

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

## Description

Imports data from a two-dimensional 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 properties)
```

**vb.net**

```
Public Function ImportData(ByVal data As Object()), ByVal columnNames As String(),
ByVal properties As DataImportProperties) As Area
```

## Parameters

### ***data***

A two-dimensional array of values to import to the worksheet. By default, the first dimension corresponds to row and the second to column (that is, Object[row][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.

### ***properties***

A [DataImportProperties](#) object that contains a set of properties to use when importing data to the area.

## Returns

An Area object representing the set of cells populated with the imported values.

## Examples

### C#

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

### vb.net

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