



ILC software build and installation tools

Jan Engels
DESY

ILD Workshop - DESY-Zeuthen
15th January 2008



ILC Software Installation



- **ilcinstall (current version v01-03)**
 - Python script for installing the LDC Software framework
 - Supports external packages: CLHEP, GSL, QT, ...
 - Automatically checks package dependencies
 - Installs without user-intervention
 - e.g. Nightly builds
 - **Fully configurable** (via configuration file)
 - Use/Link/Install packages
 - Dynamic assignment of environment/cmake variables
 - Modular & Extensible
 - Abstract class for Marlin modules (MarlinPKG)
 - **CMake** is now the default build manager
 - 'Old' Makefiles will not be supported from the next release onwards



ILC Software Releases (AFS)



- `/afs/desy.de/group/it/ilcsoft/`

- **LCIO/**
 - v01-09
- **GEAR/**
 - v00-08
- **CLHEP/**
 - 2.0.2.2
 - 2.0.3.1
- **GSL/**
 - 1.6
 - 1.8
- **CERNLIB/**
 - 2006
- ...

Software **with** dependencies

- `/afs/desy.de/group/it/ilcsoft/v01-03/`

- **LCIO/v01-09** -> link to `../LCIO/v01-09`
- **GEAR/v00-08** -> link to `../GEAR/v00-08`
- **CLHEP/2.0.3.1** -> link to `../CLHEP/2.0.3.1`
- **GSL/1.8** -> link to `../GSL/1.8`
- **CERNLIB/2006** -> link to `../CERNLIB/2006`
- **Marlin/v00-09-10**
- **MarlinUtil/v00-05**
- **MarlinReco/v00-05**
- ...

Software with
no dependencies



Using Release Configuration files



- SL3/SL4 users can use **ilcinstall** to install locally the packages they need to develop on and link against the AFS reference installation for resolving dependencies.
- 1st Download ilcinstall from:
 - http://www-zeuthen.desy.de/lc-cgi-bin/cvswweb.cgi/ilcinstall/ilcinstall.tar.gz?cvsroot=ilctools;only_with_tag=v01-03;tarball=1
- Edit the appropriate file, e.g. “**releases/release_v01-03.cfg**”
 - **Change installation path**
 - `ilcsoft = ILCSoft("$HOME/my_ilc")`
 - **Install the packages you want to work on**
 - `ilcsoft.install(PandoraPFA("HEAD"))`
 - `ilcsoft.module("PandoraPFA").download.type="ccvssh"`
 - `ilcsoft.module("PandoraPFA").download.username="engels"`
 - `ilcsoft.module("PandoraPFA").download.password="*****"`
 - `ilcsoft.module("PandoraPFA").download.login2Server=True`
 - `ilcsoft.install(MarlinReco("HEAD"))`
 - ...
 - **Change dependencies from **install** to **link****
 - `ilcsoft.link(LCIO(ilcPath+ "lcio/v01-09"))`
 - `ilcsoft.link(GEAR(ilcPath+ "gear/v00-08"))`
 - ...
 - **Run ilcinstall**
 - `ilcsoft-install releases/release_v01-03.cfg -i`

Add this 2 lines if your ~/.cvspass doesn't already contain the cvs password for your username
`/1 :pserver:engels@cvssrv.ifh.de:2405/marlinreco A^dZ`



CMake



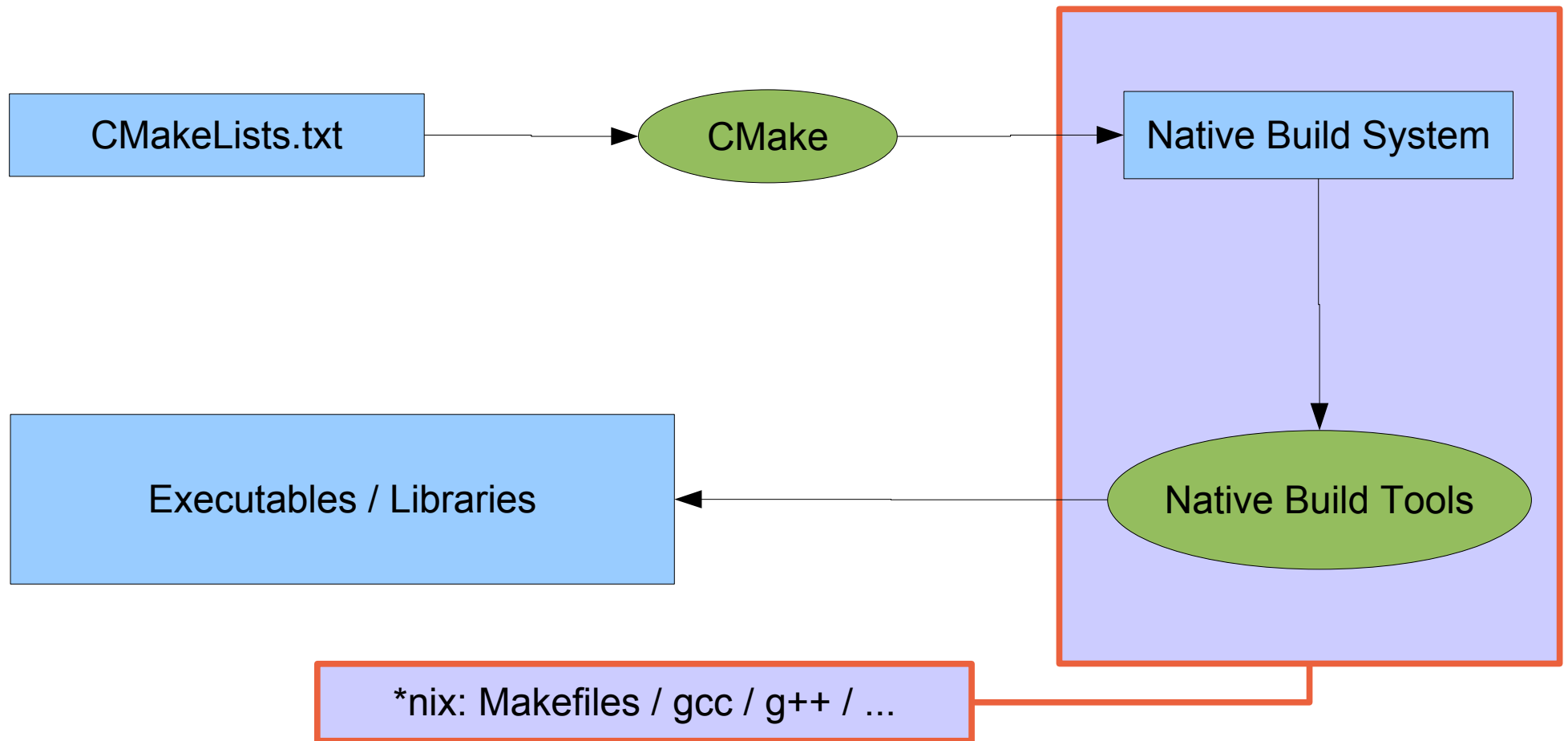
- **CMake:**

- **Cross-platform** build-system manager
- Generates **native build environments**
 - UNIX/Linux -> Makefiles
 - Windows -> VS Projects/Workspaces
 - Apple -> Xcode
- Takes as input **simple text files (CMakeLists.txt)**
 - Simple, intuitive syntax!
 - Support for regular expressions (*nix style)
- **In-source** vs **Out-of-source** builds
- Flexible, extensible & powerful
 - Macros, Modules for finding/configuring software, custom targets
- **Integrated Testing** & Packaging (**Ctest**, **CPack**)
- Open-Source :)
- Check www.cmake.org!

Successor of GNU auto-tools



CMake's Build-System Generator



Adapting your own Marlin Plugin to CMake



- Copy 4 files from `$MARLIN/examples/mymarlin`
 - **CmakeLists.txt**
 - change the project name and add missing dependencies
 - `PROJECT(mymarlin)`
 - `SET(${PROJECT_NAME}_DEPENDS "Marlin;LCIO;GEAR;..."`
 - Define additional options (if needed)
 - Ex: `OPTION(MARLIN_GUI "Set to ON to build Marlin GUI" OFF)`
 - Don't forget to also add the options at the end of the file in the 'status display' and 'write to cache' sections
 - `MESSAGE(STATUS "MARLIN_GUI = ${MARLIN_GUI}")`
 - `SET(MARLIN_GUI "${MARLIN_GUI}" CACHE BOOL "Set to ON to build Marlin GUI" FORCE)`
 - **mymarlinConfig.cmake.in**
 - rename to `<MyPlugin>Config.cmake`
 - **BuildSetup.cmake**
 - Edit & change according to your system setup!
 - You can also add here an entry for your project specific options
 - (e.g. the option `MARLIN_GUI` to switch ON/OFF building the GUI)
 - **cmake_uninstall.cmake.in**
 - **No changes needed!**

Prerequisites in
Marlin **README**



Adapting your own Marlin Plugin to CMake



- In case your plugin depends on some other external package that is not part of CmakeModules
 - Edit **CmakeLists.txt** and add before the `ADD_LIBRARY()` command
 - `INCLUDE_DIRECTORIES("/path/to/pkg/include")`
 - `LINK_DIRECTORIES("/path/to/pkg/lib")`
 - `LINK_LIBRARIES(libname) # e.g. libfoo.so -> LINK_LIBRARIES(foo)`



Building with CMake



- CD into source directory
 - `cd <path_to_your_package>`
- Create a build directory (“**out-of-source** build” concept)
 - `mkdir build ; cd build`
- Configure package
 - `cmake [options] <source_tree>`
 - e.g. `cmake -C ../BuildSetup.cmake ..`
- Build package
 - `make`
- Install it! (into path defined by `${CMAKE_INSTALL_PREFIX}`)
 - `make install`

Similar to Auto Tools

BuildSetup.cmake is just an **example** cmake script containing the project's options so you don't need to define them all on the command line. To use it you first have to **edit & modify it according to your system setup!**

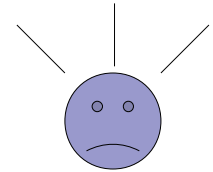
The last 2 steps (make && make install) can be merged to:
“**make install**”



Configuring Multiple Projects



Set `CMAKE_MODULE_PATH` in **10 different BuildSetup.cmake** files??!!

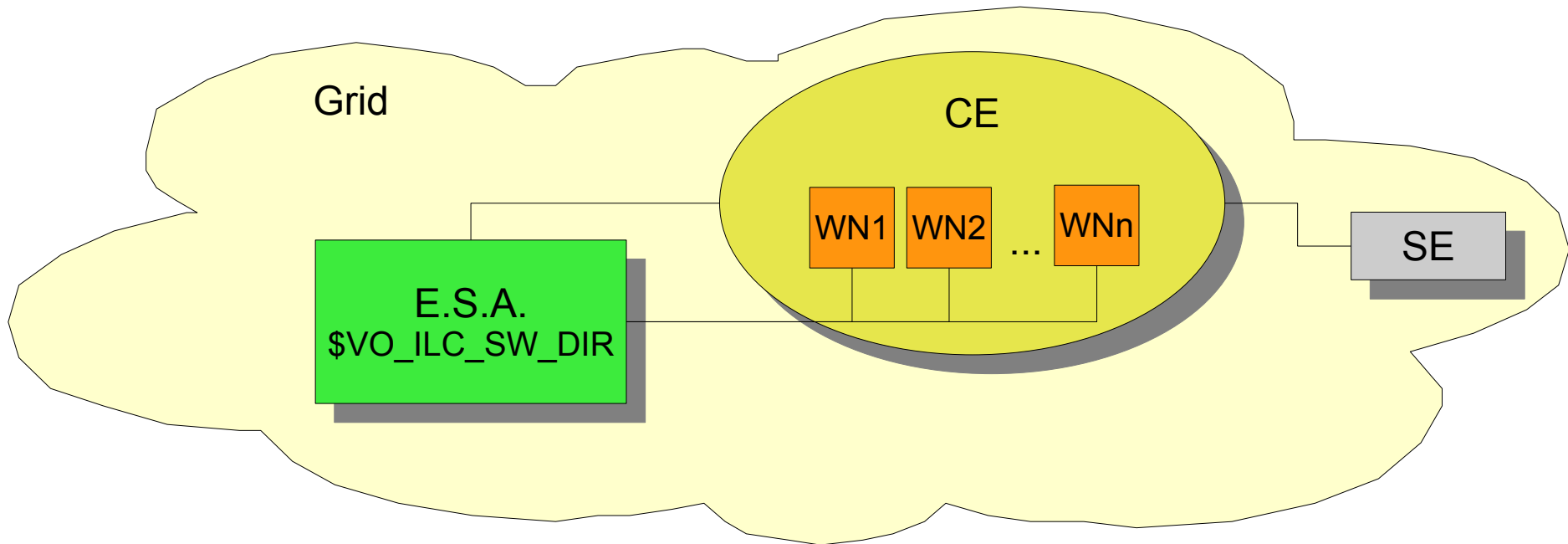


- BuildSetup.cmake vs ILCSoft.cmake
 - BuildSetup.cmake is just an **example** cmake script (not necessarily up-to-date or even working!)
 - If you have 10 different projects sharing some common options you won't definitely want to set them in all BuildSetup.cmake files!! (e.g. `CMAKE_MODULE_PATH`)
 - This also happens with `PKG_HOME` variables.
 - So, there is the need to define some variables **globally** between projects!
 - For this purpose ilcinstall automatically generates a file called `ILCSoft.cmake` which contains all **path entries** to packages defined in it's configuration file + the `CMAKE_MODULE_PATH` variable.
 - It is also possible to change project specific options in BuildSetup.cmake and use ILCSoft.cmake for resolving dependencies (see example below)
- CMake with **more than one -C** option:
 - Ex: `cmake -C ../BuildSetup.cmake -C $LCSOFT/ILCSoft.cmake -D GSL_HOME=/ilc/gsl/1.8 ..`
 - **Latest scripts (called with -C) overwrite values from previous ones!!**
 - **Command line definitions (-D) have biggest priority!!**

```
$LCSOFT=/afs/desy.de/group/it/ilcsoft/v01-03
```



Grid Installation



- ilcsoft v01-03 installation will be available on the grid **soon!**
- Make use of ILC Experiment Software Area
 - Environment variable exists per site for referring this area: `${VO_ILC_SW_DIR}`
 - Visible to all Worker-nodes
- Installation will be available at:
 - `${VO_ILC_SW_DIR}/ilcsoft/vXX-YY`
 - `export LD_LIBRARY_PATH=${VO_ILC_SW_DIR}/ilcsoft/vXX-YY/lib:${LD_LIBRARY_PATH}`



Status of the ILCSoft Install/Build-System



ILCInstall:

- Current version: **v01-03**
- Currently Supported modules:
 - LCIO, GEAR, LCCD, RAIDA, Marlin, MarlinUtil, MarlinReco, CED, CEDViewer, PandoraPFA, LCFIVertex, SiliconDigi, Overlay, Eutelescope, **CKFit**, **MarlinTPC**, **Mokka**, CLHEP, GSL, CERNLIB, CondDBMySQL, QT, CMake, CmakeModules, ROOT, Geant4, Java, AIDAJNI/JAIDA, MySQL

Recently added
No install/build support

CMake:

- **CmakeModules** - Current version: **v01-05**
- Currently Supported:
 - LCIO, GEAR, LCCD, RAIDA, Marlin, MarlinUtil, MarlinReco, CED, CEDViewer, PandoraPFA, LCFIVertex, SiliconDigi, Overlay, Eutelescope, **CKFit**, **MarlinTPC**, CLHEP, CERNLIB, CondDBMySQL, GSL, ROOT, Java, AIDAJNI/JAIDA



Summary & Outlook



- **Summary:**

- **Nightly builds** being done for testing the ILC software under **SL3/SL4**
- Recent test-installation under SL5 completed successfully
- **Shared libraries & Marlin plugins** support with **CMake**
- Installation and build tools have now reached a stable level
- We were able to remove some dependencies
 - LCIO no longer depends on CLHEP
 - HepPDT (Marlin, MarlinUtil, CEDViewer)

- **Outlook:**

- Installation of ilcsoft on the **Grid... High Priority!**
- Exchange support of 'old Makefiles' for better integration of cmake in ilcinstall
 - e.g. automatic generate BuildSetup.cmake?
 - Old version of ilcinstall will still be available
- Improve integrated testing in cmake (**ctest**)

Thank you! Your feedback is welcome!

