What you need to know when selecting a grid

By Daniel Jebaraj, Vice President
Syncfusion, Inc.

Types of grids

One of the most common misconceptions is the notion that a grid is a grid. In reality, there are several different types of grids and choosing the right one for your needs can help save substantial time and effort. Choosing the wrong grid can result in tremendous effort to make it work the way you need, which is quite frustrating. We will look at the different types of grids and the ways in which they are typically used.

1. Cell-oriented grid

- **No assumptions on layout**–Cell-oriented grids do not make any assumptions on the layout of data. They work very well when you wish to have complete control over the layout. They allow you to display data that flows over multiple cells and embed controls such as charts that occupy several underlying cells. You can mix different kinds of cells in the same column.
- **Excel-like behavior**–Each cell is independent of other cells. Commonly, you're able to set 50 differrent attributes on each cell.
- **Formula support**–Cell-oriented grids typically offer support for cell-level formulas, similar to Excel.
- **No assumptions on the source of data**–Cell oriented grids do not expect data to be in a certain format. They allow data to be provided on demand from any source. Binding to data usually involves the implementation of simple callbacks.

2. Data-bound grid controls

- **Homogenous columns**–Data-bound grid controls assume that data in a column will be the same type. They work best when most of your data comes from a straight tabular data source and can be displayed in the same layout. They do not work well if you wish to have more control over the layout of your display.
- **Binding to standard types**–Data-bound grid controls can be bound to any standard data source with a few lines of code.
- **Rich metadata**–Data-bound grids utilize type metadata and other information available from data-binding interfaces. Operations such as sorting, filtering, and grouping are also easier to implement with data-bound grids because of this rich metadata.
- **Support for editing**–Support for editing is implemented in most data-bound grid controls. If the backing data store supports editing, adding new rows, and deleting current rows, such support will automatically become available in data-bound grids.
- **Support for automatic updates**–Data-bound grid controls can automatically display changes to the data source to which they are bound, provided the data source implements appropriate interfaces.
- **Business objects**–Data-bound grid controls support displaying data in business objects, provided such data will support one of the commonly used .NET data-binding interfaces.
- **Delegating common operations to the server**–It is typically desirable to delegate such operations to the server. This is the default behavior with most controls.
- **Support for expressions or formulas**–Data-bound grid controls do not typically support cell-level formula calculations (like Excel). They instead offer unbound columns where simple expressions may be used to calculate the displayed value. If you need Excel-like formulas, then a cell-oriented grid is a better choice.
- **Displaying related data**–Data-bound grids that support displaying related information display such data inline. This is referred to as a hierarchical display or nested table. It is possible for a hierarchical grid to support editing and differing levels in nested tables.
- **Displaying foreign key references**–When you have a foreign key relationship, data-bound grid controls should easily display values from the related table.
- **Displaying grouped data**–Grouping classifies a list of data based on one or more fields. Data-bound grids can support grouping, along with custom summaries and data updates with multiple field groupings.

3. Pivot grids

Pivot grids are very powerful and allow for the display and analysis of massive amounts of data in a summarized, condensed format. Please be sure to test with at least five times the data you expect to work with (both rows and columns). Also, test with several grouping levels on both the row and column axes.

4. Tree grids

Tree grids resemble tree controls, but instead of displaying just one column of information as with a typical tree, they display additional attributes as additional columns. Test for editing and load-on-demand support.

Features of a grid ➤

Features of a grid

Virtualization

Virtualization is another critical aspect to consider when selecting a grid control. There are two kinds of virtualizations possible with grid controls on the WPF and Silverlight platforms.

User interface virtualization

The .NET framework supports what is known as UI virtualization for ItemsControls. Most controls available on the market support UI virtualization because it is essentially implemented by the underlying Microsoft framework.

Test for the following scenarios:

- **Large window size**–Maximize the grid control and observe the impact on performance.
- **Multiple monitors**–This is an extension of the large-window-size test. Test with multiple monitors running several windows with the grid displayed.
- **Grouping and hierarchical display**–Test performance when groups and nested controls consist of more than a few thousand rows and columns.
- **Touch screens**–As of .NET 4, framework UI virtualization does not work well with touch-screen systems. Please ensure that you test with such systems if this is a deployment target for you.
- **Variable item heights**–If you need to display items of different heights, please test with this scenario.
- **Data virtualization**–UI virtualization does not help to minimize the need for actual displayed data to be kept in memory. It is possible to get around this requirement by implementing a virtualized list that sits on top of the underlying data. This approach works for WPF controls but does not work with Silverlight at this time.

Cell architecture and user interaction

Grid controls allow for the display of several kinds of cells. When testing performance, be sure to test with the actual cell types that you will be using and not just text boxes or static text.

Validation

Ensure that entry-time validation is enforced by cell editors and that support for IDataErrorInfo by data sources is implemented.

Selection behavior

Controls should allow the selection of rows, columns, and arbitrary collections of cells. If your users are used to Microsoft Excel, this is one area where behavior that closely mimics Excel may be very desirable.

Clipboard operations

Controls should allow clipboard operations with rows, columns, and arbitrary collections of cells. If your users are familiar with Microsoft Excel, this is yet another area where behavior that closely mimics Excel may be very desirable. Also evaluate the copy-paste behavior of the control with Microsoft Excel. Do you need rich text formatting (perhaps formulas), or is plain text enough?

Frozen rows and columns

Frozen rows and columns are typically positioned at the top, bottom, left, or right of a grid control. It is important to test if multiple rows and columns can be frozen.

Appearance customization

Customization should be possible using themes. Bonus points if they do, since this work can be quite daunting if approached without direct support from the vendor.

Importing and exporting data

- **Exporting**–Exporting to Excel is usually supported by grid controls. Please be sure to test this well. PDF and Word exporting are two additional export options that may be of interest to you.
- **Importing**–There are two kinds of importing from Excel. One is complete import, where the entire file is opened in the grid, and the second is selective import, importing just a section of data.

Globalization

- **Right-to-left support**–Right-to-left support is essential if you plan to support Hebrew and Arabic markets.
- **Printing**–Check for print-preview and printing support.
- **Automated testing**–Automated testing is typically implemented using HP Quick Test Professional or Microsoft's Coded UI support in Visual Studio.NET 2010. Check with the control vendor to ensure that they support test automation issues as part of their support system.

Syncfusion®
Deliver innovation with ease®