

# Succell Login Modules

---

The Succell Login modules are a pair of modules for adding login and logout functionality to DotNetNuke tabs. Both modules are fully skinnable with custom ascx files, which can easily be deployed with custom skin packages. Additionally, the Logout module offers an option to select the tab, which is displayed after logging out.

## Installation

To install the module, you must first install the Succell.ModuleFramework package, which contains libraries used by the Login module package. A single dummy module named "Succell ModuleFramework" is deployed with this package, which should be marked as a premium module, since it is not intended to be used inside a portal.

Afterwards you can install and use the Succell.Login package.

## The Login module

The login module is a simple yet fully skinnable replacement for the standard login control.

It can be used as a SkinObject for direct usage in a skin, or as a module that is typically placed in a side pane for direct access, or on its own tab to be used as the default login tab.

## Usage as a SkinObject

When you want to use the Login module as a SkinObject, you can simply register the Login control from the Succell.Dnn.Modules.Login.SkinObjects namespace in the Succell.Dnn.Modules.Login assembly:

```
<%@ Register TagPrefix="scl" assembly="Succell.Dnn.Modules.Login"
namespace="Succell.Dnn.Modules.Login.SkinObjects" %>
```

You can then use the Login control anywhere in the ascx file:

```
<scl:Login runat="server" id="login" />
```

For a list of all properties, you can consult the .NET reference topic for the standard ASP Login Control, as it is the base class for the SkinObject. The most important property is the *LayoutTemplate* property. With it, you can specify you own custom skin for the login control:

```
<scl:Login runat="server" id="login">
  <LayoutTemplate>
    <%-- your login template here --%>
  </LayoutTemplate>
</scl:Login>
```

The topic of skinning is discussed later in this document.

The other commonly used property is *LogonTabId*. It allows you to specify a tab to redirect the user to, when he or she is successfully logged on. The allowed values are:

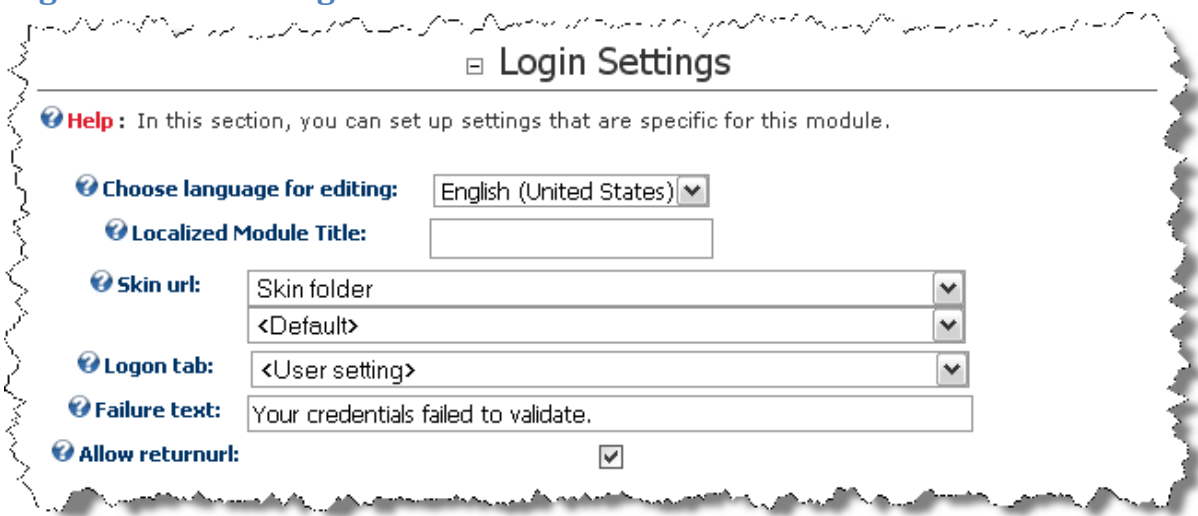
- a tab's id, to redirect to that **specific tab**,
- -1 to redirect to the portal's **Home tab**,
- -2 to remain on the currently **active tab**, or
- -3 to use the tab id specified in the **User Settings**. This is the default value.

```
<scl:Login runat="server" id="login" logontabid="-3" />
```

## Usage as a module

You can add the Login module to any tab by selecting "Succell Login" in the module dropdown list. The module's content is hidden, when viewed by authenticated users, so you should change the view permissions to "unregistered users" only. Otherwise logged in users will be shown an empty module container.

## Login module settings



Login Settings

**Help:** In this section, you can set up settings that are specific for this module.

**Choose language for editing:** English (United States) ▼

**Localized Module Title:**

**Skin url:** Skin folder ▼  
<Default> ▼

**Logon tab:** <User setting> ▼

**Failure text:** Your credentials failed to validate.

**Allow returnurl:**

The first setting for the module is the **localized module title**. This setting is only available, if you are using Apollo Software's ModuleLocalization package, which is available free from Apollo Software's homepage <http://www.apollosoftware.nl/>. You can use this setting to specify a localized title for the module.

The **skin url setting** is used to specify the url, from which to obtain the skin. This skin must be an ascx control that you can place anywhere in your portal. Typically you will deliver the skin with any custom tab skin. The default location for the Login module's skin is "[S]Login/Login.ascx", where [S] is a token for the current skin's folder. If you place the module's skin in your skin's subfolder "Login" and name it "Login.ascx", it will automatically be used by any Login module you add to your tabs.

To specify a url, you can either use the folder and file chooser or specify a custom url. The custom skin url has four tokens, that can be used to access special locations in your portal:

- [M] is expanded to the module's folder, i.e. "DesktopModules/Succell.Login/".
- [S] is expanded to the tab's skin's url, i.e. "Portals/0/Skins/MySkin/".
- [C] is expanded to the container's url, i.e. "Portals/0/Containers/MyContainers/".
- [P] is expanded to the portal's root folder, i.e. "Portals/0/".

Example skins are included with the module and can be used by changing the skin url folder to “*Module folder/Skins*” and choosing a Login skin.

The **Logon Tab** setting is used to specify, where the user should be redirected to after successfully logging in. This can be any of your portal’s tabs, the portal’s home tab, the tab the user was on when logging in, or the tab specified in the DotNetNuke user settings. Simply choose the desired option in the setting’s dropdown list.

The **Failure Text** setting specifies the text to display when the login fails. This text is only displayed, if the selected skin supports this. It is displayed in the FailureText control.

By checking **Allow returnUrl** you can specify, if a returnUrl parameter is used when redirecting after logging in.

## Skinning the Login module

As the Login module uses the standard ASP.NET Login control as its basis, its rules of templating are also valid here. The basic rules are, that you require at least two *ITextControls* (i.e. asp:TextBox), one named “UserName” and one named “Password”, to enable the module/SkinObject to find the data. Additionally you need any *IButtonControl* (i.e. asp:Button, asp:LinkButton), that has its *CommandName* property set to “login”. A basic example can be found in the module’s Skins subfolder under “*SimpleLogin.ascx*”:

```
<%@ Control Language="C#" ClassName="SimpleLogin" %>
```

```
Username: <asp:TextBox runat="server" id="UserName" validationgroup="login"
/><br />
Password: <asp:TextBox runat="server" id="Password" validationgroup="login"
textmode="Password" /><br />
<asp:Button runat="server" id="btnLogin" commandname="login" text="Login"
/>
```

You can also add *IButtonControls* with the *CommandName* property set to “register” and “password” to have them navigate to the portal’s user registration page (if the configuration allows this) or and password retrieval tab:

```
<asp:Button runat="server" id="btnPassword" commandname="password"
text="Forgot password?" />
<asp:Button runat="server" id="btnRegister" commandname="register"
text="Register" />
```

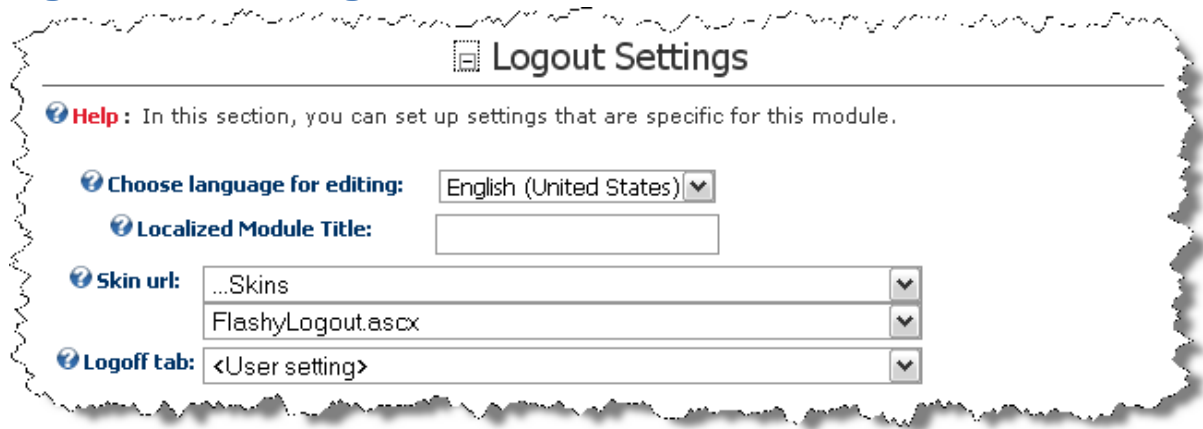
Everything else is up to you. You can create any design around these controls using everything html, css and ASP.NET have to offer.

## The Logout module

The Logout module makes it possible to place a logout control anywhere in your portal’s tabs. It typically consists of a button or link control, that, when clicked, logs out the user. As the process of logging out in DotNetNuke simply requires a specific parameter in the url and can be obtained by calling `Globals.NavigateURL(tabId, "Logoff")`, there is no SkinObject for this module.

When adding the module, you will typically want to set the visibility to “registered users” only, since displaying a logout module to an unauthenticated user does not make sense.

## Logout module settings



The Logout module has three available settings, all of which are analogous to the login setting:

The first setting for the module is the **localized module title**. This setting is only available, if you are using Apollo Software’s ModuleLocalization package, which is available free from Apollo Software’s homepage <http://www.apollosoftware.nl/>. You can use this setting to specify a localized title for the module.

The **skin url setting** is used to specify the url, from which to obtain the skin. This skin must be an ascx control that you can place anywhere in your portal. Typically you will deliver the skin with any custom tab skin. The default location for the Login module’s skin is “[S]Login/Logout.ascx”, where [S] is a token for the current skin’s folder. If you place the module’s skin in your skin’s subfolder “Login” and name it “Logout.ascx”, it will automatically be used by any Logout module you add to your tabs.

To specify a url, you can either use the folder and file chooser or specify a custom url. The custom skin url has four tokens, that can be used to access special locations in your portal:

- [M] is expanded to the module’s folder, i.e. “DesktopModules/Succell.Login/”.
- [S] is expanded to the tab’s skin’s url, i.e. “Portals/0/Skins/MySkin/”
- [C] is expanded to the container’s url, i.e. “Portals/0/Containers/MyContainers/”
- [P] is expanded to the portal’s root folder, i.e. “Portals/0/”

Example skins are included with the module and can be used by changing the skin url folder to “Module folder/Skins” and choosing a Logout skin. Skinning is discussed later in this document.

The **Logoff Tab** setting is used to specify, where the user should be redirected to when logging out. This can be any of your portal’s tabs, the portal’s home tab, the tab the user was on when logging in, or the tab specified in the DotNetNuke user settings. Simply choose the desired option in the setting’s dropdown list.

## Skinning the Logout module

When creating a skin for the Logout module, only one *IButtonControl* is needed with its *CommandName* property set to “logout”:

```
<%@ Control Language="C#" ClassName="SimpleLogout" %>
```

```
<asp:Button runat="server" id="btnLogout" commandname="logout"  
text="Logout" />
```

Everything else is up to you. You can create any design around these controls using everything html, css and ASP.NET have to offer.