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

wxWidgets:

Execute wxMSW-2.8.11.exe and install
Apply wxWidgets-2.8.11.diff patches.
Open solution with Visual Studio 2008 C++ (wxWidgets\build\mws\wx.sln).
Build DLL UNICODE library target.
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
Copy includes and libs to ginkgo dll tree

VTK:

Expand VTK-x.y.z.zip
Build from Cmake:
  Mark: Show advanced values
  Set the variables:
    VTK_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: Show advanced values
    Set the variables:
        BUILD_SHARED_LIBS = ON
        ITK_USE_OPTIMIZED_REGISTRATION_METHODS = ON
        ITK_USE_PATENTED = ON
        ITK_USE_REVIEW = ON
        ITK_USE_REVIEW_STATISTICS = 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 and 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
GTK-2.0-dev
libx11-dev
libxt-dev
libxml2-dev
libssl-dev
libwrap0-dev

GCC 4.2

wxWidgets:

Expand and apply apply wxWidgets-2.8.11.diff patches.

Debug:

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

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.

VTK:

Expand VTK source archive.

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
```

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.

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.

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:

```
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
```

```
# We need to rebuild dcmdata without diccionary.
cd dcmdata/libsrc
make builtindict
make
cp libdcmdata.a ../../../../../trunk/dll/DCMTK-101021/Linux-
/lib/release/libdcmdata.a
```

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".

Linux:

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

Mac OS X:

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