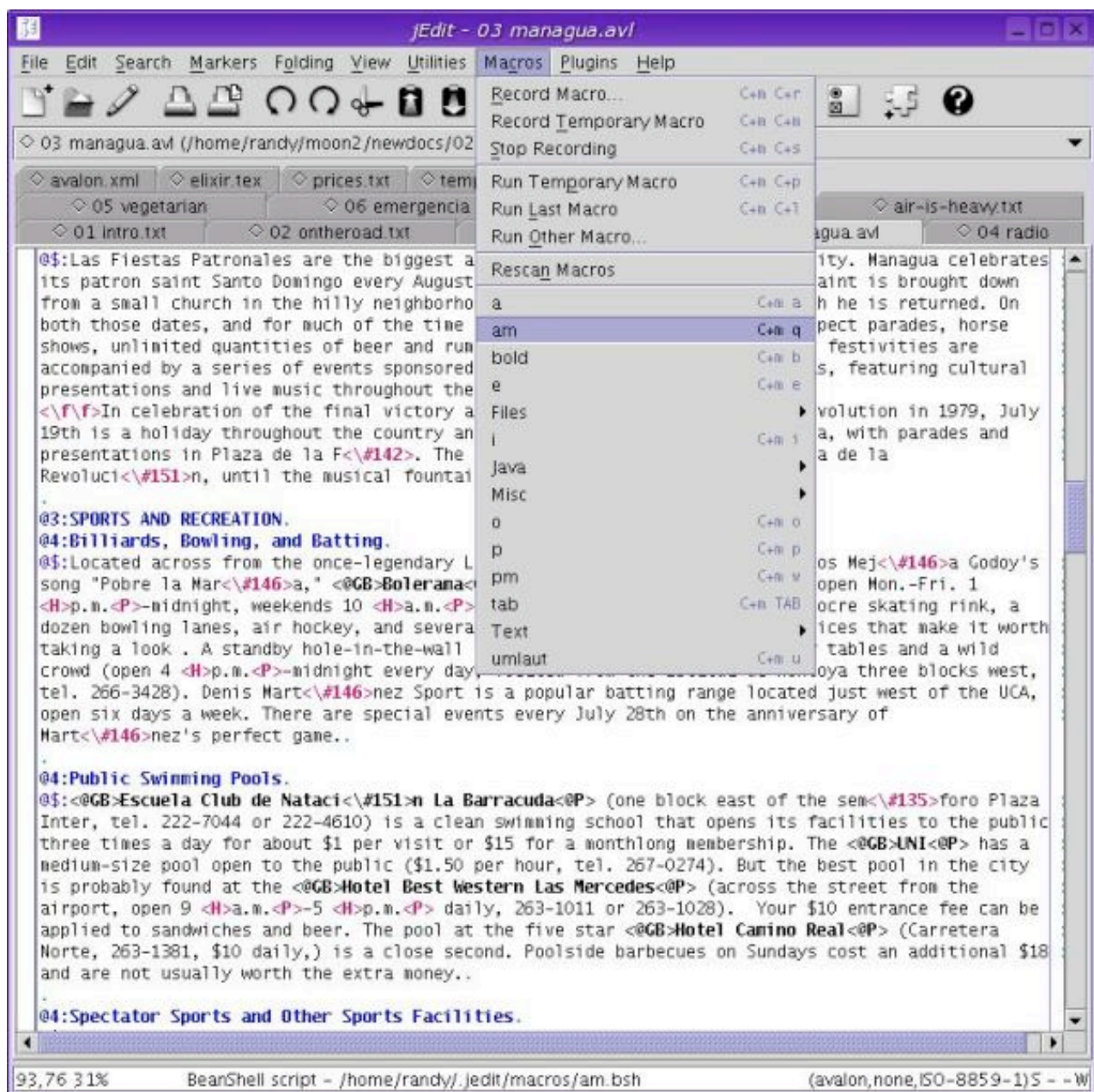




# The Woodnotes Guide to Using Jedit to Code Manuscripts for Avalon Travel Publishing

by Randall Wood, co-author of Moon Nicaragua and Moon Living Abroad Nicaragua

This document was last updated 30 September 2006. It can be found permanently at [www.therandymon.com](http://www.therandymon.com).



## Introduction

I'd just begun to work on the 2nd edition of the manuscript for Moon Nicaragua when I had a thought: working with Avalon's codes was no different than writing HTML code or computer programming, something I have a bit of experience with. And I use some rather powerful software to make dealing with programming issues easier, so why not harness it to facilitate working with Avalon codes?

The above is a regular old document I've been editing for Moon:Nicaragua 2nd edition, but instead of using a regular old word processor, I'm using a piece of software that makes the cumbersome process of coding documents for Avalon a bit easier. In the image above you can see some of the features that ease the pain of coding Avalon documents, including syntax highlighting that makes it easier for me to see the codes (and see if I've screwed up), programmable macros for inserting codes like `<\f>` with a single keystroke, and multiple buffers for working on and searching/replacing in lots of chapters at a time. Read on, and writing documents for Avalon Travel Publishing will be easier for you too.

In a nutshell, here's how Jedit will help you develop your manuscripts for ATP:

- Work easily on multiple documents all at once, including things like search and replace across all the documents, or just the documents you choose.
- Simple or complex search and replace allow you to do things like checking for outdated telephone area codes, replacing diacritical marks, and more.
- Simple macros allow you to input Avalon codes at a key stroke.
- Complex macros allow you to do multiple things at once. For example, I have one macro that searches for and replaces all diacritical marks with Avalon codes and another that
- Syntax highlighting allows you to see your codes in a way that stands out, get an idea of what the final text will look like, and spot errors in codes before they affect your manuscript.
- Multiple buffers allow you to work with lots of documents (chapters, scraps, notes, etc.) at once and in a convenient manner.
- Finally, all you need to access all this functionality is this manual and the files I've already created for you. The software is free of charge. You are exactly ten minutes away from being more productive.

## Text Editors vs. Word Processors

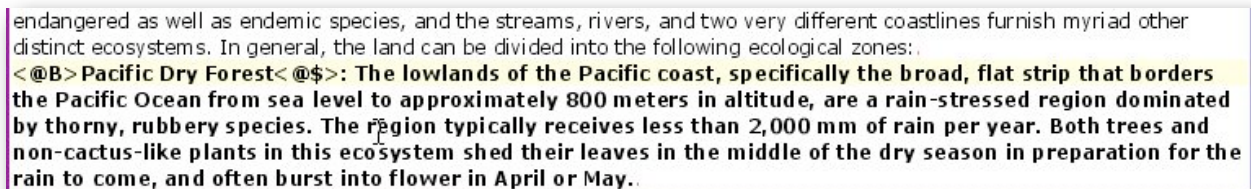
Jedit is a text editor, not a word processor. The difference is that text editors don't deal with formatting (bold, italics), tables, inserted graphics, and the other features that word processors do with ease; nor do they deal with page breaks, margins, and so on. But ATP manuscripts don't require any of those things, and formatting requires Avalon's codes, so using a text editor instead of a word processor is no loss.

Text editors deal with plain text and nothing else, but put at your fingertips powerful functions that simplifies dealing with text and markup code like Avalon's. If, when you

think of text editors you think of "Microsoft Notepad" or "Macintosh TextEdit" you're on the right track, but these rudimentary, featureless text editors don't do many useful tricks. On the other hand, the text editors that computer programmers utilize come with powerful functionality that might be useful to the casual ATP author.

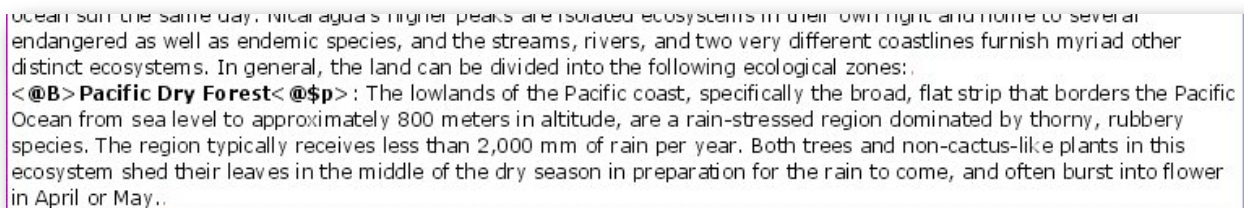
One of those text editors is called Jedit. Jedit provides the following tools that make the complex job of editing documents for Avalon less painful:

1. **Syntax highlighting:** Jedit recognizes patterns of text and highlights them appropriately. Unlike word processors, the highlighting is an illusion, a programmer's aid that makes coding easier. It makes the text easier for you to read but doesn't change the document itself. If, using your word processor, you take a sentence and italicize it, that formatting is hardwired into the document. But if, using a text editor, you instruct it to highlight every use of the expression `<#151>` your text editor will colorize those expressions just to make it easier for you to read, without changing the document itself. The advantage to Avalon authors is that this will also show you if you start a code but don't finish it, or type it incorrectly. Because Jedit looks for specific patterns, if you botch a code, it will be immediately obvious.



endangered as well as endemic species, and the streams, rivers, and two very different coastlines furnish myriad other distinct ecosystems. In general, the land can be divided into the following ecological zones:  
<@B>Pacific Dry Forest<@\$>: The lowlands of the Pacific coast, specifically the broad, flat strip that borders the Pacific Ocean from sea level to approximately 800 meters in altitude, are a rain-stressed region dominated by thorny, rubbery species. The region typically receives less than 2,000 mm of rain per year. Both trees and non-cactus-like plants in this ecosystem shed their leaves in the middle of the dry season in preparation for the rain to come, and often burst into flower in April or May..

In the image above I've made a mistake in the `<@$P>` code (Moon handbook series) and as a result the bolding continues through the rest of the paragraph, drawing attention to the mistake.



ocean sun the same day. tropical agua's higher peaks are isolated ecosystems in their own right and home to several endangered as well as endemic species, and the streams, rivers, and two very different coastlines furnish myriad other distinct ecosystems. In general, the land can be divided into the following ecological zones:  
<@B>Pacific Dry Forest<@\$p>: The lowlands of the Pacific coast, specifically the broad, flat strip that borders the Pacific Ocean from sea level to approximately 800 meters in altitude, are a rain-stressed region dominated by thorny, rubbery species. The region typically receives less than 2,000 mm of rain per year. Both trees and non-cactus-like plants in this ecosystem shed their leaves in the middle of the dry season in preparation for the rain to come, and often burst into flower in April or May..

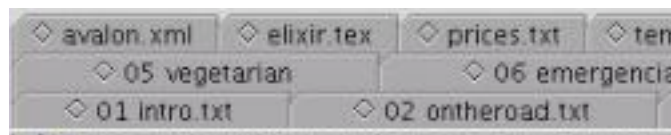
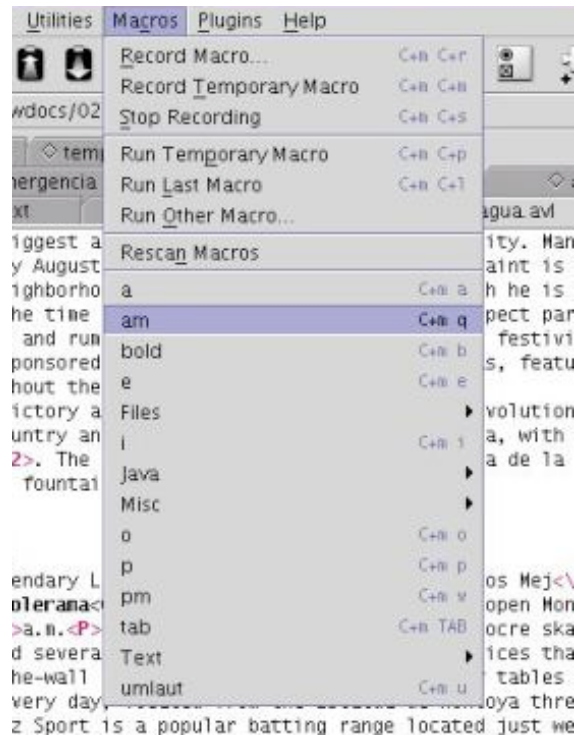
As soon as I fix the code, the highlighting reflects the correct bolding.

2. **Complex windowing capacity:** Jedit allows you to divide the workspace into two vertical panes, showing two different parts of the same document, two horizontal panes, both, or even two separate frames showing one or more documents. This flexibility helps you look at your document in any way that makes sense to you. When comparing chapters, I like having the two docs side by side to see if I've written a longer history in one than another. Similarly, it's convenient to have the main document on the left and the must-see callout on the right to make sure the text matches up.

3. **Very powerful search and replace functions:** you can search for words, patterns, tags, and use regular expressions to extend the search capacity. Using the multiple windows feature (see below) you can search and replace in the current document, all currently open documents, or even an entire directory. Imagine updating to a new code system in all documents that comprise your book, in one keystroke, or doing a global search and replace on all your documents to update a telephone area code! The incremental search capability makes it easy to go right to a certain section of document without lifting your fingers from the keyboard: efficient!

4. **Programmable macros:** I use macros to insert tags and accented characters (control-tab to automatically insert <\f>, for example), but you can record any macro you like and assign it a shortcut key. You can make new macros just by starting and stopping the macro-recorder as you work. Using this process I made a macro that searches for and replaces all characters with diacritical marks with the appropriate Avalon code, all in one simple click.

5. **Easy to work with multiple documents:** With a single keystroke you can save all open files, save a copy of the current file using a different name but continue editing the current file, and more. Combine these capabilities with the multiple windows feature, and you've got a lot of options to suit your own taste.



## 1. Download and install Jedit

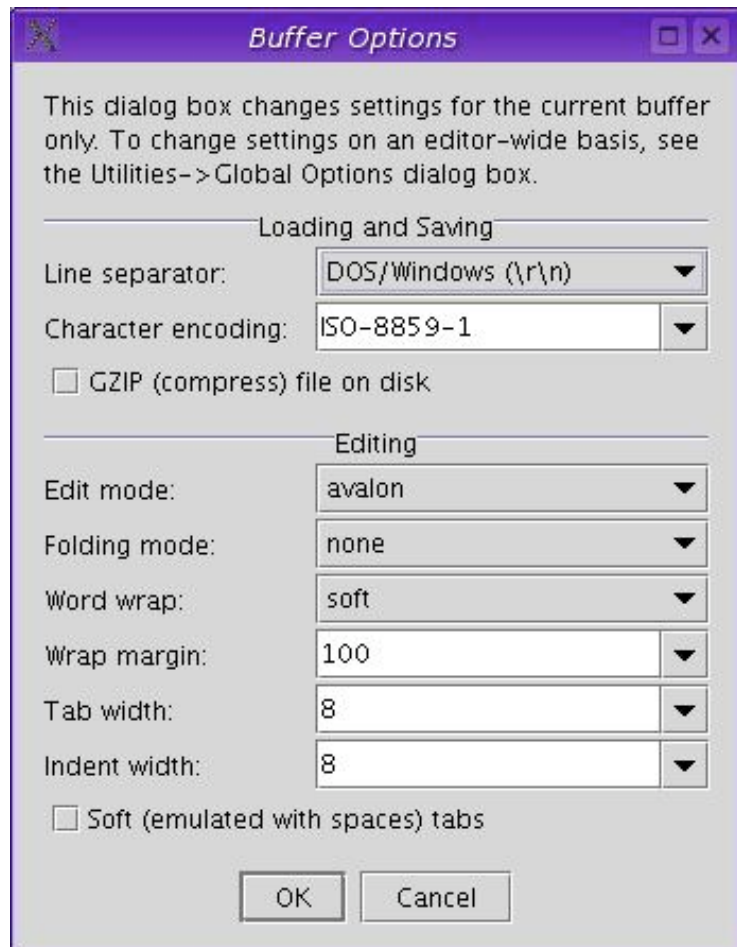
Head over to the Jedit web page – [www.jedit.org](http://www.jedit.org) – to download it. It costs nothing. Jedit is available for Mac OSX, Windows, Linux, Unix, and more. The web page has good instructions for downloading and installing the software, because the process is a bit different on each platform. You can delete this installer file from your desktop or the temporary directory after you have installed the program. Because this program requires Java, your computer should have an up to date version, or you will get an installation error (If you need to update Java you can do so by downloading the latest edition from

[www.java.com](http://www.java.com) – having an up to date copy of Java will be useful to you for other reasons as well, including for browsing some Internet sites.)

## 2. Configure Jedit

Start Jedit and open up a new document. Click on the "Buffer Options" button or select Utilities → Buffer options from the drop down menu and configure the following:

- Line separator should be set to "DOS/Windows." The issue is that Windows machines, Macs, and Unix/Linux machines all put a different character at the end of a line of text to indicate a carriage return. DOS/Windows is the lowest common denominator, so it's safe to select it.
- Character Encoding should be set to ISO-8859-1 or MacRoman if you are working with Latin languages and UTF-8 or Unicode if you work with a mix of English and Asian languages.
- Word wrap should be set to Soft wrap so Jedit behaves the way you expect it to. Otherwise it physically cuts each line at a certain length, like 72 characters (Fixed line length is good for programmers, but not appropriate for Avalon authors.)



## 3. Set Up Syntax Highlighting

### Get the Mode Files for Avalon Codes

First, in order for Jedit to understand Avalon's codes, you need a file that describes to Jedit what Avalon codes look like: it's called the mode file. Then, you need add the mode file to Jedit's catalog of codes so it knows to look for this customized file.

First get the mode file(s). I've made two mode files, one for the Moon series ([www.therandymon.com/files/moon.xml](http://www.therandymon.com/files/moon.xml)) and one for the Living Abroad series

([www.therandymon.com/files/abroad.xml](http://www.therandymon.com/files/abroad.xml)). Download the file(s) you want and save it to your desktop.

Now transfer the file to Jedit's configuration files in Jedit's "modes" subdirectory. On Windows machines your settings, macros, and mode files are located at C:\Documents and Settings\{your name}\.jedit (note the dot before "jedit"). On Linux machines they are at ~/.jedit. On Macintosh machines they are under your home directory in a hidden folder called .jedit (the best way to get there is just to open the file in Jedit, choose "save as" and then navigate your way there manually).

Second, now that you've added the modes file to the modes directory, modify Jedit's catalog file so Jedit knows the new mode file(s) exist. Using Jedit, edit navigate to .jedit/modes/ and open the file named "catalog." Then modify that file by adding one or both of the following line of code depending on which mode files you've added, that is, which code sets you intend to work with:

```
<MODE NAME="moon" FILE="moon.xml" FILE_NAME_GLOB="*.mon"  
<MODE NAME="abroad" FILE="abroad.xml" FILE_NAME_GLOB="*.abr"
```

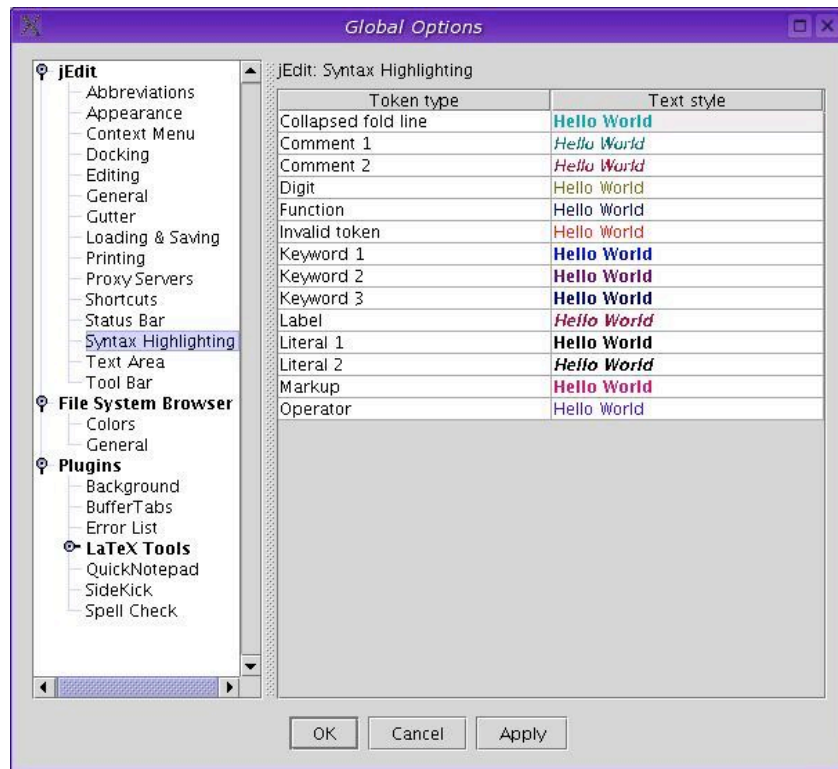
These two lines tell Jedit to use the moon.xml syntax highlighting file to edit any file whose name concludes with \*.mon and the abroad.xml syntax highlighting file to edit any file whose name concludes with \*.abr. From now on, all you have to do is give your file the right name and Jedit will figure out what to do.

To check if Jedit is aware of these two new styles, go to Edit -> Global Options and select "Editing." Under "Default editing mode," if you scroll down the list, should have "moon" and/or "abroad" as options if you installed the above file(s) correctly (incidentally, you can select one as the default mode if you'd like, but it's not necessary).

Remember, because you're working with text files, you're not changing the actual contents of your manuscript, just providing a signal so Jedit knows which syntax highlighting to apply, that is, how to graphically represent the text it finds between avalon codes.

## **Set up appropriate looks for the syntax**

Jedit looks for sequences of characters and then highlights them appropriately according to how you want it to look. How you want it to look is defined under Utilities -> Global Options -> Syntax Highlighting. I've set up the "abroad" mode file so that anything that begins with <@GB> and ends with <@\$P> is a considered type "Literal1," so if you tell Jedit to make bold all text that is recognized as Literal1, you will be able to enjoy seeing all your bold text in bold. Likewise, I've defined anything that begins with <@GI> and ends with <@\$P> as Literal2. So under the Syntax Highlighting configuration panel, ensure that Literal2 is represented in italics. The mode file also ensures that the equivalent sequences for callout text (i.e., <@CB><@P> in the Moon series) are represented as well. The mode file for the Moon series works in the same way.



Likewise, the mode files I created treat any line that begins with @ and a number (for example, @1:Food and Lodging) as type "keyword1", which you can see I've chosen to show in bold blue. Lines that start with @\$: are highlighted as "operator", which is normal blue. Any codes between brackets I've highlighted as type "Markup", for which I've chosen a reddish color. You can choose any combination of colors, bold, and italics that makes you happy, using the Syntax Highlighting configuration panel.

## Clever macros make your life easier

### Use macros for diacriticals (ñ, á, é)

I've developed my own set of macros to deal with accented characters. For example, to insert an "á" in my text (<\#151>) I can just hit "control-M a". All my macros start with control-M followed by the character I want. You can use any keys you want to create your macros.

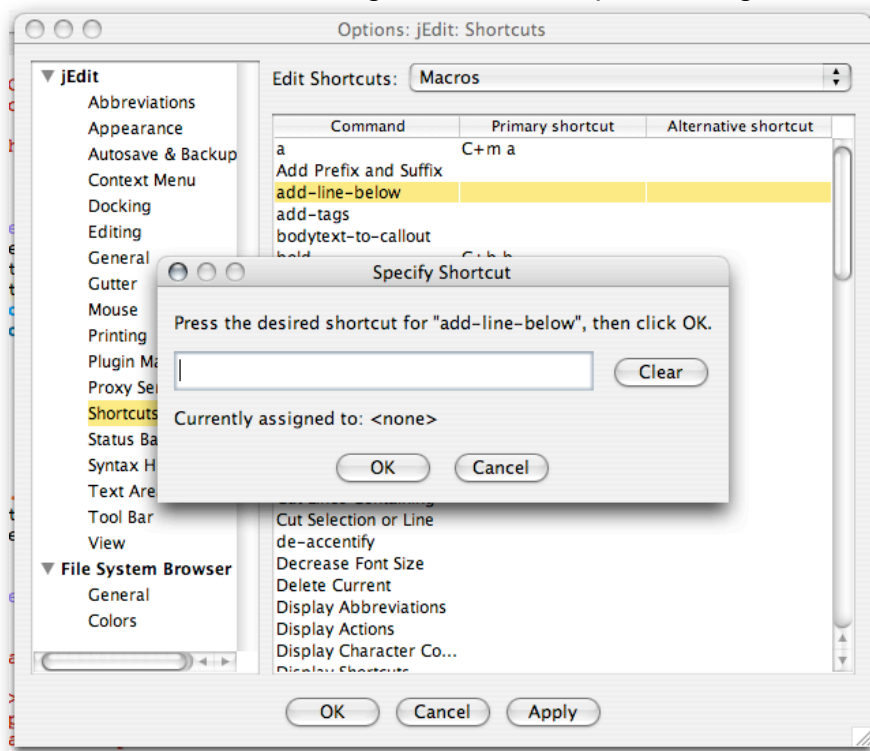
Creating keyboard shortcuts for the characters and macros you generate is a simple two step process: first you create the macro, then you assign a key sequence to it. First, create your macro. Under the macros menu, hit "Start recording macro" and then do what you want to do, for example, type <\# 1 5 1 >. When you finish, hit "Stop recording macro."

Now assign it a shortcut (a keystroke combination of your choosing) in the Global Options -> Shortcuts menu. I've got macros for most accented characters, ñ, and the like.

In each case the process is the same: start recording the macro, stop recording, save it, and assign it a shortcut.

## Use macros to code bold, italics, etc.

You can use the same trick to code your formatting. I have created a macro that generates the code `<@GB><@P>` and then puts the cursor between the two codes for me to type whatever text I want to code as bold. To create this macro start recording, type the codes and then the "left key" 4 times, then stop recording. Assign it a shortcut and you're done. This is a useful trick for dealing with things like the formatting codes for representing A.M./P.M.



## More Macro Tricks

A macro is just a list of recorded instructions. Jedit has two mechanisms that permit you to make a macro. If you know a little about programming, you can just program it. Such a program would look like this:

```
selection = textArea.getSelectedText();
String tgt=selection;
StringBuffer s=new StringBuffer(tgt.toLowerCase());
```

Since most folks aren't into that kind of stuff, the easier way is to record your macro. Press "Record Macro" from the Macros pull down menu. Jedit then proceeds to remember all the steps you take until the moment you hit "stop recording macro." No matter if you program or record, the result will be the same: a file whose extension is .bsh. All macros recorded, downloaded, or copied, get stored in the directory .jedit/macros. If you'd like to organize them into subdirectories you are free to do so.

You do not have to create that folder, but Mac's Finder and Windows Explorer are programmed not to permit you to see files whose names begin with a period, so these are not the tool to use. If you record your macro, Jedit will take you right to the correct directory for you to save it. If you download my macro package (see the next section) you should do the following instead:

1. save the macro, for example title\_case.bsh onto your desktop.



2. open up Jedit.
3. browse to that file, i.e. title\_case.bsh and open it. If you don't see it using the Jedit "open file" dialogue, make sure the little "filter" box isn't checked, in which case it's filtering out all names that don't match some criteria (like \*.txt, for example). If you open the file, you'll see the code of the macro.
4. now hit "save as" and save the file (the macro) into your jedit macro directory.
5. There is no need to restart Jedit after this operation: your macros should show up on the drop down macro menu, possibly under a subdirectory if that's where you decided to save it.

Now assign the keyboard shortcut of your preference to this macro:

1. Open up Jedit and from the Utilities --> Global Options pull down menu select the "shortcuts" tab.
2. Make sure to select the "Macros" subset of commands from the selector at the top of the dialogue so you can see the macros.
3. Click in the "shortcut" column and hit the keys you want.

## **Download my macros and be done with it**

I've developed a rich set of macros. Download [www.therandymon.com/files/macros.zip](http://www.therandymon.com/files/macros.zip), unzip it, and save the files to your macros directory. Then just assign them the shortcuts you want.

Here is a description of the macros I've created, and remember you can use Jedit to modify any of them by editing them the way you would any other file:

- add-line-below: opens up a blank line below the cursor – I use this all the time.
- make-italics: select some text, and this macro will add the codes before and after the text to make it italics.
- mooncode-to-livingcode: searches for codes used in the Moon series and replaces them with codes from the Living Abroad series (i.e. <@P> becomes <@\$P>).
- bodytext-to-callout: replaces <@GB> and replaces it with <@CB>, etc.
- bold: adds the code <@B>
- plain: adds the code <@P>
- de-accentify: searches for á, ñ, é, etc. and replaces them with ATP codes
- remove-trailing-whitespace: searches through the document and removes the additional spaces that wind up at the end of paragraphs after lots of cutting and pasting.
- italics: adds the code <@I>
- tab: adds the code <\\f>
- title\_case: Select some text and this macro will capitalize each word (i.e. for fixing @3: titles that used to be all caps)
- make-bold: select some text, and this macro will add the codes before and after the text to make it bold.

# Searching and Replacing

## Hypersearching

Jedit's search and replace facilities are fantastic. The "hypersearch" feature is the best. It shows you a list of all occurrences of what you're searching for in context. You can just as easily search all open files at once. For example, the name of a popular Nicaraguan hotel changed, which affected not only the chapter text but also several special topics. Changing them all at once was as simple as opening all those files, and doing a search and replace on all open files in one simple step.

## Incremental Searches

Here's how to use the incremental search capability: Let's say you're editing a document and you want to move the cursor to a section that includes the words "rotten mangos". You hit control-comma to launch the incremental search bar. Now start typing the words you want to search for: r o t t e n. As you type the 'r,' the cursor will move to the very next existence of the letter 'r.' As you add the 'o' it moves to the first occurrence of 'ro,' and by the time you have typed 'rott' it will have taken you to the first occurrence of that combination of letters. If you mistype, simply backspace and correct it. To skip ahead to the next occurrence, hit control-G; to skip back to the previous occurrence, hit control-H. This method has been proven the most effective way to search for words, and it streamlines your navigation through the document because your fingers never leave the keyboard.

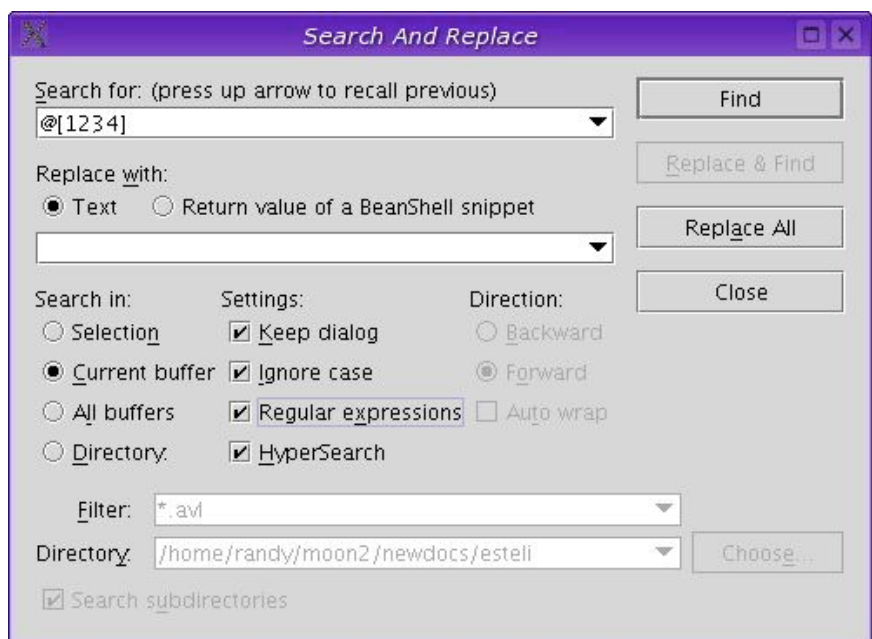
## Regular Expressions

Using regular expressions makes searching even more powerful still. Here's how you can get hypersearch results showing you all your headings: open up the find/replace dialogue and click the "regular expression " box. Now do a search for "@[1234]" (no quotes). You're instructing Jedit to search for the @ sign followed by a numeral from 1 to 4. That excludes the common @\$ tag. I use this trick to make sure all my @3 headings are in capitals, not title case.

## Using search to create a document outline

Avalon requires an outline of all headings, in which you can place callouts and such. Use the "copy lines" macro to do this in a simple three-step process.

1. First select Macros →



Clipboard → Copy lines containing. When the dialogue box pops up, input the "@" character. That will put onto the clipboard every line in your document containing the @ symbol.

2. Second, open up a new document and hit paste (control-v Windows/Linux, command-v Mac). That will paste the contents of the clipboard. All the lines containing email addresses are there too, unfortunately, since they too had @ in them! Now we'll clean it up:
3. Now step through the document with the arrow keys. Any line you don't want in the document you can delete without having to select any text with the mouse, simply by hitting control-D (command-D Mac). So, arrow down, control-D. In under 60 seconds you'll have your outline.

## Some Final Words

### Sources of Help

Jedit comes with a very well written, straight forward manual that will help you learn how to use it. Once you've installed Jedit, press <F1> to read the help manual. If you have additional questions or need help, you may also contact me.

### About Microsoft Word

Microsoft Word – the world's most popular word processor – uses a binary document format remains the world's second best-kept trade secret after “the colonel's secret recipe.” Word docs are only readable – with no formatting errors – using Microsoft's software. All writers should be very aware of document formats and the risk of software lock in. Read more about this at [www.therandymon.com](http://www.therandymon.com).

### Resources

Jedit is written in the Java programming language, which means any modern computer can run this software without modification. Read more about Java at [www.java.com](http://www.java.com).

Jedit is also open-source software. It was written by programmers for their own use, and they've agreed to make it available for anyone for free, and encourage those who are able to do so to contribute to its further development. Open source software means other programmers have the right to modify it or contribute to it to help strengthen the software, or even redistribute it as a new product. If there's a function you wish Jedit made available, you can probably simply request it in the on-line forums. Read more about open source software at [www.osf.com](http://www.osf.com). It's changing the tech world.

## **Change log for this document**

This document is a work in progress; let me know where it needs clarification or modification.

2006-09-30: Major rewrite and reorganization, improved discussion of syntax highlighting and macros, added graphics from Macintosh and Windows.

2006-01-05: Moved the mode file to a new location and updated the link.

2005-09-13: Updated mode/catalog information for Windows and uploaded a better mode file.

2005-02-02: Updated the tips/tricks, and tightened the text in other places.

2004-10-19: Updated the download instruction, fixed one changed Jedit download link, and rewrote the regexp section.

2004-07-07: reorganized the entries and numbered the steps; added the bit about incremental searches.

2004-07-03: added more specific instructions for Windows users and rewrote how to edit the catalog. Windows info courtesy of Edward Hasbrouck, author of The Practical Nomad (thanks EH)

2004-07-01: uploaded original document.