



Designing the Layout of External Content Using the Widgets Tool

First Published: April 21, 2014

Revised: October 16, 2014

This module describes how to design the layout for display of the data that you have integrated and mapped for output in Cisco StadiumVision Director using the External Content Integration feature. This layout is required to create a content object referred to as a *widget*, that you can then publish with the standard features of Cisco StadiumVision Director using playlists and scripts for eventual display on TVs.

This module includes the following topics:

- [Prerequisites for Using the Widgets Tool, page 61](#)
- [Restrictions for Using the Widgets Tool, page 62](#)
- [Information About Using the Widgets Tool, page 62](#)
- [How to Design the Layout of External Content for Display, page 74](#)
- [Configuration Examples, page 85](#)
- [What to Do Next, page 115](#)

Prerequisites for Using the Widgets Tool

Before you design the layout of external content, be sure that the following requirements are met:

- You have configured the external data sources and mapped the statistics to output fields in the External Content Integration feature.
- The data from the external source is being properly ingested by Cisco StadiumVision Director.
- You have added any standalone graphics that you want to add to the layout by importing them into Cisco StadiumVision Director from **Control Panel > Content > Import**.

Restrictions for Using the Widgets Tool

Before you design the layout of external content using the Widgets tool, consider the following restrictions:

- The Widgets layout tool can only be used for data sources that have been integrated into Cisco StadiumVision Director using the External Content Integration feature.
- Cisco StadiumVision Director only supports widgets that are created internally using the Widgets tool.
- Up to 16 different sessions can be supported in the Widgets tool on a single computer.
- The maximum graphic size supported is 1920x1080, which is also the maximum size of the area supported in the Widgets layout tool.



Note If you attempt to use a larger graphic in the Widgets tool, it might not be displayed.

- The PicToScreen component has the following restrictions:
 - A maximum of 5 images can be rotated using the PicToScreen component. Therefore, up to the first 5 images received from the data feed can be displayed. A new set of images can be retrieved at the next polling interval.
 - Supports binding of image data fields only from a source feed. Therefore, the data source must be defined with an image field for output display. For more information on field mapping in the data source, see the [“Selecting Input Statistics and Mapping to Output Fields for Display” section on page 42.](#)
- The TextToScreen component supports binding of text data fields from a source feed. Therefore, the data source must be defined with a title, description, or other text data field for output display.
- The Widgets tool has the following font restrictions:
 - Only the Arial font type with the ASCII character set is supported.



Note While you might be able to see a supported localized language and create text components with non-ASCII characters in the Widgets tool, these text components might not be rendered as expected on the DMP.

- The maximum font size is limited to 128 due to a restriction by the DMP.
 - Importing of fonts is not supported.
- The following functions are not supported in the Widgets tool:
 - Undo
 - Deleting a group of multi-selected components

Information About Using the Widgets Tool

This section includes the following topics:

- [Components, page 63](#)
- [Layers, page 66](#)

- [Properties and Effects, page 67](#)
- [Data Binding and Preview, page 73](#)
- [User Interface Characteristics, page 74](#)

Components

The Widgets tool includes image-based and text-based components, and a tabular list component. There is also a special Data Pull component for certain environments where a change in the way data is retrieved and processed by Cisco StadiumVision Director is needed.

New components are added to the widget area simply by dragging the object type and dropping it onto the canvas or by double-clicking the component.

The Widgets tool includes the following components:

- [Standalone Text and Graphic Components, page 63](#)
- [Data Feed Components, page 63](#)
- [List Component, page 65](#)
- [Data Pull Component, page 66](#)

Standalone Text and Graphic Components

Two component types are intended for data with standalone text or graphics:

- Text Area
- Graphic

The Widgets tool uses the Graphic component to support the addition of standalone images. Each time a graphic component is added to a widget, it is added in numerical order as a Layer on the canvas.

Data Feed Components

Cisco StadiumVision Director Release 3.1 introduced two new component types to support generic data sources and feeds, whose content is either image-based or text-based.

P2S (PicToScreen)

The PicToScreen component supports only image data fields from the data source, and can display up to 5 images from a data feed. The images appear in a fixed overlapping cascade that rotates through the available images from the feed ([Figure 1](#)).



Note The cascading layout of the images cannot be changed. However, you can alter the position and orientation of the top graphic within the Widgets tool.

Figure 1 Example of PicToScreen Widget Display of 5 Maximum Images



Default Images and Display Behavior

Cisco StadiumVision Director has default images that are used as placeholders for any images that cannot be found or when the number of images received does not equal the number defined in the “Images to Display” property of the PicToScreen component (up to 5).

If Cisco StadiumVision Director must insert default images when enough images are not received from the data feed, then only the good images from the data feed are rotated through the cascading presentation. The default images will remain at the back of the stack in the presentation.

The default image is called DefaultImg.jpg and is found in the content library. You can change the default image by uploading your own image into the content library and change the “ContentName” property in the P2S component.

Image Orientation

You can change the position and rotation of the top graphic in the PicToScreen component. However, you cannot control the position or rotation of individual images within the stack.

T2S (TextToScreen)

The TextToScreen component can support any number of messages from the data feed.

[Figure 3](#) shows an example of a fixed background graphic component, with a TextToScreen component overlay in [Figure 3](#).

Figure 2 Example of TextToScreen Widget Background Graphic



Figure 3 Example of TextToScreen Widget Display



List Component

Cisco StadiumVision Director Release 3.2 introduces the List component, whose content is either image-based or text-based.

The List component can be used to create tables by grouping individual list components side-by-side either vertically or horizontally. The default format for the List component is a horizontal row of two cells. By changing the orientation to a vertical layout, the component becomes a single column of cells.

You can change the number of items in the list by changing the number in the Cells property and each list can be rendered as text or images in the List Render option, among other properties.

Figure 4 shows an example of a widget created for a transit schedule.

Figure 4 Example of Schedule Widget Using List Components



In this example, there is a Graphic component with a background image at the first layer of the widget (Figure 5). Notice that the route names are built into the fixed graphic on the left side, based on knowledge of the XML content in the data source for the widget, so that the data bound in the adjacent List components correlates.

Figure 5 Example of Schedule Widget Background Graphic



The remainder of the schedule widget uses five single-column list components each containing seven cells/items in a vertical orientation (Figure 6). These lists display dynamic content from a single data source through both field mapping and configuration of the expression editor.

For more information about using List components in a widget, see the “Configuration Examples” section on page 85.

Figure 6 Example of Schedule Widget Using List Components

City Zone	2:12	2:42	3:12	Good Service
Babylon	2:12	2:42	3:12	Good Service
Far Rockaway*	2:12	3:12	4:12	Good Service
Hempstead	2:42	3:42	4:42	Good Service
Long Beach	2:42	3:42	4:42	Service Change
Montauk	2:12	2:42	3:12	Good Service
Oyster Bay*	2:12	4:12	4:42	Good Service

Data Pull Component

The Data Pull component is a special component that is only needed in a widget for certain data sources larger than 16 KB and whose data does not need more immediate synchronization across large numbers (100-200) of DMPs (such as for menu board data).

The Data Pull component changes how the data source is processed by Cisco StadiumVision Director by pulling the data feed and sending it to the Flash template, which overcomes the 16 KB multicast packet size normally pushed to the DMP.

In widgets without a data pull component, only specific output fields from the data source are sent using multicast. Widgets with a data pull component pull all output fields in the data source.

Layers

Similar to standard graphics editing applications, each time a Text Area or Graphic component is added to a widget, it is labeled in numerical order (by type) as a *Layer* on the canvas. Unless reordered, the first component added is at the bottom of the components added to the canvas, and the last component added is the topmost layer. In the user interface, the Layers box also lists the layers in stacking order with the topmost layer at the top of the list. In general, a background image should be at the bottom of the list.

You can reorder the layers by dragging and dropping them into a different order in the Layers box of the Widget tool.

The first graphic component added to the widget is identified as “graphic 1,” and the first text area is labeled “text area 1.” Components are numbered sequentially as they are added. Therefore, if you delete a component, that component number is not replaced.



Tip

You can rename a component by double-clicking it.

Properties and Effects

The Properties panel allows you to set the characteristics of a component including explicitly setting its size, location, and rotation on the canvas.

Depending on the component type, some additional Properties and Effects are available:

- [Data Pull Component Properties, page 67](#)
- [List Component Properties and Effects, page 67](#)
- [Standalone Text Area Component Properties, page 69](#)
- [Standalone Graphic Component Properties, page 70](#)
- [PicToScreen Component Properties and Effects, page 71](#)
- [TextToScreen Component Properties and Effects, page 72](#)

Data Pull Component Properties

[Table 1](#) describes the property option available for Data Pull component. You can either pull data one time, or you can set an interval of time to repeat pulling data from the data source.



Note

Although there are dimensional properties for the Data Pull component in the widget, they really do not need to be set.

Table 1 *Data Pull Component Property*

Property Name	Description	Values
Pull Once	Gets data from the data source one time and sends it to the DMP Flash template data registry	Default is disabled.
Pull Interval (seconds)	Length of time to wait before pulling the data source from Cisco StadiumVision Director and send it to the DMP Flash template data registry.	Range is 15-3000. Default is 60.

List Component Properties and Effects

[Table 2](#) describes the property options that can be applied to List components.



Tip

Due to the number of the Properties and Effects options for the List component, you might need to use the resize handle at the top of the panel and/or the scroll bar to see all options.

Table 2 *List Component Properties*

Property Name	Description	Values
x	Number of pixels that defines the starting position of the component horizontally.	Range is 0–1920.
y	Number of pixels that defines the starting position of the component vertically.	Range is 0–1080.

Table 2 List Component Properties (continued)

Property Name	Description	Values
width	Number of pixels that defines the horizontal width of the component.	Range is 0–1920.
height	Number of pixels that defines the vertical height of the component.	Range is 0–1080.
rotation	Number of degrees to rotate the component box.	Range is 0–360. Default is 0.
Cells	Number of cells defined in the list.	Default is 2.
Cell Width	Number of pixels that defines the horizontal width of the cell.	Range is 10–1080. Default is 200.
Cell Height	Number of pixels that defines the height of the cell.	Range is 10–1080. Default is 100.
Vertical align	Vertical alignment of content in the cell.	Possible values are: <ul style="list-style-type: none"> • Top (Default) • Middle • Bottom
Horizontal align	Horizontal alignment of content in the cell.	Possible values are: <ul style="list-style-type: none"> • Left (Default) • Center • Right
Grid	Option to control display of cell borders.	Enabled (checkbox is marked)
Cell Padding	Number of pixels that defines the padding between cells.	Range is 1–20. Default is 2.
List Order	Alignment of the list component as either a row (Horizontal) or column (Vertical).	Possible values are: <ul style="list-style-type: none"> • Horizontal (Default) • Vertical
List Render	Rendition type for list component items as either text or images.	Possible values are: <ul style="list-style-type: none"> • Text (Default) • Image
Font Name	Font style for the list. Note Cisco StadiumVision Director only provides the Arial font by default. To support additional fonts, you must install them using the Software Manager.	Possible values are: <ul style="list-style-type: none"> • Arial (Default) • Available custom fonts.

Table 2 List Component Properties (continued)

Property Name	Description	Values
Font Size	Size of the font for text-based list components.	Default is 36. Maximum is 128.
Font Color	Color of the font for text-based list components. Tip Remember that the Widgets tool displays a white background for ease of use, but if you are not using any background images, a black font will <i>not</i> be visible on the TV screen because the TV background is actually black.	Hexadecimal value of the color or by selection in the color grid. Black is the default.

Table 3 describes the effects that can be applied to List components.

Table 3 List Component Effects

Effect Name	Description	Values
Background Color Condition	Background color for the cell property when there is an increase/decrease of the cell value (for a numeric value) or upon change of the value.	Possible values are: <ul style="list-style-type: none"> None (Default) Increase and Decrease (Number) On Change
Background color display duration (Seconds)	Length of time to display background color of the cell property.	Default is 0.
Format	Format to display for list items including number formats, date formats, and time formats.	Do Not Format (Default)
Prefix Symbol (max 3 Characters)	Character string to prepend to the list item.	Maximum of 3 characters.
Suffix Symbol (max 3 Characters)	Character string to append to the list item.	Maximum of 3 characters.

Standalone Text Area Component Properties

In addition to setting the size, location, and rotation on the canvas for Text Area components, you also can specify the character string for the component, and select a font size and color.

Table 4 describes the property options that can be applied to standalone Text Area components.

Table 4 Text Area Component Properties

Property Name	Description	Values
x	Number of pixels that defines the starting position of the component horizontally.	Range is 0–1920.
y	Number of pixels that defines the starting position of the component vertically.	Range is 0–1080.

Table 4 Text Area Component Properties (continued)

Property Name	Description	Values
width	Number of pixels that defines the horizontal width of the component.	Range is 0–1920.
height	Number of pixels that defines the vertical height of the component.	Range is 0–1080.
rotation	Number of degrees to rotate the component box.	Range is 0–360. Default is 0.
Text	Character string to display within the text area component. Note The Text property does not support line wrap, so the length of the string that can be supported is the canvas width.	Default is “TextAreaN,” where <i>N</i> is a number in sequential order of the number of Text area components added to the canvas.
Font Size	Size of the Arial font for the Text property.	Default is 36. Maximum is 128.
Color	Color of the Text property. Tip Remember that the Widgets tool displays a white background for ease of use, but if you are not using any background images, a black font will <i>not</i> be visible on the TV screen because the TV background is actually black.	Hexadecimal value of the color or by selection in the color grid. Black is the default.
Font Name	Font style for the Text property. Note Cisco StadiumVision Director only provides the Arial font by default. To support additional fonts, you must install them using the Software Manager.	Possible values are: <ul style="list-style-type: none"> • Arial (Default) • Available custom fonts.

Standalone Graphic Component Properties

In addition to setting the size, location, and rotation on the canvas for Graphic components, you can import an image and also specify that it retain the dimensions of the original image.

Table 4 describes the property options that can be applied to standalone Graphic components. No effects are supported for Graphic components.

Table 5 Graphic Component Properties

Property Name	Description	Values
x	Number of pixels that defines the starting position of the component horizontally.	Range is 0–1920.
y	Number of pixels that defines the starting position of the component vertically.	Range is 0–1080.
width	Number of pixels that defines the horizontal width of the component.	Range is 0–1920.
height	Number of pixels that defines the vertical height of the component.	Range is 0–1080.

Table 5 *Graphic Component Properties (continued)*

Property Name	Description	Values
rotation	Number of degrees to rotate the component box.	Range is 0–360. Default is 0.
Content Name	Filename of the image to be displayed in the component. Note The image must be uploaded to the Content Management System (CMS) from the Control Panel > Content > Import area of Cisco StadiumVision Director to be made available to the widget.	Unspecified. Default images from the system are used.
Use image original size	When selected, changes the image size in the component to the size stored in the CMS. Note If you select this option and then unselect it, the image does not revert to the settings in the widget.	Default is disabled.

PicToScreen Component Properties and Effects

[Table 6](#) describes the property options and [Table 7](#) describes the effects that can be applied to PicToScreen components.

Table 6 *PicToScreen Component Properties*

Property Name	Description	Values
x	Number of pixels that defines the starting position of the component horizontally.	Range is 0–1920.
y	Number of pixels that defines the starting position of the component vertically.	Range is 0–1080.
width	Number of pixels that defines the horizontal width of the component.	Range is 0–1920.
height	Number of pixels that defines the vertical height of the component.	Range is 0–1080.
rotation	Number of degrees to rotate the component box.	Range is 0–360. Default is 0.
Images to Display	Maximum number of images to display from the data feed. Tip The value of the Images to Display property should match the value of the “Number of messages keep” data option for an Atom or RSS feed data source.	Range is 1–5. Default is 5.
Use default image if no image present.	When enabled, replaces any missing images that might occur from the data feed (up to the maximum number for display) with a default image.	Default is enabled.

Table 6 *PicToScreen Component Properties (continued)*

Property Name	Description	Values
Content Name	Filename of an image to be used as the default when default images are enabled for the component.	Unspecified. Default images from the system are used.
Image Duration	Length of time (in seconds) to display the current set of images.	Range is 5 to 25. Default is 5.

Table 7 *PicToScreen Component Effects*

Effect Name	Description	Values
Image Duration	Length of time (in seconds) to display the current set of images.	Range is 5 to 25. Default is 5.

TextToScreen Component Properties and Effects

[Table 8](#) describes the property options and [Table 9](#) describes the effects that can be applied to TextToScreen components.

Table 8 *TextToScreen Component Properties*

Property Name	Description	Values
x	Number of pixels that defines the starting position of the component horizontally.	Range is 0–1920.
y	Number of pixels that defines the starting position of the component vertically.	Range is 0–1080.
width	Number of pixels that defines the horizontal width of the component.	Range is 0–1920.
height	Number of pixels that defines the vertical height of the component.	Range is 0–1080.
rotation	Specifies the orientation of the component.	Horizontal or vertical. Default is horizontal.
Font Size	Size of the Arial font for the text that appears in the component.	Default is 36. Maximum is 128.
Font Color1	Sets a font color for the first text data field in a feed, and alternates with Font Color 2. If there is only one message, then only Font Color 1 is used. Tip Remember that the Widgets tool displays a white background for ease of use, but if you are not using any background images, a black font will <i>not</i> be visible on the TV screen because the TV background is actually black.	Hexadecimal value of the color or by selection in the color grid. Default is black.

Table 8 *TextToScreen Component Properties (continued)*

Property Name	Description	Values
Font Color2	Sets a font color for the second text data field in a feed and alternates with Font Color 1.	Hexadecimal value of the color or by selection in the color grid. Default is black.
Font Name	Font style for the component. Note Cisco StadiumVision Director only provides the Arial font by default. To support additional fonts, you must install them using the Software Manager.	Possible values are: <ul style="list-style-type: none"> • Arial (Default) • Available custom fonts.

Table 9 *TextToScreen Component Effects*

Effect Name	Description	Values
Fade	Gradually shows the text data field in the component.	—
Message display duration (Seconds)	Length of time (in seconds) that the text from the data feed is displayed.	Range is 5–25. Default is 5.

Data Binding and Preview

After you have mapped the output fields for an external data source, you can bind that data to a component in a widget and see a preview of that live content feed.



Note

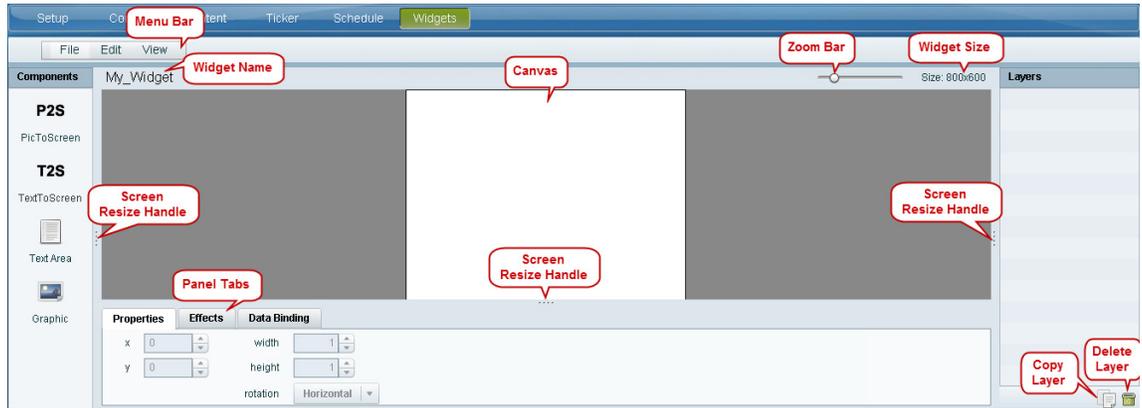
Live data is polled for preview every 15 seconds in the Widgets tool (including for the Data Pull component). There can be a slight delay before you see an update of the content within a component. This polling interval is for preview only and is not the same polling interval used in the data source configuration.

Only certain data fields are supported for the PicToScreen and TextToScreen components. For more information, see the [“Restrictions for Using the Widgets Tool”](#) section on page 62.

User Interface Characteristics

Figure 7 shows the areas of the user interface for a new widget.

Figure 7 *New Widget Interface*



The Widgets tool interface supports some of the following functions and characteristics:

- Multi-selection of components using standard Control key sequences to move components as a group.
- Showing or hiding the bounding box of a component.
- “Zoom-to-fit” function by double-clicking the zoom knob.
- Change the widget dimensions by double-clicking the widget size in the upper right of the screen.

How to Design the Layout of External Content for Display

This section includes the following tasks:

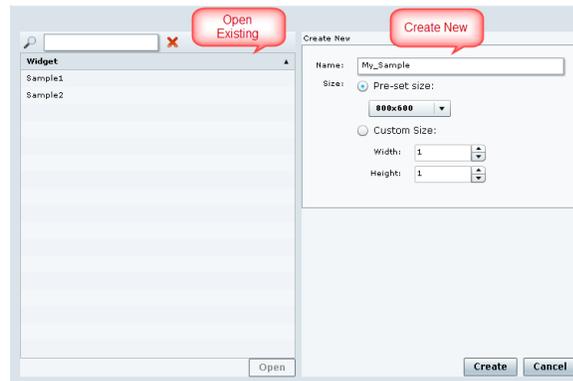
- [Accessing the Widgets Tool, page 75](#)
- [Creating New Widgets, page 75](#)
- [Working with Images, page 77](#)
- [Adding Text Areas, page 79](#)
- [Showing the Component Bounding Box, page 79](#)
- [Resizing and Rotating Components, page 81](#)
- [Working with Large Data Sources, page 82](#)
- [Binding External Data to a Widget, page 83](#)
- [Modifying Existing Widgets, page 84](#)

Accessing the Widgets Tool

To access the Widgets tool, complete the following steps:

-
- Step 1** Log into Cisco StadiumVision Director as an administrator.
- Step 2** From the main menu, click **Control Panel > Widgets**.
A dialog box opens for you to select an existing widget or create a new one (Figure 8).

Figure 8 Dialog Box to Open Existing or Create New Widgets

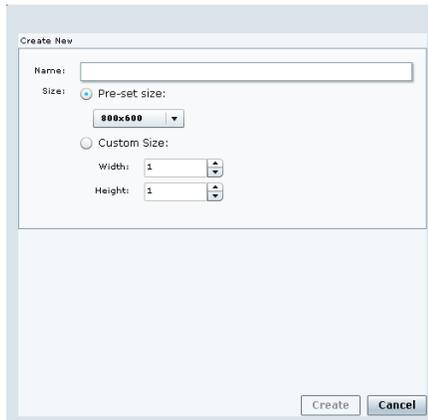


Creating New Widgets

You can create a new widget when you first access the Widgets tool, or you can create a new widget from within the tool.

To create a new widget from within the tool, complete the following steps:

-
- Step 1** From the menu bar in the Widgets tool, click **File > New**.
The Create New dialog box is displayed (Figure 9).

Figure 9 Create New Dialog Box

Step 2 In the **Name** box, type a name for your new widget.

Step 3 Do one of the following to specify the size of the widget:

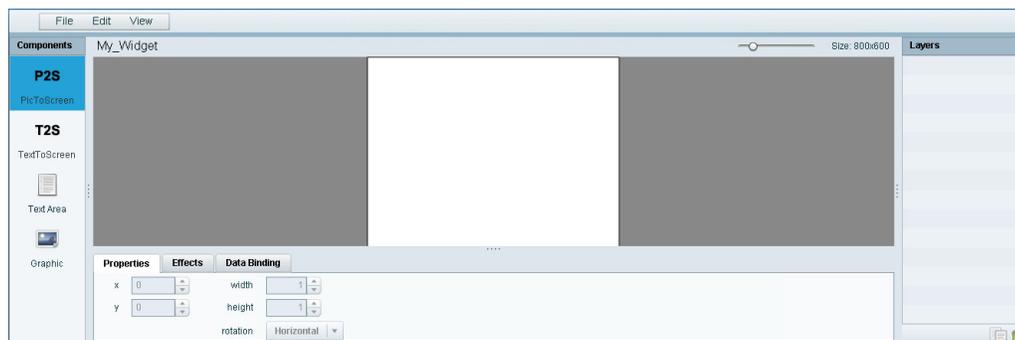
- Click **Pre-Set** size and click the drop-down list to select a predefined dimension (WxH) for the canvas in pixels.
- Click **Custom Size** and specify a number (in pixels) for the width and height dimensions of the canvas.

Step 4 Click **Create**.

The tool displays a white box of the size that you created, with the dimensions that you specified displayed in the upper right corner ([Figure 10](#)).

**Note**

The background canvas of the widget is actually transparent, but it is not shown with the graphics-standard checkerboard pattern to ease visibility of the components.

Figure 10 New Widget Screen

Step 5 Click **File > Save**.

Changes to the widget are saved.

Working with Images

This section includes the following topics:

- [Adding Standalone Images, page 77](#)
- [Resizing and Rotating Images, page 78](#)

Adding Standalone Images

Images can be added as a background to your layout in the Widgets tool, and they also can be bound to a data field using the External Content configuration, such as to add a team logo to a score field that will appear in the widget.

To support the addition of a graphic bound to a particular external source data field, see the “[Modifying the Output Format of a Statistic](#)” section on page 55.

The Widgets tool uses the Graphic component to support the addition of images as a background or in other places on your canvas.

Prerequisite

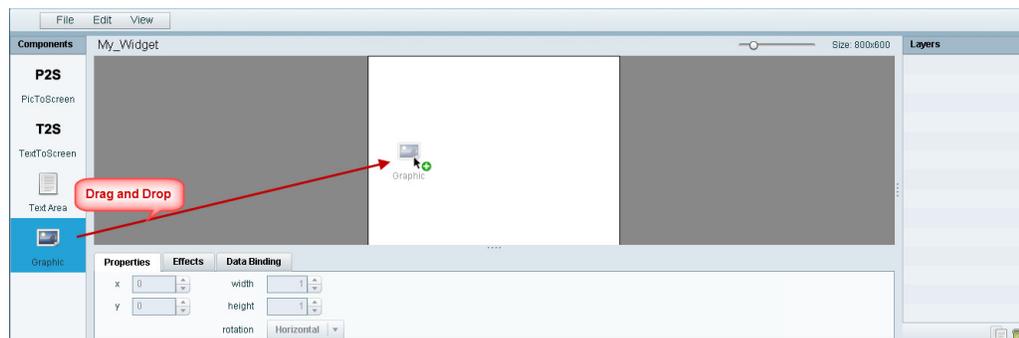
Before you can add an image to a widget, you must import the image file into Cisco StadiumVision Director by going to **Control Panel > Content > Import** to select and open a file. In the Import Content dialog box, specify any options and click **Upload**.

Procedure

To add an image to a widget, complete the following steps:

- Step 1** Open or create the widget where you want to add an image.
- Step 2** In the Components box, select the Graphic component and drag and drop it onto your widget canvas ([Figure 11](#)).

Figure 11 Drag and Drop Graphic Component on Widget Canvas



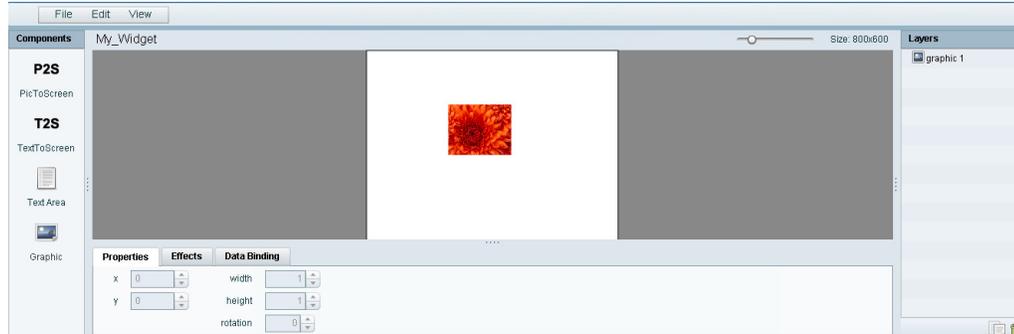
Tip

Double-click a component to automatically add it to the canvas.

- Step 3** To add an image, be sure that the graphic component where you want to add the image is selected, and click **Change**.

- Step 4** In the Select Image dialog box, highlight the image that you want to add and click **Select**. The image is added and scaled to the size of your graphic component box (Figure 12).

Figure 12 Image Added to Graphic Component



- Step 5** Click **File > Save**. Changes to the widget are saved.

Resizing and Rotating Images

Images will automatically scale to the size of the graphic component box in the widget.



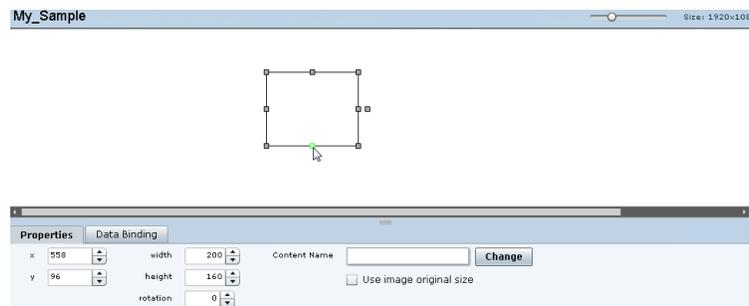
Tip

To find the size of your image file, use the List view for your images in **Control Panel > Content**. Select your image and look at the size in the Dimension field in the “Other content metadata” box.

You can scale the graphic to its original size after you add it, or if you modify the image size in the widget, you can revert back to its original size using the “Use original image size” option, which is recommended.

Figure 13 shows the options available on the Properties panel for a graphic component and displays the different bounding box handles that can be used to modify the size of a component box.

Figure 13 Graphic Component Properties Panel and Bounding Box Handles



You can modify the size and position of an image in the following ways:

- Explicitly setting values on the Properties panel.

- Selecting the bounding box and using the handles to modify it.
- With the graphic component selected, select the “Use original image size” checkbox to revert the image to its original size when imported.

For more information, see the [“Resizing and Rotating Components” section on page 81](#).

Adding Text Areas

A text area component is added to a widget in the same way as a graphic component, but with some different properties. A text area supports a character string, and font size and color selection.

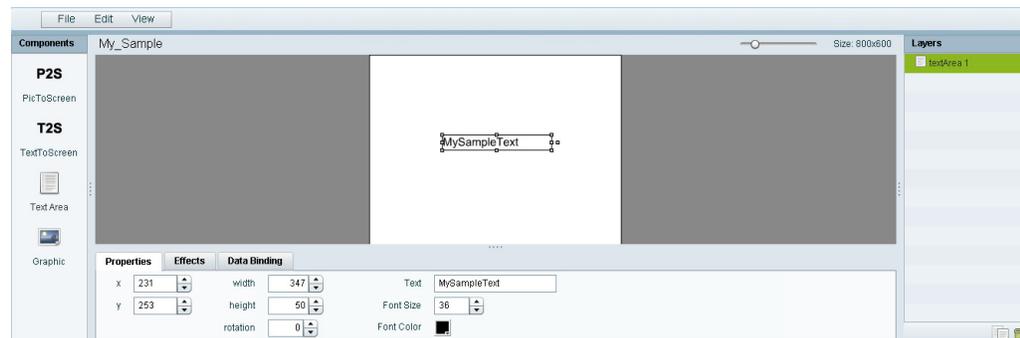
The default font is Arial. You can specify other fonts for the widget if you have installed custom fonts for Cisco StadiumVision Director using the Software Manager.

To add a text area to a widget, complete the following steps:

-
- Step 1** Open or create the widget where you want to add the text area.
 - Step 2** In the Components box, select the Text Area component and drag and drop it onto your widget canvas.
 - Step 3** In the Properties panel, type the character string for the box, and select the font size and color.

[Figure 14](#) shows the options available on the Properties panel for a sample text area component.

Figure 14 Text Area Component Properties



- Step 4** Resize or position the text area as needed. For more details, see the [“Resizing and Rotating Components” section on page 81](#).
 - Step 5** Click **File > Save**.
Changes to the widget are saved.
-

Showing the Component Bounding Box

When you add a component onto the canvas, a temporary bounding box appears while the component is selected. If you move away from that component into another area of the canvas, you will not be able to see the boundaries of the component box.

Figure 15 shows an example of a text area and graphic component displaying the default setting without a visible boundary when the component is unselected. Notice that while the graphic component is selected the temporary bounding box is displayed in Figure 15, but when unselected it appears to disappear from the canvas (Figure 16).

Figure 15 Default Setting Without Bounding Box and Selected Component

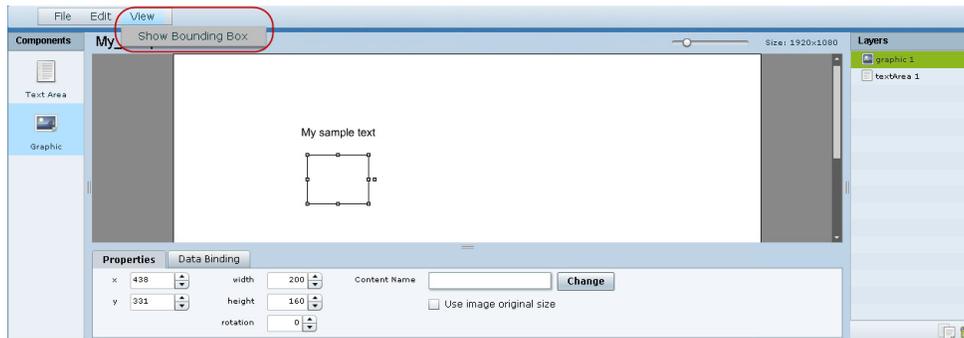
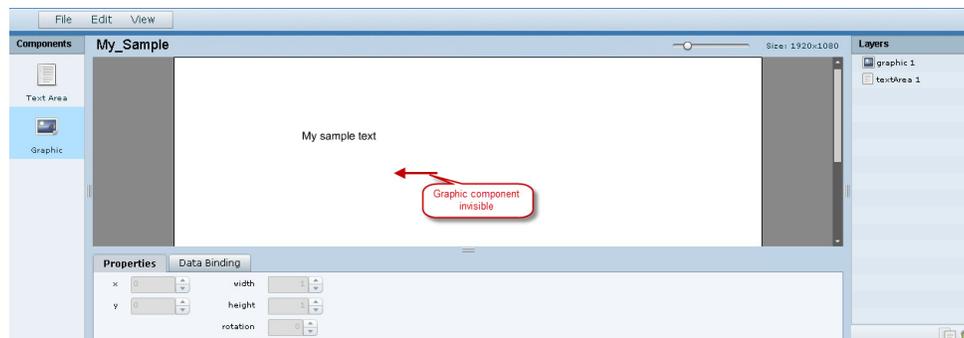


Figure 16 Default Setting Without Bounding Box or Selected Components



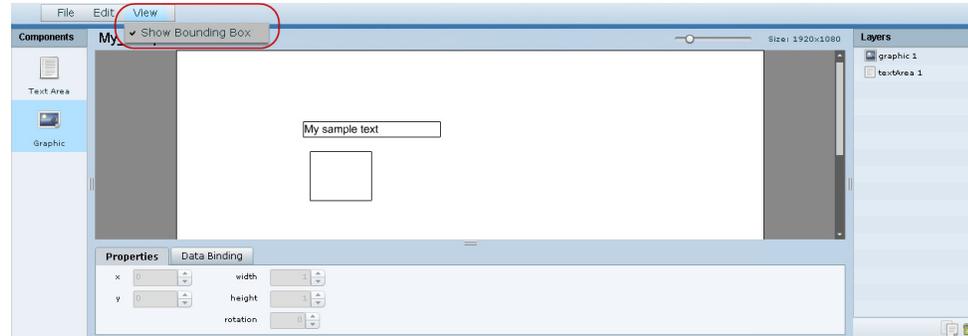
To display a boundary around all components in the widget, complete the following steps:

-
- Step 1** From the menu bar, go to **View**.
 - Step 2** Click **Show bounding box**.

A checkmark appears beside the option when the bounding box is enabled for display, and all components will display a boundary around them.

Figure 17 shows an example of a text area and graphic component after the Show bounding box option is enabled.

Figure 17 Show Bounding Box Option Enabled



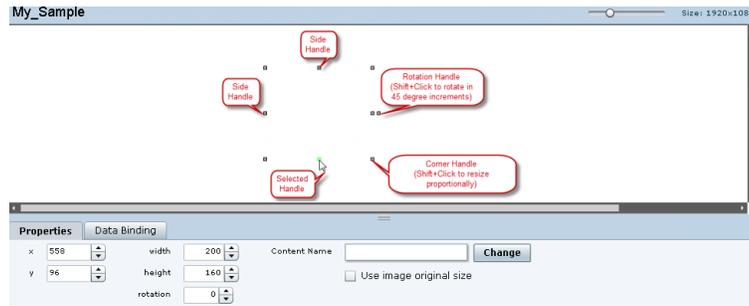
- Step 3** Click **File > Save**.
Changes to the widget are saved.

Resizing and Rotating Components

You can resize and rotate both Text Area and Graphic components in the following ways:

- Change the component location:
 - Select the component and manually drag it to the location that you want on the canvas.
 - Specify the x/y horizontal and vertical values directly in the Properties panel.
- Change the component size:
 - Select and drag a handle on the component's bounding box to enlarge or reduce the component size. If using a handle on a side of the box, only that dimension is changed. If using a corner handle, both dimensions of the box are changed.
 - Shift+Click on a corner handle to resize the box proportionally.
 - Specify the width and height values directly in the Properties panel.
- Change the component angle:
 - Select the rotation handle and drag the box up or down to rotate it.
 - Shift+Click on the rotation handle and drag the box up or down to change the angle in 45 degree increments.
 - Specify the rotation value in degrees directly in the Properties panel.

Figure 18 identifies the bounding box handles that you can use to resize and rotate components.

Figure 18 Bounding Box Handles

Working with Large Data Sources

Cisco StadiumVision Director truncates data packet to 16 KB due to packet size limits on the DMP, so usable data plus overhead cannot exceed 16 KB under normal operation. The 16 KB limit is for *all* external data referenced in a script (possibly by multiple widgets).

If you are working with certain data sources larger than 16 KB and whose data does not need more immediate synchronization across large numbers of (100-200) DMPs (such as for menu board data), you should add the Data Pull component to your widget.

To disable multicast push of a data source in a widget and enable data pull for all fields, complete the following steps:

-
- Step 1** Open or create the widget where your large data source is bound.
 - Step 2** In the Components box, select the Data Pull component and drag and drop it onto your widget canvas.
 - Step 3** In the Properties panel, do one of the following to specify the frequency with which you want to get the data source from Cisco StadiumVision Director:
 - Click the Pull Once checkbox to do a one-time data pull from the data source.
 - In the Pull Interval (seconds) option, specify the number of seconds to wait before pulling data again from the data source.



Tip The Data Pull component is not a presentation component and it will not appear on the display. It does have dimensions and coordinates but they are not necessary to be applied unless you simply want to move or resize the component to simplify your view of the rest of the widget on the canvas.

- Step 4** In the Data Binding panel, select the data source.
- Step 5** Click **File > Save**.

Changes to the widget are saved.

Binding External Data to a Widget

The widget supports the binding of external data to a text area component so that you can display dynamic output fields, such as game periods, scores, and so on from an external source.



Note

If you add a text string to the component, it will be replaced by the real-time data from the output field that you specified in the Data Binding panel.

Prerequisite

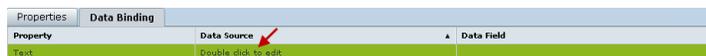
Before you can bind external data to a widget, you must have configured your external data source and mapped the output fields by going to **Control Panel > Setup > External Content**. For more information, see the “[Configuring External Content Integration in Cisco StadiumVision Director](#)”.

Procedure

To bind external data to a widget, complete the following steps:

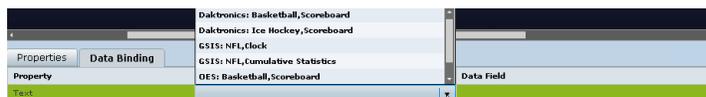
- Step 1** Open or create the widget where you want to add the text area.
- Step 2** In the Components box, select the Text Area component and drag and drop it onto your widget canvas.
- Step 3** In the Properties panel, select the font size and color. Optionally, specify a character string if you want to temporarily see some content in the component while the widget is open.
The font size and color will be applied to the bound data.
- Step 4** In the Data Binding panel, double-click in the first cell under the Data Source heading and open the drop-down list ([Figure 19](#)).

Figure 19 Accessing the Data Source List



- Step 5** Click the drop-down list and select the data source from which you want to bind data. Use the scroll bar to reveal all data sources as necessary ([Figure 20](#)).

Figure 20 Selecting the Data Source



- Step 6** Double-click inside the cell for the data field and select the drop-down list to select the data output field that you want to display in the text area component ([Figure 21](#)).



Note

This step does not apply to the Data Pull component, where all fields in the data source are automatically included and individual fields cannot be selected.

Figure 21 *Selecting the Data Field*

- Step 7** Click **File > Save**.
Changes to the widget are saved.
-

Modifying Existing Widgets

This section includes the following tasks:

- [Changing the Size of a Widget, page 84](#)
- [Deleting Widgets, page 84](#)
- [Renaming Widgets, page 85](#)

Changing the Size of a Widget

To change the size of a widget, complete the following steps:

- Step 1** Open the widget whose size you want to change.
Step 2 From the **Edit** menu, select **Widget Dimension**.



Timesaver

You can also double-click the widget size in the upper right corner of the screen to open the Change Widget Dimension dialog box. See the [“User Interface Characteristics” section on page 74](#).

- Step 3** In the Change Widget Dimension dialog box, specify the new width and height values.
Step 4 Click **Change**.
-

Deleting Widgets

To delete a widget, complete the following steps:

- Step 1** From the **File** menu, select **Manage Widgets**.
Step 2 In the Manage Widget dialog box, select the widget that you want to remove.
Step 3 Click **Delete**.
-

Renaming Widgets

To rename a widget, complete the following steps:

-
- Step 1** From the **File** menu, select **Manage Widgets**.
 - Step 2** In the Manage Widget dialog box, select the widget that you want to rename.
 - Step 3** Click **Rename**.
The widget name is changed to an editable box.
 - Step 4** Type a new name for the widget.
 - Step 5** Click **Close**.
-

Configuration Examples

This section includes the following topics:

- [Basic Widget Configuration Example, page 85](#)
- [Menu Board Widget Using DMB Themes Configuration Example, page 88](#)
- [TextToScreen Widget with Merged Generic Data Sources Configuration Example, page 100](#)
- [Tabular Widget Using List Components for a Transit Schedule Configuration Example, page 108](#)

Basic Widget Configuration Example



Note

This example demonstrates use of the Text Area and Graphic components with data binding. The figures are based on the original interface for the External Content Integration feature; Therefore, some of the figures do not reflect all of the latest available components in the interface.

This section shows a basic widget configuration example to bind external data for an OES Ice Hockey Scoreboard data source. The example uses an image background (Figure 22) that creates static headings for Period, Time Outs, and Score, with companion text area components that will be created as layers on top of the background.

These text areas will be positioned beneath the background image headings and will have real-time data bound to them to appear in the widget.

Figure 22 *Image Background for Scoreboard Widget*



To create a widget and bind data to it, complete the following steps:

- Step 1** Create a new widget with size 1920x1080.
- Step 2** In the Components box, select the Graphic component and drag and drop it onto your widget canvas.
- Step 3** With the graphic component selected, go to the Properties panel and click **Change**.
- Step 4** In the Select Image dialog box, click on the background image that you want to add and click **Select**. The image is added and scaled to the size of your graphic component box.
- Step 5** In the Properties tab, specify the following values (Figure 23):
 - a. x and y—0
 - b. width —1920
 - c. height—1080
 - d. rotation—0

Figure 23 *Background Image Added to Graphic Component*



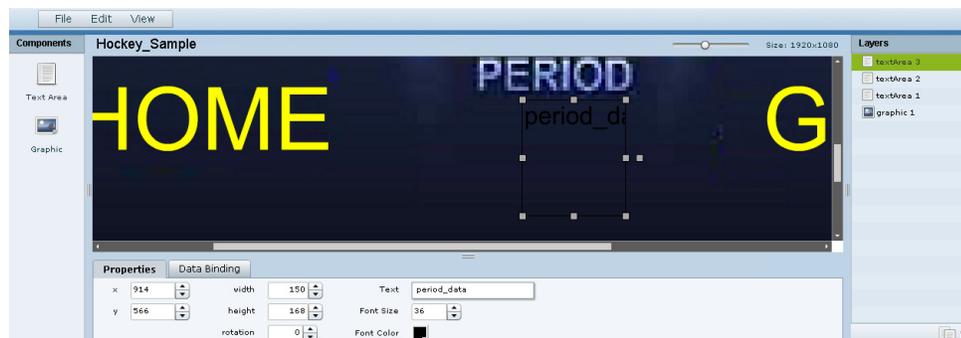
- Step 6** Create two new text area components to add “HOME” and “GUEST” as the specified text string positioned as layers on top of the background graphic component (Figure 24).

Figure 24 Addition of Home and Guest Text Areas

- a. Drag and drop the Text Area component onto the background graphic.
- b. In the Properties panel, specify the following values:
 - Text box—Type “HOME”
 - Font size—128
 - Font color—Choose yellow
- c. Select the handles on the text area box and expand its boundaries enough to reveal the text.
- d. Select the text area component and drag it to the left side of the background graphic over the Time Outs Left and Score fields.
- e. Repeat steps a–d using “GUEST” text and drag the component to the right side of the background graphic.

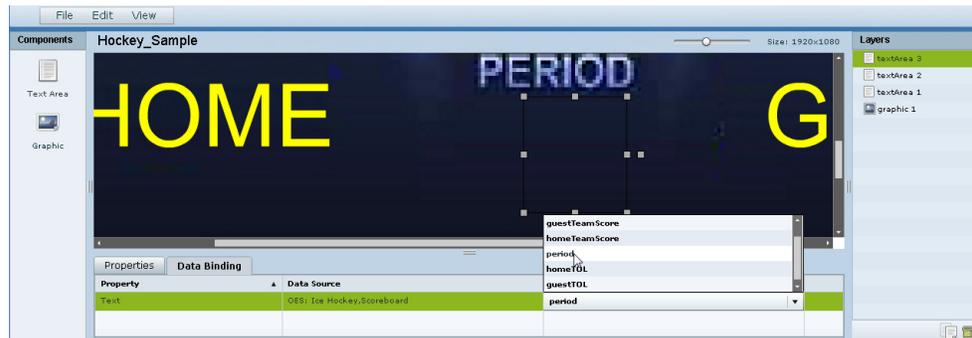
Step 7 Add Text Area components to bind data for Period, Time Outs, and Score areas for both teams:

- a. Drag and drop the Text Area component onto the background graphic.
- b. (Optional) Zoom in as needed to see the component better.
- c. Position and resize the component beneath the heading for which you are going to bind corresponding data (Figure 25).
- d. (Optional) Add text to temporarily identify what data will be in the box, such as “period_data.”

Figure 25 Text Area Component to Bind Period Data

- e. In the Data Binding panel, select the **OES: Ice Hockey, Scoreboard** data source and the **period** data field (Figure 26).

Figure 26 *Period Data Field Binding for OES Ice Hockey Scoreboard Source*



f. Repeat steps a–e for all other data fields that you need to bind.

Step 8 Click **File > Save**.

Changes to the widget are saved.

Menu Board Widget Using DMB Themes Configuration Example

This advanced example shows how to create a menu board widget by configuring a data source for a menu board that uses a menu theme that you have defined in the Cisco StadiumVision Director Dynamic Menu Board (DMB) application. The benefit of using a DMB theme for your widget-based menu boards is that it allows you to more easily show categories of items on your menu.

To successfully configure a menu board widget you must be familiar with the content and format of your menu theme data to be mapped into the List components. If you are using a standard menu theme from DMB, see the “[Appendix A: Default Menu Theme Reference](#)” module of the *Cisco StadiumVision Director Dynamic Menu Board and Store Configuration Guide* as a useful reference.

[Figure 27](#) shows the menu board widget described in this configuration example.

Figure 27 *Franks Menu Board Widget Example*



This example highlights the following functionality in the External Content Integration Feature:

- DMB menu theme data source configuration.
- Selection of output fields for the background image, column headings, menu items, and menu images.
- Indexing elements in the XML data using the XPath expression editor to select the fields that you want to include on your menu board.
- Widget creation using the following components:
 - Graphic
 - List
 - Text
 - Data Pull

Task List

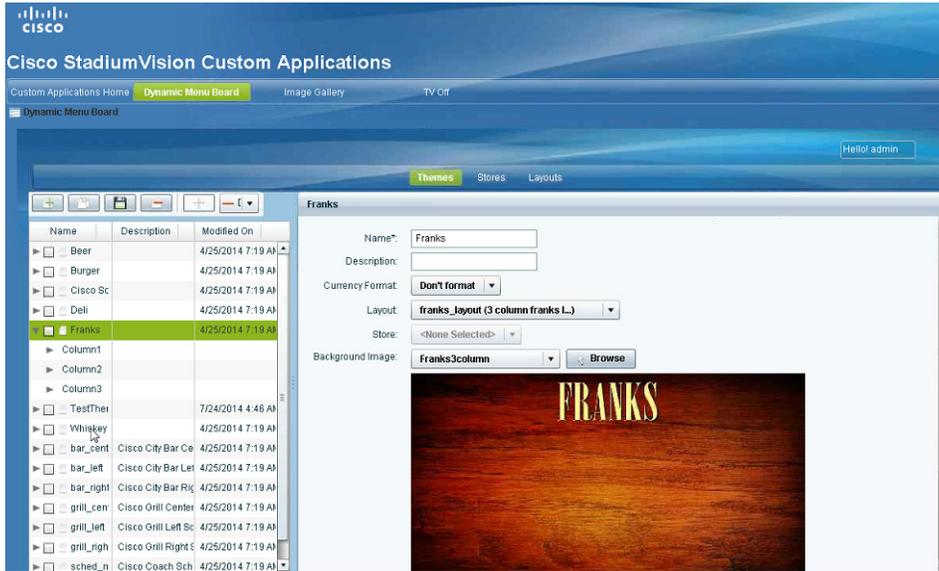
- [Create the Menu Theme in the DMB Application, page 89](#)
- [Create the Data Source for the Menu Theme Widget, page 91](#)
- [Select the Input Fields to be Mapped to the Menu Board Widget, page 91](#)
- [Define the Mapping Expressions, page 94](#)
- [Enable the Data Source, page 95](#)
- [Create the Menu Board Widget, page 95](#)
- [Configure the Background Graphic Component for the Menu Board Widget, page 95](#)
- [Configure the List Components for Each Set of Mapped Data, page 97](#)
- [Configure the Text Area Components for the Group Headings, page 99](#)
- [Add the Data Pull Component, page 100](#)

Create the Menu Theme in the DMB Application

Before you can use a menu theme as a data source, you must create the theme in the DMB application, or use one of the default themes. For more information about creating menu themes, see the [Cisco StadiumVision Director Dynamic Menu Board and Store Configuration Guide](#).

In this example, the Franks default DMB menu theme is used as the data source for the menu widget. [Figure 28](#) shows the layout details for the Franks theme.

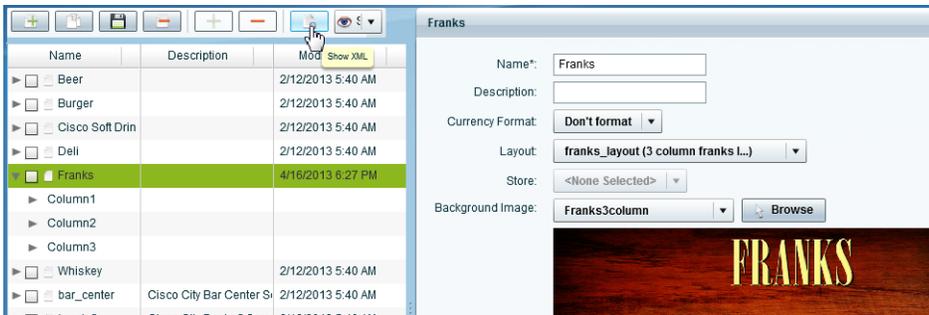
Figure 28 Franks Layout—Default DMB Menu Theme



Tip

As a reference for the Data Integration configuration, you can view the XML data for your menu theme from the DMB application (Figure 29).

Figure 29 Show XML Icon for DMB Themes

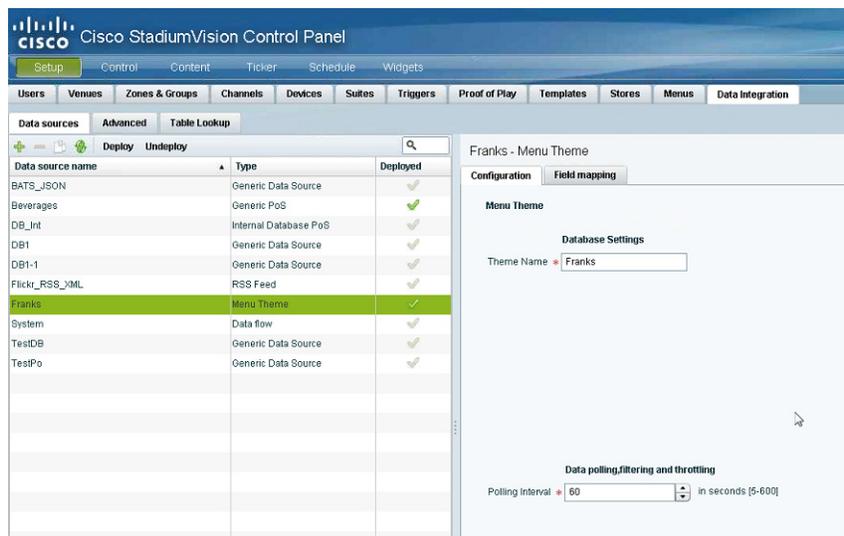


The “Appendix A: DMB Default Menu Theme Reference” module in the *Cisco StadiumVision Director Dynamic Menu Board and Store Configuration Guide* also describes useful details for the default menu themes, including identification of columns and headers for a better understanding of the field mapping task in the data source configuration. Figure 31 shows an excerpt of this information from that document for the Franks layout.

Create the Data Source for the Menu Theme Widget

The Theme Name for the data source must match the name of the theme that you are using in the DMB application. Figure 30 shows creation of the Menu Theme data source type using the Theme Name “Franks” for our example.

Figure 30 Menu Theme Data Source Configuration



For more information about creating menu theme POS data sources, see the “Configuring the Connection to a Menu Theme POS Data Source” section on page 30.

Select the Input Fields to be Mapped to the Menu Board Widget

To perform this task, you need to understand the content and format of the data that you want to display on your menu board and how the output XML is defined by the Data Integration feature.

In the Franks default menu theme, there are three columns and five headers defined (Figure 31). The five possible headers to be used in the menu board are: Entrees, Snacks, Beer, Beverages, and Souvenir.

Figure 31 Franks Layout—Columns and Headers



Franks layout—3-column display with yellow text.

- Background graphic (Franks3column)—Menu title "Franks" part of graphic, but not configurable.
- Three columns (not configurable):
 - Column1 start at (24, 248)
 - Column2 start at (666, 248)
 - Column3 start at (1292, 248)
- Headers—Five textual, predefined (**configurable**):
 - Two per column for Column1 (Entrees / Snacks) and Column2 (Beer / Beverages); one for Column3 (Souvenir)
 - Configured as text in *Name* field only (Yellow, Arial 54 pt).
- Menu item fields (**configurable**):
 - *Image*
 - *Name*
 - *Price*
- Menu item font: Yellow, Arial, 36 pt.



In the XML for POS menu theme data integration, a “Group” instance corresponds to unique column/header combinations for the theme. Therefore in the Franks theme, there are five possible groups.

Table 10 Group Identification

Column/Header Combination	Data	Group No.
Column 1 / Header 1	Entrees	1
Column 1 / Header 2	Snacks	2
Column 2 / Header 1	Beer	3
Column 2 / Header 2	Beverages	4
Column 3 / Header 1	Souvenir	5

In our example shown in [Figure 27](#), we will select two groups (Entrees and Snacks), along with two instances each of item names, images, and prices for each group.

[Figure 32](#) shows an overview of the input fields that are selected for output in our example.



Note

You must drag certain input fields multiple times to create the number of instances that you need on the menu board. Then, you will need to index the expression to identify each instance.

Figure 32 Menu Theme Field Mapping Workflow

Input field	Sample data	Output field	Mapping expression	Data preview
Menu				
Background		bg	/Menu/Background@src	http://10.194.170.145:9090/image/
name	Franks3column	gname1	/Menu/Group[1]@name	Entrees
src	http://10.194.172.46:9090/image/image_gallery/image_...	gname2	/Menu/Group[2]@name	Snacks
Group				
name	Entrees	name1	/Menu/Group[1]@name	2 Items
cal		name2	/Menu/Group[2]@name	2 Items
desc2		image1	/Menu/Group[1]@image	2 Items
description	Entrees	image2	/Menu/Group[2]@image	2 Items
image		price1	/Menu/Group[1]@price	2 Items
Item				
name	Hot Dog			
cal				
desc2				
description	Hot Dog			
image	http://10.194.172.46:9090/image/image_gallery/image_...			
price	\$5.50			

To select the input fields, complete the following steps:

Step 1 (Optional) To use the background associated with the menu theme for the menu board, do the following:

Tip If you want to use a different background than what is in the theme, you can skip this step and simply add a graphic component to the widget.

- Under the **Background** list, locate the **src** field and drag it over to the Output field panel.
- In the Add Output Field dialog box, type the name that you want to use or accept the default.
- Click **OK**.

Step 2 To add groups (categories) for the menu board, do the following:

- Under the **Group** list, locate the **name** field and drag it over to the Output field panel.
- In the Add Output Field dialog box, type a new name for the for the group, such as “gname1” and click **OK**.
- Repeat this step for as many groups as you want to add and give each a unique name.

In our example, two group fields called “gname1” and “gname2” were created.



Note In our example, Figure 32 shows that three groups were actually added to the Output field panel for the data source. However, only two of these groups are actually used in our example widget. Therefore, realize that more output fields can be selected for the data source than you necessarily plan to bind in any single widget.

Step 3 To add items for each group, do the following:

- Under the **Item** list, locate the **name** field and drag it over to the Output field panel.
- In the Add Output Field dialog box, type a new name for the item, such as “itemname1” and click **OK**.
- Repeat this step for as many items as you want to add and give each a unique name.

In our example, two item fields called “name1” and “name2” were created.

- Step 4** To add images for each item, do the following:
- Under the **Item** list, locate the **image** field and drag it over to the Output field panel.
 - In the Add Output Field dialog box, type a new name for the item, such as “image1” and click **OK**.
 - Repeat this step for as many images as you want to add and give each a unique name.

In our example, two image fields called “image1” and “image2” were created.

- Step 5** To add prices for each item, do the following:
- Under the **Item** list, locate the **price** field and drag it over to the Output field panel.
 - In the Add Output Field dialog box, type a new name for the item, such as “price1” and click **OK**.
 - Repeat this step for as many images as you want to add and give each a unique name.

In our example, two price fields called “price1” and “price2” were created.

Define the Mapping Expressions

When you add the group name field to the output field mapping, you might notice that the following default mapping expression is created:

```
/Menu/Group/@name
```

Each field of that type that you add will have this same default expression, which means that only one group would be identified. So, you need to modify this default expression to index each group that you define, and for each field that you have added.

Figure 33 highlights how the index [1] and [2] is added to each output field to identify each element as part of Group 1 or Group 2. Expressions indexed for Group [1] will be associated with the Entrees category, and those indexed for Group [2] will be associated with the Snacks category in this example.

Figure 33 Mapping Expression Example

```

/Menu/Group[1]/@name
/Menu/Group[2]/@name
/Menu/Group[3]/@name → Not used in widget
/Menu/Group[1]/item/@name
/Menu/Group[2]/item/@name
/Menu/Group[1]/item/@image
/Menu/Group[2]/item/@image
/Menu/Group[1]/item/@price
/Menu/Group[2]/item/@price

```

To define the mapping expression to index each group instance, complete the following steps:

- Step 1** In the Output field panel, select the field.
- Step 2** Click the pencil (Edit) icon.
- Step 3** In the Change Output Field dialog box, go to the Expression option and type [1] or [2] after the Group label (before the slash) and click **OK**.

Enable the Data Source

To enable the data source configuration, complete the following steps:

-
- Step 1** In the left pane, select the data source.
- Step 2** Click **Deploy**.
A message box appears asking you to restart the application.
- Step 3** If you have completed all of the changes that you want to make for the data source, restart the application from the **Management Dashboard**:
- Go to **Tools > Data Integration > Restart Application**.
 - In the Data Integration panel, click Play (>).
- Tip** The Play button is located on the bottom left of the Data Integration panel.
-

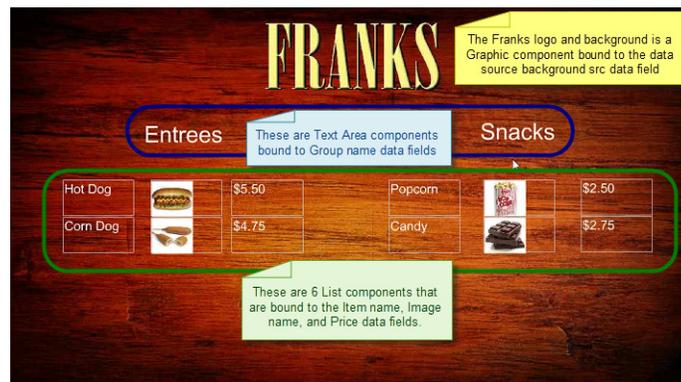
Create the Menu Board Widget

After you have deployed the POS data source, you need to create a new widget for the menu board. For full-screen menu boards, 1920x1080 is the common size.

For details about how to create a widget, see the [“Creating New Widgets” section on page 75](#).

[Figure 34](#) shows the different component types that are used to bind the data fields from the menu theme data source in our example.

Figure 34 Menu Board Widget Components Overview

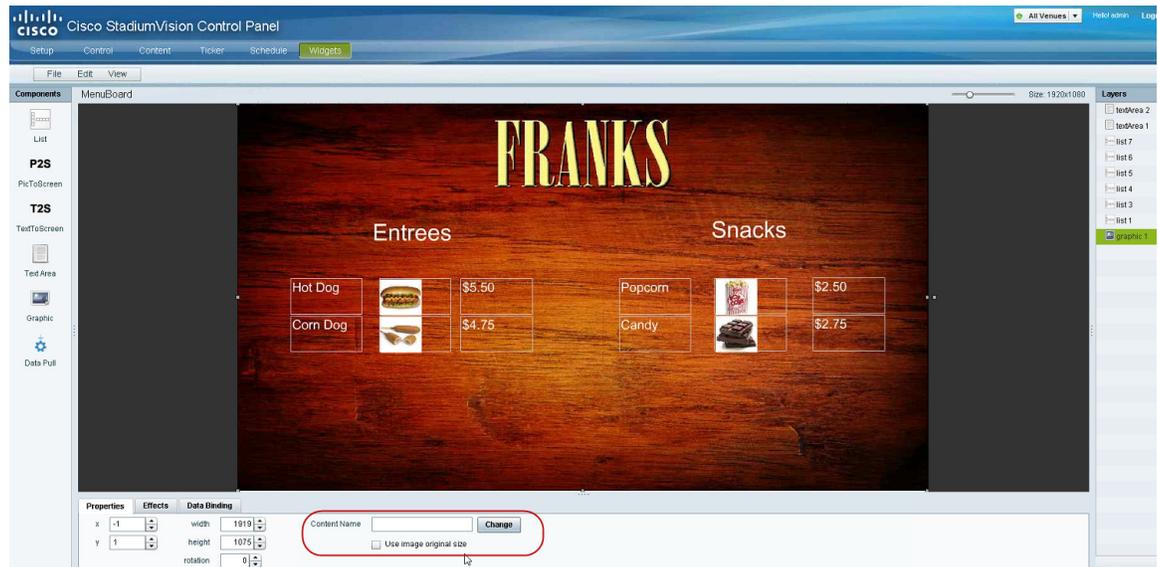


Configure the Background Graphic Component for the Menu Board Widget

To add a background for your menu board widget, you need to configure a Graphic component as the first layer on your canvas. This graphic component can either be bound to the source background from your menu theme data source, or it can be selected from the Cisco StadiumVision Director CMS.

Figure 35 shows where you can use the Content Name on the Properties panel to browse for an image to use as the background. If you want to use a separate image for your menu board background, then that image must be uploaded to the Cisco StadiumVision Director CMS before it can be added to the graphic component.

Figure 35 *Selecting an Image for the Background Graphic on the Properties Panel*

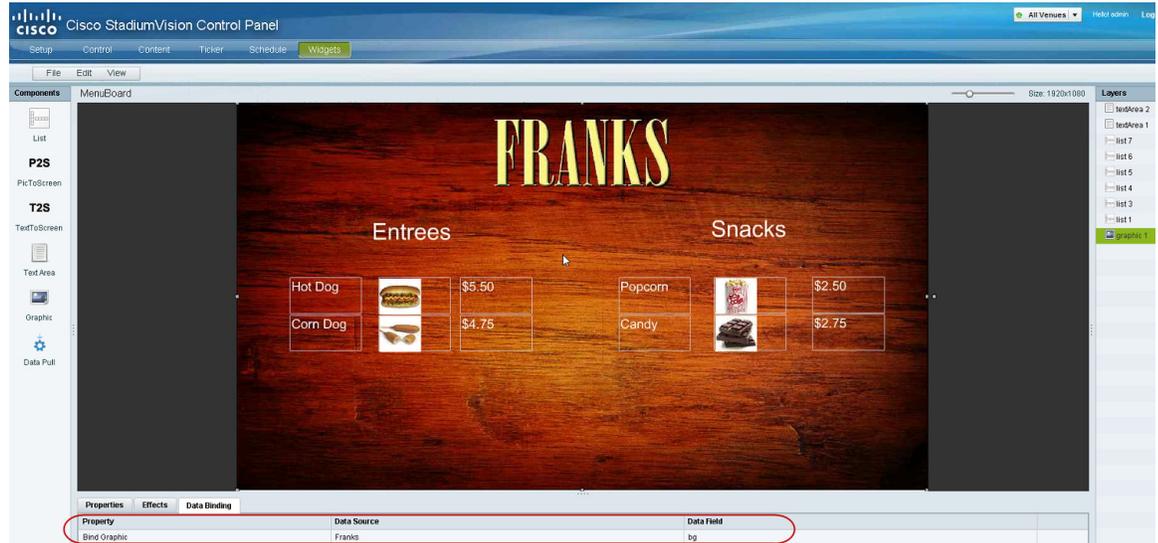


Be sure to specify the size for the graphic component that reflects the size of your menu board. For more information about graphic component properties, see the [“Standalone Graphic Component Properties” section on page 70](#).

For information about how to add an image, see the [“Working with Images” section on page 77](#).

Figure 36 shows the alternative way to select a background graphic for a menu theme, by using the Data Binding configuration. In this example for setting the background, the **bg** data field that we mapped as an available output field in the data source is selected for Data Binding for the graphic component. When you bind the background from the menu theme data source, the image URL to the DMB image gallery is used.

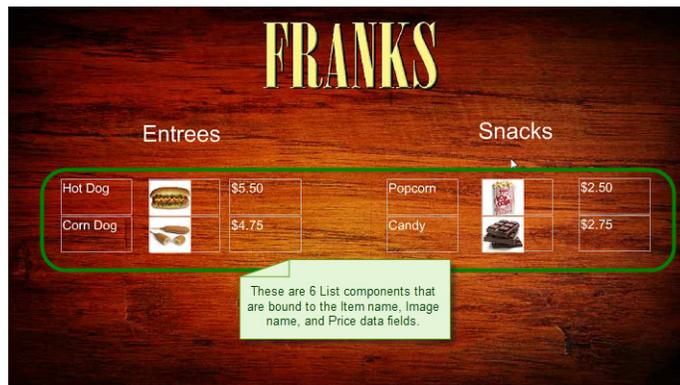
Figure 36 Binding the Menu Theme Data Source Background



Configure the List Components for Each Set of Mapped Data

For each of the group headings (two) in the example, you need a List component for the item name, image, and price for a total of six lists (Figure 37).

Figure 37 Six List Components for the Franks Menu Theme Example



To configure the list components, complete the following steps:

Step 1 Drag and drop the List component onto the canvas.

Step 2 From the **Data Binding** panel, select the Data Source and Data Field.

In our example, “Franks” would be the Data Source, with the following Data Fields selected for each list:

- name1
- image1
- price1
- name2
- image2
- price2

Step 3 From the **Properties** panel, specify the settings for each list.

Figure 38 shows an example of the properties set for the selected image list under the Entrees heading.

Figure 38 Image List Component Properties



- a. Set List Order to **Vertical**.
- b. Set List Render:
 - **Text**—For name and price data fields.
 - **Image**—For image data fields.
- c. (Optional) Click **Grid**.

This allows you to see borders around the cells.

- d. (Optional) Specify the location of the list and the cell dimensions.

Tip For the first list, you can just use the component handles to resize the list. As you move list components on the canvas to align them (Figure 39). From there, you can use the Properties panel to set all of the other list components to those same dimensions.

Figure 39 List Component Grid Lines



e. Specify other properties as desired.

Configure the Text Area Components for the Group Headings

In our example, we configure two Text Area components to create the group headings: Entrees and Snacks.

To configure the text area components, complete the following steps:

- Step 1** Drag and drop the Text Area component onto the canvas.
- Step 2** From the **Data Binding** panel, select the Data Source and Data Field.
In our example, “Franks” would be the Data Source, with the following Data Fields selected for each list:
 - gname1
 - gname2
- Step 3** From the **Properties** panel, specify the settings for each Text Area.
[Figure 40](#) shows an example of the properties set for Entrees group heading.

Figure 40 Text Area Properties



Add the Data Pull Component

As a best practice for menu board data, which can become sizable, you should add the Data Pull component to the menu board widget. Although you can see this component on your widget canvas, it does not appear on the published menu board.

For more information about how to add the data pull component, see the [“Working with Large Data Sources” section on page 82](#).

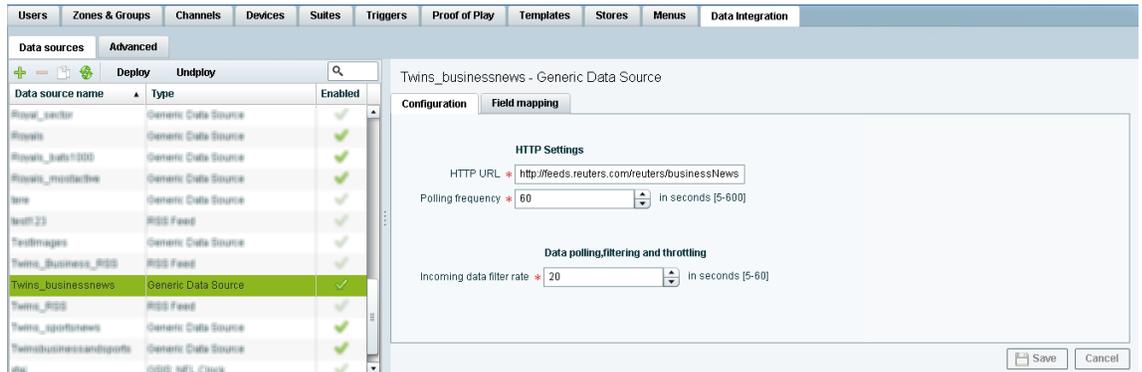
TextToScreen Widget with Merged Generic Data Sources Configuration Example

This advanced example shows how to create generic data sources from RSS feeds, filter the data, and combine the feeds into a new data source for layout in the Widgets tool. The example highlights the following functionality in the External Content Integration feature:

- Generic data source configuration of an RSS data feed.
- Provision of XML sample data.
- Selection of specific item ranges in a data feed using the XPath expression editor.
- Merging of data sources into a data view.
- Data view configuration as a data source.
- Preview of received data in the data source.
- Use of a background graphic in the layout.
- Overlay of a TextToScreen component with data bound from merged data sources (a data view).
- Use of color properties in the TextToScreen component.

To create a TextToScreen widget with merged data sources, complete the following steps:

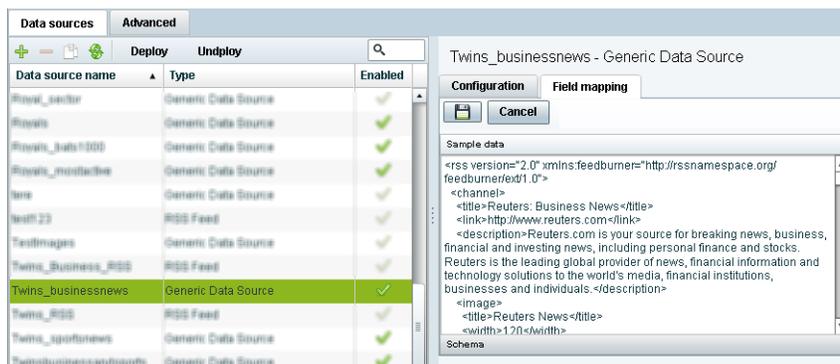
-
- Step 1** From the Control Panel, click **Data Integration**.
For more information, see the [“Accessing the External Content Integration Interface” section on page 23](#).
- Step 2** Add a new Generic Data Source with the HTTP message type.
For more information, see the [“Adding a New Data Source” section on page 23](#).
- Step 3** In the Configuration panel for the new Generic Data Source, type the URL to your RSS feed.
[Figure 41](#) creates a new Generic Data Source named “Twins_businessnews” with the following HTTP URL “http://feeds.reuters.com/reuters/businessNews”. The rest of the configuration uses the default values.

Figure 41 HTTP Configuration for First Generic Data Source

Step 4 Click **Field Mapping**.

Step 5 Click the pencil icon to open the Sample data box, paste the sample XML data from the feed, and save it. For more information, see the [“Providing Sample XML Data for Generic Data Sources”](#) section on page 39.

Figure 42 shows a portion of the XML data pasted into the Sample data box.

Figure 42 Sample Data for First Generic Data Source

Step 6 Expand the RSS input fields made available from the sample data and select a text-based input field to be used in the output display.

For more information, see the [“Selecting Input Statistics and Mapping to Output Fields for Display”](#) section on page 42.

Step 7 Select the Expression mapping option and click **Ok**.

The default XPath expression for the data field is added to the Mapping expression field. For an RSS title element, the default expression is `“/rss/channel/title/text()”`.

Step 8 Click the pencil icon in the Output field icon bar to modify the XPath expression to filter the data provided by the feed.

For more information, see the [“Modifying the Default XPath Expression”](#) section on page 57.

Figure 43 shows a modification to the title data field to add the `“item[position()]>=2andposition()<5]”` predicate to the default `/rss/channel/title/text()` expression:

```
/rss/channel/item[position()]>=2andposition()<5]/title/text
```

In this example, the data will be filtered to pull the 2nd, 3rd, and 4th items in the title element of the feed.

Figure 43 Output Data Filter Using XPath Expression for First Generic Data Source

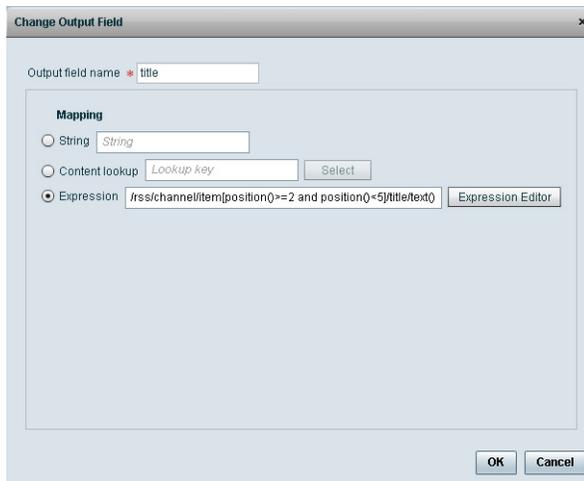
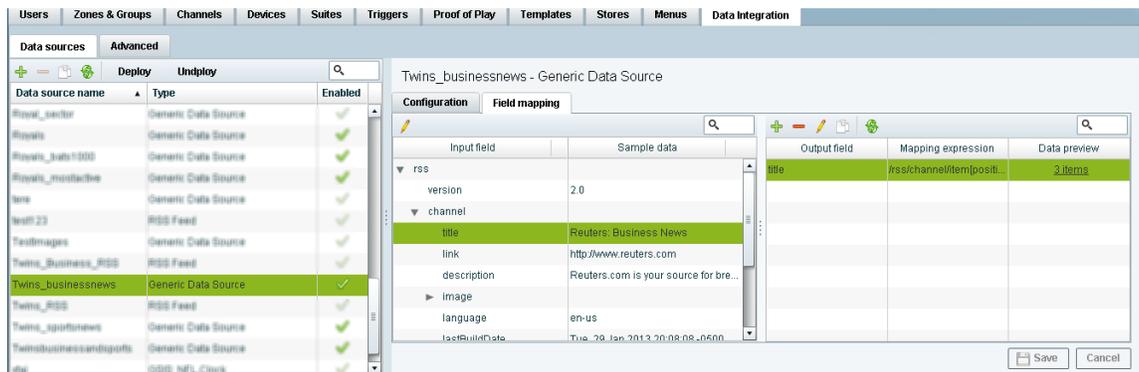


Figure 44 shows selection of the “title” field for output display with the modified mapping expression and the 3 items available for data preview from the RSS feed.

Figure 44 Field Mapping for First Generic Data Source



Step 9 Click the link under Data preview to open the “Preview for” window, and verify the data from the RSS feed.

Based on the filter used in our example to pull 3 items from the title element, a link named “3 items” appears under Data preview. Figure 45 shows the preview of the three title items from the first generic data source RSS feed.

Figure 45 Data Preview for First Generic Data Source with Business Data



Step 10 Repeat Step 2–9 for the second generic data source whose RSS feed content you want to combine with the first generic data source.

Figure 46 creates a second Generic Data Source named “Twins_sportsnews” with the following HTTP URL “http://feeds.reuters.com/reuters/sportsNews”. The rest of the configuration uses the default values.

Figure 46 HTTP Configuration for Second Generic Data Source

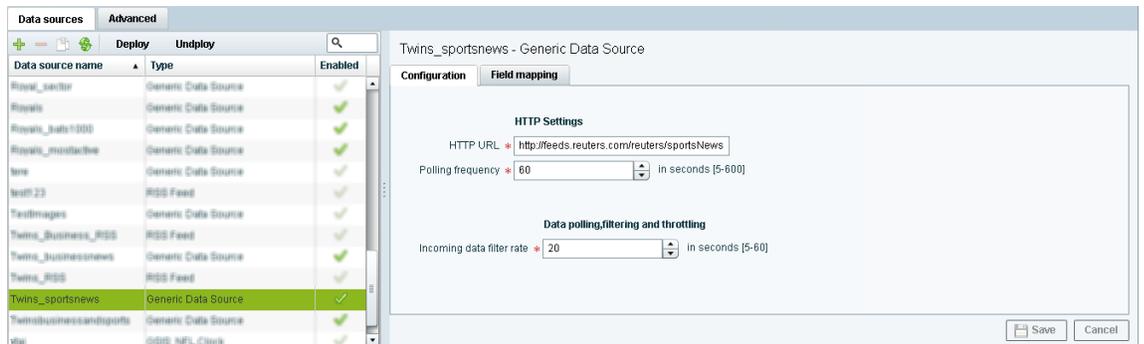


Figure 47 shows the corresponding XPath expression to filter the data to pull the 4th, 5th, and 6th items from the title element for the second generic data source.

Figure 47 Output Data Filter Using XPath Expression for Second Generic Data Source

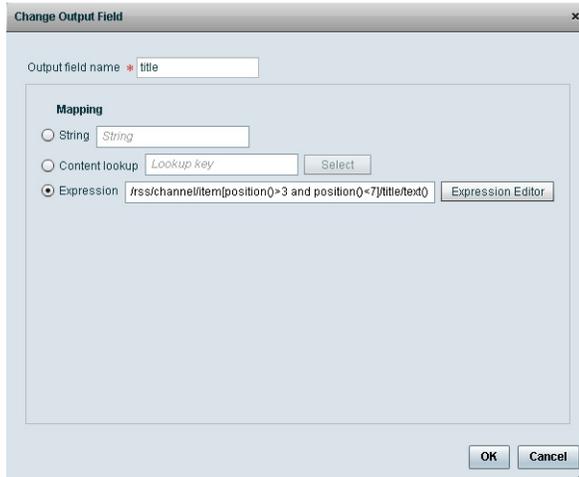


Figure 48 shows the preview of the 3 sports titles filtered from the second generic data source.

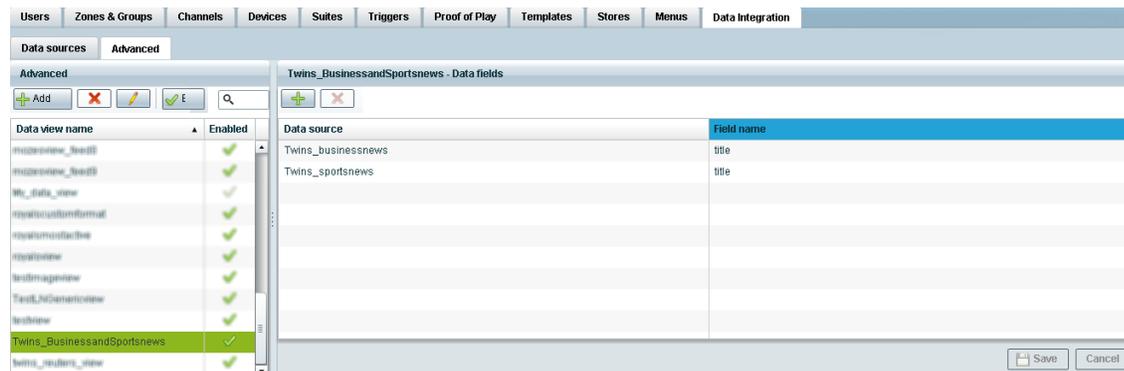
Figure 48 Data Preview for Second Generic Data Source with Sports Data



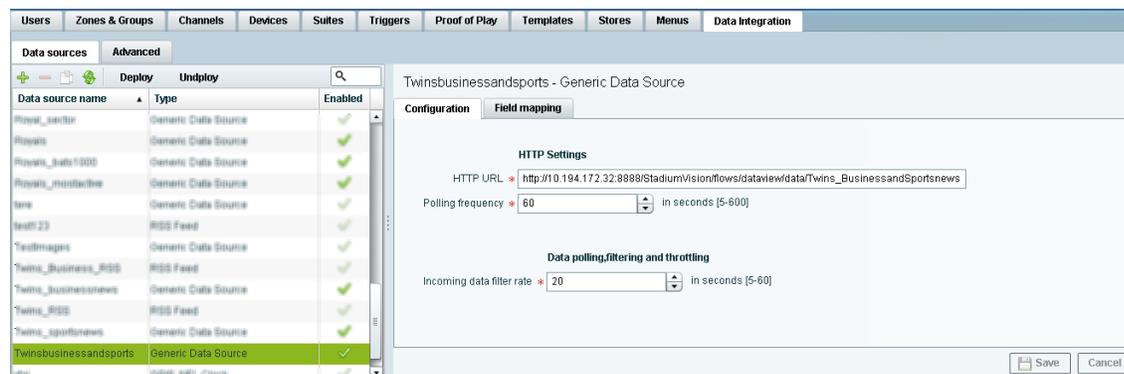
Step 11 To combine content from multiple data sources, click **Advanced** to create a data view and add the data sources that you created.

For more information, see the [“Working with Data Views”](#) section on page 49.

Figure 49 shows creation of a data view named “Twins_BusinessandSportsnews” with the “Twins_businessnews” and “Twins_sportsnews” data sources included.

Figure 49 Data View Creation with Two Generic Data Sources Added

- Step 12** Configure a third generic data source that connects internally to the created data view. For more information, see the “[Configuring the Data View as a New Data Source](#)” section on page 54. [Figure 50](#) shows configuration of the internal HTTP URL using the Cisco StadiumVision Director server IP address and the name of the “Twins_BusinessandSportsnews” data view.

Figure 50 Data View HTTP Configuration

- Step 13** Go to Field Mapping and provide the sample XML data for the combined data sources. The following is an example of the sample XML used:



Note The “entry” node is required in the sample data.

```
<command>
  <type>xmlDataFeed</type>
  <version>1</version>
  <channel>com.cisco.sv.databind</channel>
  <xmlDataFeed>
    <hashcode>45adde4260b54cb41765345a21efd3dc</hashcode>
    <xmlDataFeedEntry>
      <name>.Twins_businessnews.title</name>
      <value>
        <entry index="0">Exclusive:Thermo Fisher weighs Life Tech takeover -
sources</entry>
        <entry index="1">Chesapeake CEO McClendon steps down after year of tumult</entry>
      </value>
    </xmlDataFeedEntry>
  </xmlDataFeed>
</command>
```

```

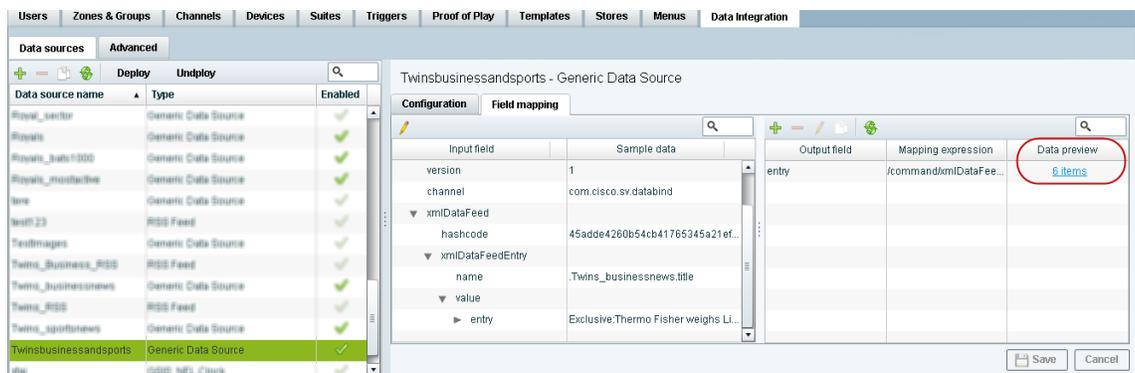
<xmlDataFeedEntry>
  <name>.Twins_sportsnews.title</name>
  <value>
    <entry index="0">Never-say-die Kings rally to haunt Canucks again</entry>
    <entry index="1">49ers' Moss: "I'm the greatest receiver ever"</entry>
    <entry index="2">MLB investigating doping report involving top players</entry>
  </value>
</xmlDataFeedEntry>
</xmlDataFeed>
</command>

```

After the XML sample data is saved, the data sources are combined.

Figure 51 highlights the new link to 6 items available for preview, which includes the 3 items from the business data source and 3 items from the sports data source.

Figure 51 Data View Field Mapping



Step 14 Click the link in the Data preview column to verify the content for display.

Figure 52 shows the combined title items from both the business and sports feeds.

Figure 52 Data View Field Mapping



Step 15 Go to the Widgets tool.

For more information, see the “Accessing the Widgets Tool” section on page 75.

Step 16 Create a new widget.

For more information, see the “Creating New Widgets” section on page 75.

Step 17 Add a Graphic component to the canvas and adjust the size to accommodate the title fields from the RSS feeds.

For more information, see the [“Working with Images”](#) section on page 77.

Step 18 Add a TextToScreen component to the canvas to overlay the Graphic component.

Step 19 Bind the data view data source to the TextToScreen component.

[Figure 53](#) shows binding of the “Twinsbusinessandsports” data source.

Figure 53 Data Binding of the Data Source Created From the Data View

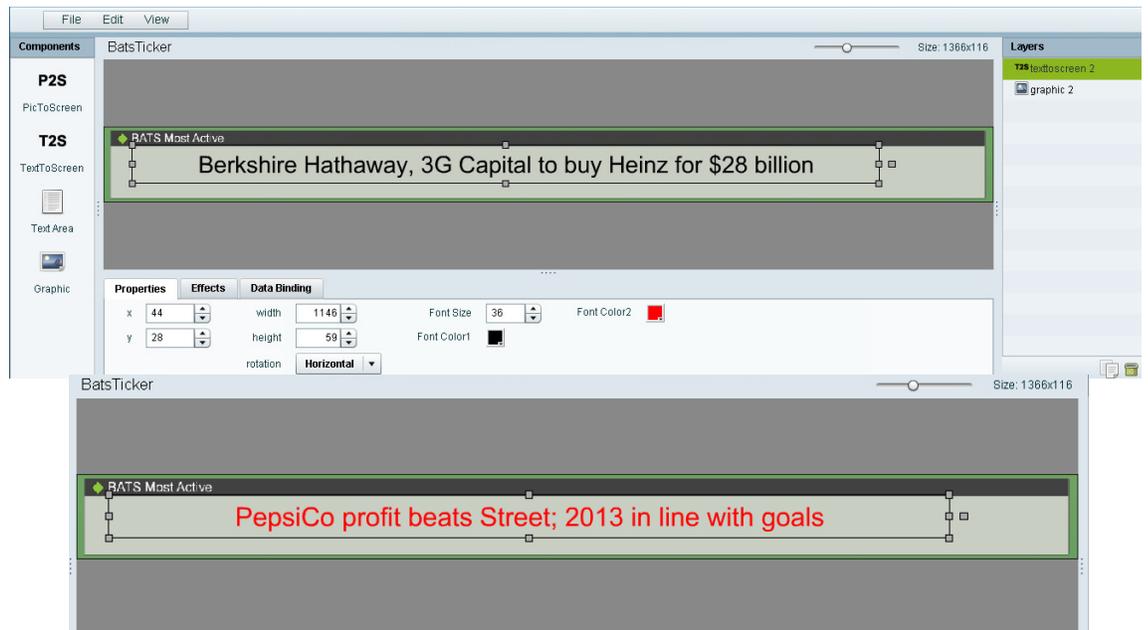
Property	Data Source	Data Field
Bind Messages	Twinsbusinessandsports	entry

Step 20 In the Properties panel, set the options and choose the font colors.

[Figure 54](#) shows the properties used with black set for the first color, and red set for the second color. The bound data automatically appears in the widget. Since two different colors were set in the component properties, the colors will alternate for each message.

Notice that the first message of the combined data that was displayed in the Data Integration preview in [Figure 52](#) appears in the Widget in black font, followed by the second message in red.

Figure 54 TextToScreen Data Preview of First Message Color



Tabular Widget Using List Components for a Transit Schedule Configuration Example

This advanced example shows how to create a generic data source for a transit schedule and map the data source fields to list components to create a schedule widget. To successfully configure a schedule widget you must be familiar with the content and format of your XML data to be mapped into the List components.

This example includes the following tasks:

- [Create the Graphics for the Transit Schedule Widget, page 108](#)
- [Create the Data Source for the Transit Schedule Widget, page 109](#)
- [Provide the Sample XML Data for the Transit Schedule Widget, page 109](#)
- [Map the Data Source Input Fields and Modify the Default Expressions, page 110](#)
- [Create the Transit Schedule Widget, page 113](#)

Create the Graphics for the Transit Schedule Widget

In this schedule example, the sample XML data is known to consist of seven routes and a background graphic was designed with those fixed route names as part of the graphic along with other creative elements for the schedule design ([Figure 55](#)). This background graphic is uploaded to the Content Screen in the Cisco StadiumVision Director Control Panel.

Figure 55 *Schedule Widget Background Graphic Using Explicit Route Names*



Also for this example, the sample XML data is known to provide the following status values:

- Good Service
- Service Change
- Delay

Figure 57 shows a snapshot of some of the sample XML data for the schedule widget data source. The “depTime” and “status” fields will be used in this widget example.

Figure 57 Sample XML Data for the Schedule Widget Data Source

Input field	Sample data
direction0	
trip	
id	823
route	Far Rockaway
direction	Outbound
depTime	2:12 pm
depUTC	1353784320
status	
status	Good Service

Map the Data Source Input Fields and Modify the Default Expressions

In this transit schedule widget example, five output fields will be created from the data source to display departure times and status in the widget as shown in Figure 58 and Figure 59:

- depTime1
- depTime2
- depTime3
- status
- status_icon



Note

The status_icon output field will be created based on the value of the XML status field to look up a graphic that corresponds to that value from the Cisco StadiumVision Director content library. For example, the status “Good Service” will have a suffix of “.png” added to create the value “Good Service.png” (a green checkmark graphic). This requires (as per the introduction to this example) that you have created a graphic of this name and uploaded it to the Content library in Cisco StadiumVision Director.

To further understand the field mappings and their corresponding data expressions, consider the following XML data excerpt for the Far Rockaway route (Figure 58).

Figure 58 XML Route Data for Far Rockaway

```

<route>
  <id>Far Rockaway</id>
  <routeType>2</routeType>
  <name>Far Rockaway</name>
  <color>6E3219</color>
  <colorText>FFFFFF</colorText>
  <trips>
    <direction0>
      <trip>
        <id>823</id>
        <route>Far Rockaway</route>
        <direction>Outbound</direction>
        <depTime>2:12 pm</depTime>
        <depUTC>1353784320</depUTC>
      </trip>
      <trip>
        <id>824</id>
        <route>Far Rockaway</route>
        <direction>Outbound</direction>
        <depTime>3:12 pm</depTime>
        <depUTC>1353787920</depUTC>
      </trip>
      <trip>
        <id>825</id>
        <route>Far Rockaway</route>
        <direction>Outbound</direction>
        <depTime>4:12 pm</depTime>
        <depUTC>1353791520</depUTC>
      </trip>
    </direction0>
  </trips>
  <status>
    <status>Good Service</status>
    <group>FAR ROCKAWAY</group>
  </status>
  <html>

```

 This is depTime1

 This is depTime2

 This is depTime3

 This is status

Figure 59 Output Fields for the Schedule Widget Data Source

Input field	Sample data	Output field	Mapping expression	Data preview
trip		depTime1	/stops/stop/routes/route/t...	-
id	823	depTime2	/stops/stop/routes/route/t...	-
route	Far Rockaway	depTime3	/stops/stop/routes/route/t...	-
direction	Outbound	status	/stops/stop/routes/route/...	-
depTime	2:12 pm	status_icon	fn:lookupContent(fn:conc...	-
depUTC	1353784320			
status	Good Service			
group	BABYLON			

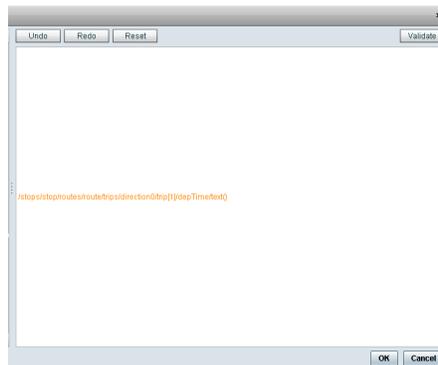
These five output fields will be bound to five corresponding list components in the Widgets tool as shown in red in [Figure 60](#).

Figure 60 Example of Schedule Widget Using List Components

City Zone	2:12	2:42	3:12	Good Service
Babylon	2:12	2:42	3:12	Good Service
Far Rockaway*	2:12	3:12	4:12	Good Service
Hempstead	2:42	3:42	4:42	Good Service
Long Beach	2:42	3:42	4:42	Service Change
Montauk	2:12	2:42	3:12	Good Service
Oyster Bay*	2:12	4:12	4:42	Good Service

To map the data source input fields and modify the default expressions, complete the following steps:

- Step 1** Expand the data source input fields made available from the sample data and select the “depTime” input field to be used in the output display. Name this “depTime1.”
For more information, see the [“Selecting Input Statistics and Mapping to Output Fields for Display”](#) section on page 42.
 - Step 2** Select the Expression mapping option and click **OK**.
The default XPath expression for the “depTime” data field is added to the Mapping expression field: “/stops/stop/routes/route/trips/direction0/trip/depTime/text().”
 - Step 3** With the “depTime1” output field highlighted, click the pencil icon in the Output field icon bar to modify the XPath expression to filter the data provided by the feed.
 - Step 4** In the Change Output Field dialog box, click **Expression Editor**. Add “[1]” to the “trip” element.
For more information, see the [“Modifying the Default XPath Expression”](#) section on page 57.
- [Figure 57](#) shows an example of the Expression Editor and corresponding string: “/stops/stop/routes/route/trips/direction0/trip[1]/depTime/text()”.

Figure 61 Expression Editor for the depTime1 Output Field

- Step 5** Repeat this process to add two more output fields and name them “depTime2” and “depTime3” with corresponding “trip[2]” and “trip[3]” changes made to the default strings in the Expression Editor.



Tip The [1], [2], and [3] syntax added to the trip element specifies that the first trip deptime value, second trip deptime value, and third trip deptime value are pulled from the XML data. In our sample data for the Far Rockaway route in [Figure 58](#), you can see that these times will be “2:12,” “3:12,” and “4:12” as displayed in [Figure 60](#).

- Step 6** In the Field Mapping panel, select the “status” Input field and drag it to the Output field box. This field uses the expression “/stops/stop/routes/route/status/status/text()”
- Step 7** In the Field Mapping panel, select the “status” Output field and click the copy icon.
- Step 8** Name the output field “status_icon” and click **OK**.
- Step 9** Complete the following steps to create an expression that will result in finding the icon that correlates to the status text field value. This field will be used in a list component to display icons beside a list component for the corresponding text status field as shown in [Figure 60](#).
- With the “status_icon” output field highlighted, click the pencil icon in the Output field icon bar to modify the XPath expression to filter the data provided by the feed.
 - In the Change Output Field dialog box, click **Expression Editor**.
 - Drag the concat and lookupContent functions over to the expression and edit the string to match this example:

```
fn:lookupContent(fn:concat('/stops/stop/routes/route/status/status/text()', '.png'))
```

This expression results in a string named “statusvalue.png” and the content for a graphic of that same name will be looked for in the CMS and displayed in the cell.

Create the Transit Schedule Widget

To create the transit schedule widget, complete the following steps:

- Step 1** From the Control Panel, click **Widgets**.
- Step 2** Create a new widget of size 1920x1080.

Step 3 Drag the Graphic component onto the canvas and in the Properties panel, click **Change** beside the Content Name box. Select the background graphic file from the Content screen and click **Select**.

Step 4 Drag the List component onto the canvas and modify the following Properties:

- Cells—7
- Vertical align—Middle
- Horizontal align—Center
- Grid—Check this to ease the sizing of the component. Remember to turn it off again when formatting is complete.
- List Order—Vertical
- List Render—Text



Note When you repeat this step for the “status_icon” output field, be sure to set the List Render property to Image.

Step 5 (Optional) Under Effects, set the Background Color Condition if you want a certain background color to highlight any change in value for that cell, or two different colors for an increase and decrease in value.

- a. Click the corresponding Color Picker to select a color from a palette or specify a specific hexadecimal number for a color.
- b. In the Background color display duration (Seconds), specify the number of seconds that you want the cell background color change to remain on the display.

Step 6 Click Data Binding and modify the following values:

- Select the name of the data source that you created for the widget.
- Select the name of the Data Field that you want to map to this component (depTime1).

Step 7 Drag the component on top of the background graphic and resize it to match the size of the first column on the background graphic.

Step 8 When complete, repeat this process by copying the first List component for the four remaining data sources:

- Bind each list to the corresponding output fields that you created (depTime2, depTime3, status_icon, and status).
- Drag the components and resize them to match the size of the columns on the background graphic.
- Once all List components are created and positioned properly, clear the Grid property checkbox.

Step 9 Save the widget.



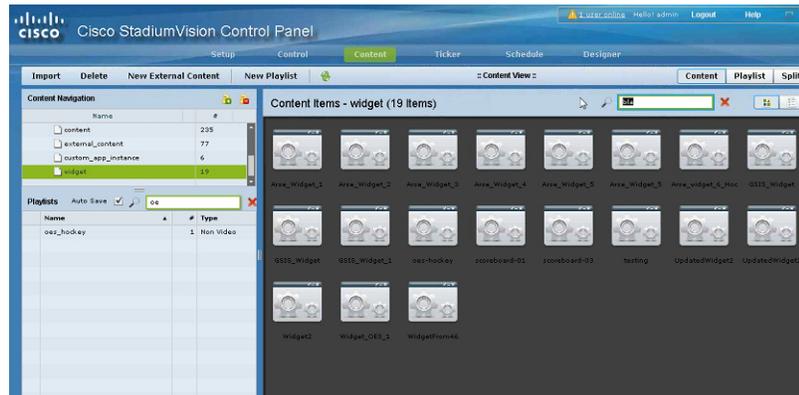
Tip

If you do not see any preview data from your data source, be sure that you have enabled the data source configuration. For more information, see the [“Enabling the Data Source Configuration” section on page 43](#).

What to Do Next

After you have completed creating a widget, complete the following steps to publish it for display:

1. Create a playlist from the Control Panel for the widget that you created. Widgets are categorized in the “By Type” folder under the “widget” type.



2. Create an event script to run the playlist.
3. Schedule the script to display the content on a TV.

