m_utl_delta_extract.sas File Reference

Utilities

Utility macro to enable incremental loading of a target table

---

Description

The macro can be used to extract data from a source SAS dataset or database table incremetially by processing delta loads. The macro compares both the maximum keys and number of records on the source and target table to detect changes. Depending on the comparison result, it will be decided wether to perform a full or a delta load from the source table into the target table.

Note

In case of encrypted SAS datasets, the ENCRYPTKEY= parameter must be provided as part of the CREDS credentials string.

Autors

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

Date

2020-02-02 00:00:00

Version

20.1.02

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 source or SAS dataset. The default value is: _NONE_.
Input	dataset	Alias of the TABLE= parameter.
Input	data	Alias of the TABLE= parameter.
Input	creds	Optional. Specifies the ENCRYPTKEY= parameter value if there is an encrypted table involved.
Input	key	Specifies the source and target indexed column. The default value is: _NONE_.
Input	key_sort	Boolean [Y|N] parameter to specify wether the data is to be sorted by the KEY column before appending to the target table. The default value is: N.
Input	tgt_table	Full LIBNAME.TABLENAME name of the target table or SAS dataset. If the table does not exist it will be created with an initial full table load.
Input	include	Optional. Blank separated list of columns to limit the result column list to only those columns listed in the include list parameter.
Input	exclude	Optional. Blank separated list of columns to limit the result column list by excluding those columns listed in the exclude list parameter.
Output	trecs_mvar	Numeric parameter containing the total number of records in the target table after table loading. The default value for TRECS_MVAR is: _target_recs_.
Output	drecs_mvar	Numeric parameter containing the number of records loaded into the target table by a delta load. The default value for DRECS_MVAR is: _delta_recs_.
Output	full_mvar	Boolean [0|1] parameter to specify wether a full or delta load was used to update the target table. The default value for FULL_MVAR is: _full_extract_.
Input	debug	Boolean [Y|N] parameter to provide verbose mode information. The default value is: N.

Returns

- Incremental load into a given target table.

Calls

- m_utl_create_index.sas
- m_utl_delete_index.sas
- m_utl_get_col_type.sas
- m_utl_get_max_value.sas
- m_utl_nlobs.sas
- m_utl_print_message.sas
- m_utl_varlist.sas

Usage

Example 1: Show help information:

```
%m_utl_delta_extract(?)
```

Example 2: Step 1 - Initialize global macro variable output parameters:

```
* Initialize macro vars;
%let _target_recs_=;
%let _delta_recs_=;
%let _full_extract_=;
```

Example 2: Step 2 - Perform an initial full load extraction of SASHELP.class (18 records):

```
%m_utl_delta_extract(
   table      = SASHELP.class(obs=18)
 , key        = Name
 , tgt_table  = WORK.class
 , include    = Name Age Sex Weight
 , trecs_mvar = _target_recs_
 , drecs_mvar = _delta_recs_
 , full_mvar  = _full_extract_
 , debug      = Y
   );

data WORK.result;
   TargetRecs = &_target_recs_.;
   DeltaLoad  = &_delta_recs_.;
   FullLoad   = &_full_extract_.;
run;

proc print data=WORK.class;
run;

proc print data=WORK.result noobs;
run;
```

Example 2: Step 3 - Perform an update delta load extraction of SASHELP.class (1 record):

```
%m_utl_delta_extract(
   table      = SASHELP.class
 , key        = Name
 , key_sort   = Y
 , tgt_table  = WORK.class
 , include    = Name Age Sex Weight
 , trecs_mvar = _target_recs_
 , drecs_mvar = _delta_recs_
 , full_mvar  = _full_extract_
 , debug      = Y
   );

data WORK.result;
   TargetRecs = &_target_recs_.;
   DeltaLoad  = &_delta_recs_.;
   FullLoad   = &_full_extract_.;
run;

proc print data=WORK.class;
run;

proc print data=WORK.result noobs;
run;
```

Example 2: Step 4 - Perform an update delta load extraction of SASHELP.class (0 records):

```
%m_utl_delta_extract(
   table      = SASHELP.class
 , key        = Name
 , key_sort   = Y
 , tgt_table  = WORK.class
 , include    = Name Age Sex Weight
 , trecs_mvar = _target_recs_
 , drecs_mvar = _delta_recs_
 , full_mvar  = _full_extract_
 , debug      = Y
   );

data WORK.result;
   TargetRecs = &_target_recs_.;
   DeltaLoad  = &_delta_recs_.;
   FullLoad   = &_full_extract_.;
run;

proc print data=WORK.class;
run;

proc print data=WORK.result noobs;
run;
```