

Suite Is-25.4

Documentation Reference

File:

Developing-Pwa\Adding-@Mention-For-Fields\Adding-@Mention-For-Fields

The @mention component is a typeahead feature introduced for Progressive views. This component is supported for the Character field, where you can type in to provide your inputs. You can set the metadata for this component by using Developer Studio.

When you define a menu for a Character field, the Enable Menu Hotkeys property is automatically set to **False**. However, any hotkey definitions created earlier remain as they are. You can't add any duplicate hotkey definitions. Ensure that the menu you attach to a hotkey is a Search Menu.

Related topics

[Process-commands](#)

[Field-Properties](#)

[Developing-Progressive-Web-Applications](#)

[Using-Run-Process-and-PROCESS-commands](#)

Example

To understand how the @mention component works, consider a scenario where you're composing a new email in Microsoft Outlook for Office 365. As soon as you type @ you get a list of users in your organization and the user that you select gets added to the To list. You also get additional information about the user, such as name, email address, and a display picture if available, which helps you identify the correct user when there are multiple users of the same name.

As a scenario specific to IS, if an incident manager is working with an incident, the incident manager can type @ in the activity notes to get a list of agents. This helps the incident manager select a particular agent from the list of available agents.

To support the @mention component for Progressive views, the following properties are added in Developer Studio:

Property name	Property type	Property values	Description
Enable Menu Hotkeys	Boolean	<ul style="list-style-type: none">TrueFalse (Default)	Specifies if the menu hotkeys are to be enabled. To define menu hotkeys, you must set the value of this property to True . Use this property to define hotkey definitions. By default, the text "No Hotkey Defined" is displayed as the value for this property.
Menu Hotkeys	String	A string	

To configure the @mention component

1. On the Progressive view of a form, double-click a Character field.
2. On the **Properties** panel, under the Display property group, for the Enable Menu Hotkeys property, select the **True** value.
3. In the **Value** column for the Menu Hotkeys property, click the ellipsis button.
The **Menu Hotkeys** dialog box is displayed.

Menu Hotkeys

Hotkey	Menu Name	Active Link	Labels	Auto Complete Key Stroke #
@				
!				
#				
\$				
%				
^				
&				
*				
?				
+				
-				
_				

Add Delete

OK Cancel

Click **Add**, and then perform the following steps:

Column	Step
Hotkey	Click the first row in this column, and then select a character from the list.
Menu Name	Click the first row in this column, and then click the ellipsis button to select a menu from the Menu Selector dialog box.
Active Link	Click the first row in this column, and then click the ellipsis button to select an active link from the Active Link Selector dialog box.
	The active link that you attach to a hotkey must be an existing one and associated with the current form.

Labels

Click the first row in this column, and then in the box, type in labels that are separated by a semicolon.

Auto Complete Key Stroke #

Enter an integer value.

5. Click **OK**.

Parameters for hotkey definition

The following parameters can be a part of any hotkey definition:

Parameter	Description
Hotkey	When you press this character key, the @mention component is executed based on the other parameters that you've provided in the hotkey definition. This parameter is the mandatory part of any @mention definition.
Menu Name	This is a name of the menu that is used to populate the popup. This parameter is the mandatory part of any @mention definition.
Active Link	This is a name of the active link that gets executed when you select an item from the popup. This parameter is an optional part of any @mention definition.
Labels	This is a set of labels that are defined for a menu. This parameter is an optional part of any @mention definition. You can add any text or you can add \$menuLabel\$ where the label gets replaced by the menu label value during execution. <i>menuLabel</i> represents the label that is selected when using a search menu.
Auto Complete Key Stroke #	You can enter an integer value value for this parameter. For example, if you enter the value 3, a drop-down list appears after you type three characters in the menu. If you enter the value zero (0), the drop-down list appears as soon as type a hotkey character key.

There are scenarios when you want the users of your application to have the option of selecting the type of content that they see in an autosuggest list when using @mention. You can create such a search field by using the `CHANGE_HOTKEY_MENU` command.

To enable users to select the type of content displayed in @mention

In the following example scenario, as an application developer, you add the `CHANGE_HOTKEY_MENU` command to enable the application users to decide the type of content in the `@mention` list. However, note that real-world implementations can vary considerably depending on your business needs.

Example:

Problem

Apex Global wants to have a ticketing application in which one of the fields requires the application users to be able to use `@mention` to add mentions of either people or assets. The default behavior is that both assets and people are displayed in the `@mention` autosuggestion list. Apex Global wants the application users to be able to select an option to see only assets or only people for ease of use.

Solution

Seth, the application developer, creates two buttons right next to the field. Users can click one button to make `@mentions` suggest only assets or click the other button to make the `@mention` show only usernames. Seth uses a run process `CHANGE_HOTKEY_MENU` *fieldID key menuName* in an active link in each of the buttons so that the content displayed by the `@mention` autosuggest list is replaced by the content specified in the active link.

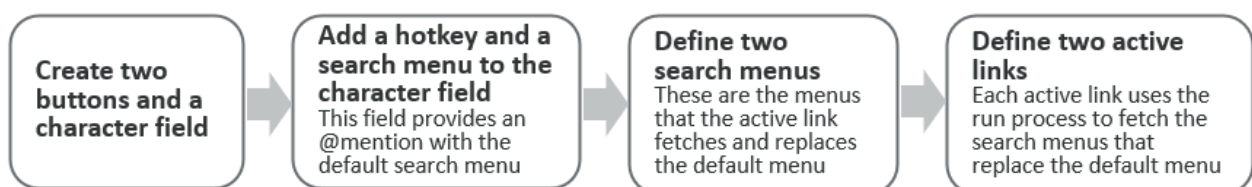
Before you begin

Make sure that you fulfil the following conditions before you perform the steps for creating fields where users can switch between `@mention` content.

- In Developer Studio, you must have a form that contains usernames and a form that contains asset names.
- You must have predefined search menus of type Search.

Process for creating fields where users can switch between `@mention` content

The following diagram shows a high-level overview of the steps required to create fields where users can switch between different types of `@mention` content:



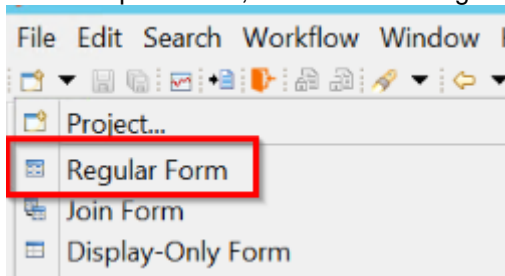
Create two buttons and a character field. The character field consists of a default `@mention` menu. The buttons are assigned active links and menus which should replace the default `@mention` menu when the buttons are clicked by application users.

The following animation explains a high-level concept of the use case:

To create the field and buttons

In this procedure, we create a character field with a default menu, two buttons that contain an active link that uses theCHANGE_HOTKEY_MENU.

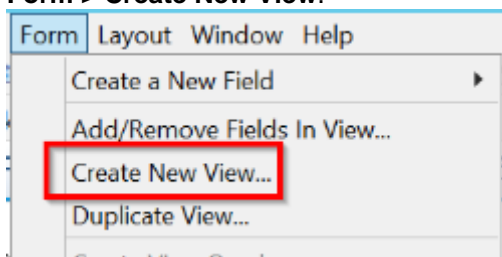
1. In Developer Studio, create a New Regular Form.



Create a Progressive View.

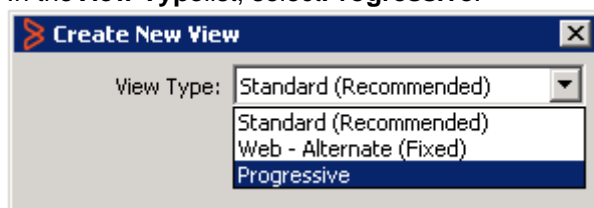
Steps (2)

1. **Form > Create New View.**



The Create New View dialog box opens.

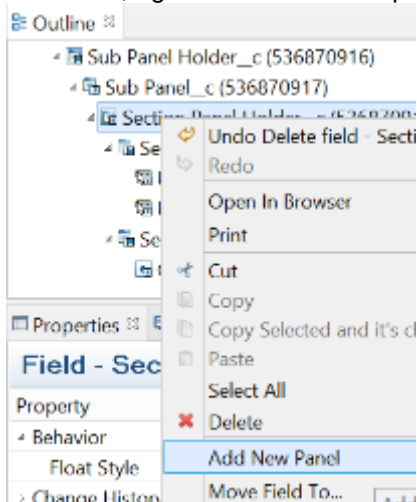
2. In the**View Type**list, select**Progressive**.



3. Add two buttons to the view.
4. Name the buttons as**Assets Only**and**Users Only**.
5. (*Optional*) Add a new section panel.

Steps (2)

1. In **Outline**, right-click the section panel holder and select **Add New Panel**.



2. In the Properties tab, select the Sub Panel Holder, and in Properties, click **Display > Orientation** and select **Horizontal**.
This changes the orientation to show the new section panel to the right of the section with the buttons.


To define the hotkey for the character field

1. Add a character field and add a hotkey associated with a search menu to it.

1. From the Palette, drag **Character** into the view or the section panel.
Developer Studio assigns it a default name such as **Character Field_c**.

2. Select the character field and rename it to **Character Field**.




3. In the Properties tab, click **Display > Enable Menu Hotkey** and select **True**.
This enabled you to set a hotkey for the character field.
4. Below the **Enable Menu Hotkey** property, click the ellipsis () button next to **Menu Hotkeys**.
The Menu Hotkeys dialog opens.

Click **Add**.

Developer Studio adds @ in the Hotkey column.

Best practice

To fetch email IDs in the @mention options, you must add # as the hotkey instead of @ in the Hotkey column to avoid errors.

6. In the Menu Name column, click the ellipsis () button.
The Menu Selector dialog opens.
7. Find the search menu that lists usernames and assets you want to associate with the hotkey and click **OK**.

8. In the Auto Complete Key Stroke # column, add **3**.

The autosuggest is activated when application user types at least three characters.

In the Properties tab, set a value in pixels for the **Hotkey Menu Max Height** property.

This is the maximum height of the @mention list when it displays a large number of results. The default value of the Hotkey Menu Max Height property is 200.

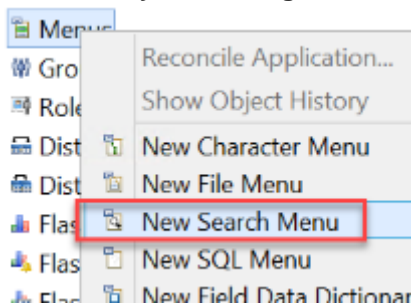
Best practice

We recommend a value between 200 to 500 pixels for the **Hotkey Menu Max Height** property.

2. Save the form with a name.
3. In this example name the form as **ChangeHotkeyMenu**.

To create a search menu that contains usernames

1. Create a new search menu.
2. Select **AR System Navigator > aRServerName > All Objects > Menus > New Search Menu**.



The hotkey supports only search menus.

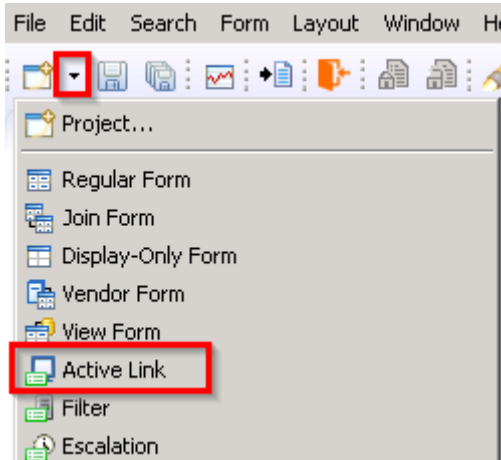
3. Click the ellipsis (**...**) button next to **Form Name**.
The Form Selector dialog opens.
4. Find and add the user form that contains all usernames, and click **OK**.
5. Save the search menu.
In this example, name the menu as **ChangeHotkeyMenuM1**.

To create a search menu that contains asset names

1. Create another search menu and name it as **ChangeHotkeyMenuM2**.
This menu must list all asset names.
2. Click the ellipsis (**...**) button next to **Form Name**.
The Form Selector dialog opens.
3. Find and add the form that contains all asset names, and click **OK**.
4. In the Label Fields section, click **Add**.
The Field Selector dialog opens.
5. Select Asset Name, and click **OK**.
6. Save the search menu.
In this example, name the menu as **ChangeHotkeyMenuM2**.

To create an active link for the Users Only button

1. Create an active link for the **Users Only** button named **ChangehotkeymenuAL1**.
2. Select **New > Active Link**.



Add the following settings to the active link:

Setting	Value, selection, or function	Additional details
Associated Forms > Add	ChangeHotkeyMenu	
Execution Options > Button/Menu Field	Users	This is the button that triggers the active link.
Run If Qualification	1=1	This triggers the run process in the If Actions without exception. Make sure that there is a single space between each attribute.
		<i>fieldID</i> is the ID of Character Field .
If Actions > Add action > Run Process	CHANGE_HOTKEY_MENU <i>fieldID</i> @changehotkeymenuM1\$NULL\$	changehotkeymenuM1 is not dynamic and must always be hardcoded (static). This active link runs when you type three characters, that is, when you type @ in the Character Field and it fetches information from the ChangehotkeymenuM1 search menu.

To create an active link for the Assets Only button

1. Create another active link for the **Assets Only** button named **ChangehotkeymenuAL2**.
2. Select **New > Active Link**.

Add the following settings to the active link:

Setting	Value, selection, or function	Additional details
---------	-------------------------------	--------------------

Associated Forms > Add

ChangeHotkeyMenu

Execution Options >
Button/Menu Field

Assets

Run If Qualification

1=1

This is the button that triggers the active link fetches and uses it to replace the default menu of the character field.

This triggers the run process in the **If Actions** without exception. Make sure that there is a single space between each attribute.

fieldID is the ID of **Character Field**.

If Actions > Add action > Run Process

CHANGE_HOTKEY_MENU*fieldID*@changehotkeymenuM2activeLink1labelsT

changehotkeymenuM2 is not dynamic and must always be hardcoded (static) with the menu name. This active link runs when you type three characters, that is, when you type @ in the **Character Field** and it fetches information from the **ChangehotkeymenuM2** search menu.

4. Save the changes.

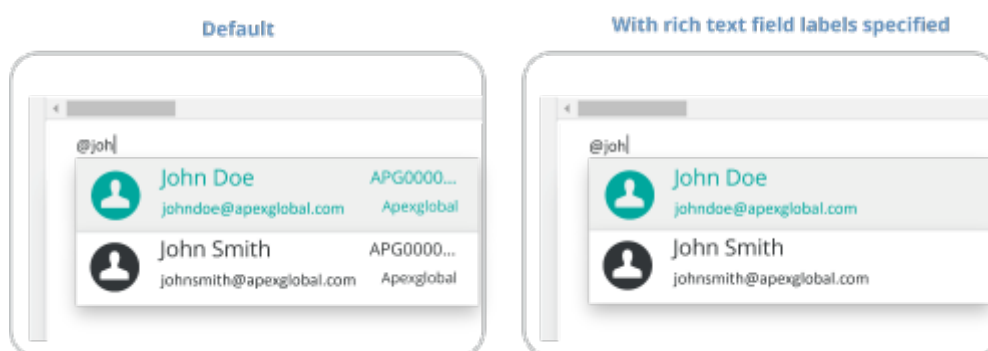
Important

In the case of rich text enabled fields, in the run process command, you can additionally specify the fields to be displayed in the @mention list as shown in the following example:

CHANGE_HOTKEY_MENU*fieldID*@changehotkeymenuM2activeLink1fieldID3 \$rtfLabel1\$ \$rtfLabel2\$

The **\$rtfLabel1\$** and **\$rtfLabel2\$** values are used to limit the display of @mention options list to just these two fields. If you do not specify any values, the first four fields available in the menu are displayed by default. You can specify up to four different values to be displayed in the @mention options list.

The following graphic illustrates the display of @mentions options with default information and when restricted to only employee name and email ID:



Validating the character field in PWA

View the form in the progressive web application screen to check whether it works as intended.

1. In a browser, log in to PWA.
For example, **<serverName>:8080/arsys/pwa/#/login**
2. Change the URL to include the Progressive View form that you created in Developer Studio.
<serverName>:8080/arsys/pwa/#/forms/<serverName>/<formName>/<progressiveViewName>
3. The PWA screen displays the character field and the buttons.
4. In **Character Field**, type @ followed by three characters.
The field displays users and assets.
5. Click **Users Only** and type @ followed by three characters.
The field displays usernames.
6. Click **Assets Only** and type @ followed by three characters.
The field displays assets names.

Where to go from here

Learn more about the syntax and additional options within the process command in the [Process-commands](#) topic.

File:

Developing-Pwa\Adding-Context-Menus-For-Right-Clicking-Of-

As a developer, you create various objects such as buttons and character fields in a Progressive Web Application (PWA). You might want to enhance the user experience in an application by providing the application users with the option to right-click an object and view a context menu. The context menu provides various options such as opening links in a new tab or a new window.

Related topic

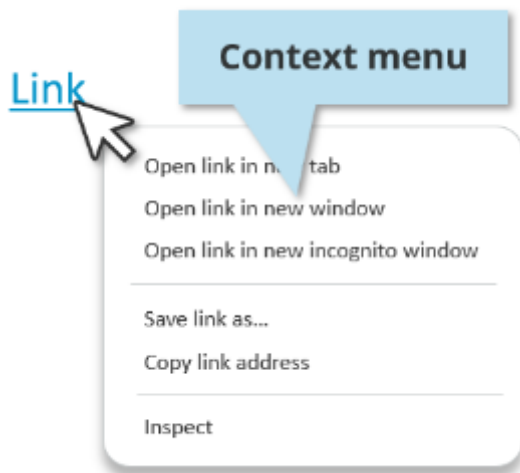
[Creating-a-popup-window-in-Progressive-Web-Applications](#)

[Adding-rich-text-fields-in-Progressive-Web-Applications](#)

Scenario

In Apex Global, Seth the developer wants to provide the application users with the ability to right-click links and view a context menu that displays options such as **Open link in new tab**, and **Open link in new window**. To enable context menus in the application, Seth adds a few properties by using Developer Studio.

Application users see a context menu as shown in the following example illustration:



You can enable application users to right-click and view a context menu for the following objects:

- Character fields
- Button field with Button Type set to Tertiary
- Table list column fields

Before you begin

- You must have created a character field, tertiary button, or a table column in Developer Studio.

You must set the following properties based on the object you create:

Object	Property	Value
Character field or character menu	Display Type	Edit
Display > Show URL	True	
Display > Display as text	True	
Button	Button Type	Tertiary
Table column	Display Type	Read Only

Important

These fields must display a non-null value.

To add context menus for right-clicking the interface objects in Progressive Web Applications

1. Open Developer Studio.
2. Search the form containing the object in which you want to create the link.
3. Open the form.
4. Select the object.
The object can be a character field, a tertiary button field, or a table list column field.
5. In the Properties pane, set **Display > HREF Link** to **True**.
6. Set **HREF** to a URL that you want users to open on the PWA screen.
Only static URLs are supported.

Set **HREF Target** to one of the following options:

- **Open in the full body of the window**—The destination page occupies the entire browser window.
- **Open in a new tab**—Opens the link in a new tab.
- **Open in the same frame**—Opens the link within the iframe view in which the link appears.
- **Open in the parent frame**—Opens the link in the parent iframe view.

8. Click **Save**.
9. Open the Mid Tier configuration tool and sync the cache.
`http://<MidtierHostName>:<PortNumber>/arsys/shared/config/config_cache.jsp`

To test the configuration

1. In a browser, log in to PWA.
(On-premises) **`http://<midtierHostName>:<portNumber>/arsys/pwa/#/login`**
(SaaS) **`http://<helixBaseURL>/arsys/pwa/#/login`**
2. Change the URL to include the Progressive View form that you created in Developer Studio.

(On-premises) **`http://<midtierHostName>:<portNumber>/arsys/pwa/#/forms/<AR_Server_Name>/<formName>/`**

(SaaS) **`http://<helixBaseURL>/arsys/pwa/#/forms/<AR_Server_Name>/<formName>/`**

4. Confirm that a context menu appears when you right-click the object.

You can also open the context menu by using standard keyboard shortcuts of the browser.

File:

Developing-Pwa\Adding-Icons-And-Images\Adding-Background

When developing an application, you can use an image as the background of a panel field, of the cells in a cell-based table field, or of a form view. You can use a reference to aAR System serverimage object, or you can store the image as an embedded object in the field.

To support the Image Reference functionality for panel and cell-based table fields, Remedy AR System clients and supporting applications such as theMid Tiermust be release 7.5.00 or later. When previous clients open a form containing an image reference, the image is converted to embedded format.

Related topics

[Adding-icons-and-images-to-Progressive-Web-Applications](#)

[Creating-and-modifying-image-objects](#)

[Adding-icons-to-UI-elements-on-forms-by-using-Developer-Studio](#)

[Previews-of-icons-available-in-Developer-Studio-to-support-PWA](#)

The following video (3:54) provides a summary of adding static or SVG images inDeveloper Studiofor progressive views:

■ Watch Video: <https://www.youtube.com/watch?v=349xdes86qM>

▶ <https://www.youtube.com/watch?v=349xdes86qM>

Using transparent images

Developer StudioandMid Tiersupport.**gif**and.**png**images, whichcanhave transparent backgrounds. You can use.**gif**and.**png**images anywhere that images are supported inAR System server, including

- Backgrounds on form views
- Backgrounds on standalone panel fields

PNG images that use*alpha transparency*are supported by Firefox and Safari. PNG and GIF images that use*binary transparency*are supported by allAR System servercompatible browsers.

To create graphics with transparent backgrounds that let the surface on which they are placed show through, use.**gif**and.**png**images with the Display as Flat Image property. For example, use such images to create borderless buttons. The following figure shows a.**gif**coffee cup image with a transparent background on a regular button and on a borderless button.

A .gif file with transparent background on regular and borderless buttons



To build toolbars, you can use transparent images on borderless buttons as shown in the following figure.

Transparent images as toolbar buttons



Tip

To add decorative graphics to forms, use borderless buttons without workflow.

Most browsers always use a visual aid, such as a dotted outline, to indicate that borderless buttons have focus. Internet Explorer, however, does this only when the form has certain background colors.

To add an image to a panel field or a cell-based table

1. Select the panel field or the working cell of the cell-based table field.
2. In the **Properties** tab, click the Background Image property, and click its **ellipsis** button.
3. In the Background Image dialog box:
 1. If necessary, click **Clear Image** to delete an existing image.
 2. Browse for the image to display on the button.
You can select **.bmp**, **.jpeg**, **.jpg**, **.gif**, and **.png** files. The selected image appears in the Preview area.
 3. To save the image to a different file or folder, click **Save to File**.
 4. Set the Image Type:
 - **Embedded Image**—The image is stored in the field display properties as an **ARByteList**. In this case, the image is embedded in the form and is therefore downloaded with the form whenever the form is refreshed by the client.
 - **Image Reference**—A reference to a shared image object is stored in the field display properties. In this case, the image is stored as an image object in AR System server. When the form is downloaded, the image is cached separately, so the image does not have to be refreshed along with the form. This allows for faster form refresh time.
 5. Click **OK**.
4. To position the graphic horizontally in the panel or cell, select the Background Image Horizontal property, and then select one of the following options:
 - **Left**—Position the left edge of the image at the left edge of the panel or cell.
 - **Center**—Position the horizontal center of the image at the horizontal center of the panel or cell.
 - **Fill**—Resize the image horizontally to fit the width of the panel or cell.
 - **Right**—Position the right edge of the image at the right edge of the panel or cell.
 - **Tile**—If the width of the image is smaller than the width of the panel or cell, tile the image horizontally in the panel or cell. If the width of the image is larger than the width of the panel or cell, Tile has no

effect.

This property works together with the Background Image Vertical property to control the overall position and dimensions of the image.

5. To position the graphic vertically in the pane or cell, select the Background Image Vertical property, and then select one of the following options:

- **Top**—Position the top edge of the image at the top edge of the panel or cell.
- **Center**—Position the vertical center of the image at the vertical center of the panel or cell.
- **Fill**—Resize the image vertically to fit the height of the panel or cell.
- **Bottom**—Position the bottom edge of the image at the bottom edge of the panel or cell.
- **Tile**—If the height of the image is smaller than the height of the panel or cell, tile the image vertically in the panel or cell. If the height of the image is larger than the height of the panel or cell, Tile has no effect.

This property works together with the Background Image Horizontal property to control the overall position and dimensions of the image. However, the Background Image Vertical property is not available in progressive views.

6. To add a background image for each device type, namely desktop, mobile, and tablet, click the ellipsis button for the Background Image property under Progressive-Desktop, Progressive-Mobile, and Progressive-Tablet respectively.

File:

Developing-Pwa\Adding-Icons-And-Images\Adding-Icons-And-I

As a developer, you can add icons and images to the progressive web application screens you create by using Developer Studio.

To add custom icons, you must use CSS. To learn more about using CSS, see [Rebranding Progressive Web Application screens](#).

Use the information in the following topics to create and modify image objects that you want to use in Progressive Web Applications (PWA):

Action

Reference

Use an image as the background of a panel field, of the cells in a cell-based table field, or of a form view.

[Adding-background-images-to-fields-and-form-views](#)

Create and manage image objects.

[Creating-and-modifying-image-objects](#)

Add icons to a UI element on a form in the Progressive Web Application.

[Adding-icons-to-UI-elements-on-forms-by-using-Developer-Studio](#)

Refer to the list of icons that are available in Developer Studio so that you can use them in a PWA

[Previews-of-icons-available-in-Developer-Studio-to-support-PWA](#)

Related topics

[Adding-images-to-buttons](#)

[Adding-images-to-tables](#)

File:

Developing-Pwa\Adding-Icons-And-Images\Adding-Icons-To-UI

Developer Studio provides some predefined icons that are available Out-Of-The-Box (OOTB) for a Progressive Web Application (PWA). You can add an icon to a UI element on a form in Developer Studio. To know what icons are available to support PWA, see [Previews-of-icons-available-in-Developer-Studio-to-support-PWA](#).



To add custom icons to the PWA screens, you must update the CSS mentioned in [Rebranding Progressive Web Application screens](#).

To add a icon to UI elements on forms by using Developer Studio

1. Copy an icon name from the [list of categorized icons](#).
2. In Developer Studio, paste the icon name as the value for a property that supports icons. You must paste the icon name as it is.
3. View the form in PWA to see the icon. To know how to enable a PWA, see [Enabling-Progressive-Views-for-forms-by-using-Centralized-Configuration-Settings](#). You can see that the icon is displayed on the form in PWA.

Example: Adding the d-icon-user_o icon

1. Copy the d-icon-user_o icon name from the [list of categorized icons](#).
2. In Developer Studio, drag the Character field onto the form.
3. Paste the icon name as the value for the Default Thumbnail Image property. The Default Thumbnail Image property is one of the properties where you can use an icon name.
4. View the form in PWA to see the icon. To know how to enable a PWA, see [Enabling-Progressive-Views-for-forms-by-using-Centralized-Configuration-Settings](#).

You can see the d-icon-user_o icon () on the form in PWA. If you use the d-icon-user_circle icon name, you can see the person icon in a circle .

Related topics

[Adding-icons-and-images-to-Progressive-Web-Applications](#)

[Previews-of-icons-available-in-Developer-Studio-to-support-PWA](#)

File:

Developing-Pwa\Adding-Icons-And-Images\Creating-And-Modifi

As a developer, you can use images as backgrounds for form views and certain field types, including buttons, panels, and cell-based tables when creating a Progressive Web Application (PWA). An image object is an image stored in the AR System server database with information defining the image as aAR System server object. The image object type appears in the AR System Navigator pane at the end of the All Objects list (see the following figure).

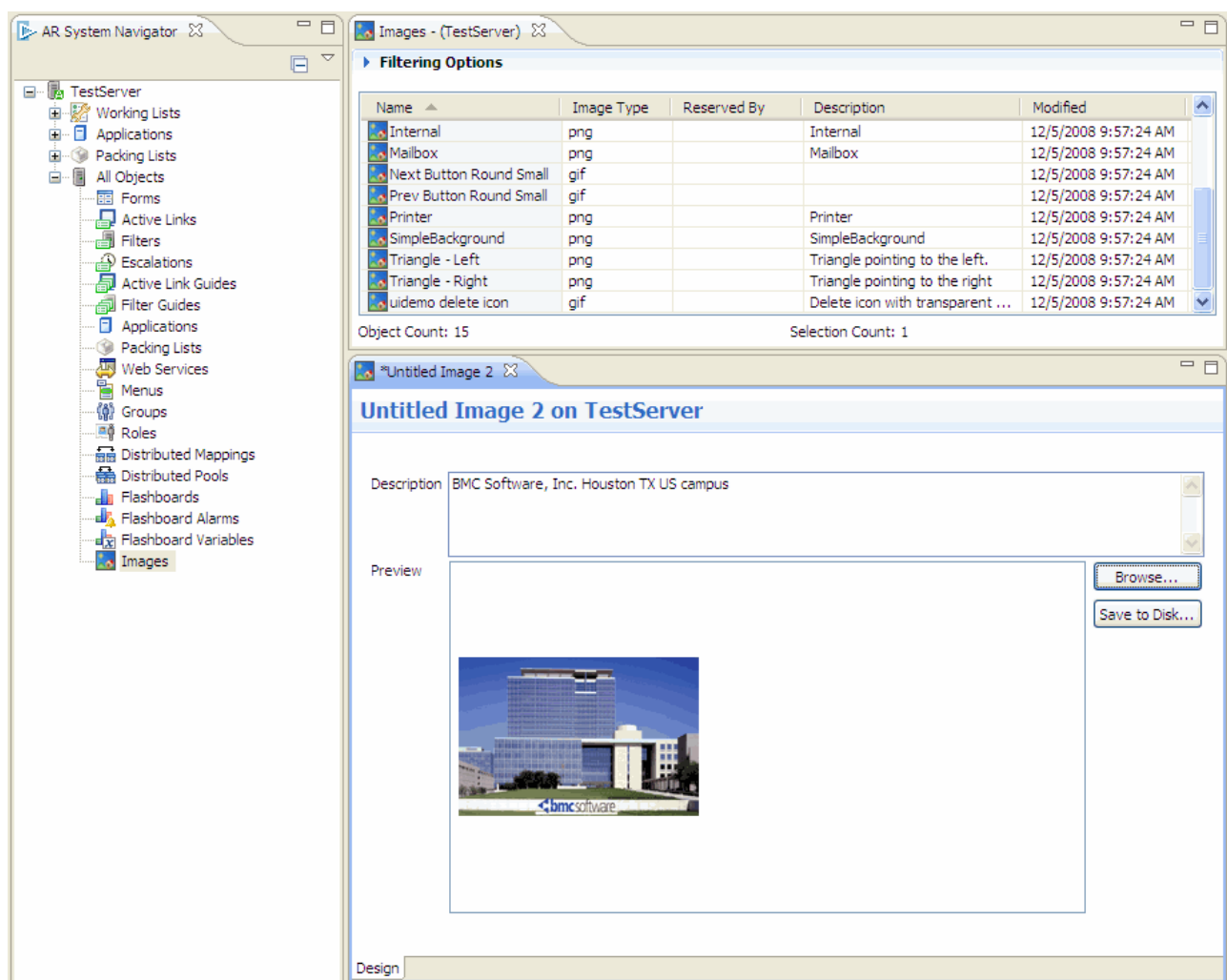
Related topics

[Adding-icons-and-images-to-Progressive-Web-Applications](#)

[Adding-background-images-to-fields-and-form-views](#)

[Adding-icons-to-UI-elements-on-forms-by-using-Developer-Studio](#)

[Previews-of-icons-available-in-Developer-Studio-to-support-PWA](#)



To use the same image in multiple locations, such as the background of related forms, you do not need to store a copy of the image in the display properties of each form view or field that uses the image. Instead,

store the image once as an image object, and include it by reference in form view and field display properties. Using an image object, you can also avoid the 4 MB size limit on the field display property.

AR System server supports the following image types:

- Windows bitmap (**.bmp** file extension)
- Joint Photographic Experts Group format (**.jpeg** or **.jpg** file extensions)
- Graphics Interchange Format (**.gif** file extension)
- Portable Network Graphics format (**.png** file extension)

To convert existing references and image files to shared images, use the **ImageExtractor.bat** utility. See [ImageExtractor-jar-ImageExtractor-bat](#).

To create an image object

1. In AR System Navigator, expand **serverName > All Objects**.
2. Right-click **Images**, and select **New Image**.
A blank image object opens in the editing area.
3. In the **Description** field, enter a description of the image.
The description appears in the list of images in the Object Selector dialog box when you add an image to a field or form view. Enter up to 255 characters.
4. Click **Browse**, and navigate to the directory containing the image.
5. To display the available images in the directory, select the various image types from the Files of Type list.
6. Select the image to store in AR System, and click **Open**.
The image appears in the preview pane of the image object.
7. Save the image object, assigning a unique name in the Save Image As dialog box.
Images are indexed on the name you assign and by the image ID. Image names can have up to 255 characters.
Although AR System server permits backslashes in image object names, you should *not* include them. To prevent attacks on your system, you should exclude unsafe characters from your web server. By default, most web servers do not allow the backslash character (\) in URLs for security reasons. For example, Tomcat does not allow backslashes in URLs unless you set a parameter to permit them. If your system uses such a web server, images whose object name contains backslashes might not appear in your browser.

To open an image object

1. In AR System Navigator, expand **serverName > All Objects**.
2. Double-click **Images**.
3. From the Images list, double-click the image object to open.

To save an image to a file outside AR System server

1. Open the image object, and then click **Save to Disk**.
2. Navigate to the directory where you want to save the file.
3. Enter an appropriate file name, and then click **Save**.

To rename an image object

1. In AR System Navigator, expand **serverName > All Objects**.
2. Double-click **Images**.
3. Right-click the image to be renamed, and then select **Rename**.
4. Enter the new name, and then save the image object.

You cannot rename an open object in Developer Studio. If a warning indicates the object has an open editor, close the image and repeat these steps.

To delete an image object

Warning

Deleting a shared image object removes it from all form views and fields. You cannot undo this operation in Developer Studio.

1. In AR System Navigator, expand **serverName > All Objects**.
2. Double-click **Images**.
3. Right-click the image to be renamed, and then select **Delete**.
4. Confirm the deletion when prompted.
5. In the second confirmation dialog, specify whether to remove references to the image from the display properties of fields and objects.
 - Click **Yes** to remove the reference from the listed fields or objects.
 - Click **Yes to All** to remove all references to the image.
 - Click **No** to retain the reference in the listed field or object.
 - Click **No to All** to retain all references to the image.

If you retain a reference, the image object name stays in the display property, and you can create another image object with the same name to restore the reference.

File:

Developing-Pwa\Adding-Icons-And-Images\Previews-Of-Icons-

As a developer who wants to add icons to your Progressive Web Application (PWA), you need to see a preview of the icons that you want to add. You can sign in to this documentation page and view the PDF listing the icon previews.

Related topics

[Adding-icons-and-images-to-Progressive-Web-Applications](#)

[Adding-icons-to-UI-elements-on-forms-by-using-Developer-Studio](#)

For a Progressive Web Application (PWA), two properties are available in Developer Studio, namely Icon and Icon Position. Use these properties to add and position icons to the UI elements on a form. For example, you can add an icon to a button in Developer Studio. The following table provides more information about the properties:

Property name	Property type	Default value	Description
Icon	String	N/A	You can specify the name of an icon as a value for this property.
Icon Position	String	Left	You can specify the position of an icon as a value for this property.

The following fields in Developer Studio support icons:

- Button
- Character
- Thumbnail
- Trim Text
- Vertical Navigation

PDF with previews of icons

The following PDF (sign in required) lists the predefined icons and the corresponding icon class names that are available Out-Of-The-Box (OOTB) in Developer Studio for PWA.

You must [log in](#) or [register](#) to view this page

File:

Developing-Pwa\Adding-Rich-Text-Fields\Adding-Rich-Text-Fie

As an application developer, you can add character fields as rich text format (RTF) fields to enable your application users to add formatted text, add images along with text, and so on.

Application users can perform the following tasks by using RTF fields:

- Edit and format text.
- Add content from Microsoft Word, Microsoft Outlook, or HTML pages.
- Add images and links along with text.

Important

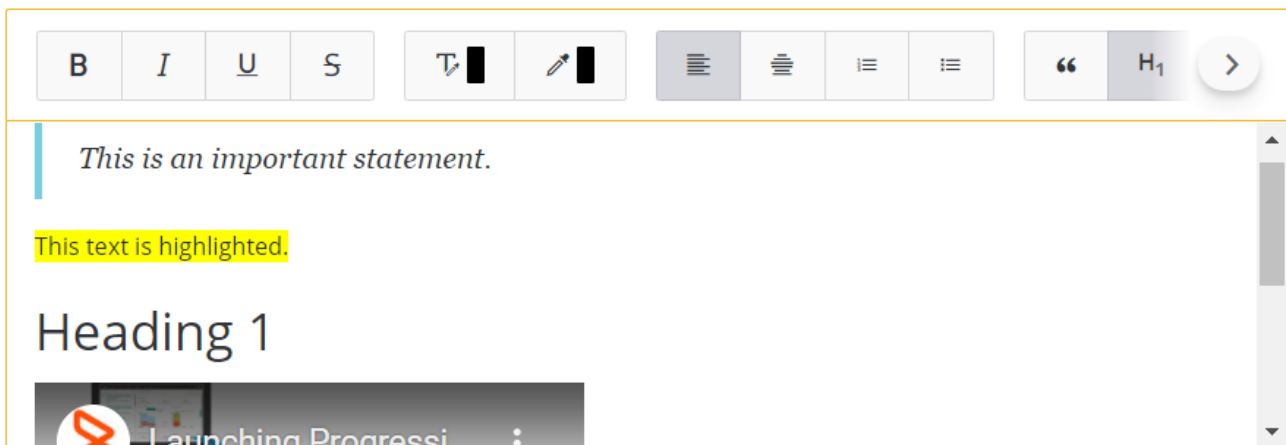
Rich text fields are displayed as plain text fields in mobile views.

Related topics

[Adding-icons-and-images-to-Progressive-Web-Applications](#)

[Customization-best-practices-for-developing-PWA](#)

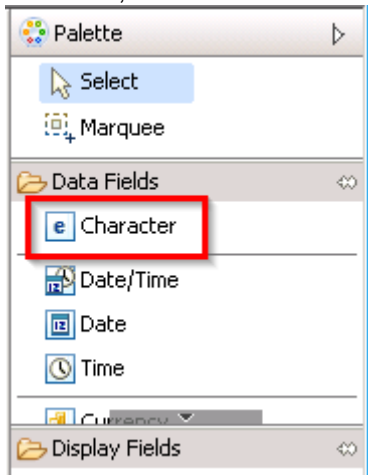
[Enabling rich text fields on PWA screens by running a utility](#)



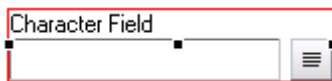
To add RTF fields or advanced RTF fields in Progressive Web Applications

1. In Developer Studio, create a progressive view.

2. In Palette, in the Data Fields section, drag **Character** into the view.



3. Select the character field.



4. On the Properties pane, perform one of the following steps:

- To create a simple rich text field, in the **Display > Display Type** list, select **Rich Text**. This option creates an RTF field in PWA but does not provide you any additional configuration options.

To create an advanced rich text field in the **Display > Display Type** list, select **Advanced Rich Text**. Select **Advanced Rich Text** to create an RTF field along with additional configuration options. For example, you can select the RTF toolbar options that are displayed in the application along with other features described in the following section. In the select **Display > Display as text** list, select **True** to display the rich text in view-only mode.

Best practice

If you source text from a rich-text-enabled field in Mid Tier and display it within a field in a PWA screen, you must make sure that the field in PWA also has rich text enabled. In such a scenario, if you do not enable rich text in the field in PWA, the field may display text along with the HTML source.

5. Save the changes.

To perform additional configurations to advanced RTF fields in Progressive Web Applications

Optionally, you can add additional properties to the advanced RTF (ARTF) fields after you select **Advanced Rich Text** in Developer Studio to do the following:



- Add a toggle that enables application users to switch the RTF field to show plain text.
- Enable application users to use @mention in the RTF fields.
- Show or hide advanced RTF palette options.


Enable @mention to the rich text field.

1. On the **Properties** panel, under the Display property group, for the Enable Menu Hotkeys property, select the **True** value.
2. In the **Value** column for the Menu Hotkeys property, click the ellipsis button.
The **Menu Hotkeys** dialog box is displayed.

Hotkey	Menu Name	Active Link	Labels	Auto Complete Key Stroke #	RTF Labels
@					
!					
#					
\$					
%					
^					
&					
*					
?					
+					
-					
~					

Click **Add**, and then perform the following steps:

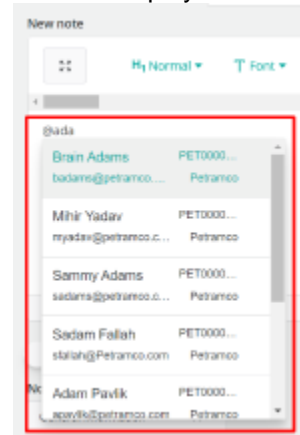
Property		Step
Hotkey		Click the first row in this column, and then select a character from the list. When application users type this character, the @mention list appears. Click the first row in this column, and then click the ellipsis button  to select a menu from the Menu Selector dialog box. This is the menu from which information for the @mention list is obtained. Click the first row in this column, and then click the ellipsis button  to select an active link from the Active Link Selector dialog box.
Menu Name		
Active Link		The active link that you attach to a hotkey must be an existing one and associated with the current form.
Labels		Leave this column empty. This column is related to only character fields without any RTF enabled. Enter an integer value.
Auto Complete Key Stroke #		This is the number of characters application users have to enter before the @mention list appears.

Click the first row in this column, and then click the ellipsis button  to select the fields from which the @mention list obtains data.

For example, if you specify a form with all user details in the Menu Name column, you can select the fields representing the following data:

- User name
- Device ID
- User email
- Company name

In such a scenario, the @mention in the PWA screens displays the information as shown:



RTF Labels

The displayed order of the information is the same as the order in which they are in the RTF Label column.

Important

- The @mention list in the PWA screen displays only the first four fields you select in the RTF Labels column.
- If you select more than four fields, the @mention performs the search in all those fields even though it displays only the first four fields in the PWA screen.
- If you do not specify anything in the RTF Labels column, Developer Studio uses the first four available fields in the order specified in the form and then displays it in the @mention list in the application.

Best practice

Since @mention searches through all the fields you add in the RTF Label column, make sure you add those fields which make search easier to the application users.

- Click **OK**.
- Show only relevant RTF field options.
 - On the **Properties** panel, beside the **Display > Additional RTF Toolbar Configuration** property, click the field.

Enter the list of options you want the RTF field to display with the following syntax:

Option1, Option2 | Option3, Option4, Option5

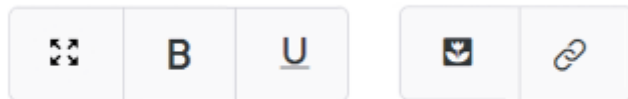
The pipe character (|) specifies that the options after it are in a new group. The syntax is case sensitive.

Example syntax for adding RTF toolbar options

For example, enter **Maximize, Bold, Underline | Image, Link** to display the Maximize, Bold, and Underline options with the RTF field.

Additional RTF Toolbar Configuration	Maximize, Bold, Underline Image, Link
--------------------------------------	---

The PWA screen displays the following options on the RTF toolbar:



Values for RTF toolbar options

You must use the following values for the corresponding RTF toolbar options:

Type	RTF toolbar option	Description	Value
Basic style 	B	Bold text	Bold
	Italicize text	Italic	
	Underline text	Underline	
	Strikeout text	Strike	
	Subscript	Subscript	
	Superscript	Superscript	
	Remove formatting	RemoveFormat	
	Copy formatting	CopyFormatting	
Style 	H₁ Format ▼	Format	Format
	Font	Font	
	Font size	FontSize	
Color 	Text color		TextColor
	Background color	BGcolor	

Editing



Paragraph



Link



Insert



clipboard



Replace

Select all



Indent

Justify to the left

Justify to center

Justify to right

Justify

Numbered list

Bulleted list

Quote



Remove link

Add anchor



Add table

Add horizontal rule

Embed YouTube
video

Insert page break



Copy

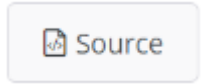
Paste

Paste as plain text

Paste Word content

Undo

Redo



Find

Replace

SelectAll

Outdent

Indent

JustifyLeft

JustifyCenter

JustifyRight

JustifyBlock

NumberedList

BulletedList

Blockquote

Add link

Unlink

Anchor

Add image

Table

HorizontalRule

Youtube

PageBreak

Cut

Copy

Paste

PasteText

PasteFromWord

Undo

Redo

Maximize

View as source

Find

Outdent

Link

Image

Cut

Maximize

Source

3. Save the changes.

To validate the RTF field or advanced RTF field in PWA

1. In a browser, log in to PWA.

<serverName>:8080/arsys/pwa/#/login

2. Change the URL to include the Progressive View form that you created in Developer Studio.

<serverName>:8080/arsys/pwa/#/forms/<serverName>/<formName>/<viewName>

(Optional) If you want to display more changes you made in Developer Studio, you must sync the cache in Mid Tier.

1. Log in to Mid Tier configuration.

<serverName>:8080/arsys/shared/config/config.jsp

2. Click **Sync Cache**.

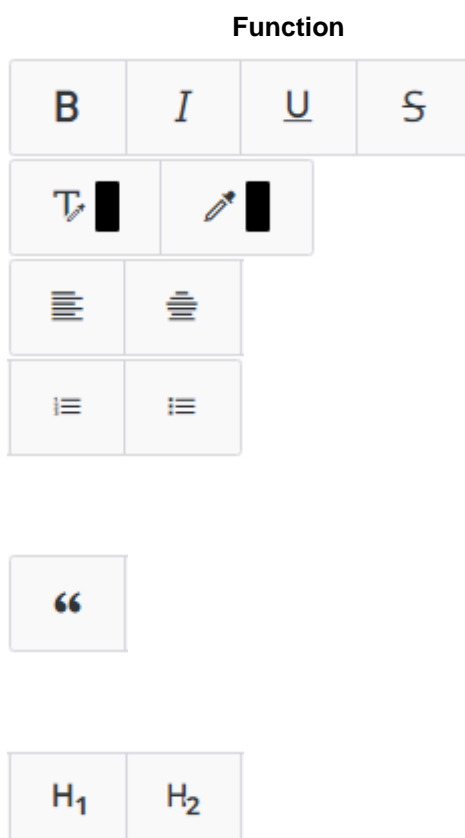
3. Refresh the browser displaying the progressive view form.

4. As an application user, you can perform the following tasks in the RTF editor:

- Add content that uses standard HTML markup, this RTF editor renders the content without any loss of formatting.
- Edit the existing rich text content.
- Copy content from Microsoft Word, Microsoft Outlook, or HTML while retaining most of the supported formatting.

You can view the following options in the RTF editor and advanced RTF editor:

Options in the RTF editor



Description

Make the selected text bold, italicized, underlined, or strikethrough.

Select the text color and text highlighting background.

Set the paragraph so that it aligns evenly on the left or center, or justifies the text.

Create an ordered or unordered list.

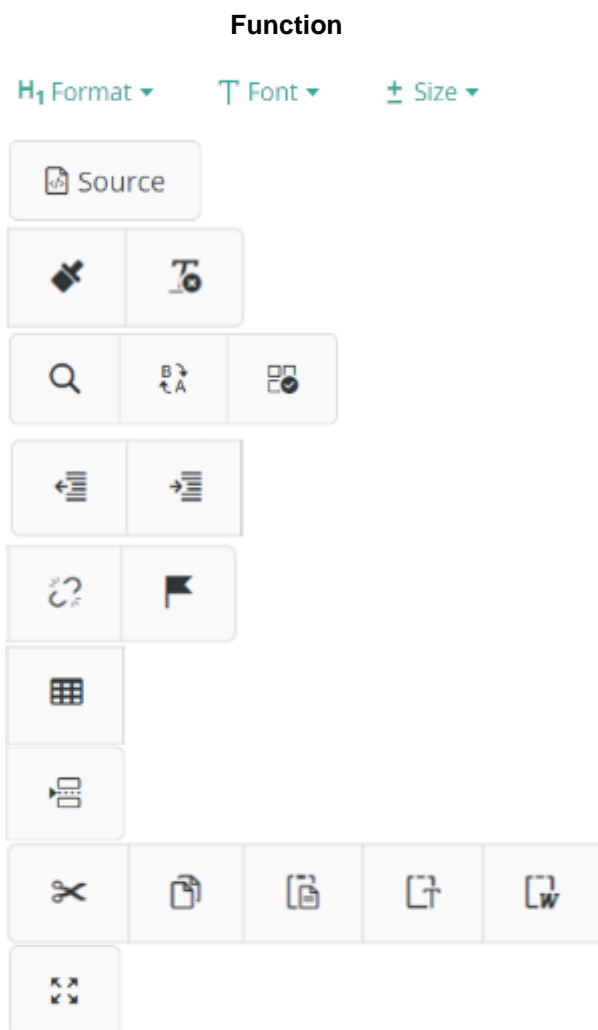
Create special formatting to any text so that it is displayed prominently as shown in the following example:

This is an important statement.

Set paragraph formatting style such as headings.



Additional options in advanced RTF editor



How does advanced RTF handle HTML and domains?

- Any content created in earlier releases that has HTML is executed and rendered according to the HTML tags.
- HTML tags are compared to the Administration email configuration whitelist.

Make the selected text subscript or superscript.

Embed a YouTube video when you provide the embed URL.

Insert a URL.

The URL must contain the protocol such as <http://> or <https://>.

For example,

enter <http://www.docs..com> instead of entering www.docs..com.

Insert an image into the RTF field.

You can also copy and paste images into the RTF field.

The undo button removes the last 30 changes made in the RTF editor or reverses the undo action.

Description

Set text formatting style such as headings, font, and size.

Display the text along with source.

Copy formatting and remove text formatting.

Search, replace, and select all text.

Text indentation.

Remove hyperlink from text and add an anchor.

Insert tables.

Insert page breaks for printing.

Cut, copy to clipboard, paste from clipboard, paste as plain text, and paste from Word

Maximize the field to occupy the browser screen.

- If the tags are not in the whitelist, they are removed and then displayed by the advanced RTF field.

File:

Developing-Pwa\Configuring-The-Progress-Bar-Widget\Configuring

The Progress bar widget is a component that you can configure to show the progress of a particular event or the steps of a particular process. You can also configure it to show the milestones on a timeline. For example, you can use this widget to display the status of your ticket. Mid Tier displays this widget on a form for which a progressive view is enabled. In Developer Studio, you can enable the Progress bar widget by adding a table, and then setting the Display Type of the table to **Progress Bar View**.

A table on a form represents one Progress bar on the UI and the number of rows in that table represent the number of nodes on the Progress bar.

Related topics

[Developing-Progressive-Web-Applications](#)

[Defining-tables-to-display-data](#)

[Flow-panel-holders-for-Progressive-Web-Applications](#)

The following video (3:37) describes how to configure and use the progress bar widget:

■ Watch Video: https://www.youtube.com/watch?v=-T90EgZ_ZtI

▶ https://youtu.be/-T90EgZ_ZtI

The following image shows the Progress bar widget:



Before you begin

- To configure a Progress bar widget, you must have a minimum of six columns selected from a table.
- You must define the mappings for all columns.
- You can't define a duplicate mapping.

To configure the Progress bar widget

1. On the progressive view of a form, double-click a Table field.
2. On the **Properties** panel, under the Display property group, for the Display Type property, select the **Progress Bar View** value.
3. In the **Value** column for the Tree/Table Property property, click the ellipsis button.
The **Tree/Table Property** dialog box is displayed.

Tree/Table Property

Data Source: SERVER

Server Name:

Form Name:

Qualification:

Fields from Remote Form and Local Form Untitled Regular Form 17

Field Name	Field Type	Data Source
Assigned To	Character	Data
Create Date	Date/Time	Data
Last Modified	Character	Data
Modified Date	Date/Time	Data
Request ID	Character	Data
Short Description	Character	Data
Status	Selection	Data

Table Columns

Column Title	Field	Field Type
--------------	-------	------------

Up

Down

OK Cancel

4. Perform the following steps:
 1. In the **Form Name** box, add any form from where you want to fetch the data in the table.
 2. In the **Qualification** box, add a condition as per your requirement.
 3. Add the minimum required six columns for the mappings.
You can add any additional columns, if required.
 4. Click **OK**.
5. Click a column in the table.
6. On the **Properties** panel, under the Display property group, in the Progressbar Field property, define the required mappings such as title, status, and instance ID.
7. Double-click the Table field.
8. (Optional) On the **Properties** panel, under the Display property group, for the Mapping Identifier property, enter a value.

If you don't specify any value, the default colors are displayed on a Progress bar. To learn how to define a color for the Progress bar, see [AR System Progress Bar Widget Mapping form](#).

9. (Optional) On the **Properties** panel, under the Display property group, for the **Show Date As** property, select any option.

You can specify whether the tooltips on the SLA progress bar display absolute dates, relative dates, or both.

AR System Progress Bar Widget Mapping form

While designing a Progress bar, if you want to define a color for the Progress bar based on certain rules, you can use the AR System Progress Bar Widget Mapping form. The value in the **Mapping Identifier** box on this form must match the value for the Mapping Identifier property in Developer Studio. You can use this form before or after setting the properties for the Progress bar in Developer Studio.

The following table provides information about the fields available on the AR System Progress Bar Widget Mapping form:

Field name	Field type	Description
Mapping Identifier	Character	Specifies the value that you must provide for the in a table to attach a mapping.
Status Value	Character	Specifies the status value for which you want to set the bar color.
Progress Bar Color	Selection:	Specifies the color of the bar for the provided status value.
	<ul style="list-style-type: none">• Danger• Success• Warning• Info• Inactive	
Precedence	Integer	Specifies the precedence for the overall status of the bar, in case there are multiple SLTs on the same bar.
Node Icon	Selection:	Specifies the icon for the node on the progress bar.
	<ul style="list-style-type: none">• Check• Cross• None	
Node Fill Color	<ul style="list-style-type: none">• Danger• Success• Warning• Info• Inactive• None	Specifies the color for the node on the progress bar.

File:

Developing-Pwa\Configuring-The-Results-List-In-Progressive-V

Related topics

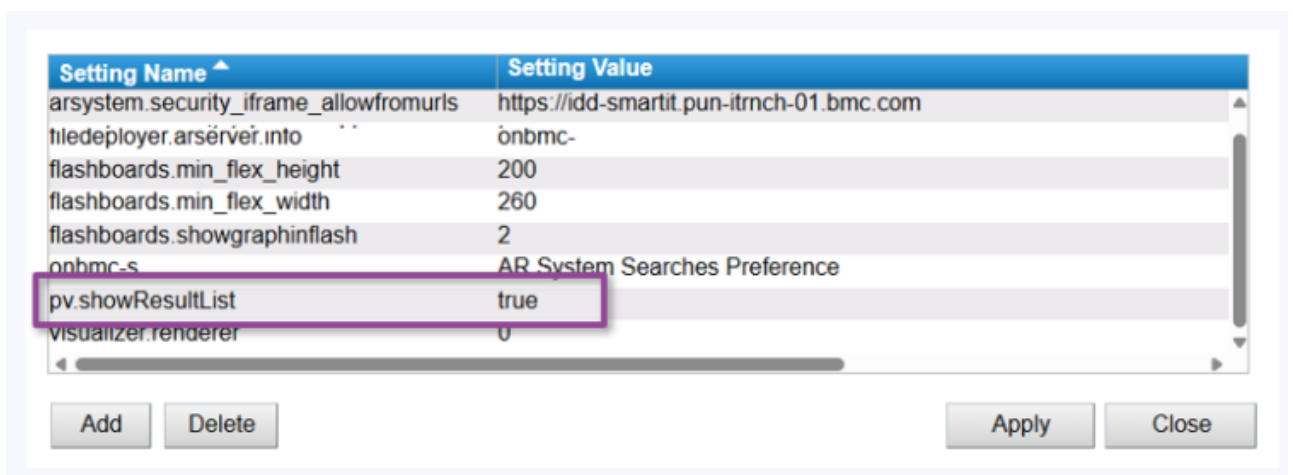
[Enabling Progressive Views for forms by using Centralized Configuration Settings](#)

[Updating-configuration-settings-by-using-the-AR-System-Configuration-Generic-UI-form.](#)

As an administrator, configure the Results List in Developer Studio to provide users with a modern interface for managing records. Use the Results List in Progressive View to perform actions such as creating, searching, or updating records directly within a form in IS.

Task 1: To enable the CCS parameter in Mid Tier

1. In a browser, open the AR System Administration Console, and select **System > General > Centralized Configuration**.
2. In the AR System Configuration Generic UI form, from the **Component Name** list, select **arsystem..arsys.midtier-DockerMTClusterUser** component to add a CCS parameter.
3. Click **Add**.
4. Enter **pv.showResultList** name of the setting and set its value to **true**. For more information, see [Configuration settings N-R](#).
- The following image shows the **pv.showResultList** property set to **true**:



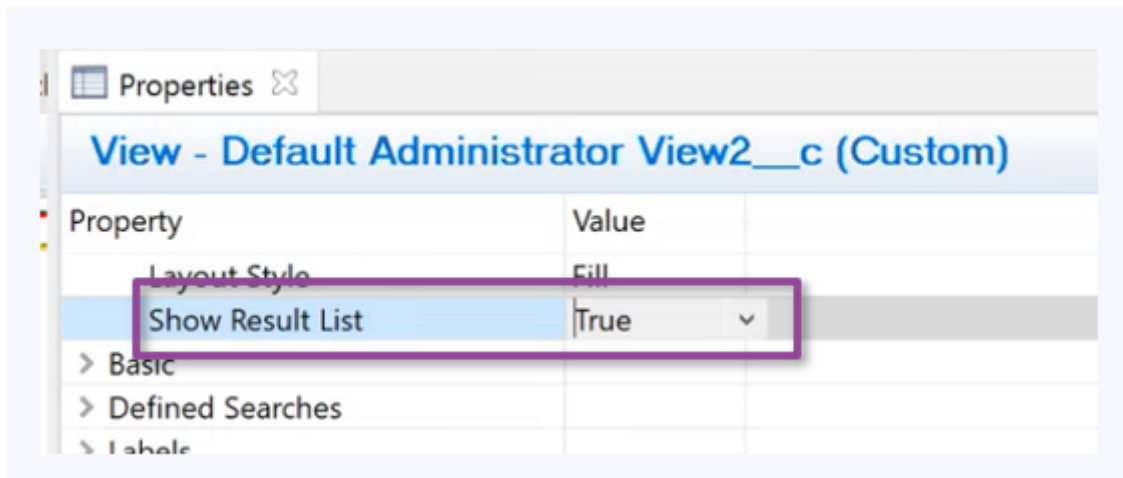
5. Click **Apply** and **Close**.
6. Flush the cache from the Mid Tier. For more information, see [Configuring cache settings for the Mid Tier](#).

Task 2: To enable the Results List in Developer Studio

Open a form in Developer Studio.

2. Select the form for which you want to enable the Results List and navigate to the **Properties** tab.

- On the **Properties** tab, under the **Appearance** section, set the **Show Result List** property to **True**. For more information, see [Setting form view properties](#).
- The following image shows the **Show Result List** property:

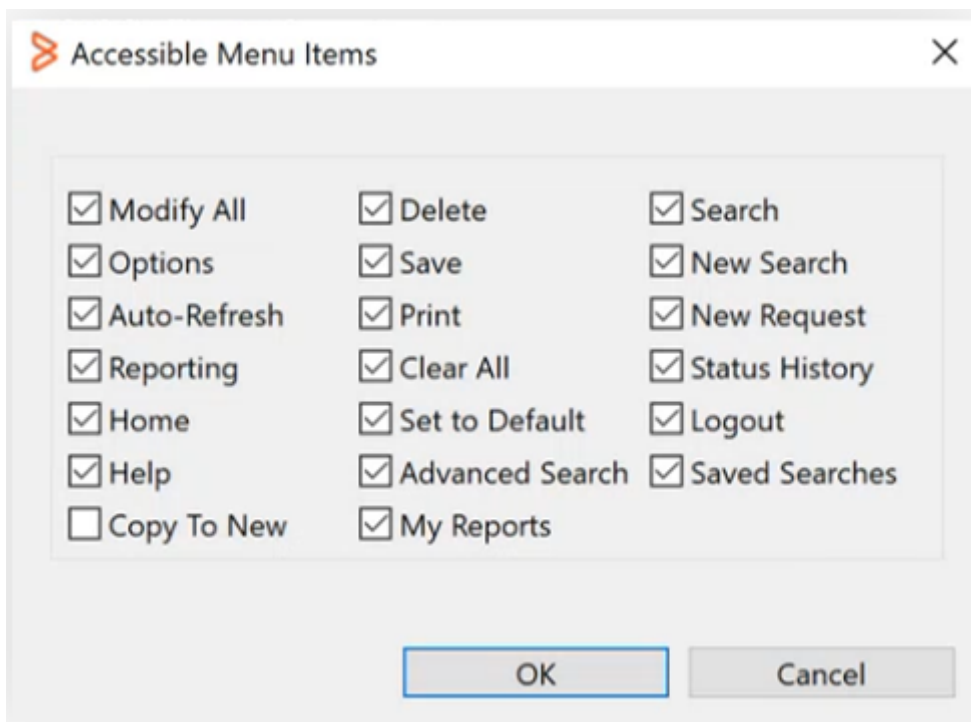


Important

The Results List is view-specific. If a form has multiple views, enable the property for each view where you want the Results List to appear.

Task 3: To configure the Results List in Developer Studio

- On the **Properties** tab, under the **Menu Access** section, click **Accessible Menu Items** to select the actions to be visible in the Results List. For more information, see [Setting form view properties](#).
- The following image shows the **Accessible Menu Items** box:



- On the **Properties** tab, under the **Result List** section, set the **Size of Chunk** value. For more information, see [Setting form view properties](#).
- The following image shows the **Size of Chunk** property with its value set to zero:

▼ Results List	
Initial Row Selection	Select First,...
Size of Chunk	0
Results Color	(none)
Table Drill Down	True ▼

-
3. To use the Results List in Progressive View, open the form where you've enabled it in a browser. The following image shows the Results List interface in the form opened in PWA screen:

The screenshot shows the BMC Helix Progressive View interface. At the top, there is a navigation bar with the BMC Helix logo and the text "Progressive View". Below this is a search bar with a magnifying glass icon and a "Current mode: Search" button. The main content area contains a form with several fields: "Submitter" (text input), "Drop-Down List Field_c" (dropdown menu with "Select" as the current value), "Date/Time Field_c" (text input with "\$TIMESTAMP\$" as the value), "Short Description" (text input), "Radio Button Field_c" (radio button), and "Date Field_c" (text input). The interface is clean and modern, with a light blue and white color scheme.

Although many of the menu options available in applications built by using Developer Studio are supported in Progressive View for creating Progressive Web Applications, some features, such as Reporting, are not available in Progressive Web Application screens. To see a comparative list of Results List menu items available in Mid Tier and Progressive View, see [Differences between Developer Studio features available for Mid Tier and Progressive Web Applications](#).

File:

Developing-Pwa\Creating-A-Popup-Window\Creating-A-Popup-

In a Progressive View, an open popup window looks like an action blade. The popup window does not open in a new window or on a new tab. Instead, it appears as a child window as an action blade by using a docked panel component. The popup window closes automatically when you close the parent window. You cannot move a popup window outside the parent window.

To specify the width of a docked panel:

The blade width is derived from the Width property of the view that you are using in the Open Window action. The following is the behavior:

- (View Width \leq 250 AND View Width $>$ 0) = Small = 25% of the whole page
- (View Width $>$ 250 AND View Width \leq 500) = Medium = 50% of the whole page
- (View Width $>$ 500 AND View Width \leq 750) = Large = 75% of the whole page
- (View Width $>$ 750) = Extra large = 100% of the whole page

The following image shows an open popup window in a desktop view:

The image shows a desktop view of a web application. On the left is a main form with various fields: 'Menu Field (required)', 'Request ID', 'Last Modified By', 'Create Date', 'Assigned To', 'Submitter (required)', 'Modified Date', 'DropDown Field', and 'Date Field_c'. To the right of the main form is a vertical toolbar with buttons: 'Call set field', 'Save Button', 'Call Service Action', and 'Call Open Window'. On the far right is a docked panel titled 'Simple DockedPanel' which contains a duplicate of the main form's fields. The docked panel has a close button (X) in the top right corner and three buttons at the bottom: 'Commit', 'Set Field', and 'Open Popup'.

The following image shows two open popup windows in a desktop view:

Menu Field (required)

Request ID

Last Modified By

Create Date

Assigned To

Submitter (required)

Modified Date

DropDown Field

Select

Date Field_c

Simple DockedPanel

×

Request ID

Submitter (required)

Create Date

Assigned To

Last Modified By

Modified Date

Status

Short Description (required)

Testing

Commit

Set Field

Open Popup

Simple DockedPanel

➤

Request ID

Submitter (required)

Create Date

Assigned To

Last Modified By

Modified Date

Status

Short Description (required)

Commit

Set Field

Open Popup

The following image shows a mobile view, where the action blade occupies full-screen height and width, which is the default behavior of the docked panel component:



Simple DockedPanel



Request ID

Submitter (required)

Create Date



Assigned To

Last Modified By

Modified Date

☐ Status

Short Description (required)

Commit

Set Field

Open Popup

Related topics

[Developing-Progressive-Web-Applications](#)

File:

Developing-Pwa\Creating-A-Progressive-View-In-Developer-Stu

In Developer Studio, create a Progressive Web Application (PWA) by creating the components of the application interface such as fields, forms, and workflows in a Progressive View.

Before you begin

You must enable Progressive View for forms by using the Centralized Configuration. To know more, see [Enabling Progressive Views for forms from the Centralized Configuration](#).

To create a Progressive View in Developer Studio to create Progressive Web Applications

1. In Developer Studio, create a New Regular Form.

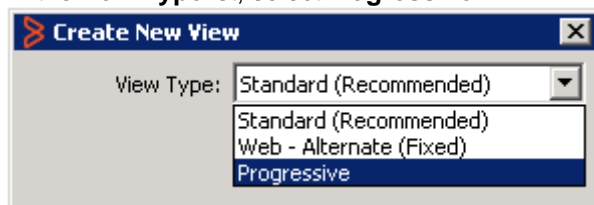
Create a Progressive View.

Steps (2)

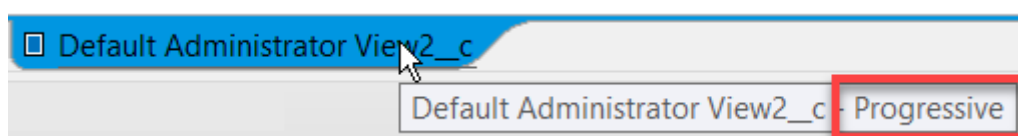
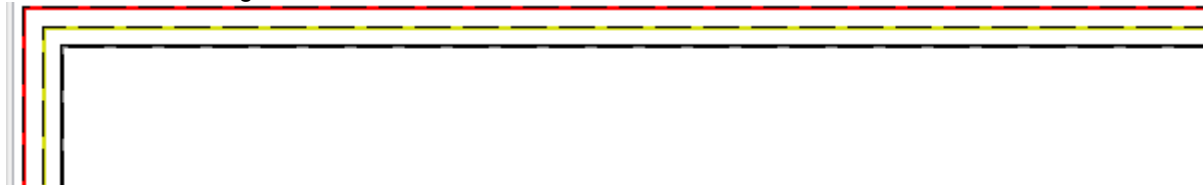
1. Select **Form > Create New View**.

The Create New View dialog box opens.

2. In the **View Type** list, select **Progressive**.



The Progressive View appears with panel holders having the following colors and the tooltip at the view tab is suffixed with "Progressive".



Where to go from here

[UI-layout-of-Progressive-Web-Applications](#)

[Flow-panel-holders-for-Progressive-Web-Applications](#)

Related topics

[Enabling Progressive Views for forms from the Centralized Configuration](#)

[Developing-Progressive-Web-Applications](#)

[Accessing-and-navigating-Developer-Studio](#)

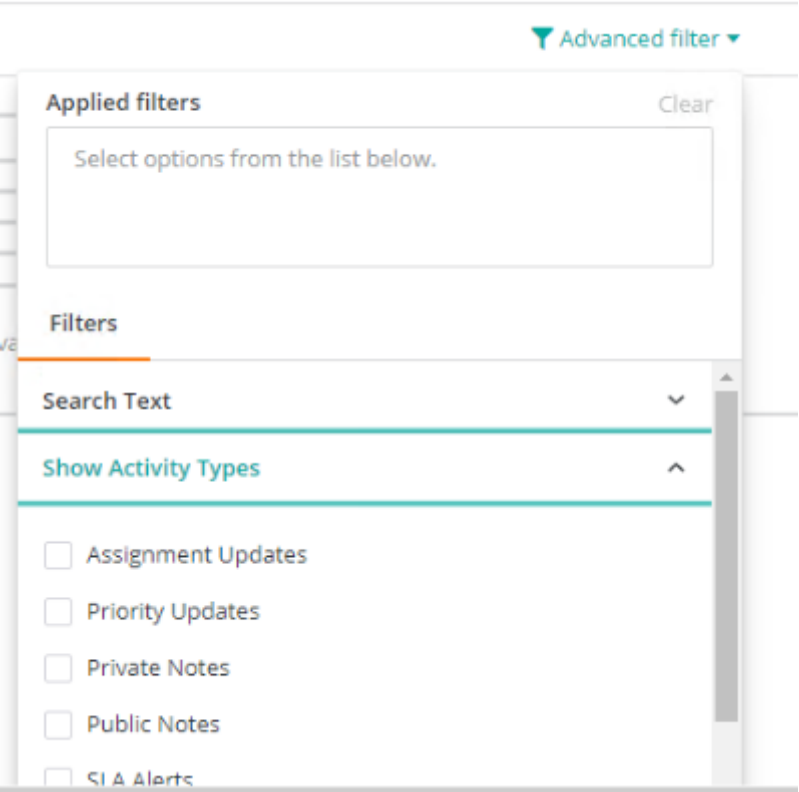
[Developing-applications](#)

File:

Developing-Pwa\Creating-An-Advanced-Filter\Creating-An-Adv

The Advanced filter component enables you to apply advanced filtering options to the fields on a form for which a Progressive view is enabled. You can configure this component on the Table field in Developer Studio. After you configure this component, you can view it in a Progressive view. The Advanced filter component is a UI element. It is not related to the Filter, which is a workflow component in AR System.

The following image shows an Advanced filter in a progressive view:



Related topics

[Defining-tables-to-display-data](#)

[Creating-a-faceted-filter](#)

The following table lists the characteristics of the Advanced filter:

Characteristic	Example
Dependent filter values	Depending on the value that you select in one filter, the values in another filter are populated.
Filter options within a filter	The Asset Console features options, such as People, Support Group, and Organization.
Filter within filter	The Operating System section in the Asset Console features filters within a filter.

Specifying a custom display by overriding type	The CPU Count section in the Asset Console lets you specify a custom display by overriding type.
Specifying the display position of a filter	The Change Calendar filter in Change Request section lets you specify its display position.
Default options based on the logged in user	The To be approved by (Me, My Support Groups) and Schedule dates (24H, 48H) filters in the Ticket Console section feature the default options based on the logged in user.
Associating a menu to a filter	The filters in the Ticket Console section associate a menu with them.

To support the Advanced filtering component for progressive views, the following properties are added in Developer Studio:

Property name	Applicable for field	Property value	Description
Filter	<ul style="list-style-type: none"> Table - List View Table - Cell Based 	<ul style="list-style-type: none"> True False (Default) 	Specifies if the Advanced filter is to be enabled at table-level.
Filter External ID	<ul style="list-style-type: none"> Table - List View Table - Cell Based 	A field ID	Specifies the external ID of the Advanced filter.
Filter	Column	<ul style="list-style-type: none"> True False (Default) 	Specifies if the Advanced filter is to be enabled at column-level.
Display As Select	Column	<ul style="list-style-type: none"> True False (Default) 	Specifies if the options in the Advanced filter are to be displayed as check boxes that you can select.
Use Value As Qualification Column		<ul style="list-style-type: none"> True False (Default) 	Specifies if the value of a menu that contains a qualification is to be used as is when you apply the Advanced filter.
Filter QBE Match	Column	<ul style="list-style-type: none"> Anywhere Equals Leading (Default) 	<ul style="list-style-type: none"> Leading—The leading characters in the query string are matched. Equals—The characters in the query string are precisely matched. Anywhere—The characters in the query string are matched irrespective of their location.

Filter Display Order	Column	An Integer	<p>Specifies the order for a menu item in the Advanced filter.</p> <p>Specifies a search menu for the filter. You have to set the type Search for the menu. You can use this only when the Filter property value is set to True and is mapped to a character field.</p> <p>This property takes precedence over the Menu Name property under Attributes of the character field.</p>
Menu Name	Column	<p>You must associate a menu of type search to this property.</p>	<p>The following video clip shows how you can associate a Menu Name property with a predefined search menu. In this example, we associate Menu Name with Menu:Features so that end users in the custom application see searchable options from Menu:Features.</p> <p>Another example where you can use this property is for usernames that you want displayed when users type in a search field.</p>

Button Field	Button	You must associate a field of type filter to the Button Field property	<p>Displays a filter in place of a button. Use this property to provide a button ID to display the filter in place of a button. This property is available in the Filter section.</p> <p>The following video clip shows how you can display a filter in the place of a button. The initial steps in Developer Studio involve associating a predefined filter with the button:</p>
Groups	Table	You must associate a group.	<p>Creates a group of filters within one section.</p> <p>Click the property to launch a dialogue where you can define a group. This property is available only when you set the Filter Type to Advanced.</p> <p>Only the following filters types are allowed to be part of groups:</p> <ul style="list-style-type: none"> • Character This can be with or without a menu. • Integer • Selection This filter type is always rendered as checkboxes. <p>The following video clip shows how you can add a filter to a filter group:</p>



Show Apply Button	<ul style="list-style-type: none"> Table - List View Table - Cell Based 	<ul style="list-style-type: none"> False (Default) True 	<p>Specifies if the Apply filters button is to be shown on the filter. In addition, it also displays the Cancel and Remove all buttons.</p> <p>This property is available in the Filter section only when you set the Filter Type to Advanced.</p> <p>The following video clip shows how you can add the Apply filters, Cancel, and the Remove all buttons.</p> <p>Set this property to True to display a filter.</p>
Filter Visibility	Table Column	<ul style="list-style-type: none"> False (Default) True 	<p>This property is available in the Filter section. You can add this property if the column Filter property value is set to True.</p> <p>Generates a filter qualification when end users select options.</p> <p>To set this property, the Filter property must be set to True and Table Type must set to Advanced.</p>
Generate Qualification	Table Column	<ul style="list-style-type: none"> False (Default) True 	<p>Use the following command in an active link along with the Generate Qualification property to fetch the generated qualification:</p>

To configure the Advanced filter component

You must set the values for the properties that are available for the Table field and the Column field.

To set the values for the properties for the Table field

1. On the progressive view of a form, double-click a Table field.
2. On the **Properties** panel, under the Attributes group, for the Filter property, select the **True** value.

3. Beside the Qualification Field property, in the Value column, click the ellipsis  button.
The Qualification Field dialog box opens.
4. Select a field from the available fields, and then click **OK**.
The values for the properties for the Table field are set.
5. Under Attributes, beside Tree/Table Property, in the Value column, click the ellipsis  button and add the following additive qualification:

EXTERNAL(\$qualificationField\$)



The screenshot shows the 'Tree/Table Property' dialog box. The 'Qualification' field is highlighted with a red rectangle and contains the text 'EXTERNAL(\$Character Field_c\$)'. Above this field, there are dropdown menus for 'Data Source' (set to 'SERVER'), 'Server Name' (empty), and a text field for 'Form Name' (set to 'User'). Below the 'Qualification' field, there are two sections: 'Fields from Remote Form and Local Form adv' and 'Table Columns'. The 'Table Columns' section has a table with two columns: 'Column Title' and 'Field'.

6. Save the changes.

To set the values for the properties for the Column field

1. In the **Value** column for the Tree/Table Property property, click the ellipsis button.
2. On the **Tree/Table Property** dialog box, add a field as a table column, and then click **OK**.
3. Double-click the column field on the table, and then for the Filter property, select the **True** value.
4. Drag a Character field to the form, and then double-click it to view the **Properties** panel.
5. In the **Value** column for the Menu Name property, click the ellipsis button.
6. On the **Menu Name** dialog box, select a menu of type Search from the available menus, and then click **OK**.
7. Double-click the Table field, in the **Value** column for the Tree/Table Property property, click the ellipsis button.
8. On the Tree/Table Property dialog box, add the Character field with a menu that you've added in step 6 as a table column, and then click **OK**.
9. Double-click the column that you've added in step 8, and then for the Filter property, select the **True** value.
10. Select a value for the following properties as required:
 1. Display As Select
 2. Use Value As Qualification
 3. Filter QBE Match
 4. Filter Display Order
11. Click **Save** to save your settings.

Tip

To display the filter labels in alphabetical order on the PWA screen, you must select **Sort On Label** in the menu as shown in the following example:

The screenshot shows a configuration window for 'SHR:TKD:SV_ActivityFilter_ShowActivityTypes'. It includes fields for 'Form Name' (SYS:Menu Items) and 'View Name'. Below these is a 'Label Fields' table with one entry: 'Menu Lab...' with Field ID '1000000008' and Data Type 'Character'. To the right of the table are buttons: 'Add..', 'Remove', 'Remove All', 'Up', and 'Down'. Below the table is a checkbox labeled 'Sort On Label', which is highlighted with a red rectangle. Below this is a 'Value Field' (Workflow Extras) and a 'Qualification' field containing the expression: '(Menu Type' = \$304438401\$) AND ('Form Name01' = \$302300503\$) AND ('Menu Visible' = "Yes")

Name	Field ID	Data Type
Menu Lab...	1000000008	Character

☒ Sort On Label

Value Field: Workflow Extras

Qualification: (Menu Type' = \$304438401\$) AND ('Form Name01' = \$302300503\$) AND ('Menu Visible' = "Yes")

File:

Developing-Pwa\Creating-Dynamic-Selection-Lists-Using-Json

Fetch JSON data and pass it to a character menu in Developer Studio to create a list or to populate a field in Progressive Web Applications. Use this method of creating dynamic selection lists in situations where the content to be displayed varies according to various user actions and settings. You can fetch JSON data by using webhooks, REST filters, or active links.

The following example procedure demonstrates:

- How to use the **JSON Menu Label**, **JSON Menu Value**, and **JSON Field ID** properties in a character field to extract values from JSON data and display the data as a list.
- How to create an active link that uses the JSONMAP function to extract a specific value from JSON data and display it in a character field.

This example shows how the **JSON Menu Label**, **JSON Menu Value**, and **JSON Field ID** properties, and the JSONMAP function work. However, note that real-world implementations can vary considerably depending on your business needs.

The following video (4:06) explains the concept and also demonstrates how the JSON properties and the JSONMAP function works:

■ [Watch Video: https://www.youtube.com/watch?v=3eRIML5mOgl](https://www.youtube.com/watch?v=3eRIML5mOgl)

▶ <https://youtu.be/3eRIML5mOgl>

Before you begin

- You must have a basic understanding of progressive web applications and their development. For more information, see [Developing-Progressive-Web-Applications](#).
- You must know how to fetch JSON data by using workflows that have webhooks, REST filters, or active links.
- You can download the [JSONMAP_example_fields.deffile](#) and import it into Developer Studio to create the fields and settings described in the following procedure.

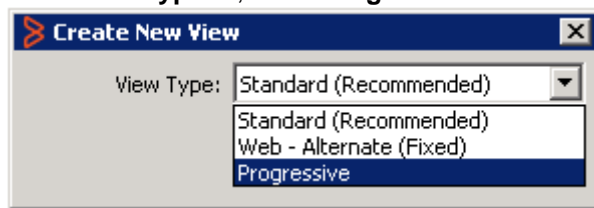
To create a dynamic selection list for progressive web applications by using JSON data

1. In Developer Studio, create a New Regular Form.

Create a Progressive View.

Steps (2)

1. Select **Form > Create New View**.
The Create New View dialog box opens.
2. In the **View Type** list, select **Progressive**.



3. Drag four character fields into the view.

In this example, name the fields **JSON**, **Json Exp**, **Output**, and **Character Field**.

JSON

Json Exp

Output

Character Field

4. In the Properties pane, change the **Display Type** property of the character fields to **Drop-Down List**.
5. Select the **JSON** field, and in the **Properties** tab, change the **Database > Input Length** to **0**.

Setting **Input Length** at a value of **0** enables the field to store content of any length.

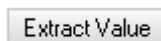
In real-world scenarios, the **JSON** field receives the JSON data automatically by means of a workflow or a webhook that you have already defined. Also, the field may be hidden from the Progressive View because it simply holds the JSON data. In this example, we manually paste JSON data into the **JSON** field in Progressive View.

Select **Character Field** from the view and in the **Properties** tab, add the following values under **Attributes**:

Property	Value
JSON Menu Label	#colors[*].color##colors[*].category
JSON Menu Value	colors[*].id
JSON Field ID	The field ID of the JSON field

The **Character Field** displays menu labels of the format color and category.

7. In the **Menu Style** list, select **Append**.
This option displays the **Character Field** in Progressive View as a list where you can select multiple options.
8. Define a button that runs the active links that you defined for the **Output** field.



Select the **Extract Value** button field in the view and add an active link with the following settings:

Setting	Value, selection, or function
If Actions	Set Fields
Data Source	CURRENT SCREEN

Field	Output
Value	JSONMAP(\$JSON\$, \$Json Exp\$)

This field uses the JSONMAP function, which picks up the data from the **JSON** field and the expression that you enter in the **Json Exp** field, and displays it in the **Output** field.

Select the **Character Field** from the view and define an active link with the following settings:

Setting	Value, selection, or function
Execution Options > Field	Character Field
Menu Choice	Selected
If Actions	Set Fields
Data Source	CURRENT SCREEN
Field	Output
Value	JSONMAP(\$JSON\$, \$Json Exp\$)

This active link runs when you select any item in the **Character Field** list in Progressive View, and displays the output of the JSONMAP function in the **Output** field.

11. Save the changes.

Validating the selection list in Progressive View

1. In a browser, log in to PWA.
<serverName>:8080/arsys/pwa/#/login
2. Change the URL to include the Progressive View form that you created in Developer Studio.
<serverName>:8080/arsys/pwa/#/forms/<serverName>/<formName>/<viewName>

Paste the following example JSON to the **JSON** field.

Example JSON

```
"colors": [
{
  "id":1,
  "color":"black",
  "category":"hue",
  "type":"primary",
  "code":{
    "rgba": [255,255,255,1],
    "hex":"#000"
  }
},
{
  "id":2,
  "color":"white",
  "category":"value",
  "code":{
    "rgba": [0,0,0,1],
```

```

"hex": "#FFF"
},
{
  "id": 3,
  "color": "red",
  "category": "hue",
  "type": "primary",
  "code": {
    "rgba": [255, 0, 0, 1],
    "hex": "#FF0"
  }
},
{
  "id": 4,
  "color": "blue",
  "category": "hue",
  "type": "primary",
  "code": {
    "rgba": [0, 0, 255, 1],
    "hex": "#00F"
  }
},
{
  "id": 5,
  "color": "yellow",
  "category": "hue",
  "type": "primary",
  "code": {
    "rgba": [255, 255, 0, 1],
    "hex": "#FF0"
  }
},
{
  "id": 6,
  "color": "green",
  "category": "hue",
  "type": "secondary",
  "code": {
    "rgba": [0, 255, 0, 1],
    "hex": "#0F0"
  }
}
]
}

```

As soon as you paste the example JSON, the **Character Field** displays the list of colors and the categories.

Character Field

Select ▼

black hue

white value

red hue

blue hue

yellow hue

green hue

4. To the **Json Exp** field, add the following expression:

colors[id=1].color

5. Click **Extract Value**.

The active link uses the expression in the **Json Exp** field and displays **black** in the **Output** field. The value **black** is associated with id=1 in the example JSON data.

JSON

```
{"colors": [{"id": 1, "color": "black", "category": "hue", "type": "primary", "code": {
```

Json Exp

colors[id=1].color

Output

black

Character Field

Select

Extract Value

(Optional) If you want to validate additional changes you made in Developer Studio, you must sync the cache in Mid Tier.

1. Log in to Developer Studio configuration.

<serverName>:8080/arsys/shared/config/config.jsp

2. Click **Sync Cache**.

3. Refresh the browser to display the updated Progressive View form.

Results

The **JSON Menu Label**, **JSON Menu Value**, and **JSON Field ID** properties in a character field extract the required data from the JSON data and display that information in a list. The JSONMAP function extracts specific values that you want from the JSON data and displays it in the associated field when you run the active link.

Related topics

[Creating-fields-that-auto-complete-in-Progressive-Web-Applications](#)

[Creating-selection-lists-in-Progressive-Web-Applications](#)

[Creating-list-tables-and-setting-selection-mode-by-using-run-processes-in-Progressive-Web-ApplicationsJSON examples that you can use](#)
[Online JSON validator](#)

File:

Developing-Pwa\Creating-Fields-That-Auto-Complete\Creating-

For forms viewed in a browser, you can turn on an "automatic complete" feature for character fields that have a menu attached to them. When this feature is configured for a field and a user begins typing text into the field, a list of values that match the text appears. When multiple values are presented, the user can change the selection with the up and down arrow keys or using the mouse. To select a highlighted option, the user presses Enter or click it with a mouse.

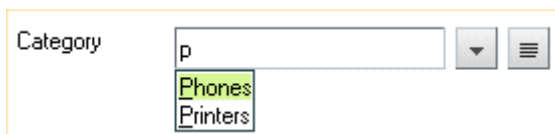
Related topics

[Adding-mention-for-fields-in-Progressive-Web-Applications](#)

[Creating-dynamic-selection-lists-for-Progressive-Web-Applications-by-using-JSON-data](#)

[Creating-selection-lists-in-Progressive-Web-Applications](#)

Standard view



A screenshot of a web form labeled "Category". It features a text input field containing the letter "p". To the right of the input field are two buttons: a downward-pointing arrow and a menu icon (three horizontal lines). A dropdown menu is open below the input field, displaying two options: "Phones" (which is highlighted with a green background) and "Printers".

Progressive view

Customer (required)

mary

+



Mary Carney
mcarney@phylum.com

Phylum



Mary Mann
M.Mann@calbroservice.com

Calbro Service



Mary Mann
mary.mann@petramco.com

Petramco



Mary Morstan
mmorstan@petramco.com

Petramco

Attach Files

By default, the suggestion list is case-insensitive, and it displays the *value* of the option, not the label. (You can change the configuration so that the menu's *labels* are displayed instead of its values.)

The results in the suggestion list can differ among databases. For example, for an Oracle database, the results are case sensitive by default. For other databases, the results are case insensitive.

The Menu/Row/Level Choice workflow condition runs when auto complete is used to fill in the value. Remember these tips when configuring workflow with character fields that have the automatic complete feature turned on:

- If workflow manipulates *values* of menu selections, you might want to configure auto-complete to show labels rather than values so that the workflow runs properly.
- The Return/Table or Level Dbl-Click workflow condition does not run when the user presses Enter as part of the Auto Complete feature (that is, when the selection list is displayed). At other times, the workflow condition runs normally.

For more information, see [Defining-workflow-to-automate-processes](#).

The following video (0:40) provides an overview of using the auto-complete or typeahead feature.

■ Watch Video: <https://www.youtube.com/watch?v=qU0O9nIJJ5M>

▶ <https://youtu.be/qU0O9nIJJ5M>

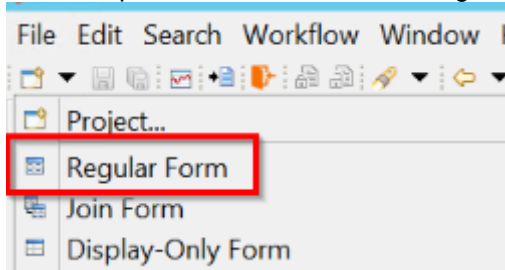
The following video (5:41) shows how to create typeahead fields and explore various options available to customize them.

■ Watch Video: <https://www.youtube.com/watch?v=hhvyBy99Ujg>

▶ <https://youtu.be/hhvyBy99Ujg>

To enable or disable auto-complete

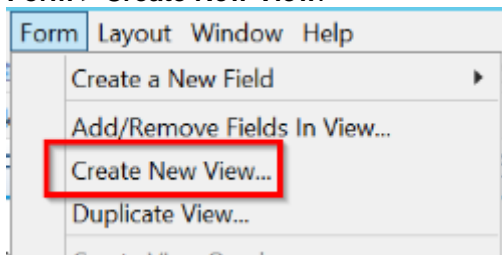
1. In Developer Studio, create a New Regular Form.



Create a Progressive View.

Steps (2)

1. **Form > Create New View.**



The Create New View dialog box opens.

2. In the **View Type** list, select **Progressive**.



3. Create a character field in the view.
4. Attach a menu to it.
5. Select the character field.

In the **Properties** tab, set the following properties:

Property

Description

Auto Complete

Determines what types of matches are displayed.
The options are:

- **None**(the default)—Disables auto-complete.
- **Leading Match**—Lists menu options that begin with the text that the user entered in the field.
- **Anywhere Match**—Lists menu options that include any text that the user entered in the field.
- This property is used only for character fields with *asearchmenu* attached. The property is ignored for all other fields.
- The number of characters the user must type to trigger auto complete. For example, if you enter 3, auto complete displays the list of matching values after the users enters 3 or more characters. The default is 0.
- Use the Auto Complete After Keystrokes property to set the typeahead value for search menus in Progressive Views.
- Choose the value carefully to get efficient results as this can have performance implications. If menu is expected to return huge amount of data, set this value to a higher number such as 3 or more.
- If you have enabled the Auto Complete property and set the **Auto Complete After Keystrokes** to zero (0), the dropdown will appear as soon as you click or tab to focus on the character field.

Auto Complete After Keystrokes

Determines whether the menu box is displayed (False) or hidden (True). The default is False. For large data sets, set this option to True so that users cannot open a menu. Instead, the menu data is used for auto-completion purposes only.
Determines whether menu *values* or *labels* are displayed. The options are:

Auto Complete Hide Menu Button

Auto Complete Match By

- **Value**(the default)—Menu values are displayed and used for matching while typing, as well as for completion.
- **Label**—Menu labels are displayed and used for matching while typing, but the menu value is used for completion. (This option is not normally used unless workflow on menu choice operates on the value, but you want to show the label to the user.)

At runtime, users can disable auto-complete by pressing Escape while typing in a character field. Auto-complete remains canceled until the field loses focus; it is re-enabled the next time the field gets focus.

To configure auto-complete to match by label

1. Follow the steps in [To enable or disable auto-complete](#) to enable Auto Complete for a character field.
2. Select the character field.

In the **Properties** tab, click **Auto Complete Match By**, and select one of the following values:

- **Value**(default)—Menu values are displayed and used for matching while typing, as well as for completion.
- **Label**—Menu labels are displayed and used for matching while typing, but the menu value is used for completion. (This option is not normally used unless workflow on menu choice operates on the value but you want to show the label to the user.)

Considerations for search menus with large data sets

Fields with search menus that represent a large data set (for example, over 10,000 entries) can cause performance issues and might not show the complete set of matches if the query limit is exceeded. Additionally, if a menu on the character field has thousands of entries, you might not want to display the menu with the field. To avoid these issues, remember these tips:

- Make sure that the menu is a *search* menu whose values are character fields in the target data source. Set the Refresh option for the menu to On Open.
- Because the data set is large, hide the menu button for the character field. (Set the character field's Auto Complete Hide Menu Button property to True.) This ensures that the menu data is used only for auto-completion purposes.
- Optionally, enable a limit on the number of characters required to be typed before matching begins. (Enter an integer in the character field's Auto Complete After Keystrokes property.)

Workflow considerations

Remember these tips when configuring workflow with character fields that have the automatic complete feature turned on:

- The Menu/Row/Level Choice workflow condition executes when auto complete is used to fill in the value.
- If workflow manipulates *values* of menu selections, you might want to configure auto-complete to show labels rather than values so that the workflow runs properly.
- The Return/Table or Level DbI-Clk workflow condition does not run when the user presses **Enter** as part of the Auto Complete feature (that is, when the selection list is displayed). At other times, the workflow condition executes normally.

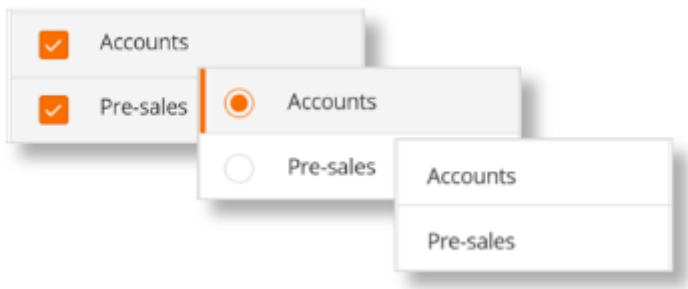
For more information about workflow conditions, see [Defining-workflow-to-automate-processes](#).

File:

Developing-Pwa\Creating-List-Tables-And-Setting-Selection-Mode

As a developer, you can create a table that can switch between selection modes. You can design a workflow that returns values which in turn are used by the `PERFORM-ACTION-TABLE-ROW-SELECTION-MODE` command to switch table lists between multiple selection, single selection, or view-only.

The following graphic shows the three selection modes that you can create for selection lists:



Related topics

[Using-Run-Process-and-PROCESS-commands](#)

[Creating-table-fields](#)

Before you begin

- You must have a basic understanding of progressive web applications and their development. For more information, see [Developing-Progressive-Web-Applications](#).
- You must create forms that contain the information that you want to display in the table. In the following procedure, we map this form to the list table we create.
- You may have to define workflows that return values that can be used by the `PERFORM-ACTION-TABLE-ROW-SELECTION-MODE` command.

To create a list table and set the selection mode by using a run process in a Progressive Web Application

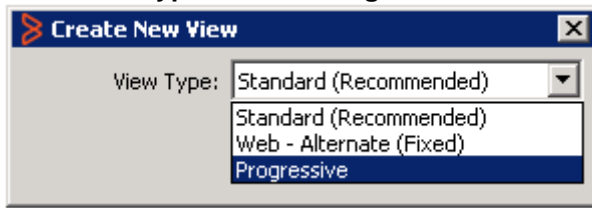
Step 1: To create a progressive view with a table showing information


1. In Developer Studio, create a regular form.

Create a Progressive View.

Steps (2)

1. **Form > Create New View.**
The Create New View dialog box opens.
2. In the **View Type** list, select **Progressive**.



3. From the Palette, drag and drop **Table - List View** into the view.
4. In the Properties tab, at **Attributes > Tree/Table Property**, in the **Value** column, click the ellipsis () button.
The Tree/Table Property dialog opens.
5. Search for a form name and map the Table Columns to fields from a remote form.
For example if the table needs to show a list of departments and geographical locations, you can accordingly add remote forms with this information.
6. From the Palette, drag and drop **Button** into the view.
You can rename the button to indicate its function. In this example, rename it to **Multiple to Single Select**.
7. Save the form.

Step 2: To create an active link and add the Run Process command

1. Select the button and create a new active link.
Select **New > Active Link**.
2. Select a server in the Select Server dialog box.
The Untitled Active Link screen appears.
3. In the Associated Forms section, click **Add**.
The Form Selector dialog box opens.
4. Select the form in the Progressive View which has the table and the button.
5. In **Active Link > Execution Options > Button/MenuField**, select the button that triggers the active link.
6. Right-click the **Actionpanel** header.
7. Select **Add Action > Run Process**.

In the **Command Line** field, enter the following command:

PERFORM-ACTION-TABLE-ROW-SELECTION-MODE *selectionMode* *tableFieldID*

Enter the following values of *selectionMode* to make a corresponding switch in selection modes of the lists:

Value of <i>selectionMode</i>	Selection mode of list switches from
0	Any selection mode to multiple selection mode
1	Any selection mode to view-only
2	Any selection mode to single selection

In place of **tableFieldID**, you can also add **tableField\$** or **tableName\$**, where **tableField** and **tableName** are names of the table field table. The value of **selectionMode** can be passed by another field based on how you design active links but the values must be one among **0,2**, or **1**.

9. Save the active link.

Results

When you display this form in progressive views, you can see a table list and a button that allows you to change the form, for example, multiple selection to a single selection, if you add **2** as the value of **selectionMode** in the command.

<div><div><input type="checkbox"/></div><div>Sales</div></div> <div><div><input checked="" type="checkbox"/></div><div>Accounts</div></div> <div><div><input checked="" type="checkbox"/></div><div>Pre-sales</div></div> <div><div><input type="checkbox"/></div><div>Human resources</div></div> <div><div><input type="checkbox"/></div><div>Facilities</div></div> <div><div><input type="checkbox"/></div><div>Operations</div></div> <div><div><input type="checkbox"/></div><div>Purchase</div></div>	<div><div><input type="radio"/></div><div>Sales</div></div> <div><div><input checked="" type="radio"/></div><div>Accounts</div></div> <div><div><input type="radio"/></div><div>Pre-sales</div></div> <div><div><input type="radio"/></div><div>Human resources</div></div> <div><div><input type="radio"/></div><div>Facilities</div></div> <div><div><input type="radio"/></div><div>Operations</div></div> <div><div><input type="radio"/></div><div>Purchase</div></div>	<div><div>Sales</div></div> <div><div>Accounts</div></div> <div><div>Pre-sales</div></div> <div><div>Human resources</div></div> <div><div>Facilities</div></div> <div><div>Operations</div></div> <div><div>Purchase</div></div>
Multiple selection	Single selection	View only

File:

Developing-Pwa\Creating-Resizable-Panes\Creating-Resizable-

As a developer, you can create panes that application users can resize according to their preferences. When you add resizable panes, users can expand the panes that have large amounts of information, thereby improving readability and user experience.

Related topic

[Flow-panel-holders-for-Progressive-Web-Applications](#)

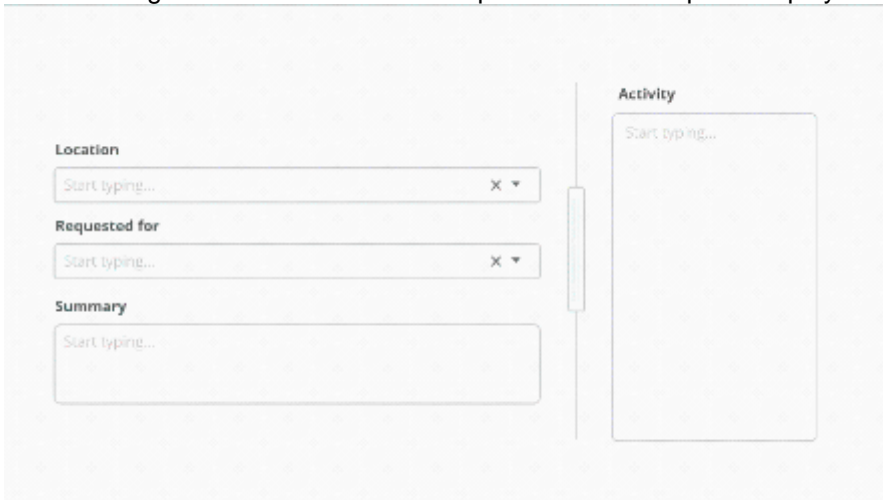
[Panel-holder-display-types-in-Progressive-View](#)

Scenario

At Apex Global Inc., the change request screen has an activity pane where the volume of content is sometimes large. Due to the fixed width of the activity pane, users need to scroll the pane to view all the content. Making the pane width adjustable could improve the user experience.

To enable users to resize the activity pane, the developer Seth creates a resizable pane by using Developer Studio. Users can increase the width of the activity pane when they want to view more content.

The following animation shows an example of a resizable pane displayed on a PWA screen:



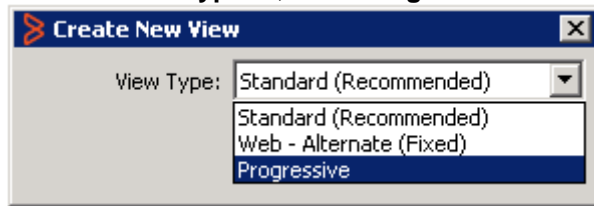
To create resizable panes in Progressive Web Applications

In Developer Studio, create a progressive view.

1. Select **Form > Create New View**.

The Create New View dialog box is displayed.

2. From the **View Typelist**, select **Progressive**.



Developer Studio automatically creates a section panel.

2. From the Palette pane, drag a Splitter Panel Holder into the view.



Set a value for the property **Display > Splitter Min Width Percent**.

This is the minimum width to which users can resize the pane.

The Default value of **Splitter Min Width Percent** is 20.

4. Click **Save**.
5. Open the Mid Tier configuration tool and sync the cache.

`http://<MidtierHostName>:<PortNumber>/arsys/shared/config/config_cache.jsp`

To test and review the resizable panes in Progressive Web Applications

1. In a browser, log in to PWA.
(On-premises) **`http://<midtierHostName>:<portNumber>/arsys/pwa/#/login`**
(SaaS) **`http://<helixBaseURL>/arsys/pwa/#/login`**
2. Change the URL to include the Progressive View form that you created in Developer Studio.
(On-premises) **`http://<midtierHostName>:<portNumber>/arsys/pwa/#/forms/<AR_Server_Name>/<formName>/`**
(SaaS) **`http://<helixBaseURL>/arsys/pwa/#/forms/<AR_Server_Name>/<formName>/`**
3. Confirm whether the resizable pane can be set to the minimum width.
The width of the pane you set is saved across sessions, that is, when you log in again, you will see the previously set width of the pane.

File:

Developing-Pwa\Creating-Search-Menus-With-Better-Readability

As an application developer, you can create search menus for Progressive Web Applications (PWA). These search menus improve the user experience by providing suggestions that help application users with populating fields.

search menus with display property set to Menu Section present diverse information with better formatting compared to search menus set to Section Panel. Search menus displaying multiple types of information with better formatting have improved readability and easier navigation.

Related topics

[UI-layout-of-Progressive-Web-Applications](#)

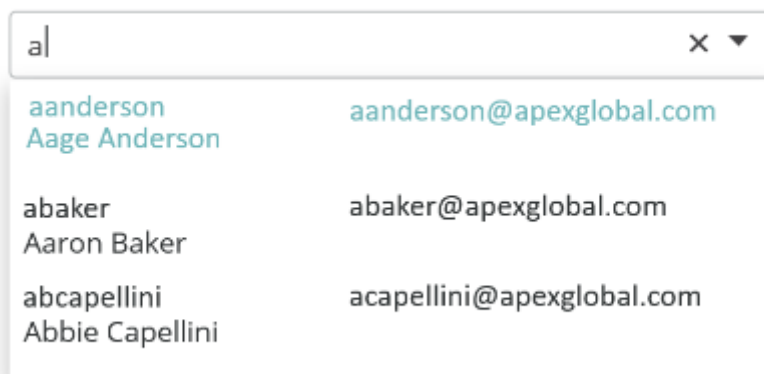
[Flow-panel-holders-for-Progressive-Web-Applications](#)

[Creating-search-menus](#)

Scenario

At Apex Global, Seth the developer wants to create an **Assigned to** field that enables application users to assign a ticket to an employee in Apex Global. Seth has found that application users search by using either the username, employee name, or email ID. To improve the search experience, Seth wants to create search suggestions in the **Assigned to** field which shows the username, employee name, and email ID, so that application users find it easier to search for the appropriate employee.

Assigned to (required)



aanderson	aanderson@apexglobal.com
Aage Anderson	
abaker	abaker@apexglobal.com
Aaron Baker	
acapellini	acapellini@apexglobal.com
Abbie Capellini	

Before you begin

You must have a form that has the data required by the search menu with display property set to Menu Section .

To create smart menus in Progressive Web Applications

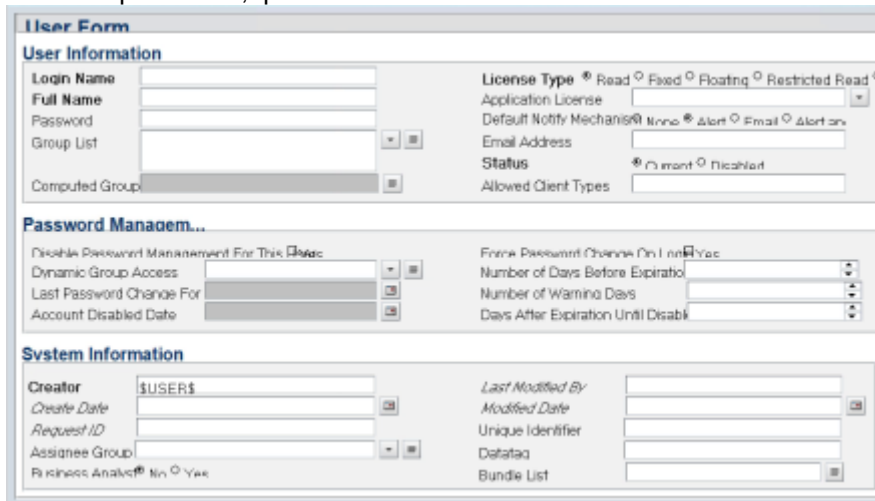
The following video (3:28) and procedure describe the example of creating a search menu with display property set to Menu Section by using data from the User Form.

■ Watch Video: <https://www.youtube.com/watch?v=1QHi28coXqw>

▶ <https://www.youtube.com/watch?v=1QHi28coXqw>

To create the panels with fields

1. In Developer Studio, open the User Form.



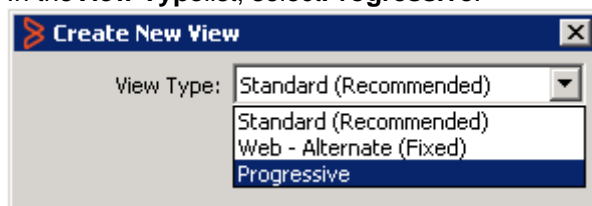
Create a Progressive View.

Steps (2)

1. Select **Form > Create New View**.

The Create New View dialog box opens.

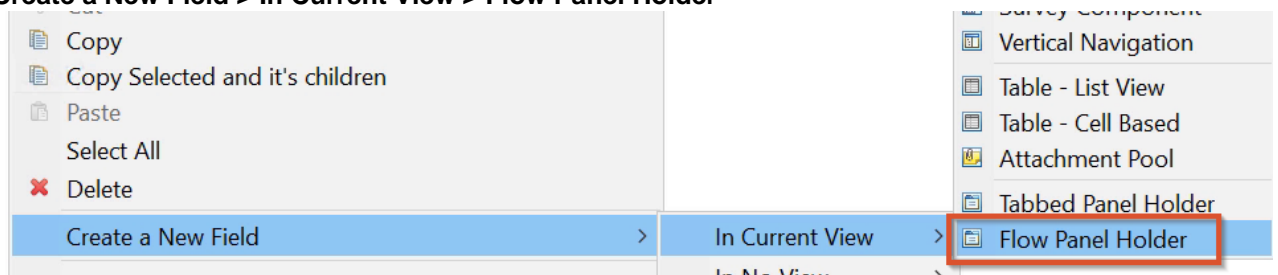
2. In the **View Type** list, select **Progressive**.



Developer Studio automatically creates a Section Panel.

2. Right-click within the section panel and create a flow panel holder in the current view.

Create a New Field > In Current View > Flow Panel Holder

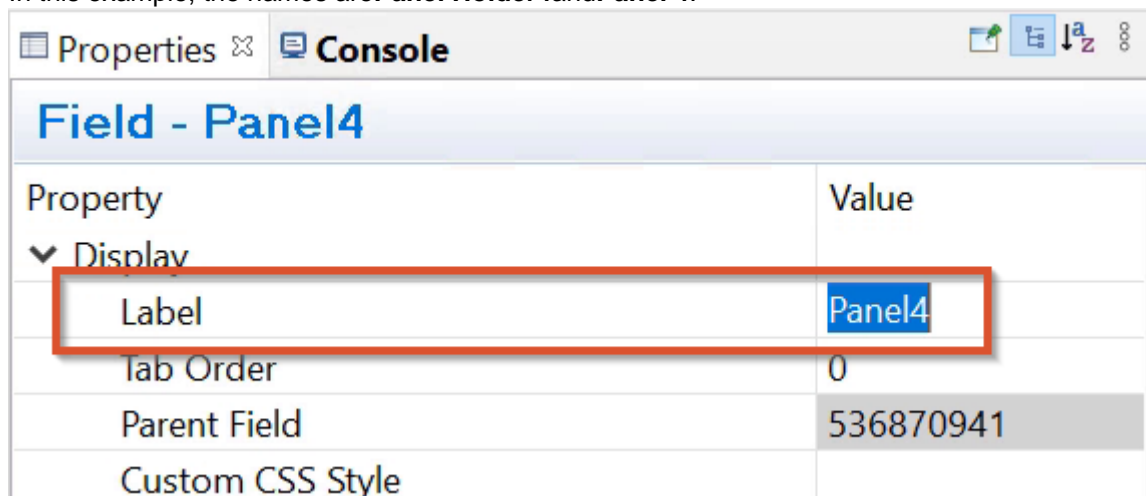


Adjust the panel size to allow the addition of more panels. You can see in the Outline pane that a new

panel holder and panel are created.

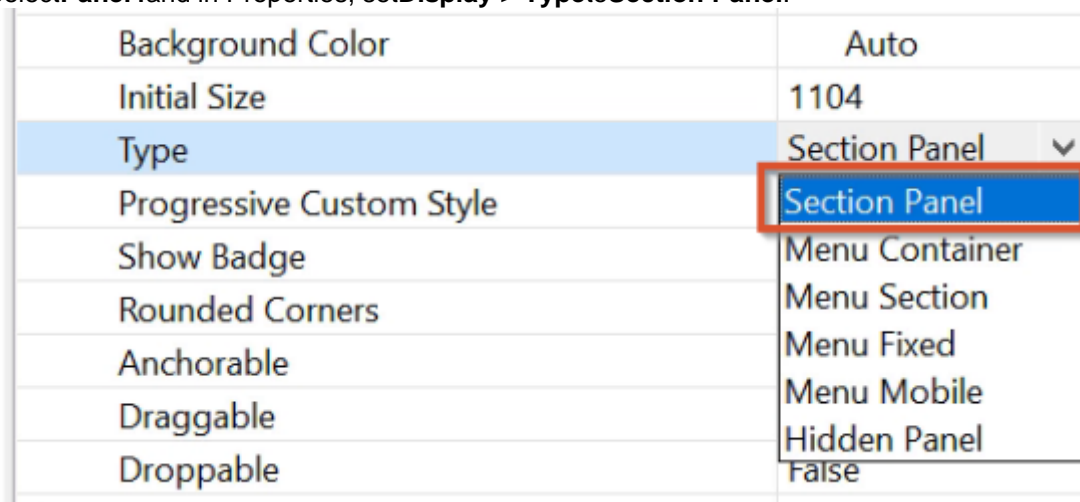
3. Name the new panel holder and panel.

In this example, the names are **Panel Holder4** and **Panel 4**.



Property	Value
▼ Display	
Label	Panel4
Tab Order	0
Parent Field	536870941
Custom CSS Style	

4. Select **Panel4** and in Properties, set **Display > Type** to **Section Panel**.



Background Color	Auto
Initial Size	1104
Type	Section Panel ▼
Progressive Custom Style	Section Panel
Show Badge	Menu Container
Rounded Corners	Menu Section
Anchorable	Menu Fixed
Draggable	Menu Mobile
Droppable	Hidden Panel
	False

1. Inside **Panel4**, create another flow panel holder.

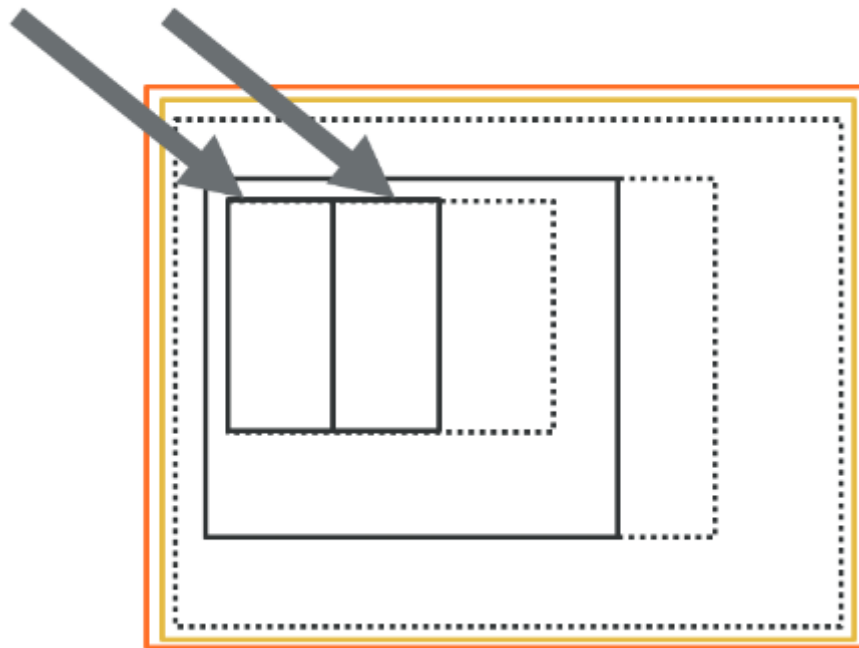
1. Name the panel holder and panel.

For example, **Panel Holder5** and **Panel5**.

2. Create another panel within **PanelHolder5**.

You can see the two new panels within the panel.

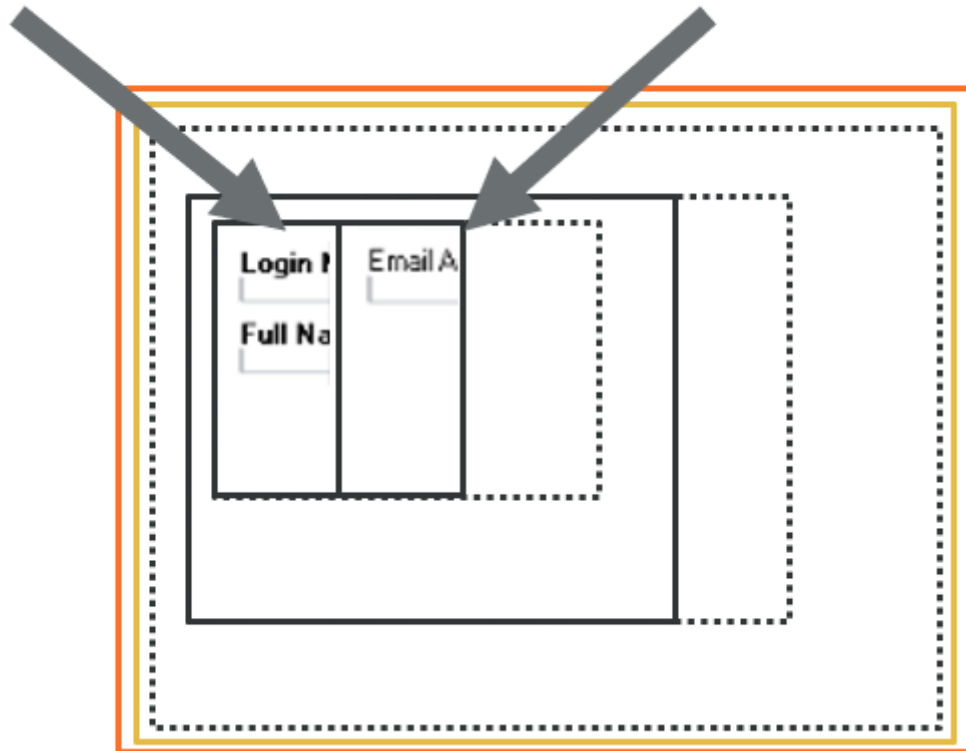
Panels



1. Select the two newly created panels.
2. In Properties, set **Progressive Desktop > Progressive Width**.
In this example, the value is set to **6/12**. These two panels occupy half the screen according to the 12-column layout. You can set **Progressive Width** to any other value based on how you want to
3. Add the **Login Name**, **Full Name**, and **Email Address** fields by navigating to **Form > Add/Remove Fields in View**.
These are the fields that are displayed by the search menu with display property set to Menu Section.

**Full Name and
Login Name**

Email



Select the two panels containing the **Login Name**, **Full Name**, and **Email Address** fields and in Properties, set **Display>Type** to **Menu Section**.

Best practice

When creating search menus with display property set to Menu Section, include only the following fields or objects within the view:

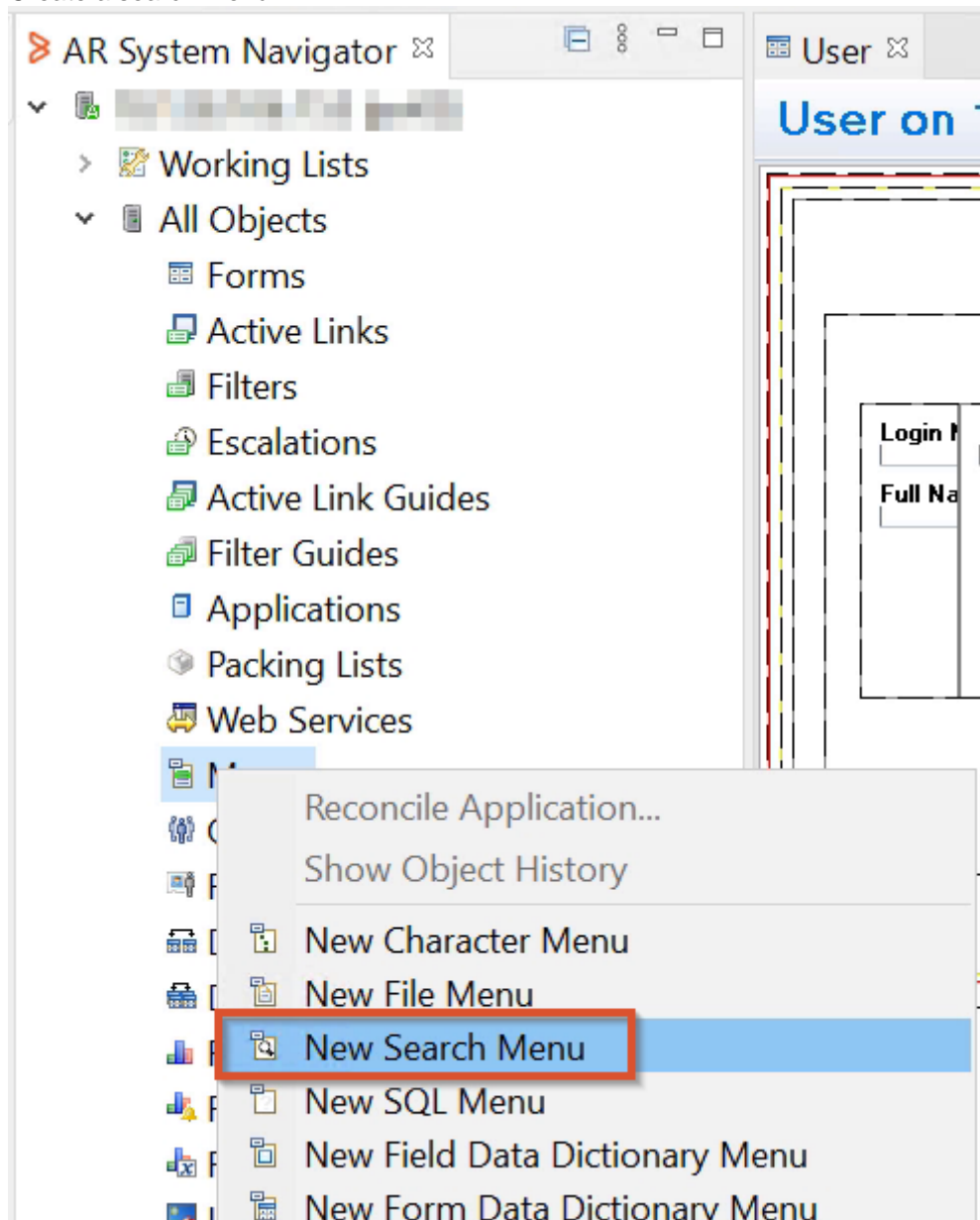
- Character
- Trim Text
- Check Box
- Radio Button
- Date
- Panel
- Drop-Down List

2. Change the **Label** and **Name** property of the view to **sample**.

▼ Basic	
Master View for Server Processing	False
Label	sample
Name	sample
View Type	Progressive
View ID	536870940
Locale	

To create the search menu

1. Create a search menu.



2. In the search menu, in the **Form Name** field, type **User**, and in the **View Name** field, type **sample**.
The **View Name** must refer to the view that we just created and named **sample**.
3. Under **Label fields**, add the three fields **Email Address**, **Full Name**, and **Login Name**.
4. In **Value Field**, type **Login Name**.
5. In **Qualification**, type **1=1**.
6. Save the menu as **sample_menu**.
7. Create a form in Progressive View with a character field.
Name the form and the field.
8. Select the character fields and change the **Menu Name** property to **sample_menu**.
9. Set the **Chunk Size** property to 5.
Chunking helps with the performance of the search menu with display property set to Menu Section.
10. Save the changes.
11. Open the Mid Tier configuration tool and sync the cache.
http://<MidtierHostName>:<PortNumber>/arsys/shared/config/config_cache.jsp

To validate the search menu in Progressive Web Applications

1. In a browser, log in to PWA.
(On-premise) **http://<midtierHostName>:<portNumber>/arsys/pwa/#/login**
(helix) **http://<helixBaseURL>/arsys/pwa/#/login**
2. Change the URL to include the Progressive View form that you created in Developer Studio.

(On-premise) **http://<midtierHostName>:<portNumber>/arsys/pwa/#/forms/<AR_Server_Name>/<formName>/**

(helix) **http://<helixBaseURL>/arsys/pwa/#/forms/<AR_Server_Name>/<formName>/**
4. In the character field, start typing any name.
The search menu displays search suggestions consisting of multiple types of information in a well-formatted way.

Assigned to (required)

The screenshot shows a search input field with the letter 'a' entered. Below the field is a dropdown menu with three rows of suggestions. Each row contains a name and an email address. The first row is highlighted in light blue.

aanderson Aage Anderson	aanderson@apexglobal.com
abaker Aaron Baker	abaker@apexglobal.com
abcapellini Abbie Capellini	acapellini@apexglobal.com

Important

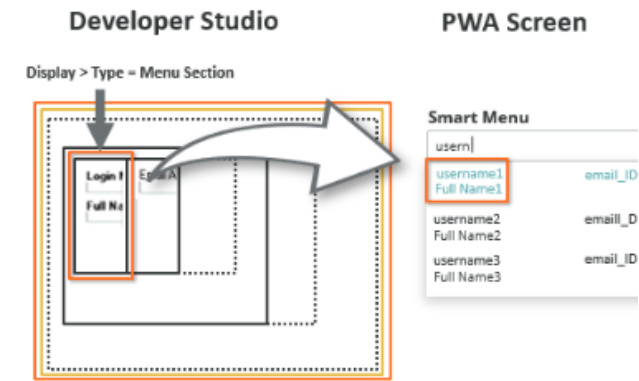
The following table helps you understand the differences between the Menu Section property and the Section Panel property so that you can decide the appropriate option:

Property

Behavior

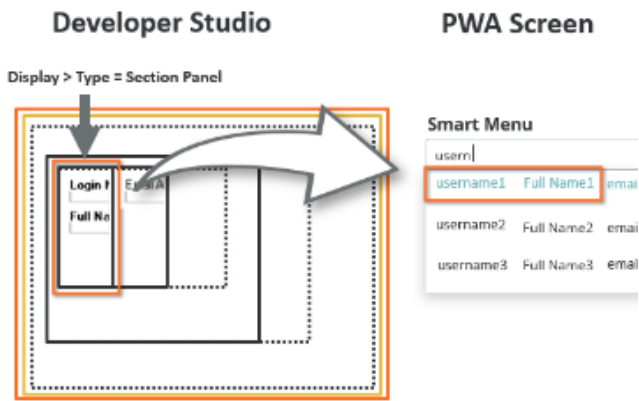
A panel with the **Display > Type** property set to **Menu Section** retains the relative positions of the fields that it encloses when displayed on the PWA screen. The following example illustration shows two fields within the menu section positioned one above another in Developer Studio are displayed one above another on PWA screens as well.

Menu Section



A panel with the **Display > Type** property set to **Section Panel** does not retain the relative positions of the fields it encloses when displayed on PWA screens. The following illustration shows an example of two fields arranged one above another in Developer Studio. However, on a PWA screen, they are displayed side-by-side.

Section Panel



File:

Developing-Pwa\Creating-Selection-Lists\Creating-Selection-Li

As an application developer, you can create a list where application users can select multiple options. Users can use this list to populate a field.

Example scenario: selection of multiple stakeholders

Seth, the developer wants service desk agents to add a number of stakeholders to the list of people who are notified about the status changes of a ticket.



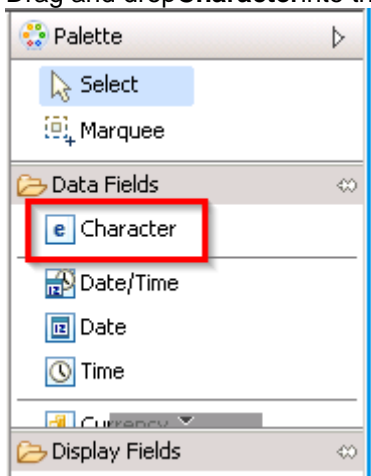
John.doe@calbro.com, John.smith@calbro.com, Joe.bloggs@calbro.com X ▼

- ☒ John.doe@calbro.com
- ☒ John.smith@calbro.com
- ☐ Jane.smith@calbro.com
- ☒ Joe.bloggs@calbro.com
- ☐ Joe.smith@calbro.com

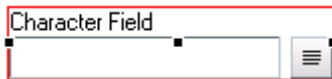
For this, Seth wants to create a selection field consisting of stakeholders listed with check boxes against each name so that the service desk agent can quickly select and add stakeholders to the CC list.


To create selection lists in Progressive Web Applications


1. InDeveloper Studio, create a Progressive View.
2. Drag and drop**Character**into the view.



3. Select the character field.



4. In the Properties tab, beside **Attributes > Menu Name**, click the ellipsis () button.
The Menu Name dialog opens.
5. In the **Name** field, enter the name of the form which contains the options you want to display in the list.
You must select a menu of category Search.

 Menu Name



6. In the **Menu Style** list, select **Append**.
If you select **Overwrite**, you get a list that allows selecting only one item.
7. In the **Display > Display Type** list, select **Drop-Down List**.
8. (Optional) Under **Display**, in the **Menu Delimiter** field, enter the character that should appear between the selected options when this data is saved.
For example, if you enter a slash (/) as the delimiter, you generate the following data within the application:
Option1/Option2/Option3
Irrespective of the delimiter you specify, application users see a comma (,) as the delimiter in the field.
9. Save the changes.

Validating the multiple selection list in PWA

1. In a browser, log in to PWA.
<serverName>:8080/arsys/pwa/#/login
2. Change the URL to include the Progressive View form that you created in Developer Studio.
<serverName>:8080/arsys/pwa/#/forms/<serverName>/<formName>/<viewName>
(Optional) If you want to display more changes you made in Developer Studio, you must sync the cache in Mid Tier.
 1. Log in to Mid Tier configuration.
<serverName>:8080/arsys/shared/config/config.jsp
 2. Click **Sync Cache**.
 3. Refresh the browser displaying the progressive view form.

Related topics

[Creating fields that auto-complete in Progressive Web Applications](#)

[Creating dynamic selection lists for Progressive Web Applications by using JSON data](#)

[Adding mention for fields in Progressive Web Applications](#)

File:

Developing-Pwa\Css-Classes-Used-In-Progressive-Web-App-S

The following table lists out the CSS classes that are being used in Progressive Web App (PWA) screens:

CSS Class Name	Description
mb-1	Sets margin-bottom to .25rem
mb-2	Sets margin-bottom to .5rem
mb-0	Removes margin-bottom
mt-3	Sets margin-top to 1rem
m-0	Sets margin to 0 across all sides
pr-2	Sets padding right to .5rem
pl-0	Sets padding-left to 0
border-0	Removes border from the element across all sides
w-100	Sets width to 100%
	Indicates whether to use relative position for vertical dropdown.
detach	Default behavior for dropdown shown in vertical navbar is absolute. If you want the position of the dropdown to be relative, use the CSS class detach. This class is specific to tab panels. Indicates how browser distributes space between and around content items along the main-axis of a flex container, and the inline axis of a grid container.
justify	
text-primary	Displays text color in blue
rev-ml-1rem	rev-ml-1rem=Sets margin-left to -0.25rem
rev-mr-1rem	rev-mr-1rem=Sets margin-right to -0.25rem

Related topics

[UI-layout-of-Progressive-Web-Applications](#)

[Adding-icons-and-images-to-Progressive-Web-Applications](#)

[Previews-of-icons-available-in-Developer-Studio-to-support-PWA](#)

File:

Developing-Pwa\Customization-Best-Practices-For-Developing

Consider the following best practices when developing or customizing Progressive Web Applications (PWA).

Adhering to these best practices helps you to achieve the following outcomes:

- Leverage the full potential of PWA features, such as responsiveness.
- Enhance the experience of your application users.
- Avoid errors when running workflows.

Whitepaper on PWA implementation methodology

Make sure that you read the whitepaper at the following link before referring to the best practices in this topic: [Community blog: Whitepaper on ITSM PWA Implementation Methodology](#)

The information in the whitepaper helps you achieve the following objectives:

- Understand PWA concepts
- Learn how Smart IT PWA screens are developed
- Perform customization and localization of PWA screens

Important

Before enabling Progressive Views, make sure that you have AR System server, Mid Tier, and Developer Studio version 20.08 or later installed.

Related topics

[Developing-Progressive-Web-Applications](#)

[UI-layout-of-Progressive-Web-Applications](#)

[Adding-rich-text-fields-in-Progressive-Web-Applications](#)

[Panel-holder-display-types-in-Progressive-View](#)

Use supported fields and properties in Progressive View

You must only use the fields supported by progressive views to build a Progressive Web Application. For each field on a form, the **Properties** tab in Developer Studio lists the properties of that field.

Fields and properties help you perform the following tasks:

- Create elements in your application interface.
- Set properties that determine the look and behavior of the field during operations performed in a browser.

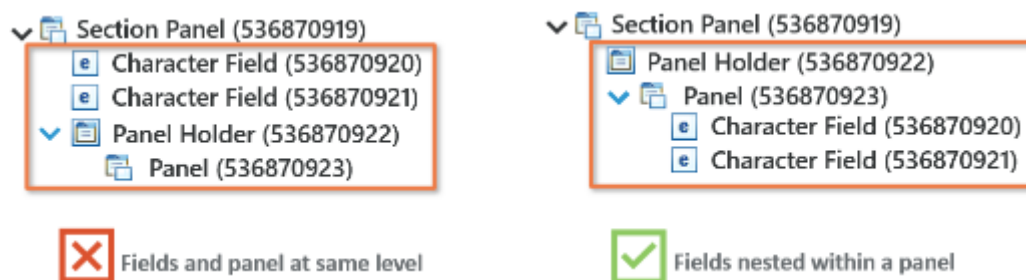
The properties listed in the **Properties** tab vary depending on the type of field you create or modify. For the list of supported fields and properties, see [Field properties](#).

Set permissions, visibility, and hierarchy for fields and sections

When creating fields and sections, you must adhere to the following best practices:

- Set section panel permissions for users to be able to view the section panel and its contents.
- In a panel, set the **Visible** property to **True** to enable users to see the contents of that panel.
- Share the same panels between multiple form views where required.
Display properties are associated with specific views and do not interfere with other form views, unlike other properties.
- Move temporary fields that don't need localization outside the form view while keeping them within the form.
Use the Add/Remove Fields in View dialog to move a field out of the view.
- Maintain panel and field hierarchy—Do not place fields and flow panels at the same level in the hierarchy.

The following illustration shows how the hierarchy should appear in Developer Studio:



The following video (1:14) demonstrates the best practices when creating fields and sections:

■ Watch Video: <https://www.youtube.com/watch?v=uKdaUBgwfJg>

▶ <https://youtu.be/4QzFHEaRwIM> <https://youtu.be/uKdaUBgwfJg>

Skip to specific parts of the video:

[0:06](#) Setting panel permissions for users

[0:24](#) Maintaining panel and field hierarchy

[0:41](#) Setting panel visibility

Best practices to create and use flow panel holders

A Flow panel holder is available when you enable a Progressive view for your form. For more information, see [Progressive View quick start guide](#). The Flow panel holder supports the properties required to enable a progressive view. Use a Flow Panel holder for aligning fields on a form. You can also group the fields to ensure correct alignment of fields on a form. This panel does not have a panel header, and supports

Container panel holder (Default), Sub Panel holder, and Section Panel holder types. The following video (4:52) demonstrates creating Fixed panels in Developer Studio when you enable progressive views for a form:

■ Watch Video: <https://www.youtube.com/watch?v=m33BWG-mEoI>

▶ <https://youtu.be/m33BWG-mEoI> The following video (13:03) demonstrates creating complex layouts with flow panels:

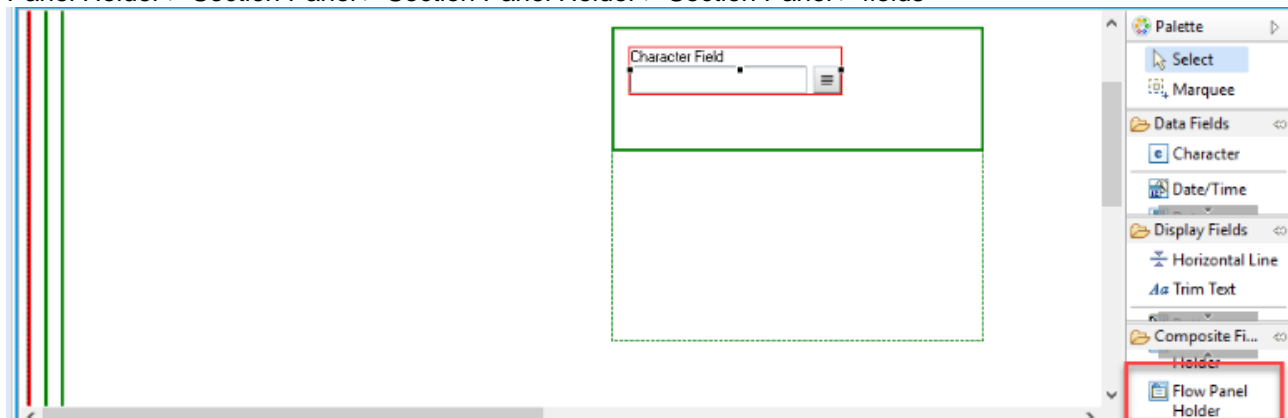
■ Watch Video: <https://www.youtube.com/watch?v=TnwUS2RF28Q>

▶ <https://youtu.be/TnwUS2RF28Q>

Panel arrangement sequence

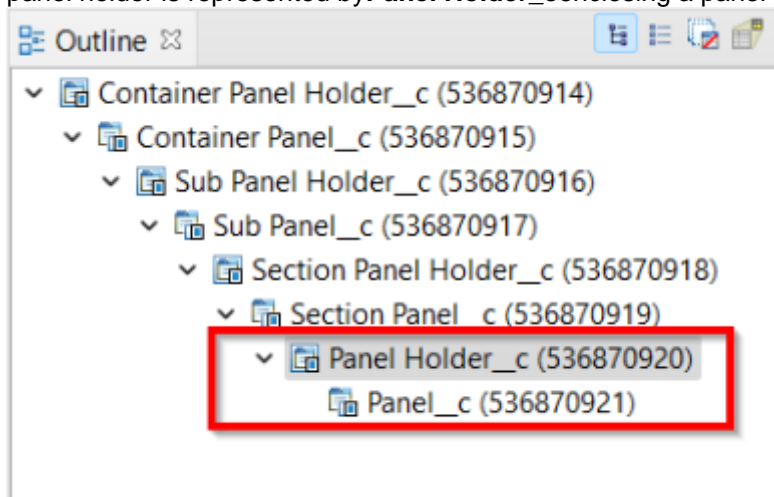
The panels are arranged in the following sequence or hierarchy:

Flow Panel holder > Container Panel Holder > Container Panel > Sub Panel Holder > Sub Panel > Section Panel Holder > Section Panel > Section Panel Holder > Section Panel > fields



- A progressive view can hold only a Container type Flow panel holder and a Fixed panel holder.
- By default, a Flow panel holder has one panel as a child panel.
- You can change the read-only property type of any panel only when you change the property type of Flow panel holder.

Tip When you add Flow Panel Holder to the Progressive View from the palette in Developer Studio, the flow panel holder is represented by a panel holder in the Outline pane. In the following example figure, the flow panel holder is represented by **Panel Holder_c** enclosing a panel named **Panel_c**:



Container Flow panel holder

- By default, a Container type Flow panel is added to a Flow panel holder.
- You can delete a Container Flow panel only if you remove all progressive views.
- A Container Flow panel holder can have one or more Sub panels.

Section Flow panel holder

- You can add a Section Flow panel holder only under a Sub panel or Section panel.
- A Section panel can either have a Section type Flow panel holder or non-panel fields.
- The Section Flow panel holder added by framework contains only Section panel as children.

Sub panel holder

The Sub panel holder behaves as a Section panel on the desktop client. However, it is displayed as a tab on a mobile client.

- You can add a sub panel holder only under the container panel.
- A sub panel holder can have one or more sub panels.

For more information about developing PWA, see [Developing-Progressive-Web-Applications](#).

▶ <https://youtu.be/m33BWG-mEoI> The following video (13:03) demonstrates creating complex layouts with flow panels:

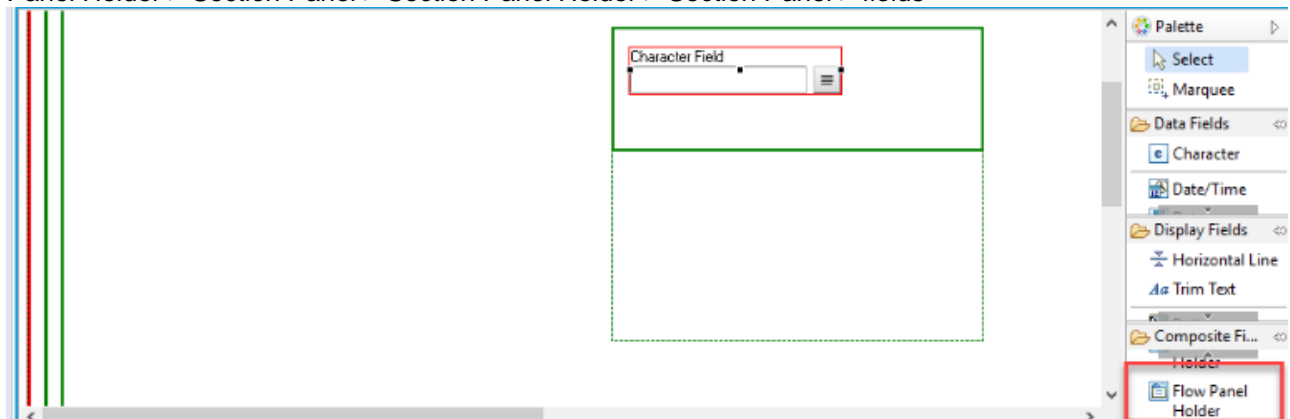
■ Watch Video: <https://www.youtube.com/watch?v=TnwUS2RF28Q>

▶ <https://youtu.be/TnwUS2RF28Q>

Panel arrangement sequence

The panels are arranged in the following sequence or hierarchy:

Flow Panel holder > Container Panel Holder > Container Panel > Sub Panel Holder > Sub Panel > Section Panel Holder > Section Panel > Section Panel Holder > Section Panel > fields



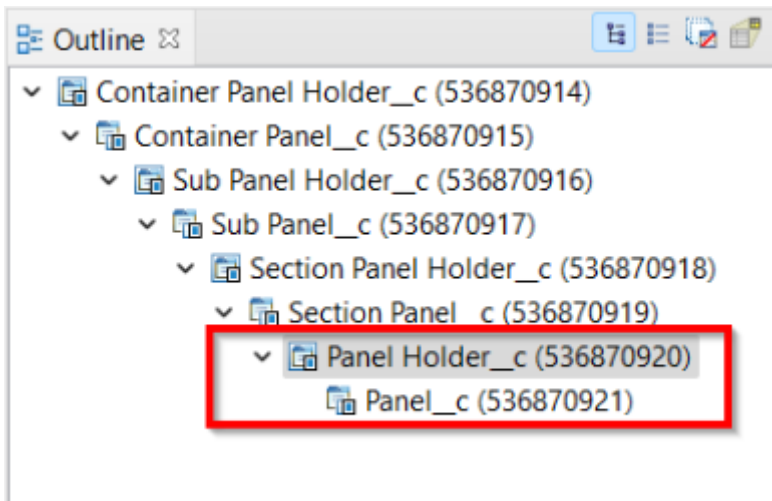
- A progressive view can hold only a Container type Flow panel holder and a Fixed panel holder.

- By default, a Flow panel holder has one panel as a child panel.
- You can change the read-only property type of any panel only when you change the property type of Flow panel holder.

Tip

When you add Flow Panel Holder to the Progressive View from the palette in Developer Studio, the flow panel holder is represented by a panel holder in the Outline pane.

In the following example figure, the flow panel holder is represented by **Panel Holder_c** enclosing a panel named **Panel_c**:



Container Flow panel holder

- By default, a Container type Flow panel is added to a Flow panel holder.
- You can delete a Container Flow panel only if you remove all progressive views.
- A Container Flow panel holder can have one or more Sub panels.

Section Flow panel holder

- You can add a Section Flow panel holder only under a Sub panel or Section panel.
- A Section panel can either have a Section type Flow panel holder or non-panel fields.
- The Section Flow panel holder added by framework contains only Section panel as children.

Sub panel holder

The Sub panel holder behaves as a Section panel on the desktop client. However, it is displayed as a tab on a mobile client.

- You can add a sub panel holder only under the container panel.
- A sub panel holder can have one or more sub panels.

For more information about developing PWA, see [Developing-Progressive-Web-Applications](#).

Best practices when localizing Progressive Web Application screens

Use the localization toolkit and the AR System Message Catalog form to localize your Progressive Web Application (PWA).

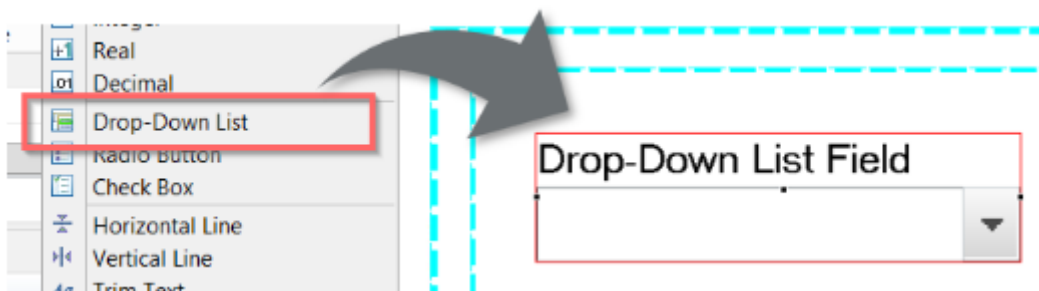
You must adhere to the following best practices when localizing a PWA:

- Use independent forms only so that the related Active Links and Filters are also considered.
- For the form fields that you want to localize, set the value of the "Localize Label" property in Display Properties to **True**.
- When you delete a package definition, the package definition is deleted, but the extracted strings are not deleted from the localization database. If you create a package definition with the same name as the deleted one and export, you get the extracted strings from the deleted package definition as well. To avoid this issue, we recommend that you explicitly delete those strings from the database.

For more information about localizing PWA, see [Localizing-a-Progressive-Web-Application](#).

Update associated workflows when altering Drop-Down List fields

As a developer, you can create a list for PWA users by using the Drop-Down List field. The following graphic shows an example Drop-Down List field:



The Drop-Down List field lists the options that you add by using the Selections dialog box. You must then define active links that refer to these selection values. You must make sure that you do not remove any of the selections shown in the following example graphic:

Selection Values	Alias	ID
Option 1	Option 1	0
Option 2	Option 2	1
Option 3	Option 3	2

Default Value:

ID Enumeration:

For example, if you delete any of the selections in the Selection dialog, the Progressive Web Application users see a blank field in place of the drop-down list. This is because the active link that opens the drop-down list refers to these three options. When one option is missing, the active link breaks the workflow and the field does not get displayed. If you must delete any of the selections, you must also make sure that the associated workflows referring to the selections are updated accordingly.

The following video (1:00) explains the best practice of making sure workflows are updated along with selection lists.

■ Watch Video: <https://www.youtube.com/watch?v=4QzFHEaRwIM>

▶ <https://youtu.be/4QzFHEaRwIM>

Check execution order when modifying workflows

When adding any additional steps in your workflows, you must do the following:

1. Get to know the execution order of all the other existing workflows by viewing the active link logs.
2. Add the additional workflow with the correct execution order depending on the outcome you want to achieve.

The following video (0:52) demonstrates how to view and set the execution order of workflows:

■ Watch Video: <https://www.youtube.com/watch?v=gaXqNLvCdBA>

▶ <https://youtu.be/4QzFHEaRwIMhttps://youtu.be/gaXqNLvCdBA>

Best practice to display activity note type in Smart IT by updating the activity template

When Smart IT PWA screen users create a new note, they can select a note type. But by default, the note type is not displayed in the activity pane. In this example procedure, we update the activity template in the AR System Resource Definitions form so that the Smart IT activity note information displays the activity note

type.

The following graphic shows a high-level overview of the steps you need to perform in this procedure.



The following video (2:01) shows how to display the activity note type to the activity note message in ITSM: Smart IT.

■ Watch Video: <https://www.youtube.com/watch?v=-npsSMfbvs>

▶ <https://youtu.be/-npsSMfbvs>

To update activity template to display activity note type in the Activity tab

Search and download the **SHR_SV_Activity_Template_Generic.html** template file.

1. Open the AR System object list page and AR System Resource Definitions form.

http://<MidtierHostName>:PortNumber>/arsys/forms/<AR_Server_Name>/AR+System+Mid+Tier+Object+List

2. In the **Name** field, enter **AR System Resource Definitions form**.

3. Click **Open Search** to open the form in search mode.

The **AR System Resource Definitions** form opens.

4. In the **AR System Resource Definitions** form, in the **Name** field, enter **SHR_SV_Activity_Template_Generic.html**.

Current mode: Search

Search [New search] [New request] [Modify all] Searches My

bmcsoftware

Request ID []

Name SHR_SV_Activity_Template_Generic.html

Status []

5. Click **Search**.

The search result shows **SHR_SV_Activity_Template_Generic.html**

6. In the Resource section, select **SHR_SV_Activity_Template_Generic.html**.

7. Click **Save to Disk**.

2. Open the **SHR_SV_Activity_Template_Generic.html** template file in a source code editor such as Notepad++.
3. Add the following variable in the template file:
varworkLogType= \$<field ID>\$
4. Replace the field ID with the field ID of the incident, which is 304427491.
The field ID is 304422931 for work orders. After this change, you have the following line of code:
varworkLogType=`\$304427491\$`;
5. In the template file, search for the following line of code:
**messageText = peopleText + valField_0 + ' ' + addedNoteStr + '
' + generateLinkForURL(valField_1)**
6. Replace it with the following line of code to add the note type variable to the activity notes message:
**messageText =peopleText+valField_0 + ' ' + addedNoteStr + ' (' + notetype + ') ' + '
' + generateLinkForURL(valField_1);**
7. Save the changes to the template file.
8. In the AR System Resource Definitions form, delete the existing template file (SHR_SV_Activity_Template_Generic.html) and add the file you just updated.
9. Open the Mid Tier configuration tool and flush the cache to update the changes.
10. OpenITSM: Smart ITand create a new note in the incident page.
You can see that the note type information is displayed in the activity note message.

Create custom activity messages

Create custom activity messages in Smart IT PWA screens. For example, you can change the activity messages to show more information on the changes that were made by a specific user.

The following video (3:16) show how you can change the activity message that is displayed in Smart IT PWA screens.

■ Watch Video: https://www.youtube.com/watch?v=GPqPunk1z_8

▶ https://youtu.be/GPqPunk1z_8

Identify missing active link guides during customization

When customizing the PWA, you might have to change workflows such as active links and filters. When updating workflows, make sure that the active link guides are not edited or removed in Developer Studio. If the custom message does not appear as expected, it is important to check the logs to see if the active link guides or any workflows are not working as expected.

The following video (1:03) shows you how to make sure the active link guides are working.

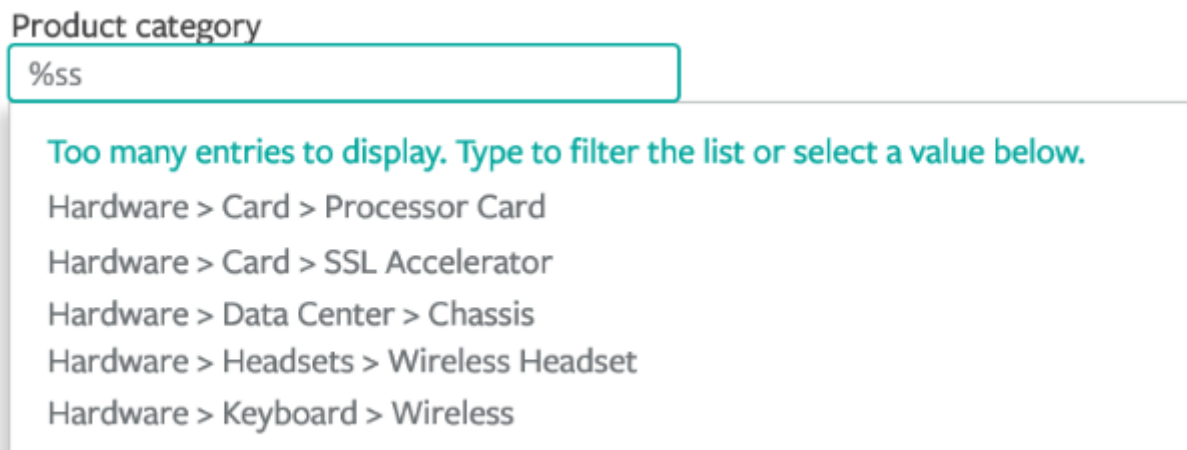
■ Watch Video: <https://www.youtube.com/watch?v=6De09pnOPqY>

▶ <https://youtu.be/6De09pnOPqY>

Change the chunk size in search fields for optimum results and performance

Search fields on PWA screens have a **Chunk Size** display property in Developer Studio that determines the maximum number of matching results that are displayed.

In the following example graphic, **Chunk Size** is set to **5** and hence not more than 5 matching results are displayed.



To view all the search suggestions, set the **Chunk Size** property for the field to zero.

Setting **Chunk Size** to zero might affect the performance of the application based on factors such as database size.

We recommend that you optimize the chunk size to view the required number of results while making sure performance is reasonable.

The following video (0:17) shows how to change the chunk size of a field in Developer Studio.

Display messages in prompt bar instead of pop-up window when loader is enabled

To display a message to the application user after an active link executes, use the Message section of the Active Link in Developer Studio to add the message and the message text.

When displaying a message through an active link, you must ensure the following best practices:

- To show the message with the loader enabled, you must select **Show Message in Prompt Bar** in the active link message settings.

Message configuration window showing:

- Type: Note
- Text: Email Sent test
- Number: 10000
- ☒ Show Message in Prompt Bar

For example, if you have created a message action that is followed by the `ENABLE-WAIT-CURSOR-ON-LONGRUNNING`

`PROCESS` command to display the loader, you must select **Show Message in Prompt Bar** to make sure the workflow is not interrupted. For more information about

the `ENABLE-WAIT-CURSOR-ON-LONGRUNNING-PROCESS` command, see [Process-commands](#).

- To show the message as a pop-up window, you must set the message to display before the `ENABLE-WAIT-CURSOR-ON-LONGRUNNING-PROCESS` command runs.

Text with ampersand (&) must have spaces to avoid matching with HTML elements

Plain text with an ampersand (&) that is not preceded and followed by spaces is processed as HTML. If you have any plain text containing ampersand (&), make sure it is preceded by and followed by a space.

For example, if you add a value such as **Test&litude**, the portion of the string **&is** assumed to be an ampersand (&), and that value is replaced by **Test&litude**.

Use active links to fetch labels of list options

In ITSM, when character fields are associated with a menu, the value of the label for the menu and the value returned by the menu are different. In such cases, you must use an active link workflow with **Menu Choices** selected as the **Execution Option** to set the label based on the value returned by the menu.

The following active link is an example of an active link which resets the field back to the label of the option selected by application users:

CHG:Infrastructure Change form, Field ID: 1000000881, z1D Status Reason field in Best Practice View, active link named CHG:SHR:SetStatusReason_100.

However in PWA, if the field is in enabled state, upon menu selection, the system automatically fetches the label and displays it in the field. In this scenario, there is no need to write an explicit workflow.

Important

If the field is in disabled or read-only state, you must create an active link that fetches the label of the selected option.

You can create an active link similar to the active link CHG:SHR:SetStatusReason_100 mentioned in the example.

Avoid selection fields when using autocomplete

When you create a search field in Developer Studio that has autocomplete, make sure you avoid adding fields with **Selection** as the data type. The following fields are of data type **Selection**:

- Drop-Down List
- Radio Button
- Check Box

If any of these fields of selection data type exist in the source forms that provide the data for the search field, the autocomplete may not work as expected.

Set Display Type to correctly display phone numbers or email IDs in character fields

When you display an email ID or a phone number in a character field on a PWA screen, the phone number or email ID may not display correctly when they are longer than usual. To avoid this, you must select the following display type properties for the character field:

information to display	Property to add to character field
Email ID	Display Type > Email ID
Phone number	Display Type > Phone Number

Close an existing loader before opening a pop-up window

When a pop-window is set to open when there is a loader running, as a developer, you must make sure that the loader closes before the pop-up window opens. If the loader is not stopped before a pop-up window appears, the loader could prevent access to the pop-up window.

The following animation shows an example of a loader:



To make sure that the loader stops before the pop-up window opens, you must perform the following high-level steps:

1. Identify the active link that opens the pop-up window.
2. Insert the following run process before the open window action:
`ENABLE-WAIT-CURSOR-ON-LONG-RUNNING-PROCESS 0`
This ensures that the loader closes before the pop-up stops.

3. (Optional) You may run the loader again after the open window action by using the following run process:
ENABLE-WAIT-CURSOR-ON-LONG-RUNNING-PROCESS 1

For more information about the ENABLE-WAIT-CURSOR-ON-LONG-RUNNING-PROCESS command, see [Process-commands](#).

Enable searching by support groups aliases

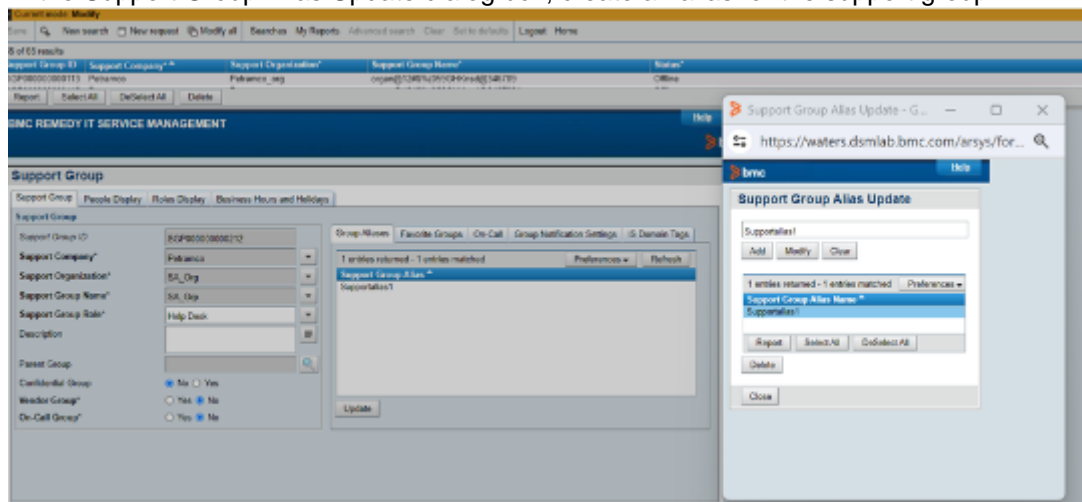
As an application developer, perform the following procedure to provide your application users with the ability to search by support group aliases on PWA screens.

Prerequisites

Create a support group alias for the support group.

1. Log into Mid Tier.
2. Open the Support Group window.
Application Administration Console > Support Group > Group Alias panel
3. In the **Support Group** tab, click **Update**.

In the Support Group Alias Update dialog box, create an alias for the support group.



For more information about support groups, see [Creating support groups](#).

To create a support group search on PWA screens

1. Log into Developer Studio.
2. Open the CTM:Support Group Alias Lookup form.
3. Create a Progressive View for the CTM:Support Group Alias Lookup form.

4. Create a search menu based on the CTM:Support Group Alias Lookup form.

TESTALIAS_HPDI:INC:SV_AssigneeGroup on

Refresh: On Open
Data Source: SERVER
Server:
Form Name: CTM:Support Group Alias Lookup
View Name: SV_SuppGrpAlias
Label Fields

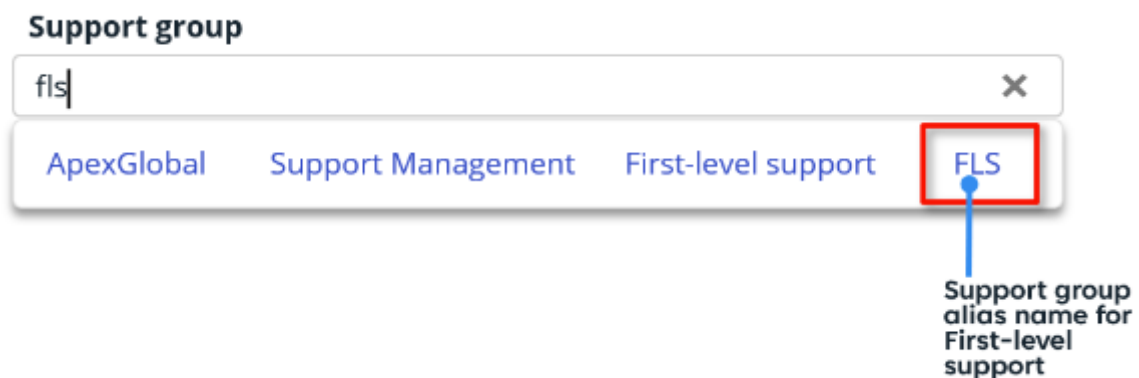
Name	Field ID	Data Type
Support Group Alias Name	10000000293	Character
Company	10000000001	Character
Support Organization	10000000014	Character
Support Group Name	10000000015	Character

☒ Sort On Label
Value Field: Support Group ID
Qualification
[('Company' = \$10000000032\$) OR ('Company' = \$10000000001\$) OR ('Company' = *-Global-*)] AND ('Status' = 'Enabled')]

For information about creating a search menu, see [Creating search menus](#).

5. Attach the smart menu to the custom or overlaid support group fields in Progressive View.
6. Validate these changes by opening a ticket on a PWA screen and searching by support group alias names.

The following graphic shows an example where searching by support group alias names is enabled.



Best practices for enhanced performance

Follow these best practices for better performance of PWA:

- Disable logs for both Mid Tier and AR System. Enable logs only when troubleshooting issues. For more information, see [Enabling logs for performance issues](#).
- When creating workflows with multiple calls, use parallel calls. This ensures that the time taken by the workflow to complete is limited to that of the longest call. If workflow calls are made one after another, then the workflow duration is the sum of the time taken for all the calls. For more information, see [Creating active links](#).
- Enable archiving for transactional data for all the modules. For more information, see [Archiving data](#).
- Write qualifications that lead to optimum SQL queries. For example, avoid a qualification that will lead to a full table scan.

- Ensure that you include correct and optimized DB indexes.

Additional resources

 [Webinar: Progressive Web Apps Introduction & Best Practices](#)

 [Webinar: Progressive Web Application \(PWA\) Development In Helix ITSM](#)

File: Developing-Pwa\Developing-Pwa.html

Progressive web applications (PWA) are applications that use common web technologies including HTML, CSS, and JavaScript. These applications are platform-independent and can be used across different devices such as desktops, tablets, and mobiles.

Best practice

Use IS for developing new codeless and code-based applications.

Use Developer Studio for customizing PWA screens for existing applications such as ITSM.

You can develop a PWA in Developer Studio. PWA technology provides you with the ability to create applications and additions to the ITSM applications that work across multiple platforms. For example, the same application runs on an iOS device or an Android device without you having to develop a separate client for each device.

Progressive web applications automatically fit the screen of the device they run on.

Related topics

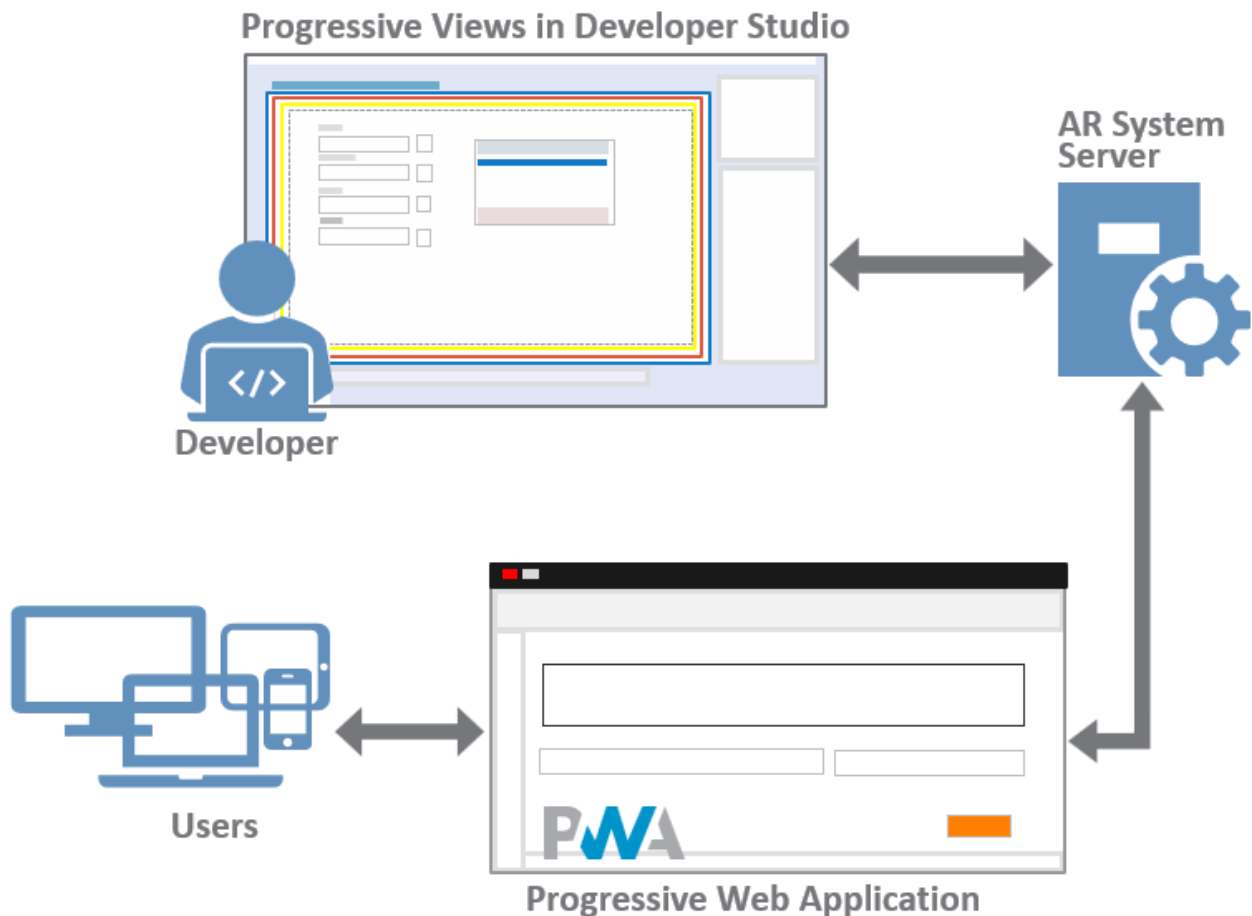
[Comparison-between-Developer-Studio-and-IS-capabilities](#)

[Mid-Tier-architecture](#)

[Enabling-Progressive-Views-for-forms-by-using-Centralized-Configuration-Settings](#)

[Adding custom fields to your views by using Smart ITCommunity: Video series for PWACommunity blog: Whitepaper on ITSM PWA Implementation Methodology](#)

The following diagram shows the high-level process of developing applications with PWA technologies by using Developer Studio.



Topics related to developing PWA in Developer Studio

Read through the following topics to learn more about developing a Progressive Web Application (PWA) in Developer Studio.

Action

Reference

To understand the basic concepts of what PWAs are and their advantages

[Progressive-Web-Applications-overview](#)

To understand the Mid Tier architecture in general as well as Mid Tier architecture for PWAs

[Mid-Tier-architecture](#)

To understand how flow panel holders can in creating create PWA screens and how they help with making screens responsive to varying screens

[Flow-panel-holders-for-Progressive-Web-Applications](#)

To understand Progressive Web Application screens with a few examples

[Progressive-web-application-screens](#)

To know the basics of accessing and using Developer Studio

[Accessing-and-navigating-Developer-Studio](#)

To create a Progressive Web Application you must first enable Progressive View for forms in Developer Studio

[Enabling-Progressive-Views-for-forms-by-using-Centralized-Config](#)

To know best practices when developing and customizing a PWA

[Customization-best-practices-for-developing-PWA](#)

To know the differences between the Developer Studio features for Mid Tier versus the features available in Progressive View

To learn about the localization support for a PWA

To know how to use autocomplete for character fields in search menus

To achieve better performance by setting the active links to run in parallel

To know the properties supported when developing a progressive view

Creating filters and advanced filters in progressive views that enable PWA users to search records and to fill forms

To know the icons available in Developer Studio for PWA and how to use them in your applications

To create various types of selection lists

[Creating dynamic multiple selection lists for Progressive Web Applications by using JSON data](#)

Add fields that allow editing and formatting of text, adding images, and so on

Add a pause to a character field in a workflow

Create a survey in the application

Manage attachments in the application

Add typeahead to mention users in the organization

Create a progress bar that shows stages of a workflow or process in your application

Add autocomplete to a field

Adding images in the interface of your application

Automatically aligning fields in Developer Studio

Change the look and feel of your application to suit the branding and styles of your organization

[Differences-between-Developer-Studio-features-available-for-Mid-Tier-versus-Progressive-View](#)

[Localizing-a-Progressive-Web-Application](#)

[Creating-search-menus](#)

[Creating-active-links](#)

[Field properties](#)

[Creating-filters](#)

[Creating-an-advanced-filter-in-Progressive-Web-Applications](#)

[Creating-a-faceted-filter](#)

[Previews-of-icons-available-in-Developer-Studio-to-support-PWA](#)

[Adding-icons-to-UI-elements-on-forms-by-using-Developer-Studio](#)

[Creating-selection-lists-in-Progressive-Web-Applications](#)

[Adding-rich-text-fields-in-Progressive-Web-Applications](#)

[Adding-a-pause-interval-in-a-character-field-for-a-workflow-in-PWA](#)

[Creating-a-survey-questionnaire](#)

[Managing-attachments-in-Progressive-Web-Applications-by-using-Developer-Studio](#)

[Configuring-the-mention-component](#)

[Configuring-the-Progress-bar-widget-in-Progressive-Web-Applications](#)

[Automatically-completing-menu-entries](#)

[Adding-background-images-to-fields-and-form-views](#)

[Aligning-fields](#)

[Rebranding-Progressive-Web-Application-screens](#)

Videos related to developing PWA in Developer Studio

You can watch the following videos to know more about developing progressive web applications:

Video title	Description
Webinar: Progressive Web Apps Introduction & Best Practices	Watch Video: https://www.youtube.com/watch?v=sVJXAhfVzQk
Webinar: Progressive Web Application (PWA) Development In Helix ITSM	Watch Video: https://www.youtube.com/watch?v=-J-ENCJ8bc4

Overview of Progressive Web Applications

■ [Watch Video:](#)

<https://www.youtube.com/watch?v=7DnZnfo6uZ8>

 <https://youtu.be/7DnZnfo6uZ8>

■ [Watch Video:](#)

https://www.youtube.com/watch?v=Yal3s28w_XY

 https://youtu.be/Yal3s28w_XY

■ [Watch Video:](#)

https://www.youtube.com/watch?v=s14_I_r9DwQ

Use cases for Progressive Web Applications.

 https://youtu.be/s14_I_r9DwQ

This video (2:08) provides use cases for Progressive Web Application.

■ [Watch Video:](#)

<https://www.youtube.com/watch?v=e2tOOdpBAB0>

Localizing a Progressive Web Application

 <https://youtu.be/e2tOOdpBAB0>

This video (7:31) provides an overview of how to localize a Progressive Web Application.

■ [Watch Video:](#)

<https://www.youtube.com/watch?v=m33BWG-mEol>

Panel holder display types, creating fixed panels and complex layouts with flow panels

 <https://youtu.be/m33BWG-mEol>

This video (4:52) demonstrates creating Fixed panels in Developer Studio when you enable progressive views for a form.

■ [Watch Video:](#)

<https://www.youtube.com/watch?v=TnwUS2RF28Q>

 <https://youtu.be/TnwUS2RF28Q>

This video (13:03) demonstrates creating complex layouts with flow panels.

■ [Watch Video:](#)

https://www.youtube.com/watch?v=-T90EgZ_Ztl

Configuring a progress bar widget

 https://youtu.be/-T90EgZ_Ztl

This video (3:37) describes how to configure and use the progress bar widget.

■ [Watch Video:](#)

<https://www.youtube.com/watch?v=sFkBJ-cjcH0>

Managing attachments in PWA by using a remote form

 <https://youtu.be/sFkBJ-cjcH0>

This video (5:15) shows how to create attachment pools that allow variable number of attachments in progressive web applications.

Troubleshooting PWA

■ Watch Video:

<https://www.youtube.com/watch?v=MfSTrjhwau8>

 <https://youtu.be/MfSTrjhwau8>

This video (3:40) shows you how to use additional log settings and generated HTML to diagnose and troubleshoot PWA

■ Watch Video:

<https://www.youtube.com/watch?v=349xdes86qM>

Adding images to buttons and background images to fields and form views.

 <https://www.youtube.com/watch?v=349xdes86qM>

This video (3:54) provides a summary of adding static or SVG images in Developer Studio for progressive views.

■ Watch Video:

<https://www.youtube.com/watch?v=3eRIML5mOgl>

Creating dynamic selection lists for Progressive Web Applications by using JSON data

 <https://youtu.be/3eRIML5mOgl>

This video (4:06) explains the concept and also demonstrates how the JSON properties and the JSONMAP function works.

■ Watch Video:

<https://www.youtube.com/watch?v=v0VVhzPHQww>

Creating a survey questionnaire

 <https://youtu.be/v0VVhzPHQww>

This video (5:33) provides a summary of creating a survey questionnaire in Developer Studio for progressive views:

This video (3:54) provides a summary of adding static or SVG images in Developer Studio for progressive views:

Adding images in PWA

■ Watch Video:

<https://www.youtube.com/watch?v=349xdes86qM>

 <https://www.youtube.com/watch?v=349xdes86qM>

■ Watch Video:

https://www.youtube.com/watch?v=_JhOoLN2agY

Aligning fields

 https://youtu.be/_JhOoLN2agY

This video (3:50) provides a summary of how automatic alignment of fields happens in progressive views in Developer Studio.

Creating typeahead fields for PWA

■ [Watch Video:](#)

<https://www.youtube.com/watch?v=hhvyBy99Ujg>



<https://youtu.be/hhvyBy99Ujg>

File:

Developing-Pwa\Enabling-Progressive-Views-For-Forms-By-Us

As a developer, you must enable progress views for out-of-the-box forms and custom forms when creating Progressive Web Applications (PWAs). Use the information in this topic to understand the methods for enabling progressive views for Smart IT and custom applications.

Related topics

[Developing-Progressive-Web-Applications](#)

[Creating-a-Progressive-View-in-Developer-Studio-to-create-Progressive-Web-Applications](#)

Before you begin

- Make sure that you have installed AR System server, Mid Tier, and Developer Studio.
If you are a ITSM user, configure a client gateway to access and customize your application by using Developer Studio. For more information, see the following topics in the Subscriber documentation:
 - [Client Gateway connectivity](#)
 - [ITSM customizations, configurations, and changes](#)
 - [ITSM Customization policy](#)

(Smart IT only) If you plan to use the **Progressive Views configuration** form, in the Centralized Configuration, make sure that the following parameters are set to the appropriate values:

Parameter	Value
mid_tier.base.url	Mid Tier URL of Progressive Web Application
arsystem.security_iframe_allowfromurls	Smart IT base URL
Enable-Progressive-Views	T

For more information, see [Enabling the Progressive Web Application screens](#).

To enable Progressive Views for forms by using the Centralized Configuration Settings

In Centralized Configuration, set the **Enable-Progressive-Views** property to **T** (True) as shown in the following image:

Component Type: com.bmc.arsys.server.shared

Component Name:

Setting Name	Setting Value
Access-Control-Allow-Headers	Pragma,X-Requested-By,Authorization,Content-Typ
Access-Control-Allow-Headers	Pragma,X-Requested-By,Authorization,Content-Typ
Access-Control-Allow-Origin	http://clm-aus-u5ng01:8080
Apply-Cognitive-For-Categorization	No
Apply-Cognitive-For-Swarming	No
Auth-Token-Signature-Salt	*****
Cognitive-Incident-AutoResolve	No
Cognitive-Service-Confidence-Threshold	90.0
Cognitive-Service-Training-Data-Generator-Threads	1
Cognitive-Service-Url	sdfasdfs
Cognitive-Service-User-Name	dfsdfsasdf
Cognitive-Service-User-Password	*****
Enable-Cognitive-Service	T
Enable-Progressive-Views	T
Jetty-Port	8008
Record-Object-Relationships	T
Redirect-URL	cmdb clm-aus-tr1703 http://clm-aus-tr1703.bmc.co
Upgrade-Status	

Buttons: Add, Delete, Apply, Close

InDeveloper Studio, select the form for which you want to enable progressive views:

1. From the**Form**menu, select**Create New View**.
2. From the**View Typelist**, select**Progressive**.

Create New View

View Type: **Progressive**

Developer Studioapplies certain restrictions on modifying progressive views. For more information, see[Customization-best-practices-for-developing-PWA](#).

3. Launch theMid Tierby using the following URL format:

ARSystemServerName:portNumber/arsys/pwa/#/forms/ARSystemServerName/formName/NewDefault%20Admin

For example:

Server1:8080/arsys/pwa/#/forms/Server1/My%20Form/NewDefault%20Administrator%20View

File:

Developing-Pwa\Flow-Panel-Holders\Flow-Panel-Holders.html

A Flow panel holder is available when you enable a Progressive view for your form. For more information, see [Progressive View quick start guide](#).

The Flow panel holder supports the properties required to enable a progressive view. Use a Flow Panel holder for aligning fields on a form. You can also group the fields to ensure correct alignment of fields on a form. This panel does not have a panel header, and supports Container panel holder (Default), Sub Panel holder, and Section Panel holder types.

The following video (4:52) demonstrates creating Fixed panels in Developer Studio when you enable progressive views for a form:

■ Watch Video: <https://www.youtube.com/watch?v=m33BWG-mEol>

▶ <https://youtu.be/m33BWG-mEol>

The following video (13:03) demonstrates creating complex layouts with flow panels:

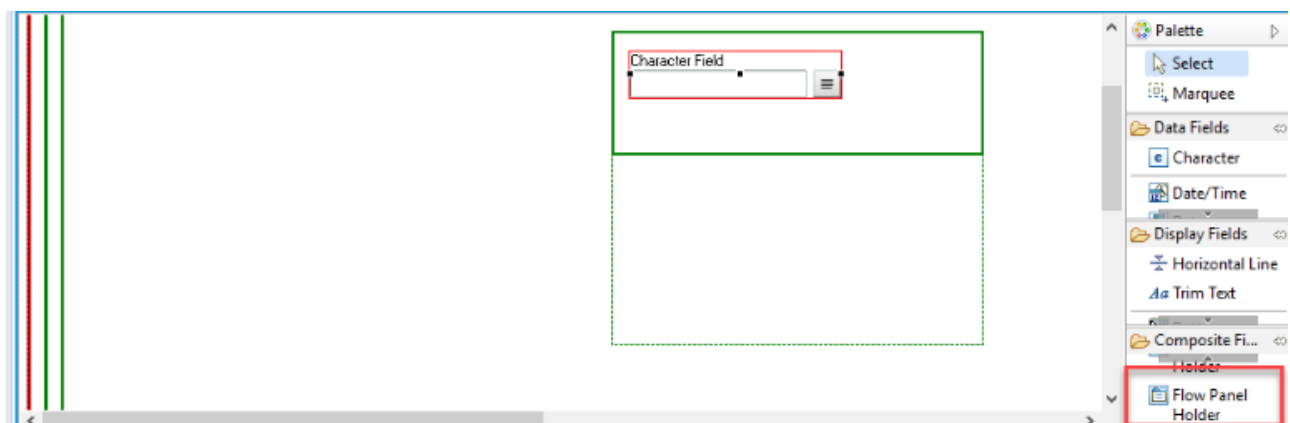
■ Watch Video: <https://www.youtube.com/watch?v=TnwUS2RF28Q>

▶ <https://youtu.be/TnwUS2RF28Q>

Panel arrangement sequence

The panels are arranged in the following sequence or hierarchy:

Flow Panel holder > Container Panel Holder > Container Panel > Sub Panel Holder > Sub Panel > Section Panel Holder > Section Panel > Section Panel Holder > Section Panel > fields

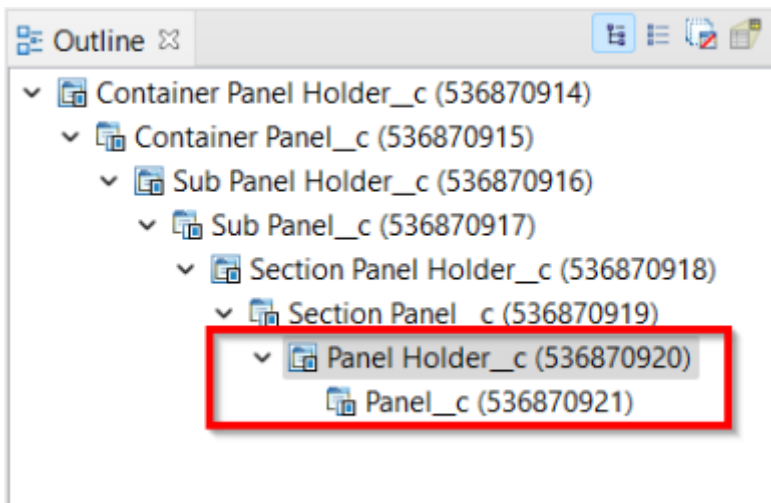


- A progressive view can hold only a Container type Flow panel holder and a Fixed panel holder.
- By default, a Flow panel holder has one panel as a child panel.
- You can change the read-only property type of any panel only when you change the property type of Flow panel holder.

Tip

When you add Flow Panel Holder to the Progressive View from the palette in Developer Studio, the flow panel holder is represented by a panel holder in the Outline pane.

In the following example figure, the flow panel holder is represented by **Panel Holder_c** enclosing a panel named **Panel_c**:



Container Flow panel holder

- By default, a Container type Flow panel is added to a Flow panel holder.
- You can delete a Container Flow panel only if you remove all progressive views.
- A Container Flow panel holder can have one or more Sub panels.

Section Flow panel holder

- You can add a Section Flow panel holder only under a Sub panel or Section panel.
- A Section panel can either have a Section type Flow panel holder or non-panel fields.
- The Section Flow panel holder added by framework contains only Section panel as children.

Sub panel holder

The Sub panel holder behaves as a Section panel on the desktop client. However, it is displayed as a tab on a mobile client.

- You can add a sub panel holder only under the container panel.
- A sub panel holder can have one or more sub panels.

For more information about developing PWA, see [Developing-Progressive-Web-Applications](#).

Related topics

[Panel-holder-display-types](#)

[Grouping-fields-with-panels](#)

File:

Developing-Pwa\Localizing-A-Progressive-Web-Application\Lo

provides a [localization toolkit](#) in Developer Studio to localize your AR System server applications from English to other supported locales. Use the localization toolkit and the AR System Message Catalog form to localize your Progressive Web Application (PWA).

The localization toolkit is used to extract the localizable strings from the objects. After extracting, export the strings to a Tab Separated Values (TSV) file, translate the strings and import the translated strings into the AR System Message Catalog form by using the ARMsgCatalogEntryCreator utility. This utility creates new records in the AR System Message Catalog form. The PWA screens use these records in the AR System Message Catalog form and display localized strings based on the browser locale.

Related topics

[Progressive View quick start guide](#)

[Using the localization toolkit \(Video\)](#)

[Synchronizing-Mid-Tier-cache](#)

To localize a progressive web application, you don't need the static, locale-specific views created by the localization toolkit for standard applications. For PWA, the localization toolkit is used only to extract the localizable strings from the objects and to export the strings into a TSV file.

PWA uses the AR System Message Catalog form to store and retrieve translations as required. Thus, as a Remedy developer, you do not need to create locale-specific views for the same objects by using the new form.

If you are a ITSM user, you must configure a client gateway to access and customize your application by using Developer Studio.

Review the [Best practices for localizing a progressive web application](#) section below before you perform the localization process.

The following video (3:50) provides an overview of how to localize a Progressive Web Application.

■ [Watch Video: //www.youtube.com/watch?v=t-3YflcljU](https://www.youtube.com/watch?v=t-3YflcljU)

▶ <https://youtu.be/t-3YflcljU>

PWA supports the following languages out-of-the-box:

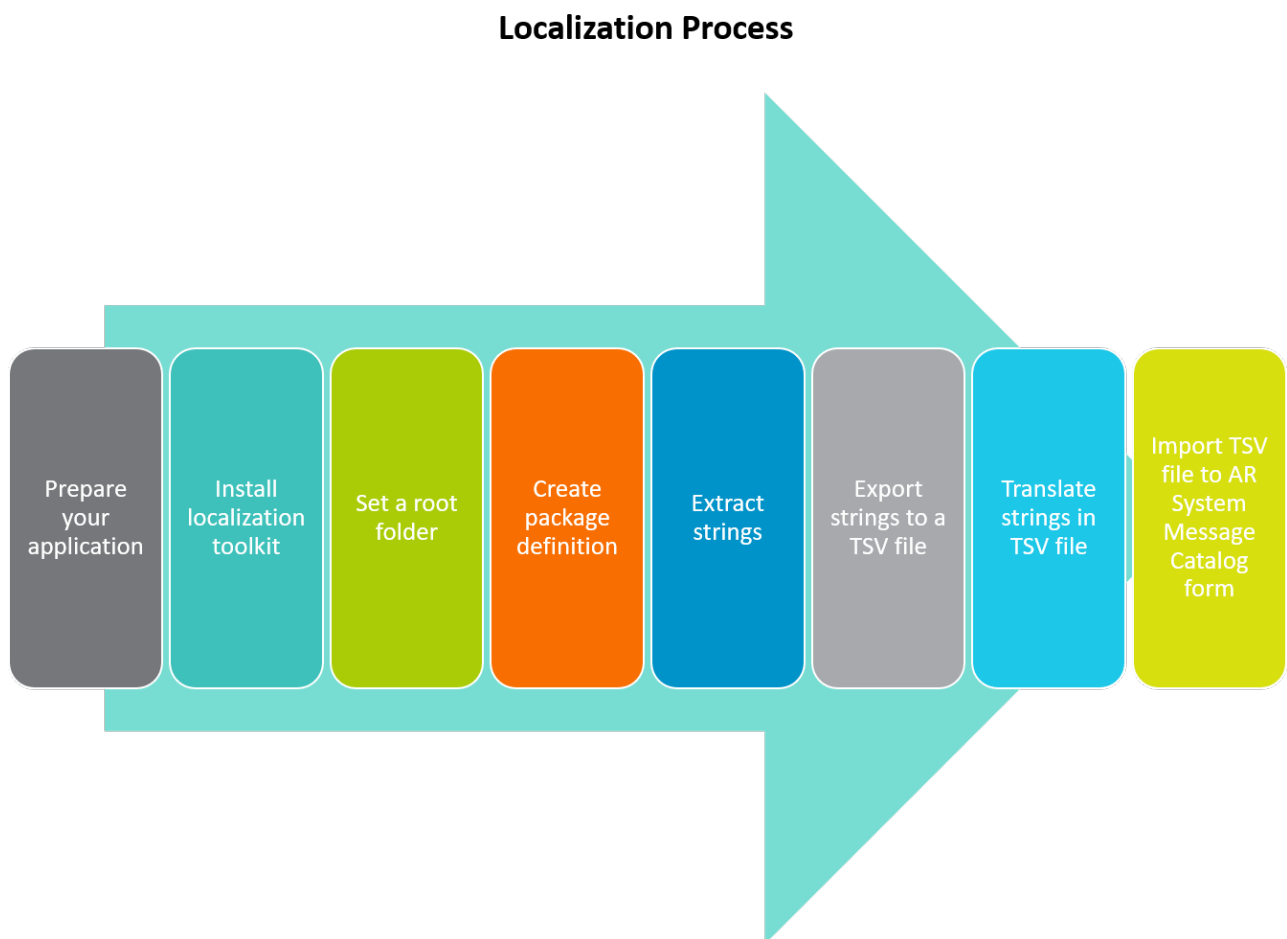
- en—English (default)
- de—German
- es—Spanish
- fr—French
- iw_IL—Hebrew

- it—Italian
- ja- Japanese
- ko-Korean
- pt_BR—Portuguese
- ru—Russian
- zh_CN—Chinese

If you want to localize your Progressive Web Application in any other language, use the following localization process.

Localization process

Perform the following steps in Developer Studio to complete the localization process:



1. Prepare your application that you want to localize.
2. Install the localization toolkit.
3. Set a root folder.
4. Create a package definition for the forms that you want to localize.
5. Extract the strings that you want to localize.
6. Export the strings in a file that has the default Tab Separated Values (TSV) format.
7. Translate the strings in the TSV file.

8. Import the TSV file to the AR System Message Catalog form.

To prepare your application that you want to localize

To be ready for localization, your server must be configured for localization. Additionally, applications must have the correct localization properties set for forms, their associated views, and fields. By default, all forms, views, and fields are configured to be localized, so you should not have to manually set each property. Conversely, if you have specific forms, views, or fields that you *do not* want localized, set the property accordingly.

For instructions, see [Preparing-your-application-to-be-localized](#).

To install the localization toolkit

When you install the localization toolkit, the localization toolkit files are added to the **DeveloperStudio** folder of your AR System installation. (The default installation directory is **C:\Program Files\ Software\ARSystem\ DeveloperStudio**.)

The localization toolkit plug-ins are placed in the **plugins** folder with the other Developer Studio plug-ins. A Localization Package Definitions branch is added to each server in the AR System Navigator in Developer Studio.

For instructions, see [Installing-the-localization-toolkit](#).

To set a root folder

To begin the localization process, set a root folder, which stores:

- The localization database (**I10n.db**)
- The log file that logs the translation progress (**I10n.log**)
- Package definition files that you create for translation

The default root folder is the **BMCRemedyDeveloperStudio\InstallDir\workspace\I10N** folder.

For instructions, see [Setting-a-root-folder](#).

To create a package definitions for the forms

The Localization Package Definition editor in Developer Studio makes it easy to create a localization package definition for your application and any other forms that you want in the package definition.

For instructions, see [Creating-a-localization-package-definition](#).

To extract the strings that you want to localize

After you create a localization package definition, you can extract the strings for localization into the localization database (**I10n.db** in the root folder).

For instructions, see [Extracting-the-strings-for-localization](#).

To export the extracted strings to a TSV file

For Progressive Views, you must use the TSV format.

Important

- While exporting strings for localization, you can opt to export only the items that are related with Progressive Views.
- The time taken to extract or export strings depends upon the number of items involved. Therefore, it can take a while to export large number of items, for example, 1000 or 2000.
- You cannot use a TSV file to import translations through the localization toolkit. This file is generated specifically for the AR System Message Catalog form. Use the [command-line utility](#) to import a TSV file in to the AR System Message Catalog form after translation.

Perform the following steps:

1. In Developer Studio, open the localization package definition.
2. Click **Export Strings**.
3. In the Localization Item Export dialog box, enter the full path to a target **.tsv** file that will contain the extracted strings.
If the path you enter does not have a TSV file, then it is created. If you specify a full path, the file is created at that location. If you specify only a file name, the file is created in the root folder.
The default root folder is the *BMCRemedyDeveloperStudioInstallDir\workspace\I10N* folder.
4. Select the locale you want to export.
5. Select the state of the strings that you want to export.

Click **OK**.

Only those items that are set to the state you selected in step 5 (for example, "Ready to Translate") are exported to the **.tsv** file. For more information, see [Reviewing the status of the translation progress](#).

Tip

For each locale, add a new column with localized strings that map to the **Source Text** column to the exported file. Add the locale code to the header of the new localized column.

Tab Separated Values (TSV) file

The following image shows a sample **TSV** file *before translation*:

	A	B	C	D	E	F	G	H
1	Message-Type	Form Name	View ID	Field ID/Message Number	Source Text			
2	16 L10N		536870912		1 Request ID			
3	16 L10N		536870912		2 Submitter			
4	16 L10N		536870912		3 Create Date			
5	16 L10N		536870912		4 Assigned To			
6	16 L10N		536870912		5 Last Modified By			
7	16 L10N		536870912		6 Modified Date			
8	16 L10N		536870912		7 Status			
9	19 L10N		536870912		7 New			
10	19 L10N		536870912		7 Assigned			
11	19 L10N		536870912		7 Fixed			
12	19 L10N		536870912		7 Rejected			
13	19 L10N		536870912		7 Closed			
14	16 L10N		536870912		8 Short Description			
15	16 L10N		536870913	536870914	Panel Holder			
16	16 L10N		536870913	536870916	Panel Holder			
17	16 L10N		536870913	536870918	Panel Holder			
18	16 L10N		536870913	536870915	Container Panel			
19	16 L10N		536870913	536870917	Sub Panel			
20	16 L10N		536870913	536870919	Section Panel			
21	16 L10N		536870913	536870920	Character Field			
22	35 L10N		536870913	536870921	Button			

The following table provides information about the columns in the exported TSV file:

Column name	Description
Message-Type	The type of the object for which the string is extracted. See Types of objects exported .
Form Name/Message ID	The name of the Form or Active Link or Filter
VUI ID	The View ID of the Form
Field ID/Message Num	Field ID or message number from the Active Link or Filter actions The string to be localized. It includes the following: <ul style="list-style-type: none"> Field labels, such as Data Fields, Control Fields, and Selection Fields Selection alias for fields such as Radio button and Drop-down list Help text from forms and fields Alternative text for fields Message strings from Active Links and Filters
Source Text	

Important

In a TSV file, characters such as comma (,) are replaced by tags such as <comma> in strings.

Because TSV is a delimiter-separated file, characters such as commas or new lines in the file result in localization strings occupying multiple cells and rows instead of just one cell. Therefore, these are replaced

by tags as shown in the following table:

Character	Description	Tag used
\r\n	Carriage return and new line	<CrLf>
\n	New line	<Lf>
\r	Carriage return	<Cr>
,	Comma	<Comma>

Types of objects exported

[Click here to see the types of objects currently exported in a TSV file](#)

Message type	Type of object	Description of the string
0	System Message	Belongs to a System message action.
1	Active Link Message	Belongs to an Active Link message action.
2	Filter Message	Belongs to a Filter message action.
3	Active Link Help Text	Belongs to the Help Text property of an Active Link.
4	Form Help Text	Belongs to the Help Text property of a Form.
5	Field Help Text	Belongs to the Help Text property of a Field.
6	Container Description	Belongs to the Description property of a Container.
7	List Menu Definition	Belongs to the Definition property of a List Menu.
8	External Report	Belongs to an External Report.
9	Container Label	Belongs to the Label property of a Container.
10	Container Help	Belongs to the Help property of a Container.
11	Application Help	Belongs to the Help property of an Application.
12	Application About	Belongs to the About property of an Application.
13	Application Help Index	Belongs to the Help Index property of an Application.
14	Flashboards System Message	Belongs to the System Message property of Flashboards.
15	Flashboards Label	Belongs to the Label property of Flashboards.
16	Field Label	Belongs to the Label property of a Field.
17	Field Help Text	Belongs to the Help Text property of a Field.
18	Alternative Text	Belongs to the Alternative Text property of a Field.

19	Selection Alias	Belongs to the Alias property of a Selection Field.
20	Label for Empty Value	Belongs to the Empty Value property of a Label.
21	Alias Singular	Belongs to the Singular property of an Alias Field.
22	Alias Plural	Belongs to the Plural property of an Alias Field.
23	Alias Short Singular	Belongs to the Short Singular property of an Alias Field.
24	Alias Short Plural	Belongs to the Short Plural property of an Alias Field.
25	Entrypoint Label Default New	Belongs to the Default New property of an Entrypoint Label.
26	Entrypoint Label Default Search	Belongs to the Default Search property of an Entrypoint Label.
27	Attach Add Label	Belongs to the Add Label property of an Attach Field.
28	Attach Delete Label	Belongs to the Delete Label property of an Attach Field.
29	Attach Display Label	Belongs to the Display Label property of an Attach Field.
30	Attach Save Label	Belongs to the Save Label property of an Attach Field.
31	Attach File Name Title	Belongs to the File Name Title property of an Attach Field.
32	Attach File Size Title	Belongs to the File Size Title property of an Attach Field.
33	Attach Label Title	Belongs to the Label Title property of an Attach Field.
34	Attach Deselect Label	Belongs to the Deselect Label property of an Attach Field.
35	Button Text	Belongs to the Text property of a Button.
36	Menu Text	Belongs to the Text property of a Menu.
37	Menu Help	Belongs to the Help property of a Menu.
38	Tooltip	Belongs to the Text property of a Button.
39	HTML Text	Belongs to the Text property of HTML.
40	Enums Labels	Belongs to the Labels property of an Enumeration.
41	Text	Belongs to the Text property.
42	Table Selection Column Label	Belongs to the Column Label property of a Table Selection.

43	Table Root Node Alternative Text	Belongs to the Alternative Text property of a Table Root Node.
44	Table Entries Returned	Belongs to the Entries Returned property of a Table.
45	Table Refresh	Belongs to the Refresh property of a Table.
46	Table Delete	Belongs to the Delete property of a Table.
47	Table Report	Belongs to the Report property of a Table.
48	Table Select All	Belongs to the Select All property of a Table.
49	Table Deselect All	Belongs to the Deselect All property of a Table.
50	Table Preference	Belongs to the Preference property of a Table.
51	Table Chunk Next	Belongs to the Chunk Next property of a Table.
52	Table Chunk Previous	Belongs to the Chunk Previous property of a Table.
53	Row Label	Belongs to the Label property of a Row.
54	Row Label Plural	Belongs to the Label Plural property of a Row.
55	Table Auto Refresh	Belongs to the Auto Refresh property of a Table.
56	Table Unread	Belongs to the Unread property of a Table.
57	Table Read	Belongs to the Read property of a Table.
58	Table Not Refresh	Belongs to the Not Refresh property of a Table.
59	Tree Table Custom Value	Belongs to the Custom Value property of a Tree Table.
101	Message	Belongs to the Message property.
102	Data	Belongs to the Data property.
103	Entry Point	Belongs to the Entry Point property.
104	Menu	Belongs to the Menu property.
105	Description	Belongs to the Description property.
106	Guide Help Text	Belongs to the Help Text property of a Guide.
107	App Help Text	Belongs to the Help Text property of an Application.

To translate the strings in a TSV file

Send the exported TSV file to a vendor for translation.

Best practice

For accurate translation, we recommend that you send the file to professional translators. (Open source and commercial editors are available for purchase by several third-party vendors.)

After the translation is completed, the TSV file has an additional sixth column which contains all the translations. The TSV file contains localized strings starting from the sixth column. That is, each column represents a single locale.

The following image shows a sample **TSV file after translation**:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Message-ID	Form Name	View ID	Field ID/Message	Source Text	ja										
2	16	L10N	536870912	1	Request ID	기록을 생성하려면 '연락처 조직 관리자' <comma> '연락처 지원 관리자' 또는 '연락처 인력 관리자' 권한이 있어야 합니다.										
3	16	L10N	536870912	2	Submitter	라이선스 유형이 변경되었습니다. 관련 번들을 추가한 후 기록을 저장하십시오.										
4	16	L10N	536870912	3	Create Date	라이선스 유형이 변경되었습니다. 모든 번들을 삭제한 후 기록을 저장하십시오.										
5	16	L10N	536870912	4	Assigned To	이 번들을 삭제하면 번들에 대한 권한 및 라이선스가 제거됩니다.										
6	16	L10N	536870912	5	Last Modified By	입력한 검색 기준과 일치하는 기록이 없습니다.										
7	16	L10N	536870912	6	Modified Date	볼 수 있는 기존 작업 정보가 없습니다.										
8	16	L10N	536870912	7	Status	이 티켓을 액세스 권한이 없는 회사에 할당했습니다. 더 이상 티켓을 보거나 수정할 수 없습니다. 양식을 지우려면 '삭제' 버튼을 클릭하십시오.										
9	16	L10N	536870912	8	Short Description	서비스 필드에 검색할 전체 또는 일부 서비스 이름을 입력하십시오.										
10	16	L10N	536870913	536870914	Panel Holder	입력한 값에 해당하는 ci가 없습니다. 유효한 ci를 입력하십시오.										
11	16	L10N	536870913	536870915	Container Panel	입력한 값에 해당하는 ci가 없습니다. 유효한 ci를 입력하십시오.										
12	16	L10N	536870913	536870916	Panel Holder	입력한 값에 해당하는 ci가 없습니다. 유효한 ci를 입력하십시오.										
13	16	L10N	536870913	536870917	Sub Panel	입력한 값에 해당하는 ci가 없습니다. 유효한 ci를 입력하십시오.										
14	16	L10N	536870913	536870918	Panel Holder	입력한 값에 해당하는 서비스가 없습니다. 올바른 서비스 이름을 입력하십시오.										
15	16	L10N	536870913	536870919	Section Panel	입력한 값에 해당하는 서비스가 없습니다. 올바른 서비스 이름을 입력하십시오.										
16	16	L10N	536870913	536870920	Character Field	입력한 값에 해당하는 서비스가 없습니다. 올바른 서비스 이름을 입력하십시오.										
17	16	L10N	536870913	536870922	Attachment Pool	입력한 값에 해당하는 서비스가 없습니다. 올바른 서비스 이름을 입력하십시오.										
18	16	L10N	536870913	536870923	Attachment	\$300469300\$은(는) 저장된 회사 \$1000000001\$에 유효하지 않은 비용 센터입니다. [검색] 버튼을 사용하여 유효한 비용 센터를 선택하십시오.										
19	19	L10N	536870912	7	New	확인할 비용 센터를 입력해야 합니다.										
20	19	L10N	536870912	7	Assigned	보조 비용 센터가 있으면 기본 비용 센터를 제거할 수 없습니다.										
21	19	L10N	536870912	7	Fixed	자신의 프로파일에만 대체 승인자를 추가할 수 있습니다.										
22	19	L10N	536870912	7	Rejected	이 기능을 실행하려면 인시던트 번호가 필요합니다. 이 인시던트의 고객을 선택하여 인시던트 번호를 생성하십시오.										

To import a TSV file containing translated strings to the AR System Message Catalog Form

You can use the ARMsgCatalogEntryCreator utility that reads the TSV file and creates or updates an entry in the AR System Message Catalog form. The ARMsgCatalogEntryCreator utility accepts only a TSV file as an input.

Perform the following steps to import a TSV file to the AR System Message Catalog form using the ARMsgCatalogEntryCreator utility:

Before you use this utility, sort the TSV file four times—first by the Form Name column, next by the Field ID column, next by the View ID column, and finally by the Message Type column.

Important

Make sure that your AR System servers supports the locale that you want to translate to. Otherwise, records with corrupted data, instead of the actual data, might get created in the AR System Message Catalog form.

2. Navigate to <ARSystemInstallDir>/ARSystem/artools and find the **armsgcatalogentrycreator.bat** file.

From a command prompt, run the **armsgcatalogentrycreator.bat** file by using the following command:

ARMsgCatalogEntryCreator utility command syntax

armsgcatalogentrycreator.bat [x] [u] [p] [t] [f]

ARMsgCatalogEntryCreator utility command example

armsgcatalogentrycreator.batf<LocationofTSV file>\Translations.tsvu<Username>x<Server Name>

The following table describes the inputs for the Command Line Interface (CLI) and parameters for connecting to the AR System:

Parameter descriptions

Parameter	Description
x	Name of the server to connect to.
u	Name that identifies the user account for the AR System server.
	Password for the user account.
p	If you have a blank password, you can ignore this option.
t	TCP port number to connect to.
	If the port number is unknown, use 0.
f	Name of the TSV file.

Depending upon the row data, an entry is created or updated. For every locale, a separate entry is created per row. For example, a single row entry in the TSV file with three translations (that is, three different locales) creates three new entries for that string.

4. On the **Cache Settings** page of the Mid Tier Configuration Tool, click **Flush Cache**.

Important

The Selection Alias type of object (Message type 19) is handled in a special way. Suppose, there is a Selection Alias type of object called "Status" that has five aliases as New, Assigned, Fixed, Rejected, and Close. For each alias, instead of creating a separate entry, the aliases are combined into a single record.

For example:

```
{"New": "Neu", "Assigned": "Zugewiesen", "Fixed": "Fest", "Rejected": "Abgelehnt", "Closed": "Geschlossen"}
```

In this example, each alias is separated by a comma (,) and contains two values—the left one is the actual string that is being exported and the right one is the translated string.

Important

In the TSV file, columns after the Source Text column hold the translated data. If the translated data column for a language has the *same string* as the Source Text (English) column, the ARMsgCatalogEntryCreator utility does not create a record for the same string in the AR System Message Catalog Form.

For example, if you have a string 'ALL' in the English and the Russian language columns, the utility skips creating a new record for the string 'ALL' for the Russian language column.

This is applicable for all message types except **19 (Selection Field)** as all the values are combined in a single record.

Best practices for localizing a progressive web application

- Use independent forms only so that the related Active Links and Filters are also considered.
- For the form fields that you want to localize, set the value of the "Localize Label" property in Display Properties to **True**.
- When you delete a package definition, the package definition is deleted, but the extracted strings are not deleted from the localization database. If you create a package definition with the same name as the deleted one and export, you get the extracted strings from the deleted package definition as well. To avoid this issue, we recommend that you explicitly delete those strings from the database.

File:

Developing-Pwa\Managing-Attachments-Using-A-Remote-Form

As an application developer, you can define an attachment pool by using a Table field in Progressive View to enable end users to add any number of attachments in Progressive Web Applications (PWA). An attachment pool contains attachment fields that enable users to store text, graphics, audio, or video with a request.

To create an attachment pool, in Developer Studio, you must first create a regular form with an attachment pool in Standard View, and then associate it with a Table field of type attachment in Progressive View.

If you create an attachment pool as described

in [Managing-attachments-in-interfaces-by-creating-attachment-pools](#), end users can add only the number of attachments you specify in Developer Studio. Any change in the number of attachment fields, requires you to add more attachment fields in the Attachment Pool field in Developer Studio.

Related topics

[Enabling-Progressive-Views-for-forms-by-using-Centralized-Configuration-Settings](#)

[UI-layout-of-Progressive-Web-Applications](#)

[Customization-best-practices-for-developing-PWA](#)

[Workflow-considerations-for-attachment-pools](#)

The following video (5:15) shows how to create attachment pools that allow variable number of attachments in progressive web applications:

■ Watch Video: <https://www.youtube.com/watch?v=sFkBJ-cjcH0>

▶ <https://youtu.be/sFkBJ-cjcH0>

To create an attachment pool in PWA, you must perform the following high-level steps:



1

[Create a regular form in Standard View with an attachment pool.](#)

2

[Create a table field of type attachment in Progressive View.](#)

3

[Map the table field to the regular form.](#)

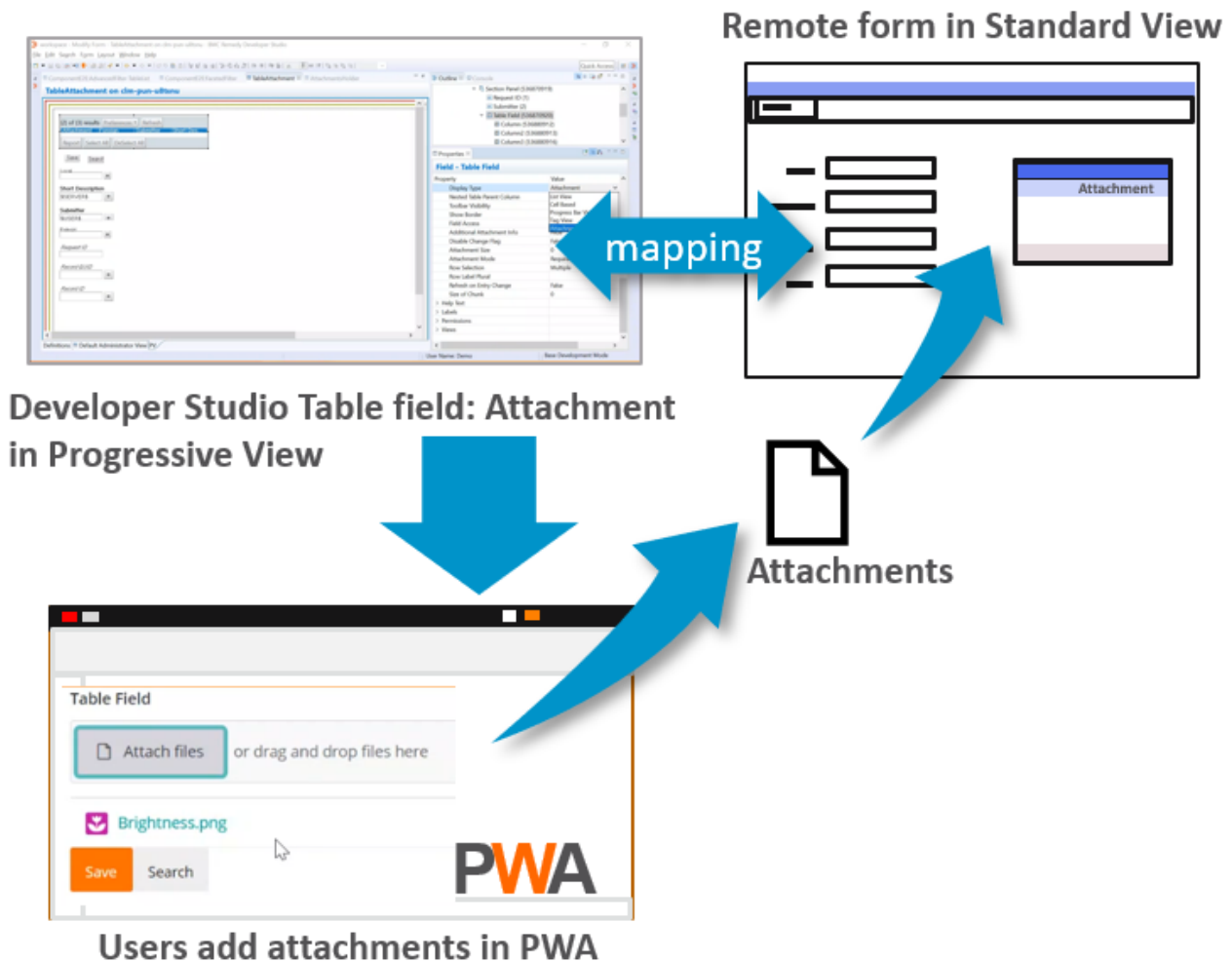
4

[Define active links to enable end users to save the attachments and to view them.](#)

5

[Validate whether the attachment pool works by logging in as a user.](#)

The following concept diagram explains how the table field set to type Attachment that you must define in Progressive View in Developer Studio maps to the remote form you must define in Standard View in Developer Studio.



Before you begin

- You must have a basic understanding of defining workflows. For more information see [Defining-workflow-to-automate-processes](#).

You must know how to create active links. For more information, see [Defining-Set-Fields-actions-to-assign-values-based-on-conditions](#).

Best practice

To create an application where you allow end users to delete data from the main form that contains attachments in the Progressive View, make sure that the related data from the remote form is also deleted. To delete the remote form data, we recommend that you create a strong association between

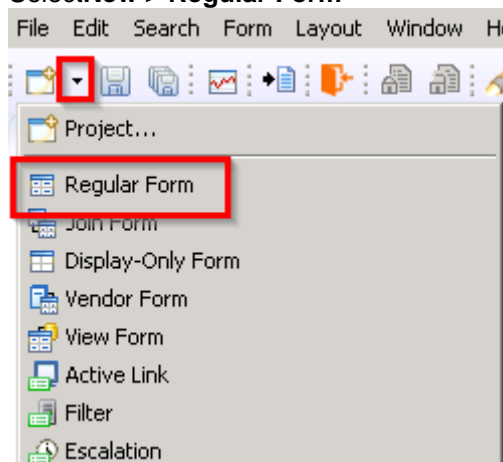
the parent form and the remote form with the same qualification as that of the table field of type attachment. This makes it possible for users to perform a cascade delete. For information about associations, see [Creating-associations](#).

To create an attachment pool for PWA by using a remote form

Step 1: To create a regular form with an attachment field in Standard View

1. Open Developer Studio and create a regular form in Standard View.

Select **New > Regular Form**

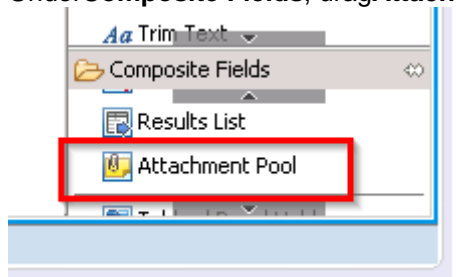


2. Save the form with a name.


Use a name that will help you search for the form easily. For example, **AttachmentFormSV**.

We use this form, that you just saved, in the Standard View as the remote form which stores the attachments.

3. Under **Composite Fields**, drag **Attachment Pool** and drop it into the view.



4. Select the attachment pool and in the properties, under **Attributes**, click the ellipsis next to **Attachment Fields**.

Property	Value
[-] Attributes	
Attachment Fields	<0> Attachment Fields 
[+] Change History	
[+] Database	
[+] Display	
[+] Help Text	

The Attachment Fields dialog box opens.

5. Click **New**.

An attachment field appears.

You can rename the field to identify it easily; for example, **AttachmentFieldSV**.

6. To add the field to the attachment pool field, click **OK**.

The regular form displays the attachment field.

File Name	Max Size	Attach Name
NULL	0	AttachmentFieldSV

7. At the bottom of the view editor, click the view tab.
8. (Optional) Under Properties, edit the **Label** and **Name** properties to rename the view.
9. Set default values for all the required fields.
Required fields are in bold.

Request ID	<input type="text"/>	
Submitter	\$USER\$	<input type="button" value="Menu"/>
Create Date	<input type="text"/>	<input type="button" value="Calendar"/>
Assigned To	<input type="text"/>	<input type="button" value="Menu"/>
Last Modified By	<input type="text"/>	<input type="button" value="Menu"/>
Modified Date	<input type="text"/>	<input type="button" value="Calendar"/>
Status	<input checked="" type="radio"/> New <input type="radio"/> Assigned <input type="radio"/> Fixed <input type="radio"/> Rejected <input type="radio"/> Closed	
Short Description	NA	<input type="button" value="Menu"/>

The preceding image shows an example in which, in Properties, we set a **Default Value** to **\$USER\$** for **Submitter** and **NA** for **Short Description**.

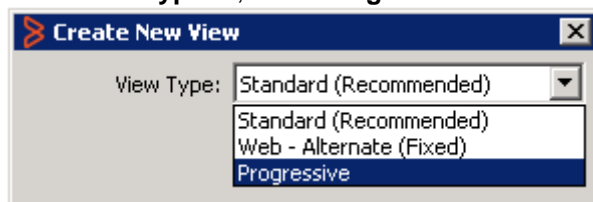
10. Click **Save** ().

Step 2: To create a table field of type Attachment

1. In Developer Studio, select **Form > Create New View**.

The Create New View dialog box opens.

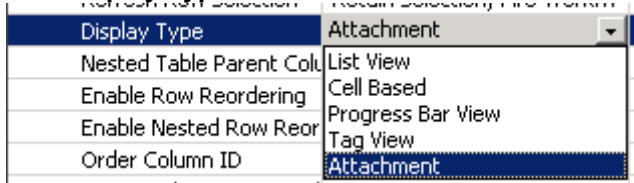
2. In the **View Type** list, select **Progressive**.



A new view is created; For example, **DefaultAdministratorView2_c**.

3. (Optional) Under Properties, edit the **Label** and **Name** properties to rename the view.
4. Drag and drop the **Table - List View** field.
5. On the **Properties** panel, expand the Display property group.

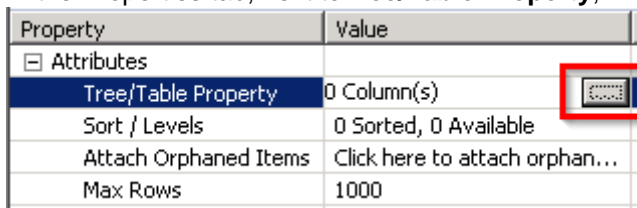
- Click **Display Type** and then select **Attachment** from the list.




The option **Attachment** is available only when the view is set to Progressive.

Step 3: To map the table field to the regular form created in Standard View

- In the Properties tab, next to **Tree/Table Property**, in the **Value** column, click the ellipsis () button.



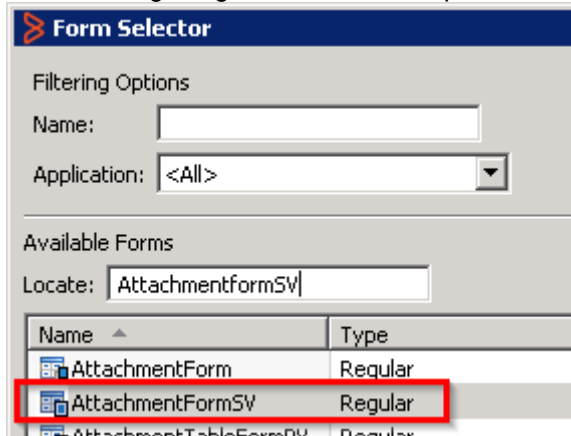
The Tree/Table Property dialog box opens.

- On the **Tree/Table Property** dialog box, next to the **Form Name** field, click the ellipsis ().

The Form Selector dialog box opens.

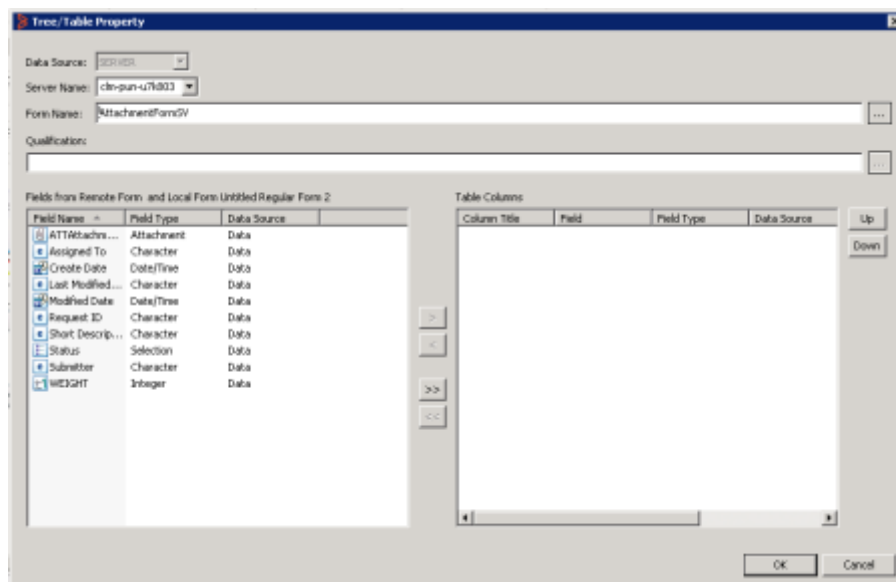
- Select the regular form in the Standard View that you created.

The following image shows the example in which we select **AttachmentFormSV**.

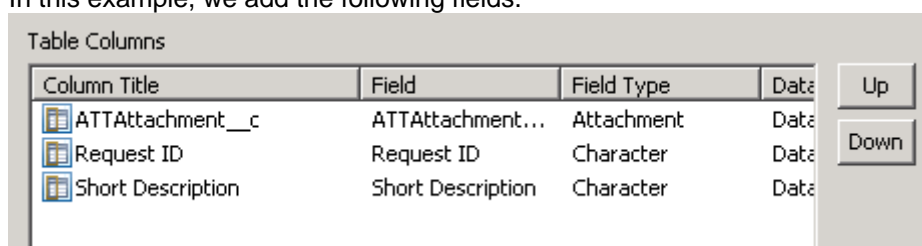


This regular form acts as the remote form that holds the attachment data.

When you close the Form Selector dialog box, you see that the Tree/Table Property dialog box with the available fields.

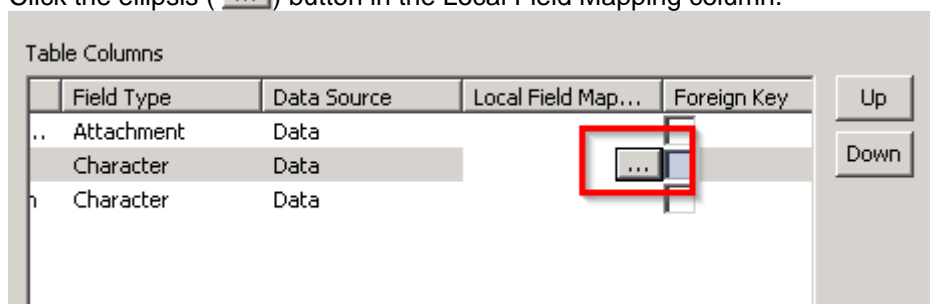


4. Add the field of type **Attachment** and any other required columns to the table. In this example, we add the following fields.



You must map at least one field of type Attachment.

5. Click the ellipsis (**...**) button in the Local Field Mapping column.



The Field Selector dialog box opens.

You might have to scroll sideways to view the Local Field Mapping column.

6. Select the field of the same data type and click **OK**.

Name	Field ID	Form	Data Type
Assigned To__c	4	Untitled Regular Form 2	Character
Create Date__c	3	Untitled Regular Form 2	Date/Time
Last Modified By__c	5	Untitled Regular Form 2	Character
Modified Date__c	6	Untitled Regular Form 2	Date/Time
Request ID__c	1	Untitled Regular Form 2	Character
Short Description__c	8	Untitled Regular Form 2	Character
Status__c	7	Untitled Regular Form 2	Selection
Submitter__c	2	Untitled Regular Form 2	Character

For example, for the column with title Request ID, you can select **Request ID__c**.

7. Add the **Local Field Mapping** for all columns in the table except the one that has Data Type set to **Attachment**.

This is required to pass data from the Progressive View form back to the regular form in Standard View.

8. Select the **Foreign Key** check box for at least one column.

The **Qualification** field is automatically populated and this qualification is used to fetch the data from the remote form. You must make sure that the **Foreign Key** is a unique identifier.

9. Click **OK** and save the form.

In this example, save the form as **AttachmentFormPV**.

10. (Optional) In **Display Properties**, set a value for **Size of Chunk** to define the number of attachments that are displayed in the PWA.

When end users add more attachments than this number, a **Show More** option appears.

11. Define an active link to enable end users to save the attachments as described in [Enable end users to save the attachments by defining an active link](#).

Enable end users to save the attachments and refresh the form by using active links

You must create a button to save attachments and also add an active link so that end users can view uploaded attachments.

Step 1: To create buttons to save and refresh in the Progressive View

1. In the Progressive View, under **Display Fields**, drag and drop **Button** into the view.



This is the button that end users click to save their attachments after adding them to the Progressive View form.

2. Select the new button and update the following required properties:

1. **Database > Name**

2. **Database > ID**

Optionally, you can enter ID as 1001 to enable saving the attachments. In such a case, you do not need to define an active link to commit changes as described later in this procedure.

3. **Display > Button Label**

Change the button label to **Save**.

4. **Permissions > Permissions.**

Add permissions as required.

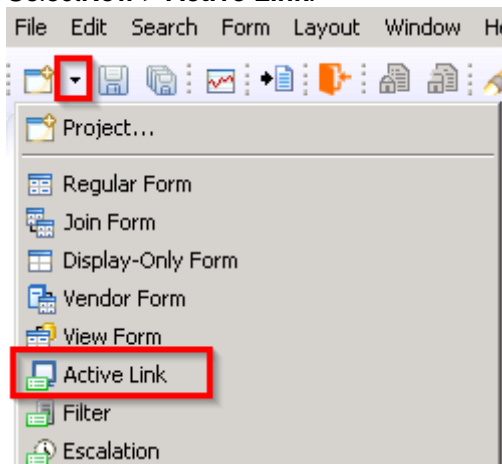
3. Add another button.

4. Modify the properties as required and change the label to **Refresh**.

Step 2: To create active links to commit changes

1. Create a new active link.

Select **New > Active Link**.



2. Select a server in the Select Server dialog box.

The Untitled Active Link screen appears.

3. In the Associated Forms section, click **Add**.

The Form Selector dialog box opens.

4. Select the form in the Progressive View which has the button.


In this example, select **AttachmentFormPV**.

5. In the Execution Options section, beside the **Button/Menu Field**, click the ellipsis () button.


The Field Selector dialog box opens.

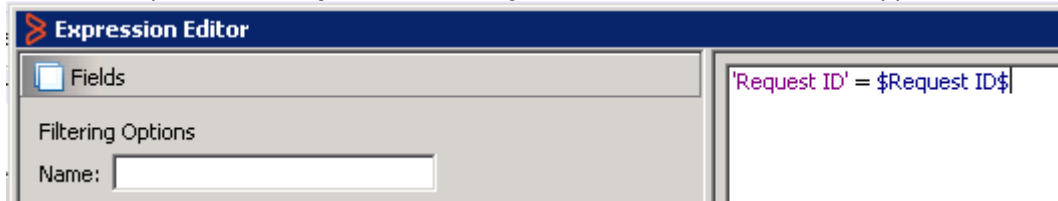
6. Select the button for which you want to create the active link.

In this example, it is the button for which you have set the **Button Label** value to **Save**. Whenever this button is clicked this active link is run.

7. In the **Self Action** section, right-click and select **Add Action > Commit Changes**.
8. In the **Self Action** section, right-click and select **Add Action > Open Window**.
9. In the **Window Type** list select **Display Directly**.
10. Beside the **Form View Name** field, select the ellipsis () button.
The View Selector dialog box opens.
11. Select the name of the view that contains the button for which you want to add the active link.
In this example, select **DefaultAdministratorView2_c**.

Step 3: To create a qualification for the active link

1. Click the ellipsis () button beside the **Qualification** field.
The **Expression Editor** dialog box opens.
2. In the **Forms** list, select the form in Progressive View which contains the button.
3. In the **Available Fields** list, select **Request ID**.
You can see the qualification being created in the field.
4. Complete the qualification.
In this example, enter '**Request ID**' = **\$Request ID\$** as this field is also mapped in the remote form.



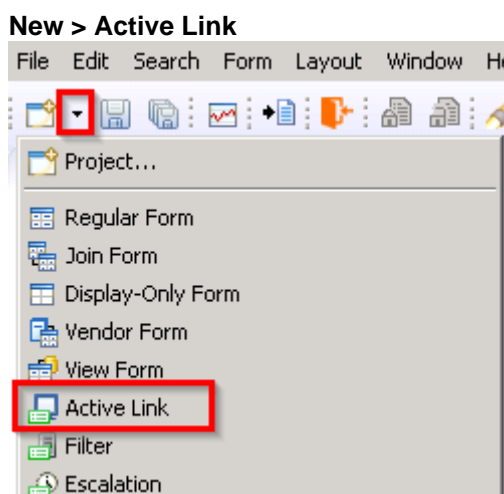
Retain the default settings and save this active link.

In this example, save this as AttachmentAL.

After the end users add attachments in the PWA and click **Save**, this active link saves a record in the remote form along with the attachments, and then reloads the PWA entry and all of its attachments in view mode.

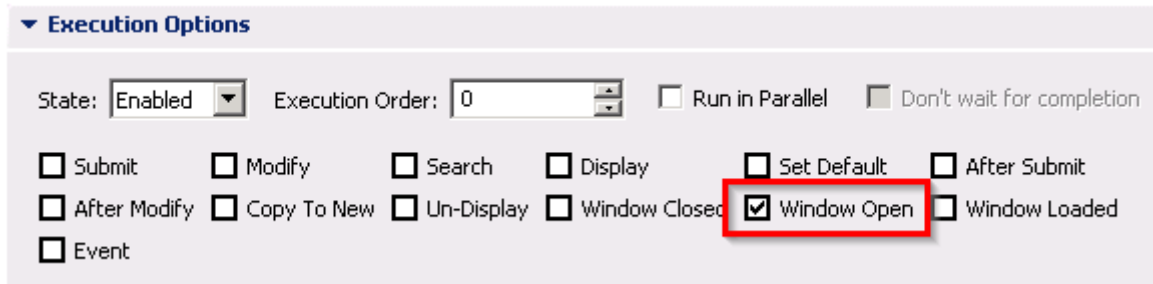
Step 4: Create an active link to refresh the view

1. Create a new active link.



2. Select a server in the Select Server dialog box.
The Untitled Active Link screen appears.

3. In the Associated Forms section, click **Add**.
The Form Selector dialog box opens.
4. Select the form in the Progressive View which has the button.
In this example, select **AttachmentFormPV**.
5. In the Execution Options section, select **Window Open**.



▼ Execution Options

State: Execution Order: ☐ Run in Parallel ☐ Don't wait for completion

☐ Submit ☐ Modify ☐ Search ☐ Display ☐ Set Default ☐ After Submit
☐ After Modify ☐ Copy To New ☐ Un-Display ☐ Window Closed ☒ Window Open ☐ Window Loaded
☐ Event

6. Beside the **Button/Menu/Field**, click the ellipsis () button.
7. The Field Selector dialog box opens.
8. Select the button for which you want to create the active link.
In this example, it is the button for which you have set the **Button Label** value to **Refresh**.
9. In the **Self Action** section, right-click and select **Add Action > Change Field**.
10. Click the ellipsis () button beside **Field** and select the table field.
11. Select the check box beside **Refresh Tree/Table**.
12. Save the active link.

Step 5: To validate whether the attachment pool works

1. Refresh your browser cache.
<serverName>:8080/arsys/shared/config/config.jsp
2. Click **Sync Cache**.
You must perform this step every time you make changes in Developer Studio.
3. In a browser, sign in to PWA with valid credentials.
<serverName>:8080/arsys/pwa/#/login

Change the URL to include the Progressive View form that you created in Developer Studio.

<serverName>:8080/arsys/pwa/#/forms/<serverName>/<formName>/<viewName>

In this example, the URL

is **<serverName>:8080/arsys/pwa/#/forms/<serverName>/AttachmentFormPV/DefaultAdministratorView2_c**

Important

If there are any required fields in the remote form in the standard view, you must make sure that they are mapped to the form in Progressive View so that users are able to view them in PWA. If required fields exist and are not displayed, users will not be able to complete the form in PWA.

5. Attach a few files and click **Save**.
The attachments are saved and displayed.

Important

The size of the attachment that end users can upload is determined by the **Field > Max Size** property in the remote form. For more information about this property, see [Managing attachments in interfaces by creating attachment pools](#).

File:

Developing-Pwa\Organizing-Fields-By-Using-Collapsible-And-A

As a developer working on PWA screens, you can simplify the user experience by making use of collapsible or accordion panels to organize the options. This structure declutters the screen and helps users quickly find what they are looking for.

By using these panels, you can also make the screen more visually appealing and easier to navigate, which can lead to higher engagement and user satisfaction.

Related topic

[Flow-panel-holders-for-Progressive-Web-Applications](#)

[Panel-holder-display-types-in-Progressive-View](#)

[Creating-resizable-panes-in-Progressive-Web-Applications](#)

Scenario

At Apex Global, the Change Request screen has some fields that require general information such as change request location and the employee name. Seth, a developer working for Apex Global, organizes the options on this screen into different panels. The users can opt to open only those panels with fields to which they are adding information while closing the other panels. By using these panels, users can have a clutter-free and well-organized view when working with change requests.

The following animation shows an example where the fields are grouped into three panes within a Collapsible Panel Holder and displayed on a PWA screen:

General

Change request: location
Start typing... x

Requested for
Start typing... x

Summary
Start typing... x

Change location

Region
Start typing... x

Site
Start typing... x

Assignment

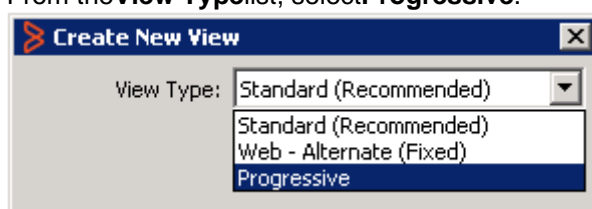
Change coordinator group
Start typing... x

Change manager
Start typing... x

To organize fields by using collapsible or accordion panels in Progressive Web Applications

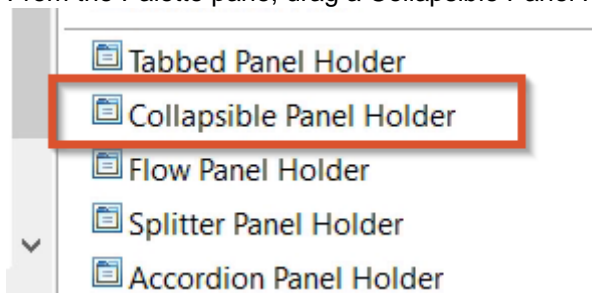
In Developer Studio, create a progressive view.

1. Select **Form > Create New View**.
The Create New View dialog box is displayed.
2. From the **View Typelist**, select **Progressive**.

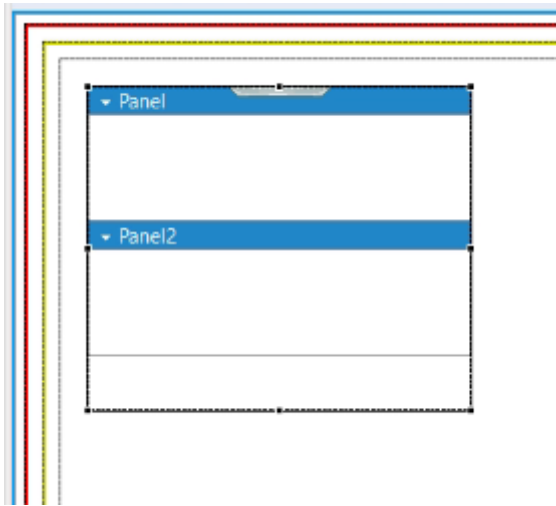


Developer Studio automatically creates a Section Panel.

2. Perform one of the following steps:
 - From the Palette pane, drag a Collapsible Panel Holder into the view.



The view shows a Collapsible Panel Holder with two panels within it.



- From the Palette pane, drag Accordion Panel Holder into the view.
Accordion panels are similar to collapsible panels except that at least one panel is always open.

3. Click **Save**.

4. Open the Mid Tier configuration tool and sync the cache.

`http://<MidtierHostName>:<PortNumber>/arsys/shared/config/config_cache.jsp`

To test and review the panels in Progressive Web Applications

1. In a browser, log in to PWA.

(On-premises) **`http://<midtierHostName>:<portNumber>/arsys/pwa/#/login`**

(SaaS) **`http://<helixBaseURL>/arsys/pwa/#/login`**

2. Change the URL to include the Progressive View form that you created in Developer Studio.

(On-premises) **`http://<midtierHostName>:<portNumber>/arsys/pwa/#/forms/<AR_Server_Name>/<formName>/`**

(SaaS) **`http://<helixBaseURL>/arsys/pwa/#/forms/<AR_Server_Name>/<formName>/`**

4. Confirm that the collapsible and accordion panels hide or show options.

File:

Developing-Pwa\Panel-Holder-Display-Types-In-Progressive-Vi

As a developer, you must first build the application interface by using various basic elements such as panels and panel holders before you can add elements such as fields and buttons. Panel holders are basic structural elements you can create by using Developer Studio. The following display types are available for panels in a panel holder.

Tabbed

In a tabbed display, only one panel in the container is visible at a time. To view each panel, users click its tab or press the Tab key.

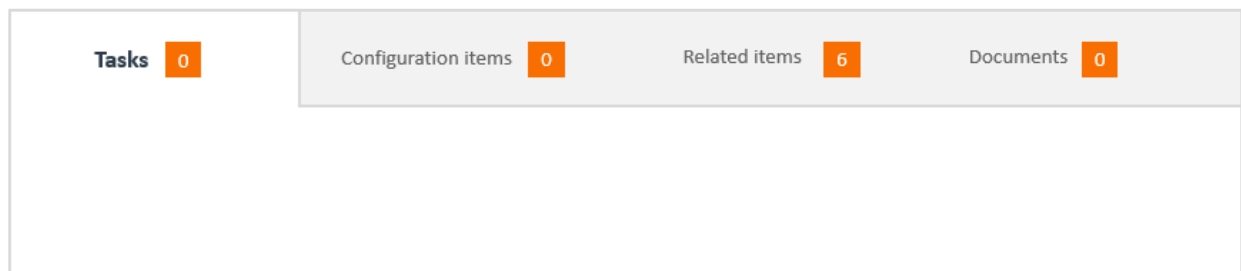
Only the horizontal orientation is available for tabbed panel holders. Tabbed panel holders do not have Border Thickness or Border Color properties.

Related topics

[Flow-panel-holders-for-Progressive-Web-Applications](#)

[Setting-panels-to-scroll-at-different-rates-in-Progressive-Web-Applications](#)

Tabbed panel holder



If a tabbed panel holder has more tabs than fit in one row, Developer Studio hides the remaining tabs. However, you can scroll the tabs to see them.

In browsers, multiple rows of tabs are not supported. Instead, if the number of tabs exceeds the width of the panel holder, left and right arrows appear. Users click the arrows to scroll to the tabs that extend beyond the width of the holder. The focus remains on the current panel when scrolling occurs.

Best practice

A performance impact might occur if the tabbed panel contains a table that is populated with large amount of data. We recommend using collapsible, or Accordion style panel.

Collapsible (stacked)

In a collapsible display, multiple panels in a panel holder can be viewed at the same time, either horizontally or vertically.

By default, if a collapsible panel holder contains space that is not allocated to a panel, the extra space is added to the bottom or right-most expanded panel in the holder. This prevents white space from appearing in the holder itself. In addition, you can control the allocation of extra space by assigning priorities to panels and by using the Fit to Content property. For more information, see [Avoiding scroll-bars-in-panels](#).

At design time, collapsible panels appear to have a fixed size. But at runtime, as various panels in the holder are expanded and collapsed by users or hidden and unhidden by workflow, the other panels in the holder shrink and expand to accommodate the changing space in the holder. This can create visual differences between panels at design time and runtime.

Collapsible panel holder

Panel 1

Field 1
 ✕ ▼

Field 2
 ✕ ▼

Field 3
 ✕ ▼

Panel 2

Field 4
 ✕ ▼

Field 5
 ✕ ▼

Accordion

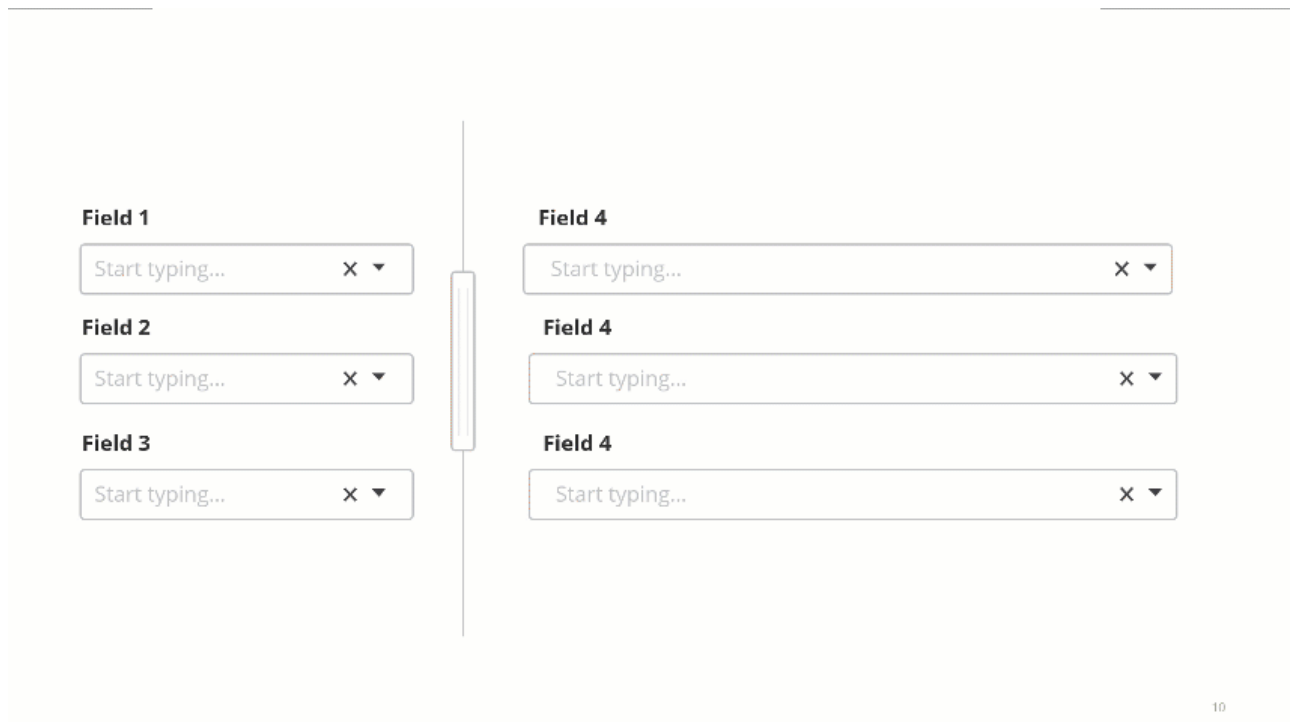
An accordion display is similar to the collapsible pane except that the content of one panel in the panel holder is always visible. Only the headers for the remaining panels are visible. Headers cannot be hidden in accordion displays.

At runtime, when you click a panel header, that panel is expanded to display its contents. If another panel

was open, it is automatically collapsed and only its header is visible.

Splitter

In a splitter display, multiple panels in a container can be viewed at the same time, either vertically or horizontally, and can be dynamically resized by dragging a splitter control on either side.



Flow panel holder for Progressive Web Applications

A Flow panel holder is available when you enable a Progressive view for your form. For more information, see [Progressive View quick start guide](#).

The Flow panel holder supports the properties required to enable a progressive view. Use a Flow Panel holder for aligning fields on a form. You can also group the fields to ensure correct alignment of fields on a form. This panel does not have a panel header, and supports Container panel holder (Default), Sub Panel holder, and Section Panel holder types.

The following video (4:52) demonstrates creating Fixed panels in Developer Studio when you enable progressive views for a form:

■ Watch Video: <https://www.youtube.com/watch?v=m33BWG-mEol>

▶ <https://youtu.be/m33BWG-mEol>

The following video (13:03) demonstrates creating complex layouts with flow panels:

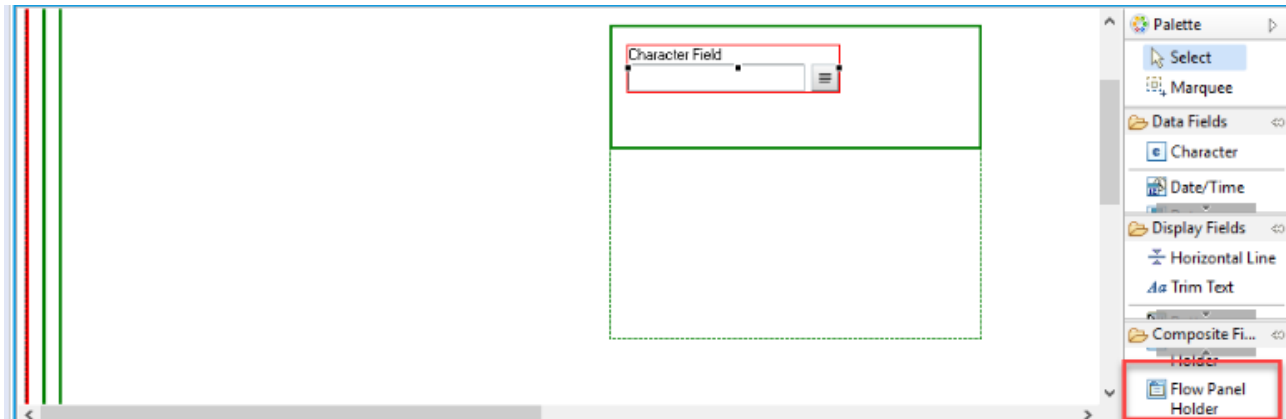
■ Watch Video: <https://www.youtube.com/watch?v=TnwUS2RF28Q>

▶ <https://youtu.be/TnwUS2RF28Q>

Panel arrangement sequence

The panels are arranged in the following sequence or hierarchy:

Flow Panel holder > Container Panel Holder > Container Panel > Sub Panel Holder > Sub Panel > Section Panel Holder > Section Panel > Section Panel Holder > Section Panel > fields

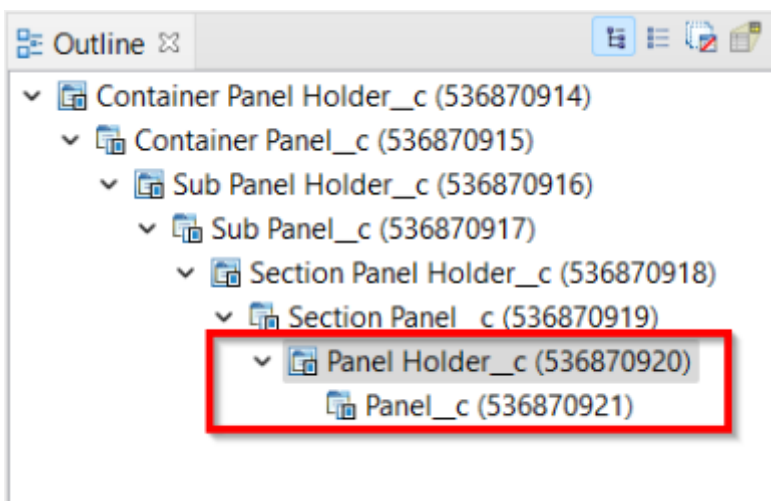


- A progressive view can hold only a Container type Flow panel holder and a Fixed panel holder.
- By default, a Flow panel holder has one panel as a child panel.
- You can change the read-only property type of any panel only when you change the property type of Flow panel holder.

Tip

When you add Flow Panel Holder to the Progressive View from the palette in Developer Studio, the flow panel holder is represented by a panel holder in the Outline pane.

In the following example figure, the flow panel holder is represented by **Panel Holder_c** enclosing a panel named **Panel_c**:



Container Flow panel holder

- By default, a Container type Flow panel is added to a Flow panel holder.
- You can delete a Container Flow panel only if you remove all progressive views.

- A Container Flow panel holder can have one or more Sub panels.

Section Flow panel holder

- You can add a Section Flow panel holder only under a Sub panel or Section panel.
- A Section panel can either have a Section type Flow panel holder or non-panel fields.
- The Section Flow panel holder added by framework contains only Section panel as children.

Sub panel holder

The Sub panel holder behaves as a Section panel on the desktop client. However, it is displayed as a tab on a mobile client.

- You can add a sub panel holder only under the container panel.
- A sub panel holder can have one or more sub panels.

For more information about developing PWA, see [Developing-Progressive-Web-Applications](#).

File:

Developing-Pwa\Rebranding-Progressive-Web-Application-Screens

You can rebrand Progressive Web Application (PWA) screens according to your company standards and requirements by using custom cascading style sheet (CSS) files and images.

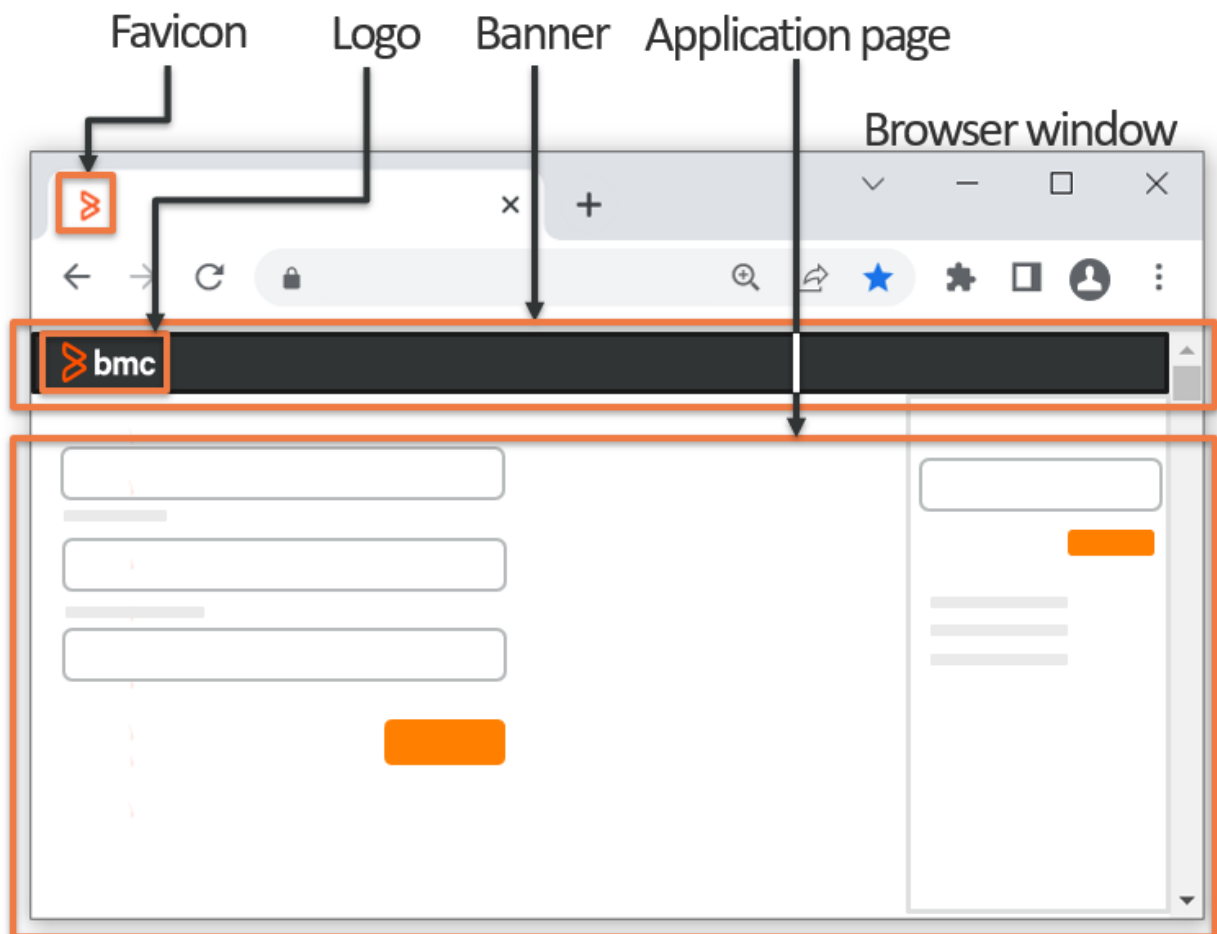
Important

If you are viewing the PWA screens from Smart IT, then the banner, logo, and favicon are from within Smart IT and so you must use the rebranding utility in classic Smart IT to change the banner, logo, and favicon. To learn more, see [Rebranding Smart IT on the Universal Client](#).

Related topics

[Developing-Progressive-Web-Applications](#)

[Rebranding Smart IT on the Universal Client](#)



Before you begin

Before you begin the rebranding process, you must ensure the following:

- (for on-premises deployments) You must have **AR Administrator**, **groupID:1** permissions to be able to access the PWA style configuration page.
- (for SaaS) You must submit a request to be able to access the PWA style configuration page.
- You must know how to work with or create a new CSS file.
- You must first set custom values and settings in a CSS file that you can then upload by using the rebranding config page.
- You must have a favicon file of the type ICO which you can upload from the rebranding config page.
- The custom CSS file must be named **custom.css** and the favicon file must be named **favicon_pwa.ico**. You can rename a JPEG or PNG file with the file extension of **.ico** to create a ICO file.
- In a multiple server environment, you must make sure that one of the servers is set as the home server or that the value for the home server should not be NULL in the Mid Tier configuration.

If you want to add a custom image, you must either convert the image to Base64 format or have a URL to the image location.

You can use free online tools such as [Base64.guru](https://base64.guru/) to convert an image to Base64 format.

For example, the following image and its Base64 format in plain text are shown:

Image in PNG




Image as plain text Base64

iVBORw0KGgoAAAANSUhEUgAAB4QAAAGOCAYAAAB/k772

To rebrand Progressive Web Application screens

The following steps show how to rebrand PWA screens by adding a customized CSS file and a favicon file.

1. Open the PWA style configuration page.
<MidTierURL>/arsys/pwa/#/config
2. Click the **Attach files** button.
The Open dialog appears.
3. Add one or both of the files, custom.css and favicon.ico.
4. Click **Upload**.
The files are added to the AR System Resource Definitions form.
5. Sync the Mid Tier cache to see the updated styles and favicon in the PWA screens.
<MidTierURL>/arsys/shared/config/config.jsp
6. (Optional) Click **Undo** to revert changes to out-of-the-box styles and favicon.
You must sync the Mid Tier cache to view the changes.
7. (Optional) If you want to revert any of the file to the out-of-the-box style, click **Undo**  beside the filename.
Click the **Undo** button to remove both the files.

Example CSS file

The following example CSS file shows the some of the customizations you can make. Download the file and open in a text editor to view or edit the content.

[Example_custom.css](#)(you must change the filename to custom.css before you can use this)

Example_custom.css

/* The following elements can be used to change styles in both Progressive Web Applications (PWA) and Smart IT PWA screens */

/* Set the text color, background color and border color for primary buttons */

```
.btn-primary{
color:#fff;
background-color:#ED1C24;
border-color:yellow;
font-family:monospace;
}
```

/* Set the text color, background color and border color for secondary buttons */

```
.btn-secondary{
/* color: #313538;
background-color: #f9f9fa;
border-color: #d5d5dc; */
```

```
color:blue;
background-color:#def5f2;
border-color:yellow;
}
```

/* Set the hyperlink color to green */

```
.a-hamburger,.btn-link{
font-weight:400;
/*color: #6ed588;*/
color:green;
}
```

/* specific color change for toolbar icons: assigntome, email, chat, follow, print, refresh, edit icons */

```
.d-icon-user_plus,.d-icon-envelope,.d-icon-send,.d-icon-star,.d-icon-printer,.d-icon-refresh,.d-icon-pencil
{
/* color: #ad3636; */
color:green;
}
```

/* this is to set anchor navigation color to blue in the create change screen*/

```
.anchor-navigator__link:hover>.anchor-navigator__link-title{
/* color: #1a1dec; */
color:blue;
}
```

/* You can change styles of the following elements only in standalone PWA and not in Smart IT PWA screens.

To change the style of the banner, favicon, logo, and header, you must use the use the Rebranding Utility described in the topic "Rebranding Smart IT on the Universal Client" in Smart IT documentation.*/

```
/* change the color of standalone PV signed in as link to red */
.a-profile-dark.menu-profile__signed{
color:red;
}__
```

```
/* change the color of standalone PV username link to red */
.menu-profile__username{
color:red;
}__
```

```
/* hide the banner completely */
/* adapt-navigation {
display:none;
} */
```

```
/* Set background color of banner as white */
.a-header{
min-width:320px;
border:0;
background:white;
}
```

```
/* replace ootb logo with a custom logo in base64 format*/
/*
```

```
.logo-helix.logo-light {
background-image:attach:xwiki:Service-Management.IS.IS.is254.Developing-Progressive-Web-Applications.Rebranding-F
url("data:image/svg+xml;base64,<insert base64 format>")
} */
```

```
/* replace ootb logo with a custom logo from a URL*/
/*
```

```
.logo-helix.logo-light {
background-image:attach:xwiki:Service-Management.IS.IS.is254.Developing-Progressive-Web-Applications.Rebranding-F
url("<insert image URL>")
} */
```

```
/* Hide the ootb product name header: "Progressive View" */
.a-product__name{
display:none;
}__
```

```
/* Set a custom header after the logo*/
.logo-helix:after{
padding-left:1rem;
position:absolute;
color:#ED1C24;
content:"Smart View";
font-style:italic;
font-weight:200
font-family:fantasy;
text-align:left;

}
```

Example output

Consider the example of changing the secondary buttons and hyperlink buttons to have values as show in the following sample code:

```
/* Set the text color, background color and border color for secondary buttons */
.btnsecondary{

color:black;
backgroundcolor: #C0C3C4;
bordercolor:#6F9850;
}

/* Set the hyperlink color to green */
.btnlink{
fontweight:400;
color:red;
}
```

Code

What it means

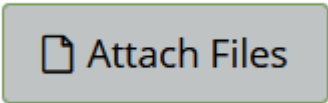
Output

For secondary buttons, the following style settings apply:

.btn-secondary{

color:black;
background-color:#C0C3C4;
border-color:#6F9850;

- The text color is black.
- Background color of the button is **#C0C3C4** or a shade of grey
- Border color of the button is **#6F9850** or a shade of green



.btn-link{
font-weight:400;
color:red;
}

For hyperlink buttons, the following style settings apply:

- The text color is set to red.
- The font weight is 400.

Browse

The updated styles are displayed as shown in the following image:

The screenshot displays the BMC Helix ITSM console interface. The top navigation bar includes the BMC Helix logo, 'ITSM', and various menu items like 'Dashboard', 'Console', 'Calendar', 'Smart Recorder', and 'More'. The main content area is divided into several sections for incident management. The 'Incident type (required)' section has a dropdown menu showing 'User Service Resto...'. The 'Service' section has a text input field. The 'Description' section has a large text area and an 'Attach Files' button. The 'Operational category' and 'Resolution category' sections each have a 'Browse' button and a 'Recommendations' link. The 'Product category' and 'Resolution product category' sections also have 'Browse' buttons and 'Recommendations' links. The 'Impacted areas' section includes fields for 'Company', 'Region', 'Site group', and 'Site'. At the bottom right, there are 'Cancel' and 'Save' buttons. The 'Attach Files' button and 'Browse' buttons are styled with a grey background and a green border, while the 'Recommendations' links are styled with red text and a font weight of 400.

File:

Developing-Pwa\Setting-Panels-To-Scroll-At-Different-Rates\Se

When panels on application screens have different quantities of content, a normal scroll bar creates blank spaces in the panels that have less content. A sticky panel helps avoid such blank spaces by scrolling the panel with more content at a different rate.

You can add sticky panels in Progressive Web Applications (PWA) screens with multiple panels by using Developer Studio. Adding sticky panels helps improve user experience when your users scroll screens with multiple panels displaying different amounts of content.

Sticky panels scroll in such a way that the panel with more content scrolls first so that scrolling of all the panels completes at the same time. Application users can see the content of the panels easily, without any of the content going out of view.

Scenario

At Apex Global, the change request screen has an activity panel which often has more scrollable content than the main panel. When users scroll down to view the activity panel content, the content in the main panel goes out of view.

The following animation shows a scenario where the panel on the left scrolls out of view leaving an empty space:

To address this issue, the application developer, Seth, adds a property in Developer Studio that causes both the panels to scroll at different rates based on their content. The panel with more content scrolls first while the panel with less content remains sticky, until both panels have the same amount of scrollable content.

The following animation shows how a sticky panel behaves when scrolling:

Related topics

[Panel-holder-display-types-in-Progressive-View](#)

[Flow-panel-holders-for-Progressive-Web-Applications](#)

[Navigation-fields](#)

Before you begin

You must start with an existing form that has more than one panel with the following properties:

Property	Value
Display > Display Type	Flow Panel
Display > Type	Section Panel
Display > Orientation	Vertical

For example, the PWA screen in Smart IT such as theSHR:SV_TicketDisplay formhas two flow panels with these properties.

To set panels to scroll at different rates in Progressive Web Applications

1. In Developer Studio, open the form for which you want to create sticky panels.
For example, open theSHR:SV_TicketDisplay form.
2. Select the Flow Panel Holder.
For example, in theSHR:SV_TicketDisplay form, select panels with IDs 304428481 and 304436081.
3. In Properties, under**Display > Custom CSS Style**,add the following string:
sticky-holder
If there are previously existing values in the Custom CSS Style field, the string**sticky-holder**must be preceded by a space.
4. (*Optional*) If the form has horizontal navigation fields, remove the following string from the**Custom CSS Style**property:
detach
5. Click**Save**and sync the Mid Tier cache.
For more information about how to sync the cache, see[Synchronizing-Mid-Tier-cache](#).

To validate the behavior of the panels in Progressive Web Applications

1. In a browser, log in to PWA.
<serverName>:8080/arsys/pwa/#/login
2. Change the URL to include the Progressive View form that you created in Developer Studio.
<serverName>:8080/arsys/pwa/#/forms/<serverName>/<formName>/<viewName>
The panels in the PWA show 'sticky' behavior whenever there is more content in one of the panels.

Troubleshooting

If the panels do not appear or if there is an issue with scrolling, perform the following steps:

- Confirm that the workflows and field properties are correctly defined in Developer Studio.
- If the page has horizontal navigation fields, remove the following string from the Custom CSS Style property:
detach

File:

Developing-Pwa\Setting-User-Preferences-And-Ccs-Parameters

As an AR administrator, you can set preferences for web clients and Progressive Web Applications (PWA) users by using the AR System User Preference form. You can make settings for each user such as logging, locale-based settings, and so on, for better control of AR System behavior as well as better user experience.

Related topics

[Setting-user-preferences](#)

[Developing-Progressive-Web-Applications](#)

[Setting up the time zone](#)

[Updating-configuration-settings-by-using-the-AR-System-Configuration-Generic-UI-form](#)

To set the user preferences for Progressive Web Applications

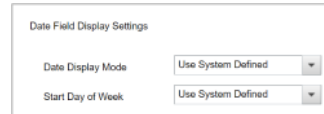
1. Log into Action Request System and open the AR System User Preference form.
Applications > AR System Administration > AR System Administration Console > User Preferences > My User Preferences.

Set any of the following PWA-related user preferences:

Tab	Area Name	Option	Description
Logging	Client	Active Links	Enables logging of active links on the client.
Server	API	Enables logging of APIs running on the server.	
Filter	Enables logging of filters on the server.		
Database	Enables logging of database activity.		

Progressive View

Date Field Display
Settings



Date Display Mode

- Use System Defined
- Absolute
- Relative

Use System

Defined displays the date in the format specified by the `system.pwa_date_field_display` parameter.

To learn more about this CCS parameter, see [Configuring date and time formats](#).

Absolute displays the full date and time when the record was created.

Relative displays the elapsed time since the record was created.

Start Day of Week

Sets the specified day as the first day of the week.

For a complete list of user preferences, see [Setting centralized preferences with the AR System User Preference form](#).

3. Click **Save**.

To set centralized configuration settings parameters for Progressive Web Applications

The centralized configuration forms store the configuration settings. You can modify the settings by using the **AR System Configuration Generic UI** form. The settings are case- and space-sensitive. For more information, see [Updating configuration settings by using the AR System Configuration Generic UI form](#). For the entire list of configuration settings that are used in AR System, see [Configuration settings](#).

Best practice

We recommend that you use the **AR System Configuration Generic UI** form to modify the configuration settings. Do not use the `ar.cfg` file to modify the configuration settings on the **AR System Configuration Generic UI** form.

1. In a browser, open the AR System Administration Console, and click **System > General > Centralized Configuration**.
2. In the AR System Configuration Generic UI form, from the **Component Name** list, select the component to which you want to add a setting.
3. Click **Add**.

Enter the name of the setting that you want to add and then enter its value.

The following table lists the PWA-related settings available in centralized configuration:

Setting	Description	Server group configuration	Maps to
arsystem.pv_menu_debounce_time (Component name: arsystem..arsys.midtier)	On progressive view enabled forms, determine the wait time in milliseconds for sending expand menu call after entering Autokey stroke in a menu. Default value: 500	Yes	AR System Administration Console > System > General > Centralized Configuration

Use the locale set in the User Preference form as the first priority for the forms with progressive views enabled. The language the application users see is based on this setting.

Valid values:

- true
- (Default) false

useARUserPreferenceFor

(Component type:
arsystem...arsys.midtier)

Yes

Example use case:
Apex Global wants to make sure all the employees see the same language in the Progressive Web Application irrespective of the geography they log in from. Set this parameter to **true** to set the language of the Progressive Web Application according to the setting for **User Locale** in the User Preference form. Therefore, Apex Global implements a uniform locale-based experience to all its employees.

AR System
Configuration Generic
UI form.

Use the time zone set in the User Preference form as the first priority for the forms with progressive view enabled. The time zone that the application users see is based on this setting.

Valid values:

- true
- (Default) false

useARUserPreferenceFor

(Component type: arsystem..arsys.midtier)

The screenshot shows the 'AR System User Preference' form with the 'Locale' tab selected. The 'Time Zone' field is highlighted with a red box. The form includes fields for Login Name, Short Description, Status, User Locale, Time Zone, Display Date/Time Style (WUT), Custom Date/Time Format (WUT), and Currency.

Yes

AR System Configuration Generic UI form.

Example use case:

Apex Global wants to make sure all the employees see the same time zone and time settings in the Progressive Web Application irrespective of the geography they log in from. Set this parameter to **true** to set the time zone of the Progressive Web Application according to the setting for **Time Zone** in the User Preference form.

Therefore, Apex Global implements a uniform experience to all its employees.

Use the date and time format set for **Display Date/Time Style (Web)** in the User Preference form as the first priority for the forms with Progressive View enabled. The date and time format that the application users see is based on this

useARUserPreferenceForDateTimeFormat setting.

(Component type:

(Component name

arsystem..arsys.midtier-Do

Yes

AR System
Configuration Generic
UI form.

Valid values:

- **Custom**
- **Short**
- **Long**

To learn more about the values, see [Web-client-preferences](#).

Enables display of a confirmation message when users remove an attachment on Progressive Web Application (PWA) screens.

pv.enable_attachment_on_delete_confirmation The default value for this option is **false**,

(Component

name:arsystem..arsys.midtier-Do

>arsystem..arsys.midtier-Do (display to user)

which means the application does not display a message when attachments are removed. As an administrator, you can set this option to **true** to enable confirmation messages when PWA users remove attachments.

Yes

AR System
Configuration Generic
UI form

Enables the display of a loader on PWA screens when you run the following command:

```
ENABLE-WAIT-CURSOR-ON-LONG-RUNNING-PROCESS
```

pv.show_transparent_loader

2

(Component name:
com..arsys.server)

Valid values:

No

AR System

Configuration Generic
UI form

- **true**(Default)
- **false**

To learn more about this command, see [Process commands](#).

pv.menu.in.table.supported

(Component name:
arsystem..arsys.midtier)

Enables menu attached fields and active links within cell-based tables on Progressive Web Applications (PWA) screens.

Yes

AR System

Configuration Generic
UI form.

Default value: False

For a complete list of configuration settings, see [Configuration settings](#).

5. Click **Apply**.
6. Click **Close**.

File:

Developing-Pwa\Ui-Layout-Of-Pwa\Ui-Layout-Of-Pwa.html

To create views for Progressive Web Applications (PWA), you must first understand that the positioning of panels containing UI elements in Progressive Views is based on the twelve-column layout explained in the following sections.

To design the views in Developer Studio, add Panels within Panel Holder, and then add fields to the Panels. You can also use nested panels and appropriately place fields within. Panels and Panel Holders create a layout for the Mid Tier.

Related topics

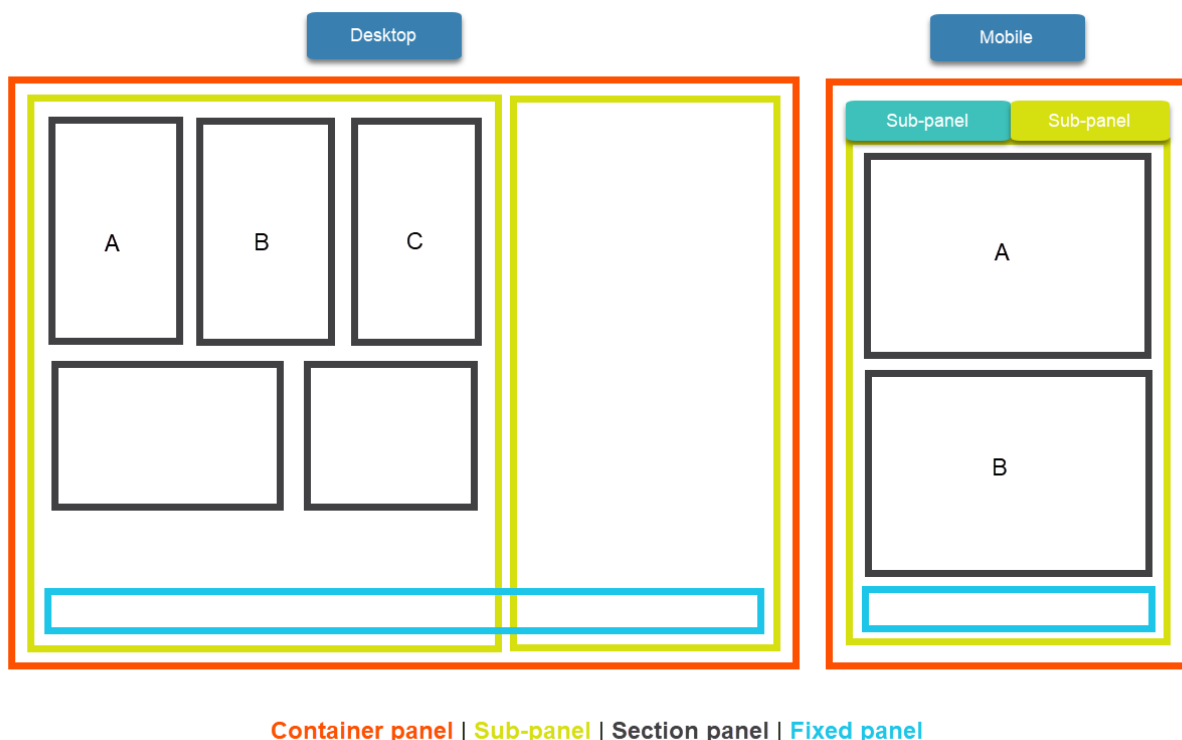
[Flow-panel-holders-for-Progressive-Web-Applications](#)

[Panel-holder-display-types-in-Progressive-View](#)

The following options are available for the Layout Style property:

- Container panel – Occupies the entire view port (100% screen width).
- Sub-panel – Holds sections. On mobile devices, occupies the entire width and is displayed as tabs.
- Section panel – Holds fields or nested sections.
- Fixed panel – Holds a section. Aligns to the bottom of a screen.

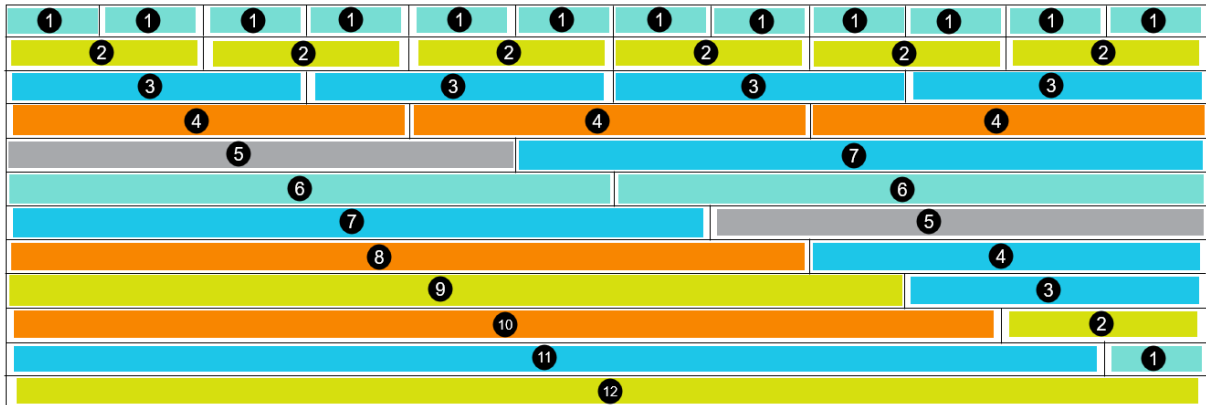
The following diagram illustrates a sample UI layout of Progressive Views:



The responsive design automatically adjusts the Panels.

Positioning elements on the screen

Use the twelve-column layout to position elements horizontally on the screen. The following image illustrates the twelve-column layout:



A developer can implement the preceding twelve-column layout by using developer-friendly properties that are added in Developer Studio. Refer to the following example:

To create a three-column layout, set the **Progressive Width** values as follows:

- Panel 1 - 4
- Panel 2 - 4
- Panel 3 - 4

These settings create a layout with orientations that vary based on screen size.

The illustrations in the following table show how the UI layout varies according to the device and screen size:

Device	UI layout
Desktops and tablet screens	

Mobiles screens



From the illustrations, you can see that PWA screens have the same layout in tablet views and desktop views. PWA screens display the mobile view only when you view the application screens on mobiles. As a result, you may see differences in how the application layout is displayed on tablets and desktops when compared to mobiles.