

# ExcelTemplate.BindData(System.Data.DataSet, String, DataBindingProperties)

## Description

Sets an ADO.NET DataSet as a data source to bind to template data markers.

### C#

```
public void BindData(System.Data.DataSet data, System.String dataSourceName, DataBindingProperties property)
```

### vb.net

```
Public Sub BindData(ByVal data As System.Data.DataSet, ByVal dataSourceName As String, ByVal [property] As DataBindingProperties)
```

## Parameters

### *data*

The DataSet to use as the data source.

### *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*

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

### *SARuntimeException*

`BindData` will throw this exception if the data source contains more rows than the worksheet can hold.

If there is more than one data marker referring to a data source and the data source is forward only, the exception will be thrown only if the source is larger than all bindings can hold.

## Remarks

You can set several data sources for a single template. Use the following methods to set template data sources: [BindCellData](#), [BindColumnData](#),

## Examples

### C#

```
ExcelTemplate xlt = new ExcelTemplate();

OleDbConnection Conn = new OleDbConnection();
DataSet OrdersDs = null;
try
{
    Conn.ConnectionString = Application["connstring"].ToString();

    ///--- SQL Query for orders
    string OrdersSQL =
        "SELECT Orders.OrderID, Customers.CompanyName As Customer, " +
        "Orders.OrderDate, " +
        "([Order Details].UnitPrice * [Order Details].Quantity) " +
        "As [OrderTotal] " +
        "FROM Orders, [Order Details], Customers " +
        "WHERE Orders.OrderID=[Order Details].OrderID AND " +
        "Orders.CustomerID=Customers.CustomerID AND Orders.EmployeeID=?";
    OleDbCommand CmdOrders = new OleDbCommand(OrdersSQL, Conn);
    CmdOrders.Parameters.Add("@EmployeeID", EmployeeId);
    OleDbDataAdapter AdptSales = new OleDbDataAdapter(CmdOrders);
    OrdersDs = new DataSet();
    AdptSales.Fill(OrdersDs, "Orders");
}
xlt.BindData(OrdersDs,
    "Orders",
    xlt.CreateDataImportProperties());
```

```
Dim xlt As New ExcelTemplate()
Dim Conn As New OleDbConnection()
Dim OrdersDs As DataSet = Nothing
Try
    Conn.ConnectionString = Application("connstring").ToString()

    '--- SQL Query for orders
    Dim OrdersSQL As String = _
        "SELECT Orders.OrderID, Customers.CompanyName As Customer, " & _
        "Orders.OrderDate, " & _
        "([Order Details].UnitPrice * [Order Details].Quantity) " & _
        "As [OrderTotal] " & _
        "FROM Orders, [Order Details], Customers" & _
        "WHERE Orders.OrderID=[Order Details].OrderID AND " & _
        "Orders.CustomerID=Customers.CustomerID AND Orders.EmployeeID=?"
    Dim CmdOrders As New OleDbCommand(OrdersSQL, Conn)
    CmdOrders.Parameters.Add("@EmployeeID", EmployeeId)
    Dim AdptSales As New OleDbDataAdapter(CmdOrders)
    OrdersDs = New DataSet()
    AdptSales.Fill(OrdersDs, "Orders")
End Try
xlt.BindData(OrdersDs, _
    "Orders", _
    xlt.CreateDataImportProperties())
```