Sutton-Slevinski Collaboration
Internet-Draft
Intended status: Informational
Expires: November 10, 2016

S. Slevinski
SignPuddle

May 9, 2016

The SignPuddle Standard for SignWriting Text
draft-slevinski-signwriting-text-07

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.

Formal SignWriting uses ASCII strings to name logographic signs. The mathematical names are explained with tokens and regular expression patterns. Symbol keys reference the symbols of the International SignWriting Alphabet 2010. Coordinates define X and Y number values for 2-dimensional placement. Signs are written in a spatial SignBox, where each symbol is positioned with a 2-dimension coordinate. For sorting, each sign can have an optional temporal sequence of symbols that is outside of the SignBox and the visible text. To create sentences, completed signs are written sequentially, interspersed with punctuation symbols.

The query language of Formal SignWriting uses a lite markup, similar to FSW, to define a variety of searching possibilities. The spatial SignBox can be searched for symbols or ranges of symbols. For each symbol or range, the search can specify if the symbol only needs to be found somewhere in the SignBox, or if the symbol needs to be found near certain coordinates. The temporal sequence can be searched for starting symbols, written as a sequential list of symbols and ranges of symbols. When searching the temporal sequence, the search results will be limited to signs that start with a matching temporal sequence. Each query string is transformed into one or more regular expressions. The regular expressions are used to quickly search large amounts of data.

The styling string of Formal SignWriting uses a lite markup to define a variety of styling options. The entire sign can be customized for padding, coloring, and size. Individual symbols within a sign can be customized for coloring and size.

SignWriting 2010 is the modern implementation and international specification of the SignWriting Script for the internet community that includes TrueType Fonts and a compact JavaScript library. SignMaker is a standards based editor, utilizing HTML, CSS, JavaScript, SVG, TrueType Fonts, and PNG images. SignMaker can be used to create a private dictionary or to view dozens of sign language dictionaries derived from SignPuddle Online.

For Unicode, there are several encodings possibilities. Formal SignWriting is UTF-8. The plane 15 encoding is isomorphic with Formal SignWriting strings, using 3 characters for each symbol, along with structural marker characters and number characters. The plane 16 encoding is focused on the symbols only, using 1 character for each symbol. The Unicode 8 specification uses 1 to 3 characters on plane 1 to name each symbol of the International SignWriting Alphabet 2010.

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 symbol encoding of the International SignWriting Alphabet 2010. The third discusses the SignPuddle Standards: licences, infrastructure, and compatibility.

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, 2016.

Table of Contents

1. 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 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, Portugese 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.

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 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.

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 larger character encoding model has been stable since the initial release of Modern SignWriting on January 12th, 2012.

The 2 families of the SignWriting Script are Handwriting for the writer and Block Printing for the reader. Block Printing uses more features and Handwriting often uses less. Block printing is used in education, publishing, and is the basis of the computerized model.

1.1. 2-Dimensional Signs

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. The size of the signbox varies with the symbols written inside.

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.

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 signbox. 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 signbox. When SignWriting is used for authorship, the expressive view is most often used.

The 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 symbols, such as the fill pattern for the hand shape palms or the tail for the movement arrows.

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 sign will be written from a single perspective.

1.2. Punctuation and Text

Logographic signs are mixed with punctuation to form text.

Punctuation is a single symbol and separates a series of 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.

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 sign 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).

1.3. Block Printing

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!"

With Block Printing, a sign is a cluster of several symbols arranged in 2-dimensions space. Each symbol has a definite appearance and understanding within an established symbol set. The exact form of each symbol is structured, standardized, and highly featural.

Each symbol can have two aspects. The first is the line that defines the positive shape of the symbol. The second aspect is the fill (or negative space) of the symbol that is sometimes used inside the lines for palm facing, and inside some arrow heads and tails. Not every symbol has fill. Fill matters when symbols overlap. The negative space of the symbol on top will cover part of the symbol underneath.

The Block Printing family is aimed at the needs of the reader and the publisher. The Block Printing family has been standardized with a fully developed model.

1.4. Handwriting

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!"

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 loose it's clarity over time as the writer is distanced from the writing. This is common for Shorthand.

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 the various marks should be related to the standardized symbols in

appearance and meaning. Once developed, this style of writing is great for taking notes in a class.

Shorthand is a skill of the proficient writer [SHORTHAND]. In 1982, Sign Language Stenographers could record sign language with SignWriting Shorthand at normal signing speed [STENOGRAPHY]. To develop this skill, practice and special training were required.

The marks of SignWriting Shorthand are a personal style of quick and efficient strokes with a highly developed reception to what signifies meaning. The 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.

2. Formal SignWriting

According to Wikipedia, "In mathematics, computer science, and linguistics, a formal language is a set of strings of symbols that may be constrained by rules that are specific to it." [FORMAL]

Formal SignWriting defines a formal language for the signed languages of the world. Any sign of any sign language can be written as a string of ASCII characters.

Formal 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. The final model has been stable since January 12th, 2012.

The mathematical names of Formal SignWriting are plain text strings of characters. These names are described with regular expressions. Formal languages and regular expressions are used to solve fundamental problems.

Characters	Description	Example
*	Match a literal 0 or more times	ABC* matches AB, ABC, ABCC, ...
+	Match a literal 1 or more times	ABC+ matches ABC, ABCC, ABCCC, ...
?	Match a literal 0 or 1 times	ABC? matches AB or ABC
{#}	Match a literal "#" times	AB{2} matches ABB
[ ]	Match any single literal from a list	[ABC] matches A, B, or C
[-]	Match any single literal in a range	[A-C] matches A, B, or C
( )	Creates a group for matching	A(BC)+ matches ABC, ABCBC, ABCBCBC, ...
(|)	Matches one of several alternatives	(AB|BC|CD) will match AB, BC, or CD

Regular Expression Basics

Table 1

The Formal SignWriting encoding model makes explicit those features which can be effectively and efficiently processed. The mathematical names are 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	Description
A	Sequence Marker
B	SignBox Marker
L	Left Lane Marker
M	Middle Lane Marker
R	Right Lane Marker
w	Writing BaseSymbols
s	Detailed Location BaseSymbols
P	Punctuation BaseSymbols
i	Fill Modifiers
o	Rotation Modifiers
n	Number from 250 to 749

The Tokens of Formal SignWriting

Table 2

These tokens are used in patterns to form written sign language. The following token patterns fully describe SignWriting Text.

2.1. Symbol Keys

Symbol keys can be described with 3 tokens: base symbol, fill modifier, and rotation modifier.

Token Pattern	Description
w	Writing BaseSymbols.
s	Detailed Location BaseSymbols.
P	Punctuation BaseSymbols.
i	Fill Modifiers.
o	Rotation Modifiers.
wio	A writing symbol as 3 tokens of writing base, fill modifier and rotation modifier. Writing symbols can be used in the spatial SignBox or the temporal sequence.
[ws]io	A writing symbol or a detailed location symbol as 3 tokens of base, fill modifier, and rotation modifier. Writing symbols and detail location symbols can be used in the temporal sequence.
Pio	A punctuation symbol as 3 tokens of punctuation base, fill modifier, and rotation modifier. Punctuation symbols divide signs into sentences.

Symbol Key Tokens

Table 3

Symbol keys are 6 characters long. The first character of a symbol key is always "S". The next 3 characters identify the symbol base. The last two characters identify the fill and rotation modifiers respectively.

Regular Expression	Description
S	Start of symbol key
[123][0-9a-f]{2}	Symbol key base
[0-5]	Fill modifier
[0-9a-f]	Rotation modifier
S[123][0-9a-f]{2}[0-5][0-9a-f]	Symbol key definition

Symbol Key Definition

Table 4

Symbol keys can be divided between several types. Each type has a starting and ending base value.

Type	Start	End	Description
all symbols	100	38b	All symbol keys occur in this range.
writing	100	37e	Symbols that can be used in the spatial SignBox or the temporal sequence.
hand	100	204	Various handshapes
movement	205	2f6	Contact symbols, small finger movements, straight arrows, curved arrows and circles.
dynamic	2f7	2fe	Dynamic symbols are used to give the "feeling" or "tempo" to movement.
head	2ff	36c	Symbols for the head and face.
hcenter	2ff	36c	Used to determine the horizontal center of a sign. Same as the head type.
vcenter	2ff	375	Use to determine the vertical center of a sign. Includes the head and trunk types.
trunk	36d	375	Symbols for torso movement, shoulders, and hips.
limb	376	37e	Symbols for limbs and fingers.
location	37f	386	Details location symbols can only be used in the temporal sequence.
punctuation	387	38b	Punctual symbols are used to divide signs into sentences.

Symbol Key Type and Ranges

Table 5

2.2. Coordinates

Coordinates can be described with 2 tokens: number and number. These numbers represent the X and Y coordinates respectively.

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 to specify the spatial relationship between symbols.

Token Patterns	Description
n	Number from 250 to 749
nn	Coordinate with X and Y values as 2 numbers

Coordinate Tokens

Table 6

There are 2 definitions for a coordinate. The more general definition simply defines 3 numbers followed by an "x" followed by 3 more numbers. The more explicit definition correctly restricts the number range from 250 to 749. The general coordinate definition is adequate for processing.

General 3 digit number definition: [0-9]{3}

General coordinate definition: [0-9]{3}x[0-9]{3}

Explicit number definition from 250 to 749:
(2[5-9][0-9]|[3-6][0-9]{2}|7[0-4][0-9])

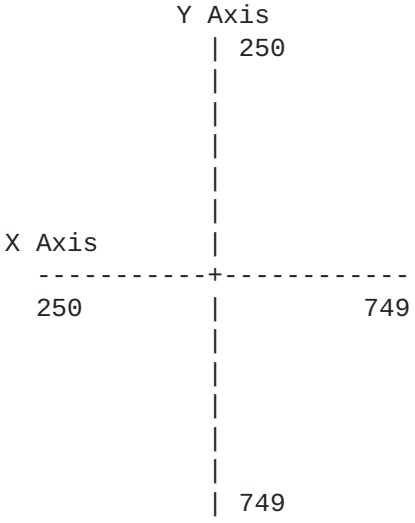
Explicit coordinate definition: (2[5-9][0-9]|[3-6][0-9]{2}|7[0-4][0-9])x(2[5-9][0-9]|[3-6][0-9]{2}|7[0-4][0-9])

2.3. Spatial SignBox

The visual image of a logographic sign is a 2-dimension arrangement of symbols inside of a signbox. Each signbox has a defined width, height, and 2-dimensional center that can be calculated from the plain text.

Each logographic sign exists on its own 2-dimensional signbox. Each point on the signbox is identified with an X and a Y coordinate.

Each signbox has a defined center. Formal numbers range from 250 to 749. Informal number have no limit.



Symbols are placed on the signbox with coordinates that represent the top-left of the symbol image. Symbol images may overlap.

The Spatial SignBox can be described with 8 tokens.

Token Pattern	Description
B	SignBox Marker
L	Left Lane Marker
M	Middle Lane Marker
R	Right Lane Marker
w	Writing BaseSymbols
i	Fill Modifiers
o	Rotation Modifiers
n	Number from 250 to 749
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

Spatial SignBox Tokens

Table 7

The Spatial SignBox is assigned to a lane, has a preprocessed maximum coordinate and zero or more writing symbols with X and Y coordinates.

Symbol key definition: S[123][0-9a-f]{2}[0-5][0-9a-f]

Coordinate definition: [0-9]{3}x[0-9]{3}

SignBox definition: [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})*

2-dimensional space does not have a normative 1-dimensional order. When symbols overlap, the relative order of the overlapping symbols is important. Otherwise, the exact string order of the spatial symbols is unpredictable.

2.4. Temporal Sequence

Signs are written in 2-dimensional space which does not have a normative 1-dimensional order. Any 1-dimensional order of 2-dimensional space is subjective. Some 1-dimensional orders may be canonical according to a particular theory, but there are a variety of theories on setting a 1-dimensional order.

The temporal sequence describes a 1-dimensional order that is separate from the spatial SignBox, rather than ordering the 2-dimensional space directly. The temporal sequence is written as an optional prefix to a spatial SignBox. The temporal sequence will use the same symbols that are used in the spatial SignBox, but it does not need to use all of them and it is not limited to only those symbols. The temporal sequence is a list of writing symbols and/or detailed location symbols that identify temporal order and additional analysis. A valid sequence must contain at least one symbol and can not contain punctuation.

The temporal sequence allows for sorting that is universally supported through binary string comparison.

There are several theories on the best way to structure a temporal sequence. The most productive is based on the SignSpelling Sequence theory of Valerie Sutton. A temporal sequence is structured as a series of starting handshapes followed by optional movements, transitional handshapes, movement, and end handshapes. Only symbols of type "hand" and "movement" should be used in this first section. The last section of the temporal sequence should contain symbols of of type "dynamic", "head", "trunk", and "limb".

Detailed location symbols of type "location" can be used in a temporal sequence, but are rarely (if ever) needed for general writing.

A temporal sequence can be described with 5 tokens.

Token Patterns	Description
A	Sequence Marker
w	Writing BaseSymbols
s	Detailed Location BaseSymbols
i	Fill Modifiers
o	Rotation Modifiers
(A([ws]io)+)?	An optional temporal sequence to be used as a prefix for a SignBox

Temporal Sequence Tokens

Table 8

The temporal prefix starts with a sequence marker and includes an ordered list of writing symbols and detailed locations.

Regular Expression	Description
(A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?	An optional temporal sequence as a sequence marker followed by one or more symbols.

Temporal Sequence Definition

Table 9

2.5. Sentences

General signs are written as a spatial SignBox of symbols in 2-dimensional space. Sortable signs include a temporal sequence as a 1-dimensional prefix to the spatial SignBox.

Signs are mixed with punctuation to form text. Punctuation is a single symbol and separates a series of 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.

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 sign 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).

Regular Expression	Description
Pionn	a punctuation symbol as a punctuation base symbol with a preprocessed minimum coordinate
(((A([ws]io)+)?Bnn(wionn)*)|Pionn)+	a sign text for horizontal writing as a string of signboxes (with optional prefixes) and punctuation
(((A([ws]io)+)?[LMR]nn(wionn)*)|Pionn)+	a sign text for vertical writing as a string of signboxes in lanes (with optional prefixes) and punctuation

Sentence Token Patterns

Table 10

Sentences mix signs with punctuation to form text.

Punctuation definition: S38[7-9ab][0-5][0-9a-f][0-9]{3}x[0-9]{3}

Formal SignWriting text definition: ((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})*

3. Query Language

The query language is a lite ASCII markup similar to Formal SignWriting. Any Formal SignWriting string can easily be converted into several different query string, depending on the search parameters.

The query string is a concise representation for a much larger and detailed set of regular expressions. The regular expressions can be used to quickly and accurately search large files and databases containing Formal SignWriting.

A filter and repeat pattern of searching is used as a series of match criteria. A file, database, or text input is searched using a sequence of steps. Each step applies a single match criteria. Matching results are collated and the next search criteria is applied. The pattern of searching the previous results continues until all regular expressions have been used.

There are two main sections of a query string. The first searches the spatial signbox. The second searches the temporal sequence. Both sections use the same definition for a symbol or a range. The symbol search can match an exact symbol, or a set of related symbols. For the fill and rotation modifiers, the "u" character is a wildcard. The "u" stands for unknown and will match all values rather than a specific character. The range search can match a range of base symbols. The base symbol range consists of 2 values: the starting base symbol and the ending base symbol. Every symbol between these 2 base symbols will be matched.

Symbol Search: S[123][0-9a-f]{2}[0-5u][0-9a-fu]

Range Search: R[123][0-9a-f]{2}t[123][0-9a-f]{2}

The full query string definition allows for the possibility of searching the temporal sequence and the spatial signbox at the same time.

Query String: Q((A(S[123][0-9a-f]{2}[0-5u][0-9a-fu]|R[123][0-9a-f]{2}t[123][0-9a-f]{2})+)?T)?(S[123][0-9a-f]{2}[0-5u][0-9a-fu]([0-9]{3}x[0-9]{3})?|R[123][0-9a-f]{2}t[123][0-9a-f]{2}([0-9]{3}x[0-9]{3})?)*(V[0-9]+)?

3.1. Searching the Spatial Signbox

The spatial signbox is a list of symbols with 2-dimensional placement. The query "Q" will find all signs regardless of the symbols used or their placement.

It is possible to specify one or more symbols (or ranges of symbols) that must be included in the signbox to indicate a match. The order of the symbols is not important. Each symbol (or range) can include an optional coordinate. The coordinate is a restriction on the match, such that a symbol must be used within a certain variance of the coordinate to qualify as a match.

The variance is a number value, 0 or greater with a default value of 20. A variance of 0 will only find symbols used at an exact coordinate. A variance of 5 will match the symbols used at a coordinate, plus or minus 5 for both X and Y numbers.

Symbol Search with Optional Coordinate: S[123][0-9a-f]{2}[0-5u][0-9a-fu]([0-9]{3}x[0-9]{3})?

Range Search with Optional Coordinate: R[123][0-9a-f]{2}t[123][0-9a-f]{2}([0-9]{3}x[0-9]{3})?

Variance: (V[0-9]+)?

Spatial Signbox Search Query: Q(S[123][0-9a-f]{2}[0-5u][0-9a-fu]([0-9]{3}x[0-9]{3})?|R[123][0-9a-f]{2}t[123][0-9a-f]{2}([0-9]{3}x[0-9]{3})?)*(V[0-9]+)?

Query	Description
Q	All signs
QS100uu	Signs with the index handshape in the spatial order
QS100uu480x480	Signs with the index handshape in the spatial order used near coordinate (480,480)
QS100uu480x480V0	Signs with the index handshape in the spatial order used at the exact coordinate (480,480)
QS100uuR2fft36c	Signs with the index handshape and a symbol from the head & face range

Spatial Signbox Query Examples

Table 11

3.2. Searching the Temporal Sequence

The temporal sequence is a list of symbol keys. The query "QT" will find all signs that include a temporal sequence.

It is possible to specify the start of the temporal sequence by identifying a series of symbols and/or ranges. The query will start with an "QA" and end with a "T", such as "QA...T". Between the "QA" and "T", a series of symbol searches and/or range searches will specify the desired start of the temporal sequence. The order of the symbols and ranges is important.

Temporal Sequence Search Query: Q((A(S[123][0-9a-f]{2}[0-5u][0-9a-fu]|R[123][0-9a-f]{2}t[123][0-9a-f]{2})+)?T)?

Query	Description
QT	All signs that include the temporal sequence
QAS100uuT	Signs with a temporal sequence that starts with the index handshape
QAS100uuR100t204S20500T	Signs with a temporal sequence that starts with the index handshape, followed by any handshape, followed by the single contact

Temporal Sequence Query Examples

Table 12

3.3. Transformation to Regular Expression

The conversion from query string to regular expressions has been fully implemented in the SignWriting 2010 JavaScript Library, the SignWriting Server, and the SignWriting Icon Server.

The query language to regular expressions generator uses the following regular expression structures as building blocks.

Temporal Sequence Prefix: (A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)

SignBox Prefix: [BLMR]([0-9]{3}x[0-9]{3})

Spatial Symbols: (S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*

The Temporal Sequence Prefix is a structural marker followed by one or more symbols. For the query string "QT", the prefix is required. For the general "Q", the prefix is optional so "?" is appended to the Temporal Sequence Prefix regular expression.

The SignBox Prefix is a combination of structural marker and preprocessed maximum coordinate. Every constructed regular expression will include the SignBox Prefix.

The Spatial Symbols is zero or more symbol definitions and associated coordinates. The Spatial Symbols regular expression is used for every search. For both "Q" and "QT", it is the only symbol matching used. When searching for specific symbols and ranges, the general

Spatial Symbols definition will sandwich the specific search definitions.

Searching for number ranges with regular expressions requires a unique technique. This technique was described to the LinkedIn Regular Expression Experts at the end of 2011 [DIGIT_SEARCH]. Searching for number ranges in hexadecimal with regular expressions is slightly more complicated but uses the same solution.

4. Styling String

The styling string of Formal SignWriting uses a lite markup to define a variety of styling options. The entire sign can be customized for padding, coloring, and size. Individual symbols within a sign can be customized for coloring and size.

Colors can be written as CSS color names or as color hex values.

CSS Color Names: [a-zA-Z]+

Color Hex Values: [0-9a-fA-F]{3}([0-9a-fA-F]{3})?

The styling string is divided into 2 sections: one for the entire sign and one for individual symbols. The styling string starts with a single dash, after which is the section about the entire sign. A second dash, if present, marks the start of the section about the individual symbols.

Sign section only: -(sign)

Symbol section only: --(symbols)

Both sections: -(sign)-(symbols)

The full styling string definition allows for the possibility of styling the entire sign and individual symbols at the same time. The order of the styling options is important.

Query String: -C?(P[0-9]{2})?(G_([0-9a-fA-F]{3}([0-9a-fA-F]{3})?|[a-zA-Z]+)_)?(D_([0-9a-fA-F]{3}([0-9a-fA-F]{3})?|[a-zA-Z]+)(,([0-9a-fA-F]{3}([0-9a-fA-F]{3})?|[a-zA-Z]+))?_)?(Z([0-9]+(\.[0-9]+)?|x))?(-(D[0-9]{2}_([0-9a-fA-F]{3}([0-9a-fA-F]{3})?|[a-zA-Z]+)(,([0-9a-fA-F]{3}([0-9a-fA-F]{3})?|[a-zA-Z]+))?_)*(Z[0-9]{2},[0-9]+(\.[0-9]+)?(,[0-9]{3}x[0-9]{3})?)*)?

4.1. Styling the Entire Sign

There are several options for styling an entire sign.

C Colorize

P Padding

G Background

D Detail colors

Z Zoom level

4.1.1. Colorize

Colorizing a sign will set the color of each symbol based on its classification.

Hand 0000CC

Movement CC0000

Dynamic FF0099

Head 006600

Body 000000

Detailed Location 884411

Punctuation FF9900

Styling String	Description
-C	Colorize the symbols of the sign

Table 13

4.1.2. Padding

Padding is applied around the entire sign. A two-digit number is used to set the padding.

Styling String	Description
-P01	A padding of 1 around the sign

Table 14

4.1.3. Background

By default, the background of a sign is transparent. The background color can be set with a CSS color name or with a color hex value. The color name or value must be surrounded by underscores.

Styling String	Description
-G_lightblue_	Background color of light blue.
-G_f00_	Background color as 3 hex values.
-G_ff0000_	Background color as 6 hex values.

Table 15

4.1.4. Detail Colors

By default, each symbol has a line color of black and a fill color of white. The line color for all of the symbols can be set with a CSS color name or with a color hex value. The color name or value must be surrounded by underscores. Setting the fill color is optional. To set the fill color, put a comma and the fill color after the line color but before the closing underscore.

Styling String	Description
-D_red_	Line color of red.
-D_red,yellow_	Line color of red with a fill color of yellow.

Table 16

4.1.5. Zoom Level

By default, a sign is set to zoom level 1. The zoom level can be set with an integer or a decimal number.

Alternatively, the zoom level can be set to lower-case 'x', for extendable. The SVG created will not specify the width or height, so that the sign image will fill whatever container it is placed inside.

Styling String	Description
-Z2	Zoom level of 2
-Z15.7	Zoom level of 15.7
-Zx	Zoom level of extendable

Table 17

4.2. Styling Individual Symbols

There are two options for styling individual symbols. Individual symbols are identified by a two-digit number, which identifies the order the symbol appears in the SignBox.

D Detail colors

Z Zoom level

4.2.1. Detail Colors

By default, each symbol has a line color of black and a fill color of white. The line color for an individual symbol can be set with a CSS color name or with a color hex value. The color name or value must be surrounded by underscores. Setting the fill color is optional. To set the fill color, put a comma and the fill color after the line color but before the closing underscore.

Styling String	Description
--D01_red_	First symbol line color of red.
--D01_red,yellow_	First symbol line color of red with a fill color of yellow.
--D01_red_D02_green_	First symbol line color of red and second symbol line color of green.

Table 18

4.2.2. Zoom Level

By default, each symbol is set to zoom level 1. The zoom level of individual symbols can be set with an integer or a decimal number.

Additionally, an offset coordinate can be specified with an individual symbol's zoom level. The offset coordinate of 500x500 is considered no offset for either the x or y value.

Styling String	Description
-Z03,2	Third symbol zoom level of 2
-Z04,15.7	Fourth symbol zoom level of 15.7
-Z04,1.5,480x500	Fourth symbol zoom level of 1.5 with a -20 offset applied to the X value of the symbol's placement coordinate.

Table 19

5. SignWriting 2010

SignWriting 2010 is the modern implementation and international specification of the SignWriting script for the internet community that includes TrueType Fonts and a compact JavaScript library.

5.1. Font Creation Tools

The SignWriting 2010 Tools [SW10_TOOLS] is a collection of text files and python scripts used to build TrueType fonts using FontForge. The source SVG are available in the SignWriting 2010 Fonts project.

5.2. TrueType Fonts

The SignWriting 2010 Fonts [SW10_FONTS] are available for download in SVG format and TrueType fonts. The TrueType fonts can be installed on Windows, Linux, Mac, and iOS. Font installation for Android is not currently supported because the fonts can not be installed due to OS limitations.

Rather than installing the fonts, it is possible to use a font-face statement in CSS to conditionally load the fonts when needed.

```
@font-face {
  font-family: "SignWriting 2010";
  src:
    local('SignWriting 2010'),
    local('SignWriting_2010'),
    url('https://cdn.rawgit.com/Slevinski/signwriting_2010_fonts/\
master/fonts/SignWriting%202010.ttf') format('truetype');
}
@font-face {
  font-family: "SignWriting 2010 Filling";
  src:
    local('SignWriting 2010 Filling'),
    local('SignWriting_2010_Filling'),
    url('https://cdn.rawgit.com/Slevinski/signwriting_2010_fonts/\
master/fonts/SignWriting%202010%20Filling.ttf') format('truetype');
}
```

The fonts have been tailored for the SignWriting 2010 JavaScript library.

5.3. JavaScript Library

The SignWriting 2010 JavaScript Library [SW10_JS] provides support for SignWriting images and queries. It leverages the TrueType fonts without any additional requirements. The SignWriting 2010 JavaScript library is contained in a single file and can be included in any HTML page or JavaScript environment. The library includes a guide, API documentation, and a testing suite.

5.4. SignMaker Editor

SignMaker [SIGNMAKER] is a standards based editor, utilizing HTML, CSS, JavaScript, SVG, TrueType Fonts, and PNG images.

SignMaker is browser based without the need for a server connection. It can be used online or it can be downloaded [SM_DOWNLOAD] and run directly from the user's computer.

The primary online website on SignBank [SM_SIGNBANK] can be used to create a private dictionary in the browser's LocalStorage or view dozens of sign language dictionaries from around the world.

The secondary online website on GitHub.io [SM_GITHUB] can be used to create a private dictionary in the browser's LocalStorage.

6. Unicode Integration

SignWriting Text is integrated with Unicode in a varieties of ways. The TrueType fonts of SignWriting 2010 are compatible with each of the varieties.

6.1. UTF-8

Formal SignWriting is based on ASCII, so it can be supported anywhere ASCII is supported. ASCII is a subset of UTF-8, with a one-to-one character correspondence. Anywhere UTF-8 is supported, the size of the Formal SignWriting strings is equal to the ASCII encoding of 8 bits per character. With UTF-32, the size of the string is 4 times that of the ASCII encoding, requiring 32 bits for each character.

The Formal SignWriting strings have an equivalent encoding on plane 15 of the Private Use Area.

6.2. Private Use Area Plane 15

The x-Character-SignWriting coded character set is isomorphic with Formal SignWriting strings. Instead of ASCII characters, x-Character-SignWriting uses code points on Plane 15 of Unicode. Symbols are defined using 3 characters. Structural markers are defined using 1 character each. Number characters are defined using 1 character each.

Description	Formal SignWriting	x-Character-SignWriting
Sequence Marker	A	U+FD800
SignBox Markers	B, L, M, R	U+FD801 to U+FD804
Fill Modifiers	0 to 5	U+FD810 to U+FD815
Rotation Modifiers	0 to 9 and a to f	U+FD820 to U+FD82F
Symbol Base	S100 to S38b	U+FD830 to U+FDABB
Numbers	250 to 749	U+FDE06 to U+FDFF9

x-Character-SignWriting Definition

Table 20

6.3. Private Use Area Plane 16

The x-ISWA-2010 is a 16-bit coded character set that has a unique codepoint for each symbol of the ISWA 2010. The coded character set can be mapped to plane 16 by adding value U+100000 to any codepoint.

A simple formula transforms a symbol key into a codepoint. Given a symbol key as variable "key", in JavaScript the function is defined as:

```
var code = ((parseInt(key.slice(1,4),16) - 256) * 96) +
((parseInt(key.slice(4,5),16))*16) + parseInt(key.slice(5,6),16) +
1;
```

6.4. Unicode 8

The symbols of the International SignWriting Alphabet 2010 have been approved for Unicode 8 [UNICODE8]. Every symbol of the ISWA 2010 can be described with 1 to 3 characters.

Due to the variable size of the symbol description, both sorting and searching have been compromised. These issues, and others, have been reported to the Unicode Consortium. During UTC #144 [UTC_144], SignWriting was discussed at length, but the issues are unresolved [UNICODE8_ISSUES].

A special design document [SW_DESIGN] was prepared for UTC #144 action item 144-A33 that shows three SignWriting examples and describes the three different representations for each example.

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 two name spaces for the Center for Sutton Movement Writing that require definition and extension: x-ISWA-2010 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.

9. References

[CODING_SYSTEM]
Butler, C. and R. Channon, "Transcription systems as input to coding systems: SignWriting & SignTyp", <http://www.signwriting.org/archive/docs7/sw0623_TISLR_2010_SignWriting_SignTyp_Poster.pdf>.

[DELEGS] "DELEGS Editor", <http://www.delegs.com/DelegsPage>.

[DIGIT_SEARCH]
Slevinski, S., "3 Digit Number Searching", <https://www.linkedin.com/grp/post/1066587-85595980>.

[FEATURES]
Hulst, H. and R. Channon, "Why dynamic features?", <http://www.purdue.edu/tislr10/pdfs/van%20der%20Hulst%20Channon.pdf>.

[FORMAL] "Formal Language on Wikipedia", <https://en.wikipedia.org/wiki/Formal_language>.

[ISWA_FONT]
Slevinski, S., "International SignWriting Alphabet 2010 Font Reference", <http://signpuddle.net/iswa>.

[ISWA_REF]
Slevinski, S., "International SignWriting Alphabet 2010 HTML Reference", <http://signbank.org/iswa>.

[JSPAD] "JSPad", <http://www.mat.info.gifu-u.ac.jp/jspad/>.

[MSW] Slevinski, S., "Modern SignWriting", <https://github.com/Slevinski/msw>.

[NOTATION]
Hulst, H. and R. Channon, "Notation Systems", <http://homepage.uconn.edu/~hdv02001/Articles-pdfs/131%20-%20Notation%20Systems.pdf>.

[SHORTHAND]
Sutton, V., "SignWriting Shorthand", <http://www.signwriting.org/lessons/cursive/shorthand>.

[SIGNMAKER]
Slevinski, S., "SignMaker Editor", <https://github.com/Slevinski/signmaker>.

[SM_DOWNLOAD]
Slevinski, S., "SignMaker Download Link", <https://github.com/Slevinski/signmaker/archive/gh-pages.zip>.

[SM_GITHUB]
"SignMaker on GitHub.io", <http://slevinski.github.io/signmaker>.

[SM_SIGNBANK]
"SignMaker on SignBank", <http://signbank.org/signmaker.html>.

[SP_ONLINE]
"SignPuddle Online", <http://signpuddle.org>.

[SPML] "SignPuddle Markup Language Source Files",
<http://signbank.org/signpuddle2.0/data/spml>.

[STENOGRAPHY]
Sutton, V., "Sign Language Stenography",
<http://www.signwriting.org/lessons/cursive/byhand5.html>.

[STUDIO] "SignWriter Studio", <http://signwriterstudio.com>.

[SW10_FONTS]
Slevinski, S., "SignWriting 2010 Fonts",
<https://github.com/Slevinski/signwriting_2010_fonts>.

[SW10_JS] Slevinski, S., "SignWriting 2010 JavaScript Library",
<https://github.com/Slevinski/sw10js>.

[SW10_TOOLS]
Slevinski, S., "SignWriting 2010 Tools",
<https://github.com/Slevinski/signwriting_2010_tools>.

[SW_DESIGN]
Slevinski, S., Anderson, D., and K. Whistler, "SignWriting
Design, With Three Examples and Their Representation",
<http://www.unicode.org/L2/
L2015/15219-signwriting-design.pdf>.

[SWIFT] "SignWriting improved fast transcriber",
<http://www.researchgate.net/publication/230720646_SWift_a
_SignWriting_improved_fast_transcriber>.

[SWIS] Slevinski, S., "SignWriting Icon Server source",
<https://github.com/Slevinski/swis>.

[SWIS_LABS]
"SignWriting Icon Server on Wikimedia Labs",
<http://swis.wmflabs.org>.

[SWSERVER]
Slevinski, S., "SignWriting Server source",
<https://github.com/Slevinski/swserver>.

[SWSERVER_LABS]
"SignWriting Server on Wikimedia Labs",
<https://swserver.wmflabs.org>.

[SWSERVER_SIGNPUDDLE]
"SignWriting Server on SignPuddle.net",
<https://signpuddle.net>.

[UNICODE8]
"Unicode 8", <http://unicode.org/versions/Unicode8.0.0/>.

[UNICODE8_ISSUES]
Slevinski, S., "Issues with SignWriting in Unicode 8",
<http://www.slideshare.net/StephenSlevinski/sign-writing-in-unicode-8-issues>.

[UTC_144] "Minutes of UTC Meeting 144",
<http://www.unicode.org/L2/L2015/15187.htm>.

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 [MSW]. Modern SignWriting has example text and concretely defines the processes available. It fully documented the text encoding with regular expressions.

The Formal SignWriting strings are exactly the same as they appear in the Modern SignWriting document. The query language is nearly the same, with a compatible improvement for searching the temporal sequence.

Appendix B. International SignWriting Alphabet 2010

The International SignWriting Alphabet 2010 [ISWA_REF] is a collection of visually iconic symbols that exists in a layered hierarchy (Appendix B.3). The ISWA 2010 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 [ISWA_FONT] have been stable since their initial release on October 20th, 2010.

Valerie Sutton

- hand crafted and organized 30K plus individual glyphs
- created a 2 dimension PNG of 3 colors for each
- named each individual glyph with 6 degrees of significance

- font name: ISWA 2010 Sutton

Steve Slevinski

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

Adam Frost

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

B.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 represents 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.

B.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-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 21

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 22

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 23

B.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.

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.

The 7 Categories of the ISWA 2010

Table 24

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.

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

The 30 groups with symbol ID segment.

Table 25

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.

B.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.

The combined character sequence is used in Formal SignWriting and the x-Character-SignWriting coded character set.

B.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 ^ (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 26

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 27

Each base has a defined validity set for fills.

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	2^0	1
2	2^1	2
3	2^2	4
4	2^3	8
5	2^4	16
6	2^5	32
7	2^6	64
8	2^7	128
9	2^8	256
10	2^9	512
11	2^10	1024
12	2^11	2048
13	2^12	4096
14	2^13	8192
15	2^14	16384
16	2^15	32768

Table 28

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.

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 29

Appendix C. SignPuddle Standard

The SignPuddle Standard for SignWriting text has been stable since January 12th, 2012.

C.1. Licenses

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

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

The previous generation of open source projects are licensed under the GPL 2 for MediaWiki and GPL 3 for the general software on Github.

The current generation of open source projects on GitHub are licensed under the MIT License.

C.2. Infrastructure

C.2.1. SignPuddle Online

SignPuddle Online [SP_ONLINE] 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 [SPML]. Dozens of sign languages from around the world are represented. Each language can have several collections of SignWriting.

C.2.2. SignWriting Server

The SignWriting Server is the next generation SignWriting server code for SVG images and JSON data. The SVG creation is stable and fully supports the styling string.

The API is documented with API Blueprint and contains a concise guide and extensive examples. The API is still in the initial stages of development, but it will provide user authenticated access to read and edit SignPuddle Online data in the near future.

The SignWriting Server is available on SignPuddle.net [SWSERVER_SIGNPUDDLE] for all SignWriting projects.

A mirror server is available on Wikimedia Labs [SWSERVER_LABS] for Wikimedia projects.

Additional SignWriting Servers can be created directly from the GitHub source [SWSERVER].

C.2.3. SignWriting Icon Server

The SignWriting Icon Server is the previous generation SignWriting server code for SVG, PNG, and other image formats. The image creation is stable and fully implemented. The API is a proof of concept with only an initial level of support.

The main server is available on Wikimedia Labs [SWIS_LABS] for all SignWriting projects.

Each SignWriting Icon Server provides the SignWriting Thin Viewer as a site script and as a bookmark.

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

C.2.4. Wikimedia Incubator

The SignWriting Script has been enabled on Wikimedia Incubator using the SignWriting Gadget.

C.3. Compatibility

SignTyp, SignWriter Studio, the DELEGS Editor, SWift, and more.

C.3.1. SignTyp

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

> Notation Systems by Harry van der Hulst and Rachel Channon. [NOTATION]
>
> Why dynamic features? by Harry van der Hulst and Rachel Channon. [FEATURES]
>
> Transcription systems as input to coding systems: SignWriting & SignTyp by Charles Butler and Rachel Channon. [CODING_SYSTEM]

C.3.2. SignWriter Studio

SignWriter Studio [STUDIO] 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 chooser 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 chooser do not work well and gives a grouped list of symbols for the base group.

C.3.3. DELEGS Online

The DELEGS Editor [DELEGS] 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.

C.3.4. SWift

SWift is a SignWriting improved fast transcriber [SWIFT] from Claudia Savina Bianchini, Fabrizio Borgia, and Maria De Marsico. SWift is under active development. The design "guides and simplifies the editing process".

SWift uses an alternate symbol hierarchy than the ISWA 2010. A conversion library is planned in the future to support Formal SignWriting strings.

C.3.5. JSPad

JSPad [JSPAD] is Windows and Mac OS X software for editing the words and sentences of sign language, created by the Matsumoto Laboratory of Gifu University in Japan.

Author's Address

Stephen E Slevinski Jr
SignPuddle

Email: slevin@signpuddle.net