Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Wed, 26 Aug 2020 15:12:49 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I have aancountered a curious problem while testing the Turtle/VirtualGui port.
(Note: This is tested on a Linux machine with default LANG=UTF-8 encoding. and this problem
affects both old and new Turtle packages as the VirtualGui package shares the same clipboard
code.)

The problem is GetWString and GetString do not work properly if called on a clipboard text.

TextCtrl derived classes are affected by this, as they use the same mechanism.

Sreenshot with explanation:

Now, the culprit seems to be the VirtualGui clipboard String/WString conversion code
(VirtualGui/Clip.cpp, 64-86):


```
String GetString(PasteClip& clip)
{
 GuiLock __;
 if(clip.Accept("wtext")) {
  String s = ~clip;
  return WString((const wchar *)~s, wstrlen((const wchar *)~s)).ToString(); / << Adds random
chars. Why is this though? It is already a string. For charset conversion?

        //  return ToUtf16(s).ToString();   // Seems to work fine...
        //  return s.ToWString().ToString(); // Seems to work fine...

 }
 if(clip.IsAvailable("text"))
  return ~clip;
 return Null;
}

WString GetWString(PasteClip& clip)
{
 GuiLock __;
 if(clip.Accept("wtext")) {
  String s = ~clip;
  return WString((const wchar *)~s, wstrlen((const wchar *)~s)); // Adds random chars...

        //  return ToUtf16(s);   // Seems to work fine...
```

```
        //  return s.ToWString(); // Seems to work fine...

 }
 if(clip.IsAvailable("text"))
  return (~clip).ToWString();
 return Null;
}
```

Best regards,
Oblivion