

Arrays and Collections as Data Sources

Intro

Use a variety of .NET data types to populate ExcelTemplate including string[], ArrayList, and Hashtable.

Methods used:

- **BindData** - For using 2-dimensional, grid-like data stored in an array. Arrays can be arranged in [row,col] (normal) or [col,row] (transposed) fashions.
- **BindColumnData** - For use with a 1-dimensional array, collection, or dictionary. Data is filled in the column.
- **BindRowData** - For use with a 1-dimensional array, collection, or dictionary. Populates with only a single row of the data source. There must be a data marker for every column you want to be shown.

Code

[illegible]

```

// When binding to two-dimensional arrays, the second parameter
// to BindData (dataMarkerName) must be a String[] array.
// The values in that array correspond to the data marker names:
// %=TwoDimArray.FirstName
// %=TwoDimArray.LastName
// %=TwoDimArray.Position

string[] names = { "FirstName", "LastName", "Position" };
xlt.BindData(twoDimNormal, names, "TwoDimArray", basicBindingProperties);

// %=TwoDimTrans.FirstName
// %=TwoDimTrans.LastName
// %=TwoDimTrans.Position

string[,] twoDimTranspose = {{"Nancy", "Michael", "Adrian"},
                              {"Davolio", "Suyama", "King"},
                              {"Sales Manager", "HR Representative", "IS
Support"}}};
DataBindingProperties bindTransposed = xlt.CreateDataBindingProperties();
bindTransposed.Transpose = true;
xlt.BindData(twoDimTranspose, names, "TwoDimTrans", bindTransposed);

// When binding to an IDictionary collection, the "Key" is
// the Data Marker Name.
// The following entries bind to the data marker names:
// %=HashTable.FirstName
// %=HashTable.LastName
// %=HashTable.Position

IDictionary hashTable = new Hashtable();
hashTable.Add("FirstName", "Nancy");
hashTable.Add("LastName", "Davolio");
hashTable.Add("Position", "Sales Manager");

xlt.BindRowData(hashTable, "HashTable", basicBindingProperties);

// %=Address.Street
// %=Address.City
// %=Address.State

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

// Process the template to populate it with the Data Source data
xlt.Process();

xlt.Save(@"..\..\ExcelOutputFiles\ArrayDataSource_output.xlsx");
xlt.Save(@"..\..\ExcelOutputFiles\ArrayDataSource_output.xlsx");

```

```
} }
```

Downloads

- Template:[ArrayDataSourceTemplate.xlsx](#)
- Output:[ArrayDataSource_output.xlsx](#)