Basic Usage

Jump to bottom

Oz edited this page on Jan 21, 2023 · **12 revisions**

Below are a few examples that show how to use some of the main features of bit7z.

📁 Extracting files from an archive

```cpp
#include <bit7z/bitfileextractor.hpp>

try { //bit7z classes can throw BitException objects
    using namespace bit7z;

    Bit7zLibrary lib{ "7za.dll" };
    BitFileExtractor extractor{ lib, BitFormat::SevenZip };

    //extracting a simple archive
    extractor.extract( "path/to/archive.7z", "out/dir/" );

    //extracting a specific file
    extractor.extractMatching( "path/to/archive.7z", "file.pdf", "out/dir/" );

    //extracting the first file of an archive to a buffer
    std::vector< byte_t > buffer;
    extractor.extract( "path/to/archive.7z", buffer );

    //extracting an encrypted archive
    extractor.setPassword( "password" );
    extractor.extract( "path/to/another/archive.7z", "out/dir/" );
} catch ( const bit7z::BitException& ex ) { /* do something with ex.what()...*/ }
```

💼 Compressing files into an archive

```cpp
#include <bit7z/bitfilecompressor.hpp>

try { //bit7z classes can throw BitException objects
    using namespace bit7z;

    Bit7zLibrary lib{ "7z.dll" };
    BitFileCompressor compressor{ lib, BitFormat::Zip };
```

```cpp
    std::vector< std::string > files = { "path/to/file1.jpg", "path/to/file2.pdf" };

    //creating a simple zip archive
    compressor.compress( files, "output_archive.zip" );

    //creating a zip archive with a custom directory structure
    std::map< std::string, std::string > files_map = {
        { "path/to/file1.jpg", "alias/path/file1.jpg" },
        { "path/to/file2.pdf", "alias/path/file2.pdf" }
    };
    compressor.compress( files_map, "output_archive2.zip" );

    //compressing a directory
    compressor.compressDirectory( "dir/path/", "dir_archive.zip" );

    //creating an encrypted zip archive of two files
    compressor.setPassword( "password" );
    compressor.compressFiles( files, "protected_archive.zip" );

    //updating an existing zip archive
    compressor.setUpdateMode( UpdateMode::Append );
    compressor.compressFiles( files, "existing_archive.zip" );

    //compressing a single file into a buffer
    std::vector< byte_t > buffer;
    BitFileCompressor compressor2{ lib, BitFormat::BZip2 };
    compressor2.compressFile( files[0], buffer );
} catch ( const bit7z::BitException& ex ) { /* do something with ex.what()...*/ }
```

📄 Reading archive metadata

```cpp
#include <bit7z/bitarchivereader.hpp>

try { //bit7z classes can throw BitException objects
    using namespace bit7z;

    Bit7zLibrary lib{ "7za.dll" };
    BitArchiveReader arc{ lib, "archive.7z", BitFormat::SevenZip };

    //printing archive metadata
    cout << "Archive properties" << endl;
    cout << " Items count: "   << arc.itemsCount() << endl;
    cout << " Folders count: " << arc.foldersCount() << endl;
    cout << " Files count: "   << arc.filesCount() << endl;
    cout << " Size: "          << arc.size() << endl;
    cout << " Packed size: "   << arc.packSize() << endl;
    cout << endl;

    //printing archive items metadata
```

```
    cout << "Archive items";
    auto arc_items = arc.items();
    for ( auto& item : arc_items ) {
        cout << endl;
        cout << " Item index: "   << item.index() << endl;
        cout << "  Name: "        << item.name() << endl;
        cout << "  Extension: "   << item.extension() << endl;
        cout << "  Path: "        << item.path() << endl;
        cout << "  IsDir: "       << item.isDir() << endl;
        cout << "  Size: "        << item.size() << endl;
        cout << "  Packed size: " << item.packSize() << endl;
    }
} catch ( const bit7z::BitException& ex ) { /* do something with ex.what()...*/ }
```

Documentation for **bit7z** *v4.0.9*
Copyright © 2014 - 2024 Riccardo Ostani ([@rikyoz](#))

▸ **Pages** 40

bit7z

- [Downloads](#)
- [License](#)
- [Support](#)

Getting Started

- [Building the library](#)
  - [Build options](#)
- [Installing the library](#)
- [Requirements](#)
  - [7z DLLs](#)
- [Basic Usage](#)
- [Advanced Usage](#)

API Reference (v4.0.5)

main classes and type aliases

- [Bit7zLibrary](#)
- [BitArchiveEditor](#)
- [BitArchiveReader](#)
- [BitArchiveWriter](#)
- [BitException](#)
- [BitFileCompressor](#)
- [BitFileExtractor](#)
- [BitMemCompressor](#)