Robot Vision Systems

Lecture 6: Rapid prototyping in OpenCV using Python and Ceemple

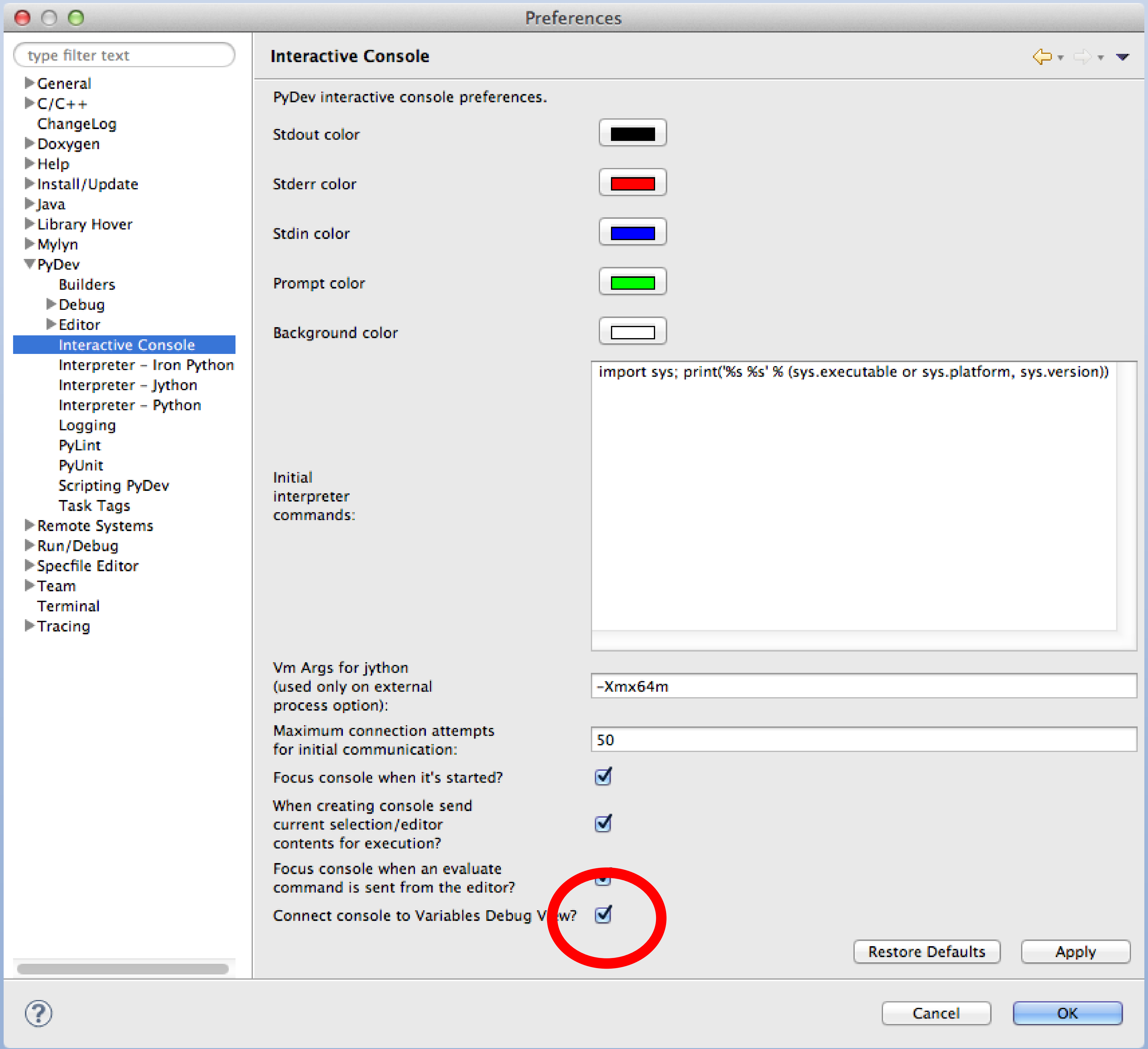
Michael Felsberg

michael.felsberg@liu.se

Python

- General purpose programming language
- Interpreted high-level language
- Readability: clear and expressive syntax
- Large standard library
- Multiple programming paradigms, a.o. OO
- Reference implementation CPython free and open source
- Version 3 can be used with OpenCV 3
- Integrated in Eclipse by means of **PyDev**

Interactive Workspace

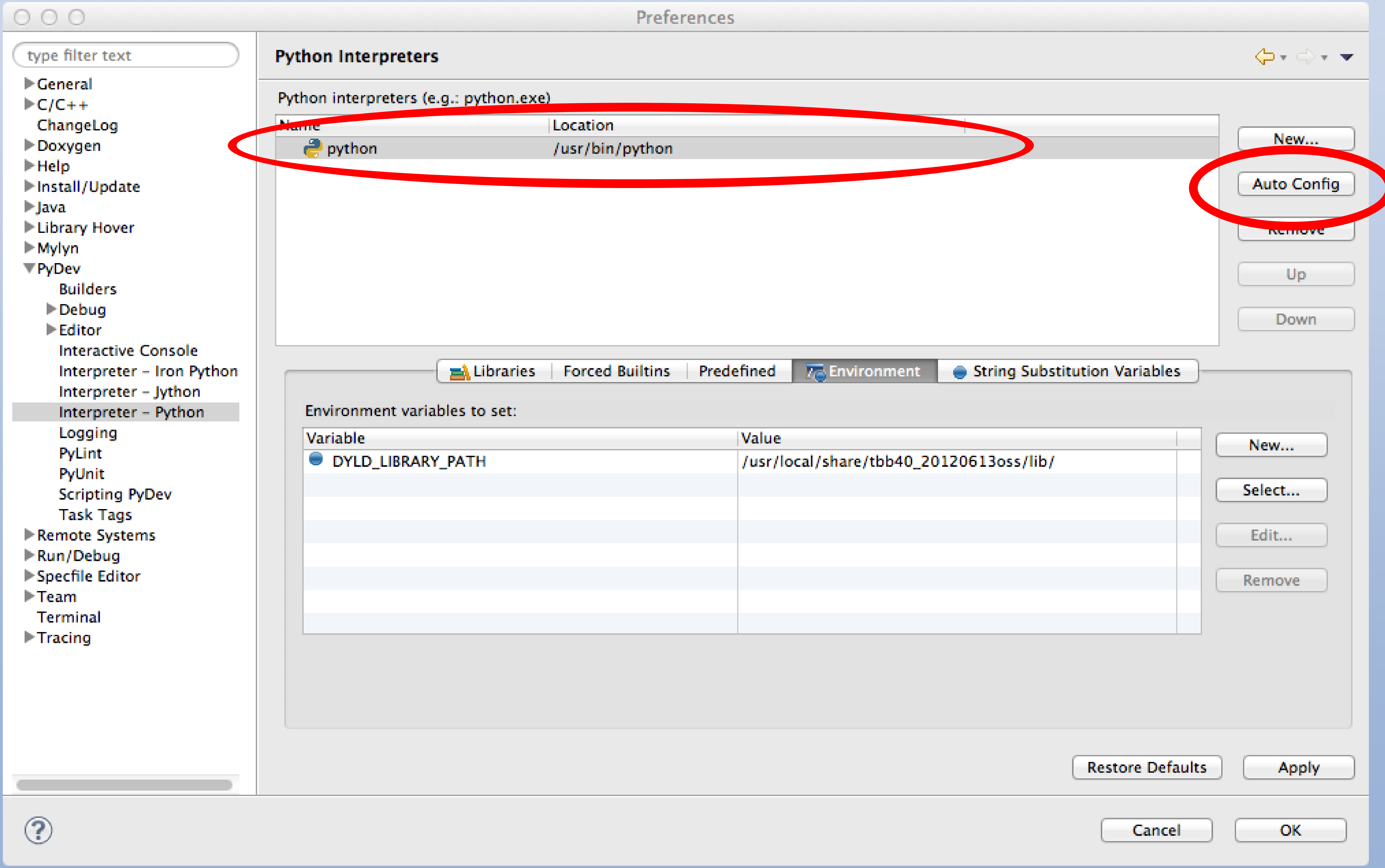


Interactive Console

PyDev interactive console preferences.

Stdout color	⬛
Stderr color	🟥
Stdin color	🟦
Prompt color	🟩
Background color	⬜
Initial interpreter commands:	`import sys; print('%s %s' % (sys.executable or sys.platform, sys.version))`
Vm Args for jython (used only on external process option):	-Xmx64m
Maximum connection attempts for initial communication:	50
Focus console when it's started?	✓
When creating console send current selection/editor contents for execution?	✓
Focus console when an evaluate command is sent from the editor?	✓
Connect console to Variables Debug View?	✓

Interactive Workspace



Preferences

type filter text

- General
- C/C++
- ChangeLog
- Doxygen
- Help
- Install/Update
- Java
- Library Hover
- Mylyn
- PyDev
  - Builders
  - Debug
  - Editor
  - Interactive Console
  - Interpreter – Iron Python
  - Interpreter – Jython
  - Interpreter – Python
  - Logging
  - PyLint
  - PyUnit
  - Scripting PyDev
  - Task Tags
- Remote Systems
- Run/Debug
- Specfile Editor
- Team
- Terminal
- Tracing

Python Interpreters

Python interpreters (e.g.: python.exe)

Name	Location
python	/usr/bin/python

New... Auto Config Remove Up Down

Libraries | Forced Builtins | Predefined | Environment | String Substitution Variables

Environment variables to set:

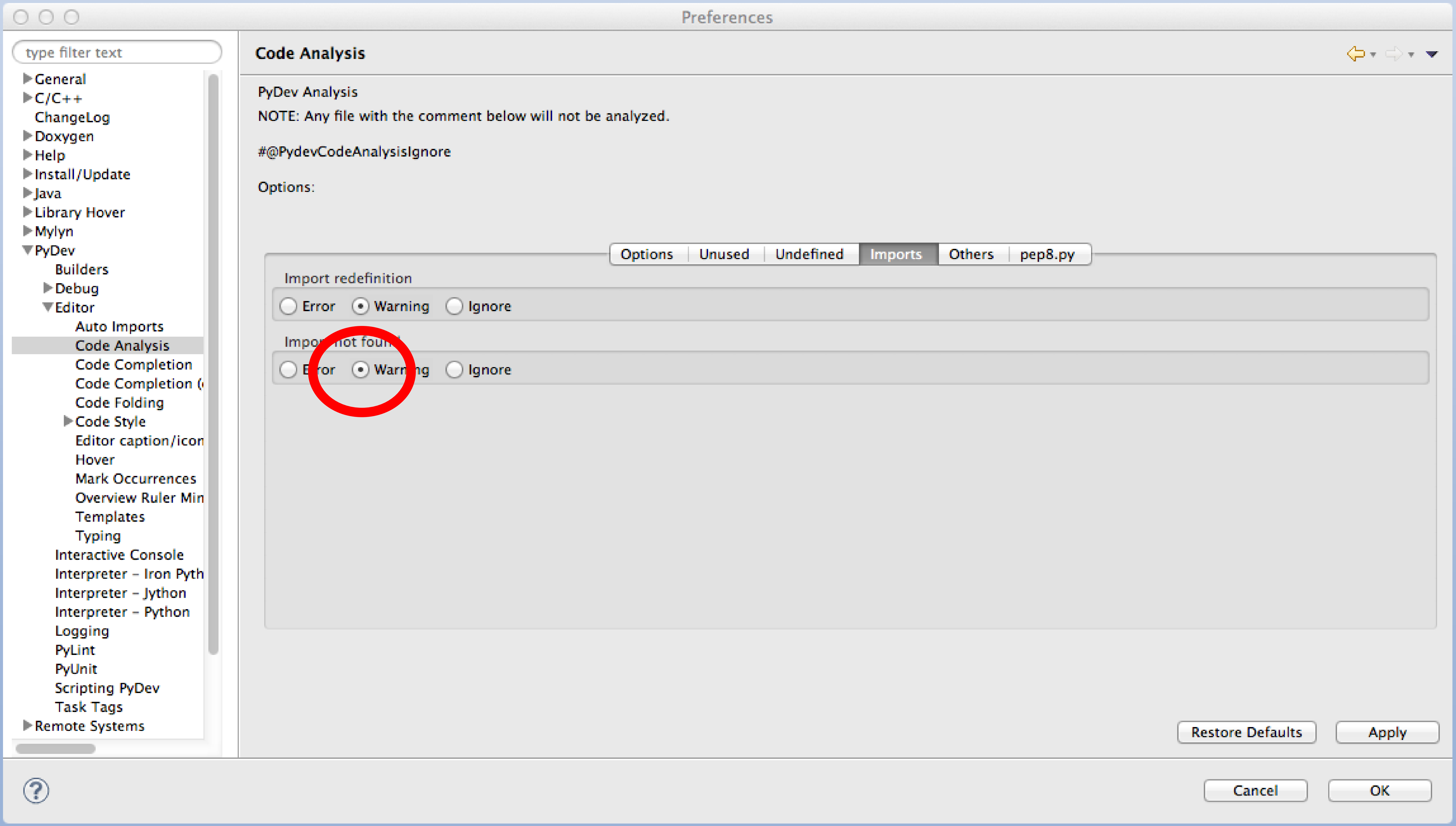
Variable	Value
DYLD_LIBRARY_PATH	/usr/local/share/tbb40_20120613oss/lib/

New... Select... Edit... Remove

Restore Defaults Apply

Cancel OK

Interactive Workspace



Preferences

type filter text

- General
- C/C++
- ChangeLog
- Doxygen
- Help
- Install/Update
- Java
- Library Hover
- Mylyn
- PyDev
  - Builders
  - Debug
  - Editor
    - Auto Imports
    - Code Analysis
    - Code Completion
    - Code Completion (
    - Code Folding
    - Code Style
    - Editor caption/icon
    - Hover
    - Mark Occurrences
    - Overview Ruler Min
    - Templates
    - Typing
  - Interactive Console
  - Interpreter - Iron Pyth
  - Interpreter - Jython
  - Interpreter - Python
  - Logging
  - PyLint
  - PyUnit
  - Scripting PyDev
  - Task Tags
- Remote Systems

Code Analysis

PyDev Analysis

NOTE: Any file with the comment below will not be analyzed.

#@PydevCodeAnalysisIgnore

Options:

Options | Unused | Undefined | Imports | Others | pep8.py

Import redefinition

Error [✓ Warning] Ignore

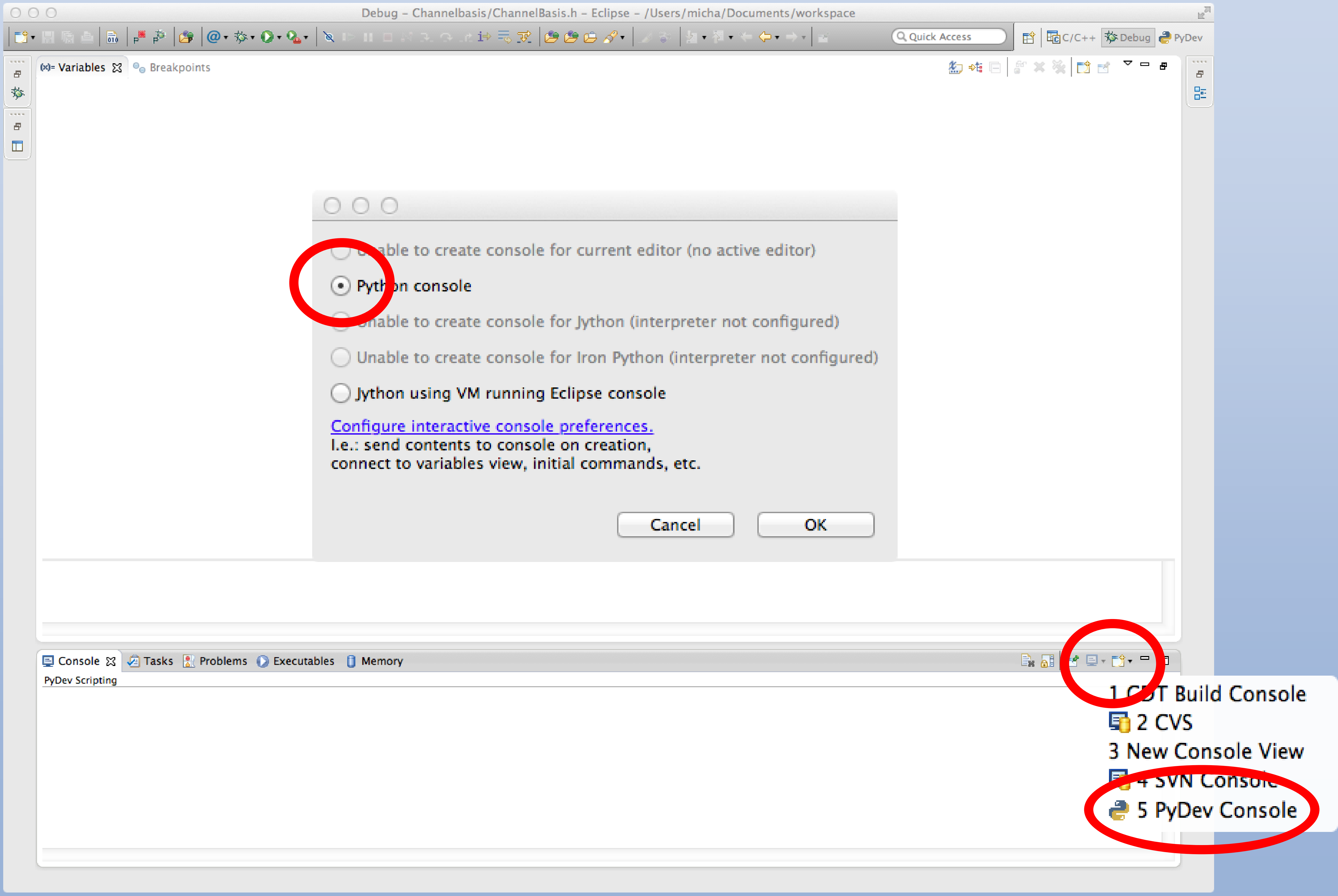
Import not found

Error [✓ Warning] Ignore

Restore Defaults   Apply

Cancel   OK

Open Console

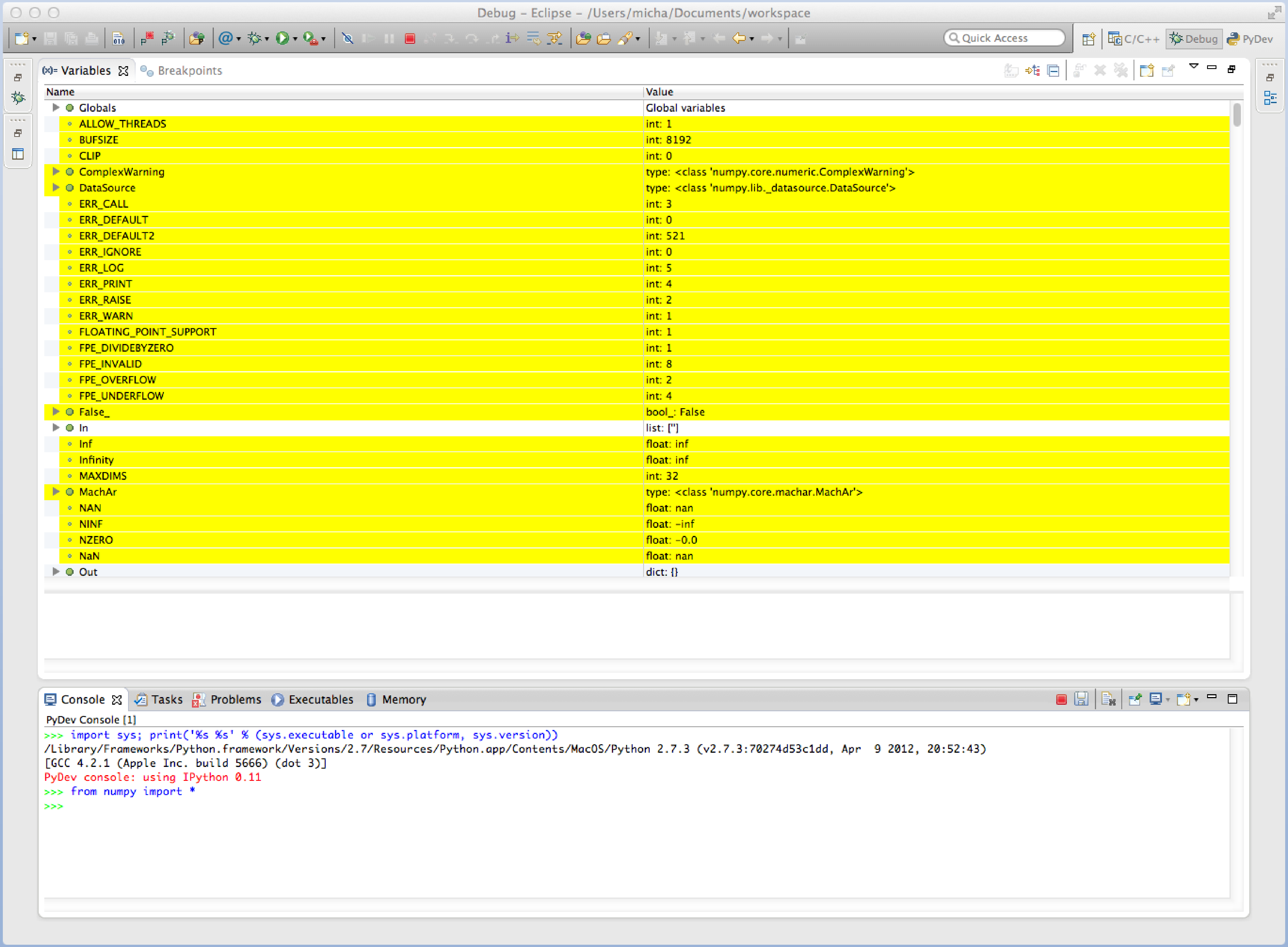


Debug – Channelbasis/ChannelBasis.h – Eclipse – /Users/micha/Documents/workspace

Unable to create console for current editor (no active editor)
[✓ Python console]
Unable to create console for Jython (interpreter not configured)
Unable to create console for Iron Python (interpreter not configured)
Jython using VM running Eclipse console

Configure interactive console preferences.
I.e.: send contents to console on creation,
connect to variables view, initial commands, etc.

Import NumPy



Name	Value
Globals	Global variables
ALLOW_THREADS	int: 1
BUFSIZE	int: 8192
CLIP	int: 0
ComplexWarning	type: <class 'numpy.core.numeric.ComplexWarning'>
DataSource	type: <class 'numpy.lib._datasource.DataSource'>
ERR_CALL	int: 3
ERR_DEFAULT	int: 0
ERR_DEFAULT2	int: 521
ERR_IGNORE	int: 0
ERR_LOG	int: 5
ERR_PRINT	int: 4
ERR_RAISE	int: 2
ERR_WARN	int: 1
FLOATING_POINT_SUPPORT	int: 1
FPE_DIVIDEBYZERO	int: 1
FPE_INVALID	int: 8
FPE_OVERFLOW	int: 2
FPE_UNDERFLOW	int: 4
False_	bool_: False
In	list: ['']
Inf	float: inf
Infinity	float: inf
MAXDIMS	int: 32
MachAr	type: <class 'numpy.core.machar.MachAr'>
NAN	float: nan
NINF	float: -inf
NZERO	float: -0.0
NaN	float: nan
Out	dict: {}

```
PyDev Console [1]
>>> import sys; print('%s %s' % (sys.executable or sys.platform, sys.version))
/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/Python 2.7.3 (v2.7.3:70274d53c1dd, Apr  9 2012, 20:52:43)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)]
PyDev console: using IPython 0.11
>>> from numpy import *
>>>
```

NumPy

- Python extension
- Multi-dimensional arrays
- High-level functions
- Similar to MATLAB, but more modern
- Also based on LAPACK
- Further extensions by means of SciPy and Matplotlib (native SVG support!)
- **OpenCV Mat are wrapped to NumPy arrays**

Example

- x = linspace(0,2*pi,100)
- y = sin(x)
- Better use ‘import numpy’ and explicitly writing numpy.sin(x) etc
- Result can be plotted:
  - –from matplotlib import pyplot
  - –pyplot.plot(x, y)
  - –pyplot.show()

Using OpenCV in Python

- OpenCV functions are in Python module cv2
  - import cv2
- OpenCV1 is no longer supported
- Use autoexpand in Eclipse and search in documentation to find function names
- Problem with Ceemple: missing Python bindings
  - Install Python: WinPython (Windows) or via apt-get
  - Download OpenCV 3 (binary (Windows) or build (*), see http://milq.github.io/install-opencv-ubuntu-debian/)
  - Copy cv2.pyd to Lib\site-packages (Windows) or cv2.so to /usr/local/lib/python2.7/site-packages
    (*) experimental: copy first /opt/ceemple/lib/* to build/lib/ before running “make opencv_python2”

Example: Read form Cam

- –capture = cv2.VideoCapture(0)
- –[status,img] = capture.retrieve()
- –cv2.imshow("camera",img)
- –cv2.waitKey(0)
- –cv2.destroyAllWindows()

- Note that *status* contains binary flag
- Without waitKey(0), window will not be created (0 means: infinitely long)

Example: Color Edges

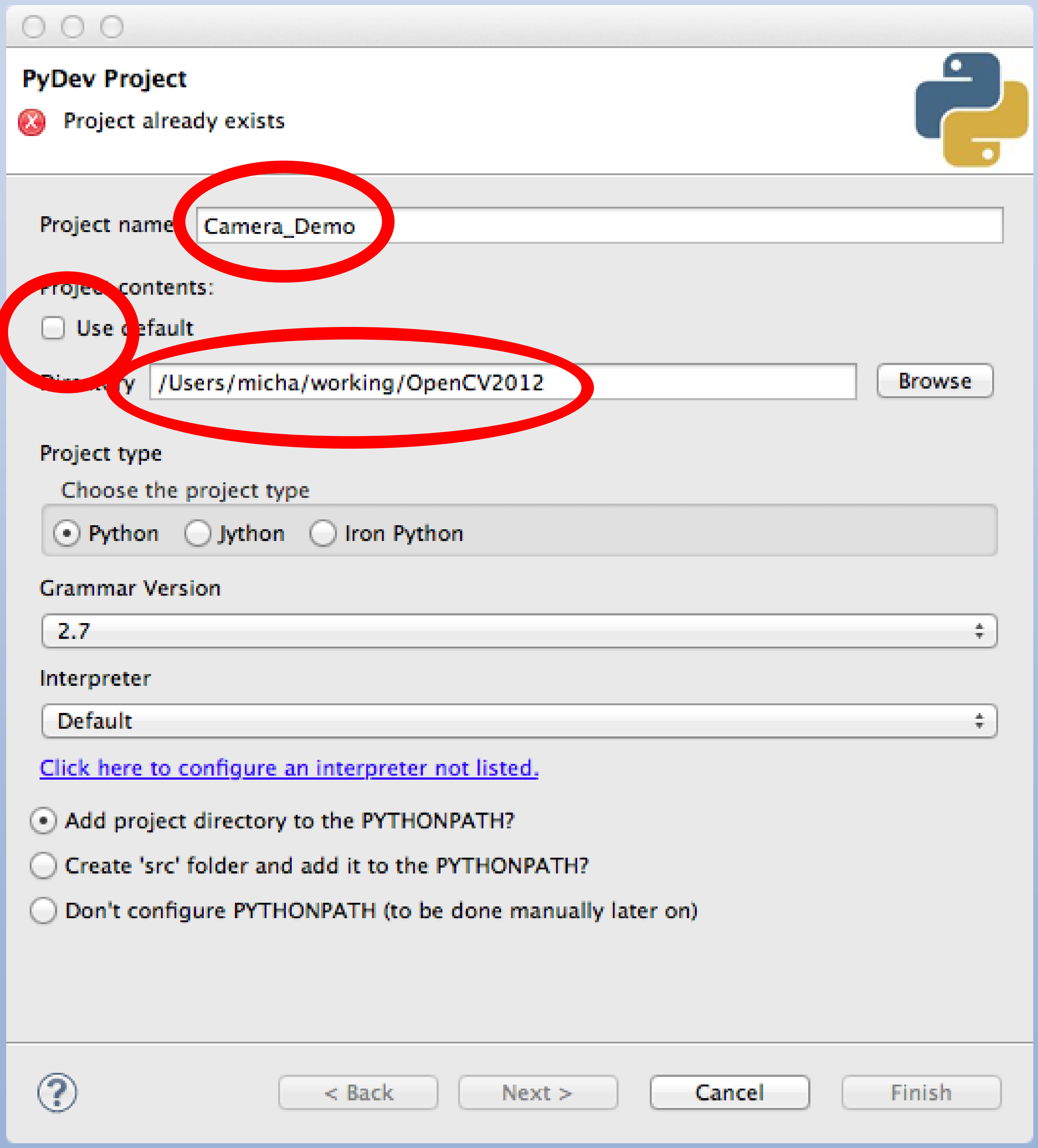
- –h2 = numpy.array([[-1.0, 0 , 1]]).T.dot(
          numpy.array([[1, 2, 1]]))
- –edgex = cv2.filter2D(img, cv2.CV_32F, h2.T)
- –edgey = cv2.filter2D(img, cv2.CV_32F, h2)
- –mag = cv2.magnitude(edgex,edgey)
- –cv2.imshow("camera",
   cv2.convertScaleAbs((255.0/mag.max())*mag))
- –cv2.waitKey(0)
- –cv2.destroyAllWindows()

• Note that *magnitude* only works with floats

Generating Scripts

- As in MATLAB: just pipe your command line commands into a text-file
- Suffix: .py
- You may run the script from command-line by python my_script.py

Generating Projects



PyDev Project

Project already exists

Project name: Camera_Demo

Project contents:

[ ] Use default

Directory: /Users/micha/working/OpenCV2012

Project type

Choose the project type

[✓ Python] Jython Iron Python

Grammar Version: 2.7

Interpreter: Default

Click here to configure an interpreter not listed.

[✓ Add project directory to the PYTHONPATH?]

Create 'src' folder and add it to the PYTHONPATH?

Don't configure PYTHONPATH (to be done manually later on)

Package and Modules

- A PyDev project is just a container for packages
- Packages correspond (in a certain way) to C++ namespaces and are containers for modules
  - –Next step: generate package
- Modules correspond to .cpp files and are containers for functions and scripts
  - –Next step: generate module (‘main’) and add code

Prototyping in Ceemple

- No interactive console (drawback or advantage?)
- Only on Windows (drawback)
- Same syntax (advantage)
- Matplotlib is not available (drawback)
- Weak support for debugging of Mat (drawback)
- Not all math available (drawback)
- Faster execution (advantage)
- No extra testing needed (advantage)