CRIWARE®

DS用『救声主』、『ファイルマジック』を体験しよう！

株式会社CRI・ミドルウェア

2007年9月27日

CEDEC2007　ワークショップ資料

CRI Middleware Co., Ltd.
Audio & Visual Solution Company

CRIWARE®

DS用ミドルウェアを使ってゲーム開発をお手軽に！
『ファイルマジック』編

CRI FileSystem
ファイルマジック™
for NINTENDO DS®

CRI Middleware Co., Ltd.
Audio & Visual Solution Company

ファイルマジック編の目次

『ファイルマジック』の特長（1）

ファイルマジックは圧縮・展開に対応したファイルシステムです。

■展開機能付きファイル読み込みライブラリ

（1）ファイル読み込み時に圧縮ファイルを自動的に展開。
→圧縮されているか否かを気にせずにファイル読み込みが可能なAPI。

（2）ファイル名やファイル識別子（ID）によるファイルアクセスが可能。
→ファイルをパッキング(圧縮)し、ファイル名やIDでアクセス。
→もちろん、ROM上の通常ファイルを読み込むことも可能です。

（3）独自の圧縮技術『Majik-Decomp™（マジック・ディコンプ™）』を搭載。
→詳細は後ほど。

『ファイルマジック』の特長（2）

■ 非同期データ読み込み

（1）非同期にデータを読み込むことが可能。
　　→例えば、「救声主」で音楽を再生しながらの読み込みも可能。

■ 圧縮機能付きパッキングツール

（1）圧縮が有効か否かを自動判別し、複数のファイルをパッキング可能。

（2）お手軽な「GUI版」ツール、
　　細かな指定が可能な「コンソール版」、
　　Excelの機能を利用した「Excelファイル」をご提供。
　　圧縮・パッキングを行う「DLL（.NET）」の利用も可能。

『ファイルマジック』の特長（3）

■ 圧縮のメリット

（1）ROM容量を抑え、コスト削減に繋がります。

→非圧縮時では収まらなかったデータもコンパクトに。

（2）データが小さくなり、ROMリソースへのアクセスを減らします。

→データ読み込みと他のストリーミングなどに有利。

■ パッキングのメリット

（1）個々のファイルオープンに掛かる負荷を減らします。

（2）ファイル名などのディレクトリ情報を減らすことが出来ます。

→ID情報を用いることで、1ファイルあたり最小6バイトで管理可能。

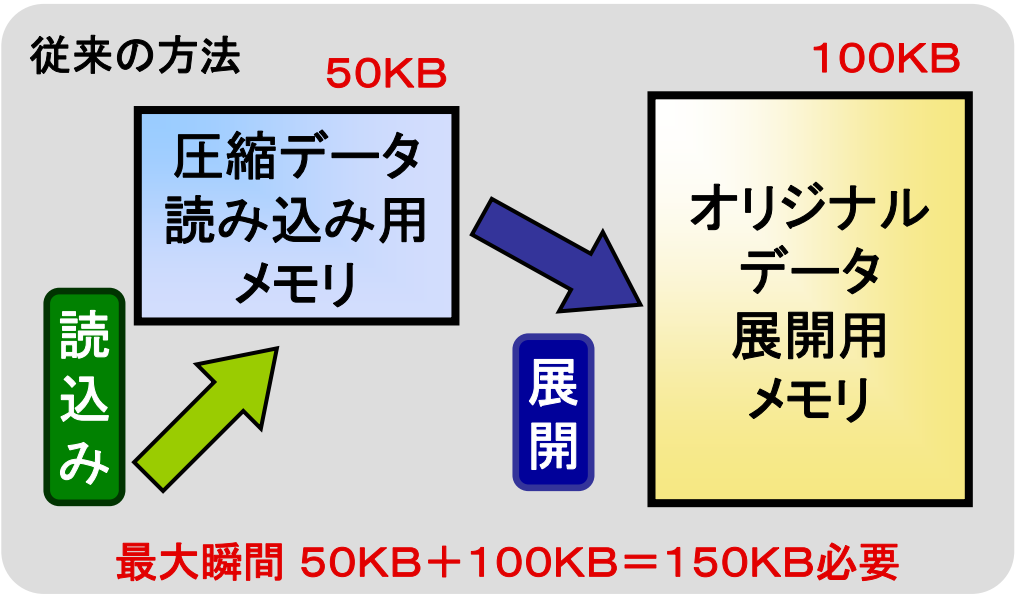
Majik-Decomp™の特長

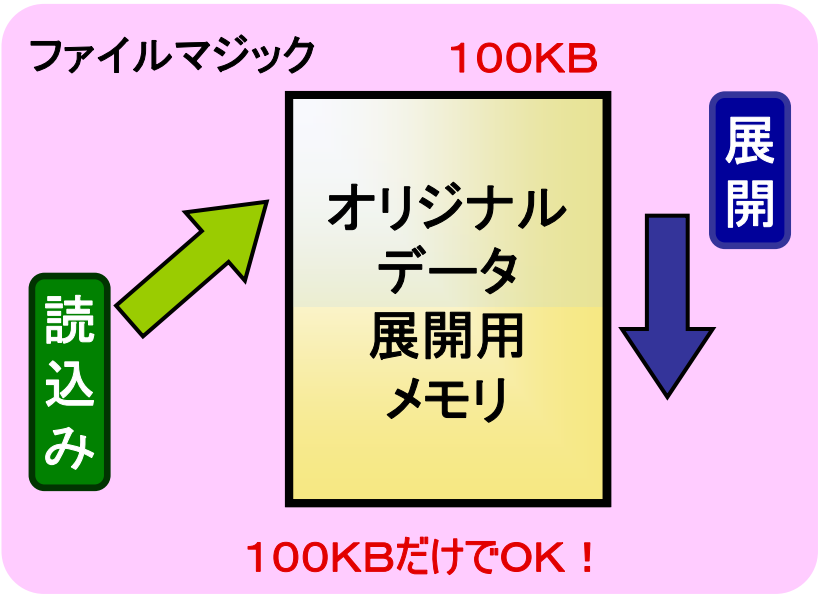
■ バックグラウンド展開

– バックグラウンドで展開を行うため、アプリケーションの処理を妨げません。

■ 自己エリア展開

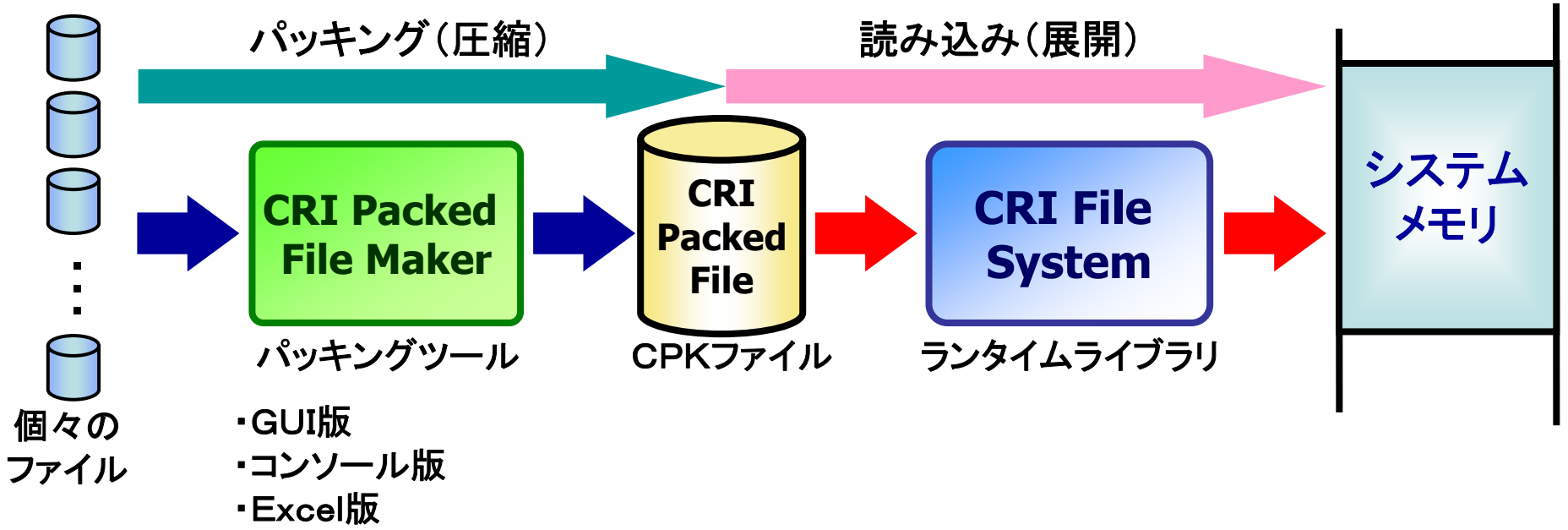
– 別途ワーク不要。オリジナルデータ領域だけで展開することができます。





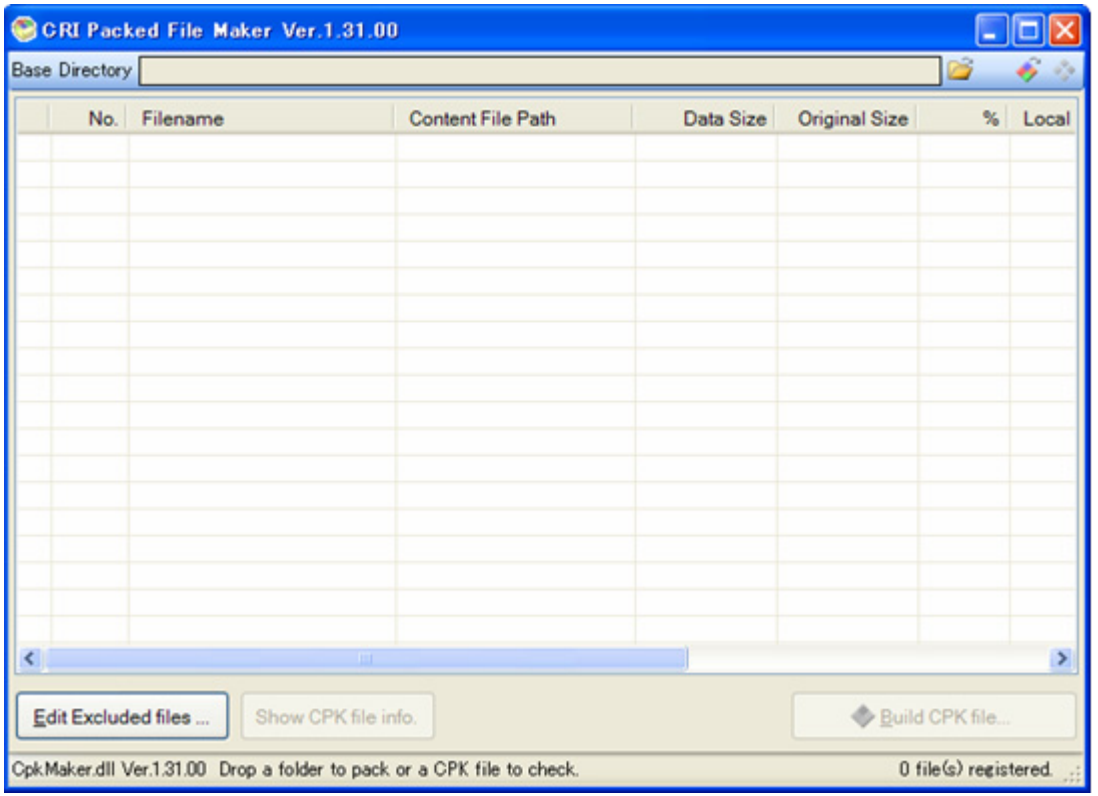
パッキングから読み込みまでのワークフロー

■ 専用ツールにより、CPKファイルへパッキング（圧縮）
■ ランタイムライブラリにより、読み込み（展開）



実践：ファイルをパッキング（圧縮）してみよう（1）

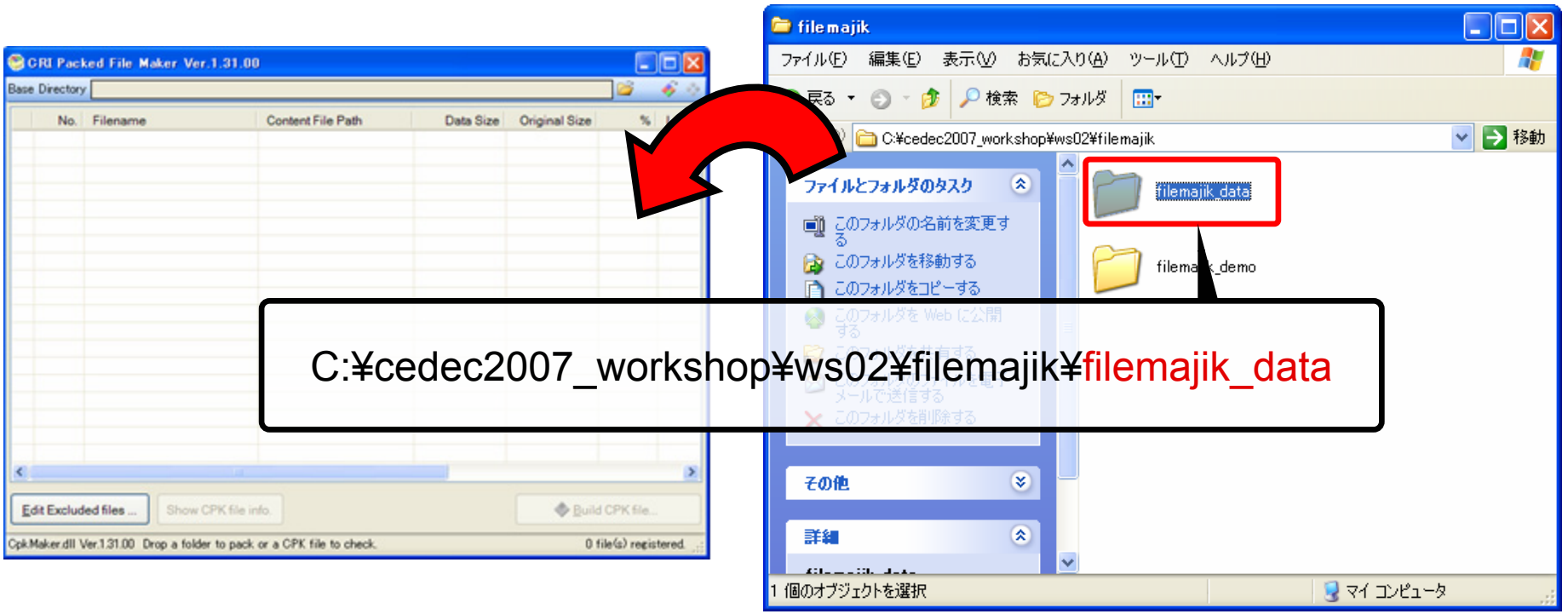
GUI版パッキングツール「CRI Packed File Maker」を起動する。



起動直後の「CRI Packed File Maker」

実践：ファイルをパッキング（圧縮）してみよう（2）

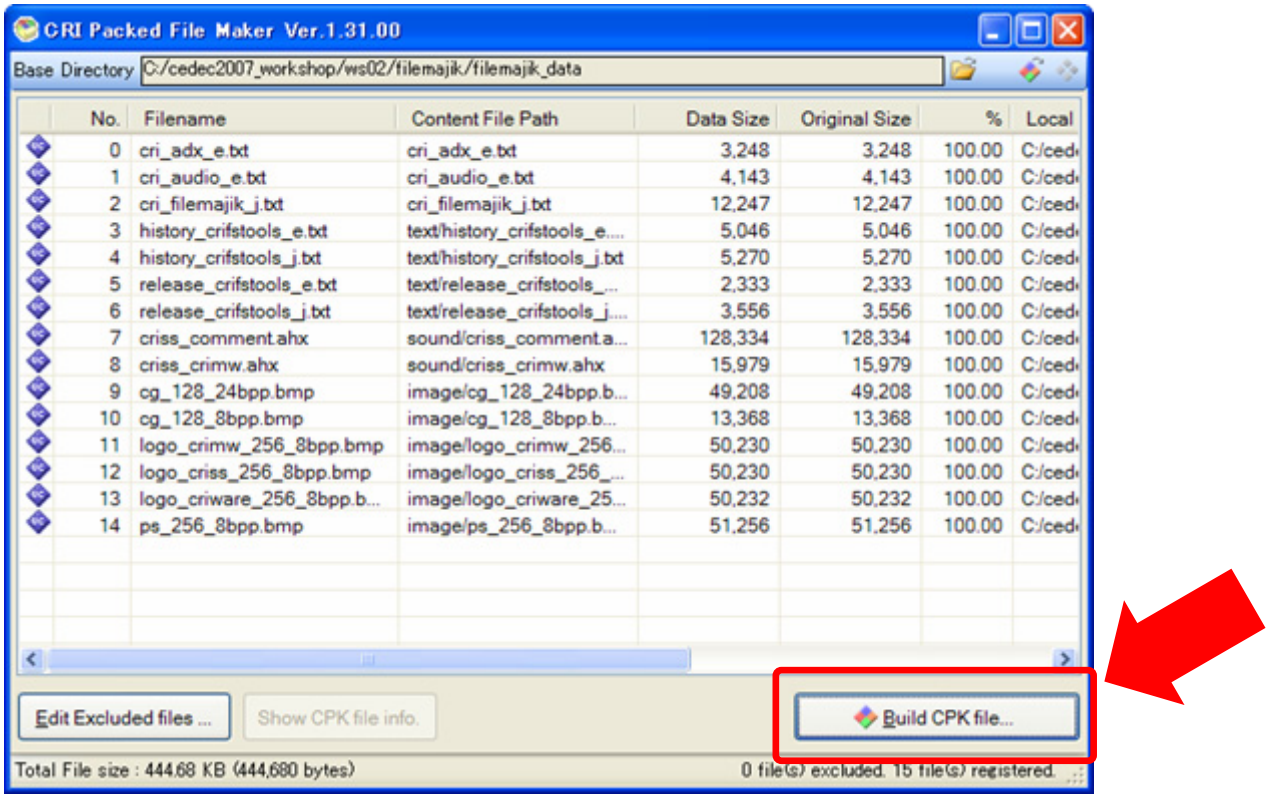
エクスプローラからフォルダをドラッグ &ドロップして登録。



★ ドラッグ &ドロップされたフォルダ以下（サブフォルダを含む）のファイルがパッキングされます。
★ このとき、フォルダ構造も維持されてパッキングされます。

実践：ファイルをパッキング（圧縮）してみよう（3）

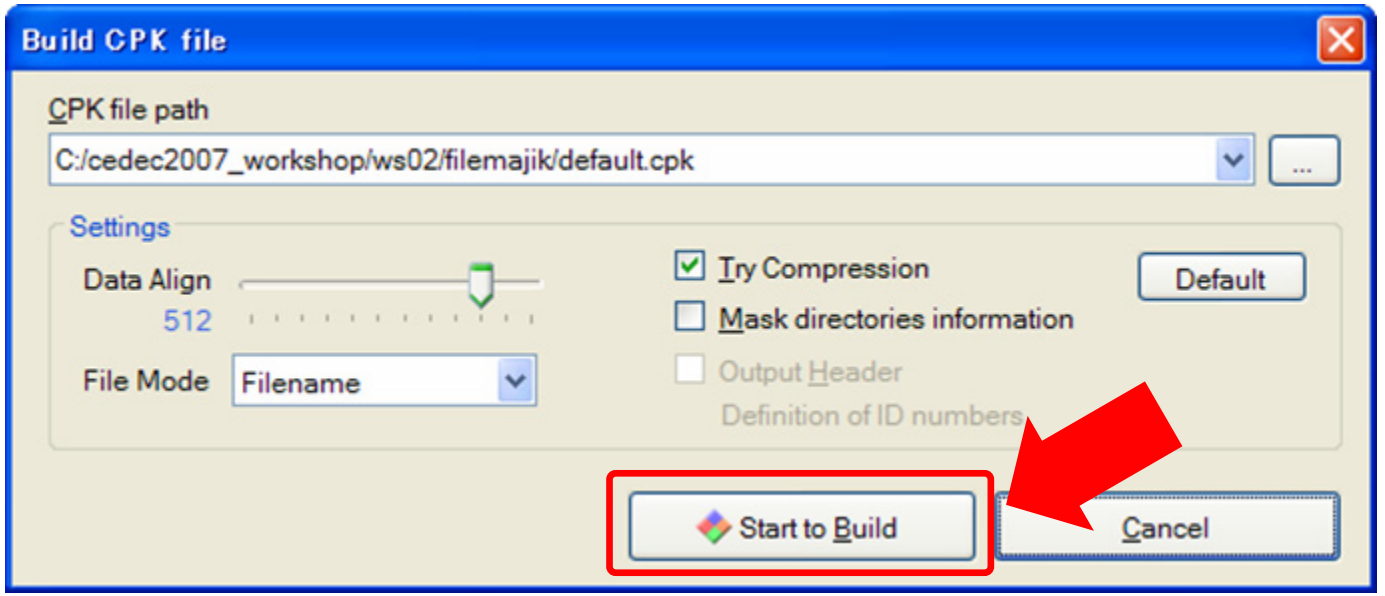
登録されたら、「Build CPK file…」ボタンを押下。



ファイル登録後の「CRI Packed File Maker」

実践:ファイルをパッキング(圧縮)してみよう(4)

パッキング開始！「Start to Build」ボタンを押す。



★設定パラメータ詳細(今回デフォルト設定のまま作成します)

CPK file path
出力ファイルのパス設定

Data Align
データのアライメントサイズ

File Mode
ファイル名、IDなどCPKに含めるアクセス情報設定

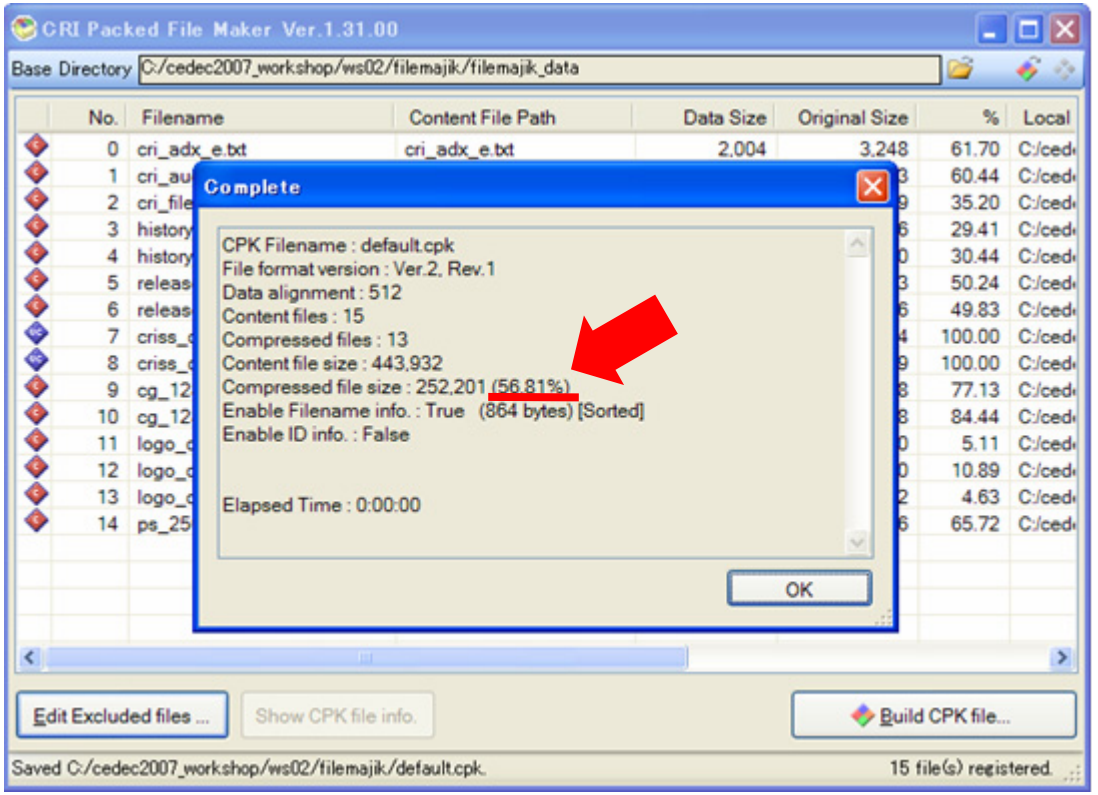
Try Compression
圧縮をするか否かの設定

Mask directories information
CPKディレクトリ情報のマスクの有無

Output Header
ID情報を記述したヘッダファイル出力の有無

実践：ファイルをパッキングしてみよう（5）

パッキング完了！CPKファイルの情報を見てみよう。

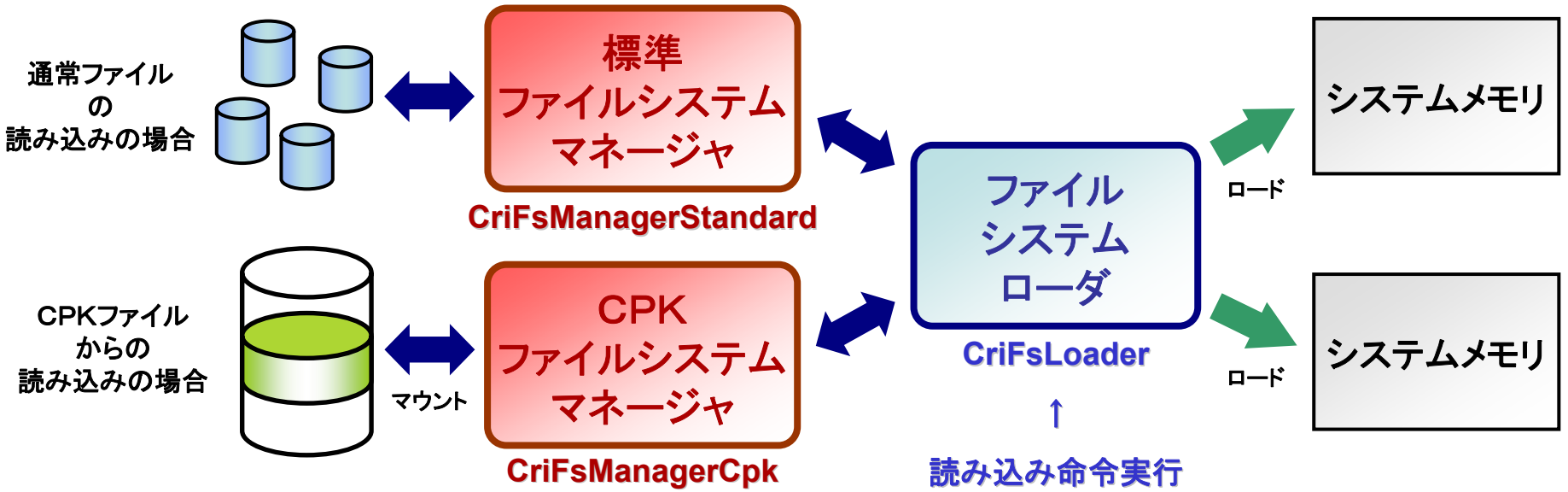


パッキング完了時の「CRI Packed File Maker」

ライブラリの仕組み

（1）**ファイルシステムマネージャ　CriFsManager～**
- デバイスのようなもので、データの読み出し場所を意味します。
- 「標準ファイルシステムマネージャ」と「CPKファイルシステムマネージャ」があります。

（2）**ファイルシステムローダ　CriFsLoader**
- データの読み込みを行うオブジェクトです。FSマネージャを指定して利用します。



ファイルシステムマネージャの作成とマウント

■ CPKファイルシステムマネージャの作成とマウント

```
// ① CPKファイルシステムマネージャの作成
CriFsManagerCpk *fsmng = CriFsManagerCpk::Create(heap);

// ② CPKファイルシステムマネージャにCPKファイルをマウント
fsmng->MountCpkFile("sample.cpk");

// ③ マウント完了待ちループ
for (;;) {
    //  マウント完了の確認
    if ( fsmng->GetMountStatus() == MOUNT_STATUS_COMPLETE ) {
        break; //  マウント完了（ループを抜ける）
    }
}
```

C++言語APIの他に
C言語APIもあります。

★「heap」はCRI独自のヒープ管理システムハンドル。
★ 標準ファイルシステムマネージャ（CriFsManagerStandard）であれば、マウント処理は不要。

ファイルシステムローダの作成とロード

■ ファイルシステムローダの作成とロード

```
// ④ ファイルシステムローダの作成
CriFsLoader *loader = CriFsLoader::Create(heap);

// ⑤ 読み込みの実行 （第1引数は、CPKファイルシステムマネージャ）
loader->Load(fsmng, "data.bin", read_buffer, read_size);

// ⑥ 読み込み完了待ちループ
for (;;) {
   // 読み込み完了の確認
   if ( loader->GetStatus() == CriFsLoader::LOAD_STATUS_COMPLETE ) {
        break; // ロード完了（ループを抜ける）
   }
}
```

圧縮されていても、
されていなくても、
Load関数一発！

★ そのデータが圧縮されているかどうかを全く気にせずにロードが可能！
★ FSマネージャを差し替えるだけで、標準ファイル（非CPK）の読み込みが可能！

実践：CPKファイルからデータを読み込んでみよう（1）

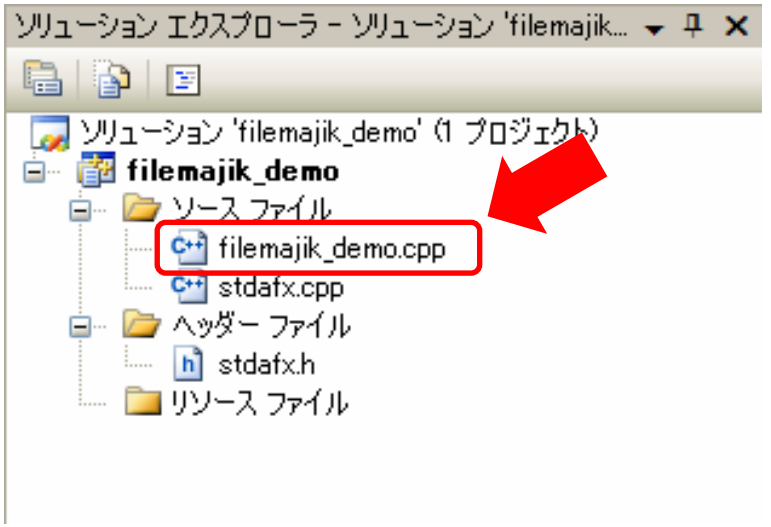
PC版ファイルマジックで
CPKファイルを
読み込みます。

（1）VisualStudioの起動。
以下のソリューションファイルを開く。

C:¥cedec2007_workshop¥ws02¥filemajik¥filemajik_demo¥filemajik_demo.sln

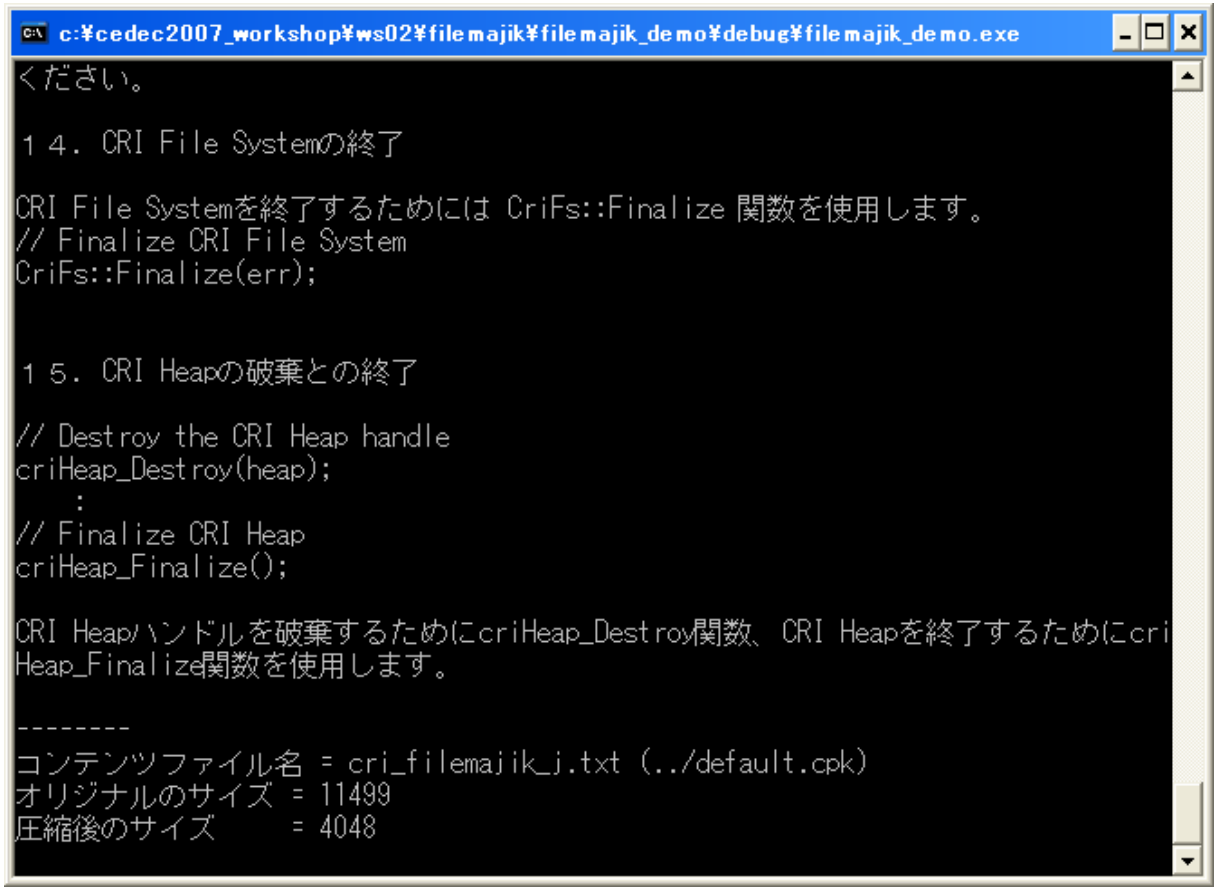
（2）ソースファイルを見てみる。

filemajik_demo.cpp



（3）「F5」キーでビルド＆実行。

実践：CPKファイルからデータを読み込んでみよう（2）



プログラム実行結果

DSならではの注意点

■ 重要：バッファアライメントは32バイトに！

条件１：DMAを利用してファイルを読み込むためには、
32バイトアライメントのバッファを与える必要があります。

→正しく指定されていない場合、DMAが利用できず大幅にパフォーマンスが低下します。が

→あらかじめ、FS_Init関数でDMAを有効にしてください。

■ 重要：ファイル配置は512バイトアライメントに！

条件２：DMAを利用してファイルを読み込むためには、
CPK内のファイルが512アライメント配置である必要があります。

→パッキングツールで、ファイルアライメントを必ず512に指定してください。

→しかしながら、ファイルマジックは512アライメントでなくとも、出来るだけDMAを利用できるように分割してファイルを読み込みます（パフォーマンスは低下します）。

CRIWARE®

DS用ミドルウェアを使ってゲーム開発をお手軽に！
『救声主』編

CRI Middleware Co., Ltd.
Audio & Visual Solution Company

『救声主』編の目次

- 救声主の紹介
- 素材の準備
- プロジェクトの作成
- 素材の登録
- エンコードの設定
- エンコードの開始
- プレビュー
- ニンテンドーDSへの組込み
  - 単純再生
  - マルチストリーム再生
  - クロスフェード
- 応用
  - メモリ再生
  - モノラル音声の音量アップ
  - スレッドモデルフレームワーク
  - ファイルマジックとの連携

『救声主』デモ

ADPCMと救声主を聞き比べてみよう！

512MbitのROMに
モノラル音声ファイルを
- 最大3時間
- 最大6時間
- 最大9時間(救声主のみ)

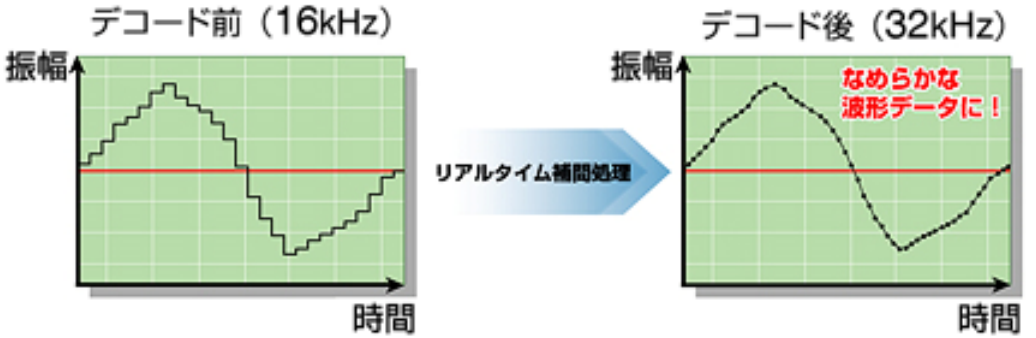
収められるビットレートにした場合
ADPCMと救声主で音質を比較します。

CRI Sound Streamer『救声主』の紹介（1）

・セリフ再生に特化した圧縮技術

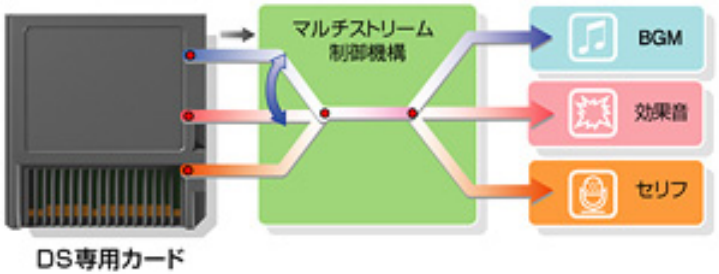
512MbitROMに
ガイド用音声(24kbps)で最大９時間
音質重視 (70kbps)　で最大２時間

・リアルタイム補間技術



CRI Sound Streamer『救声主』の紹介 (2)

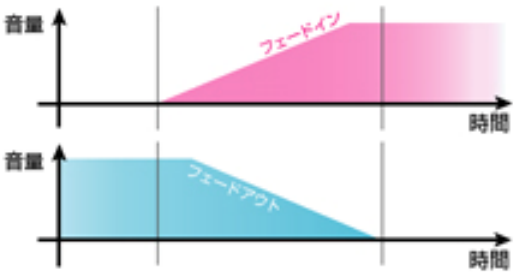
・マルチストリーミング再生機能



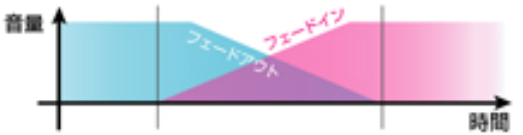
・ループ＆フェード再生機能

■ループ再生
任意のサンプルでループポイントを決めることができます。

■フェードイン・フェードアウト
フェード時間を任意にコントロールしてフェードイン・フェードアウトを設定できます。



■クロスフェード
2つの音声をストリーミング再生しながら相互にフェードして自然に遷移させることができます。



CRI Sound Streamer『救声主』の紹介 (3)

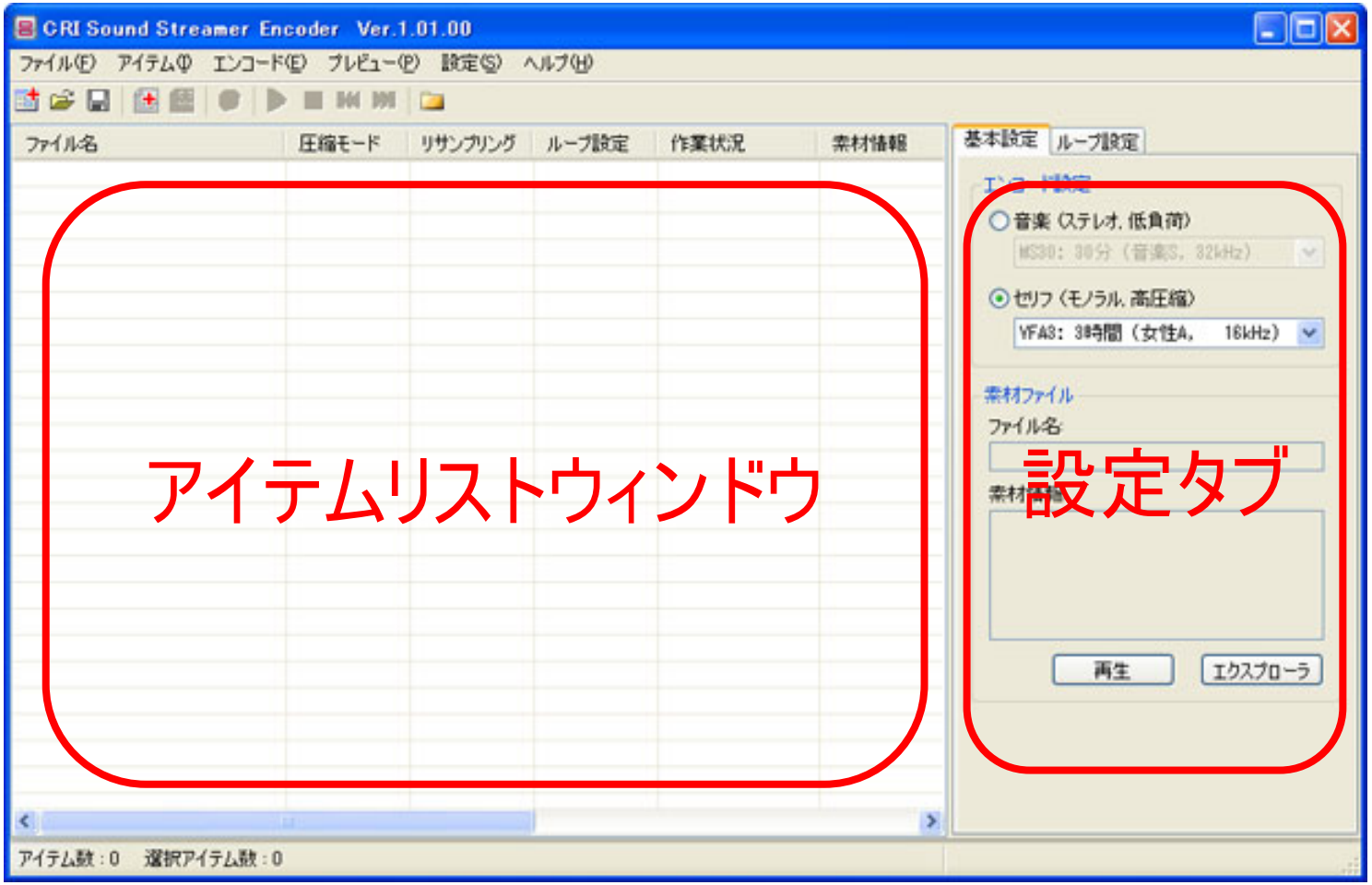
・関数1つで手軽に再生

→ criSsPly_Play(ssply, “voice.ahx”);

・簡単操作で使いやすいエンコードツール

→ CRI Sound Streamer Encoder

CRI Sound Streamer Encoder画面の説明



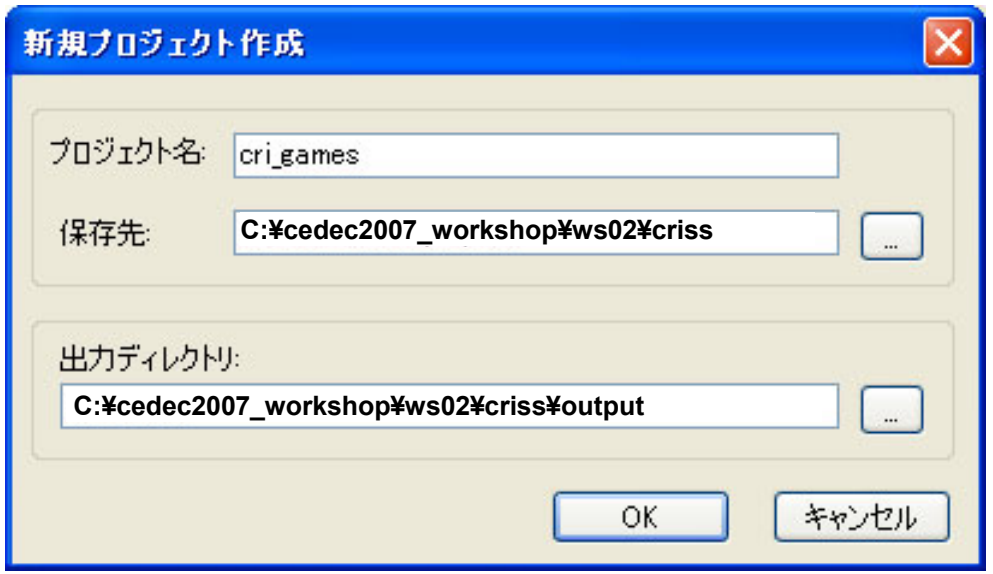
素材の準備

C:¥CRI¥COMMON¥tooldata¥criss　：サンプル素材データ

C:¥cedec2007_workshop¥ws02¥criss　：プロジェクト作成フォルダ

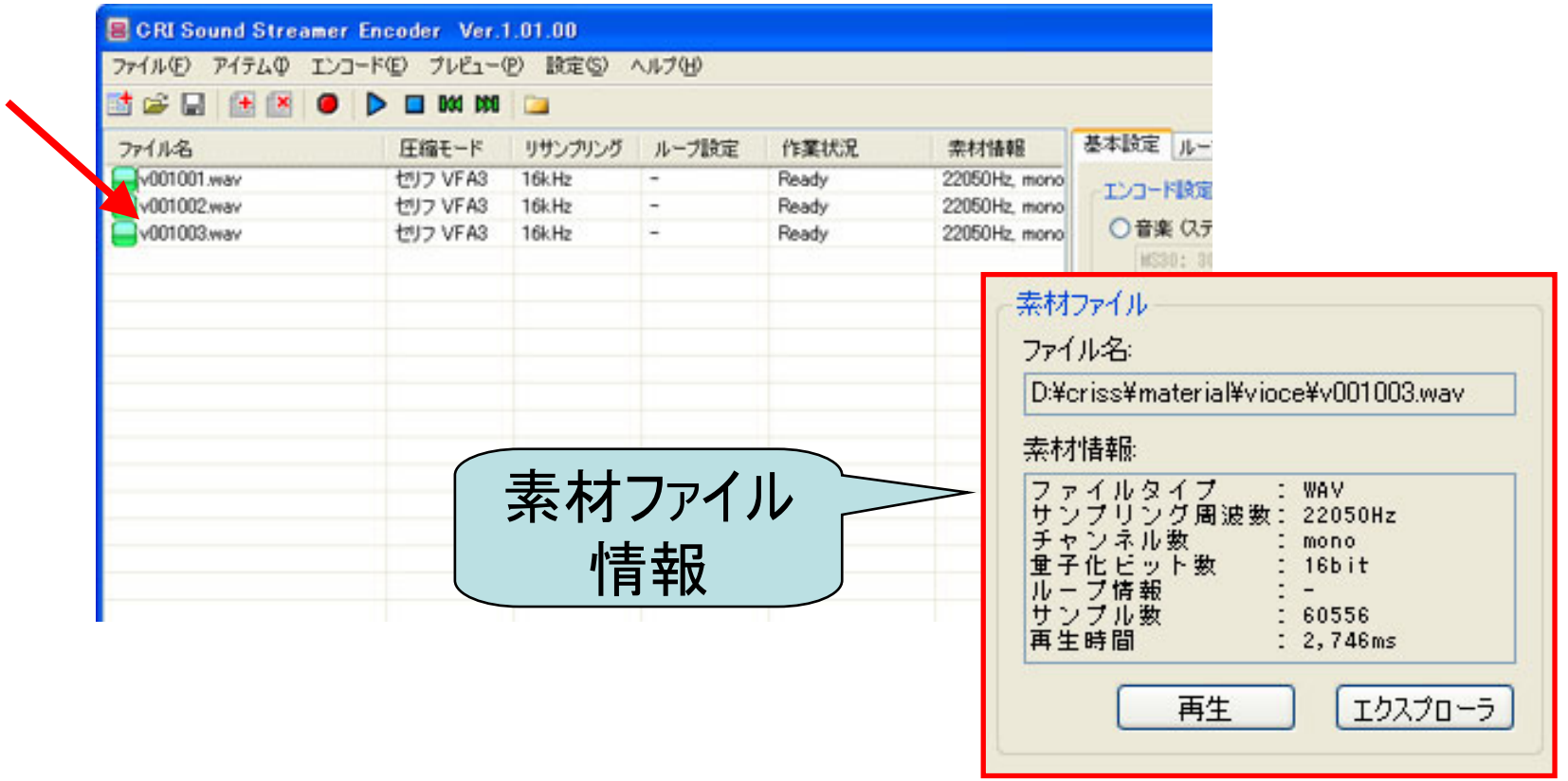
プロジェクトの作成

プロジェクトを新規作成してみよう！



素材ファイルの登録

素材ファイルをドラッグ＆ドロップ！



エンコードテンプレートの選択

エンコードテンプレートを選択しよう！

音楽の場合
(ステレオ or
モノラル)

セリフの
場合
(モノラル)

エンコードテンプレートの指針

■ 素材がステレオ音楽(※)の場合

音楽(ステレオ、低負荷)を選択。
※ モノラル音楽も可。

■ 素材がモノラルの場合

(a) ファイルサイズ優先

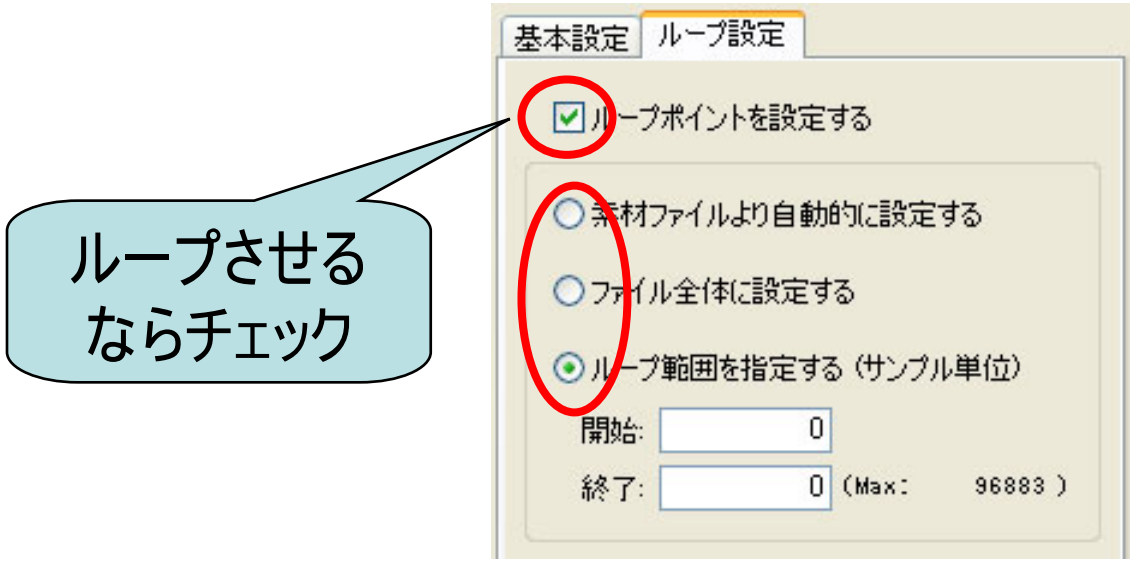
セリフ(モノラル、高圧縮)を選択。VGC9が最高圧縮。
音質とサイズのバランスを見てVGC9～VFS2で調整する。

(b) 音質優先

セリフ(モノラル、高圧縮)を選択。女性の声ならVFS2が最高音質。
男性の声でも声質によってVFS2～VFC5の方が良いことも。

ループ設定

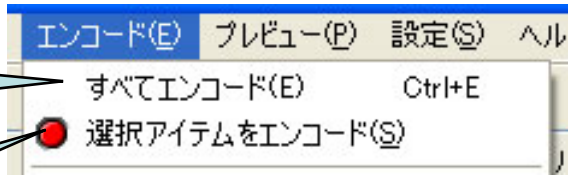
音楽の場合はループ設定をしよう！



エンコードの開始

エンコードしてみよう！

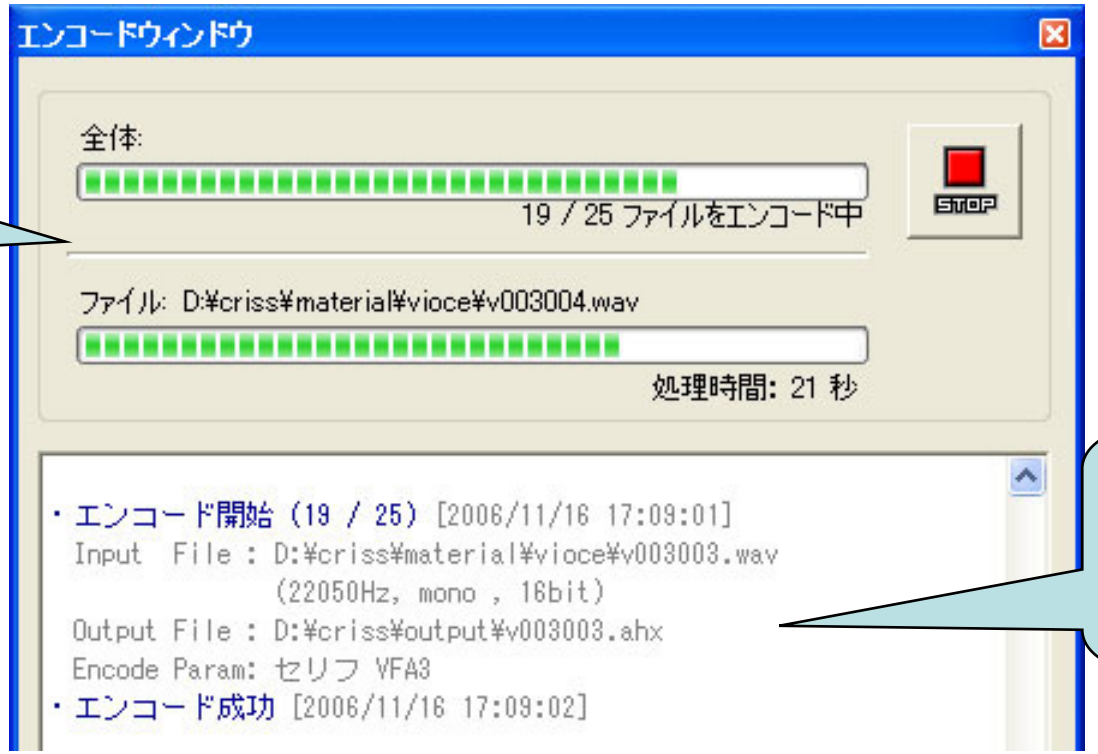
アイテムリストの全ての
素材をエンコード



アイテムリストで選択した
素材だけをエンコード

エンコード中

しばしお待ちを...



エンコードの進行状況

エンコードのログ

エンコード終了

アイテムリストのアイコンを確認しましょう！

アイテムリストの
アイコンイメージ


：未エンコードアイテムです。


：エンコードに成功したアイテムです。

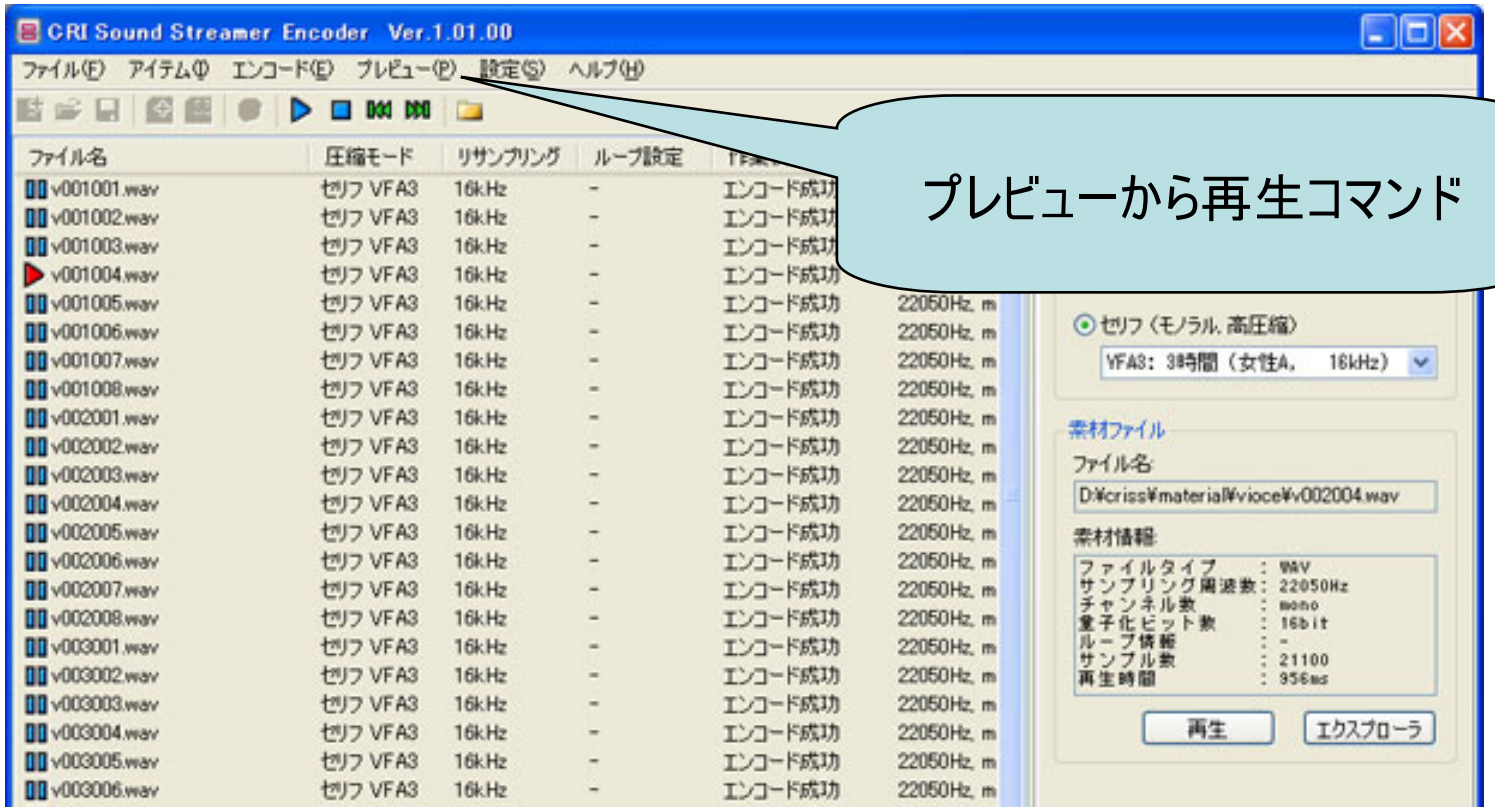

:前回の作業でエンコードが完了しているアイテムです。


:エンコード時にエラーが発生したり、エンコードを中断したアイテムです。


:素材ファイルが存在しない、もしくは素材情報が取得できないアイテムです。

プレビュー

エンコードしたファイルを再生してみよう！



効率良いエンコードをするために

■ 無音部のノイズ

セリフ・高圧縮モードは無音部分の圧縮も行うので、事前に素材から無音部のノイズを取り除いておいてください。

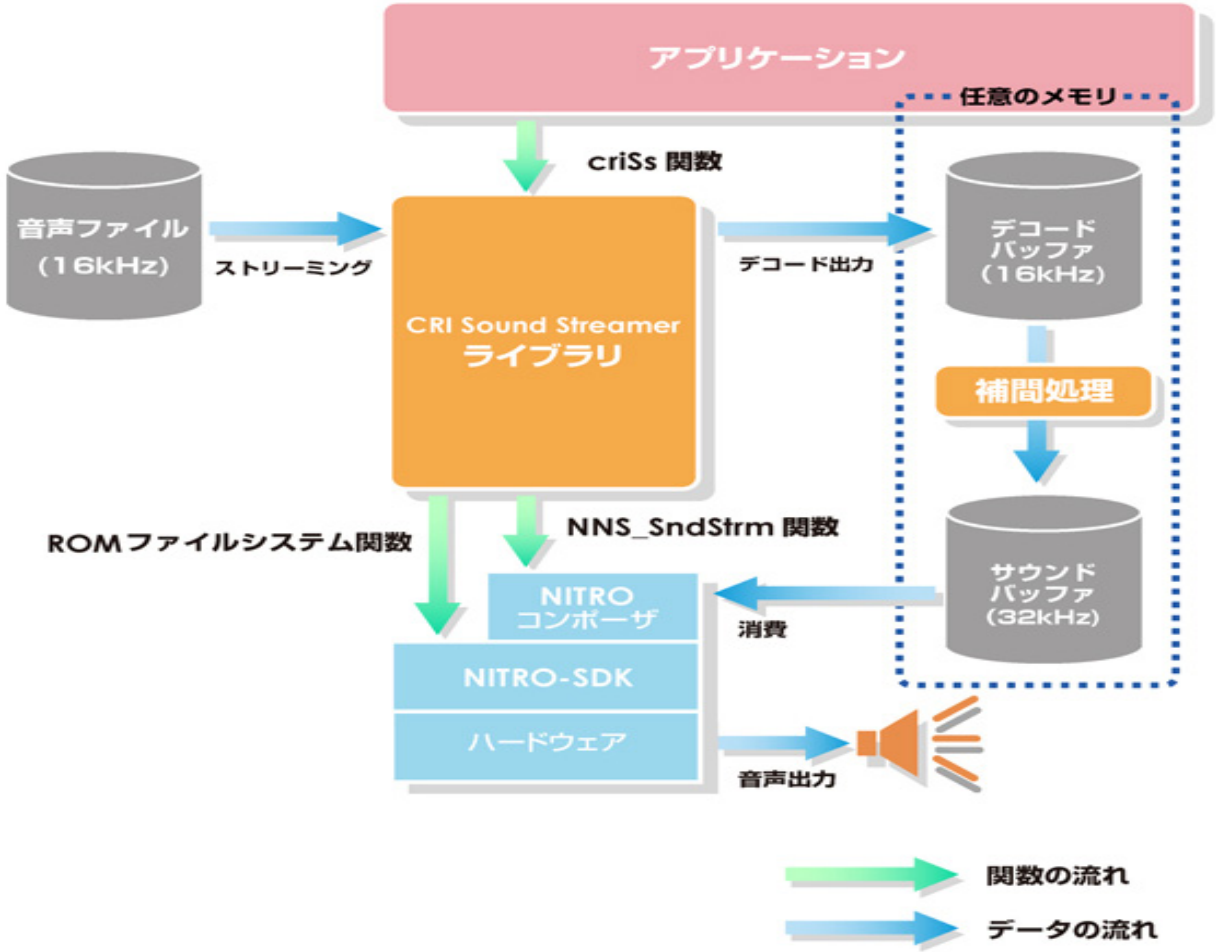
圧縮後のファイルサイズに10～20%くらいの差がでます。

■ 小鳥のさえずり、鈴の音

VCG9～VCA7は、ファイルサイズを小さくできますが「小鳥のさえずり」や「鈴の音」のような高い周波数帯域の音が消えてしまいます。

VMC6から音質を上げていくか、音楽モードを選択してください。

ニンテンドーDSへの組込み



単純再生

音楽を再生してみよう！

プレーヤー
ハンドル

メモリを確保,解放する
関数ポインタ指定

チャンネル数指定
1:モノラル, 2:ステレオ

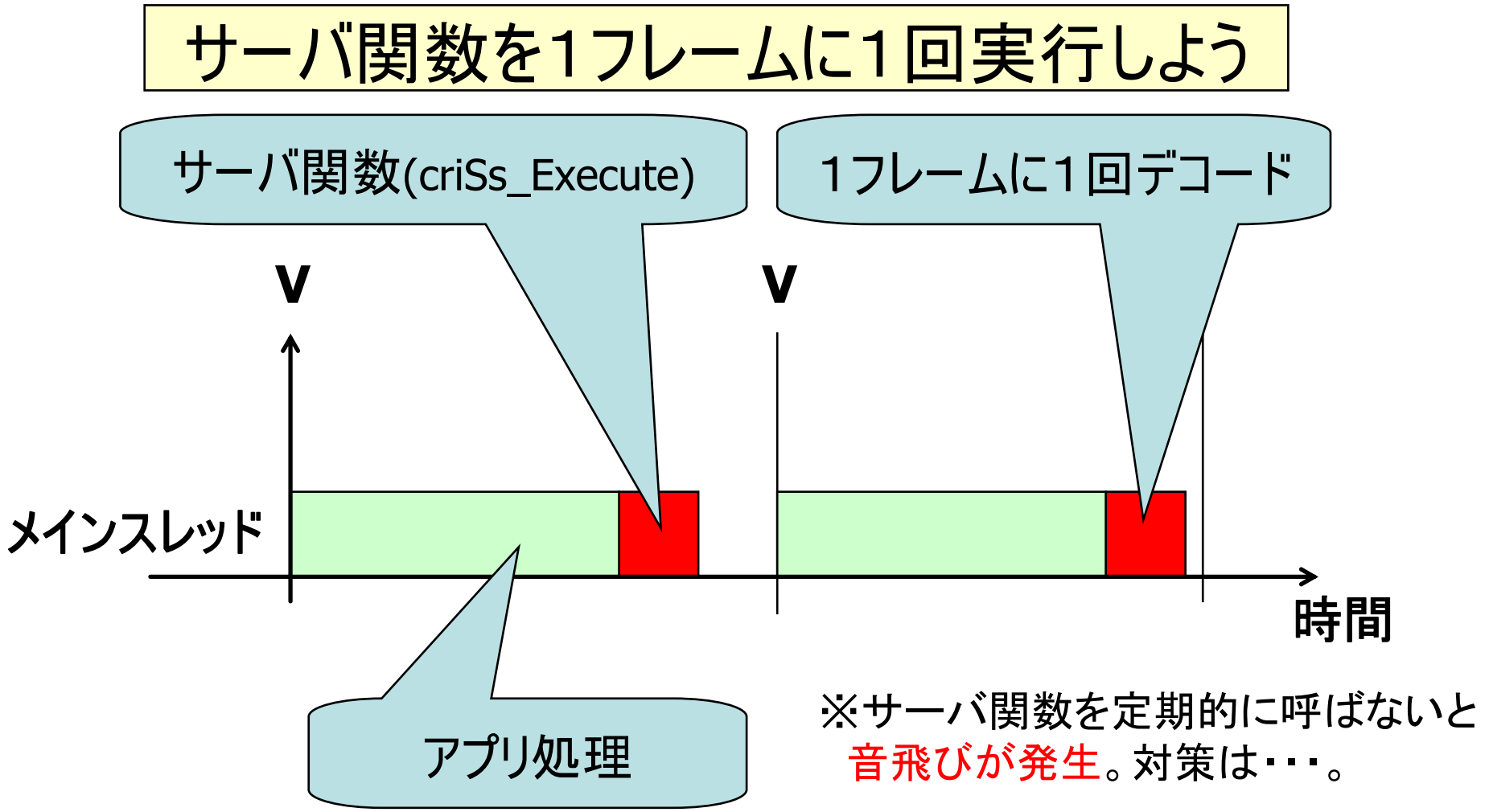
```
// プレーヤーハンドル作成
ssply = criSsPly_Create(&heap, 2, FALSE);

// 再生開始
criSsPly_Play(ssply, "tutorial_32k.adx");
```

クロスフェード
TRUE:有, FALSE:無

ファイル名

サーバ関数について


サーバ関数を1フレームに1回実行しよう
サーバ関数(criSs_Execute)
1フレームに1回デコード
メインスレッド
時間
アプリ処理
※サーバ関数を定期的に呼ばないと
音飛びが発生。対策は・・・。

マルチストリーム再生

複数の音声を同時に再生してみよう！

```
// プレーヤーハンドル作成
ssply_bgm = criSsPly_Create(&heap, 2, FALSE);
ssply_voice = criSsPly_Create(&heap, 1, FALSE);

// 再生開始
criSsPly_Play(ssply_bgm,  "cssSampleSong1_32k.adx");
criSsPly_Play(ssply_voice, "utrecht_male_16k_eng.ahx");
```

最大4ハンドル
作成可能

クロスフェード

2つの音声をクロスフェードしてみよう！

```
// プレーヤーハンドル作成
ssply_bgm = criSsPly_Create(&heap, 2, TRUE);

// 再生開始
criSsPly_Play(ssply_bgm, "cssSampleSong1_32k.adx");

;

criSsPly_Play(ssply_bgm, "cssSampleSong2_32k.adx");
```

クロスフェード有効

同じハンドルを使用して再生

[応用]メモリ再生

メインメモリ上の音声を再生してみよう！

データアドレスを
8桁の16進で表記

データサイズを
8桁の16進で表記

```
// メモリファイルシステム再生
criSsPly_Play(ssply, "MFS:△△△△△△△△.×××××××××");
```

※ループ再生対応

データアドレスを
直接指定

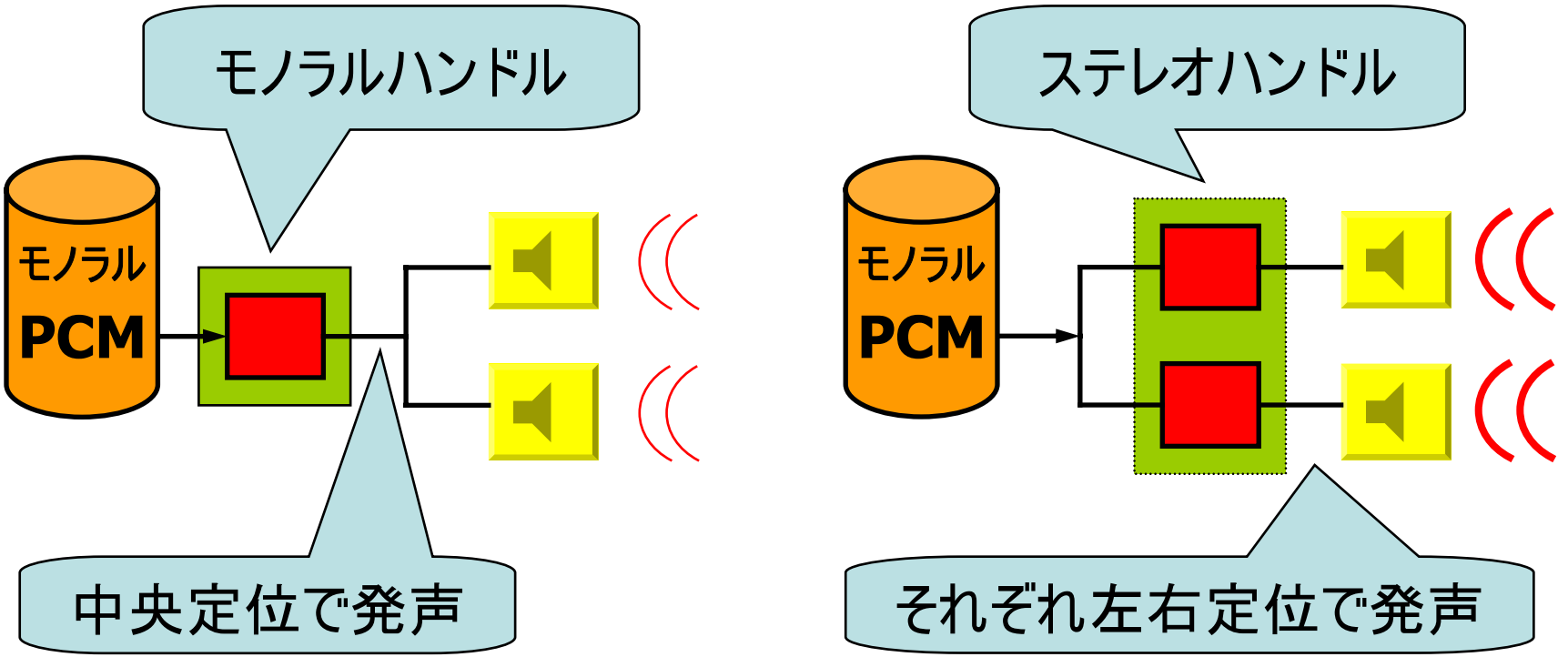
データサイズを
直接指定

```
// メモリ直接再生
criSsPly_PlayFromMemory(ssply, data_address, data_size);
```

※ループ再生非対応, 低レイテンシ

[応用]モノラル音声の音量アップ

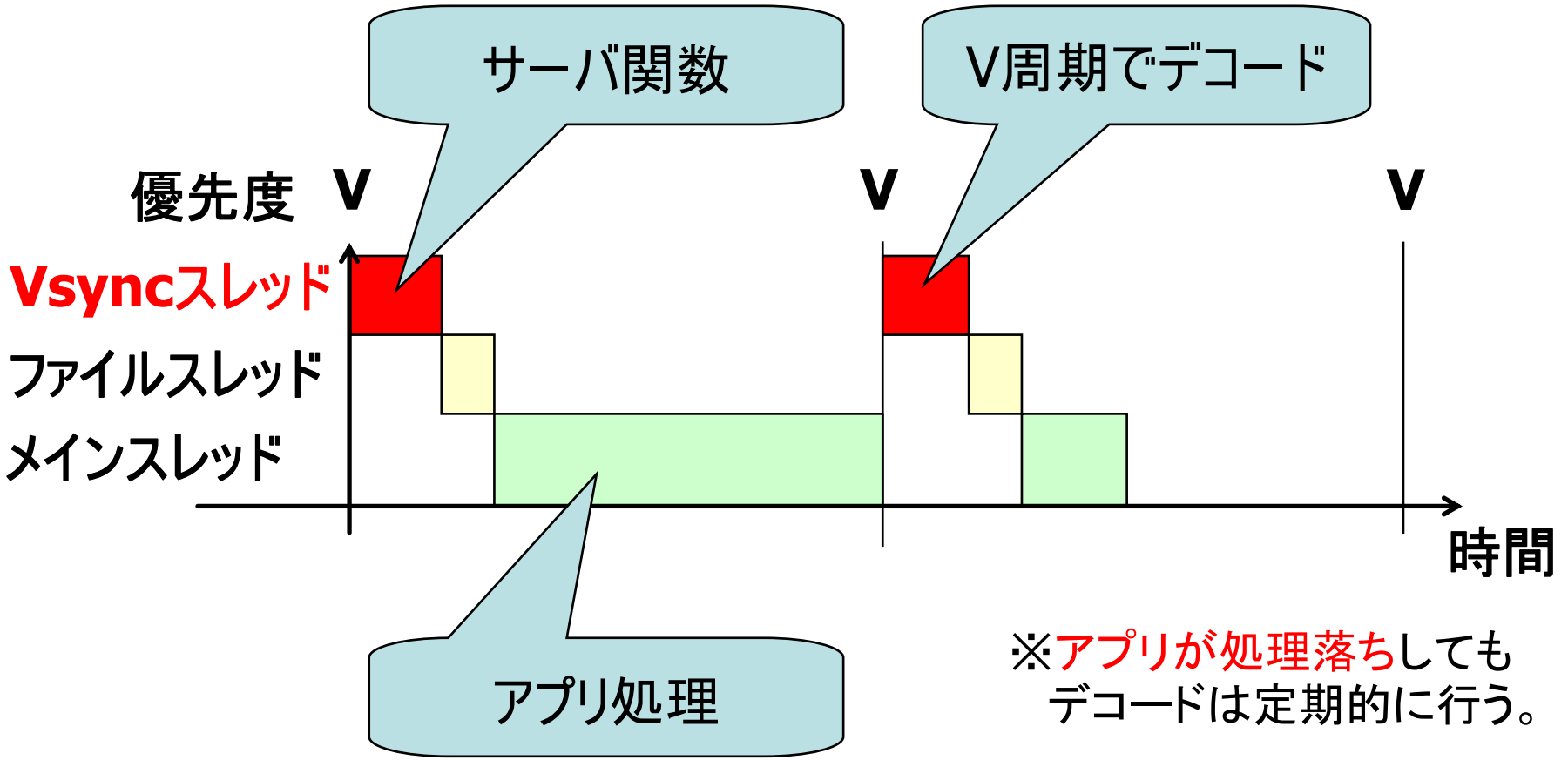
モノラル音声をステレオハンドルで音量アップ！



：DSのサウンドチャンネル (最大16個)

[応用]スレッドモデルフレームワーク

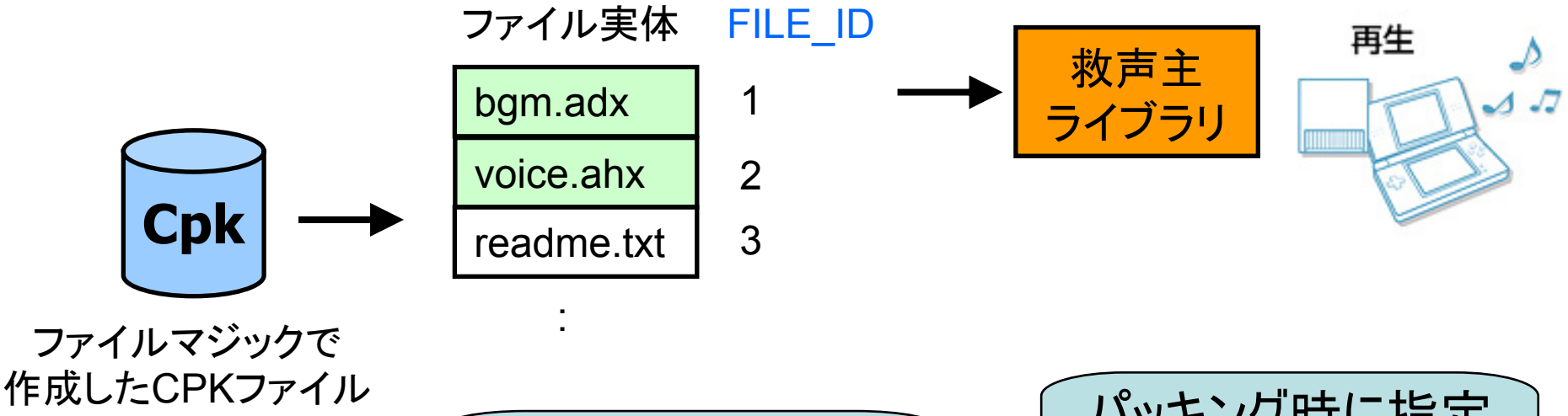
スレッドを使って定期的にデコード！



※アプリが処理落ちしても
デコードは定期的に行う。

[応用]ファイルマジックとの連携

CPKファイルから音声再生！



ファイルマネージャ

パッキング時に指定したファイルID

```
// 再生開始
criSsPly_PlayFromCpkFileById(ssply, fsmng, FILE_ID);
```

お問い合わせ先

URL

http://www.cri-mw.co.jp/inquiry

メール

ファイルマジック　：FileMajik@cri-mw.co.jp

救声主　：Kyuseisyu@cri-mw.co.jp

CEDEC当社講演内容に関するご質問やお問い合わせは、
お気軽に下記メールアドレスまでご連絡下さい。
cedec2007@cri-mw.co.jp