September 2023

Proteus User Guide
Data validation application for business plan tables

Ofwat

About this document

The purpose of this document is to:

- introduce the data validation software application named, 'Proteus';
- explain what it is intended to do;
- explain how to run the application; and
- explain how to interpret its output.

Introduction

At PR24 (Price Review 24) we will receive a large amount of data from companies. We will store the data in a secure, accessible way facilitating the efficient running of our PR24 models. We load all key regulatory data into our database, 'Fountain'. Data on Fountain is considered the 'single version of the truth' which is used to inform companies' determinations.

We developed a software application, 'Proteus', to help us to identify some of the potential issues that may arise in how companies populate our business plan tables. It is created for Windows computers. This application allows Ofwat to quickly check if data entered into companies' business plan tables are consistent with our specific data standards.

Proteus will help to ensure data integrity and compliance within Ofwat's data solutions in a proactive way.

At this stage, the primary objective of Proteus is to help us identify:

- data discrepancies that might prevent us loading data into fountain;
- a specific set of potential unexpected changes to our published template;
- a specific set of potential data validation issues (for example, whether text data is entered where numeric data is expected); and
- certain types of missing data

We will not rely on Proteus to help us identify all data issues.

Although Proteus is intended for internal usage, we are publishing it in the interests of transparency on GitHub. We will continue to publish any updates to this application on GitHub: https://github.com/Ofwat/Proteushttps://github.com/Ofwat/Proteus.

What does Proteus do?

Proteus is a data validation and verification software application that helps ensure that the collected data reflect its intended purpose. The application checks that data in the destination file (a company's set of business plan tables) matches certain expectations.

The tool automatically performs a level of verification of the populated data, matching all the records and flagging exceptions.

Proteus validates user input. Proteus' main functionality is:

1. ensuring the data entered is in the correct type, format and within a valid range;
2. detecting types of missing data; and
3. detecting if unintended changes have been made – for example, accidentally deleting a row, change of header names, delete a mandatory column.

Proteus is developed using Python code. This enables rapid analysis of large amounts of data. Proteus analyses the standardised data going into Fountain, that is, the F_Outputs worksheets. It does not directly analyse the individual business plan tables themselves.

This means that Proteus will only be able to capture a subset of potential data issues, albeit important ones.

Data Validation Rules – Explanation

Proteus applies seven data validation rules to help identify any errors or discrepancies in the collected data that may have occurred during the data population process. These checks are built into the application. They are each applied to the data in the F_output sheets. The outcome is the creation of an "Error Log" file. The rules deployed on Proteus are as follows.

Proteus applies seven data validation rules to help identify any errors or discrepancies in the collected data that may have occurred during the data population process. These checks are built into the application. They are each applied to the data in the F_output sheets. The outcome is the creation of an "Error Log" file. The rules deployed on Proteus are as follows:

Rule 1: This checks that the headers are correctly named and in the right order. It is case sensitive. This would, for example, identify alteration of the column names or any column deletion. An example error message is as follows:

> *'Acronym' column name is not correct, please note name is case sensitive. Please correct the header's names for this Excel Sheet: fOut_OUT, before data validation proceeds!*

Rule 2: The 'Reference' column contains 'Bon codes'. Every Bon code is a string of text. Fountain requires that Bon codes follow a certain pattern (a 'regular expression'). This validation rule checks that the Bon codes consist of the following in sequence: an upper-case letter; zero or more upper case letters or underscores; at least one digit; zero or more number upper case letters, underscores, or digits; and no lower case letters. Bon codes must not contain \- characters. An example error message is as follows:

*Error in Row 4: 'Reference': *_001WR_PR24 doesn't match the regular expression"*

Rule 3: This checks that all the Bon codes in the 'Reference' column have the suffix "_PR24". This helps us identify data originating from the business plan tables rather than, say, APR data. An example error message is as follows:

Error in Row 4: Reference: RR1_001WR does not has a suffix _PR24

Rule 4: This checks that data in the 'Unit' column is less than 21 characters. An example error message is as follows:

Error in Row 5: Reference: RR1_001WN_PR24: Unit must be <= 21 characters

Rule 5: This checks that data in the 'Description' column is less than 230 characters. An example error message is as follows:

Error in Row 5: Reference: RR1_001WN_PR24: Description must be <= 230 characters

Rule 6: This checks that values (that is, anything under the 'Years' headers) entered by the user are formatted correctly. For example, if the 'Unit' column is named as Text, the user should enter a string, not a number. An example error message is as follows.

Error in Row: 212, Reference: CW18_001_PR24, Column: 2025-30, Value: 0.0, This is a text field, please check that your input is in text format

Rule 7: This checks if there is any alteration or deletion of Bon codes in the 'Reference' column. It does this by comparing the original Bon codes with the target Bon codes, where target Bon codes are those sent by the companies after data submission. An example error message is as follows:

Reference (Bon codes) in worksheet in the original version but not in the amended version: RR1_001WWN_PR24

How to run this software

Proteus application should run after completing with data the version 6 of the business plan table template. Here are the details on how to run the Proteus application. For your reference here: Github can view an instructional video.

There are three key files required to run Proteus application. These are:

- **Comparison template** This is version 6 of the business plan table template published on our website. It does not contain any user input. This file is used as the basis of comparison with a company's set of business plan tables. The Excel file name is: Comparison.xlsx.

- **Populated template** This is the 'comparison template' although it should contain user input. Please fill in all the relevant sheets before running the application.

- **Proteus.exe**: The .exe file is a Windows-specific executable file format. When the user triggers it, the computer runs the code that the file contains.

The user should have the populated template, comparison template and Proteus.exe stored in the same location/folder. Proteus.exe file has dependencies on the excel files and stores them together to provide visibility to the application.

The steps to run Proteus are as follows.

1. Download the Proteus.exe file and the comparison template. Store the above files and the populated template in the same folder.
2. The comparison template file name is case sensitive. It is essential not to make any changes to this file.
3. After data completion save the populated template in the file format (.xlsx). To do so open the workbook you want to save. Click File > Save As. Pick the place where you want to save the workbook. For example, select Computer to save it in a local folder where Proteus.exe and Comparison template is also saved. In the Save as type list, click the file format Excel Workbook (*.xlsx).
4. Double-click the Proteus.exe file to run it. This brings up a console that asks user input filename of the data on which they wish to run the data validation rules.
5. The message that the user should see on the Windows Command Prompt is: "Enter filename (full path)".
6. Add the full (absolute) path of the populated template and press Enter to run the file. (A full path refers to the complete details needed to locate a file or folder, starting from the root element, and ending with the other subdirectories). For example, a full path is: C:\Users\Maria.diapouli\OneDrive - OFWAT\Python\validation_tool\original\Populated.xlsx
7. Once the user has entered the file's name, the console will close.
8. If the application runs successfully, the user will see an error log file in the same folder. Please look at the "Example of Error Log file .txt" as an indicator on what this error log file would look like, if no errors are detected.
9. Open the Error Log File to view the results of the data validation rules. (If the application couldn't run then the error log file will be empty. There is no information to display.)
10. At the beginning of the file there is a timestamp showing when the error log was created. Every time the application runs a new version of error log file is created.

Interpreting and correcting errors

Proteus uses the 'f_output' worksheets, named fOut_[description data tables], as the basis of comparison. To reduce the risk of accidental changes to these, these worksheets are hidden in our excel template.

If you see any error messages, it will be helpful to see the relevant f_output sheet that Proteus has examined. These are the steps to do so:

1. Open the populated template and unhide the relevant sheets. (Right-click on any visible sheet. On the menu that appears, select the sheets in the Unhide dialog that appears, and then select OK.)
2. Relevant sheets are those specified in the Error log and have shown errors. The name of the sheet is also shown alongside any data validation rule in the Error log.
3. In the Error log file, the first row shows a timestamp of the Proteus execution time. The message in the Error log file starts: “This file was created at:”
4. Bellow the timestamp the user can see the Excel sheet name and a list with the seven data validation rules.
5. Each validation rule is numbered and accompanied by a short description.
6. If the rule is applied successfully, the user message will be: “Success, no errors were detected!”
7. If there is an error the user could see it under data validation description. For reference, please look at the data validation rules explanation for error examples.
8. Each error will show the row position and Boncode/Reference (first column) value as indicators on where changes should be made.
9. The user will make the necessary changes on the relevant sheets and re-run the application.
10. If the errors have been fixed a new error log file will not display previous errors.
11. The user can re-run the application as many times as it needed to until no errors are present.

How to download Proteus.exe from GitHub

The Proteus application is hosted on Github. Below are the incrustations on how to download the .exe file.

1. Select the 'Proteus' repository on GitHub.
2. Click on the 'proteus.exe' tab.
3. In the center of the screen click on 'view raw'.
4. Your computer should be downloading the file - save the file in your desired location.
5. The file is saved with a '.crdownload' file extension. This is a temporary file extension used by the web browser.
6. Rename it to 'Proteus.exe'. Without renaming the file you cannot run the application.
7. A dialogue box will appear asking you to confirm your changes. Select ok.

8. The .exe file should be ready to open.

What is included?

Below are the files we have provided on Github to enable smooth run of Proteus:

README file: A guide that gives users a detailed description of the project. Documentation with guidelines on how to use the project. It has instructions on how to download and run the application .exe. Alternatively, there are instructions on how to run the project by executing python script (Proteus.py) instead of running Proteus.exe file.

Example of log file, named Error_Log_04.07.2023_08.33: Example of the expected outcome of error log file if the application run successfully.

Proteus.exe file: An executable file contains instructions that system executes when user double clicks the file icon. (On GitHub in the Python Code folder we also publish the associated python code.)

Comparison template, named "Comparison.xlsx": This is version 6 of the business plan table template. This is the copy of the Excel file created before the user enters data.

Instructional Video: A visual tool showing viewers how to do run the application.

Version published: 1 September 2023

Ofwat (The Water Services Regulation Authority) is a non-ministerial government department. We regulate the water sector in England and Wales.

Ofwat
Centre City Tower
7 Hill Street
Birmingham B5 4UA
Phone: 0121 644 7500

This document is also available from our website at www.ofwat.gov.uk.

Any enquiries regarding this publication should be sent to mailbox@ofwat.gov.uk.