



ドキュメント名	ID
ParkDiff 機能仕様書	P K

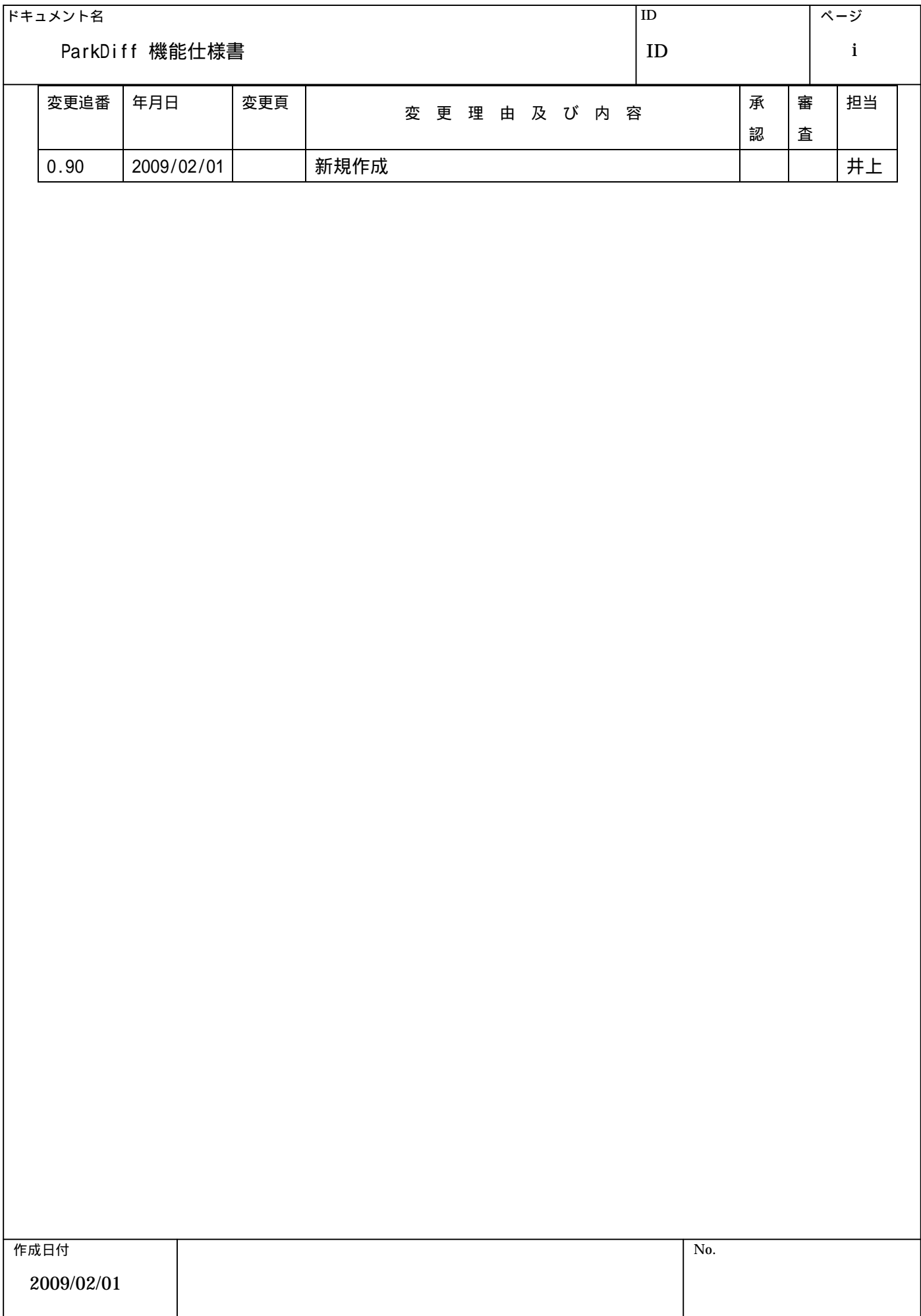
パーク製 Diff ライブラリ

# ParkDiff 機能仕様書

株式会社パーク  
アプリケーション開発部

承認	審査	作成

作成日付	No.
2009/02/01	





ドキュメント名	ID	ページ
ParkDiff 機能仕様書	ID	i

## 目次

1. はじめに .....	1
1.1. 目的 .....	1
1.2. 目的と主な機能 .....	1
1.3. 動作条件 .....	1
2. 機能概要 .....	2
2.1. 文字列差異抽出 .....	2
3. 機能詳細 .....	2
3.1. 文字単位差異抽出機能 .....	2
3.2. 行単位差異抽出機能 .....	2
4. 提供インターフェース .....	3
4.1. 提供メソッド .....	3
4.2. 提供差異情報詳細 .....	3
5. サンプルコード .....	4
5.1. 比較文字列 .....	4
5.2. サンプルコード .....	4
5.3. 出力結果 .....	6
5.4. 出力結果 加工例 .....	7
6. 動作保証範囲 .....	8
7. 試用版の機能制限 .....	8
8. トラブルシューティング .....	9
8.1. OutOfMemoryError が発生する .....	9
8.2. ClassNotFoundException が発生する .....	9
8.3. ライセンス認証に失敗する .....	9
9. 著作権 .....	10
9.1. 著作権について .....	10

作成日付	No.
2009/02/01	



ドキュメント名	ID	ページ
ParkDiff 機能仕様書	ID	1

## 1. はじめに

### 1.1. 目的

本ドキュメントは株式会社パーク製のテキスト差異抽出 Java ライブラリ「ParkDiff」の機能仕様を記したものです。

### 1.2. 目的と主な機能

与えられたふたつの文字列を比較し、異なる部分を検出します。

一般的な文字列差異比較の機能を Java のクラスライブラリとして提供することにより、軽量でシンプルな構成、既存システムへの容易な導入・応用を可能にします。

### 1.3. 動作条件

Java Platform Standard Edition(Java 2SE)5 (Update 17 以降)が動作する環境。

作成日付		No.
2009/02/01		



ドキュメント名	ID	ページ
ParkDiff 機能仕様書	ID	2

## 2. 機能概要

### 2.1. 文字列差異抽出

引数で渡されたふたつの文字列の差異を検出します。

差異情報には、ある差異の 開始位置/終了位置/文字数 が含まれます。

結果として差異の個数分の要素を持つ配列が提供されます。

## 3. 機能詳細

### 3.1. 文字単位差異抽出機能

引数で渡されたふたつの文字列の差異を検出し、文字単位での差異情報を提供します。

### 3.2. 行単位差異抽出機能

引数で渡されたふたつの文字列の差異を検出し、行単位での差異情報を提供します。

作成日付		No.
2009/02/01		

ドキュメント名	ID	ページ
ParkDiff 機能仕様書	ID	3

## 4. 提供インターフェース

以下のインターフェースを提供します。

### 4.1. 提供メソッド

以下のメソッドを提供します。

修飾子	返回值	メソッド名	引数	動作内容
public	DiffResult[]	DiffLine	String orgLine, String diffLine	行単位での差分を抽出。
public	DiffResult[]	DiffChar	String orgText, String diffText	文字単位での差分を抽出。
public	boolean	isLicensed		ライセンス有無を返す。

### 4.2. 提供差異情報詳細

上記メソッドで返される DiffResult[] の要素詳細です。

修飾子	型	プロパティ名	詳細
public	boolean	Modified	追加 / 修正 / 削除が行われたかを示す。
public	int	OriginalStart	比較元における変更開始位置 ( 1 )
public	int	OriginalEnd	比較元における変更終了位置 ( 1 )
public	int	OriginalLength	比較元における変更された文字列の長さ ( 2 )。 修正 / 削除が行われた場合のみ値が入る。 比較先に追加されたただけの場合には 0 が入る。
public	int	ModifiedStart	比較先における変更開始位置 ( 1 )
public	int	ModifiedEnd	比較先における変更終了位置 ( 1 )
public	int	ModifiedLength	比較先における変更した文字列の長さ ( 2 )。 修正 / 追加が行われた場合にのみ値が入る。 比較元から削除された場合には 0 が入る。

- ( 1 ) 行差異抽出時の単位は行(先頭から見て xxx 行目の意)、  
文字差異抽出時は単位は文字数(先頭から見て xxx 文字目の意)となります。
- ( 2 ) 行差異抽出時の単位は行(xxx 行変更された、の意)、  
文字差異抽出時は単位は文字数(xxx 文字変更された、の意)となります。

作成日付	No.
2009/02/01	

ドキュメント名	ID	ページ
ParkDiff 機能仕様書	ID	4

## 5. サンプルコード

### 5.1. 比較文字列

行数	比較元	比較先
0	あああ	あああ
1	いいい	かかか
2	ううう	ううう
3	えええ	えきえ
4	おおお	くくく

### 5.2. サンプルコード

```
public static void main(String[] args) {

    textA = "あああ\r\nいいい\r\nううう\r\nえええ\r\nおおお";
    textB = "あああ\r\nかかか\r\nううう\r\nえきえ\r\nくくく";

    //文字分抽出
    Diff diff = new Diff();
    DiffResult[] ret = diff.DiffChar(textA, textB);
    System.out.println("Diff Char -----");
    System.out.println("Result Num = " + ret.length);
    DiffResult result = null;
    for(int i = 0; i < ret.length; i++) {
        result = ret[i];
        System.out.println("  result[" + i + "]");
        System.out.println("      Modified   : " + result.Modified);
        System.out.println("      ModifiedLength : " + result.ModifiedLength);
        System.out.println("      ModifiedStart : " + result.ModifiedStart);
        System.out.println("      OriginalLength : " + result.OriginalLength);
        System.out.println("      OriginalStart  : " + result.OriginalStart);
    }

    //行差分抽出
    Diff diffLine = new Diff();
    DiffResult[] retLine = diffLine.DiffLine(textA, textB);
```

作成日付	No.
2009/02/01	



ドキュメント名 ParkDiff 機能仕様書		ID ID	ページ 5
<pre>System.out.println("Diff Line -----"); System.out.println("Result Num = " + retLine.length); DiffResult resultLine = null; for(int j = 0; j &lt; retLine.length; j++) {     resultLine = retLine [j];     System.out.println("  result[" + j + "]");     System.out.println("    Modified   : " + resultLine.Modified);     System.out.println("    ModifiedLength : " + resultLine.ModifiedLength);     System.out.println("    ModifiedStart  : " + resultLine.ModifiedStart);     System.out.println("    OriginalLength : " + resultLine.OriginalLength);     System.out.println("    OriginalStart  : " + resultLine.OriginalStart); } }</pre>			
作成日付 2009/02/01		No.	





ドキュメント名	ID	ページ
ParkDiff 機能仕様書	ID	6

### 5.3. 出力結果

Diff Char -----

Result Num = 4

result[0]

Modified : true

ModifiedLength : 3

ModifiedStart : 5

OriginalLength : 3

OriginalStart : 5

result[1]

Modified : true

ModifiedLength : 1

ModifiedStart : 16

OriginalLength : 0

OriginalStart : 16

result[2]

Modified : true

ModifiedLength : 0

ModifiedStart : 18

OriginalLength : 1

OriginalStart : 17

result[3]

Modified : true

ModifiedLength : 3

ModifiedStart : 20

OriginalLength : 3

OriginalStart : 20

Diff Line -----

Result Num = 2

result[0]

Modified : true

ModifiedLength : 1

ModifiedStart : 1

OriginalLength : 1

作成日付	No.
2009/02/01	

ドキュメント名	ID	ページ
ParkDiff 機能仕様書	ID	7

```

OriginalStart : 1
result[1]
  Modified      : true
  ModifiedLength : 2
  ModifiedStart  : 3
  OriginalLength : 2
  OriginalStart  : 3

```

#### 5.4. 出力結果 加工例

上記結果を加工した一例です。

##### [行差異]

●変更前:	●変更後:
あああ	あああ
い	か
う	う
え	え
お	<<<

##### [文字差異]

●変更前:	●変更後:
あああ	あああ
い	か
う	う
え	え
お	<<<お

作成日付	No.
2009/02/01	

ドキュメント名	ID	ページ
ParkDiff 機能仕様書	ID	8

## 6. 動作保証範囲

動作保証範囲は下記の通りです。

項目	制限事項
文字数	比較元、比較先ともに各 1,000,000 文字以下
差異検出数	20,000 差異以下 一度に返される DiffResult[] の要素数と同義です。

上記動作保証範囲を超えての使用時に発生した不具合等はサポートの対象外とさせていただきます。

## 7. 試用版の機能制限

ライセンスご購入前の本製品は試用版として動作します。

試用版は下記の通り機能が制限されます。

項目	制限事項
文字数	比較元、比較先ともに各 256 文字以下
差異検出数	50 差異以下 一度に返される DiffResult[] の要素数と同義です。

作成日付		No.
2009/02/01		

ドキュメント名	ID	ページ
ParkDiff 機能仕様書	ID	9

## 8. トラブルシューティング

予想されるトラブルとその対策について記述します。

### 8.1. OutOfMemoryError が発生する

比較する文字列が長すぎる場合などに発生する恐れがあります。

文字列を短く区切ることで回避できます。

もしくは、JavaVM のヒープメモリサイズを引き上げることで回避可能な場合があります。

実行時、JavaVM 引数に `-Xmx000m` (000 は指定するヒープサイズとして下さい。単位はメガバイトです)を加えて実行して下さい。

例)

ヒープサイズを最大 128MG とする場合、下記のようにして実行します。

`$ Java -Xmx128m <実行するクラス名>`

### 8.2. ClassNotFoundException が発生する

クラスファイルが見つからない場合に発生します。以下の点をご確認下さい。

- ParkDiff.jar が存在し、且つ正しくパスが通っている。
- LicenseKey.jar が存在し、且つ正しくパスが通っている。
  - ✧ LicenseKey.jar は試用版でも必要となります。
  - ✧ 試用版ダウンロード時に含まれていた LicenseKey.jar をご使用下さい。

### 8.3. ライセンス認証に失敗する

PdLicenseException が発生する、もしくはライセンスご購入後も機能制限が解除されない場合、以下の点をご確認下さい。

- ライセンス購入後の LicenseKey.jar (試用版ダウンロード時に含まれたものとは別物)が存在し、且つ正しくパスが通っている。
  - ✧ 試用版に含まれた LicenseKey.jar を参照している場合、ライセンスご購入後にお送りする LicenseKey.jar と入れ替えて下さい。
- メソッド isLicensed を実行し、true が返ることを確認して下さい。

作成日付		No.
2009/02/01		



ドキュメント名	ID	ページ
ParkDiff 機能仕様書	ID	10

## 9. 著作権

### 9.1. 著作権について

本仕様書の著作権は株式会社パークに帰属します。

本仕様書に記載されている内容(サンプルコードを除く)の一部または全てを、著作権者の許可なしに複製、転載、改変することを禁じます。

作成日付		No.
2009/02/01		