ecCodes software

Downloading & Installation

Shahram Najm

Development Section
Forecast Department

ECMWF

Downloading and installing

- ecCodes uses CMake for compilation and installation

- CMake is a cross-platform free software program for managing the build process of software using a compiler-independent method

- Download ecCodes from:
  https://software.ecmwf.int/wiki/display/ECC/Releases

- ecCodes installation instructions:
  https://software.ecmwf.int/wiki/display/ECC/ecCodes+installation

Building with CMake

- Unpack distribution

```
tar -xzf eccodes-x.y.z-Source.tar.gz
```

- Create a separate directory where to build ecCodes:

```
mkdir build; cd build
```

- Run cmake pointing to the source and specify the installation location:

```
cmake  ../eccodes-x.y.z-Source \
-DCMAKE_INSTALL_PREFIX=/path/to/install
```

- Compile, test and install:

```
make
ctest
make install
```

Fortran API: compiling and linking

- To use the F90 API, you would typically build your Fortran programs like this:

```
gfortran -o prog.exe prog.f90 \
  -I/path/to/install/include \
  -L/path/to/install/lib \
  -Wl,-rpath,/path/to/install/lib \
  -leccodes_f90 -leccodes
```

- Assuming you use the GNU Fortran compiler
- Assuming you have installed the shared library
- If you installed the static library (with the cmake option: `-DBUILD_SHARED_LIBS=OFF`), then you do not need the “rpath”

C API: compiling and linking

- To use the C API, you would typically build your C programs like this:

```
gcc -o prog.exe prog.c \
  -I/path/to/install/include \
  -L/path/to/install/lib \
  -Wl,-rpath,/path/to/install/lib \
  -leccodes
```

- Assuming you use the GNU C compiler
- Assuming you have installed the shared library
- If you installed the static library (with the cmake option: `-DBUILD_SHARED_LIBS=OFF`), then you do not need the “rpath”

Python API

- For the Python API, no compiling/linking is required.
- All you need to do is point to the location of the ecCodes Python module.
- This is done with the environment variable PYTHONPATH

E.g.

```
export ECC_PY=/path/to/install/lib/python2.7/site-packages
export PYTHONPATH=$ECC_PY:$PYTHONPATH
```

- Assuming you have Python version 2.7.x
- This defines the location of the ecCodes Python module and augments python’s default search path for module files

Tools

- To use the tools (executables like bufr_dump), you need to extend the Unix PATH environment variable

E.g.

```
export ECC_BIN=/path/to/install/bin
export PATH=$ECC_BIN:$PATH
```

Finally…

- Now you have the full power of ecCodes at your fingertips!

Help and Support

- For issues, bugs and feature requests:
  Software.Support@ecmwf.int

- ecCodes Home page:
  https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home

- Please use the Forum for general discussions:
  https://software.ecmwf.int/wiki/display/ECC/Forums

☺