

Move DataRow in a DataTable

```
////////////////////////////////////  
////////////////////////////////////  
//  
// This File is Part of the CallButler Open Source PBX  
// (http://www.codeplex.com/callbutler  
//  
// Copyright (c) 2005-2008, Jim Heising  
// All rights reserved.  
//  
// Redistribution and use in source and binary forms, with or  
// without modification,  
// are permitted provided that the following conditions are  
// met:  
//  
// * Redistributions of source code must retain the above  
// copyright notice,  
// this list of conditions and the following disclaimer.  
//  
// * Redistributions in binary form must reproduce the above  
// copyright notice,  
// this list of conditions and the following disclaimer in the  
// documentation and/or  
// other materials provided with the distribution.  
//  
// * Neither the name of Jim Heising nor the names of its  
// contributors may be  
// used to endorse or promote products derived from this  
// software without specific prior  
// written permission.  
//  
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND  
// CONTRIBUTORS "AS IS" AND  
// ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
```

```
LIMITED TO, THE IMPLIED
// WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT,
// INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR
// PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY,
// WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE)
// ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE
// POSSIBILITY OF SUCH DAMAGE.
```

```
//
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
```

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data;
```

```
namespace WOSI.Utilities
{
    public class DataUtils
    {
        public static DataRow CreateRowCopy(DataTable newParentTable,
        DataRow dataRow)
        {
            DataRow newRow = newParentTable.NewRow();

            for (int index = 0; index < dataRow.Table.Columns.Count;
            index++) { newRow[index] = dataRow[index]; } return newRow; }
        public static void MoveDataRowUp(DataRow dataRow) { DataTable
        parentTable    =    dataRow.Table;    int    rowIndex    =
```

```
parentTable.Rows.IndexOf(dataRow); if (rowIndex > 0)
{
DataRow newRow = parentTable.NewRow();

for (int index = 0; index < dataRow.ItemArray.Length; index++)
{    newRow[index]    =    dataRow[index];    }
parentTable.Rows.Remove(dataRow);
parentTable.Rows.InsertAt(newRow, rowIndex - 1); dataRow =
newRow; } } public static void MoveDataRowDown(DataRow
dataRow) { DataTable parentTable = dataRow.Table; int rowIndex
= parentTable.Rows.IndexOf(dataRow); if (rowIndex <
parentTable.Rows.Count - 1) { DataRow newRow =
parentTable.NewRow(); for (int index = 0; index <
dataRow.ItemArray.Length; index++) { newRow[index] =
dataRow[index];    }    parentTable.Rows.Remove(dataRow);
parentTable.Rows.InsertAt(newRow, rowIndex + 1); dataRow =
newRow; } } } } [/csharp]
```