

spm8Batch SDK 2012-08

In general you will focus on writing your algorithm as most of the bash wrapper functionality is already provided.

A command in the spm8Batch suite will write a bash script in the path `matlabScripts/spm8Batch/[command]/[year_month]`, as well as any supporting “.m” file.

The names of these files are automatically generated by the wrapper code.

The process is then launched into either background (default) or into the foreground (‘-B’ flag)

spm8Batch SDK 2012-08

```
#!/bin/bash

#####
#
# command to reconstruct spiral images from pFiles
#
#####

#
# Main reconstruction function.
#

VERSION="1.0"
VERSIONDATE="2012-08-20"

# Find out where the current command lives and execute some common code.

theFullCommand="$0 ${*}"
execDIR=`pwd`

# Find out where the current command lives and execute some common code.

theCommand=`which $0`
thisDir=`dirname $theCommand`

thisCommand=`echo $theCommand | awk -F/ '{print $NF}'`

allowedOptions="BDdfMNvz"

# This piece of code all of the spm8Batch scripts will use
. ${thisDir}/auxiliary/commonCode_AllScriptsStart

# This is the common in-line code for the warping steps.
. ${thisDir}/auxiliary/commonCode_reconSpiral

# This is common in-line code for all spm8Batch script to make sure arguments were passed
. ${thisDir}/auxiliary/commonCode_checkIfArgsPassed

##
# The actual function.
#

--:-- reconSpiral Top L39 Git-spm8Batch/daisyChained (Shell-script[bash])-----
Beginning of buffer
```

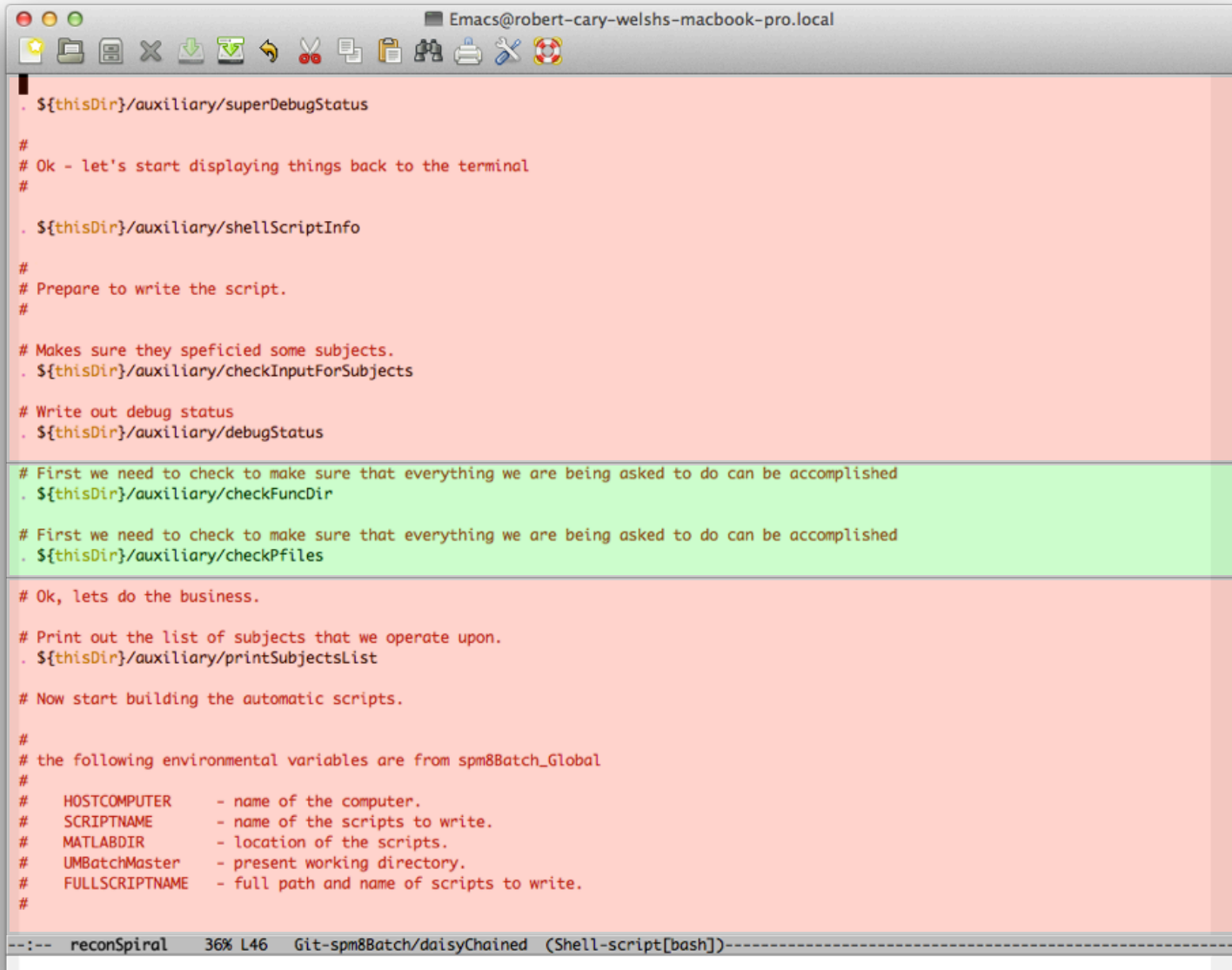
enable flags

command specific checks

Code highlighted in red is standard code that all modules needs.

Code in green is the command specific code.

spm8Batch SDK 2012-08



```
Emacs@robert-cary-welshs-macbook-pro.local

. ${thisDir}/auxiliary/superDebugStatus

#
# Ok - let's start displaying things back to the terminal
#

. ${thisDir}/auxiliary/shellScriptInfo

#
# Prepare to write the script.
#

# Makes sure they specified some subjects.
. ${thisDir}/auxiliary/checkInputForSubjects

# Write out debug status
. ${thisDir}/auxiliary/debugStatus

# First we need to check to make sure that everything we are being asked to do can be accomplished
. ${thisDir}/auxiliary/checkFuncDir

# First we need to check to make sure that everything we are being asked to do can be accomplished
. ${thisDir}/auxiliary/checkPfiles

# Ok, lets do the business.

# Print out the list of subjects that we operate upon.
. ${thisDir}/auxiliary/printSubjectsList

# Now start building the automatic scripts.

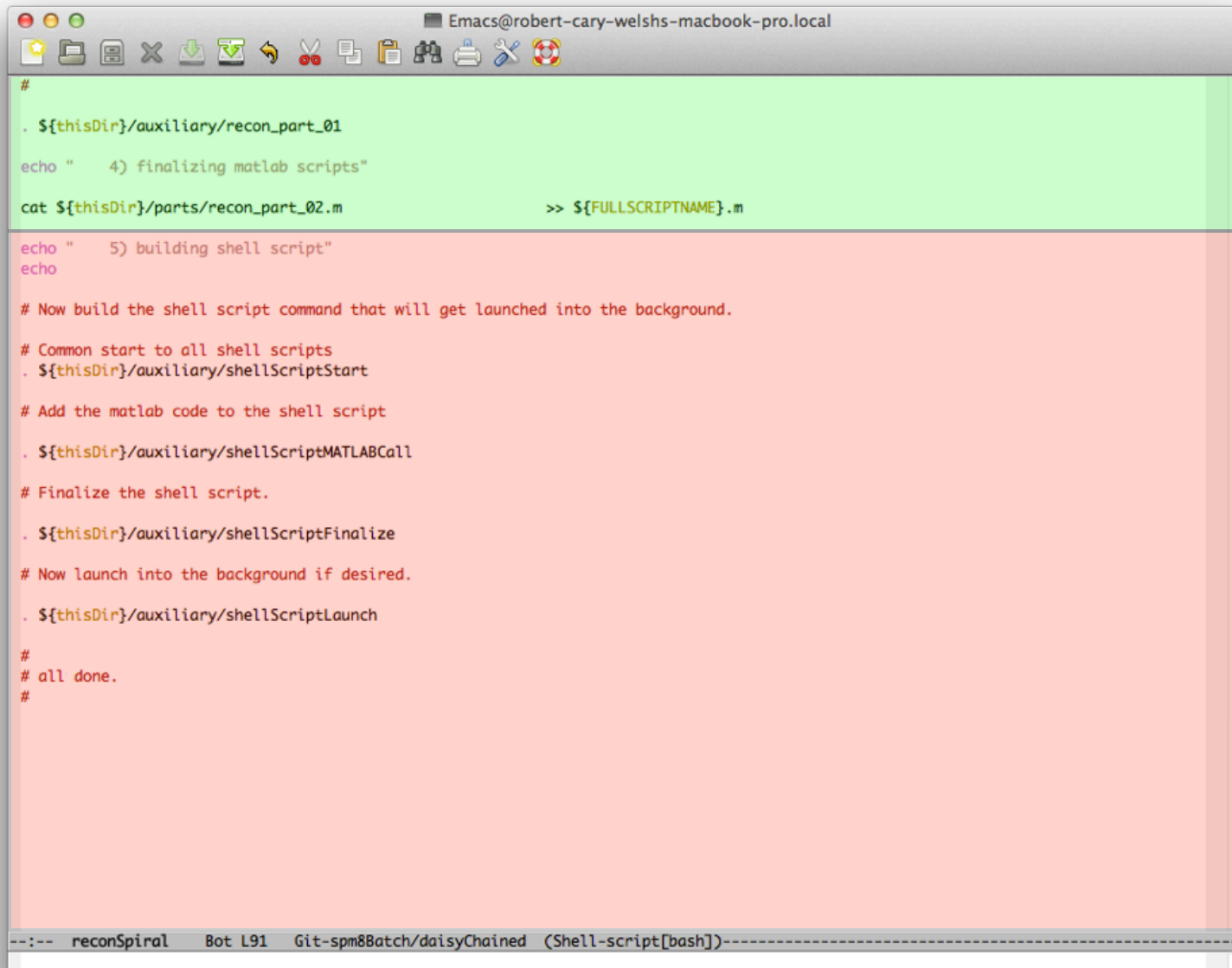
#
# the following environmental variables are from spm8Batch_Global
#
#   HOSTCOMPUTER    - name of the computer.
#   SCRIPTNAME      - name of the scripts to write.
#   MATLABDIR       - location of the scripts.
#   UMBatchMaster    - present working directory.
#   FULLSCRIPTNAME   - full path and name of scripts to write.
#

--:-- reconSpiral 36% L46 Git-spm8Batch/daisyChained (Shell-script[bash])
```

command specific checks

Code highlighted in red is standard code that all modules needs.
Code in green is the command specific code.

spm8Batch SDK 2012-08



```
#
. ${thisDir}/auxiliary/recon_part_01
echo "    4) finalizing matlab scripts"
cat ${thisDir}/parts/recon_part_02.m          >> ${FULLSCRIPTNAME}.m

echo "    5) building shell script"
echo

# Now build the shell script command that will get launched into the background.

# Common start to all shell scripts
. ${thisDir}/auxiliary/shellScriptStart

# Add the matlab code to the shell script
. ${thisDir}/auxiliary/shellScriptMATLABCall

# Finalize the shell script.
. ${thisDir}/auxiliary/shellScriptFinalize

# Now launch into the background if desired.
. ${thisDir}/auxiliary/shellScriptLaunch

#
# all done.
#
```

command specific code
for building scripts

Code highlighted in red is standard code that all modules needs.
Code in green is the command specific code.

spm8Batch SDK 2012-08

To build a module you need the following:

[module] -- written in bash, but no '.sh'

auxiliary/initialize_[module]

auxiliary/help/[module]_Help

auxiliary/help/help_[module]_options

spm8Batch SDK 2012-08

Environmental Variables:

UMSTREAM_STATUS_FILE - if this variable is defined then abort status will be written. if the file does not exist or is empty then the command was successful.

This is passed into matlab processing and is a global variable known as **UMBatchStatusFile**

spm8Batch SDK 2012-08

Dependencies/reconSpiral

reconSpiral

auxiliary/commonCode_reconSpiral

auxiliarychechReconParts

auxiliary/initialization_reconSpiral

auxiliary/checkFuncDir

auxiliary/checkPfiles

auxiliary/recon_part_01

parts/recon_part_01.m

parts/recon_part_02.m

help/help_reconSpiral_Options

help/reconSpiral_Help

matlabScripts/UMBatchRecon.m