



Universal Email Help Documentation

Copyright © 1999-2007 WebAssist.com Corporation.
All rights reserved.

Preparing your pages for Universal Email

Universal Email can be configured to use server-side information and form elements to set properties for the email application. Prior to applying the Universal Email server behavior to the page, it is important that your pages contain the necessary attributes which will be used by the server behavior. The following sections detail any information that your pages may need configured in advance to prepare for the application of the server behavior.

- [Using forms and Universal Email](#) - Forms allow you to interact with or gather information from visitors to a website and use this information in your email application. This lesson discusses how to integrate your form information and reviews defining properties of form elements.
 - [Using recordsets and Universal Email](#) - Email properties can be defined by a Dreamweaver Recordset value. You will need to add the Recordset to your page before you can define the email object. This lesson discusses different methods for retrieving information from recordsets within Universal Email.
 - [Using server-side code and Universal Email](#) - Email properties can also be defined using values set on the server. This lesson discusses where it is acceptable to place server-side values when configuring Universal Email, as well as methods for using server-side code to specify Body content.
-

Using forms and Universal Email

One method for defining the properties of an email is retrieving information from form elements on the page to which the Universal Email server behavior is applied.

Values for properties can be retrieved from any form element for which a value can be set.

Form elements which allow multiple selections (radio groups, menus, lists) allow you to use the value for a specified selection.

The following email properties can be defined using form elements. Links direct you to the description of the property and the description provides implementation suggestions:

- [Redirect after Send](#)
A radio button value can be used to set the path for the redirect location. This may be useful in situations where the redirect location may vary dependent on a radio button selection.
- [Subject](#)
The subject line of an email can be retrieved from a textfield, using a pre-defined value or the value entered by the user when its parent form is submitted. Other options include using a hidden value set from other form elements (radio button selections, hidden form elements, etc.)
- [Importance](#)
The priority of the email can be defined by a value of a selected radio button, a textfield or hidden form element.
- [From](#)
The sender of the email can be retrieved from a value entered by a client in a text field, from a list of available recipients, a radio or checkbox selection, or a hidden form element.
- [To](#)
The recipient(s) of the email can be retrieved from a value entered by a client in a text field, from a list of available recipients, a radio or checkbox selection, or a hidden form element.
- [CC](#)
The recipient(s) of the email can be retrieved from a value entered by a client in a text field, from a list of available recipients, a radio or checkbox selection, or a hidden form element.
- [BCC](#)
The recipient(s) of the email can be retrieved from a value entered by a client in a text field, from a list of available recipients, a radio or checkbox selection, or a hidden form element.
- [Attachments](#)
Attachment location and file names can be retrieved from a value entered by a client in a text or file field, from a list of files, a radio or checkbox selection, or a hidden form element.
- [Body](#)
The content of the email can be retrieved from a value entered by a client in a text field, from a list of available recipients, a radio or checkbox selection, or a hidden form element.

To learn more about configuring values associated to form elements, please refer to [Help > Using Dreamweaver > Making Pages Dynamic > Creating Interactive Forms](#)

To set the value of a form element:

1. Select a form element on the page.
2. Within the property inspector, enter the information to be retrieved in the appropriate value field.
3. For text and text area form elements this is the Init Val.

4. For checkboxes and radio buttons, this is the Checked Value.
5. For hidden form elements, this is the Value.
6. For menus and lists, use the List Options dialog, accessible through the List Options button, to specify appropriate values for each item available in the list.

Note: Form elements should be specified using unique names in order to distinguish them when information is retrieved from them.

To retrieve the value from a form element:

1. Open the page which contains the form.
 2. Access the [Universal Email dialog box](#).
 3. Click the tab which contains the property you wish to define.
 4. For the property to be defined, select the data bindings icon (lightning bolt). The data bindings interface will open.
 5. Expand the form listed and select the appropriate form element to retrieve the value from.
 6. Click OK. The necessary code populates the Universal Email dialog to retrieve the value from the selected form element.
-

Using recordsets and Universal Email

One method for defining the properties of an email is retrieving information from a recordset on the page to which the Universal Email server behavior is applied.

The Universal Email dialog box contains features to directly configure recordset content to populate the TO, CC, BCC, and Attachment properties. Please see the [To Tab](#) section for more information.

Body content can be retrieved from a recordset by specifying server-side code directly within the Body tab of the Universal Email dialog box. We recommend creating your body page in Dreamweaver and using the Get from File button to import content directly into the Universal Email dialog. See [Using Server-side code and Universal Email](#) for more information.

To learn more about configuring recordsets, please refer to Help > Using Dreamweaver > Making Pages Dynamic > Defining Sources of Dynamic Content > Defining a Recordset

To define email application attributes from a recordset:

1. Open the page containing the recordset.
2. Access the [Universal Email dialog box](#).
3. Select the tab for the property to be configured.
4. Click the Select Data Bindings button (lightning bolt) next to the property to be defined.
The Select Data Bindings dialog opens.
5. From the Field control list, expand the recordset which you are retrieving data from.
6. Click the recordset column which the information will be retrieved from.
7. The Code: field will be populated with the necessary code to retrieve the information from the recordset.
8. Click OK.

The Code referenced within the Select Data Bindings dialog box will be populated into the appropriate field within the Universal Email dialog box.

9. Continue to configure the server behavior as appropriate and Click OK to save the configuration.
-

Using server-side code and Universal Email

Universal Email supports all server-side code references in defining properties for your email application. Each property defined within the Universal Email server behavior will accept server-side code directly upon configuration. This allows you to dynamically define email properties based on values retrieved from the server, making your email completely customizable.

The [previous lesson](#) discusses how the Universal Email dialog box contains features to directly configure recordset content for email properties using the dynamic attributes button. This populates the attribute with the server-side code required to retrieve the recordset information. You have the option of writing this code directly in the configuration field as well for both recordsets, or any other server-side code reference you wish to use.

The Body tab also allows server-side code references, including recordsets, to create body content based on attributes returned by the server. We recommend creating your body page in Dreamweaver and using the Get from File button to import it directly into the Universal Email dialog. See the [Body Tab](#) section for more information.

Getting started guide

The Getting started guide is designed to inform you of the requirements for using Universal Email, as well as guide you through the process of creating a simple but functional email application using Universal Email. It's aimed at users who have created web pages in Dreamweaver but are unfamiliar with particular aspects of Dreamweaver.

The following chapters outline server requirements for Universal Email, how to add support for additional email objects, and sample static and dynamic email applications including the steps necessary to create them.

- [Supported server models](#)
 - [Adding additional email object support](#)
 - [Sample Applications](#)
-

Supported server models

Universal Email creates code which can be used for the following application server models and scripting options:

- Microsoft Active Server Pages (ASP)
 - JavaScript
 - VBScript
 - Coldfusion (CFML)
 - PHP
-

Adding additional email object support

Universal Email can be extended to support additional email objects. Use the API to define the properties required by Universal Email.

Refer to [Adding Email Objects](#) for information specific to adding additional email object support to Universal Email.

Learn more about some useful resources offered by WebAssist.

Sample applications

Universal Email ships with two sample applications to teach you how to create custom email applications:

- Using a form to manually create and send an email
- Using recordsets and server-side code to dynamically create and send an email

The sample files which correspond to these lessons are installed with the extension and can be located at:

AppData/Roaming/Adobe/Dreamweaver 9/Configuration/Shared/WebAssist/UniversalEmail/Help/02_Start/Samples

Sample files are available in all the server models supported by Universal Email: ColdFusion, ASP JavaScript, ASP VBScript, and PHP.

These applications demonstrate the three techniques covered in [Preparing your pages for Universal Email](#):

- [Using forms and Universal Email](#)
- [Using Recordsets and Universal Email](#)
- [Using Server-side code and Universal Email](#)

To setup and use these sample files:

1. Define a Site within Dreamweaver which corresponds to your server model and references the appropriate server model directory found in the Samples location displayed above (js, vb, or cf). In creating the site definition, we recommend using the remote location and testing server features to provide easy access for both viewing the pages as well as editing them in Dreamweaver (see Dreamweaver help for more information on site definitions and their features)
2. In Dreamweaver, open the page with which you will be working. Within the Server Behaviors Panel Group, select and open the existing application of Universal Email on the page. The Universal Email dialog box will open and display the Email Object tab. From the list of available email objects, select the one which you use with your web server and configure the necessary properties that appear on the tab (See [Email Object Configuration](#) for more information.) Once you have finished configuring the server behavior to work with your object, select OK in the dialog box and save the page. Repeat this step for all pages which you are using.
3. Copy or upload the directories containing your updated files to a location used by your web server.
4. The page which uses server-side code to retrieve email properties uses database content and requires an ODBC connection. The database is located in the db directory found in the Samples location specified above. Configure an ODBC connection for a MS Access database which points to the location of this database and name it "WAUE".
5. Upon completion of these steps, you should be able to browse and use these files on your web server, as well as edit them directly through Dreamweaver.

Note: The rsContent sample pages retrieve email addresses from the MS Access database provided. The addresses used in this database are placeholders and are invalid. It is necessary to update these entries in the database, specify your personal email test addresses and verify the sample email was sent.

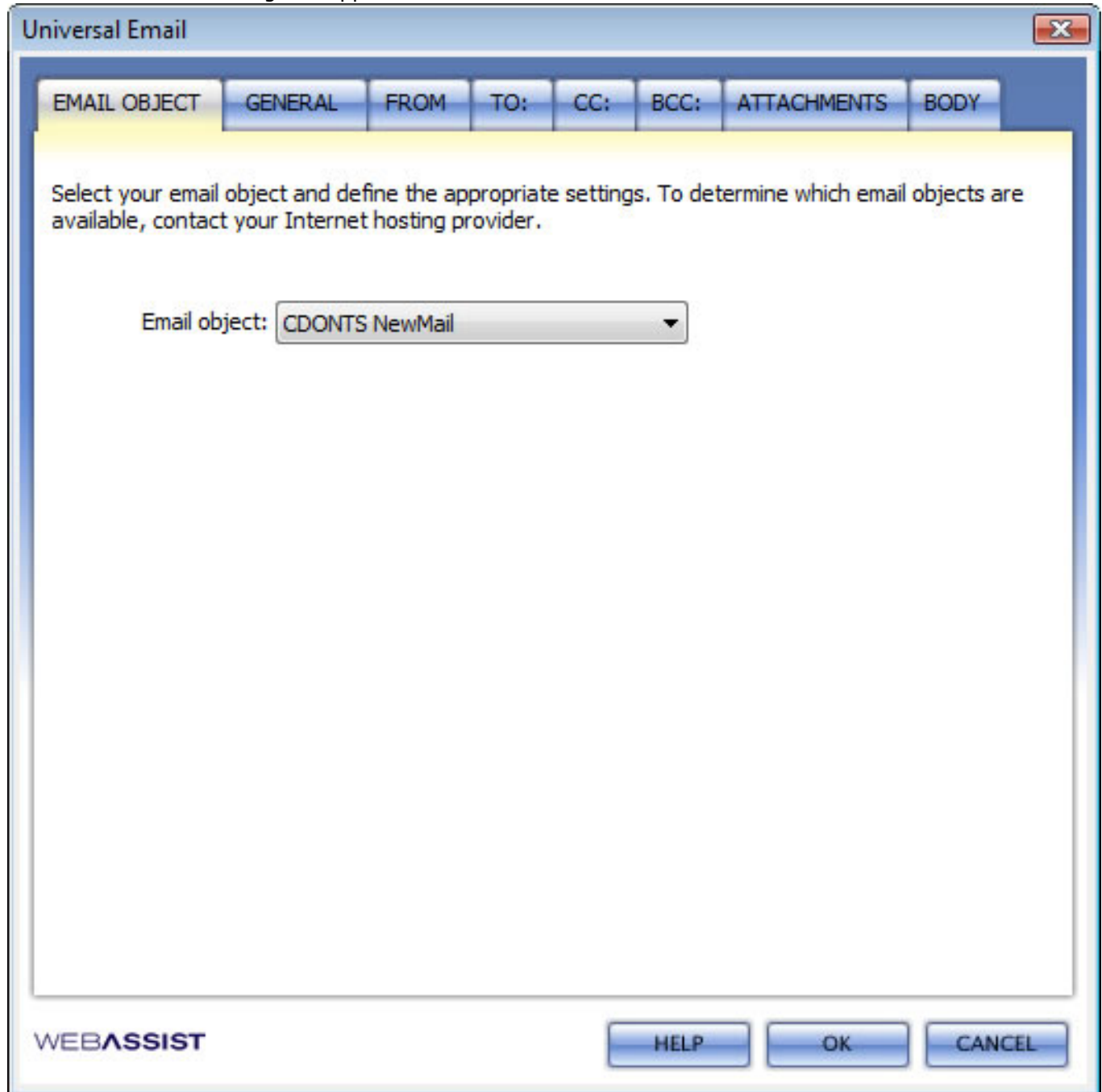
Using Universal Email

This chapter provides a series of brief lessons that guide you through the process of creating custom email applications using the visual authoring tools for Universal Email. For information on each email object in Universal Email, see [Email Object Configuration](#).

To insert the Universal Email server behavior:

1. Select the page that is going to be used to send the email.
2. Choose Insert > WebAssist > Universal Email.

The Universal Email dialog box appears.



3. Use the tab interface for the server behavior to define all the properties for the email object. The server behavior requires the Email Object, To, and From tabs be defined before the server behavior can be applied to

the page. The following lessons cover how to define each tab:

- [Email Object tab](#)
- [General tab](#)
- [From: tab](#)
- [To: tab](#)
- [CC: tab](#)
- [BCC: tab](#)
- [Attachments tab](#)
- [Body tab](#)

4. Select **OK** to apply the server behavior and it will generate the code necessary for the email functionality to work on the page. The ASP version of the server behavior also creates an include folder in the root of your site called `WA_Universal_Email` that contains the supporting files.
-

Email Object tab

The Email Object tab allows you to select the email object type. The rest of the tab is dynamically generated for each email object type.

Select one of the email object links from the list below to learn about the specific properties for that email object.

ASP:

- [AspEmail](#)
- [AspMail](#)
- [CDONTS NewMail](#)
- [CDOSYS](#)
- [SMTPmail](#)
- [w3 JMail](#)

Coldfusion:

- [CFMAIL](#)

PHP:

- [PEAR Mail using SMTP](#)
- [Mail](#)

Universal Email



EMAIL OBJECT

GENERAL

FROM

TO:

CC:

BCC:

ATTACHMENTS

BODY

Select your email object and define the appropriate settings. To determine which email objects are available, contact your Internet hosting provider.

Email object: CDONTS NewMail

WEBASSIST

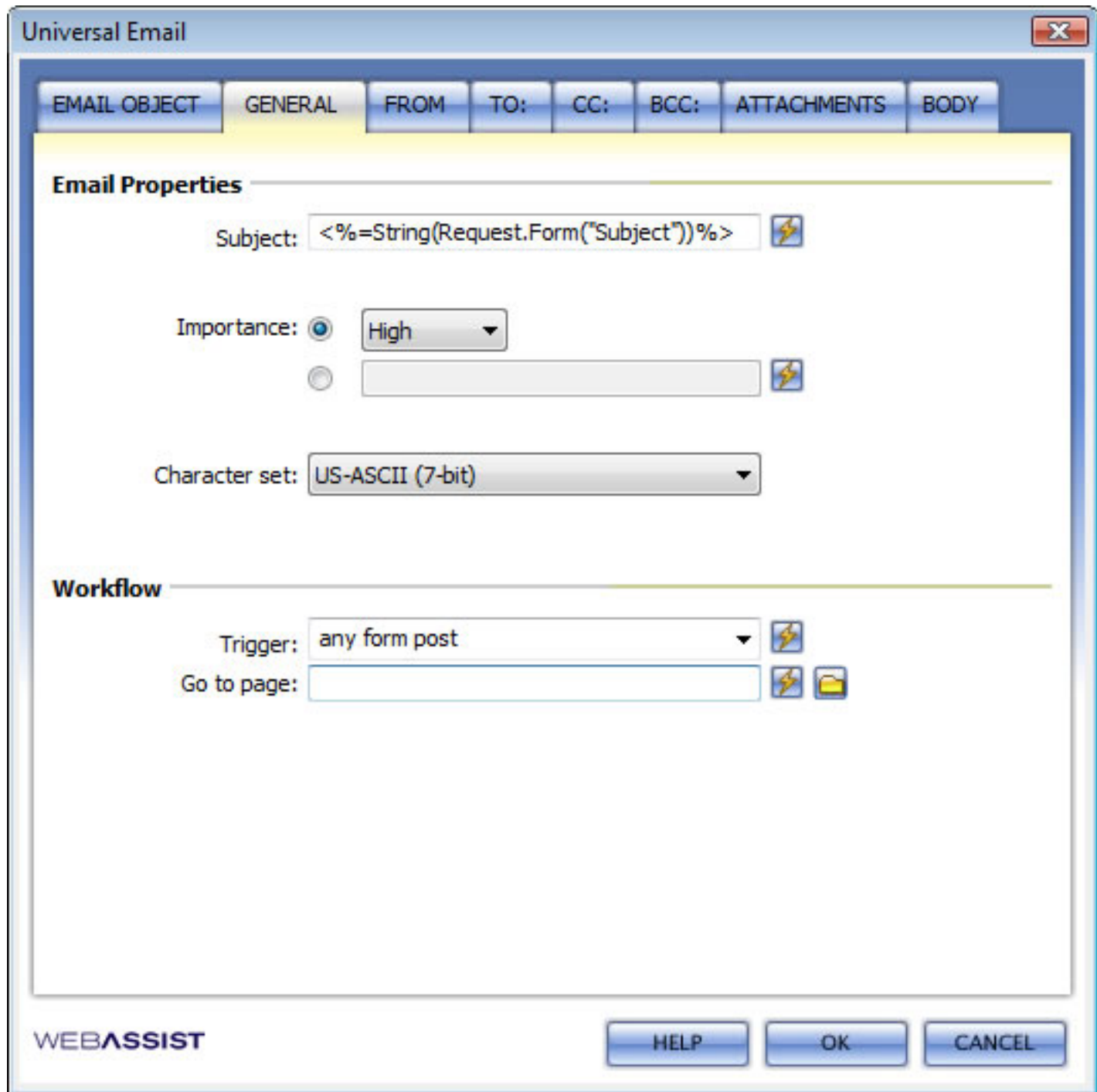
HELP

OK

CANCEL

General tab

The General tab of the Universal Email dialog box configures the subject, importance, and character set selected email, as well as when the email is sent, and where the browser redirects afterwards is configured on this tab.



The image shows a screenshot of the 'Universal Email' dialog box, specifically the 'GENERAL' tab. The dialog box has a title bar with 'Universal Email' and a close button. Below the title bar are several tabs: 'EMAIL OBJECT', 'GENERAL', 'FROM', 'TO:', 'CC:', 'BCC:', 'ATTACHMENTS', and 'BODY'. The 'GENERAL' tab is selected and highlighted in yellow. The main content area is divided into two sections: 'Email Properties' and 'Workflow'. In the 'Email Properties' section, there is a 'Subject:' label followed by a text box containing the code '<%=String(Request.Form("Subject"))%>' and a lightning bolt icon. Below this is an 'Importance:' section with two radio buttons. The first radio button is selected and is followed by a dropdown menu showing 'High'. The second radio button is unselected and is followed by an empty text box and a lightning bolt icon. Below that is a 'Character set:' label followed by a dropdown menu showing 'US-ASCII (7-bit)'. In the 'Workflow' section, there is a 'Trigger:' label followed by a dropdown menu showing 'any form post' and a lightning bolt icon. Below that is a 'Go to page:' label followed by an empty text box, a lightning bolt icon, and a folder icon. At the bottom left of the dialog box is the 'WEBASSIST' logo. At the bottom right are three buttons: 'HELP', 'OK', and 'CANCEL'.

Subject

Sets the content for the subject line included in the email. The following options are available to determine this content:

- a static value - enter text for the value that will be constant as the subject of the email

- retrieving a value from a form element on the page - see [Using forms and Universal Email](#)
- retrieving a value from a dynamic data source - values can be retrieved using server-side code (see [Using server-side code and Universal Email](#) for more information)

Importance

Sets the priority value for the email created. Text values must adhere to the following naming convention and are case sensitive:

- High
- Default
- Low

Static values of High (1), Default (3) and Low (5) are available through the dropdown list.

Code may be entered in the same formats as listed above to specify the [Subject tab](#). Values should reflect a scale from 1 to 5 with 1 as High, 3 as Default, and 5 as Low when retrieved using a dynamic data source.

Character Set

Selects the character set to define for the email header, allowing you to select regional character sets as necessary to ensure the email recipient can render the message appropriately. This options defaults to the US ASCII standard.

Trigger

Specifies the event which will send the email from the page. The following three options are always available:

- **Any Form Post:** Any form posted to the current page that the server behavior is applied to will send the email.
- **Before Page Load:** The email send 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 send the email.
- **eCart [cart name] is not empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
- **eCart [cart name] is empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
- **Button [button name] pressed:** The server behavior is triggered if the specified button on the page is clicked.
- **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.
- **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.

Go to page

Specifies the redirect destination page after the email is sent. Leaving this field blank configures the page to redirect to itself.

The following formats are accepted to specify the redirect location:

- an http: location
(e.g. <http://www.domain.com/folder/destination.asp>)

- a path relative within the defined site
(e.g. /folder/destination.asp)
- a path relative to the document
(e.g. file:///C:/folder/destination.asp)

The Browse button opens the Select a File dialog. Select a path relative to the site or enter a path relative to the document.

The dynamic attribute button opens the Select Data Bindings dialog. Select a data binding available through the dialog or enter server-side code to retrieve data from the page. See [Using server-side code and Universal Email](#) for more information.

Note: The preferred convention is to use a path relative to the site for pages within the same Dreamweaver site definition. This facilitates changes in location of the entire site without reconfiguring file location paths.

From: tab

The From tab of the Universal Email dialog box sets the identity of the sender of the email in your email application, allowing you to specify the email address and the signature text used to identify the sender. Most mail objects require that a valid email address is used to specify the sender. There are three methods for setting the email address and name of the sender:

- Static value - enter text for the value that will be constant as the sender of the email
- retrieve a value from a form element on the page - see [Using forms and Universal Email](#)
- retrieve a value from a dynamic data source - values can be retrieved using server-side code (see [Using server-side code and Universal Email](#) for more information)



The screenshot shows the 'Universal Email' dialog box with the 'FROM' tab selected. The dialog has a title bar with a close button (X) and a tabbed interface with buttons for 'EMAIL OBJECT', 'GENERAL', 'FROM', 'TO:', 'CC:', 'BCC:', 'ATTACHMENTS', and 'BODY'. The 'FROM' tab is active and contains the following text: 'Enter the email address you want replies sent to in the From field. The email address you choose must be recognized as valid email syntax or the email will not be sent.' Below this text are two input fields: 'From name:' with the value '<%=String(Request.Form("FromName"))%:' and a lightning bolt icon; and 'From email:' with the value '<%=String(Request.Form("FromEmail"))%>' and a lightning bolt icon. At the bottom of the dialog are three buttons: 'HELP', 'OK', and 'CANCEL'. The 'WEBASSIST' logo is visible in the bottom left corner.

To: tab

The To: tab of the Universal Email dialog box sets the recipient(s) in the To line of the email in your application and determines whether emails are sent individually or as one email.

To specify recipient(s), click the Add button and edit the entry in the dialog using the form fields below the list of configured recipients. The *Enter Value* field allows you to specify a static value, or retrieve a dynamic value using data bindings. The *Recordset Loop* option allows you to select from a list of Recordsets available on the page, and specify the column that contains the email addresses of your recipients. The arrow buttons allow you to control the order that the emails are sent based on the order of specified recipients.

The screenshot shows the 'Universal Email' dialog box with the 'TO:' tab selected. The dialog has several tabs: EMAIL OBJECT, GENERAL, FROM, TO: (selected), CC:, BCC:, ATTACHMENTS, and BODY. Below the tabs is a text area with instructions: 'To send your email to individual email addresses, enter a static or dynamic value in the Entered value field and click Add. To send to a group of email addresses stored in a database, choose Recordset Loop and select the desired recordset and data column.'

Below the instructions is a 'Send email:' dropdown menu set to 'Once Per Entry'. There are 'Add' and 'Delete' buttons with plus and trash icons, and two arrow buttons for sorting.

A table with two columns, 'Type' and 'Value', is shown. The first row is highlighted in blue and contains 'Entered value' and '<%=String(Request.Form("To"))%>'. The table is empty below this row.

Below the table are two options: 'Enter value:' with a radio button selected and a text field containing '<%=String(Request.Form("To"))%>', and 'Recordset loop:' with a radio button unselected, a dropdown menu, and a 'List of Columns' dropdown menu.

At the bottom left is the 'WEBASSIST' logo, and at the bottom right are 'HELP', 'OK', and 'CANCEL' buttons.

The following options are available to specify recipient information:

Send Email

Specifies that individual emails will be sent to each specified recipient without listing other To line recipients, or that one email will be sent with all recipient information specified in the To line. The following selections correspond to these options:

- Once per entry
- To all entries at once

To add recipient information:

1. Click the Add button to add a recipient entry to the list control.
2. Specify the recipient information using either the [Enter Value](#) or [Recordset Loop](#) options.
3. Repeat this process for all necessary recipient information.

Enter value

Specifies the recipient(s) using one of the following methods:

- a static value - enter text for the value that will be constant as the sender of the email
- retrieve a value from a form element on the page - see [Using forms and Universal Email](#).
- retrieve a value from a dynamic data source - values can be retrieved using server-side code (see [Using server-side code and Universal Email](#) for more information)

Recordset loop

Specifies the recipient(s) by retrieving the information from a recordset on the page. The page will retrieve all records for the specified recordset column and use that information to create the list of recipients.

To remove recipient information:

1. Select the recipient information you wish to be removed from the control list.
2. Click the Minus icon to remove the selected information

To order configured recipient information in the To line:

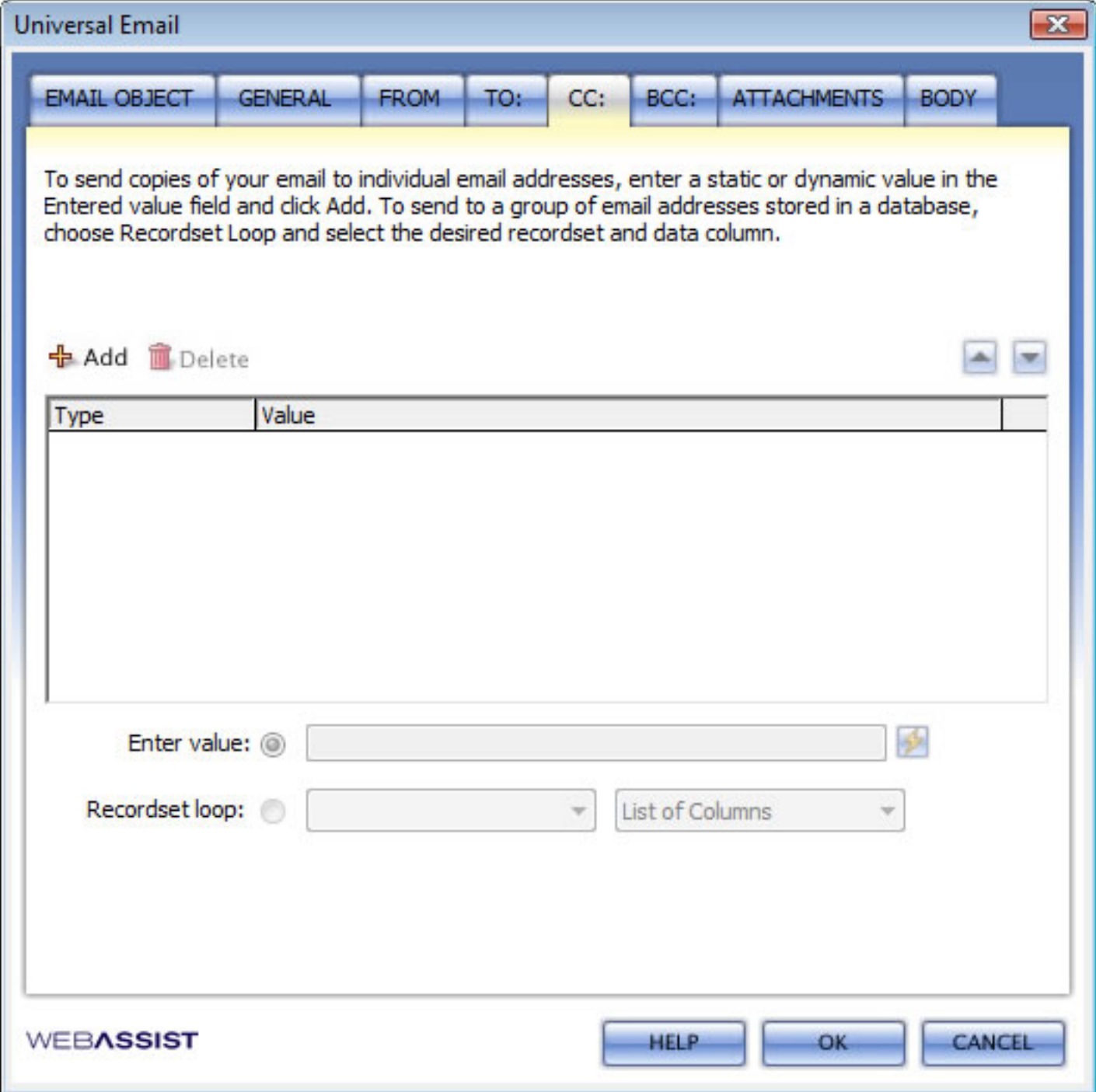
1. Select the recipient information in the control list.
 2. Use the Up and Down arrows to move the selected data one option up or down in the presented order.
 3. Repeat this process until your information is in the order desired for all configured recipient data.
-

CC: tab

The CC: tab of the Universal Email dialog box sets the recipient(s) in the CC line of the email in your application.

Options for configuration of CC content are the same as utilized by the To: line, with the exception that options for sending a single email or multiple emails to multiple recipients is only available for specification in the To: tab.

Please refer to the [To: tab](#) section of this help for more information.



The screenshot shows the 'Universal Email' dialog box with the 'CC:' tab selected. The dialog has a title bar with a close button (X) and a tabbed interface with buttons for 'EMAIL OBJECT', 'GENERAL', 'FROM', 'TO:', 'CC:', 'BCC:', 'ATTACHMENTS', and 'BODY'. The 'CC:' tab is highlighted in yellow. Below the tabs, there is instructional text: 'To send copies of your email to individual email addresses, enter a static or dynamic value in the Entered value field and click Add. To send to a group of email addresses stored in a database, choose Recordset Loop and select the desired recordset and data column.' Below this text are 'Add' and 'Delete' buttons with icons, and two arrow buttons. A table with two columns, 'Type' and 'Value', is present but empty. Below the table are two input sections: 'Enter value:' with a radio button and a text field, and 'Recordset loop:' with a radio button, a dropdown menu, and a 'List of Columns' dropdown menu. At the bottom left is the 'WEBASSIST' logo, and at the bottom right are 'HELP', 'OK', and 'CANCEL' buttons.

Universal Email

EMAIL OBJECT GENERAL FROM TO: CC: BCC: ATTACHMENTS BODY

To send copies of your email to individual email addresses, enter a static or dynamic value in the Entered value field and click Add. To send to a group of email addresses stored in a database, choose Recordset Loop and select the desired recordset and data column.

+ Add Delete

Type	Value
------	-------

Enter value:

Recordset loop: List of Columns

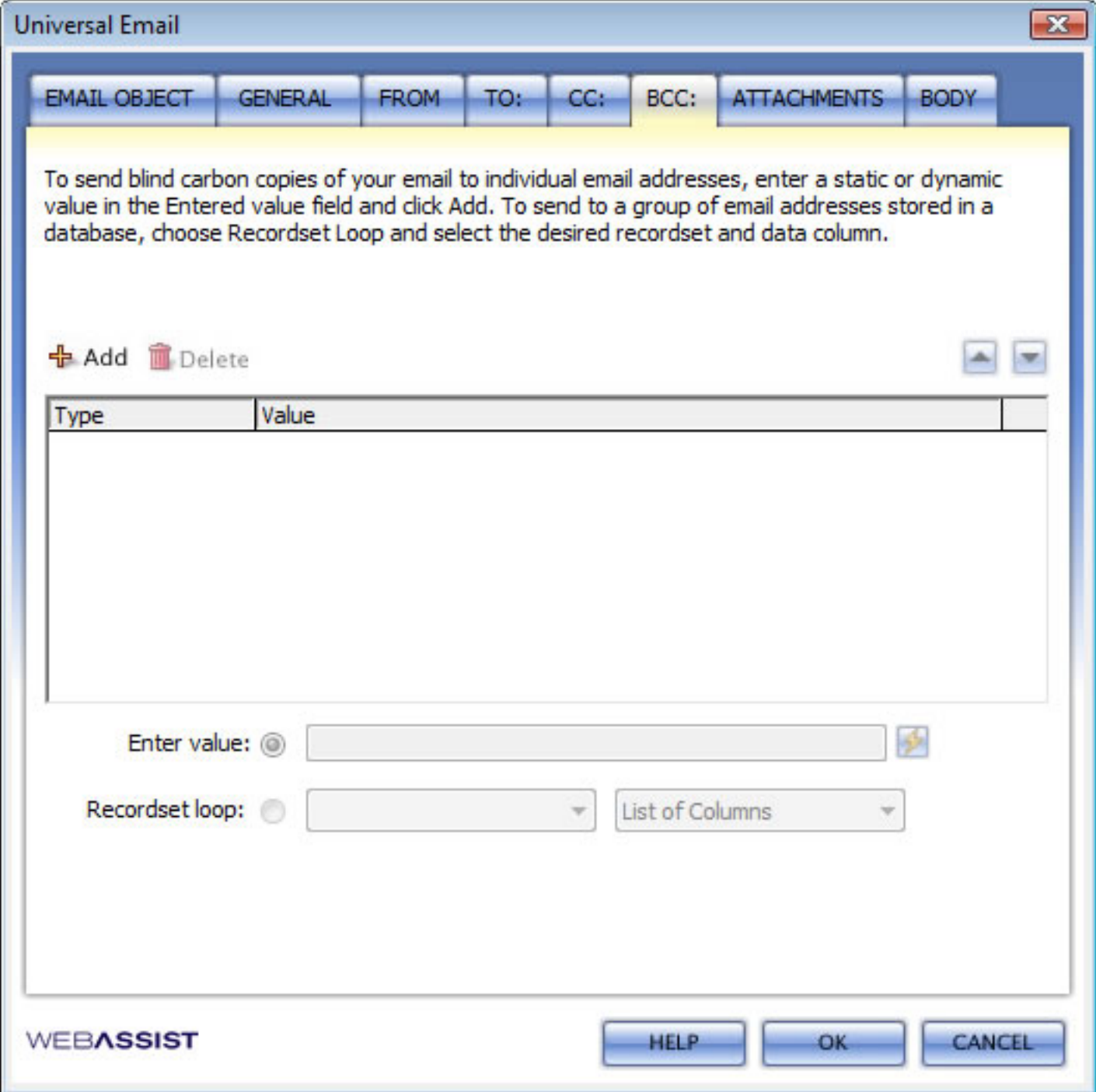
WEBASSIST HELP OK CANCEL

BCC: tab

The BCC tab of the Universal Email dialog box sets the recipient(s) in the BCC line of the email in your application.

Options for configuration of BCC content are the same as utilized by the To: line, with the exception that options for sending a single email or multiple emails to multiple recipients is only available for specification in the To tab.

Please refer to the [To: tab](#) section of this help for more information.



The screenshot shows the 'Universal Email' dialog box with the 'BCC:' tab selected. The dialog has a title bar with a close button (X) and a tabbed interface with the following tabs: EMAIL OBJECT, GENERAL, FROM, TO:, CC:, BCC: (selected), ATTACHMENTS, and BODY. Below the tabs, there is a yellow instruction box: 'To send blind carbon copies of your email to individual email addresses, enter a static or dynamic value in the Entered value field and click Add. To send to a group of email addresses stored in a database, choose Recordset Loop and select the desired recordset and data column.' Below this, there are '+ Add' and 'Delete' buttons. A table with two columns, 'Type' and 'Value', is present but empty. Below the table, there is a radio button labeled 'Enter value:' followed by a text input field and a lightning bolt icon. Below that, there is a radio button labeled 'Recordset loop:' followed by a dropdown menu and a 'List of Columns' dropdown menu. At the bottom left is the 'WEBASSIST' logo, and at the bottom right are 'HELP', 'OK', and 'CANCEL' buttons.


Universal Email

EMAIL OBJECT GENERAL FROM TO: CC: BCC: ATTACHMENTS BODY

To send blind carbon copies of your email to individual email addresses, enter a static or dynamic value in the Entered value field and click Add. To send to a group of email addresses stored in a database, choose Recordset Loop and select the desired recordset and data column.

+ Add Delete

Type	Value
------	-------

Enter value: 

Recordset loop: List of Columns

WEBASSIST HELP OK CANCEL

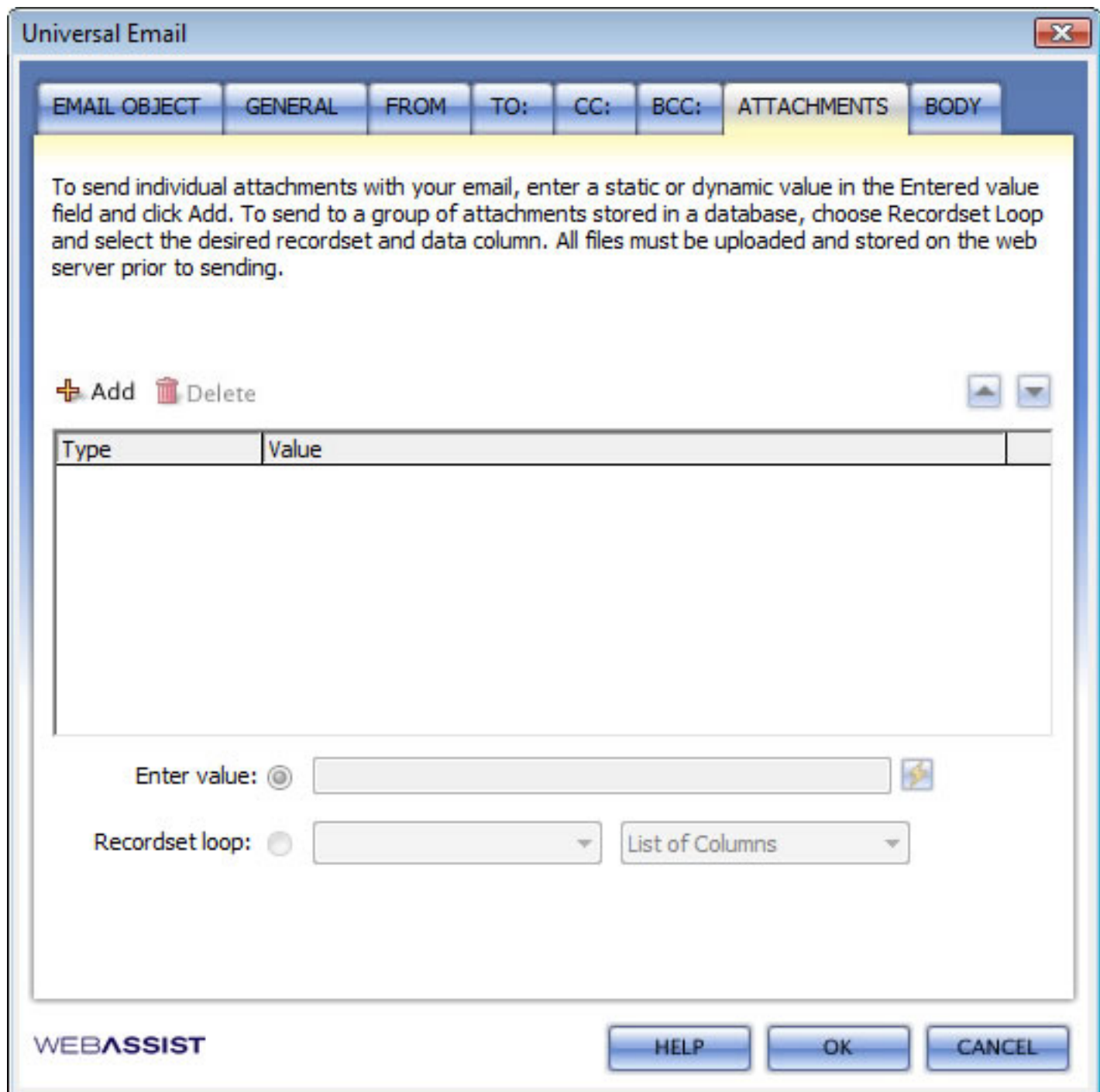
Attachments tab

The Attachments tab of the Universal Email dialog box specifies the file(s) to be delivered with the email created by your application.

Options for configuration of file attachments are the same as utilized by the To: tab line (Please refer to the [To: tab](#) of this help for more information).


The only distinction of note is file attachment information requires specification of a location and file name. The following options are acceptable for specifying the location and names of files to be attached:

- Static Values: a path and filename relative within the defined site (e.g. /folder/file.zip). The extension configures relative paths specified in this manner, but only when specifying static values.
- a path and filename relative to the document (e.g. file:///C:/folder/file.zip). The full path is required if the value is being retrieved from a recordset value or from a form element on the page.



The screenshot shows the 'Universal Email' dialog box with the 'ATTACHMENTS' tab selected. The dialog has a title bar with a close button (X) and a tabbed interface with buttons for 'EMAIL OBJECT', 'GENERAL', 'FROM', 'TO:', 'CC:', 'BCC:', 'ATTACHMENTS', and 'BODY'. Below the tabs is a yellow instruction box: 'To send individual attachments with your email, enter a static or dynamic value in the Entered value field and click Add. To send to a group of attachments stored in a database, choose Recordset Loop and select the desired recordset and data column. All files must be uploaded and stored on the web server prior to sending.' Below this are '+ Add' and 'Delete' buttons with a trash icon, and up/down arrow buttons. A table with two columns, 'Type' and 'Value', is present. Below the table are two radio buttons: 'Enter value:' (selected) and 'Recordset loop:'. The 'Enter value:' option has a text input field and a file icon. The 'Recordset loop:' option has a dropdown menu and a 'List of Columns' dropdown menu. At the bottom are 'WEBASSIST' and 'HELP', 'OK', and 'CANCEL' buttons.

Type	Value
------	-------

Enter value: 

Recordset loop:

Modifying Universal Email

The Universal Email server behavior can easily be re-accessed and modified. This allows you to change your pages through the user interface without the need to directly edit code on the page.

To modify the Universal Email server behavior:

1. Open the page you wish to modify.
2. Access the Server Behaviors Panel from the Application Panel Group.
3. Within the Panel Group, double click the instance of the Universal Email server behavior you wish to modify.

The Universal Email dialog box opens.

4. Make any necessary modifications in accordance with the configuration information outlined in the [Using Universal Email](#) section.
5. Select **OK**.

The appropriate changes will be made to the page and any necessary include files will be added or updated.

Note: If changing the email object being used, the include file utilized by the previous object selected will not be removed from the WA_Universal_Email include folder. This will need to be done manually.

Reference

This section provides the following resources for configuration of your email application and for adding email objects to use with Universal Email.

- [Email Object Configuration](#)
 - [Adding Email Objects](#)
-

Body tab

The Body tab of the Universal Email dialog box specifies the message content for the email created by your application.

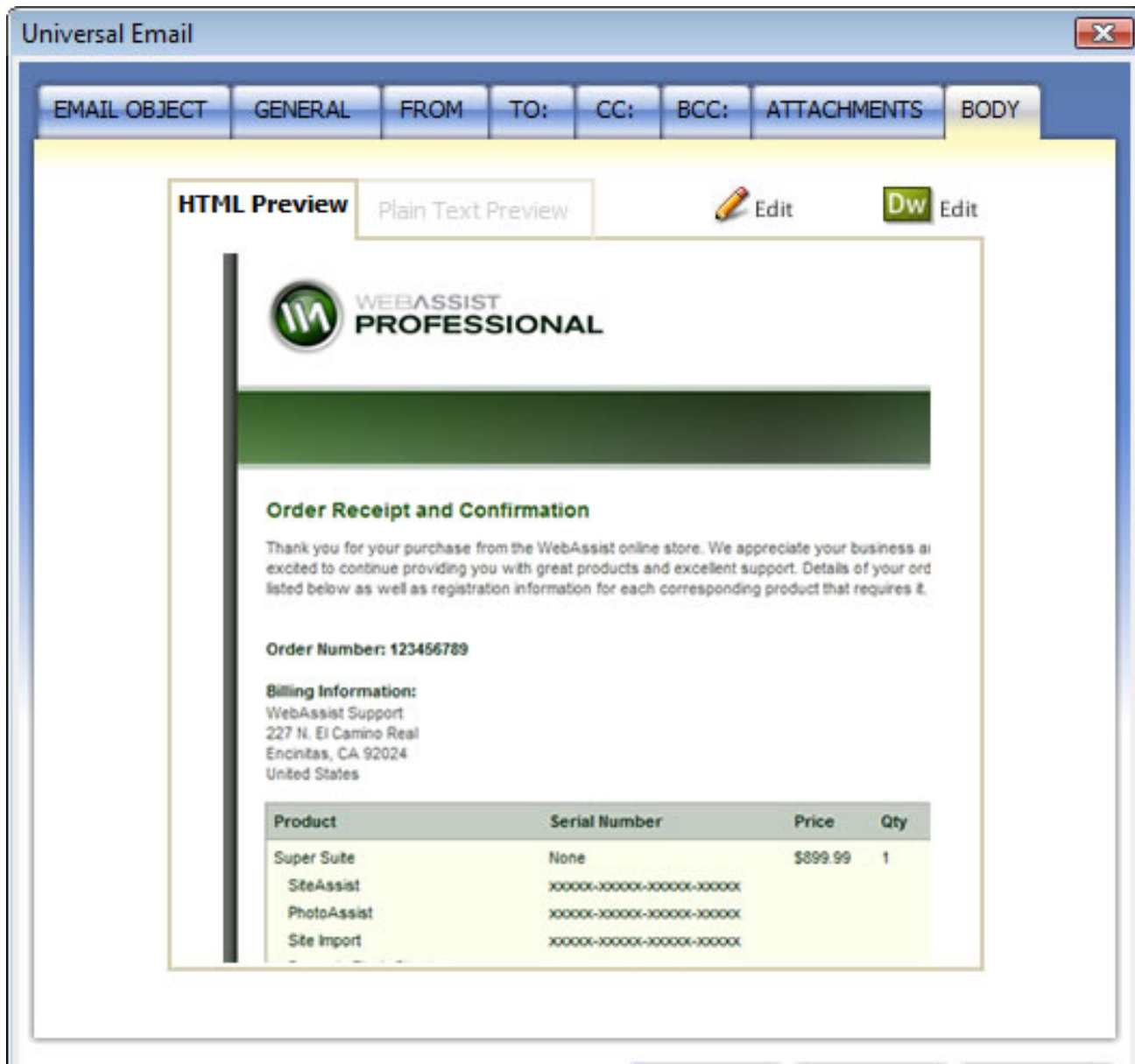
Specified email content can be previewed in this dialog for both HTML and Plain Text version in the corresponding tabs.

For the current tab selected, you may specify content either using a text editor opened using the Edit (pencil) icon, or by editing a file directly within Dreamweaver opened using the Edit (DW) icon.

Editing your email content within Dreamweaver allows you to take advantage of its available features, including html development, dynamic data tools, and css capabilities. Additional functionality provided by other WebAssist software, such as eCart and DataAssist, can also be used to assist in developing your email content.

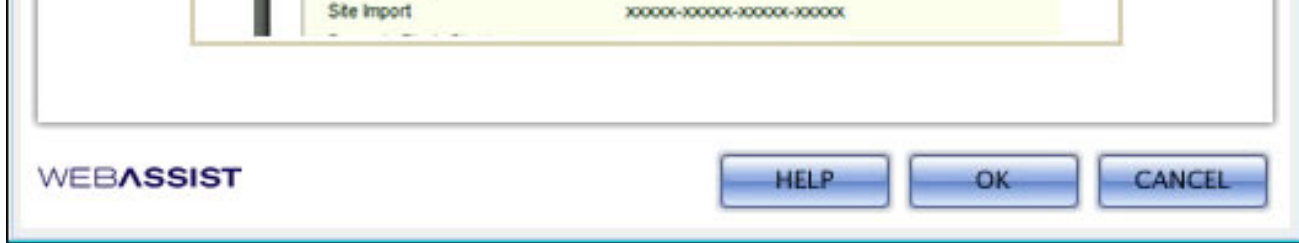
The page opened to edit your content is a temporary page, and saving it will return the updated content to the Universal Email dialog. It should be noted:

- Relative locations for links and images need to be updated to contain the full path. Once an email has been sent by your application, it will need a full path to a location, as it will be outside of your staged environment and in somebody's mailbox.
- CSS will work if specified in-line, but any style declarations that are normally placed in the head will need to be specified in the editable region called "emailhead" when created in this document window. This editable region is only visible in code view in DW.
- Any server-side code that you wish to be run on the page that would be normally be placed above the head tag needs to be specified on the page you are applying the Universal Email server behavior to.



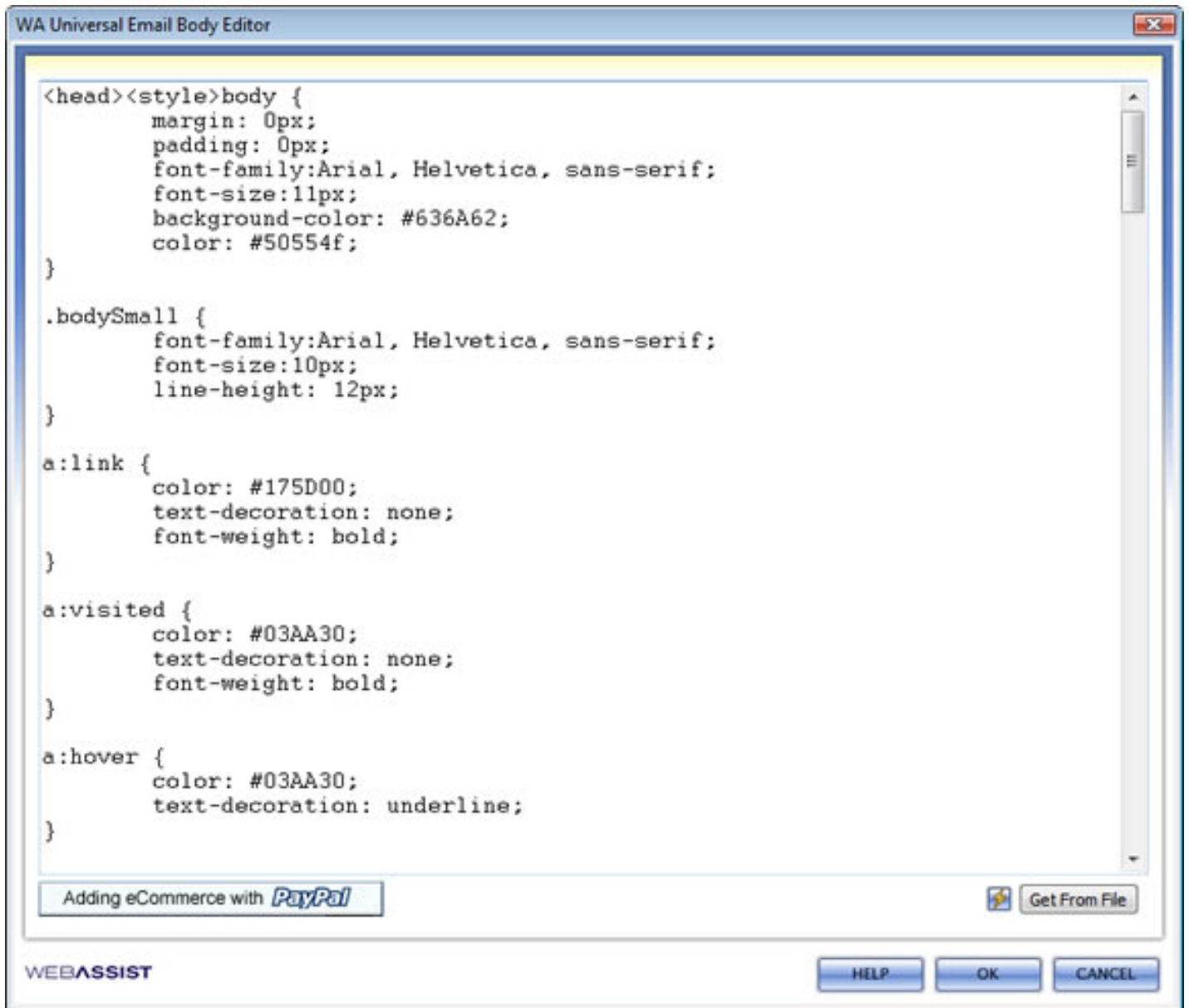
The screenshot shows the 'Universal Email' dialog box with the 'BODY' tab selected. The 'HTML Preview' tab is active, displaying a preview of an email. The email content includes a logo for 'WEBASSIST PROFESSIONAL', a title 'Order Receipt and Confirmation', a thank you message, an order number (123456789), billing information, and a table of products.

Product	Serial Number	Price	Qty
Super Suite	None	\$899.99	1
SiteAssist	XXXXXXXX-XXXXXX-XXXXXX-XXXXXX		
PhotoAssist	XXXXXXXX-XXXXXX-XXXXXX-XXXXXX		
Site Import	XXXXXXXX-XXXXXX-XXXXXX-XXXXXX		



In the text editor, there are three features of the user interface available to help in creating the content for your message:

- Content window - allows you to directly type in text or code to be used within the body content.
- Adding eCommerce with Paypal - this image links directly to help documentation explaining how to add PayPal payment buttons to your email body content.
- Get from File button - opens the Select a File For Your Mail Body dialog box to choose a source document to import body content. When importing ASP, PHP, or CFML pages, source code using server-side content is supported.



Adding email objects

Universal Email can be configured to support additional email objects. It contains an extensible architecture which allows configuration information specific to an email object to be added and supported.

Note: Additional objects cannot be specified for Coldfusion support. CFMAIL is the only supported CFML option.

Universal Email email object definition files

All objects definition files for the Universal Email Extension are located in the following folder on this machine:

One definition file is placed in this folder upon installing the extension. This file contains the information for all email objects currently supported by the extension: WAUniversalEmail.xml

This chapter explains how to create additional XML files which specify the necessary properties to support other email objects. Each of the following sections detail the necessary tags and functions required for configuration of your XML file.

- [Parent tag](#)
- [Child Tags](#)
- [Defining the User Interface \(DEFUI\)](#)
- [Available defined form elements](#)

To create an XML file to support additional email objects:

1. Select File > New
2. In the New Document dialog box, select Basic from the Category window.
3. From the Basic Page window, select XML.
4. Click OK.

A blank XML document will open.

5. Specify the appropriate tags, functions and their corresponding properties in accordance with the guidelines outlined in the following pages for the email object you wish to add (Refer to the existing email object definition XML file noted above as well as the outlined sections).
6. Save the file and place it in the same directory as the existing email object definition XML file.

The next time Dreamweaver is started, the extension will automatically recognize the new XML file and include its configuration information in the list of available email objects supported in Universal Email.

Note: It is a good idea to save a copy of this file in a safe location as it will need to be transferred manually if you install the extension on a different machine or if you remove or re-install the extension.

Email object configuration

The Email Object tab specifies the email object utilized for this email implementation. Configuration options specific to each available email object appear upon selection from the list.

Universal Email supports the following email objects:

- [AspEmail](#)
- [AspMail](#)
- [CDONTS NewMail](#)
- [CDOSYS](#)
- [CFMAIL](#)
- [PEAR Mail using SMTP](#)
- [PHP Mail](#)
- [SMTPmail](#)
- [w3 JMail](#)

Please refer to the links above for configuration details for each supported object.

If you wish to integrate an Email Object not currently supported by Universal Email, please refer to [Adding Email Objects](#).

AspEmail

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

SMTP 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"%>`).

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.

Username 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:

SMTP 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"%>`).

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 utilized by Internet Information Server (IIS). It requires no additional configuration through Universal Email 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:

SMTP 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"%>`).

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.

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.

Authenticate

Determines whether to use authentication or not. Default if not given is false.

Username 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.

SMTPmail

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

SMTP 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"%>).

SMTP log file

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).

Username 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:

SMTP 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"%>).

Username 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>

CFMAIL

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

SMTP 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

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 Macromedia's Coldfusion Developer Center, located at:

<http://www.macromedia.com/desdev/mx/coldfusion/>

PEAR Mail using SMTP

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

Authenticate

Determines whether to use authentication or not. Default if not given is false.

Username 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. Only enabled if Authenticate is selected.

Path to PEAR:

The installation location of PEAR within the context of your webserver.

SMTP 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"%>).

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.

Note: For more information and support for the PEAR Mail object, please refer to:

<http://pear.php.net/package/Mail>

PHP Mail

The following information is required for configuring your email application to use the native mail() functions available in PHP:

SMTP 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"%>).

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.

Note: For more information and support for the PEAR Mail object, please refer to:

<http://us3.php.net/mail>

Using PayPal Buttons in Emails

WebAssist has partnered with PayPal to bring you the PayPal eCommerce Toolkit, a FREE extension allowing you to add PayPal payment buttons directly within your web pages and HTML emails. Using this Toolkit, you can add the following PayPal button directly to any html page, including HTML based email:

- Single Item Purchase buttons - create a payment button for your product or service and let customers immediately pay you with any major credit card.
- Add to Cart buttons - add items to your PayPal shopping cart utilized by your web store
- Subscriptions buttons - offer recurring payments through email. Set up an ongoing charge or payment plan for customers, set up a free or paid trial period, and also charge a setup fee to initiate the subscription.
- Donations buttons - provide people the opportunity to donate money to your organization directly

To get the PayPal eCommerce Toolkit:

<http://www.webassist.com/Products/ProductDetails.asp?PID=18>

To create emails that contain PayPal buttons:

1. Create a web page in Dreamweaver, either in HTML, ASP or CFML.
2. Specify the Body content for your email in the page you've created.
3. Use the PayPal eCommerce Toolkit to create your PayPal payment buttons within the file.
4. Save the file.

You have created a source file for the body content of your email.

5. Open the Universal Email dialog box (see [Using Universal Email](#))
6. Click on the Body tab.
7. Click the Get from File button (see the [Body tab](#) section for more information).
8. In the Select a File dialog box, locate the file created in steps 1 through 4.
9. Click OK.

The source code for the page will be imported into the Body specification box, including the Payment button functionality. Your email now has PayPal payment buttons incorporated into its content.

Note: This feature requires that the email format be either HTML or multipart. See the [Subject, importance, and format](#) section for more information.

Parent tag

Tag Name: Parent tag

<WA_UniversalEmail>

Arguments: name, multipart

A WA_UniversalEmail tag defines each email object supported within the extension.

There are two arguments available in this tag:

- **name:** the name of the email object. This attribute selects the text which will represent the object in the user interface.
Note: It is recommended the name attribute be kept under 25 characters to prevent performance issues on the Macintosh.
 - **multipart:** declares if multipart email messages are supported to enable this feature in the General tab of user interface as a mail formatting option. The multipart property is assumed false if not included.
-

Defining the User Interface (DEFUI)

This section is divided into three parts:

1. Design criteria for the Definition UI
2. UI manipulation functions
3. UI naming conventions

Design Criteria for the Definition UI

The definition user interface is put into a layer when the corresponding mail object is selected.

UI Boundaries

Maximum width 500 pixels
Maximum height 300 pixels

Form Elements

You may declare as many form elements as you like, as long as they are not in the [list of defined form objects](#) found in the following section.

Including Files

The Universal Email UI is contained in the "Configuration/Commands/" folder of the Dreamweaver platform you are developing on. Therefore, any media used (images, etc...) should be referred to with a relative path from this folder.

UI manipulation functions

There are three functions defined for use with the DEFUI:

1. **openFileBrowse()**

Argument: Form element to set as a string
Returns: Nothing

Typically used with input buttons that open file browse windows for a defn field.

Example:

On click of the button, open a file browser and set the def2 form element to the resulting selection.

```
<input type="button" value="Browse..." name="d2browse" onClick="openFileBrowse('def2');">
```

2. **openDataBinding()**

Argument: Form element to set as a string
Returns: Nothing

Used with the lightning bolt image to open the dynamic data dialogue box.

Example:

On click of the button, open the dynamic data dialogue and set the def2 form element to the resulting selection.

```
<input type="image" src="../../shared/UltraDev/images/Boltup.gif" onClick="openDynamicBinding('def2');">
```

3. **setRadioButton()**

Arguments: Radiobutton form element to set as a string, index of selection
Returns: Nothing

Example:

If the user's input to the BCCEV field is not blank, set the first instance of the BCCOp radio button group to checked.

```
<input type="text" name="BCCEV" style="width:240px" onBlur="if (findObject('BCCEV').value != '')
setRadioButton('BCCOp',0)">
```

UI Naming Conventions

There are only two conventions that must be followed when naming your DEFUI form elements.

1. Do not repeat form elements listed in the section [Available defined form elements](#).
 2. Any elements you would like to send to the definition function must be named "defn" where n is a numeric index starting from 1 that corresponds to the entries property of the [DEFUI tag](#).
-

Available defined form elements

A:

Att
AttCol
AttEV
AttList
AttOp
AttRS

B:

BCC
BCCCol
BCCEV
BCCList
BCCOp
BCCRS
Body

C:

CC
CCCcol
CCEV
CCList
CCOp
CCRS
COM
COMSelect

D:

DefLayer

E:

ExpandBody

G:

General

H:

HTMBrowse

I:

ImpList
ImpListOrVal
ImpVal **M:**
MailBody
MailFrom
MailRedirect
MailSubject

R:

Redbrowse

S:

SendAs
SendEmail
SendEmail2

T:

Tab0
Tab1
Tab2
Tab3
Tab4
Tab5
Tab6
Tab7
theForm
To
ToCol
ToEV
ToList
ToOp
ToRS

U:

updateCOMList

W:

WABrowse

Child tags

Within the parent tag, three child tags require definition for the specified object. These tags are shown in the example below and outlined in the sections that follow:

```
<WA_UniversalEmail name="mymailobject">
<DEFUI>
...
</DEFUI>
<HELPERJS>
...
</HELPERJS>
<HELPERVB>
...
</HELPERVB>
</WA_UniversalEmail>
```

Definition User Interface tag (DEFUI tag)

Encapsulates the HTML code to be used in the Universal Email dialog box. This code will specify the layout and form elements used to capture parameters required for configuration of Universal Email and the specified email object.

Helper Tags

The remaining two tags are the HelperJS and HelperVB tags. These tags populate the relative include file defined in a site containing pages with the Universal Email server behavior applied. The HelperJS tag populates the Javascript version of these support functions and the HelperVB tag specifies the VBscript equivalent, depending on which client scripting model is specified in the site definition.

Detailed information on each tag is available in the following two sections:

- [DEFUI](#)
 - [Helper Tags](#)
-

DEFUI

Tag Name: Definition User Interface (DEFUI)

Tag: <DEFUI>

Arguments: entries, types

This tag encapsulates the HTML for any properties that must be set when the Server.CreateObject call is made for the email object. Some email objects require additional parameters to be set when the email object is declared. This tag allows you to define the user interface for the additional parameters. Leave blank if no UI is necessary.

You can build a user interface with as many form elements as you like but there are some necessary naming conventions that must be followed. Helpful functions, an explanation of naming conventions, and all other necessary information for the Definition user interface can be found in the Creating a Definition User Interface section of this documentation.

There are two arguments available in this tag:

- **entries:** Number of arguments passed from the Definition User Interface to the mail object definition function.
 - Note:** Form elements should use an incrementing naming convention (e.g. "def1", "def2", ... "defn".)
 - Tip:** Default attributes for these form elements can be specified using the form elements value tag.
- **types:** Comma-separated list of form element types. They should be listed in the order they are displayed in the email object tab of the Universal Email dialog box. The following is a list of the corresponding argument values for each form element type.
 - t = text element (text or password)
 - l = list element
 - c = checkbox
 - r = radio button
 - af = absolute file path
 - uf = url file path

Sample 1:

Tag argument examples

CDO Mail does not require any extra input for defining the email object:

```
<WA_UniversalEmail name="CDO Mail"...>  
<DEFUI entries="0" types=""></DEFUI>
```

However, SoftArtisans' SMTP Mail does.

```
<WA_UniversalEmail name="SoftArtisans SMTP Mail">  
<DEFUI entries="4" types="t,af,t,t">  
...  
</DEFUI>
```

Sample 2:

SMTPmail interface (also refer to the email object XML file)

```
<WA_UniversalEmail name="SoftArtisans SMTP Mail">
```

```
<DEFUI entries="4" types="t,af,t,t">
<table width="100%" cellpadding="0" cellspacing="0" border="0">
  <tr>
    <td width="30%">&nbsp;</td>
    <td width="50%">&nbsp;</td>
    <td width="20%">&nbsp;</td>
  </tr>
  <tr>
    <td align="right" valign="baseline">Remote Host:</td>
    <td colspan="2" valign="baseline">
      <input type="text" value='<%=Request.ServerVariables("SERVER_NAME")%>' name="def1"
style="width:260px">
      <input type="image" src="../../shared/UltraDev/images/Boltup.gif" onClick="openDynamicBinding('def1');"></
td>
    </td>
  </tr>
  <tr>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td align="right" valign="baseline">SMTP Log:</td>
    <td colspan="2" valign="baseline">
      <input type="text" value="" name="def2" style="width:195px">
      <input type="button" value="Browse..." name="d2browse" onClick="openFileBrowse('def2');">
      <input type="image" src="../../shared/UltraDev/images/Boltup.gif" onClick="openDynamicBinding('def2');">
    </td>
  </tr>
  <tr>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td valign="baseline" align="right">Server Login:</td>
    <td colspan="2" valign="baseline">
      <input type="text" name="def3" style="width:100px">
      (Only available with full version)
    </td>
  </tr>
  <tr>
    <td valign="baseline" align="right">Password:</td>
    <td>
      <input type="text" name="def4" style="width:100px">
    </td>
    <td valign="baseline">&nbsp;</td>
  </tr>
</table>
</DEFUI>
...

```

Helper tags

Tag Names: HelperJS and HelperVB

Tag: <HELPERJS>, <HELPERVB>

Properties: global vars

These two tags populate the email object's relative include file when the ServerBehavior is run for JavaScript and VBscript sites. Encapsulated within these tags are the helper functions required by the Universal Email.

Each of these tags must define the following 8 functions:

- WAUE_AddAttachment()
- WAUE_AddBCC()
- WAUE_AddCC()
- WAUE_AddRecipient()
- WAUE_BodyFormat()
- WAUE_Definition()
- WAUE_SendMail()
- WAUE_SetImportance()

Detailed information and code samples are available in the next section ofr each of these functions.

These eight functions, and any other helper functions, must be written in both JavaScript and VBScript and encapsulated in the HELPERJS and HELPERVB tags respectively.

These functions will sometimes require setting a global variable. Utilize the globalVars attribute to specify these properties. The globalVars attribute accepts a comma-separated list of numbers which correspond to functions that use global variables to set properties.

Example: Using global variables

The CDO Mail object passes the importance as a string to the WAUE_SendMail() function. This will require a global variable in the helper functions.

The variable should have the same name as the function, without a lower-case 'g' replacing the preceding "WAUE_"naming convention, e.g.:

```
<HELPERJS globalVars="8">  
var gSetImportance = "";
```

Helper functions

There are eight helper functions required by each helper tag. These eight functions are in numeric and alphabetic order. Each is used for the corresponding email properties configured in the Universal Email dialog box.

WAUE_AddAttachment()

Arguments: mail object, attachment path

Returns: mail object

Corresponding global variable: gAddAttachment (optional)

This function is called for every attachment in the "Attachment" grid control when the Universal Email code runs. It should be used to call the method, add to the property, or add to the global variable for attachments defined in the API documentation for the COM object you are integrating.

WAUE_AddBCC()

Arguments: mail object, email address

Returns: mail object

Corresponding global variable: gAddBCC (optional)

This function is called for every attachment in the "BCC" grid control when the Universal Email code runs. It should be used to call the method, add to the property, or add to the global variable for the BCC field defined in the API documentation for the COM object you are integrating.

WAUE_AddCC()

Arguments: mail object, email address

Returns: mail object

Corresponding global variable: gAddCC (optional)

This function is called for every attachment in the "CC" grid control when the Universal Email code runs. It should be used to call the method, add to the property, or add to the global variable for the CC field defined in the API documentation for the COM object you are integrating.

WAUE_AddRecipient()

Arguments: mail object, email address

Returns: mail object

Corresponding global variable: gAddRecipient (optional)

This function is called for every attachment in the "To" grid control when the Universal Email code runs. It should be used to call the method, add to the property, or add to the global variable for the to field defined in the API documentation for the COM object you are integrating.

WAUE_BodyFormat()

Arguments: mail object, body format

Returns: mail object

Corresponding global variable: gBodyFormat (optional)

This function is called to set the body format depending on your selection in the "Body" tab of the UI. Body format is 0 for HTML, 1 for Plain Text. This corresponds to how you would like to send the text that is in the body text area.

WAUE_Definition()

Arguments: any defn fields in the DEFUI

Returns: mail object

Corresponding global variable: gDefinition (optional)

This function is called to define the email object. The number of arguments must match the number of entries from you DEFUI tag.

Example:
This is an example of the two supporting functions for an email object that requires extra properties set from the DEFUI. Any values passed through the DEFUI will be added to the function call. Server-Side code will be translated properly from ANY DEFUI entry fields. The number of arguments for this function MUST match the "entries" property of the corresponding DEFUI tag.

```
function WAUE_Definition(Host,Port) {  
    retVal = Server.CreateObject("SampleMail ");  
    retVal.Host = Host;  
    retVal.Port = Port;  
    return retVal;  
}
```

```
function WAUE_Definition(Host,Port)  
    retVal = Server.CreateObject("SampleMail")  
    retVal.Host = Host  
    retVal.Port = Port  
    WAUE_Definition = retVal  
end function
```

WAUE_SendMail()

Arguments: Mail Object, Attachments, BCC, CC, To, Importance, From, Subject, Body

Returns: nothing

Does Not Support Global Variable

This function is used to send emails. All of the variables set in the mail object are passed to the WAUE_SendMail() function in the order listed above. Any properties not set in the other helper functions should be set here.

Example 1:

CDO Mail requires five arguments to send the email. Two of the properties (BCC, and CC) were not able to be set in the other helper function. Therefore, the WAUE_SendMail() function defined for CDO Mail is:

```
function WAUE_SendMail(mailObj, mailAttachments, mailBCC, mailCC, mailTo,  
                        mailImportance, mailFrom, mailSubject, mailBody) {  
    if (mailCC != "") {  
        mailObj.CC = mailCC;  
    }  
  
    if (mailBCC != "") {  
        mailObj.BCC = mailBCC;  
    }  
  
    mailObj.Send(mailFrom, mailTo, mailSubject, mailBody, mailImportance);  
}
```

Note that the function ignores any of the variables that have been manipulated in the other helper files.

For those email objects that support multipart email messages, the body will be sent as one string that is easily split on the <multipart> tag. Once split into a two member array the first member is the text version the second is the HTML version.

Example 2:

AspEmail uses the AltBody parameter for the text version of a multipart message, and Body for the HTML version. Here is an excerpt from the SendMail function for AspEmail:

```
function WAUE_SendMail(mailObj,mailAttachments,mailBCC,mailCC,mailTo,  
                        mailImportance,mailFrom,mailSubject,mailBody) {  
    ...  
    if (gBodyFormat == 2) {  
        mailBodyArray = mailBody.split("<multipartbreak>");
```

```
    mailBody = mailBodyArray[1];
    mailObj.AltBody = mailBodyArray[0];
}
mailObj.Body = mailBody;
...
}
```

WAUE_SetImportance()

Arguments: mail object, importance

Returns: mail object

Corresponding global variable: gSetImportance (optional)

This function is called to set the importance of the Email when it is sent. The importance value is taken from the list or entryfield based on the user's selection. The dropdown list importance will be passed as a number if selected from the dropdown list.

- 0) Low
 - 1) Normal
 - 2) High
-