© Dipl.-Ing. Jörn Jacobs, M.A., Meissnerweg 10, 64289 Darmstadt, Germany

Unicode, Classical Chinese Texts and the Use of Free Software

0 General

In recent years, substantial changes have occurred in the field of literary „production“. There are no longer any real manuscripts, literally speaking. In former times, an author could, if he so wished, leave any typographic matters to the publisher and /or the printer. Today, unless you are a world-famous writer, it's ready-to-print files that are expected from you. Well, a sinologue in the west, even in the old days, maybe never just left his manuscript with the typesetting office either. Anyway, typographic matters, ere the typesetters domain, and later the domain of special typesetting computers, today are present within any personal computer, and are supposed to be used and mastered correctly and efficiently! Therefore, whoever does more with Chinese than just reading or handwriting text, i.e. whoever

- wants a full text search within Chinese texts (scanning text corpora, using Internet search engines),
- envisages the production of typographically acceptable texts containing western alphabets as well as Chinese characters, and finally who
- wants to include characters and/or character versions not included in the character fonts supplied, and wants to do this in a reasonably efficient way,

is likely to encounter one or more of the following problems:

- The operating system installed on the computer at hand, though provided with “Asian character handling capabilities", yet has to be configured accordingly.
- The fonts needed are are partly missing.
- There are no straightforward “typewriter” programs/editors or input modes, respectively, that would facilitate fast input of those texts or text elements containing Chinese characters.
- Sometimes, on the screen and/or later on the printouts, certain characters or character variants are missing.
- Certain character variants from old texts, though definitely listed e.g. in the Kāngxī Zìdiǎn, did not find their way into the standardized character sets.

When having problems with commercially available software, one could in theory go back to the vendor, if the product were to have these properties and features. Such claims are seldom realistic, however. Moreover, requirements as pronounced by a sinologue are difficult to communicate to a “normal” user of word processing; chances are high you get many answers, but only a few out of these will relate to the problem. Many people consider Japanese Kana to be “Chinese” (which is, in fact, etymologically true!). Commercial software is normally not alterable. The information necessary to do this is usually not provided for. In some cases it is even explicitly forbidden to modify anything at all; or modifications are strongly discouraged.

Of course one could try and use, from the very beginning, software that is being used in China. Doing so will surely solve many problems, though this may not necessarily be true for those problems typical for the philology of ancient texts and their archaic characters or character variants. Furthermore, it is my experience (of some years back, though), that such systems may have some limitations in areas typically important only for European languages, like syllabification, word-wraparound, language specific characters, diacritics, etc.

Many problems can be avoided in an elegant way by using **free**[*] **software**:

- such software can be downloaded and installed quickly and easily from one of the trustworthy http- or ftp-sites; access may be via www.sourceforge.net or the respective „mirrors“ in one's own country (usually hosted at universities); a DSL (digital subscriber loop) connection is fast enough in any case.
    - Installation is operating system dependent and should not normally be any difficult. “Laymen”, however, may encounter minor problems; advice from someone having some knowledge of the operating system will be of help. By the way, users of LINUX-like operating systems run nearly no risk at all as far as computer viruses are concerned, because most installation procedures do not run any executable files directly at installation. With other operating systems, using CDs/DVDs from a known source (e.g. from a computer magazine) is maybe the better way.
- Software to be used on the computer can be added in a step-by-step manner as necessary or desired
- all this is at very little or no cost at all, neither at the beginning nor at a later time
- the software may be analyzed, modified, and handed on to others
- because everything is usually open source, the functioning as such is a subject of discussion in the respective internet groups. You get a good insight into what really happens and you get rapid answers to questions either from experienced users or even the author(s) of the software. Contributions in the relevant user groups are problem oriented and give you in-depth explanations if necessary, whereas with commercial software, there usually is just discussions about the mysteries of hearsay or hidden properties of that software, the source code of which intentionally is kept a secret. For those interested in solving some programming problems at operating system level, comprehensive information is available in the countless program descriptions (“man pages” = manuals) that form an integral part of all free operating systems.
- Free software usually adheres strictly to standards. Thus the combination with other portions of software often leads to new possibilities and even to the ad-hoc-creation of small task solvers. As an example, the Unicode overview in the following chapter has been created thus:
    - the relevant files were retransformed from “.pdf” back into to “text” format using the “pdftotext” command
    - every file's first line was taken, because the info we want was just there, using the “head” command.
    - The resulting lines were all written into a file.

  All this was carried out by a so called *shell script*“ (=a computer programm using the commands of the operating system):

```
# /bin/bash
for X in /subdirectory/*.pdf
```

[*] I.e.: It can be obtained free from any obligation, preferably at no cost, for use by the general public, and preferably as open source software.

```
do
pdftotext $X „-“ | head -1 >>resultfile
done
```

- free software is mostly up-to-date; currently this includes features such as:
    - Using Unicode for all character coding,
    - Implementation of practically all keyboard layouts used in the world (mind this is absolutely essential for the writing of texts in **all** languages **other than English**!)
- there is, and this has to be gratefully appreciated, a very big number of idealists who, either on their own or within network groups, are writing software they present to the public free of charge, for discussion an for general use.
- free software often is available in different versions, for the various operating systems (Linux, FreeBSD, McOSxx, often including even Winxx).

More specifically now, i.e. from a “writing sinologue's” viewpoint, the world of free software is therefore a pleasantly positive and productive one. One can make good use of:

- free operating systems like **Linux, FreeBSD** etc. which, when using **KDE** as the desktop manager, allow for instantaneous switching between the various keyboard layouts,
- freely available Unicode-coded chinese character sets like e.g. **Firefly-Sung**,
- the **code tables** published by the **Unicode** Consortium (can be downloaded as pdf-files), an invaluable source of information for all special cases. The placement of the individual characters within those tables, and with that the character coding itself, this is a nice surprise for the sinologue, is done using the classical 214 elements radical system! More on this below. It is advisable to spend a little money an do a printout of these code tables. For it does happen at times that some Internet browser gets stuck over a character (usually a fairly rare one) which it is unable to display (because it is missing in the font currently installed). Often the respective Unicode position is given instead (or can be dug out from the html source code). A quick glance into the table therefore given us the character in question. Furthermore, the printouts can serve as a helping basis for characters still to be created (see next item over, and later in more detail).
- For modifications and additions to character sets (e.g. true type fonts, ttf), there is a really great program called **fontforge**. It is also highly suitable for just browsing through a font.
- You can use a program called **autotrace** to process drafts of characters (either as a photograph or a scan from a handwritten sketch, or other source image) in such a way that the above-mentioned fontforge program can transform these into truly scalable true type fonts.
- Research on the more rare characters, their respective pronounciation(s) and their loci in the dictionaries (e.g. Kāngxī Zìdiăn, etc.) are facilitated through the **Unihan data base** http:// www.unicode.org/ charts/unihan.html (can be used for online search, or as a searchable text file which is downloadable at ftp://ftp.unicode.org/ Public/UNIDATA/ Unihan.zip))
- And of course there are the application programs as such, e.g.
    - The well-known **Open Office Writer** text processor which doesn't leave any

wishes unfulfilled,

- the **Texmacs** publication software which fully stands in the tradition of „**TeX**“ - well-known with scientists - yielding well-structured, typograhically acceptable high-grade texts. Both of them do output pdf-files now, if desired.
- An editor called **mined-2000.14**, fully based on Unicode, permitting, through its numerous keyboard input modes, the writing of texts in many languages, including Chinese: in this case including phonetic (pīnyīn) and radical+ strokes input. There's even a mode which displays, for every character under the cursor, the respective Unicode-position as well as possible pronounciations (current mandarin and cantonese, old Tang era, and others). This editor permits easy input of the Chinese portions of a mixed text; these portions would then be inserted easily into a fully-featured word processing system subsequently used, like Open Office Writer, by using the “insert file” function of the latter.
- Today's web browsers, e.g. **Firefox** (**Iceweasel**), **Konqueror** etc., all of them capable of handling both Unicode coded texts or web sites, and equally the older codings for Chinese characters such as GB2312, Big5 etc.
-

This way, you are prepared for all practical situations. The following chapters are supposed to explain the use of these programs, as well as to try and give some background information. The necessity for this becomes vitally apparent the moment you sit at somebody else's computer wanting to on-the-fly enter some lines of Chinese text: Some knowledge of input methods, character sets and fonts is definitely necessary. The description and the examples are from the viewpoint of a sinologist as the user, with the example of software available for the Linux operating system.

1 Today's computer character sets (fonts)

All Computers today are equipped with a multitude of character sets, their incarnations in fact being named fonts; most of these, however, are just a “variance in beauty” of the basic latin characters, and hopefully include the more frequent cases of diacritical marks on some of them. Supply of “unusual” character sets, such as IPA (International phonetic alphabet), Hebrew, Chinese, Korean, Japanese Kana, however, is often very limited and sometimes buggy. Moreover, though provided, they sometimes have to explicitly be “switched on”.

2 Character set coding; Unicode

In the earlier times, apart from proprietary solutions that made the users totally dependent on the computer (and printer) models used, there were attempts of introducing order into the practical work of “writing with a computer”, and thus some standardization. The aim then was to facilitate the exchange of texts between computers of different make or version. Thus

Early codings for latin characters were, e.g.:

- CCITT No. 2 (the international teletype alphabet),
- EBDIC,

- CCITT No. 5 (nearly 100% identical to the US „ASCII-Code“).

The latter found itself partly modified with a number of “national extensions”. In fact some positions within the code set (sized only 127 characters by itself), such as the ones for “[“,”\”,”]”,”^”,”{“,”|”,”}”,”~” etc. were redefined to mean “Umlaut”s in Germany, accented characters for French, etc. etc. Not only did that introduce a burden for the practical exchange internationally (the codes used were only partly identical, and, even worse: You did not necessarily always “see” what you typed), it still did not offer enough room for all of the characters necessary for even the languages using only latin script, let alone any coverage of other wide-spread writing systems like e.g. Chinese.

Yet a fully standardized way of character coding is a necessity even with today's WYSIWYG (“What you see is what you get”) way of work, for the sake of compatibility with other computer systems and users, transmission via e-mail being a frequent example. Unfortunately, at the beginning, there have been a number of competing standards. Well, to be honest, better have multiple, but well defined standards, rather than none at all. But the transcoding from one norm into the other was neither economic nor as error-free as it should be, even if it was “just a side task” to the computer: The respective software, especially in the case of Chinese characters, is far from being trivial.

Early codings for Chinese characters are, for example:

- the Japanese JIS-Kanji
    - JIS C 6226-1978. with 2965 Level 1 KANJI, 3384 Level 2 KANJI, 453 non-KANJI.
    - JIS X 0208-1983, ≈ JIS C 6226-1978, but minor changes.
- China: Guobiao (GB)
    - GB2312-1980, with 6763 Hanzi, 682 non-Hanzi

      (Note: GB18030 already corresponding to Unicode)
- China (Hongkong, Taiwan) Big5, since 1984, with 13062 characters.

In goes without saying that the afore-mentioned Chinese language character sets readily included the European characters, though not completely.

Also, there were variants for traditional / simplified characters, respectively, in the GB and Big5 character sets; because of the well-known fact that there is not always a direct one-to-one correspondence between traditional and simplified, simple “switching” between these two lead to ugly errors at times. Those who want to convert existing texts even today may find the following program useful: It is called „**b2g.pl**“ (it is a perl script running under linux); it converts in both directions and also within Unicode text. We do, however, leave the issue of older coding systems now and concentrate on

The so-called **UNICODE** coding in use today, which

- is a world-wide Standard, worked out in close collaboration with China, Japan and Korea, the countries of special interest e.g. for sinologues, and completely and totally replaces the coding schemes mentioned above,
- contains, in addition to the latin-based European characters (with practically **all** special characters and accents included now), the IPA characters, also Hebrew, Arabian, Thai, and nota bene

- practically all Hanzi (Kanji, Hangul), i.e. all Chinese-type characters used in China, Japan and Korea, viz.:
  - a particularly great number (more than 70000, see below) characters; this should be sufficient for all “normal” cases (except, sometimes, when dealing with some versions of classical texts). The Standard contains
  - both traditional and simplified characters within the same code tables, the simplified ones usually placed in the vicinity on the traditional ones, and under the same radical.
- The standard itself and the code tables are available in the Internet. They can be downloaded as pdf-files (particularly useful for classical philology, in order to check whether “that peculiar-looking character” had yet been included in the standard, which is often the case).
- The characters are arranged, within the various code tables (the basic table is called **„CJK Unified Ideographs“** , then there is the so-called **CJK Unified Ideographs Extension A**, the **CJK Unified Ideographs Extension B** etc.), according to the generally well-known 214 elements radical system, and by stroke number and variants within. Knowing this, makes you maintain your oversight over the matter, and you can find your way through the tables even without special software. This can be of relevance if you should ever, as mentioned earlier, sit at someone else's computer, there's no Chinese character software installed, and yet you are asked to insert some Chinese characters – the font is there – by using the “insert special character” dialogue window, which is small and only shows a tiny fraction of those thousands available in that huge character table.

The following is an **overview of the most relevant Unicode code tables** sorted by types of characters. Those areas that are particularly relevant to sinology are numbered, in column 1, by the likelyhood of occurrence (in average modern text, that is).

	Name	Characters (examples)	Coding area (Hex)	Number
	C0 Controls and Basic Latin	!“..123...ABC...abc..	0000– 007F	128
	C1 Controls and Latin-1 Supplement	¡¢£...ÆÇ...ûüýþÿ	0080– 00FF	128
	Latin Extended-A	ĀāĂă...żŽžſ	0100– 017F	128
	Latin Extended-B	⅃ǁ Ɓ...DŽ...LJ...ǽ...ɏ	0180– 024F	208
	IPA Extensions	ɐɑɒ...ʁʂʃ...ʒ...ʮʯ	0250– 02AF	96
	Spacing Modifier Letters	Diakritics:...ˊ˝...˚...˿	02B0– 02FF	80
	Hebrew	è . . . ״ֽ. . . א ב ג ד ת	0590– 05FF	112
	Phonetic Extensions	Non-IPA-phonetic characters	1D00– 1D7F	128
	Phonetic Extensions Supplement	Further IPA characters and diacritics	1D80– 1DBF	64

	Name	Characters (examples)	Coding area (Hex)	Number
	Latin Extended Additional	ḀḁḂḃ…Ẩẩ… ▪▪	1E00– 1EFF	256
	Braille Patterns	⠁⠂⠃⠄ …⠑⠒…⣽⣾⣿	2800– 28FF	256
	Latin Extended-C	Ⱡⱡ…Ⱳ…▯	2C60– 2C7F	32
1	CJK Radicals Supplement	⺀⺁⺅ …⻈ …⻳	2E80– 2EFF	128
1	Kangxi Radicals	⼀⼁⼂⼃…⿓⿔⿕	2F00– 2FDF	224
	Ideographic Description Characters	⿰⿱⿲⿳⿴⿵⿶⿷⿸⿹⿺⿻ …	2FF0– 2FFF	16
	Kanbun	…㆒㆓㆔㆕㆖㆗㆘㆙㆚㆛㆜㆝ ㆞㆟	3190– 319F	16
1	CJK Strokes	㇀㇁㇂㇃㇄㇅㇆㇇㇈㇉㇊㇋㇌㇍㇎㇏ ㇐㇑㇒㇓㇔㇕㇖㇗㇘㇙㇚㇛㇜㇝㇞㇟ ㇠㇡㇢㇣	31C0– 31EF	48
2	CJK Unified Ideographs Extension A	㐀㐁㐂㐃㐄㐅㐆㐇㐈㐉㐊㐋㐌.. 䀀䀁䀂䀃䀄䀅䀆…䶪䶫䶬䶭䶮 䶯䶰䶱䶲䶳䶴䶵	3400– 4DBF	6592
1	CJK Unified Ideographs	一丁丂七……瀀瀁……言訁 訂訃 訄訅訆…讠 计订讣认讥讦……… 車軋軌軍軎軏軐…车轧轨轩轪 轫转轭轮………鿁鿂鿃	4E00– 9FCF	20944
	Modifier Tone Letters	꜀…꜖▯	A700– A71F	32
	Latin Extended-D	꜠꜡…Ꞁ…ą̄ꟾꟿ	A720– A7FF	224
2	CJK Compatibility Ideographs	豈更…茶刺… ▪▪	F900– FAFF	512
	Alphabetic Presentation Forms	ﬀﬁﬃ…ﬠﬡ…פֿﭏ	FB00– FB4F	80
3	CJK Unified Ideographs Extension B	𠀀𠀁𠀂𠀃𠀄𠀅𠀆……言言訇訁 訂𧥠𧥡…𬣳……𪛑𪛒𪛓𪛔𪛕𪛖	20000–2A6D6	42720
2	CJK Compatibility Ideographs Supplement	丽丸乁𠄢你侮侻倂.....黾鼅鼏鼖 鼻𪘀	2F800–2FA1F	544

Furthermore, there are so-called „private use areas“, the use of which should, however, be avoided because using such positions for characters effectively means leaving the area of standardization. Texts containing such characters would inevitably be bound to the very character font used and text portability is very limited; any display/editing/printing on a "normal" computer (having "only" the standardized character sets) would lead to errors and/or lacunae in the text, unless you escape to the .pdf-Format, which, in turn, is not reasonable at least for texts intended to be edited or processed any further.

Considering the 'number of code positions' column (rightmost in the table) makes evident, that the probability of encountering, somewhere within a text, a non-standardized character has substantially decreased. Whereas in former times, only some 4,500 characters were at hand, this number is now up in the 70,000s. Thus philological reality has greatly improved through Unicode. Under one condition: The very character fonts installed on individual computers should be complete and error-free. See Item 4.1: “The problem of incomplete character fonts”.

The order in which the various code area are defined, within the Unicode standard, usually leads to a puzzling and annoying behaviour of most software. When using, for the input of just a few characters, the afore-mentioned “insert special character” mode of the word processor, the characters are displayed in numerical order. Thus you are offered in some strange sequence

- all „CJK unified ideographs **Extension A**“ characters first , and
- only further down then you will see **„CJK unified ideographs“** (the basic character set) and the other areas.

3. Inputting Chinese characters in Internet search engines

3.1 Input modes for Chinese characters installable at operating system level:

As for usual word processor software, see the special chapter on the **mined** editor; it facilitates easy input of the Chinese sections within a mixed text, which then can be inserted (using the “insert file” funtion) into the main text already under editing e.g. with the Open Office Writer.

Mail software and internet browsers do not normally have an “insert file” function. Specially installed input modes are required here. There is a number of programs around, such as **SCIM** (see below), the installation and configuration of which depends on the Linux distribution (Ubuntu, SuSe, Debian, etc.). I did'nt yet test these. Therefore just a few possible links:

- http://www.tldp.org/HOWTO/Chinese-HOWTO.html v1.04, 2 June 1998
- http://www.gerrit-worldwide.de/chinesischlinux2_eng.php, last update 01.16.2004
- scim-0.8.2-1.i586.rpm and scim-chinese-0.2.6-1.i586.rpm (for pinyin input method). (11/2003)

also, the classical shareware program called NJStar is reported to be running under Linux using the **wine** emulator.

- Running NJStar Chinese/Japanese WP on Linux and Mac under WINE 1.x, see www.njstar.com

3.2 A hack using the software at hand:

For those without a Chinese character input system installed: The following procedure facilitates the input of Chinese characters (and **any** other Unicode character alike) quite comfortably:

- Use any text editor and create the following simple HTML document. The decisive element is the „charset=UTF-8“ command. One would save this document under e.g. „seekthis.htm“. Also keep a backup copy of this little gem.

```
<html>

<head> <meta http-equiv="content-type"  content=" text/html; charset=UTF-8">
</head\>
\<body>
Hint to user: Place search phrase here in UTF-8 coding and save this file. Later
when being opened with inet browser, copy relevant phrase into search engine's
input field:<phrase>
</body>
</html>
```

- Open up the **mined** editor and load the above html file, follow the hint by inserting the desired search phrase, making good use of mined's various input modes, save the file, e.g. under *seekthis.htm,* and leave mined.

- Open the internet browser (e.g. **firefox, iceweasel, konqueror**,...) and have the above, local html file displayed. All characters will appear correctly and they now can be copied into the search field of the search engine used.

All this can even be made fully automatic by using the following, fairly trivial shell script to be placed on the desktop and made “clickable”:

```
# /bin/bash
/home/user/mined/uterm -e /home/user/mined/mined   /home/user/mined/seekthis.htm
firefox   /home/user/mined/seekthis.htm
```

Note. The location in the home directory is just an example.

This works surprisingly well, especially if the Internet browser already contains a search field (most do). If not, just call the browser up a second time, call the search engine's main page there, and copy the phrase from the first window to the second one via drag and drop.

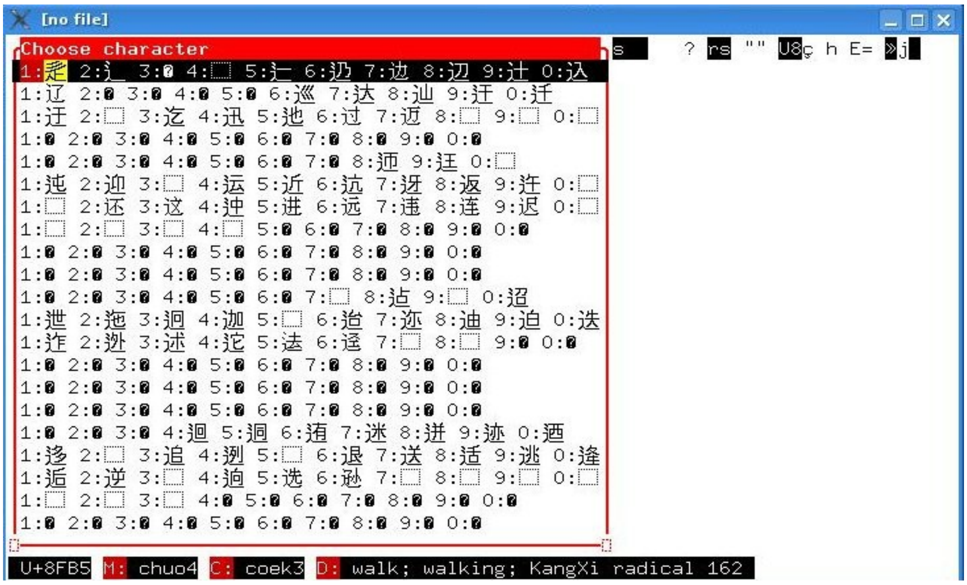
Excursus: Notes on the „mined“ editor.

Of course this editor is not as comprehensive as a fully-featured word processing system like „Open Office Writer“; thus, once started you cannot switch fonts within a text for a better coverage, e.g. into HAN NOM A. Therefore, instead of just typing “mined' to start mined, it may come handy calling **mined** like this:

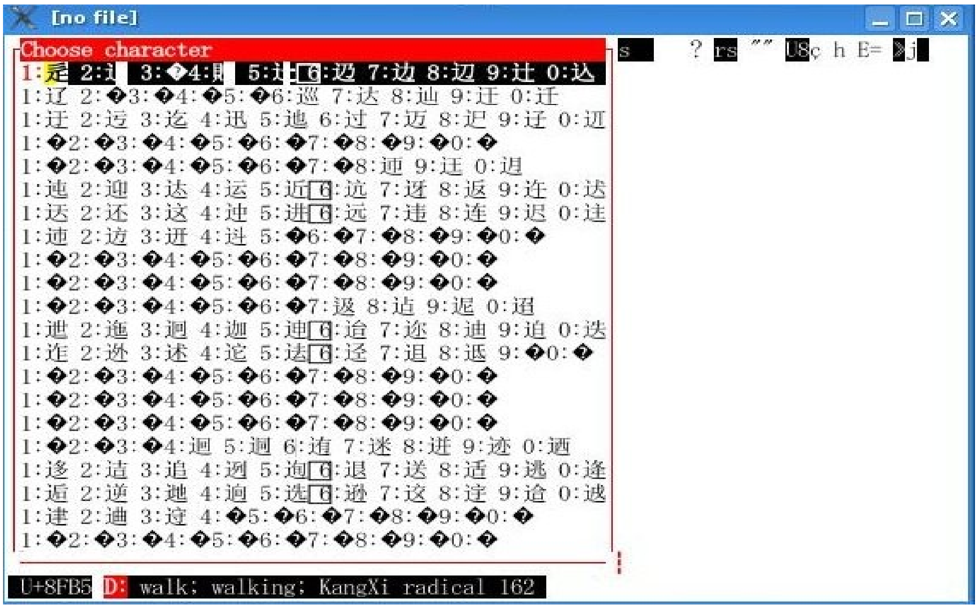
```
xterm -fa 'HAN NOM A' -e mined <dateiname>,
```

whereby the far more comprehensive character set is used explicitly. Unfortunately I did not yet find a way of likewise using HAN NOM B. Thus, those particularly rare characters contained in „CJK Unified Ideographs extension B“ cannot be displayed in mined. Their codes, however, can be input easily, and that is more than half of the game! If necessary, verify the codes in the Unicode code tables. A file created this way, once inserted into **Open Office Writer**, will readily display the correct characters once the HAN NOM B character set is selected there.

The following screen shots show the **mined** editor in the radical/stroke number input mode, radical 162 as an example. The first image shows a simple “mined” call, the second image shows the far better coverage if mined is called up via the method that explicitly states 'HAN NOM A' described above.



```
[no file]
Choose character                                      s    ? rs "" U8ç h E= »j
1:辵 2:辶 3:� 4:□ 5:⻌ 6:辸 7:边 8:辺 9:辻 0:込
1:辽 2:� 3:� 4:� 5:� 6:巡 7:达 8:辿 9:迀 0:迁
1:迂 2:□ 3:迄 4:迅 5:迆 6:过 7:迈 8:□ 9:□ 0:□
1:� 2:� 3:� 4:� 5:� 6:� 7:� 8:� 9:� 0:�
1:� 2:� 3:� 4:� 5:� 6:� 7:� 8:迊 9:迋 0:□
1:迌 2:迎 3:□ 4:运 5:近 6:迒 7:迓 8:返 9:迕 0:□
1:□ 2:还 3:这 4:迚 5:进 6:远 7:违 8:连 9:迟 0:□
1:□ 2:□ 3:□ 4:□ 5:� 6:� 7:� 8:� 9:� 0:�
1:� 2:� 3:� 4:� 5:� 6:� 7:� 8:� 9:� 0:�
1:� 2:� 3:� 4:� 5:� 6:� 7:� 8:� 9:� 0:�
1:� 2:� 3:� 4:� 5:� 6:� 7:□ 8:迠 9:□ 0:迢
1:迣 2:迤 3:迥 4:迦 5:□ 6:迨 7:迩 8:迪 9:迫 0:迭
1:迮 2:迯 3:述 4:迱 5:迲 6:迳 7:□ 8:□ 9:� 0:�
1:� 2:� 3:� 4:� 5:� 6:� 7:� 8:� 9:� 0:�
1:� 2:� 3:� 4:� 5:� 6:� 7:� 8:� 9:� 0:�
1:� 2:� 3:� 4:� 5:� 6:� 7:� 8:� 9:� 0:�
1:� 2:� 3:� 4:迴 5:迵 6:迶 7:迷 8:迸 9:迹 0:迺
1:迻 2:□ 3:追 4:迾 5:□ 6:退 7:送 8:适 9:逃 0:逄
1:逅 2:逆 3:□ 4:逈 5:选 6:逊 7:□ 8:□ 9:□ 0:□
1:□ 2:□ 3:□ 4:� 5:� 6:� 7:� 8:� 9:� 0:�
1:� 2:� 3:� 4:� 5:� 6:� 7:� 8:� 9:� 0:�
U+8FB5 M: chuo4 C: coek3 D: walk; walking; KangXi radical 162
```



```
[no file]
Choose character                                      s    ? rs "" U8ç h E= »j
1:辵 2:辶 3:�4:辶 5:辶6:辸 7:边 8:辺 9:辻 0:込
1:辽 2:�3:�4:�5:�6:巡 7:达 8:辿 9:迀 0:迁
1:迂 2:迃 3:迄 4:迅 5:迆 6:过 7:迈 8:迉 9:迂 0:迈
1:�2:�3:�4:�5:�6:�7:�8:�9:�0:�
1:�2:�3:�4:�5:�6:�7:�8:迊 9:迋 0:迌
1:迌 2:迎 3:迏 4:运 5:近6:迒 7:迓 8:返 9:迕 0:迖
1:迗 2:还 3:这 4:迚 5:进6:远 7:违 8:连 9:迟 0:迠
1:迪 2:迲 3:迸 4:迁 5:�6:�7:�8:�9:�0:�
1:�2:�3:�4:�5:�6:�7:�8:�9:�0:�
1:�2:�3:�4:�5:�6:�7:�8:�9:�0:�
1:�2:�3:�4:�5:�6:�7:迻 8:迠 9:迡 0:迢
1:迣 2:迤 3:迥 4:迦 5:迧6:迨 7:迩 8:迪 9:迫 0:迭
1:迮 2:迯 3:述 4:迱 5:迲6:迳 7:迴 8:迵 9:�0:�
1:�2:�3:�4:�5:�6:�7:�8:�9:�0:�
1:�2:�3:�4:�5:�6:�7:�8:�9:�0:�
1:�2:�3:�4:�5:�6:�7:�8:�9:�0:�
1:�2:�3:�4:迴 5:迵 6:迶 7:迷 8:迸 9:迹 0:迺
1:迻 2:迼 3:追 4:迾 5:迿6:退 7:送 8:适 9:逃 0:逄
1:逅 2:逆 3:逇 4:逈 5:选6:逊 7:逋 8:逌 9:逍 0:逎
1:逑 2:逓 3:逕 4:�5:�6:�7:�8:�9:�0:�
1:�2:�3:�4:�5:�6:�7:�8:�9:�0:�
U+8FB5 D: walk; walking; KangXi radical 162
```

4. Corrections and Extensions of Character sets

4.1 The Problem of „incomplete character sets“

The character sets provided together with the operating system (“Distribution XYZ, version ## in the case of Linux) are usually taken for granted. One should, though, investigate them a little using „f**ontforge**“ and compare them with the Unicode code tables. The following is often true:

- There are missing areas. Maybe just the „CJK Unified Ideographs, positions 4E00–9FCF are there, giving you 20,944 characters „only“.
- there are gaps within the character set itself. These are mostly characters deemed “rare and never encountered in practical cases” by the creators of these fonts.
- There also are some code positions in the standard with no characters assigned.

If these gaps are too large and not acceptable, one should strive for an alternative character set. A character set presently freely available is called „Firefly Song“ and can be downloaded at http://firefly.Idv.Tw/apt/firefly-font/ fireflysung-1.3.0.tar.gz.Once istalled it appears as „AR PL New Sung“, containing 17,378 glyphs.

Another interesting character set named „HanNom v2005 Release“ is available at http://sourceforge.net/projects/vietunicode/files/hannom/hannom%20v2005/hannom.zip/download, in two resolutions: (HanNom.zip) and(HanNomH.zip). Both files contain two character sets each:

HAN NOM A :

- Radicals Supplements [u+2F00 …u+2FD5 (214 glyphs)]
- CJK Unified Ideographs Extension A [u+3400 ...u+4DB5 (6,572 glyphs)]
- CJK Unified Ideographs [u+4E00 ...u+9FA5 (20,832 glyphs)]
- Private Use Area [u+E000 ... (glyphs temporarily used by the Dictionary compilers team)]

HAN NOM B:

- CJK Unified Ideographs Extension B [u+20000 ...u+2A6D6 (42,702 glyphs)]
- Fillers used by font designers on purpose [u+2A6D0 ...]

Yet another character set might doze in some computers: Named SimSun or NsimSun, resp., featuring 22,141 characters, which includes „Extension A“. It is normally found in the WinXP partition.

Well, Sinologues working exclusively in modern Chinese literature would normally not encounter any practical difficulty at all, out of the above, because the characters used in these texts are all standardized and mostly very current and thus not likely to have been “forgotten”. If they were, one would either create these characters and place them in the correct code positions, or, and this is maybe the clearer approach, create a small character set under a new name containing just these very characters.

4.2 The problem of „old texts“

In the case of characters in ancient Chinese, that do not have a “modern” correspondance in the traditional characters, first make sure the character wasn't overlooked. As the Unicode documentation is complete (unlike many fonts installed in the computer), one would, after hopefully having identified the character in a character dictionary (Kāngxī Zìdiăn strongly recommended), therefore go through the Unicode code tables. Mind scrutinizing all parts of the standard, viz.

- „basic“,
- „supplement“, and
- Extensions A and B.

Each of them is arranged according to the 214-radicals system. Therefore a sinologue quickly feels right at home. The main purpose of this exercise is avoiding an unnecessary home-made creation of a character which was thought nonstandard, but which does appear happily in the standard.

Finally you find out, about that character, either of the following:

	standardized*	Not standardized
Is contained in the character set used	A	B
Is missing in the character set used	C	D

Case A: The normal case; nothing to be done

Case B: Not likely to be encountered with original, i.e. unmodified character sets, apart from characters in the so-called private use areas.

Case C: Not as rare as one would think. The character is standardized, but not contained in the character set/font being used. The code position is “empty”, for the following reasons:

- „Sparseness”: only the more frequent characters have been included
- Unicode standard's so-called „extensions“ and „supplements“ are missing (such a character set would thus be rather unsuited for philological work)
- the character set is non-final. You got hold of an early version of its development
- any other reasons of not filling the very code position
- software bugs in the word processing program: The code positions in question do contain the characters, but the software cannot access them. As a last resort one would have to treat such a case as case D below. An example: OpenOffice 2.0 was unable to access the HanNom-B-character set (because HanNom-B uses code positions of Hex 20000 and above, I understand). OpenOffice 3.0, the good news,

happily displays and processes these! So, you better use OpenOffice3.0.

Case D: This is the standard case for

- all characters that, being unusual variants, did not find entry into the standard, and which, however, you would like to present, by philological reasons, exactly as the variant given in the text. As has been stated before, with approx. 70,000 code positions in the standard, this is not likely to occur so very often, provided a reasonable completeness of the character sets actually used. Furthermore all
- characters that are to be reproduced in their true (archeologically correct) form. This does not detract from the fact that a transcription into the character form as would be used today (evidence for this would normally rely on philological comments) is still desirable in order to make the old text more readable. For all those special characters within some text one would create one' s own special font named “SomeText-Extra-Chars.ttf”. The technique of designing these will be looked at in the next chapter. Obviously, doing this, for those special characters, means leaving the path of standardization, but keeping up order elsewhere. A text produced under such a regime is not fully transferable (i.e. completely and readily readable and reprocessable by anyone) any more, unless this special “Extra-Chars” font were installed at the destination computer as well. Usually, and that should be the safe and normal case, you just transfer a “pdf”-file, for pdf files automatically incorporate information on any “foreign” characters into the file.

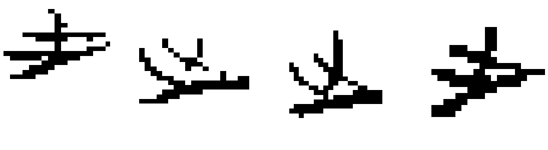
4.3 Solution for missing characters. The make of a new character via *handmade sketch/digital photograph* ⇒ *„autotrace“* ⇒ *„fontforge“*

General note: For all cases where a new character could be derived from other, existing ones via minor modifications only, thus no real graphic artwork is needed, the design work could be done fully from within *fontforge. For details see the documentation of the fontforge program.*

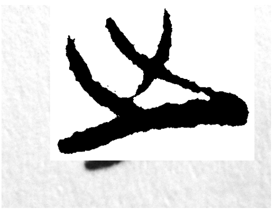
One would normally start off a suitable sketch or draft on paper. This has to be transformed into some kind of picture file. That means, either

- Scanning a handmade sketch of the character
- do a close-up digital photograph of the handmade sketch
- do a close-up digital photograph of the character as given by the archeological sources

An example: We are out for documenting the ways of writing of the 之 „zhì“ character within the Guōdiàn-Text, the gǔwén-character of which, by the way, appears as 㞢 (see Kāngxī Zìdiǎn; 㞢 is even contained in Unicode's „CJK Extension A“). We therefore want to create something like the following:



This is how the character appears in the original text (Guōdiàn Laozi, chapter 2, samples from this handwritten archeological



text). We decide for the second of these, and make a digital photograph (for scientific/educational purposes) using the original publication. We cut the image to size using e.g. **GIMP**, but leave it unchanged otherwise. The image is then stored under the name GD-Kap2-zhi002.jpg. We now need it in the .bmp format, and also as a black-and-white-only image, i.e. with a colour depth of “2” only. No problem at all for any graphic software. Finally the image file comes out of GIMP as „GD-Kap2-zhi002-BW.bmp“.

As we are aiming at a small, independent, unicode-coded true-type character set named „GD-Handw-spec-example.ttf“ of our own, and in order to be conforming to the standard, we should choose the *private use area (*hex E000-F8FF) for the coding, and there of course the first position, viz. hex E000. To this end, we open “fontforge“ and select “new”. An empty character set with only those well-known latin characters etc. is displayed. But that's not what we want! Under „encoding“ we select „*Reencode*“, and there „*ISO 10646-1 (Unicode, full)*“. The code space available becomes huge now, and we do need that, because our position hex E000 is “way up at decimal 57,344“. Double click on field E000 opens a window for us, to be creative with our new character's design. By the way, we have now entered the sphere of vector graphics. All lines within characters are formed by „*splines*“ , i.e. closed contours containing “ink”. It's these very splines that enable real true type fonts, fonts that are freely scalable to any size.

Being creative could mean:

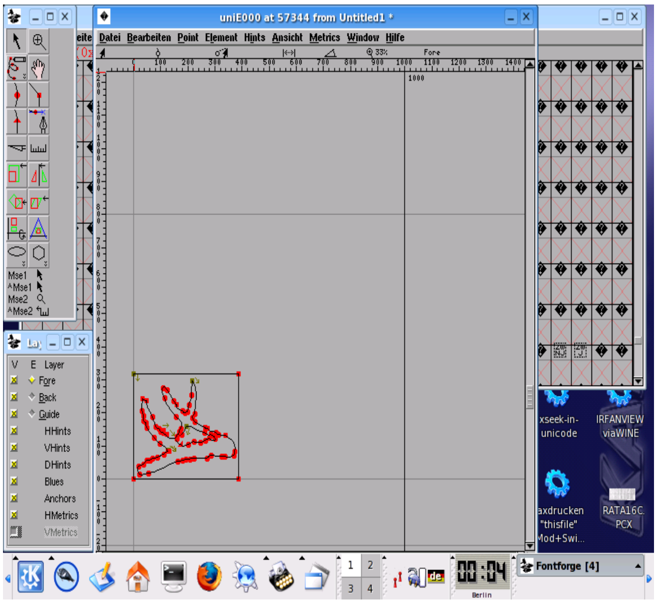
- create a character directly by composing splines
- loading an image file as underlying basis, and hand-trace the image
- have an image file be transformed into splines automatically, and only simplify/polish them as necessary

We take the third approach by using the **autotrace** software. A command line such as

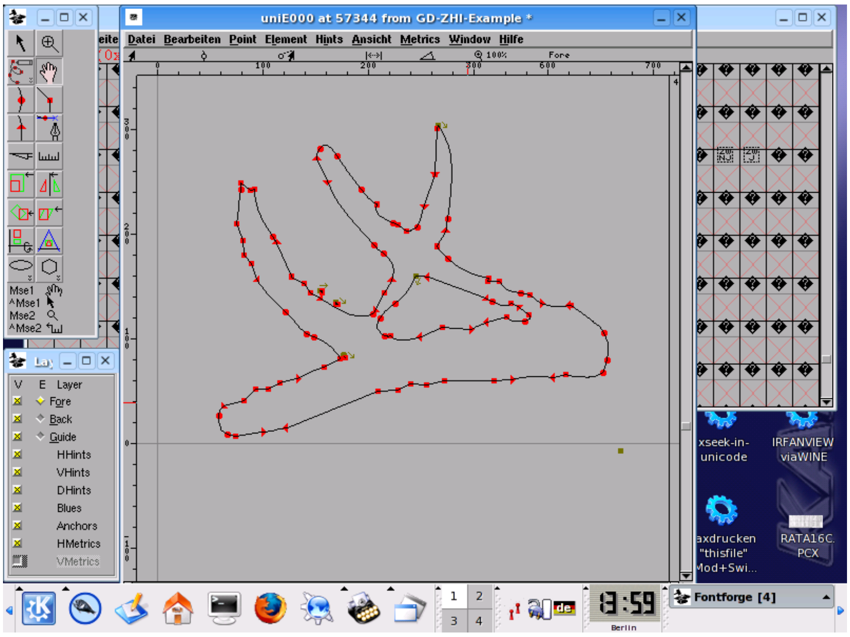
```
autotrace -input-format bmp -output-format eps GD-Kap2-zhi002-BW.bmp >GD-Kap2-zhi002-BW.eps
```

transforms the above „GD-Kap2-zhi002-BW.bmp“ file into a vector graphics one of the „.eps“ type.

We then reopen **fontforge** as described above, and go to the U+E000 code position. Double click on that position opens an editing window. Under “file” we select “import” and load GD-Kap2-zhi002-BW.eps, the file just transformed by autotrace. The maze of lines is maybe still too small, we have to adjust the size. Of course operating **fontforge** needs some practice and research, but all is implemented and documented nicely (documentation either on-line or through download and printout, which is recommended). All tools are on the left hand side, pull-down menues are at the top. The screen shot is a typical view of fontforge, with our “zhi” under creation:



Note one thing: autotrace considers the image's margin part of the image. Therefore move the cursor onto those 4 lines and erase them individually using the DEL key. Also, the number of points should be reduced, making the desing less complicated and more even-looking. Not too much, though, only in cases where jaggedness arose from reproduction, because an even ink flow should be anticipated. A possible result can be seen at the next picture.

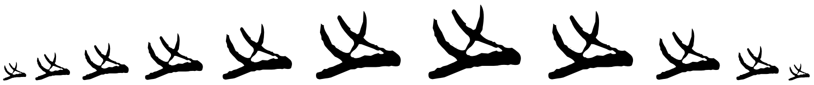


Now we're nearly done. We save our work under a name like GD-Handw-spec-examples.sfd. We then select “file/ „*generate font*“ and get the final result, our little special true-type-font named GD-Handw-spec-examples.ttf.

Note: this .ttf-file is not as large as one might think it might be: although our code position is very high up, our “one-character-only-font” is still surprisingly small in kbytes.

We now install this true-type-font on our computer. This is a little system dependent; KDE has a special menu item for it. Henceforth we can use it as a scalable character like all the other characters.

Proof: You can see our here inserted in a phrase! And it is equally beautiful no matter its size:



key words: chinese characters, classical chinese texts, chinese fonts, free software, open source, linux, freeBSD, KDE, fontforge, autotrace, mined, iceweasel, firefox, open office writer, yet another chinese character's mini-HOWTO.