



# The ISPF Edit Clipboard: Demystified



By **Lionel B. Dyck and William J. Smith**

**“F**ailure is success, if we learn from it.”  
~ *Malcolm Forbes*

As used in computer technology, a clipboard is, “A temporary memory area, used to transfer information within a document being edited or between documents or between programs. The fundamental operations are **cut**, which moves data from a document to the clipboard, **copy**, which copies it to the clipboard, and **paste**, which inserts the clipboard contents into the current document in place of the current selection.” (The Free On-line Dictionary of Computing, <http://www.dictionary.com/browse/clipboard>)

In general, virtually all operating systems and word processors implement the virtual clipboard functionality, but some impose restrictions upon the format or types of data that one may cut for placement in the clipboard. For example, the cut and paste operations may only support text and prohibit use of all graphical data.

IBM’s *ISPF Program Development*

*Facility* (PDF) editor implements the CUT and PASTE operations as primary line commands. IBM defines the CUT command as an editor operation that “*saves lines to one of 11 named clipboards for later retrieval by the PASTE command.*” In addition, the PASTE command “*moves or copies lines from a clipboard into an edit session.*”

Surprisingly, and despite being used liberally within the “ISPF Edit and Edit Macros” manual (SC19-3621-01), a formal definition of clipboard is nowhere to be found. However, the “ISPF Planning and Customizing” manual (GC19-3623-01) states:

“Edit clipboards are used by the Edit CUT and PASTE commands and are kept in data spaces that last the life of the TSO session. Data spaces are allocated by the CUT command and released by the PASTE command, and are paged out when not in use. This can be a number between one and 11. The default is 11.”

Moreover, IBM states that the ISPF product installer may configure the maximum size for the edit clipboards in 4K increments in the ISPF configuration table during or after installation. The default value of zero indicates that the edit data space size (239 4K blocks) of the value set through the IEFUSI SMF exit should be used.

Before IBM formally incorporated support for the clipboard within the ISPF PDF editor, customers implemented numerous variations of CUT and PASTE and contributed them in the public domain to the CBT web site ([www.cbttape.org](http://www.cbttape.org)). In addition, some of the more sophisticated customer implementations provided extensions that allowed the clipboard to persist across ISPF and TSO sessions. This article discusses only the native ISPF PDF CUT and PASTE primary line commands as implemented by IBM.

### Why use the Clipboard?

Among the numerous reasons for using the PDF clipboard are the following:

1. Holding records from one edit session of a data set or PDS Member for pasting in a subsequent invocation of the PDF editor.
2. Productivity: As a cached repository for records used repeatedly, and which necessitates convenient, quick-and-easy access.

### CUT

The CUT command saves copies of specified records—i.e., rows or lines of data—from an active edit session into a clipboard. Only the ISPF PDF editor and its companion VIEW application know of (and have access to) the clipboard's in-storage data spaces. Clipboard data are persistent across multiple invocations of ISPF within the *same* TSO session. A TSO LOGOFF will purge all data in all clipboards. In other words, the clipboard content in all data spaces is purged by the initiation of a TSO LOGOFF command. Clipboard data are only accessible using the ISPF edit PASTE primary line command.

## Syntax of the CUT Primary Line Command (see Figure 1)

### Range

The range of lines acted upon by a cut operation may be specified using labels, row (line) numbers, or reserved ISPF system 'z' variables such as ZFIRST, ZLAST and ZCSR.

If the user omits the range designation, then the range is assumed to be copied (C/CC) or moved (M/MM) from tagged rows. However, if the user has not tagged any rows, then ALL rows are copied to the clipboard.

This includes both user-defined labels (e.g., .ROWA) and ISPF defined system labels (e.g., .ZFIRST, .ZLAST, and .ZCSR).

### Clipboard

**DEFAULT** is the ISPF default named clipboard. A user does not need to specify it as a command operand.

**Clipboard-name** is any user-chosen name that conforms to the naming convention for partitioned data set (PDS) members. Note that a maximum of 10, *user-defined* clipboards are allowed.

### Options

**APPEND** adds specified rows (records/lines) to any existing content (data) in the clipboard. If the clipboard is empty, or does not exist, then the PDF editor creates the in-storage clipboard (i.e., the data space) on behalf of the user.

**REPLACE** is the default option. It will overwrite and thus *replace* all existing content in the clipboard with the specified data.

**X** copies only those rows in the specified range that are explicitly eXcluded from display within the edit window.

**NX** will only copy those rows in the specified range that are *not* eXcluded from display within the edit window.

**DISPLAY** is a standalone operand that cannot be used with any other operands: It will display all currently defined clipboards.

### PASTE

The ISPF edit PASTE primary line command copies data from the specified

(remove) data from the clipboard. Hence, it is available for use in subsequent (repeated) paste operations.

**PASTE**

The ISPF edit PASTE primary line command copies data from the specified clipboard into the specified location within the currently edited (or viewed) data set or partitioned data set (PDS) member.

**Syntax of the PASTE Primary Line Command (see Figure 2)**

**Clipboard**

**DEFAULT** is the *default* clipboard name. **DEFAULT** does not need to be specified with the command.

**Clipboard-name** is any name that conforms to the naming convention for partitioned data set members. Note that a maximum of 10, *user-defined* clipboards are allowed.

**Location:** The label is any user-defined (or system) label. The location may be designated using the A (After) or B (Before) row tags.

**Options**

**DELETE** removes (purges) all data from the clipboard.

**KEEP**, the default action, does not delete

(remove) data from the clipboard. Hence, it is available for use in subsequent (repeated) paste operations.

**The Basics**

The **CUT** command syntax provides insight on its capabilities. The basic syntax is:

**CUT**

By default, all records (i.e., rows/lines) open in the current edit session are copied to the default clipboard.

Using line numbers, provide additional granularity in the selection of what records (lines) are copied to the clipboard. For example, **CUT 5 10** will copy records five through 10 to the default clipboard.

In addition, labels can be substituted for line numbers; they provide line number independence as noted below:

**CUT .A .B**

Or **CUT .A 99**

By default, all content (data) in the

CUT	Range	Clipboard	Options
	.ZFIRST .ZLAST (d)	DEFAULT (d)	APPEND
	#1 #2	Clipboard-name	REPLACE (d)
			X
			NX
(d) = default			DISPLAY

**Figure 1: CUT Syntax**

PASTE	Clipboard	Location	Options
	DEFAULT (d)	AFTER label	DELETE
(d) = default	Clipboard-name	BEFORE label	KEEP (d)

**Figure 2: Paste Syntax**

clipboard is replaced. By using the APPEND option, the user is free to extend existing data in the clipboard by adding more records.

user-defined clipboards JC, MH, and RH. Each clipboard has the number of lines (records) that constitute the content of each

clipboard is replaced. By using the APPEND option, the user is free to extend existing data in the clipboard by adding more records. For example,

### **CUT 1 5**

### **CUT 15 25 APPEND**

The first operation copies records one to five into the default clipboard and replaces any existing data. The second cut operation selects records 15 through 25 and appends—i.e., adds—these lines (records) to the data in the default clipboard after the existing data from lines one through five.

IBM provides with ISPF an edit macro named ISRONLY. ISRONLY is distributed in the SISPSAMP target library. It is designed to be copied into a production SYSEXEC or SYSPROC library with suggested name of ONLY. When executed, ONLY displays in the edit view window only the rows that satisfy a given selection criteria using the FIND and EXCLUDE primary line commands. To copy only those non-excluded records after using the ONLY edit macro, use the NX option. For example,

### **CUT NX**

Using a clipboard:

### **CUT .ZFIRST .ZLAST NX TEST**

Or **CUT NX TEST**

The above examples will copy all non-excluded records from the first to the last rows in the currently open data set to a clipboard named TEST. Note that these two sample commands produce *identical* results.

To display the status of all clipboards use the command **CUT DISPLAY** (or **CUT DIS**). See Figure 3 for an example.

This example of CUT DISPLAY includes the DEFAULT clipboard along with three

user-defined clipboards JC, MH, and RH. Each clipboard has the number of lines (records) that constitute the content of each named clipboard. Note the clipboard manager names six commands that operate upon the clipboards that are listed below them.

The 'User Comment' field can only be added or updated in the Clipboard Manager panel and in the Rename panel.

To PASTE data from a clipboard is just as easy. For example,

### **PASTE**

By itself, the above PASTE command will copy the contents of the DEFAULT clipboard into the currently open data set at a location specified by either the A (After) or B (Before) row tags.

### **PASTE KEEP**

The **PASTE KEEP** command operates the same as PASTE. However, it will not delete (purge) the contents of the clipboard.

### **PASTE AFTER .B TEST KEEP**

This command will copy the data from the TEST clipboard after the row with the .B row tag; upon completion, it will not delete any data from the clipboard.

Without question, the clipboard is a profoundly useful feature of the PDF editor. Moreover, it operates with equal versatility in the companion PDF VIEW application. Whether introduction of the PDF clipboard preceded or followed similarly named functionality accompanying graphical user interfaces on workstations is the subject of speculation by computer historians. Clearly, given the number of different environments (e.g., Windows, Linux, AIX, office suites, etc.) that support a clipboard, the function is deemed essential. Without question, IBM's implementation of the clipboard within the PDF editor is well executed.

### **Changing the CUT and PASTE default actions: EDSET**

IBM provides an ISPF edit command,

SHARE user community. The following discussion addresses these existing functional shortfalls within the existing ISPF clipboard

### Changing the CUT and PASTE default actions: EDSET

IBM provides an ISPF edit command, **EDSET**, which allows the user to change the CUT and PASTE default actions. **EDSET** must be issued within an edit, or view, session.

Figure 4 is an edited snippet of the edit and view settings panel.

### SETCLIP: An Open Source ISPF Dialog

#### Extending the Clipboard

The user community has requested the following two useful clipboard functional enhancements:

1. Retain the clipboard across ISPF sessions and across TSO logons, and
2. Easily define clipboards in bulk.

The authors are open source advocates, authors, and active participants in the

SHARE user community. The following discussion addresses these existing functional shortfalls within the existing ISPF clipboard implementation with an open source ISPF application, which offers a workable solution to both requirements.

**SETCLIP** is an open source ISPF application written in REXX. It is available for download from the CBT web site at [www.cbttape.org](http://www.cbttape.org) (file # 313) and at [www.lbdsoftware.com](http://www.lbdsoftware.com).

To invoke this application from any ISPF Panel issue the command: **TSO %SETCLIP**

Note that the % denotes the extended implicit invocation of a REXX EXEC by the TSO terminal monitor program. As such, it denotes that SETCLIP is not a TSO command processor. TSO executes SETCLIP directly from the SYSPROC/SYSEXEC DDNAME concatenation in the user's LOGON procedure (thus bypassing a lengthy LPALST/LNKLST/STEPLIB search by z/OS contents supervision BLDL/FIND SVC 18).

Initially, SETCLIP presents the following panel to the user (see Figure 5).

Clipboard manager			
B - Browse	C - Clear	O - Toggle Read-only	
E - Edit	R - Rename	D - Delete	
Name	Lines	User	Comment
- DEFAULT	0		ISPF Default Clipboard
- JC	3		
- MH	17		
- RH	21		

Figure 3: ISPF Clipboard Manager

CUT default . . 2	1. Append	PASTE default . . 2	1. Delete
	2. Replace		2. Keep

Figure 4: EDSET Example

```
----- ( SETCLIP - ISPF Edit Clipboards ) -----
Command ==>
```

```

----- ( SETCLIP - ISPF Edit Clipboards ) -----
Command ==>

      ClipBoard  Dataset(member)
Opt Name      Containing the Clipboard Text      Rows
1st  Last

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

Opt:  B Browse E Edit D Delete
Rows: Default is all  Enter to Save and Create  Cancel to exit

Notes: 1. The table will be sorted after each update automatically.
       2. Create these clipboards at ISPF Start by using the ZSTART
          profile variable (Issue command SET to do this)

```

**Figure 5: SETCLIP ISPF Panel**

This panel allows the user to define up to 10 user clipboards. As noted earlier, this is the limit established by IBM, in addition to the DEFAULT clipboard (for 11 named clipboards).

To define a user clipboard enter a maximum of *eight, alphanumeric and/or national characters* (adhering to PDS member naming conventions) for the ClipBoard Name, followed by the Dataset, or PDS Member, where the records reside. Optionally, the first and last rows (records) can be defined if only a few of the records are to be copied into the clipboard.

Use B (Browse) or E (Edit) to browse or edit the data set, or D (Delete) to remove it from the list.

Figure 6 demonstrates three user clipboards:

1. A JCL JOB statement
2. An ISPF Edit Macro REXX header model
3. An ISPF REXX Exec header model.

After supplying the above fields in the panel and either pressing Enter or PF3 (End), the

clipboards will be created. The new clipboards will only be available until LOGOFF. The clipboards will persist if a user stops and restarts ISPF. However, a TSO LOGOFF command will discard them, as the data spaces are lost unless as noted the user executes TSO % SETCLIP after LOGON and ISPF start.

### Enabling Clipboards across ISPF Sessions and TSO Logons

As documented on the panel, the ISPF ZSTART profile variable (introduced with z/OS V2R1) can be used to cause SETCLIP to be run each time ISPF starts. Consequently, a maximum of 10, user-defined clipboards are available for immediate use.

There are two ways to update the ZSTART ISPF profile variable (see Figure 7).

The first choice is:

1. Open ISPF Dialog Test (ISPF Option 7)
2. Select option 3
3. Insert a row as follows:

```

----- ( SETCLIP - ISPF Edit Clipboards ) -----
Command ==>

```

```

----- ( SETCLIP - ISPF Edit Clipboards ) -----
Command ===>

  ClipBoard  Dataset(member)                                Rows
Opt Name    Containing the Clipboard Text                1st  Last
-  JC       'USER01.JCL.CNTL($JOB)'                         -  -
-  MH       MY.EXEC(MACHDR)                                  -  -
-  RH       MY.EXEC(HEADER)                                  -  -
-  -       -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  -       -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  -       -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  -       -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  -       -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  -       -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  -       -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  -       -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -

Opt:  B Browse E Edit D Delete
Rows: Default is all  Enter to Save and Create  Cancel to exit

Notes: 1. The table will be sorted after each update automatically.
       2. Create these clipboards at ISPF Start by using the ZSTART
          profile variable (Issue command SET to do this)

```

Figure 6: SETCLIP ISPF Panel Example

```

Variable P A Value
-----+-----1-----+-----2-----+---
_____ ZSTART  _P  ISPF;tso %setclip x

```

Figure 7: ZSTART

The TSO %SETCLIP X will invoke the SETCLIP application and the non-blank PARM is used to inform the EXEC to update all user clipboards bypassing display of the prompting panel.

The second choice is to use the SET command within the SETCLIP application. The SET command will create, or update, the ZSTART variable to simplify this process.

## Conclusion

The authors have described how to use the IBM ISPF Edit CUT and PASTE commands, the use of clipboards, and an open source extension to simplify using user clipboards.

We believe that this article presents valuable information that clarifies and extends use of the ISPF clipboard functionality. **ETJ**

**Lionel B. Dyck** is a z/OS consulting systems programmer who has developed many open source tools for the z/OS community. He has shared these tools on his website ([www.lbdsoftware.com](http://www.lbdsoftware.com)) and on the CBTape ([cbttape.org](http://cbttape.org)). Among his contributions to the z/OS community are XMITIP and FTPBatch. Email: [lbdyck@gmail.com](mailto:lbdyck@gmail.com)

**William J. Smith**, with 42 years as a systems programmer, developer, and an M.A. in IT education, is recognized as a passionate college instructor, mentor, mainframe evangelist, technical writer, and SMP/E packaging SME. He served 11 years as IBM Lead Representative to SHARE for DFSMS storage development. Email: [sfowjs@gmail.com](mailto:sfowjs@gmail.com)