Building The Virtual Machine

Igor Stasenko

RMoD Team
INRIA Lille Nord-Europe

Edinburgh
ESUG, 2011

Outline

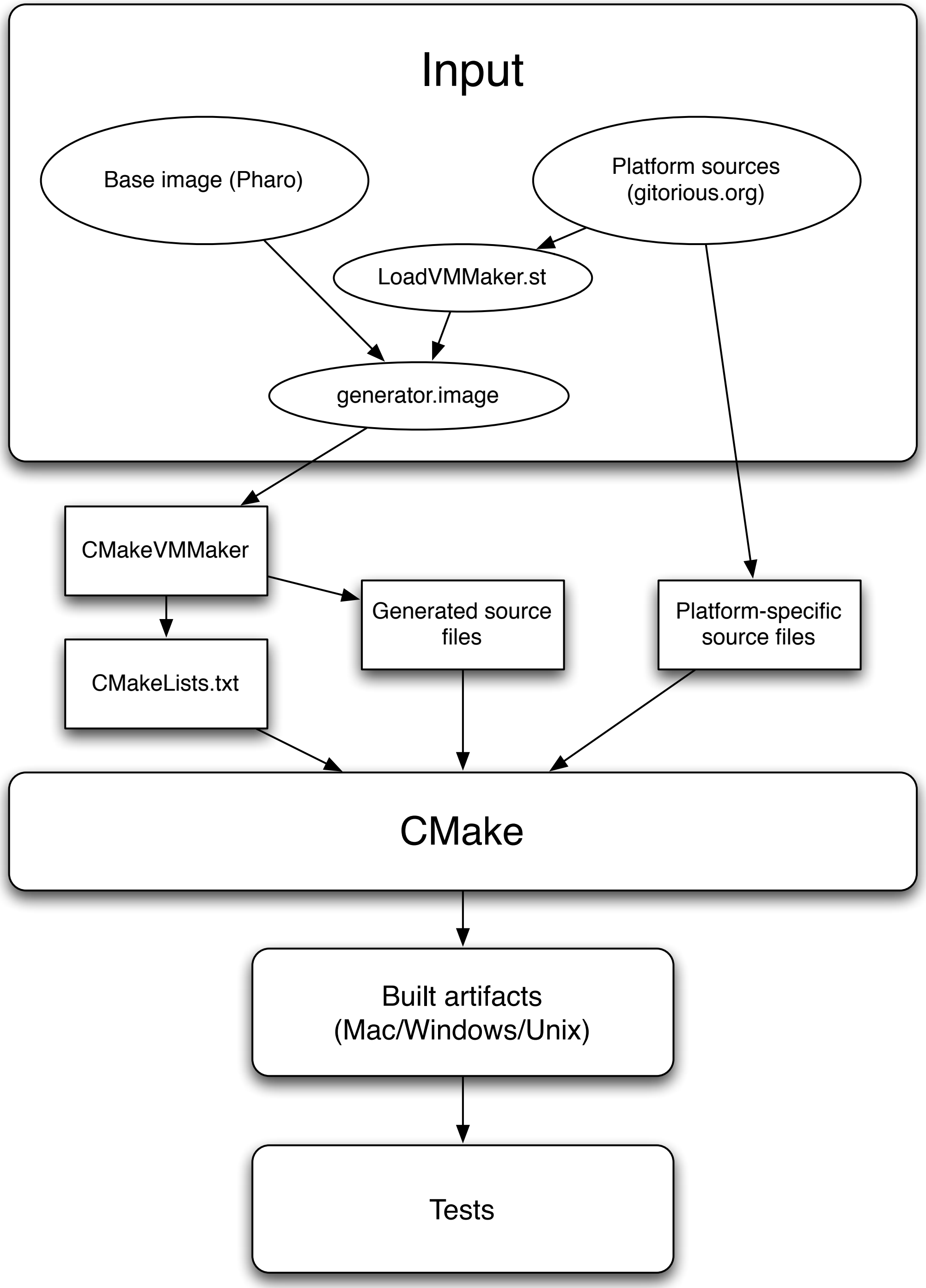
- Automated VM build system
- Preparing your environment for build
- Build VM
- Customizing builds

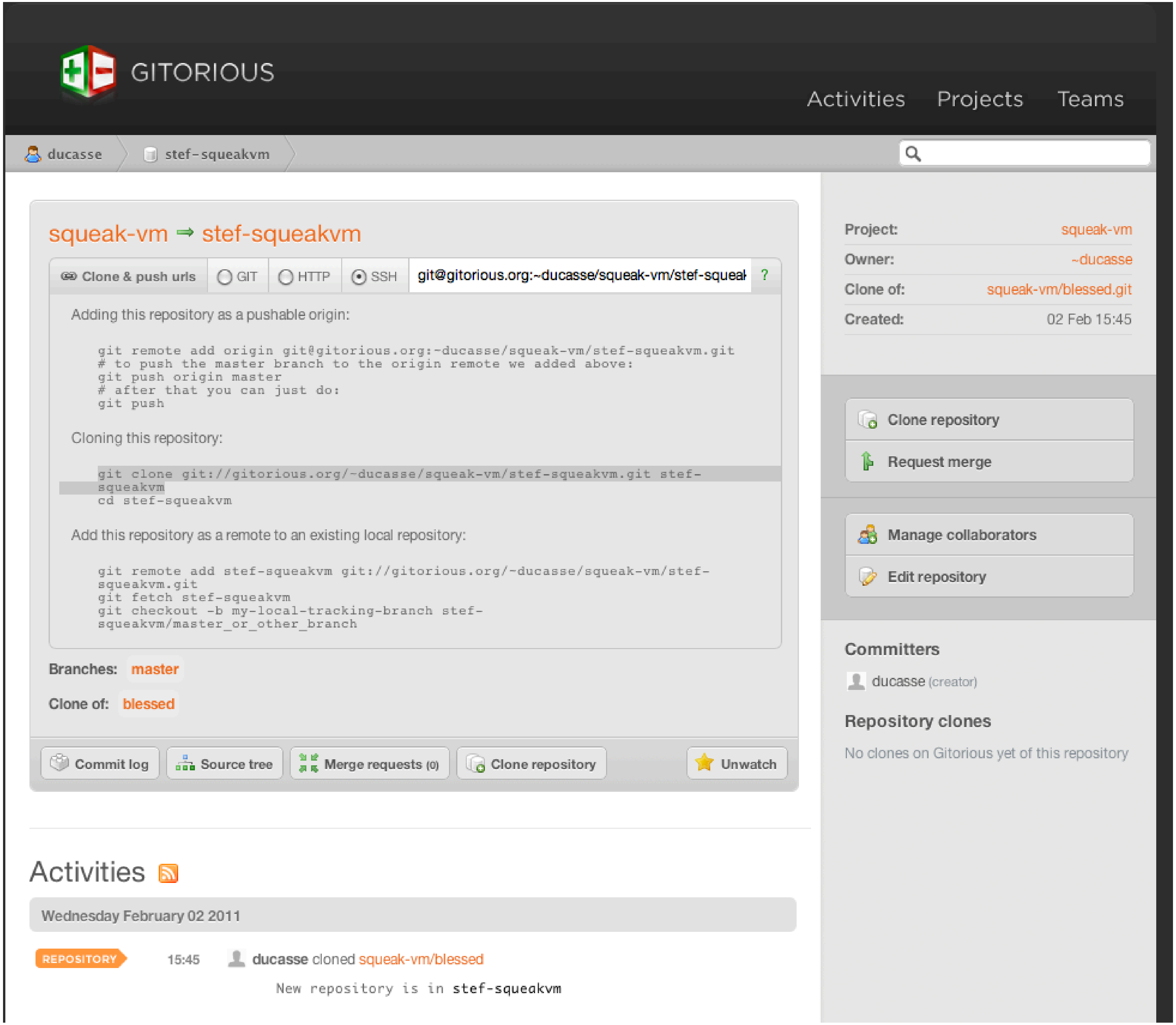
Automated build system

Git tracker Job

VM Builder Job

VM test runner job





squeak-vm ⇒ stef-squeakvm

Clone & push urls ○ GIT ○ HTTP ● SSH git@gitorious.org:~ducasse/squeak-vm/stef-squeak ?

Adding this repository as a pushable origin:

```
git remote add origin git@gitorious.org:~ducasse/squeak-vm/stef-squeakvm.git
# to push the master branch to the origin remote we added above:
git push origin master
# after that you can just do:
git push
```

Cloning this repository:

```
git clone git://gitorious.org/~ducasse/squeak-vm/stef-squeakvm.git stef-squeakvm
cd stef-squeakvm
```

Add this repository as a remote to an existing local repository:

```
git remote add stef-squeakvm git://gitorious.org/~ducasse/squeak-vm/stef-squeakvm.git
git fetch stef-squeakvm
git checkout -b my-local-tracking-branch stef-squeakvm/master_or_other_branch
```

Branches: master

Clone of: blessed

Project:	squeak-vm
Owner:	~ducasse
Clone of:	squeak-vm/blessed.git
Created:	02 Feb 15:45

Committers

ducasse (creator)

Repository clones

No clones on Gitorious yet of this repository

Activities

Wednesday February 02 2011

REPOSITORY 15:45 ducasse cloned squeak-vm/blessed

New repository is in stef-squeakvm

Why git?

Git is distributed source management tool, while SVN is centralized one.

This means that git fits better for developing and managing open-source projects driven by community rather than single or selected team of developers.

With gitorious.org, one can easily make own branch publicly available, without need to ask for VIP's permission.

Getting VM platform sources

```
git clone git://gitorious.org/cogvm/blessed.git cogvm
```

(but do not try it right now!!)

For our exercise, download sources tarball from here:
https://ci.lille.inria.fr/pharo/view/Cog/
Job named: Cog Git Tracker (blessed)

cog.tar.gz
vmmaker-image.zip

Sources directory structure

/ <root dir>	
platforms/	
Cross/	common code for all platforms
Mac OS/	
RiscOS/	platform-specific source code & configuration files/tools
unix/	
win32/	
src/	VMMaker generated code
build/	directory used for cmake configuration files
results/	built VMs are stored here

Loading VMMaker into image

- take a fresh Pharo 1.3 or Pharo 1.4 image
- place it into /build subdir
- evaluate a script from

    codegen-scripts/LoadVMMaker.st

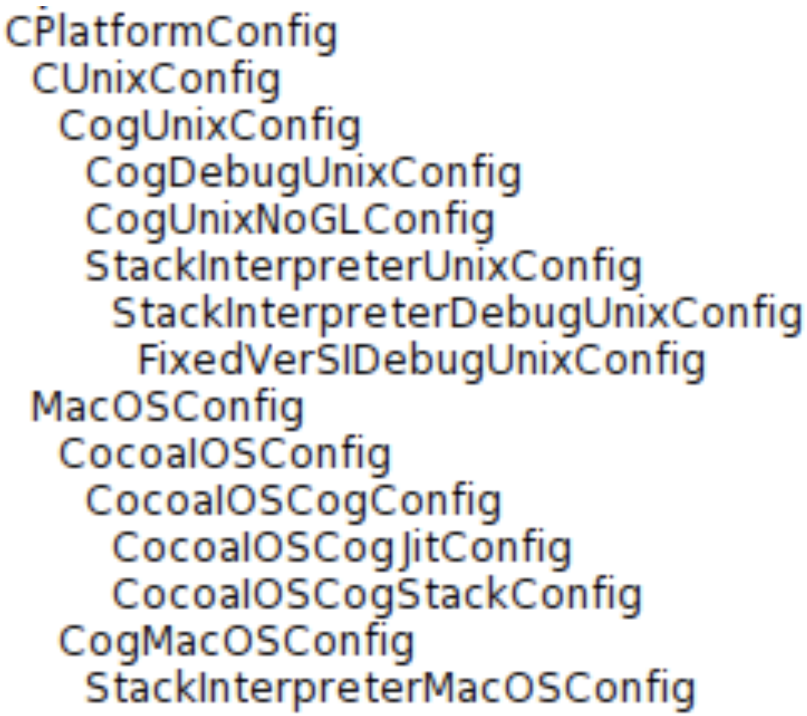
  or do it by youself:

```
Gofer new
        squeaksource: ‘MetacelloRepository’;
        package: ‘ConfigurationOfCog’;
        load.
ConfigurationOfCog project lastVersion load
```

Generating source code + makefiles

<MyConfiguration> generateWithSources

Depends what you building:



```
CPlatformConfig
  CUnixConfig
    CogUnixConfig
      CogDebugUnixConfig
      CogUnixNoGLConfig
      StackInterpreterUnixConfig
        StackInterpreterDebugUnixConfig
          FixedVerSIDebugUnixConfig
  MacOSConfig
    CocoaIOSConfig
      CocoaIOSCogConfig
        CocoaIOSCogJitConfig
        CocoaIOSCogStackConfig
    CogMacOSConfig
      StackInterpreterMacOSConfig
```

Preparing build environment

Mac
Windows
Unix

Too much for the slide :)

Follow a guide, made by Mariano Martinez Peck:

http://code.google.com/p/cog/wiki/Guide

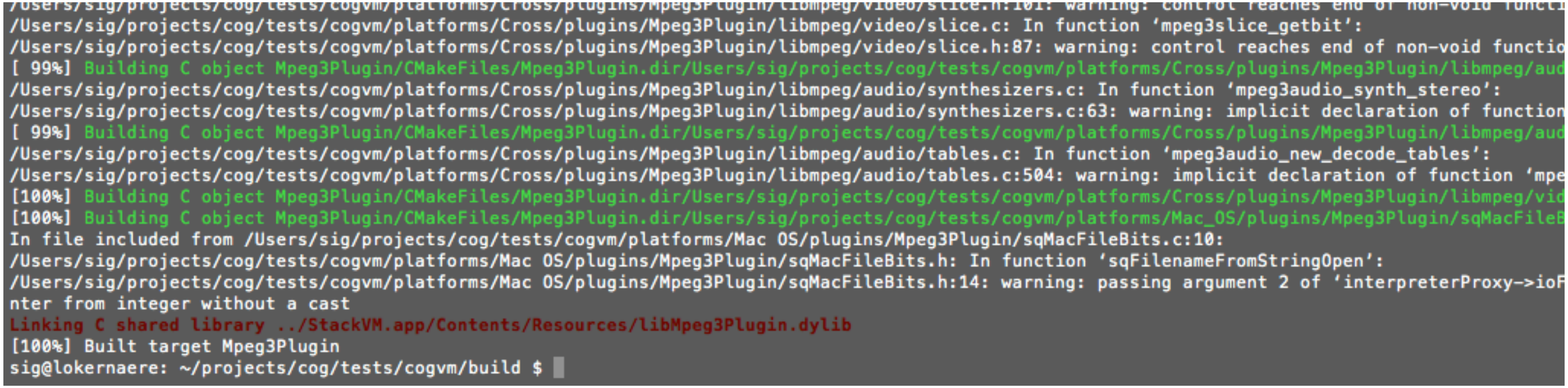
Building VM

cd build

cmake .   (on Windows: cmake -G"MSYS Makefiles .)

make

.. run ..



```
/Users/sig/projects/cog/tests/cogvm/platforms/Cross/plugins/Mpeg3Plugin/libmpeg/video/slice.h:101: warning: control reaches end of non-void functi
/Users/sig/projects/cog/tests/cogvm/platforms/Cross/plugins/Mpeg3Plugin/libmpeg/video/slice.c: In function 'mpeg3slice_getbit':
/Users/sig/projects/cog/tests/cogvm/platforms/Cross/plugins/Mpeg3Plugin/libmpeg/video/slice.h:87: warning: control reaches end of non-void functio
[ 99%] Building C object Mpeg3Plugin/CMakeFiles/Mpeg3Plugin.dir/Users/sig/projects/cog/tests/cogvm/platforms/Cross/plugins/Mpeg3Plugin/libmpeg/aud
/Users/sig/projects/cog/tests/cogvm/platforms/Cross/plugins/Mpeg3Plugin/libmpeg/audio/synthesizers.c: In function 'mpeg3audio_synth_stereo':
/Users/sig/projects/cog/tests/cogvm/platforms/Cross/plugins/Mpeg3Plugin/libmpeg/audio/synthesizers.c:63: warning: implicit declaration of function
[ 99%] Building C object Mpeg3Plugin/CMakeFiles/Mpeg3Plugin.dir/Users/sig/projects/cog/tests/cogvm/platforms/Cross/plugins/Mpeg3Plugin/libmpeg/aud
/Users/sig/projects/cog/tests/cogvm/platforms/Cross/plugins/Mpeg3Plugin/libmpeg/audio/tables.c: In function 'mpeg3audio_new_decode_tables':
/Users/sig/projects/cog/tests/cogvm/platforms/Cross/plugins/Mpeg3Plugin/libmpeg/audio/tables.c:504: warning: implicit declaration of function 'mpe
[100%] Building C object Mpeg3Plugin/CMakeFiles/Mpeg3Plugin.dir/Users/sig/projects/cog/tests/cogvm/platforms/Cross/plugins/Mpeg3Plugin/libmpeg/vid
[100%] Building C object Mpeg3Plugin/CMakeFiles/Mpeg3Plugin.dir/Users/sig/projects/cog/tests/cogvm/platforms/Mac_OS/plugins/Mpeg3Plugin/sqMacFileB
In file included from /Users/sig/projects/cog/tests/cogvm/platforms/Mac OS/plugins/Mpeg3Plugin/sqMacFileBits.c:10:
/Users/sig/projects/cog/tests/cogvm/platforms/Mac OS/plugins/Mpeg3Plugin/sqMacFileBits.h: In function 'sqFilenameFromStringOpen':
/Users/sig/projects/cog/tests/cogvm/platforms/Mac OS/plugins/Mpeg3Plugin/sqMacFileBits.h:14: warning: passing argument 2 of 'interpreterProxy->ioF
nter from integer without a cast
Linking C shared library ../StackVM.app/Contents/Resources/libMpeg3Plugin.dylib
[100%] Built target Mpeg3Plugin
sig@lokernaere: ~/projects/cog/tests/cogvm/build $
```

Customizing builds: plugins

Adding custom plugins:

```
CogMacOSConfig new
  addExternalPlugins: #( FT2Plugin );
  addInternalPlugins: #( UnixOSProcessPlugin );
  generateSources; generate.
```

You could also completely replace the plugins to build, using #internalPlugins: and #externalPlugins: to supply own set of plugins for build.

Creating own configuration

- as you do for anything in smalltalk:
  - subclass from appropriate XYZConfig class
  - change flags/settings
  - provide own default plugins
  - etc

```
MyConfig generateWithSources.
```