

# Colorful Editing

BY JIM MOORE

When I stop to consider the percentage of time that users spend in the ISREDIT program while in ISPF, I guess that it is somewhere between 60 and 80 percent of all time logged during an ISPF session. In other words, edit is *the* big ticket item in ISPF as far as time and usage goes. Since so much time is spent using the ISPF editor, anything that you learn about it is very valuable and can always be put to good use.

Version 4 of ISPF introduces color options for editing. This month, I'll look at how to add color to your day-to-day editing. The color components are far more than just eye-pleasing as we shall see. The use of intelligent color in an edit session can help you write more accurate code — JCL, Assembler, COBOL, REXX, etc. — because it makes common coding mistakes very obvious.

## GETTING ACCUSTOMED TO COLOR

For long-time ISPF users such as myself, color is initially shocking. I'll be the first to admit this. For many years, the basic monotone look of data being edited has become familiar and comfortable. When color is applied, a typical first impression might be something like: "*How garish! I don't like the way that looks.*" This was my first reaction. However, now that I've tried color for a while, I am learning more ways to exploit it. Slowly but surely, I am becoming more dependent upon color to really help me cut down on mistakes. It's getting to a point where the monotone look of edit is starting to seem odd to me!

## COLOR IS NOT FOR EVERYTHING

Turning on color is certainly not a one-size-fits-all proposition. Many types of data that are edited will not benefit from it at all.

Color is really only helpful for programming languages, like the ones I mentioned earlier. If you are editing raw text, data files with packed/binary values or any other non-traditional data, color won't even be turned on.

---

**The use of intelligent color in an edit session can help you write more accurate code — JCL, Assembler, COBOL, REXX, etc. — because it makes common coding mistakes very obvious.**

---

## HOW TO ACTIVATE COLOR

There are two ways to activate color in an edit profile: By using the *Edit* drop-down menu from within the edit session or by using the HILITE primary command. Initially, you should try the *Edit* method. To do so, activate the *Edit* drop-down. It appears second from the left on the edit action bar. This can only be done from within an active edit session. Choose Hilite, which is option 3. A pop-up dialog will execute. There will be a list of languages arrayed in a column down the left-hand side of the panel. Note the first choice: *Automatic*. This is what I have been using. It is very nifty. When selected, the coloring scheme is automatically determined. If you are editing BAL source code, BAL coloring is used. Automatically, JCL, COBOL, or whatever language you edit will be colored accordingly.

On the right, near the top of the pop-up panel is another set of choices. Select number 2: *Color the program*. If you don't select this, coloring won't activate.

Finally, there are several other selections on the pop-up like parentheses matching and cursor string highlighting. Try turning them all on by selecting them with a slash. When you exit, a long message will appear in a pop-up that informs you that you have selected color for your edit session. Upon returning to your source code, you should immediately see the color activated.

## COLOR PROBLEMS

One of my consulting partners is color-blind. He has a tough time seeing red and green properly. He discovered that the default colors can be changed. This is done by choosing one of the drop-down menus from the Hilite utility's action bar.

Not all terminals or emulators support color. This is especially true of older hardware and software. If your terminal doesn't support color, you will not be able to select the Hilite choice nor use the HILITE primary command. Time to upgrade.

## TIP OF THE MONTH

When using the APPEND primary command at the DSLIST, you do not need a closing quote on the mask that you are APPENDING to the list. This is consistent with *Other Dataset Name* fields on ISPF panels.

---

Another thing to be aware of is that the **HILITE** setting is stored in the edit profile. This is like many other edit profile settings: NULLS, RECOVERY, STATS and many others. What this means is that color must be activated in datasets with different low-level qualifiers: Turning color on in a dataset name that ends in CNTL will not turn it on in a dataset whose name ends in COBOL.

---

### THE AUTOMATIC COLOR SETTING

The reason I prefer this setting is that the editor detects the language source that you are editing and automatically colors it properly. Try it. Edit a REXX EXEC with automatic coloring on. Then try a JCL member. Make sure these members are both in the same PDS with color activated.

The tutorial on **HILITE** explains what the editor looks for when attempting to determine the language type. It also mentions that with color activated, a lot of in-memory scanning is done every time you press enter. Very large files being edited with color turned on could cause a slight performance problem.

Here is something interesting to try. When you have automatic coloring turned on and the data is being properly colored, issue the PROFILE primary command. Note how the language that you are editing appears after the word **HILITE**. For example, if you are editing REXX, the profile display will display **HILITE REXX**. For COBOL it will show **HILITE COBOL**. This is pretty darn cool.

---

**Although it may take some getting accustomed to, I highly recommend that all ISPF edit users at least experiment with the use of color in their edit sessions.**

---

---

### HOW COLOR AIDS CODING ACCURACY

When coloring is active for an edit session, common coding problems become very obvious. Try this: Deliberately omit the closing quote on any quoted string and press enter. Wow! Everything lights up. OK, close that quoted string.

Simply having the reserved words of a language appearing in a unique color is helpful. When you misspell a reserved word, it doesn't change to the proper color.

Comment lines in any language and in-stream data in JCL are very apparent with coloring activated. There is even a default color for a common group of special characters like ampersands and equal signs.

---

### THE HILITE PRIMARY COMMAND

Recently, I have been experimenting with this primary command. There are some special parameters that you can pass to it to help locate mismatches in common logic constructs like DO/END and IF/ELSE. It also can be used to simply turn coloring on and off. I am really interested in learning how to use the logic matching parameters. There

have been many times in my programming career that I really, really could have used something like this. As I discover the best uses for them, I'll certainly pass them along.

---

### CONCLUSION

Although it may take some getting accustomed to, I highly recommend that all ISPF edit users at least experiment with the use of color in their edit sessions. It will help you write more accurate code. Really. Also, spend a few minutes in the tutorial. Just read a little bit each day and then experiment with what you have read about. The use of color in the ISPF editor is an easy, painless and "colorful" way to work smarter! 



---

NaSPA member Jim Moore is the president of Concentrated Logic Corporation, a Glendale Heights, Ill.-based software development firm specializing in TSO/ISPF/PDF and database design. He can be reached at [conlogco@ix.netcom.com](mailto:conlogco@ix.netcom.com).

©1998 Technical Enterprises, Inc. For reprints of this document contact [sales@naspa.net](mailto:sales@naspa.net).