

Asteria[✦]

テストフレームワーク



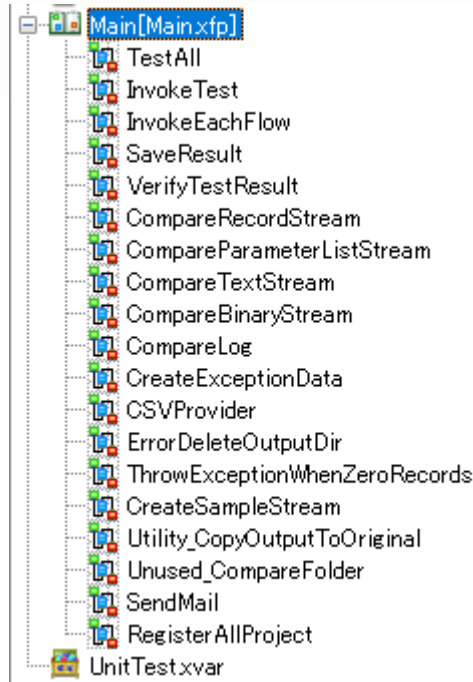
テストフレームワークとは？

- テスト実行用のフローと各種コンポーネントのセット
- テスト対象のユーザーのフローをテスト
 - コンポーネントのテストに実績があります
- テスト対象のフローの出力結果を比較
- テスト結果は次の4種類で確認
 - テキストファイル比較
 - エラー内容比較
 - バイナリーファイル比較
 - パラメータリスト比較
 - レコード比較



- コンポーネント
 - [data dir]/system/lib/components/fcutest.jar
- ライブラリ
 - [data dir]/system/lib/userlib/diffutils-1.2.1.jar
- フロー
 - [data dir]/home/[user]/Main.xfp
 - [data dir]/home/[user]/UnitTest.xvar
- フロー実行時の設定ファイル
 - [data dir]/home/[user]/exclude.properties
 - [data dir]/home/[user]/test.properties
- コネクション
 - SendTestCompletedMailConnection





フロー名	説明
TestAll	全プロジェクトのテスト
InvokeTest	1プロジェクトのテスト
RegisterAllProject	全プロジェクトの登録

テストフローを作成している時や
あるフローだけテストしたい場合



InvokeTestで1プロジェクトだけテスト



外部変数セットの編集

ファイル名

外部変数セット名

説明

● 定数

変数名	データ型	初期値	説明
TEST_LOG	String	test.log	テストログのファイル名
ORIGINAL_DIR	String	original	テスト結果の比較元フォルダ
OUTPUT_DIR	String	output	テスト結果の出力フォルダ
DIFF_LOG	String	diff.log	比較結果の出力ファイル名
MSG_DIFF_NO_ORIGINAL	String	No original file.	オリジナルファイルがないときのメッ...
MSG_DIFF_NOT_SUPPOR...	String	Not supported.	ファイルタイプがサポートされていな...
ERRCD_FAILED_TO_DIFF	String	9000	比較に失敗した時のエラーコード
ERRCD_FAILED_TO_EXEC...	String	9001	実行に失敗した時のエラーコード

● リクエスト変数

変数名	データ型	初期値	説明
SuccessCount	Integer	0	
FailureCount	Integer	0	
ErrorCount	Integer	0	

特に変更する必要はありません



- exclude.properties
 - テスト対象としないプロジェクトを指定する
 - [プロジェクト名]=true

```
ExcludeProject1=true  
ExcludeProject2=true
```

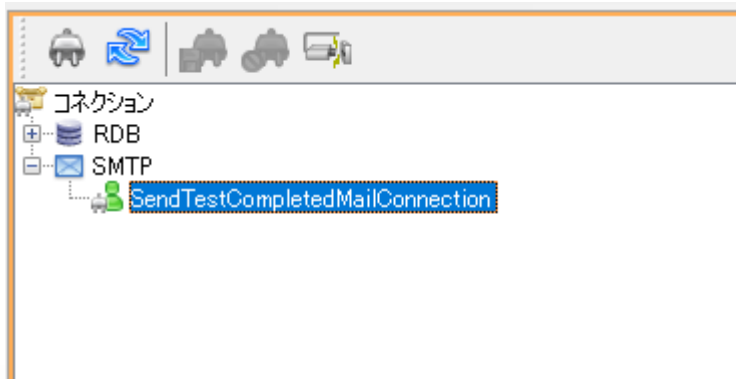


- test.properties
 - 各種プロパティファイルとして使用可能
 - 現在は結果メールの送信先をmail.toで指定
 - SendMailフローのMapper3でTable関数によりファイルを読み込み、mail.toの値を送信先アドレスとして設定

```
mail.to=dev@example.com
```



- SendTestCompletedMailConnection
 - 結果メールの送信に使用するSMTPコネクション
 - 事前に作成しておきます



テストフレームワークの使用手順



1. testframework.zipを適当なフォルダーに解凍すると次の2つのフォルダーが解凍されます
 - homeフォルダー
 - systemフォルダー
2. systemフォルダーを[data dir]/systemに上書コピーします
 - テスト用のjarがコピーされます
3. FSMCでテスト用のユーザーを作成しそのホームフォルダーに1.で解凍されたhome/qatestの中のファイル・フォルダーをコピーします
 - テスト用のフローとテスト対象のサンプルフローがコピーされます



4. ASTERIAを起動します
5. 通知用のSMTPコネクションとなる次の名前のSMTPコネクションを作成します
 - SendTestCompletedMailConnection
6. テスト用ユーザーのホームフォルダーにあるtest.propertiesのmail.toに通知先のメールアドレスを指定します
7. フローデザイナーでテスト用のユーザーに接続し「ツール>コンポーネント/マッパー関数の取得」から「fcutest.jar」をダウンロードします



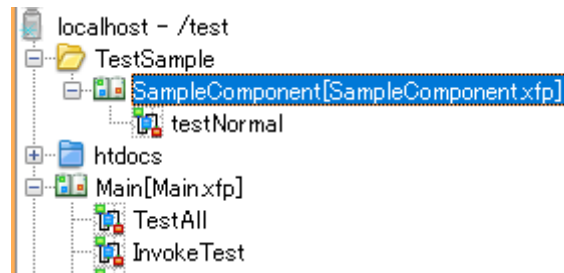
8. フローデザイナーを再起動しテスト用のユーザーで接続します
9. Mainプロジェクトをコンパイルします
10. MainプロジェクトのRegisterAllProjectフローを実行しテスト対象のプロジェクトを登録します
11. MainプロジェクトのTestAllフローを実行します
12. 通知先のメールに結果が送信されますのでログを確認します
 - テスト用のユーザーのホームフォルダーにあるtest.logとdiff.logをzipしメールに添付しています



テストフローの作成



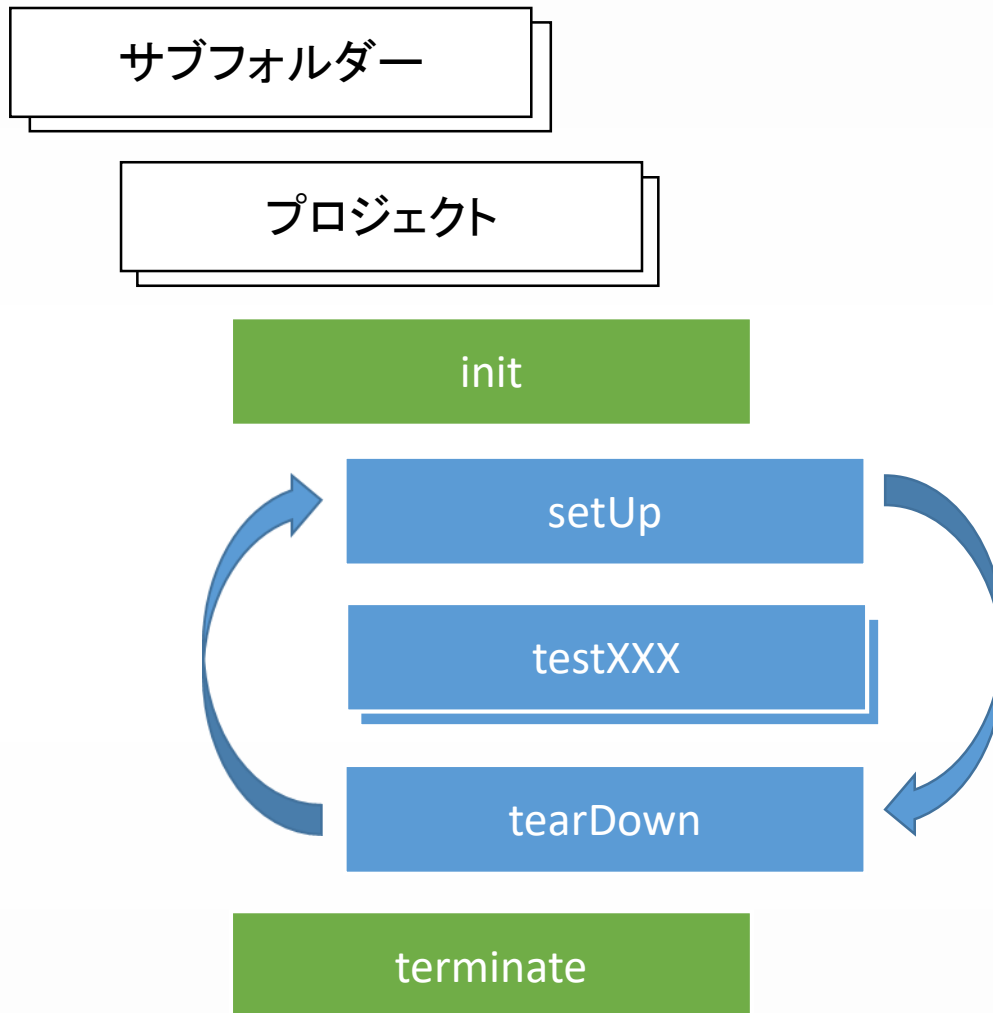
- テストフレームワークのMain.xfpのあるフォルダーにサブフォルダーを作成
- サブフォルダーにテスト対象のプロジェクトを作成



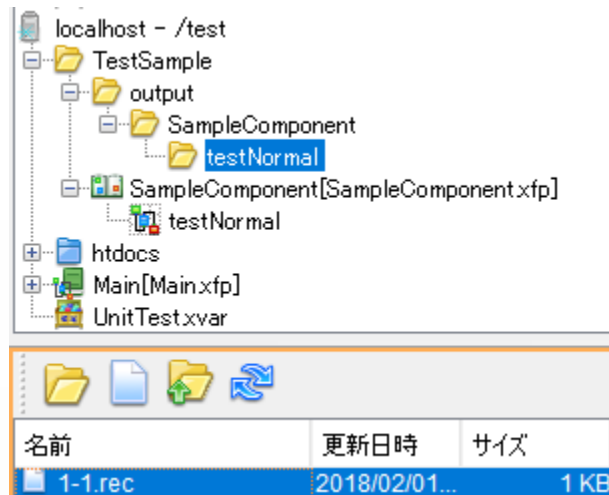
- テスト対象のフローはtestで始まる名前を付けます
 - testOneRecord
 - testTwoRecord
 - etc
- 特別なフロー

フロー名	説明
init	プロジェクトごとにテストの最初に実行されます
terminate	プロジェクトごとにテストの最後に実行されます
setUp	テスト対象のフローの実行前に実行されます
tearDown	テスト対象のフローの実行後に実行されます

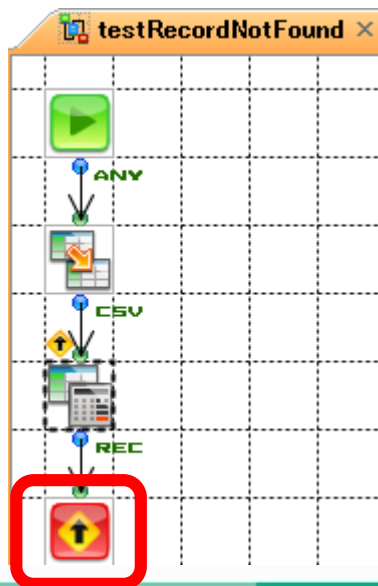




- テストフローはテスト結果であるストリームを出力します
 - 出力ストリームの型は何でも構いません
- ストリーム型に応じたファイル名で自動保存されます
 - [サブフォルダー]/**output**/[プロジェクト名]/[フロー名]/



- エラーが発生するテスト、例えば、「ファイルが存在しないエラー」などをテストしたい場合
 - フローはAbortコンポーネントで終了します
 - エラーが発生するコンポーネントのエラー処理としてMain.CreateExceptionDataフローを指定します



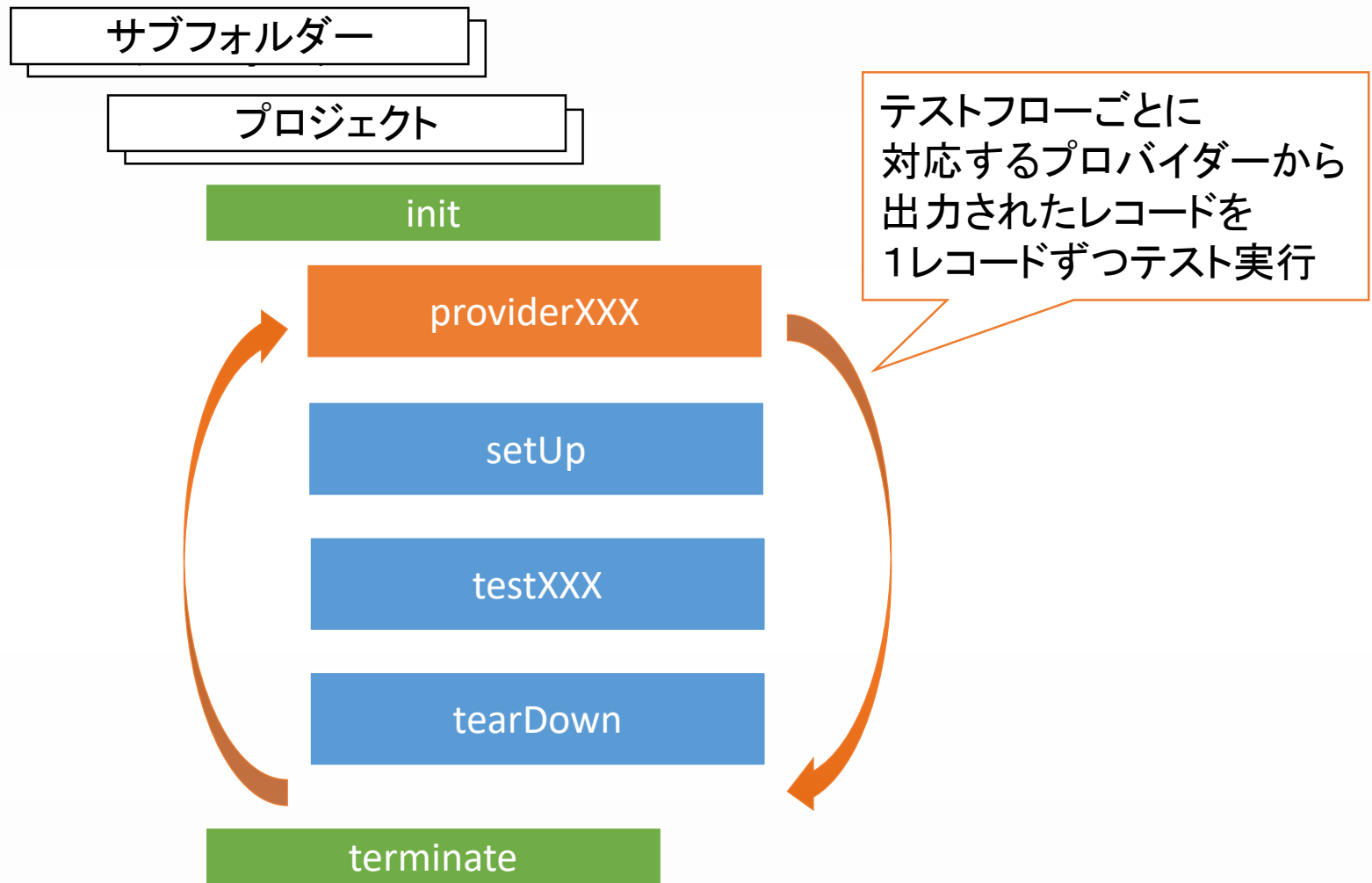
基本	キー項目	集計項目
プロパティ名	値	
名前	RecordAggregate1	
有効桁数	0	
小数点の扱い	四捨五入	
NULLを無視	はい	
空文字を無視	はい	
値の変換に失敗した場合の...	0にする	
エラー処理		
汎用	(なし)	
レコードが無い	Main.CreateExceptionData	



- 同じフローでプロパティやフロー変数の値などを変更してテストしたい場合はデータプロバイダーを使用します
- データプロバイダーには2種類あります
 - フローでデータを供給する
 - CSVファイルでデータを供給する
- テストフローに対応してデータプロバイダーを作成します
 - フローのプロバイダー：provider[テストフロー名]
 - CSVファイルのプロバイダー：provider[テストフロー名].csv

例) testOneRecord → providerOneRecord.csv





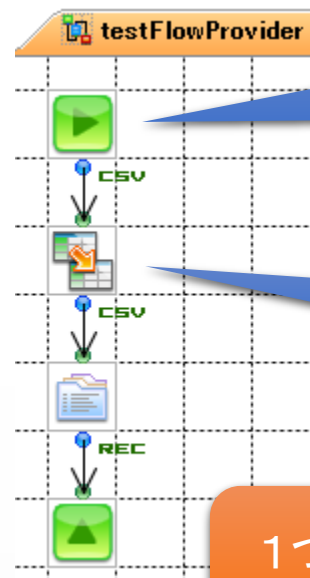
- フローによるデータプロバイダー

プロバイダーフロー



RecordかCSVを
出力します

テストフロー



プロバイダーフローの
出力が1レコードずつ
入ってきます

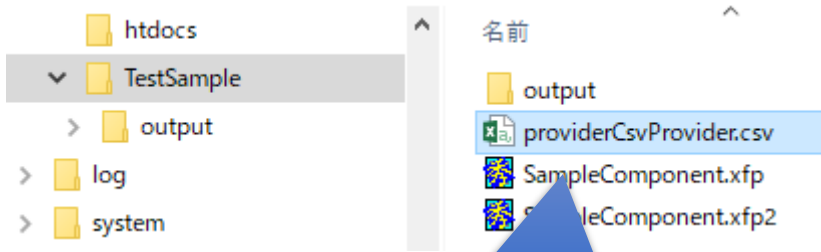
コンポーネントのプロ
パティにマッピング

1つのフローでプロパティの
値の組み合わせのテスト



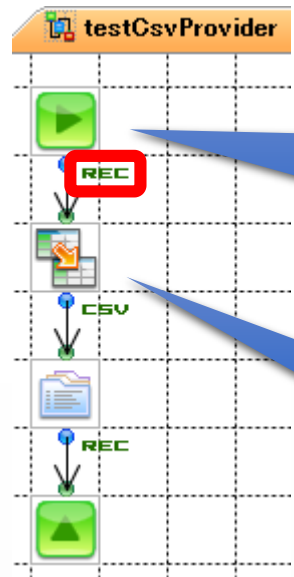
- CSVファイルによるデータプロバイダー

CSVファイル



CSVファイルに値を保存

テストフロー



CSVファイルの1行ずつがRecordに変換されて入ってきます

コンポーネントのプロパティにマッピング



- コンポーネントを実行して結果のストリームを出力
 - プロパティの組み合わせ
 - 入出カストリームの型
- 実行結果を一時保存しそれを取得する
 - フロー変数、ファイル、RDBに保存
- 別ユーザーのフローを実行して結果を取得する
 - フローサービスAPIやflow-ctrlで別ユーザーのフロー実行
 - RDBやWEBサービスの結果を取得
 - 出力されたファイルをoutputフォルダーにコピー
- エラーケース



- テスト結果が正しいかどうかはファイルを比較して判定します
 - 単純なファイル比較のみ
- 比較対象となる正しいテスト結果のファイルは次のフォルダーに置きます
 - [サブフォルダー]/**original**/[プロジェクト名]/[フロー名]/

テストフローが完成したら**output**にある出力ファイルを
originalにコピーすればOK



- テスト結果は次の2種類のログに出力されます
 - test.log – TSVファイル
 - diff.log – テキストファイル
- test.logには各テストの実行情報が出力されます

```
2018/02/02 19:04:47.912 . 1 TestSample SampleComponent testNormal 1
2018/02/02 19:04:47.945 F 3 TestSample SampleComponent testFlowProvider 1 *,All
```

カラム	説明	カラム	説明
1	実行日時	6	テストフロー名
2	結果 .: 成功、F: 比較エラー、E: 実行エラー	7	テスト番号(データプロバイダーを使用した場合のレコード番号)
3	実行時間(ms)	8	テストデータ(データプロバイダーを使用した場合の入力データ)
4	フォルダー名	9	エラーメッセージ
5	プロジェクト名		



- diff.logには結果比較での差分が出力されます

```
No original file. : C:/asteriahome/home/test/ExifGet/original/ExifGet/testWildcard/1-1.rec
```

```
=====
```

```
original:: 1:
```

```
FileName:String:output,FilePath:String:C:\asteriahome\home\test\TestSample\output,File
```

```
Date:DateTime:2018-02-02T19:01:28.726 JST,FileSize:Integer:0,FileType:String:directory
```

```
-----
```

```
revised :: 1:
```

```
FileName:String:original,FilePath:String:C:\asteriahome\home\test\TestSample\original,File
```

```
eDate:DateTime:2018-02-02T19:04:41.474 JST,FileSize:Integer:0,FileType:String:directory
```

```
revised :: 2:
```

```
FileName:String:output,FilePath:String:C:\asteriahome\home\test\TestSample\output,File
```

```
Date:DateTime:2018-02-02T19:04:47.889 JST,FileSize:Integer:0,FileType:String:directory
```

```
=====
```

比較元のファイルが存在しない

OriginalとOutputの出力が違う



ログで重要なこと

- test.logの2カラム目がすべて「.」
- diff.logが空のファイル



上記以外はテスト失敗！



- テストフローをどのように書くかが重要
 - 毎回実行結果が同じになるようにする
- コンポーネントの作成時にはプロパティはすべてマッピング可能にしておく
 - コンポーネントの直前にマッパーを置いてすべてのプロパティに値をマッピングすると、データプロバイダーを使ってプロパティの値を変更することが簡単になる
 - booleanやchoiceのプロパティでもマッピングは可能
 - コンポーネントヘルプに選択肢の値が書いてあります



- 正規表現を使った比較をしたい
- テスト結果を整形したい
- 通知方法を変更したい

テストフレームワークはフローなので
簡単に機能追加や変更ができます

