

Modellistica Medica

Maria Grazia Pia

INFN Genova

Scuola di Specializzazione in Fisica Sanitaria

Genova

Anno Accademico 2002-2003

Lezione 26

Installing Geant4

Lecture material by Gabriele Cosmo

Geant 4

Software Installation, release 4.1

Gabriele Cosmo, Gabriele.Cosmo@cern.ch

Geant4 Users' Workshop Tutorial

Salamanca - July 15-19, 2002

Outline

- Supported platforms & compilers
- External software packages and tools
- Working area and installation area
- Toolkit installation
 - Configuring the environment manually
 - Using the *Configure* installation script
- Building an executable
- Integrating Geant4 in a software framework

Supported platforms & compilers

- UNIX systems

- DEC-OSF/1 v.4.0, cxx 6.1
 - G4SYSTEM: DEC-cxx
- SUN-SunOS v.5.7, CC v.5.2 (WS6)
 - G4SYSTEM: SUN-CC

(*)



- Linux systems

- RedHat 6.1, g++ egcs 1.1.2
 - G4SYSTEM: Linux-egcs
- RedHat 7.2 or greater, g++ gcc 2.95.2
 - G4SYSTEM: Linux-g++



- Windows systems

- Win/NT/98/2000/XP and CygWin, MSVC++ 6.0 SP5
 - G4SYSTEM: WIN32-VC



() Tested but no longer under support !*

Required software

- A UNIX shell and related basic UNIX commands
- C++ compiler
 - Optional: FORTRAN compiler (f77/g77) is required for building utility tools in the *g3tog4* module
- GNU Make
 - GNU g++ is required for dependencies pre-processing
- CLHEP library
- Native STL (Standard Template Library)
- The Geant4 toolkit source code

External software packages - 1

Visualization/GUI tools (optional):

- X Windows
- OpenGL or MesaGL
- DAWN (PostScript renderer)
 - DAVID (Geometry debugging tool based on DAWN)
- Open Inventor or HEP Inventor
 - requires OpenGL/MesaGL and the Hepvis module
- OPACS and Open Scientist
 - interactive environment, including GUI
- Momo
 - Tcl/Tk or Java-based GUI environment
 - GGE, GPE graphics editors
- VRML browser

External software packages - 2

Module and Tools for analysis (optional)

- AIDA (Abstract Interfaces for Data Analysis)
 - JAS (Java Analysis Studio)
 - Lizard (AIDA Interactive Analysis Environment)
 - Open Scientist (Interactive Analysis Environment)

Persistency (optional) (*)

- HepODBMS module
- An ODMG compliant database system
 - Objectivity/DB

() Major design iteration foreseen for release 5.0*

Working area & Installation area

- Why two different areas ?
 - To allow centralized installation of the Geant4 kernel libraries and related sources in a multi-user environment
 - To decouple user-developed code and applications from the kernel
 - To allow an easy integration of the Geant4 software in an existing software framework
 - To allow multiple installations of the kernel and user code
- Working and Installation area can be the same
- Are controlled by two environment variables
 - **G4WORKDIR** and **G4INSTALL**

Configuring the environment:

the manual approach for installation

- Identify the system used for the installation
 - G4SYSTEM
- Identify the area of installation (i.e. path where the source code and the kernel libraries should be based)
 - G4INSTALL
 - Optionally, specify a different path for the kernel libraries and/or the temporary object files
 - G4LIB, G4TMP
 - Optionally, specify a different path for exporting of source header files
 - G4INCLUDE

Configuring the environment: the manual approach for installation

• Specify the path of installation for CLHEP

- CLHEP_BASE_DIR

- should point to the area where `include/` and `lib/` are placed from the standard CLHEP installation procedure
 - Paths can be customised: `CLHEP_INCLUDE_DIR`, `CLHEP_LIB_DIR`
- the CLHEP library name is assumed to be: `[lib]CLHEP[.a|.lib]`
 - A different name can be explicitly specified: `CLHEP_LIB`

• Specify the graphics/UI drivers to install

- `G4VIS_BUILD_<name>_DRIVER`
- `G4UI_BUILD_<name>_DRIVER`
- the path to the related graphics/(G)UI packages, if required
 - Ex. - `OGLHOME` for OpenGL drivers

Configuring the environment: the manual approach for installation

- Specify installation specific attributes
 - G4DEBUG
 - To build libraries including debug symbolic information
 - By default, optimised mode is selected
 - G4LIB_BUILD_SHARED
 - To specify if to build kernel libraries as shared libraries
 - Static archive libraries are built by default
 - Adding also G4LIB_BUILD_STATIC will build both
 - G4_NO_VERBOSE
 - For better performance, verbosity code can be left out by defining this flag. The default is with verbosity on

Starting the installation

- Choose the installation layout
 - Maximum granularity of libraries (*granular* libraries)
 - Ideal for developers and local installations
 - Link list of libraries automatically generated
 - Triggered with “gmake” from `$G4INSTALL/source`
 - Category compound libraries (*global* libraries)
 - Convenient for a centralized multi-users installation
 - Default for shared libraries builds
 - Triggered with “gmake global” from `$G4INSTALL/source`
- Installing source header files
 - `G4INCLUDE` defines the installation path
 - Triggered by “gmake includes” from `$G4INSTALL/source`

Configuring the environment to use Geant4

- Specify the working area: `G4WORKDIR`
 - If not, Geant4 assumes `G4INSTALL` as the working area
 - Products of application builds are placed in `$G4WORKDIR`
 - Binaries in `$G4WORKDIR/bin`
 - Object files and other temporary files in `$G4WORKDIR/tmp`
- Specify which graphics drivers, (G)UI drivers you want to use from the current installation (optional)
 - `G4VIS_USE_<name>`
 - `G4UI_USE_<name>`
- Specify paths to physics data (optional)
 - `G4LEVELGAMMADATA, G4LEDDATA, G4RADIOACTIVEDATA, NeutronHPCrossSections`

Using the `Configure` script for installation & configuration

- The `configure` script guides through the whole installation process described so far by defining the proper environment and triggering the actual installation

```
./Configure -install
```

- Once the environment has been configured, `configure` stores the current installation setup

- The installation setup will become the default for the current installation, in case future changes to the installation are necessary

- Once the installation is complete, `configure` can be used to generate shell scripts for configuring the user environment to build a Geant4 application according to the current installation

```
./Configure
```

- Generates `env[.sh/.csh]` scripts in the user's current directory
- It assumes the user specifies a working directory (`G4WORKDIR`)
 - In case not, the user's home directory is set as default `G4WORKDIR` path

Building an executable

- Configure the environment according to the current installation
 - Source or integrate the shell script generated by `Configure`
- Define the working area (`G4WORKDIR`)
- Build any of the available examples:

```
cp -r $G4INSTALL/examples $G4WORKDIR
cd $G4WORKDIR/examples/novice/N01
gmake
```


Integrating Geant4 in a framework

- Consider Geant4 as an external software package
- Well define its area of installation
 - For global libraries and source header files
- Choose an installation setup which best matches the project needs
- Adopt or integrate a configuration script reflecting the current installation