Subject: Re: String improvements
Posted by rylek on Sun, 26 Feb 2006 14:39:55 GMT
View Forum Message <> Reply to Message

As concerns the exact meaning of the "start character position" in ReverseFind, I'm quite for the proposed change (to search from character with the index n, not n-1). In the matter of negative string offsets (e.g. in the Find / ReverseFind method) I must admit that I used to be a big fan of such "useful hacks", but over the years I moved a lot in Mirek's direction and I'm more and more willing to sacrifice a line or two of code if I gain a cleaner and more intuitive interface.

I believe the negative string offsets are a good example, a very similar situation arises also in the container interfaces. If we decide to implement negative character offsets for Find / ReverseFind, it seems to me counter-intuitive not to implement them everywhere else, like in the Left / Right / Mid methods for starters. After that we'll start to consider using the same syntax for the [] operator, like using s[-1] to address the last string character, and this will be the moment when the real trouble begins, because the necessary runtime check will potentially cost us some performance, with the dubious by-effect of introducing a feature which is quite as likely to come handy in certain situations as it's error-prone in the sense that it may hide erroneous code design (like searching a string which is not long enough) in a relatively tricky way.

Come to it, I don't have anything against a new method to search a terminal portion of the string. I'm afraid that my experience and opinion in this matter is a bit similar to Mirek's in that I use even the existing Find / ReverseFind methods relatively seldom (by a simple check I've just found a single occurrence of ReverseFind in all my "commercial" project source code). However I'm not very fond of this "statistical programming" (I don't use a feature, ipso facto it is not necessary or useful) and a few practical examples are quite likely to convince me that we do good to the U++ String interface by enriching it with a few additional search functions (by the way, the one I really do miss is searching a string for a substring, which, if I'm not mistaken, is still missing).

Regards

Tomas