

EESy Solutions

Engineering Equation Solver Newsletter

No. 20, Spring 2010

Welcome

This is the 20th issue of **EESy Solutions**, a newsletter developed to provide news, tips, and tricks relating to Engineering Equation Solver. EES has been a commercial software product for over 20 years. Did you miss any of the previous issues? The previous issues are available from our web site: www.fchart.com.

Instant Update Service

EES uses a different model for updating than most other commercial programs. Each time there is a change in the EES program, either to correct a problem or to add a new feature, the version number is incremented by 0.001 and the latest version is placed on our website. Although the program has become very robust and stable, there have been 200 new versions released since the last EESy Solutions was composed in Spring 2009. Any user who has current subscription to our Instant Update Service can download the latest version. All new licenses of EES are provided with one year of Instant Update Service. The fee to continue Instant Update Service after the first year is 20% of current cost of the program per year. Contact F-Chart Software if you wish to re-subscribe to Instant Update Service.

What's New?

New features are added to EES very carefully. A concern is that the program will become so feature-loaded, it will be difficult to use, which is a complaint that sometimes is directed to Microsoft products. Another concern is stability. Any change in the program introduces the possibility of errors and unwanted interactions. Over the years, EES has become very stable, and we wish to keep it that way. On the other hand, there are many ways that the program can be improved. We have been particularly responsive to users who have provided suggestions for improvements. If you have a suggestion for an enhancement, send it to info@fchart.com.

Some of the new features that have been added in the last year are as follows.

NASA Ideal Gas Data

Thermodynamic data are internally provided for 24 ideal gases. The NASA property program (McBride et al, NASA/TP-2002-211556, 2002) provides property data for 1262 ideal gases. These data are accessible in EES through a call to the NASA external program. However, the NASA ideal gases can now be used in EES property functions in exactly the same manner as for the built-in ideal gases. The online help provides the list of the 1262 gases. As an example, the following EES statement returns the specific entropy of ideal gas N₂O₄.

```
s=entropy(N2O4,T=600 [K],P=400 [kPa])
```

Additional Property Data

High accuracy property data have been added for methanol, parahydrogen, Siloxane_1 (hexamethyldisiloxane) Siloxane_2 (octamethylcyclotetrasiloxane). Property data have been improved for nitrogen and normal hydrogen. The range of applicability for hydrogen has been extended from original limits of 500 K and 40 MPa to 1000 K and 2000 MPa.

Lithium Bromide Water Properties

New property correlations have been implemented for lithium bromide-water solutions based on the publication by Patek, J. and Klomfar, J., Int. J. of Refrigeration, Vol 29, pp. 566-578, (2006). These property correlations are provided in the LiBrH₂O.lib external EES program. Information and examples for these functions are available by selecting Function Info from the Options menu, clicking EES library routines and then selecting LiBrH₂O.lib and clicking the Function Info button. The older LiBr.DLL routines are no longer distributed.

Improved Warnings

A major effort has gone into improving the warnings that EES provides. EES provides warnings for a variety of purposes. For example, EES will display a warning if a property

correlation is used outside of its range of applicability. This situation can occur during the calculation process as EES iterates to find the solution to the equations, even if the final solution is within the range of applicability of the correlations. The Warning dialog has been revised to show the last warning issued for each equation. If the warning occurred during the iteration process, but not in the final solution, it is shown in gray type. Updating the guess values normally removes warnings of this type and also reduces the time required for calculation. Moving the mouse cursor above any of the warnings in the Warnings dialog will result in the equation that caused the warning being selected in the Equations window. Many of the warnings have been revised to make them more understandable. Careful attention to the Warnings dialog will result in your program being more robust and calculating results more quickly.

UnitSystem\$

UnitSystem\$ is a new string function that takes one argument, which can be 'Temperature', 'Pressure', 'Energy', 'Entropy', 'Volume', or 'Trig'. The function returns a string indicating the units that EES is configured to operate in for this specification. For example, if EES is configured to operate in kJ units on a mass basis, the following equation,

```
E$=UnitSystem$('energy')
```

will set E\$ to 'kJ/kg'. The major use of this function is to check the units supplied by a user in a function that you have written. Combined with the UnitsOf\$ function, which returns the units of the variable provided as the argument, this UnitSystem\$ function can be used to provide explicit warnings or error messages relating to units. The warning and error messages can be displayed with the Warning and Error procedures

Highlight in Color

A highlight tool button has been added in the Professional version below the menu bar to allow highlighting text in the Equations window in a choice of seven colors. Colorizing the text makes it easy to find.

LibHuAirProp

An external property library for real moist air, steam, water and ice has been developed by Prof. Han-Joachim Kretzschmar and his colleagues in the Department of Technical Thermodynamics at the Zittau/Goerlitz University of Applied Sciences in Germany. The external library is provided as an EES-compatible DLL. It installs into the EES UserLib folder and is thereby available whenever EES is started. Versions of the library are provided for both English and SI units. It is a professionally prepared software product with excellent online support. If you are interested in high accuracy property data for air-water mixtures, this library may be just what you are looking for. Contact Dr. Kretzschmar hj.kretzschmar@hs-zigr.de or through his website at www.thermodynamics-zittau.de.

Textbooks

We have often been asked if there exists an EES textbook. The 310 page EES manual is not a textbook, but it does provide a great deal of general information on using EES. The current English manual and older versions in French and Spanish can be downloaded from the F-Chart website <http://www.fchart.com>.

Another useful source of information on how to use EES is provided in the *Heat Transfer* textbook by G.F. Nellis and S.A. Klein that was published in December, 2008 by Cambridge University Press. Information about this book is on <http://www.cambridge.org/nellisandklein>. Although there are other textbooks that use EES, this is the first textbook that directly integrates EES with the engineering content. The book includes many detailed EES example problems that solve a variety of heat transfer problems from start to finish. In addition to demonstrating heat transfer principles, the examples illustrate many capabilities and best practices for using EES. The heat transfer libraries that are now provided with EES were developed in conjunction with this textbook. A thermodynamics textbook is currently being prepared.

Tips and Tricks

Among the capabilities that EES provides, the ability to check equations for unit consistency is among the most important. Unit checking is even more important when it is done in a Function, Procedure or Subprogram that may be used as a library program. This issue will demonstrate some capabilities in EES to simplify checking and saving units in Functions, Procedures and Subprograms.

An EES Subprogram is a standalone EES program that can be called from the main EES code or from a Function or Procedure. Information is passed to and from the Subprogram through the arguments provided in the Call statement or with a \$Common directive. EES checks the units for the Equations in a Subprogram in the same manner as it does for the main EES code.

An EES Function returns a single result. An EES Procedure is like a Function, but it can return more than one result and it is accessed with the Call statement. Both Functions and Procedures employ assignment statements, rather than equalities as used in the main body of EES. Assignment statements assign the value on the left of the equal sign to the result of the expression on the right side of the equal sign. All of the variable on the right side of the equal sign must have values or an error will be issued. The programming required for Functions and Procedures is very similar to that used in modern computer languages such as C, FORTRAN, Pascal, and others. Anyone familiar with EES will realize quickly that it takes much more effort to obtain a desired result with assignment statements than with the powerful equalities provided in the main body of EES and in Subprograms. However, the advantage of Functions and Procedures is that they process the equations in the order in which they are entered and as a result, they can use logic statements, such as If-Then-Else and Repeat-Until. They also allow variables to be reassigned. For example, it is perfectly legal to set $x=x+1$ in a Function and a Procedure, but this statement would generate an error in a Subprogram or in the main body of EES.

To demonstrate the unit checking possible in a Function, let's write a simple function that

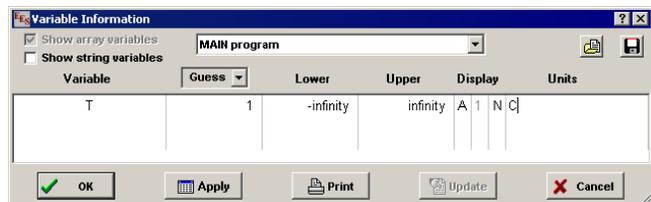
converts a temperature in °F to °C. This function is not really needed since EES provides this capability with the ConvertTemp library function, but we will use this simple function to demonstrate how to check the units supplied to the function and set the units of the return value. Here's the function. It's very simple.

```
Function TC(TF)
  TC=(TF-32 [F])*5 [C]/9 [F]   "convert F to C"
end
```

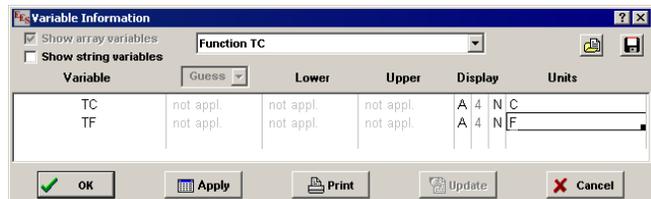
Test the function with the following statement in the main EES code.

```
T=TC(77 [F])
```

EES should return $T=25$, which is the equivalent temperature in °C. However, two additional capabilities should be added to this function to make it more useful. First, it should set the units of the return value of TC and second, it should check to ensure that the value of TF provided to the function is in °F. To set the units of the variables in the function, select Variable Info from the Options menu. It will open showing the variables in the Main program, which is just T in this case. Set the units for T to C.



Click the Apply button. Then click on the drop-down list at the center top of the dialog to choose the variables for Function TC. Enter units of C for variable TC and units of F for variable TF.



Click OK. When you solve this program, EES should correctly display that $T=25$ [C] and no unit consistency issues. This is how units are set in functions. The units for variables in Procedures and Subprograms are set in the same manner. It is good practice to always set the units of all variables appearing in the program.

If the TC function is called with a variable that does not have units of F, EES will issue its standard unit consistency checking warning.

OK. So you've written an EES Function, (or a Procedure or Subprogram) and you entered all of the units. You tested it and it works great. Now, you want to use this function in another EES program. If you just copy the function as text and paste it into another program, it will work but all of the unit information that you entered will be gone. This information is not copied to the Clipboard with the text. Of course, if your Function is as simple as the one considered in the example, you could simply reenter the units in the new program. But in general, you will have more complicated routines and nobody wants to reenter information.

There are several alternative ways to save the units in the routine and use them in another program. The simplest alternative is to save the file that contains the Function(s) in a *.lib. (Replace * with the name of the file.) Note that the *.lib file could contain more than one Function as well as Procedures and Subprograms. However it is best to delete the equations in the Main equations window that follow after the last routine before saving the *.lib file.

The *.lib file could be used directly as an external library. If the *.lib file is saved in the EES UserLib folder, it will be automatically loaded when EES is started. Also, all of the units that were set for the variables in the Functions, Procedures, and Subprograms are saved with the equations in the *.lib file and the unit checking will operate as it did in the first program. If the *.lib file is saved in a location other than the UserLib folder, it can be loaded using the Load Library command in the File menu or using a \$Include directive.

Suppose you do not wish to use your Function as an external file, but rather just transfer it with its units to another EES file. This can be accomplished in two ways. The easiest way is to use the Merge command in the File menu. First, place the cursor at the top of the file where the Function will be merged. Then select the Merge command to merge the *.lib file. The Function will appear and the units for the variables will be set as they were in the original EES program.

Another alternative available in the Professional version is to use *.var files. A *.var file saves information about the variables in a selected Function, Procedure, Subprogram or an entire program. The saved information includes the name of the variable, its guess values and limits (not applicable for Functions and Procedures), its units and its display characteristics. To save the *.var file, choose the Variable Info command from the Options menu. Next select the routine for which the variable information will be saved. If you select the Main program, then all of the variables for all routines and the main program will be saved. Click the Save Var File icon at the upper right of the dialog menu. A standard Windows dialog will appear asking you where you wish to save the file. Complete the file name and the save process is done.

To later use the *.var file, click the Open Var File icon that appears to the left of the Save Var File icon at the upper right of the Variable Info dialog. A standard Windows dialog window will appear, allowing you to select the *.var file that you wish to load. After clicking OK, all of the variable information for existing variables will be replaced with the information in the file. Using *.var files in this manner not only transfers the units of the variables, but the guess values and limits as well. This capability is a great way to backup the variable information in any EES program.