



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

Notices

© 2016 Progress Software Corporation and/or one of its subsidiaries or affiliates. All rights reserved.

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

Business Making Progress, Corticon, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, Deliver More Than Expected, Icenium, Kendo UI, Making Software Work Together, NativeScript, OpenEdge, Powered by Progress, Progress, Progress Software Developers Network, Rollbase, RulesCloud, RulesWorld, SequeLink, Sitefinity (and Design), SpeedScript, Stylus Studio, TeamPulse, Telerik, Telerik (and Design), Test Studio, and WebSpeed are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries. AccelEvent, Analytics360, AppsAlive, AppServer, Arcade, BravePoint, BusinessEdge, DataDirect Spy, DataDirect SupportLink, DevCraft, DigitalFactory, Fiddler, Future Proof, High Performance Integration, JustCode, JustDecompile, JustMock, JustTrace, OpenAccess, ProDataSet, Progress Arcade, Progress Profiles, Progress Results, Progress RFID, Progress Software, ProVision, PSE Pro, SectorAlliance, Sitefinity, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, WebClient, Who Makes Progress, and Xervo are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks contained herein may be trademarks of their respective owners.

Please refer to the Release Notes applicable to the particular Progress product release for any third-party acknowledgements required to be provided in the documentation associated with the Progress product.



September 2016

Last updated with new content: Version 1.0

Table of Contents

Chapter 1: Introduction to the Kendo UI Designer7

Chapter 2: Creating an app9

- Creating an app: first step.....10
- Main editing screen.....10
- Adding data providers.....11
- Adding data sources.....11
- Adding and editing modules.....12
- Adding and editing views.....13
 - Login view.....14
 - Landing page view.....15
 - Data-Grid view.....15
 - Data-Grid-Form view.....17
 - Data-Grid-Separate-Form view.....19

Introduction to the Kendo UI Designer

The Kendo UI Designer is part of 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 pre-built, data-driven views. 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're presented with a list of web apps you've previously created, as well as options to:

- display existing apps in a card or list view
- search apps by keyword
- sort apps alphabetically
- edit or delete an existing app (by clicking the gear icon in the card or list item)
- create a new app

You can access this home screen at any time by clicking the Progress logo in the upper-left corner of the screen.

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 module is provided for you at creation: it contains a customizable log in screen and a landing page.
- *Views*: The screens that make up each module. Views are data-driven, composed of predefined templates that the Designer populates with your data sources.

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](#)
- [Main editing screen](#)
- [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 home screen, 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. **Location:** The location for the files associated with your app. This should match your workspace and project folder of the associated Web UI project in Progress® Developer's Studio for OpenEdge® (PDSOE).
3. **Description:** An optional description of your app. This text will appear next to your app on the home screen.

When finished, click **Create App**, which will take you to the main editing screen.

Note: You can also start editing a pre-existing app by simply clicking on the card or row representing the app, depending on the view. The gear menu on this screen allows you to **Delete**. The **Edit** option does NOT provide access to the full set of app editing tools, but rather allows you to add a brief description and add a logo to the app. You access the same edit dialog you see when clicking on the pencil icon next to the app name in the main editing screen. See [Main editing screen](#) on page 10 for more information.

See also

[Main editing screen](#) on page 10

Main editing screen

Once you've created a new app (or selected **Edit** from an existing app's drop-down menu), you'll enter the main editing screen. From this screen, you can:

- Click on the pencil icon next to the app name to:
 - Add an optional description of the app for your own reference (not visible to the app user).
 - Add a logo that will appear in the header of the main page of the app. The name of the logo you enter must be the name of an image file located in `application-folder/src/assets/images`.
- **Preview:** Show a preview of what the finished app looks like and how it behaves. Preview runs the latest build of the web app. You will be given the option to to select **Build & Preview** if you want to build before running the web app.
- **Build:** Generates the HTML and JavaScript source code for the app.
- Add and edit data providers.
- Add and edit modules.
- Edit the Application module, which contains views for a log in screen and a landing page.

See also

[Adding data providers](#) on page 11

Adding data providers

To add a data provider:

1. Click **Add Data Provider** on the main editing screen.
2. Enter a **Name** for the data provider. The name cannot contain spaces. Only letters, numbers, dashes, and underscores are permitted.
3. Enter the **Service URI**. This is the location of the object data service where calls are sent during runtime.
4. Enter the **Catalog URI**. This is the location of the catalog file generated when the Progress Data Object Service is built on the server. See the help system in Progress Developer's Studio for OpenEdge (PDSOE) for more information.

Note: Once entered and saved, the name, service URI, and catalog URI of a data provider cannot be further edited.

5. Choose the security model for the service URI from the **Authentication Model** menu. This should match the Authentication of the server for the Data Object Service.
6. (Optional) You have the option to manually create data sources once you've finished creating your data provider, but 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. A data source is created using the schema definition in the catalog, and when auto-create 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.
 - 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
 - A ProDataSet with more than one top-level table: A data source for each top-level table is created with the following naming convention: resource name of the ProDataSet + "." + table name.
7. (Optional) Select **Create Data Sources for child tables** to allow the Designer to automatically generate data sources for child tables for a ProDataSet with related tables. These are named resource name + "." + table name. You must select **Auto-create Data Sources** to enable this option.

With one or more data providers created, you can now create data sources using those providers.

See also

[Adding data sources](#) on page 11

Adding data sources

To add a data source from an existing data provider:

1. Click **Edit** next to the given data provider in the main editing screen. 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 single fields back and forth between the lists. The order that the fields appear in the **Included Fields** list is the same order that will be used for the columns and/or form fields, so drag and drop the included fields into the order you prefer.
6. When you select any particular field, you can specify the **Label** and **Editor Type**. 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 picker. For more information on semantic types, see *Kendo UI Builder by Progress: Modernizing OpenEdge Applications*.

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 record identifier fields.

7. (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 filtering and counting the data. Business Entities provide the JSON Filter Pattern (JFP) for the READ method and code for a count method. (You are prompted to enter the name of the count method if **Client-side processing** is unchecked.) See "Updating Business Entities for access by Telerik DataSources and Rollbase external objects" in *OpenEdge Development: Web Services* for an example implementation of the pattern and count method.

Note that if you create one or more views using your data source and then come back to change some of the details of that data source, e.g., labels, editor types, those changes will NOT be reflected in the views you've already created. The changes will be in effect for any new views you create, but those changes will not be propagated back to views already created.

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

See also

[Adding and editing modules](#) on page 12

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 main editing screen, 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 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 main editing screen.

Once you've created the module, it will be added to the list of modules on the main editing page. Click **Edit** to enter the module editing screen. Once there, you can:

- add and edit views
- save a view in progress
- revert a view to its saved state
- preview your application
- build your application (i.e., generate the HTML and JavaScript source code)

The next step in creating your module is to add one or more views.

See also

[Main editing screen](#) on page 10

[Adding and editing views](#) on page 13

Adding and editing views

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

- the built-in Application module contains two views, a login page and a landing page
- the modules you create can contain views with combinations of data grids and forms that you customize

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

1. From the module editing screen, 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: The name you enter here is what will appear in the navigation panel in the running app. The name you enter while actually editing the view (e.g., **Grid Name**, **Data Form name**, etc.) is for your internal use only.

3. Choose the type of view. What follows is a basic summary, although each type has different properties and behaviors that can be customized, as described in later topics:

- **Data-Grid:** A read-only, 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 display showing the information in a selected row. This view can be read-only or user editable. If editing is enabled, users edit information in the form display.
 - **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 display on a separate screen. If editing is enabled, the screen switches to a form display suitable for editing.
4. (Optional) 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 indicate three custom functions to handle events:
- **Init Event Function:** fires the first time the view renders.
 - **Show Event Function:** fires after the view is rendered.
 - **Hide Event Function:** fires when the view is replaced in the layout.

A Row Select Event Function is also supported, and you can specify that once your view is created. Any function you name in these fields must be defined in `application-folder/src/scripts/ModuleName-ViewName/view-factory.js`. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

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

Login view

To edit the log in view:

1. From the main editing screen, click **Edit** next to the **Application** module, at which point the login view will automatically open. If you're already in the module editing screen, select **login** from the list of views.
2. You can edit the following details of the log in screen:
 - **Custom HTML sections:** You can specify custom HTML code for the top, middle, and bottom of the login window. The Designer automatically generates three files for you in the folder `application-folder/src/html/Application-login`. Add your HTML code to the three files as desired.
 - **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/src/assets/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.
 - **Login Event Function:** The name of a function that runs when the user clicks the **Login** button. Any function you name in this field must be defined in `application-folder/src/scripts/Application-login/view-factory.js`.

Landing page view

To edit the landing page view:

1. From the main editing screen, click **Edit** next to the **Application** module, at which point the login view will automatically open. If you're already in the module editing screen, select **landing-page** from the list of views.
2. **Custom HTML sections:** You can edit the top and bottom HTML sections for the landing page, which appear above and below icons representing the application's modules. The Designer automatically generates two files for you in `application-folder/src/html/Application-landing-page`. Add your HTML code to the files as desired.

Data-Grid view

Selecting the Data-Grid view arranges the included fields from your data source by column, in the same order you arranged them in the **Included Fields** list in the data source. The Data-Grid has a number of customizable properties, listed on the right of the screen:

- **Grid Name:** A name for internal use only. The name that will appear for navigation on the main page of the app is the name you chose when first creating the view.
- **Grid Title:** The title that will appear when this view is actually on the screen in the app.
- **Data Provider:** The different data provider for this grid.
- **Data Source:** The desired data source from the chosen data provider.
- **Grid Columns:** Click **Edit** to change the included columns and various properties of the columns. See [Editing grid columns](#) on page 16 for more information.
- **Page Size:** The number of rows that will be displayed per page in the grid.
- **Row Template Function Name:** A JavaScript function name that returns the text for a Kendo UI row template. If specified, the row template is used to format all the rows in the grid. 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*.
- **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.
- **Custom Sections:** You can add your own custom HTML code for three sections in the grid view:
 - **Top Section:** The area above the Grid Title.

- **Middle Section:** The area below the Grid Title but above the grid itself.
- **Bottom Section:** The area below the grid.

The Designer automatically generates three files for you in the folder `application-folder/src/html/ModuleName-ViewName`. Add your custom HTML code to those files as desired.

- **Grid Events:** Use **Row Select Event Function** to indicate a JavaScript function that will run when a grid row is selected by the app user. The code should be included in the file `application-folder/src/html/ModuleName-ViewName/view-factory.js`. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

See also

[Adding data sources](#) on page 11

[Editing grid columns](#) on page 16

Editing grid columns

From this dialog, you can change several properties of the columns. For example:

- 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 single fields back and forth between the lists.
- When you select a particular field, you can:
 - Change the **Title Text**, which is the column heading users will see in the app.
 - **Enable HTML coding:** 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.
 - **Format:** This can be used for additional customization in how the field is presented. Use the argument `{0}` to represent the content of the column. For example, you can add additional text to be displayed. If a column holds a numeric value measured in kilograms, then:

```
{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 should 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.

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

Data-Grid-Form view

Selecting the Data-Grid-Form view creates a split screen. On the left, the form arranges the included fields from your data source by column, in the same order as they appear in the **Included Fields** list. On the right, single highlighted rows are presented in a form view. This view can be read-only or user editable.

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

- **Grid Name:** A name for internal use only. The name that will appear for navigation on the main page of the app is the name you chose when first creating the view.
- **Title:** The title that appears when this view is actually on the screen in 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 data provider for this grid.
- **Data Source:** The desired 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 16 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 18 for more information.
- **Page Size:** The number of rows that will be displayed per page in the grid.
- **Row Template Function Name:** A JavaScript function name that returns the text for a Kendo UI row template. If specified, the row template is used to format all the rows in the grid. 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*.
- **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 11.

- **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.
- **Custom Sections:** You can add your own custom HTML code for three sections in the grid view:
 - **Top Section:** The area above the Grid Title.
 - **Middle Section:** The area below the Grid Title but above the grid itself.
 - **Bottom Section:** The area below the grid.

The Designer automatically generates three files for you in the folder `application-folder/src/html/ModuleName-ViewName`. Add your custom HTML code to those files as desired.

- **Grid Events:** Use **Row Select Event Function** to indicate a JavaScript function that will run when a grid row is selected by the app user. The code should be included in the file `application-folder/src/html/ModuleName-ViewName/view-factory.js`. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

See also

[Editing grid columns](#) on page 16

[Editing form fields](#) on page 18

Editing form fields

From this dialog, you can change several properties of the form fields. For example:

- 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 single fields back and forth between the lists.
- When you select a particular field, you can:
 - Specify the **Label Text** for the field.
 - **Format:** This can be used for additional customization in how the field is presented. Use the argument `{0}` to represent the content of the column. For example, you can add additional text to be displayed. If a column holds a numeric value measured in kilograms, then:

```
{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 should 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.

Data-Grid-Separate-Form view

Selecting the Data-Grid-Separate-Form view works in much the same way the Data-Grid-Form view works, except with this view, the user sees only the grid until they select a row. At that point, the view changes to the form, which you can make read-only or user editable, as with the Data-Grid-Form view. The Data-Grid-Separate-Form has a number of customizable properties, listed on the right side of the screen:

- **Grid Name:** A name for your internal use only. The name that will appear for navigation on the main page of the app is the name you chose when first creating the view.
- **Title:** The title that appears when this view is actually on the screen in 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 data provider for this grid.
- **Data Source:** The desired 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 16 for more information.
- **Form Fields:** Click **Edit** to change various properties of the form fields. See [Editing form fields](#) on page 18 for more information.
- **Page Size:** The number of rows that will be displayed per page in the grid.
- **Row Template Function Name:** A JavaScript function name that returns the text for a Kendo UI row template. If specified, the row template is used to format all the rows in the grid. 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*.
- **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.
- **Custom Sections:** You can add your own custom HTML code for three sections in the grid view:
 - **Top Section:** The area above the Grid Title.
 - **Middle Section:** The area below the Grid Title but above the grid itself.
 - **Bottom Section:** The area below the grid.

The Designer automatically generates three files for you in the folder `application-folder/src/html/ModuleName-ViewName`. Add your custom HTML code to those files as desired.

- **Grid Events:** Use **Row Select Event Function** to indicate a JavaScript function that will run when a grid row is selected by the app user. The code should be included in the file `application-folder/src/html/ModuleName-ViewName/view-factory.js`. See *Kendo UI Builder by Progress: Modernizing OpenEdge Applications* for more information about extensions and event functions.

See also

[Editing grid columns](#) on page 16

[Editing form fields](#) on page 18