

# Worksheet.ImportData(Object(,), String(), Cell, DataImportProperties)

## Description

Imports data from a rectangular array of objects to cells in the worksheet. The new data will overwrite values and formulas in the target worksheet cells, but existing formatting will be preserved.

### C#

```
public Area ImportData(System.Object[,] data, System.String[] columnNames, Cell cell, DataImportProperties props)
```

### vb.net

```
Public Function ImportData(ByVal data As Object(,), ByVal columnNames As String(), ByVal cell As Cell, ByVal props As DataImportProperties) As Area
```

## Parameters

### *data*

A rectangular array of values to import to the worksheet. The first dimension corresponds to row and the second to column. Thus, an array of data { {"A","X"}, {"B","Y"}, {"C","Z"} } would be inserted into the worksheet as:

A	X
B	Y
C	Z

If you enable [DataImportProperties.Transpose](#), the format will be [row][column], so: { {"A","B","C"}, {"X","Y","Z"} } would be inserted into the worksheet as:

A	X
B	Y
C	Z

### *columnNames*

An array of column names for the imported data. `columnNames` and the column dimension of `data` must contain the same number of elements.

### *startCell*

The cell at which to start entering the imported values.

### *props*

A [DataImportProperties](#) object that contains a set of properties that will determine the behavior of the data import.

## Returns

An [Area](#) object representing the set of cells populated with the imported values.

## Examples

### C#

```
Area importedArea = ws.ImportData(dataArray, fieldNames, cel, importProps);
```

### vb.net

```
Dim importedArea As Area = ws.ImportData(dataArray, fieldNames, cel, importProps)
```