SAS Analytics for IoT Enhanced Data Load Macro

Overview

The SAS Analytics for IoT Enhanced Data Load Macro is an alternate tool that can be used to load the data mart for the SAS Analytics for IoT application. It provides additional functionality and can be used to load the data mart when the traditional data load macro does not meet your needs.

Additional Functionality

The following section describes the additional functionality supported by the enhanced data load macro.

Data Sources

The enhanced data load macro supports any type of data source supported by SAS Viya including:

- SAS datasets
- Database tables
- SAS Event Stream Processing data
- CSV Files

Datetime Value Precision

The precision of datetime values is maintained regardless of the granularity.

Macro Restart Functionality

Functionality is included to restart the enhanced data load macro from any step if it does not complete successfully. This means you do not have to restart the macro from the beginning if a data load fails. This can be especially useful on large data loads.

Filter Input Data Source

The enhanced data load macro supports filtering the input data at the data source level. This can be useful for large data sources where the entire source is not needed for analysis.

Executed Entirely in CAS

To take advantage of the power of Cloud Analytic Services (CAS), the entire enhanced data load macro executes using only CAS. This improves performance in most cases.

Save Data Mart Tables

Because the SAS Analytics for IoT data mart is stored using CAS, it resides entirely in memory. Therefore, if the memory becomes unloaded for any reason you must reload the data mart. The enhanced data load macro allows you to save the data mart tables to a permanent location. The tables can then be easily reloaded and the Analytics for IoT data mart will be restored.

Code Written in CASL

To leverage the performance and functionality of CAS, the enhanced data load macro is written almost entirely in the Cloud Analytic Services Language (CASL).

Process Description

The enhanced data load macro process is very similar to the production data load process. A parameter file continues to be used to control the process and the SAS Analytics for IoT data mart is loaded upon successful completion of the process.

Macro Steps

The enhanced data load macro is divided into individual steps each acting independently. This allows you to restart the macro from any of the steps.

The following table lists the steps and provide a description of each for the enhanced data load macro:

Step	Description
Step 1	Read the parameter file and load input tables to a temporary location
Step 2	Change the format of variable DATETIME in the measures table to improve precision
Step 3	Create PAM_ASSET_DIM table in data mart

Step 4	Create PAM_TAG_DIM table in data mart
Step 5	Create PAM_ASSET_LOC_MEASURE_FACT table in data mart
Step 6	Create PAM_FACT_TAG_MEASURES_STATS table in data mart
Step 7	Create PAM_EVENT_DIM table in data mart if required
Step 8	Create PAM_ASSET_LOC_EVENT_FACT table in data mart if required
Step 9	Create PAM_FACT_EVENT_STATS table in data mart if required
Step 10	Create hierarchies if required
Step 11	Create PREDEFINED_DATE table in data mart
Step 12	Create copy of data mart tables and save to backup_caslib if specified
Step 13	Clean up process

Parameter File

The enhanced data load macro uses a parameter file to control the process. The following table lists the available parameters, whether they are required, and provides a brief description of each parameter:

Parameter	Required?	Description
datetime_column	Yes	Name of column containing datetime values in measures file or table
device_id_column	Yes	Name of column containing DEVICE_ID values
event_aggreg_missing_method	Only if event measures loaded	Method used to aggregate missing values in event measures file or table. Default value is AVERAGE.
event_id_column	Only if event measures loaded	Name of column containing EVENT_ID values
measure_table	Yes	Path and name of measures file or table

event_table	No	Path and name of events file or table
event_measures_table	No	Path and name of event measures file or table
event_hierarchy_table	No	Path and name of event hierarchy file or table
device_table	No	Path and name of devices file or table
device_hierarchy_table	No	Path and name of devices hierarchy file or table
sensor_table	No	Path and name of sensors file or table
sensor_hierarchy_table	No	Path and name of sensors hierarchy file or table
measure_filter	No	SAS expression used to filter measures prior to being loaded
value_column	Yes	Name of column containing measurement values in measures file or table
missing_char_value	No	Character to use to represent missing values in the data mart. Default value is ?.
label_char_seperator	No	Character to replace the spaces in column names.
sensor_aggreg_missing_method	Yes	Method used to aggregate missing values in measures file or table. Default value is AVERAGE.
sensor_id_column	Yes	Name of column containing SENSOR_ID values
sensor_interpol_missing_method	Yes	Method used to interpolate missing values in measures file or table. Default value is NONE.
node_column	No	Name of column containing node values in hierarchy files or tables. Default value is NODE.
parent_column	No	Name of column containing parent node values in hierarchy files or tables. Default value is PARENT.
level_column	No	Name of column containing level number values in hierarchy files or tables. Default value is LEVEL_NUM.

Executing the Macro

There are several steps you need to perform to execute the enhanced data load macro:

- Load the macro source code
- Create a program file (.sas) that includes the following:
  - Statement to create a CAS session
  - Statement to assign a caslib for the imput data source
  - Statement to assign a caslib for the output data if used
  - Statement to call the macro with appropriate parameters
- Execute the program file you created

```
%aiot_main(
    parameter_file=/tmp/AIot/params/parameter_file.txt,
    print_parameters=y,
    first_step=1,
    input_caslib=mySrclib,
    backup_caslib=,
    replace_backup=,
    load_pregen=,
    debug=n );
```

Load Macro Source Code

You must load the macro source code before you can execute the macro. You can do this by opening file AIoT_CASL.sas in SAS Studio V and submitting it.

Create Program File

Create a new program file (.sas) in SAS Studio V or open one of the example files. The program file to execute the enhanced data load macro must include the following statements:

- Create a CAS session
- Assign a caslib for the imput data source
- Assign a caslib for the output data if used
- Call the macro with appropriate input parameters

CAS Session

You must start a CAS session for the macro to execute. The example code includes the following statement that you can use:

```
cas casl sessopts=(timeout=99 locale="en_US");
```

CASLIB References

At a minimum, you will need a caslib for your input data source (input_caslib). If you are saving a copy of the data mart, you will also need a caslib for the output (backup_caslib).

A CAS session can have only one active caslib. A caslib becomes active when it is the first caslib assigned and no other caslibs are assigned, or they are assigned with the notactive option.

Ensure you assign the caslib referenced by input_caslib as the active caslib. If you are using the backup_caslib option, ensure you use the notactive option. The following example assigns the input_caslib as the active caslib and the backup_caslib as the notactive caslib:

```
caslib mySrclib
  datasource=(srctype='dnfs')
  path="/tmp/AIot/params/parameter_file.txt"

1-- input_caslib is active

caslib myHdplib
  datasource=(
    srctype="hadoop"
    server="myserver"
    hadoopjarpath="/home/dbclients/hadoop/cloudera/clusters/DIP/lib"
    hadoopconfigdir="/home/dbclients/hadoop/cloudera/clusters/DIP/conf"
    authDomain=HadoopAuth
    schema=aiot)
  notactive;

2 -- backup_caslib is notactive
```

Macro Call

The macro call for the enhanced data load macro is as follows:

```
%aiot_main(
  parameter_file= ,
  print_parameters= ,
  first_step= ,
  input_caslib= ,
  backup_caslib= ,
  replace_backup=  ,
  load_pregen= ,
  debug=  );
```

Input Parameters

The following table list the input parameters of the enhanced data load macro and provides a description and valid values (if applicable) for each:

Parameter	Description
parameter_file	Fully qualified name of parameter file
print_parameters	Whether or not to print the parameter file parameters and their values in the SAS Log. Valid values are Y and N.
first_step	Process step where macro is to begin. Use 1 for normal operations. Valid values are 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, and 13.
input_caslib	The caslib library reference of the input data. If load_pregen is set to Y, the input_caslib parameter should reference the location of a set of previously saved tables.
backup_caslib	The caslib library reference of the location to where a copy of the data mart tables will be saved.
replace_backup	Whether or not the copy of the data mart tables being saved should be replaced. Valid values are Y and N. Default value is N.
load_pregen	Specifies to load a previously saved set of data mart tables should be used as input. Valid values are Y and N. If load_pregen is set to Y, the only other required parameter is input_caslib. All other parameters are ignored.
debug	Generates additional information in the SAS Log. Valid values are Y and N.

Examples

The following examples show different ways to use the enhanced data load macro. All these examples are included in examples directory on this site.

Example 1 – Loading a set of csv files

```
/*************************************************************************/
/* CAS statement to start CAS session.                                   */
/*************************************************************************/

cas casl sessopts=(timeout=99 locale="en_US");


/*************************************************************************/
/* CASLIB statement for input_caslib.                                    */
/*************************************************************************/

caslib mySrclib
  datasource=(srctype='dnfs')
  path="/tmp/AIot/params/parameter_file.txt"


/*************************************************************************/
/* Macro call.                                                           */
/*************************************************************************/

%aiot_main(
  parameter_file=/tmp/AIot/params/parameter_file.txt,
  print_parameters=y,
  first_step=1,
  input_caslib=mySrclib,
  backup_caslib=,
  replace_backup=,
  load_pregen=,
  debug=n );
```

Example 2 – Loading a set of csv files and saving output to Hadoop

```
/*************************************************************************/
/* CAS statement to start CAS session.                                   */
/*************************************************************************/

cas casl sessopts=(timeout=99 locale="en_US");
```

```
/*************************************************************************/
/* CASLIB statement for input_caslib.                                     */
/*************************************************************************/

caslib mySrclib
  datasource=(srctype='dnfs')
  path="/tmp/AIot/params/parameter_file.txt"


/*************************************************************************/
/* CASLIB statement for backup_caslib.                                    */
/*************************************************************************/

caslib myHdplib
  datasource=(
    srctype="hadoop"
    server="myserver"
    hadoopjarpath="/home/dbclients/hadoop/cloudera/clusters/DIP/lib"
    hadoopconfigdir="/home/dbclients/hadoop/cloudera/clusters/DIP/conf"
    authDomain=HadoopAuth
    schema=aiot)
  notactive;


/*************************************************************************/
/* Macro call.                                                           */
/*************************************************************************/

%aiot_main(
  parameter_file=/tmp/AIot/params/parameter_file.txt,
  print_parameters=y,
  first_step=1,
  input_caslib=mySrclib,
  backup_caslib= myHdplib,
  replace_backup=Y,
  load_pregen=,
  debug=n );
```

Example 3 – Loading a previously saved set of tables

```
/*************************************************************************/
/* CAS statement to start CAS session.                                   */
/*************************************************************************/

cas casl sessopts=(timeout=99 locale="en_US");
```

```
/*************************************************************************/
/* CASLIB statement for input_caslib.                                     */
/*************************************************************************/

caslib mySrclib
  datasource=(
    srctype="hadoop"
    server="myserver"
    hadoopjarpath="/home/dbclients/hadoop/cloudera/clusters/DIP/lib"
    hadoopconfigdir="/home/dbclients/hadoop/cloudera/clusters/DIP/conf"
    authDomain=HadoopAuth
    schema=aiot);


/*************************************************************************/
/* Macro call.                                                           */
/*************************************************************************/

%aiot_main(
  parameter_file=,
  print_parameters=,
  first_step=,
  input_caslib=mySrclib,
  backup_caslib=,
  replace_backup=,
  load_pregen=Y,
  debug=n );
```

Example 4 – Copying a previously saved set of tables to another data source.

```
/*************************************************************************/
/* CAS statement to start CAS session.                                   */
/*************************************************************************/

cas casl sessopts=(timeout=99 locale="en_US");


/*************************************************************************/
/* CASLIB statement for input_caslib.                                     */
/*************************************************************************/

caslib mySrclib
  datasource=(
```

```
    srctype="hadoop"
    server="myserver"
    hadoopjarpath="/home/dbclients/hadoop/cloudera/clusters/DIP/lib"
    hadoopconfigdir="/home/dbclients/hadoop/cloudera/clusters/DIP/conf"
    authDomain=HadoopAuth
    schema=aiot);


/*************************************************************************/
/* CASLIB statement for backup_caslib.                                   */
/*************************************************************************/

caslib myOralib
  datasource=(
    srctype="oracle"
    path=oraclev11
    authDomain=OracleAuth)
  notactive;


/*************************************************************************/
/* Macro call.                                                           */
/*************************************************************************/

%aiot_main(
  parameter_file=,
  print_parameters=,
  first_step=,
  input_caslib=mySrclib,
  backup_caslib= myOralib,
  replace_backup=Y,
  load_pregen=Y,
  debug=n );
```

Example SAS Log

```
01OCT2019 14:39:09 NOTE: Starting program from step 1
Parameter list:
DATETIME_COLUMN DATETIME
DEVICE_ID_COLUMN DEVICE_ID
EVENT_AGGREG_MISSING_METHOD AVERAGE
EVENT_ID_COLUMN
```

```
MEASURE_TABLE measures.csv
EVENT_TABLE
EVENT_MEASURE_TABLE
EVENT_HIERARCHY_TABLE
DEVICE_TABLE
EVICE_HIERARCHY_TABLE
SENSOR_TABLE sensors.csv
SENSOR_HIERARCHY_TABLE sensors_hierarchy.csv
MEASURE_FILTER
VALUE_COLUMN VALUE
MISSING_CHAR_VAL ?
LABEL_CHAR_SEPARATOR
SENSOR_AGGREG_MISSING_METHOD AVERAGE
SENSOR_ID_COLUMN SENSOR_ID
SENSOR_INTERPOL_MISSING_METHOD NONE
NODE_COLUMN NODE
PARENT_COLUMN PARENT
LEVEL_COLUMN LEVEL_NUM
```

```
01OCT2019 14:39:09 Step 1 - NOTE: Loading input data from ...
01OCT2019 14:39:09 Step 1 - NOTE: Table MEASURES found.
01OCT2019 14:39:09 Step 1 - NOTE: Table MEASURES loaded successfully.
01OCT2019 14:39:09 Step 1 - NOTE: Table SENSORS found.
01OCT2019 14:39:09 Step 1 - NOTE: Table SENSORS loaded successfully.
01OCT2019 14:39:10 Step 1 - NOTE: Table SENSORS_HIERARCHY found.
01OCT2019 14:39:10 Step 1 - NOTE: Table SENSORS_HIERARCHY loaded
successfully.
01OCT2019 14:39:10 Step 2 - NOTE: Changing datetime column type for column
DATETIME in table MEASURES...
01OCT2019 14:39:10 Step 2 - NOTE: Existence successfully verified for the
MEASURES table.
01OCT2019 14:39:10 Step 2 - NOTE: Datetime column type for column DATETIME in
table MEASURES changed successfully.
01OCT2019 14:39:10 Step 3 - NOTE: Creating the PAM_ASSET_DIM table...
01OCT2019 14:39:10 Step 3 - NOTE: Table PAM_ASSET_DIM created successfully.
01OCT2019 14:39:10 Step 4 - NOTE: Creating the PAM_TAG_DIM table...
01OCT2019 14:39:10 Step 4 - NOTE: Existence successfully verified for the
MEASURES table.
01OCT2019 14:39:10 Step 4 - NOTE: Table PAM_TAG_DIM created successfully.
01OCT2019 14:39:10 Step 5 - NOTE: Creating the PAM_ASSET_LOC_MEASURE_FACT
table...
01OCT2019 14:39:10 Step 5 - NOTE: Existence successfully verified for the
MEASURES table.
01OCT2019 14:39:11 Step 5 - NOTE: Table PAM_ASSET_LOC_MEASURE_FACT created
successfully.
```

```
01OCT2019 14:39:11 Step 6 - NOTE: Creating the PAM_FACT_TAG_MEASURES_STATS
table...
01OCT2019 14:39:11 Step 6 - NOTE: Existence successfully verified for the
PAM_ASSET_LOC_MEASURE_FACT table.
01OCT2019 14:39:11 Step 6 - NOTE: Table PAM_FACT_TAG_MEASURES_STATS created
successfully.
01OCT2019 10:39:11 Step 7 - NOTE: Skipping PAM_EVENT_DIM creation...
01OCT2019 10:39:11 Step 8 - NOTE: Skipping PAM_ASSET_LOC_EVENT_FACT
creation...
01OCT2019 10:39:11 Step 9 - NOTE: Skipping PAM_FACT_EVENT_STATS creation...
01OCT2019 10:39:11 Step 10 - NOTE: Skipping hierarchies for DEVICE..
01OCT2019 10:39:11 Step 10 - NOTE: Skipping hierarchies for EVENT..
01OCT2019 10:39:11 Step 10 - NOTE: Processing hierarchies for SENSOR..
01OCT2019 14:39:11 Step 10 - NOTE: Existence successfully verified for the
SENSORS_HIERARCHY table.
01OCT2019 14:39:11 Step 10 - NOTE: Structure successfully verified for the
SENSORS_HIERARCHY hierarchy table.
01OCT2019 10:39:11 Step 10 - NOTE: Column LEVEL_NUM found in table
SENSORS_HIERARCHY....
01OCT2019 14:39:12 Step 10 - NOTE: PAM_TAG_DIM renamed successfully.
01OCT2019 14:39:12 Step 10 - NOTE: Hierarchies processed successfully.
01OCT2019 14:39:12 Step 10 - NOTE: PAM_TAG_DIM updated successfully.
01OCT2019 14:39:12 Step 11 - NOTE: Creating the PREDEFINED_DATE table...
01OCT2019 14:39:12 Step 11 - NOTE: Table PREDEFINED_DATE created
successfully.
01OCT2019 10:39:12 Step 12 - NOTE: Output caslib not specified. Tables will
not be saved.
01OCT2019 14:39:12 Step 13 - NOTE: Cleaning up...
Start time : 01OCT2019 14:39:09 End time : 01OCT2019 14:39:12
```