

Using an Array as a Data Source

The following sample generates an Excel spreadsheet from a template using arrays as data sources.

Code sample: Using an Array as the Data Source

[C#] [VB.NET]

The first data marker in the template is `*%=%$SimpleArray*`. `*%=%$*` indicates that the data source is either a simple variable or a 1-dimensional array that is a data source for a single column.



Data Marker Format for 1-D Arrays

For a single column data marker that will be bound to a 1-dimensional array data source by the method [BindColumnData](#), use the data marker format `*%=%$DataMarkerName*`. For a set of data markers with multiple fields that will be bound to a 1-dimensional array data source by the method [BindRowData](#), use the data marker format `*%=%DataMarkerName.Field*` (without a `*$*`). For more information, see [Creating Data Markers](#).

In `ArrayDataSource.aspx.cs`, the [BindColumnData](#) method is called to set the data source for `*%=%$SimpleArray*` to a one-dimensional array:

```
string[] onedim = {"SoftArtisans", "OfficeWriter", "ExcelTemplate"};x
lt.BindColumnData(onedim, "SimpleArray",xlt.CreateDataBindingProperties());
```

```
Dim onedim As String() = {"SoftArtisans", "OfficeWriter", "ExcelTemplate"}
xlt.BindColumnData(onedim, "SimpleArray", xlt.CreateDataBindingProperties())
```

[ExcelTemplate.BindColumnData](#) sets a data source for a template column to a 1-dimensional array of objects. The method's first parameter, `onedim`, is the data source array. The second parameter, `"SimpleArray"`, is the name of the template data marker to which the data source binds. The third parameter, `xlt.CreateDataBindingProperties()`, is a [DataBindingProperties](#) object that contains values that change how the data are bound to the template; in this case, the default property values are used.

The method [BindData](#) binds the data markers `*%=%TwoDimArray.#1*`, `*%=%TwoDimArray.#2*`, and `*%=%TwoDimArray.#3*` to a 2-dimensional string array:

```
string[][] twodim = {
    new string[]{"Nancy", "Davolio", "Sales Manager"},
    new string[]{"Michael", "Suyama", "HR Representative"},
    new string[]{"Adrian", "King", "IS Support"}
};
string[] names = {"FirstName", "LastName", "Position"};
xlt.BindData(twodim, names,"TwoDimArray", xlt.CreateDataBindingProperties());
```

```
Dim twodim()() As String = New String()() { _
    New String(){"Nancy", "Davolio", "Sales Manager"}, _
    New String(){"Michael", "Suyama", "HR Representative"}, _
    New String(){"Adrian", "King", "IS Support"} _
}
Dim names As String() = {"FirstName", "LastName", "Position"}
xlt.BindData(twodim, names, "TwoDimArray", xlt.CreateDataBindingProperties())
```

`BindData`'s first parameter is the 2-dimensional array of strings to use as the data source. The second parameter specifies an array of column names.

The third parameter - "TwoDimArray" - is the name of the template data marker to which the data source binds. The fourth parameter - xlt.CreateDataBindingProperties() - is a [DataBindingProperties](#) object that contains values that change how the data are bound to the template; in this case, the default property values are used.

There are three "TwoDimArray" data markers; their fields are specified by ordinal (#1, #2, and #3). The columns of values in the data source bind to the data markers by order. That is, the first column of values will populate **%%=TwoDimArray.#1**, the second will populate **%%=TwoDimArray.#2**, and the third **%%=TwoDimArray.#3**.

[BindRowData](#) sets a data source for the row of data markers **%%=Address.Street**, **%%=Address.City**, and **%%=Address.State** to a 1-dimensional array:

```
string[] addressvalues = {"3 Brook St.", "Watertown", "MA"};
string[] addressnames = {"Street", "City", "State"};
xlt.BindRowData(addressvalues,addressnames,"Address",xlt.CreateDataBindingProperties()
);
```

```
Dim addressvalues As String() = {"3 Brook St.", "Watertown", "MA"}
Dim addressnames As String() = {"Street", "City", "State"}
xlt.BindRowData(addressvalues, addressnames, "Address",
xlt.CreateDataBindingProperties())
```

The first parameter of BindRowData, addressvalues, specifies the array of values to bind to the data markers. The second parameter, addressnames, is the array of data marker field names. The values in the data source array will bind to the data marker fields by field name. The third parameter, "Address", is the name of the data marker to which ExcelTemplate binds the data. The fourth parameter, xlt.CreateDataBindingProperties(), is a [DataBindingProperties](#) object that contains values that change how the data are bound to the template; in this case, the default property values are used.

Data Marker	Code
%%=\$SimpleArray	<div><pre>string[] onedim = {"SoftArtisans", "OfficeWriter", "ExcelTemplate"}; xlt.BindColumnData(onedim, "SimpleArray", xlt.CreateDataBindingProperties());</pre></div> <div><pre>Dim onedim As String() = {"SoftArtisans", _ "OfficeWriter", _ "ExcelTemplate"} xlt.BindColumnData(onedim, _ "SimpleArray", _ xlt.CreateDataBindingProperties())</pre></div>

%%=TwoDimArray.#1

%%=TwoDimArray.#2

%%=TwoDimArray.#3

```
string[][] twodim = {
    new string[] { "Nancy", "Davolio",
        "Sales Manager" },
    new string[] { "Michael", "Suyama",
        "HR Representative" },
    new string[] { "Adrian",
        "King",
        "IS Support" }
};
string[] names = { "FirstName",
    "LastName", "Position" };
xlt.BindData(twodim,
    names,
    "TwoDimArray",

xlt.CreateDataBindingProperties();
```

```
Dim twodim()() As String = New
String()() { _
    New String() { "Nancy", "Davolio",
        "Sales Manager" }, _
    New String() { "Michael", "Suyama",
        "HR Representative" }, _
    New String() { "Adrian", _
        "King", _
        "IS Support" }
}
Dim names As String() = { "FirstName",
    "LastName", "Position" }
xlt.BindData(twodim, _
    names, _
    "TwoDimArray", _

xlt.CreateDataBindingProperties()
```

%%=Address.Street

```
string[] addressvalues = {"3 Brook  
St.",  
    "Watertown",  
    "MA"};  
string[] addressnames = {"Street",  
    "City", "State"};  
xlt.BindRowData(addressvalues,  
    addressnames,  
    "Address",  
  
xlt.CreateDataBindingProperties());
```

```
Dim addressvalues As String() = {"3  
Brook St.", _  
    "Watertown", _  
    "MA"}  
Dim addressnames As String() =  
{"Street", "City", "State"}  
xlt.BindRowData(addressvalues, _  
    addressnames, _  
    "Address", _  
    xlt.CreateDataBindingProperties())
```