



Nocturnal Aviation Software

FTranEdit User Manual

v.2.0

Table of Contents

| | |
|------------------------------------|-----------|
| Features | 1 |
| What is FTranEdit? | 1 |
| Installing FTranEdit | 2 |
| Setting up FTranEdit | 3 |
| General Preferences | 3 |
| Editor Preferences | 3 |
| Code Preferences | 4 |
| Comment formatting | 4 |
| Templates | 4 |
| Using FTranEdit | 6 |
| Basic Editing | 6 |
| The Editor Window | 6 |
| Using FTranEdit's Special Features | 7 |
| The FTranEdit Menus | 12 |
| The FTranEdit Menu | 12 |
| The File and Edit Menus | 12 |
| The Format Menu | 12 |

| | |
|------------------------------|-----------|
| The Insert Menu | 12 |
| The Window Menu | 13 |
| The Help Menu | 13 |
| Appendix:Legal | 14 |
| Sparkle Framework | 14 |
| CocoaFob | 17 |
| CodexFab | 18 |
| FsprgEmbeddedStore.framework | 18 |

Features

What is FTranEdit?

FTranEdit is a stand-alone editor for Fortran programmers. It works exactly like the source code editor built into Nocturnal Aviation Software's FTranProjectBuilder, but with the ability to open and tile multiple source files, if desired.

FTranEdit is like any other Mac text editor, but with features added to make Fortran programming faster and more efficient. It adds:

- User configurable syntax highlighting of Fortran keywords, intrinsic functions and more,
- Comment/uncomment blocks of code,
- Auto upcasing of Fortran keywords, if desired,
- User editable templates for procedures and control structures and more,
- Tab markers to jump to marked spots in the code,
- Auto alignment of code,
- User configurable auto formatting of comment blocks,
- One command matching of procedure names at beginning and end,
- One command insertion of a real_kind type,
- Autocomplete parentheses,
- Autocomplete variables and procedures,
- XCode-like popup menu that shows the current procedure being edited, selectable to jump to another,
- Status bar showing current line, total number of lines, current line length, current column and length of selection,
- Tiling of multiple windows on the screen.

Installing FTranEdit

FTranEdit installs as a standard Mac application. Simply drag and drop it to the applications folder. To uninstall, simply drag it to the trash.

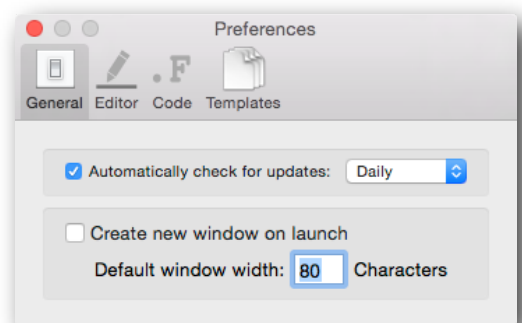
Setting up FTranEdit

Go to the preferences window, FTranEdit > Preferences..., to select your defaults, set up your window and text coloring, coding style preferences and templates.

General Preferences

The choices in the General Preferences pane are

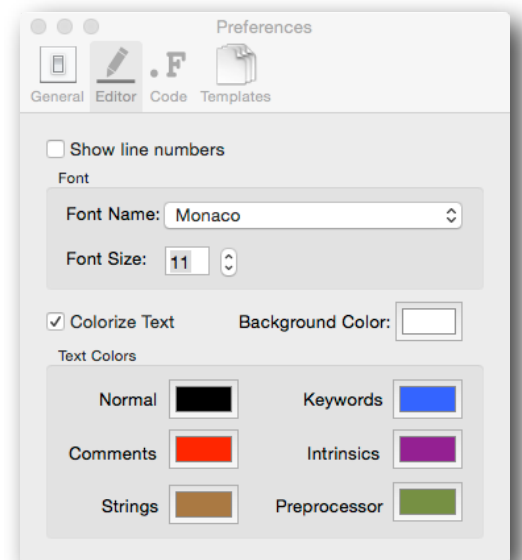
- **Automatically check for updates.** Select this if you want FTranEdit to go online to find updates. Choose the frequency in the pulldown menu. If you don't, you can update manually from the FTranEdit application menu.
- **Create new window on launch.** Select this if you want a blank document to open when you start FTranEdit.
- **Default window width.** Set the character width of the window to be wide enough to show a full line without wrapping. (Fortran 90+ limits lines to 132 characters, Fortran 77 to 80.)



Editor Preferences

Use the **Editor pane** to set the appearance of text in the window. The options are:

- **Show line numbers:** Select this option if you want line numbers to be shown to the left of your source code.
- **Font:** Choose the font name and size for all Fortran source code. You should choose a fixed width font like Monaco or Courier for best results.
- **Colorize Text:** Most people like their source code colored. If you don't, deselect this option.
- **Background Color:** Choose the background color for your source code.
- **Text Colors:** Customize the colors of source code elements here.

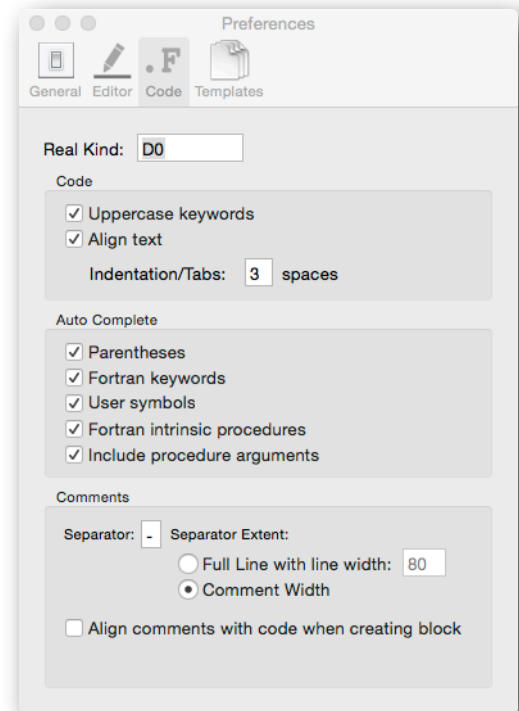


Code Preferences

Use the Code preferences pane to set up how you like your code to look. FTranEdit will help you to format your code so that you can make it look the way you like it quickly and easily.

The code box lets you set standard formatting options:

- **Real Kind:** If you define your own real kind and don't want to type it every time you define a variable, set the name that you use here. Then a command-I will insert your real kind with one keystroke.
- **Uppercase keywords:** Fortran is case insensitive. Many people like to have keywords in the program to be upper case. When you select this option, Fortran keywords will be switched to upper case as you type, unless they are within a comment.
- **Align text:** Select this option to have a new line of code indented to the same number of spaces as the line above it.
- **Indentation:** This sets the number of spaces the Shift Left (command-[)/Right (command-]) will shift selected lines.
- **Auto complete parentheses:** Select this option if you want a right parenthesis automatically inserted when you type the left one. This will also put balanced parentheses around a selection when you type a left parenthesis.
- **Auto complete Fortran keywords:** Select this option if you want keywords like CHARACTER, etc. to be automatically completed for you.
- **Symbols, Procedures, Arguments:** Select these options if you want autocompletion of all the procedures in your file and/or Fortran intrinsic procedures. If you select "Include procedure arguments", the arguments and placeholders will be included. If you are programming in F77 primarily, or use implicit interfaces a lot (no modules/interfaces), you will not want to include the procedure arguments.



```

OPEN( UNIT= fUnit, FILE = plotFileName )
CALL ExportToTecplot( thePlotter, sem )
ExportToTecplotI( this = !^, N = !^, sem = !^ )
CLO ExportToTecplotNoI( this = !^, sem = !^ )
CAL ExportToTecplot(this = !^, sem = !^ )
CAL ExportToTecplot( this = !^, N = !^, sem...
END EXPONENT(X = !^ )

```

~~~~~



Files are indexed for their symbols after a save. If completions for recently created variables or procedures are not being shown, save your file before trying again.

## Comment formatting

One feature that FTranEdit has that you won't find in other editors is the ability to format comments with a single keystroke. You set how you want comment blocks to look in the Comments box in the Code preferences pane. See the section *Formatting* in the chapter *Using FTranEdit* on how to format sections of text as comment blocks.

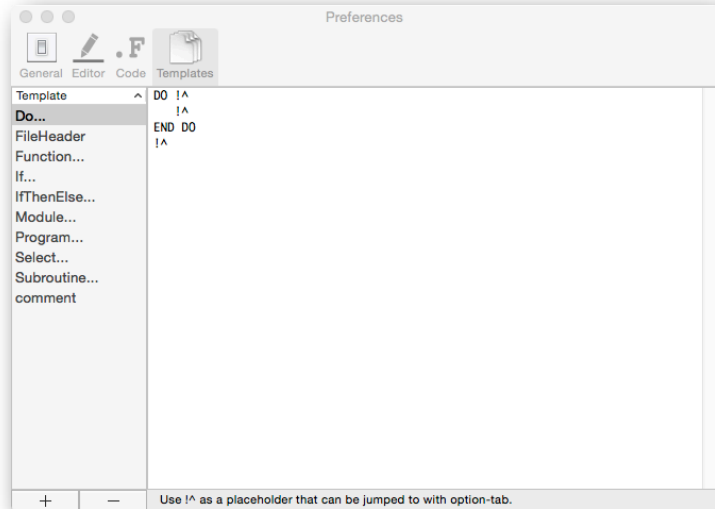
## Templates

FTranEdit allows you to define and modify templates to insert commonly used code blocks quickly. You modify and define templates in the Templates preferences pane.

When you select the Templates preferences pane you are presented with a list of predefined templates for code blocks like DO loops or subroutines. You can edit these to look the way you like them. Insert !^ to mark a placeholder. (In the

editor, typing <option-tab> will move you from one marker to the next.) You invoke a template by typing the first part of the template name, e.g. Sub, and type escape <esc>. (If you have autocompletion of procedures enabled and a procedure completion completion is brought up, hit <esc> twice.) A list of options (if any, e.g. if/end if, if/then/else) will appear. Select the option you want with the arrow keys and type a return, enter or tab. The template will be inserted and aligned ready for you to option-tab to the first marker.

You can create your own templates by using the (+) button at the bottom of the table. Name your template and type the text in the text view on the right. The templates can contain any text you like. Delete templates with the (-) button.





# Using FTranEdit

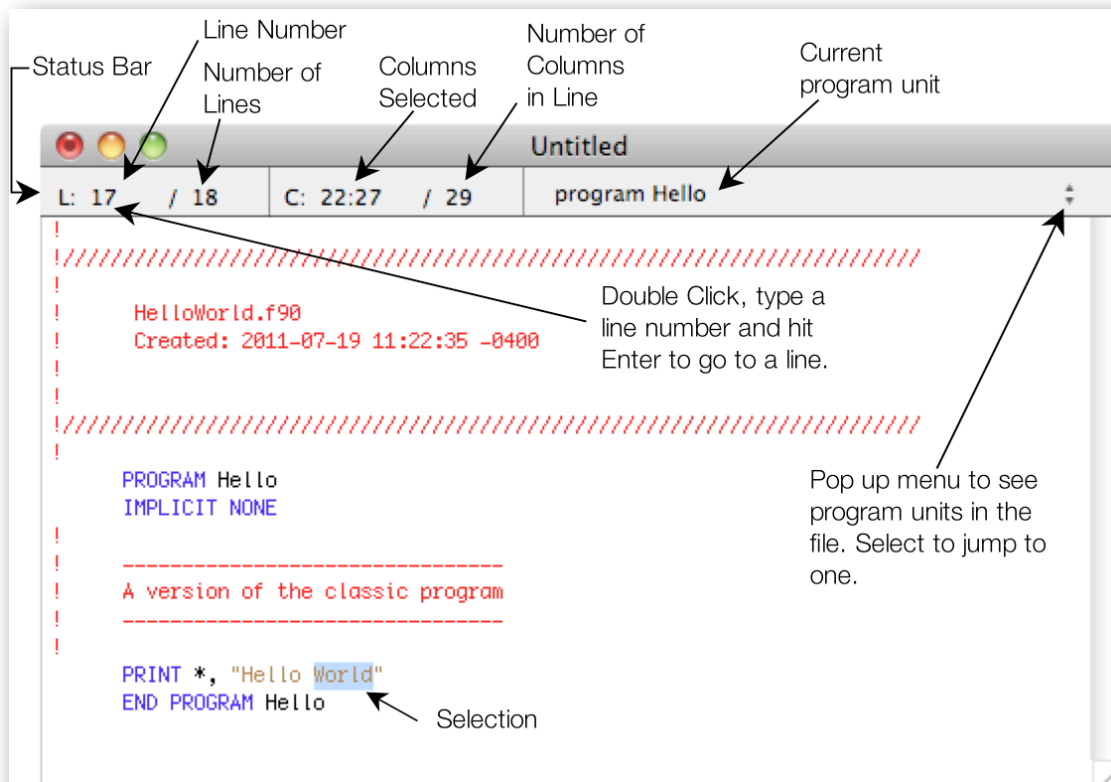
## Basic Editing

FTranEdit works like any other Mac text editor with special features to make writing Fortran code as fast and easy as possible. It implements cut/copy/paste/undo like any other Mac program. A few basic additions are

- Double clicking on a parenthesis will select the text between it and its counterpart.
- Type option-tab to jump from the current location to the next jump marker, !^ . FTranEdit beeps if there isn't one.

## The Editor Window

The editor window has a number of useful features, shown below:

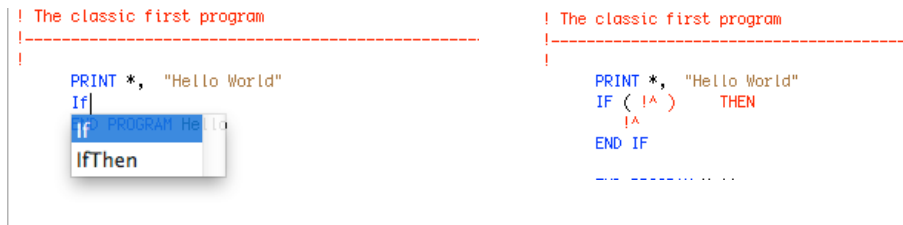


The Status Bar tells you which program unit/subroutine/function you are editing at any given time. To jump to another simply use the popup menu.

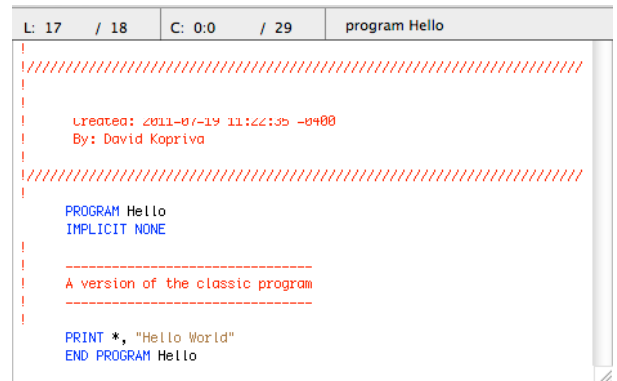
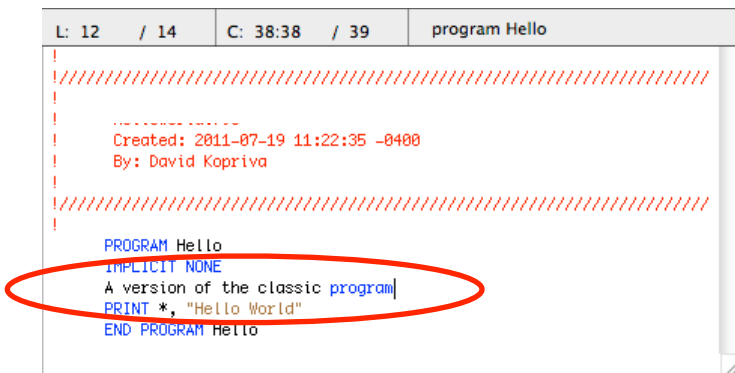
## Using FTranEdit's Special Features

FTranEdit has these features to make coding go faster:

- Templates:** You insert a template that is defined in the templates pane of the preferences window by typing the first few letters of the template and then hit the escape button, e.g. type pro<esc> for the PROGRAM template. This will bring up a popup menu from which you select the template that you want to insert. Use the arrow keys if there is more than one. ( E.g. If, If-Then-Else). Type <return>, <tab>, or <enter> and the template is inserted, ready for you to fill in. Type <option-tab> to jump to the first insertion point, if any. If you have procedure autocompletion enabled, hit <esc> twice to enter the template completion mode.

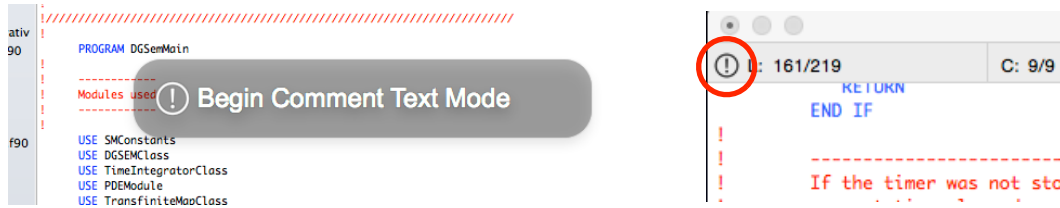


- Formatting.** Formatting is done from the Format menu. You can comment out or uncomment a selection of code. You can shift lines of code left or right to align them as you like. To make commenting fast and easy you can create a comment block automatically. Simply type your comment, select the text you want blocked off, and type Command-Y. See the section on Preferences to find the options available for the formatting.



If you have FTranEdit automatically formatting your code, like upcasing and coloring keywords, you can quickly turn that feature off by switching to comment text mode (option-command-C). When you enter this mode, a message will flash on the screen and a marker will be placed at the upper left of the editor area to remind you.

You exit the comment text mode by typing option-command-C again. To save time, it is automatically exited when you use one of the comment formatting commands.

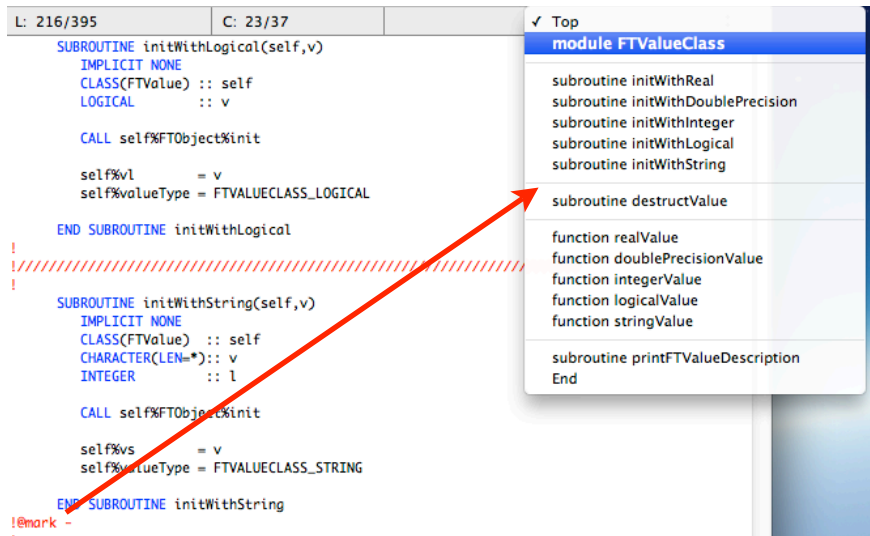


- Insert Program Unit Name.** Fortran likes its program units (PROGRAM, FUNCTION, SUBROUTINE, MODULE, plus INTERFACE and TYPE ) to be named - at the beginning *and at the end*. If you change the the name of a procedure or module, you normally would need to (remember to) scroll down to the corresponding END statement and type or paste the name in. FTranEdit can do that automatically for you with a single command. Simply place the cursor in or select the procedure name and hit Command-U to have the program unit name inserted (or replaced) at its END statement.

- Program Unit Popup Menu Formatting:** You can add separators to organize the program unit popup menu. Simply insert a line

!*@mark* -

in your source code and a separator is placed in the menu.



## Autocompletion

Autocompletion speeds up your coding by making suggestions for what you are typing. Great if you use long names. With autocompletion enabled, suggestions will be made after a short pause (1/2 second) in your typing. Simply select the suggestion that you want and hit <tab> to insert. To ignore the suggestions, keep typing, or hit <delete>.

```

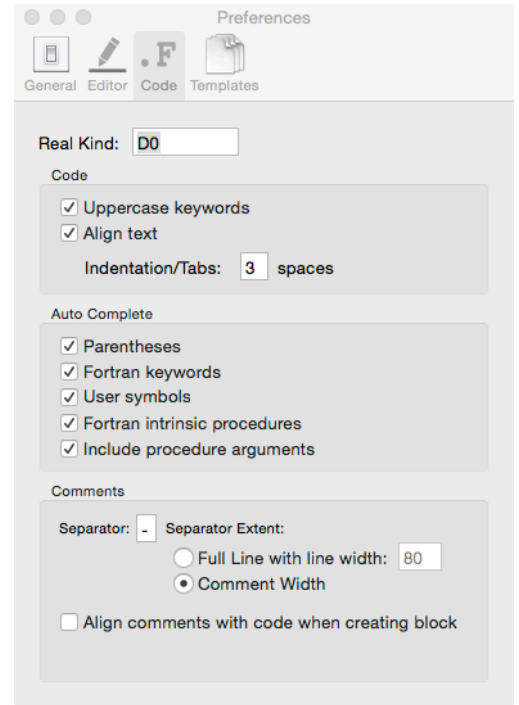
OPEN( UNIT= fUnit, FILE = plotFileName )
CALL ExportToTecplot( thePlotter, sem )
ExportToTecplotI( this = !^, N = !^, sem = !^ )
CLO ExportToTecplotNol( this = !^, sem = !^ )
CAL ExportToTecplot(this = !^, sem = !^ )
CAL ExportToTecplotI( this = !^, N = !^, sem...
END EXPONENT(X = !^ )
    
```



FTranEdit lets you choose between two procedure completion formats. The first, seen above, inserts the procedure name and the arguments. Typing option-tab moves you between the jump markers “!^”. With this format, you never have to remember the full name and argument lists of your Fortran’s procedures.

Named arguments are only available in Fortran with an explicit interface, so if your code is F77, or you do not have your functions in modules/interfaces, you cannot use this form for the argument list. In such cases, you can set the preferences to choose to not include the argument lists, and functions will appear only with a single insertion marker, “(!^)”.

You set the format in the Code preferences. You can selective enable or disable the autocompletion of Fortran keywords, intrinsic procedures, and your own procedures. Select Include procedure arguments to choose the format that the completions will take.



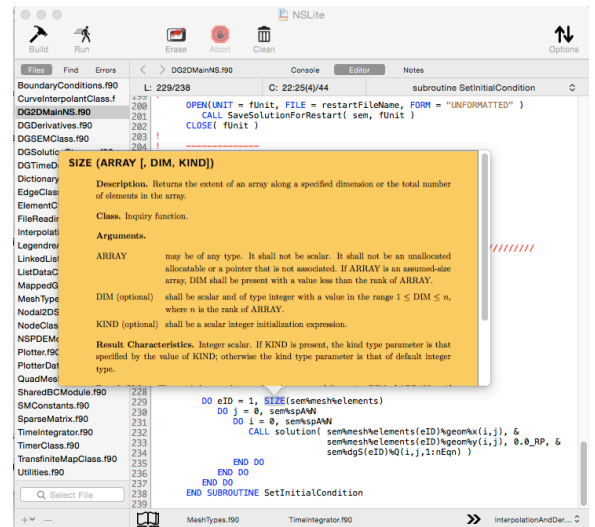
Finally, for those who use templates a lot, hit <esc> twice to access templates that start with the same characters as a keyword or procedure. The first escape gets you out of completion mode, the second brings up the suggestions for your templates.

Files are indexed for their symbols after a save. If completions for recently created variables or procedures are not shown, save your file before trying again.



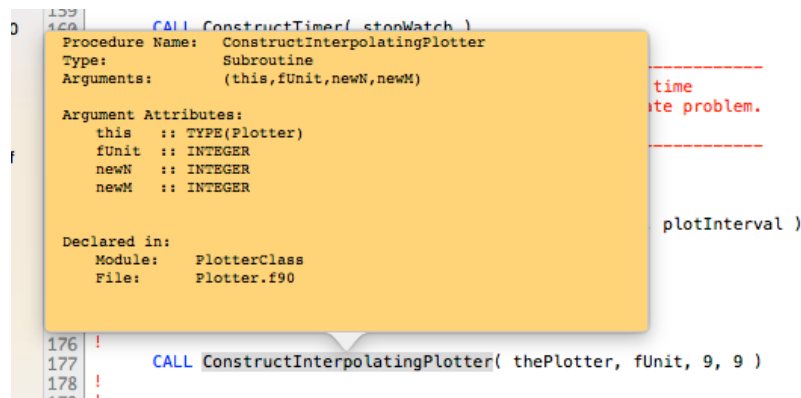
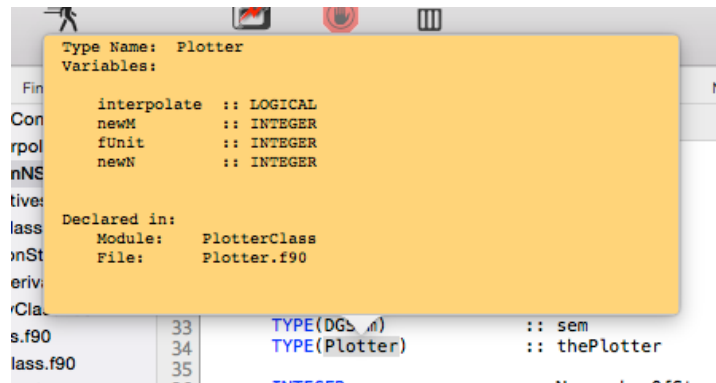
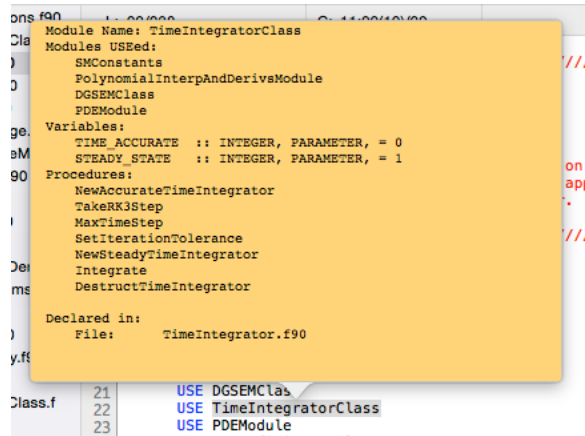
## Intrinsic Procedure Specifications

You can get a popover window with information about Fortran intrinsic subroutines and functions by selecting the name and choosing Show Info... (command-option-I) found in the popup menu shown by right-or command-clicking the selection. As with any standard Mac popover, drag it to create a standalone window that can be resized and moved wherever you like.



## User Symbol Specifications

You can also get a summary of your own symbols -- variables, procedures and modules -- by invoking the Show Info... (command-option-I) found in the popup menu shown by right- or command- clicking a selection. The summary will depend on what type of symbol is selected. See below for examples. Note that, unlike in FTranProjectBuilder, you can only see information about symbols defined in the file being edited.

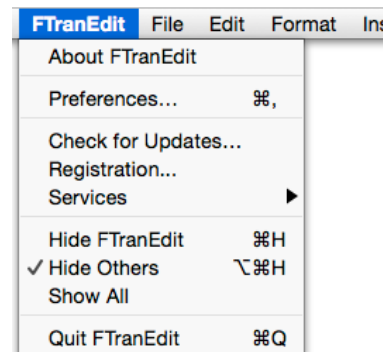




# The FTranEdit Menus

## The FTranEdit Menu

The program menu contains the usual menu items with two additions. You can manually check for updates to FTranEdit by selecting the Check for Updates... item. If you want FTranEdit to check automatically at a specified interval, select the Preferences... menu item and check the appropriate boxes. See “Setting Up FTranEdit” for more details. You can also check your license registration here.



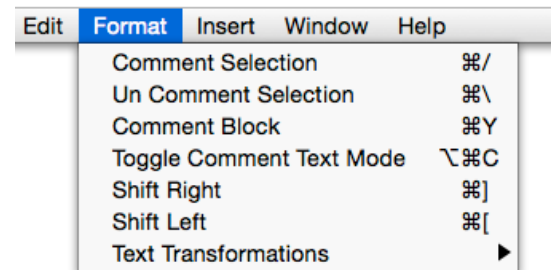
## The File and Edit Menus

The File and Edit menus contain the usual Mac operations.

## The Format Menu

The Format menu is used to format text being edited by the source code editor. The items are:

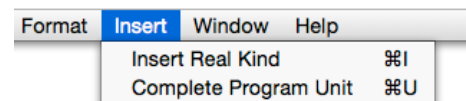
- Comment/Un Comment Selection:** Select a line or lines in a source code file and chose one of these to insert or remove a “!” at the beginning of each line.
- Comment Block:** Use this command to format a selection as a comment block. See the “Source Code Editor” chapter for details.
- Toggle Comment Text Mode:** Toggle automatic formatting on and off to support typing comments.
- Shift Left/Right:** Select a line or lines in a source code file and choose one of these to move the lines left or right. The number of spaces in the shift is set in the Code preferences.
- Text Transformation:** Convert a selection to upper case, lower case, or capitalize.



## The Insert Menu

The insert menu contains two items:

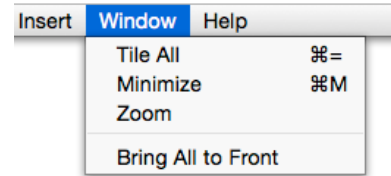
- Insert Real Kind:** The first menu item inserts the Real Kind parameter that you define in the Code preferences. (The default is D0.) For speed, you are most likely going to use the keyboard equivalent, command-I.
- Complete Program Unit:** Place the cursor in the Program/Subroutine/Function/Module name and select this menu item to have the name



automatically inserted after the program unit's End statement. Great for when you change the name of a procedure and don't want to scroll down to change it at the bottom.

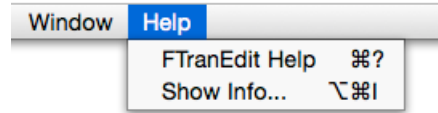
## The Window Menu

The Window menu adds a tiling command to the usual, Apple supplied items. Selecting Tile All will resize your editor windows and tile them across the screen. If there are more windows than can fit, given the column width of a window that you set in the General Preferences, two or more rows of windows will be created.



## The Help Menu

This is a standard Mac help menu plus the Show Info... command.





# Appendix: Legal

## Sparkle Framework

Copyright (c) 2006 Andy Matuschak

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

### EXTERNAL LICENSES

=====

Thanks to Brent Simmons, whose RSS reader class was modified and used for appcast support. His class is licensed as follows:

Copyright (c) 2002, Brent Simmons

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\*

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\*

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\*

Neither the name of ranchero.com or Brent Simmons nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT

SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This project uses software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org>). This toolkit is licensed as follows:

```
/* =====  
* Copyright (c) 1998-2005 The OpenSSL Project. All rights reserved.  
*  
* Redistribution and use in source and binary forms, with or without  
* modification, are permitted provided that the following conditions  
* are met:  
*  
* 1. Redistributions of source code must retain the above copyright  
* notice, this list of conditions and the following disclaimer.  
*  
* 2. Redistributions in binary form must reproduce the above copyright  
* notice, this list of conditions and the following disclaimer in  
* the documentation and/or other materials provided with the  
* distribution.  
*  
* 3. All advertising materials mentioning features or use of this  
* software must display the following acknowledgment:  
* "This product includes software developed by the OpenSSL Project  
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"  
*  
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to  
* endorse or promote products derived from this software without  
* prior written permission. For written permission, please contact  
* openssl-core@openssl.org.  
*  
* 5. Products derived from this software may not be called "OpenSSL"  
* nor may "OpenSSL" appear in their names without prior written  
* permission of the OpenSSL Project.  
*  
* 6. Redistributions of any form whatsoever must retain the following  
* acknowledgment:  
* "This product includes software developed by the OpenSSL Project  
* for use in the OpenSSL Toolkit (http://www.openssl.org/)"  
*  
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY  
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR  
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR  
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED  
* OF THE POSSIBILITY OF SUCH DAMAGE.  
* =====
```

## Nocturnal Aviation Software

\*  
\* This product includes cryptographic software written by Eric Young  
\* ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)). This product includes software written by Tim  
\* Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).  
\*  
\*/

### Original SSLeay License

/\* Copyright (C) 1995-1998 Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com))  
\* All rights reserved.  
\*  
\* This package is an SSL implementation written  
\* by Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)).  
\* The implementation was written so as to conform with Netscapes SSL.  
\*  
\* This library is free for commercial and non-commercial use as long as  
\* the following conditions are aheared to. The following conditions  
\* apply to all code found in this distribution, be it the RC4, RSA,  
\* lhash, DES, etc., code; not just the SSL code. The SSL documentation  
\* included with this distribution is covered by the same copyright terms  
\* except that the holder is Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).  
\*  
\* Copyright remains Eric Young's, and as such any Copyright notices in  
\* the code are not to be removed.  
\* If this package is used in a product, Eric Young should be given attribution  
\* as the author of the parts of the library used.  
\* This can be in the form of a textual message at program startup or  
\* in documentation (online or textual) provided with the package.  
\*  
\* Redistribution and use in source and binary forms, with or without  
\* modification, are permitted provided that the following conditions  
\* are met:  
\* 1. Redistributions of source code must retain the copyright  
\* notice, this list of conditions and the following disclaimer.  
\* 2. Redistributions in binary form must reproduce the above copyright  
\* notice, this list of conditions and the following disclaimer in the  
\* documentation and/or other materials provided with the distribution.  
\* 3. All advertising materials mentioning features or use of this software  
\* must display the following acknowledgement:  
\* "This product includes cryptographic software written by  
\* Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com))"  
\* The word 'cryptographic' can be left out if the rouines from the library  
\* being used are not cryptographic related :-).  
\* 4. If you include any Windows specific code (or a derivative thereof) from  
\* the apps directory (application code) you must include an acknowledgement:  
\* "This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com))"  
\*  
\* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND  
\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
\* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE  
\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
\* SUCH DAMAGE.

\*

\* The licence and distribution terms for any publically available version or  
\* derivative of this code cannot be changed. i.e. this code cannot simply be  
\* copied and put under another distribution licence  
\* [including the GNU Public Licence.]  
\*/

## CocoaFob

Written by Gleb Dolgich

Twitter: @glebd

Web: <<http://pixelespressoapps.com>>

CocoaFob is distributed under the BSD License

<<http://www.opensource.org/licenses/bsd-license.php>>

Copyright &copy; 2009-2011, PixelEspresso. All rights reserved.

Redistribution and use in source and binary forms, with or without modification,  
are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list  
of conditions and the following disclaimer. Redistributions in binary form must  
reproduce the above copyright notice, this list of conditions and the following  
disclaimer in the documentation and/or other materials provided with the  
distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED  
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR  
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON  
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS  
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### # Credits

[0] The Mac developer community that continues to amaze me.

[1] Base32 implementation is Copyright (C) 2007 by Samuel Tesla and comes from  
Ruby base32 gem: <<http://rubyforge.org/projects/base32/>>. Samuel Tesla's blog is  
at <<http://blog.aliengo.com/tag/base32/>>.

[2] OpenSSL key generation HOWTO: <<http://www.openssl.org/docs/HOWTO/keys.txt>>

[3] Handling URL schemes in Cocoa: a blog post by Kimbro Staken  
<<http://www.xml-databases.org/WK/blog/1154?t=item>>

[4] Registering a protocol handler for an App: a post on CocoaBuilder mailing  
list, <<http://www.cocoabuilder.com/archive/message/cocoa/2009/2/2/229297>>

[5] PHP implementation courtesy of Sandro Noel, <<http://gesosoft.com>>

## CodexFab

Credits:

-----

This project includes code from the following open source projects:

[1] CocoaFob Copyright 2009 by Gleb Dolgich <<http://github.com/gbd/cocoafof/tree/master>>

[2] MSZLinkedView by Marcus Zarra <<http://www.cimgf.com/2008/03/03/core-animation-tutorial-wizard-dialog-with-transitions/>>

License:

-----

CodexFab LicenseExample is Copyright © 2009 MachineCodex Software.  
CodexFab LicenseExample is released under a Creative Commons Attribution 3.0 License.  
<<http://creativecommons.org/licenses/by/3.0/>>

## FsprgEmbeddedStore.framework

MIT License: [http://en.wikipedia.org/wiki/MIT\\_License](http://en.wikipedia.org/wiki/MIT_License)

Copyright (c) 2010 FastSpring, LLC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.