

TIBCO WebFOCUS®

App Studio for the Developer Studio User

March 2021

DN4501647.0321



Contents

1. Introduction to WebFOCUS App Studio	5
The App Studio Paradigm and Terminology	5
Terminology List.....	6
Installing App Studio and Developer Studio on the Same Machine	8
WebFOCUS App Studio Documentation Overview	8
Recommendations for Editing Developer Studio Files in App Studio	9
2. Developer Studio and App Studio Interface Differences	11
Comparing the Default Developer Studio and App Studio Interfaces	11
Main Interface Commands	15
WebFOCUS Administration	17
Development Areas	18
Creating Procedures	20
Configuring Environments	22
Environments Detail Panel.....	24
Managing Your WebFOCUS Environment	25
Filtering Files	25
Filtering for Additional File Types.....	26
View Options	30
3. Creating Procedures	33
Accessing and Using the Procedure View Panel	33
4. Creating Reports	37
Creating Reports	37
Creating SQL Reports	38
Report Options Dialog Box	39
Field Properties Dialog Box	49
Adding Fields Using Smart Context	52
Field Length Limit	53
The Object Inspector Setup Tab	55
&WFFMT Report Variable	56
5. Creating HTML Pages	57
Creating HTML Pages	57

Recreating Save Selection Files.....	58
Using the Properties Panel	58
Using the Settings Panel	60
Requests & Data Sources Panel	63
Using the Tasks & Animations Panel	66
Tasks Section.....	67
Generating a Chart From a Report	70
Making an HTML Page Responsive	71
6. Creating, Editing, and Managing Synonyms and Adapters	77
Creating and Editing Synonyms	77
Configuring Adapters	79
Rebuilding a Data Source	81
Automatically Adding Synonyms to Projects	83
A. Developer Studio Functionality Deprecated in App Studio	85
Deprecated Functionality List	85
Legal and Third-Party Notices	89

Introduction to WebFOCUS App Studio

This topic describes the App Studio paradigm, includes a terminology list, provides installation instructions, and offers recommendations for editing Developer Studio files in App Studio.

Note: The images in this content were captured using App Studio Release 8.2 Version 03. The images may look slightly different depending on the release you are using.

In this chapter:

- ❑ [The App Studio Paradigm and Terminology](#)
 - ❑ [Installing App Studio and Developer Studio on the Same Machine](#)
 - ❑ [WebFOCUS App Studio Documentation Overview](#)
 - ❑ [Recommendations for Editing Developer Studio Files in App Studio](#)
-

The App Studio Paradigm and Terminology

App Studio is the WebFOCUS desktop development environment that you can use to create reports, charts, HTML pages, documents, and much more. Building on the industry-familiar Microsoft Office® ribbon interface, App Studio offers a simplified user experience and workflow, allowing developers to create application content immediately, without a costly and time-consuming learning curve. Additionally, it dramatically increases development efficiency by eliminating the need for developers to utilize multiple tools to piece together an application.

App Studio uses panels, canvases, and a ribbon to eliminate the complex, multi-tool paradigm that is so prevalent in other business intelligence developer solutions. You can rapidly create and design all of your applications within a single, fully-integrated environment.

App Studio is customizable. You can add your own shortcuts using the Quick Access Toolbar. You are able to move your canvases next to one another, so that you may view and develop them, side by side. Panels can also be pinned, hidden, or moved to any position around the interface.

App Studio enables you to work in several development modes at the same time. This development environment provides access to a variety of features and modes from which you can build your applications. The modes available to you in App Studio include, but are not limited to, Report mode, Chart mode, HTML mode, and Document mode.

In addition to its native modes, App Studio invokes WebFOCUS tools, such as the Chart tool, Alert Assistant, and Reporting Objects. These tools are invoked as development modes that are contained within a canvas. This means that when you create a chart, App Studio invokes the WebFOCUS tool as a development mode on the Chart canvas.

Terminology List

The following tables provide a reference for corresponding terms and labels that have changed between the Developer Studio and App Studio user interface.

Developer Studio	App Studio
Toolbar	Ribbon: tabs and groups
Main Toolbar	Home tab
Commands Toolbar	Home tab
WebFOCUS Environments Toolbar	WebFOCUS Administration menu
Report Painter toolbars	Report tab, Format tab, Data tab, Layout tab, View tab, Images tab, Field tab, Appearance tab, Active Report Options tab, Modeling tab
HTML Composer toolbars	Components tab, Controls tab, Positioning tab, Utilities tab
Document Composer toolbars	Insert tab, Positioning tab, Utilities tab
Synonym Editor	Metadata tab

Developer Studio	App Studio
Tool	Canvas or Mode
Graph Tool	Chart canvas, Chart mode
Report Painter	Report canvas, Report mode
HTML Composer	HTML canvas, HTML mode

Developer Studio	App Studio
Document Composer	Document canvas, Document mode
Synonym Editor	Metadata canvas

Developer Studio	App Studio
Window	Panel
Developer Studio Explorer	Environments Tree panel
Procedure Viewer, Task Viewer	Procedure View panel
Properties window	Properties panel

Developer Studio	App Studio
Dialog Box	Dialog Box, Panels, Tabs, Groups
Properties and Settings dialog box	Settings panel
Report Options dialog box	Report tab, Format tab, Data tab, Style group, Field tab, Layout tab, Active Report Options tab, Modeling tab
Field Options dialog box	Report tab, Format tab, Data tab, Style group, Field tab, Layout tab, Active Report Options tab, Modeling tab, Field Properties dialog box
Properties dialog box	File/Folder Properties panel

Installing App Studio and Developer Studio on the Same Machine

App Studio and Developer Studio can be installed, configured, and run on the same machine, at the same time. To ensure the proper functionality of App Studio and Developer Studio, you must appropriately configure the web and/or application server, and ensure that you have sufficient memory.

The following are three installation scenarios, and the appropriate web application configuration for each installation.

- ❑ If WebFOCUS is already installed on the same system, you can install App Studio and Developer Studio by selecting the *Use WebFOCUS configuration* option in the components window during the installation process. This option is selected by default.

This ensures that both products use the web application for WebFOCUS. One set of context roots will be created and used by both products.

- ❑ If WebFOCUS is not installed on the same system, you can install the Managed Reporting Edition of App Studio and Developer Studio, where the products can only access a remote WebFOCUS environment.

- ❑ If WebFOCUS is not installed on the same system, you can install the stand-alone versions of App Studio and Developer Studio. The first product that you install uses the default configuration and generates the default context roots for the web application. When you install the second product, select the *Advanced Configuration* option to specify a second set of context roots.

This allows App Studio and Developer Studio to have specific context roots for their own web application independently. In this scenario, Apache Tomcat™ will load both web applications, and more memory will be used as a result. If memory is a concern, you can copy only the context root XML files for the product that you want to run and save the other context root XML files in a directory outside the Apache Tomcat directory.

WebFOCUS App Studio Documentation Overview

The following documentation components are available for App Studio users:

- ❑ *WebFOCUS App Studio Installation and Configuration* manual
- ❑ Context-sensitive online Help system
- ❑ *WebFOCUS App Studio User's Manual*
- ❑ *Maintain Data Language Reference*

Maintain Data Getting Started

You can access the online Help system through the product. Press F1 on your keyboard, or click *Help* throughout the interface, to open the Help window. The online Help system is designed to provide you all of the information that you need to develop content in App Studio.

You can also access the Help Wizard by selecting the *Help Wizard* check box in the *View* group on the *Home* tab of the App Studio ribbon. The Help Wizard content dynamically changes, depending on where you are in the interface.

Recommendations for Editing Developer Studio Files in App Studio

You can open all Developer Studio files in App Studio. If you choose to edit these files in App Studio, consider the following recommendations.

- When saving your changes, choose the *Save As* option. This option saves the file under a different name without changing the original file.
- You can run Developer Studio and App Studio simultaneously. However, it is not advised to edit the same file in both, at the same time.
- You can edit procedure (.fex) files in App Studio and Developer Studio if the features that you use to edit those files exist in both products.
- When you open an HTML page in App Studio, the HTML page will undergo a conversion process. The conversion occurs in memory. You may execute the page in App Studio to verify correct functionality upon execution. If you save the page, the converted code will be saved. If you exit the tool without saving, the converted code will not be saved. You cannot reopen converted pages in Developer Studio. Only HTML pages created in Release 7.0 Version 6.9 and higher can be directly converted. For earlier releases, you will need to first open your HTML page in a Release 8 version of Developer Studio, and save it, before opening it in App Studio.

This topic outlines the differences between the Developer Studio and App Studio interfaces. You can create the same types of files in both products. However, the process you follow has changed. In Developer Studio, you use menus, dialog boxes, toolbars, and windows to create your files. In App Studio, you use the ribbon, panels, and canvases to create your files.

In this chapter:

- [Comparing the Default Developer Studio and App Studio Interfaces](#)
 - [Main Interface Commands](#)
 - [WebFOCUS Administration](#)
 - [Development Areas](#)
 - [Creating Procedures](#)
 - [Configuring Environments](#)
 - [Managing Your WebFOCUS Environment](#)
-

Comparing the Default Developer Studio and App Studio Interfaces

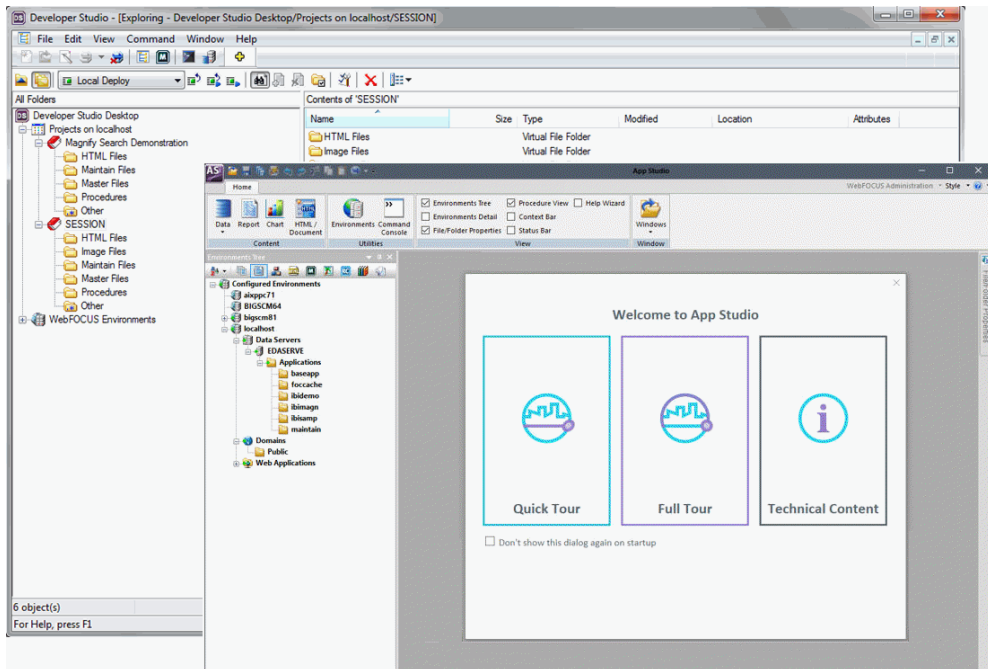
This section identifies the differences between Developer Studio and App Studio upon startup. Both products are launched by clicking on their respective icons.

Developer Studio opens with the Explorer window, which displays your environments, projects, and files. You can use the Main toolbar, Command toolbar, and the Menu bar to access specific tools for creating different files.

App Studio opens with a Welcome screen, where you can access the Getting Started video, and a suite of how-to videos that open within the software window. It also provides a direct link to the App Studio resources that are available on the WebFOCUS 8.2.x Information Center. The Home tab provides commands for creating content, options to manage and edit your environments, and accessing the Command Console. Because App Studio focuses on what you are creating, instead of how you create it, you no longer need to know how to access the many tools found in Developer Studio. You only need to know what content you want to create and where you want to create it.

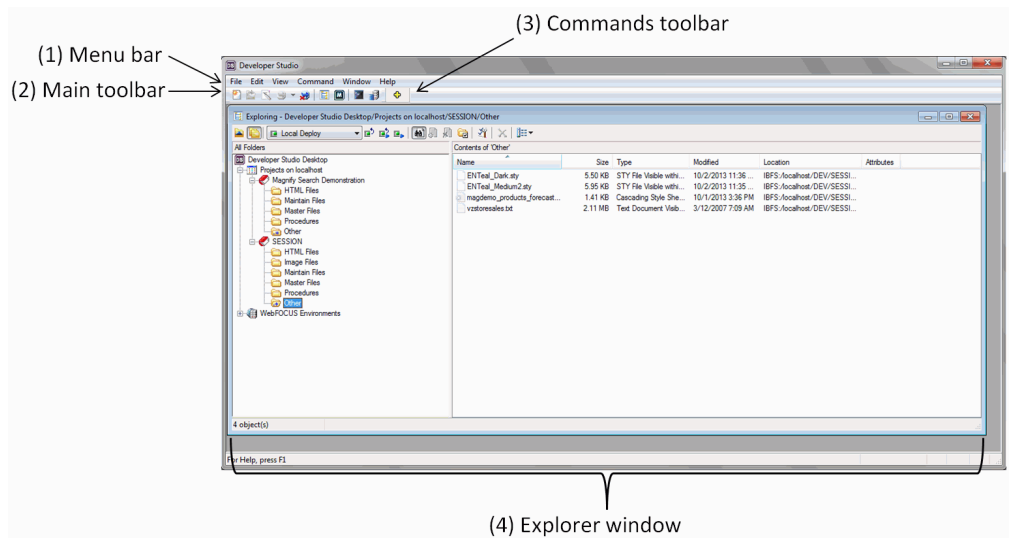
Comparing the Default Developer Studio and App Studio Interfaces

The following image shows a comparison of the default Developer Studio and App Studio interfaces.



Developer Studio

The following image shows the default Developer Studio interface, and highlights the Menu bar, Main toolbar, Commands toolbar, and Explorer window.

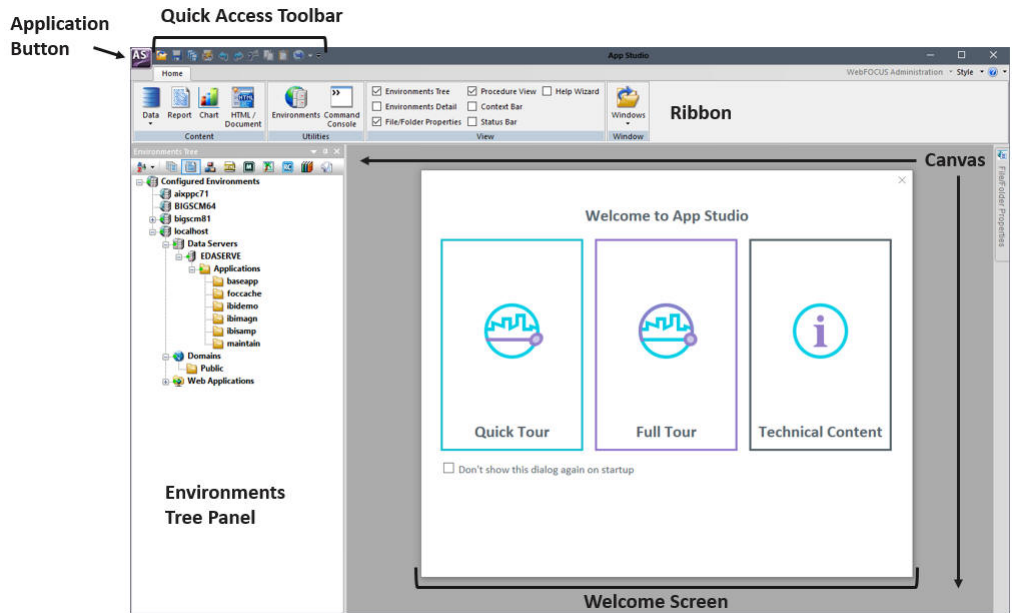


1. **Menu bar.** Displays pull-down menus. The menus and options are context-sensitive, and are determined by the active window or selected object.
2. **Main toolbar.** Provides quick access to commonly performed functions. These buttons always appear on the toolbar, but their behavior is determined by the active window and the selected object.
3. **Commands toolbar.** Provides quick access to any external program added in the External Tools tab of the Developer Studio Options dialog box. When you add an external program, it appears as an additional icon on the Commands toolbar.
4. **Explorer window.** This window is based on the Microsoft Windows Explorer tree structure, and provides access to development tools so you can quickly build and deploy web-based reporting applications.

To begin creating content in Developer Studio, you need to know what tool to use, and where to create it. Some development areas do not allow you to create certain file types.

App Studio

The following image shows the default App Studio interface, and highlights the Welcome screen, Application button, Quick Access Toolbar, ribbon, Environments Tree panel, and the canvas.



The App Studio interface components are identified as follows:

- ❑ **Welcome screen.** Provides access to resources that help you get started with the product as you develop content.
- ❑ **Application button.** Provides access to the Application menu of file-related commands.
- ❑ **Quick Access Toolbar.** Displays frequently used commands, such as Open, Save, Quick Print, Undo, Redo, Cut, Copy, Paste, and Run.
- ❑ **Ribbon.** Displays both static and contextual tabs of commands that you can use to build and customize applications.
- ❑ **Environments Tree panel.** Provides access to the WebFOCUS environments you configure and files you create under each development area.
- ❑ **Canvas.** Provides an active preview of your application development.

To begin creating content in App Studio, you simply need to know what type of file you want to create. For example, if you know you want to create a report, click *Report* on the Home tab to begin creating your report.

Main Interface Commands

This section identifies the differences between the Developer Studio and App Studio main commands.

In Developer Studio, the Main toolbar and Commands toolbar contain the commands that provide quick access to commonly performed functions. Their behavior is determined by the active window and the selected object. Depending on where you are in the Explorer window, certain toolbar commands may be inactive.

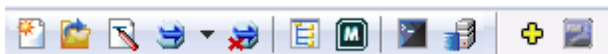
In App Studio, the Application menu, Quick Access Toolbar, and Home tab serve the same purpose as the Main toolbar and Commands toolbar. They contain similar commands to the Main toolbar and Commands toolbar, and contain additional exclusive commands that are only available in App Studio.

You can use the Application menu to create a new document, open an existing document, save an active document, and print an active document. The Options button opens the App Studio Options dialog box, where you can set user preferences, similar to the Developer Studio Options dialog box. The Application menu also lists recently opened documents.

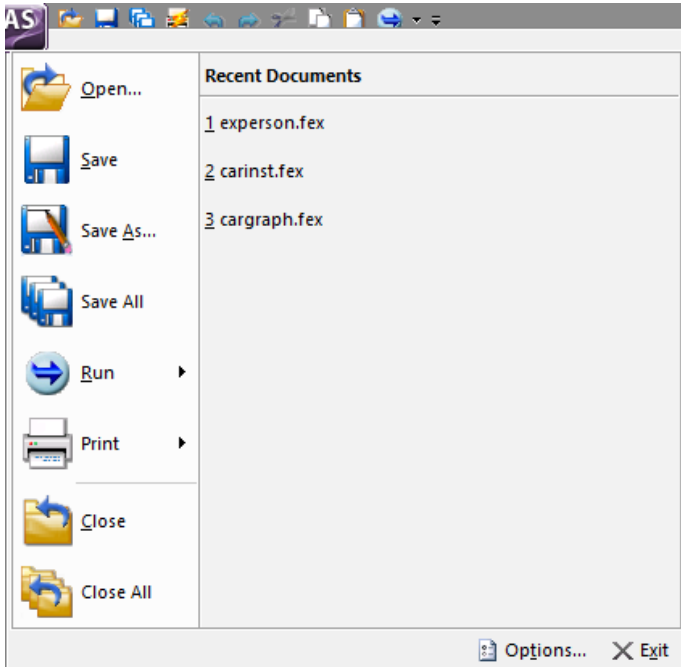
The Quick Access Toolbar contains frequently used commands. You can add or remove commands by using the shortcut menu on any ribbon command. The Quick Access Toolbar contains the Open, Save, Print, Undo, Redo, Cut, Copy, Paste, and Run commands by default.

You can use the Home tab to access the features and functions necessary to build an application, create new components, open existing components, run existing components, and edit components. If you have RStat installed, the Modeling group will be present on the Home tab. The Home tab is the only tab in App Studio that is always present.

The following image shows the Main toolbar and Commands toolbar in Developer Studio.



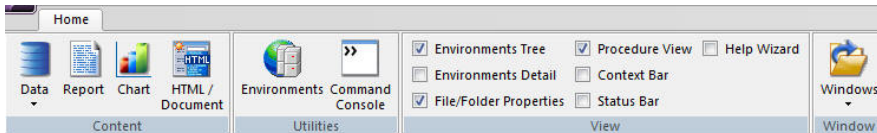
The following image shows the Application menu in App Studio.



The following image shows the Quick Access Toolbar in App Studio.



The following image shows the Home tab in App Studio.



The Application Menu, in App Studio, has an area for recently used documents. This allows you to quickly access a previously active document.

You cannot edit the commands on the Main toolbar and Commands toolbar. However, the Quick Access Toolbar can be customized to contain your most frequently used commands. You can use this to avoid switching between tabs to click a command.

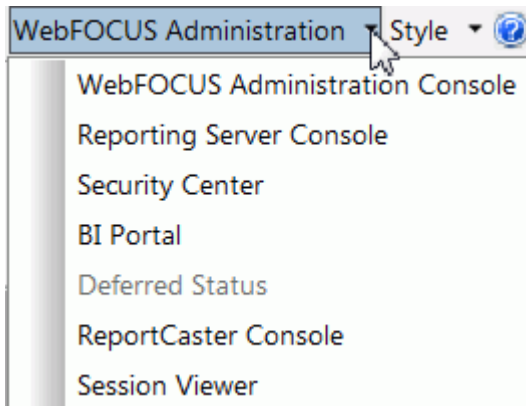
Developer Studio does not clearly display what content you can create. You need to know what tool to use and in which area that tool can be used. App Studio requires that you only know what content you want to create. Once you know what content you want to create, you can use the Home tab. The Content group contains commands that you can use to begin creating content.

WebFOCUS Administration

This section identifies the differences between Developer Studio and App Studio when managing your environments. In Developer Studio, you use the WebFOCUS Environments toolbar to access administration options such as the BI Portal, the ReportCaster Console, the WebFOCUS Administration Console, and others. This toolbar is only available when you select a server in the Developer Studio Explorer. The WebFOCUS Environments toolbar is shown in the following image.



In App Studio, you can access the same administration options from the WebFOCUS Administration menu, which is located in the upper-right corner of the ribbon. The WebFOCUS Administration menu is shown in the following image.



The WebFOCUS Administration menu is only available when you select an environment in the Environments Tree panel. The options available to you change, depending on the development area that you select in the Environments Tree panel. Deferred Status is only available when you select a Domains node in the Environments Tree panel. The Reporting Server Console is only available when you select a Web Applications node in the Environments Tree panel.

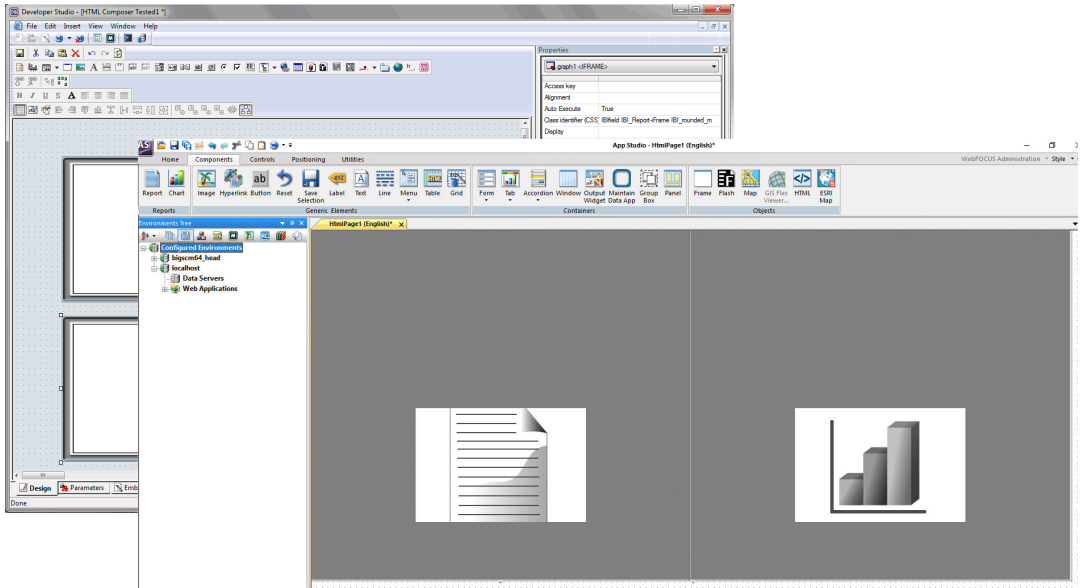
Development Areas

This section identifies the differences between Developer Studio and App Studio when creating different content. In Developer Studio, you create content with different tools, while in App Studio, you create content using different canvases.

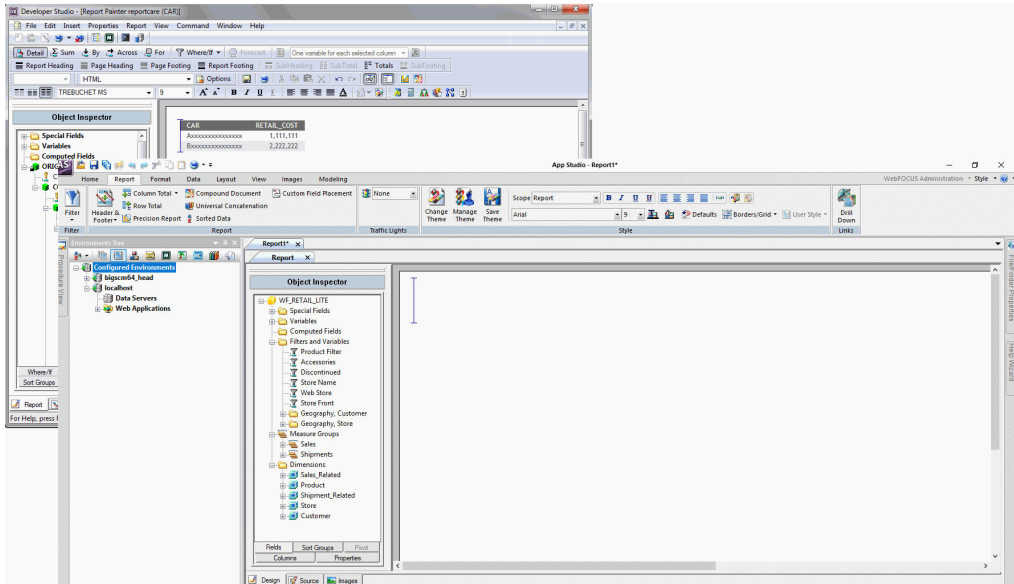
Developer Studio offers a selection of graphical development tools for creating content. These tools are contained in their own windows and identified by their distinctive names. Each tool has a set of toolbars that you can use to create your content. Some tools, such as HTML Composer, have multiple windows and dialog boxes that are used for setting the properties of objects.

The ribbon is contextual. As you develop content in a specific canvas, tabs open to assist you with development. For example, if you are creating a report using the Report canvas, the ribbon will only open the tabs that you need to build a report. If the HTML canvas is active, the ribbon opens the tabs needed to build an HTML page. Some canvases also use panels to develop content. These panels replace some of the windows and dialog boxes found in the Developer Studio tools. For example, the Properties and Settings dialog box, found in HTML Composer, is now the Settings panel in the HTML canvas. Not all canvases have panels associated with them. Some panels are only available for specific canvases. For example, the Settings panel only affects the HTML canvas, and will only be available when the HTML canvas is active. If the Report canvas is active, the Settings panel is not available.

The following image shows a comparison of the HTML Composer in Developer Studio, and the HTML canvas in App Studio.

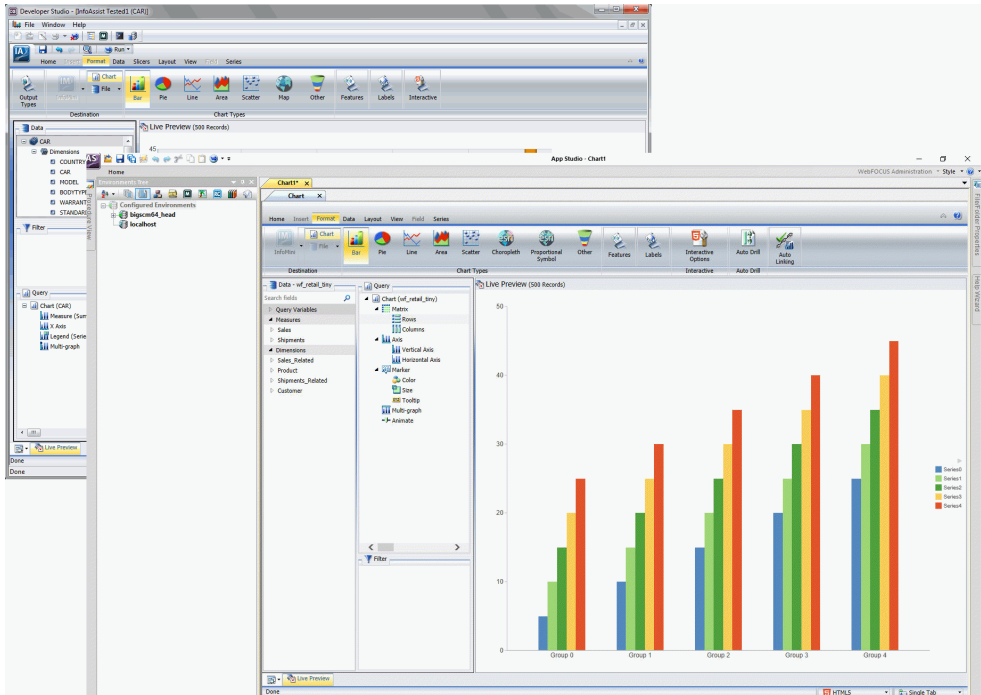


The following image shows a comparison of Report Painter in Developer Studio, and the Report canvas in App Studio.



Creating Procedures

The following image shows a comparison of the Graph tool in Developer Studio, and the Chart canvas in App Studio.



Each tool in Developer Studio opens in its own window. If you have multiple tools open, it can become difficult to navigate between them. Each canvas in App Studio has its own tab. You can easily switch between the different canvases that you open, by clicking the tab of the canvas you want to use.

Some tools in Developer Studio have multiple toolbars stacked on top of one another, which could be confusing to find a specific option. In App Studio, each command is categorized in a group, which is also categorized in a tab on the ribbon. On the ribbon, each tab name indicates the type of commands you will find in that tab. For example, the Components tab, in the HTML canvas, contains the components you can add to your HTML page.

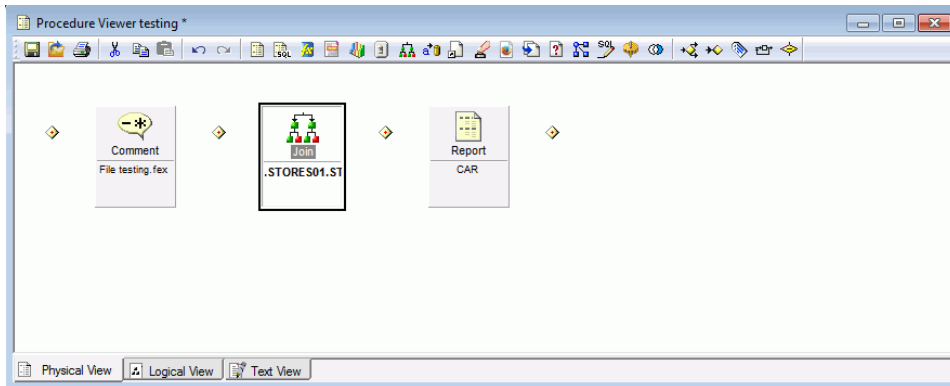
Creating Procedures

This section identifies the differences between the Procedure Viewer in Developer Studio, and the Procedure View panel in App Studio. Both are used to create multi-component procedures. However, one is a window and the other is a panel.

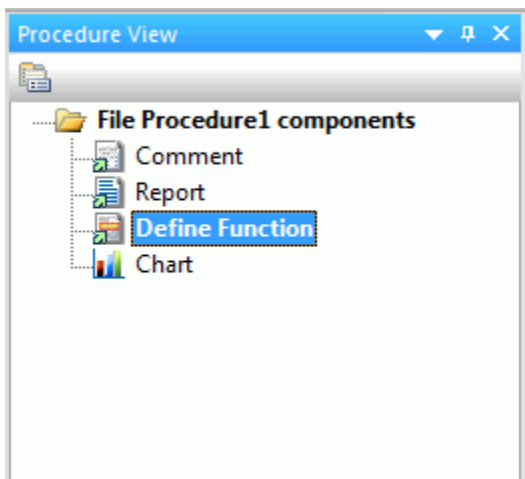
In Developer Studio, you use the Procedure Viewer to connect multiple components together, using the component connectors. The Procedure Viewer opens in a new window. When you create a new component, that component also opens in a new window or tool. The components that you add to the procedure are shown in the toolbar at the top of the Procedure Viewer, or are shown as options when you click the component connector.

In App Studio, you can use the Procedure View panel to create multi-component procedures. Right-click a component or the procedure folder, point to *New*, and then click the name of the component that you want to add. The component will be added to the procedure after the component you right-clicked. If a component in the procedure is open, a green arrow is shown next to the component icon.

The following image shows the Procedure Viewer in Developer Studio.



The following image shows the Procedure View panel in App Studio.



As you create your multi-component procedure, you can begin to accumulate multiple windows. This can increase the difficulty in navigating between multiple tools. App Studio displays each component as a tab underneath a procedure tab. This makes navigating between canvases simple, and fast. You can move the Procedure View panel around the canvas or set it to auto-hide so that you can access it without compromising development space.

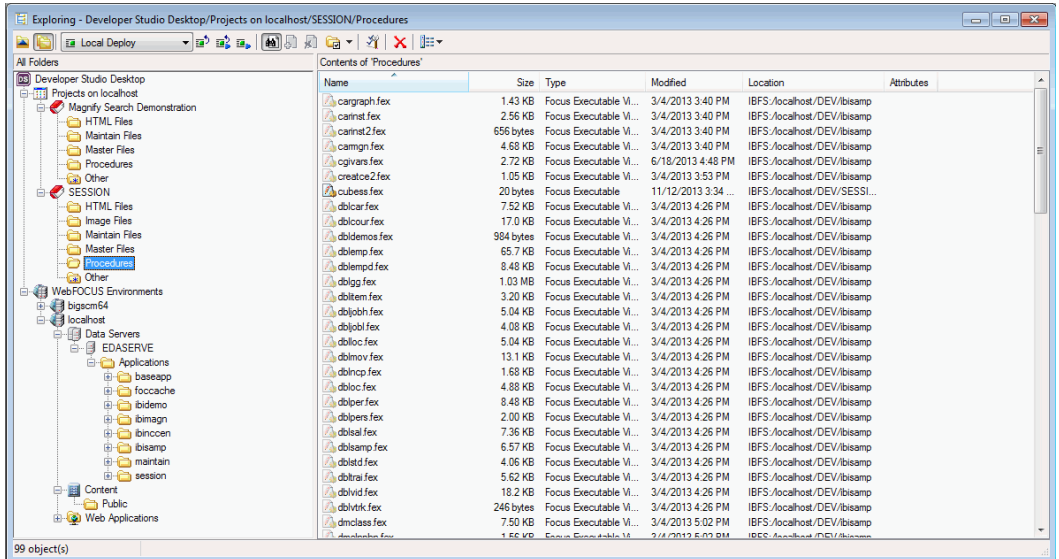
Configuring Environments

This section describes the difference between the Explorer window in Developer Studio and the Environments Tree panel in App Studio.

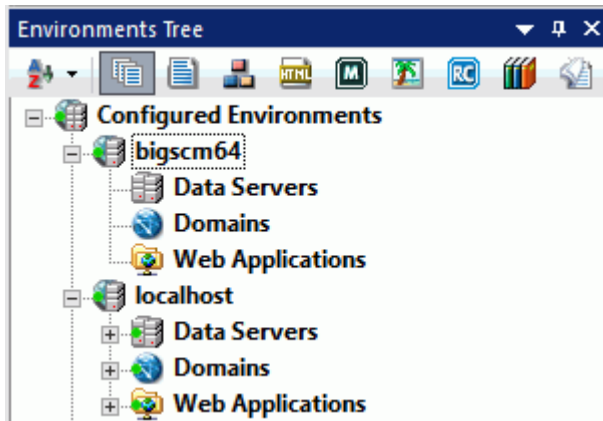
Developer Studio uses the Explorer window to manage your WebFOCUS environments and content. You can expand the project folder where your files are grouped into different virtual folders depending on their type. Different options are available based on the area that you are developing in.

App Studio uses the Environments Tree panel to manage your WebFOCUS environments and content. You can expand nodes on the Configured Environments tree to view your projects and application files. You can click the filtering commands on the top of the Environments Tree panel to view a specific file type within a folder. App Studio also allows you to open existing content, create new content, and then develop your application by accessing shortcut menus from the Configured Environments tree. If you double-click a file name, it will open in the appropriate mode or canvas. If you right-click an application folder, project folder, or file within the Configured Environments tree, a shortcut menu opens. The options in each shortcut menu depend on the development area or file type that you select.

The following image shows the Explorer window in Developer Studio.



The following image shows the Environments Tree panel in App Studio.



A key difference between the Explorer window and the Environments Tree panel is the ability to filter content by file type. In App Studio, you can use the filtering options at the top of the Environments Tree panel to view a specific type of file. These files are no longer kept in different virtual folders like they were in Developer Studio. You can also sort the order in which you view the content. For example, you can view the content in descending or ascending alphabetical order.

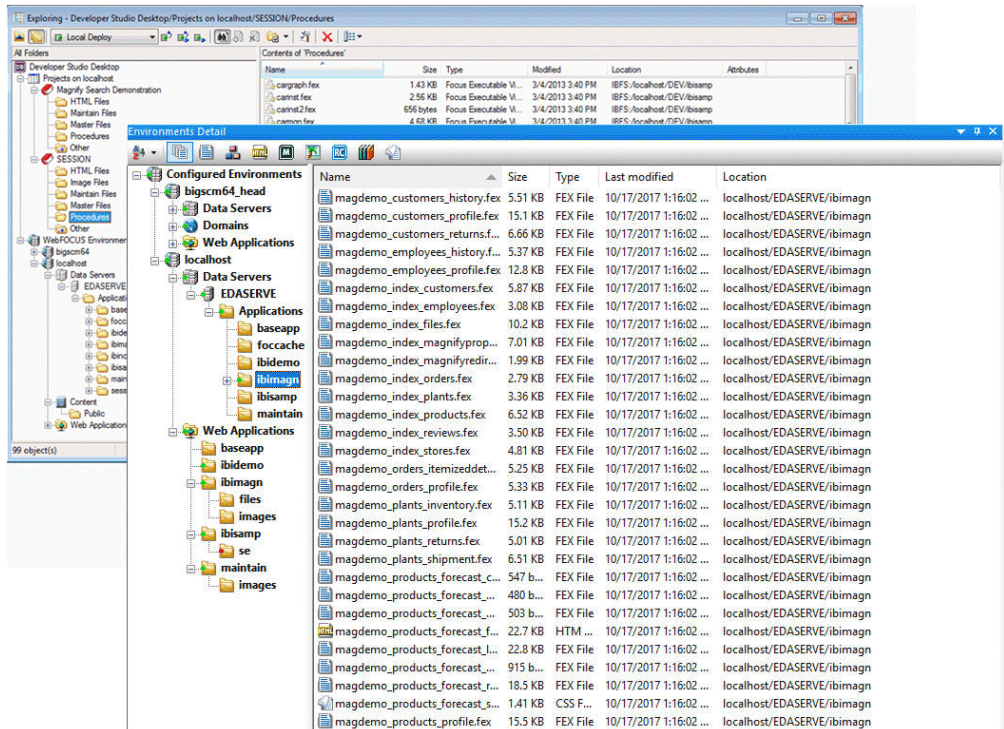
By default, the Environments Tree panel appears to the left of the App Studio interface. You can resize, move, or auto-hide this panel anywhere around the interface.

Environments Detail Panel

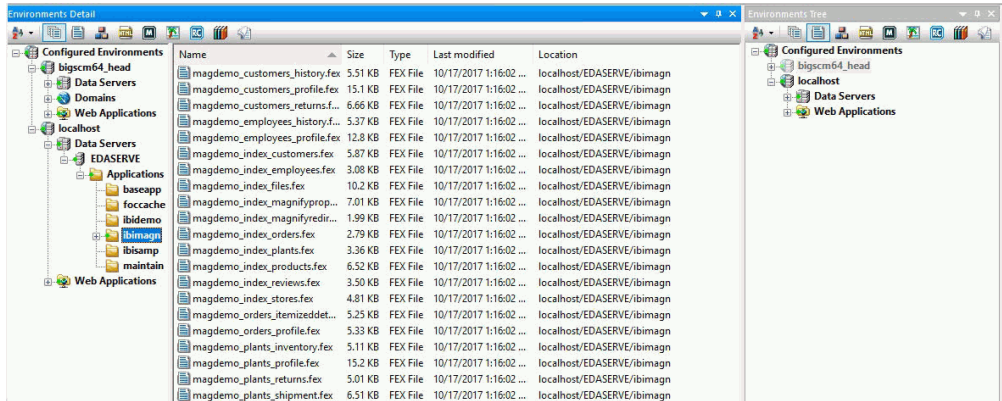
The Environments Detail panel is an optional panel you can use to display your configured environments. This panel functions similar to the Environments Tree panel in App Studio but resembles the Explorer window in Developer Studio.

The Environments Detail panel is similar to the Environments Tree panel, except that content is displayed in the Details pane rather than under the Configured Environments tree. Do not use the View Options command to sort content in the Environments Detail panel. Instead, click the columns at the top of the Details pane to sort content by either name, size, type, last modified, or location.

The following image shows a side-by-side comparison of the Environments Detail panel in App Studio and the Explorer window in Developer Studio.

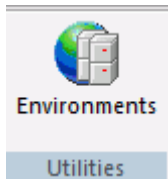


The following image shows the Environments Detail panel and the Environments Tree panel in App Studio.



Managing Your WebFOCUS Environment

In App Studio, you can manage your WebFOCUS environments by clicking the *Environments* command on the *Home* tab, in the *Utilities* group, as shown in the following image.



This command opens the Environments List dialog box, similar to Developer Studio. You can add a new environment, remove an environment, or edit the properties of an existing environment. For more information on configuring your WebFOCUS environments, see the *WebFOCUS App Studio User's Manual*.

Filtering Files

Developer Studio and App Studio separate application folders and project folders differently. In Developer Studio, virtual folders are used to group your content by file type. App Studio does not use virtual folders. Instead, use the filtering options at the top of the Environments Tree panel to view the content in an application folder or project folder. The Environments Tree toolbar filtering options are shown in the following image.



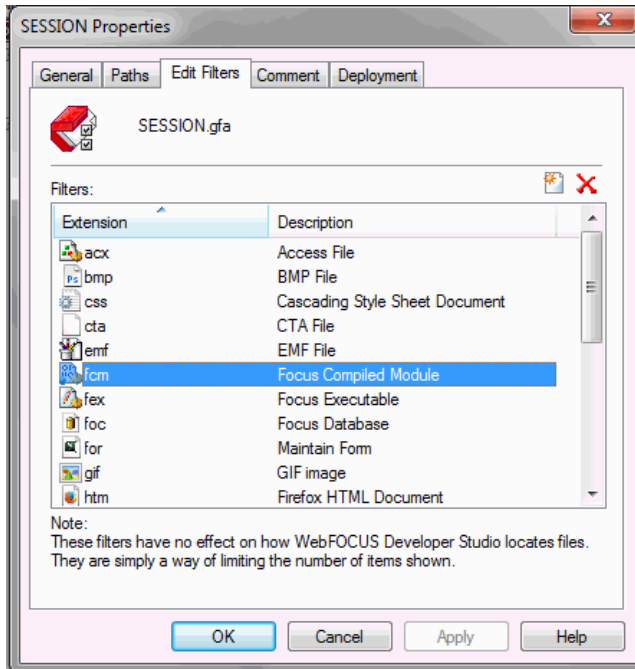
You can use the following filtering options to sort specific files:

- View Options.** Opens a menu where you can select how you want to view the sorted content. You can change the folders to view your content in ascending or descending alphabetical order.
- Show All files.** Displays all files in a folder.
- Show only Procedure files.** Displays only the procedure files in a folder.
- Show only Master Files.** Displays only the Master Files in a folder.
- Show only HTML files.** Displays only the HTML files in a folder.
- Show only Maintain files.** Displays only the Maintain files in a folder.
- Show only Image files.** Displays only the image files in a folder.
- Show only ReportCaster files.** Displays only the ReportCaster files in a folder.
- Show only Library files.** Displays only the library files in a folder.
- Other files.** Displays any content that is not a procedure file, Master File, HTML file, Maintain file, or an image file.

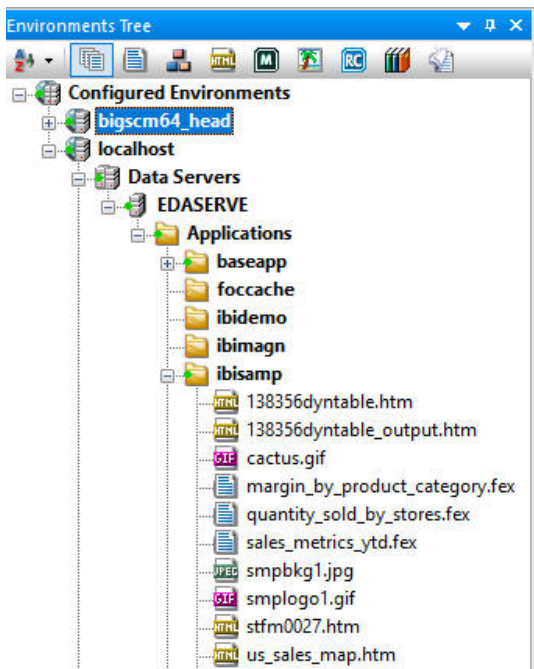
Filtering for Additional File Types

In Developer Studio, to filter and view additional file types in the Explorer window, along with the default filtered file types, you must create a new filter. To do this, you must specify the file type in the Edit Filters tab, in the Properties dialog box. You can access the Properties dialog box from the project folder shortcut menu.

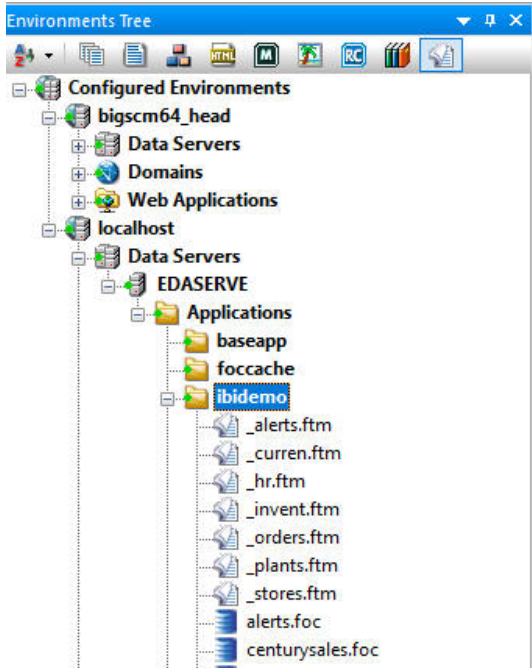
The Edit Filters tab is shown in the following image.



In App Studio, you do not need to create filters to view additional file types. Instead, you can use the *Show All files* filtering command, located on the Environments Tree toolbar. This command displays all files in a project or folder, regardless of their file type. The *Show All files* command is selected in the following image.

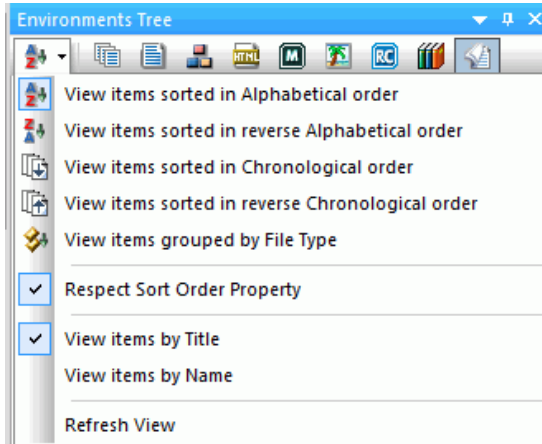


In App Studio, you can also use the *Other files* command, located on the Environments Tree toolbar, to display files that are not procedure files, Master Files, HTML files, Maintain files, or image files. The *Other files* command is selected in the following image.



View Options

App Studio allows you to sort and view content differently using the View Options menu. This menu contains options to sort the content alphabetically, by date modified, or by file type. You can also switch between viewing the content by title or by name. The View Options menu is shown in the following image.



You can use the following view options to sort content in a specific order:

- View items sorted in Alphabetical order.** Sorts the content in alphabetic order from A to Z.
- View items sorted in reverse Alphabetical order.** Sorts the content in reverse alphabetic order from Z to A.
- View items sorted in Chronological order.** Sorts content from the most recently written file to the oldest written file.
- View items sorted in reverse Chronological order.** Sorts content from the oldest written file to the most recently written file.
- View items grouped by File Type.** Sorts the content by file type. This is used when using the Other files sort option.
- Respect Sort Order Property.** Respects the Sort Order Property value specified in the File/Folder Properties panel.
- View items by Title.** Displays all content by title. If any content does not have a title, the name of the content will be used instead.
- View items by Name.** Displays all content by name.

- ❑ **Refresh View.** Refreshes what is currently visible on your screen, in the Environments Tree panel. This also refreshes any dependencies of the files shown in the Environments Tree panel, even if they are not in view.

Chapter 3

Creating Procedures

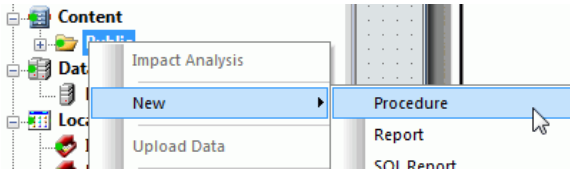
This topic describes how to create a procedure in App Studio. You can add the following components to a procedure: Report, Define, Define Function, Chart, Set, Join, Use, Allocation, HTMLForm, OLAP Dimensions, Match, or Dialogue Manager. Add these components to a procedure using the Procedure View panel, similar to the way that you use the Procedure Viewer in Developer Studio.

In this chapter:

- [Accessing and Using the Procedure View Panel](#)

Accessing and Using the Procedure View Panel

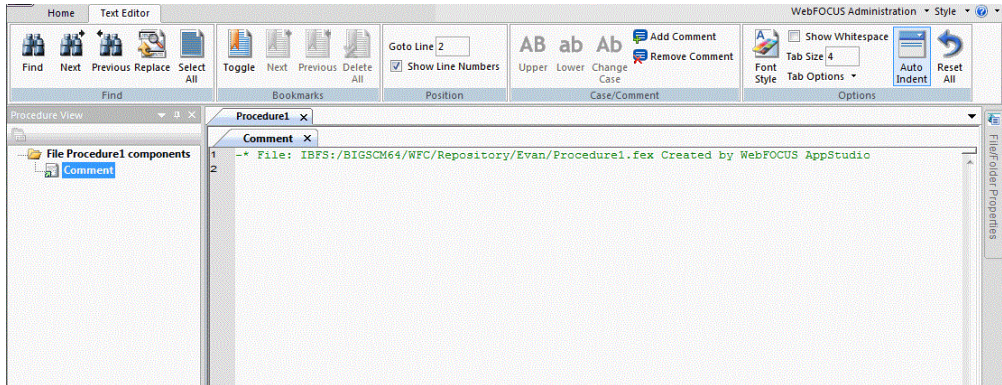
In App Studio, you can use the Procedure View panel to create multi-component procedures. You can access the Procedure View panel by right-clicking an application folder or project in the Environments Tree panel, pointing to *New*, and then clicking *Procedure*, as shown in the following image.



The Procedure View panel opens.

Accessing and Using the Procedure View Panel

After creating your procedure, the Procedure View panel and Procedure canvas open, as shown in the following image. The first component of every procedure file is a comment component. You can close this canvas and create your procedure or leave it open.

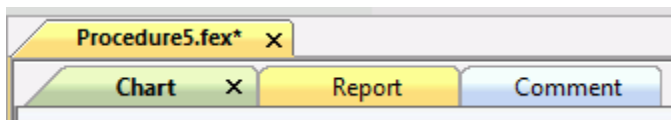


To create new components in the Procedure View panel, right-click the comment or the procedure folder, point to *New*, and click the component type you want to create. The component will be added to the procedure after the component that you right-clicked. For example, right-click on the comment file, point to *New*, and click *Report*.

If a component of the procedure is open, a green arrow is shown next to the component icon, as shown in the following image.



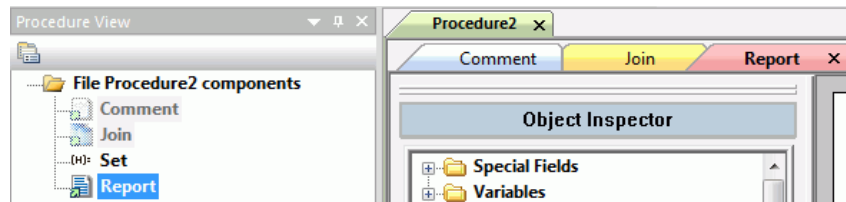
When multiple components are open at the same time, they are categorized underneath the procedure tab. The following image shows a chart component, report component, and comment component, all open under the Procedure5.fex tab.



This functionality allows you to switch between canvases quickly. You can close the components when you are done editing, and leave the components that you are still working on open. You can open and close these components as you develop your procedure.

Example: Multi-component Procedure

The following image shows a multi-component procedure in the Procedure View panel, with Comment, Join, Set, and Report components. The Comment, Join, and Report components are open. Notice that the Report canvas is currently active.



Note:

- The Procedure View panel is automatically available when you create a report.
- If you unintentionally close the Procedure View panel, select *Procedure View* on the *Home* tab, in the *View* group of the App Studio ribbon.

Creating Reports

This topic describes the differences between Developer Studio and App Studio when creating a report. In Developer Studio, you use toolbars to customize your report, while in App Studio, you use tabs on the ribbon to customize your report. Some of the key differences in the Report canvas, when compared to Report Painter, are which commands the Report Options dialog box and Field Properties dialog box display, how fields are added to the canvas, and how reports and SQL reports are created.

In this chapter:

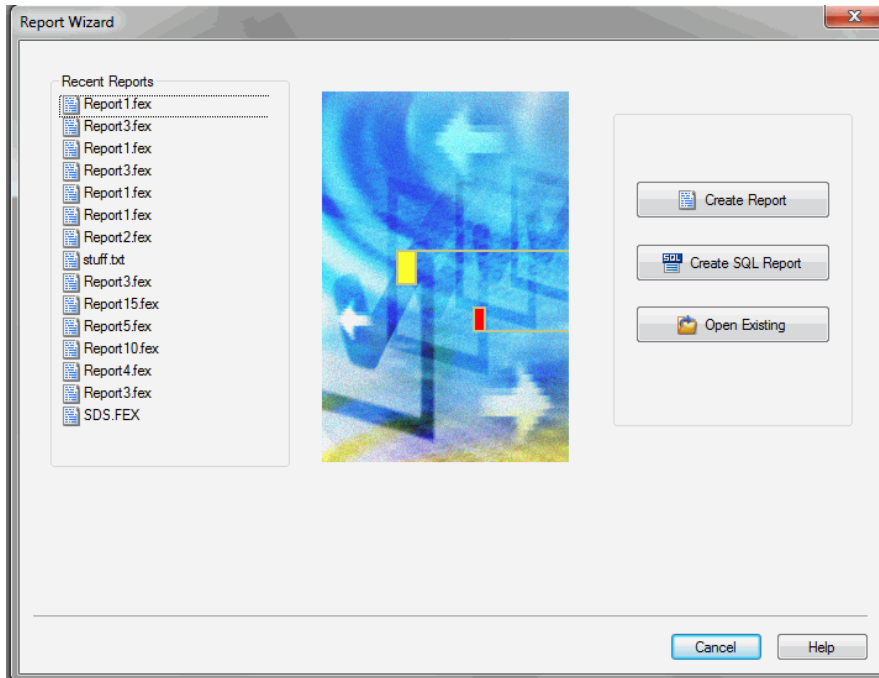
- [Creating Reports](#)
 - [Creating SQL Reports](#)
 - [Report Options Dialog Box](#)
 - [Field Properties Dialog Box](#)
 - [Adding Fields Using Smart Context](#)
 - [Field Length Limit](#)
 - [The Object Inspector Setup Tab](#)
 - [&WFFMT Report Variable](#)
-

Creating Reports

In Developer Studio, you create reports using Report Painter. You can access Report Painter from a component connector in the Procedure Viewer, or a shortcut menu in the Explorer window.

In App Studio, you can create reports using the Report canvas. When you create a new report, use the Report Wizard. You can access the Report Wizard from the Report command on the Home tab, or a shortcut menu in the Environments Tree panel. You can also create a report, as a component of a procedure, in the Procedure View panel. A report created in the Procedure View panel opens the Report canvas, but not the Report Wizard.

The Report Wizard is shown in the following image.

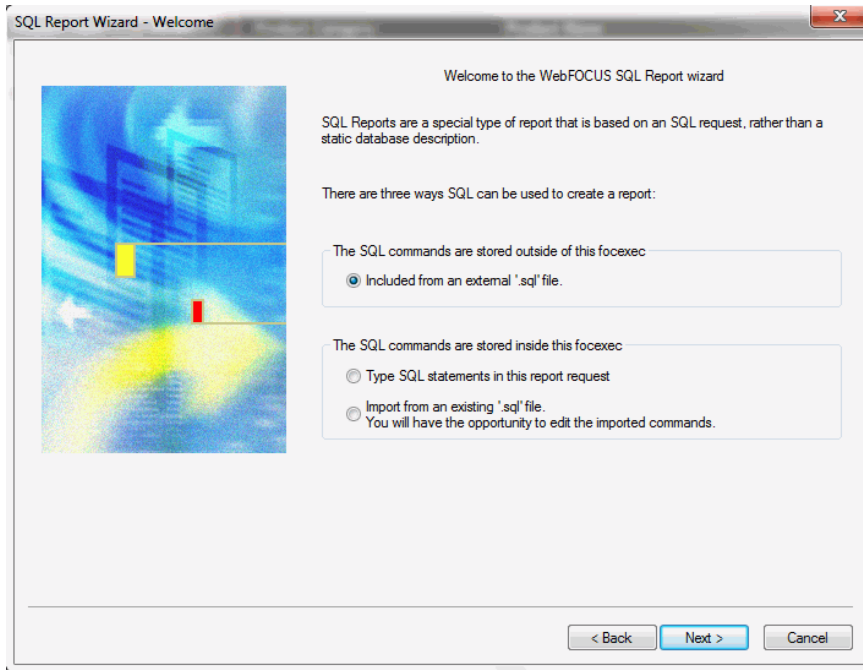


Creating SQL Reports

In Developer Studio, you create SQL reports using the SQL Report Wizard. You can access the SQL Report Wizard from a component connector in the Procedure Viewer, or a shortcut menu in the Explorer window.

In App Studio, you can create SQL reports using the Report Wizard. You can access the Report Wizard from the Report command on the Home tab, or a shortcut menu in the Environments Tree panel. You can also create an SQL report in the Procedure View panel.

The SQL Report Wizard is shown in the following image.



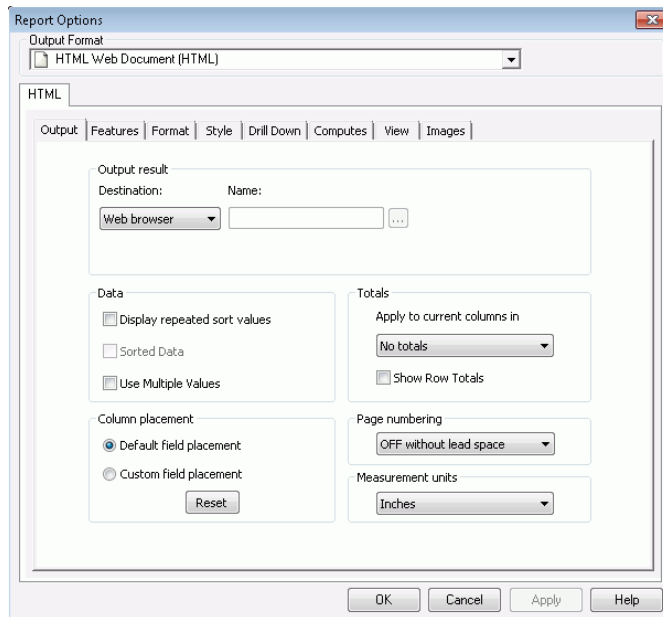
Report Options Dialog Box

In Developer Studio, the Report Options dialog box contains options that you can use to customize your report. This dialog box is categorized by tabs. For example, the Style tab contains styling options for your report.

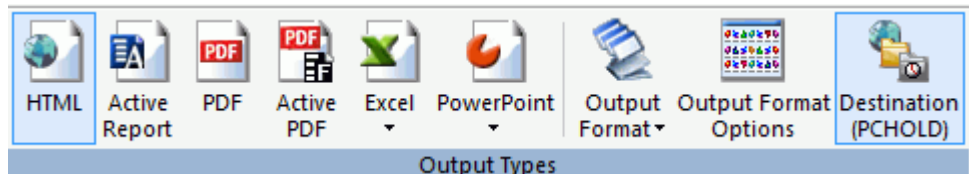
In App Studio, all of the options found in the Report Options dialog box have been moved to the ribbon tabs. Some of those commands can also be accessed from shortcut menus throughout the product. For example, you now use commands in the Output Types group, on the Format tab, to change the format of your report.

Output tab

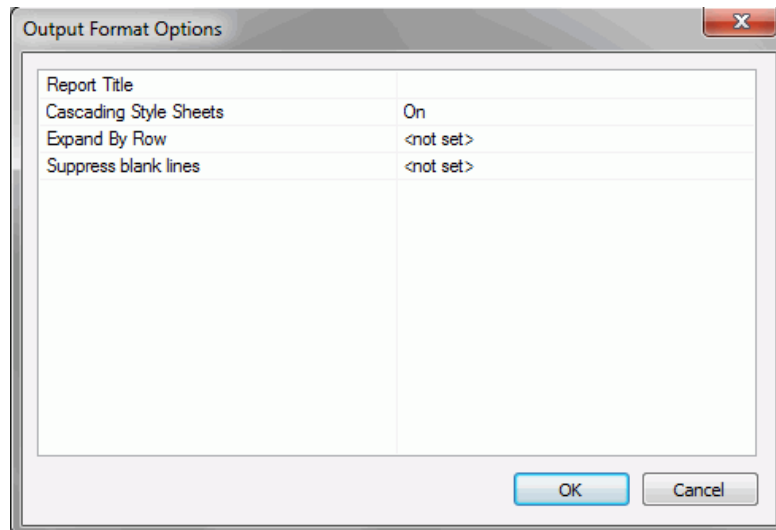
In Developer Studio, the Output tab contains options that change how your report is shown at run time, as well as the report destination. For example, you can choose to run your report as a PDF in a new window. The Output tab is shown in the following image.



In App Studio, some of those options are now on the ribbon. You can access them on the Format tab, in the Output Types group. From this group you can change the output format of your report and the destination of the report, when it is run. The Output Types group is shown in the following image.

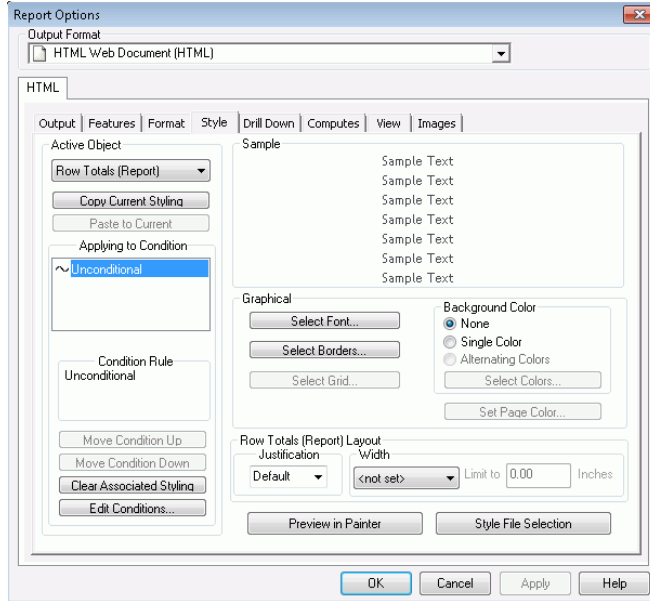


Some of the Developer Studio Output tab options are also found in the Output Format Options dialog box, accessed from the Output Format Options command, in the Output Types group, on the Format tab. The Output Format Options dialog box displays different options depending on the output format you select. The Output Format Options dialog box is shown in the following image.

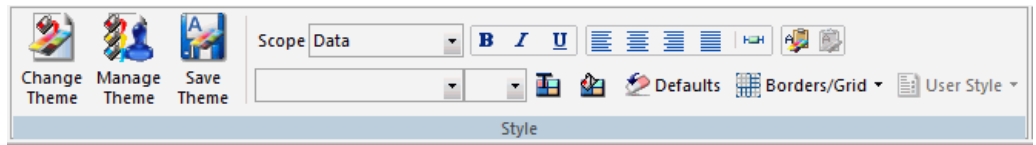


Style tab

In Developer Studio, the Style tab contains options that you can use style your report. The Style tab is shown in the following image.

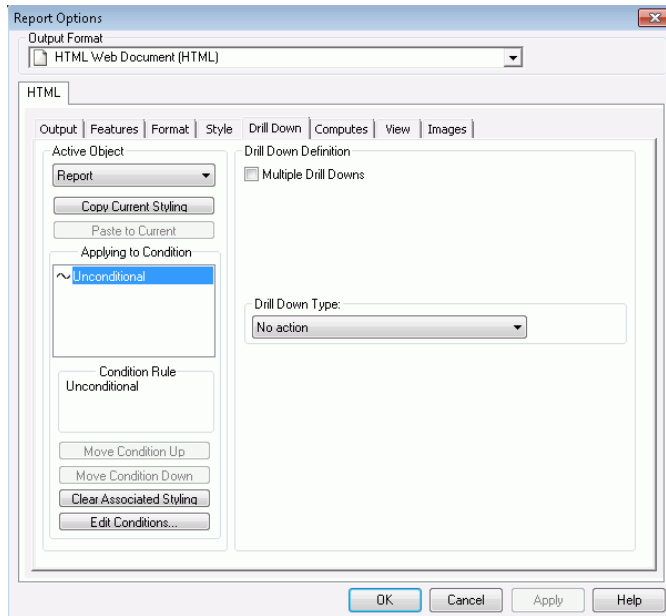


In App Studio, these options are now on the ribbon. You can access them on the Report tab, in the Style group. You can also manage the theme of your report from this tab. The Style group is shown in the following image.

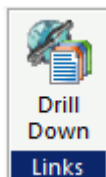


Drill Down tab

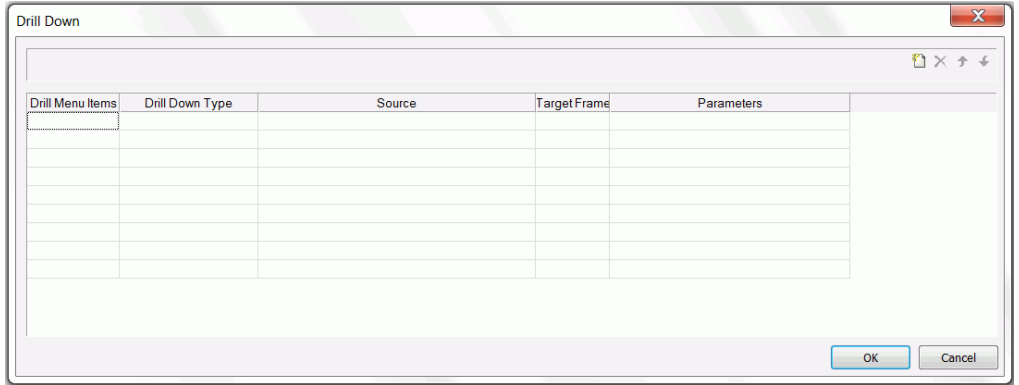
In Developer Studio, the Drill Down tab contains options that you can use to execute drill down actions when a field is clicked at run time. The Drill Down tab is shown in the following image.



In App Studio, these options are now on the ribbon. You can access them on the Report tab, in the Links group. Click *Drill Down* to open the Drill Down dialog box. The Drill Down command is shown in the following image.

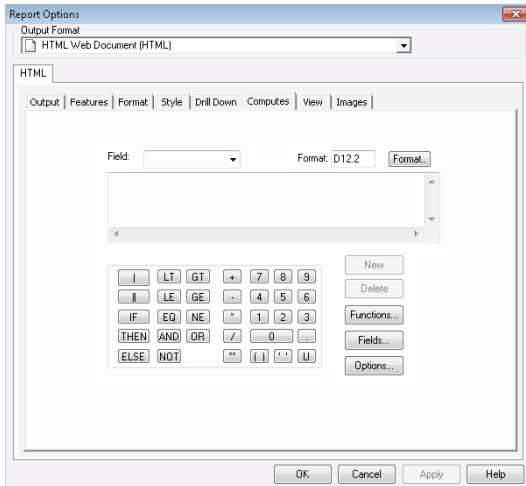


The Drill Down dialog box is shown in the following image.

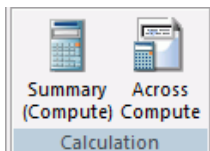


Computes tab

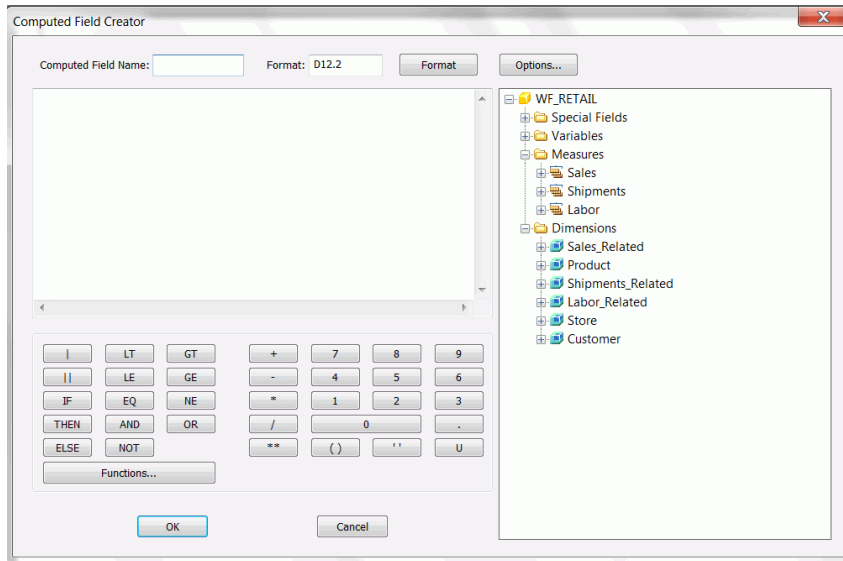
In Developer Studio, the Computes tab allows you create calculated computes within the tab. The Computes tab is shown in the following image.



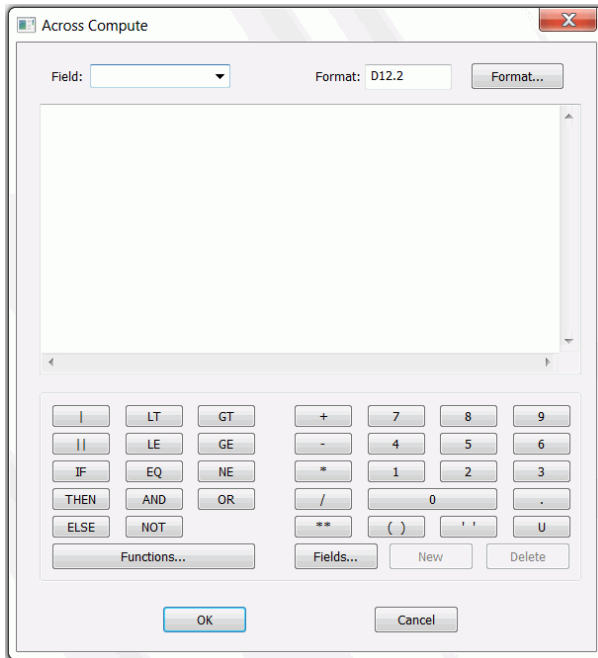
In App Studio, these options are now on the ribbon. You can access them on the Data tab, in the Calculation group. The Calculation group is shown in the following image.



The Summary (Compute) command opens the Computed Field Creator dialog box, as shown in the following image.

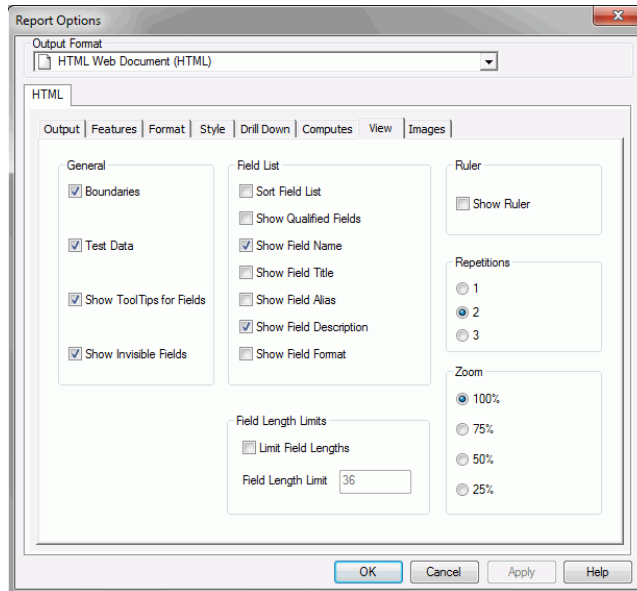


The Across Compute command opens the Across Compute dialog box, as shown in the following image.

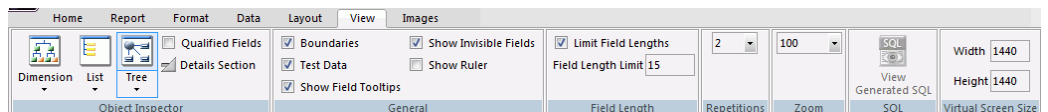


View tab

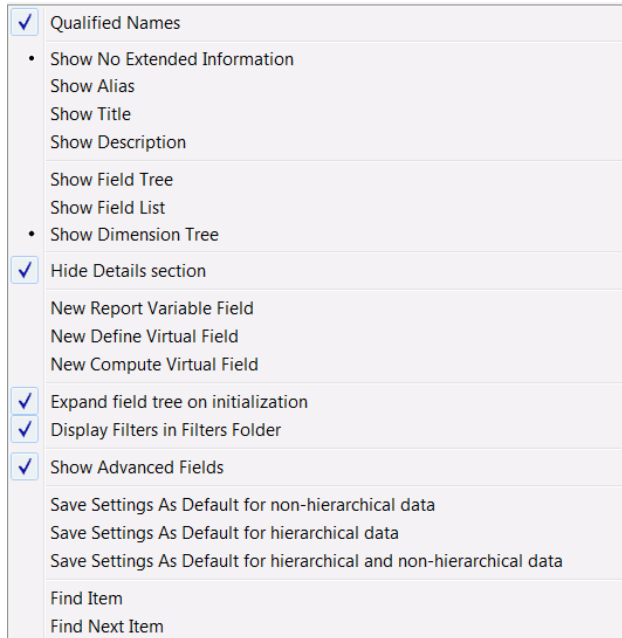
In Developer Studio, you can use the View tab to display report data differently and also to display the way fields are presented in the Object Inspector differently. The View tab is shown in the following image.



In App Studio, these options are now on the ribbon. You can access them on the View tab, in the Object Inspector group, General group, Field Length group, Repetitions group, and Zoom group. The View tab is shown in the following image.

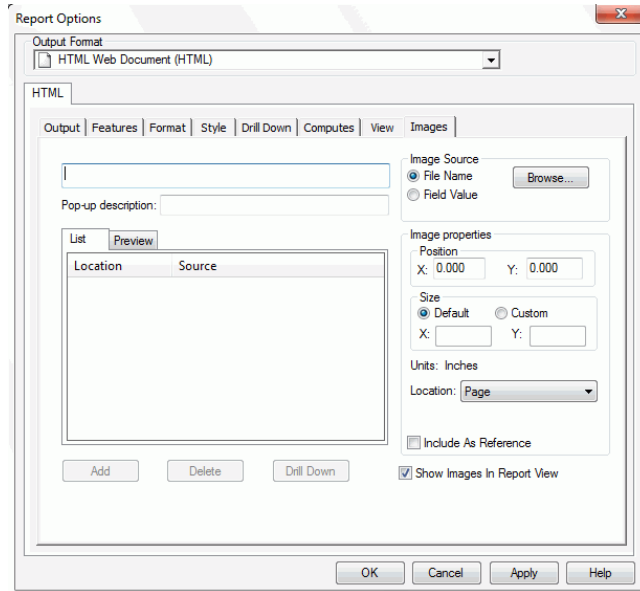


You can also access the Object Inspector group options from the Object Inspector shortcut menu. When an option is used in the Object Inspector group, that same option is selected in the Object Inspector shortcut menu. For example, if you choose to display the Field List by title, the Object Inspector shortcut menu will show that option as selected. The App Studio Object Inspector shortcut menu is shown in the following image.

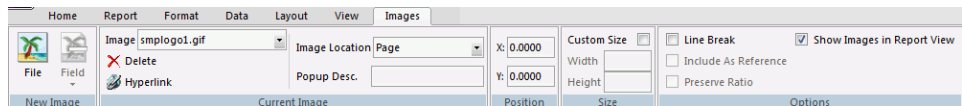


Images tab

In Developer Studio, you can use the Images tab to add images to your report. The Images tab is shown in the following image.



In App Studio, these options are now on the ribbon. You can access them on the Images tab, in the New Image group, Current Image group, Position group, Size group, and Option group. The Image tab is shown in the following image.



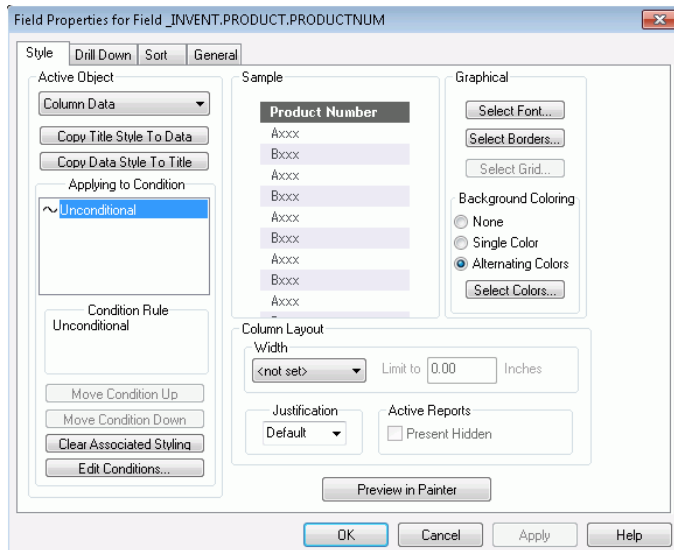
Field Properties Dialog Box

The Developer Studio Field Properties dialog box contains options that change the style and functionality of a selected field. The dialog box is categorized by tabs. These tabs indicate what the options affect. For example, the Sort tab contains sorting options that will modify the selected field.

Many of the options you find in the Field Properties dialog box in Developer Studio, are available in App Studio, on the ribbon, in the Field tab. The Field tab is only available when you select a field in your report. The General tab is only available for Detail and Aggregate field types. The Sort tab is only available for the Sort Down field type.

Style tab

In Developer Studio, you can style a selected field using the Style tab. The Style tab is shown in the following image.

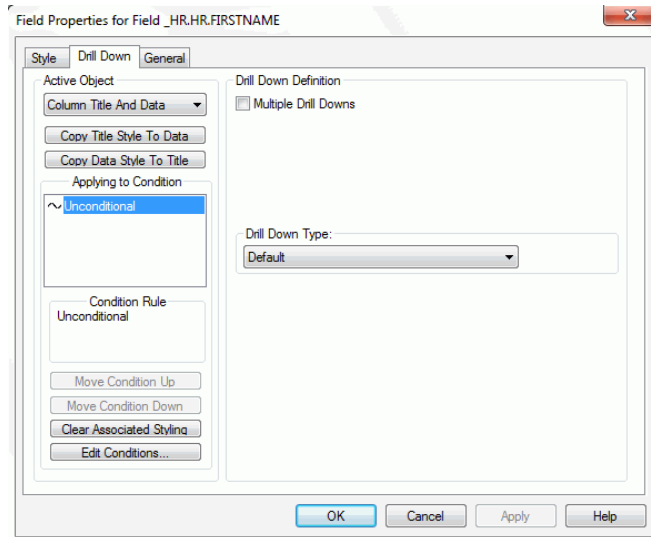


In App Studio, these options are now on the ribbon. You can access them on the Appearance tab, in the Style group. The Style group is shown in the following image.

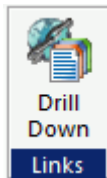


Drill Down tab

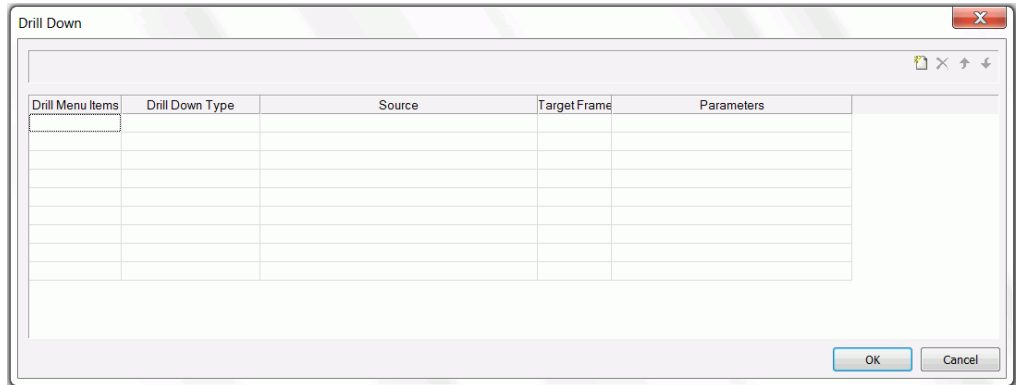
In Developer Studio, the Drill Down tab contains options that you can use to execute drill down actions when a field is clicked at run time. The Drill Down tab is shown in the following image.



In App Studio, these options are now on the ribbon. You can access them on the Appearance tab, in the Links group. Click *Drill Down* to open the Drill Down dialog box. The Drill Down command is shown in the following image.



The Drill Down dialog box is shown in the following image.



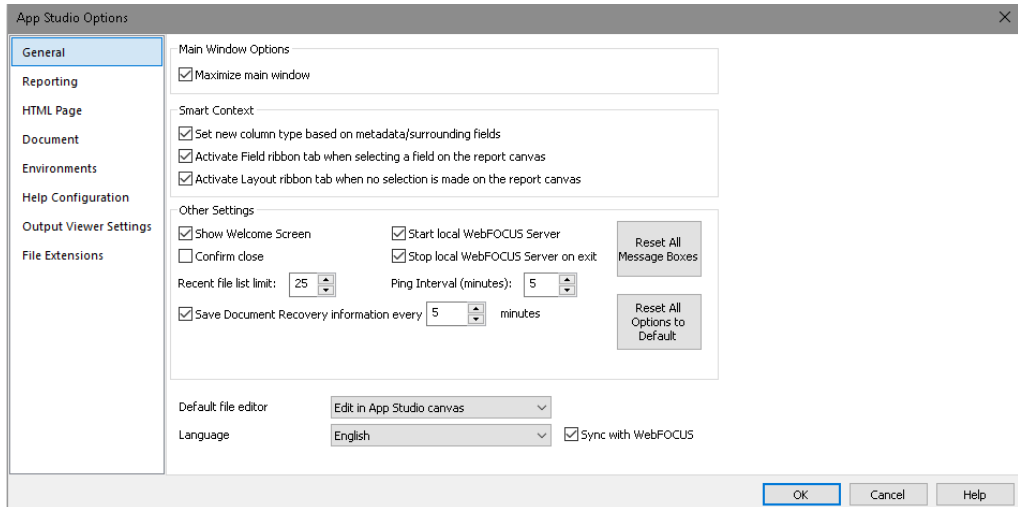
Adding Fields Using Smart Context

In App Studio, the Report canvas uses smart context functionality to determine what column type a field should be, given the metadata of the field and/or the position of the field on the canvas.

Depending on the metadata of the field, and whether you have the smart context option turned on or off, fields that you add to your report will be added as either an aggregate, BY, SUM, or Detail.

- If smart context is on, and the metadata indicates that a field should be a measure or dimension, it will automatically be added as an aggregate or sort down field type, respectively.
- If smart context is on, but the metadata does not indicate whether a field is a measure or dimension, then alphanumeric fields and date fields are added as BY fields, while numeric fields are added as SUM fields.
- If smart context is off, all fields are added as DETAIL fields.

The smart context options are contained in the App Studio Options dialog box, in the General tab, as shown in the following image.

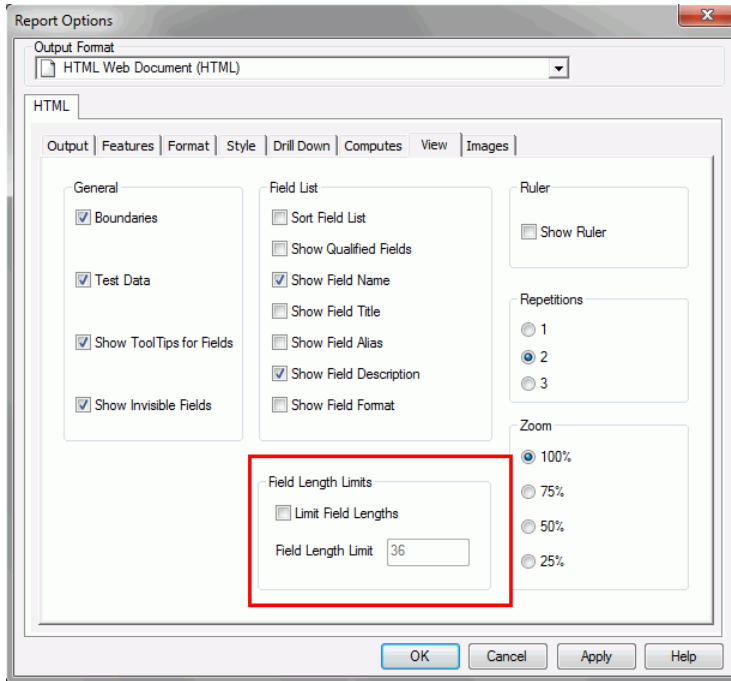


Field Length Limit

In Developer Studio, the Limit Field Lengths command is off by default. The Limit Field Lengths command is found in the View tab of the Report Options dialog box and controls the maximum length of a field or the maximum number of characters that can appear in the report. The Field Length Limit is set to 36 by default.

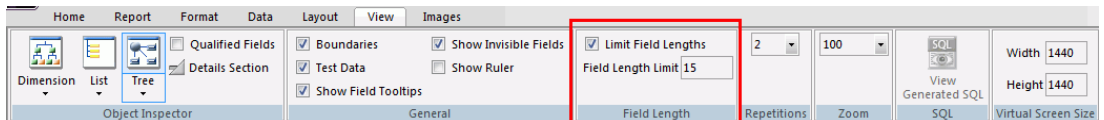
Field Length Limit

The Report Options dialog box is shown in the following image with the Limit Field Lengths command and Field Length Limit command highlighted.



In App Studio, these commands are now on the ribbon. You can access them on the View tab, in the Field Length group. The Limit Field Lengths command is on by default. The Field Length Limit is set to 15 by default. This is so a field does not display an overwhelming amount of information when initially added to the report.

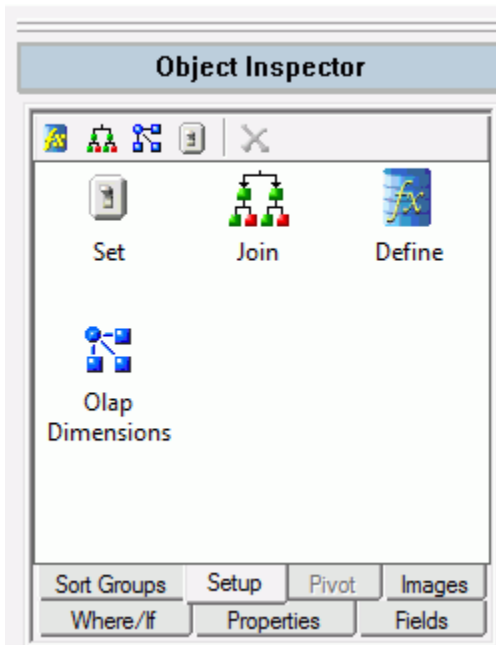
The View tab is shown in the following image with the Field Length group highlighted.



The Object Inspector Setup Tab

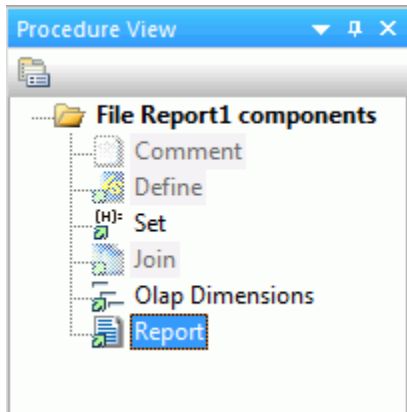
In Developer Studio, you use the Object Inspector Setup tab to create Defines, Joins, Sets, and OLAP Dimensions that precede your report.

The Setup tab is shown in the following image.



In App Studio, the Setup tab has been removed from the Object Inspector. To create Defines, Joins, Sets, and OLAP Dimensions that precede your report, you can use the Procedure View panel.

The Procedure View panel is shown in the following image.



For more information on using the Procedure View panel, see [Accessing and Using the Procedure View Panel](#) on page 33.

&WFFMT Report Variable

In Developer Studio, the &WFFMT report variable is created and available for selection from the Object Inspector when you select User as the output format.

In App Studio, the &WFFMT is no longer displayed in the Object Inspector when the User output format is selected. To edit the &WFFMT variable, on the *Format* tab, in the *Output Types* group, click *Output Format Options*.

Creating HTML Pages

This topic describes how to create an HTML page, and how to use the Properties panel, Settings panel, Requests & Data sources panel, and the Tasks & Animations panel. In Developer Studio, you use toolbars, windows, and dialog boxes to customize your HTML page. In App Studio, you use tabs and panels. Some of the key differences in the HTML canvas, when compared to HTML Composer are, how to set properties and settings, and how to create tasks and requests.

In this chapter:

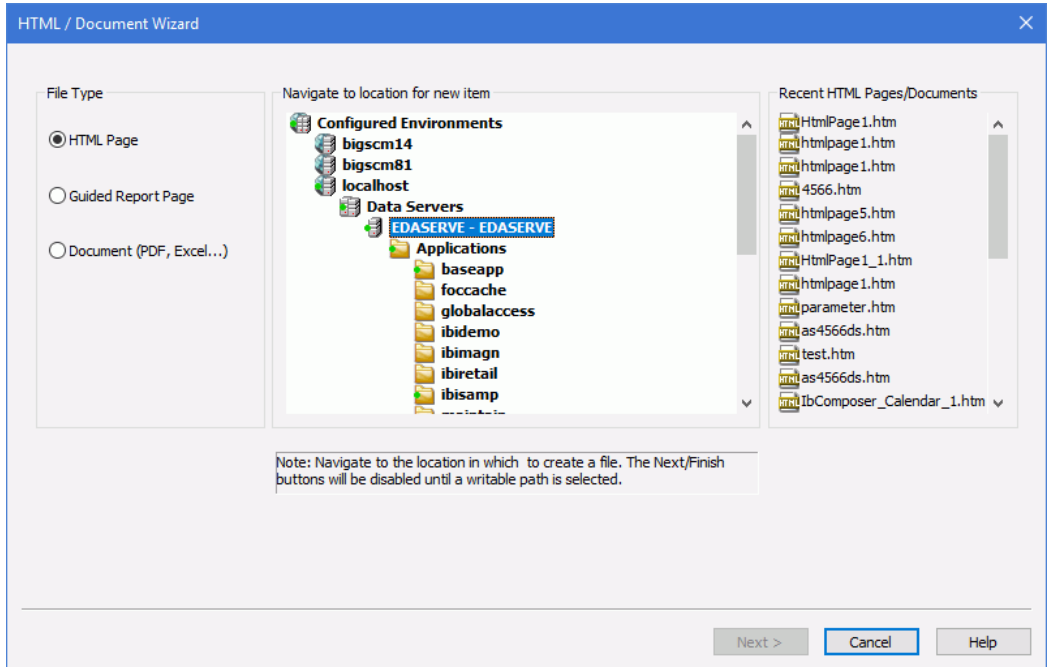
- [Creating HTML Pages](#)
 - [Using the Properties Panel](#)
 - [Using the Settings Panel](#)
 - [Requests & Data Sources Panel](#)
 - [Using the Tasks & Animations Panel](#)
 - [Generating a Chart From a Report](#)
 - [Making an HTML Page Responsive](#)
-

Creating HTML Pages

In Developer Studio, you create an HTML page using HTML Composer. You access HTML Composer from a shortcut menu in the Explorer window.

In App Studio, you create an HTML page using the HTML canvas. When you create a new HTML page, you use the HTML/Document Wizard. You can access the HTML/Document Wizard from the HTML/Document command on the Home tab or from the shortcut menu in the Environments Tree panel.

The HTML/Document Wizard is shown in the following image.



Recreating Save Selection Files

You can add a Save Selection button to your HTML page. At run time, the user can select given parameters and save them in a snapshot by clicking the Save Selection button. This creates a static HTML file in the same folder as the source file.


The Save Selection files that were created in Developer Studio can not be converted to App Studio.

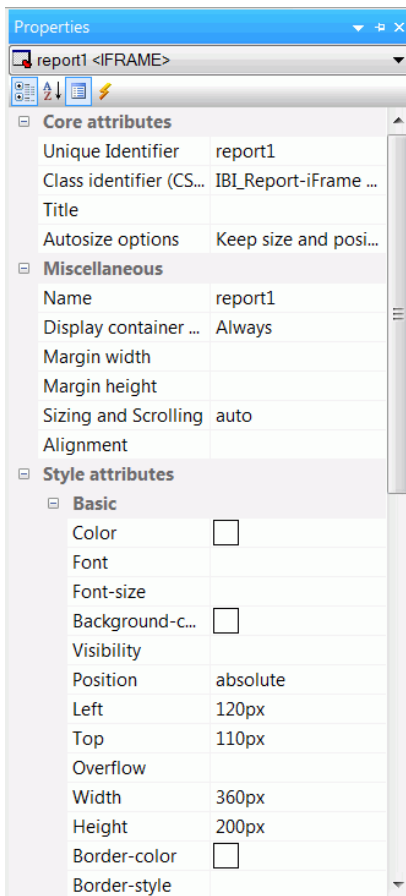
To recreate a Save Selection file, the user can run the source file in App Studio, select the desired parameters, and click Save Selection. The resulting Save Selection file is viewable in App Studio.

Using the Properties Panel


In Developer Studio, you use the Properties window to edit the properties of objects in your HTML page. In App Studio, this is done using the Properties panel. You can view properties or events using the commands at the top of the Properties panel. You cannot view both properties and events simultaneously.

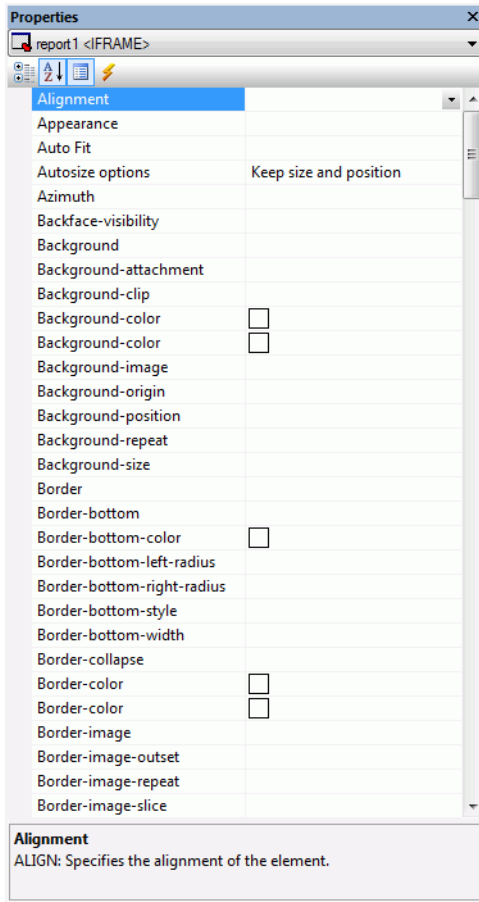
The Properties panel contains more advanced organization options than the Properties window. You can sort the properties in the Properties panel using the sort commands at the top of the panel. These sort options display the properties by category or in alphabetical order. Only one of these commands can be selected at a time.

The Categorized command  sorts the properties into separate categories, based on what that property does. For example, in the Properties panel, key properties are categorized under the Core attributes category, while stylistic properties are categorized under the Style category. You can collapse or expand property groups from view. The following image shows the Properties panel with the properties sorted by category.



Using the Settings Panel

The Alphabetical command  sorts the properties alphabetically, from a to z. The following image shows the Properties panel with the properties sorted alphabetically.



Using the Settings Panel

In Developer Studio, you use the Properties and Settings dialog box to modify the population settings of components, controls, and parameters. The settings that you modify change depending on which component, control, or parameter you select.

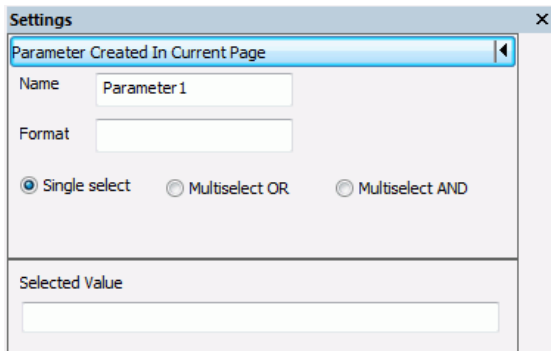
In App Studio, you use the Settings panel to modify the population settings of components, controls, and parameters. The Settings panel has multiple sections that are only available when you select an object or when no object is selected. This is similar to the way the Properties and Settings dialog box functions in Developer Studio.

When you select an object, or when you add an object to the canvas, the settings for that object display in the Settings panel. Settings are grouped under sections, and the sections and settings change depending on which component, control, or parameter is selected.

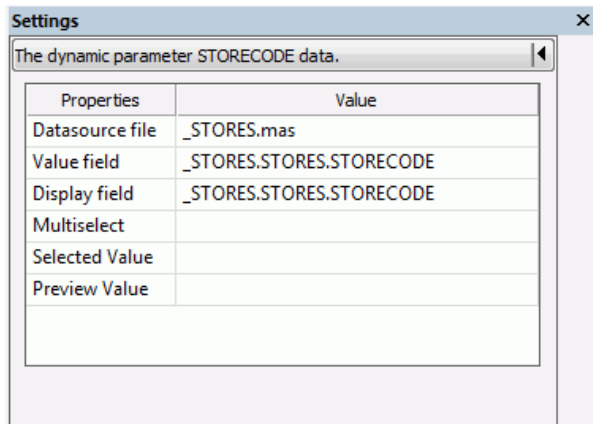
When no object is selected on the canvas, you can manage your external cascading style sheet files and JavaScript files. The Settings panel displays this section by default. You can add external cascading style sheet files and JavaScript files, located in your WebFOCUS environment, using the CSS button  and JavaScript button . You can reference web-accessible cascading style sheet files and JavaScript files by typing the URL of those files, in the URL/Find File area, as shown in the following image.



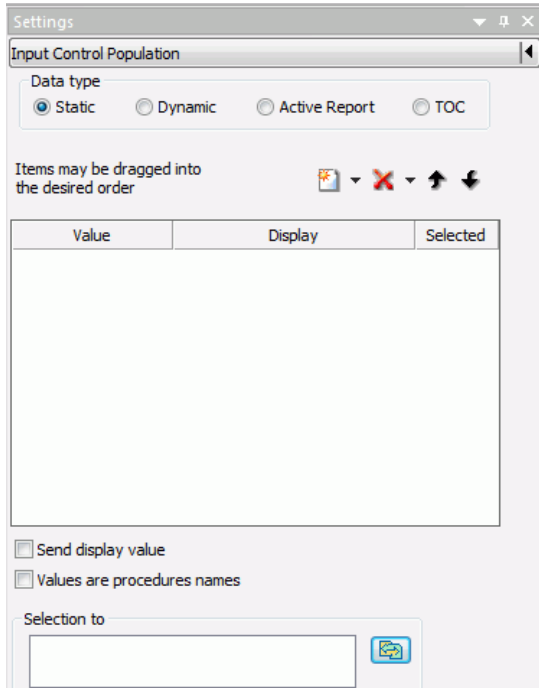
The following image shows the Settings panel for a parameter created in the HTML page.



The following image shows the Settings panel for a dynamic parameter.



The following image shows the Settings panel for an input control.



You can drag fields and parameters from the Requests & Data sources panel to the Settings panel when the Input Control Population section is available and the Data type is set to Dynamic.

Requests & Data Sources Panel

A request is a definition of something that can be executed. For example, you can execute an embedded procedure, external procedure, HTML page, URL, and more. If a request has parameters, those parameters will be indented and listed under the request name. By default, all parameters are sent to the Reporting Server. However, you can choose to not send a parameter by using the shortcut menu of the parameter, and clicking *Don't Send*. If you choose to not send a parameter, you must verify that your procedure can work without that value. For example, if the procedure contains a -DEFAULT value for the parameter.

When creating an external request that uses an existing file, the request inherits the name of that external file. For example, if you create an external request that uses an existing WebFOCUS procedure called `Parameter_Report`, then your external request will inherit the name `Parameter_Report`.

Each request displays two folders beneath it, the Columns and Parameters folders. The Columns folder contains the fields used in the request. The Parameters folder contains the parameters used in the request. The Parameters folder is expanded by default, while the Columns folder is collapsed by default.

The Data Sources folder contains all data sources used for embedded procedures and default control population requests in the HTML page. Referenced data sources are not displayed in the Requests & Data sources panel. You can add additional data sources to the HTML page by embedding additional requests. You can also add additional data sources by right-clicking on the Data Sources folder and using the shortcut menu to click *Add Data Source*.

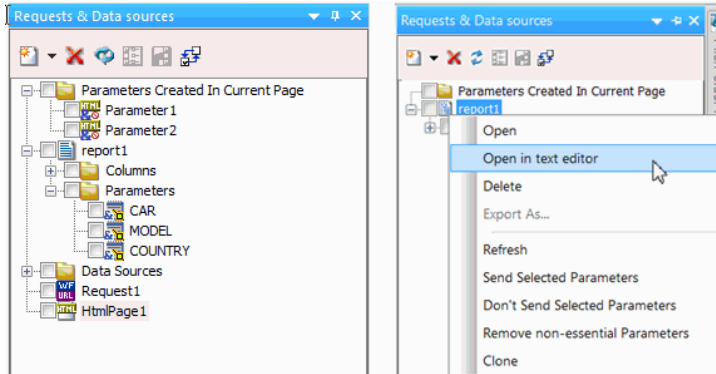
You can refresh all requests, except empty requests, using the shortcut menu. Right-click on a request and click *Refresh*.

You can also duplicate a request by using the shortcut menu on the request you want to duplicate, and clicking *Clone*. When you clone an embedded request, you can create a copy of the embedded request with the name, *name_1*. *Name* is the name of the original embedded request. For example, if you clone a request named *SalesReport*, the cloned request would be called *SalesReport_1*. Cloning an external request does not make a second copy of the referenced procedure.

You can also make edits to a selected procedure using the shortcut menu and clicking *Open in text editor*, as shown in the rightmost image below. For an embedded procedure, the new procedure tab opens with an extension of *.vrt* to show that it is virtual. With a referenced procedure, the procedure tab opens with an extension of *.fex*.

In the Requests & Data sources panel, the icon next to each item indicates the type of request or parameter. For example, a parameter that was created in the current HTML page has the parameter symbol and a plus sign on it. Another example is how an external URL request displays with the word *URL* on it.

Two sample Requests & Data sources panels are shown in the following images.



You can drag requests from the Requests & Data sources panel to the HTML canvas to create objects and controls for the request.

Procedure: How to Create an Embedded Procedure Request in App Studio

1. Add an embedded procedure to the Requests & Data sources panel. Click the *New* drop-down arrow, point to *Embedded Request*, and then click *New Report*, *New Chart*, or *New Document* to create and embed a new procedure.

or

Click *Import Existing* to embed an existing procedure file.

A new request is created in the Requests & Data sources panel. If you are creating a new procedure, the Open File dialog box opens, prompting you to select a Master File. If you are embedding an existing file, the Open File dialog opens, prompting you to select a procedure file.

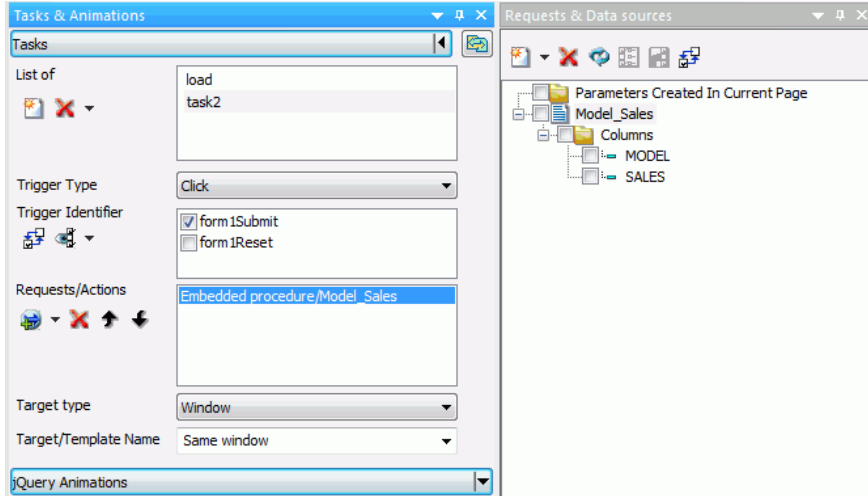
2. Select a Master File or procedure file and click *OK*.

If you are creating a new procedure, the canvas for that type of procedure opens. Once you are done creating the procedure, save and close the canvas.

3. In the Requests & Data sources panel, you may drag the request that you created onto the canvas.

Using the Tasks & Animations Panel

A procedure object is created on the canvas. If the procedure contains parameters, a form control is also added to the canvas. A task is created in the Tasks section of the Tasks & Animations panel. An example of the Requests & Data sources panel and the Tasks & Animations panel for a request is shown in the following image.

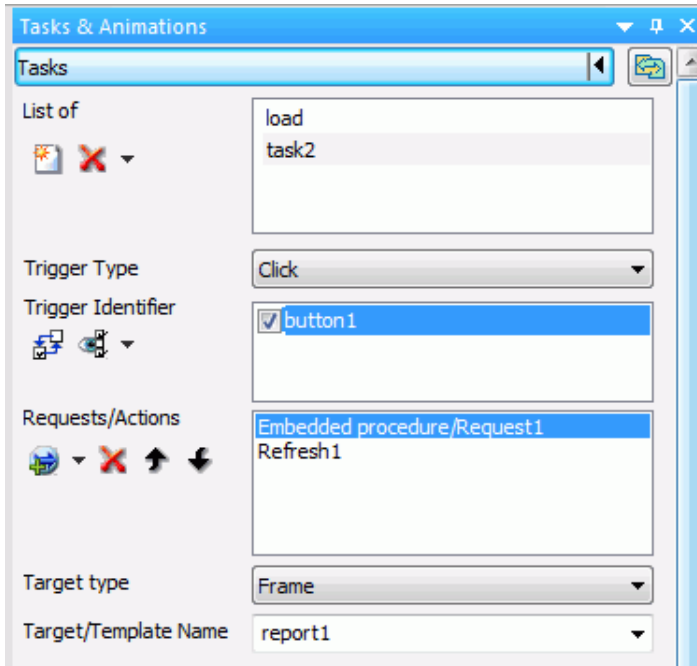


Using the Tasks & Animations Panel

The Tasks & Animations panel contains two sections: Tasks and jQuery Animations.

Tasks Section

The Tasks section of the Tasks & Animations panel allows you to execute a request when a certain criteria is met. For example, you can create a task that runs an embedded report when a button is clicked. You can use this section with the Request & Data sources panel and jQuery Animations section. An example of a task, created in the Tasks section, is shown in the following image.



Procedure: How to Create a Hyperlink in App Studio

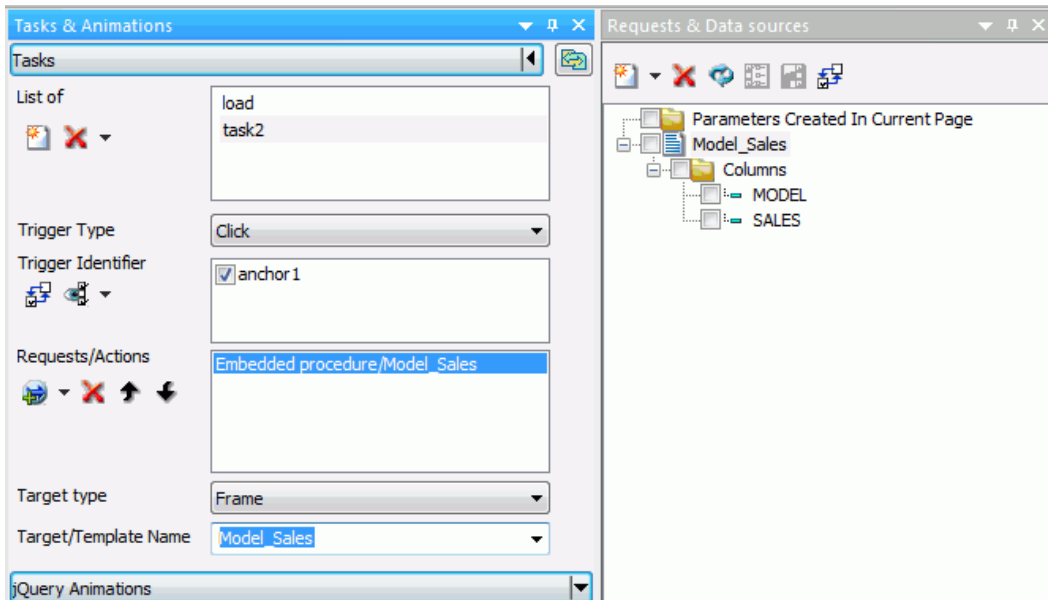
In Developer Studio, you use the Hyperlink Properties dialog box to create hyperlinks. In App Studio, use the Hyperlink component and the Tasks & Animations panel to create hyperlinks. The following is an example of creating a hyperlink.

1. In the *Components* tab, in the *Generic Elements* group, click *Hyperlink*.
A task is added to the Tasks section of the Tasks & Animations panel, with a trigger type and trigger identifier preselected.
2. In the Requests & Data sources panel, create a new external request for a WebFOCUS Procedure.
3. Drag the request on to the canvas to create a frame.

4. In the Tasks section of the Tasks & Animations panel, in the Requests/Actions list, click the Requests selection command, point to *Run Request*, and select the request you created in step 2.
5. In the Target Type drop-down list, select *Frame*.
6. In the Target/Template Name drop-down, select the frame you created in step 3.

When the HTML page is run and you click on the hyperlink, the WebFOCUS procedure you chose will run in the frame.

The following image shows the Requests & Data sources panel and the Tasks & Animations panel after creating this hyperlink example.



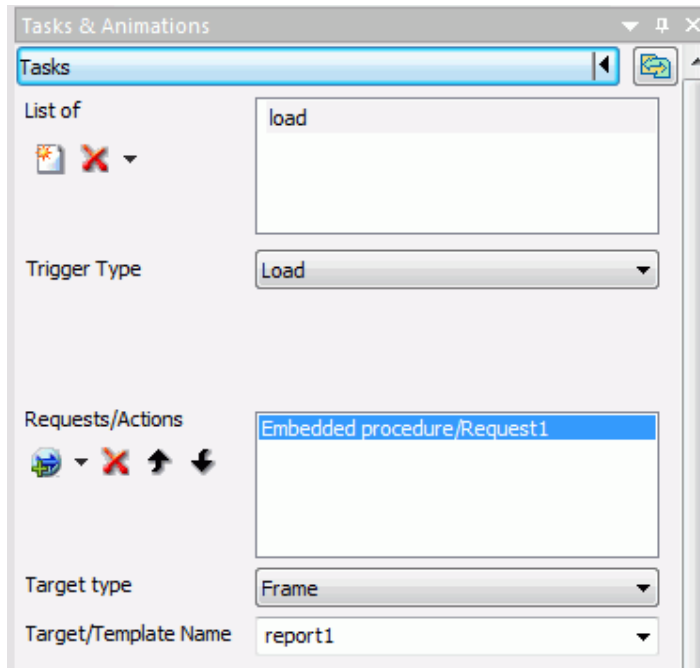
Procedure: How to Automatically Execute a Request When a Page is Loaded

In Developer Studio, you use the Auto Execute property, in the Properties window, to set a report or graph to execute when your HTML page is loaded. In App Studio, use the Requests & Data sources panel and the Tasks sections of the Tasks & Animations panel to create a request that executes when your HTML page is loaded. To do this, you must use the load task in the Task Section. The load task is automatically added to every page and runs all selected requests when the page initially loads.

1. In the Tasks section of the Tasks & Animations panel, select the *load* task.
2. In the Trigger Type drop-down list, select the *Load* trigger type. This option is selected by default for a load task.

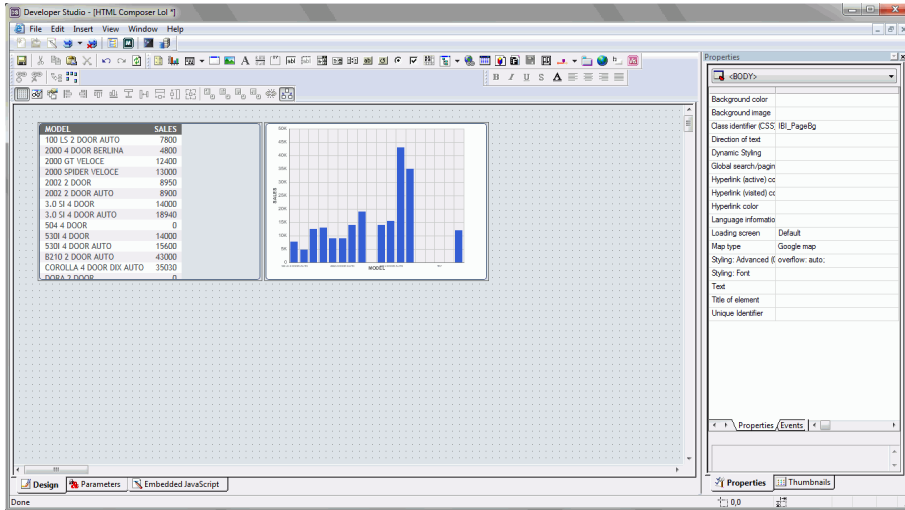
3. From the Requests/Actions list, select a request.
4. From the Target type drop-down list, select the destination of the procedure. For example, you can have the procedure run in a new window or in a report frame on the HTML page.
5. Select which frame the procedure will run in. This option is only needed if you want to run the procedure, in a frame, on the HTML page, and not a new window.

The following image is an example of the Tasks section for a procedure that will automatically execute when the page is loaded.

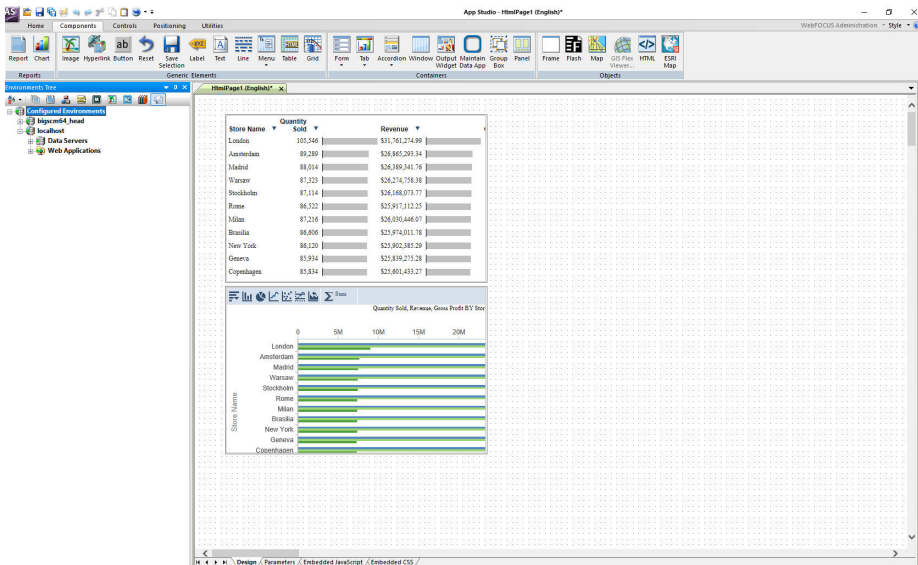


Generating a Chart From a Report

In Developer Studio, when you generate a graph from a report, that graph is created on the right-side of the report, as shown in the following image.



In App Studio, when you generate a chart from a report, that chart is created below the report, as shown in the following image.



Making an HTML Page Responsive

Responsive web design is an approach to webpage creation where content automatically realigns to fit a variety of different window sizes and screen sizes. This allows the content from one HTML file to automatically adapt to multiple footprints such as a desktop, laptop, tablet, or smartphone.

There are two ways to create responsive pages in App Studio:

- ❑ **Using a Responsive Template.** This method lets you create a page by populating responsive Output Widget containers in a pre-defined template or a layout that you create at design time. You can use the HTML/Document wizard to specify a layout that is either row-based or column-based. For more information, see [How to Select a Responsive Template](#) on page 71.
- ❑ **Using the HTML Canvas Responsive Mode.** This “free form” method lets you create a responsive page with no pre-defined layout. This manual approach is perhaps not as fast as using a template, but may be preferable if you want to create a responsive page with containers of many different types. For more information, see [How to Create New Content in Responsive Mode](#) on page 73.

This method can also be used to convert an existing HTML file to responsive. For more information, see [How to Convert an HTML File to Responsive](#) on page 74.

Procedure: How to Select a Responsive Template

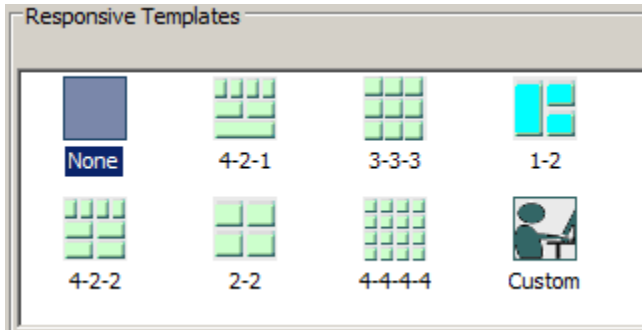
Select a responsive template from the HTML/Document Wizard as follows:

1. Open the HTML/Document Wizard using one of the following methods:
 - a. On the Home tab, in the Content group, click *HTML/Document*; or
 - b. In the Environments Tree panel, right-click an application folder, click *New*, and click *HTML/Document*.

The HTML/Document Wizard opens. If you used option **a.** above, the HTML/Document Wizard contains a navigation window that you can use to specify the folder where you want to save the new file.

2. Click *Next*.

The Templates, Settings, and Themes page opens, containing the Responsive Templates section, as shown in the following image.



3. Click on a desired responsive template.

- Selecting *None* results in a generic HTML canvas that is not responsive.
- Selecting *Custom* lets you design a layout that differs from those offered by the responsive templates.

For purposes of this procedure, do not select *None* or *Custom*.

4. Use the Input Controls section at the bottom of the page as follows:

- Select *Auto-hidden* to hide the controls input panel. This is the default. At run time, users display the panel by clicking the down arrow in the title bar.
- Select *Always show* to display the controls input panel. At run time, users hide the panel by clicking the up arrow in the title bar.
- Clear the check box for *Use a single input control panel* if you want to use widget-level controls instead of, or in addition to, page-level.

5. Select *Run requests on load* to load the contents of all the widgets when the page first opens.
6. Select *Add page header* if you want to add space above the title bar, within the responsive panel1 <DIV>. You could use this area to add a graphic and/or company logo, for example.
7. Click *Finish* to close the HTML/Document Wizard.

The HTML canvas opens with the automatically generated responsive layout corresponding to the selected template.

Note: It is a best practice to expand to full screen when working with a responsive layout. This lets you access the greatest number of widgets without having to scroll.

Procedure: How to Create New Content in Responsive Mode

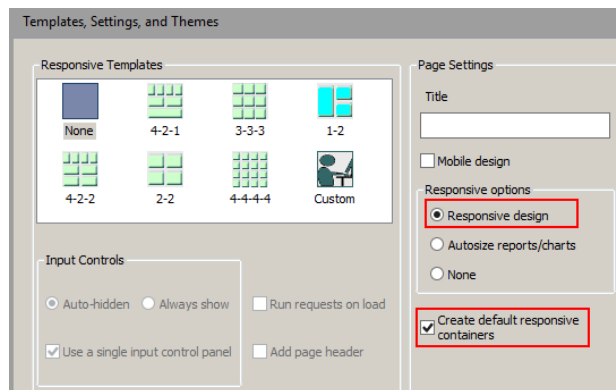
You can create new responsive content using the HTML canvas in responsive mode, as defined in this procedure.

1. Open the HTML/Document Wizard using one of the following methods:
 - a. On the *Home* tab, in the *Content* group, click *HTML/Document*; or
 - b. In the Environments Tree panel, right-click an application folder, click *New*, and click *HTML/Document*.

The HTML/Document Wizard opens.

2. Click *Next*.

The Templates, Settings, and Themes page opens, containing the Page Settings section, as shown in the following image.



3. In the Page Settings section, select *Responsive design*.

Note:

- ❑ The Create default responsive containers option is selected, by default. When you create a control using the New Parameters dialog box, the forms and the controls will be created with responsive containers.
 - ❑ If you clear the Create default responsive containers check box, no responsive containers will be drawn for the form object.
 - ❑ You can make a non-responsive container responsive by selecting the container and, in the Properties panel, selecting Yes for the Enable Responsive property.
4. Click *Finish*.

The HTML canvas opens. In the Properties panel, in the DOCUMENT properties, the Responsive design value is set to Yes.

As an alternative to using the HTML/Document Wizard, you can open a file in the HTML canvas and manually change the Responsive design property value to Yes.

You can add content to the HTML canvas before turning Responsive Design on. However, there are advantages to turning Responsive Design on before adding content. Design time benefits include:

- Having the responsive minimum and maximum widths established
- Having the maximum width visually indicated by the horizontal line on the right side of the canvas
- Having the Preview Runtime capability to preview the run-time appearance

Procedure: How to Convert an HTML File to Responsive

1. To enable responsive behavior in an existing HTML file, double-click the file in the Environments Tree panel, or right-click it and click *Open*.

The HTML file is open in the HTML canvas.

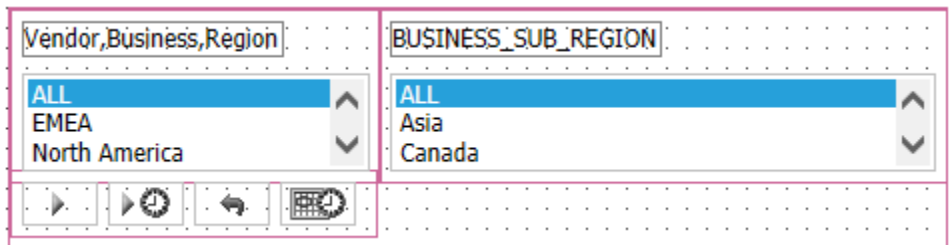
2. In the Properties panel, select *DOCUMENT* from the drop-down menu.

The Document properties are displayed.

3. In the Responsive design property, select Yes from the drop-down menu.

A message appears asking of you want to group control elements into responsive containers.

4. Indicate your preference for responsive containers as follows:
 - a. Click Yes to add multiple containers in the control area as shown in the following image.



Note there are three containers within the fourth, larger container. The containers are red, indicating that they are responsive.

- b. Click *No* to add no containers. This may be preferable if your page contains many nested objects, otherwise each object will get its own container.

You can add additional containers at a later time:

1. Multi-click the control objects you want to group in a container.
2. Right-click one of the highlighted objects and select *Add selected to new container* from the shortcut menu.
3. To merge multiple containers, multi-select the containers, select *HTML object manipulation* from the shortcut menu, and then select *Merge containers*.
4. To make a frame responsive, select it and open the Properties panel. Set the *Enable responsive* property value to *Yes*.
5. To move contents from one container to another, right-click the HTML canvas and select *Toggle Drag and Drop* from the shortcut menu. The cursor changes to a crosshair to indicate that objects can be moved. Select an object and drag it to any container you want. Click *Preview Runtime* on the Utilities tab to preview your changes.
6. Click the *Run* button on the Application menu and view the HTML output full screen. Gradually decrease the width to observe the responsive folding.

Chapter 6

Creating, Editing, and Managing Synonyms and Adapters

This topic describes the differences between Developer Studio and App Studio when creating, editing, and managing synonyms and adapters. In Developer Studio, you use the toolbars in the Synonym Editor to create and edit a synonym. In App Studio, you use the tabs on the ribbon that are available when the Metadata canvas is active. One of the key differences between the Synonym Editor in Developer Studio and the Metadata canvas in App Studio is how they are invoked.

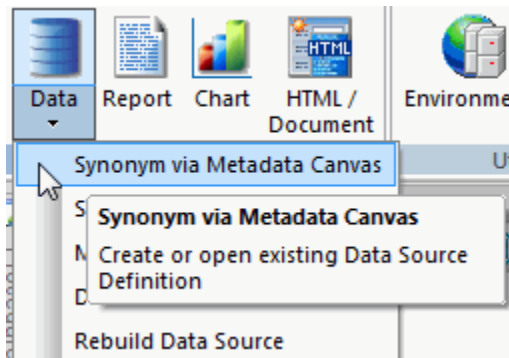
In this chapter:

- ❑ [Creating and Editing Synonyms](#)
- ❑ [Configuring Adapters](#)
- ❑ [Rebuilding a Data Source](#)
- ❑ [Automatically Adding Synonyms to Projects](#)

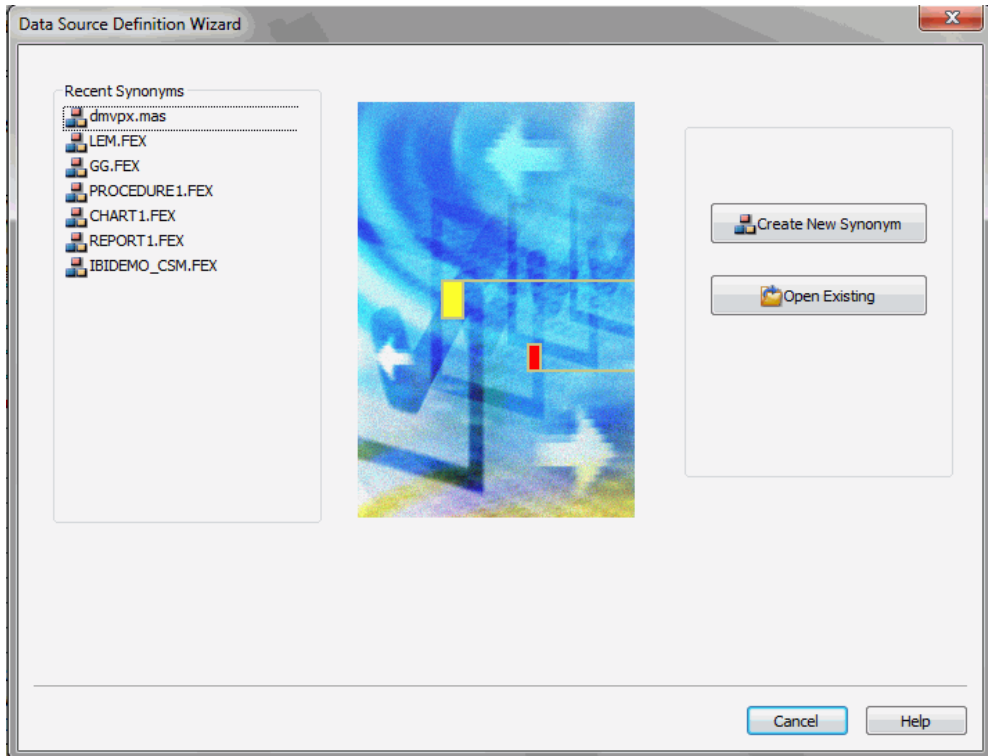
Creating and Editing Synonyms

In Developer Studio, you can create new synonyms using the Synonym Editor. To access the Synonym Editor, right-click a Master File folder, point to *New*, and then click *Synonym via Metadata Canvas*.

In App Studio, you can create new synonyms and edit existing synonyms using the Metadata canvas. To access the Metadata canvas, go to the *Home* tab, and in the *Content* group, click *Data*, and then click *Synonym via Metadata Canvas*, as shown in the following image.



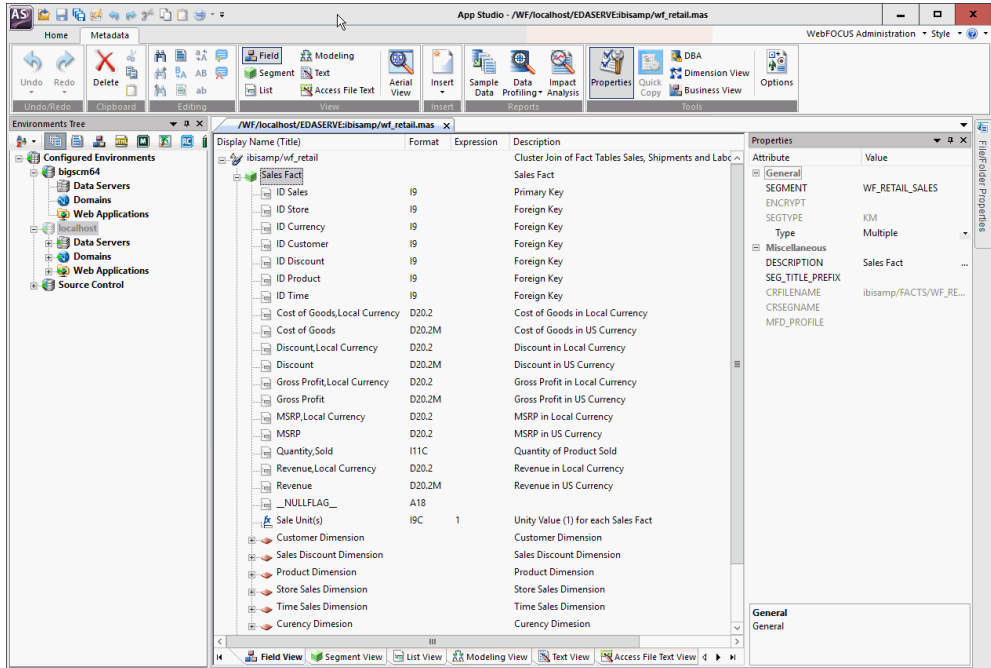
The Data Source Definition Wizard opens, as shown in the following image.



If you click *Create New Synonym*, the Select Application pane of the Data Source Definition Wizard is displayed. Select a location to create your synonym and click *Finish* to open the Metadata canvas.

If you click *Open Existing*, the Selecting Data Source pane of the Data Source Definition Wizard is displayed. Select a data source to edit and click *Finish* to open the Metadata canvas.

The Metadata canvas and Metadata tab open, as shown in the following image.

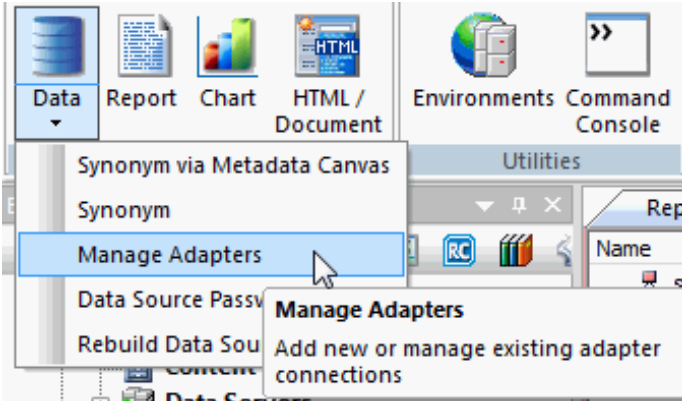


If you clicked New Synonym, you can begin creating a new data source once you are in the Metadata canvas. If you clicked Open Existing, you can edit your existing data source in the Metadata canvas.

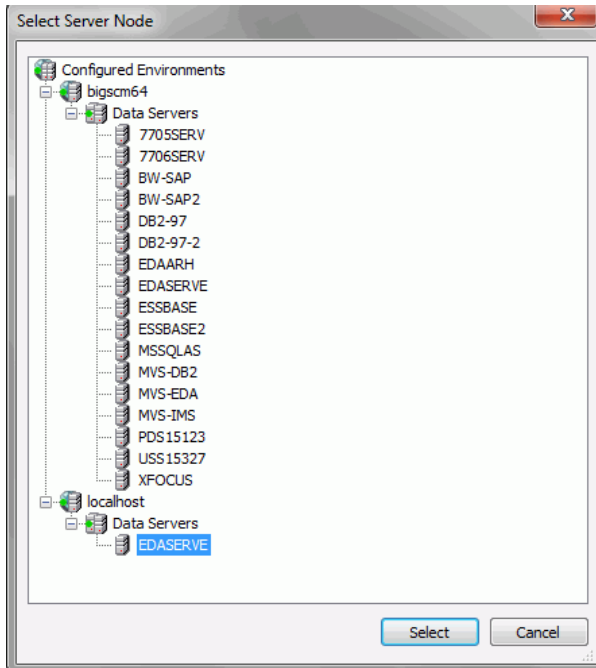
Configuring Adapters

In Developer Studio, you use the Create Synonym tool to configure the adapters that you use to access synonyms. To configure which adapters are available for use, right-click a Master File folder and then click *Synonym*.

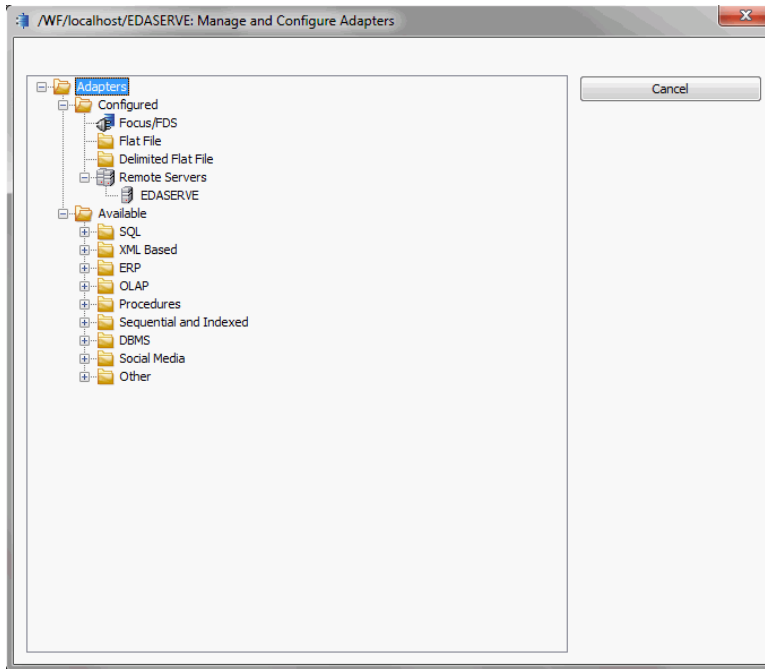
In App Studio, you can use the Manage Adapters command to configure the adapters that you use to access synonyms. To configure your available adapters, go to the *Home* tab, and in the *Content* group, click *Data*, and then click *Manage Adapters*, as shown in the following image.



The Select Server Node dialog box opens. Select an application folder, and click *Select*, as shown in the following image.



The Manage and Configure Adapters dialog box opens, as shown in the following image.



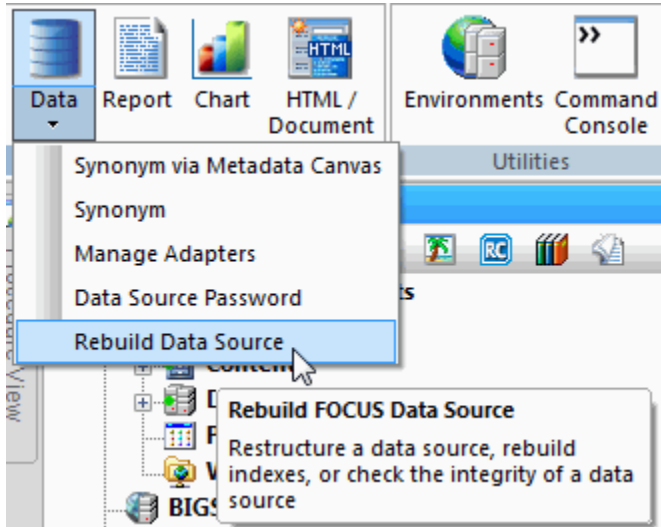
You can edit the settings of your currently configured adapters, listed under the Configured folder, or configure a new adapter by selecting one from the Available folder.

Rebuilding a Data Source

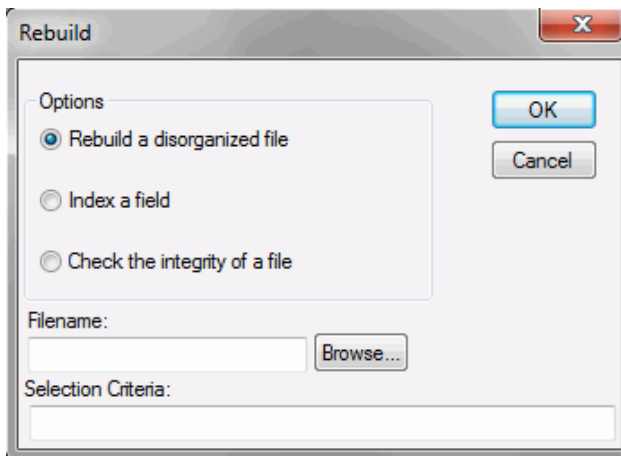
In Developer Studio, you click Utilities on the Commands toolbar and then select Rebuild to access the Rebuild dialog box.

Rebuilding a Data Source

In App Studio, you can use the Rebuild Data Source command to access the Rebuild dialog box, where you can rebuild data sources, index a field, or check the integrity of a file. To access the Rebuild Data Source command, in the Content group, click *Data*, and then click *Rebuild Data Source*, as shown in the following image.



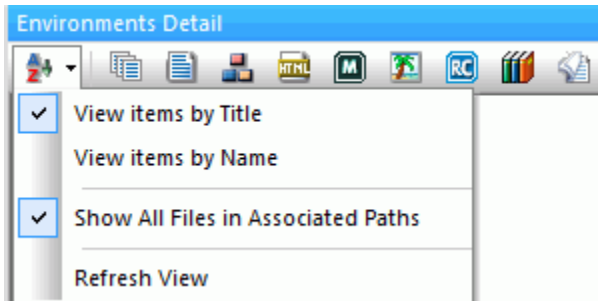
The Rebuild dialog box opens, as shown in the following image.



Automatically Adding Synonyms to Projects

In Developer Studio, when you create a synonym in the Synonym Editor, it is automatically added to the current project and is visible in the Explorer window.

In App Studio, for a synonym that you create in the Metadata canvas to show in the Configured Environments tree of the Environments Detail panel, you must select the Show All Files in Associated Paths option in the View Options drop-down menu. The View Options drop-down menu and the Show Files in Associated Paths option are shown in the following image.



Note: If you add a synonym to a project, that synonym will show whether Show All Files in Associated Paths is selected or not.

Developer Studio Functionality Deprecated in App Studio

This appendix details Developer Studio functionality that has been deprecated in App Studio.

In this appendix:

- ❑ [Deprecated Functionality List](#)

Deprecated Functionality List

The following Developer Studio functionality has been deprecated in App Studio due to improvements in design, workflow, and usability.

Item	Reason/Notes
Creating Visual Discovery ActiveX Dashboards using HTML Composer	<p>Reason: The new Visual Discovery Advanced Edition (AE) Workbench is now used to create Visual Discovery dashboards. Visual Discovery ActiveX does not contain as much functionality as the Visual Discovery AE product. When using Visual Discovery ActiveX, you can only use Internet Explorer to view these types of pages.</p> <p>Notes: The Visual Discovery license for App Studio only provides the options that you need to open and edit an existing dashboard in the HTML canvas. In App Studio, you can edit existing Visual Discovery controls in the Settings panel. You cannot add new Visual Discovery controls to a dashboard in App Studio.</p>
Maintain Development Environment	<p>Reason: Maintain development is being integrated into App Studio as part of the HTML canvas, in a future release.</p> <p>Notes: App Studio and Developer Studio can be run simultaneously. To create Maintain files, continue to use Developer Studio. To create files that are not Maintain files, use App Studio.</p>

Item	Reason/Notes
Text editor tab in Document Composer	<p>Reason: Text editing should be done outside of the user interface. This is because any edits made in the Text Editor that are not supported by the user interface will produce unexpected results or errors.</p>
Logical View	<p>Reason: This feature is not widely used in Developer Studio.</p>
RIA (Bindows) Themes for new HTML pages	<p>Reason: This feature is not widely used in Developer Studio.</p> <p>Notes: You can open and edit existing RIA pages in App Studio and they will be converted to use standard themes and jQuery animations. The option to create new RIA pages has been removed. Instead, you can now use jQuery Animations to animate objects on your HTML page.</p>
Viewing FOCUS Files in App Studio	<p>Reason: FOCUS files are not able to be edited, so they do not display when filtering the Environments Tree panel to show Master Files.</p>
Initial Presentation for Active Reports	<p>Reason: This was initially implemented because active reports did not work with the GRAPH command. Now that active reports work with the GRAPH command, and much more chart functionality is available, Initial Presentation is no longer needed.</p>
Project Deployment	<p>Reason: This feature is not widely used in Developer Studio.</p> <p>Notes: Instead of using Project Deployment, we recommend creating separate WebFOCUS client configurations for Development, Test, and Production.</p>

Item	Reason/Notes
Workspaces	<p>Reason: This feature is not widely used in Developer Studio.</p> <p>Notes: Developing on a remote environment in Data Server application folders and/or Managed Reporting domain folders provides a similar organizational scheme for application content.</p>
Select All option in Report Painter	<p>Reason: The Select All option has been removed with the paradigm change from toolbars to the ribbon.</p> <p>Notes: The functionality to select all fields in the Object Inspector still exists. In order to select all fields in the Object Inspector, you would select the first field and then Shift + click the last field.</p>
HTML Layout Templates in HTML Composer	<p>Reason: This feature is not widely used in Developer Studio.</p> <p>Notes: Existing pages that use templates can be run and modified in App Studio.</p>
Refresh of the Design view tab by switching to Embedded JavaScript/CSS view tab.	<p>Reason: You can now refresh all objects on the canvas by using the <i>Refresh All</i> command, in the <i>Miscellaneous</i> group, on <i>Utilities</i> tab, while in the HTML canvas.</p>

Legal and Third-Party Notices

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, FOCUS, iWay, Omni-Gen, Omni-HealthData, and WebFOCUS are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2021. TIBCO Software Inc. All Rights Reserved.

Index

A

active reports

 Initial Presentation [86](#)

ActiveX [85](#)

adapter [79](#), [81](#)

administration [6](#)

App Studio interface

 application button [14](#)

 canvas [14](#)

 Configured Environments panel [14](#)

 Quick Access Toolbar [14](#)

 ribbon [14](#)

App Studio

 interface components [14](#)

Application menu [15](#)

B

Bindows [86](#)

C

Calculation group [44](#)

canvas [18](#)

 Chart [18](#)

 Document [18](#)

 HTML [18](#), [57](#)

 Metadata [18](#)

 Procedure [18](#)

 Report [18](#)

Chart canvas [6](#)

Commands toolbar [6](#), [15](#)

D

data source [77](#)

Data tab [44](#)

Developer Studio Explorer [7](#), [22](#)

Document canvas [7](#)

Document Composer [7](#), [18](#)

 Text Editor [86](#)

E

environments [22](#)

 adding [22](#)

 editing [22](#)

 managing [22](#)

F

field [52](#)

Field Properties dialog box [49](#)

 Drill Down tab [51](#)

 Style tab [50](#)

Field tab [49–51](#)

files [9](#)

 editing [9](#)

 filtering [25](#)

 recommendations [9](#)

 sorting [25](#)

files [9](#)
 viewing [25](#)
Format tab [40](#)

G

Graph tool [6, 18](#)
group [6](#)

H

Home tab [6](#)
HTML canvas [6](#)
HTML canvas in responsive mode [71](#)
HTML Composer [6, 18, 57](#)
 Templates [87](#)
hyperlink [63](#)

I

Images tab [47, 49](#)
installation [8](#)
interface [11](#)
 App Studio [11](#)
 Developer Studio [11](#)

L

Links group [43, 51](#)
Logical View [86](#)

M

main interface [11](#)

Main toolbar [6, 15](#)
Maintain [85](#)
Metadata canvas [7, 77](#)

O

onload [67](#)
Output types group [40](#)

P

panel [7](#)
 Environments Tree [7, 22](#)
 Procedure View [7](#)
 Properties [7](#)
 File/Folder Properties [7](#)
 Properties [58](#)
 Settings [7, 60](#)
parameter [63](#)
procedure [20, 33, 63](#)
 embedded [63](#)
Project Deployment [86](#)
Properties and Settings dialog box [7, 60](#)
Properties panel [7](#)
Properties window [7, 58](#)

R

report [37, 63](#)
Report canvas [6, 37](#)
Report Options dialog box [39, 43](#)
 Computes tab [44](#)
 Drill Down tab [43](#)

Report Options dialog box [39](#), [43](#)

Images tab [47](#), [49](#)

Output tab [40](#)

Style tab [42](#)

Report Painter [6](#), [18](#), [37](#)

Select All [87](#)

Report tab [42](#), [43](#)

request [63](#)

Responsive

convert to responsive [74](#)

HTML canvas [71](#)

methods [71](#)

new content [73](#)

selecting a template [71](#)

template [71](#)

web design [71](#)

RIA [86](#)

ribbon [6](#)

S

Settings panel [7](#)

smart context [52](#)

SQL report [38](#)

Style group [42](#), [50](#)

synonym [77](#)

adapter [77](#)

creating [77](#)

editing [77](#)

T

tab [6](#)

Active Report Options [6](#)

Components [6](#)

Controls [6](#)

Data [6](#)

Document Composer [6](#)

Field [6](#)

Format [6](#)

Home [15](#)

Insert [6](#)

Layout [6](#)

Metadata [6](#)

Modeling [6](#)

Positioning [6](#)

Report [6](#)

Styling [6](#)

Utilities [6](#)

task [67](#)

autoexecute [67](#)

load [67](#)

Tasks and Animations panel [66](#)

Requests section [63](#)

Tasks section [67](#)

terminology [6](#)

tool [18](#)

toolbar [6](#)

Report Painter [6](#)

V

Visual Discovery [85](#)

W

Workspaces [87](#)