TempMap v1.0.0

User Guide

For Windows, you start the temperature mapping program by typing TempMap in the Command Prompt window or by double-clicking on the TempMap.exe icon.

When you run TempMap, you will see the following screen. The different TempMap fields are numbered, and a description of each is provided below.

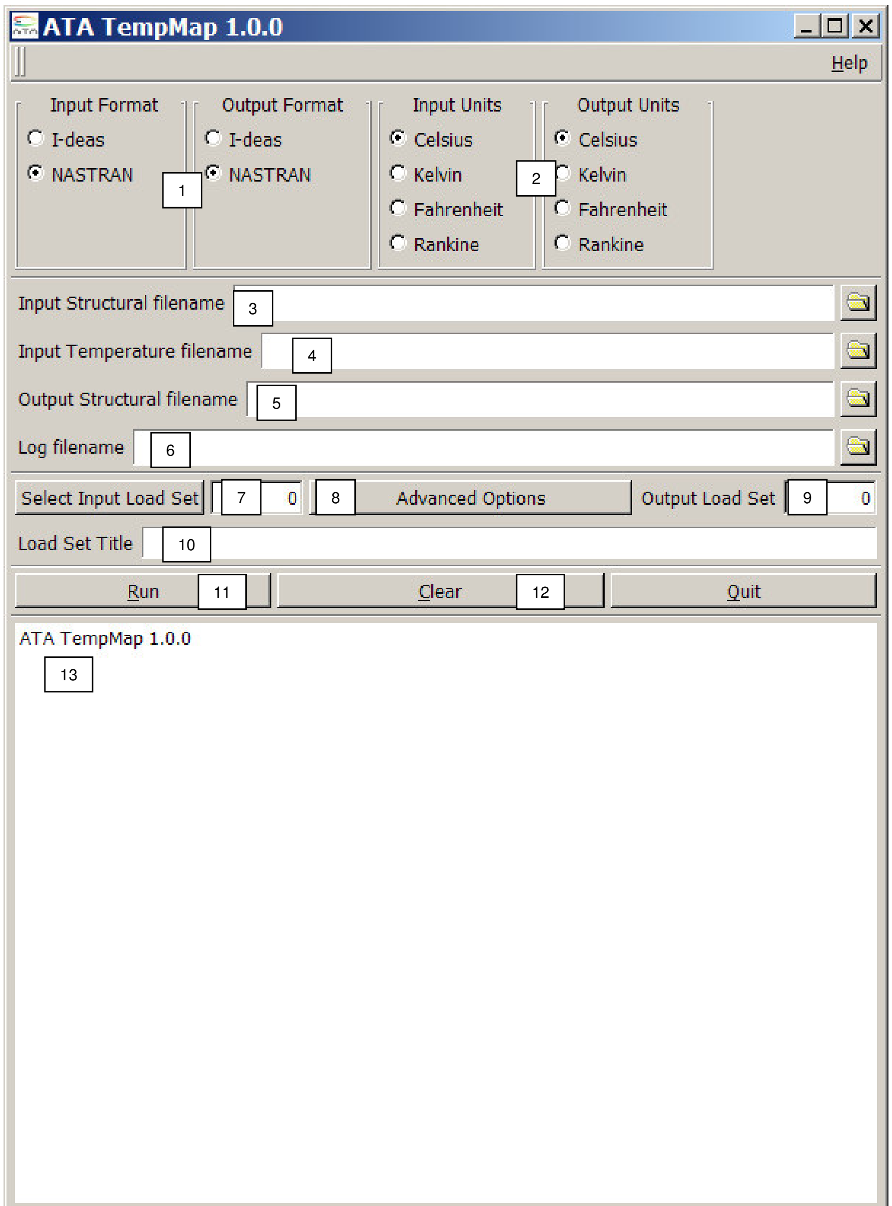

Figure 1. TempMap Display showing the key input and output fields.

[1] Input/Output formats
These toggles allow the user to define the type structural file that will be used for input and output. Currently, TempMap can read and write I-deas and NASTRAN files.

[2] Input/Output units
These toggles allow the user to define the input and output units. This allows the user to convert the temperatures to the appropriate units. A Units data set is written to the I-deas universal file, and the units are written as comments in the NASTRAN output file.

[3] Input Structural filename
This is the finite-element model (FEM) used for the temperature mapping.

From I-deas, the user will need to write out a universal file of the FEM. This FEM must contain at least one Boundary Condition Load Set containing Nodal Temperatures (see I-deas help on how to create Nodal Temperatures).

For NASTRAN, the model must contain loads written as TEMP cards. No other temperature cards are currently recognized by TempMap. In addition to loads and grids, coordinate systems are read, and the grid coordinates are transformed to Coordinate System 0.

For any input model type, the model must use same length units as the temperature data, i.e. if the temperature model has length in inches, the model file must also be in inches.

[4] Input Temperature file name
Enter the name of the file containing the temperature data the user want to apply to the user FEM. The format and contents of this file are defined in Appendix A.

The name of the input file will be used to seed the output structural file and log file names.

[5] Output Structural filename
This file contains the mapped temperature load set data. The name must not be the same as the input structural and temperature files.

The I-DEAS loads will be written out to Data Sets 792 and 2414 and the NASTRAN loads will be written as TEMP cards.

[6] Log filename
This file contains the log of all processing messages.

[7] Select Input Load Set
The user selects a Boundary Condition Set containing Node Temperatures. If the user presses the button, TempMap will determine all of the loads sets in the structural model, and give a user a list to choose from. The user can also enter the load set ID manually.

[8] Advanced Options

By clicking on this button, TempMap brings up the "Advanced Options" window, shown in Figure 2 below.

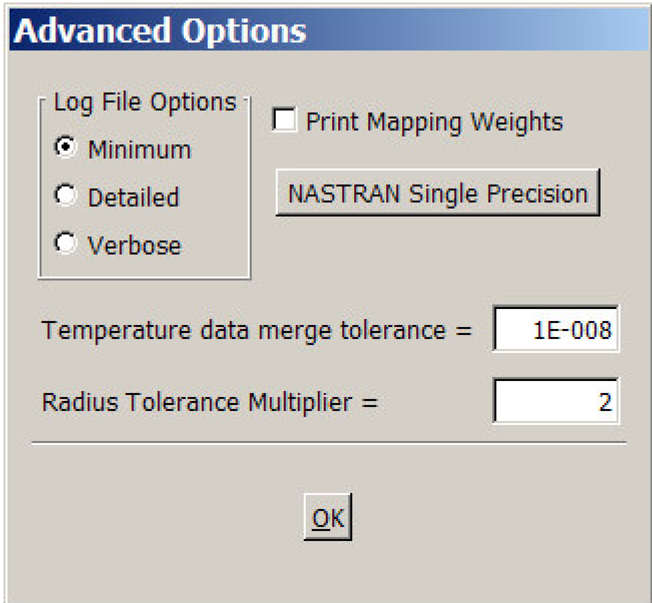

Figure 2. Advanced Options Window.

The "**Log File Options**" adjusts the amount of information written to the log file. Use minimum for most cases.

The "**Print Mapping Weights**" toggle allow the user to write out the mapping weights for the mapping from the thermal to structural model. The file is named with the same as the log file, except the file extension will be ".map", i.e. for a log file "example.log", the map file will be "example.map". The default is that the weights will not be printed

The format of the file is the structural grid id, then the number of thermal grids used to map the temperature, then the thermal grid id and weight for each of the thermal grids used in the mapping. The field are separated by tabs. An example is shown below:

```
53    2     15    0.993109   35    0.006891
```

The "**NASTRAN Single Precision / NASTRAN Double Precision**" toggle allows the user to toggle the output in NASTRAN single- or double-precision formats. The default format is single precision.

The "**Temperature data merge tolerance**" option sets the radius at which two temperature data points are considered to be coincident. The default is 1E-08, and the minimum is 0.0.

The "**Radius Tolerance Multiplier**" option allows the user to select a multiplier on the radius which is use for finding nearest temperature point. This option is useful if there significant space between the temperature data and the representative structure. For example, if there is a non-structural thermal protective coating on a structure, the temperature mesh may be on the outer surface of the protective coating, and the structure may be on the inner mold line. The user should change this value only if TempMap issues warning about not finding temperatures for elements. The default is 2.

[9] Output Load Set

I-deas requires a unique load set number for each load set in a FEM. The program automatically creates a unique load set number based on the input structural file when the user presses the "Select Input Load Set". If the user is performing multiple temperature mappings, make sure all of the user load set IDs are unique.

For the I-deas output option, TempMap also creates a result set containing the computed temperatures, so that the user can view contour plots of the data in post-processing. The result set has the same name as the load set, and its data type is "temperature".

[10] Load Set Title

The title is used in the output file as a description of the model. If the temperature file has a title, this will be the default.

[11] Run

Pressing this button begins the mapping process.

[12] Clear

This will clear all input.

[13] Output window

This window will contain the output messages from the mapping procedure. Typical output from the program is shown below:

```
Running ATA TempMap 1.0.0...

* Scanning universal file ...
  - 173 Nodes in universal file
  - 151 Elements in universal file
  - Mapping data on Temperature Set:
    1 - TEMPERATURE SET 1
    Output Load Set and Results Set will be:
    2 - NONE

* Reading Nodes from universal file ...
  - 173 Nodes stored

* Reading Temperatures from universal file ...
  - 173 Temperatures stored

* Reading Temperature data ...
  - 4 Temperature data points read

* Computing bounding boxes for structural and temperature models ...
  - 100% of FEM within temperature model boundary

* Sorting Temperature data ...
```

```
  - Looking for duplicates
  - No coincident Temperature Data Points found
  - Temperature data sorted

* Mapping Temperatures to FEM ...
  - 100 Mapped node temperatures computed
  - 173 Mapped node temperatures computed

* Writing mapped temperatures ...
  - 173 Mapped node temperatures written to output file

* 0 Warnings

* 0 Errors

* TempMap completed successfully
```

If less than 95% of the FEM lies within the temperature model, the program issues a warning but continues mapping. If the overlap is less than 1%, it stops processing and shows the user an error message (see below). This error typically occurs when the coordinate transformation definition (Appendix A) is not correct. TempMap writes full details of the two bounding boxes to the log file.

```
* Computing bounding boxes for structural and temperature models ...
  -   0% of FEM within temperature model boundary

* 1 WARNINGS - SEE LOG FILE

* 1 ERRORS - SEE LOG FILE

* TempMap COMPLETED WITH ERRORS
```

Finally, while TempMap is designed to a robust tool to map temperatures from a temperature model to a Finite-element model, the user should look at the results of the temperature mapping to ensure that the data is correct.

TempMap License

TempMap uses the Sentinel RMS Server for licensing. This is the current license server for ATA products. Please go to http://www.ata-e.com/software/rmsserver/index.html for more details on this server.

TempMap Revision History

Version 1.0

- Original release of TempMap

Appendix A: Temperature File Contents and Format

The temperature data file is delimited into blocks of data by keywords. Not all of the keywords are necessary, but if included, the keywords must be in the order shown below:

1. **TITLE**. Optional. If present, must be first record in file.
2. **COORD**. Optional. Defines transformation from the temperature model coordinate system to FEM coordinate system. If not present, mapping program assumes the temperature model and FEM coordinate systems are identical.
3. **SYMM**. Optional. Allows the temperature data to be reflected through a symmetry plane.
4. **TIMES**. Optional (*Required for multiple times steps*). Allows for temperatures at multiple times or cases.
5. **OUTTIMES.** Optional. Allows for multiple output times.
6. **DYNAMIC.** Optional. Prints out data for a transient analysis. Only the "DYNAMIC" keyword is needed. The "DYNAMIC" keyword is ignored for I-deas.
7. **TEMPERATURE.** Required. Coordinates of the points and the temperature at these points. At least one required.
8. **CYL.** Optional. Sets the coordinate system as cylindrical. Otherwise, the data is in Cartesian coordinates. If this card is used, it must be on the next non-comment line following TEMPERATURE.

An alternate format of the temperature data file replaces the TEMPERATURE keywords with GRIDS and TEMPS. This will be discussed later.

The format of the temperature data file is as follows:

1. For comments, use a '$'. Any information after the first '$' will be ignored:
   For example:

```
$  This entire line is a comment
3.4 2.7 4.5 0.1   $ all data after the first '$' are ignored.
```

2. For a title, two lines are necessary. The first line has the keyword "TITLE" as its first 5 characters, and the second line contains the title. For example:

```
TITLE
The next uncommented line is the title for the results
```

3. For a coordinate transformation, 4 lines are necessary. The first line contains the keyword "COORD" as its first 5 characters, and the next 3 lines contain the definition of the coordinate system transformation, as shown below. The data lines are free-format:

```
COORD
X0 Y0 Z0 $ The offset from TM origin to FEM origin in the FEM CS.
X1 Y1 Z1 $ TM X-axis vector in FEM coordinate system (FEM CS).
X2 Y2 Z2 $ A vector defining the TM XZ plane in the FEM CS.
```

The transformation from the temperature model to FEM is shown by the following equations:

$$\overrightarrow{X}_{FEM} = \begin{bmatrix} \vec{X} \\ \vec{Y} \\ \vec{Z} \end{bmatrix}^T \overrightarrow{X}_{TEMP} + \overrightarrow{X0}$$

where :

$\overrightarrow{X}_{TEMP}$ = coordinates in the temperature model coordinate system

$\overrightarrow{X}_{FEM}$ = coordinates in the FEM coordinate system

$$\vec{X} = \frac{\overrightarrow{X1}}{\left|\overrightarrow{X1}\right|}, \quad \vec{Y} = \frac{(\overrightarrow{X2} \times \overrightarrow{X1})}{\left|\overrightarrow{X2} \times \overrightarrow{X1}\right|}, \quad \vec{Z} = \vec{X} \times \vec{Y}$$

$$\overrightarrow{X0} = (X0, Y0, Z0), \quad \overrightarrow{X1} = (X1, Y1, Z1), \quad \overrightarrow{X2} = (X2, Y2, Z2)$$

4. For the “SYMM” block, 3 lines are necessary. The first line contains the keyword “SYMM” as its first 4 characters, and the next 2 lines contain the definition of a symmetry plane in the FEM coordinate system, as shown below. The data lines are free-format:

```
SYMM
X0 Y0 Z0 $ A point on the symmetry plane in the FEM CS.
X1 Y1 Z1 $ The normal of the symmetry plane in the FEM CS.
```

For each body point not on the symmetry plane, an additional temperature point location is created by the following equations:

$$\overrightarrow{X}_{PSYMM} = \overrightarrow{X}_P - 2\left[(\overrightarrow{X}_P - \overrightarrow{X0}) \bullet \hat{X}1\right]\hat{X}1$$

where :

$\overrightarrow{X}_P$ = Original temperature point coordinates.

$\overrightarrow{X}_{PSYMM}$ = coordinates of symmetric temperature point.

$$\hat{X}1 = \frac{\overrightarrow{X1}}{\left|\overrightarrow{X1}\right|}, \quad \overrightarrow{X0} = (X0, Y0, Z0), \quad \overrightarrow{X1} = (X1, Y1, Z1)$$

5. For the input time data, at least 2 lines are needed. The first line contains the keyword “TIMES” as its first 5 characters, the next line contains the number of time values, and the third line and beyond contains the time values in ascending order, with 4 values per line, except for the last line, which can have 1 to 4 values.

```
TIMES
N           $ Number of times
T1 T2 T3 T4 $ Time values
...
TN-1 TN     $ Final row can have less than 4 values
```

If only 2 lines are input, the time values will default to 1, 2, 3, etc. For instance:

```
TIMES
10
```

will generate times 1 to 10.

6. For the outputs time data, at least 3 lines are needed. The first line contains the keyword "OUTTIMES" as its first 8 characters, the next line contains the number of time values, and the third line and beyond contains the time values in ascending order, with 4 values per line, except for the last line, which can have 1 to 4 values.

```
OUTTIMES
N            $ Number of times
T1 T2 T3 T4  $ Time values
T5 T6 T7 T8  $ Additional Time values
...
TN-1 TN      $ Final row can have less than 4 values
```

   If input times are defined but no output times are defined, the output times will be the same as the input times. Also, if output times are between input times, TempMap will interpolate the temperatures between two input times. Any output times greater than or less than the input times will be ignored.

7. For the temperature data, at least two lines are needed. The first contains the keyword "TEMPERATURE" as its first 11 characters, and each line that follows contains the coordinates of a grid point with the temperature at that point. The data lines are free-format. An example is shown below:

```
TEMPERATURE
X1 Y1 Z1 T1 $ Point and temperature
X2 Y2 Z2 T2
X3 Y3 Z3 T3
```

   If the data is in cylindrical coordinates, the temperature data will be as follows:

```
TEMPERATURE
CYL
R1 TH1 Z1 T1 $ Point and temperature
R2 TH2 Z2 T2
R3 TH3 Z3 T3
```

   Where THi are the angles in degrees. The cylindrical coordinates will be read, converted to Cartesian coordinates, and stored in memory.

If input times are defined, additional lines are needed for each temperature, with 4 values per line, except for the last line, which can have 1 to 4 values, as shown below:

```
TIMES
6
T1 T2 T3 T4
T5 T6
TEMPERATURE
X1    Y1    Z1    T1_1 $ Point and temperature
T1_2  T1_3  T1_4  T1_5 $ Additional Temperatures
T1_6                   $ Additional Temperatures
X2    Y2    Z2    T2_1
T2_2  T2_3  T2_4  T2_5
T2_6
```

For Cylindrical coordinates:

```
TIMES
6
T1 T2 T3 T4
T5 T6
TEMPERATURE
CYL
R1    TH1   Z1    T1_1 $ Point and temperature
T1_2  T1_3  T1_4  T1_5 $ Additional Temperatures
T1_6                   $ Additional Temperatures
R2    TH2   Z2    T2_1
T2_2  T2_3  T2_4  T2_5
T2_6
```

The following are valid temperature files:

```
TITLE
EXAMPLE TEMPERATURES
COORD
    9.30000E+01    0.00000E+00    0.00000E+00
    1.00000E+00    0.00000E+00    0.00000E+00
    0.00000E+00    0.00000E+00    1.00000E+00
$
SYMM
    9.30000E+01    0.00000E+00    0.00000E+00
    0.00000E+00    1.00000E+00    0.00000E+00
$
$    X, inch         Y, inch         Z, inch         T
$
TEMPERATURE
     1.14044E+01    -8.17367E+00    -2.34699E+01     24.1345
     6.42327E+00    -6.33882E+00    -2.34646E+01     11.4598
     1.14044E+01    -1.83098E+00    -3.27500E+01     10.2369
     6.46169E+02    -2.91713E+01    -3.26565E+01    -24.8273
     1.97866E+02    -2.94464E+01    -3.26963E+01    -48.8448
     2.35048E+02    -3.51424E+01    -2.62059E+01    -12.9118
```

and

```
TITLE
EXAMPLE TEMPERATURES, MULTIPLE TIMES
COORD
    9.30000E+01    0.00000E+00    0.00000E+00
    1.00000E+00    0.00000E+00    0.00000E+00
    0.00000E+00    0.00000E+00    1.00000E+00
SYMM
    9.30000E+01    0.00000E+00    0.00000E+00
    0.00000E+00    1.00000E+00    0.00000E+00
TIMES
3
0.0   1.0   2.0
OUTTIMES
4
0.0 0.5 1.0 1.5
DYNAMIC
$    X, inch         Y, inch         Z, inch         T1
$    T2              T3 ...
TEMPERATURE
     1.14044E+01    -8.17367E+00    -2.34699E+01     24.1345
     24.1345         24.1345
     6.42327E+00    -6.33882E+00    -2.34646E+01     11.4598
     11.4598         11.4598
     1.14044E+01    -1.83098E+00    -3.27500E+01     10.2369
     10.2369         10.2369
```

Keep in mind that the records are free-format; the indentations and adherence to columns shown here are used just for legibility, not because they are required.

The alternate temperature data file is delimited into blocks of data by keywords. Not all of the keywords are necessary, but if included, the keywords must be in the order shown below:

1. **TITLE**. Optional. If present, must be first record in file.

2. **COORD**. Optional. Defines transformation from the temperature model coordinate system to FEM coordinate system. If not present, mapping program assumes the temperature model and FEM coordinate systems are identical.
3. **SYMM**. Optional. Allows the temperature data to be reflected through a symmetry plane.
4. **TIMES**. Optional (*Required for multiple times steps*). Allows for temperatures at multiple times or cases.
5. **OUTTIMES.** Optional. Allows for multiple output times.
6. **DYNAMIC.** Optional. Prints out data for a transient analysis. Only the "DYNAMIC" keyword is needed. The "DYNAMIC" keyword is ignored for I-deas.
7. **GRIDS.** Required. Point IDs and coordinates of the points. At least one is required.
8. **CYL.** Optional. Sets the coordinate system as cylindrical. Otherwise, the data is in Cartesian coordinates. If this card is used, it must be on the next non-comment line following GRIDS.
9. **TEMPS.** Required. Point IDs and the temperature at these points. At least one is required. The number of Point IDs and the values of the Point IDs must match the GRIDS cards/

For the GRIDS and TEMPS data, at least two lines are needed for each. The first contains the keyword "GRIDS" as its first 5 characters, and each line that follows contains the point ID for a grid point and the coordinates of the grid point.

The GRIDS are followed by the "TEMPS". The first contains the keyword "TEMPS" as its first 5 characters, and each line that follows contains the point ID for a grid point with the temperature at that point.

The data lines are free-format. An example is shown below:

```
GRIDS
GR1 X1 Y1 Z1 $ Point ID and coordinates
GR2 X2 Y2 Z2
GR3 X3 Y3 Z3
TEMPS
GR1 T1 $ Point ID and temperature
GR2 T2
GR3 T3
```

If the data is in cylindrical coordinates, the temperature data will be as follows:

```
GRIDS
CYL
GR1 R1 TH1 Z1 $ Point ID and coordinates
GR2 R2 TH2 Z2
GR3 R3 TH3 Z3
TEMPS
GR1 T1 $ Point ID and temperature
GR2 T2
GR3 T3
```

Where THi are the angles in degrees. The cylindrical coordinates will be read, converted to Cartesian coordinates, and stored in memory.

If input times are defined, additional lines are needed for each temperature, with 4 values per line, except for the last line, which can have 1 to 4 values, as shown below:

```
TIMES
6
T1 T2 T3 T4
T5 T6
GRIDS
GR1 X1 Y1 Z1 $ Point ID and coordinates
GR2 X2 Y2 Z2
TEMPS
GR1   T1_1  T1_2  T1_3 $ Point ID and temperature
T1_4  T1_5  T1_6       $ Additional Temperatures
GR2   T2_1  T2_2  T2_3
T2_4  T2_5  T2_6
```

For Cylindrical coordinates:

```
TIMES
6
T1 T2 T3 T4
T5 T6
GRIDS
CYL
GR1 R1 TH1 Z1 $ Point ID and coordinates
GR2 R2 TH2 Z2
TEMPS
GR1   T1_1  T1_2  T1_3 $ Point ID and temperature
T1_4  T1_5  T1_6       $ Additional Temperatures
GR2   T2_1  T2_2  T2_3
T2_4  T2_5  T2_6
```

The following are valid alternate format temperature files:

```
TITLE
EXAMPLE TEMPERATURES
COORD
    9.30000E+01    0.00000E+00    0.00000E+00
    1.00000E+00    0.00000E+00    0.00000E+00
    0.00000E+00    0.00000E+00    1.00000E+00
$
SYMM
    9.30000E+01    0.00000E+00    0.00000E+00
    0.00000E+00    1.00000E+00    0.00000E+00
GRIDS
     1     1.14044E+01    -8.17367E+00    -2.34699E+01
     2     6.42327E+00    -6.33882E+00    -2.34646E+01
     3     1.14044E+01    -1.83098E+00    -3.27500E+01
TEMPS
     1     24.1345
     2     11.4598
     3     10.2369
```

and

```
TITLE
EXAMPLE TEMPERATURES, MULTIPLE TIMES
COORD
    9.30000E+01    0.00000E+00    0.00000E+00
    1.00000E+00    0.00000E+00    0.00000E+00
    0.00000E+00    0.00000E+00    1.00000E+00
SYMM
    9.30000E+01    0.00000E+00    0.00000E+00
    0.00000E+00    1.00000E+00    0.00000E+00
TIMES
5
0.0   0.5   1.0  1.5  2.0
OUTTIMES
4
0.0 0.5 1.0 1.5
DYNAMIC
GRIDS
     1     1.14044E+01    -8.17367E+00    -2.34699E+01
     2     6.42327E+00    -6.33882E+00    -2.34646E+01
     3     1.14044E+01    -1.83098E+00    -3.27500E+01
TEMPS
         1   24.1345   24.1345   24.1345
   24.1345   24.1345
         2   11.4598   11.4598   11.4598
   11.4598   11.4598
         3   10.2369   10.2369   10.2369
   10.2369   10.2369
```

TempMap also allows the user to include additional files by use of an #include statement. If the user has an input file with the following:

```
TITLE
EXAMPLE TEMPERATURES
COORD
    9.30000E+01    0.00000E+00    0.00000E+00
    1.00000E+00    0.00000E+00    0.00000E+00
    0.00000E+00    0.00000E+00    1.00000E+00
$
SYMM
    9.30000E+01    0.00000E+00    0.00000E+00
    0.00000E+00    1.00000E+00    0.00000E+00
$
#include "file2.dat"
```

And a **file2.dat** that has the lines:

```
$
$    X, inch          Y, inch          Z, inch          T
$
TEMPERATURE
     1.14044E+01    -8.17367E+00    -2.34699E+01     24.1345
     6.42327E+00    -6.33882E+00    -2.34646E+01     11.4598
     1.14044E+01    -1.83098E+00    -3.27500E+01     10.2369
     6.46169E+02    -2.91713E+01    -3.26565E+01    -24.8273
     1.97866E+02    -2.94464E+01    -3.26963E+01    -48.8448
     2.35048E+02    -3.51424E+01    -2.62059E+01    -12.9118
```

TempMap will read both files and assemble them together.