Content Services Toolkit

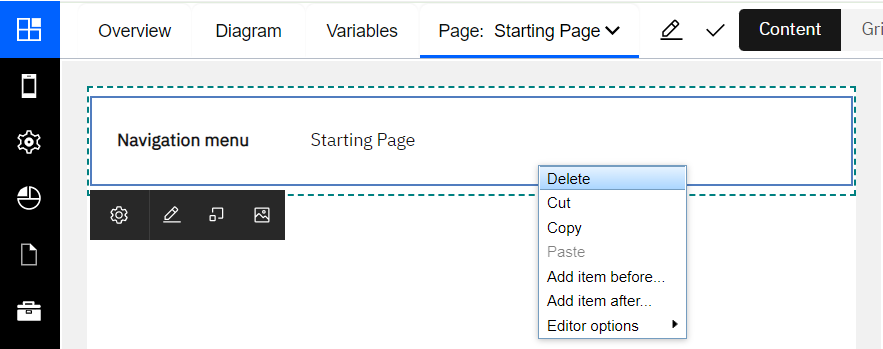
Updated: 10/6/2022

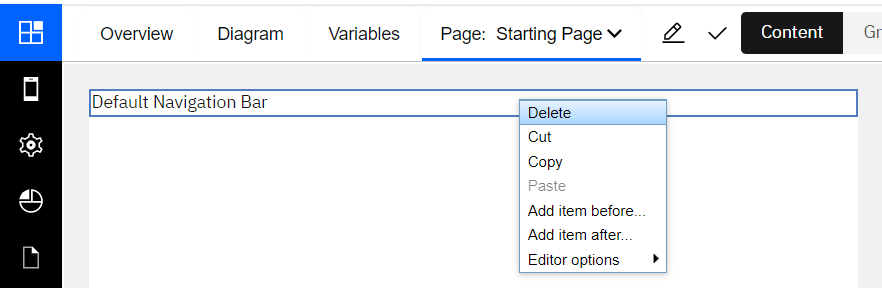
This document has the following sections:

1. [Create Your Application](#_1._Create_Your)
2. [Content List – Configuration](#_2._Content_List)
3. [Document Thumbnail - Configuration](#_3._Document_Thumbnail)
4. [Content Properties – Configuration](#_4._Content_Properties)
5. [Document Viewer – Configuration](#_5._Document_Viewer)
6. [Navigator - Configuration](#_7._Navigator_–)

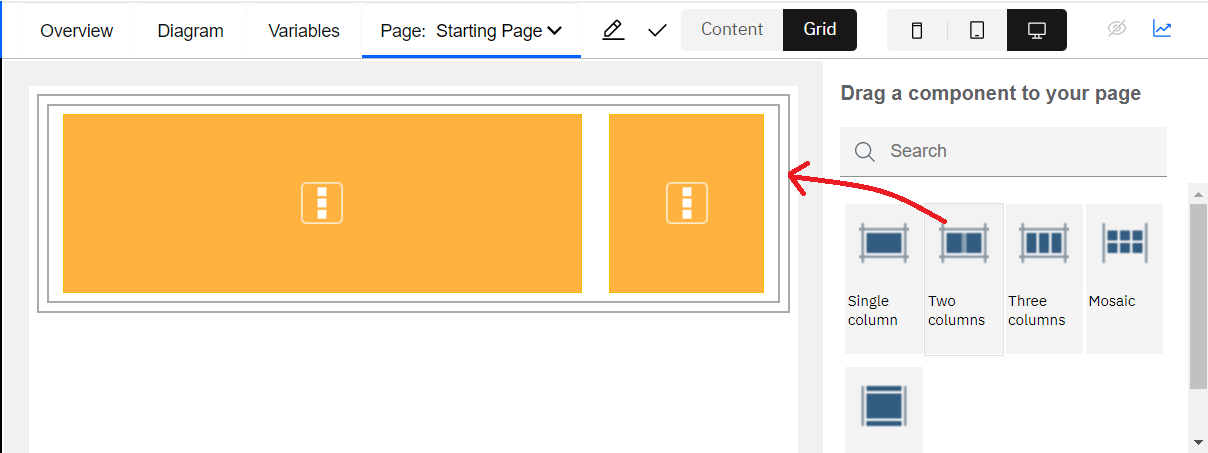
# 1. Create Your Application

Create your application. If present, feel free to delete the **Navigation menu Starting Page** and **Default Navigation Bar** (they are not necessary for this example).





Switch to “Grid” view and specify a two-column grid with the left side using a horizontal span of 9 and the right side a span of 3 (you can drag-n-drop the separator bar in between).

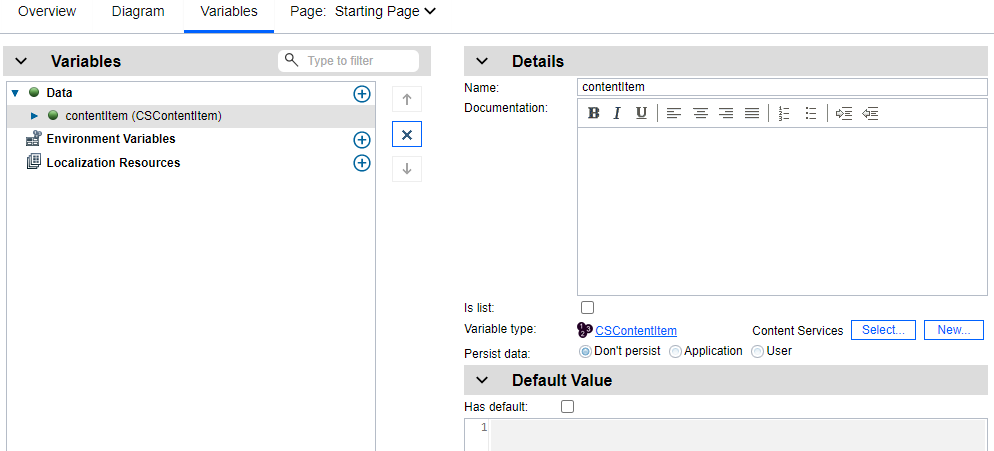


Switch back to “Content” view.

Add the Content Services toolkit to your application. Find “Toolkits” in the left panel and click the “+” icon to add the toolkit. Adding the toolkit enables the toolkit view components to be seen in the component palette on the right.

Open your application’s “Variables” tab

Create the “contentItem” variable, (Is list: false, Variable type: CSContentItem, click the “Select…” button to pick the CSContentItem variable type):

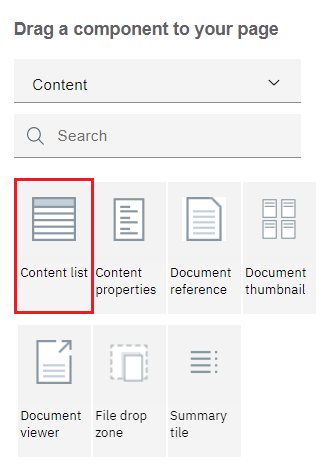


# 2. Content List - Configuration

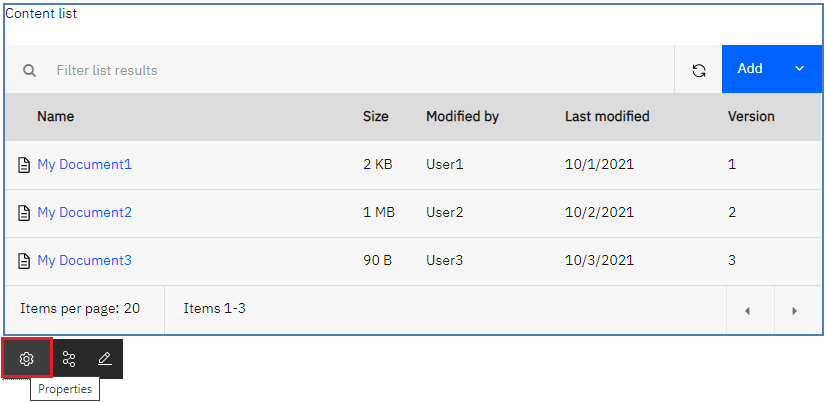
In the component selection palette on the right, select the “Content” component views and select the Content list component, drag and drop it into the left grid region in the canvas area.

NOTE: The content services toolkit provides several content components that can be used to build applications with:

* Content List – List view for displaying documents and folders
* Content Properties – Properties table for displaying document or folder properties
* Document Reference – Displays a document and can expose actions
* Document Thumbnail – Displays document thumbnail images
* Document Viewer – Displays documents (requires Navigator)
* File Drop Zone – For adding documents to a repository
* Plus, others…



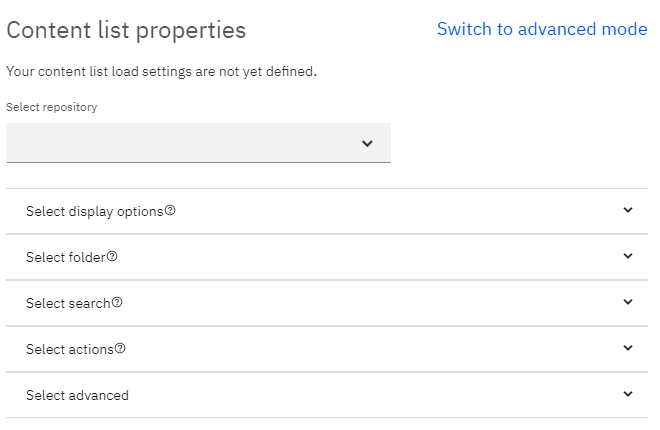
Select the Content list and open the properties dialog (left click on the component):



Click the gear icon (Properties).

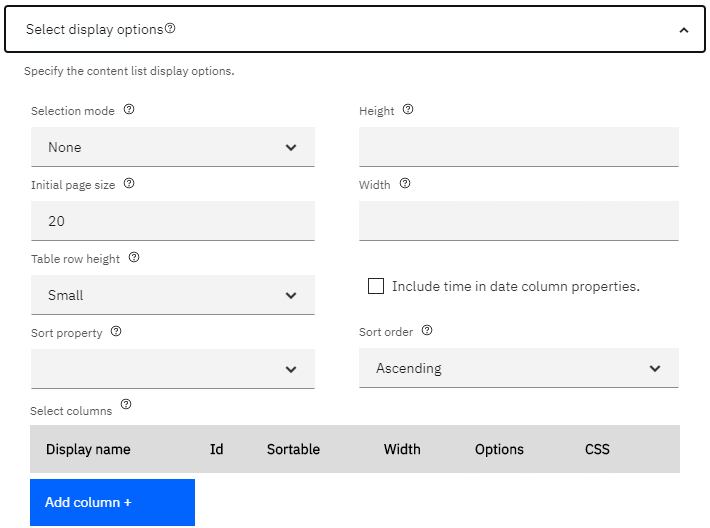
In the Select repository field, select the repository you want to use. This is a required setting.

NOTE: If your properties dialog opens into “Advanced mode”, select the “Switch to basic mode” link in the upper right.



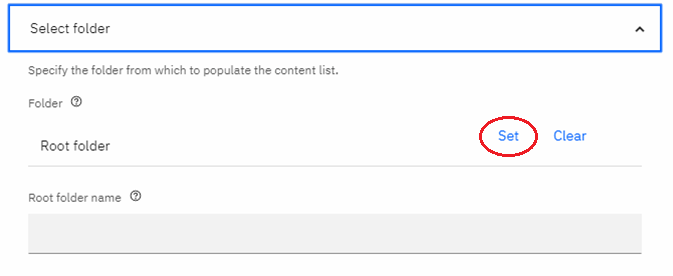
Open the “Select display options” section.

You can optionally enter new configuration values for several display settings here, a new paging size for the list view (default is 20 if not specified), specify that the time portion of dates be displayed in date column properties, define a new default column sort property and order, or define new column properties for the list view itself, plus many others… If no column properties are defined, the content list will display with its own default set.



If you want the content list to display folder contents, open the “Select folder” section and click the “Set” link to navigate and select the folder you want to populate the list view with.

You can optionally enter the Root folder name to display in the breadcrumb above the list (in case your folder does not have a user-friendly name).



To display search results, open the “Select search” section and enter the following settings (settings listed with ‘**\***’ are required). If you have previously configured a folder in the Select folder section above, you’ll need to delete that folder specification first (it has precedence over the search settings):

**Search in folder**: Folder you want to search in.

**Search only this folder**: Specifies whether to include searching sub-folders.

**Search for:** Specifies the kind of items you would like to search for (documents, folders or

abstract objects).

**Search type**: Specify the type of objects to return in the results.

**\*Search property**: Name of the property you want to search on.

**\*Operator**: Supported operator per property type and cardinality, such as Starts With, Ends With, Like, Not Like, =, <, >, <>, >=, <=, etc.

**Default value**: Value to search for. If no default value is entered, the list is initially displayed empty, and the user must type in the search field to initiate a query at runtime.

**Hide search field**: Can hide the search field at runtime. The editable flag can be used to lock down the query parameters if you do not want the user to change them.

**Text Options**: Specify a text search option (CBR), the default is not enabled.

**Enable search filter**: Boolean value to turn on the search filter capabilities. Will need to configure the search filter properties.

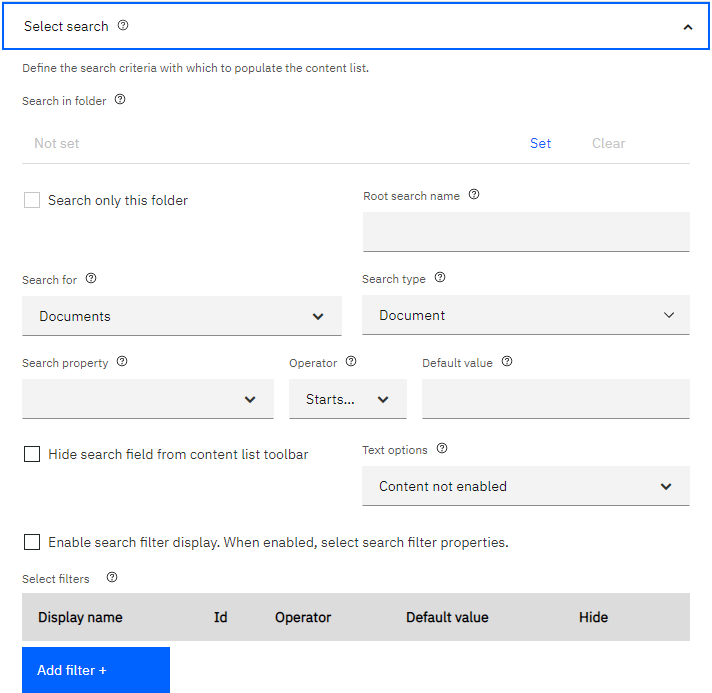
**Select filters:** Specify a set of properties you would like to filter the results on. The property

values can be modified at runtime by clicking the funnel icon on the toolbar . If property

filters have been specified, then the “Search property”, “Operator” , and “Default value” settings above are not used.

**Note the following behavior with properties that have choice lists**:

If the property has a choice list and the display values are different than the actual values, you must type in the actual value. There currently isn’t any mechanism to resolve the display name to the actual value.

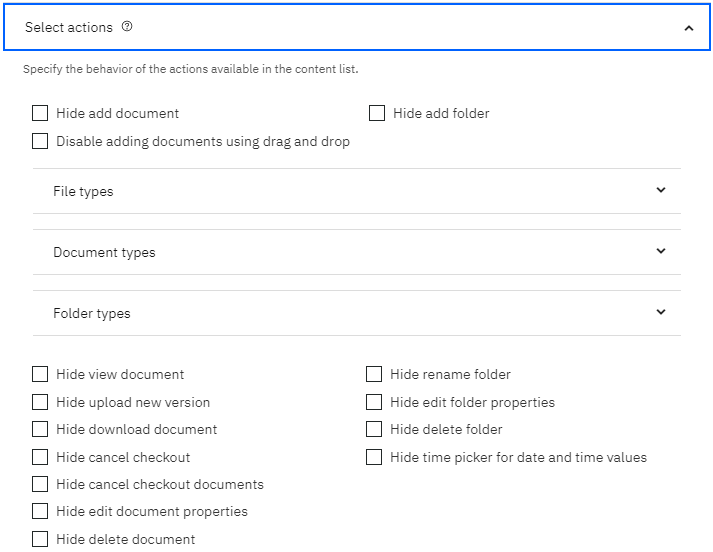


Open the “Select actions” section.

These settings allow the user to hide any of the listed actions from the context menu.

Users can also specify a list of classes that will be exposed from the Add Document and Add Folder modal dialogs (instead of retrieving the entire list from the repository), and which file types are accepted when adding documents.

Nothing needs to be set here.



Click OK to close the Properties dialog.

Save your settings (button on application banner bar):



Run your application in preview mode (Preview link in upper far right of the page). You should see the list view rendered with the results from the folder or search you specified.

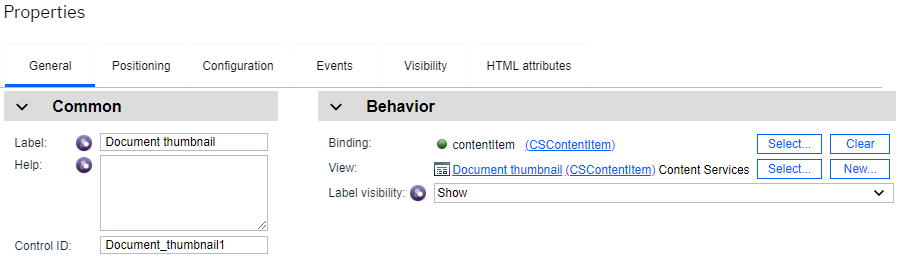
# 3. Document Thumbnail – Configuration

Select the Document Thumbnail component from the palette, place it in the smaller grid region on the right and open its properties dialog.

NOTE: For the Thumbnail component to render real document thumbnails, the Thumbnail AddOn must be installed into the Content Engine repository (object store). This AddOn generates the actual thumbnail images when documents are uploaded to the repository. If no thumbnail has been generated, the mime type image will be displayed instead. Depending on the load on the system, the thumbnail image may not be generated for a moment or two.

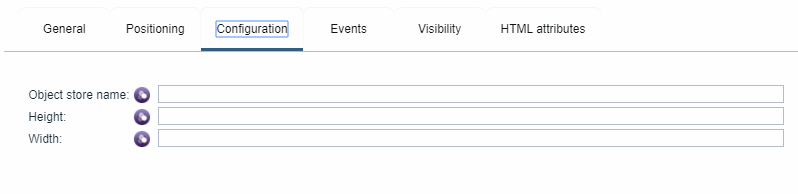
On the “General” tab, select the “Binding” property to be the variable called “contentItem”. You created this variable above as part of the Content list configuration.

NOTE: Opening the thumbnail components properties dialog will place the user in the “advanced mode” of properties display. The thumbnail component does not have a newer “view configurator” for the presentation of the configuration settings.



Next, switch to the “Configuration” tab.

In the Object store name field, enter the symbolic name of the Content Engine repository.

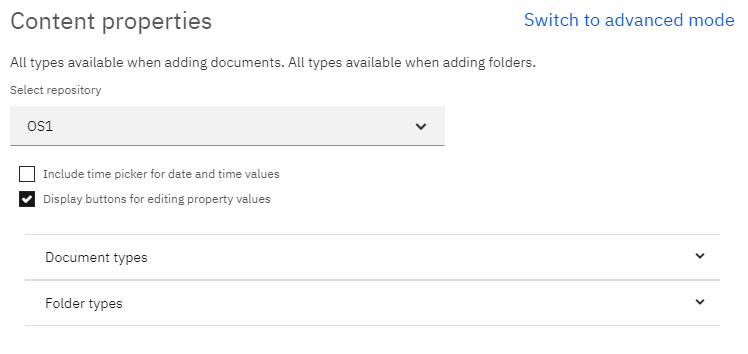


# 4. Content Properties – Configuration

Select the Content Properties component from the palette, place it in the smaller grid region on the right below the Thumbnail and open the properties dialog.

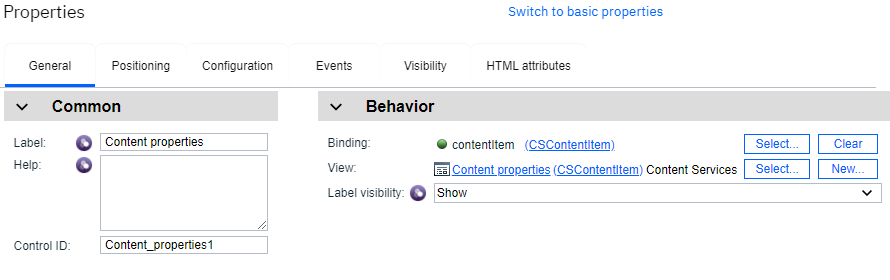
In the Select repository field, select the name of the Content Engine repository.

Select the Display buttons for editing property values if you want to edit the properties at runtime from this panel.



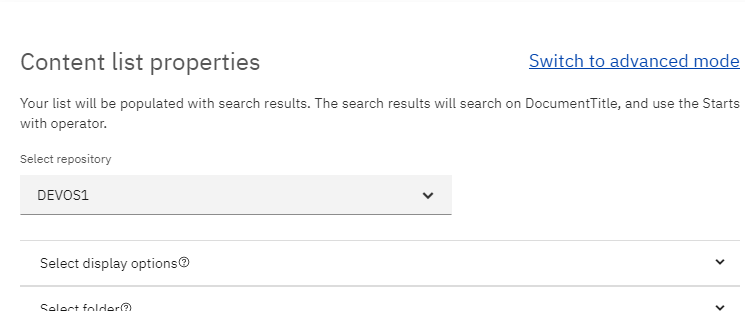
Select the “Switch to advanced mode” link.

On the “General” tab, select the “Binding” property to be the variable called “contentItem”. This variable you created above as part of the Content list configuration.



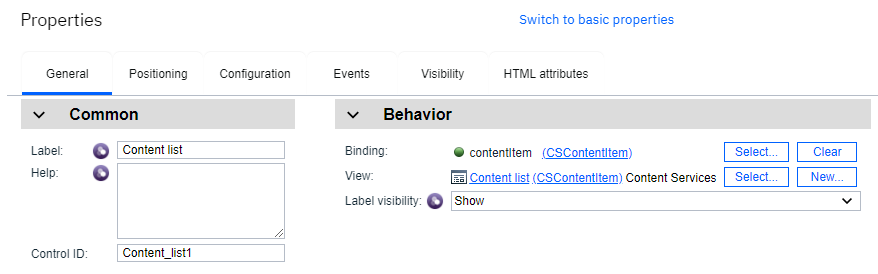
Click OK to close the Properties dialog.

Select the Content list and open its properties dialog, select the “Switch to advanced mode” link.



On the “General” tab, select the “Binding” property to be the variable called “contentItem”. This variable you created above as part of the Content list configuration.

Both the Content List, Thumbnail and the Content Properties view components must be bound to the same application-based variable to enable the change event mechanism to work correctly between the views. The change event mechanism works to update each view component when something in the bound contentItem object changes.



Close the Properties dialog and click the save button on the banner bar.

Run your application in Preview mode. You should see the list view rendered and when you select a document from the list view, the thumbnail and property list components will display the image and properties for the selected item.

# 5. Document Viewer – Configuration

NOTE: The Document viewer component requires that your application be running inside Navigator with a desktop configured for “Platform and Content” mode. Authentication should be set to the repository. Otherwise, the user will need to manually log into the repository on the very first loading of the document.

* Desktop Type: Platform and content desktop
* Authentication: Connection=OS1
* Connections: APPENGO, OS1

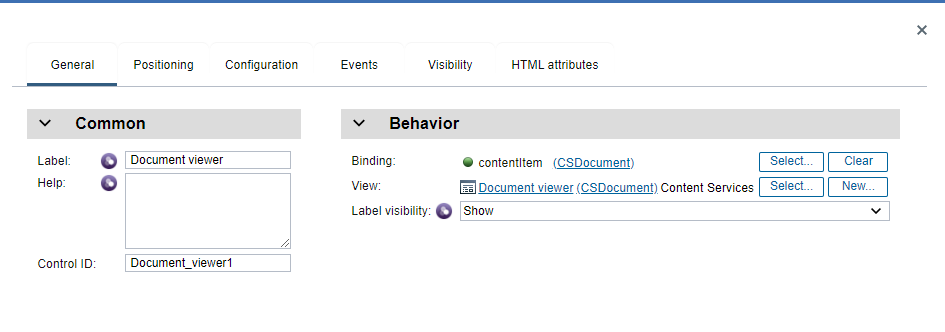
In the example above OS1 is the repository.

This component uses Navigator’s bookmark.jsp URL running inside an iframe that points back to Navigator (it will utilize any viewer map configuration you’ve specified in your desktop). Complete this configuration first, then go on to the following section to setup the viewer to run inside your navigator cloud application.

Select the Document Viewer component from the palette, place below the list view (or wherever you want) and open its properties dialog.

On the “General” tab, select the “Binding” property to be the variable called “contentItem”. This is the variable you created above as part of the Content List configuration.

On the “Configuration” tab set the Object store name to the same setting as your other components. The Viewer component has a default size of 600px wide by 480px high. You can set the Width and Height to a reasonable size for your page layout.



Close the dialog and save your settings and run Preview again…

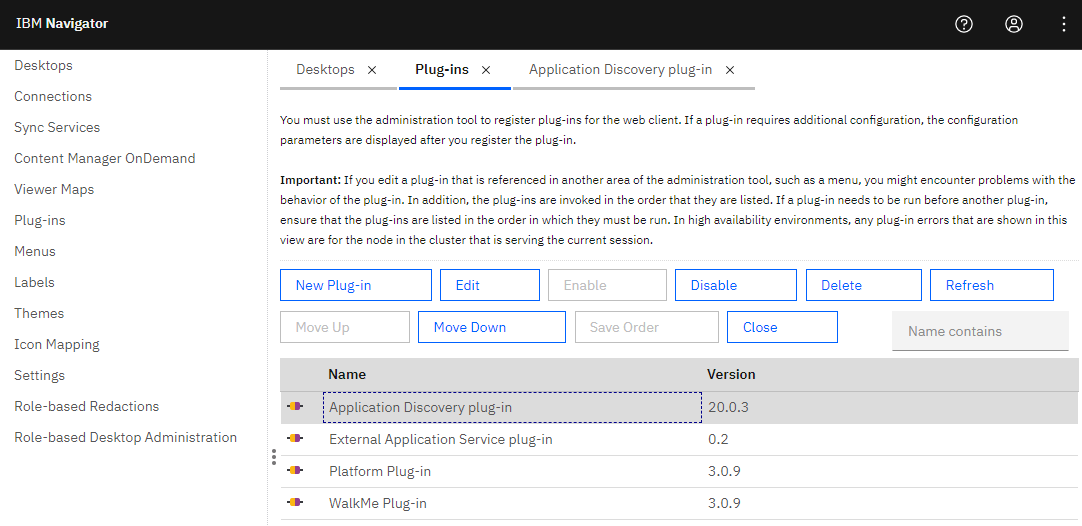
# 6. Navigator – Configuration

Ensure that the platform plugin jar is installed, and the plugin is enabled into your Navigator environment. The plugin jar file name is:

application-discovery.jar

Run Navigator and login into Administration

Open the Plug-ins feature view and verify the Application Discovery plug-in is running.

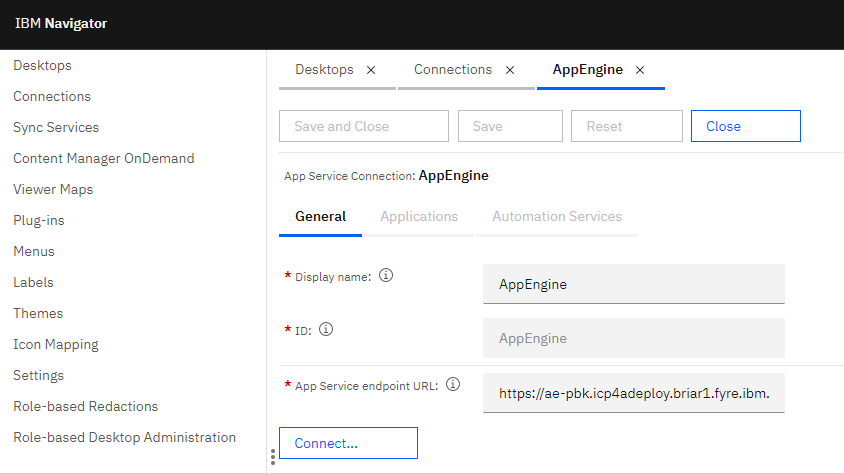


Under “Connections” feature view, click New Connection button, and give it a name:

“AppEngine”

Type in App Service endpoint URL to your app server.

Click Connect…

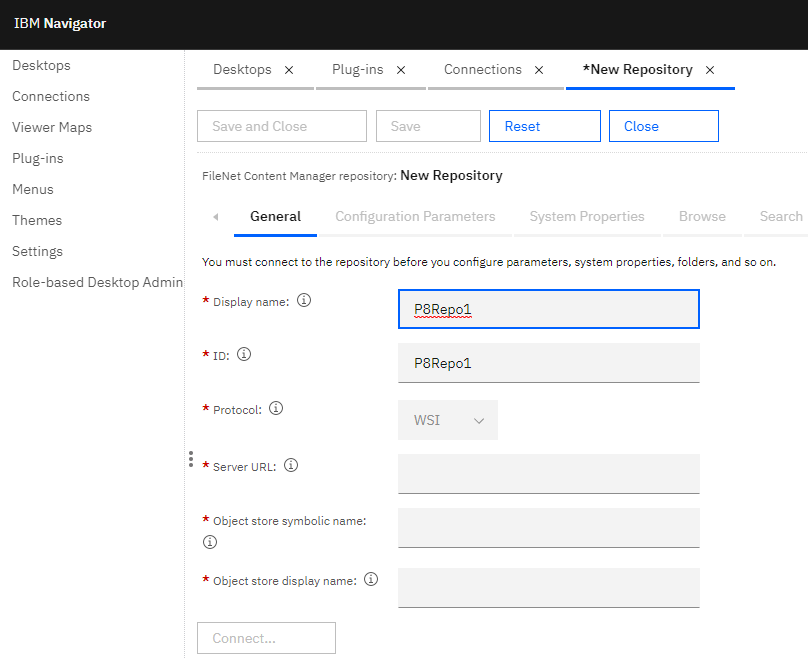


You must have a P8 repository created that will be selected into your desktop.

Switch to the “Repositories” tab on the Connections feature view and select “New Repository”.

Enter your repository information, you may need to run the ACCE admin tool to gather the object store information etc.

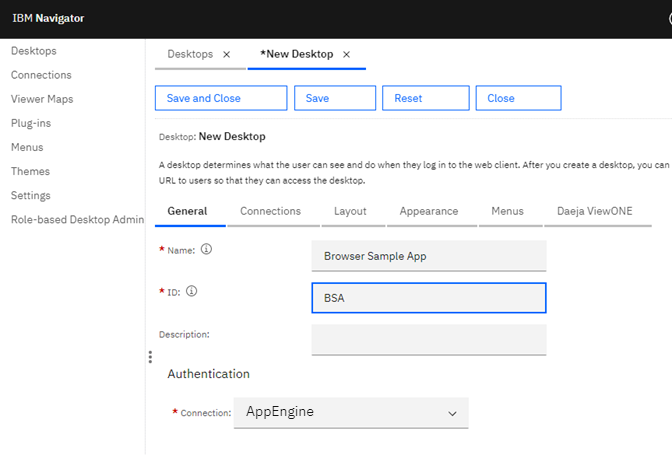
Ensure that you can connect by clicking the Connect… button.



Open the Desktops feature view and create a new desktop.

Point the Authentication “Connection” of your desktop to the connection you just created.

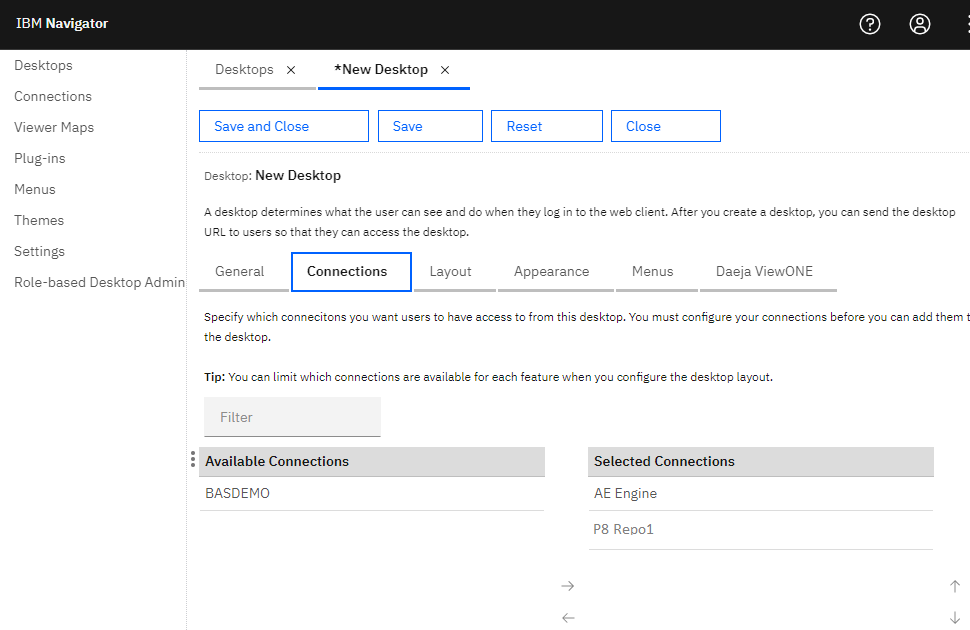
You can also optionally specify a viewer map here too… if you would like to specify a specific viewer.



Click on the “Connections” tab.

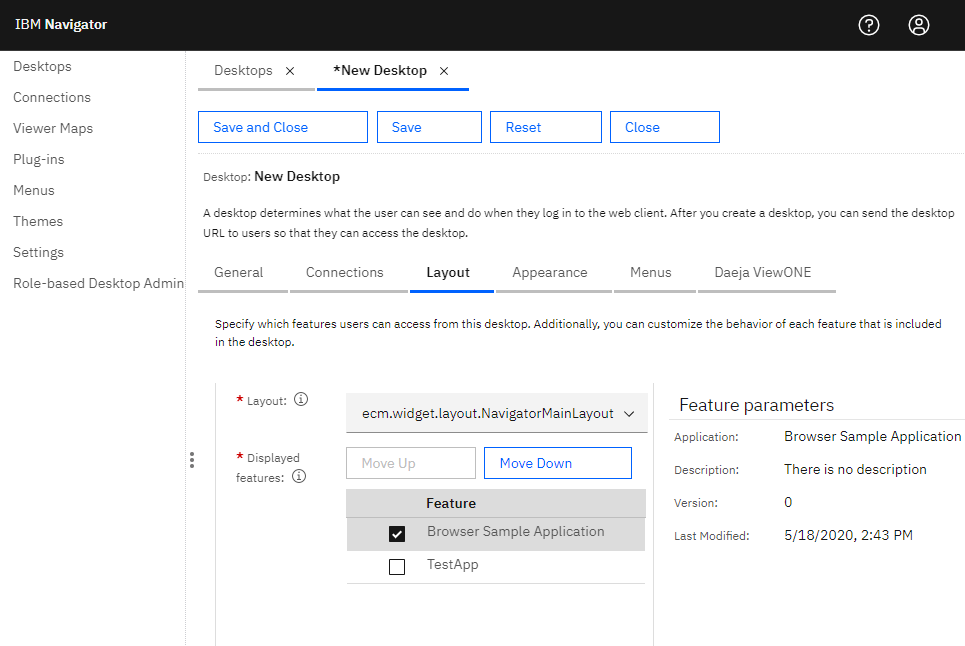
Ensure the App Engine connection is already selected (on right)

Select the Repository where your content will be located, you should have both a connection and repository (a repository is required for the viewer component to work correctly):



Finally, select the “Layout” tab

In the Desktop Features selection area, select your BAStudio application (Browse Sample Application in this screen shot) as the feature. You can also have other Navigator features enabled if you are running in “Platform and Content” mode.



Save and login to Navigator using your newly created desktop… Enjoy 😊