WebGrid.NET Enterprise 6.0 White Paper

Classic Paging

Overview

WebGrid.NET Enterprise is famous with its VirtualLoad™ paging mode since its first release in year 2003. In its sixth generation, WebGrid.NET Enterprise now comes with traditional (classic) paging in addition to the VirtualLoad paging mode.

Due to popular demand, classic paging is finally here being one of the most wanted features in WebGrid.NET Enterprise. Although VirtualLoad paging is best suitable for modern Web application, many end users still prefer to use classic paging mode. Classic paging mode is defined by a set of navigation controls (such as First, Previous, Next and Last) to control the active page. In classic paging, the total number of visible rows is a constant number (eg, 25).

To enable classic paging, simply set *PagingMode* to *ClassicPaging*. You can customize the number of visible rows in one page by setting the *PagingSize* property. By default, the *PagingStyleUI* is set to *FirstPrevNextLast* style, which will display a set of paging commands in the status bar.

The default PagingStyleUI provides five basic user interfaces for paging purpose, which is located at status bar area:

- First command. Move data view to the first page.
- Previous command. Move data view to the previous page.
- Next command. Move data view to the next page.
- Last command. Move data view to the last page.
- Goto command. Display available pages in context menu for direct paging. To try this functionality, click on the dropdown arrow besides the page status text.
- Page Status. Display the current paging status.

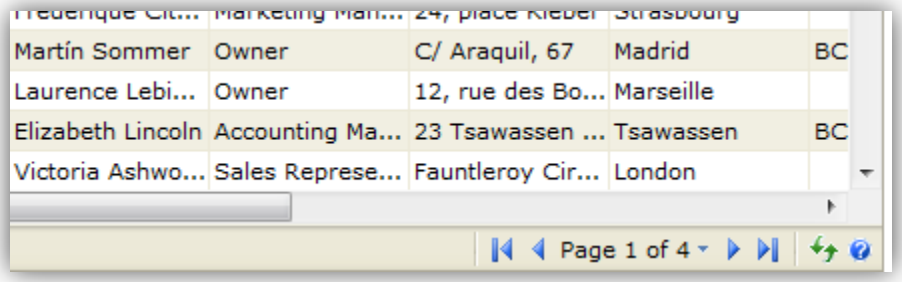
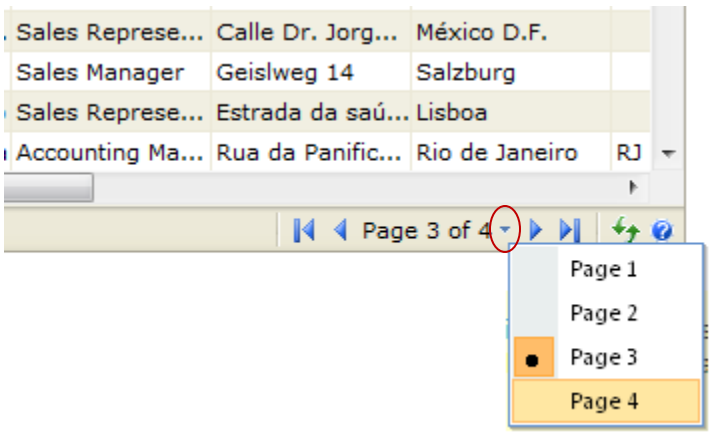


Figure 1. Classic Paging feature in WebGrid with default paging UI and commands

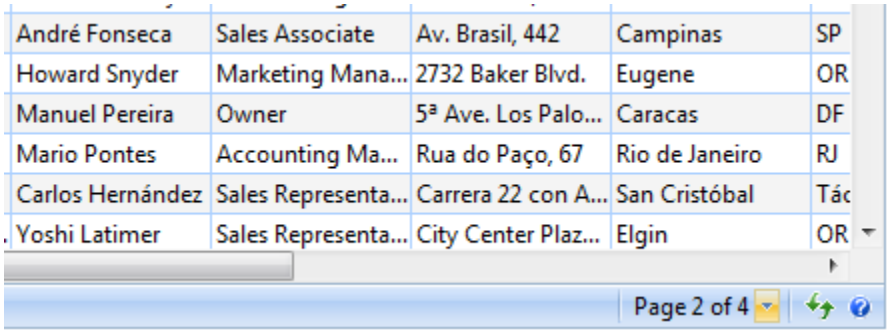
Paging Styles UI

WebGrid.NET Enterprise 6.0 provides three built-in user interface styles for the classic paging feature. The paging styles are:

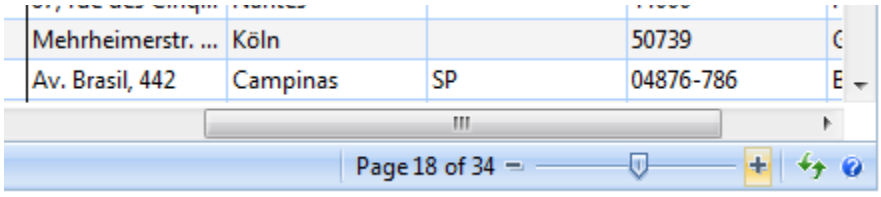
- *FirstPrevNextLast* style. This is the default value of paging style UI. This style includes the basic commands for navigating between pages. It also includes a handy “Jump To” interface accessible through dropdown menu.



- *SimpleDropDown* style. In the scenario where the screen real estate is limited, this paging style is the most recommended, as it includes only basic navigation commands presented in simple and minimalistic design.



- *Slider* style. This new innovative style resembles modern slider bar for intuitive navigation in minimalistic fashion. This style is perfect for every rich Web application that demand sophisticated user experience.



The *Slider* style paging UI introduces a new user experience to navigate between long pages. Instead of showing numerous different paging commands, Slider style sports a sleek, minimalist and visually compelling user interface in the status bar to let users perform data navigation in a smart and smoother experience.

The Slider User Interface consists of several elements:

- Minus command. This command instructs WebGrid to show the previous page view.
- Slider Track. You can click on the slider track to quickly forward the pages. When you hold down the left mouse button for more than 1 second, it will smoothly move toward the current mouse cursor.
- Slider Thumb. Similar to the scrollbar behavior in Windows®, you can drag the slider thumb to the left or right direction and release mouse button as you reach destination page. While you drag the slider thumb, a handy tooltip will appear to indicate the page number where the scroll thumb is located.
- Plus command. This command instructs WebGrid to show the next page view.

Thanks to the new Classic Paging in WebGrid.NET Enterprise™ 6.0, now you can combine it together with LiveFreeze™ column freezing, grouping, filtering, editing and other features to create high performance and effective, thin-client data-driven Web application.

There are two important properties related to Slider paging style UI:

- *PagingLatencyOnSlide*. When you drag the slider thumb to change the active page index, WebGrid does not perform AJAX callback immediately. Instead, it will wait for several milliseconds as specified in *PagingLatencyOnSlide* property to avoid “bogus callbacks”. This feature improves user experience and increasing efficiency by invoking paging after end user has stop sliding for several milliseconds.

- *PagingSliderWidth*. By default, the slider bar width is set to 100px. You can customize the slider bar’s width according to the screen real estate of your Web page. For instance, in a Web page which have only one WebGrid as the data presenter, you may want to have larger slider bar (such as 300px) to allow your users to conveniently scroll through long pages.

Custom Paging

Similar to *Custom Paging* provided in VirtualLoad™ paging feature, the classic paging also allow you to perform custom data retrieval for paging. By default, the classic paging will use *Automatic* data retrieval mode. In automatic mode, WebGrid determines the total datasource rows based on the given datasource and display the current page appropriately.

Oftentimes, you need to optimize load performance so that only the active page view rows are retrieved at once. This is also known as load on demand technique. WebGrid.NET Enterprise™ 6.0 allows you to perform custom data retrieval for classic paging by setting *PagingLoadMode* to *Custom*.

In this release, WebGrid allows you to implement custom paging through *datasource control*. There are two data source controls that support native custom paging:

- *ObjectDataSource*. This datasource control comes with ASP.NET 2.0.
- *ISDataSource*. This is a much advanced and superior datasource control included with WebGrid.NET Enterprise™ 5.0 or later. To learn more about ISDataSource, please visit Intersoft DataSource Control Website.

In general, custom paging implementation requires three important steps:

1. Set *EnablePaging* of the datasource control to True.
2. Provides a method for *Select*. This method requires three parameters: startRow, maximumRows, and sortExpression.
3. Provides a method for *SelectCount*.

The heart of the custom paging is to provide custom logic for retrieving rows according to the current view of the WebGrid. This is done via custom codes in the Select method. The following is a C# sample for select method.

```
namespace dsNorthwindTableAdapters
{

    public partial class OrdersTableAdapter : System.ComponentModel.Component
    {
        public DataTable GetData(int startRowIndex, int maximumRows, string sortExpression)
        {
            dsNorthwind.OrdersDataTable dt = new dsNorthwind.OrdersDataTable();

            int topRows = startRowIndex + maximumRows;

            if (!string.IsNullOrEmpty(sortExpression))
                sortExpression = " order by " + sortExpression;

            OleDbDataAdapter adapter = new OleDbDataAdapter();
            adapter.SelectCommand = new OleDbCommand("SELECT TOP " + topRows.ToString() +
                " * FROM Orders" + sortExpression, this.Connection);
            adapter.Fill(dt);

            return dt;
        }
    }
}
```

Next, the select count method is required to pass the total number of data rows in the table in order for WebGrid to display the total pages. The following is a C# sample for select count method.

```
namespace dsNorthwindTableAdapters
{

    public partial class OrdersTableAdapter : System.ComponentModel.Component
    {
        public int GetCount(string sortExpression)
        {
            OleDbDataAdapter adapter = new OleDbDataAdapter();
            adapter.SelectCommand = new OleDbCommand(
                "Select COUNT(*) From Orders", this.Connection);

            this.Connection.Open();
            int result = (int)adapter.SelectCommand.ExecuteScalar();
            this.Connection.Close();

            return result;
        }
    }
}
```

Walkthrough: Implementing Custom Paging using ObjectDataSource

[TBD]

Walkthrough: Implementing Custom Paging using ISDataSource

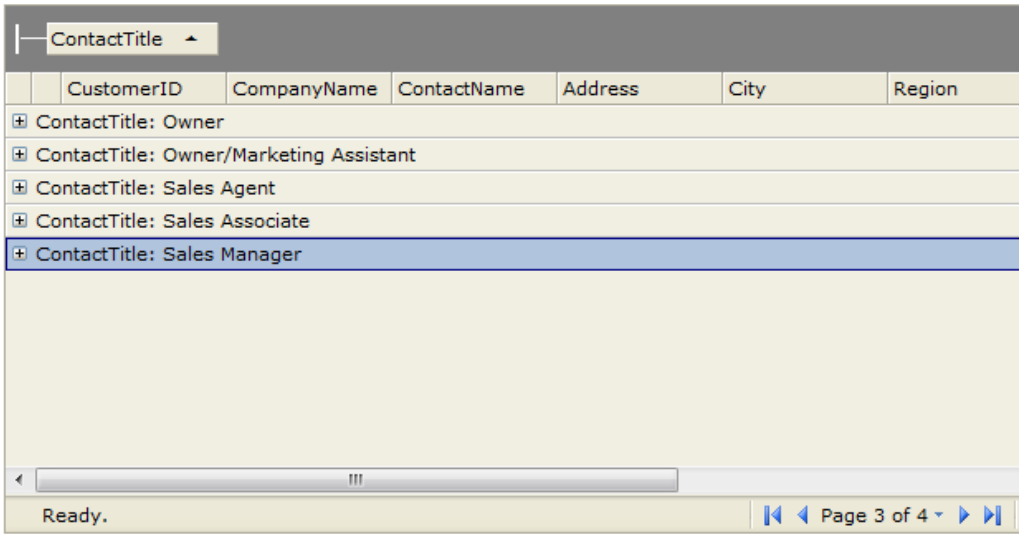
[TBD]

Extended Behaviors

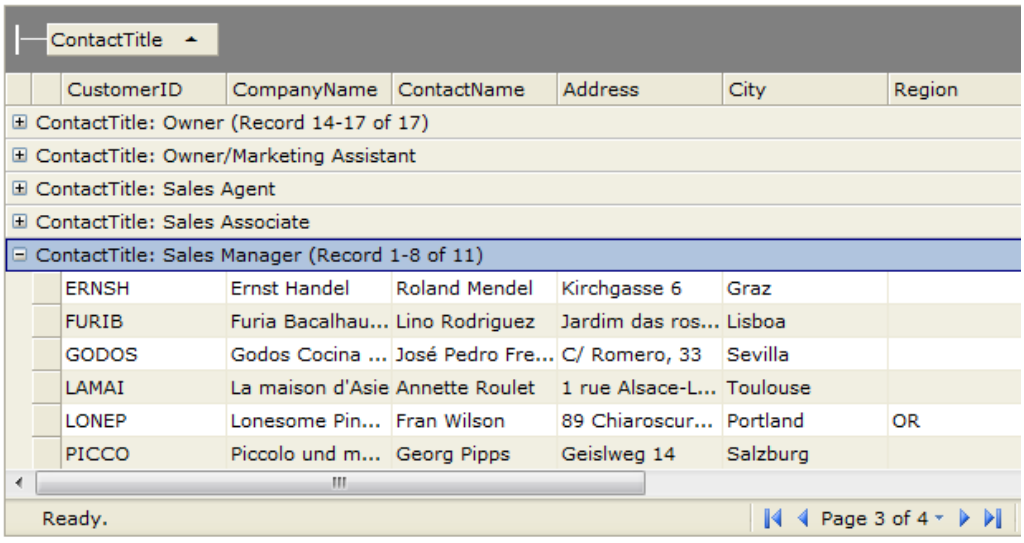
With the introduction of classic paging in version 6.0, WebGrid includes extended behavior when used together with *Grouping* feature. When the grouping is enabled and classic paging is used, WebGrid will display the grouped rows of the respective rows in the active page.

Unlike the flat view, grouped rows in paged view may include only partial rows. This may cause misunderstanding on viewer side especially when group total or column aggregation is enabled. For this purpose, WebGrid includes a special paging feature for grouping purpose called *PagingDetectPartialGroupRows*. By default, this feature is not turned on.

The following screenshot illustrates a page view when Grouping is enabled and *PagingDetectPartialGroupRows* is set to False.



The following screenshot illustrates a page view with *PagingDetectPartialGroupRows* enabled.



As seen in the two illustrations above, the first WebGrid simply shows the group rows of the active page index, without telling the readers whether the group rows are complete. For instance, the Owner group actually contains 17 rows but only 4 rows are displayed due to the paging (called partial grouped rows). The second WebGrid appropriately tells readers that the Owner group consisted of partial rows (eg, Record 14-17 of 17).

The *PagingDetectPartialGroupRows* feature automatically detects possible partial grouped rows of first and last row of the active page view. By using advanced delta mechanism, this feature doesn't sacrifice performance as it doesn't use multiple-loop detection mechanism.

However, please note that this feature doesn't take affect when Custom Paging is enabled. The main reason is because Custom Paging returns only current paged view records which do not include the previous or next records of the table.

Classic Paging with Other Features

As Classic Paging feature is implemented lately in the sixth generation while WebGrid already have hundreds of rich and combinable features in place, the Classic Paging feature should support existing features and non-conflict scenarios without additional workarounds or efforts.

The Classic Paging feature has been extensively tested to meet the high quality and product standards by providing compatibility when used together with the following key features/scenarios:

- Flat Grid
- Flat with Custom Paging
- Grouping
- Grouping with Partial Row Detection
- Multiple Grouping
- Sorting
- Multiple Sorting + Grouping
- Filtering
- Refresh / RefreshAll
- Add
- Add + Group
- Edit
- Edit + Group / + Sort
- Delete
- Delete + Group / + Sort
- Self Referencing
- Self Referencing + Load On Demand
- Self Referencing + Load On Demand + Grouping + DetectPartialGroupRows
- Column Set

- Preview Row
- Hierarchical with multiple child tables
- Hierarchical + (Add/Edit/Delete)
- Group Total
- Column Total
- Group + Hierarchical
- Column Freezing
- Exporting

Important notes:

- In non self-referencing mode and paging load mode is Automatic, the paging will be applied directly during the databinding process. This ensures high performance and scalability over large datasource.
- When used together with self referencing in preload mode (load-on-demand is off), the paging will be applied after the initialization of self referencing. The paging size will be applied to the root rows, excluding the child rows. This behavior ensures effective navigation between pages of the available root rows.
- When self referencing is enabled and grouping is enabled, you can still enable *PagingDetectPartialGroupRows* feature to deliver better record information to user. For more information about partial group rows detection feature, see Extended Behaviors

Customizing Texts and Images

The classic paging feature introduces several new text items to the localizable language files. The following shows the list of added text items related to classic paging feature:

- Under CommonText
  - PagingLoading
  - PagingStatus
  - PagingPartialGroup
- Under Tooltip
  - PagingFirst
  - PagingPrev
  - PagingNext
  - PagingLast
  - PagingGoTo
  - PagingSlider
- Under Paging
  - PagingFormat

To learn more about localization feature and how to customize text settings in WebGrid, see [LINK: Localization Feature]

In addition to new text items, you can also customize the images used by the classic paging feature. The following shows the list of new image settings related to classic paging feature:

- Under ImageSettings
    - PagingMoveFirst
    - PagingMovePrevious
    - PagingMoveNext
    - PagingMoveLast
    - PagingDropdown
    - PagingSliderMinus
    - PagingSliderPlus
    - SliderThumbImage

The image files for the classic paging feature is available in both physical files and SmartWebResources™ assembly. For hassles-free development and deployment, it is highly recommended to enable SmartWebResources™ feature. To learn more SmartWebResources, please refer to [LINK: Deployment Guide].

Client Side References

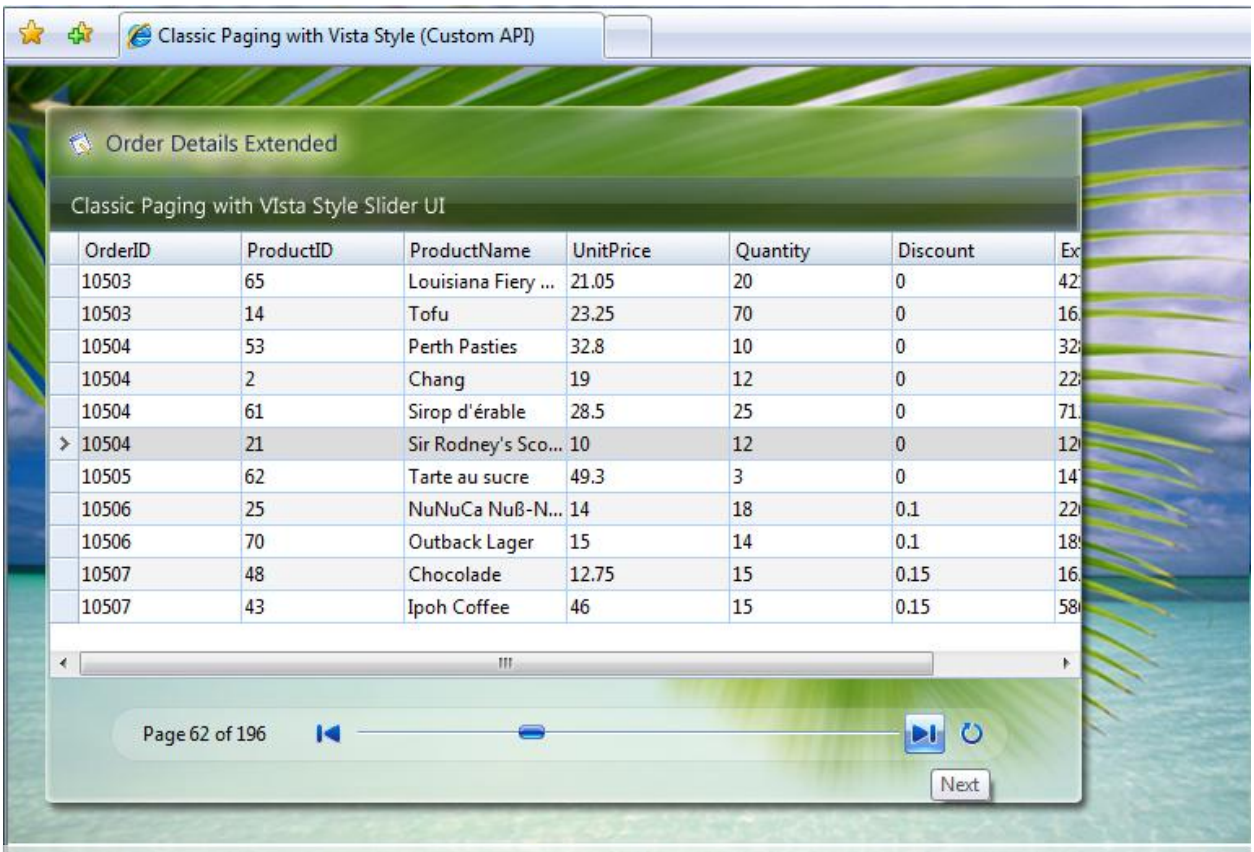
The Classic Paging feature in WebGrid.NET Enterprise® 6.0 is built upon rock-solid, robust object model and API. This enables developers to create their own user interface and perform WebGrid's paging functionality by accessing the provided API.

The following is the list of new methods/API related to classic paging:

- WebGrid class.
    - IsClassicPaging()
    - GetCurrentPage()
    - GetTotalPages()
    - GotoFirstPage()
    - GotoLastPage()
    - GotoPreviousPage()
    - GotoNextPage()
    - GotoPage(targetPageNumber)
    - CreatePagingSlider(containerId, sliderId)
    - SynchronizeSliderUI()

For the complete client side object model and API list, please see [LINK: Client Side References].

For an example that demonstrates classic paging API, please refer to *Classic paging with Vista Style* sample in **Integrated Samples**. The following is the screenshot of *Classic paging with Vista Style* sample.


Classic Paging with Vista Style (Custom API)
Order Details Extended
Classic Paging with VIsta Style Slider UI
OrderID ProductID ProductName UnitPrice Quantity Discount
10503 65 Louisiana Fiery ... 21.05 20 0
10503 14 Tofu 23.25 70 0
10504 53 Perth Pasties 32.8 10 0
10504 2 Chang 19 12 0
10504 61 Sirop d'érable 28.5 25 0
10504 21 Sir Rodney's Sco... 10 12 0
10505 62 Tarte au sucre 49.3 3 0
10506 25 NuNuCa Nuß-N... 14 18 0.1
10506 70 Outback Lager 15 14 0.1
10507 48 Chocolade 12.75 15 0.15
10507 43 Ipoh Coffee 46 15 0.15
Page 62 of 196
Next