



CT04 - SAS® Keyboard Shortcuts Enhanced Using %Clipbrd() Macro Function

Jean-Michel Bodart, 10th October 2017



Introduction: SAS clipboard, keyboard shortcuts



The CLIPBOARD access method



Returning clipboard text with a macro-function



The user-written macro-function %Clipbrd()



Using macros in keyboard shortcuts



Conclusions

Introduction (1)

■ SAS® Clipboard on Windows®

■ Copy & paste text

- from/to almost any window & application
- via menus, toolbars and shortcuts

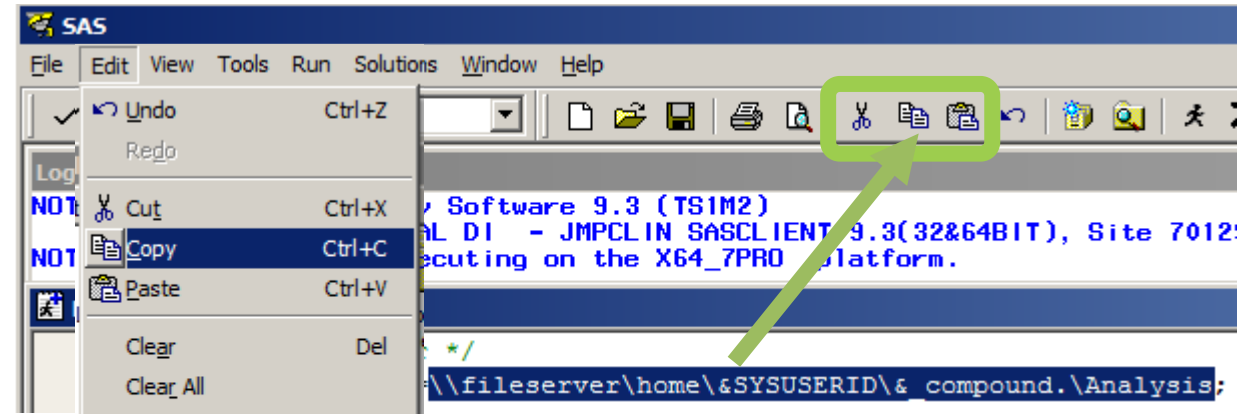
■ Keyboard shortcuts

1. Microsoft Windows

- Ctrl+C, Ctrl+V, Ctrl+X ...

2. SAS: KEYS Window shortcuts

- Common to Program Editor, Enhanced Editor, Log, Output, Explorer, Notepad
- F9 → SAS Command: KEYS
(or: Tools→Options→Keys)
- DM 'keydef ' 'SHF F6' '
' 'vt &syslast' ' ';



Key	Definition
F1	[help
F2	reshow
F3	end; /*gsubmit buffer=default*/
F4	recall
F5	wpgm
F6	log
F7	output
F8	zoom off:submit
F9	keys
F11	command focus
F12	
SHF F1	subtop
SHF F2	
SHF F6	vt &syslast
SHF F7	left
SHF F8	right
SHF F9	
SHF F10	wpopup
SHF F11	
SHF F12	
CTL F1	
CTL F2	
CTL F3	
CTL F11	
CTL F12	

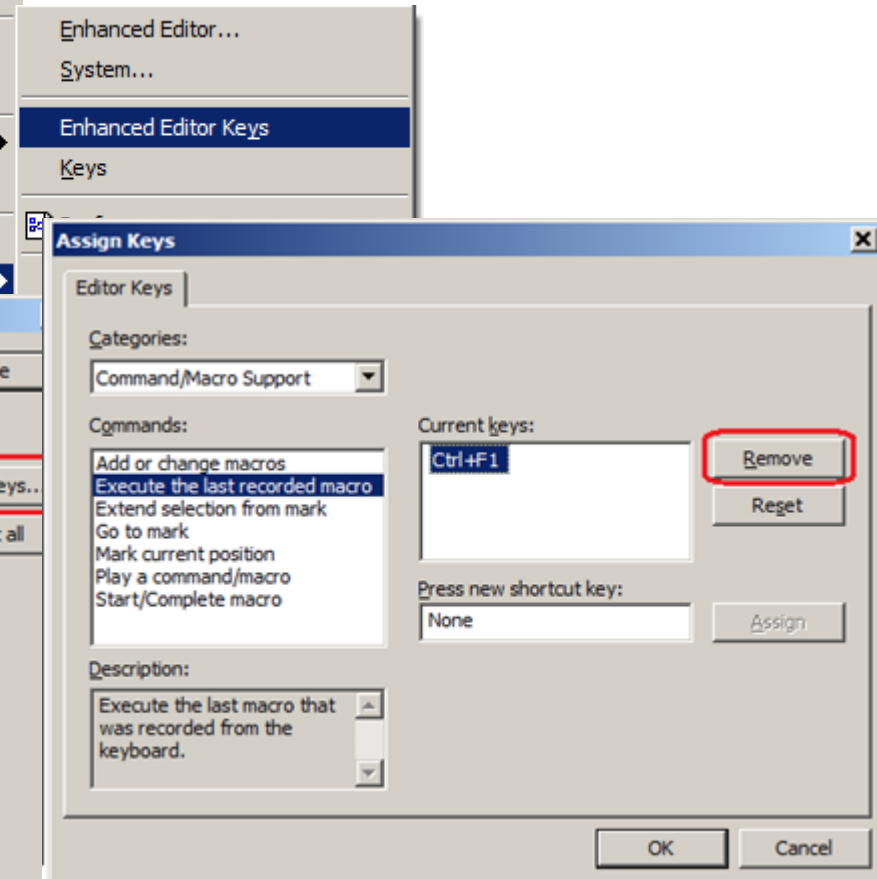
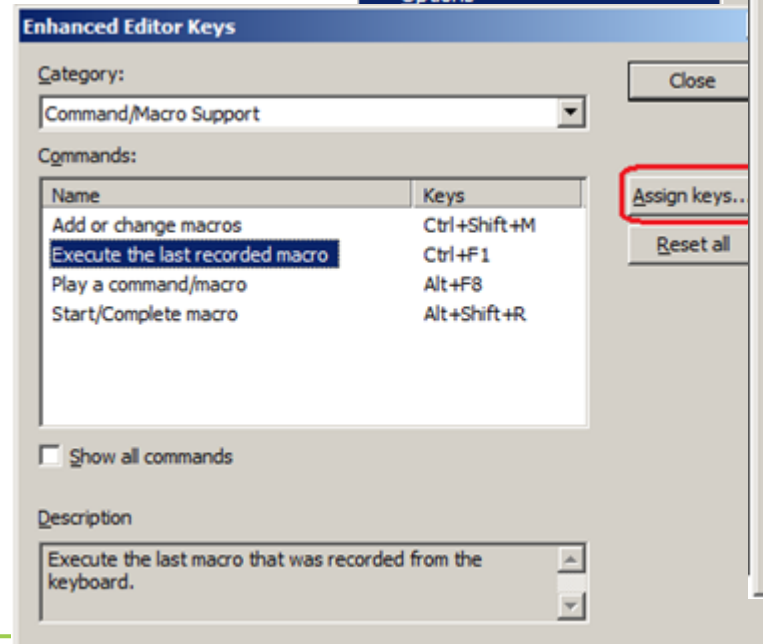
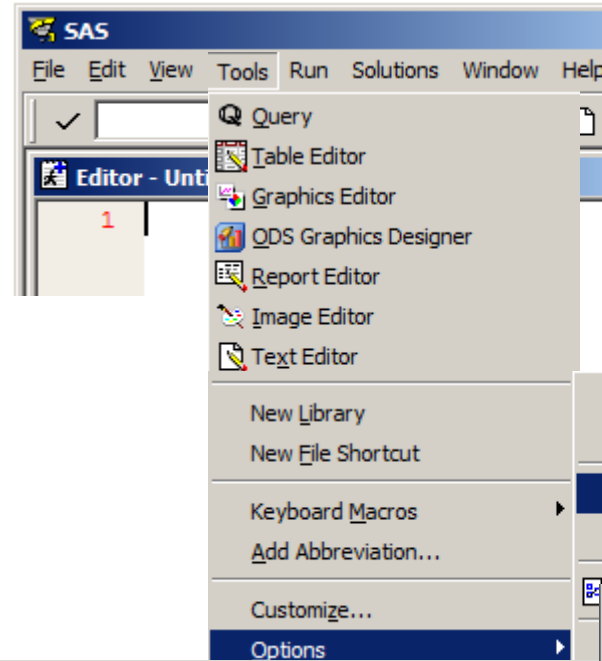
Key	Definition
ALT F1	
ALT F2	
ALT F3	
ALT F11	
ALT F12	
CTL B	libname
CTL D	dir
CTL E	clear
CTL G	
CTL H	help
CTL I	options
CTL J	
CTL K	cut
CTL L	log
CTL M	mark
CTL Q	filename
CTL R	rfind
CTL T	title
CTL U	unmark
CTL W	access
CTL Y	
RMB	wpopup
SHF RMB	
CTL RMB	
MMB	
SHF MMB	
CTL MMB	

Introduction (2)

■ Keyboard shortcuts

3. 'Keyboard macros'

- Enhanced Editor only
- Assigned/removed with menu: Tools → Options → Enhanced Editor Keys
- **Cannot** execute SAS commands nor SAS macros
- **Override** KEYS window shortcuts:
 - By default: Ctrl+F1, Ctrl+F2, Ctrl+G, Ctrl+Y, F1, F2, Shift+F2



The CLIPBOARD access method (1)

■ FILENAME fileref **CLIPBRD** <~~BUFFER=paste-buffer-name~~> ; BUFFER option not applicable under Windows

*Data step with infile & input statements:
line length limited to 256 characters
despite setting LRECL option to max*

```
option LRECL=32767;
filename myclip clipbrd;
data _null_;
    infile myclip length=len;
    length line $32767;
    input line $varying32767.
    len;
    put len= line=;
run;
```

External file functions (see online help) overcome this limitation

```
data _null_;
    rc = filename('clippy', ' ', 'clipbrd');
    fid = fopen('clippy', 's', 32767, 'V');
    length line $32767;
    do while(fread(fid) = 0);
        len = frlen( fid );
        rc = fget(fid, line, len);
        put len= line=;
    end;
    rc = fclose(fid);
    rc = filename('clippy');
run;
```

The CLIPBOARD access method (2)

```
%macro clip1;
  %local rc fileref fid len line;
  %let rc = %sysfunc(filename(fileref, , clipbrd));
  %if &rc eq 0 %then %do;
    %let fid = %sysfunc(fopen(&fileref, s, 32767, v));
    %if &fid ne 0 %then %do;
      %do %while(%sysfunc(fread(&fid))=0);
        %let len = %sysfunc(frlen(&fid));
        %let rc = %sysfunc(fget(&fid, line, &len));
        %put len = &len line = %superq(line);
      %end;
      %let rc = %sysfunc(fclose(&fid));
    %end; %else %put failed to open clipboard fileref.;
    %let rc = %sysfunc(filename(fileref));
  %end; %else %put %sysfunc(sysmsg( ));
%mend;
%clip1;
```

■ In macro:

■ External file functions

- Via %sysfunc()
- Quotes around text literals removed
- Filename() with empty 2nd argument

■ Additionally:

- Checks / error messages (sysmsg())
 - Error if keyboard contains no text
- Local macro-variables
 - prevent interferences outside the macro
- Macro quoting with %superq()
 - prevent macro processing

Returning Clipboard Text with a Macro-Function

```
%macro clip1;
  %local rc fileref fid len line return_val;
  %let rc = %sysfunc(filename(fileref, , clipbrd));
  %if &rc eq 0 %then %do;
    %let fid = %sysfunc(fopen(&fileref, s, 32767, v));
    %if &fid ne 0 %then %do;
      %do %while(%sysfunc(fread(&fid))=0);
        %let len = %sysfunc(frlen(&fid));
        %let rc = %sysfunc(fget(&fid, line, &len));
        %let return_val = &return_val %superq(line);
      %end;
      %let rc = %sysfunc(fclose(&fid));
    %end; %else %put failed to open clipboard fileref.;
    %let rc = %sysfunc(filename(fileref));
  %end; %else %put %sysfunc(sysmsg( ));
  &return_val
%mend;
%let cb=%clip1(); %put Clipboard text (upcased): %upcase(&cb);
```

■ The Clipboard Text is

- Not written to the log anymore
- Concatenated in macro-variable **return_val**

■ Final reference to &return_val

- left unprocessed by Macro Processor
- becomes the **returned value of %clip1()**
- which can be **stored** in a macro-variable and **processed further**, e.g. with **%upcase()**

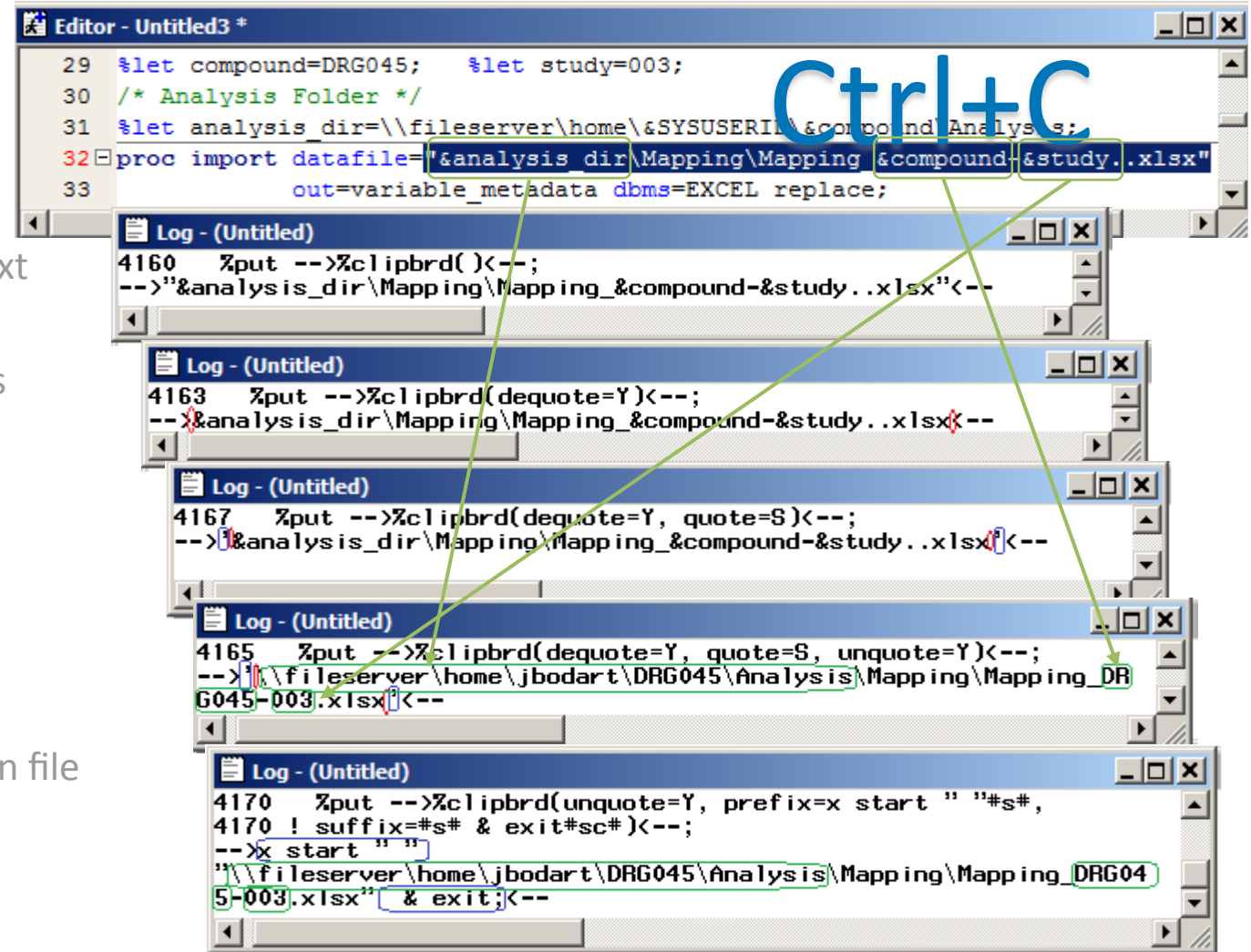
The Macro-Function %Clipbrd()

■ Based on similar code

+ optional parameters :

- **DEQUOTE = Y**
 - Suppress matching quotes around text
- **QUOTE = Y / S**
 - Enclose text in double / single quotes
- **UNQUOTE = Y / F**
 - Resolve macro references – Caution: unmatched symbols (, ', ",), &, %
 - F: delayed until final step
- **PREFIX = string**
- **SUFFIX = string**
 - Build a complete SAS command (open file in associated Windows application)

■ See paper / backup slides for more



```
Editor - Untitled3 *
29 %let compound=DRG045; %let study=003;
30 /* Analysis Folder */
31 %let analysis_dir=\\fileserver\home\&SYSUSERID\&compound\Analysis;
32 proc import datafile='&analysis_dir\Mapping\Mapping_&compound-&study..xlsx'
33 out=variable_metadata dbms=EXCEL replace;

Log - (Untitled)
4160 %put -->%clipbrd()<--;
-->'&analysis_dir\Mapping\Mapping_&compound-&study..xlsx'<--

Log - (Untitled)
4163 %put -->%clipbrd(dequote=Y)<--;
-->'&analysis_dir\Mapping\Mapping_&compound-&study..xlsx'<--

Log - (Untitled)
4167 %put -->%clipbrd(dequote=Y, quote=S)<--;
-->'&analysis_dir\Mapping\Mapping_&compound-&study..xlsx'<--

Log - (Untitled)
4165 %put -->%clipbrd(dequote=Y, quote=S, unquote=Y)<--;
-->'\\fileserver\home\jbodart\DRG045\Analysis\Mapping\Mapping_DRG045-003.xlsx'<--

Log - (Untitled)
4170 %put -->%clipbrd(unquote=Y, prefix=x start " "s#,
4170 ! suffix=#s# & exit#s#)<--;
-->x start " "
"\\fileserver\home\jbodart\DRG045\Analysis\Mapping\Mapping_DRG045-003.xlsx" & exit;<--
```

Macros in KEYS window shortcuts

Macro-variables

- **F12: vt &syslast**
 - Opens **last created dataset** in ViewTable
- **Ctrl+F1: x start " "&output" & exit;**
 - Opens **last created external file** in default windows application (if path **manually assigned** to macro-variable &output)

```
data librefs;  
  set sashelp.vslib;  
run;  
%let output = %sysfunc(pathname( work,  
L))\test.xls;  
proc export data=sashelp.class  
  outfile="&output" dbms=EXCEL  
  replace;  
  sheet = "Class";  
run;
```

Macro-functions

- **Ctrl+F2: vt %Clipbrd(unquote=F)**
 - Opens **any dataset or view** in ViewTable, name **copied from log/ program/ document**
- **Ctrl+F3: %Clipbrd(prefix=x start " "#s#, dequote=Y, quote=Y, unquote=F, suffix=& exit#sc#)**
 - Opens **any external file** in default windows application, filename **copied from any log/ program/ document**
 - with double-quoted filename
 - **macro- variables and functions resolved** (defined in current SAS session)
 - adding **PREFIX** and **SUFFIX** to return **complete commands** (with #s# and #sc# to be replaced by space and semicolon)

Conclusions

- **Principles for developing SAS macro-functions**

- *specific code example*

- **Macros can extend KEYS window shortcuts capabilities**

- *simple/complex actions*

- **Macro-function %Clipbrd() uses clipboard access method to:**

- *Retrieve clipboard text*

- **Return value used to:**

- *Build and execute specific commands*
 - open external files/ SAS items in appropriate applications

- **Full code on www.PhuseWiki.org**



- http://www.phusewiki.org/wiki/index.php?title=SAS_macro-function_%25CLIPBRD
- or search: %clipbrd
- shared under MIT license

- **For additional *features, parameters* and *examples*, see:**

- *Paper*
- *(Backup) slides*
- *PhUSE Wiki*



Questions...



ROI²
return on investment
through
return on information

 **Business & Decision**
Life **Sciences**

Back-up Slides

'Keyboard macros' (Enhanced Editor)

■ Examples (defaults):

- **F1**: Show context Help
- **Ctrl+F1**: Replay last sequence of keystrokes (recorded as a 'keyboard macro' with Alt+Shift+R)
- **Ctrl+F2**: Mark/ unmark current line
- **F2**: Go to next marked line
- **Shift+F2**: Go to previous marked line
- **Ctrl+G**: Go to specific line number
- **Ctrl+Y**: Redo last edit (previously undone with Ctrl+Z)

Note: shortcuts in **bold blue** override corresponding KEYS window shortcuts

■ Other categories:

- **code folding**
- **moving cursor**
- **extending selection**
- **selection operations** ((un)indent/, (un)comment, upper/lowercase)
- **inserting date/time**
- **find & replace**
- **toggle file attribute read-only/ writeable**

%Clipbrd() Parameters (Continued)

Other optional parameters

- **LINESEP = string** (default: 1 space)
 - Replace **newlines** in returned value by 0 or more **printable** characters
- **TABS = Y/n/[chars]/string**

Split lines into fields at separators:

 - **Y**: sequences of 6 spaces (*by which the clipboard access method replaces tabs*)
 - **n** (≥ 2): sequences of at least **n** spaces
 - **[chars]**: any character specified
 - **String**: the exact substring
- **FSEP = string** (default: 1 space)
 - Replace field separators in returned value by 0 or more **printable** characters
- **TR = H / V**
 - Transpose to Horizontal / Vertical format by replacing line $\leftrightarrow$ field separators

Ctrl+C

	A	B	C
1	PARAMCD	ALB	Albumin (g/L)
2	PARAMCD	ALBU	Albumin (g/dL)
3	PARAMCD	ALP	Alkaline Phosphatase (IU/L)
4	PARAMCD	ALPS	Alkaline Phosphatase (U/L)
5	PARAMCD	AMYLA	Amylase (U/L)
6	PARAMCD	BASO	Basophils (%)

```

Log - (Untitled)
4122 option ls=64;
4123 %put -->%clipbrd()<--;
-->PARAMCD    ALB    Albumin (g/L) PARAMCD    ALBU
Albumin (g/dL) PARAMCD    ALP    Alkaline Phosphatase
(IU/L) PARAMCD    ALPS    Alkaline Phosphatase (U/L)
PARAMCD    AMYLA    Amylase (U/L) PARAMCD    BASO

Log - (Untitled)
4126 option ls=64;
4127 %put -->%clipbrd(linesep=%str( | ))<--;
-->PARAMCD    ALB    Albumin (g/L) | PARAMCD    ALBU
Albumin (g/dL) | PARAMCD    ALP    Alkaline Phosphatase
(IU/L) | PARAMCD    ALPS    Alkaline Phosphatase (U/L) |
PARAMCD    AMYLA    Amylase (U/L) | PARAMCD    BASO

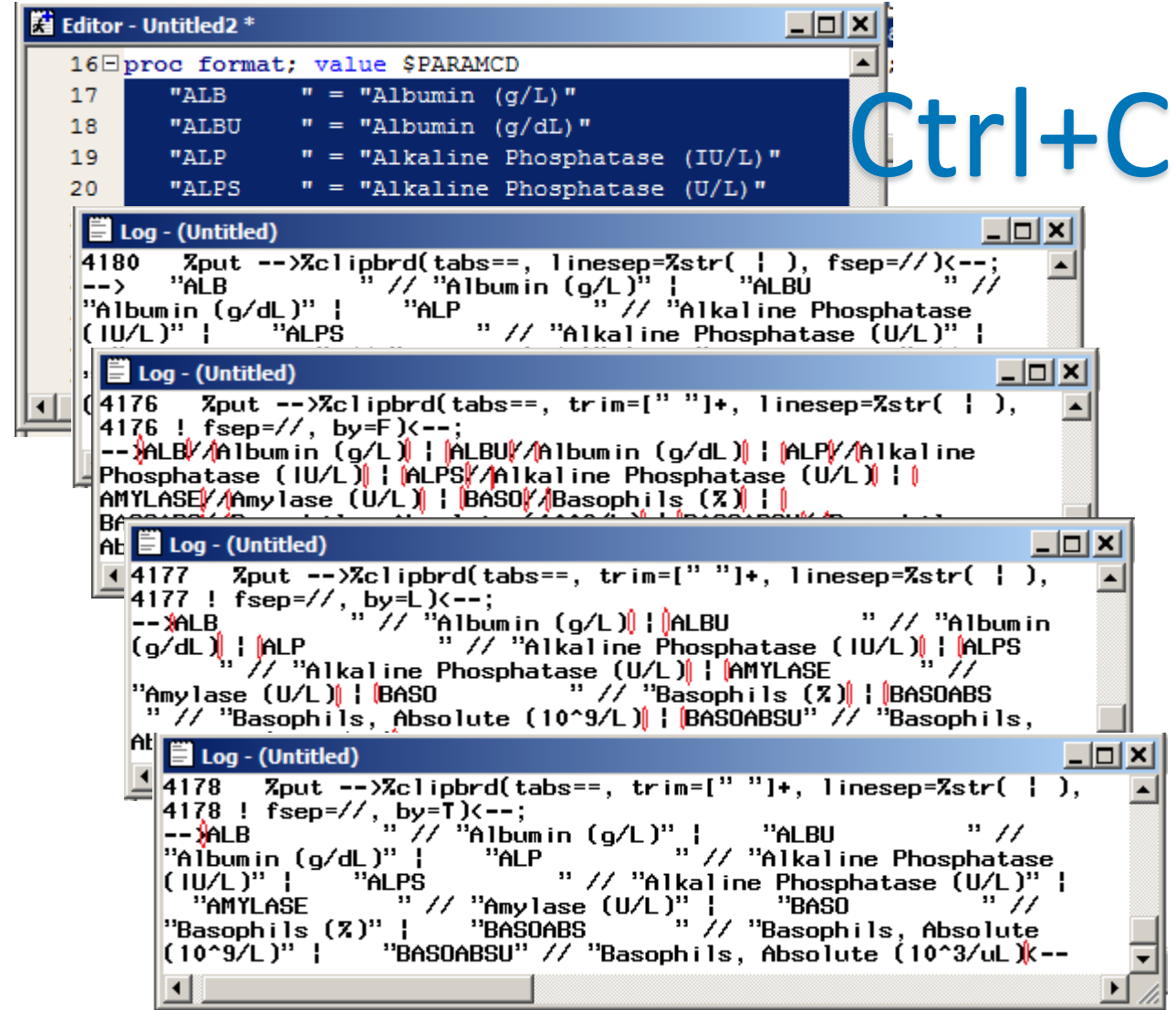
Log - (Untitled)
4134 option ls=64;
4135 %put -->%clipbrd(linesep=%str( | ), tabs=Y, fsep=//)<--;
-->PARAMCD//ALB//Albumin (g/L) | PARAMCD//ALBU//Albumin (g/dL)
| PARAMCD//ALP//Alkaline Phosphatase (IU/L) |
PARAMCD//ALPS//Alkaline Phosphatase (U/L) |
PARAMCD//AMYLA//Amylase (U/L) | PARAMCD//BASO//Basophils (%)
| PARAMCD//BASOABS//Basophils, Absolute (10^9/L) |
PARAMCD//BASOABSU//Basophils, Absolute (10^3/uL)<--

Log - (Untitled)
4138 option ls=64;
4139 %put %clipbrd(linesep=%str( | ), tabs=Y, fsep=//, tr=V);
PARAMCD | ALB | Albumin (g/L) | PARAMCD | ALBU | Albumin (g/dL)
| PARAMCD | ALP | Alkaline Phosphatase (IU/L) | PARAMCD | ALPS
| Alkaline Phosphatase (U/L) | PARAMCD | AMYLA | Amylase
(U/L) | PARAMCD | BASO | Basophils (%) | PARAMCD | BASOABS |
    
```

%Clipbrd() Parameters (Continued)

■ More optional parameters:

- *LTRIM, RTRIM, TRIM = [chars] (+)/string/perl reg. expr.*
 - Remove character(s)/ strings/ patterns matching RegExp from left, right, or both sides of the text
- *PRXCHG = Perl Regular Expr.*
 - s/search_pattern/replace_text/
- *BY = F / L / T* (default: L)
 - Perform transforms (L/R/TRIM, PRXCHG, PREFIX, SUFFIX, UNQUOTE, QUOTE, DEQUOTE) on individual Fields (F), Lines (L) or Total (T) clipboard contents



Ctrl+C

```
16 proc format; value $PARAMCD
17   "ALB"   = "Albumin (g/L)"
18   "ALBU"  = "Albumin (g/dL)"
19   "ALP"   = "Alkaline Phosphatase (IU/L)"
20   "ALPS"  = "Alkaline Phosphatase (U/L)"

4180 %put -->%clipbrd(tabs==, linesep=%str( | ), fsep=//)<--;
--> "ALB" // "Albumin (g/L)" | "ALBU" //
"Albumin (g/dL)" | "ALP" // "Alkaline Phosphatase
(IU/L)" | "ALPS" // "Alkaline Phosphatase (U/L)" |

4176 %put -->%clipbrd(tabs==, trim=" ")+, linesep=%str( | ),
4176 ! fsep=//, by=F)<--;
--> ALB Albumin (g/L) | ALBU Albumin (g/dL) | ALP Alkaline
Phosphatase (IU/L) | ALPS Alkaline Phosphatase (U/L) |
AMYLAASE Amylase (U/L) | BASO Basophils (%) |
BASOABS Basophils, Absolute (10^9/L) | BASOABSU Basophils, Absolute (10^3/uL)

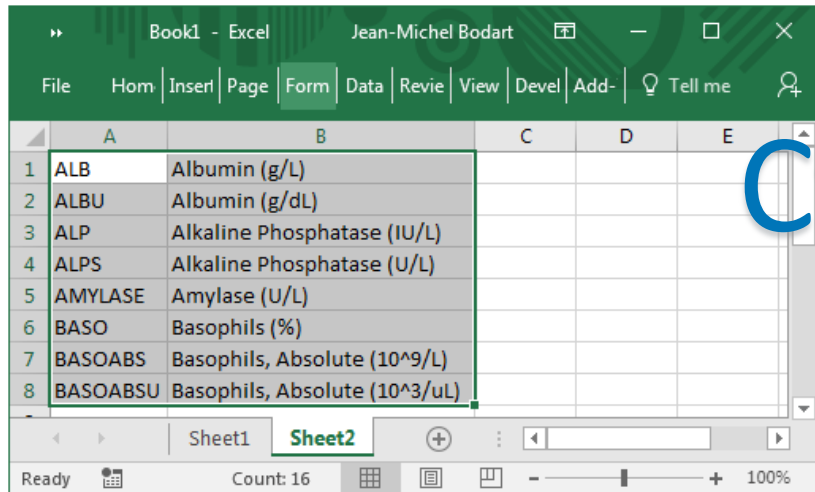
4177 %put -->%clipbrd(tabs==, trim=" ")+, linesep=%str( | ),
4177 ! fsep=//, by=L)<--;
--> ALB // "Albumin (g/L)" | ALBU // "Albumin
(g/dL)" | ALP // "Alkaline Phosphatase (IU/L)" | ALPS
// "Alkaline Phosphatase (U/L)" | AMYLAASE //
"Amylase (U/L)" | BASO // "Basophils (%)" | BASOABS
// "Basophils, Absolute (10^9/L)" | BASOABSU // "Basophils, Absolute (10^3/uL)"

4178 %put -->%clipbrd(tabs==, trim=" ")+, linesep=%str( | ),
4178 ! fsep=//, by=T)<--;
--> ALB // "Albumin (g/L)" | ALBU //
"Albumin (g/dL)" | ALP // "Alkaline Phosphatase
(IU/L)" | ALPS // "Alkaline Phosphatase (U/L)" |
AMYLAASE // "Amylase (U/L)" | BASO //
"Basophils (%)" | BASOABS // "Basophils, Absolute
(10^9/L)" | BASOABSU // "Basophils, Absolute (10^3/uL)"
```

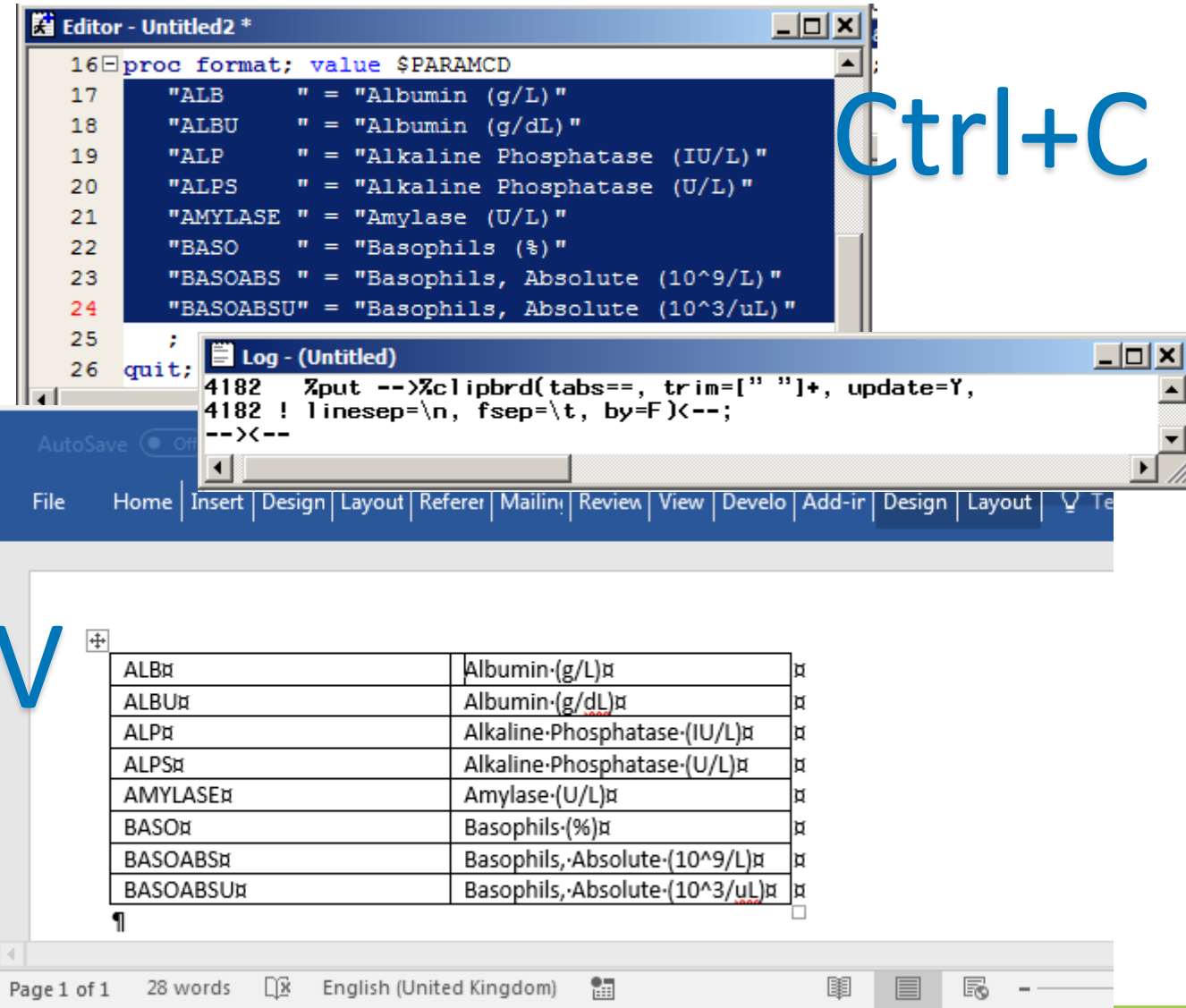
%Clipbrd() further enhancements (1)

■ **Write modified contents back to clipboard, return no value:**

- Through **same fileref** with external file functions `fput()` and `fwrite()`
- **Parameter: UPDATE = Y**
 - `LINESEP=\n` : re-creates newlines
 - `FSEP=\t` : re-creates tabs
 - Requires special macro technique
 - paste into spreadsheet/ Word table



1	ALB	Albumin (g/L)
2	ALBU	Albumin (g/dL)
3	ALP	Alkaline Phosphatase (IU/L)
4	ALPS	Alkaline Phosphatase (U/L)
5	AMYLASE	Amylase (U/L)
6	BASO	Basophils (%)
7	BASOABS	Basophils, Absolute (10^9/L)
8	BASOABSU	Basophils, Absolute (10^3/uL)



Ctrl+C

```
16 proc format; value $PARAMCD
17   "ALB"   = "Albumin (g/L)"
18   "ALBU"  = "Albumin (g/dL)"
19   "ALP"   = "Alkaline Phosphatase (IU/L)"
20   "ALPS"  = "Alkaline Phosphatase (U/L)"
21   "AMYLASE" = "Amylase (U/L)"
22   "BASO"   = "Basophils (%)"
23   "BASOABS" = "Basophils, Absolute (10^9/L)"
24   "BASOABSU" = "Basophils, Absolute (10^3/uL)"
25 ;
26 quit;
```

Ctrl+V

```
4182 %put -->%clipbrd(tabs==, trim=[" "], update=Y,
4182 ! linesep=\n, fsep=\t, by=F)<--;
--><--
```

ALB	Albumin (g/L)
ALBU	Albumin (g/dL)
ALP	Alkaline Phosphatase (IU/L)
ALPS	Alkaline Phosphatase (U/L)
AMYLASE	Amylase (U/L)
BASO	Basophils (%)
BASOABS	Basophils, Absolute (10^9/L)
BASOABSU	Basophils, Absolute (10^3/uL)

%Clipbrd() further enhancements (2)

■ Check external file/ SAS item exists, of specified type

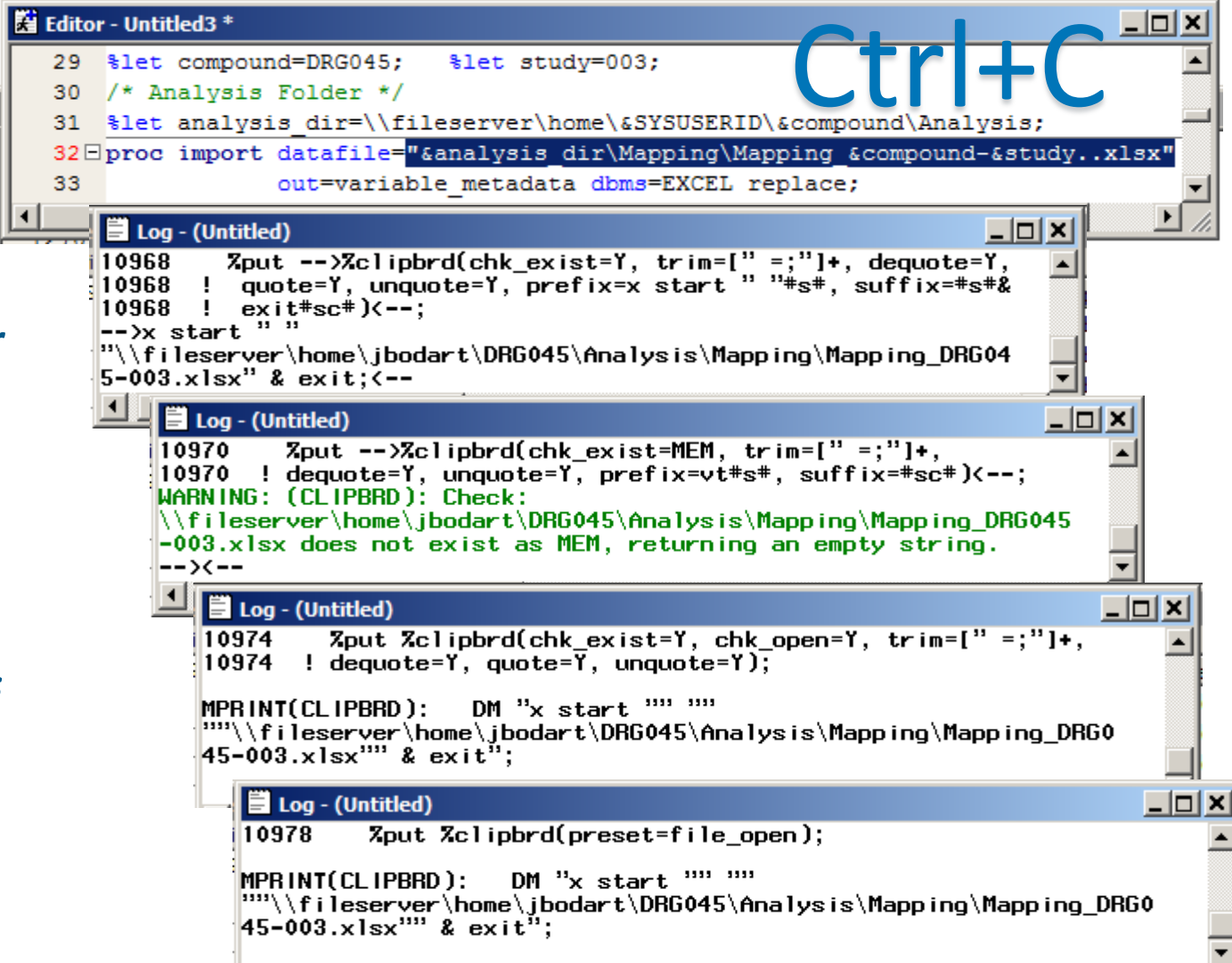
- Parameter: *CHK_EXIST* = Y/N/EXT/SAS/MEM/DATA/VIEW/CAT/ACCESS/CENTRY/LIBREF/FILEREF/URL/centry_type

■ Insert preferred “open” command for file/ item type in return value

- Parameter: *CHK_OPEN* = Y/N/WIN/EXPL/SAS
 - Overrides PREFIX and SUFFIX parameters
 - Supports multiple files (BY=F/L)

■ Presets of common parameter values

- Parameter: *PRESET* = file / file_open / file_SAS_open / file_EXT_open / file_path_open



The screenshot displays a SAS Editor window titled 'Editor - Untitled3 *' and three Log windows titled 'Log - (Untitled)'. A large blue 'Ctrl+C' watermark is overlaid on the top right. The Editor window shows the following code:

```
29 %let compound=DRG045; %let study=003;
30 /* Analysis Folder */
31 %let analysis_dir=\\fileserver\home\&SYSUSERID\&compound\Analysis;
32 proc import datafile="&analysis_dir\Mapping\Mapping &compound-&study.xlsx"
33 out=variable_metadata dbms=EXCEL replace;
```

The first Log window shows the execution of %Clipbrd() with parameters chk_exist=Y, trim=["="], dequote=Y, quote=Y, unquote=Y, prefix=x start, and suffix=#s#. The output is the file path: '\\fileserver\home\jbodart\DRG045\Analysis\Mapping\Mapping_DRG045-003.xlsx'.

The second Log window shows the execution of %Clipbrd() with parameters chk_exist=MEM, trim=["="], dequote=Y, quote=Y, unquote=Y, prefix=vt#s#, and suffix=#sc#. A warning is displayed: 'WARNING: (CLIPBRD): Check: \\fileserver\home\jbodart\DRG045\Analysis\Mapping\Mapping_DRG045-003.xlsx does not exist as MEM, returning an empty string.'

The third Log window shows the execution of %Clipbrd() with parameters chk_exist=Y, chk_open=Y, trim=["="], dequote=Y, quote=Y, and unquote=Y. The output is the file path: '\\fileserver\home\jbodart\DRG045\Analysis\Mapping\Mapping_DRG045-003.xlsx'.

The fourth Log window shows the execution of %Clipbrd() with the parameter preset=file_open. The output is the file path: '\\fileserver\home\jbodart\DRG045\Analysis\Mapping\Mapping_DRG045-003.xlsx'.

%Clipbrd() further enhancements (3)

■ Write modified contents back to clipboard, return no value:

- Parameter *UNIFW = Y*
 - space-padding field values to max field width
 - paste as well-aligned columns in character-based text editor
- *PRESET= table_to_text*

Ctrl+C

Ctrl+V

ALB	Albumin·(g/L)
ALBU	Albumin·(g/dL)
ALP	Alkaline·Phosphatase·(IU/L)
ALPS	Alkaline·Phosphatase·(U/L)
AMYLASE	Amylase·(U/L)
BASO	Basophils·(%)
BASOABS	Basophils, Absolute (10 ⁹ /L)
BASOABSU	Basophils, Absolute (10 ³ /uL)

```
8536 %put -->%clipbrd(tabs=Y, update=Y, linesep=\n, fsep=%str(8536! | ), unifw=Y, by=F, confirm=N)<--;
--><--
```

```
1 ALB | Albumin (g/L)
2 ALBU | Albumin (g/dL)
3 ALP | Alkaline Phosphatase (IU/L)
4 ALPS | Alkaline Phosphatase (U/L)
5 AMYLASE | Amylase (U/L)
6 BASO | Basophils (%)
7 BASOABS | Basophils, Absolute (10^9/L)
8 BASOABSU | Basophils, Absolute (10^3/uL)
9 |
```

%Clipbrd() further enhancements (4)

■ Review and edit parameter values at invocation

- *Parameter: CONFIRM = Y/
list of parameter names*

➤ Unique shortcut for a variety of tasks:

`%clipbrd(preset=file,
confirm=Y)`

The screenshot shows a 'CONFIRM' dialog box with a title bar. Below the title bar, it says 'Command ===>' and 'Review, Edit and Confirm the values of CLIPBRD Parameters:'. A line of asterisks separates the title from the list of parameters. The parameters are listed on the left, and their current values are in text boxes on the right. The parameters and their values are: prefix (x start "'#s#), suffix (#s#& exit#sc#), linesep (empty), fsep (empty), tabs (Y), by (T), prxchg (s/(\w\\)\|\\|1/), tr (empty), un ifw (empty), dequote (Y), quote (Y), unquote (Y), update (N), ltrim (['"= '"]+), rtrim (['"., ;)"]+), trim (empty), trim_by (empty), chk_exist (empty), chk_open (empty), and chk_open_sttmt (Y). At the bottom, it says 'Click or use [Tab] to move between fields. [Pg Down] when done.'

Parameter	Value
prefix	x start "'#s#
suffix	#s#& exit#sc#
linesep	
fsep	
tabs	Y
by	T
prxchg	s/(\w\\)\ \\ 1/
tr	
un ifw	
dequote	Y
quote	Y
unquote	Y
update	N
ltrim	['"= '"]+
rtrim	['"., ;)"]+
trim	
trim_by	
chk_exist	
chk_open	
chk_open_sttmt	Y

My own shortcuts using %Clipbrd()

■ **Ctrl+E: %clipbrd(file, wedit#s#, #scs#)**

- *Open external file in a new Enhanced Editor window*

■ **Ctrl+L: submit**
'%_viewlog(%clipbrd(dequote=Y, unquote=Y, trim=[" "]+));'

- *Create a coloured HTML version from a saved log file, with a summary of any Errors, Warnings, Notes of interest and hyperlinks to their actual location in full log*

■ **Ctrl+W: %clipbrd(file_ext_open)**

- *Open external file in associated default windows application*

■ **Ctrl+Q: %clipbrd(file_sas_open)**

- *Open SAS file (dataset, view, catalog, log, program...) in preferred applicable SAS window*

*Note the use of presets, and of **positional parameters** instead of named parameters – to shorten the macro calls*