

ExcelTemplate.BindRowData(Object(), String(), String, DataBindingProperties)

Description

Sets an array of objects as a data source for a single row in a template.? When binding an array horizontally, you must insert a datamarker for each element you want displayed.

C#

```
public virtual void BindRowData(System.Object[] rowData, System.String[] columnNames, System.String dataSourceName, DataBindingProperties property)
```

vb.net

```
Public Overridable Sub BindRowData(ByVal rowData As Object(), ByVal columnNames As String(), ByVal dataSourceName As String, ByVal [property] As DataBindingProperties)
```

Parameters

rowData

An object to use as the data source.

columnNames

The names of the columns to get from the data source. If the `columnNames` parameter is null, field binding can only be performed by ordinal (for example, `%%=DSN.#1` or `%%=$DSN`). If `columnNames` is specified, both ordinal field binding and named field binding can be used.

dataSourceName

The name of the set of data markers at which to insert the values imported from the data source. `dataSourceName` must be specified, but can be left as null or an empty string if this is the first data source bound AND the data markers in the template use the [short data marker syntax](#) or refer to the datasource by number rather than name. Note: `dataSourceName` does not include a data marker's column name, for example, the `dataSourceName` for `%%=Products.ProductID` is "Products."

property

The [DataBindingProperties](#) object which contains information about how the data should be bound to the template. `property` Must be specified, but the [DataBindingProperties](#) need not be set beforehand. To bind data to a template with the default [DataBindingProperties](#), pass in `ExcelTemplate.CreateDataBindingProperties()` as the `property` value. Otherwise, use the `ExcelTemplate.CreateDataBindingProperties()` method to generate a new [DataBindingProperties](#) object and set the [DataBindingProperties.MaxRows](#), [DataBindingProperties.Transpose](#), and/or [DataBindingProperties.WorksheetName](#) properties for the workbook.

Exceptions

ArgumentNullException

`BindRowData` will throw this exception if `null` (C#) or `Nothing` (VB.NET) is passed to the method.

ArgumentException

Examples

C#

```
//--- Set a data source for the row of data markers
//--- %=Address.Street, %=Address.City, and
//--- %=Address.State
string[] addressvalues = {"1330 Beacon St.", "Brookline", "MA"};
string[] addressnames = {"Street", "City", "State"};
xlt.BindRowData(addressvalues,
                addressnames,
                "Address",
                xlt.CreateDataBindingProperties());
```

vb.net

```
'--- Set a data source for the row of data markers
'--- %=Address.Street, %=Address.City, and
'--- %=Address.State
Dim addressvalues As String() = {"1330 Beacon St.", "Brookline", "MA"}
Dim addressnames As String() = {"Street", "City", "State"}
xlt.BindRowData(addressvalues, _
                addressnames, _
                "Address" _
                xlt.CreateDataBindingProperties())
```