

The Opmantek Installer



Opmantek Installer

All Opmantek products make use of an interactive installer program that greatly simplifies both initial installation AND upgrading an existing installation.



As at 1 October 2020, the latest versions of opCharts, opConfig, opEvents and opReports, include a new build system which is not binary compatible with versions released before this date. When upgrading OMK Applications released before 1 October 2020, you will need to upgrade all products to the latest version.

This document explains the most essential installer features.

- [Installation of Opmantek Applications](#)
 - [Prerequisites](#)
 - [Opmantek Applications Download Formats](#)
 - [Preparation: Is /tmp directory mounted noexec and/or is fapolicyd service installed and running?](#)
 - [Debugging PAR Script Unpack Locations](#)
 - [Debugging fapolicyd](#)
 - [Starting the Installer](#)
 - [Available Installer Options](#)
 - [Logs and Backups](#)
 - [Software Dependencies](#)
 - [Product Coexistence, Migration and Upgrades](#)
 - [Integration and Initial Configuration](#)
 - [FAQ](#)
- [Upgrading NMIS 9 Compatible Opmantek Applications](#)
- [Uninstalling Opmantek Applications](#)
 - [Using the Uninstaller](#)
 - [Manual Removal](#)

Installation of Opmantek Applications

Prerequisites

- With the exception of Open-Audit, which can be installed on Windows Server or Linux, all Opmantek applications are available for Linux 64-bit systems only.
Redhat/CentOS6, Debian 7, Ubuntu 10 and newer are supported (basically anything running glibc 2.3 and up).
- The latest versions of our applications can be found at: <https://opmantek.com/network-tools-download/>
- To run the installer you need superuser/root access on the system in question.
- Please note that as of Feb 2017, all Opmantek applications require that /tmp is mounted with execute permissions (i.e. mounted *without* the `noexec` mount flag).
See below for an alternate procedure if your /tmp is non-executable.

Opmantek Applications Download Formats

In the past our applications were provided in the form of a compressed tar file, which required some manual steps for unpacking and installer invocation. As of February 2016 we've switched to a self-extracting download format which makes this aspect much more user-friendly.

All Opmantek product releases from 16 June 2016 onwards include the installer in both pre-compiled and source form, to ensure that you can install the software on a system without Perl present. The source form of the installer is provided for diagnostic purposes; by default the self-extracting run file will start the pre-compiled installer version.

When you download an Opmantek Application, the file will be called `<product name>-<version>.run` and your browser will likely prompt you regarding what to do with this '.run' file; you should tell it to Save the file. If you are installing the application onto a different system than the one where you downloaded the file, you'll have to use `scp` or some other file transfer method of your choice to transfer the .run file to the target system.



Best Practice

As a best practice, Opmantek recommends you create a dedicated directory, perhaps named "installs" to download and run the installer from. If you are using the Opmantek VM we recommend creating this folder in `/data/installs/`

Preparation: Is /tmp directory mounted noexec and/or is fapolicyd service installed and running?

The OMK and NMIS installers will not execute successfully unless it is able to unpack itself and is then able to execute the ./pre-install.sh script within the unpacked directory:

```
sudo ./opReports-4.2.2-test-noexec.run
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Uncompressing opReports 4.2.2 100%
./opReports-4.2.2-test-noexec.run: line 655: ./pre-install.sh: Permission denied
```

Here is the method employed in testing to get OMK Applications functioning correctly in these restrictive environments:

- **General Changes required by both fapolicyd and noexec mounted /tmp (required by OMK Installers, but not by NMIS Installers):**

```
# Create an install directory which we can use in a less restricted fashion to get OMK Applications'
installed and functioning
# - for this example we have chosen directory '/data/installs/':

sudo -i
mkdir -p /data/installs/

# Set an environment variable globally to accomodate Perl::PAR module, which is used to create and
execute OMK daemons and scripts, if needed:
# check the environment variable is not already set (should not return reference to PAR_GLOBAL_TMPDIR if
not set in this file):
cat /etc/environment

# check the environment variable is not already set in some other manner (should not return anything if
PAR_GLOBAL_TMPDIR is not already exported):
echo "${PAR_GLOBAL_TMPDIR}"

# set up PAR_GLOBAL_TMPDIR if needed:
#
# IMPORTANT:
#       If you require OMK scripts, that don't explicitly require root privileges, able to be
executed by users without root privileges:
#       Set PAR_GLOBAL_TMPDIR to a suitable directory outside of the OMK install
directory (/usr/local/omk in this example);
#       The OMK install directory currently has root as both owner and group by default.
#       Users without root privileges won't therefore have execute capability in the
OMK install directory structure.
#
mkdir -p /usr/local/omk/var/lib/common/
chmod 1777 /usr/local/omk/var/lib/common/
echo 'PAR_GLOBAL_TMPDIR="/usr/local/omk/var/lib/common/"' >> /etc/environment

# Fix perl warning: Please check that your locale settings:
#
#       LANGUAGE = (unset)
#       LC_ALL = (unset)
#       LANG = "en_US.UTF-8"
#       are supported and installed on your system.
. /etc/environment;{ [[ -z "${LANG:-}" ]] && echo "LANG=en_US.utf-8";[[ -z "${LC_ALL:-}" ]] && echo
"LC_ALL=C"; } >> /etc/environment

# To allow group omkadmin to execute OMK scripts using PAR_GLOBAL_TMPDIR we need to set permissions on
/usr/local/omk/var directory to allow this:
sudo chmod 0750 /usr/local/omk/var;
sudo chown root:omkadmin /usr/local/omk/var;

# To allow user nmis to to execute OMK scripts using PAR_GLOBAL_TMPDIR we add user nmis to group
omkadmin:
sudo usermod -a -G omkadmin nmis;

# reboot to get the PAR_GLOBAL_TMPDIR exported globally if it needed to be set:
reboot
```

```

# check PAR_GLOBAL_TMPDIR is exported after reboot:
echo "${PAR_GLOBAL_TMPDIR}"
/usr/local/omk/var/lib/common/

# Unfortunately systemd services do not pick up this global environment variable, so each OMK systemd
service needs to be edited:
# first we check the needed 'EnvironmentFile' entry is not already included with:
sudo systemctl cat omkd
# then, if necessary, edit omkd service
sudo systemctl edit omkd

# Ensure the service is configured to use PAR_GLOBAL_TMPDIR environment variable as set in /etc
/environment
# by adding the following entry to [Service]
# - add [Service] section if it is not already present:

[Service]
EnvironmentFile=/etc/environment

# edit each OMK systemd service in this manner if needed, for example:
sudo systemctl edit opchartsd
sudo systemctl edit opconfigd
sudo systemctl edit opevents
sudo systemctl edit opflowd

# NMIS9 plugin SubnetImport.pm, which is executed by nmis9d.service daemon, executes opcharts-cli.pl so
this service too
# needs to be configured to have PAR_GLOBAL_TMPDIR environment variable in its environment, as set in
/etc/environment:
#
# first we check the needed 'EnvironmentFile' entry is not already included with:
sudo systemctl cat nmis9d
# then, if necessary, edit nmis9d service
sudo systemctl edit nmis9d

# Ensure the service is configured to have PAR_GLOBAL_TMPDIR environment variable in its environment, as
set in /etc/environment
# by adding the following entry to [Service]
# - add [Service] section if it is not already present
# - note the '--' which instructs nmis9d.service not to fail on file /etc/environment not being found:

[Service]
EnvironmentFile=-/etc/environment

# reload the edited services
sudo systemctl daemon-reload

# restart the OMK services
sudo /path/to/omk/bin/checkomkdaemons.sh restart

# restart the nmis9d service
sudo systemctl restart nmis9d

# To ensure cron jobs read /etc/environment and pick up the environment variable
PAR_GLOBAL_TMPDIR,
# prepend the following code to the command:
export $(/usr/bin/xargs < /etc/environment) | :;
# For example /etc/cron.d/opreports:
# was
#
# this cron schedule runs the opReports scheduler every 5 minutes
#
# m h dom month dow user command
*/5 * * * * root /usr/local/omk/bin/opreports-scheduler.exe
# and becomes

```

```

# this cron schedule runs the opReports scheduler every 5 minutes
#
# m h dom month dow user command
*/5 * * * * root export $(/usr/bin/xargs < /etc/environment) | :; /usr/local/omk/bin
/opreports-scheduler.exe

```

- **fapolicyd Whitelisting Change (required by OMK Installers, but not by NMIS Installers):**

- # For setting new fapolicyd rules, please read:
<https://www.mankier.com/5/fapolicyd.rules>

```

# For OMK services and scripts to function correctly we will need to add a rule to whitelist needed
directories in fapolicyd
# such that root (uid=0) can execute scripts in the listed directories:
# - for this example we have chosen directory '/data/installs/' and /path/to/omk/ is /usr/local/omk/:

# Insert the following block of 2 rules immediately after the '%languages=' entry, making this the first
rule in /etc/fapolicyd/fapolicyd.rules.
# See the paragraph further below 'Debugging fapolicyd' for the method used to identify the additional
two 'Fix' rules for RHEL8 in this case.
# The additional 'Fix rule' may have a different file paths from those in this case.
# Please do follow the steps in paragraph further below 'Debugging fapolicyd' and debug for at least a
week to ensure your fapolicyd implementation is complete!

# Allow Installs:
allow perm=any uid=0 : dir=/data/installs/
# Allow NMIS9:
# Fix dec=deny_audit perm=open auid=-1 pid=1931352 exe=/usr/local/nmis9/bin/wmic : path=/usr/lib64
/libresolv-2.28.so ftype=application/x-sharedlib
allow_audit perm=any uid=0 : path=/usr/lib64/libresolv-2.28.so
# Allow OMK PAR:
allow perm=any uid=0 : dir=/usr/local/omk/bin/,/usr/local/omk/script/,/usr/local/omk/var/lib/common/,
/data/omk/var/lib/common/,/usr/local/omk/lib/common/PAR/
#
# Fix "dec=deny_audit perm=execute OR open auid=-1 pid=108878 exe=...opmantek.pl : path=/usr/lib64/ld-
2.28.so ftype=application/x-sharedlib"
# Fix "dec=deny_audit perm=open auid=-1 pid=27086 exe=/usr/local/omk/bin/opa-cli.exe : path=/usr/lib64
/libpthread-2.28.so ftype=application/x-sharedlib"
# Fix "dec=deny_audit perm=open auid=-1 pid=27086 exe=...opa-cli.pl : path=/usr/lib64/libnsl.so.2.0.0
ftype=application/x-sharedlib"
# Fix dec=deny_audit perm=open auid=-1 pid=809147 exe=/data/omk/var/lib/common/par-726f6f74/cache-
17e3f7c595dddb09f71069f2282bdbb1943a69e3/opreports-scheduler.exe
# : path=/usr/lib64/libnsl-2.28.so ftype=application/x-sharedlib
# Fix dec=deny_audit perm=open auid=-1 pid=2306975 exe=/usr/local/nmis9/bin/wmic : path=/usr/lib64
/libcrypt.so.1.1.0 ftype=application/x-sharedlib
# Fix dec=deny_audit perm=open auid=-1 pid=1680601 exe=/data/omk/var/lib/common/par-726f6f74/cache-
17e3f7c595dddb09f71069f2282bdbb1943a69e3/opreports-scheduler.exe
# : path=/usr/lib64/libdl-2.28.so ftype=application/x-sharedlib
# Fix dec=deny_audit perm=open auid=-1 pid=1749856 exe=/data/omk/var/lib/common/par-726f6f74/cache-
00548e237c0c0fdd9581d8236e7b57e47c9024b4/opa-cli.pl
# : path=/usr/lib64/libm-2.28.so ftype=application/x-sharedlib
# Fix dec=deny_audit perm=open auid=-1 pid=2211812 exe=/data/omk/var/lib/common/par-726f6f74/cache-
15aafc77ad1c5493483a4c1937bb5e81cc946bd0/patch_access.exe
# : path=/usr/lib64/libutil-2.28.so ftype=application/x-sharedlib
# dec=deny_audit perm=open auid=-1 pid=2306975 exe=/data/omk/var/lib/common/par-726f6f74/cache-
17e3f7c595dddb09f71069f2282bdbb1943a69e3/opreports-scheduler.exe
# : path=/usr/lib64/libc-2.28.so ftype=application/x-sharedlib
allow_audit perm=any uid=0 : path=/usr/lib64/ld-2.28.so,/usr/lib64/libpthread-2.28.so,/usr/lib64/libnsl.
so.2.0.0,/usr/lib64/libnsl-2.28.so,/usr/lib64/libcrypt.so.1.1.0,/usr/lib64/libdl-2.28.so,/usr/lib64/libm-
2$
# Fix persistent issues where above 'Fix' strategies aren't working 100% with 'trust=1' strategy: Add
each script to trust with following commands
# which need to be run after each applicable OMK App install and OMK App upgrade:
# fapolicyd-cli --file add /usr/local/omk/bin/opa-cli.exe
# fapolicyd-cli --file add /usr/local/omk/bin/opreports-scheduler.exe
# fapolicyd-cli --file add /usr/local/omk/bin/opreports-scheduler.exe
# Fix dec=deny_audit perm=execute auid=-1 pid=2183440 exe=/usr/sbin/logrotate : path=/usr/lib64/ld-2.28.

```

```

so ftype=application/x-sharedlib
#       fapolicyd-cli --file add /usr/sbin/logrotate
# Update the fapolicyd trust database after making any change using `fapolicyd-cli --file ...`
#       fapolicyd-cli --update
# Check files in the fapolicyd trust database with this command:
#       fapolicyd-cli --dump-db|grep -v rpmdb

# Update fapolicyd with the additional rule we have inserted:
sudo fapolicyd-cli --update

# Reboot at this point is not absolutely necessary, but reinforces that settings are working as intended
sudo reboot

# restart the OMK services
sudo /usr/local/omk/bin/checkomkdaemons.sh restart

```

- **noexec mounted /tmp Change (required by OMK Installers and NMIS Installers):**

```

# Please read the next paragraph 'Starting the Installer' too, for more details on the installer!

# The Opmantek installers cannot install from a noexec mounted directory, so we need to install from an
alternative location:
# - for this example we have chosen directory '/data/installs/':
sudo mkdir -p /data/installs
cd /data/installs

# We need to instruct only this command we are about to execute with current directory set as
environment variable TMPDIR:
# Place the installer in the current directory, then ..

# This command as given sets TMPDIR to the current directory (not /tmp/) - there is a space character
between the two stop characters in 'TMPDIR=. ./'
# TMPDIR=.<space>./

TMPDIR=. ./opReports-4.2.2-test-noexec.run

# To be safe we ensure TMPDIR is unset after successful execution of the installer by executing this
command:
unset TMPDIR

# The installer will now install using current directory as /tmp
# Unfortunately when the installer runs in this way, not using /tmp/ directory, it doesn't clean up
after itself, so we clean up manually:
ls
opReports-4.2.2-test-noexec.run
selfgz3021223337

# The installer always unpacks to a directory of glob pattern selfgz*, so we clean up by removing
directory selfgz3021223337:
rm -rf selfgz*

```

Debugging PAR Script Unpack Locations

Bash one-liner command to find the scripts executing as PAR executables on a server and where their PAR unpacked directory is located:

```

sudo find / -type f ! -name 'main.pl' -regextype posix-egrep -regex '^.*/par-[^/]+/cache-[^/]+/inc/script.*' 2>
/dev/null|xargs -I{} sudo ls -lA '{}';

-rw-r--r--. 1 root root 37741 Feb  7 15:09 /tmp/par-726f6f74/cache-
5372daa5f1e09cab20da623efbb20c3c4f14bla8/inc/script/opcharts-cli.pl

# Using this approach I tracked down the above execution of opcharts-cli.pl to be by an NMIS9 plugin
SubnetImport.pm, which is executed by nmis9d.service daemon

# Unfortunately systemd services do not pick up this global environment variable, so nmis9d service needs to be
edited:
# first we check the needed 'EnvironmentFile' entry is not already included with:
sudo systemctl cat nmis9d
# then, if necessary, edit nmis9d service
sudo systemctl edit nmis9d

# Ensure the service is configured to use PAR_GLOBAL_TMPDIR environment variable as set in /etc/environment
# by adding the following entry to [Service]
# - add [Service] section if it is not already present
# - note the '--' which instructs nmis9d.service not to fail on file /etc/environment not being found:

[Service]
EnvironmentFile=/etc/environment

# reload the edited service
sudo systemctl daemon-reload

# restart the nmis9d service
sudo systemctl restart nmis9d

```

Debugging fapolicyd

Stop the fapolicyd service and run fapolicyd in the foreground with '--debug-deny' parameter:

```

sudo systemctl stop fapolicyd

# For debugging fapolicyd, please read:
# https://www.mankier.com/8/fapolicyd
# ...
# --debug-deny
# leave the daemon in the foreground for debugging. Event information is written to stderr only when the
decision is to deny access.

sudo /usr/sbin/fapolicyd --debug-deny

Loaded 16 rules
Changed to uid 990
Initializing the database
fapolicyd integrity is 0
backend rpmdb registered
backend file registered
Loading rpmdb backend
Loading file backend
Checking database
Importing data from rpmdb backend
Importing data from file backend
Entries in DB: 28117
Loaded from all backends(without duplicates): 28117
Database checks OK
added /dev/shm mount point
added / mount point

```

```

added /var mount point
added /boot mount point
added /tmp mount point
added /data mount point
added /run/user/1000 mount point
Starting to listen for events
rule=15 dec=deny_audit perm=execute auid=-1 pid=2302 exe=/usr/local/omk/bin/opa-cli.exe : path=/tmp/par-726f6f74/cache-00548e237c0c0fdd9581d8236e7b57e47c9024b4/opa-cli.pl ftype=application/x-executable
rule=15 dec=deny_audit perm=execute auid=-1 pid=2303 exe=/usr/local/omk/bin/opreports-scheduler.exe : path=/tmp/par-726f6f74/cache-815c07b0877113fa7553963226f8855aall160121/opreports-scheduler.exe ftype=application/x-executable
rule=15 dec=deny_audit perm=execute auid=-1 pid=2306 exe=/usr/local/omk/bin/opa-cli.exe : path=/tmp/par-726f6f74/cache-00548e237c0c0fdd9581d8236e7b57e47c9024b4/opa-cli.pl ftype=application/x-executable
rule=15 dec=deny_audit perm=execute auid=-1 pid=2542 exe=/usr/local/omk/bin/baseline.exe : path=/tmp/par-726f6f74/cache-62f960e7d5fb11c6bcb34fba76fe5030b04477c/baseline.exe ftype=application/x-executable
rule=15 dec=deny_audit perm=execute auid=-1 pid=2695 exe=/usr/local/omk/bin/opreports-scheduler.exe : path=/tmp/par-726f6f74/cache-815c07b0877113fa7553963226f8855aall160121/opreports-scheduler.exe ftype=application/x-executable
...
...

# When finished debugging, press CTRL+C to kill this foreground fapolicyd process:
^C shutting down...

# I traced the above few issues returned while debugging to cron jobs not reading /etc/environment
# and therefore not picking up the environment variable PAR_GLOBAL_TMPDIR
# Here is the solution to this issue:
#
# To ensure cron jobs cron jobs read /etc/environment and pick up the environment variable PAR_GLOBAL_TMPDIR,
# prepend the following code to the command:
export $(/usr/bin/xargs < /etc/environment) ||:;
# For example /etc/cron.d/opreports:
# was
#
# this cron schedule runs the opReports scheduler every 5 minutes
#
# m h dom month dow user command
*/5 * * * * root /usr/local/omk/bin/opreports-scheduler.exe
# and becomes
#
# this cron schedule runs the opReports scheduler every 5 minutes
#
# m h dom month dow user command
*/5 * * * * root export $(/usr/bin/xargs < /etc/environment) ||:; /usr/local/omk/bin/opreports-scheduler.exe

# Restart the fapolicyd service when debugging is finished:
sudo systemctl start fapolicyd

```

Starting the Installer

Starting the self-extracting installer is trivial: you simply tell your shell to run it.

Assuming your downloaded file is called `opProduct-version.run`, you would do one of the following:

1. The simplest way to achieve this is to type `"sh ./opProduct-version.run"`
2. You can also modify the permissions of the `.run` file to indicate that it is executable, then start it directly
To do so, you'd run `"chmod u+x ./opProduct-version.run"` followed by `"./opProduct-version.run"`

The installer will first run an archive integrity check, decompress the archive, then start the interactive phase of the installation.

Please note that the installer needs to run with `root` privileges, and will terminate with an error message if this requirement is not met.

Available Installer Options

You can see an overview of the available options related to the self-extracting aspect when you start a run file with `--help`:

- `--keep` ensures the unpacked data left behind (in the `opProduct-version` directory) after the interactive installer component has finished.
- `--noexec` causes no interactive installer component to be run.

The combination of `--noexec` and `--keep` provides the equivalent of unpacking the tar files provided with earlier releases.

You can also pass options to the interactive installer component, but these must follow after a `--` delimiting argument:

- NMIS9 Compatible OMK Application Installers created on or after 2020-08-18 will include a new option `-D`: *Dependency Check Mode* to assist with [installation in a disconnected \(air-gapped\) environment](#):
 - When run with *Dependency Check Mode* enabled the installer will not perform an install, but write a file containing a list of dependencies to the `/tmp/` directory.
For example, `sh ./opCharts-4.1.1.run -- -D` run as a normal user, would start the installer in Dependency Check Mode (`-D`) and only create a list of dependencies at `/tmp/omk_dependency_check_opcharts` upon completion.
- If you want to perform a *simulation run* of the installation, use the `-n` option: the installer will only print what it *would* do, what files it *would* copy and so on, but will not perform any of these steps.
- By default the installer is interactive and will prompt you for decisions and confirmations; If you want to run it in *non-interactive batch mode*, use the `-y` option.
In this case all dialogs and prompts are automatically answered with the default answer (usually 'y').
- The `-p preseed-file` option, supports a feature for better non-interactive installations: answers to the installer's questions can now be preseeded before the installer is invoked. See [Smarter non-interactive installation with Preseeding](#)
- Please note that in non-interactive mode the installer will abort upgrades if critical incompatibilities (e.g. license type) are detected; the option to overrule the installer in such situations is only present when the installer is running interactively.
- Certain installer choices can be preset for non-interactive mode:
 1. **Setting the environment variable `NO_LOCAL_MONGODB` to a non-empty value instructs the installer to not install a local MongoDB server even if none is present.**
Please note that you will have to manually adjust the Opmantek daemon init scripts or systemd unit files after installation, as these express the dependency on a local MongoDB installation.
 2. **NMIS9 compatible Opmantek Applications released after 13 May 2020 will offer a more comprehensive option `-m f` or `-m F` to instruct the installer to skip all MongoDB related instructions completely during the install. This option is more comprehensive than environment variable `NO_LOCAL_MONGODB`. `NO_LOCAL_MONGODB` only prevents install of MongoDB, but does not skip other MongoDB related instructions that will be executed by the installer.**
Please note that you may have to manually adjust the Opmantek daemon init scripts or systemd unit files after installation, should these express the dependency on a local MongoDB installation.
- If you want to install the product into a non-standard directory, you can pass the argument `-t <targetdir>` to the installer component.
Please note that you will have to adjust a number of configuration files in this case.
- It is possible to generate more detailed diagnostic output in the installer log file, using the `-d` option.

For example, `sh ./opFlow-3.0.5.run --keep -- -n` would start the installer in simulation mode (`-n`) and leave the unpacked files behind (`--keep`) when done.

Logs and Backups

The installer saves a log of all actions taken, files copied etc. in the installation directory as `install.log`, ie. normally it'll be in `/usr/local/omk/install.log`. Subsequent upgrades or installations of other Opmantek products will add to that logfile, so you may very well want to remove or clear the `install.log` file before upgrading or adding extra software.

Unless this is the very first installation of an Opmantek product on this system, the installer will offer taking a backup of all affected files before the installation commences. This backup will be saved in the `root` user's home directory as `omk-backup-YYYY-MM-DD.tgz`. The backup includes:

- all the directories that the installer will later copy files to,
- the `conf` directory,
- the old software manifest,
- and the old `install.log`.

Software Dependencies

Wherever possible the installer will help you with the installation of any missing software dependencies, using `yum` or `apt-get` depending on your operating system platform.

You'll see a prompt similar to this:


```
+++++
Required package httpd is not installed.
+++++

opEvents requires package httpd to be installed and configured.

Do you want to install this package now?
Type 'y' or hit <Enter> to accept, any other key for 'no':
```

If you answer this prompt with 'n' the installer will continue the installation, but the software will likely not work (at all or partially) until you **manually** fix the missing dependency.

In other cases where the dependency is a "soft" one or where automatic installation isn't an option you will be shown a warning dialog about the missing dependency and the installer will wait until you confirm before continuing.

Product Coexistence, Migration and Upgrades

Before installing any Opmantek software components, a thorough check of the existing state of your system will be made to ensure that the new product does integrate correctly with other already existing Opmantek products. This check relies on the software manifests stored in the installation directory (default `/usr/local/omk`) and the product tarball, and thus won't be fully precise if no manifests exist.

When an installation of older/legacy Opmantek products is detected or if the manifest is missing, then the installer will take a comprehensive backup snapshot of your installation directory first. This is to ensure that you could revert back to the pre-installation state quickly and with minimal downtime, should the installer unexpectedly fail the coexistence check or break existing old applications. Here is an example of the prompts in this situation:

```
+++++
An old legacy installation was detected.
+++++

The installer has found a pre-existing installation of one or more
Opmantek products in /usr/local/omk.

The installation can proceed but may cause disruptions to installed
legacy products other than opEvents.

If you agree to continue, the installer will take a backup snapshot
of your complete previous installation and then prepare the
installation environment for opEvents.

Do you want to continue the installation?
Type 'y' or hit <Enter> to accept, any other key for 'no': y
Creating legacy snapshot, please wait...

Snapshot created, file name: /root/omk-legacy-2014-07-14.tgz

The installer has created a full snapshot of your previous installion
in /root/omk-legacy-2014-07-14.tgz. The installation of opEvents will now proceed.

Should you need to revert to your previous installation status,
simply remove all contents of /usr/local/omk and unpack the snapshot:
rm -rf /usr/local/omk/* && tar -C / -xzf /root/omk-legacy-2014-07-14.tgz

Hit <Enter> when ready to continue:
```

If the installer detects an unresolvable conflict between the module dependencies for your existing products and the new product, it will abort the installation with a detailed error message: in this case we recommend that you contact Opmantek Support for a resolution.

For product upgrades the installer will perform the same check and upgrade only the files and modules that are required, taking great care to not damage the function of any other existing Opmantek products. In that case the installer will also recommend a shut down of any Opmantek daemons before the installation commences, so that all files can be copied safely and without negatively affecting running daemons.

Integration and Initial Configuration

After all necessary files have been installed in their appropriate locations the installer will take care of integrating your product with the operating system, web servers and so on.

Typically this will at the minimum involve the installation of up-to-date init scripts for the Opmantek daemon, integration of the Opmantek GUI with your Apache webserver, setting up of log rotation and the optional first start of the Opmantek daemon. The dialogs in question are all very similar to the following:

```
+++++
Updated init script for the Opmantek daemon available
+++++

Ok to install the init script for the Opmantek daemon?
Type 'y' or hit <Enter> to accept, any other key for 'no': y
```

If you answer the prompt with 'n' the installer will continue after displaying a brief outline of the steps you'll have to take manually later and a confirmation dialog:

```
+++++
Opmantek Daemon Startup
+++++

The Opmantek daemon can now be started, but you might want to delay
that until you have adjusted the configuration files.

Do you want to start the Opmantek daemon now?
Type 'y' or hit <Enter> to accept, any other key for 'no': n
Skipping start of OMKD
Please note that you will have to start the Opmantek daemon to activate
the Opmantek GUI. You can do so by running 'service omkd start' as
the root user.

Hit <Enter> when ready to continue:
```

The installer will also offer to copy any missing default configuration files from the `install` to the `conf` directory to provide you with a basic initial configuration to start with.

In case of an upgrade it'll offer to import any new default config settings. Furthermore, you will be given the opportunity to have all your configuration files compared to the defaults:

```
+++++
Detecting Configuration Changes
+++++

Would you like to see an overview of all changed configuration items?
Type 'y' or hit <Enter> to accept, any other key for 'no':
Performing config diff check, please wait...

The configuration comparison tool has detected some differences
between the shipped defaults (in /usr/local/omk/install) and
the active settings (in /usr/local/omk/conf).

The affected files are:  EventActions.nmis opCommon.nmis

A detailed listing of these differences has been
saved in /tmp/opEvents-config-diffs-2014-07-14.

You should review those differences (using less or an editor like
nano, vi or emacs) and adjust your configuration settings accordingly.

Hit <Enter> when ready to continue:
```

Finally, at the end of the installation process you'll see a message like this:

```

+++++
opEvents is Ready for Configuration
+++++

This initial installation of opEvents is now complete.

However, to configure and fine-tune the application suitably for
your environment you will need to make certain configuration adjustments.

We highly recommend that you visit the documentation site for opEvents at

https://community.opmantek.com/display/opEvents/Home

The next step is to determine what configuration changes
will be required for your environment.

If you have started the Opmantek and the opEvents daemons,
then your new opEvents dashboard should now be accessible at
http://<HOSTNAME_OR_IP>/omk/opEvents/

If your browser is running on the same machine as opEvents was
installed onto, this would be http://localhost/omk/opEvents/

+++++
installation complete.
+++++

```

FAQ

- What's this warning about "incorrect checksum detected"?
This can happen very infrequently, if you are installing an older Opmantek application on top of newer ones, or if you've made extensive changes to your system's Opmantek files.
We strive hard to line up our releases properly so that everything meshes cleanly, but every now and then there are minor changes to files that older installer versions aren't quite aware of.

In general this warning dialog is safe to answer with 'yes' and the installer will leave your system in a consistent working state (by replacing the unrecognizable/mismatching file with a known good version from the shipped product).

Please feel free to submit your comment here or email us with your questions!

Upgrading NMIS 9 Compatible Opmantek Applications

We have made significant changes on our internal code for all our applications to work on Opmantek's latest and fastest platform, however, previously installed product are not compatible with these new changes.

Compatibility Table	
Application	Version
NMIS 9	> 9.1.0G
opCharts 4	>= 4.20
opConfig 4	>= 4.20
opEvents 3	>= 3.2.0
opHA 3	>= 3.20
opReports 4	>= 4.20

To find out more about this upgrade please read:

- [Upgrading Opmantek Applications](#)

- [Product Compatibility](#)

Uninstalling Opmantek Applications

Because Opmantek applications share code and modules wherever possible, uninstalling a single application is not completely trivial.

Using the Uninstaller

As of September 2016, all application releases include an uninstaller tool which performs a *limited* uninstallation of a particular application. It's easy to use, but primarily disables an application without removal of application data or files. The uninstaller offers a simulation mode, too. You simply start it up with the application module in question, e.g.

```
# -n invokes the simulation mode
/usr/local/omk/bin/uninstaller.exe -n opCharts
...
Would remove opCharts from load_applications list in opCommon.nmis.
Would restart service omkd.
Would stop service nmisd.
Would move init script nmisd to /root/uninstall-backup.
```

Manual Removal

If you desire a more permanent and complete application removal you will have to remove *all* Opmantek applications: it is infeasibly complicated to determine which files and code modules are removable and which have to remain behind to keep the remaining applications in working shape.

A checklist for complete removal would involve the following steps:

- Removal of all Opmantek daemon init scripts from `/etc/init.d`
This may include init scripts for `omkd`, `nmisd`, `opeventsd`, `opconfigd`, `nfdump`, `opflowd`.
You should stop the daemons before removing the init scripts.
- Removal of `cron` schedules for the Opmantek applications
This may include files in `/etc/cron.d` named `oae`, `opaddress`, `opconfig`, `opevents`, `opflow`, `opreports`, `optrend`.
- Removal of all of `/usr/local/omk`, `/var/log/omk`, `/data/omk`
The latter two may not be present (but will be if your system started as an [Opmantek Virtual Appliance](#)).
- Cleanup or removal of the Opmantek applications' MongoDB databases
Unless you are actively using MongoDB you might simply stop the `mongod` daemon and remove the database files (typically under `/var/lib/mongodb` or `/data/mongodb` for the Opmantek VM).