

# m\_utl\_xpt2vw.sas File Reference

## Utilities

Utility macro to import a XPT transport file to a SAS view

---

### Description

This program converts a SAS transport (XPT) file into a SAS view. The macro is able to handle V5 transport files written by the SAS XPORT engine. It should also be able to handle V8 or V9 extended transport files created by companion vw2xpt macro.

### Autors

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

### Date

2020-09-07 00:00:00

### Version

20.1.09

### 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	in_file	Specifies the name and location of the transport file with XPT extension. The default value for IN_FILE is: _NONE_.
Input	out_lib	Specifies the full name of the output library reference name. The default value is: _NONE_.
Input	memlist	Specifies the SAS dataset member in the IN_FILE transport file containing the SAS view creation code structure information. The default value for MEMLIST is: VW_DESCRIBE.
Input	replace	Boolean [Y N] parameter to specify if the output view is to be replaced in case it already exists in the OUT_LIB library. The default value is: Y.
Input	debug	Boolean [Y N] parameter to provide verbose mode information. The default value is: N.

## Returns

- Output SAS view based on code from member VW\_DESCRIBE.

## Calls

- [m\\_utl\\_print\\_message.sas](#)
- [m\\_utl\\_print\\_mtrace.sas](#)

## Usage

Example 1: Show help information:

```
%m_utl_xpt2vw(?)
```

Example 2: Step 1 - Create a SAS DSV view description table VW\_DESCRIBE:

```
data WORK.vw_describe;
  source = 'WORK.dsv_class';
  libref = 'WORK';
  view   = 'DSV_CLASS';
  type   = 'SASDSV';
  code   = 'data WORK.dsv_class / view=WORK.dsv_class; set SASHELP.class; run;';
run;
```

Example 2: Step 2 - Export the description table to a XPT transport file:

```
%loc2xpt(
  libref      = WORK
, memlist     = vw_describe
, filespec    = %unquote(%bquote('%sysfunc(getoption(WORK))/class.xpt'))
, format      = AUTO
);
```

Example 2: Step 3 - Import the class.xpt transport file into a SAS view:

```
%m_utl_xpt2vw(
  in_file = %str(%sysfunc(getoption(WORK))/class.xpt)
, out_lib = WORK
, replace = Y
, debug   = Y
);

proc contents data=WORK.dsv_class;
run;
```

Example 3: Step 1 - Create a SAS SQL view description table VW\_DESCRIBE:

```
data WORK.vw_describe;
  source = 'WORK.sql_class';
  libref = 'WORK';
  view   = 'SQL_CLASS';
  type   = 'SASESQL';
  code   = 'proc sql; create view WORK.sql_class as select * from SASHELP.class; quit;';
run;
```

Example 2: Step 2 - Export the description table to a XPT transport file:

```
%loc2xpt(
  libref      = WORK
, memlist     = vw_describe
, filespec    = %unquote(%bquote('%sysfunc(getoption(WORK))/class.xpt'))
, format      = AUTO
);
```

Example 2: Step 3 - Import the class.xpt transport file into a SAS view:

```
%m_utl_xpt2vw(
  in_file = %str(%sysfunc(getoption(WORK))/class.xpt)
, out_lib = WORK
, replace = Y
, debug   = Y
);

proc contents data=WORK.sql_class;
run;
```

## **Copyright**

Copyright 2008-2020 Paul Alexander Canals y Trocha.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.