Sergio J. Rey
Mark V. Janikas

ProjectMaker for Space-Time Analysis of Regional Systems

How to Create STARS Projects with ProjectMaker

Version 0.8.2

Released 2006-01-13

Contents

List of Figures

Chapter 1

Building STARS Projects

1.1 Creating and Managing STARS Projects

1.1.1 Overview of Changes

The **Project Maker** utility has gone through a major overhaul since the last release. A series of python scripts were created to surplant the *ShapeLib* utility that was formerly required to manage mapping files. In short, this means that the program is entirely independent of any additional binaries. This occured behind the scenes, so for many users this change will go unnoticed.

After the first release it became evident that most of the problems associated with our program had to do with confusion surrounding the creation of STARS projects. As such, we drastically improved the capabilities of **Project Maker** and made it easier for the common user to employ.

The following sections serve as a tutorial for creating STARS projects. We use a real data set that comes standard in this release. Please feel free to walk through the steps both in this document and in the **Project Maker** environment.

1.2 Project Maker: Tutorial

We believe the easiest way to instruct the user on creating a new STARS project is to provide a step-by-step tutorial with "real" data. The example contains data for the counties of California. The files are located in the "data" directory that is included in the STARS binary. Upon installation you should have copied "data" to a directory that has write permissions.[1] For this example we will assume you copied "data" to a directory named "myDataDir". So the path to the data is: "myDataDir/data".

In this example we want to create a STARS project from an ArcView shapefile that contains several cross-sectional variables. Furthermore, we have additional data on per capita incomes and population from 1969–2001 that we want to augment our new project with. The following sections will guide you through this procedure.

[1] See the README file included in the STARS download.

1.2.1 Launch Project Maker

To launch **Project Maker** double click on the "ProjectMaker" application.[2] You will encounter the main screen for **Project Maker** which contains four main menus: **File, Data, Tables, Plot**.

1.2.2 Create a New Project: California Example

1. Click **File-Create New STARS Project**

   a You will be prompted to choose which type of file is going to be used as the *base data* for the new project. Your two choices are **ArcView** and **CSV**. If your project has a map, you must choose the **ArcView** option. You can always add data from comma delimited files later in the project making process. The steps that follow are identical for the CSV option, but the plotting feature will be nullified. For this example choose: **ArcView**.

   b You will then be prompted to choose a file that contains your base data. Base data for **ArcView** projects are found in ***.dbf** files. Navigate to the "myDataDir/data" data directory and choose the file: "california.dbf". Click "OK".

   c This is your first chance to name your new project. You will encounter an empty entry field. If you leave this entry blank, the prefix of the base data file will be the name of your project (i.e. california in this case). For this example we are going to name the project CA. So type "CA" in entry field for Project Prefix. Click "OK".

   d Now you will have to declare the time series information for your project. First you will be prompted for the *type* of series. The California example we are working with contains annual data as shown in Figure 1.2, so check the "Annual" button and click "OK". See Section 1.3.4 for a full explanation of the different time series types available for STARS.

   e Next, you must provide the start year and the end year for your time series. So place 1969 and 2001 in their corresponding entry spaces as shown in Figure 1.3, and click "OK".

   f You will now encounter a dialogue box (Figure 1.4) where you must choose a field from your original data to use as the names for your cross-sectional units. There are four empty fields in this view. In the case of California, you only need to select "NAME" for the "Unique Field" and click "OK". This will associate any shape in your shapefile with its corresponding name. Futher information on this procedure can be found in section 1.3.5.

   g The next step involves selecting what type (if any) weights matrices you would like to create based on the shape files. Select the type you want as shown in Figure 1.5 and click "OK".

   h You have just finished the initial stage of creating a STARS Project. Now it is time to add some data!

[2] For those of you running STARS from the source code you should navigate to the "stars/src" directory and enter "python ProjectMaker.py" at the command line.

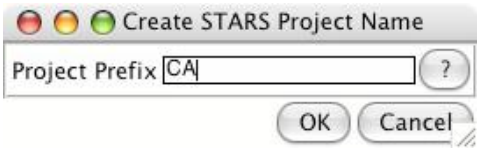


Figure 1.1: Name your Project

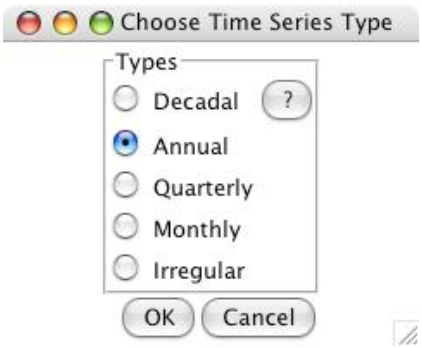


Figure 1.2: Choose the Time-Series Type

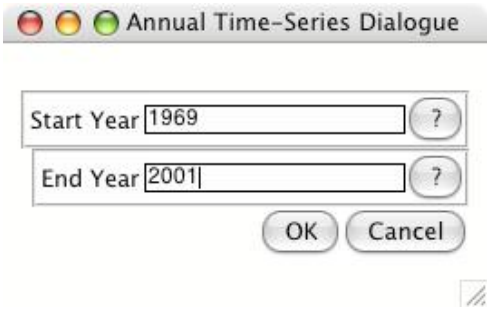


Figure 1.3: Enter the Start and End Periods

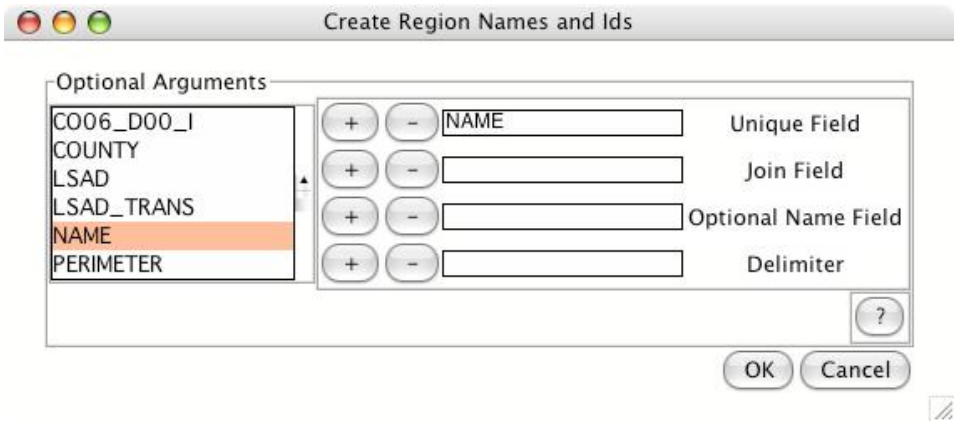


Figure 1.4: Choose the Cross-Sectional Names Field (California)

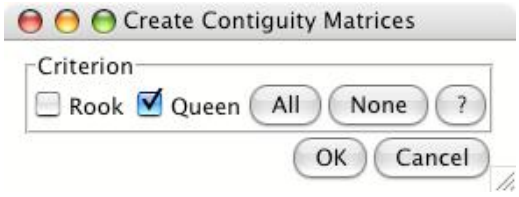
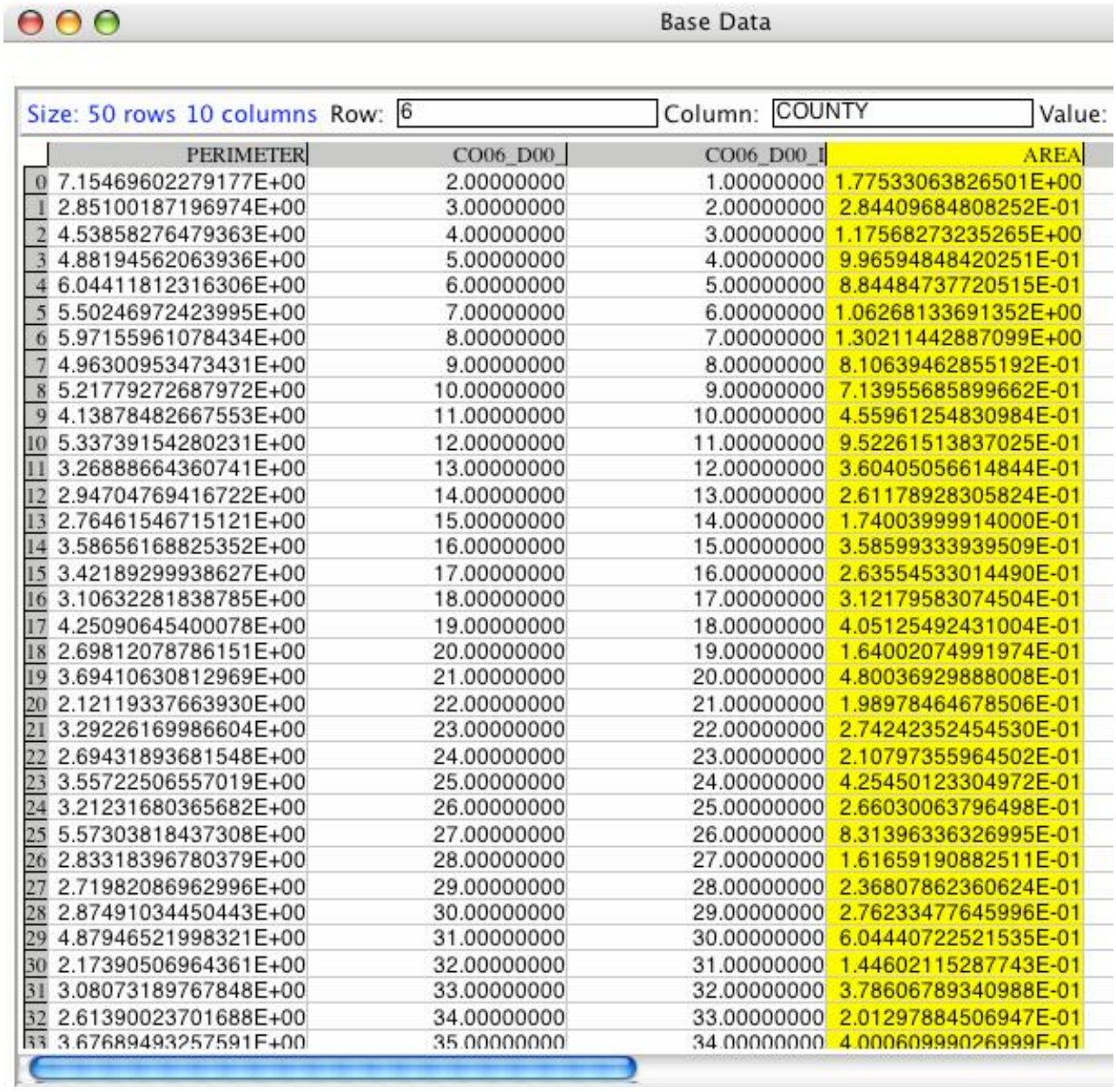


Figure 1.5: Choose the Types of Weights Matrices



	PERIMETER	CO06_D00_	CO06_D00_I	AREA
0	7.15469602279177E+00	2.00000000	1.00000000	1.77533063826501E+00
1	2.85100187196974E+00	3.00000000	2.00000000	2.84409684808252E-01
2	4.53858276479363E+00	4.00000000	3.00000000	1.17568273235265E+00
3	4.88194562063936E+00	5.00000000	4.00000000	9.96594848420251E-01
4	6.04411812316306E+00	6.00000000	5.00000000	8.84484737720515E-01
5	5.50246972423995E+00	7.00000000	6.00000000	1.06268133691352E+00
6	5.97155961078434E+00	8.00000000	7.00000000	1.30211442887099E+00
7	4.96300953473431E+00	9.00000000	8.00000000	8.10639462855192E-01
8	5.21779272687972E+00	10.00000000	9.00000000	7.13955685899662E-01
9	4.13878482667553E+00	11.00000000	10.00000000	4.55961254830984E-01
10	5.33739154280231E+00	12.00000000	11.00000000	9.52261513837025E-01
11	3.26888664360741E+00	13.00000000	12.00000000	3.60405056614844E-01
12	2.94704769416722E+00	14.00000000	13.00000000	2.61178928305824E-01
13	2.76461546715121E+00	15.00000000	14.00000000	1.74003999914000E-01
14	3.58656168825352E+00	16.00000000	15.00000000	3.58599333939509E-01
15	3.42189299938627E+00	17.00000000	16.00000000	2.63554533014490E-01
16	3.10632281838785E+00	18.00000000	17.00000000	3.12179583074504E-01
17	4.25090645400078E+00	19.00000000	18.00000000	4.05125492431004E-01
18	2.69812078786151E+00	20.00000000	19.00000000	1.64002074991974E-01
19	3.69410630812969E+00	21.00000000	20.00000000	4.80036929888008E-01
20	2.12119337663930E+00	22.00000000	21.00000000	1.98978464678506E-01
21	3.29226169986604E+00	23.00000000	22.00000000	2.74242352454530E-01
22	2.69431893681548E+00	24.00000000	23.00000000	2.10797355964502E-01
23	3.55722506557019E+00	25.00000000	24.00000000	4.25450123304972E-01
24	3.21231680365682E+00	26.00000000	25.00000000	2.66030063796498E-01
25	5.57303818437308E+00	27.00000000	26.00000000	8.31396336326995E-01
26	2.83318396780379E+00	28.00000000	27.00000000	1.61659190882511E-01
27	2.71982086962996E+00	29.00000000	28.00000000	2.36807862360624E-01
28	2.87491034450443E+00	30.00000000	29.00000000	2.76233477645996E-01
29	4.87946521998321E+00	31.00000000	30.00000000	6.04440722521535E-01
30	2.17390506964361E+00	32.00000000	31.00000000	1.44602115287743E-01
31	3.08073189767848E+00	33.00000000	32.00000000	3.78606789340988E-01
32	2.61390023701688E+00	34.00000000	33.00000000	2.01297884506947E-01
33	3.67689493257591E+00	35.00000000	34.00000000	4.00060999026999E-01

Figure 1.6: Base Data for California

1.2.3 Working With Data

There are several methods that can be used to place data in STARS projects. They fall under the follwing categories: Convert, Merge, Join. There are also three different types of data:

CS Cross-Sectional Data

TS Time-Series Data

CSTS Cross-Sectional/Time-Series (Panel) Data.

Convert Data: California Example

You will notice that when you finished the initial stage of creating a project you were left with a table containing your base data.[3] As of right now, none of these fields have been added to our STARS project. Usually the first stage in adding data to a project involves **converting** these fields into STARS variables. Let's get our feet wet by continuing with the California example.

1. Click **Data-Variable-Convert-Base Data to CS**

   - You will encounter a dialogue box with all of the base data *Fields* in one listbox, and a series of *Optional Arguments* in the other.
   - Let's choose "AREA" as a variable we would like to add to our STARS project. To do so, highlight "AREA" in the listbox and click the "+" button.[4]
   - Highlight "Sum" as your *Optional Argument*, and click "OK". (See Figure 1.7)[5]
   - You have just create a new "CS" variable for your STARS project. To view your results you can use the appropriate table.

For a complete explanation of all the **Convert Data** methods see section 1.3.8.

Merge Data: California Example

Suppose you want to add some data to your project that was not included in your base datafile. The Merge and Join methods will allow you to augment your project appropriately. We are going to start with **Merging Data**. For a complete explanation of merging data see Section 1.3.9.

1. Click **Data-Variable-Merge-CSTS Data**

   - You will encounter a directory/file browser that should identify only files with a ***.csv** extension.
   - Navigate to the "myDataDir/data" directory. You will notice several files available.
   - The file "caPCR.csv" contains per capita income data from 1969–2001.[6] Select this file and click "Open". (See Figure 1.8)

[3] Please note that the data tables are limited to contain 50 rows. This prevents the package from slowing down when the number of observations increases.

[4] You could choose more than one variable to convert at the same time.

[5] A complete explanation of the *Optional Arguments* can be found in 1.3.6.

[6] Standardized to be relative to the California average across time periods.

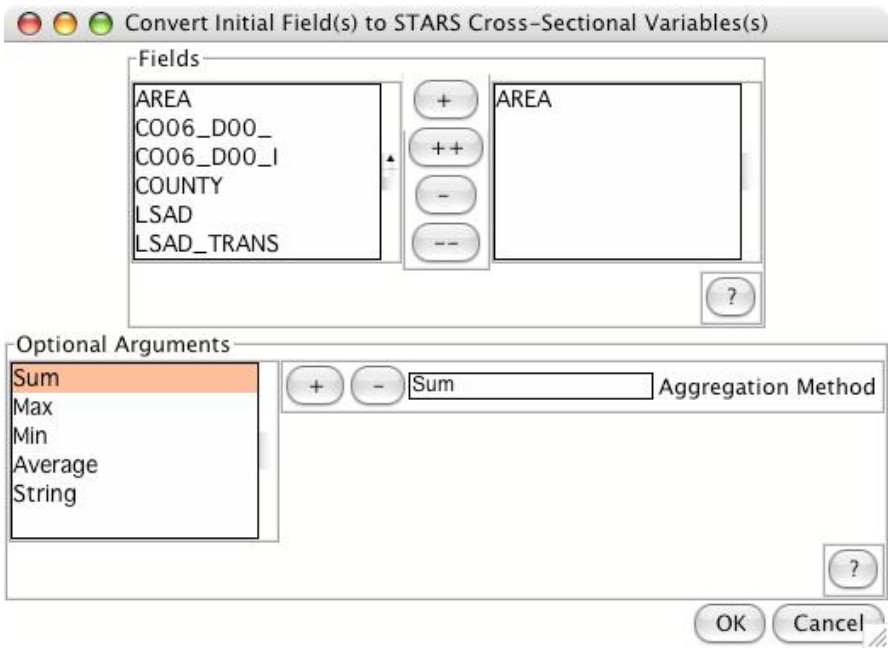


Figure 1.7: Convert CS Data (California)

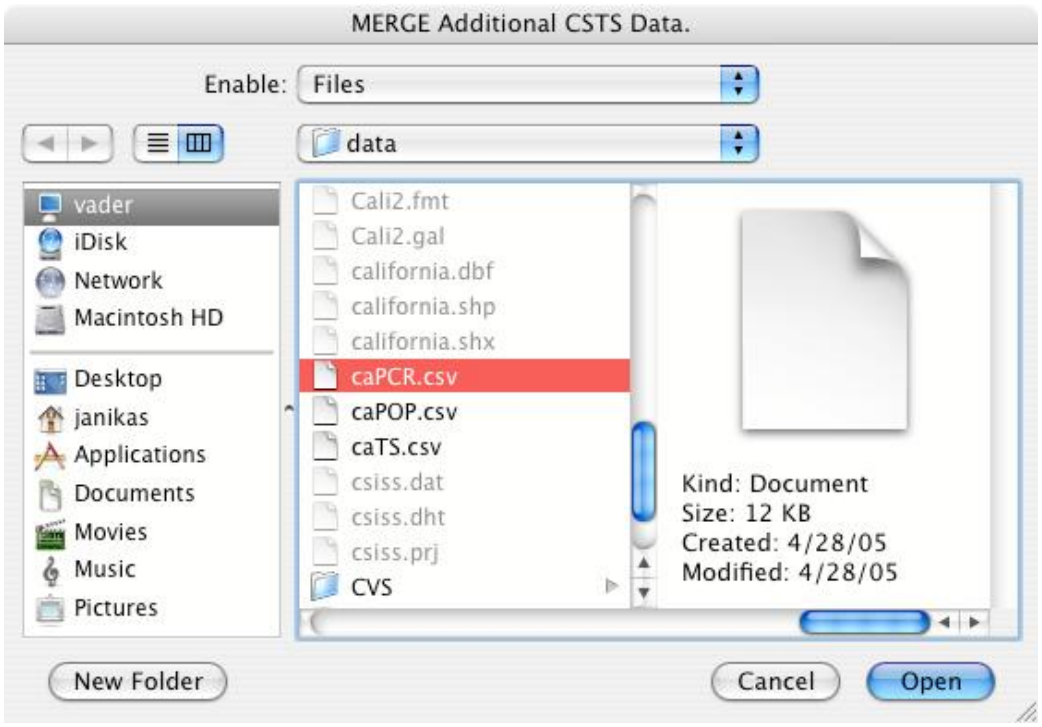


Figure 1.8: Merging CSTS Data (California)

Thats it! That is all there is to it! Merging data is easy on the front end, but requires that the data is ordered in the same way as the project you just created. How do you do that? See Section 1.3.9 for a complete explanation.

You could take this opportunity to repeat this process for the file "caPOP.csv", which contains population values for the CA counties over the time series. Furthermore, merging TS and CS data is just as easy. You can click **Data-Variable-Merge-TS Data** and add the file "caTS.csv" to see an example. This data contains the population and per capita income values for California as a whole over the time period.

Join Data: California Example

Joining data is similar to merging data, but the data does not have to be in a specific order. The data must be correctly matched through the use of a common field; one of which is in your *base data*, the other is in the new data file itself. See Section 1.3.10 for a full description. Let's try an example with California.

1. Click **Data-Variable-Join-CS Data**
    - You will encounter a directory/file browser that should identify only files with a ***.csv** extension.
    - Navigate to the "myDataDir/data" directory. You will notice several files available.
    - Choose the file "caJoinCS.csv".
    - You will encounter a dialogue box where you must choose the appropriate fields to match.
    - The **Master Field** contains CS variables that have already been created. We are going to choose "csnames" as our **master**. The "COUNTY" field is the **slave** in this instance. The master and the slave fields have the same values in them, but the order is different. Behind the scenes your new data will be matched accordingly. Figure 1.9 demonstrates the correct way to join this data file. Select "OK".

The steps for joining CSTS data are practically identical.[7]. Try clicking: **Data-Variable-Join-CSTS Data** and select "caJoinCSTS.csv". Again, use "csnames" as the master and "COUNTY" as the slave.

Adding a GAL Matrix: California Example

Often users have GAL (Geographic Algorithim Library) matrix files already created and want to add them to the project. It is important to note that the cross-sectional "ids" within the "*.gal" file, must be *identical* to those in your new project.[8] Again, anytime a user would like a print-out of the cross-sectional order of their project they can use the Write Cross-Section Names menu option.

To continue with the California example:

1. Click **Data-Matrix-Import GAL Binary**
    - You will encounter a directory/file browser that should identify only files with a ***.gal** extension.

[7] Again, see Section 1.3.10 for a full explanation.

[8] The "ids" of your cross-sectional units are the CS variable *csids* in your project.

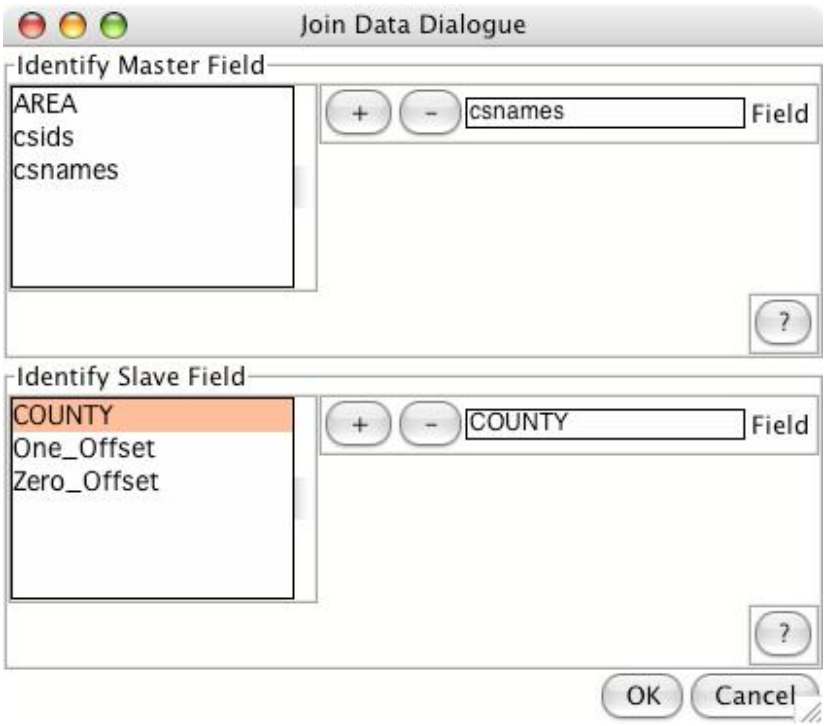


Figure 1.9: Joining CS Data (California)

- Navigate to the "myDataDir/data" directory. You will notice several files available (some of them might have been created in the initial stages of the project.).
- Choose the file "Cali1.gal".[9]
- Select "OK".

1.2.4 Finishing Up: California Example

Once you have added all of the data and matrices you want in your new STARS project, it will be time to write the files. If your project has a shapefile, you must plot your map first before you can proceed. In the case of California:

1. Click **Plot-Plot Map**
   - You will be prompted to choose the type of projection you want for your map.
   - Figure 1.14 contains the dialogue box with your choices.
   - Choose a type and select "OK".

Figure 1.10 contains the plotted map with the "Mercator" projection. If you decide you do not like the projection you initially chose, you can go back and change it before you save your data. Just repeat the steps until you find a projection that suits you.

[9]This GAL file is identical to the "Queen Contiguity Matrix" created in the initial stage.

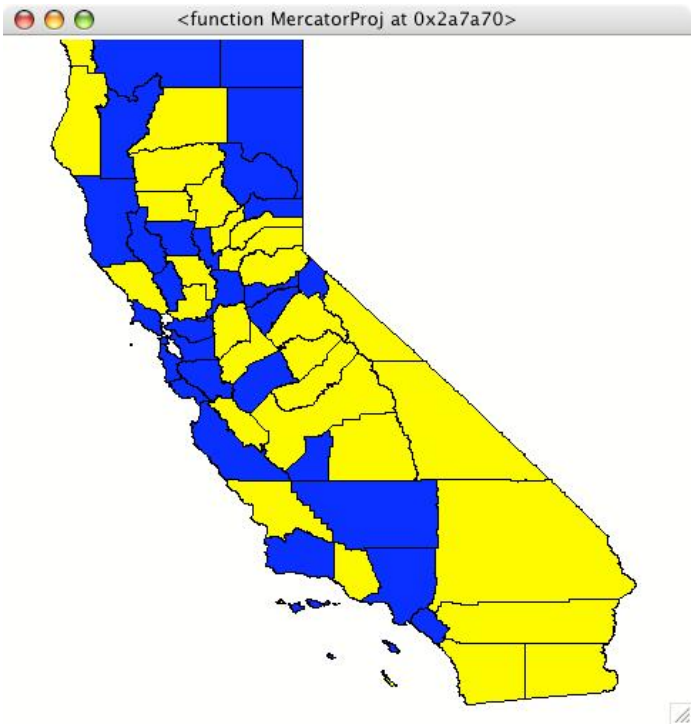


Figure 1.10: Mercator Projection for California

Now you can save your project and quit the **Project Maker** environment. Thats it! Your all done! Now you can open your "CA.prj" file in the STARS environment and start your exploration!

1.3 Project Maker: General

The following sections provide a more general explanation of the inner workings of the **Project Maker** utility.

1.3.1 Write Cross-Section Names

When a user wants to merge data or import a matrix, the order of the cross-sections must match with the current project. Most of the times this order is not known initially. One can obtain a file with the appropriate order by clicking **File-Write Cross-Section Names**. It is important to note that this file can only be created for a **new project** after the initial base data has been read and organized.

1.3.2 Saving Your Project

There are two ways to save your STARS Project:

1. **File-Save Project** Use this option if you are content with the name you have given your new project. It will write all of the appropriate files.

2. **File-Save Project As** Use this option if you want to either change the name of your project, or if you want to save your project to a new directory.

Rememeber that you must plot you map first if a shapefile is involved in your project.

1.3.3 CSV Projects

STARS does not require that your project contain a map. A comma-delimited file ***.csv** can be used to create a project. There are limitations however, as all of the mapping utilities in STARS will be nullified. For those interested in this type of case, you can proceed through the California Example again, but with the **CSV** option. The *base data* file for California is: "myDataDir/data/california.csv". The remaining steps are the same, save the plotting feature.

1.3.4 Time-Series Types

There are several canned types of time-series projects in STARS. Here is a brief description:

1. Decadal: Provide the start year and the end year when creating the project.
2. Annual: Provide the start year and the end year when creating the project.
3. Quarterly: Provide the start quarter, start year, end quarter, and end year.
4. Monthly: Provide the start month, start year, end month, and end year.
5. Irregular: For all other cases, it is best to use the irregular time-series. Just provide the number of time intervals in your project.

One Time Period

If your project only has one time period then you need to fake it. Try using **Irregular** as you time-series type and select 2 intervals. You wont be able to use the visualization techniques in STARS for time-series variables but you wont have any, so you should'nt miss it.

1.3.5 Creating Names and IDs

This is an extremely important stage in the creation of your project, because it serves as the accounting system between your maps and your data. There are four entry spaces in this view (See Figures 1.4): "Unique Field", "Join Field", "Optional Name Field", and "Delimiter". In most circumstances, your base data file (***.dbf, *.csv**) will have a unique name field (i.e. you will only see each name once down the column). In this case, you should always choose that field for "Unique Field" and leave the other entries blank.

Using the Join Field

Sometimes the base data file has more than one cross-section with the same name. For example, if your study area included the counties and municipios along the US/Mexico border you would find that that there are more than one cross-section with the name "Hidalgo". You could imagine quite a few repetitive names if you were to use all of the US counties as your study area. In this case, you can go two routes:

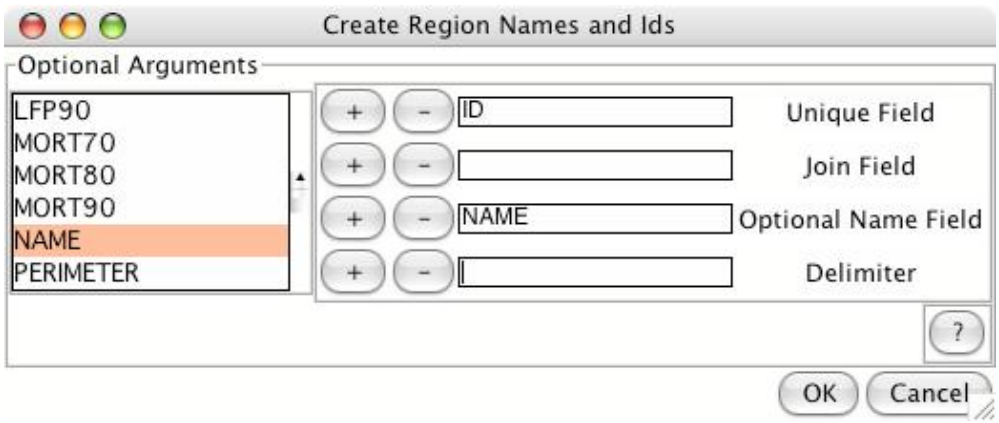


Figure 1.11: Joining with a Unique Field

1. In most cases there is another field that is unique to each cross-section. Identify this field and place it in the "Unique Field" entry space. If you want your cross-sections to be named as per this field, then you are done. However, you can select another field to be used as the names by placing it in the "Optional Name Field" entry space (i.e. your "Unique Field" will be used for matching, but the "Optional Name Field" will be used for actually naming your cross-sections.) In this case, you could have more than one cross-section with the same name in your completed STARS project. Figure 1.11 is an example where "ID" represents a unique field, and "NAMES" represents the cross-sectional names you want associated with your STARS project.

2. In some cases there is not a unique field for naming your cross-sections. When confronted with this situation you may want to combine two fields to make a unique one. For example, if one were to be using the US counties as their study area each cross-section could have a state (fips) code. While the name field is not unique, you can be certain that within each state their is no repetition of county names. To make unique names you could place the state codes in the "Unique Field" entry space, and the names in the "Join Field" entry space. The "Delimiter" field allows you to assign the string character you would like to use to combine the two. Figure 1.12 shows an example where "STATE" is the state code, "NAME" is the name field, and the string ":" is used to combine the two to make a unique cross-sectional name field. The result for a county named "Hidalgo" in state "01" would be: "01:Hidalgo".

Aggregation: The Case Of Islands

The last major circumstance one may confront in the creation of Names and IDs in **Project Maker** has to do with aggregation. Some shapefiles contain cross-sections that are made up of a series of islands. For example, Santa Barbara County in California is made up of a major land mass and several islands. In the accounting framework for California we used the "NAMES" field for a unique identifier. Behind the scenes, an aggregation method was employed to combine all shapes with the same "NAME" to form a single cross-sectional unit. In essence, the "NAME" field was not unique, but instead of joining fields or using a seperate unique field to keep all of the initial

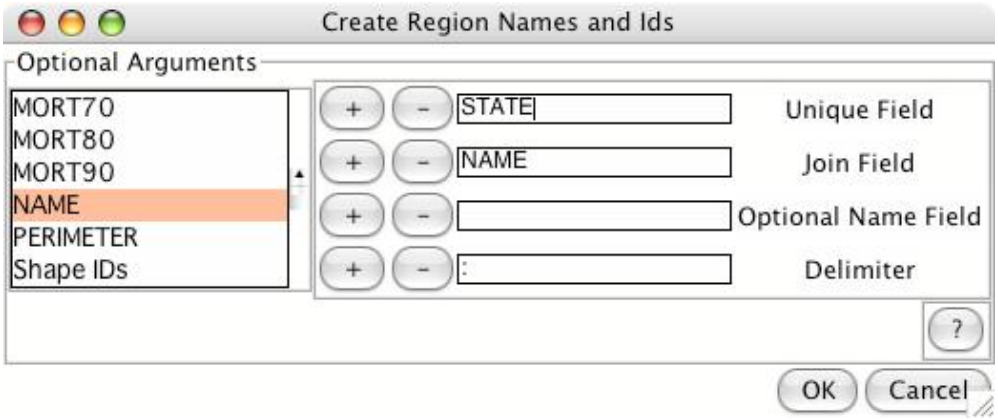


Figure 1.12: Joining without a Unique Field

shapes seperate, we *aggregated* to the number of unique values in the field. Hence, 68 shapes in the base data, became 58 unique cross-sections in the STARS project. We did this because there are 58 counties in California and the data corresponds to this number. In some cases, one would not want to aggregate, in which case they would look for a field in the base data that is truly unique.

The aggregation process is pretty straightforward, but it leads to several optional arguments that should be considered when confronted with this situation.

1.3.6 Optional Arguments

When an aggregation occurs in the base data it is necessary to provide optional arguments when converting base data to STARS variables. If a new cross-sectional unit is made up of a series of islands, each seperate shape will have attributes that need to be aggregated as well. Your choices include:

- Sum: Adds the values for all the shapes that make up each cross-section. We used this in the California Example for the field "AREA". This makes sense because we are aggregating several shapes into one, and we would probably want the total area for each cross-section.

- Max: Returns that maximum value for the variable in question for all shapes that make up each cross-section.

- Min: Returns that minimum value for the variable in question for all shapes that make up each cross-section.

- Average: Returns that average value for the variable in question for all shapes that make up each cross-section.

- String: Returns a string value for the aggregated cross-section. Users can use this for regime based variables, or to create a field with which to Join Data

1.3.7 Importing Weights Matrices

STARS has the ability to add GAL (Geographic Algorithim Library) matrix files to a project. See Section 1.2.3 for a an example.

1.3.8 Convert Data

This section provides a general description of converting data in the **Project Maker** environment.

- **Data-Variable-Convert-Base Data to CS**
  Creates a CS variable from your original base data. You will be prompted to select a field name from your original data and an optional argument.

- **Data-Variable-Convert-Base Data to CSTS**
  Creates a CSTS variable from your original base data. You will be prompted to select a series of fields from your original data and an optional argument. The number of fields you select must be identical to the number of time-periods in your study, and the fields should be selected in the correct order.

- **Data-Variable-Convert-Base Data to CSTS (Batch)**
  Creates a CSTS variable from your original base data on a Batch basis. **Project Maker** looks for base data field names with similar prefixes and lists them for the user to choose from. The user can select as many as they want. Figure 1.13 contains an example image. There are 5 variables that had similar prefixes in more than one field: "EDU", "FERT", "LFP", "MORT", "TFR". They also had similar suffixes: "70", "80", "90". In the ***.dbf** or ***.csv** base data file "EDU" would have 3 columns: "EDU70", "EDU80", "EDU90". In Figure 1.13 you will notice that we are going to create 3 new CSTS variables: "EDU", "FERT", "MORT". We use "70" and "90" as the start and end periods respectively. We then indicate that the increments between time periods is "10". Lastly, you have a choice of optional arguments. The batch method is a quick and effective way to create CSTS variables from more than one column in your original data file.

- **Data-Variable-Convert-Cross-Section to Panel**
  Creates a CSTS variable from CS variables you have already created. You must provide a name for your new CSTS variable. You can also choose whether you want to delete the CS variables you used to create the new CSTS variable.

- **Data-Variable-Convert-Panel to Cross-Section**
  Creates t = # of time periods CS variables from an existing CSTS variable. For example, if you had annual data for income (1970:1990), and the existing CSTS variable was named "income", the resulting CS variable would be "income1970"... "income1990". You can also choose whether you want to delete the CSTS variable you used to create the new CS variables.

1.3.9 Merge Data

Merging data to your new STARS project is rather easy within the **Project Maker** environment, but requires that the data is ordered in an appropriate manner. When a new Project is being made, the Cross-Sectional order is related to the order of the original base data. The user usually

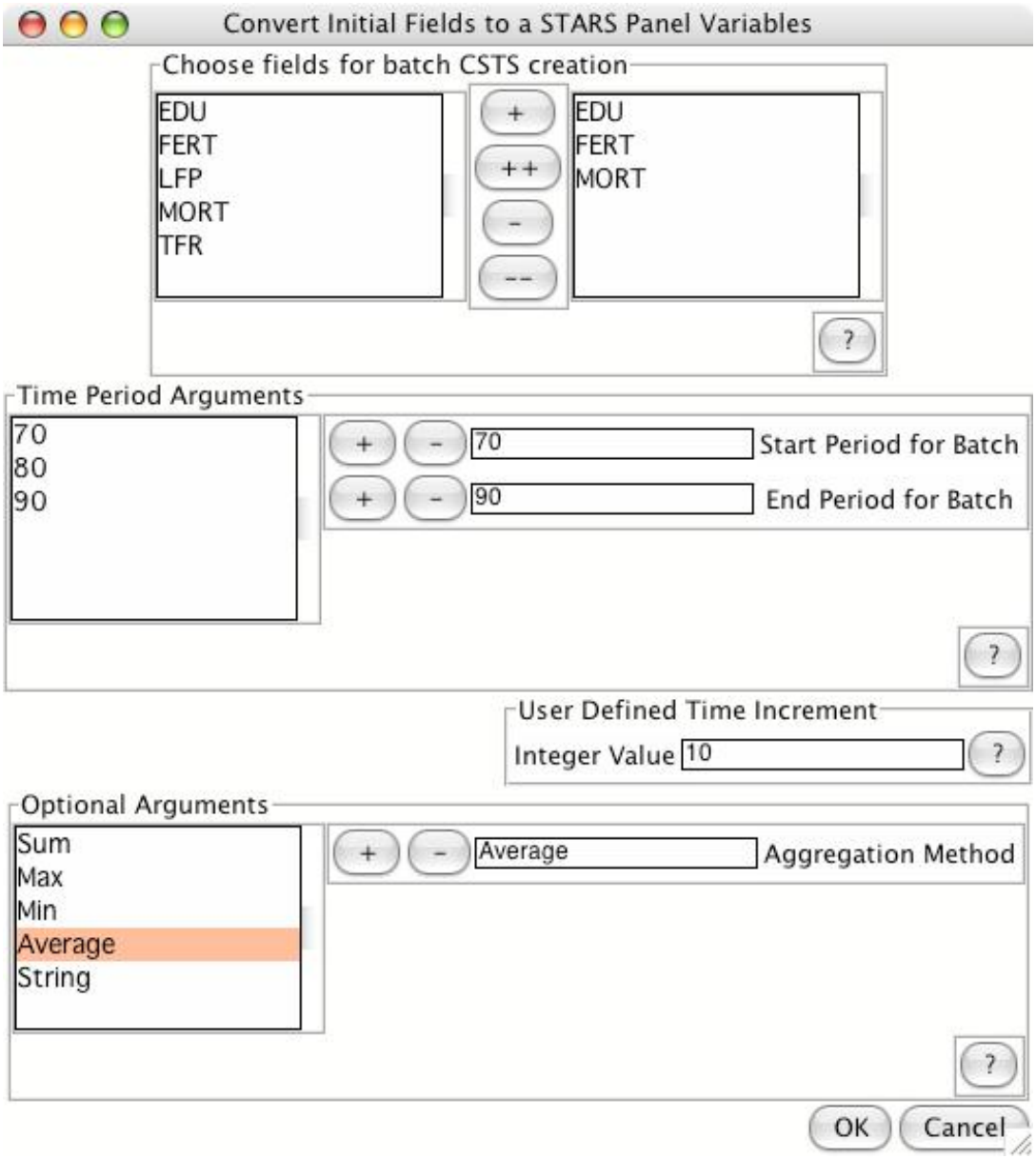


Figure 1.13: Creating Variables Via Batch

does not know this order to being with, so they often have to print the order to a file. If the data is ordered correctly then there are several options for merging data:

- **Data-Variable-Merge-CS Data**
  The user selects the ***.csv** file that contains CS variables in a column format. The first line in the ***.csv** contains the names of the new CS variables, and the remaining lines contain the values in the correct order. There should be $n + 1$ rows in the ***.csv**, one for the variable names, and n for the number of cross-sections.

- **Data-Variable-Merge-TS Data**
  The user selects the ***.csv** file that contains TS variables in a column format. The first row contains the names of the time-series variables, and the remaining lines contain the values. There should be $T + 1$ rows in the ***.csv**, one for the variable names, and T for the number of time periods.

- **Data-Variable-Merge-CSTS Data**
  The user selects the ***.csv** file that contains CSTS variable in a column format. You can only merge one CSTS variable per file. Therefore, the first line of the ***.csv** should only contain one variable name. The remaining lines contain the values in the correct order. There should be $n+1$ rows in the ***.csv**, one for the variable name, and n for the number of cross-sections. There should also be T columns in this file (the first row being the exception), one for each time period in the study.

1.3.10 Join Data

The join data method is different from merging because the data does not have to be in a pre-specified order. The join method uses a common field that exists in your new data file and in your new project to match the data appropriately. Often users will use "names" or "codes" to match their data. You can see an example in the California Tutorial. You can join data for CS or CSTS variables:

- **Data-Variable-Join-CS Data**
  The user selects the ***.csv** file that contains CS variables in a column format. The first line in the ***.csv** contains the names of the new CS variables as well as a name for a field that will be used for matching. There should be $n + 1$ rows in the ***.csv**, one for the variable names / matching field name, and n for the number of cross-sections. There should be $k + 1$ columns in the file: k for the number of new variables, and one extra for the matching field. The field for matching in your ***.csv** file is labeled as the "Slave". You will be prompted to match it with it's "Master" from the **Project Maker** environment. See Section 1.9 for an example.

- **Data-Variable-Join-CSTS Data**
  The user selects the ***.csv** file that contains CSTS variable in a column format. The first line in the ***.csv** contains the names of the new CSTS variable in order of T as well as a name for a field that will be used for matching. For example, the first row of your file may look like: "NAME", "VAR_1", "VAR2_" ... "VAR_T"
  There should be $n + 1$ rows in the ***.csv**, one for the variable names / matching field name, and n for the number of cross-sections. There should be $T + 1$ columns in the file: T for the

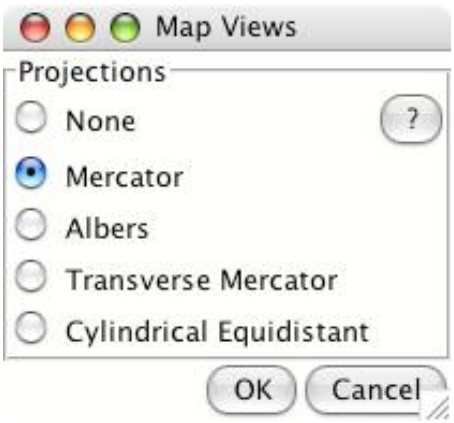


Figure 1.14: Possible Projections for Plotting

number of time periods, and one extra for the matching field. The field for matching in your ***.csv** file is labeled as the "Slave". You will be prompted to match it with it's "Master" from the **Project Maker** environment. See Section 1.9 for an example.

1.3.11 Tables

The **Tables** allows the user to view their data. The choices are as follows:

1. **Specific Variables:** Allows the use to select individual variables that have been added to the project for viewing.
2. **CS Variables:** Creates a table with all of the CS variables created so far.
3. **TS Variables:** Creates a table with all of the TS variables created so far.
4. **CSTS Variables:** Creates a table with all of the CSTS variables created so far.
5. **CS and CSTS Variables:** Creates a table with all of the CS and CSTS variables created so far.
6. **Base Data** Re-creates a table containing all your *base data*.

1.3.12 Plotting and Projections

All new projects that contain maps must be projected and plotted before the project can be completed. Figure 1.14 contains the possible projections for you STARS project map.

Resources

The project homepage for STARS is at http://stars-sf.py.

The STARS-User mailing list is at https://sourceforge.net/mailarchive/forum.php?forum_id=41297

The development page for STARS is at https://sourceforge.net/projects/stars-py/.

License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their

rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate

copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices
    stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in
    whole or in part contains or is derived from the Program or any
    part thereof, to be licensed as a whole at no charge to all third
    parties under the terms of this License.

    c) If the modified program normally reads commands interactively
    when run, you must cause it, when started running for such
    interactive use in the most ordinary way, to print or display an
    announcement including an appropriate copyright notice and a
    notice that there is no warranty (or else, saying that you provide
    a warranty) and that users may redistribute the program under
    these conditions, and telling the user how to view a copy of this
    License. (Exception: if the Program itself is interactive but
    does not normally print such an announcement, your work based on
    the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or

collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any

such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License

```
    along with this program; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
    Gnomovision version 69, Copyright (C) year  name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type ‘show c’ for details.
```

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Bibliography