

Thanks.

I could not resist but to try this little common-case optimization:

```
template <class B>
int AString<B>::FindFirstOf(int len, const tchar *s, int from) const
{
    ASSERT(from >= 0 && from <= GetLength());
    const tchar *ptr = B::Begin();
    const tchar *e = End();
    const tchar *se = s + (len * sizeof(tchar));
    if((s[0] & s[1]) != 0) {
        if(s[2] == 0) {
            __BREAK__;
            tchar c1 = s[0];
            tchar c2 = s[1];
            for(const tchar *bs = ptr + from; bs < e; bs++) {
                tchar ch = *bs;
                if(ch == c1 || ch == c2)
                    return (int)(bs - ptr);
            }
            return -1;
        }
        if(s[3] == 0) {
            tchar c1 = s[0];
            tchar c2 = s[1];
            tchar c3 = s[2];
            for(const tchar *bs = ptr + from; bs < e; bs++) {
                tchar ch = *bs;
                if(ch == c1 || ch == c2 || ch == c3)
                    return (int)(bs - ptr);
            }
            return -1;
        }
        if(s[4] == 0) {
            tchar c1 = s[0];
            tchar c2 = s[1];
            tchar c3 = s[2];
            tchar c4 = s[3];
            for(const tchar *bs = ptr + from; bs < e; bs++) {
                tchar ch = *bs;
                if(ch == c1 || ch == c2 || ch == c3 || ch == c4)
                    return (int)(bs - ptr);
            }
        }
    }
```

```
    }  
    return -1;  
  }  
}  
for(const tchar *bs = ptr + from; bs < e; bs++)  
  for(const tchar *ss = s; ss < se; ss++)  
    if(*bs == *ss)  
      return (int)(bs - ptr);  
return -1;  
}
```

Seems to be 2x faster for these "common cases"...

Mirek
