ANDUTTEYE SOFTWARE SUITE

WWW.ANDUTTEYE.COM

Documentation

Version:1.143

Table of Contents

1. Preface

The documentation is on a “as is basis” without any warranties of any kind.

AES Andutteye Software Suite is a module based software that are targeting different tasks of systems administration. With an implemented Andutteye Software Suite environment one can monitor and manage your complete IT infrastructure from a central server through the central web interface. The different modules can also be used in standalone mode and is then administrated from the command line.

Since AES Andutteye Software Suite are divided in modules not all software needs to be installed. If one only needs monitoring capabilities only monitoring software needs to be installed.

This documentation will help you to install and configure a complete AES Andutteye Software Suite system. It will also give you information on various functionality, arguments and configuration settings that one can use.

1. AES Andutteye Software Suite

As one will notice in this documentation and in the software AES is sometimes used which is a shorter name for Andutteye Software Suite. Sometimes Andutteye Software Suite is much to long to say.

2. AES Andutteye Software Suite Modules

The different modules currently present in AES is

Monitoring	System monitoring of hardware, software, processes, filesystems etc.
Management	System management. Installation, patches, upgrades, filedistribution etc.
Syslog	Syslogserver, save syslogmessages in the central database etc.
Cache	System monitoring in secure networks, same capabillities like monitoring.
Changeevents	Eventsdatabase searchable.
Plugins	Monitoring plugins.

3. Information and support

There are many ways that one can get information and support on AES Andutteye Software Suite installation and implementation. Dont hesitate to retrive help by the different support channels below.

Website	www.andutteye.com
On line forum	www.andutteye.com
Documentation	www.andutteye.com
Commercial phone support	www.thundera.se
Commercial email support	www.thundera.se
Commercial consulting support	www.thundera.se
Updates and patches	www.thundera.se
Man pages	man <aes component> ex:man andutteyed
Web interface	The ? Button on the left of all columns.

2. Installation of Monitoring module.

1. Requirements.

The Andutteye Software Suite server software needs some additional software and components to be able to work probably.

Required Mysql database
Required Apache (httpd)
Required Php
Required Php-mysql
Required Php-gd
Required Perl-DBI
Required Perl-DBD-mysql
Additional mod_ssl (To be able to use the web interface in https mode.)

These softwares are usually bundled and shipped with the most commonly Linux distributions today. Before you start to install Andutteye Software Suite, make sure that the components above work properly.

On the systems where only the monitoring agent shall be used, only Perl is needed. In 99 cases of 100 perl is installed by default with the operating system itself.

2. Download and install manually.

Download the Andutteye software and verify the checksums and gpg signatures. Read the detailed information on the specific downloads on www.andutteye.com on howto do that.

Make sure that mysql is started and perform the following steps as the root user.

Go to the top of you system so that the software are placed on the right locations.

```
$>cd /
```

Extract the server software.

```
$>tar -zxvf andutteye-server-2.2-2.tar.gz
```

Extract the client software.

```
$>tar -zxvf andutteye-client-2.2-2.tar.gz
```

Create a usergroup for andutteye

```
$>groupadd andutteye
```

Create a andutteye useraccount.

```
$>useradd -g andutteye -c “Andutteye Software Suite user” -m andutteye
```

Correct ownerships and permissions on extracted files.

```
$>chown -R andutteye:andutteye /opt/andutteye
$>chmod -R 770 /opt/andutteye
$>chmod -R 755 /var/www/html/andutteye
$>chmod 755 /etc/profile.d/aesurveillance.sh
```

4. Download and install by rpm.

Install the rpm packages as root

```
$>rpm -ivh andutteye-client*.rpm andutteye-server*.rpm
```

5. Build and create the Andutteye database.

The software is now installed and have correct permissions. Now its time to build the andutteye database.

As the root user execute the following command. Make sure that the mysql database is started. If not start it before you execute the install program.

Execute the following program and follow the instructions. Make sure that you remember what username and password you are specifying.
`$>/var/www/html/andutteye/install/install.sh`

If everything went well we can now start to configure the Andutteye monitoring server listener. If it didn't went well, make sure that the database is started and try again. Make sure that you delete the andutteye database before you try to execute the installation program again.

You can do this with the following command.
`$>mysqladmin drop andutteye`

2. Configure the Andutteye monitoring server listener.

Its now time to configure the Andutteye monitoring server listener. Edit /opt/andutteye/etc/andutteye_server.conf with you favorite editor. Change the username, password and database to the information you specified when creating the andutteye database.

our $DB="andutteye";
our $DBUSER="andutteye";
our $DBPWD="andutteye";

Make sure that you only change the characters in between the “ “. In this case the word *andutteye.*

1. Configure the Andutteye web interface.

Its now time to configure the Andutteye web interface. Edit /var/www/html/andutteye/config/config.php with your favorite editor. Change the username, password and database to the information you specified when creating the andutteye database. Also change the tmpadmin password to something unique.

All web pages are placed under /var/www/html/andutteye which is default document root on Red hat, Mandrake and Suse. If you have specified a different document root on your system you can manually move the andutteye directory and all its sub content to your document root.

```
$HOST       = "localhost";
$USER       = "andutteye";
$PASS       = "andutteye";
$DB         = "andutteye";
$TMPADMINPASSWORD="AndutteyeRocks";
```

2. Set the important system environment variables.

Andutteye software are using two environment variables to be able to locate its software and submodules. These parameters are set in /etc/profile.d/aesurveillance.se and are called:

ANDUTTEYESURVEILLANCE_LOCATION=”/opt/andutteye”
ANDUTTEYESURVEILLANCE_OS=”linux”

In Redhat, Suse and Mandrake all programs under /etc/profile.d are sourced in to every users shell, but on for example Unix systems and other Linux distributions like Gentoo this system inst used. Then you manually have to set them so that the variables are exported both to the root user and to the andutteye user.

You can do this in a simple manner by adding following line to /etc/profile if you want to set them for the whole system

. /etc/profile.d/aesurveillance.sh

Or in /home/andutt/.bash_profile and /root/.bash_profile if you only want to set them for the root and andutteye user.

To verify this try to do.

As root
```
$>su -
```

Verify that the parameters are set. If not recheck the steps above and try again.
```
$>env | grep -i andutteye
```

As root
```
$>su – andutteye
```

Verify that the parameters are set. If not recheck the steps above and try again.

```
$>env | grep -i andutteye
```

If the parameters are set, then you are good to go to the next step.

3. Initial start of Andutteye monitoring server listener

We can now try to start the server listener.

As the root user perform the following step.

```
$>su – andutteye
```

First we try to start the listener manually.

```
$>/opt/andutteye/server/andutteye_server -listen 32000 -debug=5
```

Following output shall pop up on the terminal.

```
** Running server by hand in debugmode
**
** Debuglevel:5 Daemon mode:0 logdir:/opt/andutteye/var
**
** ANDUTTEYESURVEILLANCE_LOCATION are set to:/opt/andutteye
** Setting configfilelocation to:/opt/andutteye/etc/andutteye_server.conf
** Fork_all_connections mode    :NO
** SSL connection mode          :NO
## Daemonmode is not specified will not fork and run in daemon mode
## Andutteye listener information logged to database
## Verify under Serverlog in webbinterface that the server
## have logged Beginning listener on port:32000....
## Then everything should be OK, try starting your agents
## and see if statistics and info are coming in and are loaded.
## by the serverlistener.
## Creating server socket, waiting for incoming connections.
```

Everything works fine. Abort execution of the program by hitting CTRL + C. We can now start the serverlistener by its startprogram

As root execute following program

```
$>/etc/init.d/andutteye_server start
```

4. Initial start of Andutteye monitoring agent.

Now when the serverlistener are started and are waiting for incoming connections when can start to configure our monitoring agent.

Andutteye monitoring agent don't need any preconfiguring since its diagnose the system and creates monitors and configuration based on what processes that are started and defined and what file systems that are mounted and so on. No system is completely identical.

Create a andutteye configuration based on your system by executing.

```
$>su – andutteye
$>/opt/andutteye/bin/andutteyed -genconfig -andutteyeserver=localhost
-andutteyeport=32000
```

When the generation phase are completed then try to monitor based on the configuration.

```
$>su – andutteye
$>/opt/andutteye/bin/andutteyed -monitor -debug=2
```

Abort the program execution with CTRL + C, if everything looked ok then start the agent by its startprogram.

As root user.

```
$>/etc/init.d/andutteyed start
```

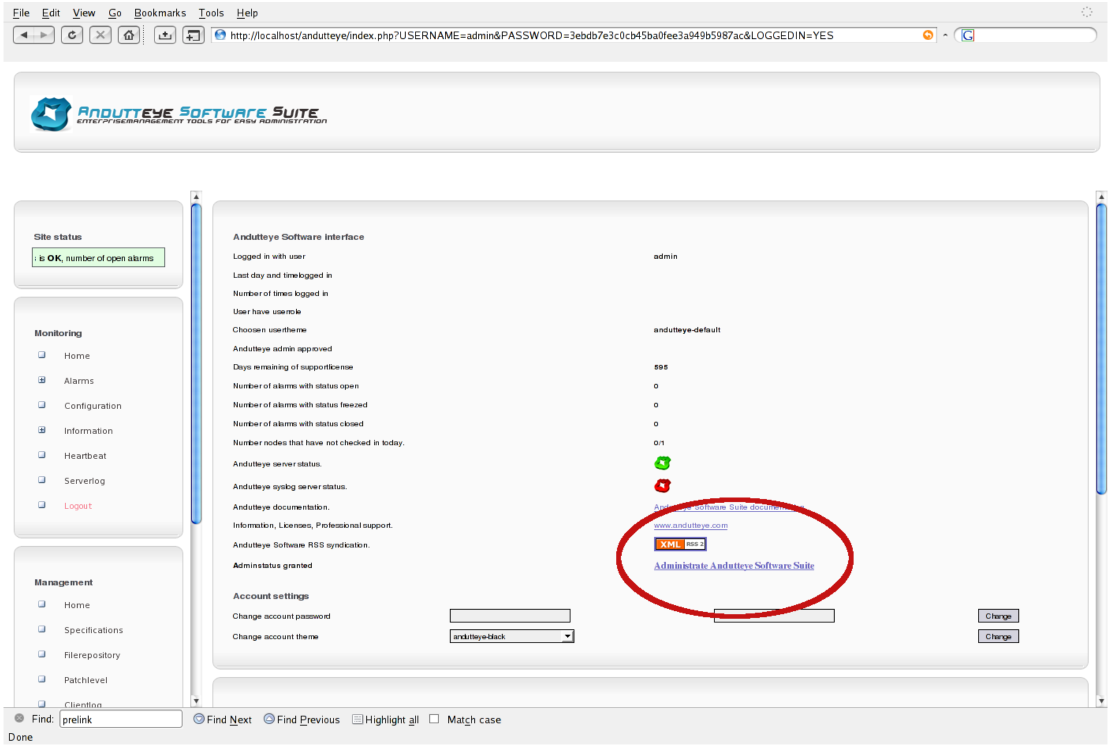
The generation phase can be re executed how many times you want. -genconfig argument has also many more additional parameters you can use, such as -sendemail= and -runprogram= for default email recipient and recovery program for every created monitor. Execute `man andutteyed` for complete arguments list.

We are now ready to start working in the enterprise interface.

5. Login as admin on the web interface

If your web server are started you shall now be able to browse to the webinterface with a web browser. Browse to `http://<yourserver or ipadress>/andutteye` . The Andutteye Software Suite webinterface shall now appear.

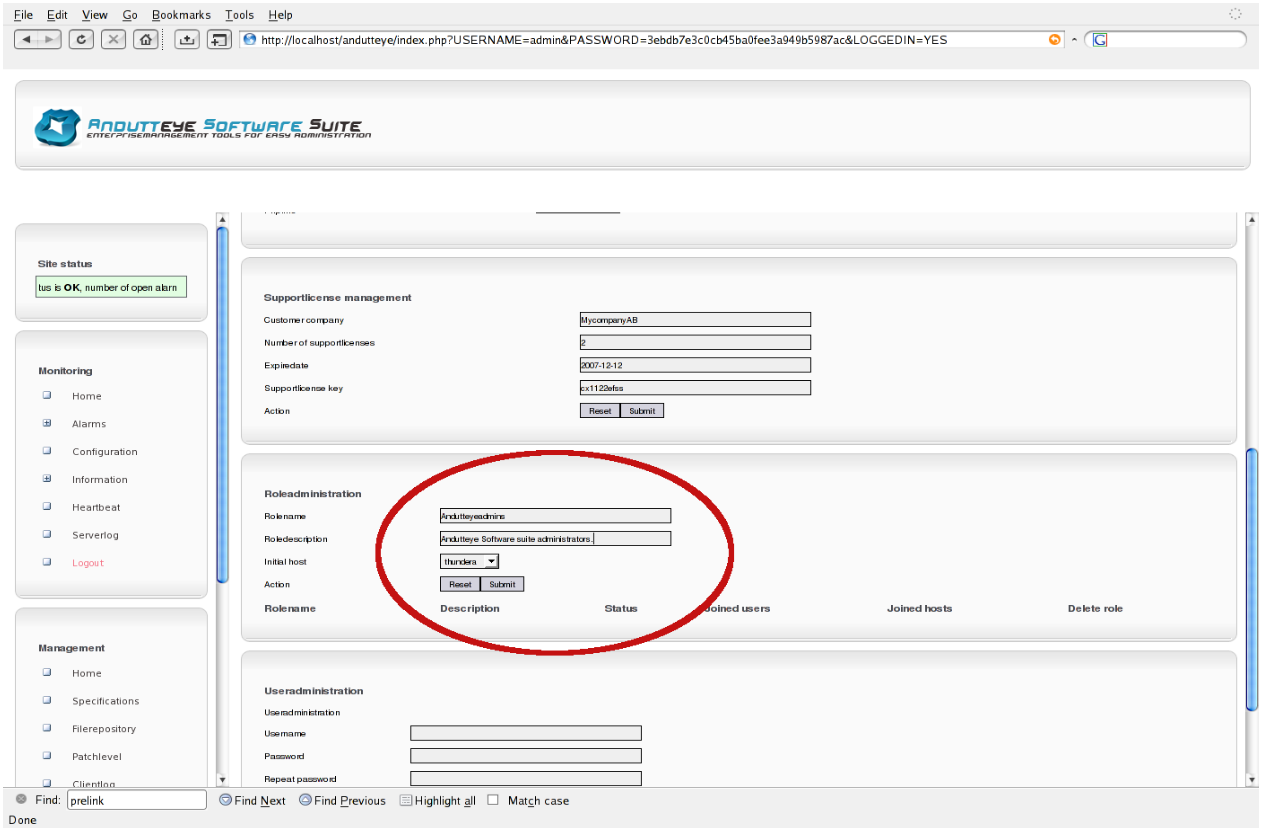
Log in as admin and the admin password that you previously specified in the config.php file. Then press the Home button on the left side, the main information page shall now be loaded in the right frame of the interface. There should now be a blinking link called **Administrace Andutteye Software Suite**. Click on that to enter the admin section.

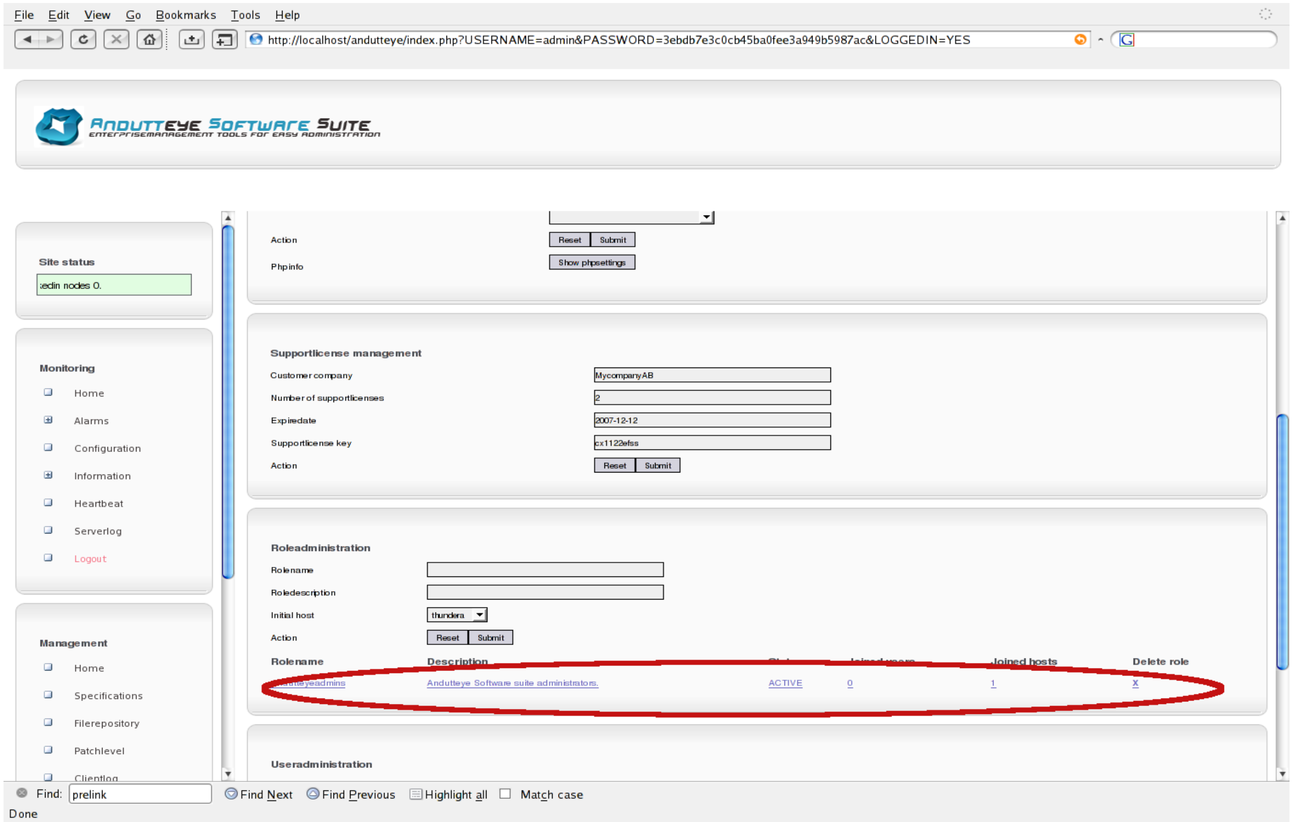


6. Create a user role.

Andutteye is using roles to authenticate users to andutteye objects and modules. The role and its permissions are validated and the users that are connected to it are granted or revoked to the object/module. Therefor you only have to grant or revoke permissions in one place instead for example 100 users that shall have permissions on a object.

Create a users role with your logged in system as default system.


Supportlicense management
Customer company
Number of supportlicenses
Expiredate
Supportlicense key
Action
Roleadministration
Rolename
Roledescription
Initial host
Action
Andutteyeadmins
Andutteye Software suite administrators.
thundera
Reset
Submit
Useradministration
Username
Password
Repeat password

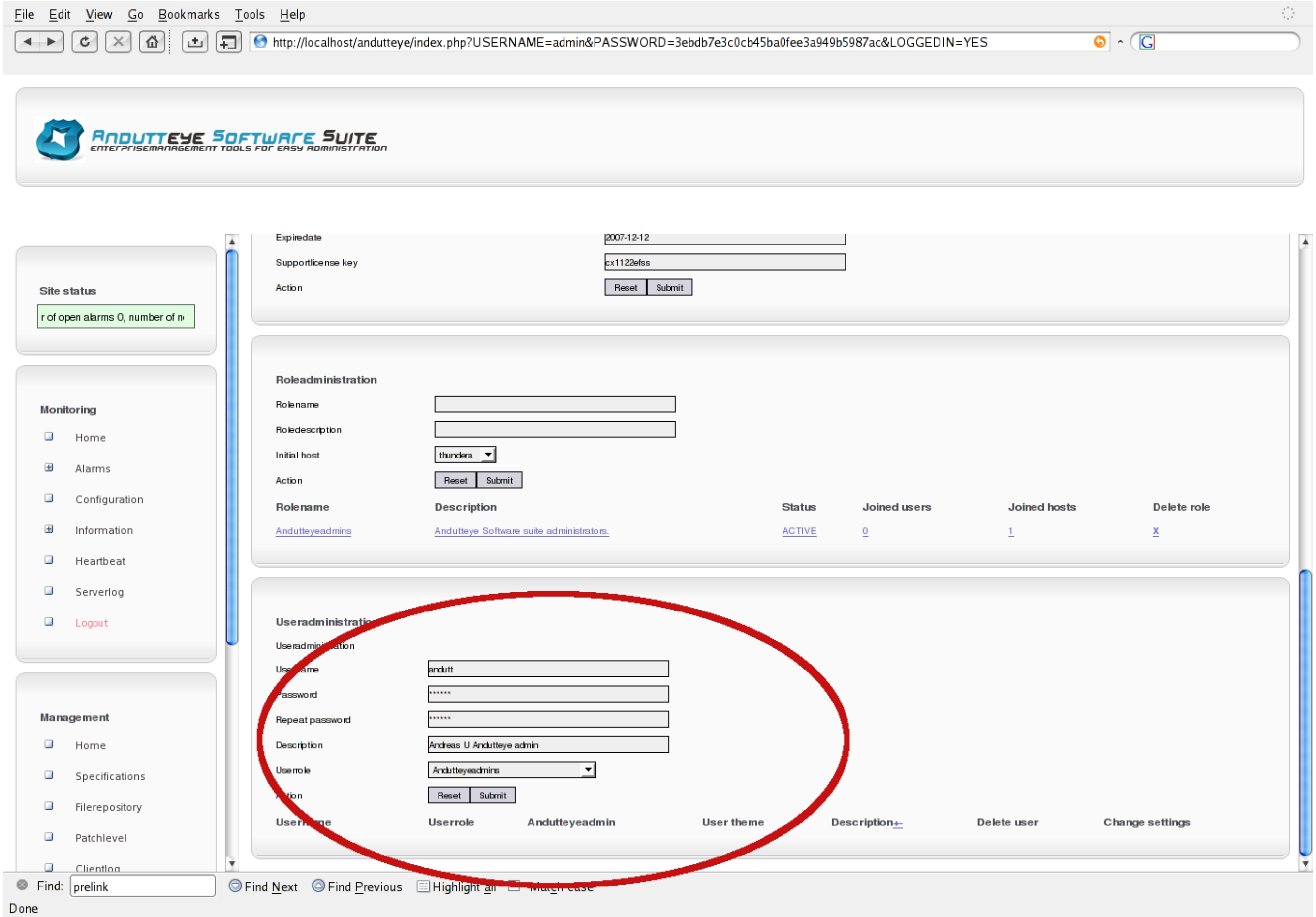


Now click on the line with the new created role to access the permissions section.

Now click on change permissions button for the specific server to grant or revoke permissions to the specific role. The grantfileobject and grantrpmobject is a part of Andutteye Management and isn't relevant for the monitoring administration. The change will affect all users that have this specific role chosen for them. If a new system is checking in it must be added to the specific role to appear in the web interface for the users.

7. Create a user.

Now create a user that you can access and work in the web interface with. Make sure that you are choosing the correct user role. After that you have successfully created the account, select a user theme for the user and click on the change button to commit changes. The default user theme are andutteye-default. If granting andutteye admin to a user it will override all user roles and the user will be able to see, change and delete every checked in system and all information for it.



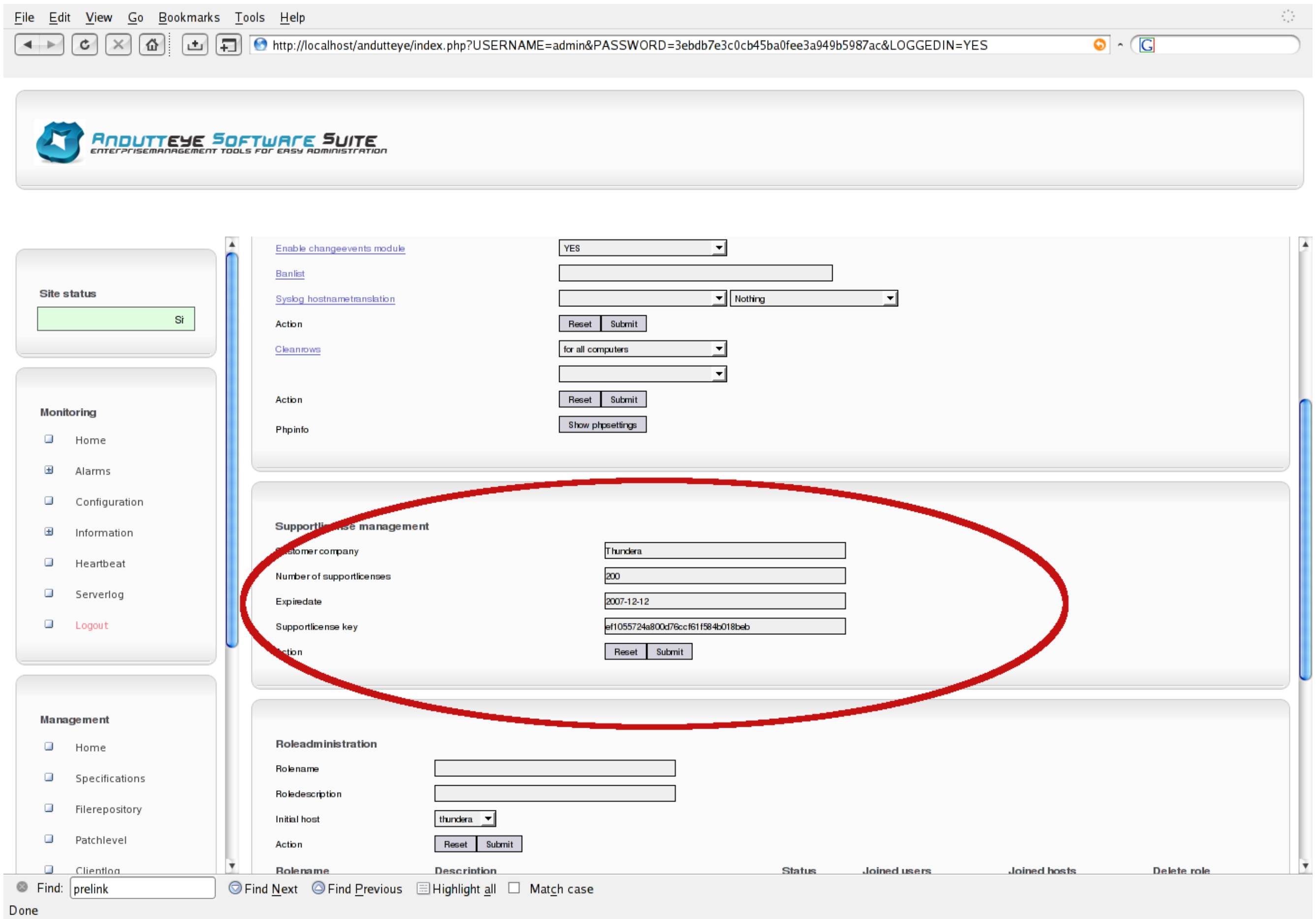
8. Insert support license information

Now insert the license information that you have received when signing up for support or generate a key on the web portal andutteye.com. You will be able to work with as many systems as specified in your key for until the expire date is a curing 30 days before the expire date appears a blinking warning text will pop up in the main information section of the interface which will give you a hint of that its time to update your key. When the key have expired you are unable to work with any system in the interface until a correct and valid key are inserted.

Use following key, until this section is removed.

Company:	ThunderaAB
Number of supportlicenses:	200
Expiredate:	20101022

Supportlicense key: c1d0c42c319730435a3bfc829f767eb0



9. Log out admin and log in with your new user

Log out admin with the logout button on the left side of the interface and login with your newly created user account. You should no be able to see and work with your systems.

3. Installation of Management module.

1. Requirements.

The Management module only needs ssh to be installed to be able to work properly. Management can also be used in different modes. Such filemode or databasemode. In databasemode the Additional packages also needs to be installed.

Required ssh
Additional Apache (httpd)
Additional Php
Additional Php-mysql
Additional Php-gd
Additional Perl-DBI
Additional Perl-DBD-mysql
Additional mod_ssl (To be able to use the webinterface in https mode.)

These softwares are usually bundled and shipped with the most commonly Linux distributions today. Before you start to install Andutteye Software Suite, make sure that the components above work properly.

10.Install Andutteye Management software.

11.Before installing.

AES Management will help you as an administrator to deploy, install, upgrade and in real time verify that your systems are configured and have the integrity that you centrally have configured it.

Which means that before you even install the software you need to have policies ready of what you want to accomplish. How shall a base system look like on you company? Which packages shall be installed? Which files are important to have control on so that one are following your companies security policies and routines? AES gives you the tools to deploy and verify this but its up to the administrator to have clear visions on how your IT environment shall look like.

12.Download and install manually.

Start to install a clean Linux system which you think is a good "base" system for your company. With minimum packages installed. It will later be used to create a base bundle that can be used as foundation of your future systems.

Download the Andutteye software and verify the checksums and gpg signatures. Read the detailed information on the specific downloads on www.andutteye.com on how to do that.

Perform the following steps as the root user.

Go to the top of you system so that the software are placed on the right locations.
```
$>cd /
```

Extract the management software.
```
$>tar -zxvf andutteye-management-2.2-2.tar.gz
```

Create a usergroup for aemanagement
```
$>groupadd aemanagement
```

Create a aemanagement useraccount.
```
$>useradd -g aemanagement -c “Andutteye Software Suite management user” -m
aemanagement
```

13. Download and install by rpm.

Install the rpm packages as the root user
```
$>rpm -ivh andutteye-management*.rpm
```

Make sure that the enviroment variable file are having the right permissions. If you are installing in a non profile.d system add the information in `/etc/profile.d/aemanagement.sh` to /etc/profile or some enviroment file that are sourced in the users shell.
```
$>chmod 755 /etc/profile.d/aemanagement.sh
```

The default location of the Andutteye management software are /var/jail/aemanagment. That because you can run the Management module in a change root environment if one are using the module in terminal/text mode.

14. Configure repository user and ssh enviroment.

As the root users perform following steps.
```
$>cd ~root
$>cd .ssh
```

Generate the ssh keys that the Management agent will use to authenticate against the repository. The public key must later on be distributed to all systems that will use Andutteye Management. Se a blank password on the key, just press enter when the password question

appears.
```
$>ssh-keygen -t rsa -b 2048 -f aemanagement
```

Open the public key you have just created with cat or more command and copy the contents of the key. It will later on be inserted in the aemanagement users authorized_keys2
```
$>cat ~root/.ssh/aemanagement.pub
```


ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAmglm8w9LhfX0Y+E1mt5QC/07j8rJwtABJBG9YK0a4zEVJHe/H
kh8QwijQ9K7efFqJteLUGyjMvGjBKpu1HX2xj9rnda/FXAtxiZq5EnSDBIcNOm6W+7uv/uEn+tcE1
AeprxMF8dgV2s4aeys8wUAoDDRPOez9AQN5VGeuuFWWxwNPfxx5YOAnUl5vnV1fyvjj6K8cCc4oCv
2ecII8vuz6yNCrilhHJsz1pt3Zca5omRFDZy5SnBRXB6Ejnw0/22dq1/4TgLWijSPtQY2AdDYH0Lj
cjrQJR5Jq0mriURV2278dFEhgR8YxSSpeLjQUw+cdmkwg5axvsptXRKDn6kdXQ==
root@thundera

Switch user to the aemanagement user.
```
$>su – aemanagement
```

Create symlinks to where the Management module directorys are located.
```
$>ln -sf /var/jail/aemanagement/out out
$>ln -sf /var/jail/aemanagement/in in
$>ln -sf /var/jail/aemanagement/bin bin
$>ln -sf /var/jail/aemanagement/log-client log-client
$>ln -sf /var/jail/aemanagement/specifications specifications
$>ln -sf /var/jail/aemanagement/bundles bundles
$>ln -sf /var/jail/aemanagement/config config
$>ln -sf /var/jail/aemanagement/packages packages
$>ln -sf /var/jail/aemanagement/tmp tmp
$>ln -sf /var/jail/aemanagement/work work
$>ln -sf /var/jail/aemanagement/files files
```

Create the .ssh directory and set the right permissions.
```
$>mkdir .ssh
$>chmod 700 .ssh
```

Create a file called authorized_keys2 with your favorite editor.
```
$>cd .ssh
$>vi authorized_keys2
```

Paste in the public key you copied in the prior step.

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAmglm8w9LhfX0Y+E1mt5QC/07j8rJwtABJBG9YK0a4zEVJHe/H
kh8QwijQ9K7efFqJteLUGyjMvGjBKpu1HX2xj9rnda/FXAtxiZq5EnSDBIcNOm6W+7uv/uEn+tcE1
AeprxMF8dgV2s4aeys8wUAoDDRPOez9AQN5VGeuuFWWxwNPfxx5YOAnUl5vnV1fyvjj6K8cCc4oCv
2ecII8vuz6yNCrilhHJsz1pt3Zca5omRFDZy5SnBRXB6Ejnw0/22dq1/4TgLWijSPtQY2AdDYH0Lj
cjrQJR5Jq0mriURV2278dFEhgR8YxSSpeLjQUw+cdmkwg5axvsptXRKDn6kdXQ==
root@thundera

Make sure that the file have correct permissions
```
$>chmod 600 authorized_keys2
```

Create a file with your editor called environment in your ssh directory and add following text in it. One must also make sure that PermitUserEnvironment are set to yes in your sshd_config. Otherwhise set it to yes and perform kill -HUP on sshd to make it to reread its configuration.
`ANDUTTEYEMANAGEMENT_REPOSITORY=/var/jail/aemanagement`

Set the correct permissions on the file.
`$>chmod 600 environment`

Now as the root user try to log on using the new key. If all permissions are correct you will be able to log on without any pass phrase.
`$>ssh -i ~root/.ssh/aemanagement aemanagement@localhost`

If it worked continue with the next step, if not make sure that public key authentication and user enviroment are enabled in your sshd_config and the all permissions stated above are committed to the files.

Switch user to the aemanagement user.
`$>su – aemanagement`

Edit the file again with your favorite editor.
`$>vi .ssh/authorized_keys2`

Add following text before the key so it looks like this.
```
no-pty,no-X11-forwarding,no-port-forwarding,command="bin/aemanagement-
sshwrapper.pl" ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAmglm8w9LhfX0Y+E1mt5QC/07j8rJwtABJBG9YK0a4zEVJHe/H
kh8QwijQ9K7efFqJteLUGyjMvGjBKpu1HX2xj9rnda/FXAtxiZq5EnSDBIcNOm6W+7uv/uEn+tcE1
AeprxMF8dgV2s4aeys8wUAoDDRPOez9AQN5VGeuuFWWxwNPfxx5YOAnUl5vnV1fyvjj6K8cCc4oCv
2ecII8vuz6yNCrilhHJsz1pt3Zca5omRFDZy5SnBRXB6Ejnw0/22dq1/4TgLWijSPtQY2AdDYH0Lj
cjrQJR5Jq0mriURV2278dFEhgR8YxSSpeLjQUw+cdmkwg5axvsptXRKDn6kdXQ==
root@thundera
```

That will force all requests that are using the ssh-key to pass the aemanagement-sshwrapper.pl All request that isn't a defined Management request will be dropped by the wrapper.

Create the /opt/aemanagement directory as the root user.
`$>mkdir /opt/aemanagement`

Copy the aemanagement.pl program from aemanagement-agent directory and place it in the new directory.
`$>cp aemanagement.pl /opt/aemanagement`

Fix the permissions.
`$>chmod -R 770 /opt/aemanagement`

Edit the program with your editor and change host, port and sshkey parameters to match your current setup.

15.Populate the package repository.

The management module can handle multiple rpm package repositorys in paralell. Its the servers distribution: parameter in its specification that specifies which package repository to use. Its now time to create a package repository and populate in with the distributions packages.

As the root user, switch directory to $ANDUTTEYEMANAGEMENTREPOSITORY
```
$>cd $ANDUTTEYEMANAGEMENT_REPOSITORY/packages
```

Create a directory that will be the identifier for this distribution. It can be called anything. Alpha and alpha numerical values and -. can be used in the identifier.
```
$>mkdir redhat-es4
```

Change to the new directory
```
$>cd redhat-es4
```

The management module can handle multiple patch levels for the distribution. That offers another dimension of patching and system upgrading capabilities. It will be covered further in this documentation. For now we create the first base patchlevel where we place all packages that are distributed on the os vendors distribution media.

Create the base patchlevel
```
$>mkdir 0
```

Change to the new directory
```
$>cd 0
```

Copy all packages from the media to this patch level. This can done on many ways depending on which operatingsystem one are using. If one are using cd, DVD or downloading from Internet.
$>cp /media/cdrom/Redhat/RPMS/*.rpm .

When all packages that you want to have in the repository are copied to this patchlevel its time to create a AES management register of the packages. This will make the processing and validation of the systems really fast.
```
$>$ANDUTTEYEMANAGEMENT_REPOSITORY/bin/aemanagement-genindex.pl -directory=
$ANDUTTEYEMANAGEMENT_REPOSITORY/packages/redhat-es4/0
```

The register process will now start and finish when all packages have been verified. This means that all packages that have been registered can be deployed. If one are placing new packages in the patch level the register process program must be rexecuted for the patch level.

16.Create the base bundle.

Bundles are a AES term and can be translated to building blocks or channels that can be included in the systems specification. A bundle contains specified packages. For example a file called redhat-es4-httpd that are containing the httpd related packages and which versions it shall use. We will go deeper into the bundle specifications later in this documentation

Create a base bundle based on this system. Switch directory to the bundles directory.

```
$>$ANDUTTEYEMANAGEMENT_REPOSITORY/bin/aemanagement-genindex.pl
-generatebaselist > redhat-es4-base
```

A file called redhat-es4-base shall now be present in the bundles directory.

17.Create a specification.

The system specification is where all important AES parameters are saved. Here the systems package profile is specified, which distribution it shall use, which patch level it shall have. If AES have permission to change the system to look like its central profile or only notify the administrator that differences have been found. For short everything. Its now time to create a specification profile for your system.

Copy the example specification to your system.

```
$>cp example `hostname`
```

Will copy the example file to the hostname that are specified for your system. The hostname are used by all AES modules to recognize which system its running on. Now edit the file with a editor and change distribution to the one we now have created, redhat-es4. Change patch level to 0 since we only have one patch level yet.

Set GROUP and WHERE to something that resembles this system and what functionality it has. It will be used in the file override system. For example GROUP:ANDUTTEYE WHERE:PROD or GROUP:JBOSS WHERE:TEST. We will cover the fileoverride system later in the documentation. All parameters can be changed in runtime.

Set allow-rpmupdate to:no so the management agent cant install or uninstall anything until we have made this system up2date. Also set allow-configupdate to no.

Now include the bundle we just created in the bundles section. So it looks like this.

```
bundles:redhat-es4-base
```

It means that this server shall include all packages that resides in `redhat-es4-base` bundle.

18. Making the base bundle up2date.

AES doesnt support multiple packages in the same bundle so therefor you probably have to do some manual corrections in the bundle and on your system before everything is up2date. This is also a good rule to think of when one is building ones own packages. If you have configured rpmupdate:no and configupdate:no we can now try to make the management system up2date.

Execute the andutteye management agent by hand.
`$>/opt/aemanagement/aemanagement.pl`

Probably it will fail on some package issue. Review the client log that was created under the log-client directory to get a clue on what went wrong. Correct the bundle, specification to solve the problem that caused the error message.

4. Module design and functionality

1. Monitoring

1. Monitor framework

The monitoring framework was created to make monitoring easy and less time consuming. The monitor examples and prefixes below one must know to change the monitoring profile by hand direct in the systems monitoring configuration file andutteyed.conf. If running AES in enterprise mode with monitoring server and webinterface , all this administration are performed from the central webinterface and standard forms.

2. Monitoring frameworktypes and prefix

The monitor framework is built for one to get an easy and structured way to create new monitors and change current ones. AES monitor framework covers different kinds of systems monitoring. To most common things one want to monitor has a monitor specification that makes it very simple to administer and change in daily production. The current monitor types that are covered by the AES framework for now is.

MA	Memory average monitor
LA	Load average monitor
SA	Swap average monitor
PS	Process monitor

FS	File system monitor
FM	File modification monitor
FT	File trace/pattern match monitor
PH	Ping host/communication monitor
EVERY	Execute a program or command on every control loop.

The things that are covered by the framework can be in a very easy manner be added or removed from the central monitoring profile of your system. For example for a new process we like to monitor (smb, Samba) one only adds a monitor specification, **PS=smb**. The same for a new filesystem /smb, **FS=/smb**. If one want to be notified if the file /etc/sudoers have been changed, add a filemodification monitor, **FM=/etc/sudoers**. All sysems monitoring profiles can be changed on the localsystem or in the central AES interface.

3. Monitor framework recovery capabillities

Every monitor type have recovery capabilities that andutteyed can execute if any monitor have been triggered. Following capabilities are default and that all monitoritypes uses and shall have defined. If the recovery capabilltiy isnt enabled it always sould be set to **[NO]**.

SENDEMAIL=<emailadress>
SENDEMAIL=NO
SENDEMAIL=andutt.andutt@andutteye.com,nisse.andutt@andutteye.com

Here one can specify an emailadress, emailgroup or multiple emailadress that shall be notified if the specific monitor is triggered. The MESSAGE value will be inserted in the email if it is defined. Multiple emailadresses must be delimited with [**,**].

RUNPROGRAM=<program or script>
RUNPROGRAM=NO
RUNPROGRAM=/tmp/myrecoverscript.sh
RUNPROGRAM=/bin/netstat~-anp~|~grep~32000

Here one can specify a program, command or script that shall be executed if the monitor is triggered. Commands or programs with arguments can be used but all whitespaces must be delimited with [**~**]. Andutteyed will monitor the exitcode of the recovery program or command and will create an active alarm if the program have a nonzero exitcode.

SEVERITY=<severity>
SEVERITY=HARMLESS|WARNING|CRITICAL|FATAL
SEVERITY=CRITICAL

Here one can specify which severity the monitor shall have. There are four different severitys used in AES. HARMLESS, Information or noncritical information. WARNING, Low impact for the overall system state. CRITCAL, Big impact for the overall system state. FATAL, Fatal impact of the overall system state. A severity of a monitor must always be defined.

SCHEDULE=<hour-hour>
SCHEDULE=NO
SCHEDULE=08-17
SCHEDULE=DISABLED

Here one can specify in between which hours of the day the specific monitor shall be active, for example between 08:00 to 14:00 on set schedule to SCHEDULE=08-14. If the monitor shall be disabled just set schedule to disabled.

LIMIT=<limit>
LIMIT=NO
LIMIT=3

Here one can specify a monitoring limit. This will force andutteyed to not to monitor or take recovery action on a triggered monitor until the monitor have changed state and gone down again. For example LIMIT=5 will make that andutteyed only execute recovery actions five times. The LIMIT parameter can be used with numerical values. 1-100000000000.

MESSAGE=<personal alarm message>
MESSAGE=NO
MESSAGE=My~service~testprogram~has~encountered~an~error.

Here one can specify a personal message that shal be used if a monitor is triggered. The message overrides the standard alarm message. Sentences up to 255 sings can be used but all whitespaces must be delimited with [**~**].

Monitor types and example below.

Some monitor types have other fields and variables that one must set. See the specific monitor type for more information.

```
PS=acpid,UP,RUNPROGRAM=NO,SENDEMAIL=NO,SEVERITY=WARNING,SCHEDULE=NO,MESSAGE=NO,LIMIT=NO

FS=/,85,90,95,RUNPROGRAM=NO,SENDEMAIL=NO,SCHEDULE=NO,MESSAGE=NO,LIMIT=NO

FM=/etc/passwd,RUNPROGRAM=NO,SENDEMAIL=andutt@localhost,SEVERITY=FATAL,SCHEDULE=NO,MESSAGE=Password~file~has~been~modified.,LIMIT=NO

FT=/etc/sudoers,^kalle,15,RUNPROGRAM=NO,SENDEMAIL=NO,SEVERITY=CRITICAL,SCHEDULE=NO,MESSAGE=NO,LIMIT=NO

EVERY=/tmp/test.sh,RUNPROGRAM=NO,SENDEMAIL=NO,SEVERITY=WARNING,SCHEDULE=NO,FLAGS=-execute~-flags,MESSAGE=NO,LIMIT=NO
```

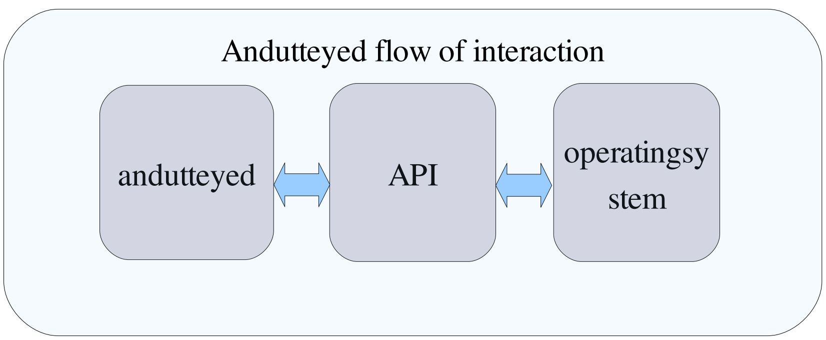
```
PH=127.0.0.1,RUNPROGRAM=NO,SENDEMAIL=NO,SEVERITY=CRITICAL,SCHEDULE=NO,MESSAGE=NO
,LIMIT=NO
```

2. Andutteyed the monitoring agent.

1. Andutteyed architecture.

Andutteyed is the actual monitoring agent that every system that shall be monitored must have installed. Andutteyed contains the logical layer of the monitoring agent. This mean that its not directly communicating with the operating system but handles all Andutteye specific tasks. As the communication layer, encryption and validation of Andutteye settings and validation of data from the under layer API.

The flow of interaction is illustrated below.



Andutteyed is using the environment variables ANDUTTEYESURVEILLANCE_LOCATION and ANDUTTEYE_SURVEILLANCE_OS to determine where the andutteye monitoring software is installed and which underlaying operating system API to use. So for use on Linux ANDUTTEYE_SURVEILLANCE_OS is set to Linux on on windows to windows etc.

Therefore its very simple to port the andutteye monitoring agent to different operating system, just port the API to the new os, since andutteyed only want data formatted in a certain way, thats the API:s task. The same andutteyed is used for every operating systems which gives the same support and monitoring framework operating system independent.

2. Andutteyed arguments

Andutteyed can be in any time executed by hand and have many arguments that can be supplied for specific tasks. Following are available.

-convconfig
Convert the andutteyed.conf with new functionality and reformat it if nessasary.

-genconfig
Perform a diagnose of the system and generate a andutteye configuration based on the system.

-runprogram
Specify a default script or command that shall be executed for every created monitor.
-sendemail
Specify a default email reciptient or email group that shall be notified for every created monitor.
-andutteyeserver
Specify the andutteye server dns name or ipadress.
-andutteyeport
Specify the andutteye server port which the andutteye server is listening on.
-addall
Add all found services. Thoose who isnt started add them to suppose to be down.

-monitor
Monitor the system based on the generated configuration.

-debug
Specify debuglevel of the program from 0-5 where -debug=5 shows everything
-logdir
Specify where the log of the andutteyed execution shall be saved.
-daemon
When -daemon is specified andutteyed refork to / and releases the shell.
-smsformat
Reformat all email notifications to be as short as possible.

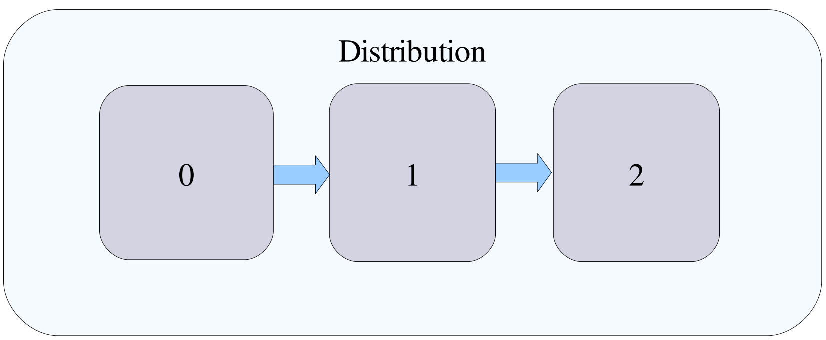
3. Monitor autodiscovery and monitor creation.

Andutteyed monitor generation can be done completely automatic with the -genconfig argument. Andutteyed will then discover your system and verify which processes that are started and are defined services. Which file system currently mounted and create a andutteyed configuration file. Therefor the manual administration are decreased to almost nothing. The generation phase can be re executed as many times as you want, the old configuration file will then be overwritten. Additional arguments can also be supplied when executing -genconfig to even more decrease administration, such as default recovery program or command, default email recipients or email groups that shall be notified, andutteyeserver and andutteye port that shall be used.

3. Management

1. Patchlevels.

Patchlevels is the logical level under a distribution repository, both the distribution and the different patchleves are normal directorys under $ANDUTTEYEMANAGEMENT_REPOSITORY/packages. A patchlevel contains packages that one wants to distribute and to be available to AES managed servers. During the installation one already created patchlevel 0 that contains the packages that was distributed with from your operating system vendor. See the illustation below how a distrbution repository with three patchlevels can look like.



The patchlevel design gives one another dimensions of administration. In the managed servers specification a distribution parameter and patchlevel parameter are set. Those two parameters decides which packages and which version and release of packages the server shall have installed.

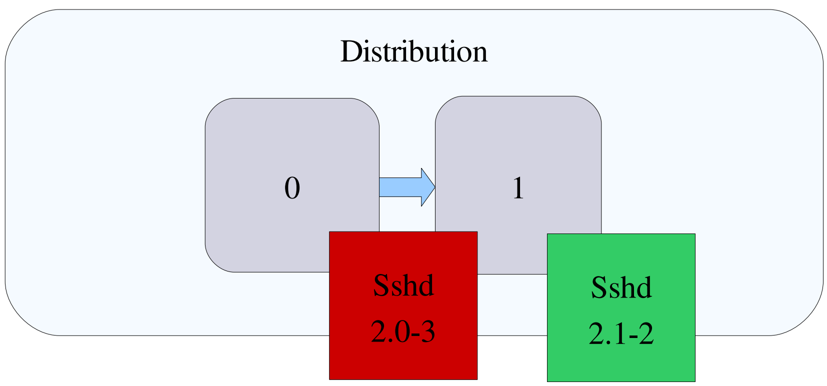
Lets see a senario.

In patchlevel 0 all distribution packages are registered, we have a sshd package with version 2.0 release 3 in that patchlevel. Now our vendor have released some patches that we want to patch our AES managemed servers with. So we create a new patchlevel directory under our distribution directory called 1. All new downloaded patches are placed in the patchlevel 1 directory. In patchlevel 1 now a new sshd package lies and a bunch of others. The new sshd package has version 2.1 release 2.

We register the new packages and patchlevel with the aemanagement-genindex.pl program, when the program has completed the new patches are available to our AES managed system. But, nothing will happen yet, since all our servers now have patchlevel 0 defined in its specification.

We now change patchlevel on a test system to use patchlevel 1 instead of 0. Now things will start to happen.

When the testsystems AES agent verifies the current package profile for the system i sees that it have sshd version 2.0 release 3 currently installed. It will then see which versions and releases of sshd that are available **up to its defined patchlevel**. Which means that it will start at patchlevel 0 and verify all available patchlevels to the defined patchlevel which in this case is 1. So in this case AES will notifice that a newer package is registered in patchlevel 1 and will install this on the system.



With the patchlevel design one:s systems can separately be patched, installed and upgraded without inflicting on other systems. Its also very simple to maintain and administer one or three hundreds of packages in a patchlevel with the automatic registration process and a simple parameter change of the patchlevel parameter.

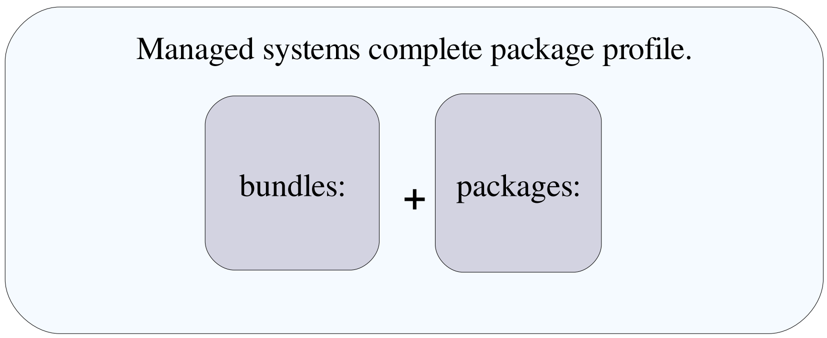
Some information about patchlevels that is good to know

- Only one package with the same name can reside in the same patchlevel.
- Packages inside a patchlevel must be registered by the aemanagement-genindex.pl program otherwise new packages is “invisible” to AES until the registration process have been executed.
- Patchlevels can be locked by administrator which are done automaticly by AES programs so that patchlevels can be administered and maintained without inflicting managed systems.
- Patchlevels can only be named numeric integers, 0->10000000000.

2. Bundles.

Bundles are files that can be compared with a channel or a build block. A bundle contains a package or many packages. The bundle is used when more then one package is to be added to a managed system. A bundle included for a system is a part of the systems complete package profile

See illustration below:



Bundles are used to keep the systems specification file “clean” so its readable. The packages: parameter inside the specification can be used to add or subtract packages from the systems complete profile.

A systems profile can look like this.

```
packages:+cvs–0--0,+subversion--1.0--4
bundles:redhat-es4-base,redhat-es4-httpd,redhat-es4-mysql
```

Thoose two parameters specifies the complete package profile for a system. As one can see for this system it shall have three bundles. Basesystem, httpd, mysql. The bundles can be named anything one like and contain any package one thinks.

The format of specifying packages inside a bundle can look like this.

```
$>cat redhat-es4-mysql
mysql-common 0 0
mysql-server 0 0
php-mysql 2.1 4
perl-dbd-mysql 0 0
```

Format: <package name> <package version> <packager release>

So here comes a smart part with patchlevels and AES. As one can see the most packages have version 0 and release specified. Which means that AES will take the newest version and release of the current package **up to defined patchlevel for the system**.

On the php-mysql package a version and release value is set which means that AES will use version 2.1 release 4 of php-mysql **even if a newer package exists up to defined**

patchlevel of the system. This can be very useful for example java and that kind of software that most be upgraded in a more controlled way. But with packages set to best version and best release you can imagine how little administration it is to upgrade 200 packages on a system, just change the patchlevel parameter in the specification and AES takes care of the rest.

3. Best file match override system.

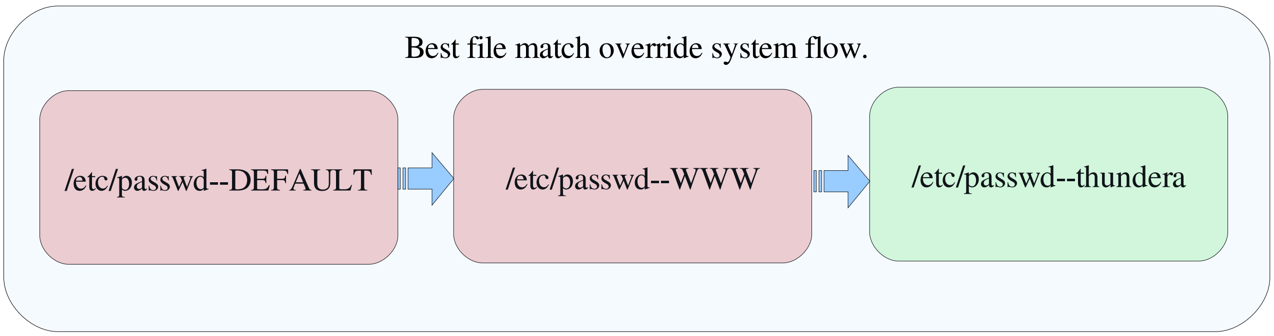
AES control files with a best file match override system where the group, location, patchlevel and hostname parameters in the specification and in combination decides which exact files the system shall have implemented. The best match found for the specific file will be used. One can add more parameters and words that shall be validated in the override system by adding those to aemanagement-config.conf.

```
# The filetype parameter is an array that can be filled with dynamic filegroups
that will be used before standard group, location and patchlevel matching.
our @filetypes   = "DEFAULT";
```

The AES override system will try to match files in 16 different combinations in following order where 16 is the best match and 1 is the worse match.

1. file—DEFAULT
2. file—DEFAULT--PATCHLEVEL
3. file—GROUP
4. file—GROUP--PATCHLEVEL
5. file—LOCATION
6. file—LOCATION--PATCHLEVEL
7. file—ARCHTYPE
8. file—ARCHTYPE--PATCHLEVEL
9. file—GROUP—LOCATION
10. file—GROUP--LOCATION--PATCHLEVEL
11. file—GROUP--ARCHTYPE
12. file—LOCATION--ARCHTYPE
13. file—GROUP--LOCATION--ARCHTYPE
14. file—GROUP--LOCATION--ARCHTYPE--PATCHLEVEL
15. file—HOSTNAME
16. file—HOSTNAME—PATCHLEVEL

For a system called thundera with group:WWW and location:TEST the override system would function like this.



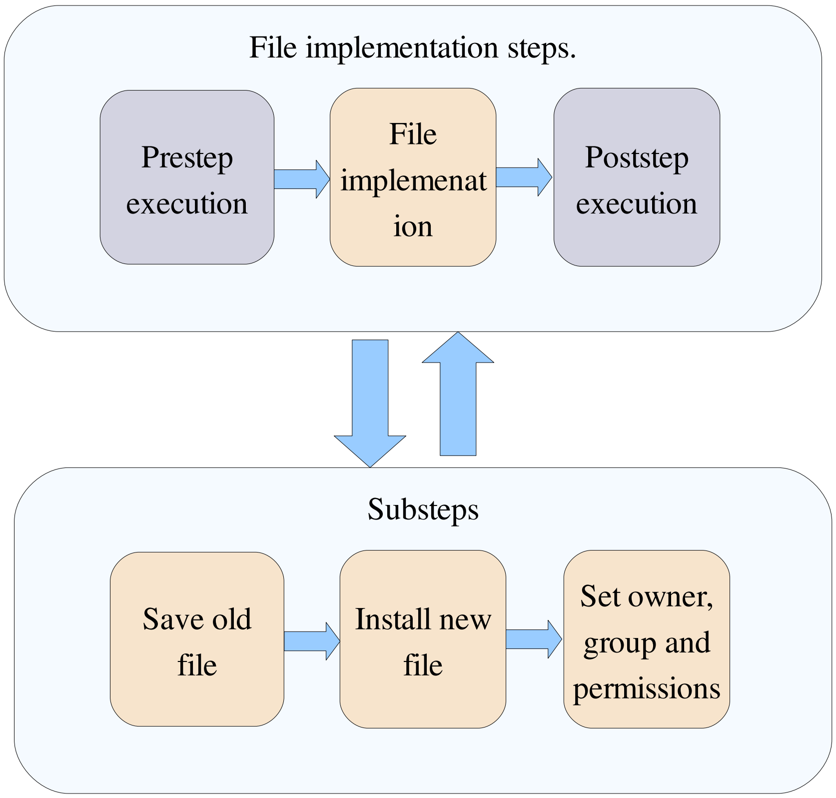
The best match was in this case the hostname, thundera. When working with AES file management it can look like this.

```
$>cd $AEMANAGEMENT_REPOSITORY/files/redhat-es4/etc
$>ls
passwd—DEFAULT passwd—PROD passwd—TEST passwd—thundera group—DEFAULT
```

One works in real time and if a file is changed AES will notice this and install the changed file on the systems that shall have it according to the file match system. One can imagine how powerful this is. A –DEFAULT file can be used for many hundreds of systems but systems on certain locations or that have special tasks can override the default file with files that have better “status” in the override system. The override system make sure that the administration haves as minimal administration as possible.

4. Pre and post execution of commands and programs.

AES can be configured to execute commands or programs before and after a file is being implemented or replaced. This can be useful if for example httpd.conf shall be replaced. Then the httpd service can be stopped before, the file is being implemented, and the service is being started again. With this functionality one can install, upgrade and administer the systems fully automatic in daily production.



Pre and post steps and owner, group and file permissions are specified in the file register $ANDUTTEYEMANAGEMENT_REPOSITORY/config/aemanagement-filesettings.conf. When a new file is added to the file repository the file must also be added to the file register otherwise cant AES know what actions to take when the file shall be implemented.

- A file must always be specified in aemanagement-filesettings.conf.
- Pre and post step are optional.
- AES agent will abort if now settings are found in the file register.

5. The host specification

The host specification is the managed systems AES profile. All back end components and AES logic are using parameters defined in the specification to be able to execute specific tasks. A specification can look like this.

```
packagetype:rpm
distribution:redhat-es4
```

```
patchlevel:1
group:THUNDERA
location:PROD
archtype:i386
description:Thundera development system
status:active
exclude:
packages:+cvs--0--0
bundles:redhat-es4-base,redhat-es4-httpd,redhat-es4-mysql
allow-rpmupdate:no
allow-configupdate:no
allow-syslog:yes
email:andutt@thundera.se
```

like this. The specification for the managed system shall exist under $ANDUTTEYEMANAGEMENT_REPOSITORY/specifications and shall be file named after the managed systems host name, produced by executing the `uname -n` command on the managed system.

Packagetype
: Specifies which package system that are being used on the managed system. For now only rpm are supported though apt and aes package systems will soon be supported also.

Distribution
: The distribution specifies which operating system distribution this system shall use. The distribution parameter must correspond with a distribution directory created under $ANDUTTEYEMANAGEMENT_REPOSITORY/packages

Patchlevel
: The patch level parameter specifies which patch level this system shall have and use. The patch level parameter must correspond with a patch level directory created under $ANDUTTEYEMANAGEMENT_REPOSITORY/packages/<distribution>/<patch level>.

Group
: The group parameter is being used by the file match override system and it can be anything you like. Ex WWW for Web servers.

Location
: The location parameter is being used by the file match override system and it can be anything you like. Ex PROD for Production.

Archtype
: The where parameter is being used by the file match override system and it can be anything you like. Ex i386.

Description
: The description parameter is just a system description and it can be anything you like. Ex Thundera development system.

Status

The status parameter states if the system is allowed to be controlled by AES or not. If it is allowed it should be set to active, if one are performing maintenance or some other work on the system the set it to disabled and AES agent will not have permission to control the system.

Packages

Here packages that one want to include or exclude from the overall bundle profiles shall be specified. Ex +cvs—0—0,+subversion--2.1--4

Bundles

Here shall bundle building blocks be specified that the system shall use. And it can be anything one want.
Ex redhat-es4-base.

Allow-rpmupdate

Here one shall specify if AES have permission to correct rpm differences on the managed system and the central profile. For example if someone have installed packages by hand on the managed system and not registered the in AES centrally. Then AES will uninstall the packages of allow-rpmupdate are set to yes. If set to no it will only notify the administrator by email if any are specified, in the logs under log-client and in syslog if it are enabled.

Allow-configupdate

Here one shall specify if AES have permission to correct file differences on the managed system. For example if someone have changed a AES managed file directly on the system. Then AES would want to distribute and install the file registered in AES. If set to no it will only notify the administrator by email if any are specified, in the logs under log-client and in syslog if it are enabled.

Allow-syslog

Here one shall specify if AES have permission to notify the local syslog if it fins differences on the managed system. Ex yes or no.

Email

Here shall email receptions, email groups be specified if AES shall email someone if it finds differences on the managed system. Leave it empty if none shall be notified.