

# TRANSFORM A SAS® DATASET TO AN EXCEL FILE FORMAT

Pravin vijay Varatharaj, Cognizant Technology Solutions

## ABSTRACT

For many of us, using SAS® and Microsoft Excel together is unavoidable, it is often necessary for a SAS® programmer to transform a SAS® dataset to an Excel file format. Several methods for performing this transformation have been published in previous SAS® conference proceedings.

In windows, PC SAS® uses DDE (Dynamic Data Exchange) mechanism in SAS® enables SAS® to control Excel template. This paper talks about how to transform a SAS® dataset to an Excel template file format from the Linux server or Unix platform, Python language can be used to transform the SAS® datasets to Excel template.

Python package like sas7bdat can be used to read the SAS® datasets and packages like openpyxl can be used to write the content to the Excel template. Python language helps to break the dependency of DDE communication mechanism to transform a SAS® dataset to an Excel template file format and developers to schedule the jobs in the crontab to automate.

## INTRODUCTION

This paper describes a custom Python function that transform a SAS® dataset to an Excel template file format. The python function is driven by certain preset parameters which can be overridden by the users.

This function can be used in batch programs as well as interactive Python sessions.

## BACKGROUND AND ASSUMPTIONS

This paper assumes that users are familiar with the Python programming language environment (i.e. reading SAS® dataset, working with Excel). The macro has been designed for a Linux based system, but with some minor modifications, it can work in other environments as well.

This code requires SAS® dataset which needs to be transformed and Excel template file.

## TRANSFORM A SAS® DATASET TO AN EXCEL FILE FORMAT.

Python is open source and freely available for download. Many operating systems such as Linux and Mac OS X (but not Windows) have it installed by default. Python is installed with a standard library that contains several modules, some written in C, some in Python, each of which addresses a unique functionality. Because Python is open source, community users can also build their own modules and through code sharing repositories such as Git, contribute them to the whole Python community. Additional packages not available with the standard library can be downloaded. An import statement in the Python program then gives access to its functionality. This transform SAS® Dataset to an excel format has various advantages:

- You can use Python script (Sample Python script explained later in this paper) to read SAS® datasets and export them to excel template.

- You can schedule the python codes in schedulers (e.g. Cron) for automated run on the Windows and Linux environment.

There are two python packages used in this paper to transform a SAS® dataset to an Excel. All the packages are optional or replaceable and maybe omitted by users if the pre-requisites are met.

1. Python module "sas7bdat" has capability of reading the SAS® dataset without any dependency on SAS® software.

**Python Version:**

2.6+, 3+

**To install, run:**

```
pip install sas7bdat
```

**Import The "sas7bdat" module:**

```
from sas7bdat import SAS7BDAT
```

**Read the SAS dataset:**

```
read_dataset = SAS7BDAT(r"/root/sample_dataset.sas7bdat
```

**Pandas DataFrame can be achieved by using the "to\_data\_frame" method.**

```
read_dataset_df = read_dataset.to_data_frame()
```

**Link for reference:**

<https://pypi.org/project/sas7bdat/>

2. Python module "openpyxl" has capability of to read or write Excel 2010 xlsx/xlsm/xltx/xltx files.

**Python Version:**

2.6+, 3+

**To install, run:**

```
pip install openpyxl
```

**To open the Excel template:**

```
wb_template = openpyxl.load_workbook(r"/root/sample_excel.xlsx")
```

**To write the Excel template:**

```
wb_template_sheet1.cell(row=1,1).value=['1']
```

**Link for reference:**

<https://pypi.org/project/openpyxl/>

## Sample Python Code:

```
import openpyxl
from sas7bdat import SAS7BDAT
read_dataset = SAS7BDAT(r"/root/sample.sas7bdat")
read_dataset_df = read_dataset.to_data_frame()
dataset_df = read_dataset_df
wb_template = openpyxl.load_workbook(r"/root/sample.xlsx")
wb_template_sheet1 = wb_template["Sheet1"]
dataset_df_col = len(dataset_df.columns)
dataset_df_row = len(dataset_df.index)
for i1 in range(0,dataset_df_row):
    for j1 in range(1,dataset_df_col):
        try:
            wb_template_sheet1.cell(row=i1+2,column=j1).value=sas_ps1.iloc[i1,j
1]
        except Exception as e:
            err = str(e)
wb_template.save(r"/root/sample_excel.xlsx")
```

## **CONCLUSION**

Transform a SAS® DATASET to an excel template(. Xltx) using simple python code. Users can use simple Python code to transform SAS® Dataset to an excel template and provide an ability to read SAS® datasets and export them to excel template from Linux servers .

## **CONTACT INFORMATION**

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

Pravinvijay Varatharaj  
Cognizant Technology Solutions  
Tampa, FL 33626  
Phone: 1 (813) 724-4046  
E-mail: [pravin.vijay@Cognizant.com](mailto:pravin.vijay@Cognizant.com) / [vpvijay87@gmail.com](mailto:vpvijay87@gmail.com)