

最小化修订方案的实施情况介绍

——为最短时间内实现蒙古文编码的相对稳定和统一——

An introduction to the execution of our Solution with Minimal Modifications

——for achieving a stable and unified Mongolian
encoding as soon as possible



梁金宝 (Liang Jinbao)

2019.4.3

目录 | MENU



01

概述
Outline

02

分析与分类存在问题
Problem Classification

03

我们做了哪些工作
What have we done?

04

对未来工作的建议
Advice for future work

概述 | OUTLINE



1. 客观事实

现在的蒙古文编码确实存在一些问题，并且在部分应用场景下从制定最初开始，持续影响着蒙古文在信息技术领域的正确使用。这是客观事实。为了解决这些问题，陆续有不同个人和不同机构在尝试各种改善方案。如纯字形方案、虚拟图形字母方案、改善的语言字母方案、语言音节方案等等。

1. Objective Fact

The current Mongolian code does have some problems, and in some application scenarios, from the beginning of the development, it continues to affect the correct use of Mongolian in the field of information technology. This is an objective fact. In order to solve these problems, different individuals and different institutions are trying to improve various solutions. Such as a pure glyph scheme, a virtual graphic letter scheme, an improved language letter scheme, a language syllable scheme, and the like.

概述 | OUTLINE



2. 我的建议

我认为眼前最急需要做的事情是，在最短时间内对现有方案进行最小化的改善性修订，让蒙古文在最短时间内拥有一个比较稳定和统一的编码方案。先让用户能够比较稳定地使用起来，然后我们再进行更进一步理想化方案的研究、论证、测试工作。那时候我们就不会这么匆匆忙忙去完成制定标准的工作。在参与者如此众多、意见分歧如此大的时候制定新标准，我觉得三年都有点匆忙。但对于现在的蒙古文编码处境来说，让我们等待一年都有点漫长。所以我们的当务之急不是追求理想方案，而是以最小修订内容、在最短时间内实现相对稳定而统一的编码方案。

2. My Advice

In my opinion, the most urgent thing at present is that we have to improve the existing schemes with minimal modifications in the shortest possible time, so that Mongolian will have a stable and unified coding scheme in the shortest time. Let users enjoy stable use of the language, then we could further the idealized scheme with more research, demonstration, and test. At that point we wouldn't be so hurried to set up a standard. We have many units and individuals participated in the re-engineering of standards and implementation standards. It is difficult to resolve the disagreements between them and build one standard in three years. However, under the current circumstances of Mongolian coding situation, even a year will also make us feel too long. We have to build a stable and single standard in the shortest amount of time.

概述 | OUTLINE



3. 问题归类

经过分析各种应用场景下发现的问题，我们很容易知道当前的编码存在的问题大致归类为如下三种类型：

- ① 部分软件系统对复杂文本渲染引擎的支持不够完善
- ② 同形异码问题及由此导致的网络安全问题
- ③ 不同厂家软件系统之间编码互相不兼容

3. Problem classification

After analyzing the problems that have been discovered so far, it is easy to know that the problems existing in coding now are roughly classified into the following three types of problems:

- ① Some software systems are not fully supporting the complex text rendering engines.
- ② The confusable issue and the internet security issue caused by it.
- ③ Incompatible coding between different manufacturers' software products.



1. 部分软件系统对复杂文本渲染引擎的支持不够完善

Some systems are not fully supported for complex text rendering engines

比如windows XP, Photoshop, 嵌入系统控制的LED显示屏, 旧版浏览器等等。我们应该往技术的发展方向看齐, 历史的存量软件的问题总会有其他方法来解决。就像现在大家已经在做的那样。我们改善标准的时候不应该受这些因素的干扰。对此类问题我们没有做任何努力去解决它。

Such as windows XP, Photoshop, embedded system control LED display, old browser and so on. We should follow the trend of the technology development. There are always other ways to solve the problem of historical stock software. Just like people are already doing it now. We should not be disturbed by these factors when we improve our standards. We are not trying to solve these kind of problems.



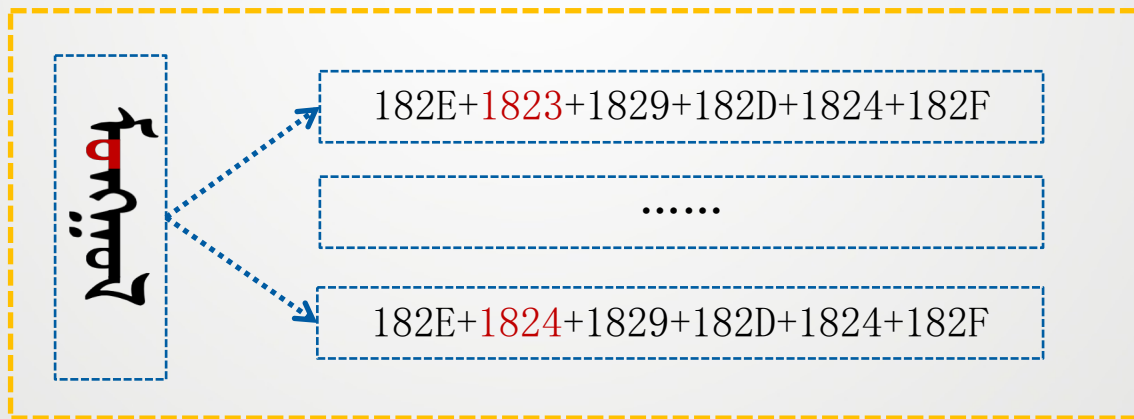


2. 同形异码问题及由此导致的网络安全问题

The confusable issue and the internet security issue caused by it

同形异码问题是当前编码方案本身固有的问题，我们无法通过对当前方案进行修订来解决。我们能做的只有在【忠于语言文字字母结构的编码方案带来的便利性】和【它附加带来的同形异码复杂性】之间的平衡考虑。

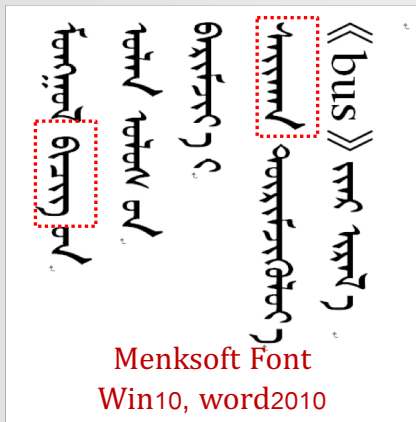
The homomorphic problem is an inherent problem in the current coding scheme, and we cannot solve it by modifying the current scheme. All we can do now is keep the balance between 'the convenience brought by the coding scheme loyal to the linguistic letter structure' and the 'complexity of the isomorphic complexity it brings'.





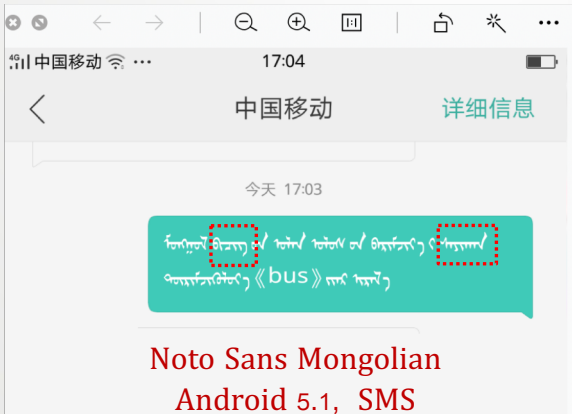
3. 不同厂家软件产品之间编码不兼容

Incompatible coding between different manufacturers' software products 8



只要一般用户一说“编码标准不统一”、“现在的编码方案有缺陷”，那么基本都是被这一类问题所困扰的。而Unicode正式标准里并不包含解决这一问题的规范内容。它是语言文字系统的字母变形规则（也叫转换规则）。只要我们有了一套统一的转换规则去共同遵循，那么这些问题都会迎刃而解。我们最小化修订方案的最主要内容就是规范和统一它。

As long as the average user says “the coding standard is not uniform” and “the current coding scheme is flawed”, it is basically plagued by this type of problem. The Unicode standard content does not contain the normative content to solve this problem. It is the letter deformation rule of the language text system, which is referred to as the conversion rule. As long as we have a unified set of conversion rules to follow together, then these problems will be solved. The main content of our minimization of revisions is to standardize and unify it.



最小化修订方案

The Solution with Minimal Modifications

我们做了哪些修订工作？

依据第二次蒙古文编码工作组会议决议（2018年4月7日，美国）内容，我们对第三类问题（不同厂家软件产品之间编码不兼容）进行了更有针对性的十项工作。我们会在下页开始做详细介绍。

What revisions have we made ?

According to the report of Mongolian Working Group Meeting 2 (April 7,2018, USA), we have carried out more targeted ten items on the third type of problem (Incompatible coding between different manufacturers' software products). It will be introduced on the next page. There is an agreement that the phonetic model should not be abandoned.

The key point of the report for MWG2

1. There is an agreement that the phonetic model should not be abandoned.
2. All attendees want to have a stable Mongolian encoding.
3. For NNBSF, further clarifications are needed and perhaps there are potential changes.
4. We need one set of OpenType rules for rendering Mongolian text based on the current model. Having competing font standards is detrimental to the user community.



1. 用MSC模型解决NNBSP引起的问题

Solve problems caused by NNBSP with MSC model



现状: 在iOS, macOS, Android, Office, Browser等软件的所有或部分版本上, 在蒙古文中使用NNBSP时, 总会出现格附加成分显示错误的情况。这种现象已经持续了很长时间没能彻底解决。尽快并彻底解决此问题的方法只有一个: 放弃使用NNBSP。

Status: When using NNBSP in Mongolian, there will always be cases where the extra component is displayed incorrectly in all or part of the version of iOS, MacOS, Android, Office, Browser. This phenomenon has not been solved for a long time. There is only one way to solve this problem quickly and completely: abandon the use of NNBSP.

解决方法: MVS和NNBSP合并为一个MSC

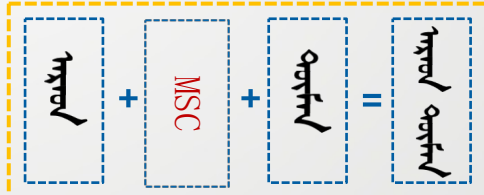
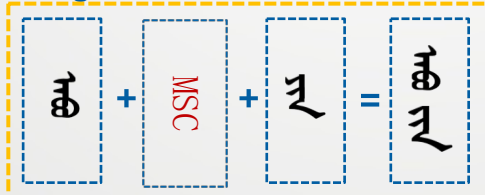
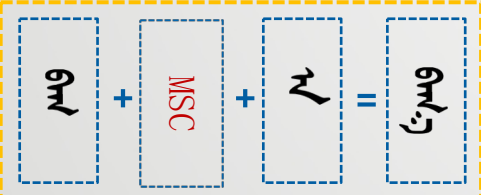
在蒙古文中使用的分隔符有两个: MVS和NNBSP。其功能为①不断词②分割词末元音A/E③分割格附加成分。我们将MVS重新命名为MSC, 码位不变, 在有需求使用MVS和NNBSP时完全用MSC来替代。其优势非常明显:

1. 彻底解决现存问题 (不间断空格&分隔元音和附加成分)
2. 向后兼容已有MVS和NNBSP功能。
3. 方便用户使用。无需记MVS、NNBSP等深奥概念, 只记单词中间有视觉分割的地方就使用MSC, 推荐键位: 破折号(-)。

Solution: Combine MVS&NNBSP into MSC

There are two separators used in Mongolian: MVS and NNBSP. The function is 1.continuous word, 2.split word vowel A/E, 3. split additional component. We renamed MVS as MSC and used MSC wherever there is MVS and NNBSP requirements, which is a complete replacement for these two control symbols.

MSC, Mongolian Suffix Connector, U+180A



我们还有必要舍近求远地使用NNBSP吗?
Do we still need to use NNBSP?



2. 用NIRUGU解决ZWJ引起的问题 Solve problems caused by ZWJ with NIRUGU

182D MONGOLIAN LETTER GA
 → 0433 cyrillic small letter ghe
 ~ 182D first form (isolate)
 ~ 182D first form (initial)
 ~ 182D first form (medial)
 ~ 182D first form (final)
 ~ 182D 180B second form (initial)
 ~ 182D 180B second form (medial)
 ~ 182D 180B second form (final)
 ~ 182D 180C third form (medial)
 ~ 182D 180D fourth form (medial)



200D+182D+200D =>

200D+182D+180D+200D =>

ZWJ+18xx+ZWJ => **OK**

ZWJ+18xx+FVS+ZWJ => **NG**

Win10, Word2010

现状: 在蒙古文中强制生成位置上下文环境时，也就是显示孤立字形（上中下）时，需要使用ZWJ(U+200D)。但是在很多平台上使用的时候，跟NNBSP（U+202F）一样总会出现不稳定的错误现象。并且从编码制定最基础开始一直都没有彻底解决。

Status: ZWJ (U+200D) is required when forcing the location context in Mongolian, that is, when displaying isolated glyphs (initial, middle, final). However, when used on many platforms, there is always an unstable error like NNBSP (U+202F). And it has not been completely solved since the beginning of the code-design.

解决方法 | Solution:

- 用NIRUGU代替ZWJ
- Replace ZWJ with NIRUGU

180A+182D+180A =>

180A+182D+180D+180A =>

OK



3. 明确定义NIRUGU和ZWJ的字符属性

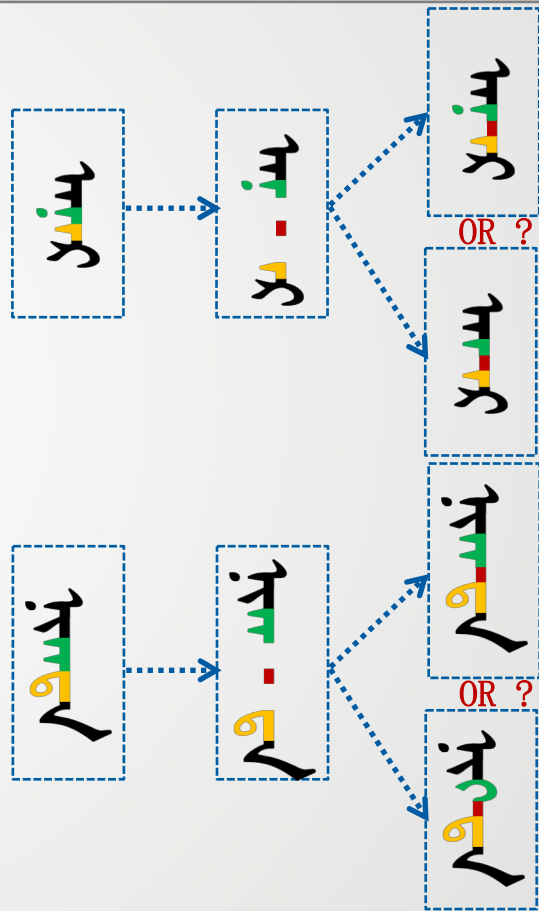
Clearly define the UCD-Attributes of NIRUGU and ZWJ

现状: 蒙古文字母是依据上下文环境来自动变形的。所以我们在词中间插入并不具备语法属性的NIRUGU和ZWJ的时候，我们需要明确定义它的透明属性。即，是上下方向上传递语法属性，还是阻断语法属性。到目前为止并没有明确定义，所以不同厂家的做法是不同的。这也编码不兼容的原因之一。

Status: Mongolian letters are automatically deformed according to the context. So when we insert NIRUGU and ZWJ with no grammatical attributes in the middle of the word, we need to explicitly define its transparent attributes. That is, whether the syntax attribute is passed in the up and down direction or the syntax attribute is blocked. So far there is no clear definition, so the practices of different manufacturers are different. This is also one of the reasons for coding incompatibility.

解决方法 | Solution:

- 建议NIRUGU上下传递语法属性。ZWJ阻断语法属性
- I recommend that NIRUGU pass syntax attributes up and down. ZWJ blocks syntax attributes.





4. 统一连写位置模型和语言学词里位置模型之间的定义分歧

Unify the differences for cursive-joining model and linguistic-joining model¹³

Linguistic-Joining Model						Cursive-Joining Model					
1828	?	MONGOLIAN LETTER		NA		1828	?	MONGOLIAN LETTER		NA	
	No.	form	fvs	psdef	cxdef		No.	form	fvs	psdef	cxdef
ISOL	1	?		★	★	ISOL	1	?		★	★
	2	ᠠ	1				2	ᠠ	1		
INIT	1	?		★	★	INIT	1	?		★	★
	2	ᠠ	1				2	ᠠ	1		
MEDI	1	ᠠ	2	★	★	MEDI	1	ᠠ	2	★	★
	2	ᠠ	1		★		2	ᠠ	1		★
	mvs1	ᠠ	1		★	FINA	1	ᠠ	2	★	★
	mvs2	ᠠ	2				2	ᠠ	1		
FINA	1	ᠠ	2	★	★						
	2	ᠠ	1								

连写位置模型和词里位置模型其实并不冲突。在连写位置模型上认定为连写下的字形，在词里位置模型上完全可以认定为词中字形。这只是互相认同方式的不同而已。在我们的最小化修订方案中，经过对FVS分配方案的调整，这两种模型已经没有任何分歧了。这是在前两次MWG1, MWG2上的重要意见分歧之一。

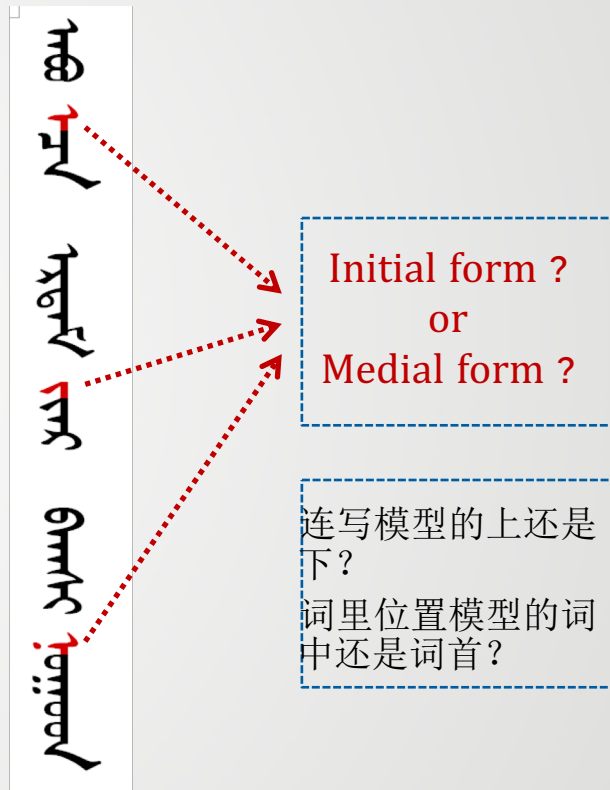
The cursive-joining model and the linguistic-joinig model do not actually conflict. The glyph identified as a final form on the cursive-joining model can be identified as a middle form on the linguistic model. This is only different in the way of identification. In our Solution with Minimal Modifications, after the adjustment of the fvs allocation scheme, there is no disagreement between the two models. This is one of the important differences of opinion on the MWG1 & MGW2.



5. 规范格附加成分首字母字形的词里位置属性 The Solution with Minimal Modifications

连写位置模型偏向于技术实现，词里位置模型偏向于语言学理论。为了让编码方案从技术层面更严谨，更有系统性。所以我们对格附加成分相关变体进行了未调整。但是我们应当清楚，无论从技术层面上怎么调整连写位置属性，他不会影响到语言学上去怎么解释和怎么认定。即使在技术层面上，你将一个字形归类为下形，但是在语言学上我们完全可以叫它为词中字形。这是一点都不冲突的。

The cursive-joining model is biased towards technical implementation, and the linguistic-joining model is biased toward linguistic theory. In order to make the coding scheme more rigorous and systematic, we have slightly adjusted the positional property of first letter for mongolian suffix. However, we should be clear that no matter how technically adjust the positional attributes, he will not affect how to interpret and identify from the linguistics. Even on a technical level, you classify a glyph as a final form, but in linguistics we can call it a middle form. This is not at all conflicting.





6. 语法上下文无关的静态分配方法（FVS）

A Context-Independent static assignment method for FVS

Unicode Code Chart是个非常简单的一张表格，他能承载的信息比较少。为了简化FVS和字母变体之间的对应关系，我们应当采取一个变体固定对应一个FVS的方式。而不是一个变体在这个上下文环境下使用这个FVS，在另一个上下文环境下使用那个FVS。这就是针对FVS用法不统一问题的解决方法。

The Unicode Code Chart is a very simple form that can carry less information. In order to simplify the correspondence between FVS and letter variants, we should adopt a variant to fix the corresponding FVS. Instead of a variant, use this FVS in this context and use that FVS in another context.

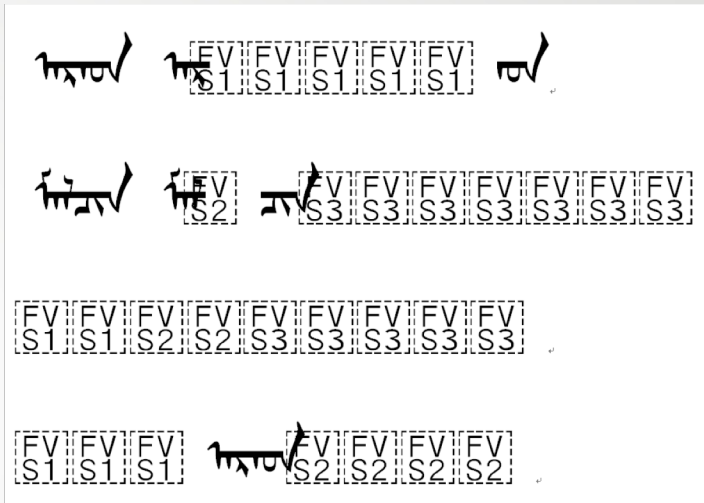
This is the solution to the problem of Inconsistent-Use-of-FVSs.



7. 给无效及多余控制符提供显示提示性错误图形的机制

Display error glyph for invalid and redundant control characters

在现在的编码模型下，不可见的FVS是个非常让人困扰的事情。但是我们可以通过技术手段大大提高用户体验的。虽然是一个小小的优化，但是对提升用户体验效果非常明显，所以这里单独列出来了。解决方法是在用户以错误方式使用FVS时，或用户使用了多余的FVS，那么字体给他显示提示性的错误图形。



In the current coding model, invisible FVS is a very troublesome thing. But we can greatly improve the user experience through technical means. Although it is a small optimization, it is very obvious to improve the user experience, so it is listed separately here. The solution is that when the user uses FVS in a wrong way, or the user uses an extra FVS, the font shows him a hinting error graph.



8. 规范和统一转换规则 Normative and Unified Shapping Rules

实际上，现在的大部分“编码不统一”的原因就在于没有统一转换规则。基于现在的编码方案制定稳定而统一的转换规则是我们今年一年来努力工作的重点。虽然从结构性，易读性等方面有很多需要优化的地方，但是我们线性方式列举了所有变形规则。

In fact, the lack of a unified conversion rule is the major cost of the current “unification of coding”. The development of a stable and unified transformation rule based on the current coding scheme is the key point of our hard work this year.

规则语言: 中文 English 版本: 1.0.3 (已发布版)

独立形式 (2)		词中形式 (4)	
独立规则	词里规则	独立规则	词里规则
MANZ10	MANZ11	MANZ20	MANZ21
ZWJ + 1828		ZWJ + 1828 + FVS1 + ZWJ	
解释	解释	解释	解释
ZH	ZH	ZH	ZH
[+ FVS2] + ZWJ			

规则统计

字母数量	36	变形数量	235
规则数量	独立规则: 235 词中规则: 349	默认变形: 239 强制变形: 110	

关闭

9. 规则测试样例集

The Word test set for shapping rules

无论哪种信息产品，其大规模的系统性测试是必要的。我们也搜集整理了涵盖所有变形规则的单词测试集。并对各厂家字体进行变形规则复合型检测。

No matter what type of information product, it is necessary to do large-scale of systematic testing. We also collected a set of word tests that cover all the deformation rules. And the deformation test compound detection is performed on each manufacturer's word.

	No.	form	fvs	Testing Words							
				ruleid	Expected	NewOyun	NewMSBaiti	MonQagan	DlhQagan	DlhHawang	
ISOL	1	↑	0	MND10	1	?	?	?	?	?	?
				MND11	1	?	?	?	?	?	?
	2	↑	1	MND20	1	↑	↑	↑	↑	↑	↑
				MND21	1	↑	↑	↑	↑	↑	↑
INIT	1	↑	0	MNS10	1	?	?	?	?	?	?
				MNS11	1	↑	↑	↑	↑	↑	↑
	2	↑	1	MNS20	1	↑	↑	↑	↑	↑	↑
				MNS21	1	↑	↑	↑	↑	↑	↑
1	↓	2	MNZ10	1	↑	↑	↑	↑	↑	↑	
			MNZ11	1	↑	↑	↑	↑	↑	↑	↑
				2	↑	↑	↑	↑	↑	↑	↑
3	↑	↑		↑	↑	↑	↑	↑			

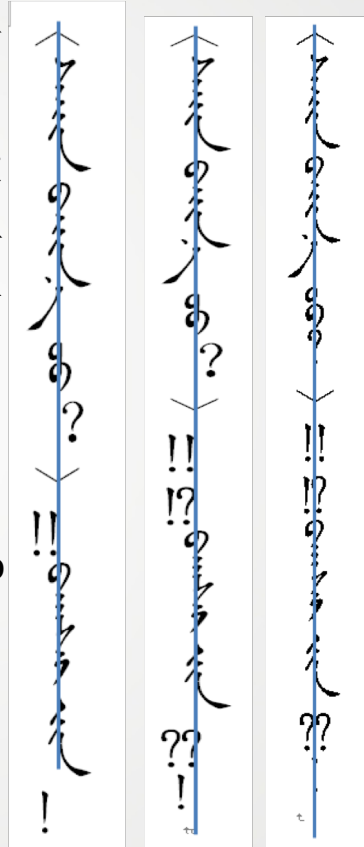


10. 尝试性地规范了标点符号的字形要求及使用方法

Exploratoryly clarifying the specification of punctuation in Mongolian

蒙古文排版中使用的标点符号有两种。一种是蒙古文本身有的，比如“、”。另一种是从其他文种编码区域借用的。比如：《，》。但是目前还没有很详细的标点符号使用规范。比如视觉设计要求，使用要求等。我们知道这些可能不属于编码标准的范围内内容，但是它对蒙古文使用很重要。所以我们尝试性地制作了标点符号推荐性规范文档。

There are two types of punctuation marks used in Mongolian typesetting. One is the Mongolian text, such as “、”. The other is borrowed from other language coding areas. such as:“,”. However, there is currently no detailed specification for the use of punctuation. Such as visual design requirements, usage requirements, etc. We know that these may not be within the scope of the coding standard, but it is important for Mongolian use. So we tentatively made the punctuation recommendation specification document.



1. 竖排对齐特征
2. 字形内部空白区域比例特征
3. 使用方式（前后空格）

1. Vertical alignment feature
2. Proportional feature of blank area inside glyph
3. How to use the text (pre-space, after-space)

开发&测试&改进&实施

Development&Testing&Improvement&shishi

自从上次第二次蒙古文编码工作会议之后，我们并没有停留在方案研究等基础性工作上。在一年的时间内，我们联合各个字体厂商和相关研究机构通力合作，进行了最小化修订方案的可行性论证、产品开发、产品测试、产品实施等各个环节的工作。目前已经制作了相互完全兼容的一百多套字体，并在三十多家企事业单位的五十多个工程项目中推荐使用。其中包括制作七千多种蒙古文电子图书的工程。我们的字体将会很快像社会无偿提供。

Since the last MWG2, we have not stayed in basic work such as program research. In a year's time, we worked together with various font manufacturers and related research institutions to carry out the work of the-Solution-with-Minimal-Modifications of demonstration, development, testing, implementation and other aspects. At present, more than 100 sets of fonts that are fully compatible with each other have been produced and recommended in more than 50 engineering projects of more than 30 enterprises and institutions. This includes the production of more than 7,000 Mongolian e-books. Our fonts will soon be available free of charge to society.



德力海公司字体 (27)

The Fonts for Delehi Company (27)

 T 传统蒙古文白正体	 T 传统蒙古文黑正体	 T 传统蒙古文新白体	 T 传统蒙古文新黑体	 T 传统蒙古文报白体
 T 传统蒙古文报黑体	 T 传统蒙古文标黑体	 T 传统蒙古文等线体1	 T 传统蒙古文哈旺体	 T 传统蒙古文回鹘体
 T 传统蒙古文空心体	 T 传统蒙古文美术体	 T 传统蒙古文三国体	 T 传统蒙古文手楷体1	 T 传统蒙古文行书体
 T 传统蒙古文竹笔体	 T 传统蒙古文美术体2	 T 传统蒙古文黑体	 T 传统蒙古文手楷体2	 T 传统蒙古文报文体
 T 传统蒙古文标题	 T 传统蒙古文等线体2	 T 传统蒙古文白体	 T 传统蒙古文楷体	 T 传统蒙古文ZS
 T 传统蒙古文ZX	 T 传统蒙古文图门硬笔			



蒙科立公司字体 (28)

The Fonts for Menksoft Company (28)

  蒙古文白体 T 包括传统蒙古文、托忒文、锡伯文、满文以及蒙、托、满三种文字阿礼嘎礼字母	  传统蒙古文报体 T	  传统蒙古文黑体 T	  传统蒙古文哈旺体 T	  传统蒙古文标题体 T
  传统蒙古文竹体 T	  传统蒙古文细体 T	  传统蒙古文乌仁体 T	  传统蒙古文木刻体 T	  传统蒙古文粗体 T
  传统蒙古文其木德体 T	  传统蒙古文阿木古楞体 T	  传统蒙古文龙德布体 T	  传统蒙古文宽干体 T	  传统蒙古文其木格体 T
  传统蒙古文图如体 T	  传统蒙古文硬笔体 T	  传统蒙古文棒体 T	  传统蒙古文编结体 T	  传统蒙古文云体 T
  传统蒙古文乌云嘎体 T	  传统蒙古文索那体 T	  传统蒙古文舒体 T	  传统蒙古文清明体 T	  传统蒙古文牌匾体 T
  传统蒙古文马镫体 T	  传统蒙古文露体 T	  传统蒙古文额日和体 T		



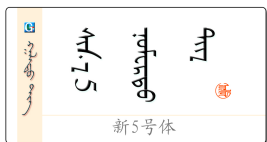


嘎拉图公司字体 (50)

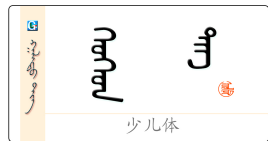
The Fonts for Galtu Company (50)



木板体



新5号体



少儿体



清鉴体



扁宋体



求书



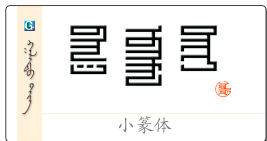
甘珠儿体



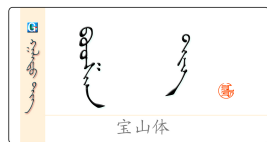
大篆体



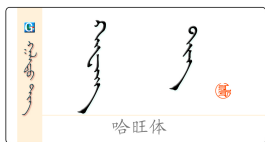
蒙古国体



小篆体



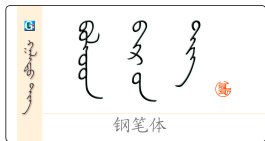
宝山体



哈旺体



北京体



钢笔体



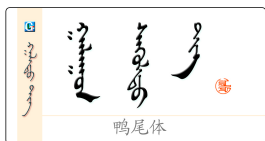
铅刻体



回鹘式体



嘎拉图体



鸭尾体



细圆体



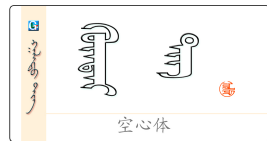
三国体



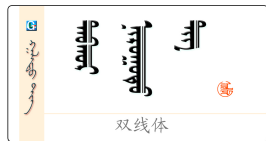
书信体



卡通体



空心体



双线体



空腰体



黑体



琥珀体



五线体



板笔白体

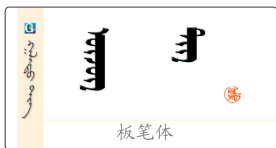


1号标体



嘎拉图公司字体 (50)

The Fonts for Galtu Company (50)



板笔体



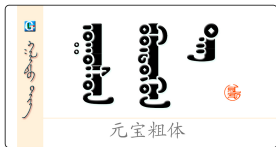
积木细体



积木粗体



元宝细体



元宝粗体



雪峰体



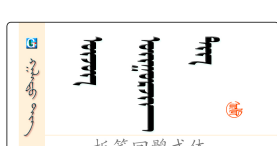
白体



板笔辩体



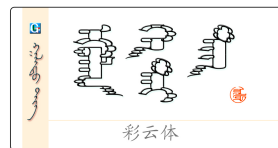
板笔黑体



板笔回鹘式体



秀丽体



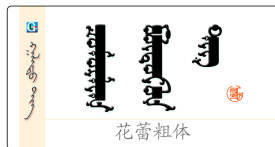
彩云体



牌匾粗体



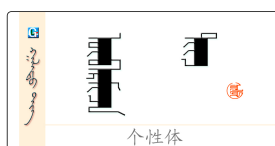
牌匾细体



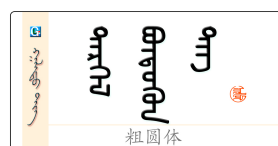
花蕾粗体



壮体



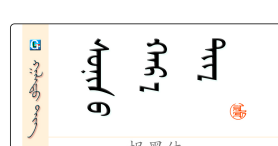
个性体



粗圆体



上黑体



报黑体



字体是编码方案的实现方式，并且制作一套即美观又符合OpenType标准的字体是个费时费力的工作，艺术和技术哪个都不能欠缺。如果标准是统一的，那么字体也是通用的。没有必要每个人、每个公司、每个机构都要制作自己的字体。我们制作的这些上百种蒙古文字体，都是向全世界免费公开发布的，只要有使用需求的地方都可以无偿使用。

Fonts are the way to implement coding schemes, and it is a time-consuming and laborious task to create a set of fonts that are aesthetically pleasing and conform to the OpenType standard. Art and technology can't be lacking. If the standard is uniform, the font is also generic. There is no need for everyone, every company, every institution to make their own fonts. The hundreds of Mongolian scripts we produce are all freely released to the world, and can be used free of charge as long as there is a need for use.



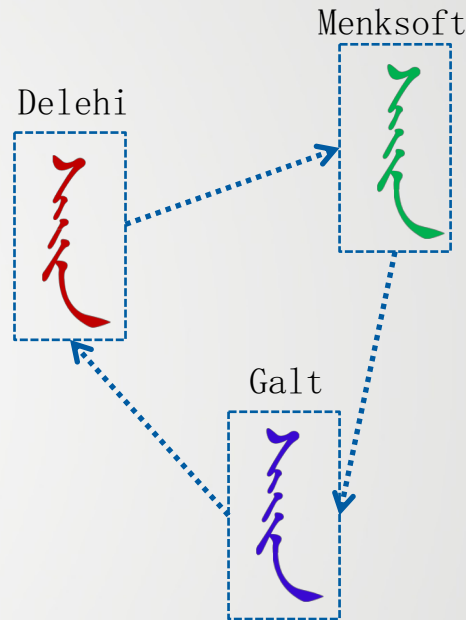
已实现的 & 还未实现的 Implemented & Not Yet Implemented

我们已做到的 | What We Have Done

1. 不同厂商产品之间的编码兼容
Code compatibility between different vendors' products
 2. 各大操作系统平台上不乱码（字体升级到本方案后）
No garbled on major operating system platforms (after font upgrade)
- ※我们认为在基于现在编码模型，这是我们以最小的代价能够做到的最好的效果。
We think that based on the current coding model, this is the best we can do at the least cost.

我们没能做到的 | What we can't implement:

1. 让XP和Photoshop等软件直接支持unicode蒙古文
Let XP and Photoshop directly support unicode Mongolian
2. 让嵌入系统控制的LED显示屏直接显示unicode蒙古文
Let the embedded system control LED display directly display unicode Mongolian.
3. 在现在的编码模型上消除同形异码现象
Eliminate the confusable issue on current coding models



What about on iOS | Android | Windows | Linux ?

下一步计划的意见

The Advice For Next Step

事情比较复杂、参与的人多、分歧也比较大。我建议分阶段来实施。这样我们容易取得阶段性成果。我建议**分三步进行**。这样我们步步为营，大家一起努力一步一步地往前推进此项工作。

Things are more complicated, there are more people involved, and the differences are larger than in the past. I recommend that implement our assignment in several stages. It will more easier for us to achieve a phased result. I recommend that put our task **in three steps**. In this way, we can work step by step, and everyone can works together to advance our task.

第一步: 聚焦在码表

The First Step: Focus on the code charts

在第一步, 我们应该集中精力在Unicode Code Chart ①上达成一致意见。即我们在基于这张表格解答别人的如下三个疑问的时候, 我们的解答应该是一致的:

1. 蒙古文字母有哪些? 它的码位是什么?
2. 每个蒙古文字母在不同词里位置上都有哪些变体?
3. 当某一个变体不能通过上下文来正确显示的时候, 我们应该用哪个FVS来强制请求出这个字形? (静态分配和动态分配的差异)

1828	†	MONGOLIAN LETTER NA
		→ 043D Н cyrillic small letter en
②	~ 1828	† first form (initial)
	~ 1828	† first form (medial)
	~ 1828	✓ first form (final)
	~ 1828	† 180B † second form (initial)
	~ 1828	† 180B † second form (medial)
	~ 1828	† 180C ✓ third form (medial)
	~ 1828	† 180D † fourth form (medial)

In the first step, we should focus on reaching a consensus on the Unicode Code Chart. That is, when we answer the following three questions from others based on this form, our answers should be consistent :

1. What are the Mongolian letters? What is its code point ?
2. What variants do each Mongolian letter have in the position of cursive-joining model ?
3. When a variant cannot be properly displayed by context-driven, which FVS should we use to force the glyph to be requested ? (fvs static allocation & fvs dynamic allocation)



在第二部，我们在转换规则（变形规则）层面达成一致意见，并形成统一的Unicode Technical Note。在上次MWG2的会议决议里也提到此问题，但是我们忙于对技术笔记的制定，测试的多项工作，没能完成此项任务。但是现在我们已经完成了前期工作，所以已经具备了开始制作的条件。希望多方一起努力完成此项工作。

In the second step, we agreed on the transferring rules (shaping rules) level and formed a unified Unicode Technical Note. This issue was also mentioned in the last MWG2 Report, but we are busy with the development of technical notes, testing a lot of work, failed to complete this task. But now that we have completed the preliminary work, we have the conditions to start production. I hope that many parties will work together to complete this work.

The point of the report for MWG2 (April 7,2018, USA)

- It was decided that creating a Unicode Technical Note that would be a visual guide to the behavior of the Mongolian script would be very beneficial. Liang Jinbao and Kamal Mansour agreed to cooperate to create such a document on Mongolian script behavior.

完成了前两步工作，我们就有了充足的时间去研究更多的、更理想化的编码方案。我们在追求理想方案的道路上，如果只看远方不看眼前，那么我们有可能总会原地踏步……

After completing the first two steps, we have plenty of time to study more and more idealized coding schemes. On the road to pursuing the ideal solution, if we only focus on the distance and do not pay attention to the front, then we may always staying in the same place..

非常感谢 | THANKS

