xbcompress Sample

*このサンプルは Microsoft Game Development Kit (2020 年 11月) と互換性があります*

# 説明

このサンプルでは、すべての Gaming.\*.x64 プラットフォームでサポートされている Windows 8 で導入された [Compression API](https://docs.microsoft.com/en-us/windows/win32/cmpapi/-compression-portal) を示します。

パラメーターを指定せずにツールを実行すると、次のようなヘルプ画面が表示されます。

Text

Description automatically generated

# サンプルのビルド

単純なコマンドライン ツールとして、*ゲーム コマンド プロンプト*を使用して直接ビルドできます。

cl /EHsc /D\_WIN32\_WINNT=0x0A00 /Ox /MT compresstool.cpp /Fexbcompress.exe xgameplatform.lib

CMake 3.15 以降を使用できます。

cmake -B out .

cmake –build out

CMake プリセットもあります (CMake 3.19 で導入)。

cmake --list-presets

cmake --preset=x64-Debug

cmake --build out\build\x64-Debug

または、VS IDE から CMakeLists.txt を開くことができます (CMake 3.15 統合には VS 2019 16.3 以降が必要です)。

* ツールをコンソールに少々 “xbcp デプロイ” しやすくするために、静的 Visual C++ ランタイムを使用してビルドします。一般に、タイトルの DLL ベースのランタイムには /MD を使用することをお勧めします。
* このコマンド ライン ツールで使用される API は、onecore\_apiset.lib、xgameplatform.lib、および WindowsApp.lib に存在します。この場合、PC と Xbox の両方で onecore\_apiset.lib アンブレラ ライブラリを安全に使用できます (これは CMake の構成方法です)。ここでも、タイトルには他のアンブレラ ライブラリや kernel32.lib ではなく、xgameplatform.lib を使用することをお勧めします。
* このツールは、onecore\_apiset.lib の代わりに、\_WIN32\_WINNT=0x0602 (Windows 8) または \_WIN32\_WINNT=0x0603 (Windows 8.1) と cabinet.lib をリンクしてビルドできます。Windows 7 以前では、Compression API はサポートされていません。

# 使用

*このツールは、テストの自動化、サンプル、デモ、迅速なプロトタイプなど、最小限の依存関係がある「質より速度を重視する」 CPU ベースの圧縮ソリューションが必要とされる開発シナリオに使用することを目的としています。* ***小売りコンテンツのシナリオの場合、DirectStorage、BCPack、 サードパーティ製ライブラリ、従来の "file-system-in-a-file" ソリューションなど、他にも多くのオプションがあり、それらの方が適しています。***

このサンプルは、Windows 10 ホスト PC、Xbox システム OS、Xbox Game OS と互換性のあるシンプルなコマンドライン ツールです。これを使用して、ファイルを圧縮または圧縮解除できます。

xbcompress.exe mylargefile.bin

- または -

xbcp /x/title xbcompress.exe xd:\

xbrun /x/title /O d:\xbcompress.exe d:\mylargefile.bin

これにより、‘mylargefile.bi\_’ が現在のディレクトリまたは D:\ ディレクトリに書き込まれます。既定では、このファイルは LZMS 圧縮を使用して圧縮されます。

ファイルを展開するには、**/u** スイッチを使用します。

xbcompress /u mylargefile.bi\_

- または -

xbrun /x/title /O d:\xbcompress.exe /u d:\mylargefile.bi\_

これにより、‘mylargefile.bin’ が現在のディレクトリまたは D:\ に書き込まれます。

LZMS 圧縮スキームは、サイズが 2 MB を超えるファイルに適していると見なされます。少しコンパクトなサイズで、少し速い圧縮速度が必要な場合は、代わりに **/z** スイッチを使用して MSZIP で圧縮できます。

# 実装

このサンプルは、従来型の MS-DOS ユーティリティ COMPRESS.EXE と EXPAND.EXE からインスピレーションを得ています。このツールによって生成された ‘\_’ ファイルは、EXPAND.EXE とは互換性がなく、認識されません。圧縮ファイルは常に ‘\_’ で終わります。ファイル拡張子が 3 文字以上の場合、最後の文字は ‘\_’ に置き換えられます。それ以外の場合、 '.\_’ は拡張子として付加されます。

コードをかなり単純にするために、このツールでは圧縮 API ‘バッファー モードを使用します。API は、データをブロックに分割し、圧縮されたデータ ブロックでの圧縮解除に必要なメタデータをエンコードします。

圧縮されたファイルは、次の単純なヘッダーで始まります。

|  |  |  |
| --- | --- | --- |
| ファイル オフセット | フィールドの長さ | 説明 |
| 0 | 8 | ファイル形式を一意に識別するマジック バイト シーケンス。  0x41, 0x46, 0x43, 0x57, 0x47, 0x50, 0x53, 0x4d |
| 9 | 1 | 圧縮モードです。  現在次のモードのみがサポートされています。   * COMPRESS\_ALGORITHM\_LZMS (5) * COMPRESS\_ALGORITHM\_MSZIP (2) |
| 10 | 1 | ファイル形式バージョン。  現在 0x41 ('A') |
| 11 | 2 | 圧縮された名前が決定されたときに '\_' に変更された最後の文字 (UTF-16LE)。代わりに '.\_' が追加された場合、この値は 0 です。 |
| 13 | 4 | 元の圧縮されていないデータ ブロックのサイズ (バイト単位)。  *コードのシンプルさを維持するために、このファイル形式では最大 4 GB のファイル サイズのみがサポートされます。* |

XBCOMPRESS.EXE によって生成されたファイルを圧縮解除するランタイム コードの例は、ATGTK\ReadCompressedData.h/.cpp にあります。

# 更新履歴

|  |  |
| --- | --- |
| リリース | 注意 |
| 2021 年 4 月 | 初回リリース |
| 2022 年 1 月 | CMake クリーンアップとプリセット ファイルの追加。 |