

04/09/2012

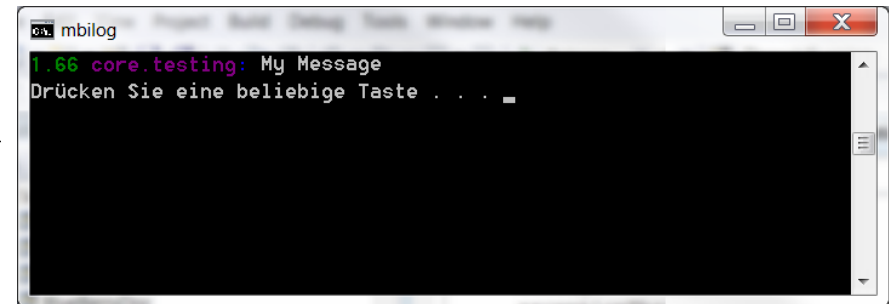
Logging in MITK

Alfred Franz

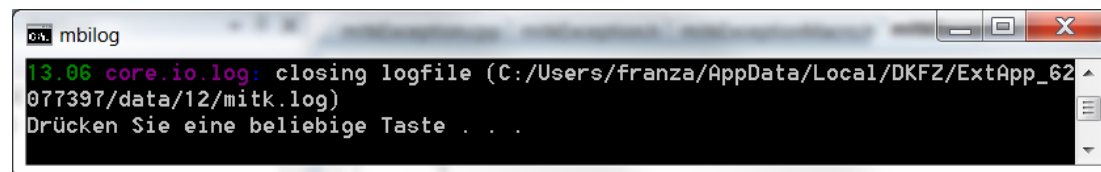
- Logging in MITK:
 - MITK_INFO << „My logging message“;
 - std::cout should not be used!

```
MITK_INFO << "My Message";
```

- By default, all logging messages
 - are written to the console
 - and to a text file

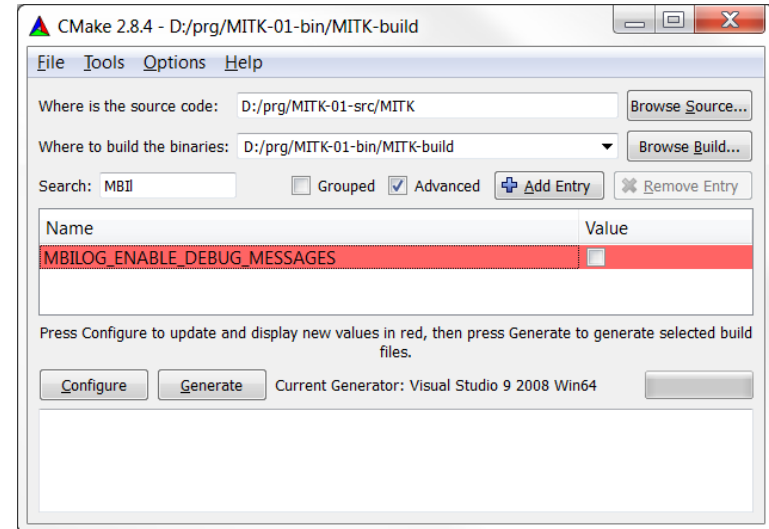


*name & folder of logging file
is given when closing MITK:*



- Logging Levels
- Logging Object Information
- Categorized Logging Messages
- Conditional Logging
- MITK Logging Plug-In

- **Info (MITK_INFO):** information messages
- **Debug (MITK_DEBUG):** designed for debug output, only displayed if the CMake-Variable `MBILOG_ENABLE_DEBUG_MESSAGES` is on
- **Warning (MITK_WARN):** messages about unexpected or potentially problematic states in the program flow
- **Error (MITK_ERROR):** notify the user about corrupt states in the program flow
- **Fatal (MITK_FATAL):** report corrupt states that lead directly to a crash of the program



```
MITK_INFO << "My Message";  
  
MITK_DEBUG << "My Debug Message";  
  
MITK_WARN << "Warning Message";  
  
MITK_ERROR << "Error Message";  
  
MITK_FATAL << "Fatal Message";
```

- As with `std::cout`, you can also log object information
- Example:

```
MITK_INFO << "I want to log my vector size: " << m_vector.getSize();
```

- To categorize logging messages, they can be assigned to specific topics.
- Therefore classes can be added by using brackets.

```
MITK_INFO << "no class";  
MITK_INFO("MyClass") << "single class";
```

- Subclasses are also possible.

```
MITK_INFO("MyClass")("MySubClass") << "class with subclass";
```

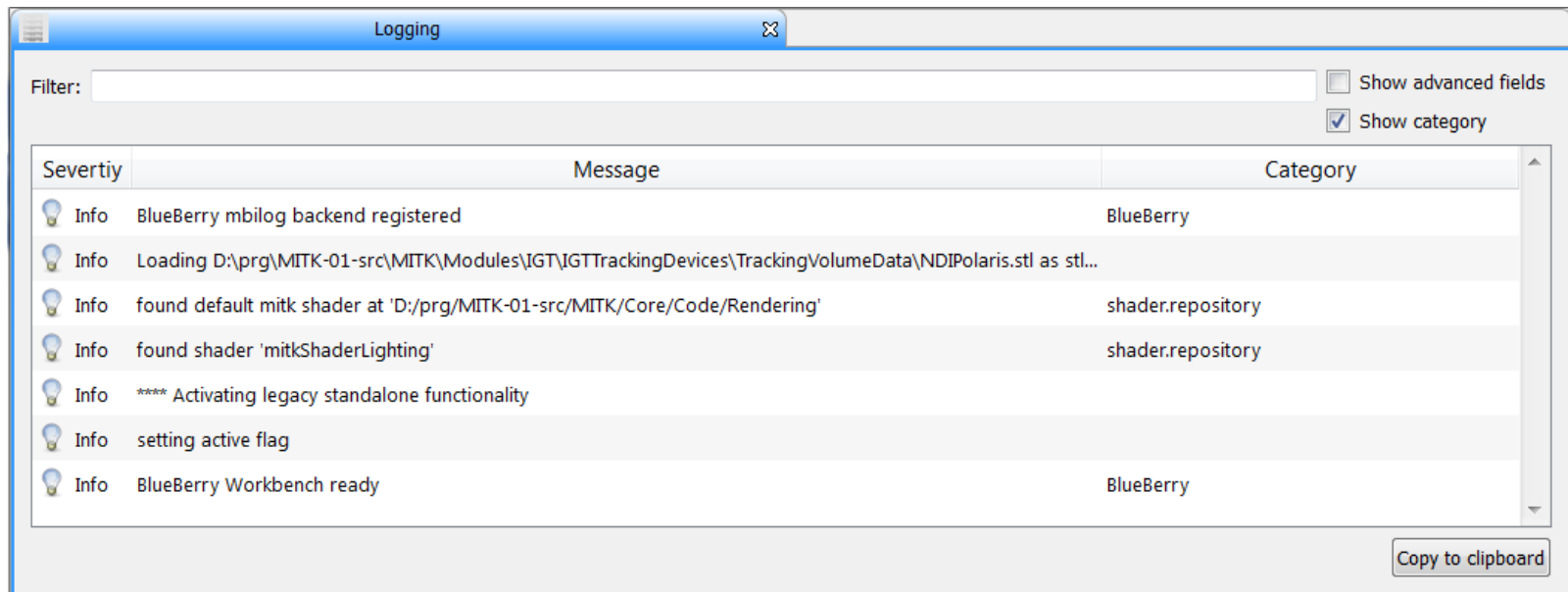
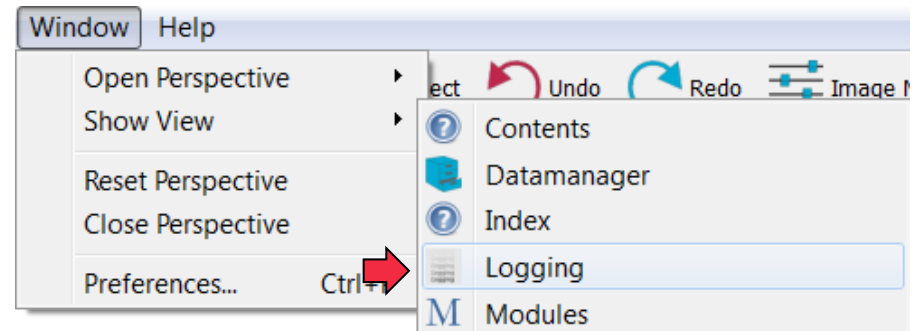
- It's easy to filter the messages by specific classes, when using the logging plug-in.

- Conditions for logging are also possible.
- Therefore boolean conditions can be given in brackets.
 - The logging message is only logged if the condition is fulfilled.

- Example:

```
MITK_INFO(x>1000) << "x is too large"; //logs only if x > 1000
```

- Enabled by default, only activate:
- Features:
 - Filter your message
 - Copy them to the clipboard
 - Advanced fields (class, line of code, etc.) available




```
MITK_DEBUG << "Hello World";
```

was added to the code, but even in the debug mode no „Hello World“ appears. What went wrong?

1. Logging was not started using the command `StartLogging()`.
2. The program was started without debugging.
3. The cmake variable for debug logging is off.
4. There was an error during the superbuild of MITK.