# VBA Utilities Reference

## Available Modules:

Funcs: Useful functions

### Funcs

Funcs.**GetRangeAsArr**(rng *As Range*, returnArr *As Variant*[, arrType *As Integer*])

Convert a worksheet range to an array

**Parameters:**

* **rng** – Range object to get data from
* **returnArr** – Variant array to set with return data
* **arrType** – (optional) Integer for how to format return array
  + 0 – Return an array of arrays, with each nested array being a row
  + 1 – Return a 2D array, with dimension 1 as rows and dimension 2 as columns

Funcs.**ConvertEmpty**(val *As Variant*, opt *As Integer*) *As Variant*

Convert empty values based on opt parameter

**Parameters:**

* **val** – Value to convert. If not mpty, this value will be returned
* **opt** – How to handle empty values
  + 0 – Convert to empty
  + 1 – Convert to an empty string (“” or vbNullString)
  + 2 – Convert to 0
  + 3 – Convert to Null

**Returns:** Converted value

Funcs.**TrimEmpty**(arr *As Variant*) *As Variant*

Trim empty values from the end of a 1D array. Start index is retained, and original array is unchanged

**Parameters:**

* **arr** – Array to be trimmed

**Returns:** Trimmed array

Funcs.**GetODBCVersion**() *As String*

Look for an ODBC driver on this PC. Returns the highest version found

**Returns:** String for ODBC driver found

Funcs.**GetSQLConnStr**(database *As String*[, server *As String*) *As String*

Get a SQL connection string. Automatically finds the ODBC driver version

**Parameters:**

* **database** – Target database
* **server** – (optional, default “dotssmsqlprod”) Target server

**Returns:** SQL connection string

**Usage:**

dim conn As New ADODB.Connection

conn.Open Funcs.GetSQLConnStr(“Pavement”)

...

conn.Close

Funcs.**ShellRun**(cmd *As String*) *As String*

Run a shell command and get text back from StdOut

**Parameters:**

* **cmd** – Shell command to execute

**Returns**: Output from StdOut

Funcs.**GetFolder**([title *As String*]) *As String*

Prompt the user to select a folder

**Parameters:**

* **title** – (optional, default “Select a Folder”) Prompt text in file dialog

**Returns**: Path of user selected folder. Empty string if user cancels

Funcs.**GetFile**(

[title *As String*,

{filterName1 *As String*, filterFileTypes1 *As String*}, …]) *As String*

Prompt the suer to select a file

**Parameters:**

* **title** – (optional, default “Select a File”) Prompt text in file dialog
* **{filterName1, filterFileTypes1}** – (optional) Pairs of filter names and file types (e.g. “Images”, “\*.jpg; \*.png”)

**Returns:** Path of user selected file. Empty string if user cancels

## Available Classes:

DataUtils: A pandas-like interface for importing/exporting files

SheetData: Store and interact with the data (non-header) portion of files

HeaderData: Store and interact with the header portion of files

### DataUtils

A pandas-like interface for importing/exporting files such as CSVs and excel files

**Attributes**

DataUtils.**data** *As SheetData* – Container for data portion of file

DataUtils.**header** *As HeaderData* – Container for header portion of file

**Methods**

DataUtils.**InitFromXL**(filePath *As String*[, dataStartRow *As Integer*, sheetIndex *As Integer*, sheetName *As String*, handleEmpty *As Integer*, disableScreenUpdate *As Boolean*)

Initialize the object from an excel file

**Parameters:**

* **filePath** – String for filepath of target file
* **dataStartRow** – (optional) Specify the header row for the data section of the file. If not specified, the code will try to find this automatically
* **sheetIndex** – (optional) Numerical index for which sheet in the workbook to use. Overridden by sheetName
* **sheetName** – (optional) Name of which sheet in the workbook to use. Overrides sheetIndex
* **handleEmpty** – (optional) How to handle empty values as specified in Funcs.ConvertEmpty
* **disableScreenUpdate** – (optional, default True)

DataUtils.**InitFromSheet**(ws *As Worksheet*[, dataStartRow *As Integer*, handleEmpty *As Integer*)

Initialize the object from a worksheet object

**Parameters:**

* **filePath** – String for filepath of target file
* **dataStartRow** – (optional) Specify the header row for the data section of the file. If not specified, the code will try to find this automatically
* **handleEmpty** – (optional) How to handle empty values as specified in Funcs.ConvertEmpty

DataUtils.**GetRunLength**([colIdx *As Integer*, colName *As String*) *As Long*

Get the length of this file if it is a run

**Parameters:**

* **colIdx** – (optional) Specify the column to use as chainage (0-indexed). Defaults to the leftmost column
* **colName** – (optional) Specify the column name to use as chainage. Overrides colIdx

**Returns:** Run length

### SheetData

Store and interact with the data portion of a file

**Attributes**

SheetData.**numCols** *As Integer* – Number of columns in the dataset

SheetData.**numRows** *As Integer* – Numbe of rows in the dataset

**Methods**

SheetData.**InitFromRange**(dataRange *As Range*[, handleEmpty *As Integer*])

Initialize the object from a Range object

**Parameters:**

* **dataRange** – Range object containing the data to be loaded. The first row should be column names
* **handleEmpty** – (optional, default 0) Handle empty values as described in Funcs.ConvertEmpty

SheetData.**SetColumnNames**(columnNames *As Variant*)

Set the column names of the data

**Parameters:**

* **columnNames** – Names of the columns. Non-strings will be converted to strings

SheetData.**AddRow**(row *As Variant*)

Add a row of values to the end of the data set. Can only be called in the column names have been set

**Parameters:**

* **row** – Array of values. Must be the same length as the number of columns

SheetData.**GetRowByIdx**(idx *As Integer*) *As Variant*

Get a row of data by its index (0-indexed)

**Parameters:**

* **idx** – row index

**Returns:** Variant array containing row data

SheetData.**GetColByIdx**(idx *As Integer*) *As Variant*

Get a column of data by its index (0-indexed)

**Parameters:**

* **idx** – row index

**Returns:** Variant array containing column data

SheetData.**GetRowByVal**(val *As Variant*) *As Variant*

Get a row of data by the leftmost value in that row. If this value appears in multiple rows, only the first instance is returned

**Parameters:**

* **val** – value to search for in the leftmost column

**Returns:** Variant array containing row data

SheetData.**GetColByVal**(val *As Variant*) *As Variant*

Get a column of data by its column name. If this value appears in multiple columns, only the first instance is returned

**Parameters:**

* **val** – Column name

**Returns:** Variant array containing column data

SheetData.**GetVal**(col *As Variant*, row *As Variant*[, useColIdx *As Boolean*, useRowIdx *As Boolean*]) *As Variant*

Get a single data point using indices or row/column names

**Parameters:**

* **col** – Column index or name
* **row** – Row index or name
* **useColIdx** – (optional, default True) If true, use column index. If false, use column name
* **useRowIdx** – (optional, default True) If true, use row index. If false, use leftmost row value

**Returns:** Data point

### HeaderData

Store and interact with the header portion of a file

**Attributes**

HeaderData.**numRows**  *As Integer* – Number of rows in the dataset

**Methods**

HeaderData.**InitFromRange**(dataRange *As Range*)

Initialize the object from a Range object

**Parameters:**

* **dataRange** – Range object containing the data to be loaded

HeaderData.**GetVal**(col *As Integer*, row *As Integer*) *As Variant*

Get a single header value by its row/column indices

**Parameters:**

* **col** – Column index (0-indexed)
* **row** – Row index (0-indexed)

**Returns:** Header value

HeaderData.**GetRow**(row *As Variant*[, useIdx *As Boolean*]) *As Variant*

Get a row of header values. Search by index or by the value in the leftmost column. If searching by value, this value is not included in the return. If this value appears in multiple rows, only the first instance is returned

**Parameters:**

* **row** – Row index or leftmost search value
* **useIdx** – (optional, default True) Use a row index if true, otherwise search for the value in the leftmost column

**Returns**: Value or array of values from that row. If searching by value, this value is not included in the return