

4/28/2015

MITK 2015.05

Presenting the new Release

Stefan Kislinskiy

dkfz.

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION



50 Years – Research for
A Life Without Cancer

Topics

- External libraries
- Build system
- Minimal compiler versions & C++ 11
- Qt 5
- BlueBerry

External library updates



- ITK 4.7.1



- VTK 6.2



- OpenCV 2.4.11



- Poco 1.6.0



- Recent CTK commit

New external libraries



- Redland Resource Description Framework Libraries
 - Raptor RDF Syntax Library
 - Rasqal RDF Query Library
 - Redland RDF Library



- VIGRA – Vision with Generic Algorithms
 - Machine Learning, Random Forests, ...



- HDF5 Library – Hierarchical Data Format
 - Organize and store large amounts of data



Build system – CMake 3.2

- CMake default export macro style
 <ModuleName>__EXPORT
 <MODULENAME>_EXPORT
- Support of CMake usage requirements:

```
mitk_create_module([moduleName]
  [INCLUDE_DIRS
    [PUBLIC|PRIVATE|INTERFACE] dir1 ...]
  [DEPENDS
    [PUBLIC|PRIVATE|INTERFACE] module1 ...]
  [PACKAGE_DEPENDS
    [PUBLIC|PRIVATE|INTERFACE] package1[|comp1[+comp2]...] ...]
)
```




Build system – Usage requirements example

```
mitk_create_module(  
  
  INCLUDE_DIRS  
    PUBLIC  ${MITK_BINARY_DIR}  
    PRIVATE src/Algorithms src/Controllers src/Interactions  
  
  DEPENDS  
    PUBLIC  mbilog CppMicroServices  
  
  PACKAGE_DEPENDS  
    PUBLIC  ITK|ITKTransform+ITKImageGrid+ITKImageFeature  
           VTK|vtkFiltersTexture+vtkFiltersParallel  
    PRIVATE ITK tinyxml OpenGL  
  
)
```


Build system – New source tree layout

- Unification
 - *Core, CppMicroServices* → Modules directory
 - *BlueBerry* plugins → Plugins directory
- De facto standard C/C++ source code organization:



```
MyModule
├── doc      <-- documentation (dox, png, md, ...)
├── include <-- public header files (h, txx)
├── resource <-- Qt and CppMicroServices resources
├── src     <-- sources and private headers (cpp, h, qrc, ui, ...)
└── test   <-- unit tests
```

Build system – New build tree layout



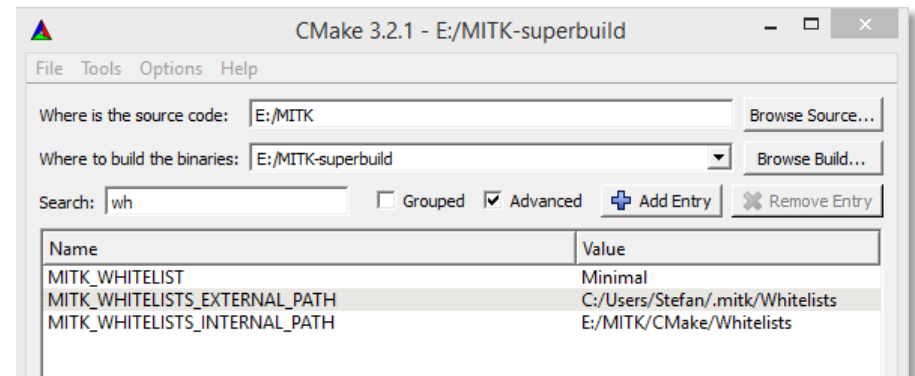
```
MITK-superbuild
├── ep          <-- external project install prefix
│   ├── bin
│   └── include
│   ...
├── MITK-build <-- actual MITK build tree
└── MITK-Data  <-- MITK test data
```


Build system – Whitelists

- Only build MITK components you really need
- Speed up build time
- No need to touch source tree



- MITK_WHITELIST
MITK_WHITELISTS_EXTERNAL_PATH
MITK_WHITELISTS_INTERNAL_PATH



Build system – Minimal whitelist example



- MITK/CMake/Whitelists/Minimal.cmake

```
set(enabled_modules
  # List of module directory names
  CppMicroServices
  Core
)

set(enabled_plugins
  # List of plugin directory names
)
```

Build system – Whitelists vs. build configurations

- Whitelists
 - MITK-build level
 - „*Everything not on a whitelist is completely invisible to the build system.*“
- Build configurations
 - MITK-superbuild level
 - Cover external libraries
 - Switch plugins on/off
 - May contain CMake script

Minimal compiler versions



- MSVC 2012 Update 4



- AppleClang 5.0



- Clang 3.4
- GNU 4.7.3

Minimal compiler versions



- MSVC 2012 Update 4



- AppleClang 5.0



- Clang 3.4
- GNU 4.7.3

Ubuntu 12.04

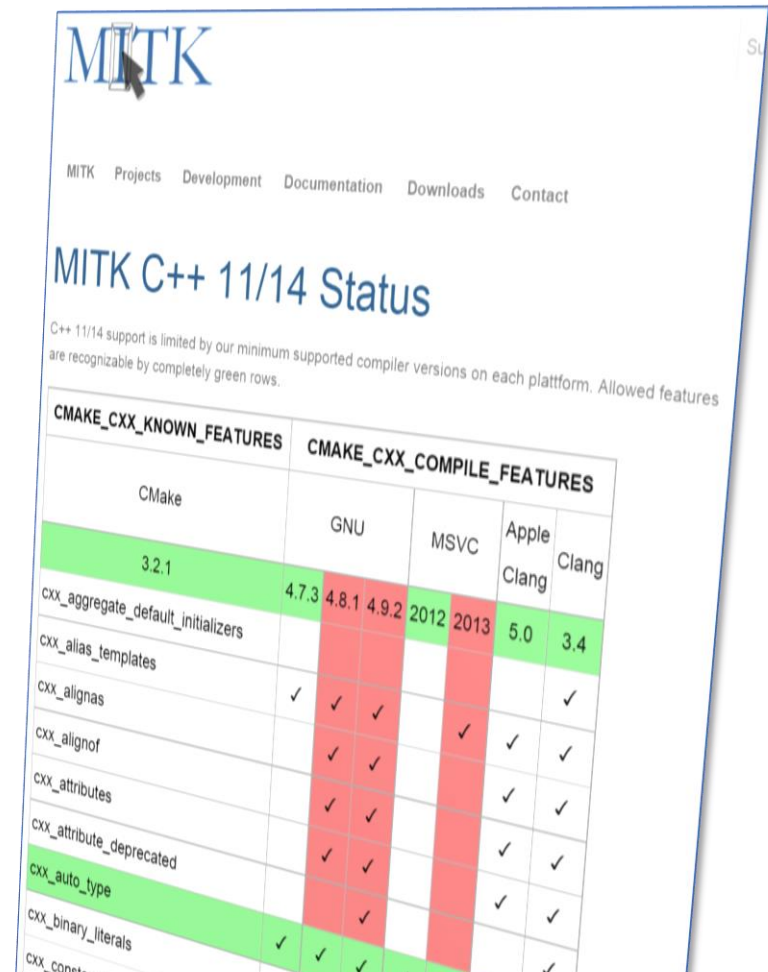


```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
sudo apt-get update
sudo apt-get install gcc-4.7 g++-4.7
```

```
CMAKE_C_COMPILER:FILEPATH=/usr/bin/gcc-4.7
CMAKE_CXX_COMPILER:FILEPATH=/usr/bin/g++-4.7
```

Minimal compiler versions – C++ 11

- auto
- nullptr
- override
- final
- Range-based for loops
- Lambdas
- Strong enums
- And more ...



MITK C++ 11/14 Status

C++ 11/14 support is limited by our minimum supported compiler versions on each platform. Allowed features are recognizable by completely green rows.

CMAKE_CXX_KNOWN_FEATURES	CMAKE_CXX_COMPILE_FEATURES							
	CMake	GNU			MSVC	Apple Clang	Clang	
3.2.1		4.7.3	4.8.1	4.9.2	2012	2013	5.0	3.4
cxx_aggregate_default_initializers								
cxx_alias_templates								
cxx_alignas	✓	✓	✓		✓		✓	✓
cxx_alignof		✓	✓				✓	✓
cxx_attributes		✓	✓				✓	✓
cxx_attribute_deprecated		✓	✓				✓	✓
cxx_auto_type			✓				✓	✓
cxx_binary_literals	✓	✓	✓					✓
cxx_constexpr		✓	✓					✓



Qt 5



- Qt 5 is default
- Qt 4 is deprecated (we mean it)



Qt 5 – Migration from Qt 4 (1/3)

- Mostly straight forward
 - `QString::toAscii()` → `QString::toLatin1()`
 - `#include <Q[Class]>`
 - Qt documentation & StackOverflow are your friends





Qt 5 – Migration from Qt 4 (2/3)

- Example module CMakeLists.txt:

```
mitk_create_module(MitkQtWidgets
  PACKAGE_DEPENDS
    PUBLIC Qt4|QtGui+QtOpenGL Qt5|Widgets+OpenGL
)
```



Qt 5 – Migration from Qt 4 (3/3)

- Plugin activator example:

+

```
class org_mitk_your_Activator : // ...  
{  
    Q_OBJECT  
#if QT_VERSION >= QT_VERSION_CHECK(5, 0, 0)  
    Q_PLUGIN_METADATA(IID "org_mitk_your_plugin")  
#endif  
    // ...
```

.h

-

```
#if QT_VERSION < QT_VERSION_CHECK(5, 0, 0)  
    #include <QtPlugin>  
    Q_EXPORT_PLUGIN2(org_mitk_your_plugin, mitk::org_mitk_your_Activator)  
#endif
```

.cpp



BlueBerry (1/4)

- Custom Commands and Handlers

Add UI contributions to menus, toolbars, status bars, and context menus



BlueBerry (2/4)

- New Perspectives preference page

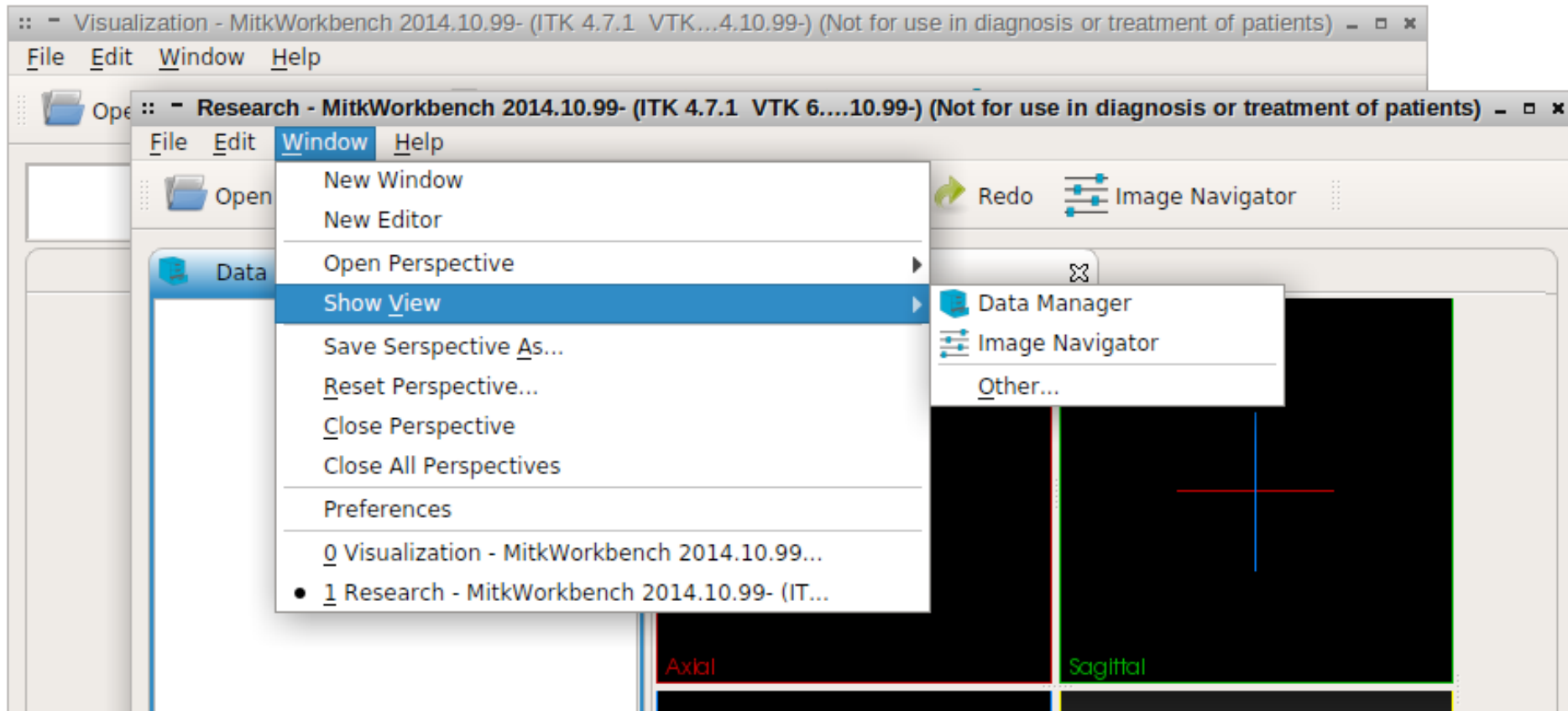
The screenshot shows the 'Preferences' dialog box with the 'Perspectives' page selected. The left sidebar contains a tree view with 'Perspectives' highlighted. The main area is titled 'Perspectives' and contains the following elements:

- Open a new perspective:** A section with two radio buttons: 'In the same window' (selected) and 'In a new window'.
- Available perspectives:** A list of perspectives with icons and labels:
 - Dicom Perspective (with a red 'x' icon)
 - P1 Extended Perspective
 - P1 Extended Perspective
 - Help (with a question mark icon)
 - Minimal Example Perspective (with a red 'x' icon)
 - Minimal Example Perspective (with a red 'x' icon)
 - P1 Minimal Perspective 1
 - P2 Minimal Perspective 2
 - Research (default) (with a magnifying glass icon)
 - Viewer (with a red 'x' icon)
 - Viewer Perspective (with a red 'x' icon)
 - Visualization (with a globe icon)
- Buttons:** 'Make Default', 'Revert', and 'Delete' buttons are located to the right of the list.
- Note:** A note at the bottom states: "Note: 'Revert' removes the customization from the selected perspective. This only applies to newly opened perspectives."
- Footer Buttons:** 'OK', 'Import...', 'Export...', and 'Cancel' buttons are located at the bottom of the dialog.



BlueBerry (3/4)

- New Window menu





BlueBerry (4/4)

- Improved launching of MITK Workbench based applications

```
int main(int argc, char* argv[])  
{  
    mitk::BaseApplication app(argc, argv);  
  
    app.setSingleMode(true);  
    app.setApplicationName("MITK Workbench");  
    app.setOrganizationName("DKFZ");  
  
    app.setProperty("blueberry.application", "org.mitk.qt.extapplication");  
  
    return app.run();  
}
```

.cpp

Thank you for
your attention!

Further
information
on www.dkfz.de

dkfz.

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

50 Years – Research for
A Life Without Cancer