



[General Programming](#) » [String handling](#) » [Text Conversion](#)

Guide to BSTR and C String Conversions

By [Robert Pittenger](#)

An article on converting to/from C strings and various VB BSTR string types

VC6, VC7, VC7.1Win2K, WinXP, Win2003, MFC, Dev

Posted: **21 Aug 2003**

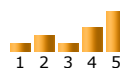
Views: **185,258**

Bookmarked: **65 times**

24 votes for this article.



Popularity: 5.23 Rating: 3.79 out of 5



 [Download demo project - 11 Kb](#)

Introduction

One of the confusing aspects of Windows programming is managing the conversion of Visual Basic style strings to/from C language style strings. It isn't that it is so difficult, it is just difficult to remember the details, it is usually not done often, the MSDN documentation so voluminous that it is difficult to find answers to your questions. But the worst part is that you could perform some typecast that compiles fine, but doesn't work the way you expect. This results code that doesn't work, and the bugs are hard to track down. After some experience, you learn to make sure your string conversions are doing what you expect.

C strings are arrays of characters terminated by a NULL character. Visual Basic strings differ in that the length of the string preceded the characters in the string. So a VB string knows it's own length. In addition, all VB strings are Unicode (16 bits per character).

String Types

BSTR/C String conversions are required if:

- You are doing COM programming in C/C++
- You are writing multiple language applications, such as C++ DLL's accessed by Visual Basic applications.

C Language String Types and Classes

This article deals with the following C/MFC/ATL string types:

- `char/wchar/TCHAR` -- The C strings for ANSI and Unicode
- `CString` -- The C++/MFC class wrapper for C strings
- `BSTR` -- The Visual Basic string type
- `_bstr_t` -- A C++ class wrapper for the Visual Basic string type
- `CComBSTR` -- Yet another C++ class wrapper for the Visual Basic string type used predominately in ATL code

Demo Project

The demo project is just an MFC dialog based application with buttons for each type of conversion. It is built using VC++ 6.0. It uses a couple of support functions you may find helpful:

```
BSTR GetBSTR()
{
    _bstr_t bstr1(_T("This is the test string.));

    BSTR bstr;

    bstr = bstr1.copy();

    return bstr;
}
```

```
CComBSTR GetComBSTR()
{
    CComBSTR bstr("This is the test string.");

    return bstr;
}
```

```
void CVbsDlg::ShowBSTR(BSTR bstr)
{
    _bstr_t bstrStart(bstr);

    CString s;

    s.Format(_T("%s"), (LPCTSTR)bstrStart);

    AfxMessageBox(s);
}
```

Conversions

So let's get to it. Here are the conversion techniques:

Converting BSTR to _bstr_t

```
// BSTR to _bst_t

BSTR bstrStart = GetBSTR();

// use the constructor
_bstr_t bstrFinal(bstrStart);

ShowBSTR(bstrFinal);

// Use the = operator
bstrFinal = bstrStart;

ShowBSTR(bstrFinal);
```

Converting a _bstr_t to BSTR

You may want to get a **BSTR** from a `_bstr_t` class.

```
// _bstr_t to BSTR
_bstr_t bstrStart(_T("This is the test string.));

BSTR bstrFinish;

// use _bstr_t::copy member function
bstrFinish = bstrStart.copy();

ShowBSTR(bstrFinish);

// use = operator
bstrFinish = bstrStart;

ShowBSTR(bstrFinish);
```

Converting a CComBSTR to BSTR

You may want to get a **BSTR** from a **CComBSTR** class.

```
// CComBSTR to BSTR
CComBSTR bstrStart(_T("This is the test string.));

BSTR bstrFinish;

// use the = operator
bstrFinish = bstrStart;

ShowBSTR(bstrFinish);

// use the Copy member function
bstrFinish = bstrStart.Copy();

ShowBSTR(bstrFinish);
```

Converting _bstr_t to CComBSTR

```
// _bstr_t to CComBSTR
_bstr_t bstrStart(_T("This is the test string.));

CComBSTR bstrFinish;

bstrFinish.AppendBSTR(bstrStart);

ShowBSTR(bstrFinish);
```

Converting BSTR to C String

(Note :- conversion that only works in Unicode)

```
// BSTR to C String

BSTR bstrStart;

bstrStart = GetBSTR();

TCHAR szFinal[255];

// direct conversion from BSTR to LPCTSTR only works in Unicode
_stprintf(szFinal, _T("%s"), (LPCTSTR)bstrStart);
AfxMessageBox(szFinal);

_bstr_t bstrIntermediate(bstrStart); // convert to _bstr_t
CString strFinal;

// you have to go through _bstr_t to have it work in ANSI and Unicode
_stprintf(szFinal, _T("%s"), (LPCTSTR)bstrIntermediate);

// Or using MFC

strFinal.Format(_T("%s"), (LPCTSTR)bstrIntermediate);

AfxMessageBox(strFinal);
```

Converting `_bstr_t` to C String

(this works in both ANSI and Unicode)

```
_bstr_t bstrStart(_T("This is the test string.));
TCHAR szFinal[255];

_stprintf(szFinal, _T("%s"), (LPCTSTR)bstrStart);

AfxMessageBox(szFinal);
```

Converting `CComBSTR` to `LPCTSTR`

(not possible, must go through `_bstr_t`)

```
// CComBSTR to C String
CComBSTR bstrStart("This is the test string.");

_bstr_t bstrIntermediate(bstrStart);

TCHAR szFinal[255];

_stprintf(szFinal, _T("%s"), (LPCTSTR)bstrIntermediate);

AfxMessageBox(szFinal);
```

Converting `LPCTSTR` to `_bstr_t`

(Use a constructor or = operator)

```
// LPCTSTR to _bstr_t
LPCTSTR szStart = _T("This is the text string");

// Use the constructor
_bstr_t bstrFinal(szStart);

ShowBSTR(bstrFinal);

// or use = operator
bstrFinal = szStart;

ShowBSTR(bstrFinal);
```

Converting LPCTSTR to CComBSTR

Use a constructor or `CComBSTR::Append` function

```
// LPCTSTR to CComBSTR

// Use a constructor
LPCTSTR szStart = _T("This is the text string");

// Use the constructor
CComBSTR bstrFinal(szStart);

ShowBSTR(bstrFinal);

// Or use the Append function
bstrFinal.Empty();
bstrFinal.Append(szStart);

ShowBSTR(bstrFinal);
```

Conclusion

Well I tested all of the conversion in the demo project. If you need to try others, download the demo for easy modification. I am sure I will hear if there are any mistakes!

License

This article has no explicit license attached to it but may contain usage terms in the article text or the download files themselves. If in doubt please contact the author via the discussion board below.

A list of licenses authors might use can be found [here](#)


About the Author

Robert Pittenger

Location:  United States

Member

Discussions and Feedback

 **15 messages** have been posted for this article. Visit <http://www.codeproject.com/KB/string/bstrsproject1.aspx> to post and view comments on this article, or click [here](#) to get a print view with messages.

[PermaLink](#) | [Privacy](#) | [Terms of Use](#)
Last Updated: 21 Aug 2003
Editor: [Nishant Sivakumar](#)

Copyright 2003 by Robert Pittenger
Everything else Copyright © [CodeProject](#), 1999-2009
Web11 | [Advertise on the Code Project](#)