



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

Table of Contents

Table of Contents	1
Introduction.....	3
Types of Installation	3
Installation Features.....	3
Installation/Upgrade of DotNetNuke 3.0.....	4
Setting Up your Site Ready for Installation	4
Web.config Installation/Upgrade settings	5
Member Roles Prototype Settings	7
DotNetNuke Install Template	9
<dotnetnuke> Node	9
<scripts> Node	10
<superuser> Node	10
<desktopmodules> Node	11
<portals> Node	11
Installation of Optional Resources	11
Installation Process	12
Upgrade Process.....	14
Installing Additional Portals	16
Example Portals.resources File	17
Installing Additional Resources	19
Modules.....	20
Skins/Containers.....	20
Language Packs	20
What went wrong?.....	21
ERROR: Could not connect to database specified in connectionString for SqlDataProvider.	21
Cannot open database requested in login '{Database}'. Login fails. Login failed for user '{user}'	22



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

Could not find stored procedure 'dbo.GetPortals'	23
Appendix – Example Install Template	25



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

Introduction

With the release of version 3.0, DotNetNuke has introduced an enhanced Installation/Upgrade procedure. In version 2.1.2 and earlier, Installation/Upgrade was controlled by AutoUpgrade. In addition some resources (Skins, Modules, Containers) could be installed using the Web Upload control, which was part of the File Manager. Both of these processes have been significantly enhanced.

This document describes these enhancements.

Types of Installation

There are three major types of installation/site provisioning available in DotNetNuke and these will be covered in this document

- Installation of DotNetNuke 3.0 or Upgrade of an existing installation
- Installation/Creation of new Portals
- Installation of Resources
 - o Skins/Containers
 - o Modules
 - o Language Packs

Installation Features

There are also a number of “features” in the application that

- AutoUpgrade – the ability to automatically detect whether an installation/upgrade is required and install or upgrade automatically
- Install Template – An xml based template that can control the installation, including host settings, desktop modules and portals
- DNNConfig file – the file saves the current database version, to avoid a call to the Database on every App_Start
- New Install Page (Install.aspx) – this page provides feedback on the installation/upgrade process
- Install Folder – this folder provides administrators with the ability to bulk FTP new resources to a location on the file system, and either through a scheduled event or through the Install page. Bulk install these resources
- Upload Resource – Admin or Host UI components that deal with resources that can be installed, have an “Upload” action button that can be used to launch the installation.



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

Installation/Upgrade of DotNetNuke 3.0

One of the biggest changes in version 3 of DotNetNuke, is the enhancement to the Installation/Upgrade process. In earlier versions of DotNetNuke, an AutoUpgrade function detected the Database Version, compared it with the Assembly Version, and ran any incremental scripts required to bring the Database Version up to the current Assembly Version.

There was no conceptual difference between an Install and an Upgrade – only the number of scripts that were executed. So, for example, an upgrade from 2.1.1 to 2.1.2 required the execution of only the 2.1.2 script, whereas an upgrade from 2.0.1 required the execution of 5 scripts – 2.0.2, 2.0.3, 2.0.4, 2.1.1, 2.1.2. A fresh Installation of 2.1.2 required the execution of 19 version scripts.

DotNetNuke 3.0 modifies this process so that fresh installs do not have to execute 34 scripts (as of 3.0.12), but instead execute 2 scripts DotNetNuke.Schema and DotNetNuke.Data. Upgrades still execute the incremental scripts.

Setting Up your Site Ready for Installation

If you are installing a new instance of DotNetNuke then the first steps you need to accomplish will depend on whether you are creating a site on a local intranet or on a remote server. The complete process will be described here. If DotNetNuke is being installed into a Shared Hosting Plan, then the Hosting Provider has probably already set up the site and database for you.

1. Unzip the package to the physical location where the Application will reside. On local intranet configurations this can be anywhere you want (eg C:\DotNetNuke). On remote hosting servers you will need to upload the files to your website following the procedures provided by your hosting provider.
2. In a local intranet configuration, create a Virtual Directory in IIS called DotNetNuke that points to this physical directory. Make sure that Default.aspx is defined as a default document for this Virtual Directory.
3. Create a Database in SQL Server and Create a User with dbowner privileges.
4. Set the connection string in web.config (ie. `<add key="SiteSqlServer" value="Server=(local);Database=DotNetNuke;uid=;pwd=;" />`). Here, replace the Server, Database, uid, and password by the relevant values for your database. In Shared Hosting environments, you will have been provided these values.
5. Give the ASP.NET worker process the appropriate Security access to the folders of your website. In Windows 2000/IIS5 this is the {Server}\ASPNET User Account, while in Windows 2003/IIS6 this is the NT AUTHORITY\NETWORK SERVICE User Account. To enable this use File Explorer to locate the physical folder you used in (1) above. Right-click on the folder, and select "Properties", and choose the Security Tab. The required permissions are as follows:
 - Root (and all child folders) – Read Access
 - Root – Write Access (and Modify Access if expecting to create child portals)

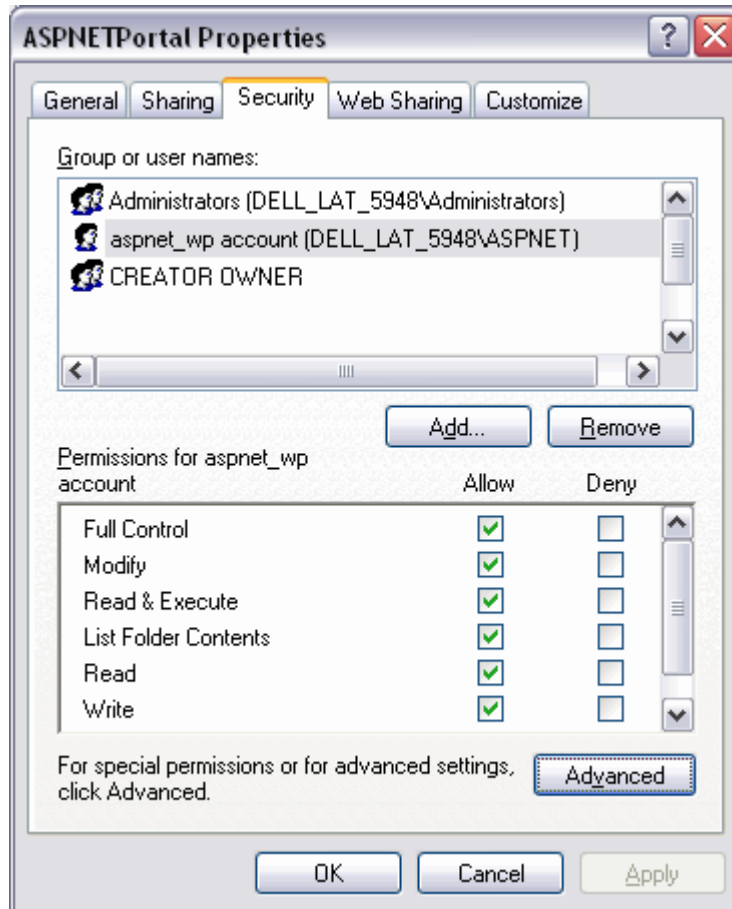


DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

- DesktopModules – Write, Modify Access (if installing 3rd party Private Assemblies)
- Portals (and all child folders) – Write, Modify Access (to allow File Manager access and Skin/Container access)



If you are likely to be installing Language Packs you will need Write and Modify Access to all child folders of the root.

Web.config Installation/Upgrade settings

Before browsing to your site and initiating the installation, there are a number of **web.config** settings, that you might want to set, in addition to the SiteSqlServer connection string.

In the **<appSettings>** group of settings, just below SiteSqlServer are three settings that control the Installation/Upgrade.

<appSettings>



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```
<add key="SiteSqlServer" value="Server=(local);Database=DotNetNuke;uid=;pwd=;" />
<add key="MachineValidationKey"
value="F9D1A2D3E1D3E2F7B3D9F90FF3965ABDAC304902" />
<add key="MachineDecryptionKey"
value="F9D1A2D3E1D3E2F7B3D9F90FF3965ABDAC304902F8D923AC" />
<add key="MachineValidationMode" value="SHA1" />
<add key="InstallTemplate" value="DotNetNuke.install.resources" />
<add key="AutoUpgrade" value="false" />
<add key="UseDnnConfig" value="false"/>
<!-- Show missing translation keys (for development use) -->
<add key="ShowMissingKeys" value="false" />
</appSettings>
```

The first of these settings ("**InstallTemplate**") defines an install template, and is only used on Installation of DotNetNuke 3.0. We will describe this template in detail later in this document. At this stage it is only important to realize that you can set this value to control the installation.

The second setting "**AutoUpgrade**" determines whether the Upgrade/Install process runs automatically. By default this value is set to true. Whatever the setting, DotNetNuke will determine the current Database Version and Assembly Version and redirect to the Install page if they are different. If set to true, the Installation or Upgrade will proceed automatically. If set to false, the Install page will report the differences and stop.

The third setting "**UseDnnConfig**" which is also set to true by default, tells DotNetNuke whether to read the current Database version from the **dnn.config** file in the /Install folder. As upgrades do not happen on every Application Start using this file can reduce the number of Database hits, and improve performance. At the end of Installation/Upgrade the current version is written to this file. If the Assembly Version is the same as the version in this file, nothing will happen. If it is different then AutoUpgrade will confirm that the Database Version is indeed different and proceed accordingly.

There are two additional settings that it is important to review. Much further down the web.config file the data providers are defined.

```
<data defaultProvider="SqlDataProvider">
  <providers>
    <clear />
    <add name="SqlDataProvider"
      type="DotNetNuke.Data.SqlDataProvider, DotNetNuke.SqlDataProvider"
      connectionStringName="SiteSqlServer"
      upgradeConnectionString=""
      providerPath="~\Providers\DataProviders\SqlDataProvider\"
      objectQualifier=""
      databaseOwner="dbo" />
  </providers>
</data>
```

The **objectQualifier** attribute is used to control the prefix applied to every object in the SQL Server database. By default this is set to an empty string "", so for instance the "Portals" table is



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

created as "Portals". If, for example, the **objectQualifier** was set to "dnn_" then the "Portals" table would be created as "dnn_Portals" and stored procedures etc that referenced the Portals table would reference it as dnn_Portals. Setting this value is recommended, as it allows DotNetNuke to exist within an environment where multiple applications might need to use the same database.

In the case of Upgrades it is important to confirm that this value is consistent with the existent installation ie. once DotNetNuke is installed the objectqualifier cannot be changed.

The **databaseOwner** attribute is used to identify the user that "owns" the database objects. By default it is set to "dbo" and in most cases this attribute should be left this way. However, some SQL Server DBAs (in particular in Shared Hosting Environments) may not grant the user you identify in the connection string db_owner privileges. In this case, set the **databaseOwner** attribute to your login user. Note this user will need to have the following SQL Permissions, at a minimum:

- db_datareader (necessary for dotnetnuke)
- db_datawriter (necessary for dotnetnuke)
- db_ddladmin (necessary for MemberRoles)
- db_securityadmin (necessary during installation of MemberRoles)

The previous two settings are only relevant for the default MS SQL Server Data Provider. If you are using an alternate data provider, you will have to refer to its documentation to configure the provider properly.

Member Roles Prototype Settings

DotNetNuke 3.0 includes a prototype of the member/roles provider that will be included in the upcoming release of ASP.NET 2.0 (codenamed "Whidbey"). There are a number of settings that can be configured for this provider in the web.config file. However, DotNetNuke only really supports modifying 4 of the parameters.

```
<memberrolesprototype>
  <membership userIsOnlineTimeWindow="15">
    <providers>
      <add name="DNNSQLMembershipProvider"
        type="DotNetNuke.Security.Membership.DNNSQLMembershipProvider,
DNNSQLMembershipProvider"
        connectionStringName="SiteSqlServer"
        enablePasswordRetrieval="true"
        enablePasswordReset="true"
        requiresQuestionAndAnswer="false"
        minRequiredPasswordLength="4"
        minRequiredNonalphanumericCharacters="0"
        requiresUniqueEmail="false"
        passwordFormat="Encrypted"/>
```



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```
        applicationName="/"
        description="Stores and retrieves membership data from the local Microsoft SQL
Server database" />
    </providers>
</membership>
<roleManager
    cacheRolesInCookie="true"
    cookieName=".ASPXROLES"
    cookieTimeout="30"
    cookiePath="/"
    cookieRequireSSL="false"
    cookieSlidingExpiration="true"
    createPersistentCookie="false"
    cookieProtection="All">
    <providers>
        <add name="DNNSQLRoleProvider"
            type="DotNetNuke.Security.Role.DNNSQLRoleProvider, DNNSQLRoleProvider"
            connectionStringName="SiteSqlServer"
            applicationName="/"
            description="Stores and retrieves roles data from the local Microsoft SQL Server
database" />
    </providers>
</roleManager>
<profile enabled="true">
    <providers>
        <add name="AspNetSqlProvider"
            type="DotNetNuke.Users.Profile.DNNSQLProfileProvider,
DNNSQLProfileProvider"
            connectionStringName="SiteSqlServer"
            applicationName="/"
            description="Stores and retrieves profile data from the local Microsoft SQL
Server database" />
    </providers>
</properties>
    <add name="FirstName" type="string" allowAnonymous="true" />
    <add name="LastName" type="string" allowAnonymous="true" />
    <add name="Unit" type="string" allowAnonymous="true" />
    <add name="Street" type="string" allowAnonymous="true" />
    <add name="City" type="string" allowAnonymous="true" />
    <add name="Region" type="string" allowAnonymous="true" />
    <add name="PostalCode" type="string" allowAnonymous="true" />
    <add name="Country" type="string" allowAnonymous="true" />
    <add name="Telephone" type="string" allowAnonymous="true" />
    <add name="Fax" type="string" allowAnonymous="true" />
    <add name="Cell" type="string" allowAnonymous="true" />
    <add name="Website" type="string" allowAnonymous="true" />
```



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```
<add name="IM" type="string" allowAnonymous="true" />
<add name="TimeZone" type="integer" allowAnonymous="true" />
<add name="PreferredLocale" type="string" allowAnonymous="true" />
</properties>
</profile>
<anonymousIdentification
  enabled="true"
  cookieName=".ASPXANONYMOUS"
  cookieTimeout="100000"
  cookiePath="/"
  cookieRequireSSL="false"
  cookieSlidingExpiration="true"
  cookieProtection="None"
  domain="" />
</memberrolesprototype>
```

minRequiredPasswordLength – controls the minimum password length. This has been set to 4 in the default web.config, in order to support the historic use of host/host for the username and password of the default superuser account. Increasing the default will require that the passwords provided in the **<superuser>** and **<administrator>** nodes of the install template be modified.

minRequiredNonalphanumericCharacters – controls the minimum number of non-alphanumeric characters required in the password. This has been set to 0 as historically DotNetNuke has not required non-alphanumeric characters. Again, setting this value to any positive number will require the **<superuser>** and **<administrator>** nodes of the install template be modified.

cookieName – both the rolemanager and anonymousIdentification section of the provider allow you to set the cookie name. This is probably not necessary, but is available.

DotNetNuke Install Template

As mentioned above, a clean installation uses a template file to control the installation. In web.config the **"InstallTemplate"** setting determines which file is used, so you can either edit the default file **DotNetNuke.install.resources** or create your own and edit the **"InstallTemplate"** setting. The install template is an xml file located in the /Install folder.

The following sections provide an explanation of the nodes/attributes in this file. The Appendix includes a copy of the template that is packaged in the Release.

<dotnetnuke> Node

The **<dotnetnuke>** node is the root node of the template. All other nodes are children of **<dotnetnuke>**. There are 7 immediate children

- **<description>** child



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

Provides a description of the template.

- **<scripts> child**
Defines the script files that builds the Database (these files must be located in the folder indicated by the "providerPath" attribute of the default Data provider in web.config) - see later
- **<version> child**
The database version that the <scripts> files build. This is usually the current version, but if the version is less than the current version then the appropriate upgrade scripts are run to make the database version = app version. For example if the version of the template is 3.0.9 and the Assembly Version is 3.0.12 then the scripts defined by <scripts> are run followed by 3.0.10, 3.0.11 and 3.0.12.
- **<superuser> child**
Information about the superuser (Host Account) - see later
- **<settings> child**
The default HostSettings for the application – the child nodes of this node correspond to the HostSettings table in the database, and are managed under the Host/Host Settings menu (after installation)
- **<desktopmodules> child**
The desktop modules to install for this installation – see later
- **<portals> child**
The portal(s) to be installed – see later

<scripts> Node

The < scripts > node defines the scripts that build the Database. It can contain any number of <script> child nodes.

- **<script name="xxx"> child**
Each <script> child node defines a single script that is to be executed. The name attribute can be used to identify the script file (ie what it does). The element value is the filename of the script file.

<superuser> Node

The <superuser> node configures the default superuser (or host account). It has 7 immediate children.

- **<firstname> child**
The first name of the superuser
- **<lastname> child**
The last name of the superuser
- **<username> child**
The user name of the superuser
- **<password> child**
The password of the superuser – note that the password must meet the requirements configured in the <memberrolesprototype> settings above
- **<email> child**



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

- The email address of the superuser
- **<locale> child**
The locale (culture) of the superuser
- **<timezone> child**
The timezone (expressed in minutes from GMT) of the superuser

<desktopmodules> Node

The <desktopmodules> node configures the desktopmodules to be installed. It has a number of <desktopmodule> nodes, one for each module to be installed. The schema for the <desktopmodule> nodes is similar to the structure used for custom modules in the dnn file. In fact the schema for the <desktopmodule> node is the same as the schema for the <folder> node in the dnn Module Installer file (see the Module Dev Guide for more info on these nodes) except that we are not adding files in this process so there are no <files> nodes.

<portals> Node

The <portals> Node configures the default portals. It can contain a number of <portal> nodes which have 8 immediate child nodes.

- **<portalname> child**
The name of the portal (will display in the TitleBar)
- **<administrator> child**
Configures the Administrator User for the portal - has the same schema as the <superuser> node – see above
- **<description> child**
A description of the portal (used in the Html Page META Tags)
- **<keywords> child**
keywords for the portal (used in the Html Page META Tags)
- **<templatefile> child**
The filename of the template file used for the portal (must be physically located in the Portals/_default directory)
- **<homedirectory> child**
Sets the portal's home directory (usually left blank – defaults to Portals/[PortalId])
- **<portalaliases><portalalias> child**
Portal Aliases for the portal (usually left blank – defaults to the location where the installation runs)
- **<ischild> child**
Must be blank (future use)

Installation of Optional Resources

Optional Resources such as language packs, skins and Private Assemblies can also be installed automatically during the Installation Process. The /Install folder has six sub-folders, Container, Language, Module, Portal, Skin and Template. The zip package – described later in this document – for the appropriate resource is placed in the appropriate folder, and installed automatically by the Installer.



DotNetNuke Installation Guide

Date: 3/11/2005

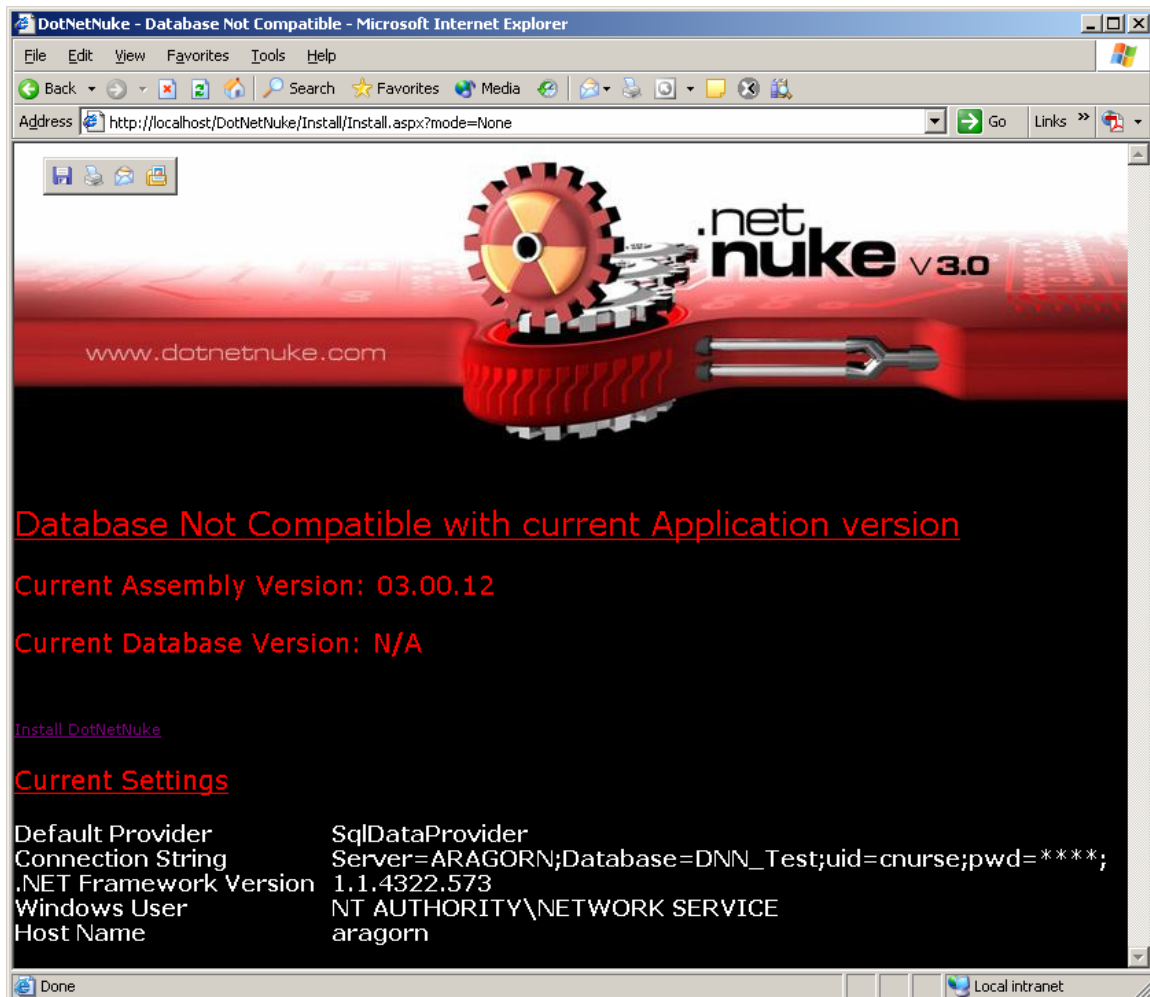
Author: Charles.Nurse

Installation Process

So we have configured our web.config file and our install template and we are ready to install DotNetNuke. The installation is quite strait-forward. Browse to the root folder – in a local environment this will probably be <http://localhost/DotNetNuke>, while on a remote installation it will be www.mydomain.com.

The AutoUpgrade function will detect that the Database is empty and redirect to Install.aspx. If the “**AutoUpgrade**” setting is set to false then the Install page will report that the Database and Assembly versions are not compatible.

The administrator can then trigger the Install by using the url – “**Install/Install.aspx?mode=Install**”, or by selecting the link on the Page.



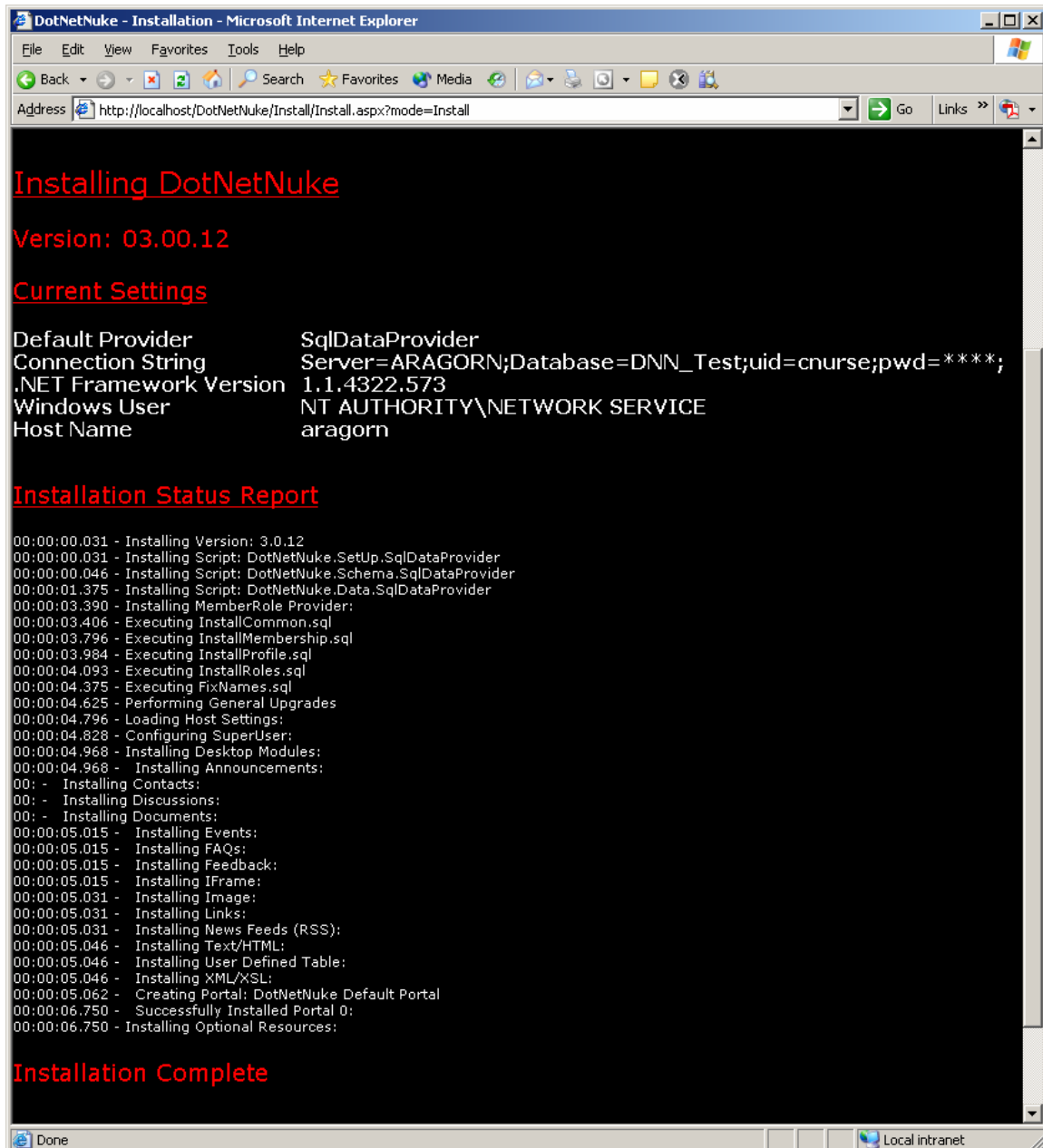


DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

If 'AutoUpgrade' is set to true then the Installation process starts automatically. No matter which way the installation is triggered, feedback similar to the following will be displayed.



Lets have a look at what happens in this default installation.



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

1. The Installer reports the version of the template being installed
2. The Installer reports that the installation scripts were executed. (any incremental scripts for later versions would then be executed)
3. The Installer reports that the scripts required for the MemberRole Provider were executed, as well as an extra script required to fix the default names of some of the constraints
4. The sections of the Install template are then parsed
 - a. Host Settings
 - b. SuperUser
 - c. Desktop Modules (each module installed is listed)
5. The portal(s) are created.
6. Optional Resources are then Installed

Upgrade Process

The Upgrade process is similar, except that not as much happens.

The AutoUpgrade function will detect that the Database is different and redirect to Install.aspx. If the “**AutoUpgrade**” setting is set to false then the Install page will report that the Database and Assembly versions are out of sync, as above.

The administrator can then trigger the Upgrade by using the url – “**Install/Install.aspx?mode=Install**” or by selecting the link on the page.

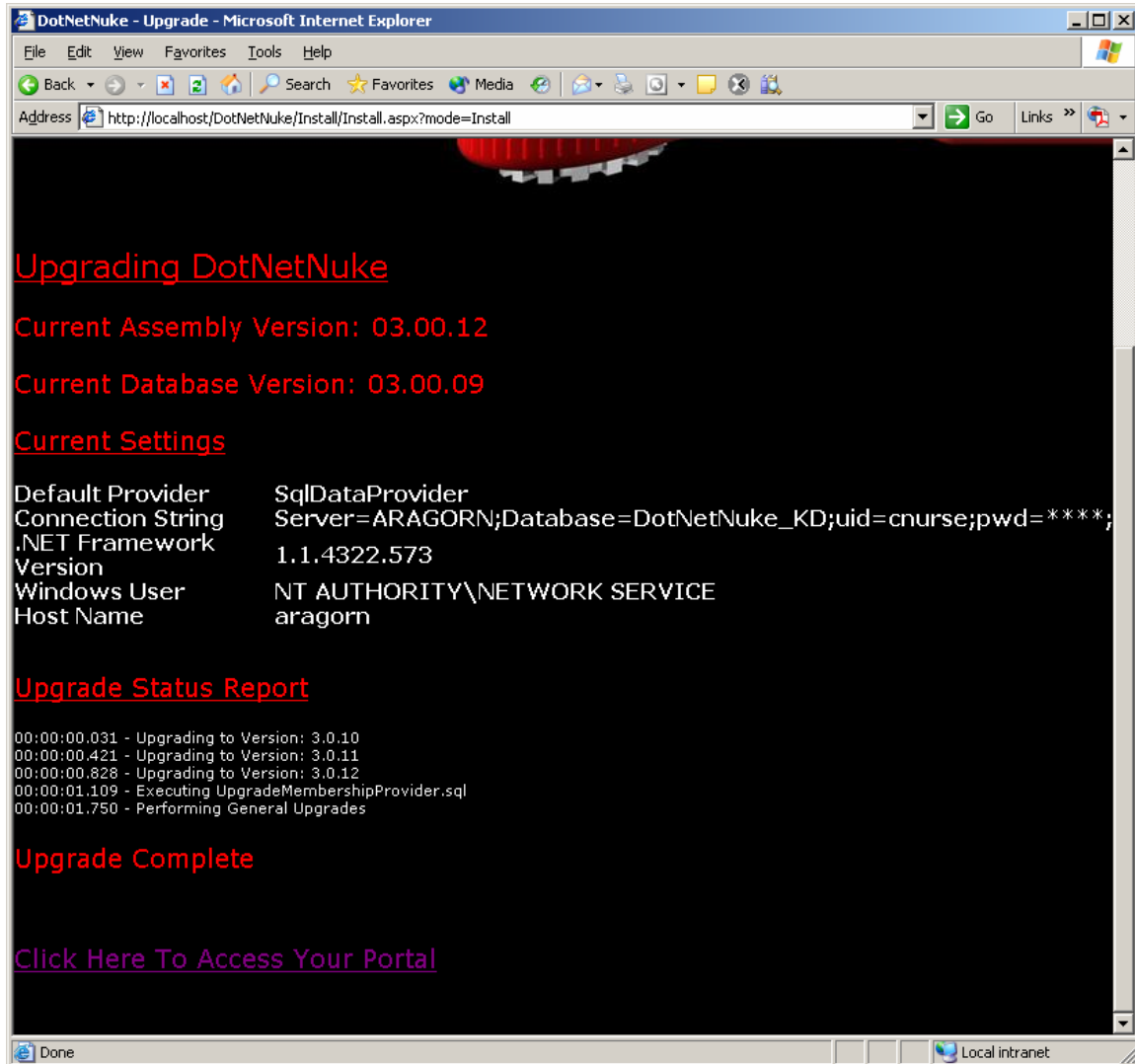
If ‘**AutoUpgrade**’ is set to true then the Upgrade process starts automatically, providing feedback similar to the following.



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse



In this case the Installer reports that the Assembly Version is 3.0.12 but the database version is 3.0.9. It executes the scripts 3.0.10, 3.0.11 and 3.0.12 as well as a script to update the MemberRole database objects.



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

Installing Additional Portals

There are two ways to add additional portals once DotNetNuke has been installed. The first, most obvious method is to login as Host, select the Portals option from the Host menu, and enter the relevant information for the Portal.

There is, however, a second method for installing portals, that can batch install multiple portals, in one operation.

This second approach is modeled after the installation of the portals during the DotNetNuke installation process. A template file "**portals.resources**" can be placed in the /Install/Portal



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

folder. This file is similar to the install template, except that the only child nodes processed are the **<portals><portal>** nodes.

Example Portals.resources File

```
<dotnetnuke>
  <portals>
    <portal>
      <portalname>DotNetNuke Personal Site</portalname>
      <administrator>
        <firstname>Administrator</firstname>
        <lastname>Account</lastname>
        <username>admin</username>
        <password>admin</password>
        <email></email>
      </administrator>
      <description>Default Personal Site</description>
      <keywords>Default, DotNetNuke, CMS, Web, Future</keywords>
      <templatefile>Personal Site.template</templatefile>
      <portalaliases>
        <portalalias>localhost/DotNetNuke/ps</portalalias>
        <portalalias>ARAGORN/DotNetNuke/ps</portalalias>
        <portalalias>192.168.1.102/DotNetNuke/ps</portalalias>
      </portalaliases>
      <ischild>true</ischild>
    </portal>
    <portal>
      <portalname>DotNetNuke Club Site</portalname>
      <administrator>
        <firstname>Administrator</firstname>
        <lastname>Account</lastname>
        <username>admin</username>
        <password>admin</password>
        <email></email>
      </administrator>
      <description>Default DotNetnuke Club Site</description>
      <keywords>Default, DotNetNuke, CMS, Web, Future</keywords>
      <templatefile>Club or Organization Site.template</templatefile>
      <portalaliases>
        <portalalias>localhost/DotNetNuke/club</portalalias>
        <portalalias>ARAGORN/DotNetNuke/club</portalalias>
        <portalalias>192.168.1.102/DotNetNuke/club</portalalias>
      </portalaliases>
      <ischild>true</ischild>
    </portal>
  </portals>
</dotnetnuke>
```



DotNetNuke Installation Guide

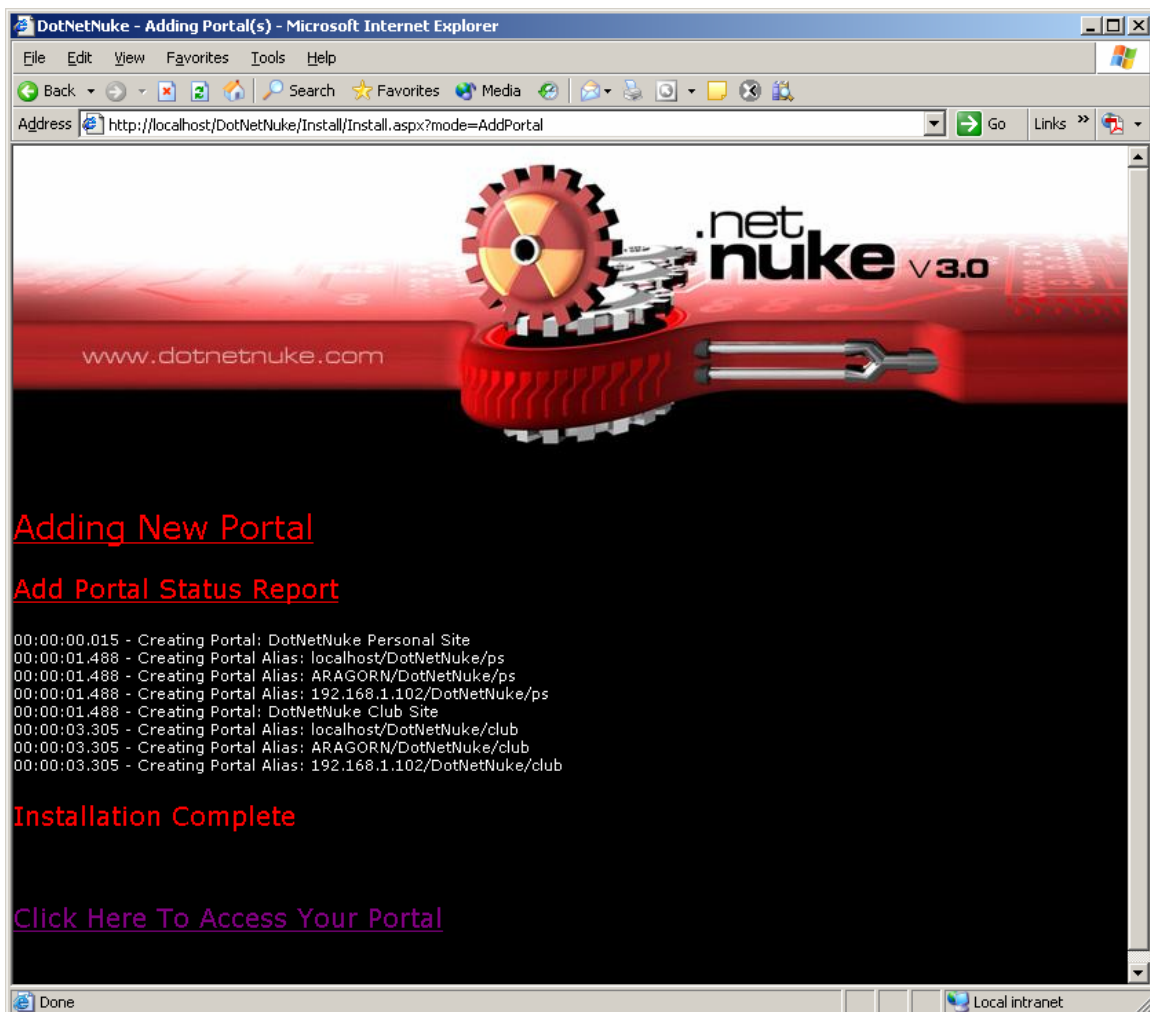
Date: 3/11/2005

Author: Charles.Nurse

Once the file is saved to the Install/Portal folder, the installation of the additional portals defined in this file can be triggered in one of two ways.

1. Using the scheduled event "Install Resources"
2. Manually browsing to the Install page with the following url
"Install/Install.aspx?mode=AddPortal".

When you use the latter method the Install page again provides feedback, similar to that shown below.





DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

Installing Additional Resources

Additional resource such as Skins, Modules (packaged as Private Assemblies) and Language Packs can also be installed in two ways.

1. Through the Appropriate Admin/Host menu in the Application. Each resource can be installed through the User Interface element appropriate for that resource. Thus a Module is installed through the Host/Module Definitions menu.
2. Adding the Installation package to the appropriate folder under /Install. The /Install folder has a number of sub-folders. Zip files placed in these folders can be installed by browsing to the url "**Install/Install.aspx?mode=InstallResources**", or through the Install Resources Scheduled Event. If installed using the Install page, a status report similar to the one shown below is produced.





DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

The Resource needs to be packaged as a zip file before being installed. As this document focuses on the installation we do not describe here how to package the resource. This is covered in other documents.

The following are the resources that can be installed, using both of these approaches

Modules

- Through Application UI Host/Module Definitions, Select the “Upload New Module” action from the Action menu or from the link at the bottom of the control
- Through Installer by placing zip file in /Install/Module.
- Module Development and Packaging are described in the “DotNetNuke Module Developer’s Guide”

Skins/Containers

- Through Application UI Skins menu option on either the Host or Admin menus, Select the “Upload Container” or “Upload Skin” actions from the Action menu or from the link at the bottom of the control
- Through Installer by placing zip file in /Install/Skin or /Install/Container.
- Skin/Container Development and Packaging are described in the “DotNetNuke Skinning” document.

Language Packs

- Through Application UI Host/Languages, Select the “Upload Language Pack” action from the Action menu or from the link at the bottom of the control
- Through Installer by placing zip file in /Install/Language.



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

What went wrong?

In this section we will attempt to provide answers to some common failure scenarios.

ERROR: Could not connect to database specified in connectionString for SqlDataProvider.



This error is caused by an incorrect connection string – in the example above the Database is mis-spelt, it should be DNN_Test not DNN_Teest



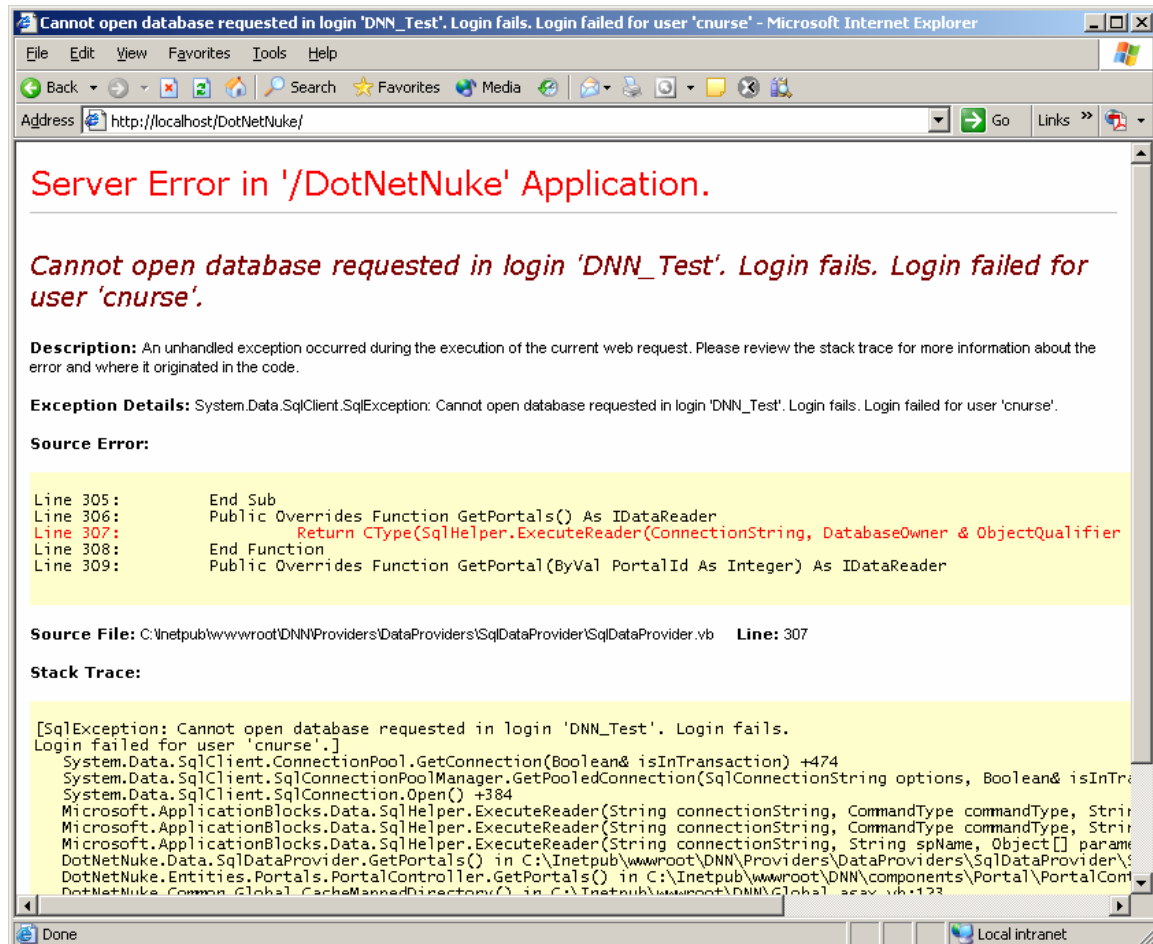
DotNetNuke Installation Guide

Date: 3/11/2005

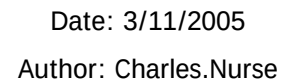
Author: Charles.Nurse

Cannot open database requested in login '{Database}'. Login fails. Login failed for user '{user}'

Also - Login failed for user '{user}'



This error is caused when the user specified in the connection string cannot log in to the database – in this case “cnurse” has not been created in the SQL Server database. It can also occur if the user credentials provided in the connection string are incorrect (eg the password is not entered correctly)





DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

These scenarios are actually quite similar. In both cases the **dnn.config** file is reporting that the Application and Database are in sync, when in fact they are not. In the first case it is because the {**objectqualifier**} is incorrect, in the second the database is actually empty.



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

Appendix – Example Install Template

```
<dotnetnuke>
  <description>This is the default DotNetNuke Host Installation Template</description>
  <scripts>
    <script name="Setup">DotNetNuke.SetUp.SqlDataProvider</script>
    <script name="Schema">DotNetNuke.Schema.SqlDataProvider</script>
    <script name="Data">DotNetNuke.Data.SqlDataProvider</script>
  </scripts>
  <version>03.00.12</version>
  <superuser>
    <firstname>SuperUser</firstname>
    <lastname>Account</lastname>
    <username>host</username>
    <password>host</password>
    <email>host</email>
    <locale>en-US</locale>
    <timezone>0</timezone>
  </superuser>
  <settings>
    <ControlPanel>Admin/ControlPanel/IconBar.ascx</ControlPanel>
    <Copyright>Y</Copyright>
    <DemoPeriod></DemoPeriod>
    <DemoSignup>N</DemoSignup>
    <DisableUsersOnline>Y</DisableUsersOnline>
    <EncryptionKey Secure="True"></EncryptionKey>

    <FileExtensions>swf,jpg,jpeg,jpe,gif,bmp,png,doc,xls,ppt,pdf,txt,xml,xsl,css,zip</FileExtension
s>
    <HostCurrency>USD</HostCurrency>
    <HostEmail></HostEmail>
    <HostFee></HostFee>
    <HostPortalId>0</HostPortalId>
    <HostSpace></HostSpace>
    <HostTitle>DotNetNuke</HostTitle>
    <HostURL>http://www.dotnetnuke.com</HostURL>
    <PaymentProcessor>PayPal</PaymentProcessor>
    <PerformanceSetting>3</PerformanceSetting>
    <ProcessorPassword Secure="True"></ProcessorPassword>
    <ProcessorUserId Secure="True"></ProcessorUserId>
    <ProxyPassword Secure="True"></ProxyPassword>
    <ProxyPort></ProxyPort>
    <ProxyServer></ProxyServer>
    <ProxyUsername Secure="True"></ProxyUsername>
    <SiteLogBuffer>1</SiteLogBuffer>
    <SiteLogHistory>0</SiteLogHistory>
```



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```
<SiteLogStorage>D</SiteLogStorage>
<SkinUpload>G</SkinUpload>
<SMTPServer></SMTPServer>
<SMTPAuthentication></SMTPAuthentication>
<SMTPUsername Secure="True"></SMTPUsername>
<SMTPPassword Secure="True"></SMTPPassword>
<UseCustomErrorMessages>Y</UseCustomErrorMessages>
<UseFriendlyUrls>Y</UseFriendlyUrls>
<UsersOnlineTime>20</UsersOnlineTime>
<SchedulerMode>2</SchedulerMode>
<AutoAccountUnlockDuration>10</AutoAccountUnlockDuration>
</settings>
<desktopmodules>
  <desktopmodule>
    <name>Announcements</name>
    <description>This module renders a list of announcements. Each announcement
includes title, text and a "read more" link, and can be set to automatically expire after a particular
date.</description>
    <version>01.00.00</version>

    <businesscontrollerclass>DotNetNuke.Modules.Announcements.AnnouncementsController,
DotNetNuke.Modules.Announcements</businesscontrollerclass>
    <modules>
      <module>
        <friendlyname>Announcements</friendlyname>
        <controls>
          <control>
            <src>DesktopModules/Announcements/Announcements.ascx</src>
            <type>View</type>
            <helpurl></helpurl>
          </control>
          <control>
            <key>Edit</key>
            <title>Edit Announcements</title>
            <src>DesktopModules/Announcements/EditAnnouncements.ascx</src>
            <type>Edit</type>
            <helpurl></helpurl>
          </control>
        </controls>
      </module>
    </modules>
  </desktopmodule>
  <desktopmodule>
    <name>Contacts</name>
```



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

<description>This module renders contact information for a group of people, for example a project team. Contacts includes an edit page, which allows authorized users to edit the Contacts data stored in the database.</description>

<version>01.00.00</version>

<businesscontrollerclass>DotNetNuke.Modules.Contacts.ContactController,
DotNetNuke.Modules.Contacts</businesscontrollerclass>

<modules>

<module>

<friendlyname>Contacts</friendlyname>

<controls>

<control>

<src>DesktopModules/Contacts/Contacts.ascx</src>

<type>View</type>

<helpurl></helpurl>

</control>

<control>

<key>Edit</key>

<title>Edit Contacts</title>

<src>DesktopModules/Contacts/EditContacts.ascx</src>

<type>Edit</type>

<helpurl></helpurl>

</control>

</controls>

</module>

</modules>

</desktopmodule>

<desktopmodule>

<name>Discussions</name>

<description>This module renders a group of message threads on a specific topic. Discussion includes a Read/Reply Message page, which allows authorized users to reply to existing messages or add a new message thread. The data for Discussion is stored in the SQL database.</description>

<version>01.00.00</version>

<businesscontrollerclass>DotNetNuke.Modules.Discussions.DiscussionController,
DotNetNuke.Modules.Discussions</businesscontrollerclass>

<modules>

<module>

<friendlyname>Discussions</friendlyname>

<controls>

<control>

<src>DesktopModules/Discussions/Discussion.ascx</src>

<type>View</type>

<helpurl></helpurl>

</control>

<control>

<key>Edit</key>



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```
<title>Edit Discussions</title>
<src>DesktopModules/Discussions/DiscussDetails.ascx</src>
<type>Anonymous</type>
<helpurl></helpurl>
</control>
</controls>
</module>
</modules>
</desktopmodule>
<desktopmodule>
  <name>Documents</name>
  <description>This module renders a list of documents, including links to browse or
download the document. Documents includes an edit page, which allows authorized users to edit
the information about the Documents (for example, a friendly title) stored in the SQL
database.</description>
  <version>01.00.00</version>
  <businesscontrollerclass>DotNetNuke.Modules.Documents.DocumentController,
DotNetNuke.Modules.Documents</businesscontrollerclass>
  <modules>
    <module>
      <friendlyname>Documents</friendlyname>
      <controls>
        <control>
          <src>DesktopModules/Documents/Document.ascx</src>
          <type>View</type>
          <helpurl></helpurl>
        </control>
        <control>
          <key>Edit</key>
          <title>Edit Documents</title>
          <src>DesktopModules/Documents/EditDocs.ascx</src>
          <type>Edit</type>
          <helpurl></helpurl>
        </control>
      </controls>
    </module>
  </modules>
</desktopmodule>
<desktopmodule>
  <name>Events</name>
  <description>This module renders a list of upcoming events, including time and
location. Individual events can be set to automatically expire from the list after a particular date.
Events includes an edit page, which allows authorized users to edit the Events data stored in the
SQL database.</description>
  <version>01.00.00</version>
```



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```
<businesscontrollerclass>DotNetNuke.Modules.Events.EventController,
DotNetNuke.Modules.Events</businesscontrollerclass>
<modules>
  <module>
    <friendlyname>Events</friendlyname>
    <controls>
      <control>
        <src>DesktopModules/Events/Events.ascx</src>
        <type>View</type>
        <helpurl></helpurl>
      </control>
      <control>
        <key>Edit</key>
        <title>Edit Events</title>
        <src>DesktopModules/Events/EditEvents.ascx</src>
        <type>Edit</type>
        <helpurl></helpurl>
      </control>
      <control>
        <key>Settings</key>
        <title>Events Settings</title>
        <src>DesktopModules/Events/Settings.ascx</src>
        <type>Edit</type>
        <helpurl></helpurl>
      </control>
    </controls>
  </module>
</modules>
</desktopmodule>
<desktopmodule>
  <name>FAQs</name>
  <description>FAQs allow you to manage a list of Frequently Asked Questions and their
corresponding Answers.</description>
  <version>01.00.00</version>
  <businesscontrollerclass>DotNetNuke.Modules.FAQs.FAQsController,
DotNetNuke.Modules.FAQs</businesscontrollerclass>
  <modules>
    <module>
      <friendlyname>FAQs</friendlyname>
      <controls>
        <control>
          <src>DesktopModules/FAQs/FAQs.ascx</src>
          <type>View</type>
          <helpurl></helpurl>
        </control>
        <control>
```



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```
<key>Edit</key>
<title>Edit FAQs</title>
<src>DesktopModules/FAQs/EditFAQs.ascx</src>
<type>Edit</type>
<helpurl></helpurl>
</control>
</controls>
</module>
</modules>
</desktopmodule>
<desktopmodule>
  <name>Feedback</name>
  <description>Feedback allows visitors to send messages to the Administrator of the
portal.</description>
  <version>01.00.00</version>
  <businesscontrollerclass></businesscontrollerclass>
  <modules>
    <module>
      <friendlyname>Feedback</friendlyname>
      <controls>
        <control>
          <src>DesktopModules/Feedback/Feedback.ascx</src>
          <type>View</type>
          <helpurl></helpurl>
        </control>
        <control>
          <key>Settings</key>
          <title>Feedback Settings</title>
          <src>DesktopModules/Feedback/Settings.ascx</src>
          <type>Edit</type>
          <helpurl></helpurl>
        </control>
      </controls>
    </module>
  </modules>
</desktopmodule>
<desktopmodule>
  <name>IFrame</name>
  <description>IFrame is an Internet Explorer browser feature which allows you to display
content from another website within a frame on your site.</description>
  <version>01.00.00</version>
  <businesscontrollerclass></businesscontrollerclass>
  <modules>
    <module>
      <friendlyname>IFrame</friendlyname>
      <controls>
```



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```
<control>
  <src>DesktopModules/IFrame/IFrame.ascx</src>
  <type>View</type>
  <helpurl></helpurl>
</control>
<control>
  <key>Edit</key>
  <title>Edit IFrame</title>
  <src>DesktopModules/IFrame/EditIFrame.ascx</src>
  <type>Edit</type>
  <helpurl></helpurl>
</control>
</controls>
</module>
</modules>
</desktopmodule>
<desktopmodule>
  <name>Image</name>
  <description>This module renders an image using an HTML IMG tag. The module
simply sets the IMG tags src attribute to a relative or absolute URL, so the image file does not
need to reside within the portal. The module also exposes height and width attributes, which
permits you to scale the image. Image includes an edit page, which persists these settings to the
portals configuration file. </description>
  <version>01.00.00</version>
  <businesscontrollerclass></businesscontrollerclass>
  <modules>
    <module>
      <friendlyname>Image</friendlyname>
      <controls>
        <control>
          <src>DesktopModules/Images/ImageModule.ascx</src>
          <type>View</type>
          <helpurl></helpurl>
        </control>
        <control>
          <key>Edit</key>
          <title>Edit Image</title>
          <src>DesktopModules/Images/EditImage.ascx</src>
          <type>Edit</type>
          <helpurl></helpurl>
        </control>
      </controls>
    </module>
  </modules>
</desktopmodule>
<desktopmodule>
```



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```
<name>Links</name>
<description>This module renders a list of hyperlinks. Links includes an edit page,
which allows authorized users to edit the Links data stored in the SQL database.</description>
<version>01.00.00</version>
<businesscontrollerclass>DotNetNuke.Modules.Links.LinkController,
DotNetNuke.Modules.Links</businesscontrollerclass>
<modules>
  <module>
    <friendlyname>Links</friendlyname>
    <controls>
      <control>
        <src>DesktopModules/Links/Links.ascx</src>
        <type>View</type>
        <helpurl></helpurl>
      </control>
      <control>
        <key>Edit</key>
        <title>Edit Links</title>
        <src>DesktopModules/Links/EditLinks.ascx</src>
        <type>Edit</type>
        <helpurl></helpurl>
      </control>
      <control>
        <key>Settings</key>
        <title>Links Settings</title>
        <src>DesktopModules/Links/Settings.ascx</src>
        <type>Edit</type>
        <helpurl></helpurl>
      </control>
    </controls>
  </module>
</modules>
</desktopmodule>
<desktopmodule>
  <name>News Feeds (RSS)</name>
  <description>News Feed allows you to consume syndicated news feeds in Rich Site
Summary (RSS) format.</description>
  <version>01.00.00</version>
  <businesscontrollerclass></businesscontrollerclass>
  <modules>
    <module>
      <friendlyname>News Feeds (RSS)</friendlyname>
      <controls>
        <control>
          <src>DesktopModules/News/RssModule.ascx</src>
          <type>View</type>
        </control>
      </controls>
    </module>
  </modules>
</desktopmodule>
```



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```

        <helpurl></helpurl>
    </control>
    <control>
        <key>Edit</key>
        <title>Edit News Feeds (RSS)</title>
        <src>DesktopModules/News/EditRss.ascx</src>
        <type>Edit</type>
        <helpurl></helpurl>
    </control>
</controls>
</module>
</modules>
</desktopmodule>
<desktopmodule>
    <name>Text/HTML</name>
    <description>This module renders a snippet of HTML or text. The Html/Text module
includes an edit page, which allows authorized users to the HTML or text snippets directly. The
snippets are stored in the SQL database.</description>
    <version>01.00.00</version>
    <businesscontrollerclass>DotNetNuke.Modules.Html.HtmlTextController,
DotNetNuke.Modules.Html</businesscontrollerclass>
    <modules>
        <module>
            <friendlyname>Text/HTML</friendlyname>
            <controls>
                <control>
                    <src>DesktopModules/HTML/HtmlModule.ascx</src>
                    <type>View</type>
                    <helpurl></helpurl>
                </control>
                <control>
                    <key>Edit</key>
                    <title>Edit Text/HTML</title>
                    <src>DesktopModules/HTML/EditHTML.ascx</src>
                    <type>Edit</type>
                    <helpurl></helpurl>
                </control>
            </controls>
        </module>
    </modules>
</desktopmodule>
<desktopmodule>
    <name>User Defined Table</name>
    <description>User Defined Table allows you to create a custom data table for managing
tabular information.</description>
    <version>01.00.00</version>

```



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```
<businesscontrollerclass></businesscontrollerclass>
<modules>
  <module>
    <friendlyname>User Defined Table</friendlyname>
    <controls>
      <control>
        <src>DesktopModules/UserDefinedTable/UserDefinedTable.ascx</src>
        <type>View</type>
        <helpurl></helpurl>
      </control>
      <control>
        <key>Edit</key>
        <title>Edit User Defined Table</title>

<src>DesktopModules/UserDefinedTable/EditUserDefinedTable.ascx</src>
        <type>Edit</type>
        <helpurl></helpurl>
      </control>
      <control>
        <key>Manage UDT</key>
        <title>Manage UDT</title>

<src>DesktopModules/UserDefinedTable/ManageUserDefinedTable.ascx</src>
        <type>Edit</type>
        <helpurl></helpurl>
      </control>
    </controls>
  </module>
</modules>
</desktopmodule>
<desktopmodule>
  <name>XML/XSL</name>
  <description>This module renders the result of an XML/XSL transform. The XML and
XSL files are identified by their UNC paths in the xmlsrc and xslsrc properties of the module. The
Xml/Xsl module includes an edit page, which persists these settings to the SQL
database.</description>
  <version>01.00.00</version>
<businesscontrollerclass></businesscontrollerclass>
<modules>
  <module>
    <friendlyname>XML/XSL</friendlyname>
    <controls>
      <control>
        <src>DesktopModules/XML/XmlModule.ascx</src>
        <type>View</type>
        <helpurl></helpurl>
```



DotNetNuke Installation Guide

Date: 3/11/2005

Author: Charles.Nurse

```
</control>
<control>
  <key>Edit</key>
  <title>Edit XML/XSL</title>
  <src>DesktopModules/XML/EditXml.ascx</src>
  <type>Edit</type>
  <helpurl></helpurl>
</control>
</controls>
</module>
</modules>
</desktopmodule>
</desktopmodules>
<portals>
  <portal>
    <portalname>DotNetNuke Default Portal</portalname>
    <administrator>
      <firstname>Administrator</firstname>
      <lastname>Account</lastname>
      <username>admin</username>
      <password>admin</password>
      <email></email>
    </administrator>
    <description>Default DotNetNuke Portal</description>
    <keywords>Default, DotNetNuke, CMS, Web, Future</keywords>
    <templatefile>DotNetNuke.template</templatefile>
    <portalaliases>
      <portalalias></portalalias>
    </portalaliases>
    <ischild>false</ischild>
  </portal>
</portals>
</dotnetnuke>
```