

```
using System;
using System.Collections;
using System.Data;
using System.Web.UI;
```

Hierarchical DataSet.cs

```
//-----
//
//                                     HierarchicalDataSet.cs
//
//      This class translates a DataSet into IHierarchicalDataSource
//      that can be used to bind Hierarchical data to a TreeView
//-----
//
//                                     Modification Control Log
//
//      Date      By      Description
//-----
//      01-14-14  JV      Initial creation of program
//-----

namespace Reimbursements.Utilities
{
    /// <summary>
    /// A class that translates a DataSet into IHierarchicalDataSource that can be used to bind
    /// Hierarchical data to a TreeView
    /// </summary>
    public class HierarchicalDataSet : IHierarchicalDataSource
    {
        #region Member Variables

        // Local Variables
        DataSet _DataSet;
        string _IdColumnName;
        string _ParentIdColumnName;

        #endregion

        #region Constructors

        /// <summary>
        /// The constructor of the class
        /// </summary>
        /// <param name="dataSet">The dataset that contains the data</param>
        /// <param name="idColumnName">The Primary key column name</param>
        /// <param name="parentidColumnName">The Parent Primary key column name that identifies the Parent-
        Child relationship</param>
        public HierarchicalDataSet(DataSet dataSet, string idColumnName, string parentIdColumnName)
        {
            _DataSet = dataSet;
            _IdColumnName = idColumnName;
            _ParentIdColumnName = parentIdColumnName;
        }

        #endregion

        #region Enums

        #endregion

        #region Properties

        #endregion
    }
}
```

```

#region Event Handlers

// Event Handlers
public event EventHandler DataSourceChanged;

#endregion

#region Methods

public HierarchicalDataSourceView GetHierarchicalView(string viewPath)
{
    return new DataSourceView(this, viewPath);
}

#region Supporting Methods

string GetChildrenViewPath(string viewPath, DataRowView row)
{
    return viewPath + "\\" + row[_IdColumnName].ToString();
}

DataRowView GetParentRow(DataRowView row)
{
    _DataSet.Tables[0].DefaultView.RowFilter = String.Format("{0} = {1}", _IdColumnName, row
[_ParentIdColumnName].ToString());
    DataRowView parentRow = _DataSet.Tables[0].DefaultView[0];
    _DataSet.Tables[0].DefaultView.RowFilter = "";
    return parentRow;
}

string GetParentViewPath(string viewPath)
{
    return viewPath.Substring(0, viewPath.LastIndexOf("\\"));
}

bool HasChildren(DataRowView row)
{
    _DataSet.Tables[0].DefaultView.RowFilter = String.Format("{0} = {1}", _ParentIdColumnName, row
[_IdColumnName]);
    bool hasChildren = _DataSet.Tables[0].DefaultView.Count > 0;
    _DataSet.Tables[0].DefaultView.RowFilter = "";
    return hasChildren;
}

#endregion

#endregion

#region Private Classes

/// <summary>
/// Private classes that implement further interfaces
/// </summary>
class DataSourceView : HierarchicalDataSourceView
{
    #region Member Variables

    // Local Variables
    HierarchicalDataSet _HierDataSet;

```



```
        string _ViewPath;

    #endregion

    #region Constructors

    public DataSourceView(HierarchicalDataSet hierDataSet, string viewPath)
    {
        _HierDataSet = hierDataSet;
        _ViewPath = viewPath;
    }

    #endregion

    #region Enums

    #endregion

    #region Properties

    #endregion

    #region Methods

    public override IHierarchicalEnumerable Select()
    {
        return new HierarchicalEnumerable(_HierDataSet, _ViewPath);
    }

    #endregion
}

/// <summary>
/// Private classes that implement further interfaces
/// </summary>
class HierarchicalEnumerable : IHierarchicalEnumerable
{
    #region Member Variables

    // Local Variables
    HierarchicalDataSet _HierDataSet;
    string _ViewPath;

    #endregion

    #region Constructors

    public HierarchicalEnumerable(HierarchicalDataSet hierDataSet, string viewPath)
    {
        _HierDataSet = hierDataSet;
        _ViewPath = viewPath;
    }

    #endregion

    #region Enums

    #endregion
```

```
#region Properties

#endregion

#region Methods

public IEnumerator GetEnumerator()
{
    if (_ViewPath == "")
    {
        _HierDataSet._DataSet.Tables[0].DefaultView.RowFilter = String.Format("{0} is null",
_HierDataSet._ParentIdColumnName);
    }
    else
    {
        string lastID = _ViewPath.Substring(_ViewPath.LastIndexOf("\\") + 1);
        _HierDataSet._DataSet.Tables[0].DefaultView.RowFilter = String.Format("{0} = {1}",
_HierDataSet._ParentIdColumnName, lastID);
    }

    IEnumerator i = _HierDataSet._DataSet.Tables[0].DefaultView.GetEnumerator();
    _HierDataSet._DataSet.Tables[0].DefaultView.RowFilter = "";
    return i;
}

public IHierarchyData GetHierarchyData(object enumeratedItem)
{
    DataRowView row = (DataRowView)enumeratedItem;
    return new HierarchyData(_HierDataSet, _ViewPath, row);
}

#endregion

}

/// <summary>
/// Private classes that implement further interfaces
/// </summary>
class HierarchyData : IHierarchyData
{
    #region Member Variables

    // Local Variables
    HierarchicalDataSet _HierDataSet;
    DataRowView _Row;
    string _ViewPath;

    #endregion

    #region Constructors

    public HierarchyData(HierarchicalDataSet hierDataSet, string viewPath, DataRowView row)
    {
        _HierDataSet = hierDataSet;
        _ViewPath = viewPath;
        _Row = row;
    }

    #endregion

    #region Enums
```



```
#endregion

#region Properties

public bool HasChildren
{
    get
    {
        return _HierDataSet.HasChildren(_Row);
    }
}

public object Item
{
    get
    {
        return _Row;
    }
}

public string Path
{
    get
    {
        return _ViewPath;
    }
}

public string Type
{
    get
    {
        return typeof(DataRowView).ToString();
    }
}

#endregion

#region Methods

public IHierarchicalEnumerable GetChildren()
{
    return new HierarchicalEnumerable(_HierDataSet, _HierDataSet.GetChildrenViewPath(_ViewPath,
_Row));
}

public IHierarchyData GetParent()
{
    return new HierarchyData(_HierDataSet, _HierDataSet.GetParentViewPath(_ViewPath),
_HierDataSet.GetParentRow(_Row));
}

#endregion
}

#endregion
}
```