Administration of Children Services

Macro Documentation

# Dataload Master Macro

*Alec Kleyer*

# Contents

# 1 Synopsis

The Master excel workbook contains several macros that are used to import new user-created data, validate that data, and then convert the data to an ASCII file.

- Data is imported from the user template located in the folder specified in the "DataFile Location" spreadsheet inside the Master workbook.
- Once data is imported, necessary validation macros should be run to ensure the imported data meets the appropriate guidelines.
- Once validation is complete, the imported, and now validated, data will be saved as a pipe-delimited file to the same 'MIS' folder as the Master excel workbook.
- Lastly, the error report will be displayed in a separate sheet, "Empty-Cells", in addition to being sent out to a list of recipients provided in another sheet, "Email Recipients".

# 2 Dataload Spreadsheet

## 2.1 Structure of Dataload Spreadsheet

The following table illustrates the structure of the Dataload, user template, spreadsheet. The appropriate way to read this table is to go down the first column, followed by the second column. The columns continue onto the next page then the second column restarts on this page.

| | |
|---|---|
| FMS Vendor ID | FCCN Stipend |
| FMS Contract No. | FCCN Admin |
| DOE ID | HS UPK City Admin WorkComp |
| ACCIS | HS UPK City Admin Insurance |
| Program Number | HS UPK City Admin Facility |
| Contractor | HS UPK City Admin Total |
| Site | CC UPK City Admin WorkComp |
| Site Address | CC UPK City Admin Insurance |
| Site City | CC UPK City Admin Facility |
| Site Zip | CC UPK City Admin Total |
| Borough/Zip | HS City Admin WorkComp |
| City Lease? | HS City Admin Insurance |
| DCC Model | HS City Admin Facility |
| Model | HS City Admin Total |
| Preschool Slots Award | HS CTL City Admin WorkComp |
| Toddler Slots Award | HS CTL City Admin Insurance |
| Infant Slots Award | HS CTL City Admin Facility |
| Preschool Dual Slots | HS CTL City Admin Total |
| Preschool HS Slots | HS Contractor UPK |
| Preschool CC Slots | CC Contractor UPK |
| Preschool Dual Slots UPK | HS City Admin UPK |
| Preschool HS Slots UPK | CC City Admin UPK |
| Preschool CC Slots UPK | Total UPK |
| Total UPK Slots | Combined Head Start |
| GFDC Preschool Award Slots | Contractor Head Start |
| GFDC Toddler Award Slots | City Head Start |
| GFDC Infant Award Slots | Required Non Federal Match |
| RFDC Preschool Award Slots | NFM Contractor UPK HS |

| | |
|---|---|
| RFDC Toddler Award Slots | NFM Contractor HS CTL |
| RFDC Infant Award Slots | NFM City UPK HS |
| Total FCCN Award Slots | NFM City HS CTL |
| Infant | Funded NFM |
| Toddler | NFM Shortfall |
| Preschool | CC GDCIN Contractor Contribution |
| Site Award | CC GDCTD Contractor Contribution |
| City Admin | CC GDCPS Contractor Contribution |
| Total Funding | CC FCCN Contractor Contribution |
| HS Contractor UPK | CC UPK Contractor Contribution |
| CC Contractor UPK | HS/HS CTL/HS UPK Contractor Contribution |
| Total Contractor UPK | Combined Contractor Contribution |
| Head Start | Full Contractor Contribution |
| Head Start CTL | — |
| Child Care | — |

# 3 Master Workbook

## 3.1 Structure

The Master workbook was carefully structured to organize where data would be validated, error reports would be printed, and to provide easy-to-manipulate data for the macros.


### 3.1.1 Data Table

The 'Data Table' spreadsheet is the location of which the new data will be imported. After data has been updated in this spreadsheet, the validation will proceed in the same sheet.

Two buttons were placed near the beginning of the page. One, labeled "Import Data", is used to import the data from the user template. The next button, labeled "Validate Text/Save to ASC/Email Errors", is used to validate the data, export the data to an ASCII file, and email the error report to a specified email list within a further sheet.

### 3.1.2 EmptyCells

The EmptyCells spreadsheet is used to output the accumulated error report. Upon running the validation macro, any errors that come up will be placed in this spreadsheet during run-time.

**Empty Cells Report**

| *Row* | *Column* |
|---|---|
| 95 | 1 |

In the first part of the spreadsheet is the error report specifically for empty cells. If there are any empty cells within the 'Data Table' spreadsheet, the macro will pick them up and output the row and column number of that cell.

| Data Validation Report | | |
|---|---|---|
| *Row* | *Column* | |
| 95 | 1 | Vendor ID |
| 101 | 1 | Vendor ID |
| 102 | 5 | Program Number |

In the next part is the validation report. Five columns in the 'Data Table' spreadsheet are required to undergo a validation test. Any cells within those five columns that do not meet a certain requirement will be displayed in the error report in this spreadsheet with the row and column number. In addition, the name of the column will be shown adjacent to the outputted row and column.

### 3.1.3 Data Table Copy

This spreadsheet is mainly used for testing purposes from the developer side. Whenever a change needs to be made to a macro it is first tested in this spreadsheet before implementing the changes on the primary sheet.

### 3.1.4 DataFile Location

The DataFile Location spreadsheet keeps the location of two folders and one file name being used by its macros.

In the first cell, A1, is the folder location of the user template data. A macro uses this cell to retrieve the folder path and access the user template within. If the location of the user template was changed then this cell will have to be updated in response.

In the second cell, A2, is the temporary name of the user template that needs to be accessed for importing. If any changes are made to the user template file name then this cell will need to be changed as well. The file extension is not needed for the cell.

In the third cell, A3, is the folder location of where the final ASCII file will be saved. This is the file that will be uploaded to the oracle database. If the folder location of the final ASCII file needs to be changed, then simply change this contents of this cell to the new location. Inside this folder is also where the Master workbook is located, but changing the contents of this cell will not change the location of where this Master workbook will be saved.

### 3.1.5 Email Recipients

The email recipients spreadsheet is used by the macro that sends the error report to a list of users. That macro grabs the list of emails within this spreadsheet and emails the final error report to that list.

If any emails have to be removed, updated or added, then refrain from leaving any blank cells between emails. The macro runs through the column and stops at a blank cell. If there are blank cells between emails then the macro will not pick up any additional emails after that blank cell.

# 4 Data Import Macro

## 4.1 Description

Upon changes to the user data template, validation checks must occur. The validation macro is located in the master workbook to assure security. In order for the data to be passed through the macro and validated, a macro was developed to copy the contents of the user template and import the selected data into the master workbook.

## 4.2 How It Works - Technical Details

For ease of use a button was placed at the beginning (top-left most part) of the first sheet, Data Table. This button, labeled Import Data, runs the module named ImportData( ).

This module assigns the active, master, workbook to a variable wbThis to determine the location of where the new data will be imported into. The user data template is opened from the file location specified inside the DataFile Location sheet and set to wbTarget to identify the target workbook from which the data will be imported.

A range is then specified, selected and then copied within the module; this is hardcoded but can be changed to be dynamic. Lastly, wbThis is reopened and the updated data is pasted starting at the specified cell.

# 5 Validation Macro

## 5.1 Description

The main purpose of the Master workbook is to validate the updates from the user data template. Once the new data has been imported the validation process can now begin. In the next section you will find the validation requirements for the new data.

## 5.2 Validation Requirements

| No | Field name | Mandatory | Length | Note |
|---|---|---|---|---|
| 1 | Vendor ID | Yes | 10 characters | Macro should preserve the leading zeros. Must be 10 characters. |
| 2 | Program No. | Yes | 7 characters | Macro should preserve the leading zeros. Must be 7 characters. |
| 3 | ACCIS ID | Yes | 5 characters | Macro should preserve the leading zeros. Must be 5 characters. |
| 4 | City Lease | Optional | 3 characters | Possible values: Yes or No or Empty. |
| 5 | Contract No. | Yes | 11 characters | Macro should preserve the leading zeros. Must be 11 characters. |

### 5.2.1 Extra Notes

**City Lease**

- If the original spreadsheet says Y, macro should translate it to Yes
- If the original spreadsheet says N, macro should translate it to No.
- If the original spreadsheet says 'Yes' or 'No'; macro should copy as such.

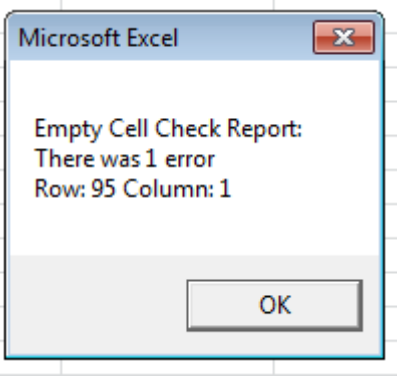
## 5.3 How It Works

To initiate the validation of the imported data, a button was placed in the beginning (top-left most part) of the first sheet, 'Data Table', labeled 'Validate Text/Save to ASC/Email Errors'. To elaborate, this button will run the validation macro that will be discussed here, the conversion macro, and the emailed error report macro, which will both be discussed in later sections.

### 5.3.1 Empty Cells

Upon clicking the button, the macro named FindEmptyCells( ) will execute. Before any computations are executed, though, the sheet 'EmptyCells' is cleared to start a fresh error report.

A for loop is used to loop through the mandatory columns which were mentioned in the previous section. The loop in this module selects any cell that has no value inside of it and fills it with a red background in order to make it easier for locating the errors once the script finishes.




As the loop is running and potentially picking up empty cells, the report in the next sheet 'EmptyCells' gets updated with the specific row and column number of the found issue. Then, at the end of the loop, a msgbox appears with the accumulated error report. The string variable containing the message is sent over to the next macro to append to the data validation report.

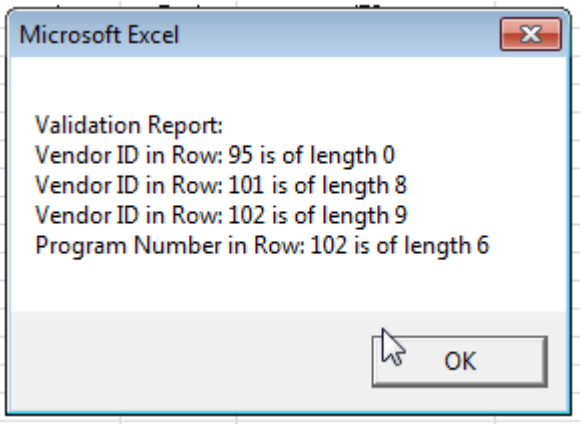
### 5.3.2 Data Validation

At the end of the FindEmptyCells( ) macro, ValidateFields(String), is called; this is the module responsible for validating the imported data based upon

the previous table.

Within this module are five similar for loops, one for each column that needs validating: VendorID, ContractID, ACCIS, Program, and City Lease. Following the provided table of necessary validations, the for loops run chronologically through each of the five columns.

As in the previous FindEmptyCells( ) macro, the errors are instantaneously placed in the sheet 'EmptyCells'. The information provided in the validation report is similarly the row and column number, while also mentioning what the name of the column is.




At the conclusion of the script a msgbox will appear presenting the found errors. The string that contains the potential errors from the validation script is appended to the string containing the error report from the last module to form one error report message. The variable finalMsg, which is holding the two appended error reports, is a publicly declared variable so it can be used by any macro within that module.

# 6 ASCII Conversion and Email Macro

## 6.1 XLS to ASCII Conversion

### 6.1.1 Description

Once new data is imported and validated the next macro runs, CreateCSV ( ), in order to save the data as a pipe-delimited ASCII file.

File Naming Convention: VERP_BUDGET_DATA_YYYYMMDDHHMMSS.asc

#### Pre-Requirements

Before the data is converted, some pre-requirements have to be met.

- Formulas must be removed from all cells. Only the values should be retained.
- All $ symbols should be removed.
- Round $ amounts to two digits after decimal.
- Remove any empty columns
- Remove any filters applied.
- Remove first two rows (one with column numbers and second with merged field headings). The third row with column headers (row 3) shall be retained.

### 6.1.2 How It Works - Technical Details

When this macro is called, initially it tries to open a file that is not there, thus creating a new file by the above file naming convention. The location of where this file will be saved to is located within the DataFile Location sheet.

Since the file is required to be pipe delimited, the delimiter was hard coded into the macro. If the delimiter ever needs to be changed for future purposes, it would need to be changed in the macro itself. If for whatever reason the delimiter needs to be changed frequently, and changing it in the code becomes

non-efficient, a developer could make the macro read the delimiter from a cell within a spreadsheet. Therefore, a user could change the delimiter from one of the sheets, or a new sheet, without having to access the macro.

The last row is set using the 6th column due to potential of problems if row chosen was one that needed to be validated. Two nested for loops run through all rows and columns and output the cell contents into the newly created ASCII file. The pipe-delimiter is attached to the end of the cell content, unless the row equals the last row. The file is then closed and the next macro is called with a parameter of the file name of the file that was just created.

## 6.2 Email Error Report Macro

### 6.2.1 Description

Upon completion of the validation and exportation of data to an ASCII file, the accumulated error report is then sent out, along with the ASCII file, to a list of recipients.

### 6.2.2 How It Works - Technical Details

This macro takes in a string parameter of the file name of the file that was created in the previous macro. The contents of the email are set as variables prior to configuration to allow easier future change.

Next, the SMTP server information and CDO objects are configured according to provided information. Finally, the main options associated with emailing from a macro are initialized to the variables that were set earlier.