Subject: Re: BUG? or Not BUG? LoadFile(filename) and then getting wrong data
Posted by Sender Ghost on Sun, 07 Aug 2011 20:47:27 GMT
View Forum Message <> Reply to Message

Hello, David.

silverx wrote on Sat, 06 August 2011 10:58
Should I use some other way to load the data into the program?

You can use FileStream directly to read file or just FileIn:

```
#include <Core/Core.h>

using namespace Upp;

inline String PrintUsage() {
 StringBuffer sb;
 sb << "Prints hex view of selected file\n\n\
Syntax:\n"
<< GetExeTitle() << " [file]\n\n\
Options:\n\
file\t\t Path to selected file\n";
 return sb;
}

CONSOLE_APP_MAIN
{
 const Vector<String>& cl = CommandLine();

 if (cl.GetCount() == 0)
 {
  SetExitCode(0);
  Cout() << PrintUsage();
  return;
 }

 String filePath(NormalizePath(cl[0]));

 if (!FileExists(filePath))
 {
  SetExitCode(1);
  Cerr() << Format("File \"%s\" doesn't exists\n", filePath);
  return;
 }

 FileIn fs;
```

```
 if (!fs.Open(filePath))
 {
  SetExitCode(1);
  Cerr() << Format("Can't read \"%s\" file\n", filePath);
  return;
 }

 const int wrapCount = 16;
 int lineCount = 0,
  columnCount = 0;

 StringBuffer text;

 while (!fs.IsEof())
 {
  if (columnCount == 0)
   Cout() << Format("%7s:", Format64Hex(lineCount++ * wrapCount));

  const int data = fs.Get();
  text.Cat(!iscntrl(data) ? char(data) : '.');
  const String hex = Format64Hex(data);
  Cout() << Format(" %2s", hex);

  if (++columnCount == wrapCount)
  {
   columnCount = 0;
   Cout() << " | " << ~text << '\n';
   text.Clear();
  }
 }

 if (text.GetCount() > 0)
 {
  const String fmt = String().Cat() << '%' << 3 * (wrapCount - columnCount + 1) << 's';
  Cout() << Format(fmt, " | ") << ~text << '\n';
 }
}
```

With OutStream implementation, while copying (for files larger than 128 bytes):

```
#include <Core/Core.h>

using namespace Upp;

inline String PrintUsage() {
 StringBuffer sb;
 sb << "Prints hex view of selected file\n\n\
```

```
Syntax:\n"
<< GetExeTitle() << " [file]\n\n\
Options:\n\
file\t\t Path to selected file\n";
 return sb;
}

class HexStream : public OutStream {
private:
 const int wrapCount;
 int columnCount,
  lineCount;
 Stream *stream;
public:
 HexStream(Stream *toStream) : stream(toStream), wrapCount(16), columnCount(0), lineCount(0)
{ }

 virtual void Out(const void *data, dword size)
 {
  StringBuffer text;

  const byte *s = (const byte *)data;

  for (dword i = 0; i < size; ++i)
  {
   if (columnCount == 0)
    *stream << Format("%7s:", Format64Hex(lineCount++ * wrapCount));

   const byte& value = s[i];
   text.Cat(!iscntrl(value) ? char(value) : '.');
   const String hex = Format64Hex(value);
   *stream << Format(" %2s", hex);

   if (++columnCount == wrapCount)
   {
    columnCount = 0;
    *stream << " | " << ~text << '\n';
    text.Clear();
   }
  }

  if (text.GetCount() > 0)
  {
   const String fmt = String().Cat() << '%' << 3 * (wrapCount - columnCount + 1) << 's';
   *stream << Format(fmt, " | ") << ~text << '\n';
  }
 }
};
```

```
CONSOLE_APP_MAIN
{
 const Vector<String>& cl = CommandLine();

 if (cl.GetCount() == 0)
 {
  SetExitCode(0);
  Cout() << PrintUsage();
  return;
 }

 String filePath(NormalizePath(cl[0]));

 if (!FileExists(filePath))
 {
  SetExitCode(1);
  Cerr() << Format("File \"%s\" doesn't exists\n", filePath);
  return;
 }

 FileIn fs;

 if (!fs.Open(filePath))
 {
  SetExitCode(1);
  Cerr() << Format("Can't read \"%s\" file\n", filePath);
  return;
 }

 HexStream stream(&Cout());
 /*const int64 count =*/
 CopyStream(stream, fs);
}
```

Edit: Added example with OutStream.
