



EASE-ily Make the Most of Eclipse with Python

NOBUGS DAWN SWG WORKSHOP

20th October 2016

Who We Are



Tracy Miranda

Eclipse evangelist, Science steering committee member

Jonah Graham

January project lead, CDT committer

Christian Pontesegger

EASE project lead, Code & Me blogger

EASE-ily Making the Most of Eclipse with Python

SCRIPTING ECLIPSE WITH PYTHON

AGENDA

- ❑ Hello World
- ❑ Python & EASE Background
- ❑ Getting started
 - EASE Modules
 - Executing EASE Scripts
 - Magic Keyword Headers
- ❑ Python & EASE: The Vision & The Roadmap
- ❑ Hands-on-Scripting
 - Automate, Extend, Integrate & Share with EASE

EASE-ily Making the Most of Eclipse with Python

HELLO WORLD



Setup for Exercises

Setup Instructions :

Open up EASE.htm in Eclipse or in a browser

<https://github.com/jonahkichwacoders/EASE-Python-Examples>



<https://github.com/jonahkichwacoders/EASE-Python-Examples>



Disclaimer

We will be writing and running powerful scripts that manipulate files, projects and settings in your workspace.

While we have done our best to ensure the scripts are safe for general consumption, please ensure there is nothing too precious in your workspace!



Hello World!

1. Launch Eclipse & switch to Scripting Perspective
2. Ensure you have a 'Py4J Script Shell' open
3. Open file EASE.htm and follow the instructions for 'Hello World' section
4. When you finish, let us know your Python version (sys.version)!

EASE-ily Making the Most of Eclipse with Python

EASE & PYTHON BACKGROUND

EASE Scripting Framework

Allows for easy extension and control of Eclipse IDE using your favourite scripting language.

Eclipse Advanced Scripting Environment

Python
(Py4J)

Jython

Javascript
(Rhino,
Nashorn)

Groovy

Native
Java

[Your
Interpreter
Here]

Python & EASE: A brief history

EclipseCon France, June 2015:

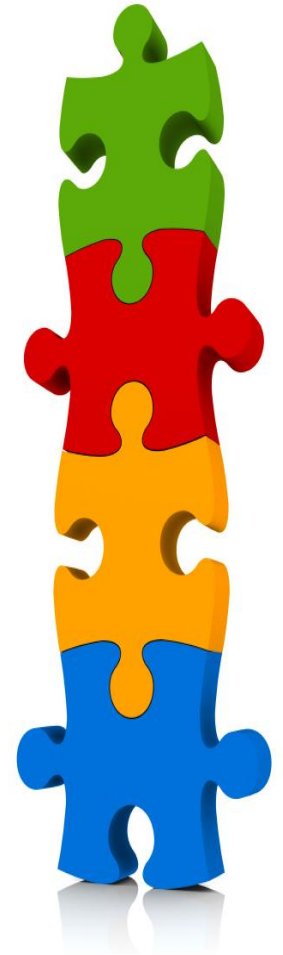
“Integrating Python and Java in Eclipse”

EclipseCon Europe, October 2015:

“I Love Scripting”

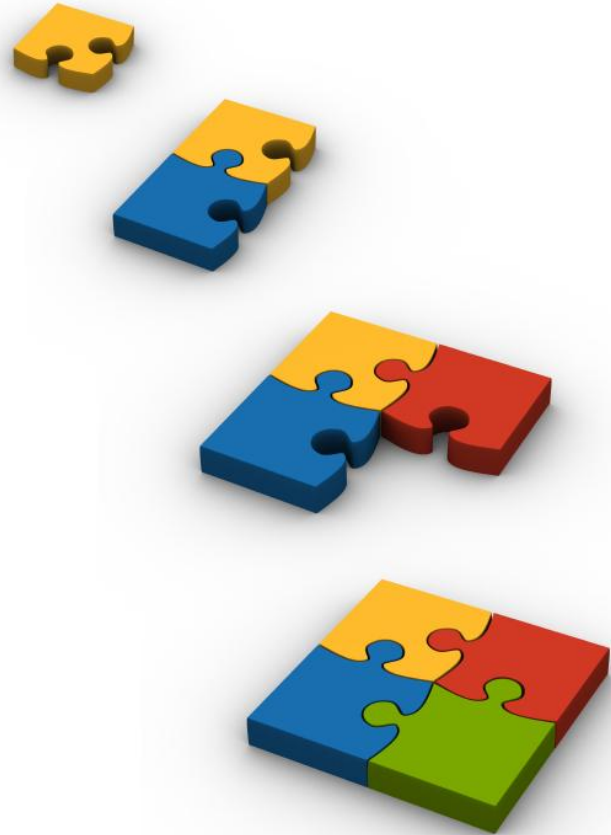
EclipseCon France, June 2016:

“EASE-ily Making the Most of Eclipse with Python”



Why Python?

- Great beginner language
- Heavily used for Science
- Heavily used for Embedded e.g. GDB
- Lots and lots of fast, powerful libraries e.g. numpy



Enabling Technology: Py4J

- ☑ “A Bridge between Java and Python”
- ☑ Allows Python to access JAVA classes and vice versa
- ☑ Open source, BSD licence
- ☑ Used by Apache Spark for PySpark
- ☑ Developed by Barthelemy Dagenais
- ☑ New ‘Pinned-thread’ threading model to work with Eclipse/SWT



Eclipse Advanced Scripting Module

- ☑ Eclipse Project in Incubation
- ☑ Modular approach
- ☑ Good Eclipse integration e.g. Keywords, script explorer, etc

EASE-ily Making the Most of Eclipse with Python

GETTING STARTED

EASE MODULES

- Logical groupings of functionality
- Provide easy access to Eclipse and Java APIs

UI

Resources

Platform

Launch

Charting

Modeling



Modules Exercises

Exercise 2.1 UI Module

Hint: *showInputDialog, showInfoDialog*

Exercise 2.2 Resources Module

Hint: *createProject, getProject*

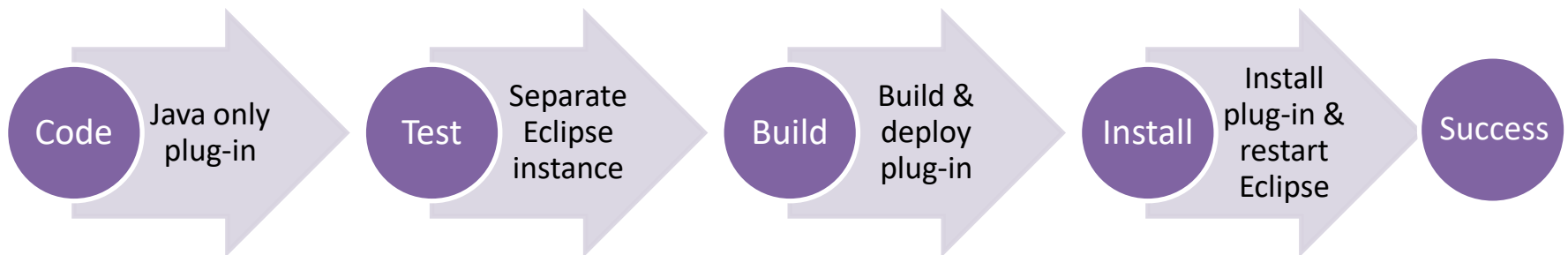
Hint: *IProject.addFile, createFile, refreshResource*

EASE-ily Making the Most of Eclipse with Python

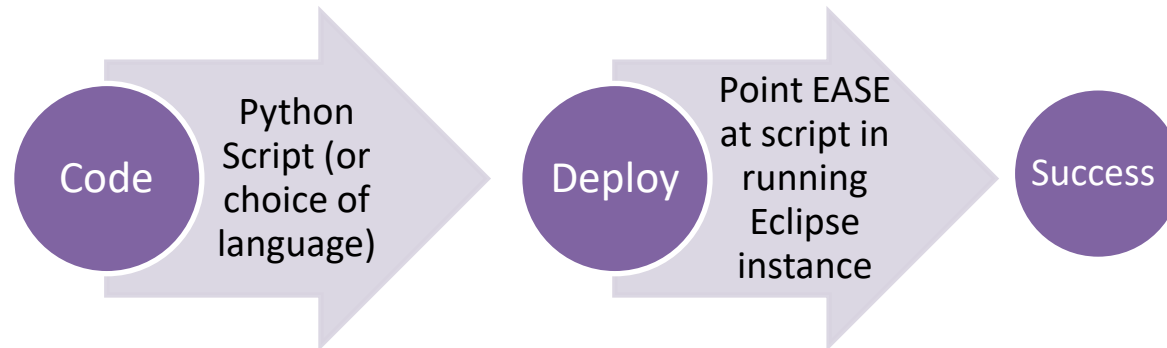
PYTHON & EASE: VISION AND ROADMAP

<https://github.com/jonahkichwacoders/EASE-Python-Examples>

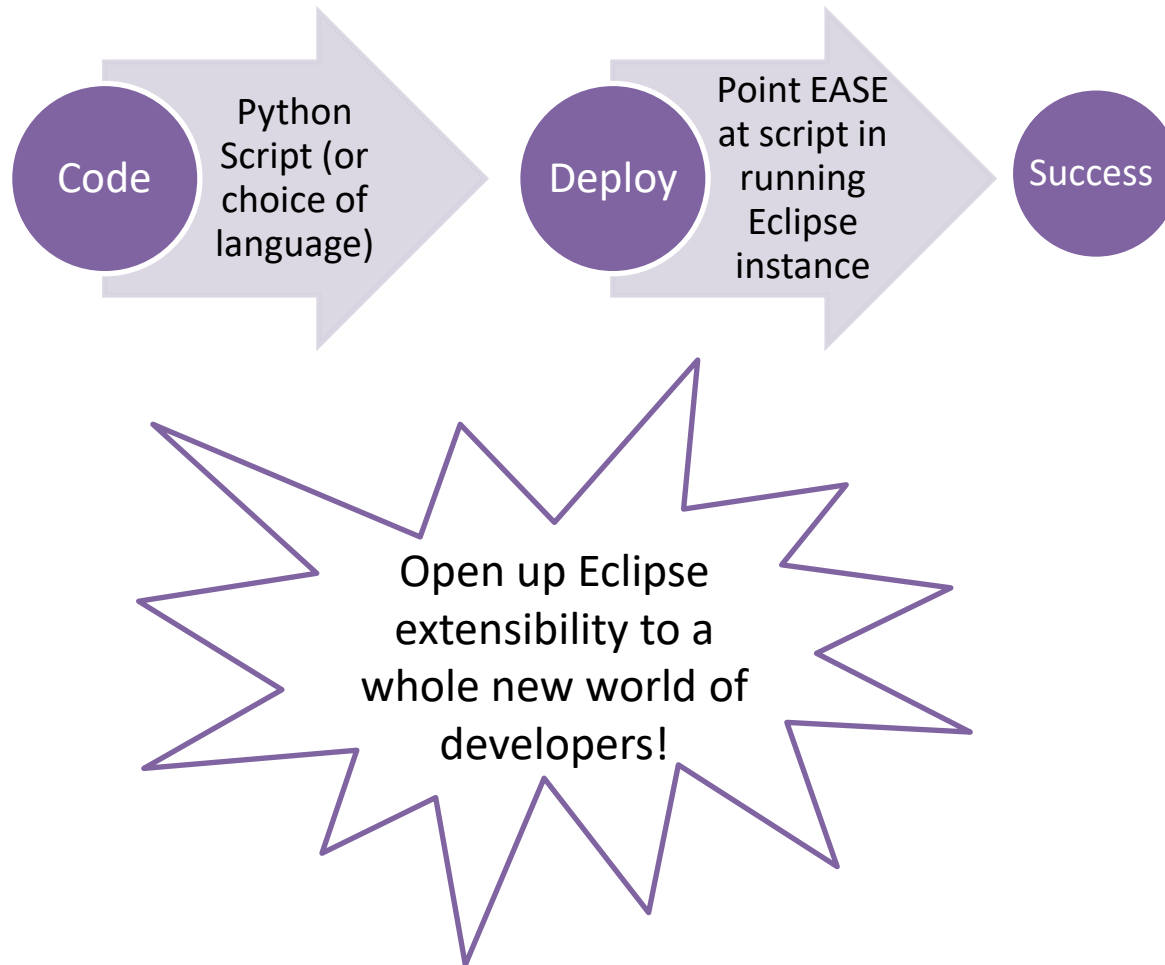
Extending Eclipse (Traditional)



Extending Eclipse with EASE



Extending Eclipse with EASE



Python Scripting with EASE

Be more
productive

Extend Eclipse
quickly

Integrate
Python
libraries

Share scripts
with others
easily

Python Scripting RoadMap

Short Term (Funded)

- Py4J Interpreter Integration
- Critical bug fixes
- Release Engineering

Near Term

- Pythonic Behaviour (Usability)
- PyDev Integration
- AnalysisRPC
- CDT Integration

Future

- Remote Support
- Debugging Support
- IPython Notebook
- Recording

Pythonic Behaviour (Usability)

Namespace pollution

Running scripts with Python directly

More Python-friendly API

PyDev Integration

Autocompletion in console

Autocompletion in editor

Integration of PyDev Console

IPython terminal support

Shared/Automatic Interpreter Configuration

AnalysisRPC

Custom solution to move data around
Enhanced Remote Procedure Call

- Datasets (ndarrays)
- Regions Of Interest (ROIs)
- Exceptions

Language Independent

Server and client written for Java and Python

Integration with Science WG Projects
(Visualisation, Workflows and Data Structures)

Python

```
x = numpy.ndarray(...)  
y = numpy.ndarray(...)  
rpcclient.Plotter.line( x, y)
```

FLATTEN

Java

```
import org.eclipse.dawnscli.IDataset  
class Plotter {  
    public void line(  
        IDataset x, IDataset y) {  
        // ...  
    }  
}
```

UNFLATTEN

Transport

(XML-RPC + Disk/Memory)

Meta and simple data:
Target → Plotter
Method → line
x → /tmp/x
y → /tmp/y

Datasets:
/tmp/x
/tmp/y

Java

```
interface IFilter {  
    IDataset filter(IDataset data);  
}  
//create client for PyFilter  
client.newProxyInstance(IFilter.class)  
IDataset result = proxy.filter(data)
```

FLATTEN

Python

```
class PyFilter:  
    def filter(data):  
        # manipulate ndarray data  
        return data  
# create server  
server.add_handler(PyFilter())
```

UNFLATTEN

Transport

(XML-RPC + Disk/Memory)

Meta and simple data:
Target → PyFilter
Method → filter
data → /tmp/data

Datasets:
/tmp/data

CDT Integration

Visual Studio Macro-like support

Debug Module

C Editor Support Module

GDB Python Interface Integration

EASE-ily Making the Most of Eclipse with Python

EXERCISES

Python Scripting with EASE

Be more
productive

Extend Eclipse
quickly

Integrate
Python
libraries

Share scripts
with others
easily

EASE-ily Making the Most of Eclipse with Python

WRITING EASE MODULES

EASE-ily Making the Most of Eclipse with Python

CONCLUSION

EASE-ily Make The Most of Eclipse

AUTOMATE

EXTEND

INTEGRATE

SHARE

Thank you & Questions



Tracy Miranda

tracy@kichwacoders.com

[@tracymiranda](#)

Jonah Graham

jonah@kichwacoders.com

Christian Pontesegger

christian.pontesegger@web.de