

BL-08 ファイル操作機能

■ 概要

◆ 機能概要

- TERASOLUNA Batch Framework for Java ver 3.x で使用するファイル操作機能は、TERASOLUNA Batch Framework for Java ver 2.x で使用していたファイル操作機能と同一のものを利用して、ファイル操作を行う。
- 本項目では、TERASOLUNA Batch Framework for Java ver 3.x でファイル操作機能を使用する場合の TERASOLUNA Batch Framework for Java ver 2.x との違いのみを説明するものとし、ファイル操作機能の詳細な説明は別資料の「BC-02 ファイル操作機能」の機能説明書を参照すること。

◆ コーディングポイント

- 本説明書でのコーディングポイントは、別資料の「BC-02 ファイル操作機能」のコーディングポイントと異なる以下の項目についてのみ説明を行う。
 - ・ ファイル操作クラス（FileControl）を利用する例その他の項目については、別資料の「BC-02 ファイル操作機能」を参照すること。
- 次ページからのコーディングポイントの中で、TERASOLUNA Batch Framework for Java ver 3.x で本機能を使用する際に大事なポイントについては、二重線の吹き出しを使用して強調している。

【凡例】

```
...  
@Autowired  
protected FileControl fileControl = null;  
...
```

TERASOLUNA Batch Framework for Java
ver 3.x での大事なポイント。

- ファイル操作クラス (FileControl) を利用する例

- Bean 定義ファイル(commonContext.xml)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>
.....

```

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。

- 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)

```

@Component
public class Sample001BLogic implements BLogic {
  .....

  @Autowired
  protected FileControl fileControl = null;
  (…中略…)

  public int execute(BLogicParam arg0) {
    // ファイルのコピー (相対パスを設定する例)
    // /si1/chohyo/testtxt を/si1/chohyo/testFile.txt にコピー。
    // 基準パスは「/si1/」
    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)
    // /si1/chohyo/testFile.txt を/si1/output/testFile.txt に移動。
    // 基準パスは「/si1/」
    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)
    // /si1/chohyo/testFile.txt を削除。
    //基準パスは「/si1/」
    fileControl.deleteFile("chohyo/testFile.txt");
    .....
    // ファイルのコピー (絶対パスを設定する例)
    // /si1/chohyo/test.txt を/si1/chohyo/testFile.txt にコピー。
    fileControl.copyFile("/si1/chohyo/test.txt", "/si1/chohyo/testFile.txt");
    .....
  }
  (…以下略…)
}

```

ファイル操作機能を利用するビジネスロジックは、@Autowired を使用して FileControl インタフェース実装クラスをビジネスロジックに DI する。

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  
// /si1/output/mergeFile.csv に統合。  
// /si1/chohyo/output001.csv  
// /si1/chohyo/output002.csv  
// /si1/chohyo/output003.csv  
// 基準パスは「/si1/」  
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。  
// /si1/chohyo/test.txt を/si1/chohyo/testFile.txt にコピー。  
FileUtility.copyFile("si1/chohyo/test.txt", "/si1/chohyo/testFile.txt");  
.....  
// ファイルの移動。  
// /si1/chohyo/testFile.txt を/si1/output/testFile.txt に移動。  
FileUtility.renameFile("si1/chohyo/testFile.txt",  
    "/si1/output/testFile.txt");  
.....  
//ファイルの削除。 /si1/chohyo/testFile.txt を削除。  
FileUtility.deleteFile("si1/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。