

---

# Command Reference: ReadTableCellsFromExcel()

Read cells from a Microsoft Excel workbook file and insert into a table

Version 10.27.00, 2014-02-28

The `ReadTableCellsFromExcel()` command reads cells from a Microsoft Excel worksheet and transfers them into a table row. Currently a single table row can receive the output. The command is designed to extract values from named ranges in the Excel file, for example to harvest data from a data-entry form. Use this command in a template to iterate through rows in a table matched with separate Excel files.

TSTool uses the Apache POI software, version 3.9 (<http://poi.apache.org>) to read/write the Excel file and consequently functionality is constrained by the features of that software package.

The following dialog is used to edit the command and illustrates the syntax for the command.

**Edit WriteTableCellsToExcel() Command**

This command writes cells from a table row to specific cell in a Microsoft Excel worksheet.  
The Excel file (workbook) and worksheet must already exist. See the `NewExcelWorkbook()` command.  
Currently only a single table row can be output by matching with the column filter. If necessary use a template that loops over rows.  
The cell locations in Excel can be specified using a named range, or A1-style address.  
It is recommended that the location of the Excel file be specified using a path relative to the working directory.  
The working directory is: C:\Projects\CO-CWCB-CDM-BIP\BIP-DataSheets\TSTool\MasterIPPToDataSheets-Version2

Table ID:  Required - table to write.

Output (workbook) file:

Worksheet:  Optional - worksheet name (default=first sheet).

Column filters to include rows:   Optional - column patterns to include rows (default=include all).

Column to cell address map:   Required - indicate column to cell address mapping (default=none).

Keep file open?:  Optional - keep Excel file open? (default=False).

Command:

ReadTableCellsFromExcel

## ReadTableCellsFromExcel() Command Editor

The command syntax is as follows:

```
ReadTableCellsFromExcel (Parameter=Value,...)
```

## Command Parameters

| Parameter             | Description   | Default   |
|-----------------------|---|---|
| InputFile             | The name of the Excel workbook file (*.xls or *.xlsx) to read, as an absolute path or relative to the command file location.  | None – must be specified.   |
| Worksheet             | The name of the worksheet in the workbook to read.  | Read the first worksheet.   |
| KeepOpen              | Indicate whether to keep the Excel file open (True) or close after creating (False). Keeping the file open will increase performance because later commands will not need to reread the workbook. Make sure to close the file in the last command that writes to the Excel file.  | False   |
| ColumnCellMap         | Indicate how to map table column names to Excel addresses. The format of the parameter is:<br>ColumnName1:ExcelAddress1,<br>ColumnName2:ExcelAddress2, ...<br>where the Excel addresses can be specified using A1 notation, or named range. <b>The column names cannot be repeated.</b>   | None – must be specified.   |
| TableID               | Identifier for table to be updated.   | None – must be specified.   |
| ColumnInclude Filters | Indicate table column names and pattern to use to include rows, in order to match a single row for output. The format of the parameter is:<br>ColumnName1:Pattern1,<br>ColumnName2:Pattern2, ...<br>where patterns can contain * to match a substring. For example, if used with a template to loop through table rows, use to match a unique identifier value in a column. | Include all rows (which will generate an error if the number of rows is not equal to one) |
| IfTableRow NotFound   | Indicate the action if the row is not found for the transfer, one of: <ul style="list-style-type: none"> <li>Append – append a new row to the table</li> <li>Ignore – ignore the data</li> <li>Warn – generate a warning</li> <li>Fail – generate a failure</li> </ul>  | Warn  |