Project GeoSim: **Sense of Place**
Technical Manual, Version 0.1

Sense of Place uses several text files for configuration and data. This manual describes those files and their formats. A *Sense of Place* user may change the configuration and/or data files.

1 *Sense of Place* Configuration File: `config.gcf`

The file `config.gcf` must be in the same directory (or folder on Macintosh) as the *Sense of Place* executable file (`snsplace.exe` on MS-DOS, Sense of Place on Macintosh and `snsplace` on Unix). This configuration file is a plain ASCII text file that may be edited with any text editor software.

`config.gcf` is a **Project GeoSim** database file. The format of an entry in a **Project GeoSim** database file is:

parameter= *value*

where *parameter* is a string that is the name of the parameter, such as *database*, and *value* is a string that represents the value that the parameter is assigned. Values include file names, path names, numbers and Boolean values (true or false). This is an example of an entry in the file:

```
database= SP_config
```

Any line in the file that begins with a pound sign (`#`) is a comment that will be ignored when *Sense of Place* reads the file. If you want to temporarily "remove" a line, simply place a pound sign in front of it. When you want to restore the line, remove the pound sign. The following are the valid parameter names that *Sense of Place* uses. Each description is followed by the entry that `config.gcf` originally contained when first installed.

DataBase is the name of this **Project GeoSim** database. It must be the first entry in the file and its value must always be `SP_config`.

```
DataBase= SP_config
```

DataPath is the path to the country/region datafiles. This path is relative to the directory in which the executable program (e.g. `snsplace.exe`) resides. Directory or folder names in the path must be separated by a *forward slash*: `/`. Do not use a back slash (\) or colon (:) to separate names in the path.

```
DataPath= dbase/
```

InfoPath is the path to the help files. This path is relative to the directory in which the executable program (e.g. `snsplace.exe`) resides. Directory or folder names in the path must be separated by a *forward slash*: `/`. Do not use a back slash (\) or colon (:) to separate names in the path.

```
InfoPath= info/
```

InterfaceFile is the name of the file that contains the user interface information, such as window and button names and positions.

```
InterfaceFile= snsplace.inf
```

StMapFile is the name of the state map image file. The state map appears in the lower left hand corner of the screen when the program begins. The value for this parameter should not be changed.

```
StMapFile= stmap.rlc
```

CntyMapFile is the name of the county map image file. The county map appears in the lower right hand corner of the screen when the program begins. The value for this parameter should not be changed.

```
CntyMapFile= cntymap.rl0
```

BStore is a Boolean parameter. If BStore = true, then *Sense of Place* will use backing store to save and restore the screen when windows pop up and disappear. If BStore = false, then *Sense of Place* will redraw the screen when a pop-up window is closed. Using backing store (i.e., BStore = true) is faster, but requires more RAM. If you are short on RAM, set BStore to false.

```
BStore= true
```

ConfirmQuit is a Boolean parameter. If ConfirmQuit = true, then the user will always be asked to confirm a request to quit the program. If ConfirmQuit = false, then the program will terminate immediately whenever the user selects "Quit" from the "File" menu.

```
ConfirmQuit= true
```

YourState and YourCounty are the names of your home state and county, respectively. The maps feature the county and state named with these parameters when the program begins. We have set these parameters to the home county and state of Virginia Tech. You should reset them to your own state and county.

```
YourState= VA
YourCounty= Montgomery
```

IntroHelpDesired is a boolean parameter. If IntroHelpDesired = true, then a brief help screen is shown whenever the program begins. If IntroHelpDesired = false, the help screen is bypassed.

```
IntroHelpDesired= false
```

MapBoxLineWidth is the width frame of the movable map box on the state map. This portion of the state map inside this box is shown (at greater magnification) in the county map. The legal values for this parameter are `thick` and `thin`.

```
MapBoxLineWidth= thin
```

CntyMapMagnify is the default magnification of the county map when the program begins. For example, if CntyMapMagnify = 3, then the county map is drawn at 3 times its actual size. The legal values for this parameter are integers in the range 3-8.

```
CntyMapMagnify= 3
```

NDataSections is the number of database sections currently in use. This number must represent the number of DataSection (see below) entries in the configuration file.

```
NDataSections= 23
```

DataSection is the name of a section of the database. There is a DataSection declaration for each section of the database. The name of a section indicates what type of variable it contains. You can omit a section by removing its declaration from the configuration file. (Remember that you can temporarily remove a declaration by putting a pound sign in front of it.) Remember to change the NDataSections value if you remove a data section. You may notice that some DataSection declarations have some additional text enclosed in "pipes" (|). Section 2.2 explains how this extra text is used.

```
DataSection= age
DataSection= ancestry
DataSection= banks
DataSection= bpermits |Building Permits|
DataSection= business
DataSection= crime
DataSection= earnings
DataSection= educ |Education|
DataSection= farms
DataSection= health
DataSection= househld |Households|
DataSection= housing
DataSection= income
DataSection= laborfc |Labor Force|
DataSection= localgov |Local Government|
DataSection= physical
DataSection= pop |Population|
DataSection= poverty
DataSection= retail
DataSection= service
DataSection= unemploy |Unemployment|
DataSection= vitals
DataSection= votes
```

2 Data Files

Each database section specified in `config.gcf` has a pair of files associated with it: a dictionary file, which contains information about the section's variables, and a data file, which contains the actual data. Both files of a pair must share the same prefix, which is usually a name that describes the general subject for the section, such as "ancestry", or "banks". The suffix for dictionary files is `.spd`, which stands for "**S**ense of **P**lace **d**ictionary file". The suffix for data files is `.spb`, which stands for "**S**ense of **P**lace **b**inary data file".

These files must be in the directory path specified by the `DataPath` entry in `config.gcf`, which is relative to the directory or folder that contains the executable program (usually /snsplace). Upon installation, the directory path is `dbase/`. Typically then, the full path and file name for the banks data file, for instance, is `/snsplace/dbase/banks.spb`.

2.1 Excluding a Variable from a Database Section

A variable can be "removed" from a database section by editing its dictionary file. "Removing" a variable means excluding it from the selection list in the program, so that students will never see it. Each dictionary file has two columns — labeled "CntyUse" and "StUse", respectively — which tell *Sense of Place* whether or not to include variables. A "1" in the "CntyUse" column signals *Sense of Place* to include a variable, while a "0" indicates that the variable should be excluded. The same is true for values in the "StUse" column. For example, here are the steps to take to remove "Bank deposits per capita" from the county variable list:

1. Find the file `banks.spd`.
2. Find the "LongLabel" column.
3. Find the row with "Bank deposits per capita" in the "LongLabel" column.
4. Change the "1" in this row and the"CntyUse" column to "0".

2.2 Creating Your Own Database Sections

You can create your own sections for the *Sense of Place* database. To do so, you need to create a dictionary file and data file for the section. *Sense of Place* data files are stored in binary — rather than text — format. Binary format has several advantages over text format: it makes the data files smaller, it allows the program to read them faster, and it makes the program itself smaller.

The main disadvantage of binary files is that you cannot use a text editor to create or read them. To remedy this problem, we have created utility programs that translate binary files into text format, and text files back into binary format. Thus, you can translate one of our data files into text format, and use it as a model for creating your own (with a text editor). Then you can translate your text file into binary format, so that *Sense of Place* can read it. The name of the program that translates binary files into text format is `spb2spa`. The program that translates text files into binary format is called `spa2spb`. The names of these programs are derived from the suffixes for binary files (`spb`) and text files (`spa`, which stands for "**S**ense of **P**lace **a**scii data file".) The dictionary files are text files, so you can create and read them with a text editor.

Here is a summary of the procedure for creating your own database sections:

1. Collect the data.
2. Create a dictionary file for the section. Use one of our dictionary files as a model.

3. Create an ascii text version of your data file. Use `spb2spa` to obtain an ascii text version of one of our files to use as a model.

4. Translate your text file into binary format with `spa2spb`. (Note that the dictionary file must be present in the current directory for `spa2spb` to work.)

5. Place the dictionary and data files in the database directory with our files.

6. Create a DataSection entry for your section in `config.gcf`. Use the prefix for the two files as the value for the entry. If the prefix is too short for *Sense of Place* to use in its selection list, you can put a longer name after the entry. For example:

   ```
   DataSection= bpermits |Building Permits|
   ```

   The name must be enclosed in "pipes" (`|`).