关于字符集编码

Reference

- http://www.cnblogs.com/skynet/archive/2011/05/03/2035105.html
- http://docs.oracle.com/javase/6/docs/api/java/nio/charset/Charset.html
- http://wiki.eclipse.org/Eclipse.ini
- http://docs.oracle.com/javase/6/docs/api/java/io/InputStreamReader.html
- http://illegalargumentexception.blogspot.com/2009/05/java-rough-guide-to-character-encoding.html
- http://www.cnblogs.com/mailingfeng/archive/2012/03/26/2418445.html

基础知识

- 计算机中储存的信息都是用二进制数表示的；而我们在屏幕上看到的英文、汉字等字符是二进制数转换之后的结果。

- 通俗的说，按照何种规则将字符存储在计算机中，如'a'用什么表示，称为"编码"；反之，将存储在计算机中的二进制数解析显示出来，称为"解码"，如同密码学中的加密和解密。在解码过程中，如果使用了错误的解码规则，则导致'a'解析成'b'或者乱码。

- 字符集（Charset）：是一个系统支持的所有抽象字符的集合。字符是各种文字和符号的总称，包括各国家文字、标点符号、图形符号、数字等。

- 字符编码（Character Encoding）：是一套法则，使用该法则能够对自然语言的字符的一个集合（如字母表或音节表），与其他东西的一个集合（如号码或电脉冲）进行配对。即在符号集合与数字系统之间建立对应关系，它是信息处理的一项基本技术。通常人们用符号集合（一般情况下就是文字）来表达信息。而以计算机为基础的信息处理系统则是利用元件（硬件）不同状态的组合来存储和处理信息的。元件不同状态的组合能代表数字系统的数字，因此字符编码就是将符号转换为计算机可以接受的数字系统的数，称为数字代码。

ASCII

- ASCII（American Standard Code for Information Interchange，美国信息交换标准代码）是基于拉丁字母的一套电脑编码系统。它主要用于显示现代英语，而其扩展版本EASCII则可以勉强显示其他西欧语言。它是现今最通用的单字节编码系统（但是有被Unicode追上的迹象），并等同于国际标准ISO/IEC 646。

- ASCII字符集：主要包括控制字符（回车键、退格、换行键等）；可显示字符（英文大小写字符、阿拉伯数字和西文符号）。

- ASCII编码：将ASCII字符集转换为计算机可以接受的数字系统的数的规则。使用7位（bits）表示一个字符，共128字符；但是7位编码的字符集只能支持128个字符，为了表示更多的欧洲常用字符对ASCII进行了扩展，ASCII扩展字符集使用8位（bits）表示一个字符，共256字符。

ASCII 字符代码表 一

高四位 / 低四位		ASCII非打印控制字符										ASCII 打印字符												
		0000					0001					0010		0011		0100		0101		0110		0111		
		0					1					2		3		4		5		6		7		
		+进制	字符	ctrl	代码	字符解释	+进制	字符	ctrl	代码	字符解释	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	ctrl
0000	0	0	BLANK NULL	^@	NUL	空	16	▶	^P	DLE	数据链路转意	32		48	0	64	@	80	P	96	`	112	p	
0001	1	1	☺	^A	SOH	头标开始	17	◀	^Q	DC1	设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q	
0010	2	2	☻	^B	STX	正文开始	18	↕	^R	DC2	设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r	
0011	3	3	♥	^C	ETX	正文结束	19	‼	^S	DC3	设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s	
0100	4	4	♦	^D	EOT	传输结束	20	¶	^T	DC4	设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t	
0101	5	5	♣	^E	ENQ	查询	21	§	^U	NAK	反确认	37	%	53	5	69	E	85	U	101	e	117	u	
0110	6	6	♠	^F	ACK	确认	22	▬	^V	SYN	同步空闲	38	&	54	6	70	F	86	V	102	f	118	v	
0111	7	7	•	^G	BEL	震铃	23	↨	^W	ETB	传输块结束	39	'	55	7	71	G	87	W	103	g	119	w	
1000	8	8	◘	^H	BS	退格	24	↑	^X	CAN	取消	40	(	56	8	72	H	88	X	104	h	120	x	
1001	9	9	○	^I	TAB	水平制表符	25	↓	^Y	EM	媒体结束	41	)	57	9	73	I	89	Y	105	i	121	y	
1010	A	10	◙	^J	LF	换行/新行	26	→	^Z	SUB	替换	42	*	58	:	74	J	90	Z	106	j	122	z	
1011	B	11	♂	^K	VT	竖直制表符	27	←	^[	ESC	转意	43	+	59	;	75	K	91	[	107	k	123	{	
1100	C	12	♀	^L	FF	换页/新页	28	∟	^\	FS	文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|	
1101	D	13	♪	^M	CR	回车	29	↔	^]	GS	组分隔符	45	-	61	=	77	M	93	]	109	m	125	}	
1110	E	14	♫	^N	SO	移出	30	▲	^6	RS	记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~	
1111	F	15	☼	^O	SI	移入	31	▼	^-	US	单元分隔符	47	/	63	?	79	O	95	_	111	o	127	Δ	^Back space

注：表中的ASCII字符可以用：ALT + "小键盘上的数字键" 输入

GB2312

- 中国专家把那些127号之后的奇异符号们（即EASCII）取消掉，规定：一个小于127的字符的意义与原来相同，但两个大于127的字符连在一起时，就表示一个汉字，前面的一个字节（他称之为高字节）从0xA1用到 0xF7，后面一个字节（低字节）从0xA1到0xFE，这样我们就可以组合出大约7000多个简体汉字了。在这些编码里，还把数学符号、罗马希腊的 字母、日文的假名们都编进去了，连在ASCII里本来就有的数字、标点、字母都统统重新编了两个字节长的编码，这就是常说的"全角"字符，而原来在127号以下的那些就叫"半角"字符了。

- 上述编码规则就是GB2312。GB2312或GB2312-80是中国国家标准简体中文字符集，全称《信息交换用汉字编码字符集·基本集》，又称GB0，由中国国家标准总局发布，1981年5月1日实施。GB2312编码通行于中国大陆；新加坡等地也采用此编码。中国大陆几乎所有的中文系统和国际化的软件都支持GB2312。GB2312的出现，基本满足了汉字的计算机处理需要，它所收录的汉字已经覆盖中国大陆99.75%的使用频率。对于人名、古汉语等方面出现的罕用字，GB2312不能处理，这导致了后来GBK及GB 18030汉字字符集的出现。

GB2312简体中文编码表

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
A1A0			、	。	·	ˉ	ˇ	¨	〃	々	—	～	‖	…	‘	’
A1B0	“	”	（	）	〈	〉	《	》	「	」	『	』	〖	〗	【	】
A1C0	±	×	÷	∶	∧	∨	∑	∏	∪	∩	∈	∷	√	⊥	∥	∠
A1D0	⌒	⊙	∫	∮	≡	≌	≈	∽	∝	≠	≮	≯	≤	≥	∞	∵
A1E0	∴	♂	♀	°	′	″	℃	$	¤	￠	￡	‰	§	№	☆	★
A1F0	○	●	◎	◇	◆	□	■	△	▲	※	→	←	↑	↓	〓	

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
A2A0		ⅰ	ⅱ	ⅲ	ⅳ	ⅴ	ⅵ	ⅶ	ⅷ	ⅸ	ⅹ					
A2B0		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
A2C0	16.	17.	18.	19.	20.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
A2D0	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)	①	②	③	④	⑤	⑥	⑦
A2E0	⑧	⑨	⑩	∈		(一)	(二)	(三)	(四)	(五)	(六)	(七)	(八)	(九)	(十)	
A2F0		Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ	Ⅹ	Ⅺ	Ⅻ			

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
A3A0		！	＂	＃	￥	％	＆	＇	（	）	＊	＋	，	－	．	／
A3B0	０	１	２	３	４	５	６	７	８	９	：	；	＜	＝	＞	？
A3C0	＠	Ａ	Ｂ	Ｃ	Ｄ	Ｅ	Ｆ	Ｇ	Ｈ	Ｉ	Ｊ	Ｋ	Ｌ	Ｍ	Ｎ	Ｏ
A3D0	Ｐ	Ｑ	Ｒ	Ｓ	Ｔ	Ｕ	Ｖ	Ｗ	Ｘ	Ｙ	Ｚ	［	＼	］	＾	＿
A3E0	｀	ａ	ｂ	ｃ	ｄ	ｅ	ｆ	ｇ	ｈ	ｉ	ｊ	ｋ	ｌ	ｍ	ｎ	ｏ
A3F0	ｐ	ｑ	ｒ	ｓ	ｔ	ｕ	ｖ	ｗ	ｘ	ｙ	ｚ	｛	｜	｝	￣	

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
A4A0		ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ	お	か	が	き	ぎ	く
A4B0	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た
A4C0	だ	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は
A4D0	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み
A4E0	む	め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ
A4F0	ゐ	ゑ	を	ん												

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
A5A0		ァ	ア	ィ	イ	ゥ	ウ	ェ	エ	ォ	オ	カ	ガ	キ	ギ	ク
A5B0	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ
A5C0	ダ	チ	ヂ	ッ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ
A5D0	バ	パ	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ
A5E0	ム	メ	モ	ャ	ヤ	ュ	ユ	ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ
A5F0	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ									

Unicode

- Unicode编码系统为表达任意语言的任意字符而设计。它使用4字节的数字来表达每个字母、符号，或者表意文字(ideograph)。每个数字代表唯一的至少在某种语言中使用的符号。（并不是所有的数字都用上了，但是总数已经超过了65535，所以2个字节的数字是不够用的。）被几种语言共用的字符通常使用相同的数字来编码，除非存在一个在理的语源学(etymological)理由使不这样做。不考虑这种情况的话，每个字符对应一个数字，每个数字对应一个字符。即不存在二义性。不再需要记录"模式"了。U+0041总是代表'A'，即使这种语言没有'A'这个字符。

- 在计算机科学领域中，Unicode（统一码、万国码、单一码、标准万国码）是业界的一种标准，它可以使电脑得以体现世界上数十种文字的系统。Unicode 是基于通用字符集（Universal Character Set）的标准来发展，并且同时也以书本的形式[1]对外发表。Unicode 还不断在扩增，每个新版本插入更多新的字符。直至目前为止的第六版，Unicode 就已经包含了超过十万个字符（在2005年，Unicode 的第十万个字符被采纳且认可成为标准之一）、一组可用以作为视觉参考的代码图表、一套编码方法与一组标准字符编码、一套包含了上标字、下标字等字符特性的枚举等。Unicode 组织（The Unicode Consortium）是由一个非营利性的机构所运作，并主导 Unicode 的后续发展，其目标在于：将既有的字符编码方案以Unicode 编码方案来加以取代，特别是既有的方案在多语环境下，皆仅有有限的空间以及不兼容的问题。

- Unicode是字符集，UTF-32/ UTF-16/ UTF-8是三种字符编码方案。

UTF-8

- UTF-8（8-bit Unicode Transformation Format）是一种针对Unicode的可变长度字符编码（定长码），也是一种前缀码。它可以用来表示Unicode标准中的任何字符，且其编码中的第一个字节仍与ASCII兼容，这使得原来处理ASCII字符的软件无须或只须做少部份修改，即可继续使用。因此，它逐渐成为电子邮件、网页及其他存储或传送文字的应用中，优先采用的编码。互联网工程工作小组（IETF）要求所有互联网协议都必须支持UTF-8编码。

- UTF-8使用一至四个字节为每个字符编码：
  - 128个US-ASCII字符只需一个字节编码（Unicode范围由U+0000至U+007F）。
  - 带有附加符号的拉丁文、希腊文、西里尔字母、亚美尼亚语、希伯来文、阿拉伯文、叙利亚文及它拿字母则需要二个字节编码（Unicode范围由U+0080至U+07FF）。
  - 其他基本多文种平面（BMP）中的字符（这包含了大部分常用字）使用三个字节编码。
  - 其他极少使用的Unicode辅助平面的字符使用四字节编码。

对程序来说，哪些地方需要字符编码

- 程序文件存储的编码
    - Java文件
- 编译时的编解码
    - class文件
- 执行时虚拟机解码
- 输入Java API的解码，输出Java API的编码
- 控制台输入的编码、控制台输出的编码

Java 支持Charset

- US-ASCII
    - Seven-bit ASCII, a.k.a. ISO646-US, a.k.a. the Basic Latin block of the Unicode character set
- ISO-8859-1
    - ISO Latin Alphabet No. 1, a.k.a. ISO-LATIN-1
- UTF-8
    - Eight-bit UCS Transformation Format
- UTF-16BE
    - Sixteen-bit UCS Transformation Format, big-endian byte order
- UTF-16LE
    - Sixteen-bit UCS Transformation Format, little-endian byte order
- UTF-16
    - Sixteen-bit UCS Transformation Format, byte order identified by an optional byte-order mark

Java文件保存

- Eclipse
  - Properties for IOHelper.java
    - Resource
      - Text file encoding

输入Java API的解码，输出Java API的编码

- InputStreamReader
  - public InputStreamReader(InputStream in)
  - Creates an InputStreamReader that uses the default charset.
- Parameters:
  - in - An InputStream

输入Java API的解码，输出Java API的编码

- InputStreamReader
  - public InputStreamReader(InputStream in, Charset cs)
  - Creates an InputStreamReader that uses the given charset.
- Parameters:
  - in - An InputStream
  - cs - A charset
- Since:
  - 1.4

编译和执行

- 编译
  - javac -encoding UTF-8 src/IOHelper.java
- 执行
  - java -classpath bin -Dfile.encoding=utf-8 IOHelper

控制台输出

- Eclipse
    - Your Project
        - Run Configurations
        - Encoding

控制台输入

- Eclipse
    - eclipse.ini
        - -Dfile.encoding =UTF-8//虚拟机参数