15.11.11

CPack

Peter F. Neher

What is CPack?

dkfz.

- Platform independent packaging system for software distributions
- Integrated with CMake
- It allows you to generate:
  - Linux RPM, deb and archive (tgz, zip, …) distributions of both binaries and source code.
  - Windows installer
  - Mac OS X packages

CPack without CMake

- **Can be used directly by specifying a CPackConfig.cmake**

```
SET(CPACK_CMAKE_GENERATOR "Unix Makefiles")
SET(CPACK_GENERATOR "STGZ;TGZ;TZ")
SET(CPACK_INSTALL_CMAKE_PROJECTS "/home/andy/vtk/CMake-bin;CMake;ALL;/”)
SET(CPACK_NSIS_DISPLAY_NAME "CMake 2.5")
SET(CPACK_OUTPUT_CONFIG_FILE "/home/andy/vtk/CMake-bin/CPackConfig.cmake")
SET(CPACK_PACKAGE_DESCRIPTION_FILE "/home/andy/vtk/CMake/Copyright.txt")
SET(CPACK_PACKAGE_DESCRIPTION_SUMMARY "CMake is a build tool")
SET(CPACK_PACKAGE_EXECUTABLES "ccmake;CMake")
SET(CPACK_PACKAGE_FILE_NAME "cmake-2.5.0-Linux-i686")
SET(CPACK_PACKAGE_INSTALL_DIRECTORY "CMake 2.5")
SET(CPACK_PACKAGE_INSTALL_REGISTRY_KEY "CMake 2.5.0")
SET(CPACK_PACKAGE_NAME "CMake")
SET(CPACK_PACKAGE_VENDOR "Kitware")
SET(CPACK_PACKAGE_VERSION "2.5.0")
SET(CPACK_PACKAGE_VERSION_MAJOR "2”)
…
```

Using CPack with CMake

dkfz.

- INCLUDE(CPack)
- CMake will generate CPack configuration file automatically.
- Generates new target called "package" in your build system.
- CPack will use CMake's install mechanism to automatically populate the package.
- Just call “make package” in the build directory.

Example

HelloWorld.cxx

```
#include <iostream>
int main(int, char *[])
{
  std::cout << “Hello World! << std::endl;
  return EXIT_SUCCESS;
}
```

CMakeLists.txt

```
PROJECT(HelloWorld)
ADD_EXECUTABLE(HelloWorld HelloWorld.cxx)
INSTALL(TARGETS HelloWorld DESTINATION hello)
SET(CPACK_GENERATOR "TGZ”)
INCLUDE(CPack)
```

The END