



Superior DotNetNuke Solutions.

PF Mega Drop-Down Menu & Navigation Documentation

Installation Instructions and Owners Documentation

June, 2011



General Instructions

Thank you for purchasing our PackFlash Mega Drop-Down Menu and Navigation Module, and make sure to check out all of *PackFlash's* other premium modules for your future projects. Our goal at *PackFlash* is to give webmasters the skins and modules they need to create truly professional presentation. Let us know if there is anything we can do better.

The following instructions will lead you, step by step, through the installation of the module pack, We ***strongly recommend*** that you back-up your website prior to doing any module installations. While we do rigorous testing on all of our products, we cannot account for every scenario, and, again, ***strongly recommend*** that you back-up your website prior to any module uploads to your DNN portal.

Table of Contents

| | |
|---|----|
| General Instructions | 2 |
| I. INSTALLATION INSTRUCTIONS for PackFlash Mega Drop-Down Menu and Navigation Module..... | 5 |
| 1. INSTALLATION INSTRUCTIONS for DNN 5.0+..... | 5 |
| II. Incorporating the Menu Into Your Skin | 9 |
| III. EXAMPLES OF USAGE OF THE PF NAVIGATION SKIN OBJECT WITHIN SKINS..... | 11 |
| IV. MANAGING MENU CONTENT AND COLUMN LAYOUT | 13 |
| 3-0. Setting up the Admin System | 13 |
| Where is the Mega Menu Admin Tool? | 13 |
| What if the Mega Menu Admin Tool Isn't Found? | 13 |
| First Time Use of the Tool | 13 |
| Important Note About the Tool..... | 14 |
| 3-1. Adding Images and Text Content..... | 14 |
| 3-2. About Columns | 16 |
| 3-3. Assigning Menu Items to a Column | 16 |
| 3-4. Adding New Pages (Menu Items) to a Column..... | 17 |
| 3-5. Creating New Columns..... | 17 |
| 3-6. Managing Content for Different Languages..... | 17 |
| V. REGISTERING THE PACKFLASH NAV IN YOUR SKIN..... | 19 |
| VI. REVIEW OF MENU SETTINGS..... | 19 |
| 1. Main Menu Settings..... | 19 |
| 2. Sub-Menu Settings | 20 |
| 3. Admin Menu Settings..... | 22 |
| VII. PUBLIC PROPERTIES OF THE PF NAVIGATION SKIN OBJECT..... | 24 |
| VIII. STYLING THE MENU | 29 |
| IX. EXPLANATION OF COLUMNS OPTIONS..... | 30 |
| Default Column Handling (without using AutoColumnsToGenerate) | 30 |
| Use of the AutoColumnsToGenerate Property | 33 |
| Non-"Complex" Formatting, Using DisplayFormat="MegaDropDown" | 35 |



PackFlash Mega Menu Skin Object and Admin Module Installation Instructions and Documentation

| | |
|--|----|
| X. Page Icon Handling | 35 |
| XI. UNINSTALLATION INSTRUCTIONS for PackFlash Mega Drop-Down Menu and Navigation Module..... | 37 |

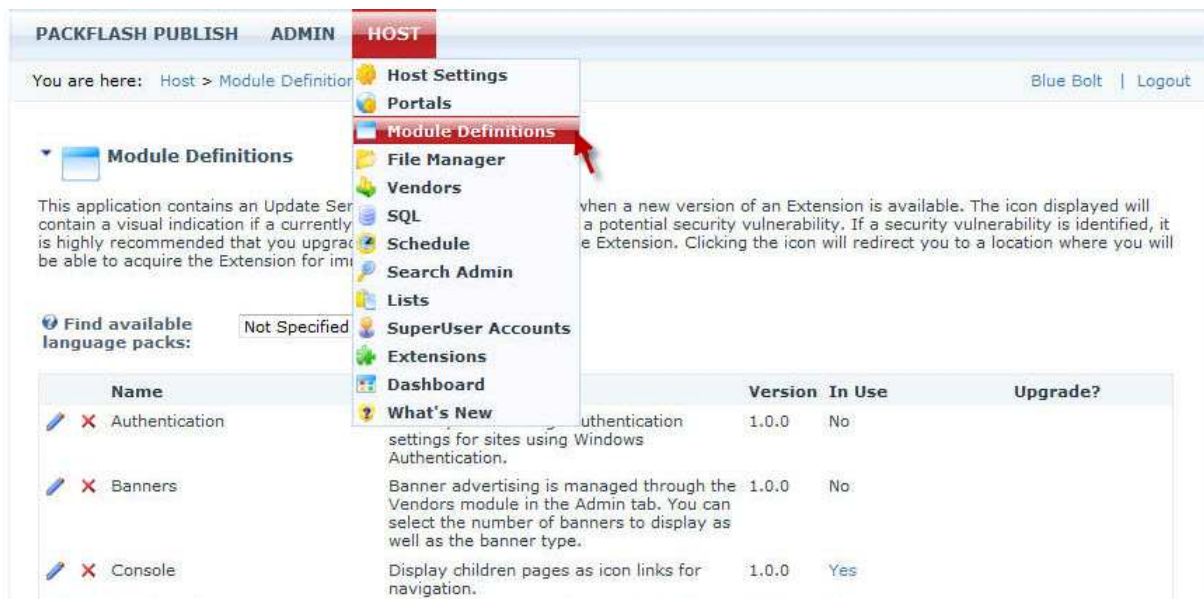
I. INSTALLATION INSTRUCTIONS for PackFlash Mega Drop-Down Menu and Navigation Module

Please follow these instructions.

1. INSTALLATION INSTRUCTIONS for DNN 5.0+

1-1. Log in as **Host** and select **Edit mode** from the menu.


1-2. Navigate to **Host -> Module Definitions**.



The screenshot shows the PackFlash Mega Menu Admin interface. The top navigation bar includes 'PACKFLASH PUBLISH', 'ADMIN', and 'HOST'. The 'HOST' dropdown menu is open, showing a list of menu items: Host Settings, Portals, Module Definitions (highlighted with a red arrow), File Manager, Vendors, SQL, Schedule, Search Admin, Lists, SuperUser Accounts, Extensions, Dashboard, and What's New. The 'Module Definitions' section is expanded, showing a list of modules with columns for Name, Version, In Use, and Upgrade?. The 'Authentication' module is highlighted.

















| Name | Version | In Use | Upgrade? |
|----------------|---------|--------|----------|
| Authentication | 1.0.0 | No | |
| Banners | 1.0.0 | No | |
| Console | 1.0.0 | Yes | |






1-3. Select **Install Module** from Module Definitions menu.

▼  **Module Definitions**

This application contains an Update Service which displays an icon when a new version of an Extension is available. The icon displayed will contain a visual indication if a currently installed Extension contains a potential security vulnerability. If a security vulnerability is identified, it is highly recommended that you upgrade to the newer version of the Extension. Clicking the icon will redirect you to a location where you will be able to acquire the Extension for immediate installation.

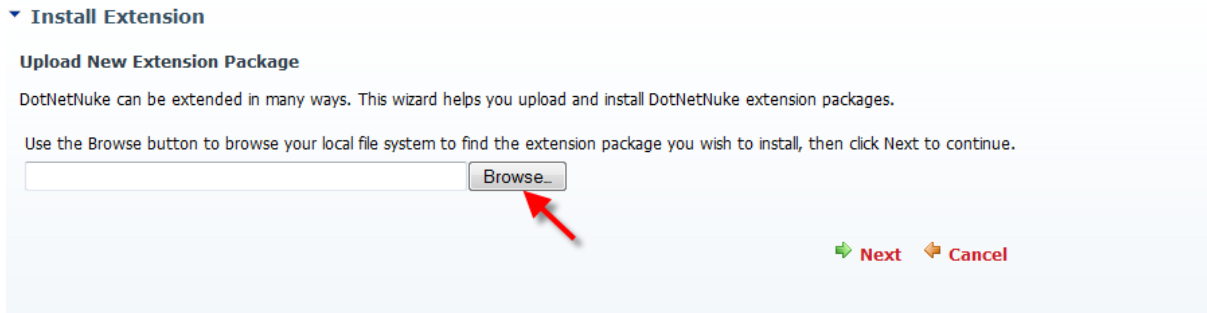
🔍 Find available language packs:

| Name | Description | Version | In Use | Upgrade? |
|---|---|---------|--------|----------|
|   Authentication | Allows you to manage authentication settings for sites using Windows Authentication. | 1.0.0 | No | |
|   Banners | Banner advertising is managed through the Vendors module in the Admin tab. You can select the number of banners to display as well as the banner type. | 1.0.0 | No | |
|   Console | Display children pages as icon links for navigation. | 1.0.0 | Yes | |
|  Dashboard | Provides a snapshot of your DotNetNuke Application. | 1.0.0 | Yes | |
|  Extensions | Allows a Super User to manage the various extensions, such as Skins, Modules Language Packs, providers etc. | 1.0.0 | Yes | |
|   Tabs | Administrators can manage the Tabs within the portal. This module allows you to create a new tab, modify an existing tab, delete tabs, change the tab order, and change the hierarchical tab level. | 1.0.0 | Yes | |
|   Users and Roles | Administrators can manage the security roles defined for their portal. The module allows you to add new security roles, modify existing security roles, delete security roles, and manage the users assigned to security roles. | 1.0.0 | Yes | |
|   Vendors | Administrators can manage the Vendors and Banners associated to the portal. This module allows you to add a new vendor, modify an existing vendor, and delete a vendor. | 1.0.0 | Yes | |
|   Whats New | Displays Whats New information for each release. | 1.0.0 | Yes | |

 **Install Module**
 **Create New Module**
 **Import Module Definition**



1-4. In **Install Extension** menu:

a. in **Upload New Extension Package** - Click **Browse** button to locate **PackFlashNavigation Module** and then click **Next**,



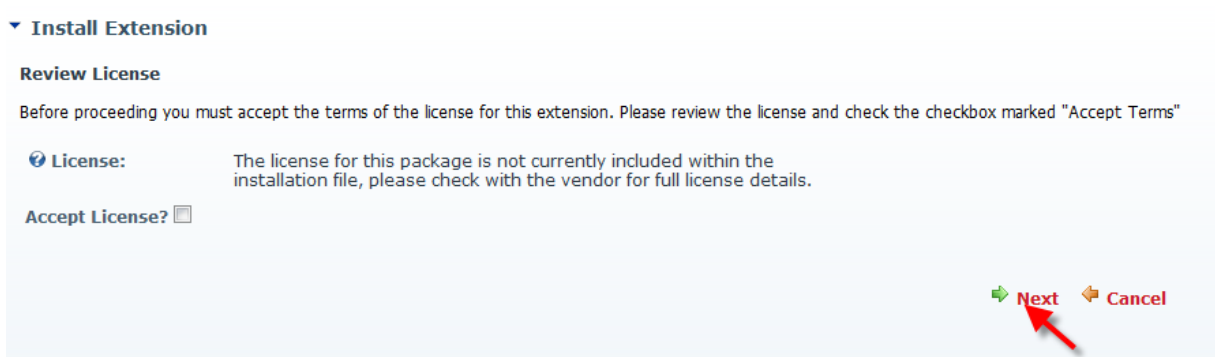
b. in **Package Information** - click **Next**,



d. in **Release Notes** - click **Next**,



e. in **Review License** - select **Accept License** box and click **Next**,



1-5. Click **Return link**.

II. Incorporating the Menu Into Your Skin

Here is a step by step process to incorporating the mega menu into an existing skin. As an example, we'll demonstrate how to convert the MinimalExtropy -1024 skin that comes with the DNN installation.

1. Install the sample Mega skins that are provided as part of the trial for the Mega Menu. Later, we'll need to copy code from this install, under the assumption that it is installed as a "host" skin, residing under the `<root>/portals/_default` folder
2. Duplicate and re-name your skin file (the .ascx file). In our example, we'll copy the file:
`<root>/portals/_default/Skins/MinimalExtropy/index 1024.ascx`
and re-name it to:
`<root>/portals/_default/Skins/MinimalExtropy/mega.ascx`
3. Open this new file (mega.ascx) in a text editor.
4. Locate the standard menu registration at the top of the page file. In our example, this registration is on line 5:

```
<%@ Register TagPrefix="dnn" TagName="NAV" Src="~/Admin/Skins/Nav.ascx" %>
```
5. Locate the instance of the standard menu within the skin file. Do this by searching for the combination of "TagPrefix" and "TagName" as found in the registration line, above. In this case, we are searching for "dnn:NAV" (note the colon). This finds the following instance at line 57:

```
<dnn:NAV runat="server" id="dnnNAV" ProviderName="DNNMenuNavigationProvider" IndicateChildren="false" ControlOrientation="Horizontal" CSSControl="mainMenu" />
```
6. Open the Sample1 file from the Mega skin samples in a text editor:
`/Portals/_default/Skins/SampleMegaSkins/Sample1_jQuery.ascx`
At line 17 of this file, you'll find the registration line for the Mega Menu. Copy this line:

```
<%@ Register TagPrefix="PF" TagName="NAV" Src="~/DesktopModules/PfDnnNavigation/SkinObject.ascx" %>
```
7. Returning to the new Mega.ascx file, replace the standard nav registration (line 5) with the registration for the Mega Menu skin object (copied in step 6).
8. Replace the menu instance of the standard nav (at line 57 of mega.ascx) with the code for Mega Menu instance, found at line 56 of Sample1_jQuery.ascx. (We have recently added comments in the sample files to indicate the "snip" points.)

```
<div class="TopNavigationMain">
<div class="HeaderSearch"></div>
<div class="TopNavContainer">
  <PF:NAV runat="server" ID="HorzNav"
  Theme="black_caliper_sample" HideJavascript="true"
  ThemeCssInSkinFolder="true" ExpandAll="True"
  UseOrdinalID="true" MenuClass="MenuWrapperMega"
  ContentType="SiteMapTabs" DisplayFormat="MegaDropDownComplex"
  AdminMenuDisplaySetting="SuppressAdminMenu" />
</div>
</div>
```

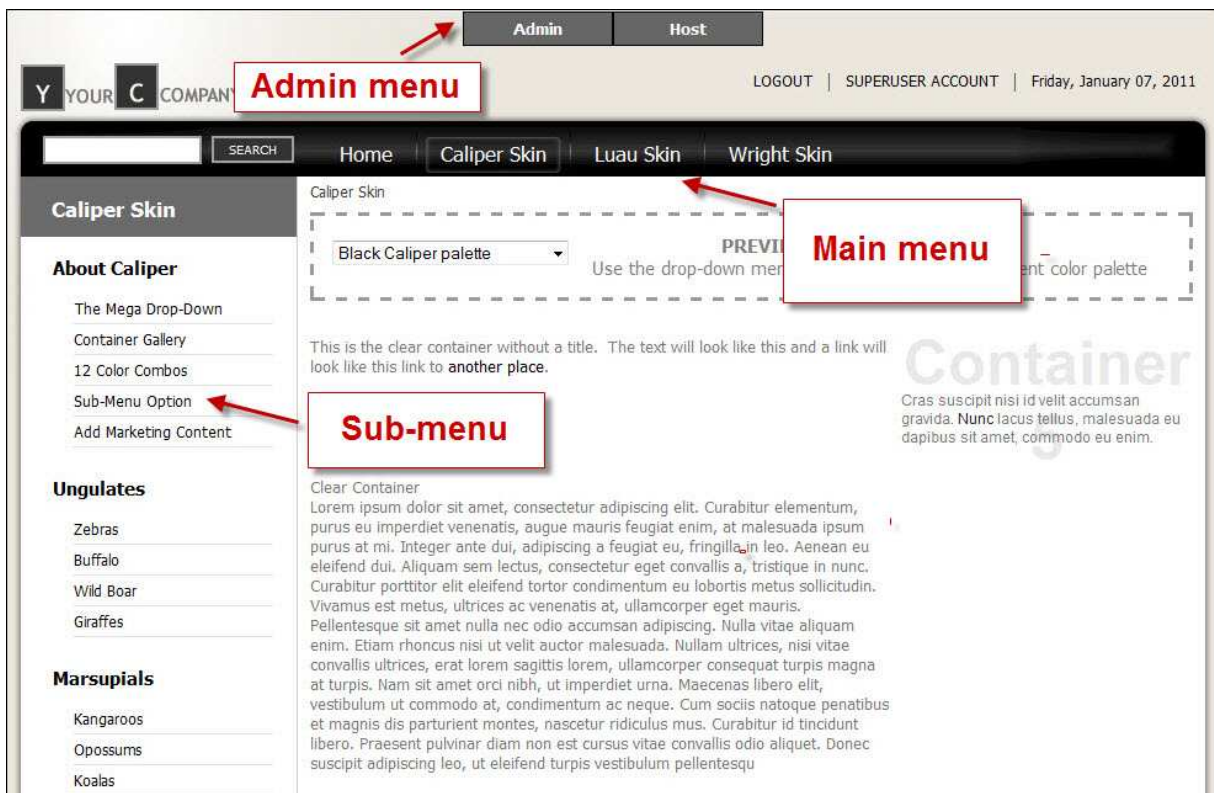
An explanation of all of the attributes for the menu is found in section 7 of this document. In this case, the “wrapper” divs are needed by the CSS for the theme used by Sample1. Note the Theme selection in red.

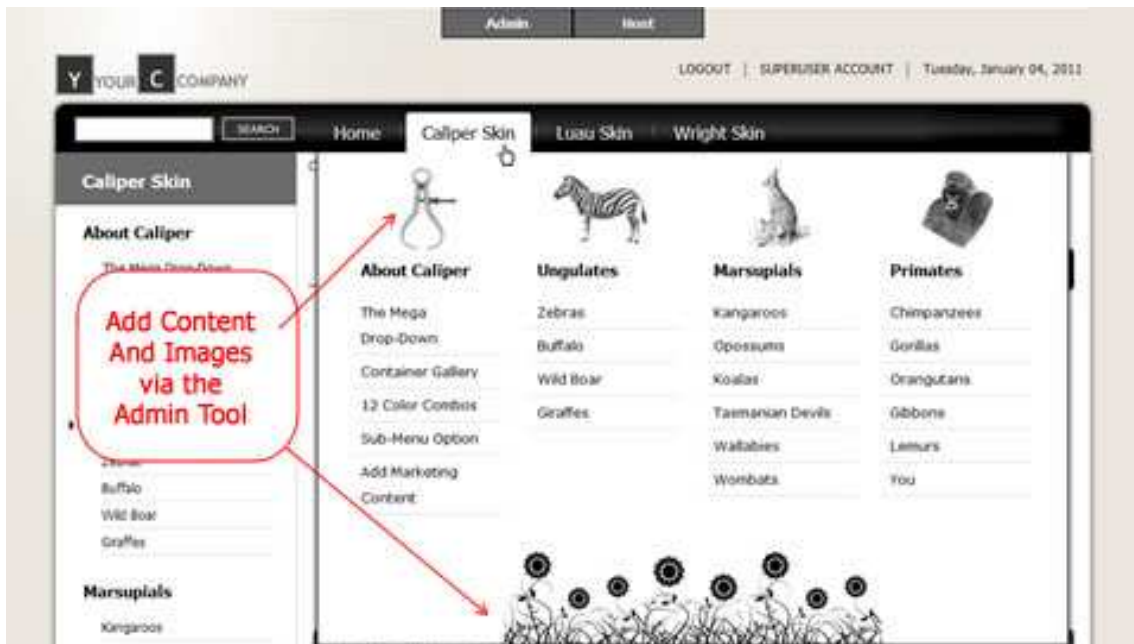
9. At this point the menu HTML will be built and if you were to assign the mega skin to a page, you would see all of the links for all of your pages built out as part of a bulleted list. To get them to properly display as mega menus, we need to copy the CSS theme code.
10. For our example, we’re going to apply the theme found in the Sample1 skin, which is called “black_caliper_sample”.
11. Create a sub-folder called themes under the MinimalExtropy skin folder. The full path will be:
`<root>/portals/_default/Skins/MinimalExtropy/Themes`
12. Find the following folder, under the sample skins and copy it to the Themes folder you just created under MinimalExtropy:
`/Portals/_default/Skins/SampleMegaSkins/Themes/black_caliper_sample`
Note that you want to copy all of the contents of this folder, including the CSS file and images.
13. Important new point: the Sample 1 theme will not work for quirks mode. This is why we include the file Sample1_jQuery.doctype.xml with the skin - it enforces the Transitional DocType (more info on the doctype xml at <http://www.dotnetnuke.com/Community/Wiki/tabid/1409/Page/Doctype/Default.aspx>). You can also manage this via the admin at Host Settings > Host Details > Fallback Skin Doctype
14. Apply the new Mega skin to a page and you should now see the the PackFlash Mega Drop-Down menu.

III. EXAMPLES OF USAGE OF THE PF NAVIGATION SKIN OBJECT WITHIN SKINS

For PackFlash skins, we sometimes include up to 3 instances of the navigation object within a single skin. In this scenario, the menus are use for:

1. The main menu: a horizontal menu living in the site mast-head (header). This main menu might be a Mega Drop-Down, or a “static” button bar.
2. The sub menu: a vertical menu along the left side of the site, showing children of the currently active top-level (root level) page.
3. The admin menu: living at the top of the skin, but only visible when an admin or host is logged in. This menu shows only the admin/host menu trees, while the main menu is set to suppress the admin/host menus.





The skin object will automatically build out the menus according to the page structure found in DotNetNuke. Additional images and content can be added through the admin tool found under the Admin menu (this page should have been created during the install process).

IV. MANAGING MENU CONTENT AND COLUMN LAYOUT

Note: page creation and ordering is handled the “normal” DNN way, by adding or editing a page, or through the Admin > Pages function. The Mega Menu Admin tool allows you to associate content to your menus, and to assign menu items to columns within a menu.

3-0. Setting up the Admin System

Where is the Mega Menu Admin Tool?

The installation process should create a DNN page and install the Mega Menu Admin tool on the page. The admin tool can be found under the DNN “Admin” menu: Admin > PackFlash Mega Menu

What if the Mega Menu Admin Tool Isn’t Found?

It is possible that the auto installation of the Mega Menu Admin could fail, or you might be using the tool in multiple portals within a single DNN installation. In either case, it is easy to set up the Mega Menu Admin manually:

1. Create a page upon which to drop the Mega Menu Admin module. You can name this page whatever you like and place it anywhere in your site. We normally name it “PackFlash Mega Menu” and set the “Admin” page as the parent.
2. Drop the module called PF MegaMenu Admin onto this new page. Ideally you will place the module into a pane that spans the full width of your content area.

First Time Use of the Tool

In order for the admin to properly display a preview of the menu while you are making edits you need to specify which skin should be used to generate the preview. The first time you go to the Mega Menu Admin module you will be prompted to select a skin. The skin you select needs to have the PackFlash Menu Skin Object installed in it. Generally speaking, you should select the skin that will be used as the portal default.

After you select the skin, click the “Get Skin Settings” button. Assuming this results in a green “success” message, then click the “Update” button to save your skin choice.

This process results in the creation of a copy of the skin which was selected. An example of how the new skin is named follows:

Original Skin (that was selected): My_Portal_Skin

Copied & Rename Skin: My_Portal_Skin_ADMIN_PREVIEW_ONLY

This “preview” version of the skin will be used by the admin tool so that changes can be made to the menu without affecting the “live” or “production” menu.

Important Note About the Tool

When you select a skin to be used for previewing the menu, the system will create a DUPLICATE skin file and make some changes to the this duplicate file so that it will work properly with the admin system. This duplicate file is created in the same skin folder where the original skin lives. The new file will have a suffix added to it so that it is clear that the skin shouldn't be used for public facing pages. For instance, if the original skin was called "My_Skin", the new file will be called "My_Skin_ADMIN_PREVIEW_ONLY".

3-1. Adding Images and Text Content

You can add HTML content to your mega menus. The HTML content must be associated with a menu item (a page on your site).

- Go to Admin menu -> "PackFlash Mega Menu Admin" -> "Manage Menu".
- Select a page where would you like add the content from a sub-menu on the left navigation.
- LEFT click on a page and select "Manage Content".
- Add the content (text/image/HTML) in the editor before or after preferred menu item.
- Choose "Content Alignment" from a drop-down menu (Above, Above and Centered, Above and Left Aligned, Above and Right Aligned, Left of the Menu Text, Right of the Menu Text or leave option "Apply no inline styles" like it is)
- Click "Save" at the end (found at the bottom of the page).

The following illustration shows how to manage content for a given menu item:

Menu is in PREVIEW mode Switch to LIVE mode

Select a Page
Left-Click to

left-click on a page to associate menu content with it

Content can go before or after the menu item

Home
Videos
Gadgets
Column 1
Phone
Thumb
Night Vision Camera
Rubik's Slide
Column 2
Column 3
Column 4
Mammals
Aux1
Aux2
Resources
Publish

Edit Column or Target
Manage Content

HTML Before the Menu Item, **Phone**

Add text, images, or any HTML (including CSS styles) to be included with the menu link. Exact placement of the HTML will depend on the CSS stylesheet for your skin, along with inline CSS styling that can be specified below the text editor.

Basic Text Box Rich Text Editor

Normal Tahoma, Arial... 15px

B I U abc

Apply CSS ...

Enter content here

Design HTML

Words: 4 Characters: 19

Content Alignment: if desired an inline CSS style can be applied to the content for alignment purposes

Above and Centered (display: block; text-align: center;)

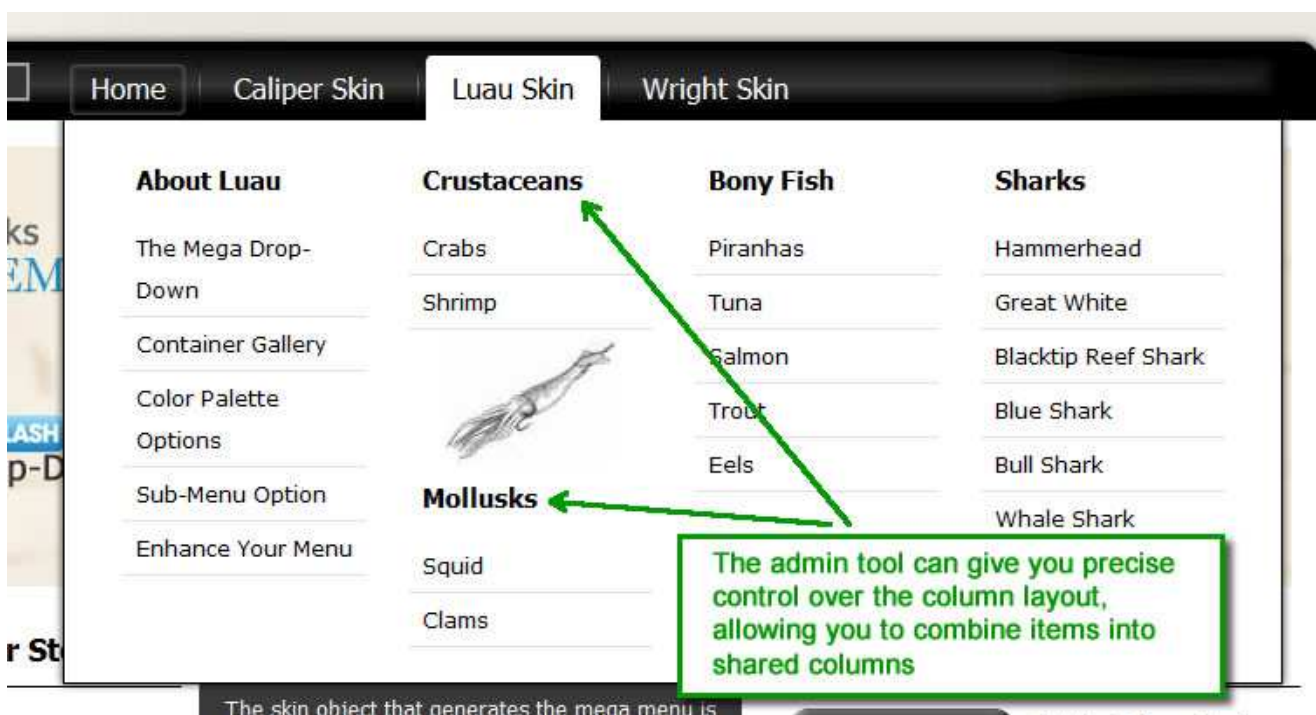
HTML After the Menu Item, **Phone**

Add text, images, or any HTML (including CSS styles) to be included with the menu link. Exact placement of the

3-2. About Columns

The PackFlash Mega Menu System provides multiple ways of controlling the columns of menu items within your menus. If the PackFlash menu skin object which is embedded into your skin is set to “auto columns” mode (example: AutoColumnsToGenerate=“4”) then columns will be automatically managed with no input from the administrator.

When the auto columns mode is not on, then the admin system can be used to manually manage the column assignments for all menu items. The next 3 sections discuss how to accomplish manual column assignments. For an in-depth explanation of how columns work, see section IX, near the end of this document.



3-3. Assigning Menu Items to a Column

If your skin is set up to manage columns manually (i.e. not in Auto Columns mode) you can manage the column assignments within the mega menu admin tool.

- Go to Admin menu -> "PackFlash Mega Menu Admin" -> "Manage Menu".
- Using the tree navigation in the left side of the admin page, find the menu item which you would like to move to a different column.
- LEFT-click on the menu item, which will open a context menu.
- Select "Edit Column or Target" from a menu.

- Use "Column" drop-down menu to change the existing column to preferred column.
- Click "Save".
- Click "Menu Preview" to review it.
- If you are ready to apply these changes and see them on the live site, click "Publish". Otherwise save your changes as a New File and then you can re-upload the updated menu when you are ready to publish.

3-4. Adding New Pages (Menu Items) to a Column

- Create the new sub-pages in Admin and make sure to choose the right location (tabs location).
- Modify their location in preferred column (see next instructions above).

3-5. Creating New Columns

- On top level (tabs level in top navigation):

Make sure you have the pages created in Admin in order to see the columns in PF Navigation Mega Menu.

- On lower level (Column 1 etc):

LEFT click in a sub-menu in left navigation and choose "Add Column".

NOTE: you can also delete this column from the pop-up menu.

3-6. Managing Content for Different Languages

The mega menu and the admin tool are multi-lingual capable. When using the admin tool, simply use the language selector within your skin to select that language that you'd like to work with. The admin tool will automatically display the pages and content for the selected language.

If you try to switch languages "in the middle" of making content updates, without first saving or publishing the updates, you will receive a warning message.

When saving menus in a multi-lingual environment be aware that menus for all languages are saved in the same folder, and that you can potentially overwrite a menu for another language if you give it the same name. Therefore it is best to include a reference to the language in the menu name.



PackFlash Mega Menu Skin Object and Admin Module Installation Instructions and Documentation

Option #2:

- Go to Admin menu -> "PackFlash Mega Menu Admin" -> "Manage Menu".
- Select a column you would like to move to a different location.
- LEFT click on the Column you would like to move somewhere else.
- Choose "Move Up" or "Move Down" from a sub-menu in a left navigation.
- Click "Menu Preview" to review it.
- If you are ready to apply these changes and see them on the live site, click "Publish". Otherwise save your changes as a New File and then you can re-upload the updated menu when you are ready to publish.

V. REGISTERING THE PACKFLASH NAV IN YOUR SKIN

Include the following registration code near the top of your skin, beside the “normal” DotNetNuke object registrations:

```
<%@ Register TagPrefix="PF" TagName="NAV"  
Src="~/DesktopModules/PfDnnNavigation/SkinObject.ascx" %>
```

VI. REVIEW OF MENU SETTINGS

1. Main Menu Settings

```
<PF:NAV runat="server"  
  ID="HorzNav"  
  ContentType="SiteMapTabs"  
  DisplayFormat="MegaDropDown"  
  MaxMenuLevel="2"  
  Theme="black_caliper"  
  ThemeCssInSkinFolder="true"  
  MenuClass="MenuWrapperMega"  
  AdminMenuDisplaySetting="SuppressAdminMenu"  
  AutoColumnsToGenerate="4" />
```

An explanation of the settings:

- **ID="HorzNav"** – as usual, an ID must be assigned to the skin object instance. It can be any name you like, but it must be unique within the skin file.
- **ContentType="SiteMapTabs"** – Insures that the entire site map will be returned. Alternatively, “MegaDropDown” could have been specified for the same result.
- **DisplayFormat="MegaDropDown"** – results in HTML output per the PackFlash MegaDropDown specification.

- **MaxMenuLevel="2"** – limits the menu to 3 levels of hierarchy: the root level tabs, plus child and grandchild pages.
- **Theme="black_caliper"** – this is the name of the folder where the theme.css file lives for menu formatting.
- **ThemeCssInSkinFolder="true"** – allows the theme CSS to be bundled and installed with the skins. The path to the theme.css stylesheet will be: <SiteRoot>/Portals/<Site-Root-OR-_default>/SkinFolderName/Themes/ThemeName/theme.css
- **MenuClass="MenuWrapperMega"** – the DIV that contains the menu will be assigned to this class.
- **AdminMenuDisplaySetting="SuppressAdminMenu"** – the admin and host menus will NOT be included in the main navigation so that more space can be allocated to the actual user-facing menu and avoiding spatial issues if the admin/host menus cause the menu width to exceed the skin width.
- **AutoColumnsToGenerate="4"** – the menu nodes will be automatically configured into 4 columns. Note that having a value of greater than 0 for this property means that it will take precedence over column assignments made with the admin tool. For a fuller explanation of this setting, see the section on Column Options

2. Sub-Menu Settings

```
<PF:NAV runat="server"  
  ID="LeftNav"  
  DisplayFormat="HTMLul"  
  ContentType="AutoChildTabs"  
  Theme="black_caliper"  
  ThemeCssInSkinFolder="true"  
  ExpandAll="True"  
  MenuClass="LeftMenuWrapper"  
  NoTheme="true" />
```

An Explanation of the Settings:

- **ID="LeftNav"** – as usual, an ID must be assigned to the skin object instance. It can be any name you like, but it must be unique within the skin file.
- **DisplayFormat="HTMLul"** - the generated menu will be in the form of a nested UL list

- **ContentType**="AutoChildTabs" – the object will determine which of the root level tabs is active the menu will consist of children of this tab
- **Theme**="black_caliper" – the CSS them to be used is black_caliper. This setting is not really needed because the "NoTheme" property is set to "true".
- **ThemeCssInSkinFolder**="true" – indicates that the theme file will live under the skin directory. As above, this setting is not used because "NoTheme" is set to true. We leave the setting in place, however, for developer's reference.
- **ExpandAll**="True" - this setting will cause the entire sub-menu hierarchy to be visible, including pages that are not part of the currently active path. Depending on your size of your menu and personal preference you may wish to set this to false.
- **MenuClass**="LeftMenuWrapper" - the DIV that contains the menu will be assigned to this class.
- **NoTheme**="true" – because we include the CSS for all the menus on a given skin in a single theme stylesheet (black_caliper in this example) there is no need to include multiple references to the theme file. Setting NoTheme="true" eliminates the redundant reference to the black_caliper stylesheet from the Head section of the HTML.

3. Admin Menu Settings

```
<PF:NAV runat="server"  
  ID="AdminNav"  
  AdminMenuDisplaySetting="ShowOnlyAdminMenu"  
  ContentType="SiteMapTabs"  
  DisplayFormat="MegaDropDown"  
  Theme="black_caliper"  
  ThemeCssInSkinFolder="true"  
  MenuClass="AdminMenu"  
  NoTheme="true"  
>
```

An Explanation of the Settings:

- **ID="AdminNav"** – as usual, an ID must be assigned to the skin object instance. It can be any name you like, but it must be unique within the skin file.
- **AdminMenuDisplaySetting="ShowOnlyAdminMenu"** – this setting, along with **ContentType="SiteMap"** results in a menu that includes only the admin and host menu trees according to the users access privileges
- **DisplayFormat="MegaDropDown"** - the generated menu will have markup to display the drop-down menus
- **ContentType="SiteMapTabs"** – the object will determine which of the root level tabs is active the menu will consist of children of this tab
- **Theme="black_caliper"** – the CSS them to be used is black_caliper. This setting is not really needed because the “NoTheme” property is set to “true”.
- **ThemeCssInSkinFolder="true"** – indicates that the theme file will live under the skin directory. As above, this setting is not used because “NoTheme” is set to true. We leave the setting in place, however, for developer’s reference.
- **MenuClass="AdminMenu"** - the DIV that contains the menu will be assigned to this class.
- **NoTheme="true"** – because we include the CSS for all the menus on a given skin in a single theme stylesheet (black_caliper in this example) there is no need to include multiple references to the theme file. Setting **NoTheme="true"** eliminates the redundant reference to the black_caliper stylesheet from the Head section of the HTML.



PackFlash Mega Menu Skin Object and Admin Module Installation Instructions and Documentation

VII. PUBLIC PROPERTIES OF THE PF NAVIGATION SKIN OBJECT

A note about terminology in this documentation: the DotNetNuke convention of “tab” is used to refer to a page within a DotNetNuke website.

DisplayFormat

Values must match the enum, DisplayFormat. Allowed values include:

- HtmlUL – results in a simple UL list. Depending on CSS styling this can result in either a horizontal or vertical menu.
- MegaDropDown – produces markup required for a drop-down navigation, basically a UL list with extra divs incorporated to represent the drop-downs and columns.
- HtmlTable – produces an HTML table with extra table cells on either side of the menu item. Due to more modern methods, this approach is not recommended. Set the HideJavascript property to false if this setting is used.
- SeparatedList – produces a simple list of menu items separated by HTML that you specify. Usually used in conjunction with RootTabID or IncludedTabs property. See also: SeparatorHTML property.

Notes: the enum values “Columnar”, “Tree”, “HorizontalWithSub” are not currently supported. The value of “Test” is used for internal PackFlash development only. None of these should be used at this time.

ContentType

Values must match the enum, ContentType. Allowed values include:

- AutoChildTabs – produces an automated sub-menu by determining the currently active root level tab and finding its children. Most often used in conjunction with DisplayFormat of HtmlUL to create a vertical left menu.
- SiteRootTabs – includes *only* the root level pages of the site. Most useful for creating a top/main navigation when drilling into this site menu is not needed or desired. When combined with a 2nd instance of the menu, set to AutoChildTabs, results in a combination that produces the conventional “C-clamp” navigation.

- ChildrenOfTab – works similar to the AuthChildTabs option, but rather than automatically determining the root of the menu, the value of the RootTabID property will be used.
- SiteMapTabs & MegaDropDown – both of these values produce the same result: a menu including every visible page in the system will be produced. It is recommended that the MaxMenuLevel property be used to limit the menu depth if the intended DisplayFormat is MegaDropDown.
- SpecifiedTabs – produces exactly the list of links as specified in the IncludedTabs property.

AdminMenuDisplaySetting

Values must match the enum, AdminMenuDisplay. Allowed values include: IncludeAdminMenu, SuppressAdminMenu, ShowOnlyAdminMenu

HideJavascript

This is a Boolean property, expecting a value of “true” or “false”. Default value is false. This value should be set to true if the DisplayFormat is set to HtmlTable.

NoTheme

This is a Boolean property, expecting a value of “true” or “false”. The default value is “false.”

This setting is useful in two scenarios: if you are including multiple menu objects in the same skin and both use the same theme, then you can set NoTheme to “true” for either object to eliminate a redundant reference to the theme.css file. Or, if you wish to consolidate your CSS into a non-theme file, such as skin.css, then this setting will eliminate the un-needed stylesheet reference.

IncludedTabs

Expected value is a comma delimited list of tab IDs. Example: 34,78,102,87,176. IDs should be in the desired display order.

Inclusions [deprecated: use IncludedTabs instead]

ExcludedTabs

Expected value is a comma delimited list of tab IDs that should NOT be included in the menu

Exclusions [deprecated: use ExcludedTabs instead]

Theme

The theme folder which is expected to contain the file "theme.css" unless the ThemeFile property is changed from the default. The default location for the theme folder is at the following path: <SiteRoot>/DesktopModules/PfDnnNavigation/Themes/<ThemeSettingValue>. Alternatively, the theme can live under the skin folder path by setting the ThemeCssInSkinFolder property to “true”.

ThemeCssInSkinFolder

This is a Boolean property, expecting a value of “true” or “false”.

Default value is False. Setting to True would indicate that the theme.css file is to be found under the skin folder. Example path (automatically constructed based on the skin path):

/Portals/0/skins/[SkinName]/Themes/[ThemeName]/theme.css

See also: Theme property

OpenExternalLinkInNewWindow

This is a Boolean property, requiring a value of “true” or “false”.

If this flag is TRUE then the skin object will check to see if the URL matches one of the "internal" domains, and if not then open the link in the new window. Applies when the admin user sets the page “Link URL” option to “URL (A Link to an External Resource).”

See also: InternalDomains property

InternalDomains

This should be a comma-delimited list of domains which are considered "internal." Examples:

"www.mydomain.com,mydomain.com,dev.mydomain.com." When used with the

“OpenExternalLinkInNewWindow” property, this can force menu links which point to 3rd-party (external) websites to open in a new browser window.

See also: OpenExternalLinkInNewWindow property

ExpandAll

Indicates whether all menu items should be expanded, regardless of active state. This property is only referenced when DisplayFormat is set to HtmlUL. The most common use of this is to create a left/sub nav where the entire page hierarchy is visible.

See also: DisplayFormat property

MenuClass

The MenuClass value is applied to the containing DIV tag for the menu. If this property is not set then the following defaults will apply, depending on the DisplayFormat setting.

See also: DisplayFormat property

RootTabID

The ID of the root tab to use when generating the menu. This property only applies when ContentType is set to "ChildrenOfTab". This is most typically used with the "HtmlUL" DisplayFormat, in which case the specified root tab name and link will be output to a DIV tab just before the UL list with a special class designation. If it is desired to NOT display the root tab, use CSS to set the display property to "none."

See also: ContentType property, DisplayFormat property

MaxMenuLevel

A numeric value is required. If this property is set then the menu will not include pages that exceed this level setting. In keeping with DotNetNuke convention, root level tabs have a level of 0, their children will have a level of 1, and so on.

Comments: when using DisplayFormat of "MegaDropDown" it is good practice to set the MaxMenuLevel property to 2, which limits the menu to 3 levels of hierarchy. This is typical of how such menus are usually set up, although sometimes one will see a menu which includes 4 levels of hierarchy.

SeparatorHtml

This property is valid when used with DisplayFormat is set to "SeparatedList". The default value for this property is " | ", which produces a pipe separator with spaces on either side.

See also: DisplayFormat property

HideCopyright

This is a Boolean property, requiring a value of "true" or "false". If set to "false" the PackFlash copyright notice will not be output as a comment above the menu markup. Default setting is "true." Of course, this comment is not visible to anyone viewing the webpage.

UseOrdinalID

This is a Boolean property, requiring a value of "true" or "false". The default value is "false." When set to true, menu items will be output to the markup with the TabIDs as their ID within the markup. If it is set to false, then a unique ID will be generated by system, according to the menu items position within the hierarchy. This gives skin developers the flexibility to create special rules for singular menu items which can be referenced by their DotNetNuke Tab ID, or according their hierarchical position.

Examples of how this option affects the markup when DisplayFormat is set to "MegaDropDown":

UseOrdinalID = true

```
<li class="Inactive" id="Link584"><a href="/mypage.aspx">My Page</a>
</li>
```

UseOrdinalID = false

```
<li class="Inactive" id="Menu1_L1_9"><a href="/mypage.aspx">My Page</a>
</li>
```

UseSettingsForAdminPreview

This is a Boolean property, requiring a value of “true” or “false”. If set to “true” this designates that the settings from this instance of the skin object will be used to drive the preview within the administrative tool for the menu. This is useful in the case where you have multiple menus on one skin. For example, PackFlash skins sometimes have 3 menu instances: a main menu, a sub/left menu, and an admin menu.

AutoColumnsToGenerate

A numeric value is required. If the value is greater than 0, then a set number of columns will automatically be generated within the drop menus. See the section, “Explanation of Columns” for more information on the effect of this setting. There is also more information in the forums. If this property is not set, the default value is -1.

SuppressIcons

This is a Boolean property, requiring a value of “true” or “false”. If this property is not set, the default value is false. If you wish to NOT display page icons within your menu, set this property to “true”.

IconAfterMenuText

This is a Boolean property, requiring a value of “true” or “false”. If this property is not set, the default value is false. If you wish for menu icons to be displayed after the menu text (to the right, depending on your CSS rules) then set this property to “true”.

AuxAdminTabs

Expects a comma-delimited list of DNN tab IDs (page IDs). This property allows you to specify other pages which are to be included with the Admin and Host menus.

See also: AdminMenuDisplaySetting

ShowHiddenPagesToAdmins

This is a Boolean property, requiring a value of “true” or “false”. If this property is set to true then “hidden” pages will be included in the menu when an administrator is logged in. A page is set to be “hidden” if the DNN option to “Include in Menu” is not checked.

VIII. STYLING THE MENU

You can control the menu appearance through CSS. These CSS rules can exist in any stylesheet referenced by your skin, including the skin.css.

The styling can be made “portable” by creating a theme. To create a theme:

- Create a “Themes” within the folder for the skin that you are using
- Create a sub-folder under “Themes” which will act as the name of your theme. For instance. If you are using a skin called, “MyCustomSkin,” (assumed to be in the _default folder for now) and want to create a theme called “MyCustomTheme,” you’ll create the following path:
`<root>/Portals/_default/skins/MyCustomSkin/Themes/MyCustomTheme`
- Create a file called Theme.css within this folder
- Put your styling rules for the menu in this folder
- Set your skin object to refer to this theme:
 - ThemeExistsInSkinFolder=”true”
 - Theme=”MyCustomTheme”

IX. EXPLANATION OF COLUMNS OPTIONS

With the release of version 1.2 of the mega menu skin object, there is a new property which gives new functionality related to managing columns. The new property is called `AutoColumnsToGenerate`. This property expects a numeric value to be passed which stipulates the number of columns that will be built into the menus. If this property is not explicitly set, or has a value of less than 1, then the default column handling will apply.

Default Column Handling (without using `AutoColumnsToGenerate`)

In order to get "true" columns within the mega menus, the `DisplayFormat` needs to be set to "MegaDropDownComplex." (See below for an explanation of how the "regular" option of "MegaDropDown" is rendered.) If the `AutoColumnsToGenerate` option is NOT set then the menu will be built so that every Level 1 menu item gets its own column div. See the illustration below:

Default Column Handling for DisplayFormat="MegaDropDownComplex"

Level-0-Item

Level-0-Item

Level-0-Item

```
<div class="shadow">
```

```
<div><ul>
```

```
<div class="Column">
```

```
<ul>
```

```
<li>Lvl 1 Item a</li>
```

```
<ul>
```

```
</div>
```

```
<div class="Column">
```

```
<ul>
```

```
<li>Lvl 1 Item b</li>
```

```
<ul>
```

```
</div>
```

```
<div class="Column">
```

```
<ul>
```

```
<li>Lvl 1 Item c</li>
```

```
<ul>
```

```
</div>
```

```
<div class="Column">
```

```
<ul>
```

```
<li>Lvl 1 Item Longer d</li>
```

```
<ul>
```

```
</div>
```

```
<div class="Column">
```

```
<ul>
```

```
<li>Lvl 1 Item Longer e</li>
```

```
<ul>
```

```
</div>
```

```
<div class="Column">
```

```
<ul>
```

```
<li>Lvl 1 Item Long f</li>
```

```
<ul>
```

```
</div>
```

```
<div class="Column">
```

```
<ul>
```

```
<li>Lvl 1 Item g</li>
```

```
<ul>
```

```
</div>
```

```
<div class="Column">
```

```
<ul>
```

```
<li>Lvl 1 Item h</li>
```

```
<ul>
```

```
</div>
```

```
</ul></div>
```

```
</div>
```

Each Level 1 item is placed in its own div (class="Column"). How the Column div behaves is up to the CSS. In our Sample1 skin, the Divs float and are centered. This can cause wrapping and alignment issues if there a lots of Level 1 items (as above).

If there were any level 2 items, they would exist as UL/LI stuctures nested in the parent Level 1 LI.

The advantage to this "default" approach is that you can use the admin tool to combine Level 1 items into columns exactly as you wish. As you combine Level 1 items, they will be nested as LI's into a common UL element. Thus, when used with the admin tool, this approach is multimately more flexible than the automated column approach.

The skin object code to produce the above menu looks like this:

```
<PF:NAV runat="server" ID="HorzNav" Theme="black_caliper_sample" HideJavascript="true"
ThemeCssInSkinFolder="true" ExpandAll="True" UseOrdinalID="true"
MenuClass="MenuWrapperMega" ContentType="SiteMapTabs" DisplayFormat="MegaDropDownComplex"
AdminMenuDisplaySetting="SuppressAdminMenu" />
```

This screen shot illustrates how such a menu might look as it actually renders. Note that this example illustrates a menu full of only Level 1 items:



You can see in the above illustration how the 4th row is not aligned with the rows above. This is because each menu item is in its own free-floating "Column" div. The CSS spacing applied to the Column div also adds a lot of unnecessary space. Using the very same settings (within the skin object, and within the CSS) can produce a different result if the page count and hierarchy is different. There is a screen shot, below, of the same menu with AutoColumnsToGenerate set to 4.

Use of the AutoColumnsToGenerate Property

With this property, you can set the menu to automatically create a set number of columns, combining Level 1 items in sequence.

AutoColumnsToGenerate="2"

Level-0-Item
Level-0-Item
Level-0-Item

```

<div class="shadow">
  <div><ul>
    <div class="Column">
      <ul>
        <li>Lvl 1 Item a</li>
        <li>Lvl 1 Item c</li>
        <li>Lvl 1 Item e</li>
        <li>Lvl 1 Item g</li>
      </ul>
    </div>
    <div class="Column">
      <ul>
        <li>Lvl 1 Item b</li>
        <li>Lvl 1 Item d</li>
        <li>Lvl 1 Item f</li>
        <li>Lvl 1 Item h</li>
      </ul>
    </div>
  </ul></div>
</div>

```

Exactly 2 "Column" divs are created and the Level 1 items are divided between them. The behavior of the columns is dependent upon your CSS rules, but this is roughly how they would display in our Sample 1B skin.

Any level 2 items would be nested as UL/LI elements within the parent LI.

The skin object code to produce the above menu looks like this:

```

<PF:NAV runat="server" ID="HorzNav" Theme="black_caliper_sample"
HideJavascript="true" ThemeCssInSkinFolder="true" ExpandAll="True"
UseOrdinalID="true" MenuClass="MenuWrapperMega" ContentType="SiteMapTabs"
DisplayFormat="MegaDropDownComplex" AutoColumnsToGenerate="2"
AdminMenuDisplaySetting="SuppressAdminMenu" />

```

This screen shot illustrates how our Sample 1 skin renders with AutoColumnsToGenerate set to "4":



Non-"Complex" Formatting, Using DisplayFormat="MegaDropDown"

The MegaDropDown DisplayFormat option is useful but has a key limitation: you cannot combine multiple Level 1 items into the same column. In our skins, PackFlash typically uses this simple DisplayFormat to create the Admin/Host menus as the top of the skin (most of our samples demonstrate this). Here is a graphic illustration of how the HTML is constructed in this scenario:

HTML Output for DisplayFormat="MegaDropDown"

Level-0-Item
Level-0-Item
Level-0-Item

```

<div class="shadow">
  <div>
    <ul>
      <li>Level 1 Item A
        <ul>
          <li>Lvl 2 Item</li>
          <li>Lvl 2 Item</li>
          <li>Lvl 2 Item</li>
          <li>Lvl 2 Item</li>
        </ul>
      </li>
      <li>Level 1 Item B
        <ul>
          <li>Lvl 2 Item</li>
          <li>Lvl 2 Item</li>
          <li>Lvl 2 Item</li>
          <li>Lvl 2 Item</li>
        </ul>
      </li>
      <li>Level 1 Item C
        <ul>
          <li>Lvl 2 Item</li>
          <li>Lvl 2 Item</li>
          <li>Lvl 2 Item</li>
          <li>Lvl 2 Item</li>
        </ul>
      </li>
    </ul>
  </div>
</div>

```

Limitation: there are no "Column" divs, which limits display formatting and Level 1 items cannot be combined into shared columns. For instance, you could not write CSS such that Level 1 Item A and Level 1 Item B shared a column while Level 1 Item C was in its own column.

X. Page Icon Handling

With the Mega Menu version 1.2, by default, any icons assigned to pages through the standard DNN icon functionality will be shown in the menu. In previous versions, the icons were only shown for admin/host menus, and the assignment of icons for other menu nodes needed to be handled through the HTML Before feature. It is not possible to have an icon, and HTML Before and HTML After all active for any menu node.

By default, the icon IMG tag will be included just before the menu text, within the menu node's A tag, as follows:

```
<li class="Inactive" id="Menu2_Col3_L1_3">  
<a href="menuItemURL/default.aspx">  
  
Menu Item Link Text  
</a>  
</li>
```

(in this example, line breaks are added for readability)

Note the class `IconImgBefore`, which is automatically applied to the `IMG` tag. This class can be used to add alignment rules to the icon. For the admin menu in our sample skins the following rule is applied:

```
.AdminMenu ul#PfMegaAdmin .IconImgBefore  
{ margin-right: 3px; display:block; float: left; }
```

This rule could be applied within the `theme.css` file or within the `skin.css` file. To apply such a rule to the “main” menu, instead of to the admin menu, change the class reference, as follows:

```
ul#PfNavMega .IconImgBefore  
{ margin-right: 3px; display:block; float: left; }
```

To turn icons off:

Use the “`SuppressIcons`” property within the skin object. Here is an example of the usage:

```
<PF:NAV runat="server" ID="HorzNav" Theme="black_caliper_sample"  
HideJavascript="true" ThemeCssInSkinFolder="true" ExpandAll="True"  
UseOrdinalID="true" MenuClass="MenuWrapperMega" ContentType="SiteMapTabs"  
DisplayFormat="MegaDropDownComplex" AutoColumnsToGenerate="4" SuppressIcons="false"  
AdminMenuDisplaySetting="SuppressAdminMenu" />
```

To have the icons display to the right of the menu text:

Use the `IconAfterMenuText` property within the skin object. Here is an example of the usage:

```
<PF:NAV runat="server" ID="HorzNav" Theme="black_caliper_sample"  
HideJavascript="true" ThemeCssInSkinFolder="true" ExpandAll="True"  
UseOrdinalID="true" MenuClass="MenuWrapperMega" ContentType="SiteMapTabs"  
DisplayFormat="MegaDropDownComplex" AutoColumnsToGenerate="4"  
IconAfterMenuText="true" AdminMenuDisplaySetting="SuppressAdminMenu" />
```

XI. UNINSTALLATION INSTRUCTIONS for PackFlash Mega Drop-Down Menu and Navigation Module

Please follow these instructions.

7-1. Log in as the **Host** and select **Edit mode** from the menu.

7-2. Navigate to **Host ->Extensions**.

The screenshot shows the PackFlash Mega Admin interface. The top navigation bar includes 'HOME', 'PACKFLASH MAPPING', 'PUBLISH', 'ADMIN', and 'HOST'. The 'HOST' menu is open, showing options like 'Host Settings', 'Portals', 'Module Definitions', 'File Manager', 'Vendors', 'SQL', 'Schedule', 'Languages', 'Search Admin', 'Lists', 'SuperUser Accounts', 'Skins', 'Extensions', and 'Dashboard'. The 'Extensions' option is highlighted with a red arrow. The main content area shows the 'Extensions' page with a table of installed extensions.

| Name | Type | Version | Upgrade? |
|---------------------------------------|------------------|---------|----------|
| DotNetNuke | Framework | 5.0.1 | |
| Default Authentication | Auth_System | 1.0.0 | |
| X DNN_LiveIDAuthentication | Auth_System | 1.0.1 | |
| X DNN_OpenIDAuthentication | Auth_System | 2.0.0 | |
| Default Language | CoreLanguagePack | 1.0.0 | |
| X Authentication | Module | 1.0.0 | |
| X Banners | Module | 1.0.0 | |
| Dashboard | Module | 1.0.0 | |
| Extensions | Module | 1.0.0 | |
| X Feed Explorer | Module | 1.0.0 | |










7-3. Click on the icon next to **PackFlashNavigation Module** all components individually and make the uninstallation for each module separately.

Module Definitions

Installed Modules

This application contains an Update Service which displays an icon when a new version of a Module or Language Pack is available. The icon displayed will contain a visual indication if a currently installed Module contains a potential security vulnerability. If a security vulnerability is identified, it is highly recommended that you upgrade to the newer version of the Module. Clicking the icon will redirect you to a location where you will be able to acquire the Module or Language Pack for immediate installation.

Locales: <Not Specified>

| Module Name | Description | Version | Upgrade Available? |
|---|---|----------|--------------------|
| DotNetNuke | DotNetNuke Community Edition | 04.09.05 | |
| [Skin Objects] | Skin Objects are User Controls which can be used to provide custom functionality to your Skin files. | | |
|  PackFlash_VideoDashboard | Video description | 01.00.00 | |
|  PackFlash_VideoDetails | When an article is created an "Article Detail" page is created automatically and a publish detail module is automatically put on that page. A detail module would also be put on any page that is set to display article details. You will almost never need to put this module on a page manually. | 01.00.00 | |
|  PackFlash_VideoDisplayIssue | The issue list module displays can display an issue in a TOC format and has the ability to display previous issues. It also has a grid view to display previous issues in. | 01.00.00 | |
|  PackFlash_VideoIssueTeaser | The issue teaser module displays the current issue and should be used in conjunction with the publish issue list module. | 01.00.00 | |
|  PackFlash_VideoList | Publish list can be used to display various list of articles on your site using predefined templates | 01.00.00 | |
|  PackFlash_VideoRelatedContent | Related content module needs to be placed manually on an article detail page. It displays related articles and files. | 01.00.00 | |
|  PackFlash_VideoRSS | The publish rss module can be used to create rss feeds for a given article category in the system. | 01.00.00 | |
|  PackFlash_VideoSearch | Publish search is a search textbox that you can put anywhere on your site to enable searching of articles. It works with conjunction with the publish search results module. | 01.00.00 | |
|  PackFlash_VideoSearchResults | publish search result module is automatically put on a page that is set to display search results through a publish search module. | 01.00.00 | |