

Java用データ圧縮ライブラリ

# RELC for Java

(J2SE 1.3.1 以上 / DoJa 3.0 以上)

ユーザーズマニュアル

01版

RELC for Java の著作権は富士通デバイス株式会社が保有しています。

Sun、Sun Microsystems、Java、J2SE は、米国およびその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。

「i アプリ/アイアプリ」, 「iαpli/アイアプリ」, 「DoJa」は株式会社 NTT ドコモの商標または登録商標です。

1. 本資料に記載した製品および製品の仕様につきましては、製品改善のため予告なしに変更することがあります。したがって、ご使用を検討の際は、本資料に記載の情報が最新のものであることを弊社技術担当、あるいは弊社営業担当にご確認ください。
2. 本資料に記載された情報・回路図は、当社製品の応用例として使用されており、実際に使用する機器への搭載を目的としたものではありません。また、これらの情報・回路図の使用に起因する第三者の特許権、その他権利侵害について、当社はその責任を負いません。
3. 本資料に記載された製品は、一般事務用、パーソナル用、家庭用、通常の産業用等の一般的用途を想定して設計・製造されているものであり、原子力施設における核反応制御、航空機自動飛行制御、航空交通管制、大量輸送システムにおける運行制御、生命維持のための医療用機器、兵器システムにおけるミサイル発射制御など、極めて高度な安全性が要求され、仮に当該安全性が確保されない場合、直接生命・身体に対する重大な危険性を伴う用途（以下「ハイセイフティ用途」という）に使用されるよう設計・製造されたものではございません。お客様は、当該ハイセイフティ用途に要する安全性を確保する措置を施すことなく、本製品を使用しないでください。ハイセイフティ用途に使用される場合は、当社担当営業までご相談ください。
4. 本資料に記載した内容を、弊社に無断で転載または複製することはご遠慮ください。
5. 本資料に記載された製品が、「外国為替および外国貿易法」に基づき規制されている貨物または技術に該当する場合には、本製品を輸出するに際して、同法に基づく許可が必要となります。

## 目次

1	概要.....	1
2	製品の構成 .....	1
3	クラスライブラリ作成環境 .....	1
4	動作確認環境.....	1
5	クラスライブラリを組み込む際の注意事項.....	1
5.1	i アプリ作成上の注意 .....	1
6	クラスライブラリ概要 .....	2
6.1	RelcInputStream クラスの概要 .....	3
6.1.1	フィールド説明 .....	3
6.1.2	メソッド概要 .....	3
6.2	RelcInputStreamNullBlock クラスの概要 .....	4
6.2.1	フィールド説明 .....	4
6.2.2	メソッド概要 .....	4
6.3	RelcOutputStream クラスの概要 .....	5
6.3.1	フィールド説明 .....	5
6.3.2	メソッド概要 .....	5
7	RELC for Java での独自例外 .....	6
8	使用例 .....	7
8.1	圧縮処理 .....	7
8.2	復元処理 .....	9
9	PC 版評価プログラムについて.....	11

## 1 概要

本製品 (RELC for Java) は、可逆圧縮方式 (ロスレス方式) を採用したデータ圧縮ライブラリです。

## 2 製品の構成

次のクラスライブラリが同封されています。

• Relc.jar	J2SE 用ライブラリ
• DoJa¥com¥fdi¥fujitsu¥com¥Relc¥*.class	DoJa用ライブラリ
• doc¥index.html	API リファレンス (javadoc で作成)
• RelcEva.exe	PC 版評価プログラム

製品パッケージの全体構成については、“README.TXT” を参照してください。

## 3 クラスライブラリ作成環境

本製品の作成環境は、次のとおりです。

Java2 環境	: Java2 SDK, Standard Edition Version 1.3.1 ( Windows 版 )
DoJa環境	: icpli Development Kit for DoJa 3.0 Ver.2.00 ( 日本語版 )

## 4 動作確認環境

本製品の動作確認環境は、以下の環境です。

Java2 環境	: J2SE 1.3.1 / 1.4.2 ( Windows 2000 / XP が動作する DOS/V 機 )
DoJa環境	: DoJa 3.0 / 3.5 プロファイル対応携帯電話端末

## 5 クラスライブラリを組み込む際の注意事項

クラスライブラリを組み込む際の注意事項について説明します。

### 5.1 i アプリ作成上の注意

DoJa¥com¥fdi¥fujitsu¥com¥Relc¥\*.class は preverify.exe を使用して作成されておりますので、そのままご利用いただけます。また、Java2 用の Relc.jar をご利用になる場合は、作成されたアプリケーションに対して preverify.exe を使用して i アプリ用に変換してください。

## 6 クラスライブラリ概要

本製品には、次の3つのクラスが含まれています。

クラスの詳細については、doc 以下に格納された html ドキュメントを参照してください。

- RelcInputStream                   : 圧縮されたストリームからバイトデータを復元するクラスです。
  
- RelcInputStreamNullBlock       : RelcInputStreamの簡易仕様版です。KIND\_ZERO のみに対応しています。(DoJa用にのみ実装されています。)
  
- RelcOutputStream               : バイトデータを圧縮されたストリームに書き込むクラスです。

## 6.1 RelcInputStream クラスの概要

RelcInputStream は、圧縮されたストリームからバイトデータを復元して読み出す機能を提供します。

圧縮データの形式は、KIND\_NO\_INFO, KIND\_ZERO, KIND\_ORG\_SIZE, KIND\_CODE\_SIZE の全てに対応しています。

また、圧縮モードに関係なく、復元可能です。

### 6.1.1 フィールド説明

[フィールド名]	[説明]
VERSION	バージョン情報。

### 6.1.2 メソッド概要

以下のメソッドが通常利用するものとして準備されています。

[メソッド名]	[説明]
RelcInputStream	コンストラクタ。
read	ストリームから復元されたデータを読み出す。
skip	ストリームから復元されたデータをスキップ。
close	ストリームを閉じる。
available	内部バッファ内の残りバイト数を調べる。
getOrgSize	ストリームの復元サイズを調べる。 (KIND_ORG_SIZE のみ有効)
getCodeSize	ストリームの圧縮サイズを調べる。 (KIND_CODE_SIZE のみ有効)

## 6.2 RelcInputStreamNullBlock クラスの概要

RelcInputStreamNullBlock は、KIND\_ZERO のみに対応している RelcInputStream の簡易仕様版です。本クラスは、DoJa用にのみ実装されています。

### 6.2.1 フィールド説明

[フィールド名]	[説明]
VERSION	バージョン情報。

### 6.2.2 メソッド概要

以下のメソッドが通常利用するものとして準備されています。

[メソッド名]	[説明]
RelcInputStreamNullBlock	コンストラクタ。
read	ストリームから復元されたデータを読み出す。
skip	ストリームから復元されたデータをスキップ。
close	ストリームを閉じる。
available	内部バッファ内の残りバイト数を調べる。

### 6.3 RelcOutputStream クラスの概要

RelcOutputStreamは、バイトデータをストリームに書き込むことによって、圧縮データを作成する機能を提供します。

作成可能な圧縮データの形式は、KIND\_NO\_INFO, KIND\_ZERO, KIND\_ORGSIZE から選択可能です。また、圧縮モードとして圧縮能力を数値(0もしくは1~128)で変更することが可能です。0(MODE\_MAX)を指定すると、圧縮能力は最も高く(最高圧縮)なりますが、圧縮速度は低速です。128を指定すると圧縮能力はMODE\_MAXより若干低下しますが、圧縮速度は若干高速になります。更に小さい値(MODE\_HIGH, MODE\_MID, MODE\_LOW, または 128より小さい値)を指定すると、圧縮能力が低下する代わりに、圧縮速度は向上します。

#### 6.3.1 フィールド説明

[フィールド名]	[説明]
VERSION	バージョン情報。
KIND_NO_INFO	KIND_NO_INFO 型の圧縮データ作成の指定。
KIND_ZERO	KIND_ZERO 型の圧縮データ作成の指定。
KIND_ORGSIZE	KIND_ORGSIZE 型の圧縮データ作成の指定。
MODE_MAX	最高圧縮モード指定 (0)。
MODE_HIGH	高圧縮モード指定 (32)。
MODE_MID	中圧縮モード指定 (16)。
MODE_LOW	低圧縮モード指定 (4)。

#### 6.3.2 メソッド概要

以下のメソッドが通常利用するものとして準備されています。

[メソッド名]	[説明]
RelcOutputStream	コンストラクタ。
write	バイトデータを圧縮してストリームへ書き込む。
close	ストリームを閉じる。
closeGetSize	ストリームを閉じ、出来上がった圧縮データのサイズを得る。

## 7 RELC for Javaでの独自例外

RELC for Javaが判別する独自の例外は、全て IOException として発生します。  
独自例外の IOException には、次のメッセージが送り込まれます。

[メッセージ]	[説明]
RELC format error	圧縮データのフォーマットに異常が見つかりました。
parameter error	パラメータが異常です。
not enough buffer	指定されたバッファ配列の要素数が不足しています。
not match org-size	コンストラクタで指定されたデータサイズと圧縮に利用したデータサイズが一致していません。
can not use	サポートされていない機能が呼び出されました。
INTERNAL ERROR	想定されていない呼び出し方法が使用されました。

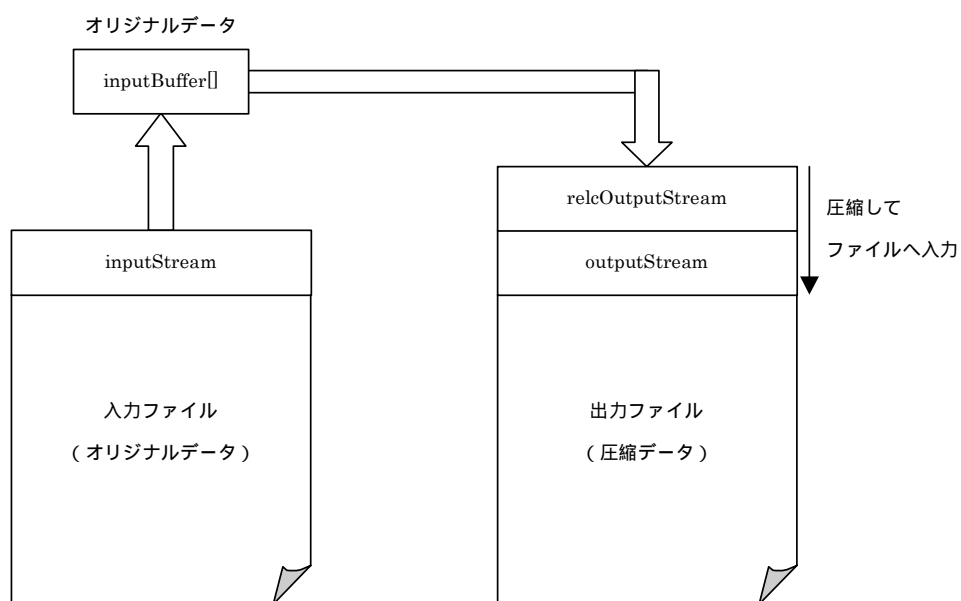
## 8 使用例

RELC for Javaのインタフェースを使用した1ファイルの圧縮処理と復元処理の例を示します。

圧縮/復元処理は1ファイル毎に行い、入力ストリームのデータが無くなるまで処理を続行します。

### 8.1 圧縮処理

オリジナルデータを格納したファイルを圧縮し、圧縮ファイルを作成する例を示します。



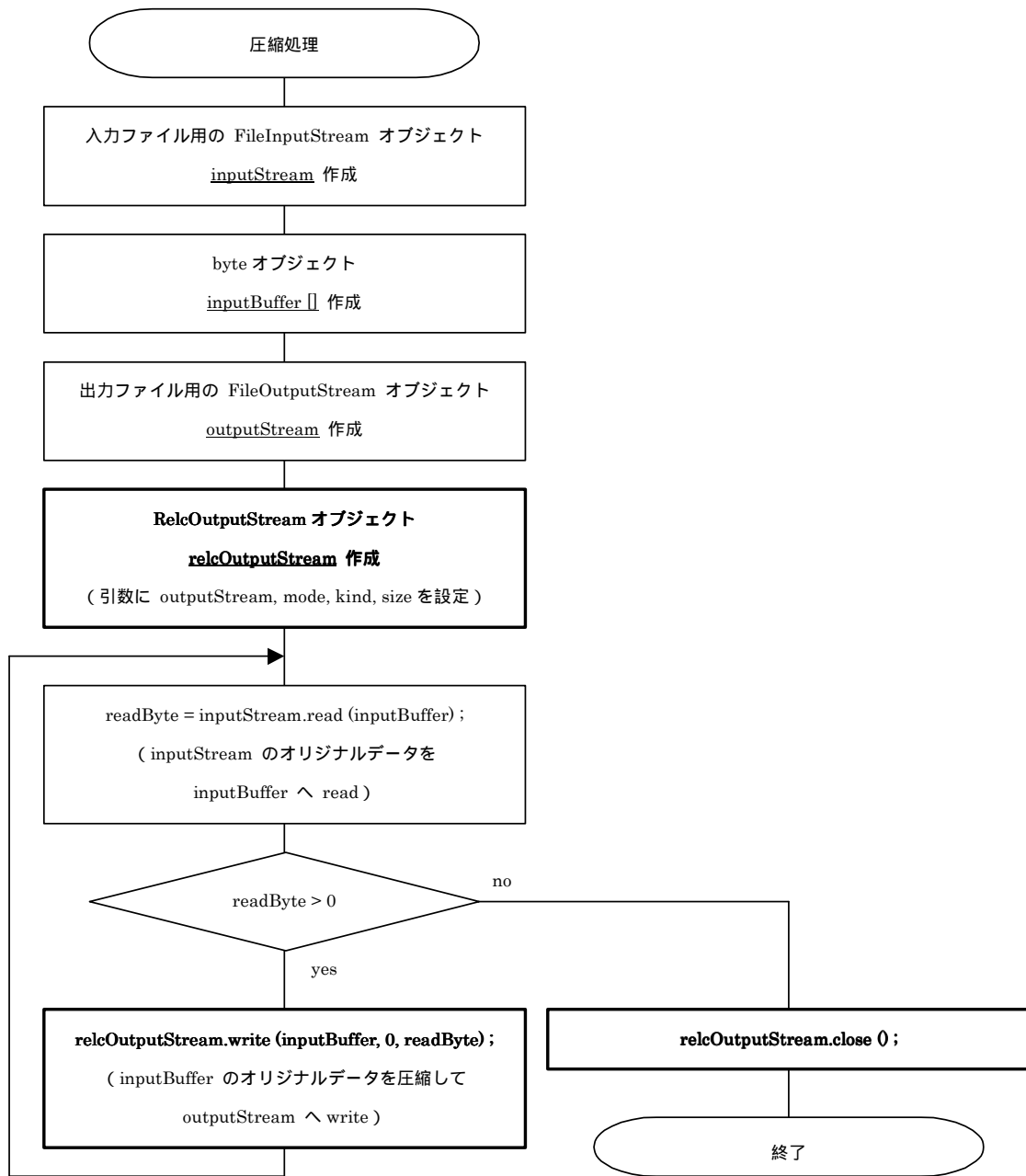
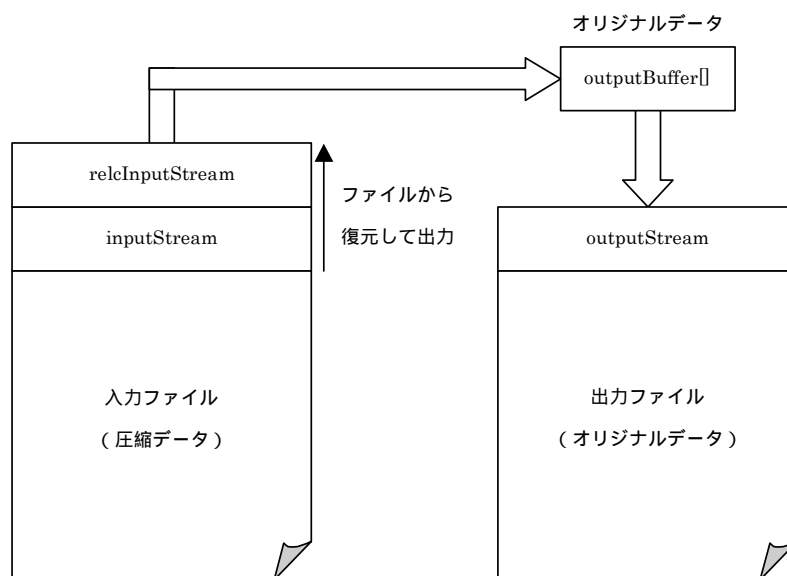


図1 圧縮処理フロー

## 8.2 復元処理

圧縮データを格納したファイルを復元し、オリジナルデータのファイルを作成する例を示します。



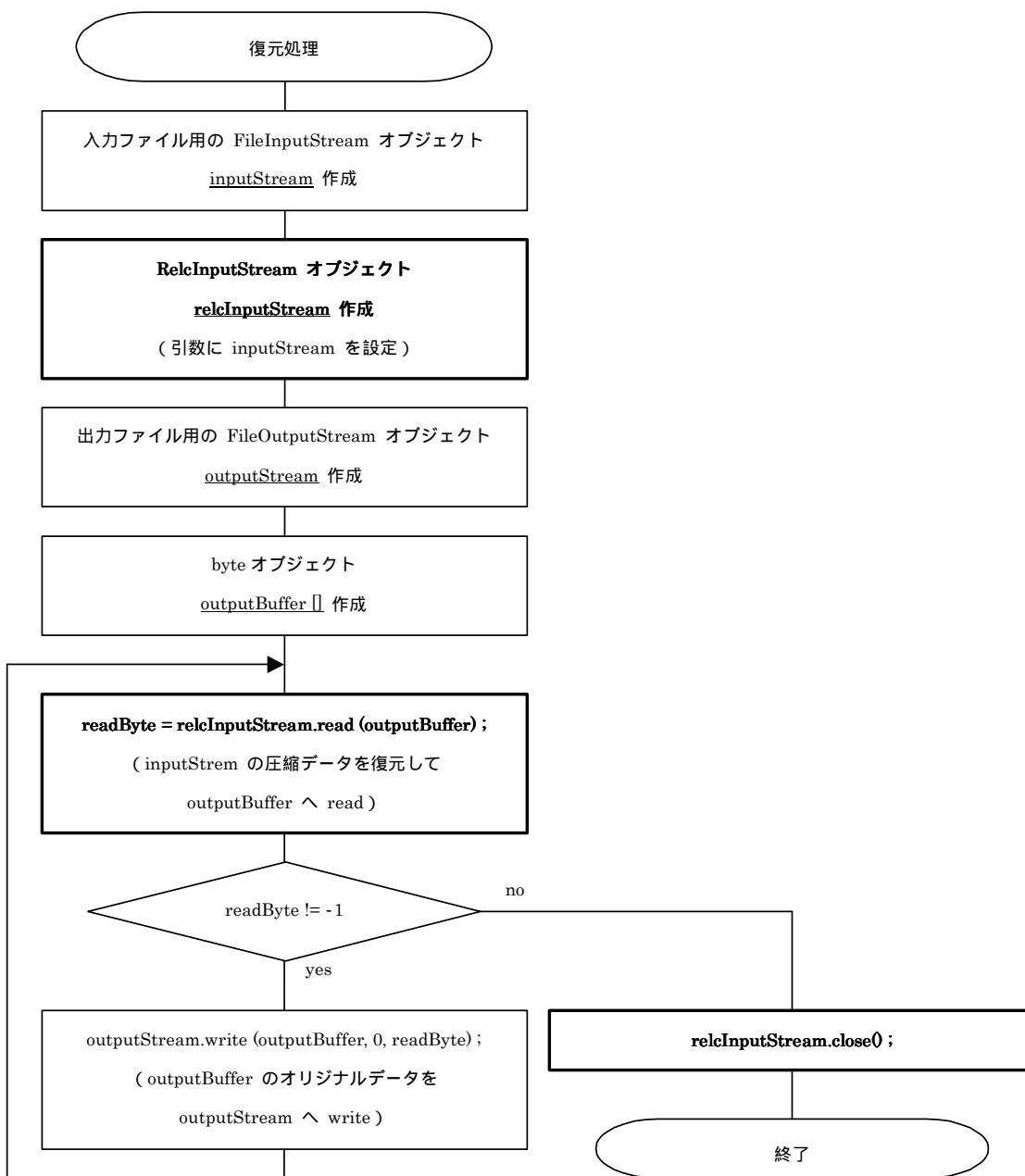


図2 復元処理フロー

## 9 PC 版評価プログラムについて

本製品には、PC 上で圧縮ファイルを作成、および、作成された圧縮ファイルを復元するための評価プログラム (RelcEva.exe) が添付されています。このプログラムは、評価用のみ利用できます。

(このプログラムを配布することはできません。システムに利用する場合は、別途ライセンス製品を購入する必要があります。)

### ・起動方法

prompt> RelcEva **option infile outfile [mode]**

option **書式** {E[N|OC]|D[N]}<sup>1</sup>

**圧縮** 圧縮処理の場合は、'E' を指定します。  
圧縮形式を指定する場合は、'E' に続けてオプション 1 を記述します。

オプション 1	説明
N	KIND_NO_INFO 形式の圧縮ファイルを作成します。
O	KIND_ORGSIZE 形式の圧縮ファイルを作成します。
C	KIND_CODESIZE 形式の圧縮ファイルを作成します。

オプション 1 を省略した場合は、KIND\_ZERO 形式の圧縮ファイルを作成します。

'O' と 'C' は併記することも可能です。その場合、KIND\_ORGSIZE | KIND\_CODESIZE 形式の圧縮ファイルを作成します。

**復元** 復元処理の場合は、'D' を指定します。  
圧縮形式を指定する場合は、'D' に続けてオプション 2 を記述します。

オプション 2	説明
N	KIND_NO_INFO 形式の圧縮ファイルを復元します。

オプション 2 を省略した場合は、KIND\_NO\_INFO 形式以外の圧縮ファイルを復元します。

infile 入力ファイル名を指定します。

outfile 出力ファイル名を指定します。


mode 圧縮モードを指定します。(0 または 1 ~ 128 )  
指定の際の目安として本製品の定数値を示します。

値	定数
0	MODE_MAX
32	MODE_HIGH
16	MODE_MID
4	MODE_LOW

圧縮時のみ有効で、復元時は無視されます。

RelcInputStreamNullBlock をご利用の場合は、KIND\_ZERO のみをご利用ください。

<sup>1</sup> 大文字小文字の区別はありません。また、記号の意味は次のとおりです。  
[...] は、その中の要素は記述が省略可能であることを示します  
{.|.} は、その中の要素内で一つを選択することを示します

  
FUJITSU