EventSentry Overview

Part I Introduction 1

Part II Quick Overview 2

Part III General 3

1 Packages 4
- Event Log Packages 4
- Health Packages 4

2 Managing Agents 6

3 Variables 7

Part IV Security 9

1 Compliance 10

2 Database 14
- Network Traffic Encryption 15
  - Microsoft SQL Server 15
    - Install Certificate Services (Windows 2000-2003) 22
    - Install Certificate Services (Windows 2008-2012) 28
  - PostgreSQL 37
  - MySQL 39

3 Agents 45

4 Collector 46

5 Sysmon 47

Part V Event Log Monitoring 48

1 Event Log Alerts 49
- Organizing Filters with Folders 49
- Excluding Events 50

2 Security Alerts 52
- Large Amount of Audit Failures (Threshold) 52
- Wrong Password (Threshold) 53
- File Changes 55

3 Recurring Events 60
- Verifying Backups & AntiVirus Updates 60
- One Event Every 10 Minutes 62

Part VI Event Log Consolidation 64

1 Archival and Purging 65

2 Off-Peak Consolidation over WAN 66

3 Conserving disk space and optimizing performance 67

Part VII System Health Monitoring 72

1 Service Monitoring 73
2 Disk Space Monitoring 74
3 Performance Monitoring 75
4 Event Log Backups 76
5 Monitor IIS Web Sites 77
iis_list_stopped_w3svc_sites.vbs 80

Part VIII Actions 83

1 Flexible Email Notifications with variables 84
2 The Process Actions 86
Sending events to a laser printer 86
eventprint.vbs 87
Emailing entries from a log file 89

1 Introduction

This documents aims to help you configure EventSentry quickly to accomplish common monitoring tasks, such as configuring database consolidation, disk space monitoring and more. This document contains many helpful tips and suggestion from our developers, customers and support team. It was created based on customer feedback and the most common questions received by the NETIKUS.NET support team.

This document is most suited towards users who are already familiar with EventSentry.

This document is an addition to the general EventSentry Help and the EventSentry Overview guide. Both documents are included in the EventSentry installer and can also be found under Start -> Programs -> EventSentry or can be downloaded from our web site at http://www.eventsentry.com/support/request in various different formats.

This document is not intended to be a substitution for the help document, please see the EventSentry help for more detailed information on all features and configuration options.

2 Quick Overview

There are a few basic concepts that are important to understand before you continue to use this guide:

1. Agents

EventSentry uses agents to monitor your servers and workstations. The EventSentry agent needs to be running on a machine in order for it to be monitored. The agents can either send data for select action types through the collector, or alternatively perform all notifications themselves directly, e.g. sending an email, writing to a database, etc..

You can easily install, uninstall and update agents using the "Remote Update" feature of EventSentry.

2. Remote Update

The Remote Update feature allows you to initially install the EventSentry agent on remote computers, push the latest configuration changes, update agents and stop/start the EventSentry agent on or more remote computers.

Remote Update works using SMB and RPC to connect to the remote computers, just like you would with applications like regedit and when accessing remote shares. As such, you will need to be able to access the registry and shares on the remote computer in order to use remote update. These services are available by default.

3. Heartbeat Monitoring

Heartbeat monitoring can monitor the uptime of remote computers using ping/icmp, can check remote TCP ports, monitor the state of the EventSentry service and retrieve system metrics via SNMP. The heartbeat feature does not require an agent on the remote computers and relies on the "EventSentry Heartbeat Agent" to be running on at least one computer in your network.

Please see the Quickstart Guide for a more thorough and graphical introduction into EventSentry.

3 General

The pages in the General chapter offer recommendations on basic EventSentry features that are not related to a particular feature such as disk space or performance monitoring.

3.1 Packages

Starting with version 2.70, EventSentry is now configured using packages. We distinguish between Event Log-, Log File, System Health- and Compliance Tracking Packages.

When creating packages, make sure that you organize them in a logical manner. The better you organize the packages, the easier the administration of EventSentry will be and the more time you will be able to save over time.

3.1.1 Event Log Packages

Generic Event Log Packages

For Event Log Packages we recommend that you organize them based on the software or Operating System type they work with. This is illustrated pretty well when you have a default install of EventSentry that ships with about one dozen of default packages.

For example, if you are monitoring both servers and workstations, then you can create one generic package for servers and one generic package for workstations. There are many events that are logged on all Windows servers and workstations (regardless of their role) that you will probably want to exclude. It is then easy to assign each package to your server group(s) and workstation group(s) respectively, assuming that you organized your computers in that way.

Event Log Packages based on Software

If you have a lot of different server (or workstation) software products installed, then it would also make sense to categorize those into packages. This makes it easier and more straight-forward to assign multiple packages to a single server or group.

For example, if your network consists of servers that have IIS, Exchange Server, Backup Software etc. installed, then simply create a package for each of those applications and assign them to the servers running those services.

You can even go a step beyond and configure an event log package for Auto Detection. This makes it possible for a filter package to automatically assign itself based on the existence of a particular service. For example, you can create a global IIS package that will activate itself when the W3SVC service exists.

3.1.2 Health Packages

System Health Packages

Health packages are a little bit more difficult to organize since they cannot easily be distinguished by the software installed, as are Filter Packages.

Monitoring < 10 Servers

If you are managing a small amount of servers, for example fewer than 5-10, then it is usually easy to organize your health packages. You could either create one package with common properties (e.g. services, disk space, performance) and assign this package to all servers. If some servers have special needs (e.g. application scheduler), then you can create an additional package for that.

Monitoring > 10 Servers

To keep the total number of health package at a reasonable number, we recommend that you create one health package that includes monitoring objects which apply to all servers. For example, you will probably want to monitor installed applications and a baseline performance on all servers, so create a package called "Generic Server Health" and assign this to all servers.

For monitoring objects only needed by some servers or groups we recommend that you divide monitoring objects into multiple packages. You can then apply those packages as needed to servers and groups. Please see the pages under the **System Health Monitoring** chapter for more suggestions on how to organize system health packages.

3.2 Managing Agents

It is important that you keep the agents running on the monitored machines up to date with the latest version. NETIKUS.NET periodically releases patches with bug fixes and minor feature additions to its customers. Please visit http://www.eventsentry.com/downloads/version-history periodically or monitor this page using URL Watch to be notified when a patch is released.

Verifying the Version installed on Agents

After you installed a new version or a patch, you can verify the build of your agent by clicking on the computer name on the top left part of the tree on the left and viewing the welcome screen. The build is shown in the Agent Information part of the welcome screen.

To check whether all machines are running the latest agent, select "Remote" -> "Check Agent Status" and click on the green arrow on the toolbar. This will show you the version of the agent running on the remote machines, for example **3.3.1.42**. The last part of this number (e.g. **42**) is the build number of the version (**3.3.1**).

To update the remote agents, select "Remote" -> "Update Agent(s)" and click the green arrow in the toolbar again.

3.3 Variables

Variables in EventSentry are an extremely useful tool to make configuring EventSentry, especially in larger installations, easier. EventSentry distinguishes between two different types of variables:

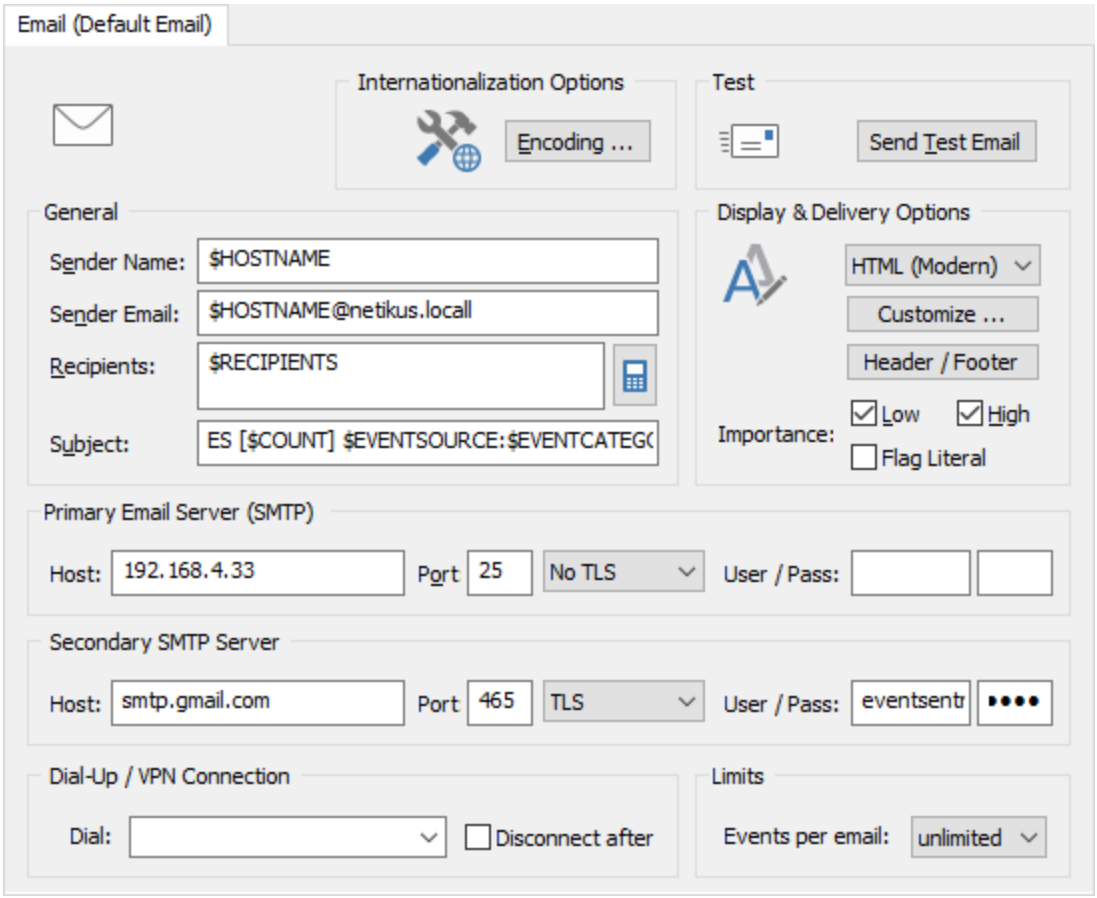
- Runtime Variables
- (Configurable) Variables

Using configurable variables for example, you can have emails sent to different recipients based on the group a computer is a member of. But many other similar applications are possible, including:

- Send emails to different recipients based on the group membership of a computer, without having to create more than one SMTP target
- Consolidate data to different databases based on the group membership of a computer, without having to create more than one ODBC target
- Use a different SMTP server based on the group membership of a computer

Runtime Variables

Runtime Variables are variables that are automatically created by EventSentry which can then be used in various configuration objects, such as notifications and filters. A good example for a runtime variable is the $HOSTNAME variable. This variable automatically resolves to the current NetBIOS host name when used. The most common application for this variable is an email notification, where the $HOSTNAME variable is used to use the computer name as the sender of an email. The screenshot below illustrates this even better:



In the above example, the sender name and sender email fields use the $HOSTNAME variable, whereas the subject field uses the $EVENTID, $EVENTSOURCE and other event record related fields. But you can also use runtime variables with other features, for example with the event log backup feature. There, you can use the $YEAR, $MONTH, $DAY, $HOUR, ... variables in the filename to make sure that you always have a unique event log backup file name.

For a full list of supported variables see http://www.netikus.net/software/eventsentry/configvariablesdetails.htm.

Regular Variables

Regular variables are different since they are defined by you and can be customized on a per-group level.

Let's take a common scenario: You are monitoring 50 servers, which are assigned to a number of different groups, and would like to configure an email notification. However, instead of assigning the same email recipient(s) to all servers, you would like to have emails sent from servers in the Database group sent to the DBA, and emails sent from servers in the Web Servers group sent to the web developer and so forth.

So, instead of setting up different notifications with different filter packages, you can make your life at lot easier by **creating a variable for the email recipients**. Here is what we would like to accomplish:

Database Servers Group:	Send email to dba@yourcorp.com and admin@yourcorp.com
Web Servers Group:	Send email to webmaster@yourcorp.com and admin@yourcorp.com
File Servers Group:	Send email to admin@yourcorp.com

1. Define a new variable called **EMAILRECIPIENTS** (or whichever name you prefer). You can define variables through **Tools->Variables** or by right-clicking the Computer Groups container. When you define a variable, you will also **set the default value** of it. This is important, since this default value will be automatically used if the value is not overwritten on a group-level. In our scenario, we'd simply use admin@yourcorp.com as the default value.

2. Right-click a group and select **Set Variables ...** to override the value of the variable. In our scenario, we would right-click the Database Servers Group, select "Set Variables ..." and double-click the EMAILRECIPIENTS value. You will notice that the default value of the variable is set to its initial value. Now, simply enter the group-value "dba@yourcorp.com,admin@yourcorp.com" and click OK. Repeat these steps for the other groups as well.

3. Now we are ready to use the variable in the actual email notification, so click your email target and replace the Recipients field with the name of the variable - **$EMAILRECIPIENTS**. Remember that variables always **start with a $** sign to indicate that what you are entering is a variable.

Variables can be used in most (but not all) fields, but check the documentation at http://www.netikus.net/software/eventsentry/configvariablesdetails.htm to see in which fields a variable is supported.

Please refer to the documentation for more information on variables.

4 Security

While it is always recommended and desirable to setup any type of software in a secure manner, ensuring that your EventSentry environment is setup securely can be particularly crucial when using EventSentry to help comply with regulatory compliance requirements such as NIST 800-171, CMMC, SOX, PCI, HIPAA and others.

4.1 Compliance

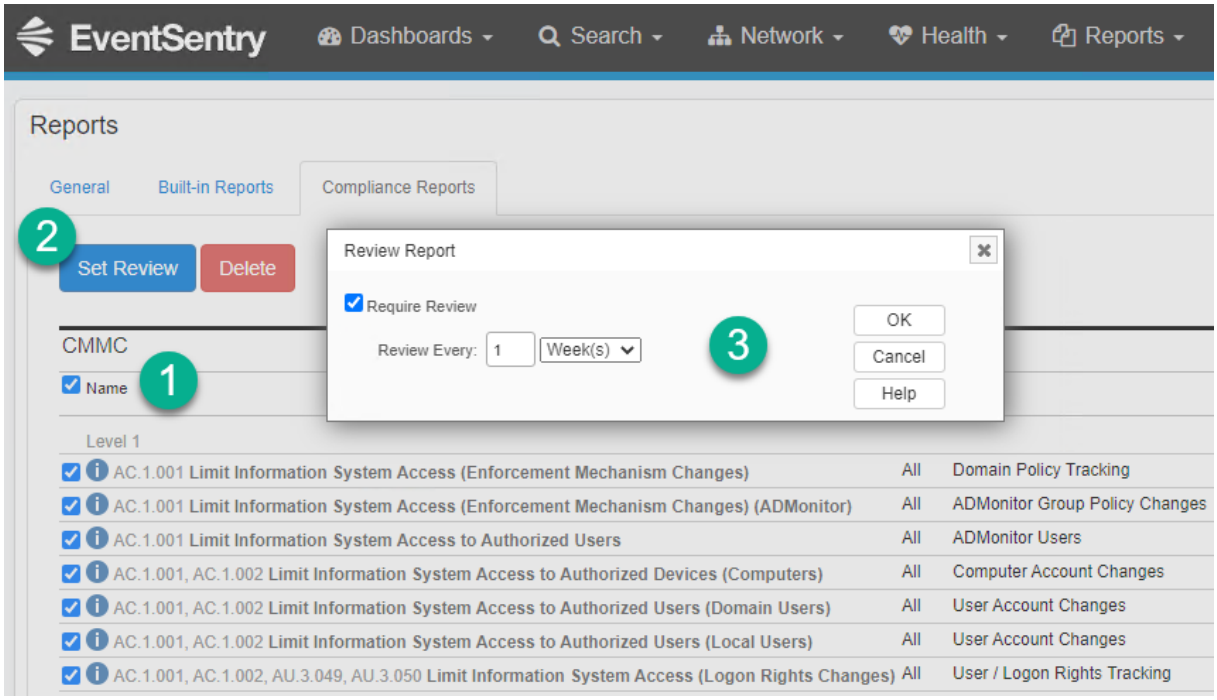
EventSentry can help users comply with a variety of compliance frameworks, including NIST, CMMC, PCI and others. This section shows all available compliance components in EventSentry along with the required steps to setup compliance.

Every EventSentry installation includes the following components that assist with you compliance:

- Compliance Reports
- Compliance Dashboard (select compliance packages only)
- Event Log Package "Compliance"
- Validation Scripts

Compliance Reports (Web Reports)

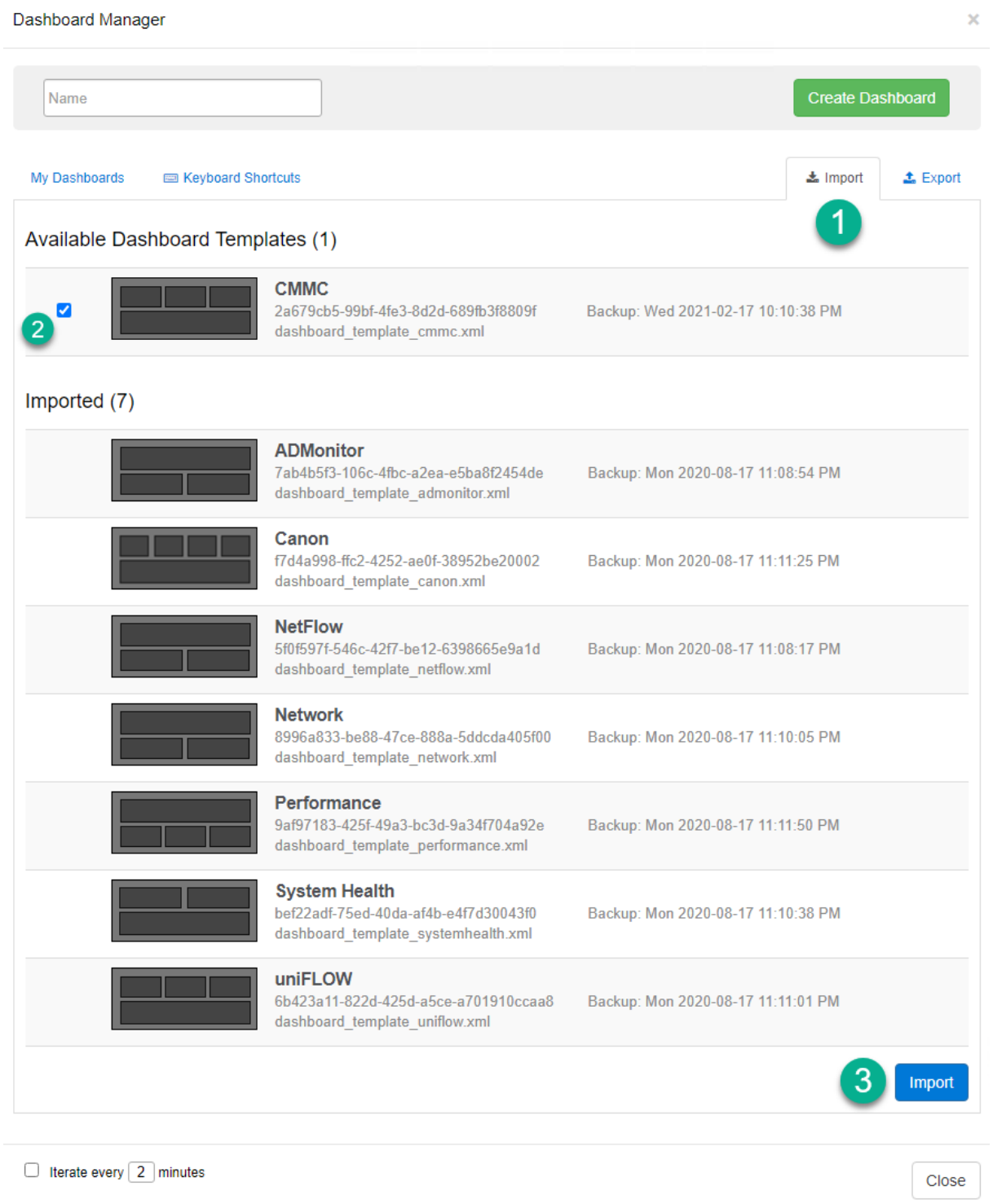
Compliance reports are built-in reports that are included with every EventSentry installation but not activated by default. Compliance reports can be enabled in the web reports by navigating to Reports → Compliance → Modify Requirements and selecting the applicable compliance requirement(s). After the reports are imported it's recommended to set the desired review period for all reports by clicking on the name check box and selecting "Set Review".



After the desired review period has been set the reports can either be run manually or scheduled with jobs. Jobs can either dispatch reports via email or store the resulting report in the file system.

Compliance Dashboard (Web Reports)

EventSentry ships with a number of dashboard templates, including templates for compliance requirements like CMMC. To import a dashboard template first load any dashboard from the "Dashboards" menu and then click on the "Change" and then the "Settings" link. In the "Dashboard Manager" click on "Import" and select the respective dashboard template. All imported dashboard templates can be customized after they are imported.



Event Log Compliance Package (Management Console)

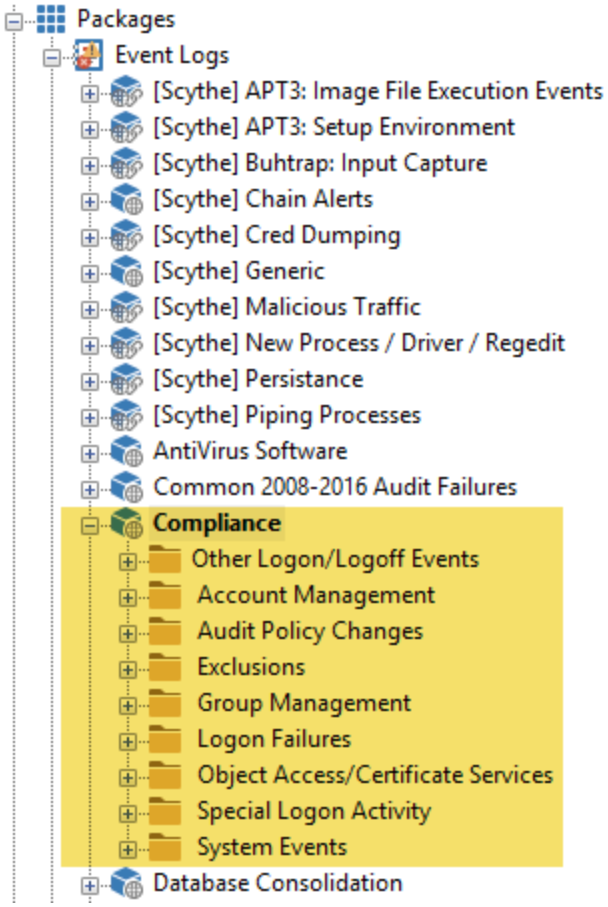
EventSentry also ships with the "Compliance" event log package that is enabled regardless of which compliance reports and/or dashboards are utilized. The package includes a variety of event log filter rules for events that are important for any environment and various compliance requirements. The compliance event log package can be found in the management console under Packages → Event Logs → Compliance.

Difference between compliance reports and compliance event log package:

Event Log Package	Reports
Supports real-time alerts for certain security events, such as group membership changes	Jobs can be scheduled at specific times or intervals
Is not tailored towards specific compliance requirement	Are created for specific compliance requirements
Control which raw events are stored in the EventSentry database	

The "Compliance" event log package is configured for the "Primary Database" action by default, which ensures that all covered events are stored in a database and available for later analysis. Some compliance reports may also rely on events collected by this package (if not covered by the "Database Consolidation") package.

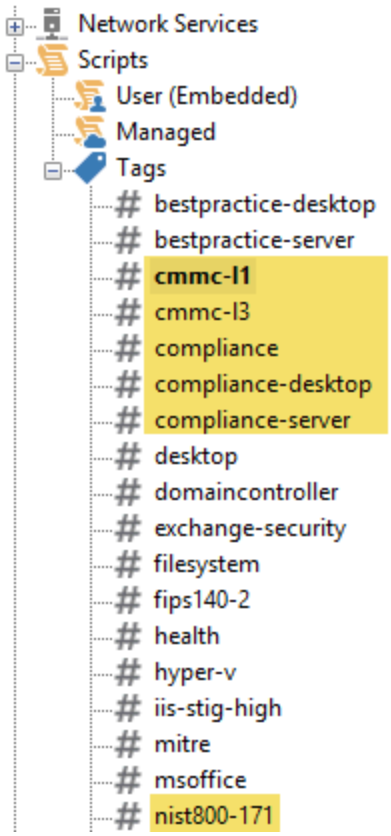
Individual filters in the package can be configured to send alerts to actions in addition to the Primary database. To accomplish this, the "Override actions in this package" option needs to be unchecked. Click the package → click "Properties" in the ribbon → Uncheck "Override actions of all objects in this package" → edit actions of the respective filter(s).



Validation Scripts (Management Console)

Compliance frameworks often dictate certain security settings, such as disabling SMBv1, which encryption algorithms to use and more. EventSentry's validation scripts continuously compare critical

settings on your monitored hosts with baselines set by the respective compliance framework. Non-compliant settings are quickly identified in the web reports on a dashboard or the validation scripts section. Validation scripts can be reviewed under "Scripts" in the management console.



To setup validation scripts for compliance, follow the steps below in the management console:

1. Navigate to Packages → Validation Scripts
2. Click on "Add" in the ribbon to add a new package and give it a descriptive name (e.g. CMMC)
3. Select the newly created package and select "Add" under "Validation Scripts" in the ribbon
4. Select the newly created "Scripts" object
5. In the top section "Assigned Tags & Scripts" add all relevant tags, e.g. "compliance-server"
6. Click the "Add" button on the bottom in the "Database" section to configure in which database validation script output is stored

4.2 Database

When consolidating events into a central database, then you will need to make sure that nobody can gain unauthorized access to your database. If somebody can get administrative access to your SQL database, then the intruder has the ability to compromise your data integrity by deleting or modify data.

Make sure you use a **strong password** for the database administrator (e.g. sa or root) and only give this password to authorized users.

All of the security steps listed below will have no effect if the administrator's login is compromised.

EventSentry Agents

The EventSentry agents are designed to only use the **eventsentry_svc** login to access the database, primarily to add data to the database. This login is created when you install EventSentry with the setup or when you initialize the database through the action dialog in the management console.

The **eventsentry_svc** user is only allowed **minimum access** to the objects (tables, columns) in the EventSentry database, for example this user cannot retrieve stored event log records from the database. As such, even if the password were to be compromised, the intruder would still not be able to retrieve useful information from the EventSentry database.

Never use an administrative login (e.g. sa or root) when configuring the database action in EventSentry.

The password for the eventsentry_svc user is stored in the registry, but only members of the **local Administrators group** have permission to access the EventSentry configuration in the registry.

When utilizing the collector service, check the "Enhanced Security" check box on the database action dialog which prevents the DB login credentials from being transmitted to the remote agents.

EventSentry Web Reports

The EventSentry Web Reports use the **eventsentry_web** user to access the database, which has a different set of permissions in the EventSentry database than the eventsentry_svc user. The **eventsentry_web** login is created when you install EventSentry with the setup or when you initialize a new database through the action dialog in the management console.

The **eventsentry_web** user is only allowed **minimum access** to the objects (tables, columns) in the EventSentry database, for example this user cannot add or delete event log records from the database. As such, even if the password were to be compromised, the intruder would still not be able to modify or delete records from the EventSentry database, though it could be used to retrieve data.

The password of the **eventsentry_web** user is stored in the configuration file of the web reports, the **WebReports\conf\configuration.xml** file which by default is located in the installation folder of EventSentry (e.g. C:\Program Files\EventSentry).

In order to keep the password of the **eventsentry_web** user secure, make sure that only authorized users have direct access to the **WebReports\conf\configuration.xml** file on the web server.

Encryption

If the EventSentry agents are transmitting event log data over an insecure medium, then it is recommended to either utilize the collector service or encrypt SQL communication between the client (any EventSentry agent) and the database server. See Network Traffic Encryption for more information.

4.2.1 Network Traffic Encryption

Encryption with Collector
Network traffic between the agents and the database can be encrypted by utilizing the Collector service, introduced in EventSentry v3.2. The collector supports secure TLS encryption between the agents and the collector, ensuring that data collected and transmitted by the agent cannot be intercepted by a third party.

Traffic between the collector and the respective database is not encrypted, as such it's recommended to install the collector on the database server or the same subnet as the database.

Encryption with the Database
The following chapters contain instructions on how to encrypt SQL traffic between the EventSentry agents and the database server hosting the EventSentry database when the collector service is not being used.

- Microsoft SQL Server
- PostgreSQL (Built-In)
- MySQL

4.2.1.1 Microsoft SQL Server

Encrypting traffic between the database and the agents is generally not necessary when utilizing the collector service, introduced in EventSentry v3.2.

Most ODBC drivers, including Microsoft SQL Server®, transmit network traffic in clear text which can be a problem in security sensitive environments. Microsoft SQL Server® supports protocol encryption which encrypts all traffic between the client (=EventSentry agent) and the Microsoft SQL Server®.

Using protocol encryption requires the following prerequisites:

- Certificate Services installed on machine running SQL Server
- Latest SQL Server ODBC drivers installed on all clients (MDAC)

This chapter will guide you through the process of setting up Certificate Services and requesting a certificate so that SQL server can use protocol encryption. This chapter is based on using Windows Server 2003 for the OS and Microsoft SQL Server® 2000/Microsoft SQL Server® 2005 for the database.

1. Install Certificate Services
If certificate services are not already installed on your domain, follow the appropriate instructions below:

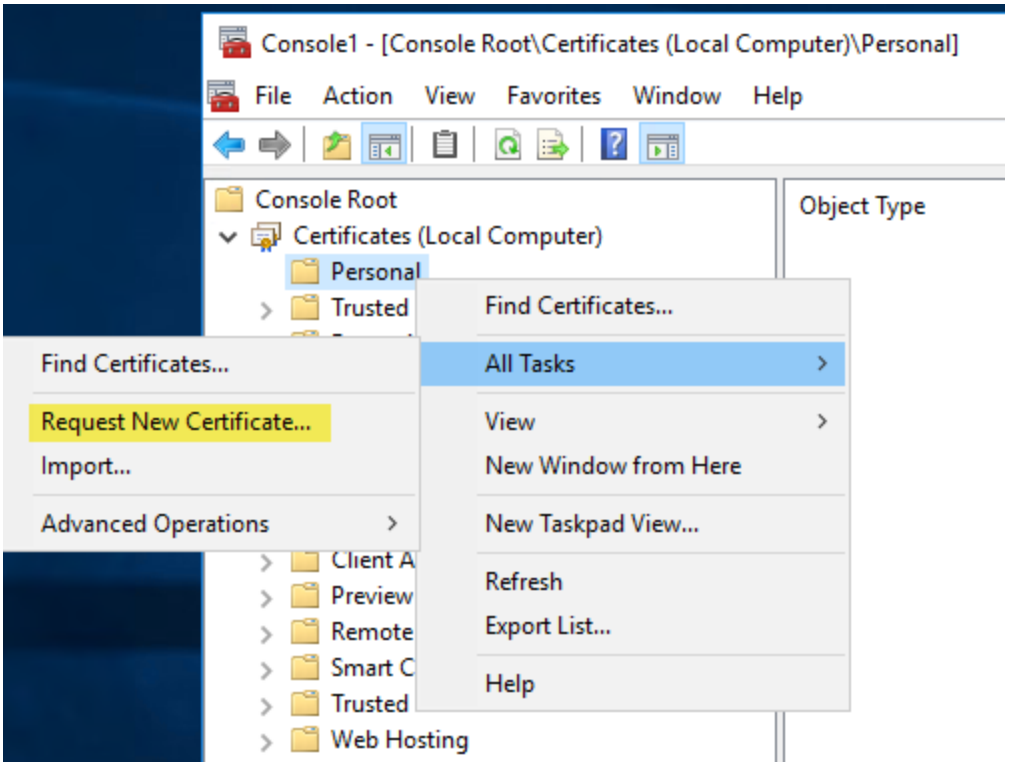
Windows Server 2003
Windows Server 2008 and later

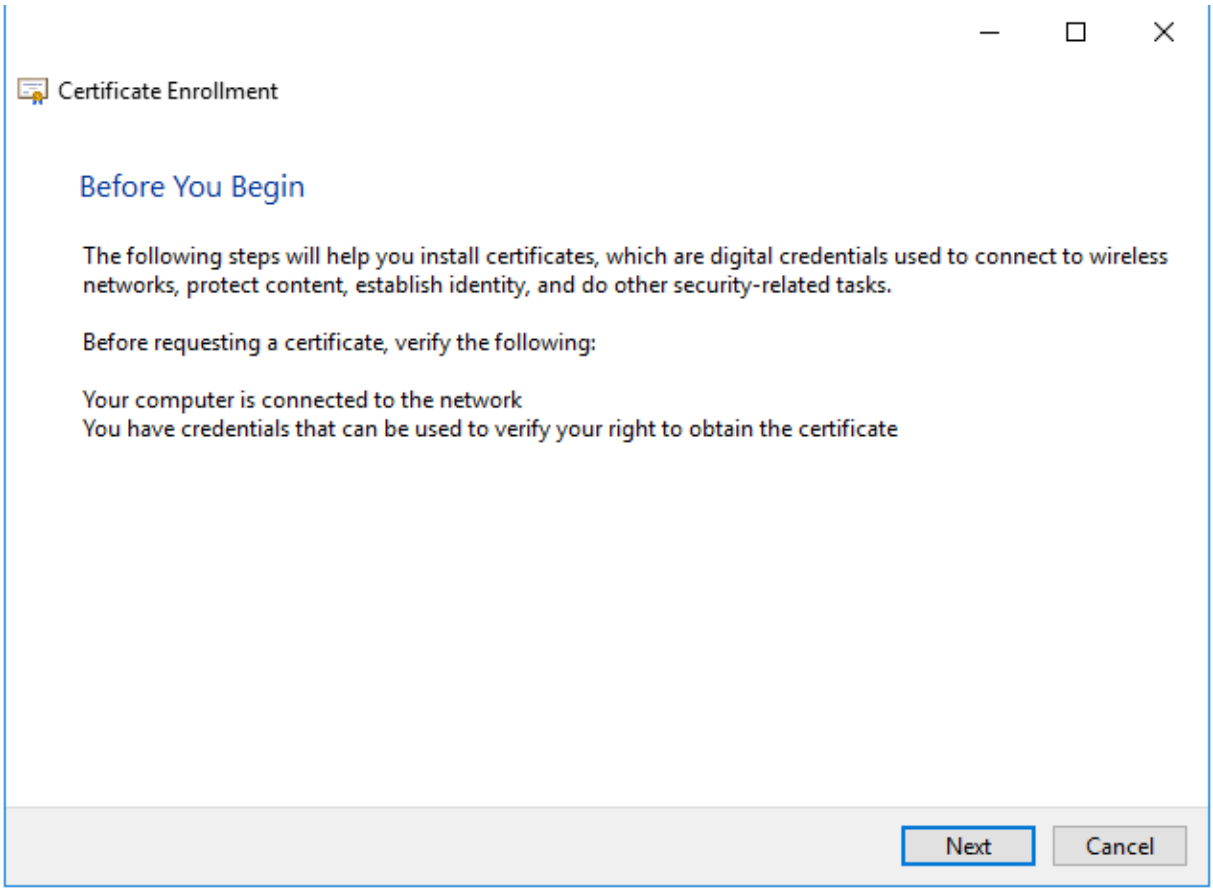
2. Configuring the MMC snap-in
In order to manage/create certificates you need to configure an MMC for the certificate services. To open the Certificates snap-in, follow these steps:

- To open the MMC console, click Start, and then click Run. In the Run dialog box type: **mmc**
- On the Console menu, click Add/Remove Snap-in....
- Click Add, and then click Certificates. Click Add again.
- You are prompted to open the snap-in for the current user account, the service account, or for the computer account. Select the **Computer Account**.
- Select **Local computer**, and then click Finish.
- Click Close in the Add Standalone Snap-in dialog box.
- Click OK in the Add/Remove Snap-in dialog box. Your installed certificates are located in the **Certificates** folder in the **Personal** container.

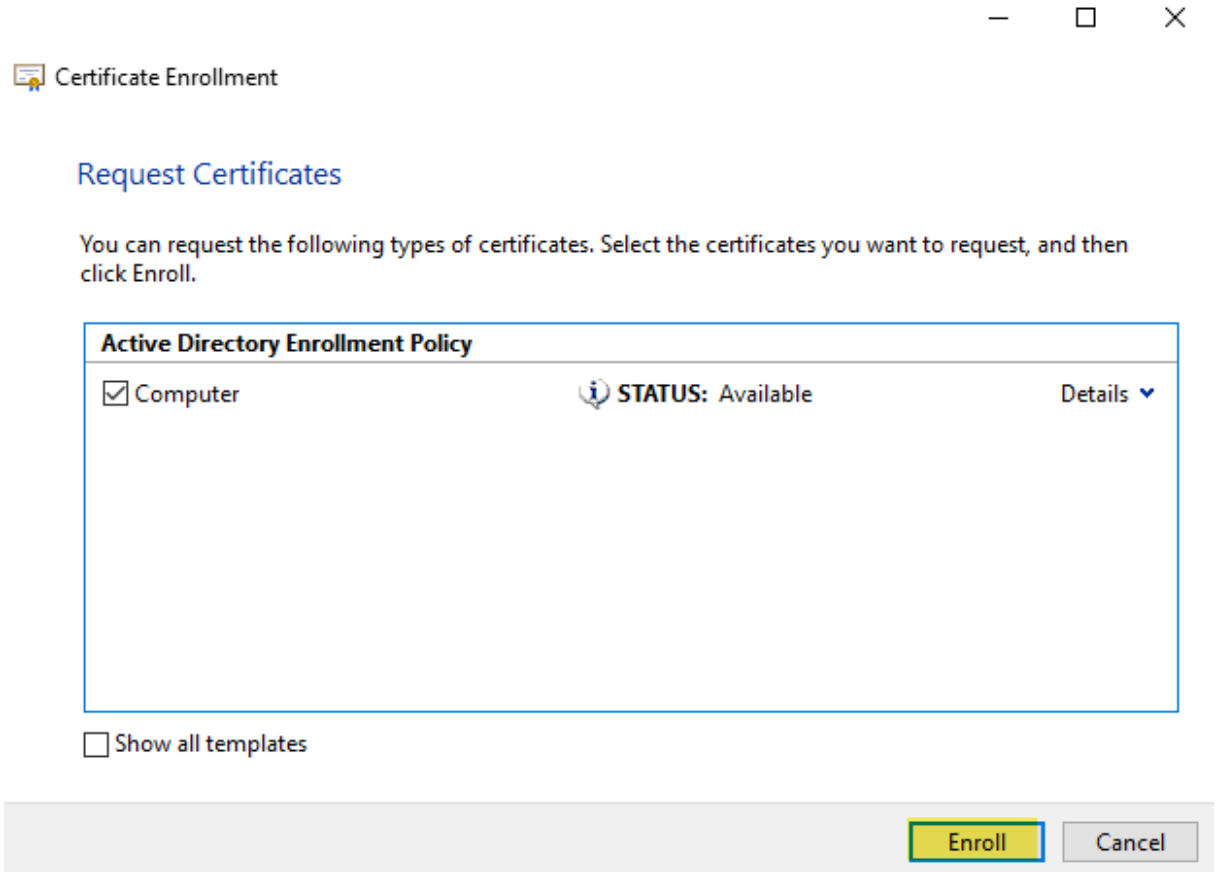
3. Installing a certificate on the server

In the MMC, click to select the Personal folder in the left-hand pane. Right-click in the right-hand pane, point to All Tasks, and then click Request New Certificate....which will bring up the dialogs shown below:





The **Certificate Request Wizard** dialog box opens. Click Next. Select Computer as the Certificate type.



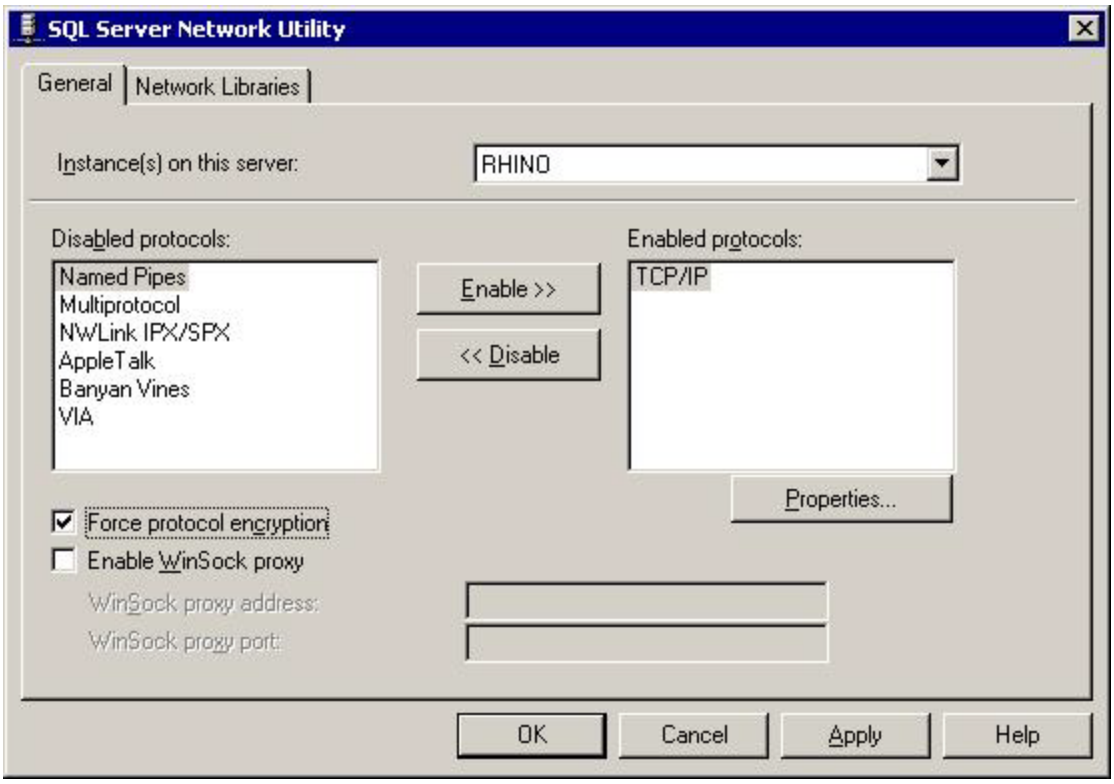
In the **Friendly Name** text box you can type a friendly name for the certificate or leave the text box blank, and then complete the wizard. After the wizard finishes, you will see the certificate in the folder with the fully qualified computer domain name.

4. Requiring database encryption for all communication

Once the certificate is installed you can configure the SQL Server to "Force protocol encryption".

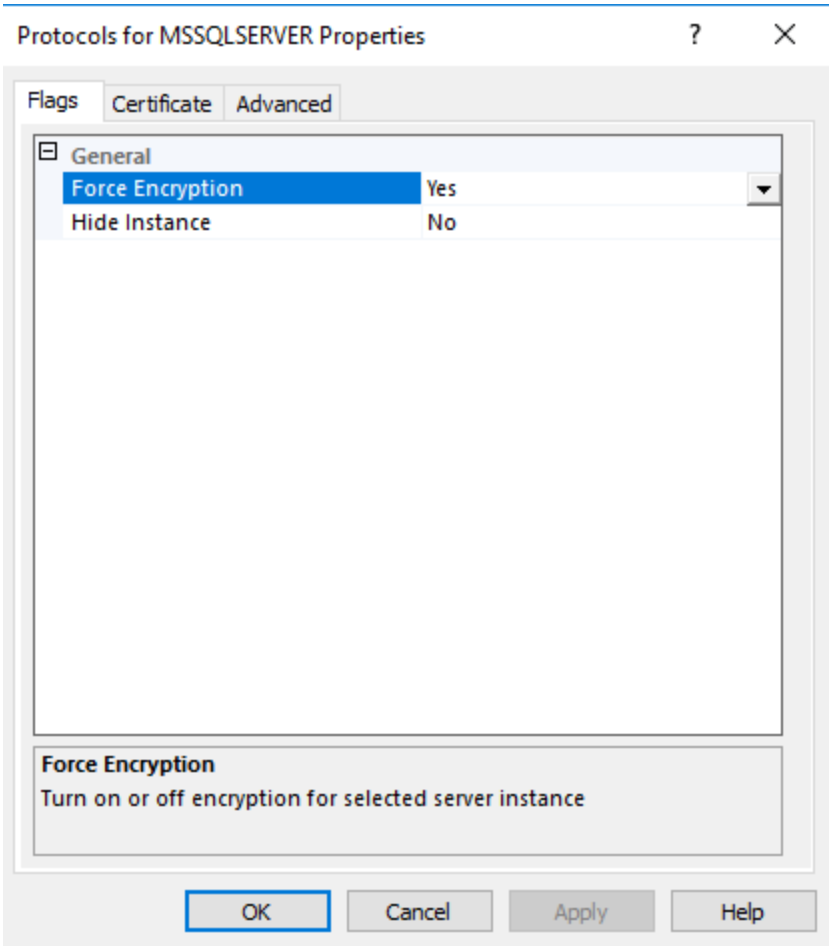
- **For SQL Server 2000**

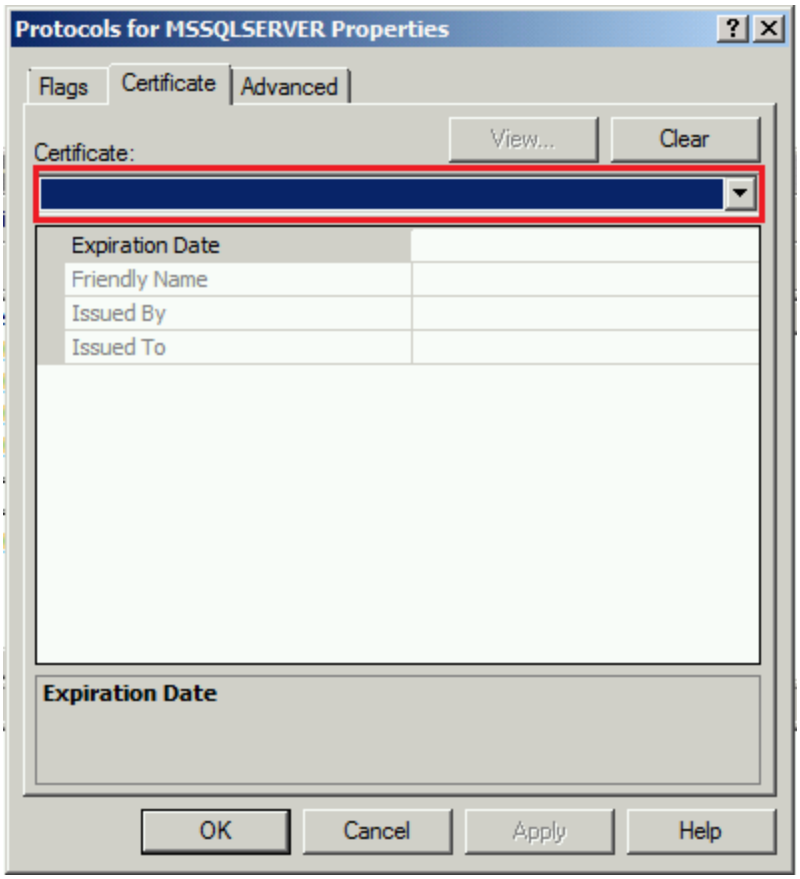
Navigate to Start -> Programs -> Microsoft SQL Server and open the "Server Network Utility". Activate the "Force protocol encryption" checkbox and click OK. From now on clients will encrypt all traffic when they communicate with the database server.



- **For SQL Server 2005 and later**

Navigate to "Start -> Programs -> Microsoft SQL Server 2005-> Configuration Tools" and open the "SQL Server Configuration Manager". Expand "SQL Server 2005 Network Configuration". Right click on "Protocols for MSSQLSERVER" and choose Properties. Set "Force Encryption" to "Yes" then click on the Certificate tab where you have to select the certificate you created above.





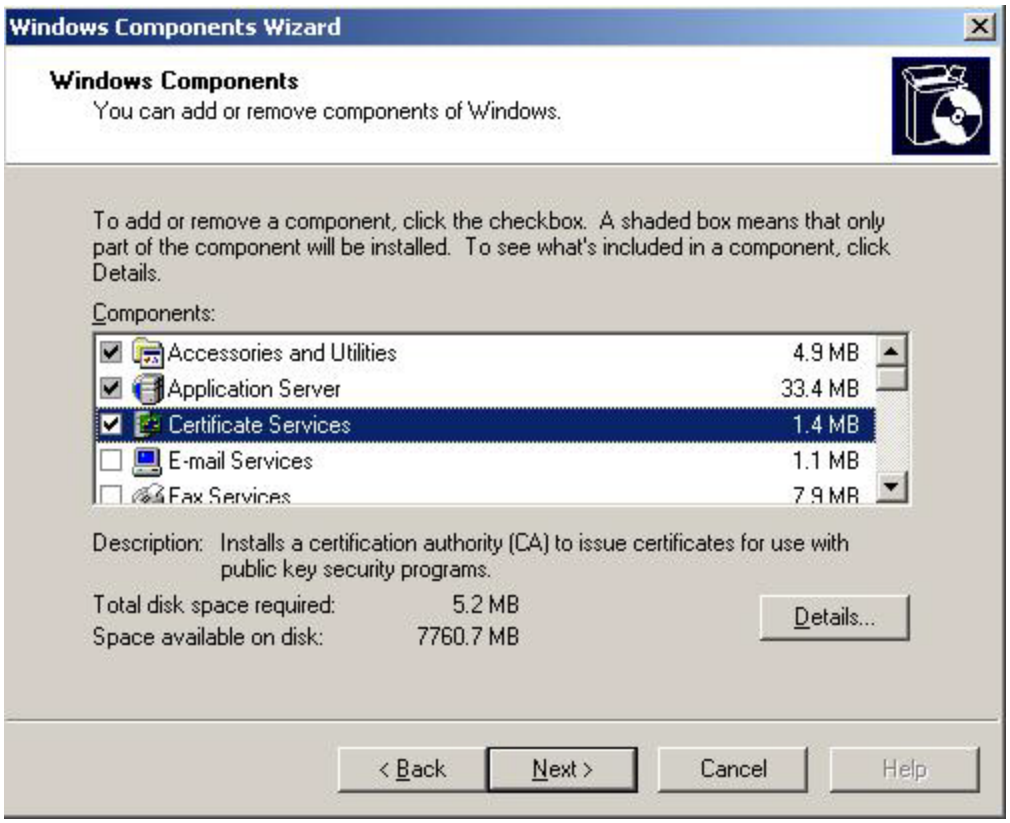
Windows Server 2003 and earlier: All clients communicating with the SQL Server will need an up-to-date SQL Server ODBC driver installed in order to support encryption. If a machine is unable to communicate with the database server after you enabled encryption, installing the latest MDAC (Microsoft Data Access Components) from MDAC Downloads will usually resolve the problem.

4.2.1.1.1 Install Certificate Services (Windows 2000-2003)

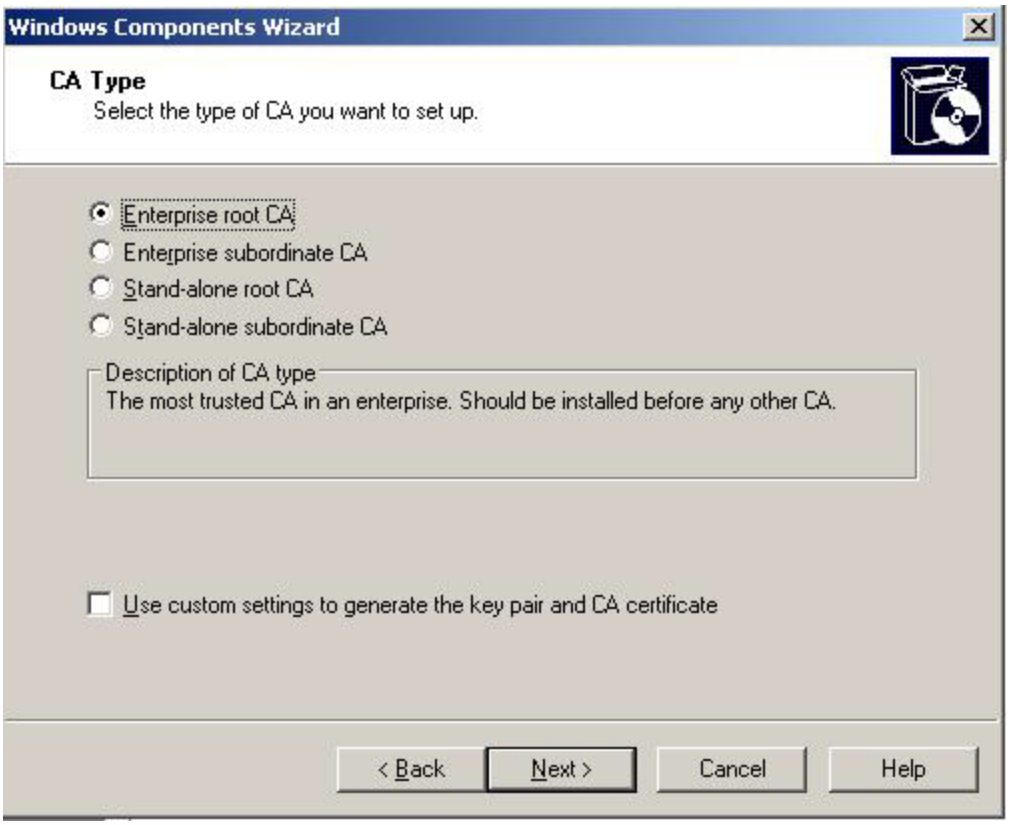
1. Installing Certificate Services

You will only need to follow these steps if you do not have certificate services running in your domain. If you already have a certificate server in your domain then you can skip step 1.

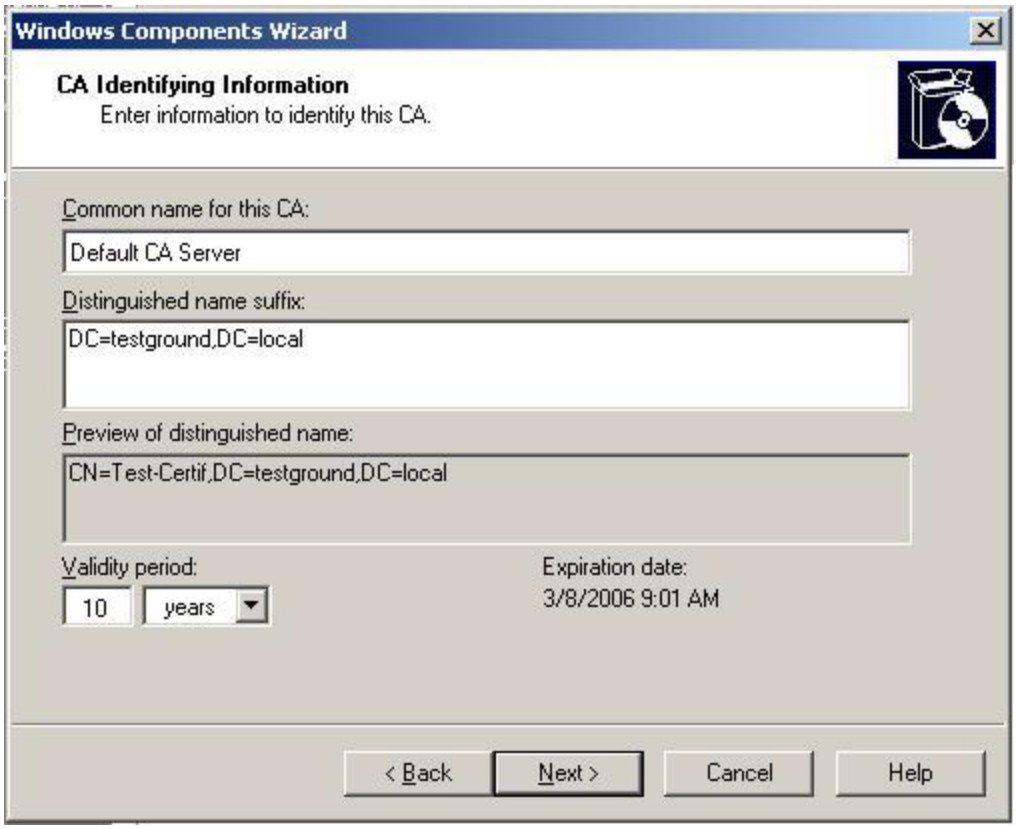
Navigate to "Start -> Settings -> Control Panel -> Add/Remove Programs" and click "Add/Remove Windows Components" which will bring up a screen similar to the one shown below:



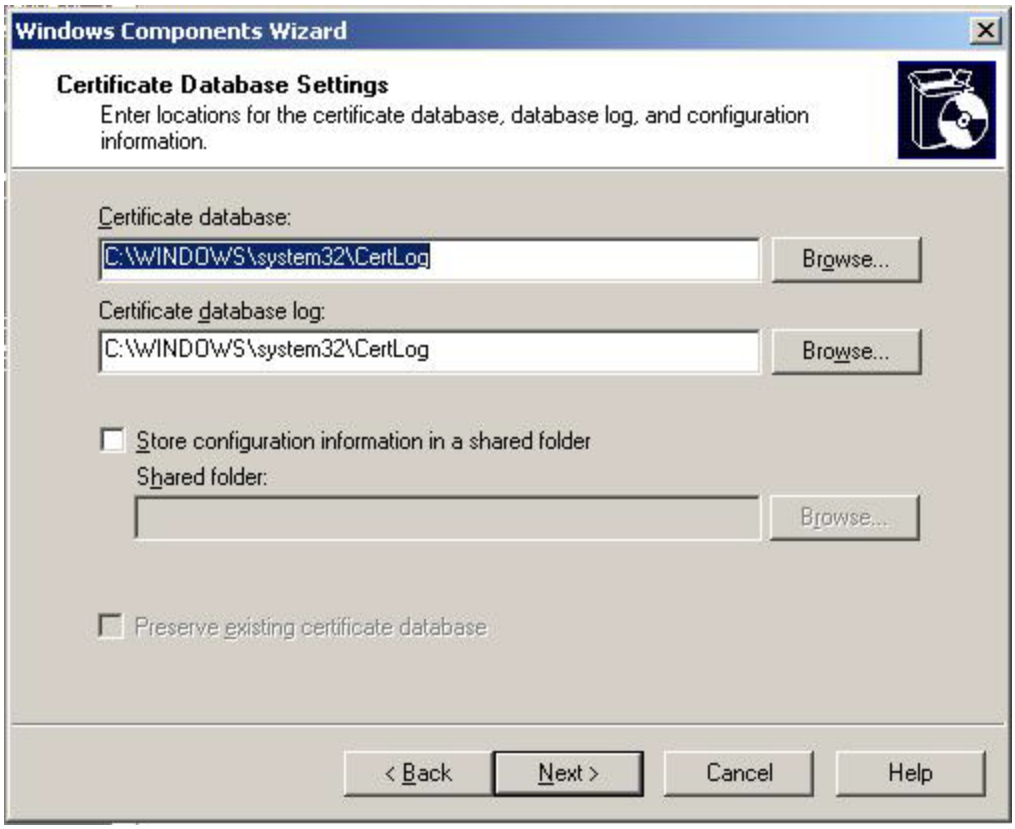
Check "Certificate Services" and click next. Click "Yes" on the confirmation dialog if the imposed restrictions are OK. On the next screen select the appropriate certificate authority type for your network. Please refer to the Windows Server documentation for more information. In our example we will be installing an "Enterprise Root CA" since it is the first CA server:

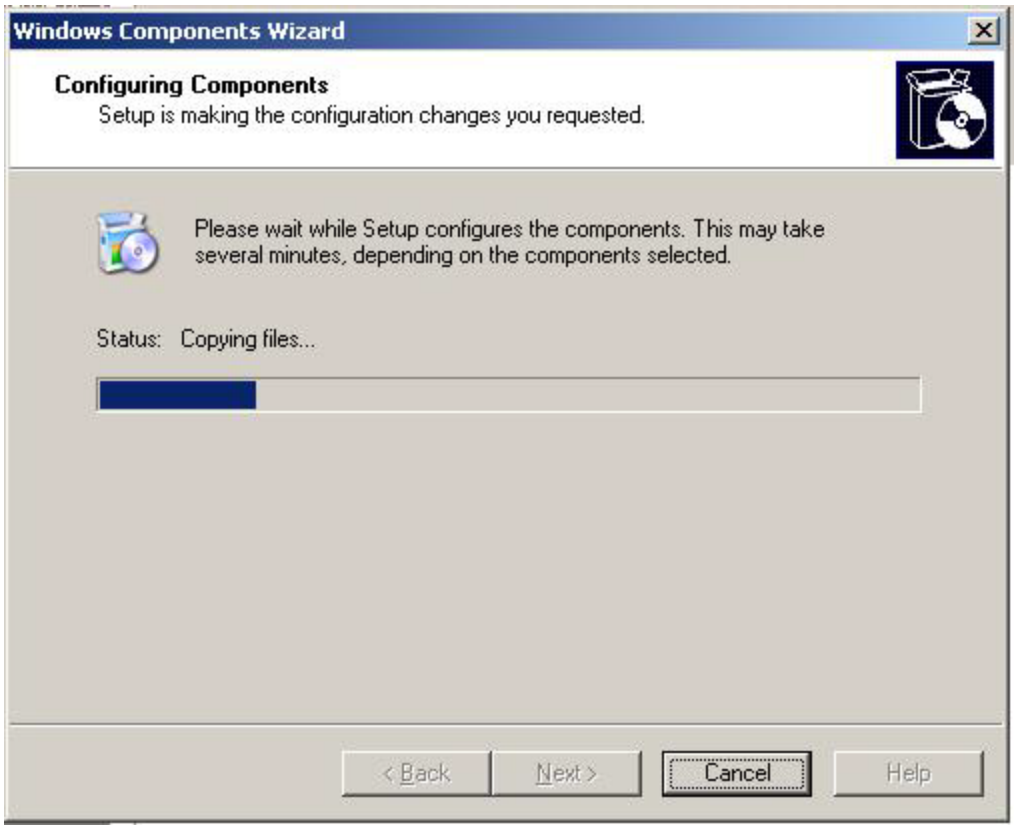


Please note that the following screenshots might look differently depending on the type of CA you select here (the screenshots shown are based on the "Enterprise root CA" selection). On the next screen enter the "CA Identifying Information". Make sure that you enter a good common CA name and a specify a validity period that is long enough:



After clicking Next confirm the following dialogs and click "Finish" to complete the setup of the CA.

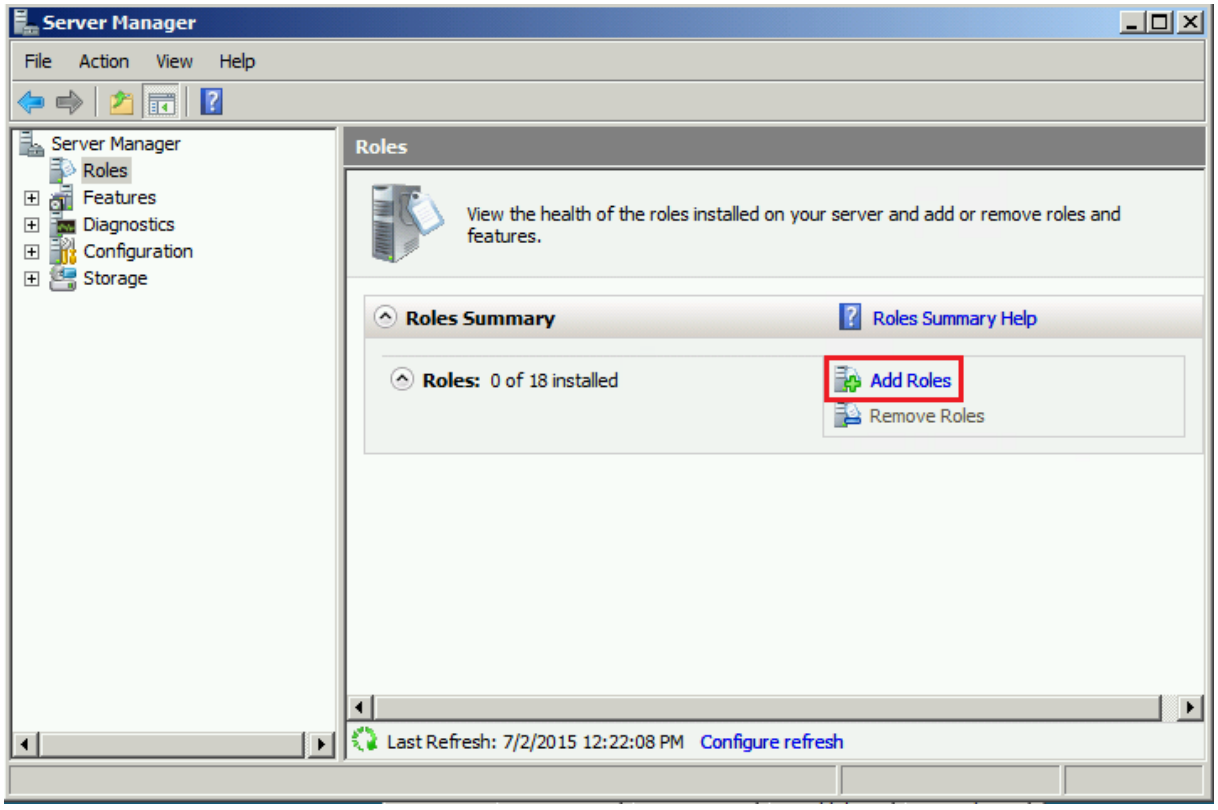




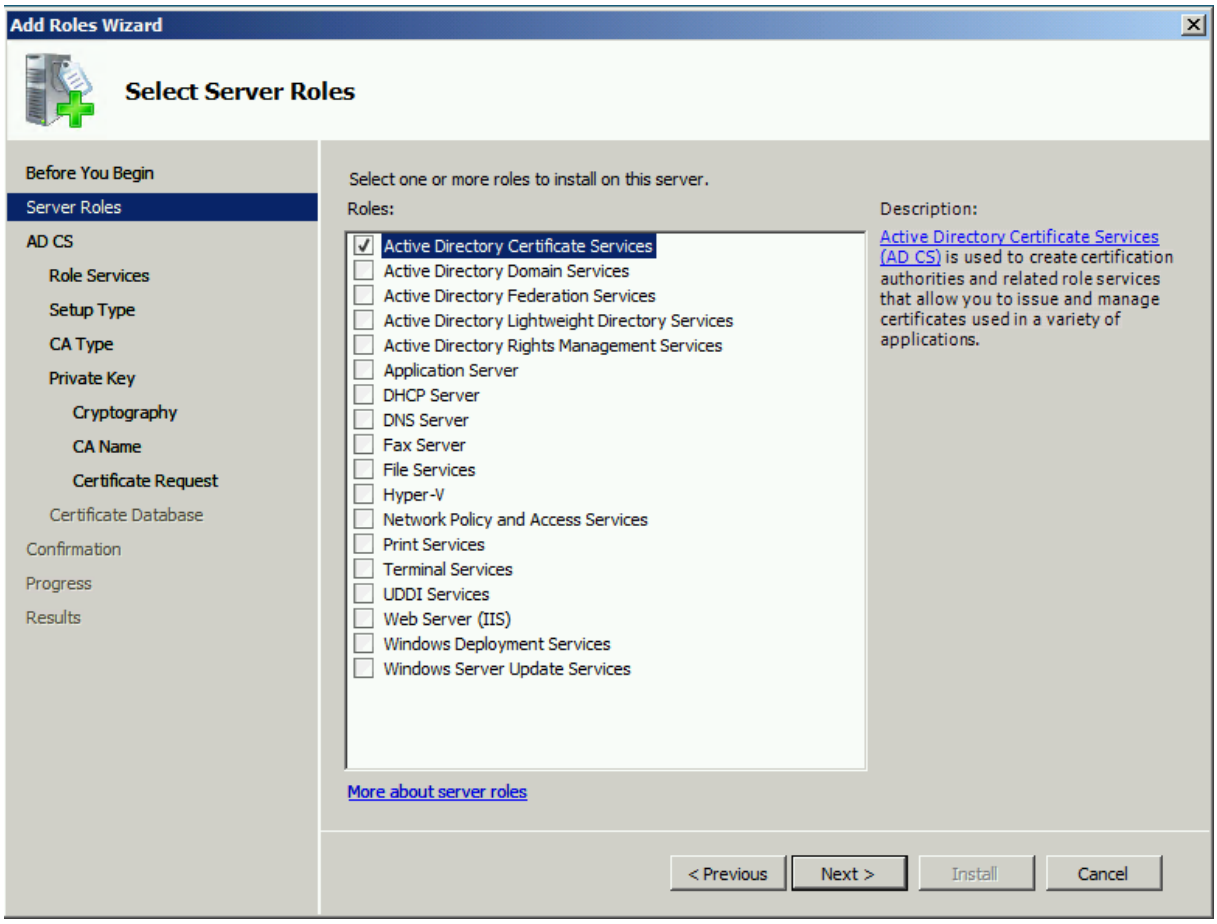
4.2.1.1.2 Install Certificate Services (Windows 2008-2012)

You will only need to follow these steps if you do not have certificate services running in your domain. If you already have a certificate server in your domain then you can skip this step.

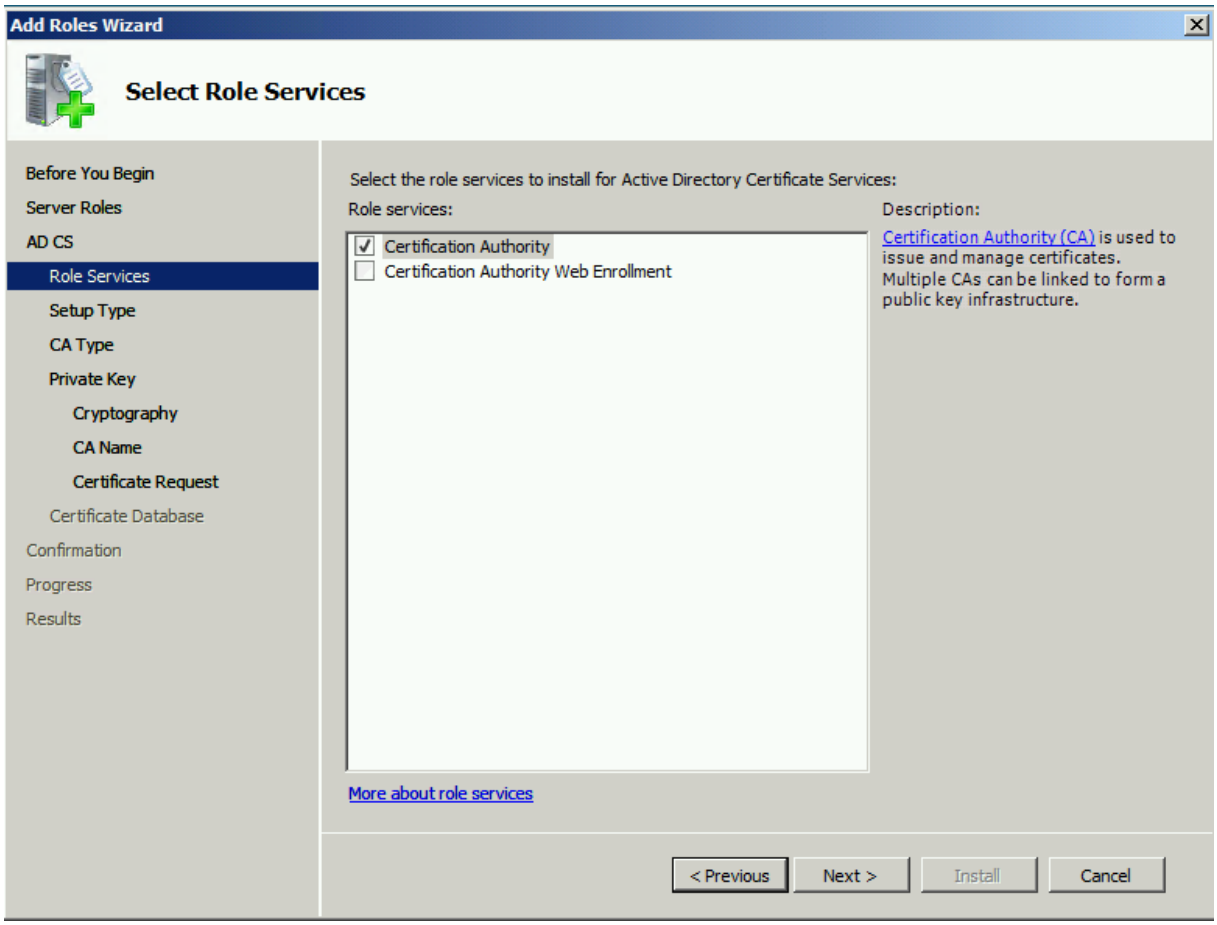
Open "Server Manager":



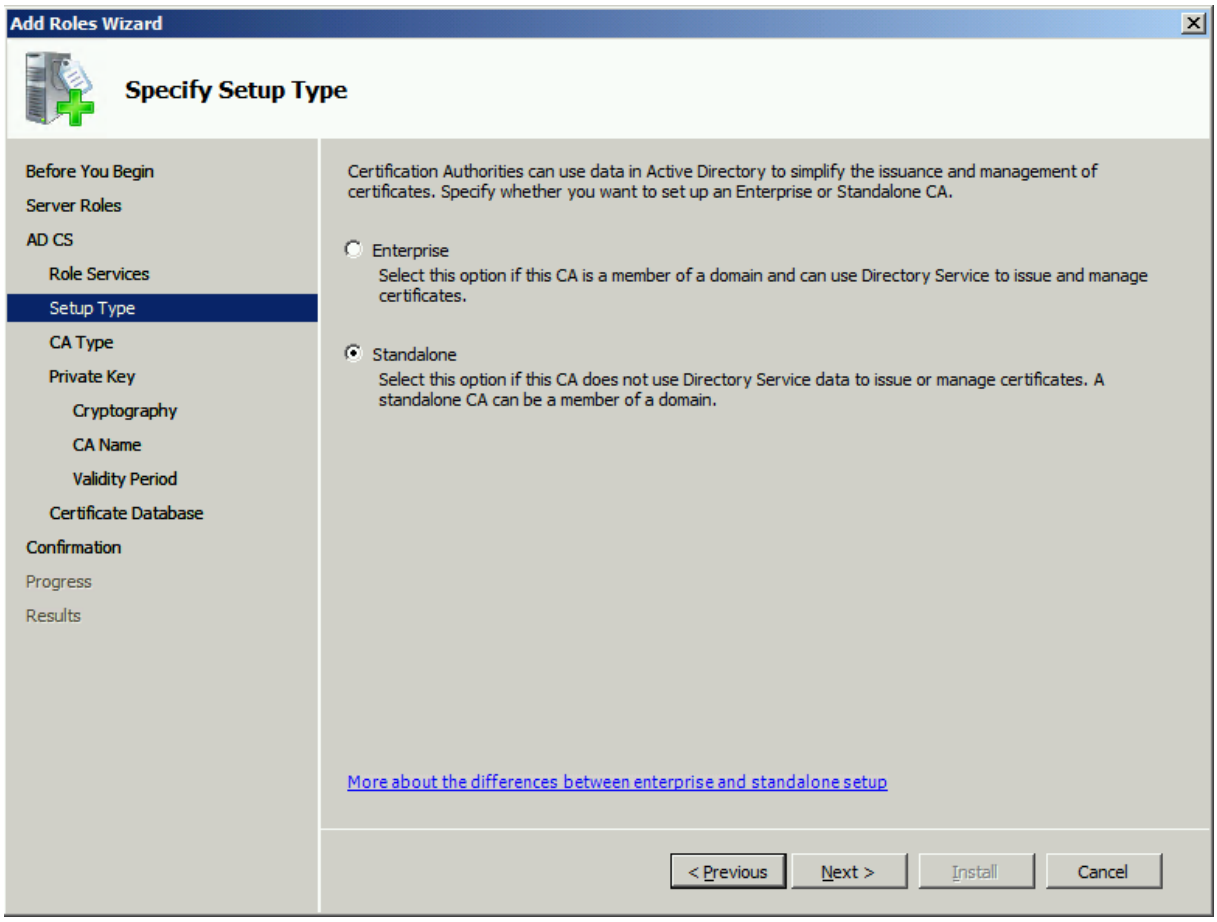
In "Server Manager" select Roles in the left pane, then Add Roles in the right pane.



Place a check mark in the check box for Active Directory Certificate Services. Click Next. On the 'Introduction to Active Directory Certificate Services' window, you can read up on the certificate services technology, how to manage a CA, and naming. Click Next.

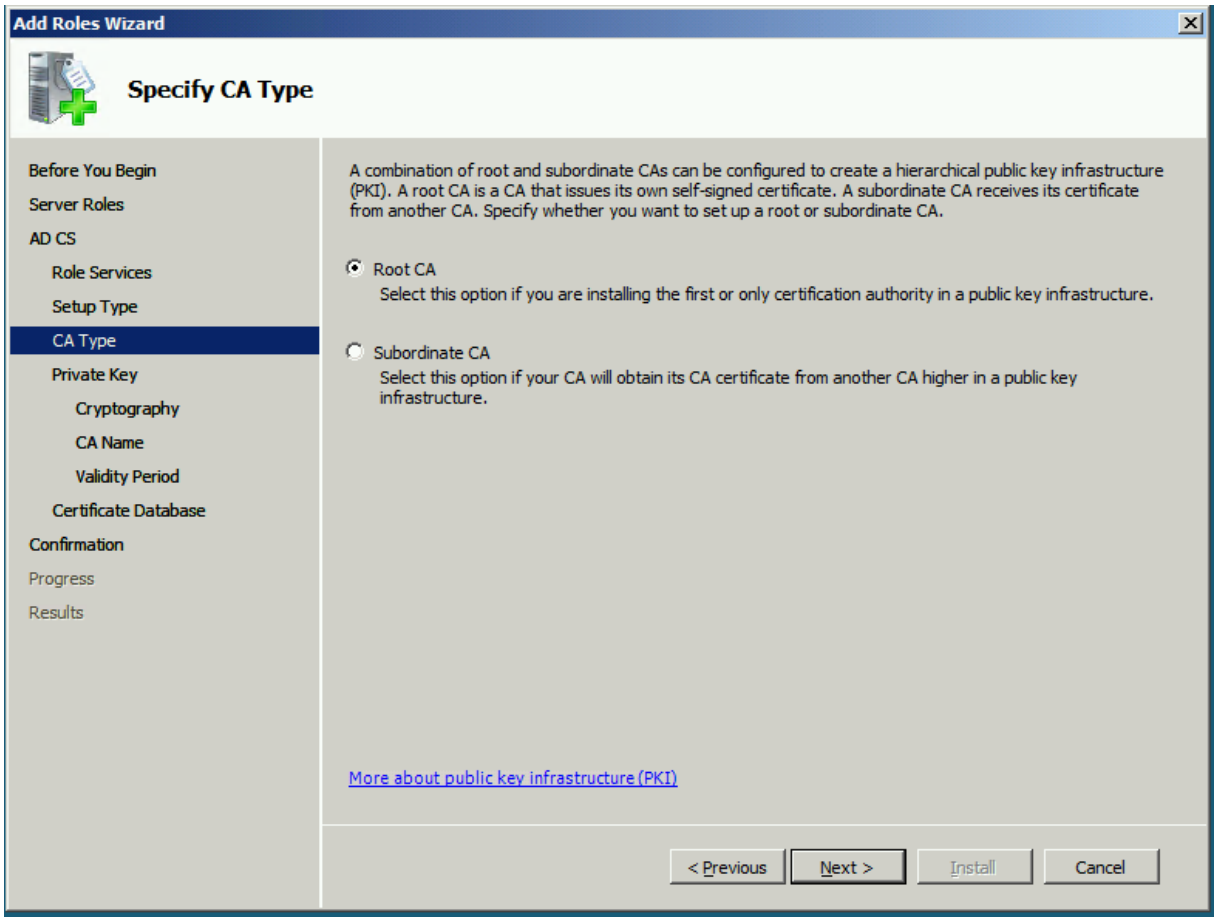


On the ‘Select Role Services’ page, make sure Certification Authority is selected. Click Next.

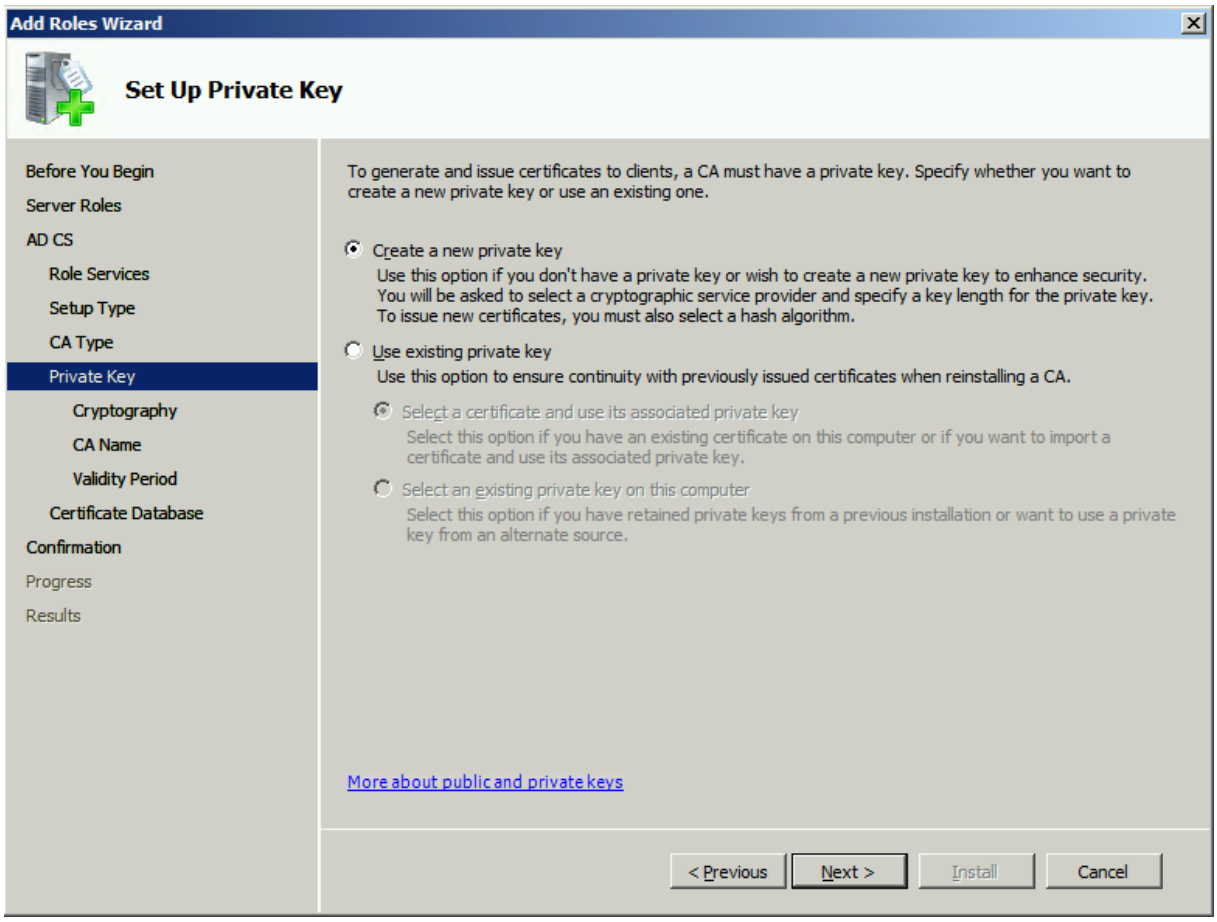


On the ‘Specify Setup Type’ page, select Standalone. Click Next.

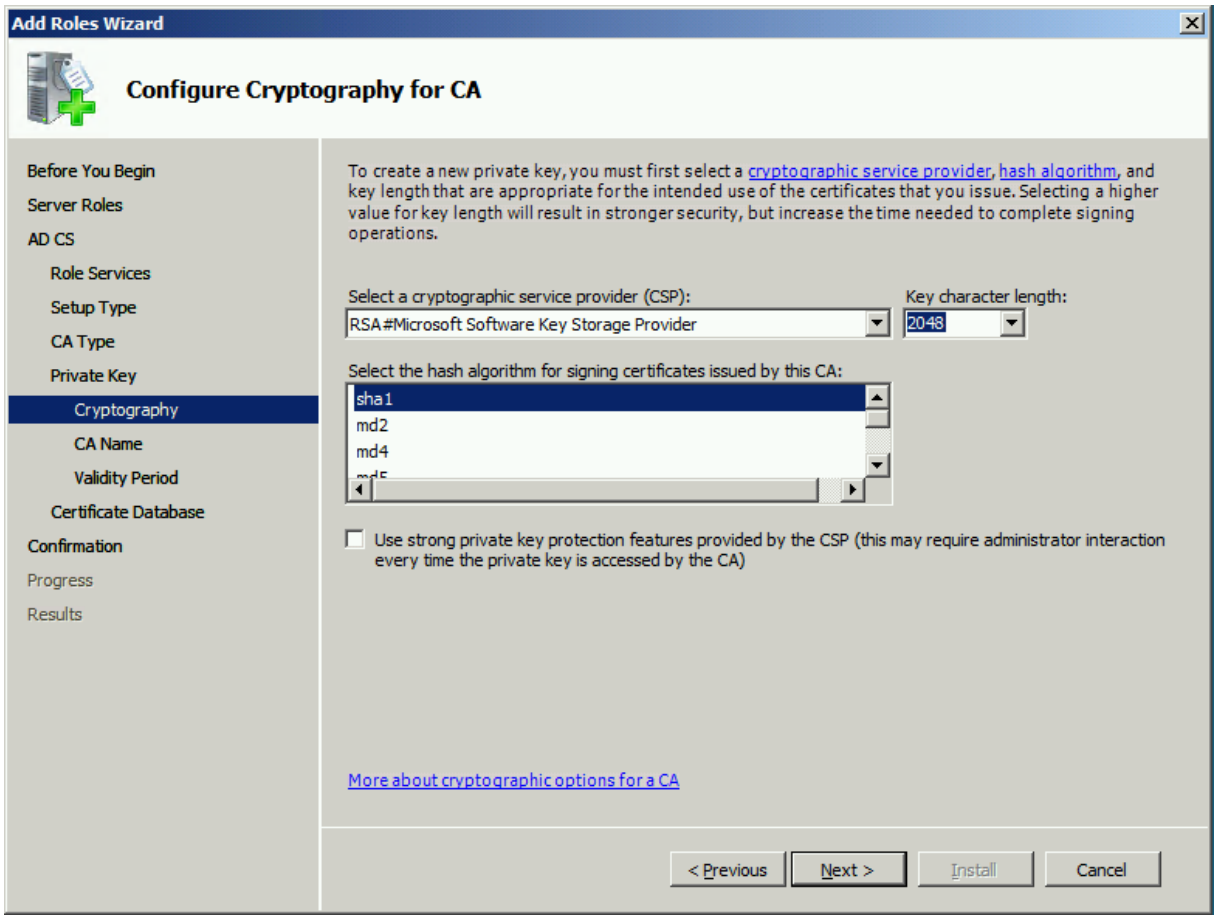
The screen shots shown are based on the "Enterprise root CA" selection, there may be differences depending on the type of CA you select.



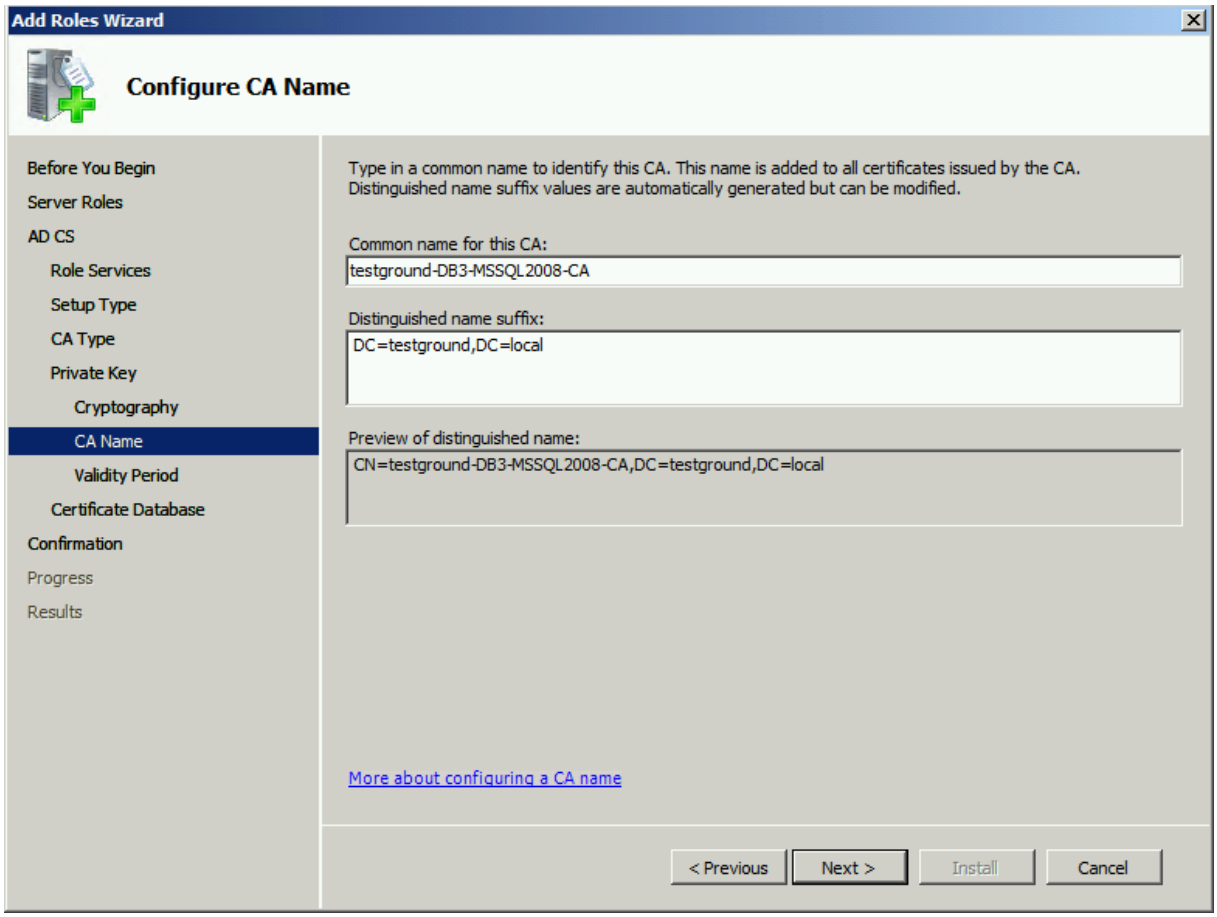
On the ‘Specify CA Type’ page, leave Root CA selected and click Next.



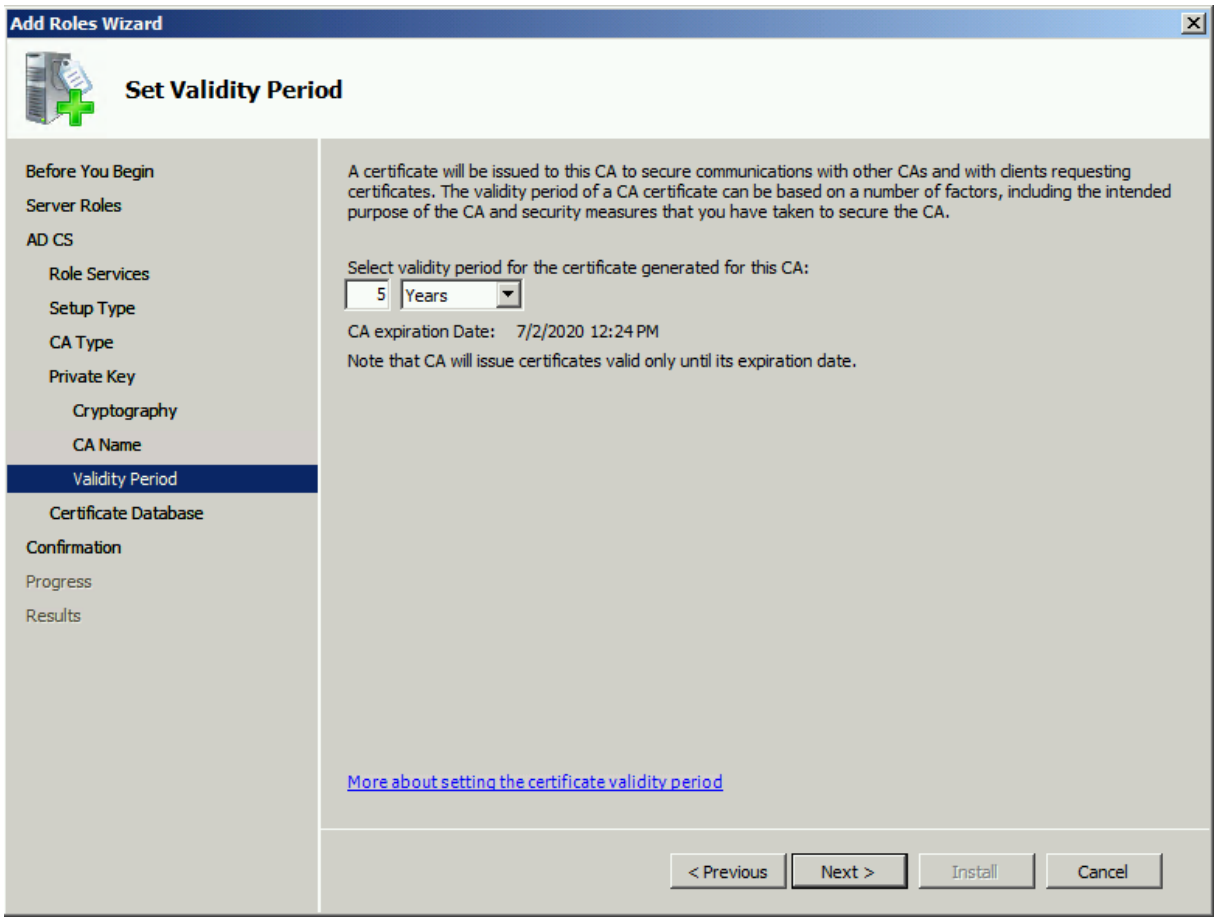
On the ‘Set Up Private Key’ page, leave Create a new private key selected and click Next.



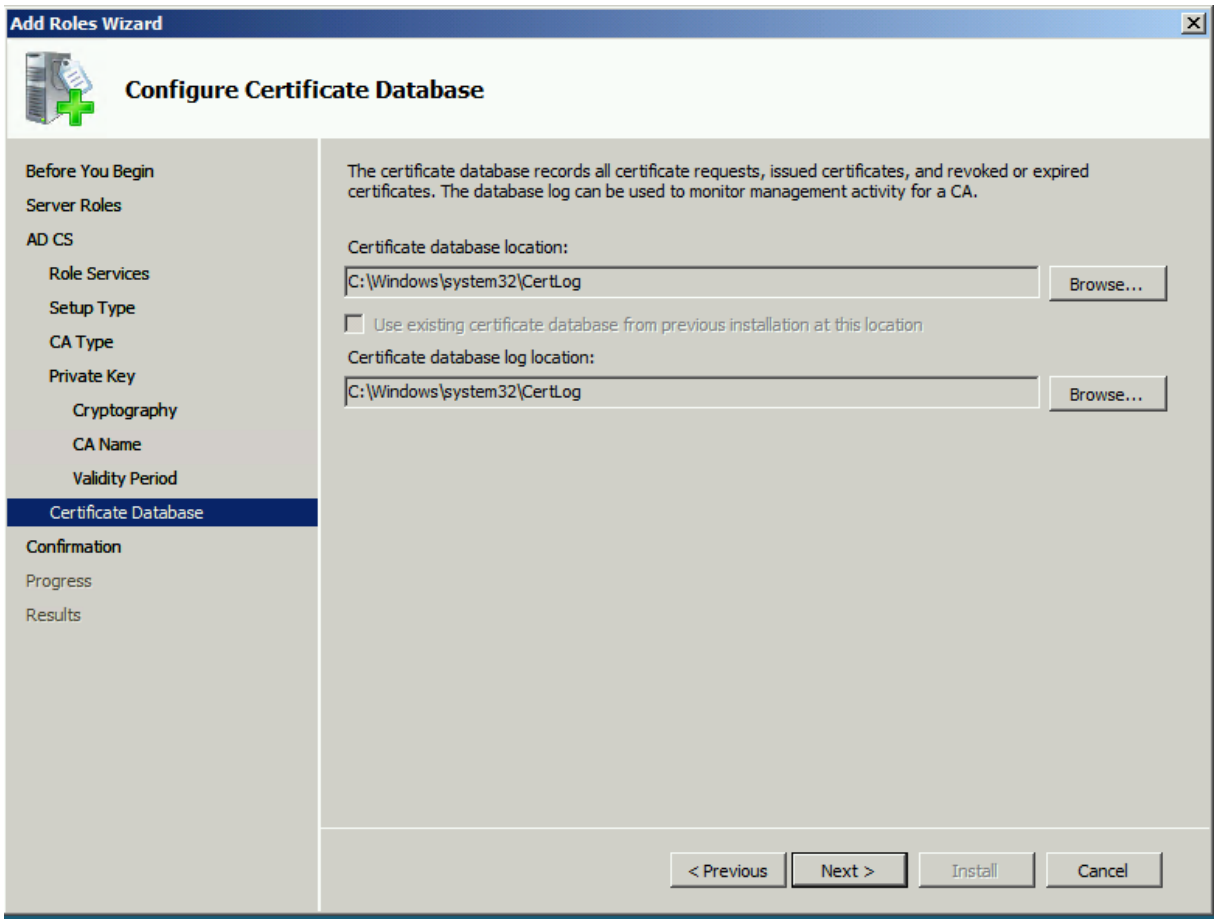
On the Configure Cryptography for CA page, leave the defaults selected or adjust as necessary and click Next.



On the ‘Configure CA Name’ page, set the common name to the same as the server name. Click Next.



On the ‘Set Validity Period’ page, set to 5 years or adjust based on your needs. Click Next.



On the 'Configure Certificate Database' page, leave the defaults set or adjust to your needs. Click Next.

On the 'Confirm Installation Selections' page, you can review your choices or click Back to make changes. Once satisfied click Next.

After the 'Installation Progress' page finishes, you can view your 'Results'.

4.2.1.2 PostgreSQL

Encrypting traffic between the database and the agents is generally not necessary when utilizing the collector service, introduced in EventSentry v3.2.

First, install OpenSSL from http://slproweb.com/products/Win32OpenSSL.html in order to create the required certificates.

Open the command prompt as an administrator (Run as administrator) and navigate to the OpenSSL directory (c:\OpenSSL-Win32 by default):

1. Set the environment variable for OPENSSL_CONF:

```
set OPENSSL_CONF=c:\OpenSSL-Win32\bin\openssl.cfg
```

2. Generate a CA certificate:

```
openssl genrsa 2048 > ca-key.pem
openssl req -sha1 -new -x509 -nodes -days 3650 -key ca-key.pem -out ca-
cert.pem
```

3. Generate a server certificate:

```
openssl req -sha1 -newkey rsa:2048 -days 3650 -nodes -keyout server-
key.pem -out server-req.pem
openssl x509 -sha1 -req -in server-req.pem -days 3650 -CA ca-cert.pem -
CAkey ca-key.pem -set_serial 01 -out server-cert.pem
openssl rsa -in server-key.pem -out server-key.pem
```

4. Generate a client certificate:

```
openssl req -sha1 -newkey rsa:2048 -days 3650 -nodes -keyout client-
key.pem -out client-req.pem
openssl x509 -sha1 -req -in client-req.pem -days 3650 -CA ca-cert.pem -
CAkey ca-key.pem -set_serial 01 -out client-cert.pem
openssl rsa -in server-key.pem -out server-key.pem
```

5. Convert and move the generated files to the postgreSQL data directory:

```
openssl x509 -outform pem -in ca-cert.pem -out root.crt
openssl rsa -in server-key.pem -out server.key
openssl x509 -outform pem -in server-cert.pem -out server.crt
copy root.crt "C:\Program Files (x86)\EventSentry\data96\root.crt"
copy server.key "C:\Program Files (x86)\EventSentry\data96\server.key"
copy server.crt "C:\Program Files (x86)\EventSentry\data96\server.crt"
```

The built-in PostgreSQL database on EventSentry v3.2 and earlier is just named "data", e.g. C:\Program Files (x86)\EventSentry\data.

6. Edit Postgresql.conf

   Location: C:\Program Files (x86)\EventSentry\data96\postgresql.conf

7. Set "ssl = on" and uncomment the line.

   Location: Line 80

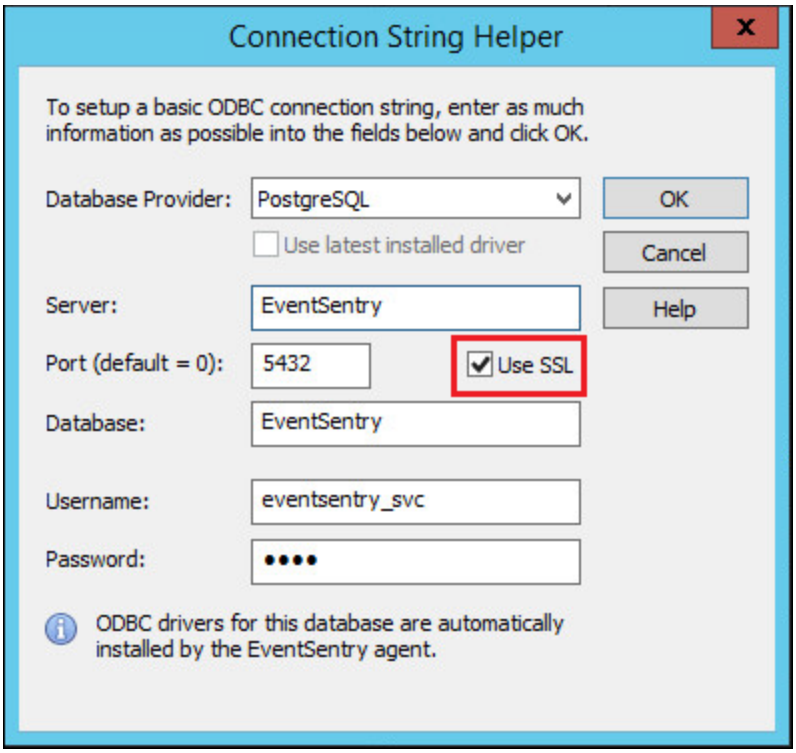
8. Save postgresql.conf

9. Restart EventSentry Database service

10. Open EventSentry Management Console, expand "Actions" and click "Primary Database"

11. Click "Create ..." next to your connection string.

12. Select "Use SSL" and verify the rest of the data before clicking "Ok"



13. Verify your connection string contains "sslmode=require"

14. Click "Test" and verify the test entry was successfully written to the database.

15. Save the configuration in EventSentry Management Console

4.2.1.3 MySQL

Encrypting traffic between the database and the agents is generally not necessary when utilizing the collector service, introduced in EventSentry v3.2.

First, install OpenSSL from http://slproweb.com/products/Win32OpenSSL.html in order to create the required certificates.

Open the command prompt as an administrator (Run as administrator) and navigate to the OpenSSL directory (c:\OpenSSL-Win32 by default):

1. Set the environment variable for OPENSSL_CONF:

```
set OPENSSL_CONF=c:\OpenSSL-Win32\bin\openssl.cfg
```

2. Generate a CA certificate:

```
openssl genrsa 2048 > ca-key.pem
openssl req -sha1 -new -x509 -nodes -days 3650 -key ca-key.pem -out ca-
cert.pem
```

3. Generate a server certificate:

```
openssl req -sha1 -newkey rsa:2048 -days 3650 -nodes -keyout server-
key.pem -out server-req.pem
openssl x509 -sha1 -req -in server-req.pem -days 3650 -CA ca-cert.pem -
CAkey ca-key.pem -set_serial 01 -out server-cert.pem
openssl rsa -in server-key.pem -out server-key.pem
```

4. Generate a client certificate:

```
openssl req -sha1 -newkey rsa:2048 -days 3650 -nodes -keyout client-
key.pem -out client-req.pem
openssl x509 -sha1 -req -in client-req.pem -days 3650 -CA ca-cert.pem -
CAkey ca-key.pem -set_serial 01 -out client-cert.pem
openssl rsa -in server-key.pem -out server-key.pem
```

5. Create a subfolder in the main MYSQL directory named "MySQL-SSL" and transfer the files there.

6. Update **my.ini** and add the following lines

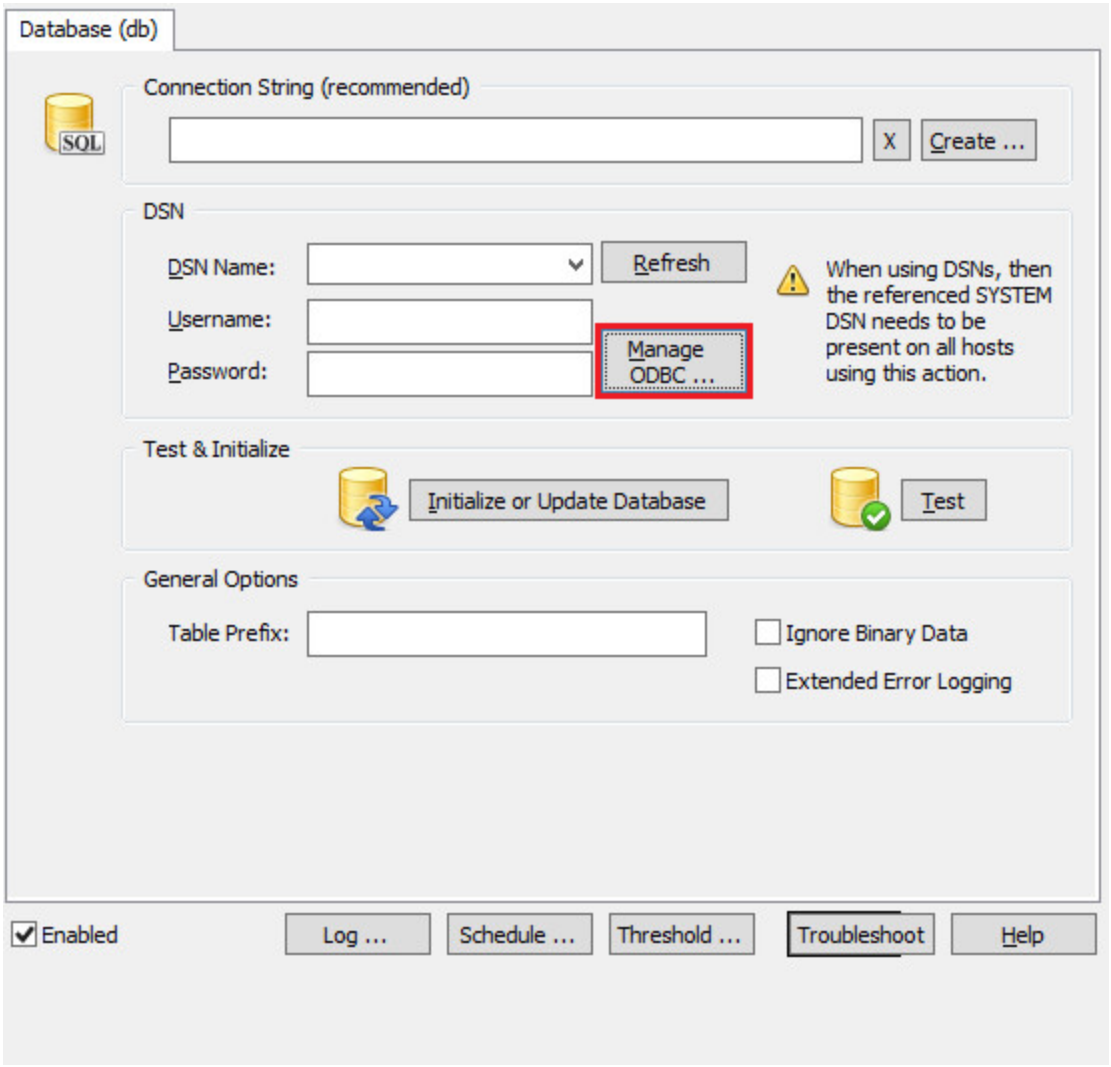
```
[client]
ssl-ca=C:\Program Files\MySQL\MySQL-SSL\ca-cert.pem
ssl-cert=C:\Program Files\MySQL\MySQL-SSL\client-cert.pem
ssl-key=C:\Program Files\MySQL\MySQL-SSL\client-key.pem

[mysqld]
ssl-ca=C:\Program Files\MySQL\MySQL-SSL\ca-cert.pem
ssl-cert=C:\Program Files\MySQL\MySQL-SSL\server-cert.pem
ssl-key=C:\Program Files\MySQL\MySQL-SSL\server-key.pem
```

7. Restart the MySQL server.

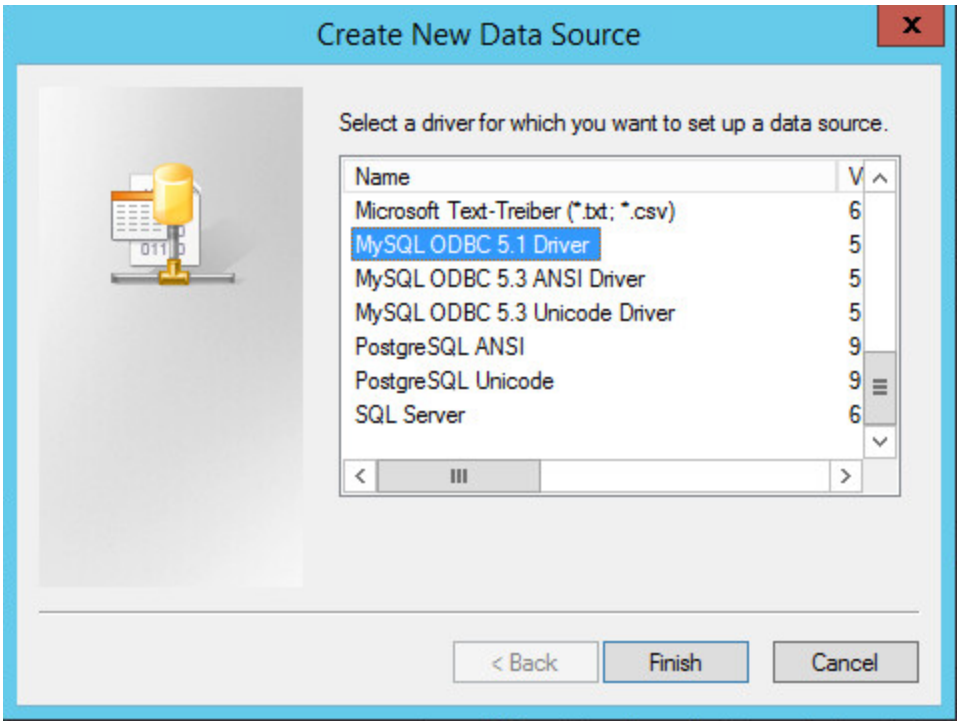
8. Transfer the generated files to your EventSentry server

9. From within EventSentry Management Console click the "Actions" tab and select or add "Database"

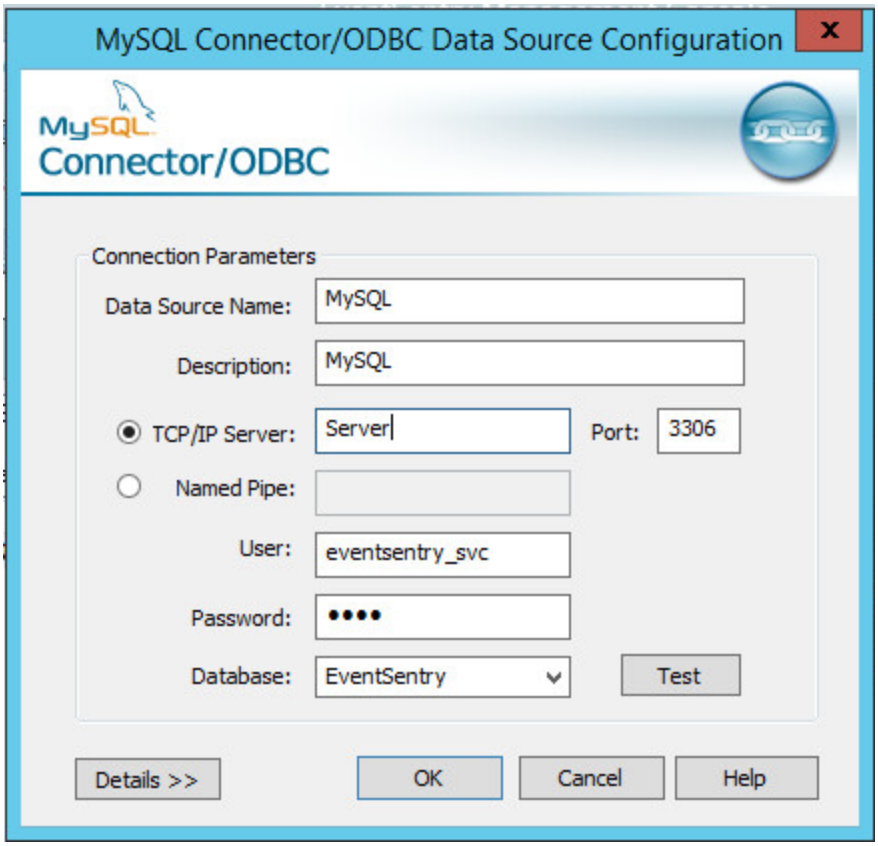
10. On the right panel click "Manage ODBC ..."



11. Click "System DSN" and "Add..."
12. Select "MySQL ODBC 5.1 Driver" from the list and click "Finish"

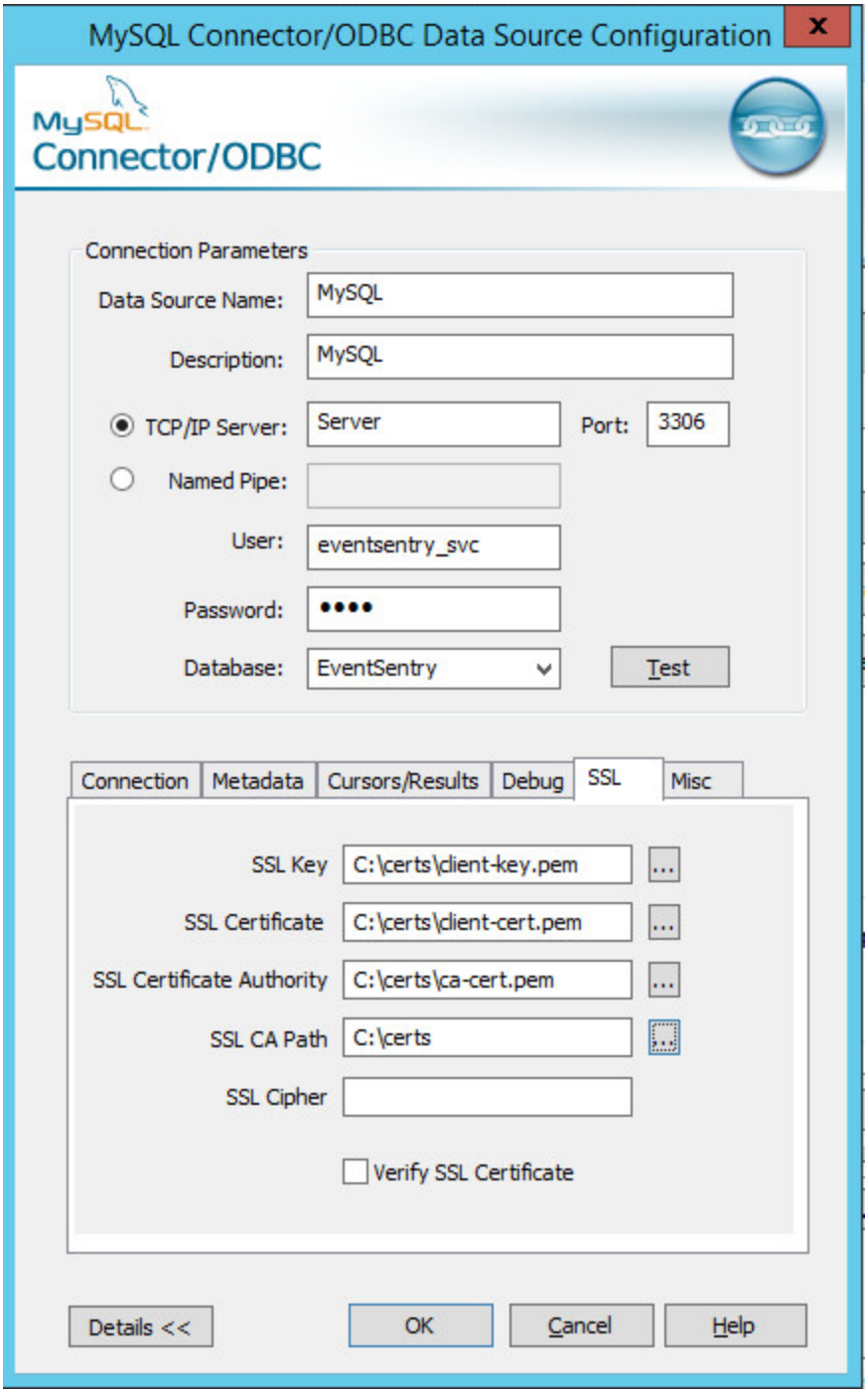


13. Insert your information into the prompt and click "Details > >"



14. Click the SSL tab and setup the following fields:

SSL Key: This is the **client-key.pem** file
SSL Certificate: This is the **client-cert.pem** file
SSL Certificate Authority: This is the **ca-cert.pem** file
SSL CA Path: This path is where the previous three files reside.



15. Deploying the client certificate files

The three client certificate files need to be deployed to all agents transmit data encrypted to the MySQL server. The files can be deployed manually, using an existing 3rd party solution, or using the free AutoAdministrator tool ("File Management" feature) from NETIKUS.NET.

16. Click "Test" and you should have a successful connection using SSL.

4.3 Agents

Even though the EventSentry agents have little attack surface and no security vulnerabilities have been discovered with the EventSentry agents in the past, it might be desirable to modify the account the **EventSentry service** is running under.

By default, the **EventSentry** service runs under the **LocalSystem** account, which gives the EventSentry agent nearly unlimited access to most system resources on the local machine. This is necessary since a regular user, for example, does not have enough permissions to read the security event log or read performance data.

If you are running Windows 2000 or higher, then you can manually change the account the agent is running under by following these steps below:

Create User Account

1. Create a new regular domain user account in your domain, e.g. "EventSentry". It is recommended that you specify in the user account description that this account is used by the EventSentry agents.

Give Permissions for EventSentry Configuration

2. Windows 2003 & later: Open the registry editor **regedit.exe** and select the key **HKLM\Software\netikus.net\EventSentry**. Then, right-click the key and select **Permissions** from the menu and add the newly created user account to the list with **Full** permissions.
3. If you plan on using debug logging, then the newly added user also needs write access to the **%SYSTEMROOT%** directory so that the debug log files which reside in this directory can be created and updated.

Give Permissions for Security Event Log

4. Open the **Domain Security Policy** (Start -> Programs -> Administrative Tools) and navigate to **Security Settings -> Local Policies -> User Rights Assignment**.
5. Add the newly added user to **Log on as a service**.
6. Add the newly added user to **Manage auditing and security log**.

Give Permissions for Performance Monitoring

7. Windows 2003 & later: Open the registry editor **regedit.exe** and select the key **HKLM\Software\Microsoft\Windows NT\CurrentVersion\Perflib**. Then, right-click the key and select **Permissions** from the menu and add the newly created user account to the list with **Read** permissions.

Change Service

8. Open the **Services** application (Start -> Programs -> Administrative Tools) and locate the **EventSentry** service. Double-click the service and select the **Log On** tab.
9. Select "This account" and specify the new user account for the service.
10. You will have repeat steps 5-6 on all computers running the EventSentry agent.

4.4 Collector

Utilizing the collector service, introduced in EventSentry v3.2, will enhance security in a variety of ways as described below.

Traffic Encryption
The collector supports both clear text (not recommended) as well as TLS encryption. Installing the collector service on a more recent version of Windows will ensure that a more secure cipher with a higher bit length will be utilized.

Security Level
Only the medium or high security levels are recommended.

Basic
The basic security level is only recommended for environments where the remote agents connecting to the management console are not listed in the management console.

Medium
The medium security level only lets agents which are listed in a group in the management console connect. This security level should work in almost all environments and is necessary when agents are connecting externally from the Internet through a firewall.

High
The high security level only supports environments where a reverse lookup of a remote agent's IP address can successfully be resolved to a host name, which will in turn need to match the host name configured in the management console. This security level may not work in a scenario where a remote agents connect from the Internet.

Enhanced Action Security
To avoid transmitting the database login credentials (the password of the **eventsentry_svc** user) to remote agents, it's recommended to configure a database action for enhanced security, which prevents the password from being transmitted and stored on the agent(s). Since the agents connect the collector and not the database, the agents do not require the login credentials.

Since only the EventSentry agents support the collector, any host running other EventSentry software (e.g. database import utility, heartbeat service, network services) will still require the full database connection details. A host running additional EventSentry components can be configured to be a trusted host, by editing the host and checking the "Trusted Host" check box.

Network Authorization
Configuring authorized and blocked networks is recommended wherever possible. Blocked networks always take precedence over authorized networks.

4.5 Sysmon

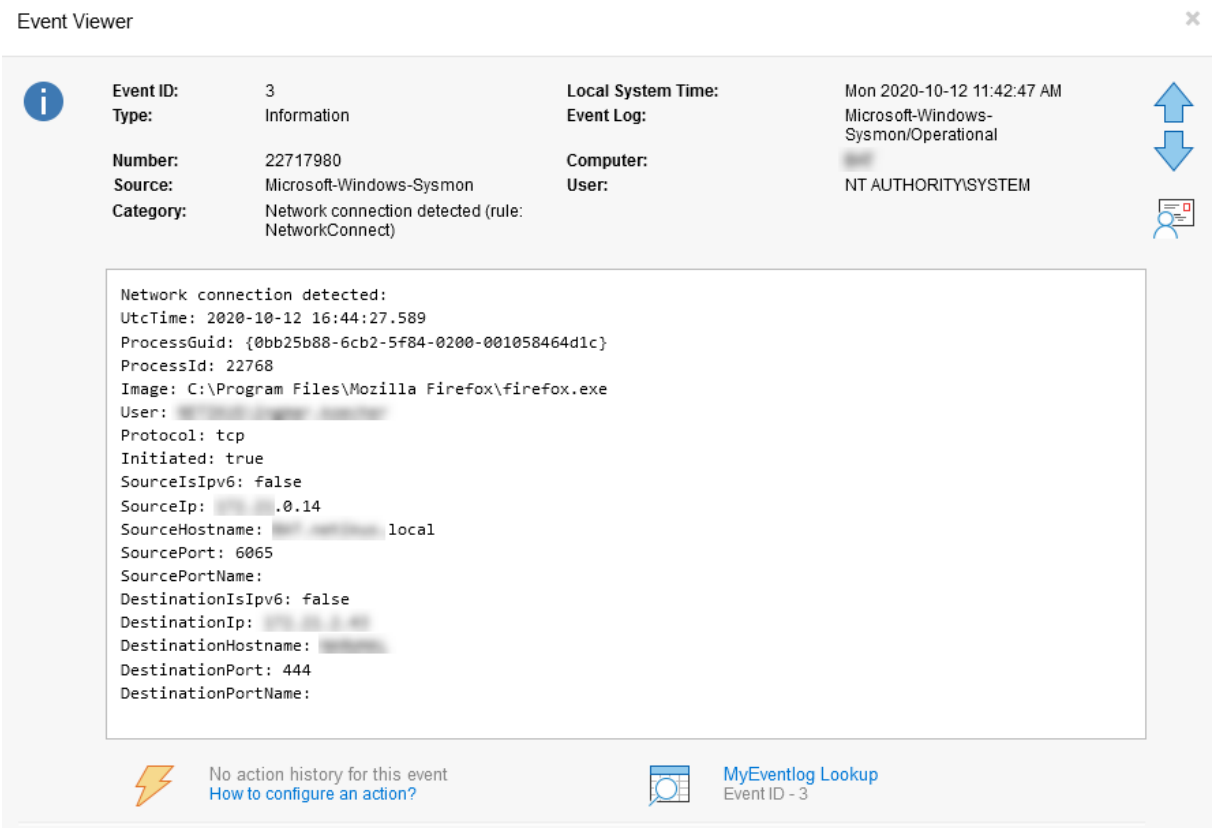
Sysmon, a free utility that is part of the Microsoft Sysinternals Suite, enhances the built-in process auditing capabilities of Windows by providing additional details about processes, most importantly network activity. Analyzing all network activity of processes can help detect Malware and other threats in real time as well as aid with forensic analysis.

Since Sysmon is a free utility and can be automatically installed with EventSentry, it is recommended to deploy Sysmon to all monitored hosts and capture relevant events from Sysmon. The type of information that Sysmon captures is configurable, and the configuration template provided by SwiftOnSecurity (see below) is a good starting point.

To install Sysmon using the SwiftOnSecurity template, follow the steps below:

- Download Sysmon and extract the executables into a temporary folder
- Download the SwiftOnSecurity template from https://github.com/SwiftOnSecurity/sysmon-config and store it in the same directory as sysmon
- Run **`sysmon64.exe -accepteula -i sysmonconfig-export.xml`**
- Monitor the **`Microsoft-Windows-Sysmon/Operational`** event log
- Setup alerts (optional)

Built-In event log filters that can identify malicious activity based on Sysmon events are available here.



Sysmon event showing a network connection initiated by Firefox

5 Event Log Monitoring

EventSentry monitors your event logs so that you can receive certain events via email and to consolidate some or all events into a supported database. You can control which events are forwarded to which notification (please remember that a database is, from EventSentry's point of view, just another notification) with filters.

As such, you will need at least one filter package with an **include** filter to receive events via email. You can use **exclude** filters to exclude certain events from being forwarded to a notification.

Exclude Filters and Thresholds

One of the biggest challenges with receiving event log alerts through mediums such as email is to cut down on the number of alerts you receive. We have addressed this problem with exclude filters and threshold options which can be set for include filters. Exclude filters, as the name implies, allow you to exclude certain events from reaching a notification. Thresholds allow you to limit the number of events that are processed based on time intervals.

Catch-All Packages

Filter Packages can be configured to be "Catch-All" packages, meaning that the filters they contain will processed after all other include filters are processed. This is not relevant for exclude filters (which are always processed before a notification is sent out), but important when you work with include filters that have thresholds applied to them.

If you are using threshold filters that are not in the same package as your "Catch-All" filter (see Event Log Alerts) then it is important that you set the filter package containing your main include filter to be a "Catch-All" package.

5.1 Event Log Alerts

Event Log alerts allow you to receive critical system information through notifications such as email, pager and so forth. It is important to understand that all System Health features of EventSentry log errors and warnings to the application event log which makes it imperative to have event log filters setup that forward warnings and errors to you.

EventSentry ships with a number of default packages containing mostly exclude filters. These exclude filters have been setup to cut down on the number of false positives you would receive compared with a single include filter that forwards all errors and warnings.

When setting up your filter rules, you can basically take two different approaches:

1. Receive all warnings and errors except for certain warning and error events that are non-critical
2. Receive only selected events

This is similar to the approach you have to take when configuring firewalls: You can either configure the firewall to let everything through but block certain services, or block everything and only let certain services through.

We recommend that you take the first approach and configure EventSentry to send you all Errors and Warnings and exclude non-critical Warnings and Errors you might be getting.

The reasoning behind this is quite simple - it is almost impossible to know in advance what events you will be receiving from your servers. By only including events that you anticipate, you are potentially loosing out on being notified when a serious and unexpected error occurs.

5.1.1 Organizing Filters with Folders

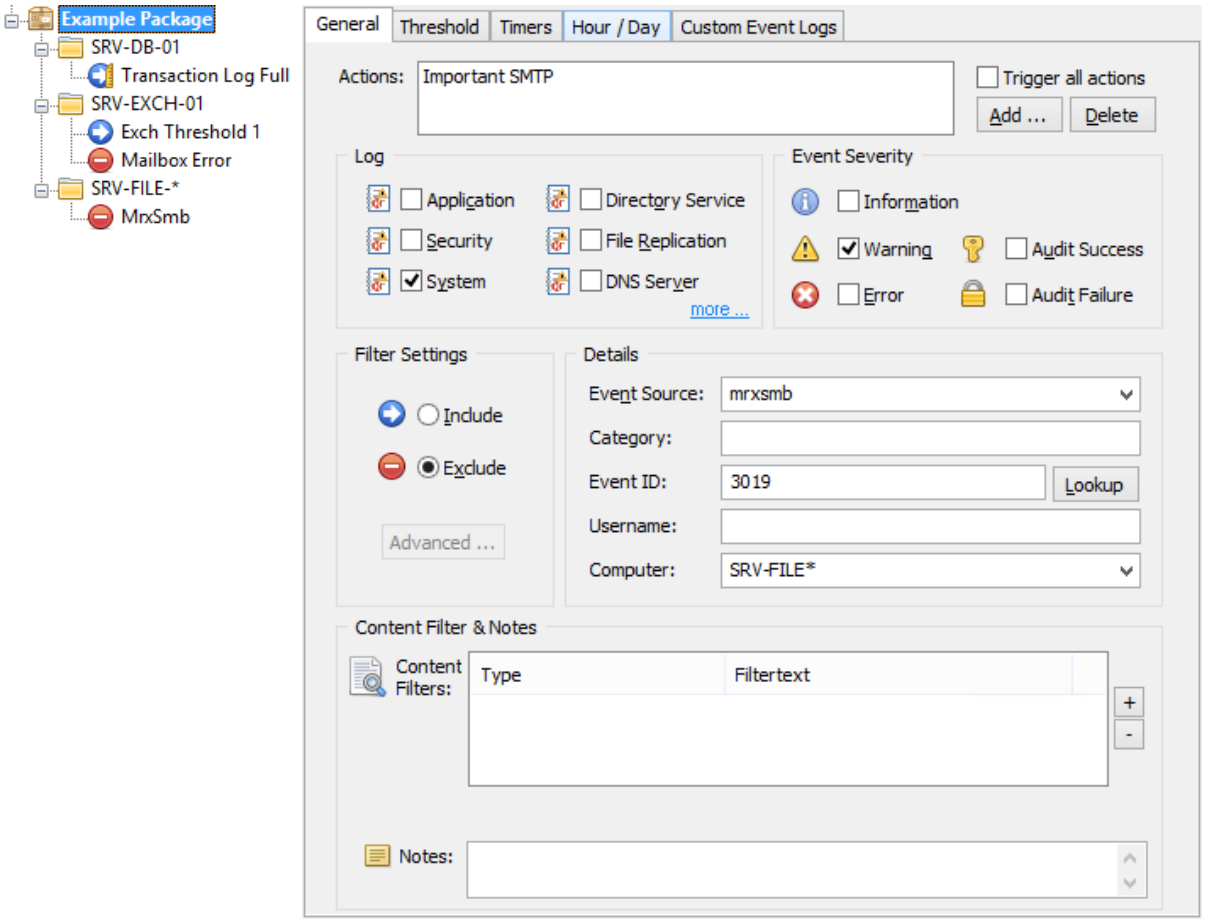
Filters are always grouped into one or more event log packages, and are often organized and assigned in one of three ways:

- filters apply to all hosts network-wide, and are assigned globally
- filters apply to a select number of hosts which share common properties - the package is applied to a group or select number of hosts
- filters apply to a single host only, package is applied only to one host

While creating an event log package designated for a single host can make sense, we recommend organizing filters with folders whenever there are some commonalities among more than one host.

For example, when managing filters for 5 servers, all of which require a small number of customer rules, it can be helpful to create folders based on the host names. But of course other naming schemes can work equally well - as long as they make sense to you and your team.

To make sure that a filter inside the package only applies to a select number of computers, specify the computer name in the "Computer" field of the filter. The "Computer" field supports multiple host names separated by a comma as well as wild card characters.



In the example above, the filters are grouped into folders, whereas each computer (or multiple computers in the case of SRV-FILE-*) has its own folder. The filter itself is always associated with one or more computers.

You can assign this package then either to the computers in question, or make the package global. Making this package global is possible since the filters only apply to computers whose name matches the "Computer" field in the filter.

5.1.2 Excluding Events

The amount of warning and error events logged to the Windows application and system event logs can sometimes be overwhelming. Unfortunately, many events that are not critical errors or warnings are still logged as such to the event log(s) and will by default be forwarded to your email.

With EventSentry you can exclude events that you are not interested in receiving either from all or selected notifications through Exclude Filters. For example, you can exclude repetitive non-critical events from your email notification, but still have them forwarded to a database.

The default installation of EventSentry comes with a number of packages that attempt to reduce and suppress the amount of false alerts you will receive from EventSentry by email by including several filter packages with exclude filters for the most common warnings and errors (e.g. the Windows NT-2k-2k3 Server package).

> The vast amount of software and event log messages logged make it extremely difficult however to offer exclusions for all non-critical events. If you did create custom exclude filters then we encourage you to export your configuration and email it to us so that we can include this information in our default filter packages.

Creating additional Filter Packages for Exclude Filters

If the default packages do not exclude enough events, and non-important error messages are still emailed to you, then you can create one or more filter packages with custom exclusion filters.

If you only plan on adding a small amount of exclude filters (e.g. 5-10) then it is usually enough to create just one filter package. You can also use folders inside the filter package to group filters if they share common elements.

If you plan on adding a large amount of filters however (e.g. 30) then we recommend that you create multiple filter packages for your exclude filters. We generally recommend that you group your exclude filter packages based on the type of events they include. For example, you can create one filter package for your file servers, and one exclude filter package for IIS related events and then assign those two packages accordingly.

Excluding events directly from the built-in Event viewer

Instead of creating an exclude filter manually from scratch you can have EventSentry create most of the filter properties automatically for you.

Simply locate the event you would like to exclude in the built-in event viewer (if the event is on a remote computer then you will need to open the event log on the remote computer first by right-clicking the "Event Log Viewer" container), right-click the event and select "Add Exclude Filter".

When prompted, enter a name and select a package for the filter and EventSentry will create the filter in the package for you. Once the filter has been created you will need to assign a notification (if no notifications are set on a package level) and optionally move it to a different position in the package.

5.2 Security Alerts

Receiving security alerts via email (or similar notifications) often requires additional steps so that your email inbox is not flooded with audit failure events.

Filter thresholds allow you to accommodate most scenarios in which you want to receive notifications based on events in the security event log. Common scenarios include requirements such as:

- be notified if a user attempts to login with a wrong password more then X times in Y minutes
- be notified if there are a large amount of audit failures during a short time interval
- be notified when a .exe file in the system32 directory has been modified
- be notified when certain applications (.exe files) are launched

The pages in this chapter will explain how to accomplish some of the above scenarios using filters and threshold options. Please click the following links for more examples:

1. Threshold filter to detect a large amount of audit failures
2. Threshold filter to be notified when a user logs in with wrong password
3. Include filter to detect file changes in selected directories

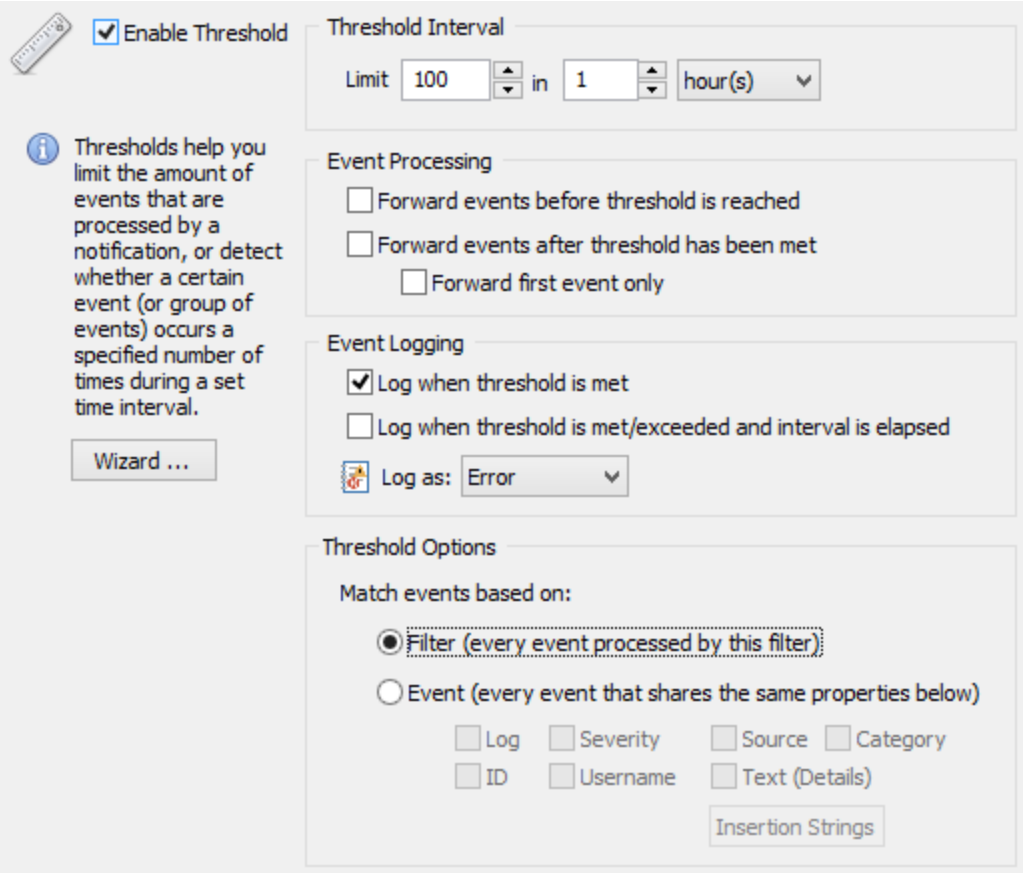
5.2.1 Large Amount of Audit Failures (Threshold)

You can use threshold filters in a variety of scenarios, one of them being to notify you when a large amount of audit failures are written to the event log.

This is especially useful when used in combination with a database consolidation: Once unusually high activity is detected in the security log you can immediately investigate the events collected in the central database.

Let's assume that any given domain controller gets approximately 50 audit failures an hour, and you would like to be notified if more than 100 are logged in an hour.

To accomplish this, create an include filter that matches Audit Failure events (*e.g. Log=Security;Severity=Audit Failures*) and add the following threshold options to the filter. An explanation is giving below the screenshot.



Event Logging: Log when threshold is met

Checking this box will ensure that an **Error** event (according to the pull down selection right below it) is logged to the event log when 100 events have been written to the event log. The actual events are not forwarded to the notification.

Match events based on: Filter

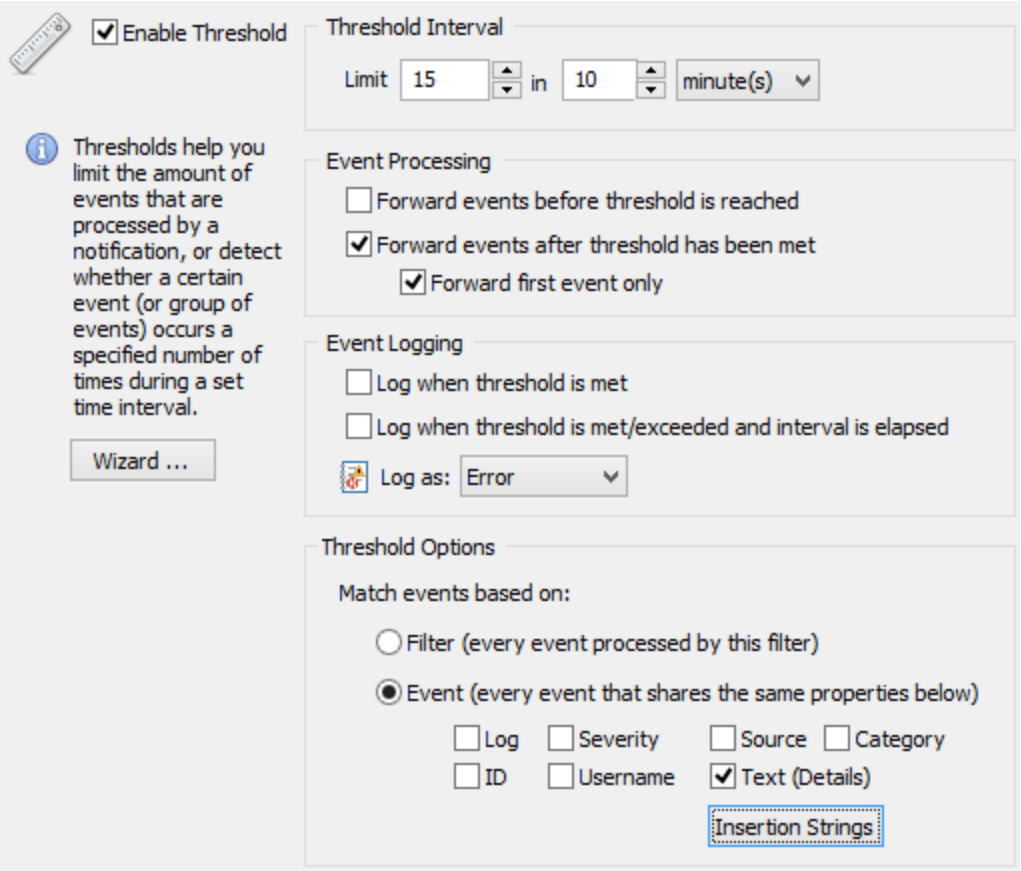
Since we need to match all events, regardless of their detailed properties, the filter should increase its internal threshold counter with every event that matches the filter.

5.2.2 Wrong Password (Threshold)

Audit Failures pertaining to failed logon attempts are a common scenario on domain controllers in large networks, and setting up a filter to notify you when a user types in the wrong password will most likely result in hundreds of emails being sent to you.

In this example we will want to be notified if somebody types the wrong password more than 15 times during 10 minutes (or less).

To accomplish this, create an include filter that matches failed login attempts (*e.g. Log=Security;Severity=Audit Failures;Event Source=Security;Event ID=675*) and add the following threshold options to the filter. An explanation is giving below the screenshot.



Event Processing: Forward events when/after threshold has been met

This option ensures that you are receiving events after the threshold of 15 has been met, however this could still result in many emails being sent if somebody is trying 100 different passwords. The option Forward first event only will make sure that you only receive the first event after the threshold has been met, that is the 16th event.

Match events based on: Text (Details)

The default option for thresholds is "Match events based on Filter", which means that the internal counter used by the threshold filter is increased every time an event matches the filter, even if it is from different user accounts. This is clearly not desirable in this case, as we want the filter to have a separate internal counter for each user.

This setting essentially tells the filter to keep/start a separate counter for each unique event text it counters. For this example an event text that is logged on the domain controller might look like this:

```
Pre-authentication failed:
User Name: myuser
User ID: MYDOMAIN\myuser
Service Name: krbtgt/NETIKUSNET
Pre-Authentication Type: 0x2
Failure Code: 0x18
Client Address: 192.150.3.20
```

It is also possible to filter this event on the workstation, in which case the event id would be 529 and the event text would look different:

```
Logon Failure:
Reason: Unknown user name or bad password
```

```
User Name: myuser
Domain: MYDOMAIN
Logon Type: 7
Logon Process: User32
Authentication Package: Negotiate
Workstation Name: WORKSTATION1
```

It might be tempting to click the **Username** checkbox instead of the **Text (Details)** checkbox, however this would not work since all events are logged by the NT AUTHORITY\SYSTEM user account.

Event logs should be investigated immediately using the web reports on event viewer when EventSentry notifies you of many failed login attempts so that corrective action (e.g. temporarily disable the user) can be taken.

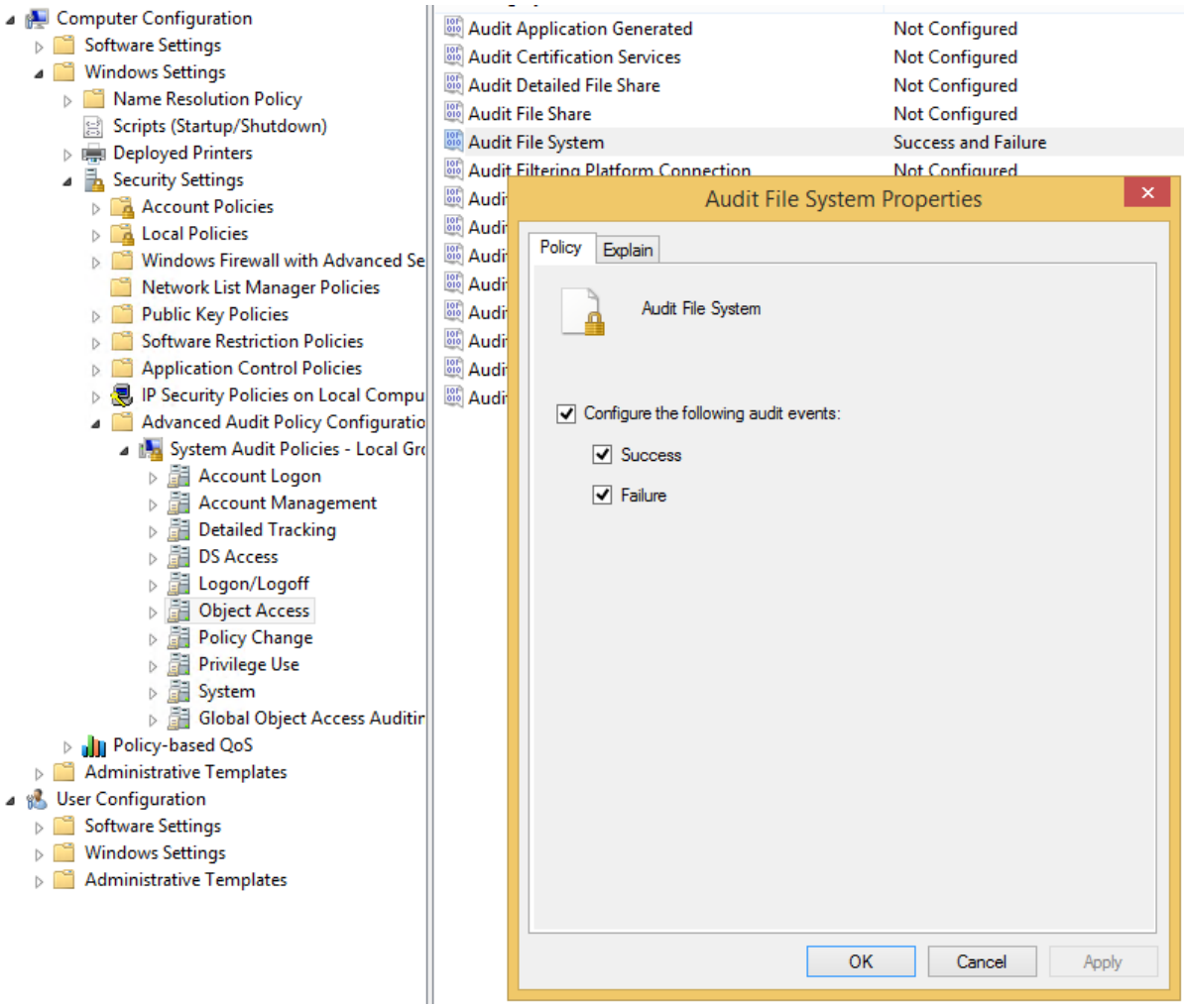
5.2.3 File Changes

If you combine the object auditing capabilities of the Operating System with event log monitoring capabilities then you can **be notified when a file is deleted in a directory**.

In the following example we will configure Windows and EventSentry to notify us when a file is deleted from the **C:\Documents** folder.

1. Enable Object Auditing

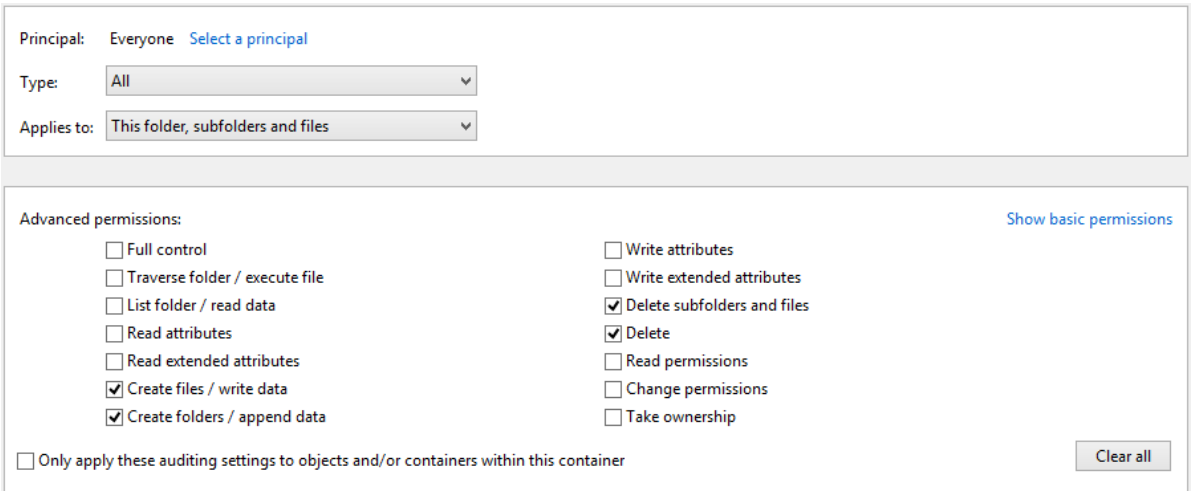
Before we can enable auditing on a folder, we need to enable "Audit object access" in the group policy of your domain or server. You can find this auditing object in the "Security Settings - Advanced Audit Policy Configuration - System Audit Policies - Object Access" container. Make sure that at least "Success" is selected:



2. Auditing a folder on Windows

After object access has been enabled, you need to configure auditing in the file system. Using explorer, navigate to the folder you want to audit (**C:\Documents** in our case), right-click the folder and select "Properties".

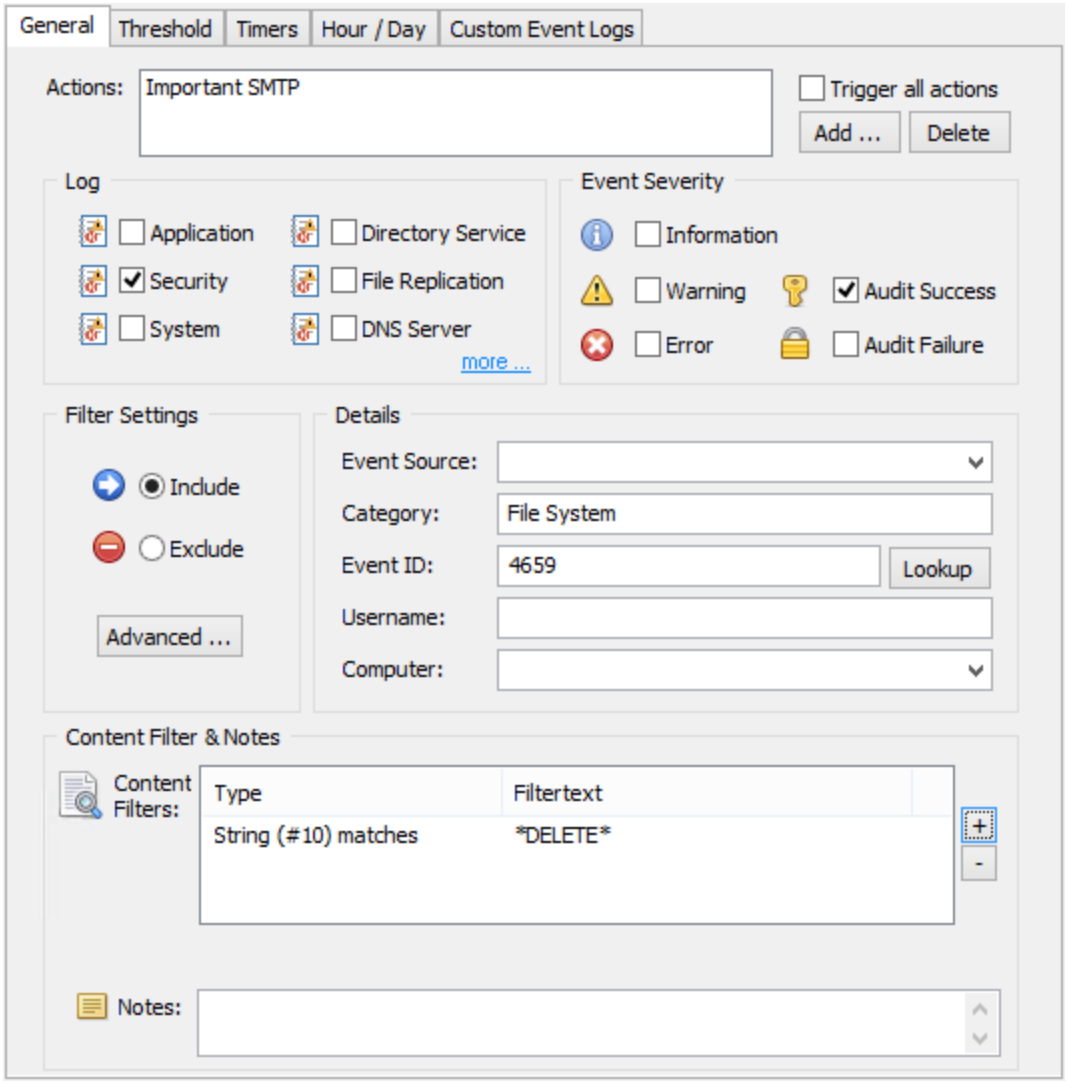
On the "Security" tab, click the "Advanced" button to get to the "Advanced Security Settings" for the folder. There, click the "Auditing" tab and select "Add". Now specify an account you would like to audit (we recommend "Everyone") and select the following types of Access shown in the screen shot below:



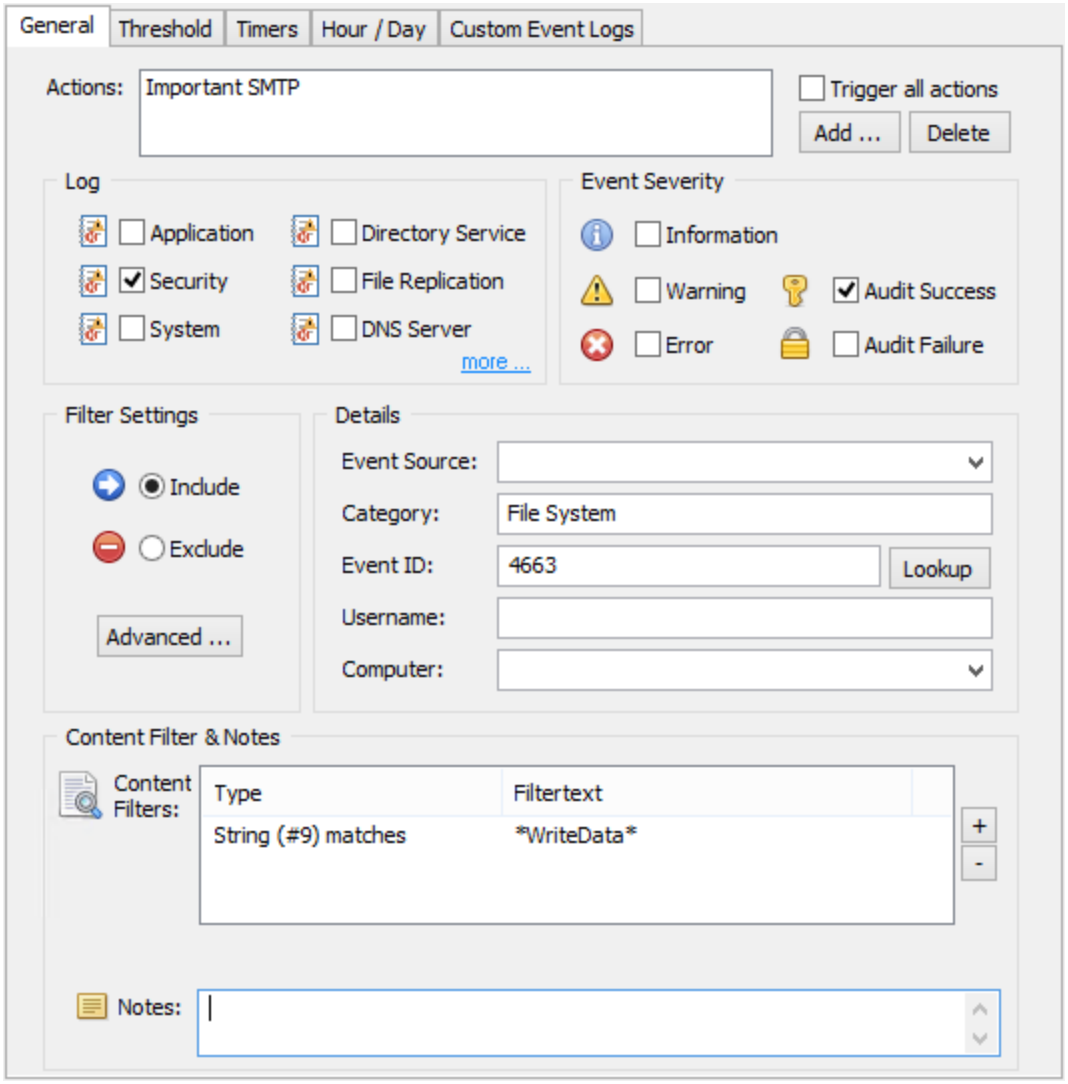
After dismissing all the open dialogs with OK auditing will be enabled in the selected folder and EventSentry is ready to forward events.

3. Creating an Include Filter

Now that the OS will log write access to the **C:\Documents** directory, we can add a filter that will forward Audit Success events to a notification based on the properties of the event and the details of the event message. The filter below shows how to setup the filter text for this particular event 4659:



To know when a file is being changed instead of being deleted, adapt the filter so it matches the screen shot below:



5.3 Recurring Events

Many software packages, including but not limited to backup software and antivirus software, log events to the event log when certain repetitive tasks, such as a Virus definition update or a backup job have completed successfully.

Instead of being notified every time such an event happens, EventSentry can verify that these events have been written to the event log during a preset time interval using the "Recurring Event" feature. If an expected event does not appear in the event log, EventSentry will write an Error to the Application event log.

Setting up recurring event is fairly easy, and requires you only to know what type of event you are expecting and when you are expecting that event.

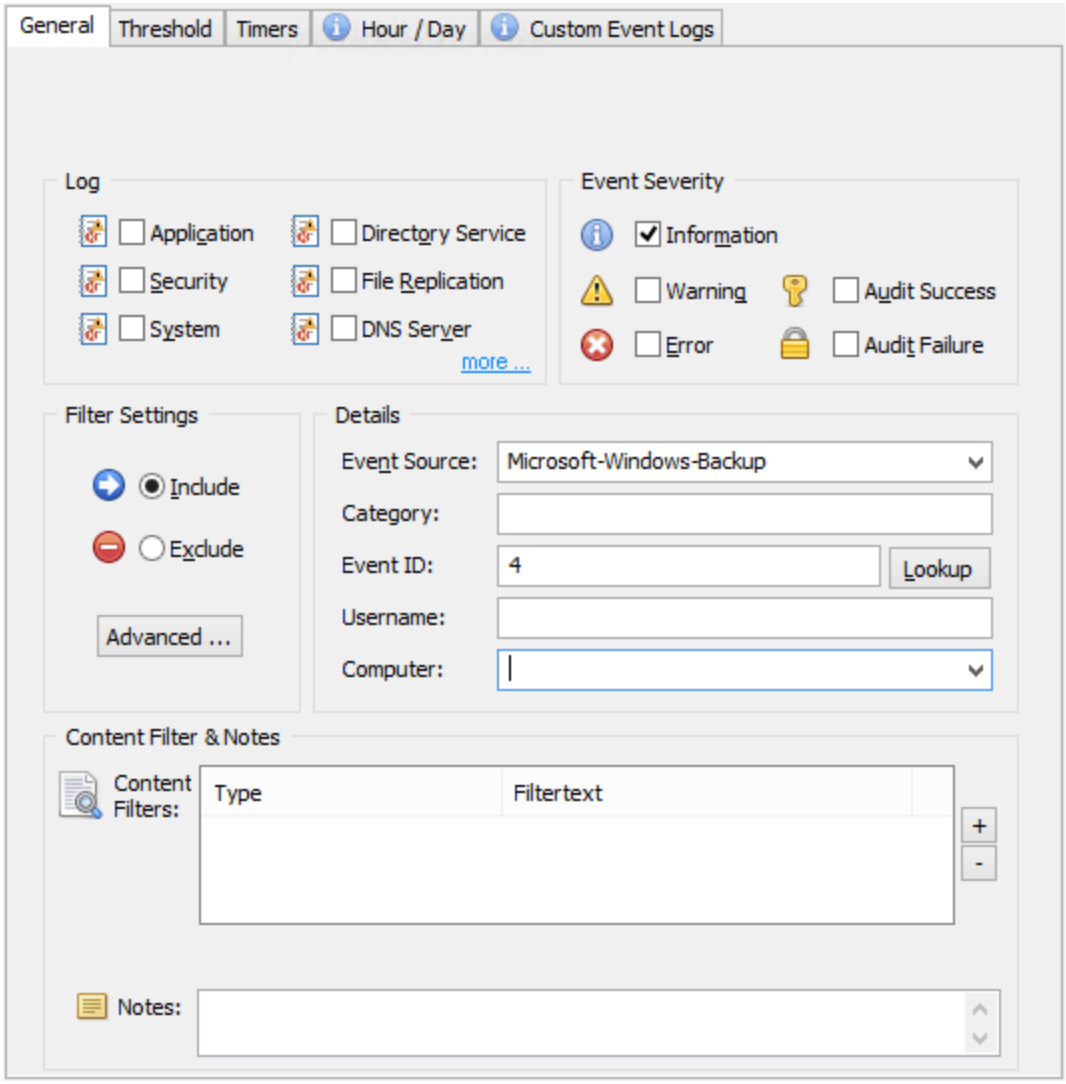
For more information on recurring events and more in-depth example please see the Backup with EventSentry and NTBackup guide, available from the Guides section at http://www.netikus.net.

5.3.1 Verifying Backups & AntiVirus Updates

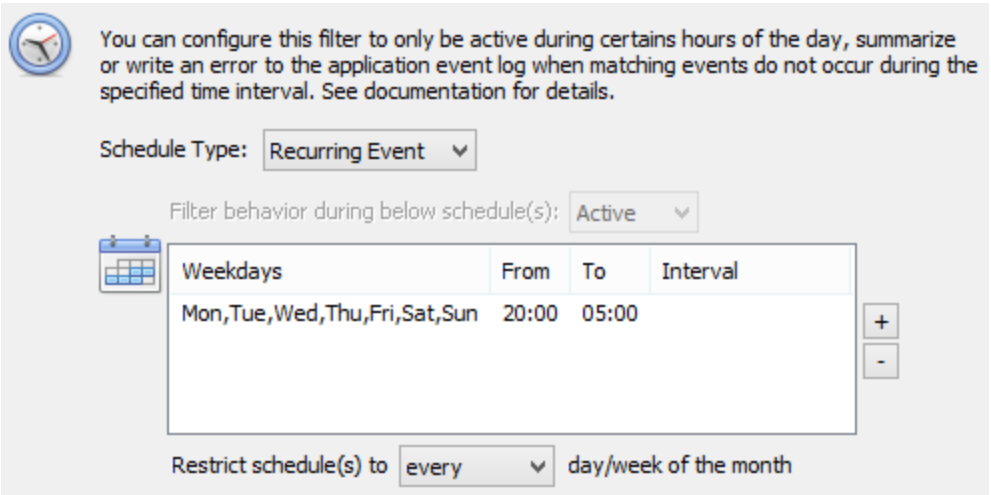
In this example we will setup a filter that will notify us when a backup event is not written to the event log between 8PM and 5AM in the morning from Tuesday to Saturday. Windows backup logs an informational event to the Microsoft-Windows-Backup event log with event id 4 and the event source "Microsoft-Windows-Backup".

1. Creating the recurring event filter

Create a new package or add a new filter to an existing package with general settings as shown in the screen shot below:



Then, configure the recurring event options by clicking on the **Hour/Day** tab and duplicating the screenshot settings shown below:



The pushed buttons represent the hours between which the event should occur, and the schedule type needs to be set to **Recurring Event**. If the event configured in the General tab of the filter does not appear in the event log at the specified time, then EventSentry will log an **Error** to the Application event log with the event id of **10620**. Please see Recurring Event Filters in the manual for additional information.

If you don't have a Catch-All filter in place that forwards errors from the event logs to you, then you will need to add a 2nd filter to this or another assigned package that will forward the error (LOG=Application;Source=EventSentry;ID=10620) logged by EventSentry to you.

5.3.2 One Event Every 10 Minutes

Let's say you have a custom application, written in-house, which logs an event to the event log every 5 minutes. You need to know when that application has stopped logging this key event, as it most likely indicates the application is no longer running or is experiencing some other issue preventing it from working properly.

Starting with version 3.0.1, the recurring event filter supports checking for any type of event in as little as one minute intervals.

To get notified when an event is no longer being generated, an event log include filter needs to be created and configured as a "Recurring Event" filter. We will use a 10-minute interval (opposed to a 5-minute interval) to allow for a buffer, but the filter can of course be set for 5 minutes as well.

Recurring Event Filter

Create an event log "Include" filter and configure the "General" tab based on the event properties.

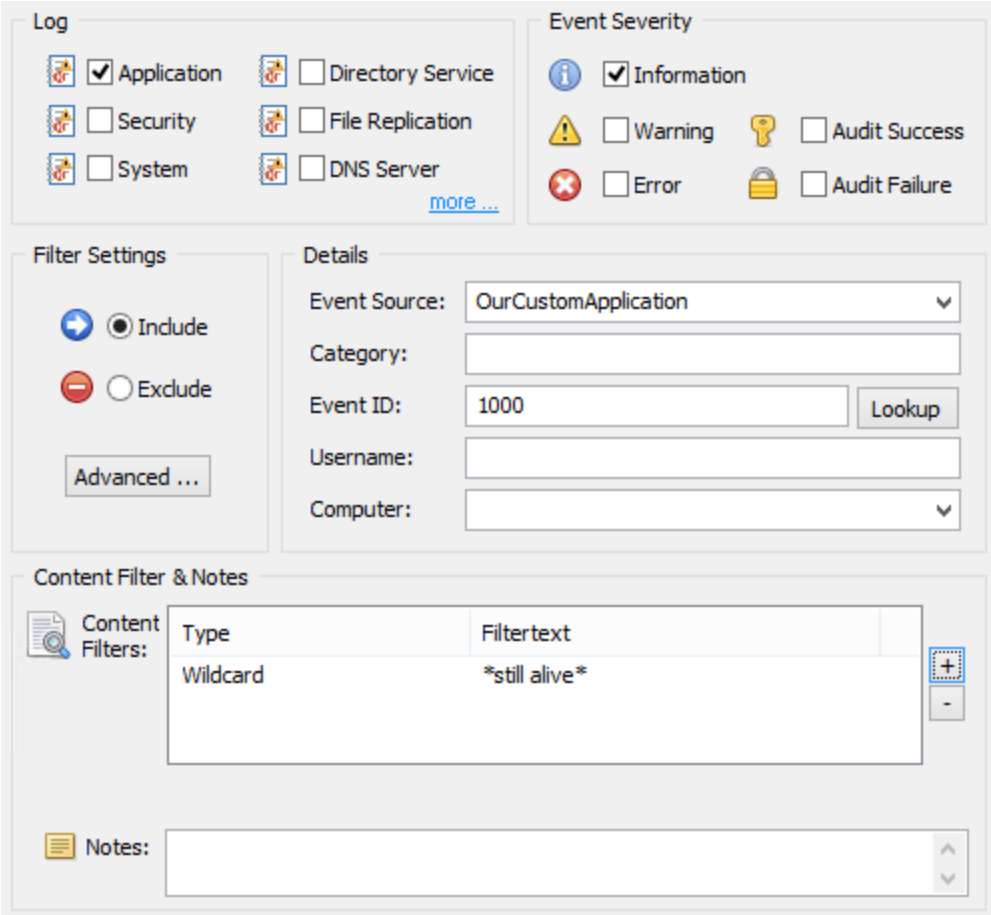


Figure 1: The general settings of the recurring filter

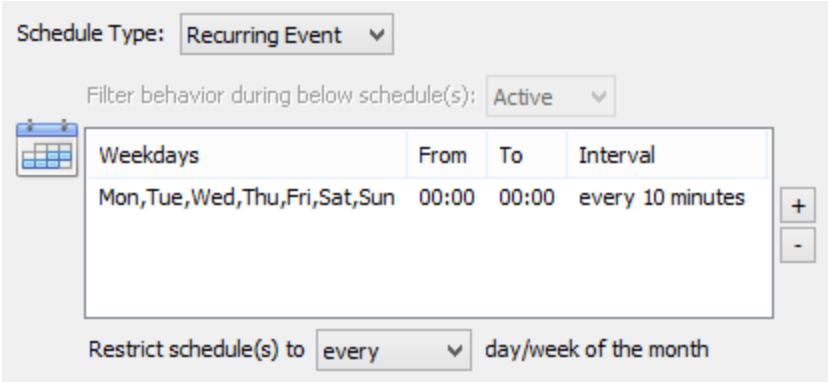


Figure 2: The **Hour / Day** settings with a 10-minute interval

6 Event Log Consolidation

Many government regulations in the United States and other countries require you to collect and archive event logs for a certain period of time. With EventSentry you can consolidate all or some of your event logs, according to your rules, into any of the supported databases.

Database Consolidation can be setup very quickly and requires only a few steps. All of the steps below are automatically performed by the installer if you have one of the supported databases available during the installation. Please see Steps to Event Log Consolidation for more information.

- Create a database action in the management console
- Click the "Initialize or Update database" button to invoke the configuration assistant
- Create an event log filter that forwards some or all events to the database

Connection Strings vs. System DSNs

When creating your database notification you have the choose between using a System DSN and a connection string. We strongly recommend that you use a connection string instead of a DSN, since using a DSN will require you to create that same DSN on every computer that will be writing to the EventSentry database. If you have to use a DSN, then you can use AutoAdministrator to push/duplicate an existing DSN to remote computers.

ODBC Drivers

Built-In Database (PostgreSQL)
Included with EventSentry and installed automatically when PostreSQL is used.

Microsoft SQL Server
Included with Microsoft Windows

MySQL
Included with EventSentry and installed automatically when MySQL is used. MySQL also offers an MSI package that can be rolled out using Active Directory.

Oracle
It is fairly complicated to install the ODBC drivers on a Microsoft Windows machine since you are required to install everything using the Java-based Oracle Universal Installer. If you plan on using Oracle then please keep in mind that you will have to install the ODBC drivers using the Oracle Universal Installer on every computer that is to write to the database.

6.1 Archival and Purging

Purging old records periodically

Collecting event logs can create an enormous amount of data into your database, and purging old records that do not have to be stored anymore is essential. The page Purging Records Periodically offers detailed instructions on how to delete old data from your EventSentry database.

It is at this point not possible to move unneeded records to a 2nd database for long-term archival. Please see the next section for a work-around.

Creating a Database for Archival

If you need to have one database for immediate and fast access to event log data (e.g. the last 30 days), but also need to store events for archival (e.g. store events for 360 days), then you can configure EventSentry to use two databases.

The first database will receive all the necessary events, and events older than 30 days will be purged every day or week. This way the database will remain small and access to the event log information will be very fast.

The second database will also receive all the necessary events, however only events older than 360 days will be purged every week or every month. The two databases can be on the same server or on completely different databases or database engines. As long as the databases have been successfully initialized with the Database Setup Wizard you can store events in it.

You can also easily query both databases from your web browser by creating a 2nd profile in the EventSentry Web Reports.

6.2 Off-Peak Consolidation over WAN

It is possible to schedule event log consolidation during non-business / off-peak hours if the servers or workstations you are monitoring are located across a WAN. This makes it possible to reduce bandwidth consumption significantly during business hours.

This functionality can be easily achieved by setting a summary notification on the filter(s) that are used to forward events to a database. Since you can assign different packages to different servers/groups, it is easily possible to configure machines located in the same LAN as the database server to write events immediately to the database, yet schedule remote machines across a WAN during off-peak hours.

1. Creating a new group

If only some of your monitored machines are located across a WAN then it is recommended that you create a new group for those machines - if you haven't done that already. Right-click the "Computer Groups" container and select "Add Group". Assign a descriptive name to the groups.

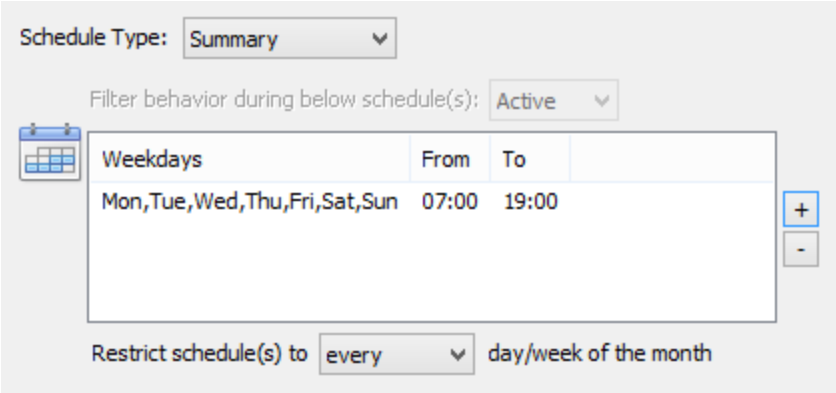
2. Create a new filter package

Right-click the "Filter Packages" container and select "Add Package". Assign a descriptive name to the package (e.g. "Database Consolidation over WAN"). You can skip this step if you already have a filter package that is only used by servers across the WAN.

3. Creating a summary notification filter

Add a new **include** filter to the package and configure it to forward the desired types of events to the database notification, for example all Information, Warning, Error and Audit Failure events. However, unlike a regular filter, we will assign a summary notification to this filter so that events are queued and not sent immediately during business hours.

To assign a summary notification, click the **Hour/Day** tab of the newly created filter and make sure that all the hours during which you want to queue events are raised. Every push button represents one hour of the day, and in the example below we will queue events from 7AM to 7PM, whereas events between 7PM and 7AM will be sent to the database immediately:



Summary notifications are quite flexible and can also be used to receive a daily email report from a server for example. For more information on summary notifications see the manual.

6.3 Conserving disk space and optimizing performance

Consolidating event log and system health data into a central database is an extremely useful feature, but the wealth of data being collected can cause disk space and performance problems in small and large networks alike. This chapter will explain how to reduce the amount of data being logged by:

- Identifying top event log entries being logged (without loosing critical information)
- Suggesting performance monitoring
- Examining process tracking options

The chapter will also give recommendations on database performance optimization.

Event Log Data

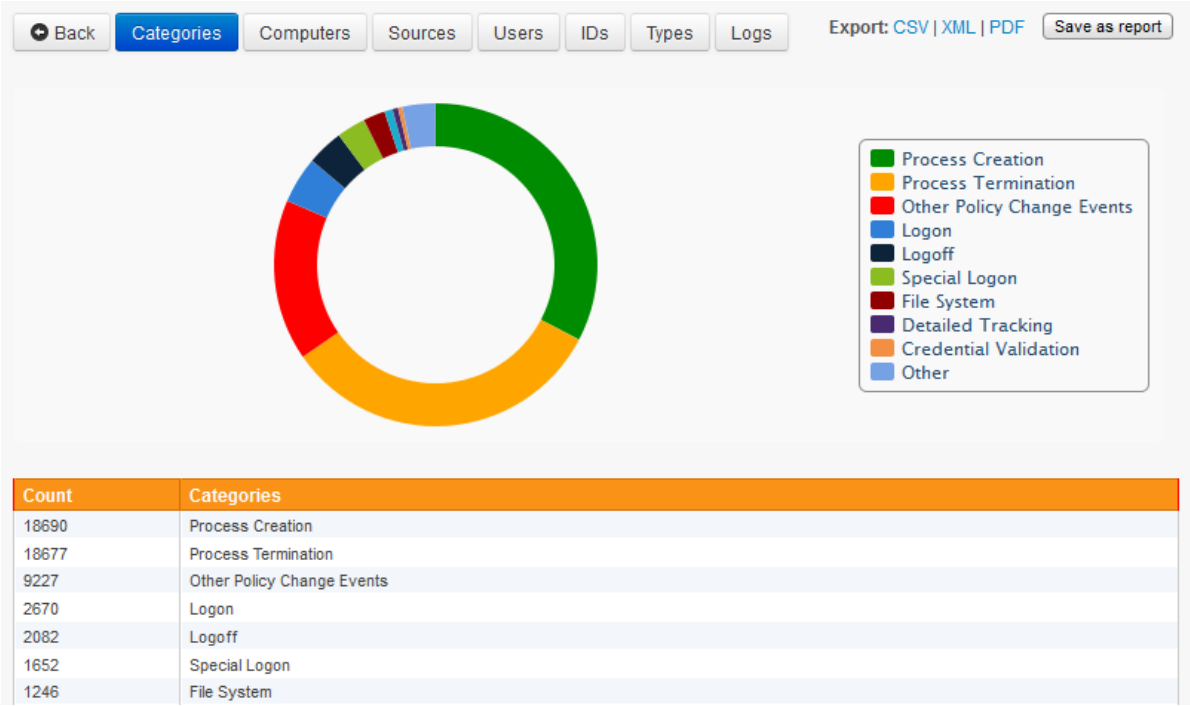
It is very easy to fill up any database when consolidating all event log entries from all monitoring machines to a central database. With heavy auditing in place, it is very easy to create millions for records every day on a small network and thus bring /any database server down to its knees. As such, if you find that the EventSentry database is growing out of control, then it is important that you first identify which non-essential event log records are being consolidated, so that they can be excluded later.

To find out which events are using up most of the space, simple navigate to "Logs - Event Log" and review the default "Summary" view which shows the most active computers, event logs, sources, types etc. You may need to adapt the duration field which is set to "Last Hour" by default.

Identifying the culprit

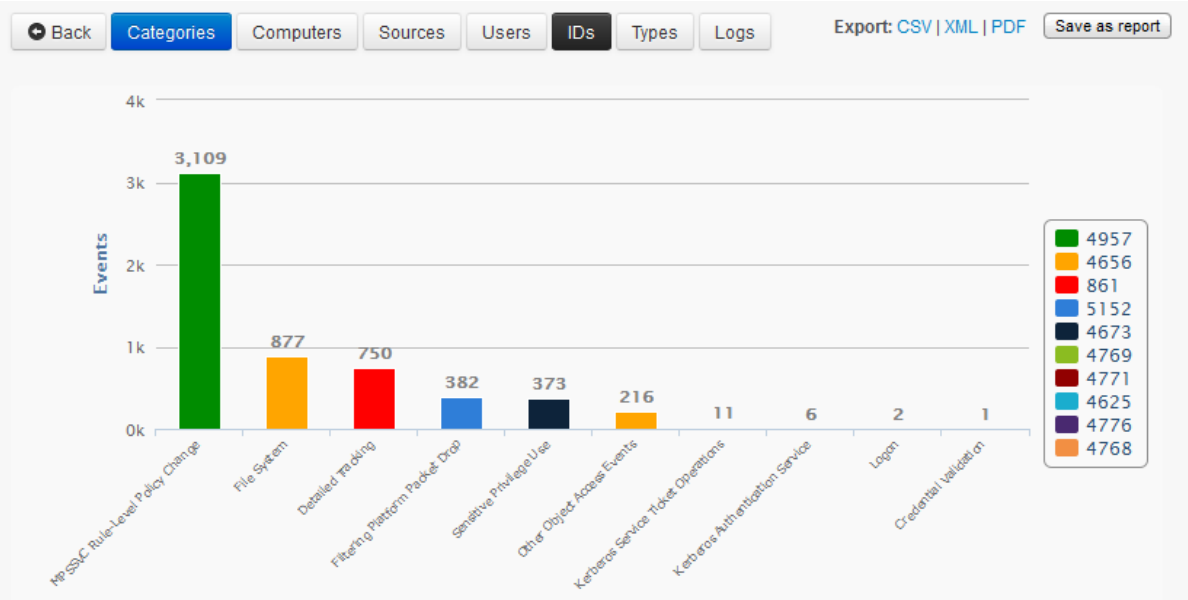
For example, to find which event categories appear most often in the EventSentry database, navigate to the EventSentry web reports and open the "Event Log" page (under Logs). Then click the blue **Categories** or **icon** next to it which yield a bar chart about all available categories. You might also want to modify the search time scope on the top right.

The left example below shows that 67% (hover over the pie chart with the mouse to see the percentage) of all events written in the last 24 hours are from the **Process Creation** and **Process Termination** category for example.



Count	Categories
18690	Process Creation
18677	Process Termination
9227	Other Policy Change Events
2670	Logon
2082	Logoff
1652	Special Logon
1246	File System

Instead of grouping events by only one dimension (categories in this case), we can also group them by a 2nd dimension - for example event ids. With the log query now restricted to only events from the security event log and audit failures, clicking the "IDs" button will yield a chart similar to the one below:



Count	Categories	IDs
3109	MPSSVC Rule-Level Policy Change	4957
877	File System	4656
750	Detailed Tracking	861
382	Filtering Platform Packet Drop	5152
373	Sensitive Privilege Use	4673
216	Other Object Access Events	4656
11	Kerberos Service Ticket Operations	4769
5	Kerberos Authentication Service	4771
2	Logon	4625
1	Credential Validation	4776
1	Kerberos Authentication Service	4768

5,727 matching records

The list immediately shows which event IDs are most prevalent, and with which event category they are associated with.

Determining which events to exclude

Now that know which events use up 97% of the disk space, you can run detailed event searches to see if the events which are being logged need to be consolidated. For the first line, simply click the **Reset Form** button and then select the event ID **562** and the event category **Object Access.** When running the search, you are encouraged to limit the search results (e.g. 500 records) and also specify a time range, such as the last 24 hours for example.

Repeat this process for all events that occupy the majority of database space to determine which events can be excluded. Once you have identified events which can be excluded, setup one or more exclude filters for these events. Please consult the help file or Excluding Events in this document for more details on excluding events.

Performance Monitoring

Even though performance data uses less disk space than event log consolidation (since no event message are being logged), performance data can still fill up the EventSentry database quickly, resulting in slow queries and a large database size. Fortunately, EventSentry's performance collection feature is very flexible and you can accumulate very useful and descriptive performance data without using up too much space in the database.

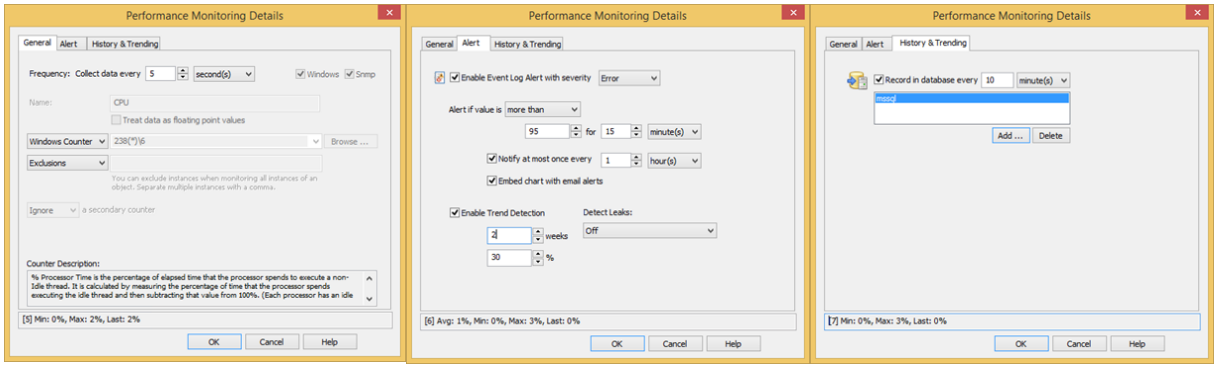
When collecting counter data in the EventSentry database, we recommend that you always set the database logging interval significantly higher than the counter collection interval. EventSentry automatically calculates the average over all collected performance values over one database iteration, ensuring that the data stored in the database is very accurate.

The screen shots below show a correctly configured database configuration for the **Processor(*)\% Processor Time** performance counter, which records the current CPU usage in percent.

Even though the CPU performance counter is queried every 5 seconds (Polling Interval), the calculated average is only written to the database every 10 minutes.

Even with this configuration that "only" logs information to the database every 10 minutes, you will get 144 data points for this counter every day. If you were to write performance counter data to the database every 5 seconds instead, then you would accumulate 17280 data points every day instead!

You might not be able to follow these suggestions if you need a more accurate picture of counter data, but you are still encouraged to make sure that the database logging interval is larger than the polling interval.



Process Tracking

Process Tracking data is similar to event log data, though it too uses less space in the database than event log data since no event messages are being recorded. Process Tracking can give you an enormous amount of information about the state of processes on a given server or workstation at any given time, but it can also fill up the EventSentry database quickly.

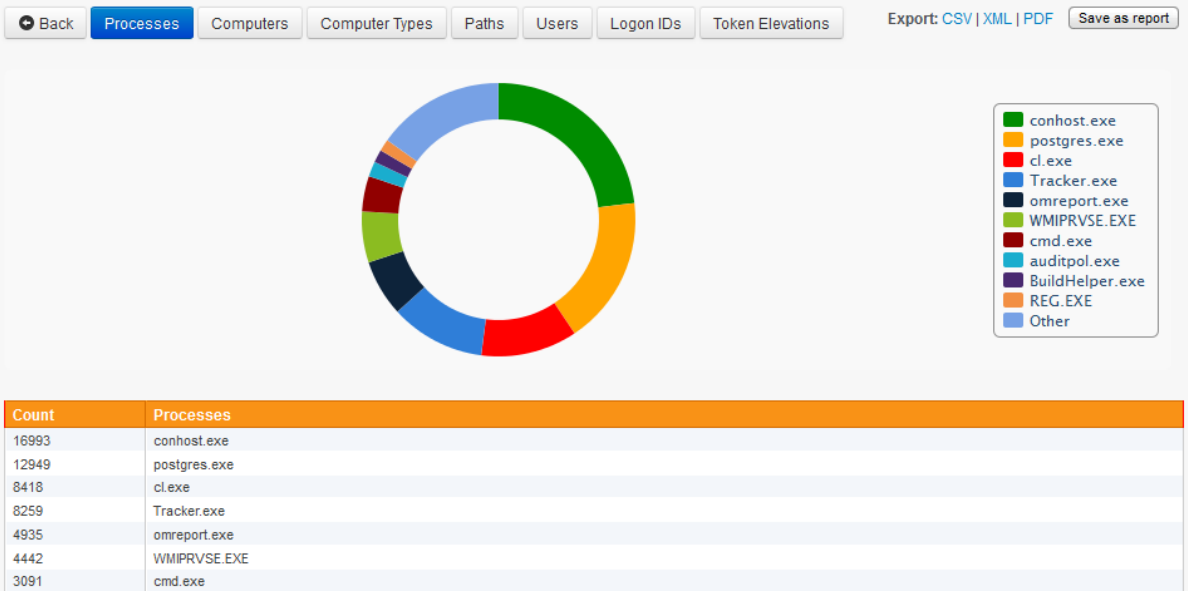
For example, many servers execute processes on a regular basis that do not need to be recorded, for example a monitoring server might execute the ping.exe process every 10 seconds, resulting in 8640 rows of data every day. We can reduce the amount of data logged by excluding unneeded information from the database.

Identifying the culprit

To find which process appears most often in the EventSentry database, navigate to the EventSentry web reports and open the "Processes" page (under "Compliance"). Then click the **Processes** header. You might also want to further restrict the search by only showing records from the last 24 hours for example.

The screenshot to the right shows that the **conhost.exe**, **postgres.exe** and **cl.exe** processes are being executed most often, accounting for more than 50% of the process log entries by adding approximately 40.000 rows every day.

To stop these processes from being monitored, simply exclude them in the **tracking package** that contains the **process tracking** object.



Count	Processes
16993	conhost.exe
12949	postgres.exe
8418	cl.exe
8259	Tracker.exe
4935	omreport.exe
4442	WMIPRVSE.EXE
3091	cmd.exe

Database Optimization, Hardware Planning

If you have followed the steps above and ensured that you are only logging the data that you need to the EventSentry database, then you can tackle the next step, making sure that your database is running on the proper hardware and performing well.

It is obvious that no software can do wonders when the underlying hardware is insufficient. Please consider the following suggestions if you plan on making a hardware purchase for a new database server, or are considering a hardware upgrade for your database server. Please note that these suggestions assume that your database server will only be used for the EventSentry database.

Disk Subsystem

The disk subsystem is one of the most crucial components of a database server, if you have slow disks then your database queries will almost always be slow, even when you have ample CPU power and memory available. The ideal disk subsystem should look like this:

Number of disks	RAID Level	Used for
2 x 10k or 2 x 15k SCSI	1 (mirror)	Operating System
2 x 10k or 2 x 15k SCSI	1 (mirror)	Database Transaction Logs
5 (or more) x 15k SCSI	5	Database

The more disks you can provide for the database partition, and the faster the disks, the better the query response time will be.

It is highly recommended that you put the EventSentry database on a separate partition to avoid disk defragmentation. We have seen performance improvements with disk defragmentation utilities such as Raxco's PerfectDisk® in some cases.

Memory

The amount of memory installed is also crucial, and your database server should have at least 2Gb of memory. If you are running Microsoft SQL Server, then 4Gb or more are recommended.

CPU

The number of CPUs is not as crucial as the previous two components (one CPU should suffice in most cases), however you should still ensure that a recent CPU model (e.g. Pentium IV Xeon 2.8Ghz+) is installed.

In addition to hardware optimization, you also need to ensure that your database is optimized. Please see the Database Tips in EventSentry help manual for more information.

7 System Health Monitoring

Even though event log monitoring can detect many problems on a server, it is not enough to ensure optimum system health and availability. For this reason EventSentry monitors the following components of the Operating System in addition to monitoring the event logs to ensure optimum system availability:

- Services
- Disk Space
- Processes
- Performance
- Software Installations

The following chapters will offer some guidance on how to setup system health packages, especially on larger networks.

All alerts generated by system health features (e.g. a stopped service, high CPU usage) are logged to the application event logs of the server being monitored. This ensures that you and fellow system administrators can review alerts after they happened.

However, you will need to monitor the application event logs and create one or more filters so that alerts are forwarded to an email or pager for example.

7.1 Service Monitoring

Service Monitoring can notify you when services (and/or drivers) change status, are removed or added to the monitored computer. Service status information can also be logged to a database so that the service status can be queried through the web reports.

When monitoring services you basically have to decide whether you want to monitor all services with the option of excluding non-critical ones, or whether you want to monitor only selected, (e.g. mission-critical) services.

Default Configuration

The default configuration of EventSentry is configured to monitor all services with the exception of approximately 20 non-critical services which are excluded. These services (e.g. WinHttpAutoProxySvc) frequently change their status from stopped to running and vice versa, and being notified of these status changes is nothing but a nuisance. The advantage of this approach is that all other services are monitored, and you don't have worry about forgetting to monitor a service that was recently added to the system.

If you are getting notifications of a service that is non-critical then you can simply add it to the list of excluded services.

Monitoring only selected services

If you are only interested in monitoring particular services, then simply set the monitor type to "Only monitor services listed" and add the services you want to monitor. If you have a large amount of computers with different services you want to monitor, then we recommend that you create multiple packages, each listing a certain set of services. You can then assign these packages as needed to your servers or groups.

7.2 Disk Space Monitoring

Setting up disk space monitoring can be difficult if you have many servers with different logical drive structures, and the recommendations from the previous chapter also apply to disk space monitoring to some degree.

Package Optimization

In order to minimize the number of diskspace-enabled system health packages we recommended that you organize the disk/logical drive partitions based on a common schema on all of your servers whenever possible. The table below shows an example:

Drive Letter(s)	Purpose	Comments (if any)
C	Operating System	
D	Paging File	This drive will always have low disk space since the space is allocated to the page file
E - G	Database Storage	This drive might be low on disk space if database space is pre-allocated
K - P	Critical Files	Stores regular files (office documents, etc.)
Q - R	Non-Critical Files	Stores non-critical files that can be deleted and disk space alerts are not needed

If you have a uniform policy such as the one listed above then it is easy to organize your health packages, and it will also help with your server administration. For the above policy, you could create the following disk space packages:

1. Disk Space - Operating System
2. Disk Server - Database Storage
3. Disk Server - Critical Files
4. Disk Server - General

Each of these packages can then have their own threshold limits and can be applied to computers and groups according to their role(s). The 4th package (General) could have settings for all other drives that cannot be categorized.

7.3 Performance Monitoring

Performance Monitoring offers to main benefits: Being alerted when a performance counter exceeds a threshold and collecting performance counters in a database. As with all system health features, we recommend that you create multiple packages when necessary.

Package Optimization

If you only collect performance data in a database for some computers, but need alerts from all computers then you can create two separate packages: One for alerts, and one for database logging.

In most cases you will want to create a general performance package that will monitor counters such as CPU usage, memory utilization, disk queue length and network usage and then set this package to be global. Create additional packages with performance monitoring objects for additional server software (e.g. IIS, SQL, Exchange) and apply them as needed.

Suggestions - Impact on Server Performance

The impact that performance monitoring has on your servers depends on the number of counters you are monitoring and how often you are reading performance data. In most cases the impact on your servers' performance will be insignificant.

Polling Interval

A small interval (e.g. 1 second, 2 seconds) will result in very accurate data but will put more stress on the monitored system than a larger interval (e.g. 5 seconds, 10 seconds). If you need accurate data then a polling interval of 5 seconds is a good compromise, 10 seconds should be enough for most other cases.

Threshold Alerts

The values entered in this section depend on the counter being monitored, but you should ensure that the time interval is not too short. If it is, then you will probably receive a lot of false positives (especially for high CPU usage).

Database

We recommend that you always check the "Log Average" checkbox when the database logging interval is higher than the polling interval (which it should be).

Let's assume that you are monitoring the CPU usage on a server and polling the data every 5 seconds and you have the database interval set to 5 minutes (even this seemingly large interval means that approximately 8640 entries will be written every month for this counter, per server). EventSentry will store all recent counter data for the entire database interval (5 x 12 = 60 values) and then write the average every 5 minutes to the database. If you don't set the "Log Average" option, then the current counter value - every 5 minutes - will be written to the database. This would obviously not create a very accurate picture of this performance counter.

7.4 Event Log Backups

There are a few things to consider when scheduling the backing up and/or clearing of event logs.

Backing up Event Logs

When backing up the event logs you might need to take extra steps when logging to a non-local drive (network share). This is because the EventSentry agents run under the security context of the "LocalSystem" by default. This built-in account has administrative privileges on the local system, but by default does not have any permissions on remote computers and network shares. As such, an event log backup to a remote network share will most likely fail if you do not take additional configuration steps.

You have two options to work around this issue:

- Run the EventSentry agent(s) under a domain user account that has administrative privileges on the servers it monitors and also has permissions to write to the network share.
- Configure the network share to allow the remote computer account (e.g. TIBET$) to have write access.

Both examples are explained further in our KB article 18.

Backing up AND Clearing Event Logs

If you configure EventSentry to backup and clear the event logs with the same schedule to a network share then you will need to take extra steps to work around a limitation of Microsoft Windows. This is because EventSentry (or Windows) will authenticate to the remote share using the credentials the "Event Log" service is running under, the "LocalSystem" account by default.

Please see our KB article 21 and the MS KB article 329974 (section MORE INFORMATION on the bottom) for more information and a solution.

7.5 Monitor IIS Web Sites

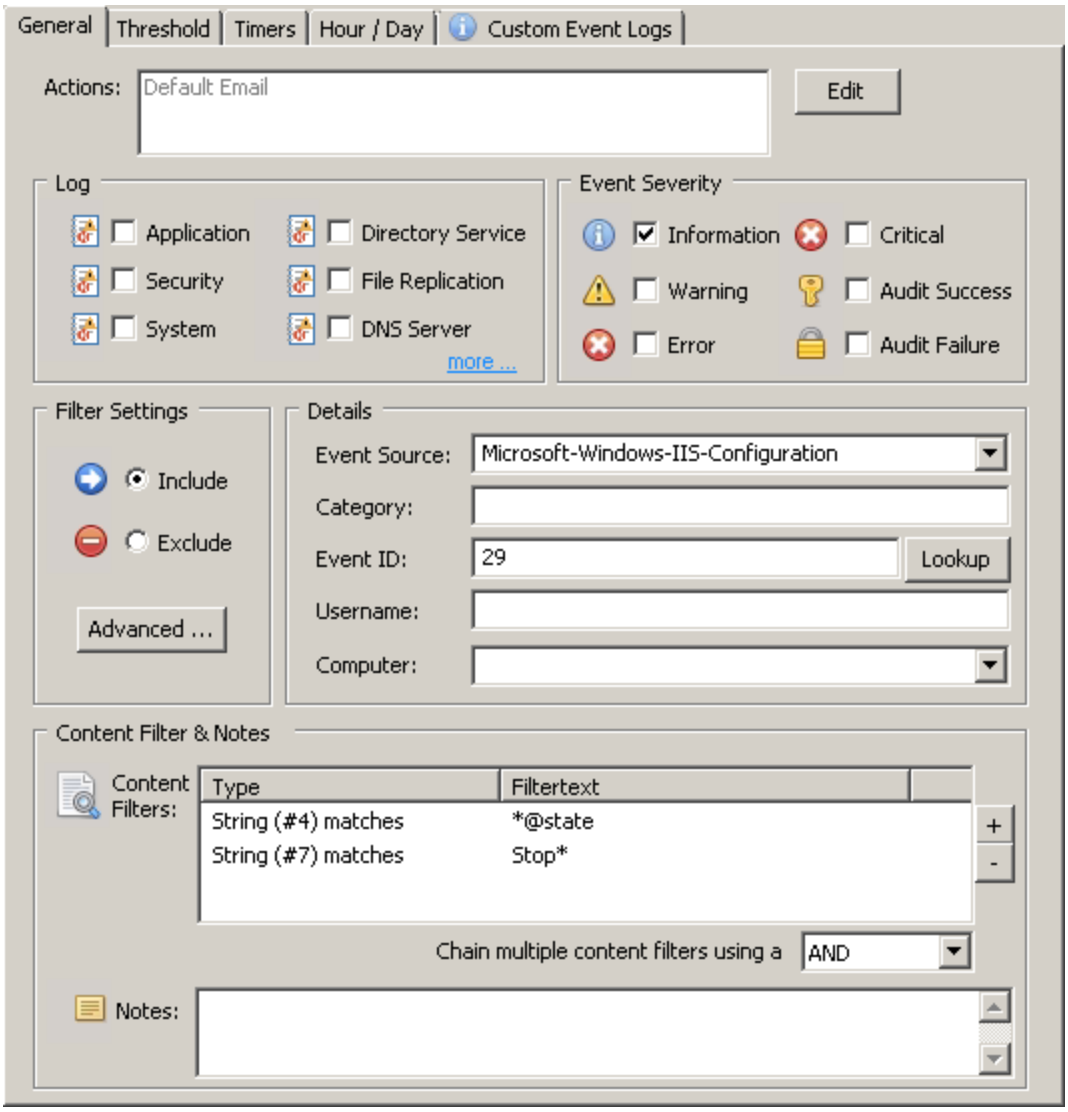
IIS Web sites can be monitored with a VBScript (Windows 2003 and earlier), or monitoring through the Microsoft-IIS-Configuration operational event log (Windows 2008 and later). EventSentry can then issue email (or other) alerts whenever an IIS site is stopped or changes its status.

Windows 2008 and later

Windows 2008 and later can log all status change of IIS sites to the Microsoft-Windows-IIS-Configuration event log. By monitoring this event log, we can detect a stopped site in real-time.

1. In the Windows event viewer, navigate to the **Microsoft-Windows-IIS-Configuration-Operational** event log
2. Right-click the log and select "Enable Log"
3. In EventSentry, create a new include filter which looks for the following event properties:

- Custom event log "Microsoft-IIS-Configuration/Operational"
- Event Severity: Information
- Event ID: 29
- Event Source: Microsoft-Windows-IIS-Configuration



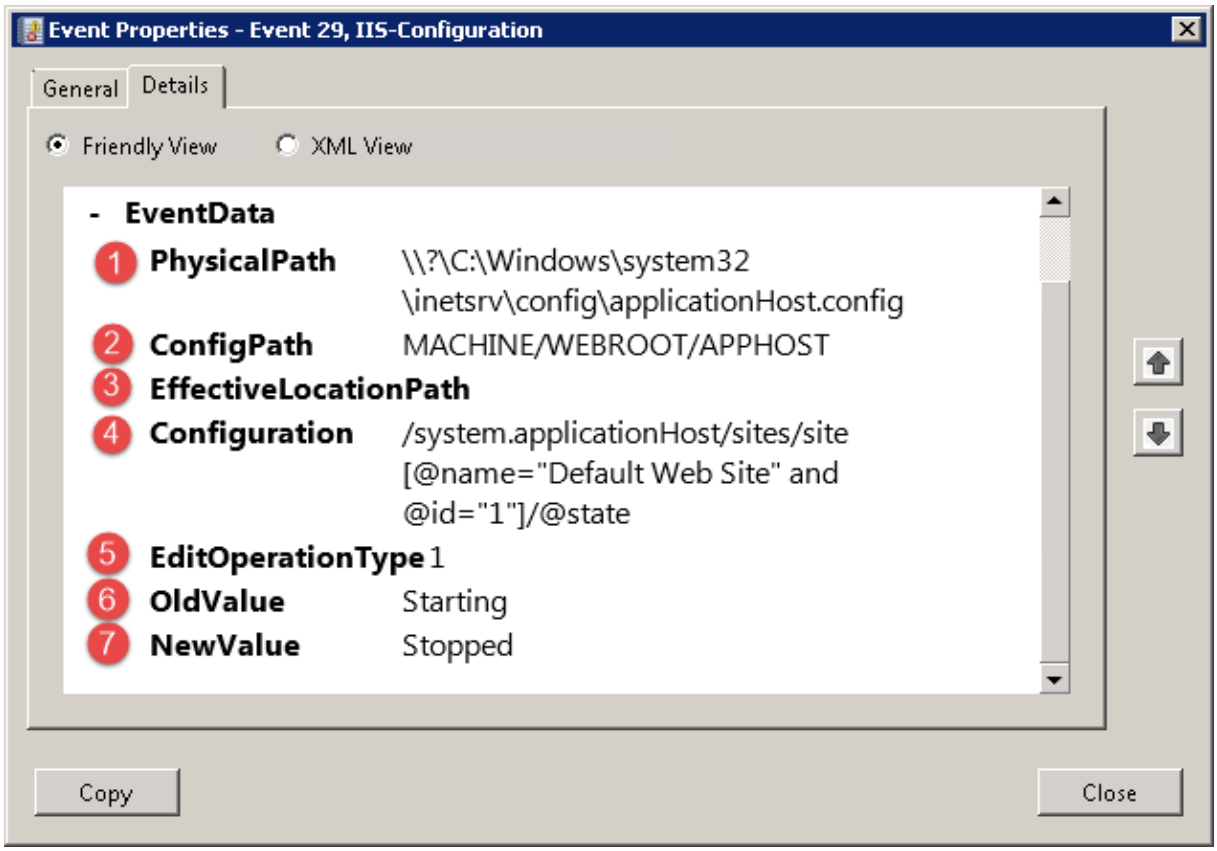
To further restrict the include filter to only alert when the site is stopped, we add a content filter (in the "Content Filter & Notes" section) with the following condition:

Insertion String #4 matches "*@state"
Insertion String #7 matches "Stopped"

The content filters should be chained using **AND**.

The email subject can optionally also be changed (when using email notifications) to something like "IIS Site Stopped" as well. The previously created filter can also be cloned by copying and pasting the filter, and replacing "Stopped" with "Started", to get a proper notification whenever a site is being started or stopped as well.

To determine which insertion string to filter against, simply open the event in question in the Windows event viewer, switch to the "Friendly" view and view the insertion strings under **EventData**. The insertion strings are parsed sequentially, with the first entry ("PhysicalPath") being insertion string #1.



Windows Server 2003 and earlier

Follow the steps below to setup IIS monitoring on Windows Server 2003 and earlier.

Create the embedded script

1. Navigate to the embedded scripts (Tools -> Embedded Scripts) dialog, so that the script to monitor IIS web sites is automatically copied to the target computers.

2. Create a new embedded script, and name it "iis_monitor_sites.vbs". Make sure that you set the "Interpreter" to **cscript.exe**, which ensures that we can capture the return code (%ERRORLEVEL%) correctly.

3. Paste the contents of the iis_list_stopped_w3svc_sites.vbs file into the "Script Content" field.

Create a system health package with the application scheduler

1. Create a new System Health package (right-click "System Health Packages"), and add the "Application Scheduler" object to it.

2. In the Application Scheduler object, make sure that **Log application return code > 0 to event log as "Error"** is checked. This ensures that the EventSentry will log an error to the application event log when the script returns an %ERRORLEVEL% that is not zero.

3. Click the plus icon and select the script we created earlier (@iis_monitor_sites.vbs) from the drop-down list. Set the schedule up as recurring and configure the desired interval (e.g. every 5 minutes).

4. Assign this package to all computers running IIS. Alternatively, you can also make this package global, and use the "Auto-Detection" feature to only activate the package on computers that have the **w3svc** service running.

Note: If the script does detect an IIS site that is not running, then EventSentry will continuously log an Error to the application event log (based on the interval setup in (3)) until the site is running again.

We recommend setting up a threshold filter, to ensure that you only get a limited number of emails with the alert.

7.5.1 iis_list_stopped_w3svc_sites.vbs

```
' Supported Platforms: Windows 2003 and earlier
'
' Lists the state of all IIS web sites configured on the local machine
' and returns an %ERRORLEVEL% of 1, if at least one web site is not in
' the "Started" state.
'
' When scheduling this script with EventSentry's application scheduler,
' make sure that the interpreter is set to "cscript.exe"

Option Explicit

Dim strServer, strServerType, strServerMetaType
Dim objService
Dim returnCode

returnCode              = 0
strServer               = "localhost"
strServerType           = "Web"
strServerMetaType       = "W3SVC"

Sub EnumServersites( objService )
      Dim objServer

    For Each objServer In objService
        If objServer.Class = "IIs" & strServerType & "Server" Then
                  If SiteIsNotRunning(objServer.ServerState) Then
                        WScript.StdOut.Write "*"
                  End If

            WScript.StdOut.Write _
                objServer.ServerComment & ": " &
State2Desc( objServer.ServerState )

                  If SiteIsNotRunning(objServer.ServerState) Then
                        WScript.StdOut.Write "*"
```

```
                        returnCode = 1
                    End If

                    WScript.StdOut.Write vbCRLF
            End If
        Next
    End Sub

    Function SiteIsNotRunning( nState )

          If nState <> 2 Then
                SiteIsNotRunning = 1
          Else
                SiteIsNotRunning = 0
          End If

    End Function

    Function State2Desc( nState )

        Select Case nState
        Case 1
                'MD_SERVER_STATE_STARTING
            State2Desc = "Starting"
        Case 2
          'MD_SERVER_STATE_STARTED
            State2Desc = "Started"
        Case 3
          'MD_SERVER_STATE_STOPPING
            State2Desc = "Stopping"
        Case 4
          'MD_SERVER_STATE_STOPPED
            State2Desc = "Stopped"
        Case 5
          'MD_SERVER_STATE_PAUSING
            State2Desc = "Pausing"
        Case 6
          'MD_SERVER_STATE_PAUSED
            State2Desc = "Paused"
        Case 7
          'MD_SERVER_STATE_CONTINUING
            State2Desc = "Continuing"
        Case Else
            State2Desc = "Unknown state"
        End Select

    End Function

    SET objService = GetObject( "IIS://" & strServer & "/" &
    strServerMetaType )
    EnumServersites objService

    If returnCode <> 0 Then
```

```
      WScript.Echo vbCRLF & "WARNING: One or more IIS sites are not
running" & vbCRLF
End If

WScript.Quit returnCode
```

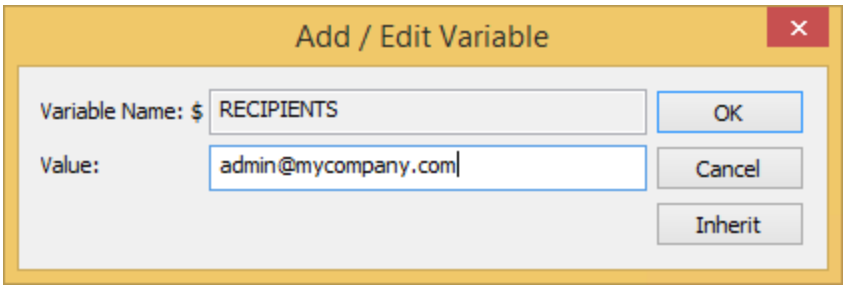
8 Actions

This chapter will help you get the most out of the built-in notifications supported by EventSentry and help you create custom notifications using the **Process** action.

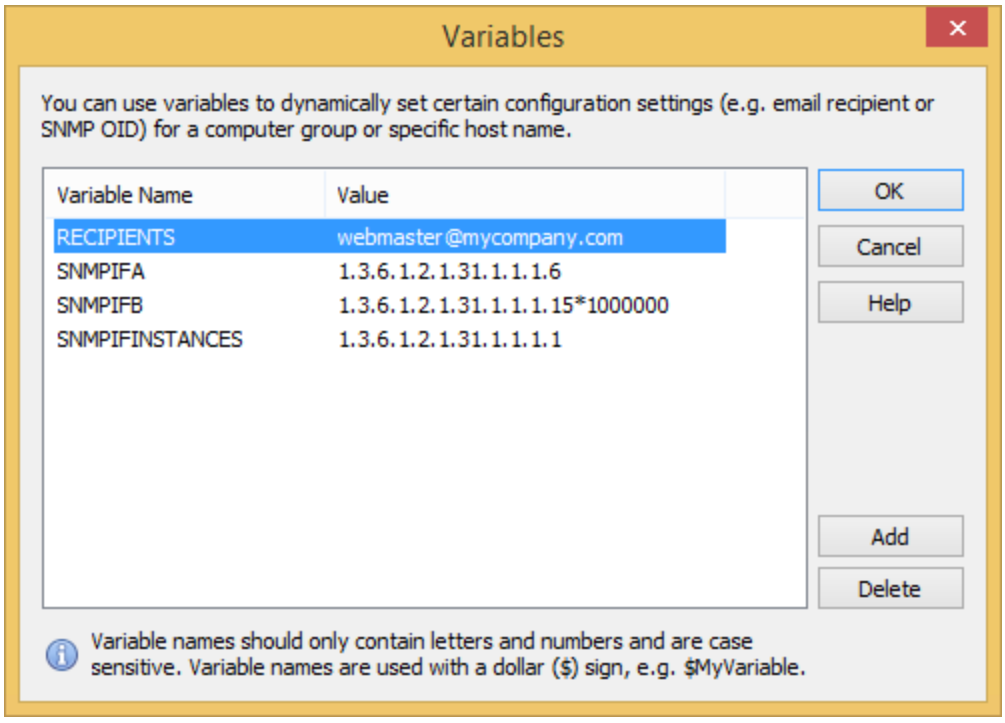
8.1 Flexible Email Notifications with variables

EventSentry supports variables that can be created globally and then overwritten on a per-group level. Since the SMTP notification supports variables in most of its input fields,you can create one SMTP target that sends email to different recipients depending on the group a server is in. This can save you from creating multiple SMTP targets with almost identical values.

1. Creating a variable
Create a new variable by navigating to Tools -> Variables -> Add. The value 'admin@mybiz.net' is only the default value for the variable, it can be overwritten for every group you have configured if you wish.



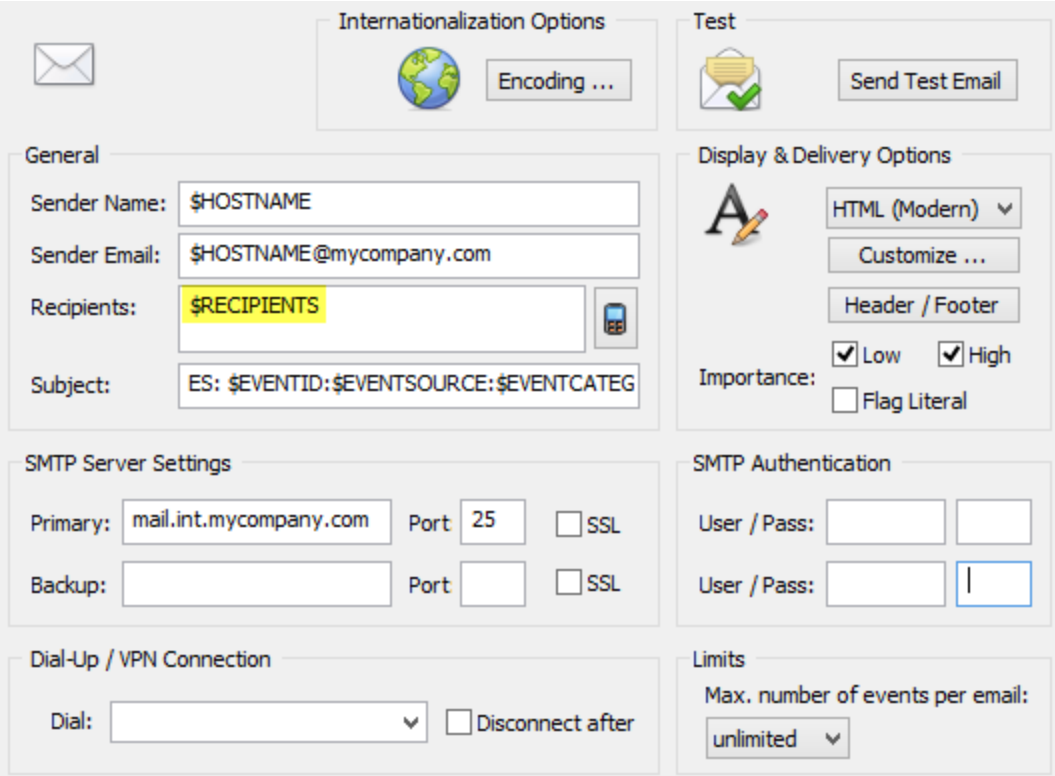
2. Setting the variable values
Right-click a group that needs a different recipient (e.g. webmaster@mybiz.com) and select "Set Variables". The resulting dialog will show you all variables with their current value for this group. If the "Inherited" column is checked then it means that the value is being inherited and has not yet been specified on a per-group level.



Double-click the variable name and specify a new value for this group. Repeat this process for every group.

3. Using the variable in the SMTP target

Now that the variables are setup you can use it for any SMTP target. Locate your SMTP target and enter $RECIPIENTS in the "Recipients" field. Now, depending on the group the computer is a member of, the email will be sent to the respective recipient of the group. If the variable has not been set for a particular group then the default value will be used.



Note the **$RECIPIENTS** variable

8.2 The Process Actions

EventSentry includes the "Process" action which allows you to forward events to custom processes, e.g.:

- Perl scripts
- Visual Basic scripts
- executables (e.g. blat.exe, etc.)

This gives you ultimate flexibility and doesn't restrict you to the notifications offered natively in EventSentry.

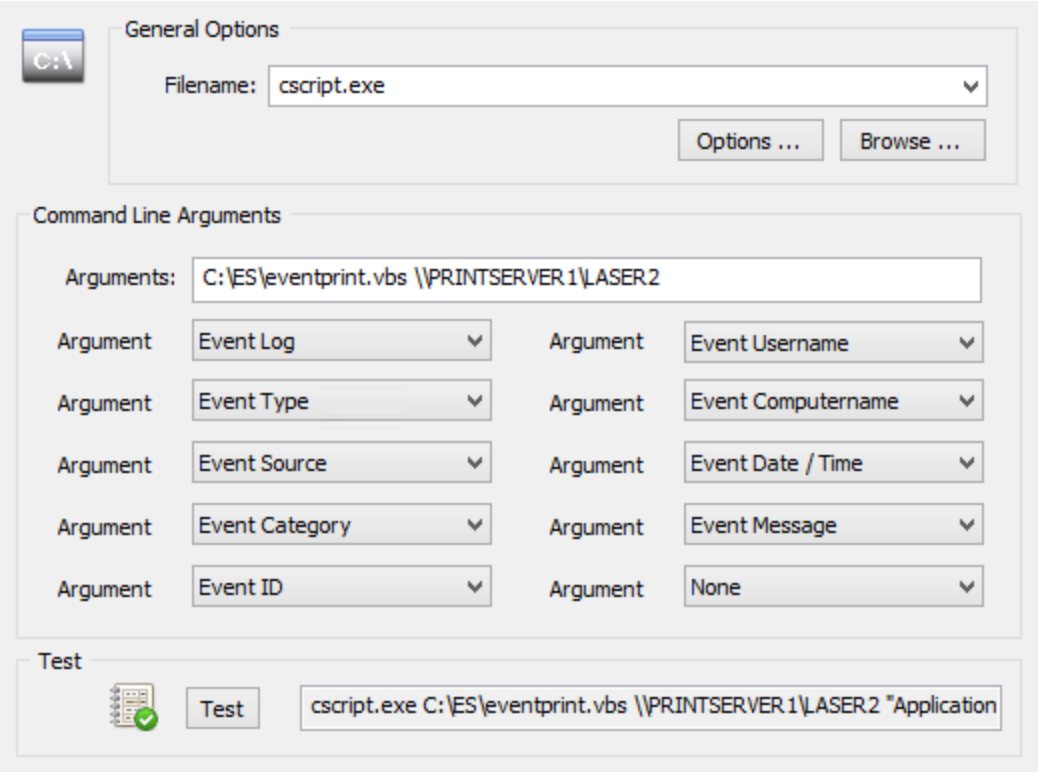
8.2.1 Sending events to a laser printer

Using a small Visual Basic script, you can send single events to any shared laser/inkjet printer on your network. Simply follow the steps below to setup a notification that will print records to a printer:

1. Installing the VBS File
Put the contents of the file eventprint.vbs (included in the sub chapter) into a text file and save it in a folder of your choice on all the computers from which you want to forward events to the printer. This step is important, the .vbs file will be executed by the EventSentry agent on every computer that is running the agent.

2. Configuring the VBS File
The VBS file uses two temporary files to send the document to the laser printer, both of which need to be configured in the VBS file if the default (C:\WINDOWS\TEMP) will not work for you.

3. Creating the process action
Add a new notification to EventSentry by right-clicking the Actions container and selecting "Add Action". Select "Process" for the notification type and configure the options similar to the options shown in the screenshot below:



You will most likely have to change the values for the **Arguments** field which points to the actual .vbs file and to the shared printer.

4. Add one or more filters

Finally add one or more filters to match the events you want to send to the printer of your choice and the setup is complete.

Please note that events sent to laser and/or inkjet printers can only be print one event per page due to the nature of the process action.

8.2.1.1 eventprint.vbs

```
Option Explicit

Const MaxCharsPerLine = 80

Dim Args

Dim EventLog, EventType, EventSource, EventCategory, EventID, EventUser,
EventComputer, EventDate, EventMessage, EventMessageFormatted

Dim fso, FileHandle
Dim TempFilePath, TempFile, PrinterPath

wscript.echo "eventprint.vbs: Prints event log records on a networked laser
printer"

' ======================================
' Define variables here:
```

```
' ========================================

TempFile    = "C:\WINDOWS\TEMP\EVENTSENTRY_PRINT.TMP"
TempFileFF  = "C:\WINDOWS\TEMP\EVENTSENTRY_FF.TMP"

' ========================================

' Make sure we have the right amount of arguments
Set args = Wscript.Arguments
If args.count < 10 Then
    wscript.echo "Not enough arguments:"
      wscript.echo "eventprint.vbs \\SERVER\PRINTSHARE EventLog EventType
EventSource EventCategory EventID EventUser EventComputer EventDate
EventMessage"
    wscript.quit(1)
End If

' Get Arguments
PrinterPath       = args(0)
EventLog          = args(1)
EventType         = args(2)
EventSource       = args(3)
EventCategory     = args(4)
EventID           = args(5)
EventUser         = args(6)
EventComputer     = args(7)
EventDate         = args(8)
EventMessage      = args(9)

' Format EventMessage
Dim EventMsgArray, Element, OneLine

EventMsgArray = Split(EventMessage, " ", -1, 1)
For Each Element In EventMsgArray

      If (Len(OneLine) + Len(Element)) > MaxCharsPerLine Then
            EventMessageFormatted = EventMessageFormatted & OneLine &
vbCRLF
            OneLine = Element & " "
      Else
            OneLine = OneLine & Element & " "
      End If
Next
EventMessageFormatted = EventMessageFormatted & OneLine

' Create temporary text file
Set fso = CreateObject("Scripting.FileSystemObject")
Set FileHandle = fso.CreateTextFile(TempFile, True)

FileHandle.Write "Event Log:      " & EventLog & vbCRLF
FileHandle.Write "Event Type:     " & EventType & vbCRLF
FileHandle.Write "Event Source:   " & EventSource & vbCRLF
FileHandle.Write "Event Category: " & EventCategory & vbCRLF
FileHandle.Write "Event ID:       " & EventID & vbCRLF
```

```
FileHandle.Write "Event User:     " & EventUser & vbCRLF
FileHandle.Write "Event Computer: " & EventComputer & vbCRLF
FileHandle.Write "Event Date:     " & EventDate & vbCRLF
FileHandle.Write "Event Message: " & vbCRLF
FileHandle.Write EventMessageFormatted & vbCRLF & Chr(12)

FileHandle.Close

' Create FF temp file, required for laser printers
Set FileHandle = fso.CreateTextFile(TempFileFF, True)
FileHandle.Write Chr(12)
FileHandle.Close

' Send files to printer
fso.CopyFile TempFile, PrinterPath
fso.CopyFile TempFileFF, PrinterPath

' Delete temp files
fso.DeleteFile(TempFile)
fso.DeleteFile(TempFileFF)

Set fso = Nothing
```

8.2.2 Emailing entries from a log file

Using two third-party tools (tail.exe and blat.exe) it is possible to automatically be emailed contents of a log file when a certain event log entry (matching one of your filters) appears.

For example, when an Audit Failure appears in the security event log that points to an authentication failure reported by IIS, then you can automatically receive an email with the most recent 25 lines of the most current IIS log file.

You will need the following free executables for this example:

1. blat.exe: http://www.blat.net/
2. tail.exe: http://unxutils.sourceforge.net/ (download UnxUtils.zip)

1. Installing the files
Copy both blat.exe and tail.exe either to the system32 directory (e.g. c:\windows\system32) or to a directory for your choice (e.g. c:\batch).

2. Configure Blat
You will need to tell blat which SMTP server it can use before you can starting using it. Run the following command:

```
blat.exe -install 127.0.0.1 youremail@domain.net
```

127.0.0.1 is the host name or IP address of your SMTP server, and youremail@domain.net is the default email address used by blat when sending an email.

3. Creating a batch file
Create a batch file with content similar to the following:

```
@ECHO OFF
```

```
for /f "Tokens=1-4 Delims=/ " %%i in ('date /t') do  set dt=%%j%%k

set FILENAME=%SYSTEMROOT%\SYSTEM32\LOGFILES\W3SVC1\EX*%dt%.log

%SYSTEMROOT%\SYSTEM32\TAIL.EXE -n 25 %FILENAME% | %SYSTEMROOT%
\SYSTEM32\BLAT.EXE - -to youremail@domain.net -subject "IIS LogFile"
```

In the above example we need to email an IIS log file which has the following format:

EXYY**MMDD**.log (YY = Year, MM = Month, DD = Day)

First we retrieve the system date and set the **dt** variable to the month and day. Then we set the FILENAME variable to the actual filename by using the **dt** variable we previously defined. The asterisk after **EX** will match any year, but this is necessary since we would need the year as a two-digit which is not supported by the **date /t** command.

Finally we pipe the output of the last 25 lines (-n 25) of the log file to blat and email ourselves the file.

4. Create a process notification target

Right-click the Notifications container, select "Add Target", specify a name and select the "Process" tab. Then point the process target to the batch file you previously created in step 3.

5. Setup a filter

Last but not least you will need to setup one or more filters that will trigger the notification you defined in step 4.

Again, you should be able to apply this example to almost any text-based log file by tweaking the batch file, but the possibilities are almost endless.