

# Table of Contents

<b>Part I Introduction</b>	<b>3</b>
<b>Part II Quick Overview</b>	<b>3</b>
<b>Part III General</b>	<b>4</b>
1 Roll-Out with MSI Package .....	4
2 Packages .....	4
Event Log Packages .....	5
Health Packages .....	5
3 Managing Agents .....	5
4 Variables .....	6
5 Secure EventSentry Setup .....	8
<b>Part IV Event Log Monitoring</b>	<b>10</b>
1 Event Log Alerts .....	11
Local Filters .....	11
Excluding Events .....	12
2 Security Alerts .....	13
Large Amount of Audit Failures (Threshold) .....	13
Wrong Password (Threshold) .....	15
File Changes .....	16
3 Recurring Events .....	19
Verifying Backups & AntiVirus Updates .....	20
One Event per Day .....	21
<b>Part V Event Log Consolidation</b>	<b>25</b>
1 Archival and Purging .....	26
2 Off-Peak Consolidation over WAN .....	27
3 Conserving disk space and optimizing performance .....	28
4 Multiple Isolated Databases .....	35
<b>Part VI System Health Monitoring</b>	<b>36</b>
1 Service Monitoring .....	37
2 Disk Space Monitoring .....	37
3 Performance Monitoring .....	38
4 Event Log Backups .....	38
5 Monitor IIS Web Sites .....	39
iis_list_stopped_w3svc_sites.vbs .....	40
<b>Part VII Actions</b>	<b>41</b>
1 Flexible Email Notifications with variables .....	41

<b>2 The Process Actions .....</b>	<b>43</b>
<b>Sending events to a laser printer .....</b>	<b>43</b>
eventprint.vbs.....	44
<b>Emailing entries from a log file .....</b>	<b>46</b>
 <b>Index</b>	 <b>0</b>

# 1 Introduction



This document 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 suggestions 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 Quickstart 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\\_help.php](http://www.eventsentry.com/support_help.php) 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. There is no central "collection" server, and the EventSentry agent needs to be running on a machine in order for it to be monitored. The agents 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, can check remote TCP ports and can also monitor the state of the EventSentry service. 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 Roll-Out with MSI Package

You can install EventSentry on all the hosts that require monitoring easily with the "Remote Update" feature of the EventSentry Management console. The only requirement is that the ADMIN\$ share and "Services" (Administrative Tools -> Services) are accessible.

In addition to using the management console however you can also roll-out the agent using our MSI package. This allows you to install the agent using methods such as the built-in Software Installation features of Active Directory or other 3rd party applications you might already have in place.

Please follow the instructions below to create a MSI package that you can use. If you do not follow the instructions below then the MSI package will **not** work.

1. From inside the Management Console, select 'File' and then 'Export'.
2. Save the file with the name 'eventsentry\_svc.reg' (the filename is important and the automatic configuration will not work if it is named differently).
3. This .REG file must be compiled into an executable program that will extract it to your C:\WINDOWS (or C:\WINNT) directory without any prompts for destination location. This can be accomplished via many different methods and compilers (only one method will be discussed here).
4. You will first need to create a ZIP file that has this REG file inside of it.
5. At this URL ( <http://www.chilkatsoft.com/ChilkatSfx.asp> ) you can download a ZIP 2 EXE program that will let you create an executable from a zip file.
6. When using this program, select your ZIP file, choose the 'Build an EXE that automatically unzips to a temporary directory' option and put your Windows directory in the 'Unzip Dir' box (this is typically C:\WINDOWS or C:\WINNT). The newly created executable will be named the same as the ZIP file but with an EXE extension.
7. When installing the Windows Installer SDK. You can minimize the components being installed by only choosing the 'Tools' option from the 'Microsoft Windows Installer SDK' package.

### 3.2 Packages

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

Filter Packages contain (event log) filters, system health packages contain health monitoring objects, and tracking packages contain tracking objects.

When creating packages, make sure that you organize them in some 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.

Please follow the links below for more information:

- [Filter Packages](#)
- [System Health Packages](#)

- Tracking Packages

### 3.2.1 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.

### 3.2.2 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.

## 3.3 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/support\\_knownproblems.php](http://www.eventsentry.com/support_knownproblems.php) 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 **2.70.4**. The last part of this number (e.g. **4**) is the build number of the version (**2.70**).

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

### Managing Agents on WAN networks

Performing remote update actions over WAN networks will take more time since EventSentry is using SMB/RPC to update files and push the latest configuration to the remote registry.

To reduce the time it takes to update remote agents you can do the following:

- Make sure that "Don't transfer package contents when package is unassigned" in your remote update preferences is activated. This will ensure that filters and health objects inside packages are not transferred with a remote update when they are not assigned.
- Reduce the number of filter, health and tracking packages in your configuration. If you have a package that you do not need, then delete the package. Note that hiding a package will still transfer it with a remote update.
- Reduce the number of filters in your filter packages. To reduce the remote update speed to an absolute minimum, delete all unneeded filters from your filter packages.

## 3.4 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:

**SMTP (Important SMTP)**

**Display Options**  
 Customize fonts and encoding  **Test**

**General**  
 Sender Name: \$HOSTNAME  
 Sender Email: \$HOSTNAME@netikus.local  
 Recipients: mr.eventlog@netikus.net  
 Subject: ES: \$EVENTID:\$EVENTSOURCE:\$EVENT

**Email Options**  
 Style: (X)HTML  
 Include Version:   
 Importance:  Low  High  Flag Literal

**SMTP Server Settings**  
 Primary: 192.168.6.77 Port: 25  
 Secondary: Port: 0

**SMTP Authentication**  
 User / Pass:    
 User / Pass:

**Dial-Up Connection**  
 Dial:   Hangup after

**Limits**  
 Max. number of events per email: unlimited  No Binary

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/eventsenry/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:

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

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/event Sentry/configvariablesdetails.htm> to see in which fields a variable is supported.



Please refer to the documentation for more information on variables.

## 3.5 Secure EventSentry Setup

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 such as Sarbanes Oxley, HIPAA or others.

Follow the steps in this chapter to ensure your EventSentry setup is as secure as possible.

### 1. Database Security

If you are 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 (MSSQL and MySQL only) or when you run the Database Setup Wizard.

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.



Do not use an administrative login (e.g. sa or root) when configuring the ODBC target 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.



Make sure that you choose a **secure password for the eventsentry\_svc** user when first setting up the EventSentry database and do not use the default password.

If you need to change the password of eventsentry\_svc user, then change it first in the database and then in the ODBC notification of EventSentry.

### 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 (MSSQL and MySQL only) or when you run the Database Setup Wizard.

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 **WebReportsConfig.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 **WebReportsConfig.xml** file on the web server.

### Encryption

If the EventSentry agents are transmitting event log data over an insecure medium, then we recommend that you use a Microsoft SQL Server database (2000 or 2005) that allows you to encrypt SQL communication between the client (any EventSentry agent) and the database server. See [Encrypting Network Traffic with MSSQL](#) for more information.

## **2. EventSentry 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

2a. Windows 2000: Open the registry editor **regedt32.exe** and select the key **HKLM\Software\netikus.net\EventSentry**. Then, select **Security -> Permissions** from the menu and add the newly created user account to the list with **Full** permissions.

2b. Windows 2003, XP: 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

7a. Windows 2000: Open the registry editor **regedt32.exe** and select the key **HKLM\Software\Microsoft\Windows NT\CurrentVersion\Perflib**. Then, select **Security -> Permissions** from the menu and add the newly created user account to the list with **Read** permissions.

7b. Windows 2003, XP: 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 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 be 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.

## 4.1 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 losing out on being notified when a serious and unexpected error occurs.

### 4.1.1 Local Filters

With the introduction of Filter Packages in version 2.70 of EventSentry, Local Filters are no longer supported. Instead, you can use one or more filter packages to emulate the behavior of Local Filters.

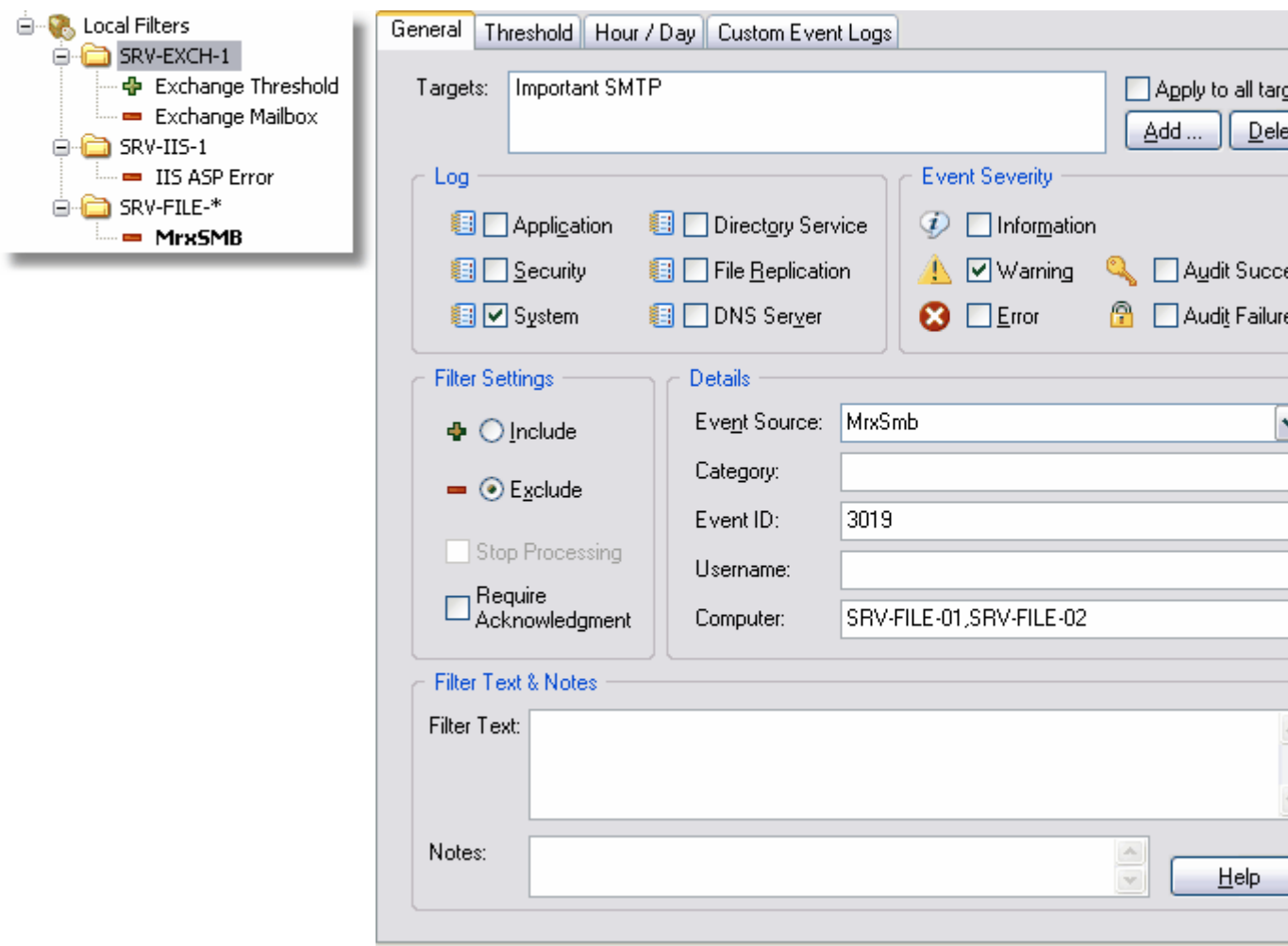
#### One Package per Computer

If you have multiple computers that each require a unique set of filters then you can create a filter package for each of those computers and assign the respective filter package to the computer in question. Since the filters in that package will only be processed by the computer the package is assigned to, this configuration is similar to the Local Filter feature found in earlier versions of EventSentry. Yet, it gives you the ability to centrally manage the configuration.

#### One Package for all Computers

Instead of creating one package for each computer (which might not be ideal if you have a large number of computers), you can also create one filter package (e.g. named "Local Filters") and add multiple filters to this one package.

To make sure that a filter inside the package only applies to one computer, specify the computer name in the "Computer" field of the filter. This ensures that the filter will only be processed by the computer specified in this field. The screenshots below show this better:



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 will only be processed on those computers where the computer name matches the "Computer" field in the filter.

#### 4.1.2 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.

## 4.2 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 than 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

### 4.2.1 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.

The screenshot shows the 'Threshold' configuration window for EventSentry. The 'Hour / Day' tab is active. The 'Enable Threshold' checkbox is checked. The 'Threshold Interval' is set to a limit of 100 in 1 hour(s). Under 'Event Processing', all three checkboxes are unchecked. Under 'Event Logging', 'Log when threshold is met' is checked, and 'Log as' is set to 'Error'. Under 'Threshold Options', 'Filter (every event processed by this filter)' is selected, and 'Text (Details)' is checked under 'Match events based on:'. A 'Help' button is at the bottom right.

#### 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.



Since the threshold event is not forwarding events to the notification, you will need to review the event logs or the database in order to review which audit failures have been logged to the event log.

#### 4.2.2 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.

The screenshot shows a configuration window for Event Log Monitoring with the following sections:

- Enable Threshold:**  Enable Threshold
- Threshold Interval:** Limit  in
- Event Processing:**
  - Forward events before threshold is reached
  - Forward events when/after threshold has been met
  - Forward first event only
- Event Logging:**
  - Log when threshold is met
  - Log when threshold is met/exceeded and interval is elapsed
  - Log as:
- Threshold Options:**

Match events based on:

  - Event (every event that shares the same properties below)
    - Log  Severity  Source  Category
    - ID  Username  Text (Details)
  - Filter (every event processed by this filter)

Help

*Thresholds help you limit the amount of events that are processed by a notification, or detect whether a certain event (or group of events) occurs a specified number of times during a set time interval.*

#### 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.



We suggest that you investigate the event logs immediately and take corrective action (e.g. temporarily disable the user) when EventSentry notifies you of many failed login attempts. Using database consolidation will help you correlate event log entries among multiple machines.

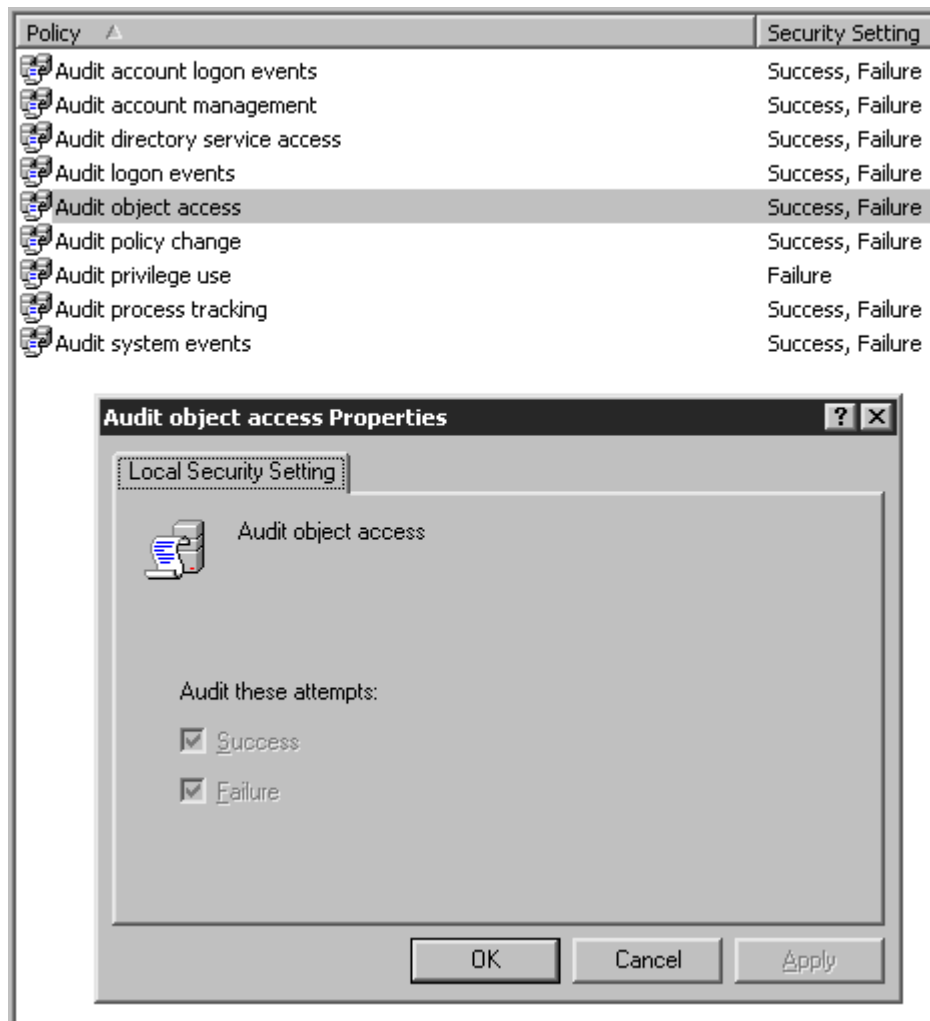
### 4.2.3 File Changes

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

In the following example we will configure the OS and EventSentry to notify us when an EXE file is either changed or added to the %SYSTEMROOT%\System32 directory.

#### 1. Enable Object Auditing

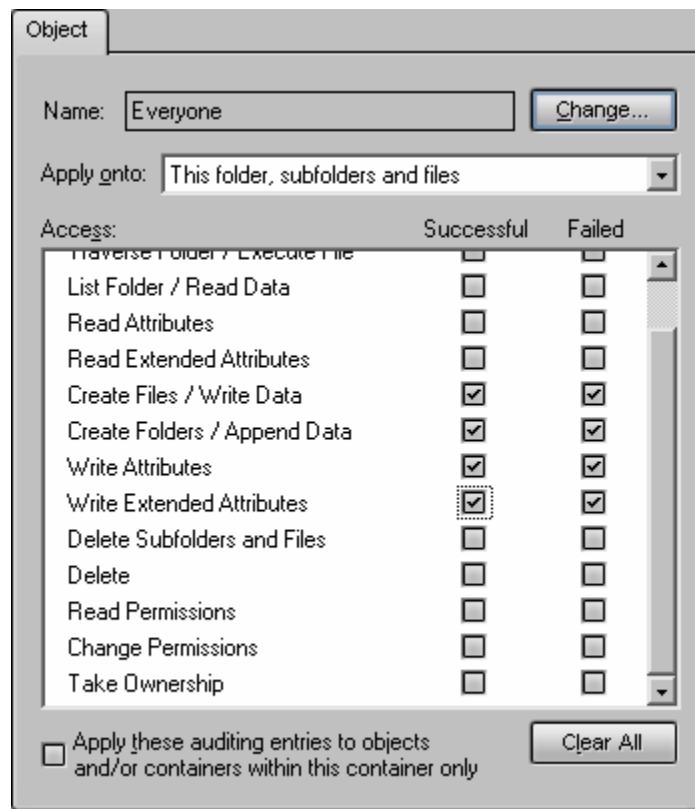
Before we can enable auditing on a folder, we need to enable "Audit object access" in the group policy of your domain or server. You can find this auditing object in the "Local Policies -> Audit Policy" container. Make sure that at least "Success" is selected:



## 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 (%SYSTEMROOT%\System32 in our case), right-click the folder and select "Properties".

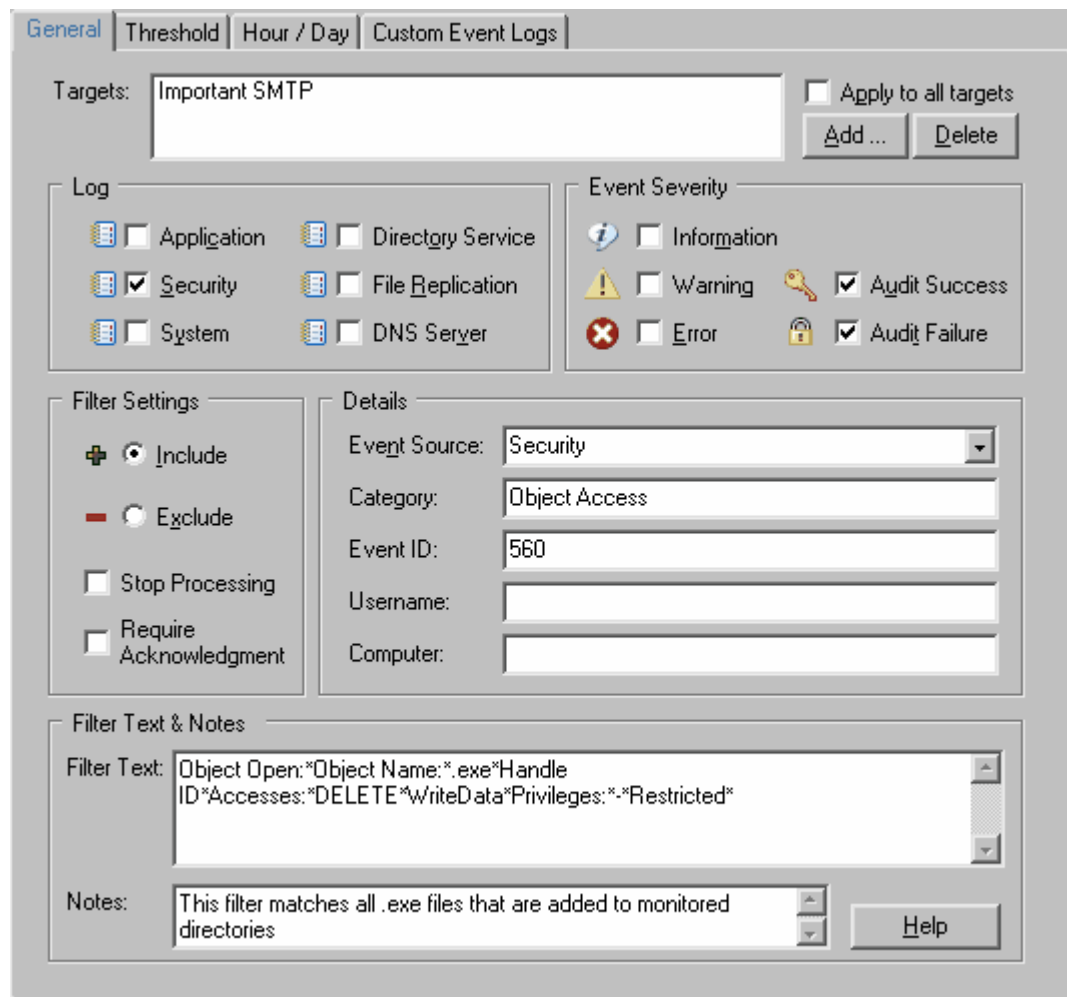
On the "Security" tab, click the "Advanced" button to get to the "Advanced Security Settings" for the folder. There, click the "Auditing" tab and select "Add". Now specify an account you would like to audit (we recommend "Everyone") and select the following types of Access shown in the screenshot below:



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

### 3. Creating an Include Filter

Now that the OS will log write access to the %SYSTEMROOT%\System32 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:



Don't forget to assign this package to a group or computer in order for the filter to become effective.

## 4.3 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.

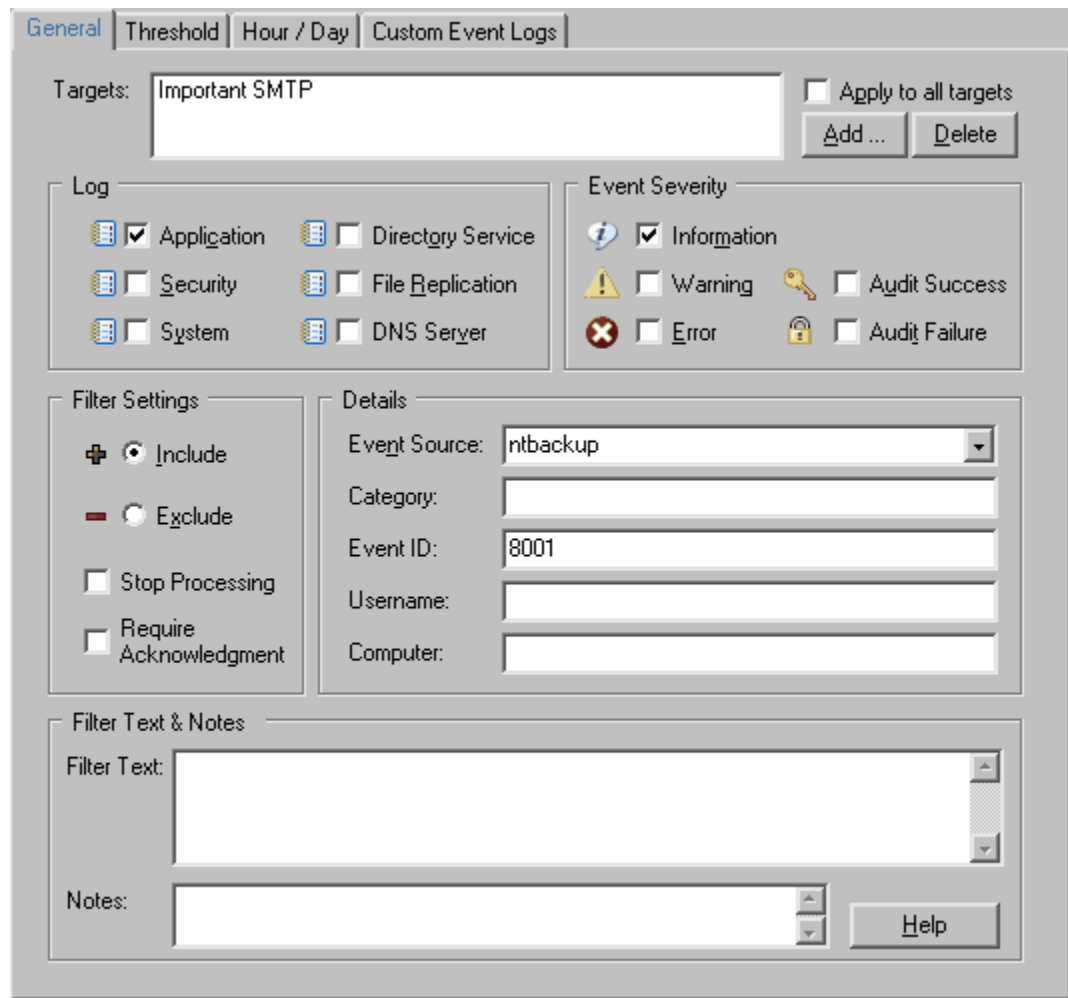
For more information on recurring events and more in-depth example please see the Backup with EventSentry and NTBackup guide, available from the Guides section at <http://www.netikus.net>.

### 4.3.1 Verifying Backups & AntiVirus Updates

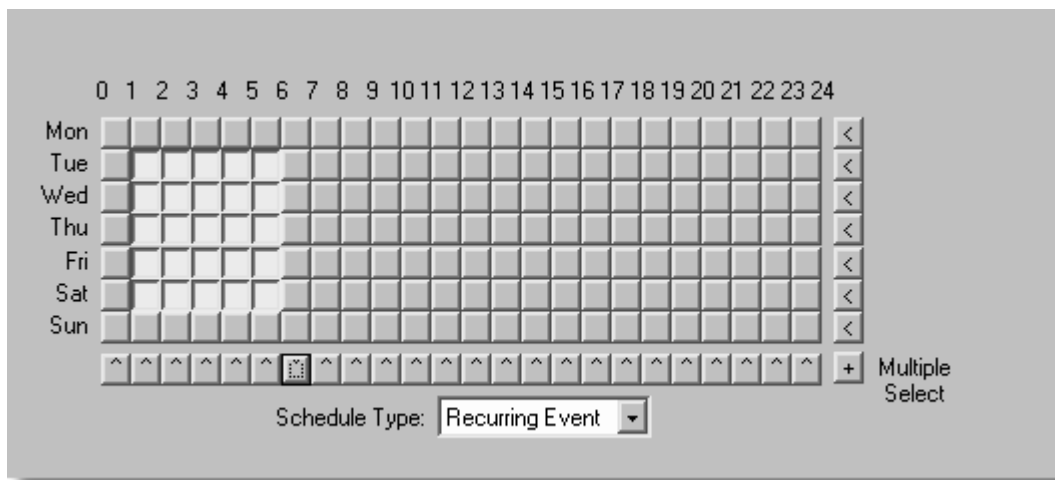
In this example we will setup a filter that will notify us when a backup event is not written to the event log between 1AM and 6AM in the morning from Tuesday to Saturday. NTBackup logs an informational event to the application event log with the event id of 8001 and an event source of "NTBackup" (NTBackup actually logs this event for every drive or object that was successfully backed up - usually resulting in more than just one of those events being written, but for the sake of simplicity we will assume that we are only interested in one event).

#### 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 screenshot 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.

### 4.3.2 One Event per Day

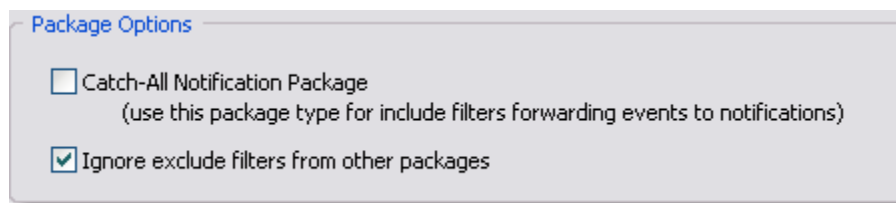
One problem with the way the recurring events feature currently works is that you cannot configure it to look for one or more events during a 24-hour time period - on a daily basis. This is because if you select/push all buttons, the recurring event filter doesn't know when your time period starts and ends, since it is the change from a raised to a pushed (and vice versa) that indicates this.

In order to be notified that one or more events do not occur at least once a day you will need to create two recurring event filters, one threshold filter and one exclude filter. You will also need to create a new package for the filters and configure the package to **Ignore exclude filters from other packages**.

In this example will make sure that the process **notepad.exe** is executed at least once a day.

#### Creating a filter package

Create a filter package by right-clicking the **Filter Packages** container. Once you have entered a name for the package, right-click the package and select **Edit**. Configure the package to ignore exclude filters from other packages.



#### 1st Recurring Event Filter

The first recurring event filter will look for your event between 12AM to 12PM and write an **10620** error to the event log if the event does not occur. We call this filter "1st Time Period".

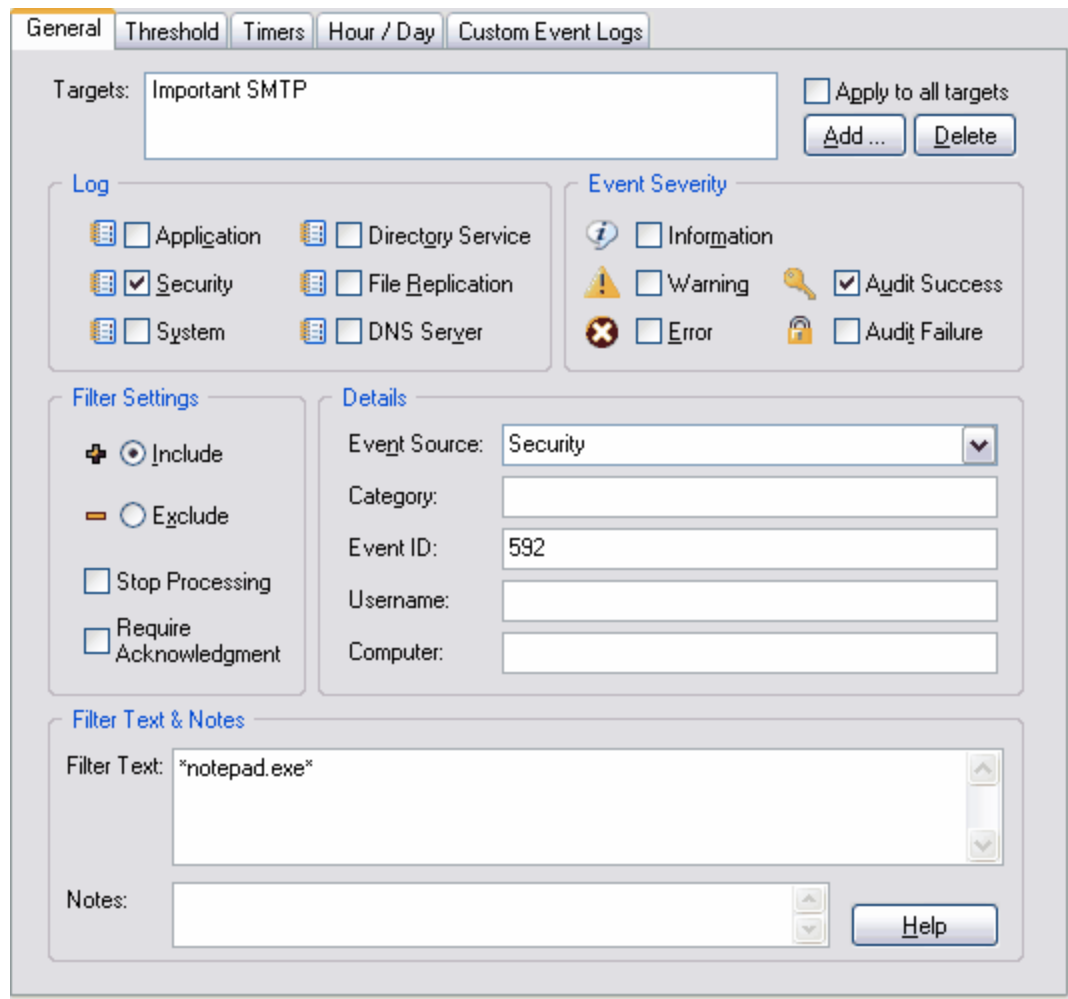
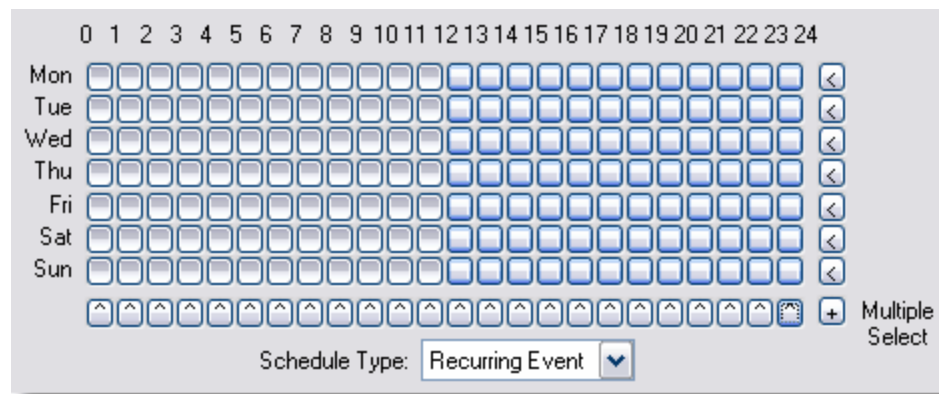


Figure 1: The general settings of our 1st recurring filter

Figure 2: The **Hour/Day** settings, covering midnight to 12PM

### 2nd Recurring Event Filter

The second recurring event filter will look for your event between 12PM to 12AM and write an **10620** error to the event log if the event does not occur. The **General** tab of this filter has to look identical to that of your first recurring event filter, figure 1 in our example.

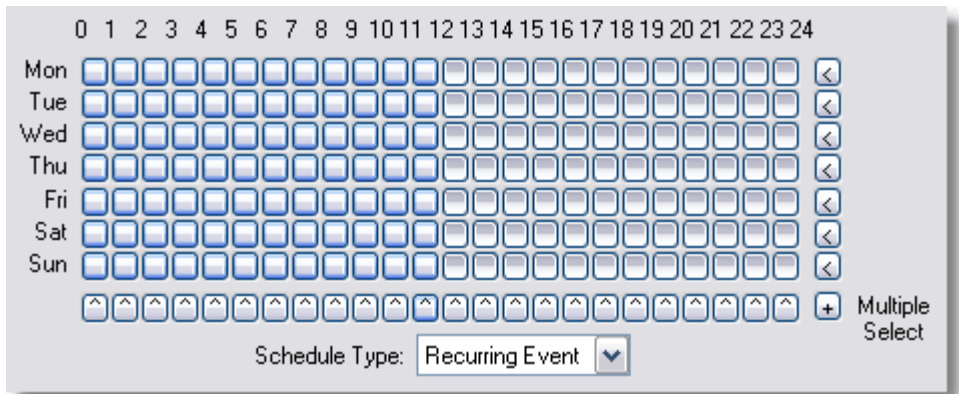


Figure 3: The **Hour/Day** settings, covering 12PM to midnight

**Threshold Filter**

It is of course OK to not have the event appear during one of the two time periods, but we need to be notified if we receive two **10620** events from the previous filters on any given day. As such, we will set the threshold filter to log an error when both recurring events logged an error within 13 hours.

The **General** tab of this filter would be configured to match the recurring event filters (note the **Filter Text**) so that it won't interfere with other recurring event filters, and the **Threshold** tab would alert us if we see more than one of these recurring events in 13 hours. The threshold filter will log an event with id **10601** to the event log when this happens.

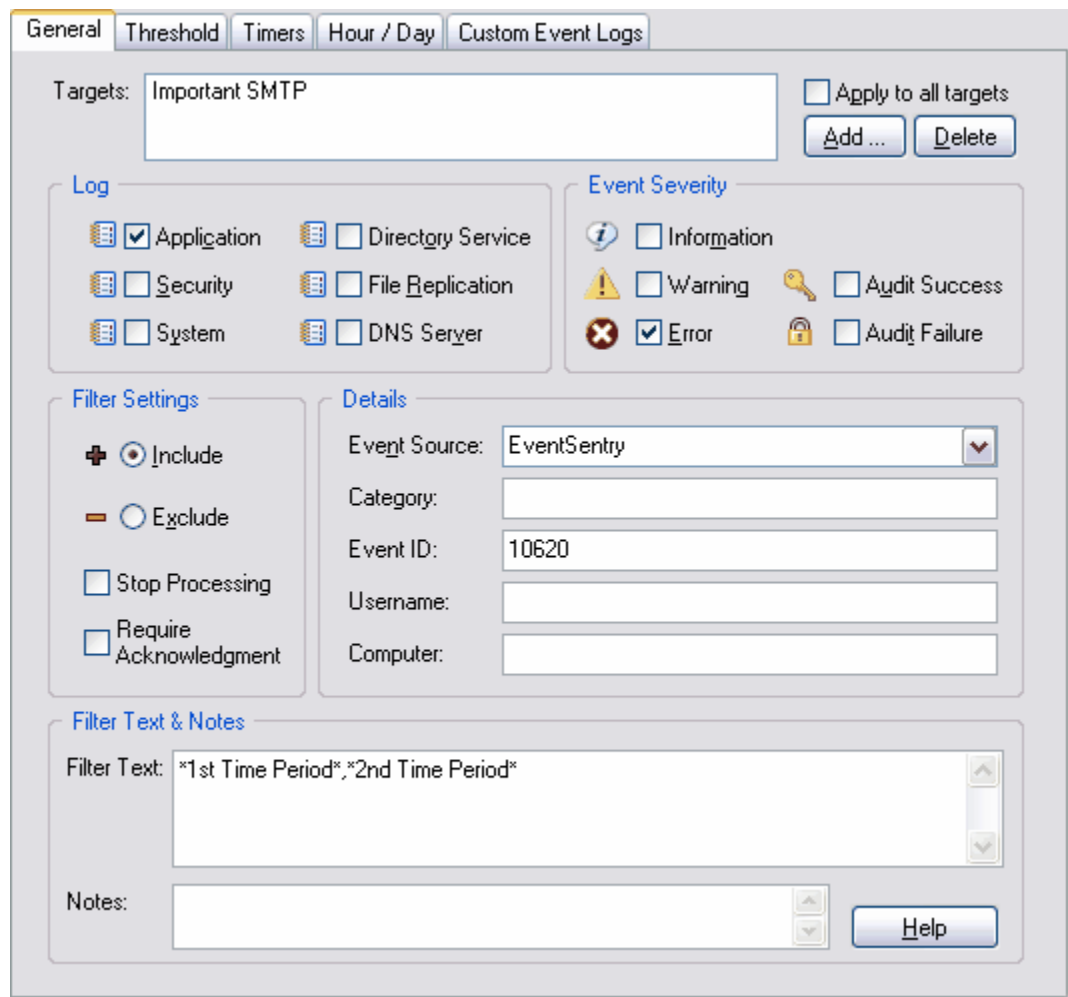


Figure 4: Match only recurring events used for this feature

The screenshot shows the 'Threshold' tab of a configuration window. It includes sections for 'Enable Threshold', 'Threshold Interval' (set to 2 in 13 hours), 'Event Processing' (with options for forwarding events), 'Event Logging' (with 'Log when threshold is met' checked and 'Log as' set to 'Error'), and 'Threshold Options' (with 'Filter' selected). A help button is located at the bottom right.

General Threshold Timers Hour / Day Custom Event Logs

Enable Threshold

Threshold Interval

Limit 2 in 13 hour(s)

Event Processing

Forward events before threshold is reached

Forward events when/after threshold has been met

Forward first event only

Event Logging

Log when threshold is met

Log when threshold is met/exceeded and interval is elapsed

Log as: Error

Threshold Options

Match events based on:

Event (every event that shares the same properties below)

Log  Severity  Source  Category

ID  Username  Text (Details)

Filter (every event processed by this filter)

Help

Thresholds help you limit the amount of events that are processed by a notification, or detect whether a certain event (or group of events) occurs a specified number of times during a set time interval.

Figure 5: Log an error if we see more than one recurring event error

### Exclude Filter

The exclude filter is necessary (or recommended) if you use the default filter setup which includes catch-all filters that forward errors to your email. Since recurring event filters always log events as **errors**, you would be notified as soon as the first (or second) recurring event filter doesn't find the event, which would not be very helpful since you would get this alert every day. As such, you will exclude events generated by the two recurring event filters, and instead receive alerts from the threshold filter.



It is imperative that you place this exclude filter in a package **other** than the one you just created. If you fail to do this then the threshold filter will never match since it matches the same events the exclude filter excludes.

Simply place the exclude filter in a package that already excludes other events for your email notification. This is why we configured the package to ignore exclude filters from other packages earlier.

## 5 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 a database (Microsoft SQL Server, MySQL, Oracle and Microsoft Access are supported through ODBC).

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 are using Microsoft SQL Server or Microsoft Access. Please see Steps to Event Log Consolidation for more information.

- Create a database
- Create an ODBC connection
- Run the Database Setup Wizard to create all necessary users, tables and indexes
- Create database notification target that points to the database (use a connection string whenever possible)
- Create an event log filter that forwards some or all events to the database

If you have a Microsoft SQL Server database available then it is highly recommended that select the "Setup MSSQL" option during the installation, which will create and initialize the database and setup a basic configuration in EventSentry.

#### **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**

Microsoft Windows 2000 and higher only ship with the "SQL Server" and "Microsoft Access" ODBC drivers by default, meaning that you will have to install an ODBC driver on the monitored servers if you are not using Microsoft SQL Server.

##### Microsoft SQL Server and Microsoft Access

Both drivers are installed by default on Windows 2000 and higher.

##### MySQL

It is fairly easy to install the MySQL ODBC driver on a monitored server since MySQL offers a setup routine that installs the MySQL ODBC driver. 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.

## **5.1 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.

## 5.2 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.

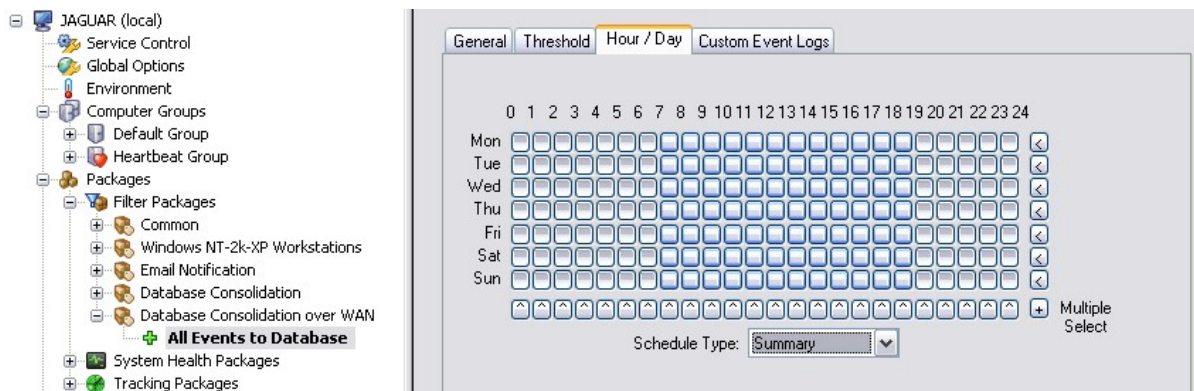
### 2. Create a new filter package

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.

## 5.3 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 losing 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 of 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.

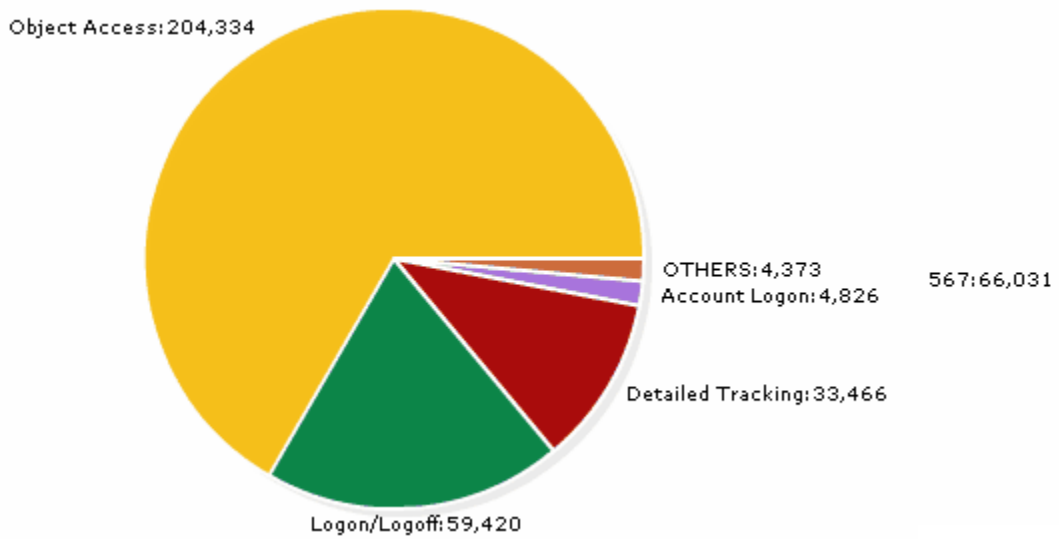
To find out which events are using up most of the space, you can use the **Group By** feature of the event log search in the web reports. The **Group By** feature lets you find the topmost events by a variety of criteria, for example by event id or event category.

#### 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 Search" page (under EVENT LOG). Then, click **Category** in the **Group By** field, select the **Show Chart** checkbox and click search. You might also want to further restrict the search by only showing records from the last 24 hours for example.

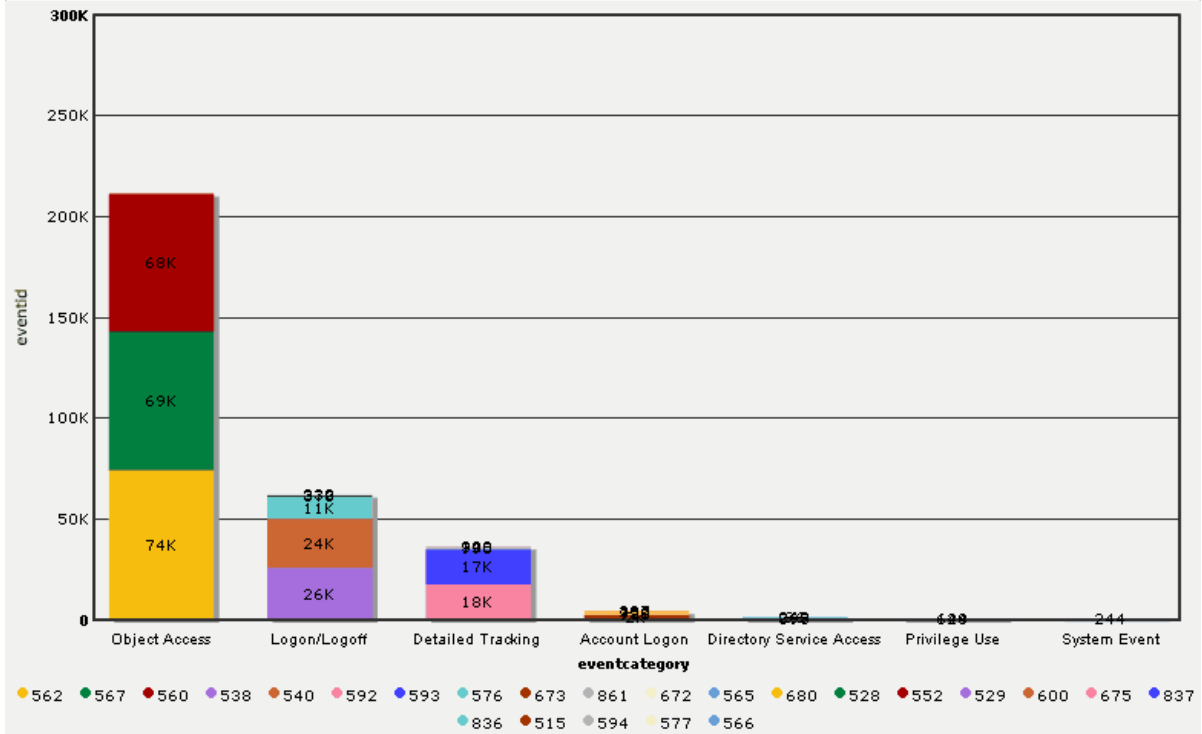
The left example below shows that 67% (move over the piechart with the mouse to see the percentage) of all events written in the last 24 hours are from the **Object Access** category, 19% are from the **Logon/Logoff** category, and 11% are from the **Detailed Tracking** category. Put together, these three categories account for 97% of all events being logged to the database, out of a total of about 35 event categories.

If you want to know which event IDs from the **Object Access** category are being logged, then select "Object Access" from the category pull-down menu, and click **ID** in the **Group By** list box and repeat the search. The right example below shows the results of this query.



However, you can also group the output by two fields at a time, for example you can select both the **category** and **ID** field from the **Group By** list box. The output below shows the results of this query (only the top 10 rows are shown in the screenshot):

Total	eventcategory	eventid
74210	Object Access	562
68736	Object Access	567
68363	Object Access	560
25725	Logon/Logoff	538
24300	Logon/Logoff	540
17533	Detailed Tracking	592
17375	Detailed Tracking	593
10907	Logon/Logoff	576
2434	Account Logon	673
995	Detailed Tracking	861



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 you 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 messages 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 use the **Log Average** feature

**Performance Monitoring Details**

Name: CPU  
Enter a descriptive and unique name here

Counter: Processor(\*)\% Processor Time Browse ...

Exclusions:   
You can exclude instances when monitoring all instances of an object. Separate multiple instances with a comma.

**Polling Interval**  
Check counter every 5 second(s)

**Threshold Alert**  
 Enable Alert Log to Event Log as: Error  
 Alert if value is more than 80 for 8 minute(s)  
 Notify at most once every 1 hour(s)

**Database**  
 Log to ODBC target  Log Average  
 mssql every 10 minute(s)

OK Cancel Help

and set the database logging interval significantly higher than the counter collection interval.

The screenshot on the right shows a correctly configured database configuration for the **Processor (\*)% Processor Time** performance counter, which records the current CPU usage in percent.

While the CPU performance counter is queried every 5 seconds (Polling Interval), the counter information is only written to the database every 10 minutes.

Of course, just writing the current CPU usage to the database every 10 minutes would not provide a very accurate picture of a server's performance, but activating the **Log Average** feature will log the average across the database logging interval

(10 minutes) to the database.

In this example, EventSentry accumulates 120 data points during the 10-minute period (600 seconds / 5 seconds), and will then **calculate & log the average** after 10 minutes have elapsed.

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

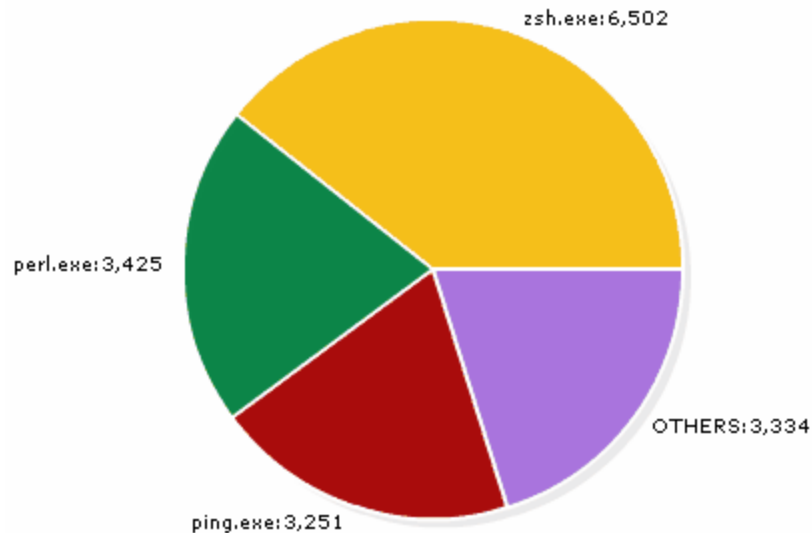
Process Tracking data is similar to event log data, though it too uses less space in the database than event log data since no event messages are being recorded. Process Tracking can give you an enormous amount of information about the state of processes on a given server or workstation at any given 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 "Process Tracking" page (under TRACKING). Then, click **Filename** in the **Group By** field, select the **Show Chart** checkbox and click search. 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 **zsh.exe**, **perl.exe**



and **ping.exe** processes are being executed most often, accounting for about 80% of the process log entries by adding approximately 13,000 rows every day.

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

### 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.

## 5.4 Multiple Isolated Databases

It is possible to configure EventSentry to consolidate information into separate MSSQL databases (e.g. location A writes to database A, location B writes to database B) without giving users/administrator from one site access to any other sites. By following the steps below you can achieve this setup without having to setup multiple ODBC DSNs or virtual directories on IIS.

This scenario is mostly useful for larger networks where information between the various sites needs to be isolated and not accessible by other sites. The instructions below will assume that IIS and Microsoft SQL Server are setup on the same physical machine.

1. On the machine where IIS is installed, run the EventSentry setup with the "Setup IIS" option or setup the EventSentry web reports manually (click for details).
2. Open the IIS manager and right click on the "EventSentry" virtual directory and select Properties. Click on the "Directory Security" tab, then click the Edit button under "Authentication and Access Control". Make sure that "Enable anonymous access" is **not** checked and "Integrated Windows authentication" and "Basic authentication" are checked.



If IIS is not installed on the same machine as your database server, then you will have to **clear the Integrated Windows Authentication** so that only **Basic Authentication** is checked. It is highly recommended in this case that you enable SSL on this server so that username and password are not transmitted in clear text.

3. From the Start menu run the EventSentry Database Setup Wizard and make sure that "Create SQL File Only" is checked on step 2. Complete the wizard and save the file as **evententry\_table\_setup.sql**.
4. Open the **evententry\_table\_setup.sql** file with a text editor (e.g. notepad) and replace the strings

```
with password = 'svcpwd', check_policy = off
with
FROM WINDOWS
```

#### Repetitive Steps

Steps 5 - 9 need to be repeated for every additional database/site that is being created.

5. Create a new Windows user account (e.g. *EventSentry\_Admin\_Site1*) either on the database server or in your Active Directory, this user account will be used to access the EventSentry information through the web reports.
6. Create an empty database (e.g. *EventSentry\_Site1*).
7. Using notepad or any other text editor, open the SQL file which was created with step 3 and replace

all instances of **evententry\_web** with the name of the user account created in step 4, including the domain information. For example, replace **evententry\_web** with **DOMAIN \EventSentry\_Admin\_Site1**.

8. Load and execute the SQL File using either Query Analyzer or the SQL Server Management Studio.
9. Open the web reports in a browser (e.g. <http://server1/EventSentry>) and navigate to "MAINTENANCE - Profile Editor". Edit the default profile or add a new profile if this is not the first database you are setting up.

Edit the ODBC DSN / Connection String: section by entering a connection string similar to the one shown below:

```
driver={SQL Server};server=DBSERVER;Network=DBMSSOCN;database=EventSentry_Site1
```

where "DBSERVER" is the host name of your database server and "EventSentry\_Site1" the name of your first database. It is important that you omit the username and password information from the connection string, so that users will be required to authenticate themselves when they try to view a profile.

Click the **Apply** button to save the profile and verify that you can access the database with the username you created in step 4 (e.g. EventSentry\_Admin\_Site1).



Instead of setting up subsequent profiles with the profile editor, you can also copy and paste the profile sections from the WebReportsConfig.xml file and simply adjust the NAME, TITLE and ODBCDSN sections in the XML file.

### Conclusion

Once you have at least two database setup, you should be able to switch between them simply by switching the profile name on the top right. You should be required to login with the username created in step 5 (e.g. EventSentry\_Admin\_Site1). This setup is possible because the login information is passed from IIS to the database server, and access to a particular database is only granted if the username is valid for the requested database.

## 6 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.

## 6.1 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.

## 6.2 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 disk-space-enabled system health packages we recommended that you organize the disk/logical drive partitions according to a scheme on all of your servers. 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.

## 6.3 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.

## 6.4 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.

## 6.5 Monitor IIS Web Sites

While EventSentry does not natively monitor individual web sites inside IIS, you can easily achieve this functionality using a visual basic script in conjunction with the Application Scheduler feature. This will allow you receive an email anytime an IIS web site is not running.

Follow the steps below to setup IIS monitoring.

### 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 **cscrip.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.

### 6.5.1 iis\_list\_stopped\_w3svc\_sites.vbs

```
' 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

```

## 7 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.

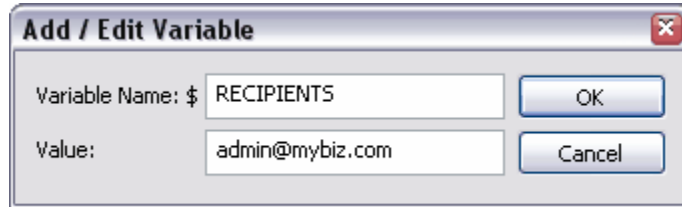
### 7.1 Flexible Email Notifications with variables

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. Then, 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.

The screenshot shows the configuration interface for an action in EventSentry. It is divided into several sections:

- HTML Font Options:** Font: Verdana, Size: 11px. Includes a 'Test' button.
- General:** Sender Name: \$HOSTNAME, Sender Email: \$HOSTNAME@mybiz.com, Recipients: \$RECIPIENTS, Subject: ES: \$EVENTID:\$EVENTSOURCE:\$EVENT.
- Email Options:** Style: (X)HTML, Include Version: checked, Importance: Low and High checked, Flag Literal: unchecked.
- SMTP Server Settings:** Primary: 122.128.2.73, Port: 25; Secondary: (empty), Port: 0.
- SMTP Authentication:** Two fields for User / Pass.
- Dial-Up Connection:** Dial: (dropdown), Hangup after: unchecked.
- Limits:** Max. number of events per email: unlimited, No Binary: unchecked.

Note the **\$RECIPIENTS** variable

## 7.2 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.

### 7.2.1 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

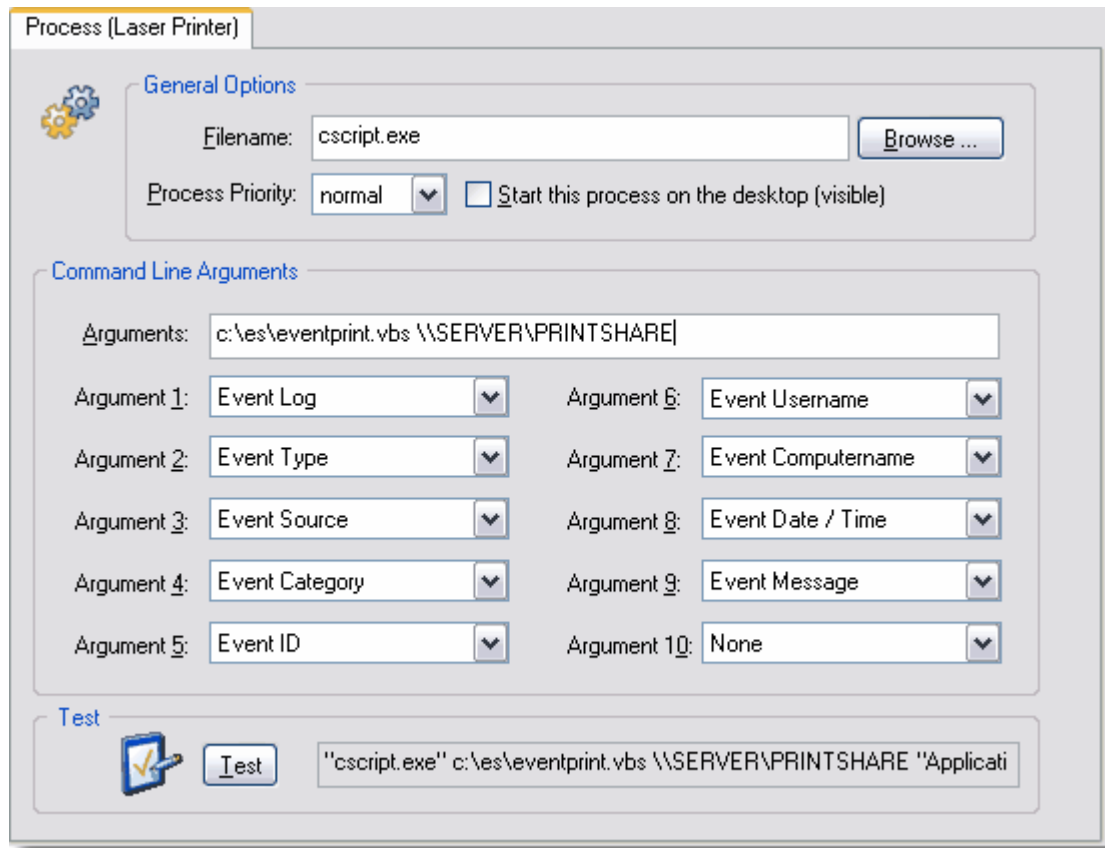
Put the contents of the file eventprint.vbs (included in the sub chapter) into a text file and save it in a folder of your choice on all the computers from which you want to forward events to the printer. This step is important, the .vbs file will be executed by the EventSentry agent on every computer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

#### 7.2.1.1 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

```

## 7.2.2 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.blatt.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:

```
EXYYMMDD.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.