

# Installing NMIS9 Compatible OMK Applications in a Disconnected (Air-Gapped) Environment

- [Note:](#)
- [Requirements](#)
- [Principle of Least Privilege](#)
- [Dependency Check Mode](#)
- [Centos 7 Example](#)
  - On the disconnected (air-gapped) computer we will firstly determine which pre-installation required packages need to be downloaded:
  - Using the connected computer with same OS and Version we will now fetch the pre-installation required packages:
  - On the disconnected (air-gapped) computer we will now install the downloaded required packages:
- [Debian 9 Example](#)
  - On the disconnected (air-gapped) computer we will firstly determine which pre-installation required packages need to be downloaded:
  - Using the connected computer with same OS and Version we would now fetch the pre-installation required packages:
  - On the disconnected (air-gapped) computer we will now install the downloaded required packages:
- [Related Topics](#)
  - [Packages and Modules Summary](#)
  - [Example Dependency Lists](#)
    - [Pristine Centos 7.8.2003 Dependencies](#)
    - [Pristine Debian 9.12 Dependencies](#)

## Note:

The software has a version checker to let you know if there are any updates to your installed modules. This version check utilizes the clients browser to run, to disable this go to /usr/local/omk/templates/opmantek/welcome.html.ep and remove the block of code that starts with "// Used to make the version badge green or red based on the users installed version", around line 276. After removing do not forget to restart daemons.

```
271         window.open(url, target);
272     }
273 }
274 });
275
276 // Used to make the version badge green or red based on the users installed version
277 $( ".version-badge" ).each(function( index ) {
278
279     //get the version and app name from the span element
280     var product = $(this).data('appname');
281     if (product.includes("Open-Audit")) {
282         product = "Open-Audit";
283     }
284     var version = $(this).data('appversion');
285
286     //cache element for later use
287     var element = $(this);
288
289     var checkurl = "https://services.opmantek.com/api/versioncheck/"
290     +encodeURIComponent(product)+"-"+encodeURIComponent(version);
291
292     // only make the request if we have enough data;
293     if(product !== undefined && version !== undefined){
294         $.getJSON(checkurl, function( data ) {
295             if(data.status === 1){
296                 element.addClass('green');
297             }else if(data.status === 0){
298                 element.addClass('blue');
299             }else{
300                 element.addClass('red');
301             }
302         });
303     }
304 });
305
```

## Requirements

This method works best with a connected computer with same OS and Version.

This enables one to ascertain the dependent packages that need to be downloaded by the Package Manager for the disconnected (air-gapped) computer.

Strategy will be to only install packages if absolutely necessary to minimise need to vet new packages.

Root privileges will only be used when absolutely necessary.

## Principle of Least Privilege

When run with 'Dependency Check Mode' enabled '-D', the Opmantek Installer cannot be run as root.

Should you want to run the Opmantek Installer with least privileges you may need to create a user with least privileges.

The Opmantek Installer does not take least privileges to extremes and leaves it in the hands of the person installing to run the Installer with least privileges.

This is important as, for example, by default the nmis user has 0770 permissions on the nmis directory, and a user in the nmis directory will have write and execute permissions on the nmis directory.

Run the following one-liner bash script to check your permissions on the nmis and omk directories if they exists, were you to run the installer as yourself - outputs only when write and execute permissions are detected:

```
NMIS=/usr/local/nmis9;OMK=/usr/local/omk;[ -x "$NMIS" ] && echo "Has execute permissions: $NMIS";[ -w "$NMIS" ] && echo "Has write permissions: $NMIS";[ -w "$OMK" ] && echo "Has write permissions: $OMK"
```

## Dependency Check Mode

NMIS9 Compatible OMK Application Installers created on or after 2020-08-18 will include a new option '-D': *Dependency Check Mode*. When run with *Dependency Check Mode* enabled '-D', the Opmantek Installer cannot be run as root.

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.

**Below is a Centos 7 Example and Debian 9 Example illustrating how to get dependencies installed on the disconnected (air-gapped) computer.**

**These examples are focused on dependencies needed to unpack the installer during execution.**

**However, the same strategy can be adopted to install dependencies, other than NMIS9, listed during the Dependency Check.**

## Centos 7 Example

On Redhat and Centos OS, tar and libnsl may need to be installed as they are needed to unpack the installer runfile.

1. **On the disconnected (air-gapped) computer we will firstly determine which pre-installation required packages need to be downloaded:**

```
First we get the architecture:
uname --m
x86_64

Then we check if tar package is installed:
rpm -q tar.x86_64
tar-1.26-35.el7.x86_64
# tar package is installed.
# We will only consider downloading and installing it later if we find we are unable to unpack the Opmantek Installer Runfile.

# Next we check if libnsl package is installed:
rpm -q libnsl
package libnsl is not installed
# libnsl is not installed.

# From the above checks on this disconnected (air-gapped) computer, we have ascertained that the following packages are needed:
libnsl
```

2. **Using the connected computer with same OS and Version we will now fetch the pre-installation required packages:**

```
# Navigate to a clean directory to download packages to
mkdir -p /tmp/omk_opXyz_packages
cd /tmp/omk_opXyz_packages/
```

```
# Download each of the dependent packages:
yumdownloader --resolve libnsl --archlist=x86_64
```

```
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
epel/x86_64/metalink | 26 kB 00:00:00
* base: centos.mirror.serversaustralia.com.au
* epel: mirror.xeonbd.com
* extras: centos.mirror.serversaustralia.com.au
* updates: mirror.nsw.coload.com.au
base | 3.6 kB 00:00:00
epel | 4.7 kB 00:00:00
extras | 2.9 kB 00:00:00
updates | 2.9 kB 00:00:00
```

```
No Match for argument libnsl
Nothing to download
```

```
# At first sight it appears we don't need libnsl for the OS + Version we are running this example on.
```

```
# If the Opmantek Installer Runfile fails to unpack we may need to download this package from an OS Repository that is not enabled by default.
```

```
# Since we don't have any package to download in this example we will download the tar package.
```

```
# We will assume for the purposes of this example that the tar package needs to be updated to a newer version.
```

```
# The command to download tar package with dependencies is:
```

```
yumdownloader --resolve tar --archlist=x86_64
```

```
# Downloaded packages will now be in the current directory and ready for transferring to the disconnected (air-gapped) computer where they will be installed.
```

```
ls -la
```

```
-rw-rw-r--. 1 user user 865848 Nov 12 2018 tar-1.26-35.el7.x86_64.rpm
```

### 3. On the disconnected (air-gapped) computer we will now install the downloaded required packages:

```
# Install each of the downloaded required packages.
```

```
# Since we don't have any package to install in this example we will install the tar package.
```

```
# We will assume for the purposes of this example that the tar package needs to be updated to a newer version.
```

```
sudo yum install tar-1.26-35.el7.x86_64.rpm
```

## Debian 9 Example

On Debian and Ubuntu OS, tar, which is normally installed by default, may need to be installed as it is needed to unpack the installer runfile.

### 1. On the disconnected (air-gapped) computer we will firstly determine which pre-installation required packages need to be downloaded:

```
First we get the architecture:
```

```
uname --m
```

```
x86_64
```

```
# Next we check if tar package is installed:
```

```
type tar
```

```
tar is /bin/tar
```

```
# tar package is installed
```

```
# We will only consider downloading and installing it later if we find we are unable to unpack the Opmantek Installer Runfile.
```

```
# From the above checks on this disconnected (air-gapped) computer, we have ascertained that no packages are needed.
```

```
# Since we don't have any package to download in this example we will download the tar package.
```

```
# We will assume for the purposes of this example that the tar package needs to be updated to a newer version.
```

### 2. Using the connected computer with same OS and Version we would now fetch the pre-installation required packages:

```
# Navigate to a clean directory to download packages to
mkdir -p /tmp/omk_opXyz_packages
cd /tmp/omk_opXyz_packages/
```

```
# Since we don't have any package to download in this example we will download the tar package.
# We will assume for the purposes of this example that the tar package needs to be updated to a newer version.
# The command to download tar package with dependencies is:
```

```
apt-get download tar
```

```
Get:1 http://ftp.us.debian.org/debian stretch/main amd64 tar amd64 1.29b-1.1 [759 kB]
```

```
Fetch: 759 kB in 2s (305 kB/s)
```

```
# Downloaded packages will now be in the current directory and ready for transferring to the disconnected (air-gapped) computer where they
will be installed:
```

```
ls -la
```

```
-rw-rw-r--. 1 user user 4463036 Apr 3 21:12 759354 Oct 31 2016 tar_1.29b-1.1_amd64.deb
```

### 3. On the disconnected (air-gapped) computer we will now install the downloaded required packages:

```
# Install each of the downloaded required packages, here using tar as an example, even though it is not actually needed in this case:
sudo apt-get install tar_1.29b-1.1_amd64.deb
```

## Related Topics

- [Packages and Modules Summary](#)
- [Example Dependency Lists](#)
  - [Pristine Centos 7.8.2003 Dependencies](#)
  - [Pristine Debian 9.12 Dependencies](#)