



Great Ideas. Always Flowing.™

Tailored HTML

User Guide

Table of contents:

1	INTRODUCTION	4
1.1	What's new in v3.0?	5
2	INSTALLATION PROCEDURE	6
3	ADDING TAILORED HTML MODULE TO A PAGE	12
4	TAILORED HTML MAIN MENU	14
5	ADDING NEW CONTENT	15
5.1	Defining content for authorized users	16
5.2	Defining content for unauthorized users	17
5.3	Managing the SEO parameters	18
6	MANAGING TAILORED HTML MODULE SETTINGS	20
6.1	Managing the Languages	21
6.2	Managing the Tokens	22
6.3	Creating a Querystring Token Type	23
6.3.1	Editing a Querystring Token Type	24
6.3.2	Deleting a Querystring Token Type	25
6.4	Creating an SQL Token Type	26
6.4.1	Editing an SQL Token Type	27
6.4.2	Deleting an SQL Token Type	28
6.5	Creating a Tailored Initial SQL Databind	29
7	LICENSING YOUR COPY OF TAILORED HTML	31
8	DELETING AN INSTANCE OF THE "TAILORED HTML" MODULE	32

List of figures:

Figure 1:	Installation procedure (step 1/9)	6
Figure 2:	Installation procedure (step 2/9)	7
Figure 3:	Installation procedure (step 3/9)	7
Figure 4:	Installation procedure (step 4/9)	8
Figure 5:	Installation procedure (step 5/9)	8
Figure 6:	Installation procedure (step 6/9)	9
Figure 7:	Installation procedure (step 7/9)	10
Figure 8:	Installation procedure (step 8/9)	10
Figure 9:	Installation procedure (step 9/9)	11
Figure 10:	Adding a module to a page	12
Figure 11:	Opening the module main menu	12
Figure 12:	Opening the main menu	14
Figure 13:	Choosing option "Add Content"	15
Figure 14:	Options for adding the new content	15
Figure 15:	Defining content for authorized users	16
Figure 16:	Example of the content displayed to the authorized user	17
Figure 17:	Defining content for unauthorized users	17
Figure 18:	Accessing the SEO parameters	18
Figure 19:	Managing the SEO parameters	19
Figure 20:	Choosing option "Settings"	20
Figure 21:	Choosing option "Tailored Text/HTML Settings"	20
Figure 22:	Available Tailored HTML settings	21
Figure 23:	Managing the Languages	21
Figure 24:	Options for managing "Text/HTML Settings"	22
Figure 25:	Setting up session token type	23
Figure 26:	Example of the Querystring token	24
Figure 27:	Editing a Querystring Token Type (step 1/2)	24
Figure 28:	Editing a Querystring Token Type (step 2/2)	25
Figure 29:	Deleting a Querystring Token Type	25
Figure 30:	Creating a SQL Token Type	26
Figure 31:	Example of the Sql token	27
Figure 32:	Editing an SQL Token Type (step 1/2)	27

Figure 33: Editing an SQL Token Type (step 2/2).....	28
Figure 34: Deleting an SQL Token Type	28
Figure 35: Creating a Tailored Initial SQL Databind.....	29
Figure 36: Licensing your copy of Tailored HTML module (step 1/2)	31
Figure 37: Licensing your copy of Tailored HTML module (step 2/2)	31
Figure 38: Deleting the module (step 1/2).....	32
Figure 39: Deleting the module (step 2/2).....	32

1 INTRODUCTION

The “Tailored HTML” module allows you to add text/html to your web site that can be different for unauthorized users, registered users, and even individual users specifically.

This module allows you to include a client’s user or contact information within a text or html module. In addition, information about the date their subscription expires; or maybe just address them directly by name throughout your web site.

Tailored HTML module includes over fourteen default parameters or tokens that can be included within the text or HTML of your module.

Note: this manual assumes prior experience with DotNetNuke platform. For further information about DotNetNuke, please consult the DotNetNuke manual.

Module Features:

- Add rich user specific text/html content to your DotNetNuke portal
- Enter text for anonymous users, and registered users, add tokens/parameters throughout your text that will then be replaced when the user is signed in
- Creating custom parameters/tokens based on querystring variables passed to the page, session variables, or even dynamic SQL parameters
- **Example default parameters include**
 - **First Name** - SuperUser
 - **Last Name** - Account
 - **Full Name** -SuperUser Account
 - **Display Name** - SuperUser Account
 - **Email Address** – host
 - **UserName** - host
 - **Portal Name** - Data Springs Inc. - DotNetnuke Modules
- **Example custom parameters include**
 - **Querystring parameters** - Pass querystring values to the page and the parameter within the text/html will be replaced with the querystring value
 - **Session parameters** - Allow any session variable to be easily displayed to the user
 - **SQL parameters** - With SQL parameters your options are virtually endless; Easily pull from any database table information. Example: Select BirthDate from your customizable where UserID = 1. This feature can also be utilized with Dynamic Registration.
- Ability to set the default value if no data is present

1.1 What's new in v3.0?

- New DotNetNuke 6.x Interface - Including updated pop up interfaces, DNN 6 stylesheet classes, and updated GUI.
- New features for extended tokens with references for the current portal alias, URL, and if the portal is a child or parent - use these in SQL queries to render back specific content per portal alias / web site url!
- New SEO Enhancements! Now all of your tokens for querystring, session, and SQL can create customized SEO pages. Include all tokens within...
 - Page Title
 - Page Keywords
 - Page Description

2 INSTALLATION PROCEDURE

Included in your download are either two (If you only purchased the PA) or four (If you purchased the PA and Source Code) zip files for two different versions of DNN (3.x and 4.x).

One zip file is the source code to the application, and the other zip file is the module which you can upload to your site.

When you extract the files, you will notice that it extracts four zip files (**note:** only two if you just purchased the PA).

- DataSprings_TailoredHTML_ModulePAInstallationFile_DNN45Plus_30_00_00_StandardEdition.zip
- DataSprings_TailoredHTML_ModulePAInstallationFile_DNN45Plus_30_00_00_EnterpriseEdition.zip
- DataSprings_TailoredHTML_ModulePAInstallationFile_DNN45Plus_30_00_00_DeveloperEdition
- DataSprings_TailoredHTML_V30_ModuleSourceCode_ReviewLicenseAgreement.zip

In order to install your “Tailored HTML” module, login with an account to your DNN site as a host or administrator account. Once logged in, navigate to the Host section and click “Module Definitions”.

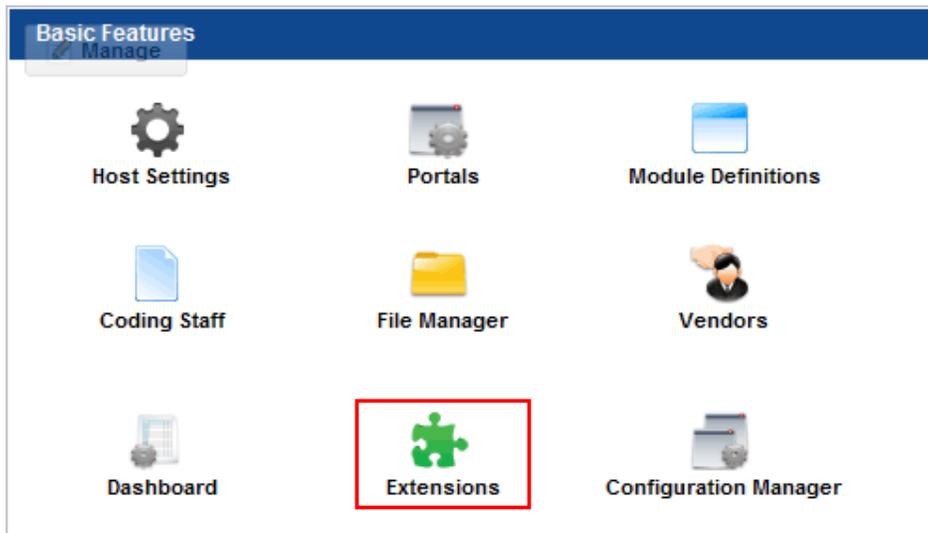


Figure 1: Installation procedure (step 1/9)

The following screen will be displayed.

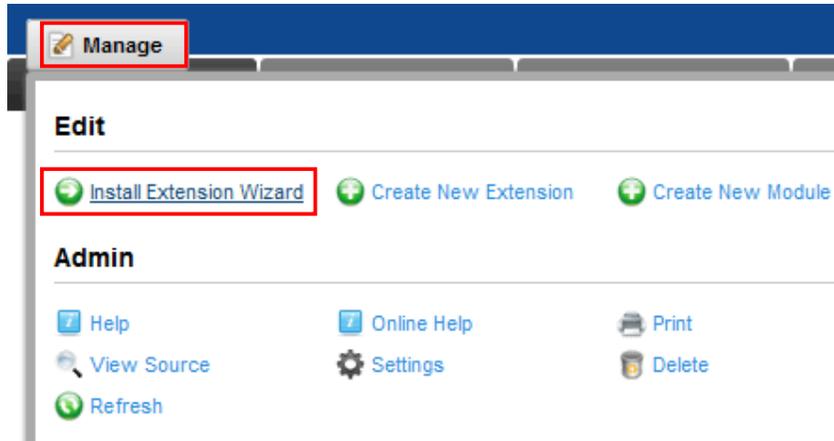


Figure 2: Installation procedure (step 2/9)

Click “Manage” and then choose the “Install Extension Wizard” option from the menu. The following screen will be displayed.

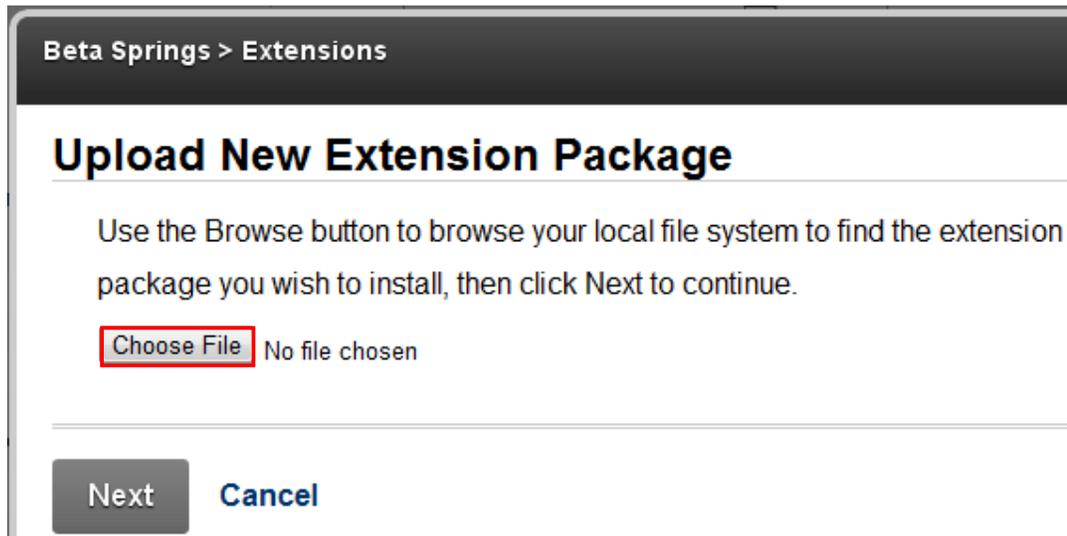


Figure 3: Installation procedure (step 3/9)

Click the “Choose File” to start browsing your local file system and locating the extension package you want to install.

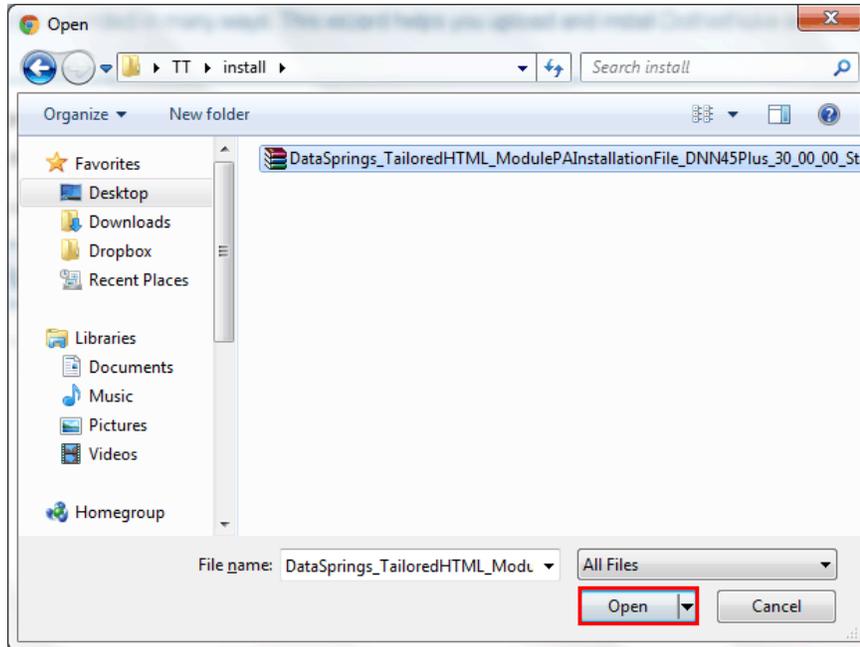


Figure 4: Installation procedure (step 4/9)

Locate the installation file and click “Open” button. The following screen will be displayed.

Beta Springs > Extensions

Upload New Extension Package

Use the Browse button to browse your local file system to find the extension package you wish to install, then click Next to continue.

DataSprings...Edition.zip

Figure 5: Installation procedure (step 5/9)

Click “Next” to continue installing the module and the following page will be displayed.

Beta Springs > Extensions

Name:  Tailored HTML

Type:  Module

Friendly Name:  Tailored HTML

Icon File: 

Description:  Tailored Text allows you to add text/html to your anonymous users, registered users, and even s
Tailored Text can be unique per language setup for content localization within each text module c

Version:  3.0.0

Owner:  Data Springs, Inc

Organization:  Data Springs, Inc

Url:  <http://www.datasprings.com/products/dnn-modu>

Email Address:  dnnsupport@datasprings.com

Next

Cancel

Figure 6: Installation procedure (step 6/9)

Review the module information in this page and click “Next” to continue. The following page will be displayed.

Beta Springs > Extensions

Release Notes

You can review the Release Notes for this package.

Release Notes: The 3.0 release reflects new er specify all tokens (SQL, Querys title, page description, and pag specific updates related to DNI interface enhancements for DN

Next

Cancel

Figure 7: Installation procedure (step 7/9)

This page contains “Release Notes”. Review the release notes and click “Next” to continue. The following page will be displayed.

Beta Springs > Extensions

Accept License?

Next

Cancel

Figure 8: Installation procedure (step 8/9)

Scroll to the bottom of the page; tick the “Accept License” checkbox and click “Next” to continue. The “Package Installation Report” will be displayed containing information on whether the installation was successful or not at the bottom.

Beta Springs > Extensions

Package Installation Report

 See below for the results of the package installation

Info Created - DS_icon1.gif
Info Creating backup of previous version - EditTailoredHTML.ascx
Info Component installed successfully - File
Info Installation committed
Info Installation successful. - Tailored HTML
Info Deleted temporary install folder
EndJob Installation successful.

Return

Figure 9: Installation procedure (step 9/9)

After installing the module, you can add it to a desired page – see section 3.

Note: please keep track of any errors that appear during the installation; these errors can be helpful if your module has problems.

3 ADDING TAILORED HTML MODULE TO A PAGE

In order to add “Tailored HTML” module to a desired page follow these steps:

1. Select the “Add New Module” radio button
2. Choose “Tailored HTML” from the “Module” pull down menu
3. Click on the “Add Module”

Figure 10: Adding a module to a page

The “Tailored HTML” module will be added to the page.

Figure 11: Opening the module main menu

The following options are available:

- **Manage menu** – hover the module title to display the **Manage** option which will allow you to access the menu with all relevant settings (see section 4 for further information)
- **Licensing the module** – see section 7 for further information on licensing your copy of Tailored HTML module
- **Setup Initial Settings** – clicking this link will take you to a page where you can modify the initial module settings (see section 6)
- **Setup Template** – click this link to start defining the content for both authorized and unauthorized users (see section 5)

4 TAILORED HTML MAIN MENU

This section of the document will describe the options available inside the “Tailored HTML” main menu. In order to access the main menu, place your mouse on the module title and the **Manage** option will be displayed. Click **Manage** to open the main menu.

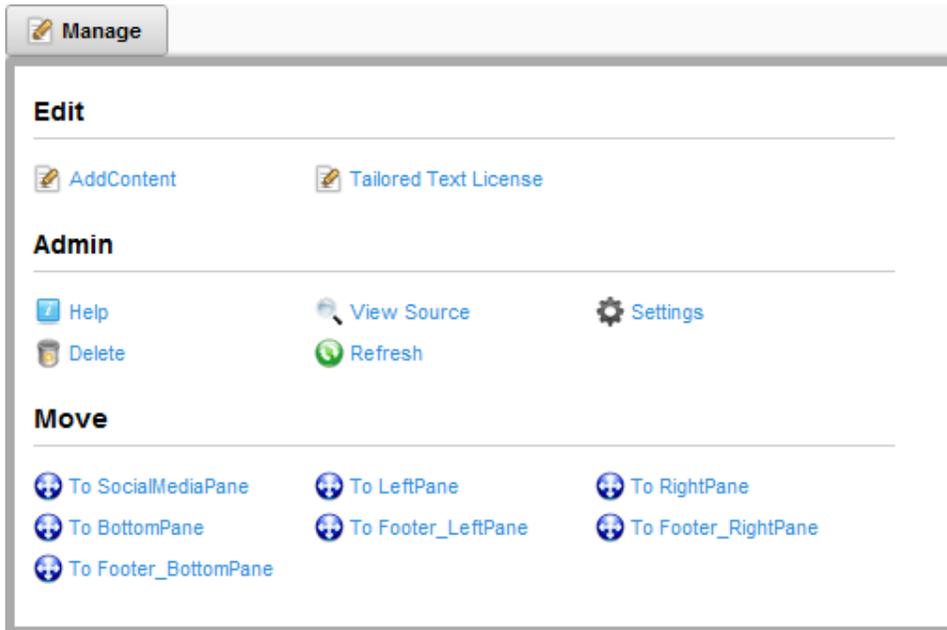


Figure 12: Opening the main menu

The following options are available inside the main menu:

- **Edit** – the following editing options are available:
 - **Add Content** – option for adding new content (see section 5)
 - **Tailored Text License** - see section 7 for further information on licensing your copy of Tailored HTML module
- **Admin** – the following administrative options are available:
 - **Help** – this user guide
 - **Settings** – option for managing “Tailored HTML Settings” (see section 6)
 - **Delete** – the option for deleting the module instance (see section 8)
- **Move** – options for placing the module in the desired part of the page:

5 ADDING NEW CONTENT

In order to start adding new content, choose option “Add Content” from the main menu.

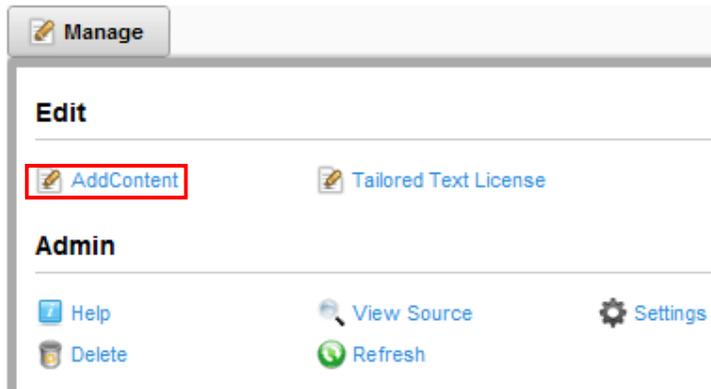


Figure 13: Choosing option “Add Content”

The following page will be displayed.

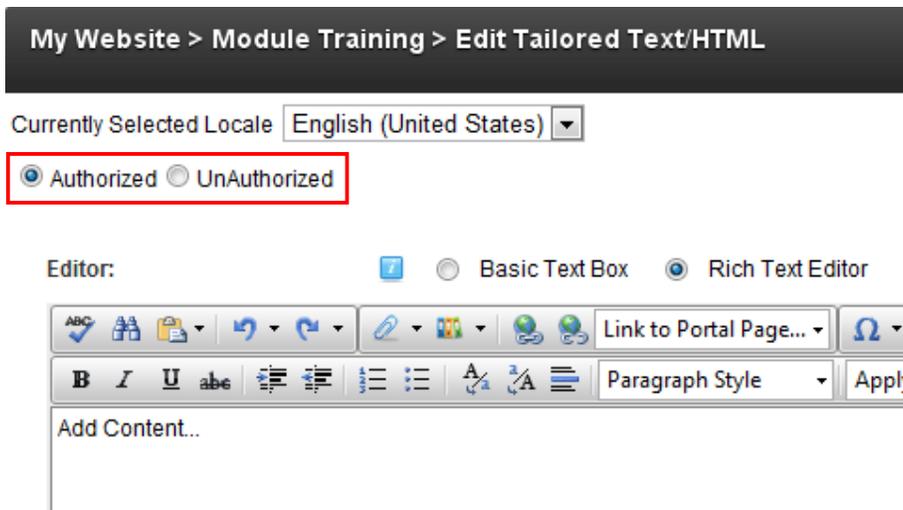


Figure 14: Options for adding the new content

The upper part of the page contains options for choosing the desired type of content:

- **Authorized** – which is meant for defining content displayed to the authorized users i.e. text that will be displayed to the user after he logs in (see section 5.1)
- **Unauthorized** – unauthorized i.e. text displayed to visitors which do not log in (see section 5.2)

5.1 Defining content for authorized users

In order to start defining content which will be displayed to the authorized users, select **Authorized** in the upper part of the page.

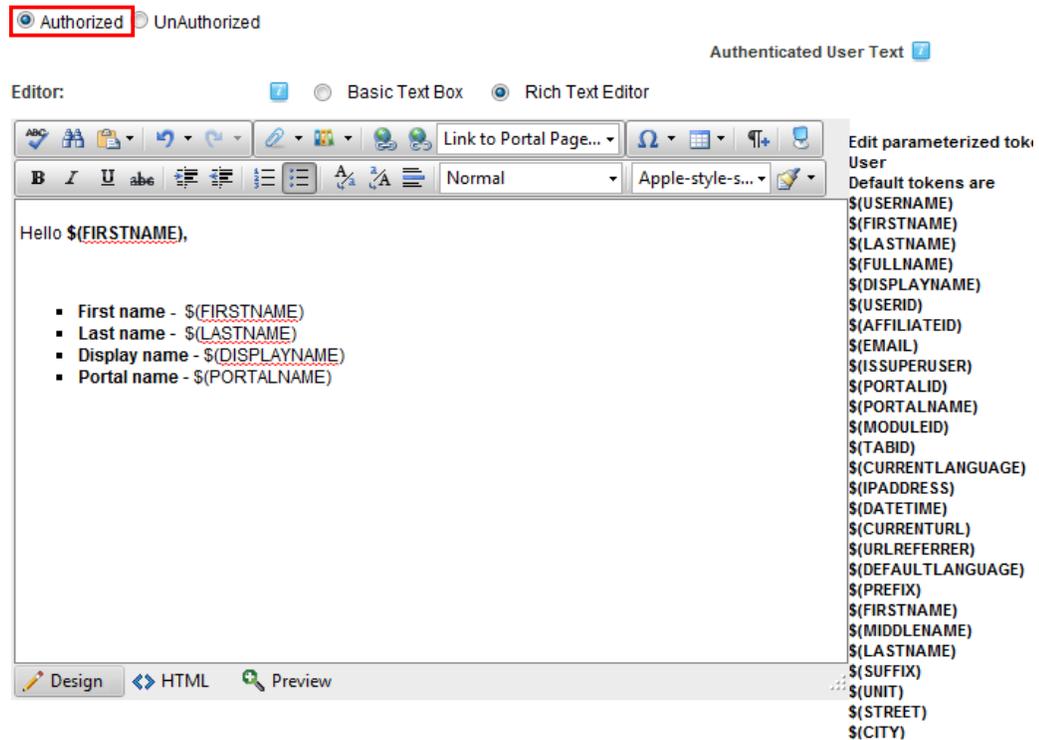


Figure 15: Defining content for authorized users

The following parameters and options are available inside this screen:

- **Text area** – this part of the screen is used for entering your text and combining it with the available tokens displayed on the right
- **Tokens** – a list of available tokens which you can use to address your authorized users

You can use the text area to combine your text with the tokens which will be replaced with information from the user's profile, so the result will be addressing the user by his first name.

The result as displayed in the screen below is accomplished by typing "Hello" followed by the "\$\$(FIRSTNAME)" token.

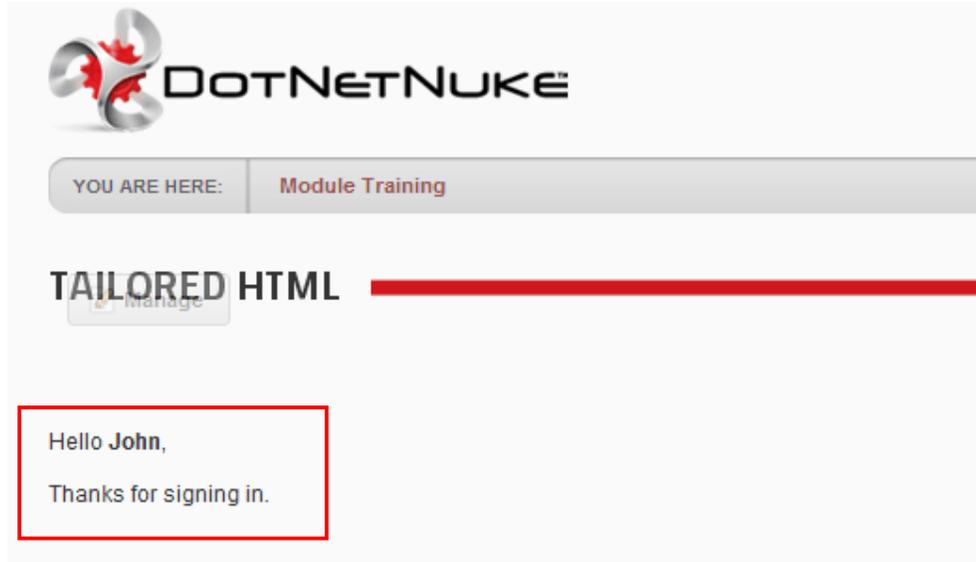


Figure 16: Example of the content displayed to the authorized user

5.2 Defining content for unauthorized users

The content area for unauthorized users can be used for easily adding new content to your website by using the WYSIWYG editor and thus formatting the layout of the page.

This area can also be used in combination with the tokens on the right to display general information about your system such as **\$(PORTALNAME)** in case you wish to display the name of your portal, **\$(DATETIME)** to always display the current date and time, etc.

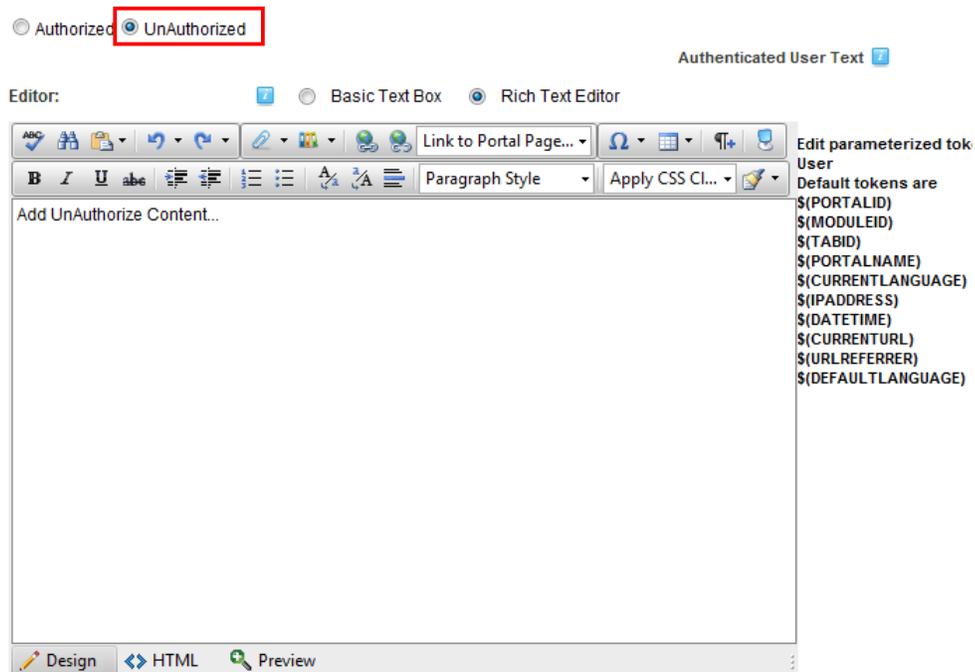


Figure 17: Defining content for unauthorized users

5.3 Managing the SEO parameters

The “Tailored Text SEO Tools” allow the page title, description, and keywords to include tokenized data from all available Tailored Text/HTML tokens including standard tokens and tokens from querying, session, and also SQL bound data.

If you were using Tailored Text to pull up an individual record from the database, you might want to reference columns from within that database for the page title, page description, and page keywords. This allows each individual record to have a unique page title and keywords/description for search engines to reference!

For example: Tailored Text was referencing a record within a directory of boats that might be for sale. You could have a title such as “Luxury Boat Directory - \$(BoatTitle)’ so that the title of the boat would be listed from that database record. You can utilize every token to create dynamic and very SEO friendly pages.

You can find the section for managing the SEO below the text area for entering content.

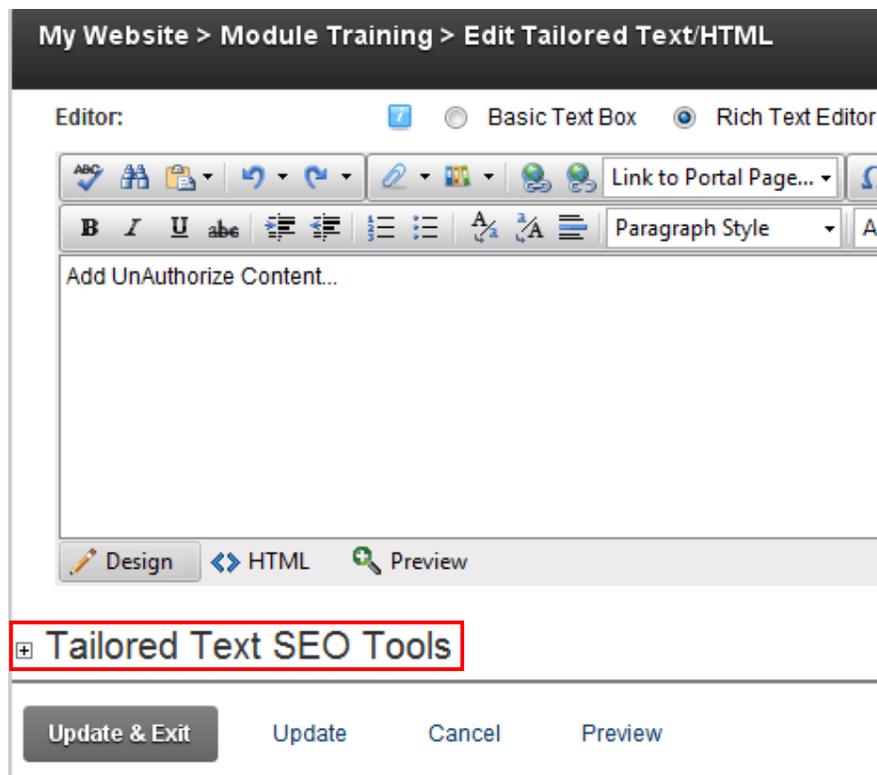


Figure 18: Accessing the SEO parameters

Click the + symbol to expand the page and display the available options.

Tailored Text SEO Tools

Alternate Title:

Alternate Description:

Alternate Keywords:

Search Summary (Optional):

Figure 19: Managing the SEO parameters

The following parameters are available:

- **Alternate Title** – enter the alternate title for the page in case you wish to use a different one than the one currently defined
- **Alternate Description** – enter the page description; this can be taken by the search engine to display your page description within the search results
- **Alternate Keywords** – specify the desired keywords (**note**: the keywords should be separated by comma)
- **Search Summary (optional)** – enter the search items which will be displayed as the “Summary” within the DotNetNuke Search functionality.

After setting the desired parameters, click **Update & Exit** to save the changes.

6 MANAGING TAILORED HTML MODULE SETTINGS

In order to start managing “Tailored HTML” module settings, choose option “Settings” from the main menu.

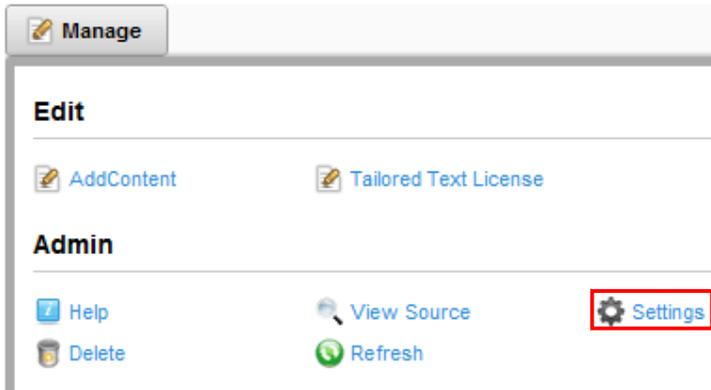


Figure 20: Choosing option "Settings"

The following screen will be displayed.

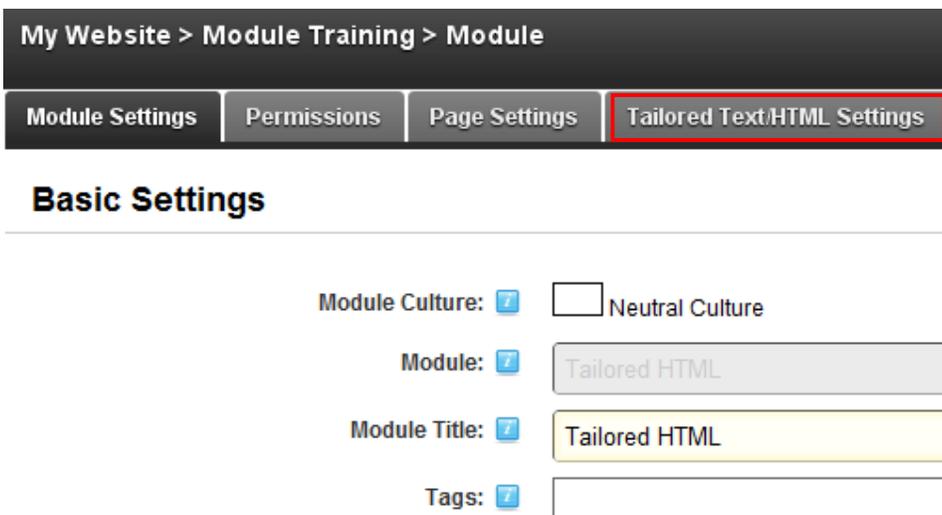


Figure 21: Choosing option "Tailored Text/HTML Settings"

Click the “Tailored HTML Settings” and the page with the available settings will be displayed.

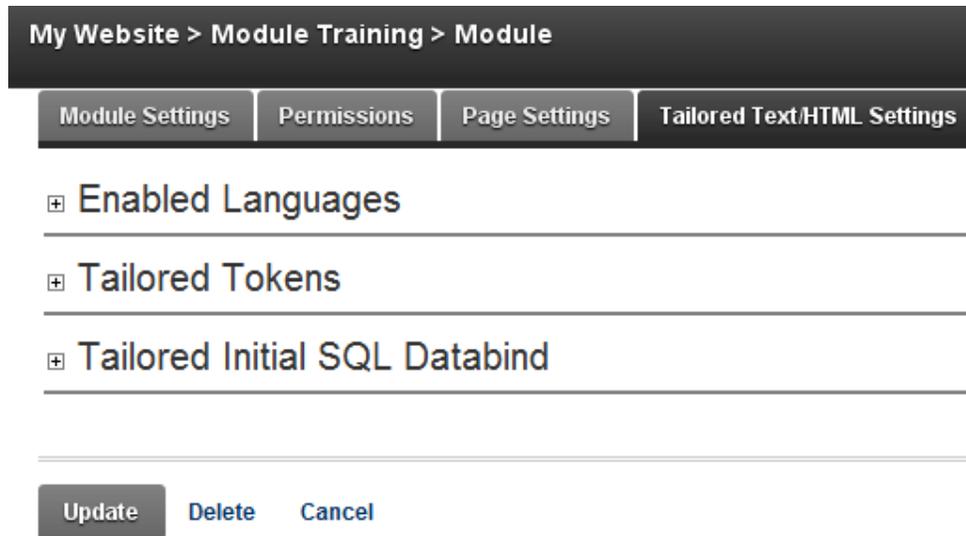


Figure 22: Available Tailored HTML settings

The following settings are available:

- **Enabled Languages** – the option for enabling the use of different languages in the module (see section 6.1)
- **Tailored Tokens** – the option for managing tailored tokens (see section 6.2)
- **Tailored Initial SQL Databind** – the option for setting up the initial SQL databind (see section 6.4)

6.1 Managing the Languages

The Tailored HTML module allows you to setup different languages which you can then display to either authorized or unauthorized users. At least one language setup and defined for the module to display properly.

In order to start managing the languages, choose “Settings” from the main menu and then click + next to the “Enabled Languages” label.

Enabled Languages

Select the languages that you would like to enable for Tailored Text. Tailored Text can be displayed for the site and also based on if the user is authorized or not. You must have at least one language enabled.

Add Language

Name	Key	
English (United States)	en-US	Delete

Figure 23: Managing the Languages

The following options are available:

- **Add** – the option for adding a new language; this pull down menu is populated by the languages which have been uploaded/enabled for the site under **Admin -> Languages**
- **Delete** – click to delete the desired language

6.2 Managing the Tokens

The Tailored HTML module allows you to create and manage custom tokens. In order to start managing the tokens, choose “Settings” from the main menu and then click + next to the “Tailored Tokens” label.

Tailored Tokens

Token	TokenType	TokenKey	DefaultText		
DirectoryUserID	Sql	select username from users where userid=\$(myuserid)	No data was returned for this userID	Edit	Delete
myuserid	QueryString	myuserid	No UserID Returned	Edit	Delete

Token 

TokenType  

TokenKey 

DefaultText 

[Submit](#)

Figure 24: Options for managing "Text/HTML Settings"

The following parameters are available inside this screen:

- **Token** – this is for the token name to identify the custom token
- **TokenType**
 - **Session** - this type of token will allow you to retrieve session variables based on the key. Session variables can be created with other DotNetNuke modules, applications, or website. Session variables are typically hidden from the user. If you are creating a session token variable you will store the variable with a name or key. Make sure to use the same name or key in the 'TokenKey' field below. To read more about storing session variables you can review our Dynamic Registration or Dynamic Forms module.
 - **QueryString** – this type of token will allow you to retrieve query string variables and insert them into your text/html. Querystring variables can be passed to the module and will be replaced.
 - **For example:** if you submitted a form from another module (for example Dynamic Forms) and selected the option to pass the value of that field to the querystring, the redirect URL would be <http://www.yoursite.com/default.aspx?TabID=YourTabID&YourFieldName=YourFieldValue>
 - A more realistic example would be to pass the users first name along a querystring variable and retrieve it with this token.
 - The URL would get passed as <http://www.yoursite.com/default.aspx?TabID=36&FirstName=John&LastName=Doe&Phone=555-555-1234>
 - The tokenkey would need to be setup for each variable you wish to include within Tailored HTML. For example if you want to include the parameter for the user's first name the key would be

firstname. You could then insert this parameter into the text.
Example, Dear \$(FirstName), thanks for submitting the form

- **SQL** – SQL tokens provide you the most complex and customized token configuration options by allowing you to pull information from any database table. Additionally SQL tokens can include ‘other’ tokens, so for example if you needed to pull information from a table you can include parameters from the standard, session, or querystring tokens. Note: SQL Tokens need to pull only one column and one row of data.
 - **Example:** Select Lastname from Users where UserID = \$(UserID)
 - **Example:** Select ExpireDate from SpecificTable where LastName = \$(LastName)
 - **Example:** Select count(*) from CustomerSales where CustomerID = \$(CustomQuerystringCustomerIDToken)
- **TokenKey** - this is the key or query (key for session and querystring, SQL statement for SQL tokens)
- **DefaultText** – this is the text that will be displayed if no results are returned from either the session, querystring, or SQL token.

6.3 Creating a Querystring Token Type

In order to create a Querystring Token Type, choose option “Settings” from the main menu and click on the “+ “ symbol next to the “Tailored HTML” label. The lower part of the page contains parameters for creating a token.

The screenshot shows a configuration form for a session token type. It consists of four rows, each with a label and a value field:

- Token:** A text input field containing the value "myuserid".
- TokenType:** A dropdown menu with "QueryString" selected.
- TokenKey:** A text input field containing the value "myuserid".
- DefaultText:** A text input field containing the value "No UserID Returned".

Below the fields is a blue "Submit" button.

Figure 25: Setting up session token type

The following parameters are available inside this screen:

- **Token** – enter the name of the token which will help you differentiate it from other tokens in the list
- **TokenType** – pull down menu for choosing the token type; choose “QueryString” option
- **TokenKey** – input field for entering token key (e.g. “myuserid”)
- **DefaultText** – input field for entering a default text which will be displayed in case no value has been retrieved (e.g. “No UserID Returned”)

After setting the desired parameters, click “Submit” to complete the procedure of creating a Querystring token. The following screen demonstrates the created token.

Tailored Text/HTML ▾

This demonstration will display several default and custom parameters for Tailored Text/HTML.

Example of default parameters:
 The user who is currently signed in is: agazibara
 The first name for the user currently signed in is: Aleksandar

Custom Parameters:

1. Querystring Token Example
 A custom token was setup as querystring parameter called MyUserID. Tailored Text/HTML can retrieve this custom querystring token if it is passed along the URL.

Example: <http://www.datasprings.com/default.aspx?TabID=732&MyUserID=700>
 The UserID retrieved for this custom token was: 700

2. SQL Token Example
 A custom token was setup as a SQL token. SQL can retrieve values via the SQL statement, additional parameters can be used that are default default parameters or custom parameters.

Example: This SQL query actually pulls the UserID that was passed along the querystring (you could also just use the standard default parmeter for userID as well but for this example we will display utilizing our other custom token). This particular query pulls the users response to a Dynamic question from the Dynamic Registration module.

Again, remember that this parameter requires a value to be passed along the querystring:

The user responded they they heard of Data Springs From: Snowcovered

Figure 26: Example of the Querystring token

6.3.1 Editing a Querystring Token Type

In order to edit a Querystring token type, choose option “edit” next to the desired token.

☐ Tailored Tokens

Token	TokenType	TokenKey	DefaultText		
DirectoryUserID	Sql	select username from users where userid=\$(myuserid)	No data was returned for this userID	Edit	Delete
myuserid	QueryString	myuserid	No UserID Returned	Edit	Delete

Figure 27: Editing a Querystring Token Type (step 1/2)

The following screen will be displayed.

☐ Tailored Tokens

Token	TokenType	TokenKey	DefaultText	
DirectoryUserID	Sql	select username 's where userid=\$(n	No data was returned for this userID	Edit
<input type="text" value="myuserid"/>	QueryString ▾	<input type="text" value="myuserid"/>	No UserID Returned	<input type="button" value="Update"/> <input type="button" value="Cancel"/>

Figure 28: Editing a Querystring Token Type (step 2/2)

Change the desired values inside this screen and click on the “Update” button to save the changes.

6.3.2 Deleting a Querystring Token Type

In order to delete a Querystring token type, choose option “Delete” next to the desired token.

☐ Tailored Tokens

Token	TokenType	TokenKey	DefaultText		
DirectoryUserID	Sql	select username from users where userid=\$(myuserid)	No data was returned for this userID	Edit	Delete
myuserid	QueryString	myuserid	No UserID Returned	Edit	<input type="button" value="Delete"/>

Figure 29: Deleting a Querystring Token Type

6.4 Creating an SQL Token Type

In order to create an SQL Token Type, choose option “Settings” from the main menu and click on the “+” symbol next to the “Tailored Tokens” label.

The lower part of the page contains parameters for adding a new token.

The screenshot shows a web form for creating an SQL token type. It contains the following elements:

- Token:** A text input field containing "DirectoryUserID".
- TokenType:** A dropdown menu with "Sql" selected.
- TokenKey:** A text area containing the SQL query: "select username from users where userid=\$(myuserid)".
- Test SQL Query:** A blue link text.
- DefaultText:** A text input field containing "No data was returned for this userID".
- Submit:** A blue button located below the DefaultText field.

Figure 30: Creating a SQL Token Type

The following parameters are available inside this screen:

- **Token** – enter the name of the token which will help you differentiate it from other tokens in the list
- **TokenType** – pull down menu for choosing the token type; choose “Sql” option
- **TokenKey** – input field for entering token key (**e.g.** “select Response from DynamicRegistration_QuestionResponse A Inner Join DynamicRegistration_Question B on A.DynamicQuestionID = B.DynamicQuestionID Where ShortFieldName = 'HeardAbout' AND UserID = \$(MyUserID)”)
- **DefaultText** – input field for entering a default text which will be displayed in case no value has been retrieved (**e.g.** “No data was returned for this userID.”)

After setting the desired parameters, click “Submit” to complete the procedure of creating a new SQL token. The following screen demonstrates the created token.

Tailored Text/HTML
▼

This demonstration will display several default and custom parameters for Tailored Text/HTML.

Example of default parameters:
 The user who is currently signed in is: agazibara
 The first name for the user currently signed in is: Aleksandar

Custom Parameters:

1. Querystring Token Example
 A custom token was setup as querystring parameter called MyUserID. Tailored Text/HTML can retrieve this custom querystring token if it is passed along the URL.
 Example: <http://www.datasprings.com/default.aspx?TabID=732&MyUserID=700>
 The UserID retrieved for this custom token was: 700

2. SQL Token Example
 A custom token was setup as a SQL token. SQL can retrieve values via the SQL statement, additional parameters can be used that are default default parameters or custom parameters.
 Example: This SQL query actually pulls the UserID that was passed along the querystring (you could also just use the standard default parameter for userID as well but for this example we will display utilizing our other custom token). This particular query pulls the users response to a Dynamic question from the Dynamic Registration module.
 Again, remember that this parameter requires a value to be passed along the querystring:
 The user responded they they heard of Data Springs From: Snowcovered

Figure 31: Example of the Sql token

6.4.1 Editing an SQL Token Type

In order to edit an SQL token type, choose option “edit” next to the desired token.

☐ Tailored Tokens

Token	TokenType	TokenKey	DefaultText		
DirectoryUserID	Sql	select username from users where userid=\$(myuserid)	No data was returned for this userID	Edit	Delete
myuserid	QueryString	myuserid	No UserID Returned	Edit	Delete

Figure 32: Editing an SQL Token Type (step 1/2)

The following screen will be displayed.

☐ Tailored Tokens

Token	TokenType	TokenKey	DefaultText	
DirectoryUserID	Sql	select username	No data was returned for	Update Cancel
myuserid	QueryString	myuserid	No UserID Returned	Edit

Figure 33: Editing an SQL Token Type (step 2/2)

Change the desired values inside this screen and click on the “Update” button to save the changes.

6.4.2 Deleting an SQL Token Type

In order to delete an SQL Token Type, choose option “Delete” next to the desired token.

☐ Tailored Tokens

Token	TokenType	TokenKey	DefaultText		
DirectoryUserID	Sql	select username from users where userid=\$(myuserid)	No data was returned for this userID	Edit	Delete
myuserid	QueryString	myuserid	No UserID Returned	Edit	Delete

Figure 34: Deleting an SQL Token Type

6.5 Creating a Tailored Initial SQL Databind

In order to create a tailored initial SQL databind, choose option “Settings” from the main menu and click on the “+ “ symbol next to the “Tailored Initial SQL Databind” label. The following page will be displayed.

☐ Tailored Initial SQL Databind

Enable initial SQL data bind: 

Enable debug mode?: 

Only enable when querystring value is present?: 

Only display module when 'TailoredSource' querystring value is present and a match?: 

Initial SQL Query: 

External Connection String (Optional): 

Figure 35: Creating a Tailored Initial SQL Databind

The following parameters are available:

- **Enable initial SQL data bind** – select to enable the tailored text to initially render / bind data from a SQL query
 - If enabled, the tailored text will retrieve data based on the SQL table rendered and the short field names.
- **Enable debug mode** – select to enable the debug mode; when the debug mode is enabled the module will render the exact query that is being processed.
- **Only enable when querystring value is present** - Enter the querystring parameter you would like to enable the initial SQL database.
 - **For example:** if you passed a querystring variable such as DBBind and a value of true, it would enable the SQL binding feature (if it was false, it would not enable the binding feature). This is useful if you want the binding feature to be disabled for initial submission; however maybe you want to enable the binding feature to update a record. This feature only works if you already have enabled the initial SQL data bind feature.
- **Only display module when 'TailoredSource' querystring value is present and match** - select if you want to hide the module unless the TailoredSource querystring object matches the value entered within this textbox.

- **For example;** if you entered 'Open' it would only display if you passed the queystring object of TailoredSource=Open.
- **Initial SQL Query** - select if you would like to display SQL Bind query being executed.
- **External Connection String (Optional)** - select an external connection string for this SQL Query; the external connection string will be used instead of the default DNN connection string.

After setting the desired parameters, click "Update" to save the changes.

7 LICENSING YOUR COPY OF TAILORED HTML

In order to license your copy of Tailored HTML module, choose “Tailored Text License” from the main menu.

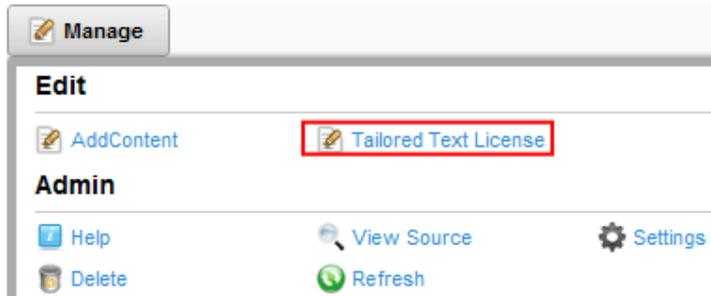


Figure 36: Licensing your copy of Tailored HTML module (step 1/2)

The following page will be displayed.

My Website > Module Training > Data Springs Product Licensing

DNN Version:	6.0.0
Product:	Tailored HTML
Product Variant:	Standard Edition
Product Version:	3.0.4232.31904
Machine Key:	E8EBFF68-39EA-46B0-950A-CFD11D58FD0B
Host Title:	DotNetNuke
Portals:	My Website (training.betasprings.com)
IP Address:	89.216.213.151
Contact / Developer Name:	<input type="text"/>
Contact / Developer Email:	<input type="text"/>
Customer Name:	<input type="text"/>
Invoice ID:	<input type="text"/>

BY CHECKING THE BOX INDICATING I AGREE TO THE TERMS AND

Register / Submit License
Exit

Figure 37: Licensing your copy of Tailored HTML module (step 2/2)

The following information should be supplied (**note**: all parameters are required):

- **Contact/Developer name and email address** – please enter the name and email of the module developer (**note**: important because in some situations we might need to contact the developer directly)
- **Customer Name** – enter your name
- **Invoice ID** – enter the Invoice ID you have received in your email form Snowcovered.com after the purchase

8 DELETING AN INSTANCE OF THE “TAILORED HTML” MODULE

In order to delete an instance of the “Tailored HTML” module, choose option “Delete” from the main menu.

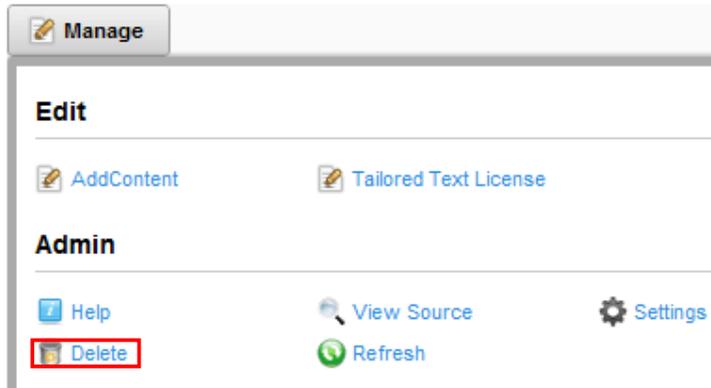


Figure 38: Deleting the module (step 1/2)

The following dialog window will be displayed.

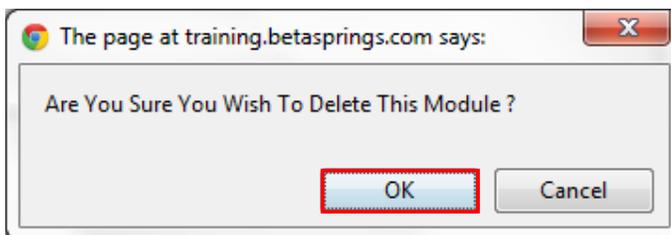


Figure 39: Deleting the module (step 2/2)

Click on the “OK” button and the module instance will be deleted from the page.