KDAB
www.kdab.com

Modern CMake with Qt and Boost

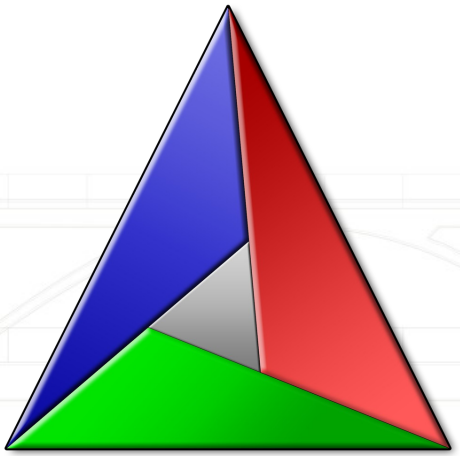
Stephen Kelly
stephen.kelly@kdab.com
KDAB

Stephen Kelly

KDAB
www.kdab.com

- C++/Qt user since 2006
- KDE contributor since 2007
- KDAB Engineer since 2009
- Qt contributor since 2009
- CMake contributor since 2011
- Boost contributor since 2013
- Living in Berlin


Qt



KDAB, Qt & KDE

KDAB
www.kdab.com

- Modelview maintainer
- Metatype system
- CMake files maintainer
- Consulting/training
- KDE libraries

CMake

KDAB
www.kdab.com

- Top contributor to CMake 2.8.11, 2.8.12
- Transformed how CMake works
- Designed for Qt, KDE and Boost

stephen@hal:~/dev/src/boost-trunk{mystuff}

$ git diff –shortstat ea38de1f3b8b..HEAD

2029 files changed, 2482 insertions(+), 51267 deletions(-)

KDAB
www.kdab.com

Why?

Why?

KDAB
www.kdab.com

- Dependencies
- Versioning
- Distributable

Why?

KDAB
www.kdab.com

- Dependencies
- Versioning
- Distributable
- Portability
- Compiler-specific flags
- Dependency-specific flags

KDAB
www.kdab.com

Why CMake?

Why CMake?

KDAB
www.kdab.com

- Find dependencies
- Cross-platform
- Multiple generators

Why CMake?

KDAB
www.kdab.com

- Find dependencies
- Cross-platform
- Multiple generators
- Buildsystem abstractions
- Dependency abstractions
- Compiler feature abstractions

Dependencies

KDAB
www.kdab.com

```
find_package(Qt5Widgets 5.2 REQUIRED)
add_executable(myapp main.cpp)
target_link_libraries(myapp
    Qt5::Widgets
)
```

Dependencies

KDAB
www.kdab.com

```
# find_package(Qt5Widgets 5.2 REQUIRED)
add_executable(myapp main.cpp)
target_link_libraries(myapp
    Qt5::Widgets # Error diagnostic
)
```

```
add_library(Qt5::Widgets
                        SHARED IMPORTED)
set_property(TARGET Qt5::Widgets
     LOCATION
     "${somewhere}/lib/libQt5Widgets.so"
)
```


```
...
set_property(TARGET Qt5::Widgets
     INTERFACE_INCLUDE_DIRECTORIES
     "${somewhere}/include/QtWidgets"
)
```


```
...
set_property(TARGET Qt5::Widgets
    INTERFACE_COMPILE_DEFINITIONS
    "QT_WIDGETS_LIB"
)
```

```
...
set_property(TARGET Qt5::Core
     INTERFACE_COMPILE_DEFINITIONS
     "$<$<CONFIG:Debug>:QT_DEBUG>"
)
```

```
#if !defined(__PIC__)
#  error "Compile your code "
         "with -fPIC or -fPIE."
#endif
```

Qt5WidgetsConfig

KDAB
www.kdab.com

```
set_property(TARGET Qt5::Core
     INTERFACE_COMPILE_OPTIONS
     "$<AND:\
       $<STREQUAL:\
          $<TARGET_PROPERTY:TYPE>,\
          EXECUTABLE\
       >,$<OR:\
          $<COMPILER_ID:GNU>,\
          $<COMPILER_ID:Clang>\
     >:-fPIE>"
)
```


```
...
set_property(TARGET Qt5::Core
     INTERFACE_POSITION_INDEPENDENT_CODE
     "ON"
)
```

Qt specific features

KDAB
www.kdab.com

- Automatic Qt code generation features
  - Automatic moc invocation
  - Automatic uic invocation
  - Automatic rcc invocation

```
set(CMAKE_AUTOMOC ON)

class MyClass : public QObject
{
    Q_OBJECT
};
```

```
set(CMAKE_AUTORCC ON)

find_package(Qt5Widgets REQUIRED)
add_executable(myapp
    main.cpp
    myresources.qrc
)
```

```
set(CMAKE_AUTOUIC ON)

#include "ui_mywidget.h"

MyWidget::MyWidget(QWidget *parent)
    : QWidget(parent),
      ui(new Ui::MyWidget)
{
}
```

Header-only libraries

KDAB
www.kdab.com

```
add_library(boost::any
                    IMPORTED INTERFACE)
# No LOCATION!
set_property(TARGET boost::any
     INTERFACE_INCLUDE_DIRECTORIES
     "${someprefix}/include"
)
```

Transitive usage requirements

Boost dependency graph

Transitive usage requirements

KDAB
www.kdab.com

```
add_library(kdab::sqlate
                    IMPORTED INTERFACE)
set_property(TARGET kdab::sqlate
     INTERFACE_COMPILE_DEFINITIONS
     "BOOST_MPL_LIMIT_VECTOR_SIZE=50"
)
```

Transitive usage requirements

KDAB
www.kdab.com

```
add_library(other::lib
                    IMPORTED INTERFACE)
set_property(TARGET other::lib
     INTERFACE_COMPILE_DEFINITIONS
     "BOOST_MPL_LIMIT_VECTOR_SIZE=30"
)
```

Transitive usage requirements


```
add_executable(user main.cpp)
target_link_libraries(user
    kdab::sqlate
    other::lib
)

-DBOOST_MPL_LIMIT_VECTOR_SIZE=50
```

Transitive usage requirements


```
set_property(TARGET kdab::sqlate
     INTERFACE_COMPILE_DEFINITIONS
     "QT_DISABLE_DEPRECATED_BEFORE=0x050100"
)
```

Transitive usage requirements

KDAB
www.kdab.com

```
set_property(TARGET other::lib
    INTERFACE_COMPILE_DEFINITIONS
    "QT_DISABLE_DEPRECATED_BEFORE=0x050200"
)
```

Transitive usage requirements

KDAB
www.kdab.com

```
add_executable(user main.cpp)
target_link_libraries(user
    kdab::sqlate
    other::lib
)

-DQT_DISABLE_DEPRECATED_BEFORE=0x050100
```

Transitive usage requirements

KDAB
www.kdab.com

```
add_executable(user main.cpp)
target_link_libraries(user
    Qt4::QtCore
    Qt5::Core # Diagnostic
)
```

Transitive usage requirements


```
set_property(TARGET Qt5::Core
     INTERFACE_QT_MAJOR_VERSION
     "5"
)

set_property(TARGET Qt4::QtCore
     INTERFACE_QT_MAJOR_VERSION
     "4"
)
```

KDAB
www.kdab.com

C++11

KDAB
www.kdab.com

C++14

KDAB
www.kdab.com

C++98

C89

C11

C99

C1Y

Transitive compiler features

KDAB
www.kdab.com

```
add_executable(new-app main.cpp)
target_compile_features(new-app
    cxx_member_templates
    cxx_constexpr
    cxx_generic_lambda
    c_restrict
    gnuxx_typeof
)
```

Optional compiler features

KDAB
www.kdab.com

```
write_compiler_feature_header(
    FILE "mycompiler_detection.h"
    PREFIX MyPrefix
    FEATURES
      cxx_static_assert
      cxx_final
      cxx_variadic_templates
)
```

Optional compiler features

KDAB
www.kdab.com

```
#if defined(__clang__) \
  && __has_feature(cxx_static_assert)
#define MyPrefix_STATIC_ASSERT(a) \
                        static_assert(a, #a)
#else
...
#endif
```

Optional compiler features

KDAB
www.kdab.com

```
#if defined(__clang__) \
  && __has_feature(cxx_static_assert)
#define MyPrefix_STATIC_ASSERT(a) \
                        static_assert(a, #a)
#else
template<bool test>
struct MyPrefixStaticAssert;
template<>
struct MyPrefixStaticAssert<true> {};
#define MyPrefix_STATIC_ASSERT(a) \
    MyPrefixStaticAssert<a>;
#endif
```

Optional compiler features

KDAB
www.kdab.com

```
#if defined(__clang__) \
  && __has_feature(cxx_variadic_templates)
  || defined(__GNUC__) && (VER >= 4.5)
#define MyPrefix_VARIADIC_TEMPLATES 1
#else
#define MyPrefix_VARIADIC_TEMPLATES 0
#endif
```

Optional compiler features

KDAB
www.kdab.com

```
#if ...
#define MyPrefix_FINAL final
#elif defined(_MSC_VER) && ...
#define MyPrefix_FINAL sealed
#else
#define MyPrefix_FINAL
#endif
```

Summary

KDAB
www.kdab.com

- CMake designed for Qt and Boost
  - Diagnostics
  - Built-in code-generator support
  - Usage requirements
    - Transitive
    - Conditionals
  - Header-only library abstraction
  - Compiler feature support


any questions?
FUN GOPHER