UX Studio: Tips and Best Practices

Last Modified on 03/09/2021 4:43 am EST

Edit an item template

You can customize your form's grid elements by editing their template.
For example, you can edit a grid's *Add new record* button so that it displays a different icon and label.

You can customize the following grid elements:

- Columns
- Delete/Edit/Add new record buttons
- Combo boxes
- Form: Select a form to replace the default grid.

Implement styles

Every control has a CSS sheet with properties, where you can apply styles to a control.

You can change the styles of your form. You can modify the markup or add a style attribute to the controls themselves. It is recommended that you create a separate style sheet and save it as a style file that you can reuse in your project.

Best practices

Follow these guidelines to improve performance and user experience.

- Create a simple hierarchy of controls.
- Display a limited number of controls on a view.
- Use AJAX and Update Panels to refresh the page wisely.
  You might be limited in how often to refresh the page, or which parts of the form to refresh. Therefore, use AJAX technology extensively, as it improves performance and user experience, and reduces the weight of the page.
  Here is an example that refreshes the page section smoothly:

  ```
  ----Controls----
  ```

- Choose the appropriate controls for user input data.
  For example, for integer data, use a numeric control. This way, you help the user input validation without the need to use a Validator control. For example, avoid using a text box with a validator for integer data.
- Disable View State for controls that do not collect data.
- Divide views into logical parts.
  For example, if you have 20 views, do not place them all in one activity, rather split them into several activities. This way enables efficient, simultaneous development, in which you can split work between multiple developers using Source Control.
- Limit the use of expressions in the markup, especially data binding expressions or expressions that invoke service queries. Multiple queries can harm performance, because the ASP.NET technology executes expressions several times.

- Consult the ASP.NET resources on the internet for additional best practices that can improve performance and user experience.