Subject: Some 'missing' string functions
Posted by mdelfede on Fri, 25 Jan 2008 11:37:12 GMT

View Forum Message <> Reply to Message

Porting a small app that uses std::string to Upp::String I noticed that many build-in string functions are not implemented in Upp::String.

In particular, some character-locating functions (find_first_not_of(), find_first_of(), find_last_not_of().....) functions are quite useful sometimes.
Also Compare() function is missing some way to compare parts of the string. It's easy to implement with Mid() + Compare, but it involves a string copy, so it's slow. Some sort of

String::Compare(aString, start, len)

could be useful and much faster than taking the substring and comparing it.

Last but not least, the ReverseFind() function can find only a char, not a string inside a given string, as rfind() function in std.

Ciao

Max

Subject: Re: Some 'missing' string functions
Posted by mirek on Fri, 25 Jan 2008 22:05:45 GMT

View Forum Message <> Reply to Message

mdelfede wrote on Fri, 25 January 2008 06:37Porting a small app that uses std::string to Upp::String I noticed that many build-in string functions are not implemented in Upp::String.

In particular, some character-locating functions (find_first_not_of(), find_first_of(), find_last_not_of().....) functions are quite useful sometimes.

Quote:
Also Compare() function is missing some way to compare parts of the string. It's easy to implement with Mid() + Compare, but it involves a string copy, so it's slow. Some sort of

String::Compare(aString, start, len)

Should have start1 and start2 IMO.

Anyway, I use memcmp in such cases usually...

Quote:
Last but not least, the ReverseFind() function can find only a char, not a string inside a given string, as rfind() function in std.

OK.

Mirek

Subject: Re: Some 'missing' string functions
Posted by mirek on Fri, 25 Jan 2008 22:09:24 GMT
View Forum Message <> Reply to Message

Quote:
find_first_not_of(), find_first_of(), find_last_not_of()

BTW, do you know whether STL somehow optimizes these?

I am rather thinking about adding "Filter" variant here...

void FindFirst(int (*filter)(int c), int from = 0)

Mirek

Subject: Re: Some 'missing' string functions
Posted by mdelfede on Sat, 26 Jan 2008 13:06:25 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 25 January 2008 23:09Quote:
find_first_not_of(), find_first_of(), find_last_not_of()

BTW, do you know whether STL somehow optimizes these?

I am rather thinking about adding "Filter" variant here...

void FindFirst(int (*filter)(int c), int from = 0)

you'd need also

void FindFirst(int (*filter)(char *s), int from = 0)

as std:: has also such functions. For example :

int i = s.find_first_not_of("ab", 5)

gives the index of first character in s starting from index 5 which is neither 'a' nor 'b'.
That's useful to skip some character in a line, used for example in Astyle to skip spaces and tabs :

int i = s.find_first_not_of(" \t", 5)

The filter idea is not bad at all, and you could also add some wrapper for simpler cases.

BTW, another stuf I think is missing is a constant that is returned when no match is found. std:: uses string::npos, which should have a value of -1 but makes code reading easer.

Ciao

Max

Subject: Re: Some 'missing' string functions
Posted by mirek on Sat, 26 Jan 2008 13:24:04 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Sat, 26 January 2008 08:06
BTW, another stuf I think is missing is a constant that is returned when no match is found. std:: uses string::npos, which should have a value of -1 but makes code reading easer.

Ciao

Max

Well, I do not know. <0 is a common way for U++ to say "not found", used everywhere.

Mirek

Subject: Re: Some 'missing' string functions
Posted by mdelfede on Sat, 26 Jan 2008 13:49:41 GMT
View Forum Message <> Reply to Message

luzr wrote on Sat, 26 January 2008 14:24mdelfede wrote on Sat, 26 January 2008 08:06

BTW, another stuf I think is missing is a constant that is returned when no match is found. std:: uses string::npos, which should have a value of -1 but makes code reading easer.

Ciao

Max

Well, I do not know. <0 is a common way for U++ to say "not found", used everywhere.

yes, you're right... that's because Upp uses 'int' as string index, where std:: uses size_t which is unsigned... because of that I've got some problem translating the code to Upp.
I'll change all indexes to 'int' in code, and put error checking as '< 0' instead '== -1'.

Ciao

Max

Subject: Re: Some 'missing' string functions
Posted by phirox on Fri, 06 Jun 2008 21:21:22 GMT
View Forum Message <> Reply to Message

I really needed a similar thing as find_first_of, and found this topic. It seems there still isn't an implementation or the suggested Filter method, so I wrote my own:

It is modelled after Find and should be added to String.h and AString.hpp. I tested it with String and WString, and couldn't find any bugs. A FindFirstNotOf, FindLastOf, etc. shouldn't be so hard to copy from this model.


```
int   FindFirstOf(int len, const tchar *s, int from) const;
int   FindFirstOf(const tchar *s, int from = 0) const;
int   FindFirstOf(const String& s, int from = 0) const   { return FindFirstOf(s.GetCount(), ~s, from);
}

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));
 for(const tchar *bs = ptr + from; bs < e; bs++)
  for(const tchar *ss = s; ss < se; ss++)
   if(*bs == *ss)
```

Page 4 of 10 ---- Generated from   U++ Forum

```
    return (int)(bs - ptr);
 return -1;
}
template <class B>
int AString<B>::FindFirstOf(const tchar *s, int from) const
{
 return FindFirstOf(strlen__(s), s, from);
}
```

Subject: Re: Some 'missing' string functions
Posted by mirek on Sat, 07 Jun 2008 14:12:44 GMT

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

Subject: Re: Some 'missing' string functions
Posted by mr_ped on Mon, 09 Jun 2008 06:32:15 GMT
View Forum Message <> Reply to Message

I'm too lazy to check the whole source, so maybe these are stupid questions, but I have to ask anyway:
 const tchar *ptr = B::Begin();
 const tchar *e = End();

Why just "End();" without B::, when "B::Begin();" is used? (feels unclean to me)

    __BREAK__;

  ... someone was debugging something.

Subject: Re: Some 'missing' string functions
Posted by hans on Mon, 09 Jun 2008 08:51:31 GMT
View Forum Message <> Reply to Message

The function has a bug, because in

 const tchar *se = s + (len * sizeof(tchar)); :

the multiply with sizeof(tchar) is nonsense, pointer arithmetic
is defined to work with object size already;

And the function makes too many assumptions too, namely it accesses memory after len, which
may work for String objects,
but not in general case:

A valid call may be

char* s= new char('A');
string.FindFirstOf(1, s, 0);

So I would suggest to change this function to

```
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;
 if(len == 1) {
  tchar c1 = s[0];
  for(const tchar *bs = ptr + from; bs < e; bs++) {
   if(*bs == c1)
    return (int)(bs - ptr);
  }
  return -1;
 }
 if(len == 2) {
   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(len == 3) {
   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(len == 4) {
  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;
}
```

Regards,
Hans

Subject: Re: Some 'missing' string functions
Posted by mirek on Mon, 09 Jun 2008 12:27:03 GMT
View Forum Message <> Reply to Message

Thanks, this indeed is much more correct.

Mirek

Subject: Re: Some 'missing' string functions
Posted by mirek on Mon, 09 Jun 2008 12:28:32 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Mon, 09 June 2008 02:32
[/code]
Why just "End();" without B::, when "B::Begin();" is used? (feels unclean to me)

__BREAK__;

  ... someone was debugging something.

Yeah, I was checking what compiler produces there Anyway, this one was removed before commiting.

Mirek

Subject: Re: Some 'missing' string functions
Posted by captainc on Thu, 28 Aug 2008 08:13:31 GMT
View Forum Message <> Reply to Message

I'm trying to use FindFirstOf and I am getting this error when compiling:
c:\program files\upp-svn\uppsrc\core\AString.hpp(114) : error C2039: 'End' : is not a member of 'Upp::String0'
 C:\Program Files\upp-svn\uppsrc\Core/String.h(133) : see declaration of 'Upp::String0'
     c:\program files\upp-svn\uppsrc\core\AString.hpp(111) : while compiling class template member function 'int Upp::AString<B>::FindFirstOf(int,const char *,int) const'
     with
     [
         B=Upp::String0
     ]
     C:\Program Files\upp-svn\uppsrc\Core/Topt.h(205) : see reference to class template instantiation 'Upp::AString<B>' being compiled
     with
     [
         B=Upp::String0
     ]
     C:\Program Files\upp-svn\uppsrc\Core/String.h(281) : see reference to class template instantiation 'Upp::Moveable<T,B>' being compiled
     with
     [
         T=Upp::String,
         B=Upp::AString<Upp::String0>
     ]

Focus was brought to this section of code:
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 = B::End();
 const tchar *se = s + len;
 if(len == 1) {
  tchar c1 = s[0];

```
  for(const tchar *bs = ptr + from; bs < e; bs++) {
   if(*bs == c1)
    return (int)(bs - ptr);
  }
  return -1;
```
My source line is:
String whitespace(" \n\t");
pos = _title.FindFirstOf(whitespace);

Subject: Re: Some 'missing' string functions
Posted by mirek on Thu, 28 Aug 2008 13:19:25 GMT
View Forum Message <> Reply to Message

Confirmed & fixed.

Quick fix, add:

```
 const char *End() const     { return Begin() + GetLength(); }
```

Mirek