SecurityAssist
Help Documentation

SecurityAssist Wizard

Authentication is the process by which website users are granted access to designated areas of a website by providing information verifying their identity. Identity information stored in a database is used to compare provided user information and determine if access should be allowed or not.

The SecurityAssist Wizard gives developers the power to quickly deploy a set of pages to manage the user registration and authentication experience for a website. As well, it works with your pre-defined database to allow the insertion, storage, and retrieval of user information specific to the authentication process. The pages created contain the following user authentication functionality: registration, user update, log in, and password retrieval.

This wizard provides extensive page customization features. Users can select from a number of pre-defined templates, styles, and fonts, or use an existing template in their Dreamweaver site as the format for the pages created by the wizard. Once the wizard is completed, the pages are opened so that users can continue to customize their pages using Dreamweaver.

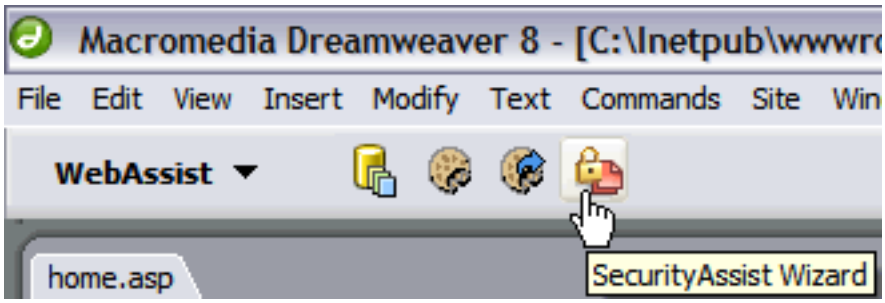
Note: The wizard is not re-entrant. Users will be able to edit functionality defined in the wizard using the Server Behaviors that are applied during the insertion process. However, the wizard may be used multiple times within a single site if you have multiple areas requiring distinct authentication functionality.

The following sections detail the steps necessary to complete SecurityAssist Wizard:

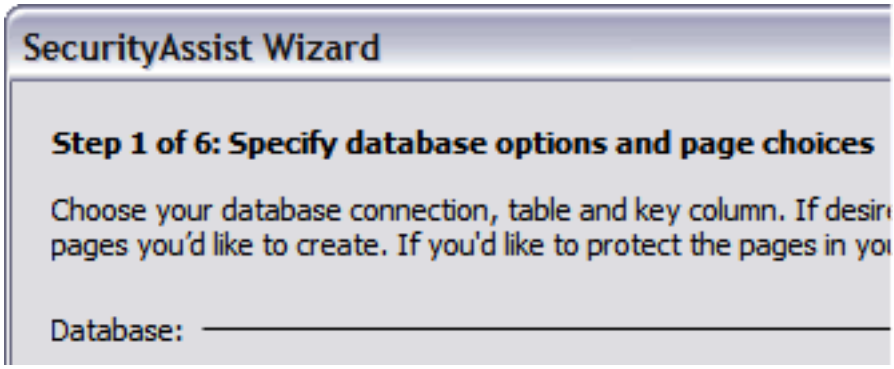
- Specify database options and page choices
- Select layout options
- Registration page options
- User update page options
- Log in page options
- Email password page options

To access the SecurityAssist Wizard:

1. In Dreamweaver, go to Insert panel > WebAssist > SecurityAssist Wizard



2. The SecurityAssist wizard opens:



3. Use the Back and Next buttons to move through the steps of the wizard.
4. Click the Finish button to generate the configured pages and open the Rules Manager.

Configuration details

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

Individual server behaviors are added to these pages by the wizard specific to authentication and emailing passwords. These server behaviors can be applied individually as well, and can be updated through the server behaviors panel at any time.

Dreamweaver's native Insert Record and Update Record server behaviors are applied to perform the insertion and update of user profiles in the database.

More information about the SecurityAssist server behaviors can be found through the following links:

- Authenticate User server behavior
- Email password server behavior

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 SecurityAssist 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">

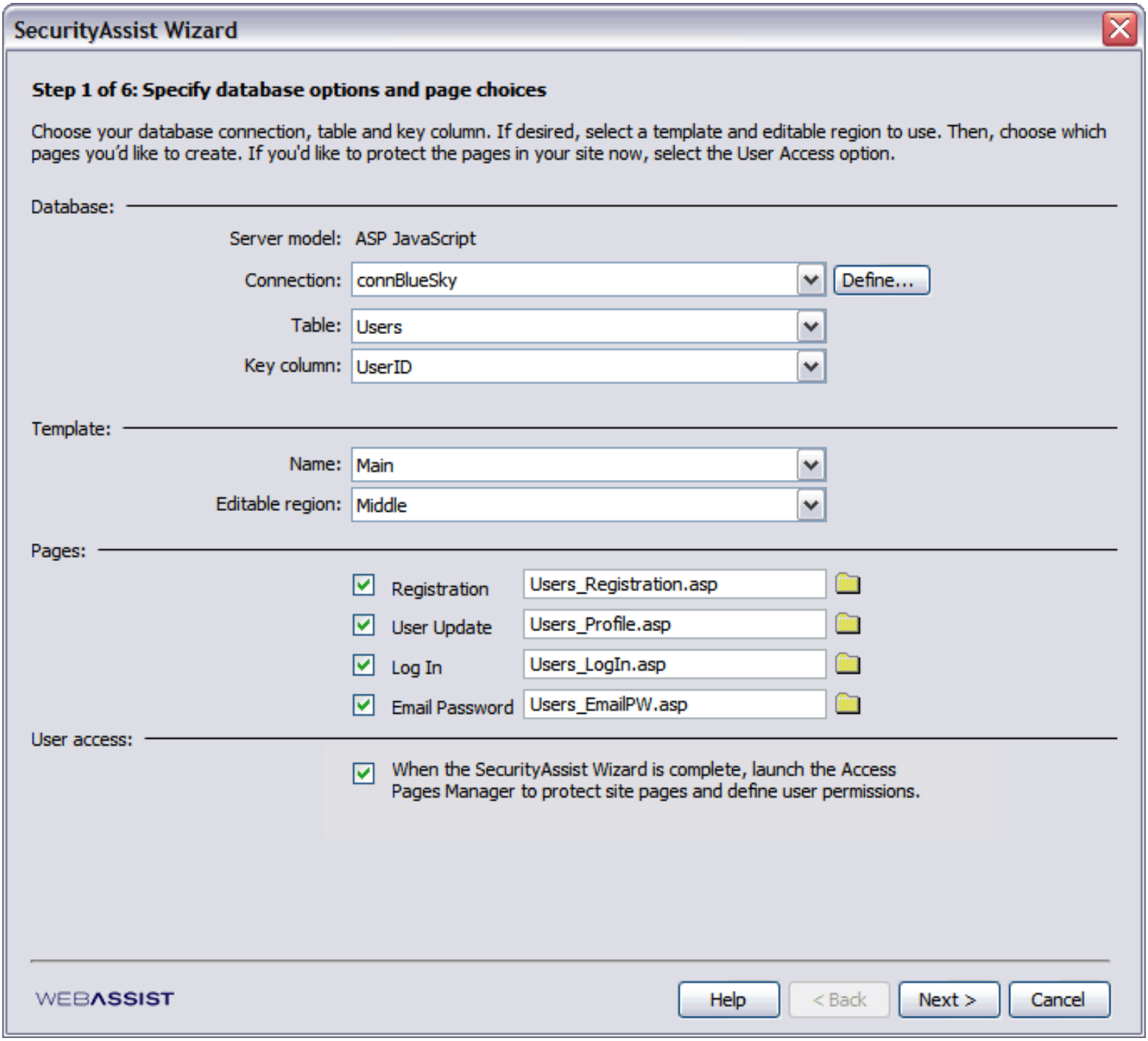
- Your ColdFusion Server must have session variables enabled. Log in to your ColdFusion Server Administrator and under *Server Settings*, click the *Memory Variables* link to navigate to that configuration section. Make sure that **Enable Session Variables** is checked, and click the *Submit Changes* 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.

Important: If you run the SecurityAssist wizard twice, and specify the same pages to be generated, the Authentication User Server behavior code footprint will be placed on the page twice. Please ensure that only one application of this server behavior is correctly applied to the page if this situation occurs.

Specify database options and page choices

The first step of the SecurityAssist wizard defines the connection to your database, selects a pre-defined template (if any) to apply to the pages created, and names the pages created for the page types selected.

You also have the option of automatically launching the Access Pages Manager upon completion of the wizard. This feature allows you to define which pages require authentication based on a set of defined access rules. Please see Access Pages Manager for more information.



Database: The following criteria is necessary to properly configure your database information 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 defined prior, or click the *Define* button to configure a database connection.

- **Table:** Selects the table within the chosen database that contains the user information used by the pages you are creating.
- **Key column:** Specifies the column with the database that is a unique key for the records within the selected table. This is typically an autonumber field, and is detected by default for the table selected.

Template: Specifies Dreamweaver templates available for use from the current site definition that can be applied to the pages created by the wizard:

- **Name:** Select the name of the template to be applied to your pages, or select none if you don't wish to use an existing template.
- **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: Four pages containing authentication functions are available for creation though the SecurityAssist wizard, and are listed below. Select the page types to be created, specify the name and location for the selected pages, and the next steps in the wizard configures the details specific to each:

- Registration: Adds a new user to the database, including any required registration information.
- User update: Allows a user to update information in their user profile that you define as editable.
- Log in: Verifies a user against the database and determines if they exist, and where to redirect based on login success or failure.
- Email password: Allows users to request that their password be emailed to the email address defined in their user profile.

User Access: When selected, launches the Access Pages Manager upon completion of the wizard and define which pages require authentication based on a set of defined access rules. Please see Access Pages Manager for more information.

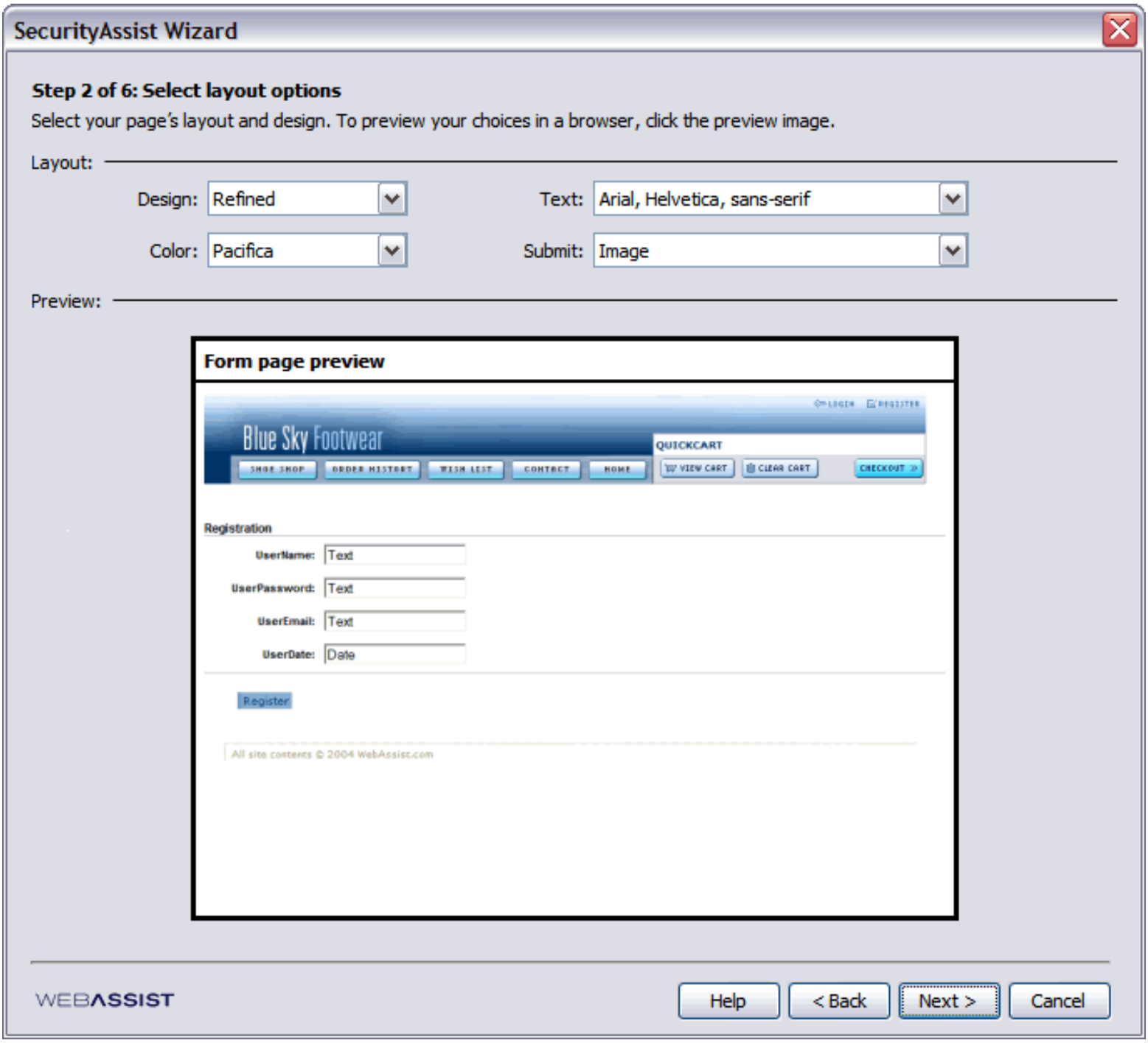
Related topics

- Features: SecurityAssist Wizard
- Features: Access Pages Manager

Select layout options

Step two of the wizard specifies the layout and design attributes for the pages created.

The Preview pane displays how the newly created pages will appear in your site based on your selections. Click the Preview pane to open the configured layout in a separate browser window and get a more detailed look.



Layout: Defines the overall look and feel of the pages:

- **Design:** Sets the layout structure of the content.
- **Text:** Sets the selected styles for the fonts included in the layout.
- **Color:** Sets the color scheme for the selected layout.
- **Submit:** Select whether a form button or an image is used for form submission.

Related topics

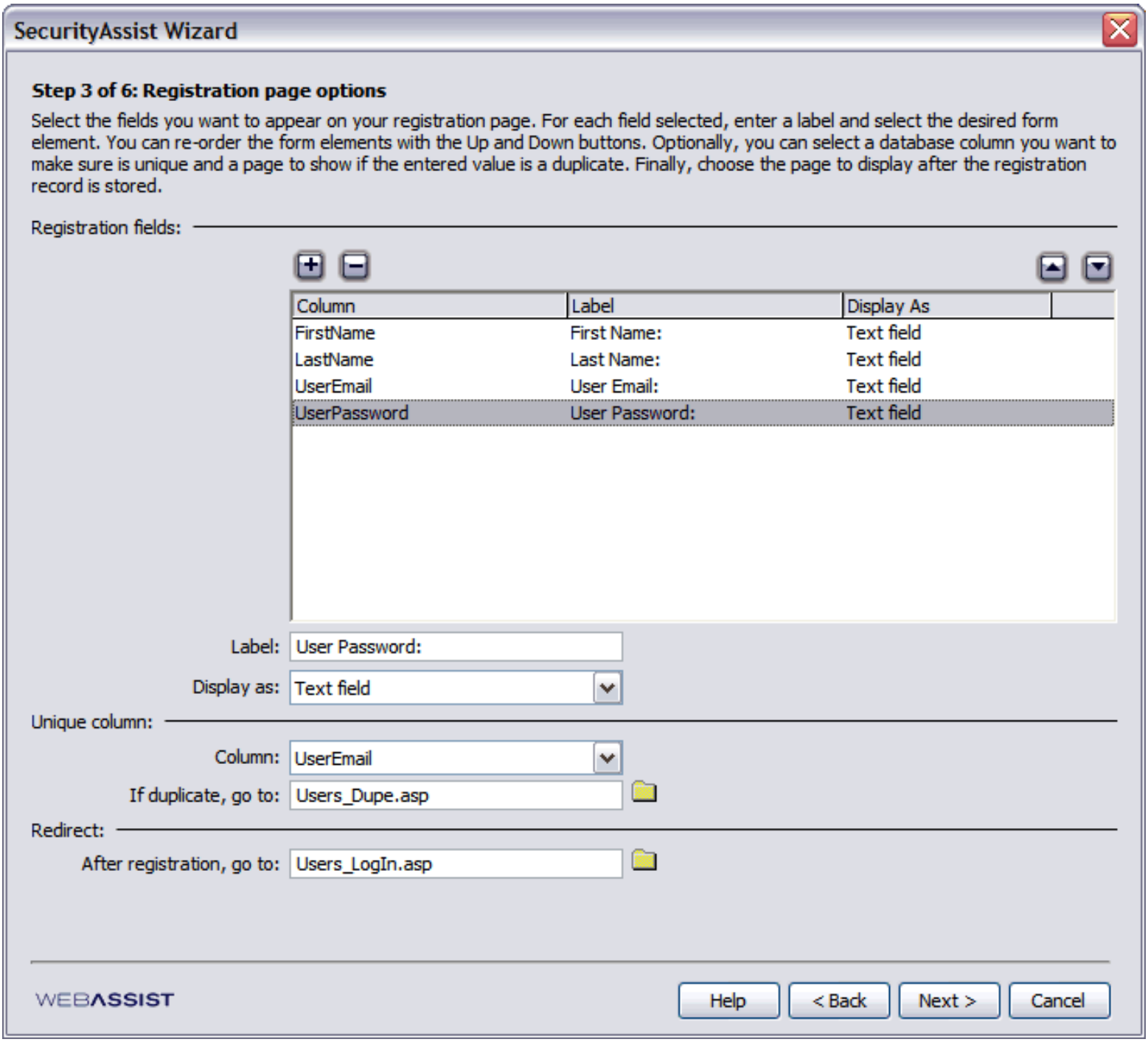
- Features: SecurityAssist Wizard

Registration page options

The registration page uses a form to gather the desired information from the user and create a new user record in the database. Form fields are mapped to database columns to insert data for the new record.

In the Unique Column area, specify the database column that identifies a unique value specified by the registrant. New records will not be created if the value entered by the user for the selected column already exists in the database. It is recommended that the email address be used as the unique identifier to work properly with the Email Password functionality.

Users are redirected to the specified location upon submitting the form and completing registration.



Registration fields: Configure the columns from the selected database table to be displayed in the registration form and to receive the corresponding data for the new user record. *Note*: It is appropriate to use validation in the form to ensure that the data entered matches the data type for the database column it will be inserted into.

- Use the Add (+) and Remove (-) buttons to specify and remove the database columns to be inserted into using the insert form. The Add button allows you to select from the remaining columns available from the table selected in step one. The Minus button

removes a selected column from the display pane.

- Use the Up and Down buttons to move a selected form field in the display to a different location in the given hierarchy. The fields will be ordered within the generated registration form according to this order.
- To modify a field, select it within the display pane. Once selected, you can modify the label and the type of form field for record information inserted in the field.
    - **Label:** Specifies the label text that modifies the form element.
    - **Display as:** Sets the type of form element used to enter/select user information. The types of form elements available are:
        - **Text field**
        - **Text area**
        - **Menu**
        - **Hidden field**
        - **Check box**
        - **Radio group**
        - **Password field**

          ***Note*:** Sophisticated form configuration using menus, radio groups, and checkboxes, is covered in more detail in the section Form Element Configuration in this help documentation.

Unique column: Specifies the column in the user table that should be unique for the information specified by the new user. The common usage scenario is to specify the column for the email address, which is recommended.

- **Column:** Selects the column for the unique user data from the list of available data columns.
- **If duplicate, go to:** Specify the redirect location that the user is sent to if a record exists containing the submitted value for the unique column. If left blank, the page will submit to itself. You may use site relative locations (i.e. ../../) to specify a file location or a full http:// address to redirect to another site.

Redirect: Specifies the redirect location upon successfully completing the registration form. You may use site relative locations (i.e. ../../) to specify a file location or a full http:// address to redirect to another site.

Related topics

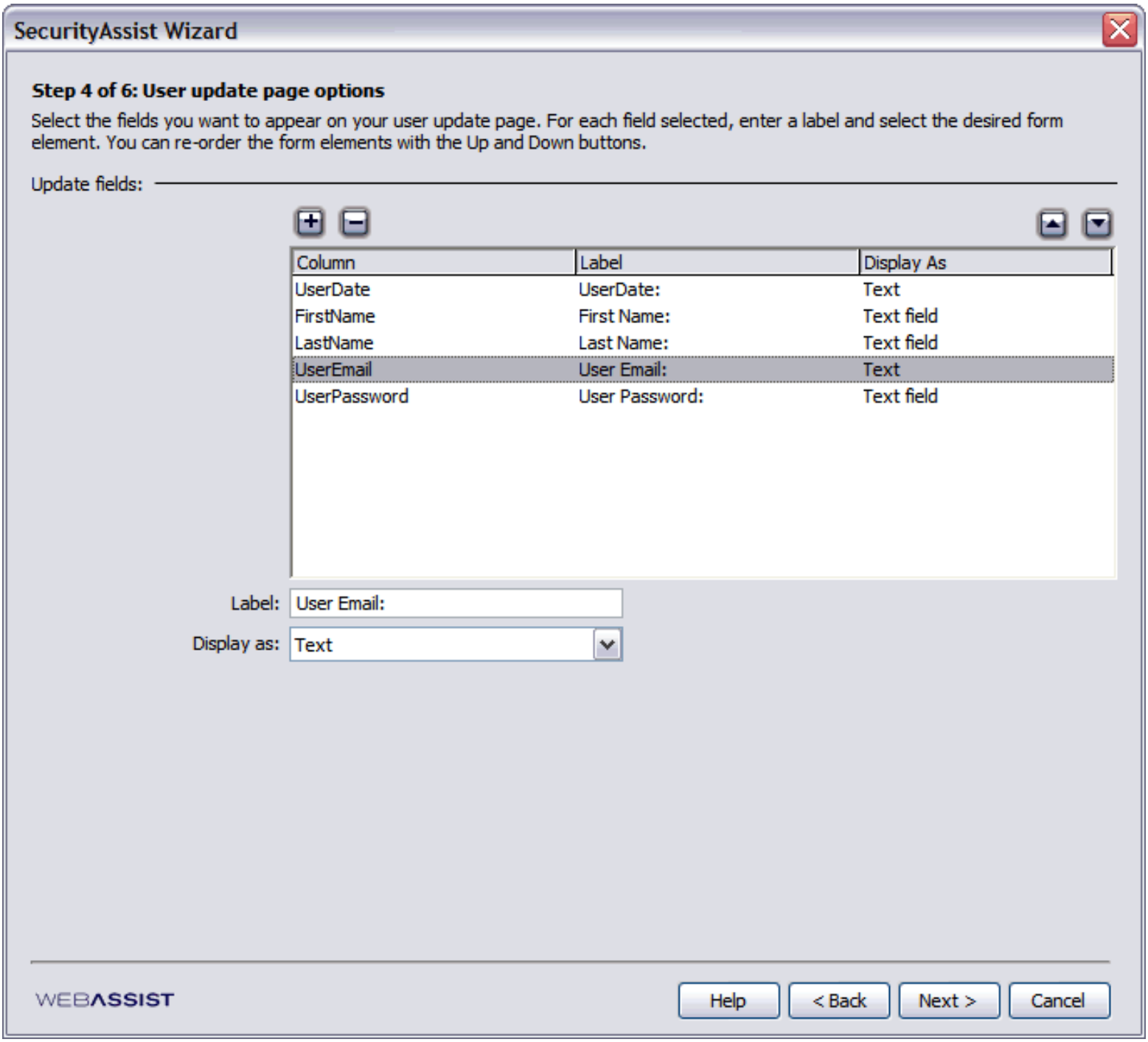
- Features: SecurityAssist Wizard
- Form Element Configuration

User update page options

Update pages display a given user's record in a form, allowing changes to be made and submitted to the database for specified fields.

Specify the fields that can be updated using the form and configure the order that they are displayed. Select the fields that are to be displayed on the page as text.

Note: The unique column specified in step 3 is automatically given a display type of Text. If you wish to make this field updateable, please note that duplicate error checking will not occur against the database for identical values in the database column being updated.



Search fields:

Use the Add (+) and Remove (-) buttons to add and delete database columns to be displayed or updated on the update form page. The Add button allows you to select from the remaining columns available from the table selected in step one. The Minus button removes a selected column from the display pane.

Use the Up and Down buttons to move a selected form field in the display to a different location in the given hierarchy. The fields will be

ordered within the generated update form according to this order.

To modify a field, select it within the display pane. Once selected, you can modify the label and the type of form field to be displayed in the form.

- **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**
  - **Hidden field**
  - **Check box**
  - **Radio group**
  - **Password field**
  - **Text**

    ***Note*:** Sophisticated form configuration using menus, radio groups, and checkboxes, is covered in more detail in the section Form Element Configuration in this help documentation

Related topics

- Features: SecurityAssist Wizard
- Form Element Configuration

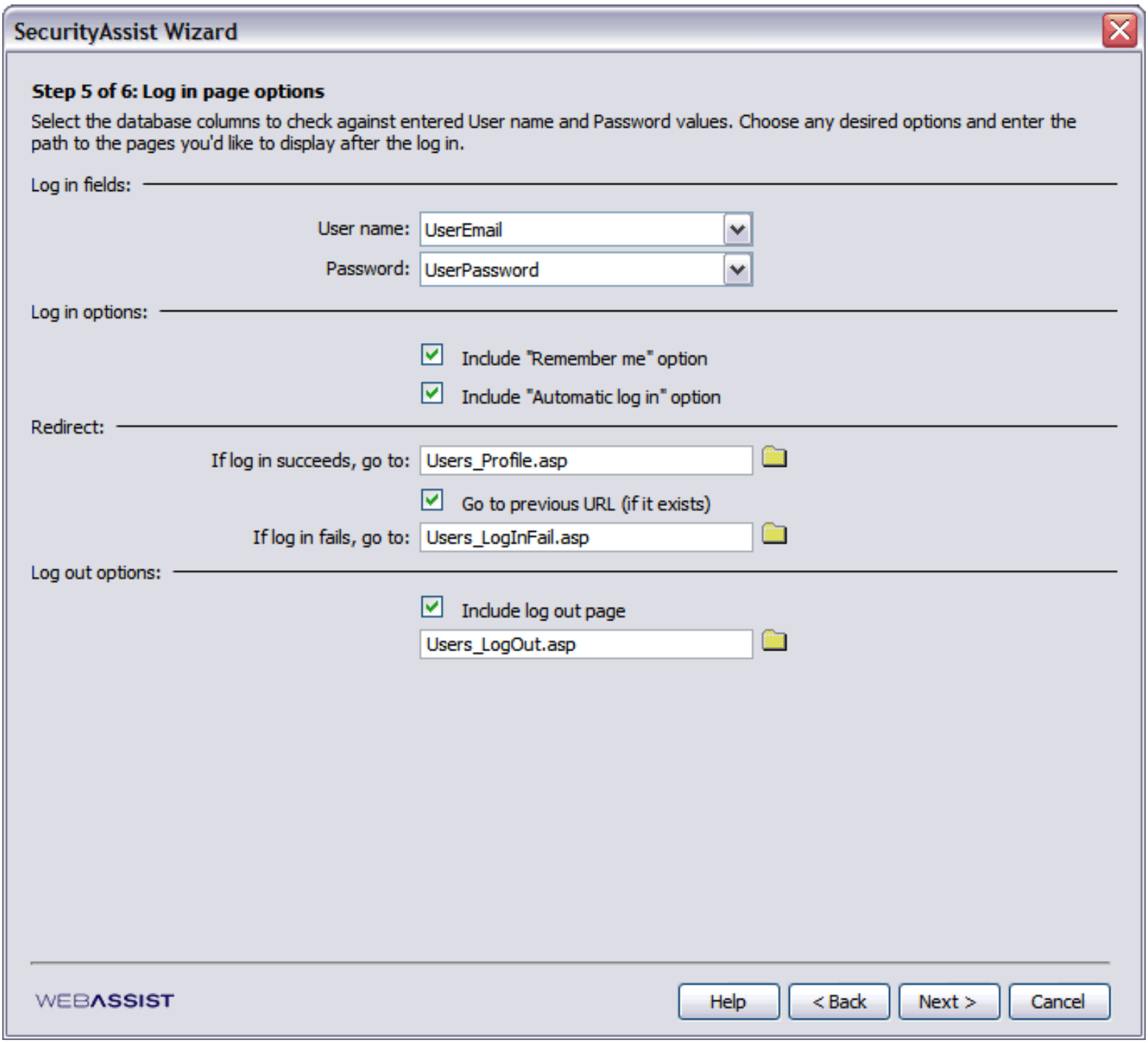
Log in page options

The log in page contains a form that asks for a user's login and password information.

Users can be offered the option to have this page pre-populated with their log in information by having this information stored locally in a cookie on their machine. As well, automatic login can also be offered based on the information stored in this cookie.

You can specify separate redirect locations based upon successful and unsuccessful attempts to login.

As well, a logout page can be created to serve as a destination page when users attempt to logout. Logout functionality is added to clearing all session variables if Cookies Toolkit (version 3.0.3 or higher), bundled with SecurityAssist, is installed.



Log in fields: Specifies the database columns specific to the user name and password for a user. The common implementation is to use the email address provided by the user as the username.

Log in options: Offers the options to the user to have this page pre-populated with their log in information on their next visit by storing it locally in a cookie. As well, automatic login can also be offered based on the information stored in this cookie.

Coldfusion users: Please note that for sites staged on *localhost*, the 'Remember me' option will not work correctly. When deployed to a remote server where the domain is specified, this functionality will work appropriately, but should be tested directly.

In addition, limitations with IE7 prevent this checkbox from working correctly in Coldfusion, due to a limitation in Cookies Toolkit where the *cfcookie* tag is getting a domain and path declared in the Set Cookie behavior. A temporary workaround for this is to remove path and domain attributes and values from the *Set Cookie* value. Currently this disbales seeing the Set Cookie Value behavior in the Behaviors panel after the edit, even though the code will still be present and functional on the page. This will be addressed in a future relase of Cookies Toolkit.

Note: Log in options are only available when the Cookies Toolkit extension, included in your SecurityAssist purchase, is installed.

Redirect: Specifies the redirect locations based on a successful or a failed log in attempt. Successful attempts can also be redirected to the last page viewed prior to attempting to login by checking the "Go to previous URL" box. You may use site relative locations (i.e. ../../) to specify a file location or a full http:// address to redirect to another site.

Log out options: Specifies whether a logout page is to be created, what its name should be, and the location of the file relative to the root of the site.

Related topics

- Features: SecurityAssist Wizard

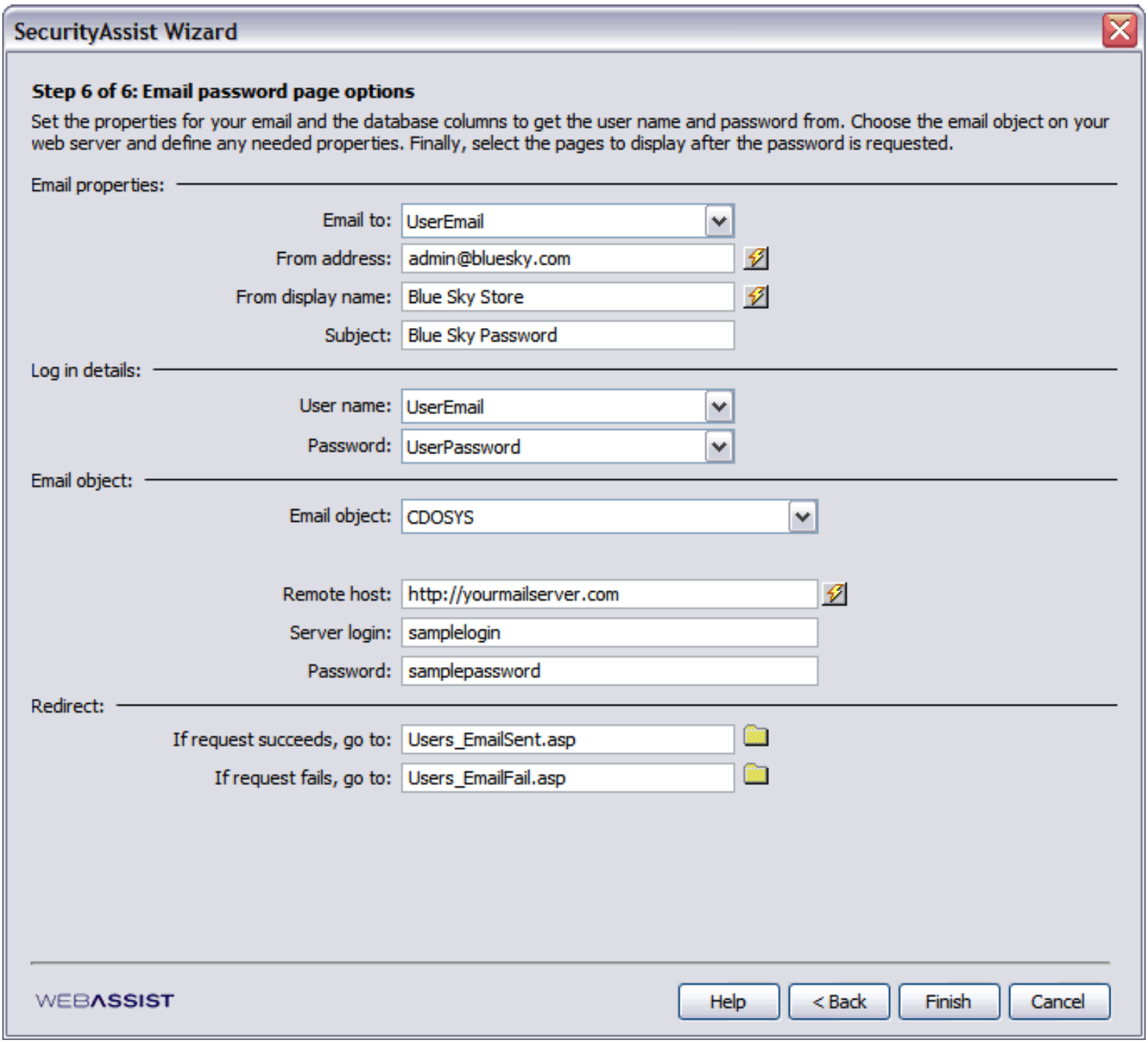
Email password page options

Creates a page that sends the password information for a user to the email address in their user profile.

The email sent is sent as plain text, and can have the display name, from address, and subject line customized.

Multiple email objects are supported, and distinct redirect locations can be specified depending on success or failure in locating the user info and sending the email.

The Email Password server behavior is applied to this page, and can be updated and configured in more detail specific to the email object selected upon completion of the wizard.



Email properties: specifies the information specific to the configuration and formatting of the email:

- **Email to**: Specifies the database column to be searched for the email address based on the list of available columns for the user table specified in step 1 of the wizard.

- **From address**: Specifies the email address displayed as the sender of the email. This information can be specified directly or retrieved dynamically using available data bindings.
- **From display name**: Specifies the display name for the sender of the email. This information can be retrieved dynamically using available data bindings.
- **Subject**: Specifies the subject line of the email

Log in details: The columns containing the information necessary to login that is sent to the user.

- **User name**: Specifies the database column to be referenced for the user name based on the list of available columns for the user table specified in step 1 of the wizard.
- **Password**: Specifies the database column to be referenced for the password based on the list of available columns for the user table specified in step 1 of the wizard

Email object: Select the email object used by your web server and website to send email. Several options are available based on the server language selected, and are detailed in the Email Password server behavior section. The most common configuration fields are specified in this wizard, but more detailed options are also available specific to each object through the Email Password server behavior applied to the page.

Redirect: Specifies the redirect locations based on a successful or a failed lookup for the user profile for the specified email. You may use site relative locations (i.e. ../../) to specify a file location or a full http:// address to redirect to another site.

Related topics

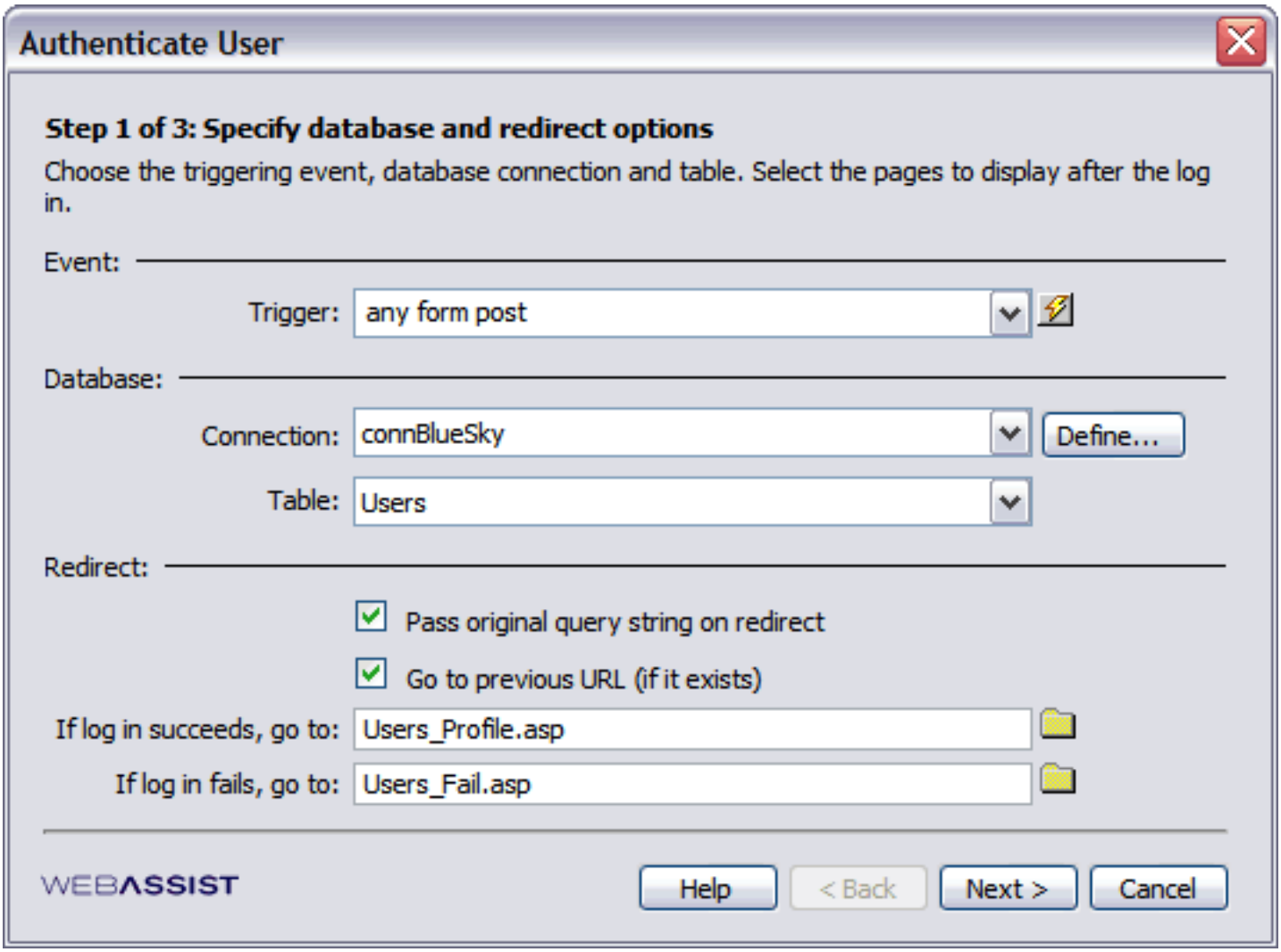
- Features: SecurityAssist Wizard
- Features: Email Password server behavior

Authenticate User server behavior

This server behavior authenticates a user against a specified table in the database, and redirects to a specified location based on a successful or unsuccessful user lookup.

It specifies the columns in the user table that are to be compared against, and the corresponding form elements or other data source used to submit that data for authentication.

As well, if authentication is successful, it allows you to define session variables that can maintain any data specific to that user based on the information in their user record, allowing you to use that information during their visit, and also determine what parts of the site they have access to in conjunction with the Access Rules Manager.



Event: Specify the page event or trigger used to authenticate the login information against the user table in the database:

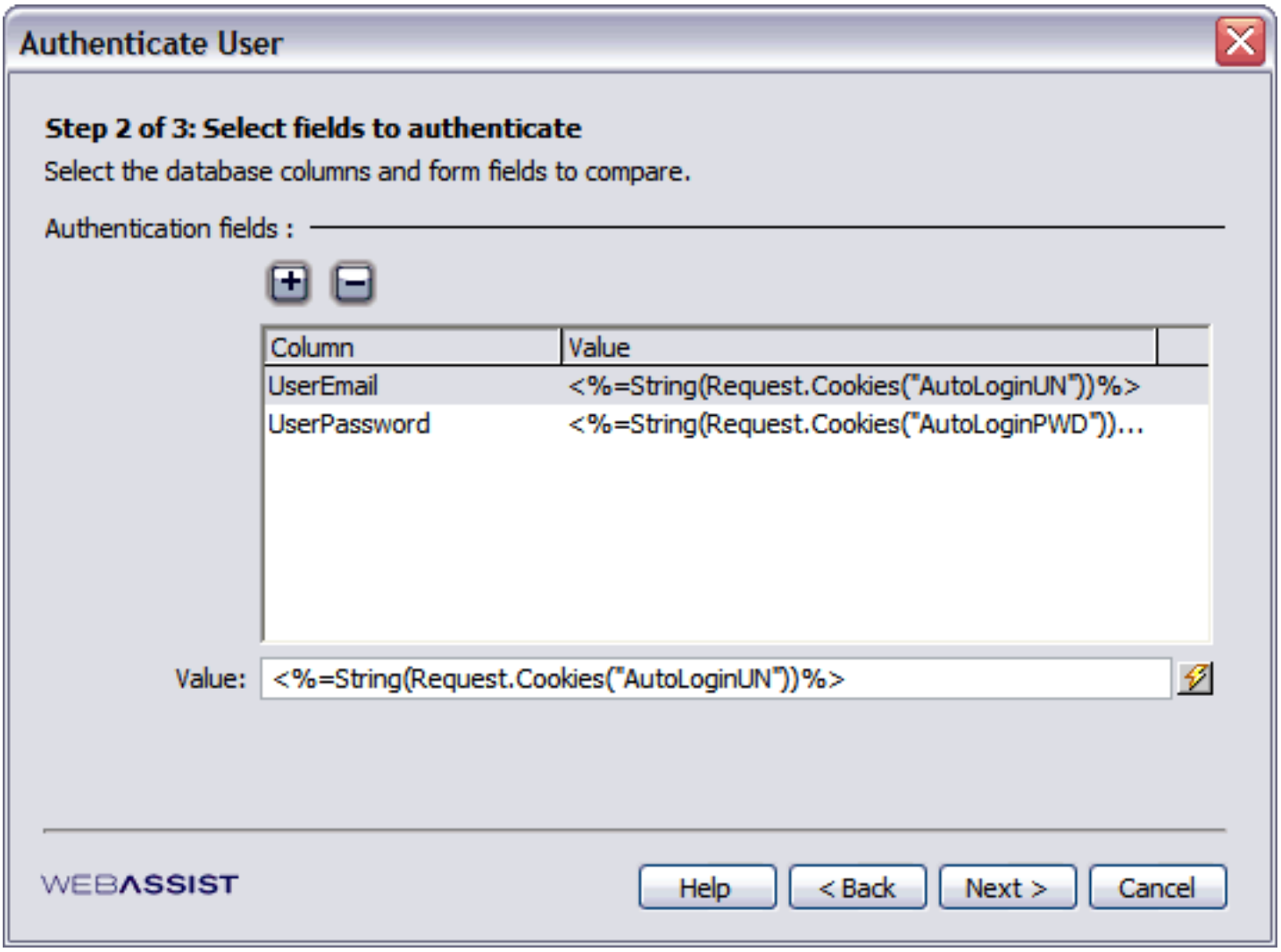
- **Trigger:** Available triggers are:
  - any form post
  - before page load
  - current page submit
  - a selected button on the page

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 to authenticate the user.

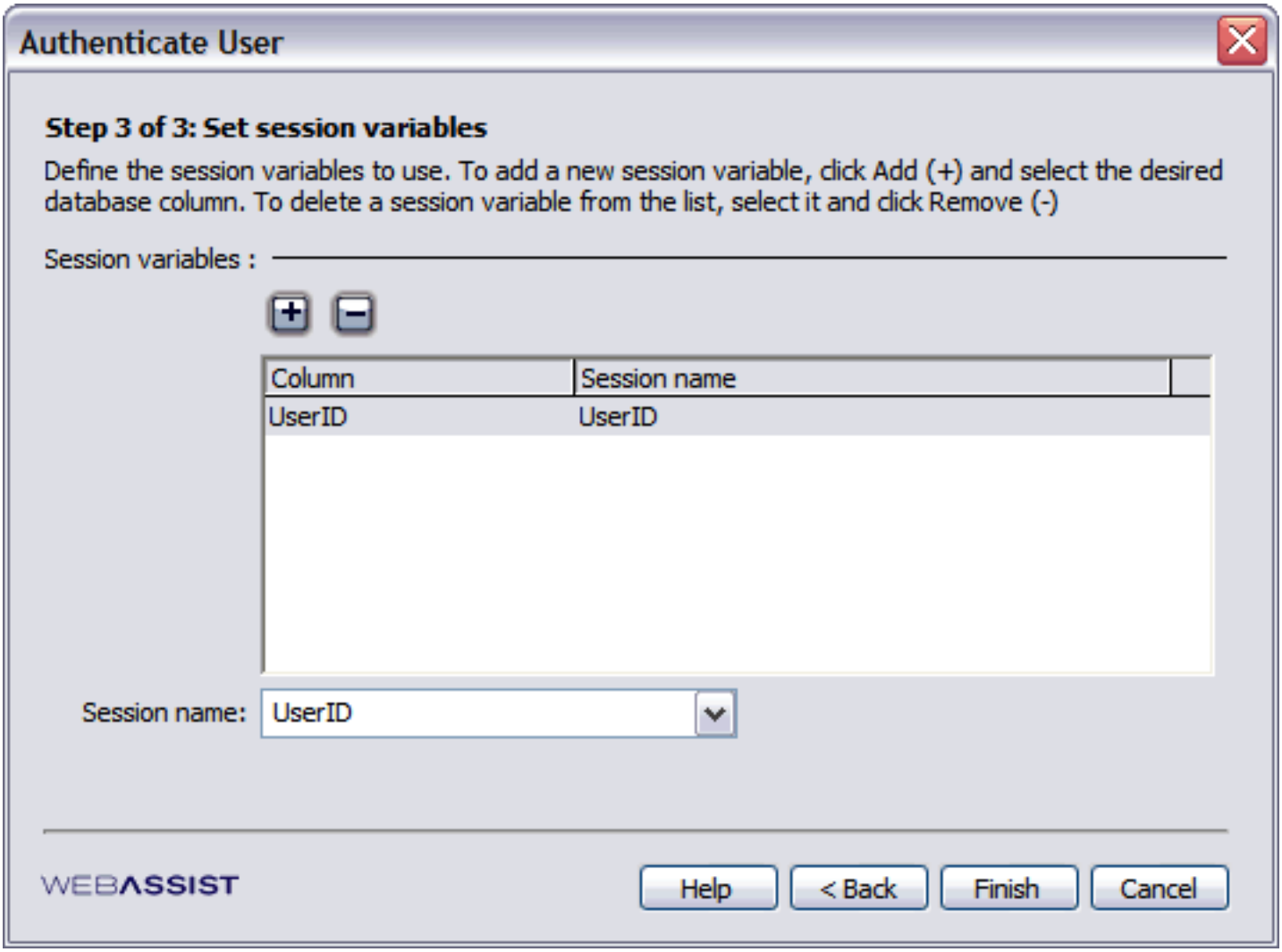
Redirect: Sets the actions that take place based on the authentication sequence.

- **Pass original querystring:** If a query is currently maintained in your session from prior to authentication, checking this box will pass that querystring on to the redirect page.
- **Go to previous URL:** When checked, successful authentication attempts are redirected to the last page viewed prior to attempting to login.
- **If login succeeds, go to:** The redirect location for a successful login attempt. You may use site relative locations (i.e. ../../) to specify a file location or a full http:// address to redirect to another site.
- **If login fails, go to:** The redirect location for a failed login attempt. You may use site relative locations (i.e. ../../) to specify a file location or a full http:// address to redirect to another site.



Map the form elements or other login data source to be authenticated to the fields database column that contains those values in the user table.

- **Column:** The database column being checked for authentication of a given user profile value.
- **Value**: Sets the source of the data to be authenticated for the selected column. Typically, this will be a form element, but you may also use server-side code to define the information to be authenticated.



Define the session variables that are maintained for comparison to rules (see Access Rules Manager for more info) to determine which pages the authenticated user has access to and does not. At a high level, the UserID is sufficient to maintain login, but you may wish to define other variables to maintain user information that might be important during the website experience as an authenticated user, or to determine different levels of access.

- **Column:** Sets the database column containing the user info for the session variable being defined.
- **Session name**: Specifies the name of the session variable to be maintained upon authentication for the selected database column.

Related topics

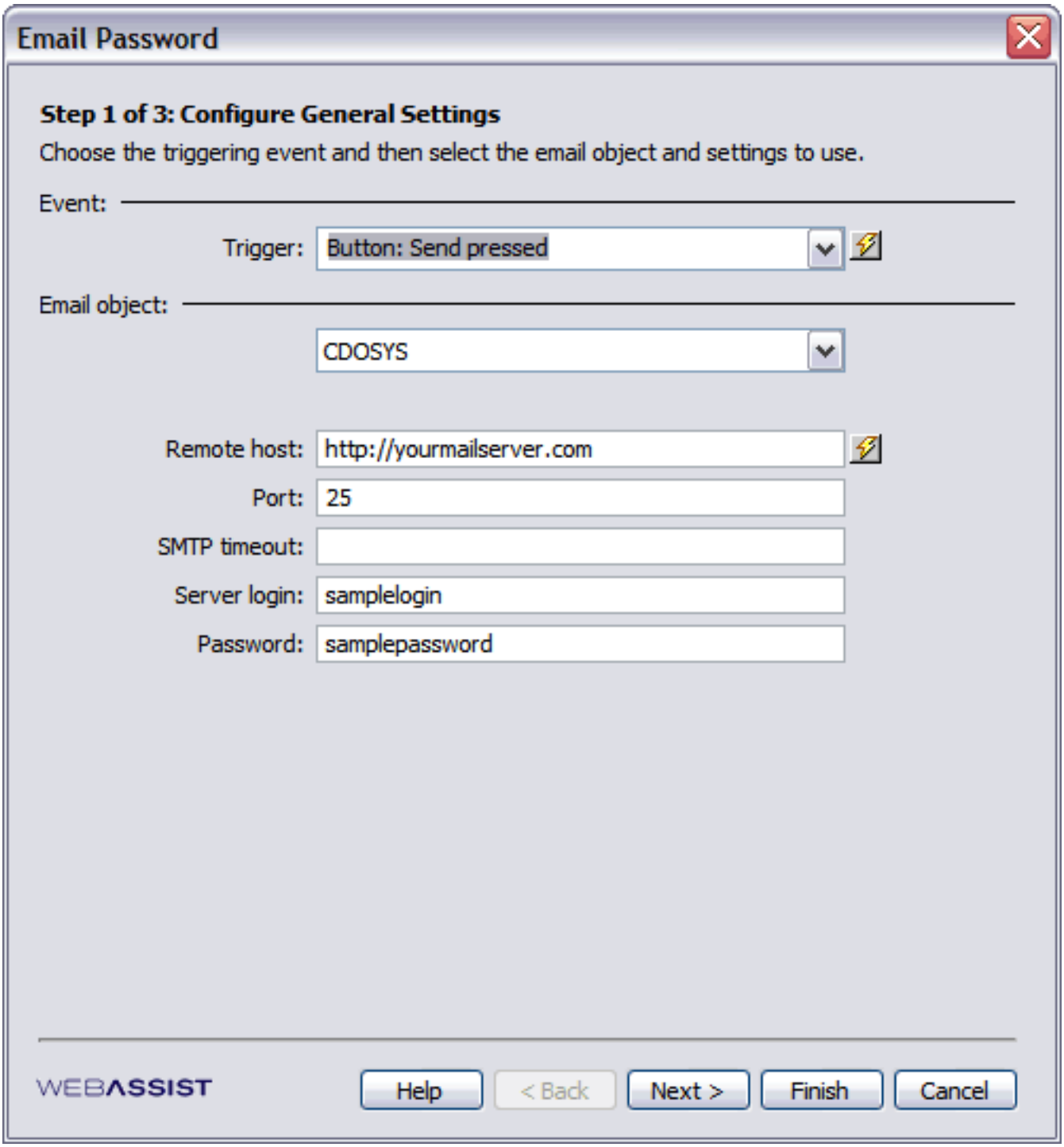
- Features: SecurityAssist Wizard
- Features: Access Rules Manager

Email Password server behavior

This server behavior, based on a a trigger on a page, performs a search on a specified table using submitted data to confirm the user's identity as the person associated with that user record. If it does not exist or the comparison fails, the page is redirected to a specified location. If it does exist, an email containing specific user information retrieved from selected database columns is sent to the email address associated to that user profile.

Several email objects are available for the server languages supported for this extension, and can be configured for use directly within the server behavior.

Only plain text format is available as a send format. The From address, Display name, Subject and Message body of the email can be customized specific to your needs.

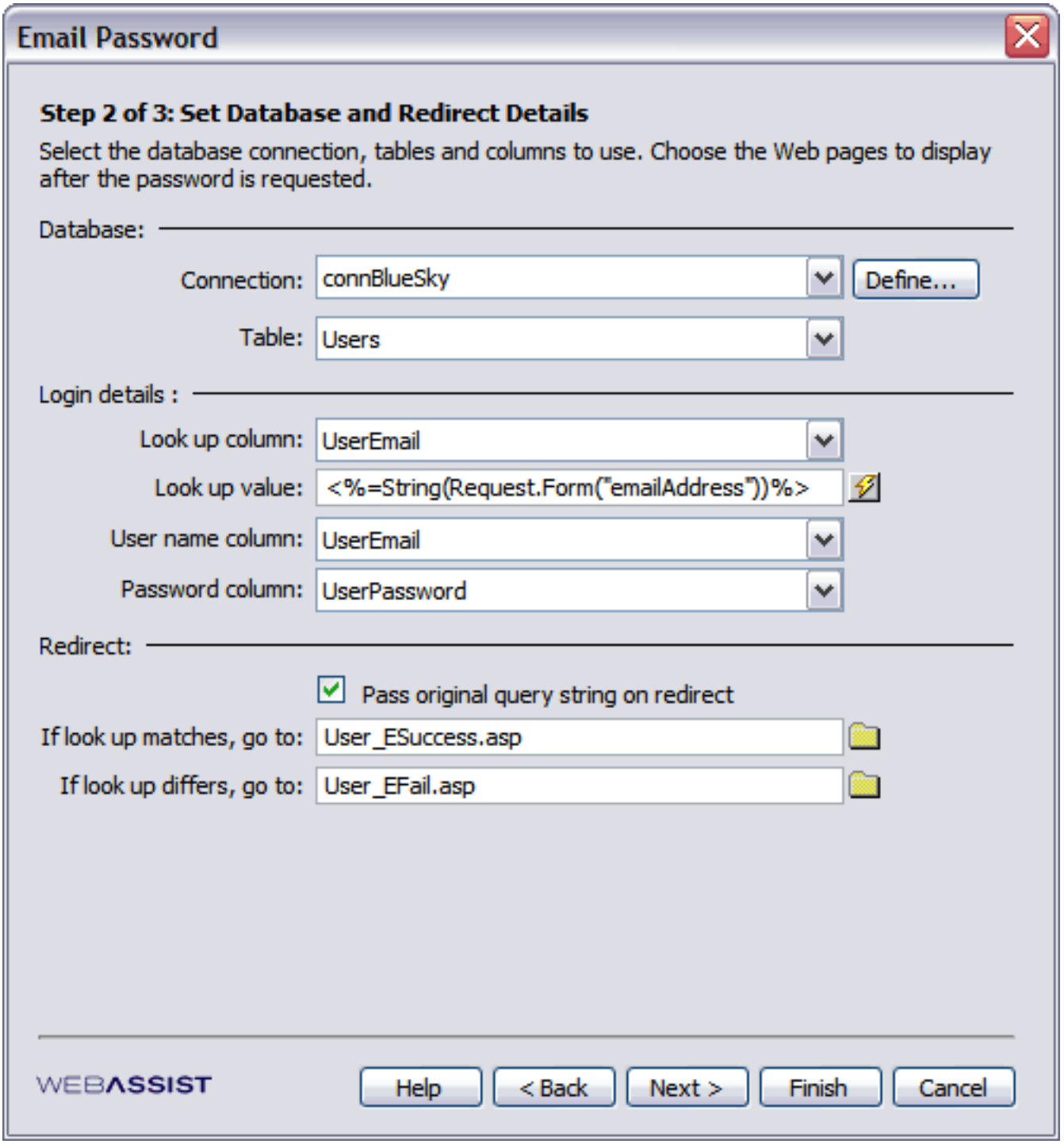


Event: Determine the page event or trigger used to perform the user record lookup and sending of the email.

- **Trigger:** Available triggers are:
  - any form post
  - before page load
  - current page submit
  - a selected button on the page

Email object: Specifies the email object used to perform the sending of the forgot password email. Configuration options specific to each available email object appear upon selection from the list and are detailed in the following sections:

- AspEmail (ASP)
- AspMail (ASP)
- CDONTS NewMail (ASP)
- CDOSYS (ASP)
- CFMAIL (Coldfusion)
- Mail for Linux (PHP)
- Mail for Windows (PHP)
- SMTPmail (ASP)
- w3 JMail (ASP)



Configure the database connection, the form element used to submit the comparison data, the user columns to be retrieved from the database, and the redirect locations based on success and failure.

Database: the following criteria is necessary to properly configure your database to be accessible to retrieve the necessary user data:

- **Connection:** Selects the database connection used to connect to the database to retrieve the user data. 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 user information to be retrieved.

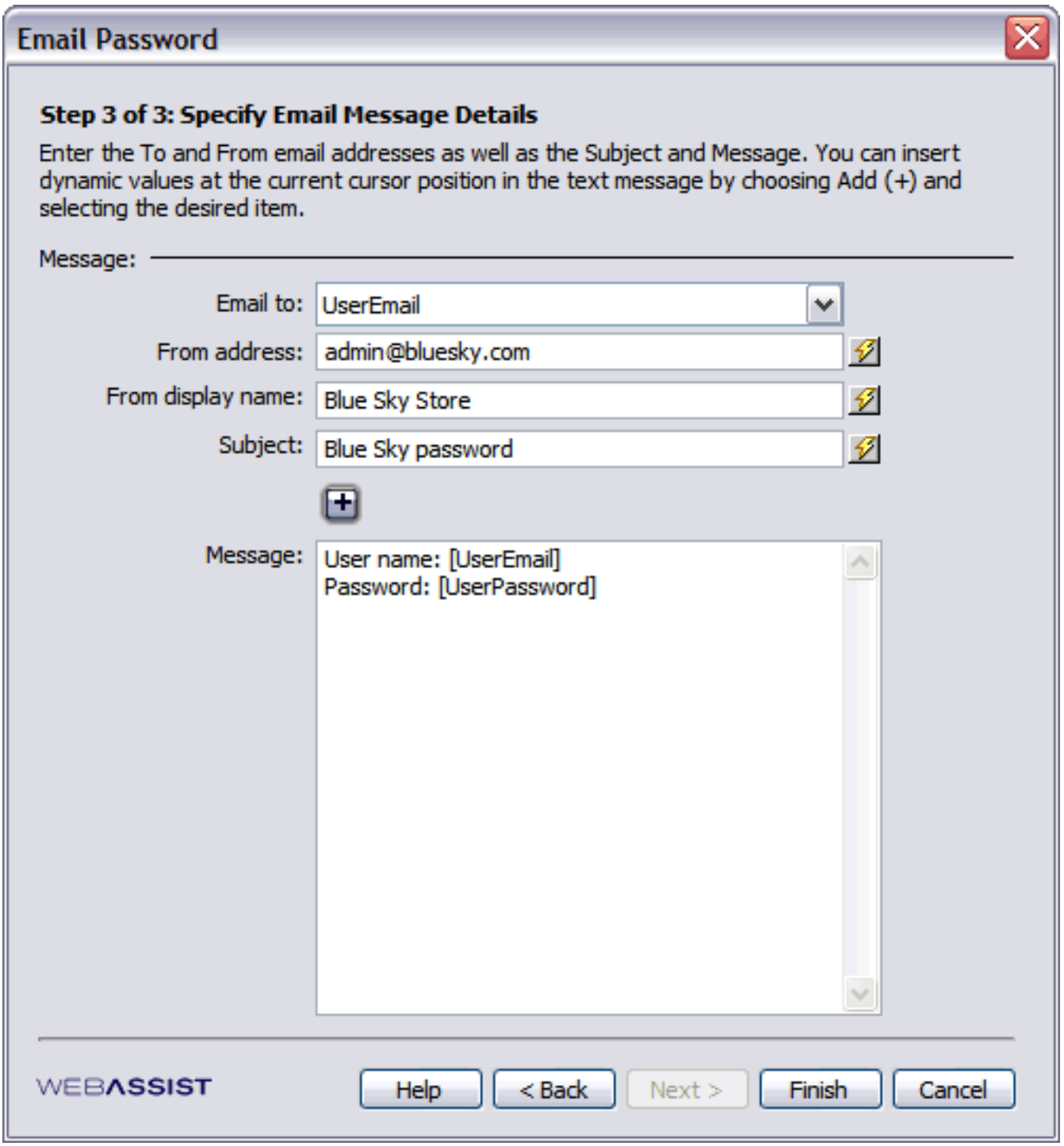
Log in details: the following criteria is necessary to perform the comparison of the submitted comparison data to a column in the user table in the database and retrieve the corresponding user login information:

- **Look up column**: the column in the user table that contains the data that confirms the identity of the person submitting the user data retrieval request.

- **Look up value**: the form element or other data binding used to specify the data to be confirmed in the database.
- **User name column**: the database column in the user table containing the user login.
- **Password column**: the database column in the user table containing the user password.

Redirect: Specifies the redirect locations based on a successful or a failed retrieval attempt.

- **Pass original querystring on redirect**: If a query is currently maintained in your session from prior to performing the password retrieval, checking this box will pass that querystring on to the redirect page.
- **If look up matches, go to**: The redirect location for a successful retrieval attempt. You may use site relative locations (i.e. ../../) to specify a file location or a full http:// address to redirect to another site.
- **If look up differs, go to**: The redirect location for a failed retrieval attempt. You may use site relative locations (i.e. ../../) to specify a file location or a full http:// address to redirect to another site.



Customize the content of your email, including the sender address, display name, subject, and message body.

Email to: Specifies the database column from the user table in the database containing the email address the user details should be sent to.

From address: Specifies the email address displayed as the sender of the email. This information can be specified directly or retrieved dynamically using available data bindings.

From display name: Specifies the display name for the sender of the email. This information can be specified directly or retrieved dynamically using available data bindings.

Subject: Specifies the subject line of the email. This information can be specified directly or retrieved dynamically using available data bindings.

Message: Configures the content of the message delivered to the recipient. You can insert values from the database for the given user record into the current cursor position in the text message by selecting them through the (+) button. Log in and password info is included by default.

Related topics

- Features: SecurityAssist Wizard

AspEmail

The following information is required for configuring your email application to use the AspEmail object:

Host

This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. <%="server_name"%>).

Port

This specifies the designated port on the computer which the email object will use to send email. The default is 25 and will work for most configurations.

Server Login and Password

These fields allow you to specify a username and password to access the SMTP server the mail object uses. Restricting SMTP access provides security against outside sources sending mail through your server. This feature is only available for the full version of the email object.

For more information and support for the AspEmail object, please refer to http://support.persits.com/

AspMail

The following information is required for configuring your email application to use the AspMail object:

Remote Host

This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. <%="server_name"%>).

For more information and support for the AspMail object, please refer to http://www.serverobjects.com/comp/Aspmail4.htm

CDONTS NewMail

The CDONTS NewMail object is the CDO mail object utiized by Internet Information Server (IIS). It requires no additional configuration through SecurityAssist other than specification.

For more information and support for the CDO Mail object, please refer to Microsoft's TechNet site.

CDOSYS

The following information is required for configuring your email application to use the CDOSYS object:

Remote Host

This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. <%="server_name"%>).

Port

This specifies the designated port on the computer which the email object will use to send email. The default is 25 and will work for most configurations.

SMTP Timeout

This sets the number of seconds which the email object should attempt to connect to a specified SMTP server to relay the generated email.

Server Login and Password

These fields allow you to specify a username and password to access the SMTP server for use by the mail object. Restricting SMTP access provides security against outside sources sending mail through your server.

For more information and support for the CDOSYS object, please refer to Microsoft's TechNet site.

CFMAIL

The following information is required for configuring your email application to use the CFMAIL object:

Mail server

This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

User name and Password

These fields allow you to specify a username and password to access the SMTP server for use by the mail object. Restricting SMTP access provides security against outside sources sending mail through your server.

Port

This specifies the designated port on the computer which the email object will use to send email.

Timeout

This sets the number of seconds which the email object should attempt to connect to a specified SMTP server to relay the generated email.

For more information and support for the CFMAIL object, please refer to Adobe's Coldfusion Developer Center, located at:

http://www.adobe.com/devnet/coldfusion/

Mail for Linux

The following information is required for configuring your email application to use the PHP Mail for Linux:

Server Name

This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. <%="server_name"%>).

Port

This specifies the designated port on the computer which the email object will use to send email. The default is 25 and will work for most configurations.

Return Path:

Sets this header to contain definitive information about the email address and route back to the message's originator. Not required.

Organization

Sets the value for the Organization header class of the email. Not required.

X-Mailer

Sets the header that describes the mailer program that was used to create this message. Not required.

Character Set

The character set to use for the body of the email. Default is iso-8859-1. Not required.

Mail for Windows

The following information is required for configuring your email application to use PHP's Mail for Windows:

Server name

This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. <%="server_name"%>).

Port

This specifies the designated port on the computer which the email object will use to send email. The default is 25 and will work for most configurations.

Return Path:

Sets this header to contain definitive information about the email address and route back to the message's originator. Not required.

Organization

Sets the value for the Organization header class of the email. Not required.

X-Mailer

Sets the header that describes the mailer program that was used to create this message. Not required.

Character Set

The character set to use for the body of the email. Default is iso-8859-1. Not required.

SMTPmail

The following information is required for configuring your email application to use the SMTPmail object:

Remote Host

This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. <%="server_name"%>).

SMTP Log

This provides the name of the log file to which actions by the email object can be written. The location can be specified by using the browse button if it will be on the current machine or may be pulled from a specified recordset using the dynamic data function (this field will accept server side code).

Server Login and Password

These fields allow you to specify a username and password to access the SMTP server for use by the mail object. Restricting SMTP access provides security against outside sources sending mail through your server.

For more information and support for the SMTPmail object, please refer to: http://support.softartisans.com/

w3 JMail

The following information is required for configuring your email application to use the w3 JMail object:

Mail Server

This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. <%="server_name"%>).

Server login and Password

These fields allow you to specify a username and password to access the SMTP server for use by the mail object. Restricting SMTP access provides security against outside sources sending mail through your server.

For more information on the w3 JMail object, please refer to http://www.dimac.net

Access Control

Access control is a powerful component of SecurityAssist, allowing you to determine the access that users have to information; from restricting access to regions on a single page or across multiple pages.

The following features allow you to manage and customize the access to content on your site:

- Access Rules Manager
- Access Groups Manager
- Show Region server behavior
- Page Access server behavior
- Access Pages Manager

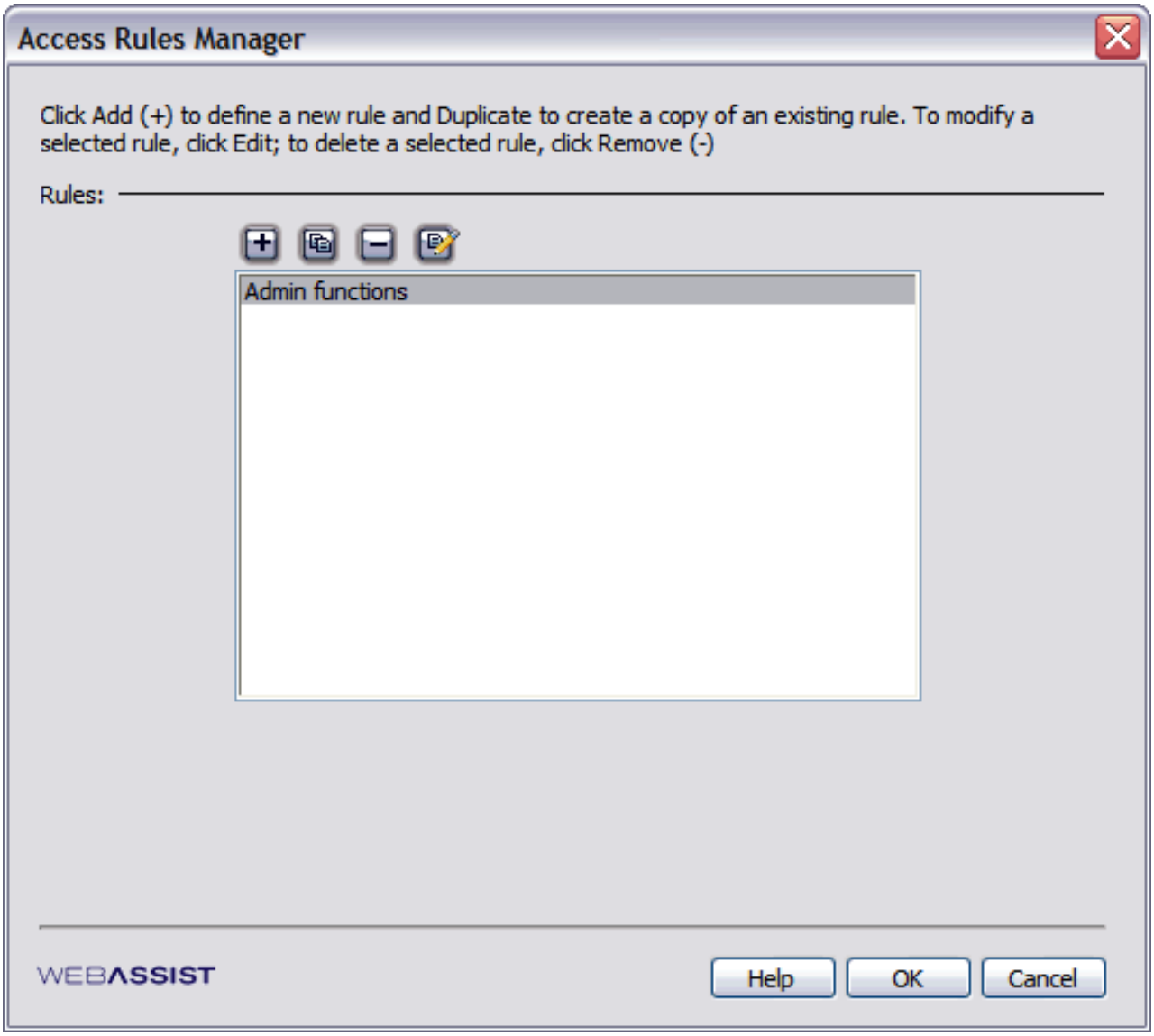
Access Rules Manager

Rules are the basis for determining accessing to content using SecurityAssist. Rules define the criteria that determines access, and are compared to available user information based on authentication to determine the level of access specific to a given user.

Once defined, rules are applied to regions within pages using data bindings or the Show Region server behavior, to individual pages using the Page Access server behavior, or to multiple pages using the Access Pages Manager.

Access

To access the Access Rules Manager, go to Modify > SecurityAssist > Access Pages Manager...



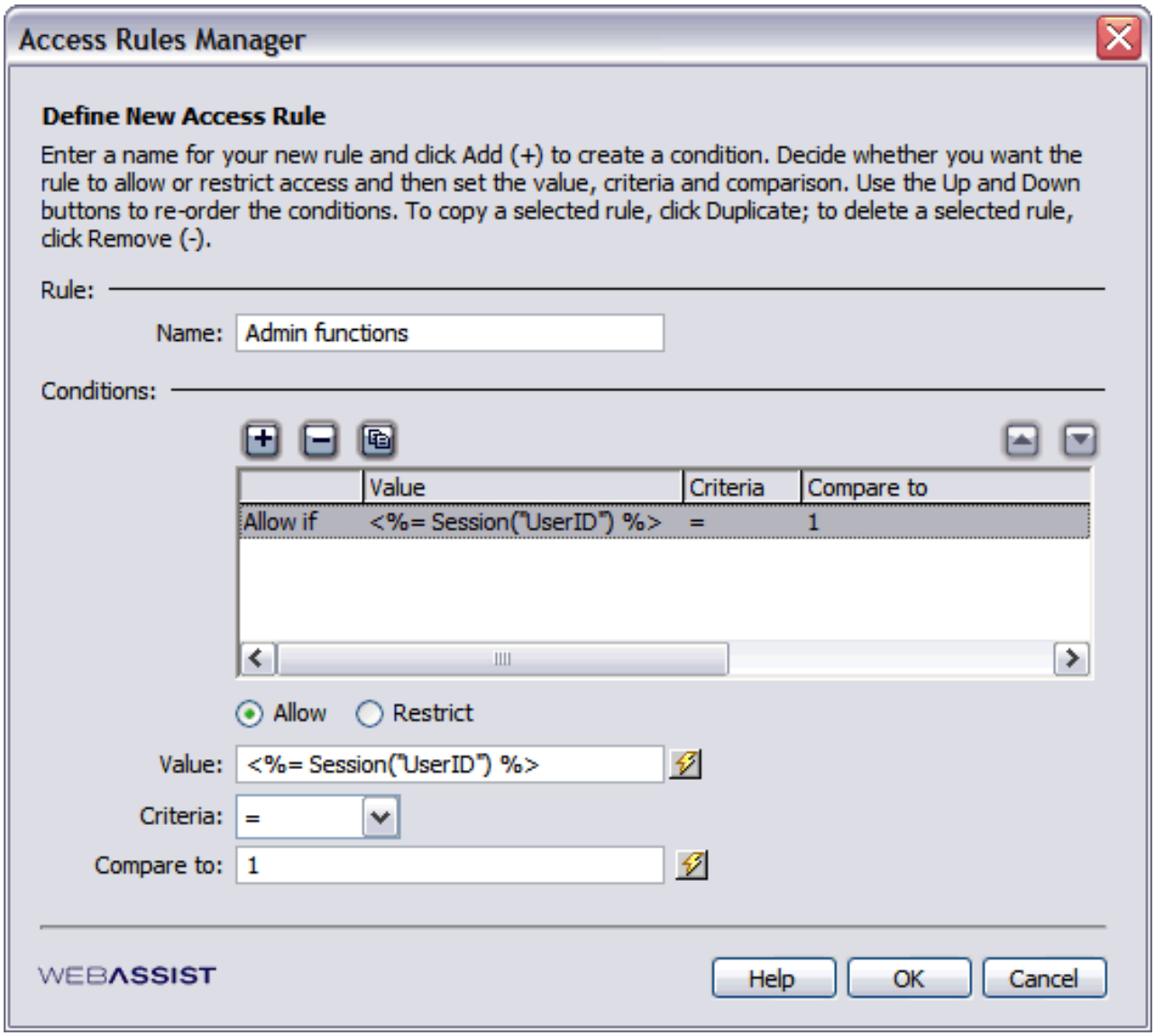
: Opens the Define New Access Rule user interface to specify a new rule.

: Duplicates an existing rule selected in the list of available rules.

: Removes a selected rule from the list of available rules.

: Opens the Edit Access Rule user interface to edit an existing rule selected in the list of available rules.

The following interface details the configuration of a new rule. The edit interface contains the same configuration parameters.



Name: Specifies the name of the rule.

Conditions: Specifies the conditions that must be satisfied for the rule to take effect.

: Adds a new condition to be configured to the list of defined conditions.

: Removes a selected condition from the list of defined conditions.

: Duplicates a selected condition to use as a template for a new condition.

: Moves a selected condition up one spot in its relative position within the list.

: Moves a selected condition down one spot in its relative position within the list.

Allow / Restrict: Determines if the rule is allowing or restricting access based on the condition defined. If both Allow and Restrict conditions are used, all Restrict conditions must be evaluated before the Allow condition. Only one true Allow statement is permitted per rule.

Value: For a selected condition, specifies the value that is evaluated. This value can be specified statically, or can be retrieved from a dynamic source using the available data bindings.

Criteria: Defines the criteria used to evaluate the specified value. Available options are:

- **=** : equals

- **< >** : does not equal
- **<** : less than
- **< =** : less than or equal to
- **>** : greater than
- **> =** : greater than or equal to
- **In group**: Evaluates if a user is part of a defined access group. Access groups are managed using the Access Groups Manager, and a session variable is used to specify the access level for a given user as it is defined in the database. More information on group management can be found in Access Groups Manager.

Compare to: Specifies the standard that the value field is evaluated against to determine if the criteria is met or not. This can be a static value or a dynamic value retrieved from the available data bindings. When In group criteria is selected, all defined groups are listed to select from for comparison and a Groups Manager button appears; select the Groups Manager button to add a new group or modify the existing ones.

Related topics

- Features: Access Groups Manager
- Features: Show Region server behavior
- Features: Page Access server behavior
- Features: Access Pages Manager

Access Groups Manager

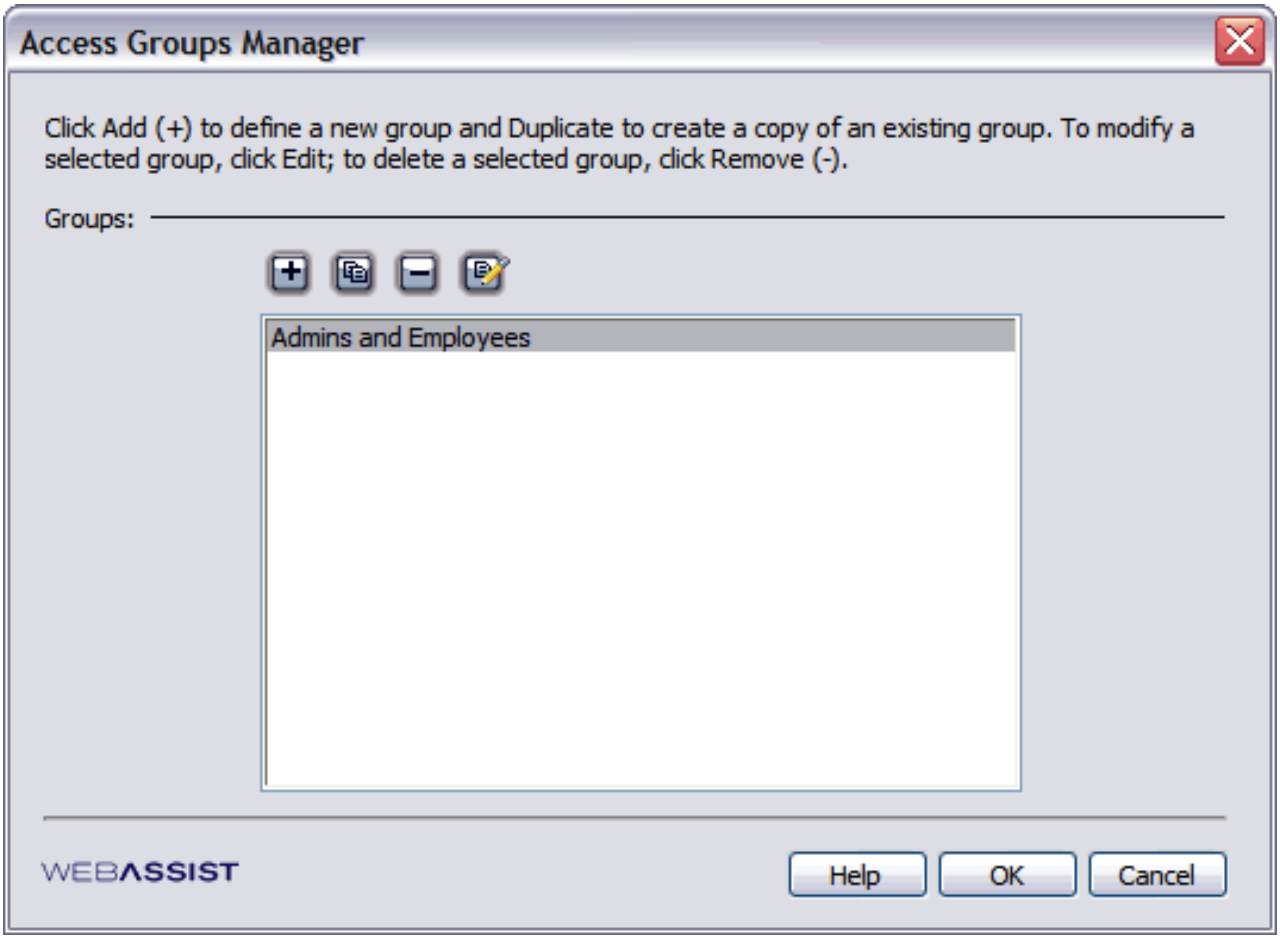
Access Groups are a method of organizing users according to access level criteria. By defining varying levels of access, regions of a site can be configured specific to the needs of a variety of access criteria.

To organize access groups, the members of a group are defined when configuring the group, and member information can be validated against specified values in the database stored relative to the user. That way, the access level for a given user can be set in the session, and used to determine access levels during the duration that they are logged in to the website.

The way access is determined is by referencing groups in the access criteria for specified access rules for the site. See Access Rules Manager for more information on defining rules and referencing groups in rule definitions.

Access

To access the Access Groups Manager, go to Modify > SecurityAssist > Access Groups Manager...



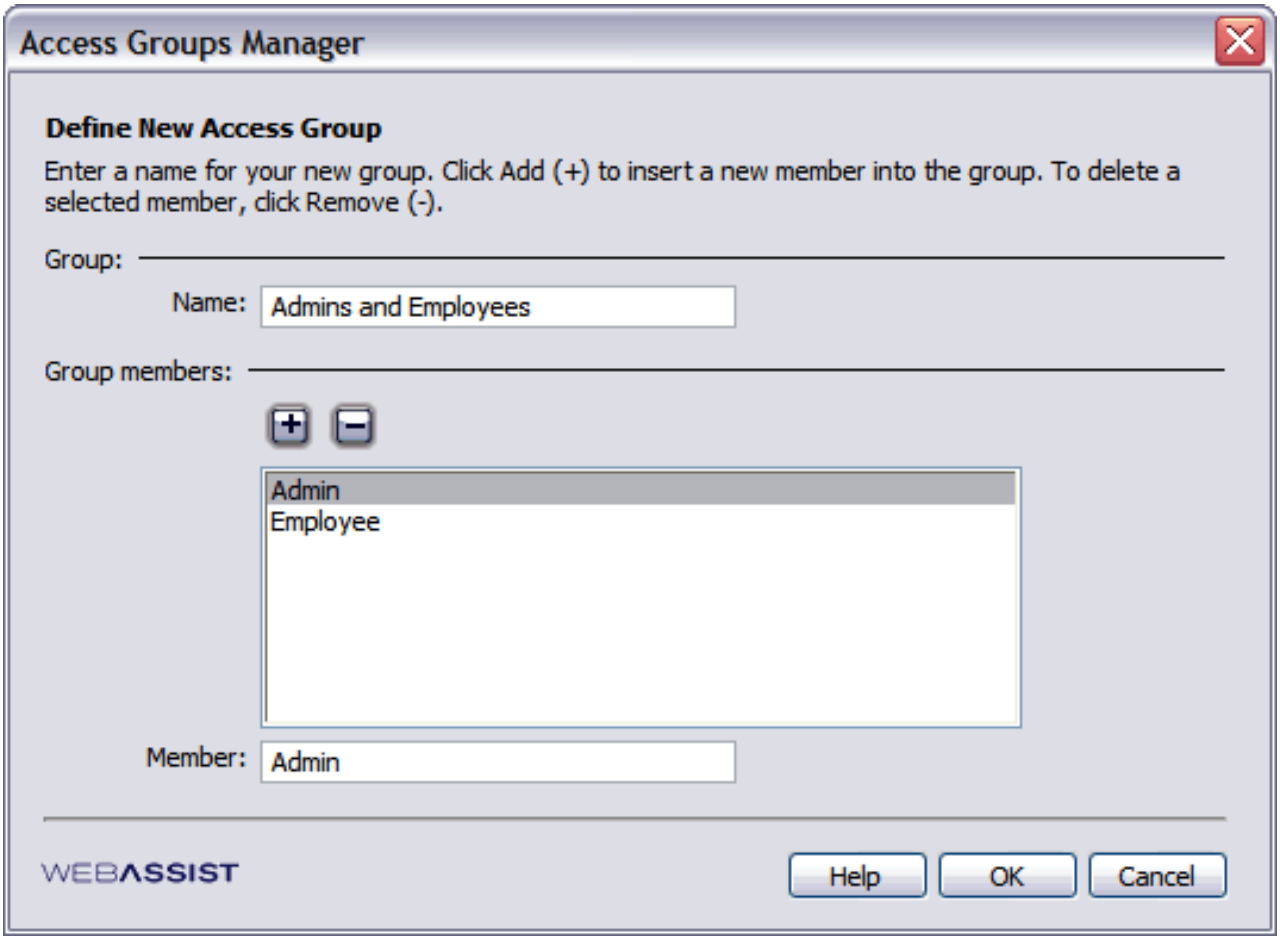
: Opens the Define New Access Group user interface to specify a new rule.

: Duplicates an existing rule selected in the list of available rules.

: Removes a selected rule from the list of available rules.

: Opens the Edit Access Rule user interface to edit an existing rule selected in the list of available rules.

The following interface details the configuration of a new access group. The edit interface contains the same configuration parameters.



Name: Specifies the name of the access group.

Group members: Details the member criteria to include a user in the group.

 : Adds a new group to the list of defined groups.

 : Removes a selected group from the list of defined groups.

Related topics

- Features: Access Rules Manager

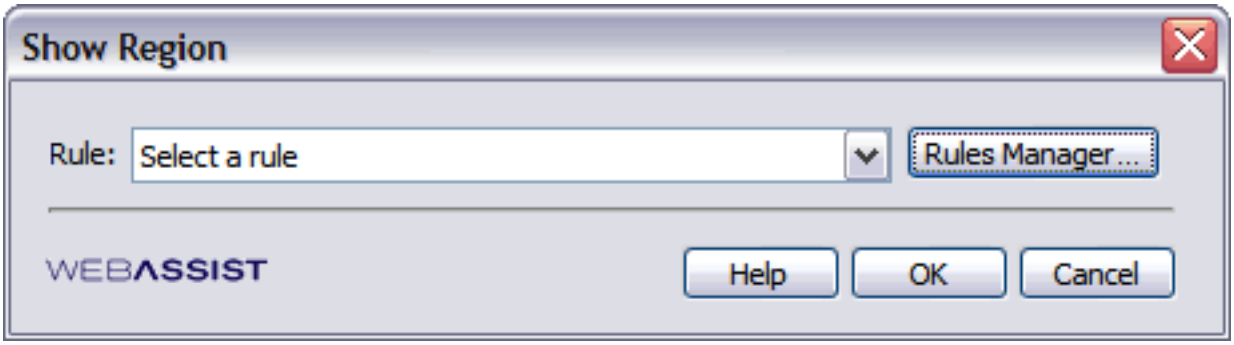
Show Region server behavior

This server behavior is applied against a display region on the page and checks against a rule configured using the Access Rules Manager to determine if the current user logged in has sufficient access rights to view that content. If so, the region is displayed; if not, it is not included when the page is rendered by the server and sent to the end user.

Access

To access the Show Region server behavior:

- Select the display region to be managed
- Go to the Server Behaviors panel, click the Plus button (+), and select SecurityAssist > Show Region



Rule: Specifies the rule that determines if the highlighted region is to be displayed or not.

Rules manager: Opens the Access Rules Manager to administer the rules configured for the current site.

Related topics

- Features: Access Rules Manager

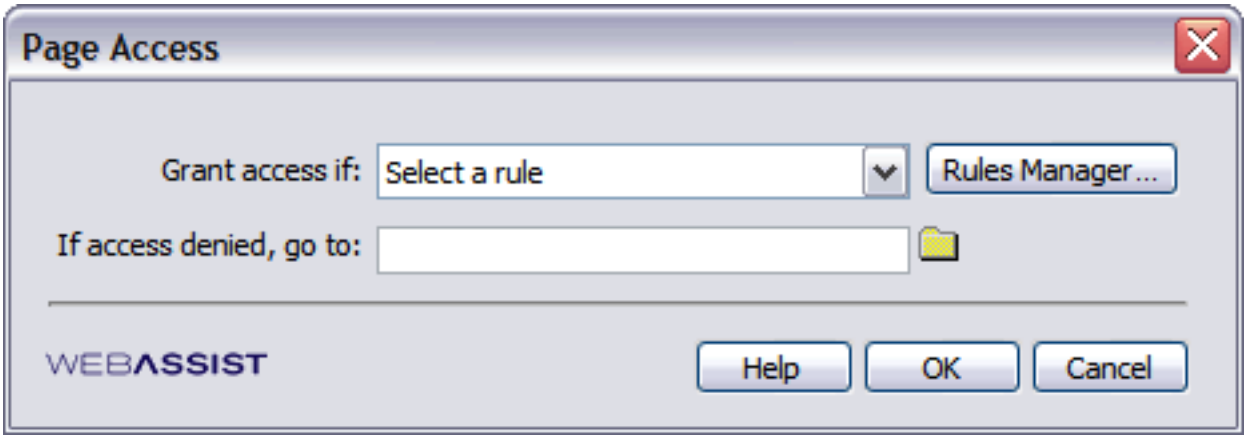
Page Access server behavior

This server behavior is applied to a page and determines if the page is to be loaded or redirected to a specified location based on satisfying a specified rule.

Access

To access the Page Access server behavior:

- Select the display region to be managed
- Go to the Server Behaviors panel, click the Plus button (+), and select SecurityAssist > Page Access



Grant access if: Specifies the applied access rule. Each defined rule is presented with it's inverse. For example, if you define a rule Administrator, both Administrator and Not Administrator are available in the list.

Rules manager: Opens the Access Rules Manager to administer the rules configured for the current site.

If access denied, go to: Specifies the redirect location if the user fails to meet the authentication criteria in the specified rule. You may use site relative locations (i.e. ../../) to specify a file location or a full http:// address to redirect to another site.

Related topics

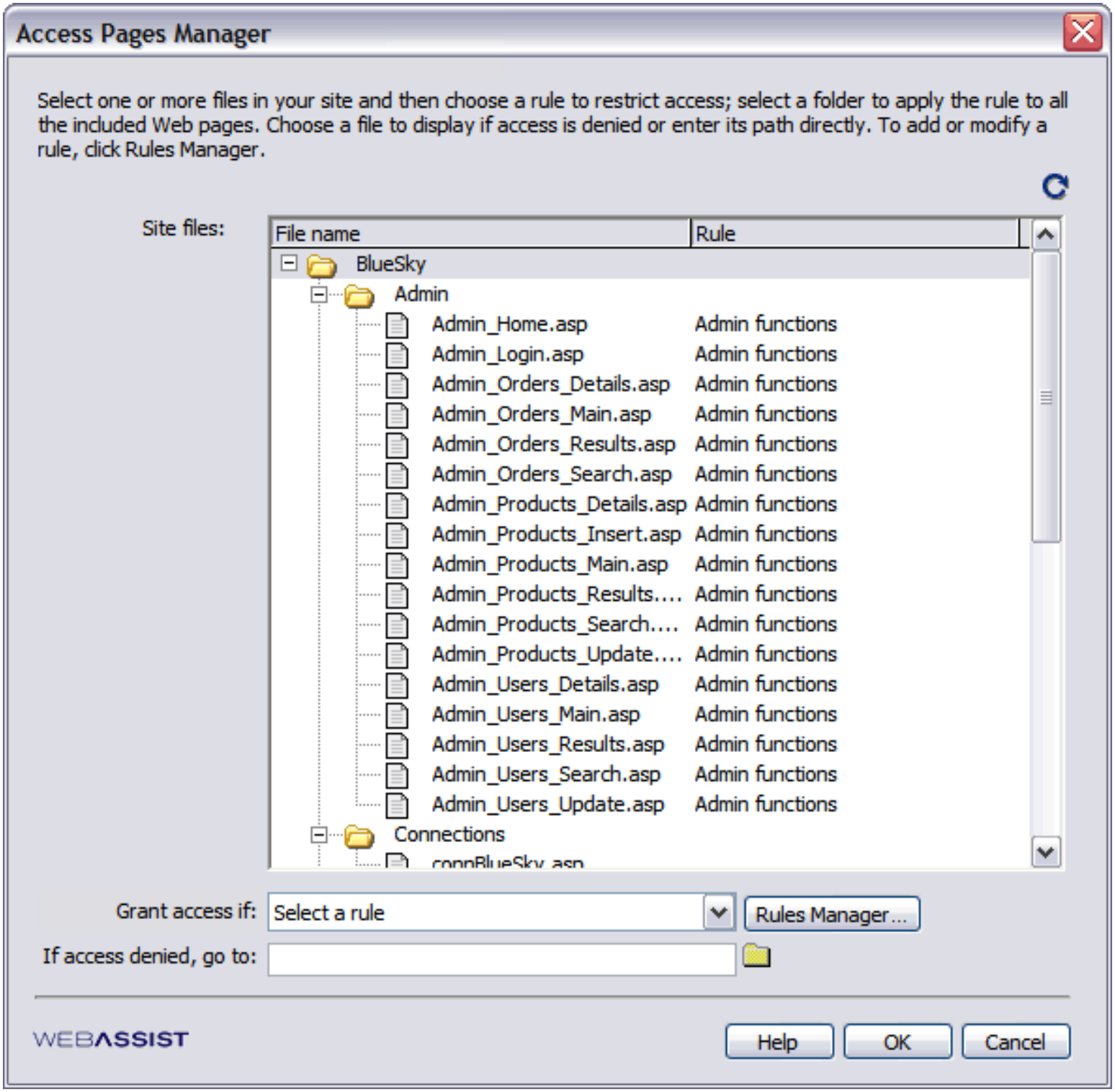
- Features: Access Rules Manager

Access Pages Manager

Manages the application of the Page Access server behavior across multiple pages.

Access

To access the Access Pages Manager, go to Modify > SecurityAssist > Access Pages Manager...



Select a single page or multiple pages to apply rule criteria to using the available form fields.

Grant access if: Specifies the applied access rule. Each defined rule is presented with it's inverse. For example, if you define a rule Administrator, both Administrator and Not Administrator are available in the list.

Note: an asterisk next to the rule name in the file display area indicates that the applied rule is not defined for the current site.

Rules manager: Opens the Access Rules Manager to administer the rules configured for the current site.

If access denied, go to: Specifies the redirect location if the user fails to meet the authentication criteria in the specified rule. You may use site relative locations (i.e. ../../) to specify a file location or a full http:// address to redirect to another site.

Related topics

- Features: Access Rules Manager
- Features: Page Access server behavior

Password Encryption

SecurityAssist provides encryption features that allow you to encrypt passwords used for authentication when they are stored in the database, as well as randomly generate passwords as part of your registration and update profile pages. These features provide an extra level of security when storing passwords in your database.

SecurityAssist utilizes an SHA-1 encryption format, which provides one way data encryption. One way encryption prevents the data from being unencrypted while allowing the same data to always be encrypted in the same way every time. Thus, at log in, the user can enter their standard password and the encrypted version is compared to the encrypted value stored in the database.

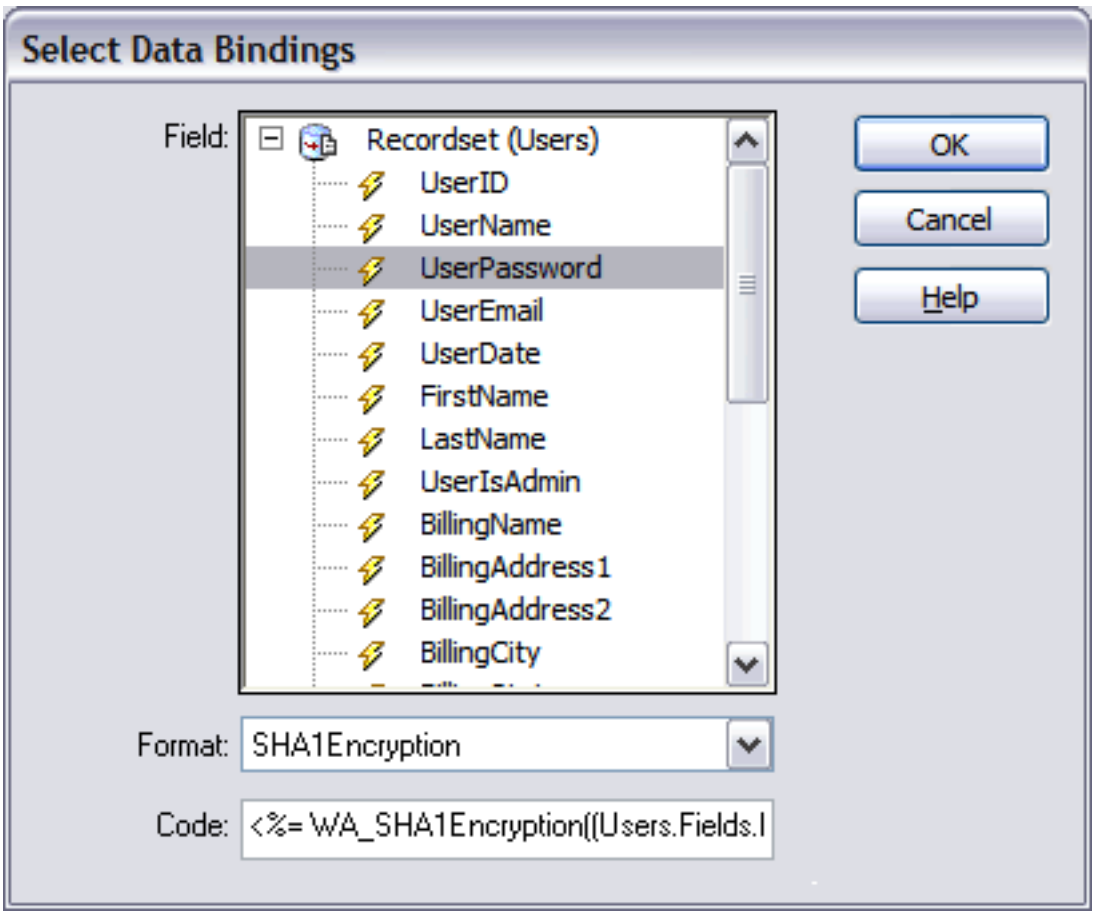
Important Note: When using SecurityAssist for password encryption, the type of password field in your database must be a Var Char type of at least 40 characters long. If authentication is not working, check that the password field in your database meets this minimum character length.

SHA-1 Encryption

Data encryption for passwords is applied directly to the recordset being used to insert or update the password information in the user table in the database.

This can be accomplished in the Bindings panel by binding the Password value from your recordset to the form element making the update, and with that binding still selected, choosing SecurityAssist > SHA-1 Encryption from the Format column.

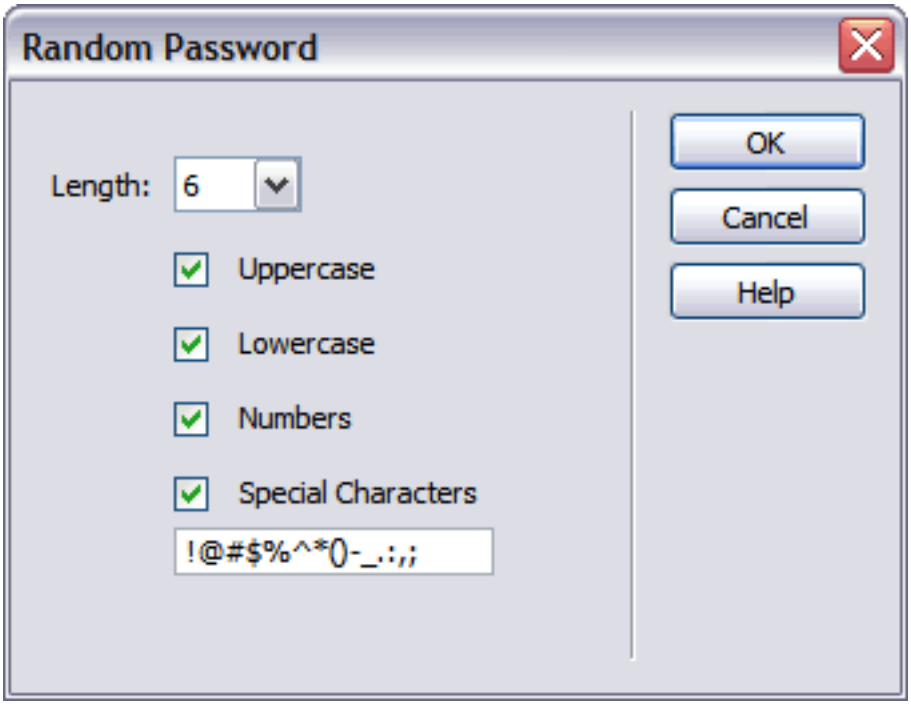
If making the encryption association in a server behavior or the SecurityAssist Wizard, you can also use the Data Bindings window to make this selection by associating the SHA-1 Encryption format to the recordset binding:



Random Password

The Random Password data binding can be made available through the Bindings panel by selecting the (+) button and going to SecurityAssist > Random Password.

This binding can be applied at any point that a password requires retrieval or is updated in the database by applying the binding to form element or server behavior performing the operation.



The password generated will be of the length selected from the dropdown list, and will include any of the selected options. If selecting special characters, those listed in the provided text field will be available for password generation in addition to any selected options (i.e., Uppercase, Lowercase, Numbers)

Form Element Configuration

When SecurityAssist inserts certain form elements - menus, radio buttons and checkboxes - additional steps are required to make these elements function properly. For example, with a menu form element, the labels and corresponding values must be assigned to either static or dynamic values. The following sections broadly outline steps to be taken when configuring these form elements:

- Lists/Menus
- Radio buttons
- Checkboxes

Lists / Menus

Form elements declared to be Menus in SecurityAssist are inserted without either labels or values assigned. Labels and values can be assign static or dynamic. Dynamic labels and/or values often require an additional recordset.

For more information about assigning static labels and values, see **Help > Dreamweaver Help**, under **Making Pages Dynamic > Creating Forms > Reference > Setting the List Values dialog box options**.

For more information about dynamic labels and values, see **Help > Dreamweaver Help**, under **Making Pages Dynamic > Creating Forms > Reference > Setting the Dynamic List / Menu dialog box options**.

Radio buttons

Designating a form element as a radio group in SecurityAssist results in two radio buttons, with separate labels and the same name being inserted. The values of both radio buttons are set to the same chosen data field; you'll need to modify the Checked Value of each radio button in the Property inspector to accurately reflect the value you want to pass to the database field. You may want to add additional radio buttons to your group; if you do, be sure apply the same name to all buttons in the same group.

For more information about dynamic radio buttons, see **Help > Dreamweaver Help**, under **Making Pages Dynamic > Creating Forms > Reference > Setting the Dynamic Radio Group dialog box options**.

Checkboxes

SecurityAssist inserts a checkbox for a designated field with a placeholder value of 1. Under most circumstances, you'll need to choose a database field that accepts a Boolean (Yes/No; True/False; 1,0) value. This process is initiated by selecting the checkbox and then choosing Dynamic from the Property inspector.

Important: the 'value' atribute of the checkbox is set for this field by the wizard, but the 'checked' attribute is not configured. Consequently, this field will not default to checked on the page, even if the value returned to this field represents a checked value. You will need to write custom code to set the state of this field to checked when the appropriate value corresponding to the checked state is specified.

For more information about dynamic checkboxes, see **Help > Dreamweaver Help**, under **Making Pages Dynamic > Creating Forms > Reference > Setting the Dynamic CheckBox dialog box options**.