

Enhanced OpenCDISC Validator Report for Quick Quality Review

Ajay Gupta, PPD, Morrisville, NC

ABSTRACT

OpenCDISC validator provides great compliance checks against CDISC outputs like SDTM, ADaM, SEND and Define.xml. This validation tool will provide a report in Excel or CSV format which contains errors, warnings, and notice, information. At the initial stage of clinical programming when the data is not very clean, this report can sometimes be very large and tedious to review. Also, if there is data or code list issues in the report then the user needs to check the physical SAS® data sets or SDTM controlled terminology separately which can be very time consuming. In order to expedite quality review time, this paper will introduce an enhanced version of the OpenCDISC validator report. This enhanced report will have SDTM data (only row with issue) and SDTM terminology added as separate worksheets in the original report. Later, hyperlinks between each message in the report and the related SDTM data or SDTM code list worksheets are added using the excel formulas and Visual Basic for Application (VBA). These hyperlinks will further provide point to click options to check the data and code list related issues immediately in the enhanced report which will save significant time with minimal coding. This enhanced report can be further developed to cover ADaM, and SEND databases.

INTRODUCTION

In 2004 the Clinical Data Interchange Standards Consortium (CDISC) recommended the use of the Study Data Tabulation Model (SDTM) standard for submitting clinical data to a regulatory agency. Since that time the pharmaceutical and biotechnology industries have worked tirelessly to implement this standard for the submission of clinical data and its related metadata in order to facilitate the review process of determining the safety and efficacy of a drug. The OpenCDISC community was established to build a framework for the implementation of the CDISC Standard. In fact, this community of professionals created the OpenCDISC Validator tool that performs numerous checks on clinical data to ensure compliance with the standard. This tool validates both collected data as well as its respective metadata.

OpenCDISC is an excellent validation tool. However, using the application poses a real challenge with respect to understanding its concise error and warning messages. Some of the messages do not provide a lot of insight resolving data issues. It's nice to know that a particular error occurred numerous times; however, it is more important to understand the error, where and why it happened. Is the problem systemic? For example, in the OpenCDISC report, the Subject Visits (SV) domain seems to have lot of issues, but only because it contains non-randomized subjects who don't belong there. Least of all, a lot of time is spent trying to decipher messages some of which seem extraneous. In order to use OpenCDISC efficiently, it is necessary to realize the multi-disciplined nature of the validation process, which goes beyond the application, specifically: CDISC, SAS, and clinical data. Also, the user should understand how OpenCDISC functions with respect to validation checks and data/metadata issues. The report only points to the observation number of the domain that resulted in the issue. This leads to increased efforts required by the user to investigate the issue as the user needs to explore the data outside the report using secondary tools.

Admittedly, CDISC requirements for standardization are extensive, and always evolving. The validation process involves various types of checks to ensure compliance, including the metadata describing the clinical data. In fact, the clinical data, stored as SAS transport (XPT) data sets, must match that which is specified in the Case Report Tabulation – Data Definition (Define-XML) document. OpenCDISC uses both transport data sets and the Define-XML to perform the validation. Besides metadata checks, the application also checks for appropriate controlled terminology values (e.g. F, M, or U for the variable DM.SEX) and standard formats, such as using the ISO 8601 format for date/time values. OpenCDISC does not guarantee 100% compliance. However, it does a good job of detecting most data issues that would otherwise delay a submission. This application is a timely and important asset to conformance of the clinical trials data with the submission standards.

Wouldn't it be great if the OpenCDISC validator report was enhanced to have not only the issues but the data points that resulted in those issues as well as the list of Controlled Terminology for the variable in one combined enhanced report?

In order to expedite quality review time, this paper will introduce an enhanced version of the OpenCDISC validator v2.0.1 or above report. This enhanced report will have SDTM data (only those rows with issues) and SDTM terminology added as separate worksheets in the original report. Later, hyperlinks between each message in the report and the related SDTM data or SDTM code list worksheets are added using the Excel formulas and Visual Basic for Application (VBA) macros via DDE (Dynamic Data Exchange). These hyperlinks will further provide point and click options to check the data and code list related issues immediately in the enhanced report which will save

significant amount of time with minimal coding. This enhanced report can be further developed to cover Analysis Data Model (ADaM), and Standard for Exchange of Nonclinical Data (SEND) databases. The entire process involves SAS, Excel formulas, and VBA macros which are automated by a SAS macro called %Create_Enhanced_OCV.

TECHNIQUE & MECHANISM

The general process of adding new features in an OpenCDISC validator report is as follows:

1. Read "Details" worksheet from OpenCDISC validator report in SAS.
2. Subset all SDTM data sets by merging the column "Record" and "Domain" from step 1. Export all data sets to file 'Data.xls'.
3. Export all the subset data sets to original OpenCDISC validator report in separate worksheets. For e.g. LB and CM worksheets are added in below output 1. This is handled by the excel VBA macro.
4. Read SDTM Terminology worksheet in SAS and export this worksheet to the original OpenCDISC report. For e.g. worksheet "Controlled_Term" is added in below output 1. This is handle by the excel VBA macro.
5. Copy worksheet "Details" to "Details_Data" using VBA macro. For e.g. worksheet "Details_Data" is added in below screenshot 1.
6. Add hyperlinks in "Details_Data" tab using excel formulas and VBA macros. These hyperlinks will link multiple worksheets to "Details_Data" tab. Please see screenshots below for detail. Column "Data_Link" will have hyperlinks connecting to the subset SDTM data in each worksheet per domain as listed in step 3 and the "CT_Link" column will have hyperlinks connecting Controlled terminology in the "Controlled_Term" worksheet as listed in step 4.
7. Save the file.

To automate these steps, the DDE solution is leveraged to build the communication bridge between SAS v9.2 and Excel 2003. Excel v4.0 macro functions (Microsoft Corporation, 2000) can be sent from SAS via DDE to enable SAS to take control of Excel.

%CREATE_ENHANCED_OCV

To facilitate and automate the above discussed steps, a SAS macro called %Create_Enhanced_OCV was developed for SAS v9.2 or above. The user can easily extend the macro to fit other SAS versions. This macro will perform the following steps:

- Assign require SAS options and various locations for e.g. data location, OpenCDISC report location.
- Delete old Excel files using macro %DELETE_DATA_FILE macro. This macro is further discussed in later part of paper.
- Export data to Excel using macro %EXCEL_DATASET macro. This macro is further discussed in later part of paper.
- Execute VBA macros through %EXCEL_VBA_FORMULA macro. This macro is further discussed in later part of paper.

There are only four keyword parameters:

- **Data_path**: Define the data location.
- **OCV_path**: Define the OpenCDISC validator report path.
- **OCV_filename**: Define the OpenCDISC validator report filename.
- **CT_filename**: Define the Controlled terminology filename. Please copy this file to the location noted in the parameter &ocv_path.

Please see comments in SAS macro below for details.

```
%MACRO CREATE_ENHANCED_OCV (data_path=,ocv_path=,ocv_filename=,ct_filename=);  
  
  /*Save and Set options*/  
  %LET saveoptions=%SYSFUNC (GETOPTION(xsync)) %SYSFUNC (GETOPTION(xwait));  
  OPTIONS NOXWAIT NOXSYNC ;  
  
  /*Initiate OCV report path and filename*/  
  %LET in=&OCV_path.&OCV_Filename;  
  /*Initiate Controlled Terminology file path and filename*/
```

```

%LET in_ct=&OCV_path.&CT_Filename;
/*Initiate data file path and filename*/
%LET in_data=&OCV_path.data.xls;
/*Initiate Enhanced OCV file path and filename*/
%LET out=&OCV_path.Enhanced_OCV_Report.xlsx;
/*Macro Call to delete old files*/
%DELETE_DATA_FILE (OCV_path=&ocv_path,out=&out)

/*Macro Call to create data.xls*/
%EXCEL_DATASET(data_path=&data_path,ocv_path=&ocv_path,
               ocv_filename= &ocv_filename)

/*Macro call to execute VBA macros*/
%EXCEL_VBA_FORMULA(in=&in,in_data=&in_data,in_ct=&in_ct,out=&out)

/*reset options to initial values*/
OPTIONS &saveoptions;

%MEND CREATE_ENHANCED_OCV;

```

Below is the sample macro call to macro %CREATE_ENHANCED_OCV.

```

%CREATE_ENHANCED_OCV(data_path=U:\Data,ocv_path=U:\OCV\,
                    ocv_filename=opencdisc-report-2016-02-22T05-41.xlsx,ct_filename=SDTM Terminology.xls);

```

ENHANCED OPENCDISC REPORT OUTPUTS:

Please see below screenshots (with description below) from enhanced OpenCDISC report for detail.

| | A | B | D | E | F | G | H | I | J | K |
|------|----------------------|----------------------|--------|-----|-----|-----------|-------------|------------------------|-------------------------|--|
| 1 | Data Link | CT Link | Domain | Rec | Cou | Variables | Values | OpenCDISC | Publisher | Message |
| 279 | Link | Link | CM | | 1 | CMDOSFRQ | 1 SACHET BD | CT2002 | FDAC341 | CMDOSFRQ value not found in 'Frequency' extensible |
| 281 | Link | Link | CM | | 1 | CMDOSFRQ | 2 HOURLY | CT2002 | FDAC341 | CMDOSFRQ value not found in 'Frequency' extensible |
| 282 | Link | Link | CM | | 1 | CMDOSFRQ | DAY | CT2002 | FDAC341 | CMDOSFRQ value not found in 'Frequency' extensible |
| 283 | Link | Link | CM | | 1 | CMDOSFRQ | 3 HOURLY | CT2002 | FDAC341 | CMDOSFRQ value not found in 'Frequency' extensible |
| 701 | Link | | LB | | 1 | LBTESTCD | 01H66 | SD0018 | FDAC058 | Invalid value for LBTESTCD variable |
| 702 | Link | | LB | | 2 | LBTESTCD | 01H66 | SD0018 | FDAC058 | Invalid value for LBTESTCD variable |
| 703 | Link | | LB | | 3 | LBTESTCD | 01H66 | SD0018 | FDAC058 | Invalid value for LBTESTCD variable |
| 3368 | | | | | | | | | | |
| 3369 | | | | | | | | | | |
| 3370 | | | | | | | | | | |

Output 1: Enhanced OpenCDISC report with hyperlinks in columns Data_Link and CT_Link.

| | A | B | C | D | E | F |
|------|------------------------|---------------|------------------------------|---------------|------------------------|-------------------------|
| 1 | Code | Codelist Code | Codelist Extensible (Yes/No) | Codelist Name | CDISC Submission Value | CDISC Synonym(s) |
| 2099 | C71113 | | Yes | Frequency | FREQ | Frequency |
| 2100 | C64526 | C71113 | | Frequency | 1 TIME PER WEEK | One Time Per Week |
| 2101 | C64497 | C71113 | | Frequency | 2 TIMES PER WEEK | BIS; Twice per week |
| 2102 | C98861 | C71113 | | Frequency | 2 TIMES PER YEAR | 2 Times Per Year |
| 2103 | C98859 | C71113 | | Frequency | 3 TIMES PER MONTH | 3 Times Per Month |
| 2104 | C64528 | C71113 | | Frequency | 3 TIMES PER WEEK | TIS; Three times a week |
| 2105 | C98860 | C71113 | | Frequency | 3 TIMES PER YEAR | 3 Times Per Year |
| 2106 | C98852 | C71113 | | Frequency | 4 TIMES PER MONTH | 4 Times Per Month |

Output 2: Hyperlink from row 279 in Column “CT_Link” from worksheet “Details_Data” connected to “Frequency” code list for issue CT2002.

| Record | STUDY | DOMAIN | USUBJID | LBSEQ | LBREFID | LBSPID | LBTEST | LBTEST | LBCAT | LBSCAT | LBORR | LBORR | LBORN | LBORN |
|--------|-------|--------|--------------------|-------|---------|--------|--------|---------|---------------|---------------------|-------|-------|----------------|-------|
| 1 | DUMMY | LB | DUMMY/ E0001001 | 1 | A2 | | 01H66 | pPRAS40 | BIOMARK ER | Pharmaco dynamic | 0.243 | | Absorban ce | |
| 2 | DUMMY | LB | DUMMY/ E0001001 | 2 | A3 | | 01H66 | pPRAS40 | BIOMARK ER | Pharmaco dynamic | 0.251 | | Absorban ce | |
| 3 | DUMMY | LB | DUMMY/ E0001001 | 3 | A1 | | 01H66 | pPRAS40 | BIOMARK ER | Pharmaco dynamic | 0.221 | | Absorban ce | |

Output 3: Hyperlink from row=701 in Column "Data_Link" from worksheet "Details_Data" connected to Record=1 in LB worksheet for issue SD0018.

SAS MACRO TO EXPORT DATA IN EXCEL

To export common records from the OpenCDISC validator report and data sets in Excel file Data.xls, a SAS macro called %EXCEL_DATASET was developed in SAS v9.2 or above. This macro will read all data sets from a given location and later export the common data in Excel file. Please see comments in SAS macro below for details.

```

/*Macro to export data to excel files*/
%MACRO EXCEL_DATASET(data_path=,ocv_path=,ocv_filename=);

/*Select all SAS data sets from a given location */

FILENAME dirlist PIPE "dir "&data_path" /b ";

DATA dirlist1 ;
    INFILE dirlist LENGTH=reclen ;
    INPUT filename $varying1024. reclen ;
    IF UPCASE(SCAN(filename,2,'.')) IN ('SAS7BDAT');
    filename1=UPCASE(STRIIP(SCAN(filename,1,'.')));
RUN;

PROC SORT DATA=dirlist1;
    BY filename;
RUN;

/*Create macro variables for filename and count the total number of files*/

PROC SQL NOPRINT;
    SELECT COUNT(*)
    INTO :n
    FROM dirlist1

    SELECT filename1
    INTO :file1-:file%cmpres(&n)from dirlist1
    ;
QUIT;

/*Read "Details" worksheet from OCV report*/

LIBNAME dat2 EXCEL "&OCV_path.&OCV_filename" HEADER=YES MIXED=YES;

DATA full_ocv;
    SET dat2."Details$"n ;
RUN;

LIBNAME dat2 clear;

/*Select record in SAS data sets by merging the OCV report and data set using
domain and record variable*/

```

```

%DO i = 1 %TO &n ;

    LIBNAME dat "&data_path" ACCESS=READONLY;

    DATA full_&&file&i;
        SET dat.&&file&i;
        record=_n_;
    RUN;

    DATA ocv1_&&file&i.(KEEP=record );
        SET full_ocv;
        WHERE UPCASE(domain)="&&file&i" and NOT MISSING(record);
    RUN;

    PROC SORT DATA=full_&&file&i;
        BY record;
    RUN;

    PROC SORT DATA=ocv1_&&file&i. OUT= ocv_&&file&i. NODUPKEY;
        BY record;
    RUN;

    DATA &&file&i;
        MERGE ocv_&&file&i. (IN=a) full_&&file&i.(IN=b);
        BY record;
        IF a and b;
    RUN;

    PROC SQL NOPRINT;
        SELECT COUNT(*) INTO:n&i from &&file&i;
    QUIT;

/*If there is a common record present in the "Details" worksheet and data set
then export the data set in unique worksheet*/

    %IF &&n&i. %THEN %DO;

        LIBNAME dat1 EXCEL "&OCV_path.Data.xls" ;

        DATA dat1.&&file&i;
            SET &&file&i;
        RUN;

        LIBNAME dat1 CLEAR;

    %END;

%END ;

%MEND EXCEL_DATASET;

```

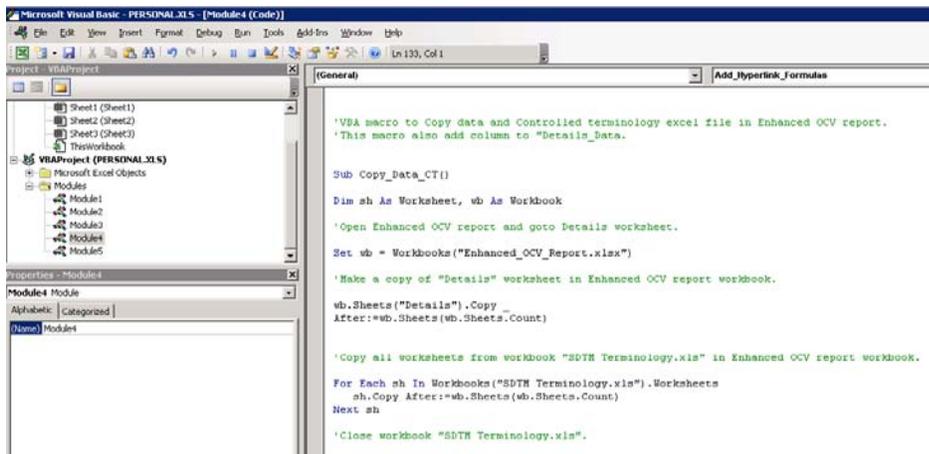
RECORD VBA MACROS IN EXCEL TO ADD DATA AND HYPERLINKS IN OPENCDISC VALIDATOR REPORT

For many, perhaps most SAS developers the easiest and fastest way to write VBA code is to first go into a Microsoft application (in this paper Microsoft Word is used as the application), turn on the macro recorder, perform the steps and functions desired, and then terminate the macro recorder. The end result is a stored VBA program (macro) that can be executed at a later date. This makes it possible for a SAS developer to automate tasks in the Microsoft application, and therefore vastly improve the functionality of an integrated system that takes advantage of the relative strengths of the SAS System and the Microsoft application.

Note, below macros are developed using Excel version 2003 and minor adjustments will be needed when use with other version of Excel.

Steps to record a VBA Macro are as follows:

1. Open Excel.
2. Select Tools, Macro, and Record New Macro from the pull down menus.
3. Enter the name of macro as 'Copy_Data_CT' or 'Add_Hyperlink_Formulas'.
4. Select Ok. Make sure to store the macro in the Personal Macro Workbook.
5. Stop Recording.
6. Click "ALT+F11".
7. Go to Personal VB project module.
8. Paste the 'Copy_Data_CT' or 'Add_Hyperlink_Formulas' VBA macro. Please see screenshots below.
9. Select **Ok**.
10. Select **Ok**.
11. Close the document.



The screenshot shows the Microsoft Visual Basic Editor window for 'PERSONAL.XLS'. The Project Explorer on the left shows the 'VBAProject (PERSONAL.XLS)' with a 'Modules' folder containing 'Module1' through 'Module5'. The Properties window shows 'Module1' selected. The main code window displays the following VBA code for the 'Copy_Data_CT' macro:

```
'VBA macro to Copy data and Controlled terminology excel file in Enhanced OCV report.
'This macro also add column to "Details_data".

Sub Copy_Data_CT()
    Dim sh As Worksheet, wb As Workbook

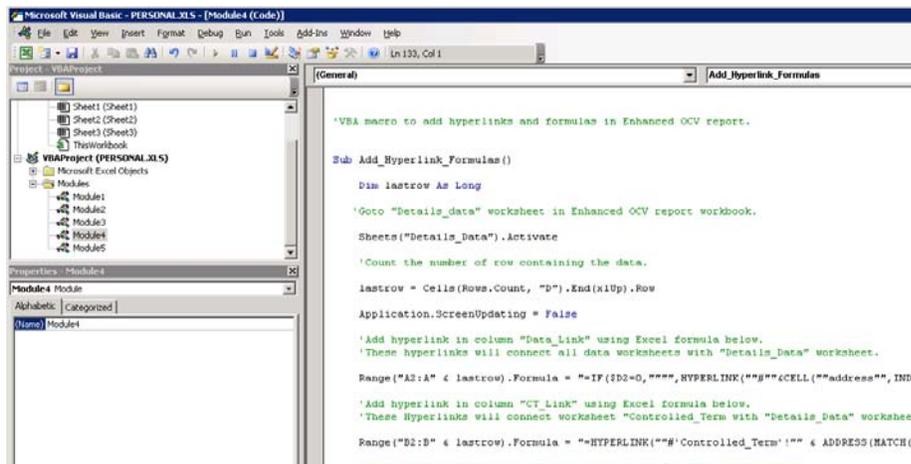
    'Open Enhanced OCV report and goto Details worksheet.
    Set wb = Workbooks("Enhanced_OCV_Report.xlsx")

    'Make a copy of "Details" worksheet in Enhanced OCV report workbook.
    wb.Sheets("Details").Copy _
    After:=wb.Sheets(wb.Sheets.Count)

    'Copy all worksheets from workbook "SDTM Terminology.xlsx" in Enhanced OCV report workbook.
    For Each sh In Workbooks("SDTM Terminology.xlsx").Worksheets
        sh.Copy After:=wb.Sheets(wb.Sheets.Count)
    Next sh

    'Close workbook "SDTM Terminology.xlsx".
```

Display 1: 'Copy_Data_CT' VBA macro in Visual Basic Editor.



The screenshot shows the Microsoft Visual Basic Editor window for 'PERSONAL.XLS'. The Project Explorer on the left shows the 'VBAProject (PERSONAL.XLS)' with a 'Modules' folder containing 'Module1' through 'Module5'. The Properties window shows 'Module1' selected. The main code window displays the following VBA code for the 'Add_Hyperlink_Formulas' macro:

```
'VBA macro to add hyperlinks and formulas in Enhanced OCV report.

Sub Add_Hyperlink_Formulas()
    Dim lastrow As Long

    'Goto "Details_data" worksheet in Enhanced OCV report workbook.
    Sheets("Details_data").Activate

    'Count the number of row containing the data.
    lastrow = Cells(Rows.Count, "D").End(xlUp).Row

    Application.ScreenUpdating = False

    'Add hyperlink in column "Data_Link" using Excel formula below.
    'These hyperlinks will connect all data worksheets with "Details_data" worksheet.
    Range("A2:A" & lastrow).Formula = "=IF(SD2=0,"" "",HYPERLINK("&""&CELL("&""&ADDRESS"&""& INDR

    'Add hyperlink in column "CT_Link" using Excel formula below.
    'These Hyperlinks will connect worksheet "Controlled_Term with "Details_data" worksheet
    Range("B2:B" & lastrow).Formula = "=HYPERLINK("&""&"#&""&Controlled_Term"&""& ADDRESS(MATCH(S
```

Display 2: 'Add_Hyperlink_Formulas' VBA macro in Visual Basic Editor.

VBA MACRO: COPY_DATA_CT

Please see below VBA code for **COPY_DATA_CT** which will perform the following tasks:

- Copy worksheet Controlled Terminology into the OpenCDISC validator report. Later rename the worksheet to "Controlled_Term".

- Create a copy of the worksheet "Details" into the OpenCDISC validator report and rename it to "Details_Data".
- Add columns Data_Link, CT_Link, CT_Name in worksheet "Details_data" of workbook OpenCDISC validator report. These columns are needed to add the hyperlinks
- Copy all worksheets from Data.xls in OpenCDISC validator report.
- Close and save all files.

Please see comments in green text for detail.

```

Sub Copy_Data_CT()
    Dim sh As Worksheet, wb As Workbook

    'Open Enhanced OCV report and goto Details worksheet.

    Set wb = Workbooks("Enhanced_OCV_Report.xlsx")

    'Make a copy of "Details" worksheet in Enhanced OCV report workbook.

    wb.Sheets("Details").Copy _
        After:=wb.Sheets(wb.Sheets.Count)

    'Copy all worksheets from workbook "SDTM Terminology.xls" in Enhanced OCV
    report workbook.

    For Each sh In Workbooks("SDTM Terminology.xls").Worksheets
        sh.Copy After:=wb.Sheets(wb.Sheets.Count)
    Next sh

    'Close workbook "SDTM Terminology.xls".

    Workbooks("SDTM Terminology.xls").Close

    'Remove "Readme" worksheet from Enhanced OCV report workbook.

    Application.DisplayAlerts = False
    Sheets("Readme").Select
    ActiveWindow.SelectedSheets.Delete

    'Save workbook Enhanced OCV report.

    ActiveWorkbook.Save

    'Rename copy of "Details" worksheet to "Details_data" in Enhanced OCV report
    workbook.

    wb.Sheets("Details (2)").Name = "Details_Data"

    'Rename 'SDTM--' CT worksheet to "Controlled_Term" in Enhanced OCV report
    workbook.

    For Each sh In Sheets
        If Left(sh.Name, 4) = "SDTM" Then
            sh.Name = "Controlled_Term"
        End If
    Next sh

    'Save workbook Enhanced OCV report.

    ActiveWorkbook.Save

```

```

'Add columns Data_Link, CT_Link, CT_Name in "Details_data" in Enhanced OCV
report workbook.

Sheets("Details_Data").Activate
Columns("A:A").Select
    Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
    Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
    Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
'Insert variable name in first row.
Range("A1").Select
    ActiveCell.FormulaR1C1 = "Data_Link"
Range("B1").Select
    ActiveCell.FormulaR1C1 = "CT_Link"
Range("C1").Select
    ActiveCell.FormulaR1C1 = "CT_Name"
'Highlight all three columns heading to green.
Range("A1:C1").Select
    With Selection.Interior
        .ColorIndex = 50
        .Pattern = xlSolid

End With

'Hide column C.
Columns("C:C").EntireColumn.Hidden = True

'Copy all worksheets from workbook "Data.xls" in Enhanced OCV report workbook.

For Each sh In Workbooks("Data.xls").Worksheets
    sh.Copy After:=wb.Sheets(wb.Sheets.Count)
Next sh

'Close workbook "Data.xls".
Workbooks("Data.xls").Close

'Save workbook Enhanced OCV report.
ActiveWorkbook.Save

End Sub

```

VBA MACRO: ADD_HYPERLINK_FORMULAS

Please see below VBA code for **ADD_HYPERLINK_FORMULAS** which will perform the following tasks:

- Open Enhanced OpenCDISC validator report and go to worksheet "Details_Data".
- Count the number of rows containing the data.
- Insert hyperlink in column "Data_Link" using Excel formula. These hyperlinks will connect all data worksheets with "Details_Data" worksheet.
- Select the codelist name from the comments using Excel formula in column "CT_Name"
- Insert hyperlink in column "CT_Link" using Excel formula. These Hyperlinks will connect worksheet "Controlled_Term" with "Details_Data" worksheet.
- Close and save all files.

Please see comments in green text for detail.

```

Sub Add_Hyperlink_Formulas()

    Dim lastrow As Long

    'Goto "Details_data" worksheet in Enhanced OCV report workbook.

    Sheets("Details_Data").Activate

```

```

'Count the number of row containing the data.

lastrow = Cells(Rows.Count, "D").End(xlUp).Row

Application.ScreenUpdating = False

'Add hyperlink in column "Data_Link" using Excel formula below.
'These hyperlinks will connect all data worksheets with "Details_Data"
\worksheet.

Range("A2:A" & lastrow).Formula =
"=IF($E2="","", "", IF($D2=0, "", HYPERLINK("#"&CELL("address", INDEX(INDIRECT
(D2&"!A:A"), MATCH(E2, INDIRECT(D2&"!A:A"), 0))), "Link"))"

'Add hyperlink in column "CT_Link" using Excel formula below.
'These Hyperlinks will connect worksheet "Controlled_Term with "Details_Data"
\worksheet.

Range("B2:B" & lastrow).Formula =
"=IF($C2="","", "", HYPERLINK("#'#Controlled_Term!'!" & ADDRESS(MATCH($C2,
Controlled_Term!$D:$D, 0), 1), "Link"))"

'Select the codelist name from the comments using Excel formula below.

Range("C2:C" & lastrow).Formula =
"=IF(ISERROR(FIND("codelist", K2, 1)), "", MID(K2, FIND("'", K2) + 1, FIND("'", K
2, FIND("'", K2) + 1) - FIND("'", K2) - 1))"

Application.ScreenUpdating = True

'Save and Close workbook Enhanced OCV report.

ActiveWorkbook.Save
Workbooks("Enhanced_OCV_Report.xlsx").Close

End Sub

```

Note, if the data is missing from columns “Record” or “CT_Name” or “Domain” in the Enhanced OpenCDISC validator report then the above macro will put a blank value instead of link. There is a possibility that due to some data issues the user will see error values such as ‘#N/A’, ‘#REF!’, ‘#VALUE!’ instead of links. These values require further investigation and can be replaced with a blank by adding additional conditions in the Excel formula.

SAS MACRO TO EXECUTE THE VBA MACROS

To execute VBA macros a SAS macro called **%Excel_VBA_Formula** was developed for SAS v9.2 or above. On a high level the macro **%EXCEL_VBA_FORMULA** will perform the following tasks:

- Open Excel through SAS via DDE and clear all messages.
- Create a copy of OpenCDISC report.
- Open Excel files containing the Controlled Terminology and data.
- Execute VBA Macros ‘Add_Hyperlink_Formulas’ and ‘Copy_Data_CT’.
- Save and close Excel files.

Please see comments in SAS macro code below for details.

```

/*Macro to execute VBA macros*/
%MACRO EXCEL_VBA_FORMULA(in=, in_data=, in_ct=, out=);

/*Start Excel*/

%LET rc=%SYSFUNC(SYSTEM(START EXCEL));

```

```
/* Add sleep time to avoid errors from occurring due to delays in the execution of
previous SAS commands/data steps*/
```

```
DATA _null_;
  X=SLEEP(5);
RUN;
```

```
FILENAME excel DDE 'Excel|System';
```

```
DATA _null_;
  X=SLEEP(5);
RUN;
```

```
DATA _null_;
  FILE excel;
/*Clear all messages using below statement*/
  PUT '[Error(false)]';
/*Open OCV report*/
  PUT '[Open("' &in" ')]';
  X=SLEEP(2);
/*Save OCV report as "Enhanced OCV report.xlsx"*/
  PUT '[Save.as("' &out" ')]';
  X=SLEEP(2);
/*Close Excel*/
  PUT '[Close]';
  PUT '[Quit]';
RUN;
```

```
/*Start Excel*/
%LET rc=%SYSFUNC(SYSTEM(START EXCEL));
```

```
DATA _null_;
  X=SLEEP(5);
RUN;
```

```
FILENAME excel1 DDE 'Excel|System';
```

```
DATA _null_;
  X=SLEEP(5);
RUN;
```

```
DATA _null_;
  FILE excel1;
/*Clear all messages using below statement*/
  PUT '[Error(false)]';
/*Open CT Terminology.xls*/
  PUT '[Open("' &in_ct" ')]';
  X=SLEEP(5);
/*Open data.xls*/
  PUT '[Open("' &in_data" ')]';
  X=SLEEP(5);
/*Open Enhanced OCV report*/
  PUT '[Open("' &out" ')]';
  X=SLEEP(5);
/*Execute VBA macro Copy_Data_CT from personal workbook*/
  PUT '[RUN("personal.xls!Copy_Data_CT")]';
  X=SLEEP(10);
/*Execute VBA macro Add_Hyperlink_Formulas from personal workbook*/
  PUT '[RUN("personal.xls!Add_Hyperlink_Formulas")]';
  X=SLEEP(2);
RUN;
```

```

DATA _null_;
  X=SLEEP(5);
RUN;

FILENAME excel_ DDE 'Excel|System';

DATA _null_;
  FILE excel_;
  /*Clear all messages and close Excel*/
  PUT '[Error(false)]';
  PUT '[Quit]';
RUN;

%MEND EXCEL_VBA_FORMULA;

```

Note, the SLEEP function is a useful function and can be used frequently in the SAS command/data step that is dependent upon completion of previous jobs. This will help avoid errors from occurring due to delays in the execution of previous SAS commands/data steps.

DELETE EXISTING EXCEL OUTPUTS FROM PRODUCTION LOCATION

To delete Excel file for e.g. Data.xls and Enhanced_OCV_Report.xlsx (if exists from previous execution), a SAS macro called %DELETE_DATA_FILE was developed for SAS v9.2 or above. Please see comments in SAS code below for details

```

/* Macro to Delete all existing excel files from a given location*/
%MACRO DELETE_DATA_FILE(OCV_path=,out=);

  FILENAME fileref "&OCV_path.Data.xls";

  /* Delete the old output file*/

  %LET rc0=%SYSFUNC(FEXIST(FILEREF));
  %IF &rc0=1 %THEN %DO;
    %LET rc1=%SYSFUNC(FDELETE(FILEREF));
    %PUT &rc1;
  %END;

  FILENAME fileref1 "&out.";

  /* Delete the old output file*/

  %LET rc2=%SYSFUNC(FEXIST(FILEREF1));
  %IF &rc2=1 %THEN %DO;
    %LET rc3=%SYSFUNC(FDELETE(FILEREF1));
    %PUT &rc3;
  %END;

%MEND DELETE_DATA_FILE;

```

Above macro will delete any existing Excel output to avoid any errors.

CONCLUSION

Macro %Create_Enhanced_OCV will add hyperlinks in the OpenCDISC validator report without any manual pre-processing. These hyperlinks will further provide point to click options to check the data and code list related issues immediately in the enhanced report. This enhanced report takes the typical OpenCDISC validator to a much higher user-friendly level. It's a one stop report where the user has easy access to the results of the OpenCDISC checks, actual data points that caused the issues as well as the Controlled Terminology. It increases efficiency in reviewing the output report and almost eliminates all the above mentioned issues with a typical OpenCDISC report. The user has a better understanding of the results of the checks and thus results in better quality of data for submission while reducing the amount of time require for reviewing the report. This enhanced report can be further developed to cover ADaM, and SEND databases.

REFERENCES

Gupta Ajay, 2013. Combining SAS ® LIBNAME and VBA Macro to Import Excel ® file in an Intriguing, Efficient way. Proceedings of the PharmaSUG 2013 Conference, paper CC11.

<http://www.opencdisc.org/> and <http://www.cdisc.org>

Redner Virginia and Gerlach John, 2011. Resolving OpenCDISC Error Messages Using SAS®. Proceedings of the PharmaSUG 2011 Conference, paper CD07.

Guo Annie, 2012. OpenCDISC Plus. Proceedings of the PharmaSUG 2012 Conference, paper AD18.

Lee Kevin, 2013. Running OpenCDISC in SAS. Proceedings of the PharmaSUG 2013 Conference, paper AD05.

Roper, C.A. 2000. Using SAS and DDE to execute VBA macros in Microsoft Excel. Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference, paper 98.

Websites: <http://msdn.microsoft.com>, <http://stackoverflow.com/>, <http://www.google.com/>

ACKNOWLEDGMENTS

Thanks to Lindsay Dean, Daniela Popa, Ken Borowiak, David Gray, Richard DAmato, Lynn Clipstone, Thomas Fritchey, Lynn Mullins, Latonya Murphy, Sajeet Pavate, and PPD Management for their reviews and comments. Thanks to my family for their support.

DISCLAIMER

The content of this paper are the works of the authors and do not necessarily represent the opinions, recommendations, or practices of PPD.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Ajay Gupta, M.S.
Enterprise: PPD
Address: 3900 Paramount Parkway
City, State ZIP: Morrisville, NC-27560
Work Phone: (919)-456-6461
Fax: (919)-654-9990
E-mail: Ajay.Gupta@ppdi.com,
Ajaykailasgupta@aol.com



SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.