

DotNetNuke

Introduction to DotNetNuke

Mr. Jaydeep Bhatt

VISION CONSULTANTS



DotNetNuke Training Setup

- Hardware
 - *Xeon Server with 4 GB RAM (8GB Recommended)*
 - *Overhead Projector*
- Software
 - *Windows Server 2012 R2*
 - *IIS Server*
 - *DotNet Framework 4.5, 4.0 and 3.5*
 - *MS SQL Server 2012 R2 or better with MS SQL Studio*
 - *Latest Visual Studio 2013 Update 4*
 - *MS Office with Power Point*



Recommended Downloads

- Latest DotNetNuke
<http://dotnetnuke.codeplex.com>
- MSBuild Community Tasks
<https://github.com/loresoft/msbuildtasks/releases>
- Christoc's DotNetNuke Module and Theme Development Template
<http://christoctemplate.codeplex.com>
- Code Endeavors DotNetNuke Templates
<http://codeendeavortemplate.codeplex.com>



Day 1: Installation

- What is DotNetNuke
 - *History & Evolvment of DNN.*
- Why DotNetNuke
 - *Reduce Cost.*
- Major Areas of DNN Use
 - *Building Portals & Sites.*
 - *Making Money through Module Development.*
 - *Minting Money with DNN Skinning.*
- Install DNN
 - *Various Options, Auto, Typical & Custom.*
- Moving between Development & Production
 - *Backing up Installation & Database*
 - *Restoring from Backup*



Day 2: Configuration

- Roles
 - *Creating & assigning Roles to Users.*
- Users
 - *Adding Users and assigning Roles.*
- Pages
 - *Adding Pages.*
 - *Updating Pages Settings.*
 - *Setting Page Permissions.*
- Site Settings
 - *Applying Skins at Portal, Page & Module Level*
 - *Using Default, Portal & Module CSS files*
- Module Settings
 - *Assigning Module Permissions.*



Day 2: Configuration

- SEO
 - *Submitting site to Google.*
 - *Defining default Keywords & Description for the Site.*
 - *Overriding Keywords & Description for the individual Page.*
- Users
 - *Adding Users and assigning Roles.*
 - *Assigning Module Permissions.*



Day 3: Extensions

- What are Extensions
 - *Modules.*
 - *Skins.*
 - *Providers.*
- Installing & Upgrading Extensions
 - *Module Definitions.*
- Running Database Scripts
 - *Host -> SQL.*
- Using Popular Extensions
 - *Light Box Gallery Module.*
 - *Report Module.*
 - *Form & List Module.*



DotNetNuke CSS Precedence

1. Module.css

- (~/DesktopModule/ModuleName/module.css)

2. Default.css

- (~/Portals/_default/default.css)

3. Skin.css

- (~/Portals/PortalID/Skins/SkinPackageName/skin.css)

4. Container.css

- (~/Portals/PortalID/Containers/ContainerPackageName/container.css)

5. Style.css (from module template if any)

- (~/DesktopModule/ModuleName/Templates/template.css)

6. Portal.css

- (~/Portals/PortalID/portal.css)

7. Inline.css

- (hard-coded in HTML)



skin.css

It's 2011 and building websites using web standards is a must in every web project. This topic and its pros and cons have been discussed for the past decades by many industry experts. Since web standards encourage the separation of content and its presentation, web designers often use a separate (or sometimes than one) CSS file to design their sites.

Therefore, take this similar approach and apply to DotNetNuke skinning by having all the CSS for typography, layout and positioning, navigation, and other design elements in skin.css. By doing so, the content administrator will have less opportunity to alter the original design unless he/she has access to the skin itself



container.css

- There are two strategies to consider when approaching container design. Either eliminate **container.css** for best performance or separate the style between skins and containers completely by putting all container related CSS in **container.css** file.
- Elements that should go into container.css are headings, design elements for containers, container module navigation. When using **container.css** file, it is advisable to make sure the package can be installed and used in different websites independently.
- **Portal.css** is perfect for site editors when they do not have access to the skin package. It is often used for styling specific site content.



Day 3: Development

- Module Development
 - *Language.*
 - *Model.*
- Using Templates
 - *Built-in DNN Template.*
 - *Auto Package Christoc Template.*
 - *Code Endeavors Templates.*
 - *My Own bhattji Template.*
- DotNetNuke API
 - *Programmatically Creating User & Roles.*
 - *Assigning Roles to User.*
- State Management
 - *Using Session to retain values across modules.*



Day 4: Development

- Debugging Module
 - *Attach to w3wp.exe*
 - *Admin -> Event Viewer*
- To Package and Deploy Module
- Skin Development
 - *Using Skin Objects.*



Day 4: Micro ORM

- What is Micro ORM
- Using PetaPOCO
 - *In .Net Apps*
 - *In DNN 6*
 - *In DNN 7+*
- PetaPOCO considerations
 - *Object Qualifier not supported.*
 - *Null Date value not supported.*
 - *Work-arounds.*
- PetaPOCO sample Module Development
 - *Best Practice Development.*



Day 5: Integration

- WebAPI - Service Framework for DNN7
 - *Developing Hello World Service using WebAPI*
 - *Using Sample Authentication over WebAPI*
- Running Scheduled jobs
 - *DotNetNuke Scheduler*
 - *Developing Scheduler Job for DNN*
 - *Adding & Configuring Item in DNN*



Day 5: Integration

- List of Useful Tools & Sites
 - *Codeplex.com*
 - *MSBuild.Community.Tasks*
 - *MyGenerationSoftware*
- Problem Solving
 - *Q & A on previous day work.*
 - *More detail on selected topics.*
 - *Useful Tips & Tricks.*
 - *Best Practice Development.*
 - *Preparing Hosting Environment.*





Thank You

Mr. Jaydeep Bhatt

9377838485

jaydeep@bhattji.com

<http://bhattji.com>

<http://vision-consultants.com>

VISION CONSULTANTS

