

# Area.ImportData(Object(,), String(), DataImportProperties)

## Description

Imports data from a rectangular array of objects to the specified [Area](#). 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,
DataImportProperties props)
```

### vb.net

```
Public Function ImportData(ByVal data As Object(,), ByVal columnNames As String(),
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*

A string array of column names.

### *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#

```
DataImportProperties importProps = wb.CreateDataImportProperties();  
importProps.Transpose = true;  
Area importedArea = a.ImportData(dataArray, fieldNames, importProps);
```

### vb.net

```
Dim importProps As DataImportProperties = wb.CreateDataImportProperties()  
importProps.Transpose = True  
Dim importedArea As Area = a.ImportData(dataArray, fieldNames, importProps)
```