

Sitecore Forms Module Reference manual

Reference manual for administrators and developers.

1 Contents

1	Contents.....	2
2	Functions.....	3
2.1	ChangeAttribute.....	3
2.2	ExtranetLogin.....	4
2.3	ExtranetLogout.....	4
2.4	FillList.....	5
2.5	ClearList.....	5
2.6	HideControl.....	6
2.7	ShowControl.....	6
2.8	MailinglistSubscribe.....	7
2.9	MailinglistUnsubscribe.....	7
2.10	MailingListGetMailingLists.....	8
2.11	PageControlChangePage.....	10
2.12	PopupOpen.....	11
2.13	Redirect.....	12
2.14	ReadFromDatabase.....	13
2.15	SaveToDatabase.....	15
2.16	SaveToNarrowTable.....	16
2.17	SendMail.....	17
2.18	SessionRead.....	18
2.19	SessionSave.....	18

2 Functions

The following functions are available in the standard package. If you have purchase the Pro version of Forms, you can develop your own functions.

2.1 ChangeAttribute

This function can change attributes on controls of certain types. It is commonly used to make text boxes read only, assign text to labels, or enable/disable controls.

Parameter name	Description	Default value
Control	The control to set attribute for	
Attribute	Which attribute to set	
NewValue	The new value of the attribute	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

Not all combinations of attributes work for all controls. You cannot set the readonly attribute on a label simply because it doesn't have such an attribute. The following combinations are valid:

Attribute	Applies to following controls
visible	All controls
enabled	Controls of the WebControl family
checked	Checkbox
text	Label, TextBox
selectedvalue	Controls of the ListControl family
innertext	Controls of the HTMLGenericControl family
readonly	TextBox

The **WebControl** family includes AdRotator, Calendar, CheckBoxList, CompareValidator, RadioButtonList, RangeValidator, RegularExpressionValidator, RequiredFieldValidator, ValidationSummary, Button, CheckBox, RadioButton, DropDownList, HyperLink, Image, ImageButton, GroupBox, Label, ListBox, TextBox

The **ListControl** family includes CheckBoxList, RadioButtonList, DropDownList, ListBox,

If you attempt to change an invalid attribute on a control, an exception is thrown.

2.2 ExtranetLogin

This function logs into Sitecore extranet.

Parameter name	Description	Default value
Domain	The domain to log into. Leave blank to log into current domain	Extranet
Username	Name of extranet user	
Password	Password of extranet user	
RedirectSuccess	The URL to redirect to if login is successful. Leave blank if you do not wish to redirect.	
FailureControl	The control to make visible if login is unsuccessful	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

The ExtranetLogin workflow is that it attempts to login to Extranet with the username and password. If the login is successful, a redirect to RedirectSuccess URL is done, if the string is not empty. If the login fails, the visible attribute on the FailureControl is set to true.

2.3 ExtranetLogout

This function logs the current user out of Sitecore extranet.

Parameter name	Description	Default value
Domain	The domain to log into. Leave blank to log into current domain	Extranet

2.4 FillList

This function will fill a control with the values of sub nodes from Sitecore. It is commonly used to fill drop down boxes or list views with selectable values.

Parameter name	Description	Default value
Control	The control to populate	
Sitecorepath	The path to the parent of the sitecore nodes that you wish to use as data source	/sitecore/content/home
ValueField	The name of the field that contains the values	@name
TextField	The name of the field that contains the texts	@name
FillOnPostBack	Should list be filled on post back	Yes
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

Some controls can contain a different value from the value being shown on screen (drop down boxes and list controls for example). The ValueField contains the value and TextField contain the text being displayed.

The FillOnPostBack feature allows you to control when the list should be filled. Whenever you press a button, the form is reloaded (a post back). If you always fill the list, the same items will be added to the list each time the post back occurs.

As a general rule: If you use FillList on an OnClick() event, set FillOnPostBack to Yes. If you use FillList on an OnLoad() event, set FillOnPostBack to false.

Use @name if you wish to get the node name.
Use @id if you wish to get the GUID of the node.

2.5 ClearList

This function clears the contents of a control, usually a list or a dropdown box.

Parameter name	Description	Default value
Control	The control to clear	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

2.6 HideControl

This function hides a control.

It is commonly used to make controls disappear on certain events. For example, if a user has 2 options, and the first option requires less text boxes for input, you can use this function to hide the text boxes when user selects option 1.

Parameter name	Description	Default value
Control	The control to hide	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

When HideControl is called, the display style is set to "none".

2.7 ShowControl

This function shows a control.

It is commonly used to make controls appear on certain events. For example, if a user has 2 options, and the second option requires more text boxes for input, you can use this function to enable the text boxes when user selects option 2.

Parameter name	Description	Default value
Control	The control to show	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

When ShowControl is called, the display style is set to "block" and the visible attribute is set to true.

2.8 MailinglistSubscribe

This function allows users to subscribe to the Sitecore Mailinglist.

NOTE! This requires the Sitecore Mailinglist to be installed.

Parameter name	Description	Default value
Name	Name of subscriber	
Email	Email of subscriber	
Mailinglists	Pipe () separated list of mailing list GUIDS	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

To get a pipe separated list of mailing list GUIDS, use the FillList function to assign GUIDS to a checkbox list. Assign "@name" to TextField. Set SitecorePath to "/sitecore/content/modules/mailling list/mailling lists" (default mailing list path). Set the ValueField to "@id".

In order to use the mailing list, you must add the following to the "sites" part in web.config:

```
<site
  name="website"
  virtualFolder="/"
  physicalFolder="/"
  rootPath="/sitecore/content"
  startItem="/home"
  language="en"
  database="web"
  content="master" ← Must be added
  domain="extranet"
  allowDebug="true"
  cacheHtml="true" />
```

2.9 MailinglistUnsubscribe

This function allows users to unsubscribe from the Sitecore Mailinglist.

NOTE! This requires the Sitecore Mailinglist to be installed.

Parameter name	Description	Default value
Email	Email of subscriber	
Mailinglists	Pipe () separated list of mailing list GUIDS	
UnsubscribeAll	Set to TRUE if you wish to unsubscribe from all mailing lists	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

If you set the UnsubscribeAll to TRUE, you can leave the Mailinglists parameter empty, as it won't be used.

In order to use the mailing list, you must add the following to the "sites" part in web.config:

```
<site
  name="website"
  virtualFolder="/"
  physicalFolder="/"
  rootPath="/sitecore/content"
  startItem="/home"
  language="en"
  database="web"
  content="master" ← Must be added
  domain="extranet"
  allowDebug="true"
  cacheHtml="true" />
```

2.10 MailingListGetMailingLists

This function allows you to fill a list with existing mailing lists.

NOTE! This requires the Sitecore Mailinglist to be installed.

Parameter name	Description	Default value
Control	The control to fill. Must be a list control	
IncludeDescription	True if list description should be included	False
FillOnPostBack	Should list be filled on post back	Yes
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

If you choose to include the mailing list description, the list text will be [name] – [description], for example:

Sitecore mailing list – News and updates

The FillOnPostBack feature allows you to control when the list should be filled. Whenever you press a button, the form is reloaded (a post back). If you always fill the list, the same items will be added to the list each time the post back occurs.

As a general rule: If you use FillList on an OnClick() event, set FillOnPostBack to Yes. If you use FillList on an OnLoad() event, set FillOnPostBack to false.

In order to use the mailing list, you must add the following to the "sites" part in web.config:

```
<site
  name="website"
  virtualFolder="/"
  physicalFolder="/"
```



```
rootPath="/sitecore/content"  
startItem="/home"  
language="en"  
database="web"  
content="master" ← Must be added  
domain="extranet"  
allowDebug="true"  
cacheHtml="true" />
```

2.11 PageControlChangePage

This function is used with the PageControl custom control to switch between the subpages inside a PageControl.

Parameter name	Description	Default value
PageControl	The page control	
Page	The page to change to, First, Last, Next or Previous	
Index	If you don't choose a page to change to, you can specify a absolute page index	
ButtonPrev	Select the button you use as the "Previous" button and it will be dimmed if the current page is the first.	
ButtonNext	Select the button you use as the "Next" button and it will be dimmed if the current page is the last.	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

In the Page field, you can specify a relative page to change to, the first page, the last page, the next page or the previous page.

If you leave the Page field blank, you can specify an absolute index of the page you wish to change to.

If you changes to an index of a page that does not exist, the page index will be set to -1 and no page is shown.

If you are on the last page, and you choose "Next", the page will not change. The same goes for the first page.

Before the page control changes page, the OnSave() event of the current sub form is called. Use this event to save the form data either in a session variable, or in a database.

Please note that the OnSave() event occurs after the OnLoad() event of the form.

2.12 PopupOpen

This function opens a popup window.

Parameter name	Description	Default value
Width	Window width	640
Height	Window height	480
Left	Window left position	100
Top	Window top position	100
Toolbar	Show toolbar?	No
Location	Show location bar?	No
Directories	Show popular websites bar?	No
Status	Show status bar?	No
Menubar	Show menu bar?	No
Scrollbars	Show scroll bars?	No
Resizable	Is window resizable?	No
Dependent	(Netscape only) Close the popup window when calling window closes?	No
Url	The Url of the page to show in the popup window	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

The URL parameter can be an absolute URL (<http://www.sitecore.net>) or a relative URL (/mypages/mypage.aspx).

2.13 Redirect

This function redirects the browser to a new web page. It is commonly used when users has pressed a [Submit] button, to redirect to a "thank you for submitting" page.

Parameter name	Description	Default value
Url	Url of the page to redirect to	
EndResponse	Indicates whether execution of the current page should terminate.	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

2.14 ReadFromDatabase

This function can read from any database that has an OleDb connection (Access, SQL Server ...) and fill a control with the data.

It is commonly used to fill list controls or dropdown boxes with values from an external data source. This function is the database equivalent to the FillList function.

Parameter name	Description	Default value
Destination	The name of the destination control	
SQL	The SQL statement that reads from the database	
FillOnPostBack	Should list be filled on post back	Yes
HtmlDecode	Should data be html decoded before added to destination?	Yes
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

The SQL statement can select from up to 2 fields. Controls that can have 2 values (list boxes, dropdown boxes) will have the first value assigned as the text and the second value assigned as the value. Controls that can have 1 value (text boxes, check boxes ...) will only read the first value.

An example SQL statement could be:

```
select GiftName, id from Gifts
```

If assigned to a dropdown box, the value from GiftName would be the name in the box. The value from id would be the value.

If you read from a database table with several records into a text box, the contents of the text box will be the concatenated values.

The **SQL** statement must be formatted correctly. Use {} to encapsulate the names of the controls where the values must come from.

Take the member database as an example: My form contains an eName control, and a Submit button. On the submit button I add the ReadFromDatabase function and gives the following SQL string as parameter:

```
select address from MEMBERS where name = '{eName}'
```

The function will replace the {eName} string with the value from the eName control.

If you by accident write an invalid control name, the value will be empty.

Use the **HtmlDecode** parameter to decode the database contents before adding it to the destination control.

The **FillOnPostBack** feature allows you to control when the list should be filled. Whenever you press a button, the form is reloaded (a post back). If you always fill the list, the same items will be added to the list each time the post back occurs.

As a general rule: If you use ReadFromDatabase on an OnClick event, set FillOnPostBack to Yes. If you use ReadFromDatabase on an OnLoad event, set FillOnPostBack to false.

The function relies on a setting in the web.config:

FormsDatabaseConnectionString

The connection string for the database to read and write to.

2.15 SaveToDatabase

This function can write values from controls on your form to any database that has an OleDb connection (Access, SQL Server ...).

It is commonly used to save data entered on the form to an external data source. For example, if you have a members database, you can use this function to save the contents of the form to the database.

Parameter name	Description	Default value
SQL	SQL insert statement	
HtmlEncode	Should data be html encoded before added to database?	Yes
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

The **SQL** statement must be formatted correctly. Use {} to encapsulate the names of the controls where the values must come from.

Take the member database as an example: My form contains an eName and eAddress control, and a Submit button. On the submit button I add the SaveToDatabase function and gives the following SQL string as parameter:

```
insert into MEMBERS (name, address) values ('{eName}','{eAddress}')
```

The function will replace the {eName} string with the value from the eName control, and {eAddress} with the value from the eAddress control.

If you by accident write an invalid control name, the value will be empty.

Use the **HtmlEncode** parameter to Html encode control values. Using HtmlEncode ensures that " and ` can be saved. Furthermore it reduces the possibility of SQL server injection and HTML injection attacks. HtmlEncode replaces " with ", < with < and so on.

The function relies on a setting in the web.config:

FormsDatabaseConnectionString

The connection string for the database to read and write to.

2.16 SaveToNarrowTable

Parameter name	Description	Default value
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

This function can read all values from all fields on your form and save the values to a standard narrow table provided by Forms. The table is located at /sitecore modules/Forms/Database/savedb.mdb.

To create a SQL version of the database, use the script located at:

```
\\sitecore modules\Forms\Data Viewer\sqlserver\databaseinstall.sql
```

The function can provide you with a quick database save option that can be used if you wish to save all data but does not worry about the table layout. It is especially useful when debugging or as an audit log.

The function relies on settings in the web.config:

FormsDatabaseMode

Specify "Access" if you wish to save to an Access database. Specify "SQLServer" if you wish to save to a SQL server.

FormsAccessConnectionString

The connection string for the Access database. If you do not specify any connection string, one will be provided to you, and it will expect to find an Access database at /sitecore modules/Forms/Database/savedb.mdb.

FormsSQLServerConnectionString

The connection string for the SQL Server. You must specify a connection string if you wish to use a SQL Server.

2.17 SendMail

This function sends a mail.

It is commonly used to get data from an order submission form, or with a "Contact Us" form.

Parameter name	Description	Default value
From	From email	
To	To email	
Subject	Subject of mail	
BodySitecorePath	The path to a node containing the body of the mail message	
BodyField	The name of the HTML field containing the body of the message.	
MailServer	Name of mailserver to send mail.	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

Messages are always send as HTML mails.

The MailServer field defines which mailserver to use when sending mails:

- Enter a mailserver to use.
- If blank, the mailserver is looked up in web.config, setting "SMTPserver".
- If this is also blank, the default mailserver is used.

The body of the message is retrieved from a Sitecore node as defined in BodySitecorePath (the path to the node) and BodyField (the name of the field).

Body and Subject can be encoded to contain values from controls. Use {} to encapsulate the names of the controls.

For example: I have a submission form with the text fields eName and eAddress. If I would like to have the values of these fields in my email, I would define my email body like this:

"The following person has submitted his data: {eName} {eAddress}".

The {eName} and {eAddress} is replaced with the values of eName and eAddress controls respectively.

If you by accident write an invalid control name, the value will be empty.

2.18 SessionRead

This function reads the value of a session variable and stores it in a control. It is commonly used to retrieve data shared between several forms, or when using sub forms.

Parameter name	Description	Default value
Control	The control to store value in	
SessionVariable	The name of the session variable	
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute

2.19 SessionSave

This function saves the values of all controls as session variables. The names of the variables are the control ID.

It is commonly used to share data between several forms, or when using sub forms.

Parameter name	Description	Default value
Validation	Determines if validation should effect the execution of the function (only used when client side validation is disabled)	Always Execute