Ginkgo CADx compilation with unmodified libraries.

Tool dependencies:
> CMake >= 2.8

Library dependencies:
> GTK+ 2.x, OpenSSL 0.9.x, DCMTK 3.6.x, ITK 3.20.x, VTK 5.6.x, WxWidgets >=2.8.11 (OpenGL enabled)

Procedure:
> Download your distribution specific devel packages of required library dependencies.
> cd Ginkgo_CADx-*/src
> mkdir build
> cd build
> cmake ../ -DCMAKE_BUILD_TYPE=Release -DUSE_PATCHED_LIBS:BOOL=FALSE
-DUSE_CUSTOM_WX:BOOL=FALSE -DUSE_CUSTOM_VTK:BOOL=FALSE
-DUSE_CUSTOM_ITK:BOOL=FALSE -DUSE_CUSTOM_DCMTK=FALSE
-DCUSTOM_PACKAGE:BOOL=FALSE
> make

Warning This build is not fully tested on any plattform, so please send us feed back with any information, suggestion or patch (if possible) you could provide.

Midleware dependencies compilation

Windows:
Tools and deps:
> Ms. Visual Studio 2008 C++ (Express or best)
> CMake >=2.6

NOTE: Build operations may be done on Debug and relesase modes.

wxWidgets:
> Download wxMSW-2.8.11
> Apply wxWidgets-2.8.11.diff patches.
> Open solution with Visual Studio 2008 C++ (wxWidgets\build\mws\wx.sln). It will ask you to convert some files, have to convert them.
> Build DLL UNICODE library target.
> Open "setup.h" file and modify the following constant:
#define wxUSE_GLCANVAS 0, set it to 1.
> Copy includes and libs to ginkgo dll tree. Copy dll files to path where Ginkgo CADx is executed.

DCMTK:
> Download DCMTK 3.6.0
> Apply dcmtk-3.6.0.diff patches.
> Build VS project with Cmake
> Change from the code generation, rhe run time library from MTd to MDd on debug compile mode, and from MT to MD on release compile mode
> Copy includes and libs to ginkgo dll tree
VTK:
> Download and expand VTK.
> Build from Cmake:
>> Mark: advanced

Set the variables:
      BUILD_SHARED_LIBS = ON
      VTK_USE_GUISUPPORT = ON
      VTK_USE_PARALLEL = ON
Open solution with Visual Studio 2008 C++ (VTK\VTK.sln)
Build dynamic library target.
Copy includes and libs to ginkgo dll tree. Copy dll files to path where Ginkgo CADx ix executed.

ITK:
Expand InsightToolkit-3.20.0.tar.gz
Apply ITK-3.20.0.diff patches.
Build VS project with Cmake
    Mark: advanced
    Set the variables:
      BUILD_SHARED_LIBS = ON
      ITK_USE_PATENTED = ON
      VNL_CONFIG_ENABLE_SSE2 = ON
Open solution with Visual Studio 2008 C++ (ITK\ITK.sln)
Build Dynamic library target.
Copy includes and libs to ginkgo dll tree. Copy dll files to path where Ginkgo CADx ix executed.

Cairowin32:
Download cairo 1.8.10 and pixman 0.17.10 from:
    http://cairographics.org/releases/
Create a static library from scratch.
Copy includes and libs to ginkgo dll tree.

OpenSSL:
Download OpenSSL 1.0.0d Windows binary distribution from:
    http://www.slproweb.com/products/Win32OpenSSL.html
Copy includes and MD static libraries to ginkgo dll tree.

Mac OS X:
Tools and deps:
    CMake >=2.6
    XCode
    GCC 4.2

wxWidgets:
Expand and apply apply wxWidgets-2.8.11.diff patches.

Debug:
    ./configure --enable-monolithic --enable-dynlib --disable-shared --enable-unicode --enable-debug --enable-dataobj --enable-dataviewctrl --prefix=/opt/local/wxdebug

Release:
    ./configure --enable-monolithic --enable-dynlib --disable-shared --enable-unicode --disable-debug -enable-optimise --enable-dataobj --enable-dataviewctrl --prefix=/opt/local/wxrelease

make
sudo make install

Copy includes and libs to ginkgo dll tree.

DCMTK:

Download latest version with git:

git clone http://git.dcmtk.org/dcmtk.git <dir>

Apply dcmtk.git.diff patches.

export CFLAGS=-m32
export CPPFLAGS=-m32
export CXXFLAGS=-m32
./configure --with-openssl --with-zlib --with-libpng --with-libxml --enable-static --disable-shared –without-png
make
sudo make install

Copy includes and libs to ginkgo dll tree.

VTK:

Expand and export following variables in terminal:

export CFLAGS=-m32
export CPPFLAGS=-m32
export CXXFLAGS=-m32

Debug:

cmake ../VTK-* -DBUILD_TESTING:BOOL=OFF
-DVTK_DEBUG_LEAKS:BOOL=ON -DVTK_USE_COCOA:BOOL=OFF
-DVTK_USE_CARBON:BOOL=ON -DCMAKE_BUILD_TYPE=Debug
-DCMAKE_INSTALL_PREFIX=/opt/local/vtkdebug

Release:

cmake ../VTK-* -DBUILD_TESTING:BOOL=OFF
-DVTK_DEBUG_LEAKS:BOOL=OFF -DVTK_USE_COCOA:BOOL=OFF
-DVTK_USE_CARBON:BOOL=ON -DCMAKE_BUILD_TYPE=Release
-DCMAKE_INSTALL_PREFIX=/opt/local/vtkrelease
make
sudo make install

Copy includes and libs to ginkgo dll tree.

ITK:

Expand and apply apply ITK-3.20.0.diff patches.

Debug:

cmake ../ITK-* -DBUILD_EXAMPLES:BOOL=OFF -DBUILD_TESTING:BOOL=OFF
-DITK_USE_PATENTED:BOOL=ON -DVNL_CONFIG_ENABLE_SSE2:BOOL=ON
-DCMAKE_OSX_ARCHITECTURES=i386 -DCMAKE_BUILD_TARGET=Debug
-DCMAKE_INSTALL_PREFIX=/opt/local/itkdebug

Release:

cmake ../ITK-* -DBUILD_EXAMPLES:BOOL=OFF -DBUILD_TESTING:BOOL=OFF
-DITK_USE_PATENTED:BOOL=ON -DVNL_CONFIG_ENABLE_SSE2:BOOL=ON
-DCMAKE_OSX_ARCHITECTURES=i386 -DCMAKE_INSTALL_PREFIX=/opt/local/itkrelease
make
sudo make install

Copy includes and libs to ginkgo dll tree.

Linux:

Tools and deps:

CMake >=2.6 :   sudo apt-get install cmake

GTK-2.0-dev :   sudo apt-get install libgtk2.0-dev
libx11-dev : sudo apt-get install libx11-dev
libxt-dev :   sudo apt-get install libxt-dev
libxml2-dev :   sudo apt-get install libxml2-dev
libssl-dev :   sudo apt-get install libssl-dev
libwrap0-dev :   sudo apt-get install libwrap0-dev
GCC 4.2 :   sudo apt-get install gcc

if it show an error like this:

```
metaemotion@ubuntu:~$ sudo apt-get install gcc-4.2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
El paquete gcc-4.2 no está disponible, pero algún otro paquete hace referencia
a él. Esto puede significar que el paquete falta, está obsoleto o sólo se
encuentra disponible desde alguna otra fuente
E: El paquete gcc-4.2 no tiene candidato para su instalación
metaemotion@ubuntu:~$ gcc -v
Using built-in specs.
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 4.4.3-4ubuntu5' -
-with-bugurl=file:///usr/share/doc/gcc-4.4/README.Bugs --enable-languages=c,c++,
fortran,objc,obj-c++ --prefix=/usr --enable-shared --enable-multiarch --enable-l
inker-build-id --with-system-zlib --libexecdir=/usr/lib --without-included-gette
xt --enable-threads=posix --with-gxx-include-dir=/usr/include/c++/4.4 --program-
suffix=-4.4 --enable-nls --enable-clocale=gnu --enable-libstdcxx-debug --enable-
plugin --enable-objc-gc --disable-werror --with-arch-32=i486 --with-tune=generic
 --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --ta
rget=x86_64-linux-gnu
Thread model: posix
gcc version 4.4.3 (Ubuntu 4.4.3-4ubuntu5)
metaemotion@ubuntu:~$
```

Maybe you have gcc already installed,
so type "gcc -v" to see the actual version.

OpenGL :   sudo apt-get install build-essential
                   sudo apt-get install libgl1-mesa-dev
                   sudo apt-get install libglu1-mesa-dev
                   sudo apt-get install freeglut3-dev
chrpath :   sudo apt-get install chrpath

WxWidgets:
Download wxWidgets-2.8.11.

Apply wxWidgets-2.8.11.diff patches. Terminal should show something like this:

```
patching file include/wx/aui/auibar.h
patching file include/wx/aui/auibook.h
patching file include/wx/intl.h
patching file include/wx/prntbase.h
patching file src/aui/auibar.cpp
patching file src/aui/auibook.cpp
patching file src/common/intl.cpp
patching file src/common/prntbase.cpp
patching file src/generic/dirctrlg.cpp
patching file src/generic/grid.cpp
patching file src/gtk/dataview.cpp
patching file src/gtk/frame.cpp
patching file src/tiff/tif_dirread.c
patching file src/tiff/tif_warning.c
```

Debug:
./configure --enable-monolithic --enable-dynlib --enable-shared --enable-unicode --enable-debug --with-opengl --enable-dataobj --enable-dataviewctrl --disable-compat26 –prefix=/opt/local/wxdebug

make
sudo make install
make clean

Release:
./configure --enable-monolithic --enable-dynlib --enable-shared --enable-unicode --enable-optimise --disable-debug --with-opengl --enable-dataobj --enable-dataviewctrl --disable-compat26 --prefix=/opt/local/wxrelease

make
sudo make install

Copy includes and libs to ginkgo dll tree from files created with make install located in "/opt/local/".

VTK:
Expand VTK source archive.

make
sudo make install
make clean

Debug:
cmake ../VTK* -DBUILD_TESTING:BOOL=OFF -DBUILD_SHARED_LIBS:BOOL=ON -DVTK_DEBUG_LEAKS:BOOL=ON -DCMAKE_BUILD_TARGET=Debug -DCMAKE_INSTALL_PREFIX=/opt/local/vtkdebug

make
sudo make install
make clean

Relase:
        cmake ../VTK* -DBUILD_TESTING:BOOL=OFF
-DBUILD_SHARED_LIBS:BOOL=ON -DVTK_DEBUG_LEAKS:BOOL=OFF
-DCMAKE_BUILD_TARGET=Release -DCMAKE_INSTALL_PREFIX=/opt/local/vtkrelease

        make
        sudo make install

        Copy includes and libs to ginkgo dll tree.

ITK:

Expand and apply apply ITK-3.20.0.diff patches.Terminal should show something like this:

```
patching file Utilities/gdcm/src/CMakeLists.txt
patching file Utilities/gdcm/src/gdcmFile.cxx
patching file Utilities/gdcm/src/gdcmJPEGFragment.cxx
patching file Utilities/gdcm/src/gdcmJPEGFragment.h
patching file Utilities/gdcm/src/gdcmJPEGFragmentsInfo.cxx
patching file Utilities/gdcm/src/gdcmJpeg.cxx
patching file Utilities/gdcm/src/gdcmJpeg12.cxx
patching file Utilities/gdcm/src/gdcmJpeg16.cxx
patching file Utilities/gdcm/src/gdcmJpeg8.cxx
patching file Utilities/gdcm/src/gdcmJpegLS.cxx
patching file Utilities/gdcm/src/gdcmJpegLScolortransform.h
patching file Utilities/gdcm/src/gdcmJpegLSconfig.h
patching file Utilities/gdcm/src/gdcmJpegLScontext.h
patching file Utilities/gdcm/src/gdcmJpegLScontextrunmode.h
patching file Utilities/gdcm/src/gdcmJpegLSdecoderstrategy.h
patching file Utilities/gdcm/src/gdcmJpegLSdefaulttraits.h
patching file Utilities/gdcm/src/gdcmJpegLSencoderstrategy.h
patching file Utilities/gdcm/src/gdcmJpegLSheader.cpp
patching file Utilities/gdcm/src/gdcmJpegLSheader.h
patching file Utilities/gdcm/src/gdcmJpegLSinterface.cpp
patching file Utilities/gdcm/src/gdcmJpegLSinterface.h
patching file Utilities/gdcm/src/gdcmJpegLSjpegls.cpp
patching file Utilities/gdcm/src/gdcmJpegLSlookuptable.h
patching file Utilities/gdcm/src/gdcmJpegLSlosslesstraits.h
patching file Utilities/gdcm/src/gdcmJpegLSprocessline.h
patching file Utilities/gdcm/src/gdcmJpegLSpublictypes.h
patching file Utilities/gdcm/src/gdcmJpegLSscan.h
patching file Utilities/gdcm/src/gdcmJpegLSstdafx.cpp
patching file Utilities/gdcm/src/gdcmJpegLSstdafx.h
patching file Utilities/gdcm/src/gdcmJpegLSstreams.h
patching file Utilities/gdcm/src/gdcmJpegLSutil.h
patching file Utilities/gdcm/src/gdcmPixelReadConvert.cxx
patching file Utilities/itkjpeg/jdmarker.c.orig
```
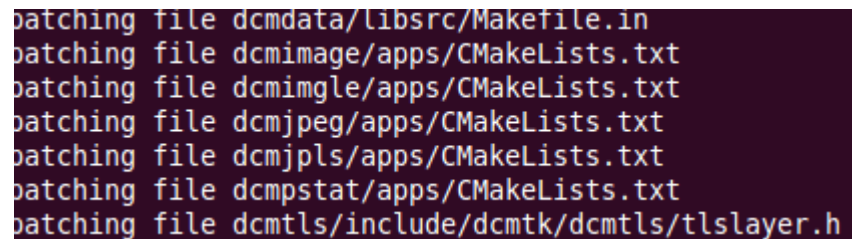
        Debug:
        cmake ../InsightToolkit-* -DBUILD_EXAMPLES:BOOL=OFF
-DBUILD_SHARED_LIBS:BOOL=ON -DBUILD_TESTING:BOOL=OFF
-DITK_USE_PATENTED:BOOL=ON -DVNL_CONFIG_ENABLE_SSE2:BOOL=ON
-DCMAKE_BUILD_TARGET=Debug -DCMAKE_INSTALL_PREFIX=/opt/local/itkdebug

Release:
cmake ../InsightToolkit-* -DBUILD_EXAMPLES:BOOL=OFF
-DBUILD_SHARED_LIBS:BOOL=ON -DBUILD_TESTING:BOOL=OFF
-DITK_USE_PATENTED:BOOL=ON -DVNL_CONFIG_ENABLE_SSE2:BOOL=ON
-DCMAKE_BUILD_TARGET=Release -DCMAKE_INSTALL_PREFIX=/opt/local/itkrelease

make
sudo make install

Copy includes and libs to ginkgo dll tree.

DCMTK:
Expand and apply apply dcmtk-3.6.0.diff patches. Terminal should show something like this:

Debug:
cmake ../dcmtk-3.6.0 -DBUILD_SHARED_LIBS:BOOL=ON
-DDCMTK_WITH_ZLIB:BOOL=ON -DDCMTK_WITH_TIFF:BOOL=OFF
-DCMAKE_BUILD_TARGET=Debug -DCMAKE_INSTALL_PREFIX=/opt/local/dcmtkdebug
Release:
mkdir dcmtkbuildrelease
cd dcmtkbuildrelease
cmake ../dcmtk-3.6.0 -DBUILD_SHARED_LIBS:BOOL=ON
-DDCMTK_WITH_ZLIB:BOOL=ON -DDCMTK_WITH_TIFF:BOOL=OFF
-DCMAKE_BUILD_TARGET=Release -DCMAKE_INSTALL_PREFIX=/opt/local/dcmtkrelease

make
sudo make install

Copy includes and libs to ginkgo dll tree.

Ginkgo CADx compilation

All:

For libraries, plugins and langage translations to be provided as "bundle" (with executable), the following structure is required*:

    executable_dir/<ginkgo_executable>
    executable_dir/GinkgoCADX.so*
    executable_dir/<wxWidgets dynamic libraries>
    executable_dir/<vtk dynamic libraries>
    executable_dir/<itk dynamic libraries>
    executable_dir/<dcmtk dynamic libraries>
    executable_dir/lang/<langcode>/<mo files>
    executable_dir/Plugins/<Ginkgo CADx extension dynamic libraries>

In MacOS X this structure is slight different:

    Ginkgo CADx.app/Contents/Info.plist
    Ginkgo CADx.app/Contents/PkgInfo
    Ginkgo CADx.app/Contents/MacOS/Ginkgo_CADx
    Ginkgo_CADx.app/Contents/MacOS/<dynamic libraries>
    Ginkgo_CADx.app/Contents/PlugIns/<Ginkgo CADx extension dynamic libraries>
    Ginkgo_CADx.app/Contents/Resources/lang/<langcode>/<mo files>

Windows:

    Deps: Ms. Visual Studio 2008 C++ (Express or best)
    Open src/ginkgo/ginkgo.sln with Ms. Visual Studio and select "buid".
    Opening Ginggo proyect we should manually include a file because its name has changed:
        "gingocadx.vcproj"

    <u>Compiling Debug Mode:</u>
    Will appear an error with the file "conformance.xml" caused by space characters on windows copying process. We must to modify the properties from the file, setting:
        Custom Build Step -> General -> Command Line: **copy "$(InputPath)" "$(OutDir)"**
    Some warnings will be avoided adding some directives, for example:
        #if !defined(HAVE_STATIC_CAST)
          #define HAVE_STATIC_CAST 1
        #endif
    Using VTK-5.6.1 must be deleted a function declaration from the vtkSmartVolumeMapper file, because its a more recently version of vtk.
    Some open_cv files will provocate some problems, so we must delete them from the fooextension and lightvisualizator extensions folders.

    <u>Compiling Release Mode:</u>
    The same error from conformance.xml file will appear, but we can solve it following exactly the same steps which appear on Debug Mode.

    When compiling process has been sucessful, an execution error may appear. You should copy all ".dll" files to:
        ginkgocadx-2.5.1.0\src\ginkgocadx\Release

Linux:

    You could use deploy.sh script on src/
    For more information, read its contents.
Go to the ginkgo-cadx path and type:
    mkdir build
    cd build

```
cmake .. -DCUSTOM_PACKAGE:BOOL=TRUE -DUSE_PATCHED_LIBS:BOOL=TRUE
-DUSE_CUSTOM_WX:BOOL=TRUE -DUSE_CUSTOM_VTK:BOOL=TRUE
-DUSE_CUSTOM_ITK:BOOL=TRUE -DUSE_CUSTOM_DCMTK:BOOL=TRUE
-DUSE_SYSTEM_SQLITE:BOOL=FALSE -DCMAKE_INSTALL_PREFIX=./deploy
make BUILD_ALL
```

Mac OS X:

You could use deploy.sh script on src/
For more information, read its contents.