

BC-02 ファイル操作機能

■ 概要

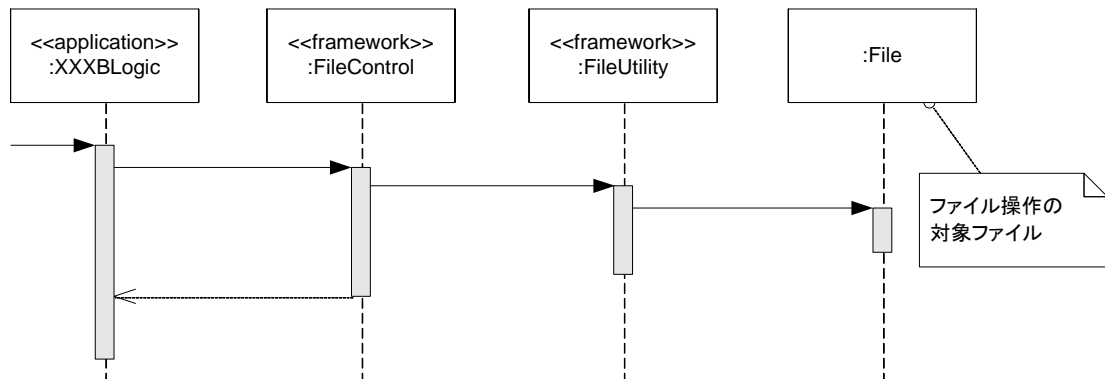
◆ 機能概要

- ファイル操作機能は以下の機能を提供する。
 - ファイル名変更
変更元ファイルと変更先のファイルを指定して、ファイル名の変更を行う。
 - ファイル移動
ファイルの移動は「ファイル名変更」を利用し、パスを変更することにより実現する。
 - ファイルコピー
コピー元ファイルとコピー先ファイルを指定して、ファイルのコピーを行う。
 - ファイル削除
削除するファイルを指定して、ファイルの削除を行う。
 - ファイル結合
指定されたファイル名のリストにあるファイルを結合する。

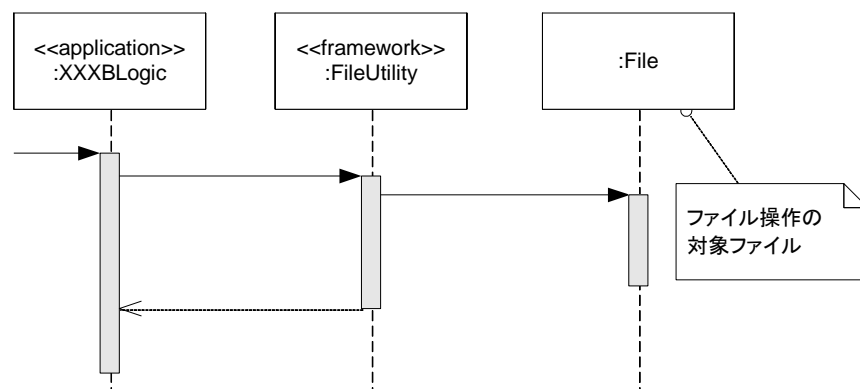
◆ 概念図

ファイル操作の概念図。

- ビジネスロジックからファイル操作インタフェースを利用して、ファイル操作を行なう場合



- ビジネスロジックからファイル操作ユーティリティクラスを直接利用して、ファイル操作を行なう場合



◆ 解説

● FileControl

ファイル操作の処理を提供するインタフェース。インタフェースが提供するメソッドの詳細についてはフレームワークの **JavaDoc** を参照のこと。ここでは、**FileControl** インタフェースで提供するファイル操作の代表的なメソッドを下記に挙げる。

➤ ファイル操作インタフェース

項番	インタフェース名	概要
1	jp.terasoluna.fw.file.util.FileControl	ファイル操作用のインタフェース

➤ ファイル操作インタフェースで提供する代表的なメソッド

項番	メソッド	概要
1	renameFile(String scrFile, String newFile)	ファイル名の変更・ファイルの移動
2	copyFile(String scrFile, String newFile)	ファイルのコピー
3	deleteFile(String scrFile)	ファイルの削除
4	mergeFile(List<String> fileList, String newFile)	fileList にあるファイルの結合

● FileControlImpl

FileControl インタフェースを実装するクラス。**FileControlImpl** クラスはビジネスロジックに **DI** して利用することができる。**FileControlImpl** クラスはファイル操作処理を実行する **FileUtility** クラスをラップしている。

FileControlImpl クラスは、属性にファイル操作を行う際に基準となるパス（基準パス）を持つ。基準パスを使用することにより、ファイルアクセス時に発生するファイルパスの環境依存の問題を回避することができる。ファイル操作を行うメソッドの引数には相対パス、または、絶対パスを設定する。ファイル操作時にエラーが発生した場合、非検査例外をスローする。

➤ 基準パスはファイル操作機能を使う上での基準となる位置を指す。基準パスを「/si1/」、相対パスを「chohyo/test.txt」とした場合、ファイルの絶対パスは「/si1/chohyo/test.txt」となる。

- FileUtility

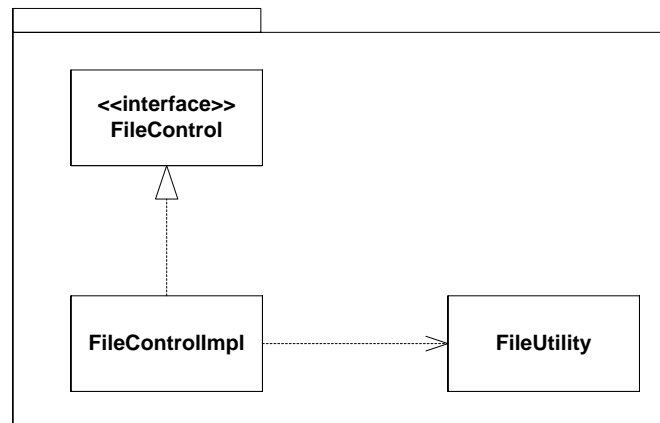
ファイル操作機能を実装するクラス。FileUtility のメソッドをビジネスロジックから直接利用することも可能である。FileUtility クラスで提供するメソッドは FileControl インタフェースで提供するものと同じになる。FileUtility クラスではファイル操作を行うメソッドの引数は絶対パスのみ設定可能である。ファイル操作時にエラーが発生した場合、非検査例外をスローする。

- ファイル操作ユーティリティクラス

項番	メソッド	概要
1	jp.terasoluna.fw.file.util.FileUtility	ファイル操作ユーティリティクラス

- ファイル操作機能のクラス図

- FileControl、FileControlImpl、FileUtility クラスの関連は下記のクラス図のとおりである。



■ 使用方法

◆ コーディングポイント

- ファイル操作クラス（FileControl）を利用する例

➤ Bean 定義ファイルの設定例

```

.....
<bean id="fileControl"
      class="jp.terasoluna.fw.file.util.FileControlImpl">
  <property name="basePath" value="${basepath}" />
  <property name="checkFileExist" value="false" />
</bean>
.....
<bean id="sampleBLogic"
      class="jp.terasoluna.batch.sample.SampleBLogic">
  .....
  <property name="fileControl" ref="fileControl" />
  .....
.....

```

FileControl インタフェースを実装するクラスをフレームワーク Bean 定義ファイルに定義する。プロパティに基準パスを設定すること。

操作後にできるファイルパスにファイルが存在する場合、処理を継続する(上書きする: true)か例外を投げて停止する(false)かを定めるフラグ。

フレームワーク Bean 定義ファイルで設定した Bean を参照する

➤ ビジネスロジックの実装例（ファイルのコピー、移動、削除処理の実装例）

```

.....
private FileControl fileControl = null;
public void setFileControl(FileControl fileControl){
  this.fileControl = fileControl;
}
.....
// ファイルのコピー（相対パスを設定する例）
// /sil/chohyo/test.txt を/sil/chohyo/testFile.txt にコピー。
// 基準パスは「/sil/」
fileControl.copyFile("chohyo/test.txt", "chohyo/testFile.txt");
.....
// ファイルの移動（相対パスを設定する例）
// /sil/chohyo/testFile.txt を/sil/output/testFile.txt に移動。
// 基準パスは「/sil/」
fileControl.renameFile("chohyo/testFile.txt", "output/testFile.txt");
.....
// ファイルの削除（相対パスを設定する例）
// /sil/chohyo/testFile.txt を削除。
// 基準パスは「/sil/」
fileControl.deleteFile("chohyo/testFile.txt");
.....
// ファイルのコピー（絶対パスを設定する例）
// /sil/chohyo/test.txt を/sil/chohyo/testFile.txt にコピー。
fileControl.copyFile("/sil/chohyo/test.txt", "/sil/chohyo/testFile.txt");
.....

```

ファイル操作機能を利用するクラスは、FileControl インタフェースとその setter が必須

各メソッドの引数はファイルの相対パス、もしくは絶対パスを記述する

➤ ビジネスロジックの実装例（ファイル結合の実装例）

```
.....
// ファイルの結合。
// 以下に挙げるファイルをリストに格納し、ファイルを/sil/output/mergeFile.csvに統合。
// /sil/chohyo/output001.csv
// /sil/chohyo/output002.csv
// /sil/chohyo/output003.csv
// 基準パスは「/sil/」
fileList.add("chohyo/output001.csv");
fileList.add("chohyo/output002.csv");
fileList.add("chohyo/output003.csv");
.....
fileControl.mergeFile(fileList, "output/mergeFile.csv");
.....
```

メソッドの2番目の引数はファイルの相対パス、もしくは絶対パスを記述する

● ファイル操作ユーティリティクラス（FileUtility）を直接利用する例

➤ ビジネスロジックの実装例（ファイルのコピー、移動、削除処理の実装例）

```
.....
// ファイルのコピー。
// /sil/chohyo/test.txt を/sil/chohyo/testFile.txtにコピー。
FileUtility.copyFile("/sil/chohyo/test.txt", "/sil/chohyo/testFile.txt");
.....
// ファイルの移動。
// /sil/chohyo/testFile.txt を/sil/output/testFile.txtに移動。
FileUtility.renameFile("/sil/chohyo/testFile.txt",
"/sil/output/testFile.txt");
.....
//ファイルの削除。 /sil/chohyo/testFile.txt を削除。
FileUtility.deleteFile("/sil/chohyo/testFile.txt");
.....
```

各メソッドの引数はファイルの絶対パスを記述する

◆ 拡張ポイント

- なし。

■ 関連機能

- なし。

■ 使用例

- 機能網羅サンプル(functionsample-batch)
- チュートリアル(tutorial-batch)

■ 備考

- なし。