

Using ISPF Edit Labels for Fun and Enhanced Productivity

By Lionel B. Dyck

Contents

- Overview 2
- SAVELAB 3
 - Features of SAVELAB 3
 - Save and Restore 3
 - List 3
 - Import and Export 4
 - Hide 4
 - Command Prompting 4
 - SHOW ALL or SHOWALL 5
- Getting Help 5
- REXXLAB 6
- COBLAB 7
- Beyond COBLAB and REXXLAB 8
- Contributing updates 8

Overview

Quoting from the IBM z/OS ISPF Edit and Edit Macros (SC19-3621-50):

A label is an alphabetic character string used to name lines or strings of data for easy reference. Because labels remain with the lines to which they are assigned, they are especially useful in keeping track of lines whose numbers might change. Most labels are assigned in macros, but certain labels are automatically assigned by the ISPF editor.

Labels provide quick access to parts of the program without having to know a record number or perform a find for some string.

Labels can be assigned to meaningful sections of code. For example, in a COBOL program the Identification, Environment, Data, and Program divisions could each have a label.

This is a fantastic feature of ISPF Edit provided by the ISPF developers and yet it is rarely used because of one simple reason – the labels disappear when the ISPF Edit session ends.

Here is an example of a label in a REXX program:

```
000344 /* ----- *
000345 | Restore Labels - default |
000346 * ----- */
.REST Do_Restore:
000348 arg rest_opt
000349 address ispexec 'vget (saverest)'
```

As you can see – the label is **.REST** on record **000347**, or what would be record **000347** if the label hadn't replaced it.

To access this record while in ISPF Edit enter on the ISPF Edit command line **L .REST** using the period as it is an integral part of the label.

An ISPF Edit Label must start with a period and then may be followed by up to 5 letters (from A to Z) that will always be converted to upper case.

This article will discuss three very useful ISPF Edit commands, which are also ISPF Edit Macros, that can be used to Save, List, Import, Export, and Restore the labels. A second command is specific to REXX code and will dynamically create, and assign, ISPF Edit Labels to each REXX label. Finally, the third command is specific to Cobol and will dynamically create, and assign, ISPF Edit Labels within the Cobol code.

Note that both REXXLAB and COBLAB will enable ISPF Edit HILITE.

SAVELAB

SAVELAB is a package found at <https://www.cbttape.org> in file 313 and at <https://github.com/lbdyck/savelab> that improves upon the ISPF Edit Label features that IBM provided.

The package consists of several REXX execs, a few ISPF panels, and a sample SAVELAB Export file.

You are probably asking what benefit does SAVELAB provide? Keep reading and see the benefits, or at least some of the benefits, as I'm sure as you use this tool that you'll discover how it improves your productivity.

Features of SAVELAB

Save and Restore

With SAVELAB all defined ISPF Edit labels can be saved so that you can easily restore them the next time you edit the data.

If labels have been saved then you can easily restore them.

If you set your Initial ISPF Edit Macro to **SAVELAB RESTORE** then the labels will be restored when you start editing, if labels exist that is.

SAVELAB SAVE will save all currently defined labels.

Note: once **SAVELAB** is used once in the ISPF Edit session you can abbreviate **SAVELAB** to **SL**. And as a bonus, the ISPF Edit **SAVE** command will be intercepted automatically, so a **SAVELAB SAVE** will be performed unless the ISPF Edit session is cancelled.

Think of the saved labels as part of the metadata for the source code that can be shared using the **EXPORT** and **IMPORT** options (below).

You can also enable SAVELAB as your Initial ISPF Edit Macro using the ISPF Edit command **IMACRO SAVELAB**, and then turn it off if you ever need to by using the ISPF Edit command **IMACRO NULL**.

List

Once the labels have been created, using **SAVELAB LIST**, or **SAVELAB SHOW**, will display a selection table of all defined labels:

```
Active Edit Labels
Command ==> _ Row 1 of 20
Command: Sort (Label) SortR (Record) Select: S or Enter
S Label Record Data
- .VER      2 ver = "2.16"
- .NOTES    94 * Notes: 1. If the ISPF Session abnormally term
- .HIST     114 * History:
- .DEFS     237 parse value '' with null free_labset quiet imac
- .DEFED    246 Address ISREdit
- .CKENT    253 Address ISPEXec 'vget (zeditcmd)'
- .CKVS     266 Check_View_SCLM:
```

As you can see, each label can be selected which will then position the Edit session on that record. The record number and the 1st part of the record data are also included to further help identify the label record.

Import and Export

The **EXPORT** option will save the label information so that you can share the labels with someone else who can then **IMPORT** them and begin to use the same labels.

Both **IMPORT** and **EXPORT** require the specification of a dataset name (PDS/PDSE). The member name is implied by the member being edited.

Hide

Sometimes it is helpful to just see the record with the label and a few records before, and after, the label. That is where **HIDE** comes into play.

```
----- 342 Line(s) not Displayed
000347 | Restore Labels - default |
000348 * ----- */
.REST Do_Restore:
000350 arg rest_opt
000351 address ispxec 'vget (saverest)'
----- 66 Line(s) not Displayed
000418 | Close and Exit |
000419 * ----- */
.CLOSE Close_Out:
000421 Address ISPEXec
000422 'tbclose savelab replcopy library(ispprof)'
----- 4 Line(s) not Displayed
```

Hide takes an optional parameter specifying the number of records to display (i.e. **SAVELAB HIDE 5**) with a default of 3.

Command Prompting

If you forget the options available just enter **SAVELAB ?** to see a popup panel where a command may be entered (or use point-and-shoot):

```

SAVELAB Command Prompt
Command ==>

Export  Export Labels (dsname required)
Import  Import Labels (dsname required)
Help    Display Tutorial Panel
Hide _  Hide all and Display n lines before/after each Label
List    List all active labels
Restore Restore saved labels
Save    Save all current labels
ShowAll Show all dataset(members) with saved labels

Enter a command or select (point-and-shoot) or F3 to cancel

```

SHOW ALL or SHOWALL

If you ever want to see what labels you have created you can use the **SAVELAB SHOW ALL**, or **SAVELAB SHOWALL**, command. From this display you can select to view the saved labels, use the saved labels from one member with the active member, or delete a saved set of labels.

The saved labels are stored in an ISPF Table for safe keeping.

```

----- Saved Label Dataset (Members) ----- Row 1 of 3
Command ==> _                               Scroll: CSR

Line: S Select D Delete U Use

S   Dataset (member)                        Date
_   LBDYCK.LIONEL.EXEC(COBLAB)              08 Jul 2023
_   LBDYCK.LIONEL.EXEC(REXXLAB)             08 Jul 2023
_   LBDYCK.LIONEL.EXEC(SAVELAB)            08 Jul 2023
***** Bottom of data *****

```

Getting Help

To learn more about SAVELAB enter **SAVELAB HELP** to view the Savelab ISPF Tutorial.

REXXLAB

REXXLAB is an ISPF Edit macro that will dynamically create ISPF Edit labels. A comparable macro could be developed for other languages.

Routine and Procedures have labels that start with text and end with a “:”

The label will be the 1st letter of each part of the routine or the 1st 5 characters of the routine name.

If the label is 5 characters, or less, then it will be used as is.

If the label with _ removed and combined is 5 characters, or less, then it will be used.

Examples:

DoIt:	.DOIT
Do_IT:	.DOIT
Do_It12:	.DOIT
Do_Another:	.DOANO
DoSomething:	.DOSOM

If a routine label contains characters that cannot be used in an editor label, (e.g., numbers or special characters) those characters will be translated to blanks.

Existing labels will remain untouched, duplicate labels will have a suffix of A, B, C, ... appended.

When used a table is displayed of all generated labels:

```
Active Edit Labels
Command ==> _ Row 1 of 6
Command: Sort (Label) SortR (Record) Select: S or Enter
S Label Record Data
_ .START 68 Start:
_ .DOIT 73 Do_It:
_ .DNEW 125 DoneWithTheTask:
_ .FL 136 Fixup_Label:
_ .GL 163 Get_Labels:
_ .CL 182 Check_Loadispf:
***** Bottom of data *****
```

And a long ISPF message with some useful information:

```
6 labels created with 0 existing.
Use command Savelab Li (or sl li) to list active labels.
Use command Savelab Save (or sl save) to save active labels which can then
be restored using the command Savelab.
```

COBLAB

COBLAB is like REXXLAB but designed to work with COBOL source code and will dynamically create ISPF Edit labels for key COBOL constructs.

Here is a sample result of issuing the **COBLAB** command while in ISPF Edit:

```
Command ==> _ Scroll ==> C
000049
.ID      IDENTIFICATION DIVISION.
.PI      PROGRAM-ID.
000052      'XMITCICS'.
000053      ENVIRONMENT DIVISION.
.DD      DATA DIVISION.
~~~~~
.WSS     WORKING-STORAGE SECTION.
~~~~~
000061      01 USER-ID                PIC X(8).
0
0 24 labels created with 0 existing.
0 Use command Savelab Li (or sl li) to list active labels.
0 Use command Savelab Save (or sl save) to save active labels which can then
0 be restored using the command Savelab.
0
```

And the list of generated labels can be found using the **SL L** command (short for **SAVELAB LIST**):

```
Active Edit Labels
Command ==> _ Row 1 of 24
Command: Sort (Label) SortR (Record) Select: S or Enter
S Label Record Data
- .ID      50 IDENTIFICATION DIVISION.
- .PI      51 PROGRAM-ID.
- .DD      54 DATA DIVISION.
- .WSS     58 WORKING-STORAGE SECTION.
- .CX      105 COPY XMITMADF.
- .LS      108 LINKAGE SECTION.
- .PD      114 PROCEDURE DIVISION.
- .OO      167 OPEN-OUTFILE.
- .WLS     199 WRITE-LINE-SECURE.
- .WL      205 WRITE-LINE.
- .CO      229 CLOSE-OUTFILE.
- .AO      243 ALLOC-OUTFILE.
```

Beyond COBLAB and REXXLAB

Anyone interested could take REXXLAB, or COBLAB, and with some, hopefully, minor alterations create a ASMLAB or CLAB or ...

Contributing updates

If you wish to contribute updates, such as an ASMLAB or CLAB or xxxLAB, just go to GitHub and fork the repository. Then make your updates and generate a pull request. Standard GitHub all the way.