Hot Off the Grid-dle!

By Jerry Karasz *BASIS Advantage 2023*

Big news! A BBjGrid control for the DWC!

In early announcements about the new Dynamic Web Client (DWC), BASIS stated that the existing BBjGrid controls (the BBjStandardGrid, the BBjDataAwareGrid, and the BBjDataBoundGrid) would not be available in the DWC. Instead, mobile and responsive layout applications in the DWC would use the newer BBjGridExWidget, an industry-leading native JavaScript grid control also available for GUI and BUI clients. And for new applications developed for the DWC, the BBjGridExWidget is definitely the better choice — it offers a much larger set of end-user features (including grouping, column reordering, drill-down capabilities, and better performance with large data sets, just to name a few).

However, taking that approach leaves legacy applications that depend on the older BBjGrids with a dilemma. Either:

1. stay with the GUI or BUI model, and put off moving to the new DWC model, or
2. undertake a legacy migration effort to recode "how the grid is used" to replace the old grid and follow the BBjGridExWidget's RecordSet retrieval model. For this option, the BBjGridExWidget is the path forward.

Several BASIS customers have already taken option #2 and updated their code, and these early reports indicate that the changes, although not totally trivial, are relatively straightforward to understand and implement.

But even at that, these options are barriers to some customers' rapid adoption of the DWC. So BASIS found a third alternative:

3. a limited BBjGrid implementation that runs in the DWC. This subset of the BBjGrid API is available in the DWC and is sufficient for the majority of grids used in today's GUI and BUI applications. See what features are implemented below.

Bonus!

As a bonus, with BDT version 22.11, the Eclipse plug-in now offers a toolbar button to run your BBj program as a DWC application. With the click of the corresponding toolbar button, you can launch any BBj program in GUI, BUI, or DWC! Development has never been easier!

Why Are You Still Reading?

Let me guess — now you want to know what constitutes a "reasonable subset" of the original BBjStandardGrid API, don't you? Well, even though capturing that information could get tedious, TeamBBj is up to the task. Read on!

Rather than going over numerous API methods and stating which are or are not implemented for the DWC's BBjStandardGrid, let's instead talk about the functionality in broad terms.

How You Can Tell for Yourself

But if you just have to know about a specific API method, then write yourself a test BBj program, run it from inside Eclipse/BDT, and check the browser's console for "unimplemented" output. Here's how:

In Chrome, for example, once your DWC application is running in the active tab, click on the "Customize and configure Google Chrome" button in the upper right corner of the browser. This is the "three vertical dots" button. From there, select **More tools > Developer tools** and examine the Console portion of the display for messages about unimplemented features similar to these:

- Unimplemented GridSetValueMessage MASK_ROUNDING
- Unimplemented GridSetValueMessage ROW_INSERTED

What is Implemented in BBj 22.10+ versions

We didn't stop with BBj 22.10, the following "core" grid functions are implemented in BBj 22.10 and later versions of BBj for BBjStandardGrids in DWC:

BBj 22.10+

- Count information (num rows or columns)
- Cell formatting properties, including text, font, color, and alignment
- Row, column, and cell selection
- Row and column formatting, including alignment
- Row and column header core functionality (width, height, text, color)
- Cell formatting, including colors
- Clearing rows, columns, and cells
- Programmatic start and end editing
- Scroll bars
- User action events (selection, focus, mouse, scroll)

BBj 22.12+

- Numeric and string cell masking (BBjInputE and BBjInputN masks)

BBj 22.13+

- Support for most cell styles, with the exception of list controls

- Cell tooltip text
- User-initiated and programmatic grid cell editing, including INPUTE, INPUTN, CHECKBOX, and PASSWORD style cells

What is Not Implemented

Primarily based on cost/benefit estimates, viability and meeting our goal of delivering a simplified performant version of the BBjGrid for browser/web apps, there are features that we are unlikely to implement in DWC unless they're trivial. As an example, cell validation requires client-server round trips. As such, it is not architecturally viable in DWC, especially from a user experience perspective.

Here is a list of the top feature differences:

- Sortable rows or columns
- User-resizable rows or columns
- Cell query buttons (also not available in BUI).
- Cell validation (also not available in BUI).
- Cell drag-and-drop
- Row and column gutters
- Cell content transformations (upper case, lower case, spaces) on saving
- Grid scrolling behavior options (paging, scroll outside, scroll updates, and so on)

What Can Be Implemented if There is a Demand for it

The following grid functions appear to be implementable but are not considered "core" functionality. They can be considered for implementation for BBjStandardGrids in DWC when and if there is a demonstrated need for them in DWC applications:

- Row manipulation functions (add, delete, hide, unhide)
- Column manipulation functions (add, delete, hide, unhide)
- Cell properties
- Row visibility (isVisible, scroll to make visible)
- Row and column header extended support (icon, image, mask)

Give it a Try

With this new grid implementation, moving to the DWC is easier than ever. But don’t take our word for it — try it for yourself. Launch your existing grid-based programs in the DWC so you can be the judge. Of course, the BBjGrid controls do not provide the same feature set in the DWC that you can get with the BBjGridExWidget, but this version of the BBjGrid can be your first step on a journey into the future with web components and dynamic UIs. But whatever you do, be sure to share your DWC adoption progress with us.