# m_utl_delete_index.sas File Reference

## Utilities

Utility macro to delete an index from a given table or dataset

---

### Description

The macro deletes an index from a given table or dataset by using the PROC DATASETS procedure. This macro can be included inline into an existing PROC DATASETS step by setting the INLINE_FLG parameter to Y. Also there is a build in check on INDEX_NAME or INDEX_COLS to verify that the given index does exist. If this is not the case, the macro returns silently.

### Autors

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

### Date

2020-02-02 00:00:00

### Version

20.1.01

### Link

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

### Parameters

| | | |
|---|---|---|
| Input | help | Parameter, if set ( or ?) to print the Help information in the log. In all other cases this parameter should be left out from the macro call. |
| Input | table | Full LIBNAME.TABLENAME name of the or SAS dataset. The default value is _NONE. |
| Input | index_name | Name of the index. If value is omitted the macro will use the INDEX_COLS to search for an index. The default value is: _NONE_. |
| Input | index_cols | Optional. List of column names separated by a blank character to search for an exising index. The default value is: _NONE_. |
| Input | inline_flg | Boolean [Y\|N] Parameter to declare if the macro is to be run inline inside a PROC DATASETS step. The Default value is: N. |
| Input | debug | Boolean [Y\|N] parameter to provide verbose mode information. The default value is: N. |

### Returns

- Deleted index from table

### Calls

- m_utl_chk_table_index.sas
- m_utl_print_message.sas

**Usage**

Example 1: Show help information:

```
%m_utl_delete_index(?)
```

For the next examples create a table with several SIMPLE and COMPOSITE indexes:

```
data WORK.class;
   set SASHELP.class;
run;

proc datasets library=WORK nolist;
   modify class;
   index create Name;
   index create comp_1 = (Name Sex);
   index create comp_2 = (Name Age);
   index create comp_3 = (Sex Age Name);
quit;

proc print data=SASHELP.vindex;
   where libname='WORK' and memname='CLASS';
run;
```

Example 2: Delete simple index on Name (Result: index Name deleted):

```
proc datasets lib=WORK nolist;
   %m_utl_delete_index(
      table      = WORK.class
    , index_name = Name
    , inline_flg = Y
    , debug      = Y
      )
quit;

proc print data=SASHELP.vindex;
   where libname='WORK' and memname='CLASS';
run;
```

Example 3: Delete simple index on Name (Result: index not deleted):

```
%m_utl_delete_index(
   table      = WORK.class
 , index_cols = Name
 , inline_flg = N
 , debug      = Y
   )

proc print data=SASHELP.vindex;
   where libname='WORK' and memname='CLASS';
run;
```

Example 4: Delete composite index on Name Age (Result: index COMP_2 deleted):

```
%m_utl_delete_index(
   table      = WORK.class
 , index_cols = Name Age
 , inline_flg = N
 , debug      = Y
   );

proc print data=SASHELP.vindex;
   where libname='WORK' and memname='CLASS';
run;
```

Example 5: Delete composite index on Name Sex (Result: index COMP_1 deleted):

```
%m_utl_delete_index(
   table      = WORK.class
 , index_name = comp_1
 , inline_flg = N
 , debug      = Y
   );

proc print data=SASHELP.vindex;
   where libname='WORK' and memname='CLASS';
run;
```

Example 6: Delete composite index on Sex Age (Result: index not deleted):

```
%m_utl_delete_index(
   table      = WORK.class
 , index_cols = Sex Age
 , inline_flg = N
 , debug      = Y
   );

proc print data=SASHELP.vindex;
   where libname='WORK' and memname='CLASS';
run;
```

**Copyright**