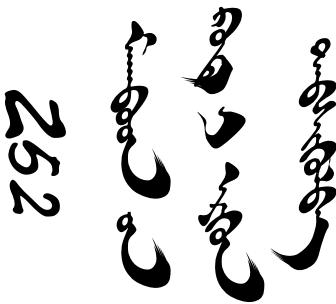


Unicode Consortium – WG2 Proposal



Title: Proposal for a Parallel, Element-Based Encoding Model for the Traditional Mongolian Script (Z52)

Author: Chuck Namujila / Zcode Team (Chuck.Namujila@gmail.com)

Date: 2026-Jan-02

Proposal Type: Technical / Encoding Model Extension

Target Standard: ISO/IEC 10646 / Unicode Standard

1. Proposal Summary

This proposal requests the allocation of a parallel, element-based Unicode encoding model for the Traditional Mongolian script. This model is designed to coexist with the existing Mongolian block (U+1800 range) rather than replace it. It aims to resolve systemic architectural limitations regarding font-level dependency, non-deterministic rendering, and data ambiguity that have hindered the script's digital viability for over two decades.

2. Script Identification

- **Script Name:** Traditional Mongolian
- **ISO 15924 Code:** Mong
- **Directionality:** Vertical, top-to-bottom, left-to-right columns.
- **Current Unicode Range:** U+1820 – U+1842 (Abstract Phonetic Model), U+180B – U+180F
- **Proposed Unicode Range:** A new block of 52 code points, preferably allocated within the Supplementary Multilingual Plane (SMP/Plane 1).

3. Background

The Traditional Mongolian script, derived from Sogdian, is a complex cursive system where letters possess distinct isolate, initial, medial, and final forms. When encoded in Unicode circa 2000, scholarly disagreement led to a compromise: Unicode encoded only "abstract base letters," delegating the resolution of actual written forms to font-level shaping logic (OpenType). However, these written variants are not optional "styles"; they are fundamental orthographic requirements for literacy.

In practice, font-level shaping logic has depended on implicit or explicit agreement among users, font developers, and scholarly communities regarding how abstract letters should be resolved into written forms. Such agreement has proven difficult to achieve and sustain consistently, particularly across regions, platforms, and generations. As a result, rendering behavior has remained fragmented and inconsistent.

4. Problem Statement

The current "Abstract Phonetic" architecture exhibits five systemic failures:

1. **Font-Level Semantics:** Fonts are forced to perform linguistic and grammatical interpretation, effectively acting as the "standard" rather than the carrier of style.
2. **Non-Deterministic Rendering:** The same Unicode sequence renders differently across various fonts, operating systems, and browsers.
3. **Data Ambiguity:** Stored text does not uniquely represent written meaning without the specific font context.
4. **Platform Dependency:** High-fidelity rendering requires advanced OpenType support, which remains inconsistently implemented on mobile and embedded devices.
5. **AI/NLP Unsuitability:** Ambiguous encoding degrades corpus quality, making the script incompatible with modern search indexing, machine translation, and Large Language Models (LLMs).
6. **Dependency on User Agreement:** The correctness of font-level shaping relies on prior agreement among users and font developers regarding grammatical interpretation and written form selection. In the absence of consensus, fonts encode divergent assumptions, leading to incompatible renderings and persistent fragmentation.

5. Proposed Solution: The Element-Based Model

We propose a secondary encoding model that explicitly represents written Mongolian forms. Rather than encoding abstract phonetic concepts, this model adopts a deterministic, element-based architecture that reflects how the script is actually written and read.

The transition from the current model to the Z52 model follows three clear architectural steps:

Step 1 — Form-Based Encoding

Rather than encoding 35 abstract phonetic letters, the encoding directly represents the orthographic reality of the script by operating at the level of initial, medial, and final written forms.

Step 2 — Elemental Decomposition

Instead of encoding all 236 initial–medial–final variants as independent code points, the model encodes 52 fundamental structural elements. These elements are combined deterministically to construct all 236 normalized written variants.

Step 3 — IME-Centered Composition

Rather than relying on highly complex font shaping logic to resolve written forms at render time, an IME-based approach preselects and assembles the required variants from the 52 elements during input. This keeps Unicode encoding stable while significantly reducing the complexity and burden placed on font implementations.

Under Z52 architecture, correctness no longer depends on post hoc agreement at the font or user level. Written forms are explicitly constructed at the point of input using standardized elements, ensuring that identical Unicode sequences always represent identical written outcomes. As a result, interoperability is achieved by design rather than by consensus. This shift transforms correctness from a social negotiation into a technical guarantee.

This element-based design allows Unicode to remain simple, deterministic, and stable, while relocating linguistic and grammatical intelligence to the input layer where it can be managed explicitly and transparently.

6. Architectural Principles

- **Orthographic Reality:** Unicode shall encode the written form directly, ensuring that stored data represents the script as it appears.
- **Separation of Concerns:**
 - **Unicode:** Provides stable, deterministic representation.
 - **FONTS:** Limited to visual presentation (style/quality) without performing linguistic logic.
 - **Input Method Editors (IMEs):** Relocates the "intelligence"—managing grammatical rules, regional conventions, and scholarly disagreements—to the input layer.
- **Data Stability:** Equivalent written forms map to identical Unicode sequences, ensuring reliable normalization and cross-platform stability.
- **Agreement-Free Interoperability:** The Z52 model eliminates reliance on user or font-level agreement for correct rendering. Orthographic decisions are resolved deterministically during input, allowing systems to interoperate without shared assumptions or negotiated conventions.

Z52 design relocates structural complexity to a one-time preprocessing phase at the point of input, following a form-based, element-compositional, and IME-centered architecture. Through the assembly of Mongolian letters from Z52 Lego-Like elements during input, the system minimizes downstream complexity in font design, Unicode encoding, and text rendering. This approach reduces the overall cost and technical burden of developing, maintaining, and using the Traditional Mongolian script, thereby supporting wider participation and adoption, especially among Generation Z younger users.

	Current Design for Traditional Mongolian	Z52 Design
Input Method Editor	Mid	Heavy
Font	Heavy	Light
Unicode	Heavy	Light
Input User (Writer)	Heavy	Light
Output User (Reader)	Heavy	Light
Development	Heavy	Light
Maintenance	Heavy	Light
User Agreement	Heavy	Nil

7. Mission Statement: Technical Preservation

Digital fragility leads to social obsolescence. If a script cannot function smoothly on a smartphone—becoming difficult to input or search—it risks being abandoned by younger generations which is already happening. Zcode (Z52) is a preservation effort to ensure the Mongolian script remains relevant in the era of mobile computing and Artificial Intelligence. By moving high-friction scholarly disagreements to the IME layer, we restore architectural clarity to Unicode and usability for end users.

8. Technical Overview (Z52)

This section provides the technical grounding for the three-step element-based architecture described in Section 5, demonstrating how 52 atomic components fully and deterministically generate all written Mongolian forms.

The Z52 model identifies 52 atomic components derived from a historical and structural analysis of Mongolian letter construction.

- **Current Problematic Scope:** The existing Unicode model encodes 35 abstract phonetic letters (U+1820–U+1842), together with five control characters—FVS1–4 and MVS (U+180B–U+180F)—which introduce external, non-orthographic control mechanisms into the writing system.
- **Resulting Expansion:** Each abstract letter requires up to four positional forms (isolate, initial, medial, final), yielding approximately 236 distinct written variants after normalization.
- **Z52 Resolution:** These 236 variants can be constructed deterministically from a closed set of 52 atomic elements, eliminating the need for abstract phonetic indirection or font-level semantic interpretation.

9. Relationship to Existing Standard

This model is **non-disruptive**. The existing Mongolian block (U+1800) remains valid for legacy data. The Z52 model is intended for modern applications requiring data integrity and AI compatibility.

10. Rendering and Implementation

Rendering becomes straightforward.. This eliminates the need for complex OpenType GSUB/GPOS features, ensuring that Mongolian text displays correctly even in simple environments like basic text editors or legacy browsers.

Z52 **avoided** font-level OpenType advanced mechanisms such as:

- Chained contextual substitutions (calt, ccmp, rlig)
- Kerning and contextual positioning (kern, GPOS)
- Single glyph substitutions (GSUB Single Substitution)
- Multiple glyph substitutions (GSUB Multiple Substitution)
- Ligature substitutions (liga, rlig)

11. Backward Compatibility and Migration

Migration is optional and driven by IME adoption. Users can choose to use Z52-compliant IMEs for new data creation while maintaining legacy text in its original format.

12. Benefits to Unicode and ISO/IEC 10646

- **Restores Architectural Correctness:** Aligning Mongolian with other deterministic scripts.
- **Eliminates Conflict:** Removes the burden of linguistic interpretation from font developers.
- **Enables Global Search:** Ensures 100% search accuracy across all platforms.
- **Future-Proofing:** Provides the clean, structured data required for AI and NLP.
- **Removes Consensus Dependency:** Correct rendering no longer depends on reaching or maintaining agreement among users, font vendors, or platforms, significantly reducing long-term fragmentation and implementation risk.

13. Conclusion

The Z52 proposal offers a technically conservative, backward-compatible, and architecturally stable path forward. By decoupling linguistic logic from the encoding layer and placing it within the IME, we ensure the Traditional Mongolian script is no longer a "headache" for the Unicode ecosystem, but a vibrant, functioning part of the global digital landscape.

Appendix 1: Z52 Zcode list

Z52 proposal introduces a total of **52 new Unicode code points**, each representing a distinct structural element used in the construction of Traditional Mongolian letters.

No.	Section	ZCode Name	ZCode Character	English Name/ Notes	Temp Unicode in Font
Core Stroke and Structural Components					
1	Core Stroke	titem ፩፻፻	ት	Crown	U+1865
2	Core Stroke	odoi_suul ፩፻፻ ፩፻፻	ጀ	Short tail	U+1866
3	Core Stroke	orhich, ፩፻፻፻	ጀ	Tail	U+1867
4	Core Stroke	vrvgsilga_or_chachilga ፩፻፻፻፻፻ / ፩፻፻፻፻፻	ጀ	Leash	U+1868
5	Core Stroke	achvg ፩፻፻፻	ጀ	Tooth	U+186A
6	Core Stroke	hvms ፩፻፻፻	ጀ	Claw	U+186B
7	Core Stroke	si l bi ፩፻፻፻	ጀ	Shin	U+186C
8	Core Stroke	gedes ፩፻፻፻	ጀ	Belly	U+186D
9	Core Stroke	nuruu ፩፻፻፻	ጀ	backbone	U+186E
Nasal Components					
10	N	n_titem ፩፻፻፻፻ ፩፻፻	ጀ	Dotted_Crown	U+186F
11	N	n_orhich ፩፻፻፻፻ ፩፻፻፻	ጀ	Dotted_tail	U+1870
12	N	n_achvg ፩፻፻፻፻ ፩፻፻	ጀ	Dotted_tooth	U+1871
B / P Components					
13	B	b_banbvr_egsilge ፩፻፻፻፻፻፻፻	ጀ	b final form	U+1872
14	B	b_banbvr_baga ፩፻፻፻፻፻፻፻ ፩፻፻	ጀ	b base / small	U+1873
15	B	b_banbvr_ih ፩፻፻፻፻፻፻፻ ፩፻፻	ጀ	b base / big	U+1874
16	P	p_chihi_egsilge ፩፻፻፻፻፻፻፻፻	ጀ	p final form	U+1875
17	P	p_chihi_baga ፩፻፻፻፻፻፻፻ ፩፻፻	ጀ	p base / small	U+1876
18	P	p_chihi_ih ፩፻፻፻፻፻፻፻ ፩፻፻	ጀ	p base/ big	U+1877
H / G Components					
19	H / G	h_hanhar ፩፻፻፻፻፻፻፻	ጀ	h base	U+1878
20	H / G	agsilga ፩፻፻፻፻፻	ጀ	h final	U+1887
21	H / G	g_hanhar ፩፻፻፻፻ ፩፻፻፻፻፻	ጀ	g base	U+1888
22	H / G	dvsaltai_agsilga ፩፻፻፻፻ ፩፻፻፻፻	ጀ	g final alternate	U+1889
23	H / G	hinhir ፩፻፻፻	ጀ	he/ge base small	U+188A
24	H / G	hunhur ፩፻፻፻	ጀ	he/ge base big	U+188B
M Components					
25	M	m_titem ፩፻፻፻ ፩፻፻	ጀ	PonyTail_base	U+188C

26	M	m_hurbelge ᠮᠤᠷᠪᠡᠯᠵᠡ	ᠮ	PonyTail_final	U+188D
27	M	m_achvg ᠮᠱᠳᠶ	ᠮ	PonyTail_medial	U+188E
L Components					
28	L	l_titem ᠥᠰᠳᠶ	ᠯ	Horn_base	U+188F
29	L	l_hurbelge ᠮᠤᠷᠪᠡᠯᠵ	ᠯ	Horn final	U+1890
30	L	l_achvg ᠮᠱᠳᠶ	ᠯ	Horn medial	U+1891
S / X Components					
31	S / X	s_agshilga ᠮᠱᠳᠶᠱᠳᠶ	ᠰ	s final	U+1893
32	S / X	s_sechig ᠮᠱᠳᠶ	ᠰ	s base	U+1894
33	S / X	x_agshilga ᠮᠱᠳᠶᠱᠳᠶ	ᠸ	x final	U+1896
34	S / X	x_sechig ᠮᠱᠳᠶᠱᠳᠶ	ᠸ	x base	U+1897
T / D Components					
35	T / D	t_senj ᠮᠱᠳᠶᠱ	ᠲ	t base	U+1898
36	T / D	d_senj ᠮᠱᠳᠶᠱ	ᠳ	d base	U+1899
Affricates / Approximants (塞擦音 / 近音)					
37	Affricates / Approximants	q_acha ᠮᠱᠳᠶ	ᠴ	q / ch base	U+189A
38	Affricates / Approximants	j_bosoo_silbi ᠮᠱᠳᠶ	ᠵ	j base	U+189B
39	Affricates / Approximants	y_erteger_silbi ᠮᠱᠳᠶ	ᠶ	y base	U+189C
40	Affricates / Approximants	r_odoi_suul ᠮᠱᠳᠶ	ᠷ	r final	U+189D
41	Affricates / Approximants	r_erbgeljin_silbi ᠮᠱᠳᠶ	ᠷ	r base	U+189E
42	Affricates / Approximants	w_gohtai_silbi ᠮᠱᠳᠶ	ᠸ	w base	U+189F
Loan / Extended Letters					
43	Loan / Extended	f_chihi_egsilge ᠮᠱᠳᠶᠱ	ᡩ	f final	U+18A0
44	Loan / Extended	f_chihi_baga ᠮᠱᠳᠶᠱ	ᡩ	f base small	U+18A1
45	Loan / Extended	f_chihi_ih ᠮᠱᠳᠶᠱ	ᡩ	f base big	U+18A2
46	Loan / Extended	k_egsilge ᠮᠱᠳᠶ	ᡴ	k final	U+18A3
47	Loan / Extended	k_baga ᠮᠱᠳᠶ	ᡴ	k base small	U+18A4
48	Loan / Extended	k_ih ᠮᠱᠳᠶ	ᡴ	k base big	U+18A5
49	Loan / Extended	c_acha ᠮᠱᠳᠶ	ᠴ	ts / c base	U+18A6
50	Loan / Extended	z_acha ᠮᠱᠳᠶ	ᠵ	z base	U+18A7
Galig Extensions					
51	Galig	galig_tvvrai ᠮᠱᠳᠶ	ᠵ	Galig zhi	U+18A8
52	Galig	galig_r_malgai ᠮᠱᠳᠶ	ᠵ	Galig Jhi	U+18AA

Zcode Naming combination is here: <https://zvvnmod.com/#/article?id=67da7a649c0c1500019fbb28>

Related Mongolian script training is here: https://www.youtube.com/watch?v=lnePiQa_WT8

<https://www.youtube.com/watch?v=eAvWO3Mm5aM>

Appendix 2: Mapping table from Existing Unicode to Zcode Proposal

Mapping tables from legacy Mongolian Unicode Standard to Zcode Composition

No	Unicode Standard (As-Is)	Letter Group	Variant	Position	Zcode Composition (To-Be)	Zcode Component Names	Temp Unicode in Font
— Vowels (元音) - Letter A							
1	U+1820 + FVS3	A	A-1	Isolate_0	ꝛ + ՚	titem + orhich	U+1865 + U+1867
2	U+1820 + FVS1	A	A-2	Isolate_1	ꝛ + ՚	titem + vrvgsilga_or_chachilga	U+1865 + U+1868
3	U+1820 + FVS2	A	A-3	Isolate_2	՚	vrvgsilga_or_chachilga	U+1868
4	U+1820	A	A-4	Initial_0	ꝛ + ՚	titem + achvg	U+1865 + U+186A
5	U+1820 + FVS1	A	A-5	Initial_1	՚	titem	U+1865
6	U+1820	A	A-6	Medial_0	՚	achvg	U+186A
7	U+1820 + FVS1	A	A-7	Medial_1	՚ + ՚	achvg + achvg (used for 2 nd half personal / mountain names, ᠠርዕስኑ እና አርዕስኑ)	U+186A + U+186A
8	U+1820 + FVS2	A	A-8	Final_0	՚	orhich	U+1867
9	U+1820 + FVS1	A	A-9	Final_1	՚ + ՚	achvg + vrvgsilga_or_chachilga	U+186A + U+1868
10	U+1820	A	A-10	Final_2	՚	vrvgsilga_or_chachilga	U+1868
— Vowels (元音) - Letter E							
11	1821 + FVS2	E	E-1	Isolate_0	՚ + ՚	titem + vrvgsilga_or_chachilga	U+1865 + U+1868
12	1821 + FVS1	E	E-1	Isolate_1	՚	vrvgsilga_or_chachilga	U+1868
13	1821 + NIRUGU	E	E-2	Initial_0	՚	titem	U+1865
14	1821 + FVS1 + NIRUGU	E	E-3	Initial_1	՚ + ՚	titem + achvg	U+1865 + U+186A
15	U+1821	E	E-4	Medial_0	՚	achvg	U+186A
16	U+1821 + FVS2	E	E-6	Final_0	՚	orhich	U+1867
17	U+1821 + FVS	E	E-7	Final_1	՚	vrvgsilga_or_chachilga	U+1868

18	MVS + U+1821	E	E-8	Final_2	ঁ	vrvgsilga_or_chachilga	U+1868
— Vowels (元音) - Letter I							
19	U+1822 +FVS3	I	I-1	Isolate_0	ঁ + ঁ	titem + hvms	U+1865 + U+186B
20	U+1822 +FVS1	I	I-2	Isolate_1	ঁ	hvms	U+186B
21	U+1822 +FVS2	I	I-3	Isolate_2	Nil	Not exist in Traditional Mongolian Scrip	Nil
22	U+1822 + [FVS2] + NIRUGU	I	I-4	Initial_0	ঁ + ঁ	titem + silbi	U+1865 + U+186C
23	U+1822 + FVS1 + NIRUGU	I	I-5	Initial_1	ঁ	silbi	U+186C
24	NIRUGU + 1822 + FVS3 + NIRUGU	I	I-6	Medial_0	ঁ	silbi	U+186C
25	NIRUGU + 1822 + FVS2 + NIRUGU	I	I-7	Medial_1	ঁ + ঁ	silbi + silbi	U+186C + U+186C
26	NIRUGU + 1822 + FVS1 + NIRUGU	I	I-8	Medial_2	ঁ + ঁ	achvg + silbi	U+186A + U+186C
27	NIRUGU + U+1822	I	I-9	Final_0	ঁ	hvms	U+186B
28	U+1822 <i>(this scenario does not exist in Unicode standard, it was covered in OpenType by Font.)</i>	I	I-10	Final_1	ঁ + ঁ	silbi + Odoi_suul (ঁ = ঁ + ঁ + ঁ)	U+186C + U+1866

— Vowels (元音) - Letter O/U

- Unification – The vowels O and U are unified into a single letter group, as their written forms are same in Traditional Mongolian script. Unicode encoding represents orthographic form rather than phonetic variation, and therefore does not distinguish between different vowel sounds when the glyph structure is the same.

29/ 30	U+1823 / U+1824	O/U	O-1	Isolate_0	ঁ + ঁ	titem + gedes	U+1865 + U+186D
31/ 32	U+1823 + FVS1	O/U	O-2	Isolate_1	ঁ	b_banbvr_baga	U+1873

	/ U+1824 + FVS1						
33/ 34	U+1823 + FVS1 / U+1824 + FVS2	O/U	O-3	Isolate_2	Nil	Not exist in Traditional Mongolian Scrip	Nil
35/ 36	U+1823 / U+1824	O/U	O-2	Initial_0	↑ + σ	titem + gedes	U+1865 + U+186D
37/ 38	U+1823 / U+1824	O/U	O-2	Initial_1	σ	gedes	U+186D
39/ 40	U+1823 / U+1824	O/U	O-3	Medial_0	σ	gedes	U+186D
41/ 42	U+1823 / U+1824	O/U	O-4	Medial_1	↑ + σ	achvg + gedes	U+186A + U+186D
43/ 44	U+1823 / U+1824	O/U	O-5	Final_0	θ	b_banbvr_baga	U+1873
45/ 45	U+1823 / U+1824	O/U	O-6	Final_1	σ	gedes	U+186D

— Vowels (元音) - Letter OE/UE

- Unification – The vowels OE and UE are unified into a single letter group, as their written forms are same in Traditional Mongolian script. Unicode encoding represents orthographic form rather than phonetic variation, and therefore does not distinguish between different vowel sounds when the glyph structure is the same.

46/ 47	U+1825 / U+1826	OE/UE	U-1	Isolate_0	↑ + σ + ,	titem + gedes + odoi_suul	U+1865 + U+186D + U+1866
48/ 49	U+1825 + FVS1 / U+1826 + FVS1	OE/UE	U-2	Isolate_1	↑ + θ	titem + b_banbvr_baga	U+1865 + U+1873
50/ 51	U+1825 + FVS2 / U+1826 + FVS2	OE/UE	U-3	Isolate_2	θ	b_banbvr_baga	U+1873
52/ 53	U+1825 + FVS3 / U+1826 + FVS3	OE/UE	U-4	Isolate_3	Nil	Not exist in Traditional Mongolian Scrip	Nil
54/ 55	U+1825 + FVS2 + NIRUGU / U+1826 +	OE/UE	U-5	Initial_0	↑ + σ + r	titem + gedes + silbi	U+1865 + U+186D + U+186C

	FVS2 + NIRUGU						
56/ 57	U+1825 + FVS1 + NIRUGU / U+1826 + FVS1 + NIRUGU	OE/UE	U-6	Initial_1	τ	gedes	U+186D
58/ 59	U+1825 / U+1826	OE/UE	U-7	Medial_0	τ	gedes	U+186D
60/ 61	U+1825 / U+1826	OE/UE	U-8	Medial_1	τ + τ	gedes + silbi	U+186D + U+186C
62/ 63	U+1825 / U+1826	OE/UE	U-9	Medial_2	τ + τ + τ	achvg + gedes + silbi	U+186A + U+186D + U+186C
64/ 65	U+1825 / U+1826	OE/UE	U-6	Final_0	θ	b_banbvr_baga	U+1873
66/ 67	U+1825 / U+1826	OE/UE	U-7	Final_1	τ	gedes	U+186D
68/ 69	U+1825 / U+1826	OE/UE	U-8	Final_2	τ + ,	gedes + Odoi_suul	U+186D + U+1866

Vowels EE / Consonant WA — Unified Letter Group (EE/WA)

- Vowel EE (a relatively recent addition, introduced primarily after the 1990s and likely influenced by the Cyrillic letter E (etc.. Европа) through Russian linguistic contact through Mongolia) and the consonant WA are unified into a single letter group. This is because their written forms are identical in Traditional Mongolian script. In accordance with Unicode principles, which encode orthographic form rather than phonetic distinction, no separate characters are assigned when multiple sounds share the same glyph structure.

70/ 71	U+1827 / U+1838	EE / WA	W-1	Isolate	τ	w_gohtai_silbi	U+189F
72/ 73	U+1827 / U+1838	EE / WA	W-2	Initial_0	τ	w_gohtai_silbi	U+189F
74/ 75	U+1827 / U+1838	EE / WA	W-3	Medial_0	τ	w_gohtai_silbi	U+189F
76/ 77	U+1827 / U+1838	EE / WA	W-4	Final_0	τ + ,	w_gohtai_silbi + Odoi_suul	U+189F + U+1866

— Nasals (鼻音)

78	U+1828	N	N-1	Isolate_0	τ	n_titem	U+186F
79	U+1828 + FVS1	N	N-2	Isolate_1	τ	titem	U+1865
80	U+1828	N	N-3	Initial_0	τ	n_titem	U+186F
81	U+1828 + FVS1	N	N-4	Initial_1	τ	titem	U+1865
82	U+1828	N	N-5	Medial_0	τ	n_achvg	U+1871
83	U+1828	N	N-6	Medial_1	τ	achvg	U+186A

						Epenthetic letter/ 垫字 (អេនិធីកុណិត)	
84	U+1828	N	N-7	Final_0	ជ	n_orhich	U+1870
85	U+1828 + FVS2	N	N-8	Final_1	ជ	orhich	U+1867
						Epenthetic letter/ 垫字 (អេនិធីកុណិត)	
86	U+1829	NG	NG-1	Isolate	រ + ឯ	achvg + hinhir	U+186A + U+188A
87	U+1829	NG	NG-2	Initial_0	NIL	NIL	NIL
88	U+1829	NG	NG-3	Medial_0	រ + ឯ	achvg + hinhir	U+186A + U+188A
89	U+1829	NG	NG-4	Final_0	រ + ឃ	silbi + rvsgsilga_or_chachilga	U+186C + U+1868
90	U+1829	NG	NG-5	Final_1	រ + រ + ឃ	achvg + silbi + rvsgsilga_or_chachilga	U+186A + U+186C + U+1868

— Core Consonants (基础辅音)

91	U+182A	B	B-1	Isolate	ឯ	b_banbvr_baga	U+1873
92	U+182A	B	B-2	Initial_0	ឯ	b_banbvr_baga	U+1873
93	U+182A	B	B-3	Initial_1	ឯ	b_banbvr_ih <i>(followed by ធម៌ gedes)</i>	U+1874
94	U+182A	B	B-4	Medial_0	ឯ	b_banbvr_baga	U+1873
95	U+182A	B	B-5	Medial_1	ឯ	b_banbvr_ih <i>(followed by ធម៌ gedes)</i>	U+1874
96	U+182A	B	B-6	Final_0	ឯ	b_banbvr_egsilge	U+1872
97	U+182B	P	P-1	Isolate	ឯ	p_chihi_baga	U+1876
98	U+182B	P	P-2	Initial_0	ឯ	p_chihi_baga	U+1876
99	U+182B	P	P-3	Initial_1	ឯ	p_chihi_ih <i>(followed by ធម៌ gedes)</i>	U+1877
100	U+182B	P	P-4	Medial_0	ឯ	p_chihi_baga	U+1876
101	U+182B	P	P-5	Medial_1	ឯ	p_chihi_ih <i>(followed by ធម៌ gedes)</i>	U+1877
102	U+182B	P	P-6	Final_0	ឯ	p_chihi_egsilge	U+1875
103	U+182C	H	H-1	Isolate	ឬ	h_hanhar	U+1878
104	U+182C	H	H-2	Initial_0	ឬ	h_hanhar	U+1878
105	U+182C	H-HE	H-3	Initial_1	ឯ	Hinhir	U+188A
106	U+182C	H-HU	H-4	Initial_2	ឯ	hunhur <i>(followed by ធម៌ gedes)</i>	U+188B
107	U+182C	HA	H-5	Medial_0	រ + ឯ	achvg + achvg	U+186A + U+186A
108	U+182C	H-HE	H-6	Medial_1	ឯ	hinhir	U+188A
109	U+182C	H-HU	H-7	Medial_2	ឯ	hunhur	U+188B

						(followed by ᠠ gedes)	
110	U+182C	H	H-8	Final_0	ᠥ	agsilga	U+1887
111	U+182D	G	GE-1	Isolate	ᠮ	g_hanhar	U+1888
112	U+182D	G	GE-2	Origin_1	ᠥ + ᠥ	achvg + achvg (ancient usage)	U+186A + U+186A
113	U+182D	G	GE-3	Initial_0	ᠮ	g_hanhar	U+1888
114	U+182D	G-GE	GE-4	Initial_1	ᠴ	hinhir	U+188A
115	U+182D	G-GU	GE-5	Initial_2	ᠴ	hunhur (followed by ᠠ gedes)	U+188B
116	U+182D	G	GE-6	Medial_0	ᠥ + ᠥ	n_achvg + n_achvg	U+1871 + U+1871
117	U+182D	G-GE	GE-7	Medial_1	ᠴ	hinhir	U+188A
118	U+182D	G-GU	GE-8	Medial_2	ᠴ	hunhur (followed by ᠠ gedes)	U+188B
119	U+182D	G	GE-9	Final_0	ᠥ	agsilga	U+1887
120	U+182D	G	GE-10	Final_1	ᠮ	dvsaltai_ag silga	U+1889
121	U+182D	G	GE-11	Final_2	ᠥ + ᡤ	silbi + vrvgsilga_or_chachilga	U+186C + U+1868
122	U+182E	M	M-1	Isolate	ᠮ	m_titem	U+188C
123	U+182E	M	M-2	Initial_0	ᠮ	m_titem	U+188C
124	U+182E	M	M-3	Medial_0	ᠮ	m_achvg	U+188E
125	U+182E	M	M-4	Final_0	ᠮ	m_hurbelge	U+188D
126	U+182F	L	L-1	Isolate	ᠯ	l_titem	U+188F
127	U+182F	L	L-2	Initial_0	ᠯ	l_titem	U+188F
128	U+182F	L	L-3	Medial_0	ᠯ	l_achvg	U+1891
129	U+182F	L	L-4	Final_0	ᠯ	l_hurbelge	U+1890
130	U+1830	S	S-1	Isolate	ᠰ	s_sechig	U+1894
131	U+1830	S	S-2	Initial_0	ᠰ	s_sechig	U+1894
132	U+1830	S	S-3	Medial_0	ሰ	s_sechig	U+1894
133	U+1830	S	S-4	Final_0	ᠰ	s_agshilga	U+1893
134	U+1831	SH	SH-1	Isolate	ᡩ	x_sechig	U+1897
135	U+1831	SH	SH-2	Initial_0	ᡩ	x_sechig	U+1897
136	U+1831	SH	SH-3	Medial_0	ᡩ	x_sechig	U+1897
137	U+1831	SH	SH-4	Final_0	ᡩ	x_agshilga	U+1896
138	U+1832	T	T-1	Isolate	ᡨ	t_senj	U+1898
139	U+1832	T	T-2	Initial_0	ᡨ	t_senj	U+1898
140	U+1832	T	T-3	Medial_0	ᡨ	d_senj (swapped with D in history)	U+1899
141	U+1832	T	T-4	Final_0	ᡨ + ᠤ	t_senj + Odoi_suul	U+1898 + U+1866
142	U+1833	D	D-1	Isolate	ᡩ	d_senj	U+1899
143	U+1833	D	D-2	Initial_0	ᡩ	d_senj	U+1899
144	U+1833	D	D-3	Medial_0	ᡨ	t_senj	U+1898

						(swapped with T in history)	
145	U+1833	D	D-4	Medial_1	𠁚 + 𠁚	gedes + achvg Epenthetic letter/ 垫字 (ᠤᠳᠱᠳ ᠤᠳᠱᠳ)	U+186D + U+186A
146	U+1833	D	D-5	Final_0	𠁚 + 、	d_senj + Odoi_suul	U+1899 + U+1866
147	U+1833	D	D-6	Final_1	𠁚 + ՚	gedes + orhich Epenthetic letter/ 垫字 (ᠤᠳᠱᠳ ᠤᠳᠱᠳ)	U+186D + U+1867
148	U+1834	CH	CH-1	Isolate	𠁚	q_acha	U+189A
149	U+1834	CH	CH-2	Initial_0	𠁚	q_acha	U+189A
150	U+1834	CH	CH-3	Medial_0	𠁚	q_acha	U+189A
151	U+1834	CH	CH-4	Final_0	𠁚 + 、	q_acha + Odoi_suul	U+189A + U+1866
152	U+1835	J	J-1	Isolate	𠁚	j_bosoo_silbi	U+189B
153							
154	U+1835	J	J-2	Initial_0	𠁚	j_bosoo_silbi	U+189B
155							
156	U+1835	J	J-3	Medial_0	𠁚	j_bosoo_silbi	U+189B
157							
158	U+1835	J	J-4	Final_0	𠁚 + 、	j_bosoo_silbi + Odoi_suul	U+189B + U+1866
159							
160	U+1836	Y	Y-1	Isolate	𠁚	y_erteger_silbi	U+189C
161							
162	U+1836	Y	Y-2	Initial_0	𠁚	y_erteger_silbi	U+189C
163	U+1836	Y	Y-3	Medial_0	𠁚	y_erteger_silbi	U+189C
164	U+1836	Y	Y-4	Final_0	𠁚	hvms	U+186B
165	U+1837	R	R-1	Isolate	𠁚	r_erbgeljin_silbi	U+189E
166	U+1837	R	R-2	Initial_0	𠁚	r_erbgeljin_silbi	U+189E
167	U+1837	R	R-3	Medial_0	𠁚	r_erbgeljin_silbi	U+189E
168	U+1837	R	R-4	Final_0	𠁚	r_Odoi_suul	U+189D

— Loan / Extended (借词/扩展)

- Unification of K Letters** - The current Unicode standard encodes two separate characters at U+183A and U+183B, representing two forms of the letter K, commonly interpreted as regional variants (e.g., Mongolia and Inner Mongolia). This separation is not based on orthographic necessity, as both characters represent the same underlying letter and share the same structural role in Traditional Mongolian script. Accordingly, this model unifies the two K characters into a single letter group, with positional and stylistic differences handled at the variant level rather than through separate code points.

169	U+1839	F	F-1	Isolate	𠁚	f_chihi_baga	U+18A1
170	U+1839	F	F-2	Initial_0	𠁚	f_chihi_baga	U+18A1
171	U+1839	F	F-3	Initial_1	𠁚	f_chihi_ih (followed by 𠁚 gedes)	U+18A2

172	U+1839	F	F-4	Medial_0	ɸ	f_chihi_baga	U+18A1
173	U+1839	F	F-5	Medial_1	ɸ	f_chihi_ih (followed by ɂ gedes)	U+18A2
174	U+1839	F	F-6	Final_0	ɸ	f_chihi_egsilge	U+18A0
175/ 176	U+183A/ U+183B	K/KH	K-1	Isolate	ɸ	k_baga	U+18A4
177/ 178	U+183A/ U+183B	K/KH	K-2	Initial_0	ɸ	k_baga	U+18A4
179/ 180	U+183A/ U+183B	K/KH	K-3	Initial_1	ɸ	k_ih (followed by ɂ gedes)	U+18A5
181/ 182	U+183A/ U+183B	K/KH	K-4	Medial_0	ɸ	k_baga	U+18A4
183/ 184	U+183A/ U+183B	K/KH	K-5	Medial_1	ɸ	k_ih (followed by ɂ gedes)	U+18A5
185/ 186	U+183A/ U+183B	K/KH	K-6	Final_0	ɸ	k_egsilge	U+18A3
187	U+183C	TS	TS-1	Isolate	ɂ	c_acha	U+18A6
188	U+183C	TS	TS-2	Initial_0	ɂ	c_acha	U+18A6
189	U+183C	TS	TS-3	Medial_0	ɂ	c_acha	U+18A6
190	U+183C	TS	TS-4	Final_0	ɂ + ,	c_acha + odoi_suul	U+18A6 + U+1866
191	U+183D	Z	Z-1	Isolate	ɂ	z_acha	U+18A7
192	U+183D	Z	Z-2	Initial_0	ɂ	z_acha	U+18A7
193	U+183D	Z	Z-3	Medial_0	ɂ	z_acha	U+18A7
194	U+183D	Z	Z-4	Final_0	ɂ + ,	z_acha + odoi_suul	U+18A7 + U+1866

— Galig (Extensions for Sanskrit and Tibetan 梵文/藏文转写扩展)

195	U+183E	HAA	HAA-1	Isolate	ɿ + ɿ	titem + zhi (galig)	U+1865 + U+18A8
196	U+183F	ZRA	ZRA-1	Isolate	ɿ	galig_tvvrai	U+18AA
197	U+1840	LHA	LHA-1	Isolate	ɿɿ	l_titem + zhi (galig)	U+188F + U+18A8

— Galig (Extensions for Chinese 汉文/中文转写扩展)

198	U+1841	ZHI	ZHI-1	Isolate	ɿ	zhi (galig)	U+18A8
199	U+1842	CHI	CHI-1	Isolate	ɂ	gedes + gedes	U+186D + U+186D

*Other variants within the range of approximately 200–236 are repetitions identified through statistical analysis of the legacy Munk_Gal font dataset.

** This count excludes variants specific to the Todo, Manchu, and Sibo scripts. Support for these scripts can be added modularly without impacting the core architecture.

*** The Z52 component set operates as a collection of atomic, LEGO-like building blocks derived from the historical structure and pedagogical tradition of Mongolian writing. These components may be freely combined without

embedding grammatical logic at the character/glyph level. Language-specific grammar rules, contextual behavior, and specialized usage are instead managed at the Input Method Editor (IME) layer, where appropriate components and variants are selected dynamically.

Sample Z52 AI PyTorch Auto Suggestion IME: <https://github.com/zmongol/ZcodeMachineLearning>

Simple Z52 IME Logic: <https://github.com/zmongol/ZcodeMachineLearning/blob/main/lib/Utils/ZcodeLogic.dart>

Sample Z52 Fuzzy IME Logic: <https://github.com/east-mod/ime>

Sample Z52 Fonts: <https://github.com/zmongol/ZcodeMachineLearning/tree/main/assets/fonts>

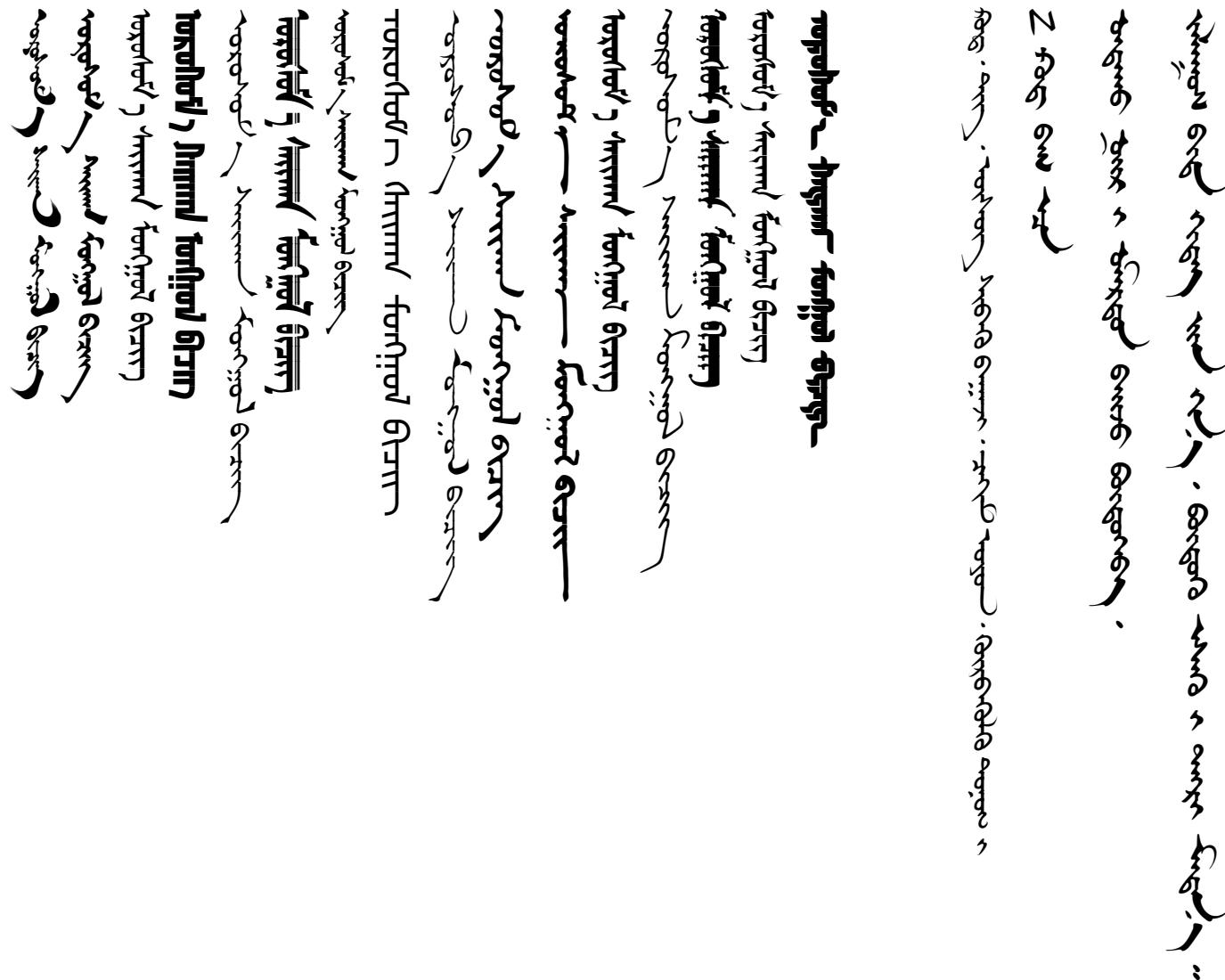
Sample Z52 Flutter App: <https://apps.apple.com/sg/app/zmongol1-0/id1551606659>

Sample Z52 Code converter: <https://mcc.zvvnmod.com/#/translate>

Sample Z52 Apps landing page: <https://install.zcodetech.com/>

Sample Z52 website: <https://zvvnmod.com/#/home>

Appendix 3: Sample Z52 Fonts illustration



Submitted by: Chuck Namujila, Lead Architect, Zcode Team , Jan 02, 2026

Unicode 联盟 – WG2 提案

标题: 关于扩建传统蒙古文并行、基于元素的编码模型 (Z52) 的提案

作者: 朝克那木吉拉 (Chuck Namujila) / Zcode 团队 (Chuck.Namujila@gmail.com)

日期: 2026 年 1 月 2 日

提案类型: 技术 / 编码模型扩展

目标标准: ISO/IEC 10646 / Unicode 标准

1. 提案摘要

本提案请求为传统蒙古文分配一种并行的、基于元素的 **Unicode 编码模型**。该模型旨在与现有的蒙古文区块 (U+1800 范围) 共存，而非取代现有标准。其目标是解决长期以来制约该文字数字化可行性的系统性架构问题，包括对字体层的过度依赖、渲染结果的不确定性，以及数据语义歧义等问题——这些问题已持续存在二十余年。

2. 文字标识

- **文字名称:** 传统蒙古文
- **ISO 15924 代码:** Mong
- **书写方向:** 纵向书写，自上而下，列从左向右排列
- **当前 Unicode 编码范围:**
 - U+1820–U+1842 (抽象语音模型)
 - U+180B–U+180F (格式控制与变体选择符)
- **拟议 Unicode 编码范围:**
 - 一个包含 52 个码位的新区块，优先建议分配在补充多文种平面 (SMP / Plane 1)

3. 背景

传统蒙古文源自粟特文，是一种高度复杂的连写文字体系，其中每个字母都具有独立形、词首形、词中形和词尾形等不同形态。Unicode 于约 2000 年对该文字进行编码时，由于学术界存在分歧，最终采取了一种折衷方案：仅编码“抽象基础字母”，而将具体书写形态的解析交由字体层 (OpenType) 完成。

然而，这些书写变体并非可选的“样式”，而是识字与阅读所必需的正字法要素。

在实际应用中，字体层的形态处理逻辑依赖于用户、字体开发者以及学术群体之间的**隐性或显性共识**，以决定抽象字母应如何解析为具体书写形态。这种共识在不同地区、不同平台及不同世代之间始终难以稳定达成并持续维持，导致渲染行为长期处于碎片化与不一致状态。

4. 问题陈述

当前的“抽象语音”编码架构存在以下五个系统性缺陷：

1. 字体层语义负担

字体被迫承担语言学与语法层面的解释职责，实际上充当了“标准制定者”，而非仅作为视觉呈现工具。

2. 非确定性渲染

相同的 Unicode 序列在不同字体、操作系统和浏览器中可能呈现出不同的显示结果。

3. 数据歧义

存储的文本在脱离特定字体环境后，无法唯一对应其书写语义。

4. 平台依赖性

高质量渲染依赖复杂的 OpenType 特性，而这些特性在移动端和嵌入式设备上的支持并不一致。

5. 不适用于 AI / NLP

编码歧义会降低语料质量，使该文字难以用于现代搜索引擎、机器翻译及大语言模型（LLMs）。

6. 对用户协议的依赖

字体层形态解析的正确性依赖用户与字体开发者在语法解释和书写形态选择上的事先共识。在缺乏统一用户协议的情况下，不同字体会内嵌不同假设，从而导致渲染结果不兼容，并加剧长期碎片化问题。

5. 拟议解决方案：基于元素的模型

本提案提出一种次级编码模型，用于明确表示蒙古文的实际书写形态。该模型不再编码抽象的语音概念，而是采用一种确定性的、基于元素的架构，直接反映文字的真实书写与阅读方式。

从现有模型过渡到 Z52 模型遵循以下三个清晰的架构步骤：

步骤一 —— 基于形态的编码

不再编码 35 个抽象语音字母，而是直接在编码层面体现词首、词中、词尾等正字法现实形态。

步骤二 —— 元素化分解

不再将约 236 种词位变体作为独立码位编码，而是编码 52 个基础结构元素，通过确定性组合构建全部规范化书写变体。

步骤三 —— 以 IME 为中心的组合机制

不再依赖复杂的字体层形态处理逻辑在渲染阶段解析书写形式，而是通过输入法编辑器（IME）在输入阶段预先选择并组合所需形态。这使 Unicode 编码保持稳定，同时大幅降低字体实现的复杂度与负担。

在 Z52 架构下，正确性不再依赖字体或用户层面的事后共识。书写形式在输入阶段即由标准化元素明确构建，确保相同的 Unicode 序列始终对应相同的书写结果。因此，互操作性由设计本身保证，而非通过协商达成。这一转变使“正确性”从一种社会性协商，转变为一种技术性保障。

该基于元素的设计使 Unicode 保持简洁、确定且稳定，同时将语言学与语法层面的智能逻辑迁移至输入层，使其能够被明确、透明地管理。

6. 架构原则

- 正字法现实性

Unicode 应直接编码书写形态，确保存储数据与实际呈现一致。

- 关注点分离

- **Unicode:** 提供稳定、确定性的表示
- **字体:** 仅负责视觉风格与质量，不承担语言逻辑
- **输入法 (IME) :** 负责语法规则、地域习惯及学术分歧的处理

- 数据稳定性

等价的书写形式映射为相同的 Unicode 序列，确保规范化与跨平台一致性。

- 无用户协议依赖的互操作性

Z52 模型消除了对用户或字体层协议的依赖。正字法决策在输入阶段以确定性方式完成，使系统无需共享假设或协商约定即可实现互操作。

Z52 设计将结构复杂性集中到输入阶段的一次性预处理过程中，遵循基于形态、元素组合、以 IME 为中心的架构。通过在输入阶段使用 Z52 类“乐高”元素组合蒙古文字母，该系统显著降低了字体设计、Unicode 编码和文本渲染的下游复杂度，从而减少整体开发与维护成本，并促进更广泛的使用与采用，尤其是在 Z 世代年轻用户群体中。

项目	传统蒙古文现行设计	Z52 设计
输入法	中等	重
字体	重	轻
Unicode	重	轻
输入用户（书写者）	重	轻
输出用户（阅读者）	重	轻
开发	重	轻
维护	重	轻
用户协议	重	无

7. 使命声明：技术性保护

数字层面的脆弱性会导致社会层面的边缘化。如果一种文字无法在智能手机上顺畅使用——难以输入或检索——它就面临被年轻世代逐渐放弃的风险，而这一趋势已经开始显现。Zcode (Z52) 是一项技术性保护工程，旨在确保蒙古文在移动计算与人工智能时代仍具生命力。通过将高摩擦的学术分歧迁移至输入法层，本提案为 Unicode 恢复架构清晰性，并为终端用户恢复可用性。

8. 技术概述 (Z52)

本节为第 5 节所述的三步式元素架构提供技术基础，说明 52 个原子组件如何以完全确定的方式生成全部蒙古文书写形式。

Z52 模型通过对蒙古文字符构造进行历史与结构分析，识别出 52 个原子级组件。

- **当前问题范围**

现有 Unicode 模型编码了 35 个抽象语音字母（U+1820–U+1842），以及 5 个控制字符（FVS1–4 与 MVS，U+180B–U+180F），这些机制在书写系统中引入了外在的、非正字法的控制逻辑。

- **由此产生的扩展**

每个抽象字母最多需要四种词位形态（独立形、词首形、词中形、词尾形），规范化后形成约 236 种不同书写变体。

- **Z52 的解决方式**

这些变体可通过一个封闭的 52 元素集合以确定性方式构建，彻底消除对抽象语音中介或字体层语义解释的依赖。

9. 与现有标准的关系

该模型具有非破坏性。现有蒙古文区块（U+1800）仍然适用于历史与遗留数据。Z52 模型面向需要数据确定性与 AI 兼容性的现代应用场景。

10. 渲染与实现

在 Z52 模型下，渲染过程变得直接而简单，无需复杂的 OpenType GSUB/GPOS 特性即可正确显示蒙古文文本，即便是在基础文本编辑器或旧版浏览器等简化环境中亦可正常工作。

Z52 避免使用以下字体层 OpenType 高级机制：

- 链式上下文替换（calt, ccmp, rlig）
 - 上下文定位与字距调整（kern, GPOS）
 - 单字形替换（GSUB 单替换）
 - 多字形替换（GSUB 多替换）
 - 连字替换（liga, rlig）
-

11. 向后兼容与迁移

迁移为可选行为，由输入法的采用情况驱动。用户可在保持现有文本不变的同时，使用符合 Z52 的输入法创建新内容。

12. 对 Unicode 与 ISO/IEC 10646 的价值

- **恢复架构正确性:** 使蒙古文与其他确定性文字系统保持一致
 - **消除实现冲突:** 将语言解释负担从字体开发者中移除
 - **支持全球检索:** 确保跨平台 100% 的搜索一致性
 - **面向未来:** 提供适用于 AI 与 NLP 的清洁、结构化数据
 - **消除用户协议依赖:** 正确渲染不再依赖用户、字体厂商或平台之间的协议，从而显著降低长期碎片化与实现风险
-

13. 结论

Z52 提案提供了一条技术上保守、向后兼容且架构稳定的发展路径。通过将语言逻辑从编码层解耦并迁移至输入法层，传统蒙古文不再是 Unicode 生态系统中的“技术负担”，而将成为全球数字环境中一个健康、可持续、可运作的文字系统。

Unicode Консорциум – WG2 санал

Гарчиг:

Уламжлалт монгол бичигт зориулсан зэрэгцээ, элементэд суурилсан кодчиллын загвар (Z52)-ын санал

Зохиогч:

Цог Намжил / Zcode баг (Chuck.Namujila@gmail.com)

Огноо:

2026-1-2

Саналын төрөл:

Техникийн / Кодчиллын загварын өргөтгөл

Зорилтот стандарт:

ISO/IEC 10646 / Unicode стандарт

1. Саналын хураангуй

Энэхүү санал нь Уламжлалт Монгол бичигт зориулсан **параллель, элементэд суурилсан Unicode** кодчиллын загварт тусгай блок хуваарилахыг хүсэж байна. Энэ загвар нь одоо байгаа Монгол бичгийн блок (U+1800 муж)-ыг **орлохгүй**, харин зэрэгцэн **оршин** ажиллахаар төлөвлөгдсөн. Зорилго нь 20 гаруй жилийн турш тус бичгийн дижитал хэрэглээний чадамжийг saatuulж ирсэн **фонтод хэт хамаарах байдал, тогтвортгүй (детерминист бус) дурслэл, өгөгдлийн хоёрдмол утга** зэрэг системийн архитектурын хүндрэлүүдийг шийдвэрлэхэд оршино.

2. Бичгийн тодорхойлолт

- **Бичгийн нэр:** Уламжлалт Монгол бичиг
- **ISO 15924 код:** Mong
- **Бичлэгийн чиглэл:** Босоо, дээрээс доош, багана нь зүүнээс баруун тийш
- **Одоогийн Unicode муж:** U+1820 – U+1842 (Хийсвэр авиа-зүйн загвар), U+180B – U+180F
- **Санал болгож буй Unicode муж:** Нэмэлт Олон Хэлний Талбар (SMP / Plane 1)-д байрлуулахыг зорьсон **52 кодын цэгтэй шинэ блок**

3. Суурь мэдээлэл

Уламжлалт Монгол бичиг нь Согд бичгээс үүсэлтэй, **наалданги, курсив** шинжтэй бөгөөд үсэг бүр дангаар, эхэнд, дунд, эцэст гэсэн байрлалын онцгой хэлбэрүүдтэй. 2000 оны орчим Unicode-д кодлох үед эрдэм шинжилгээний байр суурийн зөрүүгээс болж буулт хийж, Unicode нь зөвхөн “**хийсвэр суурь үсэг**”-ийг кодлон, бодит бичгийн хэлбэрүүдийг сонгон гаргах ажлыг **фонтын shaping логик** (OpenType)-д даатгасан. Гэвч эдгээр бичгийн хувилбарууд нь сонголттой “стиль” биш, харин **бичиг үсгийн үндсэн зөв бичих шаардлага** юм.

Практикт, фонтын shaping логик нь хийсвэр үсгийг бодит бичгийн хэлбэрт “яаж” хувиргах талаар **хэрэглэгчид, фонт хөгжүүлэгчид, эрдэмтдийн хамт олон** хоорондын ил буюу далд **зөвшилцөл** дээр тулгуурлаж ирсэн. Харин ийм зөвшилцөл нь ялангуяа бүс нутаг, платформ, үе үеийн хэрэглэгчдийн

хүрээнд тогтвортой хүрч тогтоон барихад бэрх байсаар ирсэн. Үүний улмаас дүрслэл нь өнөөг хүртэл бутархай, зөрүүтэй хэвээр байна.

4. Асуудлын тодорхойлолт

Одоогийн “Хийсвэр авиа-зүйн” архитектур нь дараах таван системийн алдааг үүсгэдэг:

- Фонт дээрх хэлзүйн/утгын ачаалал:** Фонт нь хэл шинжлэлийн ба дүрмийн тайлбар хийж, бараг “стандарт” шиг үүрэг гүйцэтгэхэд хүрдэг (стиль дамжуулагч байх ёстой атал).
- Тогтвортой дүрслэл:** Ижил Unicode дараалал нь өөр өөр фонт, үйлдлийн систем, хөтөч дээр өөрөөр харагддаг.
- Өгөгдлийн хоёрдмол утга:** Текст хадгалагдсан хэлбэрээрээ тухайн фонтыг мэдэхгүйгээр бодит бичгийн утгыг ганцаараа тодорхой илэрхийлж чаддаггүй.
- Платформоос хамааралт:** Өндөр чанартай дүрслэл хийхэд OpenType-ийн ахисан түвшний дэмжлэг шаардлагатай бөгөөд энэ нь мобайл болон суурилагдсан төхөөрөмжүүд дээр жигд хэрэгжээгүй.
- AI/NLP-д тохиромжгүй:** Хоёрдмол кодчилол нь корпусын чанарыг муутгаж, хайлт, машин орчуулга, Том Хэлний Загварууд (LLM) зэрэг орчин үеийн хэрэглээнд нийцгүй болгодог.
- Хэрэглэгчийн зөвшилцөлөөс хамаарах байдал:** Фонтын shaping зөв ажиллах эсэх нь хэрэглэгчид ба фонт хөгжүүлэгчдийн хоорондох дүрмийн тайлбар, бичгийн хэлбэр сонголтын урьдчилсан тохиролцоо-нд тулгуурладаг. Нэгдсэн зөвшилцөл байхгүй үед фонтууд өөр өөр таамаглал кодолж, дүрслэлүүд хоорондоо нийцэхгүй болж, бутрал үргэлжилдэг.

5. Санал болгож буй шийдэл: Элементэд суурилсан загвар

Бид бичигдсэн Монгол хэлбэрийг ил тод төлөөлөх **хоёрдогч кодчиллын загвар** санал болгож байна. Хийсвэр авиа-зүйн ойлголтыг кодлохын оронд, энэ загвар нь бичиг хэрхэн үнэндээ бичигдэж, уншигддаг тэр бодит байдлыг тусгах детерминист, элементэд суурилсан архитектурыг ашиглана.

Одоогийн загвараас Z52 загварт шилжих нь дараах **3 тодорхой архитектурын алхам**-аар хэрэгжинэ:

Алхам 1 — Хэлбэрт суурилсан кодчилол

35 хийсвэр авиа-зүйн үсгийг кодлохын оронд, кодчилол нь бичгийн бодит дүр төрхийг шууд төлөөлөхийн тулд **эхний, дунд, төгсгөлийн** бичгийн хэлбэрийн түвшинд ажиллана.

Алхам 2 — Элементчилэн задлах

236 орчим эх–дунд–эцсийн хувилбарыг тус бүрээр нь бие даасан кодын цэг болгохын оронд, загвар нь **52 суурь бүтцийн элемент**-ийг кодлоно. Эдгээр элементүүдийг детерминист байдлаар нийлүүлж, нормчлогдсон 236 хувилбарыг бүтээнэ.

Алхам 3 — IME төвтэй эвлүүлэлт

Дүрслэх мөчид фontoор маш нарийн shaping логик ажиллуулахын оронд, **IME (оролтын арга)** нь 52 элементээс шаардлагатай хувилбарыг **оролтын үед** урьдчилан сонгож, угсран бүрдүүлнэ. Ингэснээр Unicode кодчилол **тогтвортой** хэвээр үлдэж, харин фонт хэрэгжүүлэлтийн нарийн төвөг ба ачаалал **эрс буурна**.

Z52 архитектурын хүрээнд зөв ажиллах байдал нь фontoор эсвэл хэрэглэгчдийн “дараа нь” тохиролцох асуудал байхaa болино. Бичгийн хэлбэрүүд нь стандартчилсан элементүүдээр **оролтын цэг дээр тодорхой** угсрагдаж, ижил Unicode дараалал нь үргэлж ижил бичгийн үр дүнг илэрхийлнэ. Тиймээс нийцтэй байдал нь зөвшилцлөөр бус, **загварын өөрийн баталгаа**-гаар хангагдана. Энэ өөрчлөлт нь “зөв байх” асуудлыг нийгмийн хэлэлцээрээс **техникийн баталгаа** болгон хувиргана.

Энэхүү элементтэд суурилсан загвар нь Unicode-ийг энгийн, детерминист, тогтвортой байлгаж, харин хэлзүй-дүрмийн “ухаан”-ыг ил тод удирдаж болох **оролтын давхарга (IME)** руу шилжүүлнэ.

6. Архитектурын зарчмууд

- **Бичгийн бодит байдал:** Unicode нь бичиг хэрхэн харагддагаар нь шууд кодлох ёстой бөгөөд ингэснээр хадгалагдсан өгөгдөл нь бичгийн бодит дүрслэлтэйгээ таарна.
- **Үргийн салгалт:**
 - **Unicode:** Тогтвортой, детерминист төлөөлөл өгнө.
 - **Фонт:** Хэлзүйн логик хийхгүй, зөвхөн харагдах чанар/стилийн танилцуулга хийнэ.
 - **IME:** Дүрэм, бус нутгийн ялгаа, эрдэм шинжилгээний маргааныг оролтын түвшинд зохицуулж, “ухаан”-ыг өөртөө төвлөрүүлнэ.
- **Өгөгдлийн тогтвортой байдал:** Ижил бичгийн хэлбэрүүд ижил Unicode дараалалд хөрвөх тул нормчлол ба платформ хоорондын тогтвортой байдал хангагдана.
- **Зөвшилцөлгүй нийцтэй байдал:** Z52 загвар нь зөв дүрслэлд хэрэглэгч эсвэл фонтын зөвшилцэл шаардахгүй. Бичгийн шийдвэрүүд оролтын үед детерминист байдлаар шийдэгдэж, системүүд харилцан нийцэхийн тулд “адил таамаглал” эсвэл “хэлэлцэж тохирох” шаардлагагүй болно.

Z52 загвар нь бүтцийн төвөгшлийг оролтын үеийн нэг удаагийн урьдчилсан боловсруулалт руу шилжүүлж, хэлбэрт суурилсан, элементтээр эвлүүлдэг, IME төвтэй архитектурыг хэрэгжүүлнэ. Оролтын үед Z52 “lego” мэт элементүүдээр үсгийг угсрах замаар фонтын дизайн, Unicode кодчилол, дүрслэлд тавигдах дарамт буурна. Ингэснээр Уламжлалт Монгол бичгийг хөгжүүлэх, арчлах, хэрэглэх нийт зардал ба техникийн ачаалал буурч, ялангуяа Z үеийн zaluu хэрэглэгчдийн оролцоо, нэвтрэлтийг дэмжинэ.

	Одоогийн загвар	Z52 загвар
Оруулах арга (IME)	Дунд	Хүнд
Фонт	Хүнд	Хөнгөн
Unicode	Хүнд	Хөнгөн
Бичигч (input user)	Хүнд	Хөнгөн
Уншигч (output user)	Хүнд	Хөнгөн
Хөгжүүлэлт	Хүнд	Хөнгөн
Арчилгаа	Хүнд	Хөнгөн
Хэрэглэгчийн зөвшилцэл	Хүнд	Тэг (шаардлагагүй)

7. Эрхэм зорилго: Техникийн хадгалалт

Дижитал хэврэг байдал нь нийгмийн хоцрогдол руу хөтөлнө. Хэрэв нэг бичиг ухаалаг утсан дээр жигд ажиллахгүй, оруулах эсвэл хайх ажиллагаа төвөгтэй бол тэр бичиг залуу үеийн дунд орхигдох эрсдэлтэй—энэ нь аль хэдийн ажиглагдаж байна. Zcode (Z52) нь Монгол бичгийг мобайл тооцоолол, Хиймэл Оюун, өгөгдлийн эрин үед **хамааралтай хэвээр** байлгах техникийн хадгалалтын санаачилга юм. Эрдэм шинжилгээний зөрчилтэй, үрэлт ихтэй асуудлуудыг IME давхаргад шилжүүлснээр бид Unicode-ийн архитектурын тодорхой байдлыг сэргээж, эцсийн хэрэглэгчдийн ашиглахад хялбар байдлыг сайжруулна.

8. Техникийн тойм (Z52)

Энэ хэсэг нь 5-р хэсэгт тайлбарласан элементэд суурилсан 3 алхмын архитектурын техникийн үндэслэлийг өгч, 52 атом элемент нь бүх бичигдсэн Монгол хэлбэрүүдийг хэрхэн бүрэн, детерминист байдлаар үүсгэдгийг харуулна.

Z52 загвар нь Монгол үсгийн бүтцийг түүхэн ба бүтцийн талаас шинжилж, **52 атом бүрэлдэхүүн**-ийг тогтооно.

- Одоогийн асуудалтай хүрээ:** Одоогийн Unicode загвар нь 35 хийсвэр авиа-зүйн үсэг (U+1820–U+1842) болон 5 удирдлагын тэмдэг—FVS1–4, MVS (U+180B–U+180F)-ийг кодолдог бөгөөд энэ нь бичгийн тогтолцоонд “бичиг-зүйн бус” гадаад удирдлагын механизмыг оруулж ирдэг.
- Үүсэх тэлэлт:** Хийсвэр үсэг бүр дангаар/эхэнд/дунд/эцэст гэсэн 4 байрлалын хэлбэр шаардаж, нормчлолын дараа ойролцоогоор **236 ялгаатай бичгийн хувилбар** үүснэ.
- Z52 шийдэл:** Эдгээр 236 хувилбар нь **52 атом элементтэй хаалттай багцаас** детерминист байдлаар угсралдах боломжтой бөгөөд хийсвэр авиа-зүйн дам тайлбар эсвэл фонтын түвшний утга зүйн тайлбар шаардлагагүй болно.

9. Одоогийн стандарттай харилцаа

Энэхүү загвар нь **тасалдуулахгүй**. Одоогийн Монгол бичгийн блок (U+1800) нь хуучин өгөгдөл хүчинтэй хэвээр байна. Z52 загвар нь өгөгдлийн бүрэн бүтэн байдал, AI нийцтэй байдал шаардсан орчин үеийн хэрэглээнд зориулагдана.

10. Дүрслэл ба хэрэгжилт

Дүрслэл нь илүү **шулуун, энгийн** болно. Энэ нь OpenType-ийн GSUB/GPOS зэрэг нарийн төвөгтэй боломжуудын хэрэгцээг бууруулж, Монгол бичиг нь энгийн текст засварлагч, хуучин хөтөч зэрэг “энгийн орчин”-д ч зөв харагдах нөхцөлийг бүрдүүлнэ.

Z52 нь фонтын түвшний дараах ахисан механизмуудээс зайлсхийсэн:

- Холбоост нөхцөлт орлуулалт (calt, ccmp, rlig)
- Кернинг ба нөхцөлт байршуулалт (kern, GPOS)
- Нэг тэмдэгт орлуулалт (GSUB Single Substitution)
- Олон тэмдэгт орлуулалт (GSUB Multiple Substitution)
- Лигатурын орлуулалт (liga, rlig)

11. Буцах нийцтэй байдал ба шилжилт

Шилжилт нь **заавал биш** бөгөөд IME-ийн нэвтрэлтээр удирдагдана. Хэрэглэгчид шинэ өгөгдөл үүсгэхдээ Z52-д нийцсэн IME ашиглах боломжтой бөгөөд хуучин текстээ өмнөх хэвээр нь хадгалж үлдэнэ.

12. Unicode ба ISO/IEC 10646-д өгөх ашиг тус

- Архитектурын зөв байдлыг сэргээх:** Монгол бичгийг бусад детерминист бичгүүдтэй ижил зарчимд ойртуулна.
- Зөрчлийг арилгах:** Хэлзүйн тайлбарын ачааллыг font хөгжүүлэгчдээс салгана.
- Дэлхийн хайлтыг боломжтой болгох:** Бүх платформ дээр 100% хайлтын нарийвчлалд хүргэнэ.

- **Ирээдүйд бэлтгэх:** AI/NLP-д хэрэгтэй цэвэр, бүтэцтэй өгөгдлийг хангана.
- **Зөвшилцлөөс хамаарах хамаарлыг арилгах:** Зөв дүрслэл нь хэрэглэгч, фонт нийлүүлэгч, платформуудын хооронд байнга зөвшилцэл байгуулах шаардлагагүй болж, урт хугацааны бутрал ба хэрэгжилтийн эрсдэлийг мэдэгдэхүйц бууруулна.

13. Дүгнэлт

Z52 санал нь техникийн хувьд болгоомжтой, буцах нийцтэй, архитектурын хувьд тогтвортой шийдэл юм. Хэлзүйн логикийг кодчиллын давхаргаас салган IME-д төвлөрүүлснээр Уламжлалт Монгол бичиг Unicode экосистемийн “толгойны өвчин” биш, харин дэлхийн дижитал орчинд **идэвхтэй, ажилладаг** бичиг болж үлдэх нөхцөл бүрдэнэ.