Subject: Re: InFilterStream::Out() is buggy: a possible bugfix
Posted by mirek on Sat, 17 Apr 2021 04:09:28 GMT
View Forum Message <> Reply to Message

zsolt wrote on Mon, 12 April 2021 02:54The problem is, that it does not set some variables in the base class at a point.
The correct code would be:

```
void InFilterStream::Out(const void *p, int size)
{
 const byte *s = (byte *)p;
 if(todo) {
  dword sz = min(todo, (dword)size);
  memcpy8(t, s, sz);
  t += sz;
  s += sz;
  todo -= sz;
  size -= sz;
  pos += sz;
 }
 if(size) {
  int l = buffer.GetCount();
  buffer.SetCountR(l + size);
  memcpy8(buffer.begin() + l, s, size);
  Stream::buffer = ptr = buffer.begin();
  rdlim = buffer.end();
 }
 WhenOut();
}
```

The added two lines are:

```
+  Stream::buffer = ptr = buffer.begin();
+  rdlim = buffer.end();
```

I hope, it will not break any other things in code.

I found it, when I wanted to use that as a gzip input filter to read from an uncompressed stream in gzipped format:

```
class MyGzipInStream : public InFilterStream {
 Zlib        z;

public:
 void Open(Stream& in)    { Set(in, z); z.Compress(); }
 Zlib& GetZlib()          { return z; }
 MyGzipInStream()    { z.GZip().Header(); }
```

```
 ~MyGzipInStream()   { Close(); }
};
```

I wonder what is wrong with current code; this is how I believe it is supposed to work:

- there are no more data in buffer, call Fetch
- Fetch pushes data to Filter
- Filter pushes processed (e.g. decompressed) data back through Out
- Fetch then sets those variables that you suggest to set in Out

Filter is not called from anywhere else. So the only place where something could go bad is WhenOut event. Is that what is causing the problem? Are you using WhenOut?
-