



TempMap

Temperature Mapping

User Guide

Version 2.3.0

© 2009–2023 ATA Engineering, Inc.
13290 Evening Creek Drive S
San Diego, California 92128

The content of this document is ATA proprietary and confidential information. This document is Copyright © ATA Engineering, Inc. 2009–2023. All rights reserved. No part of this work may be reproduced or used in any form or by any means, graphic, electronic, or mechanical (including photocopying, recording, taping, or information storage and retrieval systems) without the express written permission of ATA. All copies of this document must include the copyright notice as noted above and the other information contained in the paragraph.

TABLE OF CONTENTS

- What's New ii
- 1. Overview 1
 - 1.1. Features 1
- 2. TempMap User Guide 2
 - 2.1. Standard Inputs 2
 - 2.2. Advanced Options 6
 - 2.3. Preview and PostView 8
- 3. Installation..... 11
 - 3.1. Platform Requirements 11
 - 3.2. Installing the License Server..... 11
 - 3.2.1. Installing Sentinel RMS. 11
 - 3.2.2. Environment Variable. 11
 - 3.2.3. Checking Your License Status. 12
 - 3.3. Installing TempMap 12
 - 3.4. Configuring TempMap 12
- Appendix A. Temperature File Contents and Format 13
- Appendix B. Command-Line and Batch File Operation 23
- Appendix C. Revision History for Older Releases 26

What's New

This section describes the new features and bug fixes for each release of TempMap. For the revision history for older versions, see Appendix C.

Version 2.3.0.

Enhancements

- The executable is now natively 64-bit, which allows TempMap to work with much larger datasets.
- Cleaned up the GUI layout and made other minor improvements.
- Upgraded to use version 1.7.81 of Fox Toolkit GUI.

Bug Fixes

- Prevent crashes when incorrect input files are given.

Version 2.2.0.

Enhancements

- Upgraded to use version 1.6.57 of Fox Toolkit GUI.
- Now uses 64-bit version of qdelaunay.exe.
- Increased maximum memory usage to 4 GB.
- Improved memory usage for 3D interpolation.

Bug Fixes

- Fixed memory error by releasing data blocks no longer needed.

Version 2.1.0.

Enhancements

- Improved search algorithms.
- Updated licensing to RMS v9.2.1.

Version 2.0.2.

Enhancements

- Added option for multiple temperatures or single temperature on line for double precision. This option is not available through the command line.

Bug Fixes

- Fixed bug for Nastran single-precision output.

Version 2.0.1.

Bug Fixes

- Fixed bug crash if number of grids not equal to number of temps in Temperature file, alternate format. If more grids than temperatures, removes extra grids. If more temperatures than grids exist, skips temperature data for missing grids.

Version 2.0.0.

Enhancements

- Improved mapping of data around a cylindrical surface.
- Added volumetric mapping option.
- Added command line documentation.
- Added temperature scale option to Preview/PostView.

TempMap is a trademark of ATA Engineering, Inc.
© 2009–2023 ATA Engineering, Inc.

TempMap uses the Fox Toolkit GUI, version 1.7.81. Copyright © 1997, 2022 Jeroen van der Zijp. All rights reserved. Additional details can be found in the TempMap distribution files LGPL_LICENSE.txt and FOX_TOOLKIT_LICENSE.txt

TempMap also uses Qhull, Copyright ©1993–2019 C.B. Barber, Arlington, MA, and The National Science and Technology Research Center for Computation and Visualization of Geometric Structures (The Geometry Center) University of Minnesota (qhull@qhull.org). Additional details can be found in the TempMap distribution file COPYING.txt

I-deas is a trademark or registered trademark of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries.

Nastran is a registered trademark of the National Aeronautics and Space Administration.

All other trademarks are the property of their respective owners.

List of Figures

Figure 2-1. TempMap interface showing the key input and output fields.....	2
Figure 2-2. Advanced Options window.	4
Figure 2-3. Example temperature distribution where increasing the number of points will improve the temperature mapping.....	8
Figure 2-4. TempMap Preview display.	9
Figure 2-5. TempMap PostView display.	9
Figure 2-6. Selection of time set for cases with multiple time sets.	10

1. **OVERVIEW**

TempMap™ is a temperature mapping program that takes data from a cloud of points and maps the temperatures to a finite element model (FEM). TempMap has a graphical user interface (GUI) and can also be used as a command-line program. The command-line version is useful for batch processing of data.

1.1. **Features**

The TempMap user interface offers extensive capabilities for mapping temperatures. At the same time, it provides useful defaults that will be acceptable in many cases, thus making it easy to perform basic mappings. TempMap features include the following:

- GUI and command-line options
- Ability to map data from Nastran or I-deas
- Temperature units conversion
- Ability to select input temperature load set
- Previewing and postviewing of results
- Advanced options to tailor the mapping process, including two-dimensional (2D) or three-dimensional (3D) mappings

2. **TEMPMAP USER GUIDE**

TempMap runs under the Windows operating system. The user can start the program by typing *TempMap* in a **Command Prompt** window or by double-clicking on the **TempMap.exe** icon. The best way to start it is to open a command prompt in the directory where your input files reside, or where you want your output files to reside if that is not the same directory. In that prompt, type *tempmap*. TempMap also has the ability to be run as a command-line program. To activate the command-line option, type *tempmap -t* in the command prompt window and then answer the prompts as they appear on the screen. For command-line information and batch mode operations, see Appendix B.

2.1. **Standard Inputs**

The TempMap screen is shown in Figure 2-1. The various TempMap input fields are numbered in Figure 2-1, and a description of each field is provided below.

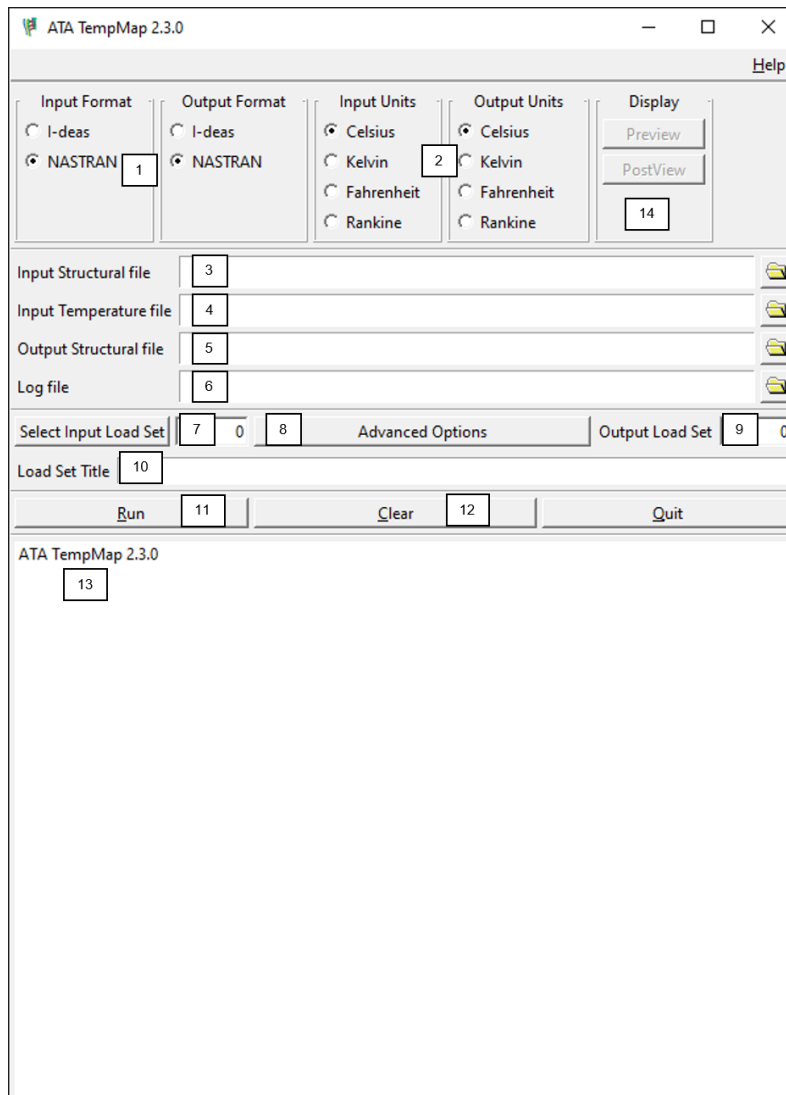


Figure 2-1. TempMap interface showing the key input and output fields.

[1] Input/Output Formats

These toggles allow the user to define the structural file format used for input and output. Currently, TempMap can read and write I-deas universal and Nastran input 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. When mapping to I-deas, a units data set is written to the output universal file. When mapping to Nastran, the units used are written as a comment in the output file.

[3] Input Structural Filename

This file includes the FEM that contains the target nodes for the temperature mapping. It must also include a dummy temperature load definition to identify which nodes will receive the mapped temperatures.

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.

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 input structural model must use the same length units as the input temperature data; i.e., if the temperature model has lengths in inches, the model file must also be in inches.

[4] Input Temperature Filename

This is the file containing the temperature data the user wants to map to the input structural 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 default 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 that of the input structural or temperature file.

I-deas loads will be written out to data sets 792 and 2414. 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 the dummy nodal temperatures that define the nodes that will have temperatures mapped to them. If the user presses this button, TempMap will identify all the load sets in the structural model and give the user a list from which to choose. The user can also enter the load set ID manually.

[8] Advanced Options

When the user clicks on this button, TempMap brings up the **Advanced Options** window, shown in Figure 2-2. These advanced options are described in section 2.2.

Figure 2-2. Advanced Options window.

[9] Output Load Set

I-deas requires a unique load set number for each load set in a FEM. TempMap automatically creates a unique default load set number based on the input structural file when the user presses the **Select Input Load Set** button. If the user is performing multiple temperature mappings, care should be taken to make sure that 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 postprocessing. 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

Pressing this button clears 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 2.3.0...
```

```
* 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%, TempMap 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, although TempMap is designed to be a robust tool to map temperatures from a temperature model to a FEM, the user should look at the results of the temperature mapping to ensure that the data are correct.

[14] Preview and PostView

The **Preview** and **PostView** buttons allow the user to display the model and the temperature point cloud. **Preview** can be used to display just the structural model, just the temperature point cloud, or both. **PostView** is disabled until a mapping is completed, at which point the user can display the mapped temperatures. A full description of the viewers is provided in section 2.3.

2.2. Advanced Options

PressMap's advanced options allow the user to change key settings to tailor the mapping procedures, and to provide additional information to the user.

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

The **Mapping Option** allows the user to specify whether the temperature data follows a surface or fills a volume. Do not set to volume if data is planar. The surface option uses a 2D Delaunay triangulation using N points set by the number of near temperatures to a node (see below). The surface can be curved, and TempMap recognizes planar, spherical, and cylindrical surfaces.

The volume option generates a 3D Delaunay triangulation based on all temperatures and then compares to volumes associated with the N points set by the number of near temperatures.

The **NASTRAN Format Option** allows the user to select the output format as either **Single Precision** or **Double Precision (single)** or **(multi)**. The default format is **Single**

Precision. For double precision, single will print only one temperature per TEMP* entry, and multi will print three temperatures per TEMP* entry.

The **Print Mapping Weights** option allows the user to write out the mapping weights for the mapping from the thermal to the structural model. The file is named 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*. This option is toggled off by default.

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. Fields are separated by tabs. An example is shown below:

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

The **Use Element Normals** option creates a local normal vector at a node based on the average of the element normals. The normal vector is used for projection of 3D temperature data onto a 2D surface local to the grid point. By default, this option is toggled off. In some cases, the temperature mapping may be improved by turning this option on.

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. With this option, if one temperature point lies within the radius of another temperature point, it is removed from the list of temperature points and the temperature at the remaining temperature point is the average of the two points. If N points are within the radius of another temperature point, the N points are removed, and the temperature at the remaining point is the average of the N+1 points.

The **Radius Tolerance Multiplier** option allows the user to select a multiplier on the radius used for finding the nearest temperature point. This option is useful if there is significant space between the temperature data and the representative structure. For example, if there is a nonstructural 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 a warning about not finding temperatures for nodes. The default is 2.

The **Number of Near Temperatures to Node** option allows the user to select the maximum number of temperature points near a node that are used to calculate the temperature mapping. Increasing the number of points helps in computing accurately mapped temperatures for data that are densely distributed in one direction but sparsely distributed in another direction, as shown in Figure 2-3. The default is 12 temperature points near the node. Please note that increasing the number of temperature points used to create the mapping will cause TempMap to run longer.

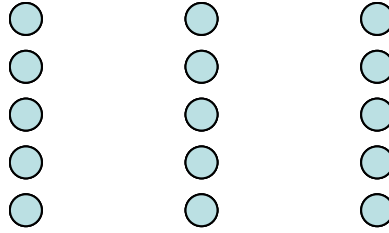


Figure 2-3. Example temperature distribution where increasing the number of points will improve the temperature mapping.

2.3. **Preview and PostView**

The **Preview** and **PostView** buttons allow the user to display the model and the temperature point cloud. **Preview** can be used to display just the structural model, just the temperature point cloud, or both. **PostView** is disabled until a mapping is completed, at which point the user can display the mapped temperatures. The temperatures are shown on a relative scale, with the blue end of the color spectrum representing the lower temperatures and the red end representing the higher temperatures. A color bar showing the minimum, average, and maximum temperatures is included in the display, and the limits can be changed by the user.

The view screen, shown in Figure 2-4 and Figure 2-5, operates in the same way for both views. The left mouse button rotates the model, the center mouse button zooms the model in and out, and the right mouse button moves the model up, down, right, and left. In addition, there are icons for fixed standard orientations. These orientations can also be accessed with a single click of the right mouse button in the display window. The **Angles** tab in the **Controls** window gives the user fine control over the view orientation. Finally, the user can toggle perspective on or off.

With the **Display Settings** tab under the **Controls** window, the user can turn on and off the point cloud, elements, and/or grids. The sizes of the point cloud and grids can also be adjusted here. The user can also change the display temperature limits. All display settings are updated when the user presses the **Refresh Graphics Window** button. Additionally, the user may select a dark background to aid in providing contrast, if necessary.

If there are multiple time sets, the user can view any of these sets. A **Time Set Selection** tab will appear under the **Controls** menu, and the user can then select from a pull-down menu (Figure 2-6).

When the user is done with the viewer, the window can be closed by clicking on the window **X** button.

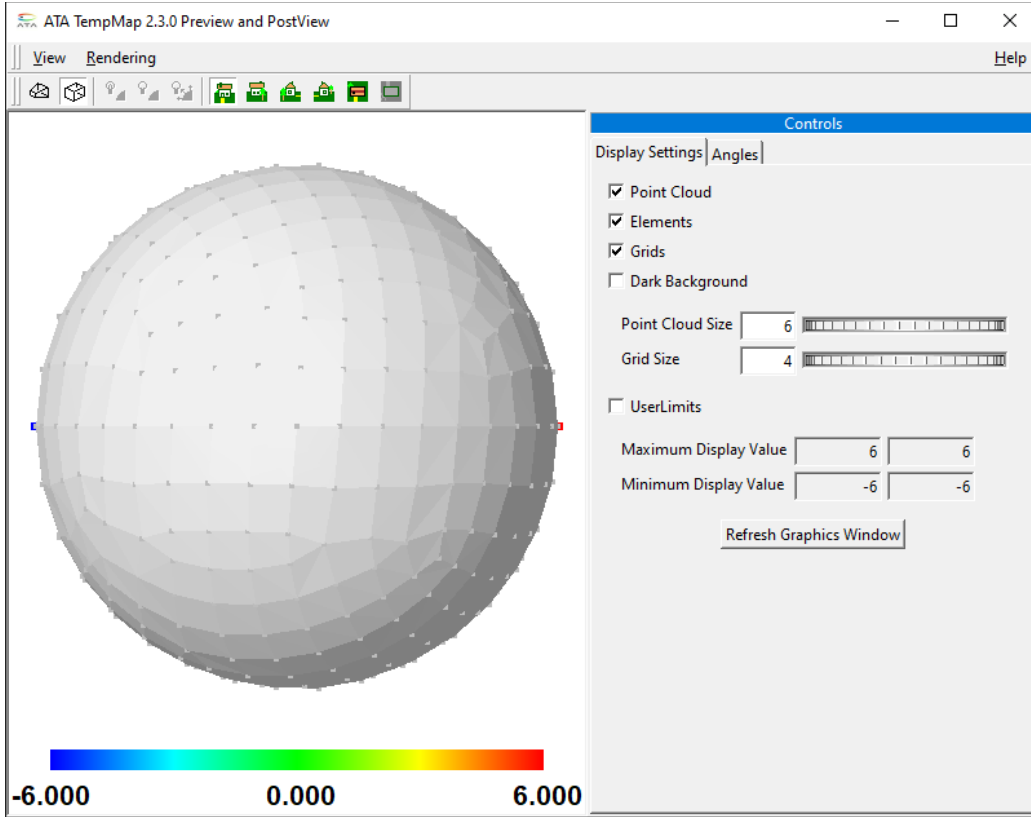


Figure 2-4. TempMap Preview display.

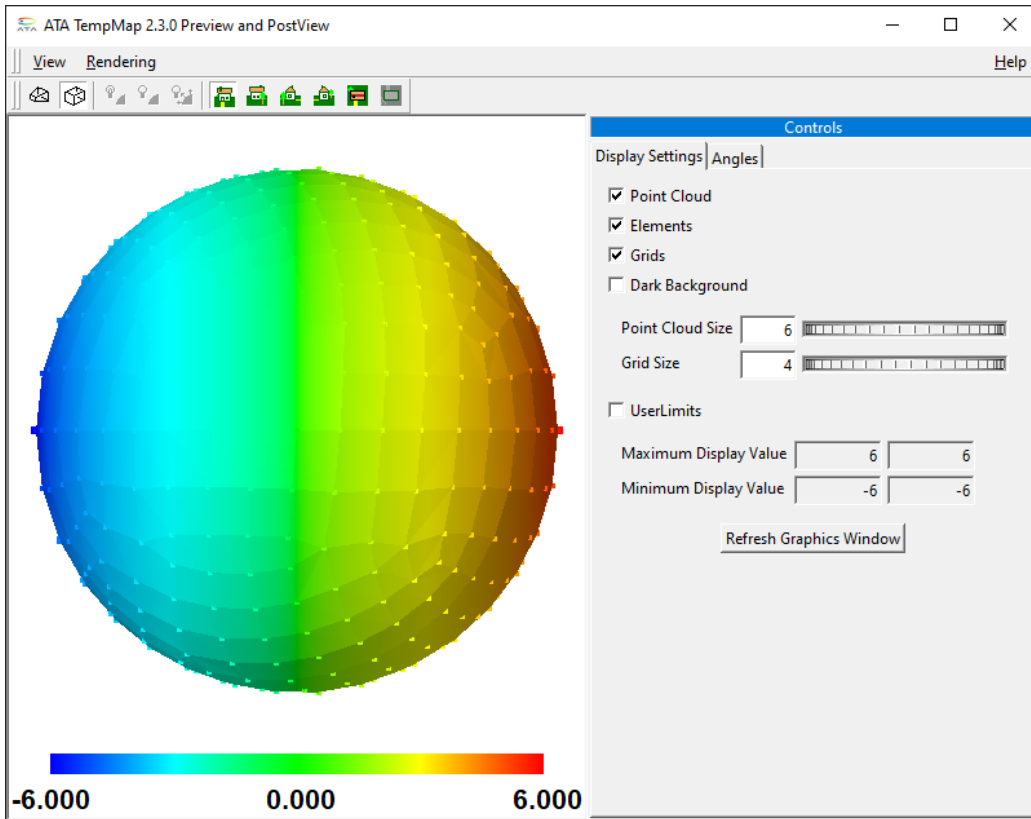


Figure 2-5. TempMap PostView display.

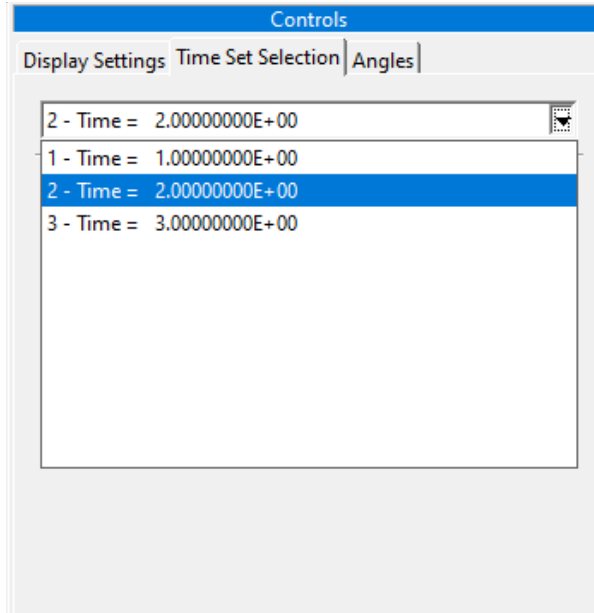


Figure 2-6. Selection of time set for cases with multiple time sets.

3. **INSTALLATION**

TempMap uses a client-server licensing scheme. You must therefore install and configure the Sentinel RMS license server (section 3.2) separately from TempMap itself (section 3.3). If you have other ATA software, the license server is already installed and you will only need to obtain a specific license file for TempMap to make it run.

3.1. **Platform Requirements**

TempMap supports 64-bit Windows platforms.

TempMap is compatible with all versions of Siemens I-deas and Siemens NX Nastran, and with MSC.Nastran v2001 or later.

3.2. **Installing the License Server**

Sentinel RMS is a robust, commercial client-server-based licensing system from SafeNet that can serve multiple licenses for multiple software products simultaneously. The server typically resides on a central computer while client software such as TempMap resides on the computers that will be utilizing the software. The client computer may or may not be the same as the license server. It can also be on a different software platform than the client.

Sentinel RMS installation is straightforward. Detailed installation instructions are shipped with the Sentinel RMS package available from ATA's website at <http://www.ata-e.com/software/rmsserver/>. This document highlights the basics of the installation process.

3.2.1. **Installing Sentinel RMS.**

On Windows, the Sentinel RMS package is an InstallShield application. As such, it must be installed by someone with administrator privileges. ATA recommends that the default selections be used during the installation.

3.2.2. **Environment Variable.**

All Sentinel RMS clients need a way to determine where the server is running. By default, the client will scan the subnet your client system is connected to and will identify any Sentinel RMS servers running on your subnet. It will then contact each until the license request is satisfied or it has run out of servers. You can control the order in which the servers are contacted through an environment variable called `LSHOST`. Set the environment variable to the hostname of the server (or servers) and separate each license server name from the next by a tilde (~). The server name must be prepended by the name 'no-net'. For example, set it to

```
no-net~server1~server2
```

where `server1` is the name of the first license server, `server2` is the name of the second license server, and so on.

If you only have one server and wish to bypass the network scan by the client, you can use the environment variable `LSFORCEHOST`. You can specify a single server with this environment variable. For example, you can set it to

```
server
```

On Windows, the environment variable can be set in the Control Panel, typically in System. Please ask your system administrator or refer to your operating system documentation for more details on how to set environment variables.

3.2.3. Checking Your License Status.

You can use the RMS License Administration application `WlmAdmin`, included in the top-level TempMap directory, to check the status of the licenses. When the GUI opens, open up Subnet Servers in the tree on the left. Under each license server, the individual licenses will be listed. Clicking on a license will display statistics about that license, including who is using it and how many are available. You can also install licenses using this application.

Prior to using TempMap, you must contact ATA to obtain a valid license file for it. To obtain a permanent license, you will need to run the `echoid.cmd` batch file that comes with the Sentinel RMS download and send the resulting `echoid.txt` file to software@ata-e.com. If you have permissions issues running this file from the installation location, you can copy the `echoid.*` files to a directory where you have write access and run it from there.

3.3. **Installing TempMap**

To install TempMap, run `TempMap_v*_Setup.exe`. TempMap will install itself into the `C:\Apps\TempMap_v###` (where `###` is the TempMap version number) directory by default.

3.4. **Configuring TempMap**

The default configuration that installs with TempMap should be sufficient for most people, so you should not have to do any additional configuration. However, if you have a nonstandard environment or wish to fine-tune how TempMap works, you may need to configure it. All possible changes will be made in the TempMap launch script.

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, the mapping program assumes that 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 time steps or cases.
5. **OUTTIMES**. Optional. Allows for multiple output time steps.
6. **DYNAMIC**. Optional. Prints out data for a transient analysis. Only the keyword **DYNAMIC** is needed. The keyword **DYNAMIC** is ignored for I-deas.
7. **TEMPERATURE**. Required. Coordinates of the points and the temperature at these points. At least one point is required.
8. **CYL**. Optional. Sets the coordinate system as cylindrical. Otherwise, the data are 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 is 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 five 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, four lines are necessary. The first line contains the keyword **COORD** as its first five characters, and the next three 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:

$$\vec{X}_{FEM} = \begin{bmatrix} \vec{X} \\ \vec{Y} \\ \vec{Z} \end{bmatrix} \vec{X}_{TEMP} + \vec{X}_0$$

where

\vec{X}_{TEMP} = point coordinates in the temperature model coordinate system

\vec{X}_{FEM} = point coordinates in the FEM coordinate system

$$\vec{X} = \frac{\vec{x}_1}{|\vec{x}_1|}, \vec{Y} = \frac{\vec{x}_2 \times \vec{x}_1}{|\vec{x}_2 \times \vec{x}_1|}, \vec{Z} = \vec{X} \times \vec{Y}$$

$$\vec{X}_0 = (X0, Y0, Z0), \vec{X}_1 = (X1, Y1, Z1), \vec{X}_2 = (X2, Y2, Z2)$$

4. For the **SYMM** block, three lines are necessary. The first line contains the keyword **SYMM** as its first four characters, and the next two 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:

$$\vec{X}_{PSYMM} = \vec{X}_P - 2[(\vec{X}_P - \vec{X}_0) \cdot \vec{X}_1] \vec{X}_1$$

where

\vec{X}_P = original temperature point coordinates

\vec{X}_{PSYMM} = coordinates of symmetric temperature point

$$\vec{X}_1 = \frac{\vec{x}_1}{|\vec{x}_1|}, \vec{X}_0 = (X0, Y0, Z0), \vec{X}_1 = (X1, Y1, Z1)$$

5. For the input time data, at least two lines are needed. The first line contains the keyword **TIMES** as its first five characters, the next line contains the number of time values, and the third line and beyond contain the time values in ascending order, with four values per line, except for the last line, which can have one to four 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 two lines are input, the time values will default to 1, 2, 3, etc. For instance:

```
TIMES
10
```

will generate times 1 to 10.

- For the output time data, at least three lines are needed. The first line contains the keyword **OUTTIMES** as its first eight characters, the next line contains the number of time values, and the third line and beyond contain the time values in ascending order, with four values per line, except for the last line, which can have one to four 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 or less than the input times will be ignored.

- For the temperature data, at least two lines are needed. The first contains the keyword **TEMPERATURE** as its first eleven 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 are 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 TH_i are theta 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 four values per line, except for the last line, which can have one to four 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
```

Please note that the records are free-format; the indentations and adherence to columns shown here are used 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 the first record in the file.
2. **COORD.** Optional. Defines transformation from the temperature model coordinate system to FEM coordinate system. If not present, the mapping program assumes that 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 time steps or cases.
5. **OUTTIMES.** Optional. Allows for multiple output time steps.
6. **DYNAMIC.** Optional. Prints out data for a transient analysis. Only the keyword **DYNAMIC** is needed. The keyword **DYNAMIC** is ignored for I-deas.
7. **GRIDS.** Required. Defines point IDs and coordinates of the points. At least one point is required.
8. **CYL.** Optional. Sets the coordinate system as cylindrical. Otherwise, the data are in Cartesian coordinates. If this card is used, it must be on the first non-comment line following **GRIDS**.
9. **TEMPS.** Required. Defines point IDs and the temperature(s) at these points. At least one is required. The numbers and values of all of the point IDs must match for each point ID defined by the **GRIDS** keywords.

For the **GRIDS** and **TEMPS** data, at least two lines are needed for each. The first contains the keyword **GRIDS** as its first five 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 five 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 are 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 TH_i are theta 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 four values per line, except for the last line, which can have one to four 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 a **#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
```

then TempMap will read both files and concatenate them together.

Appendix B. **Command-Line and Batch File Operation**

TempMap also has the ability to be run as a command-line program. To activate the command-line option, the user types *tempmap -t* in the **Command Prompt** window. The user will then answer the prompts as they appear on the screen.

This command-line option allows users to run TempMap in batch mode. This can be done by creating an input file **input.txt** that contains the responses to the TempMap queries (see below).

With a properly formatted input file, batch mode can be executed by typing *tempmap -t < input.txt* in the **Command Prompt** window.

The format of the **input.txt** files is:

Line 1. Input file format. **1 = I-deas (default), 2 = NASTRAN**

Line 2. Output file format. **1 = I-deas (default), 2 = NASTRAN**

Line 3. Input structural file name.

Line 4. Input temperature file name.

Line 5. Output file name. **blank = accept default**

Line 6. Input temperature units. **1 – Celsius (default), 2 – Kelvin, 3 – Fahrenheit, 4 – Rankine**

Line 7. Output temperature units. **1 – Celsius (default), 2 – Kelvin, 3 – Fahrenheit, 4 – Rankine**

Line 8. Temperature load set title. **blank = accept default**

Line 9. Additional options. **0 = no additional options (default), 1 = additional options**

If line 9 is blank or 0:

Line 10. Input load set. **blank = accept default of lowest numbered load set**

Line 11. Output load set. **blank = accept default of N+1, where N is the highest numbered load set**

Line 12. Additional cases to process? **0 = No (default), 1 = Yes**

If line 9 is not blank or 0.

Line 10. Mapping option. **0 = temperatures on surface (default), 1 = volumetric temperatures**

Line 11. Merge tolerance for pressure data. **blank = accept default of 1E-8**

Line 12. Radius tolerance multiplier. **blank = accept default of 2**

Line 13. Number of near temperature to use, **blank = accept default of 12**

Line 14. Log file options. **1 = minimum (default), 2 = detailed, 3 = verbose**

Line 15. Print mapping weights? **0 = No (default), 1 = Yes**

Line 16. Use 2-D element normals? **0 = No (default), 1 = Yes**

Line 17. Special options depending on output file format.

I-deas: no special operation; press **Enter** to continue

NASTRAN: NASTRAN precision? **0 = Single Precision (default), 1 = Double Precision (single), 2 = Double Precision (multi)**

Line 18. Input load set. **blank = accept default of lowest numbered load set**

Line 19. Output load set. **blank = accept default of N+1, where N is the highest numbered load set**

Line 20. Additional cases to process? **0 = No (default), 1 = Yes**

If there are additional cases to process, repeat lines 1 to 12 or 1 to 20.

An example **input.txt** file is shown below. It performs two cases. The first case is I-deas to I-deas without advanced options. The second case is Nastran to Nastran with advanced options.

```

1
1
example1.unv
temperature.txt
temperature.unv
1
1
Example 1 Temperature Mapping, I-deas to I-deas
0
1
2
1
2
2
example2.blk
temperature.txt
temperature.blk
1
1
Example 2 Temperature Mapping, NASTRAN to NASTRAN, advanced options
1
1
1.E-8
5
12
1
0
0
0
1
2
0

```

Appendix C. **Revision History for Older Releases**

Version 1.2.3.

Bug Fixes

- Fixed error related to creation of four-point interpolation by ensuring that the four interpolation points are nearly planar.

Version 1.2.2.

Enhancements

- Added command prompt capability.

Version 1.2.1.

Enhancements

- TempMap now errors only on 0% overlap; otherwise, it issues a warning.

Bug Fixes

- Fixed error in finding duplicate temperature points.

Version 1.2.0.

Enhancements

- Mapping improvements for quad shapes
- Mapping improvements for near edge
- Use of element normal is now "off" by default.

Bug Fixes

- Fixed bug in triangulation that caused fatal error.

Version 1.1.0.

Enhancements

- Added preview and postview capability.
- Added additional advanced options, allowing more flexibility in the mapping.

Version 1.0.4.

Bug Fixes

- Fixed 3D to local 2D projection error.

Version 1.0.3.

Enhancements

- Improved mapping algorithm for sparse temperature data by including local 2D Delaunay triangulation of nearby temperatures.

Version 1.0.2.

Bug Fixes

- Releases license when **X** is pressed on GUI window.
- Prints only first instance of node with no mapped temperature, which prevents the GUI text window from being overloaded with messages and crashing.

Version 1.0.1.

Enhancements

- Included a check for duplicate grid ID numbers when reading temperatures.

Bug Fixes

- Fixed documentation errors.

Version 1.0.0.

- Initial release.
