

# ELT DevEnv installations

## Frequent Asked Questions

Release 1.1

## Contents

ELT DevEnv installations .....	1
Frequent Asked Questions.....	1
What are the minimum HW requirements to run the ELT DevEnv? .....	1
Which accounts are installed with the DevEnv? .....	2
How I do create other user accounts for ELT software development? .....	2
Which release of DevEnv is running in my host?.....	2
What does the release number mean? .....	2
Is my host compliant with the installed DevEnv release? .....	3
Which environment variables are relevant to DevEnv installations? .....	3
Is there a newer DevEnv release already available?.....	4
How I can update my host to a newer release of the DevEnv? .....	5
How can I downgrade to an older DevEnv release?.....	6
Which are the YUM repositories available for DevEnv .....	6
Can I search for a specific RPM in the repositories? .....	7
How can I add an extra RPM to my host? .....	7
How can I remove an RPM from my host? .....	8
How can I find if an RPM is already installed?.....	8
Can I list the files included in an RPM? .....	9
How I do get DevEnv installation support? .....	9

### What are the minimum HW requirements to run the ELT DevEnv?

- 4x CPUs x86\_64 (Intel or AMD)
- 8GB RAM
- 100GB disk
- 1x NIC
- HW compatible with CentOS 8

You may need more memory and CPU for heavy software compilations. At ESO, VMs for the DevEnv are, by default, created with:

- 8x CPUs x86\_64 (Intel or AMD)
- 16GB RAM

However, when installing the DevEnv on a laptop or desktop you might be limited by the physical HW of the host. DevEnv can still be operative with:

- 2x CPUs x86\_64 (Intel or AMD)
- 4GB RAM

## Which accounts are installed with the DevEnv?

With the installation of the DevEnv there are three accounts available:

- Username: **root** password: <the-one-given-during-OS-installation>
- Username: **eltdev** password: **2Garch1ng**
- Username: **eltmgr** password: **pass4u**

Account **eltdev** is meant to be used for ELT software development

Account **eltmgr** is reserved to be used for ELT administration tasks (TBD).

## How I do create other user accounts for ELT software development?

As administrator of a system with the ELT DevEnv, you can create new user accounts with the command **useradd**.

Example: (as root):

```
# useradd -g elt newuser      # newuser is created under /home/newuser
# passwd newuser             # set a password for account newuser
```

You can also remove them with the command **userdel**

```
# userdel -f newuser         # account newuser is deleted, including its home dir
```

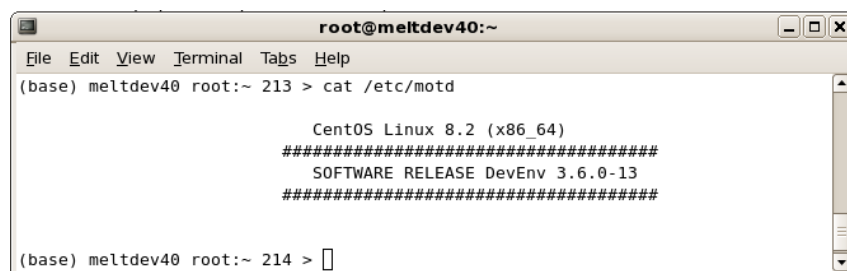
## Which release of DevEnv is running in my host?

You can check the environment variable **\$ELT\_RELEASE**:

```
% echo $ELT_RELEASE
3.6.0-13
```

This DevEnv installation also modifies the content of the file **/etc/motd** (Message Of The Day), indicating the OS and DevEnv release. Example:

```
% cat /etc/motd
```



```
root@meltdev40:~
File Edit View Terminal Tabs Help
(base) meltdev40 root:~ 213 > cat /etc/motd

                CentOS Linux 8.2 (x86_64)
                #####
                SOFTWARE RELEASE DevEnv 3.6.0-13
                #####

(base) meltdev40 root:~ 214 > █
```

## What does the release number mean?

The ELT Linux DevEnv is defined by the environment variable **\$ELT\_RELEASE**. It follows standard numeration **major.minor.patch-iteration**, where an increase in:

- the **major** number indicates a considerable change largely affecting the complete DevEnv, e.g. with a new OS. (2: CentOS-7, 3: CentOS-8).
- the **minor** number indicates a change affecting important components of the DevEnv, like toolkits or their releases.
- the **patch** number will be used to fix a severe bug of a minor release.
- the **iteration** number is meaningless. Only for integration purposes.

Typically DevEnv will be released with the following frequency:

- **Major** releases will be released with a frequency of at least 6 months. A major the release will be announced at least two months in advance.
- **Minor** releases are expected to be released with a frequency of months
- **Patch** releases will be made available to fix a severe bug in a Minor release.

## Is my host compliant with the installed DevEnv release?

The script **elt-devenv** can help you to check if your host is compliant with the installed DevEnv release: The script compares the list of RPMs installed in your host with the list of RPMs from a pristine DevEnv installation. If both lists are identical your host is compliant, any other difference is displayed: what is found vs. what is expected.

Example of a compliant host:

```
% elt-devenv -i
ELT Dev Env version: 3.6.0-13
Status: fully compatible
```

Example of a non-compliant host:

```
% elt-devenv -i
ELT Dev Env version: 3.6.0-13
Status: modifications found
Found: bison-3.0.4-10.el8.x86_64
Found: flex-2.6.1-9.el8.x86_64
Expected: lightdm-1.30.0-6.el8.x86_64
Found: lightdm-1.30.0-8.el8.x86_64
```

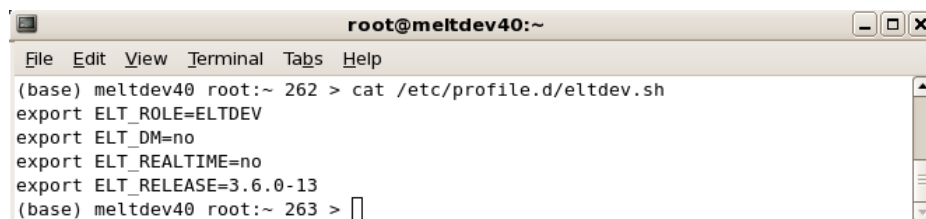
The output states that two extra RPMs were installed, and one updated.

## Which environment variables are relevant to DevEnv installations?

```
% echo $ELT_ROLE
ELTDEV
% echo ELT_DM
no
% echo $ELT_REALTIME
no
% echo $ELT_RELEASE
3.6.0-13
```

These variables are defined in **/etc/profile.d/eltdev.sh**

Example:

A terminal window titled 'root@meltdev40:~' showing the output of the command 'cat /etc/profile.d/eltdev.sh'. The output is: (base) meltdev40 root:~ 262 > cat /etc/profile.d/eltdev.sh  
export ELT\_ROLE=ELTDEV  
export ELT\_DM=no  
export ELT\_REALTIME=no  
export ELT\_RELEASE=3.6.0-13  
(base) meltdev40 root:~ 263 > 

With the following meanings:

- **\$ELT\_ROLE** with the following values:

- **ELTDEV** DevEnv was installed as a development machine.
- **MINIMAL** DevEnv has been installed with the minimal set of packages

**Note:** you can upgrade from MINIMAL to ELTDEV but not the other way around.

To upgrade from MINIMAL to ELTDEV execute, as root:

```
% export ELT_ROLE=ELTDEV
% /root/elt/puppet-force-align
```

- **\$ELT\_DM** DM stands for “Display Manager” and it sets or doesn’t set the execution of a daemon for a display manager in the console with these values:
  - **NO:** no Display Manager will be launched after the installation. This is the default value.
  - **YES:** Display Manager will be launched after the installation.

To run a “Display Manager” execute:

```
% export ELT_DM=yes
% /root/elt/puppet-force-align
```

- **\$ELT\_REALTIME** It defines if the installation should or shouldn’t include the real-time extensions. It has two unique options:
  - **NO:** no real-time workstation. This is the default value.
  - **YES:** install the real-time extension. This variant installs a real-time kernel, packages specific to real-time development and other real-time specific system configurations.

**Note:** you can upgrade to REALTIME but not the other way around.

To install the REALTIME extensions execute:

```
% export ELT_REALTIME=yes
% /root/elt/puppet-force-align; reboot
```

- **\$ELT\_RELEASE** It contains the DevEnv release, e.g. 3.6.0-7

**NOTE:** These variables are set to default values during the first ELT DevEnv installation. As root, they can be changed to other values as specified above, but they will not have any effect until you execute the puppet-force-align script. **ELT\_RELEASE** should not be modified by hand; it will reset again with the next execution of the script.

## Is there a newer DevEnv release already available?

Newer releases of the DevEnv are not automatically installed in your system (unless your sysadmin does it for you), therefore your system might be running an obsolete version of the DevEnv.

However, the file **/etc/motd** is updated by a cronjob process to highlight the latest version of DevEnv available. Example:

```
% cat /etc/motd
```

```

eitdev@meltdev41:~
File Edit View Terminal Tabs Help
(base) meltdev41 eitdev:~ 383 > cat /etc/motd

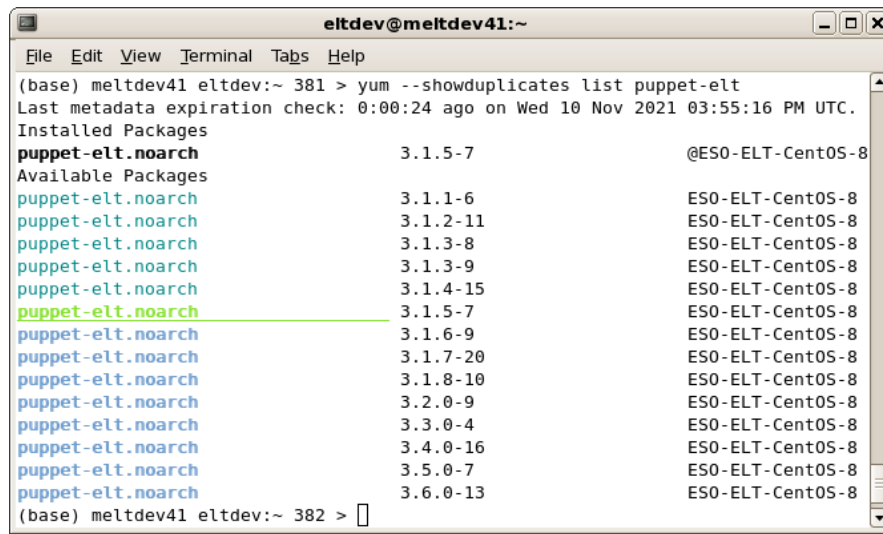
          CentOS Linux 8.2 (x86_64)
#####
SOFTWARE RELEASE DevEnv 3.1.5-7
#####
(newer DevEnv available: 3.6.0-13)

(base) meltdev41 eitdev:~ 384 >

```

You can also check all releases of the DevEnv available in the repository with the command:

```
% yum --showduplicates list puppet-elt
```



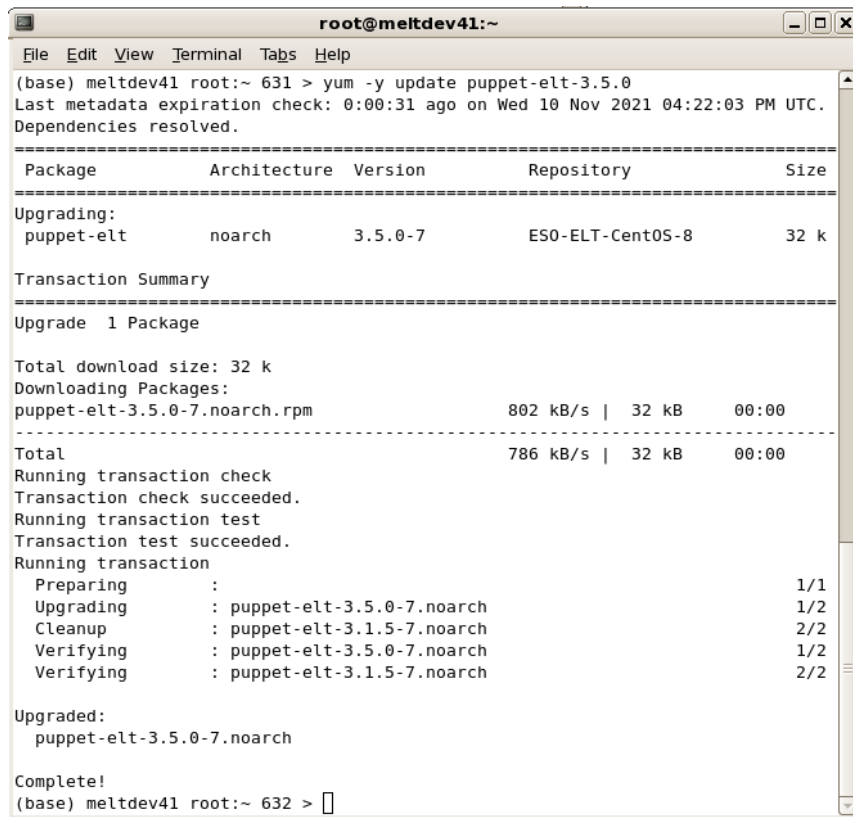
```
eltdev@meltdev41:~  
File Edit View Terminal Tabs Help  
(base) meltdev41 eltdev:~ 381 > yum --showduplicates list puppet-elt  
Last metadata expiration check: 0:00:24 ago on Wed 10 Nov 2021 03:55:16 PM UTC.  
Installed Packages  
puppet-elt.noarch           3.1.5-7           @ESO-ELT-CentOS-8  
Available Packages  
puppet-elt.noarch           3.1.1-6           ESO-ELT-CentOS-8  
puppet-elt.noarch           3.1.2-11          ESO-ELT-CentOS-8  
puppet-elt.noarch           3.1.3-8           ESO-ELT-CentOS-8  
puppet-elt.noarch           3.1.3-9           ESO-ELT-CentOS-8  
puppet-elt.noarch           3.1.4-15          ESO-ELT-CentOS-8  
puppet-elt.noarch           3.1.5-7           ESO-ELT-CentOS-8  
puppet-elt.noarch           3.1.6-9           ESO-ELT-CentOS-8  
puppet-elt.noarch           3.1.7-20          ESO-ELT-CentOS-8  
puppet-elt.noarch           3.1.8-10          ESO-ELT-CentOS-8  
puppet-elt.noarch           3.2.0-9           ESO-ELT-CentOS-8  
puppet-elt.noarch           3.3.0-4           ESO-ELT-CentOS-8  
puppet-elt.noarch           3.4.0-16          ESO-ELT-CentOS-8  
puppet-elt.noarch           3.5.0-7           ESO-ELT-CentOS-8  
puppet-elt.noarch           3.6.0-13          ESO-ELT-CentOS-8  
(base) meltdev41 eltdev:~ 382 >
```

The output of the command shows the current release of the RPM installed in the system in bright green, newer releases below in blue, and older ones above in green.

## How I can update my host to a newer release of the DevEnv?

As root, execute the command “rpm -q update puppet-elt-<rel>”. Example:

```
# yum -y update puppet-elt-3.5.0
```



```
root@meltdev41:~  
File Edit View Terminal Tabs Help  
(base) meltdev41 root:~ 631 > yum -y update puppet-elt-3.5.0  
Last metadata expiration check: 0:00:31 ago on Wed 10 Nov 2021 04:22:03 PM UTC.  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository	Size
Upgrading: puppet-elt	noarch	3.5.0-7	ESO-ELT-CentOS-8	32 k

```
=====
```

Transaction Summary

```
=====
```

Upgrade 1 Package

Total download size: 32 k  
Downloading Packages:  
puppet-elt-3.5.0-7.noarch.rpm 802 kB/s | 32 kB 00:00  
-----

Total 786 kB/s | 32 kB 00:00

Running transaction check  
Transaction check succeeded.  
Running transaction test  
Transaction test succeeded.  
Running transaction  
Preparing : 1/1  
Upgrading : puppet-elt-3.5.0-7.noarch 1/2  
Cleanup : puppet-elt-3.1.5-7.noarch 2/2  
Verifying : puppet-elt-3.5.0-7.noarch 1/2  
Verifying : puppet-elt-3.1.5-7.noarch 2/2

Upgraded:  
puppet-elt-3.5.0-7.noarch

Complete!  
(base) meltdev41 root:~ 632 >

If you just want to update to the latest DevEnv release, execute the same command without specifying any release. Example:

```
# yum -y update puppet-elt
```

You complete the DevEnv release with the execution of the puppet script:

```
# /root/elt/puppet-force-align
```

## How can I downgrade to an older DevEnv release?

There is a procedure supporting the downgrade of your DevEnv release system to an older previously running version.

For example, if your system was upgraded from release 3.1.1 to new 3.1.3, you can, after the upgrade, downgrade it back to your previous release 3.1.1 with the following sequence, as root:

```
# /root/elt/puppet-downgrade puppet-elt-3.1.1
```

The process is guaranteed under certain limitations:

Rollback or downgrade of the following packages is unsupported as the packages themselves, and dependencies, either assume an update-only or install-only process: dbus, kernel, glibc, or selinux-policy.

- The downgrade to a given release can only be done on hosts updated previously from that release. You cannot downgrade to a given older version if the host was never updated from that given version.
- There are some inherited risks to the downgrade process: x it may leave some orphan files not cleared by the undo process.
- The process will be verified to work between two consecutive releases, i.e. from the current release to the previous one. Any attempt to downgrade to an older release is not guaranteed and can only be done at your own risk.
- Only packages installed with YUM are considered in the downgrade process, and the process is unable to distinguish between packages installed or updated by the DevEnv update process; from other packages installed with YUM by hand. The downgrade process might remove/downgrade them all.

## Which are the YUM repositories available for DevEnv

The YUM repositories are listed in the directory `/etc/yum.repo.d/*.repo`. Not all of them are enabled. By default, only these files contain repositories enabled:

- **ESO-CentOS.repo** # A clone copy at ESO of the official CentOS-8 repository
- **ESO-ELT-CentOS-8.repo** # Additions and upgrades to the CentOS-8 repository

You should **not** enable any other repository as they contain RPMs that may conflict with those used by DevEnv, leading to non-compliant DevEnv.

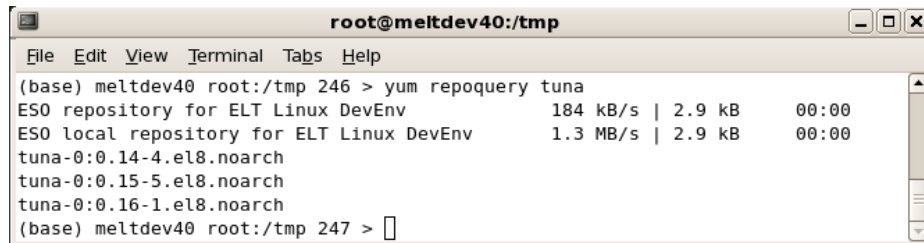
For ESO hosts only, there is another local repository that can be enabled temporarily to download a licensed package, e.g.: to install the DDS package from RTI CONNECT execute, as root:

```
# yum-config-manager --enable ESO-Local-CentOS-8
# yum -y install opt-rti_connect_dds
# yum-config-manager --disable ESO-Local-CentOS-8
```

## Can I search for a specific RPM in the repositories?

Yes, but only on those enabled repositories in `/etc/yum.repos.d/*.repo` with the command

```
# yum repoquery <package>
```



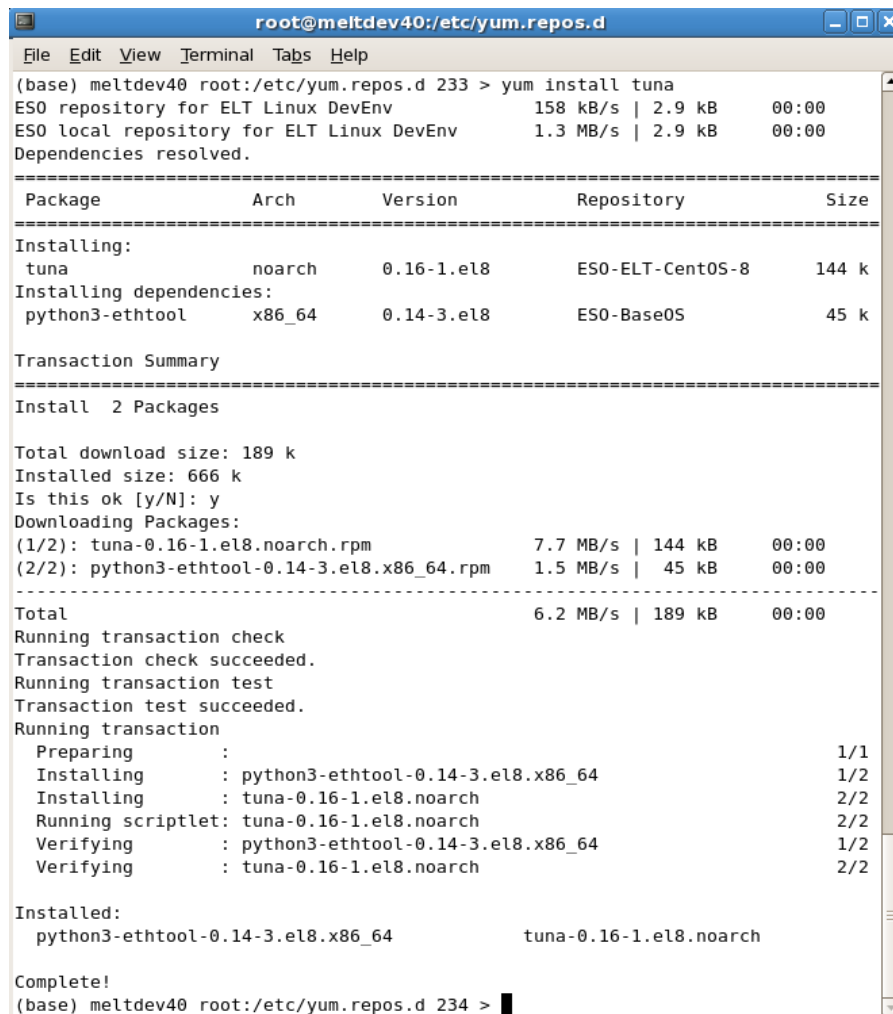
```
root@meltdev40:/tmp
File Edit View Terminal Tabs Help
(base) meltdev40 root:/tmp 246 > yum repoquery tuna
ES0 repository for ELT Linux DevEnv      184 kB/s | 2.9 kB    00:00
ES0 local repository for ELT Linux DevEnv 1.3 MB/s | 2.9 kB    00:00
tuna-0:0.14-4.el8.noarch
tuna-0:0.15-5.el8.noarch
tuna-0:0.16-1.el8.noarch
(base) meltdev40 root:/tmp 247 >
```

## How can I add an extra RPM to my host?

Two ways, and always as root:

1. with the command **yum** you can download RPMs from those enabled repositories in `/etc/yum.repos.d/*.repo`. Example:

```
# yum install <package>
```



```
root@meltdev40:/etc/yum.repos.d
File Edit View Terminal Tabs Help
(base) meltdev40 root:/etc/yum.repos.d 233 > yum install tuna
ES0 repository for ELT Linux DevEnv      158 kB/s | 2.9 kB    00:00
ES0 local repository for ELT Linux DevEnv 1.3 MB/s | 2.9 kB    00:00
Dependencies resolved.
-----
Package           Arch      Version      Repository      Size
-----
Installing:
tuna               noarch    0.16-1.el8    ES0-ELT-CentOS-8 144 k
Installing dependencies:
python3-ethtool    x86_64    0.14-3.el8    ES0-Base05      45 k
-----
Transaction Summary
-----
Install 2 Packages

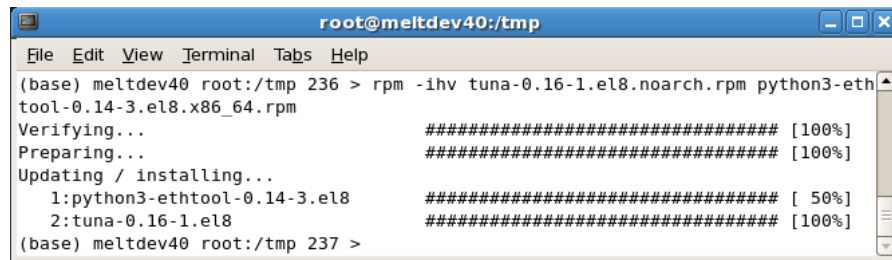
Total download size: 189 k
Installed size: 666 k
Is this ok [y/N]: y
Downloading Packages:
(1/2): tuna-0.16-1.el8.noarch.rpm        7.7 MB/s | 144 kB    00:00
(2/2): python3-ethtool-0.14-3.el8.x86_64.rpm 1.5 MB/s | 45 kB     00:00
-----
Total                                       6.2 MB/s | 189 kB    00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing    : python3-ethtool-0.14-3.el8.x86_64 1/2
  Installing    : tuna-0.16-1.el8.noarch           2/2
  Running scriptlet: tuna-0.16-1.el8.noarch        2/2
  Verifying     : python3-ethtool-0.14-3.el8.x86_64 1/2
  Verifying     : tuna-0.16-1.el8.noarch           2/2

Installed:
python3-ethtool-0.14-3.el8.x86_64          tuna-0.16-1.el8.noarch

Complete!
(base) meltdev40 root:/etc/yum.repos.d 234 >
```

With yum, the package and its dependencies will be first downloaded and then installed.

2. With the command ***rpm -ihv <packages>***, you can install packages previously downloaded by hand in your system. The same example above with ***rpm***:



```
root@meltdev40:/tmp
File Edit View Terminal Tabs Help
(base) meltdev40 root:/tmp 236 > rpm -ihv tuna-0.16-1.el8.noarch.rpm python3-eth
tool-0.14-3.el8.x86_64.rpm
Verifying... ##### [100%]
Preparing... ##### [100%]
Updating / installing...
 1:python3-ethtool-0.14-3.el8 ##### [ 50%]
 2:tuna-0.16-1.el8 ##### [100%]
(base) meltdev40 root:/tmp 237 >
```

Notice, that with the command ***rpm*** you need to give your primary RPM (tuna) and all its missing dependencies (python3-ethtool).

## How can I remove an RPM from my host?

As root, and with the command ***rpm -e <packages>***.

Example:

```
# rpm -e tuna python3
```

**Notice** that the command ***rpm*** will fail to remove a package if other packages depend on it.

Example:

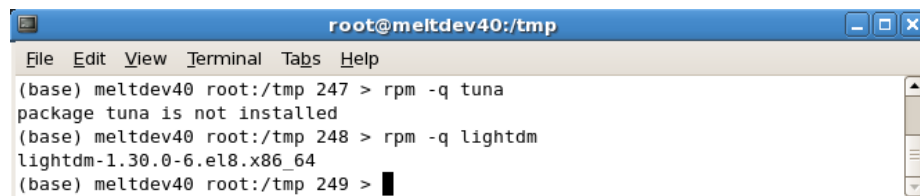


```
root@meltdev40:/tmp
File Edit View Terminal Tabs Help
(base) meltdev40 root:/tmp 240 > rpm -e python3-ethtool
error: Failed dependencies:
 python3-ethtool is needed by (installed) tuna-0.16-1.el8.noarch
(base) meltdev40 root:/tmp 241 >
```

## How can I find if an RPM is already installed?

With the command ***rpm -q <package>***

Example:



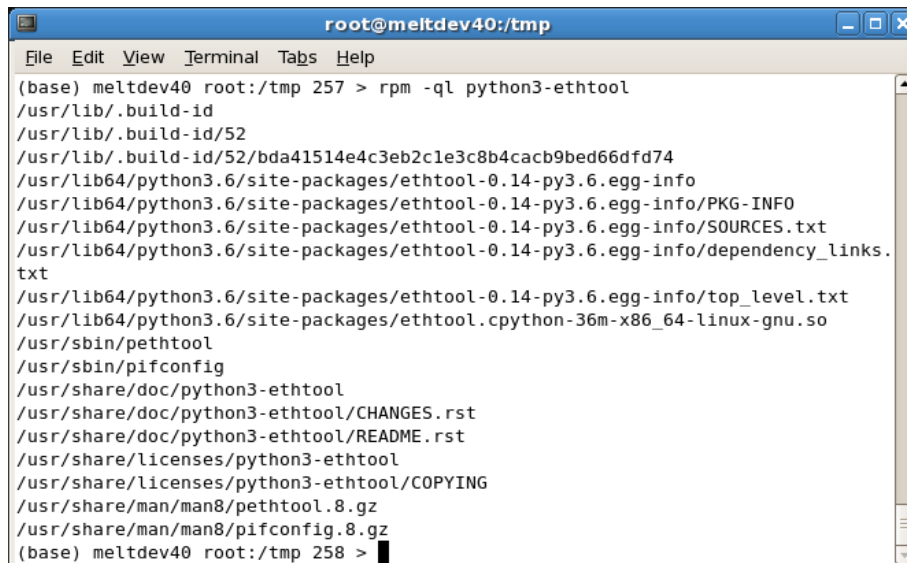
```
root@meltdev40:/tmp
File Edit View Terminal Tabs Help
(base) meltdev40 root:/tmp 247 > rpm -q tuna
package tuna is not installed
(base) meltdev40 root:/tmp 248 > rpm -q lightdm
lightdm-1.30.0-6.el8.x86_64
(base) meltdev40 root:/tmp 249 >
```



## Can I list the files included in an RPM?

With the command `rpm -ql <package>`, if the RPM is already installed.

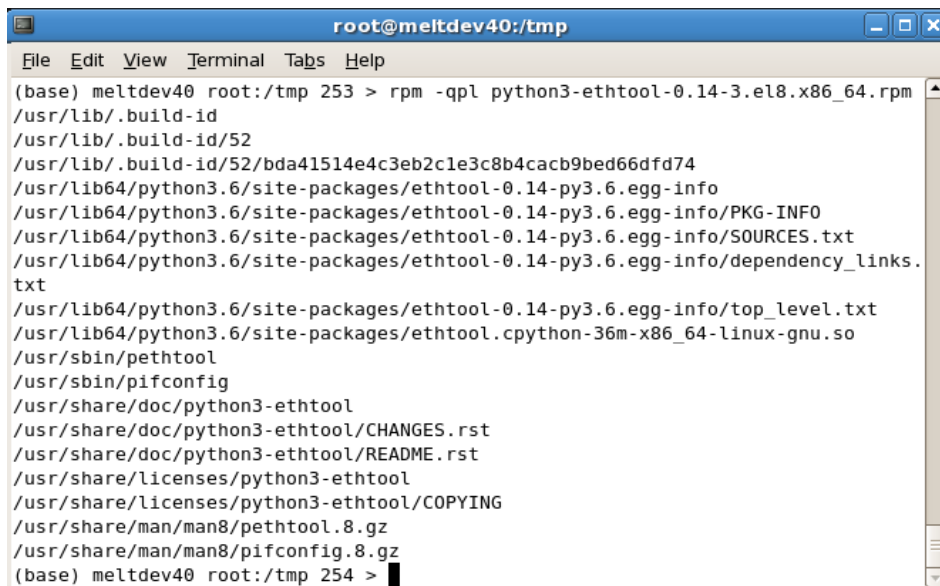
Example:



```
root@meltdev40:/tmp
File Edit View Terminal Tabs Help
(base) meltdev40 root:/tmp 257 > rpm -ql python3-ethtool
/usr/lib/.build-id
/usr/lib/.build-id/52
/usr/lib/.build-id/52/bda41514e4c3eb2c1e3c8b4cacb9bed66dfd74
/usr/lib64/python3.6/site-packages/ethtool-0.14-py3.6.egg-info
/usr/lib64/python3.6/site-packages/ethtool-0.14-py3.6.egg-info/PKG-INFO
/usr/lib64/python3.6/site-packages/ethtool-0.14-py3.6.egg-info/SOURCES.txt
/usr/lib64/python3.6/site-packages/ethtool-0.14-py3.6.egg-info/dependency_links.txt
/usr/lib64/python3.6/site-packages/ethtool-0.14-py3.6.egg-info/top_level.txt
/usr/lib64/python3.6/site-packages/ethtool.cpython-36m-x86_64-linux-gnu.so
/usr/sbin/pethtool
/usr/sbin/pifconfig
/usr/share/doc/python3-ethtool
/usr/share/doc/python3-ethtool/CHANGES.rst
/usr/share/doc/python3-ethtool/README.rst
/usr/share/licenses/python3-ethtool
/usr/share/licenses/python3-ethtool/COPYING
/usr/share/man/man8/pethtool.8.gz
/usr/share/man/man8/pifconfig.8.gz
(base) meltdev40 root:/tmp 258 >
```

With the command `rpm -qpl <file.rpm>` if the RPM has not been yet installed.

Example:



```
root@meltdev40:/tmp
File Edit View Terminal Tabs Help
(base) meltdev40 root:/tmp 253 > rpm -qpl python3-ethtool-0.14-3.el8.x86_64.rpm
/usr/lib/.build-id
/usr/lib/.build-id/52
/usr/lib/.build-id/52/bda41514e4c3eb2c1e3c8b4cacb9bed66dfd74
/usr/lib64/python3.6/site-packages/ethtool-0.14-py3.6.egg-info
/usr/lib64/python3.6/site-packages/ethtool-0.14-py3.6.egg-info/PKG-INFO
/usr/lib64/python3.6/site-packages/ethtool-0.14-py3.6.egg-info/SOURCES.txt
/usr/lib64/python3.6/site-packages/ethtool-0.14-py3.6.egg-info/dependency_links.txt
/usr/lib64/python3.6/site-packages/ethtool-0.14-py3.6.egg-info/top_level.txt
/usr/lib64/python3.6/site-packages/ethtool.cpython-36m-x86_64-linux-gnu.so
/usr/sbin/pethtool
/usr/sbin/pifconfig
/usr/share/doc/python3-ethtool
/usr/share/doc/python3-ethtool/CHANGES.rst
/usr/share/doc/python3-ethtool/README.rst
/usr/share/licenses/python3-ethtool
/usr/share/licenses/python3-ethtool/COPYING
/usr/share/man/man8/pethtool.8.gz
/usr/share/man/man8/pifconfig.8.gz
(base) meltdev40 root:/tmp 254 >
```

## How I do get DevEnv installation support?

**For ESO internal use only:** Support to the ELT Linux DevEnv installation will be provided via creating a new ticket in our JIRA ticketing system **EELTMGR** or with an email to **eeltmgr@eso.org**

For external users and developers: please send any problem regarding the ELT Linux DevEnv installation to the JIRA ticketing system associated with your project.