参考：<http://www.cnblogs.com/sunshore/p/3548154.html>

今天在C#里试了下它使用的default encoding:（我的电脑是中文的windows7)

Encode defaultEncoding = Encode.Default;

Console.WriteLine("the system default encoding: ",defaultEncoding.ToString());

得出的结果是： DBCSCodePageEncoding, 于是对于这种编码很是奇怪，在网上搜寻了一翻，有所收获。

1. C#里参 Encode.Defalut的解释是： Gets an encoding for the operation system's current ANSI code page.

2.来自百度百科

DBCS：double-byte character set

最初的128个代码是ASCII,较高的128个代码中的某些总是跟随著第二个[位元组](http://baike.baidu.com/view/572354.htm" \t "_blank)(位,Bit)。这两个位元组(位,Bit)一起（称作首位元组和跟随位元组）定义一个字元，通常是一个复杂的象形文字。

虽然中文、日文和韩文共用一些相同的象形文字，但显然这三种语言是不同的，而且经常是同一个象形文字在三种不同的语言中代表三件不同的事。Windows支持四个不同的[双位元组字元集](http://baike.baidu.com/view/555393.htm" \t "_blank)：内码表932（日文）、936（简体中文）、949（韩语）和950（繁体汉字）（所谓的内码表，根据我的理解就是windows 里的code page）。只有为这些国家（地区）生产的Windows版本才支持DBCS。明白Unicode和DBCS之间的区别很重要。Unicode使用（特别在C程式设计语言环境里）[宽字元](http://baike.baidu.com/view/555413.htm" \t "_blank)集。Unicode中的每个字元都是16位元宽而不是8位元宽。在Unicode中，没有单单使用8位元数值的意义存在。相比之下，在双位元组字元集中我们仍然处理8位元数值。有些[位元组](http://baike.baidu.com/view/572354.htm" \t "_blank)自身定义字元,而某些位元组则显示需要和另一个位元组共同定义一个字元。

3.博客上有个人写了个C#程序得到了.net支持的所有的编码集：

<http://www.cnblogs.com/skyivben/archive/2011/01/13/1935086.html>

从文章中给的表可以看出DBCSCodePageEncoding包括中文，日文，朝鲜文及中文繁体的编码方式。

4.在百度文库里可以找到ASCII, DBCS, Unicode三种编码的区别：

<http://wenku.baidu.com/link?url=Nn7712MBnorET9woMsTbkFAYkDKHbsb7vubHFN8JorB1qCKVYp5y_ujzvzuyAMsJhjAFJS5WxE0QPVp1DH-CpQk8h9lVB_hM5Uq9jyNPCim>

5.还有一个新发现，原来在Linux下也可以支持C#语言的开发，

<http://www.cnblogs.com/skyivben/archive/2009/11/19/1605821.html>

笔记：

——————————

string testStr = "DBCSCodePageEncoding, 于是对于这种编码很是奇怪,参考：http://www.cnblogs.com/sunshore/p/3548154.html";

byte[] u8Bytes = Encoding.UTF8.GetBytes(testStr);

string decodeStr = Encoding.UTF8.GetString(u8Bytes);

Console.WriteLine(decodeStr);

if (Encoding.GetEncoding(65001) == Encoding.UTF8)

{

Console.WriteLine("pagingEncoding-65001==utf-8");

}

Console.WriteLine("Encoding.Default:DisplayName:{1}--CodePage:{0}", Encoding.Default.EncodingName,Encoding.Default.CodePage);

foreach (EncodingInfo info in Encoding.GetEncodings())

{

Console.WriteLine("{0} ==> {1}",info.CodePage,info.DisplayName);

}

Encoding.GetEncodings()可以得到所有的编码页（codePage）,每一个编码页对应一个编码，如65001对应编码UTF-8，65000对应UTF-7等

Encoding.GetEncoding(65001)即可获得Encoding.UTF8

Encoding.Default指的是本地操作系统的，如中文的可能为GB2312