Subject: Re: Choosing the best way to go full UNICODE
Posted by mirek on Wed, 31 May 2017 12:41:28 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Wed, 31 May 2017 13:43Without RLE, I think you can get around this by positions not representing characters, but sequence starts.

But that is just implementation issue. Semantically, it is the same.

FYI, my new "String" implementation would go like this

```
class MString {
  String text;
  int character_count; // number of code units groups, e.g. graphemes
  union {
    byte *offsets; // for tiny strings
    word *woffsets; // for small strings
    dword *loffsets; // for really big strings
  };

  String operator[](int pos) { // not often used likely, more for demo
    int pos1 = GetOffset(pos);
    return text.Mid(pos1, GetOffset(pos + 1) - pos1);
  }
}
```

Quote:
The real challenge is to standardize these operations so you don't have to repeat them.

Yes. Hence MString. Note that with MString, after you perform Insert of another MString or Remove, you just update offset table, no need to iterate anything again.

Quote:
PS: the high level stuff still is StringWalker territory.

Depends on what how you define "high level" :)

For me, really high level abstraction is where you abstract from those little pesky issues and handle everything in high-level units (graphemes / characters).