IBM Db2 Web Query for i Developer Workbench

Release 2.4.0

Contents

Chapter 1

Introducing Db2 Web Query Developer Workbench

Db2 Web Query is a complete, web-ready data access and reporting system that connects users to data. It accesses and processes information located in any format, on any platform, and presents that information to end users through a web browser.

Business intelligence (BI) needs continue to evolve, as organizations seek to make more information from more sources available to more people inside and outside the enterprise. Developers need powerful, yet flexible tools in order to satisfy the increasingly sophisticated BI content demands posed by a growing number of employees, customers, suppliers, partners, and other stakeholders.

Db2 Web Query Developer Workbench is the Windows-based graphical user interface (GUI) environment for creating advanced reporting applications. Db2 Web Query Developer Workbench is the most robust BI application development platform on the market today.

In this chapter:

What Is Db2 Web Query Developer Workbench?

Building on the industry-familiar Microsoft Office Ribbon Interface, Developer Workbench offers a simplified user experience and workflow, empowering 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 a BI application.

Db2 Web Query Developer Workbench provides:

- **Rapid deployment.** Installation and setup are fast and simple.
- **Ease of use.** A familiar ribbon interface, which can be customized to individual preferences, allows users to get up to speed quickly.
- **Maximum flexibility.** Build and deploy ad hoc or instant queries that go beyond the prescriptions of any particular application. Developers can also add dimensions to data structures at any time without affecting other queries, since the underlying data source is not changed.

- **Support for complex requests.** Unlike other BI development environments, which require the use of complex code to handle anything more than the most simple of requests, Developer Workbench allows users to create any request, no matter how sophisticated, using the same intuitive interface.

Developer Workbench eliminates the complex multi-tool paradigm that exists with most other development solutions, providing a single, fully-integrated environment for rapidly designing and creating reports, dashboards, InfoApps, and other types of BI apps and content, including:

- Income statements, balance sheets, and other types of tabular or financial reports.
- Charts and visualizations. A library of more than 120 HTML5 chart types meet the needs of every information consumer.
- Dashboards that draw data from one source, or multiple disparate systems. Dashboards can be viewed online or offline in disconnected mode, with full interactivity and analytical capabilities.
- InfoApps, business intelligence applications that are form-driven and extremely simple to use, providing the most effective way to make information accessible to all stakeholders, as well as for custom and SaaS implementations.
- Mobile BI applications that run on any device and on any platform.
- Responsive designs that conform to the device form factor, maximizing the user experience.
- Metadata creation and modeling, such as multifact star schemas, joins, and business views.

Db2 Web Query Developer Workbench Capabilities

In Developer Workbench, you can do the following:

Access data and descriptions. Using the Data Management Console, you can create new synonyms, and view or modify existing synonyms, in a graphical user interface. Synonyms enable you to access and interpret data sources for use in reporting applications. Capabilities include metadata design and development, with full visual modeling of schemas.

Create reporting applications. Build reporting procedures in the Report Canvas, Chart Canvas, or HTML Canvas, which can include the following components.

- **Reports.** Display your data in a tabular format. You can create stand-alone reporting procedures, or add the report to a document or an HTML page.

- **Charts.** Communicate your data in a more illustrative format. You can translate complex data into an easy-to-read and understandable visual format. This often provides a new perspective to the information that users view. You can create stand-alone charting procedures or add the chart to a document or an HTML page.
- **HTML pages.** Build webpages for users to launch, view, and analyze data. You can add multiple reports and charting procedures to an HTML page, in addition to the controls that enable you to manipulate the data. You can also apply styling through the use of cascading style sheets (CSS), JavaScript files, and jQuery animations.
- **Visualizations.** Create charts, maps, and grids to visually represent your data. You can add multiple visuals to the canvas to create a complete visualization, and you can update, change, or revise the visualization at any time to account for shifts in data needs.

Customize reporting applications. In addition to customizing your reporting applications by applying styling and color, you can add the following components to a procedure.

- **Virtual fields (DEFINE or COMPUTE statements).** Create temporary fields that are treated as real fields stored in the data source.
- **Define functions.** Create user functions that you can use in components of a procedure. You can retrieve your stored functions from the Functions Arguments dialog box.
- **Joins.** Define relationships between two or more data sources so that a report can use the data from all sources at once. You can also merge data sources by creating logical expressions on the Match canvas.

Db2 Web Query Architecture

This topic briefly explains the main Db2 Web Query components and implementations. For a complete description of Db2 Web Query, see the Db2 Web Query documentation.

Db2 Web Query and Your Network

Db2 Web Query integrates into your existing network by connecting your web server to your data. End users access Db2 Web Query applications through a web browser, so they need only the following elements:

- **Web browser.** To access Db2 Web Query applications, users need a browser and a TCP/IP connection to a web server.
- **Web server.** WebSphere Liberty handles requests by returning files to a browser or by executing processes that provide additional functionality. You can provide Db2 Web Query functionality by connecting to the web server using Java servlet calls.

- **Data.** Db2 Web Query can access data from almost anywhere. Once you have configured data access and described that data, you can report on it.

Db2 Web Query Components

There are two main Db2 Web Query components.

- **Db2 Web Query Client.** The Db2 Web Query Client resides on the web server and connects Db2 Web Query to the web through Java servlets. When a user makes a request from Developer Workbench or a browser, the Db2 Web Query Client receives and processes the request by passing it to the Db2 Web Query Reporting Server.
- **Db2 Web Query Reporting Server.** The Reporting Server resides on machines that can access your data. The Reporting Server provides data access, number crunching, and report generation functionality.

Db2 Web Query Developer Workbench Architecture

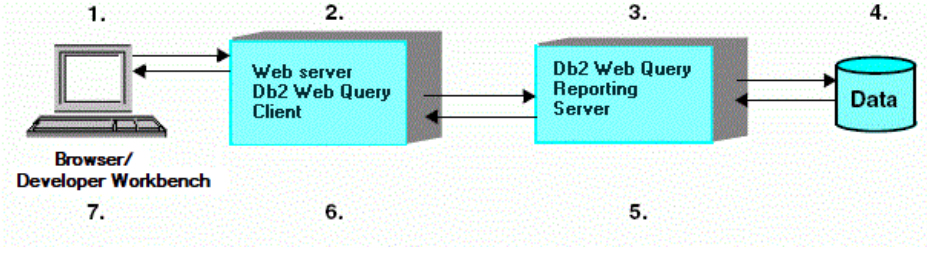
This topic briefly explains the main Developer Workbench component and implementation.

- **Developer Workbench Graphical User Interface (GUI).** Graphical development and code generation features for application development. From Developer Workbench, it is necessary to connect to a Db2 Web Query environment before you begin to manage metadata or develop applications.

  Note: Developer Workbench is a 64-bit application. Developer Workbench does not support a 32-bit operating system.

Db2 Web Query and Db2 Web Query Developer Workbench Processing

The following figure illustrates how Db2 Web Query and Developer Workbench process requests. Each step is explained below the figure.



1. A user makes a request and passes parameters by calling a Db2 Web Query Servlet through links and forms on a webpage, or through Developer Workbench.

2. The request and parameters come to the Db2 Web Query Client on the web or application server, which processes the parameters and creates a request for the Reporting Server.
3. The Reporting Server receives the request, processes it, and accesses any necessary data.
4. Data is retrieved from data sources to process the request.
5. The Reporting Server processes the user request using the retrieved data.
6. The response is returned to the Db2 Web Query Client on the web or application server.
7. The response is returned to the user.

Launching Db2 Web Query Developer Workbench

Once you have installed Developer Workbench, you can launch the product and begin developing applications.

- From the Start menu, expand the *Db2 Web Query for i* program group, expand the *Db2 Web Query for i Developer Workbench 90* folder, and then double-click *Db2 Web Query for i Workbench*.

Note: If you have previously configured Db2 Web Query environments on your machine, those environments appear when you launch Developer Workbench.

Chapter 2

Navigating in Db2 Web Query Developer Workbench

Developer Workbench provides an easy-to-use interface for creating applications, such as reports, charts, and HTML pages.

The first time you launch Developer Workbench, a default interface opens. You can modify the default interface by customizing the appearance of the application window and rearranging the placement of the main interface components.

These changes are automatically saved and made available the next time you launch the product.

In this chapter:

Opening the Application Menu of File-Related Commands

The Application button, represented by the Developer Workbench icon 

, opens the Application menu of file-related commands. You can save an active document and print an active document. The Options button opens the Developer Workbench Options dialog box, where you can set user preferences. The Application menu also lists recently opened documents. The Application button is always available. It is located in the upper-left corner of the Developer Workbench interface.

Tip: Double-click the Application button to exit Developer Workbench. If you exit Developer Workbench in this way, active (open) applications will not be saved.

From the Application menu, you can perform the following actions:

- **Save.** Save the active document.
- **Save As.** Save the active document with a new name.
- **Save All.** Save all open documents with the current file names. There will be no prompting for file names.
- **Run.** Run the report, chart, or HTML page. When you click the *Run* menu, the Message Viewer Options submenu opens.

  From the Message Viewer Options submenu, you can select from among four Message Viewer Options that affect the Developer Workbench Viewer display when a report, chart, or HTML page is run. The options are:

  - When you select *Message Viewer OFF* and run a report, the Developer Workbench Viewer window displays the report without any messages.
  - When you select *Message Viewer ON* and run a report, the Developer Workbench Viewer window displays the report and a message.
  - When you select *Display command lines,* the Developer Workbench Viewer window displays the report and the command lines.
  - When you select *Display Dialogue Manager commands,* the Developer Workbench Viewer window displays the report and the Dialogue Manager commands, and the result of their evaluation.
- **Print.** Print the active document. When you click the *Print* menu, the Preview and print the document submenu opens.

  From the Preview and print the document submenu, you can print the active document using the current print options, preview the active document before printing, and configure your print options.

  Note: The Print and Quick Print options are only available when you edit content in the Text Editor, or when in a tab containing plain text.
- **Close.** Close the active document.
- **Close All.** Close all open documents. If changes were made to a file, you will be prompted to save the changes.

Note:

- In the Application menu, the terms *document* and *active document* represent all files which you can create in Developer Workbench.
- Spaces are not allowed in file names. When naming a report, chart, or HTML page in their respective wizards or in the Save As dialog box, spaces are replaced with underscores (_). When naming a procedure, spaces and underscores are not permitted, and when you press the Space bar, nothing happens. If the domain is set to display by title, the wizards, Open dialog box, and Save As dialog box will still expect a proper name, with no spaces, to be entered.

Reference: Save As Dialog Box

The Save As dialog box opens when you click *Save As* from the Application menu. You can use this dialog box to save a file to another location or to provide a different file name.

Configured Environments list

Contains a list of currently Configured Environments. You can navigate to different environments and development areas to display different files in the file list.

File list

Displays all files of a specific type in a selected folder.

File name text box

Use this text box to type the name of a file you want to open.

Note: If you are in the Workspaces area and the Configured Environments tree is set to display by title, the Save As dialog box will only require a name to be entered.

File Type drop-down list

Displays the filter based on applicable file type.

Details Toggle

Toggles between displaying the files in the file list with details (date modified, size, and so on) or in a list. Displaying files with details is the default.

Note: If you choose to display details or not, Developer Workbench remembers your choice and will keep the setting you chose the next time the Open File dialog box is open. For example, if you choose to display file details the first time you invoke the Open File dialog box, then the next time you invoke the Open File dialog box, file details are shown.

Refresh Button

When clicked, refreshes the list of available files.

Setting User Preferences Using the Options Dialog Box

You can customize Developer Workbench by setting specific user preferences, using the following tabs in the Developer Workbench Options dialog box. To open the Options dialog box, click *Options* in the Application menu.

Reference: General Tab

The General tab contains options for starting Developer Workbench, minimizing the main window, and other settings. The following options are available.

Main Window Options

Option	Description
Maximize main window	Maximizes the application window when you begin each session.

Other Settings

Option	Description
Show Welcome Screen	Controls the appearance of the Welcome screen when Developer Workbench is launched.
Confirm close	Displays a prompt for users to confirm that they want to exit Developer Workbench.
Recent file list limit	Allows you to set the number of recently used files that appear in the Recent file lists. The default value is 25.
Ping Interval (minutes)	Sets the frequency at which requests are sent to the web or application server so your connection does not time out. The default is 5 minutes.

Option	Description
Save Document Recovery information every 5 minutes	Sets the frequency at which a copy of the work in progress will be automatically saved. The default is 5 minutes.
Reset All Message Boxes	Resets all message boxes to their default state. For example, if you selected the *Don't show this message again* check box in any message box, clicking this button will reset all message boxes. Once you click this button, it becomes inactive until you choose not to display a message box in the product.
Reset All Options to Default	Restores all options in Developer Workbench to the default settings.
Default file editor	Lists file editor options, such as: ❏ Edit in Developer Workbench canvas ❏ Edit in Text Editor ❏ Edit in Windows registered tool **Note:** The Edit in Windows registered tool option is only visible if the file type you are accessing has been associated with a Windows application. The file editor used determines which shortcut menu options are available and controls the default behavior for double-clicking and right-clicking files.

Option	Description
Language	Provides a drop-down list that allows you to change the language of the Developer Workbench screen text. After you change the language, restart Developer Workbench to make the change effective. The languages listed are selected in the Dynamic Language Switch setting in the Db2 Web Query Administration Console. The Sync with WebQuery check box is selected, by default. When this check box is selected, the language used for Developer Workbench and Db2 Web Query are synced with the language selected in the Language drop-down list. If this check box is not selected, Developer Workbench uses the language selected in the Language drop-down list and Db2 Web Query uses the language selected in the Environments dialog box. If you make changes, restart Developer Workbench to make the change effective.

Reference:

HTML Page Tab

The HTML Page tab enables you to set grid settings for an HTML page, preview reports and charts, set single or multiple layers, and alter the frequency in which you refresh thumbnails.

Note: If you make changes to the HTML Page settings in the Developer Workbench Options dialog box (Grid Settings, Preview Settings), you need to press F5 to refresh an HTML page that is currently open in order to see the changes.

The following options are available.

Grid Settings

Option	Description
Show Grid	Enables you to view a grid while you create an HTML page. The Show Grid check box is selected, by default. Clear this check box if you do not want to develop with a grid.
Snap to Grid	Enables you to snap to grid on demand. The Snap to Grid check box is selected, by default. Clear this check box if you do not want to enable the snap to grid option.
Width	Customizes the width of your grid. The default is 10 pixels.
Height	Customizes the height of your grid. The default is 10 pixels.

Form type

Option	Description
None	Specifies no form object in the HTML page.
Single layer	Specifies a single layer form in the HTML page. This is the default setting.
Multiple layer	Specifies a multiple layer form in the HTML page.
Form Settings	Opens the Form Settings dialog box. The options for Form Settings are described in *Form Settings Dialog Box*.

Tab Characters

Option	Description
Insert spaces	Indicates that, when the Tab key is pressed, the number of spaces specified in the Tab size option is inserted.

Option	Description
Keep tabs	Indicates that, when the Tab key is pressed, a tab character is inserted.
Tab size	Specifies the number of spaces inserted when you press Tab. The default setting is 4.

Check boxes

Option	Description
Show 'New Parameters' dialog	Sets the New Parameters dialog box to appear in the HTML page. This is the default setting.
Default caching option	Sets the Default caching option to run in the HTML page. By default, this option is not selected.
Auto Arrange Objects	Sets the objects in the HTML page to arrange automatically. This is the default setting.

Preview settings

Option	Description
Reports and Charts Preview	Enables you to preview reports and charts before saving and deploying the HTML page. This is the default setting.
Simulated Data	Enables you to view the HTML page based on simulated data from Db2 Web Query.
Live Data	Enables you to view reports and charts using live data in the HTML page. This is the default setting.
Record limit for reports	Sets the record limit for reports. Values range from -1 to 999. The default setting is 500.
Record limit for input controls	Sets the record limit for input controls. Values range from -1 to 999. The default setting is 10.

Option	Description
Use Prefix	Sets a prefix to run before each report or chart component on an HTML page.
Refresh thumbnails every seconds	Sets the time interval for refreshing thumbnails. Values range from 0 to 999 seconds. The default setting is 20.
Default Theme	Sets the default theme for an HTML page. The default setting is Default.
HTML	Sets the type of an HTML page. The default setting is Custom.

Reference: **Form Settings Dialog Box**

The following table describes the options available in the Form Settings dialog box.

Option	Description
To the left of the input	Sets the prompt orientation to appear to the left of the input controls.
Above the input	Sets the prompt orientation to appear above the input controls. This is the default setting.
Distance between prompt and input	Sets the distance between prompt and input. The range is from 0 to 99 pixels. The default setting is 5 pixels.
Horizontal distance between controls	Sets the horizontal distance between controls. The range is from 0 to 99 pixels. The default setting is 10 pixels.
Vertical distance between controls	Sets the vertical distance between controls. The range is from 0 to 99 pixels. The default setting is 10 pixels.
Number of columns	Sets the number of columns in the form object. The default setting is 4.

Option	Description
Add schedule button	Adds a schedule button to the form object. This is the default setting. This button can also be inserted from the New Parameters dialog box.
Add defer button	Enables deferred running of a report. This is the default setting. This button can also be inserted from the New Parameters dialog box.
Start each chain on a new line	Starts chains on a new line. This is the default setting.

Reference:

Document Tab

The Document tab enables you to set grid settings for a document, preview reports and charts, and alter the frequency in which you refresh thumbnails. The following options are available.

Grid Settings

Option	Description
Show Grid	Enables you to view a grid while you create a document. The Show Grid check box is selected, by default. Clear this check box if you do not want to develop with a grid.
Snap to Grid	Enables you to snap to a grid on demand. The Snap to Grid check box is selected, by default. Clear this check box if you do not want to enable the snap to grid option.
Width	Customizes the width of your grid. The default setting is 10 pixels.
Height	Customizes the height of your grid. The default setting is 10 pixels.

Tab Characters

Option	Description
Insert spaces	Indicates that, when the Tab key is pressed, the number of spaces specified in the Tab size option is inserted.
Keep tabs	Indicates that, when the Tab key is pressed, a tab character is inserted.
Tab size	Specifies the number of spaces inserted when you press Tab. The default setting is 4.

Preview Settings

Option	Description
Reports and Charts Preview	Enables you to preview reports and charts before saving and deploying the document. This is the default setting.
Simulated Data	View the document based on simulated data from Db2 Web Query.
Live Data	View reports and graphs using live data in the document. This is the default setting.
Record limit for reports	Sets the record limit for reports. Values range from -1 to 999. The default setting is 500.
Use Prefix	Sets a prefix to run before each report or chart component on a document.
Refresh thumbnails every seconds	Sets the time interval for refreshing thumbnails. Values range from 0 to 999 seconds. The default setting is 20.

Reference: Environments Tab

Note: The Environments tab does not apply to Db2 Web Query. The Environments options are used to control the development areas that are visible in the Configured Environments tree. At least one option must be selected and applied to all configured environments.

The following options are available.

Environments settings

Option	Description
Show Data Servers area	Displays the Data Servers area in the Configured Environments tree.
Show domain area	Displays the domain area in the Configured Environments tree.
Show Web Applications area	Displays the Web Applications area in the Configured Environments tree.
Show master files for "Show all content files"	Displays Master Files when the *All Files* option is selected in the Environments Tree panel toolbar.

Reference: Help Configuration Tab

The Help Configuration tab enables you to configure the location of the web-based Help system for Developer Workbench. It provides the fields in which you can modify the Developer Workbench Help configuration information to access the online Help system from your environment. The following options are available.

Option	Description
Protocol	Specify the protocol, either HTTP or HTTPS.
Host	Type the name of the machine where the Help resides.
Port	Type the port number of the web and/or application server.
Help Context Root	Type the context root for the location where the Help is hosted.

Reference: **Output Viewer Settings Tab**

The Output Viewer Settings tab enables you to set a specific browser to use for output, and to set the navigation options for the output (run in new window, run in same window). The following options are available.

Browser Setup

Option	Description
Browser Setup	Select a browser to use for output. **Note:** ❏ The Browser Setup section is populated with a list of browsers that are installed on the Developer Workbench machine. ❏ Developer Workbench works with Mozilla Firefox®, Google Chrome™, and Microsoft Edge® browsers. Microsoft Edge is the default browser for Windows 10 operating systems. **Note:** As of April 1, 2021, Internet Explorer is no longer supported. ❏ If you are using Microsoft Edge as your external browser, ensure that the latest msedgedriver.exe version is installed and updated in your DevWorkbench /bin folder. ❏ The browser that is highlighted when you close the dialog box is the browser that will be used for execution.
Browser executable path	Sets the location of the browser executable. This field box is populated based on the browser selection.
Web driver location	Sets the location of the web driver for the browser. This field box is populated based on the browser selection.
Test URL	Indicates the test URL for testing the browser setup.

Option	Description
Test browser setup	Tests the browser setup, using the test URL. If the test is successful, the webpage will display in the browser and you will receive a success message. If the test is unsuccessful, you will receive a failed message.
Timeout	Sets the timeout interval (in seconds) for testing the browser setup.

Navigation Options

Option	Description
Run in new window/tab*	Sets the output to run in a new window or tab.
Run in same window	Sets the output to run in the same window.

Reference: File Extensions Tab

The File Extensions tab lists the currently supported file extensions. They are all selected, by default, and you can clear the check box for any file types that you do not want retrieved from the server.

Note: You must exit and restart Developer Workbench for your selections to take effect.

Option	Description
File Type	File extension for a specific type of file.
Display	If selected, displays this file type throughout Developer Workbench.

Accessing Frequently Used Commands

The Quick Access Toolbar provides access to frequently used commands, and the option to customize the toolbar with the commands that you use most often. By default, the Save, Save All, Quick Print, Undo, Redo, Cut, Copy, Paste, Run, and Customizing Quick Access Toolbar buttons appear on the Quick Access Toolbar, as shown in the following image.

Note:

- When using the undo and redo commands, consider that these actions may not take effect on options or information that is changed in every panel. In addition, if you change multiple options in one panel, for example, the undo or redo command will reverse all of those indicated changes in one action.
- The Cut, Copy, and Paste commands can be used on text, objects on the HTML canvas and Document canvas, and fields on the Report canvas.

The Customize Quick Access Toolbar button (down pointing arrow) appears on the Quick Access Toolbar and opens the Customize Quick Access Toolbar menu. This menu contains a list of the default commands on the toolbar. You can clear these commands to hide them from the toolbar. There are also options to add more commands with the Customize dialog box, move the Quick Access Toolbar below the ribbon, and hide the ribbon.

You can also add commands using the Customize dialog box. To access the Customize dialog box, select *More Commands* from the Customize Quick Access Toolbar menu. From the Customize dialog box, you can choose which commands you want to add to or remove from the Quick Access Toolbar, as well as the order in which the commands appear.

The Quick Access Toolbar is always available. By default, it is located in the upper-left corner of the Developer Workbench interface. To move the Quick Access Toolbar below the ribbon, click the arrow button, and then click *Show Below the Ribbon*.

Reference: Customize Dialog Box

Using the Customize dialog box, you can add or remove commands from the Quick Access Toolbar. By providing access to the Customize Keyboard dialog box, you can also customize the keyboard by setting keyboard shortcuts for commonly performed tasks.

Choose commands from

Opens a list of available areas in Developer Workbench. The Commands section dynamically repopulates with the commands that are associated with your selection.

Commands

A list of commands from a specific area of Developer Workbench. You can select commands and use the Add or Remove buttons to add or remove them from the Quick Access Toolbar. Commands are represented by the icon that displays on the ribbon. Commands that have no icon and a vertical arrow are command groups. Commands that have an icon and a horizontal arrow are commands with submenus. You can add all commands, command groups, and commands with submenus to the Quick Access Toolbar.

Add

Adds a selected command from the Commands list to the Quick Access Toolbar.

Remove

Removes a selected command from the Quick Access Toolbar. If no command is selected, the last command on the Quick Access Toolbar is removed.

Up

Moves a command up in order on the Quick Access Toolbar.

Down

Moves a command down in order on the Quick Access Toolbar.

Reset

Reverts the Quick Access Toolbar to use the default commands (Open, Save, Quick Print, and Run).

Show Quick Access Toolbar below the Ribbon

Displays the Quick Access Toolbar below the ribbon, when selected.

Customize

Opens the Customize Keyboard dialog box. For more information on the Customize Keyboard dialog box, see *Customize Keyboard Dialog Box* on page 31.

OK

Applies the changes you made to the Quick Access Toolbar.

Cancel

Exits the Customize dialog box without applying any changes you made to the Quick Access Toolbar.

Help

Opens the Developer Workbench Help.

Procedure: How to Add Commands to the Quick Access Toolbar

You can quickly add commands to the Quick Access Toolbar by right-clicking the ribbon, groups, or the individual commands, and selecting *Add to Quick Access Toolbar*, or you can follow the procedure below.

1. On the Quick Access Toolbar, click the arrow, and then click *More Commands*.

   The Customize dialog box opens.

2. In the Choose commands from list, select a command group.
3. Select a command name, and then click *Add*.

   The command name appears in the Quick Access Toolbar list.

   If you select a command group or a command with a submenu, all associated commands are accessible from the Quick Access Toolbar.

   To remove a command name from the Quick Access Toolbar list, select the command, and then click *Remove*.

4. Click the arrow buttons to rearrange the order of command names in the list.

   This also rearranges how the command buttons appear in the Quick Access Toolbar.

5. Click *OK* to save your changes.

Reference: Customize Keyboard Dialog Box

The Customize Keyboard dialog box is used to create and edit keyboard shortcuts to commands. You can access the Customize Keyboard dialog box by clicking the *Customize* button in the Customize dialog box from the Quick Access Toolbar.

Categories

A list of the available areas in Developer Workbench. The commands associated with that area are displayed in the Commands section.

Commands

A list of commands from a specific area of Developer Workbench. You can select commands and then assign a keyboard shortcut to them.

Current Keys

Displays the current keyboard shortcuts associated with the selected command.

Press new shortcut key

Press the keyboard shortcut you want to assign to a specific command. Once the Assign button is clicked, this keyboard shortcut will then be displayed in the Current Keys list for the selected command.

Set Accelerator for

Enables you to set an accelerator for one of the following: Default, HtmlPage, or Document.

Description

Provides a description of the selected command.

Assign

Allows you to assign a keyboard mapping to a command.

Remove

Allows you to remove an assigned keyboard mapping.

Reset All

Resets all keyboard mapping assignments for every command. A warning displays to ensure that want to perform this action.

Close

Exits the Customize Keyboard dialog box and return to the Customize dialog box.

Procedure: How to Create Keyboard Shortcuts

1. To open the Customize dialog box, do one of the following:
   - ❏ Click the drop-down arrow on the Quick Access Toolbar and then click *More Commands*.
   - ❏ Use the shortcut menu on the ribbon or Quick Access Toolbar and then click *Customize Quick Access Toolbar*.
2. Click *Customize*.

   The Customize Keyboard dialog box opens.
3. From the Categories list, select an area that contains the command for which you want to create a shortcut.
4. From the Commands list, select a command.
5. In the Press new shortcut key area, enter the key combination that you want to use with the command you selected.

   You can make a shortcut up to three keystrokes long (For example, Ctrl+Shift+R).

6. Click *Assign*.

   A shortcut is now assigned to the command you selected.

7. Click *Close* to exit the Customize Keyboard dialog box.

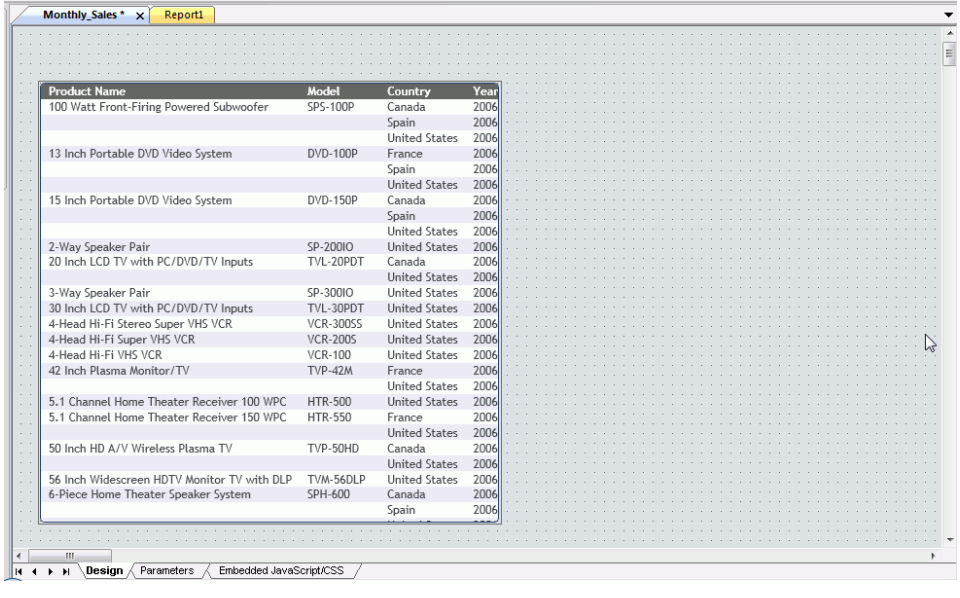
Developing Applications in the Canvas Area

You can use Developer Workbench features and functions to develop applications in the canvas area. The canvas is designed to maximize your design-time options. At run time, elements appear in the browser based on the order in which they were added to the canvas, left to right, and top to bottom. Your canvas may have the same appearance as the browser at run time, but it is not required. You can drag elements to areas of the canvas as needed to make it easier to work with them.

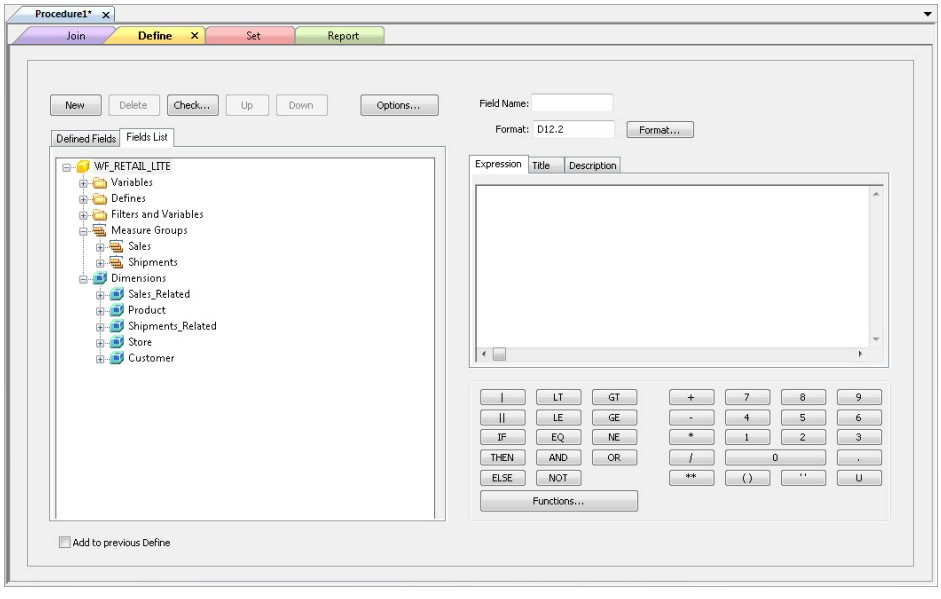
The size of your canvas depends on the placement and location of panels around the interface window.

As you develop in Developer Workbench, tabs open in the canvas area. Since you can develop an HTML page at the same time that you create a report, each canvas opens with a different colored tab. The colored tab at the top of each canvas displays the name of the file you are developing.

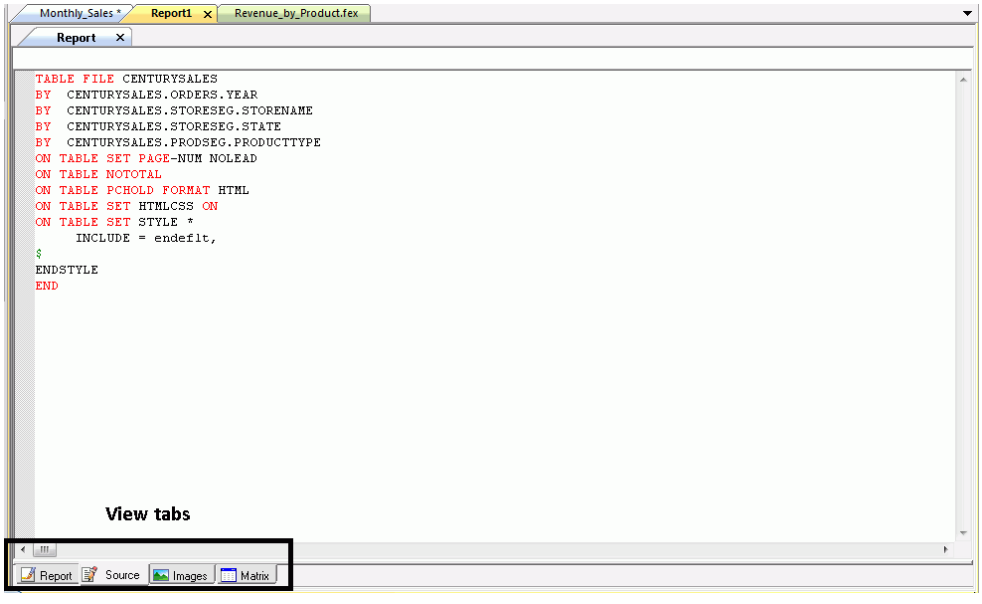
If you chose to name the file after you develop the content, a default file name, such as Report1, appears on the tab, as shown in the following image.



A second set of tabs opens when you create procedures with multiple components. For example, you can add Set, Define, and Join components to a procedure at any time. These canvases open as tabs below the report or chart procedure, as shown in the following image.



In addition to colored tabs, some canvases also contain View tabs. A View tab appears at the bottom of the canvas and changes the display of information on the canvas. View tabs, which display different views of the same component, are shown in the following image.



View tabs, such as those on the HTML canvas, also enable additional features and functions that you can use to customize an application. For example, in the HTML canvas, you can add JavaScript functionality to an HTML page through the Embedded JavaScript view tab.

You can develop Developer Workbench content in the following canvases.

Creating Charts Using the Chart Canvas

The Chart canvas enables you to create different types of charts. You can select from a variety of chart types and output formats, and add custom features to a chart. To access the Chart canvas, on the *Home* tab, in the *Content* group, click *Chart*. The Chart Wizard opens.

Creating Cascading Style Sheets Using the Cascading Style Sheet Canvas

The Cascading Style Sheet canvas allows you to create a cascading style sheet. To access the Cascading Style Sheet canvas, right-click any folder or file in the Environments Tree panel and in the shortcut menu point to *New*, point to *Text Editor*, and then click *Cascading Style Sheet*.

Working With the JavaScript and CSS Canvas

The Embedded JavaScript canvas and the Embedded CSS canvas have a built-in Auto Complete feature to assist in the creation of code segments, similar to functionality offered by Microsoft IntelliSense® and other code editors. When you begin typing a string, array, number, or user-defined object, a list of available JavaScript methods, related to those code segments, displays.

The Auto Complete feature is on by default. To turn it off, right-click in the Cascading Style Sheet canvas and clear *Auto Complete* from the shortcut menu.

Note: If there is a syntax error, the Auto Complete features will not display.

Editing Component Code Using the Text Editor Canvas

The Text Editor canvas allows you to edit component code. To access the Text Editor canvas, right-click a component in the Procedure View panel, and then click *Open in Text Editor*.

Creating HTML Pages Using the HTML Canvas

The HTML canvas enables you to create HTML pages for Db2 Web Query. Designed for business users, you do not need to have experience in HTML, CSS, XML, or JavaScript. The HTML canvas is a graphical interface that is easy to use.

Note: HTML canvas files are intended to be created and edited with the HTML canvas only. You can either modify HTML pages created in the HTML canvas or create HTML pages using a text editor. When these pages are opened in the canvas, a message displays that allows you to decide whether to open the file in the canvas or the text editor.

If you are a web developer and want to write your own code instead of using the HTML canvas, you can use the authoring tool of your choice and also use RESTful Web Services.

Note: When opening an HTML file that was not created in the HTML canvas, you will be prompted to open the file only in a Windows Associated tool.

To access the HTML canvas, on the *Home* tab, in the *Content* group, click *HTML/Document*. When the HTML/Document Wizard opens, select *HTML Page* from the available options. The HTML canvas opens.

The HTML canvas uses HTML 5 Document Type Definitions (DTD).

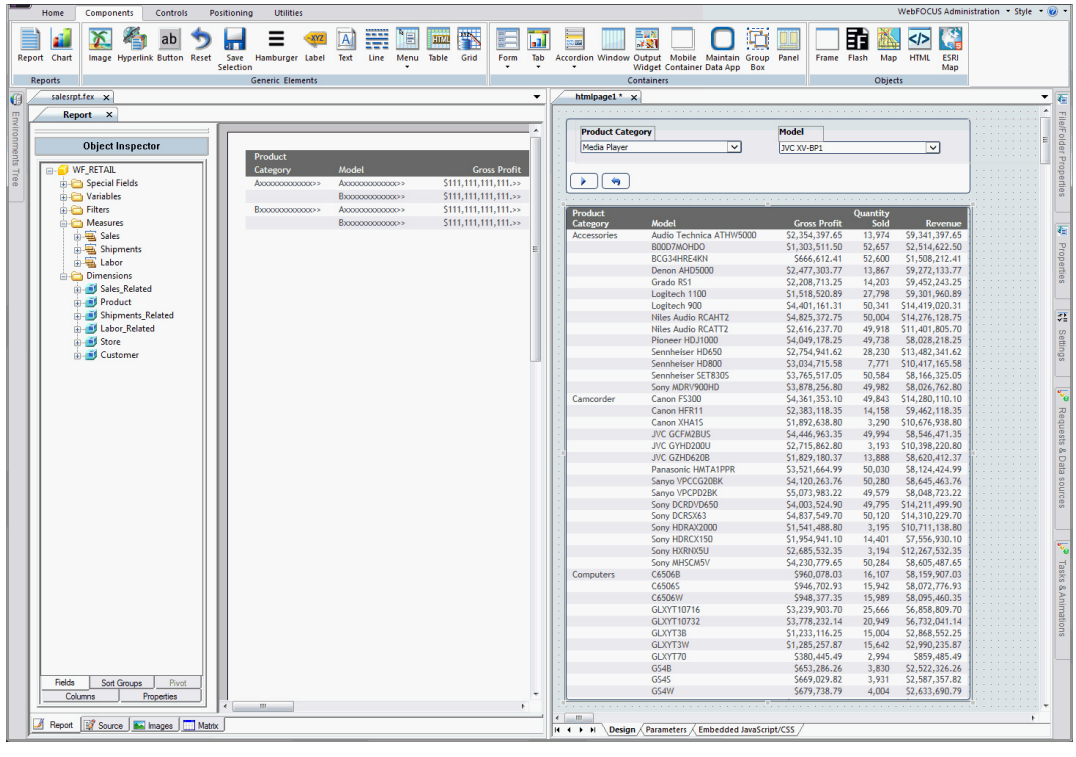
Creating and Styling Reports Using the Report Canvas

The Report canvas provides you with many features that enable you to create and style reports. To access the Report canvas, on the *Home* tab, in the *Content* group, click *Report*. The Report Wizard opens. This is the InfoAssist tool that is available in the Web Query browser interface. Follow the prompts in the Report Wizard to create a report. The Report canvas opens.

Customizing Canvases

You can switch between the canvases by clicking the colored tab at the top of the canvas. You can also develop in the canvases side by side.

The following image shows an HTML canvas next to a Report canvas.



Procedure: How to Move Canvases for Side-by-Side Development

To create side-by-side canvas development, follow these steps:

1. Drag one canvas tab away from its original position.

A pointer with a document icon  appears.

2. Drag the pointer with the document icon to the upper-right corner of the canvas.

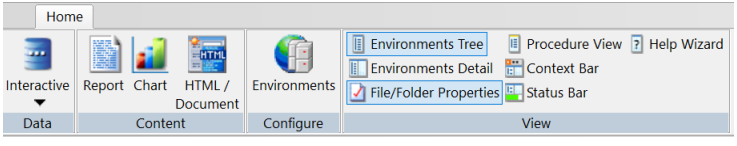
   The canvases now appear side by side.

   Tip: Repeat the preceding steps to develop on additional canvases side by side.

Accessing Features With the Ribbon

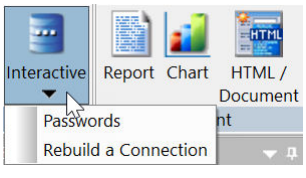
The Developer Workbench ribbon replaces traditional toolbars with a single command bar that organizes commands and controls into a series of static and contextual tabs. From the ribbon, you can also access the Style menu and the Help menu.

The default ribbon is shown in the following image.



Each tab on the ribbon, whether static or contextual, contains commands and controls that are organized into groups, according to their function or relationship. All tabs are named and all groups are labeled. Icons are used to represent the commands or controls in each group. By default, all icons are labeled.

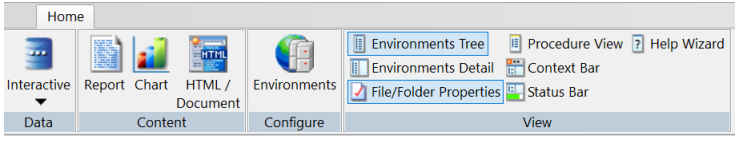
In addition to single-click commands, some commands on the ribbon contain a submenu of options that opens when you click the arrow associated with the command. These arrows can appear below or next to a label. An example of this is the Interactive command, which opens a submenu of options, as shown in the following image.



Note: There is no default ribbon size. As your monitor or application window size changes, the commands and controls on the ribbon resize themselves to fit the available space. Large monitors or application windows will display larger icons and entire groups. Smaller monitors or application windows may display groups as a single, labeled icon that opens a submenu.

Building an Application Using the Home Tab

The Home tab enables you to access the features and functions necessary to build an application, create new components, open existing components, run existing components, and edit components. The Home tab contains the Data, Content, Configure, and View groups. The Home tab is always available and is static. It is located in the upper-left corner of the Developer Workbench interface, below the Quick Access Toolbar, as shown in the following image.



Note: The Home tab is the only static tab in Developer Workbench.

Managing Data Using the Data Group

The Data group includes tools that you can use to access and rebuild a data file. The Data group is shown in the following image.

The commands in the Data group are:

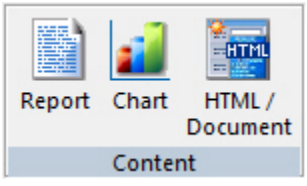
Interactive

Contains the following options:

- **Passwords.** The Password dialog box opens. You can enter the password for the data source you have selected in your Environments Tree panel.
- **Rebuild a Connection.** The Rebuild dialog box opens. You can rebuild a disorganized file, index a specific field from a file, or check the integrity of a file.

Creating an Application Using the Content Group

The Content group contains the components or tools that you can use to create an application. These include: Report, Chart, and HTML/Document. The Content group is shown in the following image.



The commands in the Content group are:

Report

Opens the Report canvas in report mode after you enter a procedure name and select a location for your report from the Report Wizard.

In report mode, you can create and style simple or complex reports, add data to the Report canvas, and style that data creating a graphical representation of the report page. This allows you to view how the report displays at run time.

Using report mode, you can:

- Display and sort data.
- Select records.
- Include totals, subtotals, column calculations, headings, footings, and images.
- Format columns.
- Style fonts, colors, and grids.
- Add drill downs to detailed reports and URLs.
- Save output in many types of formats for display and reuse.
- Create HTML Analytic Documents.

Chart

Opens the Chart canvas after you enter a procedure name and select a location for your chart from the Chart Wizard.

In chart mode, you can easily transform almost any type of data into effective custom charts. You can create a variety of charts, such as bar charts, line charts, scatter charts, or pie charts, to help users analyze data in a different way. You can include selection criteria by defining parameters for your data. Additionally, you can apply drill-down capabilities and conditional styling to highlight specific data on a chart.

The Chart canvas contains a ribbon of features and options that allow you to add and style the data in your chart. Contextual tabs open within the Chart canvas, rather than on the Developer Workbench ribbon.

HTML/Document

Opens the HTML/Document Wizard.

HTML mode provides the features you need to create HTML pages so that end users can interact with your data. You can build and customize an HTML page in the HTML canvas, using the contextual tabs on the ribbon to add objects to the page. HTML mode is fully integrated with JavaScript and cascading style sheets (CSS).

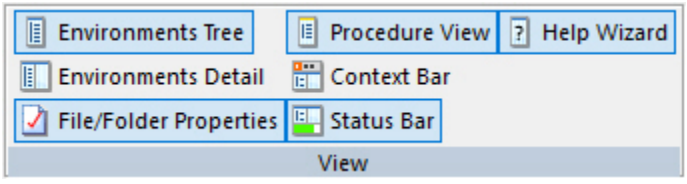
In the HTML canvas, you can:

- Build an HTML launch page.
- Add push buttons, hyperlinks, and other controls to launch Db2 Web Query reports in your application.
- Create a launch page for one or more reports that contain parameters.
- Create a complete HTML page by adding multiple reports and charts.
- Create an advanced report layout by including images, frames, and other web objects.
- Modify the location, size, and properties of all objects in your page layout.
- Set background, font, and other page properties.

In the Document canvas, you can coordinate and distribute layouts made up of multiple reports and charts in a single document. You can position reports and charts anywhere on a single page or combine a series of layouts within a single document. When creating compound reports in the Document canvas, PDF, HTML, HTML and PDF Analytic Documents, PowerPoint, and Excel are available as output formats.

Controlling the Display of the Screen Using the View Group

The View Group enables you to control what displays on your screen. For example, you can choose whether to display or hide the Environments Tree or Help Wizard. The View group is shown in the following image.



The commands in the View group are:

Environments Tree

Displays or hides the Environments Tree panel, which displays on the left side of the screen and displays environment information.

Environments Detail

Displays or hides the Environments Detail panel.

File/Folder Properties

Displays or hides the File/Folder Properties panel.

Procedure View

Displays or hides the Procedure View panel.

Context Bar

Displays or hides the Context Bar. The Context Bar displays under the ribbon.

Status Bar

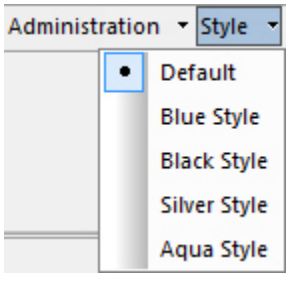
Displays or hides the Status Bar. The Status Bar displays along the bottom of the screen.

Help Wizard

Displays or hides the Help Wizard. The Help Wizard content dynamically changes, depending on where you are in the interface.

Customizing the Appearance of the Interface Using the Style Menu

You can customize the appearance of the Developer Workbench interface from the Style menu. From this menu, you can select a theme with which to customize the Developer Workbench interface. The themes available to you are determined by the themes that you have installed for Microsoft® Office. The Style menu is always available. It is located in the upper-right corner of the Developer Workbench interface, as shown in the following image.

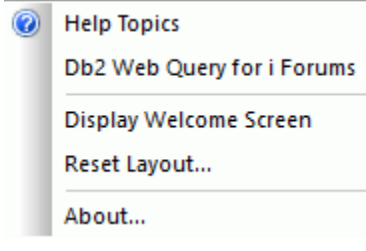


Note:

- When you close Developer Workbench, the theme that you have selected is saved and used the next time you launch the product.
- When using a style other than Default, the drop-down arrow is removed from the panels. You can still access the menu by right-clicking the top frame of the panel.

Accessing Db2 Web Query Developer Workbench Help Content

You can access the Developer Workbench Help content, *Db2 Web Query for i Forums*, Developer Workbench licensing information, and other product options from the Help menu. The Help button is always available. It is located in the upper-right corner of the Developer Workbench interface, as shown in the following image.



From the Help menu, you can do the following:

- Access the online Help system. Click *Help Topics* to open the Developer Workbench Help window.

- Access the IBM developerWorks® forums. Click *Db2 Web Query for i Forums* to access our online developer center and more than a message board. It is an interactive network of developers from almost every profession and industry, collaborating on solutions and sharing tips and techniques.
- Reopen the Welcome screen after it has been closed. Click *Display Welcome Screen* to reopen the Welcome screen.
- Reset the interface layout. Click *Reset Layout* to return the Developer Workbench interface settings, such as the ribbon, Quick Access Toolbar, panels, and styling, to their default. This change will not occur until Developer Workbench is restarted.
- View license and product information. Click *About* to open the About Db2 Web Query for i Developer Workbench dialog box.

Working With the Environments Tree Panel

The Environments Tree panel is the primary navigational aide in Developer Workbench. Using it, you can always stay oriented to the Repository or web application that you are working in. You can use the panel to move or copy objects between folders, copy folders, share content with other Developer Workbench users, or hide content.

Note:

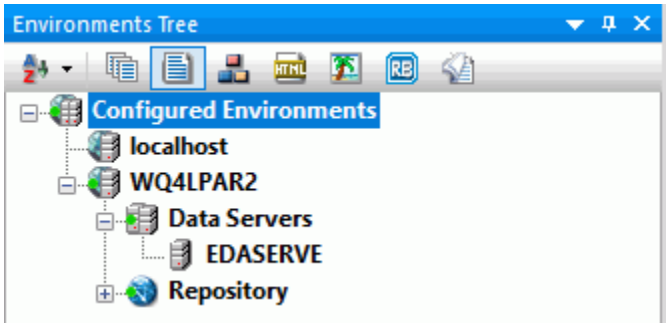
- Take care when copying and pasting Content folders from one environment to another. Content may be affected if the metadata in the source environment does not match the metadata in the target environment. The use of Change Management packages is recommended to move Content folders from one environment to another in Developer Workbench.
- When you copy and paste from the Repository area to the Data Servers area, the IBFS properties (metadata) are lost and cannot function. The same issue occurs if you copy and paste from the Repository area to your desktop, and then copy and paste from your desktop back to the Data Servers area.

From the Environments Tree panel, you can:

- Customize the appearance of the panel.
- Filter content listed under the Db2 Web Query environments in the Configured Environments tree.
- Access configured Db2 Web Query environments to create new content or modify existing content.

- Refresh an application or folder in the Configured Environments tree, to view new files and updated file information, and omit deleted files.

The Environments Tree panel is shown in the following image.



The Environments Tree panel opens with Developer Workbench.

Reference: **Environments Tree Toolbar**

The Environments Tree toolbar contains the buttons and menus that you need to navigate, filter, and sort the information that appears in the Configured Environments tree.

View Options

Presents options for viewing items (for example, alphabetical sorting or grouping), as described in the following table.

Option	Name	Function
	View items sorted in Alphabetical order	Sorts the files alphabetically from A to Z.
	View items sorted in reverse Alphabetical order	Sorts the files alphabetically from Z to A.
	View items sorted in Chronological order	Sorts the files by the last saved time, in descending order.
	View items sorted in reverse Chronological order	Sorts the files by the saved time, in ascending order.

Option	Name	Function
	View Items grouped by File Type	Sorts the files by file type.
	Respect Sort Order Property	Respects the Sort Order property value specified in the File/Folder Properties panel. This is the default.
	View items by Title	Displays the files by title and sorts the files by title. If an item does not have a title, its name is shown.
	View items by Name	Displays the files by name and sorts the files by name. If an item does not have a title, its name is shown.
	Refresh View	Refreshes all of the files and folders that you see in the Configured Environments tree. *Refresh View* also shows the dependencies of the file or folder that you have selected.
	Show All Content files	Filters the tree to show all content file types, except for Master Files. To view Master Files, click *Show only Master files*.
	Show only Procedure files	Filters the tree to show only this file type.
	Show only Master files	Filters the tree to show only this file type.
	Show only HTML files	Filters the tree to show only this file type.
	Show only Image files	Filters the tree to show only this file type.
	Show only Report Broker files	Filters the tree to show only this file type.
	Other files	Filters the tree to show other file types.

Viewing Nodes in the Configured Environments Tree

The Configured Environments tree displays the Db2 Web Query Environments, Repository, and Data Servers nodes. These are the development areas where you can create content.

If a node under the Configured Environments Tree has been populated and contains content, a green dot is shown on the node icon. If a node has been populated and does not contain any content, a red dot is shown on the node icon. If a node has not been populated, Developer Workbench does not know if there is any content in that node and no dot is shown on the node icon.

Note: By default, the Configured Environments tree sorts files and folders by title. If you change the sorting options to *View Items by Name*, the names of your files and folders will display. Whether you display content by title or by name, the Repository and Data Servers nodes do not display differently.

You can expand the nodes on the Configured Environments tree to view your configured environments, folders, and application files. Use the filtering commands on the toolbar to sort files alphabetically, display by name or title, or include the associated paths.

From the Configured Environments tree, you can manage content for each of your Db2 Web Query environments. This includes creating new folders or files, opening existing files, and copying files. Right-click a node, folder, or file to view the shortcut menu of options. You can also open files by double-clicking them.

You can also use the Refresh Descendants option to refresh a specific application or folder. Right-click the application or folder that you want to refresh, and then click *Refresh Descendants*.

Note:

- You can use the shortcut menu on any file, select *New*, and click the type of content you want to create, to create new content in the same location.

You can copy files from your desktop to the Configured Environments Tree.

From the Configured Environments node, you can access the following shortcut menu option:

- **Add.** Opens the Db2 Web Query Environments Properties dialog box where you can add additional environments.

From your Db2 Web Query environment, in the Environments Tree panel, you can access the following shortcut menu options:

- **Db2 Web Query for i Home Page.** Opens the Db2 Web Query Home Page. The Home Page serves as a centralized place for working with your data, creating, organizing, and sharing content, scheduling reports and procedures, and performing administrative tasks.
- **Sign In.** Allows you to sign in to your Db2 Web Query environment. This option is only available if you are not already signed in to your Db2 Web Query environment.
- **Sign Out.** Allows you to sign out of your Db2 Web Query environment. This option is only available if you are already signed in to your Db2 Web Query environment.

Note: You can copy files that are not in Developer Workbench and paste them in the Configured Environments tree to use those files when creating your application.

Reference: Data Servers

The Data Servers node lists each Reporting Server that Web Query can access.This node expands to display the files that you can use to develop Db2 Web Query applications successfully.

The process of populating the application paths of the files in the Data Servers node is done on a background thread. While this background thread is running, you can navigate on the tree, open a folder, and create files. However, you will be unable to save a new file, or do anything that requires a path list, such as create a new style sheet, until the background thread completes. This section of the Configured Environments tree, which is visible, will populate.

From a data server, you can access the following shortcut menu options:

- **Db2 Web Query Reporting Server Console.** Opens the Reporting Server Console where you can manage applications and data sources, and administer the server.
- **Refresh Descendants.** Refreshes the contents of the application or folder in the Configured Environments tree to omit deleted files, and show new files and file information, such as date, time, and size.

From the Applications folder, you can access the following shortcut menu options:

- **New Application Directory.** Creates a new application folder.
- **Configure Application Path.** Opens the Application Path dialog box from which you can create applications, add applications to the search path, map and reorder applications, delete applications or mappings from the path.

- **Refresh Descendants.** Refreshes the contents of the application or folder in the Configured Environments tree to omit deleted files, and show new files and updated file information, such as date, time, and size.

From an application folder or a file, you can access the following shortcut menu options:

- **New.** Creates a new folder or file. The following shortcut menu options are available when pointing to *New*:
  - **Application.** Creates a new application folder in your current application folder.
- **Duplicate.** Creates a copy of the selected file in the Data Servers node.
- **Copy.** Copies the selected file to the clipboard where you can paste it to a different node.
- **Paste.** Pastes the cut or copied file.
- **Rename.** Changes the name of the selected folder or file.
- **Delete.** Deletes the selected folder or file. You will be prompted to confirm your deletion. Click *No* if you do not want to delete an item, or click *Cancel* to abort the delete operation. If you choose to abort the delete operation while deleting a list of items, Cancel will prevent all subsequent items from being deleted.
- **Properties.** Displays the folder properties in the Data Servers node.
- **Refresh Descendants.** Refreshes the contents of the application or folder in the Configured Environments tree to omit deleted files, and show new files and updated file information, such as date, time, and size.

Reference: Repository

The Repository node is storage space for data or information, and allows you to use Developer Workbench to administer and develop against a Db2 Web Query environment. The Repository node lets you manage resources and applications on remote servers, as well as on your local machine if you have performed a full installation of Developer Workbench. You can create and edit application files on all remote servers from one easily accessible interface. You can also create and administer reports from a Windows application rather than a web browser.

From a Db2 Web Query node, you can access the following shortcut menu options:

- **Impact Analysis.** Enables you to generate a list that identifies the procedures that access a specific Master File or field within a Master File.
- **New Folder.** Creates a new folder.

- **Mode Manager.** Allows you to edit private files, if you have the proper authorization setting.
- **Physical View** or **Shared View.** Enables you to toggle between the two. Shared View displays all folders (including empty folders) in the hierarchy that includes shared content. Physical View suppresses empty folders in the hierarchy that includes shared content. Physical View displays all folders that contain shared content.
- **Refresh Descendants.** Refreshes the contents of the application or folder in the Configured Environments tree to omit deleted files, and show new files and updated file information, such as date, time, and size.

From a folder or file, in the Repository node, you can access the following shortcut menu options:

- **Impact Analysis.** Enables you to generate a list that identifies the procedures that access a specific Master File or field within a Master File.
- **New.** Creates a new folder or file. The following shortcut menu options are available when pointing to *New*:
  - **Procedure Using InfoAssist.** Creates a new procedure folder or file in your current application folder.
  - **Reporting.** Expands a submenu containing the following options.
    - **Report.** Creates a new report.
    - **SQL Report.** Creates a new SQL report.
  - **HTML File.** Creates a new HTML page.
  - **InfoAssist.** Expands a submenu containing the following options.
    - **Chart.** Creates a new chart using the Chart canvas.
    - **SQL Chart.** Creates a new SQL chart.
    - **Visualization.** Creates a new visualization.
  - **Text Editor.** Expands a submenu containing the following options.
    - **Cascading Style Sheet.** Creates a new cascading style sheet.
    - **Db2 Web Query StyleSheet.** Creates a new Db2 Web Query StyleSheet.
  - **Folder.** Creates a new folder.

- **Publish/Unpublish.** Allows or suppresses public access to every file within the selected folder. Multiple folders can be selected at one time by pressing the Ctrl key while selecting the desired folders.
- **Hide/Show.** Hides or shows every file in the selected folder, to users who are permitted to access the folder. Multiple folders can be selected at one time by pressing the Ctrl key while selecting the desired folders. You can specify which users can see this content by changing the *Do not show on user's list* property in the File/Folder Properties panel.
- **Duplicate.** Creates a copy of the selected file in the Domains node.
- **Copy/Paste.** Copies or pastes a folder or file.
- **Rename.** Changes the name of the selected folder or file.

  To ensure that all instances of the original name and references are updated, it is recommended that you use the Save As option to rename an HTML page.
- **Delete.** Deletes the selected folder or file. You will be prompted to confirm your deletion. Click *No* if you do not want to delete an item, or click *Cancel* to abort the delete operation. If you choose to abort the delete operation while deleting a list of items, Cancel will prevent all subsequent items from being deleted.
- **Properties.** Displays the folder properties in the Repository node.
- **Refresh Descendants.** Refreshes the contents of the application or folder in the Configured Environments tree to omit deleted files, and show new files and updated file information, such as date, time, and size.

Reference: Node Population

Population is a check to see what content, if any, is contained within a node in the Configured Environments tree. Node population occurs when you select a node in the Configured Environments tree. The node icon will be different depending on whether population has occurred. If a node contains files after population a green dot appears on the node icon. If a node does not contain files after population, a red dot appears on the node icon.

If a node has been populated and contains content, deleting that content will change the green dot to a red dot. Conversely, if a node has been populated and contains no content, creating content in that node will change the red dot to a green dot.

Reference: Master File Shortcut Commands

The following shortcut menu options are available for a Master File:

- **Open in Text Editor.** Opens the Master File in the Text Editor canvas.
- **Edit in Windows Associated Tool.** Opens the Master File in your Windows associated tool. Note that this only appears if you have a Windows associated tool in which to open the file.
- **Sample Data.** Displays sample output for the Master File.
- **Impact Analysis.** Enables you to generate a report that identifies the procedures that access a specific Master File or field within a Master File.
- **Print.** Prints the Master File.
- **Check.** Tests the validity of the Master File.
- **Edit Access File as Text.** Enables you to view and manually edit the Access File synonym.
- **New.** Creates a new folder or file. The following shortcut menu options are available when pointing to *New*:
  - **Application Directory.** Creates a new application folder in your current application folder.
- **Duplicate.** Creates a copy of the selected file.
- **Copy/Paste.** Copies the selected file to the clipboard where you can paste it to a different node.
- **Rename.** Changes the name of the selected file.
- **Delete.** Deletes the selected file. You will be prompted to confirm your deletion. Click *No* if you do not want to delete an item, or click *Cancel* to abort the delete operation. If you choose to abort the delete operation while deleting a list of items, Cancel will prevent all subsequent items from being deleted.
- **Properties.** Displays the folder properties in the File/Folder Properties panel.

Note: When a Master File is opened in the Text Editor canvas or the Metadata canvas, all other open options from the shortcut menu are unavailable.

Opening Files in the Configured Environments Tree

In addition to opening files through the Application menu, or from the Quick Access Toolbar, you can also open files from the development areas in the Configured Environments tree.

- To open a file in its native canvas, double-click the file or right-click the file, and then click *Open*.
- To open a file in the Text Editor canvas, right-click the file and then click *Open in Text Editor*.
- To open a file in a Windows Associated Tool, right-click the file and then click *Edit in Windows Associated Tool*. This option appears only if a Windows associated tool is available. When you access this functionality, the file will open in a separate dialog box using the relevant Windows tool.

Note: Files can only be opened in one area of Developer Workbench at a time to ensure that your changes are not overwritten by another opened version of the file. However, you are able to open a file in Developer Workbench and in a Windows Associated Tool at the same time. To ensure that your changes are saved properly, you should only modify one opened version of the file at a time.

Setting Up Access to Db2 Web Query Environments

As you set up access to Db2 Web Query environments, your settings are retained in a file named wfscom.xml. Db2 Web Query environment settings are typically stored in the following locations.

drive:\Users*user_id*\AppData\Roaming\Information Builders\wfscom.xml

where:

user_id

Is your Windows user ID.

Note:

- This file and directory might not be visible, by default. To see this directory, open the Windows Control Panel, click *Folder options*, and then click the *View* tab. Select *Show hidden files, folders, and drives*, and click *OK*.
- User IDs and passwords stored in wfscom.xml are encrypted to keep them confidential.
- Db2 Web Query environment properties apply to your current Developer Workbench configuration.

Procedure: How to Add a Db2 Web Query Environment

A Db2 Web Query environment consists of a web server, a Db2 Web Query Client, and a Reporting Server. Adding Db2 Web Query environments lets you create and manage multiple environments, such as development, test, and production instances.

To add a Db2 Web Query environment, do the following:

- ❑ Specify a web server that includes a host name and port number.
- ❑ Configure the HTML alias/context root to access the Db2 Web Query Client.
- ❑ Provide the appropriate user credentials if web server authentication is required.

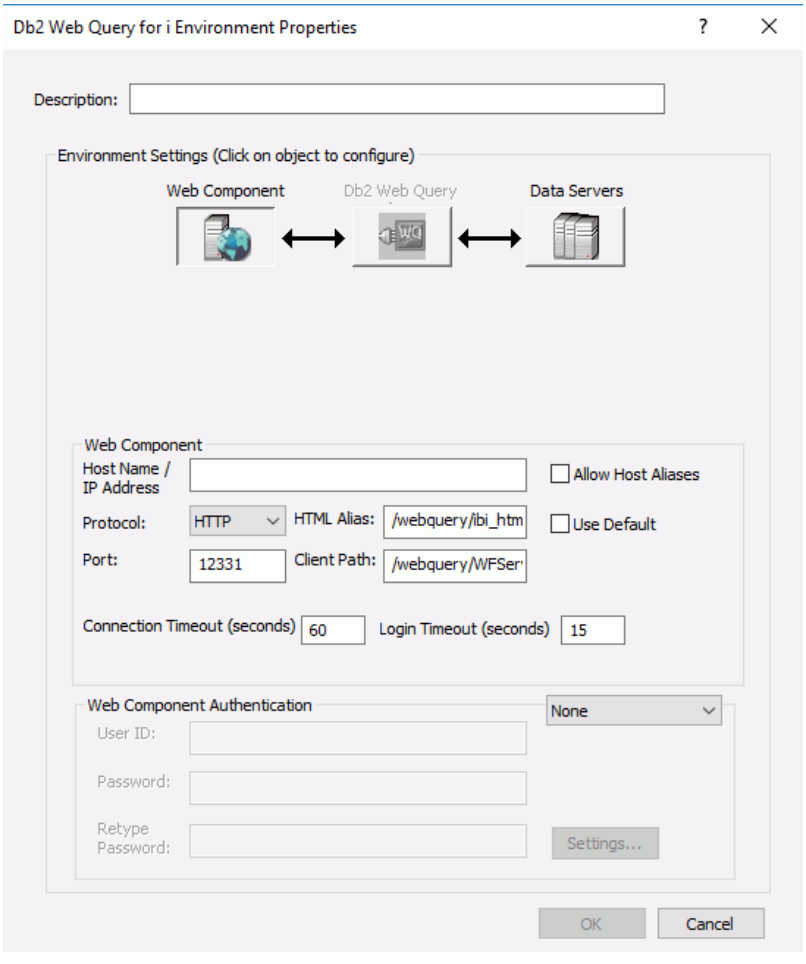
The following procedure provides the information you need to add a Db2 Web Query Environment.

1. On the *Home* tab, in the group, click *Environments*.

   The Environments List dialog box opens. This dialog box lists all Db2 Web Query environments defined for Developer Workbench.

2. Click *Add*.

   The Db2 Web Query Environment Properties dialog box opens, as shown in the following image.



   At the top of the dialog box is the Description field, followed by the Environment Settings area. The Environment Settings area contains a series of buttons that represent components in a Db2 Web Query environment, and the required parameter fields to configure that environment. When you configure an environment, the areas below the buttons can change according to the parameters that are required.

3. In the Description field, type a description for the Db2 Web Query environment. This description appears in the Configured Environments tree.

Note: The description cannot contain special characters, such as ., /, \, ?, *, and others. A message with characters not allowed appears if one of the restricted characters is detected during validation.

4. In the Web Component area, specify the web server information.

   For some environments, once you specify the web server, all other settings default. If the Db2 Web Query environment you are accessing does not use default settings, or the components require authentication, click the appropriate button in the Environment Settings area to provide the parameters. The areas that follow explain the parameters available for each component.

   Note: Db2 Web Query environment properties must be supplied in a specific order. For example, if web server security is enabled, you cannot set the Db2 Web Query Client script name until you have provided valid web server credentials. Similarly, you cannot retrieve a list of Reporting Servers until you have provided a valid Db2 Web Query Client Path.

   As you select a component button in the Db2 Web Query Environment Properties dialog box, Developer Workbench ensures that it has the necessary information before it displays the properties of that component in the lower part of the dialog box. If the required information is not available, you cannot proceed to the next component.

5. Click *OK*.

   The Db2 Web Query Logon - Db2 Web Query dialog box opens and prompts you for a Db2 Web Query ID and password.

   There is a check box option on the logon dialog box for remembering the ID or password. By default, it is not selected. If you select this box, your credentials are stored and encrypted in the wfscom.xml file, the local configuration file that stores information processed by the Developer Workbench communication layer.

   To clear stored credentials, open the Db2 Web Query Environment Properties dialog box and select the environment that users need to make changes. Click *Db2 Web Query Client*. Under Db2 Web Query Credentials, delete the User ID and Password information, and then clear the Supply Credentials check box.

   Note: A logon dialog box can also open for connection to the web server, application server, or Reporting Server, depending on the security implemented in the Db2 Web Query environment that is being accessed.

6. Type the appropriate user name and password, and then click *Logon*.

   The Db2 Web Query Environment Properties dialog box closes.

7. In the Environments List dialog box, click *OK*.

Specifying Web Component Properties

The *Web Components* button is typically selected by default. Web Component properties specify how Developer Workbench accesses the web server. The web server must be identified before any other components.

The following properties are available:

- **Host Name/IP Address.** Specifies the host name or IP address where your web server is running. This field is required and has no default value.
- **Allow Host Aliases.** If checked, enables you to configure multiple Db2 Web Query instances that have been installed on the same host machine.
- **Protocol.** The protocol to use for accessing the web server (HTTP or HTTPS).
- **Port.** The TCP/IP port for accessing the web server. Port 12331 is the default.
- **HTML Alias.** Identifies the alias used to access content from the Db2 Web Query environment directory, ibi_html. In Developer Workbench, there is no need to configure an ibi_html alias. The ibi_html content is accessed through the webquery alias. The configuration of this alias is:

  ```
  /webquery/ibi_html
  ```

  where:

  `webquery`

  Is customizable.

  `ibi_html`

  Is constant.

- **Client Path.** Specifies how calls are made from Developer Workbench to the web server. By default, when you add a new Db2 Web Query environment, it is set to use the Db2 Web Query Servlet with the default webquery context path:

  ```
  /webquery/WFServlet
  ```

  where:

  `webquery`

  Is customizable.

  `WFServlet`

  Is constant.

If the Db2 Web Query environment uses a non-default context path, clear the *Use Default* check box and provide the correct Client Path. For example:

```
/webquery/WFServlet
```

If the Client Path is incorrect for the environment, you receive an error when you click the *Db2 Web Query* button at the top of the page, or when you click *OK* to exit and save your changes.

If you do not know your path, ask your Db2 Web Query Administrator or check the Db2 Web Query Administration Console of the environment to which you want to connect. The Client Path settings for the environment are located under *Utilities* and *Client Selection*.

- **Use Default.** Specifies that the default ibi_html alias is used. Keep this option selected unless you change the HTML Alias value.

  Note:

  - If the Client Path field is empty, and the Use Default check box is selected, there is a problem connecting to the Db2 Web Query Client. Ensure your web server is started and that you typed the correct properties on the Web Components page. If you cannot connect, contact your Db2 Web Query Administrator.
  - If while providing a custom HTML Alias and Client Path, your connection fails because of incorrect information and the Client Path is deleted, you can clear *Use Default* to restore the values you entered. Click the *Use Default* check box to edit the HTML Alias and Client Path, and type the correct information. If a custom webquery alias or context path is used, it needs to be entered in both the HTML Alias and Client Path fields. For example:

    HTML Alias:

    ```
    /webquery/ibi_html
    ```

    Client Path:

    ```
    /webquery/WFServlet
    ```

- **Connection Timeout.** Default time to connect to a Db2 Web Query environment is 60 seconds. However, in some slow systems this needs to be increased to avoid connection failures when the timeout period is reached. If timeout issues are experienced, increase this to 240 or higher. Increasing the timeout can address issues with slow environments or environments that have many resources (applications/files) during development.

- **Login Timeout.** Specifies how long Developer Workbench should wait for the login credentials to be validated. The default time is 15 seconds. You can set this to a higher value, if necessary. You can configure this setting for each environment.
- **Web Component Authentication.** Specifies whether authentication is required on the web server. To set security, select *Basic* in the drop-down list, and type a web server user ID and password. If this is set to *None*, the web server allows anonymous access.

  To configure the web server for an environment that uses browser-based authentication protocols, such as x.509, select *Browser-based Login* in the drop-down list.

Db2 Web Query Properties

When you click the *Db2 Web Query* button, Developer Workbench makes a connection to your web server to retrieve information about the Db2 Web Query environment. Therefore, you have to first specify Web Component properties, and your web server must be running.

You are prompted to sign in to Db2 Web Query to verify your configuration. You will also be prompted to sign in to Db2 Web Query if you click *OK* to exit the dialog box.

The following properties are available:

- **Client Path.** Is specified in the Db2 Web Query Component properties area, and only appears in this pane for reference. This field is read-only and can only be changed in the Db2 Web Query Component properties area.
- **Select Language.** Specifies the language of the Db2 Web Query Client.
- **Supply Credentials.** Specifies whether to supply and store credentials when you connect to Db2 Web Query and access the environment.

When the correct Db2 Web Query Client Path is provided, you can specify properties for the remaining component.

Data Server Properties

You can set authentication and view available Reporting Servers by clicking the *Data Servers* button. When you select *Data Servers*, Developer Workbench connects to the Db2 Web Query Client and retrieves a list of servers from its communication configuration file (odin.cfg).

The following property is available:

- **Supply Credentials.** If selected, you can type a Db2 Web Query Reporting Server ID and password for the server highlighted in the list. Clicking *Set* stores the credentials with the environment properties, and the ID entered is shown next to the server in the list. The credentials are checked the next time you use a feature on that server, not when you click *Set*.

Procedure: How to Edit an Existing Db2 Web Query Environment

1. On the *Home* tab, in the group, click *Environments.*

   The Environments List dialog box opens.

2. Select the environment that you want to edit.
3. Click *Properties*.

   The Db2 Web Query Environment Properties dialog box opens.

4. Edit the Db2 Web Query environment accordingly, and then click *OK* to accept your changes.
5. In the Environments List dialog box, click *OK*.

Mode Manager

If you have the *manage private resources* permission, you can use Mode Manager to view and edit private files. To enable Mode Manager, right-click on the Repository node and click *Mode Manager*. Your view refreshes, displaying all private files in the repository. You can open and make changes to these private files as needed. When you are finished, right-click on the Repository node and click *Mode Normal*. Your view refreshes, hiding the private files in the Repository node.

Note: When a file in the repository is marked as shared, and you do not have permission to save the original file, you will only be allowed to use the *Save As* command to save the file. This will allow you to save the file under a different name, leaving the original file untouched.

Working With the Environments Detail Panel

The Environments Detail panel is an optional panel, which contains panel customization commands, the Environments Detail toolbar, the Configured Environments tree, and the Details pane. To display the Environments Detail panel, on the *Home* tab, in the *View* group, select the *Environments Detail* button.

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.

From the Environments Detail panel, you can:

- Customize the appearance of the panel.
- Filter content listed under the Db2 Web Query environments in the Configured Environments tree.
- Access configured Db2 Web Query environments to create new content or modify existing content.
- Refresh an application or folder in the Configured Environments tree, to view new files and updated file information, and omit deleted files.

When you select a folder in the Configured Environments tree, the content within that folder is shown in the Details pane.

Reference: Environments Detail Toolbar

The Environments Detail toolbar contains the buttons and menus that you need to navigate and filter the information that appears in the Configured Environments tree.

View Options

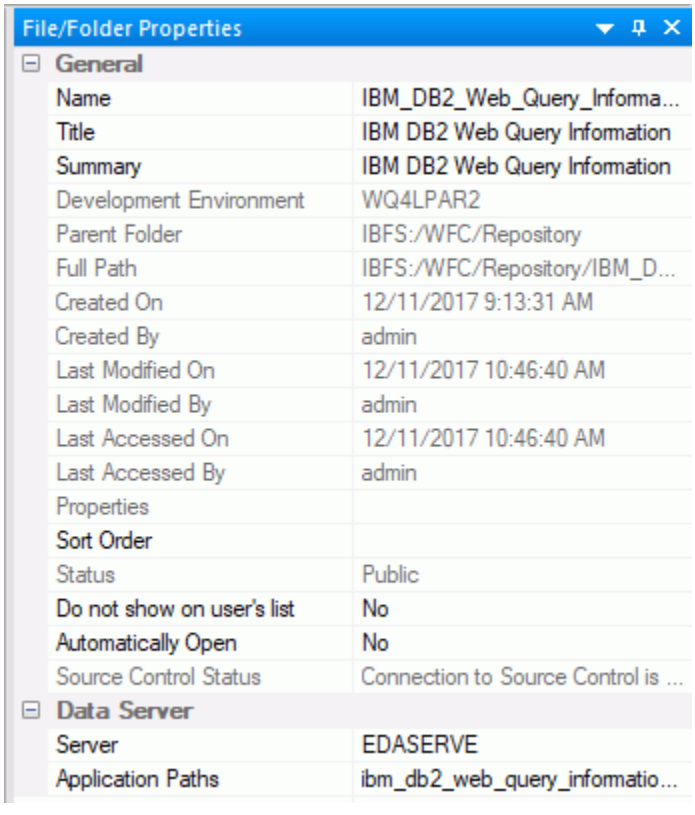
Presents options for viewing items (for example, alphabetical sorting or grouping), as described in *Environments Tree Toolbar* on page 45. Note that the available icons for the Environments Detail toolbar and the Environments Tree toolbar are the same.

Note: To sort content in the Environments Detail panel, you do not use the View Options command. Instead, you would click the columns at the top of the Details pane to sort content by name, size, type, last modified, or location.

Viewing File and Folder Properties

The File/Folder Properties panel shows overview properties of a folder or file selected in the Environments Tree panel. For example, if you highlight a folder in the Environments Tree panel, the File/Folder Properties panel shows information like the name and title of the folder and the date it was last modified. If you highlight a report file in the Environments Tree panel, the File/Folder Properties panel shows things like whether the report is enabled for Auto Link or Auto Drill.

The File/Folder Properties panel is shown in the following image.



The options within this panel change according to the type of file or folder you select. You can view the properties of a file or folder in the Data Servers and repository areas of the Configured Environments tree.

Copying Property Values in the File/Folder Properties Panel

You can copy a property value from the File/Folder Properties panel. Right-click a property value and click *Copy* to copy it to the clipboard. This is particularly useful if you want to capture the full path of an open file.

Editing Properties in the File/Folder Properties Panel

The following table defines the properties that you can change for each node within the File/Folder Properties panel.

Note: The changes that you make in the File/Folder Properties panel are not reflected in the Configured Environments tree until you click outside of the panel.

Data Servers

Element	Editable Panel Properties
Node	❑ **Title.** Changes the title of the Data Server node in the Configured Environments tree. ❑ **Server Path.** Indicates the fully-qualified path to the server.
Primary Folder	❑ **Title.** Changes the title of the primary folder in the Configured Environments tree. ❑ **Server Path.** Indicates the fully-qualified path to the server.
Subfolder	❑ **Name.** Changes the name of the selected subfolder in the Configured Environments tree.
File	❑ **Name.** Changes the name of the selected file in the Configured Environments tree.

Repository

Element	Editable Panel Properties
Node	❑ **Title.** Changes the title of the Repository node in the Configured Environments tree. ❑ **Server Path.** Indicates the fully-qualified path to the server.

Element	Editable Panel Properties
Folder	❑ **Name.** Changes the name of the folder in the Configured Environments tree. ❑ **Title.** Changes the title of the folder in the Configured Environments tree. ❑ **Summary.** Adds summary information to the folder properties. ❑ **Sort Order.** Specifies the order in which to list the folder or item in the tree. ❑ **Automatically Open.** This value is set to No, by default. If Yes is selected, this folder will automatically be open in the next session of Developer Workbench. ❑ **Server.** This value defaults to the server that is currently in use. You can use the drop-down list to affiliate a different server with the folder. ❑ **Application Paths.** This value defaults to the application path indicated for the selected folder. You can modify this value as needed.

Element	Editable Panel Properties
File	❑ **Name.** Changes the name of the file in the Configured Environments tree. ❑ **Title.** Changes the title of the file in the Configured Environments tree. ❑ **Summary.** Adds summary information to the folder properties. ❑ **Sort Order.** Specifies the order in which to list the file in the tree. ❑ **Shown to other users.** This value is set to Yes, by default. If Yes is selected, this file will display on the user list. If No is selected, the file is hidden from other users. ❑ **Prompt for parameters.** This value is set to Yes, by default. If No is selected, you will not be prompted for parameters. ❑ **Only run as deferred report.** This value is set to No, by default. If you select Yes, the report or procedure will run only as a deferred report. ❑ **Use Title for Deferred Report Description.** This value is set to No, by default. If you select Yes, the title is used instead of the name, as the description for a deferred report. ❑ **Schedule only.** This value is set to No, by default. If Yes is selected, you will only be able to schedule the report or procedure. ❑ **Enable Auto Linking.** This value is set to No, by default. If Yes is selected, Auto Linking enables you to develop and maintain numerous referenceable reports and charts that can be linked together. ❑ **Auto Link Target.** Indicates whether the file is set as a target for an Auto Link drill down. This value is set to No, by default.

Element	Editable Panel Properties
File	❑ **Enable Auto Drill.** This value is set to No, by default. If Yes is selected, Auto Drill allows you to navigate through different levels within the dimension hierarchy of your data source. ❑ **Server.** This value defaults to the server that is currently in use. You can use the drop-down list to affiliate a different server with the file. ❑ **Application Paths.** This value defaults to the application path indicated for the selected file. You can modify this value as needed.

Application Paths Dialog Box

The Applications Paths of a Db2 Web Query folder display in the File/Folder Properties panel when you select the folder. You can add, remove, or reorder the associated application paths in the Application Paths dialog box.

Open the Application Paths dialog box by clicking the ellipsis button next to the Associated Paths information. The Application Paths dialog box displays a list of applications that you can associate with your domain and a list of applications that are currently associated with your domain.

Adding or Removing Applications

To add an application to the domain, double-click the name of the application in the Available list. Click *OK* to view your changes in the File/Folder Properties panel.

To remove an application from the domain, double-click the name of the application in the Selected list. Click *OK* to view your changes in the File/Folder Properties panel.

Reordering Applications

To reorder an application in the Available list, click the name of the Application and then click the up or down arrows. Click *OK* to view your changes in the File/Folder Properties panel.

Editing Areas of Functionality Using the View Tabs

You can use the view tabs to edit different areas of functionality for the same document. These tabs show information that would be inaccessible from any other view tab.

Designing the Layout of a Page Using the Design View Tab

The Design View tab is accessible when you are creating an HTML page. From here, you can design the layout of your page by adding controls and components to the canvas. This tab is selected by default when you are creating an HTML page.

Creating and Modifying Parameter Values Using the Parameters View Tab

The Parameters View tab is accessible when you are creating an HTML page or In-Document Analytic dashboard. From here, you can create and modify parameter values and input controls, and customize parameter conditions. You can also bind parameters to controls and chain controls to one another.

The Parameters tab consists of the following components:

- Input control objects.

  You may select the input control object to view and edit the Properties and settings of the control.

  - Creating an input control from the Design view prompts you to create a bound parameter on the Parameters tab.
  - Editing an input control, which is inserted when setting input controls for new parameters.

- Add new parameters.

  Right-click anywhere on the Parameters tab to add a new parameter.

  Note: Manually adding a parameter creates an unbound parameter.

- Refresh unresolved parameters.

  All parameters on the parameters tab are parsed every two minutes to check if any are unresolved. If there are any unresolved parameters, their surrounding polygon is colored red. If you want to check for unresolved parameters on demand, right-click and select *Refresh unresolved*.

- Binding controls and parameters.

  Input controls and parameters can be bound and unbound from the Parameters tab.

  You may bind a parameter to an input control, or bind an input control to a parameter.

  - Binding a parameter to a control makes it an incoming parameter that will populate the control. Drag a parameter object to a control object on the Parameters tab.

- Binding a control to a parameter will populate the parameter. Drag a control object to a parameter object on the Parameters tab.

- Chain one control to another.

  Chaining will populate controls based on the selected value from the prior control in the chain. You can chain static and dynamic controls, link or unlink parts of a chain, and create conditions on links in a chain. Chains are represented by lines connecting control objects on the Parameters tab. By clicking the arrow head in a link of a chain, the Properties and settings dialog box enables you to modify and set properties and conditions of the chain.

  Note: Chaining is applicable only for controls, not parameters.

Creating JavaScript and Cascading Style Sheets Using the JavaScript and CSS View Tabs

The Embedded JavaScript and Embedded CSS View tabs are accessible when you are creating an HTML page. From here, you can create JavaScript code and cascading style sheets (CSS) for use in your HTML page. You can reference existing cascading style sheet files and JavaScript files by typing the URL in the URL/Find File area of the Settings panel. You can also search for these files by using the CSS or JavaScript commands.

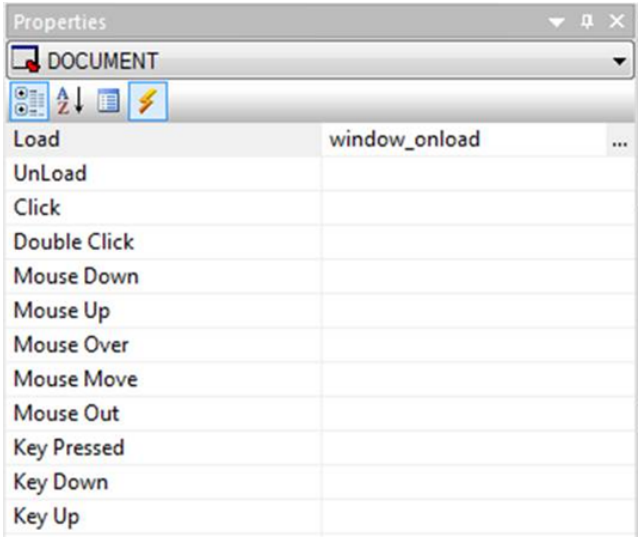
Embedded code is used only in the host HTML file. External code is available to be used by multiple files.

Procedure: How to Customize Titles Using jQuery Tooltips

The following procedure describes how to customize titles, using jQuery and Internal CSS.

1. Create an HTML page.
2. Add a component or control to the page.
3. Navigate to the Properties panel.
4. Type a value for the Title property, under the Core attributes section.
5. Click the canvas to select the *DOCUMENT* object.
6. On the Properties panel, click the lightning bolt icon on the toolbar to show the events.

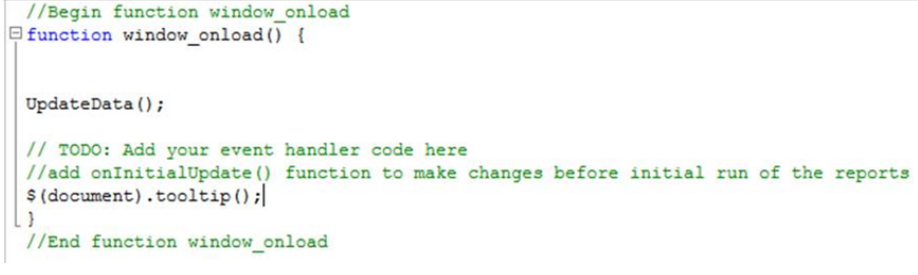
7. For the Load event, click the column on the right to display the ellipsis button, as shown in the following image.



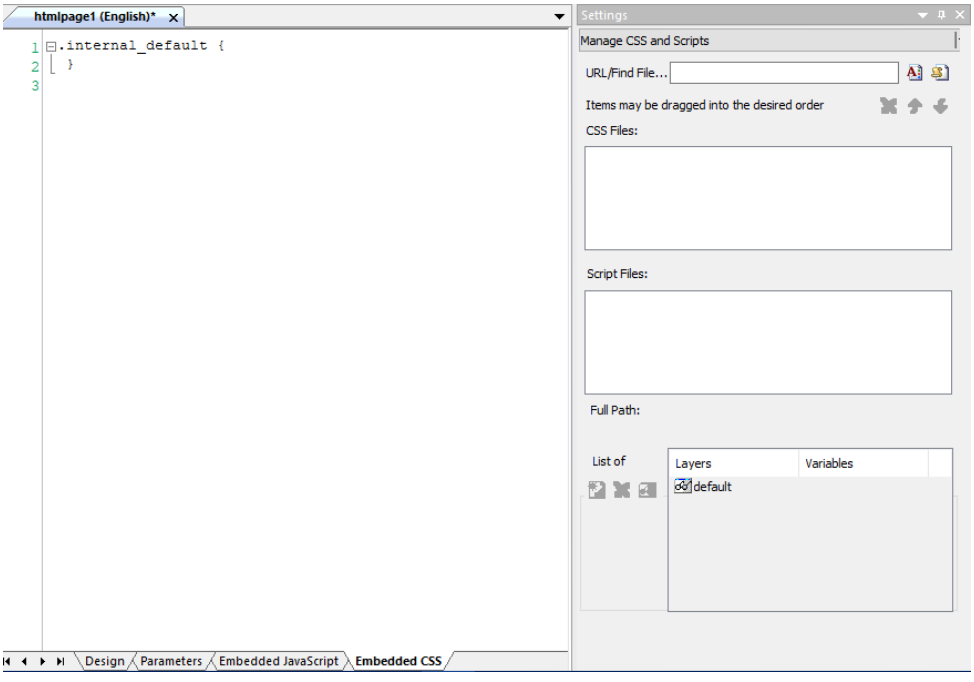
8. Select the Embedded JavaScript view tab on the HTML canvas.

9. After the comment lines, add a new line, and type the following code, as shown in the following image.

```
$(document).tooltip();
```



10. Navigate to the Embedded CSS tab, as shown in the following image.



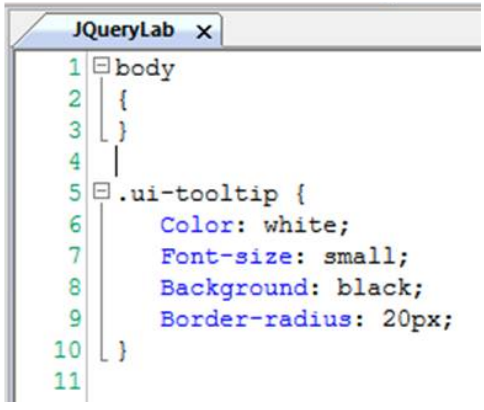
11. Enter a new line after the first class definition and type the following as the new class name.

```
.ui-tooltip
```

12. Add the following CSS Styles, as shown in the following image.

- ❏ Color: white;
- ❏ Font-size: small;
- ❏ Background: black;

- Border-radius: 20px;



Note: You can add any styling that you want.

13. Save and run the HTML page. You can hover the mouse over objects to show the styled tooltips.

Customizing the Panels

Similar to the ribbon tabs, Developer Workbench features open panels that provide additional options to assist in the development of your application. Not all panels open at the same time, nor do they provide the same options. When you launch Developer Workbench for the first time, only the Environments Tree panel appears docked to the edge of the canvas. The File/ Folder Properties panel also appears, but is auto hidden.

All panels, except the Environments Tree panel, are auto hidden by default. You can use the panel customization options, which are located in the upper-right corner of the panel, to change the appearance and location of the panels around the canvas.

Procedure: How to Rearrange Panels on the Canvas

To rearrange a panel on your canvas using the panel properties menu, follow these steps:

1. Click the *Window Position* button .

   The panel properties menu opens.

2. Select an option from the panel properties menu.

   The options available are:

   - **Floating.** Undocks the panel and allows you to move it freely around the canvas.

- **Docking.** Docks the panel onto the edge of your canvas where it last appeared.
- **Tabbed Document.** Opens the panel in the canvas area, as a tab. This option is unavailable in Developer Workbench.
- **Auto Hide.** Hides the panel as a tab on the edge of your canvas. When you pause over the tab, the panel reopens.
- **Hide.** Closes the panel.

Note: You can also open the panel properties menu if you right-click the top of the panel.

Procedure: How to Pin and Unpin Panels to the Canvas

You can pin a panel to the canvas to keep it open while you develop your application. When you unpin the panel, it becomes auto hidden again, and appears as a tab on the edge of the canvas. When you hover the mouse over this tab, the panel reopens.

- To pin a panel to your canvas, click the *Auto Hide* horizontal pin button  .
- To unpin a panel to your canvas, click the *Auto Hide* vertical pin button  .

Procedure: How to Close and Reopen Panels

- To close a panel, click *Close*  in the upper-right corner of the panel.
- To reopen the panel, go to the *Home* tab, and in the *View* group, select the appropriate panel button.
- To reopen a panel in the HTML canvas, go the *Utilities* tab, and in the *View* group, select the appropriate check box.

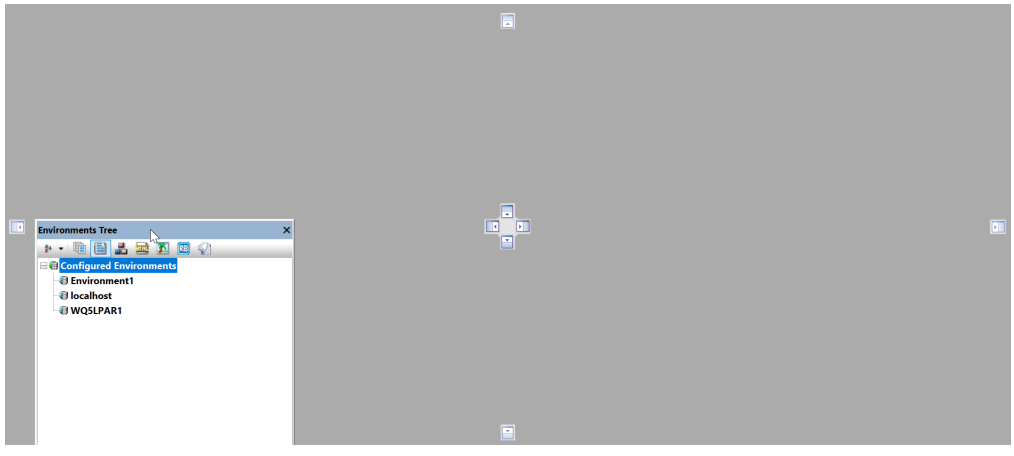
Using Handles

You can dock floating panels within the canvas to ease accessibility. When you drag a panel across the Developer Workbench canvas, a set of handles appears. These handles guide the placement of the panel, and automatically dock the panel to the edge of the canvas upon which you rest the mouse.

Procedure: How to Dock Panels Around the Canvas Using Handles

1. From the panel customization toolbar, click the arrow, and then click *Floating*.
2. Drag the panel across the canvas and rest the mouse on a handle.

When you rest the mouse over a handle, a shaded area appears. This shaded area provides a preview of where the panel will dock if you release the mouse, as shown in the following image.

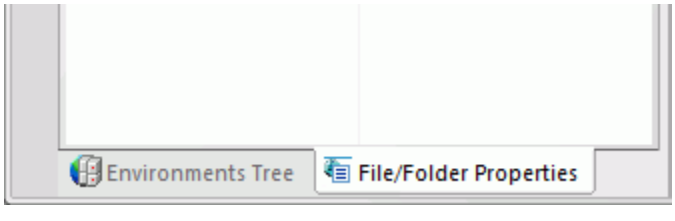


3. Release the mouse when you are satisfied with the previewed location of the panel.

   The panel automatically docks to the edge of the canvas.

Additional panels open as you develop applications throughout the product. To conserve space around the canvas, you can combine these panels. When you mouse over a docked panel, a tabbed handle option appears.

This handle option combines the panels into a single container with tabs. You can switch between the panels by clicking the appropriate tab at the bottom of the container. An example of the Environments Tree tab and File/Folder Properties tab, in a combined panel container, are shown in the following image.



Chapter 3

Creating Reports

Reports are the foundation of business intelligence and business analytics, enabling you to gain valuable insights and make improved business decisions. In Developer Workbench, you can create and edit reports using the Report canvas. The Report canvas is driven by the same InfoAssist tool that is available in the Db2 Web Query browser interface. When you create a new report, you use the Report Wizard. In report mode, you can create and style simple or complex reports, add data to the Report canvas, and style that data creating a graphical representation of the report page. This allows you to view how the report displays at run time.

This topic provides an overview of reporting, the types of reports you can create with Developer Workbench, how to launch the Report Wizard to create reports, and a sample GUI procedure, with its associated Db2 Web Query syntax, that you can use to create each type of report.

For information on the Report Wizard, see *Launching the Report Wizard* on page 75.

In this chapter:

Launching the Report Wizard

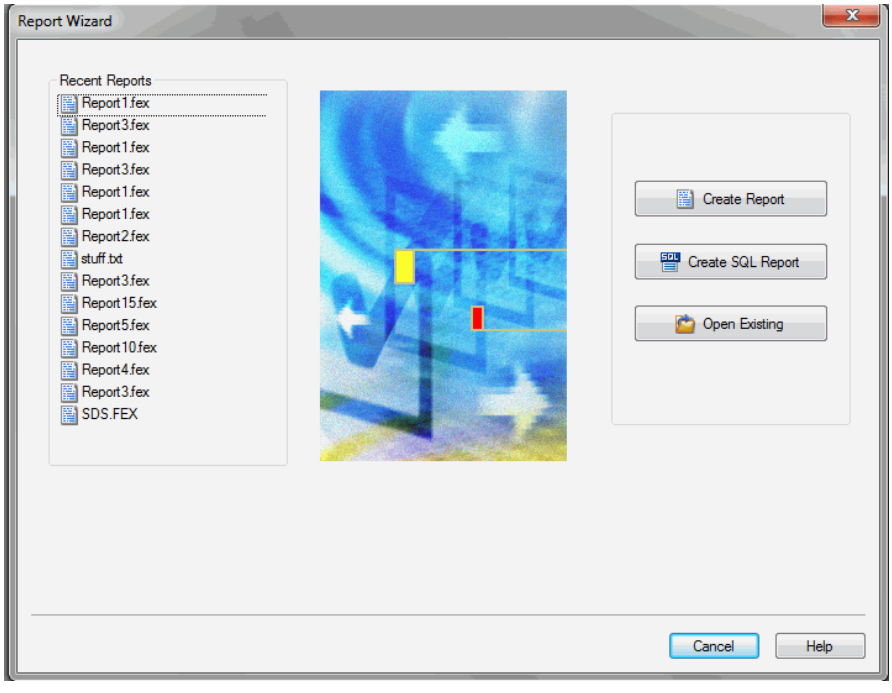
The Report Wizard allows you to create a new report procedure or SQL report procedure, and open an existing report.

To open the Report Wizard:

- On the *Home* tab, in the *Content* group, click *Report*.
- In the Environments Tree panel, right-click an application folder, point to *New*, then to *Reporting*, and click *Report*.

  Note: If you choose to create a new report using the shortcut menu in the Environments Tree panel, you will skip the first screen of the Report Wizard. You will not have to specify whether you are creating a report or SQL report. You will instead be brought to the Select Data Source window, with the location for your report already selected.

The Report Wizard opens, as shown in the following image.



From the Report Wizard, you can create a new report or SQL report procedure or open a recent procedure.

Procedure: How to Create an SQL Report From an External .sql File

After you launch the Report Wizard, you can begin to create your SQL report procedure. This enables you to browse and select external procedures that exist in the domain.

Note: If you choose to create a new SQL report using the shortcut menu in the Environments Tree panel, you will skip the first screen of the Report Wizard. You will not have to specify whether you are creating a report or SQL report. You will instead be brought to the SQL Report Wizard - Welcome window.

1. Click *Create SQL Report*.

   The Report Wizard - Choose location for the new SQL Report window opens.

2. Select a location for your SQL report and click *Next*.

   The SQL Report Wizard - Welcome window opens.

3. Click the *Included from an external '.sql' file* option.

4. Click *Next*.

   The SQL Report Wizard - Data access information window opens.

5. In the Select the SQL database engine area, select a database from the drop-down list. The drop-down list changes depending on your server configuration.

6. In the Select connection area, choose a connection from the drop-down list generated from the engine you selected.

7. Click *Next*.

   The SQL Report Wizard - Include external SQL file window opens.

8. Click *Browse* to select an external SQL file name or type the external SQL file name in the field.

   Optionally, you can run with limited records by clicking *Run SQL*. By default, the *Run with limited records* check box is selected so you can test your procedure with a read limit if the engine supports it. There is a field box next to the check box in which you can enter the number of records to be read. The default limit is 100.

9. Click *Next*.

   The SQL Report Wizard - Summary of SQL options window opens. Review your options and click *Next*, which opens the Report Wizard - SQL Data Source window.

10. Navigate to where you want to create the new procedure.

11. In the Enter a Procedure Name field, type a name for the SQL report.

    Note: Typing a procedure name is not required to create a new report. If you do not type a procedure name in the Enter a Procedure Name field, a default procedure name will be provided until you save the procedure.

12. Click *Finish*.

    The Report canvas opens. You can now add data to your report and format the data using the tabs on the ribbon.

Procedure: How to Pass SQL Commands to the RDBMS Using SQL Passthru

After you launch the Report Wizard, you can begin to create your SQL report procedure by typing SQL commands that will be passed on to the RDBMS with the SQL Passthru feature.

Note: If you choose to create a new SQL report using the shortcut menu in the Environments Tree panel, you will skip the first screen of the Report Wizard. You will not have to specify whether you are creating a report or SQL report. You will instead be brought to the *Report Wizard - Choose location for the new SQL Report* window, with the location for your report already selected.

1. Click *Create SQL Report*.

   The Report Wizard - Choose location for the new SQL Report window opens.

2. Select a location for your SQL report and click *Next*.

   The SQL Report Wizard - Welcome window opens.

3. Click *Type SQL statements in the report request*.
4. Click *Next*.

   The SQL Report Wizard - Data access information window opens.

5. In the Select the SQL database engine area, select a database from the drop-down list. The drop-down list changes depending on your server configuration.
6. In the Select the connection area, choose a connection from the drop-down list generated from the engine you selected.
7. Click *Next*.

   The SQL Report Wizard - Enter SQL statements window opens.

8. In the field box, type the SQL statements you want to pass to the RDBMS.

   Optionally, you can run with limited records by clicking *Run SQL*. By default, the *Run with limited records* check box is selected, so you can test your procedure with a read limit if the engine supports it. There is a field box next to the check box in which you can enter the number of records to be read. The default limit is 100.

9. Click *Next*.

   The SQL Report Wizard - Summary of SQL options window opens.

10. Review your options and click *Next*.

    The Report Wizard - SQL Data Source window opens.

11. Navigate to where you want to create the new procedure.
12. In the Enter a Procedure Name field, type a name for the SQL report.

    Note: Typing a procedure name is not required to create a new report. If you do not type a procedure name in the Enter a Procedure Name field, a default procedure name will be provided until you save the procedure.

13. Click *Finish*.

    The Report canvas opens. You can now add data to your report and format the data using the tabs on the ribbon.

Procedure: How to Import SQL Commands From an Existing .sql File

You can import commands from existing .sql files. This enables you to modify SQL code after importing it from an external file to the procedure being built. You can modify the request using bits of the code.

Note: If you choose to create a new SQL report using the shortcut menu in the Environments Tree panel, you will skip the first screen of the Report Wizard. You will not have to specify whether you are creating a report or SQL report. You will instead be brought to the *Report Wizard - Choose location for the new SQL Report* window, with the location for your report already selected.

1. Click *Create SQL Report*.

   The Report Wizard - Choose location for the new SQL Report window opens.

2. Select a location for your SQL report and click *Next*.

   The SQL Report Wizard - Welcome window opens.

3. Click *Import from an existing '.sql' file* and click *Next*.

   The SQL Report - Data access information window opens.

4. In the Select SQL database engine area, select a database from the drop-down list. The drop-down list changes depending on your server configuration.

5. In the Select the connection area, choose a connection from the drop-down list generated from the engine you selected.

6. Click *Next*.

   The SQL Report Wizard - Import external SQL file window opens.

7. Type the SQL file name that you want to import or click *Browse* to select it.

   Optionally, you can run with limited records by clicking *Run SQL*. By default, the *Run with limited records* check box is selected, so you can test your procedure with a read limit if the engine supports it. There is a field box next to the check box in which you can enter the number of records to be read. The default limit is 100. You can edit these options here or in the next step.

8. Click *Next*.

   The SQL Report Wizard - Enter SQL statements window opens.

9. In the field box, type the SQL statements you want to pass to the RDBMS.
10. Click *Next*.

    The SQL Report Wizard - Summary of SQL options window opens.

11. Review your options and click *Next*.

    The Report Wizard - SQL Data Source window opens.

12. Navigate to where you want to create the new procedure.
13. In the Enter a Procedure Name field, type a name for the SQL report.

    Note: Typing a procedure name is not required to create a new report. If you do not type a procedure name in the Enter a Procedure Name field, a default procedure name will be provided until you save the procedure.

14. Click *Finish*.

    The Report canvas opens. You can now add data to your report and format the data using the tabs on the ribbon.

Chapter 4

Creating Charts and Visualizations

A chart often conveys meaning more clearly and effectively than data displayed in tabular form. A chart enables you to visually communicate quantitative information. On a chart, you can give data a shape and form, and reveal patterns and relationships among many data values. A chart can highlight anomalies that require further investigation.

It is important that you choose a chart that is appropriate for your data. Developer Workbench provides a complete chart library of both basic and advanced charts. You can choose from a wide variety of charts to best represent the data that you want to display.

In Developer Workbench, you can create and edit charts using the Chart canvas. The Chart canvas is driven by the same InfoAssist tool that is available in the Db2 Web Query browser interface. When you create a new chart, you use the Chart Wizard.

This topic provides an overview of the types of charts you can create with Db2 Web Query Developer Workbench and how to launch the Chart Wizard to create charts.

For more information on creating visualizations, click the Help icon  , located in the upper-right corner of the canvas.

In this chapter:

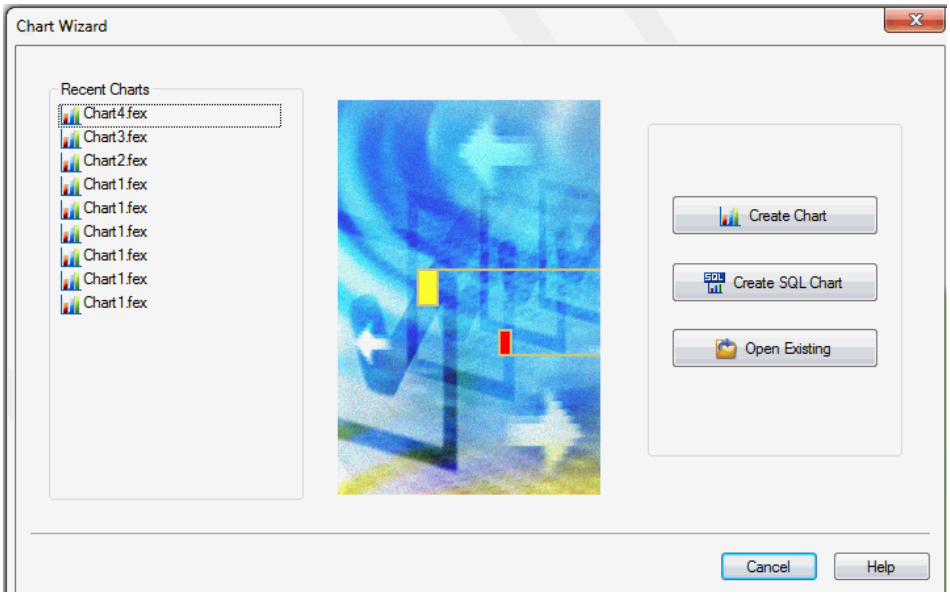
Launching the Chart Wizard

To open the Chart Wizard:

- On the *Home* tab, in the *Content* group, click *Chart*. Use this option to create a chart using the Chart canvas.
- In the Environments Tree panel, right-click an application folder, point to *New*, point to *InfoAssist*, and click *Chart* or *SQL Chart* to create a chart using the chart canvas.

  Note: If you choose to create a new chart using the shortcut menu in the Environments Tree panel, you will skip the first screen of the Chart Wizard. You will not have to specify whether you are creating a chart or SQL chart, since you already clicked one of these options. You will instead be brought to the Select Data Source window, with the location for your chart already selected.

The Chart Wizard opens, as shown in the following image.



From the Chart Wizard, you can create a new chart or SQL chart procedure or open a recent procedure.

Procedure: How to Create an SQL Chart From an External .sql File

After you launch the Chart Wizard, you can begin to create your SQL chart procedure.

Note: If you choose to create a new SQL chart using the shortcut menu in the Environments Tree panel, you will skip the first screen of the Chart Wizard. You will not have to specify whether you are creating a chart or SQL chart. You will instead be brought to the SQL Chart Wizard - Welcome window, with the location for your chart already selected.

1. Select *Create SQL Chart*.

   The Chart Wizard - Choose location for the new chart window opens.

2. Navigate to where you want to create the new chart and click *Next*.

   The SQL Chart Wizard - Welcome window opens.

3. Click the *Included from an external '.sql' file* option. This enables you to browse and select external procedures that exist in any folder in the Data Servers or Domains areas.

4. Click *Next*.

   The SQL Chart Wizard - Data access information window opens.

5. In the Select the SQL database engine area, select a database from the drop-down list. The drop-down list changes depending on your Reporting Server configuration.
6. In the Select the connection area, choose a connection from the drop-down list generated from the engine you selected.
7. Click *Next*.

   The SQL Chart Wizard - Include external SQL file window opens.
8. Click *Browse* to select an external SQL file name or type the external SQL file name in the field.

   Optionally, you can run with limited records by clicking *Run SQL*. By default, the *Run with limited records* check box is selected so you can test your procedure with a read limit if the engine supports it. There is a field box next to the check box in which you can enter the number of records to be read. 100 is the default limit.
9. Click *Next*.

   The SQL Chart Wizard - Summary of SQL options window opens.
10. Review your options and click *Finish*.

    The Chart Wizard - SQL Data Source window opens.
11. Navigate to where you want to create the new procedure.
12. In the Enter a Procedure Name field, type a name for the SQL chart.

    Note: Typing a procedure name is not required to create a new chart. If you do not type a procedure name in the Enter a Procedure Name field, a default procedure name will be provided until you save the procedure.
13. Click *Finish*.

    The Chart canvas and the Procedure View panel open. You can now add data to your chart and format the data using the tabs in the ribbon.

Procedure: How to Pass SQL Commands to the RDBMS Using SQL Passthru

After you launch the Chart Wizard, you can begin to create your SQL chart procedure.

Note: If you choose to create a new SQL chart using the shortcut menu in the Environments Tree panel, you will skip the first screen of the Chart Wizard. You will not have to specify whether you are creating a chart or SQL chart. You will instead be brought to the *Chart Wizard - Choose location for the new SQL Chart* window, with the location for your chart already selected.

1. Click *Create SQL Chart*.

   The Chart Wizard - Choose location for the new SQL Chart window opens.

2. Navigate to where you want to create the SQL chart and click *Next*.

   The SQL Chart Wizard - Welcome window opens.

3. Click *Type SQL statements in the chart request*. This enables you to type SQL commands that will be passed on to the RDBMS with the SQL Passthru feature.

4. Click *Next*.

   The SQL Chart Wizard - Data access information window opens.

5. In the Select the SQL database engine area, select a database from the drop-down list. The drop-down list changes depending on your Reporting Server configuration.

6. In the Select the connection area, choose a connection from the drop-down list generated from the engine you selected.

7. Click *Next*.

   The SQL Chart Wizard - Enter the SQL statements window opens.

8. In the field box, type the SQL statements you want to pass to the RDBMS.

   Optionally, you can run with limited records by clicking *Run SQL*. By default, the *Run with limited records* check box is selected so you can test your procedure with a read limit if the engine supports it. There is a field box next to the check box in which you can enter the number of records to be read. The default limit is 100.

9. Click *Next*.

   The SQL Chart Wizard - Summary of SQL options window opens.

10. Review your options and click *Finish*.

    The Chart Wizard - SQL Data Source window opens.

11. Navigate to where you want to create the new procedure.

12. In the Enter a Procedure Name field, type a name for the SQL chart.

    Note: Typing a procedure name is not required to create a new chart. If you do not type a procedure name in the Enter a Procedure Name field, a default procedure name will be provided until you save the procedure.

13. Click *Finish*.

    The Chart canvas and the Procedure View panel open. You can now add data to your chart and format the data using the tabs in the ribbon.

Procedure: How to Import SQL Commands From an Existing .sql File

You can import commands from existing .sql files.

Note: If you choose to create a new SQL chart using the shortcut menu in the Environments Tree panel, you will skip the first screen of the Chart Wizard. You will not have to specify whether you are creating a chart or SQL chart. You will instead be brought to the *Chart Wizard - Choose location for the new chart* window, with the location for your chart already selected.

1. Open the Chart Wizard and click *Create SQL Chart*.

   The Chart Wizard - Choose location for the new chart window opens.

2. Navigate to where you want to create the new chart and click *Next*.

   The SQL Chart - Welcome window opens.

3. Click *Import from an existing '.sql' file* and click *Next*. This enables you to modify SQL code after importing it from an external file to the procedure being built. You can modify the request using bits of the code.

   The SQL Chart - Data access information window opens.

4. In the Select the SQL database engine area, select a database from the drop-down list. The drop-down list changes depending on your Reporting Server configuration.

5. In the Select the connection area, choose a connection from the drop-down list generated from the engine you selected.

6. Click *Next*.

   The SQL Chart Wizard - Import external SQL file window opens.

7. Type the SQL file name that you want to import or click *Browse* to select it.

   Optionally, you can run with limited records by clicking *Run SQL*. By default, the *Run with limited records* check box is selected so you can test your procedure with a read limit if the engine supports it. There is a field box next to the check box in which you can enter the number of records to be read. The default limit is 100. You can edit these options here or in the next step.

8. Click *Next*.

   The SQL Chart Wizard - Enter SQL statements window opens.

9. In the field box, type the SQL statements you want to pass to the RDBMS.

10. Click *Next*.

    The SQL Chart Wizard - Summary of SQL options window opens.

11. Review your options and click *Next*.

    The Chart Wizard - SQL Data Source window opens.

12. Navigate to where you want to create the new procedure.

13. In the Enter a Procedure Name field, type a name for the SQL chart.

    Note: Typing a procedure name is not required to create a new chart. If you do not type a procedure name in the Enter a Procedure Name field, a default procedure name will be provided until you save the procedure.

14. Click *Finish*.

    The Chart canvas and the Procedure View panel open. You can now add data to your chart and format the data using the tabs in the ribbon.

Chapter 5

Creating HTML Pages

In Developer Workbench, you can create and edit HTML pages using the HTML canvas. When you create a new HTML page, you use the HTML/Document Wizard.

This topic describes how to launch the HTML/Document Wizard to create HTML pages. It also identifies and explains the tabs and panels that are available when you are developing HTML pages in the HTML canvas.

Note: When using the HTML canvas in Developer Workbench, it is recommended to set the Scale and Layout display settings on the PC to 100% so that Developer Workbench renders correctly.

In this chapter:

Launching the HTML/Document Wizard

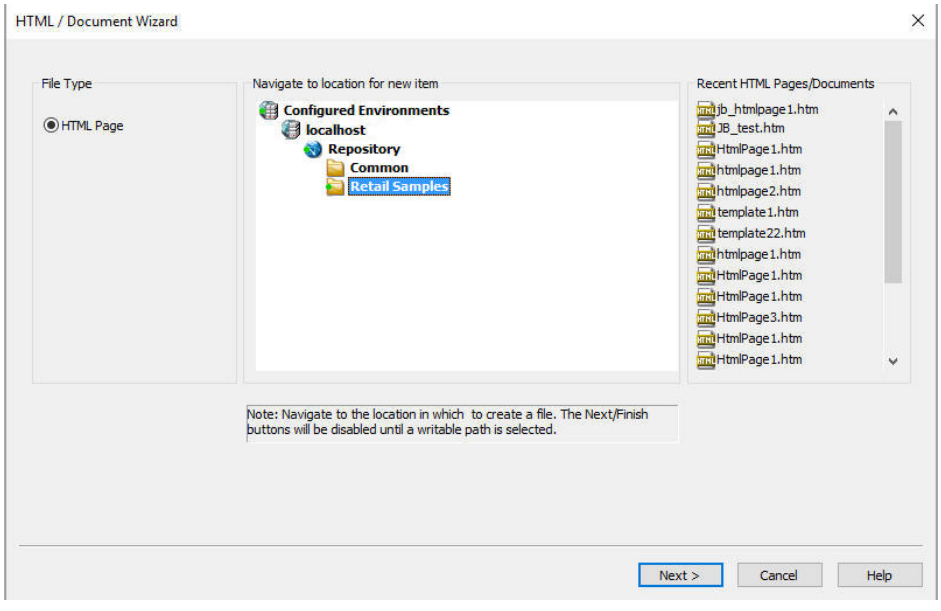
The HTML/Document Wizard allows you to create a new HTML Page, or open a recent procedure.

To open the HTML/Document Wizard:

- On the *Home* tab, in the *Content* group, click *HTML/Document*.
- In the Environments Tree panel, right-click an application folder, point to *New*, and click *HTML File*.

Note: If you choose to create a new HTML page using the shortcut menu in the Environments Tree panel, your screen will not include a Navigator panel since the location for your HTML page is already selected.

The HTML/Document Wizard opens, as shown in the following image.



From the HTML/Document Wizard, you can open a recent HTML Page/Document or create a new one.

Procedure: How to Create an HTML Page Using the HTML/Document Wizard

After you have launched the HTML/Document Wizard, you can begin to create your HTML Page.

1. In the File Type area, select *HTML Page*.
2. Navigate to where you want to create the new procedure or select a recent procedure and click *Next*.

   Note:

   - The Next button is available after you select a valid path.
   - If you used the shortcut menu in the Environments Tree panel to create your HTML page, a location is already selected.

3. Optionally, you can add themes and settings to your HTML page.
4. Click *Finish*.

The HTML canvas, File/Folder Properties panel, Properties panel, Settings panel, Requests & Data sources panel, and Tasks & Animations panel open. You can now add data to your HTML page and format the data using the tabs in the ribbon.

5. To save your HTML page, click the *Save* button on the Quick Access Toolbar or select *Save* from the Application menu.
6. To run your HTML page, click *Run* on the Quick Access Toolbar or click *Run* from the Application menu.

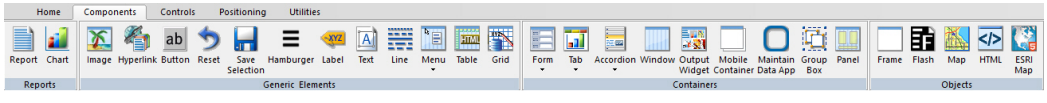
Accessing HTML Page Components

The following topics describe the tabs and panels that are available when you are developing HTML pages in the HTML canvas.

Inserting Components in an HTML Page Using the Components Tab

You can add a variety of components to an HTML page found in the command groups in the Components tab. For example, suppose you want to add a button to your page that a user can click to refresh the data. The button is a component. You select it from the Generic Elements command group in the Components tab, then add it to your page as desired.

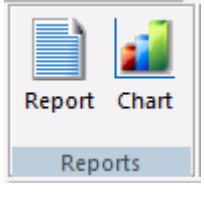
The Components tab contains the Reports, Generic Elements, Containers, and Objects groups, as shown in the following image.



When you click a command from a ribbon, your mouse pointer turns into a cross-hair to let you draw the object in the HTML canvas. If you click a command in error, press the Escape key to return your mouse to a pointer.

Inserting a Report or Chart in an HTML Page Using the Reports Group

From the Reports group, you can add a report or chart to your HTML page. The Reports group is shown in the following image.



The commands in the Reports group are:

Report

Inserts a report object. You can add reports to the HTML canvas that will display when you run the layout. Reports are added by referencing existing reports in the repository.

You can also include parameters in a report whose values can be assigned with controls that are added with the HTML canvas.

Chart

Inserts a chart object. You can add charts to the HTML canvas that will display when you run the layout. Charts are added by referencing existing charts in the repository.

Procedure: How to Add a New Report or Chart to an HTML Page

1. Insert a report or chart object by doing one of the following:

   - ❏ On the *Components* tab, in the *Reports* group, click the *Report* or *Chart*.
   - ❏ Right-click in the layout and select *New Report* or *New Chart* from the shortcut menu.

   The pointer changes into a crosshair.

2. Drag the crosshair to create a report or chart object and adjust it to the size that you want.

   A report or chart object is created in the layout and assigned the name report*(n)* or chart*(n)*, where *n* is a number. The object will appear in gray and white to indicate that the placeholder does not have a report or chart associated with it. Once a report or chart is associated with the object, the object displays the contents of the report or chart if live or simulated data is active (live data is the default) or a colored placeholder if preview is off in the HTML Page tab, located in the Developer Workbench Options dialog box.

3. Create a report or chart by doing one of the following:

   - ❏ Double-click the placeholder.

     or

   - ❏ Right-click the placeholder and select *New Report* for a report, or *New Chart* for a chart.

   The Open File dialog box appears.

4. Select the Master File you want to use and click *OK*.

   The Report canvas opens for reports and the Chart canvas opens for charts.

5. Optionally, after creating the report or chart, you can change its properties by adjusting the properties displayed in the Properties panel.

Procedure: How to Add an Existing Report or Chart to a Layout

1. Insert a report or chart object by doing one of the following:

   - ❏ On the *Components* tab, in the *Reports* group, click *Report* or *Chart*.
   - ❏ Right-click in the layout and select *New Report* or *New Chart* from the shortcut menu.

   The pointer changes into a crosshair. Drag the crosshair to create a report or chart object and adjust it to the size you want.

   A report or chart object is created in the layout. The object will appear in gray and white to indicate that the placeholder does not have a report or chart associated with it. Once a report or chart is associated with the object, the object displays the contents of the report or chart if live or simulated data is active (live data is the default) or a colored placeholder if preview is off in the HTML Page tab, located in the Developer Workbench Options dialog box.

2. Right-click the report or chart.

   - ❏ Select *Reference existing procedure*.

   The Open File dialog box appears.

3. Select the procedure you want to add to the layout.
4. Click *OK*.

   The report or chart object appears in the Design view of the HTML canvas.

Adding Basic Elements to an HTML Page Using the Generic Elements Group

You can add a variety of basic elements. The Generic Elements group is shown in the following image.

The commands in the Generic Elements group are:

Image

Inserts an image. You can add an image to the layout. This is useful for including graphics, such as a company logo.

You can insert an image into your report layout and add a hyperlink. After you run your report and click the image, you can launch a URL or run a report the same way by clicking a hyperlink or push button.

Note: When inserting images, images must be referenced from a specific directory location.

Hyperlink

Inserts a hyperlink.

Button

Inserts a button. A push button enables you to execute a report or link to a URL or HTML file. This behavior is similar to a hyperlink.

Note: You can either double click to change the text or can use the Text property in the Properties panel.

Reset

Inserts a reset button. A reset button enables you to revert the entire page back to its initial settings.

Save Selection

Inserts a Save Selection button. 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 that is saved in the same folder as the source file, by default.

The static Save Selection files are not editable in the HTML canvas or in the text editor. To verify that a file is a Save Selection file:

1. Right-click the file in the Configured Environments tree and select *Properties* from the shortcut menu.

   The File/Folder Properties panel opens.

2. The value for the Properties attribute is *tool=saveparam*. This indicates that the file is not editable in the HTML canvas or in the text editor.

Hamburger

Inserts a hamburger control, which can be used to hide or display a form or control panel. A control panel is a panel component that has the Control panel property enabled.

To use a hamburger control, add it to the page, select it, and open the Properties panel. In the Hamburger specific section, select a form or control panel on the page from the Navigation and controls panel drop-down menu. You can also use the Properties panel to set the animation used when the hamburger control is clicked, and automatically hide the form or control panel when a request is executed on the page.

When you run the HTML page, the selected form or control panel does not appear until you click the hamburger control. You can hide it by clicking the hamburger control again.

Label

Inserts a label. The label component lets you create and name a label, and link it to a control. To link a label to a control, you can select a value from the HtmlFor property drop-down list in the Properties panel, or you can select the *Bind label to* option from the shortcut menu, when the label and control are both selected.

Text

Inserts a text box. You can add text to the layout. This is useful for including headings for your webpage, or adding directions or explanations for your report or chart.

Line

Inserts a line. You can add a horizontal or vertical line to the layout. This is useful for distinguishing between sections of your launch or display page.

Menu

Inserts a menu. You can add a horizontal or vertical menu to the layout.

Table

Inserts a table.

Grid

Inserts a grid.

Procedure: How to Toggle the Display of a Form Using a Hamburger Control

A hamburger control allows you to show and hide a form, saving space on an HTML page.

1. Create an HTML page.
2. Add a chart or report component to the page, and reference a procedure that includes at least one parameter.

   The New Parameters dialog box opens.
3. Click *OK* to create a form on the page using the default options.
4. On the *Components* tab, in the *Generic Elements* group, click *Hamburger* and draw a hamburger control onto the page.

   Note: Forms appear below and to the right of the hamburger control when it is clicked. Make sure to leave enough space between the hamburger control and the right side of the page to display the entire form.

5. Select the hamburger control and open the Properties panel.
6. Link the hamburger control to a form. In the Hamburger specific section of the Properties panel, click the drop-down menu for the Navigation and controls panel property, and select the form that you want to access using the hamburger control.

   You can also link a hamburger control to a control panel. To create a control panel, on the *Components* tab, in the *Containers* group, click *Panel*, and draw a panel onto the page. Select it and, in the Properties panel, set the Control panel property to *Yes*.
7. Optionally, set the Animation and Auto-hide panel properties.

   The Animation property allows you to set the animation that occurs when the form appears.

   When the Auto-hide panel property is enabled, clicking the submit button in the form hides the panel. If Auto-hide panel is not enabled, you can hide the form again by clicking the hamburger control.
8. Run the HTML page. The form is hidden, by default, and it appears when you click the hamburger control.

Procedure: How to Format Text in an HTML Page

You may apply various formatting and style options to words and individual text characters within the text object.

Note: Any formatting and styling that you have applied to individual text strings within the text object will remain unchanged. Changes made to the entire text object are only applied to part of the text string that has not been formatted.

1. On the *Components* tab, in the *Generic Elements* group, click *Text*.

   The pointer changes to a crosshair.
2. Drag your pointer across the canvas to create a text object.

   A text object with the default text, *Enter text*, is created.
3. Select the text that you wish to format:

   - To format the entire text object, single-click the text object in the layout.
   - To format an individual word or text character, highlight part of the text within the text object.
4. Right-click the text, point to *Style*, and then click *Font*.

   The Style Composer dialog box opens with Font selected automatically.

   Note: You can also access font formatting options in the Properties panel.

5. Select from the formatting options available. You can change the type, style, color, size, and effect of the font.

   Note: When you enter a *Specific* font size, the unit defaults to px (pixels).

6. Click *OK* to close the Style Composer dialog box.

The formatting options that you selected are applied to the text.

Procedure: How to Insert a Bulleted List or Numbered List Into a Text Element

To insert a bulleted list or numbered list into a text object:

1. On the *Components* tab, in the *Generic Elements* group, click *Text*.

   The pointer changes to a crosshair.

2. Drag your pointer across the canvas to create a text object.

   A text object with the default text, *Enter text*, is created.

3. Add multiple lines of text to the text object.
4. Highlight and right-click the text that you want to include in the list.
5. In the shortcut menu, select one of the following list options:

   - **Bullets**
     - Disc
     - Circle
     - Square
     - None
   - **Numbering**
     - Numbers
     - Lowercase Letters
     - Uppercase Letters
     - Small Roman numerals
     - Large Roman numerals
     - None

Note:

- Alternatively, you can select a bullet type before typing text to begin the list. Pressing enter will begin the next item in the list on a separate line.
- To change the bullet or number list type of an existing list, place your pointer on the list level you want to change and reselect a bullet or number list type. Selecting *None* will remove the bullets or numbers for that level and move any nested lists up one level. In order to switch between bullets and numbers, you must first remove the current list option by selecting *None* and then applying the list option you want.

Procedure: How to Insert Nested Lists Into a Text Element

To insert a nested list into the text object:

1. On the *Components* tab, in the *Generic Elements* group, click *Text*.

   The pointer changes to a crosshair.

2. Drag your pointer across the canvas to create a text object.

   A text object with the default text, *Enter text*, is created.

3. Add a list to the text object. For more information, see *How to Insert a Bulleted List or Numbered List Into a Text Element* on page 95.
4. Place your pointer after a list item.
5. Right-click, point to *Nested List* and then select a bulleted or numbered list option.

   A list is started within the current list, allowing you to enter text on that list level.

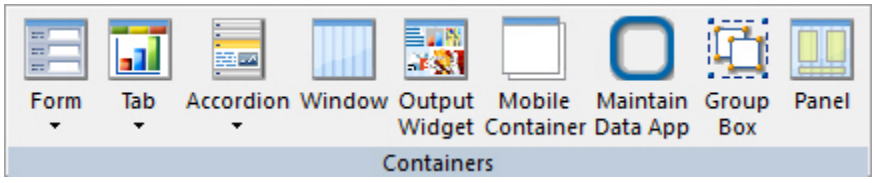
   Note: Pressing Tab while your pointer is on the same line as a list item will move that item one level down, resulting in a nested list. The bullet or number type selected is the next list type in the right-click shortcut menu. For example, if you have a bulleted list that uses the disc bullet type, pressing Tab to move an item down one level will cause that nested list to have the circle bullet type.

   You can continue to nest lists within other lists by using the same steps shown above.

   Note: You cannot skip a list level. For example, in order to insert a nested bulleted list or nested numbered list on a lower level, there must be a list one level up from it.

Adding Containers to an HTML Page Using the Containers Group

You can add specific containers that group objects together on an HTML page. The Containers group is shown in the following image.



The commands in the Containers group are:

Form

Inserts either a multi-layer form or single-layer form.

Tab

Inserts a tab control. Tab controls enable you to create multiple pages in one HTML form and present a better display for viewing secondary information. You can select *Top*, *Bottom*, *Left*, or *Right*. This means you can choose to add a tab control that displays tabs at the top, bottom, left, or right on the control.

You can drag and resize this object, at run time, if the Enable dragging and Enable resizing properties, in the Properties panel, are set to *Yes*.

To change the header text, you can either double click and change the header text or enter a value in the Selected Page: Title property.

When a tab control object is added to the layout, each tab control consists of:

- A tab item.

  A tab item is the tab label. You may edit the name of the tab item, style the tab item, and add multiple tab items. Each tab item is associated with a tab body.

- A tab body.

  A tab body is the tab page where you associate your components, such as report and graph objects, images, and lines.

The Tab control can be displayed as a full screen or part of an HTML page.

Accordion

Inserts an Accordion styled box. You can select *Horizontal* or *Vertical*. You can drag and resize this object, at run time, if the Enable dragging and Enable resizing properties, in the Properties panel, are set to *Yes*.

To change the header text, you can either double click and change the header text or enter a value in the Selected Page: Title property.

Window

Inserts a window. You can drag and resize this object, at run time, if the Enable dragging and Enable resizing properties, in the Properties panel, are set to *Yes*.

To change the header text, you can either double click and change the header text or enter a value in the Selected Page: Title property.

Output Widget

Inserts an output widget. The output widget container includes buttons in the widget title bar that let you display or hide filter controls and that let you maximize to display full-screen, and minimize to return it to its original size.

Mobile Container

Inserts a mobile container containing an output widget. You can right-click the mobile container to add multiple pages, and then create tasks using the Go to page action to navigate between them. If you right-click a page in a mobile container and click *Selected page as prototype*, any subsequently added pages are created as clones of the prototype page.

The output widgets on mobile container pages include hamburger controls, allowing you to access collapsible forms and control panels to interact with the page at run time.

You can enable responsive design on an HTML page that includes a mobile container so that the mobile container and the output widgets in it can resize to comfortably fit on mobile devices.

Maintain Data App

Inserts a Maintain Data app window. This component is not applicable to Db2 Web Query, as it requires the Maintain feature.

Group Box

Inserts a group box. A group box can be used to create a border around a group of objects, for example, forms or reports and charts.

Panel

Inserts a panel to group objects together. The panel is invisible at run time.

Procedure: How to Use a Form Object to Create a Pop-Up Dialog Box

You can use a form object to create a pop-up dialog box.

1. Create a form object.
2. Create a button.
3. Select the form.
4. In the Properties panel, change the *Display container as* property to *As popup dialog*.
5. Create a new task in the Tasks section of the Tasks & Animations panel.
6. Use the button you create as the trigger.
7. Make the button object toggle the visibility of the form you created.
8. Run the page and click the button to display the form as a dialog box.

Procedure: How to Create a Pop-Up Dialog Box or Pop-Up Window From a Form Control

1. Add a button object to your HTML page.
2. Add a form control to your HTML page.
3. Delete the run and back buttons created with the form control.
4. While the form control is selected, in the Properties panel, change the Display container as property to either *As popup dialog* or *As popup window*.

   As popup dialog causes the form to display as a pop-up dialog box. When the dialog box is displayed, you must click *Close* to return to your HTML page.

   As popup window causes the form to display as a pop-up window. When the window is displayed, you can resize and move the window around your screen.
5. Create a new task in the Tasks section of the Tasks and Animations panel that uses the button object to toggle the visibility of the form you created.
6. Run your HTML page.

   When you click the button you added to the HTML page, the form will be displayed as either a pop-up dialog box or a pop-up window, depending on what option you selected.

Reordering Tabs, Accordions, and Windows

You can reorder tabs, accordions, and windows by dragging the page that you want to appear first onto the page that you want it to display before. For example, if you have 3 tabs, Tab1, Tab2, and Tab3, and you drag Tab3 onto Tab1, Tab3 will now display ahead of Tab1. The order of the tabs will then be Tab3, Tab1, Tab2. You can reorder windows when they are in tile view.

Procedure: How to Navigate Between Pages in a Mobile Container

You can add multiple pages to a mobile container on an HTML page to provide different sets of content with expandable forms and control panels. You can add a task using the Go to page action to allow users to move between pages.

1. Create a new HTML page.
2. Open the Properties panel and, with the *DOCUMENT* object selected, set the Enable responsive property, in the Responsive design section, to *Yes*.

   Using responsive design will make the content on the page easier to use on mobile devices.
3. On the *Components* tab, in the *Containers* group, click *Mobile Container* and draw a mobile container on the page.

   The mobile container contains one page, by default.
4. Click on the output widget in the first page of the mobile container, and open the Properties panel. Enter a new title for the page using the Widget Title property.

   Since the Go to page action differentiates between pages by the widget title, using different titles for each one helps to ensure you select the correct page to navigate to.
5. Add content to the first page inside the output widget.
6. Create a control panel that will include a button to navigate to the second page. The control panel can also include filter controls to interact with the content on the page.
   a. On the *Components* tab, in the *Containers* group, click *Panel*, and create a panel on the page.
   b. On the *Components* tab, in the *Generic Elements* group, click *Button* and draw a button inside the panel.

      Optionally, double-click the button to change the text in it.
   c. Select the panel containing the button and open the Properties panel. Set the Control panel property to *Yes*.

      This turns the panel into a control panel that can be accessed using the hamburger control in the mobile container output widget.
7. Select the mobile container output widget and open the Properties panel. Select the control panel that you created previously from the menu next to the Navigation and controls property.
8. Right-click the mobile container, outside of the output widget, and click *Add Page*.

   A second page with an output widget is created in the mobile container.
9. Repeat steps 4 through 7 to create the second page. Follow these steps again for any additional pages that you want to add.

10. Create two new tasks using the Go to page action to navigate between the first and second page.
    a. Open the Tasks & Animations panel.
    b. Click the *New* button to create a new task.
    c. Leave the Trigger Type as *Click*.
    d. For the Trigger Identifier, select the first button that you created, which is in the collapsible panel on the first page of the mobile container.
    e. Click the arrow next to the *Requests selections* button, point to *Go to page*, point to the name of the mobile container, and select the name of the second page in the mobile container.

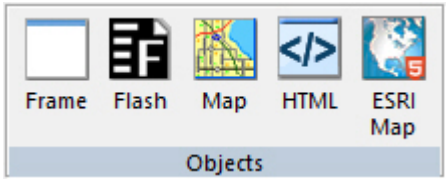
       This will allow you to access the second page from a button accessible from the first page.
    f. Click *New* to create a second task.
    g. Leave the Trigger Type as *Click*.
    h. For the Trigger Identifier in the second task, select the button that will be accessible from the second page of the mobile container.
    i. Click the arrow next to the *Requests selections* button, point to *Go to page*, point to the name of the mobile container, and select the name of the first page in the mobile container.

       This will allow you to return to the first page using a button accessible on the second page.
11. Save the HTML page and access it from a mobile device.

    When you tap the hamburger control, a panel with a button opens. Tap the button to access the other page created using the mobile container.

Adding Objects to an HTML Page Using the Objects Group

You can add objects other than reports, charts, elements, or containers to customize your HTML page. The Objects group is shown in the following image.



The commands in the Objects group are:

Frame

Inserts a frame object. You can use a frame to embed additional web sources or run reports. You can also use a frame as the output location or target for a drill-down report. You can also use a frame to run a table of contents report, an OLAP report, a PDF report, or an Excel® report.

Flash

Inserts a Flash component. You can add SWF files that are Adobe® Flash Player compatible to accompany reports or graphs on an HTML page.

Note: When inserting Flash animations, only files that are 1 MB or smaller can be run using the HTML canvas.

Map

Inserts a map object. You can add a Google™ or ESRI map to your HTML page. Maps are services offering powerful, user-friendly mapping technology that can be customized to show points on a map with drill-down capabilities. You can customize the map properties and bind them to a report or chart. This feature is not applicable to Db2 Web Query. Inserting a map can be done by inserting an existing Chart that has been formatted as an ESRI map.

HTML

Inserts an HTML object. Developers can use the Settings panel to type a valid snippet of HTML code for the HTML object, such as HTML code to browse for a file.

ESRI Map

Inserts an ESRI map object. You can use the Settings panel to configure the properties and components. This feature is not applicable to Db2 Web Query. Inserting a map can be done by inserting an existing Chart that has been formatted as an ESRI map.

Changing Load Order With HTML Object Manipulation

In an HTML page, objects load in the order in which they were added to the canvas. A developer using raw HTML has the option of editing the source code to change the load order. Such edits leverage the Document Object Model (DOM) for HTML and XML documents.

In Developer Workbench, however, the source code cannot be edited. Therefore, the HTML object manipulation menu is provided to allow you to change the load order, if necessary.

To change the load order, select multiple objects or components in the HTML canvas, then right-click and select HTML object manipulation. The following menu options are available:

- **Append selected.** Available when the last multi-selected object is a container. Use this option to move one or more multi-selected objects into the container.
- **Swap selected.** To swap positions for two multi-selected objects.
- **Delete contains only.** To delete containers only.
- **Merge containers.** To merge containers.

Accessibility Support for Displaying Objects in the Order of the Document Object Model

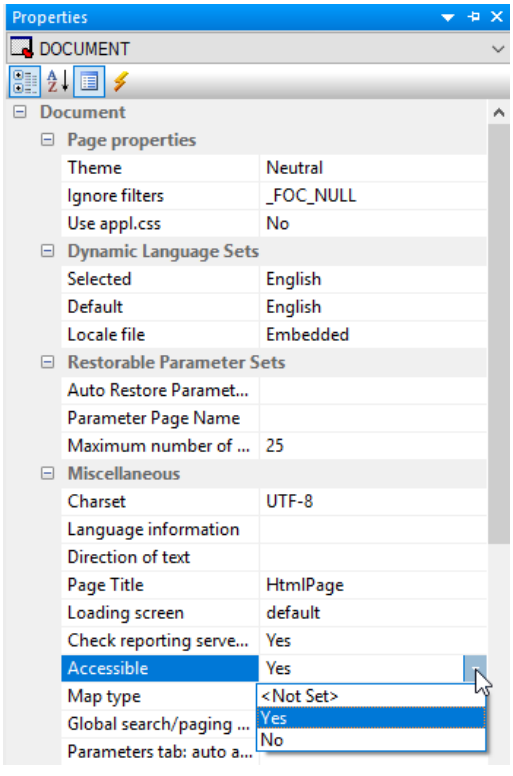
You can add elements and objects independently of screen reading order and then change the order later on during the development process.

Using the Accessible property, available on the Properties panel for a document, you can indicate that the Document Object Model (DOM) should be rewritten in the order of objects on the page, left to right, top to bottom. In addition, all tab index values should be set to the value -1.

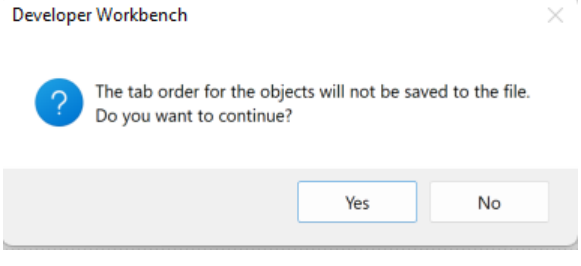
Procedure: How to Display Objects in the Order of the DOM

1. Create a new HTML page with multiple controls on the page.

2. On the Properties panel for the document, select *Yes* from the Accessible drop-down menu, as shown in the following image.



The following tab order warning message appears, as shown in the following image.



Note: The Warning message displays only once, when the Accessibility property is set to *Yes*.

Selecting *Yes* will write the tab order of objects on the page, left to right, top to bottom. Selecting *No* will keep the tab order in the page.

Inserting Controls in an HTML Page Using the Controls Tab

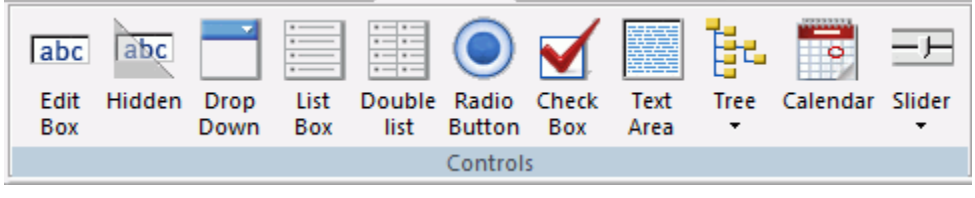
You can use the Controls tab and group to insert controls in an HTML page. Controls enable you to prompt users for a parameter value. When you create a parameter as part of a report or graph, the HTML canvas automatically adds a control, Submit button, and Reset button for the parameter to your layout, and the parameter appears on the Parameters tab. You can also add an input control and bind it to a parameter.

Controls, with the exception of a text box which does not supply a list of possible values, can supply values with a dynamic or static list of values:

- A dynamic list retrieves values from a specified data source when the request is run.
- A static list consists of a list of values you supply. These values do not change unless you change them.

An Analytic Document control lists Analytic Document values that mimic In-Document Analytic menu items. The Analytic Document controls cannot be associated to any parameters in the layout. This type of control can only be associated with an Analytic Document in the layout.

The properties of a control, as well as the parameters associated with each control, can be controlled with the Properties tab of the Properties window, and with the Parameters tab. The Controls tab is shown in the following image.



The controls are:

Edit Box

Inserts a text box. A text box enables you to enter a value in an entry field. You can specify a list of static or dynamic accepted values in Settings panel and, at run time, when you type the first letter of a value into the text box, that value will be listed as an autocomplete suggestion.

Change the Type property for the edit box control to *File* to use the edit box to browse for files from your system, to *Password* to automatically mask the text for password entry, or to *Hidden* to make the control invisible.

Hidden

Inserts a hidden control. A hidden input control allows parameter values to be used in a control without the user seeing them. When a hidden control is used, the current input control assigned to the parameter will not be visible. The value of the parameter can be entered in the Properties panel of the Parameters tab, or supplied through chaining.

Drop Down

Inserts a drop-down list. A drop-down list enables you to select single or multiple values from a list of supplied values. In order to provide multiple values, the procedure must be set up to accept multiple values.

- A single-select drop-down list enables you to select only one value for each time a request is run.
- A multiselect drop-down list enables you to select multiple values using the check boxes adjacent to the values.

You can use a dynamic or static list of values for the drop-down list.

Note: The height of a drop down control is based on the font size of the text within it, and not the height property of the control object.

List Box

Inserts a list. This enables you to select single or multiple values at one time:

- A single-select list enables you to select only one value for each time a request is run.
- A multiselect list enables you to select multiple values by using the Ctrl key while selecting values.

In order to provide multiple values, the procedure must be set up to accept multiple values.

List box values can be dynamic or static.

Double list

Inserts a double list. Displays multiselect values. This enables you to view a list of the available values and add or remove them from one list to another. At run time, a report is generated based on the values that are added.

Radio Button

Inserts a radio button. Radio buttons enable you to select a single value from a list of supplied values. Radio button values can be static or dynamic.

Use the Type property to control the appearance of the radio buttons. From the drop-down list box in the Type field, select *Standard* to display standard radio buttons.

From the drop-down list box in the Type field, select *Push button* to display push buttons.

Check Box

Inserts a check box. Check boxes enable you to select a single value from a list of supplied values. Note that if there are multiple check box input controls that are grouped together, you may select the Multiple properties for each control. Multiple ensures that you can select a single value from each check box control. Check box list values can be dynamic or static.

Use the Type property to control the appearance of the check boxes. From the drop-down list box in the Type field, select *Standard* to display standard check boxes.

From the drop-down list box in the Type field, select *Push button* to display push buttons.

Text area

Inserts a text area. A text area is a single-select control that enables you to enter multiple lines of text that can be assigned to a single variable. The behavior is similar to a text box, but you are not restricted to entering just one line of text. For example, if you want to assign a paragraph (multiple lines of text) to a variable that can be referenced by a procedure, you can add the paragraph to a text area from the Properties panel on the Parameters tab.

If you want to display text in the control:

- When hard wrap is enabled, at run time, a line break is shown at the point where it was specified.
- For soft wrap, the line should wrap automatically depending on the size of the text area control.

If you want to submit values in the control:

- For single-select text area controls, only the first value up to the end of the line will be submitted.
- For multiselect text area controls:
  - All of the values separated by OR or AND will be submitted. For example, "ENGLAND" OR "FRANCE".
  - To indicate one value with a comma, type ENGLAND,FRANCE.
  - To indicate two values with a comma, type "ENGLAND","FRANCE".

- To indicate two values, type the values on separate lines. For example:

  ENGLAND

  FRANCE

Tree

Inserts either a single-source tree control or multi-source tree control. By using a tree structure in an HTML report, you can show hierarchical data from a multi-dimensional data source (for example, SAP BW), that uses the parent/child model. You may also use a tree control for non-hierarchical data sources. Level hierarchies are not supported. The behavior of the tree control is integrated with the parameter definition. If a parameter is defined as a single value and that parameter is bound to a tree control, the tree control uses option buttons for each node in the hierarchy. If the parameter is defined as Multiselect OR or Multiselect AND, and that parameter is bound to a tree control, then the tree control uses check boxes for each node in the hierarchy, enabling you to select multiple nodes.

Calendar

Inserts a calendar. Date parameters can utilize a built-in calendar control that enables you to select the desired date or range of dates in a pop-up dynamic calendar. A procedure or parameter that is added to or referenced in the HTML canvas, and contains date parameters, will have a Calendar control type available in the New Parameters dialog box. When you select the Calendar control type, a text box with a calendar icon displays in the Design view of the layout. The text box is the only control available for the calendar, and the icon always displays to the right of the text box. The icon cannot be positioned independently from the text box. Note that when programmatically returning a date to the calendar, the date must be in a Db2 Web Query date format that specifies the complete date from the list of supported data types in the Popup Calendar Settings section of the Settings panel.

Slider

Inserts a horizontal or vertical slider control that has a numeric range of values to be used with a report or chart. This enables you to use a slider bar to select from a range of values.

Adding Paging/Search Capabilities to a Control

With a List Box, Drop Down List, or the first control of a Double List, you can add additional paging and search capabilities. Designed for controls that have a large number of valid values, this functionality enables you to:

- Page through a large list of values before making a selection.

- Search for a value within a control.

These options give you a quick way of narrowing your search when you know the value, or partial value, that you want to use to filter your page.

The *Add 'Paging/Search' control* function is available on the HTML canvas through the right-click context menu for the List Box, Drop Down List, and the first control of a Double List.

Use this function to enhance the capabilities of an individual control. If you want to create a single Paging/Search option that applies to multiple controls on a page, see *Adding a Paging/Search Control to an HTML Page* on page 111.

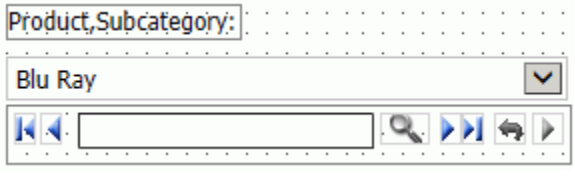
You can also create a search control, which includes only the search functionality of the paging/search control.

Procedure: How to Add Paging/Search Capability to a Control

To add paging and search capabilities to a control that has a large number of valid values:

1. On the HTML canvas, select a List Box, Drop Down List, or the first control of a Double List, then right-click the control and select *Add 'Paging/Search' control* from the shortcut menu.

   The paging/search control is added to the canvas below the input control. The following image shows the paging/search control added below a Drop Down List for Product Subcategory.

   Note that the blue left and right arrow buttons are for paging. The gray magnifying glass and arrow buttons are for searching.

   At run time, the paging/search control shows a number or page range of values. The default range is 1-50, meaning the input control starts at the first value and displays a range of 50 values per page.

2. Optionally, in the Search/paging control section of the Properties panel, with the navigator object selected, select the following default search options for the paging/search control:

   - **Match Prefix.** The search string matches the beginning of each value. When not enabled, the search string can appear at any point in the control value.
   - **Match Case.** Searches are case-sensitive.

- **Match Whole Word Only.** Search values are returned only for entire words that match the search string.
- **Instant Search.** The value is automatically searched as you type in the input control.
- **Keep selected values.** When selected, the values that you select on each page or search for are retained when you leave the page, and are shown again when you return to it. If you run the request, all previous selections are used.
- **Show settings.** Search options display in the paging/search control, by default. You can show or hide search options in the control by clicking the search arrow button.
- **Number or Range.** Specifies the initial number of values or range of values to show in the control.

3. Run the HTML page.

   The paging/search control shows the specified range. For example, 1-50.

4. Use the arrow buttons to page through the values to be displayed for the input control. You may select *First*, *Previous*, *Next*, or *Last*.

   For searching, enter a search term in the paging/search control and press the Enter key or click the *Search* (magnifying glass) button.

   To display search criteria, click the search arrow button on the far right. The search criteria list is displayed. You can select any of the following options:

   - **Match Prefix.** Search value prefix must match the prefix of the value in the input control.
   - **Match Case.** Search value must match the value in the input control exactly.
   - **Match Whole Word Only.** Search value word must exist in the string.
   - **Instant Search.** The value is automatically searched as you type in the input control.
   - **Keep selected values.** When selected, the values that you select on each page or search for are retained when you leave the page, and are shown again when you return to it. If you run the request, all previous selections are used.

   Toggle the search arrow button to hide the search criteria list.

5. Click the *Run* button to refresh the report, showing the value selected in the input control.

Adding a Paging/Search Control to an HTML Page

A global paging/search control lets you add additional capabilities to an HTML page that contains multiple instances of the following controls: List Box, Drop Down List, and the first control of a Double List. Designed for controls that have a large number of valid values, this functionality enables you to:

- Page through a large list of values before making a selection.
- Search for a value within multiple controls.

The global paging/search is added as a DOCUMENT level property in the Search/paging section of the Properties panel.

Use this option to enhance the functionality of an HTML page. If you want to enhance the functionality of an individual control, see *Adding Paging/Search Capabilities to a Control* on page 108.

Procedure: How to Add a Paging/Search Control to an HTML Page

Use this procedure to add global paging and search capabilities to an HTML page with multiple controls.

The search options for the global paging/search control override the options for individual controls. For example, if *Keep selected values* is selected for the global paging/search control, that setting is applied to all controls on the page, regardless of whether they have Keep selected values enabled.

1. In the Properties panel for the DOCUMENT object, in the Search/paging section, click the *Enable* property and select *Yes* from the drop-down menu.
2. Optionally, in the Search/paging control section of the Properties panel, select the following default search options for the paging/search control:
   - **Match Prefix.** The search string matches the beginning of each value. When not enabled, the search string can appear at any point in the control value.
   - **Match Case.** Searches are case-sensitive.
   - **Match Whole Word Only.** Search values are returned only for entire words that match the search string.
   - **Instant Search.** The value is automatically searched as you type in the input control.
   - **Auto-link.** When enabled, selecting an input control on the page automatically connects the global paging/search control to it. When not enabled, connect the global paging/search control by selecting an input control and then clicking the global paging/search control.

- **Keep selected values.** When selected, the values that you select on each page or search for are retained when you leave the page, and are shown again when you return to it. If you run the request, all previous selections are used.
- **Show settings.** Search options display in the global paging/search control, by default. You can show or hide search options in the control by clicking the search arrow button.
- **Number or Range.** Specifies the initial number of values or range of values to show in the control.

3. Run the HTML page.

   The global paging/search control appears in the upper-right corner of the window (Home position).

4. Click an input control.

   The global paging/search control appears below the selected input control.

   The label for the input control is displayed showing that it is linked to that input control. If no label is associated with the input control, the window will show the unique identifier for the input control, for example, combobox1.

   If you want multiple pages, type a range in the Search/Paging field in the format *1-2 of 4* and press the Enter key.

5. Click the *Show setting*s button on the Global Search and Paging window to expand the window and show the search and paging options.

   The Search/Paging options are:

   - **Match Prefix.** Search value prefix must match the prefix of the value in the input control.
   - **Match Case.** Search value must match the value in the input control exactly.
   - **Match Whole Word Only.** Search value word must exist in the string.
   - **Instant Search.** The value is automatically searched as you type in the input control.
   - **Auto-link to selected control.** Link the Search/Paging control to the selected control (default).
   - **Keep selected values.** When selected, the values that you select on each page or search for are retained when you leave the page, and are shown again when you return to it. If you run the request, all previous selections are used.

Toggle the *Up* arrow button to display and collapse the Search/Paging options. Click the *Home* icon on the Search/Paging window to move the window back to the Home position.

Defining the Load Order for Control Values

You can specify when controls are populated with values. If a control is configured to display a large number of values, you can set the values to populate after the content on the page is loaded, allowing the page to load more quickly.

To specify load order for a control, select it, and, in the Properties panel, select a value for the Loading procedure property. You can select one of the following values:

- **Initial loading.** The control is populated with values when the page initially loads. This is the default value.
- **Post initial loading.** Control values populate once the Load task is executed, after the rest of the page has loaded.
- **On demand.** The control is not populated automatically. To populate the control, create a task in the Tasks & Animations panel, select *Refresh* from the Requests selections menu, and set the control as the target, or use the IbComposer_populateDynamicCtrl function.
- **On demand and hide initially.** The control is not populated automatically and does not display until it is populated. To populate and display the control, create a task in the Tasks & Animations panel, select *Refresh* from the Requests selections menu, and set the control as the target, or use the IbComposer_populateDynamicCtrl function.

Note: You can set the Loading procedure property for any object that is treated as an input control, including tables, grids, and maps.

Procedure: How to Use a Button to Populate a Control

You can use the Loading procedure property to specify when controls are populated with values at run time. This example shows how to create a control object that initially appears blank but populates with values when a button is clicked.

1. In the HTML canvas, add a control to the page. On the *Controls* tab, in the *Controls* group, select a type of control and draw it onto the page.
2. Add values to the control. Select the control and open the *Settings* panel. Select a data type for the list of values in the control and populate the control.

   Optionally, bind the control to a parameter from a request referenced in the HTML page.

3. Set the control so that values do not load automatically, and instead must be prompted at run time.
    a. Select the control and open the *Properties* panel.
    b. Set the Loading procedure property. Select *On demand* or *On demand and hide initially* so that the control does not populate with values automatically. If you select *On demand and hide initially*, the control will not be visible until you prompt it to load values.
4. Add a button or other item to the page to trigger the loading of values in the control.
5. Open the *Tasks & Animations* panel and create a new task to execute control object population. Specify the following properties for the task:
    a. Set the Trigger Type for the task, for example, *Click*, which executes the task when the Trigger Identifier is clicked.
    b. Set the Trigger Identifier to the button or other object that you want to use to trigger loading of control values.
    c. Under Requests/Actions, in the Requests selections menu, click *Refresh*.
    d. Change the Target type to *Input Control*.
    e. From the Target/Template Name menu, select the control object.
6. Save and run the HTML page.

If the Loading procedure property is set to On demand, the control object is empty when the page initially loads. When you execute the task to refresh the control (for example, by clicking a button), the control populates with values.

If the Loading procedure property is set to On demand and hide initially, then when the page initially loads, the control is not visible. When you execute the task to refresh the control, the control appears and is populated with values.

Procedure: How to Enter Masked Text in a Text Box

When entering a value in a text box at run time, you may set the Mask text property so that the text is not displayed as text, but masked by default characters. This is recommended when using passwords or other sensitive information.

1. Select the Edit Box object to view the associated properties in the Properties panel.
2. From the Mask text property field, select *Yes*.
3. Run the report and enter a value in the text box.

The value being entered appears as masked text.

Procedure: How to Upload a File to an Application Folder Using an Edit Box

When you change the Type property of an edit box to *File*, you can use it to browse files on your system. You can then use the IbComposer_saveSelectedFile function to upload the file to a selected location in the Data Servers area.

1. Create an HTML page.
2. On the *Controls* tab, click *Edit Box*, and draw an edit box on the page.
3. In the Properties panel, change the Type property for the edit box to *File*.

   A Browse button is added to the edit box.
4. Optionally, in the Properties panel for the edit box, use the Accept property to specify the types of files that will be available for selection using a comma-separated list of file type specifiers.

   These file type specifiers can be extensions, including the period, such as .jpg or .fex, or MIME type strings, such as image/* or text/html.
5. In the Properties panel, click the ellipses button for the Copy file to property, select the location in the Data Servers area in which to save the uploaded files, and click *Select*.
6. On the *Components* tab, in the *Generic Elements* group, click *Button*, and draw a button onto the page.
7. Select the button and open the Properties panel.
8. At the top of the Properties panel, click the *Events* button.
9. Click the ellipses button next to Click to add an on-click event.
10. In the Embedded JavaScript tab, in the new on-click event, add the following JavaScript function:

    ```
    IbComposer_saveSelectedFile("uniqueID")
    ```

    Where *uniqueID* is the unique identifier for the edit box.
11. Run the page and click the *Browse* button on the edit box.

    Select a file that you want to upload.
12. Click the button on the page to execute the IbComposer_saveSelectedFile function.

    The file is uploaded to the folder specified in the Copy file to property of the edit box.

Procedure: How to Create a Double List Box

A double list box displays values in two lists. The list on the left shows available values that are not being used, while the list on the right shows values that have been selected for use in the parameter. You can move values within lists or between lists by using the arrow buttons between them. You can select multiple values at once by holding the Ctrl key while selecting values, and then clicking the single right arrow button, or add each value one at a time.

In order to use a double list box with a parameter, the parameter must be set up to accept multiple values. Ensure that the Variable Type for the parameter value is Multiselect OR or Multiselect AND in the procedure.

1. From the HTML canvas, insert a report with parameters that accept multiple values.

   For example, create a report with *Multiselect OR* as the variable type for the parameter, accepting a dynamic list of values from a Master File.

2. When the New Parameters dialog box is displayed, right-click the value in the Control Type column and click *Double list control*, and then click *OK*.

   A list box is created in the layout and assigned the Unique Identifier listbox*(n)*, where *(n)* is a number.

3. Save and run the HTML page.

4. Select any values selected in the available list by using the single right arrow, or select all values from the available list by using the double right arrow. Deselect values selected in the selected list by clicking the single left arrow, or deselect all values by clicking the double left arrow. When you deselect a value, it returns to the available list in the order it was originally sorted in.

   You can reorder values in the selected list by using the up and down arrows. Move the selected value up or down one spot by using the single up or single down arrow. Move the selected item to the top or bottom of the list by clicking the double up or double down arrow.

5. Click the *Run* button to run the report with the selected value parameters.

Procedure: How to Format Selected and Unselected Push Buttons for Radio Button and Check Box Controls

This procedure shows how you create a push button from a radio button or check box control. It also shows how you can apply background color to push buttons to distinguish between selected and unselected values.

1. In the HTML canvas, create a radio button or check box control by clicking *Radio Button* or *Check Box* on the *Controls* tab of the ribbon. Populate the control with multiple values.

2. Select the control and open the *Properties* panel.
3. In the Properties panel, from the Type drop-down menu, select *Push button*.

   Two additional properties appear: *Selected pushbutton background color* and *Unselected pushbutton background color*.
4. From the Selected pushbutton background color drop-down menu, select a background color.
5. From the Unselected pushbutton background color drop-down menu, select a different background color.
6. Save and run the HTML file.

   The specified background colors appear for the selected and unselected values.

Procedure: How to Add Images to Radio Buttons With Static Content

This procedure shows you how to add images to radio buttons with Static content.

1. In an HTML canvas, create a radio button control by clicking *Radio Button* on the Controls tab and drawing a control box.
2. In the Settings panel, select the *Add display image* check box.
   a. Click the *New* button to populate the control box with multiple values.

      Note: The first value defaults to *Selected*, but you can clear this check box.

   b. Right-click in the Display column for a value to select an image to include for that value. The Open File dialog box displays.

   c. Navigate to and select an image. The path to the image is added to the Display column.

   d. To change or remove the text in the Display column, double-click it and carefully make the desired changes.

e. Run the HTML page. Since the *Selected* check box for the Camera radio button is selected in the Settings panel, it is populated as the default when the page is run, as shown in the following image.

Procedure: How to Set Calendar Properties

1. Add a calendar to the HTML page by clicking the *Calendar* command on the Controls tab.

   Position the cursor over the HTML canvas. The pointer changes to a crosshair.

2. Drag the crosshair to create a calendar and adjust it to the size you want.

   A calendar placeholder is created in the layout and assigned the name calendar*(n)*, where *(n)* is a number.

3. Optionally, change the calendar properties by adjusting the properties displayed in the Properties panel.

   The Properties panel includes the following properties for calendar controls:

   - **Number of months.** Allows you to determine the number of months to show simultaneously in the calendar control. Type a single integer value to display that number of months in a single row in the control, or type two integers, separated by a comma, to show months in the specified number of rows and columns. One month appears, by default.

- **Step months.** Determines how many months to skip when clicking the forward or back arrows in the calendar control. This can be useful if you are showing multiple months at once and want to save clicks when navigating the control. The default is to move one month at a time.
- **Show On.** Determines how to expand the calendar control. The default is Button clicked, which displays the calendar when you click the calendar icon. Focus hides the calendar icon and displays the calendar control when you click the calendar text box. Both allows you to display the calendar control by clicking either the text box or the icon.
- **Show week.** When you select *Yes*, numbers the weeks of each year in the calendar control. No is the default value.
- **Show button pane.** When you select *Yes*, adds a button pane to the bottom of the calendar control. The button pane contains the Today button, to select the current date in the control, and Done button, to close the control. No is the default value.
- **First day of the week.** Determines which day of the week appears first in the calendar control.
- **Calendar image.** Allows you to change the default calendar icon to an image available in your environment. The image is cropped to the size of the calendar icon.
- **Animation.** Determines the animation that is used to display the calendar control. Show is the default value.

4. Bind an existing parameter to the calendar.

   Binding a parameter to a calendar control creates an incoming parameter. The incoming parameter value must be in a Db2 Web Query date format that specifies the complete date from the list of supported data types in Calendar Properties. The incoming parameter value will populate the calendar with date values.

   - Click the *Parameters* view tab.
   - Select the center of the parameter name object, drag the parameter to the center of the calendar object, and release the mouse to complete the binding.
   - To unbind the parameter from the calendar, select the arrow head on the line, so that the line is bold. Right-click and click *Break binding*.

5. Bind the calendar to a parameter.

Binding the calendar to a parameter will populate the parameter with a date value.

- Click the *Parameters* view tab.

  The Settings panel opens, showing the calendar setup options in the Popup Calendar Settings section. The calendar setup options enable you to set the range of dates available to the user at run time. Available dates are represented as an active hyperlink (blue and underlined). Unavailable dates are static (black without underlines).

  Note: The Settings panel for a calendar contains different options depending on the selected data type.

- Create the values for the calendar. You can create Static or Dynamic values.

  When the Current/Start date option is checked, the current date will be used in the calendar control at run time.

  The Date Range options for setting up the calendar include:

  - **Static.** This option will set a static date range in which the developer will select a start date and an end date using a pop-up calendar icon, or by clicking the month, day, or year from the controls.

    The pop-up calendar icon appears in the From and To sections when the *Static* Date Range is selected. If you click the pop-up calendar icon, a pop-up calendar appears and shows the current date selected and circled in red, by default. As you scroll through the calendar with the left/right arrows, the currently selected day will remain highlighted for each month. Clicking a date will add that date to the control. Dates can be selected by scrolling left to right, entering the month, day, and year as text, or by selecting the month, day, and year from the drop-down list and spin boxes.

  - **Relative.** This option allows you to set a specific number of days, months, and years relative to the current date. The current date (at run time) will always be the reference or starting point and the calendar will show a number of days, months, and years relative to the current date. The range could be all in the past (for example, five years prior to the current date) or all in the future (for example, five years in the future). This is selected as the default Date Range.

  - **Dynamic.** This option allows you to point to a request that returns a range of dates. Clicking the *Select custom request* drop-down list box allows you to choose a preexisting request. The request must return two date values on the same data line in XML format. The date values must be returned in a format that returns two digits for the month and day, and four digits for the year, for example, MM/DD/YYYY.

- **Limit by data.** This option limits the available dates to those that appear in the data source for the control. For example, if you set the Data Type in the Settings panel to *Dynamic* and select a data source and date field for the control, then only the dates listed in that field will be available for selection.

- Select the center of the calendar, drag the calendar to the center of the parameter object, and release the mouse to complete the binding.

Working With a Slider Control

You can use a slider control to select numeric values. A slider allows you to select a single number value from a range of available values. You can click a point in the slider to select it, or use the arrow buttons at either end of the slider to move the slider one increment at a time. If you create a slider for a static parameter, you can use the Settings panel to specify the minimum and maximum value and the increment between values on the slider. If you create a slider for a dynamic parameter, all values from the selected data source and field are available on the slider.

You can also create a double-head slider. A double-head slider control is a type of slider control in which you can select a range of values for displaying data in a report or chart. Rather than having one slider head, you have two. In order to use a double-head slider control, you must have two variables in the report or chart that will be controlled by the double-head slider control. One variable will be controlled by one head of the double-head slider control and the other variable will be controlled by the other head of the double-head slider control. The variables need to be specified in the procedure as FROM and TO.

Note: You can use the Report canvas to create a parameter that accepts a range of values using the Expression Builder dialog box. The Chart canvas requires you to manually create a parameter that accepts a range of values.

Procedure: How to Create an HTML Page That Uses a Slider Control

1. Create a new HTML page.
2. Add a report object to the HTML canvas.
3. Reference a report that contains a variable that accepts a numeric value.

   The New Parameters dialog box opens.
4. In the Control Type column of the New Parameters dialog box, right-click the first parameter, point to *Slider*, and then select either *Horizontal* or *Vertical*.
5. Click *OK* to close the New Parameters dialog box and add the slider control to the HTML page.

If you click the *Parameters* view tab at the bottom of the HTML canvas, you will see that the slider control is chained to the parameter in the report, and if you select the slider control and open the Settings panel, you will see that it uses a dynamic list of values to supply a selected value to the parameter in the report.

6. Optionally, select the slider control and open the Properties panel. The following properties are specific to slider controls:

   - **Show Input.** Select *Yes*, which is the default, to include an edit box with the slider, displaying the selected value and allowing you to change the slider selection by typing a value into it. Select *No* to hide the edit box and not display the value selected on the slider. Select *Read only* to hide the edit box and display the selected value above the slider.
   - **Show/Hide Arrows.** Select *Yes*, which is the default, to include arrow buttons that allow you to move the slider one increment at a time, or select *No* to hide the arrow buttons. You can click a point along the slider to move the slider head there.
   - **Range.** Select *No* for a single-head slider control, or select *Yes* to create a double-head slider control. For more information about creating an HTML page with a double-head slider, see *How to Create an HTML Page That Uses a Double-head Slider Control* on page 123.

7. Save and run your HTML page.

   The slider control controls what you see in your report or chart by allowing you to select a value.

Procedure: How to Create a Double-head Slider Control From the Ribbon

1. While in the HTML canvas, click either *Horizontal* or *Vertical*, from the *Slider* command, in the *Controls* group, on the *Controls* tab.

   The pointer changes to a crosshair.

2. Drag the crosshair to create a single-head slider control.
3. While the single-head slider control is selected, in the Properties panel, change the Range property from *No* to *Yes*.

   Changing the Range property to Yes changes the single-head slider control to a double-head slider control. You can now use this control to select a range of values to display.

Procedure: How to Create a Double-head Slider Control Using the New Parameters Dialog Box

1. Create a report object or chart object on the HTML canvas.

   The pointer changes to a crosshair.

2. Drag the crosshair to create a report or chart
3. Reference a report or chart with two variables used in a FROM... TO... relationship.

   The New Parameters dialog box opens.
4. In the Control Type column of the New Parameters dialog box, right-click the first parameter and then click *Slider*, and select either *Double-head Horizontal* or *Double-head Vertical*.

   Note that the Control Type for the second parameter changes to *Append to Above*. This means that the second parameter has been recognized as being part of a range and will be controlled by the double-head slider control specified in the first parameter.

Procedure: How to Create an HTML Page That Uses a Double-head Slider Control

1. Create a new HTML page.
2. Add a report object to the HTML canvas.
3. Reference a report that contains a variable that accepts a range of values.

   The New Parameters dialog box opens.
4. In the Control Type column of the New Parameters dialog box, right-click the first parameter, point to *Slider*, and then select either *Double-head Horizontal* or *Double-head Vertical*.

   Note that the Control Type for the second parameter changes to *Append to Above*. This means that the second parameter has been recognized as being part of a range and will be controlled by the double-head slider control specified in the first parameter.
5. Click *OK* to close the New Parameters dialog box and add the double-head slider control to the HTML page.
6. Click the *Parameters* view tab at the bottom of the HTML canvas.
7. Drag the unbound parameter out of the Unbound Parameters box.
8. Drag the second half of the slider control to the unbound parameter.
9. Click the slider control and click the Settings panel.
10. In the Input Control Population section, change the Data type to *Dynamic*.
11. Select the data source you used in the Data Source drop-down list.
12. In the Value Field, click the ellipsis (...) and select the field being controlled by the double-head slider control.
13. Save and run your HTML page.

The double-head slider control controls what you see in your report or chart by allowing you to select a range of values rather than one specific value or all values. The first head updates the FROM value and the second head updates the TO value.

Using Multi-Select Lists

Two types of multi-select lists can be used in a report: a regular list box or a drop-down list. When using either type of multi-select input control to supply parameter values, the Multiple property value indicates whether multiple values can be selected from a list of supplied values at run time.

Procedure: How to Create a List Box with Multiple Values

A multi-select list box enables you to select multiple values by using the Ctrl key while selecting values. In order to select multiple values in the list box, the procedure must be set up to accept multiple values. Ensure that the Variable Type for the parameter value is Multiselect OR or Multiselect AND in the procedure.

1. From the HTML canvas, insert a report with parameters that accept multiple values.

   For example, create a report with *Multiselect OR* as the variable type for the parameter, accepting a dynamic list of values from a Master File.

2. When the New Parameters dialog box is displayed, accept the default control type of *List box* and click *OK*.

   A list box is created in the layout and assigned the name listbox*(n)*, where *(n)* is a number.

3. Select *Multiple* from the Multiple drop-down list in the Properties panel.

   This indicates that multiple items can be selected from the list box.

4. Save and run the HTML page.
5. Select multiple values by using the Ctrl key while selecting values from the list box.
6. Click the *Run* button to run the report with the selected value parameters.

Procedure: How to Create a Drop-Down Check Box List with Multiple Values

A multiselect drop-down list enables you to select multiple values using the check boxes adjacent to the values. In order to select multiple values in the drop-down list, the procedure must be set up to accept multiple values. Ensure that the Variable Type for the parameter value is Multiselect OR or Multiselect AND in the procedure.

1. From the HTML canvas, insert a report with parameters that accept multiple values.

   For example, create a report with *Multiselect OR* as the variable type for the parameter, accepting a dynamic list of values from a Master File.

2. When the New Parameters dialog box appears, accept the default control type of *List box* and click *OK*.

   A list box is created in the layout and assigned the name listbox*(n)*, where *(n)* is a number.

3. Right-click the control, select *Set Control Type,* and select *Drop down list* to change the control type.

   Note: In the Properties panel, the Multiple property for the control is automatically set to *Multiple*.

4. Save and run the HTML page.
5. Select multiple values using the check boxes displayed when you click the down arrow.
6. Click outside of the control to close the drop-down list.
7. Click the *Run* button to run the report with the selected value parameters.

Saving Control Selections in a Browser Session

You can link the controls on HTML pages so that they automatically default to the same selected value when you switch between multiple pages in the same browser session. To do this, you need to assign a common value to the *Global name* property for each control that you want to link.

For example, you may have two HTML pages that contain controls with information that relates to regional sales. You can assign a value to the *Global name* property, such as *Region*, to each of the controls that you want to link. When you run those pages in the same browser session, and choose a value from one of the linked controls, such as *Southeast*, the controls on the other pages will refresh and display the information for the Southeast, by default.

Note: A selected value is retained as the default only during a single browser session. The value is not retained after you close the browser.

Procedure: How to Save Control Selections in a Browser Session

This procedure describes how to assign a value to the *Global name* property, which enables you to save control selections within a browser session.

Note: A selected value is retained as the default only during a single browser session. The value is not retained after you close the browser.

1. Create or open an HTML page that contains a control.
2. Click the control to highlight it.
3. In the Properties panel, under Miscellaneous, type a value in the *Global name* field. This value can be any alphanumeric string, such as *GlobalRegion1*.

4. Save the HTML page.
5. Repeat steps 1-4 for any controls that you want to link. You must assign the same *Global name* property value for each control that you want to link.

Using Tree Controls

You can insert a single-source tree control or a multi-source tree control in an HTML page.

Procedure: How to Add a Tree Control to an HTML Page Using an Existing Procedure

You can select an existing procedure to add to the tree control in an HTML page. When you select a procedure, it should use fields from the parent/child hierarchy and be set up as follows:

```
TABLE FILE file
SUM FST.dispfield
BY ParentUniqueField
BY UniqueField
BY datafield
ON TABLE PCHOLD FORMAT XML
END
```

where:

file

Is the name of the data source.

dispfield

Is the field whose values display in the tree control.

ParentUniqueField

Is the field that represents the parent for the parent/child hierarchy (PROPERTY = PARENT_OF).

UniqueField

Is the field that represents the unique IDs for the hierarchy members (PROPERTY=UID).

datafield

Is the field whose values are passed as the parameter value.

After the procedure is set up, follow these steps:

1. From the Controls tab, insert a tree control.

   Tip: You may select Single source Tree control or Multi source Tree control. If no type is selected, Single source Tree control is the default. Single source Tree controls must be populated from a multi-dimensional data source such as SAP BW, SSAS, or Essbase.

The pointer changes to a crosshair.

2. Drag the crosshair to create a tree control, and adjust it to the size you want.

   A tree control is created in the layout and assigned the name treecontrol*(n)*, where *(n)* is a number.

3. Optionally, you may click the *Expanded* property from the Properties panel to show the tree control expanded at run time.
4. Optionally, you may click the *Hyperlink* property from the Properties panel to show the tree nodes as hyperlinks, instead of radio buttons at run time.
5. From the Settings panel, click *Dynamic* as the Data type.
6. Select *Explicit (Requests panel)*, select a procedure from the Request drop-down list, and then click *OK*.

   The procedure name is added as the explicit procedure.

   Note: In prior releases, an explicit procedure was known as an “external” procedure.

7. Click the browse (...) button adjacent to *Value from field* to select a field from the hierarchy or type the field manually.

   The Value from field is the data source field from which the values will be retrieved.

8. Click the browse (...) button adjacent to *Display from field* to select a field from the hierarchy or type the field manually.

   The Display field is the text that represents the parameter value in the tree control.

9. Save and run the page to view the multi-dimensional data source in the tree control.

Procedure: How to Populate a Multi Source Tree Control

You can show a tree structure for a non-hierarchical data source by using a tree control. By identifying the number of levels for the tree control, you are able to populate each level of the tree control with its own procedure. Setting the number of levels creates a tree structure by which each level is its own subcontrol, chained together with no conditions.

This procedure describes how to add parameters for a tree control, where the number of levels property is set.

1. In the HTML canvas, insert a Multi source tree control.

   ❏ Click *Multi source Tree control* from the Tree control drop-down list, located on the Controls tab.

   The pointer changes to a crosshair.

2. Drag the crosshair to create a tree control, and adjust it to the size you want.

   A tree control is created in the layout and assigned the name treecontrol*(n)*, where *(n)* is a number.

3. From the Properties panel, type the Number of levels for the tree control, and press the Enter key.

   This enables you to specify the number of levels to populate.

4. Optionally, you may click the *Expanded* property from the Properties panel to show the tree control expanded at run time.

5. With the tree control selected, click the *Parameters* tab.

   The tree control object shows the set number of levels.

6. Select each level of the tree control and create the settings for its data population.

   When creating a Multi source Tree control, the static data type is not available. If creating static values for the tree control, you must create a single source tree control.

7. Optionally, to add an additional level for the tree control, right-click the tree control object on the Parameters tab and select *Add level*.

   Note: Add level only appears for a Multi source tree control object.

8. Click the added level to view the Settings panel for that level.

9. Switch to the *Design* tab of the HTML canvas to preview the populated tree control.

   Note that the Properties window drop-down list for the tree control shows each level of input values.

10. Save and run the page.

    Note: If you select a lower level node in one level and a higher level node in another level, when the procedure is executed, only the lowest level selections will take effect. For example, you have 3 levels: COUNTRY, CAR, and MODEL. Under ENGLAND, TRIUMPH, you select TR7. Under FRANCE, you select PEUGEOT. At run time, you will only receive the records for TR7, because you did not select a MODEL under the FRANCE node.

    The tree control populates each level with values.

    Note: A value must be selected for each level before you can click the *Save Selection* button.

    If a selected value is specified for a field that is not in level1, then corresponding selected values must also be specified for the preceding levels.

Only the first and second levels load at run time. If a selected value is specified for a field in level1, only values from the first two levels will be selected.

Procedure: How to Add a Tree Control to an HTML Page Using Static Values

This procedure describes how to add static data type parameters for a tree control, where the Number of levels property for the tree control is not set. This enables you to add a static list of values.

1. In the HTML canvas, insert a tree control.
2. From the Settings panel, select *Static* as the Data type.

   Static is selected, by default. You may select an item, delete it, or add a subitem.
3. Create the parameter values for the control:

   - Click the *New* button to add a list of values. The values are added in a sequential hierarchical structure. The last value added appears in the Value and Display Value fields.
   - To edit the value, manually type the desired value in the Value and Display Value fields.
   - Click *Append child item* from the Static values drop-down list to append a value at the level currently selected, and create a new value as the child of the selected value.
   - Click *Insert before* from the Static values drop-down list to insert a value before the selected value. Note the number of the value.
   - Click *Insert after* from the Static values drop-down list to insert a value after the selected value. Note the number of the value.

     Repeat these steps until the list contains all of the values you want to include.
   - Optionally, click the *Delete* button to eliminate any values.
4. Click the *Selected* check box to show the entry in the Value field as the default value.
5. Select the *Send display value* check box to send the display value, rather than the actual data, to the parameter.
6. Save and run the page to populate the tree control with static values.

Procedure: How to Create a New Tree Control From the New Parameters Dialog Box

When a report contains one or more new amper variable parameters, the New Parameters dialog box opens when you save the report and return to the HTML canvas. You can assign a new Single source or Multi source Tree control from the HTML page to the parameter from the New Parameters dialog box.

For each parameter, you will find Name and Control Type fields, a Create control check box, and options to set the Control Type to a Single source or Multi source Tree control.

1. Reference a report that contains parameters.

   When referencing a report with parameters, the New Parameters dialog box opens, prompting you to create the control type.

2. Right-click the value in the Control Type column and select the tree control from the New Parameters dialog box.

   The Control Type column refreshes, showing the selected control.

3. Click *OK* to close the New Parameters dialog box.

   The report is added and the associated parameters are bound to the tree control.

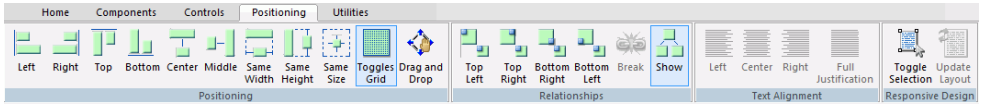
Reference: Usage Notes for Chaining Tree Controls

The following usage notes apply when chaining tree controls. You may chain controls from the New Parameters dialog box and from the Parameters tab.

- When the Multi source Tree control is a link in the chain, the New Parameters dialog box enables you to share parameters with the same multi source control.
- You can chain a Multi source Tree control to a Single source Tree control.
- You can chain a Single source Tree control to another Single source Tree control.
- You can chain a Multi source Tree control to another Multi source Tree control.
- You can chain a tree control to another non-tree control, such as Drop down list or List box.
- Chaining cannot be done with only field names.
- Dynamic population of controls with field names need to use SYSCOLUMN calls.

Positioning Objects in an HTML Page Using the Positioning Tab

Positioning options enable you to multi-select objects and position them relative to one another. The Positioning tab contains the Positioning, Relationships, Text Alignment, and Responsive Design groups, as shown in the following image.



Aligning Objects in an HTML Page Using the Positioning Group

The Positioning group contains the Left, Right, Top, Bottom, Center, Middle, Same Width, Same Height, Same Size, Toggles Grid, and Drag and Drop commands.

Left

Aligns the objects to the left. This works in relationship positioning.

Relates an object to the left of another object.

Right

Aligns the objects to the right. This works in relationship positioning.

Relates an object to the right of another object.

Top

Aligns the objects to the top of the highest selected object. This works in relationship positioning.

Bottom

Aligns the objects by the bottom of the lowest selected object. This works in relationship positioning.

Center

Aligns the objects to the center of the page. This works in relationship positioning.

Aligns objects at the horizontal center point of the object.

Middle

Aligns the objects to the middle of the page. This works in relationship positioning.

Aligns objects at the vertical center point of the object.

Same Width

Makes two or more objects the same width.

Same Height

Makes two or more objects the same height.

Same Size

Makes two or more objects the same size.

Toggles Grid

Turns the grid on and off.

Drag and Drop

Turns parent/child Drag and Drop on and off.

When this command is off (default), drag and drop is used only to reposition objects on the HTML canvas.

When this command is on, drag and drop is used to designate a parent/child relationship between two objects.

Procedure: How to Designate a Parent/Child Relationship using Drag and Drop

There are instances where you want an object in your HTML canvas to be a child to another object. For example, you may have a report (child) that you want to place within a tab container (parent).

1. Open an HTML file in which the two objects have been created.
2. Click the *Positioning* tab.
3. Click the *Drag and Drop* command to toggle it on.

   Alternatively, you can select *Toggle Drag and Drop* from the HTML canvas shortcut menu.

   The Drag and Drop command on the Positioning tab is now highlighted.
4. Click the designated child object and drag it toward the designated parent object.

   A small rectangle appears below your cursor, but the child object itself does not move.
5. Move your cursor to the designated parent object and drop.

   The parent/child relationship is now created.
6. To restore the default drag and drop, click the *Drag and Drop* command to toggle it off.

   Note: When the Drag and Drop command is on, you are unable to reorder tabs or page containers, when using the Tab, Accordion, or Window components.

Relating Objects in an HTML Page Using the Relationships Group

The Relationships group contains the Top Left, Top Right, Bottom Right, Bottom Left, Break, and Show commands.

Top Left

Relates two or more objects on a page. Two objects must be selected. Relates an object to the top left of another object.

Top Right

Relates two or more objects on a page. Two objects must be selected. Relates an object to the top right of another object.

Bottom Right

Relates two or more objects on a page. Two objects must be selected. Relates an object to the bottom right of another object.

Bottom Left

Relates two or more objects on a page. Two objects must be selected. Relates an object to the bottom left of another object.

Note: The above four commands have been deprecated and will be removed from a future release of Developer Workbench.

Break

Breaks the relationship between the objects selected.

Show

Shows or hides the relationship between objects.

Aligning Text in an HTML Page Using the Text Alignment Group

The Text Alignment group contains the Left, Center, Right, and Full Justification commands. These commands align text within the HTML page.

Left

Aligns selected text to the left.

Center

Aligns selected text to the center.

Right

Aligns selected text to the right.

Full Justification

Fully justifies the selected text, meaning that the text fills the width of the text box.

Enabling Responsive Design for an HTML Page Using the Responsive Design Group

The Responsive Design group contains the Toggle Selection and Update Layout commands. These commands are enabled when the Responsive design property is set to *Yes*.

Toggle Selection

Toggle on to allow containers that include content to be dragged to a new position. Toggle off to prevent those containers from being repositioned.

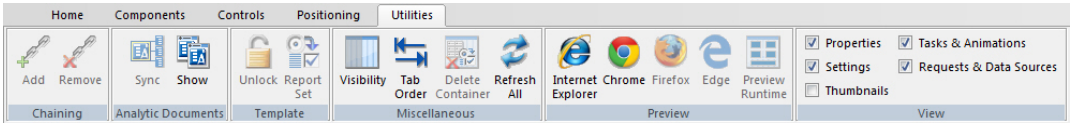
Update Layout

Click to refresh the HTML canvas after you reposition containers, to show how the containers will stack when folded. This command is activated when the Autosize Enable property is set to Yes for both the document and the object.

Changing HTML Page Properties Using the Utilities Tab

You can change HTML page properties using the Utilities tab. For example, suppose you want to test your HTML page in a specific browser. The default browser specification is considered a property of your HTML page. To run your page in a different browser, open the Utilities tab and select the desired browser from the Preview command group.

The Utilities tab is shown in the following image.



Chaining Objects in an HTML Page Using the Chaining Group

You can add or remove chaining options using the Chaining group. The Chaining group is shown in the following image.

The commands are:

Add

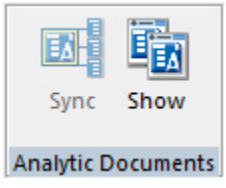
Chains two or more objects that are selected in the Parameters tab.

Remove

Unchains two or more selected, chained objects in the Parameters tab.

Synchronizing an Analytic Document in an HTML Page Using the Analytic Documents Group

You can synchronize an Analytic Document using the Analytic Documents group. The Analytic Documents group is shown in the following image.



The commands are:

Sync

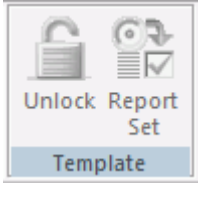
Syncs an Analytic Document.

Show

Shows the synchronization of the canvas.

Unlocking a Template for an HTML Page Using the Template Group

You can unlock a template to add controls, text, and buttons. The Template group is shown in the following image.



The commands are:

Unlock

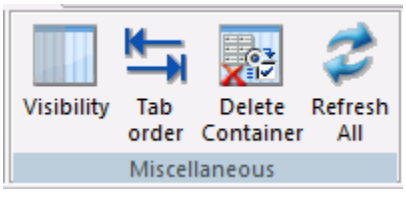
Unlocks the template that you are using and allows for full control of the page.

Report Set

In template mode, this adds a title bar and a form with controls, text, and buttons as a set.

Working With the Miscellaneous Group in the HTML Canvas

The Miscellaneous group contains the Visibility, Tab order, Delete Container, and Refresh All commands. The Miscellaneous group is shown in the following image.



The commands are:

Visibility

Toggles the visibility of hidden objects. Hidden objects are objects that have the *Visibility* property, in the Properties panel, set to *hidden*.

For more information on using the Visibility command, see *How to Toggle the Visibility of a Hidden Object* on page 136.

Tab order

Shows the order of the tabs.

Delete Container

Deletes the selected container, without deleting the container content.

Refresh All

Refreshes the HTML page.

Procedure: How to Toggle the Visibility of a Hidden Object

1. Insert an object into the HTML canvas.
2. Select the object you created, if it is not already selected.
3. In the Properties panel, change the *Visibility* property to *hidden*.

   The object on the canvas is hidden from view.
4. On the *Utilities* tab, in the *Miscellaneous* group, click the *Visibility* command.

The hidden object is now displayed on the canvas. You can click the *Visibility* command again to hide the object again. At run time, this object will be hidden.

Previewing HTML Output Using the Preview Group

The Preview group, as shown in the following image, allows you to preview HTML output in Internet Explorer, Chrome, Firefox, and Edge browsers to be sure that your content displays correctly.

The Preview Runtime option is available for Responsive HTML pages only. It uses your default browser to preview your HTML run time layout with simulated data. This may be faster than running the page if it contains large amounts of data.

Note: Alternatively, you can right-click in the HTML canvas, select *Preview*, and select a browser or Preview Runtime.

The browser preview options are:

Internet Explorer

Preview your HTML output in Internet Explorer.

Note: As of April 1, 2021, Internet Explorer is no longer supported.

Chrome

Preview your HTML output in Chrome.

Firefox

Preview your HTML output in Firefox.

Edge

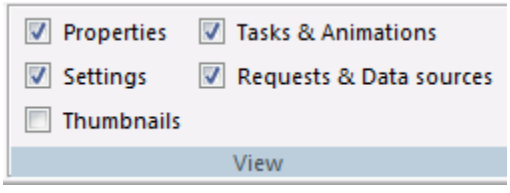
Preview your HTML output in Edge.

Preview Runtime

Available for Responsive HTML pages only. It uses your default browser to preview your HTML run time layout with simulated data.

Displaying Panels in an HTML Page Using the View Group

The View group allows you to display or hide the panels associated with an HTML Page. This group is accessible in the Utilities tab of the HTML canvas and in the Text Editor tab when accessing the Embedded JavaScript and Embedded CSS view tabs. The View group is shown in the following image.



The commands are:

Properties

Toggles the visibility of the Properties panel.

Settings

Toggles the visibility of the Settings panel.

Thumbnails

Toggles the visibility of the Thumbnails panel.

Tasks & Animations

Toggles the visibility of the Tasks & Animations panel.

Requests & Data sources

Toggles the visibility of the Requests & Data sources panel.

Using Parameters

Parameters or "variables", are a good way to make a single HTML page valuable to a wide range of end users. For example, if you have a parameter for region, users can select their own region at run time. This improves performance, saves users from scrolling through data that is not pertinent, and saves you from having to build multiple pages.

Parameter values and input controls can be created with a dynamic or static list of values:

- A dynamic list retrieves values from a specified data source when the request is run. These values are always current to your data source.
- A static list consists of a list of values you supply. These values do not change unless you change them.

- An Analytic Document control lists Analytic Document values that mimic In-Document Analytic menu items.

  Note: The Analytic Document controls cannot be associated to any parameters in the layout. This type of control can only be associated with an Analytic Document in the layout.

- The TOC control list gives you the ability to integrate a report with a Table of Contents and On Demand Paging in the HTML canvas.

  Note: TOC controls cannot be associated to any parameters in the layout. This type of control can only be associated with certain input controls in the layout.

Working With the Parameters Tab

The Parameters tab enables you to create and modify parameter values, input controls, and customize parameter conditions. You can also bind parameters to controls and chain controls to one another. The Parameters tab consists of the following components:

- Input control objects.

  You may select the input control object to view and edit the settings of the control.

  - Creating an input control from the Design view prompts you to create a bound parameter on the Parameters tab.
  - Editing an input control, which is inserted when setting input controls for new parameters.
  - Autoplay controls without associated input controls on the page, also known as virtual controls, can only be accessed at design time from the Parameters tab. They are represented by a dashed hexagon.

- Add new parameters.

  Right-click anywhere on the Parameters tab to add a new parameter.

  Note: Manually adding a parameter creates an unbound parameter.

- Rearrange the order of objects.

  When you move a parameter on the Parameters tab, a property value is automatically set. On the Properties panel, under Document properties, the *Parameters tab: auto arrange* property is automatically set to *No* when you move a parameter. This means the parameter will remain where you moved it after you navigate away, then navigate back to the Parameters tab.

You can rearrange the order of parameters and controls on the Parameters tab, which will automatically set the *Parameters tab: auto arrange* property value to *No*. This means that if you navigate away from the Parameters tab, and then return to it, the objects remain in the position you placed them. Additionally, this means that the Auto Arrange Objects option, which is selected in the HTML tab of the Developer Workbench Options dialog box by default, will be overridden.

You can change the *Parameters tab: auto arrange* property value to *Yes*, to undo any moves that you made previously, and to ensure that the objects on this tab use the auto arrange functionality.

To inherit the Auto Arrange Objects option value, you can set the *Parameters tab: auto arrange* property value to *<Not Set>*.

The *Parameters tab: auto arrange* property is available in the Properties panel. To access this property, you must select DOCUMENT from the list at the top of the panel.

When you select an object on the Parameters tab, the properties for that object will automatically display on the Properties panel. When you select a parameters object from the Properties panel drop-down, the object is automatically selected in the Parameters tab.

- Refresh unresolved parameters.

  All parameters on the Parameters tab are parsed every two minutes to check if any are unresolved. If there are, their surrounding polygon is colored red. If you want to check for unresolved parameters on demand, right-click and select *Refresh unresolved*.

- Binding controls and parameters.

  Input controls and parameters can be bound and unbound from the Parameters tab.

  You may bind a parameter to an input control, or bind an input control to a parameter.

  - Binding a parameter to a control makes it an incoming parameter that will populate the control. Drag a parameter object to a control object on the Parameters tab.

  - Binding a control to a parameter will populate the parameter. Drag a control object to a parameter object on the Parameters tab.

- Chain one control to another.

  Chaining will populate controls based on the selected value from the prior control in the chain. You can chain static and dynamic controls, link or unlink parts of a chain, and create conditions on links in a chain. Chains are represented by lines connecting control objects on the Parameters tab. By clicking the arrow head in a link of a chain, the Settings panel enables you to modify and set properties and conditions of the chain.

Note: Chaining is applicable only for controls, not parameters.

Adding a New Unbound Parameter

An unbound parameter is useful when passing a parameter value used on another page. You may also bind the new parameter to a control to create an incoming parameter, or bind a control to the parameter.

Procedure: How to Add a New Unbound Parameter

The following steps describe how to add a new parameter:

1. Right-click anywhere on the Parameters tab and select *Add parameter*.

   Enter the parameter value information. Options are *Single select*, *Multiselect OR*, and *Multiselect AND*.

2. If using single value, select *Single select*.

   Note: Single select is the default option when adding a new parameter.

   a. Enter the Selected Value to be assigned to a single variable.

   b. Enter the name for the parameter in the Name field, or keep the default name.

   c. Optionally, you may use the Format field to define the format of the parameter, such as A20, or D12.2.

      If this field is left blank, it automatically applies the Alphanumeric format to the value field.

3. If using a multiselect value, select *Multiselect OR* or *Multiselect AND*.

   The Value, Display, and Selected columns appear.

   a. Click the *New* button to enter a list of static values.

   b. In the Value column, enter the value to be passed to the selected parameter.

   c. In the Display column, enter the text that represents the parameter value in the control the user views.

   d. In the Selected column, check the box for the value you want to be selected by default. More than one value can be selected.

   Repeat these steps until the list contains all of the values you want to include.

4. Optionally, you may select values and click the *Delete* button to eliminate any values, and use the up and down arrows to rearrange the order of the values.

5. To modify the parameter value, right-click the parameter on the Parameters tab and select *Settings* to make your edits.

Tip: You may also use the Undo and Redo buttons located on the Standard toolbar. Note that undo/redo treats the entire Settings panel as one action.

6. Optionally, bind a control to a parameter to populate the parameter. Select the center of the parameter name object and drag the parameter to the center of the control object.
7. Optionally, bind the new parameter to a control to create an incoming parameter. Select the center of the control object and drag the control to the center of the parameter object.

Procedure: How to Link an Unbound Parameter to a Procedure

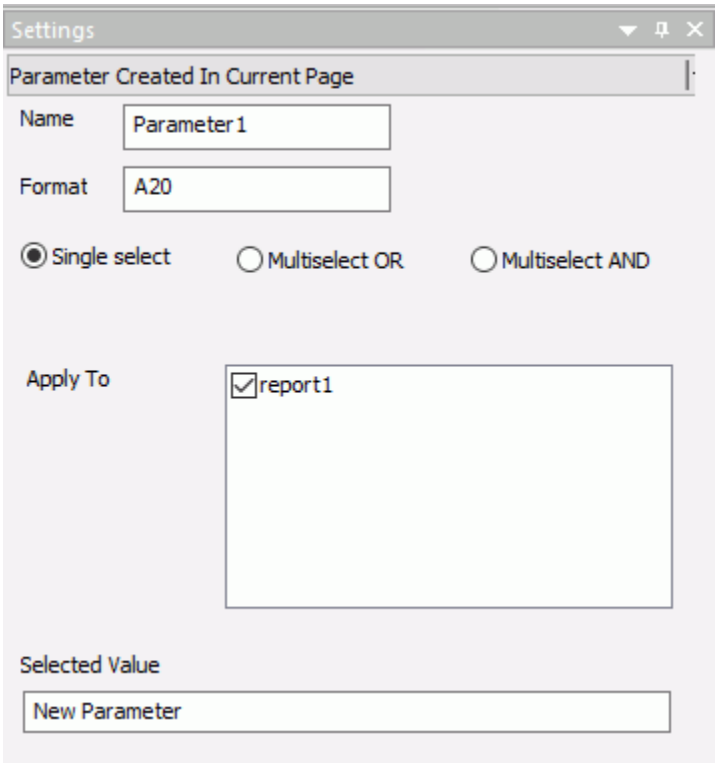
The following steps describe how to link an unbound parameter to a procedure. As an example, a heading is added to a report in the steps below.

1. Create a new HTML page.
2. From the Requests & Data Sources panel, click the *New* drop-down arrow and select *External Request - Db2 Web Query Procedure for i Using InfoAssist*.
3. Select any simple or parameterized procedure.

   Note: In this example, we are using a report.
4. Insert a frame in the canvas.
5. In the Tasks & Animations panel, select the *load* task.
6. Click the arrow in the Requests/Animations section, select *Run Request* and select your procedure.
7. Select the *Target type* as *Frame*.
8. Select the frame from the *Target/Template Name* drop-down list.
9. In the Requests & Data Sources panel, right-click *Parameters Created in Current Page* and select *New Parameter*.

   Note: You can also create a new unbound parameter as described in *Adding a New Unbound Parameter*.
10. Navigate to the Parameters tab and drag *Parameter1* outside the Unbound Parameters box.
11. Select *Parameter1*, and in the Settings panel, enter *A20* in the *Format* field.

12. Select the check box in the *Apply To* field next to the procedure you want to link, as shown in the following image.



13. Enter an alphanumeric value in the *Selected Value* field.
14. Open the procedure.
15. From the *Report* group on the *Report* tab, use the Header & Footer option to add a Report Header.
16. Enter *&Parameter1* in the Report Heading text box.
17. Run the HTML page.

    Notice that the value entered in the Settings panel for Parameter1 is displayed in the report heading.

Creating a Static List of Values

When creating a list of static values, you can select from the following options:

- Add ignore value

- ❑ Add everything value
- ❑ Add ‘No Selection’ value
- ❑ Use values from procedure
- ❑ Use values from external file
- ❑ Use values from Library
- ❑ Use values from Library (include versions)

When the options are added to the Value list, the display text can be customized, but the value cannot be changed.

You may create an unbound static parameter, an incoming static parameter (a parameter that is bound to a control), or a control that is bound to a static parameter.

Reference: Settings Panel (Incoming Static Parameter and Unbound Control)

The Settings panel appears when creating or editing a static value on the Parameters tab.

The options available depend on the type of static value.

The Settings panel contains the following fields and options when *Static* is selected as the Data type.

Data type

Determines whether values are obtained from a static or dynamic list, an Analytic Document, or table of contents.

Static. Uses a static list of values you supply. A list of static values can also be created in the Report canvas.

Static values

Is a list of supplied values for a static list.

Value. The value to be passed to the selected parameter.

Display. The text that represents the value in the control the user views. Press the Ctrl + Shift keys to add a value to the Display field.

Selected. The value to act as the default value. If the control is multiselect, more than one value can be selected.

New. Creates a new value.

Delete. Deletes a supplied value from the list.

Move Up. Moves the selected value up in the list.

Move Down. Moves the selected value down in the list.

Send display value

Select this option to send the display value, rather than the actual data, to the parameter.

Values are procedures names

Select this option to have a control populated with procedure names, so that when a value is selected, that procedure executes. The Value column is the procedure name itself and cannot be edited. The Display column is editable.

Reference: Settings Panel (Unbound Parameter)

The Settings panel appears when creating or editing a static value on the Parameters tab.

The options available depend on the type of static value.

The Settings panel contains the following fields and options when adding an unbound parameter with Single select. Single select is the default option when adding a new parameter.

Name

The default name assigned to the parameter. Optionally, you may enter a new name for the parameter.

Format

The Format field defines the format of the parameter, such as A20, or D12.2.

This field is optional. If this field is left blank, it automatically applies the Alphanumeric format to the value field.

Selected Value

Enter the selected value to be assigned to the parameter.

Static values

Is a list of supplied values for a static list.

Value. The value to be passed to the selected parameter.

Display. The text that represents the value in the control the user views. Press the Ctrl + Shift keys to add a value to the Display field.

Selected. The value to act as the default value. If the control is multiselect, more than one value can be selected.

New. Creates a new value.

Delete. Deletes a supplied value from the list.

Move Up. Moves the selected value up in the list.

Move Down. Moves the selected value down in the list.

Reference: Settings Panel (Bound Parameter)

The Settings panel appears when selecting a bound parameter on the Parameters tab. The Settings panel for a parameter is read-only, and displays the values for the bound control.

The Settings panel contains the following read-only values:

Value

Shows the selected value for the static parameter data.

Display

Shows the static parameter display value.

Procedure: How to Add a New Static Value

The steps below describe how to manually add a new static value.

You may create an unbound static parameter, an incoming static parameter (a parameter that is bound to a control), or a control that is bound to a static parameter.

1. Create a new parameter.
   a. Right-click anywhere on the *Parameters* tab and select *Add parameter*.
   b. Enter the parameter value information in the Settings panel. Options are *Single select*, *Multiselect OR*, and *Multiselect AND*.
   c. If using single value, select *Single select*.

      Single select is the default option when adding a new parameter.

      - ❑ Enter the Selected Value to be assigned to a single variable.
      - ❑ Enter the name for the parameter in the Name field, or keep the default name.
      - ❑ Optionally, you may use the Format field to define the format of the parameter, such as A20, or D12.2.

        If this field is left blank, it automatically applies the Alphanumeric format to the value field.

   d. If using a multiselect value, select *Multiselect OR* or *Multiselect AND*.

The Value, Display, and Selected columns appear.

- ❏ Click the *New* button to enter a list of static values.
- ❏ In the Value column, enter the value to be passed to the selected parameter.
- ❏ In the Display column, enter the text that represents the parameter value in the control the user views.
- ❏ In the Selected column, check the box for the value you want to be selected by default. More than one value can be selected.

  Repeat these steps until the list contains all of the values you want to include.
- ❏ Optionally, you may select values and click the *Delete* button to eliminate any values, and use the up and down arrows to rearrange the order of the values.

An unbound static parameter is useful when passing a parameter value used on another page. You may also bind the new parameter to a control to create an incoming parameter, or bind a control to the parameter.

2. Bind the new parameter to a control.

   Binding a new parameter to a control creates an incoming parameter. An incoming parameter is a static parameter that is bound to a control. The parameter value will populate the control.

   a. Select the *Design* tab and create an input control. For example, insert a list box or a drop-down list.
   b. Click the *Parameters* tab.

      The Settings panel appears for the control.
   c. Select the center of the parameter name object and drag the parameter to the center of the control object.
   d. To unbind the parameter, select the arrow head on the line, so that the line is bold, right-click, and select *Break binding*.
3. Create a control that is bound to a parameter.

   Create a control with static values and bind the control to a parameter to populate the parameter with the control values.

   a. From the Design view of the HTML canvas, select a control.

      The pointer changes into a crosshair.
   b. Drag the crosshair to create the control and adjust it to the size you want.

c. Click the *Parameters* tab.

The Settings panel appears for the control.

d. Select *Static* as the Data type.

Static is selected, by default.

e. Create the parameter values for the control:

- ❑ In the Value column, enter the value to be passed to the control.
- ❑ In the Display column, enter the text that represents the static parameter value in the control the user views.
- ❑ In the Selected column, check the box for the value you want to be selected by default. More than one value can be selected.

  Repeat these steps until the list contains all of the values you want to include.
- ❑ Optionally, you may select values and click the *Delete* button to eliminate any values, and use the up and down arrows to rearrange the order of the values.

f. Bind the new control to a parameter: Select the center of the control object and drag the control to the center of the parameter object.

g. To unbind the control, select the arrow head on the line, so that the line is bold, right-click, and select *Break binding*.

h. To change the default type of control, right-click the control object on the Parameters tab or the Design tab and select *Set Control Type*.

The options are Calendar, Check box, Drop down list, Hidden, List box, Radio button, Text Area, Edit box, Single source Tree control, and Multi source Tree control.

4. To modify the static value, right-click the control, or parameter, on the Parameters tab and select *Setting*s to make your edits.

Procedure: How to Add an Ignore Value

The add ignore value option sends _FOC_NULL to the server at run time and is intended for use with complex applications. The add ignore value option is available for Multiselect OR and Multiselect AND static parameters.

1. From the HTML canvas, use controls to supply parameter values for a report.

   A report with parameters requires that you to select values (at run time) in order to generate the output.

2. Click the *Parameters* tab.

   The parameters associated with each control can be controlled with the Parameters tab. The properties of a control can be controlled with the Properties tab.

3. Select a multiselect control object from the Parameters tab.

   The Settings panel opens.

4. Select *Add ignore value* from the Static values drop-down list.
5. Optionally, select *Send display value* to send the display value, rather than the actual data, for the parameter values in the report.
6. Close the Settings panel.
7. Run the HTML page and select the *Ignore All* value to ignore the parameter values.

Note: Sending _FOC_NULL to a procedure will result in any clause of that procedure that uses that variable to be ignored.

Procedure: How to Add an Everything Value

The add everything value option uses JavaScript to send every value present in the parameter list to the server at run time. The add everything value option is available for Multiselect OR and Multiselect AND static parameters.

The add everything value option is not available for a Double List Control.

1. From the HTML canvas, use controls to supply parameter values for a report.
2. Click the *Parameters* tab.
3. Select a multiselect control object from the Parameters tab.

   The Settings panel opens.

4. Select *Add everything value* from the Static values drop-down list.
5. Optionally, select *Send display value* to send the display value, rather than the actual data, for the parameter values in the report.
6. Run the HTML page and click the *Select All* parameter value to view all the parameter values.

Procedure: How to Use Values From a Procedure

The use values from procedure option is the default option, which populates the static list with field names predefined in the procedure. The use values from procedure option is available for Multiselect OR and Multiselect AND static parameters, and when adding static field values from the Report canvas.

1. From the HTML canvas, use controls to supply parameter values for a report.
2. Click the *Parameters* tab.
3. Select a multiselect control object from the Parameters tab.

   The Settings panel opens.
4. Select *Use values from procedure* from the Static values drop-down list.

   The field names from the procedure appear in the Settings panel.
5. Optionally, select *Send display value* to send the display value, rather than the actual data, for the parameter values in the report.
6. Run the HTML page and select the parameter values from the procedure.

Procedure: How to Import Values From an External File

The import values from an external file option enables you to use a local external file to provide values for the parameter. The import values from an external file option is available for Multiselect OR and Multiselect AND static parameters.

1. From the HTML canvas, use controls to supply parameter values for a report.
2. Click the *Parameters* tab.
3. Select a multiselect control object from the Parameters tab.

   The Settings panel opens.
4. Select *Use values from external file* from the Static values drop-down list.

   The Open File dialog box appears.
5. Select a text file from your local machine and click *Open*.

   The external file can be a file with single values on each line, or two values per line, comma-delimited.

   For example, in the following text file, *BOS* is the data value and *Boston* is the display value.

   The imported values are loaded into the Static values area of the Parameters tab.

If there is only one value on the line in the text file, the value will populate both the data value and the display values.

6. Run the HTML page to see the imported values for the selected parameter.

Creating a Dynamic List of Values

Dynamic values are available, by default, if a parameter used in the procedure is associated with the selected control. A dynamic list retrieves values from a specified data source when the request is run.

Reference: Settings Panel (Dynamic Values)

The Settings panel appears when creating or editing a dynamic parameter on the Parameters tab.

The Settings panel contains the following fields and options when Dynamic is selected as the Data type:

Data type

Determines whether values are obtained from a static or dynamic list, an Analytic Document, or table of contents.

Dynamic uses a list of values retrieved from a selected data source when the request is executed. This is the default if you use an Accept clause in a Master File to create an amper variable parameter within a procedure.

Default

Is the data source from which the values will be retrieved. This is the default value when *Dynamic* is selected as the Data Type.

Explicit (Requests panel)

Is the existing procedure that will be called.

You may modify the explicit procedure directly from the Settings panel on the Parameters tab. If you modify the request, you can save the explicit procedure and overwrite the original request.

Data Source

Is the data source from which the values will be retrieved.

Value from field

Is the data source field from which the values will be retrieved.

Display from field

Is the text that represents the parameter value in the control the user views.

There should be a relationship between the Value from field and the Display from field. The Display from field is user-friendly text corresponding to the Value from field.

Sort

Clicking *Sort* enables you to set the sort order for displaying values in dynamic list controls. This option is useful when you want to sort each control independently of the others.

By default, the request retrieves dynamic display values from the BY sort field in the request. The results display values based on the value field.

Sort by

When Sort is enabled, you may sort the display value by the Value field or the Display field selected from the Settings panel. The default is Value field.

Sort order

When Sort is enabled, you may select the sort order as Ascending or Descending. The default sort order is Ascending.

Values are procedures/html files

Indicates that the values in the control are procedures or HTML pages. You can create a task using the Run Procedures/HTML From Control request option in the Tasks & Animations panel to run the file selected in the control.

Check for duplicate values

When creating a dynamic list of values for a report, you may remove duplicate values from input controls.

Add "All" Option

Adds the option to select all data source values to the control. Alternate text can be substituted for All using the text field to the right.

Note: If the dynamic list contains only one value, the All option does not display in the control.

Add ‘No selection' option

Optimizes performance by populating a chain one control at a time, instead of all the controls when the page initially loads. Selecting the Add ‘No selection’ option enables you to populate controls when necessary.

Cache run time data

When adding dynamic parameters to the HTML page, input controls retrieve data through procedures. Select this option to cache the run-time data for the selected input control. This setting is off by default.

This setting overrides the Default caching option from the HTML Page tab, which is located in the Developer Workbench Options dialog box.

Limit values returned

Indicates that a specific number of field values will be retrieved from the data source. The specific number of fields is selected with the menu to the right.

Use last modified filter only

Selecting this option ensures that if two controls are chained into a third control, only the value selected last will be used to populate the third control.

Send display value

Select this option to send the display value, rather than the actual data, to the parameter.

Selected Value

Enter the value to be selected as the default value whenever the procedure is run.

Reference: Settings Panel (Bound Parameter)

The Settings panel appears when selecting a bound parameter on the Parameters tab. The Settings panel for a parameter is read-only and displays the values for the bound control.

The Settings panel contains the following read-only values:

Data Source

Shows the selected Master File for the parameter data source.

Value from

Shows the value field for the dynamic parameter data.

Display from

Shows the dynamic parameter display field.

Multiselect

Shows OR or AND, if there is a Multiselect OR or Multiselect AND dynamic parameter.

Selected Value

Shows the selected value, if there is one assigned to the variable.

Procedure: How to Create a Dynamic Value

The steps below describe how to create a dynamic list of values.

You may need to create a dynamic control that is bound to a parameter. Creating a control with dynamic values and binding the control to a parameter will populate the parameter with the control values.

1. From the Design view of the HTML canvas, select a control from the Controls submenu of the Insert menu. For example, insert a list box or drop-down list.

   The pointer changes into a crosshair.

2. Drag the crosshair to create the control and adjust it to the size you want.
3. Click the *Parameters* tab.

   The Settings panel opens.

4. Select *Dynamic* as the Data type.

   The dynamic value options appear.

5. Create the dynamic values for the control.
   a. If you are using a default procedure to supply dynamic values, follow the steps below.

      - Click the *browse (...)* button adjacent to the *Data Source* drop-down list.

        The Open File dialog box appears.

      - Select the Master File name and click *OK*.
      - Click the *browse (...)* button adjacent to the *Value from* field.

        The Object Inspector opens with the field names from the selected Master File.

      - Double-click a field name to add it to the Value from field.

        Tip: You may also use the Object Inspector icons to select a field and close the Object Inspector. The green icon is OK, the red icon is Cancel, double-clicking a value will select the value and close the dialog box without using any button, and pressing the Esc key will cancel the dialog box without using any button.

      - Optionally, you may click the *browse (...)* button adjacent to the *Display from* field to select a different field name for the Display field. (You may also use the Object Inspector icons to select a field and close the Object Inspector).

b. If you are using an explicit procedure to supply dynamic values, follow the steps below.

- ❏ Select *Explicit (Requests Panel)*.
- ❏ Select a procedure from the *Requests* drop-down list.

  The parameter names from the procedure are automatically added to the *Value from* field and *Display from* field.
- ❏ Optionally, you may click the *browse (...)* button adjacent to the *Value from* field and *Display from* field to select different field names.

  Tip: You may also use the Object Inspector icons to select a field and close the Object Inspector. The green icon is OK, the red icon is Cancel, double-clicking a value will select the value and close the dialog box without using any button, and pressing the Esc key will cancel the dialog box without using any button.

  Only the parameter names from the explicit procedure will be available for selection.

6. Optionally, select *Add "All" option* to add the ability to select all data source values to the control.
7. Optionally, select *Add 'No selection' option* to optimize performance by populating a chain one control at a time, instead of all the controls when the page initially loads.
8. Optionally, select *Cache run time data* to cache the run time data for the selected input control.
9. Optionally, select *Limit values returned*, and select or type the number of field values you want to retrieve from the data source in the box to the right of this option.
10. Optionally, click the *Sort* option to enable and select the sort order options for displaying values in a dynamic list control.
11. Optionally, select *Check for duplicate values* to remove any duplicate value entries from the input control at run time.
12. Bind the new control to a parameter. Select the center of the control object and drag the control to the center of the parameter object.
13. To unbind the control, select the arrow head on the line, so that the line is bold, right-click and select *Break binding*.
14. To change the default type of control, right-click the control object on the Parameters tab or the Design tab and select *Set Control Type*.

    The options are Calendar, Check box, Drop down list, Hidden, List box, Radio button, Text Area, Edit box, Single source Tree control, and Multi source Tree control.

15. To modify the dynamic value, right-click the control and select *Settings* to make your edits.

Procedure: How to Create Dynamic Parameters by Adding a Filter

You may create new dynamic parameters by adding a filter to a report or graph component in the layout.

1. To create dynamic parameters for your report or graph in the Design view, right-click the report or graph object and select *Add a filter* from the shortcut menu.

   The Filter options dialog box opens.

2. Select the field to be used for the parameter, the Multiselect option, and click *OK*.

   The New Parameters dialog box appears.

   You may select a control type for the parameter from this dialog box, or adjust them later using the Settings panel on the Parameters tab.

   If the New Parameters dialog box does not appear, ensure that *Show New Parameters dialog* is selected from the HTML Page tab. To access the HTML Page tab, select *Options* from the Window menu to open the Developer Workbench Options dialog box. From the Developer Workbench Options dialog box, select the HTML Page tab.

3. Click *OK* to close the New Parameters dialog box.
4. The filter appears above the report or graph object.

   Repeat this procedure for each additional parameter for the report or graph.

Procedure: How to Sort the Dynamic List of Values

This option is useful when you want to sort each control independently of the others.

Note: If sort options are not selected, the request retrieves dynamic display values from the BY sort field in the request, and the results display values based on the value field. Sort options are not selected, by default.

1. Select a dynamic control from the Parameters tab.

   The Settings panel opens.

2. Select *Sort* to enable the sort options.

   You may select the Sort by and Sort order options for the control.

3. Select the Sort by options:

   - Sort by Value sorts the value by the field name from the Value field. This is the default Sort by selection.

- Sort by Display sorts the value by the field name from the Display field.

4. Select the Sort order options:

   - Sort order Ascending sorts the value from lowest to highest. This is the default Sort order selection.
   - Sort order Descending sorts the value from highest to lowest.

5. Run the HTML page to see the sort results.

Procedure: How to Check for Duplicate Values

When creating a dynamic list of values for a report, you may remove duplicate values from input controls. This is useful if you are using your own procedure that does not use a structured data source.

The Check for duplicate values option is turned off, by default.

1. From the HTML canvas, create an input control with a dynamic list of values.
2. Select the input control and click the *Parameters* tab.

   The Settings panel opens for the input control.

3. Select the *Check for duplicate values* check box.

   The Check for duplicate values option is only available when creating a dynamic list of values for an input control.

4. Save and run the HTML page.

   The input control removes duplicate value entries.

Automatically Populating Fields With Dynamic Values

When the name of a dynamic parameter matches a corresponding field name in a data source, the HTML canvas automatically populates the field name values for the parameter.

The data source is populated by a default based on the first data source specified by a TABLE FILE or GRAPH FILE command. The data source field is populated for the Value and Display fields in the Settings panel of the Parameters tab (when Dynamic is selected as the Data type). This generates a layout report that is ready to run as long as the parameter names match the field names.

Example: Automatically Populating Fields With Dynamic Parameter Values

When the following report request is called from the HTML canvas with a push button control, the Settings panel for the PRODUCT parameter on the Parameters tab is automatically populated to dynamically retrieve the values of the PRODUCT field.

```
TABLE FILE GGSALES
SUM UNITS
BY PRODUCT WHERE ( PRODUCT EQ '&PRODUCT.Product:.' );
END
```

Creating a Static or Dynamic Parameter Value List

When creating a static or dynamic list of values, you may add an ALL value to the list of values and/or send the display value in a parameter.

The ALL feature allows developers to automatically add an ALL value to a list of values. An ALL value does the following:

- For dynamic parameters, the ALL feature sends a value of FOC_NONE to the Reporting Server alerting the server to bypass or ignore the parameter altogether. Ignoring the parameter would return all values in the data source.
- With static parameters, the ALL value typically uses JavaScript to return all of the values displayed in the list. This prevents you from having to select every value in the list manually. When using the ALL feature with static parameters, you can select from the following options:
  - Add ignore value. This option sends FOC_NONE to the server at run time, alerting the server to bypass or ignore the parameter altogether. It is intended for complex applications.
  - Add everything value. This option uses JavaScript to send every value present in the control list to the server at run time. The Add everything value option is only available with a Multiselect OR variable type.

When using a Dynamic or Analytic Document Data type, you may enter the value(s) to be selected as the default value whenever the procedure is run.

Note: You may also enter the selected value when adding a new unbound parameter on the Parameters tab.

Procedure: How to Send the Display Value for Static and Dynamic Controls

From the HTML canvas, you may send the display value, rather than the actual data, to the parameter. The display value can also be used for headings and footings in the report output.

1. From the HTML canvas, import or create a report that contains a parameter.

   When importing a report with parameters, the New Parameters dialog box appears prompting you to create the control type.

   The report and control is added to the HTML canvas.

2. To send the display value for the parameter selection, select the control object (for example, select listbox*n*) in the Design view and click the *Parameters* tab.

3. Navigate to the Settings panel.

   Note: The options available in the Settings panel vary, depending on the type of values (static or dynamic) you are creating.

4. For a static list of values, the Value, Display, and Selected columns appear on the Settings panel.

   - ❏ In the Value column, enter the value to be passed to the control.
   - ❏ In the Display column, enter the text that represents the static parameter value in the control the user views.
   - ❏ In the Selected column, check the box for the value you want to be selected by default. More than one value can be selected.

     Repeat these steps until the list contains all of the values you want to include.
   - ❏ Select *Send display value*.

     Tip: You may also update the display values from the Variable Editor dialog box in the Report canvas.
   - ❏ Optionally, you may select values and click the *Delete* button to eliminate any values, and use the up and down arrows to rearrange the order of the values.
   - ❏ Click the *Design* tab to view the display values in the control object of the HTML canvas.

5. For a dynamic list of values, the Value from field and the Display from field appear on the Settings panel.

   - ❏ Add the request to the Requests & Data sources panel.
   - ❏ Select *Explicit (Requests Panel)*.

- ❑ Select the procedure from the Requests drop-down list.
- ❑ Click the ellipsis button from the Value field.

  The Object Inspector opens with the field names from the selected Master File.
- ❑ Double-click a field name to add it to the Value field.

  The selected field is automatically added to the Display field.
- ❑ Optionally, you may click the Display field ellipsis button to select a different field name for the Display field.
- ❑ Optionally, select *Add "All" option* to automatically add an all value to a list of parameter values.
- ❑ Optionally, select *Add 'No selection' option* to optimize performance by populating a chain one control at a time instead of all the controls when the page initially loads.
- ❑ Select *Send display value*.

  When Send display value is selected, the &Variable gets the display value, instead of the actual value.

6. Add &*Variable_TEXT* to the report heading or footing.

   Note: If the report procedure uses it, &*Variable_TEXT* will always be passed, regardless of whether Send display value is selected.

   The heading as Display Value: &STCD_TEXT, where Store Code (STCD) is the variable name.

   Tip: This is different from adding the actual value from the data source, where clicking the variable name would add <STCD to the report heading.
7. Save and close the report to return the HTML canvas.

   When you create the Text variable (*&Variable_TEXT*), you are not prompted to set a control type for this variable when returning to the HTML canvas, as no control type is needed.
8. Run the HTML page.
9. Select the parameter for the report and run the report.

   The display value is shown in the report heading.

   Note: The HTML canvas passes &Variable_TEXT if the report procedure uses it, regardless of whether the send display value check box is checked or unchecked.

Procedure: How to Use Procedure Names as Values

The *Values are procedures* names option lets you populate a control with procedure names or HTML file names. When that procedure name is clicked, the procedure executes.

1. Create an HTML page that contains a listbox, a push button, and a report.

   Note: In this procedure a listbox is used, however, the following controls are also able to use the Values are procedures names option: double list, drop-down, radio button, and check box.

2. Select the listbox to bring up the Settings panel.

   Note: If the Settings panel does not open, select *View* and click *Settings*.

3. Select *Static* as the Data type.
4. At the bottom of the Settings panel, select *Values are procedures/html files*.
5. Click the *New* button and select procedures from your directory.

   Note: You can add multiple procedure names to the Settings panel by multi-selecting procedures while in the Open File dialog box.

6. Once the procedures have been added to the Settings panel, edit the display names of the procedures by double-clicking the display contents if they are not already highlighted.
7. Right-click the button you created and click *Create Hyperlink*.

   The Hyperlink Properties dialog box opens.

8. Create a hyperlink that opens a selected procedure from a control in the report frame created earlier.
   a. For the Action, select *Procedures from control* from the drop-down list.

      This option coincides with the Values are procedures names option found in the Settings panel. This option will point to an entire procedure for the hyperlink, rather than a simple value. This option is only available when a control on the HTML page is using the Values are procedures names option.

   b. Select *listbox1* as the Source.

      The source can be different if you use a different control. For example, combobox1, customselect1, radio1, or checkbox1.

   c. Select *Frame* as the Target Type.

      You could also select *New Window* as the target if you wanted the procedure to open in a new window.

   d. Select *report1* as the Target/Template Name.

9. Run the page.
10. Select the procedure from the listbox and click the button.

    The report is run.

Procedure: How to Use Selected Values as the Default Value

When using a Dynamic or Analytic Document Data type, you may enter the values to be selected as the default value whenever the procedure is run.

If you import a procedure (.fex) that has a dynamic prompt value, then the input box is populated with values retrieved from the data source. If the selected value is available in the data source, the values are selected, by default. If the selected value is not available in the data source, then the value(s) that you entered are ignored and the first value retrieved from the data source is selected.

1. Create or import a report that contains a parameter value.

   The report and control is added to the HTML canvas.

2. To enter the selected value to be used as the default value, select the control object (for example, select listbox*n*) in the Design view and click the *Parameters* view tab.

   The Settings panel opens.

   Note: The options available in the Settings panel vary, depending on the type of values (static or dynamic) you are creating.

3. For a dynamic list of values, the Value from field and the Display from field appear on the Settings panel.

   a. If using a default procedure to supply dynamic values, follow the steps below.

      - Select *Default*.
      - Click the ellipsis button adjacent to the Data Source field.

        The Open File dialog box appears.

      - Select the Master File name and click *OK*.
      - Click the ellipsis button adjacent to the Value from field.

        The Object Inspector opens with the field names from the selected Master File.

      - Double-click a field name to add it to the Value from field.

        The selected field is automatically added to the Display from field and the source code for the default procedure appears.

- ❑ Optionally, you may click the Display from field ellipsis button to select a different field name for the Display from field.
- ❑ Optionally, select *Add "All" option* to automatically add an all value to a list of parameter values.
- ❑ Optionally, select *Add 'No selection' option* to optimize performance by populating a chain one control at a time, instead of all the controls when the page initially loads.
- ❑ Enter the exact parameter value in the Selected Value input field, as it appears in the data source.

  Parameter values are case-sensitive.
- ❑ Optionally, you may enter more than one value by using a semicolon (;) between the values. For example, *CA;GA*.

  You may enter selected values with a semicolon (;) or a comma (,). Additionally, you may also have embedded commas or semicolons in the data values. If you have these embedded characters, you must enclose all of the values in the input area in double quotation marks ("). For example:

  ENGLAND;ITALY

  ENGLAND,ITALY

  “ENGLAND”,”IT,ALY”

  Note that even though one value in the last set has the embedded comma, both values need to be enclosed in double quotation marks (").

b. If using an explicit procedure to supply dynamic values, follow the steps below.

- ❑ Select *Explicit (Requests Panel)*.
- ❑ Select the procedure from the *Requests* drop-down list.

  The parameter names from the procedure are automatically added to the Value from field and Display from field, and the source code for the explicit procedure appears.
- ❑ Optionally, you may click the Value from field and Display from field ellipsis button to select a different field name or to type the field name manually.

  Note: Only the parameter names from the explicit procedure will be available for selection.

- ❑ Optionally, select *Add "All" option* to automatically add an all value to a list of parameter values.
- ❑ Enter the exact parameter value in the Selected Value input field, as it appears in the data source.

  Parameter values are case-sensitive.
- ❑ Optionally, you may enter more than one value by using a semicolon (;) between the values. For example, *CA;GA*.

  You may enter selected values with a semicolon (;) or a comma (,). Additionally, you may also have embedded commas or semicolons in the data values. If you have these embedded characters, you must enclose all of the values in the input area in double quotation marks ("). For example:

  ENGLAND;ITALY

  ENGLAND,ITALY

  “ENGLAND”,”IT,ALY”

  Note that even though one value in the last set has the embedded comma, both values need to be enclosed in double quotation marks (").

4. For an Analytic Document list of values, the Available Analytic Documents, Menu Option Types, and Common Columns appear on the Settings panel.

   - ❑ Select one or more Analytic Documents from the list of Available Analytic Documents. The selected report will be bound to the Analytic Document control in the layout.

     When an Analytic Document is selected, the Refresh for Analytic Documents option, in the Settings panel, is enabled, by default.
   - ❑ Select the Menu Options Types for the Analytic Document control to sort, filter, list or select columns, and/or change presentation styles of the bound Analytic Document and the associated report and graph objects synchronized to the Analytic Document.
   - ❑ Optionally, select *Add "All" option* to automatically add an all value to a list of parameter values.
   - ❑ Enter the exact parameter value in the Selected Value input field, as it appears in the data source.

     Parameter values are case-sensitive.
   - ❑ Optionally, you may enter more than one value by using a semicolon between the values. For example, *CA;GA*.

You may enter selected values with a semicolon or a comma. Additionally, you may also have embedded commas (,) or semicolons (;) in the data values. If you have these embedded characters, you must enclose all of the values in the input area in double quotation marks ("). For example:

ENGLAND;ITALY

ENGLAND,ITALY

“ENGLAND”,”IT,ALY”

Note that even though one value in the last set has the embedded comma, both values need to be enclosed in double quotation marks (").

5. For a single-select unbound parameter, the Selected Value input field appears on the Settings panel.

   - Enter the exact parameter value in the Selected Value input field, as it appears in the data source.

     Parameter values are case sensitive.

   - Optionally, you may enter more than one value by using a semicolon between the values. For example, *CA;GA*.

     You may enter selected values with a semicolon or a comma. Additionally, you may also have embedded commas or semicolons in the data values. If you have these embedded characters, you must enclose all of the values in the input area in double quotation marks. For example:

     ENGLAND;ITALY

     ENGLAND,ITALY

     “ENGLAND”,”IT,ALY”

     Note that even though one value in the last set has the embedded comma, both values need to be enclosed in double quotation marks (").

   Even though it is a Single select parameter, multiple values are specified and selected.

6. Click the *Run* button to run the report with the selected value parameters.

   The report output appears.

   The selected value, if available from the data source, is automatically selected (highlighted) in the parameter list.

If the selected value is not available in the report results, then the value that you entered is ignored and the first value retrieved from the data source is shown.

Viewing Object Attributes

The Properties panel shows the attributes of the object or objects selected on the canvas. For example, if you select a hyperlink object, the Properties panel shows the different attributes associated with a hyperlink, such as Enable Dragging. General attributes for the entire HTML page are shown when no object is selected. The Properties panel is available when you are using the HTML canvas. The attributes can be sorted by category or alphabetically.

You can also select *Events*. When *Events* is selected, a list of all available JavaScript events that can be used in conjunction with an object appears. The events that are available change depending on what type of object is selected. For example, a report object has different events available than a button object. When no object is selected, events for the HTML page are displayed. Double-clicking an event will create a function block for the selected object, using that event. You can view the created functions in the Embedded JavaScript and Embedded CSS tabs, where you can type the JavaScript code to execute when the selected event occurs.

For more information on customizing the Properties panel, see *Customizing the Properties Panel for HTML Objects Using the Properties Toolbar* on page 167.

Color Selection

For properties that affect color, such as Background-color, Border-color, and Color, you can enter the hexadecimal value of the color you want to use directly into the property, rather than opening the Color Picker dialog box. If you use the Color Picker dialog box to select a color, the hexadecimal value of that selected color will be entered into the selected property. For more information on the Color Picker dialog box, see *Color Picker Dialog Box* on page 172.

Sizing

Size property fields for a selected object can be set using the Left, Top, Width, and Height fields. The default setting is in pixels (px), and the settings are also displayed in the Locations field. You can also select <*Not set*>, *auto*, or *inherit* in the Left, Top, Width, and Height fields, but only pixels are displayed in the Location field.

The Position field also contributes to the location of an object in the HTML canvas. The default value is absolute, but you can change it to one of the values in the list below:

- **<Not Set>.** The element is positioned relative to its first positioned (not static) ancestor element.
- **inherit.** Inherits this property from its parent element.

- **absolute.** The element is positioned relative to its first positioned (not static) ancestor element. This is the default value.
- **fixed.** The element is positioned relative to the browser window.
- **relative.** The element is positioned relative to its normal position, so setting the Left field to 20px adds 20 pixels to the elements left position.
- **static.** Elements render in order, as they appear in the document flow.

Controls

The Name or Unique Identifier attribute settings for controls appear automatically in the Parameters tab of the HTML canvas.

- When you change the Name attribute for a control, the new name automatically appears in the Parameters tab.
- Some controls do not have a Name attribute. For these controls, the Unique Identifier setting automatically appears in the Properties panel, instead.

Customizing the Properties Panel for HTML Objects Using the Properties Toolbar

The Properties toolbar contains the buttons and menus that you need to navigate and customize the contents of the Properties panel.

Categorized

Groups the attributes by category.

Alphabetical

Sorts the attributes alphabetically, without categories.

Properties

Shows the property names and values. This is the default setting. Properties and Events cannot be selected at the same time.

Events

Shows the JavaScript events that can be associated with the selected object on the canvas. Properties and Events cannot be selected at the same time.

Resizing HTML Components Using the Autosize Children Option

The Autosize Children option can resize all child components on an HTML page to automatically fit the maximum available screen space on any monitor or device where it might be viewed. For example, a single component will take up one hundred percent of the available space, while two components will take up fifty percent each.

This option is used so that when your HTML page is run on different devices, components and controls do not appear off the screen and have to be scrolled to in order to use them. To accommodate this, the orientation of components may be different on different devices. For example, a report and chart that appear side-by-side when run on a desktop monitor, may appear stacked one above the other when run on a tablet. Similarly, this difference could occur when run on a 24-inch desktop monitor and a 26-inch desktop monitor.

There are autosize Enable properties that work at both the document and component levels:

- **Autosize Children Enable.** This property is set at the document level. When set to *Yes*, it enables some or all of the child components in the document to resize automatically.

  The Autosize Children Enable property is also available at the component level for the form, group box, and panel container objects.

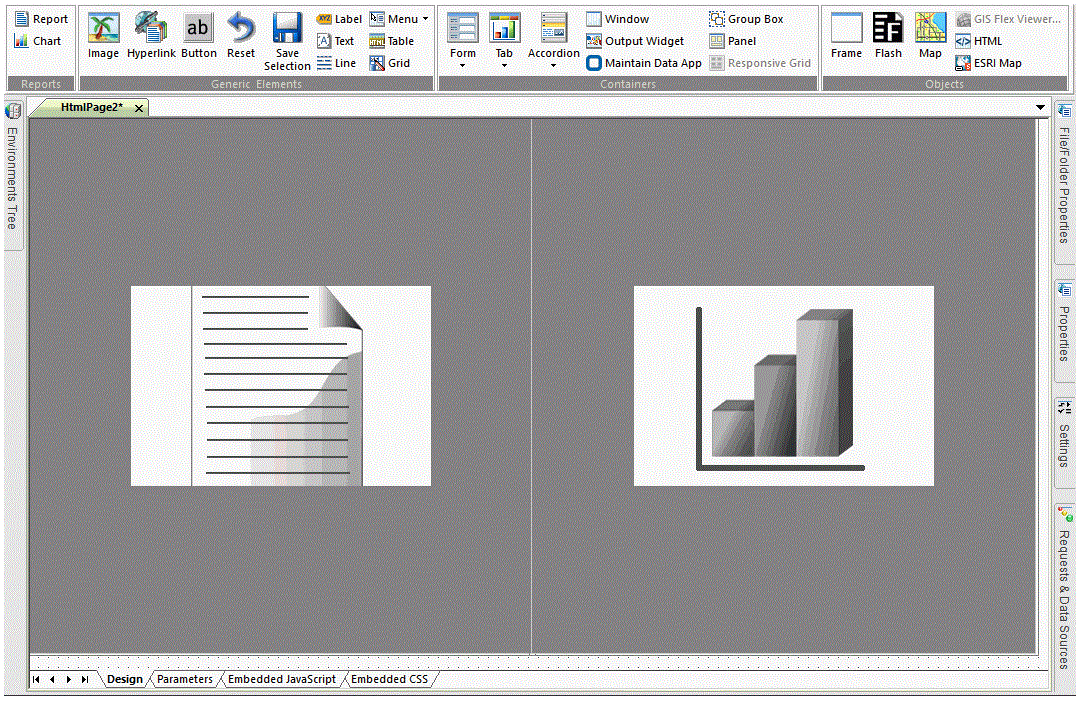
- **Autosize settings Enable.** This property is set for each component in the document. The setting for this option may be:

  - *Yes.* Enables the component to resize automatically.

    You can prevent the component from becoming too small using the Min-width and Min-height options. Type the minimum number of pixels for the width, height, or both, in the appropriate fields. When a component reaches the minimum width or height that you specified, a scroll bar appears and the size of the component does not decrease further.

  - *No.* Prevents the individual component from resizing automatically, even when the document and other components resize. Components tagged in this way are rendered first, when the HTML page runs.

The following image shows a report component and a chart component on the HTML canvas where Autosize Children is enabled. Both components share the available space equally.



You can enable Autosize Children for an existing HTML file or create a new file with the option enabled.

Procedure: How to Enable Autosize Children for an Existing HTML File

1. Open the HTML file.

   The HTML canvas opens.

2. In the Properties panel for the document, set the Autosize Children Enable property to *Yes*.
3. For each component on the page that you want to resize automatically, set the Autosize settings Enable property to *Yes*.
4. Right-click the HTML canvas and click *Update Layout*.

   The components are resized to occupy all of the space on the canvas.

Procedure: How to Create a New HTML File with Autosize Children Enabled

1. On the *Home* tab, in the *Content* group, click *HTML/Document*.

You can also create a new HTML file from the Application menu or by using the shortcut menu in the Environments Tree panel, for a folder that supports content creation.

The HTML/Document Wizard opens.

2. Navigate to where you want to create your HTML page and click *Next*.

   The Themes and Settings window of the HTML/Document Wizard opens.

3. In the Other settings area, select the *Autosize reports/charts* check box.

   Selecting this option sets the Autosize Children Enable property for the document to *Yes*.

4. Click *Finish*.

   The HTML canvas opens.

5. For each component on the page that you want to resize automatically, set the Autosize settings Enable property to *Yes*.

Working With Autosize Children

Working with Autosize Children entails some practices and conditions beyond those found in the native HTML canvas. This is because, when Autosize Children is enabled, components automatically move and resize themselves on the canvas.

Adding Components to the HTML Canvas

When you add a component to the canvas with the Autosize Children option enabled, the existing components will temporarily move out of the way so that you can draw the new one. The components will then reposition and resize automatically.

If the screen does not refresh, you can right-click the canvas and click *Update Layout*. There is a small amount of canvas space shown to the bottom and right of the screen, which allows you to access the shortcut menu. This extra canvas space is not shown at run time.

Note: If Update Layout does not resize the component automatically, check the properties. The Autosize Children Enabled property for the document must be set to *Yes*, and the Autosize settings Enable property for the component must be set to *Yes*. Components with an Autosize settings Enable property set to *No* will not resize automatically on the canvas or at run time.

To change the order of the components, drag a component to the front or to the left of another component.

The following items apply to form, group box, and panel containers:

- Forms, group boxes, and panels that contains components and controls must have the Autosize Children Enable property set to *Yes*, and the Autosize settings Enable property set to *Yes*.

- The components that are children of a form, group box, and panel have their Autosize settings Enable property automatically set to *Yes*. For controls that are children of a form, group box, or panel, you must manually set the Autosize settings Enable property to *Yes*.
- The form, group box, or panel should have the Number of Columns property set to the value you want. For example, if you want three reports or charts to be next to each other, set the value for the Number of Columns property to *3*.

If an object has the Autosize settings Enable property set to *No*, and you want that object to occupy the entire width of the available space, you must set the Width property to 100%.

Note: Reports and charts do not automatically re-execute once the page loads and the screen size changes. You must re-execute reports and charts to ensure that the content refreshes. If the output format of a chart is HTML5, the chart content refreshes automatically and you will not have to re-execute.

Procedure: How to Add Content Inside a Container

You can easily add content inside a container, such as a panel, in the native HTML canvas. However, if the Autosize Children property is enabled, the container moves when you try to add content to it. To prevent this movement, use the following steps:

1. Click to select and freeze the container to which you want to add content.
2. On the *Components* tab, in the *Generic Elements* group, choose an object, such as a text box, that you want to add to the container.
3. Click in the container again to add the element. For example, draw the text box and type the text.
4. Click the element to select it.

   In the Properties panel, set the Autosize settings Enable property to *Yes*, if you want the element to resize automatically.

   Click *No*, if you want the element to remain fixed in the container.
5. Right-click the canvas and click *Update Layout*.

   The content now appears inside the container.

Working With Containers

If you place an object that has the Autosize settings Enable property set to *No* inside a container that has the Autosize Children Enable property set to *Yes*, the object will move to the upper-left corner of the container.

When a container has the Autosize Children Enable property set to *Yes*, and a report or chart is added to that container, the Autosize Children settings property for the report or chart is automatically set to *Yes*.

When a Tab, Accordion, or Window container has the Autosize Children Enable property set to *Yes*, and multiple reports or charts are added to it, use the Number of columns property. Select the number of columns you want from the drop-down list and arrange the contents in the container, as desired. This ensures the contents will be arranged the same for all browsers. If you do not use the Number of columns property, different browsers may arrange the contents in different ways.

Reference: Color Picker Dialog Box

The Color Picker dialog box contains the Web Palette tab, Named Colors tab, System Colors tab, and Custom Color tab. From these tabs, you can pick different colors.

- **Web Palette.** You can choose a color from common web colors.
- **Named Colors.** You can choose a color from common named colors.
- **System Colors.** You can choose a color based on the colors of Developer Workbench.
- **Custom Color.** You can drag red, green, and blue sliders to create custom colors.

The hexadecimal value for any color you select or create is displayed at the bottom of the dialog box. If you want to enter a specific hexadecimal value, you must enter it using the Properties panel or the Style Composer dialog box, in the appropriate area. For example, you can enter a hexadecimal value on the Background-color property line.

Modifying Object Population Settings

In Developer Workbench, you use the Settings panel to modify the population settings of components, controls, and parameters. The Settings panel has context-sensitive sections that are displayed or hidden, depending on the object selected on the canvas. When you select an object, the settings for that object are displayed in the Settings panel.

Settings are grouped under the following sections, and the sections and settings change, depending on which component, control, or parameter is selected.

- **Manage CSS and Scripts Section**

  When no object is selected on the canvas, you can manage your cascading style sheet files and JavaScript files using the Manage CSS and Scripts section. The Settings panel displays this section by default.

  Using the CSS button and JavaScript button, you can search for cascading style sheet files and JavaScript files, located in your Db2 Web Query environment. 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.

- **Parameter Created In Current Page Section**

  To create a parameter in the current page, right-click in the Parameters tab and click *Add Parameter*. The Settings panel displays the Parameter Created in Current Page section. Type the parameter name and format. Click the radio button to specify Single select, Multiselect OR, or Multiselect AND. Type a value in the Selected Value field.

- **Dynamic Parameter Data Section**

  When a dynamic parameter is selected on the Parameters tab, the Settings panel displays the data source file, Value field, and Display field. If the value is multiselect, you can click the drop-down list in the Multiselect field and choose OR, AND, BY, ACROSS, or NONE. Next, type the Selected Value and the Preview Value.

- **Input Control Population Section**

  When a control is selected on the canvas and the Data type is set to Dynamic, you can drag fields and parameters from the Requests & Data sources panel to the Settings panel.

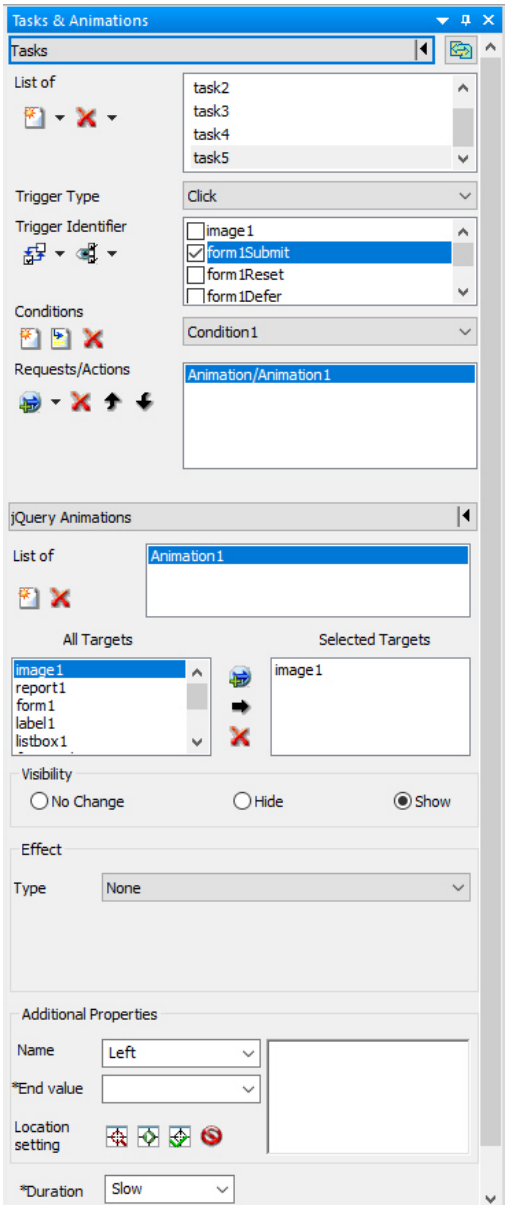
Working With the Selection To Area in the Settings Panel

The Selection to area in the Settings panel assists in directing where control and grid selections are sent. For example, you can choose to send any selections made from a drop-down control to a specific parameter.

You can click the *Use 'Requests_Data sources' panel* button to open the Requests & Data sources panel next to the Selection to area. This allows you to drag any parameter from the Requests & Data sources panel, to the Selection to area.

Using Tasks & Animations

Using the Tasks & Animations panel, you can submit requests for parsing, execute tasks when specific criteria are met, and add jQuery animations to your HTML page. The Tasks & Animations panel is shown in the following image.



For more information on customizing the Tasks & Animations panel, see *Executing jQuery Animations Using the jQuery Animations Section* on page 180.

Executing a Task Using the Tasks Section

The Tasks section of the Tasks & Animations panel allows you to execute a request when specific criteria are met. For example, you can create a task that runs an embedded report when a button is clicked.

For more information, see *How to Automatically Execute a Request When a Page is Loaded* on page 187.

Tasks

Toggle details

Opens the Tasks dialog box which displays the Tasks section of the Tasks & Animations panel horizontally, rather than vertically. Tasks are displayed in a list. This list shows the Task Name, Trigger Type, Trigger Identifier, Requests/Actions, Target Type, and Target Name, of each task, next to each other. When the Tasks dialog box is opened, the Tasks section of the Tasks & Animations panel is collapsed.

List of

A list of tasks defined in the HTML page.

New task

Creates a new task. A task is an actionable item.

Delete

Deletes a selected task. You can use the drop-down list to delete a selected task or to delete all untriggered tasks.

Trigger Type

Determines when a task should start. You can choose:

- Click
- Click link
- Double click
- Mouse down
- Mouse enter
- Mouse leave

- ❑ Mouse move
- ❑ Mouse over
- ❑ Mouse out
- ❑ Mouse up
- ❑ Blur
- ❑ Selection Changed
- ❑ Checkbox: Check
- ❑ Checkbox: Uncheck
- ❑ TBD
- ❑ Maps: Selected marker drill down
- ❑ Maps: anywhere click
- ❑ Grids: Cell Modified
- ❑ Grids: Cell Selection Changed
- ❑ Grids: Cell Edit Start
- ❑ Grids: Cell Edit Finish
- ❑ Grids: Column Sized
- ❑ Grids: Column Selection Changed

Note: The TBD trigger type can be used when a task is executed by an API call.

Trigger Identifier list

A list of all possible triggers based on what option you have selected from the Trigger filters list.

Clear Triggers Selection

Deselects any object you have selected in the Trigger Identifier list.

Buttons, Images, Hyperlinks/Selected Only

You can select whether to display buttons, images, hyperlinks, or other objects as possible Trigger Identifiers. The options are:

- ❑ Buttons, Images, Hyperlinks

- Buttons
- Images
- Hyperlinks
- Menu Items
- All
- Selected Only

When *All* is selected, objects that are not buttons, images, or hyperlinks become available for use as Trigger Identifiers. For example, with All selected, a text object can be a Trigger Identifier.

Selected only will make objects you have selected on the canvas appear in the Trigger Identifier list.

Conditions

Allows you to conditionalize the task based on the value selected in a control object. For more information, see *How to Apply a Condition to a Task* on page 187.

Requests/Actions list

A list that displays the order in which the requests, for the selected tasks, will execute any Wait for completion requests and any refresh requests. You can use Wait for completion requests to have certain requests execute before others. For example, if request1 is above the Wait for completion action and request2 is below that, then request1 will run first and when that is completed, request2 will then run.

Requests selections

Allows you to select which request(s) to add to the Requests/Actions list. Each request selected populates the Target type field with the available targets in which that request can be executed.

- **Run Request.** Allows you to execute a procedure. To make a procedure available for selection, reference it in the Requests & Data Sources panel.
- **Schedule Request.** Allows you to schedule a procedure. To make a procedure available for selection, reference it in the Requests & Data Sources panel. From the Target/Template Name menu, you can specify the scheduling method, or select *Email Library FTP* to allow users to select one of those three options.

- **Run Procedures/Html From Control.** Executes a procedure or HTML page selected from a specified control object on the page. To provide a list of procedures and HTML pages that can be executed, select the *Values are procedures/html files* check box in the Settings panel for the control.
- **Schedule Procedures From Control.** Schedules a procedure selected from a specified control object on the HTML page. From the Target/Template Name menu, you can specify the scheduling method, or select *Email Library FTP* to allow users to select one of those three options. To provide a list of procedures and HTML pages that can be executed, select the *Values are procedures/html files* check box in the Settings panel for the control.
- **Run Animation.** Displays the available jQuery animations.
- **Execute Task.** Displays the available tasks, to enable you to execute one task from within another.
- **Refresh.** Creates an action to refresh the selected target.
- **Wait for completion.** Allows you to specify that one request must complete before the next one begins.
- **JavaScript call.** Creates a JavaScript call action. When using the JavaScript call, different results occur depending on what is returned. If the JavaScript call returns 0, then all actions after it run. If the JavaScript call returns 1, then one action after the call is skipped, and all actions after that are then run. If the JavaScript call returns 2, all actions after it are skipped.

Delete

Deletes a request from the Requests/Actions list.

Move Up

Moves an item on the Requests/Actions list up one place.

Move Down

Moves an item on the Requests/Actions list down one place.

Target type

The Target type option contains a list of targets in which the request can be executed. These targets can be controls, frames, windows, or distribution methods when the Request is set to Schedule. These options can be different for specific actions.

The Window Target type will execute the request in a window.

The InfoWindow Target type will execute the request in the Db2 Web Query generated InfoWindow.

The Ajax call Target type will execute the request as an asynchronous call in JavaScript.

The Frame Target type will execute the request in a selected frame.

The Input Control Target Type is only available when you have a refresh request selected in the Requests/Actions list. This Target Type specifies that an input control will be refreshed. You specify which control should be refreshed by selecting one from the Target/Template Name list.

The Analytic Document Target Type is only available when you have a refresh request selected in the Requests/Actions list. This Target Type specifies that an Analytic Document will be refreshed. You specify the Analytic Document that will be refreshed by selecting one from the Target/Template Name list.

The Refresh BI Portal Target Type is only available when you have a refresh request selected in the Requests/Actions list. This Target Type specifies that a Db2 Web Query BI Portal page will be refreshed. If you select *All pages of the portal*, all BI Portal pages will be refreshed. If you select *Current page only*, only the current BI Portal page will be refreshed.

The JavaScript function Target Type is only available when you have selected *JavaScript call* in the Requests/Actions field. This Target Type specifies that the selected task will call a JavaScript function that you type in the Function name field.

Target/Template Name

Contains options that are associated with the Target type option.

Note: When you type a frame name in the Target/Template Name field, if the frame does not exist, a dialog box displays, asking if you want to create a new frame.

This field title changes to Function name under the following conditions:

- The Trigger Type *TBD* was selected.
- The Requests/Actions *JavaScript call* was selected.
- The Target type *JavaScript function* was selected.

In the Function name field, you can type the function that you want to call. You can also include any parameters that are required. Parameters must be included in parentheses and multiple parameters separated by commas (,). If you had previously typed a function in this field, you can select it from the drop-down menu.

Size (Width/Height)

Sets the Width and Height of an InfoWindow. Only available for the InfoWindow Target Type.

Current pages only

Refreshes the current Db2 Web Query BI Portal page. Only available when there is a request to refresh the Db2 Web Query BI portal.

All pages of portal

Refreshes all Db2 Web Query BI Portal pages. Only available when there is a request to refresh the Db2 Web Query BI Portal.

Executing jQuery Animations Using the jQuery Animations Section

The jQuery Animations section, in the Tasks & Animations panel, allows you to execute jQuery animations on your HTML page. The animation effects in the jQuery Animations section can be used individually or in conjunction with each other to create more complex animations.

For more information, see *How to Animate an Object Using the Tasks & Animations Panel* on page 183.

List of

A list of all animations present in the HTML page. This list will change when animations are added or deleted.

New

Adds a new animation to the HTML page. The newly added animation appears in the List of area.

Delete

Removes an animation from the HTML page. The deleted animation is removed from the List of area.

All Targets

A list of all possible targets that can be used in the current animation being defined.

Selected Targets

A list of all targets that will be used in the current animation.

Add canvas selection

Adds the currently selected object on the canvas to the Selected Targets list. Multiple objects can be selected and added at the same time.

Add from list

Adds a selected object from the All Targets list to the Selected Targets list. Multiple objects can be selected and added at the same time.

Remove

Removes an object from the Select Targets list when that object is selected. Multiple objects can be selected and removed at the same time.

Visibility

You can select whether or not to animate the target so that it is hidden or displayed. You can also select not to animate the target for visibility. The options are No Change, Hide, and Show.

Effect

The Effect group contains animation effects and the options you can use to customize those animations. Animation effects can make the target bounce, shake, pulsate, and much more. You can also customize the options associated with that animation effect. The options correlate to the effect used. Such options can be how many times the target bounces after you specify that you want the bounce effect.

Type

A drop-down list where you can select the animation effect you want to use.

Option

You can select which option value you would like to edit, if available. For example, selecting the Bounce animation type allows you to select Distance and Times in the Option drop-down list. When you select *Distance* or *Times*, you can then edit the value of either of those options in the Value area. If you select *Times* and then enter *3* into the Value area, your object will bounce 3 times. The options available are different depending on the effect type you choose.

Value

You can enter a number that pertains to the currently selected option and that affects the animation. For example, if you select the *Slide* type and the *Distance* option, the number you enter in the Value area will be the distance the object will slide when it is animated. The value options available are different depending on the effect type you choose.

Effect text area

As the options for the effect are set, the syntax for that effect are displayed here. You can manually adjust this syntax to affect the animation.

Additional Properties

The Additional Properties group contains animations that change the location and size of the target and the components of that target (text size, text width, border size).

Name

A drop-down list where you can select different properties that can be animated. Such properties include height, opacity, font size, and others.

End value

The value for the property you choose from the Name field. For example, if you choose *Left* and enter *10*, the object you are animating would move 10 pixels to the left. The reason you do not need to specify a start value is because the current location of the object is the start value.

Location setting

You can animate the target so that the location and size of that target is changed once the animation is activated.

Use location of selected target

This will use the location of the currently selected target. Coordinates, height, and width of the object are entered automatically into the Location setting box. This is used when the developer plans to move the selected target object and the animation will move it back.

Start location setting

Inserts an adjustable placeholder object onto the canvas, at the location of the selected target. You can move and resize this placeholder object. This object represents where the animation will move and what size it will be when the animation is complete.

Set location settings

Sets the adjustable object location and size. This will be where the animated object moves and what size it will be. It is the ending location of the selected target.

Cancel setting

Cancels the setting of the location for the animation.

Additional Properties text area

Once the location has been determined using the Location setting commands, the syntax for that location is displayed here. You can manually adjust this syntax to affect the animation.

Duration

How quickly the animation will execute. You can choose either *Slow* or *Fast*.

Toggle animation

Allows you to revert to the pre-animation state. For example, if the target moves from right to left, the next invocation of the trigger will move the target back to the original position at the right.

Procedure: How to Animate an Object Using the Tasks & Animations Panel

The following are examples for how to create and animate an HTML page with jQuery animations.

Note: Any of the jQuery animations in the jQuery Animations section can be used individually or together. Using different jQuery animations together will create a more complex animation.

1. Create an HTML page.
2. Insert an image object and a button object onto the HTML page.
3. On the Tasks & Animations panel, in the jQuery Animations section, click *New*.

   A new animation is added to the animation list.
4. To make the image object, image1, the target, do one of the following:

   - Select the image object on the canvas and click the *Add canvas selection* command.
   - Select *image1* from the All Targets area and click the *Add from list* command.
   - Double-click *image1* in the All Targets area to move it to the Selected Targets area.

   The image object, image1, is added to the Selected Targets area.
5. Select *Hide* for the Visibility option.

   This animation option will cause the image to be hidden when the button is clicked.
6. On the Tasks & Animations panel, in the Tasks section, click *New*.

   A new task is added to the tasks list.
7. Select *Click* from the Trigger Type drop-down list.
8. From the Trigger Identifier list, select the button object, *button1*.

   The button will be used to activate the animation.
9. Select *Animation1* from the Requests/Actions option.

When the animation is activated using the button, the image will hide.

Procedure: How to Select an Effect and Customize the Options for an Animation

The following is an example for how to select an effect for an animation and customize the options for the effect.

Note: Any of the jQuery animations in the jQuery Animations section can be used individually or together. Using different jQuery animations together will create a more complex animation.

1. Create an HTML page.
2. Insert an image object and a button object onto the HTML page.
3. On the Tasks & Animations panel, in the jQuery Animations section, click *New*.

   A new animation is added to the animation list.
4. To make the image object, image1, the target, do one of the following:

   - Select the image object on the canvas and click the *Add canvas selection* command.
   - Select *image1* from the All Targets area and click the *Add from list* command.
   - Double-click *image1* in the All Targets area to move it to the Selected Targets area.

   The image object, image1, is added to the Selected Targets area.
5. Select an Effect and customize the options for that effect.
   a. Select *Blind* from the Type drop-down list.

      This will cause the image to appear as if window blinds have been pulled over it.

   b. Select *Direction* from the Option drop-down list.

      Note: Each Effect has different options available to customize. Blind has the Direction option to customize where Bounce has the Distance and Times options.

   c. Select *Horizontal* from the Value field.

      This will cause the image to be animated with a horizontal blind effect.

      Note: The Value field will either be a drop-down list or a text entry box depending on what you selected for the Option drop-down list. For example, Direction will use this as a drop-down list, while Times will use this as a text entry box.
6. On the Tasks & Animations panel, in the Tasks section, click *New*.

   A new task is added to the tasks list.
7. Select *Click* from the Trigger Type drop-down list.
8. From the Trigger Identifier list, select the button object, *button1*.

   The button will be used to activate the animation.

9. Select *Animation1* from the Requests/Actions option.

When the animation is activated using the button, the image will animate with a blind effect.

Procedure: How to Create a Move or Resize Animation

The following is an example for how to create a move or resize animation.

Note: Any of the jQuery animations in the jQuery Animations section can be used individually or together. Using different jQuery animations together will create a more complex animation.

1. Create an HTML page.
2. Insert an image object and a button object onto the HTML page.
3. On the Tasks & Animations panel, in the jQuery Animations section, click *New*.

   A new animation is added to the animation list.
4. To make the image object, image1, the target, do one of the following:

   - Select the image object on the canvas and click the *Add canvas selection* command.
   - Select *image1* from the All Targets area and click the *Add from list* command.
   - Double-click *image1* in the All Targets area to move it to the Selected Targets area.

   The image object, image1, is added to the Selected Targets area.
5. Use the Location setting options, found in the Additional Properties area, to animate the object so that it moves when activated.
   a. Click the *Start location setting* command.

      A placeholder object is shown overlapping your image.
   b. Move the placeholder object to a different location on the HTML page.
   c. Click the *Set location setting* command.

      This removes the placeholder object and adds syntax to the Additional properties text area. When your animation is activated, it will move to a different location that you specified using the placeholder object.
6. Select *Fast* from the Duration drop-down list.

   This will make the animation run at a faster speed.
7. Check the *Toggle Animation* check box.

   Checking the Toggle Animation check box will allow you to click the button after the animation has run and restore the image to its original state (in its original location and unhidden).

8. On the Tasks & Animations panel, in the Tasks section, click *New*.

   A new task is added to the tasks list.

9. Select *Click* from the Trigger Type drop-down list.
10. From the Trigger Identifier list, select the button object, *button1*.

    The button will be used to activate the animation.

11. Select *Animation1* from the Requests/Actions option.

When the animation is activated using the button, the image will move to the location that you set. If you click the button a second time, it will move back to the original location.

Procedure: How to Create a Hyperlink in Developer Workbench to Drill to a Db2 Web Query Procedure

You use the Hyperlink component and the Tasks & Animations panel to create hyperlinks. The following is an example of creating a hyperlink to drill to a Db2 Web Query procedure.

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 already chosen.

2. In the Requests & Data sources panel, create a new external request for a Db2 Web Query 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 and select the request you created in step 2.
5. In the Trigger 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 Db2 Web Query Procedure you chose will run in the frame.

Procedure: How to Create a Hyperlink in Developer Workbench to Execute a URL

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 already chosen.

2. In the Requests & Data Sources panel, click the *New* drop-down arrow and select *Url Request*.
3. Type the URL in the Enter Url dialog box.

4. Navigate to the *Tasks & Animations* panel, select the task for the hyperlink, and verify that the trigger identifier displays the unique ID of the hyperlink object.
5. Click the arrow in the Requests/Actions section, select *Run Request*, and then select the URL request.
6. Select the Target type from the drop-down list (for example, *Window* or *Frame*).
7. Select the appropriate Target/Template name from the drop-down list.

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

You can use the Requests and Tasks sections 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. Select the *load* task.
2. Select the *Load* trigger type in the Trigger Type drop-down list. This option is selected by default, for a load task.
3. Select a request from the Requests/Actions list.
4. Select the destination of the procedure by using the Target type drop-down list. For example, you can have the procedure run in a new window or in a report frame on the HTML page.
5. Select the frame in which the procedure will run. This option is only needed if you want to run the procedure, in a frame, on the HTML page, and not a new window.

Procedure: How to Apply a Condition to a Task

You can apply a condition to a task so that it will only execute when certain values are selected from a control on the page. This allows you to modify run-time behavior based on user selection.

If you apply multiple conditions to a task, they must all be satisfied in order to execute the request or action associated with the task. If only one of the conditions is satisfied, the task will not execute. When the task is triggered for execution but the conditions are not all satisfied, an alert message appears indicating that the conditions are not met.

1. Create an HTML page that contains a control, a button to use as the Trigger Identifier, and a chart or report. The Trigger Identifier can be the control object.
2. Select the control and open the Settings panel.
3. Populate the control. You can use the Static data type and add new values manually, or use the Dynamic data type to take values from a data source.

4. Open the Tasks & Animations panel.
5. In the Tasks section, click the *New* button to create a new task.
6. Select a Trigger Type. This is the action that you want to use to initiate the task, for example *Click*.
7. Specify the Trigger Identifier. This is the object on the page upon which you perform the action set as the Trigger Type in order to initiate the task, such as a button that you can click.
8. Click the *New condition* button to create a condition for the task.

   The task will only execute if the condition is satisfied.
9. Click the *Edit* button to define the condition.

   The Task condition dialog box opens.
10. In the Task condition dialog box, select the following options.
    a. From the Input controls menu, select the control on the page whose values will be used in the condition.
    b. From the Values compare operator menu, select the test to use for the condition. You can set the condition to be satisfied when the selected value is equal to, not equal to, greater than, greater than or equal to, less than, or less than or equal to the compare value or values that you will specify.
    c. From the Multiselect operator menu, specify how multiple compare values should be treated. Select *All of* to specify that all compare values must be selected to satisfy the condition, or select *One of* to specify that just one of the compare values must be selected.
    d. In the Compare to values text area, supply the compare values that can be selected to satisfy the condition. You can type them manually, with one on each line, or click the ellipsis button and select values from the control.
11. Click *OK* to apply your changes and close the Task condition dialog box.
12. Optionally, click *New condition* again, select it from the list of conditions, and repeat steps 9 through 11 to add a second condition that must also be satisfied in order for the task to execute.

    You can also click the *Delete* button to delete the selected condition.
13. Add a request or action to execute when the condition or conditions are satisfied. For example, if you have a report on the page that you want to execute, click the down arrow next to the *Requests selections* button, point to *Run Request*, and select the report procedure.
14. Select an option from the Target type menu. Different target types are available for different kinds of requests and actions. The target type determines where or how the task executes.

15. Run the page.

    When you select values from the control object and execute the task, the condition is tested. If all conditions are satisfied by the values that you selected, the task executes as specified. If the conditions are not all satisfied, an alert appears.

Working with Requests & Data Sources

A request is a definition of something that can be executed. For example, you can execute a default procedure, explicit 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 in the request section 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, it 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 Db2 Web Query 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 default 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 the Data Sources folder, and using the shortcut menu to click *Add Data Source*.

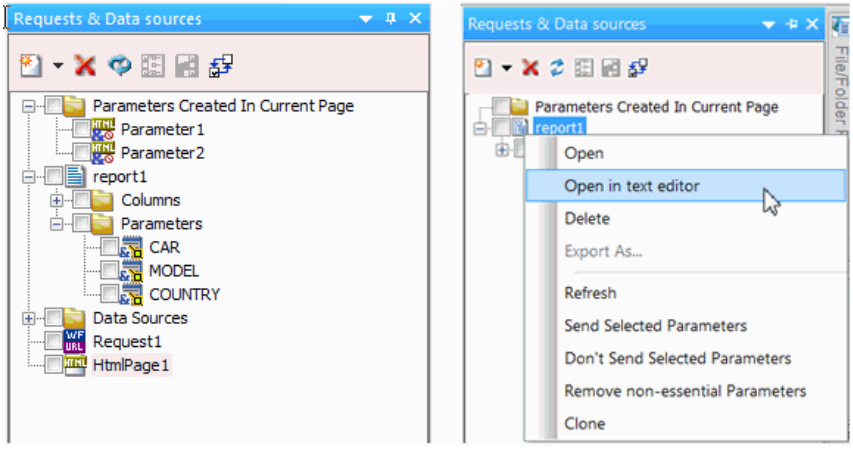
You can refresh all requests, except empty requests, using the shortcut menu. Right-click a request and click *Refresh*.

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 create a copy of the default request with the name, *name*_1. Name is the name of the original default request. For example, if you clone a request named SalesReport, the cloned request would be called SalesReport_1. Cloning an explicit 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 right-most image below. For a default procedure, the new procedure tab opens with an extension of .vrt to show it is virtual. With a referenced procedure, the procedure tab opens with an extension of .fex.

In the Requests & Data sources panel, each icon next to an 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 an addition sign (+) on it. Another example is how an external URL request has the word *URL* on it.

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



You can drag parameters and fields to the Settings panel to configure the selected component or control. You can drag requests from the Tasks & Animations panel to the HTML canvas, to create objects and controls for the request.

New request

Creates a new request. A request is an item that can be executed.

Delete

Deletes a selected request.

Refresh parameters

Refreshes the parameters used in the request.

Create Input Controls

Opens the New Parameter dialog box where you can select which input controls to create for parameters that are checked off in the Parameters list box, and if you want those controls chained to the parameter.

Save Selection

Saves the parameters you currently have checked off in the Parameters list box. You can choose to send more or less parameters than the procedure in the request requires. If you send less parameters, then you need to make sure the remaining parameters are handled by a -DEFAULT, -SET, or some other construct in the request.

Clear Selection

Clears all selected items in the Requests & Data sources panel.

Procedure: How to Create a Default Procedure Request in Developer Workbench

1. Add an embedded procedure to the Requests & Data source panel. Click the *New* drop-down arrow, point to *Embedded Request*, and then click *Import Existing* to embed an existing procedure file.

   The Open File dialog box appears, prompting you to select a procedure file.

   Note: The New Report, New Chart, or New Document options under *Embedded Request* are not supported.

2. Select a procedure file and click *OK*.

   Once you are done selecting the procedure, save and close the canvas.

3. In the Requests & Data sources panel, you may drag the request that you embedded on to the canvas.

   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.

Specifying Browser Defaults

You can use the Style Composer to control default settings for font, background properties, position mode, flow control, margins, list styles, and visual effects.

To access the Style Composer, right-click an object on your HTML page and select *Style* from the shortcut menu.

Reference: Specifying Font Styles Using the Style Composer

To specify the font styles that will be used in the browser for your HTML page, make your selections in the Font window of the Style Composer.

The Font window of the Style Composer is comprised of the following elements:

Font name

Determines the font displayed in a browser.

You can specify: *Family* (launches the Font Picker dialog box) or *System Font*.

Font attributes

Determines the attributes of the font displayed in a browser.

The options include: *Color, Italics, Small Caps*.

Size

Determines the size of the font displayed in a browser.

The options include: *Specific, Absolute, Relative*.

Bold

Determines whether the font is displayed as bold in a browser.

The options include: *Absolute, Relative*.

Effects

Determines whether the font effects are displayed in a browser.

The options include: *None, Underline, Strikethrough, Overline, Capitalization*.

Reference: Specifying Background Properties Using the Style Composer

To specify the background styles that will be used in the browser for your HTML page, make your selections in the Background window of the Style Composer.

The Background window of the Style Composer is comprised of the following elements:

Background color

Determines the background color of the HTML page.

You can specify: *Color, Transparent*.

Background image

Determines the properties of the background image displayed in a browser.

The options include: *Image, Tiling, Scrolling, Position* (Horizontal and Vertical), *Do not use background image*.

Note: When using a background image with scrolling enabled, you must specify the horizontal and vertical positions. If you do not specify these positions, your background image will not show. The horizontal and vertical positions are relative to the window and not the individual element.

Reference: Specifying Text Styles Using the Style Composer

To specify the text styles that will be used in the browser for your HTML page, make your selections in the Text window of the Style Composer.

The Text window of the Style Composer is comprised of the following elements:

Alignment

Determines the alignment of text.

You can specify: *Horizontal, Vertical, Justification*.

Spacing between

Determines the spacing between text.

You can specify spacing between the following text elements: *Letters, Lines*.

Text flow

Determines the flow of the text.

You can specify: *Indentation, Text direction*.

Reference: Specifying Position Mode Using the Style Composer

To specify the position mode that will be used in the browser for your HTML page, make your selections in the Position window of the Style Composer.

The Position window of the Style Composer is comprised of the following elements:

Position Mode

You can specify: *Position in normal flow, Offset from normal flow, Absolutely position*.

Height/Width

When Absolutely Position is selected, you can specify position indicators in the measurements.

You may specify: *Top, Left*, *Z-Index*.

Note: Z-Index is optional. It sets or retrieves the stacking order for absolute or relatively positioned objects.

Reference: Specifying Layout Styles Using the Style Composer

To specify the layout styles that will be used in the browser for your HTML page, make your selections in the Layout window of the Style Composer.

The Layout window of the Style Composer is comprised of the following elements:

Flow control

You can specify: *Visibility, Allow text to flow, Display, Allow floating objects*.

Content

You can specify: *Overflow*.

Clipping

You can specify whether or not to clip the layout from the following positions: *Top, Bottom, Left, Right*.

Printing page breaks

You can specify: *Before, After*.

Reference: Specifying Edge Styles Using the Style Composer

To specify the margins, padding, and border styles that will be used in the browser for your HTML page, make your selections in the Edges window of the Style Composer.

The Edges window of the Style Composer is comprised of the following elements:

Margins

You can specify: *Top, Bottom, Left, Right*.

Padding

You can specify: *Top, Bottom, Left, Right*.

Reference: Specifying List Styles Using the Style Composer

To specify the list styles that will be used in the browser for your HTML page, make your selections in the Lists window of the Style Composer.

The Lists window of the Style Composer is comprised of the following elements:

Lists

You can specify: *Bulleted, Unbulleted*.

Bullets

You can specify: *Style, Position, Custom bullet*.

Reference: Specifying User Interface Effects Using the Style Composer

To specify the interface styles and visual effects that will be used in the browser for your HTML page, make your selections in the Other window of the Style Composer.

The Other window of the Style Composer is comprised of the following elements:

User interface

You can specify: *Cursor*.

Tables

You can specify: *Borders, Layout*.

Reference: Specifying Border Settings Using the Style Composer

To specify the Border settings that will be used in the browser for your HTML page, make your selections in the Border window of the Style Composer.

The Borders window of the Style Composer is comprised of the following elements:

Borders

Determines which border should be styled, what type of style should be applied, the width of the border, and the color of the border.

Rounded Border

Determines the whether your border corners will be rounded. You can style each individual corner to be rounded. You can also specify by how much they are rounded and whether all corners should be equally rounded.

Chaining in the HTML Canvas

You may chain controls to one another on the Parameters tab and apply conditions to links in the chain. Chaining will populate controls based on the selected value from the prior control in the chain. You can chain static and dynamic controls, link or unlink parts of a chain, and create conditions on links in a chain. Chains are represented by lines connecting control objects on the Design or Parameters tab. Note that chaining is applicable only for controls, not parameters.

Note:

- Although you may chain controls from the Design tab, you may only create conditions to links in the chain through the Parameters tab.

  By clicking the arrow head in a link of a chain, the Settings panel enables you to modify and set the properties and conditions of the chain.

- If using static controls, you must apply conditions for each link in the chain. Conditions need to be created for each value of the control chained from, and those values must be mapped to the correct value(s) that will be displayed in the control that it is being chained to.

Automatically Chaining Parameters From the New Parameters Dialog Box

The auto chain option enables you to automatically chain selected controls from the New Parameters dialog box. Chaining populates controls based on the selected value from the prior control in the chain. The auto chain option is useful since it creates the chain, or links of a chain, automatically.

Note: Automatic chaining creates a basic chain with default functionality that does not include any conditions. You may create conditions for a chain through the Parameters tab.

When importing or referencing a report with parameters to an HTML page, the controls are not chained, by default. You may choose to include or exclude individual controls in a chain with the Chain control column from the New Parameters dialog box.

Additionally, when the auto chain option is selected, a separator is added to the parameters list on the New Parameters dialog box. A separator is used to separate controls into multiple chains and can be moved up or down in the chain sequence.

Procedure: How to Auto Chain Controls From the New Parameters Dialog Box

The auto chain option creates the chain, or links of a chain, automatically. When the auto chain option is selected, a separator is added to the parameters list. A separator is used to separate controls into multiple chains and can be moved up or down in the chain sequence.

1. In the HTML canvas, import or reference a report with parameters.

   The New Parameters dialog box opens.

2. Select *Auto chain controls in above specified order*.

   Tip: You may use the up or down arrows to change the order of the selected control before selecting this option.

   The Chain control option is selected for all controls and a separator is added as the last object to the list of parameters.

3. To create multiple chains, click the separator row and use the up or down arrows to change the location of the separator in the chain.

   Note: If the default separator is moved up, another separator is added to the end of the list.

4. Click *OK* to close the New Parameters dialog box and add the control to the HTML page.

   On the Design tab, when creating multiple chains from the New Parameters dialog box, each set of chained parameters appears on a new line, regardless of the grouping option selected from the New Parameters dialog box. This behavior is set through the *Start each chain on a new line* option, located on the Form Settings dialog box in the HTML Page tab of the Developer Workbench Options dialog box. This enables you to see the relationship of the chains within the form. Start each chain on a new line is selected, by default.

   On the Parameters tab, chains are represented by lines connecting control objects.

5. You can remove a chain by clicking *Break binding* from the shortcut menu when a chain link is selected.

Procedure: How to Chain Controls From the New Parameters Dialog Box

The Chain control column enables you to include or exclude individual controls in a chain, from the New Parameters dialog box.

1. In the HTML canvas, import or reference a report with parameters.

   The New Parameters dialog box opens.

2. Select the *Chain control* check box for the controls to be included in the chain.

   The controls are chained in the order that they appear on the New Parameters dialog box. You may use the up or down arrows to change the order of the selected control before chaining controls.

   Note: If a control is excluded from a chain, the chain automatically links only the selected controls.

3. Click *OK* to close the New Parameters dialog box and add the control to the HTML page.

   On the Design tab, when creating multiple chains from the New Parameters dialog box, each set of chained parameters appears on a new line, regardless of the grouping option selected from the New Parameters dialog box. This behavior is set through the *Start each chain on a new line* option, located on the Form Settings dialog box in the HTML Page tab of the Developer Workbench Options dialog box. This enables you to see the relationship of the chains within the form. Start each chain on a new line is selected, by default.

   On the Parameters tab, chains are represented by lines connecting control objects.

4. You can remove a chain by clicking *Break binding* from the shortcut menu when a chain link is selected.

Creating Pop-Up Controls

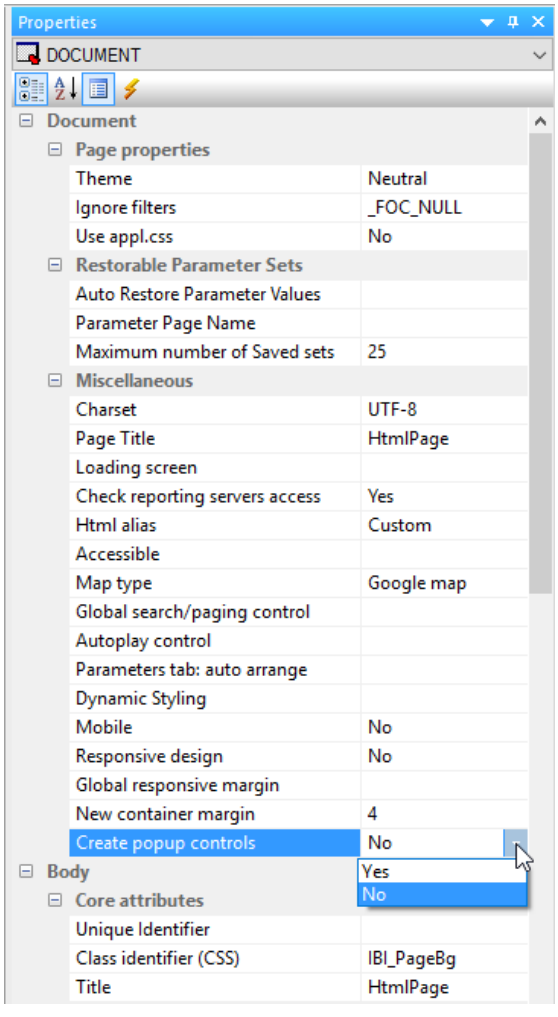
For a DOCUMENT object, you can use pop-up controls to:

- Create modern looking controls that will run seamlessly on any device.
- Create controls that look the same, but offer different functionality based on single or multiple value selection.

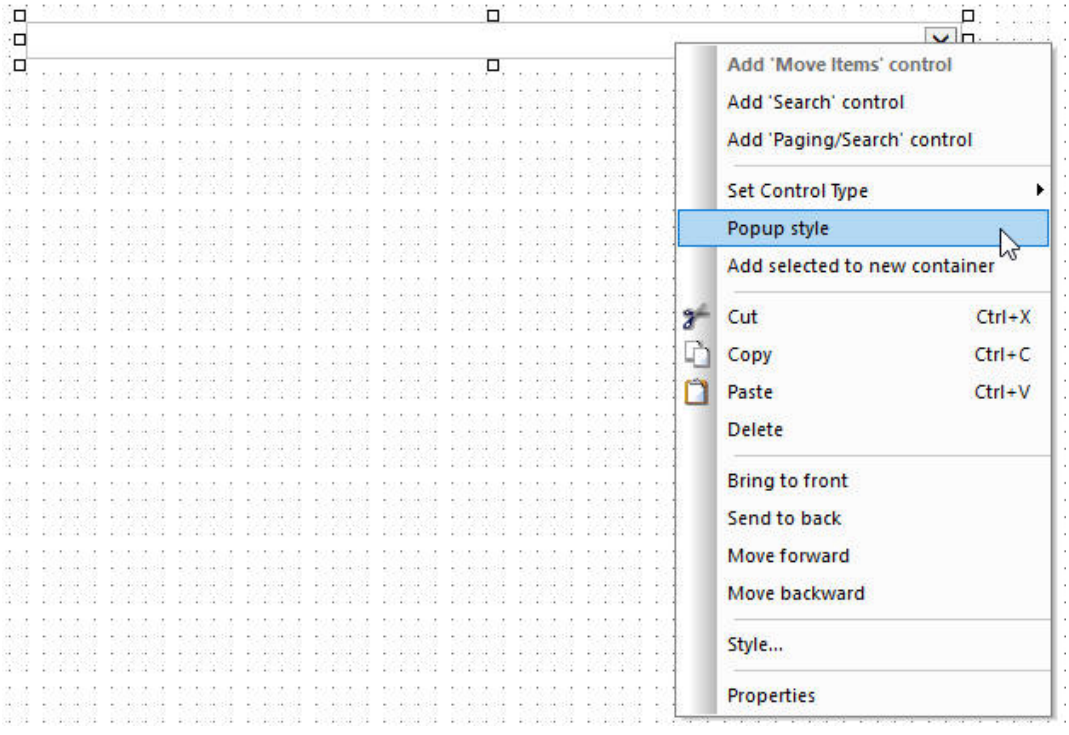
To create pop-up controls, you can:

- Set the Create popup controls property on the Properties panel for the DOCUMENT object.
- Use the Popup style right-click shortcut option, which allows a control to be an original or a pop-up style control. (Disabled for Edit Box, Text Area, Calendar, and Slider).
- In the New Parameters dialog box, set the value in the Create control column to *Popup*.

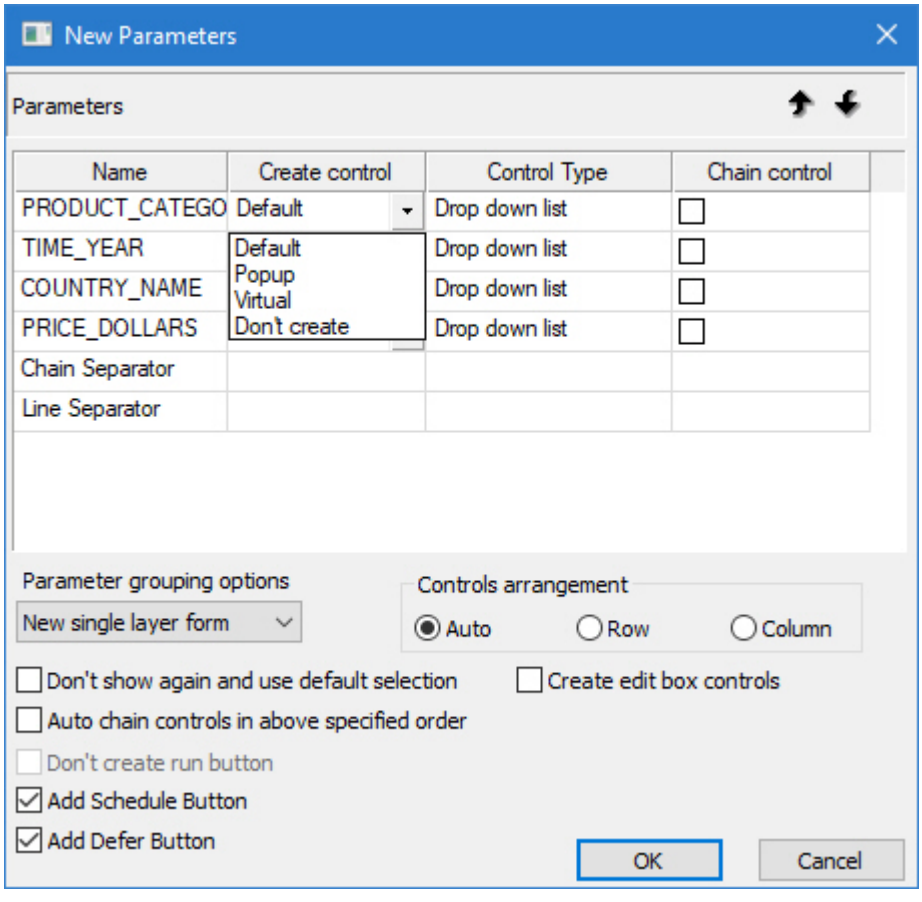
The Create popup controls property on the Properties panel is shown in the following image. Possible values are Yes and No. No is the default value.



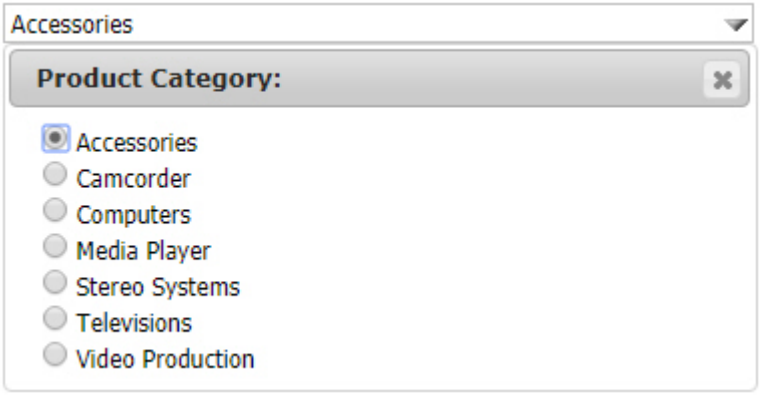
The Popup style right-click shortcut option for a drop-down control is shown in the following image. (Disabled for Edit Box, Text Area, Calendar, and Slider).



The New Parameters dialog box is shown in the following image. The Popup option is available in the drop-down menu in the Create control column.



The following image shows an example of a pop-up control.



Procedure: How to Add a Title to Pop-Up Controls

You can add a title bar and specify a customized title for pop-up controls in an HTML page.

To add a title bar, in the Properties panel, set Display Title Bar to *Yes*. When Display Title Bar is set to Yes, the Title Text property becomes available. Type a value in this field to display in the pop-up control title bar at run time.

1. Create an HTML page that includes a control object.
2. Change the control to a pop-up control.

   You can do this in one of the following ways:

   - Before adding controls to the page, open the Properties panel. For the DOCUMENT object, set the Create popup controls property to *Yes*.

     Any controls added to the page are created as pop-up controls.

   - In the New Parameters dialog box, when creating the control, select *Popup* from the Create control column.

   - Right-click the control and click *Popup style*.

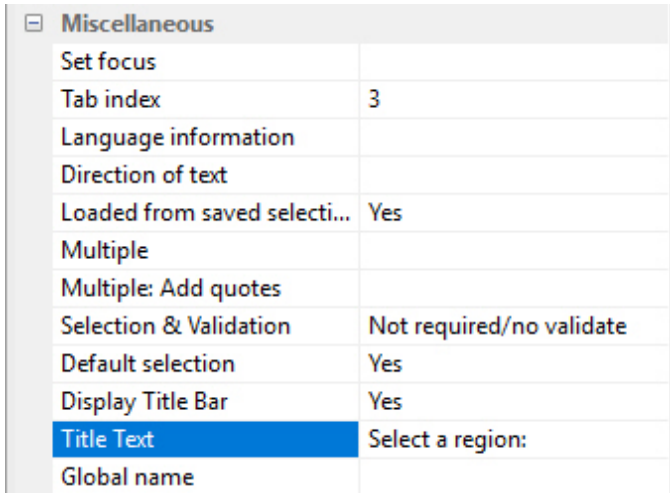
   Note:

   - Edit boxes, text areas, calendars, and sliders cannot use pop-up styling.

   - Multiselect drop-down controls use pop-up styling, by default.

3. Select the control, and open the Properties panel. When the control is set to use pop-up styling, the Display Title Bar property becomes available.

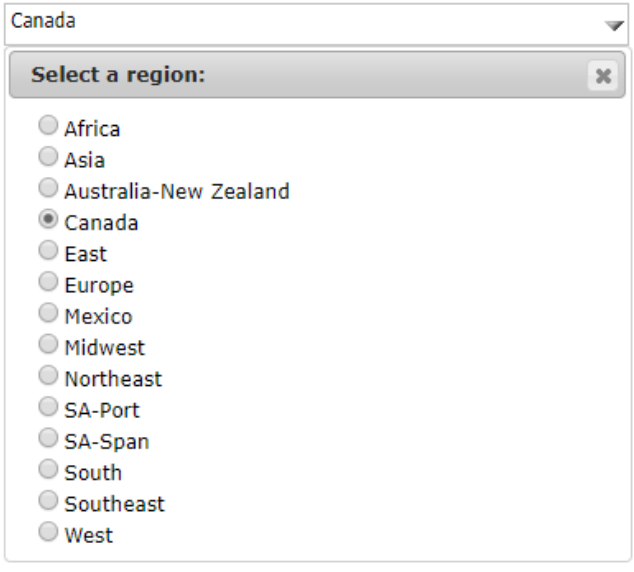
4. Set the Display Title Bar property to *Yes*.
5. When the Display Title Bar property is set to Yes, the Title Text property appears. Type the text for the control title into this field, as shown in the following image.



If you leave the Title Text field blank, the unique identifier of the control or the label assigned to the control, if there is one, is used as the title text.

6. Run the HTML page and click the control.

The text specified by the Title Text property appears on the control title bar, as shown in the following image.



Using the Chain Separator and Line Separator

The chain separator and line separator options allow for chains and parameters to be grouped or split depending on which settings are turned on. When used, the two separators do the following:

Chain separator. When *Start each chain on a new line* is on, this separator creates new chain groups on new lines. When *Start each chain on a new line* is off, this separator starts a new chain wherever it is placed in the New Parameters dialog box. The controls are positioned in one row and wrap at the end of the form.

Note: *Start each chain on a new line* is turned on, by default. This option can be found in the Form Settings dialog box, in the HTML Page section of the Developer Workbench Options dialog box.

Line separator. This separator creates a line break wherever it is placed.

Chaining Controls on the Parameters Tab

Chaining enables you to associate two or more related values. When you chain controls together, chained values are filtered as selections are made to each parameter control. For example, if you chain the PLANT parameter to the STATE parameter, only PLANT values for the currently selected STATE value will be available, instead of all the plants in the data source. Each time a selection is made, all chained controls that come after will be dynamically updated. Chaining also enables you to add, remove, and reverse the order of controls in the chain.

Values are processed with a caching mechanism that gathers all of the necessary values prior to loading the page. This method automatically combines all of the necessary requests into a single HTTP request and maps the result sets to the appropriate controls, greatly reducing the load time involved with sending multiple requests for data.

If a parameter has two or more incoming bindings, the value selected last will be displayed as the parameter.

Note: When a parameter is populated by two or more controls, the value of the last control used will be assigned to the parameter.

Procedure: How to Chain Controls on the Parameters Tab

1. Create an HTML page using input controls to supply parameter values.
2. Click the *Parameters* tab.
3. Select the center of the control object and drag the control to the center of the next control object in the chain.

   Repeat this step for each link in the chain.
4. Optionally, apply condition settings to the chain to determine how parameters are populated.

   If using static controls, you must apply conditions for each link in the chain. Conditions need to be created for each value of the control chained from, and those values must be mapped to the correct value(s) that will be displayed in the control that it is being changed to.

Procedure: How to Remove a Link in the Chain

1. While on the Parameters tab, select the arrow head on the line so that the line is bold.
2. Right-click and select *Break binding*.
3. Repeat this step for each link in the chain that you want to break.

Procedure: How to Reverse the Order of Chained Controls

1. While on the Parameters tab, select the arrow head on the line so that the line is bold, right-click, and select *Break binding*.
2. Select the center of the control object and drag the control to the center of the next control object in the chain.

   Notice the direction of the arrow between the control objects. You may reverse the direction of the link in the chain or reverse the order of the chain by changing the direction of each link.

   - ❏ If reversing the direction of a link in the chain, drag the control object in the desired order.
   - ❏ If reversing the order of a chain, drag the control objects in the desired order.

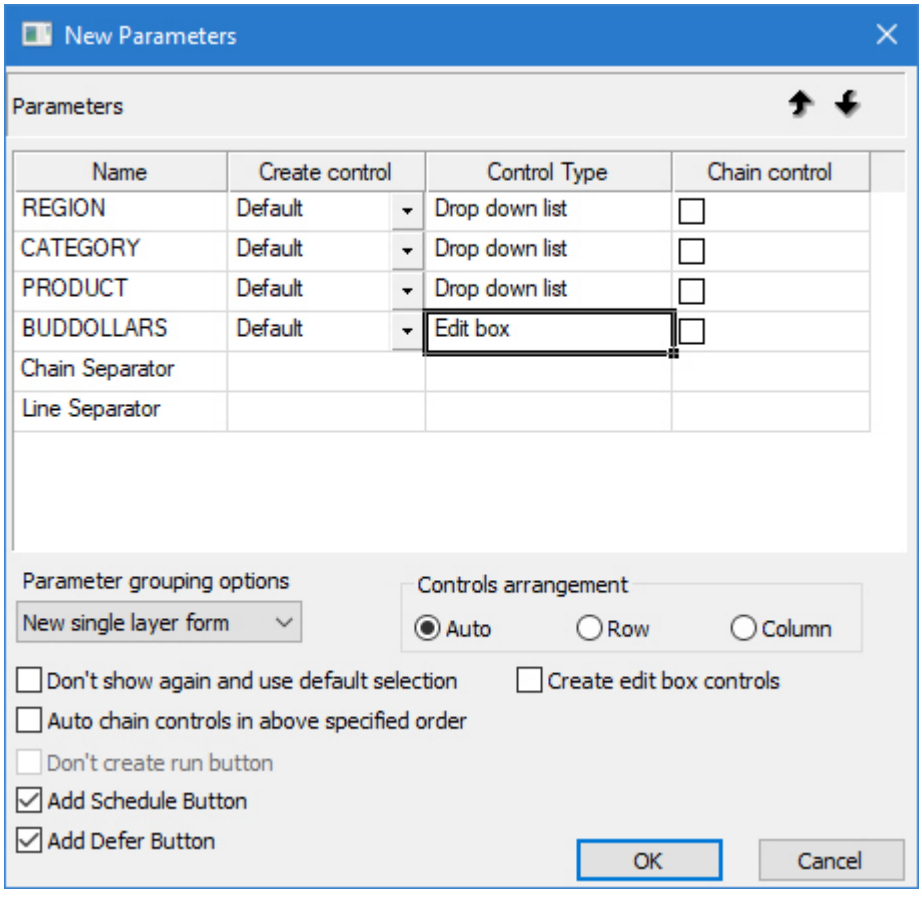
   Tip: In some scenarios, when reversing the order of chained values, you may want to move the controls from the default location on the Parameters tab so that you can better see the direction of the chain. Moving objects on the Parameters tab will not affect the Design view of your layout.

   - ❏ Press the Shift key and select the control object and bound parameter to move the objects as a set.
   - ❏ Chain the control objects together.
3. Optionally, apply condition settings to the chain to determine how parameters are populated.

Arranging Controls

You can choose the arrangement of controls in the New Parameters dialog box, without any additional steps, after the parameter controls are generated. This gives you the flexibility to design your HTML page, without the need for moving the controls after creation.

The Controls arrangement option, as shown in the following image, indicates the placement of controls.



The following are possible values for the Controls arrangement option:

- **Auto.** This value depends on the value in the Number of columns property in the Form settings dialog box. This dialog box is in the HTML Page section in the Developer Workbench Options dialog box. Based on this value, the controls will fold to the next line. The valid values for Number of columns are 1 to 99. Auto is the default value.
- **Row.** Places the controls horizontally within the form.
- **Column.** Places the controls vertically within the form.

Applying Conditions to a Chain

A chain contains conditions for each link in the chain. The conditions are linked to the values being selected in the control object. You may apply multiple conditions to one link. The settings for the condition describe how the link should behave. The following options are available:

- Apply *Actions* for the links on the chain.
- Apply the *Values compare operator* for the condition.
- Apply *Selected values* with a *Multiselect operator* for the condition.
- Apply the *Resolves parameter* values for the condition.
- Apply the *Compare operator* for the condition.

If using static controls, you must apply conditions for each link in the chain. Conditions need to be created for each value of the control chained from, and those values must be mapped to the correct values that will be displayed in the control that it is being chained to.

Reference: Settings Panel (Conditions)

The Settings panel appears when creating a condition for a chain link on the Parameters tab. A chain contains conditions for each link in the chain.

The conditions are linked to the values being selected in the control object. The settings for the condition inherit the values of the prior bound control and provide additional condition settings. This section describes the additional condition settings.

The Settings panel contains the following fields and options when creating a condition:

Conditions

The conditions list enables you to create multiple conditions for the link. *Default* is the only initial condition.

- To create a new condition, click the *New* icon. Condition*(n)* is created, where *(n)* is the number, and added to the Conditions drop-down list. You may type in a unique condition name, choose *Selected values*, and set the condition settings.
- Click the *Delete* icon to remove the selected condition from the list. Note that the *Default* condition name cannot be deleted.

Actions

Select an action for the chain link to control. The options offer variations to populate, show, hide, execute, and select the values. The list of available options are:

- **Populate, show.** Populates the control and displays it at run time. This is the default action for all conditions.
- **Populate, hide.** Populates the control and does not display it at run time.
- **Populate with alternate, show.** Populates the control with alternate values derived from a procedure, or value list, that is not the default and displays the control at run time.
- **Populate with alternate, hide.** Populates the control with alternate values derived from a procedure, or value list, that is not the default and does not display the control at run time.
- **Show.** Shows the control, but does not populate it.
- **Hide.** Hides the control, but does not populate it.
- **Execute.** Executes the bound object. For example, if you bind a control to a Submit button and change the value in the control at run time, the report/chart automatically executes when you change the value, without having to click the Submit button.
- **Select.** Selects the bound object. For example, if you bind a control to a tab item and select a value in the control at run time, the bound object (the tab item) is automatically selected as the active tab on the page.
- **Use As Target.** This action is used in conjunction with a control whose values are procedure names or HTML file names. Selecting this option, binds the control to a frame for the output.

Values compare operator

Values compare operator provides chaining logic scenarios to include such options as Equal, Not Equal, Greater Than, Less Than, and so on. This option sets the condition for how to populate the control being linked to.

Equal is the default Values compare operator.

Selected values

Selected values enable you to provide the values used in the condition. When creating a new condition, the Selected values section is activated. You may type selected values in the input box or click the ellipsis button to select values from the list.

The list of values that appears is based on the values of the prior bound control in the chain.

When selected values are entered, the Multiselect operator field is activated.

Multiselect operator

The Multiselect operator options are activated when selected values are entered for the condition. Options are *One of* or *All of*. One of is based on one of the values shown in the Selected values, being selected in the prior control, in the chain. *All of* is based on the value of all of the Selected values, being selected in the prior control, in the chain.

One of is the default multiselect operator.

Resolves parameter ("To:" field is required)

From. The From field specifies where to get the value used in the To field, if the control being chained from is an activeX control.

To. The To field is used to dynamically generate the selection list used to populate the control being chained to. This field displays the parameter whose value will drive the condition evaluation. The parameter name linked to the prior control in the chain is displayed by default. The ellipsis button provides a pop-up dialog of the other parameter values (from the report) to be resolved.

Compare operator

The compare operator provides parameter chaining logic scenarios to include such options as Equal, Not Equal, Greater Than, Less Than, and so on. This sets the compare operator to populate the control.

Equal is the default Compare operator.

Procedure: How to Create a New Condition

1. Insert a report with parameters in the HTML canvas.
2. Drag control objects on the Parameters tab to create a chain.

   Chains are represented by lines connecting control objects on the Parameters tab.

   Chaining controls will populate parameters with values at run time, based on values selected in prior controls on the chain.
3. Click a link in the chain.

   The Settings panel shows the bound control values and the *Default* condition settings for the link in the chain.
4. Click the *New* icon to create a multiple condition for the chain.

Condition*(n)* is created, where *(n)* is the number, and added to the Conditions drop-down list, and the Selected values section is activated. You may type in a unique condition name.

5. You may type in a unique condition name, choose Selected values, and set the condition settings for the new condition.
6. Optionally, you may click the *Delete* button to remove the selected condition from the list.

   Note: *Default*, the initial condition, cannot be deleted.

Procedure: How to Select the Action for a Condition

1. Insert a report with parameters in the HTML canvas.
2. Drag control objects on the Parameters tab to create a chain.

   Chains are represented by lines connecting control objects on the Parameters tab.

   Chaining controls will populate parameters with values at run time, based on values selected in prior controls on the chain.
3. Click a link in the chain.
4. In the Setting panel, select the action for the condition from the *Actions* drop-down list. For example, to hide the control being chained to, select *Hide*.

   Populate, show is the default option.

When running the HTML page, the action for the chained control is applied.

Procedure: How to Select the Values Compare Operator for a Condition

1. Insert a report with parameters in the HTML canvas.
2. Drag control objects on the Parameters tab to create a chain.

   Chains are represented by lines connecting control objects on the Parameters tab.

   Chaining controls will populate parameters with values at run time, based on values selected in prior controls on the chain.
3. Click a link in the chain.
4. In the Setting panel, select the chaining logic for the parameter being chained to, in the condition, from the Values compare operator drop-down list.

   Equal is the default option.

The compare operator is applied to the value selected.

Procedure: How to Apply Selected Values With a Multiselect Operator to a Condition

1. Insert a report with parameters in the HTML canvas.
2. Drag control objects on the Parameters tab to create a chain.

   Chains are represented by lines connecting control objects on the Parameters tab.

   Chaining controls will populate parameters with values at run time, based on values selected in prior controls of the chain.
3. Click a link in the chain.
4. In the Settings panel, click the *New* icon to create a multiple condition for the chain.

   Condition*(n)* is created, where *(n)* is the number, and added to the Conditions drop-down list, and the Selected values section is activated. You may type in a unique condition name.
5. You may type selected values in the input box or click the ellipsis button to select values from the list.

   The list of values that appears is based on the values of the prior bound control in the chain.

   Tip: You may also use the pop-up icons to select a field and close the pop-up dialog box. The green icon is OK, the red icon is Cancel, double-clicking a value will select the value and close the dialog box without using any button, and pressing the Esc key will cancel the dialog box without using any button.

   When selected values are entered, the Multiselect operator field is activated.
6. Select the chaining logic for the selected values from the *Multiselect operator* drop-down list.

   - *One of* is based on one of the values shown in the Selected values, being selected in the prior control, in the chain. This is the default selection.
   - *All of* is based on the value of all of the Selected values, being selected in the prior control, of the chain.

The selected values and multiselect operator are applied to the condition.

Procedure: How to Resolve Parameters for a Condition

It is recommended that you populate the controls first, before chaining. When you populate first, certain information is obtained that allows the determination of the best choice for *Resolves parameter*. If you chain first and then populate, the information cannot be obtained because the chaining is already established. If you chain first, you must manually set *Resolves parameter*.

1. Insert a report with parameters in the HTML canvas.
2. Drag control objects on the Parameters tab to create a chain.

   Chains are represented by lines connecting control objects on the Parameters tab.

   Chaining controls will populate parameters with values at run time, based on values selected in prior controls on the chain.
3. Click a link in the chain.
4. In the Setting panel, click the *Resolves parameter* ellipsis button to select a parameter name to resolve. If a custom procedure that has a filter (or filters) populates the control, the values list shows the parameters from the filters. If a data source populates the control, the values list shows all of the fields from the data source shown in the Object Inspector.

   The value in the Resolves parameter field should be either:

   - The field name that limits the values for the next control in the chain, if a data source populates the control.
   - The parameter name from the procedure, if a procedure populates the control.

   In most cases, this value will be populated by default and will not need to be changed.

   Tip: You may also use the pop-up icons to select a field and close the pop-up dialog box. The green icon is OK, the red icon is Cancel, double-clicking a value will select the value and close the dialog box without using any button, and pressing the Esc key will cancel the dialog box without using any button.

The parameter value is resolved in the chain if no filter is specified.

Procedure: How to Select the Compare Operator for the Parameter

An example of when to apply chaining logic is when a form offers two lists of dates so that you can select a FROM/TO date range. By chaining these parameters together and applying the Greater than parameter compare operator, this ensures that when a date is selected for the FROM parameter, only dates that follow the FROM date display in the TO date control, eliminating the possibility of selecting an invalid date range.

1. Insert a report with parameters in the HTML canvas.
2. Drag control objects on the Parameters tab to create a chain.

   Chains are represented by lines connecting control objects on the Parameters tab.

   Chaining controls will populate parameters with values at run time, based on values selected in prior controls on the chain.
3. Click a link in the chain.
4. In the Setting panel, select the chaining logic option from the Compare operator drop-down list. This sets the compare operator to populate the control.

   Equal is the default option.

The compare operator is applied to the parameter selected.

Procedure: How to Enable Cache Processing for Chained Values

You may enable cache processing for chained values in two ways:

- ❏ Enable the caching option for the HTML page and all objects on the page.
- ❏ Enable cache run time data for a dynamic control or a condition.

  A chain contains conditions for each link in the chain. The conditions are linked to the values being bound to the control object. If you change the options for the condition, it will also be applied to the control, and vice versa.

Caching options are turned off, by default.

1. To enable caching options for all objects on the HTML page, in the Developer Workbench Options dialog box:
   - ❏ Select the *HTML Page* tab.
   - ❏ Select *Default caching option*.
   - ❏ Click *OK* to close the Developer Workbench Options dialog box.
2. To enable caching options for a dynamic control or condition:

A chain contains conditions for each link in the chain. The conditions are linked to the values being chained to the control object. If you change the options for the condition, it will also be applied to the control, and vice versa.

For a dynamic control:

1. Create a dynamic input control to supply parameter values.
2. Select the dynamic control object from the Parameters tab.

   The Settings panel opens, showing the dynamic control options.
3. Select *Cache run time data* to cache the run time data for the selected input control.

   Note: This setting overrides the *Default caching option* from the HTML Page tab, in the Developer Workbench Options dialog box.
4. Select the center of the control object and drag the control to the center of the next control object.

   When binding controls, the conditions inherit the values set in the dynamic control settings.

For a dynamic condition:

1. Click a link on the chain to open the Settings panel for the condition.
2. Select *Cache run time data* to cache the run time data for the selected input control. This option is only available for dynamic controls.

   This setting overrides the *Default caching option* from the HTML Page tab, in the Developer Workbench Options dialog box.

When running the HTML page, data for the chained value is cached to improve performance.

Using JavaScript Code with HTML Canvas Pages

Although the HTML canvas is fully integrated with JavaScript, it is suggested that you do not create custom JavaScript that manipulates the HTML canvas generated controls, as Db2 Web Query cannot support such custom JavaScript code. Additionally, there is no guarantee that the JavaScript code will work correctly in future releases.

Note:

- The HTML canvas run time is a set of JavaScript files. You should not be calling the functions within them directly. Should you call these functions directly, your code may not function in future releases and IBM cannot be held responsible.

- If you want JavaScript to run after a page loads completely, but before any reports are executed, you need to create a function called onInitialUpdate (this function was called onInitialUpdate() in previous releases). If you have another function, in that function, you can call IbComposer_onInitialUpdate(). Your code should be added inline in the HTML file after the line:

  ```
  //End function window_onload
  ```

  The HTML canvas run time will call onInitialUpdate() if it exists.

Function: IbComposer_removeSelectOption

IbComposer_removeSelectOption removes values from a listbox, drop down, or the 'from' listbox of a double list control.

Syntax: How to Remove Listbox or Drop Down Values

```
IbComposer_removeSelectOption(controlID,arr[]);
```

controlID

Alphanumeric

Is the unique identifier of the control from which values are obtained.

arr[]

Alphanumeric Array

Is the array that contains single or multiple values.

Note: When the IbComposer_removeSelectOption method is called, the values specified by the second parameter (arr[]) will be removed.

Example: Removing Values From a Listbox

```
function button1_onclick(event) {
   var eventObject = event ? event : window.event;
   var ctrl = eventObject.target ? eventObject.target :
   eventObject.srcElement;
   // TODO: Add your event handler code here
   var readVals = [];
   readVals = IbComposer_getCurrentSelection('listbox1');
   IbComposer_removeSelectOption('combobox1',readVals);
}
```

Function: IbComposer_runAnimation

IbComposer_runAnimation runs the animation defined by the user in the Tasks and Animations panel.

Syntax: How to Run an Animation

```
IbComposer_runAnimation(name);
```

name

Alphanumeric

Is the name of the animation specified by the user in the Tasks and Animations panel.

Example: Running an Animation

```
function submit1_onclick(event) {
   IbComposer_runAnimation('animation1');
}
```

Function: IbComposer_triggerExecution

IbComposer_triggerExecution allows the user to execute a specific task defined in the Tasks section of the Tasks and Animations panel.

Syntax: How to Execute a Specific Task

```
IbComposer_triggerExecution(taskName,index [,paramName, paramValue]);
```

where:

taskName

Alphanumeric

Is the name of the task specified by the user in the Tasks section of the Tasks and Animations panel in the HTML canvas.

index

Numeric

The second parameter (index) is the number of the action of that task that will run first. All actions to the end will run, as well.

paramName, paramValue

Alphanumeric

Optionally, are the name and value of a parameter to be applied when the task is executed. You can specify multiple parameters by using multiple name-value pairs.

Example: Executing a Task

The following example executes the first task, called task1, when a button called submit1 is clicked.

```
function submit1_onclick(event) {
   IbComposer_triggerExecution('task1',1);
}
```

Example: Executing a Task From a Procedure

The following syntax appears in the StyleSheet section of a report procedure. It creates a JavaScript drilldown, launched from a menu item labeled Drill to InfoWindow, that uses the IbComposer_triggerExecution function to execute a task called task2, which has an index value of 1. It sets the value of the parameter &BUSINESS_SUB_REGION to the value selected from the first column in the report (N1). The report is loaded in a frame on the page, as identified by ib_frameName, that has the unique identifier report1.

```
TYPE=DATA,
     COLUMN=N1,
     DRILLMENUITEM='Drill to InfoWindow',
          JAVASCRIPT=parent.IbComposer_triggerExecution( \
     'task2' \
     '1' \
     'BUSINESS_SUB_REGION' \
     N1 \
     'ib_frameName' \
     'report1' \
     ),
          TARGET='_parent',
$
```

The report can then be added to an HTML page in a report component called report1. You can then create a task called task2, which is the first task after the load task, in the Tasks & Animations panel. Use a Trigger Type of TBD to set the task to execute when a drill-down link in the report is clicked. You can then specify a request in the Requests/Actions section to be executed using the selected parameter value when a drill-down link is clicked.

Function: IbComposer_getRequestRefProcedure

IbComposer_getRequestRefProcedure returns the name of the procedure, given the unique identifier of the request referencing the procedure. This was formerly available in the myXmlRoot document.

Syntax: **How to Return the Procedure Name**

```
IbComposer_getRequestRefProcedure(Drilldown);
```

where:

Drilldown

Alphanumeric

Is the name of the request provided.

Example: **Returning a Procedure Name**

```
function button1_onclick(event) {
   IbComposer_getRequestRefProcedure('Drilldown');
}
```

Function: IbComposer_getRequestTarget

IbComposer_getRequestTarget returns the name of all the target frames used by the request. This was formerly available in the myXmlRoot document.

Syntax: **How to Return the Target Frame Names**

```
IbComposer_getRequestTarget(Drilldown,bShow);
```

where:

Drilldown

Alphanumeric

Is the name of the request provided.

bShow

Boolean

Is an operator that can be set to true to show the names, or set to false to hide them.

Example: **Returning Target Frame Names**

```
function button1_onclick(event) {
   IbComposer_getRequestTarget('Drilldown','true');
}
```

Function: IbComposer_populateDynamicCtrl

IbComposer_populateDynamicCtrl allows you to determine from which control data is drawn from to populate a destination control.

Syntax: How to Populate a Control Dynamically

```
IbComposer_populateDynamicCtrl(controlID,fromControlId);
```

where:

controlID

Alphanumeric

Is the unique identifier of the control from which values are obtained.

fromControlId

Alphanumeric

Is the unique identifier of the control from which values are obtained, when two or more controls are chained to a destination control. For example, if listbox1 and listbox2 are both chained to listbox3, by default, the values of listbox3 will be determined by listbox1. You can use IbComposer_populateDynamicCtrl('listbox3','listbox2') to make listbox2 determine the values of listbox3. This identifier is optional.

Example: Populating a Control Dynamically

```
function button3_onclick(event) {
   var acc = IbComposer_populateDynamicCtrl('listbox3','listbox2');
   acc.selectNextPage();
}
```

Function: IbComposer_getComponentById

IbComposer_getComponentById obtains a component by using its ID.

Syntax: How to Obtain a Component by Using Its ID

```
IbComposer_getComponentById(controlID);
```

where:

controlID

Alphanumeric

Is the unique identifier of the control from which values are obtained.

Example: **Obtaining the Accordion Report By Using Its ID**

```
function button3_onclick(event) {
   var acc = IbComposer_getComponentById('accordion1');
   acc.selectNextPage();
}
```

Function: IbComposer_getCurrentSelection

IbComposer_getCurrentSelection retrieves the current selected values from a control.

Syntax: **How to Retrieve the Current Selected Value From a Control**

```
IbComposer_getCurrentSelection(controlID,[layer],[bDateObj]);
```

where:

controlID

Alphanumeric

Is the unique identifier of the control from which values are obtained.

layer

Integer

Is an optional parameter used to specify the layer number in a Multi source Tree control if a Multi source Tree control is being used. The layer number starts with 1 for the first layer.

bDateObj

Boolean

Is an optional operator that can be set to true to return a date object or to false to return a string. If you do not specify a value for this operator, it returns a string.

Example: **Retrieving the Current Selected Value for a Drop Down List**

```
function button1_onclick(event) {
   var values = IbComposer_getCurrentSelection('combobox1');
   for(var i = 0; i < values.length; i++)
   alert(values[i]);
}
```

Function: IbComposer_getCurrentSelectionEx

IbComposer_getCurrentSelectionEx retrieves the current selected actual or display values from a control. The function can also be used to get the index values for List Boxes, Drop Down Lists, and Double Lists.

Syntax: How to Retrieve the Current Selected Value, Actual Value, or Display Value From a Control

```
IbComposer_getCurrentSelectionEx(controlID,[layer]);
```

where:

controlID

Alphanumeric

Is the unique identifier of the control from which values are obtained.

layer

Integer

Is an optional parameter used to specify the layer number in a Multi source Tree control if a Multi source Tree control is being used. The layer number starts with 0 for the first layer.

Example: Retrieving the Current Selected Index Value, Actual Value, and Display Value for a Multi-Select List

```
function button1_onclick(event) {
   var values = IbComposer_getCurrentSelectionEx('combobox1');
   for(var i = 0; i < values.length; i++)
{
   alert("Index Value: " + values[i].getIndex() + "\n" +
   "Actual Value: " + values[i].getValue() + "\n" +
   "Display Value: " + values[i].getDisplayValue());
}
```

Function: IbComposer_getClickedRow

IbComposer_getClickedRow retrieves the current clicked row of an HTML table.

Syntax: How to Retrieve the Current Clicked Row of an HTML Table

```
IbComposer_getClickedRow(clickEventTarget);
```

where:

clickEventTarget

Alphanumeric

Is the name of the event for the clicked row.

Example: **Retrieving the Current Clicked Row of an HTML Table**

```
function IbComposer_getClickedRow(clickEventTarget) {
   return getCurrentClickedRow(clickEventTarget);
}
```

Function: IbComposer_getClickedColumn

IbComposer_getClickedColumn retrieves the current clicked column of an HTML table.

Syntax: **How to Retrieve the Current Clicked Column of an HTML Table**

```
IbComposer_getClickedColumn(clickEventTarget);
```

where:

clickEventTarget

Alphanumeric

Is the name of the event for the clicked column.

Example: **Retrieving the Current Clicked Column of an HTML Table**

```
function IbComposer_getClickedColumn(clickEventTarget) {
   return getCurrentClickedColumn(clickEventTarget);
}
```

Function: IbComposer_getClickedCellValue

IbComposer_getClickedCellValue retrieves the current clicked cell value of an HTML table.

Syntax: **How to Retrieve the Current Clicked Cell Value of an HTML Table**

```
IbComposer_getClickedCellValue(clickEventTarget);
```

where:

clickEventTarget

Alphanumeric

Is the name of the event for the clicked cell.

Example: **Retrieving the Current Clicked Cell Value of an HTML Table**

```
function IbComposer_getClickedCellValue(clickEventTarget) {
   return getClickedCellValue(clickEventTarget);
}
```

Function: IbComposer_setCurrentSelection

IbComposer_setCurrentSelection sets the current selected values for control parameters.

Syntax: How to Set the Current Selected Value for a Control

```
IbComposer_setCurrentSelection(controlID,arrValues,bUpdateDependencies);
```

where:

controlID

Alphanumeric

Is the unique identifier of the control for which to set the values.

arrValues

Array

Is the array of values to be set.

bUpdateDependencies

Boolean

Is an operator that can be set to true to update chained controls and triggered events. The default is false.

Example: Setting the Current Selected Value for a List Box

The following example selects the values ITALY and JAPAN from a multiselect list box called listbox1.

```
function button1_onclick(event) {
   var arr = [];
   arr.push('ITALY');
   arr.push('JAPAN');
   IbComposer_setCurrentSelection('listbox1',arr,'false');
}
```

Function: IbComposer_setCurrentSelection2

IbComposer_setCurrentSelection2 sets the current selected values for radio button and check box controls using an index value.

Syntax: How to Set the Current Selected Value for a Radio Button or Check Box Control

```
IbComposer_setCurrentSelection2(controlID,index,bUpdateDependencies);
```

where:

controlID

Alphanumeric

Is the unique identifier of the control for which to set the values.

index

Numeric

The index value of the item to select. The first item in the control object has an index value of 0.

bUpdateDependencies

Boolean

Is an operator that can be set to true to update chained controls and triggered events. The default is false.

Example: Setting the Current Selected Value for a List Box

The following example uses a button called button1 to select the third value, indicated by the index value of 2, in a radio button control called radio1.

```
function button1_onclick(event) {
   var eventObject = event ? event : window.event;
   var ctrl = eventObject.target ? eventObject.target :
eventObject.srcElement;
   // TODO: Add your event handler code here
   IbComposer_setCurrentSelection2("radio1",2,false);
}
```

Function: IbComposer_execute

IbComposer_execute executes a report or chart.

Syntax: How to Execute a Report or Chart

```
IbComposer_execute(reportID,[outputTarget]);
```

where:

reportID

Alphanumeric

Is the unique identifier of the report or chart to execute.

outputTarget

Alphanumeric

Is the optional parameter to set the target of the output. Is one of the following:

- The name of a frame.
- *'_blank'*.
- *'_target'*.
- The name of a new window.

Example: Executing a Report in a New Window

```
function button3_onclick(event) {
   IbComposer_execute('report1','newwin');
}
```

Function: IbComposer_isSelected

IbComposer_isSelected determines if a control or value is selected.

Syntax: How to Determine If a Control or Value Is Selected

```
IbComposer_isSelected(controlID,[testValue]);
```

where:

controlID

Alphanumeric

Is the unique identifier of the control being tested.

testValue

Alphanumeric

Is the optional parameter the control is being checked against.

Example: Determining If a Check Box Is Selected

```
function checkbox1_onclick(event) {
   var curValue = IbComposer_isSelected('checkbox1');
   IbComposer_showHtmlElement('form1',curValue);
}
```

Function: IbComposer_showHtmlElement

IbComposer_showHtmlElement shows or hides an HTML element.

Syntax: **How to Show or Hide an HTML Element**

```
IbComposer_showHtmlElement(elementID,bShow);
```

where:

elementID

Alphanumeric

Is the unique identifier of the element which is shown or hidden.

bShow

Boolean

Is an operator that can be set to true to show the element and false to hide it.

Example: **Showing or Hiding a Check Box**

```
function checkbox1_onclick(event) {
   var curValue = IbComposer_isSelected('checkbox1');
   IbComposer_showHtmlElement('form1',curValue);
}
```

Function: IbComposer_enableHtmlElement

IbComposer_enableHtmlElement enables or disables an HTML element.

Syntax: **How to Enable or Disable an HTML Element**

```
IbComposer_enableHtmlElement(elementID,bEnable);
```

where:

elementID

Alphanumeric

Is the unique identifier of the element which is enabled or disabled.

bEnable

Boolean

Is an operator that can be set to true to enable the element and false to disable it.

Example: Enabling or Disabling Elements

```
function checkbox2_onclick(event) {
   IbComposer_enableHtmlElement('listbox1',
IbComposer_isSelected('checkbox2','country'));
   IbComposer_enableHtmlElement('combobox1',
IbComposer_isSelected('checkbox2','car'));
   IbComposer_enableHtmlElement('listbox2',
IbComposer_isSelected('checkbox2','model'));
   IbComposer_enableHtmlElement('combobox2',
IbComposer_isSelected('checkbox2','dcost'));
}
```

Function: IbComposer_ResetDownChainControls

IbComposer_ResetDownChainControls resets the controls down the chain from the current control to have correct corresponding values.

Syntax: How to Reset Chain Controls

```
IbComposer_ResetDownChainControls(ctrl);
```

where:

ctrl

Alphanumeric

Is the unique identifier of the first control.

Example: Resetting the Chain Started by a List Box

```
function button4_onclick(event) {
   var arr = [];arr.push('ENGLAND');
   IbComposer_setCurrentSelection('listbox1',arr,'false');
   IbComposer_ResetDownChainControls('listbox1');
}
```

Function: IbComposer_selectTab

IbComposer_selectTab selects the tab specified by the tabNumberToSelect and makes it the active tab.

Syntax: How to Select a Tab and Make It Active

```
IbComposer_selectTab(tabControlID,tabNumberToSelect);
```

where:

tabControlID

Alphanumeric

Is the unique identifier of the tab control being made active.

tabNumberToSelect

Integer

Is the number of the tab to make active.

Example: **Making a Tab Active**

```
<FORM id=form1 onsubmit="OnExecute(this);
   IbComposer_selectTab('tab1',1) name="form1">
```

Function: IbComposer_selectTemplateTab

IbComposer_selectTemplateTab selects a tab on a template page and makes it the active tab.

Syntax: **How to Select a Template Tab and Make It Active**

```
IbComposer_selectTemplateTab(tabId);
```

where:

tabId

Alphanumeric

Is the unique identifier of the tab control being made active.

Example: **Making a Template Tab Active**

```
function submit1_onclick(event) {
   IbComposer_selectTemplateTab('tab5');
}
```

Function: IbComposer_goToPageOfMultiPageControl

IbComposer_goToPageOfMultiPageControl selects a page in a multipage control, such as a tab, accordion, or window.

Syntax: **How to Select a Page in a Multipage Control**

```
IbComposer_goToPageOfMultiPageControl(controlID,page)
```

where:

controlID

Alphanumeric

Is the unique identifier of the control from which values are obtained.

page

Alphanumeric

Specifies a page number (for example 1, 2, and so on), or true for the previous page and false for the next page.

Example: Selecting a Page for a Multipage Control

The following example shows how to select a page in a window multipage control. In this example, the control will start on page 3.

```
function button2_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
IbComposer_goToPageOfMultiPageControl('windowPanel1', '3');
}
```

The following example shows how to select the previous page in a window multipage control.

```
function button2_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
IbComposer_goToPageOfMultiPageControl('windowPanel1', 'true');
}
```

The following example shows how to select the next page in a window multipage control.

```
function button2_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
IbComposer_goToPageOfMultiPageControl('windowPanel1', 'false');
}
```

Function: IbComposer_getAllAmpersValues

IbComposer_getAllAmpersValues is used to retrieve the current selected values from all the controls on your page layout. It then takes those values and assembles them as a string that can be added to the end of a URL call. An example of this would be having a REGION control and multiselecting MidEast, NorthEast, and NorthWest. It will assemble these selections as shown below:

```
&REGION=%27MidEast%27%20OR%20%27NorthEast%27%20OR%20%27NorthWest%27
```

This function can be used in conjunction with the Business Intelligence Portal, where the generated string is appended to all Business Intelligence Portal calls that run reports or charts. This allows the parameter values to affect all portal components, even if new ones are added or existing ones are removed at run time.

Syntax: How to Retrieve all Parameter Values

```
IbComposer_getAllAmpersValues([verifySelection]);
```

where:

verifySelection

Boolean

Is an optional parameter. When true and when the Selection required property for the control is set to *Yes*, this returns an empty string for the parameter controls that do not have a selection made.

Note: All controls have the Selection required property. This property is set to *Yes*, by default. If a control has no valid selection made at run time, a red box appears around it and the following status bar message displays:

```
Please make required selections
```

Example: Retrieving a List of All Parameters Selected in a Report

```
function button1_onclick(event) {
   var val = IbComposer_getAllAmpersValues();
   alert(val);
   OnExecute(event);
}
```

Function: IbComposer_showLayer

IbComposer_showLayer shows or hides the specified layer.

Syntax: How to Show or Hide a Layer

```
IbComposer_showLayer(layerName,bShow);
```

where:

layerName

Alphanumeric

Is the name of the layer, which will be shown or hidden.

bShow

Is an operator that can be set to true to show the layer or false to hide it.

Example: Showing or Hiding a Layer

```
function button1_onclick(event) {
   var eventObject = event ? event : window.event;
   var ctrl = eventObject.target ? eventObject.target :
   eventObject.srcElement;
   // TODO: Add your event handler code here
   IbComposer_showLayer('Customers','true');
}
```

Function: IbComposer_preventModifyIFrameDocument

IbComposer_preventModifyIFrameDocument can prevent CSS modification of any iframe within a document, by calling it from onInitialUpdate.

Syntax: How to Prevent Modification of a Document within an IFrame

```
function onInitialUpdate() {
   IbComposer_preventIFrameUpdates();}
```

Function: IbComposer_setCalendarDatesRange

IbComposer_setCalendarDatesRange sets the range for the target calendar, given the date on the source calendar and a range in days.

Syntax: How to Set a Date Range for a Target Calendar

```
IbComposer_setCalendarDatesRange(elementId, fromDate, toDate)
```

where:

elementID

Alphanumeric

Is the unique identifier of the target calendar control for which you want to set the date range.

fromDate

Date object

Is the value of the date from the source calendar.

Note: This can be obtained by using the IbComposer_getCurrentSelection function. This function should have the third parameter set to True to return a Date object.

toDate

Date object

This value must be calculated using fromDate and the number of days needed to show the range.

Example: Setting the Date Range for a Target Calendar

```
function button1_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
     var curDate = IbComposer_getCurrentSelection('calendar1', 0, true);
       var fromDate= new Date(curDate);
       var toDate = new Date();
       var nOfDays= IbComposer_getCurrentSelection('slider1');
       if (isNaN (nOfDays-0)&& nOfDays != null) {
            alert("Please enter a numeric value!!");
       }
       else {
           var time = fromDate.getTime()+(nOfDays*3600000 * 24);
           toDate.setTime(time);
           IbComposer_setCalendarDatesRange("calendar2", fromDate,
           toDate);
       }
}
```

Function: IbComposer_goToPortalPage

You can use the IbComposer_goToPortalPage function to navigate to a specified page within the same collaborative portal. Create an HTML page with a trigger for the IbComposer_goToPortalPage function, configure the IbComposer_goToPortalPage function to target a portal page, and add the HTML page to a portal page in the same collaborative portal as the targeted portal page.

Reference: How to Navigate to a Page in a Collaborative Portal

```
IbComposer_goToPortalPage("Page_Name");
```

where:

Page_Name

Alphanumeric

Is the name of another portal page within the same collaborative portal as the HTML page containing the IbComposer_goToPortalPage function, in double quotation marks (").

Make sure to type the portal page name instead of the title. You can see the name of a portal page in the Name field of the File/Folder Properties panel. Do not include the .page extension in the IbComposer_goToPortalPage function.

Example: Navigating to a Page in a Collaborative Portal

In the following example, clicking a button with the unique identifier button1, which is available on a portal page, opens the page named Page_1, contained in the same collaborative portal.

```
function button1_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
IbComposer_goToPortalPage("Page_1");
}
```

Function: IbComposer_isSelectionRequired

The IbComposer_isSelectionRequired function allows you to determine whether a selection is required for a specified control. If selection is required from the control, the HTML element is returned. If selection is optional, null is returned.

Reference: How to Determine if Selection Is Required

```
IbComposer_isSelectionRequired("controlId");
```

where:

controlId

Alphanumeric

Is the unique identifier of the control whose status you are checking.

Example: Determining if Selection Is Required

In the following example, clicking a button with the unique identifier button1 displays a message that says whether a selection is required in combobox1, based on the value returned by IbComposer_isSelectionRequired. Since the item being checked is an HTML select element, the message is different depending on whether the returned value is [object HTMLSelectElement] or null.

```
function button1_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here

var required = IbComposer_isSelectionRequired("combobox1")
if (required == "[object HTMLSelectElement]") {
    alert("Selection is required")
    }
else if (required == null) {
    alert("No selection required")
    }

}
```

Function: IbComposer_getNumberOfRecordsReturned

The IbComposer_getNumberOfRecordsReturned function produces the number of records returned by a request executed by an Ajax call. To use the IbComposer_getNumberOfRecordsReturned function, create a task that executes a request using an Ajax call. The value in the Function name field for the task, in the Tasks & Animations panel, should be the name of the callback function. This function and the IbComposer_getNumberOfRecordsReturned function should have the same result object as an argument.

Reference: How to Return the Number of Records in a Request

```
IbComposer_getNumberOfRecordsReturned(resultObject);
```

where:

resultObject

Alphanumeric

Is a result object that matches the argument of the callback function for the Ajax call.

Example: Returning the Number of Records in a Request

For the following example, a task has been created in the Tasks & Animations panel to execute an external procedure request, with Ajax call set as the Target type. The Ajax call is specified to use a callback function named showResult. The showResult function then passes the value, called result, to the IbComposer_getNumberOfRecordsReturned function, the resulting value of which is shown in a message. This value is the number of records in the request.

```
function showResult(result)
{
alert(IbComposer_getNumberOfRecordsReturned(result));
}
```

Function: IbComposer_doSelectionByTableRow

The IbComposer_doSelectionByTableRow function returns the values of specified fields in the selected row of a table. You can use another JavaScript function, such as IbComposer_setCurrentSelection, to enable selection values in the table.

Reference: How to Retrieve Values From a Selected Row in a Table

```
IbComposer_doSelectionByTableRow("tableId",fieldnames,setSelection,display
Only);
```

where:

tableId

Alphanumeric

Is the unique identifier of the table to return values from.

fieldnames

Is an array of field names, for example:

```
["field1", "field2", "field3"]
```

dontSetSelection

Boolean

Determines whether to select values so that you can pass them to chained controls. When true, values are not selected. When false, the values are selected.

displayOnly

Boolean

When true, selections are display values. When false, selections are values from the data source.

Example: Retrieving Values From a Selected Row in a Table

In the following example, clicking a button with the unique identifier button1 displays a message containing the values of the Country and Model fields for the selected row in table1, if a row has been selected.

```
function button3_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here

var fields = ["CAR.ORIGIN.COUNTRY", "CAR.CARREC.MODEL"];
var values = IbComposer_doSelectionByTableRow("tablectrl1", fields, true,
true);

if(values && values.length)
{
for(var i = 0; i <values.length; i++)
    alert(values[i]);
}
}
```

Function: IbComposer_loadNextGroup

The IbComposer_loadNextGroup function allows you to load the next group in a table.

Reference: How to Load the Next Group in a Table

```
IbComposer_loadNextGroup("tableId",groupObject,updateDependencies);
```

where:

tableId

Alphanumeric

Is the unique identifier of the table from which to load the next group.

groupObject

Object

Is an object representing the group displayed in the table.

updateDependencies

Boolean

If true, controls chained to the table are updated when IbComposer_loadNextGroup is executed. If false, they are not updated.

Example: Loading the Next Group in a Table

In the following example, clicking a button with the unique identifier next loads the next group of the table, called tablectrl1, using the IbComposer_loadNextGroup function. A button called prev, which executes the IbComposer_loadPrevGroup function, is not enabled until you have executed IbComposer_loadNextGroup for the first time, so that it is only available if there is a previous group to display. The next button is not available if the last group in the table is displayed.

In order to execute properly, a variable called numOfRows has been used to set the length of the displayed group to 3 rows.

```
var numOfRows=3;

function tablectrl1_onbeforeload(ctrl,arrValuesToLoad) {
 if(IbComposer_isCompleteReloading("tablectrl1") &&
arrValuesToLoad.length>numOfRows)
{
IbComposer_enableHtmlElement("prev",false);
IbComposer_enableHtmlElement("next",true);
        return numOfRows;
    }
return false;
}

function next_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
var groupObj = {};
IbComposer_loadNextGroup("tablectrl1", groupObj, false);
IbComposer_enableHtmlElement("prev",true);
if(groupObj.end== groupObj.total)
    IbComposer_enableHtmlElement("next",false);

}
```

Function: IbComposer_loadPrevGroup

The IbComposer_loadPrevGroup function allows you to load the previous group in a table.

Reference: How to Load the Previous Group in a Table

```
IbComposer_loadPrevGroup("tableId",groupObject,updateDependencies);
```

where:

tableId

Alphanumeric

Is the unique identifier of the table from which to load the next group.

groupObject

Object

Is an object representing the group displayed in the table.

updateDependencies

Boolean

If true, controls chained to the table are updated when IbComposer_loadPrevGroup is executed. If false, they are not updated.

Example: Loading the Previous Group in a Table

In the following example, clicking a button with the unique identifier prev loads the previous group of the table, called tablectrl1, using the IbComposer_loadPrevGroup function. The prev button is not available if you are viewing the first group in the table, so that it is only available if there is a previous group to display. You must click the button called next, which executes the IbComposer_loadNextGroup function, in order to enable it.

In order to execute properly, a variable called numOfRows has been used to set the length of the displayed group to 3 rows.

```
var numOfRows=3;

function tablectrl1_onbeforeload(ctrl,arrValuesToLoad) {
 if(IbComposer_isCompleteReloading("tablectrl1") &&
arrValuesToLoad.length>numOfRows)
{
IbComposer_enableHtmlElement("prev",false);
IbComposer_enableHtmlElement("next",true);
        return numOfRows;
    }
return false;
}

function prev_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
var groupObj = {};
IbComposer_loadPrevGroup("tablectrl1", groupObj, false);
IbComposer_enableHtmlElement("next",true);
if(groupObj.begin== 1)
    IbComposer_enableHtmlElement("prev",false);

}
```

Function: IbComposer_isCompleteReloading

The IbComposer_isCompleteReloading can be used to check if an element has finished reloading. It returns true if the element has reloaded, and false if it has not.

Reference: How to Check if an Element Has Finished Reloading

```
IbComposer_isCompleteReloading("elementId");
```

where:

elementId

Alphanumeric

Is the unique identifier of the element whose reloading status you are checking.

Example: Checking if an Element Has Finished Reloading

In the following example, clicking a button with the unique identifier button1 displays a message indicating whether or not a table called tablectrl1 has finished reloading.

```
function button1_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
alert(IbComposer_isCompleteReloading("tablectrl1"))
}
```

Function: IbComposer_setCalendarDate

The IbComposer_setCalendarDate function allows you to input a date for a target calendar.

Reference: How to Set the Date of a Target Calendar

```
IbComposer_setCalendarDate("controlId",dateObject);
```

where:

controlId

Alphanumeric

Is the unique identifier of the calendar whose date you are setting.

dateObject

Is a JavaScript date object.

Example: Setting the Date of a Target Calendar

In the following example, clicking a button with the unique identifier button1 sends the current date to a calendar control called calendar1.

```
function button1_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
IbComposer_setCalendarDate("calendar1",new Date());
}
```

In the following example, clicking a button with the unique identifier button1 sends the specified date, June 25, 2015, to the calendar control called calendar1.

Note: In JavaScript, months start at 0 (January), so month 5 in the example below represents June.

```
function button1_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
var today= new Date();
today.setDate(25);
today.setMonth(5);
today.setYear(2015);

IbComposer_setCalendarDate("calendar1",today);
}
```

Function: IbComposer_saveSelectedFile

The IbComposer_saveSelectedFile function allows you to upload a file, selected in an edit box with the Type property set to *File*, to a specified application folder in the Data Servers area. This location is specified using the Copy file to property for the edit box.

Reference: How to Upload a File Selected in an Edit Box

```
IbComposer_saveSelectedFile("controlId");
```

where:

controlId

Alphanumeric

Is the unique identifier of the edit box where you selected the file to upload.

Example: Uploading a File Selected in an Edit Box

In the following example, clicking a button with the unique identifier button1 uploads the file selected in the edit box edit1 to a location set using the Copy file to property. The Type property for edit1 has been set to *File*.

```
function button1_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
IbComposer_saveSelectedFile("edit1")
}
```

Function: IbComposer_loadFromSelectedFile

The IbComposer_loadFromSelectedFile function allows you to display a file, selected in an edit box with the Type property set to *File*, in a container in an HTML page.

Reference: How to Display a File Selected in an Edit Box

```
IbComposer_loadFromSelectedFile(controlId,containerId,callBack);
```

where:

controlId

Alphanumeric

Is the unique identifier of the edit box where you selected the file to upload.

containerId

Alphanumeric

Is the unique identifier of the container in which to display the file selected using *controlId*, or _blank to display the file in a new browser window.

callBack

Alphanumeric

Is a function to call when the selected item loads.

Example: Displaying and Uploading a File Selected in an Edit Box

In the following example, clicking a button with the unique identifier button1 displays a preview of the file selected in the edit box edit1 in an image component called image1 and calls the IbComposer_saveSelectedFile function to save it to a specified location.

```
function button1_onclick(event) {
var eventObject = event ? event : window.event;
var ctrl = eventObject.target ? eventObject.target : eventObject.srcElement;
// TODO: Add your event handler code here
var saveFn = IbComposer_saveSelectedFile("edit1");
IbComposer_loadFromSelectedFile("edit1", "image1", saveFn)
}
```

Function: getImageScale

When creating an Esri map that uses image markers in an HTML page, you can use the getImageScale function to enable the images to scale as you zoom in and out of the map. This helps to prevent the image markers from excessively overlapping as you zoom out of the map, making it more readable, while also maintaining reasonable image sizes as you zoom in. You can also set the maximum and minimum size of the images so that they comfortably fit on your map, regardless of how many markers it contains.

The getImageScale function does not require an event, and can be added directly to your embedded JavaScript.

Reference: How to Configure Image Resizing at Zoom Levels on a Map

```
function getImageScale(layerName,scaleObj)
{
    scaleObj.scale=boolean;
    scaleObj.minSz=minSize;
    scaleObj.maxSz=maxSize;
}
```

where:

`layerName`

Is a static value that does not need to be set. All map layers using image markers are affected by the getImageScale function.

`scaleObj`

This name, for example, *scaleObj*, should match the name of the objects for the scale, minSz, and maxSz properties.

`boolean`

Boolean

Type *true* to enable image resizing, or *false* to disable.

`minSize`

Integer

The minimum height, in pixels, of the image. The minimum size is used when fully zoomed out of the map. This value overrides the sizes set for the image marker in the Settings panel, but the height to width ratio is maintained.

maxSize

Integer

The maximum height, in pixels, of the image. The maximum size is used when fully zoomed into the map. This value overrides the sizes set for the image marker in the Settings panel, but the height to width ratio is maintained.

Example: Configuring Image Resizing at Zoom Levels on a Map

In the following example, the minimum size of all image markers in the map is set to 15 pixels, and the maximum size is set to 60 pixels. When you run the HTML page, the markers expand up to a height of 60 pixels as you zoom in, and shrink down to a height of 15 pixels as you zoom out.

```
function getImageScale(layerName,scaleObj)
{
    scaleObj.scale=true;
    scaleObj.minSz=15;
    scaleObj.maxSz=60;
}
```

Event Handlers

In addition to the typical event handlers, such as onclick, there are two event handlers that enable you to perform an action before a control is loaded and after a control is loaded.

- **onbeforeload.** Performs an action before a control is loaded with values.
- **onafterload.** Performs an action after a control is loaded with values.

Example: Modifying Display Values Before a Control is Loaded

The following example uses the onbeforeload event handler to modify the value *DATSUN* with the display value of *Modified*.

```
//Begin function listbox1_onbeforeload
function listbox1_onbeforeload(ctrl,arrValuesToLoad) {
for(var i = 0; i < arrValuesToLoad.length; i++) {
    //alert(arrValuesToLoad[i].dispValue + "  " + arrValuesToLoad[i].value
+ "  " + arrValuesToLoad[i].selected);
    if (arrValuesToLoad[i].value == 'DATSUN')
{         arrValuesToLoad[i].dispValue = 'Modified';     }
    }
}
//End function listbox1_onbeforeload
```

Example: Copying Values From One Control to Another After Values are Loaded

The following example uses the onafterload event handler to copy values from one control to another.

```
//Begin function listbox1_onafterload
function listbox1_onafterload(ctrl)
{      IbComposer_getComponentById("listbox2").innerHTML =
IbComposer_getComponentById("listbox1").innerHTML }
//End function listbox1_onafterload
```

Note: If a control has more than 100 values, only 100 values will be loaded at a time, repeatedly, until all values are loaded. You will need to modify the JavaScript, as follows:

```
function listbox1_onafterload(ctrl)
{ if (arguments && arguments.length==2 && arguments[1])
IbComposer_getComponentById("listbox2").innerHTML =
IbComposer_getComponentById("listbox1").innerHTML
}
```

This Javascript will be called incrementally after each 100 items are added to the listbox drop-down control. An additional parameter (a boolean value true/false arguments[1]) will be sent. It will be set to *false* for each incremental load of 100 items, and *true* for the final call.

Creating Responsive Web Pages

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.

Unlike a generic HTML page, the containers in a responsive page reorder automatically as the screen size changes. Where a generic page shows partial content with a horizontal scroll bar, the responsive page shows full content, re-ordered, with no horizontal scroll bar.

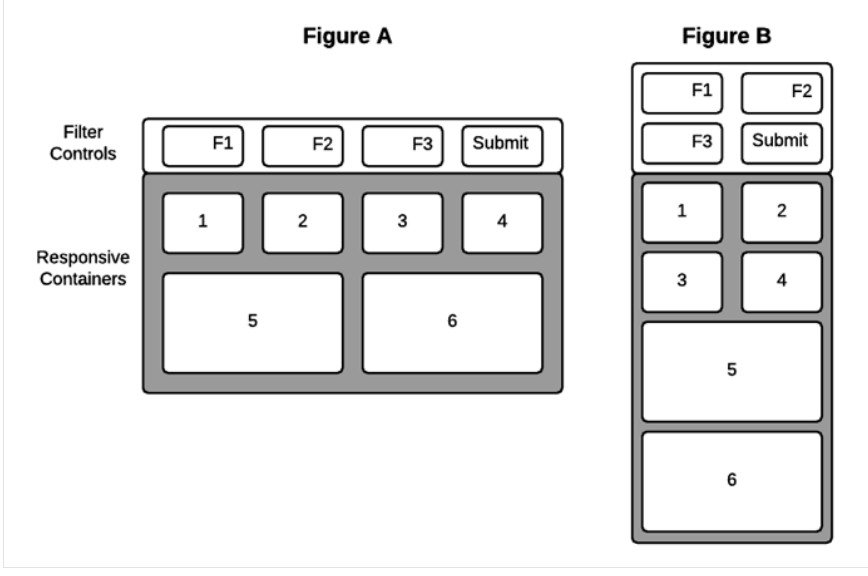
How Responsive Containers Fold

The behavior that makes a page responsive is called *folding*. This section details the way responsive containers fold, or re-arrange, as screen size changes. The order in which responsive containers fold is illustrated in the following image.

Figure A illustrates a responsive layout with a row-based 4-2 format, meaning there are four containers on row one and two containers on row two.

Figure A also illustrates the filter row, containing four controls. At run time, the filter row can either be hidden or displayed.

Figure B illustrates the way containers fold responsively when the screen width decreases. The content in each container resizes accordingly, as the screen width changes.



Note: Objects within a responsive container are positioned absolutely. That is, they do not fold. Responsive containers fold in relation to one another, but the objects inside the containers remain fixed.

Objects inside responsive containers can scale to defined limits, as detailed in *How to Set Minimum and Maximum Panel Sizes* on page 260. However, the objects within a container will not fold.

Procedure: How to Control Folding with the Flow Direction Property

You can control the column/row orientation by which responsive containers re-arrange themselves.

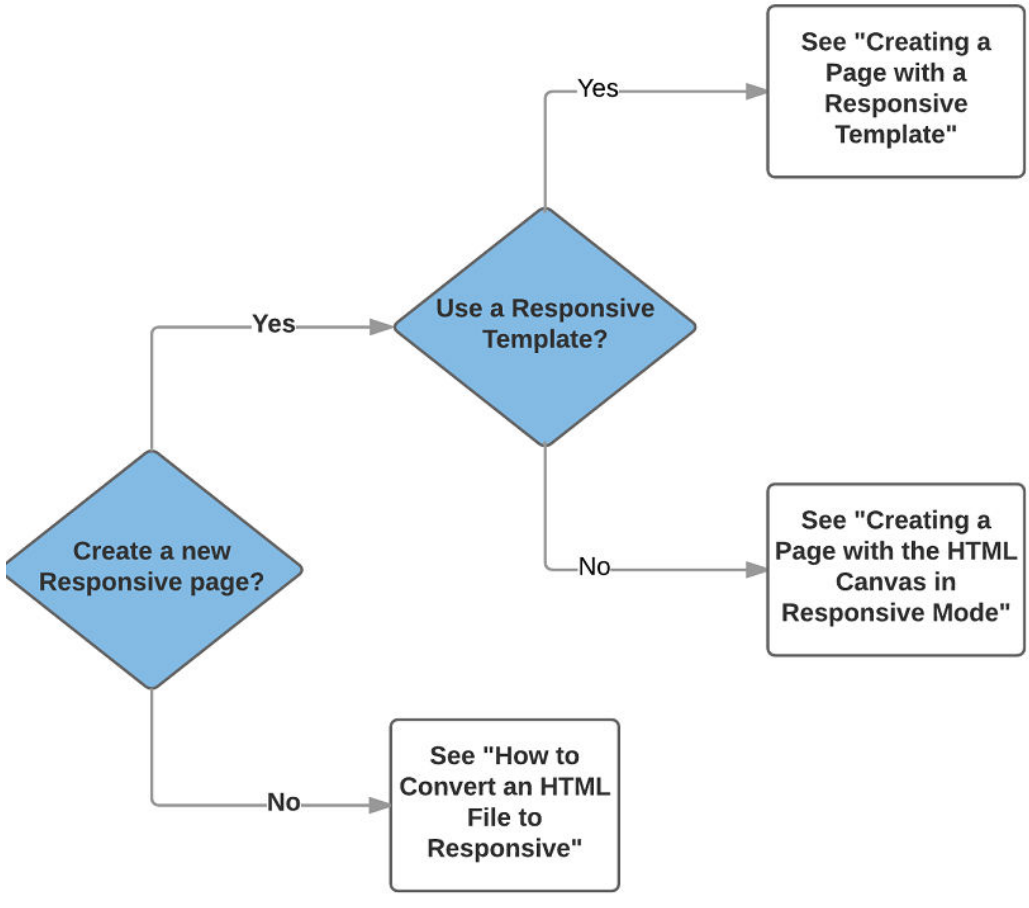
1. Select a container in a responsive HTML page.
2. Click the drop-down arrow in the *Flow Direction* field, in the Responsive design section of the Properties panel and select one of the following:
   - **Not Set.** Allows you to fold objects based on the available space (default).
   - **Row.** Allows you to fold objects horizontally by row.
   - **Column.** Allows you to fold objects vertically by column.

Note: The Flow Direction property does not display for a responsive container that contains an element that is not responsive. For example, a button contained in a responsive panel will disable (hide) the Flow Direction property for the panel. To prevent this, right-click the button and click *Add selected to new container* from the context menu. Then, when you click the original panel, the Flow Direction property is available again.

3. Run the page. Resize the browser to observe the folding.

Selecting a Responsive Method

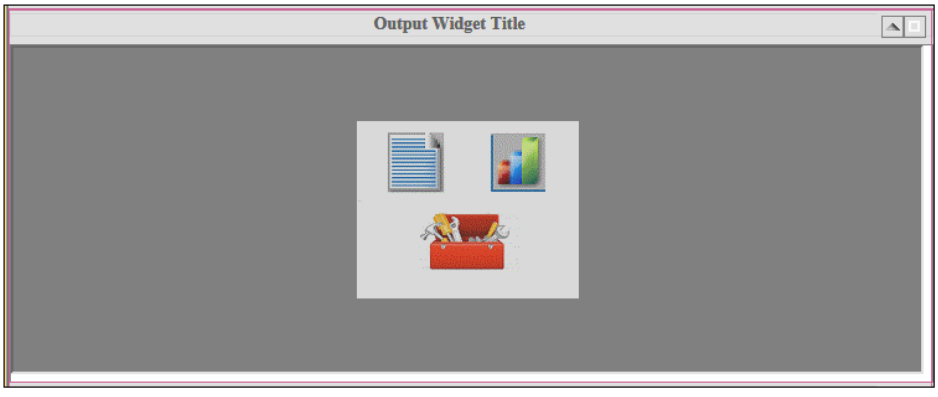
There are two ways to create responsive pages in Developer Workbench: using a Responsive Template, and using the HTML Canvas in Responsive Mode. The following diagram shows the high-level decision tree that lets you quickly find the best way to create your responsive page.



1. **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 use the HTML/Document wizard to specify a layout that is either row-based or column-based.

Key Characteristics

The Output Widget container, shown in the following image, is the key characteristic of a Responsive Template.



The Output Widget container has the following features:

- **Title Bar.** Including an editable text title.
- **Image.** Such as a company logo, that can be inserted on the left side of the title bar.
- **Arrow Toggle.** On the right side of the title bar, to display or hide widget-level controls.
- **Expand Toggle.** On the right side of the title bar, to make the widget contents display full screen or original size.

If you want to create a responsive page with one or more Output Widget containers, the Responsive Template method is the most efficient approach. This method is used only for creating new responsive HTML files. For more information, see *Creating a Page With a Responsive Template* on page 250.

2. **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.

   This method can also be used to convert an existing HTML file to responsive. For more information, see *Creating a Page With the HTML Canvas in Responsive Mode* on page 263.

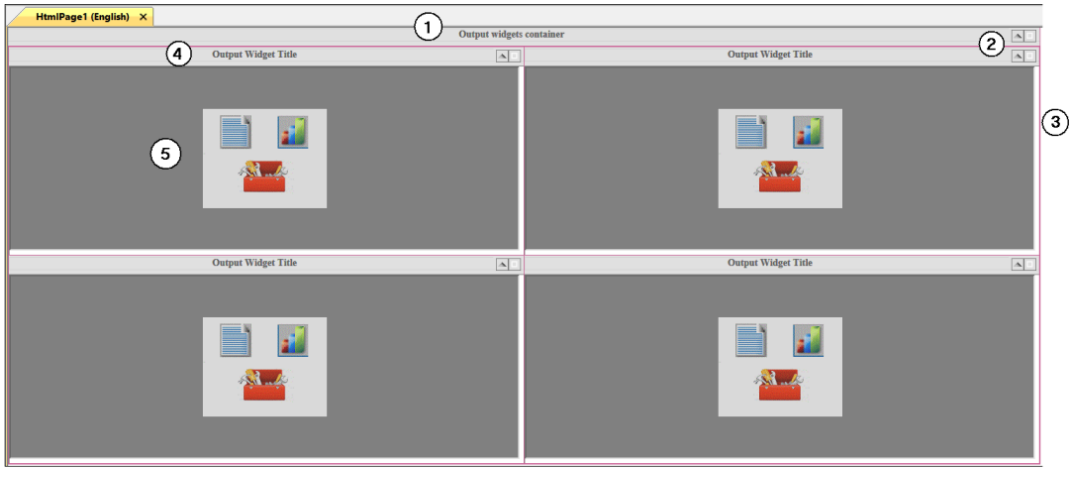
Creating a Page With a Responsive Template

There are six pre-defined responsive templates available through the HTML/Document Wizard. These represent the most commonly used page layouts. You can use the layout from a template as-is, or modify the layout further after initiating it with a template.

As an alternative to using a pre-defined template, the HTML/Document Wizard allows you to create a custom page layout at design time. You can specify the number of rows and columns, and whether the page has a row-based or column-based orientation. These and other page options are detailed in the following sections.

Understanding Responsive Templates

When you select a responsive template from the HTML/Document Wizard, it generates a responsive layout with the specified number of rows (or columns) and widget containers. The parts of the responsive layout are highlighted in the following image, which was generated using the 2-2 responsive template.



Document. Although not enumerated in the previous image, a responsive template generates an HTML document that includes the following responsive properties, editable in the Properties panel.

The following properties are available in the Responsive design section of the Properties panel when the DOCUMENT object is selected:

- **Flow Direction.** Allows you to control how responsive containers rearrange themselves. You can leave this property as Not Set, or specify to fold into rows or columns.
- **Create responsive containers.** When set to Yes, objects added to the document automatically have responsive behavior enabled. This is enabled by default in a responsive template.
- **Global responsive margin.** Defines the margin, in pixels, around all four edges of all responsive containers on the page. Templates have no margins set, by default.

- **Responsive: min width.** Sets the minimum width (in pixels or percentage) beyond which the page size cannot be decreased. The default minimum width is 320 pixels.
- **Responsive: max width.** Sets the maximum width (in pixels or percentage) beyond which the page will not rescale larger. This also defines the width of the canvas at design time. This limit is indicated by a black vertical line on the right side of the canvas, as shown in item 3 in the previous image. The default maximum width is Full screen.

  At run time, the layout populates the full browser.

The following properties are available in the Miscellaneous section of the Properties panel when the DOCUMENT object is selected:

- **Load in iFrame object.** To load the entire page as a widget (a single DIV item) for use in a BI Portal. The default value for this property is *Yes*.

1. Main Widget. Defined as windowPanel1 <DIV> in the Properties panel, the default title is *Output widgets container*.

The following properties are available in the Responsive design section of the Properties panel when the main widget object is selected:

- **Flex grow.** Uses a numeric value to specify how quickly the item expands relative to other items on the HTML page when the window size changes.

The following properties are available in the Output Widget section of the Properties panel when the main widget object is selected:

- **Template orientation.** Determines if the content folds by row or by column. The default orientation is *Row-based*. To change this, you can select *Column-based*.
- **Widget Title.** You can delete the default and type the desired title for the widget that encompasses the entire page contents.
- **Display image.** Enables you to display an image in the upper left corner of the widget. The default setting is *Yes*.
- **Image source.** Used to select the image to be displayed. Once selected, the path to the image is provided.
- **Auto-hide inputs panel.** The row of controls used to filter the content for the page is hidden, by default. To always display the row of controls, select *No*.
- **Select animation.** Enables you to select any animations that have been set for this window panel.

2. Widget display buttons. The page, and each widget within it, has two buttons in the upper-right corner: an arrow and a box. Toggle the arrow to display or hide the row of controls. Toggle the box to expand a widget to full screen or return it to original size.

3. Vertical design boundary. This is a visual indicator on the far right side of the canvas that shows the responsive maximum width set in the Properties panel for the document. The default is 1500 pixels.

4. Widget Title. Defined as windowPanel2 <DIV> in the Properties panel (for the first widget), the default widget title is *Output Widget Title*. You can delete the default and type the desired widget title in the Properties panel. At design time, a red box appears around the widget title and contents to indicate a responsive container.

5. Widget contents. Defined as iframe1 <IFRAME> in the Properties panel (for the first widget), you can right-click to add content to the widget through the following shortcut menu:

- **Reference existing procedure.** To reference an existing procedure in the widget.
- **Map.** This option is not applicable to Db2 Web Query. Inserting a map can be done using the Reference existing procedure option.
- **ESRI.** To create a new emf object in the widget. This option is not applicable to Db2 Web Query. Inserting a map can be done using the Reference existing procedure option.
- **Use as Toolbox.** A general option to use the widget, as needed. For example, you may decide to create a button to execute an action.

The content in pages derived from a responsive template loads in the following order at run time: left to right, top to bottom.

To add a header to your responsive page, right-click in the title bar and select *Add page header* from the shortcut menu. Alternatively, you could use the Add page header option in the HTML/Document Wizard, as detailed in the following procedure.

Procedure: How to Select a Responsive Template

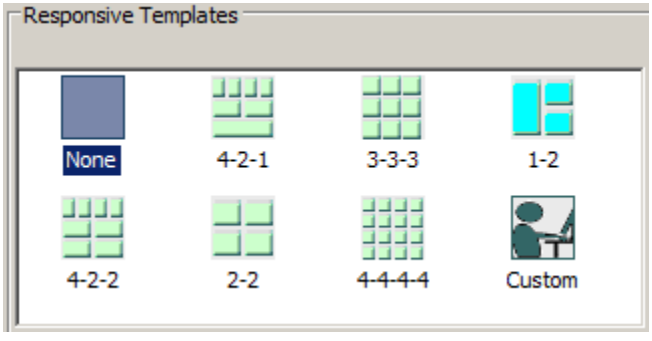
Select a responsive template from the HTML/Document Wizard as follows:

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 more information, see *How to Create a Custom Layout* on page 262.

   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 Developer Workbench to full screen when working with a responsive layout. This lets you access the greatest number of widgets without having to scroll.

Reference: Populating Widgets in a Responsive Template

A layout generated by a responsive template contains blank widgets in responsive containers. This section contains guidelines for editing and adding objects to widgets.

- To add content to a header, click the desired element from the Components tab and drag and drop it on the header area to define it.

  To increase the height of a page header, click the *Title Bar* and drag it down to expose more room above it. Click the header area to highlight the responsive red box. Drag the lower border of the red box down to increase the height of the header.

- To change a widget title, click the default title, *Output Widget Title*, and open the Properties panel. In the Widget Title field, type the new widget title.

- To display an image in the top left corner of a widget title bar, set the Display image value to *Yes*. Use the Image source field to lookup the desired image file.

- To add objects to a widget, right-click in the widget to display the shortcut menu. Click *reference existing procedure*.

  Note: When using a layout generated by a responsive template, you cannot add objects to the canvas outside the layout.

- To add multiple objects to a widget, drag the additional object from the ribbon to the widget. For example, on the *Home* tab, in the *Content* group, click the *Report* button and drag it to a widget that already contains an object, and drop it. A box is automatically drawn within the widget to contain the new report object.

  Note: All of the objects in a responsive container must be responsive. You cannot include responsive and non-responsive objects in the same responsive container.

- To set an equal margin around all four sides of a widget, enter a number of pixels in the Margin property for the widget <IFRAME>.

- To move contents from one widget to another, click the Positioning tab and, in the Responsive Design group, click *Toggle Selection*. Drag contents between widgets as needed and click *Preview Runtime*, on the Utilities tab.

Procedure: How to Add or Remove a Widget

You can change the layout of your responsive page at design time by adding or removing a widget, as detailed below.

1. Open a responsive HTML page.
2. To add a widget, use one of the following methods:
   a. Select a widget next to the location where you want to add a new one. Right-click on the title bar of that widget and select *Insert new widget before* or *Insert new widget after* from the shortcut menu.

      The new widget is added.

   b. Alternatively, you can right-click on the title bar of the HTML page and select *Template configuration...* from the shortcut menu. The Template Configuration dialog box shows the row (or column) numeric configuration. For example, 3-2.

      Change the numeric configuration to include the new widget. For example, to add a new widget to the second row, change the value to 3-3.

      The new widget is added.
3. To remove a widget, use one of the following methods:
   a. Select a widget that you want to remove. Right-click on the title bar of that widget and select *Delete* from the shortcut menu.

      The widget is removed.

   b. Alternatively, you can right-click on the title bar of the HTML page and select *Template configuration...* from the shortcut menu. The Template Configuration dialog box shows the row (or column) numeric configuration. For example, 3-2.

      Change the numeric configuration to reduce the number of widgets. For example, to remove a widget from the first row, change the value to 2-2.

      The widget is removed.

Procedure: How to Add an Additional Object to a Widget

You can put more than one object (such as a report or chart) in an output widget, as detailed in this procedure.

1. Click an object that has already been added to a widget.

   - If the widget contains a report or chart, you can add another report or chart, but not any other type of object.
   - If the widget contains a toolbox, you can add an additional report or chart, but it will not have the Autosize option set by default. You must set this option manually.

Open the Properties panel to verify that the IFRAME is selected.

2. On the *Components* tab, click the object that you want to add. For example, *Report* or *Chart*.

   The widget can be divided vertically (with the two objects, side by side) or horizontally (with the two objects one above the other). Move your cursor to the right or left side, or the top or bottom of the widget.

   The existing content automatically moves to the opposite side of your cursor.

3. Click on the side where you want to insert the new object.

   The widget is automatically divided into two equal columns or rows. The side you selected contains the report or chart icon in an iframe.

   Note: If you change your mind after you add the new iframe, you may right-click and delete it. However, the original iframe will not automatically expand to fill the original space in the widget. At run time, the content of the original iframe does fill the original space. But at design time it remains partial. You can click the original iframe and manually drag its border to fill the widget space, if you want the appearance of the page at design time to more closely reflect the appearance at run time.

4. Right-click the icon and select the desired option from the shortcut menu to create the new object. For example, *Reference existing procedure*.
5. Click the widget title bar and open the Properties panel. The Number of columns property defaults to Auto. Use the drop-down menu to select the number of columns you want in the widget. Select *2* if you have two objects side by side. Select *1* if you have two objects, one above the other.

   Run the page to preview widgets that have multiple objects to be sure the content remains legible.

Procedure: How to Apply Page-Level Controls in a Responsive Page

You may often have a responsive page that contains multiple reports that use the same controls. For example, if users can select region and month, they may want to show the same region and month selection for multiple reports in the page. In this case, you can apply common controls at the page level.

Your page can also include one or more reports that use controls that differ from the common ones. In this case, you can apply additional controls at the widget level, as defined in *How to Apply Widget-Level Controls in a Responsive Page* on page 258.

This procedure details how you apply controls at the page level.

1. Create multiple reports or charts that use the same controls and add them to widgets in a responsive layout, as detailed in *How to Select a Responsive Template* on page 253.

When you add a report with controls to a widget, the New Parameters dialog box opens. You can use this dialog to chain controls together. Chaining lets you filter control values based on the selected value from the prior control in the chain. For example, if State and City controls are chained, the City values will be filtered to show only cities in the selected state.

For more information see *Chaining in the HTML Canvas* on page 195.

2. Run the page. Note that:
   - ❑ The control names and values are appended to the page title in the title bar.
   - ❑ The controls input panel does not display. This is the default.
3. Click the down arrow in the right corner of the title bar to display the controls input panel.

   Close the HTML output and return to the HTML canvas.
4. If you want the controls input panel to display when the page opens:
   a. Click the title bar and open the Properties panel.

      The properties for windowPanel1 <DIV> are displayed.
   b. Change the Auto-hide inputs panel value to *No*.
   c. Run the page. Note that the controls input panel now does display.
5. To display the controls on the left side of the input panel:
   a. Click the controls input panel and open the Properties panel.
   b. Change the Control panel value to *Yes*.
   c. Run the page and expand it to full screen. Note that the controls in the input panel display together on the left side of the panel, where the user expects them to be.

   If you do not set the Control panel value to *Yes*, the controls will automatically expand to the right to use the full width as the page size increases.

The selections that the user makes in the controls input panel are automatically applied to all the reports on the page that share the controls.

Procedure: How to Apply Widget-Level Controls in a Responsive Page

You may often have a responsive page that contains multiple reports that use the same controls. For example, if users can select region and month, they may want to show the same region and month selection for multiple reports in the page. In this case you can apply common controls at the page level, as defined in *How to Apply Page-Level Controls in a Responsive Page* on page 257.

Your page can also include one or more reports that use controls that differ from the common ones. In this case, you can apply additional controls at the widget level.

This procedure details how you apply controls at the widget level when different controls are used at the page level.

1. Open a responsive HTML file that uses page-level controls, such as the one created in *How to Apply Page-Level Controls in a Responsive Page* on page 257.

   Add a new widget to a row as detailed in *How to Add or Remove a Widget* on page 256.

2. Create a report or chart that uses controls and add it to the new widget. For purposes of this procedure, use different controls than the ones used in the other reports in this page.

   When you add a report with controls to a widget, the New Parameters dialog box opens. By default, this dialog is set to create a control. Click *OK*.

3. Run the page. Note that the controls input panel displays in the new widget, by default.
4. To hide the controls input panel in a new widget:
   a. Add a new widget to a row as detailed in *How to Add or Remove a Widget* on page 256. Add the report you created in Step 2 to the new widget.

      The New Parameters dialog box opens.

   b. In the Parameter grouping options field select *New single layer form* and click *OK*.

      The New Parameters dialog box closes. The input panel is hidden.

The selections that the user makes in the controls input panel at the page level are automatically applied to all the widgets on the page that share controls. Additionally, the user can make a selection in the independent widget that does not share controls.

Procedure: How to Animate the Controls Input Panel

Creating a page with a responsive template enables you to add animation that appears when a user opens or closes a controls input panel at run time. This consists of creating and naming an animation, then applying it to the window panel that corresponds with the input controls, as detailed below.

1. Open an HTML page that was a) created with a responsive template; and b) contains input controls.
2. Open the *Tasks & Animations* panel. In the JQuery Animations section, click the *New* button.

   The system gives the new animation a default name. For example, *Animation1*. If you wish to rename the animation, click the default name and type the new name.

   Note: It is not necessary to specify a selected target for the animation. Leave this field blank.

3. In the Effect section, use the drop-down menu in the Type field to select an animation. If you wish, you may select an Option and Value to apply to the selected animation.
4. At the bottom of the panel, click *Toggle animation*.

   The animation is ready to be applied to the window panel.
5. In the responsive layout, click the title bar under which the control input panel appears.
6. Open the Properties panel. In the Output Widget section, click the *Select animation* field. Use the drop down menu to highlight the desired animation name. For example, *Animation1*.
7. Run the HTML page. Use the up/down arrow button on the title bar to see the animation when you hide/reveal the controls input panel.

Procedure: How to Set Minimum and Maximum Panel Sizes

Responsive panels automatically resize as the viewing window changes, but panels do not have a default minimum or maximum size. You can set minimum or maximum size, in pixels or percentages, to best suit the panel contents, as follows.

Note: Panels in a responsive page resize within the context of the minimum and maximum page width. To view or edit these values, open the Properties panel and select *DOCUMENT* from the drop-down menu at the top of the panel. For Responsive: min width, the default is 320px. For Responsive: max width, the default is Full screen.

1. Click a responsive panel that contains controls, or click the title bar of a widget, and open the *Properties* panel.

   The Properties panel for the panel <DIV> includes the following properties: Min-width, Max-width, Min-height, and Max-height. These values are blank, by default.
2. You can manually type a number of pixels or a percentage. For example, *50px* or *50%*.

   Alternatively, you can use the shortcut menu as follows:
3. Right-click a responsive panel that contains controls, or right-click the title bar of a widget. From the shortcut menu, select *Update Sizes*.

   The Update Sizes shortcut menu displays.

   Select *Update minimum size* to populate the Min-width and Min-height properties with the number of pixels or percentages used by the selected container. In other words, the container will not scale smaller than the size shown in the responsive layout.

   Note: Update minimum size does not change the object size in design time, but it does change the size in run time.

Select *Use Current* to use sizes from the selected container as follows:

- **Size as min size.** To populate the Min-width and Min-height properties with the number of pixels used by the selected container.
- **Size as max size.** To populate the Max-width and Max-height properties with the number of pixels used by the selected container.
- **Height as min height.** To populate the Min-height property with the number of pixels used by the selected container.
- **Width as min width.** To populate the Min-width property with the number of pixels used by the selected container.
- **Height as max height.** To populate the Max-height property with the number of pixels used by the selected container.
- **Width as max width.** To populate the Max-width property with the number of pixels used by the selected container.

Procedure: How to Load a Widget at Run Time

In some cases, the content you add to a widget might not load at run time, leaving the widget blank. This could occur if you did not select the *Run requests on load* option from the HTML/Document wizard, or if you added a new widget to a layout.

To cause widget content to load at run time:

1. Open the *Tasks & Animations* panel, select the *Load Task*, and then select *Load Trigger Type*.
2. In the Requests/Actions section, click the down arrow next to the Add Request button.

   The shortcut menu displays.
3. Select *Run Request* and then select an iframe, for example *iframe2*.

   The iframe is added to the Requests/Actions field and is highlighted.
4. In the Target Type field, select *Frame*. In the Target/Template field, select the iframe that corresponds to the one that is highlighted, for example *iframe2*.
5. Run the page.

   The content loads at run time.
6. Repeat these steps for all the iframes in your page that do not load at run time.

Procedure: How to Create a Custom Layout

In the HTML/Documents Wizard, the Templates, Settings, and Themes page includes six responsive templates that represent the most commonly used layouts. You can use these layouts as-is, or use them as a point of departure for a modified layout that contains more or fewer widgets. For more information, see *How to Add or Remove a Widget* on page 256.

Instead of using a template, you can create a custom layout, as detailed 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. 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.
3. In the Responsive Templates section, click *Custom*.
4. Click *Finish*.

   The layout opens and the Template configuration dialog is activated.
5. In the Orientation field, select *Row-based* or *Column-based*.
6. Use the Select row number and Number of containers in a row field as follows:
   a. With number 1 showing in the Select row number field, use the drop down list in the Number of widgets in a row field to specify the number of widgets you want in row 1.

      Note: Use the drop-down list. Do not type in the field.

   b. Select *2* in the drop-down list in the Select row number field.
   c. Use the drop-down list in the Number of widgets in a row field to specify the number of widgets you want in row 2.
   d. Repeat this sequence as needed to define your custom layout.

      Close the Template configuration dialog.

7. Define the page-level behavior of input controls as follows:

   - In the Properties panel for windowPanel1 <DIV>, set Auto-hide inputs panel to *Yes*. At run time, users reveal the panel by clicking the down arrow in the title bar. This controls page-level behavior.
   - In the Properties panel for windowPanel2 <DIV>, set Auto-hide inputs panel to *Yes*. At run time, users reveal the panel by clicking the down arrow in the title bar. This controls widget-level behavior. Repeat this for each widget for which you want to use widget-level controls.
   - Select *No* to display the controls input panel. At run time, users hide the panel by clicking the up arrow in the title bar.

8. To add a page header, right-click in the title bar of the page and select *Add page header* from the shortcut menu. You could use this area to add a graphic and/or company logo, for example.

Creating a Page With the HTML Canvas in 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. You can also use this method to convert an existing HTML file to responsive.

The following containers are responsive, by default:

- Report
- Chart
- Form (Single Layer and Multiple Layer)
- Tab (Top, Bottom, Left, and Right)
- Accordion (Vertical and Horizontal)
- Window
- Output Widget
- Group Box
- Panel
- Frame

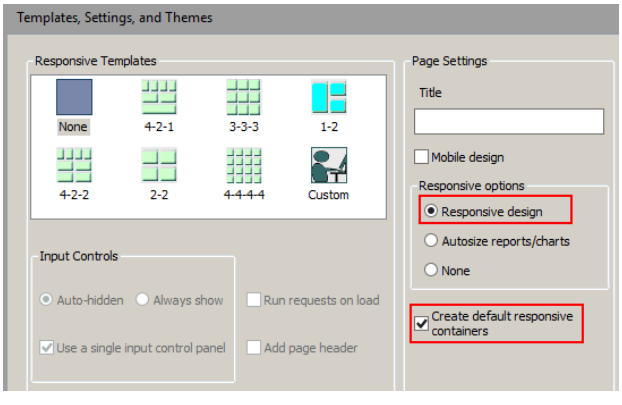
Procedure: How to Create New Content in Responsive Mode

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.
   - Controls are not responsive, by default. If you draw a control, right-click the control and select *Add Selected to a new container* to draw a responsive container around the 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.

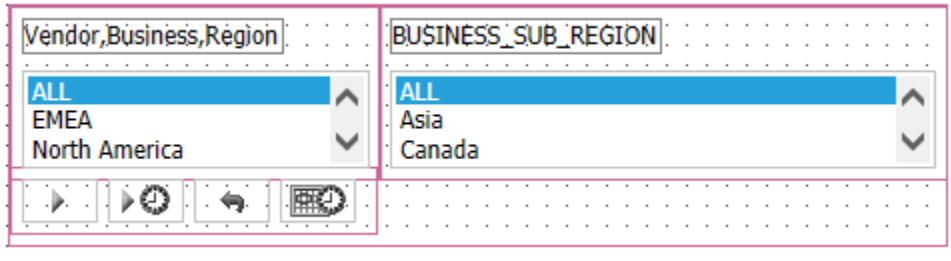
   The Document properties are displayed.

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 HTML object manipulation from the shortcut menu.
3. Select *Merge containers*.
4. To make a frame responsive, select it and open the Properties panel. Set the Enable responsive property value to *Yes*.

The HTML canvas is now in responsive mode. The frames and controls on your HTML canvas are each enclosed in a responsive container, indicated by a red box.

Note: if the HTML file had the Autosize Children option enabled, it will be superseded by the new responsive behavior, which automatically controls sizing.

The following properties appear in the Responsive design section of the Properties panel for the DOCUMENT object, whose values you may edit:

- **Enable responsive.** Enables responsive behavior in the HTML page. When set to Yes, additional responsive design properties become available.
- **Flow Direction.** Allows you to control how responsive containers rearrange themselves. You can leave this property as Not Set, or specify to fold into rows or columns.
- **Create responsive containers.** When set to Yes, objects added to the document automatically have responsive behavior enabled.
- **Global responsive margin.** Defines the margin, in pixels, around all four edges of all responsive containers on the page. Templates have no margins set, by default.
- **Responsive: min width.** Elements will not be repositioned or resized if the width of the container is less than this minimum, specified in pixels or percentage.
- **Responsive: max width.** Elements will not be repositioned or resized if the width of the container is greater than this maximum, specified in pixels or percentage.

If you select a responsive container, the following properties appear in the Responsive design section of the Properties panel:

- **Enable responsive.** Set to Yes, by default, indicating that the panel exhibits responsive behavior.

- **Flex grow.** Uses a numeric value to specify how quickly the item expands relative to other items on the HTML page when the size of its container changes.
- **Content alignment.** Controls how objects in a responsive container are aligned at run time. You can select one of the following options:
  - **Blank <Not Set>.** The objects are aligned toward the left side of the container by default.
  - **Start.** The objects are aligned on the left side of the container.
  - **End.** The objects are aligned on the right side of the container.
  - **Center.** The objects are centered within the container.
  - **Space between.** The objects are evenly distributed by width. The first object is close to the left side, the last object is close to the right side, and the objects in between are spaced equally.
  - **Space around.** The objects are evenly distributed by width and height.
- **Flow Direction.** Allows you to control how responsive containers rearrange themselves. You can leave this property as Not Set, or specify to fold into rows or columns.
- **Responsive width.** Determines whether the width of an object adjusts responsively at run time. Select *No* to keep the width set at design time in the canvas.
- **Responsive height.** Determines whether the height of an object adjusts responsively at run time. Select *No* to keep the height set at design time in the canvas.

If you select an iframe object, such as a report or chart component, in the canvas, the following properties appear in the Responsive design section of the Properties panel:

- **Enable responsive.** Determines whether the object exhibits responsive behavior.
- **Flex grow.** Uses a numeric value to specify how quickly the item expands relative to other items on the HTML page when the size of its container changes.
- **Responsive width.** Determines whether the width of an object adjusts responsively at run time. Select *No* to keep the width set at design time in the canvas.
- **Responsive height.** Determines whether the height of an object adjusts responsively at run time. Select *No* to keep the height set at design time in the canvas.

A black vertical line appears on the right side of the HTML canvas in responsive mode, as a visual indicator of the responsive maximum width. You may need to scroll right to see this line.

5. To move contents from one container to another, click the *Positioning* tab and, in the Responsive Design group, click *Toggle Selection*. Drag contents between containers as needed and click *Preview Runtime*, on the Utilities tab.
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.

Procedure: How to Restrict the First Row to Controls Only

When you create a page with a responsive template, the first row of the page is automatically restricted to input controls (when applicable). Content such as reports or charts will not fold up to populate the first row, regardless of how wide the browser may stretch at run time.

When you create a page using the HTML canvas in responsive mode, this restriction is not automatic. Therefore a report or chart could fold up to populate the first row, if the browser width is increased. This behavior may be considered undesirable.

Ensure that the first row contains controls only, as follows:

1. Open an HTML page that was:
   - ❑ Created with the HTML canvas in Responsive mode

     and
   - ❑ Contains input controls.
2. Verify that the input controls are all grouped within a single container. If not:
   a. Hold down the Ctrl key and multi-select all the controls.
   b. Right-click on one of the controls and select HTML object manipulation.
   c. From the shortcut menu, select *Add selected to new container*.

      Note that a box appears that contains all the selected controls. The box is red, indicating it is responsive.
3. To display the controls on the left side of the input panel:
   a. Click the controls input panel and open the Properties panel.
   b. Change the Control panel value to *Yes*.
   c. Run the page and expand it to full screen. Note that the first row contains controls only. Also note that the controls in the input panel display together on the left side of the panel, where the user expects them to be.

   If you do not set the Control panel value to *Yes*, the controls will automatically expand to the right to use the full width as the page size increases.

Procedure: How to Create a Responsive Page for Use in a BI Portal

The Db2 Web Query BI Portal gives users access to BI content, including dashboards, visualizations, reports, charts, graphs, maps, and interactive InfoApps. The BI Portal allows users to analyze vital information quickly, link content and reports together, and easily tailor information to their needs.

This procedure is comprised of two parts: creating a report for use in the responsive portal and creating a responsive HTML page using the HTML canvas.

1. Select or create a report with the following criteria:
   - ❏ Has dynamic parameters specified in a WHERE clause.
   - ❏ Has _FOC_NULL as the default.
   - ❏ Has a parameter that is a Multiselect OR.
   - ❏ Has a value of *No* in the File/Folder Properties panel - Prompt for Parameters property.
2. Create a responsive HTML page as follows:
   a. Open the *HTML/Document Wizard* from the *Home* tab, in the *Content* group, or from the shortcut menu in the Environments Tree panel.
   b. Click *Next*.

      The Templates, Settings, and Themes page opens.
   c. In the Page Settings area, click *Responsive design* and click *Finish*.

      The HTML canvas opens. In the Properties panel, in the DOCUMENT properties, verify that the Responsive design object property is set to *Yes* and the Load in iFrame object property is set to *No*.

      The Load in iFrame object property is used to enable objects in a BI Portal to overlap the edges of a container. For example, when you have a pop-up calendar that populates a date field, the calendar can overlap the control container to allow the user to select a date.

      Note: The Load in iFrame object property creates a <DIV> for the page instead of an <iFrame> tag. A BI Portal is limited to only one HTML Canvas page that is enclosed in a <DIV> tag.

      Note that the Responsive: min width is set to 320px and the Responsive: max width is set to Full screen. These default values can be edited in pixels or percentages.
   d. In the Requests and Data sources panel, click the drop-down arrow next to the New icon, and select the option Requests-Parameters-Filter Panel.

      The Open File dialog box opens.

e. Select the procedure you designated in Step 1 and click *OK*.

   The New Parameters dialog box opens.

f. Click *Autochain controls in above specified order*, and click *OK*.

   The New Parameters dialog box closes and the HTML canvas is opened, showing the controls in containers. The red boxes indicate that responsive design is activated.

g. Open the Tasks & Animations panel and verify that the Submit button (task 2) Requests/Actions value is Refresh1 and the Target Type is Refresh BI Portal.

h. Right-click the canvas and select *Preview Runtime* or click the *Preview Runtime* button on the Utilities tab. Note that the content expands to the maximum width. The black vertical line on the right side of the canvas is a visual indicator of the specified Full screen max width.

i. Save the HTML page.

Reference: Tips for Using Responsive Design

The following are some best practices in working with Responsive Design to create responsive HTML pages.

- When creating a new responsive HTML page:
  - Use a Responsive template to generate a layout with widgets. The HTML/Document Wizard lets you control the number of rows and the number of widgets per row, lets you determine how input controls will appear, lets you add a page header, and lets you run requests automatically when the page loads at run time.
  - Use Responsive Design in the HTML canvas to generate a layout without widgets. This method lets you design a page for a BI Portal.
- On the Positioning tab, the Positioning group contains the Same Width, Same Height, and Same Size commands that behave as follows in responsive design mode:
  - **Same Width.** Sets the minimum and maximum width to that of the control object (the previously selected object).
  - **Same Height.** Sets the minimum and maximum height to that of the control object (the previously selected object).
  - **Same Size.** Sets the minimum and maximum width and height to that of the control object (the previously selected object).

To use these commands:

1. Click the object whose size you want to duplicate.
2. Use the Control key and click the secondary objects that you want to resize.
3. Click the desired command button from the Positioning group.

- On the Positioning tab, the Responsive Design group contains the Toggle Selection and Update Layout commands. These commands are enabled in responsive design mode.
  - **Toggle Selection.** Toggle on to allow responsive containers that include content to be dragged to a new position. Toggle off to prevent those containers from being repositioned.
  - **Update Layout.** This command is activated in a responsive page when the Autosize children property is set to Yes. Click to refresh the HTML canvas after you reposition containers, to show how the containers will stack when folded.
- You can control the run-time alignment of contents within a responsive container using the *Content alignment* property. This property appears for a parent responsive container that contains multiple responsive objects.

  Click the down arrow to display the following property values:

  - **Blank <Not Set>.** The objects are aligned toward the left side of the container, by default.
  - **Start.** The objects are aligned on the left side of the container.
  - **End.** The objects are aligned on the right side of the container.
  - **Center.** The objects are centered within the container.
  - **Space between.** The objects are evenly distributed by width. The first object is close to the left side, the last object is close to the right side, and the objects in between are spaced equally.
  - **Space around.** The objects are evenly distributed by width and height.

  The alignment in the HTML canvas does not change after you specify a content alignment property value. Click *Run* on the Quick Access Toolbar to see the specified alignment in the browser.
- A black vertical line appears on the right side of the HTML canvas in responsive mode, as a visual indicator of the breakpoint width for desktop display. In order to control the behavior of your responsive containers, position them to the left of this line on the HTML canvas. Content appearing to the right of this line will fold arbitrarily.

- If you have both Mobile Design and Responsive Design enabled, Mobile Design takes precedence when you view content on the smartphone. Mobile Design uses jQuery mobile controls. Responsive Design uses browser-generated controls.

Designing Content for Smartphones

You can design content specifically to be used on smartphones. Content designed in this way is automatically sized, and takes full advantage of the page-swipe navigation and touch-screen capabilities of smartphones.

Note: Tablets can display native Developer Workbench HTML content without modification. This section pertains to designing HTML content for smartphones only.

The following steps outline how you can design HTML content for smartphones:

1. Create and save your content in an HTML file that is built in the native Developer Workbench HTML canvas.
2. Enable Mobile Layout for the HTML file by using the Enable mobile property.

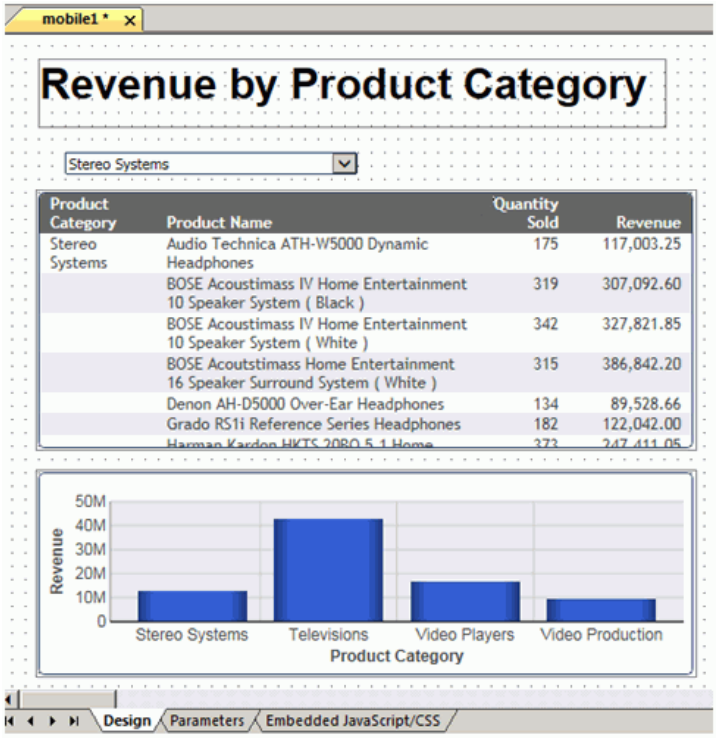
   For more information, see *Enabling Mobile Layout in an HTML File* on page 277.
3. Select either the Basic or Advanced Mobile Layout.

   For more information, see *Working With the Basic and Advanced Mobile Layouts* on page 278.
4. Populate the smartphone pages with your HTML content.
5. View the results on your smartphone.

   For more information, see *Viewing Mobile Output on a Smartphone* on page 284.

Designing Output for Smartphones With the Mobile Layout Functionality

Before you use the Mobile Layout functionality, you must create and save your content on the native HTML canvas. The example shown in the following image contains four objects: a text box containing a title, a drop-down control, a frame containing a report, and a frame containing a chart.



Frames

Frames in your HTML content automatically appear in the Mobile Layout. In the Basic Mobile Layout, individual frames appear separately on individual pages. In the Advanced Mobile Layout, all frames appear together on a single page. You can change either layout to display frames on pages as you like.

The page order, when viewed on a smartphone, is determined by the frame order in the Tasks area of the Tasks & Animations panel. You can change the page order in the Mobile Layout, as needed.

Animations that you have targeted to a frame included in the Mobile Layout will run on your smartphone.

Controls

To maximize the capabilities of a touch screen, single-select and multi-select controls both become listbox jQuery mobile controls, with one allowing single-select and the other multiselect.

Note: jQuery mobile controls do not display when you design your HTML page using the Mobile Layout, but will display when the page is viewed on a smartphone.

Other Objects

Some objects that you add to the HTML canvas are not automatically promoted to pages in the Mobile Layout. This is true for text boxes and images.

Objects that are not automatically promoted to pages appear below a horizontal line that underlines the page preview. You can drag objects above the line onto pages, as needed. Objects that remain below the line, and any animations targeted to them, will not appear when you view the pages on your smartphone.

Note: The size of an object that you drag into a page is accurately previewed in the page, so you can determine if the object needs to be resized. If so, you must resize it in the HTML canvas.

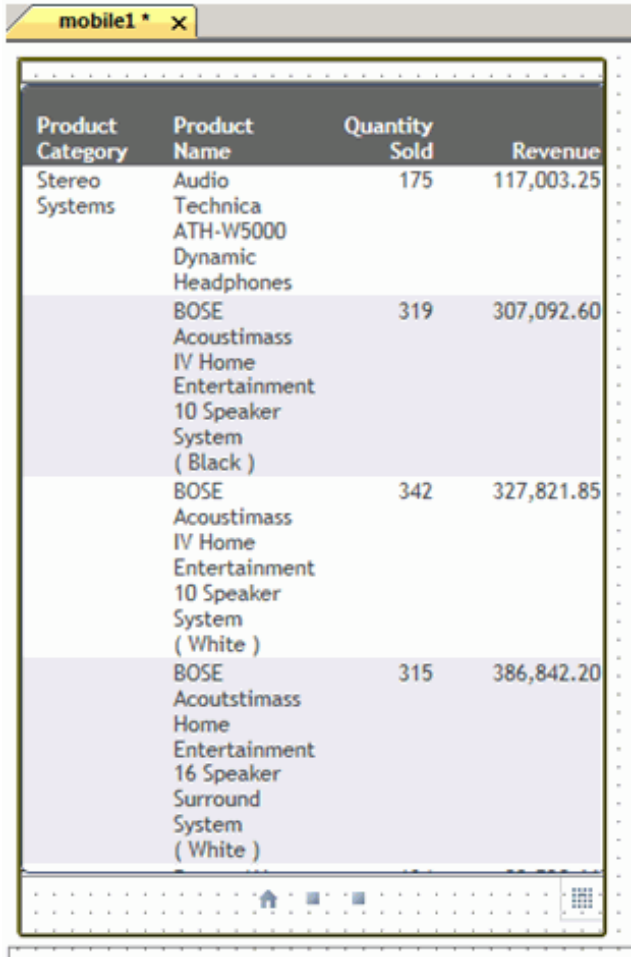
When you drag an object from below the horizontal line onto a page, the system recognizes this as a manual modification to the Mobile Layout. This triggers a message: Manual modification of mobile layout will stop default generation of the pages. It will be your responsibility to maintain it.

Accepting this means that changes to the underlying data will not flow automatically to the mobile pages. You must open the HTML file, right-click the canvas, and select *Update Layout*.

To take advantage of the default generation of the mobile pages, it is recommended that you avoid making manual modifications.

Note: You can undo manual modification of the Mobile Layout by right-clicking the canvas and clicking *Reset mobile to default*. This restores automatic update of the HTML pages, and removes any manual modifications.

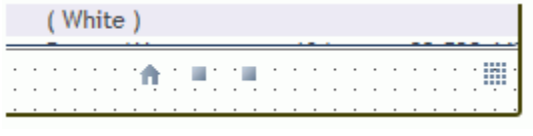
A page in Mobile Layout appears as shown in the following image. In this example, the page content displays in a frame, which is scrollable when you view it on a smartphone. Below the frame is the navigation area.



Styling for the components of the Mobile Layout is controlled by the cascading style sheet. For more information, see *Cascading Style Sheet Class Mapping List for Mobile Components* on page 284.

The Navigation Area

The following image displays the navigation area that appears below the page content.



The icons in our example indicate that there is a home page, which is represented by the house icon, and two additional pages, which are represented by two square icons. In addition, there is a page combination tool, which is represented by a tile icon.

The default home page appears first in the page order. You can change the home page as needed.

You can use the square icons and the house icon to navigate between the pages.

You can use the tile icon to vertically combine objects in a page, in cases where there is more than one object on a page. Toggle the tile icon to change vertical combinations.

Adding/Removing Pages

The following list describes the shortcut menu options on the Mobile Layout canvas, which you can use to add and remove pages.

- **Show all pages.** Displays all pages on the Mobile Layout canvas. Both the Basic and Advanced Mobile Layouts show one page, by default.
- **Add empty page.** Adds an empty page to the Mobile Layout canvas. You can then drag content into the page.
- **Remove empty page.** Deletes any empty pages on the Mobile Layout canvas.
- **Remove selected page.** Deletes a page that has content. The system recognizes this as a manual modification to the Mobile Layout. This triggers a message: Manual modification of mobile layout will stop default generation of the pages. It will be your responsibility to maintain it.

  Accepting this means that changes to the underlying data will not flow automatically to the mobile pages. You must open the HTML file, right-click the HTML canvas, and select *Update Layout*.
- **Reset mobile to default.** Restores automatic update of the HTML pages, and removes any manual modifications.

Note: When you delete content from the HTML canvas, you must also manually delete it from the Mobile Layout. Deleting content from the Mobile Layout that was previously deleted from the HTML canvas is not interpreted as a manual modification, and will preserve the default generation of the pages.

Moving Content Between Pages

You can change the order of the pages or display more than one object on a page.

To display more than one object on a page horizontally, click the object in one screen and drag it to another.

To display more than one object on a page vertically, use the tile icon  . Toggle the tile icon to remove vertical combinations.

To move a control or an object that appears below the horizontal line, drag the object to the desired page. The system recognizes this as a manual modification to the Mobile Layout. This triggers a message: Manual modification of mobile layout will stop default generation of the pages. It will be your responsibility to maintain it.

Accepting this means that changes to the underlying data will not flow automatically to the mobile pages. You must open the HTML file, right-click the HTML canvas, and select *Update Layout*.

Note: You can undo manual modification of the Mobile Layout by right-clicking the canvas and clicking *Reset mobile to default*. This restores automatic update of the HTML pages, and removes any manual modifications.

To take advantage of the default generation of the pages, it is recommended that you avoid making manual modifications.

Enabling Mobile Layout in an HTML File

An HTML file is designated for mobile output when the Enable mobile property is set to *Yes*. This designation enables a Mobile Layout option, where you can edit and preview your mobile output.

Note: An HTML file that has Mobile Layout enabled can additionally be viewed on a desktop or tablet. In these cases, page design is derived from the HTML canvas and the page design on the Mobile Layout is ignored. This eliminates the need to maintain different copies of the same pages for different output.

You can enable mobile output for an existing HTML file, or create a new file with mobile output enabled.

Procedure: How to Enable Mobile Output for an Existing HTML File

1. Open the HTML file.

   The HTML canvas opens.

2. Open the Properties panel.
3. Change the *Mobile* property from *No* to *Yes*.

The HTML canvas shortcut menu now includes mobile output options.

Procedure: How to Create a New File With Mobile Output Enabled

1. On the Home tab, in the Content group, click *HTML/Document*. You can also create a new HTML file from the Application menu or by using the shortcut menu in the Environments tree panel, for a folder that supports content creation.

   The HTML/Document Wizard opens.

2. Navigate to where you want to create your HTML page and click *Next*.

   The Themes and Settings window of the HTML/Document Wizard opens.

3. Select the *Mobile design* check box.

   Selecting this option automatically sets the *Mobile* property to *Yes* in the Properties panel.

4. Click *Finish*.

   The HTML canvas opens. The canvas shortcut menu now includes mobile output options.

Working With the Basic and Advanced Mobile Layouts

There are two different layouts for designing HTML output for smartphones, Basic and Advanced.

- **Basic Mobile Layout.** Enables you to build simple smartphone pages that you can navigate using graphical icons and page-swipe. The Basic Mobile Layout displays individual frames from your HTML canvas on individual mobile pages, by default. You can change this as needed.
- **Advanced Mobile Layout.** Offers the same features as the Basic, but also gives you the additional ability to add a table of contents with or without a logo, and add a logo on each page.

In the Advanced Mobile Layout, all frames from your HTML canvas appear together on a single mobile page, by default. You can change this as needed.

Having selected one layout, you may choose to switch to the other, if needed. For more information, see *How to Change Between Basic and Advanced Mobile Layouts* on page 280.

Procedure: How to Use the Basic Mobile Layout

You can work with the Basic Mobile Layout when you have saved content in the HTML canvas and when you have set the Enable mobile property to *Yes*.

1. Right-click the HTML canvas and click *Show Mobile Layout (basic)*.

   The Basic Mobile Layout is displayed.

Styling for the components of the Mobile Layout is controlled by the cascading style sheet. For more information, see *Cascading Style Sheet Class Mapping List for Mobile Components* on page 284.

2. Edit the mobile content in the following ways:

   - To preview all pages, right-click the Mobile Layout canvas and click *Show all pages*. This displays all pages side-by-side.
   - A horizontal line appears below the navigation area. Any objects that appear below the line come from the HTML canvas, but were not automatically added to the Mobile Layout.

     To add objects to the Mobile Layout, drag them from the area below the line to the desired place in the layout.

     The system recognizes this as a manual modification to the Mobile Layout. This triggers a message: Manual modification of mobile layout will stop default generation of the pages. It will be your responsibility to maintain it.

     Accepting this means that changes to the underlying data will not flow automatically to the mobile pages. You must open the HTML file, right-click the HTML canvas, and select *Update Layout*.

     Note: You can undo manual modification of the Mobile Layout by right-clicking the canvas and clicking *Reset mobile to default*. This restores automatic update of the HTML pages, and removes any manual modifications.

     To take advantage of the default generation of the pages, it is recommended that you avoid making manual modifications.
   - To combine pages horizontally, drag one page to another.
   - To change the home page, select the desired page, right-click the Mobile Layout canvas, and click *Set current page as home page*.

     Additionally, you can drag a page to the home page screen. If necessary, drag the old page to a different screen.

     Note: When you change the page order by manually dragging in the Mobile Layout, it overrides the page order specified in the Tasks & Animations panel.
   - If there are any empty pages in your preview, right-click the canvas and click *Remove empty pages*.

3. Save the HTML file.

You can now view the HTML file on a smartphone. The layout is derived from the Mobile Layout. For more information, see *Viewing Mobile Output on a Smartphone* on page 284.

Additionally, you can view the HTML file on a desktop or tablet. The layout is derived from the HTML canvas.

Procedure: How to Change Between Basic and Advanced Mobile Layouts

To change between Basic and Advanced Mobile Layouts:

1. Right-click the existing *Mobile Layout* canvas and click *Reset mobile to default*.

   This restores the Mobile Layout to its original state.

2. Right-click the existing *Mobile Layout* canvas and click *Back to desktop*.

   The HTML canvas opens.

3. Right-click the HTML canvas and click the new layout, *Show Mobile Layout (basic)* or *Show Mobile Layout (advanced)*.

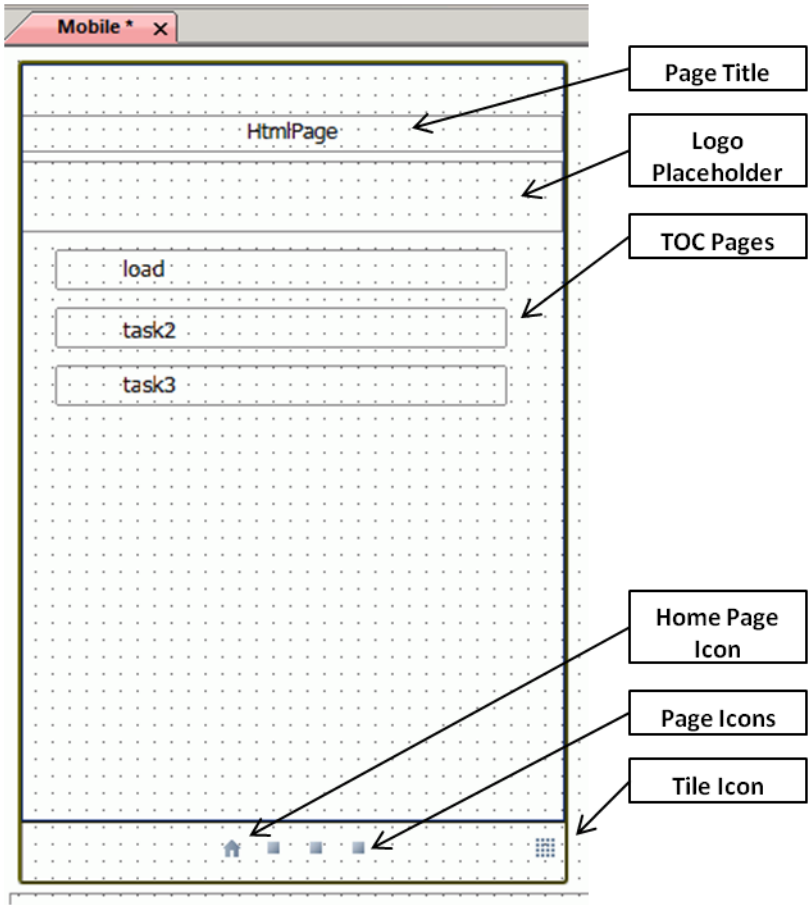
Procedure: How to Create a Table of Contents for Mobile Output

With the Advanced Mobile Layout, you can include a table of contents page to help users navigate quickly. This may be useful when your mobile output contains a large number of pages.

To design output for smartphones that includes a table of contents:

1. Create an HTML page.
2. Right-click the *HTML* canvas and click *Show mobile layout (advanced)*.

   The Mobile Layout opens and displays the table of contents preview, as shown in the following image.



The Mobile Layout default display shows a single page, which previews the appearance of the table of contents in a smartphone format.

In our example, this rectangle contains the title *HtmlPage*.

Styling for the components of the Mobile Layout is controlled by the cascading style sheet. For more information, see *Cascading Style Sheet Class Mapping List for Mobile Components* on page 284.

3. Edit your layout in the following ways:

   - To change the title, use the drop-down at the top of the Properties panel and select *table_of_contents*. In the *Title* field, type the new title.

     The new title now appears in the table of contents preview.

- You can choose to show or to hide the table of contents on a smartphone. The default is to show it.

  To hide the table of contents on a smartphone, select *No* from the *Show at run time* property field.

  In the preview, the rectangles for each page represent hyperlinks in the table of contents, which open the selected page. In our example, there are hyperlinks for the *task2* and *load* page titles.

- To change page titles, select the *Title* property field for a specified page and type the desired title.

  The new titles appear at the top of the page and on the table of contents.

4. Save the HTML file.

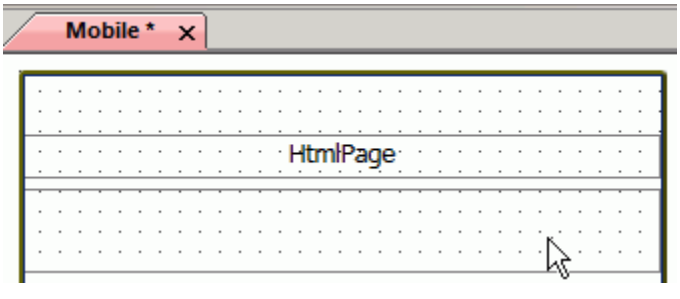
   You can now view the HTML file with a table of contents on a smartphone. For more information, see *Viewing Mobile Output on a Smartphone* on page 284.

Procedure: How to Add a Logo to a Mobile Table of Contents

You can add a logo to the table of contents page of your mobile layout.

1. Open the Advanced Mobile Layout of an HTML file that has a table of contents.

   The logo placeholder is indicated by the cursor in the following image.



2. In the drop-down at the top of the Properties panel, click *table_of_contents*.

   The table of contents properties display.

3. In the *Logo* field, click the ellipsis button.

   The Open File dialog box appears.

4. Select the desired logo file and click *OK*.

   The logo now appears in the table of contents preview.

Note: The logo placeholder box provides an accurate preview of the way the logo will appear on a smartphone. It will indicate if the graphic file you selected needs to be resized.

5. Save the HTML file.

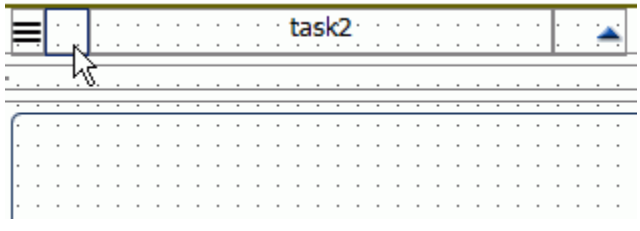
   You can now view the HTML file with a table of contents on a smartphone. For more information, see *Viewing Mobile Output on a Smartphone* on page 284.

Procedure: How to Add a Logo to Each Page of the Contents

You can add a logo to each page of the table of the contents in your mobile layout.

1. Open the Advanced Mobile Layout of an HTML file that has a table of contents.
2. Open a page, other than the table of contents page.

   The small square near the top left of the page, indicated by the cursor in the following image, is a placeholder for a logo.



3. In the drop-down at the top of the Properties panel, click *table_of_contents*.

   The table of contents properties fields display.

4. In the All swipes logo field, click the ellipsis button.

   The Open File dialog box appears.

5. Select the desired logo file and click *OK*.

   The logo now appears on each page of the preview.

   Note: The logo placeholder for the individual pages is a small box. It provides an accurate preview of the way the logo will appear on a smartphone, and indicates whether the graphic file that you selected needs to be resized.

6. Save the HTML file.

   You can now view the HTML file with a table of contents on a smartphone. For more information, see *Viewing Mobile Output on a Smartphone* on page 284.

Viewing Mobile Output on a Smartphone

There are multiple ways that mobile output from Developer Workbench can be viewed on a smartphone, depending upon the configuration of your environment. These include:

- The Mobile Faves App

These ways are outlined in the following section.

Note: Check with your system administrator for your best practice in viewing mobile output.

Mobile Faves App

The Mobile Faves app is automatically set up to obtain content from a server on the Internet, where you can view sample business content. You can add your own Db2 Web Query servers to the list, including those that contain the HTML files that are enabled for mobile output. You will see a list of available folders saved to the selected server connection. These folders contain all of the content previously selected as Mobile Favorites from your Db2 Web Query portal. To make an item appear in the Mobile Faves app, apply the *FAVES* tag to it.

iPhone or Android users can download the free Mobile Faves app from the App Store or the Google Play Store. manual.

Cascading Style Sheet Class Mapping List for Mobile Components

This section contains a list of IBI classes that are used to style various components and controls for Mobile design.

Components/Controls	Classes Used
Smartphone container border color	`.IBI_SmartPhoneContainer { border-color:#4C5E76; }`
Control panel border color	`.IBI_MobileControlPanel { border-top-color: #4C5E76; }`
Control panel font	`.IBI_MobileControlPanel { font-size:small; color: green; }`

Components/Controls	Classes Used
Page content font	`.IBI_MobilepageContent { font-size: large; color: Red; }`
TOC content font	`.IBI_MobileTocContent { font-size: large; color: Blue; }`
Label font	`.IBI_MobileLabel { font-size: large; color: Azure; }`
HTML title	`.IBI_MobileTocTitle { font-size: x-large; color: BlueViolet; }`
Task name in TOC	`.IBI_MobileTocItem { font-size: x-large; color: ForestGreen; }`
Control panel text	`.IBI_MobileSwipeHeaderText { font-size: small; color: Crimson; }`

Cascading Style Sheet Class Mapping List

This section contains a list of IBI classes that are used to style various components and controls.

Components/Controls	Classes Used
Body	IBI_PageBg
Report	IBI_Report-iFrame; IBI_rounded_m

Components/Controls	Classes Used
Chart	IBI_Report-iFrame; IBI_rounded_m
Hyperlink	IBI_LinkItem
Button	IBI_button
Reset	IBI_button
Label	IBI_ReportControlLabel
Form	IBI_ReportControlPanel; IBI_rounded_m
Submit Button	IBI_button; IBI_btn-run
Reset Button	IBI_button; IBI_btn-reset
Save Selection Button	IBI_btn-saveselection
Defer Button	IBI_btn-defer
Schedule Button	IBI_btn-schedule
Panel	IBI_Panel; IBI_rounded_m
Frame	IBI_rounded_m
Edit box	IBI_ReportControlTarget; IBI_rounded_s
Drop down	IBI_ReportControlTarget; IBI_rounded_s
List Box	IBI_ReportControlTarget; IBI_rounded_s
Double List	IBI_ReportControlTarget, IBI_btn-up, IBI_btn-down, IBI_btn-left, IBI_btn-right
Radio Button	IBI_ReportControlTarget; IBI_Radio
Check Box	IBI_ReportControlTarget; IBI_CheckBox
Text Area	IBI_rounded_m
Single Source Tree	IBI_ReportControlTarget
Multi Source Tree	IBI_ReportControlTarget

Components/Controls	Classes Used
Horizontal Slider	IBI_ReportControlTarget; IBI_rounded_s; IBI_SliderInput; IBI_buttonSliderNav; IBI_btn-left; IBI_btn-right; IBI_SliderTrack; IBI_SliderHandle
Vertical Slider	IBI_ReportControlTarget; IBI_rounded_s; IBI_SliderInput; IBI_buttonSliderNav; IBI_buttonSliderNavVert; IBI_btn-up; IBI_btn-down; IBI_SliderTrack; IBI_SliderTrackVert; IBI_SliderHandle; IBI_SliderHandleVert
Info Window	IBI_InfoWnd; IBI_InfoWndTitleBar; IBI_InfoWndFrame
Tab	IBI_MultiContentPlugin
Tab Header	IBI_pageHeader
TabHeader - Top Tab	IBI_pageHeaderTab_TopBottom
TabHeader - Bottom Tab	IBI_pageHeaderTab_Bottom
TabHeader - Left Tab	IBI_pageHeaderTab_Left
TabHeader - Right Tab	IBI_pageHeaderTab_Right
Tab Selected	IBI_pageHeader_Selected
Vertical Tab/Accordion/Window header	IBI_pageHeader_Vertical
Vertical Tab/Accordion/Window Selected	IBI_pageHeaderVertical_Selected
Tab Heading Text	IBI_pageHeaderText
Tab Heading Text - Left Tab	IBI_textVerticalTopToBottom
Tab Heading Text - Right Tab	IBI_textVerticalBottomToTop
Tab/Window/Accordion Content	IBI_pageContent
Tab/Window/Accordion Content - Top	IBI_pageContentTab_Top

Components/Controls	Classes Used
Tab/Window/Accordion Content - Bottom	IBI_pageContentTab_Bottom
Tab/Window/Accordion Content - Left	IBI_pageContentTab_Left
Tab/Window/Accordion Content - Right	IBI_pageContentTab_Right
Tab/Accordion/Window Min/Max box	IBI_windowNavBox
Tab/Accordion/Window images	IBI_windowMaxRestore; IBI_windowMaximize; IBI_windowMinimize; IBI_windowTiles; IBI_windowFlipRight; IBI_windowFlipLeft
Mobile components	
Smartphone container border color	IBI_SmartPhoneContainer
Control panel	IBI_MobileControlPanel
Page content	IBI_MobilepageContent
TOC content	IBI_MobileTocContent
Label	IBI_MobileLabel
TOC title	IBI_MobileTocTitle
TOC task name	IBI_MobileTocItem
Control panel text	IBI_MobileSwipeHeaderText

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.

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.