What is Unicode?

In this article we'll

- take a look at Unicode
- answer a few basic questions
- provide a function to convert an ASCII sequence into a Unicode decimal (UD, for short) and another to convert the UD back into an ASCII sequence
- include links to Unicode resources and more information

So what is Unicode? In the words of The Unicode Consortium:

- **Unicode provides a unique number for every character,**
  - **no matter what the platform,**
  - **no matter what the program,**
  - **no matter what the language.**

Unicode's goal is to develop a scheme of character encodings that is, essentially, language independant. How can they do that?

1. By encoding the scripts (written characters) of the world's languages. They have already encoded all of the world's major languages and there are currently over 100,000 characters in the database.
2. By widespread acceptance of their standard. The Unicode standard has been adopted by Apple, Sun, Microsoft, HP, IBM, Oracle and many others. Unicode is required by modern technologies such as XML, Java, JavaScript, Microsoft's .NET framework, LDAP, etc.

An example of an Unicode character is ♫. The beamed eighth notes is the Unicode character decimal 9835, or Unicode character hex 266B. The ♫ is not an image, but a character. As with any character, the ♫ can be formatted.

The Unicode character decimal 9835 = ♫

Unicode, in intent, encodes the underlying characters rather than the variant glyphs (renderings) for characters. In text processing, Unicode takes the role of providing a unique code point — a number, not a glyph — for each character. In other words, Unicode represents a character in an abstract way and leaves the visual rendering (size, shape, font or style) to other software, such as a web browser or word processor.

ASCII to Unicode

```
' Asc2Uni -- Converts a string of ASCII bytes into a Unicode Decimal (
ud)
Function Asc2Uni(bytes$)
  z = Asc(Left$(bytes$,1))
  For a = 1 To Min(6, Len(bytes$))
    c(a) = Asc(Mid$(bytes$, a, 1))
  Next a
  Select Case
    Case z>=  0 And z<=127
      ud = c(1)
    Case z>=192 And z<=223
      ud = (c(1)-192)*64 + (c(2)-128)
    Case z>=224 And z<=239
      ud = (c(1)-224)*4096 + (c(2)-128)*64 + (c(3)-128)
    Case z>=240 And z<=247
      ud = (c(1)-240)*262144 + (c(2)-128)*4096 + (c(3)-128)*64 + (c(4
)-128)
    Case z>=248 And z<=251
      ud = (c(1)-248)*16777216 + (c(2)-128)*262144 + (c(3)-128)*4096 _
         + (c(4)-128)*64 + (c(5)-128)
    Case z>=252 And z<=253
      ud = (c(1)-252)*1073741824 + (c(2)-128)*16777216 + (c(3)-128)*
262144 _
         + (c(4)-128)*4096 + (c(5)-128)*64 + (c(6)-128)
    Case Else
      ud = -1  ' not a Unicode byte sequence
  End Select
  Asc2Uni = ud
End Function
```

Unicode to ASCII

```
' Uni2Asc -- Converts a Unicode Decimal (ud) into a string of ASCII by
tes
Function Uni2Asc$(ud)
    bs$ = ""
    Select Case
      Case ud>=0 And ud<=127
        bs$ = Chr$(ud)
      Case ud>=128 And ud<=2047
        b1 = 192 + Int(ud / 64)
        b2 = 128 + (ud Mod 64)
        bs$ = Chr$(b1);Chr$(b2)
      Case ud>=2048 And ud<=65535
```

```
        b1 = 224 + Int(ud / 4096)
        b2 = 128 + (Int(ud / 64) Mod 64)
        b3 = 128 + Int(ud Mod 64)
        bs$ = Chr$(b1);Chr$(b2);Chr$(b3)
      Case ud>=65536 And ud<=2097151
        b1 = 240 + Int(ud / (64^3))
        b2 = 128 + (Int(ud / 4096) Mod 64)
        b3 = 128 + (Int(ud / 64) Mod 64)
        b4 = 128 + Int(ud Mod 64)
        bs$ = Chr$(b1);Chr$(b2);Chr$(b3);Chr$(b4)
      Case ud>=2097152 And ud<=67108863
        b1 = 248 + Int(ud / 16777216)
        b2 = 128 + (Int(ud / 262144) Mod 64)
        b3 = 128 + (Int(ud / 4096) Mod 64)
        b4 = 128 + (Int(ud / 64) Mod 64)
        b5 = 128 + (ud Mod 64)
        bs$ = Chr$(b1);Chr$(b2);Chr$(b3);Chr$(b4);Chr$(b5)
      Case ud>=67108864 And ud<=2147483647
        b1 = 252 + Int(ud / 1073741824)
        b2 = 128 + (Int(ud / 16777216) Mod 64)
        b3 = 128 + (Int(ud / 262144) Mod 64)
        b4 = 128 + (Int(ud / 4096) Mod 64)
        b5 = 128 + (Int(ud / 64) Mod 64)
        b6 = 128 + (ud Mod 64)
        bs$ = Chr$(b1);Chr$(b2);Chr$(b3);Chr$(b4);Chr$(b5);Chr$(b6)
    End Select
    Uni2Asc$ = bs$
End Function
```

Using the Functions

```
' Using our musical note example:
  note$ = Uni2Asc$(9835)
  print "The unicode character decimal 9835 = "; note$
  print note$; " is unicode character decimal number "; Asc2Uni(note$)
End
```

Liberty BASIC's MainWin does not support Unicode, so what you will see in the MainWin will be three old-school HI-ASCII characters --

1. the lowercase a with ^ above,
2. the small raised trademark sign (TM) and
3. the small double left angle

These are ASCII 226 153 171, or in Unicode lingo, U+00E2 U+0099 U+00AB. If you copy and paste the output into your word processor, or even WordPad, you'll see the musical note. You may need to set an option under the View menu to UTF

Unicode Resources

Alan Wood's Unicode Resources Is one of the best places to start. Its full of practical information and updates on using and developing in Unicode. His font page is the best place on the net to get Unicode fonts for MS-Windows.

Latest version of the Unicode Standard is Version 5.1.0.
The documentation for Unicode 5.1.0 is located at:
http://www.unicode.org/versions/Unicode5.1.0/

Unicode Character Database (UCD) is the set of files that define the Unicode character properties and internal mappings. As of Version 4.1.0. each version of the UCD is complete, so users do not need to assemble files from previous versions to have a complete set of files for a version.

Documentation for the Unicode Character Database (UCD) is located at:
http://www.unicode.org/reports/tr44/

Current version of the UCD is always located on the Unicode Web site at:
http://www.unicode.org/Public/UNIDATA/

XML data files Starting with Version 5.1.0, a set of XML data files are released with each version of the UCD. They make it possible to deal with UCD property data using standard XML parsing tools, instead of the specialized parsing required for the various individual data files of the UCD.

If you are looking for a certain character the code charts is a good place to search. They are organized into groups of related languages.

Fonts

To see these characters you'll need a font capable of rendering them. The Wikipedia Unicode Typeface page has a chart showing the code points supported by several unicode fonts. If you have Microsoft Office 2002 or later, you should already have Arial Unicode MS. The rest of us may want to visit Alan Wood's fonts for Windows and download Bitstream CyberBit (freeware) or Code2000 (shareware, $5), which cover the Basic Multilingual Plane (BMP, for short) as well as any font.

http://unicode.coeurlumiere.com/
This site will show you the BMP (0 to 65535 or, in Unicode, U+0000 to U+FFFF) with 4096 characters

per page. Some will be blank, however, as not every number is a valid code point. It's useful for testing that massive font you just installed. :-)

Programming Blogs

Joel Spolsky is a software developer in New York City.
Tim Bray is a self-described "General-Purpose Web Geek" who works at Sun Microsystems. This is only one of several articles he's written on UTF and related topics.

-
harmony Nov 1, 2008

P.S. This article is not complete.
If you have something to add / edit / alter, feel free to do so.