

# DailyBuild

第 2.0 版 – 2018/4/16

本ドキュメントでは、Springhead の日々のメンテナンス作業 (dailybuild の結果の確認、Wiki ページのメンテナンス等) およびそれらに関わるスクリプトについて説明する。

## 改訂履歴

第 2.0 版	スクリプトの python 化に伴う変更
第 1.0 版	初版

## 目次

0.1	処理を制御する環境変数 . . . . .	2
0.2	Script 詳細 . . . . .	4
0.2.1	DailyBuildWrapper . . . . .	4
0.2.2	DailyBuild . . . . .	5
0.2.3	TestMainGit . . . . .	6
0.2.4	SpringheadTest . . . . .	6
0.2.5	ConstDefs . . . . .	6
0.2.6	ControlParams . . . . .	6
0.2.7	Traverse . . . . .	6
0.2.8	BuildAndRun . . . . .	6
0.2.9	CloseSrcControl . . . . .	6
0.2.10	VisualStudio . . . . .	6
0.2.11	KeyInterruption . . . . .	6
0.2.12	StructuredException . . . . .	6
0.2.13	TestResult . . . . .	6
0.2.14	VersionControlSystem . . . . .	6

### 0.1 処理を制御する環境変数

以下に示す DailyBuild の各ステップは、実行時の環境変数で制御することができる。各ステップの「制御変数名」に記述された環境変数が "skip" と設定されているときに限りそのステップは実行されない。毎日の DailyBuild タスクの実行時にはこれらの環境変数はすべて未定義としておく。

これはデバッグ及びローカルでのテストのときに使用する。

各ステップの説明：

#### 1. Springhead の更新を反映する

制御変数名 DAILYBUILD\_UPDATE\_SPRINGHEAD

処理 以下のローカルレポジトリの内容を最新バージョンにする。

"I:/DailyBuild/Springhead/"

"I:/DailyBuild/DailyBuildResult/Result/"

#### 2. テストディレクトリをクリアする

制御変数名 DAILYBUILD\_CLEANUP\_WORKSPACE  
処理 テストディレクトリ (SpringheadTest – 引数で指定) をクリアする。  
指定されたディレクトリがなければ何もしない。

3. SpringheadTest にソースツリーを取得する

制御変数名 DAILYBUILD\_CLEANUP\_WORKSPACE  
処理 テストディレクトリに最新バージョンを Clone する。

4. Springhead ライブラリを作成する

制御変数名 DAILYBUILD\_EXECUTE\_STUBBUILD  
処理 呼出し時の引数及びオプションは次のとおり  
-p x64 -c Release -t 14.0 -C unuse -S (次の行に続く)  
"result/dailybuild.result" "dailybuild.control"  
result ファイルを初期化してからビルドする。

5. "src/tests" 以下のテストを実行する

制御変数名 DAILYBUILD\_EXECUTE\_BUILDRUN  
処理 呼出し時の引数及びオプションは前項と同じ。  
ただし result ファイルの初期化は行なわない (オプション -S なし)

6. "src/Samples" 以下のビルドを実行する

制御変数名 DAILYBUILD\_EXECUTE\_SAMPLEBUILD  
処理 呼出し時の引数及びオプションは前項と同じ。  
ただし result ファイルの初期化は行なわない (オプション -S なし)

7. 結果ファイルをコミットする

制御変数名 DAILYBUILD\_COMMIT\_RESULTOG  
処理 Springhead の最新コミット ID を取り出して記録ファイルを作成し、  
結果ファイルとともにレポジトリに登録する。  
ファイル "Springhead.commit.id", "result.log"  
登録先 "git.haselab.net/DailyBuild/Result"

8. 履歴ファイルを作成する

制御変数名 DAILYBUILD\_GEN\_HISTORY  
処理 テスト履歴ファイル "History.log" 及びタイムスタンプファイル  
"Test.date" を作成する。

9. ログファイルを Web にコピーする

制御変数名	DAILYBUILD_COPYTO_BUILDLOG
処理	テストログファイルを Web ウェブサーバにコピーする。
コピー先	\\haselab\HomeDirs\WWW\docroots\ (次の行に続く) springhead\daily_build\log 以下

## 10. ドキュメントを作る

制御変数名	DAILYBUILD_EXECUTE_MAKEDOC
処理	doxygen によりドキュメントを作成する ("MakeDoc.py")

## 11. 成果物を Web にコピー

制御変数名	DAILYBUILD_COPYTO_WEBBASE
処理	DailyBuild の成果物 Web サーバにコピーする
ファイル	ライブラリファイル及びドキュメント
コピー先	\\haselab\HomeDirs\WWW\docroots\ (次の行に続く) springhead\daily_build/generated 以下

## 0.2 Script 詳細

### 0.2.1 DailyBuildWrapper

タスク DailyBuild を実行するためのラップスクリプト。関連するスクリプトの変更を確実に反映させるために、次の 2 段階で DailyBuild を起動する。

```
python DailyBuild.py -u %*
python DailyBuild.py -U %*
```

(Windows)

または

```
python DailyBuild.py -u $*
python DailyBuild.py -U $*
```

(unix)

1 回目のスクリプト起動ではレポジトリ Springhead の更新のみを行なう。2 回目は更新されたスクリプトの起動であり、テストレポジトリを新たに作成してテストを実行する。

このバッチファイルをタスクスケジューラに登録することでタスク DailyBuild を実行する (Windows)。登録情報は次のとおり。

トリガー	毎日 2:02 に起動
操作	プログラムの開始
スクリプト (P)	I:/DailyBuild/Springhead/core/test/DailuBuild.bat
引数	-t 14.0 -c Release -p x64 (次の行に続く) SpringheadTest DailyBuildResult/Result

## 0.2.2 DailyBuild

0.1 で示した step 1. から step 3. までの処理を実行する。step 4. 以降の処理は、ここから TestMainGit.py を呼出して実行させる。

### 起動方法

```
python DailyBuild.py [options] test-repository result-repository
```

options ( <sup>†</sup> はデフォルトを示す)

- c CONF   ビルド構成 (Debug, Release<sup>†</sup>, Trace)
- p PLAT   プラットフォーム (x86, x64<sup>†</sup>)
- t TOOL   ツールセット (14.0<sup>†</sup>)
- u        step 1. のみ実行して終了する
- U        step 2. 以降のみ実行する
- A        step 3. 以降のみ実行する
- D        コマンドを表示するだけで実行はしない (dry run)

### 引数

test-repository    テストレポジトリ  
                  任意のディレクトリでよいが、既存のディレクトリの場合には、その内容はすべて破棄される。

result-repository   テスト結果記録用レポジトリ  
                  通常 "DailyBuildResult/Result" を指定する。

これらのディレクトリは、テストを起動するディレクトリから 3 階層上のディレクトリ (例えば "top/Springhead/core/test" ならば "top" ) からの相対パスで指定すること。

### 注意

第 2 引数で指定した “テスト結果記録用レポジトリ” へ push するためのアクセス権を得るために、git.haselab.net のユーザ名及びパスワードを記録したファイル "hasegit.user" を上記 "top" の直下に配置してある。コマンドラインからのアクセスでパスワードを指定しなくてもよい方法が見つけれられるまで、こ ZZ のファイルを移動したり削除したりしないこと。このファイルの内容は ‘DailyBuild:Haselab0’ である。

step 1. レポジトリ Springhead 及び DailyBuildResult/Result を最新の状態にする。

使用するコマンドは git pull --all、URL はそれぞれ

github.com/sprphys/Springhead

git.haselab.net/DailyBuild/Result

である。

オプション -U が指定されたならば、この step は実行しない。

オプション -u が指定されたならば、ここでプログラムを終了する。

step 2. テストレポジトリを削除する。これは毎回スクラッチからテストを実行するためである。

step 3. テスト用レポジトリを clone する。使用するコマンドは

git clone --recurse-submodules (次の行に続く)

https://github.com/sprphys/Springhead SpringheadTest

step 4. 以降 テストレポジトリのテストディレクトリ ("SpringheadTest/core/test") へ移動し、  
そこでTestMainGitを実行する。

```
python TestMainGit.py -p x64 -c Release -t 14.0 (次の行に続く)SpringheadTest DailyBuildResult/Result
```

- 0.2.3 TestMainGit
- 0.2.4 SpringheadTest
- 0.2.5 ConstDefs
- 0.2.6 ControlParams
- 0.2.7 Traverse
- 0.2.8 BuildAndRun
- 0.2.9 CloseSrcControl
- 0.2.10 VisualStudio
- 0.2.11 KeyInterruption
- 0.2.12 StructuredException
- 0.2.13 TestResult
- 0.2.14 VersionControlSystem