

NetAdvantage for ASP.NET グリッド機能チートシート

このガイドでは、WebDataGrid™ コントロールと WebHierarchicalDataGrid™ コントロールを機能ごとに比較しています。以下の機能をクリックするとコード スニペット セクションへジャンプします。

Notes	列の移動	DataBinding: AccessDataSource	行集計
WebHierarchicalDataGrid 子バンドへ動作を追加	列: 固定	DataBinding: SqlDataSource	選択: セル
	列: サイズ変更	DataBinding: ObjectDataSource	選択: 列
コード スニペット	列: バインドされていない列	DataBinding: LinqDataSource	選択: 行
アクティブ化	列: チェックボックス列	編集: 行の追加	テンプレート化: 列テンプレート
Ajax: イベント	列: 行集計	編集: 行の削除	テンプレート化: 空行
Ajax: ロード オン デマンド(自動)	CRUD: 自動	編集: 埋め込み (組み込み) エディター	グリッド イベント
Ajax: ロード オン デマンド (手動)	CRUD: 手動	- / エディター プロバイダー	各動作のイベント
Ajax: 読み込みインジケーター	DataBinding: HeirarchicalDataSource	Excel エクスポート	改訂履歴
Ajax: バーチャル スクロール	DataBinding: DataSet	PDF エクスポート	
自動的ロードオンデマンド	DataBinding: DataTable	フィルター	
列: 列の非表示	DataBinding: IEnumerable	行番号	

注意:

- ScriptManager コンポーネントのインスタンスはすべてのインスタンスが必要です。
- 各コントロール機能の比較は、WebDataGrid や WebHierarchicalDataGrid へ移行しようとお考えのお客様に最適なリソースです。

- WebHierarchicalDataGrid の列の「同じ」は手順が同じという意味であり、コードがすべて同じであるとは限りません。グリッドへのアクセスが多少異なる場合がありますが、基本的なアプローチは同じです。

WebHierarchicalDataGrid 子バンドへ動作を追加

親グリッドで有効にした動作を WebHierarchicalDataGrid インスタンスの子バンドへ公開する最も簡単な方法は、動作の EnableInheritance プロパティを最上位のグリッドで true に設定することです。動作を継承させたくない場合も子バンドで特定の動作を有効にする方法が多数あります。

WebHierarchicalDataGrid コントロールの親グリッドへ動作を追加するための手順は、このガイドの動作に関して WebDataGrid コントロールで記述したものと同様です。ただし、継承していない子バンドで動作を有効にするには、デザイナーでマークアップまたはコードでバンドに直接動作を追加する必要があります。

以下のコードスニペットは、ColumnMoving 動作を子バンドへ明示的に追加する方法を示します。この方法ですべてのグリッド動作を追加することができます。

デザイナーの場合:

1. 子バンドで動作を有効にする

- 1.1. [スマート タグ] を開きます
- 1.2. [バンドの編集] を選択します
- 1.3. バンドを選択します
- 1.4. 動作セクションを展開します
- 1.5. **ColumnMoving** をチェックします
- 1.6. ダイアログ ウィンドウで設定します

ASPX の場合:

```

<Bands>
  <ig:Band ...>
    <Columns>
      ...
    </Columns>
    <Behaviors>
      <ig:ColumnMoving />
    </Behaviors>
  </ig:Band>
</Bands>

```

コード ビハインドの場合

```

using Infragistics.Web.UI.GridControls;
...
protected void WebHierarchicalDataGrid1_InitializeBand(object sender, BandEventArgs e)
{
    e.Band.Behaviors.Add(new ColumnMoving());
}

```

コード スニペット

アクティブ化

WebDataGrid	WebHierarchicalDataGrid
1. アクティブ化機能の有効化 1.1. [スマート タグ] を開きます	1. アクティブ化機能の有効化 1.1. [スマート タグ] を開きます

- 1.2. [動作の編集] を選択します
- 1.3. [アクティブ化] をチェックします
- 1.4. ダイアログ ウィンドウで設定します

Appearance	
ActiveRowSelectorCssClass	
Behavior	
ActivationClientEvents	
ActiveCellChanged	
ActiveCellChanging	
Initialize	
Enabled	True
Misc	
ActiveCellCssClass	
ActiveColumnCssClass	
ActiveRowCssClass	
ActiveRowSelectorImageCssClass	
AutoPostBackFlags	
ActiveCellChanged	False

クライアントでアクティブ セルを設定する

```
function setActiveCell() {
    var grid = $find("<% = this.WebDataGrid1.ClientID %>");
    var behaviors = grid.get_behaviors();
    var activation = behaviors.get_activation();

    var rows = grid.get_rows();
    var row = rows.get_row(0);
    var cell = row.get_cell(1);
```

- 1.2. [動作の編集] を選択します
- 1.3. [アクティブ化] をチェックします
- 1.4. ダイアログ ウィンドウで設定します

Appearance	
ActiveRowSelectorCssClass	
Behavior	
ActivationClientEvents	
ActiveCellChanged	
ActiveCellChanging	
Initialize	
Enabled	True
Misc	
ActiveCellCssClass	
ActiveColumnCssClass	
ActiveRowCssClass	
ActiveRowSelectorImageCssClass	
AutoPostBackFlags	
ActiveCellChanged	False

クライアントでアクティブ セルを設定する - 親レベル

```
function setActiveCellTopLevel() {

    var grid = $find("WebHierarchicalDataGrid1").get_gridView();
    var behaviors = grid.get_behaviors();
    var activation = behaviors.get_activation();

    var rows = grid.get_rows();
    var row = rows.get_row(0);
    var cell = row.get_cell(1);
```

```
activation.set_activeCell(cell);
}
```

クライアントでアクティブセルを取得する

```
function getGetCell() {

    var grid = $find("<%= this.WebDataGrid1.ClientID %>");
    var behaviors = grid.get_behaviors();
    var activation = behaviors.get_activation();

    return activation.get_activeCell();
}
```

詳細

- [WebDataGrid アクティブ化](#)

```
activation.set_activeCell(cell);
}
```

クライアントでアクティブセルを設定する - 親レベル

```
function setActiveCellRowIsland() {

    var grid = $find("WebHierarchicalDataGrid1");
    var parentGrid = grid.get_gridView();

    if (parentGrid != null) {

        var rows = parentGrid.get_rows();
        var row = rows.get_row(0);
        var cell = row.get_cell(2);

        var childGrid = row.get_rowIslands(0)[0];

        if (childGrid != null) {

            var behaviors = childGrid.get_behaviors();
            var activation = behaviors.get_activation()

            var cRows = childGrid.get_rows();
            var cRow = cRows.get_row(0);
            var cCell = cRow.get_cell(0);
```

```

        activation.set_activeCell(cCell);
    }
}
}

```

クライアントでアクティブ セルを取得する - 親レベル

```

function getActiveCellTopLevel() {

    var grid = $find("WebHierarchicalDataGrid1").get_gridView();
    var behaviors = grid.get_behaviors();
    var activation = behaviors.get_activation();

    return activation.get_activeCell();
}

```

クライアントでアクティブ セルを取得する - 行アイランド (子行) レベル

```

function getActiveCellRowIsland() {

    var grid = $find("WebHierarchicalDataGrid1");
    var parentGrid = grid.get_gridView();
    var activeCell;

    if (parentGrid != null) {

```

	<pre> var rows = parentGrid.get_rows(); var row = rows.get_row(0); var childGrid = row.get_rowIslands(0)[0]; if (childGrid != null) { var behaviors = childGrid.get_behaviors(); var activation = behaviors.get_activation() activeCell = activation.get_activeCell(); } } return activeCell; } </pre> <p>詳細</p> <ul style="list-style-type: none"> • WebHierarchicalDataGrid アクティブ化
--	---

Ajax: イベント

WebDataGrid	WebHierarchicalDataGrid
1. Ajax イベントにクライアント ハンドラーを作成します 1.1. プロパティ ウィンドウを開きます	← 同じ

1.2. クライアント イベント グループを展開します

1.3. AJAXResponse および AJAXResponseError にハンドラーを作成します

1.3.1. ドロップ ダウンでクリックしてハンドラーを生成し、関数名を割り当てます

以下の例は関数をロックする方法です:

```
function onAjaxResponseError(sender, eventArgs)
```

```
{
```

```
    //成功した Ajax 要求の応答として
```

```
    //クライアント上で何か実行します
```

```
}
```

```
function onAjaxResponse(sender, eventArgs) {
```

```
    //成功した Ajax 要求の応答として
```

```
    //クライアント上で何か実行します
```

```
}
```

Ajax: ロード オン デマンド(自動)

WebDataGrid	WebHierarchicalDataGrid
実装なし	注: 基本的な設定については、自動ロードオンデマンド セクションを参照してください

	<p>グリッドの EnableAjax プロパティを True (デフォルト値) に設定した場合、データ インタラクションは非同期になります。このプロパティが False の場合、データ操作は全ページのポストバックになります。</p>
--	---

Ajax: ロード オン デマンド (手動)

WebDataGrid	WebHierarchicalDataGrid
実装なし	<p>ロード オン デマンドは、先に Page_Load で最上位のデータソースへバインドし、次に RowIslandsPopulating イベントハンドラ内で子レベルのデータへバインドします。</p> <pre> protected void Page_Load(object sender, EventArgs e) { this.WebHierarchicalDataGrid1.DataSource = this._productRepository.GetAll(); } protected void WebHierarchicalDataGrid1_RowIslandsPopulating(object sender, Infragistics.Web.UI.GridControls.ContainerRowCancelEventArgs e) { e.Cancel = true; ContainerGrid child = new ContainerGrid(); </pre>

	<pre> //列や動作などをカスタマイズして //コンテナー グリッドを詳細設定します child.InitializeRow += new InitializeRowEventHandler(child_InitializeRow); e.Row.RowIslands.Add(child); int productID = Convert.ToInt32(e.Row.DataKey[0]); child.DataSource = this._supplierRepository.GetByProductID(productID); child.DataBind(); } </pre>
--	--

Ajax: 読み込みインジケータ

WebDataGrid	WebHierarchicalDataGrid
<p>1. AjaxIndicator プロパティ グループを設定します</p> <p>1.1. プロパティ ウィンドウでグリッドのプロパティを開きます</p> <p>1.2. AjaxIndicator セクションを展開します</p> <p>1.2.1. ImageUrl を読み込む画像のパスを設定します</p> <p>1.2.2. Location をインジケータのページの位置と設定します</p> <p>1.2.3. 場所がグリッドまたはページに関連付けられているかどうかを決定するために RelativeToControl を設定します</p>	<p>← 同じ</p>

AjaxIndicator	(Has Data)
AltText	Async post
BlockArea	NotSet
BlockCssClass	
CssClass	
Enabled	NotSet
FadeInDuration	
FadeInEquationType	EaseInOut
FadeOutDuration	
FadeOutEquationType	EaseInOut
ImageUrl	~/ajax-loader.gif
Location	MiddleCenter
OffsetLeft	
OffsetTop	
RelativeToControl	True
Text	

注: 読み込みインジケータが正しく動作するには、グリッドの EnableAjax プロパティを True に設定する必要があります。

Ajax: バーチャル スクロール

WebDataGrid	WebHierarchicalDataGrid
<p>1. バーチャル スクロールを有効にする</p> <p>1.1. [スマート タグ] を開きます</p> <p>1.2. [動作の編集] を選択します</p> <p>1.3. バーチャル スクロール動作をチェックします</p> <p>1.4. ダイアログ ウィンドウで設定します</p>	<p>実装なし</p>

<ul style="list-style-type: none"> Behavior <ul style="list-style-type: none"> AverageRowHeight 20 DataFetchDelay 500 Enabled True RowCacheFactor 3 ScrollingMode Virtual ThresholdFactor 0.5 TooltipCssClass TooltipVisibility NotSet VirtualScrollingClientEvents <ul style="list-style-type: none"> Format Tool Tip Initialize MoreRowsReceived MoreRowsRequesting 	
---	--

自動的ロードオンデマンド

WebDataGrid	WebHierarchicalDataGrid
実装なし	<p>InitialDataBindDepth プロパティを最初の読み込みで必要なデータレベルに設定します。デフォルトはルート データのみを読み込む 0 です。-1 の値はロードオンデマンドを無効にし、すべてのデータ レベルを読み込みます。</p> <p>注: Ajax の手順については Ajax: ロードオンデマンド(自動) セクションを参照してください。</p>

列: 列の非表示

WebDataGrid	WebHierarchicalDataGrid
<p>1. 列を非表示にする設定</p> <p>1.1. [スマート タグ] を開いて、[列の編集] を選択します</p>	← 親グリッドも同様

<p>1.2. 選択フィールドから列を選択します</p> <p>1.3. Hidden = True に設定します。</p>	<p>子バンドに動作を追加するには、『WebHierarchicalDataGrid 子バンドに動作を追加する』を参照してください。</p>
--	--

列の移動

WebDataGrid	WebHierarchicalDataGrid																																
<p>1. 列固定動作を有効にする</p> <p>1.1. [スマート タグ] を開きます</p> <p>1.2. [動作の編集] を選択します</p> <p>1.3. 列の移動を有効にします</p> <p>1.4. ダイアログ ウィンドウで設定します</p> <table border="1" data-bbox="100 755 703 1258"> <tr> <td colspan="2">Behavior</td> </tr> <tr> <td>ColumnMovingClientEvents</td> <td></td> </tr> <tr> <td>HeaderDragEnd</td> <td></td> </tr> <tr> <td>HeaderDragStart</td> <td></td> </tr> <tr> <td>HeaderDropped</td> <td></td> </tr> <tr> <td>HeaderMove</td> <td></td> </tr> <tr> <td>Initialize</td> <td></td> </tr> <tr> <td>Enabled</td> <td>True</td> </tr> <tr> <td colspan="2">Misc</td> </tr> <tr> <td>BottomDragIndicatorCssClass</td> <td></td> </tr> <tr> <td>ColumnSettings</td> <td>(Collection)</td> </tr> <tr> <td>DragMarkupCssClass</td> <td></td> </tr> <tr> <td>DragStyle</td> <td>Slide</td> </tr> <tr> <td>EnableInheritance</td> <td>False</td> </tr> <tr> <td>MiddleDragIndicatorCssClass</td> <td></td> </tr> <tr> <td>TopDragIndicatorCssClass</td> <td></td> </tr> </table> <p>詳細</p> <ul style="list-style-type: none"> WebDataGrid の列の移動を有効にする 	Behavior		ColumnMovingClientEvents		HeaderDragEnd		HeaderDragStart		HeaderDropped		HeaderMove		Initialize		Enabled	True	Misc		BottomDragIndicatorCssClass		ColumnSettings	(Collection)	DragMarkupCssClass		DragStyle	Slide	EnableInheritance	False	MiddleDragIndicatorCssClass		TopDragIndicatorCssClass		<p>← 親グリッドも同様</p> <p>子バンドに動作を追加するには、『WebHierarchicalDataGrid 子バンドに動作を追加する』を参照してください。</p>
Behavior																																	
ColumnMovingClientEvents																																	
HeaderDragEnd																																	
HeaderDragStart																																	
HeaderDropped																																	
HeaderMove																																	
Initialize																																	
Enabled	True																																
Misc																																	
BottomDragIndicatorCssClass																																	
ColumnSettings	(Collection)																																
DragMarkupCssClass																																	
DragStyle	Slide																																
EnableInheritance	False																																
MiddleDragIndicatorCssClass																																	
TopDragIndicatorCssClass																																	

列: 固定

WebDataGrid	WebHierarchicalDataGrid
<p>1. 列固定動作を有効にする</p> <ul style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] を選択します 1.3. 列の固定をチェックします 1.4. デフォルトの列幅を指定します <div data-bbox="201 695 978 1179" style="border: 1px solid gray; padding: 10px; margin: 20px auto; width: fit-content;"> <p>Default Grid and Column Widths</p> <p>ColumnFixing Requirements The ColumnFixing behavior requires the grid to have a fixed width and all columns to have fixed pixel widths. Set these values using the fields below.</p> <p>Grid Width: <input type="text" value="400"/> px</p> <p>Default Column Width: <input type="text"/> px</p> <p style="text-align: right;"> <input type="button" value="OK"/> <input type="button" value="Cancel"/> </p> </div> <p>2. 詳細な設定が必要な場合は、ダイアログ ウィンドウで列固定動作を設定します</p>	<p>実装なし</p>

注: ピン固定された列の位置を指定するには、Fixing Behavior プロパティで FixLocation を Left または Right に設定してください。

列: サイズ変更

WebDataGrid	WebHierarchicalDataGrid																										
<p>1. 列サイズ変更動作を有効にする</p> <ul style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] を選択します 1.3. 列サイズ変更をチェックします 1.4. ダイアログ ウィンドウで設定します <table border="1" data-bbox="100 836 701 1252"> <thead> <tr> <th colspan="2">Behavior</th> </tr> </thead> <tbody> <tr> <td>ColumnResizingClientEvents</td> <td></td> </tr> <tr> <td> ColumnResized</td> <td></td> </tr> <tr> <td> ColumnResizeDragging</td> <td></td> </tr> <tr> <td> ColumnResizing</td> <td></td> </tr> <tr> <td> Initialize</td> <td></td> </tr> <tr> <td>Enabled</td> <td>True</td> </tr> <tr> <th colspan="2">Misc</th> </tr> <tr> <td>AutoPostBackFlags</td> <td></td> </tr> <tr> <td> ColumnResized</td> <td>False</td> </tr> <tr> <td>ColumnSettings</td> <td>(Collection)</td> </tr> <tr> <td>EnableInheritance</td> <td>False</td> </tr> <tr> <td>ResizeIndicatorCssClass</td> <td></td> </tr> </tbody> </table>	Behavior		ColumnResizingClientEvents		ColumnResized		ColumnResizeDragging		ColumnResizing		Initialize		Enabled	True	Misc		AutoPostBackFlags		ColumnResized	False	ColumnSettings	(Collection)	EnableInheritance	False	ResizeIndicatorCssClass		<p>← 親グリッドも同様</p> <p>子バンドに動作を追加するには、『WebHierarchicalDataGrid 子バンドに動作を追加する』を参照してください。</p>
Behavior																											
ColumnResizingClientEvents																											
ColumnResized																											
ColumnResizeDragging																											
ColumnResizing																											
Initialize																											
Enabled	True																										
Misc																											
AutoPostBackFlags																											
ColumnResized	False																										
ColumnSettings	(Collection)																										
EnableInheritance	False																										
ResizeIndicatorCssClass																											

列: バインドされていない列

WebDataGrid	WebHierarchicalDataGrid
-------------	-------------------------

1. UnboundField を作成します

- 1.1. [スマート タグ] を開きます
- 1.2. [列の編集] を選択します
- 1.3. **UnboundField** を選択します
- 1.4. [項目の追加] ボタンをクリックします
- 1.5. 選択フィールド リストで新しいフィールドを選択します
 - 1.5.1. DataFormatString を {0:c2} (値は状況によって異なります) に設定します。
 - 1.5.2. DataType を System.Decimal (値は状況によって異なります) に設定します。
 - 1.5.3. Key フィールドを一意的な値に設定します。
 - 1.5.4. Header -> Text を任意のヘッダー テキストに設定します。
- 1.6. ダイアログを閉じます

2. RowInitialized イベントでフィールドのロジックを実装します。

- 2.1. グリッドのプロパティ ウィンドウで、イベント ビューへ切り替えます
- 2.2. **RowInitialized** イベントをダブルクリックします
- 2.3. RowInitialized イベント ハンドラーの非バインド フィールドのロジックを実装します

非バインド列の作成

以下のコードはすべての列のロジック (数量 x 価格) を実装します

← WebHierarchicalDataGrid へのアプローチも通常同じです。ただし、子バンドに設定を追加して子バンドのスコープにロジックを適用する必要があります。

UnitPrice	UnitsInStock	Total Retail
18.0000	39	\$702.00

ASPX の場合:

<Columns>

...

<ig:UnboundField

Key="TotalRetail"

DataType="System.Decimal"

DataFormatString="{0:c2}">

<Header Text="Total Retail" />

</ig:UnboundField>

列:

コード ビハインドの場合:

InitializeRow イベント ハンドラーで実装します

```
using Infragistics.Web.UI.GridControls;
```

...

```
GridRecordItem unbound;
```

```
GridRecordItem bound1;
```

```
GridRecordItem bound2;
```

```
int indexOfUnbound = 4;
```

```

int indexOfBound1 = 3;
int indexOfBound2 = 2;

unbound = e.Row.Items[indexOfUnbound];
bound1 = e.Row.Items[indexOfBound1];
bound2 = e.Row.Items[indexOfBound2];

decimal unitPrice = Convert.ToDecimal(bound1.Value);
int qty = Convert.ToInt32(bound2.Value);
decimal total = unitPrice * qty;

e.Row.Items[indexOfUnbound].Value = total;

```

列: チェックボックス列

WebDataGrid	WebHierarchicalDataGrid
<p>グリッドのブール値 フィールドへのバインドは、列タイプとしてデフォルトで BoundCheckboxField を作成します。</p> <p>更に、IBooleanConverter インターフェイスを実装して、バインドしたデータをチェックボックス フィールドで表示できます。</p> <p>非ブール値データをチェックボックスとしてバインドする</p> <p>ASPX の場合:</p> <p><Columns></p>	<p>← 通常 WebHierarchicalDataGrid での手順も同じです。ただし、子バンドに設定を追加して子バンドのスコープにコンバーターを割り当てる必要があります。</p>

```

<ig:BoundCheckBoxField
  DataFieldName="UnitPrice"
  CellText=" High Unit Price"
  Key="UnitPrice">
  <Header Text="UnitPrice" />
  <CheckBox
    CheckedImageUrl="~/images/checked.png"
    UncheckedImageUrl="~/images/unchecked.png"
    PartialImageUrl="~/images/partial.png" />
</ig:BoundCheckBoxField>
</Columns>

```

コード ビハインドの場合:

```

public partial class WDG : System.Web.UI.Page
{
  private int UNIT_PRICE_COLUMN_INDEX = 0;

  protected void Page_Load(object sender, EventArgs e)
  {
    BoundCheckBoxField field = (BoundCheckBoxField)

this.WebDataGrid1.Columns[UNIT_PRICE_COLUMN_INDEX];

    field.ValueConverter = new UnitPriceConverter();
  }
}

```

```

}

public class UnitPriceConverter :
Infragistics.Web.UI.GridControls.IBooleanConverter
{1
    public object DefaultFalseValue
    {
        get { return 0; }
    }

    public object DefaultTrueValue
    {
        get { return 100; }
    }

    public bool IsFalse(object value)
    {
        if (value != null && value is decimal)
        {
            return (decimal)value < 100;
        }
        return false;
    }

    public bool IsTrue(object value)

```

<pre> { if (value != null && value is decimal) { return (decimal)value >= 100; } return false; } </pre> <p>注: 実装では、decimal 型以外の他のデータ型をサポートする必要がある可能性があります。</p>	
---	--

列: 行集計

	WebHierarchicalDataGrid
<p>1. 集計行の動作を追加する</p> <ul style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] を選択します 1.3. 集計行をチェックします <p>注: CompactRendering を On に設定して同じ行に異なる集計の詳細を表示できるため、集計をコンパクトに表示できます。</p>	<p>← 親グリッドも同様</p> <p>子バンドに動作を追加するには、『WebHierarchicalDataGrid 子バンドに動作を追加する』を参照してください。</p>

CRUD: 自動

WebDataGrid	WebHierarchicalDataGrid
<p>グリッドの CRUD アクションは一括処理されるため、自動 CRUD を設定するとページにコントロールを追加して変更メッセージをサーバーへ送信する必要があります。</p> <ol style="list-style-type: none"> 1. データソースの設定 <ol style="list-style-type: none"> 1.1. 挿入、選択、更新、削除機能を実装するデータソース コントロールにグリッドをバインドします 2. DataKeyFields プロパティを設定します <ol style="list-style-type: none"> 2.1. プロパティ ウィンドウで DataKeyFields に主要キーの値の名前を設定します。 3. 編集動作を有効にする <ol style="list-style-type: none"> 3.1. [スマート タグ] を開きます 3.2. [動作の編集] を選択します 3.3. [選択] をチェックします <ol style="list-style-type: none"> 3.3.1. RowSelectType = Single に設定します 3.4. 行セクターをチェックします 3.5. [コアの編集機能] をチェックします <ol style="list-style-type: none"> 3.5.1. AutoCRUD が True であることを確認します 3.6. セルの編集をチェックします 3.7. 行の追加をチェックします 	<p>← 親グリッドも同様</p> <p>子バンドに動作を追加するには、『WebHierarchicalDataGrid 子バンドに動作を追加する』を参照してください。</p>

<p>3.8. 行の削除をチェックします</p> <p>3.9. [OK] をクリックします</p> <p>4. コントロールを追加してポストバックを発生させる</p> <p>4.1. サーバー ボタンなどのコントロールを追加して、サーバーへのポストバックを開始します</p> <p>詳細</p> <ul style="list-style-type: none"> • WebDataGrid 自動 CRUD (英語ビデオ) 	
--	--

CRUD: 手動

WebDataGrid	WebHierarchicalDataGrid
<p>1. 手動 CRUD 操作にグリッドを設定する</p> <p>1.1. プロパティ ウィンドウで DataKeyFields に主要キーの値の名前を設定します。</p> <p>1.2. EnableAjax プロパティを False と同じに設定します。</p> <p>2. CRUD 動作を有効にする</p> <p>2.1. [スマート タグ] を開きます</p> <p>2.2. [動作の編集] を選択します</p> <p>2.3. [選択] をチェックします</p> <p>2.4. [行セレクター] をチェックします</p> <p>2.5. [アクティブ化] をチェックします</p> <p>2.6. [コアの編集機能] をチェックします</p> <p>2.6.1. AutoCRUD が False であることを確認します</p> <p>2.7. セルの編集をチェックします</p>	<p>← 親グリッドも同様</p> <p>子バンドに動作を追加するには、『WebHierarchicalDataGrid 子バンドに動作を追加する』を参照してください。</p>

2.7.1. EditModeActions -> EnableOnKeyPress を True に設定します

2.8. 行の追加をチェックします

2.8.1. EditModeActions -> EnableOnKeyPress を True に設定します

2.9. 行の削除をチェックします

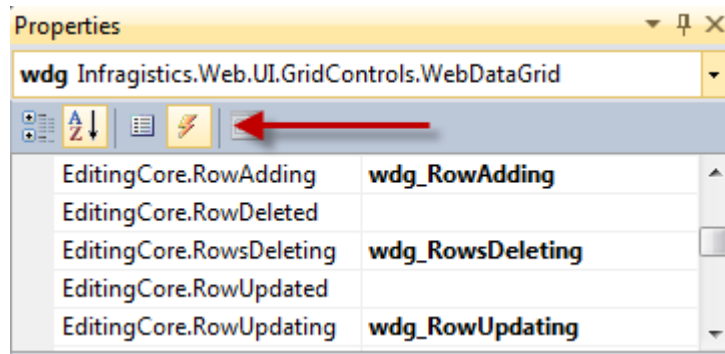
3. CRUD イベントをサブスクライブ

3.1. プロパティ ウィンドウでイベントをダブル クリックして以下のイベント ハンドラーを作成します。

3.1.1. **EditingCore.RowAdding**

3.1.2. **EditingCore.RowDeleting**

3.1.3. **EditingCore.RowUpdating**



4. CRUD ロジックの実装

4.1. 通常、グリッドへ最新データを公開するのと同様、CRUD メッセージを管理するロジックを実装する必要があります。1 つの方法として、各ページの読み込みでグリッドを再度バインドし、各ハンドラーの

CRUD 操作の実行を許可する方法があります。以下のコード スニペットにこのデザインを反映しています。

```
using Infragistics.Web.UI.GridControls;
...
public void BindData()
{
    this.wdg.Rows.Clear();
    this.wdg.DataSource = this.GetData();
    this.wdg.DataBind();
}

public IList<Person> GetData()
{
    // データを取得するために
    // データ アクセス レイヤーを呼び出します

    // 返却型に IList インターフェイスや Person 型を
    // 含みません

    // このメソッド シグネチャは、プレースホルダーのためのものです。
}

protected void Page_Load(object sender, EventArgs e)
{
```

```

if (!this.Page.IsPostBack)
{
    this.BindData();
}
}

protected void wdg_RowAdding(object sender,
Infragistics.Web.UI.GridControls.RowAddingEventArgs e)
{
    Person person = new Person()
    {
        FirstName = e.Values["FirstName"].ToString(),
        LastName = e.Values["LastName"].ToString()
    };

    this.Repository.Insert(person);
    this.BindData();
}

protected void wdg_RowsDeleting(object sender,
RowDeletingEventArgs e)
{
    int id = Convert.ToInt32(e.Row.DataKey[0]);
    this.Repository.Delete(id);
    this.BindData();
}

```

```

}

protected void wdg_RowUpdating(object sender,
Infragistics.Web.UI.GridControls.RowUpdatingEventArgs e)
{
    Person person = new Person()
    {
        Id = Convert.ToInt32(e.Row.DataKey[0]),
        FirstName = e.Values["FirstName"].ToString(),
        LastName = e.Values["LastName"].ToString()
    };

    this.Repository.Update(person);
    this.BindData();
}

```

DataBinding: HeirarchicalDataSource

WebDataGrid	WebHierarchicalDataGrid
実装なし	<p>WebHierarchicalDataSource コントロールはラッパーとして動作し、関連フィールドを介して相互に関連付ける 2 つ以上のデータソース コントロールのコネクターです。</p> <p>以下は、データソースを子ビューとして WebHierarchicalDataSource に追加</p>

する手順です。

1. WebHierarchicalDataGrid をページに追加します

2. WebHierarchicalDataSource コントロールをページに追加/設定します

2.1. WebHierarchicalDataSource コントロールをページに追加します

2.2. ID プロパティを必要な値に設定します

2.3. データソースの [スマート タグ] を開きます

2.4. DataSource の設定を選択します

2.5. [ビューの追加] をクリックします

3. 最初のデータソースを追加する

3.1. DataSource ドロップ ダウンから新しいデータソースを選択します

3.2. ダイアログ ウィンドウからデータソース タイプを選択します

3.2.1. データソース設定ウィザードでデータソースを設定し、終了をクリックします

注: データソースには、手順 4.3.1 で追加した関連データの外部キーの値に関連する外部キー フィールドのセクション 4.3 ヨンを含む必要があります。

3.3. WebHierarchicalDataSource Designer ダイアログ ウィンドウで OK をクリックします

4. 2つ目のデータソースを追加する

4.1. WebHierarchicalDataSource ダイアログウィンドウの編集で作成したデータソースの下にある「子の追加」をクリックします

4.2. 子データソース ドロップ ダウンから新しいデータソースを選択しま

す

4.3. ダイアログ ウィンドウからデータソース タイプを選択します

4.3.1. データソース設定ウィザードでデータソースを設定し、終了をクリックします

注: データソースには、手順 3.2.1 で追加した関連データのプライマリキーの値に関連するプライマリ キー フィールドのセクションを含む必要があります。

5. WebHierarchicalDataSource ダイアログ ウィンドウの編集でデータソース間のフィールドを関連付けます

5.1. 親列ドロップ ダウンをクリックします

5.1.1. フィールド オプションからプライマリ キー フィールドを選択します

5.2. 子列ドロップ ダウンをクリックします

5.2.1. フィールド オプションから外部キー フィールドを選択します

5.3. [OK] をクリックします

6. デザイナー ダイアログを閉じます

6.1. [OK] をクリックします

7. WebHierarchicalDataSource を WebHierarchicalDataGrid に関連付ける

7.1. グリッドの [スマート タグ] を開きます

7.2. データソースの選択のドロップ ダウンをクリックします

7.3. 手順 2.2 で定義した WebHierarchicalDataSource の ID を選択します

注: グリッドの列およびキー設定を再生成するかどうかのダイアログ ボ

	<p>ックスが表示される場合があります。グリッドの設定が初めての場合は [はい] をクリックします。分からない場合は、[いいえ] をクリックして列とキーオプションを手動で設定します。</p> <p>詳細</p> <ul style="list-style-type: none"> • WebHierarchicalDataSource を使用した作業の開始
--	---

DataBinding: DataSet

WebDataGrid	WebHierarchicalDataGrid
<p>DataSet を WebDataGrid へバインドするには、DataSet を DataSource プロパティに設定してからグリッドの DataBind を呼び出します。</p> <p>コード ビハインドの場合:</p> <pre>protected void Page_Load(object sender, EventArgs e) { ProductRepository repository = new ProductRepository(); DataSet ds = repository.GetAllDataSet(); this.WebDataGrid1.DataSource = ds; this.WebDataGrid1.DataBind(); }</pre>	<p>← 同じ</p>

DataBinding: DataTable

WebDataGrid	WebHierarchicalDataGrid
-------------	-------------------------

<p>DataTable を WebDataGrid へバインドするには、DataTable を DataSource プロパティに設定してからグリッドの DataBind を呼び出します。</p> <p>コード ビハインドの場合:</p> <pre>protected void Page_Load(object sender, EventArgs e) { ProductRepository repository = new ProductRepository(); DataTable ds = repository.GetAllDataTable(); this.WebDataGrid1.DataSource = ds; this.WebDataGrid1.DataBind(); }</pre>	<p>← 同じ</p>
--	-------------

DataBinding: IEnumerable

WebDataGrid	WebHierarchicalDataGrid
<p>IEnumerable を WebDataGrid コレクションへバインドするには、IEnumerable を DataSource プロパティに設定してからグリッドの DataBind を呼び出します。</p> <p>コード ビハインドの場合:</p> <pre>protected void Page_Load(object sender, EventArgs e) { ProductRepository repository = new ProductRepository();</pre>	<p>← 同じ</p>

```

IEnumerable<Product> data = repository.GetAllEnumerable();
this.WebDataGrid1.DataSource = data;
this.WebDataGrid1.DataBind();
}

```

DataBinding: AccessDataSource

WebDataGrid	WebHierarchicalDataGrid
<p>1. AccessDataSource コントロールをページに追加する</p> <ol style="list-style-type: none"> 1.1. AccessDataSource コントロールをデザイン サーフェスヘドラッグ します 1.2. プロパティ ウィンドウを開いて Id プロパティのコントロールに値を 設定します 1.3. データソースの [スマート タグ] を開きます 1.4. ウィザードに従ってデータソース コントロールの設定を完了します <p>2. データソース コントロールをグリッドに関連付ける</p> <ol style="list-style-type: none"> 2.1. グリッドの [スマート タグ] を開きます 2.2. データソース選択の隣りのドロップ ダウンをクリックします 2.3. 手順 1.2 で定義したデータソース の ID を選択します <p>注: グリッドの列およびキー設定を再生成するかどうかのダイアログ ボックスが表示される場合があります。グリッドの設定が初めての場合は [はい] をクリックします。分からない場合は、[いいえ] をクリックし て列とキーオプションを手動で設定します。</p>	<p><i>DataBinding: HierarchicalDataSource</i> の手順に従って、3.2.1 および 4.3.1 またはそのどちらかのデータソースのタイプとして AccessDataSource を選 択します。</p>

DataBinding: SqlDataSource

WebDataGrid	WebHierarchicalDataGrid
<p>1. SqlDataSource コントロールをページに追加する</p> <ol style="list-style-type: none"> 1.1. SqlDataSource コントロールをデザイン サーフエスヘドラッグします 1.2. プロパティ ウィンドウを開いて Id プロパティのコントロールに値を設定します 1.3. データソースの [スマート タグ] を開きます 1.4. ウィザードに従ってデータソース コントロールの設定を完了します <p>2. データソース コントロールをグリッドに関連付ける</p> <ol style="list-style-type: none"> 2.1. グリッドの [スマート タグ] を開きます 2.2. データソースの選択の隣のドロップ ダウンをクリックします 2.3. 手順 1.2 で定義したデータソース の ID を選択します <p>注: グリッドの列およびキー設定を再生成するかどうかのダイアログ ボックスが表示される場合があります。グリッドの設定が初めての場合は [はい] をクリックします。分からない場合は、[いいえ] をクリックして列とキーオプションを手動で設定します。</p>	<p><i>DataBinding: HierarchicalDataSource</i> の手順に従って、3.2.1 および 4.3.1 またはそのどちらかのデータソースのタイプとして SqlDataSource を選択します。</p>

DataBinding: ObjectDataSource

WebDataGrid	WebHierarchicalDataGrid
<p>1. SqlDataSource コントロールをページに追加する</p> <ol style="list-style-type: none"> 1.1. ObjectDataSource コントロールをデザイン サーフエスヘドラッグ 	<p><i>DataBinding: Hierarchical Data</i> の手順に従って、3.2.1 および 4.3.1 またはそのどちらかのデータソースのタイプとして ObjectDataSource を選択し</p>

<p>します</p> <ol style="list-style-type: none"> 1.2. プロパティ ウィンドウを開いて Id プロパティのコントロールに値を設定します 1.3. データソースの [スマート タグ] を開きます 1.4. ウィザードに従ってデータソース コントロールの設定を完了します <p>2. データソース コントロールをグリッドに関連付ける</p> <ol style="list-style-type: none"> 2.1. グリッドの [スマート タグ] を開きます 2.2. データソースの選択の隣のドロップ ダウンをクリックします 2.3. 手順 1.2 で定義したデータソース の ID を選択します <p>注: グリッドの列およびキー設定を再生成するかどうかのダイアログ ボックスが表示される場合があります。グリッドの設定が初めての場合は [はい] をクリックします。分からない場合は、[いいえ] をクリックして列とキーオプションを手動で設定します。</p>	<p>ます。</p>
--	------------

DataBinding: LinqDataSource

WebDataGrid	WebHierarchicalDataGrid
<p>1. LinqDataSource コントロールをページに追加する</p> <ol style="list-style-type: none"> 1.1. LinqDataSource コントロールをデザイン サーフェスへドラッグします 1.2. プロパティ ウィンドウを開いて Id プロパティのコントロールに値を設定します 1.3. データソースの [スマート タグ] を開きます 	<p><i>DataBinding: Hierarchical Data</i> の手順に従って、3.2.1 および 4.3.1 またはそのどちらかのデータソースのタイプとして LinqDataSource を選択します。</p>

<p>1.4. ウィザードに従ってデータソース コントロールの設定を完了します</p> <p>2. データソース コントロールをグリッドに関連付ける</p> <p>2.1. グリッドの [スマート タグ] を開きます</p> <p>2.2. データソースの選択の隣りのドロップ ダウンをクリックします</p> <p>2.3. 手順 1.2 で定義したデータソース の ID を選択します</p> <p>注: グリッドの列およびキー設定を再生成するかどうかのダイアログ ボックスが表示される場合があります。グリッドの設定が初めての場合は [はい] をクリックします。分からない場合は、[いいえ] をクリックして列とキーオプションを手動で設定します。</p>	
--	--

編集: 行の追加

WebDataGrid	WebHierarchicalDataGrid
<p>1. データソースの設定</p> <p>1.1. 選択と挿入を実装するデータソース コントロールにグリッドをバインドします</p> <p>2. DataKeyFields プロパティを設定します</p> <p>2.1. プロパティ ウィンドウで DataKeyFields に主要キーの値の名前を設定します。</p> <p>3. 編集機能を有効にする</p> <p>3.1. [スマート タグ] を開きます</p> <p>3.2. [動作の編集] を選択します</p> <p>3.3. [コアの編集機能] をチェックします</p>	<p>1. データソースの設定</p> <p>1.1. 選択と挿入を実装するデータソース コントロールにグリッドをバインドします</p> <p>2. DataKeyFields プロパティを設定します</p> <p>2.1. プロパティ ウィンドウで DataKeyFields に主要キーの値の名前を設定します。</p> <p>3. 編集動作を有効にする</p> <p>3.1. [スマート タグ] を開きます</p> <p>3.2. [動作の編集] を選択します</p> <p>3.3. [コアの編集機能] をチェックします</p>

クライアント側

```
function addRow() {
    var grid = $find("<%= this.WebDataGrid1.ClientID %>");

    行に挿入するためにすべての値で配列を
    var values = ["10000", "New Product"];

    grid.get_rows().add(values);
}
```

サーバー側

サーバーでグリッドに行を追加するには、データソースにデータを追加してグリッドを再度バインドします。

カスタマイズ

ColumnSettings で DefaultValueAsString, ReadOnly, EditorID および ValidatorID に値を提供して列毎に新しいレコードを追加するプロセスを詳細にカスタマイズできます。

ASPX の場合:

```
<Behaviors>
    <ig:EditingCore>
    </Behaviors>
```

クライアント側

```
ルート レベル
var grid = $find("WebHierarchicalDataGrid1");

var topRowIsland = grid.get_gridView();
var cellValues = ["1", "Bob", "Green", "1/2/1983"];
topRowIsland.get_rows().add(cellValues);
```

Child Level

```
var grid = $find("WebHierarchicalDataGrid1");

var childGrid =
topRowIsland.get_rows().get_row(3).get_rowIslands()[0];
cellValues = ["1", "25 Main Road", "New York", "NY", "19234"];
childGrid.get_rows().add(cellValues);
```

サーバー側

サーバーでグリッドに行を追加するには、データソースにデータを追加してグリッドを再度バインドします。

カスタマイズ

ColumnSettings で DefaultValueAsString, ReadOnly, EditorID および ValidatorID に値を提供して列毎に新しいレコードを追加するプロセスを詳細にカスタマイズできます。

```

<ig:RowAdding>
  <ColumnSettings>
    <%-- デフォルト値の設定 --%>
    <ig:RowAddingColumnSetting
      ColumnKey="Size"
      DefaultValueAsString="3" />

    <%-- ID 列を読み取り専用とし、
    値は未設定とします --%>
    <ig:RowAddingColumnSetting
      ColumnKey="Id" ReadOnly="true" />

    <%-- エディターとバリデーターを設定 --%>
    <ig:RowAddingColumnSetting
      ColumnKey="OrderDate"
      EditorID="DateTimePicker1"
      ValidatorID="myValidator1" />
  </ColumnSettings>
</ig:RowAdding>
</Behaviors>
</ig:EditingCore>
</Behaviors>

```

コード ビハインドの場合:

Page_Load イベント ハンドラーで実行します

ASPX の場合:

← WebHierarchicalDataGrid のマークアップは基本的に同じです。
WebDataGrid 同様、グリッドの適切なレベルの列をカスタマイズするだけです。

コード ビハインドの場合:

InitializeBand イベント ハンドラーで実行します

```

if (e.Band.DataMember == "Root")
{
  RowAdding rowAddingBehavior =
    e.Band.Behaviors.EditingCore.Behaviors.RowAdding;

  /* 新しいデータにデフォルト値を設定します */
  RowAddingColumnSetting sizeSetting = new
  RowAddingColumnSetting();
  sizeSetting.ColumnKey = "Size";
  sizeSetting.DefaultValueAsString = "3";
  rowAddingBehavior.ColumnSettings.Add(sizeSetting);

  /* ID 列を読み取り専用とし、値は未設定とします。 */
  RowAddingColumnSetting idSetting = new
  RowAddingColumnSetting();
  idSetting.ColumnKey = "Id";
}

```

```
RowAdding rowAddingBehavior =
this.WebDataGrid1.Behaviors.EditingCore.Behaviors.RowAdding;
```

```
/* デフォルト値の設定 */
```

```
RowAddingColumnSetting sizeSetting = new
```

```
RowAddingColumnSetting();
```

```
sizeSetting.ColumnKey = "Size";
```

```
sizeSetting.DefaultValueAsString = "3";
```

```
rowAddingBehavior.ColumnSettings.Add(sizeSetting);
```

```
/* ID 列を読み取り専用とし、値は未設定とします。*/
```

```
RowAddingColumnSetting idSetting = new RowAddingColumnSetting();
```

```
idSetting.ColumnKey = "Id";
```

```
idSetting.ReadOnly = true;
```

```
rowAddingBehavior.ColumnSettings.Add(idSetting);
```

```
/* エディターとバリデーターを設定 */
```

```
RowAddingColumnSetting orderDateSetting = new
```

```
RowAddingColumnSetting();
```

```
orderDateSetting.ColumnKey = "OrderDate";
```

```
orderDateSetting.EditorID = "DateTimePicker1";
```

```
orderDateSetting.ValidatorID = "myValidator1";
```

```
rowAddingBehavior.ColumnSettings.Add(orderDateSetting);
```

```
idSetting.ReadOnly = true;
```

```
rowAddingBehavior.ColumnSettings.Add(idSetting);
```

```
/* エディターとバリデーターを設定 */
```

```
RowAddingColumnSetting orderDateSetting =
```

```
new RowAddingColumnSetting();
```

```
orderDateSetting.ColumnKey = "OrderDate";
```

```
orderDateSetting.EditorID = "DateTimePicker1";
```

```
orderDateSetting.ValidatorID = "myValidator1";
```

```
rowAddingBehavior.ColumnSettings.Add(orderDateSetting);
```

```
}
```

```
else if (e.Band.DataMember == "SecondLevel")
```

```
{
```

```
RowAdding rowAddingBehavior =
```

```
e.Band.Behaviors.EditingCore.Behaviors.RowAdding;
```

```
/* 上記と同じ方法で
```

```
子グリッドも設定できます*/
```

```
}
```

編集: 行の削除

WebDataGrid	WebHierarchicalDataGrid
<p>1. 行削除動作を追加する</p> <ol style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] をクリックします 1.3. [選択] をチェックします 1.4. [コアの編集機能] をチェックします 1.5. 行の削除をチェックします <p>削除動作がグリッドに追加されるとグリッドからさまざまな方法で行を削除できます。</p> <p>注: グリッドにアタッチされているデータソースが自動的にグリッドによって更新されない場合、更新イベントを手動で処理してサーバーに変更を行う必要があります。</p> <p>サーバー側</p> <p>サーバーでグリッドから行を削除するには、データソースから行を削除してグリッドを再度バインドします。</p> <p>クライアント側</p> <p>選択行の削除 - ベーシック</p> <pre>var grid = \$find('<%= WebDataGrid1.ClientID %>');</pre>	<p>1. 行削除動作を追加する</p> <ol style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] をクリックします 1.3. [選択] をチェックします 1.4. [コアの編集機能] をチェックします 1.5. 行の削除をチェックします <p>削除動作がグリッドに追加されるとグリッドからさまざまな方法で行を削除できます。</p> <p>注: グリッドにアタッチされているデータソースが自動的にグリッドによって更新されない場合、更新イベントを手動で処理してサーバーに変更を行う必要があります。</p> <p>サーバー側</p> <p>サーバーでグリッドから行を削除するには、データソースから行を削除してグリッドを再度バインドします。</p> <p>クライアント側</p> <p>選択された行を任意のレベルで削除します</p> <pre>var grid = \$find('<%= WebHierarchicalDataGrid1.ClientID %>');</pre>

```

var gridRows = grid.get_rows()

var selectedRows = grid.get_behaviors().
    get_selection().
    get_selectedRows();

var rows = new Array();
var i1 = 0;

for (var i = selectedRows.get_length() - 1; i >= 0; i--) {

    rows[i1] = selectedRows.getItem(i);
    i1++;
}

grid.get_behaviors().
    get_editingCore().
    get_behaviors().
    get_rowDeleting().
    deleteRows(rows);

```

Delete Selected Rows – Using ‘Remove’ Method

```

var grid = $find('<%= WebDataGrid1.ClientID %>');
var gridRows = grid.get_rows()

```

```

var selection = grid.get_gridView().
    get_behaviors().
    get_selection();

// get_selectedRowsResolved() はグリッド全体で選択された行を取得しま
す
// 行配列を返します
var selectedRows = selection.get_selectedRowsResolved();

for (var i=0; i < selectedRows.length; i++) {
    var row = selectedRows[i];
    var containerGrid = row.get_grid();
    var gridRows = containerGrid.get_rows();
    gridRows.remove(row);
}

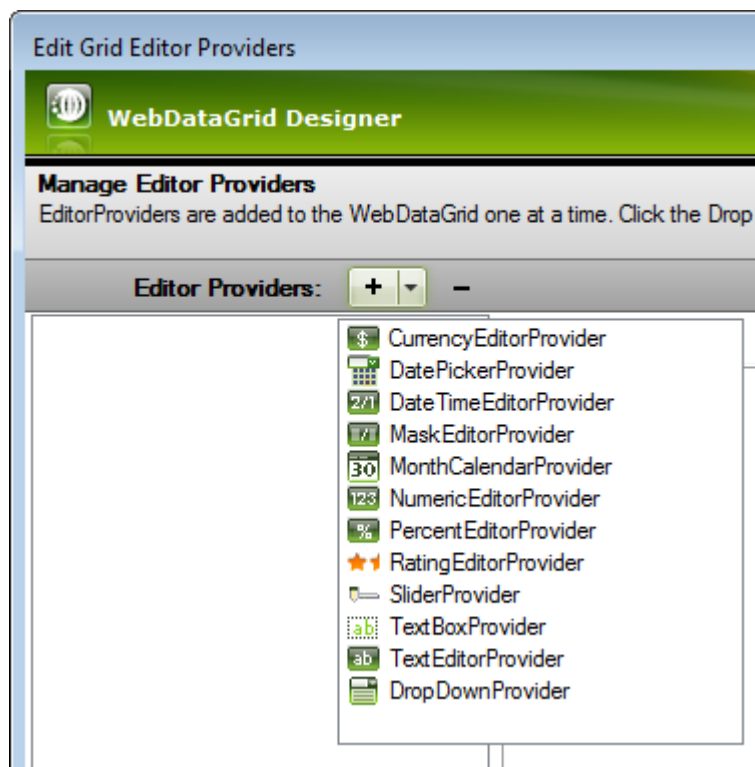
// 最初の親行を削除します
row = parentGrid.get_rows().get_row(0);
parentGrid.get_rows().remove(row);

```


<pre> var selectedRows = grid.get_behaviors(). get_selection(). get_selectedRows(); for (var i = selectedRows.get_length() - 1; i >= 0; i--) { var row = selectedRows.getItem(i); gridRows.remove(row); } Delete non-Selected Rows var grid = \$find('<%= WebDataGrid1.ClientID %>'); var row = grid.get_rows().get_row(0); grid.get_rows().remove(row); </pre>	<pre> var childGrid = grid.get_gridView(). get_rows(). get_row(0). get_rowIslands(0)[0]; // 最初の子行を削除します row = childGrid.get_rows().get_row(0); childGrid.get_rows().remove(row); </pre>
---	--

編集: 埋め込み (組み込み) エディター / エディター プロバイダー

WebDataGrid	WebHierarchicalDataGrid
<p>1. グリッドに編集プロバイダーを追加する</p> <ul style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. グリッド エディター プロバイダーの編集を選択します 1.3. 結果のダイアログで + ボタンをクリックします 	<p>← 同じ</p>



- 1.4. 必要なエディター プロバイダーを選択します
- 1.5. ダイアログ右側の詳細ペインの ID プロパティに適切な値を提供することによって設定します
- 1.6. [OK] をクリックします

2. エディター プロバイダーを列に関連付けます

- 2.1. [スマート タグ] を開きます
- 2.2. [動作の編集] を選択します
- 2.3. [コアの編集機能] をチェックします
- 2.4. セルの編集をチェックします

- 2.5. **ColumnSettings** プロパティの省略記号ボタン (...) をクリックします
- 2.6. [項目の追加] ボタンをクリックします
- 2.7. **ColumnKey** の値を選択します
- 2.8. EditorID の値を選択します
- 2.9. 開いているすべてのダイアログで OK をクリックします

Excel エクスポート

WebDataGrid	WebHierarchicalDataGrid
<ul style="list-style-type: none"> 1. Infragistics.Documents.Excel への参照を追加します。 2. WebExcelExporter をページに追加します <ul style="list-style-type: none"> 1.1. ページで WebExcelExporter をドラッグします 1.2. プロパティ ウィンドウで、eExporter に等しい Id を設定します 3. エクスポート処理を処理するボタンを作成する <ul style="list-style-type: none"> 2.1. ページにボタンをドラッグします 2.2. プロパティ ウィンドウで、btnExport に等しい Id を設定します 2.3. コードビハインドにクリック ハンドラーを作成するボタンをダブル クリックします <p>コード ビハインドの場合: btnExport_Click イベント ハンドラーで実行します</p>	<p>← 同じ</p>

```

// ファイルはワークブック形式に基づいて拡張子を取得します
this.eExporter.DownloadName = "data";

eExporter.WorkbookFormat =
Infragistics.Documents.Excel.WorkbookFormat.Excel2007;

// エクスポート モード - すべてのグリッド データまたはすべての表示データ
eExporter.DataExportMode =
Infragistics.Web.UI.GridControls.DataExportMode.AllDataInDataSource;

// エクセル セルに割り当てる前に DataFormatString を値に適用かどうかを
決定します
eExporter.DisableCellValueFormatting = true;

// スタイルがエクスポートされるかどうかを決定します
eExporter.EnableStylesExport = true;

// エクスポートされた複数のグリッド間に配置する行数
eExporter.GridsRowSpacing = 2;

eExporter.Export(new WebControl[] { WebDataGrid1,
WebDataGrid2 });

```

PDF エクスポート

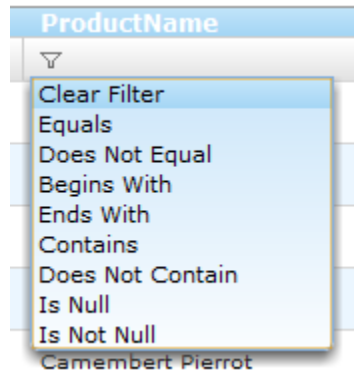
WebDataGrid	WebHierarchicalDataGrid
<p>1. WebDocumentExplorer をページに追加します</p> <p>1.1. ページで WebDocumentExporer をドラッグします</p> <p>1.2. プロパティ ウィンドウで、dExporter に等しい Id を設定します。</p> <p>2. Create a button to handle the export procedure</p> <p>2.1. Button コントロールをページヘドラッグします</p> <p>2.2. プロパティ ウィンドウで、btnExport に等しい Id を設定します</p> <p>2.3. コードビハインドにクリック ハンドラーを作成するボタンをダブルクリックします</p> <p>コード ビハインドの場合:</p> <p>btnExport_Click イベント ハンドラーで実行します</p> <pre> this.dExporter.DownloadName = "data.pdf"; dExporter.Format = Infragistics.Web.UI.GridControls.FileFormat.PDF; // エクスポート モード - すべてのグリッド データまたはすべての表示データ dExporter.DataExportMode = Infragistics. Web UI. GridControls. DataExportMode. </pre>	<p>← 同じ</p>

<pre> DataInGridOnly; dExporter.TargetPaperOrientation = Infragistics. Documents. Report. PageOrientation. Portrait; dExporter.Margins = PageMargins.Normal; dExporter.TargetPaperSize = PageSizes.A4; // ページでグリッドを非表示にします // 2 つ以上のグリッドを追加したい場合 // WebControl[] 配列として追加します dExporter.Export((WebDataGrid1, WebDataGrid2); </pre>	
--	--

フィルター

WebDataGrid	WebHierarchicalDataGrid
<p>1. 行削除動作を追加する</p> <ul style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] をクリックします 1.3. [フィルター] をチェックします <p>グリッドでフィルター動作を有効にすると、ユーザーは列をさまざまな条件に</p>	<p>1. 行削除動作を追加する</p> <ul style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] をクリックします 1.3. [フィルター] をチェックします <ul style="list-style-type: none"> 1.3.1. EnableInheritance を True に設定します。

基づいて列をフィルターできます



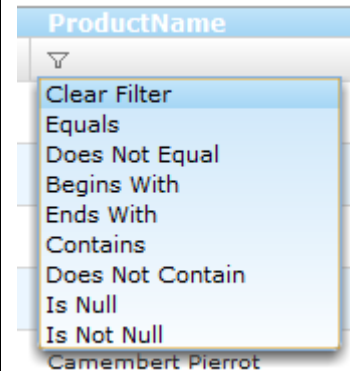
デフォルトでフィルター動作はすべての列で有効で、フィルター条件空です。
以下のスニペットは、フィルター設定をカスタマイズする方法を説明します。

単一系列のフィルターを無効にする

ASPX

```
<Behaviors>
  <ig:Filtering>
    <ColumnSettings>
      <ig:ColumnFilteringSetting
        ColumnKey="ProductName"
        Enabled="false" />
    </ColumnSettings>
  </ig:Filtering>
</Behaviors>
```

グリッドでフィルター動作を有効にすると、ユーザーは列をさまざまな条件に基づいて列をフィルターできます



デフォルトでフィルター動作はすべての列で有効で、フィルター条件空です。
以下のスニペットは、フィルター設定をカスタマイズする方法を説明します。

単一系列のフィルターを無効にする

ASPX - ルート レベル

動作。

```
<ig:Filtering EnableInheritance="True">
  <ColumnSettings>
    <ig:ColumnFilteringSetting
      ColumnKey="CompanyName"
      Enabled="False" />
  </ColumnSettings>
</ig:Filtering>
```

コード ビハインド:

```
using Infragistics.Web.UI.GridControls;
...
ColumnFilteringSetting columnSetting = new
    ColumnFilteringSetting(this.WebDataGrid1);
columnSetting.ColumnKey = "ProductName";
columnSetting.Enabled = false;

ColumnFilteringSettings settings = this.WebDataGrid1.
    Behaviors.
    Filtering.
    ColumnSettings;

settings.Add(columnSetting);
```

定義済みの列フィルター

以下のスニペットは、テキスト フィルターを実装する方法を示します。ただし、グリッドはブール値、日付、数値フィルター条件もサポートします。

.aspx

ColumnFilters

```
<ig:ColumnFilter ColumnKey="ProductName">
    <ConditionWrapper>
        <ig:RuleTextNode Rule="Contains" Value="al" />
    </ConditionWrapper>
```

```
</Behaviors>
```

ASPX - バンド レベル

```
<Bands>
```

```
    <ig:Band ...>
```

```
    ...
```

```
        <Behaviors>
```

```
            <ig:Filtering>
```

```
                <ColumnSettings>
```

```
                    <ig:ColumnFilteringSetting
```

```
                        ColumnKey="CompanyName"
```

```
                        Enabled="False" />
```

```
                </ColumnSettings>
```

```
            </ig:Filtering>
```

```
        </Behaviors>
```

```
    </ig:Band>
```

```
</Bands>
```

コード ビハインド - ルート レベル

Page_Load イベント ハンドラーで実行します

```
ColumnFilteringSetting columnSetting = new
ColumnFilteringSetting(this.WebHierarchicalDataGrid1.GridView);
columnSetting.ColumnKey = "ProductName";
columnSetting.Enabled = false;
```



```

</ig:ColumnFilter>
</ColumnFilters>

コード ビハインド
RuleTextNode textCondition = new
RuleTextNode(TextFilterRules.Contains, "a");

ColumnFilter columnFilter = new ColumnFilter();
columnFilter.ColumnKey = "ProductName";
columnFilter.Condition = textCondition;

this.grid.Behaviors.Filtering.ColumnFilters.Add(columnFilter);

this.WebDataGrid1.Behaviors.Filtering.ApplyFilter();

In JavaScript
var grid = $find("<%= this.WebDataGrid1.ClientID %>");

var columnFilter = grid.
    get_behaviors().
    get_filtering().
    create_columnFilter("ProductName");

var condition2 = columnFilter2.get_condition();
condition2.set_rule($IG.TextFilterRules.Contains);

```

```

this.WebHierarchicalDataGrid1.
    GridView.
    Behaviors.
    Filtering.
    ColumnSettings.
    Add(columnSetting);

```

Code Behind – Band Level

Execute in the InitializeBand event handler.

```

ColumnFilteringSetting settings = new ColumnFilteringSetting();
settings.ColumnKey = "SupplierID";
settings.Enabled = false;
e.Band.Behaviors.Filtering.ColumnSettings.Add(settings);

```

Code Behind – Row Island Level

Execute in the RowIslandCreated event handler

```

ColumnFilteringSetting columnSetting = new
ColumnFilteringSetting(e.RowIsland);
columnSetting.ColumnKey = "SupplierID";
columnSetting.Enabled = false;
e.RowIsland.Behaviors.Filtering.ColumnSettings.Add(columnSetting);

```

```
condition2.set_value("al");
```

```
var columnFilters = new Array(columnFilter);
```

```
grid.get_behaviors().
```

```
  get_filtering().
```

```
  add_columnFilterRange(columnFilters);
```

```
grid.get_behaviors().get_filtering().applyFilters();
```

JavaScript - ルート レベル

JavaScript - バンド レベル

定義済みの列フィルター

ASPX - ルート レベル

```
<Behaviors>
```

```
  <ig:Filtering EnableInheritance="True">
```

```
    <ColumnFilters>
```

```
      <ig:ColumnFilter ColumnKey="ProductName">
```

```
        <ConditionWrapper>
```

```
          <ig:RuleTextNode Rule="Contains" Value="al" />
```

```
        </ConditionWrapper>
```

```
      </ig:ColumnFilter>
```

```
    </ColumnFilters>
```

```
  </ig:Filtering>
```

```
</Behaviors>
```

ASPX - バンド レベル

バンド

```
<ig:Band ...>
```

```
  <Behaviors>
```

```
    <ColumnFilters>
```

```
      <ig:ColumnFilter ColumnKey="CompanyName">
```

```
        <ConditionWrapper>
```

```

        <ig:RuleTextNode
            Rule="Contains"
            Value="al" />
    </ConditionWrapper>
</ig:ColumnFilter>
</ColumnFilters>
</ig:Filtering>
</Behaviors>
</ig:Band>
</Bands>

```

コード ビハインド - ルート レベル

Page_Load イベント ハンドラーで実行します

```

ColumnFilter filter = new
ColumnFilter(this.WebHierarchicalDataGrid1.GridView);
filter.ColumnKey = "ProductName";
filter.Condition = new RuleTextNode(TextFilterRules.Contains, "al");
this.WebHierarchicalDataGrid1.
    GridView.
        Behaviors.
            Filtering.
                ColumnFilters.
                    Add(filter);

```

コード ビハインド - バンド レベル

InitializeBand イベント ハンドラーで実行します

```
ColumnFilter filter = new ColumnFilter();
filter.ColumnKey = "CompanyName";
filter.Condition = new RuleTextNode(
    TextFilterRules.Contains,
    "al");
e.Band.Behaviors.Filtering.ColumnFilters.Add(filter);
```

コード ビハインド - 列アイランド (子行) レベル

RowIslandCreated イベント ハンドラーで実行します

```
ColumnFilter filter = new ColumnFilter();
filter.ColumnKey = "CompanyName";
filter.Condition = new RuleTextNode(
    TextFilterRules.Contains,
    "al");
e.RowIsland.Behaviors.Filtering.ColumnFilters.Add(filter);
```

In JavaScript – Root Level

Execute after the grid loads on the page.

```
var whdg = $find("WebHierarchicalDataGrid1");
var topRowIsland = whdg.get_gridView();
```

	<pre> var filtering = topRowIsland.get_behaviors().get_filtering(); var cf = filtering.create_columnFilter("ProductName") cf.get_condition().set_rule(\$IG.TextFilterRules.Contains); cf.get_condition().set_value("al"); filtering.add_columnFilter(cf); </pre>
--	--

行番号

WebDataGrid	WebHierarchicalDataGrid
<p>1. 行セレクターを有効にする</p> <ul style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] を選択します 1.3. [行セレクター] をチェックします <ul style="list-style-type: none"> 1.3.1. RowNumbering を True に設定します。 	<p>← 親グリッドも同様</p> <p>子バンドに動作を追加するには、『WebHierarchicalDataGrid 子バンドに動作を追加する』を参照してください。</p>

行集計

WebDataGrid	WebHierarchicalDataGrid
<p>2. 行セレクターを有効にする</p> <ul style="list-style-type: none"> 2.1. [スマート タグ] を開きます 2.2. [動作の編集] を選択します 2.3. 集計行をチェックします 	<p>← 親グリッドも同様</p> <p>子バンドに動作を追加するには、『WebHierarchicalDataGrid 子バンドに動作を追加する』を参照してください。</p>

選択: セル

WebDataGrid	WebHierarchicalDataGrid
-------------	-------------------------

1. 選択動作を有効にする

- 1.1. [スマート タグ] を開きます
- 1.2. [動作の編集] を選択します
- 1.3. [選択] をチェックします

グリッドで選択が有効になると、ユーザーはサーバーまたはクライアントのセルを選択できます。

サーバー側

選択済みセルの取得

```
SelectedCellCollection selectedCells =
    this.WebDataGrid1.
        Behaviors.
        Selection.
        SelectedCells;
```

セルの選択

```
selectedCells.Add(this.WebDataGrid1.Rows[0].Items[1]);
```

Deselect a Cell

```
selectedCells.RemoveAt(0);
```

クライアント側

選択済みセルの取得

```
var grid = $find('<%= WebDataGrid1.ClientID %>');
```

1. 選択動作を有効にする

- 1.1. [スマート タグ] を開きます
- 1.2. [動作の編集] を選択します
- 1.3. [選択] をチェックします

グリッドで選択が有効になると、ユーザーはサーバーまたはクライアントのセルを選択できます。

サーバー側

列を選択するためのアクセスは、以下のスニペットを

RowIslandDataBound イベントに配置する必要があります。

選択済みセルの取得

```
Infragistics.Web.UI.GridControls.SelectedCellCollection selectedCells;
```

// 親バンドで列を選択します

```
if (e.RowIsland.DataMember == "SqlDataSource1_DefaultView")
{
    selectedCells = e.RowIsland.Behaviors.Selection.SelectedCells;
}
```

// 子バンドで列を選択します

```
if (e.RowIsland.DataMember == "SqlDataSource2_DefaultView"
    && e.RowIsland.ParentRow ==
    this.WebHierarchicalDataGrid1.GridView.Rows[0])
```

```
var selectedCells = grid.get_behaviors().
    get_selection().
    get_selectedCells();
```

セルの選択

```
var grid = $find('<%= WebDataGrid1.ClientID %>');
var cell = grid.get_rows().get_row(0).get_cell(1);
```

```
grid.get_behaviors().
    get_selection().
    get_selectedCells().
    add(cell);
```

Deselect a Cell

```
var grid = $find('<%= WebDataGrid1.ClientID %>');
var cell = grid.get_behaviors().
    get_selection().
    get_selectedCells().
    getItem(0);
```

```
grid.get_behaviors().
    get_selection().
    get_selectedCells().
    remove(cell);
```

```
{
    selectedCells = e.RowIsland.Behaviors.Selection.SelectedCells;
}
```

セルの選択

```
Infragistics.Web.UI.GridControls.Selection selection;
```

// 親バンドで列を選択します

```
if (e.RowIsland.DataMember == "SqlDataSource1_DefaultView")
{
    selection = e.RowIsland.Behaviors.Selection;
    selection.SelectedCells.Add(e.RowIsland.Rows[0].Items[0]);
}
```

// 子バンドで列を選択します

```
if (e.RowIsland.DataMember == "SqlDataSource2_DefaultView"
    && e.RowIsland.ParentRow ==
    this.WebHierarchicalDataGrid1.GridView.Rows[0])
{
    selection = e.RowIsland.Behaviors.Selection;
    selection.SelectedCells.Add(e.RowIsland.Rows[0].Items[0]);
}
```

クライアント側

選択済みセルの取得

```

var grid = $find('<%= WebHierarchicalDataGrid1.ClientID %>');
var parentGrid = grid.get_gridView();
var childGrid = grid.get_gridView().
    get_rows().
    get_row(0).
    get_rowIslands(0)[0];

var selectedCells;

// 親セル選択
if (parentGrid != null) {
    selectedCells = parentGrid.get_behaviors().
        get_selection().
        get_selectedCells();
}

// 子セル選択
if (childGrid != null){
    selectedCells = childGrid.get_behaviors().
        get_selection().
        get_selectedCells();
}

セルの選択

```



```

var grid = $find('<%= WebHierarchicalDataGrid1.ClientID %>');
var parentGrid = grid.get_gridView();
var childGrid = grid.get_gridView().
    get_rows().
    get_row(0).
    get_rowIslands(0)[0];

// 親セル選択
var grid = $find('<%= WebHierarchicalDataGrid1.ClientID %>');
var parentGrid = grid.get_gridView();

if (parentGrid != null) {
    var cell = parentGrid.get_rows().
        get_row(0).
        get_cell(0);

    parentGrid.get_behaviors().
        get_selection().
        get_selectedCells().
        add(cell);
}

// 子セル選択
if (childGrid != null) {

```

```

var cell = childGrid.get_rows().
    get_row(0).
    get_cell(0);

childGrid.get_behaviors().
    get_selection().
    get_selectedCells().
    add(cell);
}

```

選択: 列

WebDataGrid	WebHierarchicalDataGrid
<p>1. 選択動作を有効にする</p> <ul style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] を選択します 1.3. [選択] をチェックします <p>グリッドで選択が有効になると、ユーザーはサーバーまたはクライアントの列を選択できます。</p> <p>サーバー側</p> <p>選択された列を取得します</p> <p><code>SelectedColumnCollection</code> selectedColumns = <code>this.WebDataGrid1.</code></p>	<p>1. 選択動作を有効にする</p> <ul style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] を選択します 1.3. [選択] をチェックします <p>グリッドで選択が有効になると、ユーザーはサーバーまたはクライアントの列を選択できます。</p> <p>サーバー側</p> <p>列を選択するためのアクセスは、以下のスニペットを RowIslandDataBound イベントに配置する必要があります。</p>

```
Behaviors.
Selection.
SelectedColumns;
```

列の選択

```
selectedColumns.Add(this.WebDataGrid1.Columns[0]);
```

列の選択解除

```
selectedColumns.RemoveAt(0);
```

クライアント側

選択された列を取得します

```
var grid = $find('<%= WebDataGrid1.ClientID %>');
var columns = grid.
```

```
    get_behaviors().
    get_selection().
    get_selectedColumns();
```

列の選択

```
var grid = $find('<%= WebDataGrid1.ClientID %>');
var column = grid.get_columns().get_column(0);
```

```
grid.get_behaviors().
    get_selection().
    get_selectedColumns().
```

選択された列を取得します

```
Infragistics.Web.UI.GridControls.SelectedColumnCollection
selectedCols;
```

// 親バンドで列を選択します

```
if (e.RowIsland.DataMember == "SqlDataSource1_DefaultView")
{
    selectedCols = e.RowIsland.Behaviors.Selection.SelectedColumns;
}
```

// 子バンドで列を選択します

```
if (e.RowIsland.DataMember == "SqlDataSource2_DefaultView"
    && e.RowIsland.ParentRow ==
this.WebHierarchicalDataGrid1.GridView.Rows[0])
{
    selectedCols = e.RowIsland.Behaviors.Selection.SelectedColumns;
}
```

列の選択

```
Infragistics.Web.UI.GridControls.Selection selection;
```

// 親バンドで列を選択します

```
if (e.RowIsland.DataMember == "SqlDataSource1_DefaultView")
{
    selection = e.RowIsland.Behaviors.Selection;
```

```
add(column);
```

列の選択解除

```
var grid = $find('<%= WebDataGrid1.ClientID %>');
```

```
var column = grid.
```

```
    get_behaviors().
```

```
    get_selection().
```

```
    get_selectedColumns().
```

```
    getItem(0);
```

```
grid.get_behaviors().
```

```
    get_selection().
```

```
    get_selectedColumns().
```

```
    remove(column);
```

```
selection.SelectedColumns.Add(e.RowIsland.Columns[2]);
```

```
}
```

```
// 子バンドで列を選択します
```

```
if (e.RowIsland.DataMember == "SqlDataSource2_DefaultView"
```

```
    && e.RowIsland.ParentRow ==
```

```
this.WebHierarchicalDataGrid1.GridView.Rows[0])
```

```
{
```

```
    selection = e.RowIsland.Behaviors.Selection;
```

```
    selection.SelectedColumns.Add(e.RowIsland.Columns[2]);
```

```
}
```

列の選択解除

```
Infragistics.Web.UI.GridControls.Selection selection;
```

```
// 親バンドで列の選択を解除します
```

```
if (e.RowIsland.DataMember == "SqlDataSource1_DefaultView")
```

```
{
```

```
    selection = e.RowIsland.Behaviors.Selection;
```

```
    selection.SelectedColumns.Remove(e.RowIsland.Columns[2]);
```

```
}
```

```
// 子バンドで列の選択を解除します
```

```
if (e.RowIsland.DataMember == "SqlDataSource2_DefaultView"
```

```
    && e.RowIsland.ParentRow ==
```

```

this.WebHierarchicalDataGrid1.GridView.Rows[0]
{
    selection = e.RowIsland.Behaviors.Selection;
    selection.SelectedColumns.Remove(e.RowIsland.Columns[2]);
}

```

クライアント側

選択された列を取得します

```

var grid = $find('<%= WebHierarchicalDataGrid1.ClientID %>');
var parentGrid = grid.get_gridView();
var childGrid = grid.get_gridView().
    get_rows().
    get_row(3).
    get_rowIslands(0)[0];

var selectedColumns;

// 親列選択
if (parentGrid != null) {
    selectedColumns = parentGrid.
        get_behaviors().
        get_selection().
        get_selectedColumns();
}

```

```

// 子列選択
if (childGrid != null) {
    selectedColumns = childGrid.get_behaviors().
        get_selection().
        get_selectedColumns();
}

列の選択
var grid = $find('<%= WebHierarchicalDataGrid1.ClientID %>');
var parentGrid = grid.get_gridView();
var childGrid = grid.get_gridView().
    get_rows().
    get_row(3).
    get_rowIslands(0)[0];

var column;

// 親列選択
if (parentGrid!= null){
    column = parentGrid.get_columns().get_column(0);
    parentGrid.
        get_behaviors().
        get_selection().
        get_selectedColumns().
        add(column);
}

```

```
// 子列選択
if (childGrid != null) {
    column = childGrid.get_columns().get_column(0);
    childGrid.get_behaviors().
        get_selection().
        get_selectedColumns().
        add(column);
}
```

選択: 行

WebDataGrid	WebHierarchicalDataGrid
<p>1. 選択動作を有効にする</p> <ul style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] を選択します 1.3. [選択] をチェックします <p>グリッドで選択が有効になると、ユーザーはサーバーまたはクライアントの行を選択できます。</p> <p>サーバー側</p> <p>選択した行の取得</p> <pre>SelectedRowCollection selectedRows = this.WebDataGrid1.</pre>	<p>1. 選択動作を有効にする</p> <ul style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. [動作の編集] を選択します 1.3. [選択] をチェックします <p>グリッドで選択が有効になると、ユーザーはサーバーまたはクライアントの行を選択できます。</p> <p>サーバー側</p> <p>列を選択するためのアクセスは、以下のスニペットを RowIslandDataBound イベントに配置する必要があります。</p>

```
Behaviors.  
Selection.  
SelectedRows;
```

行の選択

```
selectedRows.Add(this.WebDataGrid1.Rows[0]);
```

行の選択解除

```
selectedRows.RemoveAt(0);
```

クライアント側

選択した行の取得

```
var grid = $find('<%= WebDataGrid1.ClientID %>');  
var selectedRows = grid.get_behaviors().  
    get_selection().  
    get_selectedRows();
```

行の選択

```
var grid = $find('<%= WebDataGrid1.ClientID %>');  
var row = grid.get_rows().get_row(0);
```

```
grid.get_behaviors().  
    get_selection().  
    get_selectedRows().  
    add(row);
```

選択した行の取得

```
Infragistics.Web.UI.GridControls.SelectedRowCollection selectedRows;
```

```
// 親バンドで列を選択します
```

```
if (e.RowIsland.DataMember == "SqlDataSource1_DefaultView")  
{  
    selectedRows = e.RowIsland.Behaviors.Selection.SelectedRows;  
}
```

```
// 子バンドで列を選択します
```

```
if (e.RowIsland.DataMember == "SqlDataSource2_DefaultView"  
    && e.RowIsland.ParentRow ==  
this.WebHierarchicalDataGrid1.GridView.Rows[0])  
{  
    selectedRows = e.RowIsland.Behaviors.Selection.SelectedRows;  
}
```

行の選択

```
Infragistics.Web.UI.GridControls.Selection selection;
```

```
// 親バンドで列を選択します
```

```
if (e.RowIsland.DataMember == "SqlDataSource1_DefaultView")  
{  
    selection = e.RowIsland.Behaviors.Selection;  
    selection.SelectedRows.Add(e.RowIsland.Rows[0].Items[0]);  
}
```


行の選択解除

```

var grid = $find('<%= WebDataGrid1.ClientID %>');
var row = grid.
    get_behaviors().
    get_selection().
    get_selectedRows().
    getItem(0);

grid.get_behaviors().
    get_selection().
    get_selectedRows().
    remove(row);
  
```

```

}
  
```

// 子バンドで列を選択します

```

if (e.RowIsland.DataMember == "SqlDataSource2_DefaultView"
    && e.RowIsland.ParentRow ==
this.WebHierarchicalDataGrid1.GridView.Rows[0])
{
    selection = e.RowIsland.Behaviors.Selection;
    selection.SelectedRows.Add(e.RowIsland.Rows[0].Items[0]);
}
  
```

クライアント側

選択した行の取得

```

var grid = $find('<%= WebHierarchicalDataGrid1.ClientID %>');
var parentGrid = grid.get_gridView();
var childGrid = grid.get_gridView().
    get_rows().
    get_row(3).
    get_rowIslands(0)[0];

var selectedRows;

// 親行選択
if (parentGrid != null){
    selectedRows = parentGrid.
        get_behaviors().
  
```

```

        get_selection().
        get_selectedRows();
    }

    // 子行選択
    if (childGrid != null){
        selectedRows = childGrid.
            get_behaviors().
            get_selection().
            get_selectedRows();
    }

    行の選択
    var grid = $find('<%= WebHierarchicalDataGrid1.ClientID %>');
    var parentGrid = grid.get_gridView();
    var childGrid = grid.get_gridView().
        get_rows().
        get_row(3).
        get_rowIslands(0)[0];

    var row;

    // 親行選択
    if (parentGrid != null){
        row = parentGrid.get_rows().get_row(0);
        parentGrid.

```

	<pre> get_behaviors(). get_selection(). get_selectedRows(). add(row); } // 子行選択 if (childGrid != null){ row = childGrid.get_rows().get_row(0); childGrid. get_behaviors(). get_selection(). get_selectedRows(). add(row); } </pre>
--	---

テンプレート化: 列テンプレート

WebDataGrid	WebHierarchicalDataGrid
<p>列テンプレートは、TemplateDataField 列またはグリッド Templates コレクションで定義されたテンプレートを特定の列に割り当てることによるのみ有効です。</p> <p>TemplateDataField の使用</p> <p>1. デザイナーを使用した TemplateDataField の作成</p>	<p>← 同じ</p>

- 1.1. スマートタグを開きます
- 1.2. [列の編集] を選択します
- 1.3. 使用可能なフィールド ペインから **TemplateField** を選択します
- 1.4. [追加] をクリックします
- 1.5. Key を一意な値に設定します
- 1.6. [OK] をクリックします

2. コードビューに切り替えてテンプレート フィールドを設定する

列:

```
<ig:TemplateDataField Key="Template1">
  <ItemTemplate>
    <asp:Button id="btnSelect" Text="Select" runat="server" />
    <%# DataBinder.Eval(
      ((Infragistics.Web.UI.TemplateContainer)Container).
        DataItem,
        "ProductName") %>
  </ItemTemplate>
</ig:TemplateDataField>
</Columns>
```

テンプレート コレクションの使用

1. 新しいグリッド テンプレートの作成

- 1.1. スマートタグを開きます
- 1.2. [テンプレート コレクションの編集] を選択します
- 1.3. [新しい項目の追加] をクリックします

1.4. TemplateID を一意な値に設定します

1.5. [OK] をクリックします

2. コードビューに切り替えてテンプレートを設定する

```
<ig:ItemTemplate ID="WebDataGrid1Template1" runat="server"
  TemplateID="formattedName">
  <Template>
    <div class="productName">
      <%# DataBinder.Eval(
        ((Infragistics.Web.UI.TemplateContainer)Container).
          DataItem,
          "ProductName") %>
    </div>
  テンプレート:
</ig:ItemTemplate>
```

3. InitializeRow イベントのテンプレートを適用する

```
private int NAME_COLUMN_INDEX = 0;

protected void WebDataGrid1_InitializeRow(object sender,
Infragistics.Web.UI.GridControls.RowEventArgs e)
{
  e.Row.Items[NAME_COLUMN_INDEX].TemplateId =
  "formattedName";
```

<p>}</p> <p>注: TemplateDataField が常に優先されるため、TemplateDataField または単一列の Templates コレクションのテンプレートを使用します。</p>	
---	--

テンプレート化: 空行

WebDataGrid	WebHierarchicalDataGrid
<p>1. デザイナーを使用して空行テンプレートを作成する</p> <ol style="list-style-type: none"> 1.1. [スマート タグ] を開きます 1.2. テンプレートの編集を選択します 1.3. 表示ドロップ ダウンから空行テンプレートを選択します 1.4. デザインまたはソース ビューのテンプレートを設定します <p>注: デザイン ビューでテンプレートの設定を終えたら、スマートタグの [テンプレート編集を終了する] をクリックしてデフォルトのデザイナービューへ戻ります。</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>WebDataGrid Tasks Template Editing Mode</p> <p>Display: Empty Rows Template ▼</p> <p>End Template Editing ←</p> </div>	<p>← 同じ</p>

コードで空行のテンプレートを作成する

```
protected override void OnInit(EventArgs e)
{
    base.OnInit(e);
    //テンプレートはPostBackごとにインスタンス化される必要があります。
    this.WebDataGrid1.EmptyRowsTemplate = new
CustomEmptyRowsTemplate();
}

private class CustomEmptyRowsTemplate : ITemplate
{
    public void InstantiateIn(Control container)
    {
        System.Web.UI.WebControls.Label label1 = new
System.Web.UI.WebControls.Label();
        label1.Text = "Empty row template";
        label1.ID = "Label1";

        container.Controls.Add(label1);
    }
}
```

グリッド イベント

WebDataGrid	WebHierarchicalDataGrid	イベント	引数
	X	ContainerGridDataBinding	DataBindingEventArgs
X		CustomDataBinding	DataBindingEventArgs
X	X	DataBinding	EventArgs
X	X	DataBound	EventArgs
X	X	Disposed	EventArgs
	X	GroupedColumnsChanged	GroupedColumnsChangedEventArgs
	X	GroupedColumnsChanging	GroupedColumnsChangingEventArgs
	X	GroupedRowInitialized	GroupedRowEventArgs
X	X	HeaderCheckBoxClicked	HeaderCheckBoxEventArgs
X	X	Init	EventArgs
	X	InitializeBand	BandEventArgs
X	X	InitializeRow	RowEventArgs
X	X	ItemCommand	HandleCommandEventArgs
X	X	Load	EventArgs
X	X	PreRender	EventArgs
	X	RowCollapsed	ContainerRowEventArgs
	X	RowExpanded	ContainerRowEventArgs
	X	RowIslandsCreated	RowIslandEventArgs
	X	RowIslandsDataBinding	RowIslandEventArgs
	X	RowIslandsDataBound	RowIslandEventArgs
	X	RowIslandsPopulated	ContainerRowEventArgs
	X	RowIslandsPopulating	ContainerRowCancelEventArgs
X	X	Unload	EventArgs

各動作のイベント

WebHierarchicalDataGrid は WebDataGrid を拡張するため、イベント引数クラスを共有するイベントがあります。

動作	クライアント イベント (Args)	Arg リンク	サーバー イベント (Args)	Arg リンク
アクティ ブ化	ActiveCellChanged (ActiveCellChangedEventArgs)	WDG WHDG	ActiveCellChanged (ActiveCellEventArgs)	同じ
	ActiveCellChanging (ActiveCellChangingEventArgs)	WDG WHDG	なし	
	なし		ActiveGroupedRowChanged (ActiveGroupedRowEventArgs)	同じ
セルの編 集	EnteredEditMode (EditModeEventArgs)	WDG WHDG	コアの編集機能 イベントを使用	
	EnteringEditMode (CancelEditModeEventArgs)	WDG WHDG		
	ExitedEditMode (EditModeEventArgs)	WDG WHDG		
	ExitingEditMode (CancelEditModeEventArgs)	WDG WHDG		

	Initialize	なし		
Clipboard	Copied	なし	なし	
	Copying			
	Cut			
	Cutting			
	Initialize			
	Pasted			
	Pasting			
列の固定	FixedStateChanged (FixedEvenArgs)	同じ	FixedStateChanged (FixedStateChangedEventArgs)	同じ
	FixedStateChanging (FixingEventArgs)	同じ	なし	
	Initialize	なし		
列の移動	HeaderDragEnd (HeaderDragEndEventArgs)	同じ	なし	
	HeaderDragStart (HeaderDragStart)	同じ		
	HeaderDropped (HeaderDroppedEventArgs)	同じ		
	HeaderMove (HeaderMoveEventArgs)	同じ		
	Initialize	なし		
	なし		ColumnMoved (ColumnMovingEventArgs)	同じ
列サイズ	ColumnResized (ColumnResizedEventArgs)	同じ	なし	

変更	ColumnResizeDragging (ColumnResizeDraggingEventArgs)	同じ		
	ColumnResizing (ColumnResizingEventArgs)	同じ		
	Initialize	なし	ColumnResized (ColumnResizingEventArgs)	同じ
コアの編集	RowAdded (RowAddedEventArgs)	WDG WHDG	RowAdded (RowAddedEventArgs)	WDG WHDG
	RowAdding	なし	RowAdding (RowAddingEventArgs)	同じ
	RowDeleted	なし	RowDeleted (RowDeletedEventArgs)	同じ
	RowDeleting	なし	RowDeleting (RowDeletingEventArgs)	同じ
	RowUpdated (RowUpdatedEventArgs)	WDG WHDG	RowUpdated (RowUpdatedEventArgs)	同じ
	RowUpdating	なし	RowUpdating (RowUpdatingEventArgs)	WDG WHDG
	CellValueChanged (CellValueChangedEventArgs)	WDG WHDG	なし	
	CellValueChanging (CancelCellValueChangingEventArgs)	WDG WHDG		
	Initialize	なし		

フィルタ ー	DataFiltered (DataFilteredArgs)	同じ	DataFiltered (FilteredEventArgs)	同じ		
	DataFiltering (CancelApplyFiltersEventArgs)	同じ	DataFiltering (FilteringEventArgs)	同じ		
	ColumnFilterAdded (ColumnFilterAddedArgs)	同じ	なし			
	EnteredEditMode (EditModeEventArgs)	WDG WHDG				
	EnteringEditMode (CancelEditModeEventArgs)	WDG WHDG				
	ExitedEditMode (EditModeEventArgs)	WDG WHDG				
	ExitingEditMode (CancelEditModeEventArgs)	WDG WHDG				
	FilterDropDownDisplayed FilterDropDownDisplaying FilterDropDownHidden FilterDropDownHiding Initialize	なし				
ページン グ	Initialize	なし			なし	
	PageIndexChanging (PagerEventArgs)	WDG WHDG				

	PageIndexChanged	なし	PageIndexChanged (PagingEventArgs)	同じ
行の追加	EnteredEditMode (EditModeEventArgs)	WDG WHDG	なし	
	EnteringEditMode (CancelEditModeEventArgs)	WDG WHDG		
	ExitedEditMode (EditModeEventArgs)	WDG WHDG		
	ExitingEditMode (CancelEditModeEventArgs)	WDG WHDG		
	Initialize	なし	編集コア イベントを使用	
行の削除	なし		編集コア イベントを使用	
行編集テンプレート	Initialize	なし	なし	
	TemplateClosed (EditRowEventArgs)	WDG WHDG		
	TemplateClosing (CancelEditRowEventArgs)	WDG WHDG		
	TemplateOpened (EditRowEventArgs)	WDG WHDG		
	TemplateOpening (CancelEditRowEventArgs)	WDG	コアの編集機能 イベントを使用	

		WHDG	
行セレクト	FooterRowSelectorClicked (MarginRowSelectorClickedEventArgs)	同じ	なし
	HeaderRowSelectorClicked (MarginRowSelectorClickedEventArgs)	同じ	
	Initialize	なし	
	RowSelectorClicked (RowSelectorClickedEventArgs)	同じ	
	RowSelectorClicking (RowSelectorClickingEventArgs)	同じ	
選択	CellSelectionChanging (CellSelectionChangingEventArgs)	WDG WHDG	なし
	ColumnSelectionChanging (ColumnSelectionChangingEventArgs)	WDG WHDG	
	Initialize	なし	
	RowSelectionChanging (RowSelectionChangingEventArgs)	WDG WHDG	

	CellSelectionChanged (CellSelectionChangedEventArgs)	WDG WHDG	CellSelectionChanged (SelectedCellEventArgs)	同じ
	ColumnSelectionChanged (ColumnSelectionChangedEventArgs)	WDG WHDG	ColumnSelectionChanged (SelectedColumnEventArgs)	同じ
	RowSelectionChanged (RowSelectionChangedEventArgs)	WDG WHDG	RowSelectionChanged (SelectedRowEventArgs)	同じ
並べ替え	ColumnSorting (SortingEventArgs)	WDG WHDG	なし	
	Initialize	なし		
	ColumnSorted (SortedEventArgs)	WDG WHDG	ColumnSorted (SortingEventArgs)	WDG WHDG
集計行	Initialize	なし	CalculateSummary (CustomSummaryEventArgs)	同じ
	SummaryCalculated SummaryDropDownDisplayed SummaryDropDownDisplaying SummaryDropDownHidden SummaryDropDownHiding		SummaryRowCalculated (SummaryEventArgs)	同じ
	CalculateCustomSummary		CalculateCustomSummary (CustomSummaryEventArgs)	同じ
バーチャ	FormatToolTip	同じ	なし	

ルスクロ ール	(VirtualScrollingFormatToolTipEventArgs)		
	Initialize	なし	
	MoreRowsReceived (MoreRowsRecievedEventArgs)	同じ	
	MoreRowsRequesting (CancelMoreRowsRequestingEventArgs)	同じ	

改訂履歴

- 2011/07/08:
 - スニペット セクションの追加:
 - AJAX ロード オン デマンド - (手動)
 - 編集: 行の追加
 - 列サマリー
 - テーブルの追加
 - グリッド イベント
 - 各動作のイベント