

# Staying Put While on the Move

BY JIM MOORE

This month, I'll discuss some new navigation techniques provided by ISPF Version 4. Also, the ISPF Workplace will be examined. This new tool has been assigned Option 11 from the venerable `ISR@PRIM` primary option menu.

## STAYING PUT WHILE ON THE MOVE

The more I work with ISPF Version 4, the more I am beginning to realize that there are many new features that make the older style *jump* navigation less important. If you have been following this column for the past several months, you may already be aware of the `START` command and its associated `SWAP LIST`, `SWAP NEXT` and `SCRNAME` methods of navigation.

`START` allows up to eight subtasks (split screens) under an ISPF session. What I have found myself doing is issuing several `START` commands for the ISPF dialogs that I most commonly use: one for Edit, one for `DSLIST`, one for `SDSF`, and so on. Then, instead of constantly *jumping* with `=2` or `=3.4`, I toggle through my screens with `SWAP NEXT` or `SWAP LIST`. Working this way, I never have to shut down what I am working on. On many days, I keep all of my `START` sessions running all day.

Another new method of navigating is the action bar method. Have you tried getting the Utilities drop-down menu active from an action bar? Try it. Next, select one of the utilities. Then from the selected utility, again get the Utilities drop-down active and select yet another utility. Now, start pressing `PF3 (END)` to back out of the stack of dialogs that you have selected via the action bar drop-downs.

Moving about ISPF in this manner is different than always using the *jump* function. It is more like a stack or nest. That is, dialogs are temporarily suspended by

**This is the essence of the object/action interface: Ask for something to be done against an object and you'll either be denied or obliged based upon the combination of what you are asking for.**

newly invoked dialogs. It's as if the newly invoked dialogs are *stacked on top of* the suspended ones. When this type of navigation is used in conjunction with multiple `START` sessions, I feel as though I am *staying put while on the move*. I never lose what is "behind" my current screen until I deliberately elect to lose it. This is very Windows-like.

I must say that this capability has been in ISPF for many years. In the past, I routinely added a `STACK` primary command to pre-Version 4 `ISPCMDS` command tables. The action for the `STACK` primary command would look something like this:

```
SELECT PANEL(ISR@PRIM) OPT(&ZPARAM)
NEWAPPL(ISP)
```

Once this command was in an active `ISPCMDS`, I could enter `STACK 3.4` or

`STACK 2` from any ISPF command line to invoke dialogs *on top of* ones I was already using. This was good enough for me until Version 4 came along. Now, IBM has incorporated this type of navigation right into ISPF. Bravo!

## THE ISPF WORKPLACE

I have been making a conscious effort to use this new dialog. I'll be perfectly honest, I have yet to become totally comfortable with it. I have been using it enough though to have some interesting observations. What is most striking about the Workplace is that it really allows you to *stay put but move*. That is, you can launch almost any ISPF utility from the Workplace without leaving it!

This is really what the Workplace offers: a single screen from which to invoke any of the many ISPF utilities. However, many years of habitual behavior are not easy to change. I still move a bit slowly when using the Workplace dialog. I am learning though, and for the remainder of this column I will pass on my observations.

## OBJECT/ACTION

To begin to familiarize yourself with the Workplace, navigate to it by issuing the jump command: `=11` from any ISPF screen. Once there, simply press your `HELP PF` Key. You will be presented with a pop-up help screen that briefly explains what IBM

## Tip of the Month

To enter a dataset list mask at the Object Workplace, you must at least begin the mask with a quote. The closing quote is optional just as it is on the `APPEND` primary command of Option 3.4.

calls the object/action interface to ISPF. At this initial HELP pop-up, notice that this is one of the newer type help screens that REQUIRES that a PF Key is equated to the word HELP. I have mentioned this several times over the past few months. Here is an example of this requirement.

Read the topics called *Specifying an object* and *Specifying an action*. There is quite a bit of information to digest in these two topics. Basically, the object/action interface will invoke whatever ISPF dialog underlies the action against the object that you type in.

What tends to confuse me (and I assume others!) is just what to enter for the Smart Action in the Action ➡ field. You select actions either from the File or SuperC choices on the Workplace action bar or by using FastPath commands. I have been trying to stick to the FastPath commands. These are all listed in the tutorial under the *Specifying an action* topic. Still, even having a list of all possible actions doesn't totally eliminate all confusion.

#### WORKPLACE EXAMPLES

I know why I am confused. It has to do with the fact that I have to have some kind of idea of what my object is and what action I am trying to perform against it. Let me present a few examples to clarify.

1. Try entering one of your own TSO datasets as the object. When you enter the object, use standard TSO/ISPF entry. That is, omit your TSO prefix and do not enclose the name in quotes. For the action, enter DL and press enter. You should be presented with

a 3.4 DSLIST that contains only one dataset name: the one that you entered as the object. This makes sense. Now try the following.

2. With the same unqualified dataset name as the object, enter a Y (enhanced allocation) as the action. Did you get a *Duplicate dataset name error*? You should have.

Finally, try this last exercise.

3. Change a character in the object dataset name so the Workplace will recognize that the dataset is not cataloged. For example, if the dataset name ends in .CNTL, change the low-level qualifier to something like .CNTLZ. Now enter a Y as the action. You will be presented with the screen from the 3.2 utility with all the SMS enhancements.

The Workplace is pretty smart! Probably smarter than us poor programmers. To recap the sequence of events, consider:


- ◆ When you asked for a DSLIST against a single data set, you were given one.
- ◆ When you asked for an enhanced allocation of an existing dataset, you were denied (correctly so).
- ◆ When you asked for an enhanced allocation for a non-existent dataset, you were obliged and presented with the proper screen.

This is the essence of the object/action interface: Ask for something to be done against an object and you'll either be denied

or obliged based upon the combination of what you are asking for.

#### CONCLUSION

I have shown you three ways to stay put (not shut down) while on the move (opening new work) without resorting to the destructive jump commands. Now, try to practice using these new methods of navigating. It's not easy. I still find myself doing jumps. But I'm trying to move about in the new ways because I know that the constant starting up and shutting down of dialogs causes quite a bit of overhead. Plus, if I have a dialog, such as edit open and active, I will almost certainly need it again during the course of a workday.

Next month, I will examine the Object Workplace in greater detail. 



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).