Subject: Re: Filtering streams for bz2
Posted by Novo on Sun, 11 Feb 2018 20:22:31 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 09 February 2018 03:54

I have tried to resolve the issue with Gate<> FilterEof in InFilterStream. That should work. Please check trunk.

Solution with FilterEof works. Thank you.

mirek wrote on Fri, 09 February 2018 03:54
BUT, after further thinking, I believe even that is overkill. In fact, I now believe that it should work without adding anything: After encountering BZ_STREAM_END, no more data are decompressed, so no more output is produced and we get nice eof via:

```
int InFilterStream::_Term()
{
 while(ptr == rdlim && !eof)
  Fetch();
 return ptr == rdlim ? -1 : *ptr;
}
```

(because after Fetch, ptr == rdlime)

What do you think?

Mirek

Setting up of FilterEof cannot be skipped because in Fetch we have this:
```
 Stream::buffer = ptr = buffer.begin();
 rdlim = buffer.end();
```

And ptr != rdlim.

Below is my test.
```
#include <Core/Core.h>
#include <plugin/bz2/bz2.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
 enum { BUF_SIZE = 65536 };
```

```
 const Vector<String>& cmdline = CommandLine();
 if (cmdline.GetCount() < 2)
  return;
 FileIn in(cmdline[0]);
 FileOut out(cmdline[1]);
#if 0
 bz2::DecompressStream bz2(in);
 char buff[BUF_SIZE];
 while (!bz2.IsEof()) {
  const int n = bz2.Get(buff, BUF_SIZE);
  out.Put(buff, n);
 }
#endif
#if 1
 // Read only a first stream.
 bz2::DecompressStream bz2(in, false);
 char buff[BUF_SIZE];
 while (!bz2.IsEof()) {
  const int n = bz2.Get(buff, BUF_SIZE);
  out.Put(buff, n);
 }
#endif
#if 0
 bz2::CompressStream bz2(out);
 char buff[BUF_SIZE];
 while (!in.IsEof()) {
  const int n = in.Get(buff, BUF_SIZE);
  bz2.Put(buff, n);
 }
#endif
}
```

Current version of code in svn/git works fine.
Thank you again!
