Installing Intel® Distribution for Python* and Intel® Performance Libraries with Anaconda*

By Robert C. (Intel) (https://software.intel.com/en-us/user/336264), published on May 28, 2016

We have worked with Continuum Analytics* to make it easy to use Intel® Distribution for Python (https://software.intel.com/en-us/python-distribution) and the Intel® Performance Libraries (https://software.intel.com/en-us/performance-libraries) (such as Intel® Math Kernel Library (Intel® MKL)) with the Conda* package manager and Anaconda Cloud*. You need at least conda 4.1.11, so first update your conda.

```
conda update conda
```

Tell conda to choose Intel packages over default packages, when available.

```
conda config --add channels intel
```

Installing the Intel® Distribution for Python*

We recommend that you create a new environment when installing. To install the core python3 environment, do:

```
conda create -n idp intelpython3_core python=3
```

If you want python 2 do:

```
conda create -n idp intelpython2_core python=2
```

If you want the full Intel distribution, replace the "core" package name with "full", like this for python3:

```
conda create -n idp intelpython3_full python=3
```

Then follow the usual directions for activating the environment. Linux/macOS users do:

```
source activate idp
```

and Microsoft Windows users do:

```
activate idp
```

You now have the core environment, including python, numpy, scipy,... You can use the usual conda install commands for additional packages. For example, to install intel sympy do:

```
conda install sympy
```

Non-intel packages are installed as usual. For example, to install affine do:

```
conda install affine
```

Available Intel packages can be viewed here: https://anaconda.org/intel/packages (https://anaconda.org/intel/packages)

Using Intel Conda* Packages with Continuum's Python*

If you want to install Intel packages into an environment with Continuum's python, do **not** add the "intel" channel to your configuration file because that will cause all your Continuum packages to be replaced with Intel builds, if available. Rather, specify the "intel" channel on the command line with "**-c intel**" parameter and the "**--no-update-deps**" flag to avoid switching other packages, such as python itself, to Intel's builds:

```
conda install mkl -c intel --no-update-deps
```

```
conda install numpy -c intel --no-update-deps
```

Installing the Intel® Performance Libraries

If you want to build a native extension that directly uses the performance libraries, then you will need to

obtain a development package that contains header files and static libraries. We have published them as conda packages for your convenience.

Make sure the Intel channel is added to your conda configuration (see above). Then install any of our available performance libraries using "conda install" as normal, such as:

```
conda install mkl-devel
```

The following table lists the available packages with a brief description for their contents:

Package Name	Lin-64	Lin-32	Win-64	Win-32	macOS-64	Description
mkl	X	X	X	X	X	Intel® Math Kernel Library (Intel® MKL) dynamic runtimes
mkl-devel	X	X	X	X	X	Intel® MKL dynamic runtimes and headers for building software
mkl-static	X	X	X	X	X	Intel® MKL static libraries and headers for building software
mkl-include	X	X	X	X	X	Intel® MKL headers only. Automatically installed along with development packages

For more complete information about compiler optimizations, see our Optimization Notice (/en-us/articles/optimization-notice#opt-en).

16 comments

Top

das, dipaditya (/en-us/user/1915423) said on Mar 21,2019

Conda update problem!!!!!

please Help!!

(/en-