Subject: Re: substring find
Posted by mirek on Mon, 06 Feb 2006 10:28:09 GMT
View Forum Message <> Reply to Message

No, small string optimization means that for small strings, you keep data inside String object, something like:

```
String {
  struct Large {
     const char *ptr;
     ....
  }
  union {
    char data[16]
    Large large_string;
  }
};
```

If you think about the issue, unshared reference counted string (and per my research, most reference counted strings are unshared) has 16 bytes overhead (4 bytes for pointer in String, 4 bytes reference count, 4 bytes length of string, 4 bytes allocation length). Then 70% of Strings has len < 15. Means, in 70% cases SSO will store the String "for free" when compared to current implementation. In remaining 30%, SSO will just use those 16 bytes to store pointer/length/alloc previously stored in shared string (storing there just reference count). Also, you will avoid alloc/free, interlocked increment/decrement etc...

Mirek