

PROJECT

SCRIPT FOR CONCATENATING AND PROCESSING SAS VARIABLE WITH ONE TO
MANY VALUES via SAS

Version 1.0

July 12, 2025

Version History

Date	Version #	Description	Author
07/12/2025	1.0a	Final Version	Vincent Taylor

SCRIPT FOR CONCATENATING AND PROCESSING SAS VARIABLE WITH ONE TO MANY VALUES via SAS

The below macro **ConcatenateMultiValuePrompt** demonstrates the process for handling SAS macro variable that contains from 1 to many values. The values for each macro SAS variable are populated using the 'Filter' definition feature via Query Builder.

```
%global SubGroup_Number ClassId PlanId LineOfBusiness;

%macro ConcatenateMultiValuePrompt (prompt= , dlm=%str( ), AllStr= );
%local i return;

% if "&&&prompt" = "_ALLVALUES_" %then
    % do;
        %let return=&AllStr;
    %end;
%else
    %do;
        %if &&&prompt._Count ge 2 %then
            %do;
                %let i=1;
                %let return=&&&prompt&i;

                %do i = 2 % to &&&prompt._Count;
                    %let return=&return&dlm&&&prompt&i;
                %end;
            %end;
        %else
            %do;
                %let return=&&&prompt;
            %end;
        %end;

    &return

%mend ConcatenateMultiValuePrompt;

/* invoke macro */

%let SubGroupNum List =

%ConcatenateMultiValuePrompt(prompt=SubGroup_Number, dlm=%str(,) ,AllStr=All
SubGroup Number Values Selected);

%let ClassId List = %ConcatenateMultiValuePrompt(prompt=ClassId,dlm=%str(,) ,AllStr=All
Plan Id Values Selected);

%let PlanId List = %ConcatenateMultiValuePrompt(prompt=PlanId,dlm=%str(,) ,AllStr=All Plan
Id Values Selected);
```

```
%let LobCode_List = %ConcatenateMultiValuePrompt (prompt=LineOfBusiness, dlm=%str (,)
,AllStr=All Line of Business Values Selected);
```

Note: You can also define macro SAS variables such as Group_Number and Enroll_Date. The Enroll_Date range would then be redefined as Enroll_Date_Min and Enroll_Date_Max indicating the date range period.

If there's a Group Name associated with Group_Number the below macro can be used to capture the Group_Name

```
%global Group_Name

%macro getgroupname;

    proc sql noprint;
        select Group_Name

            into :group_name

        from test.group_file    a
        where a.group_number in ("&group_number")
        ;
    quit;
%mend getgroupname;

%put &group_name;
```

The following macro demonstrates the process for handling and alerting users when no records are selected.

```
%global dsn exist numobs norecords Group_nme;

%macro datfnd;
    %let dsn = dataset which will be used in the Proc Report step;
    %let exist = no;
    %let numobs = 0;
    %if %sysfunc(exist(&dsn)) %then exist = Yes;
    %if &exist = Yes %then
        %do;
            %let dsnId = %sysfunc(open(&dsn));
            %let dsobs = %sysfunc(attrn(&dsnId.,nobs));
            %let rc = %sysfunc(close(&dsnId.));
            %let numobs = &dsobs.;
        %end;

    %if &numobs = 0 %then
        %do;
            %let NoRecord = &str(**** No Records Were Selected ****);
            %let Group_Nme = &Group_Number;
        %end;
```

```
%else %let NoRecord = %str();
```

```
%put exist: &exist;
```

```
%put Number of OBS: &Numobs &NoRecord;
```

```
%if &Numobs = 0 %then
```

```
  %do;
```

```
    /* Set SAS macro variables used in report header to blank */
```

```
    Proc sql;
```

```
      Insert into dataset which will be used in the Proc Report step (Group_Name,  
Group_Number)
```

```
      Values ( ' ', ' ' );
```

```
    Quit;
```

```
  %end;
```

```
%mend datfnd;
```