# Archived:Compressing and decompressing files using CEZFileBufferManager

**Archived:** This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

## Overview

This snippet shows how to compress and decompress files using the classes `CEZCompressor`, `CEZDecompressor`, and `CEZFileBufferManager`.

This snippet can be self-signed.

## MMP file

The following libraries are required:

```
LIBRARY efsrv.lib
LIBRARY ezlib.lib
```

## Header file

## Source file

```
#include <ezcompressor.h>
#include <ezdecompressor.h>
#include <ezfilebuffer.h>
#include <f32file.h>

void CompressFileL(RFs &aFs, TInt aBufferSize, TInt aMethod, const TDesC& aFileName)
{
  TInt err(KErrNone);
  RFile input;
  RFile output;

  HBufC *compressedFile = HBufC::NewLC(aFileName.Length()+2);
  _LIT(KCompressedFileName,"%S.c");
  compressedFile->Des().Format(KCompressedFileName,&aFileName);

  User::LeaveIfError(input.Open(aFs,aFileName,EFileStream | EFileRead | EFileShareAny));
  CleanupClosePushL(input);

  err = output.Create(aFs, *compressedFile,EFileStream | EFileWrite | EFileShareExclusive);

  if (err == KErrAlreadyExists)
    User::LeaveIfError(output.Open(aFs, *compressedFile,EFileStream | EFileWrite |
 EFileShareExclusive));
  else
    User::LeaveIfError(err);
  CleanupClosePushL(output);

  CEZFileBufferManager *fileBufferManager = CEZFileBufferManager::NewLC(input,output, aBufferSize);
  CEZCompressor *compressor = CEZCompressor::NewLC(*fileBufferManager, aMethod);

  _LIT(KCompressingFileText,"Compressing file %S to %S\n");
  console->Printf(KCompressingFileText,&aFileName,compressedFile);

  while (compressor->DeflateL())
    {
    // loop here until the file is compressed
    }

  CleanupStack::PopAndDestroy(5); //compressor,bufManager,output,input,compressedFile
}

void DecompressFileL(RFs &aFs, TInt aBufferSize, const TDesC& aFileName)
{
  TInt err(KErrNone);
  RFile input;
  RFile output;

  User::LeaveIfError(input.Open(aFs, aFileName,EFileStream | EFileRead | EFileShareAny));
  CleanupClosePushL(input);

  HBufC *decompressedFile = HBufC::NewLC(aFileName.Length()+2);
  _LIT(KDecompressedFileName,"%S.d");
  decompressedFile->Des().Format(KDecompressedFileName,&aFileName);

  err = output.Create(aFs, *decompressedFile,EFileStream | EFileWrite | EFileShareExclusive);
  if (err == KErrAlreadyExists)
    User::LeaveIfError(output.Open(aFs, *decompressedFile,EFileStream | EFileWrite |
 EFileShareExclusive));
  else
    User::LeaveIfError(err);
  CleanupClosePushL(output);

  CEZFileBufferManager *fileBufferManager = CEZFileBufferManager::NewLC(input,output,aBufferSize);
  CEZDecompressor *decompressor = CEZDecompressor::NewLC(*fileBufferManager);
```


(C) Copyright Nokia 2013. All rights reserved.

```
  _LIT(KDecompressingFileText,"Decompressing file %S from %S\n");
  console->Printf(KDecompressingFileText,decompressedFile,&aFileName);

  while (decompressor->InflateL())
    {
    // loop here until the file is decompressed
    }

  CleanupStack::PopAndDestroy(5); //decompressor,bufManager,output,input,decompressedFile

}
```

## Using CompressFileL() and DecompressFileL() methods

This simple example method reads command line arguments and compresses or decompresses given files.

Options:

- -c = compress
- -d = decompress
- -b n = buffer size
- -m n = method

(from ezlib.h: 0(Z_NO_COMPRESSION), 1(Z_BEST_SPEED), 9(Z_BEST_COMPRESSION), -1(Z_DEFAULT_COMPRESSION))

- filename = compressed or decompressed file

```
void doCompressionDecompressionL()
  {
  RFs fs;
  User::LeaveIfError(fs.Connect());
  CleanupClosePushL(fs);

  TBool doCompress = ETrue;
  TInt bufferSize = 32768;
  TInt method = Z_BEST_COMPRESSION;
  TInt cmdLineLen = User::CommandLineLength();

  if (cmdLineLen <= 0)
    {
    _LIT(KUsage,"Usage: program.exe [-cd] [-u buffer] [-m method] filename\n");
    console->Printf(KUsage);
    User::Leave(KErrGeneral);
    }

  HBufC *argv = HBufC::NewLC(cmdLineLen);
  TPtr argPtr=argv->Des();
  User::CommandLine(argPtr);

  TLex arguments(*argv);

  TPtrC options(arguments.NextToken());
  TBool bufferSizeSpecified = EFalse;
  TBool methodSpecified = EFalse;

  _LIT(KInvalidOption,"Invalid option %S\n");
  _LIT(KUnknownOption,"Unknown option %S\n");
  _LIT(KNoOptionSpecified,"No option specified\n");

  while (options[0]=='-' || bufferSizeSpecified || methodSpecified)
    {
    TInt index = 1;

    if (bufferSizeSpecified || methodSpecified)
      {
      if (options.Length() == 0)
        {
        console->Printf(KNoOptionSpecified);
        console->Getch();
        User::Leave(KErrGeneral);
        }
      else
        {
        TLex lex(options);
        TInt ret = KErrNone;

        if(bufferSizeSpecified)
          {
          ret = lex.Val(bufferSize);
          bufferSizeSpecified = EFalse;
          }
        else
          {
          ret = lex.Val(method);
          methodSpecified  = EFalse;
          }

        if (ret != KErrNone)
          {
          console->Printf(KInvalidOption,&options);
          console->Getch();
          User::Leave(KErrGeneral);
          }
        }
      }
    else
      {
      while (index < options.Length())
        {
        if (options[index] == 'd')
          doCompress = EFalse;
        else if (options[index] == 'c')
          doCompress = ETrue;
        else if (options[index] == 'b' )
           bufferSizeSpecified = ETrue;
        else if (options[index] == 'm')
          methodSpecified = ETrue;
```


(C) Copyright Nokia 2013. All rights reserved.

```
        else
          {
          console->Printf(KUnknownOption,&options);
          console->Getch();
          User::Leave(KErrGeneral);
          }
        index++;
        }

      if (index == 1)
        {
        console->Printf(KNoOptionSpecified);
        console->Getch();
        User::Leave(KErrGeneral);
        }
      }
    options.Set(arguments.NextToken());
    }

    TPtrC fileNamePtr(options);
    HBufC *fileNameBuf = HBufC::NewLC(fileNamePtr.Length());
    *fileNameBuf = fileNamePtr;

    if(doCompress)
      {
       CompressFileL(fs,bufferSize,method,*fileNameBuf);
      }
    else
      {
       DecompressFileL(fs,bufferSize,*fileNameBuf);
      }

    CleanupStack::PopAndDestroy(3); //fileNameBuf,argv,fs
}
```

## Postconditions

Depending on given parameters and the used method, files are compressed or decompressed.

## See also

Archived:Compressing and decompressing GZIP files using Symbian C++


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.