Kendo UI® Builder by Progress®:
Using Kendo UI Designer

Copyright

Table of Contents

Contents

1

Introduction to the Kendo UI Designer

The Kendo UI Designer is part of the Kendo UI® Builder by Progress®, an application that facilitates modernizing OpenEdge applications by creating web-based UIs. You can use the Designer to design the UIs for your application using a collection of predefined, data-driven views, or you can create custom layouts by dragging and dropping components into a user-defined Blank view. For more information about the entire Kendo UI Builder package, see Kendo UI Builder by Progress: Modernizing OpenEdge Applications. For a sample workflow for modernizing an application, see Kendo UI Builder by Progress: Sample Workflow.

When you first open the Kendo UI Designer, you are presented with a dashboard containing a menu and breadcrumbs bar, and displaying cards for any web apps you have previously created, as well as options to:

- Create a new app (**Create App**). This opens the app design page for the new app. For more information, see Creating an app: first step on page 12.

- Import an existing app, possibly from a prior release (**Import App**). The first time you open the Designer in the current release, if you had existing apps on the Designer dashboard in a prior release, or when you click **Import App** to import an external app, the Designer attempts to import these apps and displays a card for them on the dashboard. Each app card also prompts with one of the following app states, which determines action options that are available for the app (see the app action options below):

  - **UpToDate**: There are no issues with the app, and it is imported without the need for migration.

  - **Migration Required**: There is a difference between the app meta data version and the Kendo UI Builder meta data version, but the Designer **can** automatically migrate the app to the Kendo UI Builder meta data format.

  - **Incompatible**: There is a difference between the app meta data version and the Kendo UI Builder meta data version, and the Designer **cannot** automatically migrate the app to the Kendo UI Builder meta data format. You must either migrate the app and its meta data manually or recreate it in the Designer.

  - **Unavailable**: App meta data **cannot** be loaded for some other reason. You must either migrate the app manually or recreate it in the Designer.

- Sort apps alphabetically (**Sort**).

- Search apps by keyword in a search box.

- Display existing apps in a card (**Cards**) or list (**List**) view.

- Edit an existing app by clicking its card or list item. This opens the app design page for the existing app.

- Depending on the app state, duplicate, migrate, remove, or delete an existing app by clicking the gear icon in its card or list item. The following action options appear on the gear menu, based on the app state noted above:

Table 1: App action options based on app state

App state	Gear menu action options
UpToDate	Duplicate Remove Delete
Migration Required	Duplicate Migrate Remove Delete
Incompatible	Remove Delete
Unavailable	Remove

Each menu option works as follows:

- **Duplicate**: Duplicates the existing app with a new app name and possibly in a different location.

- **Migrate**: Migrates the existing app. In order to migrate an existing app, the application folder you want to migrate must contain `app` and `meta` folders from the original app.

Note: You might also need to manually move custom event function code from a prior location to the currently supported location. For example, in releases from versions 1.x, all event functions reside in *application-folder*\src\scripts*ModuleName-ViewName*\view-factory.js. In this release, any view Init event function and Hide event function that you code must reside in *application-folder*\app\src\modules*module-folder**view-folder*\router-event.js, for which defaults are provided, and all other event functions that you code must reside in *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js, for which some defaults are provided, depending on the view. For more information, see Adding and editing views on page 17.

- **Remove**: Removes the app from the Designer dashboard, but retains the app folder and files in its existing location, possibly for import at a later time.

- **Delete**: Permanently deletes the app from the Designer, including removing its folder and files from your system.

The menu bar appears above the breadcrumbs bar on every Designer page, and provides some of these same options and others that you might find useful for working with the Designer, depending on your page context. The breadcrumbs bar (starting with the **Progress** logo) tracks your progress through the Designer and allows you to return to prior pages as you work on apps. You can access this dashboard at any time by clicking the **Progress** logo in the breadcrumbs bar.

2

Creating an app

To create an app using the Kendo UI Designer, you need to create several sub-components:

- *Data providers*: Data services that provide access to data and associated logic. Each data provider is represented by one service URI.

- *Data sources*: Single tables found within data providers.

- *Modules*: The building blocks of your app, representing a single business function (e.g., order entry or inventory), which are then packaged into a single app. One built-in module is provided for you at app creation (called the Application module), which you can edit but cannot remove: it represents core app functions, including login and landing page functionality. You then create additional modules for your business functions.

- *Views*: The screens for the functions that make up each module. There are two basic types of views: built-in (login and landing-page views, which implement the Application module functions) and user-created (which implement your business functions). User-created views include:

  - *Predefined views*: Composed of data-driven, predefined view templates that the Designer populates and lays out based on your view data sources and property settings. These data-driven views have default behavior, depending on the template, that you can customize with additional coding.

  - *User-defined views*: Composed of a single Blank view template, which allows you to create your own layout based on UI components that you add separately. You can also specify multiple view data source instances from the data sources you have added to your app. You then associate one of these view data source instances with each UI component and specify one or two models that allow the UI component to access its data source and fields. Finally, you must add code to define the combined behavior of the UI components in the view.

For more information about these terms, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications.*

For details, see the following topics:

- Creating an app: first step

- App design page

- Adding data providers

- Adding data sources

- Adding and editing modules

- Adding and editing views

Creating an app: first step

When you click the **Create App** button in the upper-left corner of the dashboard, the Create App dialog box appears. Enter the following information in this dialog:

1. **App name**: The name of your application. The name cannot contain spaces. Only letters, numbers, dashes, and underscores are permitted.

2. **Theme**: Select a different color scheme if needed. Click the color-coded button to see a list of named themes to select, providing samples of each. **Metro** is the default. Click **Apply** to save your new setting.

3. **Location**: The location for the files associated with your app. This can match your workspace and project folder of the associated Web UI project in Progress® Developer's Studio for OpenEdge® (PDSOE) if you have OpenEdge installed.

4. **Description**: An optional description of the app for your own reference (not visible to the app user). This text appears both on the app card or list item on the dashboard and below the app title on the app design page.

When finished, click **Create App**, which opens the app design page.

See also
App design page on page 12

App design page

Once you've created a new app (or clicked on an existing app's card or list item on the dashboard), you enter the *app design page.* From this page, you can click:

- The pencil icon next to the app name: This opens an Edit App dialog, which is similar to the Create App dialog (see Creating an app: first step on page 12) where you can change some of the initial app settings plus its logo. The app name and location cannot be edited:

  - **Theme**: The selected color scheme. Click the color-coded button to see a list of available named themes to change it. Click **Apply** to save your changes.

  - **App Logo**: The logo that appears in the header of the app main (landing) page. The name of the logo you enter must be the name of an image file located in `application-folder\app\src\images`,

where *application-folder* is the path to the folder that contains and is named for your web app. A default image file is provided.

- **Description**: An optional description of the app for your own reference (not visible to the app user). This text appears both on the app card or list item on the dashboard and below the app title on the app design page.

- **Generate**: Generates the HTML and JavaScript source code for the app, ready to run in preview mode (when you click **Preview**).

- **Preview**: Shows a preview of what the finished app looks like and how it behaves in the default web browser using a webpack-dev-server. Preview runs the latest generation of the web app. If the app has not yet been generated, or requires generation after an update in the Designer, this option automatically generates the app before previewing it.

- **Publish**: Generates the HTML and JavaScript source code, ready for deployment using one of the following options that you choose:

  - **Debug**: Test deployment on a web server.

  - **Release**: Production deployment on a web server.

- **Modules**: Options for adding or editing modules and their views.

- **Data Providers**: Options for adding or editing data providers and their data sources.

See also

Adding data providers

To add a data provider:

1. Click **Add Data Provider** on the app design page. This displays the **Add Data Provider** dialog.

2. Select the **Provider Type** from the following options:

   - **Progress Data Provider:** A Progress OpenEdge or Rollbase Data Object Service.

   - **OData Provider:** An OData Service (read-only).

   - **Generic REST Provider:** A REST Service (read-only).

3. Enter a **Name** for the data provider. The name cannot contain spaces. Only letters, numbers, dashes, and underscores are permitted.

4. Enter the **Service URI**. For a Progress Data Provider, this is the URI of the web application that hosts the Data Object Service. For example, your service URI might look something like `http://Your-IP-Address:8980/MyDataWebApp`. For an OData Provider, this is an OData service root URL. For example, `http://Your-IP-Address:8980/OData/Inventory.svc`. For a Generic REST Provider, this is a REST service URL. For example, `http://Your-IP-Address:8980/api`

5. Click **Test Connection** to verify that the specified service is live.

6. For a Progress Data Provider only, enter the **Catalog URI**. This is the location of the Data Service Catalog file generated when the Data Object Service is built on the server. For an OpenEdge Data Object Service, see the help system in Progress Developer's Studio for OpenEdge (PDSOE) for more information.

> **Note:** In this release, the specified catalog must be unprotected at design time. If the **Catalog URI** specifies a protected URI, you must use the **Enable Local Catalog** option to specify an unprotected catalog file for design-time access, as described in the next step.

7. (Optional) For a Progress Data Provider only, select **Enable Local Catalog** if you want to use a local catalog file at design time. This allows you to browse for a catalog file to use on your system, which is then saved as part of your meta data.

> **Note:** Once entered and saved, the **Provider Type**, **Name**, **Service URI**, and **Catalog URI** of a data provider cannot be further edited in the Designer. Note also that if you use a local catalog at design time for a Progress Data Provider, when you publish a deployment build, the deployed web app uses the specified **Catalog URI** to access the catalog.

8. (Optional) For a Progress Data Provider or an OData Provider only, you have the option to manually create data sources after you have finished creating your data provider. However, you can also select **Auto-create Data Sources** to allow the Designer to automatically generate your data sources based on the resources defined in the catalog. A *data source* corresponds to a single table found in a data provider. For a Progress Data Provider, a data source is created using the schema definition in the catalog. When **Auto-create Data Sources** is selected, data sources are automatically generated for each of the following cases, given that the resource is enabled for at least one CRUD operation:

   - A single table: A data source is created using the resource name of the table from the catalog file.

   - For an OpenEdge Data Object Service only, a ProDataSet with a single, top-level table: A data source is created using the resource name of the top-level table from the catalog file

   - For an OpenEdge Data Object Service only, a ProDataSet with more than one top-level table: A data source for each top-level table is created with the following naming convention:
     `ProDataSet_resource_name + "." + table_name`

9. (Optional) For an OpenEdge Data Object Service only, select **Create Data Sources for child tables** to allow the Designer to automatically generate data sources for child tables in a ProDataSet with related tables. These child table resources are created with the same naming convention as for top-level tables.

> **Note:** You must also select **Auto-create Data Sources** to enable this option.

10. For a Progress Data Provider only, select the **Authentication Model**. This is the security model for the service URI and must match the authentication model of the web application that provides the Data Object Service.

Click **Add Data Provider** to the data provider with the current settings. With one or more data providers created, you can now create data sources using these providers.

See also

Adding data sources

To add a data source from an existing data provider:

---

1. Click **Edit** next to the given data provider in the app design page. You will be brought to the Edit Data Provider page

2. Click **Add Data Source**.

Note: If you chose the optional **Auto-create Data Sources** when creating your data provider, some of the work described below will be done for you in this dialog, but you can still manually edit the data source and individual properties and labels as needed. You can also create custom data sources in addition to the ones automatically created.

---

3. Enter a **Name** for the data source.

4. Search for or navigate to the table your app will draw data from. Double-click on the table name, and all of the available fields will appear in the **Included fields** box.

5. You can use the **Include all** and **Exclude all** buttons to move the entire list of fields back and forth between **Included Fields** and **Excluded Fields** lists. You can also drag and drop individual fields back and forth between the lists by grabbing the vertical drag bars to the left of each field item. The order that the fields appear in the **Included Fields** list is the default order that will be used for grid columns and/or form fields, so drag and drop the included fields into the order you prefer. You can later change the default field order for both the grid columns and form fields of any view that you create.

6. When you select a particular field, you can specify the **Label** and **Editor Type** properties. The label is used as the column head and/or the field label in the view. (See the sections on modules and views for more information.) You can select an editor type based on the underlying semantic type for the given field. For example, a date field might offer the option of a text-input editor or a date-input editor (date picker). For more information on semantic types, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications.*

Note: Editor types determine how a field is rendered on an editable form for either the Data-Grid-Form view or the Data-Grid-Separate-Form view. You can set optional properties and event functions on the editor type for a given field when you edit the form fields as described for instances of these views. For more information, see Editing form fields on page 26.

Note: Enter `plain-text` as the **Editor Type** to display a field as read-only on an otherwise editable form. This is useful for auto-generated fields, such as a primary key field whose value is maintained on the server.

---

7. If the field you select is specified with the `Lookup` semantic type, the field is a foreign key defined on the current data source table. Its name is defined by the data provider and appended with "`FK`*9*" by the Designer to ensure uniqueness, where *9* is an integer increment starting at `1` (for example, `SalesrepIdFK1`). A foreign key can be used to look up a unique record in a parent (top-level) table of a compatible data source that is defined in the same data provider as the current data source. The supported editor types for a foreign key include `combo-box` (the default), `drop-down-list`, and `plain-text`.

    For a foreign key field, the following additional properties appear to control the behavior of its selected **Editor Type** at run time:

    • **Data Source**: A selection from a list of data sources that contain a table with the same name as **Parent Table** and fields with the same names and types as **Parent Fields**. These data sources all share the same data provider as the current data source. The table in the selected data source is thus the parent table for look up using the foreign key.

    • **Display Field**: The field to display in the selected **Editor Type** from the record that the foreign key matches in the parent table. If the editor type is `combo-box` or `drop-down-list`, the user can change the foreign key value by selecting a new record from the list of values for this field, which represent the available records in the parent table.

- **Fields**: Read-only list of fields in the current data source table that define this foreign key.

  Note: This release supports foreign keys defined with only a single field.

- **Parent Table**: The read-only name of the parent table, qualified by the name of the remote Data Object resource that contains the table. This is also the name of one of the corresponding data sources created for the current data provider. This qualified parent table name can also be used to name the foreign key, depending on how it is defined by the current data provider.

- **Parent Fields**: Read-only list of fields in the parent data source table that defines its candidate key. These candidate key fields correspond in type and number to the fields that define the foreign key. At run time, they must match the corresponding foreign key fields in order to uniquely identify a record in the parent table.

  Note: This release supports candidate keys defined with only a single field.

8. (Optional) Select **Client-side processing** to indicate your app will retrieve all data before the view is visualized and should perform data presentation operations (sorting, filtering, and paging operations) on the client side. Otherwise, only a page of data is retrieved as the page is displayed, and these operations are performed on the server side.

   Note: If client-side processing is not checked, the Business Entity needs to include code for processing the request. This is done by enhancing the business logic to include code for filtering, sorting, and paging the data. For OpenEdge Data Object Services, Business Entities support a JSON Filter Pattern (JFP) that the Read operation method interprets by executing this code and a count method that provides the total number of records read. (You are prompted to enter the name of this count method if **Client-side processing** is unchecked.) See "Updates to allow access by Kendo UI DataSources and Rollbase external objects" in *OpenEdge Development: Web Services* for an example Business Entity implementation that handles the JFP and provides a count method.

The data source(s) you have created are now available for use by views in a module.

Note: If you create one or more views using a data source and then come back to change some of the details of that data source, such as labels and editor types, those changes will **not** be reflected in the views you have already created. These changes are in effect for any new views you create, but cannot be propagated back to views you have already created.

See also

Adding and editing modules

Once you have an accessible data source, you can create one or more modules to interact with that information:

1. From the app design page, click **Add Module**.

2. Add a **Name** for the module. The name cannot contain spaces. Only letters, numbers, dashes, and underscores are permitted.

Note: Once entered and saved, the name of the module cannot be edited.

---

3. Expand the **Icon/Color** menu and select an icon, icon color, and background color.

4. (Optional) Add a **Label** for the module. The label can contain spaces and is used to identify the module on the landing page at run time. If you do not specify a label, the module is identified using its name.

5. (Optional) Add a brief **Description**. This will appear under the module's name on the main editing page and as a tooltip for the module in the generated web app.

You can change some of the information (except for the module name) later on by clicking the pencil icon next to the module in the app design page.

Once you've created the module, it will be added to the list of modules on the app design page. The next step in creating your module is to add one or more views. Click **Edit** to enter the view design page. Once there, you can:

- Add and edit views.

- **Save** a view in progress.

- **Revert** a view to its saved state.

- **Generate** your application HTML and JavaScript source code for preview.

- **Preview** your application in the default browser.

- **Publish** your application for testing or production deployment.

See also

Adding and editing views

The Designer provides several types of views to use in your application:

- The built-in Application module contains two built-in views, a login view to authenticate any views that access data providers requiring authentication, and a landing-page view, which is the start page for your app.

- The modules you create can contain one or more pre-defined or user-defined views, both of which you can customize, depending on the type of view.

The login and landing-page views in the Application module are added for you. To add views to your own modules:

1. From the view design page, click **Add** in the upper left corner.

2. Add a **Name** for the view. The name cannot contain spaces. Only letters, numbers, dashes, and underscores are permitted.

Note: Once entered and saved, the name of the view cannot be edited.

---

3. (Optional) Add a **Label** for the view. The label can contain spaces and is used to identify the view on the views page at run time. If you do not specify a label, the view is identified using its name.

---

4. Choose the **View Type**. What follows is a basic summary. Each type of view has different properties and behaviors that can be customized, as described in later topics:

    *   **Data-Grid**: A read-only or user editable, table-like display of the information in the data source. The data source itself determines the organization and display type of the data.

    *   **Data-Grid-Form**: A split screen featuring a Data-Grid as described above and a form displayed on the same screen showing the information in a selected row. This view can be read-only or user editable. If you enable editing, the screen switches to display the form suitable for editing.

    *   **Data-Grid-Separate-Form**: Similar to the Data-Grid-Form, but in this case, only the Data-Grid is shown unless users select a row. The information from that row is then shown as a form displayed on a separate screen. If you enable editing, the screen switches to display the form suitable for editing.

    *   **Stacked-Data-Grids**: Displays information from two data source tables in a parent-child relationship. A read-only grid at the top of the view displays rows from the parent table, and a read-only or user editable grid below the parent grid displays rows from the child table that correspond to the single row currently selected in the parent. The child grid supports properties for editing the view very similar to the properties of the Data-Grid view.

    *   **Hierarchical-Data-Grid**: Displays information from two data source tables in a parent-child relationship. This is a single hierarchical grid, where rows from the parent table are displayed in a read-only mode, and rows from the child table are displayed in a read-only or user editable mode below the currently selected parent row. Although the visualization is different, the properties for editing this view are almost identical to the properties for the Stacked-Data-Grids view.

    *   **Blank**: Design your own custom view by dragging-and-dropping rows and columns to create a layout and then adding functional components to the layout for data management, field editing, data charting, scheduling, and navigation.

    Note: For all views, you are provided with a drop-down menu at the top of the screen that provides an approximate preview (without building the app) of what any given view will look like on a desktop, laptop, tablet in landscape, or tablet in portrait. This can be particularly helpful in finding out how rows will break when viewed on different devices.

5. (Optional) View event functions: After initially creating a view, you can then edit the view by clicking the gear icon next to the view's name and selecting **Edit**. This gives you the ability to change the view **Label** and specify the names of three custom functions to handle general view events:

    *   **Init Event Function**: Executes the first time the view is rendered.

    *   **Show Event Function**: Executes after the view is rendered.

    *   **Hide Event Function**: Executes when the view is replaced in the layout.

    The **Init Event Function** and **Hide Event Function** that you specify must be defined in the file, *application-folder*\app\src\modules*module-folder**view-folder*\router-event.js. This file contains empty default definitions of these functions, if specified for your views.

    The general view **Show Event Function** you specify must be defined in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. Additional view-specific and component event functions, like the **Row Select Event Function** for grids, are also supported, which you can specify in the component properties of the view. You must also define these functions in this same `controller.public.js` file, which contains empty default definitions of these functions, if specified for view and component properties.

Note that all the main files for adding HTML and JavaScript code extensions to a view are located in *application-folder*\app\src\modules*module-folder**view-folder*, where *application-folder* is the path to the folder that contains and is named for your web app, *module-folder* is the name of the folder defining the module, and *view-folder* is the name of the folder defining the view. For more information on event functions and other code extensions, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*.

The following sections describe how to edit each of the views described above:

- Login view on page 19

- Landing-page view on page 20

- Data-Grid view on page 20

- Data-Grid-Form view on page 24

- Data-Grid-Separate-Form view on page 37

- Stacked-Data-Grids view on page 39

- Hierarchical-Data-Grid view on page 44

- Blank view on page 48

Login view

To edit the built-in login view:

1. From the app design page, click **Edit** next to the **Application** module, at which point the login view will automatically open. If you are already in the view design page for the module, select **login** from the list of views.

2. You can edit the following details of the login view:

   - **Header Title**: The text that appears between the top and middle custom HTML sections.

   - **Logo**: An image that appears next to the Header Title. Enter the name of the file here, and place the file itself in *application-folder*\app\src\images.

   - **Username Label**: The text that appears above the box for app users to enter their usernames.

   - **Password Label**: The text that appears above the box for app users to enter their passwords.

   - **Events**:

     - **Login Event Function**: The name of a JavaScript function that runs when the user clicks the **Login** button.

     The code for these event functions must be included in the file, *application-folder*\app\src\modules\Application\login\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

This view also includes the following custom HTML sections where you can include your own HTML for the view. Each section is identified in the view design page with a placeholder containing descriptive text:

- **Custom top html section:** Located in the area above the Header Title, the text in this placeholder introduces the view path name of the topSection.html file where you can add your custom HTML.

- **Custom middle html section:**: Located in the area below the Header Title but above the Username Label, the text in this placeholder introduces the view path name of the `middleSection.html` file where you can add your custom HTML.

- **Custom bottom html section:**: Located in the area below the Password Label, the text in this placeholder introduces the view path name of the `bottomSection.html` file where you can add your custom HTML.

The Designer automatically generates these three files for you in *application-folder*`\app\src\modules\Application\login`. Add your custom HTML code to these files as required.

Landing-page view

To edit the built-in landing-page view:

- From the app design page, click **Edit** next to the **Application** module, at which point the login view will automatically open. Select **landing-page** from the list of views.

Other than the properties you can change by clicking the gear icon next to **landing-page** and selecting **Edit**, this view has no view properties you can set. It only displays a layout for the view showing the modules that appear on the page when a user starts up the app (with the exception of the Application module itself, which does not appear).

This view also includes the following custom HTML sections where you can include your own HTML for the view. Each section is identified in the view design page with a placeholder containing descriptive text:

- **Custom top html section:** Located in the area above the list of modules, the text in this placeholder introduces the view path name of the `topSection.html` file where you can add your custom HTML.

- **Custom bottom html section:**: Located in the area below the list of modules, the text in this placeholder introduces the view path name of the `bottomSection.html` file where you can add your custom HTML.

The Designer automatically generates these three files for you in *application-folder*`\app\src\modules\Application\login`. Add your custom HTML code to these files as required.

Data-Grid view

Selecting the Data-Grid view arranges rows of included fields from your data source by column, and in the same order you arranged them in the **Included Fields** list of the Grid Columns dialog (described below).

The Data-Grid view has a number of customizable properties, listed on the right of the view design page:

- **Grid Title**: The title that appears when this view is displayed on the screen for the app.

- **Data Provider**: The selected data provider for this view.

- **Data Source**: The selected data source from the chosen data provider.

- **Edit Mode**:

  - **ReadOnly**: This mode supports read-only access to data in the grid.

  - **Incell**: This mode supports edits to multiple rows at a time in the grid. Data editing is supported using the fields of the grid itself. You initiate editing by clicking a column within a row. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides buttons to create a new row, and to save all changes or cancel all changes to the rows in the grid.

- **Inline**: This mode supports edits to a single row at a time in the grid. Data editing is supported using the fields of the row itself. You initiate editing by clicking an edit button on the row you want to update. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides a single button to create a new row in the grid.

- **Popup**: This mode supports edits to a single row at a time in the grid. Data editing is supported using a pop-up form. You initiate editing by clicking an edit button on the row you want to update. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides a single button to create a new row in the grid.

- **Edit Options**:

  - **Allow Insert**: An option that allows new rows to be created in the grid.

  - **Allow Edit**: An option that allows existing rows to be edited in the grid.

  - **Allow Delete**: An option that allows existing rows to be deleted in the grid.

  - **Toolbar Button Labels**:

    - **Cancel**: A custom label for the button that cancels all pending edits in the grid. The default is "Cancel changes".

    - **Create**: A custom label for the button that creates a new row in the grid. The default is "New".

    - **Save**: A custom label for the button that saves all pending edits in the grid. The default is "Save changes".

  - **Row Button Labels**:

    - **Cancel Edit**: A custom label for the button that cancels all pending edits in the row. The default is "Cancel".

    - **Delete**: A custom label for the button that deletes the row. The default is "Delete".

    - **Edit**: A custom label for the button that initiates editing in the row. The default is "Edit".

    - **Update**: A custom label for the button that confirms any pending delete or edits to the row. The default is "Update".

- **Grid Columns**: Click **Edit** to change the included columns and various properties of the columns. See Editing grid columns on page 23 for more information.

- **Page Size**: The number of rows that will be displayed per page in the grid.

- **Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI rowTemplate. If specified, the row template is used to format all the rows in the grid unless you also define an altRowTemplate. For more information, see the entry for rowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-rowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*.

- **Alt Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI altRowTemplate. If specified, the altRowTemplate is used to format every other row in the grid. For more information, see the entry for altRowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-altRowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*

- **Alt Row Template Id**: The ID of an HTML altRowTemplate. If specified, the altRowTemplate must be defined in either the
  application-folder\app\src\modules*module-folder**view-folder*\topSection.html
  file, the \middleSection.html file, or the \bottomSection.html file. If both an Alt Row Template Function and an Alt Row Template Id are specified, the Row Template Id takes precedence.

- **Row Template Id**: The ID of an HTML rowTemplate. If specified, the rowTemplate must be defined in either the *application-folder*\app\src\modules*module-folder**view-folder*\topSection.html file, the \middleSection.html file, or the \bottomSection.html file. If both a Row Template Function and a Row Template Id are specified, the Row Template Id takes precedence.

- **Enable Column Filtering**: Provides the app user with the ability to filter the displayed information based on content, including filters for numeric values (e.g., is equal to, is not equal to) and text fields (e.g., starts with, does not contain, etc.).

- **Enable Grouping**: Provides the app user with the ability to group the rows according to the value of a particular column. For example, in a grid with columns representing customer names and order numbers, grouping by customer name will arrange the rows so that all of the order numbers from each given customer are grouped together. Grouping should only be used with client-side processing.

- **Enable Column Resize**: Provides the app user with the ability to resize the columns by dragging the column divider(s).

- **Enable Column Reordering**: Provides the app user with the ability to reorder the columns by dragging and dropping the column headers.

- **Enable Sorting**: Provides the app user with the ability to reorder rows by ascending or descending value in a given column.

- **Events**:

  - **Row Select Event Function**: The name of a JavaScript function that runs when a grid row is selected by the app user.

  - **Row Create Event Function**: The name of a JavaScript function that runs before a row is created for a new data source record.

  - **Row Update Event Function**: The name of a JavaScript function that runs before an existing data source record is updated in the row.

  - **Row Delete Event Function**: The name of a JavaScript function that runs before an existing data source record is deleted in the row.

  The code for these event functions must be included in the file,
  application-folder\app\src\modules*module-folder**view-folder*\controller.public.js.
  See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

This view also includes the following custom HTML sections where you can include your own HTML for the view. Each section is identified in the view design page with a placeholder containing descriptive text:

- **Custom top html section:** Located in the area above the Grid Title, the text in this placeholder introduces the view path name of the topSection.html file where you can add your custom HTML.

- **Custom middle html section::** Located in the area below the Grid Title but above the grid itself, the text in this placeholder introduces the view path name of the middleSection.html file where you can add your custom HTML.

- **Custom bottom html section::** Located in the area below the grid, the text in this placeholder introduces the view path name of the bottomSection.html file where you can add your custom HTML.

The Designer automatically generates these three files for you in
application-folder\app\src\modules*module-folder**view-folder*. Add your custom HTML
code to these files as required.

See also

Editing grid columns

From the **Grid Columns** dialog, you can change several properties of the columns:

- You can use the **Include All** and **Exclude All** buttons to move the entire list of fields from the grid data source back and forth between **Included Columns** and **Excluded Columns** lists. You can also drag and drop individual fields back and forth between the column lists by grabbing the vertical drag bars to the left of each field item. The order that the fields appear in the **Included Columns** list is the default order that will be used for the grid columns, so drag and drop the included fields into the order you prefer.

- When you select a particular field in the **Included Columns** list, you can view or set the following properties:

  - **Enable HTML Encoding**: When checked, any HTML coding included in the field is applied. For example, with HTML coding enabled, a field containing "Jane" appears as "**Jane**" instead of normal text with the tags visible.

  - **Field Name**: Read-only name of the selected field in the list.

  - **Format**: This can be used to customize how the field is presented. Use the argument {0} to represent the content of the column. You can also add additional text to be displayed. For example, if a column holds a numeric value measured in kilograms, `{0} kg` appends " kg" to every field in the column. You can also add specifiers to control the format of the field. If a field is a date value, you can specify that a column display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:

    `{0:D}`

    See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.

  - **Template**: You can use this to apply any valid Kendo template to an entire column. For example, if you have a field named LastName, then entering:

    `<strong>#: LastName #</strong>`

    applies boldface to every data item in that column. See http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-columns.template for more information.

  - **Label**: A column label that you can change from the default, which is the **Label** set for the selected field in the view data source.

  - **Width**: Initial width, measured in pixels, of the column. Leaving this value empty allows the column width to be responsive as needed.

The following options are also available for the Grid component in the Blank view:

- **Enable Sorting**: Provides the app user with the ability to reorder rows by ascending or descending value in a given column.

- **Enable Filtering**: Provides the app user with the ability to filter the displayed information based on content, including filters for numeric values (e.g., is equal to, is not equal to) and text fields (e.g., starts with, does not contain, etc.).

Data-Grid-Form view

Selecting the Data-Grid-Form view creates a horizontally split screen. On the left, a grid arranges rows of included fields from your data source by column, and in the same order as they appear in the **Included Columns** list of the Grid Columns dialog (described below). On the right, any single row that you select in the grid is presented in a form, with its fields displayed in the same order as they appear in the **Included Fields** list of the Form Fields dialog (described below). This view can be read-only or user editable, where the editing of any single row is available using the form.

The Data-Grid-Form view has a number of customizable properties, listed on the right side of the view design page:

- **Title**: The title that appears when this view is displayed on the screen for the app.

- **New Title**: The title that appears when the app user adds a new record. (This is only relevant for an editable view.)

- **Edit Title**: The title that appears when the app user edits an existing record. (This is only relevant for an editable view.)

- **Confirm Delete**: When checked, the user is prompted with a confirmation dialog when deleting an existing record. (This is only relevant for an editable view.)

- **Data Provider**: The selected data provider for this view.

- **Data Source**: The selected data source from the chosen data provider.

- **Edit Mode**: Choose **Read-Only** or **Read-Only to Edit**. Read-Only to Edit means that app users will be allowed to edit data in the form on the same screen. (The **Title** will change to **Edit Title** or **New Title**, depending on the type of edit the user is making.) The user's options for editing are determined by the default or overridden editor type, e.g., users will have access a basic editor with formatting capabilities for a rich text field.

- **Grid Columns**: Click **Edit** to change the included columns and various properties of the columns. See Editing grid columns on page 23 for more information.

- **Form Fields**: Click **Edit** to change the included fields and various properties of the form fields. See Editing form fields on page 26 for more information.

- **Page Size**: The number of rows that will be displayed per page in the grid.

- **Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI rowTemplate. If specified, the row template is used to format all the rows in the grid unless you also define an altRowTemplate. For more information, see the entry for rowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-rowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*.

- **Row Template Id**: The ID of an HTML rowTemplate. If specified, the rowTemplate must be defined in either the `application-folder\app\src\modules\module-folder\view-folder\topSection.html` file, the `\middleSection.html` file, or the `\bottomSection.html` file. If both a Row Template Function and a Row Template Id are specified, the Row Template Id takes precedence.

- **Alt Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI altRowTemplate. If specified, the altRowTemplate is used to format every other row in the grid. For more information, see the entry for altRowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-altRowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*

- **Alt Row Template Id**: The ID of an HTML altRowTemplate. If specified, the altRowTemplate must be defined in either the *application-folder*\app\src\modules*module-folder**view-folder*\topSection.html file, the \middleSection.html file, or the \bottomSection.html file. If both an Alt Row Template Function and an Alt Row Template Id are specified, the Row Template Id takes precedence.

- **Enable Column Filtering**: Provides the app user with the ability to filter the displayed information based on content, including filters for numeric values (e.g., is equal to, is not equal to) and text fields (e.g., starts with, does not contain, etc.).

Note: Filtering, sorting, and paging operations execute based on the **Client-side processing** setting for the data source. Operations are performed on the client if **Client-side processing** is selected; otherwise, the operations are performed on the server side. See the info on enabling server-side processing in Adding data sources on page 14.

- **Enable Grouping**: Provides the app user with the ability to group the rows according to the value of a particular column. For example, in a grid with columns representing customer names and order numbers, grouping by customer name will arrange the rows so that all of the order numbers from each given customer are grouped together. Grouping should only be used with client-side processing.

- **Enable Column Resize**: Provides the app user with the ability to resize the columns by dragging the column divider(s).

- **Enable Column Reordering**: Provides the app user with the ability to reorder the columns by dragging and dropping the column headers.

- **Enable Sorting**: Provides the app user with the ability to reorder rows by ascending or descending value in a given column.

- **Events**:

  - **Row Select Event Function**: The name of a JavaScript function that runs when a grid row is selected by the app user.

  - **Data Bound Event Function**: The name of a JavaScript function that runs when the grid is bound to its data source.

  The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

This view also includes the following custom HTML sections where you can include your own HTML for the view. Each section is identified in the view design page with a placeholder containing descriptive text:

- **Custom top html section:** Located in the area above the Title, the text in this placeholder introduces the view path name of the topSection.html file where you can add your custom HTML.

- **Custom middle html section::** Located in the area below the Title but above the grid and form itself, the text in this placeholder introduces the view path name of the middleSection.html file where you can add your custom HTML.

- **Custom bottom html section::** Located in the area below the grid and form, the text in this placeholder introduces the view path name of the bottomSection.html file where you can add your custom HTML.

The Designer automatically generates these three files for you in *application-folder*\app\src\modules*module-folder**view-folder*. Add your custom HTML code to these files as required.

See also

Editing form fields

From the **Form Fields** dialog, you can change several properties of the fields displayed in the form:

- You can use the **Include all** and **Exclude all** buttons to move the entire list of fields back and forth between **Included Fields** and **Excluded Fields** lists. You can also drag and drop individual fields back and forth between the field lists by grabbing the vertical drag bars to the left of each field item. The order that the fields appear in the **Included Columns** list is the default order that will be used for the form, so drag and drop the included fields into the order you prefer.

- You can use the **+ Add Group** button to add one or more field groups to the **Included Fields** list with the default label, `New Group`. A field group with this label is initially created for every grid view with a form. You can change the label by clicking the group item, which displays the **Name** property under **Properties** where you can enter a new label value. Field groups do not appear in your form unless you populate them with field items and you have more than one field group created. You can populate any field group by clicking the down arrow on the group item to open it and dragging and dropping fields from a field list or any other open field group into the current field group. At run time, field groups are displayed as tab folders on the form.

- When you select an field item in the **Included Fields** list, you can view or set the following **Properties**:

  - **Format**: This can be used to customize how the field is presented. Use the argument {0} to represent the content of the field. You can also add additional text to be displayed. For example, if a field holds a numeric value measured in kilograms, `{0} kg` appends " kg" to every value of the field. You can also add specifiers to control the format of the field. If a field is a date value, you can specify that it display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:

    `{0:D}`

    See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.

  - **Label Text**: A field label that you can change from the default, which is the **Label** set for the selected field in the view data source.

  - **Field Name**: Read-only name of the selected field in the list.

- You can also set additional properties and events under **Editor Type** *Name*, where *Name* is the name for the editor type specified the selected field in the **Included Fields** list. (**Note:** This *Name* can differ from the name used to select the editor type for the field in the Edit Data Source dialog.) These are properties and events for the specified editor type that you can change to modify its default appearance and behavior in the form, as described below.

All of the editor types have an Id property:

- **Id**: A unique string with a default value that is automatically generated by the Designer for each component. You can edit the Id to a more meaningful value as you require; however, it cannot be empty and must be unique among all the editor type Id values in the view.

The remaining editor type properties and events are listed for each editor type in the following table, where **Editor Type in Form Fields** is the name of the editor type listed for the selected field in the Form Fields dialog, and **Editor Type in Data Source** is the name used to select the editor type for the same field in the Edit Data Source dialog.

Table 2: Editor Type properties and events

Editor Type in Form Fields	Editor Type in Data Source	Properties, events, and notes

Boolean Radio Button List	bool-radio-set	A radio button set with two possible boolean values (true or false).
		
		• **Default Value**: Sets the initial state of the buttons. Check the box to set the True button value, or leave unchecked to set the False button value (the default).
		
		• **True text**: The label that appears next to the True button ("True" by default).
		
		• **False text**: The label that appears next to the False button ("False" by default).
		
		• **Events**:
		
		  • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.
		
		  The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Calendar	calendar	A graphical calendar that supports navigation and selection.
		
		• **Min** and **Max Date** (optional): The minimum (earliest) and maximum (latest) dates that can be displayed.
		
		• **Default Date** (optional): The date initially displayed.
		
		• **Date Format**: Specifies the date format used by the calendar component's underlying value( ) function. See http://docs.telerik.com/kendo-ui/api/javascript/ui/calendar#configuration-value for more information.
		
		• **Events**:
		
		  • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.
		
		  The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Editor Type in Form Fields	Editor Type in Data Source	Properties, events, and notes

Check Box	check-box	A box that displays with a check mark or empty, depending on the boolean value of the component.
		
		• **Value**: Check this box to set the initial state of the component to true, or leave the box empty to set its initial state to false.
		
		• **Disabled**: Check this box to disable the check box, or leave it unchecked to make the check box active.
		
		• **Title Text**: This optional text appears when a user hovers over the editor type.
		
		• **Events**:
		
		   • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.
		
		   The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Combo Box	combo-box	This editor type binds to a data source to display a list of values that the user can both select and edit. • **Title Text**: This optional text appears when a user hovers over the editor type. **Note:** This editor type supports the Lookup semantic type that identifies foreign key fields defined in a data source. Most of the properties of this editor type are provided by the data source where the foreign keys are defined, some of which you must set to identify the lookup (reference) data source and its displayed field. For more information, see Adding data sources on page 14.
Currency Text Box	currency-input	A specialized version of a text-box for currency. • **Decimals**: The number of decimal places displayed. • **Default Value**: The amount initially displayed in the text box. The user can edit or delete this amount as needed. • **Down Arrow Text**: The text shown when the user hovers over the down arrow. • **Min** and **Max**: The minimum and maximum values allowed, respectively. • **Placeholder**: The amount that initially appears in the text box. Unlike Default Value, this text automatically disappears when the user clicks on the text box.

Editor Type in Form Fields	Editor Type in Data Source	Properties, events, and notes

		• **Step**: The interval the entered value changes for each click of the down or up arrow.
		
		• **Title Text**: This optional text appears when a user hovers over the editor type.
		
		• **Up Arrow Text**: The text shown when the user hovers over the up arrow.
		
		• **Show Spinners**: Select to display up and down arrows that change the field value when clicked by the increasing or decreasing **Step**-specified interval. Not selected by default.
		
		• **Events**:
		
		  • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.
		
		  The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Date Picker	date-input	A specialized drop down editor type that allows you to select a date value from a drop down calendar.
		
		• **Min** and **Max Date**: The minimum (earliest) and maximum (latest) dates that can be displayed.
		
		• **Default Date**: The date initially displayed.
		
		• **Date Format**: The format the date is displayed in once the user has made a selection.
		
		• **Title Text**: This optional text appears when a user hovers over the editor type.
		
		• **Events**:
		
		  • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.
		
		  The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Editor Type in Form Fields	Editor Type in Data Source	Properties, events, and notes

Date Time Picker	date-time-input	A specialized drop down editor type that allows you to select a date and time value from a drop down calendar.
		
		• **Min** and **Max Date**: The minimum (earliest) and maximum (latest) dates and times that can be displayed.
		
		• **Default Date**: The date and time initially displayed.
		
		• **Date Format**: The format for how the date and time are displayed.
		
		• **Time Interval**: The interval, in minutes, between selectable values in the time drop-down menu, e.g., a value of 30 gives users the choice of 1:00, 1:30, 2:00, etc.
		
		• **Title Text**: This optional text appears when a user hovers over the editor type.
		
		• **Events**:
		
		  • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.
		
		  The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Disabled Text Box	plain-text	Use the disabled text box to display read-only text. • **Default Value**: The text initially displayed in the text box. You can edit or delete this text as needed. • **Title Text**: This optional text appears when a user hovers over the editor type.
Drop Down List	drop-down-list	This editor type binds to a data source to display a list of values that the user can select. • **Title Text**: This optional text appears when a user hovers over the editor type. **Note:** This editor type supports the Lookup semantic type that identifies foreign key fields defined in a data source. Most of the properties of this editor type are provided by the data source where the foreign keys are defined, some of which you must set to identify the lookup (reference) data source and its displayed field. For more information, see Adding data sources on page 14.

Editor Type in Form Fields	Editor Type in Data Source	Properties, events, and notes

Editor	editor	A customizable HTML editor.
		
		• **Encoded**: Indicates whether the editor submits encoded HTML.
		
		• **Resizable**
		
		  • **Content**: If checked, the user can resize the editor.
		
		  • **Min** and **Max**: The smallest and largest size the user can choose for the size of the editor.
		
		  • **Toolbar**: When checked, the formatting tools are limited to one row at the top of the editor, with the tools that do not fit organized into a drop-down menu. When unchecked, the tools are organized into as many rows as necessary.
		
		• **Title Text**: This optional text appears when a user hovers over the editor type.
		
		• **Tools**: Select one or more formatting tools available to your users. Once selected, any tool can be removed by clicking the **X** next to its name.
		
		• **Events**:
		
		  • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.
		
		The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Email Text Box	email-input	A specialized version of a text-box for e-mail addresses. • **Debounce**: The delay, in milliseconds, between user input and the running of the Change Event Function. • **Default Value**: The e-mail address initially displayed in the text box. The user can edit or delete this text as needed. • **Placeholder**: The e-mail address initially displayed in the text box. Unlike Default Value, this text automatically disappears when the user clicks on the text box. • **Title Text**: This optional text appears when a user hovers over the editor type. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.

Editor Type in Form Fields	Editor Type in Data Source	Properties, events, and notes
		The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Integer Text Box	integer-input	A specialized version of a text-box for integer values only. • **Default Value**: The integer initially displayed in the text box. The user can edit or delete this text as needed. • **Down Arrow Text**: The text shown when the user hovers over the down arrow. • **Placeholder**: The integer initially displayed in the text box. Unlike Default Value, this text automatically disappears when the user clicks on the text-box. • **Step**: The interval the entered value changes for each click of the down or up arrow. • **Title Text**: This optional text appears when a user hovers over the editor type. • **Up Arrow Text**: The text shown when the user hovers over the up arrow. • **Show Spinners**: Select to display up and down arrows that change the field value when clicked by the increasing or decreasing **Step**-specified interval. Not selected by default. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Numeric Text Box	numeric-input	A specialized version of a text-box for numeric values only. • **Decimals**: The number of decimals displayed. • **Default Value**: The number initially displayed in the text box. The user can edit or delete this text as needed. • **Down Arrow Text**: The text shown when the user hovers over the down arrow.

Editor Type in Form Fields	Editor Type in Data Source	Properties, events, and notes
		• **Format**: The numeric format for the input. See http://docs.telerik.com/kendo-ui/api/javascript/ui/maskedtextbox#configuration-mask for the supported numeric types. • **Placeholder**: The number initially displayed in the text box. Unlike Default Value, this text automatically disappears when the user clicks on the text-box. • **Step**: The interval the entered value changes for each click of the down or up arrow. • **Title Text**: This optional text appears when a user hovers over the editor type. • **Up Arrow Text**: The text shown when the user hovers over the up arrow. • **Show Spinners**: Select to display up and down arrows that change the field value when clicked by the increasing or decreasing **Step**-specified interval. Not selected by default. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Password Text Box	password-input	A specialized version of a text-box for passwords. • **Placeholder**: The text initially displayed in the text box. This text automatically disappears when the user clicks on the text box. • **Title Text**: This optional text appears when a user hovers over the editor type. • **Debounce**: The delay, in milliseconds, between user input and the running of the Change Event Function. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Editor Type in Form Fields	Editor Type in Data Source	Properties, events, and notes
Percent Text Box	percent-input	A specialized version of a text-box for percent values only. Use this when the percent values in the database must be multiplied by 100 before appending the % symbol for proper display (e.g., the database value of 0.256 will be displayed as 25.6%). • **Decimals**: The number of decimal places displayed. • **Default Value**: The value that initially appears in the text box. The user can edit or delete this text as needed. • **Down Arrow Text**: The text shown when the user hovers over the down arrow. • **Placeholder**: The value that initially appears in the text box. Unlike initial text, this text automatically disappears when the user clicks on the text-box. • **Step**: The interval the entered value changes for each click of the down or up arrow. • **Title Text**: This optional text appears when a user hovers over the editor type. • **Up Arrow Text**: The text shown when the user hovers over the up arrow. • **Show Spinners**: Select to display up and down arrows that change the field value when clicked by the increasing or decreasing **Step**-specified interval. Not selected by default. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Percent Value Text Box	percent-value-input	A specialized version of a text-box for percent values only. Use this when the percent values in the database only need the addition of the % symbol for proper display (e.g., the database value of 25 will be displayed as 25%). • **Decimals**: The number of decimal places displayed. • **Default Value**: The value that initially appears in the text box. The user can edit or delete this text as needed. • **Down Arrow Text**: The text shown when the user hovers over the down arrow.

Editor Type in Form Fields	Editor Type in Data Source	Properties, events, and notes

		• **Placeholder**: The value that initially appears in the text box. Unlike Default Value, this text automatically disappears when the user clicks on the text-box.
		
		• **Step**: The interval the entered value changes for each click of the down or up arrow.
		
		• **Title Text**: This optional text appears when a user hovers over the editor type.
		
		• **Up Arrow Text**: The text shown when the user hovers over the up arrow.
		
		• **Show Spinners**: Select to display up and down arrows that change the field value when clicked by the increasing or decreasing **Step**-specified interval. Not selected by default.
		
		• **Events**:
		
		  • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.
		
		  The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Phone Text Box	phone-number-input	A specialized version of a text-box for phone numbers.
		
		• **Mask**: Specifies the input mask. For example, (000) 000-0000 allows the user to enter a standard U.S., 10-digit phone number.
		
		• **Default Value**: The value that initially appears in the text box. The user can edit or delete this text as needed.
		
		• **Title Text**: This optional text appears when a user hovers over the editor type.
		
		• **Events**:
		
		  • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.
		
		  The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Editor Type in Form Fields	Editor Type in Data Source	Properties, events, and notes

Slider	numeric-slider	A rich editor type for selecting numeric values.
		
		• **Orientation**: Chooses a horizontal or vertical orientation for the slider.
		
		• **Show Buttons**: Select to display decrease and increase arrow buttons on either end of the slider. Selected by default.
		
		• **Step**: The interval that determines which values on the slider are valid, e.g., a step of 0.5 on a range of 1 to 10 allows users to select 1, 1.5, 2, 2.5, etc.
		
		• **Default Value**: The number at which the slider is initially positioned.
		
		• **Title Text**: This optional text appears when a user hovers over the editor type.
		
		• **Events**:
		
		  • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.
		
		  • **Slide Event Function**: The name of a JavaScript function that runs when the user moves the drag handle.
		
		The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Text Box	text-input	A general purpose text box. • **Placeholder**: Text that initially appears in the text box. This text automatically disappears when the user clicks on the text box. • **Title Text**: This optional text appears when a user hovers over the editor type. • **Debounce**: The delay, in milliseconds, between user input and the running of the Change Event Function. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value.

Editor Type in Form Fields	Editor Type in Data Source	Properties, events, and notes
		The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder* \controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Url Text Box	url-input	A specialized version of a text-box for URLs. • **Debounce**: The delay, in milliseconds, between user input and the running of the Change Event Function. • **Placeholder**: The URL initially displayed in the text box. Unlike Default Text, this text automatically disappears when the user clicks on the text box. • **Title Text**: This optional text appears when a user hovers over the editor type. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the underlying field value. The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder* \controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Data-Grid-Separate-Form view

Selecting the Data-Grid-Separate-Form view works in much the same way as the Data-Gird-Form view, except that the user initially sees the grid only until they select a row. With a row selected in the grid, the view changes so the user sees the form only, displaying the fields included in that form from the selected row. Like the Data-Grid-Form view, this view can be read-only or user editable, where the editing of any single row in the grid is available using the form. Buttons on the form allow the user to return the view to displaying the grid only, depending on the edit mode of the view and the status of any edits.

The Data-Grid-Separate-Form view has a number of customizable properties, listed on the right side of the view design page:

• **Title**: The title that appears when this view is displayed on the screen for the app.

• **New Title**: The title that appears when the app user adds a new record. (This is only relevant for an editable view.)

• **Edit Title**: The title that appears when the app user edits an existing record. (This is only relevant for an editable view.)

• **Confirm Remove**: When checked, the user is prompted with a confirmation dialog when deleting an existing record. (This is only relevant for an editable view.)

- **Data Provider**: The selected data provider for this view.

- **Data Source**: The selected data source from the chosen data provider.

- **Edit Mode**: Choose **Read-Only** or **Read-Only to Edit**. Read-Only to Edit means that app users will be allowed to edit data in the form on the same screen. (The **Title** will change to **Edit Title** or **New Title**, depending on the type of edit the user is making.) The user's options for editing are determined by the underlying data type, e.g., users will have access a basic editor with formatting capabilities for a rich text field.

- **Grid Columns**: Click **Edit** to change various properties of the columns. See Editing grid columns on page 23 for more information.

- **Form Fields**: Click **Edit** to change various properties of the form fields. See Editing form fields on page 26 for more information.

- **Page Size**: The number of rows that will be displayed per page in the grid.

- **Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI rowTemplate. If specified, the row template is used to format all the rows in the grid unless you also define an altRowTemplate. For more information, see the entry for rowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-rowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*.

- **Row Template Id**: The ID of an HTML rowTemplate. If specified, the rowTemplate must be defined in either the `application-folder\app\src\modules\module-folder\view-folder\topSection.html` file, the `\middleSection.html` file, or the `\bottomSection.html` file. If both a Row Template Function and a Row Template Id are specified, the Row Template Id takes precedence.

- **Alt Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI altRowTemplate. If specified, the altRowTemplate is used to format every other row in the grid. For more information, see the entry for altRowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-altRowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*

- **Alt Row Template Id**: The ID of an HTML altRowTemplate. If specified, the altRowTemplate must be defined in either the `application-folder\app\src\modules\module-folder\view-folder\topSection.html` file, the `\middleSection.html` file, or the `\bottomSection.html` file. If both an Alt Row Template Function and an Alt Row Template Id are specified, the Row Template Id takes precedence.

- **Enable Column Filtering**: Provides the app user with the ability to filter the displayed information based on content, including filters for numeric values (e.g., is equal to, is not equal to) and text fields (e.g., starts with, does not contain, etc.).

- **Enable Grouping**: Provides the app user with the ability to group the rows according to the value of a particular column. For example, in a grid with columns representing customer names and order numbers, grouping by customer name will arrange the rows so that all of the order numbers from each given customer are grouped together. Grouping should only be used with client-side processing.

- **Enable Column Resize**: Provides the app user with the ability to resize the columns by dragging the column divider(s).

- **Enable Column Reordering**: Provides the app user with the ability to reorder the columns by dragging and dropping the column headers.

- **Enable Sorting**: Provides the app user with the ability to reorder rows by ascending or descending value in a given column.

- **Events**:

  - **Row Select Event Function**: The name of a JavaScript function that runs when a grid row is selected by the app user.

  - **Data Bound Event Function**: The name of a JavaScript function that runs when the grid is bound to its data source.

  The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

This view also includes the following custom HTML sections where you can include your own HTML for the view. Each section is identified in the view design page with a placeholder containing descriptive text:

- **Custom top html section:** Located in the area above the Title for both the grid and form, the text in this placeholder introduces the view path name of the `topSection.html` file where you can add your custom HTML.

- **Custom middle html section::** Located in the area below the Title but above the grid or the form itself, the text in this placeholder introduces the view path name of the `middleSection.html` file where you can add your custom HTML.

- **Custom bottom html section::** Located in the area below both the grid and form, the text in this placeholder introduces the view path name of the `bottomSection.html` file where you can add your custom HTML.

The Designer automatically generates these three files for you in *application-folder*\app\src\modules*module-folder**view-folder*. Add your custom HTML code to these files as required.

See also

Stacked-Data-Grids view

Selecting the Stacked-Data-Grids view creates a vertically split screen with two separate grids. On the top, a parent grid arranges rows of included fields from a parent data source by column, and in the same order as they appear in the **Included Columns** list of the Grid Columns dialog for the parent grid (described below). On the bottom, a child grid arranges related rows of included fields from a child data source by column, and in the same order as they appear in the **Included Columns** list of the Grid Columns dialog for the child grid (described below).

The parent grid provides read-only access to its data source. When the user selects a row in the parent grid, the child grid displays the related rows from its child data source and provides read-only or editable access to that data source as specified by its properties. Like any grid, both the parent and child grid allow you to page through all their rows using their own page selection controls. In this way, you can page through all related child rows by paging and clicking every parent row, then paging through all the rows displayed in the child grid.

Note: Both the selected parent and child data sources must reference tables from the same selected data provider, where both tables are bound by a parent-child relationship in the same JSDO. This JSDO must be created for an OpenEdge ProDataSet resource that includes both the two corresponding temp-tables and their data-relation as part of its definition.

The Stacked-Data-Grids view has a number of customizable properties, listed on the right side of the view design page:

- **Data Provider**: The selected data provider for this view.

- **Parent Data Source**: The selected parent-grid data source from the chosen data provider.

- **Child Data Source**: The selected child-grid data source from the chosen data provider.

- **Parent Grid Properties**:

    - **Grid Title**: The title that appears for the parent grid when this view is rendered on the screen for the app.

    - **Grid Columns**: Click **Edit** to change the included columns and various properties of the columns. See Editing grid columns on page 23 for more information.

    - **Page Size**: The number of rows that will be displayed per page in the grid.

    - **Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI rowTemplate. If specified, the row template is used to format all the rows in the grid unless you also define an altRowTemplate. For more information, see the entry for rowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-rowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*.

    - **Row Template Id**: The ID of an HTML rowTemplate. If specified, the rowTemplate must be defined in either the `application-folder`\app\src\modules\`module-folder`\`view-folder`\topSection.html file, the \middleSection.html file, or the \bottomSection.html file. If both a Row Template Function and a Row Template Id are specified, the Row Template Id takes precedence.

    - **Alt Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI altRowTemplate. If specified, the altRowTemplate is used to format every other row in the grid. For more information, see the entry for altRowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-altRowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*

    - **Alt Row Template Id**: The ID of an HTML altRowTemplate. If specified, the altRowTemplate must be defined in either the `application-folder`\app\src\modules\`module-folder`\`view-folder`\topSection.html file, the \middleSection.html file, or the \bottomSection.html file. If both an Alt Row Template Function and an Alt Row Template Id are specified, the Row Template Id takes precedence.

    - **Enable Column Filtering**: Provides the app user with the ability to filter the displayed information based on content, including filters for numeric values (e.g., is equal to, is not equal to) and text fields (e.g., starts with, does not contain, etc.).

    > **Note:** Filtering, sorting, and paging operations execute based on the **Client-side processing** setting for the data source. Operations are performed on the client if **Client-side processing** is selected; otherwise, the operations are performed on the server side. See the info on enabling server-side processing in Adding data sources on page 14.

    - **Enable Grouping**: Provides the app user with the ability to group the rows according to the value of a particular column. For example, in a grid with columns representing customer names and order numbers, grouping by customer name will arrange the rows so that all of the order numbers from each given customer are grouped together. Grouping should only be used with client-side processing.

- **Enable Column Resize**: Provides the app user with the ability to resize the columns by dragging the column divider(s).

- **Enable Column Reordering**: Provides the app user with the ability to reorder the columns by dragging and dropping the column headers.

- **Enable Sorting**: Provides the app user with the ability to reorder rows by ascending or descending value in a given column.

- **Events**:

  - **Row Select Event Function**: The name of a JavaScript function that runs when a parent row is selected by the app user.

  - **Data Bound Event Function**: The name of a JavaScript function that runs when the parent grid is bound to its data source.

  The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

- **Child Grid Properties**:

  - **Grid Title**: The title that appears for the child grid when this view is rendered on the screen for the app.

  - **Edit Mode**:

    - **ReadOnly**: This mode supports read-only access to data in the grid.

    - **Incell**: This mode supports edits to multiple rows at a time in the grid. Data editing is supported using the fields of the grid itself. You initiate editing by clicking a column within a row. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides buttons to create a new row, and to save all changes or cancel all changes to the rows in the grid.

    - **Inline**: This mode supports edits to a single row at a time in the grid. Data editing is supported using the fields of the row itself. You initiate editing by clicking an edit button on the row you want to update. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides a single button to create a new row in the grid.

    - **Popup**: This mode supports edits to a single row at a time in the grid. Data editing is supported using a pop-up form. You initiate editing by clicking an edit button on the row you want to update. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides a single button to create a new row in the grid.

  - **Edit Options**:

    - **Allow Insert**: An option that allows new rows to be created in the grid.

    - **Allow Edit**: An option that allows existing rows to be edited in the grid.

    - **Allow Delete**: An option that allows existing rows to be deleted in the grid.

    - **Toolbar Button Labels**:

      - **Cancel**: A custom label for the button that cancels all pending edits in the grid. The default is "Cancel changes".

      - **Create**: A custom label for the button that creates a new row in the grid. The default is "New".

      - **Save**: A custom label for the button that saves all pending edits in the grid. The default is "Save changes".

- **Row Button Labels**:

  - **Cancel Edit**: A custom label for the button that cancels all pending edits in the row. The default is "Cancel".

  - **Delete**: A custom label for the button that deletes the row. The default is "Delete".

  - **Edit**: A custom label for the button that initiates editing in the row. The default is "Edit".

  - **Update**: A custom label for the button that confirms any pending delete or edits to the row. The default is "Update".

- **Grid Columns**: Click **Edit** to change the included columns and various properties of the columns. See Editing grid columns on page 23 for more information.

- **Page Size**: The number of rows that will be displayed per page in the grid.

- **Row Template Function Name**: The name of a JavaScript function that returns the text for a Kendo UI rowTemplate. If specified, the row template is used to format all the rows in the grid unless you also define an altRowTemplate. For more information, see the entry for rowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-rowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*.

- **Row Template Id**: The ID of an HTML rowTemplate. If specified, the rowTemplate must be defined in either the
  `application-folder\app\src\modules\module-folder\view-folder\topSection.html`
  file, the `\topParentSection.html` file, the `\middleSection.html` file, the `\topChildSection.html` file, or the `\bottomSection.html` file. If both a Row Template Function and a Row Template Id are specified, the Row Template Id takes precedence.

- **Alt Row Template Function Name**: The name of a JavaScript function that returns the text for a Kendo UI altRowTemplate. If specified, the altRowTemplate is used to format every other row in the grid. For more information, see the entry for altRowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-altRowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*

- **Alt Row Template Id**: The ID of an HTML altRowTemplate. If specified, the altRowTemplate must be defined in either the
  `application-folder\app\src\modules\module-folder\view-folder\topSection.html`
  file, the `\middleSection.html` file, or the `\bottomSection.html` file. If both an Alt Row Template Function and an Alt Row Template Id are specified, the Row Template Id takes precedence.

- **Enable Column Filtering**: Provides the app user with the ability to filter the displayed information based on content, including filters for numeric values (e.g., is equal to, is not equal to) and text fields (e.g., starts with, does not contain, etc.).

- **Enable Grouping**: Provides the app user with the ability to group the rows according to the value of a particular column. For example, in a grid with columns representing customer names and order numbers, grouping by customer name will arrange the rows so that all of the order numbers from each given customer are grouped together. Grouping should only be used with client-side processing.

- **Enable Column Resize**: Provides the app user with the ability to resize the columns by dragging the column divider(s).

- **Enable Column Reordering**: Provides the app user with the ability to reorder the columns by dragging and dropping the column headers.

- **Enable Sorting**: Provides the app user with the ability to reorder rows by ascending or descending value in a given column.

- **Selection Type**: Determines if rows in a grid can be selected:

  - **None**: No row selection is allowed (default).

  - **Single**: One row can be selected at a time.

  - **Multiple**: Multiple rows can be selected at one time.

Note: Row selection has no function except to invoke the Row Select Event for each selected row.

- **Events**:

  - **Row Select Event Function**: The name of a JavaScript function that runs when a child row is selected by the app user.

  - **Data Bound Event Function**: The name of a JavaScript function that runs when the child grid is bound to its data source.

  - **Row Create Event Function**: The name of a JavaScript function that runs before a row is created for a new data source record.

  - **Row Update Event Function**: The name of a JavaScript function that runs before an existing data source record is updated in the row.

  - **Row Delete Event Function**: The name of a JavaScript function that runs before an existing data source record is deleted in the row.

    The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

This view also includes the following custom HTML sections where you can include your own HTML for the view. Each section is identified in the view design page with a placeholder containing descriptive text:

- **Custom top html section:** Located in the area above the parent Grid Title, the text in this placeholder introduces the view path name of the topSection.html file where you can add your custom HTML.

- **Custom top parent html section:**: Located in the area below the parent Grid Title but above the parent grid itself, the text in this placeholder introduces the view path name of the topParentSection.html file where you can add your custom HTML.

- **Custom middle html section:**: Located in the area below the parent grid and above the child Grid Title, the text in this placeholder introduces the view path name of the middleSection.html file where you can add your custom HTML.

- **Custom top child html section:**: Located in the area below the child Grid Title but above the child grid itself, the text in this placeholder introduces the view path name of the topChildSection.html file where you can add your custom HTML.

- **Custom bottom html section:**: Located in the area below the child grid, the text in this placeholder introduces the view path name of the bottomSection.html file where you can add your custom HTML.

The Designer automatically generates these three files for you in *application-folder*\app\src\modules*module-folder**view-folder*. Add your custom HTML code to these files as required.

See also

Hierarchical-Data-Grid view

Selecting the Hierarchical-Data-Grid view creates a single scrolling grid structure that contains a single parent grid containing rows from a parent data source. This parent grid then embeds a child grid below each parent row that has related records in the child data source. As you scroll the grid structure and click an expander on parent rows, any child grid containing related data source rows displays below its related row in the parent grid. You can also click the expander on an expanded child grid to collapse the grid into its parent.

Both the parent and its child grids arrange the included fields from their respective data sources by column, in the same order as they appear in the **Included Columns** list of the Grid Columns dialog for the parent its child grids (described below).

The parent grid provides read-only access to its data source. When the user selects the expander on a row in the parent grid, if this parent row has related child rows, a child grid displays the related rows from its child data source and provides read-only or editable access to that data source as specified by its properties. Like any grid, both the parent and its child grids allow you to page through all their rows using page selection controls on each grid. In this way, you can page through all related child rows by paging and expanding every parent row, then paging through all the rows of each displayed child grid.

Note: Both the selected parent and child data sources must reference tables from the same selected data provider, where both tables are bound by a parent-child relationship in the same JSDO. This JSDO must be created for an OpenEdge ProDataSet resource that includes both the two corresponding temp-tables and their data-relation as part of its definition.

The Hierarchical-Data-Grid view has a number of customizable properties, listed on the right side of the view design page:

- **Data Provider**: The selected data provider for this view.

- **Parent Data Source**: The selected parent-grid data source from the chosen data provider.

- **Child Data Source**: The selected child-grid data source from the chosen data provider.

- **Parent Grid Properties**:

  - **Grid Title**: The title that appears for the parent grid when this view is displayed on the screen for the app.

  - **Grid Columns**: Click **Edit** to change the included columns and various properties of the columns. See Editing grid columns on page 23 for more information.

  - **Page Size**: The number of rows that will be displayed per page in the grid.

  - **Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI rowTemplate. If specified, the row template is used to format all the rows in the grid unless you also define an altRowTemplate. For more information, see the entry for rowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-rowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*.

  - **Row Template Id**: The ID of an HTML rowTemplate. If specified, the rowTemplate must be defined in either the

---

application-folder\app\src\modules*module-folder**view-folder*\topSection.html file, the \middleSection.html file, or the \bottomSection.html file. If both a Row Template Function and a Row Template Id are specified, the Row Template Id takes precedence.

- **Alt Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI altRowTemplate. If specified, the altRowTemplate is used to format every other row in the grid. For more information, see the entry for altRowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-altRowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*

- **Alt Row Template Id**: The ID of an HTML altRowTemplate. If specified, the altRowTemplate must be defined in either the
  application-folder\app\src\modules*module-folder**view-folder*\topSection.html file, the \middleSection.html file, or the \bottomSection.html file. If both an Alt Row Template Function and an Alt Row Template Id are specified, the Row Template Id takes precedence.

- **Enable Column Filtering**: Provides the app user with the ability to filter the displayed information based on content, including filters for numeric values (e.g., is equal to, is not equal to) and text fields (e.g., starts with, does not contain, etc.).

Note: Filtering, sorting, and paging operations execute based on the **Client-side processing** setting for the data source. Operations are performed on the client if **Client-side processing** is selected; otherwise, the operations are performed on the server side. See the info on enabling server-side processing in Adding data sources on page 14.

- **Enable Grouping**: Provides the app user with the ability to group the rows according to the value of a particular column. For example, in a grid with columns representing customer names and order numbers, grouping by customer name will arrange the rows so that all of the order numbers from each given customer are grouped together. Grouping should only be used with client-side processing.

- **Enable Column Resize**: Provides the app user with the ability to resize the columns by dragging the column divider(s).

- **Enable Column Reordering**: Provides the app user with the ability to reorder the columns by dragging and dropping the column headers.

- **Enable Sorting**: Provides the app user with the ability to reorder rows by ascending or descending value in a given column.

- **Events**:

  - **Row Select Event Function**: The name of a JavaScript function that runs when a parent row is selected by the app user.

  - **Data Bound Event Function**: The name of a JavaScript function that runs when the parent rows for the grid are bound to its parent data source.

  - **Detail Init Event Function**: The name of a JavaScript function that runs before the child rows for a parent expand.

  - **Detail Expand Event Function**: The name of a JavaScript function that runs after the child rows for a parent expand.

  - **Detail Collapse Event Function**: The name of a JavaScript function that runs after the user collapses the child rows for a parent.

The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

- **Child Grid Properties**:

  - **Grid Title**: The title that appears for each child grid when this view is displayed on the screen for the app.

  - **Edit Mode**:

    - **ReadOnly**: This mode supports read-only access to data in the grid.

    - **Incell**: This mode supports edits to multiple rows at a time in the grid. Data editing is supported using the fields of the grid itself. You initiate editing by clicking a column within a row. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides buttons to create a new row, and to save all changes or cancel all changes to the rows in the grid.

    - **Inline**: This mode supports edits to a single row at a time in the grid. Data editing is supported using the fields of the row itself. You initiate editing by clicking an edit button on the row you want to update. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides a single button to create a new row in the grid.

    - **Popup**: This mode supports edits to a single row at a time in the grid. Data editing is supported using a pop-up form. You initiate editing by clicking an edit button on the row you want to update. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides a single button to create a new row in the grid.

  - **Edit Options**:

    - **Allow Insert**: An option that allows new rows to be created in the grid.

    - **Allow Edit**: An option that allows existing rows to be edited in the grid.

    - **Allow Delete**: An option that allows existing rows to be deleted in the grid.

    - **Toolbar Button Labels**:

      - **Cancel**: A custom label for the button that cancels all pending edits in the grid. The default is "Cancel changes".

      - **Create**: A custom label for the button that creates a new row in the grid. The default is "New".

      - **Save**: A custom label for the button that saves all pending edits in the grid. The default is "Save changes".

    - **Row Button Labels**:

      - **Cancel Edit**: A custom label for the button that cancels all pending edits in the row. The default is "Cancel".

      - **Delete**: A custom label for the button that deletes the row. The default is "Delete".

      - **Edit**: A custom label for the button that initiates editing in the row. The default is "Edit".

      - **Update**: A custom label for the button that confirms any pending delete or edits to the row. The default is "Update".

- **Grid Columns**: Click **Edit** to change the included columns and various properties of the columns. See Editing grid columns on page 23 for more information.

- **Page Size**: The number of rows that will be displayed per page in the grid.

- **Row Template Function Name**: The name of a JavaScript function that returns the text for a Kendo UI rowTemplate. If specified, the row template is used to format all the rows in the grid unless you also define an altRowTemplate. For more information, see the entry for rowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-rowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*.

- **Row Template Id**: The ID of an HTML rowTemplate. If specified, the rowTemplate must be defined in either the
  `application-folder`\app\src\modules\`module-folder`\`view-folder`\topSection.html file, the \topParentSection.html file, the \middleSection.html file, the \topChildSection.html file, or the \bottomSection.html file. If both a Row Template Function and a Row Template Id are specified, the Row Template Id takes precedence.

- **Alt Row Template Function Name**: The name of a JavaScript function that returns the text for a Kendo UI altRowTemplate. If specified, the altRowTemplate is used to format every other row in the grid. For more information, see the entry for altRowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-altRowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*

- **Alt Row Template Id**: The ID of an HTML altRowTemplate. If specified, the altRowTemplate must be defined in either the
  `application-folder`\app\src\modules\`module-folder`\`view-folder`\topSection.html file, the \middleSection.html file, or the \bottomSection.html file. If both an Alt Row Template Function and an Alt Row Template Id are specified, the Row Template Id takes precedence.

- **Enable Column Filtering**: Provides the app user with the ability to filter the displayed information based on content, including filters for numeric values (e.g., is equal to, is not equal to) and text fields (e.g., starts with, does not contain, etc.).

- **Enable Grouping**: Provides the app user with the ability to group the rows according to the value of a particular column. For example, in a grid with columns representing customer names and order numbers, grouping by customer name will arrange the rows so that all of the order numbers from each given customer are grouped together. Grouping should only be used with client-side processing.

- **Enable Column Resize**: Provides the app user with the ability to resize the columns by dragging the column divider(s).

- **Enable Column Reordering**: Provides the app user with the ability to reorder the columns by dragging and dropping the column headers.

- **Enable Sorting**: Provides the app user with the ability to reorder rows by ascending or descending value in a given column.

- **Selection Type**: Determines if rows in a grid can be selected:

  - **None**: No row selection is allowed (default).

  - **Single**: One row can be selected at a time.

  - **Multiple**: Multiple rows can be selected at one time.

Note: Row selection has no function except to invoke the Row Select Event for each selected row.

- **Events**:

  - **Row Select Event Function**: The name of a JavaScript function that runs when a child row is selected by the app user.

  - **Data Bound Event Function**: The name of a JavaScript function that runs when the child rows for the grid are bound to its child data source.

  - **Row Create Event Function**: The name of a JavaScript function that runs before a row is created for a new data source record.

  - **Row Update Event Function**: The name of a JavaScript function that runs before an existing data source record is updated in the row.

  - **Row Delete Event Function**: The name of a JavaScript function that runs before an existing data source record is deleted in the row.

  The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

This view also includes the following custom HTML sections where you can include your own HTML for the view. Each section is identified in the view design page with a placeholder containing descriptive text:

- **Custom top html section:** Located in the area above the parent Grid Title, the text in this placeholder introduces the view path name of the topSection.html file where you can add your custom HTML.

- **Custom top parent html section:**: Located in the area below the parent Grid Title but above the parent grid itself, the text in this placeholder introduces the view path name of the topParentSection.html file where you can add your custom HTML.

- **Custom top child html section:**: Located in the area below each child Grid Title, immediately under the related row of the parent grid, but above the child grid itself, the text in this placeholder introduces the view path name of the topChildSection.html file where you can add your custom HTML.

- **Custom bottom child html section:**: Located in the area below each child grid but above either the next parent row or the bottom area of the parent grid, the text in this placeholder introduces the view path name of the bottomChildSection.html file where you can add your custom HTML.

- **Custom bottom html section:**: Located in the area below the parent grid, the text in this placeholder introduces the view path name of the bottomSection.html file where you can add your custom HTML.

The Designer automatically generates these three files for you in *application-folder*\app\src\modules*module-folder**view-folder*. Add your custom HTML code to these files as required.

See also

Blank view

Selecting the Blank view gives you the ability to define a custom layout and functionality of a view. The Blank view allows you to:

- Configure one or more data source instances for the view.

- Drag-and-drop rows and columns to create a responsive layout.

- Drag-and-drop content components for data management, editing, charting, scheduling, and navigation.

The first step for working with a Blank view is typically to define data source instances for the view that you can bind to most of its components. You define these view data source instances from one or more of the existing data sources that are defined for your app data providers.

Most data management, editing, charting, and scheduling components allow you to specify one of these view data source instances as the data binding for that component. When any other component is bound to the same data source instance as a data management component, the data that you access using the data management component is automatically displayed using other components that are bound to the same data source instance, and this data linkage between data management and other components is supported without any further coding.

For example, when you display a page of records (as defined by the data source instance) using the grid component, this same page of records is charted using any chart component that is bound to the same data source instance. If you change a value in a record displayed using this grid, the chart is automatically redisplayed showing any updates to charted values. Note that to manage editing of data using components other than the grid component, you must code appropriate event functions available for these components. You also have the option to code editing event functions for the grid component as well.

On a Blank view design page, lists of layout and content components appear in a **COMPONENTS** palette on the left. Properties for the view appear in a **VIEW PROPERTIES** palette on the right, and properties for any selected component that you have dragged-and-dropped on the layout area appear in a **COMPONENT PROPERTIES** palette directly below **VIEW PROPERTIES**. Between them, in the middle, is the layout area where you drag-and-drop components for the view.

See also

Adding data source instances for a Blank view

From the **View Data Sources** dialog, you can review and add one or more data source instances to be used in your Blank view. These instances are defined based on the data sources defined for your app but are scoped to a particular Blank view. To open and use this dialog:

1. Click the **Edit** button next to **View Data Sources** property at the top of the **VIEW PROPERTIES** palette.

2. Click **Add Data Item** to add a new data source instance from the data sources available for your app, or select an existing data source to review its details.

3. Enter a **Name** for the data source. This is the name you will select as a **Data Source** or **Model** when you want to associate a component with a data source. (The associated model name is the data source name appended with "Model". *Models* are explained further below.)

4. Select a **Data Provider** and **Data Source**. Any provider or source you created for your app from the app design page are selectable here.

5. Choose a **Page Size**. This number indicates how many records per page from your **Data Source** will be available to any component you associate with the data source instance. For example, if you bind this instance to a grid (or other data management) component, it displays as many records per page at run time as you specify here. Any other component in the view that is bound to the same data source instance also displays data from that same page of records, unless you add custom code to do otherwise.

6. Click **Save** when done. Note that you can delete existing data sources by using the trash can icons.

A data source is used with components that work with a set of records. From an OpenEdge ABL point of view, you can think of a data source as a temp-table. When a component works directly with a data source, the binding is two-way, and the JSDO table represented by the data source and the UI data are always in sync. Often, the component can call read/save directly. The following two components are data-bound and use the data source directly:

- Grid

- List View

These components can also be bound, instead, through a model, as described below.

Note: All of the data-bound and model-bound components and their properties are described in Adding content on page 51.

A *model* (or *data model*) is used to represent the values of a single data source record. The Kendo UI Builder automatically creates a data model for every data source instance created for a view. From an ABL point of view, you can think of a model for a data source instance as a buffer for a temp-table. When a component works with a data model, the binding is two-way, and the UI and the data are always in sync. However, there is NO built-in association between the model and any JSDO table. You, the developer, must manage the movement of values between the model and the JSDO programmatically in an appropriate event function.

This simplifies the case where you create a form, whether or not it is associated with a data-bound component. For example, to create a view with a grid and a form, the grid is bound to the data source and the form fields are "bound" through their data model. When a row is selected in the grid, the framework automatically copies the values from the data source into the data model of the same name, so the currently selected record is shown in the form. When you edit the form fields, the data model is updated; however, these changes are not automatically pushed back to the data source and its JSDO table. You must handle this behavior in an appropriate event function. For more information, see Kendo UI Builder by Progress: Modernizing OpenEdge Applications.

In addition to the Grid and List View, the components that use a data model include most of the editor components (with the exception of the Label). Some of these editor components (*list editors*) have both a data source and a model:

- Combo Box

- Drop Down List

- Radio Bullet List

The data source represents the list of available values from which to select, and the data model holds the selected item. You can think of a Combo Box, Drop Down List, and Radio Bullet List as a list of rows (data source) combined with a single column (value field) to represent the selected row (data model).

All the remaining editor components are model-bound only. Most of these model-bound components have two fields to specify their **Model** property, one field to specify the data model itself and a second field to specify the field from that model whose value the editor displays or updates.

In addition, the list components also have the option to specify a single field from their model to represent the value of the component. They all have a **Value Primitive** checkbox property that when selected provides a two-field **Model** property similar to other model-bound editor components to select both the data model and its value field.

One other model-bound editor component, Auto Complete, also has a data source as well as a model, but unlike the list editors, this data source is **not** the data source from which its value is bound. Like other editors without a data source, it relies on a model to bind its value. However, it has a different data source from which options are provided to auto-complete any value the user enters into the component field. The actual value of the Auto Complete itself (the value the user finishes entering) is bound through the model. In addition, Auto Complete has a **Value Primitive** property to provide an additional **Model** property field to specify the single field from the data model whose value the component represents.

Kendo UI Builder by Progress : Using Kendo UI Designer: Version 2.0

Note: To use a similar programming model for all the model-bound editors, including the list editors, you might always select the **Value Primitive** checkbox, if the component has one, to identify a value field. Otherwise, your code must otherwise identify the model field to represent the component value.

Finally, depending on your application, where a component has a data source as well a model to bind its value, you can select a separate data model that is associated with a different data source from the one you specify for the component. For example, similar to Auto Complete, you might specify a model for a separate data source that your code needs to query in order to process whatever value the user enters to update the component data source.

Editing the layout

The blank view supports a responsive layout for different screen sizes. Four screen sizes are supported for the blank view: Desktop, Laptop, Tablet Landscape and Tablet Portrait. The two basic layout elements in the blank view are rows and columns. You can drag and drop these elements from the **Components** palette:

- **Row**: Rows act as containers for columns. They have no additional properties of their own. You can move or delete existing rows, and you can also nest one or more rows inside a column if that column is either empty or already contains one or more rows.

- **Column**: Once a row has been added, you can drag and drop one or more columns into that row. You can delete columns and their contents by clicking on the thick top border and then clicking the trash can icon. You can also drag a column (and their contents) by its thick top border to different positions in the layout.

Note: Dragging any other component into a row from the elements available for data management, editors, charts, scheduling, navigation, and custom automatically creates a column for it if you have not already dragged and dropped one.

You can access a column's properties by clicking the thick border at the top of the column. The properties for a column are:

- **Screen widths**: You can specify, or change the width (**Column span**) of the column for each screen size. There are 12 "slots" available for columns in each row, and a column can take up one or more slots. For example, you can have 12 columns that span one slot each, or three columns that span four slots each.

  Note that the Tablet Landscape width is required and will be used for all screen sizes unless you enter a different width for the other screen sizes (Desktop, Laptop, and Tablet Portrait).

- **Hide on**: You can hide a column on a given screen size by selecting the checkbox for that screen width. You can use this feature to provide more details on larger screens while minimizing the content shown for smaller devices.

Use the display type preview at the top right of the toolbar to view the results of these changes.

Adding content

The Blank view offers a variety of data management, editing, scheduling, and navigational components. All of the components have an Id property:

- **Id**: A unique string with a default value that is automatically generated by the Designer for each component. You can use this to associate two components, like a label and a check-box, for example. Drag and drop a label and check-box, and then input the check-box's ID into the **For** property of the label to attach that label to the check-box. You can edit the Id to a more meaningful value as you require; however, it cannot be empty and must be unique among all component Id values in the view.

Like the predefined, data-driven views, the Blank view also provides the option for a custom section:

- **Custom Section**: You can add your own custom HTML code in this section. The Designer automatically generates a `topSection.html` file for you in the folder *application-folder*\app\src\modules*module-folder**view-folder*. Add your custom HTML code to this file as desired. This file is typically used to add template code for the ***Template*** properties of content components added to the Blank view.

Note: Unlike the custom section files generated for the predefined, data-driven views, none of the HTML content added to the the Blank view's `topSection.html` file is visible in the view.

Note: This topic covers data management, editing, charting, and scheduling components. The navigation components are described in .

Table 3: Data management components

Component	Properties and notes

Grid	A grid component similar to the grid provided with the pre-defined Data-Grid view.
	
	• **Data Source Name**: A grid data source instance that you have defined for the Blank view.
	
	• **Edit Mode**:
	
	  • **ReadOnly**: This mode supports read-only access to data in the grid.
	
	  • **Incell**: This mode supports edits to multiple rows at a time in the grid. Data editing is supported using the fields of the grid itself. You initiate editing by clicking a column within a row. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides buttons to create a new row, and to save all changes or cancel all changes to the rows in the grid.
	
	  • **Inline**: This mode supports edits to a single row at a time in the grid. Data editing is supported using the fields of the row itself. You initiate editing by clicking an edit button on the row you want to update. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides a single button to create a new row in the grid.
	
	  • **Popup**: This mode supports edits to a single row at a time in the grid. Data editing is supported using a pop-up form. You initiate editing by clicking an edit button on the row you want to update. Each row also provides a button to delete the associated row in the grid. In this mode, the toolbar above the grid provides a single button to create a new row in the grid.
	
	• **Edit Options**:
	
	  • **Allow Insert**: An option that allows new rows to be created in the grid.
	
	  • **Allow Edit**: An option that allows existing rows to be edited in the grid.
	
	  • **Allow Delete**: An option that allows existing rows to be deleted in the grid.
	
	  • **Toolbar Button Labels**:

Component	Properties and notes
	Cancel: A custom label for the button that cancels all pending edits in the grid. The default is "Cancel changes". **Create**: A custom label for the button that creates a new row in the grid. The default is "New". **Save**: A custom label for the button that saves all pending edits in the grid. The default is "Save changes". **Row Button Labels**: **Cancel Edit**: A custom label for the button that cancels all pending edits in the row. The default is "Cancel". **Delete**: A custom label for the button that deletes the row. The default is "Delete". **Edit**: A custom label for the button that initiates editing in the row. The default is "Edit". **Update**: A custom label for the button that confirms any pending delete or edits to the row. The default is "Update". **Grid Columns**: Click **Edit** to change the included columns and various properties of the columns. See Editing grid columns on page 23 for more information. **Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI rowTemplate. If specified, the row template is used to format all the rows in the grid unless you also define an altRowTemplate. For more information, see the entry for rowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-rowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*. **Alt Row Template Function**: The name of a JavaScript function that returns the text for a Kendo UI altRowTemplate. If specified, the altRowTemplate is used to format every other row in the grid. For more information, see the entry for altRowTemplate at http://docs.telerik.com/kendo-ui/api/javascript/ui/grid#configuration-altRowTemplate, in the *Kendo UI by Progress Documentation and API Reference*. For more information on defining a rowTemplate for a Kendo UI Builder editable grid, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* **Row Template Id**: The ID of an HTML rowTemplate. If specified, the rowTemplate must be defined in either the `application-folder\app\src\modules\module-folder\view-folder \topSection.html` file, the `\middleSection.html` file, or the `\bottomSection.html` file. If both a Row Template Function and a Row Template Id are specified, the Row Template Id takes precedence. **Alt Row Template Id**: The ID of an HTML altRowTemplate. If specified, the altRowTemplate must be defined in either the `application-folder\app\src\modules\module-folder\view-folder \topSection.html` file, the `\middleSection.html` file, or the

Component	Properties and notes

	`\bottomSection.html` file. If both an Alt Row Template Function and an Alt Row Template Id are specified, the Row Template Id takes precedence.
	
	• **Enable Column Filtering**: Provides the app user with the ability to filter the displayed information based on content, including filters for numeric values (e.g., is equal to, is not equal to) and text fields (e.g., starts with, does not contain, etc.).
	
	• **Enable Grouping**: Provides the app user with the ability to group the rows according to the value of a particular column. For example, in a grid with columns representing customer names and order numbers, grouping by customer name will arrange the rows so that all of the order numbers from each given customer are grouped together. Grouping should only be used with client-side processing.
	
	• **Enable Column Resize**: Provides the app user with the ability to resize the columns by dragging the column divider(s).
	
	• **Enable Column Reordering**: Provides the app user with the ability to reorder the columns by dragging and dropping the column headers.
	
	• **Enable Sorting**: Provides the app user with the ability to reorder rows by ascending or descending value in a given column.
	
	• **Selection Type**: Determines if rows in a grid can be selected:
	  • **None**: No row selection is allowed (default).
	  • **Single**: One row can be selected at a time.
	  • **Multiple**: Multiple rows can be selected at one time.
	
	Note: Row selection has no function except to invoke the Row Select Event for each selected row.
	
	• **Model**: The grid component can be bound directly to a data source or associated with a data source through a model. For the latter, select the model instance with this menu.
	
	• **Events**:
	  • **Row Select Event Function**: The name of a JavaScript function that runs when a grid row is selected by the app user.
	  • **Data Bound Event Function**: The name of a JavaScript function that runs when the grid is bound to its data source.
	  • **Row Create Event Function**: The name of a JavaScript function that runs before a row is created for a new data source record.
	  • **Row Update Event Function**: The name of a JavaScript function that runs before an existing data source record is updated in the row.
	  • **Row Delete Event Function**: The name of a JavaScript function that runs before an existing data source record is deleted in the row.

Component	Properties and notes
	The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

List View	A list component that presents data for display and editing from the records of a selected data source based on separate display and edit templates.
	
	• **Data Source Name**: A list view data source instance that you have defined for the Blank view.
	
	• **Template Function**: The list view requires custom code to display data. See the code example below this table. Any function you name in this field must be defined in *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js.
	
	• **Template Id**: The Id of an HTML row template. If specified, the row template must be located in the *application-folder*\app\src\modules*module-folder**view-folder*\topSection.html file.
	
	• **Edit Template Function**: A function used when the list view is in edit mode. Any function you name in this field must be defined in *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js.
	
	• **Edit Template Id**: The Id of an HTML row template when the list view is in edit mode. If specified, the row template must be located in the *application-folder*\app\src\modules*module-folder**view-folder*\topSection.html file.
	
	• **Selection Type**: Determines if items in the list view can be selected:
	
	  • **None**: No item selection is allowed (default).
	
	  • **Single**: One item can be selected at a time.
	
	  • **Multiple**: Multiple items can be selected at one time.
	
	Note: Item selection has no function except to invoke the Select Event for each selected item.
	
	• **Model**: The list view component can be bound directly to a data source or associated with a data source through a model. For the latter, select the model instance with this menu.
	
	• **Events**:
	
	  • **Cancel Event Function**: The name of a JavaScript function that runs when the user cancels an operation.
	
	  • **Select Event Function**: The name of a JavaScript function that runs when the user selects an item in the list view.

Component	Properties and notes
	• **Data Bound Event Function**: The name of a JavaScript function that runs when the list view is bound to its data source. • **Edit Event Function**: The name of a JavaScript function that runs when the user makes an edit to an item in the list view. • **Remove Event Function**: The name of a JavaScript function that runs when the user removes an item from the list view. • **Save Event Function**: The name of a JavaScript function that runs when the user saves a change to the list view. The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Sample template function for a list view:

```
templateFunction: function(data) {
    var template = kendo.template("<div id='box'>#: Name #</div>");
    var result = template(data); //Passing the data to the template
    return result;}
```

Table 4: Editor components

Component	Properties, events, and notes

Auto Complete	A text box that automatically suggests options for a value based on the user's typed input.
	
	• **Data Source Name**: A data source instance that you have defined for the Blank view. The value you enter is entered and displayed from this data source.
	
	• **Model**: The data model associated with this component.
	
	• **Data Text Field**: A field from the data source associated with the component. This is the field that displays the value from the data source that you can enter or change.
	
	• **Title Text**: This optional text appears when a user hovers over the component.
	
	• **Value Primitive**: Restricts the model to one field of the record.
	
	• **Filter Method**: If the user begins typing in the auto complete, the choices available will be limited as follows:
	
	  • **StartsWith**: Values that begin with the first characters they type. For example, if they type w, the options will be restricted to those values that begin with "w".
	
	  • **EndsWith**: Values that end with the last characters they type. For example, if they type e as the last character, the options will be restricted to those values that end with "e".

Component	Properties, events, and notes

	• **Contains**: Values that contain the characters they type. For example, if they type `ove`, the options will be restricted to those values that contain "ove".
	
	• **Events**:
	
	  • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value.
	
	  • **Filtering Event Function**: The name of a JavaScript function that runs after the user filters the results.
	
	  • **Select Event Function**: The name of a JavaScript function that runs when the user selects a value among the choices offered.
	
	The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Boolean Radio Button List	A radio button set with two possible boolean values (true or false). • **Default Value**: Sets the initial state of the buttons. Check the box to set the True button value, or leave unchecked to set the False button value (the default). • **True text**: The label that appears next to the True button ("True" by default). • **False text**: The label that appears next to the False button ("False" by default). • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Check Box	A box that displays with a check mark or empty, depending on the boolean value of the component. • **Value**: Check this box to set the initial state of the component to true, or leave the box empty to set its initial state to false. • **Disabled**: Check this box to disable the check box, or leave it unchecked to make the check box active. • **Model**: The data model associated with this component. • **Title Text**: This optional text appears when a user hovers over the component.

Component	Properties, events, and notes
	• **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Combo Box	This component binds to both a data source instance and a model to display a list of values that the user can both select and edit.
	
	• **Data Source Name**: A data source instance that you have defined for the Blank view. The list of values for the combo box are loaded from this data source.
	
	• **Model**: The data model associated with this component. The value initially displayed in the combo box is loaded from this model.
	
	• **Data Text Field**: A field from the data source associated with the component. This is the field displayed in the combo box.
	
	• **Data Value Field**: A field from the data source that uniquely identifies the selected item. For example, employee records could have an Employee Number field that uniquely identifies employees with the same last name.
	
	• **Title Text**: This optional text appears when a user hovers over the component.
	
	• **Value Primitive**: Restricts the model to one field of the record.
	
	• **Filter Method**: If the user begins typing in the combo box, the choices available will be limited as follows:
	
	   • **None**: No filtering of values.
	
	   • **StartsWith**: Values that begin with the first characters they type. For example, if they type w, the options will be restricted to those values that begin with "w".
	
	   • **EndsWith**: Values that end with the last characters they type. For example, if they type e as the last character, the options will be restricted to those values that end with "e".
	
	   • **Contains**: Values that contain the characters they type. For example, if they type ove, the options will be restricted to those values that contain "ove".
	
	• **Events**:
	
	   • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value.
	
	   • **Filtering Event Function**: The name of a JavaScript function that runs after the user filters the results.

Component	Properties, events, and notes
	• **Select Event Function**: The name of a JavaScript function that runs when the user selects a value among the choices offered. The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Currency Text Box	A specialized version of a text-box for currency. • **Decimals**: The number of decimal places displayed. • **Default Value**: The amount initially displayed in the text box. The user can edit or delete this amount as needed. • **Down Arrow Text**: The text shown when the user hovers over the down arrow. • **Min** and **Max**: The minimum and maximum values allowed, respectively. • **Model**: The data model associated with this component. • **Placeholder**: The amount that initially appears in the text box. Unlike Default Value, this text automatically disappears when the user clicks on the text box. • **Step**: The interval the entered value changes for each click of the down or up arrow. • **Title Text**: This optional text appears when a user hovers over the component. • **Up Arrow Text**: The text shown when the user hovers over the up arrow. • **Show Spinners**: Select to display up and down arrows that change the field value when clicked by the increasing or decreasing **Step**-specified interval. Not selected by default. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Date Picker	A specialized drop down component that allows you to select a date value from a drop down calendar. • **Min** and **Max Date**: The minimum (earliest) and maximum (latest) dates that can be displayed. • **Default Date**: The date initially displayed. • **Date Format**: The format the date is displayed in once the user has made a selection.

Component	Properties, events, and notes
	• **Title Text**: This optional text appears when a user hovers over the component. • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Date Time Picker	A specialized drop down component that allows you to select a date and time value from a drop down calendar. • **Min** and **Max Date**: The minimum (earliest) and maximum (latest) dates and times that can be displayed. • **Default Date**: The date and time initially displayed. • **Date Format**: The format for how the date and time are displayed. • **Time Interval**: The interval, in minutes, between selectable values in the time drop-down menu, e.g., a value of 30 gives users the choice of 1:00, 1:30, 2:00, etc. • **Title Text**: This optional text appears when a user hovers over the component. • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Disabled Text Box	Use the disabled text box to display read-only text. • **Default Value**: The text initially displayed in the text box. You can edit or delete this text as needed. • **Model**: The data model associated with this component. • **Title Text**: This optional text appears when a user hovers over the component.

Component	Properties, events, and notes

Drop Down List	This component binds to both a data source instance and a model to display a list of values that the user can select.
	
	• **Data Source Name**: A data source instance that you have defined for the Blank view. The list of values for the combo box are loaded from this data source.
	
	• **Model**: The data model associated with this component. The value displayed in the drop-down list is initially loaded from this model.
	
	• **Data Text Field**: A field from the data source or model associated with the component. This is the field displayed in the drop down list.
	
	• **Data Value Field**: A field from the data source that uniquely identifies the selected item. For example, employee records could have an Employee Number field that uniquely identifies employees with the same last name.
	
	• **Value Primitive**: Restricts the model to one field of the record.
	
	• **Default Item**: The item initially displayed in the drop down list. The user can then change the selected item as needed.
	
	• **Title Text**: This optional text appears when a user hovers over the component.
	
	• **Events**:
	
	  • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value.
	
	  • **Select Event Function**: The name of a JavaScript function that runs when the user selects a value among the choices offered.
	
	The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Editor	A customizable HTML editor.
	
	• **Encoded**: Indicates whether the editor submits encoded HTML.
	
	• **Resizable**
	
	  • **Content**: If checked, the user can resize the editor.
	
	  • **Min** and **Max**: The smallest and largest size the user can choose for the size of the editor.
	
	  • **Toolbar**: When checked, the formatting tools are limited to one row at the top of the editor, with the tools that do not fit organized into a drop-down menu. When unchecked, the tools are organized into as many rows as necessary.
	
	• **Title Text**: This optional text appears when a user hovers over the component.
	
	• **Tools**: Select one or more formatting tools available to your users. Once selected, any tool can be removed by clicking the **X** next to its name.

Component	Properties, events, and notes
	• **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*`\app\src\modules\`*module-folder*`\`*view-folder*`\controller.public.js.` See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Email Text Box	A specialized version of a text-box for e-mail addresses. • **Debounce**: The delay, in milliseconds, between user input and the running of the Change Event Function. • **Default Value**: The e-mail address initially displayed in the text box. The user can edit or delete this text as needed. • **Placeholder**: The e-mail address initially displayed in the text box. Unlike Default Value, this text automatically disappears when the user clicks on the text box. • **Title Text**: This optional text appears when a user hovers over the component. • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*`\app\src\modules\`*module-folder*`\`*view-folder*`\controller.public.js.` See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Integer Text Box	A specialized version of a text-box for integer values only. • **Default Value**: The integer initially displayed in the text box. The user can edit or delete this text as needed. • **Down Arrow Text**: The text shown when the user hovers over the down arrow. • **Min** and **Max**: The minimum and maximum values allowed, respectively. • **Placeholder**: The integer initially displayed in the text box. Unlike Default Value, this text automatically disappears when the user clicks on the text-box. • **Step**: The interval the entered value changes for each click of the down or up arrow. • **Title Text**: This optional text appears when a user hovers over the component.

Component	Properties, events, and notes
	• **Up Arrow Text**: The text shown when the user hovers over the up arrow. • **Model**: The data model associated with this component. • **Show Spinners**: Select to display up and down arrows that change the field value when clicked by the increasing or decreasing **Step**-specified interval. Not selected by default. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Label	A label for another Blank view component. • **For**: An Id of another component, which this label will be associated with. This is optional; for example, you can use a label as a user aid without tying it to a specific component. • **Text**: The actual text of the label.
Masked Text Box	A specialized version of a text-box that supports an input mask. • **Mask**: Specifies the input mask. See http://docs.telerik.com/kendo-ui/api/javascript/ui/maskedtextbox#configuration-mask for the supported mask rules. • **Prompt Char**: The character displayed in the mask before the user inputs anything. Defaults to "_". • **Default Value**: The text initially displayed in the text box. The Prompt Char value appears whenever the user deletes this text without replacing it with an acceptable value. • **Title Text**: This optional text appears when a user hovers over the component. • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Component	Properties, events, and notes
Numeric Text Box	A specialized version of a text-box for numeric values only. • **Decimals**: The number of decimals displayed. • **Default Value**: The number initially displayed in the text box. The user can edit or delete this text as needed. • **Down Arrow Text**: The text shown when the user hovers over the down arrow. • **Format**: The numeric format for the input. See http://docs.telerik.com/kendo-ui/api/javascript/ui/maskedtextbox#configuration-mask for the supported numeric types. • **Min** and **Max**: The minimum and maximum values allowed, respectively. • **Placeholder**: The number initially displayed in the text box. Unlike Default Value, this text automatically disappears when the user clicks on the text-box. • **Step**: The interval the entered value changes for each click of the down or up arrow. • **Title Text**: This optional text appears when a user hovers over the component. • **Up Arrow Text**: The text shown when the user hovers over the up arrow. • **Model**: The data model associated with this component. • **Show Spinners**: Select to display up and down arrows that change the field value when clicked by the increasing or decreasing **Step**-specified interval. Not selected by default. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Password Text Box	A specialized version of a text-box for passwords. • **Placeholder**: The text initially displayed in the text box. This text automatically disappears when the user clicks on the text box. • **Title Text**: This optional text appears when a user hovers over the component. • **Debounce**: The delay, in milliseconds, between user input and the running of the Change Event Function. • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value.

Component	Properties, events, and notes
	The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder* \controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Percent Text Box	A specialized version of a text-box for percent values only. Use this when the percent values in the database must be multiplied by 100 before appending the % symbol for proper display (e.g., the database value of 0.256 will be displayed as 25.6%). • **Decimals**: The number of decimal places displayed. • **Default Value**: The value that initially appears in the text box. The user can edit or delete this text as needed. • **Down Arrow Text**: The text shown when the user hovers over the down arrow. • **Min** and **Max**: The minimum and maximum values allowed, respectively. • **Placeholder**: The value that initially appears in the text box. Unlike initial text, this text automatically disappears when the user clicks on the text-box. • **Step**: The interval the entered value changes for each click of the down or up arrow. • **Title Text**: This optional text appears when a user hovers over the component. • **Up Arrow Text**: The text shown when the user hovers over the up arrow. • **Model**: The data model associated with this component. • **Show Spinners**: Select to display up and down arrows that change the field value when clicked by the increasing or decreasing **Step**-specified interval. Not selected by default. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder* \controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Percent Value Text Box	A specialized version of a text-box for percent values only. Use this when the percent values in the database only need the addition of the % symbol for proper display (e.g., the database value of 25 will be displayed as 25%). • **Decimals**: The number of decimal places displayed. • **Default Value**: The value that initially appears in the text box. The user can edit or delete this text as needed.

Component	Properties, events, and notes
	• **Down Arrow Text**: The text shown when the user hovers over the down arrow. • **Min** and **Max**: The minimum and maximum values allowed, respectively. • **Placeholder**: The value that initially appears in the text box. Unlike Default Value, this text automatically disappears when the user clicks on the text-box. • **Step**: The interval the entered value changes for each click of the down or up arrow. • **Title Text**: This optional text appears when a user hovers over the component. • **Up Arrow Text**: The text shown when the user hovers over the up arrow. • **Model**: The data model associated with this component. • **Show Spinners**: Select to display up and down arrows that change the field value when clicked by the increasing or decreasing **Step**-specified interval. Not selected by default. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*`\app\src\modules\`*module-folder\view-folder*`\controller.public.js`. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Phone Text Box	A specialized version of a text-box for phone numbers. • **Mask**: Specifies the input mask. For example, (000) 000-0000 allows the user to enter a standard U.S., 10-digit phone number. • **Default Value**: The value that initially appears in the text box. The user can edit or delete this text as needed. • **Title Text**: This optional text appears when a user hovers over the component. • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for this event function must be included in the file, *application-folder*`\app\src\modules\`*module-folder\view-folder*`\controller.public.js`. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Radio Button List	A radio button set with multiple values based on the instances of a specified field in the data source result set. This component binds to both a data source instance

Component	Properties, events, and notes
	and a model to provide field values for the buttons in the list. The selected button represents the value of the specified field from the corresponding record. • **Data Source Name**: A data source instance that you have defined for the Blank view. The list of values corresponding to the buttons in the list are loaded from this data source. • **Data Text Field**: A field from the data source or model associated with the component. This is the field displayed as the value of each button in the list. • **Data Value Field**: A field from the data source that uniquely identifies the selected item. For example, employee records could have an Employee Number field that uniquely identifies employees with the same last name. • **Value Primitive**: Restricts the model to one field of the record. • **Orientation**: Chooses a horizontal or vertical orientation for the button list. • **Height (px)**: Height of the button list in pixels. • **Max Height (px)**: Maximum height of the button list in pixels. • **Model**: The data model associated with this component. The values displayed in the button list are loaded from this model. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Slider	A rich input component for selecting numeric values. • **Min** and **Max**: The minimum and maximum values allowed, respectively. • **Orientation**: Chooses a horizontal or vertical orientation for the slider. • **Show Buttons**: Select to display decrease and increase arrow buttons on either end of the slider. Selected by default. • **Step**: The interval that determines which values on the slider are valid, e.g., a step of 0.5 on a range of 1 to 10 allows users to select 1, 1.5, 2, 2.5, etc. • **Default Value**: The number at which the slider is initially positioned. • **Title Text**: This optional text appears when a user hovers over the component. • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value.

Component	Properties, events, and notes
	• **Slide Event Function**: The name of a JavaScript function that runs when the user moves the drag handle. The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Text Area	A multi-line version of a text box. • **Default Text**: The text that initially appears in the text box. The user can edit or delete this text as needed. • **Placeholder**: Text that initially appears in the text box. Unlike Default Text, this text automatically disappears when the user clicks on the text box. • **Title Text**: This optional text appears when a user hovers over the component. • **Debounce**: The delay, in milliseconds, between user input and the running of the Change Event Function. • **Rows**: The number of lines of text visible in the text box. • **Columns**: The width of the text box measured in average character widths. • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Text Box	A general purpose text box. • **Default Text**: The text that initially appears in the text box. The user can edit or delete this text as needed. • **Placeholder**: Text that initially appears in the text box. Unlike Default Text, this text automatically disappears when the user clicks on the text box. • **Title Text**: This optional text appears when a user hovers over the component. • **Debounce**: The delay, in milliseconds, between user input and the running of the Change Event Function. • **Model**: The data model associated with this component. • **Events**:

Component	Properties, events, and notes
	• **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Time Picker	Allows users to select a time value from a predefined time drop-down menu or to enter a new time value. • **Default Time**: The time value initially displayed. • **Time Format**: The format for how the time value is displayed. • **Time Interval**: The interval, in minutes, between selectable values in the time drop-down menu, e.g., a value of 30 gives users the choice of 1:00, 1:30, 2:00, etc. • **Title Text**: This optional text appears when a user hovers over the component. • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Url Text Box	A specialized version of a text-box for URLs. • **Debounce**: The delay, in milliseconds, between user input and the running of the Change Event Function. • **Default Text**: The URL initially displayed in the text box. The user can edit or delete this text as needed. • **Placeholder**: The URL initially displayed in the text box. Unlike Default Text, this text automatically disappears when the user clicks on the text box. • **Title Text**: This optional text appears when a user hovers over the component. • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value.

Component	Properties, events, and notes
	The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Table 5: Chart components

Component	Properties and notes

Area Charts	Charts that plot one or more series of item values as the area under a line plotted along the item points for each series. The line and area for each item series gets a distinguishing color.
	
	• **Data Source Name**: A chart data source instance that you have defined for the Blank view.
	
	• **Title**:
	
	  • **Text**: A string that specifies a title for the chart.
	
	• **Legend**: Identifies each item series plotted in the chart by a key showing the value of its Item Name field (see the **Chart series** property) and a corresponding line and area color:
	
	  • **Background**: A color selection for the legend background of the chart. Choose a color that does not mask a generated legend color.
	
	  • **Position**: The location in the chart to display the legend, which you can select from **Top**, **Bottom**, **Left**, or **Right**.
	
	  • **Visible**: Indicates if the legend is displayed for the chart.
	
	Note: The user can identify a corresponding series in the chart by clicking its legend key, which alternately removes and re-inserts its plotted line and area in the chart.
	
	• **Value axis configuration**: Configures the axis for plotting series item values in the chart:
	
	  • **Labels**: Presentation of the major item increment values along this axis:
	
	    • **Format**: This can be used to customize how the increment values appear along the axis. Use the argument {0} to represent the increment value. You can also add additional text to be displayed. For example, if the axis is used to plot numeric values measured in kilograms, `{0} kg` appends " kg" to every major increment. You can also add specifiers to control the format of the increment values. If the increments are date values, you can specify that they display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:
	
	    `{0:D}`

Component	Properties and notes

	See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.
	
	• **Rotation of the angle**:
	
	   • **Angle**: The angle of rotation for displaying the increment values.
	
	• **Line**:
	
	   • **Visibility of line**: Indicates if the axis line is displayed.
	
	• **Major unit**: The size of the major increments labeled along the axis.
	
	• **Max axis**: The maximum item value plotted on the axis.
	
	• **Min axis**: The minimum item value plotted on the axis.
	
	• **Category axis**: Configures the axis used to identify a category for each plotted item:
	
	   • **Field**: The name of a data source field that identifies the category for the item value. For example, a Country field in a Customer data source that identifies a national category for Balance item values.
	
	   • **Labels**: Presentation of the category values along this axis:
	
	      • **Format**: This can be used to customize how the categories appear along the axis. Use the argument {0} to represent the category Field value. You can also add additional text to be displayed. For example, if the axis is used to plot numeric values that identify an age in years, `{0} years` appends " years" to every Field value. You can also add specifiers to control the format of the Field values. If the Field represents date values, you can specify that they display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:
	
	      `{0:D}`
	
	      See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.
	
	   • **Rotation of the angle**:
	
	      • **Angle**: The angle of rotation for displaying the category values.
	
	• **Series default settings**: Default settings for displaying series item values in the chart:
	
	   • **Labels setting**: Presentation of every series item label:

Component	Properties and notes

	• **Visible**: Indicates if a label is displayed for the plot point of each item value.
	
	• **Chart series**: Click **Edit** to add or change each series item to plot, including the properties of an item. See Editing chart series on page 83 for more information.
	
	• **Tooltip settings**: Settings for displaying the item value when the pointer hovers over a plot point:
	
	  • **Format**: This can be used to customize how the tooltip for each item value appears over its plot point. Use the argument {0} to represent the item value. You can also add additional text to be displayed. For example, if the items are numeric values measured in kilograms, `{0} kg` appends " kg" to every item value. You can also add specifiers to control the format of the increment values. If the increments are date values, you can specify that they display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:
	    `{0:D}`
	  See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.
	
	  • **Template Function**: Tooltips require custom code to display the item value. See the code example below this table. Any function you name in this field must be defined in *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js.
	
	  • **Template Id**: The Id of an HTML <div> template for item tooltips. If specified, the template must be located in the *application-folder*\app\src\modules*module-folder**view-folder*\topSection.html file.
	
	  • **Visible**: Indicates if each item value is displayed in the chart as a tooltip.
	
	• **Model**: The chart component can be bound directly to a data source or associated with a data source through a model. For the latter, select the model instance with this menu.
	
	• **Events**:
	
	  • **Data Bound Event Function**: The name of a JavaScript function that runs when the chart is bound to its data source.
	
	  • **Click Event Function**: The name of a JavaScript function that runs when the user clicks on an item series in the chart.
	
	The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Component	Properties and notes

Bar Charts	Charts that plot one or more series of item values as either horizontal bars or vertical columns (depending on the chart type) plotted along the item points for each series. The bar or column for each item series gets a distinguishing color.
	
	• **Data Source Name**: A chart data source instance that you have defined for the Blank view.
	
	• **Title**:
	
	    • **Text**: A string that specifies a title for the chart.
	
	• **Legend**: Identifies each item series plotted in the chart by a key showing the value of its Item Name field (see the **Chart series** property) and a corresponding bar or column color:
	
	    • **Background**: A color selection for the legend background of the chart. Choose a color that does not mask a generated legend color.
	
	    • **Position**: The location in the chart to display the legend, which you can select from **Top**, **Bottom**, **Left**, or **Right**.
	
	    • **Visible**: Indicates if the legend is displayed for the chart.
	
	Note: The user can identify a corresponding series in the chart by clicking its legend key, which alternately removes and re-inserts its plotted bars or columns in the chart.
	
	• **Value axis configuration**: Configures the axis for plotting series item values in the chart:
	
	    • **Labels**: Presentation of the major item increment values along this axis:
	
	        • **Format**: This can be used to customize how the increment values appear along the axis. Use the argument {0} to represent the increment value. You can also add additional text to be displayed. For example, if the axis is used to plot numeric values measured in kilograms, `{0} kg` appends " kg" to every major increment. You can also add specifiers to control the format of the increment values. If the increments are date values, you can specify that they display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:
	
	        `{0:D}`
	
	        See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.
	
	    • **Rotation of the angle**:
	
	        • **Angle**: The angle of rotation for displaying the increment values.
	
	    • **Line**:

Component	Properties and notes

	• **Visibility of line**: Indicates if the axis line is displayed.
	
	• **Major unit**: The size of the major increments labeled along the axis.
	
	• **Max axis**: The maximum item value plotted on the axis.
	
	• **Min axis**: The minimum item value plotted on the axis.
	
	• **Category axis**: Configures the axis used to identify a category for each plotted item:
	
	  • **Field**: The name of a data source field that identifies the category for the item value. For example, a Country field in a Customer data source that identifies a national category for Balance item values.
	
	  • **Labels**: Presentation of the category values along this axis:
	
	    • **Format**: This can be used to customize how the categories appear along the axis. Use the argument {0} to represent the category Field value. You can also add additional text to be displayed. For example, if the axis is used to plot numeric values that identify an age in years, `{0} years` appends " years" to every Field value. You can also add specifiers to control the format of the Field values. If the Field represents date values, you can specify that they display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:
	
	`{0:D}`
	
	See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.
	
	    • **Rotation of the angle**:
	
	      • **Angle**: The angle of rotation for displaying the category values.
	
	• **Series default settings**: Default settings for displaying series item values in the chart:
	
	  • **Labels setting**: Presentation of every series item label:
	
	    • **Visible**: Indicates if a label is displayed for the plot point of each item value.
	
	  • **Type**: A choice of two chart types:
	
	    • **Bar**: Horizontal item bars plotted along a horizontal value axis and vertical category axis.
	
	    • **Column**: Vertical item columns plotted along a vertical value axis and horizontal category axis.

Component	Properties and notes

	• **Chart series**: Click **Edit** to add or change each series item to plot, including the properties of an item. See Editing chart series on page 83 for more information.
	
	• **Tooltip settings**: Settings for displaying the item value when the pointer hovers over a plot point:
	
	  • **Format**: This can be used to customize how the tooltip for each item value appears over its plot point. Use the argument {0} to represent the item value. You can also add additional text to be displayed. For example, if the items are numeric values measured in kilograms, `{0} kg` appends " kg" to every item value. You can also add specifiers to control the format of the increment values. If the increments are date values, you can specify that they display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:
	
	    `{0:D}`
	
	  See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.
	
	  • **Template Function**: Tooltips require custom code to display the item value. See the code example below this table. Any function you name in this field must be defined in *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js.
	
	  • **Template Id**: The Id of an HTML <div> template for item tooltips. If specified, the template must be located in the *application-folder*\app\src\modules*module-folder**view-folder*\topSection.html file.
	
	  • **Visible**: Indicates if each item value is displayed in the chart as a tooltip.
	
	• **Model**: The chart component can be bound directly to a data source or associated with a data source through a model. For the latter, select the model instance with this menu.
	
	• **Events**:
	
	  • **Data Bound Event Function**: The name of a JavaScript function that runs when the chart is bound to its data source.
	
	  • **Click Event Function**: The name of a JavaScript function that runs when the user clicks on an item series in the chart.
	
	The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Donut Charts	Charts that plot one or more series of item values as segments of one or more concentric donuts, where each donut plots items for a single series. The segments for each item series therefore occupy one of the concentric donuts. The size of

Component	Properties and notes

	each segment in a given donut occupies space that is sized relative to the item value it represents in that series.
	
	• **Data Source Name**: A chart data source instance that you have defined for the Blank view.
	
	• **Title**:
	
	   • **Text**: A string that specifies a title for the chart.
	
	• **Legend**: Identifies each donut segment by a key that shows the value of the Item Category Field selected for the plotted item series (see the **Chart series** property) and a corresponding segment color:
	
	   • **Background**: A color selection for the legend background of the chart. Choose a color that does not mask a generated legend color.
	
	   • **Position**: The location in the chart to display the legend, which you can select from **Top**, **Bottom**, **Left**, or **Right**.
	
	   • **Visible**: Indicates if the legend is displayed for the chart.
	
	Note: The user can identify a corresponding segment in the chart by clicking its legend key, which alternately removes and re-inserts the plotted segment in the chart.
	
	• **Series default settings**: Default settings for displaying series item values in the chart:
	
	   • **Labels setting**: Presentation of every series item label:
	
	      • **Background**: A background color selection for item value labels. Choose a color that is not masked by a generated segment color for an item category.
	
	      • **Position**: The location in the chart to display the item labels, which you can select from **above**, **below**, **center**, **insideBase**, **left**, **outsideEnd**, **right**, **top**, or **bottom**. Choose the value that best distinguishes item labels both within and among different item series (donuts).
	
	      • **Template Function**: Item labels require custom code to display the item value. See the code example below this table. Any function you name in this field must be defined in *application-folder*\app\src\modules*module-folder**view-folder* \controller.public.js.
	
	      • **Template Id**: The Id of an HTML <div> template for item labels. If specified, the template must be located in the *application-folder*\app\src\modules*module-folder**view-folder* \topSection.html file.
	
	      • **Visible**: Indicates if an item label is displayed for every donut segment.

Component	Properties and notes

	• **Chart series**: Click **Edit** to add or change each series item to plot, including the properties of an item. See Editing chart series on page 83 for more information.
	
	• **Tooltip settings**: Settings for displaying the item value when the pointer hovers over its donut segment:
	
	  • **Format**: This can be used to customize how the tooltip for each item value appears over its donut segment. Use the argument {0} to represent the item value. You can also add additional text to be displayed. For example, if the items are numeric values measured in kilograms, `{0} kg` appends " kg" to every item value. You can also add specifiers to control the format of the increment values. If the increments are date values, you can specify that they display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:
	
	    `{0:D}`
	
	  See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.
	
	  • **Template Function**: Tooltips require custom code to display the item value. See the code example below this table. Any function you name in this field must be defined in *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js.
	
	  • **Template Id**: The Id of an HTML <div> template for item tooltips. If specified, the template must be located in the *application-folder*\app\src\modules*module-folder**view-folder*\topSection.html file.
	
	  • **Visible**: Indicates if each item value is displayed in the chart as a tooltip.
	
	• **Model**: The chart component can be bound directly to a data source or associated with a data source through a model. For the latter, select the model instance with this menu.
	
	• **Events**:
	
	  • **Data Bound Event Function**: The name of a JavaScript function that runs when the chart is bound to its data source.
	
	  • **Click Event Function**: The name of a JavaScript function that runs when the user clicks on an item series in the chart.
	
	The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Component	Properties and notes

Line Charts	Charts that plot one or more series of item values as a line plotted along the item points for each series. The line for each item series gets a distinguishing color.
	
	• **Data Source Name**: A chart data source instance that you have defined for the Blank view.
	
	• **Title**:
	
	  • **Text**: A string that specifies a title for the chart.
	
	• **Legend**: Identifies each item series plotted in the chart by a key showing the value of its Item Name field (see the **Chart series** property) and a corresponding line color:
	
	  • **Background**: A color selection for the legend background of the chart. Choose a color that does not mask a generated legend color.
	
	  • **Position**: The location in the chart to display the legend, which you can select from **Top**, **Bottom**, **Left**, or **Right**.
	
	  • **Visible**: Indicates if the legend is displayed for the chart.
	
	Note: The user can identify a corresponding series in the chart by clicking its legend key, which alternately removes and re-inserts its plotted line in the chart.
	
	• **Value axis configuration**: Configures the axis for plotting series item values in the chart:
	
	  • **Labels**: Presentation of the major item increment values along this axis:
	
	    • **Format**: This can be used to customize how the increment values appear along the axis. Use the argument {0} to represent the increment value. You can also add additional text to be displayed. For example, if the axis is used to plot numeric values measured in kilograms, `{0} kg` appends " kg" to every major increment. You can also add specifiers to control the format of the increment values. If the increments are date values, you can specify that they display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:
	
	`{0:D}`
	
	See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.
	
	    • **Rotation of the angle**:
	
	      • **Angle**: The angle of rotation for displaying the increment values.
	
	    • **Line**:
	
	      • **Visibility of line**: Indicates if the axis line is displayed.

Component	Properties and notes

	• **Major unit**: The size of the major increments labeled along the axis.
	
	• **Max axis**: The maximum item value plotted on the axis.
	
	• **Min axis**: The minimum item value plotted on the axis.
	
	• **Category axis**: Configures the axis used to identify a category for each plotted item:
	
	  • **Field**: The name of a data source field that identifies the category for the item value. For example, a Country field in a Customer data source that identifies a national category for Balance item values.
	
	  • **Labels**: Presentation of the category values along this axis:
	
	    • **Format**: This can be used to customize how the categories appear along the axis. Use the argument {0} to represent the category Field value. You can also add additional text to be displayed. For example, if the axis is used to plot numeric values that identify an age in years, `{0} years` appends " years" to every Field value. You can also add specifiers to control the format of the Field values. If the Field represents date values, you can specify that they display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:
	
	    `{0:D}`
	
	    See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.
	
	    • **Rotation of the angle**:
	
	      • **Angle**: The angle of rotation for displaying the category values.
	
	• **Series default settings**: Default settings for displaying series item values in the chart:
	
	  • **Labels setting**: Presentation of every series item label:
	
	    • **Visible**: Indicates if a label is displayed for the plot point of each item value.
	
	• **Chart series**: Click **Edit** to add or change each series item to plot, including the properties of an item. See Editing chart series on page 83 for more information.
	
	• **Tooltip settings**: Settings for displaying the item value when the pointer hovers over a plot point:
	
	  • **Format**: This can be used to customize how the tooltip for each item value appears over its plot point. Use the argument {0} to represent the item value. You can also add additional text to be displayed. For example, if the items are numeric values measured in kilograms, `{0} kg` appends " kg" to every

Component	Properties and notes

	item value. You can also add specifiers to control the format of the increment values. If the increments are date values, you can specify that they display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:
	
	`{0:D}`
	
	See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.
	
	• **Template Function**: The tooltip requires custom code to display the item value. See the code example below this table. Any function you name in this field must be defined in *application-folder*`\app\src\modules\`*module-folder*`\`*view-folder*`\controller.public.js`.
	
	• **Template Id**: The Id of an HTML <div> template. If specified, the template must be located in the *application-folder*`\app\src\modules\`*module-folder*`\`*view-folder*`\topSection.html` file.
	
	• **Visible**: Indicates if each item value is displayed in the chart as a tooltip.
	
	• **Model**: The chart component can be bound directly to a data source or associated with a data source through a model. For the latter, select the model instance with this menu.
	
	• **Events**:
	
	  • **Data Bound Event Function**: The name of a JavaScript function that runs when the chart is bound to its data source.
	
	  • **Click Event Function**: The name of a JavaScript function that runs when the user clicks on an item series in the chart.
	
	The code for these event functions must be included in the file, *application-folder*`\app\src\modules\`*module-folder*`\`*view-folder*`\controller.public.js`. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Pie Charts	Charts that plot item values from one or more series as slices of a single pie. The size of each slice is relative to the item value it represents in its series. **Note:** The space constraints of a single pie make this chart most appropriate for plotting a single item series at one time. • **Data Source Name**: A chart data source instance that you have defined for the Blank view. • **Title**: • **Text**: A string that specifies a title for the chart.

Component	Properties and notes

	• **Legend**: Identifies each slice by a key that shows the value of the Item Category Field selected for each plotted item series (see the **Chart series** property) and a corresponding color:
	
	  • **Background**: A color selection for the legend background of the chart. Choose a color that does not mask a generated legend color.
	
	  • **Position**: The location in the chart to display the legend, which you can select from **Top**, **Bottom**, **Left**, or **Right**.
	
	  • **Visible**: Indicates if the legend is displayed for the chart.
	
	Note: The user can identify a corresponding slice in the chart by clicking its legend key, which alternately removes and re-inserts the plotted slice in the chart.
	
	• **Series default settings**: Default settings for displaying series item values in the chart:
	
	  • **Labels setting**: Presentation of every series item label:
	
	    • **Background**: A background color selection for item value labels. Choose a color that is not masked by a generated slice color for an item category.
	
	    • **Template Function**: Item labels require custom code to display the item value. See the code example below this table. Any function you name in this field must be defined in *application-folder*\app\src\modules*module-folder**view-folder* \controller.public.js.
	
	    • **Template Id**: The Id of an HTML <div> template for item labels. If specified, the template must be located in the *application-folder*\app\src\modules*module-folder**view-folder* \topSection.html file.
	
	    • **Visible**: Indicates if an item label is displayed for every slice.
	
	• **Chart series**: Click **Edit** to add or change each series item to plot, including the properties of an item. See Editing chart series on page 83 for more information.
	
	• **Tooltip settings**: Settings for displaying the item value when the pointer hovers over its pie slice:
	
	  • **Format**: This can be used to customize how the tooltip for each item value appears over its pie slice. Use the argument {0} to represent the item value. You can also add additional text to be displayed. For example, if the items are numeric values measured in kilograms, {0} kg appends " kg" to every item value. You can also add specifiers to control the format of the increment values. If the increments are date values, you can specify that they display dates in long form, e.g., Wednesday, October 3, 1997, using this syntax:
	
	    {0:D}

Component	Properties and notes

	See http://docs.telerik.com/kendo-ui/framework/globalization/numberformatting and http://docs.telerik.com/kendo-ui/framework/globalization/dateformatting for more information on number and date formatting options, respectively.
	
	• **Template Function**: Tooltips require custom code to display the item value. See the code example below this table. Any function you name in this field must be defined in *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js.
	
	• **Template Id**: The Id of an HTML <div> template for item tooltips. If specified, the template must be located in the *application-folder*\app\src\modules*module-folder**view-folder*\topSection.html file.
	
	• **Visible**: Indicates if each item value is displayed in the chart as a tooltip.
	
	• **Model**: The chart component can be bound directly to a data source or associated with a data source through a model. For the latter, select the model instance with this menu.
	
	• **Events**:
	
	  • **Data Bound Event Function**: The name of a JavaScript function that runs when the chart is bound to its data source.
	
	  • **Click Event Function**: The name of a JavaScript function that runs when the user clicks on an item series in the chart.
	
	The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Sample template function for a chart item value tooltip or label:

```
templateFunction: function(data) {
    var template = kendo.template("<div id='box'>#: Balance #</div>");
    var result = template(data); //Passing the data to the template
    return result;}
```

Table 6: Scheduling components

Component	Properties and notes
Calendar	A graphical calendar that supports navigation and selection. • **Min** and **Max Date** (optional): The minimum (earliest) and maximum (latest) dates that can be displayed. • **Default Date** (optional): The date initially displayed.

Component	Properties and notes
	• **Date Format**: Specifies the date format used by the calendar component's underlying `value( )` function. See http://docs.telerik.com/kendo-ui/api/javascript/ui/calendar#configuration-value for more information. • **Model**: The data model associated with this component. • **Events**: • **Change Event Function**: The name of a JavaScript function that runs when the user makes a change to the component value. The code for these event functions must be included in the file, *application-folder*`\app\src\modules\`*module-folder*`\`*view-folder*`\controller.public.js`. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Table 7: Custom components

Component	Properties and notes
Custom HTML	A custom component that can contain valid HTML. • **HTML**: Text with desired HTML tags applied.

Editing chart series

From the **Chart Series** dialog, you can define a value (item) to plot for a given series:

- **Items** :

  - **+ Add Item**: Adds an item for a series. Selecting an **item** displays Properties that you can set for the **item**, depending on the chart type (Area, Bar, Donut, Line, or Pie).

- **Properties**:

  - **Item Field** (All Charts): A data source field whose value the chart will plot. For example, a Balance field in a Customer data source.

  - **Line** (Area Charts):

    - **Style** (Area and Line Charts): The style of the line used to plot this item series:

      - **step**: Plotted with a squared line between item points, like a square wave.

      - **smooth**: Plotted with a smooth curve between item points, like a sine wave.

      - **normal**: Plotted with a straight line between item points (the default).

  - **Item Name** (Area, Bar, and Line Charts): A string that you specify to identify the legend for the selected item value.

  - **Item Category Field** (Donut and Pie Charts): The name of a data source field that identifies a category for the value identified by Item Field. For example, a Country field in a Customer data source that identifies a national category for Balance values.

    Note: For Area, Bar, and Line Charts, you specify this field as part of the Category axis configuration.

  - **Start Angle** (Donut and Pie Charts): Affects the relative position of item plots around the Donut or Pie Chart.

Adding navigation

You can add standalone buttons, expanders containing any components, tab strips with tabs containing any components, or toolbars containing buttons, split buttons, button groups, separators, and templates.

Table 8: Navigation components

Component	Properties and notes
Button	A styled, clickable UI with keyboard operability for elements. - **Id**: An ID that must be specified for the button. - **Text**: The text that appears on the button itself. - **Primary**: Indicates whether the button will be highlighted with a background color. - **Events**: - **Click Event Function**: The name of a JavaScript function that runs when the user clicks this button.

Component	Properties and notes
	The code for this event function must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.
Expander	A collapsible container that when clicked, alternately expands to display, and collapses to hide its contents. To add contents, you drag at least one layout row to the expander, then drag additional components into that row, or into other rows that you have added.
	Note: You can construct a tree view by adding expanders to an existing expander row.
	• **Id**: An ID that must be specified for the expander. • **Expanded**: Indicates whether the expander is initially displayed in an expanded state. • **Text**: The text that appears on the expander itself.

Tab Strip	A container for one or more tabs that when clicked, each opens its own container to display contents for access.
	
	To add tabs to a tab strip, see the **TabStrip Items** property described below. To add contents to an existing tab:
	
	1. Select the tab from the tab strip by its label text in the drop-down menu beside its trash icon. The selected tab opens.
	
	2. Drag at least one layout row to the tab, then drag additional components into that row, or into other rows that you have added to the tab.
	
	• **Id**: An ID that must be specified for the tab strip.
	
	• **TabStrip Items**: Click **Edit** to add or remove tabs in the tab strip. See Editing tab strip items on page 86 for more information.
	
	• **Tabs Position**: Specifies where around the tab strip the tabs appear. You can select either **top**, **left**, **right**, or **bottom**. The order of tabs is always left-to-right (for **top** or **bottom**) or top-to-bottom (for **left** or **right**).
	
	• **Events**:
	
	  • **Select Event Function**: The name of a JavaScript function that runs when the user selects a tab in the tab strip.
	
	  • **Show Event Function**: The name of a JavaScript function that runs when the selected tab displays its contents.

Component	Properties and notes
	The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder* \controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Toolbar	Provides different types of items, such as buttons, toggle buttons, split buttons, button groups, and other customized elements.
	
	• **Id**: An ID that must be specified for the toolbar.
	
	• **Toolbar Items**: Click **Edit** to add or remove items on the toolbar. See Editing toolbar items on page 86 for more information.
	
	• **Resizable**: If the size of the browser window changes in a way that hides one or more tools, this option allows the toolbar to move the overflow into a drop down list.
	
	• **Events**:
	
	    • **Click Event Function**: The name of a JavaScript function that runs when the user clicks a button in the toolbar.
	
	    • **Toggle Event Function**: The name of a JavaScript function that runs when the user toggles a button in the toolbar.
	
	The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder* \controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

Editing tab strip items

From the **TabStrip Items** dialog, you can define tabs for the tab strip:

• **Items** :

    • **+ Add Item**: Adds an item (tab) to this tab strip.

• **Properties**:

    • **Text**: Label text that appears on the tab for the selected item.

Editing toolbar items

From the **Toolbar Items** dialog, you can add or remove buttons, separators, and templates with the following properties:

• **Items**: An item list with its own toolbar above it that allows you to click icons that have the following tooltips:

    • **Add Button to Toolbar**: Adds a button item to the list.

    • **Add Split Button to Toolbar**: Adds a split button item to the list.

- **Add Button Group to Toolbar**: Adds a button group item to the list.

- **Add Button to the selected ...**: Adds a button item to the currently selected split button or button group item.

- **Add Template to Toolbar**: Adds a custom template item to the list.

- **Add Separator to Toolbar**: Adds a separator to the current end of the list.

- **Properties**: After adding items, when you select a:

  - **Split button** item in the list, you see the following properties:

    - **Id**: An ID you can specify for the split button.

    - **Text**: A label text that you specify for the split button.

  - **Button group** item in the list, you see the following properties:

    - **Id**: An ID you can specify for the button group.

  - **Button** item in the list, or in a split button or button group, you see the following properties:

    - **Id**: An ID you can specify for the button.

    - **Text**: A label text that you can specify for the button.

    - **Primary**: Indicates whether the button will be highlighted with a background color.

    - **Togglable**: Indicates whether the button can be toggled on and off.

    - **Group** (in button groups only): A name you can specify for a group with which this button is associated.

    - **Events**:

      - **Click Event Function**: The name of a JavaScript function that runs when the user clicks this button.

      - **Toggle Event Function**: The name of a JavaScript function that runs when the user toggles this button when **Togglable** is checked.

      The code for these event functions must be included in the file, *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

  - **Template** item in the list, you see the following properties:

    - **Id**: An ID you can specify for the button group.

    - **Template ID**: The Id of an HTML custom template that implements this item in the list. If specified, the template must be located in the *application-folder*\app\src\modules*module-folder**view-folder*\topSection.html file.

    - **Template Function**: A function used to implement the template for this item in the list. Any function you name in this field must be defined in *application-folder*\app\src\modules*module-folder**view-folder*\controller.public.js.

  - **Separator** in the list, no properties are available for a separator.