# TEST 1

## Error conditions handled smoothly

In most cases tests should include confirmation that the macro handles invalid or error conditions smoothly, with a clear message to the user, and an appropriate return value to the calling program.

For example, given an invalid data set name, a macro should set the global macro variable CONTINUE to 0 (See [Project Programming Guidelines](http://www.phusewiki.org/wiki/index.php?title=WG5_P02_Programming_Guidelines) for further discussion).

## Checks

1. MISSING required parameter, **return 0**
   1. Missing DS
   2. Missing VAR
   3. Missing DS and VAR
2. repeat a. with INVALID required parameter, **return 0**
3. Invalid optional WHR= clause, **return 0**
   1. invalid variable specified
   2. variable type mismatch in WHR clause, using NUMERIC var
   3. variable type mismatch in WHR clause, using CHAR var

None of these are *failure* conditions, therefore there is **no change to the global CONTINUE macro variable.**

# TEST 2

## Title of this macro-specific test

Brief description and justification of this test.

For example, illustrate the need for this test using an example from clinical trials analysis.

## Checks

1. NUMERIC variable with entirely NON-MISSING values, **return 1**
   1. Variable name is the **prefix** or **suffix** of *other* NUM vars that ARE ENTIRELY missing (., \_, .m, or .z)
   2. Variable name is the **prefix** or **suffix** of *other* NUM vars that HAVE EXACTLY 1 missing val (., \_, .m, or .z)
2. CHAR variable with entirely NON-MISSING values**, return 1**
   1. Variable name is the **prefix** or **suffix** of *other* CHAR vars that ARE ENTIRELY missing ( )
   2. Variable name is the **prefix** or **suffix** of *other* CHAR vars that HAVE EXACTLY 1 missing value ( )
3. NUMERIC variable has missing values**, return 0**
   1. exactly one default missing value (.)
   2. exactly one special missing value (\_)
   3. exactly one special missing value (.m)
   4. multiple default missing values (.)
   5. mix of special missing values (e.g., \_, .m)
4. CHAR variable has missing value, **return 0**
   1. exactly 1 missing value ( )
   2. entirely missing values ( )
5. WHR clause eliminates all missing values, **return 1**
   1. WHR clause eliminates all NUM missing values
   2. WHR clause eliminates all CHAR missing values
6. WHR clause leaves some missing value, **return 0**
   1. WHR clause leaves exactly one missing NUM value (.)
   2. WHR clause leaves exactly one special NUM missing (\_)
   3. WHR clause leaves exactly one special NUM missing (.m)
   4. WHR clause leaves mix of NUM missing values
   5. WHR clause leaves exactly one CHAR missing ( )

# TEST 3

## Confirm expected log messages

In most cases tests will include failure scenarios, which should produce log messages. The following log messages are EXPECTED due to test design (i.e., intentionally testing failure conditions):

WARNING: (TEST\_*macro-name*) User must ensure PhUSE/CSS utilities are in the AUTOCALL path.

WARNING: (TEST\_ASSERT\_VAR\_NONMISSING) User must ensure PhUSE/CSS utilities are in the AUTOCALL path.

ERROR: (ASSERT\_DSET\_EXIST) Result is FAIL. Please specify a data set name.

ERROR: (ASSERT\_VAR\_NONMISSING) Please specify a variable on data set SASHELP.HEART.

ERROR: (ASSERT\_DSET\_EXIST) Result is FAIL. Please specify a data set name.

ERROR: (ASSERT\_VAR\_NONMISSING) Please specify a variable on data set .

ERROR: (ASSERT\_DSET\_EXIST) Result is FAIL. Data set SASHELP.HEARTS is NOT accessible. Try another data set.

ERROR: (ASSERT\_VAR\_EXIST) Result is FAIL. "WEIGHTS" is NOT a variable on data set SASHELP.HEART.

ERROR: (ASSERT\_DSET\_EXIST) Result is FAIL. Data set SASHELP.HEARTS is NOT accessible. Try another data set.

ERROR: (ASSERT\_VAR\_NONMISSING) Data set SASHELP.HEART is not accessible. Abort check for variable CHOLESTEROL.

ERROR: (ASSERT\_VAR\_NONMISSING) Review your where clause carefully. Test the compound clause: where=(missing(cholesterol) and (weights > 150))

ERROR: (ASSERT\_VAR\_NONMISSING) SYSMSG is: ERROR: Variable weights is not on file SASHELP.HEART.

ERROR: (ASSERT\_VAR\_NONMISSING) Data set SASHELP.HEART is not accessible. Abort check for variable CHOLESTEROL.

ERROR: (ASSERT\_VAR\_NONMISSING) Review your where clause carefully. Test the compound clause: where=(missing(cholesterol) and (weight in ('A' 'B')))

ERROR: (ASSERT\_VAR\_NONMISSING) SYSMSG is: ERROR: WHERE clause operator requires compatible variables.

ERROR: (ASSERT\_VAR\_NONMISSING) Data set SASHELP.HEART is not accessible. Abort check for variable CHOLESTEROL.

ERROR: (ASSERT\_VAR\_NONMISSING) Review your where clause carefully. Test the compound clause: where=(missing(cholesterol) and (bp\_status > 5))

ERROR: (ASSERT\_VAR\_NONMISSING) SYSMSG is: ERROR: WHERE clause operator requires compatible variables.

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 1 Missing values for variable "D1\_NUM\_VAL" on data set "TEST\_1MISS" (where=).

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 1 Missing values for variable "U1\_NUM\_VAL" on data set "TEST\_1MISS" (where=).

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 1 Missing values for variable "M1\_NUM\_VAL" on data set "TEST\_1MISS" (where=).

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 12 Missing values for variable "M\_NUM\_VAL" on data set "TEST\_NONMISS" (where=).

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 12 Missing values for variable "S\_NUM\_VAL" on data set "TEST\_NONMISS" (where=).

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 1 Missing values for variable "CHR\_VAL\_D1" on data set "TEST\_1MISS" (where=).

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 12 Missing values for variable "CHR\_VAL\_M" on data set "TEST\_NONMISS" (where=).

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 1 Missing values for variable "NUM\_VAL" on data set "TEST\_MISS\_SUBSET" (where=desc in ("missing\_1.", "non-missing")).

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 1 Missing values for variable "NUM\_VAL" on data set "TEST\_MISS\_SUBSET" (where=desc in ("special\_1\_", "non-missing")).

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 1 Missing values for variable "NUM\_VAL" on data set "TEST\_MISS\_SUBSET" (where=desc in ("special\_1m", "non-missing")).

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 16 Missing values for variable "NUM\_VAL" on data set "TEST\_MISS\_SUBSET" (where=desc contains "missing" or desc contains "special\_").

WARNING: (ASSERT\_VAR\_NONMISSING) Result is FAIL. 1 Missing values for variable "CHR\_VAL" on data set "TEST\_MISS\_SUBSET" (where=desc in ("missing\_1.", "non-missing")).