[**【转】Vim 字符编码转换**](http://www.cnblogs.com/jaiho/archive/2011/08/24/2056375.html)

如果只是要简单的转换文件编码，打开后    
       :set fileencodings=utf-8 ，然后 w （存盘）一下即可转化为 utf8 格式，  
      :set fileencodings=cp936  ，然后 w （存盘）一下即可转化为 ANSI 格式。

set fileformat=unix  
  
  
[Vim对中文编码的支持](http://my.donews.com/sawfish/2007/04/07/vim%e5%af%b9%e4%b8%ad%e6%96%87%e7%bc%96%e7%a0%81%e7%9a%84%e6%94%af%e6%8c%81/)

1、支持中文编码的基础

[Vim](http://www.vim.org/)要更好地支持中文编码需要两个特性：+multi\_byte和+iconv，可以用**|:version|**命令检查当前使用的Vim是否支持，否则的话需要重新编译。

2、影响中文编码的设置项

Vim中有几个选项会影响对多字节编码的支持：

* encoding（enc）：encoding是Vim的内部使用编码，encoding的设置会影响Vim内部的Buffer、消息文字等。在 Unix环境下，encoding的默认设置等于locale；Windows环境下会和当前代码页相同。在中文Windows环境下encoding的 默认设置是cp936（GBK）。
* fileencodings（fenc）：Vim在打开文件时会根据fileencodings选项来识别文件编码，fileencodings可以同时设置多个编码，Vim会根据设置的顺序来猜测所打开文件的编码。
* fileencoding（fencs） ：Vim在保存新建文件时会根据fileencoding的设置编码来保存。如果是打开已有文件，Vim会根据打开文件时所识别的编码来保存，除非在保存时重新设置fileencoding。
* termencodings（tenc）：在终端环境下使用Vim时，通过termencoding项来告诉Vim终端所使用的编码。

3、Vim中的编码转换

Vim内部使用iconv库进行编码转换，如果这几个选项所设置的编码不一致，Vim就有可能会转换编码。打开已有文件时会从文件编码转换到 encoding所设置的编码；保存文件时会从encoding设置的编码转换到fileencoding对应的编码。经常会看到Vim提示**[已转换]**，这是表明Vim内部作了编码转换。终端环境下使用Vim，会从termencoding设置的编码转换到encoding设置的编码。

可以用**|:help encoding-values|**列出Vim支持的所有编码。

4、具体应用环境的设置

* 只编辑GBK编码的文件

set fileencodings=cp936  
set fileencoding=cp936  
set encoding=cp936

* 只编辑UTF-8编码的中文文件

set fileencodings=utf-8  
set fileencoding=utf-8  
set encoding=cp936 或者 set encoding=utf-8

* 同时支持GBK和UTF-8编码

set fileencodings=ucs-bom,utf-8,cp936  
set fileencoding=utf-8  
set encoding=cp936 或者 set encoding=utf-8

* 如果在终端环境下使用Vim，需要设置termencoding和终端所使用的编码一致。例如：

set termencoding=cp936 或者 set termencoding=utf-8

Windows记事本编辑UTF-8编码文件时会在文件头上加上三个字节的BOM：EFBBBF。如果fileencodings中设 置ucs-bom的目的就是为了能够兼容用记事本编辑的文件，不需要的话可以去掉。Vim在保存UTF-8编码的文件时会去掉BOM。去掉BOM的最大好 处是在Unix下能够使用cat a b>c来正确合并文件，这点经常被忽略。

5、FAQ

1. 为什么在Vim中一次只能删除半个汉字？

因为encoding设置错误，把encoding设置为cp936就可以解决此问题。在Unix环境下Vim会根据locale来设置默认的encoding，如果没有正确设置locale并且没有设置encoding就会一次只能删除半个汉字。

1. VIM为什么不能输入繁体字？

把euc-cn或者GB2312改为cp936就可以了。euc-cn是GB2312的别名，不支持繁体汉字。cp936是GBK的别名，是GB2312的超集，可以支持繁体汉字。

1. VIM为什么提示不能转换？

因为在编译Vim时没有加入iconv选项，重新编译Vim才能解决。

1. 如何打开一个GBK编码的文件并另存为UTf-8编码？

保存文件时运行命令**|:set fileencoding=utf-8|**就可以了。

vim里面的编码主要跟三个参数有关：enc(encoding), fenc(fileencoding)和fencs(fileencodings)  
  
其 中fenc是当前文件的编码，也就是说，一个在vim里面已经正确显示了的文件(前提是你的系统环境跟你的enc设置匹配)，你可以通过改变 fenc后再w来将此文件存成不同的编码。比如说，我:set fenc=utf-8然后:w就把文件存成utf-8的了，:set fenc=gb18030再:w就把文件存成gb18030的了。这个值对于打开文件的时候是否能够正确地解码没有任何关系。  
  
fencs就是用来在打开文件的时候进行解码的猜测列表。文件编码没有百分百正确的判断方法，所以vim只能猜测文件编码。比如我的vimrc里面这个的设置是  
  
set fileencodings=utf-8,gb18030,utf-16,big5  
  
所 以我的vim每打开一个文件，先尝试用utf-8进行解码，如果用utf-8解码到了一半出错(所谓出错的意思是某个地方无法用utf-8正确地解码)， 那么就从头来用gb18030重新尝试解码，如果gb18030又出错(注意gb18030并不是像utf-8似的规则编码，所以所谓的出错只是说某个编 码没有对应的有意义的字，比如0)，就尝试用utf-16，仍然出错就尝试用big5。这一趟下来，如果中间的某次解码从头到尾都没有出错，那么 vim就认为这个文件是这个编码的，不会再进行后面的尝试了。这个时候，fenc的值就会被设为vim最后采用的编码值，可以用:set fenc?来查看具体是什么。  
  
当然这个也是有可能出错的，比如你的文件是gb18030编码的，但是实际上只有一两个字符是中文，那么有可能他们正好也能被utf-8解码，那么这个文件就会被误认为是utf-8的导致错误解码。  
  
至 于enc，其作用基本只是显示。不管最后的文件是什么编码的，vim都会将其转换为当前系统编码来进行处理，这样才能在当前系统里面正确地显示出来，因此 enc就是干这个的。在windows下面，enc默认是cp936，这也就是中文windows的默认编码，所以enc是不需要改的。在 linux下，随着你的系统locale可能设为zh\_CN.gb18030或者zh\_CN.utf-8，你的enc要对应的设为gb18030或者 utf-8(或者gbk之类的)。  
  
最后再来说一下新建空文件的默认编码。看文档好像说会采用fencs里面的第一个编码作为新建文件的默 认编码。但是这里有一个问题，就是fencs 的顺序跟解码成功率有很大关系，根据我的经验utf-8在前比gb18030在前成功率要高一些，那么如果我新建文件默认想让它是gb18030编码怎么 办？一个方法是每次新建文件后都:set fenc=gb18030一下，不过我发现在vimrc里面设置fenc=gb18030也能达到这个效果。