## Untranspose Macro Tip Sheet

**Purpose**: The macro untransposes wider SAS datasets back to either the less wide state that existed before the file was transposed, or to a long file. The macro can accommodate any prefixes, variable names, delimiters, ID values, and suffixes that may exist in the transposed variable names.

Named Parameters: The macro uses named parameters so that (1) default values can be assigned and (2) the various parameters only have to be specified when values other than the default values are required. We attempted, as closely as possible, to use the same option names and statements as those used for PROC TRANSPOSE. When calling the macro, the default values will be used unless you specify the desired value. Thus, if you wanted the macro to typically get your data from a libname called mydata, you would modify the parameter by specifying it in the macro declaration.

Parameter	Required	Possible Values	Default	Description
			Value	
libname_in	No	Any valid libname	work	The parameter to which you can assign the name of the SAS library that contains the dataset you want to untranspose
libname_out	No	Any valid libname	work	The parameter to which you can assign the name of the SAS library where you want the untransposed file written
data	Yes	Any valid filename	None	The parameter to which you assign the one or two-level name of the file that you want to untranspose
out	Yes	Any valid filename	None	The parameter to which you assign the name of the file that you want the macro to create
by	No	Any variable name from the file specified in the data parameter	None	The parameter to which you would assign the name of the original dataset's by variable(s)
prefix	No	Any valid SAS name characters	None	The parameter to which you assign the string (if any) that the transposed variable names begin with
var	Yes	Any valid SAS name	None	The parameter to which you assign the name(s) of the original variables that had been transposed
id	No	Any valid SAS name	None	The parameter to which you specify the variable name that was used as the ID variable (if any) when the transposed file was created. Only one variable can be assigned
Id_informat	No	Any valid SAS informat	8.	The parameter to which you can assign the informat to be used to extract the id variable's values
Id_format	No	Any valid SAS format	8.	The parameter to which you can indicate the format you want assigned to the id variable
var_first	No	YES= <pre>YES=<pre>YES=<pre>cycle</pre>prefix&gt;var<delimiter>id<suffix></suffix></delimiter></pre>N/A=<pre>Cycle</pre>prefix&gt;var<suffix></suffix></pre>	Yes	The parameter that defines whether var names precede id values in the transposed variable names
delimiter	No	Any valid SAS name characters	None	The parameter to which you assign the string (if any) that was used to separate var and ID values in the transposed variable names
suffix	No	Any valid SAS name characters	None	The parameter to which you can assign a string (if any) that the transposed variable names end with
сору	No	Any valid SAS name	None	The parameter to which you can assign the name(s) of any variables that had been copied
missing	No	Yes or no (case insensitive)	No	The parameter to indicate whether a record should be output if the only non-missing variables are the BY, ID and COPY variables
metadata	No	Any valid filename	None	The parameter to which you can specify the one or two-level SAS dataset the you want created to contain the untransposed variables' metdata
makelong	No	Yes or no (case insensitive)	No	The parameter to which you can specify that you want the macro to output records at the BY variable, ID variable value, var variable(s) level
max_length	No	Any number between 1 and 32767	None	The parameter to which you can specify the length of the _value_ variable
create_byvar	No	Any valid SAS variable name	None	The parameter to which you can have the macro create a by variable and assign sequential numbers

**Usage Examples**: The following are some examples of how you might use the macro. For each example the wide dataset's name is *have* and resides in the work library, and the less wide or long dataset created by the macro is called *want* and also resides in the work library.

dataset: *have* 

id	income2015	income2016	income2017	expenses2015	expenses2016	expenses2017	debt2015	debt2016	debt2017
1	70000	75500	80000	60000	70000	81000	no	no	yes
2	50000	52000	55000	42000	53000	60000	no	yes	yes
3	80000	90000	99000	70000	75000	85000	no	no	no

dataset: want

id	year	income	expenses	debt
1	2015	70000	60000	no
1	2016	75500	70000	no
1	2017	80000	81000	yes
2	2015	50000	42000	no
2	2016	52000	53000	yes
2	2017	55000	60000	yes
3	2015	80000	70000	no
3	2016	90000	75000	no
3	2017	99000	85000	no



macro call: %untranspose(data=have, out=want, by=id, id=year, var=income expenses debt)

dataset: *have* 

	id	income2015	income2016	income2017	expenses2015	expenses2016	expenses2017	debt2015	debt2016	debt201
	1	70000	75500	80000	60000	70000	81000	no	no	yes
	2	50000	52000	55000	42000	53000	60000	no	yes	yes
l	3	80000	90000	99000	70000	75000	85000	no	no	no



id	year	income	expenses	debt
1	2015	70000	60000	no
1	2016	75500	70000	no
1	2017	80000	81000	yes



macro call: **%untranspose**(data=have (obs=1), out=want, by=id, id=year, var=income expenses debt)

dataset: *have* 

id	income2015	income2016	income2017	expenses2015	expenses2016	expenses2017	debt2015	debt2016	debt2017
1	70000	75500	80000	60000	70000	81000	no	no	yes
2	50000	52000	55000	42000	53000	60000	no	yes	yes
3	80000	90000	99000	70000	75000	85000	no	no	no

dataset: want

id	year	income	expenses	debt
2	2015	50000	42000	no
2	2016	52000	53000	yes
2	2017	55000	60000	yes



macro call: %untranspose(data=have (where=(id eq 2)), out=want, by=id, id=year, var=income expenses debt)

dataset: *have* dataset: *want* 

id	income_2015	income_2016	income_2017	expenses_2015	expenses_2016	expenses_2017	debt_2015	debt_2016	debt_2017
1	70000	75500	80000	60000	70000	81000	no	no	yes
2	50000	52000	55000	42000	53000	60000	no	yes	yes
3	80000	90000	99000	70000	75000	85000	no	no	no

id year income expenses debt 1 2015 70000 60000 no 1 2016 75500 70000 no 1 2017 80000 81000 yes 2 2015 50000 42000 no 2 2016 52000 53000 yes 60000 yes 2 2017 55000 3 2015 80000 70000 no 3 2016 90000 75000 no 3 2017 99000 85000 no

macro call: %untranspose(data=have, out=want, by=id, delimiter=\_, id=year, var=income expenses debt)

dataset: have

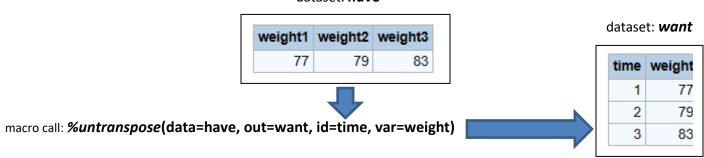
adiasei. <i>Nave</i>	dataset: <b>want</b>

id	income_2015	income_2016	income_2017	expenses_2015	expenses_2016	expenses_2017	debt_2015	debt_2016	debt_2017
1	70000	75500	80000	60000	70000	81000	no	no	yes
2	50000	52000	55000	42000	53000	60000	no	yes	yes
3	80000	90000	99000	70000	75000	85000	no	no	no

id year debt income expenses 1 2015 no 70000 60000 70000 1 2016 no 75500 1 2017 yes 80000 81000 2 2015 no 50000 42000 2 2016 yes 52000 53000 55000 60000 2 2017 yes 3 2015 no 80000 70000 90000 75000 3 2016 no 3 2017 no 99000 85000

macro call: %untranspose(data=have, out=want, by=id, delimiter=\_, id=year, var=debt income expenses)

dataset: have



dataset: *have* dataset: *want* 

income2015	income2016	income2017	expenses2015	expenses2016	expenses2017	debt2015	debt2016	debt2017
70000	75500	80000	60000	70000	81000	no	no	yes
50000	52000	55000	42000	53000	60000	no	yes	yes
80000	90000	99000	70000	75000	85000	no	no	no



macro call: %untranspose(data=have, out=want, id=year, var=income expenses debt, create\_byvar=id)

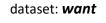
id	year	income	expenses	deb
1	2015	70000	60000	no
1	2016	75500	70000	no
1	2017	80000	81000	yes
2	2015	50000	42000	no
2	2016	52000	53000	yes
2	2017	55000	60000	yes
3	2015	80000	70000	no
3	2016	90000	75000	no
3	2017	99000	85000	no

dataset: *have* 

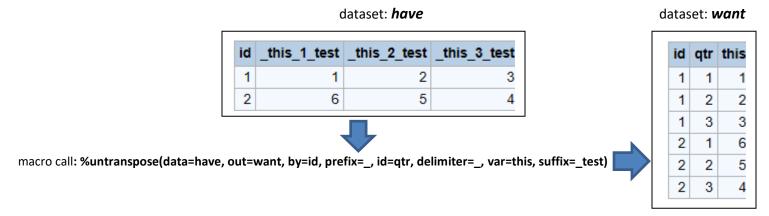
Promotion	CashBak	SavBal	CheckBal	InvestAmt
Gift	75	106	100	106
Gift	75	104	101	102
Purchase	125	114	102	108
Purchase	125	100	99	102



macro call: **%untranspose**(data=have, copy=Promotion CashBak, out=want(rename=(\_name\_=response \_value\_=value1)), var=SavBal CheckBal, create\_byvar=subject)



Promotion	CashBak	response	value1	subject
Gift	75	SavBal	106	1
Gift	75	CheckBal	100	1
Gift	75	InvestAmt	108	1
Gift	75	SavBal	104	2
Gift	75	CheckBal	101	2
Gift	75	InvestAmt	102	2
Purchase	125	SavBal	114	3
Purchase	125	CheckBal	102	3
Purchase	125	InvestAmt	108	3
Purchase	125	SavBal	100	4
Purchase	125	CheckBal	99	4
Purchase	125	InvestAmt	102	4

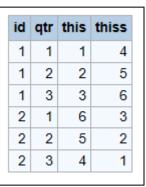


dataset: *have* dataset: *want* 

id	_this_1_test	_this_2_test	_this_3_test	_thiss_1_test	_thiss_2_test	_thiss_3_test
1	1	2	3	4	5	6
2	6	5	4	3	2	1

}

macro call: **%untranspose(data=have, out=want, by=id, prefix=\_, id=qtr, delimiter=\_, var=this thiss, suffix=\_test)** 



dataset: *have* 

id	_1_this_test	_2_this_test	_3_this_test	_1_thiss_test	_2_thiss_test	_3_thiss_test
1	1	2	3	4	5	6
2	. 6	5	4	3	2	1

4

macro call: %untranspose(data=have, out=want, by=id, prefix=\_, id=qtr, delimiter=\_, var\_first=no,var=this thiss, suffix=\_test)

dataset: want

id	qtr	this	thiss
1	1	1	4
1	2	2	5
1	3	3	6
2	1	6	3
2	2	5	2
2	3	4	1

dataset: *have* 

id thisA thisB t

hisC	thisislongerA	thisislongerB	thisislongerC
3	D	Е	F

В

dataset: **want** 

id	section	this	thisislonge
1	Α	1	D
1	В	2	Е
1	С	3	F
2	Α	6	С
2	В	5	В

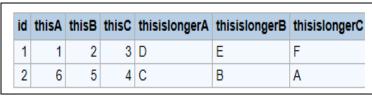
4 A

1

4 C

macro call: %untranspose(data=have, out=want, by=id, id=section, var=this thisislonger, id\_informat=\$1.,id\_format=\$1.)

dataset: *have* 





macro call: %untranspose(data=have(keep=id thisA--thisC), out=want, by=id, id=section, var=this, id\_informat=\$1.,id\_format=\$1.)

dataset: want

2 C

id	section	this
1	Α	1
1	В	2
1	С	3
2	Α	6
2	В	5
2	С	4

dataset: *have* 



customer	_0	_1	_2	_3	_4	_5	_6
1	herring	corned beef	olives	ham	turkey	bourbon	ice cream
2	corned beef	peppers	bourbon	crackers	chicken	ice cream	ice cream

macro call: **%untranspose(data=have, out=want, id=time, prefix=\_, var\_first=n/a, var=product, id\_informat=1.0,id\_format=1.0,by=customer)** 

dataset: want

customer	time	product
1	0	herring
1	1	corned beef
1	2	olives
1	3	ham
1	4	turkey
1	5	bourbon
1	6	ice cream
2	0	corned beef
2	1	peppers
2	2	bourbon
2	3	crackers
2	4	chicken
2	5	ice cream
2	6	ice cream

## dataset: *have*

```
proc format;
  value n
  1='AA'
  2='BB'
  3='CC'
 value $c
  'A'='11'
 'B'='22'
run;
data have;
 length subject 8;
 label var1='first var'
        var2='second var'
        var3='third var'
       var4='fourth var'
        var5='fifth var'
        var6='sixth var'
  format var2 n.
        var3 comma6.
        var4 $c.;
 input subject var1-var3 (var4-var6) ($);
  cards;
1 1 2 30000 A B this
2 3 2 10000 B A that
```



Obs	subject	var1	var2	var3	var4	var5	var6
- 1	1	1	BB	30,000	11	В	this
2	2	3	BB	10,000	22	Α	that



## dataset: want

Obs	aubject	_name_	_value_
1	1	var1	1
2	1	var2	2
3	1	var3	30000
4	1	var4	A
5	1	var5	В
6	1	var6	this
7	2	var1	3
8	2	var2	2
9	2	var3	10000
10	2	var4	В
11	2	var5	A
12	2	var6	that

## dataset: meta

)ba	_name_	_format	_Informat_	_label_	_length_	_type_
1	var1			first var	8	num
2	var2	N.		second var	8	num
3	var3	COMMA6.		third var	8	num
4	var4	\$C.		fourth var	2	char
5	var5			fifth var	8	char
6	var6			sixth var	8	char

macro call:%untranspose(data=have, out=want, var=var1-var6, by=subject, metadata=meta)