Logging in MITK

Alfred Franz





9/4/2012 | Page 2 Logging: Basics



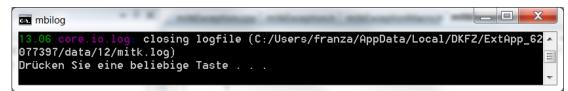
- Logging in MITK:
 - MITK_INFO << "My logging message";
 - std::cout should not be used!

MITK_INFO << "My Message";</pre>

- 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:



```
Alfred Michael Franz
MBI
```

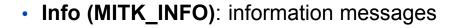
9/4/2012 |



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



9/4/2012 | Page 4 Logging Levels



- 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

ile <u>T</u> ools <u>O</u> ptions <u>F</u>	lelp			
Where is the source code:	D:/prg/MITK-01-src/MITK		Browse Source	
Where to build the binaries:	D:/prg/MITK-01-bin/MITK-build	g/MITK-01-bin/MITK-build		
Search: MBI	Grouped 📝 Advanced	4 Add Entry	Remove Entry	
Name		Val	Value	
MBILOG_ENABLE_DEBU	G_MESSAGES			
Press Configure to update a	nd display new values in red, then pres files.	s Generate to ge	enerate selected buil	
	files.	-	enerate selected buil	
Press Configure to update a	files.	-	enerate selected buil	
	files.	-	enerate selected buil	

<pre>MITK_INFO << "My Message";</pre>
<pre>MITK_DEBUG << "My Debug Message";</pre>
<pre>MITK_WARN << "Warning Message";</pre>
<pre>MITK_ERROR << "Error Message";</pre>
<pre>MITK_FATAL << "Fatal Message";</pre>





9/4/2012 | Page 5



- 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";</pre>

• Subclasses are also possible.

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

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



Page 7

9/4/2012 |



- Conditions for logging are also possible.
- Therefore boolean conditions can be given in brackets.
 - \rightarrow 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

9/4/2012 |

MITK Logging Plug-In

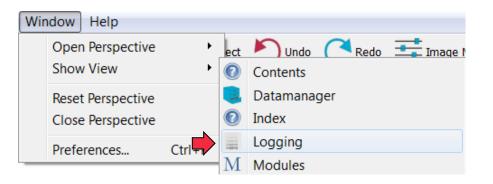


- Enabled by default, only activate:
- Features:

Page 8

- Filter your message
- Copy them to the clipboard
- Advanced fields (class, line of code, etc.) available

	Logging		
Filter:		Show advanced field	elds
Severtiy	Message	Category	-
💡 Info	BlueBerry mbilog backend registered	BlueBerry	
💡 Info	Loading D:\prg\MITK-01-src\MITK\Modules\IGT\IGTTrackingDevices\TrackingVolumeData\NDIPolaris.stl as stl		
💡 Info	found default mitk shader at 'D:/prg/MITK-01-src/MITK/Core/Code/Rendering'	shader.repository	
💡 Info	found shader 'mitkShaderLighting'	shader.repository	
💡 Info	**** Activating legacy standalone functionality		
💡 Info	setting active flag		
💡 Info	BlueBerry Workbench ready	BlueBerry	
		Copy to clipbo	ard







MITK_DEBUG << "Hello World";</pre>

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.