

# m\_val\_chk\_invalid.sas File Reference

## Validation

Validation macro to check data by using an invalid values rule

---

### Description

The macro can be used to validate data by using a selection of rules to detect invalid values of a column in a given database table or SAS dataset. There are multiple ways to build a rule. A rule can either be build by using a list of valid character or numerical values separated by a blank character (VAL\_LIST), or using a valid value range by providing a minimum and maximum value (VAL\_MIN and VAL\_MAX). Otherwise build a rule by using the minimum or maximum value independantly as greater-than-or-equal or lesser-than-or-equal respectively. The COL\_NAME parameter may contain only one column from the given source table. The routine will detect the existance of the column in the source table automatically. The column is stored including its value in the exception table. If the output target, error and/or exception tables are not defined by their according parameter values, they will be defined by the macro routine and created in the sessions WORK library. If an output exception table is given, the exceptions found by the validation will be appended to the given exception table. The same procedure is used for the error table just as long as the ACTION parameter value is not "Check".

### Note

*If the PRINT parameter value is set to Y, a SAS proc report step is used to print the validation summary status on the result tab of SAS Enterprise Guide or Stored Process Server.*

### Autors

Paul Alexander Canals y Trocha (paul.canals@gmail.com)

### Date

2021-03-27 00:00:00

### Version

21.1.03

### Link

<https://github.com/paul-canals/toolbox>

## Parameters

Input	help	Parameter, if set (Help or ?) to print the Help information in the log. In all other cases this parameter should be left out from the macro call.
Input	src_tbl	Specifies the LIBNAME.TABLENAME of the source SAS dataset or database table on which the validation is performed. The default value is: _NONE_.
Input	tgt_tbl	Specifies the LIBNAME.TABLENAME of the target SAS dataset or database table in which the valid data is stored. If this target table is not provided, this target table is created automatically in the WORK library. The default value is: _NONE_.
Input	err_tbl	Specifies the LIBNAME.TABLENAME of the target SAS dataset or database table in which the error data is stored. If this target table is not provided, this target table is created automatically in the WORK library. The default value is: _NONE_.
Input	exc_tbl	Specifies the LIBNAME.TABLENAME of the target SAS dataset or database table in which the exceptions are stored. If this target table is not provided, this target table is created automatically in the WORK library. The default value is: _NONE_.
Input	col_name	Mandatory parameter to specify the column name. This column must exist in the SRC_TBL table.
Input	val_list	Specifies a non quoted list of valid character or numerical values delimited by a blank character. If this parameter has been given a value it takes precedence over the VAL_MIN & VAL_MAX parameters.
Input	val_min	Specifies a character or numerical minimum valid value for the COL_NAME column in the source table. The COL_NAME column value should be greater than or equal to the VAL_MIN value. If the VAL_MAX is also set, the combined VAL_MIN and VAL_MAX values are used as COL_NAME between VAL_MIN and VAL_MAX.
Input	val_max	Specifies a character or numerical maximum valid value for the COL_NAME column in the source table. The COL_NAME column value should be lesser than or equal to the VAL_MAX value. If the VAL_MIN is also set, the combined VAL_MIN and VAL_MAX values are used as COL_NAME between VAL_MIN and VAL_MAX.
Input	action	Indicator [CHECK MOVE ABORT] to specify the to be taken by the validation routine. When the ACTION value is set to "Check", the source table records are validated and found exceptions written to the exception table, but also all records will be written to the output target table. When set to "Move", the error records are written to the error table and only valid records to the target table. When set to "Abort" only the found exceptions are written to the exception table, but not into any other target tables. The default value is: Check.
Input	append	Boolean [Y N] parameter value to specify if the found exceptions (and errors) are to be appended into a given exception table (and error table). The default value for APPEND is: Y.
Input	print	Boolean [Y N] parameter to generate the output by a SAS proc report step with style HtmlBlue. The default value for PRINT is: N.
Input	debug	Boolean [Y N] parameter to provide verbose mode information. The default value is: N.

## Returns

- Validation result target tables.

## Calls

- [m\\_utl\\_chg\\_delimiter.sas](#)
- [m\\_utl\\_chk\\_reqcols.sas](#)
- [m\\_utl\\_chk\\_table\\_exist.sas](#)
- [m\\_utl\\_get\\_col\\_code.sas](#)
- [m\\_utl\\_get\\_col\\_type.sas](#)
- [m\\_utl\\_nlobs.sas](#)
- [m\\_utl\\_print\\_message.sas](#)
- [m\\_utl\\_print\\_mtrace.sas](#)
- [m\\_utl\\_quote\\_elems.sas](#)
- [m\\_utl\\_varlist.sas](#)

## Usage

Example 1: Show help information:

```
%m_val_chk_invalid(?)
```

Example 2: Validate SASHELP.class to verify that all male students are minor:

```
%m_val_chk_invalid(  
  src_tbl = SASHELP.class  
, col_name = Age  
, val_max = 17  
, action = Check  
, print = Y  
, debug = Y  
);
```

Example 3: Validate SASHELP.class to verify that all student have a gender:

```
%m_val_chk_invalid(  
  src_tbl = SASHELP.class  
, col_name = Sex  
, val_list = F M  
, action = Check  
, print = Y  
, debug = Y  
);
```

Example 4: Validate SASHELP.class to ensure that all students are female:

```
%m_val_chk_invalid(  
  src_tbl = SASHELP.class  
, col_name = Sex  
, val_max = F  
, action = Move  
, print = Y  
, debug = Y  
);
```

Example 5: Validate SASHELP.class to abort when not all students are female:

```
%m_val_chk_invalid(  
  src_tbl = SASHELP.class  
, col_name = Sex  
, val_list = F  
, action = Abort  
, print = Y  
, debug = Y  
);
```

Example 6: Validate SASHELP.baseball to ensure all players salaries are listed:

```
%m_val_chk_invalid(  
  src_tbl = SASHELP.baseball  
, col_name = Salary  
, val_min = 0  
, action = Move  
, print = Y  
, debug = Y  
);  
  
proc print data=WORK.baseball_exc label;  
run;
```

Example 7: Validate SASHELP.baseball to check if all players salaries are listed:

```
%m_val_chk_invalid(  
  src_tbl = SASHELP.baseball  
  , exc_tbl = WORK.baseball_invalid  
  , col_name = Salary  
  , val_min = 0  
  , val_max = 10000  
  , action = Check  
  , print = Y  
  , debug = N  
  );  
  
%m_val_chk_invalid(  
  src_tbl = SASHELP.baseball  
  , exc_tbl = WORK.baseball_invalid  
  , col_name = logSalary  
  , val_min = 0  
  , val_max = 10000  
  , action = Check  
  , print = Y  
  , debug = N  
  );  
  
proc print data=WORK.baseball_invalid label;  
run;
```

## **Copyright**

Copyright 2008-2021 Paul Alexander Canals y Trocha.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.