- Noah - The DokDoc
  - Introduction
    - Who is Noah for?
  - Installation
    - Requirements / settings:
      - Vmware Image:
      - For sending mail from Noah
      - Noah Zipfile
    - Upgrade
- Administrator
    - Default setup
  - Configurations (Administrator Menu)
    - System License(s)
    - Core Settings
    - LDAP
    - IMAP
    - Rights
  - Administration (Administrator Menu)
      - Asset Status
      - Asset Types
      - Circuit Types
      - CPU Types
    - Tickets
      - Ticket Categories
      - Ticket Types
      - Ticket Sub Types
      - Password Types
      - OS Types
      - User Groups
- Users
  - The AssetID
  - Deleted Assets/Customers/Users
  - Cluster(s)
  - Other ID´s
  - IPv4
  - IPv6

ITS (Incident Ticket System)
Ticket Overview
Create Ticket
Create Followup
Link to Followup
Assets
Create Asset
Edit Asset
View Asset
System Menu
Orders
Create Storage Order
Create Backup Order
Create Server Order
Disksystem and Diskgroups
Documents
Uploading of files:
Extractions
Other Options
MailParser
Example of the setup of the mail parser with maildrop:
Example of the setup of the mail parser with procmail:
The Mailparser.php Script
From:
Subject:
Included
Scripts running continous
crontab_agreement_warning
crontab_create2D
crontab_check_upgrade.php
crontab_check_breached_tickets.php
crontab_imap_ticket
Crontab Example

Noah - The DokDoc

Introduction

Noah (AssetManagement) has been developed by gnf.dk (http://gnf.dk) to accomodate the need for a central registration of assets while having the option to create tickets(tickets) on the assets as an integrated part of the system.
The system is based on PHP5 (http://www.php.net) and MySQL5 (http://www.mysql.com). This for easy alteration and while not a "standard" system, it is coded by some standards implied by PHP and MySQL.

Who is Noah for?

Noah is right for you / your organisation if you are handling more than 20 assets (the system is developed with IT equipment in mind - but it is usable with minor alterations for possibly anything else.) and need to know who did what to what when.

Installation

Requirements / settings:

Noah is delivered in two ways. A zip file or a VmWare image.

Vmware Image:

http://gnf.dk/files/noah_image.zip is an ".ovf" (vmdk) file that is to be imported into your vmware environment.

The image is a Debian 5.0 (patched 20/7/2011) with an Apache2.0 and Mysql5.0 running.

System setup is :
User: noah
Password : password

root (su): password
eth0 : 10.10.5.10 255.255.255.0 gw 10.10.5.1

Mysql setup:
mysql user: root
mysql password : password

Noah is installed in /var/www/
The owner of the path and subpaths (including files) should be www-data:www-data! Login may fail if not so!
(chown -R www-data:www-data /var/www/)

Noah basic install with :
user: admin
password: password

Remember to alter the passwords as others may read this ;-) and enter a valid system license in Noah under Noah Settings/ System License(s).

For sending mail from Noah

The needed setup is basicly installed in the above mentioned image. But it needs to be altered in /etc/postfix/main.cf in regards to *mydestinations (the interface ip of the server)* and *relayhost (a mail server for sending mails onwards)*.

Remember to get ports opened in firewalls for outside access.

Noah Zipfile

There are some requirements for Noah to run.
PHP5.3+ and Mysql 5+ is to be installed on an OS running Un*x variant or Windows (as always, Un*x prefered ;-)).
The following extensions to PHP is to be included in the installation:
- php5-common
- (xpdf-utils) pdftotext installed

- (php5-mysql) mysql for php
- (php5-snmp) snmp for php
- (php5-gmp) gmp for php

Optional:
→ (php5-ldap) ldap for php
→ (php5-imap) imap for php
→ (php5-cgi) cgi for php

After unpacking the zip file, there should be a directory structure looking like the following:

admin/
css/
-images/
extensions/
images/
import
javascripts/
- jqplot/
mail_parser/
orders/
scripts/
- crontab
stats/
storage/
system/
tickets/
upgrade/
var/
- upload/
- export_files/
- ndocs/
- user_pictures/
- styles/

The /var/ and /upgrade/ directory need to be writeable from the webserver (Lighttpd/ Apache/IIS aso.).

With the Licensekey, a initial sql script will be supplied. It must be imported into the mysql database.

Upgrade

Upgrading Noah is fairly easy. Download the latest zipfile from gnf.dk to your "noah root" directory. While logged in into Noah, unzip the zipfile. Then run the upgrade.php script :(example)

#cd /var/www/upgrade

#/usr/bin/php upgrade.php
#chown -R www-data:www-data /var/www/

Make sure php5-cgi is installed for this (#apt-get install php5-cgi).

Activating the "crontab_check_upgrade.php" in the /scripts/crontab/ directory, gives an semi automatic upgrade.
The script will check for an upgrade from http://gnf.dk and if existing, downloading it to the "var" directory.

In Administration / Upgrade Noah, a link will appear to the upgrade script. This will upgrade Noah unpacking the zipfile - overwriting all existing files with new ones - and upgrading the Database to the latest version / revision.

It is also possible to use the button "Download Upgrade" to download the current upgrade from gnf.dk.

Clicking the button "Unpack and Upgrade" will unzip the latest downloaded upgrade file and concurrently run the database upgrade script. (requires superuser rights for the /upgrade and /var directories)

!IMPORTANT!

Rights must be set correct for the upgrade to work.
the "var" and "upgrade" paths needs to be writeable for the www-data user.
To make sure this is so, use the two following commands (Debian):

cd /var/www/
*chown -R www-data:www-data **
chmod -R 777 var/ upgrade/

For some scripts to work, it is needed to set up a scheduler (unix: crontab) to run php with the scripts in the scripts/crontab directory. See Scripts running continuing for further information.

Administrator

The system has some defaults included. These includes are CPU, OS (OperatingSystem), Circuit, Password, Asset Types and Asset Status. It is advised to review them for an update to match your environment.

Default setup

The user "Administrator" ("admin") is installed with a default password ("password"). It is of course advised to alter this password to a strong password that you may remember (eg.: http://ismypasswordstrong.com/).

Reset of the system is done by using MySql to clear they system with importing the /install/ Noah-install.sql. It will drop the tables before inserting new ones.

!! The system will be reset to defaults. !!

Alternative, using MySql, update the Administrator password (not loosing your data)!
mysql> UPDATE users SET password=PASSWORD('newpassword') WHERE user='admin';
mysql> FLUSH PRIVILEGES;

Configurations (Administrator Menu)

System License(s)

For the system to work with the most basic functions, you must enter a license key for the system into the System License(s).

You may receive a license key from gnf.dk based on number of active users or active assets.

Core Settings

Various settings may be altered in this part of Noah. Remember, they alter the way Noah works and looks for the users!

SuperUserEmail is to be the emailaddress that will be used in case of systemerrors or other forms of needed information..
SupportEmail is to be the emailaddress that will be used in conjunction with ITS, receiving emails via crontab_imap_ticket.php script (see Crontab Scripts) or sending emails to users/ contacts to be able to reply to.

ExpirePeriod is the amount of days a users password is valid. If set to "0" (zero), users will not be reminded of the need to change password. Their current password will not expire. It does not remove the ability to change the password.

Noah has the option at login to save login credentials in a cookie on the client pc. This is enabled by "Remember Me" option.
The time to live off the cookie is optional changeable in the Core Data Section. Default is 10 (ten) days.

Defaults:
Core Version : '1.0',
Expire Days : 90,
SuperUserEmail : 'info@gnf.dk',
Cookie Timeout (Days) : 10

An option to get your own logo in the upper left corner (max. height 55px) is also available. Select the image to use in the Custom Logo file chooser.

LDAP

If it is needed to authenticate users to an ldapserver/AD server, use this option.
The options are mandatory for assisting the DAP Host field.
If the LDAP Host field is saved, Noah will try to authenticate to that host. Use the Test LDAP button to test the connection and options.

LDAP Options

LDAP Host (ip or dns)	Give the IP or DNS name of the ldap host/AD server
LDAP Domain (dn=)	This is the domain authentication is done towards. Omit the dn=
LDAP Admin (cn=)	The Admin user of the AD that gives access to authentication of users. Omit the cn=
LDAP Password	Above Administrators password.

IMAP

It is possible to receive emails to a specific imap account and based on the subject, create/ update an ticket.

Subject line has the following options: ticket id, asset name, circuit name

IMAP Host	Give the IP or DNS name of the IMAP host
IMAP Domain	Domain servers usually require a domain.. But some also requires the username in same setting. eg.: tDomain/tUsername If so, enter domain/username here and omit the username in the IMAP User field
IMAP User	Some IMAP servers requiers username@domainname.com instead. If

	so, enter your data here and omit the domain name in the previous IMAP Domain field.
IMAP Password	User password
Ticket Indicator	Sending an email to the above user (email address) it is possible for the sender to include the Ticket ID, Asset Name and/or Circuit Name in the subject. To help Noah which part is the ticket number, this field sets default IN# as Ticket Number. That is, the sender can include IN#nnn as the ticket number his/her request is regarding. The mail will be
Everyone allowed?	This field is basicly a restrictor. Set, it allows everyone to send a mail to the IMAP User and be handled by Noah. Unset, Noah will only allow registered contacts emails to be accepted.

If an attachment is enclosed the email, it will be moved to the var/upload directory with the ID as given in the database. If the "from:" email address is connected to a customer - by the domain name after the @, or the Ticket is an RFC, the attachment will also be related to the customer found.

Rights

User is given rights depending on which files the user should have access to. It is possible to assign same rights to groups and the grouprights will be propagaited to the user. So a user has his own rights and the rights connected to the group(s) the user is part of. In the rights view, rights inherited from groups are marked with green.

Administration (Administrator Menu)

This is the part where Noah is altered to suit your setup in terms of usage. That is, you can add/remove the different types used in the system.

After an update, some defaults may vary from the original. Eg., please check the different types for additions (both in the manual but also in your own system.)

Example : upgrading v1.30 to v1.31, cpu_cores are added as a field. But it is not possible to update something that is not there as a default and you may have added cpu's that are unknown to Noah at the time of update.

Asset Status

Default status types is:

'Unknown' : Unknown State
'Switched off' : Asset has no Power
'Ready for deployment' : Asset is ready for OS
'In Deployment' : OS and/or applications are being implemented
'Ready for operation' : The Asset is awaiting the final acceptance test
'In operation' : Asset is used and in production
'Under decommission' : Asset is being taken down for storage or other use.
'Not in operation' : Asset has Power but is not in use

Asset Types

Default Asset Types is:

'Cluster' : The Cluster Virtual Master
'DiskSystem' : SAN System
'Firewall' : A physical Firewall
'VFirewall' : A virtual Firewall
'Server' : Definition of a server and its fields
'Switch' : Network switches
'Router' : Network routers
'Director' : Storage switches
'Virtual Machine' : A virtual machine
'Storage Shelf' : Unit which contains spindels
'Storage Controller' : Unit which controls mdiscs and spindels

It is possible to delete, edit and create new Asset Types. It is possible to choose which element should be a part of the type. Selections is limited to:

- Administrative
- Licenses
- Passwords
- Local Storage
- External Storage
- HBA Interfaces
- Network Interfaces
- Files
- Domains

These are the parts showing in ViewAsset and EditAsset pages.

Circuit Types

'MPLS', 'MPLS connection'
'2MB ADSL', '2 Mb Adsl connection'
'VPN', 'Vpn Connection'

CPU Types

Following cpu types is examples. It is strongly advised to alter the types to current cpu types.
'Amd X64 QuadCore 3Ghz'
'Amd X64 QuadCore 2Ghz'
'Pentium4 QuadCore 2.3Ghz'
'Intel(R) Xeon(TM) CPU 4.20GHz'
'Intel(R) Xeon(TM) CPU 3.20GHz'

Tickets

Ticket Categories

'Order', 'Order from the Order System'
'Software', 'The software is the problem'
'Hardware', 'The Issue is hardware related'

Ticket Types

'Storage Order', 'Order of Storage'
'Order Backup', 'Backup Order'
'Server Order', 'Order of Server(s)'
'Customer Created', 'A customer has called in and claimed there is an issue'
'Surveilance', 'Surveilance has tagged an ticket'

Ticket Sub Types

It is possible to add subtypes to Ticket Types.

Password Types

'ILO', 'ILO password'
'System', 'Low level password'
'SNMP', 'Snmp password per asset'

OS Types

Default OperatingSystems is :

AIX / AIXL
AmigaOS
BSD
Linux
Linux Caldera
Linux Corel
Linux Debian
Linux Kondara
Linux Mandrake
Linux Red Hat
Linux Slackware
Linux SuSE
Linux Turbo
Linux Vector
DUnix
DYNIX/ptx
HP-UX
IRIX
MAC OS 8
MAC OS 9
MAC OS 10
MAC OS X
MINIX
MS-DOS 1.x
MS-DOS 2.x
MS-DOS 3.x
MS-DOS 4.x
MS-DOS 5.x
MS-DOS 6.x
NEXTSTEP
OSF/1
QNX
SCO
Sun Solaris
System 1
System 2
System 3
System 4
System 6
System 7
System V
Tru64 Unix
Ultrix
Unisys
Unix
UnixWare
Windows 2000
Windows 2003
Windows 2003 Web
Windows 2008
Windows 2008 R2
Windows 3.x

Windows 7
Windows 95
Windows 98
Windows CE
Windows ME
Windows NT
Windows Vista
Windows XP
Xenix

User Groups

'Network', 'NW', 'Network architecture'
'Sales', 'SS', 'Sales People'
'Storage', 'ST', 'Personel with responsibility for Storage'
'Infrastructure', 'INF', 'People with responsibility for Infrastructure'
'Deployment', 'DP', 'Deployment people'
'Finance', 'Financial People'
'Administration', 'Administrative People'

Users

The AssetID

Noah is built around the idea that any asset has a unique identifier. We call it an asset id (AssetID). It is a number starting at one and ending at (insert a very large number here). *It is common to print the asset id on an adhesive sticker - some print it with barcodes for easier registration using a barcode reader.*

Noah has an export function where it is possible to export the assetids to csv. And from excel - or any other compaitble application - print to adhesive stickers.

The AssetID is the focal point of the system. Every asset in the system has an AssetID. So it is always possible to find the asset just by searching for the AssetID or locating the asset within the customer it is owned by.

Deleted Assets/Customers/Users

Why is it not possible to delete an asset or a customer? Well, the asset- or customerhistory will be lost for the deleted asset or customer. So is the relation to the tickets (tickets). "Deleted Assets" (with CustomerID 1) is a pre-built customer that will hold all deleted asset. So assets never "die". They will just face a long period of "storage" or recycling by moving the asset to another customer.

Users and Customers can be Deactivated / Activated. They cannot be deleted. The reference to the assets and its tickets and history would be lost.

Deactivated Assets, Customers or Users will be marked with the color Red or just be placed under Deleted Assets or Inactive Customers.

Cluster(s)

Setting the AssetType as a Cluster (in EditAsset) gives the option to use the asset as an cluster master, created in Clusters menu.
EditCluster under View Customer/Cluster gives the option to add shares dedicated to the cluster. It is also possible to see the nodes attached to the cluster.

Other ID´s

As an addition to the system, Circuits, Domains and Contacts among others have been added to the system. They are connected with a customer and so they have their own "ID". But it is still possible to view them from the ViewCustomer view or search for them. There might be further additions of those kind of modules in future releases of Noah.

IPv4

Assets can have an IP address (http://en.wikipedia.org/wiki/IPv4) attached. Or several for that matter. It is possible to use the CIDR annotation to assign routed networks to an interface. Interfaces has a list of those IP´s assigned to the assets owner.
An owner can have an IP segment assigned. The interfaces on the assets can assign the ip´s owned.
If an ipsegment is marked as *Private,* it is only possible to assign from the segment on assets assigned to the customer whom owns the ipsegment. If not marked *Private*, subcustomers will also have the option to get assigned ip's from the segment..

IPv6

The IP address field is ready for IPv6 and accepts such an address. No calculations will be done on the field. Only validation.

ITS (Incident Ticket System)

It is possible to keep track of tickets concerning assets. Above each asset and customer is a link (icon) to create ticket and one to view tickets.

Ticket Overview

The frontpage of Tickets gives an overview of RFC's, Opened, Active and Closed Tickets.
It is possible - from the frontpage - to goto Update Ticket by clicking the Update Ticket link (icon).

Default it will show tickets that is assigned directly to the current user and the groups the user is assigned to. It is possible to view another user or another group tickets (selected via pulldown).
Clicking "My Tickets" alters the view to current user/group tickets.

The ITS frontpage is reloading every minute.

Create Ticket

An ticket must be connected to a customer. Ticket "Title" and "Description" is required.

Create Followup

Creating an Followup to an ticket is done by clicking the Update Ticket link (icon:) and fill out the FollowUp field. After that you need to assign the followup to a user or group. Or simply Close the Ticket.
If it is you that has created an followup, you are allowed to delete or edit the followup. Use the links(icons: or) in the followup for this.

Link to Followup

Optionally it is possible to link to an followup by clicking on this image on the followup :
That will give the complete url to the ticket and followup.

Search

The search function is split up in different "areas" for optimization purposes. Default, Assets and Customers are marked for search areas. Deselecting all but Assets will make the search go faster and focused on only the assets database.

Assets

The basis for the system is the asset id. The asset id is given by the system. Almost every aspect of the system is based on the fact that users update the assets database and keep the information as correct as possible.

Create Asset

You only need a name for the asset. The asset must be assigned to a customer. The system is assigning an assetid for the asset. At creation of the asset, the system will redirect to Edit Asset (see below) for altering the rest of the asset information. On this page - in the top - you can see the newly assigned AssetID.

Edit Asset

With an asset comes many options. It may be the physical attributes like weight, power usage, location or the ip numbers attached, the storage connect aso.

This is where you edit those settings. A lot of work is done to make it easy to understand what and how it is supposed to be entered. For a more details about editing an asset, see the User Guide.

View Asset

This is where you get all of the information regarding an asset. Remember, there is an - almost- hidden field - "Passwords". Click to view them. (Not everybody looking at your screen need to know the specific passsword(s) to any given asset.

System Menu

Adding location(s), creating an order for Storage or create an extraction of data. Basic functionality is placed here.

Orders

It is possible to create orders for Storage and Backup. That way it is made sure that the right information goes to the groups that is to supply the storage or backup. Upon creation of the order, an ticket is created.

Create Storage Order

Start by choosing the customer and asset that is to receive the storage. Then enter placement {Archive, Best Performance}, size (in Gb), storage type {SQL, Exchange, Files, Backup, Mixed}
Optionally a reference to an ticket and a comment can be given.
Choose user or group to assign order to.

Create Backup Order

Start by choosing the customer and asset that is to be backed up. Then enter the IP that is to be connected to, expected size of backup (size in Gb), FQDN (Fully Qualified Domain Name) and backup type {SQL, Exchange, Files}
Optionally a reference to an ticket and a comment can be given.
Choose user or group to assign order to.

Create Server Order

Choose the customer that need the server assigned followed by specification for the server.
Optionally a reference to an ticket and a comment can be given.
Choose user or group to assign order to.

Disksystem and Diskgroups

Assigning Lun's or Vdisk's is started with an asset that is defined as a disksystem (Asset Type). After this is created, it is possible to assign diskgroups to this asset in the Storage menu. This is also where you may list the disksystems with attached storage guests (Lun or Vdisk assigned.)

Documents

In the Documents part of Noah, it is possible to create documents, attach files to the documents and folders. It is also possible to create/edit folders and put documents and attached files into the folders.

Documents will be placed in the table ndocs_documents. HTML signup language is possible.

Uploading of files:

Files will be placed in the folder *var/upload/ndocs* with the ID set in the table *ndocs_files*. As a result, it is listed as "id.puhaa" in the folder. The file is not altered in any other way than changing the name. The original filename is saved in the record *file_name* in the table *ndocs_files.*

Extractions

Some pre-defined extractions are created. Combine them with further selections and create specific extractions. Save the extractions for later re-use. The result of the extraction will be saved as a CSV file. Perfect for importing into eg. "Excel" or "Calc" for further manipulating.

Other Options

MailParser

MailParser.php is only functional in cooperation with an mailforwarder - eg. PostFix w. procmail (system filtering for mailaddresses. See setup example below). Sending an email to any given emailaddress and making the forwarder point the mail forward to MailParser.php enables creating tickets via emails. It will check for authorized emailaddresses (contacts aso.) and checking for ticket number in the Subject.
An alternate for this is the IMAP setting in Core Settings.

Example of the setup of the mail parser with maildrop:

It has been tested from scratch with the following howto:
http://www.postfix.org/MAILDROP_README.html

After installing postfix with *apt-get install postfix* and maildrop with *apt-get install maildrop*

You need to set the shell to "bin/false" for the user you will use as email address (in this case we tried with "ticket@localhost":
ticket:x:1001:1001:,,,:/home/ticket:/bin/false

Create a ".forward" file in /home/ticket/, and add the following to the file "*/home/you/.forward*":
"**|/usr/bin/maildrop -d ${USER}**"

then changen rights for the file:
chown ticket:ticket .forward
chmod 755 .forward

Example of the setup of the mail parser with procmail:

Put this in the file /etc/procmailrc

```
:0:
* ^To:.*ticket
|/usr/share/mailparser.php >> output.txt
```

the file output.txt will be the "log" file for the parser.

Put this in the file /etc/postfix/master.cf

```
procmail  unix  -  n  n  -  -  pipe
 flags=R user=vmail argv=/usr/bin/procmail -p -t -m  /etc/procmailrc $sender $recipient
```

The Mailparser.php Script

If you would like to receive email to a specific email address and make a ticket based on the subject, use the mailparser.php api. Mail parser has it's own config.php file since it may be used on a mailserver instead of the webserver Noah is running on. If so, make sure that port 3306 (mysql) is open from the mailserver to the webserver (database server is this is seperated from the webserver). Config.php should have the same data as the Noah config.php file.

The mailparser.php filters data based on the following:

From:

If the email-address is in contacts, it is accepted. Otherwise a mail will be returned to the From address stating inability to accept email. So remember, only Contacts' emails will be accepted!

Subject:

If the subject starts with the string stated in config.php (in the mail-parser directory -[*$sub_ticket = "IN#";*] in this case '*IN#*' then the script will try to create a followup with the body of the mail as the data. If not, it will create a new ticket and stop there.

Included

In the subject: If a number is present (other than the IN# number) Noah will try to locate an AssetID or CircuitID with that number and connect the found with the ticket..

Scripts running continous

In lack of a better name, it is supposed to run in an interval by eg. crontab on Un*x or AT on windows OS. Or any other scheduler available. In the image delivered (debian) it is setup in crontab running every day at 8 am.

crontab_master - The script that checks for the schedule of other crontab scripts!
(Located: /scripts/crontab/crontab_master.php)
This script checks for the interval and evt. time set in Core Data in the Cron Scripts part. It is possible to set if the script should be active or not. Also which interval (time and type) it should be run at.

crontab_get_Snmp_data
(Located: /scripts/crontab/crontab_getSnmp_port_data.php)

It runs through the database searching for assets with snmp passwords enabled on an interface. If so, it tries to get access to the asset (through the IP address registered to the interface) and fetch port information into the "ports" and "asset_port_rel" tables.

crontab_agreement_warning

(Located: /scripts/crontab/crontab_agreement_warning.php)

The script checks the agreements expire date (agreement_end_date) against the current date.

If the expire date is less than 31 days, an information is mailed to the responsible user.

If the expire date is less than 7 days, a warning is mailed to the responsible user.

If the expire date is less than 1 day, an alert is mailed to the responsible user.

crontab_create2D

(Located: /scripts/crontab/crontab_create2D.php)
The script selects data for every asset and creates an 2Dbar image. It is placed in *var/upload* and filenames are 2d_asset<assetnumber>.png.

crontab_check_upgrade.php

(Located: /scripts/crontab/crontab_check_upgrade.php)
The script is run once a day (8 am) to see if there is a new version/revision ready for download on http://gnf.dk.
If so, it downloads it to /var/Noah_upgrade.zip and unzips the /upgrade/upgrade_data.php into /var directory.
After that, if an admin user enters core admin segment in Noah, an link for upgrading will be shown.

crontab_check_breached_tickets.php

(Located: /scripts/crontab/crontab_check_breached_tickets.php)

The script selects tickets that has customer- or asset sla breached.
If the breached ticket has an followup, it will be registered as an end of the breach and it is possible to see the breached ticket in the SLA view.
See SLA for further info on SLA.

crontab_imap_ticket

(Located: /scripts/crontab/crontab_imap_ticket.php)
The script checks the imap host/user given in CoreAdmin for emails. If an email is valid it is checked for ticket number, asset name and/or circuit name. An ticket is created and the mail is deleted.
If DispatchEmail and DispatchTags are valid, it will match Subject in incomming mail with DispatchTags. A match forwards the mail to DispatchEmail.

Crontab Example

In the */scripts/crontab* directory some crontab templates are available.

Example from gnf.dk: (crontab -l shows crontab_master which runs once every minute)

```
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

1-59 * * * 1-7 /usr/bin/php /web/noah/scripts/crontab/crontab_master.php  >> /var/log/noah
```