DataAssist Help Documentation

Contents

- System Requirements
- DataAssist Wizard
- DataAssist Search Wizard
- DataAssist Search server behavior
- Managing Single Records
- Managaing Multiple Records
- Repeat Selection
- DataAssist Repeating Table
- Sort

System Requirements

WINDOWS

- Windows Vista, Windows XP
- Dreamweaver CS3, Dreamweaver 8

MACINTOSH

- 500 MHz Power PC G3 processor
- Mac PowerPC or Intel - OS 10.4.x
- Dreamweaver CS3, Dreamweaver 8

SERVER LANGUAGES

- PHP 4+
- ASP JavaScript
- ASP VBScript
- Coldfusion MX 7

DATABASES

- MySQL 5.0 (for PHP)
- Any ODBC Compliant Database (for ASP and Coldfusion)

BROWSERS

- Mozilla Firefox 2.0 (Windows)
- Internet Explorer 6.0+ (Windows)
- Safari 2 (Mac)
- Mozilla Firefox 2.0 (Mac)

DataAssist Wizard

The DataAssist Wizard creates a data management application by creating pages to manage records (insert, update, and delete), as well as search records, return search results, and display detailed record information from a specified table within a data source.

The wizard assists you in configuring the datasource used to generate the designated page sets and corresponding functionality, as well as any dynamic menus. Menus used to populate options in form fields within the search, update, and insert pages can be populated dynamically from additional tables specified in the datasource configuration. This allows for database control of the content used to populate your records. You can even run the wizard a second time to create pages to administrate the records in these lists.

A variety of design options for results and detail pages are available, as well as the ability to preview how this content appears during the configuration process.

Data from available database columns is bound to layout components within your selected detail and results page types, making the retrieval and layout of record information a simple, automated process.

Use the *Back* and *Next* buttons to navigate through each of the configuration steps of the wizard. The number of steps in the wizard is determined by the types of pages selected for creation. At a minimum, the wizard requires the creation of a results page. If you are creating a catalog specifically for your customers, you most likely will not need to create insert, update, and delete pages for your application. However, when creating a backend administration tool to maintain your catalog, it is likely you will require all these page types.

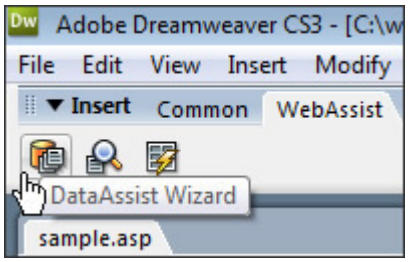
Step 1 through 4 of the wizard, detailing general and results page configuration, are required. Search page, Detail page, Update page, Insert page, and Delete page configuration steps are added dependent on what you have selected in Step 1 of the wizard. All steps are detailed here in the order they are performed, as though you elected to create all page options available.

Upon completing each configuration step in the wizard, click the *Finish* button to generate and open your pages. You can continue to customize the pages using functionality available within Dreamweaver.

Access

The following locations in Dreamweaver open the DataAssist Wizard interface:

- WebAssist Insert panel



- *Insert* > *WebAssist* > *DataAssist* > *DataAssist Wizard*

Configuration details

The following pages in this section detail the configuration options available in the wizard:

- Specify database options and page choices

  Configure your database connection, specify the page types you need created, and, if desired, select an available template from your site as the basis for your pages.

- Select layout options

  Configure the look and feel of your content (design attributes, color schemes, fonts), as well as the design for your result page record navigation controls.

- Specify results page layout options

  Configure the type of page (public or administrative), its title and header layout, and how many columns and rows are used to display the records returned.

- Results page options

  Configure the layout of your search results page based on the page type selected in step 3. For administrative page types, determine the database columns displayed for each record, and assign labels specific to each. For public page types, bind the database columns available to the corresponding layout component that displays that information for the records returned. In either page type, determine what information should link to the detail page for the given record. Configure any additional filters and sorting options to be applied to the recordset as the default query returned when no search parameters are passed to the page.

- Search page options

  Configure the search form, the search fields it contains, and the corresponding database columns that each field is compared against. Specify the type of comparison to be made for each field against its corresponding database column.

- Specify detail page layout options

  Specify the type of detail page used to display your records, as well as the type of title and header sections displayed above the record details.

- Details page options

  Configure the layout of your record detail page based on the page type selected in step 3. For administrative page types, determine the database columns displayed for the given record, and assign labels specific to each. For public page types, bind the database columns available to the corresponding layout component that displays that information for the given record.

- Update page options

  Configure the form, form fields, and corresponding database columns updated for a record passed to this page.

- Insert page options

  Configure the form, form fields, and corresponding database columns inserted to for a record created using this page.

- Delete page options

  Configure the database columns and corresponding labels for display of a given record passed to this page to be deleted.

DataAssist features applied to generated pages

Upon completing the wizard, the pages, supporting files and folders are added to the Dreamweaver site definition currently selected.

Several Dreamweaver server behaviors are leveraged through the wizard to assist in the functions of your data application. Most notably, a recordset is generated, based on the table selected for your datasource, within each page that requires it. For the results page created, the records are filtered and sorted based on the options specified in Step 3 of the wizard. Insert, Update, and Search pages that use dynamic lists configured in the wizard also contain recordsets specific to the tables that contain those list options.

In addition, the following WebAssist server behaviors are applied by the wizard to the pages created, with links to help specific to using each.

Individual server behaviors specific to data insert, update, and delete functionality for single records are added to the Insert, Update, and Delete pages created through the wizard, respectively:

- Insert Record server behavior
- Update Record server behavior
- Delete Record server behavior (This can also be applied to Detail pages through the wizard.)

Search functionality, although initiated by a search form, is always applied to the results page. The DataAssist Search server behavior controls the comparisons made by your search form against specified database columns to return filtered results. This server behavior can be applied individually outside of the wizard by applying the server behavior directly (recommended for advanced users and search configurations), or by using the DataAssist Search wizard.

In addition, records displayed on results pages can be displayed in multiple columns as well as rows. This is accomplished by displaying a specified number of records on a page for a given recordset using Dreamweaver's Repeat Region server behavior. DataAssist's Repeat Selection server behavior controls the layout of these records by controlling how many records are displayed in each row. Creating a results grid of this type can also be accomplished through an additional DataAssist feature: DataAssist Repeating Table, that applies each of these server behaviors as well.

If more records are returned to the results page than are configured for display, Dreamweaver's Recordset Navigation Bar is applied to the page to control paging through sets of records on the page, allowing the end user to navigate through the records returned.

Note for Coldfusion users: Please note that the Coldfusion pages created by the wizard require session management variables to be set for your website. If you are not configured appropriately for session management, your pages may return a ColdFusion runtime error: "*Cannot lock session scope*".

ColdFusion sites that use DataAssist have the following two requirements:

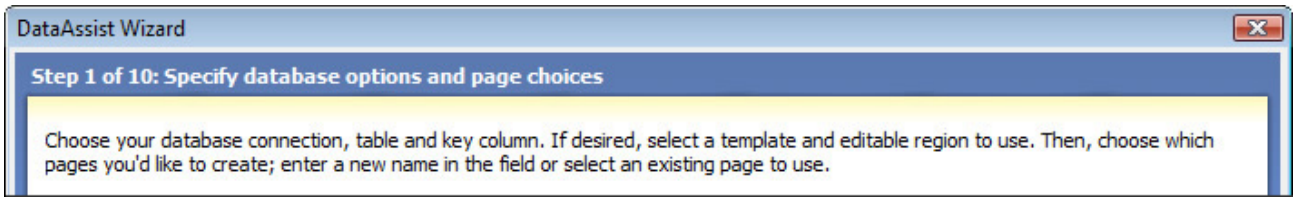
- a file named *Application.cfm* must be included at the root of your site and contain the following code:

  ```
  <cfapplication name="YourSiteName" sessionmanagement="Yes">
  ```

- Your ColdFusion Server must have session variables enabled. Log in to your ColdFusion Server Administrator and under *Server Settings*, click the *Memory Variables* link to navigate to that configuration section. Make sure that *Enable Session Variables* is checked, and click the *Submit Changes* button to update your settings. If you do not administrate your server or have it hosted elsewhere, please contact your hosting provider to confirm these settings.

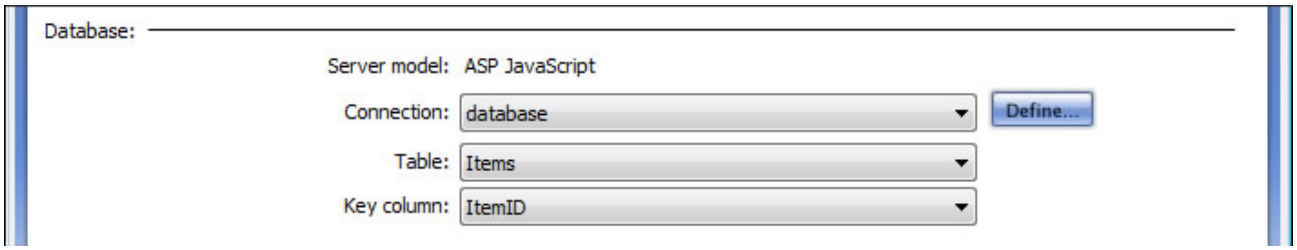
Specify database options and page choices

The first step of the DataAssist Wizard configures the datasource for your application, selects a pre-defined template (if any) to apply to the pages generated by the wizard, and selects the pages to be created. Page types that are selected for creation can use the default file names provdied, or customized specific your naming conventions.



Database

The following criteria is necessary to properly configure your datasource for use within the pages created by the wizard:



Server model: Displays the server model for the currently selected Dreamweaver site definition. Available server models are ASP JavaScript, ASP VBscript, PHP, and Coldfusion.

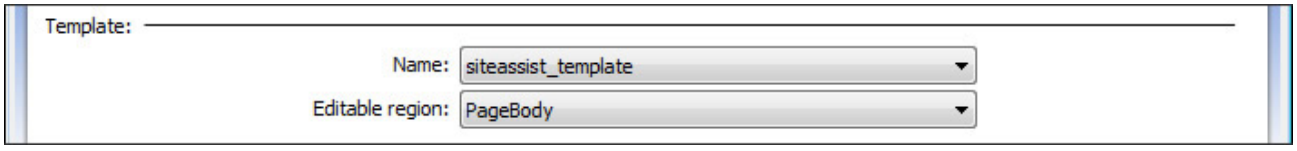
Connection: Selects the database connection used to connect to the database for your application. Select from a list of available connections, or click the *Define* button to configure a database connection.

Table: Selects the table within the chosen database that contains the records to be manged by the pages you are creating.

Key column: Specifies the column in the table that is a unique key for its records. This establishes the identity of individual records so they can be managed appropriately.

Template

This section makes Dreamweaver templates available from the current site definition that can be applied to the pages created by the wizard:

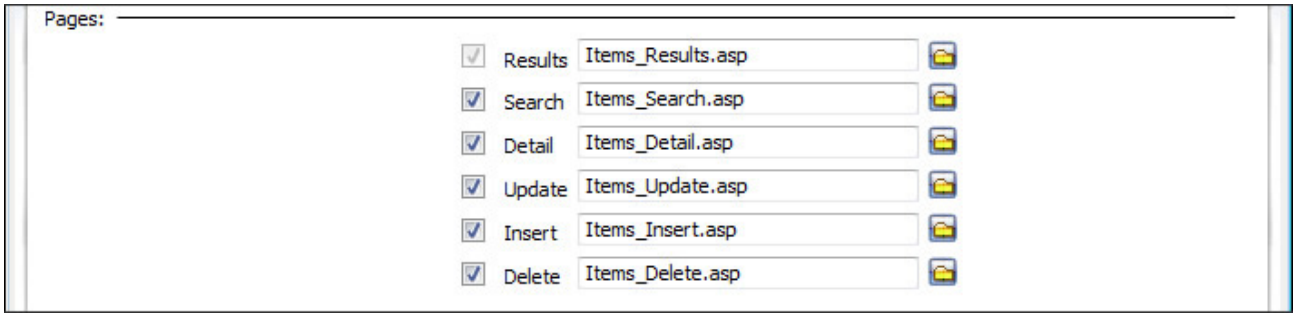


Name: If desired, select the name of the template to be applied to your pages. This list is populated by any templates available with the current site definition.

Editable region: Select the editable region (if any are specified within the selected template) where the content for all created pages is to be placed.

Pages

Six page types are available for creation though the DataAssist Wizard, as follows. Select the page types you wish to be created, specify the name and location for the selected pages, and the next steps in the wizard configure the details specific to each:



Results: Displays the records returned from the database for the given search criteria.

Search: A search form to specify criteria for returning records to a results page.

Detail: Displays specified fields for an individual record. Linked to from a results page.

Update: A form that allows you to update a specific records fields. Linked to from detail page.

Insert: A form that inserts a new record into the database.

Delete: Delete's a specific record from the database. Linked to from detail page.

Select layout options

Step two of the DataAssist Wizard specifies the layout and design attributes for the pages created. These are the global stylistic attributes applied to all pages generated by the wizard.

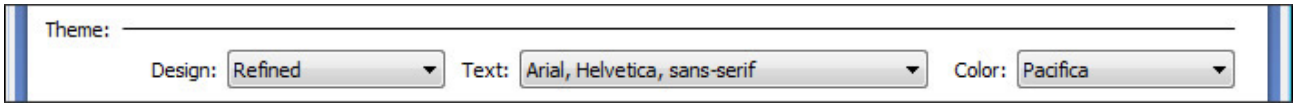
Configure the color scheme, fonts, and design components for your record navigation, results, forms, and form buttons. Determine the type of design elements used for configuration of the record navigation bar applied to your results pages.

See *Recordset paging* in the Specify results page layout options step of the wizard for more information on configuration of record navigation in your results page.

Use the Preview pane to review sample pages based on the criteria specified.

Theme

Defines the overall look and feel of the pages you are creating:



Design: Determines the style of buttons used in form pages, as well as the style of icons or button used for record navigation in results pages. Not applicable on the Results page when the navigation is configured as text (See Results page below).

Text: Sets the font family for the text used in all the page content generated in the wizard. Available options are:

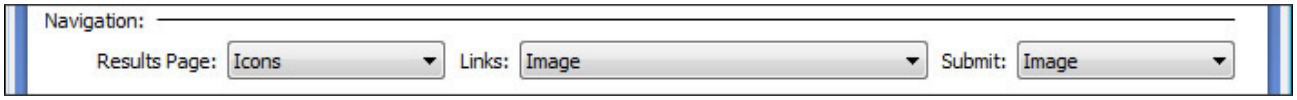
- Arial, Helvetica, sans-serif
- Times New Roman, Times, san-serif
- Courier New, Courier, mono
- Georgia, Times New Roman, Times, serif
- Verdana, Arial, Helvetica, sans-serif
- Geneva, Arial, Helvetica, sans-serif
- Tahoma, Trebuchet MS, Arial, sans-serif

Color: Sets the color scheme for the any icons, buttons, and layouts selected. Available options are:

- Pacifica
- Yosemite
- Moab
- Granite Pine
- Desert Spice
- Cabo Sunset
- Tahiti Sea
- Poppy
- Nautica
- Deep Jungle
- Slate

Navigation

The following options determine the types of page elements used to navigate records on results pages, submit forms, as well as between the pages created by the wizard:



Results page: Select the type of display element used to navigate through the list of records available on the page. Available options:

- Text
- Icons
- Image buttons

Links: Select the display element used to to link to other pages. Available options are:

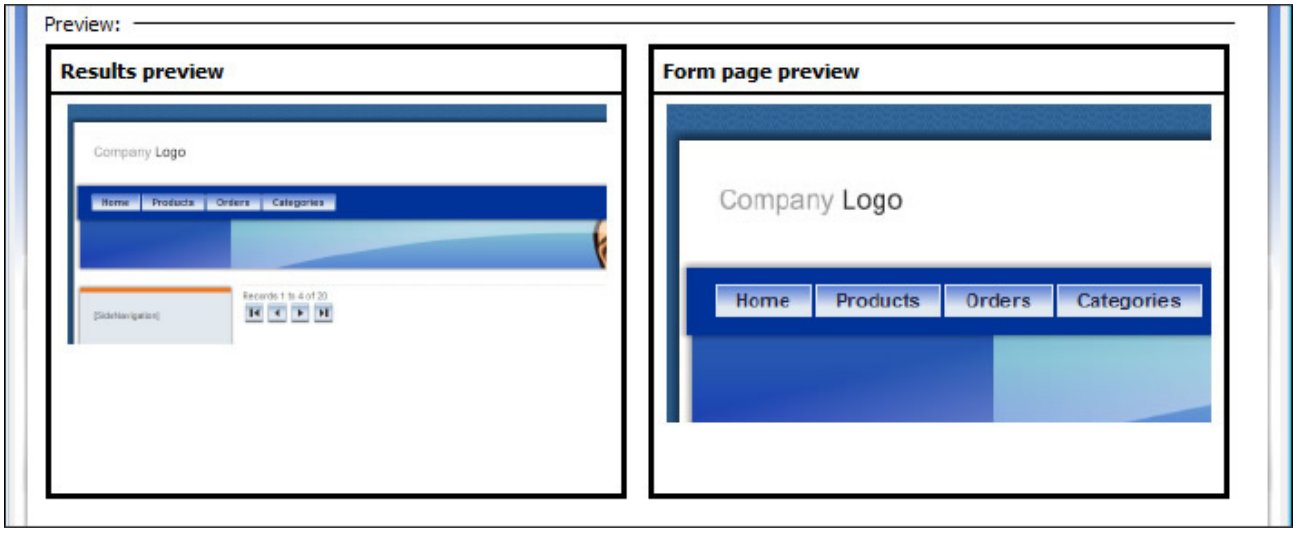
- Text
- Image

Submit: Select the display element used for form submission. Available options are:

- Button
- Image

Preview

The Preview pane displays a representation of the results page and the form pages (e.g. search, insert, update) based on the configuration options you have specified. Clicking either preview pane displays the selected layout in the user's primary browser.



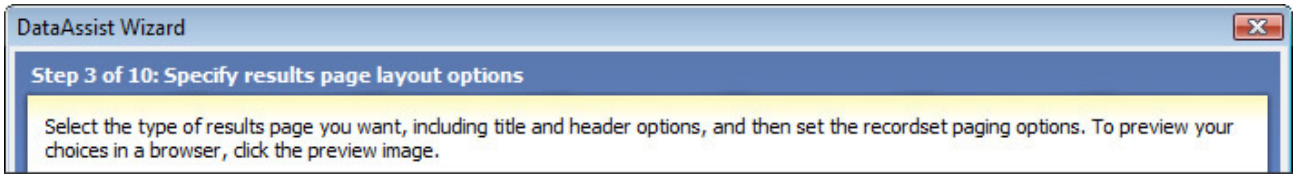
Specify results page layout options

The third step of the wizard configures the layout options for the results page that displays the list of the records returned from your datasource. You have the option of selecting an *Administrative* page type or a *Public* page type.

You can specify the number of rows to be displayed for *Administrative* page types. For *Public* page types you can specify the number of rows, as well as the number of columns.

In either page type, you can select the type of title and header content you wish displayed above the records returned, as well as the text to be returned when no records meet the search criteria passed to the results page.

Configuration options selected populate the preview pane at the bottom of the wizard, allowing you to see a sample of how the page looks prior to completing the wizard.



Results layout options

This section determines the type of page layout to be used, as well as the type of title and header layouts to be included at the top of the results page. Upon completing the wizard, the results page can be edited directly in Dreamweaver to replace the placeholder content included by your selections for the *Title* and *Header* options. You have the flexibility to specify either static content, or information retrieved from a dynamic data source.

Page type: Select from two available page types, determining the way your results are presented on the page. Options are:

- *Administrative*: Optimized for backend administration of your database and simplistic in design. They display returned records that are looped through vertically, and list the text output of your record for the columns selected (this is configured in the Results page options step of the wizard).
- *Public*: Designed to be more visually appealing, this type uses information architecture driven by marketing standards, providing stylized layout options (this is configured in the Results page options step of the wizard), including image display, and multiple records per row.

Title: Configures the content of the title region positioned at the top of the page.

- *None*: No content is placed in the title region at the top of the results page.
- *Page title*: Includes a line of placeholder text in the title region at the top of the page. As well, if you elected to include a search page in Step 1 of the wizard, a link to the search page is included immediately below the title text.
- *Description*: Contains the same content as the *Page title* option, with the addition of a paragraph of placeholder descriptive text below the Page title text (and Search page link, if applicable).
- *Image banner*: Contains the same content as the *Description* and *Page title* options, with the addition of a placeholder image between the title text and the descriptive paragraph. The placeholder image's dimensions are 550 px by 220 px.

Header: Additional placeholder content that can be placed at the top of the results page. This section is aimed at displaying information specific to merchandise, and is useful for product positioning in catalogs. Available options are:

- *None*: No content is placed in the header region at the top of the results page.
- *Single item highlight*: Adds placeholder content to highlight one item. This includes a placeholder image, left aligned, that has dimensions of 269 px by 200 px. To the right of the image, there is placeholder text for the title of the item, descriptive text, and the price, as well as text indicating where text or a button linking to detailed information for this item can be placed.
- *Dual item highlight*: Adds placeholder content to highlight two items, aligned side by side. The two sections containing these items are identical. Each includes a placeholder image positioned above the textual content, that has dimensions of 260 px by 260 px. Below the images, there is placeholder text for the title of the item, descriptive text, and the price, as well as text indicating where text or a button linking to detailed information for this item can be placed.

Recordset paging

Determines the layout for the specified number of records displayed on the page.

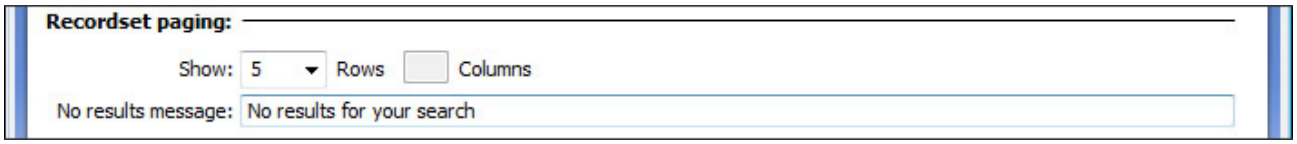
The *Administrative* page type's layout is designed around using a single record per row, so the number of rows specified equates to the number of records displayed on the page.

Public page types can display records in multiple columns as well as multiple rows. The number of columns multiplied by the number of rows specified equates to the number of records displayed on the page. Dreamweaver's Repeat Region server behavior is applied to the page to determine the number of records displayed. In conjunction, DataAssist's Repeat Selection server behavior is configured to specify how many records are displayed in a row.

When more records are returned to the page than are configured to be displayed, link navigation is included using the server behaviors included in Dreamweaver Recordset Navigation Bar, allowing you to page through all records returned at run-time.

See *Help* > *Dreamweaver Help* for more information on the Repeat Region server behavior, as well as Recordset Navigation Bar functionality.

Note: In addition to the wizard, DataAssist also contains the DataAssist Repeating Table feature, specific to configuring results layouts, which assists in applying the server behaviors mentioned above, as well as applying CSS styles to even and odd rows in your results. This is useful for configuring results pages without using the wizard, and provides more direct control over the CSS layout of your results.

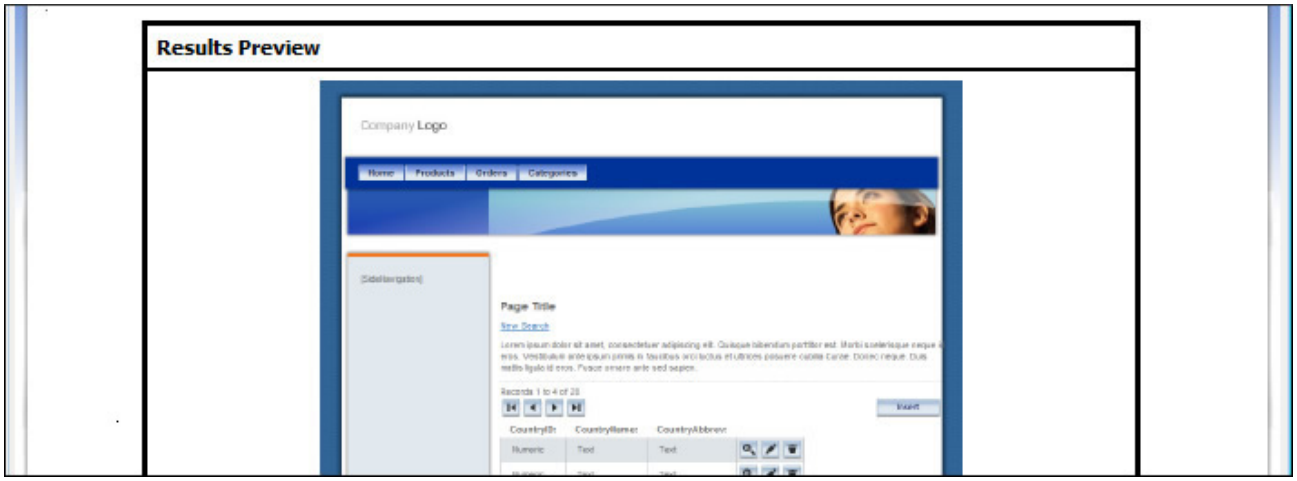


Show: Configures the number of rows and columns displayed on the results page. Dreamweaver's native Repeat Region server behavior is used to determine how many records are displayed on the page. When *Public* is selected as the page type, multiple records can also be titled horizontally within a row, using the Repeat Selection server behavior.

No results message: Specify the text displayed when no records are returned from the database for the search criteria passed to the results page.

Preview

The Preview pane displays a representation of the results page based on the configuration options you have specified. Clicking the preview pane displays the selected layout in the user's primary browser.



Results page options

Step 4 of the wizard continues configuration of the results page, specifying how record columns are displayed on the page for each record returned.

Depending on the page type selected in the Specify results page layout options step of the wizard, the method available for configuring the display of record content differs. For this reason, this page has separate sections detailing how to configure layout options for each, as each type has a unique user interface:

- Administrative page types
- Public page types

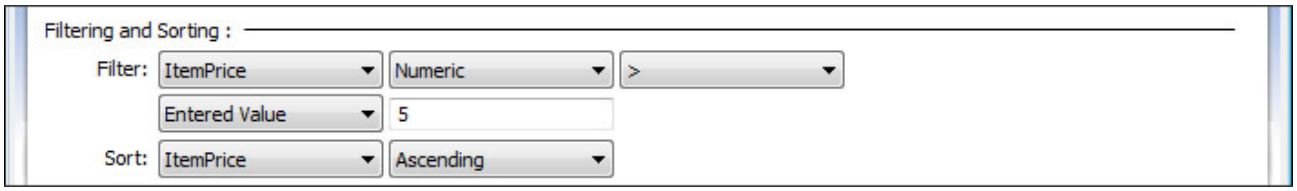
In addition, default filtering and sorting options for the records is configured in this step, updating the recordset applied to the page that returns the results. These filters and sorting options are independent of any search criteria passed to the page, and are intended to remove any records from the datasource that should never be included in the results listing.

Default filtering and sorting options are configured in the same manner, independent of page type. For this reason, this topic is addressed first, immediately following this section.

Filtering and Sorting

Configures the default filters and sorting criteria applied to the recordset returning the results. Used to remove any records from the results that should never be returned to the results page, and determine the default order of the results when they are initially returned. These options are configured directly within the recordset applied to the results page, and can be updated directly in the recordset through the server behaviors panel once the page has been created.

Note: Advanced sorting functionality can be added to the results page after completing the wizard. See Sort for more information on this additional feature in DataAssist.



Filter: Filters the records returned by default from the available datasource. Configured based on the comparison criteria specified in the given form fields according to the following formula:

[Database Field] [Format Type] [Comparison] [Variable/Parameter Type] [Variable/Parameter]

e.g. ItemPrice {Numeric} > Entered Value {5}

The above example compares a numeric database column containing the price of an item to a specified value of 5, returning all records where the price is greater than 5.

The following details each of the comparison criteria available in more detail:

- **Database Field**: selects the database column that will be compared against to filter the results
- **Format Type**: selects the format of the data in the selected database field. Validates that the data comparison is made using the appropriate format. Available types include:
  - Numeric
  - Date,
  - Date MS Access
  - Checkbox Y,N
  - Checkbox 1,0
  - Checkbox -1,0
  - Checkbox MS Access
- **Comparison**: Selects the type of comparison to be made. Comparisons against a string can be made using equals, begins with, ends with, or includes. Comparisons against a date or numeric value can be made using equals, greater than, greater than or equal to, less than, and less than or equal to. Greater than and less than comparisons are not available in MS Access databases.

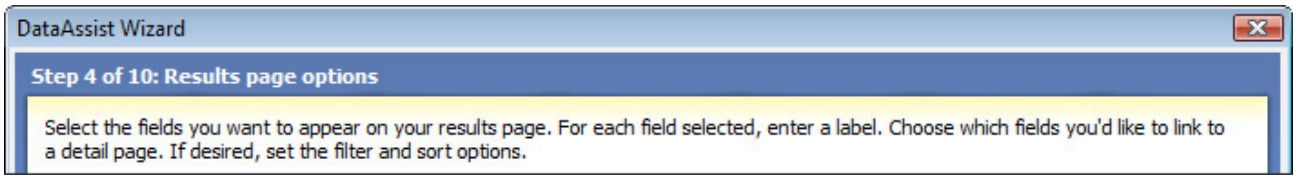
- **Variable/Parameter Type:** the type of variable or parameter that contains the comparison criteria. Ensures that the formatting is correct to appropriately make the comparison. Available types are:
  - Form Variable
  - URL Parameter
  - Cookie
  - Session Variable
  - Application Variable
  - Entered Value
- **Variable/Parameter** - the value, or the variable containing the value, that the database column is to be contrasted against using the selected comparison.

Sort: Sorts the results returned by a selected database column in ascending or descending order.

Note: Advanced sorting functionality can be added to the results page after completing the wizard. See Sort for more information on this additional feature in DataAssist.

Administrative page types

Results returned for *Public* page types are listed vertically, one above the other (see the Specify results page layout options step of the wizard for more information on this page type). A simple design implementation, this layout type returns all selected database columns in a single row for a given record. It is useful for creating administrative pages intended to manage database content through a backend interface.

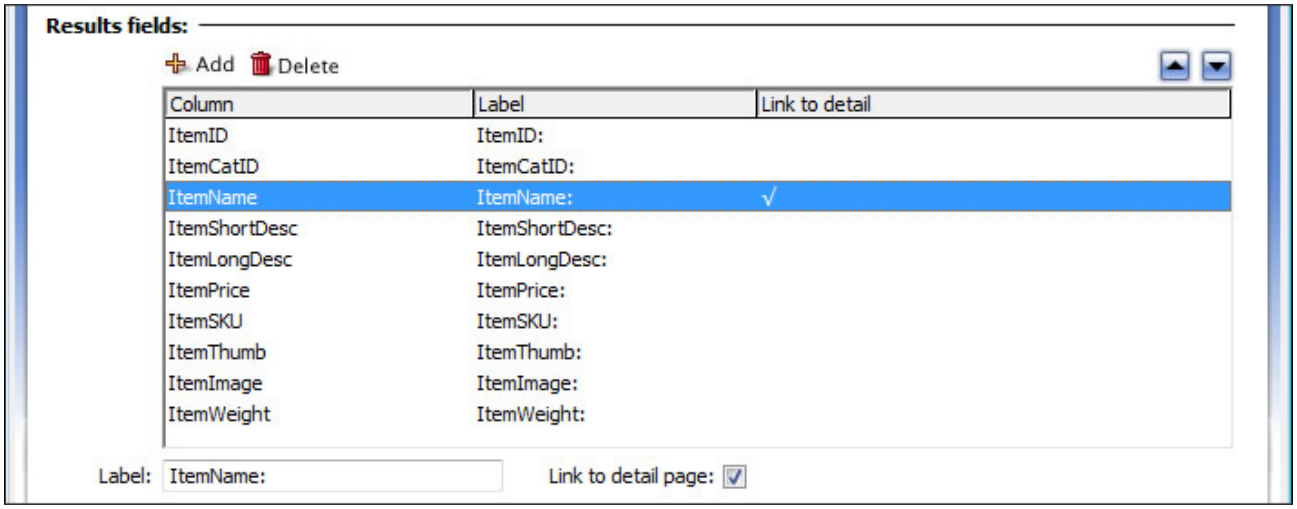


Results fields

Select the database columns to be displayed for each record returned, the labels used in the first row of the results table that identifies each column, and select if the content in a given column links to the detail page for the record selected.

By default, all columns for the current datasource are configured for display.

- Add : Click to select additional columns from the datasource to display in the results.
- Delete : Click to remove a column selected in the configuration pane from the results display.
- : Columns are ordered left to right in the results display based on the order they are listed top to bottom in the configuration pane. Use the Up and Down buttons to change a selected field's position within the hierarchy.



To modify a field, select it within the display pane. Once selected, you can modify the Label and link attribute for the field:

Label: The text displayed in the first row of the results table to identify the columns displayed.

Link to detail: When this option is selected for a given column, the contents of the column in the results page links to the details page. The detail page is populated with the information specific to the record selected on the results page.

Note: For more information on configuration of content displayed in detail pages, see the Details page options step of the wizard.

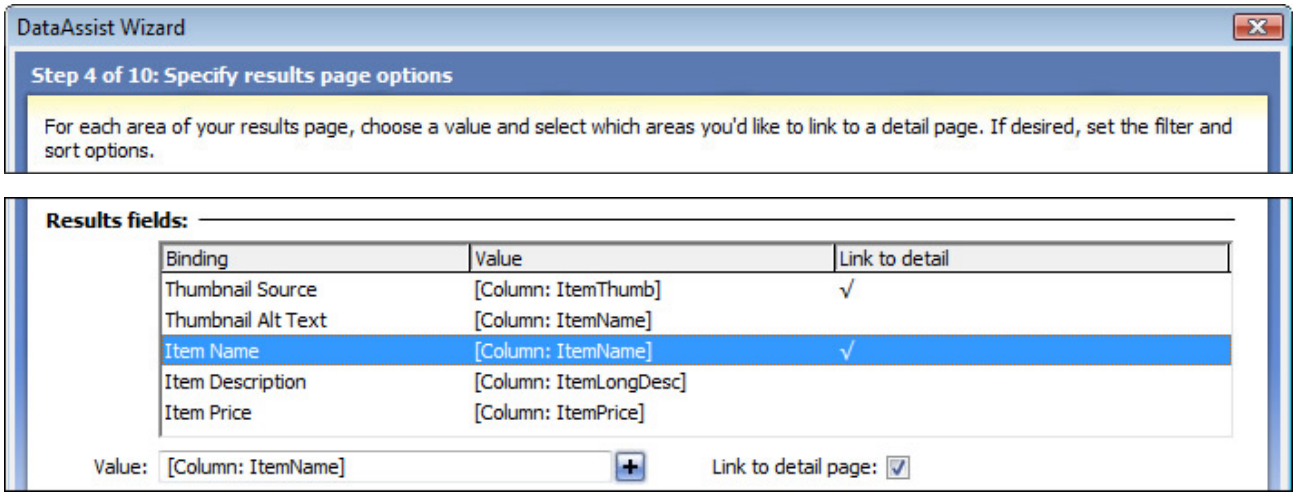
Hint: Upon completing the wizard, the results page can be edited directly in Dreamweaver, allowing you to configure display columns specific to your needs, and allowing more advanced configuration of results displays. For example, a database column may return the location of a thumbnail image, and you may decide to return this information to a column in the results display. After the page has been created, you can edit it in Dreamweaver, and add a placeholder image within that column, setting the image source equal to the database column returning the location of that image. This would allow you to display images specific to your records based on database content.

Public page types

Public page types are designed with the intent of presenting information to a consumer, such as in a product catalog (see the *Page type* section of the Specify results page layout options step of the wizard for more information on this page type).

There are two options for the display of results returned for *Public* page types; either a single record per result row, or multiple records per row, as configured in the *Recordset paging* section of the Specify results page layout options step of the wizard.

The layouts for each of these types have layout regions that correspond to information in your datasource. Columns in your datasource are bound to each of these regions using the provided control, presenting the information in your records in the prescribed wireframe format. For reference. the wireframe preview provided at the bottom of the screen indicates how each available binding corresponds to their position in the record layout.



To update the Value and link attribute of an available binding, select it within the display pane.

Value: Click the *Plus (+)* button to associate an available datasource column to the layout binding. The default value is *Default*, which associates placeholder content to the binding as appropriate for the binding type. If you wish to specify your own placeholder content to be displayed if no information is available for a given record returned by the datasource, specify the value directly between two asterisks (i.e. *Placeholder*), or enter server-side code to determine this value dynamically.

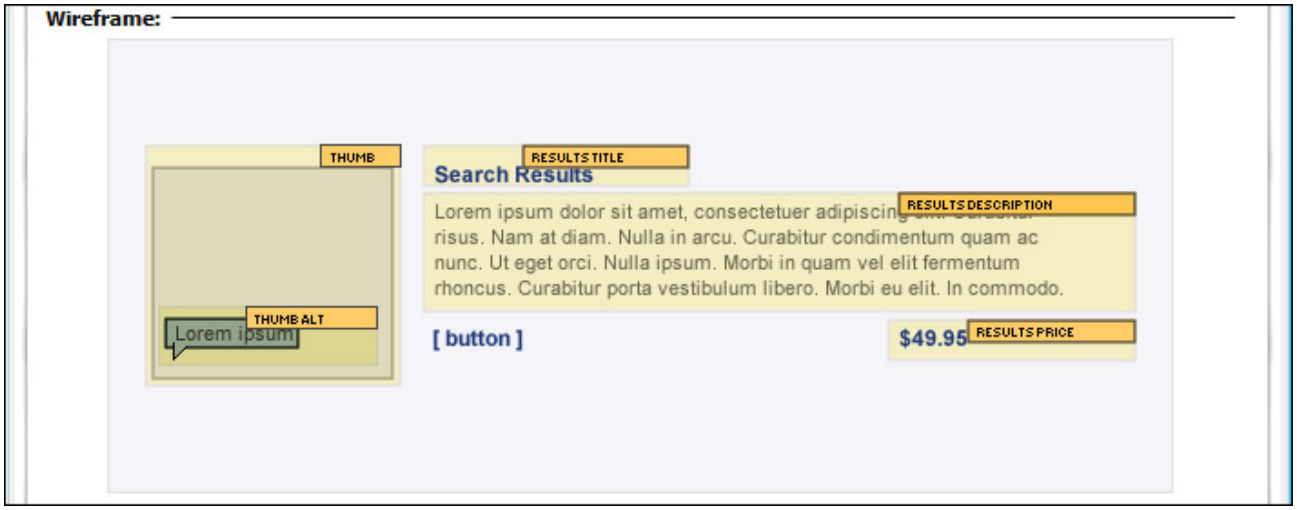
Link to detail: When this option is selected for a selected binding, the associated region in the layout links to the detail page. The detail page is populated with the information specific to the record clicked on the results page.

Note: For more information on configuration of content displayed in detail pages, see the Details page options step of the wizard.

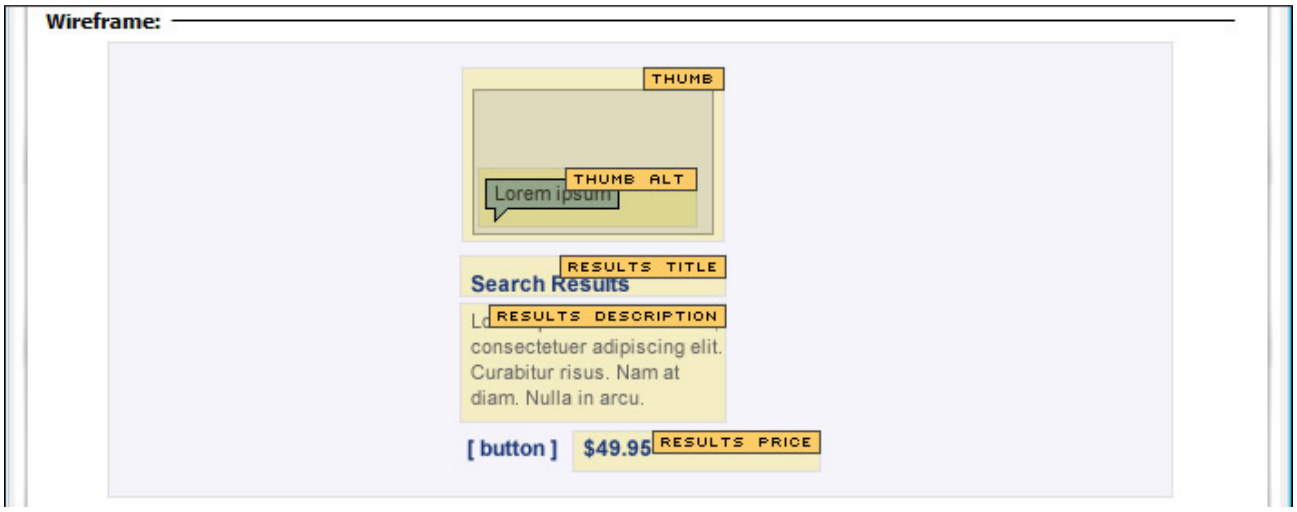
Wireframe

This section displays a sample of the layout for a single record in your results table. There are two standard layouts, selected automatically depending on whether you have configured one column or multiple columns to display individual records (based on configuration options specified in the *Recordset paging* section of the Specify results page layout options step of the wizard).

Individual record layout for a single record in a row



Individual record layout for multiple records in a row



Search page options

This step of the wizard configures a search page that filters results returned to the results page based on specified criteria. Configure a search form for the page that compares form element entries/selections against fields in your database using the criteria defined.

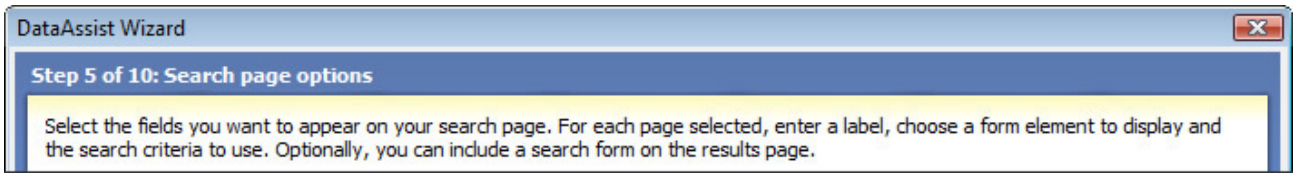
Specify the type of form element, its corresponding label within the form, and the comparison criteria used to perform a comparison against a given datasource column. The resulting form lists each comparison separately in individual vertical rows within the table.

Multiple comparisons can be made against the same database column. This requires that columns can be specified multiple times for comparison. An entry for each type of comparison and its corresponding form field needs to be added to the list of configured search fields.

Dynamic lists are available for configuration as form elements to make a search comparison. See DataAssist Dynamic Menu for more information specific to configuration of Dynamic Lists in the DataAssist wizard for Search pages, as well as Update and Insert pages.

Select whether an additional form is included in the results page to allow for searching initiated from the results page.

Note: In addition to creating the search page, this step of the wizard configures the DataAssist Search server behavior with the search criteria specified. This server behavior is applied to the results page, and filters the results set specific to the search criteria passed from the search forms present on both the search page or results page. DataAssist allows you to configure search functionality outside of the DataAssist Wizard, using either the DataAssist Search Wizard, or by directly configuring the DataAssist Search server behavior. Advanced search configurations beyond those available through either wizard is available through direct configuration of the server behavior.

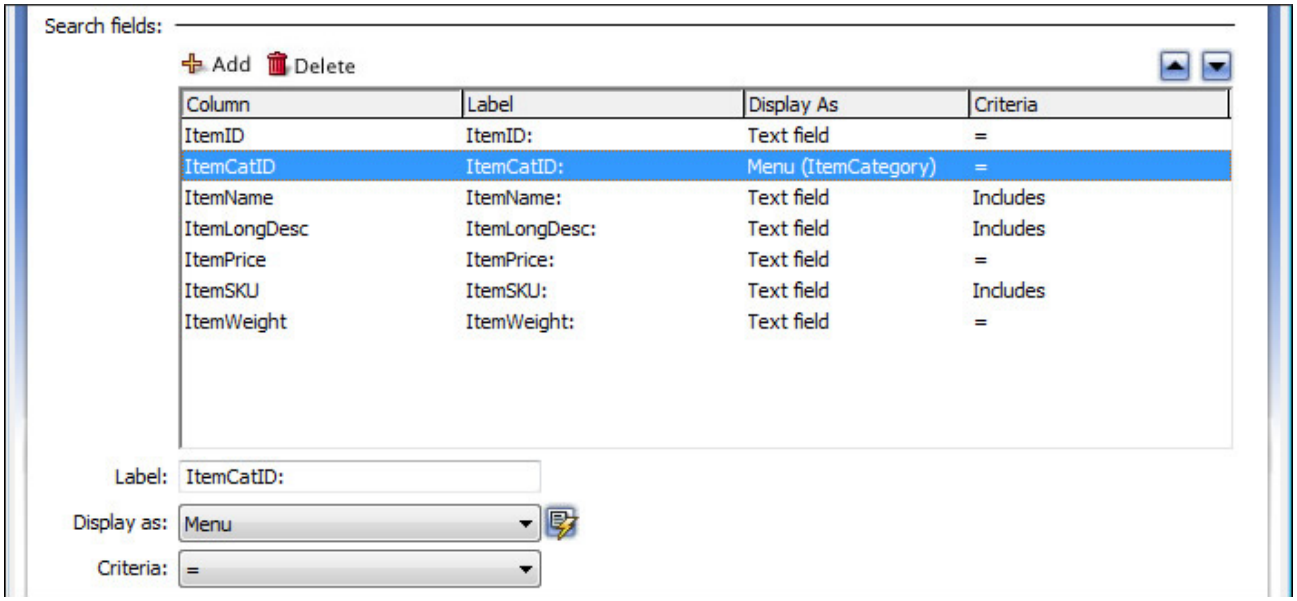


Search fields

Select the database columns that comparisons are to be made against. Configure the form element used to make the comparison against the datasource column, the form element's display label, and the comparison criteria used against the data contained in the specified column.

By default, all columns for the current datasource are configured for display in the *Search fields* control pane.

- Add : Click to add additional columns from the datasource to the search form to specify comparison criteria and the necessary form element.
- Delete : Click to remove a column comparison from the search form.
- : Form elements are listed vertically, one per row, in the search form. Use the Up and Down buttons to change a selected field's position within the display hierarchy.



To modify a column, select it within the display pane. Once selected, you can modify the Label, the form element type used to compare values against that column, and the type of comparison made:

Label: Sets the label text that identifies a form element/comparison.

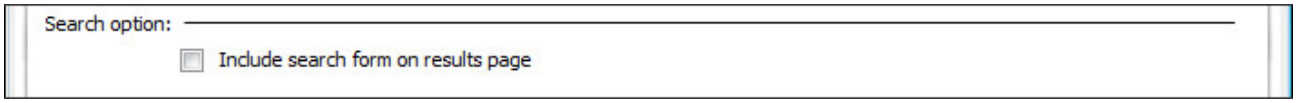
Display as: Sets the type of form element used to enter/select search criteria. The types of form elements available are:

- **Text field**
- **Text area**
- **Menu**: Menus may contain static or dynamic list information to be used for a comparison. Static entries are configured directly on the page upon completing the wizard. Dynamic list entries can be configured directly within the wizard or directly upon the page upon completing the wizard. Click the lightning bolt icon to access the user interface for dynamic menu configuration. Configuration of dynamic list entries is discussed in detail in DataAssist Dynamic Menu.
- **Hidden field**
- **Check box**
- **Radio group**
- **Password field**
- **Text**

Criteria: A specific comparison of available data in a database column against criteria defined in its associated form element. Types of comparisons available are:

- **Begins with**: Compares the entered string against the beginning characters of a database column's contents looking for an exact match.
- **Ends with**: Compares the entered string against the final characters of a database column's contents looking for an exact match.
- **Includes**: Determines if the string entered is contained in its entirety within the string for that column in the database record.
- **Equals (=)**: Compares the entered string against a database column looking for an exact match.
- **Greater than or equal to (>=)**: Looks for numeric values (date and integer) hat is equal to or greater than the entered value.
- **Less than or equal to (<=)**: Looks for numeric values (date and integer) that is equal to or less than the entered value.

Search option



Include search form on results page: Select this field to include this same search form on your results page. For improved workflow in your application, placing the search form on this page makes it easier to perform a new search without returning to the search page. This is ideal, as long as it doesn't detract from your results presentation. You can always manually remove search fields to minimize the search form applied to the results page to keep only the most common search comparisons readily available for the convenience of your users.

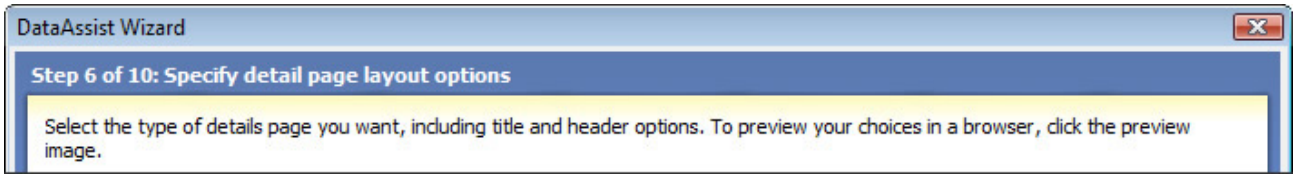
Specify detail page layout options

This step of the wizard configures the layout options for the details page that displays information for a specific record returned from your datasource. You have several detail page types available to select from that are specific to various record types.

Links to the Results page, as well as the Update and Delete pages are added as appropriate, depending on if they are specified to be created in Step 1 of the wizard, Specify database options and page choices.

You can select the type of title and header content you wish displayed above the record content.

Configuration options selected populate the preview pane at the bottom of the wizard, allowing you to see a sample of how the page looks prior to completing the wizard.



Detail layout options

This section determines the type of page layout to be used, as well as the type of title and header layouts to be included at the top of the details page. Upon completing the wizard, the detail page can be edited directly in Dreamweaver to replace the placeholder content included by your selections for the *Title* and *Header* options. You have the flexibility to specify either static content, or information retrieved from a dynamic data source.

Page type: Select from seven available page types, determining the way the record is presented on the page. This content determines how content for the record is to be configured for display in the following step, Details page options. Layout options are:

- *Standard*: Returns each column of the record specified for display in vertical rows, one database column per row. Columns displayed are configured in the next step of the wizard, Details page options.
- *Multi-Image Product*: A stylized layout displaying multiple images. Intended for catalog layouts, a primary placeholder image is displayed at the top of the record display, aligned to the left, with dimensions of 260 px by 260 px. Aligned to the right of the primary image are four secondary images, stacked two high, each having dimensions of 120px by 120 px. Beneath all of these images is a paragraph of placeholder text intended to contain a short description for the record. Beneath this short description is a region, aligned left, that has placeholder text for additional sub text, the price, as well as text indicating where text or a button linking to detailed information for this item can be placed. To the right of this region is placeholder content for a long description in the record.
- *Single-Image Product*: A stylized layout displaying a single image. Intended for catalog layouts, the placeholder image is displayed at the top of the record display, aligned to the left, with dimensions of 250 px by 200 px. Aligned to the right of the image is a region, that has placeholder text for additional sub text, the price, as well as text indicating where text or a button linking to detailed information for this item can be placed. Beneath the all of the aforementioned content is placeholder paragraphs for short and long descriptions, stacked vertically.
- *Maximized Image*: A stylized layout displaying a single image. The placeholder image is displayed at the top of the record display spanning the entire display region, with dimensions of 520 px by 370 px. Beneath this image is placeholder text for a subheading, as well as for a descriptive paragraph.
- *Text Only Details*: A simple text layout. Stacked placeholder text for a page heading, a short description and a long description.
- *North American Address*: An address display for locations stored in North American format. Stacked vertically with placeholder test for page heading, individual address components, a short description, and a long description.
- *International Address*: An address display for locations stored in International format. Stacked vertically with placeholder test for page heading, individual address components, a short description, and a long description.

Title: Configures the content of the title region positioned at the top of the page.

- *None*: No content is placed in the title region at the top of the results page.
- *Page title*: Includes a line of placeholder text in the title region at the top of the page. As well, if you elected to include a search page in Step 1 of the wizard, a link to the search page is included immediately below the title text.
- *Description*: Contains the same content as the *Page title* option, with the addition of a paragraph of placeholder descriptive text below the Page title text (and Search page link, if applicable).
- *Image banner*: Contains the same content as the *Description* and *Page title* options, with the addition of a placeholder image between the title text and the descriptive paragraph. The placeholder image's dimensions are 550 px by 220 px.

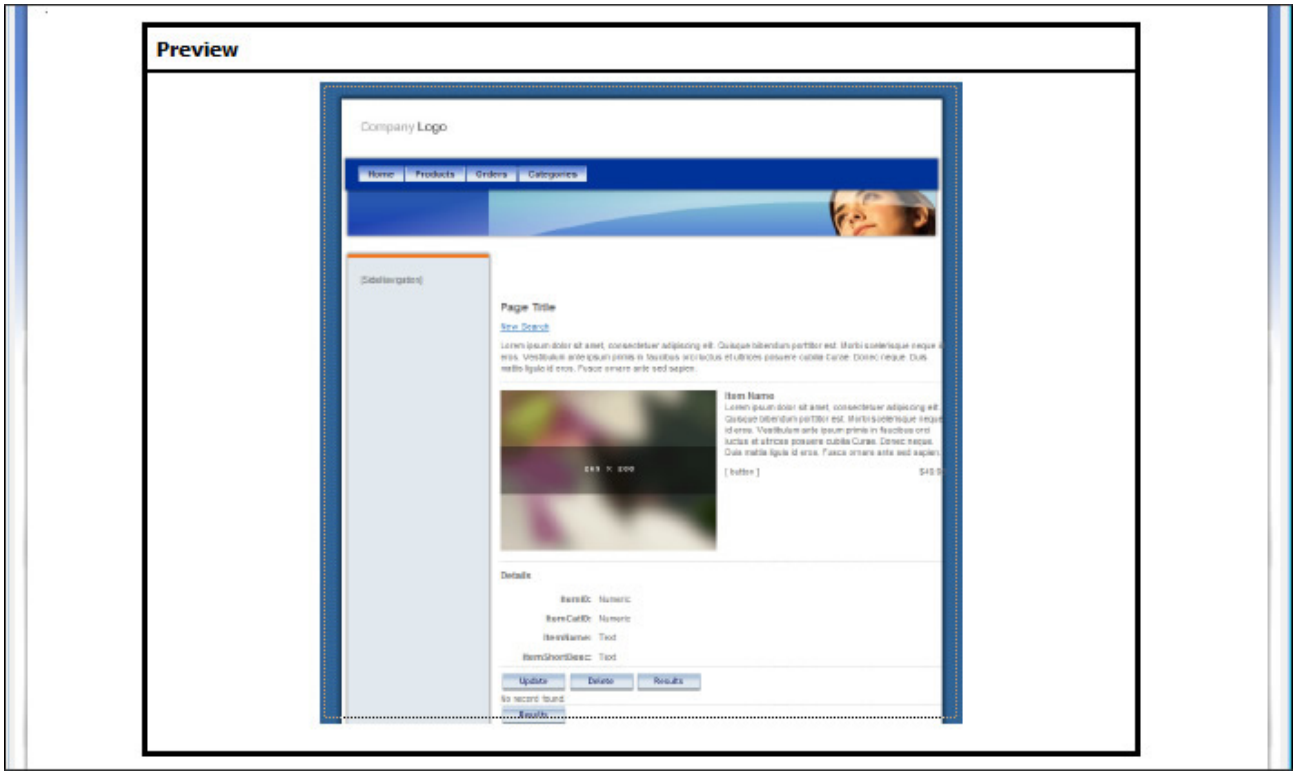
Header: Additional placeholder content that can be placed at the top of the results page. This section is aimed at displaying information specific to merchandise, and is useful for product positioning in catalogs. Available options are:

- *None*: No content is placed in the header region at the top of the results page.
- *Single item highlight*: Adds placeholder content to highlight one item. This includes a placeholder image, left aligned, that has dimensions of 269 px by 200 px. To the right of the image, there is placeholder text for the title of the item, descriptive text, and the price, as well as text indicating where text or a button linking to detailed information for this item can be placed.
- *Dual item highlight*: Adds placeholder content to highlight two items, aligned side by side. The two sections containing these items are identical. Each includes a placeholder image positioned above the textual content that has dimensions of 260 px by 260 px. Below the images, there is placeholder text for the title of the item, descriptive text, and the price, as well as text indicating where text or a button linking to detailed information for this item can be placed.

Preview

The Preview pane displays a representation of the detail page based on the configuration options you have specified. Clicking the preview pane displays the selected layout in the user's primary browser.

Note: Below the record details, the preview displays a region that contains the text "No records found", with a button linking to the results page beneath it. At run-time, these only show up if you visit the details page without a valid record ID in the query string. This region is included on the page as an alternate display to the record content area when a record passed to the page does not exist. It is displayed proximate to the record detail content by the preview engine, and in Design view in Dreamweaver, when editing the details page directly. Only one region or the other will ever appear to the end user, depending on whether the record ID is valid or not.



Details page options

The details page is linked to from the results page when a selected record's detail link is clicked. It displays selected database columns for the given record.

This step of the wizard continues configuration of the details page, specifying how record columns are displayed on the page for each record returned.

Depending on the page type selected in the Specify detail page layout options step of the wizard, the method available for configuring the display of record content differs. For this reason, this page has separate sections detailing how to configure layout options for each, as each type has a unique user interface:

- Standard page type: This a simple stylistic design that displays each database column as a unique row in a table. Requires the identification of fields to be displayed and the labels that correspond to them.
- Other page types: An additional six page designs that use more complicated layout architecture, requiring that the details from the record be bound to display regions within a given layout.

Standard page type

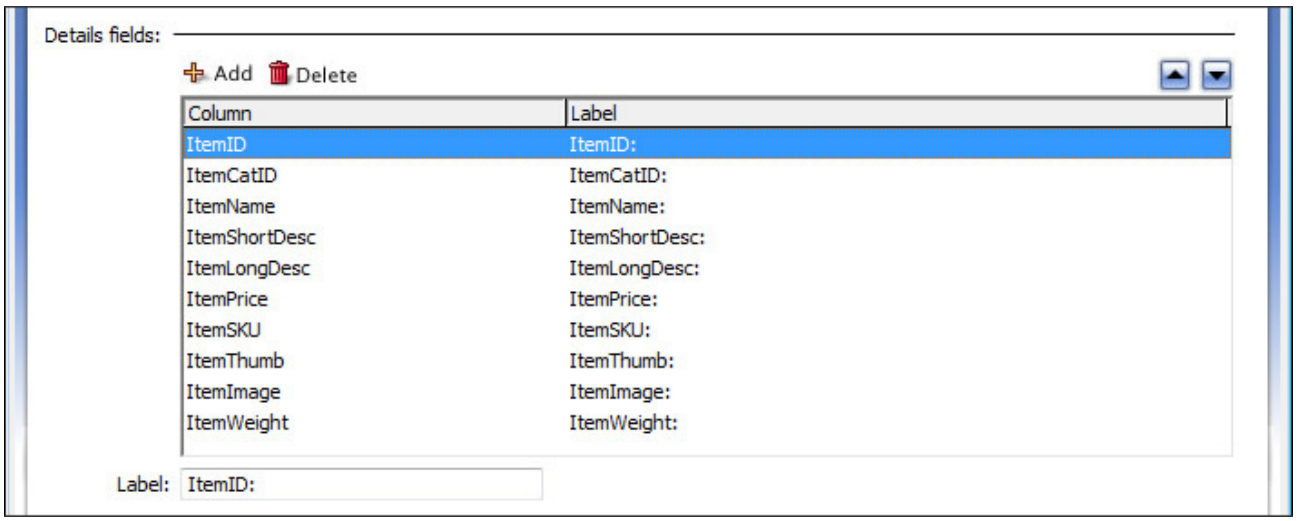
Select the database columns and corresponding labels that are to be listed vertically on the page.

Details fields

Select the database columns to be displayed on the page. Configure the display label used to identify the contents of each column for the record on the page.

By default, all columns for the current datasource are configured for display in the *Details fields* control pane.

- Add : Click to add additional columns from the datasource for display.
- Delete : Click to remove a column from the display.
- : Columns are listed vertically, one per row, in the detail display. Use the Up and Down buttons to change a selected field's position within the display hierarchy.

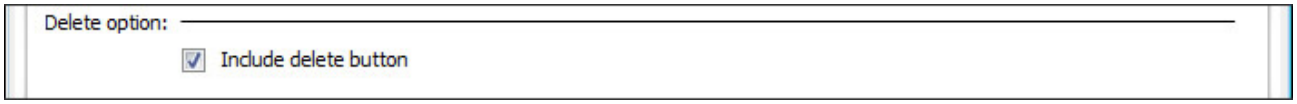


To modify a columns label, select it within the display pane.

Label: Sets the label text that identifies a column in the detail display.

Delete option

Select this checkbox to include a *Delete* button on the page. This button does not remove any records, but links to the delete page created by the wizard.



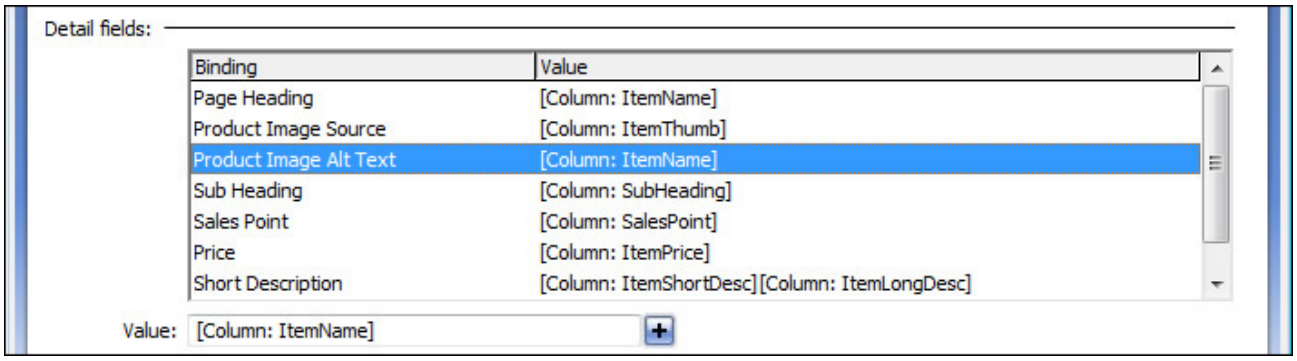
Other page types

There are six detail page designs specific to common types of data displays. they offer options for product catalog displays, text only, image displays, and address displays. Each design positions content from the record in a unique location, and carries different placeholder references specific to the type of data displayed in each.

Using the *Detail fields* control, bind the appropriate database column to the region of the page where it is displayed or referenced. The wireframe displayed at the bottom of the interface should be used as a reference for each location by identifying the binding it correlates to.

Detail fields

Select a binding to update the value to be displayed in the given layout. The default value is *Default*, which associates placeholder content to the binding as appropriate for the binding type. If you wish to specify your own placeholder content to be displayed if no information is available for a given record column, specify the value directly between two asterisks (i.e. *Placeholder*), or enter server-side code to determine this value dynamically.

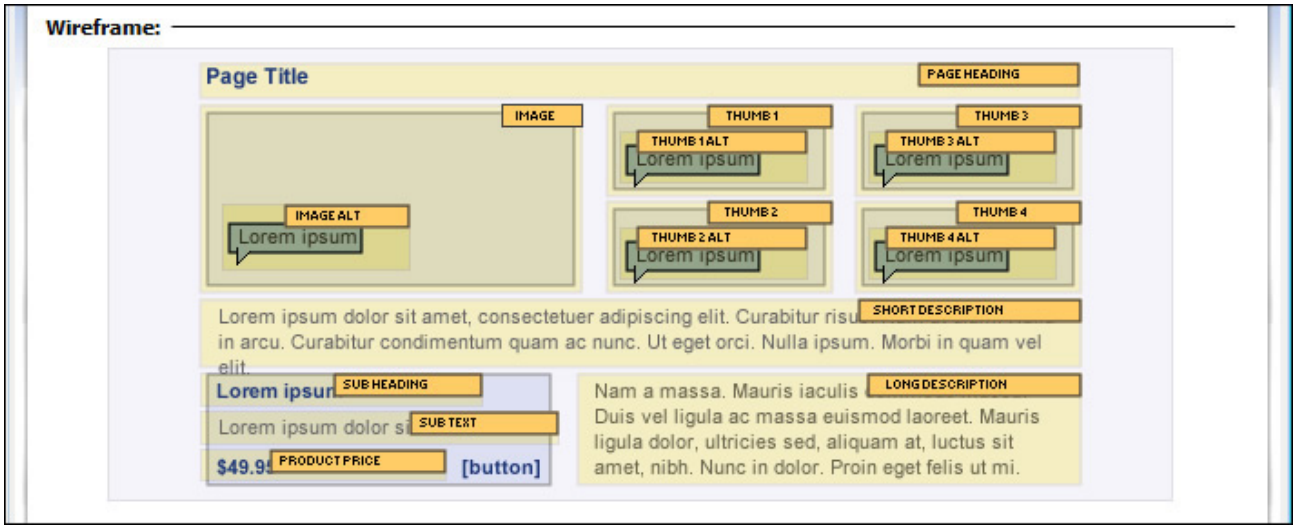


Label: Sets the label text that identifies a column in the detail display.

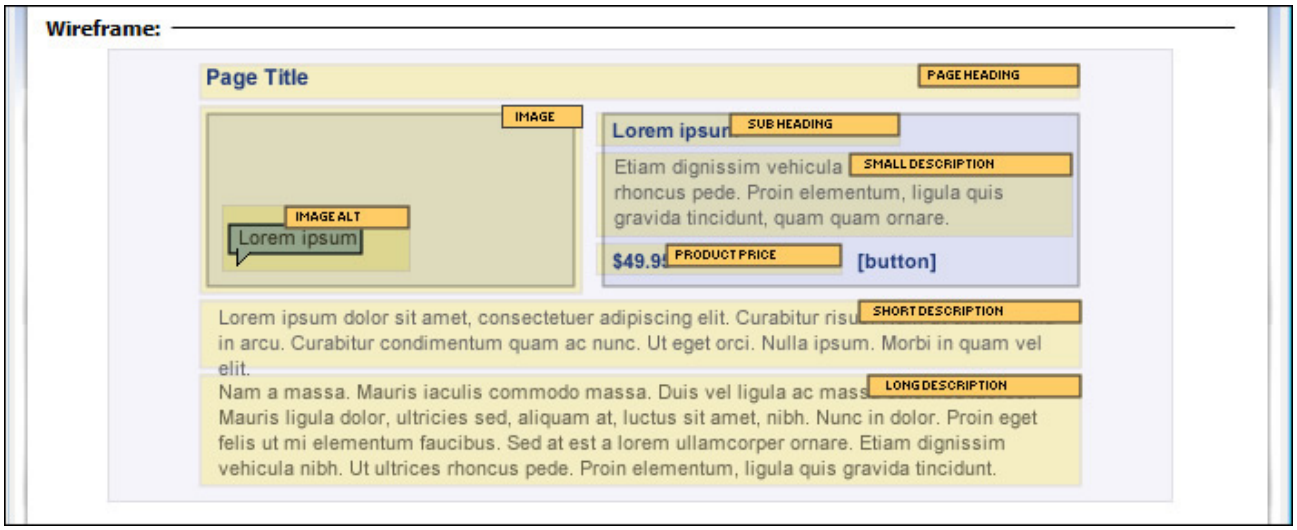
Wireframe

This section displays a sample of the layout for the detail page types selected in the prior step, Specify detail page layout options. There are six page types with wireframe previews available that can be seen below. The one page type not included is the *Standard* page type, as it lists content in vertical rows and does not require stylized bindings. Each wireframe displays a sample of where available display bindings are located in the context of a given page type.

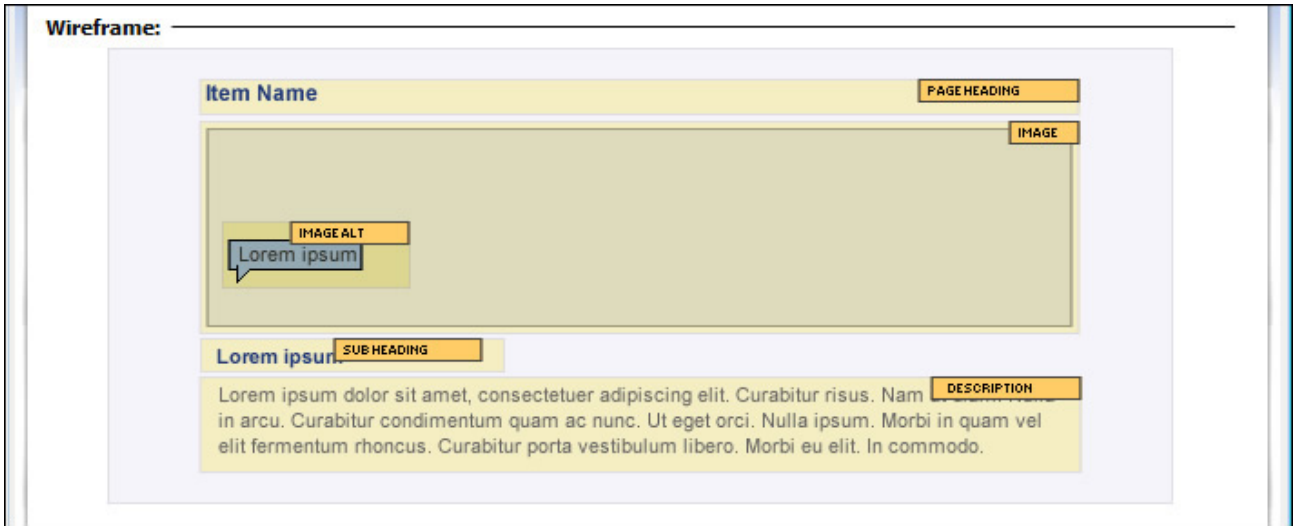
Multi-Image Product



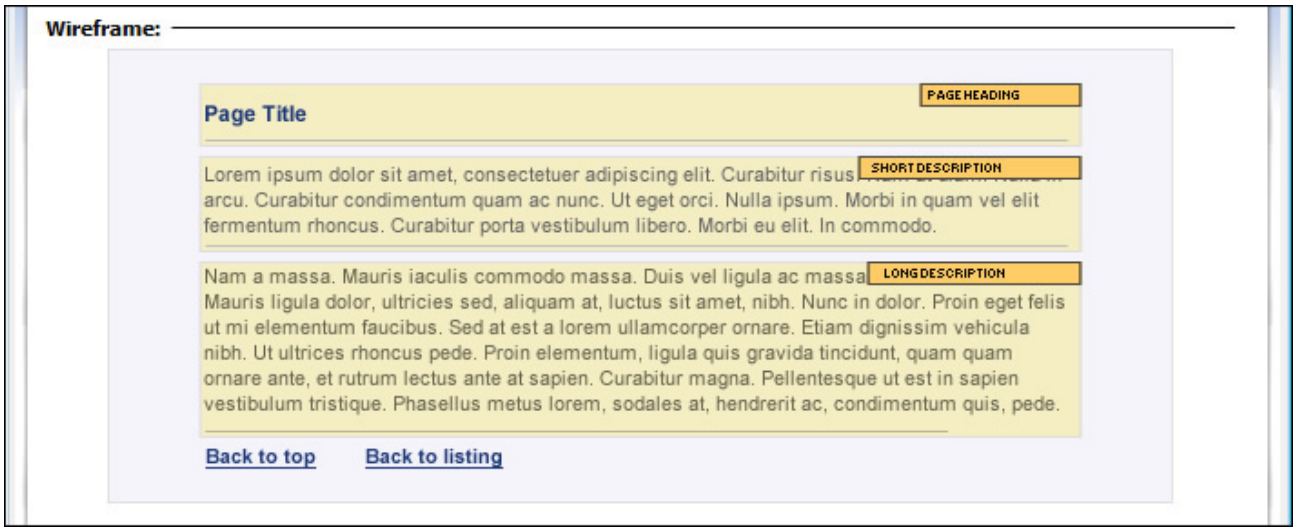
Single-Image Product



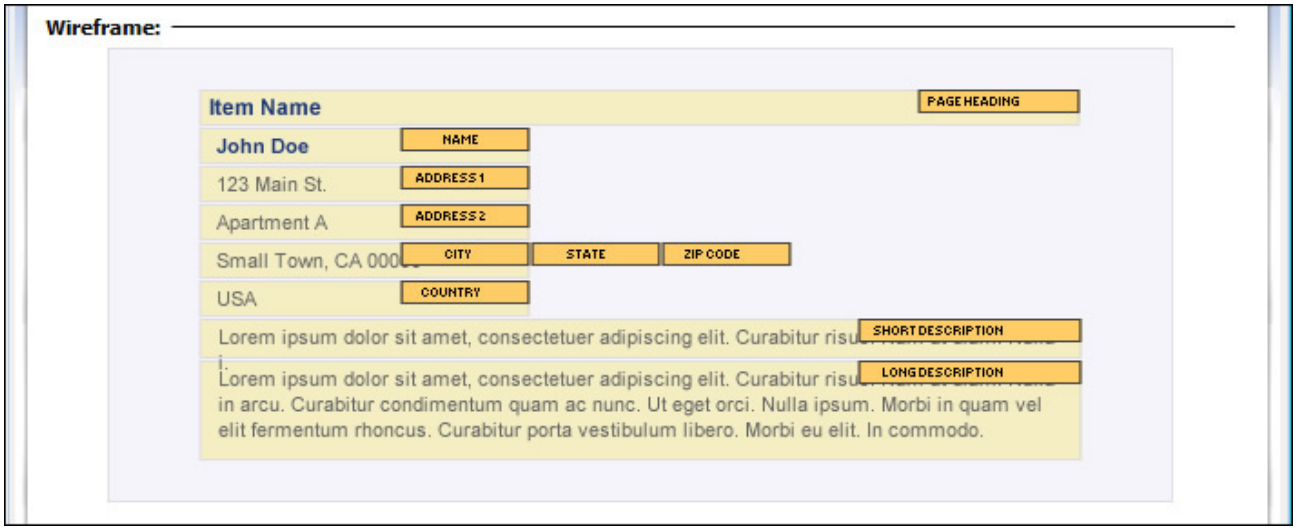
Maximized Image



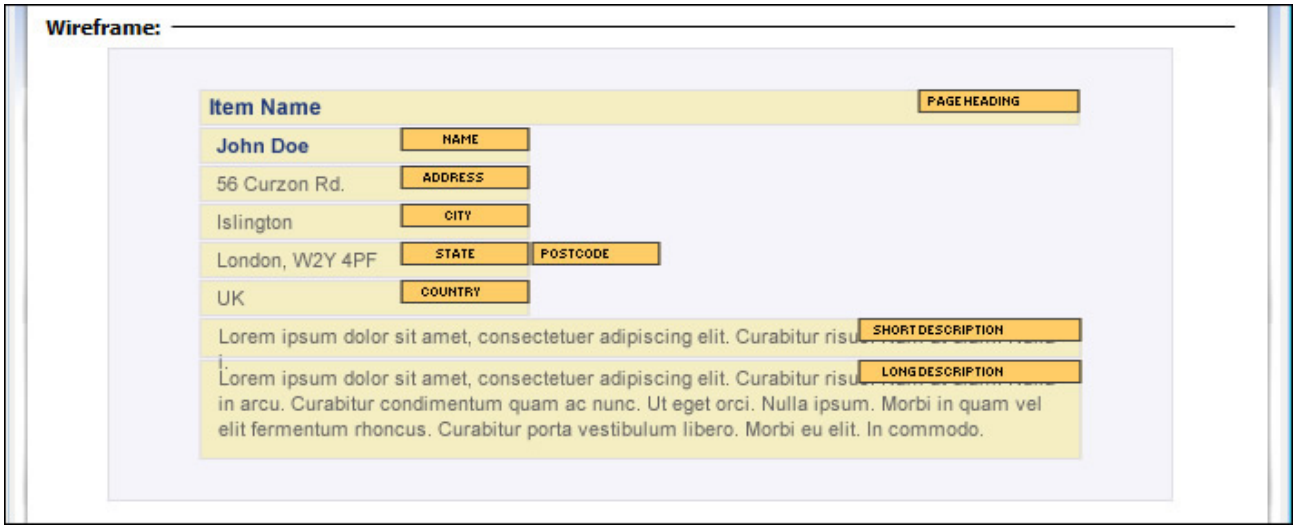
Text Only Details



North American Address



International Address



Update page options

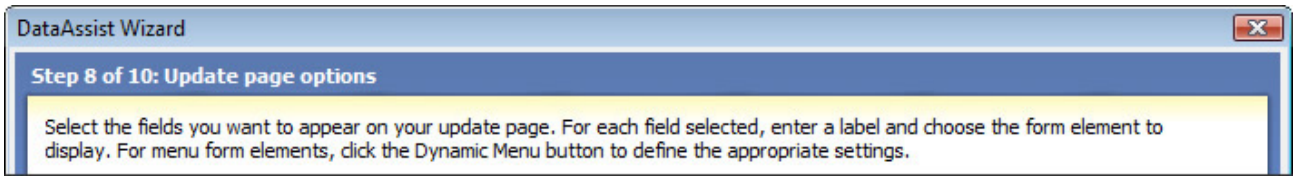
Update pages display a given record's data in a form, allowing changes to be made and submitted to the database. Update pages are linked to from detail pages and are populated with the record displayed by the detail page. They contain an Update button to perform the update, and a Cancel button to return the user to the results page.

To configure your update page, specify the database columns to be updated. For each column updated, define the form element's type and label. Columns that are not updated can still be displayed as text on the page.

Dynamic lists are available for configuration as form elements to update a column with a value selected from a list of values returned from your datasource. See DataAssist Dynamic Menu for more information specific to configuration of Dynamic Lists in the DataAssist wizard for Update pages, as well as Search and Insert pages.

A Dreamweaver recordset is applied to this page to populate it with the record passed from the results page. Additional recordsets may also be added to the page specific to populating any dynamic lists added to the form. As well, the Update Record server behavior is applied to the page to perform the update to the record in the datasource. This server behavior is also available for use outside of the wizard, allowing you to add record update functionality to pages specific to your needs.

Note: The DataAssist Wizard creates update pages for single records. The Update Multiple Records server behavior can be used outside of the DataAssist wizard in conjunction with the Repeat Selection server behavior to update multiple records to a datasource at one time. See Managing Multiple Records for more information.

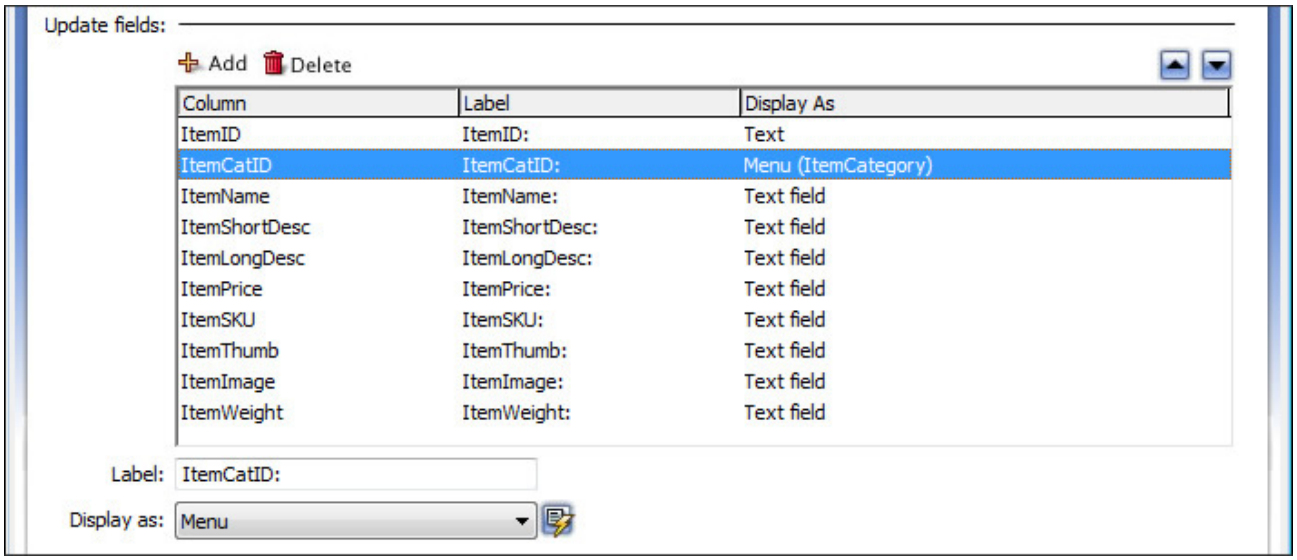


Update fields

Select the datasource columns to be updated for the given record. Also include any columns to be displayed, but not updated. Configure the type of form elements used to make the updates against specific datasource columns, as well as the labels identifying those form elements.

By default, all columns for the current datasource are configured for inclusion in the *Update fields* control pane.

- Add : Click to add additional columns from the datasource to be updated.
- Delete : Click to remove a column to be updated/displayed from the update form.
- : Form elements and text are listed vertically, one per row, in the update form. Use the Up and Down buttons to change a selected column's position within the display hierarchy.



To modify a field, select it within the display pane. Once selected, you can modify the label and the type of form element used for the update, or set the column to be displayed as text if no update should occur.

Note: Key columns should always be set to display as text and should not be available for update, as attempting to update them will cause errors in the application.

Label: Sets the label text that modifies a form element.

Display as: Sets the type of form element used to enter/select updated record information. The types of form elements available are:

- **Text field**
- **Text area**
- **Menu**: Menus may contain static or dynamic list information to be used for a comparison. Static entries are configured directly on the page upon completing the wizard. Dynamic list entries can be configured directly within the wizard or directly upon the page upon completing the wizard. Click the lightning bolt icon to access the user interface for dynamic menu configuration. Configuration of dynamic list entries is discussed in detail in DataAssist Dynamic Menu.
- **Check box**
- **Radio group**
- **Password field**
- **Text**: Displays the current contents of the record for reference without allowing the content to be updated.

Insert page options

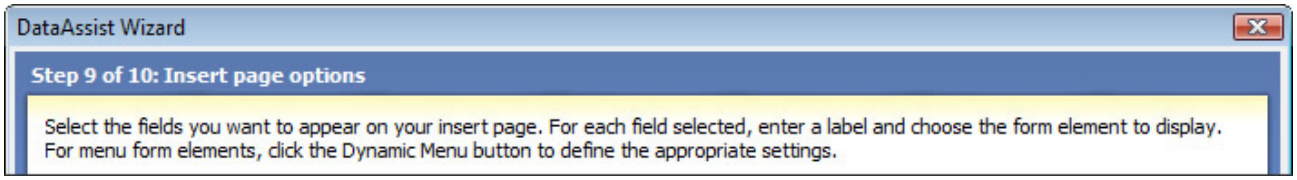
Insert pages contain the required form content to populate a record in your datasource, allowing new records to be created and submitted to the database. Insert pages are linked to on Results pages created through the wizard. They contain an Insert button to perform the record insert, and a Cancel button to return the user to the results page.

To configure your insert page, specify all the database columns to be updated. For each column updated, define the form element's type and label. If a column requires input specified by your application and not by the end-user submitting the form, specify a hidden form element to pass the value, or display that value as text. Upon completing the wizard, you can edit the insert page directly in Dreamweaver, supplying the necessary content to populate the column, either statically or through dynamic datasources configurable through Dreamweaver.

Dynamic lists are available for configuration as form elements to insert data into a column with a selection from a list of values returned from your datasource. See DataAssist Dynamic Menu for more information specific to configuration of Dynamic Lists in the DataAssist wizard for Update pages, as well as Search and Insert pages.

Dreamweaver recordsets may be added to the page specific to populating any dynamic lists added to the form. As well, the Insert Record server behavior is applied to the page to perform the insertion of the record in the datasource. This server behavior is also available for use outside of the wizard, allowing you to add single record insert functionality to pages specific to your needs.

Note: The DataAssist Wizard creates insert pages for single records. The Insert Multiple Records server behavior can be used outside of the DataAssist wizard in conjunction with the Repeat Selection server behavior to insert multiple records to a datasource at one time. See Managing Multiple Records for more information.



Insert fields

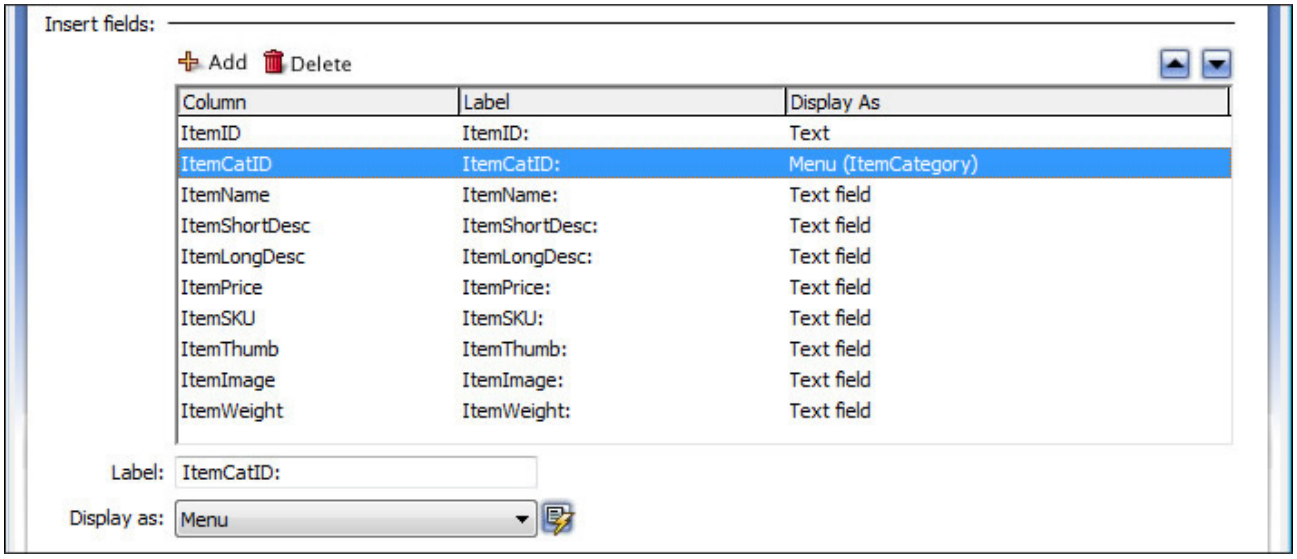
Select the datasource columns to be inserted into in the new record. Bind values for all columns that require a value in the database for the record to be created. This includes database columns that do not have a default value specified when one is not supplied and also require a value.

Data for the new record should be supplied by the end user through an available form element for a specified column, or the data inserted can be set statically as text or passed from a hidden form element.

Define the labels for all data displayed on the page that is to be inserted corresponding to each column populated for the new record.

By default, all columns for the current datasource are configured for inclusion in the update fields control pane.

- Add : Click to add additional columns from the datasource to receive data.
- Delete : Click to remove a column to be inserted into from the form. Datasource columns that are autonumbered by the database should be removed.
- : Form elements and text are listed vertically, one per row, in the insert form. Use the Up and Down buttons to change a selected column's position within the display hierarchy.



To modify a field, select it within the display pane. Once selected, you can modify the label and the type of form element used for the insert. As an alternative, set the column to be displayed as text if you wish to display a static value to be inserted that the end-user will not be able to change.

Note: Key columns that use an autonumber field in your database should not be configured to have a value passed to them for insertion. Attempting to pass a value to this column will result in errors in the application.

Label: Sets the label text that modifies a form element, or text displayed on the page that is inserted for a specified column.

Display as: Sets the type of form element used to enter/select new record information. The types of form elements available are:

- **Text field**
- **Text area**
- **Menu**: Menus may contain static or dynamic list information to be used for a comparison. Static entries are configured directly on the page upon completing the wizard. Dynamic list entries can be configured directly within the wizard or directly upon the page upon completing the wizard. Click the lightning bolt icon to access the user interface for dynamic menu configuration. Configuration of dynamic list entries is discussed in detail in DataAssist Dynamic Menu.
- **Hidden field**: Used to pass values within your data application to be inserted into a specified database column. Useful for columns that contain content that should not be available for configuration to the end user inserting the record. The value for this field can be configured directly on the page using Dreamweaver upon completing the wizard.
- **Check box**

- **Radio group**
- **Password field**
- **Text**: Sets placeholder static content to be used to insert data into a specified column. The content can be edited directly on the page upon completion of the wizard, allowing you to specify a static value, or a value from a dynamic datasource.

Delete page options

The delete page displays a given record's data and provides the opportunity to remove the record from the database. They contain a Delete button to perform the deletion of the record, and a Cancel button to return the location they navigated to the page from. Delete pages are linked to on Results and Detail pages created through the wizard.

To configure the delete page, specify all the database columns for the record and the corresponding labels to be displayed on the page that may assist in the review process for the end user, prior to determining if they wish to delete the record or not.

The Delete Record server behavior is applied to the page to perform the deletion of the record from the datasource. This server behavior is also available for use outside of the wizard, allowing you to add single record deletion functionality to pages specific to your needs.

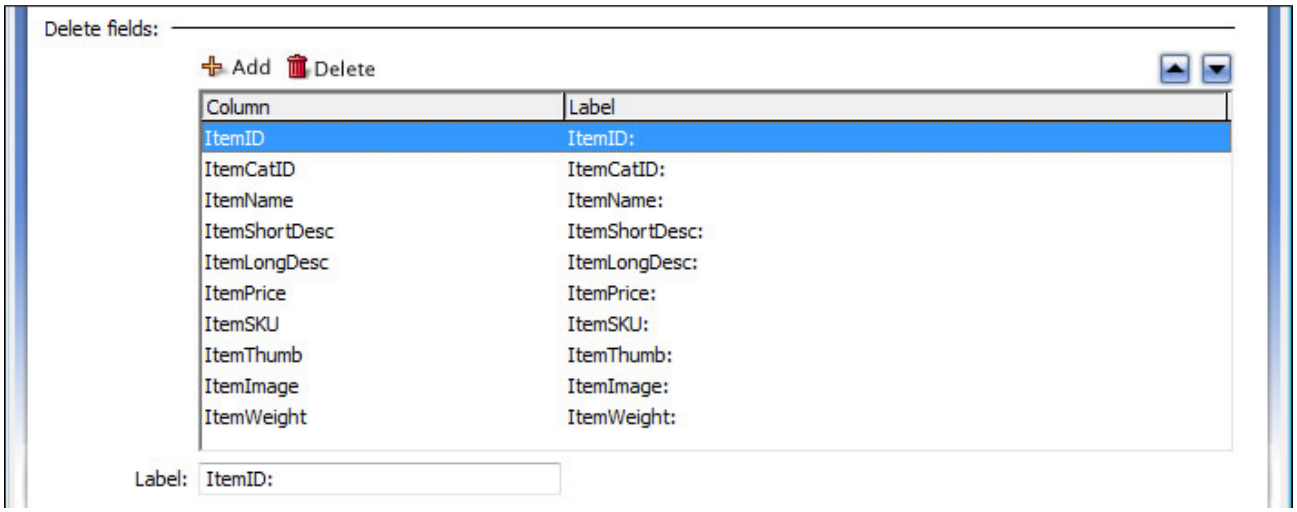
Note: The DataAssist Wizard creates delete pages for single records. The Delete Multiple Records server behavior can be used outside of the DataAssist wizard in conjunction with the Repeat Selection server behavior to delete multiple records from a datasource at one time. See Managing Multiple Records for more information.

Delete fields

Define the record columns and corresponding labels to be displayed on the page for the record to be deleted.

By default, all columns for the current datasource are configured for inclusion in the *Delete fields* control pane.

- Add : Click to add additional columns from the datasource to be displayed.
- Delete : Click to remove a selected column from the display.
- : Labels and column data are listed vertically, one per row, in the body of the delete page. Use the Up and Down buttons to change a selected column's position within the display hierarchy.



Label: Sets the label text that identifies a database column in the current record displayed.

DataAssist Dynamic Menu

Form pages created using the DataAssist Wizard provide the opportunity to use dynamic lists as an option for populating a value into a form field. This includes menus configured for Search, Insert, and Update pages created in the wizard.

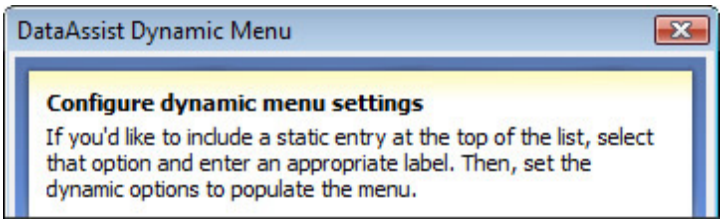
This dynamic menu interface is accessible through the  icon. When configuring these page types and the forms they contain, this icon is active when *Menu* is selected for the **Display as:** option.

Any table accessible in the datasource specified in the Specify database options and page choices step of the wizard is accessible to configure and populate your menu in the DataAssist Wizard. If the datasource necessary to retrieve your dynamic list is distinct from the datasource specified in the wizard, it is necessary to configure your dynamic menus directly on the pages created, as that record content is not accessible for configuration.

Configure whether you wish to have a static option at the top of the list, and specify the table that contains your data, and the columns in that table that correspond to the labels displayed in the list, and the values that correspond to them. The *Labels* correspond to what the end user sees in the menu. The *Values* correspond to the data the form submits when the corresponding label is selected in the menu.

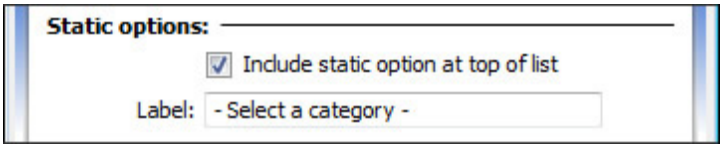
The order of the labels in the list is determined by sorting on any column in the table, either ascending or descending.

Upon completing the wizard, the menu can be updated through the Dynamic List/Menu server behavior applied to the page specific to the given menu, or by clicking on the menu in Design View and configuring it through the Property Inspector.



Static options

Places an additional item at the top of the list. Used to provide direction, or add an additional value not returned by the specified datasource. No value is associated to this selection though this interface, but it can be updated to have a value directly upon completing the wizard. As well, additional static values can similarly be added to the list upon completion of the wizard.



Include static option at top of list: Enables adding a static value to the list.

Label: Specifies the text displayed in the menu for the static option.

Dynamic options

Configures the datasource used to populate your menu by establishing the table it is located in, the location of the labels and values within the table, and the column used to sort the items in the menu.

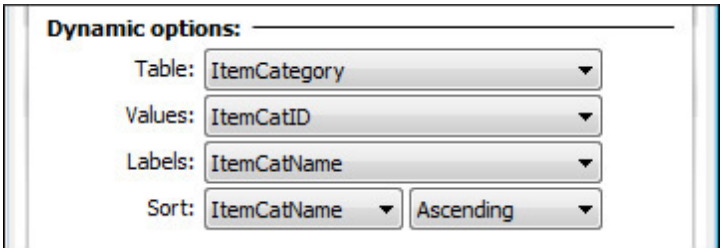


Table: The table within your datasource that contains the information needed to populate the menu.

Values: The column within your table that contains the values used when updating a record, inserting a record, or making a search comparison against a record.

Labels: The column within your table that provides the label displayed to the end user in the menu that corresponds to a given value.

Sort: Specifies the column used to determine the order that items in the list are presented. Values for the selected column can be ordered either ascending or descending.

DataAssist Search Wizard

The DataAssist Search Wizard creates a customized search page to gather criteria to make database comparisons. The current page open in Dreamweaver is the page search results are passed to.

The wizard uses a specified available recordset on the current page to make comparisons against the database based on the search page criteria. It then configures the search page and the search form it contains.

Search page configuration is flexible and can accommodate creating a brand new page and form, create a new form within an existing page, or leverage an existing page and form. When creating a new form in an existing page that uses a template, you have the option of specifying the location the form is inserted within from a list of available editable regions on the page.

Once the high level page configuration is completed, the wizard steps you through the process of configuring each search comparison specific to your search form requirements. This includes associating form elements to comparison types, and determining the order of the comparisons within the query, as well as configuring the individual comparison details.

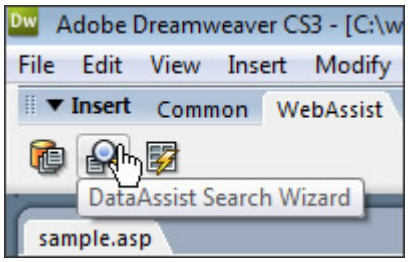
Upon completing the wizard, the search page is generated and opened, so that you can continue to customize it using Dreamweaver. Depending on the options selected, your existing search page is updated or a new page is created. The DataAssist Search server behavior is applied to the page you currently have open, and receives search results based on the search criteria passed to the current page from the search page.

Note: Additional configuration is necessary to display record results on the current page using the recordset filtered by the search. Dreamweaver's Repeat Region server behavior can loop through records for simple display in vertical rows. DataAssist's Repeating Table feature displays information in layouts that supports multiple records per row, as well as CSS attributes for alternating display rows.

Access

The following locations in Dreamweaver open the DataAssist Search Wizard:

- WebAssist Insert panel



- *Insert > WebAssist > DataAssist > DataAssist Search Wizard*

Configuration details

The following pages in this section detail the configuration options available in the wizard:

- Step 1: General configuration

  select the recordset and the type of database that is queried.

- Step 2: Search page selection

  Configure the attributes of the search page, the form it contains, and the location of the form if within a template.

- Step 3: Define your database search

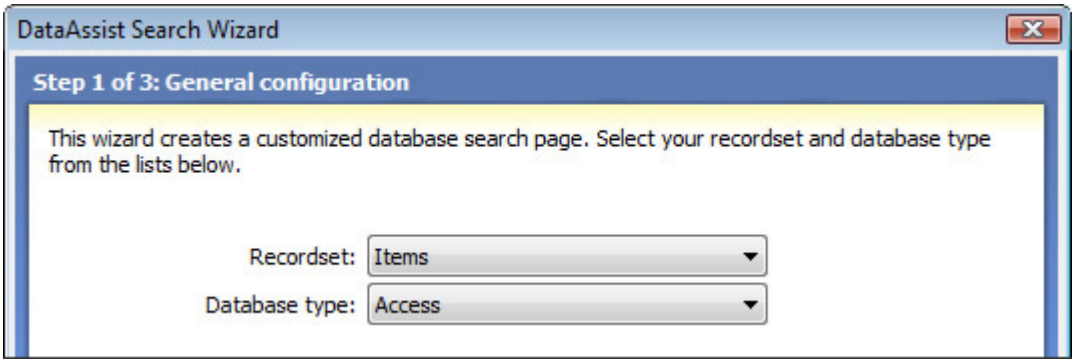
  Configure your database comparisons and associated form elements used in your search form. The following links detail the comparison types available in Step 3.

  - Advanced text search: compares entered text against specified database column(s) to see if it includes the value.
  - Date range search: compares a specified date against a minimum date and maximum date.
  - Price range search: compares a numeric value against a numeric or currency database column.
  - Numeric range search: compares a numeric value against a numeric database column.
  - Exact match search: compares several datatypes to see if a database column contains an exact match for a specified value.
  - Multiple select list or checkboxes search: configures a list or series of checkboxes to pass multiple values to make an OR comparison against a database column.
  - Checkbox search: compares the values associated to the checked and unchecked states of a checkbox to a database column.

Step 1: General Configuration

The General configuration page is the first step in configuring a query using the DataAssist Search Wizard.

Specify the recordset to be queried, and the type of database its retrieved from.



Recordset: Prior to using the wizard, a recordset pointing to the database to be queried must be placed on the page. All recordsets on the page are available for selection from the Recordset list.

Database type: The type of database being used must be specified in this dropdown list. Available options are:

- Access
- SQL Server
- ORACLE
- mySQL

Step 2: Search page selection

The second step in the DataAssist Search Wizard is defining the high level characteristics of the search page. You have the option of selecting from an existing search page, or creating a new page. If desired, you can even specify the current page as the search page, triggering the query and the returning the results on the same page. This is useful when facilitating search on a page used to display results.

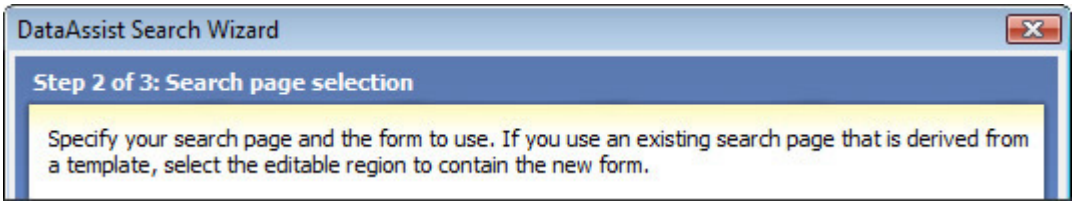
If you use an existing search page, you have the option of using an existing search form, or creating a new search form that is identified in the source code by the name you specify. If creating a new search page, you also have the option of controlling the naming convention of the form.

If you are creating a new form, the form elements are specified during the Select your search element to add step when configuring search parameters.

If using an existing search page and creating a new form, and that page has a template applied to it, select the appropriate editable region to insert the new search form within.

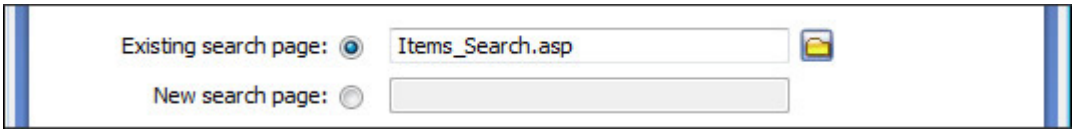
Once you have finished the DataAssist Search Wizard, additions are made to the search page based on the criteria specified in this step:

- The form that is either specified or created contains a form action set to the page you had open upon initiating the wizard.
- If a new form is created, the name of the form is used to correlate it to the DataAssist Search server behavior applied on the current page.
- If an existing form is referenced, a hidden form element is added to the form to correlate it to the particular DataAssist Search server behavior application on the current page.



Search page

Specify the search page to be used or created by the wizard.



Existing search page: Selects an existing page where search criteria is specified. The wizard retrieves the form name and form elements from the specified search page for configuration in the next step of the wizard, Define your database search.

New search page: The new page name and location are specified here to generate this page upon completion of the wizard. Configuration of the search form and form elements for this new page is in the next step of the wizard, Define your database search.

Search form

Specify a new search form if creating a new search page. Specify an existing form or create a new form if using an existing search page.

Existing form: If you specified an existing search page, select an existing form from that page from the list of options. This option will not be available if you generate a new search page through the Object.

New form: If you are generating a new search page or wish to create a new form in an existing search page, specify the name of the form. Do not use the name of a form that already present on the page if using an existing page.

Template location

If you specified an existing search page with a template applied to it, this selects the editable region in which to place the form on the page.

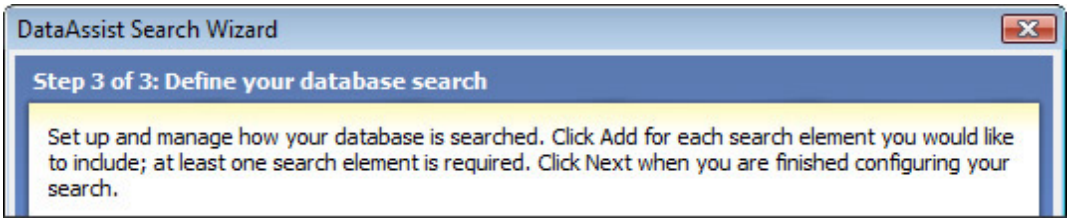
Step 3: Define your database search

The third step of the wizard is building your DataAssist Search criteria by configuring your search parameters and how they are compared against the information in your database.

Add different types of search comparisons, configure the form fields used to capture the associated comparison values, and specify the database column(s) that the comparison are made against.

Use the *Search element control* pane to manage comparisons, the order in which they appear in the query statement, and to initiate the configuration of the details specific to each. Available comparison types, and links to detailed information specific to each, is available below in the section Configuring comparisons.

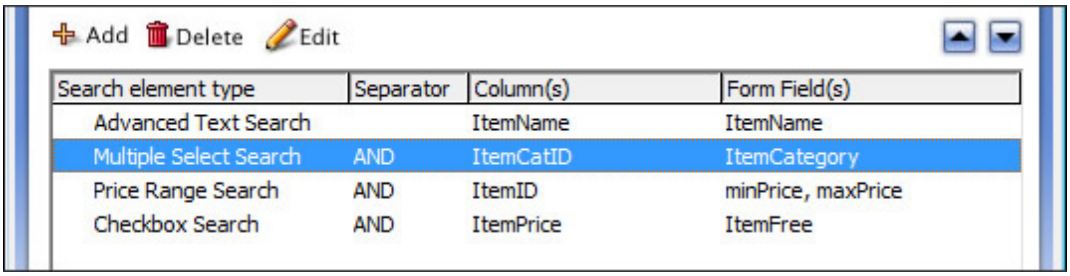
Note: The comparisons configured using this wizard are designed for standard implementations. Advanced search configurations can be achieved by either editing the application of the DataAssist Search server behavior applied to your results page by this wizard, or by applying this server behavior directly to your results page.



Search element control

The wizard creates complex query statements based on the configuration of individual comparisons against data captured in a form. Each comparison is listed and managed in this control.

- Add : Comparisons are added by clicking the *Add* (+) button. This initiates a step by step process for configuring individual comparisons (see Configuring comparisons below).
- Delete : Comparisons are removed by selecting an existing comparison and clicking the *Delete* button.
- Edit : The Edit button triggers the edit process for an existing comparison that is selected in the list.
- : Statements are ordered by selecting them in the list and using the up and down arrows to change their relative position.



Separator

For a selected search element type, this determines how it relates to the comparison listed prior to it in the hierarchy.

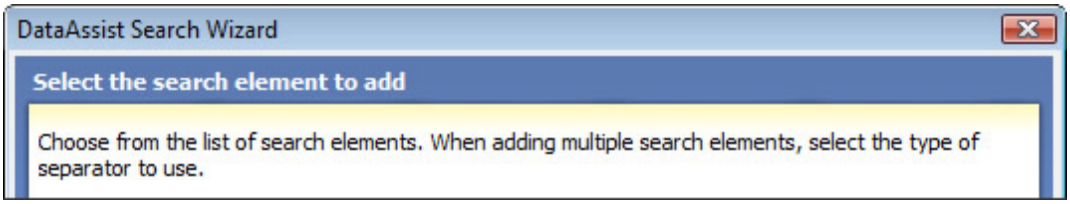
- **AND**: requires that both the prior comparison listed sequentially AND the comparison currently under configuration are satisfied by a database record for it to be returned as a result.
- **OR**: requires that either the prior comparison listed sequentially OR the comparison currently under configuration are satisfied by a database record for it to be returned as a result.

Configuring comparisons

Clicking the *Add* (+) button above the *Search element control pane* initiates a series of steps to define specific comparison types for your query.

The following details the first step, and the pages that follow details the steps specific to each comparison type available.

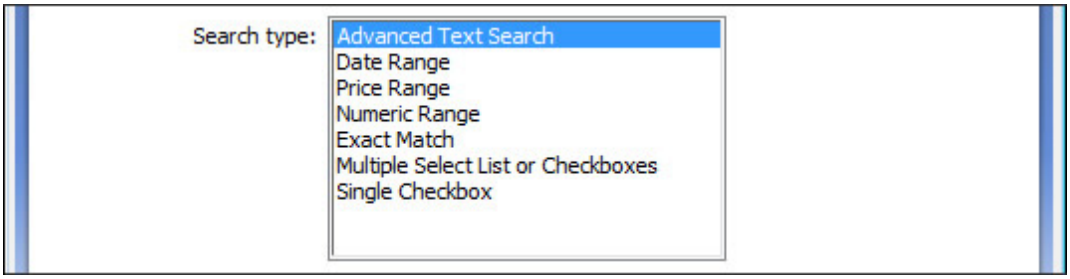
Select a comparison type, and click the *Next* button to go to the step specific to configuring the selected comparison. Upon completing a comparison configuration, click *Next* again to add that comparison to the *Search element control pane*.



Search type

The first step in the process gives you the option of selecting from a list of available comparison types.

Below is a list of each type available, within links to details specific to each.



Advanced Text Search: compares entered text against specified database column(s) to see if it includes the value

Date Range: compares a specified date against a minimum date and maximum date

Price Range: compares a numeric value against a numeric or currency database column

Numeric Range: compares a numeric value against a numeric database column

Exact Match: compares several datatypes to see if a database column contains an exact match for a specified value

Multiple Select List or Checkboxes: configures a list or series of checkboxes to pass multiple values to make an OR comparison against a database column

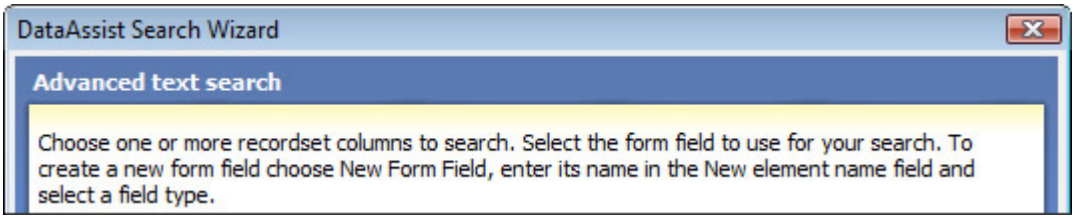
Single Checkbox: compares the values associated to the checked and unchecked states of a checkbox to a database column

Advanced text search

This step configures a text search comparison within the DataAssist Search server behavior based on selecting *Advanced Text Search* when adding a search comparison in step 3 of the DataAssist Search Wizard: Define your DataAssist Search.

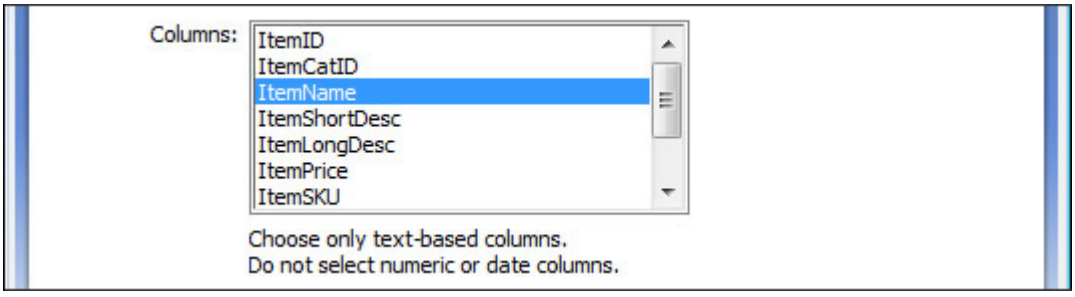
Text-based database columns are searched against using a text comparison specified through a form element. The results returned depend on whether the selected column in a given record includes the text parameter specified.

Note: The comparisons configured using this wizard are designed for standard implementations. Advanced search configurations can be achieved by either editing the application of the DataAssist Search server behavior applied to your results page by this wizard, or by applying this server behavior directly to your results page.



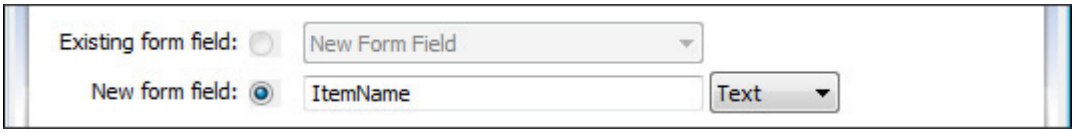
Columns

Select all columns to be searched against using the values passed from the specified form element (CTRL-left click for multiple selections). Columns included must be text-based.



Form fields

Specify the form element used to pass the text parameter necessary to make the comparison. If you are using an existing search form (specified in Step 2 of the wizard, Search page selection), you have the option of either adding a new form element or using an existing form element. If creating a new form, specify a unique name for the form field to identify and label it. Upon completing the wizard, you may use Dreamweaver to customize the search page specific to your needs.



Existing form field: If using an existing search page and form, select if you wish to use an existing form element to make the comparison. All form elements contained in the form specified in step 2 of the wizard are available in the list.

New form field: If creating a new form or creating a new search parameter in an existing form, specify the name of a new form element to add to the search form. Select the type of form element to use. Options are:

- Text
- Select
- Radio
- Check

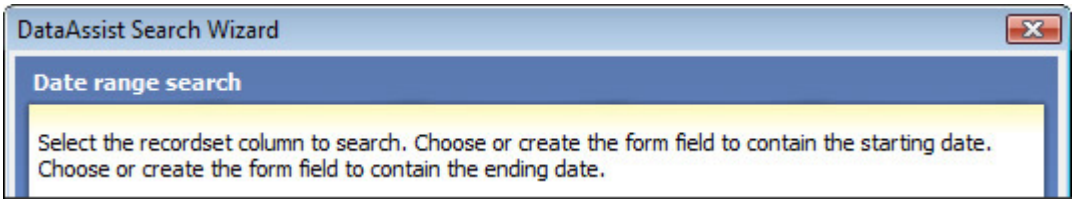
Note: Do not use numbers as the first character when naming a form element. Options for select lists, as well as values associated to list entries, radio buttons, and checkboxes must be configured on the search page using Dreamweaver upon completing the wizard.

Date range search

This step configures a date range search comparison within the DataAssist Search server behavior based on selecting *Date Range* when adding a search comparison in step 3 of the DataAssist Search Wizard: Define your DataAssist Search.

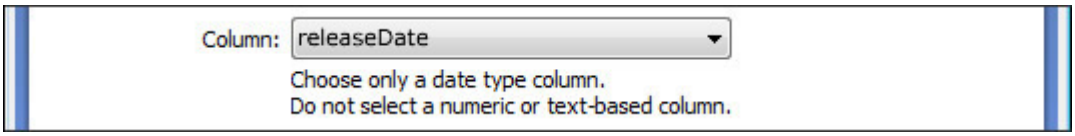
A single date database column is searched against using a time span comparison specified through a minimum date form element and maximum date form element. The results returned depend on whether the selected column in a given record contains a date that falls within the specified span of time.

Note: The comparisons configured using this wizard are designed for standard implementations. Advanced search configurations can be achieved by either editing the application of the DataAssist Search server behavior applied to your results page by this wizard, or by applying this server behavior directly to your results page.



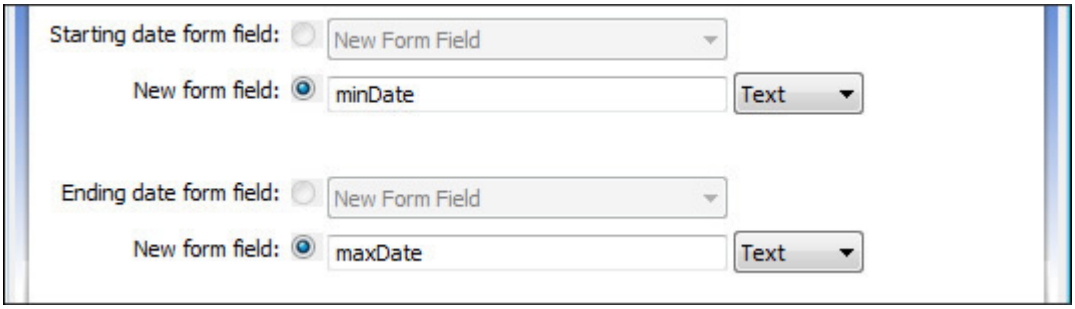
Columns

Select the date column to be searched against using the maximum and minimum values passed from the specified form elements. The column included must use a date format.



Form fields

Specify the two form element used to pass the maximum and minimum date parameters used to make the comparison. If you are using an existing search form (specified in Step 2 of the wizard, Search page selection), you have the option of either adding new form elements or using existing form elements. If creating a new form, specify a unique name for each form field to identify and label it. Upon completing the wizard, you may use Dreamweaver to customize the search page specific to your needs.



Starting date form field: Specify the form element used to pass the minimum date used for the comparison.

Ending date form field: Specify the form element used to pass the maximum date used for the comparison.

For both form fields, select an existing form field if applicable, or specify a name for a new form field to be created. All form elements contained in an existing form specified in step 2 of the wizard are available in the list. If creating a new form field in an existing form or a new form, specify the type of field to be used. Options are:

- Text
- Select
- Radio
- Check

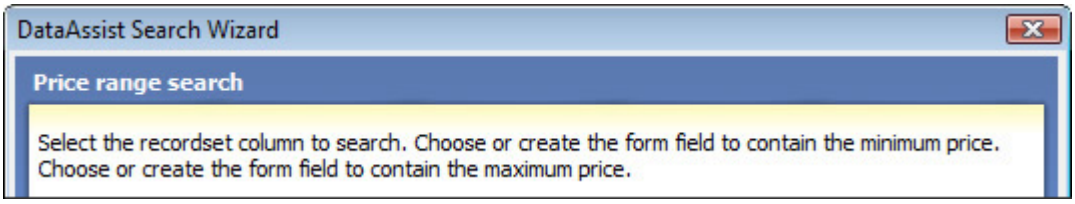
Note: Do not use numbers as the first character when naming a form element. Options for select lists, as well as values associated to list entries, radio buttons, and checkboxes must be configured on the search page using Dreamweaver upon completing the wizard.

Price range search

This step configures a price range search comparison within the DataAssist Search server behavior based on selecting *Price Range* when adding a search comparison in step 3 of the DataAssist Search Wizard: Define your DataAssist Search.

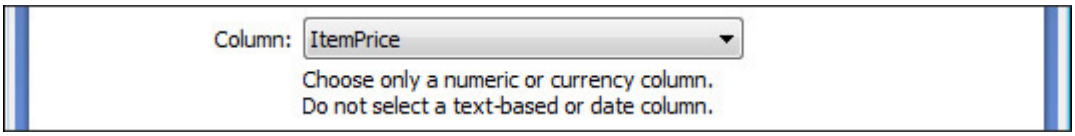
A single numeric or currency database column can be searched against using a numeric span comparison specified through a minimum price form element and maximum price form element. The results returned depend on whether the selected column in a given record contains a value that falls within the specified numeric span.

Note: The comparisons configured using this wizard are designed for standard implementations. Advanced search configurations can be achieved by either editing the application of the DataAssist Search server behavior applied to your results page by this wizard, or by applying this server behavior directly to your results page.



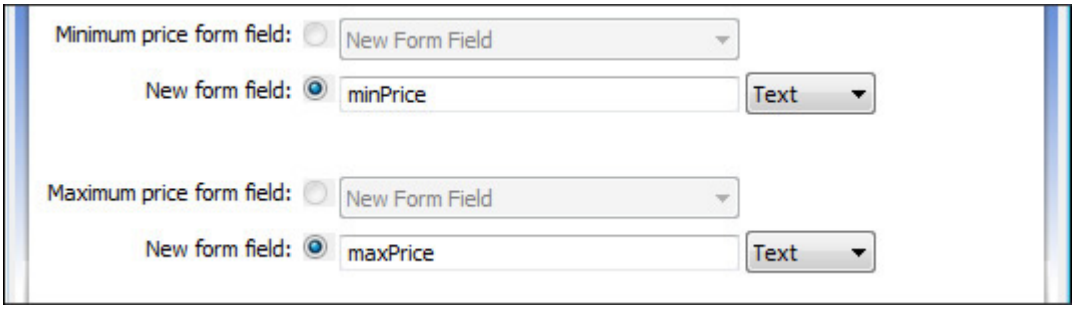
Columns

Select the date column to be searched against using the maximum and minimum values passed from the specified form elements. The column included must use a numeric or currency format.



Form fields

Specify the two form element used to pass the maximum and minimum numeric parameters necessary to make the comparison. If you are using an existing search form (specified in Step 2 of the wizard, Search page selection), you have the option of either adding new form elements or using existing form elements. If creating a new form, specify a unique name for each form field to identify and label it. Upon completing the wizard, you may use Dreamweaver to customize the search page specific to your needs.



Minimum price form field: Specify the form element used to pass the minimum numeric value used for the comparison.

Maximum price form field: Specify the form element used to pass the maximum numeric value used for the comparison.

For both form fields, select an existing form field if applicable, or specify a name for a new form field to be created. All form elements contained in an existing form specified in step 2 of the wizard are available in the list. If creating a new form field in an existing form or a new form, specify the type of field to be used. Options are:

- Text
- Select
- Radio
- Check

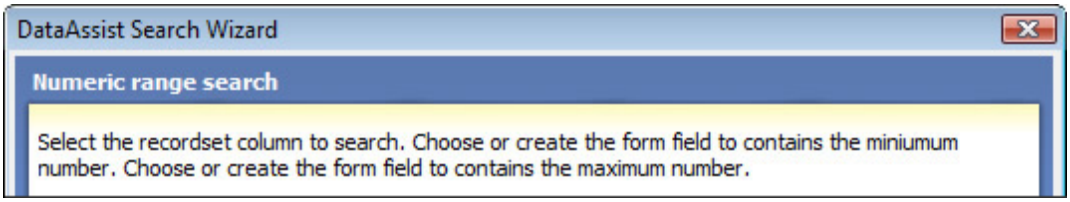
Note: Do not use numbers as the first character when naming a form element. Options for select lists, as well as values associated to list entries, radio buttons, and checkboxes must be configured on the search page using Dreamweaver upon completing the wizard.

Numeric range search

This step configures a numeric range search comparison within the DataAssist Search server behavior based on selecting *Numeric Range* when adding a search comparison in step 3 of the DataAssist Search Wizard: Define your DataAssist Search.

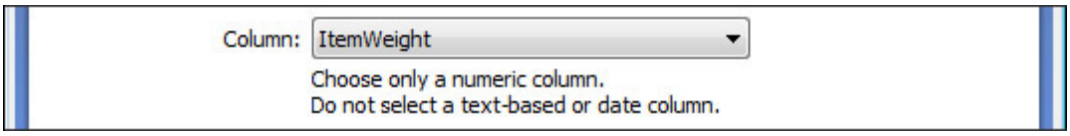
A single numeric database column is searched against using a numeric span comparison specified through a minimum number form element and maximum number form element. The results returned depend on whether the selected column in a given record contains a value that falls within the specified numeric span.

Note: The comparisons configured using this wizard are designed for standard implementations. Advanced search configurations can be achieved by either editing the application of the DataAssist Search server behavior applied to your results page by this wizard, or by applying this server behavior directly to your results page.



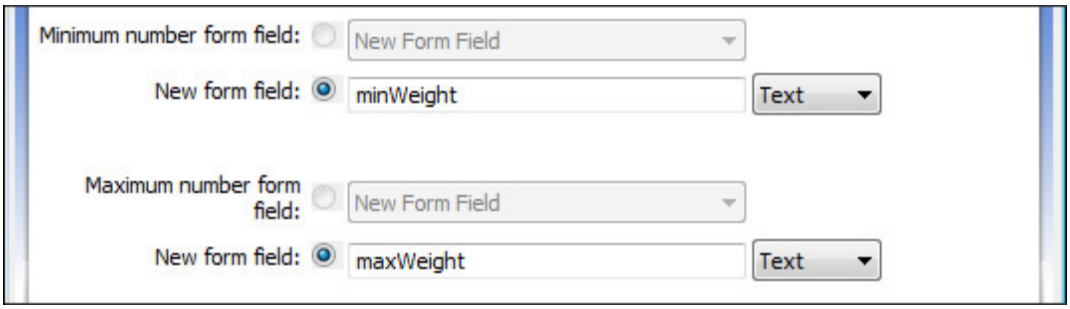
Columns

Select the date column to be searched against using the maximum and minimum values passed from the specified form elements. The column included must use a numeric format.



Form fields

Specify the two form element used to pass the maximum and minimum numeric parameters necessary to make the comparison. If you are using an existing search form (specified in Step 2 of the wizard, Search page selection), you have the option of either adding new form elements or using existing form elements. If creating a new form, specify a unique name for each form field to identify and label it. Upon completing the wizard, you may use Dreamweaver to customize the search page specific to your needs.



Minimum number form field: Specify the form element used to pass the minimum numeric value used for the comparison.

Maximum number form field: Specify the form element used to pass the maximum numeric value used for the comparison.

For both form fields, select an existing form field if applicable, or specify a name for a new form field to be created. All form elements contained in an existing form specified in step 2 of the wizard are available in the list.
If creating a new form field in an existing form or a new form, specify the type of field to be used. Options are:

- Text
- Select
- Radio
- Check

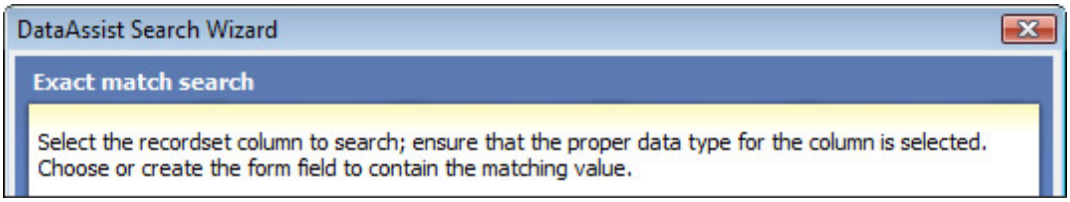
Note: Do not use numbers as the first character when naming a form element. Options for select lists, as well as values associated to list entries, radio buttons, and checkboxes must be configured on the search page using Dreamweaver upon completing the wizard.

Exact match search

This step configures an exact match comparison against either a text, number, boolean or date search column using the DataAssist Search server behavior. This is based on selecting *Exact Match Search* when adding a search comparison in step 3 of the DataAssist Search Wizard: Define your DataAssist Search.

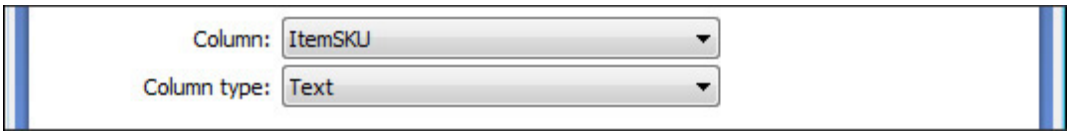
A single database column in either a numeric, bit, text, or date format is searched against using a comparison specified through a form element. Specify the appropriate database column type specific to the type of search parameter being used. The results returned depend on whether the selected column in a given record is an exact match of the parameter passed from the form element.

Note: The comparisons configured using this wizard are designed for standard implementations. Advanced search configurations can be achieved by either editing the application of the DataAssist Search server behavior applied to your results page by this wizard, or by applying this server behavior directly to your results page.



Columns

Specify a column type that matches the content of the database column to be searched against. This ensures that exact match comparisons are made correctly based on the expected column format.



Column: Select the database column to be searched against.

Column type: Select the column type that matches the type of search parameter to be passed from the form element on the search page. Available options are:

- Text
- Number/Boolean: Boolean is true or false value notated as 1 or 0 respectively. Typically captured as a bit field in the database.
- Date

Form fields

Specify the form element used to pass the parameter necessary to make the comparison. If you are using an existing search form (specified in Step 2 of the wizard, Search page selection), you have the option of either adding a new form element or using an existing form element. If creating a new form, specify a unique name for the form field to identify and label it. Upon completing the wizard, you may use Dreamweaver to customize the search page specific to your needs.

Exact match form field: If using an existing search page and form, select if you wish to use an existing form element to make the comparison. All form elements contained in the form specified in step 2 of the wizard are available in the list.

New form field: If creating a new form or creating a new search parameter in an existing form, specify the name of a new form element to add to the search form. Select the type of form element to use. Options are:

- Text
- Select
- Radio
- Check

Note: Do not use numbers as the first character when naming a form element. Options for select lists, as well as values associated to list entries, radio buttons, and checkboxes must be configured on the search page using Dreamweaver upon completing the wizard.

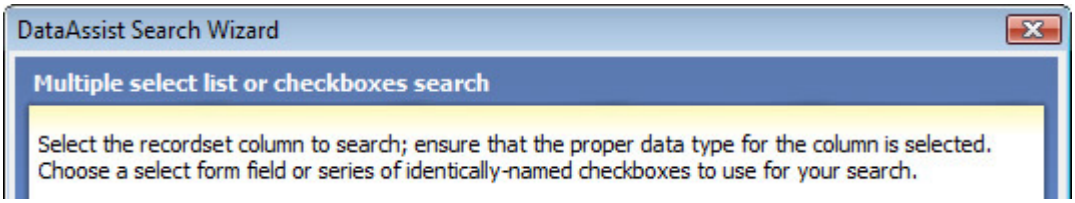
Multiple select list or checkboxes search

This step configures a comparison against either a text, number, boolean or date search column using the DataAssist Search server behavior. This is based on selecting *Multiple select list or checkboxes* when adding a search comparison in step 3 of the DataAssist Search Wizard: Define your DataAssist Search.

A single database column in either a numeric, bit, text, or date format is searched against. The value of the parameter used to make the comparison is determined by either selections in a specified multiple select list, or selections from a group of identically named checkboxes. Multiple selections pass multiple values for comparison, using an OR comparison that returns a record if any of the values exist.

Specify the appropriate database column type specific to the type of search parameter being used. The results returned depend on whether the selected column in a given record is an exact match of the parameter passed from the form element.

Note: The comparisons configured using this wizard are designed for standard implementations. Advanced search configurations can be achieved by either editing the application of the DataAssist Search server behavior applied to your results page by this wizard, or by applying this server behavior directly to your results page.



Columns

Specify a column type that matches the content of the database column to be searched against. This ensures that comparisons are made correctly based on the expected column format.

Column: Select the database column to be searched against.

Column type: Select the column type that matches the type of search parameter to be passed from the form element(s) on the search page. Available options are:

- Text
- Number/Boolean: Boolean is true or false value notated as 1 or 0 respectively. Typically captured as a bit field in the database.
- Date

Form fields

Specify the form element used to pass the parameter necessary to make the comparison. If you are using an existing search form (specified in Step 2 of the wizard, Search page selection), you have the option of either adding a new form element or using an existing form element. If creating a new form, specify a unique name for the form field to identify and label it. Upon completing the wizard, you may use Dreamweaver to customize the search page specific to your needs.

Multiple select form field: Specify the multiple select list or the name of the checkbox used to make the comparison. If using an existing search page and form, select if you wish to use an existing form element to make the comparison. All form elements contained in the form specified in step 2 of the wizard are available in the list. If performing a comparison against a series of checkboxes, they must be identically named

New form field: If creating a new form or creating a new search parameter in an existing form, specify the name of a new form element to add to the search form. Specify whether to use a Select or Check form element. When specifying a single check form element, only a single checkbox is added to the page. Upon completing the wizard, it is necessary to add additional checkboxes with the same name, but with different values to be passed as search comparisons. Be sure to label each checkbox to identify it appropriately to the end user.

Note: Do not use numbers as the first character when naming a form element. Options for select lists, as well as values associated to list entries, radio buttons, and checkboxes must be configured on the search page using Dreamweaver upon completing the wizard.

Checkbox search

This step configures a comparison against either a text, number, boolean or date search column using the DataAssist Search server behavior. This is based on selecting *Single checkbox* when adding a search comparison in step 3 of the DataAssist Search Wizard: Define your DataAssist Search.

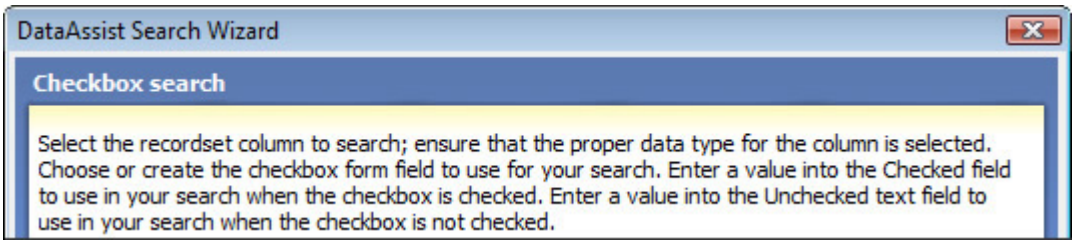
A single database column in either a numeric, bit, text, or date format is searched against. The value of the parameter used to make the comparison is determined by the status of the checkbox: checked or unchecked. You have the option of specifying the values for these two states either statically, or by retrieving the parameter values from a dynamic data source.

The type of comparison made is also configured specific to the state of the checkbox. You can check to see if the value in the database equals a given value, or does not equal a given value, specific to the parameter specified for each state.

This provides flexibility in how you approach using a checkbox search. You can configure the comparison to search using the same parameter for both checkbox states, allowing end user control over whether they want records returned containing that value or not. In contrast, you can also use distinct values for the states of the checkbox, allowing users to choose to return records based on two different value comparisons.

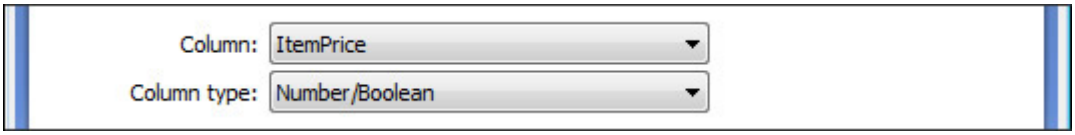
Specify the appropriate database column type for the type of search parameter being used. The results returned depend on whether the selected column in a given record is an exact match of the parameter passed from the checkbox.

Note: The comparisons configured using this wizard are designed for standard implementations. Advanced search configurations can be achieved by either editing the application of the DataAssist Search server behavior applied to your results page by this wizard, or by applying this server behavior directly to your results page.



Columns

Specify a column type that matches the content of the database column to be searched against. This ensures that comparisons are made correctly based on the expected column format.

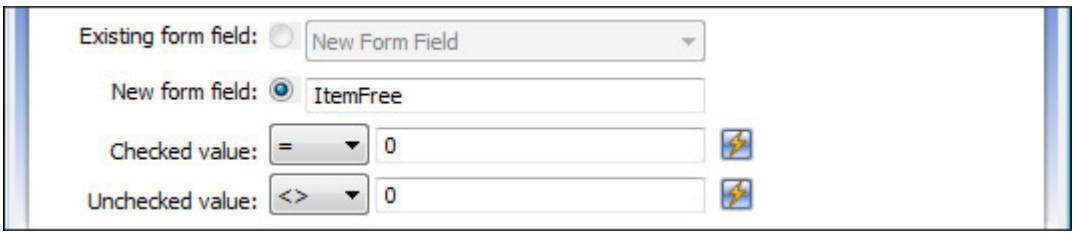


Column: Select the database column to be searched against.

Column type: Select the column type that matches the type of search parameter to be passed from the form element(s) on the search page. Available options are:

- Text
- Number/Boolean: Boolean is true or false value notated as 1 or 0 respectively. Typically captured as a bit field in the database.
- Date

Form fields



Existing form field: If using an existing search page and form, select if you wish to use an existing form element to make the comparison. All form elements contained in the form specified in step 2 of the wizard are available in the list, so be sure the element selected is a checkbox.

New form field: If creating a new form or creating a new search parameter in an existing form, specify the name used to identify the checkbox in the form.

Note: Do not use numbers as the first character when naming a form element.

Checked value: Specify the comparison made against the database column when the checkbox is checked. You can check to see if a specified value exists or does not exist. The value can be specified either statically or using available dynamic data sources.

Unchecked value: Specify the comparison made against the database column when the checkbox is unchecked. You can check to see if a specified value exists or does not exist. The value can be specified either statically or using available dynamic data sources.

DataAssist Search server behavior

The DataAssist Search server behavior returns data to the page it is applied to based on comparisons made against a database from a recordset available on the page.

The search is initiated by a trigger on the page. For applications of the server behavior performed by the DataAssist Wizard and the DataAssist Search Wizard, the trigger is typically a form request variable passed from a search form.

Common applications of the server behavior through the aforementioned wizards rely on criteria passed from a search form filled out by the end user. These implementations can be updated directly through the server behavior outside of the wizard as well. This allows you to to customize the application of the search page/results page relationship. If necessary, you may change your search page and comparison configurations through the DataAssist Search Wizard by clicking the *Open Wizard* button at the bottom of this interface.

As well, the server behavior can be applied directly to a page to perform query functions specific to your search needs. Dynamic data bindings for values that trigger the search, as well as for supplying comparison criteria against specified database columns, make this server behavior flexible for advanced query implementations.

It is recommended that if you create a search page and form using the DataAssist Search server behavior directly instead of the wizard, your search page is configured with all necessary form elements so they are accessible through the bindings interfaces during the configuration process.

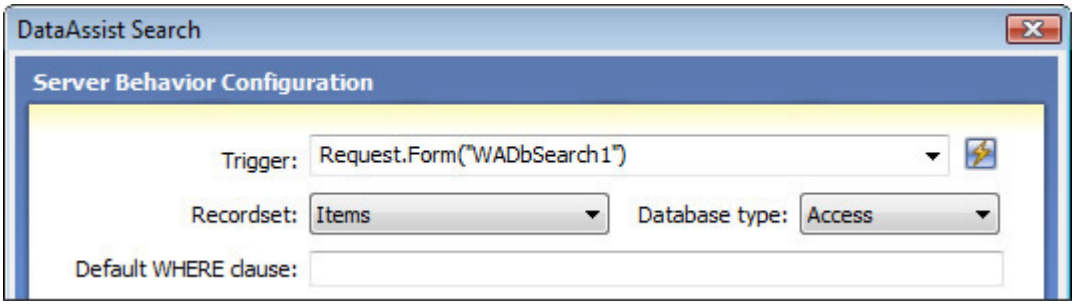
Access

The DataAssist Search server behavior is accessible in the following location:

- *Server Behaviors panel > Plus icon (+) > DataAssist > DataAssist Search*

Server Behavior Configuration

Specify the high level configuration options that determine the datasource, the event initiating the query, and the default parameters to use on the page if the page is accessed directly, or from a location that is not passing it the appropriate search parameters.



Trigger: Refers to a request, session variable, or other dynamic data event. The existence of a value for the specified variable triggers the query on a given page. A request object (such as a hidden form element) or a session variable must be submitted or passed to the page that has the server behavior applied to it in order for the query to execute. Available triggers are:

- **Any Form Post**: Any form posted to the current page that the server behavior is applied to will trigger the data retrieval.
- **Before Page Load**: The data retrieval is triggered when the current page that the server behavior is applied to is loaded.
- **On Form Submit**: Any form submitted on the same page that the server behavior is applied to will trigger the data retrieval.
- **Button [button name] pressed**: The server behavior is triggered if the specified button on the page is clicked.
- **Dynamic Data**: Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.

Note: If you used the DataAssist Search Wizard to apply the server behavior, this value is equal to the name of a hidden form element passed by your search form that was inserted by the wizard. It is typically named WADbSearch1.

Recordset: the recordset on the page used to query the database. Select from all recordsets currently applied to the page.

Database type: Identifies the database used. This properly correlates datatypes and syntax used in the code associated to the query.

Default WHERE clause: The default where clause used if the session variable is not set and the user has accessed the current page without passing the appropriate trigger to run the server behavior. Here are some examples for returning all or no records:

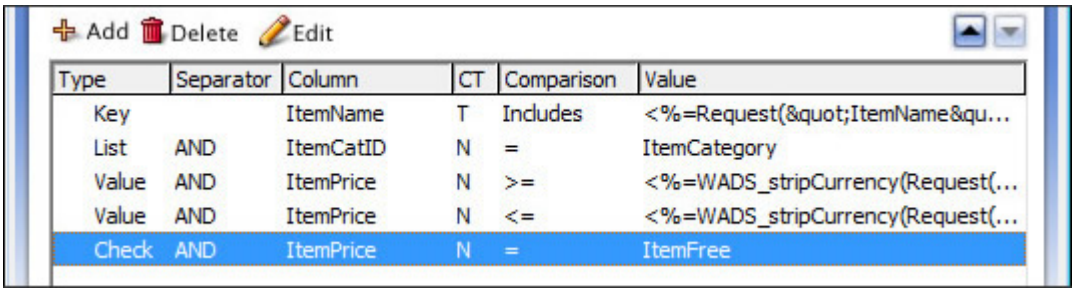
To default to no records displayed:
WHERE 0 <> 0

To default to all records displayed:
WHERE 0 = 0

Search Parameter control

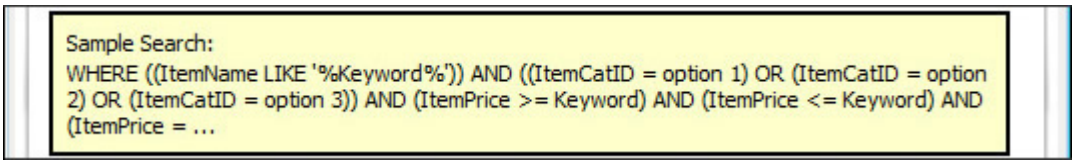
Configure individual comparisons based on a value passed to the page from a specified location. Each comparison is listed and managed in this control.

- Add : Comparisons are added by clicking the *Add* (+) button. This initiates a the configuration user interface for a new comparison to be added to the list (see Configure a Search Parameter below).
- Delete : Comparisons are removed by selecting an existing comparison and clicking the *Delete* button.
- Edit : The *Edit* button triggers the edit process for an existing comparison that is selected in the list (see Configure a Search Parameter below).
- : Statements are ordered by selecting them in the list and using the up and down arrows to change their relative position.



Sample Search

Based on the comparison specified, an example of the resulting WHERE clause used to make the database query on the page is displayed for reference.

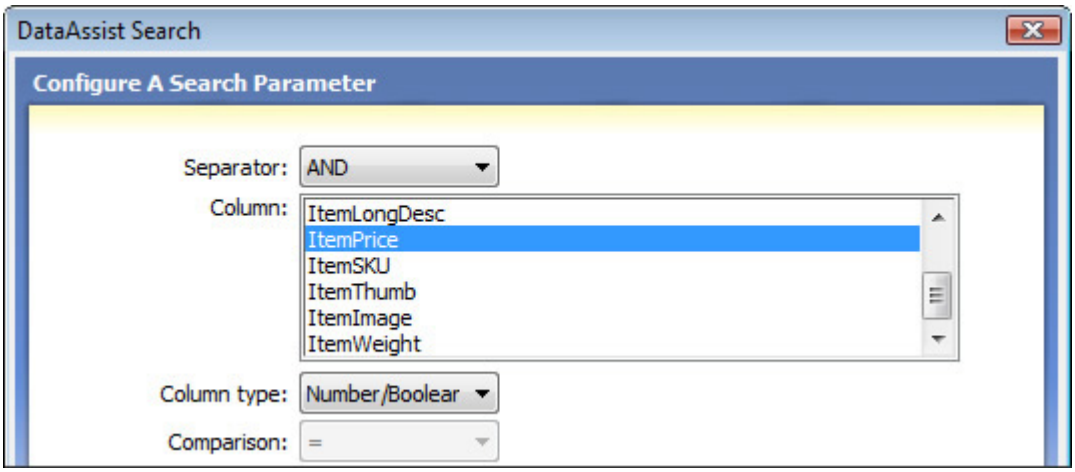


Configure a Search Parameter

Clicking the *Add* (+) button above the *Search parameter control pane* initiates the advanced configuration interface available to define specific comparison types for your query. The following details all of the criteria required to appropriately configure a given comparison.

General parameter configuration

Each comparison is comprised of the four following high level attributes:



Separator: compares multiple entry types for queries and provides filtering for records based on their entry type: "AND", "OR", "AND NOT", and "OR NOT". You will notice the first entry you make does not use a separator as it has nothing to compare to.

Column: selects the recordset column that the Entry Value will be compared to. Multiple columns selections are available only for Keyword entry values.

Column Type: Select the column type that matches the type of search parameter to be passed to the page. The server behavior attempts to identify it for you. Always check to ensure that it is correct before adding the comparison to the list of comparisons. Available options are:

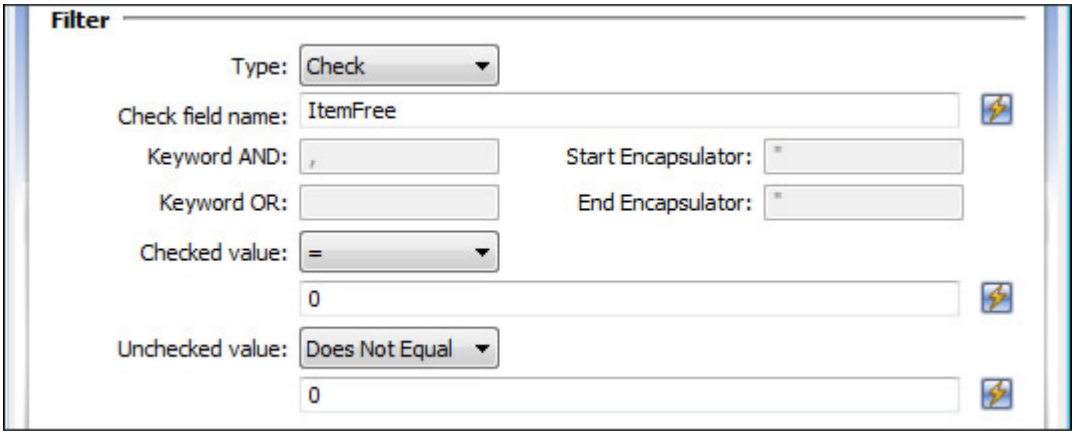
- Text
- Number/Boolean: Boolean is true or false value notated as 1 or 0 respectively. Typically captured as a bit field in the database.
- Date

Comparison: Selects the comparison to use between the column and the value in the query statement. The options and the Entry types they are available for are as follows:

- **Equals | Does not Equal**: used to compare values through exact comparisons
- **Greater than | Less Than | Greater than or equal to | Less than or equal to**: used for comparison against numerical and date database columns
- **Includes | Begins With | Ends With**: Used to find exact values anywhere within a value from the database using wildcards. For Keyword filters:
  - *Includes*: looks for specified text anywhere within a database field value.
  - *Begins With:* returns results that have specified text only at the beginning of the value in the database.
  - *Ends With:* returns results that have specified text only at the end of the value in the database.
- **Is | Is NOT**: used to check whether a value exists for the specified column.
  - *Is*: checks to make sure that a value exists in the column.
  - *Is NOT*: looks for a null value in the specified column

Filter

This section configures the attributes of the type of comparison you want to perform. Depending on the type of attribute selected, some options may be disabled if not applicable.



Type: Refers to the type of comparison you would like to perform:

- **Value**: Value entries are for comparing to constant values or server-side variables. Value comparisons allow for direct entry of server-side code to retrieve the value dynamically on the server.
- **Edit**: Edit comparisons are used to get values from edit boxes on the search page. These can be type text, check, radio button, button, etc...
- **List**: List comparisons are reserved for dropdown and multi-select lists on the search page.
- **Check**: This comparison presents you with two extra options, *Checked value* and *Unchecked value*, for which specific comparisons need to be configured for each. These refer to the actual strings you want to compare the column to if the checkbox is checked or unchecked respectively

  Note: There are issues in Coldfusion and PHP regarding unchecked values; please refer to Advanced configuration for more info.
- **Keyword**: This comparison also presents you with two extra options. It is used for edit boxes where users can do keyword searches. The two extra options, Implied And and Implied Or, refer to what the user will enter for AND and OR in the keyword field. The defaults are " , " and " " respectively.
  e.g. Entering "Fruit, Apple Vegetable" returns records for columns with both "Fruit" and "Apple" or just "Vegetable"

Value: Specifies a static value, variable, or form element on the page, whose value is compared to the selected database column. The dynamic attribute button allows you to specify this value based on available dynamic data sources.

Note: When using date columns with the "less than or equal to" or the "greater than" comparisons unexpected results may occur. Please refer to the Date Evaluation information in the Advanced configuration section.

Keyword AND: sets the character accepted as a separator within a search field for AND comparisons
e.g. for a comma: *apples, oranges* equates to *apples AND oranges*

Keyword OR: Sets the character accepted as a separator within a search field for OR comparisons
e.g. for a comma*: apples, oranges* equates to *apples OR oranges*

Start Encapsulator and End Encapsulator: Sets the characters that are accepted by the textfield that can be entered by the end user to encompass a string for comparison against search criteria, forcing the requirement that the entire string be matched.

Advanced configuration

The following sections details advanced scenarios in using the DataAssist Search server behavior that require additional configuration.

- Date Part Evaluations
- TimeStamp Evaluations
- Multiple applications of DataAssist Search server behavior to the same recordset
- Applying DataAssist Search server behavior to a recordset that contains an existing WHERE clause
- Issues in Coldfusion referencing #form.checkbox# and in PHP referencing <?php echo $_POST["checkbox"]; ?>

Date Part Evaluations

When making comparisons against date columns, DataAssist Search server behavior configures date comparisons using the complete mm/dd/yyyy format. The user interface only provides a list of available columns to select when specifying the location in the recordset that the date value comparison is made against. (see DataAssist Search server behavior for information on adding comparisons using the Plus button). When selecting a column from the list to retrieve the date value, the date returned from the database is in the mm/dd/yyyy format and therefore, the value compared against it must also be in the same format. Partial comparisons specifying only a month or a year will not work in this scenario.

It is possible to make comparisons of partial date information (month, year, etc.) by using database syntax to evaluate the value returned from the date column. This requires manually entering an evaluation string in the column for the specified comparison, as well as changing the comparison type to either Number or Text, depending on the format in which the value is returned by the date evaluation string.

For example, the following comparison is typical of date evaluations created using the Plus button:

Type	Separator	Column	CT	Comparison	Value
E		UserBDate	D	=	Bdate

This configuration has a form element called *Bdate* compared against the value found in the database column *UserBDate*. The comparison type is D for date to correlate with the database column's type. This format would require that the value retrieved from *BDate* is in the mm/dd/yyyy format.

However, to only compare against specific parts of the full date retrieved from the column, an expression can be manually entered in the column field which evaluates the column and returns only the value you wish to compare against.

For example, the figure below demonstrates the DatePart column evaluation available in Access database, which evaluates the column *UserBDate* and returns 'm', which returns a number for the month. As a number value is being returned, the comparison type (CT) needs to be set to N (number). The list *Bdate* contains all months and their corresponding values (January=1, February=2, etc.). Selecting a month from this list on a page would evaluate the *UserBDate* column value, return the numerical month value, and then compare it against the value in the list.

Type	Separator	Column	CT	Comparison	Value
L		DatePart('m',UserBDate)	N	=	Bdate

It is in this manner that full date values can be evaluated using DataAssist Search server behavior. Additional date part syntax allows filters for values of years, days, etc., allowing comparisons for more specific scenarios involving date columns.

The four database models supported in DataAssist Search server behavior all use their own syntax for evaluating date database columns. Please refer to your documentation for your database type to determine the appropriate syntax for your use.

TimeStamp Evaluations*(Available for ASP developers)*
Dates are typically stored as a number of milliseconds from a certain date. Therefore, if you search a date column with greater than and less than comparisons without specifying a time, the millisecond for midnight of the day requested is used. This creates two scenarios for date comparisons that must be addressed.

In some databases, to search for all dates greater than and not including the requested day, a 23:59:59 must be appended to the end of the requested date. For example, if you search for a date greater than '8/23/01', the database will assume you mean '8/23/01 00:00:00'. The results will therefore include a record with the inserted date '8/23/01 16:19:49' even though you might expect that only dates past and including '8/24/01' would be returned. This same confusion can be true in the less than or equal to comparison. In order to assist you with this problem, there is a function that comes equipped with DataAssist Search server behavior:

WAQB_getEndDate(datestring)

This function accepts any string and returns the date with 23:59:59 appended if no time is specified. The string is not altered if a time already exists. The query entry will be ignored if an improper date or the user does not enter a date.

The following is an example of ASP syntax where the comparison is against a request variable called "enddate":

<%=WAQB_getEndDate(Request("enddate"))%>

Multiple applications of DataAssist Search server behavior to the same recordset

DataAssist Search server behavior can be applied more than once to a single recordset. The first application of DataAssist Search server behavior to the Recordset **MUST** have a default query value. This is typically a comparison on the Primary Key column of the table or query. The most commonly used default query statement is one that will not effect the following applications of DataAssist Search server behavior.

Example:
The unique key column of the JournalEntries table is JournalID. Therefore, the default query statement is "WHERE JournalID<>0". This will not affect any of the other applications of DataAssist Search server behavior that follow.

You will note that in your successive applications to the Recordset, the first "Separator" of your query entries is not ignored. Therefore, if the DataAssist Search server behavior variable is in the request, the generated query will be appended to the previous application with that separator. This is why it is crucial to declare a "Default WHERE Clause" in the first DataAssist Search server behavior application.

Applying DataAssist Search server behavior to a recordset that contains an existing WHERE clause

DataAssist Search server behavior can be applied to recordsets with existing SQL Parameters and existing WHERE clauses. You will note that if a WHERE clause is already in the SQL string for the Recordset, the first separator of your query entries is not ignored. Therefore, if the DataAssist Search server behavior variable is in the request, the generated query will be appended to the existing WHERE clause with that separator.

Issues in Coldfusion referencing #form.checkbox#
and in PHP referecning <?php echo $_POST["checkbox"]; ?>

Please note that in Coldfusion and PHP, a checkbox within a form is not submitted in the request when it is not checked. Any QueryBuilder application in these languages that references a checkbox and makes a comparison against its unchecked value causes an error on the page, as the request will not contain the appropriate checkbox reference.

Managing Single Records

DataAssist provides three server behaviors for managing single records on a page, allowing you to insert, update, or delete individual records based on a trigger.

The DataAssist Wizard applies these server behaviors to the insert, update and delete pages configured for a data management application. As well, these server behaviors can be applied directly to your pages specifc to your needs.

They are as follows, with links to more detailed information on each:

- Insert Record
- Update Record
- Delete Record

Related topics

- DataAssist Wizard
- Managing Multiple Records

Insert Record

This server behavior performs a single database record insert when an event on the page is triggered. It inserts the record into the selected database table using the defined connection.

It is necessary to define the primary key column, and to set a session variable to store the inserted record information for use on the next page in your workflow, especially if displaying that record content.

If using an insert form on the page, you must map the form fields to the database columns designated to receive the record data. Dynamic data also is used as the source for the content used to populate your record columns.

Note: This server behavior is also applied by the DataAssist Wizard when creating a data management application specific to inserting single records into a datasource. Information specific to configuration of the server behavior within the wizard can be found in the step Insert page options.

Access

The Insert Record server behavior is accessible in the following location:

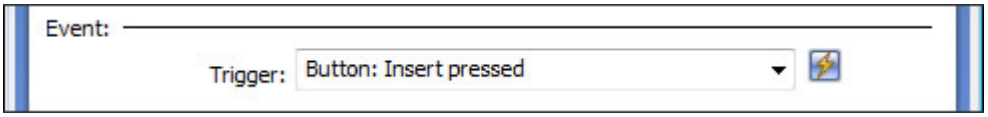
- *Server Behaviors panel > Plus icon (+) > DataAssist > Insert > Single Record Insert*

Step 1: General configuration

Configure the high level configuration details for the insert specific to the initiating event, the database table to receive the data, if a session variable should maintain the record data for further use, and where to redirect to when the insert has been completed.

Event

Determine the page event or trigger used to initiate the record insert into the database:

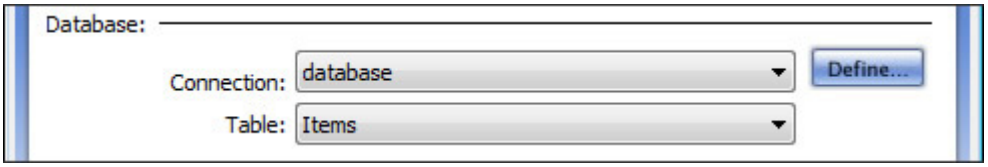


Trigger: Refers to a request, session variable, or other dynamic data events. The existence of a value for the specified variable triggers the insert on a given page. A request object (such as a hidden form element) or a session variable must be submitted or passed to the page that has the server behavior applied to it in order for the insert to execute. Available triggers are:

- **Any Form Post**: Any form posted to the current page that the server behavior is applied to triggers the data insert
- **Before Page Load**: The data insert is triggered when the current page that the server behavior is applied to is loaded.
- **On Form Submit**: Any form submitted on the same page that the server behavior is applied to trigger the data insert.
- **Button [button name] pressed**: The server behavior is triggered if the specified button on the page is clicked.
- **Dynamic Data**: Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.

Database

The following criteria is necessary to properly configure your database connection for use with this server behavior:

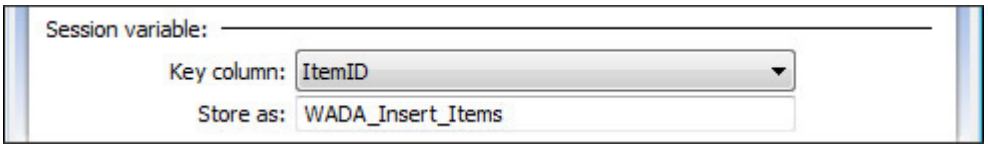


Connection: Selects the database connection used to connect to the database for your application. Select from a list of available connections defined prior, or click the *Define* button to configure a database connection.

Table: Selects the table within the chosen database that is the destination for the new record.

Session variable

Stores the record data for use moving forward after the insertion has taken place. This allows you to maintain this information for use within your data application after the insert has been completed. Useful for displaying the record content on a detail page after the insert is completed.

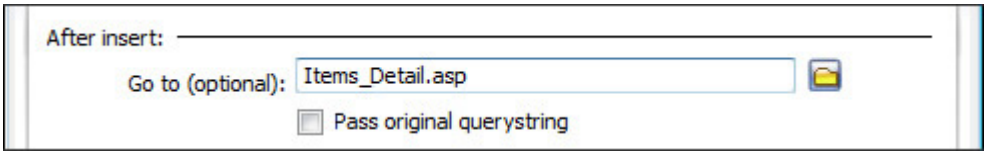


Key column: Specifies the column within the database that is a unique key for the records within the selected table. This determines the column that identifies specific records by virtue of having a unique value.

Store as: Specify the name of the session variable that the inserted record data is to be stored in. If the server behavior is applied by the DataAssist Wizard, a session variable for this purpose is configured automatically, and is available for use elsewhere within your application.

After insert

Sets the actions that take place after insertion of the record.

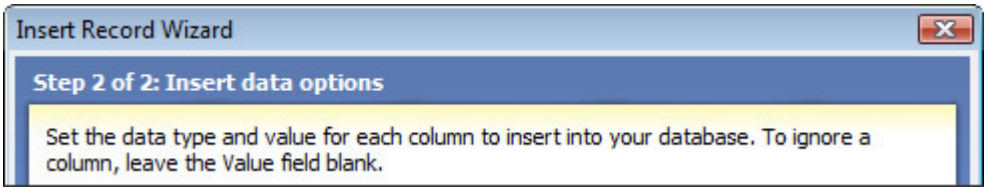


Go to: Sets the redirect location that is loaded upon insertion of the record

Pass original querystring: If a query is currently maintained in your session from actions taking place prior to the record insert (e.g. results page listings returned from a search), checking this box will pass that querystring on to the redirect page, preserving the initial query.

Step 2: Insert data options

Map the form elements or other source of specific record data to the database column receiving it. Select the correct data type for the given database column to ensure data integrity.



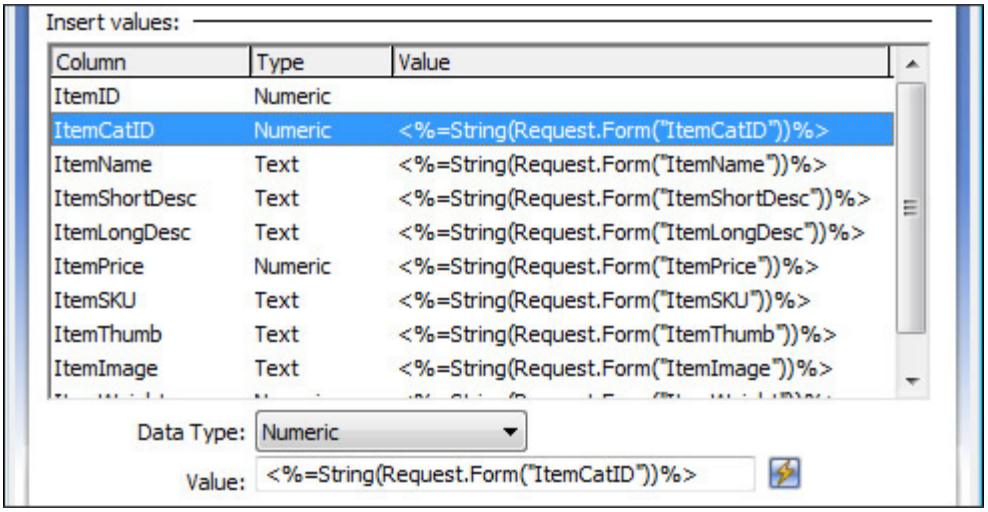
Insert values

Select the datasource columns to be inserted into in the new record. Bind values for all columns that require a value in the database for the record to be created. This includes database columns that do not have a default value specified when one is not supplied and also require a value.

When the server behavior is used in conjunction with an end-user insert form, data for the new record should be supplied through an available form element for a specified column, or the data inserted can be set statically as text or passed from a hidden form element.

In advanced applications of this server behavior, dynamic data sources can also be used to populate columns specific to the requirements of your application.

Note: Autonumber database columns should not be updated during the insert process.



Data type: Sets the data type for the selected database column the data is inserted into. This is necessary to ensure data is inserted using the correct data formatting criteria. The following is a list of available data types to select, but please refer to your database to ensure proper correlation:

- **Text:** Only accepts text values.
- **Numeric:** Only accepts numeric values.
- **Date:** Accepts date values using standard SQL syntax.
- **Date MS Access:** Accepts date values from MS Access databases.
- **Checkbox: Y,N:** Defines checkbox values in a database using Y and N.
- **Checkbox: 1,0:** Defines checkbox values in a database using 1 and 0.
- **Checkbox: -1,0:** Defines checkbox values in a database using -1 and 0.

- **Checkbox: MS Access:** Defines checkbox values in a MS Access database.

Value: Sets the source of the data to be inserted for the selected column. Typically, this will be a form element, but you may also use server-side code to define the inserted information.

Update Record

This server behavior performs a single database record update when an event on the page is triggered. It updates the record in the selected database table using the defined connection.

It is necessary to define the primary key column that identifies the record, and to establish where the record id is being passed from so that the correct record is updated.

If using an update form on the page, you must map the form fields to the database columns designated to receive the record data. Dynamic data sources can also be used as the source for the content used to populate your record columns.

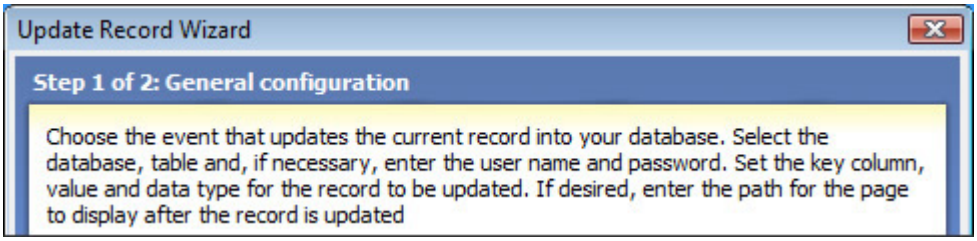
Note: This server behavior is also applied by the DataAssist Wizard when creating a data management application specific to updating a single record in a datasource. Information specific to configuration of the server behavior within the wizard can be found in the step Update page options.

Access

The Update Record server behavior is accessible in the following location:

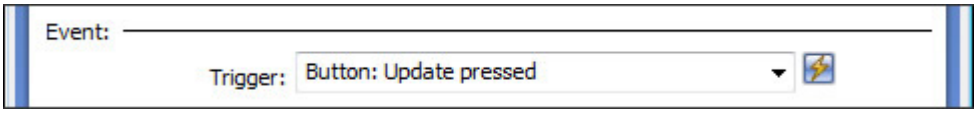
- *Server Behaviors panel > Plus icon (+) > DataAssist > Update > Single Record Update*

Step 1: General configuration



Event

Determine the page event or trigger used to update the record information to the database from the form:

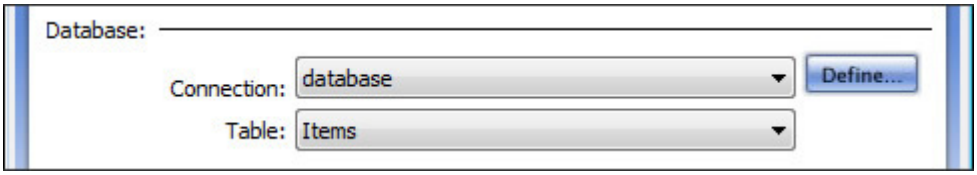


Trigger: Refers to a request, session variable, or other dynamic data events. The existence of a value for the specified variable triggers the update on a given page. A request object (such as a hidden form element) or a session variable must be submitted or passed to the page that has the server behavior applied to it in order for the update to execute. Available triggers are:

- **Any Form Post**: Any form posted to the current page that the server behavior is applied to triggers the data insert
- **Before Page Load**: The data insert is triggered when the current page that the server behavior is applied to is loaded.
- **On Form Submit**: Any form submitted on the same page that the server behavior is applied to trigger the data insert.
- **Button [button name] pressed**: The server behavior is triggered if the specified button on the page is clicked.
- **Dynamic Data**: Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.

Database

The following criteria is necessary to properly configure your database information for use with this server behavior:



Connection: Selects the database connection used to connect to the database for your application. Select from a list of available connections defined prior, or click the *Define* button to configure a database connection.

Table: Selects the table within the chosen database that contains the data to be updated.

Record

Identifies the record to be updated. Establishes the key column that is the unique identifier for the record. In conjunction, establishes the source of the value for the key column to identify the record to be updated. Select the datatype for the key column to ensure that the appropriate syntax is observed when matching the key column value.

Key column: Specifies the column within the database that is a unique key for the records within the selected table. This determines the column that identifies specific records by virtue of having a unique value.

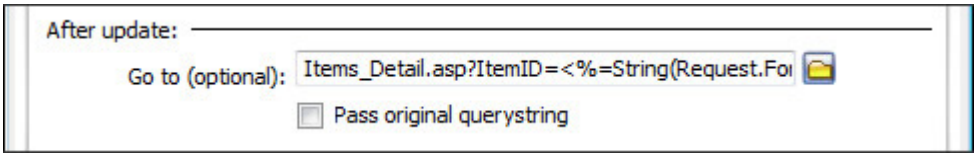
Value: Sets where the record ID is passed to the page from so that the correct record to be updated can be retrieved.

Data type: Sets the data type for the key column to ensure that the data retrieved from the value source matches the database column. This is to verify data integrity and make sure that the correct record is updated. The following is a list of available data types to select, but please refer to your database to ensure proper correlation:

- **Text:** Only accepts text values.
- **Numeric:** Only accepts numeric values.
- **Date:** Accepts date values using standard SQL syntax.
- **Date MS Access:** Accepts date values from MS Access databases.
- **Checkbox: Y,N:** Defines checkbox values in a database using Y and N.
- **Checkbox: 1,0:** Defines checkbox values in a database using 1 and 0.
- **Checkbox: -1,0:** Defines checkbox values in a database using -1 and 0.
- **Checkbox: MS Access:** Defines checkbox values in a MS Access database.

After update

Sets the actions that take place after updating of the record.

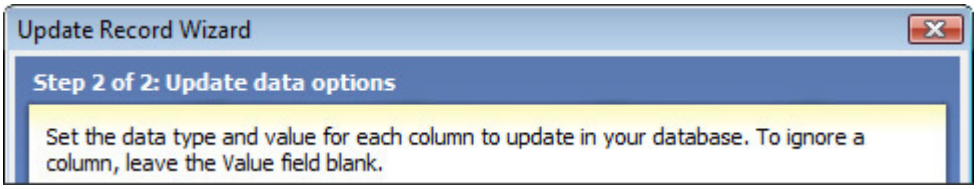


Go to: Sets the redirect location that is loaded upon updating the record

Pass original querystring: If a query is currently maintained in your session from prior to the update, checking this box passes that querystring on to the redirect page.

Step 2: Update data options

Map the form elements or other source of specific record data to the database column receiving it. Select the correct datatype for the given database column to ensure data integrity. *Note*: Autonumber database columns should not be updated during the update process.



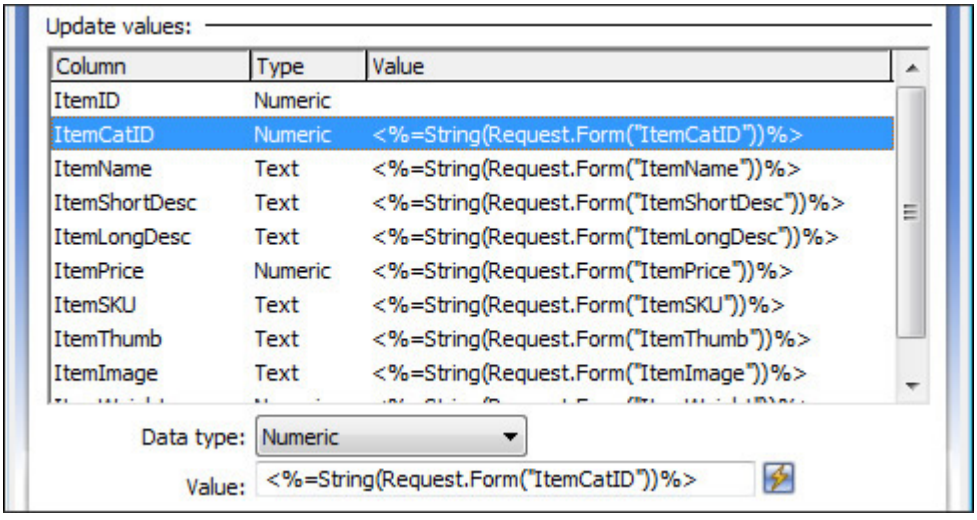
Select the datasource columns to be updated into in the new record. Bind values for all columns that are available for update through this application of the server behavior.

When the server behavior is used in conjunction with an end-user update form, data for the record should be supplied through an available form element for a specified column, or the data updated can be set statically as text or passed from a hidden form element.

In advanced applications of this server behavior, dynamic data sources can also be used to populate columns specific to the requirements of your application.

Note: Autonumber database columns should not be updated during the update process.

Update values



Data type: Sets the data type for the selected database column the data is updated into. This is necessary to ensure data is updated using the correct data formatting criteria. The following is a list of available data types to select, but please refer to your database to ensure proper correlation:

- **Text:** Only accepts text values.
- **Numeric:** Only accepts numeric values.
- **Date:** Accepts date values using standard SQL syntax.
- **Date MS Access:** Accepts date values from MS Access databases.
- **Checkbox: Y,N:** Defines checkbox values in a database using Y and N.
- **Checkbox: 1,0:** Defines checkbox values in a database using 1 and 0.
- **Checkbox: -1,0:** Defines checkbox values in a database using -1 and 0.
- **Checkbox: MS Access:** Defines checkbox values in a MS Access database.

Value: Sets the source of the data to be updated for the selected column. Typically, this will be a form element, but you may also use server-side code to define the updated information.

Delete Record

This server behavior performs a single database record deletion when an event on the page is triggered. It removes the record from the selected database table using the defined connection.

It is necessary to define the primary key column and the source of the unique ID for the record to make sure the correct record is retrieved for deletion and ultimately removed from the database.

Note: This server behavior is also applied by the DataAssist Wizard when creating a data management application specific to deleting a single record in a datasource. Information specific to configuration of the server behavior within the wizard can be found in the step Delete page options.

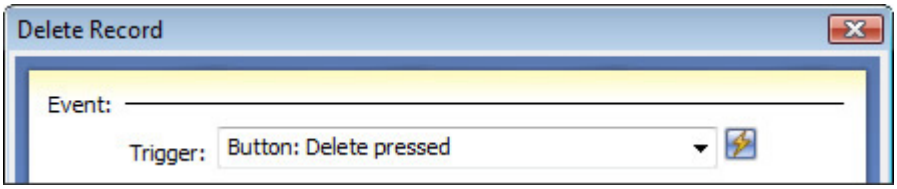
Access

The Delete Record server behavior is accessible in the following location:

- *Server Behaviors panel > Plus icon (+) > DataAssist > Delete > Single Record Delete*

Event

Determine the page event or trigger used to delete the record information from the database.

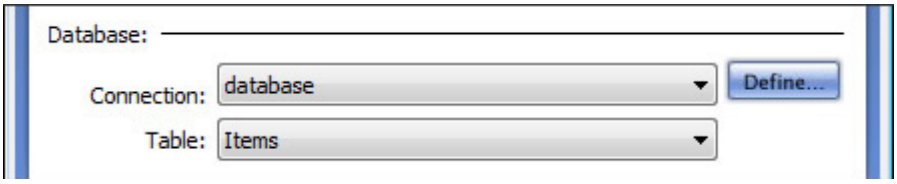


Trigger: Refers to a request, session variable, or other dynamic data events. The existence of a value for the specified variable triggers the record deletion on a given page. A request object (such as a hidden form element) or a session variable must be submitted or passed to the page that has the server behavior applied to it in order for the deletion to execute. Available triggers are:

- **Any Form Post**: Any form posted to the current page that the server behavior is applied to triggers the data insert
- **Before Page Load**: The data insert is triggered when the current page that the server behavior is applied to is loaded.
- **On Form Submit**: Any form submitted on the same page that the server behavior is applied to trigger the data insert.
- **Button [button name] pressed**: The server behavior is triggered if the specified button on the page is clicked.
- **Dynamic Data**: Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.

Database

The following criteria is necessary to properly configure your database information for use with this server behavior:



Connection: Selects the database connection used to connect to the database for your application. Select from a list of available connections defined prior, or click the *Define* button to configure a database connection.

Table: Selects the table within the chosen database that contains the record you are deleting.

Record

Stores the record data for use moving forward after the deletion has taken place.

Key column: Specifies the column with the database that is a unique key for the records within the selected table. This determines the column that identifies specific records by virtue of having a unique value.

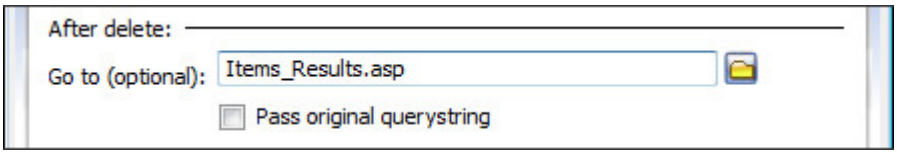
Value: Sets where the record ID is passed to the page from so that the correct record for deletion can be retrieved.

Data Type: Sets the data type for the key column to ensure that the data retrieved from the value source matches the database column. This is to verify data integrity and make sure that the correct record is deleted. The following is a list of available data types to select, but please refer to your database to ensure proper correlation:

- **Text:** Only accepts text values.
- **Numeric:** Only accepts numeric values.
- **Date:** Accepts date values using standard SQL syntax.
- **Date MS Access:** Accepts date values from MS Access databases.
- **Checkbox: Y,N:** Defines checkbox values in a database using Y and N.
- **Checkbox: 1,0:** Defines checkbox values in a database using 1 and 0.
- **Checkbox: -1,0:** Defines checkbox values in a database using -1 and 0.
- **Checkbox: MS Access:** Defines checkbox values in a MS Access database.

After delete

Sets the actions that take place after deleting the record.



Go to: Sets the redirect location that is loaded upon deleting the record

Pass original querystring: If a query is currently maintained in your session from prior to the delete, checking this box will pass that querystring on to the redirect page.

Managing Multiple Records

There are two database usages that need to be covered by data assist: the one-to-many scenario, where there are multiple related records stored in a sub table – and the many-to-many through an intermediate table scenario. These will be explained further in the usage cases described later in this document.

DataAssist provides three server behaviors for managing multiple records on a page at a time, allowing you to insert, update, or delete multiple records based on a trigger.

The DataAssist Wizard creates pages that manage single records. The following server behaviors can be used in conjunction with the Repeat Selection server behavior to modify the insert, update, and delete pages created by the DataAssist Wizard for multiple record management. As well, these server behaviors can be applied directly to your pages specific to your needs.

In addition, the Manage Relational Table server behavior is provided to control an intermediary, or relational, table that connects two other tables. In a typical scenario, a record from one table is combined with details from another table. The relational table includes foreign keys from each of the two other tables.

Each of these features are listed below, with links to more detailed information on each:

- Insert Multiple Records
- Update Multiple Records
- Delete Multiple Records
- Manage Relational Table

Related topics

- Managing Single Records
- Repeat Selection

Insert Multiple Records

This server behavior performs multiple database record insert when an event on the page is triggered. It inserts the record into the selected database table using the defined connection.

A form specified on the page contains the form elements necessary to pass data specific to columns within the database for records to be inserted. That form is contained within the DataAssist Repeat Selection server behavior, which repeats the form for a specified number of iterations, incrementing the names of the repeated form elements as well This allows for the display and possible insert of a corresponding number of simultaneous record inserts using the multiple iterations of the insert form.

It is necessary to define how the inserts are triggered on the page, and how to deal with an iteration of the form that does not contain information to be inserted for a specific column. Determine where the page is to redirect upon completing the inserts, and whether or not to preserve the querystring and pass it to the redirect page.

You must also map the form fields in your repeated insert form on the page to the database columns that will contain the record data, and ensure that they format the inserted data according to the correct datatype for the given column.

Note: This server behavior does not work with forms that use a multipart encoding type (i.e. *enctype="multipart/form-data"*)

Access

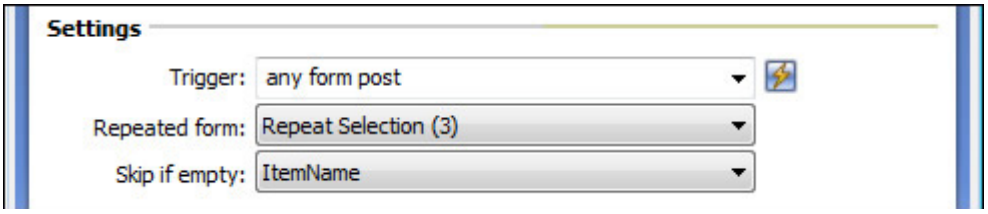
The Insert Multiple Records server behavior is accessible in the following location:

- *Server Behaviors panel > Plus icon (+) > DataAssist > Insert > Multiple Record Insert*

Step 1: General configuration

Event

Determine the page event or trigger used to insert the entered record information to the database:



Trigger: Refers to a request, session variable, or other dynamic data events. The existence of a value for the specified variable triggers the insert on a given page. A request object (such as a hidden form element) or a session variable must be submitted or passed to the page that has the server behavior applied to it in order for the insert to execute. Available triggers are:

- **Any Form Post**: Any form posted to the current page that the server behavior is applied to triggers the data insert
- **Before Page Load**: The data insert is triggered when the current page that the server behavior is applied to is loaded.
- **On Form Submit**: Any form submitted on the same page that the server behavior is applied to trigger the data insert.

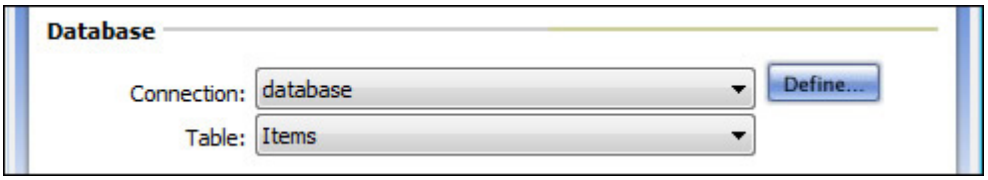
- **Button [button name] pressed**: The server behavior is triggered if the specified button on the page is clicked.
- **Dynamic Data**: Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.

Repeated form: Select a form on the page that is contained within a region encompassed by the DataAssist Repeat Selection server behavior.

Skip if empty: Skips the insert of a specific record If no value is passed for the database column selected here. If you wish to insert placeholder records into your database to be populated at a later time, but that may not contain data currently, you have the option of selecting Always insert, which will allow the insert of data containing blank fields, so long as the corresponding database columns have default values specified in the database on insert, or can accept NULL values.

Database

The following criteria is necessary to properly configure your database information for use with this server behavior:

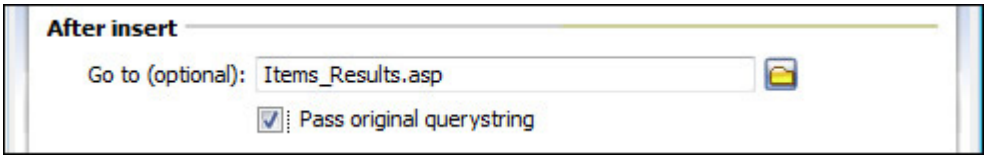


Connection: Selects the database connection used to connect to the database for your application. Select from a list of available connections defined prior, or click the *Define* button to configure a database connection.

Table: Selects the table within the chosen database that contains the records being inserted.

After insert

Specifies the destination page redirected to upon completing the records insert, and determines if a querystring that existed prior to the insert should be perpetuated to that page.

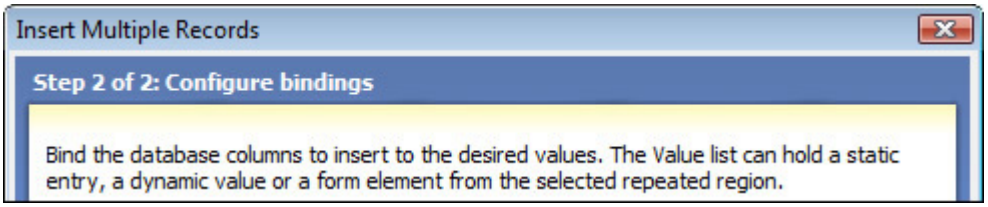


Go to: Sets the redirect location that is loaded upon insertion of the record

Pass original querystring: If a query is currently maintained in your session from prior to the insert, checking this box will pass that querystring on to the redirect page.

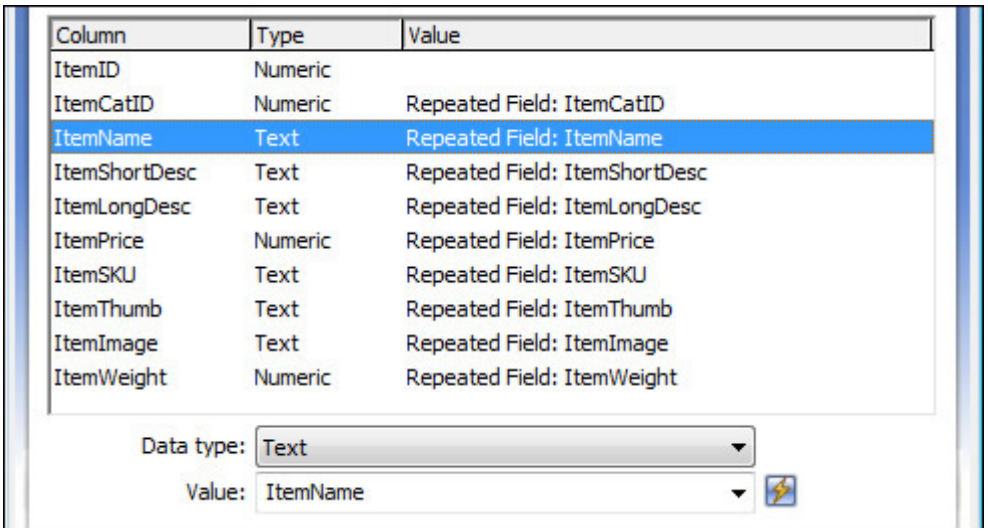
Step 2: Insert data options

Map the repeated form elements or other source of specific record data to the database column receiving it. Select the correct data type for the given database column to ensure data integrity.



Insert values

Note: Autonumber database columns should not be updated during the insert process.



Data type: Sets the data type for the selected database column the data is inserted into. This is necessary to ensure data is inserted using the correct data formatting criteria. The following is a list of available data types to select, but please refer to your database to ensure proper correlation:

- **Text:** Only accepts text values.
- **Numeric:** Only accepts numeric values.
- **Date:** Accepts date values using standard SQL syntax.
- **Date MS Access:** Accepts date values from MS Access databases.
- **Checkbox: Y,N:** Defines checkbox values in a database using Y and N.
- **Checkbox: 1,0:** Defines checkbox values in a database using 1 and 0.
- **Checkbox: -1,0:** Defines checkbox values in a database using -1 and 0.
- **Checkbox: MS Access:** Defines checkbox values in a MS Access database.

Value: Sets the source of the data to be inserted for the selected column. Typically, this will be a form element, but you may also use server-side code to define the inserted information.

Update Multiple Records

This server behavior performs multiple database record updates when an event on the page is triggered. It updates the available records in the selected database table using the defined connection.

A form specified on the page contains the form elements necessary to pass data to specific columns within the database for records to be updated. A single update form is contained within the Dreamweaver Repeat Region server behavior, which repeats the form for the selected number of records from the specified recordset. Setting the values for the form elements to the columns in the recordset populates them with corresponding column data for the given record.

When the Update Multiple Records server behavior is applied, the DataAssist Repeat Selection server behavior is applied as well (if not already applied to the form), incrementing the names of the repeated form elements to allow the records to be correctly correlated with the records they are to update. This allows for the display and update of each record populated in the repeating form using the multiple iterations of the update form.

It is necessary to define how the updates are triggered on the page, as well as where the page is to redirect upon completing the inserts, and whether or not to preserve the querystring and pass it to the redirect page.

You must also map the form fields in your repeated update form on the page to the database columns that are to contain the record data. Ensure that the inserted data is formatted according to the correct datatype for the given column.

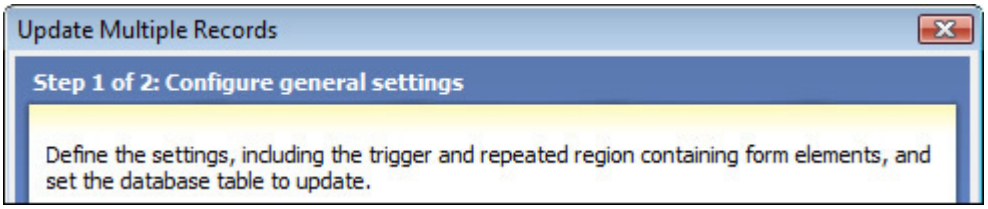
Note: This server behavior does not work with forms that use a multipart encoding type (i.e. *enctype="multipart/form-data"*)

Access

The Update Multiple Records server behavior is accessible in the following location:

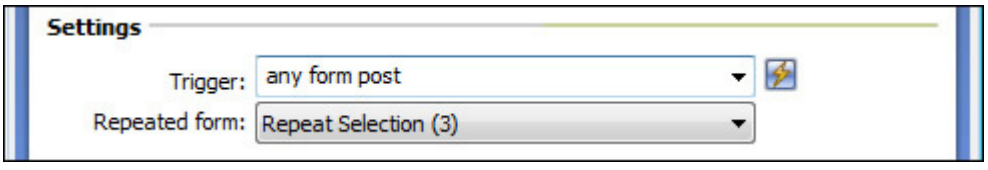
- *Server Behaviors panel > Plus icon (+) > DataAssist > Update > Multiple Record Update*

Step 1: General configuration



Event

Determine the page event or trigger used to update the record information to the database from the form:



Trigger: Refers to a request, session variable, or other dynamic data events. The existence of a value for the specified variable triggers the update on a given page. A request object (such as a hidden form element) or a session variable must be submitted or passed to the page that has the server behavior applied to it in order for the update to execute. Available triggers are:

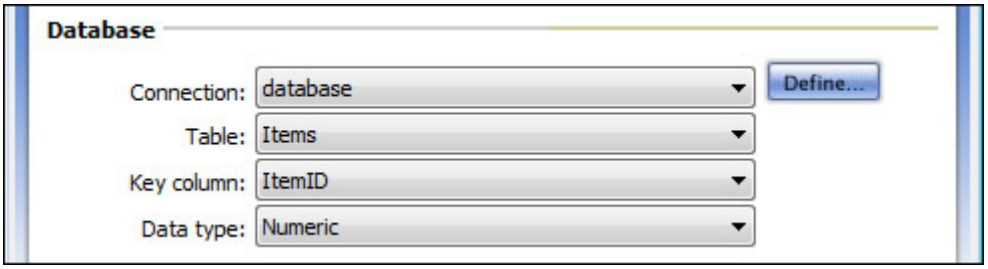
- **Any Form Post**: Any form posted to the current page that the server behavior is applied to triggers the data update.
- **Before Page Load**: The data update is triggered when the current page that the server behavior is applied to is loaded.

- **On Form Submit**: Any form submitted on the same page that the server behavior is applied to trigger the data update.
- **Button [button name] pressed**: The server behavior is triggered if the specified button on the page is clicked.
- **Dynamic Data**: Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.

Repeated form: Select a form on the page that is contained within a region encompassed by the Dreamweaver Repeat Region server behavior, or the DataAssist Repeat Selection server behavior.

Database

The following criteria is necessary to properly configure your database information for use with this server behavior:



Connection: Selects the database connection used to connect to the database for your application. Select from a list of available connections defined prior, or click the *Define* button to configure a database connection.

Table: Selects the table within the chosen database that contains the data used by the pages you are creating

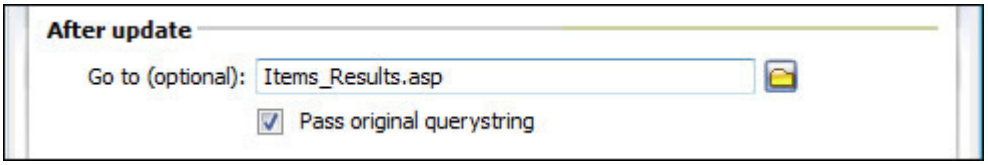
Key column: Specifies the column within the database that is a unique key for the records within the selected table. This determines the column that allows the identification of specific records by virtue of having a unique value.

Data type: Sets the data type for the key column to ensure that the data retrieved from the value source matches the database column. This is to verify data integrity and make sure that the correct record is updated. The following is a list of available data types to select, but please refer to your database to ensure proper correlation:

- **Text:** Only accepts text values.
- **Numeric:** Only accepts numeric values.
- **Date:** Accepts date values using standard SQL syntax.
- **Date MS Access:** Accepts date values from MS Access databases.
- **Checkbox: Y,N:** Defines checkbox values in a database using Y and N.
- **Checkbox: 1,0:** Defines checkbox values in a database using 1 and 0.
- **Checkbox: -1,0:** Defines checkbox values in a database using -1 and 0.
- **Checkbox: MS Access:** Defines checkbox values in a MS Access database.

After update

Sets the actions that take place after updating of the record.

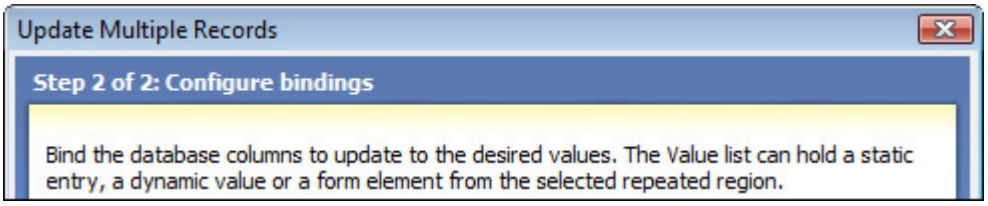


Go to: Sets the redirect location that is loaded upon updating the records

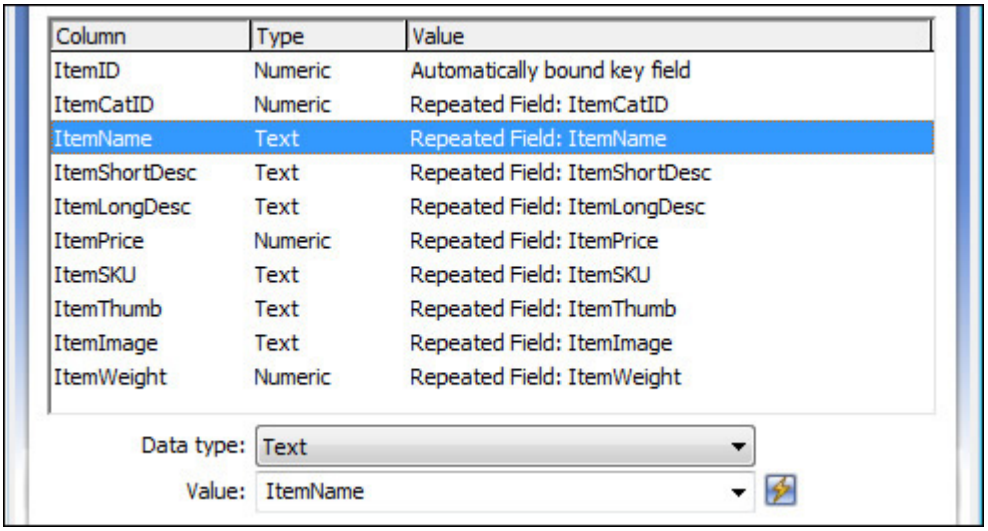
Pass original querystring: If a query is currently maintained from prior to the update, checking this box will pass that querystring on to the redirect page.

Step 2: Update data options

Map the repeated form elements or other source of specific record data to the database column receiving it. Select the correct datatype for the given database column to ensure data integrity. *Note*: Autonumber database columns should not be updated during the insert process.



Update values



Data type: Sets the data type for the selected database column the data is updated into. This is necessary to ensure data is updated using the correct data formatting criteria. The following is a list of available data types to select, but please refer to your database to ensure proper correlation:

- **Text:** Only accepts text values.
- **Numeric:** Only accepts numeric values.
- **Date:** Accepts date values using standard SQL syntax.
- **Date MS Access:** Accepts date values from MS Access databases.
- **Checkbox: Y,N:** Defines checkbox values in a database using Y and N.

- **Checkbox: 1,0:** Defines checkbox values in a database using 1 and 0.
- **Checkbox: -1,0:** Defines checkbox values in a database using -1 and 0.
- **Checkbox: MS Access:** Defines checkbox values in a MS Access database.

Value: Sets the source of the data to be updated for the selected column. Typically, this will be a form element, but you may also use server-side code to define the updated information.

Delete Multiple Records

This server behavior performs a database record deletion when an event on the page is triggered. It removes the record from the selected database table using the defined connection.

It is necessary to define the primary key column and the source of the unique ID for the record to make sure the correct record is retrieved for deletion and ultimately removed from the database.

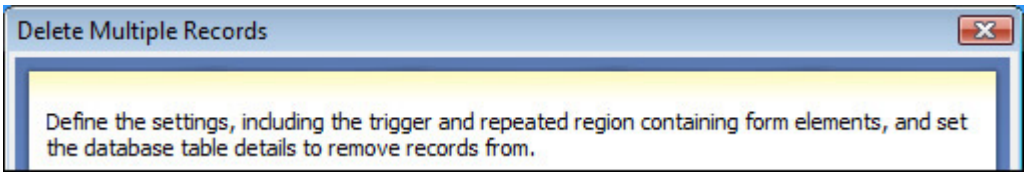
Access

The Delete Multiple Records server behavior is accessible in the following location:

- *Server Behaviors panel > Plus icon (+) > DataAssist > Delete > Multiple Record Delete*

Event

Determine the page event or trigger used to delete the record information from the database.

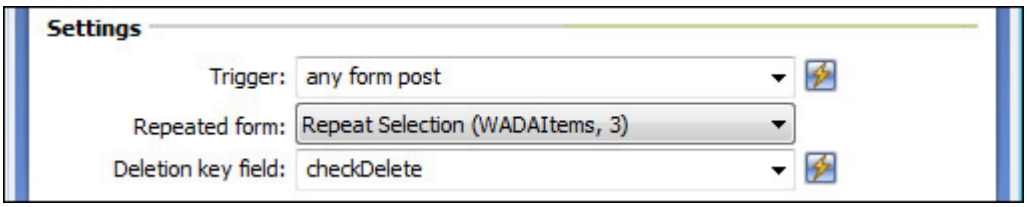


Trigger: Available triggers are:

- any form post
- before page load
- current page submit
- a selected button on the page

Database

The following criteria is necessary to properly configure your database information for use with this server behavior:

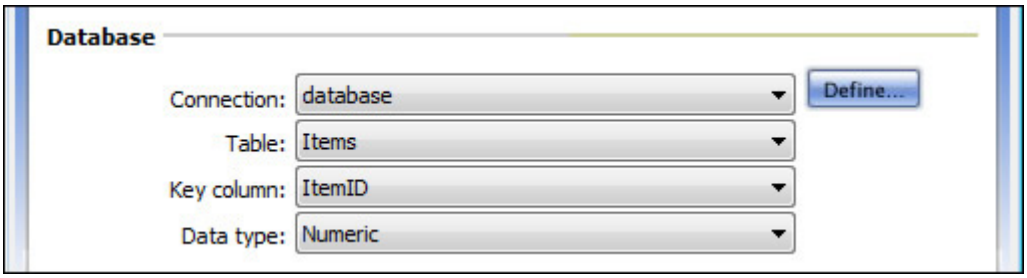


Connection: Selects the database connection used to connect to the database for your application. Select from a list of available connections defined prior, or click the *Define* button to configure a database connection.

Table: Selects the table within the chosen database that contains the data used by the pages you are creating

Record

Stores the record data for use moving forward after the deletion has taken place.



Key column: Specifies the column with the database that is a unique key for the records within the selected table. This determines the column that allows the identification of specific records by virtue of having a unique value.

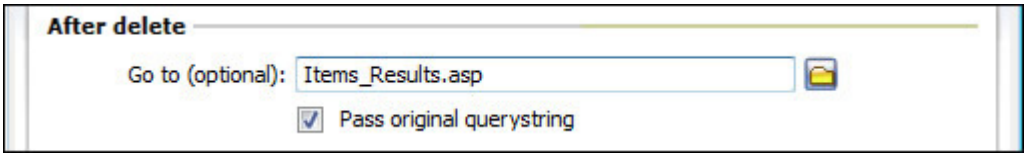
Value: Sets where the record id is passed to the page from so that the correct record for deletion can be retrieved.

Data Type: Sets the data type for the key column to ensure that the data retrieved from the value source matches the database column. This is to verify data integrity and make sure that the correct record is deleted. The following is a list of available data types to select, but please refer to your database to ensure proper correlation:

- **Text:** Only accepts text values.
- **Numeric:** Only accepts numeric values.
- **Date:** Accepts date values using standard SQL syntax.
- **Date MS Access:** Accepts date values from MS Access databases.
- **Checkbox: Y,N:** Defines checkbox values in a database using Y and N.
- **Checkbox: 1,0:** Defines checkbox values in a database using 1 and 0.
- **Checkbox: -1,0:** Defines checkbox values in a database using -1 and 0.
- **Checkbox: MS Access:** Defines checkbox values in a MS Access database.

After delete

Sets the actions that take place after deleting the record.



Go to: Sets the redirect location that is loaded upon deleting the record

Pass original querystring: If a query is currently maintained in your session from prior to the delete, checking this box will pass that querystring on to the redirect page.

Manage Relational Table

The Manage Relational Table server behavior controls the database information in an intermediary, or relational, table that connects two other tables.

In a typical scenario, a record from one table is combined with details from another table. The relational table includes foreign keys from each of the two other tables.

For example, an application might include a form that inserts new registrants for a conference; in addition to the standard information about the registrant (such as name, address, etc.) which would be stored in a Registrant table, the form might also include a list of available seminars at the conference. The seminar details are stored in a Presentations table and displayed on the form as a series of checkboxes or a multi-select list. A third table, called RegPres, contains records which connects the two other tables and allows for the application to display both information about each registrant and the registrant's chosen seminars (one-to-many) and all the registrants in any given seminar (many-to-many).

To be successful with the Manage Relational Table server behavior, a database schema must contain a minimum of three interconnected tables:

- *Main*: The principal table that includes a primary key and other data fields.
- *Options*: The secondary table that includes a primary key and other data fields.
- *Relational*: The tertiary table that includes a primary key and two foreign keys, one for the Main table and one for the Options table.

The form in the application to which the Manage Relational Table server behavior is applied typically contains data from the Options table presented in one of two ways:

- A multi-select list, populated dynamically by a recordset derived from the Options table.
- A series of checkboxes and labels, dynamically created, within a repeat region.

Access

The Manage Relational Table server behavior is accessible in the following location:

- *Server Behaviors panel > Plus icon (+) > DataAssist > Manage Relational Table*

Configuration details

The following pages in this section detail the configuration options available in this server behavior:

- Step 1: Configure general settings

  Sets the triggering event, the database connection, and relational table to use. Optionally, a page to display after the database operation is completed can be specified.

- Step 2: Configure related table main column information

  Selects the foreign key in the relational table that connects to the main table, with the proper data type and value.

- Step 3: Configure related table options column information

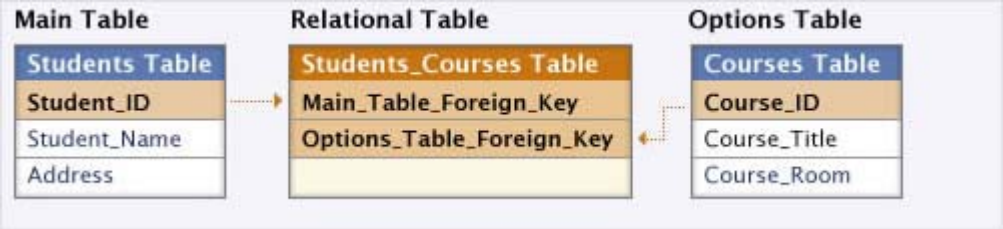
  Defines the foreign key in the relational table that connects to the options table with the proper data type.

- Step 4: Configure insert and update bindings

  Sets the relationship between database fields in the relational table and form data.

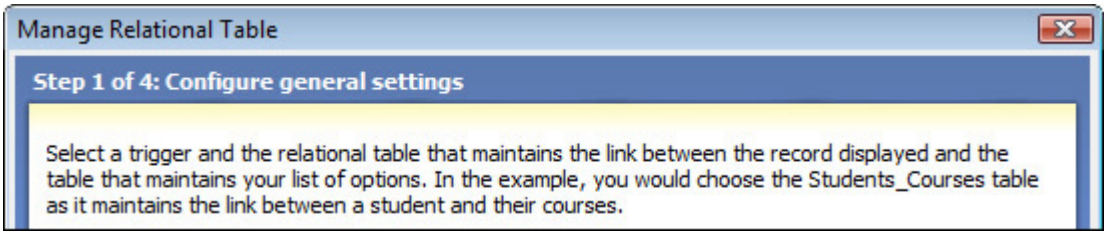
Relationship example

To help users understand how relational tables work and the relevant information requested on each page of the Manage Relational Table server behavior, each step of the server behavior includes a diagram of an example use. Descriptions in the user interface reference the diagram to clarify the process.



Step 1: Configure general settings

The first step of the Manage Relational Table server behavior sets the triggering event, the database connection, and relational table to use. Optionally, a page to display after the database operation is completed can be specified.



Settings

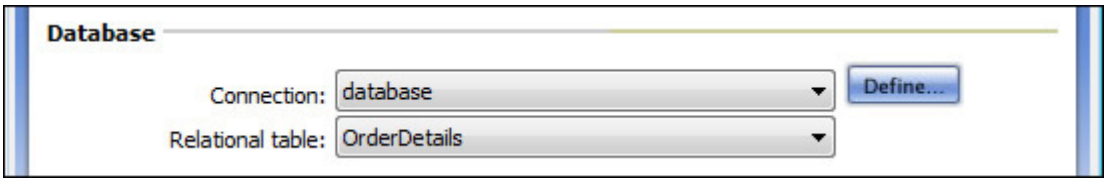
Determine the page event or trigger used to insert the entered record information to the database:

Trigger: Available triggers are:

- **any form post**: Any form posted to the current page that the server behavior is applied to will trigger the record insertion.
- **before page load**: The record insert is triggered when the current page that the server behavior is applied to is loaded.
- **current page submit**: Any form submitted on the same page that the server behavior is applied to will trigger the record insertion.
- **Button [button name] pressed**: A button on the page, typically in the form, is clicked by the user.
- **Recordset [recordset name] is not empty**: The server behavior is triggered if the specified recordset applied to the page contains no items.
- **Recordset [recordset name] is empty**: The server behavior is triggered if the specified recordset applied to the page shopping cart contains no items.

Database

Defines the database connection which contains the main, option and relational table, as described in the introductory section:

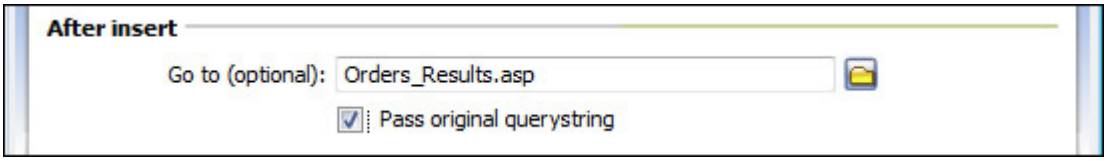


Connection: Selects the database connection used to connect to the database for your application. Select from a list of available connections, or click the *Define* button to configure a database connection.

Relational table: Selects the table in the database which connects the main and options table.

After insert

Sets the (optional) page to be displayed after the record insertion is complete.



Go to: Sets the redirect location that is loaded upon insertion/update of the record

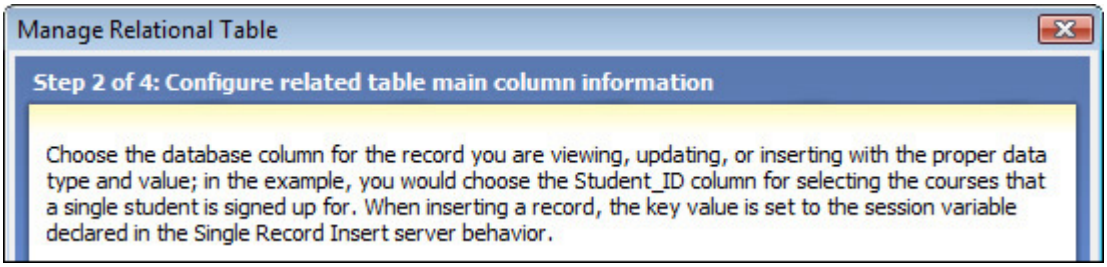
Pass original querystring: If a query is currently maintained in your session from prior to the insert, checking this box will pass that querystring on to the redirect page.

Relationship example

In the example diagram, the table Students_Courses would be selected from the Relational table list.

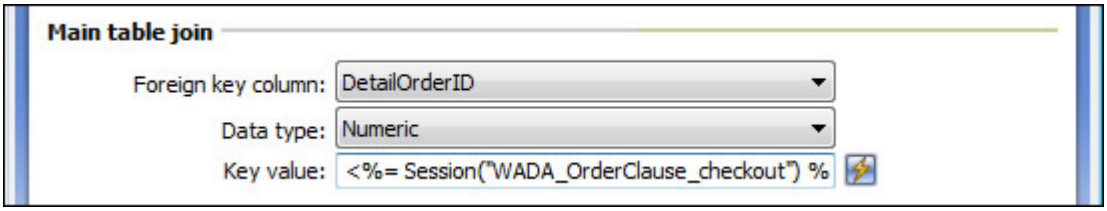
Step 2: Configure related table main column information

The second step of the Manage Relational Table server behavior sets the for primary key for the database column for the record currently viewed, updated, or inserted with the proper data type and value.



Main table join

Determines the data field in the relational table that connects to the main table and sets its data type and the key value.



Foreign key column: The data field of the relational table which contains the value of the primary key column of the main table.

Data type: The data type of the Foreign key column, typically Numeric.

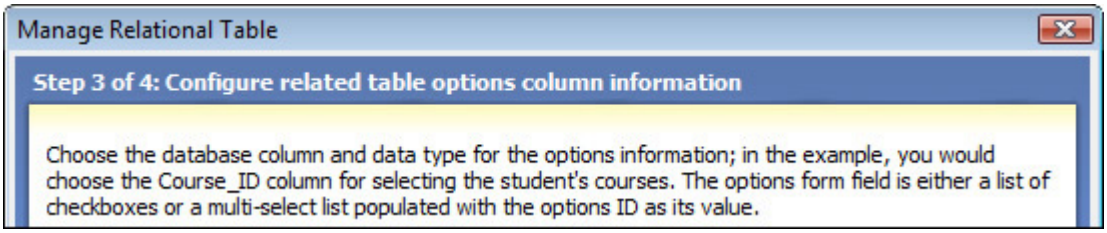
Key value: The value of the Foreign key column to apply during the database operation. When inserting a record, this field is set to the session variable declared in the Single Record Insert server behavior. For other operation, this field should be set to a request or session variable associated with the main table primary key.

Relationship example

In the example diagram, the Foreign key column would be set to Main_Table_Foreign_Key.

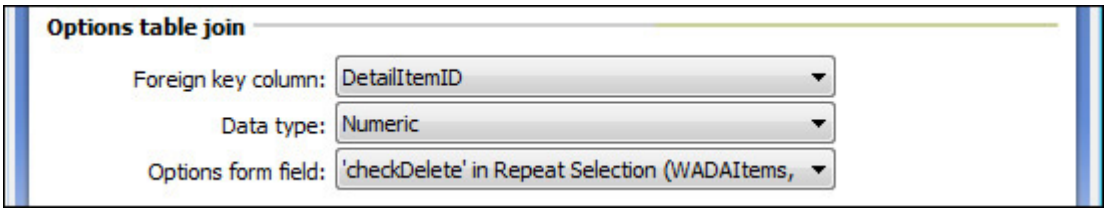
Step 3: Configure related table options column information

The third step of the Manage Relational Table server behavior sets the foreign key in the relational table that connects to the options table with the proper data type and specifies the form element which displays data from the options table.



Options table join

Determines the data field in the relational table that connects to the options table and sets its data type and the form field which contains the options table data:



Foreign key column: The data field of the relational table which contains the value of the primary key column of the options table.

Data type: The data type of the Foreign key column, typically Numeric.

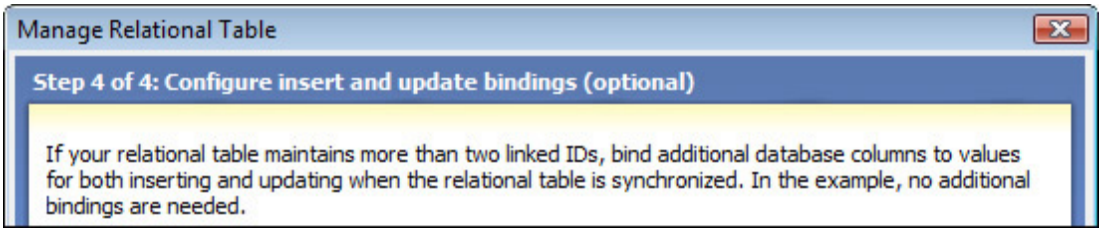
Options form field: The form field on the current page which displays data from the options table. This form field may either be a series of checkboxes and labels within a repeat region or a multi-select list.

Relationship example

In the example diagram, the Foreign key column would be set to Options_Table_Foreign_Key.

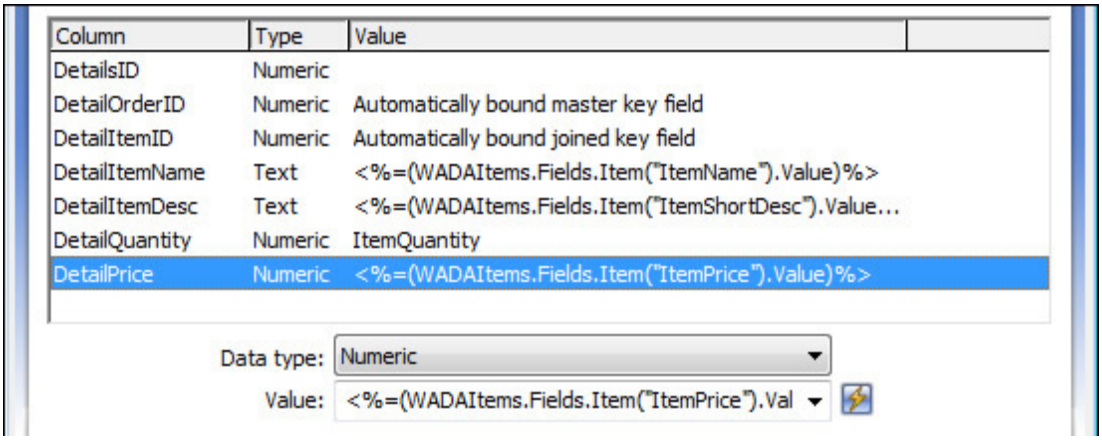
Step 4: Configure insert and update bindings

The fourth and final step of the Manage Relational Table server behavior binds form fields to the fields in the relational table. The previously determined foreign key fields for the main and options tables are automatically bound for the user. Additional bindings may be set in this step, if desired.



Bindings

The Bindings interface connects a database column to a form field or other available dynamic value. To make or modify a binding, select the desired entry from the list, choose the appropriate data type and select the desired value. For dynamic data values, click the lightning bolt icon and select the value from the Dynamic Data dialog box.



Data type: The data type of the selected database column.

Value: The value bound to the selected database column. For the main and option foreign keys, the value is pre-filled with the phrases "Automatically bound master key field" and "Automatically bound joined key field," respectively.

Relationship example

In the example diagram, no additional bindings are required as the relational table contains no other database columns other than its primary key and the two foreign keys.

Repeat Selection

The Repeat Selection server behavior allows you to select a portion of code in your page and repeat it a specific number of times. When used with a recordset on the page, records returned can be iterated through and displayed on the page in the repeated area.

Dreamweaver's Repeat Region server behavior allows you to repeat recordset content across a repeating region, but only in a single direction. It can be applied to a table row, thus repeating recordset content vertically, or it can be applied to a table cell, repeating content horizontally.

The DataAssist Repeat Selection server behavior, when used in conjunction with the the Dreamweaver Repeat Region server behavior, allows you to control the number of records displayed in each row, for multiple rows.

As this is a common application of these two server behaviors, DataAssist includes the Repeating Table feature specific to easily configuring their use together. The DataAssist Wizard also creates this common application of the two server behaviors when configuring a results page display discussed in the Recordset paging section of the wizard.

As well, Repeat Selection can be used in conjunction with Dreamweaver's Recordset Navigation Bar to move through a selection of records returned by the recordset that exceeds the number configured for display.

When the server behavior is applied, it adds a *Counter* binding available on the page through the *Bindings* panel. This binding is used directly within the server behavior as a loop counter, incrementing for each iterations of the selected region it has displayed until it reaches its configured number. It is also used primarily with the Insert Multiple Records and Update Multiple Records server behaviors to increment the names of repeated form fields. It can also be helpful for display as a row counter.

Access

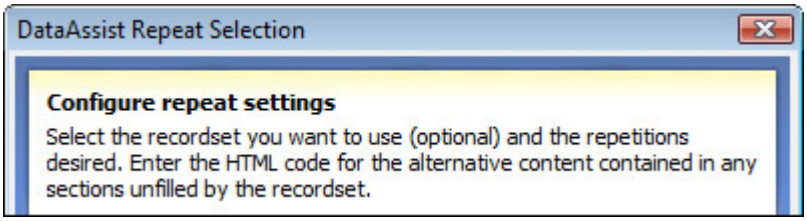
The Repeat Selection server behavior can be accessed in the following location:

- *Server Behaviors panel > Plus icon (+) > DataAssist > Repeat Selection*

Application

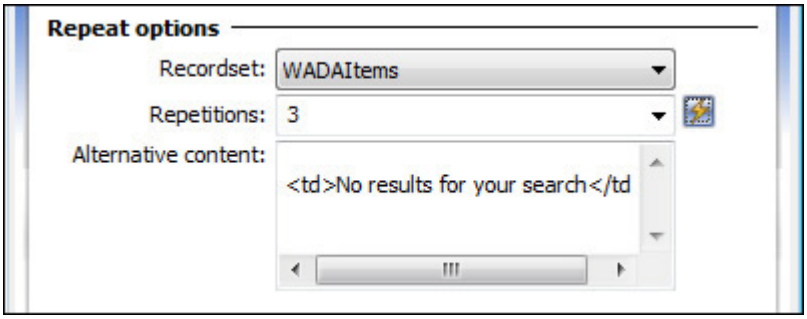
To apply the server behavior correctly, select the region on the page to be repeated. For example, if repeating a row in a table, be sure that the <tr> to be repeated is highlighted on the page at the time the server behavior is applied. Multiple types of tags that encompass an area on the page can be used, ranging, from table cells (<td>), table rows (<tr>), and full tables (<tables>) to paragraphs (<p>) list items (<li>) and DIVs ( ).

Note: Be sure that when you apply the server behavior to a table cell, you take the existing structure of the table into account. Ensure that the number of table cells displayed within the repeated selection area coincides with the number of cells displayed in other rows in the same table that are not included encompassed by the server behavior. This will prevent any display abnormalities caused by different rows in the table having a different number of table cells.



Repeat options

Configure the attributes for the content and the number of times it is to be repeated. If you are looping through a recordset and repeating a region to display content from distinct records, select the recordset on the page that returns the data to be displayed. Select the number of times the region is to be repeated. If additional iterations are specified, but there are not enough records to complete the display, enter the alternate content to be used as placeholder content.



Recordset: A list of available recordsets on the current page. Select *None* if you are not using the server behavior to display recordset content. Otherwise, select the recordset specific to the data to be displayed in the selection.

Repetitions: The number of times the selected content is to be repeated. Select from the list of available options, or enter a custom numeric value. A dynamic value can also be specified using the dynamic data source dialog accessed through the lightning bolt icon.

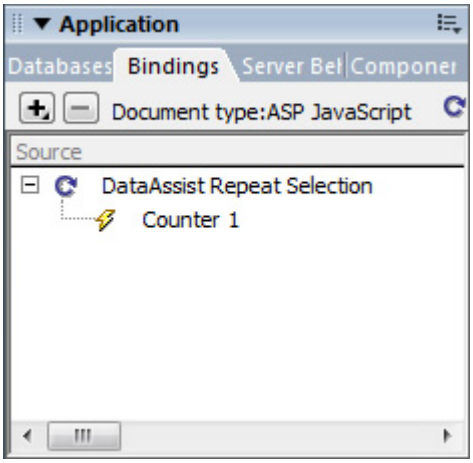
Alternative content: Additional content that is to be output when a recordset has been exhausted. This content should contain the appropriate structure to ensure that design elements are rendered appropriately. To assist in this, the following default content options are displayed depending on the selection the server behavior is applied to, and whether a recordset is selected:

- <td> selected on the page: "*<td> </td>*"
- Recordset selected in the "Recordset" field, and <tr> selected on the page: "*<tr><td> </td></tr>*"
- If more than one table cell is found within the selected <tr>, then the alternative content <tr> contains the same number of table cells containing non-breaking spaces.

Repeat Selection data binding

The Repeat Selection selection server behavior uses a *Counter* binding specific to each application of the server behavior on a given page. Dragging the binding to the page adds code that outputs the current value of the loop counter for the selected Repeat Selection data binding. This counter binding is primarily used by the Insert Multiple Records and Update Multiple Records server behaviors to increment form field names. This binding can also be used as a row counter.

Note: The counter value starts at 0, so you may wish to add 1 to the current value in the binding code if using it for display considerations (e.g. <%=RepeatSelectionCounter_1+1%>).



DataAssist Repeating Table

DataAssist Repeating Table creates a record display that contains a specified number of rows, and a specified number of records per row (columns). Alternative content, styling (alternating row colors), header and accessibility options are available for customization.

This feature simplifies the application of the Dreamweaver Repeat Region server behavior and the DataAssist Repeat Selection server behavior required to make this type of display functionality possible.

Specify the recordset containing the records for display, the number of rows and columns for the generated table to have, as well as alternative content to complete a row if the recordset content runs out.

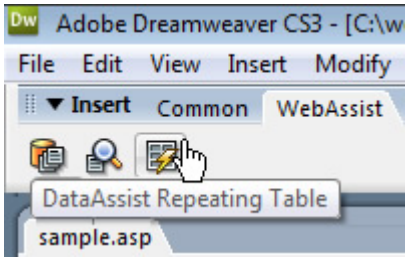
You have control over the CSS classes applied to even and odd rows in the display, and can also apply CSS options for the table such as the ID for the table. You can also select whether to display header content for database columns within the table, and how they should be positioned. Accessibility considerations for the table are also available.

In addition, this table can be used in conjunction with Dreamweaver's Recordset Navigation Bar to allow users to move through a selection of records returned by the recordset that exceeds the number configured for display.

Access

The following locations in Dreamweaver open the DataAssist Repeating Table interface to be inserted within your current cursor location on the page:

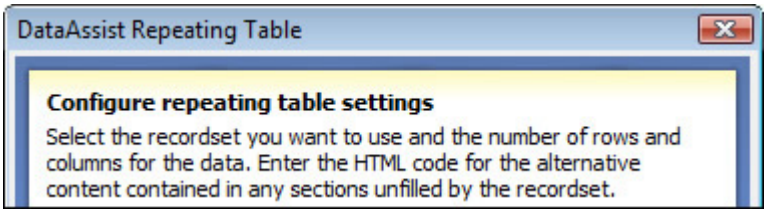
- WebAssist Insert panel



- *Insert > WebAssist > DataAssist > Repeating Table*

Configuration

Specify the recordset, table size and alternate content attributes, CSS styles specific to alternating table rows, the positioning of header content, and accessibility content.



Recordset

This list of available recordsets on the current page that can be used to populate the table.

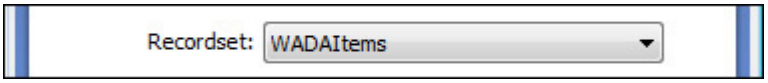
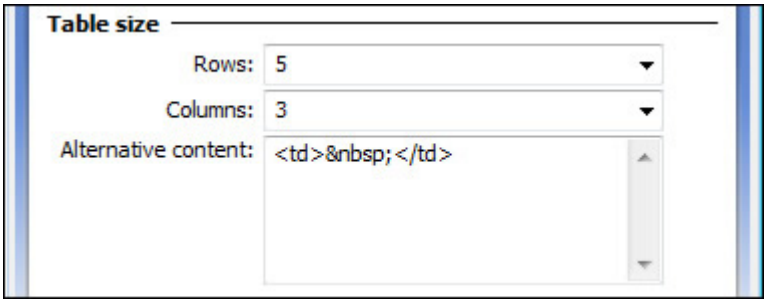


Table size

Specify the number of rows to be displayed, the number of columns (records in each row) to be displayed, and the alternative content displayed when insufficient records are available in the recordset to complete a row.



Rows: Select the number of rows to generate. Select from the list of available options, or enter a custom numeric value. A dynamic value can also be specified using the dynamic data source dialog accessed through the lightning bolt icon.

Columns: Select the number of columns to generate. Select from the list of available options, or enter a custom numeric value. A dynamic value can also be specified using the dynamic data source dialog accessed through the lightning bolt icon. Selecting *All records* disables the **Rows** options, as only one row is displayed.

Alternative content: Additional content that is to be output when a recordset has been exhausted. The default value is an empty table cell to correspond to each empty record value in the row: *"<td> </td>"*

Styling

Specify the CSS styles to be applied to the table. Select a style applicable to the entire level, as well as specific styles applicable to the odd and even rows within the table.

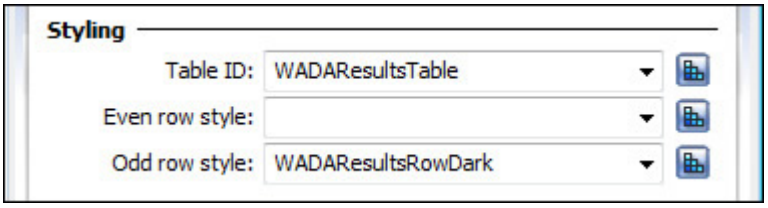


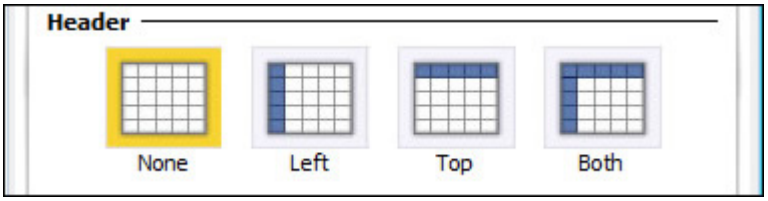
Table ID: Select or specify styles associated to ID tags to apply to the inserted table. The list displays all CSS IDs that are either available on the page, or in style sheets linked to by the page, but that have not been used on the page. If you wish to create a new style during configuration, click the button to open the *New CSS Rule* dialog, and configure the stylistic attributes for your table directly. When you have completed creating a new CSS rule, it is added to the current list for selection.

Even row style: Select the class to use for even data rows in the generated table. The list displays all CSS class names that are either available on the page, or in style sheets linked to by the page. If you wish to create a new style during configuration, click the button to open the *New CSS Rule* dialog, and configure the stylistic attributes for even rows directly. When you have completed creating a new CSS rule, it is added to the current list for selection.

Odd row style: Select the class to use for odd data rows in the generated table. The list displays all CSS class names that are either available on the page, or in style sheets linked to by the page. If you wish to create a new style during configuration, click the button to open the *New CSS Rule* dialog, and configure the stylistic attributes for odd rows directly. When you have completed creating a new CSS rule, it is added to the current list for selection.

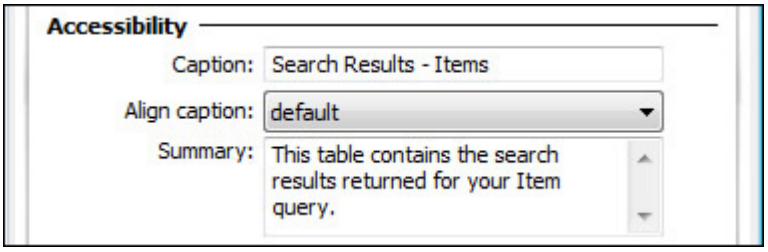
Header

Select the location, if any, of the column headers associated to the database columns returned from the recordset. Assists in identifying content within records in the display.



Accessibility

Specifies accessibility attributes applied to the table that allow people with alternate needs for accessing content to be able to get meaningful information from the data displayed.



Caption: Specifies the content included within the <caption> tag applied in the table.

Align caption: Positioning for the table caption. Available options are:

- Default
- top
- bottom
- left
- right

Summary: Populates the content for the *Summary* property applied to the <table> tag

Sort

DataAssist's Sort server behavior orders the records returned by a selected recordset to a page based on specified criteria. The sorting is triggered by a specified event on the page. This can be a form post, submission, or on page load, but is typically configured to be initiated by the end user. For this reason, you have the option of specifying dynamic data that can be tied to an event on the page, such as clicking a link, or making a selection from a list.

A session variable specified on the page tracks the current sort criteria applied to the recordset. This makes it possible for you to specify a default sort clause applied to the recordset, as well as use the *Toggle sort on subsequent visits* option that allows you to reverse the order of the records returned the next time the server behavior is triggered. For example, if using a link on the page to trigger the sort, clicking the link again reverses the sort order.

You can select multiple columns for a single sort, ordering them according to the hierarchy you choose. For example, a table containing products could have a sort applied against the price first and the name second. The results would be ordered according to the price, but records containing the same price would then be ordered according to their name.

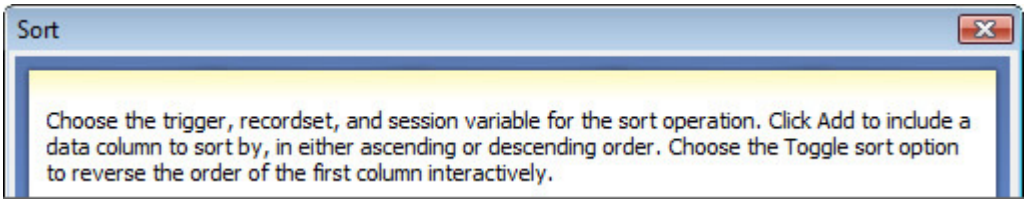
Dynamic data can be used to select the column the sort is applied to. This allows you to use a single application of the sort behavior to control multiple sort scenarios from a single event on the page. The most common example of this implementation submits a URL parameter to the page specific to the column to be sorted by, either through links (or a select list) containing values for each record column available as sort criteria. The parameter is passed using links above display columns for the recordset on the page, with the parameter in each link set to the name of the database column specific to that display column. If the server behavior is configured to sort based on the value of the parameter, each link reloads the page and orders the results according to database column value specific to each. Select lists can be configured in a similar manner, with each entry providing the appropriate database column value to be used for the sort.

Access

The Sort server behavior is accessible in the following location:

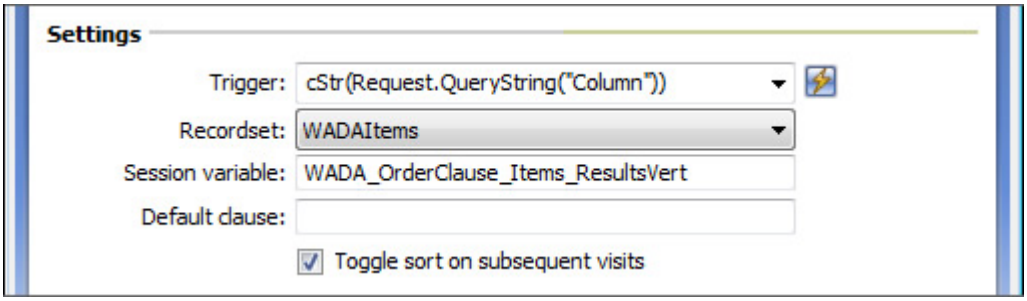
- *Server Behaviors panel > Plus icon (+) > DataAssist > Sort*

Configuration



Settings

Select the recordset on the page that is to be sorted. Specify the event on the page that triggers the sorting behavior. Specify a session variable on the page to maintain the query clause applied to the recordset for the current sort order. Set the default sort clause applied to the recordset.



Trigger: Refers to a URL parameter, session variable, or other dynamic data events. The existence of a value for the specified variable triggers the sort on a given page. A request object or session variable must be submitted or passed to the page that has the server behavior applied to it in order for the sort to execute. Available triggers are:

- **Any Form Post**: Any form posted to the current page that the server behavior is applied to triggers the data insert
- **Before Page Load**: The data insert is triggered when the current page that the server behavior is applied to is loaded.
- **On Form Submit**: Any form submitted on the same page that the server behavior is applied to trigger the data insert.
- **Button [button name] pressed**: The server behavior is triggered if the specified button on the page is clicked.
- **Dynamic Data**: Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.

Recordset: Select the recordset on the page that contains the data displayed that is to be sorted. The sort code applied updates the sort query associated to the selected recordset, ordering the results as specified in the *Sorting* pane of this interface (see below). All recordsets on the page are available for selection from the Recordset list.

Session variable: The session variable to store the current sort order upon future visits to the page in the current session. This is created by default for the given page, but the name can be changed specific to your development requirements. Also used with the *Toggle sort on subsequent visits* functionality to maintain and determine the current sort order so it can be reversed.

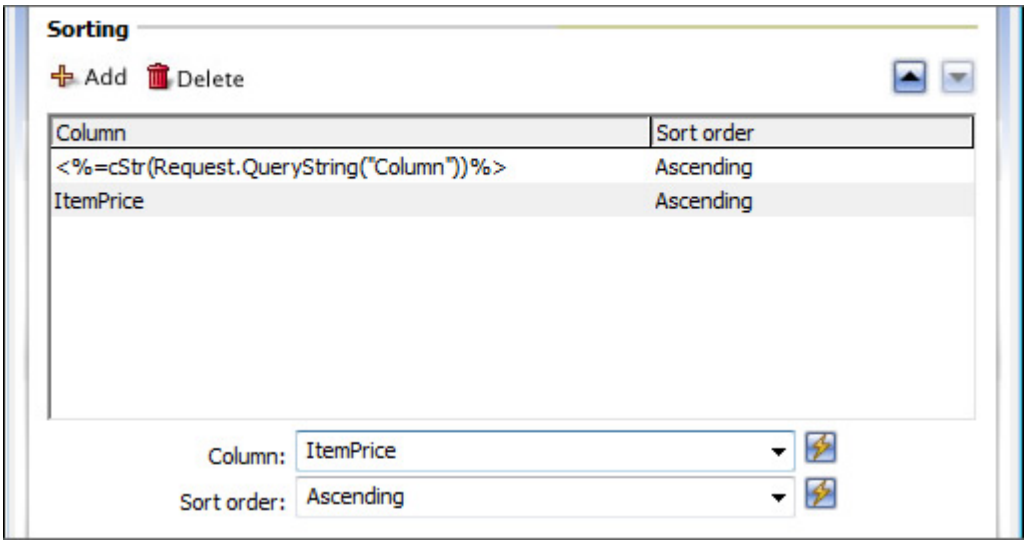
Default clause: Specify the default clause to use for ordering the results returned to the page on the first visit.

Toggle sort on subsequent visits: When checked, this reverses the order of the last set of results returned to the page. The current order is maintained in the session variable specified.

Sorting

In the panel, select the database column(s) used to perform the sort. If multiple columns are specified, the order of the columns in the panel is the order the sort criteria is applied to the recordset. Each column and sort order can be either chosen from a list or set with the bindings interface.

- Add : Click to add additional columns from the datasource to be sorted by.
- Delete : Click to remove a selected column rom the sort criteria.
- : Use the Up and Down buttons to change a selected column's position within the sort hierarchy.



Column: Specify a column directly from the list available for the selected recordset, or select a dynamic data binding that returns the column to be sorted by. Common examples are querystrings added to the page that can be set using links or select lists.

Sort order: Select whether the sort order for the specified column is ascending or descending. Can also be set dynamically.

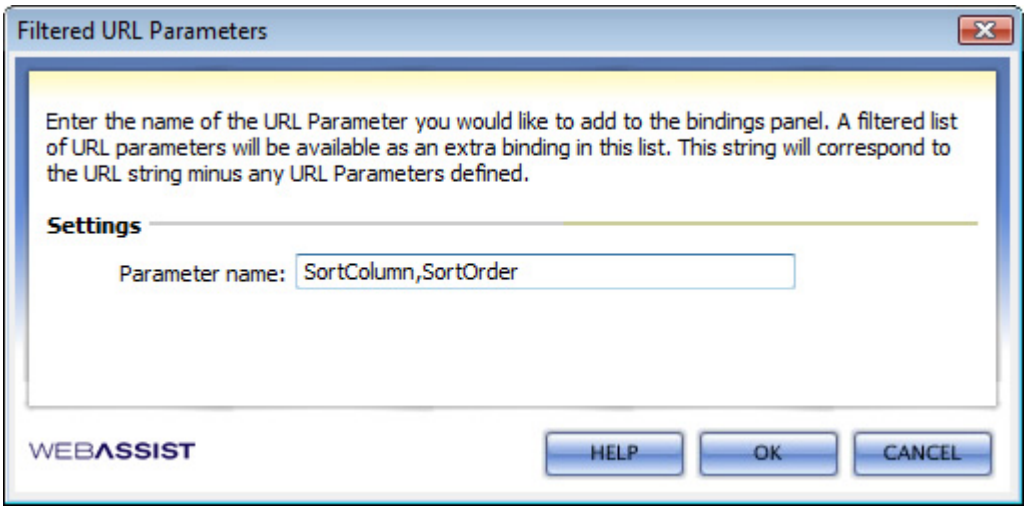
Querystring Binding

In order to use a querystring to trigger the sort on the page, it must be made available as a data binding on the page. This can be accomplished using the Querystring Binding feature included in the extension. If you wish to make multiple bindings available, specify them in the Parameter name field with each binding value separated by commas.

Access

Querystring bindings are accessible in the following location:

- *Bindings panel > Plus icon (+) > Querystring Binding*



Once the binding has been added to the page, it is accessible through the *Bindings* panel, as well as through the Dynamic data dialog found in the configuration interfaces within DataAssist and Dreamweaver.

In addition, a *Filtered Querystring* is included with the bindings you create. This binding can be appended to the URL used to trigger the sort, and works in conjunction with the paging functions available in the Dreamweaver *Recordset Navigation Bar*. It allows you to apply a sort to the records currently displayed on the page without returning to the first record. For example, if you are currently displaying 15 records on a page and have returned over 100 records in your results, you will need to page through results in order to view them. Unless you use the *Filtered Querystring* as part of your sort column criteria, any sort triggered on the page will return you to the first record for all sorted records, not just the current records displayed on the page.