Subject: GZDecompress bug
Posted by Novo on Tue, 04 Aug 2009 03:49:57 GMT
View Forum Message <> Reply to Message

GZDecompress has a bug when size of an archive is bigger than size of an archived file.

I attached a test case. Sorry, I couldn't figure out myself how to fix that.

File Attachments
1) test_gunzip.zip, downloaded 320 times

Subject: Re: GZDecompress bug
Posted by mirek on Tue, 04 Aug 2009 18:05:54 GMT
View Forum Message <> Reply to Message

Novo wrote on Mon, 03 August 2009 23:49GZDecompress has a bug when size of an archive is bigger than size of an archived file.

I attached a test case. Sorry, I couldn't figure out myself how to fix that.

I am quite confused with testcase. Everything seems to be OK, except that in the case with .gz with longer name (not used in testcase), the size of memory stream is not long enough..

Mirek

Subject: Re: GZDecompress bug
Posted by Novo on Wed, 05 Aug 2009 03:40:34 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 04 August 2009 14:05Novo wrote on Mon, 03 August 2009 23:49GZDecompress has a bug when size of an archive is bigger than size of an archived file.

I attached a test case. Sorry, I couldn't figure out myself how to fix that.

I am quite confused with testcase. Everything seems to be OK, except that in the case with .gz with longer name (not used in testcase), the size of memory stream is not long enough..

Mirek

If I understand correctly memory stream size is supposed to be equal to the size of an archived file (because I'm extracting this file into the memory stream). At least this test works fine with archives where size of an archived file is bigger than size of an archive itself.

Another .gz file, which is not used in the testcase, is just another example. If you want to use it, then idx_file_size should be set to 80.

Sorry, I forgot to describe wrong behavior. If you extract archived file using another application (I usually use Total Commander for such operations), and using test_gunzip and compare contents of extracted files you will see the difference.

Actually, the "a.a.a" file will contain the "eeFreeFreeFreeFreeFreeFree" suffix.

I tested that on Windows Vista 32 bit.

I ran a similar application on 114 files and only these two were extracted not correctly.

Subject: Re: GZDecompress bug
Posted by mirek on Wed, 05 Aug 2009 06:23:44 GMT
View Forum Message <> Reply to Message

Novo wrote on Tue, 04 August 2009 23:40
Sorry, I forgot to describe wrong behavior. If you extract archived file using another application (I usually use Total Commander for such operations), and using test_gunzip and compare contents of extracted files you will see the difference.

Actually, the "a.a.a" file will contain the "eeFreeFreeFreeFreeFreeFree" suffix.

OK, thanks. I was in impression that the size is wrong...

Mirek

Subject: Re: GZDecompress bug
Posted by mirek on Wed, 05 Aug 2009 06:49:51 GMT
View Forum Message <> Reply to Message

int GZDecompress(Stream& out, Stream& in, int size, Gate2<int, int> progress)

Here the 'size' parameter is a number of bytes to be read from in.

The size of .gz is 75 bytes, but you put there 48.

#include <Core/Core.h>

```
using namespace Upp;

CONSOLE_APP_MAIN
{
 FileOut out("u:/gztest/output.bin");
 FileIn in;

 if (!in.Open("u:/gztest/MedicalEnRu_abrv.idx.gz"))
  return;

 Buffer<char> index_data;
 MemStream index_stream;
 const int idx_file_size = 48;

 // Preallocate memory and create a memory stream ...
 index_data.Alloc(idx_file_size);
 index_stream.Create(~index_data, idx_file_size);

 GZDecompress(index_stream, in, in.GetLeft());
 index_stream.Seek(0);

 CopyStream(out, index_stream);
}
```

This works.

BTW, why MemStream? You can use 'out' as output directly without CopyStream.

Mirek

Subject: Re: GZDecompress bug
Posted by Novo on Wed, 05 Aug 2009 15:17:20 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 05 August 2009 02:49
int GZDecompress(Stream& out, Stream& in, int size, Gate2<int, int> progress)

Here the 'size' parameter is a number of bytes to be read from in.

The size of .gz is 75 bytes, but you put there 48.

GZDecompress(index_stream, in, in.GetSize());

of the input stream because it can be easily retrieved from the input stream. The only useful size I could imagine in this situation was a minimal size of the output stream (to avoid multiple reallocations).

A few lines of documentation would help me a lot in this situation.

Quote:
BTW, why MemStream? You can use 'out' as output directly without CopyStream.

Mirek

This is just a testing application. In a real application I keep uncompressed data in memory.

Thanks a lot for your help!

Subject: Re: GZDecompress bug
Posted by mirek on Wed, 05 Aug 2009 15:26:36 GMT
View Forum Message <> Reply to Message

Novo wrote on Wed, 05 August 2009 11:17

of the input stream because it can be easily retrieved from the input stream.

The idea is that in some cases, you might want to store more .gz parts in single stream... In that case you need to know when to stop decompressing (Of course, this is perhaps only useful for ZDecompress (without header), but was kept for header variant).

Mirek

Subject: Re: GZDecompress bug
Posted by Novo on Wed, 05 Aug 2009 17:16:43 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 05 August 2009 11:26Novo wrote on Wed, 05 August 2009 11:17

of the input stream because it can be easily retrieved from the input stream.

The idea is that in some cases, you might want to store more .gz parts in single stream... In that case you need to know when to stop decompressing (Of course, this is perhaps only useful for ZDecompress (without header), but was kept for header variant).

Mirek

line of documentation would make my way to understanding much shorter.

"size" can mean anything.

Subject: Re: GZDecompress bug
Posted by mirek on Wed, 05 Aug 2009 18:21:21 GMT
View Forum Message <> Reply to Message

Sorry about that. I hope to add docs soon.

Just a little note (for future ?), if I would like to give size to
out, I would use this signature:

int GZDecompress(Stream& out, int size, Stream& in, Gate2<int, int> progress)

I believe we are quite consistent with that...

Mirek

Subject: Re: GZDecompress bug
Posted by Novo on Wed, 12 Aug 2009 02:34:44 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 05 August 2009 14:21Sorry about that. I hope to add docs soon.

Just a little note (for future ?), if I would like to give size to
out, I would use this signature:

int GZDecompress(Stream& out, int size, Stream& in, Gate2<int, int> progress)

I believe we are quite consistent with that...

Mirek

IMO the "out_size" name would be more informative.
At least you would know that it is related to "out".

I figured out how to reserve memory with StringStream.

StringStream strm;
String s;

s.Reserve(512);
strm.Open(s);

It looks like that should work, thought I'd prefer syntax below.

StringStream strm(String().Reserve(512));

Subject: Re: GZDecompress bug
Posted by mirek on Wed, 12 Aug 2009 07:25:52 GMT
View Forum Message <> Reply to Message

Well, it is perhaps a good idea to add

void StringStream::Reserve(int n)

which I did

Mirek

Subject: Re: GZDecompress bug
Posted by Novo on Fri, 14 Aug 2009 04:07:33 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 12 August 2009 03:25Well, it is perhaps a good idea to add

void StringStream::Reserve(int n)

which I did

Mirek

Thanks a lot! That will let me get rid of MemStream in my code.