



Release Notes

Version 09.26.05b Build Initial



Table of Contents

INSTALLATION	2
NEW AFFTAC USERS	2
EXISTING AFFTAC USERS	2
NOTE TO ALL XP USERS	2
NEW FEATURES & BUG FIXES	3
REPOSITIONING OF INPUTS	3
BUG: USUM = 0 CAUSES COMPUTATIONAL CODE CRASH.....	3
BUG: INTERMITTENT CASES OF COMPUTATIONAL CODE NOT FINISHING OR SPURIOUS ERROR MESSAGES	3
BUG: FAILURE OF INTERNAL NUMERICAL TEST	4
MODIFICATIONS	4
REPOSITIONING OF INPUTS	4
BUG: USUM = 0 CAUSES COMPUTATIONAL CODE CRASH.....	4
BUG: INTERMITTENT CASES OF COMPUTATIONAL CODE NOT FINISHING OR SPURIOUS ERROR MESSAGES	4
BUG: FAILURE OF INTERNAL NUMERICAL TEST	5
TESTING	5

Prepared by: Dr. Scott R. Runnels
Scott Runnels Consulting
www.srconsult.com

September 28, 2005

Installation

New AFFTAC Users

Install AFFTAC from the installation package by following the steps below:

1. Unzip the installation zip file.
2. Close all Windows applications.
3. Run setup.exe, taking note of the **installation directory**.
4. If you are an existing user, you may copy your database files (*.adb) from the previous versions' installation folder to the new **installation directory**.

Existing AFFTAC Users

If you already have AFFTAC installed, simply install the GUI by saving its .exe file over the existing one in your AFFTAC installation directory.

Important Note: The consideration of discontinuities and the fraction of spray-on insulation coverage have been moved from the insulation database into the analyses. The values you have input for those aspects of the simulation must be set again: Double click the analysis in the Main Window, click **Next** in the Analysis Conditions window, **Next** in Tank Car Properties window, and then reset the values in the Insulation Choices window.

If you have problems or concerns e-mail srunnels@srconsult.com.

Note to All XP Users

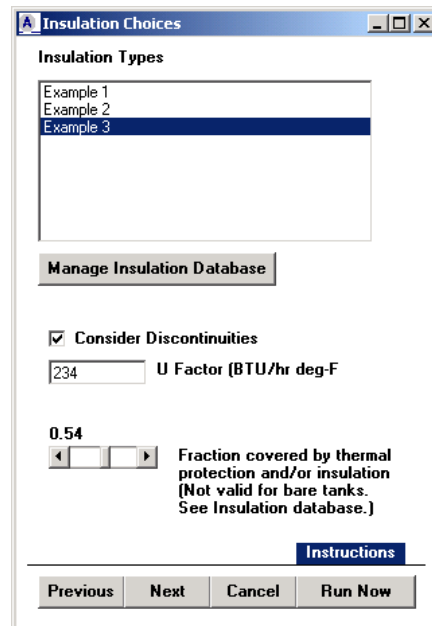
Please unzip the installation file manually:

1. Download the file.
2. Right click it and choose "Extract."
3. Move into the directory containing the extracted files and run setup.exe.

New Features & Bug Fixes

Repositioning of Inputs

The “Consider Discontinuities” and “Spray-on insulation coverage fraction” inputs were moved from inside the insulations database up to the analyses inputs level. So now the user can vary these two inputs without having to modify the insulation database. They are now located in the Insulation Options window, as shown below:



Bug: $U_{sum} = 0$ causes computational code crash

This bug was discovered during the above modification and is due to a formatting error, which caused the variable U_{sum} , when equal to zero, to be printed as “.” only.

Bug: Intermittent cases of computational code not finishing or spurious error messages

Occasionally, clicking “Run Now” in an editing sequence window or on one of the first two tabs in the Main Window would cause the computational module data to never be displayed in the GUI or spurious error messages to be displayed.

Bug: Failure of internal numerical test

For some combinations of insulation choices, an internal numerical verification test would fail. This failure was caused by inconsistent application of the `InsulationID` variable in the computational module. The internal test that compares the original `SurfaceT` predictions to the newer Newton-Raphson predictions would fail because of that inconsistent application.

Modifications

Repositioning of Inputs

1. Moved the GUI functionality for the `Idsc`, `Usum` and `SprayonInsulation` values out of the `EditInsulation` form and into the `InsulationOptions` form.
2. Moved `Idsc`, `Usum`, and `SprayOnCoverage` into the `UserInput` pseudo-class. Note that in order to make AFFTAC read previous insulation database entries, the inputs for these variables in the `Insulation` class were replaced with dummies (e.g., for backwards compatibility).
3. Introduced new database I/O version identifier, `Version=2004_002`. Added the `IO_To_TS_Version_2004_002` routine so that the `UserInput` pseudo-class will read `Idsc`, `Usum`, and the `SprayOnCoverage` data.
4. Added code that resets `DBVersion` to `Version` on output.

Bug: Usum = 0 causes computational code crash

1. Made format statement change in the `Insulation.MakeComputationalInputFile` routine.

Bug: Intermittent cases of computational code not finishing or spurious error messages

1. Added pause loop in the `Main1.Timer1_Timer` routine to help ensure it does not start reading the computational code's error log before the computational code is still writing to it.
2. Added a pause loop in `RunCurrentInputs` to give some buffer time between the C++ and Fortran computational module executions.

3. Removed `SSTab1_Click(2)` call in `Main.EditCurrentInputs` because setting `SSTab1.Tab = 1` in the line above it already calls a `_Click` function implicitly, which caused double execution of the computational modules.

Bug: Failure of internal numerical test

Modified the `SurfaceT_test` routine to apply a consistent usage of `InsulationID`, as follows:

```
if (InsulationID != 3 && InsulationID != 6)
{
    // No jacket
    VF_r = FireViewFactor*0.48*Emissivity;  VF_l = 0.48*Emissivity;
}
else
{
    // Has jacket
    VF_r = 0.48*Emissivity/(2.0-Emissivity);  VF_l = 0.48*Emissivity/(2.0-Emissivity);
}
```

Testing

GUI:

1. Installed fresh copy onto a separate PC.
2. Imported old db files to ensure backwards compatibility.
3. Ran new version of the GUI with identical inputs and then compared computational input files to ensure the same data was being transmitted.
4. Ran GUI, modified values, read into database and checked for persistence using several combinations of mouse actions.

Computational C++ Module:

1. Successfully ran against all regression tests, requiring no changes to the standards files.
2. Successfully ran against test case that previously exhibited failure of the internal verification test in `SurfaceT`.