

**The SignPuddle Standard for SignWriting Text  
draft-slevinski-signwriting-text-01**

**Abstract**

For concreteness, because the universal character set is not yet universal, and because an international standard for the internet community should be documented and stable, this I-D has been released with the intention of producing an RFC to document the character use and naming conventions of the SignWriting community on the Internet.

The SignWriting Script is an international standard for writing sign languages by hand or with computers. From education to research, from entertainment to religion, SignWriting has proven useful because people are using it to write signed languages. The SignWriting Script has two major families: Block Printing for the reader and Handwriting for the writer. The script encoding model presented in this document evolved from the Block Printing half of the SignWriting Script.

The SignWriting Text encoding model encompasses the Block Printing family of the SignWriting Script. The plain text model for the mathematical names has been stable since January 12th, 2012. The visual image can be SVG generated on the server or created with an experimental TrueType Font. The coded character sets and character encoding forms are documented with regular expressions.

The ad hoc graphemes of informal SignWriting were first created in 1974. Ad hoc graphemes are still used in the handwriting family. The standardized symbols of computerized Block Printing text began in 1986. After several generations of writers and standardized symbolsets, the ISWA 2010 has been optimized and refined as a 16-bit coded character set with several isomorphic encodings based on an ordered hierarchy with 6 degrees of significance. The International SignWriting Alphabet 2010 is a mathematical symbolset that has been stable since its initial release on May 11th, 2010.

The SignPuddle Standard for SignWriting Text is an open and freely available encoding model for sign language as text. The licenses include the Open Font License for the fonts, Creative Commons by-sa (Attribution, Share Alike) for the documentation, and the GPL for the software implementation. The technological infrastructure continues

to expand and should be fully realized by the time this I-D has become an RFC. SignPuddle Online contains almost 1 million examples of 2-dimensional signs written by the internet community. Each logogram has a mathematical name which describes the freeform placement of the symbols. These strings are the written record of the sign. This standard and emerging infrastructure are used for the sign language Wikipedia project on Wikimedia Labs. This standard is being integrated with the SignTyp linguistic coding system developed by Rachel Channon through an NSF grant. This standard was the origin for the alternate Unicode proposals.

For Unicode, the current use of the Private Use Area font characters is documented. The state of the TrueType Font is explained. A character proposal for plane 1 is included that is isomorphic with the characters that are currently used by the community.

Three appendices discuss additional topics to the standard. The first discusses the Modern SignWriting theory and example document, stable since January 12, 2012. The second discusses the founding principles of Cartesian SignWriting: a script encoding model for SignWriting Block Printing. The third discusses a common framework for written sign language grammar.

This memo concretely defines a conceptual character encoding map for the Internet community. It is published for reference, examination, implementation, and evaluation. Distribution of this memo is unlimited.

#### Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 10, 2013.

#### Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the



document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">6</a>
<a href="#">1.1.</a>	<a href="#">Overview</a>	<a href="#">6</a>
<a href="#">1.2.</a>	<a href="#">Historical Foundation</a>	<a href="#">7</a>
<a href="#">1.3.</a>	<a href="#">Current Usage</a>	<a href="#">9</a>
<a href="#">2.</a>	<a href="#">SignWriting Script</a>	<a href="#">9</a>
<a href="#">2.1.</a>	<a href="#">2-Dimensional Logograms</a>	<a href="#">10</a>
<a href="#">2.2.</a>	<a href="#">Viewpoints, Planes, &amp; Perspectives</a>	<a href="#">10</a>
<a href="#">2.3.</a>	<a href="#">Block Printing</a>	<a href="#">11</a>
<a href="#">2.3.1.</a>	<a href="#">Education</a>	<a href="#">11</a>
<a href="#">2.3.2.</a>	<a href="#">Publishing</a>	<a href="#">11</a>
<a href="#">2.3.3.</a>	<a href="#">Computerized</a>	<a href="#">12</a>
<a href="#">2.4.</a>	<a href="#">Handwriting</a>	<a href="#">12</a>
<a href="#">2.4.1.</a>	<a href="#">Cursive</a>	<a href="#">12</a>
<a href="#">2.4.2.</a>	<a href="#">Shorthand</a>	<a href="#">12</a>
<a href="#">3.</a>	<a href="#">SignWriting Text</a>	<a href="#">13</a>
<a href="#">3.1.</a>	<a href="#">Mathematical Name</a>	<a href="#">13</a>
<a href="#">3.1.1.</a>	<a href="#">Pattern String</a>	<a href="#">13</a>
<a href="#">3.1.2.</a>	<a href="#">Unordered String</a>	<a href="#">15</a>
<a href="#">3.1.3.</a>	<a href="#">Compact and Tractable</a>	<a href="#">15</a>
<a href="#">3.2.</a>	<a href="#">Visual Image</a>	<a href="#">16</a>
<a href="#">3.2.1.</a>	<a href="#">TrueType Font</a>	<a href="#">16</a>
<a href="#">3.2.2.</a>	<a href="#">Server Generated SVG</a>	<a href="#">16</a>
<a href="#">3.3.</a>	<a href="#">Character Encoding Scheme</a>	<a href="#">16</a>
<a href="#">3.3.1.</a>	<a href="#">ASCII</a>	<a href="#">16</a>
<a href="#">3.3.2.</a>	<a href="#">Unicode</a>	<a href="#">17</a>
<a href="#">3.4.</a>	<a href="#">Coded Character Set</a>	<a href="#">17</a>
<a href="#">3.4.1.</a>	<a href="#">x-ISWA-2010</a>	<a href="#">17</a>
<a href="#">3.4.2.</a>	<a href="#">x-Binary-SignWriting</a>	<a href="#">18</a>
<a href="#">3.4.3.</a>	<a href="#">x-Character-SignWriting</a>	<a href="#">18</a>
<a href="#">3.5.</a>	<a href="#">Character Encoding Form</a>	<a href="#">19</a>
<a href="#">3.5.1.</a>	<a href="#">Lite Markup</a>	<a href="#">19</a>
<a href="#">3.5.2.</a>	<a href="#">Formal SignWriting</a>	<a href="#">21</a>
<a href="#">3.5.3.</a>	<a href="#">Kartesian SignWriting</a>	<a href="#">22</a>



3.5.3.1.	Raw . . . . .	22
3.5.3.2.	Expanded . . . . .	23
3.5.3.3.	Layout . . . . .	24
3.5.3.4.	Panel . . . . .	25
3.6.	Query Language . . . . .	26
4.	ISWA 2010 . . . . .	26
4.1.	Grapheme . . . . .	27
4.2.	Symbol . . . . .	28
4.3.	Hierarchy . . . . .	30
4.4.	Combined Character Sequence . . . . .	34
4.5.	Validity . . . . .	36
5.	SignPuddle Standard . . . . .	39
5.1.	Licenses . . . . .	39
5.1.1.	Open Font License . . . . .	39
5.1.2.	Creative Commons . . . . .	39
5.1.3.	GPL . . . . .	39
5.1.4.	BSD . . . . .	39
5.2.	Infrastructure . . . . .	39
5.2.1.	International SignWriting Alphabet Fonts . . . . .	39
5.2.2.	SignWriting Icon Server . . . . .	40
5.2.3.	SignWriting Thin Viewer . . . . .	41
5.2.3.1.	CSS Text Layout . . . . .	41
5.3.	Compatibility . . . . .	42
5.3.1.	SignPuddle Online . . . . .	42
5.3.2.	Wikimedia Labs . . . . .	42
5.3.3.	SignTyp . . . . .	43
5.3.4.	SignWriter Studio . . . . .	43
5.3.5.	DELEGS Online . . . . .	44
6.	Unicode Integration . . . . .	44
6.1.	Private Use Area Font Characters . . . . .	44
6.2.	Proposal . . . . .	45
7.	IANA Considerations . . . . .	45
8.	Security Considerations . . . . .	46
Appendix A.	Modern SignWriting . . . . .	47
Appendix B.	Cartesian SignWriting . . . . .	47
B.1.	Signbox . . . . .	48
B.2.	Temporal Order . . . . .	48
B.3.	Logograph of Logographs . . . . .	49
Appendix C.	Theory of SignWriting Grammar and Encoding . . . . .	50
C.1.	Logographic Sign . . . . .	50
C.2.	Punctuation and Text . . . . .	50
C.3.	Terms . . . . .	51
C.4.	Lanes . . . . .	51
C.5.	Modes . . . . .	51
C.6.	Layout . . . . .	53
C.6.1.	Freeform . . . . .	53
C.6.2.	Restricted . . . . .	54
C.6.3.	Non-form . . . . .	54



<a href="#">C.7.</a>	Positioning . . . . .	<a href="#">54</a>
<a href="#">C.7.1.</a>	Absolute . . . . .	<a href="#">55</a>
<a href="#">C.7.2.</a>	Relative . . . . .	<a href="#">55</a>
Author's Address . . . . .		<a href="#">55</a>



## **1. Introduction**

For concreteness, because the universal character set is not yet universal, and because an international standard for the internet community should be documented and stable, this I-D has been released with the intention of producing an RFC to document the character use and naming conventions of the SignWriting community on the Internet.

The SignWriting Script is an international standard for writing sign languages by hand or with computers. From education to research, from entertainment to religion, SignWriting has proven useful because people are using it to write signed languages.

Sign languages are fundamentally different than spoken language in the quality of the segments in the stream of human speech. The SignWriting Script uses 2-dimensional logograms with freeform symbol placement to capture the spatial and simultaneous segments in the stream of signed language speech.

The SignWriting fonts and standards are freely and openly available, with no royalties or restrictions. This information is provided to promote a complete solution for an open culture in written sign language.

### **1.1. Overview**

The SignPuddle Standard for SignWriting Text is an emerging standard intended for the internet community. This memo concretely defines a fully developed model for reference, examination, implementation, and evaluation. Distribution of this memo is unlimited.

The fonts are officially available [[1](#)]. The release candidate of the SignWriting Icon Server is available on Github [[2](#)], hosted on SignBank [[3](#)] and hosted on Wikimedia Labs [[4](#)].

[Section 1](#) Introduction: includes a discussion of terminology, historical background, current usage, and this overview of the document.

[Section 2](#) SignWriting Script: includes a general discussion of the SignWriting script. Both the Block Printing and the Handwriting families are discussed.

[Section 3](#) SignWriting Text: includes a general discussion of the plain text of logograms for the mathematical names and visual images.

[Section 4](#) ISWA 2010: discusses the SignWriting grapheme, symbolset, and symbol encoding of the ISWA 2010. Symbols are visually iconic,



uniquely identified, and organized in a layered hierarchy.

[Section 5](#) SignPuddle Standard: defines the licenses, infrastructure, and the data available.

[Section 6](#) Unicode Integration: discusses the private use area font characters and the proposed characters on plane 1.

[Appendix A](#) Modern SignWriting: discusses the theory and example document released on January 12th, 2012.

[Appendix B](#) Cartesian SignWriting: presents a script encoding model for SignWriting Block Printing. Formal structures for logographic sign are mixed with punctuation to form text.

[Appendix C](#) Theory of SignWriting Grammar: discusses the common and possible script encoding models for written sign language.

## **[1.2.](#) Historical Foundation**

In 1966, Valerie Sutton invented the DanceWriting notation, which was the precursor to the entire Sutton MovementWriting System.

in 1974, Valerie Sutton invented the SignWriting Script. The subsequent development of the script was driven by input from readers and writers, both hearing and Deaf.

From 1974 to 1986 SignWriting Script was written exclusively by hand. During this time the use of the script spread around the world, and to this day it continues to be written on paper and chalkboard.

In 1981, the development of SignWriting Block Printing evolved rapidly with the publication of the SignWriting Newsletter, which was published from 1981 to 1984.

In 1984 Emerson and Stern Associates received a grant to develop a word processor for SignWriting Block Printing. The resulting software, which operated on the Apple II, supported only a minor subset of the SignWriting system. It was not subsequently used, and received no further development.

In 1986, Richard Gleaves designed and developed SignWriter as a word processor for SignWriting Block Printing. SignWriter introduced the keyboard typing model and a symbol encoding system which served as the basis for subsequent encoding systems. The initial version was for the Apple IIe, and the resulting symbolset was limited by the 128KB memory limit.



By 1995, SignWriter had been ported to MS-DOS and expanded to support multiple languages, an integrated sign dictionary, and the full SSS-95 symbolset. SignWriter DOS was distributed on the internet, and achieved widespread international use.

In 1999, the SSS-99 symbolset was created for SignWriter Java. The revamped symbolset was created without the limitations imposed upon the SSS-95.

In 2002, the SSS-2002 symbolset reorganized the structure of the symbols imposing a multi level hierarchy with the modern symbol ID. The SSS-2002 was the first symbolset used in the SignBank 2002 application by Todd Duell.

In 2004, the SSS-2004 symbolset was created after reaching widespread international use. The SSS-2004 was the first symbolset used in the SignPuddle application by Steve Slevinski. This symbolset was expanded to include international MovementWriting concepts and became known as the International MovementWriting Alphabet.

September 12, 2008, Valerie Sutton and Steve Slevinski released the ISWA 2008 under the open font license. The International SignWriting Alphabet 2008 was a major refactoring of the IMWA concept by eliminating the general MovementWriting symbols and focusing on the SignWriting script. Valerie organized and named 37,811 unique symbols. Steve analyzed and formatted the ISWA 2008, creating a 16-bit coded character set called the x-ISWA-2008. Steve also created the first iteration of Cartesian SignWriting as a script encoding model.

The ISWA 2008 was used in a production setting for a year and a half without issue. In 2010, the ISWA 2008 was updated. 576 unused symbols had a palm facing irregularity which needed to be fixed. General size and shape of the symbols did not change.

May 11th, 2010, Valerie and Steve released the ISWA 2010. The ISWA 2010 was designed as a focused refactor of the ISWA 2008 concepts. The update included a restructured hierarchy, better movement symbols, elimination of variation defects, addition of new hand shapes, and removal of hand shape variations. Revision 2 of Cartesian SignWriting script encoding model was released for the ISWA 2010. The symbolset and encoding have been stable since release, with only a cosmetic fix for symbol 01-06-017-01-03-10.

June 22nd, 2010, Steve refactored the coded character set as 12-bit rather than 16-bit to improve searching. The updated script encoding model was called Cartesian SignWriting revision 3.



October 20th, 2010, the initial release of the ISWA 2010 Font Reference. Since then, 2 years of stability and growth.

February 23rd, 2011, the addition of SVG using polygon line tracing.

September 19th, 2011, the complete SVG Refinement by Adam Frost.

January 12th, 2012, the fully realized character encoding model for SignWriting Text.

May 2nd, 2012, added database fonts.

November 1st, 2012, the prerelease of the SignWriting Icon Server.

### **1.3. Current Usage**

SignPuddle Online contains almost 1 million examples of 2-dimensional signs written by the internet community. Each logogram has a mathematical name that describes the freeform placement of the symbols. These strings are the written record of the sign. XML files organize these names by language and purpose. The ASL Dictionary has over 9 thousand entries.

This standard and emerging infrastructure are used for the sign language Wikipedia project on Wikimedia Labs ([Section 5.3.2](#)). This standard is being integrated with the SignTyp linguistic coding system developed by Rachel Channon through an NSF grant ([Section 5.3.3](#)). This standard was the origin for the alternate Unicode proposals. Compatibility with this standard is highly encouraged to efficiently leverage sign language as text.

For Unicode, the current use of the Private Use Area font characters is documented. A character proposal for plane 1 is included that is isomorphic with the characters that are currently used by the community.

## **2. SignWriting Script**

The SignWriting Script is the universal and complete solution for written sign language. It has been applied to a wide and deep international community of many sign languages including: American Sign Language, Arabian Sign Languages, Australian Sign Language, Bolivian Sign Language, Brazilian Sign Language, British Sign Language, Catalan Sign Language, Colombian Sign Language, Czech Sign Language, Danish Sign Language, Dutch Sign Language, Ethiopian Sign Language, Finnish Sign Language, Flemish Sign Language, French-Belgian Sign Language, French Sign Language, German Sign Language,





Greek Sign Language, Irish Sign Language, Italian Sign Language, Japanese Sign Language, Malawi Sign Language, Malaysian Sign Language, Maltese Sign Language, Mexican Sign Language, Nepalese Sign Language, New Zealand Sign Language, Nicaraguan Sign Language, Norwegian Sign Language, Peruvian Sign Language, Philippines Sign Language, Polish Sign Language, Portuguese Sign Language, Quebec Sign Language, South African Sign Language, Spanish Sign Language, Swedish Sign Language, Swiss Sign Language, Taiwanese Sign Language, and Tunisian Sign Language.

Initially developed in 1974, the script was written exclusively by hand for 12 years. Since then the script has spread around the world and continues to be written on paper and chalkboard.

In 1981, SignWriting Publishing rapidly evolved with Block Printing. In 1986, computerization of the SignWriting Block Printing began. The current symbol encoding of the ISWA 2010 has been stable since the font release on October 20th, 2010. The current character encoding model has been stable since the initial release of Modern SignWriting on January 12th, 2012.

### **2.1. 2-Dimensional Logograms**

A founding principle of the SignWriting Script is that signs are written in 2-dimensional signboxes. The size of the signbox varies with the symbols written inside. Both block printing and handwriting use 2-dimensional logograms.

Inside of a 2-dimensional signbox, the symbols are placed in a freeform, 2-dimensional arrangement. This feature of the script expresses spatial relation directly.

### **2.2. Viewpoints, Planes, & Perspectives**

Writing based on vision uses two viewpoints: receptive and expressive. The receptive viewpoint is based on the idea of receiving an image. For the receptive viewpoint, the right hand of a signer will be written on the left side of the canvas. When SignWriting is used for transcription, the receptive view is most often used. The related writing systems of DanceWriting and MovementWriting normally use the receptive viewpoint.

The expressive viewpoint is based on the idea of expressing a concept. For the expressive viewpoint, the right hand of a signer will be written on the right side of the canvas. When SignWriting is used for authorship, the expressive view is most often used.

There are two main writing planes: the front wall (Frontal Plane) and



the floor (Transverse Plane). The choice of writing plane can affect the shape of the graphemes, such as the fill pattern for the hand graphemes or the tail for the movement arrow graphemes.

There are two perspectives: front and top. The front perspective is a straight on view of/from the signer. The top perspective is a top-down view of the signer. Usually, a cluster will be written from a single perspective.

### **2.3. Block Printing**

Block printing is only half of the SignWriting Script. Block printing is based on the iconic symbols of the symbol set. Each of the iconic symbols is structured, standardized, and highly featural. Block printing is used in education, publishing, and is the basis of the computerized model.

Valerie Sutton writes, "SignWriting Printing is easy to read. It is designed for the reader. The Printing can be written by hand as well as by computer. If I am writing a letter to a friend in ASL, I write the letter in SignWriting Printing, taking the time to make sure that my handwritten-symbols are easy and clear to read. I try to write as clearly as if I were using a computer. Of course it is slower, but it is worth it, knowing that my friend will be able to read my letter!"

#### **2.3.1. Education**

Kids all over the earth [\[5\]](#) are learning block printing thanks to Valerie Sutton and the material she donates through the Center for Sutton Movement Writing [\[6\]](#).

#### **2.3.2. Publishing**

The history of SignWriting Publishing had a rapid development between 1981 and 1984 with the SignWriter Newspaper [\[7\]](#). Patience and concentration was needed to write neat enough for publication. Stencils and wax transfer symbols were used in painstaking work. Typesetters could consistently reproduce the iconic symbols.

Discussions during early publishing history were a catalyst for developing a way to type sign language.

The SignWriter Newspaper suspended in 1984 and resumed publication as a typed SignWriter Newsletter in 1989.



### **[2.3.3.](#) Computerized**

Block printing is the basis of the computerized SignWriting model.

Read about the Historical Foundation in [section 2.C](#) of Modern SignWriting.

Computerized SignWriting is important, but there is so much more [\[8\]](#) to the SignWriting Script [\[9\]](#).

## **[2.4.](#) Handwriting**

SignWriting Handwriting [\[10\]](#) has always been a part of the script.

Valerie Sutton writes, "SignWriting Handwriting is easier to write by hand, than the Printing. It is designed for the writer. There are several variations of Handwriting, and since most of the time, the writer is only writing for private notes, some writers create their own shortcuts that work just for them...and that is fine!"

### **[2.4.1.](#) Cursive**

A popular form of SignWriting is cursive. It can be shared among a groups of writers or it can be individualized and personal. Cursive writing is designed to have fluid marks and a natural flow. Cursive writing may use fewer features than the iconic symbols, but should be related to an iconic symbol in appearance and meaning. Once developed, this style of writing is great for taking notes in a class.

### **[2.4.2.](#) Shorthand**

Shorthand is a skill of the proficient writer [\[11\]](#). They can write SignWriting shorthand quickly and naturally.

In 1982, Sign Language Stenographers could record sign language with SignWriting Shorthand at normal signing speed [\[12\]](#). Time tests proved practice and special training were required. The marks they write are personal style of quick and efficient strokes with a highly developed reception to what signifies meaning. They understand the iconic symbols of the SignWriting Script, but their marks are personal reminders rather than a fully developed text.

The shorthand in and of itself is often an incomplete representation of the gestures that were experienced. The shorthand writing can be thought of as a short-term memory device. Often shorthand notes must be revised and extended at a later time, the sooner the better.



### **3. SignWriting Text**

SignWriting Text uses plain text that is iconic. The sequential characters specify properties in common between forms. The text is diagrammatic with defined relationships and simple structures. It clarifies likenesses that are topologically similar.

SignWriting Text is grammatically correct because it supports 2-dimensional arrangement and writing with lanes. Mathematically sized logograms are named with plain text strings based on patterns. Simple HTML and CSS are used for proper vertical layout.

This model separates visual display from layout issues. It is compatible with TrueType Fonts and server generated SVG.

The model defines several compatible coded character sets and character encoding forms.

#### **3.1. Mathematical Name**

The mathematical name of a logographic sign is a plain text string of characters. This encoding model makes explicit those features which can be effectively and efficiently processed. Formal languages and regular expressions are used to solve fundamental problems.

##### **3.1.1. Pattern String**

The mathematical name is structured with 11 different tokens. They can be grouped in 4 layers: the 5 structural makers (A, B, L, M, R), the 3 base symbol ranges (w, s, P), the 2 modifier indexes (i, o), and the numbers (n).





## Token Patterns

Pattern	Description
wio	a writing symbol as 3 tokens of writing base, fill modifier and rotation modifier
nn	coordinate with X and Y values as 2 numbers
wionn	a spatial symbol as 5 tokens, with 3 tokens for a writing symbol and 2 tokens for coordinates of top left placement
(wionn)*	zero or more spatial symbols
Bnn(wionn)*	a signbox with a preprocessed maximum coordinate and a list of spatial symbols used for horizontal writing
[LMR]	a lane marker: either left, middle or right.
[LMR]nn(wionn)*	a signbox in either the left, middle, or right lane with a preprocessed maximum coordinate and a list of spatial symbols used for vertical writing
[ws]	a writing base symbol or a detailed location base symbol
[ws]io	a writing symbol or a detailed location symbol
([ws]io)+	one or more writing symbols and/or detailed location symbols



(A([ws]io)+)?	an optional prefix as a	
	prefix marker followed by	
	one or more writing	
	symbols and/or detailed	
	location symbols	
+-----+		
Pio	a punctuation symbol as a	
	punctuation base symbol	
	with a fill modifier and	
	a rotation modifier	
+-----+		
(((A([ws]io)+)?Bnn(wionn)*) Pio)+	a sign text for	
	horizontal writing as a	
	string of signboxes (with	
	optional prefixes) and	
	punctuation	
+-----+		
(((A([ws]io)+)?[LMR]nn(wionn)*) Pio)+	a sign text for vertical	
	writing as a string of	
	signboxes in lanes (with	
	optional prefixes) and	
	punctuation	
+-----+		

Table 1

### [3.1.2.](#) Unordered String

2-dimensional space does not have a normative 1-dimensional order. A group of spatial symbols is defined as (wionn)\* which is zero or more writing symbols with 2-dimensional placement by tokens nn for each symbol. The tokens nn are meaningful and searchable. Each symbol defined with wionn is absolutely meaningful and searchable. Except for exact sign matching, the 2-dimensional order of the spatial symbols is meaningless and unreliable.

### [3.1.3.](#) Compact and Tractable

The ASCII encoding is ready to deploy with a mature infrastructure. The name of a sign with 4 symbols is 60 characters long. The plain text model fully supports the grammar of written ASL with an additional 350 characters of basic HTML and CSS. The stand alone JavaScript engine for client side viewing is 1.3 K characters and qualifies as a micro script. This script can be applied to any modern browser through a site script or initiated within a browser using a bookmark.

To search for a sign with 4 spatial symbols requires 53 characters of



query string and will create around 800 characters of regular expression.

### **3.2. Visual Image**

The visual image of a logographic sign is a 2-dimension arrangement of symbols inside of a sign box. The sign box has a defined width, height, and 2-dimensional center that can be calculated from the plain text. The SVG created by the SignWriting Icon Server is print quality.

#### **3.2.1. TrueType Font**

Ready for experimental use with several open issues. The entire ISWA 2010 is included with 2-dimensional arrangements of symbols for the logograms. The TrueType Font utilizes the temporary Unicode characters from the Private Use Area.

There are 4 open issues: the symbols are fuzzy, handshapes overlap incorrectly, arrow head/tail fill is missing, and Graphite occasionally crashes.

#### **3.2.2. Server Generated SVG**

The SignWriting Icon Server (open source on GitHub) is able to create logographic sign images from the mathematical names. The SVG is grammatically correct and print quality.

Each SignWriting Icon Server provides the SignWriting Thin Viewer as a site script and as a bookmark. The main SignWriting Icon Server is available on Wikimedia Labs and open to all. The backup SignWriting Icon Server is available on SignBank.org. New SignWriting Icon Servers can be created directly from the GitHub source.

### **3.3. Character Encoding Scheme**

Encoding schemes define how a character is written as a sequence of bytes. SignWriting Text can use any encoding schemes that supports ASCII or Unicode.

Given a sequence of bytes representing text and a stated character encoding scheme, a string of characters is unambiguous and it is easy to recreate a sequence of characters as required for plain text.

#### **3.3.1. ASCII**

Every logographic sign has a mathematical name in ASCII. ASCII is universally supported. The ASCII names are authoritative and easy to



identify. Searching with regular expressions is 4 times faster in ASCII than the equivalent Unicode.

### **3.3.2. Unicode**

Every logographic sign has a temporary name of Unicode PUA characters for client side font handling. The use of the Unicode PUA demonstrates the necessity and the capability of the proposed character set.

## **3.4. Coded Character Set**

A character is a fundamental building block of digital data. A character's smallest representation is a binary representation of a code point found in a character set. A string is an ordered sequence of characters, which is nothing more than a list of code points.

### **3.4.1. x-ISWA-2010**

The x-ISWA-2010 is a 16-bit character set that covers each symbol of the ISWA 2010. A 16-bit code is an integer between 0 and 65,535. This type of value is perfect for a primary key for database lookup or other integer index. Through a simple formula, any symbol identification can be transformed into a unique 16-bit codepoint. Font software using the SQLite fonts rely on the x-ISWA-2010 coded character set.

There are 652 BaseSymbols in the ISWA 2010, numbered from 1 to 652. Each BaseSymbol can be visualized on a grid of 6 columns and 16 rows: for the 6 fills and 16 rotations. Each symbol can be identified by 3 values of BaseSymbol, column and row.

The codes of the x-ISWA-2010 are assigned starting with the first BaseSymbol grid. The first symbol is given a code value of 1 and the codes are incremented down the first column, continue to the next column, and continue through the remaining BaseSymbols.

Given any symbol with:

BaseSymbol number =  $n$

Fill =  $f$

Rotation =  $r$

code =  $(n-1)*96 + (f-1)*16 + r$





### 3.4.2. x-Binary-SignWriting

The x-Binary-SignWriting is a 12-bit character set that covers the characters of SignWriting Plain Text. It is possible to write the name of a logographic sign with binary data. This is more of a theoretical advantage because we don't write with 12-bit characters. This form is most useful for the translation to Private Use Area Unicode.

#### x-Binary-SignWriting Character

Name	Token	BSW Codepoint(s)
Sequence Marker	A	B+100
SignBox Marker	B	B+101
Left Lane Marker	L	B+102
Middle Lane Marker	M	B+103
Right Lane Marker	R	B+104
Columns 1 thru 6 (fills)	i	B+110 - B+115
Rows 1 thru 16 (rotations)	o	B+120 - B+12F
Writing BaseSymbols	w	B+130 - B+3AE
Detailed Location BaseSymbols	s	B+3AF - B+3B6
Punctuation BaseSymbols	P	B+3B7 - B+3BB
Negative Numbers: -250 thru -1	n	B+706 - B+7FF
Positive Numbers: 0 thru 249	n	B+800 - B+8F9

Table 2

### 3.4.3. x-Character-SignWriting

The x-Character-SignWriting is a character set for SignWriting in Unicode. Take the characters of the x-Binary-SignWriting coded character set and add hexadecimal value FD700. The characters follow the same token patterns as x-Binary-SignWriting defined in [Section 3.4.2](#).



## x-Character-SignWriting Characters

Name	Token	Unicode PUA
Sequence Marker	A	U+FD800
SignBox Marker	B	U+FD801
Left Lane Marker	L	U+FD802
Middle Lane Marker	M	U+FD803
Right Lane Marker	R	U+FD804
Columns 1 thru 6 (fills)	i	U+FD810 - U+FD815
Rows 1 thru 16 (rotations)	o	U+FD820 - U+FD82F
Writing BaseSymbols	w	U+FD830 - U+FDAAE
Detailed Location BaseSymbols)	s	U+FDAAF - U+FDAB6
Punctuation BaseSymbols	P	U+FDAB7 - U+FDABB
Negative Numbers: -250 thru -1	n	U+FDE06 - U+FDEFF
Positive Numbers: 0 thru 249	n	U+FDFF0 - U+FDFF9

Table 3

### 3.5. Character Encoding Form

The character encoding form for SignWriting text are based on ASCII or Unicode. The standard Unicode CEFs of UTF-8, UTF-16, or UTF-32 can be used. For ASCII, an additional mapping layer of a lite markup is used.

#### 3.5.1. Lite Markup

ASCII characters are used to identify structure, symbols, and coordinates. It has proven to be beneficial to use a human readable lite markup of ASCII words separated by white space. Each word represents either a signbox or a punctuation. The lite markup has the advantage of a small size without requiring special Unicode or XML functions. Simple regular expressions can quickly and efficiently process the lite markup.



In the lite markup, the structural markers use the token values as the character representation.

#### Structural Marker Tokens

Token	Description
A	Sequence Marker
B	SignBox Marker
L	Left Lane Marker
M	Middle Lane Marker
R	Right Lane Marker

Table 4

In the lite markup, symbols are referenced by symbol keys: the letter 'S' followed by 5 hexadecimal values, 3 characters for the symbol base and 2 characters for the modifiers.

In the lite markup, there are 2 types of coordinates: regular fixed-width coordinates and irregular variable-width coordinates. Both types of coordinates contain 2 numbers separated by the letter 'x'.

In the lite markup, regular coordinates are always 7 ASCII characters long: 3 digits followed by the letter 'x' followed by 3 more digits. The numbers range from 250 to 749, with 500 being the center point as zero. So for regular coordinates, the string "250" is equal to the number value of -250 and "749" is equal to the number value of 249. The loose definition of regular coordinates matches numbers with 3 digits without specifying the number range. It has a regular expression of `/[0-9]{3}x[0-9]{3}/`. The strict definition of regular coordinates only matches numbers in the range from 250 to 749. It has a more verbose regular expression of `/(2[5-9][0-9]|[3-6][0-9]{2}|7[0-4][0-9])x(249|2[5-9][0-9]|[3-6][0-9]{2}|7[0-4][0-9])/`.

In the lite markup, irregular coordinates are variable width. The numbers can be positive or negative. For negative numbers, the '-' minus sign is replaced with the letter 'n'. The two numbers in the coordinate are separated by the letter 'x'. The center coordinate of (0,0) is represented by the string '0x0'. The coordinate (-250,-250) is represented by the string 'n250xn250'.



Although signs have a coordinate number limit of -250 to 249, irregular coordinates are unbounded when used for display with compounds of multiple signs and punctuation.

### 3.5.2. Formal SignWriting

Formal SignWriting is the standard format for storing the names of the signs. It uses a lite markup with the token values for structural markers (A, B, L, M, R), symbol keys, and regular coordinates. White space is used to separate words of signs and punctuation.

Regular Expressions of Formal SignWriting

Stru	Regular Expression
ctur	
e	
Symb	S[123][0-9a-f]{2}[0-5][0-9a-f]
ol	
key	
Coor	[0-9]{3}x[0-9]{3}
dina	
te	
Sign	[BLMR]([0-9]{3}x[0-9]{3})(S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*
box	
Term	(A(S[123][0-9a-f]{2}[0-5][0-9a-f]))+[BLMR]([0-9]{3}x[0-9]{3})(S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*
Punc	S38[7-9ab][0-5][0-9a-f][0-9]{3}x[0-9]{3}
tuat	
ion	
Text	((A(S[123][0-9a-f]{2}[0-5][0-9a-f]))+[BLMR]([0-9]{3}x[0-9]{3})(S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*)S38[7-9ab][0-5][0-9a-f][0-9]{3}x[0-9]{3}((A(S[123][0-9a-f]{2}[0-5][0-9a-f]))+[BLMR]([0-9]{3}x[0-9]{3})(S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*)S38[7-9ab][0-5][0-9a-f][0-9]{3}x[0-9]{3})*

Table 5





### 3.5.3. Kartesian SignWriting

Kartesian SignWriting is an alternate encoding form with several types of display variants. It uses a lite markup with the token values for structural markers (A, B, L, M, R), symbol keys, and irregular coordinates. White space is used to separate words of signs and punctuation.

Each format uses a lite markup with the token values for structural markers (A, B, L, M, R), symbol keys, and irregular coordinates. Spaces separate words for signs and punctuation.

Regular Expressions of Formal SignWriting

+-----+	-----+
Structure	Regular Expression
+-----+	-----+
Symbol key	S[123][0-9a-f]{2}[0-5][0-9a-f]
+-----+	-----+
Coordinate	n?[0-9]+xn?[0-9]+
+-----+	-----+

Table 6

#### 3.5.3.1. Raw

The raw display format string contains the minimal amount of data required to represent text. It defines signs and punctuations. The signboxes are neither centered or sized. A signbox can occur anywhere in the signbox space and the center is not assumed to be the coordinate (0,0). The maximum coordinate for a signbox is unstated. Likewise, the punctuation does not contain any placement information. Layout is impossible without access to an outside datasource.

A sign is a combination of a lane maker (BLMR), followed by zero or more symbol keys with placement coordinates.

A punctuation is represented with a single symbol key.



## Regular Expressions of Kartesian SignWriting Raw

Struc ture	Regular Expression
Signb ox	[BLMR](S[123][0-9a-f]{2}[0-5][0-9a-f]n?[0-9]+xn?[0-9]+)*
Term prefi x	A(S[123][0-9a-f]{2}[0-5][0-9a-f])+
Punct uatio n	S38[7-9ab][0-5][0-9a-f]
Text	((A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?[BLMR](S[123][0-9a-f]{2}[0-5][0-9a-f]n?[0-9]+xn?[0-9]+)* S38[7-9ab][0-5][0-9a-f])(A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?[BLMR](S[123][0-9a-f]{2}[0-5][0-9a-f]n?[0-9]+xn?[0-9]+)* S38[7-9ab][0-5][0-9a-f])*

Table 7

**3.5.3.2. Expanded**

The expanded display format string contains sizing information (width and height) for every symbol outside of the term prefix. The maximum coordinate for a signbox can be calculated by adding the symbol width and height to the symbol placement coordinate.

For any symbol key in the signbox or for punctuation, the width and height is accessed from an outside data source. The size information is written as an irregular coordinate and appended to the symbol key through a simple search and replace.

A sign is a combination of a lane maker (BLMR), followed by zero or more symbol keys with sizing information followed by placement coordinates.

A punctuation is represented with a symbol key and a size coordinate



## Regular Expressions of Kartesian SignWriting Expanded

Stru	Regular Expression
ctur	
e	
Sign	[BLMR](S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]+x[0-9]+xn?[0-9]
box	+xn?[0-9]+)*
Term	A(S[123][0-9a-f]{2}[0-5][0-9a-f])+
pref	
ix	
Punc	S38[7-9ab][0-5][0-9a-f][0-9]+x[0-9]+
tuat	
ion	
Text	((A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?[BLMR](S[123][0-9a-f]
	]{2}[0-5][0-9a-f][0-9]+x[0-9]+xn?[0-9]+xn?[0-9]+)* S38[7-9
	ab][0-5][0-9a-f][0-9]+x[0-9]+)(
	(A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?[BLMR](S[123][0-9a-
	f]{2}[0-5][0-9a-f][0-9]+x[0-9]+xn?[0-9]+xn?[0-9]+)*
	S38[7-9ab][0-5][0-9a-f][0-9]+x[0-9]+)*

Table 8

**3.5.3.3. Layout**

The layout display format string contains the maximum coordinate as a preprocessed value for signboxes and it contains the placement coordinate for punctuation. It is equivalent to the lite markup for the regular searching form, but with irregular coordinates.

A sign is a combination of a lane maker (BLMR), followed by the maximum coordinate, followed by zero or more symbol keys with placement coordinates.

A punctuation is a combination of a symbol key followed by a placement coordinate. The center is assumed to be the coordinate (0,0). The maximum coordinate is the additive inverse of the placement coordinate.



## Regular Expressions of Kartesian SignWriting Layout

Stru	Regular Expression
ctur	
e	
Sign	[BLMR]([0-9]+x[0-9]+)(S[123][0-9a-f]{2}[0-5][0-9a-f]n?[0-9]
box	]+xn?[0-9]+)*
Term	A(S[123][0-9a-f]{2}[0-5][0-9a-f])+
pref	
ix	
Punc	S38[7-9ab][0-5][0-9a-f]n?[0-9]+xn?[0-9]+
tuat	
ion	
Text	((A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?[BLMR]([0-9]+x[0-9]+
	)(S[123][0-9a-f]{2}[0-5][0-9a-f]n?[0-9]+xn?[0-9]+)* S38[7-
	9ab][0-5][0-9a-f]n?[0-9]+xn?[0-9]+)(
	(A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?[BLMR]([0-9]+x[0-9]
	)+(S[123][0-9a-f]{2}[0-5][0-9a-f]n?[0-9]+xn?[0-9]+)*
	S38[7-9ab][0-5][0-9a-f]n?[0-9]+xn?[0-9]+)*

Table 9

**3.5.3.4. Panel**

A panel display format string combines multiple signs and punctuations into a unit as a defined height column or defined width row. Each signbox contains an offset coordinate that is used to position the symbols inside of the signbox. The offset is added to the placement coordinate to determine the position of each symbol on the panel.

Each panel begins with a panel display marker "D" followed by a sizing coordinate. The top-left of the panel is taken to be the coordinate (0,0) such that the sizing coordinate can be understood as the width and height of the panel as well as the maximum coordinate.

Each panel can contain several signboxes. Each signbox has its own offset coordinate. The offset coordinate is used to determine the position of the signbox's symbols within the panel.

A full panel includes the panel prefix with several signboxes with offsets.





## Regular Expressions of Kartesian SignWriting Display

Struc ture	Regular Expression
Signbox with offsets	<code>_[BLMR]([0-9]+x[0-9]+)(S[123][0-9a-f]{2}[0-5][0-9a-f]n?[0-9]+xn?[0-9]+)*</code>
Panel	<code>D[0-9]+x[0-9]+(_[BLMR]([0-9]+x[0-9]+)(S[123][0-9a-f]{2}[0-5][0-9a-f]n?[0-9]+xn?[0-9]+)*)*</code>
Panel s	<code>D[0-9]+x[0-9]+(_[BLMR]([0-9]+x[0-9]+)(S[123][0-9a-f]{2}[0-5][0-9a-f]n?[0-9]+xn?[0-9]+)*)*( D[0-9]+x[0-9]+(_[BLMR]([0-9]+x[0-9]+)(S[123][0-9a-f]{2}[0-5][0-9a-f]n?[0-9]+xn?[0-9]+)*)*)*</code>

Table 10

### 3.6. Query Language

The query language is an ASCII lite markup similar to FSW used to search. A query will compile to a series of regular expression to search a section of text to find similar or exact sign matches. Modern SignWriting [section 9](#) clearly illustrates the searching available and the associated regular expression technology.

The query string is a concise representation for a much larger set of regular expression statements. The query string permits several types of searches for symbols, ranges and spatial relation.

## 4. ISWA 2010

The ISWA 2010 is the abstract symbolset for the x-ISWA-2010 coded character set. The symbols are visually iconic, uniquely identified, and organized in a layered hierarchy ([Section 4.3](#)).

The x-ISWA-2010 is a 16-bit coded character used in the font software to access the symbol glyphs.

The x-Binary-SignWriting is a 12-bit coded character set that does not directly encode the symbols of the ISWA 2010, but divides each symbol into a combination of 3 characters. The first character



represents the base of the symbol. The next represents the fill of the symbol. The last character represents the rotation of the symbol.

#### **4.1. Grapheme**

The grapheme is the fundamental unit of writing for the SignWriting script. Many graphemes of SignWriting are visually iconic. The main writing graphemes of SignWriting represent a visual conception: either hands, movement, dynamics, timing, head, face, trunk, or limb. The body concept is a combination of trunk and limb. The specific size and shape of each grapheme is designed to balance and complement other graphemes.

The writing graphemes are extensive and specifically organized for written sign language and sign gestures. The writing graphemes do not include the specific graphemes of DanceWriting or the general graphemes of MovementWriting.

The writing graphemes are used in clusters. A cluster is a spatial grouping of graphemes written as a single unit. The graphemes can overlap and obscure graphemes underneath. A cluster can represent a sign of a sign language or a visual performance of a sign gesture.

Detailed location graphemes are separate from the main writing graphemes. Detailed location graphemes are used individually or sequentially. They represent isolated analysis that is written outside the cluster.

Punctuation graphemes are used when writing sentences. They are used individually, between clusters.

When written by hand, lines are drawn to form each grapheme. Different styles draw different types of lines: either for personal taste, speed, or quality. The main types of handwriting are formal, cursive, and shorthand. Formal handwriting, equivalent to block printing, includes defined lines for all grapheme features, specific palm facings for hand shapes, and detailed arrow heads and tails. Cursive handwriting is more fluid and less detailed. Handwriting for personal use can omit palm facings, generalize arrows, and other liberties of personal consumption. Shorthand is a further reduction of detail, written for speed. Shorthand is a memory aid to a written record and should be rewritten soon after the notes were taken.

Understanding the ratios of size and shape for the graphemes improves hand writing. SignWriting was an exclusively handwritten script for 7 years before publishing formalized the Block Printing model.



## 4.2. Symbol

There are 37,811 symbols, each with a unique ID. A symbol ID is a sequence of six formatted numbers of increasing detail. The first dashed number defines the category (11). The first two dashed numbers define the group (11-22). The first four dashed numbers define a base (11-22-333-44). The fifth number represents the fill (55). The sixth number represents the rotation (66). A symbol ID is a combination of base ID with a valid fill and a valid rotation. A symbol ID has the format "nn-nn-nnn-nn-nn", where each "n" is a digit from 0 to 9.

The fill modifier can best be understood through the palm facing of the hand graphemes. The palm facing is based on planes. The SignWriting script uses two planes: the Front Wall (Frontal Plane) and the Floor (Transverse Plane). There are 6 palm facings. The first three palm facings are parallel with the Front Wall. The second three palm facings are parallel with the Floor. The reader can view the signer from different viewpoints (expressive or receptive) and can view the hands from different perspectives (front or top), but no matter what the viewpoint or perspective, the first three Fills represent the palm facing parallel to the Front Wall and the second three Fills represent the palm facing parallel to the Floor.

Fill	Indicator	Meaning
01	grapheme with white palm	reader sees palm of hand parallel Front Wall
02	grapheme with half black palm	reader sees side of hand parallel Front Wall
03	grapheme with black palm	reader sees back of hand parallel Front Wall
04	grapheme with white palm and broken line	reader sees palm of hand parallel Floor
05	grapheme with half black palm and broken line	reader sees side of hand parallel Floor
06	grapheme with black palm and broken line	reader sees palm of hand parallel Floor

Table 11



The fill modifier is redefined for the movement arrows of category 2.

Fill	Indicator	Meaning
01	a grapheme with a black arrow head	movement of the right hand
02	a grapheme with a white arrow head	movement of the left hand
03	a grapheme with a thin, unconnected arrow head	spatial overlapping of movement arrows for the left and right hands when they move as a unit
04	Irregular arrow stems	building blocks for complex movement

Table 12

The rest of the other bases use a fill modifier for grouping and visual organization that is meaningful only for a particular base symbol or small set.

The rotation modifier can best be understood through the hand symbols. The first 8 rotations progress 45 degrees counter clockwise. The last 8 rotations are a mirror of the first 8 and progress 45 degrees clockwise. Zero (0) degrees is understood to point to the top of the grapheme.





Rotation	Direction	Degrees from top
01	Counter Clockwise	0
02	Counter Clockwise	45
03	Counter Clockwise	90
04	Counter Clockwise	135
05	Counter Clockwise	180
06	Counter Clockwise	225
07	Counter Clockwise	270
08	Counter Clockwise	315
09	Clockwise	0
10	Clockwise	45
11	Clockwise	90
12	Clockwise	135
13	Clockwise	180
14	Clockwise	225
15	Clockwise	270
16	Clockwise	315

Table 13

#### 4.3. Hierarchy

The symbols of the ISWA 2010 are placed in a layered hierarchy for organization and access. There are 4 levels to the ISWA 2010 hierarchy: category, group, base, and symbol.

There are 7 categories. The first number of the symbol ID identifies the category. The first 5 categories contain writing symbols for use in clusters: 1) Hands, 2) Movement, 3) Dynamics & Timing, 4) Head & Face, and 5) Body. The Body category can be broken into 2



subcategories: 5.1) Trunk and 5.2) Limb.

The 6th category is Detailed Location that contains symbols used alone or in sequence, always outside the cluster. The 7th category is Punctuation that contains symbols used between clusters for text.

#### The 7 Categories of the ISWA 2010

Cat	Purpose	Name	Description
1	Writing	Hands	Handshapes from over 40 Sign Languages are placed in 10 groups based on the numbers 1-10 in American Sign Language.
2	Writing	Movement	Contact symbols, small finger movements, straight arrows, curved arrows and circles are placed into 10 groups based on planes: The Front Wall Plane includes movement that is "parallel to the front wall" and the Floor Plane includes movement that is "parallel to the floor".
3	Writing	Dynamics & Timing	Dynamics Symbols are used to give the "feeling" or "tempo" to movement. They provide emphasis on a movement or expression, and combined with Punctuation Symbols become the equivalent to Exclamation Points. The Tension Symbol, combined with Contact Symbols, provides the feeling of "pressure", and combined with facial expressions can place emphasis or added feeling to an expression. Timing symbols are used to show alternating or simultaneous movement.
4	Writing	Head & Face	Starting with the head and then from the top of the face and moving down.



5	Writing	Body	Torso movement, shoulders, hips, and the limbs are used in Sign Languages as a part of grammar, especially when describing conversations between people, called Role Shifting, or making spatial comparisons between items on the left and items on the right.
6	Detailed Location	Detailed Location	Detailed Location symbols used are used alone or in sequence outside of the cluster. They may be useful for sorting large dictionaries, refining animation, simplifying translation between scripts and notation systems, and for detailed analysis of location sometimes needed in linguistic research.
7	Punctuation	Punctuation	Punctuation symbols are used when writing complete sentences or documents in SignWriting.

Table 14

There are 30 groups. The first 2 dashed numbers in the symbol ID identify the group. The 30 groups can be divided into 3 sets of 10. The first ten are hands, category 1. The second ten are movements, category 2. The third ten are categories 3 thru 7. In order, 1 group for the Dynamics & Timing category, 1 for Head, 4 for Face, 1 for Trunk, 1 for Limb, 1 for Detailed Location, and 1 for Punctuation.



The 30 groups with symbol ID segment.

First Set	Second Set	Third Set	
01-01 Index	02-01 Contact	03-01 Dynamics &	
		Timing	
01-02 Index	02-02 Finger Movement	04-01 Head	
Middle			
01-03 Index	02-03 Straight Wall	04-02 Brow Eyes	
Middle Thumb	Plane	Eyegaze	
01-04 Four	02-04 Straight	04-03 Cheeks Ears	
Fingers	Diagonal Plane	Nose Breath	
01-05 Five	02-05 Straight Floor	04-04 Mouth Lips	
Fingers	Plane		
01-06 Baby Finger	02-06 Curves Parallel	04-05 Tongue Teeth	
	Wall Plane	Chin Neck	
01-07 Ring Finger	02-07 Curves Hit Wall	05-01 Trunk	
	Plane		
01-08 Middle	02-08 Curves Hit Floor	05-02 Limbs	
Finger	Plane		
01-09 Index Thumb	02-09 Curves Parallel	06-01 Detailed	
	Floor Plane	Location	
01-10 Thumb	02-10 Circles	07-01 Punctuation	

Table 15

There are 652 bases. The first 4 dashed numbers of a symbol ID identify the base. The 652 bases are divided between the 30 groups. For each group, there are less than 60 bases. The bases are often displayed in columns of 10.

Each base can have up to 96 symbols. All 6 dashed numbers of the symbol ID are required to identify a symbol. Each symbol is a combination of a base, fill, and rotation. The fill is identified by the 5th number of the symbol ID with possible values from 01 to 06. The rotation is identified by the 6th number of the symbol ID with possible values from 01 to 16.





#### 4.4. Combined Character Sequence

Each symbol of the ISWA 2010 can be expressed with a combination of 3 characters. The first character represents the base of the symbol. The next character represents the fill of the symbol. The last character represents the rotation of the symbol.

There are three forms the fill and rotation can use to represent their value: a hexadecimal key, an x-Binary-SignWriting character, or an x-Character-SignWriting character.

The x-Binary-SignWriting coded character set uses a 12-bit encoding. Code points in this set use a "B+" prefix along with the 3 hexadecimal digits that represent the value.

The x-Character-SignWriting coded character set uses the Private Use Area of Unicode. These code points occur on plane 15. Code points in this set use a "U+" prefix along with the 5 hexadecimal digits that represent the value.

The fill value ranges from 1 to 6. The fill key is 1 less than the value and ranges from 0 to 5.

Fill Value	Key	x-Binary-SignWriting	x-Character-SignWriting
1	0	B+110	U+FD810
2	1	B+111	U+FD812
3	2	B+112	U+FD812
4	3	B+113	U+FD813
5	4	B+114	U+FD814
6	5	B+115	U+FD815

Table 16

The rotation value ranges from 1 to 16. The rotation key is written in hexadecimal and is equal to 1 less than the value and ranges from "0" to "f".



Rotation Value	Key	x-Binary-SignWriting	x-Character-SignWriting
1	0	B+120	U+FD820
2	1	B+121	U+FD821
3	2	B+122	U+FD822
4	3	B+123	U+FD823
5	4	B+124	U+FD824
6	5	B+125	U+FD825
7	6	B+126	U+FD826
8	7	B+127	U+FD827
9	8	B+128	U+FD828
10	9	B+129	U+FD829
11	a	B+12A	U+FD82A
12	b	B+12B	U+FD82B
13	c	B+12C	U+FD82C
14	d	B+12D	U+FD82D
15	e	B+12E	U+FD82E
16	f	B+12F	U+FD82F

Table 17

Further, a 16 bit symbol code from the x-ISWA-2010 exists for each of the valid combined character sequences. This relationship can be stated as (symbol code = ((base code - 256) \* 96) + ((fill value - 1) \* 16) + rotation value). The first symbol code is 1 and the last valid symbol code is 62,504.

The first symbol has an ID of "01-01-001-01-01-01" and a symbol code of 1.



Symbol code 1 = symbol key S10000 = B+130, B+110, B+120 = U+FD830, U+FD810, U+FD820.

Symbol code 1 = ( ( hexdec('100') - 256 ) \* 96 ) + ( ( fill\_value(1) - 1 ) \* 16 ) + rotation\_value(1).

Symbol code 1 = ( ( 256 - 256 ) \* 96 ) + ( ( 1 - 1 ) \* 16 ) + 1.

Symbol code 1 = ( 0 \* 96 ) + ( 0 \* 16 ) + 1.

Symbol code 1 = 1.

#### 4.5. Validity

Although there are 6 possible fills and 16 possible rotations, not every combination of base, fill, and rotation is valid. Each base has a set of valid fills and a set of valid rotation. These validity sets contain one or more values from the defined range.

For each value, the inclusion in the validity set can be expressed with a value of "0" or "1". For fill values, lining up the digit from left to right, will result in a string 6 digits long. The value of the 6 digit number is  $2^{(\text{value} - 1)}$ .

Fill Value	1	2	3	4	5	6	Binary	Power of 2
1	X						100000	1
2		X					010000	2
3			X				001000	4
4				X			000100	8
5					X		000010	16
6						X	000001	32

Table 18

The value of any fill validity set is equal to the sum of the power of 2 for each fill value in the set. The empty set is invalid and has a sum of zero (0). The full set of all possible fills has a sum of 63.



Fill Set	1	2	3	4	5	6	Binary	Power of 2
{}							000000	0
{1,2,3,4,5,6}	X	X	X	X	X	X	111111	63

Table 19

Each base has a defined validity set for fills. The "Fills" column in the "Bases" section.

The rotation validity sets have a larger range than the fills. The possible rotation values range from 1 to 16. The power of 2 numbers are 16-bit.

Value	Binary	Power of 2
1	2 <sup>0</sup>	1
2	2 <sup>1</sup>	2
3	2 <sup>2</sup>	4
4	2 <sup>3</sup>	8
5	2 <sup>4</sup>	16
6	2 <sup>5</sup>	32
7	2 <sup>6</sup>	64
8	2 <sup>7</sup>	128
9	2 <sup>8</sup>	256
10	2 <sup>9</sup>	512
11	2 <sup>10</sup>	1024
12	2 <sup>11</sup>	2048
13	2 <sup>12</sup>	4096
14	2 <sup>13</sup>	8192





+-----+	+-----+	+-----+	+-----+
15	2 <sup>14</sup>	16384	
+-----+	+-----+	+-----+	+-----+
16	2 <sup>15</sup>	32768	
+-----+	+-----+	+-----+	+-----+

Table 20

The value of a rotation validity set is the summation of the power of 2 numbers. The minimum summation is 1. The largest possible summation is 65,535 where all 16 rotations are valid.

Each base has a defined validity set for rotations. The "Rotations" column in the "Bases" section.

Interestingly enough, there are only 12 possible validity sets in the ISWA 2010.

+-----+	+-----+	+-----+	+-----+
Sum	Binary	Set	
+-----+	+-----+	+-----+	+-----+
1	100000	{1}	
+-----+	+-----+	+-----+	+-----+
2	010000	{2}	
+-----+	+-----+	+-----+	+-----+
3	110000	{1, 2}	
+-----+	+-----+	+-----+	+-----+
7	111000	{1, 2, 3}	
+-----+	+-----+	+-----+	+-----+
15	111100	{1, 2, 3, 4}	
+-----+	+-----+	+-----+	+-----+
31	111110	{1, 2, 3, 4, 5}	
+-----+	+-----+	+-----+	+-----+
63	111111	{1, 2, 3, 4, 5, 6}	
+-----+	+-----+	+-----+	+-----+
187	11011101	{1, 2, 4, 5, 6, 8}	
+-----+	+-----+	+-----+	+-----+
255	11111111	{1, 2, 3, 4, 5, 6, 7, 8}	
+-----+	+-----+	+-----+	+-----+
511	1111111110000000	{1, 2, 3, 4, 5, 6, 7, 8, 9}	
+-----+	+-----+	+-----+	+-----+
48059	1101110111011101	{1, 2, 4, 5, 6, 8, 9, 10, 12, 13, 14,	
		16}	
+-----+	+-----+	+-----+	+-----+
65535	1111111111111111	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,	
		12, 13, 14, 15, 16}	
+-----+	+-----+	+-----+	+-----+



Table 21

## **5. SignPuddle Standard**

The SignPuddle Standard for SignWriting text is nearing a stable and fully functional version 1.

### **5.1. Licenses**

#### **5.1.1. Open Font License**

Our font software is available under SIL's Open Font License.

#### **5.1.2. Creative Commons**

Our reference material is licensed under Creative Commons attribution, share alike (by-sa).

#### **5.1.3. GPL**

The current open source projects are licensed under the GPL 2 for MediaWiki and GPL 3 for the general software on Github. Any contributions to the open source code must agree to a possible relicense in the future under a BSD like license.

#### **5.1.4. BSD**

After the financial issues [13] of the Center for Sutton Movement Writing have been addressed, the open source projects will relicensed under a more open and free BSD-like license, such as the MIT License.

### **5.2. Infrastructure**

#### **5.2.1. International SignWriting Alphabet Fonts**

The International SignWriting Alphabet 2010 (ISWA 2010) Font Reference [1] is a product of the collaboration between SignWriting inventor, Valerie Sutton, and SignWriting encoder Stephen E Slevinski Jr. Special thanks to Adam Frost's excellent work on the SVG refinement and more.

The ISWA 2010 fonts have been stable since their initial release on October 20th, 2010.

Valerie Sutton



- o hand crafted and organized 30K plus individual glyphs
- o created a 2 dimension PNG of 3 colors for each
- o named each individual glyph with 6 degrees of significance
- o font name: ISWA 2010 Sutton

Steve Slevinski

- o counted and numbered the glyphs
- o created mathematical names
- o analyzed PNGs for line and fill
- o refactored glyphs - font name: ISWA 2010 PNG Standard
- o extended glyphs - font names: ISWA 2010 PNG Inverse, Shadow, Colorized
- o traced glyphs - font names: ISWA 2010 SVG Line Trace, Shaddow Trace, Smooth, and Angular
- o refactored and extended Adam's SVG work - font name: ISWA 2010 SVG Refinement

Adam Frost

- o manually traced each and every glyph that could not be automatically rotated
- o font name: ISWA 2010 SVG Refinement
- o physically performed and photographed every hand shape
- o font name: ISWA 2010 Hand Photo
- o consulted with Valerie in places of ambiguity
- o found the Facial Irregularity, documented in the ISWA 2010 Errata

#### **5.2.2. SignWriting Icon Server**

The SignWriting Icon Server create SVG and PNG images and queries data collections using an open API. The image creation is stable and fully implemented. The API is currently under construction with only an initial level of support.



The main server is available on Wikimedia Labs [\[4\]](#) for all SignWriting projects.

A backup server is available on SignBank [\[3\]](#).

Additional SignWriting Icon Servers can be created directly from the GitHub source.

### **[5.2.3.](#) SignWriting Thin Viewer**

The SignWriting Thin Viewer uses JavaScript to wrap the sign names with basic HTML and CSS to fully supports the grammar of written ASL. This script can be applied to any modern browser through a site script or initiated within a browser using a bookmark.

#### **[5.2.3.1.](#) CSS Text Layout**

The SignWriting Thin Viewer use CSS to make SignWriting text behave more like logographic text. It uses simple math for layout. It has center data points for selecting text to copy and for searching text on a page. It uses images for individual signs and punctuation. It makes SignWriting text act more like text.

The current working prototype uses 12 CSS rules: 4 that cover every cluster, 4 that cover the data string, and 4 custom layout values for each cluster.

Common

- o position: relative;
- o background-repeat: no-repeat;
- o background-origin: content-box;
- o padding: 10px;

Data Span

- o display: table-cell;
- o vertical-align: middle;
- o font-size:0%;
- o height: inherit;

Individual





- o width: ?px
- o height: ?px
- o left: ?px
- o background-image: url(ht..

The width, height, and left values are easy to calculate using the character string. No need to access a database or wait for the image server.

The background-image must link to a SignWriting Icon Server. CSS rules will directly effect the '''url''' affecting the style of the rich text. Specify the looks of Headings 1 thru 6, bold, italic, or to indicate URL links.

### **5.3. Compatibility**

SignPuddle Online, ASL Wikipedia Project, SignTyp, SignWriter Studio, the DELEGS Editor, and more.

#### **5.3.1. SignPuddle Online**

SignPuddle Online [[14](#)] is the current home of the international community of online writers of the SignWriting Script. Online tools make it possible to create SignWriting dictionaries and documents directly on the web. Each collection is freely available as a small XML file [[15](#)]. Dozens of sign languages from around the world are represented. Each language can have several collections of SignWriting.

#### **5.3.2. Wikimedia Labs**

SignWriting has an open project on Wikimedia Labs [[16](#)]. The ASL Wikipedia Project [[17](#)] is in full swing. The Libras Wikipedia Project [[18](#)] may start soon.

In general, Wikimedia Labs creates virtual computers running Linux. They use a special tool called Puppet to configure the virtual servers. Wikimedia Labs allows you to create, manage, and analyze the virtual servers through a MediaWiki based application. Wikimedia Labs is deeply integrated but not always configured properly or documented.

Wikimedia Labs has created a project for SignWriting. I am a super user on Wikimedia Labs. I administer the SignWriting project. I can create virtual servers at will, each is called an instance. I have 2



instances running. The first is "ase10", the 10th server I created before I had everything properly configured and installed. I created "ase11" when I was trying to fix the catastrophic crash of the ASL Wikipedia. "ase11" is a basic server without MediaWiki or the SWMP.

For the public to view anything on Wikimedia Labs, you must use an IP from a limited pool. Each project has a limit of 0 IPs when it is first created. This number can be increased according to need.

I have 2 public IPs for SignWriting. The first is used by the ASL Wikipedia Project and points to "ase10". The second is currently used for the SignWriting Icon Server [4] installation for Wikimedia projects.

There is no BZS virtual server running on Wikimedia Labs. This needs to be created by a skilled and experienced Linux administrator through the Wikimedia Labs environment. BZS is pointing to the SignWriting Icon Server on "ase11".

You do not need a public IP to start development on Wikimedia Labs, only to be viewed by the public.

#### **5.3.3. SignTyp**

This standard is being integrated with the SignTyp linguistic coding system developed by Rachel Channon through an NSF grant.

Notation Systems [19] by Harry van der Hulst and Rachel Channon.

Why dynamic features? [20] by Harry van der Hulst and Rachel Channon.

Transcription systems as input to coding systems: SignWriting & SignTyp [21] by Charles Butler and Rachel Channon.

#### **5.3.4. SignWriter Studio**

SignWriter Studio [22] is a Windows-only compatible application by Jonathan Duncan. It has an alternate symbol selection technique. According to Valerie Sutton, it illustrates a unique insight into the hand shapes of the ISWA.

Jonathan Duncan writes:

SignWriter Studio has 4 ways to get the basic symbol base, and 3 ways to modify the selected base.



1) Select the base symbol from a complete list of base symbols organized in a tree view 2) Search for a hand symbol in hand search section by hand feature. 3) Select a symbol already present in the signbox. 4) Select a symbol from a Favorites section.

Then one of three choosers to define the fill and rotation will become available. 1)The hand chooser. 2)The arrow chooser. 3)The general chooser.

The Hand chooser is to quickly find the symbol for a certain, hand, plain(wall or floor), palm facing and rotation. The Hand Chooser also extends add a fourth palm facing to logically show all possible symbols in their most common uses. This chooser resembles the instruction manual explaining the use of hand shapes.

The Arrow Chooser is to quickly find arrows for a certain hand, plain(wall or floor) and rotation. This chooser resembles the instruction manual explaining the use of arrows.

The General Chooser is for symbols for which the two previous choosers do not work well and gives a grouped list of symbols for the base group.

#### **5.3.5. DELEGS Online**

The DELEGS Editor [23] from the University of Hamburg and C1 WPS GmbH in Germany is designed for Deaf Education. It is a tool for writing translation texts between spoken and signed languages.

Spoken language text is used to display horizontal SignWriting Text from left to right. The spoken language can appear beneath the sign or it can be hidden.

## **6. Unicode Integration**

SignWriting Text is integrated with Unicode in the Private Use Area.

### **6.1. Private Use Area Font Characters**

The Unicode PUA is a simple shift of the x-Binary-SignWriting coded character set. Each code is increased by decimal value 1,038,080 which is FD700 in hex. An experimental TrueType Font converts the Unicode PUA to create the visual images.



## 6.2. Proposal

A shift of the 12 bit characters of x-Binary-SignWriting by 1D700 will use the range U+1D800 to U+1DFFF, using eight 8-bit rows of Unicode Plane 1 known as the the SMP: Supplementary Multilingual Plane. These rows occur inside an unassigned section of the Notational systems.

These are the characters being used by the community. The gap between the ISWA 2010 symbols and the number sections illustrates two truths. First, the entire Sutton MovementWriting family will be encoded. Second, it doesn't really matter where the numbers are placed, perhaps plane 14.

The number characters encode the ruler principle with characters. The ruler principle is built in automatically for scripts written sequentially in one dimension. The number characters are needed for 2-dimensional logograms, where the spatial relationship between symbols is explicitly stated with X,Y Cartesian coordinates. Number characters may be a useful concept for other scripts and notations to support 2-dimensional script processing.

The entire set of characters is used for a plain text model of a 2-dimension logographic script with freeform placement of symbols.

Future additions to the ISWA 2010 will include essential hand shapes and new mouth shapes. New characters will extend the SignWriting Text model with minimal complications.

Future proposals will include the rest of the Sutton MovementWriting System.

## 7. IANA Considerations

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the code spaces of the Center for Sutton Movement Writing, in accordance with [RFC2978]. protocol, in accordance with BCP 26, [RFC2434].

See IANA: <http://www.rfc-editor.org/rfc/rfc2978.txt>

Conforms with RFC 2040.

There are three name spaces for the Center for Sutton Movement Writing that require definition and extension: x-ISWA-2010, x-Binary-SignWriting, and x-Character-SignWriting





SignWriting Text is an international standard with several coded character sets. These sets may require additional hand and mouth shapes.

The following terms are used here with the meanings defined in [BCP 26](#): "name space", "assigned value", "registration".

The following policies are used here with the meanings defined in [BCP 26](#): "Private Use", "First Come First Served", "Expert Review", "Specification Required", "IETF Consensus", "Standards Action".

## **8. Security Considerations**

None.

### URIs

- [1] <<http://signpuddle.net/iswa>>
- [2] <<https://github.com/Slevinski/swis>>
- [3] <<http://signbank.org/swis>>
- [4] <<http://swis.wmflabs.org>>
- [5] <<http://www.signwriting.org/lessons/cursive/curs007.html>>
- [6] <<http://movementwriting.org/csmw/>>
- [7] <<http://www.signwriting.org/library/history/hist005.html>>
- [8] <<http://www.signwriting.org/lessons/cursive/curs002.html>>
- [9] <[http://signpuddle.net/wiki/index.php/SignWriting\\_Script](http://signpuddle.net/wiki/index.php/SignWriting_Script)>
- [10] <<http://www.signwriting.org/lessons/cursive>>
- [11] <<http://www.signwriting.org/lessons/cursive/shorthand>>
- [12] <<http://www.signwriting.org/lessons/cursive/byhand5.html>>
- [13] <[http://signpuddle.net/wiki/index.php/The\\_Wall](http://signpuddle.net/wiki/index.php/The_Wall)>
- [14] <<http://signpuddle.org>>
- [15] <<http://signbank.org/signpuddle2.0/data/spml>>



- [16] <[https://labsconsole.wikimedia.org/wiki/Nova\\_Resource:Signwriting](https://labsconsole.wikimedia.org/wiki/Nova_Resource:Signwriting)>
- [17] <<http://ase.wikipedia.wmflabs.org>>
- [18] <<http://bzs.wikipedia.wmflabs.org>>
- [19] <<http://homepage.uconn.edu/~hdv02001/Articles-pdfs/131%20-%20Notation%20Systems.pdf>>
- [20] <<http://www.purdue.edu/tislr10/pdfs/van%20der%20Hulst%20Channon.pdf>>
- [21] <[http://www.signwriting.org/archive/docs7/sw0623\\_TISLR\\_2010\\_SignWriting\\_SignTyp\\_Poster.pdf](http://www.signwriting.org/archive/docs7/sw0623_TISLR_2010_SignWriting_SignTyp_Poster.pdf)>
- [22] <<http://signwriterstudio.com>>
- [23] <<http://www.delegs.com/DelegsPage>>
- [24] <<http://http://signpuddle.net/wiki/index.php/MSW>>
- [25] <<http://www.linkedin.com/groups/Searching-3-digit-number-simple-1066587.S.85595980?qid=9cb1768b-5413-4f7f-92b5-fbef2c243df8>>
- [26] <[http://signpuddle.net/wiki/index.php/SignWriting\\_Text\\_Reference](http://signpuddle.net/wiki/index.php/SignWriting_Text_Reference)>
- [27] <[http://signpuddle.net/wiki/index.php/MSW:Mathematical\\_Model](http://signpuddle.net/wiki/index.php/MSW:Mathematical_Model)>

## **Appendix A. Modern SignWriting**

This Internet Draft is in complete agreement with the theory and example workbook released on January 12th, 2012 called Modern SignWriting [24].

Modern SignWriting has example text and concretely defines the processes available. It fully documented the symbol encoding. The query language is by far the most important aspect of this design. Modern SignWriting [section 9](#) clearly illustrates the searching available and the associated regular expression technology. I discussed the model on the Regular Expressions Experts list of Linked In the end of 2011 [25].

Modern SignWriting is now part of the SignWriting Text Reference [26] and available in wiki form and PDF.



Entire sections of the Modern SignWriting document will be included in this I-D as progresses is made.

## **Appendix B. Cartesian SignWriting**

Cartesian SignWriting is the name of a script encoding model for SignWriting Block Printing. The mathematical model is defined by the SignWriting Text Language [27]. This language uses formal words to name terms, signs, and punctuation.

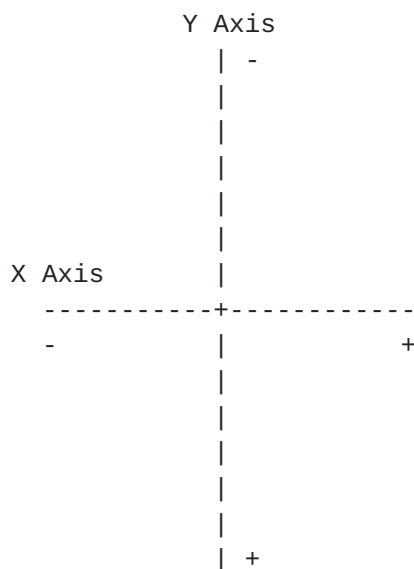
Formal structures of logographic sign are mixed with punctuation to form text. Each logographic sign is a 2-dimensional arrangement of symbols defined with cartesian coordinates.

Cartesian SignWriting is a heuristic model. The first prototypes were created in 2008. Through trial and error, the model was successively refactored to reduce the complexity and the computation cost of the implementations. The model has been optimized for common usage and processing.

### **B.1. Signbox**

Cartesian SignWriting uses coordinate based symbol placement.

Each logographic sign exists on its own 2-dimensional canvas. Each point on the canvas is identified with an X and a Y coordinate. Each canvas has a defined center. Formal numbers range from -250 to 249. Informal number have no limit.



Symbols are placed on the canvas with coordinates that represent the



top-left of the symbol image.

## **B.2. Temporal Order**

A term is a specialized sign that uses a sequential prefix before the 2-dimensional signbox.

A sequence is a list of writing symbols and/or detailed location symbols. A valid sequence must contain at least one symbol and can not contain punctuation. A sequence is an optional sign prefix used to define a temporal order.

The temporal order of a sign is distinct from the visual cluster. Neither structure can be derived from the other automatically. It requires human intelligence to correctly create the sequence from the signbox contents.

There are several theories on the best way to structure a sequence. The most productive is based on the SignSpelling Sequence theory of Valerie Sutton. A sequence is structured as a series of starting handshapes followed by optional movements, transitional handshapes, movement, and end handshapes. Only symbols from category 1 (hands) and category 2 (movement) should be used in this first section. The last section of the sequence should contain symbols of dynamics & timing, head & face, or body: categories 3, 4, and 5.

Detailed location symbols from category 6 can be used in a sequence, but are rarely (if ever) needed for a sequence in general writing.

## **B.3. Logograph of Logographs**

Cartesian SignWriting text uses a series of canvases, each with a unique coordinate space. A higher level coordinate space can be created to represent an entire panel of SignWriting Text. Either a column of vertical writing or a row of horizontal. The higher level coordinate space has an origin of (0,0). For columns, the panels share a common height. For rows, the panels share a common width.





```

      X Axis
      width
(0,0)  +-----
      |
Y  h  |
   e  |
A  i  |
x  g  |
i  h  |
s  t  |
      |
      |

```

The mathematics of the panel is defined in Modern SignWriting, [section 10](#).D Variant Display Form: Panel. The SignWriting Icon Server contains the functions required to convert a section of SignWriting Text into a series of panels. This can be useful for presentation.

The development of the rich text model defines a higher level logograph with manipulation of the DOM using CSS rules.

### [Appendix C](#). Theory of SignWriting Grammar and Encoding

Sign language is vastly different than spoken language. Instead of the sequential sounds of the voice, there is a 3 dimensional space with simultaneous action. The SignWriting Script creates 2-dimensional writing that is visually icon and full of featural information. This is true on the symbol level and on the sign level. A symbol represents phonemic information and is full of featural information to better understand the phonemes of the symbols. A sign is a 2-dimensional arrangement of symbols and is full of featural information to better understand the morphemes of the signs.

The 2 families of the SignWriting Script are Handwriting and Block Printing. The Handwriting family integrates with diacritic marks. The Block Printing family uses 2-dimensional placement with overlap and overlay.

Both of these families identify features in the writing they produce. Block Printing uses more features and Handwriting often uses less.

The Block Printing family is aimed at the needs of the reader and the publisher. The Block Printing family is ready to standardize with a fully developed model.

The Handwriting family is concerned with the needs of the writer.



The purpose is not to recreate the iconic symbols of the International SignWriting alphabet exactly by hand, but the purpose is to enable the writer to quickly write notes on paper or chalkboard. Handwriting often drops features of the SignWriting Script for efficiency and speed. If too many features are dropped, the writing may lose its clarity over time as the writer is distanced from the writing. This is common for Shorthand.

### **C.1. Logographic Sign**

A sign is a variably-size logographic word. It is a 2-dimensional combination of symbols inside of a signbox with a tight bounding box and an explicit center.

### **C.2. Punctuation and Text**

Punctuation separates signs into structured sentences. A punctuation symbol is always used alone and should not be used in a sign. Line breaks should not occur before punctuation.

### **C.3. Terms**

A term is a logographic sign with an optional prefix. The prefix is a sequential list of symbols that identify temporal order and additional analysis. Terms are special signs that are above the standard noise of SignWriting Text. The query language of Formal SignWriting support searching for general signs with the letter "Q" and searching for terms with the letters "QT".

### **C.4. Lanes**

When written vertically, SignWriting can use 3 different lanes: left, middle, and right. The middle lane is the default lane and punctuation is always used in the middle lane. No matter the lane, the center of a sign is aligned with the center of the lane.

For body weight shifts to one side or the other, the center of the cluster is aligned with a fixed horizontal offset from the middle lane into either the left or right lane.

The left and right lanes are used to represent body weight shifts and are represented by a horizontal offset from the middle lane. Body weight shifts are important to the grammar of sign languages, used for two different grammatical aspects: 1) role shifting during sign language storytelling, and 2) spatial comparisons of two items under discussion. One "role" or "item" is placed on the right side of the body (right lane), and the other on the left side of the body (left lane), and the weight shifts back and forth between the two, with the



narrator in the middle (middle lane).

### C.5. Modes

The most common writing mode is vertical.

Vertical Writing Mode

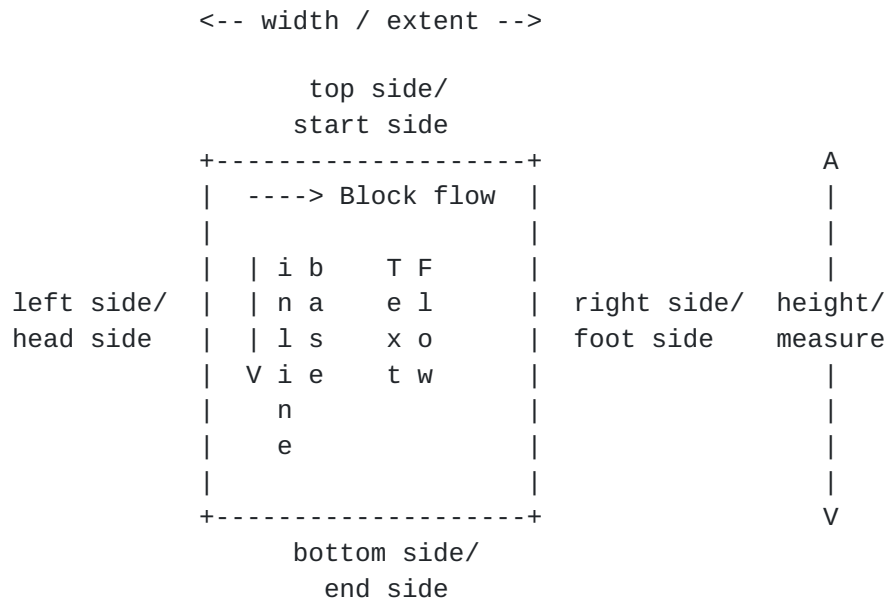


Figure 1

downward inline base

rightward block flow

vertical translate by word

variable dimensions of words

center of word aligns with the central baseline

variable over and under values from central baseline

The horizontal writing mode can lose or obfuscate important grammatical information, but is still useful, especially for translations with a spoken language.



## Horizontal Writing Mode

----> inline base

B f	Text
l l	Flow
o o	
v c w	
k	

Figure 2

**C.6. Layout**

The SignPuddle Standard for SignWriting Text uses a freeform layout with cartesian coordinates for absolute positioning. Additional layout options are included and explored.

The main issue of layout is how the writer will use the system. The balance between complexity and usability from the writer's perspective is of primary importance.

The second issue of layout involve comparison. Signs can quickly be scanned for the symbols used; however, the relative position of the symbols require an analysis of the layout. The different layouts offer different approaches for evaluation.

The third issues of layout involves variability. There are two types of variability. The first, inter-personal variability, occurs when writers pick different symbols and different details. Inter-personal variability is part of the writing system that layout can not resolve. The second, intra-personal variability, occurs when writers use the same symbols, but in slightly different positions. With layout choices, it is possible to reduce the intra-personal variability, but this reduction may harm the writing system by imposing too many restrictions on the writer.

A fourth issues of layout involves elegance and beauty. Some may consider one type of layout to be superior to another based on subjective personal opinions. SignWriting is a unique script. The ultimate choice of layout should be based on the writer's experience, comparison, and variability.

**C.6.1. Freeform**

With freeform layout, the writer decides what symbols to use and the exact symbol position. The freeform layout offers the greatest flexibility for the writer and the greatest intra-personal





variability.

Cartesian coordinates specify X and Y coordinates for the top, left of the symbol glyph. The coordinates of the symbols relate to the center of the canvas. The Cartesian Coordinate system is a more practical choices for computer processing because the equations of layout and comparison are easier. This is the current method for writing. The writer is presented with a canvas and positions each symbol independently.

Polar coordinates specify an angle and a distance from the center of the sign to the center of each symbol. Polar coordinates require the pythagorean theorem and the slope equation for standard processing.

#### **C.6.2. Restricted**

It is possible to impose restrictions on symbol placement thereby limiting the intra-personal variability of sign spellings.

For generic restrictions, instead of allowing any coordinates, it may be possible to limit the options. For example, with polar coordinates, only allow specific angles and specific distances. This has not been evaluated.

For specific restrictions it may be possible to perform a statistical analysis of the symbols used to come up with a limited number of attachment points around each symbol and a small list of predefined distances between symbols. This information would be symbol specific and could greatly reduce the intra-personal variability if successfully implemented.

#### **C.6.3. Non-form**

Some would argue that the writer should not determine the form of a sign, but should input linguistic analysis and let the layout/font manager determine the best representation for the written sign. This would change the script from a writing system into computer aided design, requiring concepts that are not part of the script and are not part of the writer's thought processes. The idea would make for an interesting project, but it is not about encoding SignWriting.

#### **C.7. Positioning**

Any of the above layout options have two choices for positioning: absolute or relative.



#### **C.7.1. Absolute**

The absolute position of each symbol relates to the center of the sign. The freeform layout section above is defined using absolute positioning.

#### **C.7.2. Relative**

A relative position relates the symbol position according to other symbols. This could be defined with a tree structure or a more complicated linked list. One or more root symbols could initialize the sign and other symbols would build from the roots. The restricted layout of polar coordinates is defined above using relative positioning.

The viability and usability of relative positioning is unknown and has not been investigated.

#### **Author's Address**

Stephen E Slevinski Jr  
SignPuddle

Email: [slevin@signpuddle.net](mailto:slevin@signpuddle.net)

