



Inscribe: Deployment and User Guide

Last Saved: 6/3/08 7:33 PM

Inscribe Version: 1.0.0

1	OVERVIEW	2
2	DEPLOYING INSCRIBE	3
3	USER'S GUIDE	4
3.1	ENVIRONMENT VARIABLES	4
3.2	SPECIFYING COMMAND LINE OPTIONS.....	4
3.3	INSCRIBE.PY – BUILDING ARCHETYPES AND GENOMES.....	4

1 Overview

Inscribe is a collection of scripts, HTML pages, and an Adobe Flash application for building Han archetype files and genome data files for use with *Stylus*. Pre-built files are available for a small fraction of characters. *Inscribe* enables users to experiment with other characters.

2 Deploying Inscribe

Using *Inscribe* requires deploying the supplied package as a locally hosted web site. The package includes a configuration file (apache.conf) for use with the Apache 1.0 or Apache 2.0 web server. Hosting *Inscribe* with another web server will require similar configuration.

The default arrangement of *Inscribe* creates three distinct web sites: One for hosting Han description text files and Archetypes, another for hosting Genes, and a third that hosts the core *Inscribe* web site.

The *Inscribe* web pages require that the Arial Unicode font (see http://en.wikipedia.org/wiki/Arial_Unicode_MS for a description and http://www.ascendercorp.com/msfonts/arial_unicode.html for purchase).

3 User's Guide

Values in **bold** must appear exactly as specified; non-**bold** items represent user-supplied values.

3.1 Environment Variables

Inscribe takes default values from a number of environment variables, listed in the table below, to reduce the number of required command line options. Each has a command line equivalent which, if specified, overrides their value.

Variable	Definition
STYLUS_INSCRIBEARGS	Default arguments
STYLUS_INSCRIBEOUT	Output path for Genes
STYLUS_HANURL	URL for Han archetype files
STYLUS_AUTHOR	Author name string

3.2 Specifying Command Line Options

All command line options have two forms: A long form using two dashes with the full option name (such as, `--quiet`) and a short form consisting of a single dash followed by the first letter of the long name (such as, `-q`). Some take an additional value following the option and separated from it by one or more spaces (such as, `--definition ./Archetypes/`).

3.3 *inscribe.py* – Building Archetypes and Genomes

Stylus requires input of two forms: Han archetype files and genome data files. *inscribe.py* is the script used to create archetype files, and it also serves as a starting point for making genome data files (*Stylus* also writes these files, but requires an initial one to start each experiment). Most often, the *Inscribe* web site will invoke the *inscribe.py* script, though researchers may also use it directly.

Inscribe uses an interactive Flash application to guide the user through several manual steps that specify archetype geometry, stroke grouping, and connectivity. Based on user inputs, *Inscribe* generates two archetype related files. Archetype-point files (carrying the `.hcf` extension) are readable text files recording the information entered through the Flash application. Archetype files (carrying the `.han` extension) are generated by converting the archetype-point files into a form containing all the pre-computed data *Stylus* requires at runtime.

inscribe.py also can create basic genes from an archetype file. These genes are not meant to be the starting point for realistic experiments – they are too regular – but a starting point from which to evolve a realistic gene.

3.3.1 Options

Option	Description
<code>--code</code> Han-unicode-number	Specify the Han character by Unicode number
<code>--definition</code> archetype-output-path	Create archetype and archetype-point files
<code>--gene</code> gene-options default	Create a gene

--output gene-output-path	Output path for created genes
--urls Han	Specify the URL for obtaining archetype files
--author	Specify the author name
--quiet	Run silently
--help	Display help content

`inscribe.py` expects to find all Han-related files – Archetypes and character descriptions – at the provided URL. It writes genes to the specified output path. The `gene` option may appear multiple times, each instance describing a different gene to create. Specifying `default` requests a full-size gene with Han stroke ordering.

`inscribe.py`, when reading or writing Han text files, Archetypes, and Genes, assumes the files are kept in numeric directories with names derived from the thousands value of the associated Han Unicode value. For example, `inscribe.py` will write genes for 4E03 to the 4000 subdirectory of the specified output path.

3.3.2 Describing a Gene

`inscribe.py` can create genome data files encoding vector proteins with independently scaled components described by the options below (colons – : – join multiple options):

Option	Description
o han or o min or o #(r),#(r),...	Specify stroke tracing order
h (+ -)#,(+ -)#	Add an incoherent region to the head
t (+ -)#,(+ -)#	Add an incoherent region to the tail
s x(+ -)#(%), s y(+ -)#(%), d x(+ -)#, d y(+ -)#	Specify protein-level scaling
g #(s x(+ -)#(%), s y(+ -)#(%), d x(+ -)#, d y(+ -)#)	Specify scaling for a stroke group #
s #(s x(+ -)#(%), s y(+ -)#(%), d x(+ -)#, d y(+ -)#)	Specify scaling for stroke #

3.3.2.1 Gene Names

`inscribe.py` derives gene names from the option string supplied, after translating characters into forms more fitting for filenames. Generally, `inscribe.py` translates colons and periods into underscores; it substitutes the string `pct` for the percent sign and then appends the result, using two dashes, to the Unicode number. `inscribe.py` reports the name used upon completion. For example, this command:¹

```
[Biologic:Stylus] $ ./inscribe.py -c 52DC -g "o12,10,3r:sx75%:g2(dx-50)" -u /Han/ -o .
```

will save the following gene named:

```
./5000/52DC--o12_10_3r_1_2_4_5_6_7_8_9_11-sx75_00pct-g2_dx-50_00.gene
```

¹ The quotes – " – are necessary to hide the parentheses from the BASH shell.

3.3.2.2 Controlling Stroke Order

Users specify a stroke order when building an archetype (typically the standard written order). However, genes need not conform to that order. The `o` option controls the order `inscribe.py` follows when tracing the strokes to build a gene. Specifying `ohan` causes strokes to be ordered according to the archetype specification; `omin` uses an algorithm that attempts to minimize the total length of moves. Other orderings are given by specifying the stroke numbers, optionally followed by an `r` (indicating to trace the stroke in the reverse direction); `inscribe.py` will supply any strokes not explicitly specified in the order they appear in the Archetype. If no ordering is requested, `inscribe.py` uses the ordering specified by the archetype.

3.3.2.3 Extending a Gene

All genes begin at the origin with the first stroke and end wherever the last stroke terminates. Extra, incoherent regions may be appended to the beginning and end of the gene through the `h` and `t` options. These options define an (x,y) point value either before (for `h`) or after (for `t`) from or to which `inscribe.py` will add incoherent vectors.

3.3.2.4 Shaping a Gene, Group, or Stroke

`inscribe.py` allows the scaling and displacement of the entire gene, one or more groups, and one or more strokes with overlapping values combined. Scaling may be either an absolute value or a percentage; displacements are always absolute values.

3.3.2.5 Examples

The following command creates a new archetype (based on user inputs from the Flash application) and a gene corresponding to U+4E28. It shrinks the entire gene along the x-axis to 50% of its original size and shifts the first group along the y-axis by 10 units and accepts `inscribe.py` defined stroke ordering:

```
[Biologic:Stylus] $./inscribe.py -c 4E28 -d /Han/ -g "sx50%:g1(dy+10)" -u /Han/ -o /Han/
```

Similarly, the following creates two new genes. The first orders the first three strokes and shrinks the third stroke by a factor of 0.8 along both dimensions. The second uses default stroke ordering and displaces the second group by 25 units long both the x and y axis':

```
[Biologic:Stylus] $./inscribe.py -c 4E28 -g "o2r,3,1r:s3(sx0.8,sy0.8)" -g "ohan:g2(dx25,dy25)" -u /Han/ -o /Genes/
```