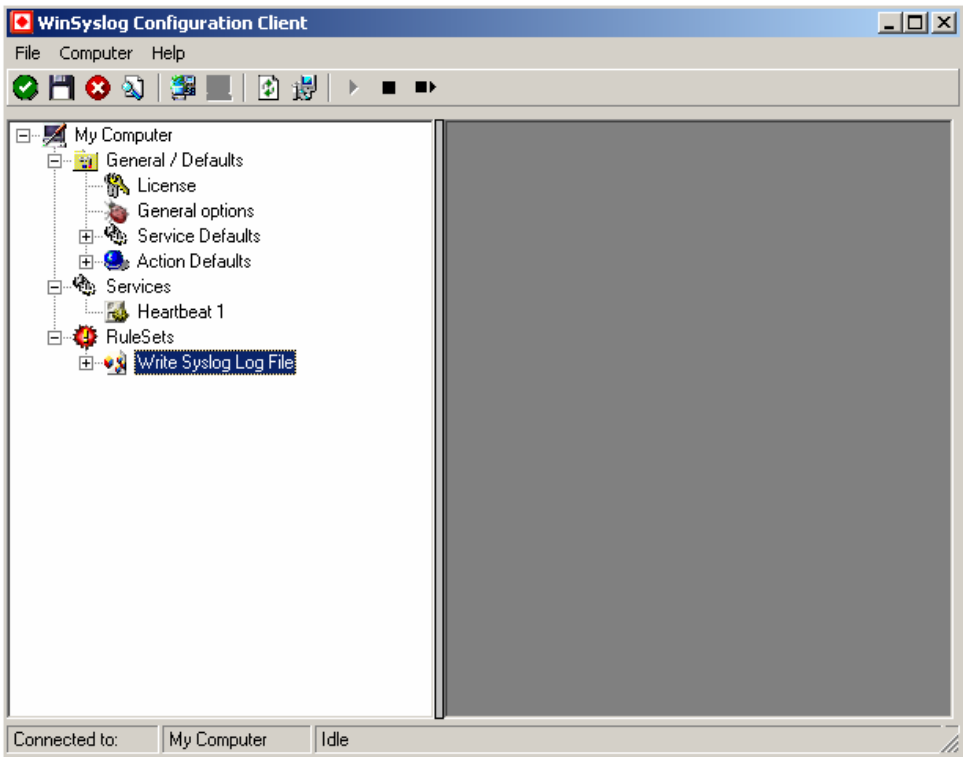


There, select file logging. Do not select any other options for this sample. Also, leave the “Create a Rule for each of the following actions” setting selected. Click “Next”.

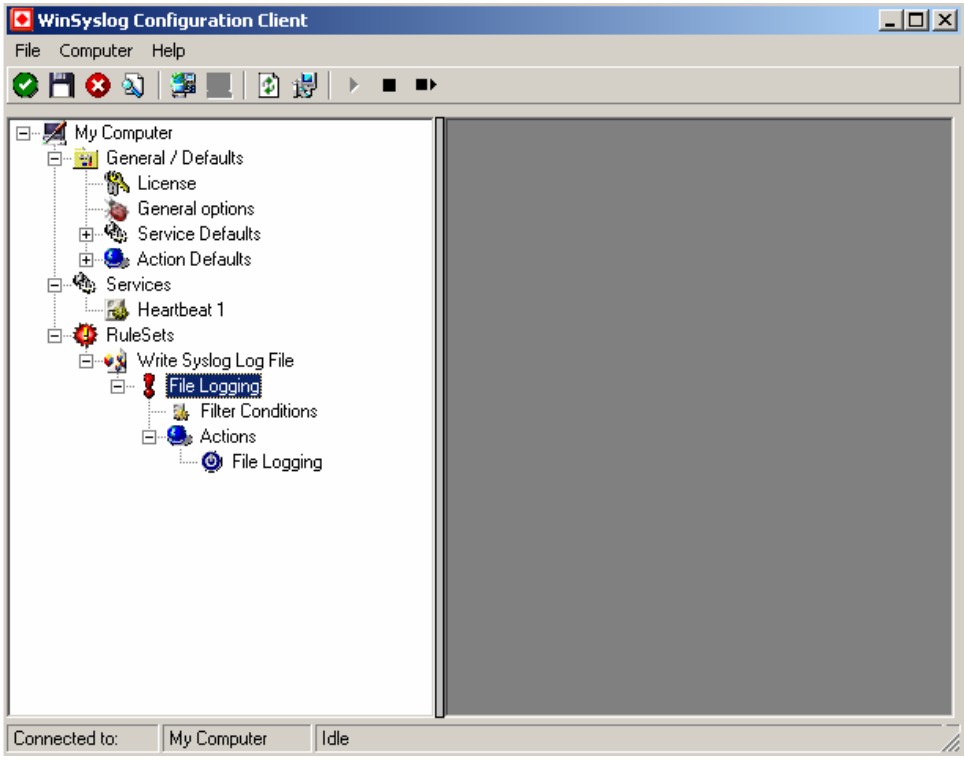


This is just a confirmation page. Click “Finish” to create the rule set.

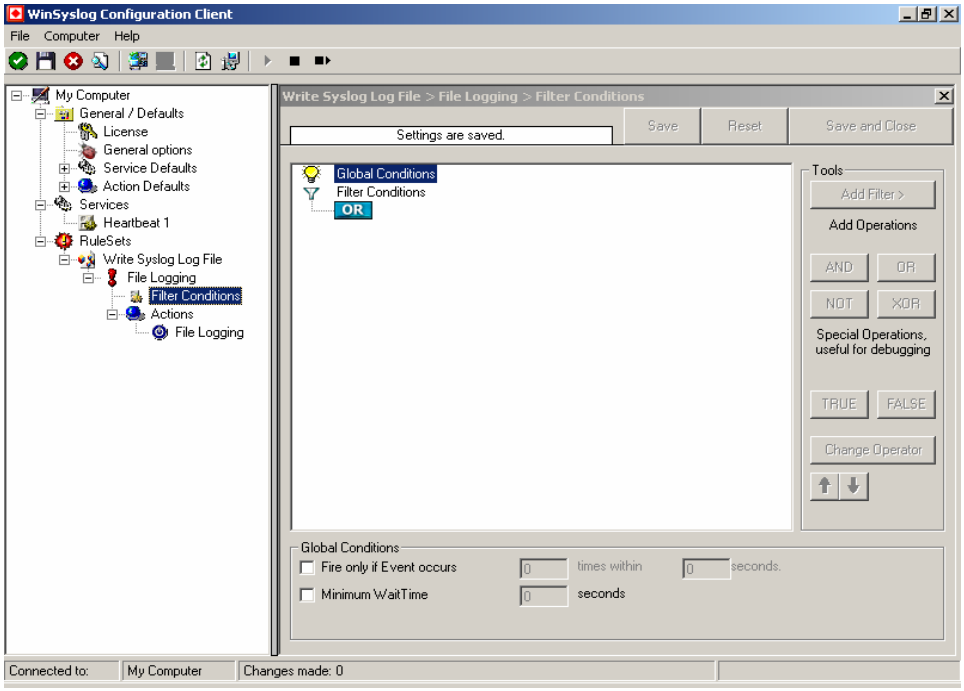
The wizard closes and the client shows a newly created rule set.



As you can see, the “Write Syslog Log File” rule set is now present. Please expand it in the tree view until you have the following screen contents:



As you can see, we have a “File Logging” action configured. We will review the settings just for your information. Click on “Filter Conditions”:



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

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

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

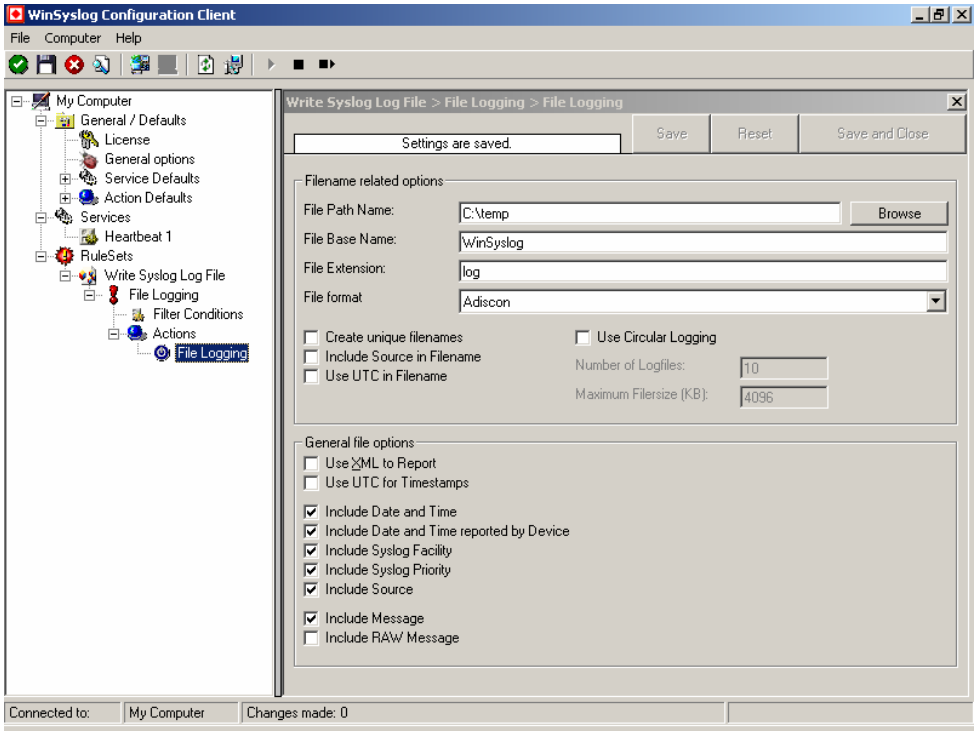
If you are note so sure about the Boolean operations, you might find the following brush-up helpful:

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

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

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

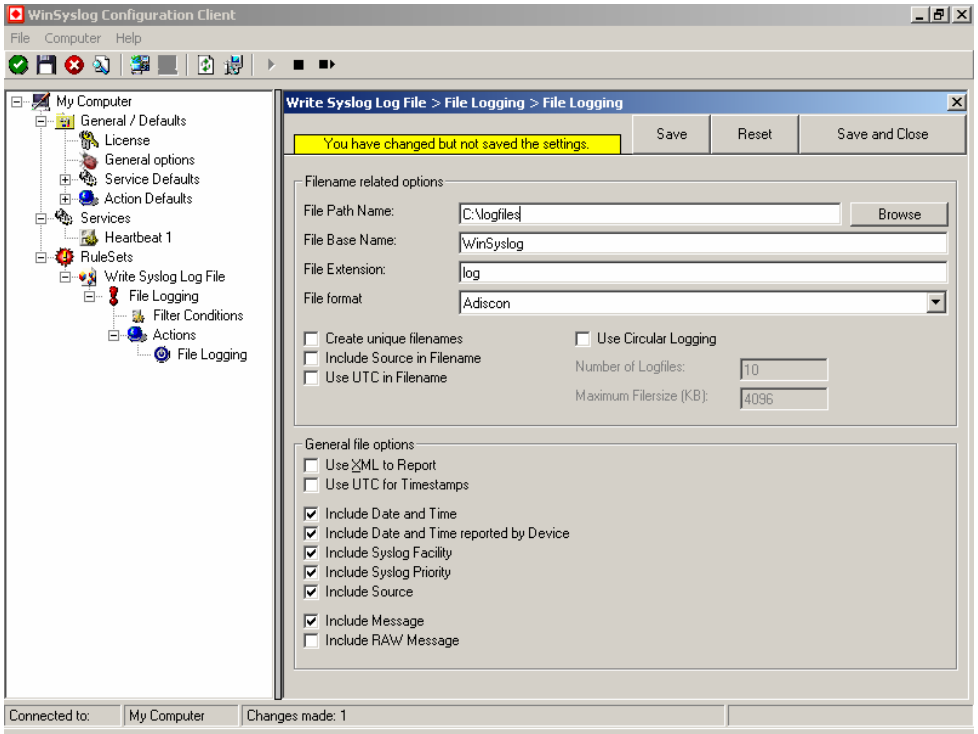
Now let us check the “File Logging” action itself. Please select it in the tree view:



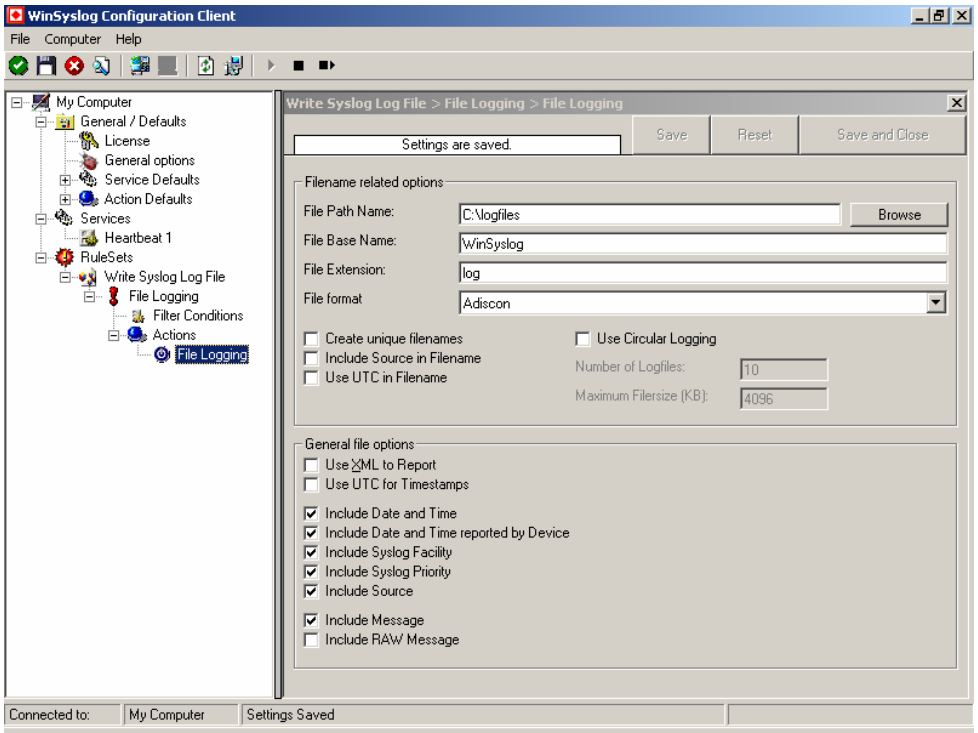
As you can see, it has been created with the default parameters. Each day, a file will be created in the C:\temp directory and its base name will be WinSyslog. It will include all information items in the file.

If you would like to store it into a separate directory or change the file name, here is the place to do it. **Important:** please make sure the directory you specify exists! If it does not yet exist, please create it before you start the service. If the directory does not exist, the service is not able to store any files.

In our example, we would like to save it to “c:\logfiles” with a base name of “Syslog” following Adiscon format. Therefore, we change these properties:



After doing so, you will notice the yellow text on top of the window. It tells you that the configuration changes have not yet been applied. To do so, press “save”.

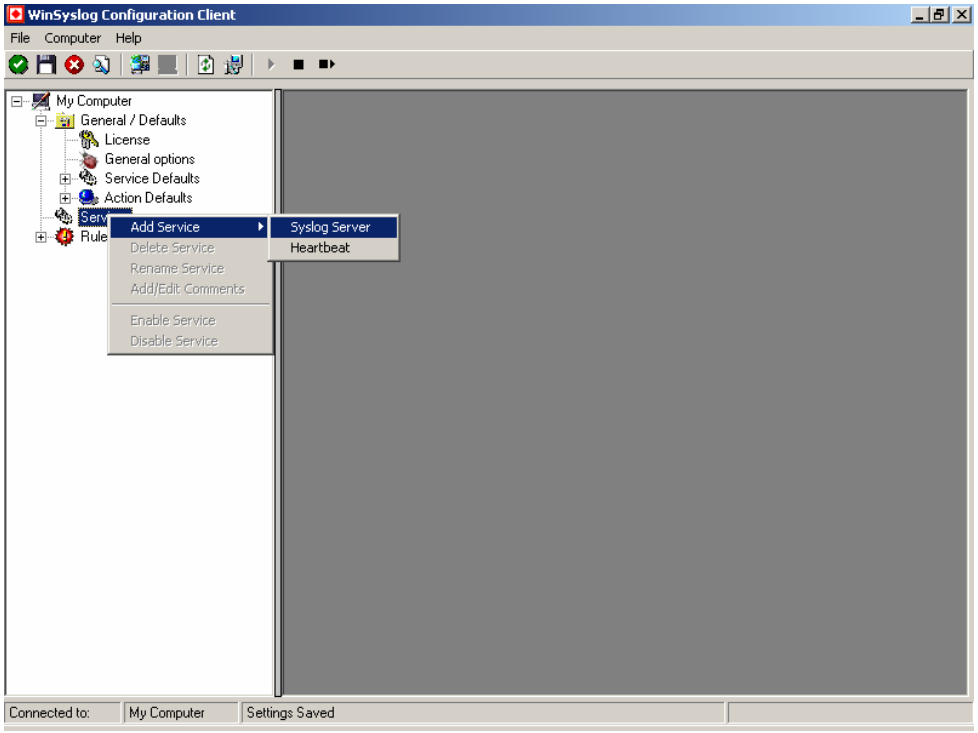


Now you have a workable rule set for logging incoming messages to a text file.

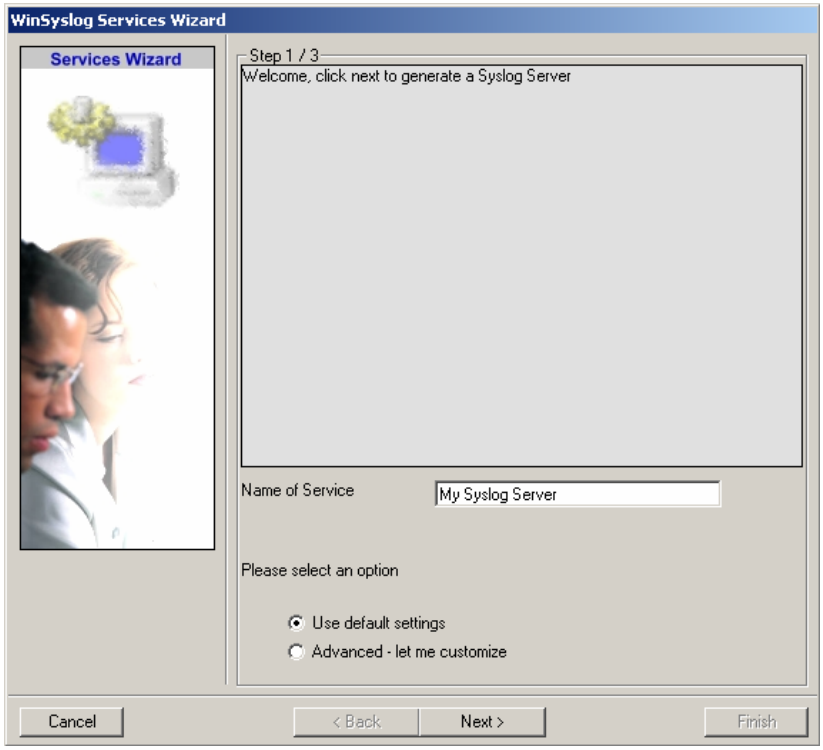
Step 2 – Create a Syslog Server Service

Now we need to define a syslog server service. A syslog server is also sometimes called a “syslog daemon”, “syslogd” or “syslog listener”. It is the process that receives incoming messages.

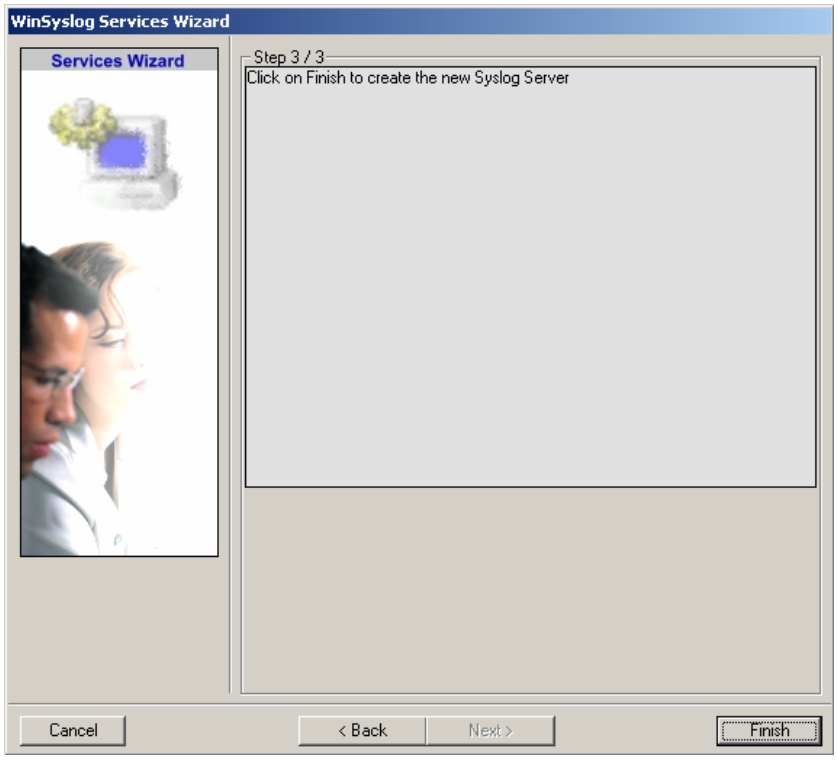
To define it, right click on “Services”, then select “Add Service” and the “Syslog Server”:



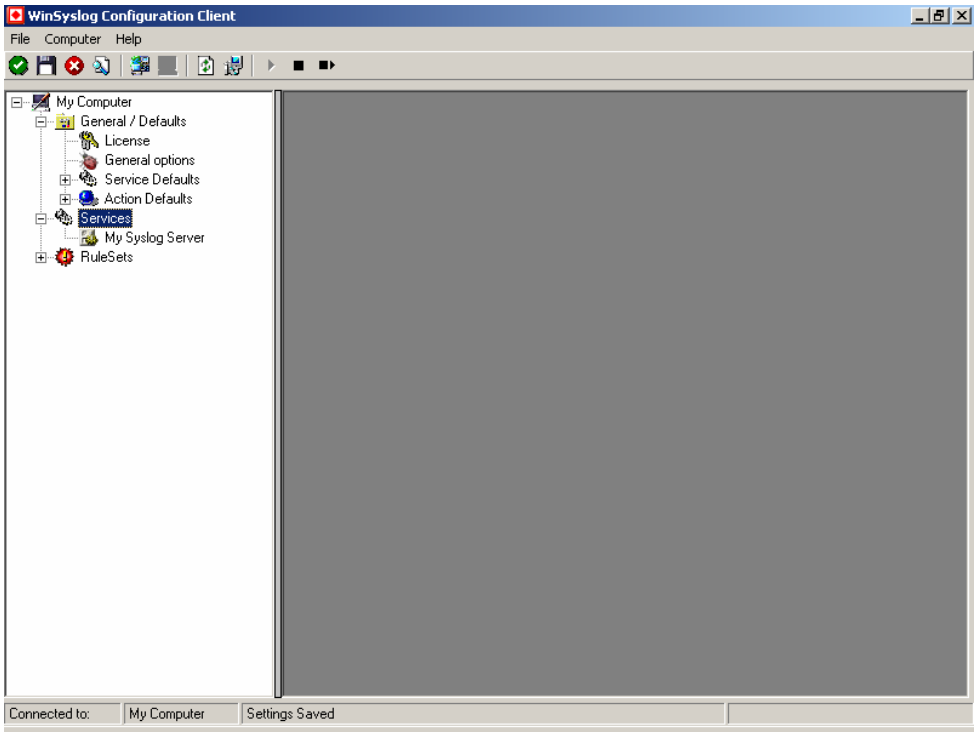
Once you have done so, a new wizard starts:



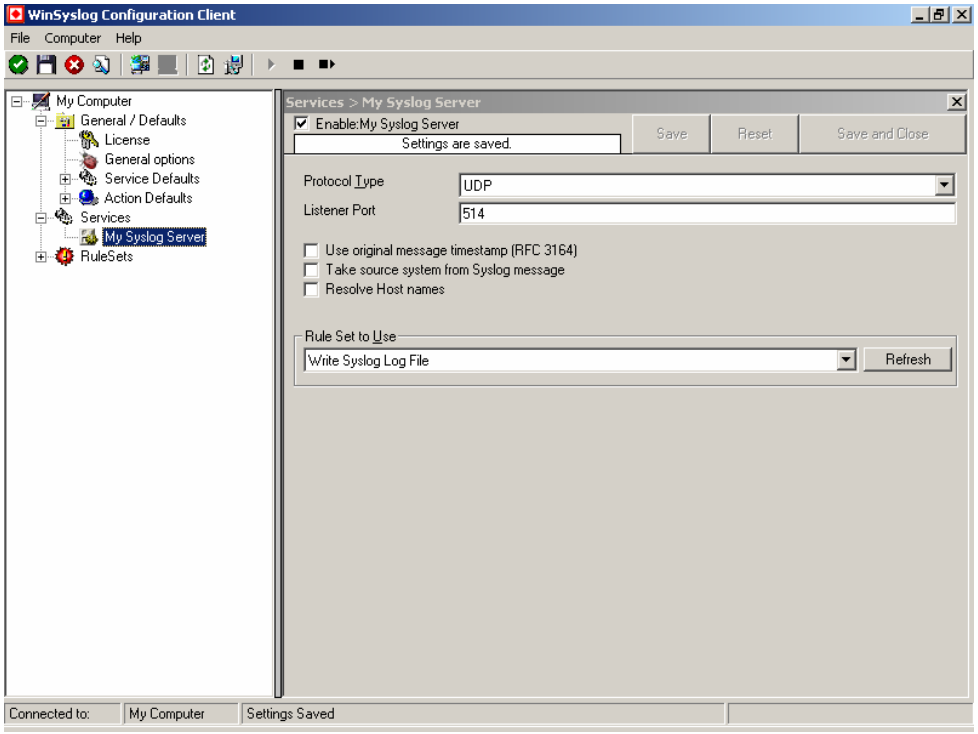
Again, you can use either the default name or any one you like. We will use “My Syslog Server” in this sample. Leave the “Use default settings” selected and press “Next”:



As we have used the default, the wizard will immediately proceed with step 3, the confirmation page. Press “Finish” to create the service. The wizard completes and returns to the configuration client. There, you will see the newly created service beneath the “Services” part of the tree view:



To check its parameters, select it:



As you can see, the service has been created with the default parameters. As such, it operates as a non-RFC compliant syslog server. Our experience shows that many devices are not yet RFC compliant. As such, checking the "Use original message timestamp" and "take source system form syslog message" often causes invalid timestamp and source system information. We strongly recommend keeping these boxes unchecked if you do not definitely know the reporting devices are syslog RFC compliant. Even compliant devices work very well with the default settings.

Please note that the "Write Syslog Log File" has been automatically assigned as the rule set to use. This is the case because we already created it and it is the only rule set. By default, the wizard will always assign the first rule set visible in the tree view to new services. If another one is to be used, you need to change it to the correct one here in the service definition.

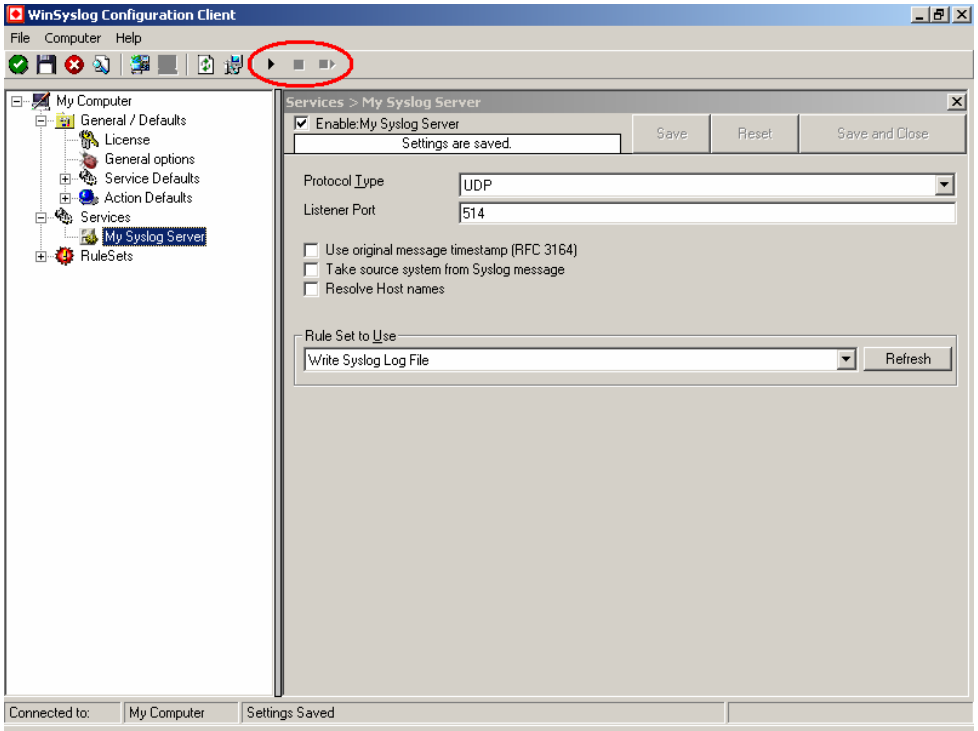
Also, please note that the wizard uses the default properties from the "Service Defaults". Obviously, if these are changed, the default properties for new services will differ.

This procedure completes the configuration of the syslog server.

Step 3 – (Re-) Start the WinSyslog Service

WinSyslog cannot dynamically read changed configurations. As such, it needs to be restarted after such changes. In our sample the service is already running so we need to re-start it. If it is not stated at all you simply need to start it.

Service control can be done with both the respective operating system capabilities (like service manager MMC) or with the configuration client. These are shown in the red surrounded area in the following screen shot:



The buttons resemble Windows service manager – start, stop and restart. In this sample, stop and restart are grayed out because the service is not running.

After service restart, the new definitions are active and WinSyslog is ready to accept and store incoming messages.

Step 4 – Configure your Syslog-Enabled Devices

Even though WinSyslog is now ready, it can only receive message if some device send them. Remember, syslog is a protocol where the server is passively waiting for incoming messages. As long as no device sends message, the syslog server will not log anything.

As there is a large variety of devices, we unfortunately cannot provide device specific instructions. However, almost all devices need to be configured with their specific configuration tool. Typically, only two settings need to be made: one to activate syslog messages at all and one with the syslog server IP address or name.

For some devices, we have step-by-step guides. Please read "Sample Syslog Device Configurations" on page 17 for further details.

Remember: the computer WinSyslog runs on now acts as a syslog server. As such, you need to find out its IP address or name and supply it to the device as the syslog server. Please note that not all devices can operate with computer names. Use the IP address, if in doubt.

Sample Syslog Device Configurations

WinSyslog can receive vital network status information from a variety of devices. As these devices are from many different vendors and have many different applications, it is impossible to provide detailed configuration information for all of them.

We provide configuration information for some well known devices. Hopefully, the samples will provide some idea of how other devices might be configured.

In order to reduce download size, this manual only includes a small number of instructions. For more devices, please visit

http://www.monitorware.com/en/syslog-enabled-products/

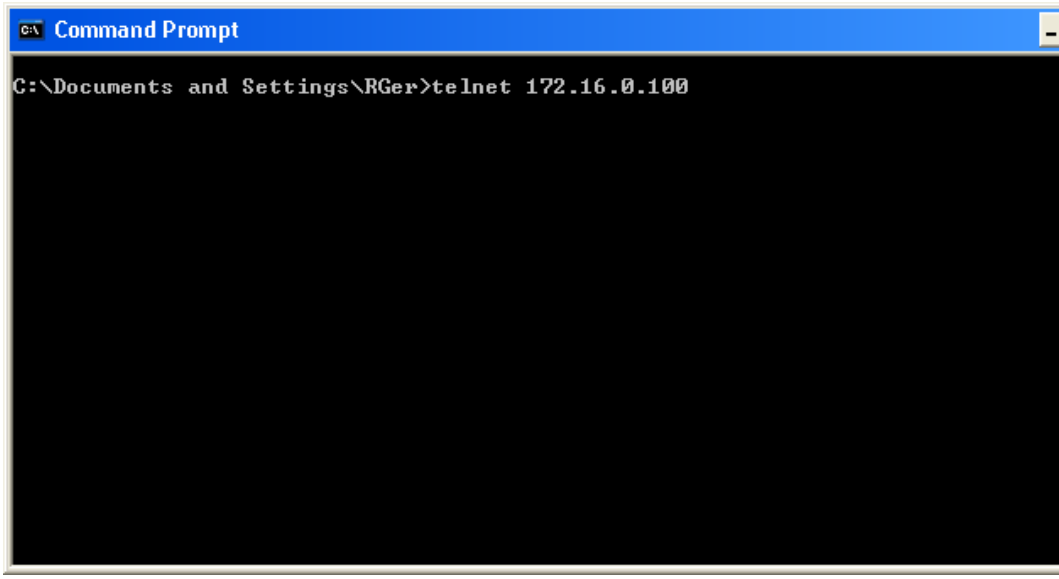
NetGear RT314 Syslog Configuration

The RT314 supports syslog. Unfortunately, syslog messages can not be enabled via the web interface. It must be done via telnet, a command line interface.

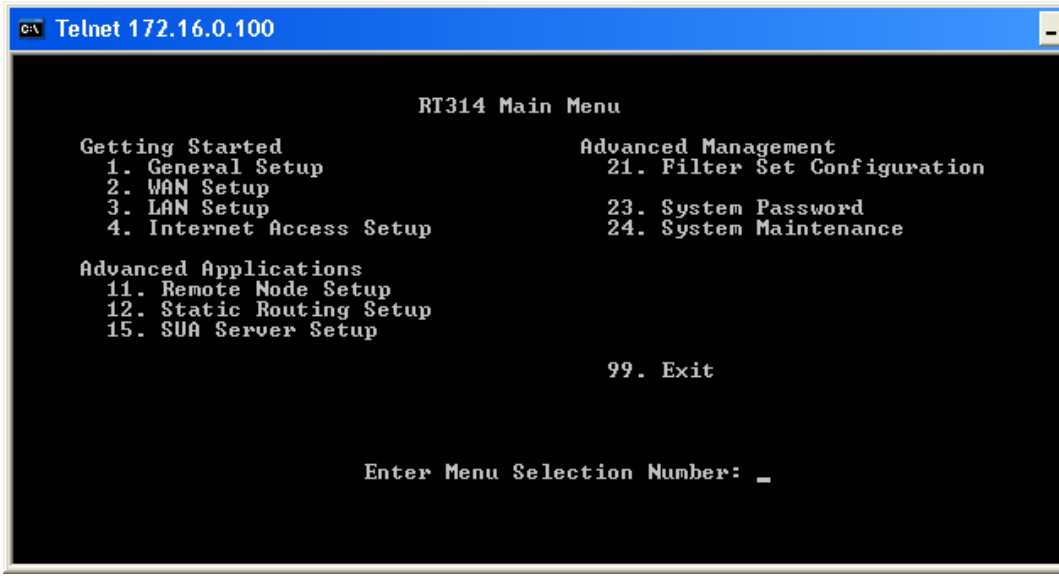
To the best of our knowledge, the NetGear RT314 is compatible to ZyXEL Prestige 314. As far as we know, both of them operate with a version of the ZyNOS operating system that supports a menu system via telnet. As such, the description here does also apply to the ZyXEL product. There might be other routers available that base on the same operating system. If in doubt, start a telnet session to your router and check if this step-by-step guide applies to your device.

In our example, we assume the router has address 172.16.0.100. The syslog server has the address 172.16.0.4.

First, open a command prompt (“DOS box”). Then, type “telnet 172.16.0.100” as shown in this sample:

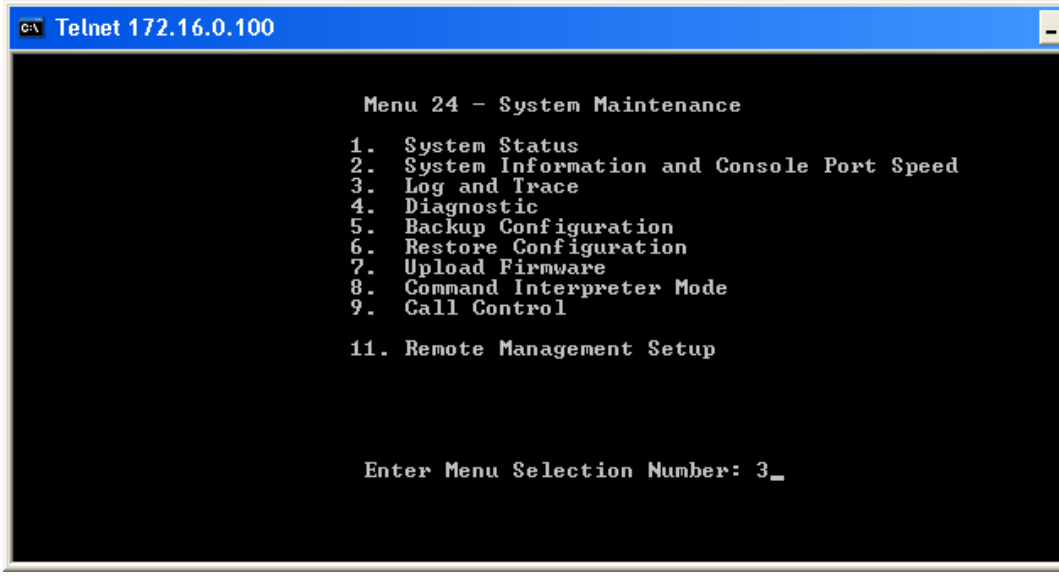


The router will prompt your for the password. Enter it and the following and the main menu will appear:

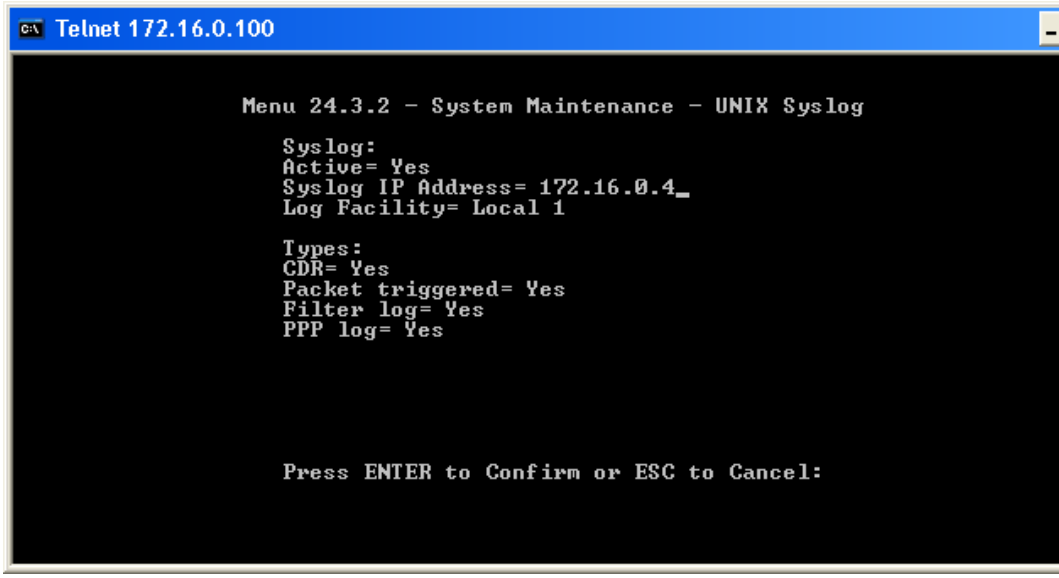


The syslog server’s address can be configured under “System Maintenance”. As such, enter 24 and press enter. The system maintenance menu appears:

There, enter 3 (as shown below) and press enter:



Now enter 2 and press enter. The syslog properties appear:



The screen shot displays the correct configuration for maximum logging. To change the properties, press enter. Each time you press enter, you will move from field to field. Once you are at the beginning of a field, you can simply type the value you would like to change. Follow the instructions on the lower left to change the configuration.

Make sure that you set “Active” to “Yes”. Otherwise the RT314 will not generate syslog messages. Under “Syslog IP Address” type the IP Address of the WinSyslog machine. Please note that you **must** use an IP address – the computer name will not work. Under “Log Facility” select the facility the messages will be sent with. The RT314 does support only LOCAL_1 to LOCAL_7 – other facilities are not supported. If in doubt, leave this setting at “Local 1”.

Under types, select which events will be sent via syslog. All those with “Yes” configured will be sent.

Please see the RT314 manual for details.

Finally, press enter to confirm your configuration choice. This will store and active the new configuration and return you to the "Log and Trace" menu. There, press, ESC to return to the "System Maintenance" menu and ESC once again to return to the main menu. There type "99" and enter to exit the RT314 configuration utility.

Please note that telnet will display a "Connection to host lost" message – this is no error but the expected behaviour.

This procedure concludes the configuration of the RT314. It will now generate syslog messages that can be received by WinSyslog.

HP JetDirect Interfaces

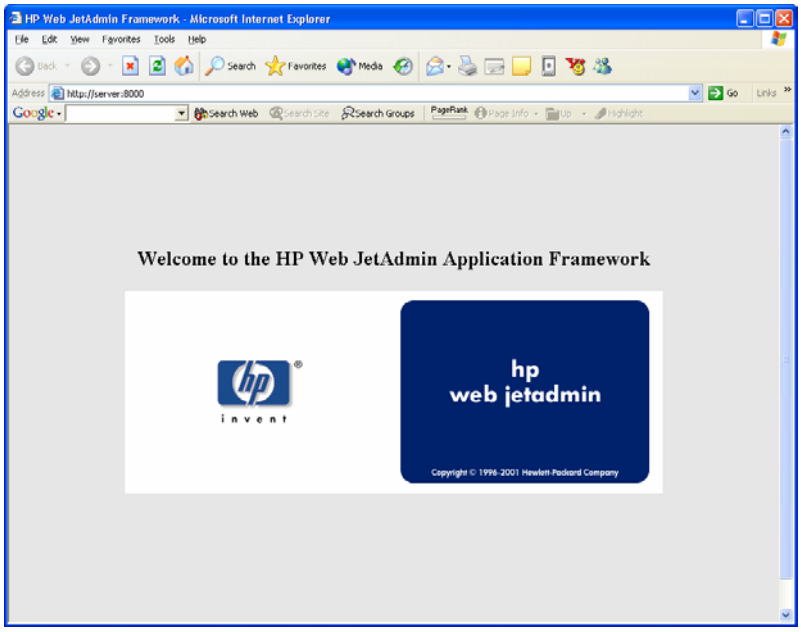
JetDirect interfaces are network print server. They are used intrnally in printers like the successful HP LaserJet series. They JetDirect is also available as external boxe to connect any brand of printer to the nework.

The HP JetDirect interfaces support syslog protocol. To the best of our knowledge, they send status as well as print job information via syslog protocol. Status notifications include things like toner low or out of paper. Print job information includes data on completed an aborted print jobs.

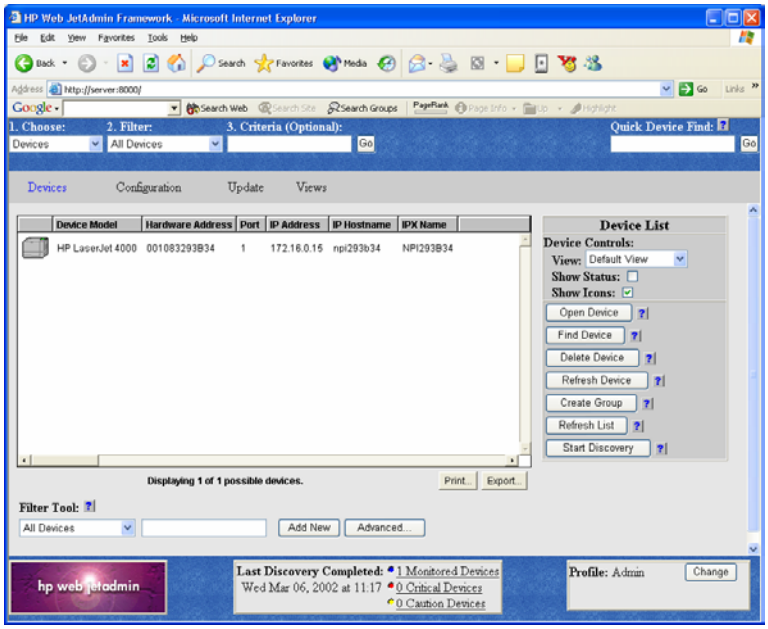
The JetDirect Interface can be configured via the so-called HP JetAdmin program. In our sample, we use the web-based JetAdmin tool (HP is actively promoting the web version today).

In our sample, we have a very basic configuration. The HP Web JetAdmin is installed on a server with the surprising name "SERVER". The printer we are configuring has the also surprising name "HP LaserJet 4000". The syslog server service is running on a machine with IP 10.0.0.1. In the sample, we configure the JetDirect interface to send syslog messages to this central server. We assume that you are already familiar with the HP Web JetAdmin program. Please note that the menues shown below can be slightly different depending on the HP Web JetAdmin version and the actual printer or JetDirect Interface model.

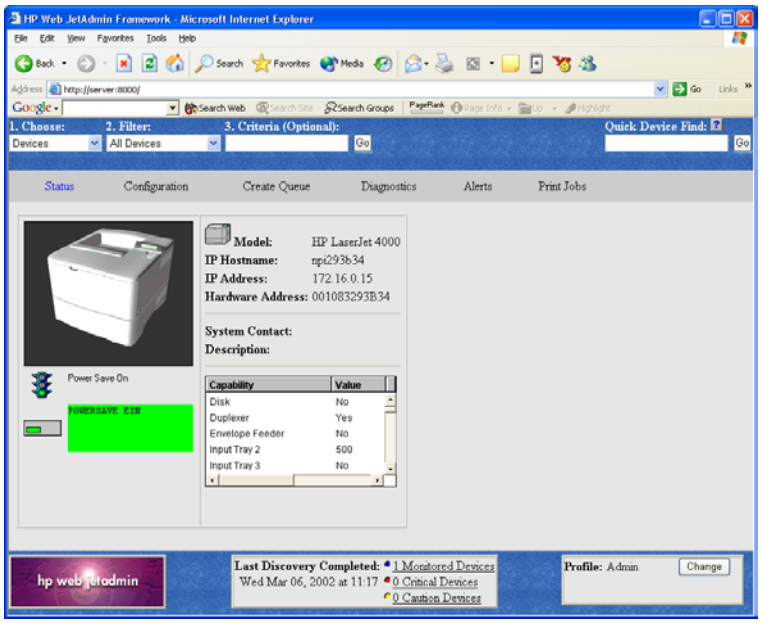
First. start the HP Web JetAdmin by pointing your browser to http://server:8000. This is the default address for Web JetAdmin. This will bring up the HP web interface.



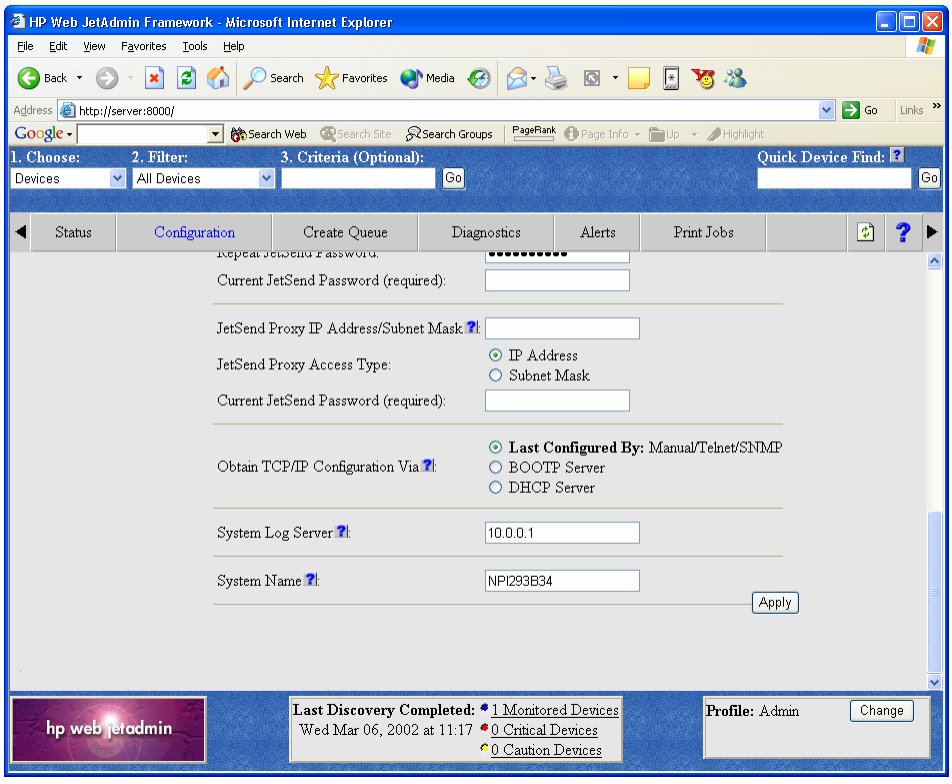
Click on the jetadmin logo and click the continue button that pops up. Please note that depending on your browser settings a number of Java security warnings pop up. You need to allow execution of the applets, otherwise JetAdmin does not work. Continue until you reach the main menu:



Double-click the printer. A screen like to following appears:

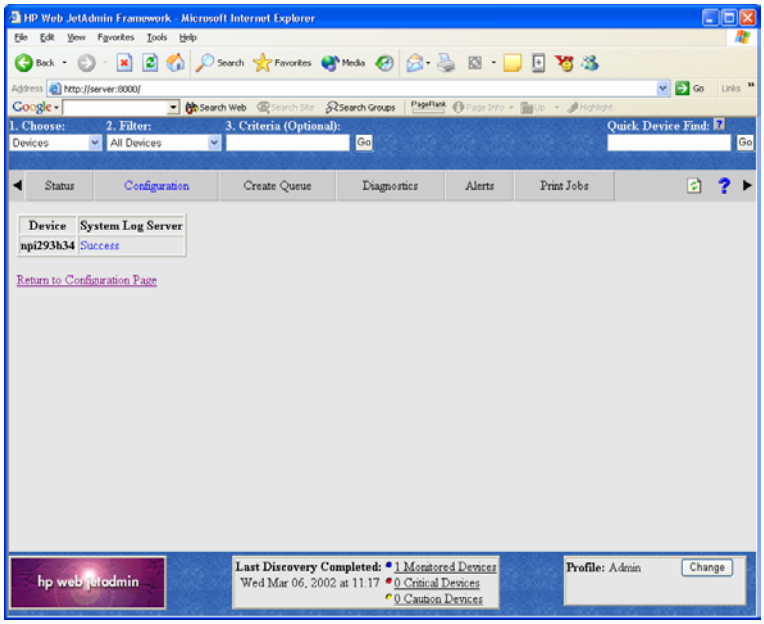


Click on the "configuration" tab. Then, select "network" in the left-hand menu.



Find the "System Log Server" entry. Here, you must enter the IP address of the system the syslog server service is running on.

After doing so, press “Apply”. You will be directed to a “success” page:



The syslog server address is now set and syslog message logging activated. You can now either return to the configuration menu or select any option in the menu available.

This procedure concludes the configuration of the HP JetDirect Interface. It will now generate syslog messages that can be received by the syslog server service.

Cisco PIX

Cisco’s PIX is a well known firewall appliance. It is highly scalable, from a small office or home environment up until an enterprise environment. PIX is very widely used.

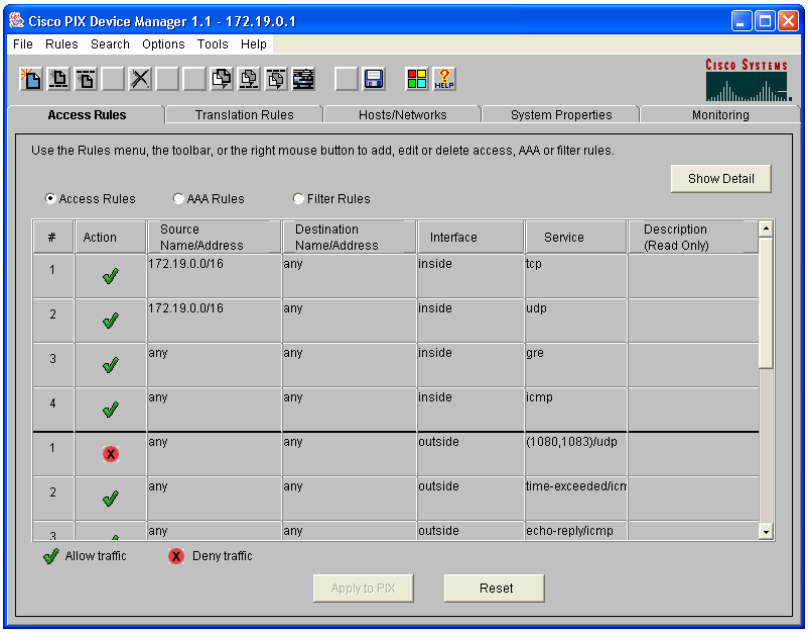
Cisco’s PIX supports syslog both over TCP and UDP. While WinSyslog supports both of these protocols, we will focus on UDP in our step-by-step guide as this is the standard protocol. So if you would like to consolidate logs from multiple devices and one of them is a PIX, you will probably take the syslog over UDP route.

PIX can be configured either via a command line interface or the so-called PIX Device Manager (PDM), an HTML configuration application that comes with the PIX. Typically, PDM is used and as such we focus on it in our step-by-step guide.

First, start PDM by pointing your Java-Enabled web browser to the PIX. Important: Use a HTTPS URL. This is badly documented by Cisco. Using http instead of https will cause your connection to fail! If, for example you PIX has the internal IP address of 172.16.0.1, use the following URL:

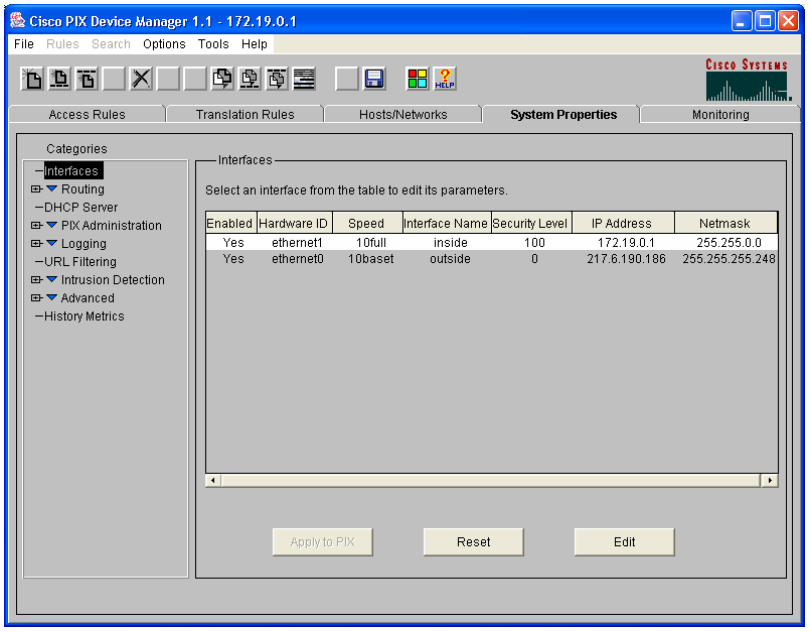
https://172.16.0.1

Once this is done, the PDM opens. Most probably, a number of Java security and certificate related questions open. Please allow the product to proceed. Also, a number of browser windows open. Finally, you should see a window similar to the following:

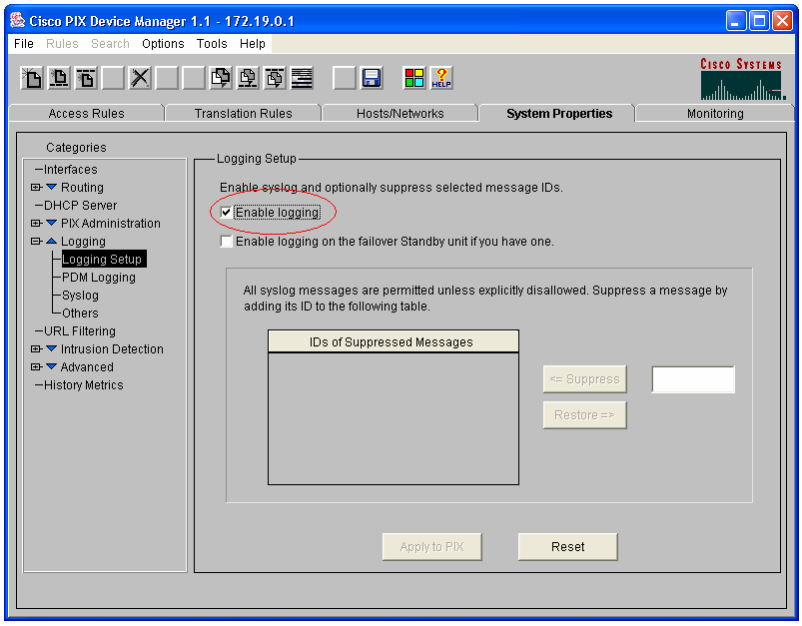


PDM Start Screen

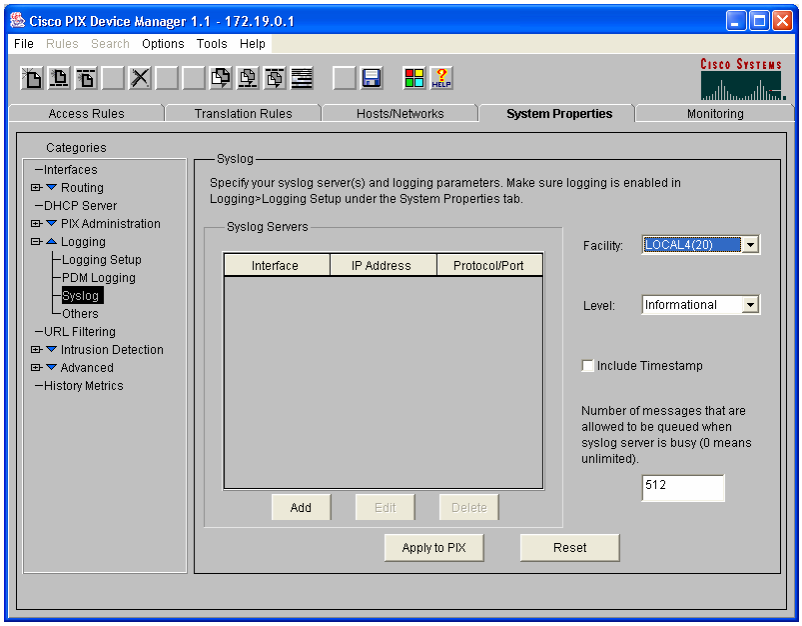
Now, switch to the system properties tab:



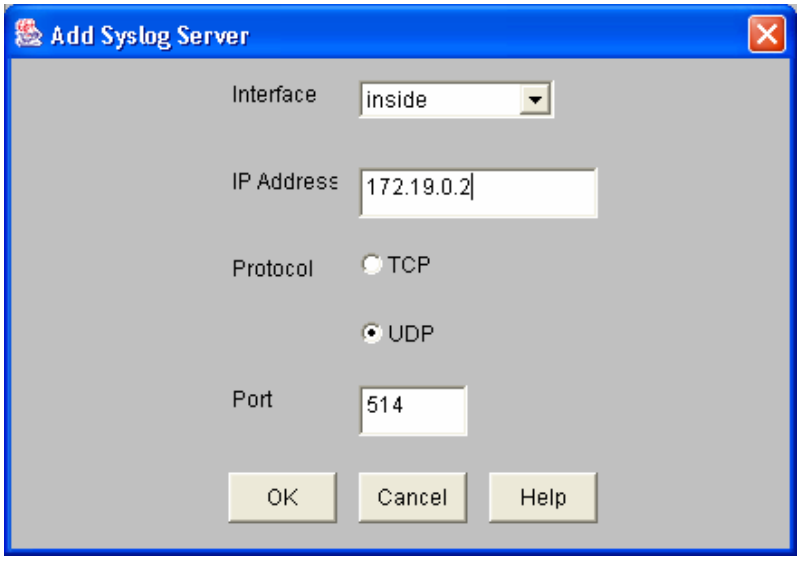
Next, expand “Logging” in the treeview and then select “Logging Setup”. A screen similar to this one appears:



Make sure the "Enable Logging" box is checked as in the screenshot. Then, select "Syslog" in the treeview. This brings you to the page where syslog servers can be configured:



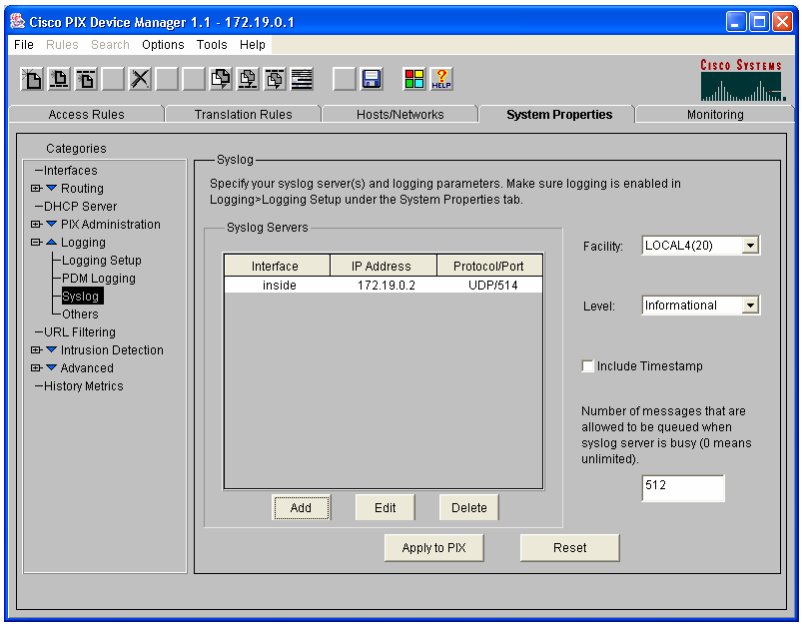
In the above example, no server is configured so far. This is the default setting for a freshly installed PIX. We will now configure a syslog server at IP 172.19.0.2. Press "Add" and the following dialog appears:



Typically, your syslog server will reside on the internal network. As such, leave the interface at “inside”. Then enter the IP Address of your syslog server into the field “IP Adress”. In the screenshot, this has already been done. Next, make sure UDP is selected as protocol. The port value of 514 is the default and also the standard. There should be little need to modify it. If you do, make sure you fully understand the implications as a wrong port can disrupt traffic.

Of course, if you would like to use TCP logging, you can do so. However, in this case Winsyslog client must be configured to have at least one syslog listener running at the specified TCP port. Also please note that other products do typically not support syslog over TCP and as such messages from these devices can not be received by a syslog over TCP receiver.

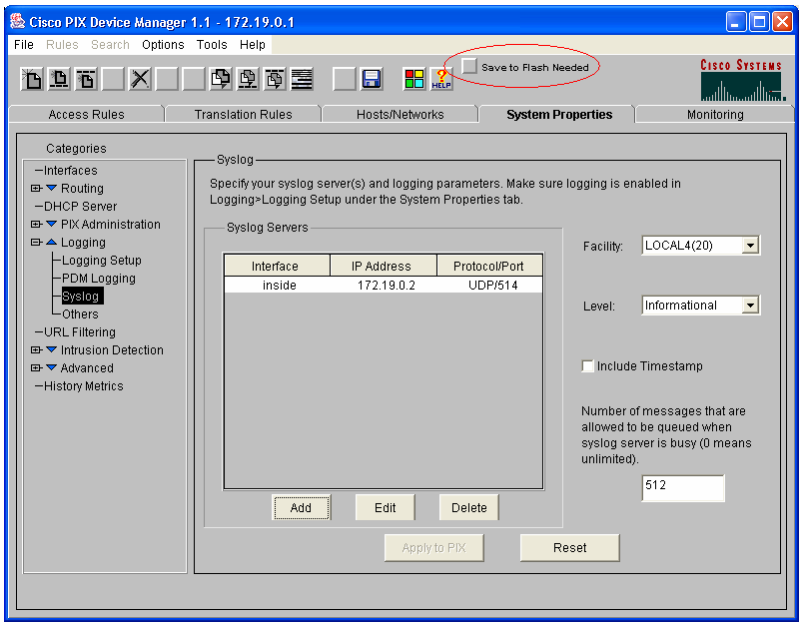
After configuring the syslog settings, be sure to press OK to return to the PDM main screen:



Here, you can modify the syslog facility and level as well as inclulde a PIX timestamp – see settings on the right.

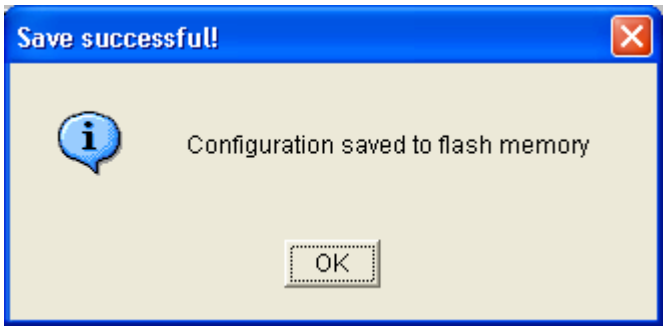
Important: the configuration you have created has not been saved so far! To save it, you must press the "Apply to PIX" button. Depending on your configuration and PIX model, the "Apply" can take some time.

Once the "Apply" is finished, you see the following screen:



Please note the new "Save to Flash Needed" button. This one can easily be overlooked. When it is present, a new PIX configuration has been created but not permanently saved on the PIX. **So you need to press "Safe to Flash Needed" in order to complete your configuration!** If you forget the step, the PIX will either not forward syslog messages at all or stop doing so after the next PIX reboot.

Make sure that you see the following dialog before continuing:



This concludes the basic configuration of your PIX. You should now receive syslog messages on the configured syslog server. You can now close Cisco's PDM. Of course, you can return at any time to change configuration settings or enable syslog messages to additional syslog servers you have created.

Other Cisco Products

All Cisco products we know support logging via syslog. This article covers all devices that use IOS (e. g. routers and switches). Unfortunately, this is not a full step-by-step guide as the others are. We are working to create a more verbose version of the Cisco guide – but we still decided to leave it in here, as it possible is useful for many users.

Syslog logging needs both to be configured as well as turned on. To configure, you must be in enable mode (see your Cisco documentation on how to enter enable mode). Then switch to configuration mode (the command is "configure terminal" or "conf t" as abbreviation). First of all, you need to specify the syslog host that the messages should be send to. This is the name or IP address of the system WinSyslog is running on. Though a DNS-resolvable name can be used, we strongly recommend using the IP address directly. If your machine has the address "195.123.45.6" then the command is "logging 195.123.45.6". Next, logging needs to be turned on. This command is "logging on". Then exit from configuration mode and save the new configuration.

This setting enables syslog logging for common messages (e. g. router configuration and startup). If you would like to have traffic-related logging activated, you need to create traffic filter rules that specify the "log" option and apply them to the interface you are interested in.

More and detailed information can be found at Cisco's web site under the "logging" command. Please note: this link is to one of Cisco's product documentation areas. You might want to search the Cisco site to find information specific to the product (router, switch, firewall, etc.) you are using

Using Interactive Syslog Server

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

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

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

Launching the WinSyslog Interactive Server

To run the Interactive Syslog Server, click the " Interactive Syslog Server" icon present in the WinSyslog program folder located in the Start menu.

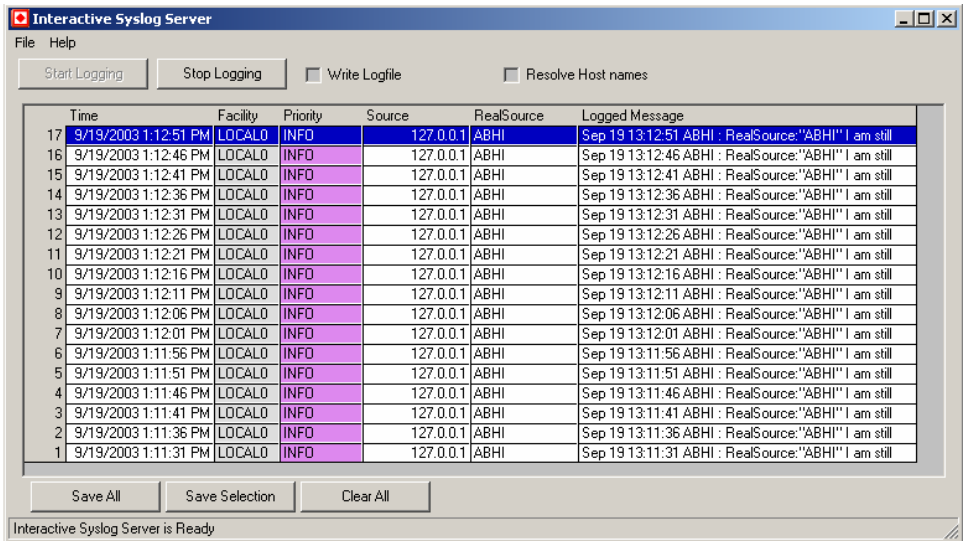
It can also be launched from the command prompt:

- Open a Command Prompt window
- Change to the drive and directory where the WinSyslog software is installed (default: "\Program Files\WinSyslog")
- Type " InteractiveSyslogServer.exe " and hit enter.

The Interactive Logging

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

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



Start / Stop Logging Buttons

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

Write Logfile

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

Resolve Host Names

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

Save All

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

Save Selection

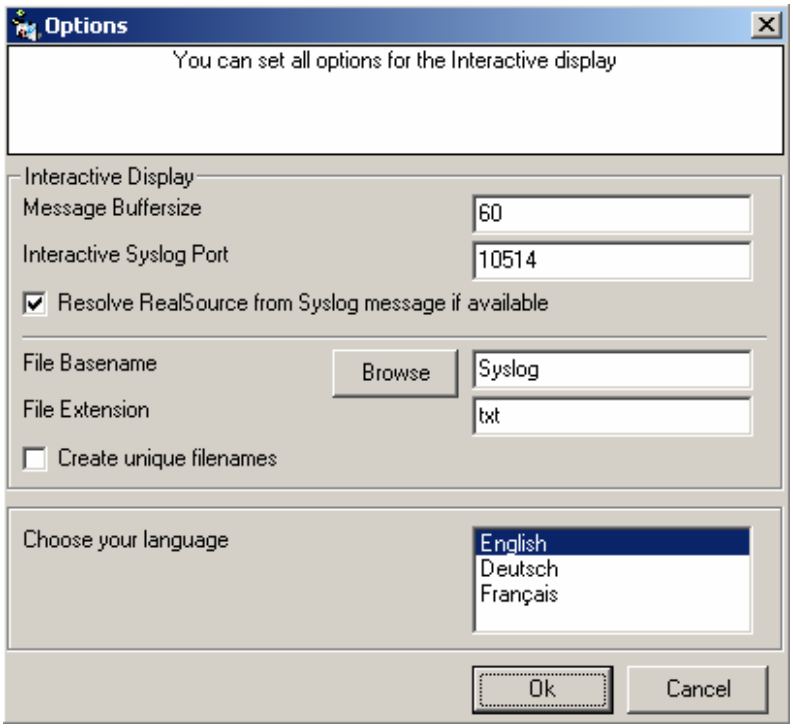
Also saves a comma-delimited file. However, only messages selected (highlighted) will be written to the file.

Clear All

Erases all messages from real-time display.

Interactive Syslog Server Options

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



Message Buffersize

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

Interactive Syslog Port

The UDP port the real-time display listens to. 0 is default from system services database. Most installations can leave it at 10514.

File Basename

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

File Extension

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

Create unique filenames

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

Language

The Interactive Syslog Server is multilingual by desing. Select the user interface language here.

Languages are set on a per user basis. They can be switched instantly without the need to restart.

Additional languages might be made available. Please check www.winsyslog.com from time to time. If you are interested in other languages and volunteer to provide translation services, please email info@adiscon.com. We will gladly help.

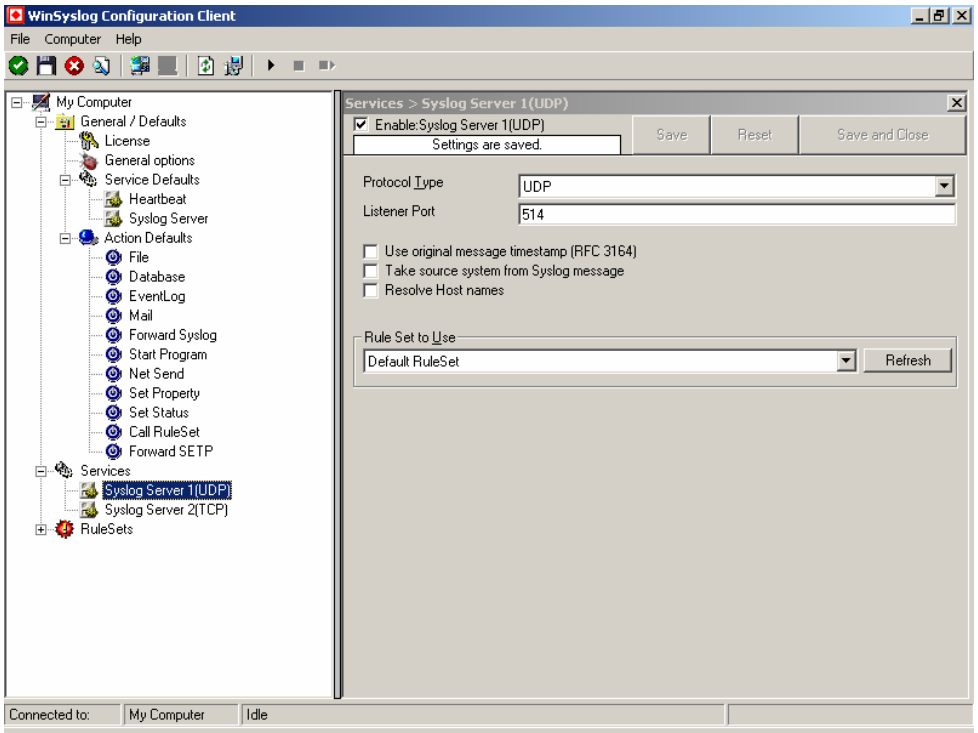
Configuring WinSyslog

WinSyslog is easy to use and powerful.

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

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

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



WinSyslog Configuration Client

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

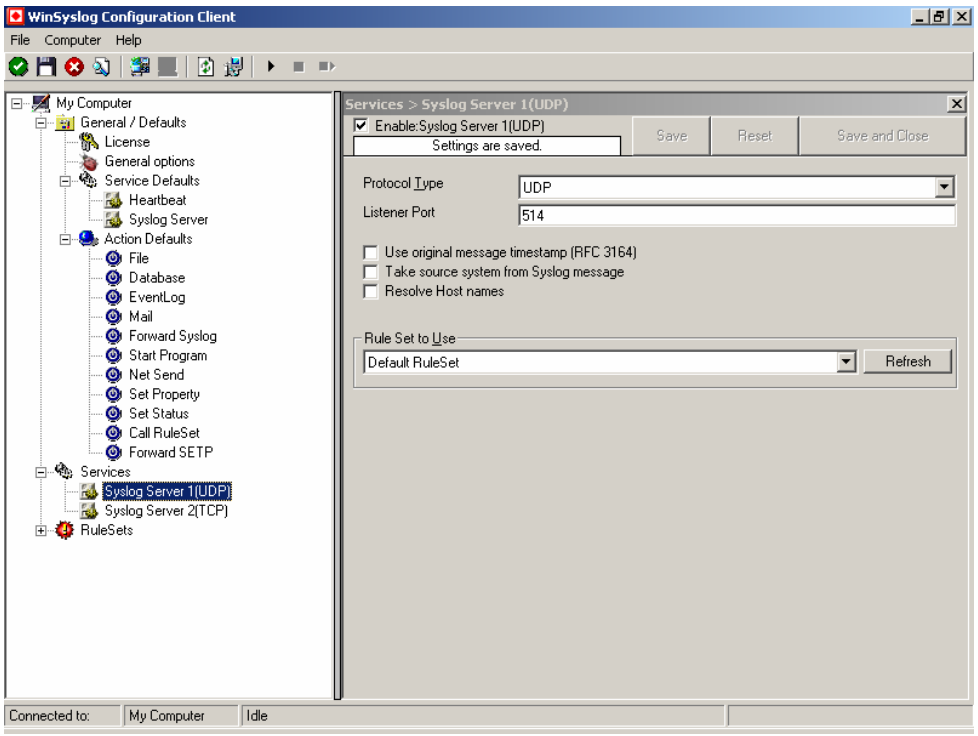
the sample above, the right hand side displays the specific parameters for a rule criterion.

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

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

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

The service definition looks like this:



WinSyslog Configuration Client - Service Definition View

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

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

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

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

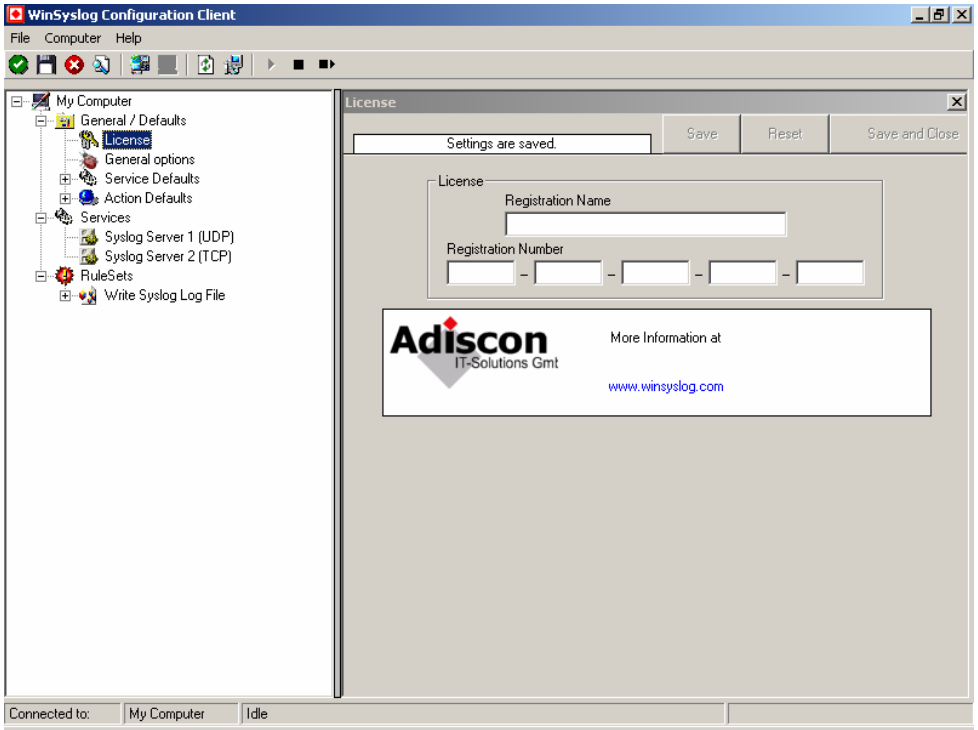
Beneath each rule set are the individual rules. As described in “Rules” on page 54, a rule’s position in the list is vitally important. Rules at the top of the rule set are executed before those further down. To move a rule up or down, simply right click it and select “move up” or “move down” from the pop up menu.

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

The following sections describe each element’s properties.

License Options

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



License Option Parameters

Registration Name

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

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

Registration Number

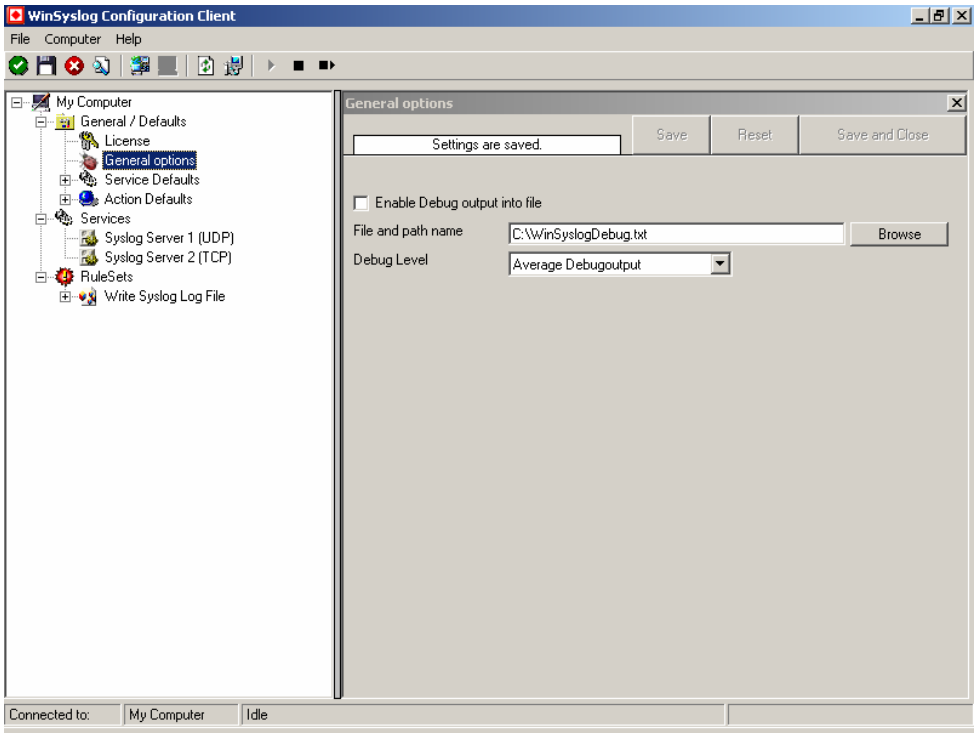
This number is provided by Adiscon. It is valid for a specific registration name. Be sure to enter the correct registration number. The client will detect invalid registration numbers and report and corresponding error.

General Options

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

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

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



Debug OptionsParameters

Enable Debug output into file

If checked, the debug log is enabled and written as the service operates. If unchecked, no debug log is written.

For performance reasons, it is highly recommended that this box is unchecked during normal operations.

File and path name

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

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

Debug Level

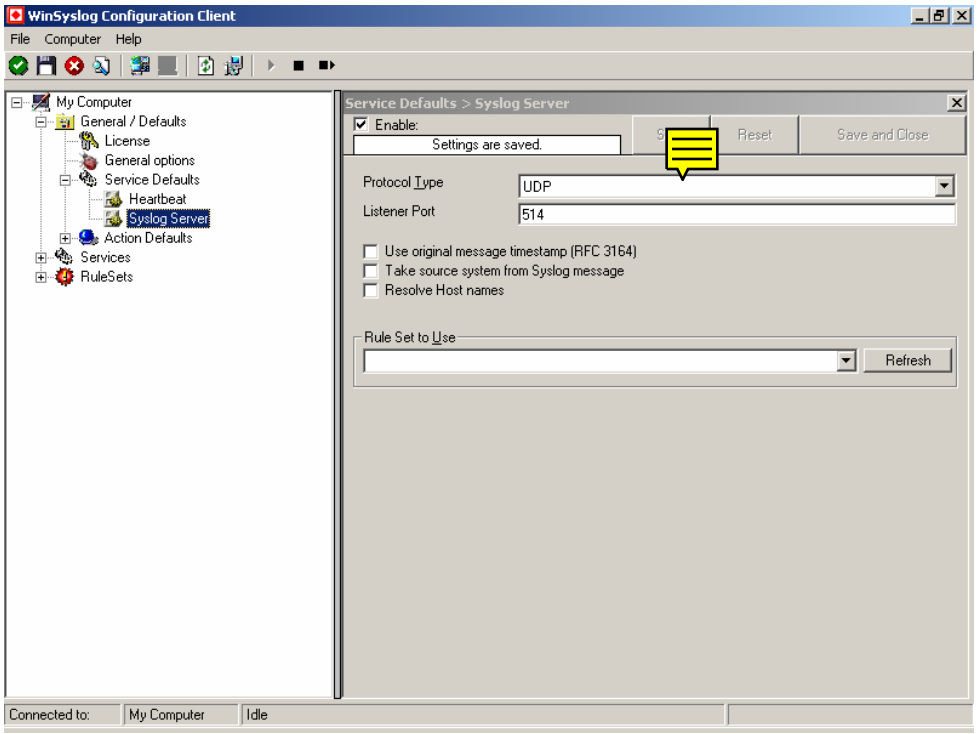
This controls the amount of debug information being written. We highly recommend only selecting “Minimum Debugoutput” unless otherwise instructed by Adiscon support.

Services

The WinSyslog product does support syslog listener services, only. Additional services are available with the other members of the Adiscon MonitorWare line of products.

Syslog Server

Configures a syslog server service.



Protocol Type

Syslog messages can be received via either UDP or TCP. One listener can only listen to one of the protocols. Typically, syslog messages are received via UDP protocol, which is the default.

Listener Port

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

Use Original Message Timestamp

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

Take source system from Syslog message

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

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

Resolve Hostnames

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

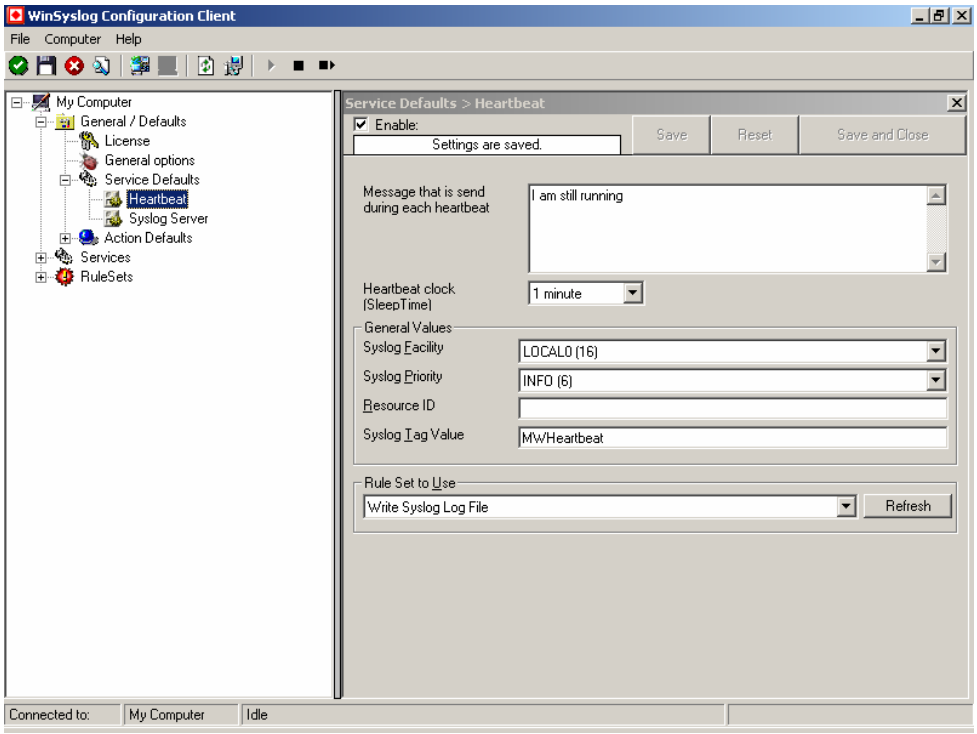
Please note that this setting does have no effect if the "Take source system from Syslog message" setting is checked. In this case, the message is always taken from the syslog message itself.

Default Rule set Name

Name of the rule set to be used for syslog server services. The rule set name must be valid.

Heartbeat

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



Message to Send

This is the message that is used as text inside the information unit. Use whatever value is appropriate. There is no check inside WinSyslog for a specific value.

Sleep Time

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

Syslog Facility

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

Syslog Priority

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

Syslog Tag Value

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

Resource ID

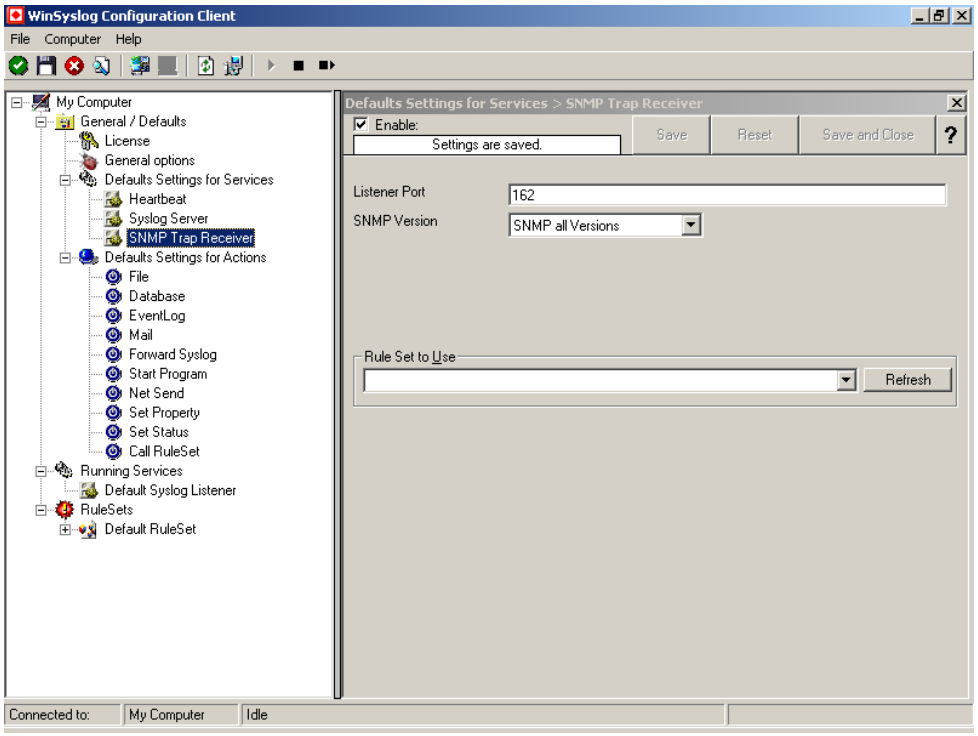
The resource id to be assigned to events created by the heartbeat process. Most useful if the message shall be forwarded to a syslog daemon.

Rule Set to Use

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

SNMP Trap Receiver

Configures a SNMP server



Listener Port

The port the SNMP server listens on. The typical (standard) value is 162. This should be changed only if there is a definite need for it. Such a need typically arises from security concerns.

SNMP Version

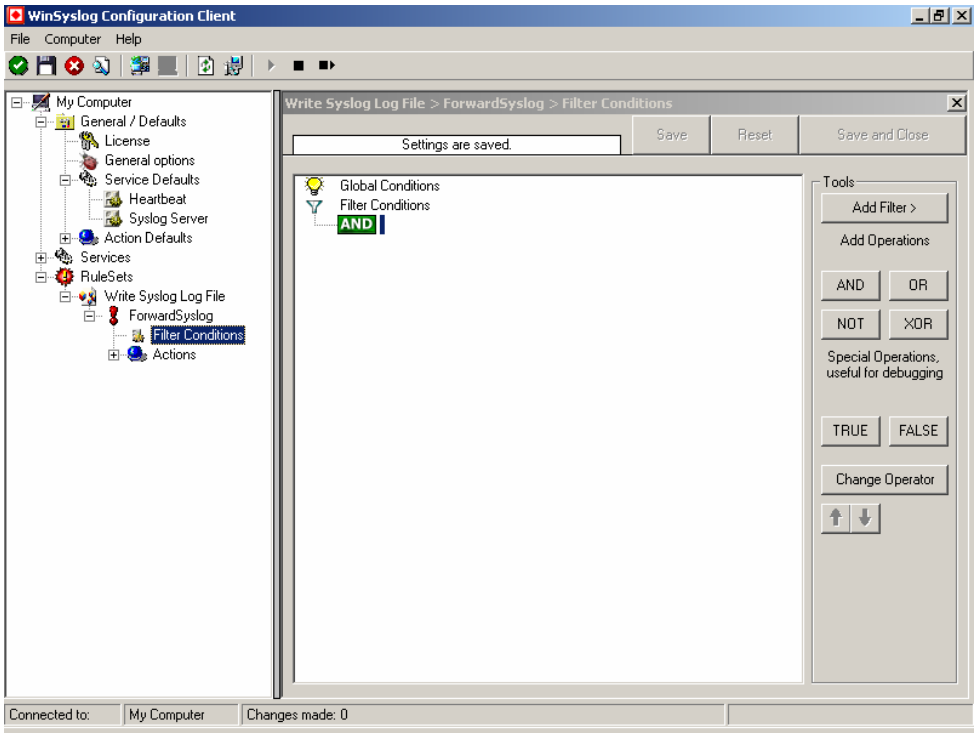
This can be used for selecting SNMP version. Either you can select any of the two (1 & 2C) versions or you can choose all versions.

Rule Set to Use

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

Filter conditions

The filter conditions dialog box contains all conditions for a given rule. For details on how filter conditions are evaluated, please see “Filter Conditions” on page 51.

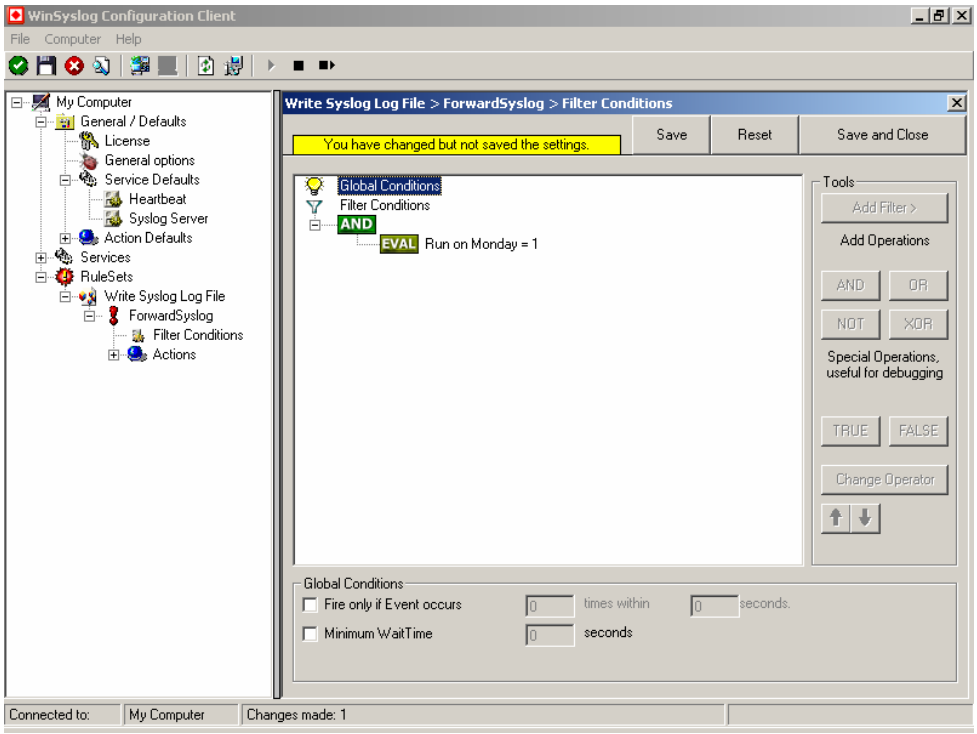


In general, Filter conditions specify **when** to apply a rule. If the filter condition evaluates to true, the rule containing those conditions is treated as matching and the actions specified in that rule will be carried out.

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

Global Conditions

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



Fire only if Event occurs

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

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

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

Minimum Wait Time

This filter condition can be used to prevent rules from firing to often. For example: a rule might be created for a particular event, which will take some time to complete. So in order to avoid re-invocation of the same rule for the same event (while the rule for the same is being carried out) you can use “Minimum Wait Time”.

Operations

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

AND

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

OR

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

NOT

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

XOR

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

TRUE

Useful for debugging, will just return TRUE.

FALSE

Useful for debugging as well, will return FALSE.

General

These are non-event log specific settings.

Source System

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

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

Message Content

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

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

This filter is of type string.

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

The following filters are available in detail:

Start time (Type=Time)

End Time (Type=Time)

Run on Monday (Type=Boolean)

Run on Tuesday (Type=Boolean)

Run on Wednesday (Type=Boolean)

Run on Thursday (Type=Boolean)

Run on Friday (Type=Boolean)

Run on Saturday (Type=Boolean)

Run on Sunday (Type=Boolean)

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 InformationUnitType available (shown below).

Syslog (Type=Boolean)

Heartbeat (Type=Boolean)

Syslog

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

Syslog Priority

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

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

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.

Actions

Actions are carried out when the filter conditions of a given rule match.

File Options

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

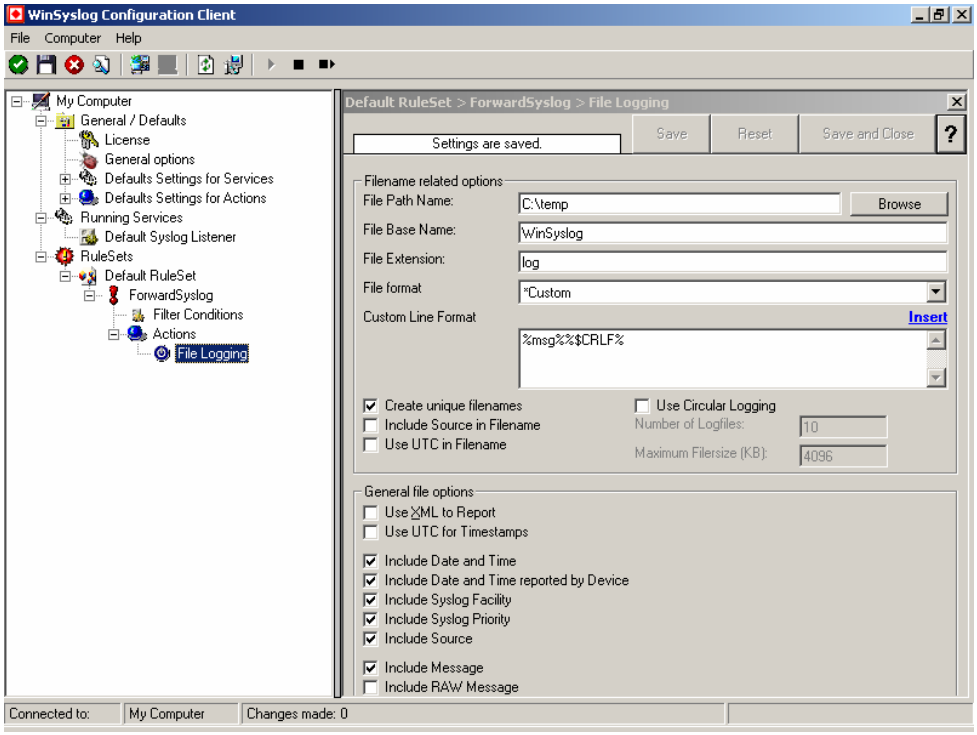
File logging is used to write text files of received messages. One file per day is written. New entries are appended to the end of the file.

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

The filename is build as follows:

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

with the parameters in brackets being configured via the dialog.



File Logging Options

Create unique Filenames

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

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

Use UTC in Filename

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

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

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

Please note that this setting does affect the file name creation only. The dates recorded inside the file are controlled by a different setting.

File Path Name

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

File Base Name

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

File Extension

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

File Format

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

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

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

The “*Custom” Format is used to assign the format in which you want the logging is be carried out. This is a default option given with the format %msg%%$CRLF% : %msg% is the message & %$CRLF% is for carriage-return Linefeed (breaking the line)

You can also use any constant values with some predefined variables. These Predefined variables can be inserted using “Insert”

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

nsert

This can be use to insert some predefined variables (in the log file these variables are replaced with their corresponding values)

General:

Source	-	%source%
Message	-	%msg%
Time Generated	-	%timegenerated%
Time Reported	-	%timereported%
Error – mnuFilterInf	-	%iut%

Syslog:

Facility	-	%syslogfacility%
Priority	-	%syslogpriority%
Syslog Tag	-	%syslogtag%

SNMP Trap:

Version	-	%snmp_version%
Uptime	-	%snmp_uptime%
Version 1 Parameter		
Community	-	%snmp_community%
Enterprise	-	%snmp_enterprise%
Generic Name	-	%snmp_generic_name%
Version 2 Parameter		
SNMP Variable 1	-	%snmp_var_1%
SNMP Variable 2	-	%snmp_var_2%
SNMP Variable 3	-	%snmp_var_3%

Include Source in Filename

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

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

Use XML to Report

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

Use UTC for Timestamps

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

Include <Fieldname>

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

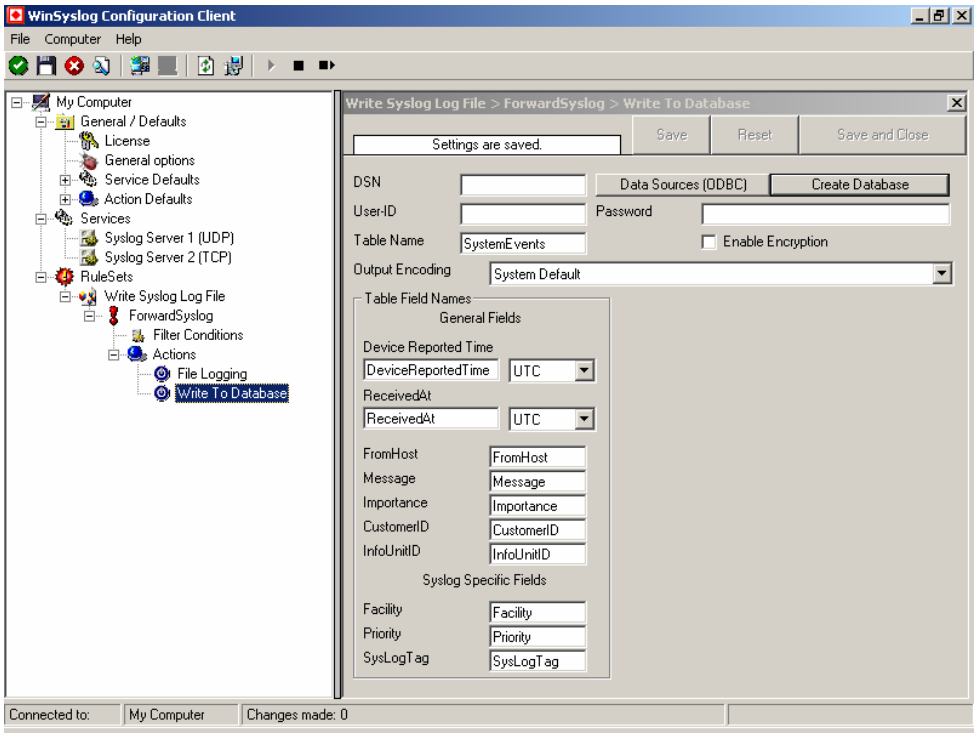
Please note the difference between the "Date and Time" and "Date and Time reported by Device". Both are timestamps. Either both are written in local time or UTC based on the "Use UTC for Timestamps" check box. However, "Date and Time" is the time the message was received by WinSyslog. Therefore, it always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

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

The "Include Message" and "Include RAW Messge" fields allow to customize the message part that is being written. The raw message is the message as it was received by WinSyslog – totally unmodified. This might be useful if a third party application is expecting raw syslog entries. The message itself is just that part of the syslog message that is being parsed as message, that is without e.g. host information or a tag value. Please note that we recommend selecting only one of these options, as otherwise two message fields will be written. Similarily, if non is selected no message is written at all. Please note that we support these configurations, too – there might be a legitimate need for them.

Database Options

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



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

DSN

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

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

User-ID

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

Password

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

Enable Encryption

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

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

Table

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

Output Encoding

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

Click here for a sample screen-shot.

Table Field Names

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

Important

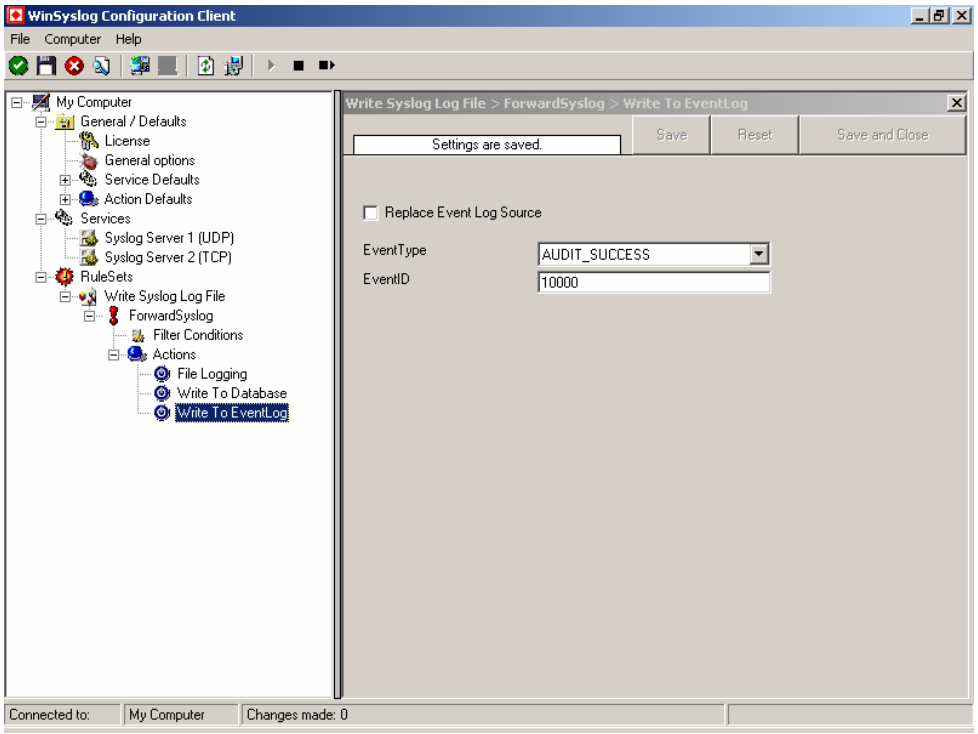
The default name for the message field - "Message" is a reserved name on Sybase database systems. If you would like to log to a Sybase database, you must change that field name. Otherwise, you will receive an ODBC error (visible in NT Event Viewer). We are unfortunately not able to change the default,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

The database conforms to the Common MonitorWare Database Format

For a specification of the database format and samples provided, please see “Database Format” on page 59.

Event Log Options

This tab is used to configure the logging to the Windows NT / 2000 or XP event log. It is primarily included for legacy purposes. There are many better ways of logging events in WinSyslog.



Replace Event Log Source

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

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

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

Click here for a sample screen-shot.

EventType

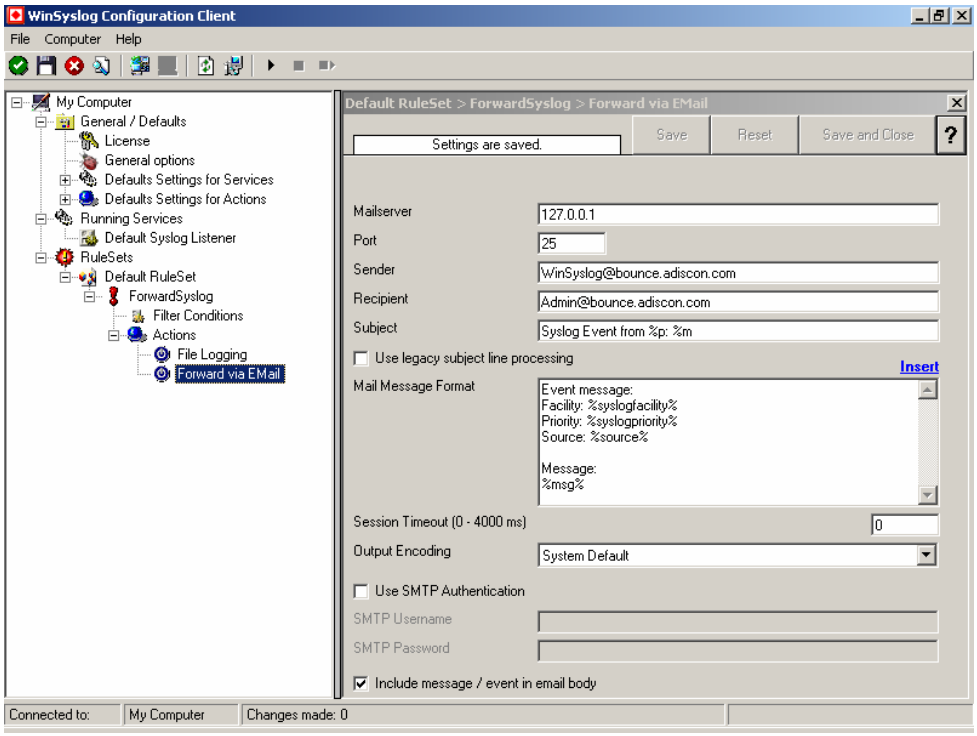
The type – or severity – this log entry is written with. Select from the available Windows system values.

EventID

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

Mail Options

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



Mailserver

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

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

Port

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

Sender

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

Recipient

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

Subject

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

The following replacement characters can be used inside the subject line:

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

In the example above, replacement characters are being used. If a message "This is a test" were received from "172.16.0.1", the resulting email subject would read:

Syslog from 172.16.0.1: This is a test

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

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

Mail Message Format

You can use "Mail Message Format" for sending a formatted mail. Write the format using constants or/and some predefined variables (which can be accessed using "Insert") in the provided text box.

nsert

This can be use to insert some predefined variables (in the log file these variables are replaced with their corresponding values)

Session Timeout

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

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

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

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

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

Use SMTP Authentication

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

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

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

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

Click here for a sample screen-shot.

Include message / event in email body

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

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

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

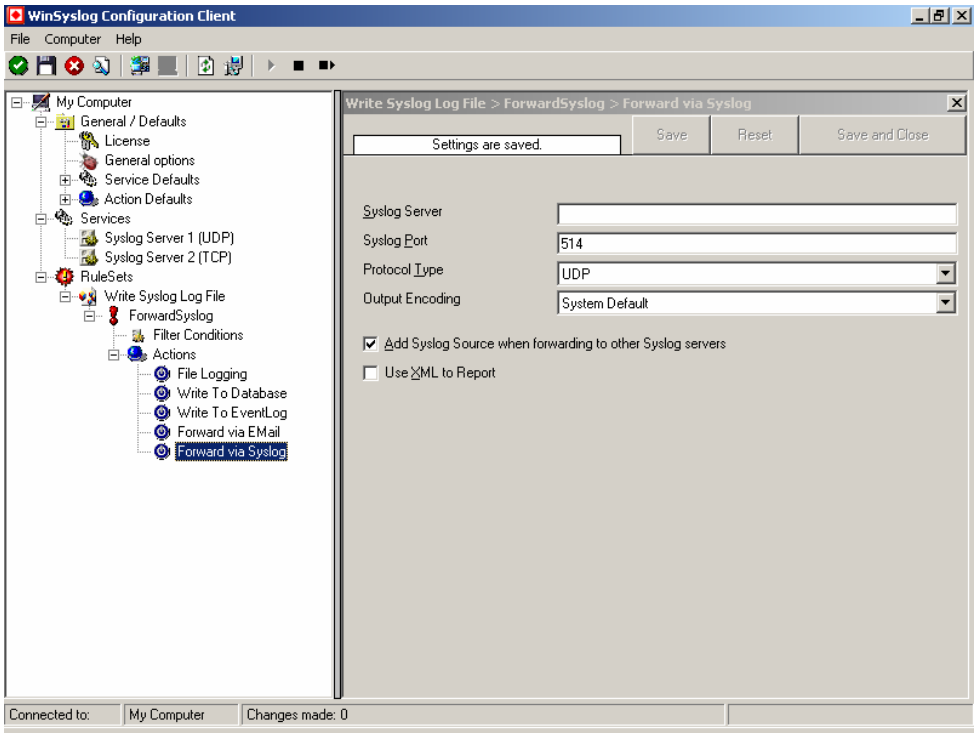
Use XML to Report

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

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

Forward Syslog Options

This dialog controls syslog forwarding options.



Forward Syslog Properties

Syslog Server

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

Syslog Port

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

Protocol Type

The Agent can forward messages via either UDP or TCP. The syslog standard allows UDP delivery only. This is also the default. Change it to TCP only if you have a very good reason to do so and you know the receiving server is capable of accepting syslog over TCP.

Output Encoding

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

Click here for a sample screen-shot.

Add Syslog Source

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

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

Click here for a sample screen-shot.

Use XML to Report

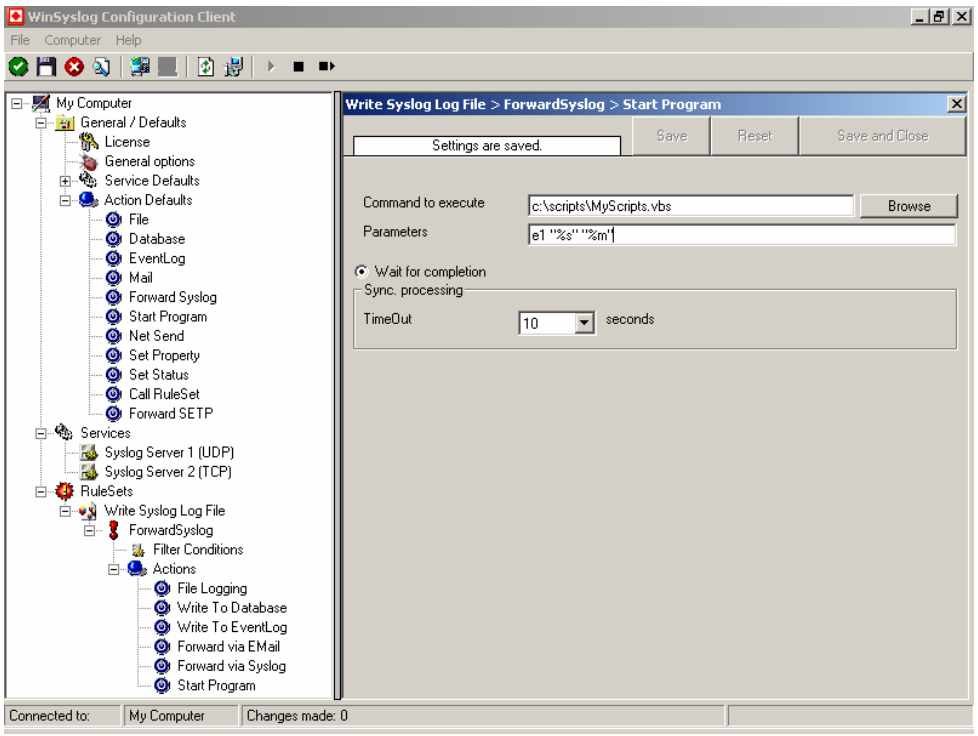
If checked, the forwarded syslog message is a complete XML-formatted information record. It includes additional information like timestamps or originating system in an easy to parse format.

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

Start Program

This dialog controls the start program options.

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



Start Program Dialog

Program to execute

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

Parameters

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

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

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

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

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

Time Out

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

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

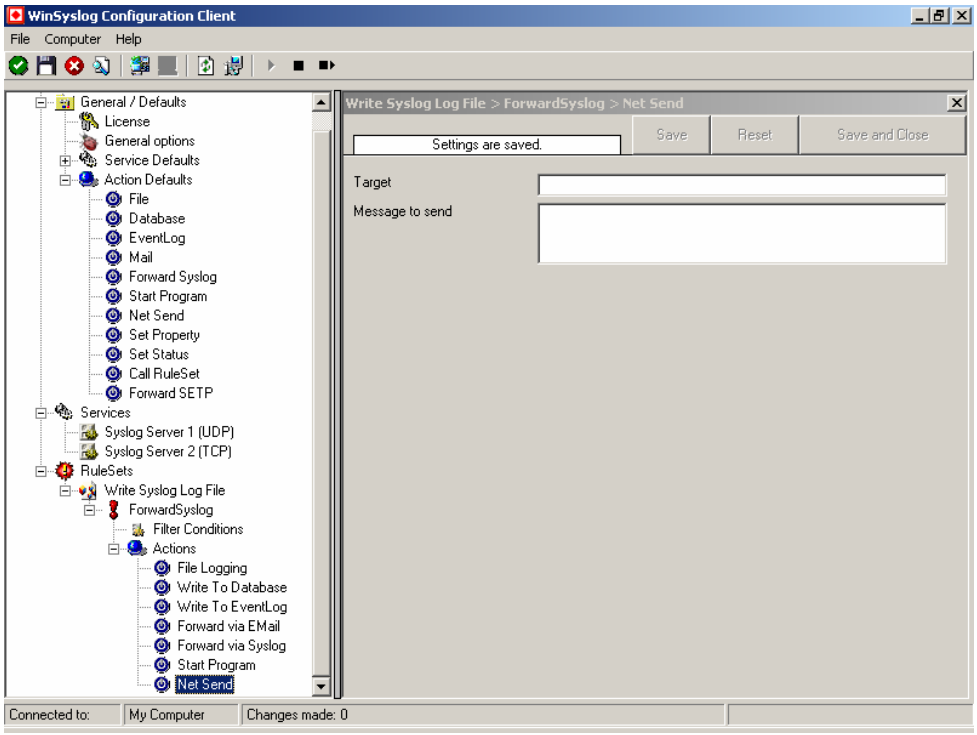
Important: Even though the timeout value can be as high as 30 seconds, we strongly recommend limiting the run time of external program to below 5 seconds. Otherwise, they could affect the overall performance too much. If the average run time is 5 seconds, the default timeout of 10 seconds ensures that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

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

Net Send

This dialog controls the net send options.

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



Net Send Dialog

Target

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

Message to send

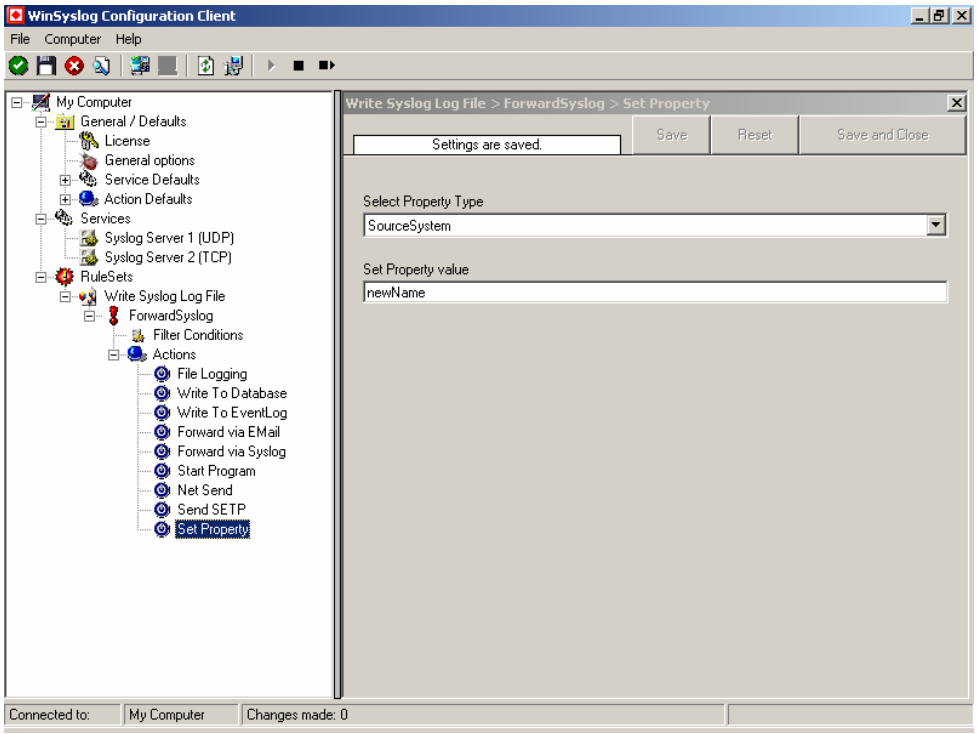
This is the message that is sent to the intended target.

Set Property

This dialog controls the set property options.

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

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



Set Property Dialog

Select Property Type

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

Set Property Value

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

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

Getting Help

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

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

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

Frequently asked Questions

For a current list of Frequently Asked Questions (FAQ), please visit

www.winsyslog.com/en/FAQ

The FAQ area is continously being updated. Some of the most important FAQ entries are also included in this manual. However, we recommend using the web site as there might be updates even to the items included in this manual.

I have an invalid source in my received syslog message - what to do?

If I look at the received syslog message source system, I see invalid names like "su", "root" and the like. These correspond to some part of the syslog message. In any case, it is not the real system name. What can I do to receive the correct name?

The problems stems from non syslog-RFC compliant systems. The syslog service does RFC compliant message parsing. Unfortunately, many existing systems are not compliant to the syslog RFC and format the message other then specified. As such, the syslog service picks up an invalid source system - simply because invalid information is where the source system should be.

Fortunately, the syslog server can be instructed to ignore the source system in the syslog message. This is the defaut mode for all installations after 2002-03-20. This is done with the "Take source system from syslog message". If that check box is checked, the source is taken from the message as specified in the syslog RFC. If it is unchecked, it is determined based on the sending system.

Adiscon's experience is that as of this writing only a limited number of systems support RFC compliant message formatting, so we recommend to uncheck this option.

For details on how to configure this, please see “Syslog Server” on page 29.

How to install WinSyslog in silent mode?

Because WinSyslog is using the Windows Installer (MSIE) it is very easy to start the Installation in silent mode.
There are two ways to do it.

1. Using the WinSyslog msi-file (Only possible if Windows Installer is installed on the target machine)
The msi-file has to be started with the following command line options (Using a sample File location):
msiexec /i C:\SetupFileName.msi /qn

2. Using the WinSyslog setup-file (Only necessary if Windows Installer isn't installed - a reboot might be required).
The setup-file has to be started with the following command line options (Using a sample File location):
SetupFileName.exe /v"/qn" /s

For more informations about the Windows Installer command line options see:
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/hh/msi/app_73eb.asp.

High CPU utilization while EventReporter and WinSyslog are running

A high CPU utilization might happen if both WinSyslog and EventReporter are installed on a single machine. It occurs under the following circumstances:

- EventReporter and WinSyslog are running on the same machine.
- EventReporter is configured to send Eventlog entries to WinSyslog (127.0.0.1 for example)
- Some actions defined in WinSyslog are generating an error (Which is also written into the NT Eventlog)

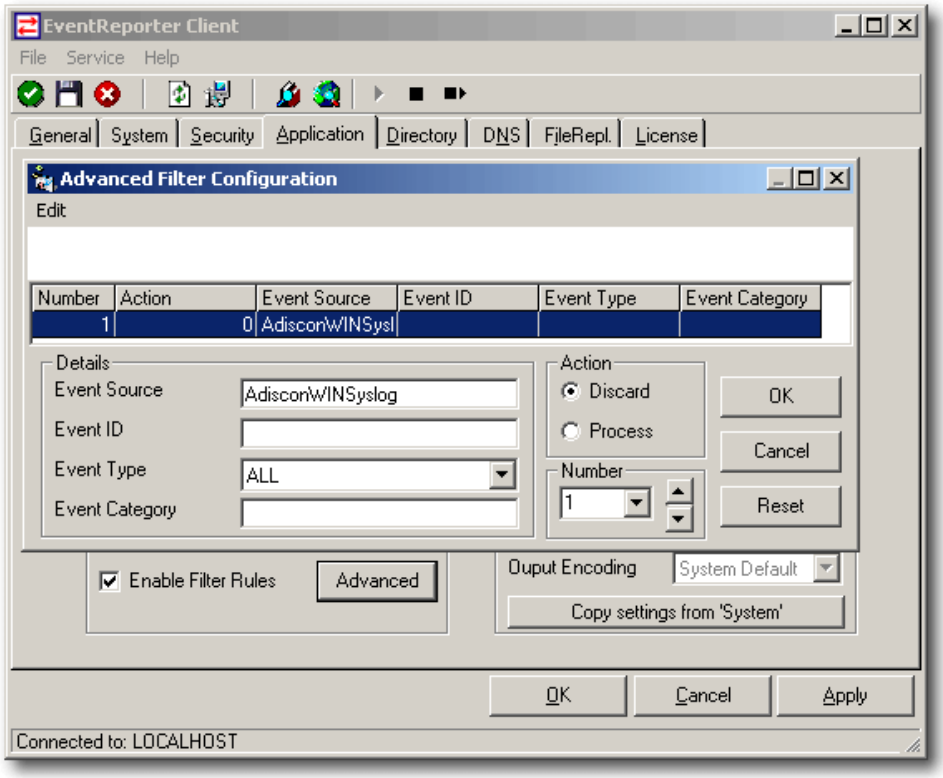
What happens?
EventReporter is sending an Eventlog entry to the local WinSyslog. The Ruleset is processed by WinSyslog and an action (ODBC-Logging for example) causes an error (Maybe the database server is down). WinSyslog writtes this error into the NT Eventlog. Next time EventReporter wakes up, the whole process repeats and that is causing the high CPU utilization.

To solve this problem stop the EventReporter Service and check the WinSyslog Ruleset for any errors and try to solve them. If you don't have success, then please contact Adiscon Support for help.

Are there other Solutions?

Yes there is another solution how to prevent this behavior. You have enable the Filter Rules for Application Logging using the Eventreporter Client. Insert a new Rule the discards all Eventlog entrys with the Source **AdisconWINSyslog**. The only

disadvantage is, that you won't get any Eventlog entrys generated by WinSyslog. See the sample screenshot below:



WinSyslog Web Site

Visit the support area at

www.winsyslog.com/en/support

for further information. If for any reason that URL will ever become invalid, please visit www.adiscon.com for general information.

Support Forum

Share questions and answers with your peers! The forum is also monitored by Adiscon support staff. Find it at

http://erftstadt.adiscon.com/exchange/root.asp?acs=anon

Email

Please adress all support request to

support@adiscon.com

An appropriate subject line is highly appreciated.

Online Seminars

Adiscon offers a selection of online seminars. This selection is continously being expanded. All available seminars can be found at:

www.adiscon.com/Common/SeminarsOnline/

When viewing the seminar selection, please keep in mind that WinSyslog is a member of the MonitorWare line of products. As such, seminars related to the common reporting engine are relevant to WinSyslog, too.

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

Phone

+49-2235-985004 (with "+" being the international dialing prefix, for example 011 in the US).

Phone technical support is limited to customers who purchased support incidents. If you would like to do so, please email info@adiscon.com.

Please note that we are in the Central European Time zone (CET). That is 1 hour east of Greenwich Time. If it is 12pm in New York, it is 9pm at our office location. Our office hours are from 9am to 5pm. So we generally advise US customers to call in early mornings and Asian customers to call in late afternoon.

Fax

Please direct your faxes to

+49-9349-928820

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

Upgrade Insurance

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

To learn more about UpgradeInsurance, please visit

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

Non-Technical Questions

Please address all non-technical questions to

info@adiscon.com

This email alias will answer all non-technical questions like pricing, licensing or volume orders.

Product Updates

The WinSyslog line of products is being developed since 1997. New versions and enhancements will be made available continuously.

Please visit

www.winsyslog.com/

for information about new and updated products.

WinSyslog Concepts

Learn what WinSyslog is made for and made of.

The WinSyslog Service offers advanced monitoring capabilities. Not only it can monitor the system it is installed on, it can also include information received from syslog enabled devices. To fully unleash WinSyslog's power, you need to learn a bit about its concepts. This chapter here has full details.

WinSyslog operates on a set of elements. These are

- Services
- Information Units
- Filter conditions
- Actions
- Rules

It is vital to understand each element and the way they interact. This chapter describes each element in detail. Also, we strongly recommend visiting

http://seminars.adiscon.com/RuleEngine/RuleEngine_files/default.htm

for an online tutorial on the rule engine.

Services

Services inside the WinSyslog Service gather the data that is processed by rules. Each service type reflects a specific set of code inside the WinSyslog Service. For example, a syslog services represents an instance of a syslog server

Typically, there can be multiple instances of the same service running, as long as their configuration parameters do not conflict. There can be multiple syslog servers on a given system as long as they listen to different ports. Consequently, there can be multiple instances of the syslog service be created. For example, there could be three of them: 2 listen to the default port of 514, but one with TCP and one with UDP and a third one listens to UDP, port 10514. All three coexist and run at the same time.

Associated rule sets

Each instance of a service has an associated rule set. This allows easy creation of customized rule sets on a per service basis. Of course, all services can also operate on a common rule set.

All services are executed as multiple threads inside the WinSyslog Service. From the operating point of view, there is only one system service called the "WinSyslog Service". If the service configuration of the WinSyslog Service is modified, the WinSyslog system service needs to be restarted in order to activate the new configuration. Later releases will have some options to automate this task.

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 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.

Most of these elements can be used as filter condition in the rule engine. Information unit specific data elements are not eligible as filter condition. 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 the service type does not directly support them.

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 listener).

Filter Conditions

Filter conditions are used inside the rule engine (described below). They help to decide when a rule is to be carried out. Filter conditions are considered to match of the outcome of the configured comparison operation is "TRUE".

Syslog Priority

For syslog information units, this is the actual syslog priority. If that filter condition is used on non-syslog originated information units, it will be a value mapped on a best effort basis to a syslog priority.

Syslog Facility

For syslog information units, this is the actual syslog facility. If that filter condition is used on non-syslog originated information units, it will be a value mapped on a best effort basis to a syslog facility.

Message content

This filter condition compares the configured string with the content of the message. Wildcards are implicitly applied to the begin and the end of the specified string.

Source System

This is the system a message originated from. It can be used to check for authorized systems to pass messages to the WinSyslog Service.

Information Unit Type

This is based on the type of service that generated the information unit. So with this setting rules can be created that act only on e.g. syslog messages or NT event reports (only supported by EventReporter or the MonitorWare Agent).

Minimum Wait Time

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

Occurrences

This is kind of the opposite of the "Minimum Wait Time". Here, multiple events must come in before a rule fires. Take another example. This time, we use a ping probe (natively available in the MonitorWare Agent – but WinSyslog can receive the alerts generated by a remote 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 the Occurrences filter condition is made for. It waits until a configured amount of the same events occur within a time frame. Only if the count is reached, the condition matches and the rule can fire.

Time

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

Weekdays

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

Actions

Write to File

The message is written to a plain text log file.

Write to Database

The message will be written to the specified ODBC database.

Write to EventLog

The message will be written to the application event log. Please note that the service intentionally does not try to make the message look like it was generated on the local system. This could be very confusing. Instead, it is written inside the message part with standard values for event source and type.

Discard

Please see below (rule engine) for a complete discussion. Effectively, the message will be discarded and any further processing of this information unit be stopped as soon as a “Discard” action is found.

Forward via Syslog

The message will be forwarded to a syslog daemon. UDP forwarding is supported. Future releases will also support TCP forwarding.

Forward via EMail

The message will be forwarded via email. Please note that each message will generate one email message. Messages are not combined to fit into a single mail. Future releases will include a delayed email writer. It will be capable of sending messages after a configured amount of time, effectively allow sending multiple messages within a single email.

Net Send

The message will be forwarded via the Windows “net send” functionality. Please note that the Windows function is not very reliable and requires the user to be logged in. As such, we recommend using “Net Send” only in combination with other actions.

Start Program

The message will be passed to an external process. The command line is specified in the action modifier.

Set Property

Allows modification of the received message itself. Typically used to re-set some values like the system name before the message is passed to further processing.

Rules

Rules are the workhorse of the WinSyslog Service. 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 WinSyslog service.

You might already know something similar to the WinSyslog 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.

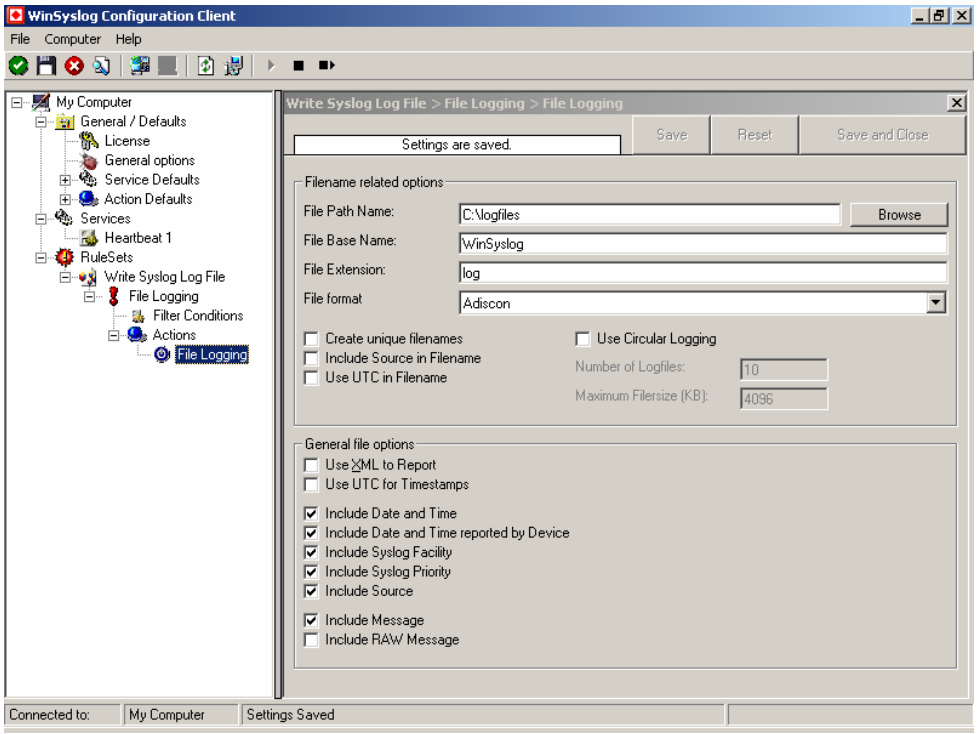
The rule base consists of the rules as configured in the client. The rule engine is the process carrying out the rules. A rule base can contain no, one or an unlimited number or rules. However, if there is no rule at all defined, no action will ever be carried out by the service. Consequently, the client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

A ***rule*** has a description and 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 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 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 WinSyslog Service. 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 conditions. 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 totally 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 can not be modified via the action modifier (e.g. the SMTP server address for email forwarding).

Below find a screenshot of a rule base with a number of rules, filter conditions and action modifiers



Sample Rule Base

But 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 1 to the last one. Each rule is checked to see if it matches. If it does, 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. Have a look at rule number 3 above. It contains the discard action. If a message matches that rule, actions 4, 5 and 6 will not be evaluated. Even if there were a match in these rules, their actions won't be carried out. So what is the discard action good for? It is used to handle common situation 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 each and every message arriving at the WinSyslog Service. 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 WinSyslog 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.

Purchasing WinSyslog

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

The License

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

Which Edition is for Me?

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

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

This includes a feature and price comparison.

How to order

Using the Online Processing System

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

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

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

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

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

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

for details.

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

Direct your orders to:

Adiscon GmbH
Franz-Marc-Strasse 144
50374 Erftstadt
Germany

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

email: order@Adiscon.com

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

Placing a Purchase Order

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

Reference

The WinSyslog Service

The Service operates in the background while your computer is running.

The WinSyslog Service is installed as a system service during setup. It typically runs on each machine being monitored. However, some machines can also be dedicated to run it for housekeeping functions (for example log consolidation).

The WinSyslog Service can be "engine only" installed. In this case, only the service is installed onto a machine. It can be customized either by directly editing the registry or copying a registry snapshot from a machine with installed client. Please note that "Engine Only" installs need a full WinSyslog license.

The WinSyslog service program is called " winsyslg.exe". It is the sole executable that needs to be distributed for mass rollouts.

The Service Account

NT Services must utilize an NT logon account in order to perform their intended tasks. The WinSyslog service is no different. The account initially used by the service is "local system". We recommend retaining this setting.

If for any reason you would like to change the service account, you can do so via the control panel "services" applet (or the "Computer Management" MMC under Windows 2000). However, you need to make sure that the new account has sufficient permissions.

Command Line Switches

The WinSyslog Service supports a limited set of command line switches. These are primarily used for unattended installations or "engine only" installs. These are:

winsyslg –h	Help, displays a short usage notice.
winsyslg –I	Installs the service
winsyslg –u	Removes (uninstalls) the service
winsyslg –v	Displays version information as well as whether or not the service is installed.
winsyslg –r	Runs the service as a forground application (use only if you have specific needs).

Formats

Database Format

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

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

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

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

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

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

Database Samples

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

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

JET (Microsoft Access) Sample

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

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

Microsoft SQL Server Sample

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

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

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

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

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

Version History

Interested how the WinSyslog Service evolved and which features are new to this build? Read it here!

This short history provides some background information about the versions available as well as their pros and cons.

This is user driven software.

Please provide us with your feedback. Many features have become reality with the help of envisioning users!

1.0

This is the initial release. It provides all the basic functionality, has some restrictions:

- there is no configuration program
- logging to stdout only
- does not run as a service

2.0

This is the feature-upgraded release. This release is available as shareware. It contains the following enhancements:

- runs as true multithreaded Win32 service process
- controllable via the control panel "services" applet

- supports logging to the Windows NT Event log
- extended log entries

3.0 beta 1

This version has been released to the public on October, 6th 2000.

- This version is much improved. It contains the following enhancements:
- WinSyslog client added
- interactive display of syslog messages
- easy service configuration
- logging to flat ASCII files
- logging to ODBC data sources
- Licensing via licensee name and license key. There is only a single executable for both the trial and the licensed version. This way, a trial installation can become a fully licensed one with even less effort.

3.0 Final Release

This version has been released to the public on October, 16th 2000. It is a production build of 3.0 beta 1. It contains the following enhancements:

- some bug fixes (Client & Service)
- minor user interface enhancements in the WinSyslog Client
- multilingual interface for WinSyslog Client

3.1 Beta 1

This version has been released to the public on October, 31st 2000. It contains the following enhancements:

- Japanese-language support
- XML based internationalization system
- increased message logging size
- fixed some minor bugs

3.1 Final Release

This version has been released to the public on December, 4th 2000. It contains the 3.1 Beta 1 enhancements plus:

- Password encryption for ODBC connection settings
- fixed a bug that caused a maximum of 256 bytes to be written to the ODBC data source (other event targets were reported correctly)
- enhanced setup program based on the Microsoft Windows Installer Service, the new standard for software installation in the Windows environment

3.2 Final Release (Build 111)

This version has been released to the public on January, 30th 2001. It contains new enhancements and some bugfixes:

- time zone used can now be configured (Localtime or UTC)
- fixed a bug in the WinSyslog Client that occurs only on Mutlimonitorsystems.
- fixed a bug that caused the client to hang when the user had insufficient access privileges to the system registry (client now displays an error message and quits gracefully)

3.3 Preview Release (Beta 1, Build 113)

This version has been released to the public on 2001-03-14. It offers major enhancements over the previous versions.

- Flexible Rule Engine - the big, big plus! Messages received are now run through rules. Each rule is associated with actions (like sending mail or writing to ODBC databases) that are carried out when the rule matches. There is an unlimited number of rules and actions.
- EMail Support - received syslog messages can now be forwarded to email recipients.
- Syslog Forwarding Support - allows to cascade syslog servers. Messages received by WinSyslog can be forwarded by syslog protocol to syslog servers on other systems.
- Remote Administration - the client can now connect to remote systems and configure them.
- Clients supports integrated Version checking via Adiscon's online eSupport site.
- Unicode based - results in faster execution under Windows NT/2000/XP and also eases internationalization.
- Web interface to syslog database - available as a separate free download. The web interface enables viewíng syslog messages from any web browser in real time.

3.3 Beta 2 (Build 114)

This version has been released to the public on 2001-03-23. It offers fixes and enhancements over the preview release.

- Added a new Registrykey bReloadRuleBase. If this value is set to 1, the WinSyslog Service reloads the Ruleset everytime when receiving a Syslog-message. This is very useful for testing and debugging a complex rule base.
- Enhanced the Client with the Rulebase Wizard, which helps all users to build a basic Ruleset. The Wizard also can Import older settings from WinSyslog 3.2 (And lower).
- Added a new Toolbar into the Client, where all function like Save or Reload ... can be called.

- Added more support for controlling the Service. The Client can now secure Start, Stop and Restart the service. If an error occurs while these actions, a detailed error message occurs
- Added more Support for Remote Configuring. That means you can configure and maintain a WinSyslog Service on other machines. This is very useful, you don't need a physical access to the machine running the WinSyslog Service.

3.3 Beta 3 (Build 115)

This version has been released to the public on 2001-04-02. It offers important fixes and enhancements over the beta 2 release.

- Fix for immediate expiration - a bug in beta 2 made enhanced features unavailable (see related news release at

  www.winsyslog.com/Common/en/News/WinSyslog-2001-04-02.asp
- Memory leak removed - beta 2 had a memory leak if ODBC errors occurred. This has been fixed.
- • More descriptive ODBC error messages - if ODBC connections fail, more detailed information is logged to the NT application event log.
- New, enhanced installation system - based on Windows Installer service and InstallShield. Now has complete repair options as well as custom setup options.

3.3 Final (Build 117/Client 3.3.31)

This version has been released to the public on 2001-04-12. It contains all features of the previous beta versions plus small changes. It is a fully supported final release meant to be used in production environments.

- Configurable syslog forwarder port - the IP port to be used when forwarding syslog messages can now be specified both globally and on a per action basis,
- bug fix in real-time logging display - priority and facility were mixed up
- some minor (cosmetic) bugs fixed

3.31 Final (Build 118/Client 3.31.40)

- bug fixed with DBCS-Encoding (WinSyslog Service) - A message encoded with DBCS-characters caused the Service to stop working. This is now fixed. All dbcs-encoded messages are right processed.
- RuleBase editing on Remote machine (WinSyslog Client) - While managing a remote machine, the RuleBase-Menu was always disabled. Now, you can also edit the RuleBase on a remote machine. Its also possible to run the Client on Windows9x and to maintaince a remote machine running Windows NT/2000.

3.32 Final (Build 119/Client 3.32.47)

- Fixed a bug in the "Send Email" function (WinSyslog Service). - When sending an email, the date was false in some timezones. This is now corrected.
- Enhanced the "Edit Rules" Window (WinSyslog Client).

3.4 Final (Build 120/Client 3.4.52)

- Windows 9x/Me file logging support - The client itself does now support logging to a flat file. This feature allows file logging under Windows 9x and Windows Me.
- Improved client display - Facilities are now displayed color coded and with full name (e.g. LOCAL0 instead of 16).
- 3.5 Final (Build 121/Client 3.5.75)
- Spanish language user interface - the WinSyslog client now supports a Spanish language user interface.
- ODBC logging enhancements - it is now possible to overwrite the default field names. This provides additional flexibility for enhanced solutions.
- Fixed a bug that could cause the WinSyslog service to stop unexpectedly if the mail server used for email delivery did refuse connection. Now, these event is properly reported and processing continues. Bug seen very seldom in reality.
- some minor bug fixes in the client application

3.6 (Build 122/ Client 3.6.112)

This version has been released to the public on 2001-09-06. The main new feature is support for Microsoft's new Windows XP operating system. It detects the operating environment automatically and adjusts accordingly.

- Enhanced the WinSyslog Client with the new Windows XP Look and Feel.
- The WinSyslog Client now fully supports the new Windows XP Fast User Switching feature. It checks if another user in another session is using the WinSyslog Interactive Server on the same port.
- New manual available in PDF Format.
- New option available for Syslog forwarding. It is now possible the add the original source of a message when forwarding to another syslog server.
- Fixed some minor bugs in the WinSyslog Client.
- WinSyslog has a new enhanced installer now. Users can now download a smaller install set which will download the Windows Installer only if necessary (it typically is not necessary under Windows 2000, Windows XP and systems with Office 2000 or above installed). In most cases this will reduce the download time.
- some minor bug fixes in the client application

3.7 (Build 124/ Client 3.7.126)

This version has been released to the public on 2001-12-06. It contains a number of user requested small enhancements as well as some bug fixes.

- Customizable email subject line. We do now have support for replacement characters. So the event source, facility, priority and message content can be included into the email subject. Great for pagers and cellular phones, which often only display the subject line of a message sent to them.
- RFC3164 compatible date and time parsing. If enabled, the receive time stamp is taken from the syslog message rather than from the local system time.
- Unique file name generation (based on system date) can now be turned off. This was requested by customers monitoring syslog files with external file monitor processes.
- File Logging data fields are now configurable. Date/Time, Facility and Priority fields can now be turned off. If so, they won't be written to the log file.
- Solved a usability issue. When using the rule wizard with standard settings, a syslog forward to local host was often accidentally created. This in turn lead to a loop where each message received was forwarded to WinSyslog itself, starting an endless iteration. Now, even when forwarding is enabled it is disabled by the product if no syslog forwarder address is specified (we formerly used a default of 127.0.0.1).
- Fixed a bug that occurred when ODBC logging was used with Oracle.
- Improved the WinSyslog Client speed. Especially slow machines with Windows XP should see a faster WinSyslog realtime log display.

4.0 RC1 (Build 301)

This version has been released to the public on 2002-01-24. It is a considerate improvement over the 3.x releases. This version has undergone 2 cycles of beta testing and is close to the final release.

4.0 RC2 (Build 302)

This version has been released to the public on 2002-02-20. It has been enhanced based on user feedback on RC1. There are some bug fixes, as well as a number of important improvements:

- dramatically improved performance for the ODBC and file write actions
- the email forwarder now works on a timeout. If multiple events are to be forwarded by email within a short period of time (a few seconds), these can be combined into a single email message. This is configurable.
- Added the SETP forwarder
- Full RFC 3164 message parsing added

- New timestamp with the date/time originally reported added
- support for sending XML formatted message format for email and file logging.
- Timestamps can now be written in UTC format into the ODBC database
- New text log file options. Including WebTrends compatible format and a raw syslog format that is required by some other syslog file readers.
- Improved help system – now available as standard Windows HTML Help

4.0 FINAL (Build 304)

This version has been released to the public on 2002-03-09. It is basically the same code, with some minor enhancements:

- some terms have been changed for better understanding (e.g. “Criteria” are now “Filter Conditions”)
- multi-lingual user interface now available in English, French and German
- documentation updates, including added step-by-step guides

4.1 (Build 308)

This version has been released to the public on 2002-04-10. It offers these new benefits:

- includes Spanish and Japanese language support
- the Interactive Syslog Server now does support all 5 languages, including French (which was missing in the 4.0 release).
- New service: heartbeat: can be used to periodically send “I am alive” packts to other syslog servers.
- Improved configuration client: configuration changes can now automatically be saved without user intervention.
- Greatly improved performance for the “write to file” and “write to database” actions
- Greatly simplified the configuration process to create one syslog message file for each reporting system (an often requested feature).
- New “set property” action allows to overwrite some message properties. Great for renaming devices.
- Documentation updates, including added device-specific step-by-step guides

4.2 (Build 316)

This version has been released to the public on 2002-06-18. It offers these new benefits:

- German and French are now available consistently throughout the product screens.

- Much improved Japanese language support. The output encoding (EUC, JIS, SJIS) can now be selected for email and syslog forward actions.
- Incoming message can now be filtered based on a lower, higher or equal syslog priority. Previously, only an exact match was supported. This greatly simplifies rule creation in common scenarios.
- Replacement characters are now supported in the "Start Program" action. This allows e.g. to pass the source system or message content to an external program.
- Improved syslog over TCP receiver – offers greater compatibility and more options.
- Support for debugging complex rule bases (log file can be written)
- Interactive Syslog Server can now be instructed not to scroll incoming messages as they arrive – often requested to save client machine performance.
- New exclude filters based on message content (all previous versions supported include filters, only)
- Support for SMTP authentication added
- File logger does now allow concurrent reads and writes to the log file. So it can be reviewed e.g. with notepad while it is constantly being written.
- Support for Windows XP visual styles added.
- Database schema change to capture new data and provide seamless interaction with upcoming changes and new products like the new web interface or the MonitorWare console.
- Enhanced WinSyslog Web Access based on the former web interface now part of the core product.
- Some bug fixes

5.0 (Build 344)

This version has been released to the public on 2003-06-18. It offers these new benefits:

- New Scaleable and flexible Filterengine -The new filter engine as very powerful, you can build complex filter conditions like known from Microsoft Network Monitor. A note for existing WinSyslog Users. After update, you have to start the WinSysogClient first. This is important, because it will automatically import your existing filters into the new Filter system.
- New Actions
- Call RuleSet Action - this Action is used to call another RuleSet for processing.
- Set Status Action - Used to set an internal status variable. Can be used together with the Status Filter. The manual will contain more information about the Status Engine in future.

- Add Comments - You can Add Comments under Services, RuleSets, Rules and Actions now. This is useful if you want to write down some notes.
- New Import / Export functions - It is now possible to Import or Export the registry settings by using a binary format.
- Import / Export RuleSets - You can Import / Export complete RuleSets into a XML Based format (Right click a RuleSet). This can be very useful if you want to duplicate RuleSets for example. The Client uses its own file extension here (.wsx = WiNSyslog XML) which is also bound to the Client. That means double-clicking such a File will automatically invoke the Client to import the RuleSet.
- Syslog Service - Enhanced the message handling (RFC 3164) to also accept not valid RFC Syslog tags.
- Database Logging Action enhanced - You can now use the new function "Create Database" to create a MonitorWare Agent valid database.
- Client enhancements - All fields which specifies seconds are replaced with a Combobox with predefined time values. It is also possible to configure custom values.
- SETP Receiver and Sender added - Only available in the Enterprise Version. Support SSL Encryption as well.
- Instant Help added - Right click everything in the configuration Treeview, and you will see a list of helpful FAQ Article's in the HowTo-Submenu.
- Custom Message Format - Fully custom able message can be configured for FileLogging, Send Email and Net send.
- File Logging - Added support for circular logging.
- Send Email - Added support for SMTP Authentication
- Run in foreground prompt - Added support to run the service executable as an interactive application in the foreground (in(at a command prompt) . Use the “-r” switch to do this.
- Process Priority - You can now configure WinSyslog's default process priority within the debug options.
- Some bug fixes

Copyrights

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

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

Glossary of Terms

EventReporter

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

Millisecond

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

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

MonitorWare Line of Products

Adiscon's MonitorWare line of products is a suite of monitoring and operations management tools. If consist of several components, each of which can be used either individually or as a complete solution. As of this writing, the following products are available:

- ActiveLogger (www.activelogger.com)
- EventReporter (www.eventreporter.com)
- MoniLog (www.monilog.com)
- MonitorWare Agent (www.monitorware.com)
- WinSyslog (www.winsyslog.com)

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

Resource ID

The resource ID is an identifier used by the MonitorWare line of products. It is a simple, administrator assigned string value. It can be used to correlate different

events – even from different source – to a specific resource. For example, on a Windows server running Microsoft Exchange, all Exchange events could be assigned to a resource id of "Exchange Server".

In MonitorWare Agent 1.0 and WinSyslog 4.0 support for resource ids is limited. The field is present and can be persisted to the database or stored in XML files, but besides this there is no value in it. Later releases of the MonitorWare line of products will much broader support the resource id.

SETP

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

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

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

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

SMTP

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

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

SMTP is highly standardized. As such, a standard email client can work with all SMTP compliant servers. In the public Internet, almost all providers offer SMTP compliant mail servers for their customer's use.

Syslog Facility

Syslog Facility is one information field associated with a syslog message. It is defined by the syslog protocol. It is meant to provide a very rough clue from what

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

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

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

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.

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.

UpgradeInsurance

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

To learn more about UpgradeInsurance, please visit

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

UTC

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

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

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

Index