# TEST 1

## Macro handles error conditions with clear messages to user

The user must supply a valid data set name and key variables that identify unique records.

Establish that the macro responds well to invalid conditions, returning a clear log message and global symbol CONTINUE = 0

## Checks

1. Invalid data set
   1. NULL data set name
   2. Non-existent data set name
   3. Invalid data set name is a partial match of a valid data set name
      1. Invalid data set is a truncation of a valid data set name.  
         E.g. ADVS\_detail is valid, ADVS is invalid
      2. Valid data set name is a truncation of an invalid data set name  
         E.g., ADVS is valid, css\_ADVS is invalid
2. Invalid unique key(s)
   1. NULL keys
   2. Single invalid key variable
   3. Multiple invalid key variables
3. Invalid INCL variable
   1. Single invalid INCL variable
   2. Multiple invalid INCL variables
4. Checks are not case sensitive – used mixed case in definitions

# TEST 2

## Single key variable

Ensure that macro correctly assess a valid and an invalid single unique key.

## Checks

1. Correctly confirm that a valid single unique key does identify unique records (PASS)
2. Correctly detect that single key with multiple related records does not identify unique records (FAIL).
3. Checks are not case sensitive

# TEST 3

## Multiple key variables

Ensure that macro correctly assess valid and invalid sets of multiple unique keys.

## Checks

1. Correctly confirm that a valid set of unique keys does identify unique records (PASS)
2. Correctly detect that a set of keys with multiple related records does not identify unique recs (FAIL).
3. Checks are not case sensitive

# TEST 4

## Handling included variables

Confirm correct diagnostic data set created for various scenarios and values of INCL

## Checks

1. Correct diagnostic data set work.fail\_auk with single non-unique key and additional INCL variables
2. Correct diagnostic data set work.fail\_auk with non-unique set of key variables and additional INCL variables
3. Checks are not case sensitive

# TEST 5

## Handling SQL where clause

Confirm behavior with optional parameter SQLWHR, which allows user to first limit data to specific records prior to testing for uniqueness according to KEYS

Test data set CLASSES contains 2 sets of records based on SOURCE value of "SASHELP" or "FLIPPED". Individually, NAME is a valid key. Together, NAME and SEX (or SOURCE) compose a valid set of unique keys.

Test data set CLASSES\_DUP contains 2 copies of CLASSES based on COPY value of 1 or 2. Individually, NAME and SEX (or SOURCE) compose a valid set of unique keys. Together, NAME, SEX (or SOURCE) and COPY compose a valid set of unique keys.

## Checks

1. Correctly identify valid single key after first subsetting data
2. Correct diagnostic data set work.fail\_auk with single non-unique key on subset of records
3. Correctly identify valid set of keys after first subsetting data
4. Correct diagnostic data set work.fail\_auk with non-unique set of key variables on subset of records
5. Checks are not case sensitive

# TEST 6

## 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 (testing failure conditions):

WARNING: (TEST\_ASSERT\_UNIQUE\_KEYS) 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\_DSET\_EXIST) Result is FAIL. Data set WORK.DNE is NOT accessible. Try another data set.

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

ERROR: (ASSERT\_DSET\_EXIST) Result is FAIL. Data set WORK.CSS\_CLASSES is NOT accessible. Try another data set.

ERROR: (ASSERT\_UNIQUE\_KEYS) Result is FAIL. Please specify a variable name.

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

ERROR: (ASSERT\_VAR\_EXIST) Result is FAIL. "NAM" is NOT a variable on data set WORK.CLASSES.

ERROR: (ASSERT\_VAR\_EXIST) Result is FAIL. "AG" is NOT a variable on data set WORK.CLASSES.

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

ERROR: (ASSERT\_UNIQUE\_KEYS) Unexpected duplicates in CLASSES with unique keys NAME (SQLOBS = 19). See WORK.FAIL\_AUK.

ERROR: (ASSERT\_UNIQUE\_KEYS) Unexpected duplicates in WORK.CLASSES with unique keys NAME AGE (SQLOBS = 19). See WORK.FAIL\_AUK.

ERROR: (ASSERT\_UNIQUE\_KEYS) Unexpected duplicates in CLASSES with unique keys NAME (SQLOBS = 38). See WORK.FAIL\_AUK.

ERROR: (ASSERT\_UNIQUE\_KEYS) Unexpected duplicates in WORK.CLASSES with unique keys NAME AGE (SQLOBS = 38). See WORK.FAIL\_AUK.

ERROR: (ASSERT\_UNIQUE\_KEYS) Unexpected duplicates in WORK.CLASSES with unique keys NAME where height < 64 (SQLOBS = 11). See WORK.FAIL\_AUK.

ERROR: (ASSERT\_UNIQUE\_KEYS) Unexpected duplicates in CLASSES\_DUP with unique keys NAME SEX where age < 13 (SQLOBS = 14). See WORK.FAIL\_AUK.

ERROR: Errors printed on page 1.

Errors printed on page 1.