



DotNetNuke 安装指南

Charles Nurse

版本: 1.0.03

最后更新: 2005 年 11 月 28 日

分类: 安装

本文档中的信息，包括 URL 和其他 Internet Web 站点参考，如有变更，恕不另行通知。使用此文档的所有风险和使用此文档造成的一切后果，由使用者本人承担。

本文示例中的公司，组织，产品，域名，电子邮件地址，logo，人名，地名和事件描述均属虚构。如有类同，存属巧合

遵守任何适用的版权法是用户的责任。在不限版权所辖权利的前提下，未经 Perpetual Motion Interactive Systems, Inc. 的明确书面许可，无论出于何种目的，均不得以任何形式或借助任何手段（电子、机械、影印或其他手段）复制或传播文档中的任何部分，或将其存储于或引入检索系统。

Perpetual Motion Interactive Systems 可能拥有对本文档内容的专利、专利申请、商标、版权或其他知识产权。除了任何 Microsoft 授权许可协议所提供的明确书面许可，拥有本文档并不赋予您任何有关这些专利、专利申请、商标、版权或其他知识产权的许可。

2005 Perpetual Motion Interactive Systems。保留所有权利。

DotNetNuke. 和 DotNetNuke logo 均系 Perpetual Motion Interactive Systems 在美国及/或其他国家（或地区）的注册商标。

其中提到的其他真实公司及产品名称可能是其各自所有者的商标。

摘要

本文描述 DotNetNuke 3.x 版本和 4.x 版本的安装过程

目录

摘要.....	3
DOTNETNUKE安装指南	6
警告	6
介绍.....	7
DOTNETNUKE3. X安装	8
设置你的站点，准备安装	8
为安装配置WEB.CONFIG文件	16
升级到DOTNETNUKE3. X	25
备份你的站点	25
准备升级.....	25
配置升级用的WEB.CONFIG	25
安装 DOTNETNUKE4.X.....	28
概述.....	28
使用哪个安装包?	28
使用STARTER KIT	29
使用或安装源码包	33
配置WEB.CONFIG	36
安装DotNetNuke	37
升级到DOTNETNUKE4.X.....	39
备份你的站点	39
准备升级.....	39
配置升级到 4.x版本的WEB.CONFIG	41
升级DotNetNuke	41
安装可选资源和模块	43
安装附加PORTALS.....	43
安装附加资源.....	46
出错情况	48
错误：不能连接到CONNECTIONSTRING中指定的SQLDATA PROVIDER数据库(COULD NOT CONNECT TO DATABASE SPECIFIED IN CONNECTIONSTRING FOR SQLDATA PROVIDER.).....	49
不能打开登陆请求的数据库 “{DATABASE}”。登陆失败。使用 “{USER}” 用户登陆失败。(CANNOT OPEN DATABASE REQUESTED IN LOGIN '{DATABASE}'. LOGIN FAILS. LOGIN FAILED FOR USER '{USER}')	50
找不到存储过程 “DBO.GETPORTALS” (COULD NOT FIND STORED PROCEDURE 'DBO.GETPORTALS')	51
附录一DotNetNuke 安装模板	51
模板示例.....	54
其它信息	55
错误和遗漏	56
附录A：文档历史	56



DotNetNuke 安装指南

警告

如果你是升级现有程序，你必须仔细阅读升级章节，尤其是关于配置 `web.config` 的部分。

在所有处理之前，请备份你的 `web.config` 文件（建议以 `web.backup.resources` 或 `web_backup.config` 命名）

介绍

在 3.0 版本发布中，DotNetNuke 引入了一种更强大的安装和升级方法，并且，在后续版本中，这一方法不断更新。

在 2.1.1 和更早的版本中，安装和升级是由自动升级（AutoUpgrade）控制的。另外，一些资源文件（皮肤，模块，容器）可以使用 Web Upload 控件来安装的，Web Upload 控件是文件管理的一部分。现在这两个过程都被大大改进了。

安装的类型

DotNetNuke 中提供三种主要的安装/站点形式

- ✧ 安装 DotNetNuke3.x
- ✧ 将现有的安装升级到 DotNetNuke3.x
- ✧ 安装 DotNetNuke4.x
- ✧ 升级一个现有的 3.x 或 4.x 安装
- ✧ 安装/创建一个新的 Portals
- ✧ 安装资源
 - 皮肤/容器
 - 模块
 - 语言包

安装特点

就安装和升级而言，程序还有一些“特征”

- ✧ 自动升级—自动检测是否需要安装/升级，并自动安装或升级
- ✧ 安装模板—可以通过一个基于 XML 的模板来控制安装，包括主机设置，模块和 Portals。
- ✧ DNNConfig 文件—这个文件保存当前数据库的版本，这样可能避免每次 App_Start 时访问数据库
- ✧ 安装日期—指定是否为一次“新”的安装自动生成“新”键
- ✧ 新安装页面（Install.aspx）—这个页面在安装/升级过程中显示回馈信息
- ✧ 安装文件夹—此文件夹通过批量 FTP 来管理在文件系统中新资源的定位，并且通过安装包，批量安装这些资源。

资源上传—在 Admin 或 Host 有专门的 UI 组件来处理能被安装的资源，在组件中有一个“Upload”按钮行为来启动安装。

DotNetNuke3. x 安装

在 DotNetNuke3.x 中，一个最大的改变就是对于安装/升级的改进。在 DotNetNuke 的早期版本中，有一个自动升级功能检测数据库版本，将此版本信息与程序集版本进行比较，然后运行能将数据库从当前版本升级到程序对应版本的脚本。

这样在安装和升级这间没什么本质的区别—区别仅在于执行的脚本的数量不同。例如，从 2.1.1 升级到 2.1.2 只需执行 2.1.2 的脚本，然而如果从 2.0.1 升级的话，就要执行五个脚本—2.0.2,2.0.3,2.0.4,2.1.1,2.1.2。一个 2.1.2 版本的全新安装共需执行 19 个版本的脚本。

DotNetNuke3.x 中修改了这一过程，全新安装不必执行 34 个版本的脚本（如 3.0.12），只需要执行 2 个脚本 DotNetNuke.Schema 和 DotNetNuke.Data。升级仍然执行剩余脚本。

设置你的站点，准备安装

如果你要安装 DotNetNuke 的新实例，你第一步要做的，取决于你是在本地 intranet 上还是远程服务器上创建站点。下面将描述整个过程。如果 DotNetNuke 被安装在一个共享主机中，那么主机的提供者可能已经为你设置好了站点和数据库。

1. 解压打包文件

第一步将打包的文件解压到程序运行的位置。从 3.1 版本开始，DotNetNuke 有两个包可用：

- ✧ DotNetNuke_X.Y.Z_Install.zip—安装包—这个包里包含 ascx,aspx 文件，同时包含其它内容文件和编译后的程序集(dlls)，但是，包里不包含 VisualBasic 代码文件。如果你不想修改核心文件，请使用这个安装包。
- ✧ DotNetNuke_X.Y.Z_Source.zip—源码包—里面包含了核心 DotNetNuke 工程的所有相关文件，包括 Visual Basic 源文件。

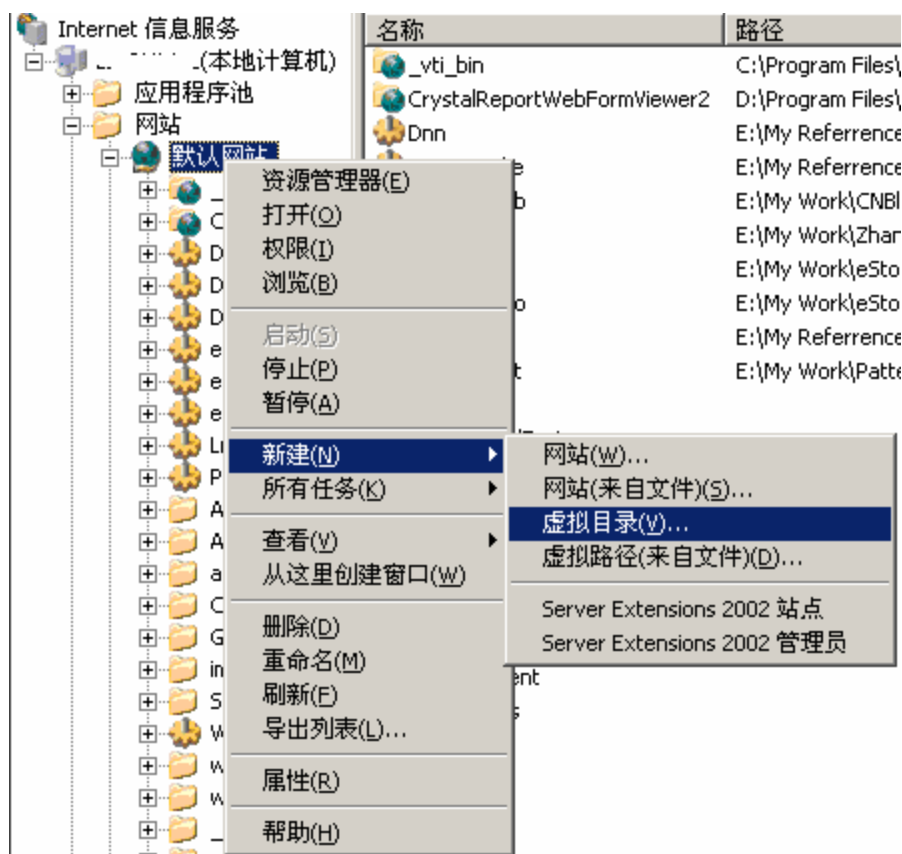
（这里 X.Y.Z 是版本号，例如：DotNetNuke_3.2.0_Install.zip）

虽然 IIS（Internet Information Server）站点的默认位置是在 C:\Inetpub\wwwroot\...，但是如果在本机 intranet 上配置，你可以将站点放在任何位置（如：C:\DotNetNuke）。

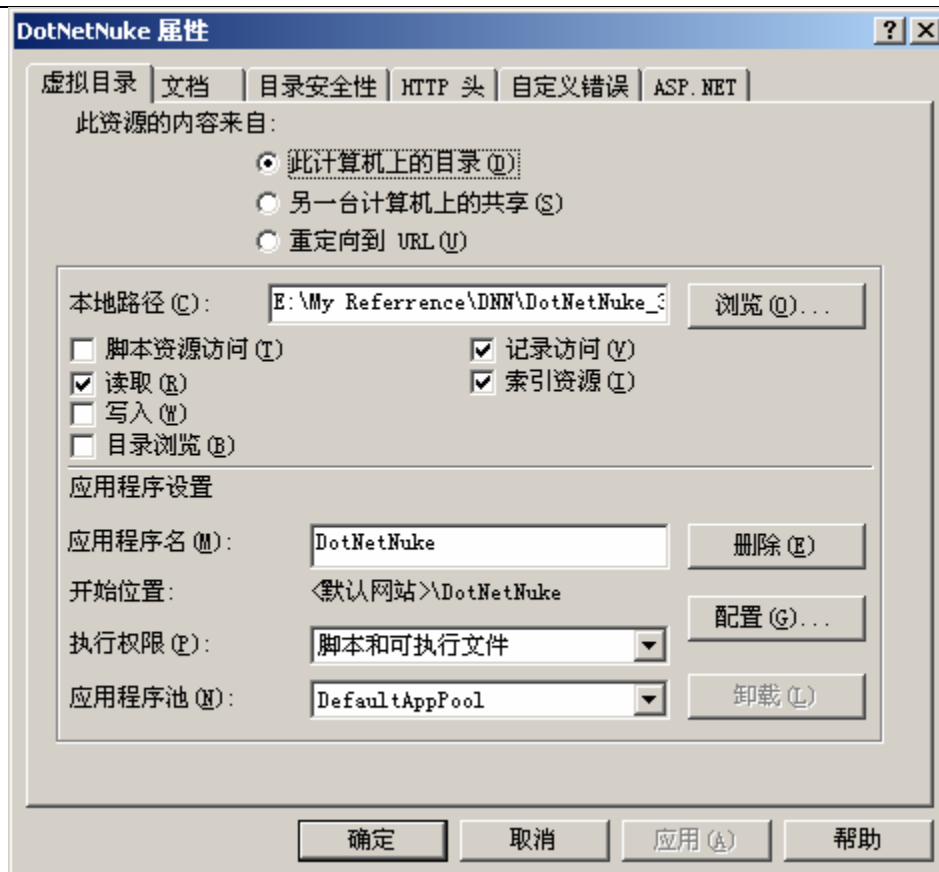
如果在远程主机上安装 DotNetNuke，你必须按照主机提供者指定的过程，将你的文件上传到你的站点。

2. 配置 IIS

在本地 intranet 上配置的话，在 IIS 中创建一个名为 DotNetNuke 的虚拟目录，指向物理路径。在 IIS6 中右击“默认网站”，选择“新建->虚拟目录”。



将启动一个向导，指导你配置你的虚拟目录。当你使用向导创建了虚拟目录之后，你还可以通过在这个新虚拟目录上右击，选择“属性”，来编辑这个虚拟目录的属性。

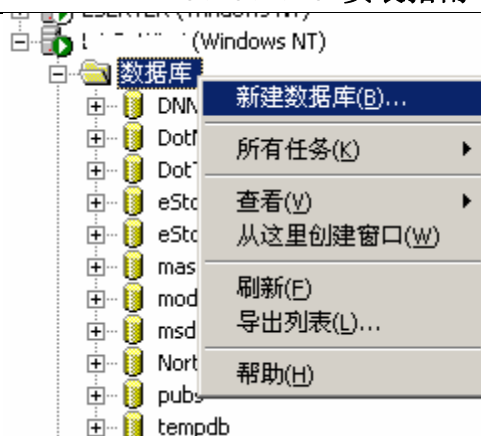


3. 在 SQLServer 中创建数据库

在远程主机环境中，你的主机提供者可能已经为你配置了 SQL Server 数据库，并且为你连接到数据库提供的指导。

如果在本地配置，你要创建一个新的数据库。怎样创建你的本地数据库，要看你使用的数据库工具是什么。

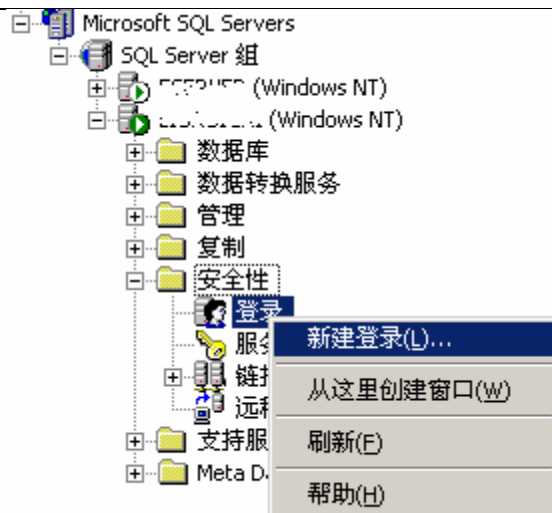
在企业管理器（SQL Server2000）中，你可以通过在“数据库”上右击，并选择“新建数据库...”来创建一个新数据库。



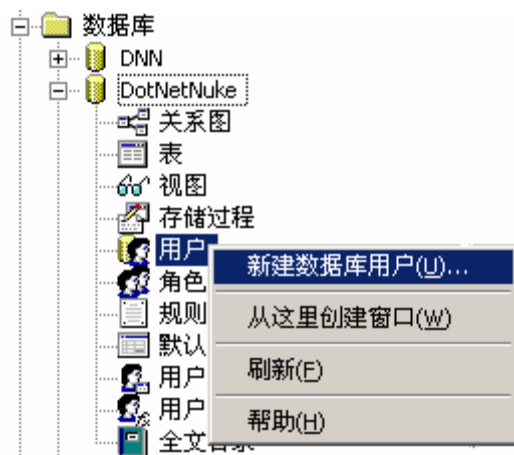
在“新数据库”窗口输入你的数据库的名字（你可以输入任意的名字，但默认名字是 DotNetNuke），点击“确定”。你也可能自己设置 Collation 名，并为数据库指定文件名，但是一般情况下这些属性使用默认值就可以了。



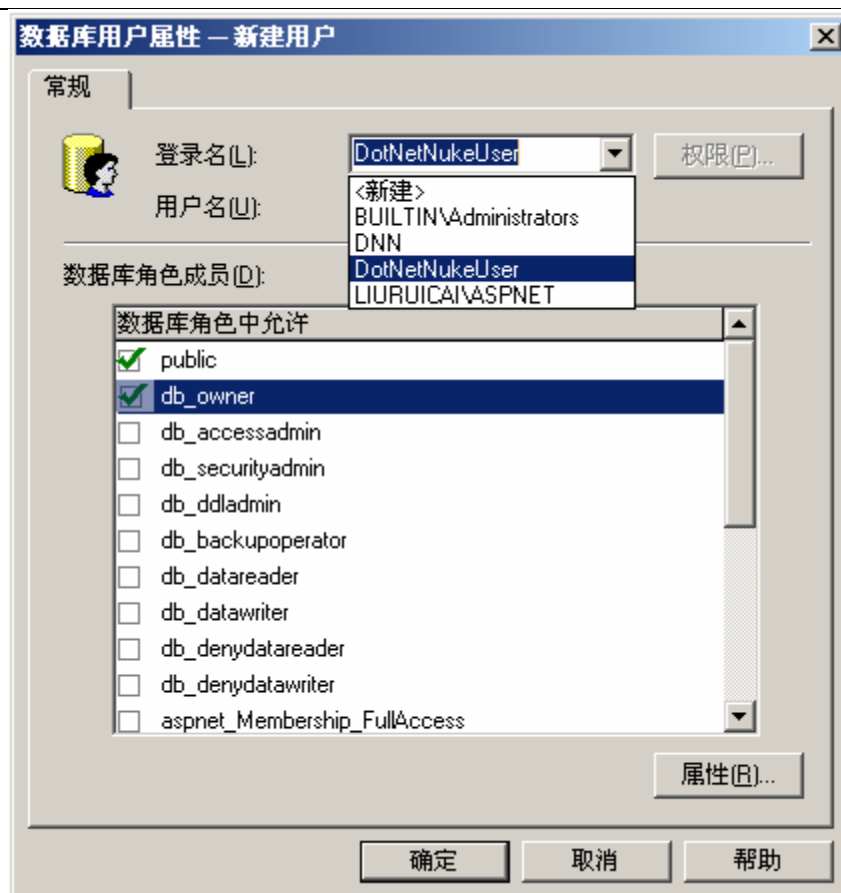
下一步是为你的 DotNetNuke 数据库配置一个数据库用户。如果还没有用户，右击“安全/登陆”



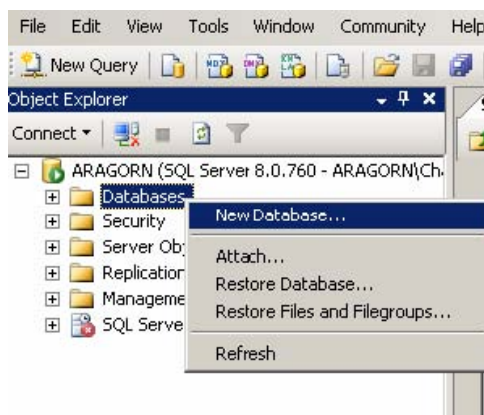
当你创建了登陆这后，在“用户”（在你刚才新建的 DotNetNuke 数据库下）上右击并选择“新用户”



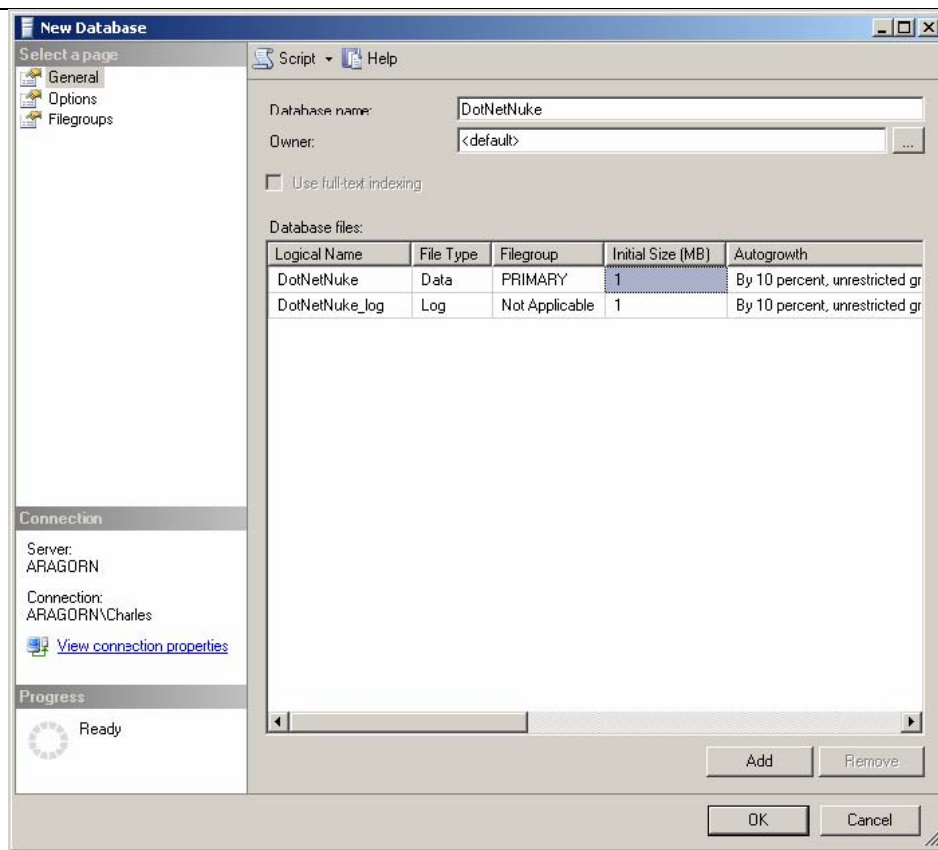
在“新用户”窗口的下拉列表中选择你要使用的用户帐号，并设置 `public` 和 `db_owner` 权限（事实上你不必设置完全的 `db_owner` 权限，后面的在 `web.config` 中配置 `data provider` 最小权限的章节中你将看到这一点）



这一过程在 SQL Server Management Studio(SQL Server 2005)中更加简单。右击 Databases 并选择 “New Database”。



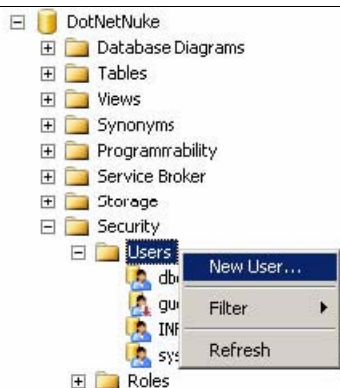
在 “New Database” 窗口中输入数据库名。



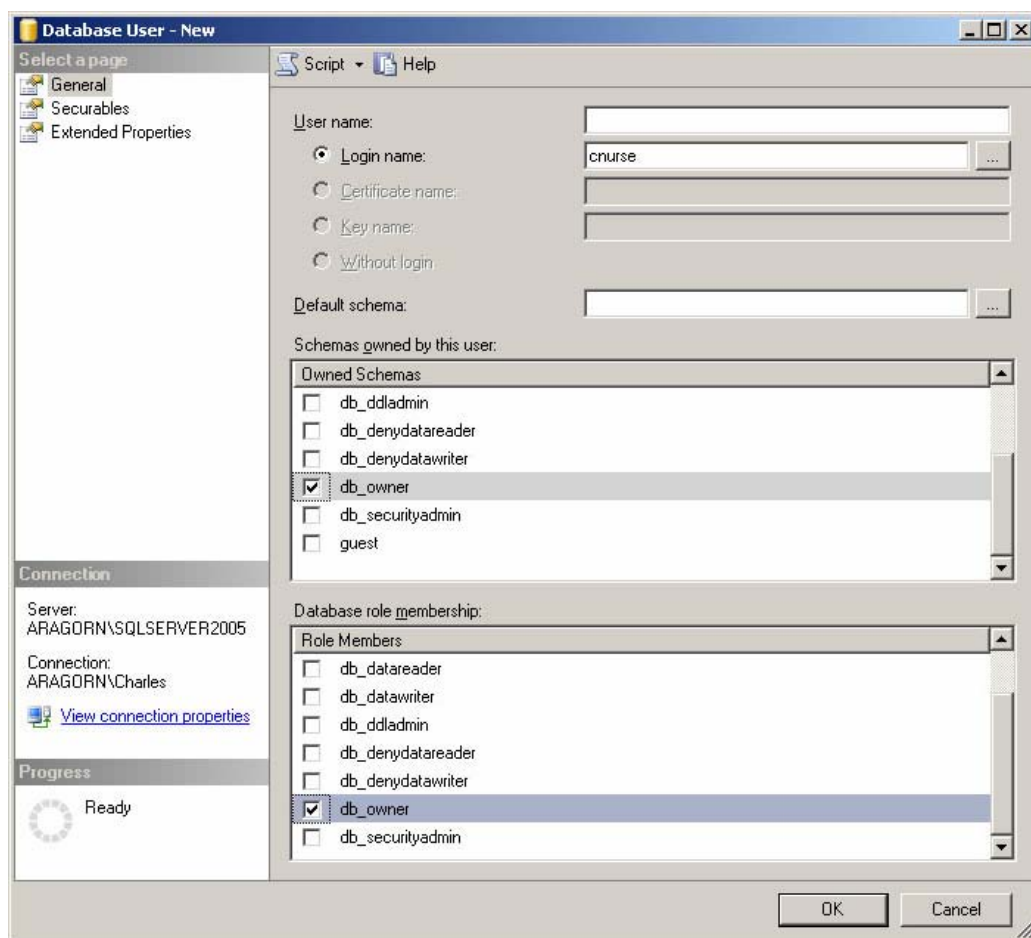
在“Security/Logins”右击，创建一个新登陆。



在你新建的数据库上右击，为你的新数据库添加一个新用户（在 SQL Server Management Studio 中，这位置是在你的数据库的 Security 节点下）



确认设置了 db_owner 权限。

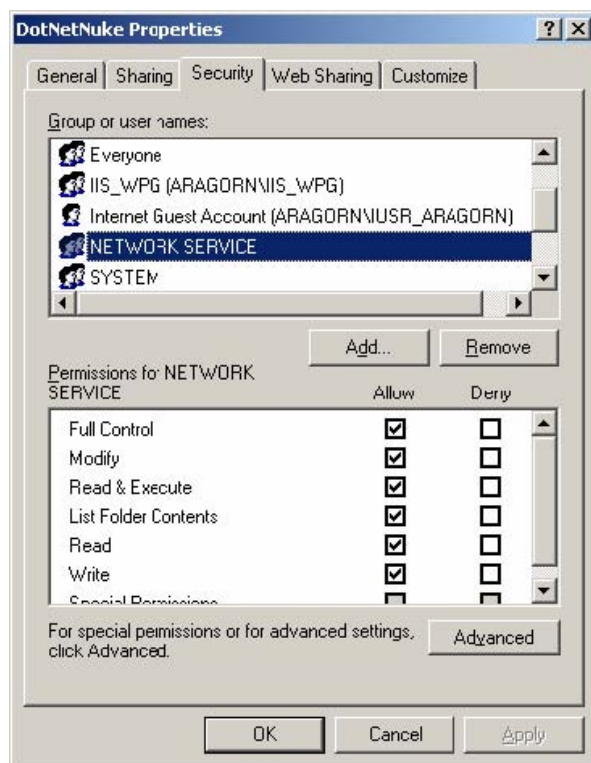


4. 设置文件权限

下一步是为 Asp.NET 工作进程安装 DotNetNuke 设置 NTFS 文件权限。这是因为 DotNetNuke 需要创建路径和文件。根据你的 windows 版本不同，你要设置权限的帐户是不一样的。

- ✧ 在 Windows2000/IIS5 中的帐户是(Server)\ASPNET 帐户，
- ✧ 在 Windows2003/IIS5 中的帐户是 NT AUTHORITY\NETWORK SERVICE 帐户

要设置权限，可以通过资源浏览器找到你在上面（1）中的物理文件夹，右击这个文件夹选择“属性”，选择“安全”标签页。



对于 DotNetNuke 下的各个文件夹的访问权限，照下面设置：

- 根目录（和所有子文件夹）—读取
- 根目录—写入，修改（如果要安装第三方提供的程序集）
- DesktopModules 目录—写入，修改权限（允许文件管理器访问皮肤/容器）
- 如果你还想安装语言包，那么你必须对根目录下的所有子文件夹有写入和修改权限
- Configuring web.config for Installation

为安装配置 web.config 文件

所有的 ASP.NET 应用程序都需要 web.config 文件。在 3.1 之前的版本中，这个文件包含在

安装包中。在 3.1 和后续版本中这个文件不在包含在安装包中。如果你是安装 3.1 或后续版本，你必须首先把 release.config 改名为 web.config

在你浏览并初始化安装你的站点之前，你需要在 web.config 中进行一些设置。在 <appSettings> 设置组中，有 5 种设置来控制安装/升级

```
<appSettings>
  <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="true" />
  <add key="UseDnnConfig" value="true"/>
  <add key="InstallMemberRole" value="true" />
  <!-- Show missing translation keys (for development use) -->
  <add key="ShowMissingKeys" value="false" />
  <add key="EnableWebFarmSupport" value="false" />
  <add key="EnableCachePersistence" value="false" />
</appSettings>
```

第一个设置(“SiteSqlServer”)定义程序使用的数据库（例如：<add key="SiteSqlServer" value="Server=(local); Database=DotNetNuke; uid=;pwd=;" />）。使用你所用的数据库的相应的值来替换 Server,Database,uid 和 password 的值。如果是共享主机，主机提供者会提供这些值。在本机上安装，你一般不需要修改 Server 的值。然而，在某些情况下，你可能需要把 Server 的值改成你的本地计算机名。

第二个设置(“InstallTemplate”)定义一个安装模板。我们将在本文的后面详细描述这个模板。这里只是提示你一下，可以设置这块内容来控制安装选项。但默认模板适用于绝大多数情况。

第三个设置(“Auto Upgrade”)指示是否自动执行安装/升级过程。默认值为 true。不管怎么设置 DotNetNuke 都将检测当前的数据库和程序集版本，如果版本不同，则重定向到安装页面。如果这一项设置为 true，那么将自动执行安装或升级处理。如果设置为 false，安装页面将报告版本差异，然后停止。

第四个设置(“UseDnnConfig”)，默认值也是 true，它指示 DotNetNuke 是否从/Install 文件夹下的 dnn.config 文件中读取当前数据库版本。升级不能在每次 Application Start 的时候执行，使用这个文件能减少数据库压力，并提高性能。在安装/升级结束时，当前的版本信息会写入这个文件中。如果程序集的版本和这个文件记录的版本相同，那就什么也不做。如果不同，那么自动升级(AutoUpgrade)将确认数据库版本是否的确不同，并依此处理。

第五个设置(“InstallMemberRole”)随意安装 MemberRoles provider 脚本。由于这些脚本需要设置更高一级的 SQL Server 权限，可能一些主机提供者不允许运行这些脚本。如果他们不提供这种权限，那么，在你安装 DotNetNuke 这前，你要先要求主机提供者执行这些脚本。而且，如果和你其它使用 MemberRole provider 的程序（例如：Community Server）使用共享数据库，那么，如果其它程序已经安装了，你就不必再重新运行这些脚本了。

Data Provider 设置

有两个非常重要的关于数据访问的设置。下面是在 web.config 文件中定义 data provider 的部分

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

ObjectQualifier 属性用来控制数据库中每个对象名的前缀。默认是空字符串""，因此，象"Portals"表创建为"Portals"。举例来讲，如果把 objectQualifier 属性设置成"dnn_"，那么"Portals"表就会被创建成"dnn_Portals"，并且涉及 Portals 表的存储过程也会使用"dnn_Portals"。建议设置这个属性。因为，DotNetNuke 允许在同一环境中存在多个使用相同数据库的应用程序。

DatabaseOwner 属性用于指定数据库对象的“所有者”。默认情况下设置为“dbo”，并且大多数数据情况下应该保留这个值。然而，有一些 SQL Server 数据库（尤其是共享主机环境）可以不允许用户以 db_owner 权限认证联接字符串。在这种情况下，把 databaseOwner 属性置成你的登陆名。这个用户至少应该具有下列 SQL 权限：

- db_datareader(dotnetnuke 必须)
- db_datawriter(dotnetnuke 必须)
- db_ddladmin(MemberRoles 必须)
- db_securityadmin(MemberRoles 安装过程需要)

以上两个设置仅与默认的 MS SQL Server Data Provider 相关。如果你使用其它 data provider，你必须参照这个 provider 的文档来作相应的配置。

Navigation Provider 设置

在 DotNetNuke3.2 中，主要的导航系统为抽象为 Navigation Providers

```
<navigationControl defaultProvider="SolpartMenuNavigationProvider">
  <providers>
    <clear/>
    <add name="SolpartMenuNavigationProvider"
      type="DotNetNuke.NavigationControl.SolpartMenuNavigationProvider,
      DotNetNuke.SolpartMenuNavigationProvider"
      providerPath="~\Providers\NavigationProviders\SolpartMenuNavigationProvider\" />

    <add name="DNNMenuNavigationProvider"
      type="DotNetNuke.NavigationControl.DNNMenuNavigationProvider,
      DotNetNuke.DNNMenuNavigationProvider"
      providerPath="~\Providers\NavigationProviders\DNNMenuNavigationProvider\" />
  </providers>
</navigationControl>
```

```
<add name="DNNTreeNavigationProvider"
      type="DotNetNuke.NavigationControl.DNNTreeNavigationProvider,
      DotNetNuke.DNNTreeNavigationProvider"
      providerPath="~\Providers\NavigationProviders\DNNTreeNavigationProvider\" />

<add name="DNNDropDownNavigationProvider"
      type="DotNetNuke.NavigationControl.DNNDropDownNavigationProvider,
      DotNetNuke.DNNDropDownNavigationProvider"
      providerPath="~\Providers\NavigationProviders\DNNDropDownNavigationProvider\" />

</providers>
</navigationControl>
```

defaultProvider 属性默认设置为 SolpartMenuNavigationProvider，但如果你要使用其它的 provider，你只要把 defaultProvider 属性设置成你想使用的 provider 的名字就可以了。

成员角色原型设置

在即将发布（译者翻译时已经发布）的 ASP.NET2.0(称为“Whidbey”)中，提供一种新的成员/角色 provider 机制，在 web.config 中有一系列的设置用来配置这个 provider。在 DotNetNuke3.0 中包含这一机制的原型，但是 DotNetNuke3.0 仅支持修改四种参数：

```
<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"
            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 SQLServer
            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" />
  <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-控制密码的最小长度。Web.config 中的默认配置是 4，这是因为一直以来默认的超级用户使用 host/host 作为用户名和密码。要增加默认值，需要修改安装模板中位于<superuser>和<administrator>节点的的 passwords provider。

minRequiredNonalphanumericCharacters-表示在密码中至少要包含多少个非字母字符。默认值为 0，因为以前的 DotNetNuke 不要求非字母字符。同样的，把这个属性的值修改成任何正数时，要修改安装模板中的的<superuser>和<administrator>。

cookieName-在 provider 的 rolemanager 和 anonymouIndentification 节都可以设置 cookie 的名字。可能没必要设置，但是可以设置。

在 .NET2.0 下使用 Medium Trust 运行 3.2 版

虽然 DotNetNuke3.2 在 ASP.NET1.1 下编译，但是它完全兼容.NET2.0。如果你或你的主机提供者没有将站点配置为 Medium Trust，那么你不必作任何不同设置，就可以使 DotNetNuke2.0 的绝大部分内容运行于.NET2.0 这下。

如果 DotNetNuke3.2 以 Medium Trust 运行于.NET2.0，你需要在 web.config 的所有配置节中添加一个属性设置，如下所示。如果在.NET1.1 环境下运行，则不需要设置这个属性（实际上

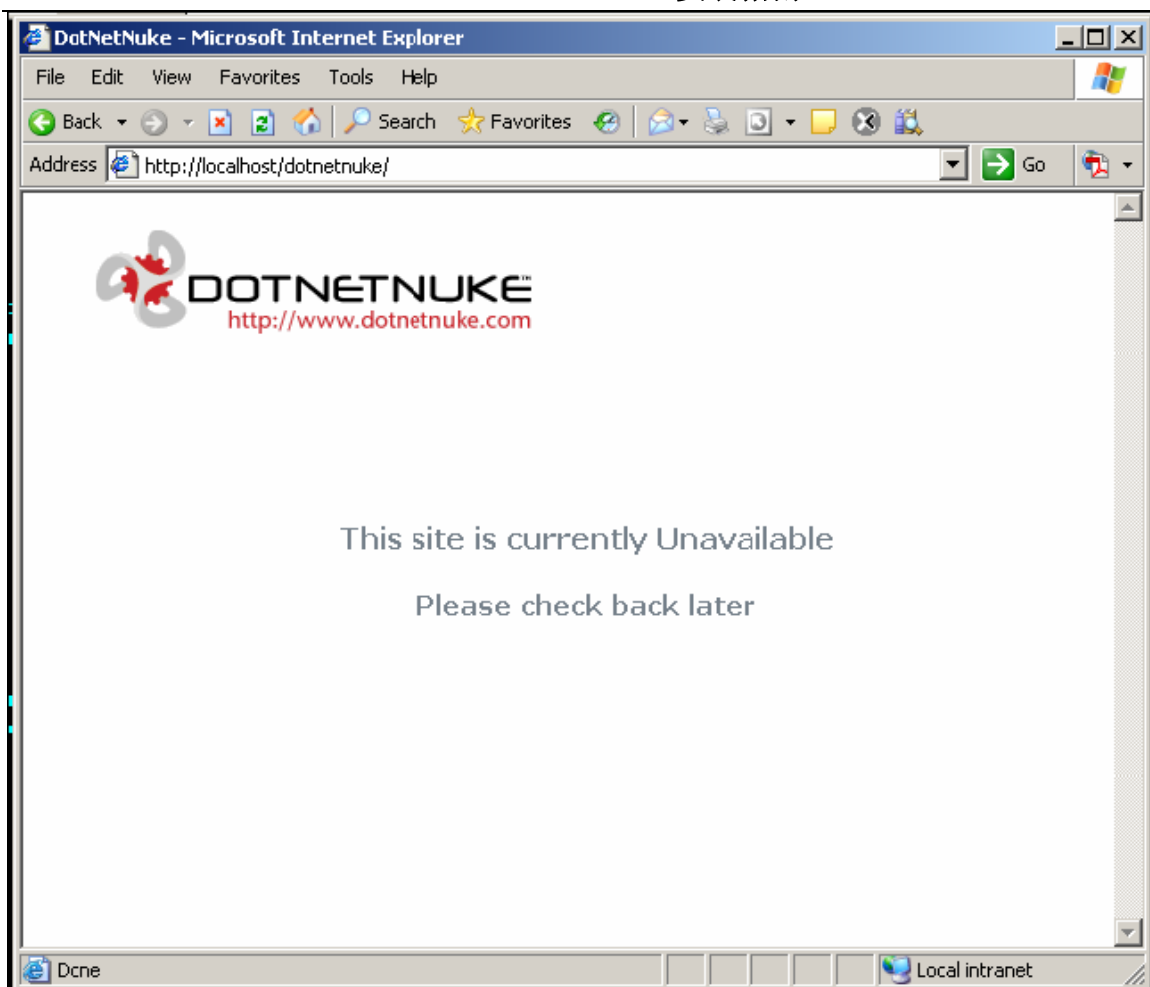
会造成错误)

```
<configSections>
  <sectionGroup name="dotnetnuke">
    <section name="data" requirePermission="false"
      type="DotNetNuke.Framework.Providers.ProviderConfigurationHandler,
      DotNetNuke" />
    <section name="logging" requirePermission="false"
      type="DotNetNuke.Framework.Providers.ProviderConfigurationHandler,
      DotNetNuke" />
    <section name="scheduling" requirePermission="false"
      type="DotNetNuke.Framework.Providers.ProviderConfigurationHandler,
      DotNetNuke" />
    <section name="htmlEditor" requirePermission="false"
      type="DotNetNuke.Framework.Providers.ProviderConfigurationHandler,
      DotNetNuke" />
    <section name="navigationControl" requirePermission="false"
      type="DotNetNuke.Framework.Providers.ProviderConfigurationHandler,
      DotNetNuke" />
    <section name="searchIndex" requirePermission="false"
      type="DotNetNuke.Framework.Providers.ProviderConfigurationHandler,
      DotNetNuke" />
    <section name="searchDataStore" requirePermission="false"
      type="DotNetNuke.Framework.Providers.ProviderConfigurationHandler,
      DotNetNuke" />
    <section name="friendlyUrl" requirePermission="false"
      type="DotNetNuke.Framework.Providers.ProviderConfigurationHandler,
      DotNetNuke" />
    <section name="caching" requirePermission="false"
      type="DotNetNuke.Framework.Providers.ProviderConfigurationHandler,
      DotNetNuke" />
    <section name="authentication" requirePermission="false"
      type="DotNetNuke.Framework.Providers.ProviderConfigurationHandler,
      DotNetNuke" />
  </sectionGroup>
  <sectionGroup name="memberrolesprototype">
    <section name="membership" requirePermission="false"
      type="Microsoft.ScalableHosting.Configuration.MembershipConfigHandler,
      MemberRole,Version=1.0.0.0, Culture=neutral, PublicKeyToken=b7c773fb104e7562" />
    <section name="roleManager" requirePermission="false"
      type="Microsoft.ScalableHosting.Configuration.RolesConfigHandler,
      MemberRole,Version=1.0.0.0, Culture=neutral, PublicKeyToken=b7c773fb104e7562" />
    <section name="profile" requirePermission="false"
      type="Microsoft.ScalableHosting.Configuration.ProfileConfigHandler,
      MemberRole,Version=1.0.0.0, Culture=neutral, PublicKeyToken=b7c773fb104e7562" />
    <section name="anonymousIdentification" requirePermission="false"
      type="Microsoft.ScalableHosting.Configuration.AnonymousIdConfigHandler,
      MemberRole,Version=1.0.0.0, Culture=neutral, PublicKeyToken=b7c773fb104e7562" />
  </sectionGroup>
</configSections>
```

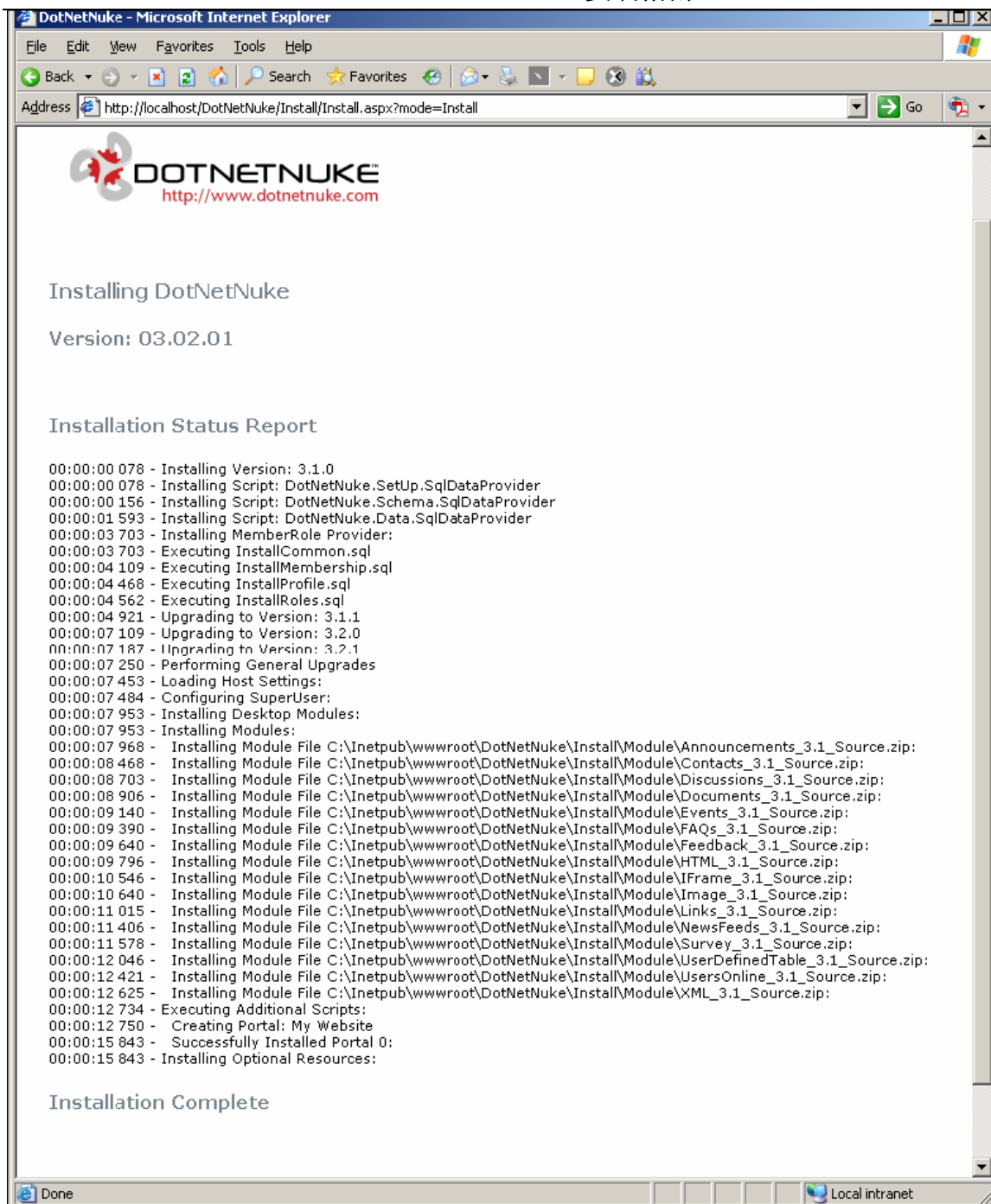
安装 DotNetNuke

至此，我们已经完成web.config文件和安装模板的配置，准备安装DotNetNuke3.x了。安装过程非常简单直接。使用浏览器浏览文件夹根目录-本地安装是目录可能是<http://localhost/DotNetNuke>，远程安装的话就是你的域名www.mydomain.com

自动升级功能将检测数据库是否为空，如果“AutoUpgrade”被设置成 false，那么一个“Under Construction”将提示站点不可用。然后管理员可以使用 url “Install/Install.aspx?mode=Install”来激活安装



如果“**AutoUpgrade**”设置为 `true`，那么安装过程将自动执行。无论使用哪种形式激活安装，都会显示下面这样的安装回馈：



我们看一下默认安装发生了什么

- ✧ 安装程序报告正被安装的模板的版本，同时一个所需的额外脚本修正一些约束的默认名字。
- ✧ 安装程序报告已被执行的安装脚本。（被执行的用于新版本的新增脚本）
- ✧ 安装报告已执行 **MemberRole Provider** 所需的脚本，
- ✧ 然后解析安装模板的部分
 - ◆ 主机设置
 - ◆ 超级用户
- ✧ 然后模块被安装
- ✧ **Portal** 被创建
- ✧ 可选资源被安装，例如：附加皮肤，容器

升级到 DotNetNuke3. x

备份你的站点

在升级之前，如果可以的话，一定要备份整个站点。至少也要备份一下 web.config

准备升级

在你备份好你的 web.config 文件之后，你就可以使用新版本的 DotNetNuke 覆盖你现有的版本。任何在当前安装中不再需要的文件，都将被升级过程删除。

请参考上一章回顾一下可用的包。

配置升级用的 web.config

在安装 DotNetNuke3.x 一章中我们提到过，zip 安装包中不再包含 web.config 文件的拷备。在这一节的升级现有 DotNetNuke 站点（尤其是 3.0.13 或后续版本）中，你会明白这么做的原因

如上所述，DotNetNuke 版本 3 添加了成员角色（memberRole）原型。在 AppSettings 中有三个键为这一组件控制密码加密

```
<add key="MachineValidationKey" value="F9D1A2D3E1D3E2F7B3D9F90FF3965ABDAC304902" />  
<add key="MachineDecryptionKey" value="F9D1A2D3E1D3E2F7B3D9F90FF3965ABDAC304902F8D923AC" />  
<add key="MachineValidationMode" value="SHA1" />
```

这些键可能看上去无关紧要，但是如果你的 MachineValidationKey 和 MachineDecryptionKey 的值和上面不同，那你为升级设置 web.config 文件时可要小心了。

为使你的升级成功，你必须按照下列主要过程

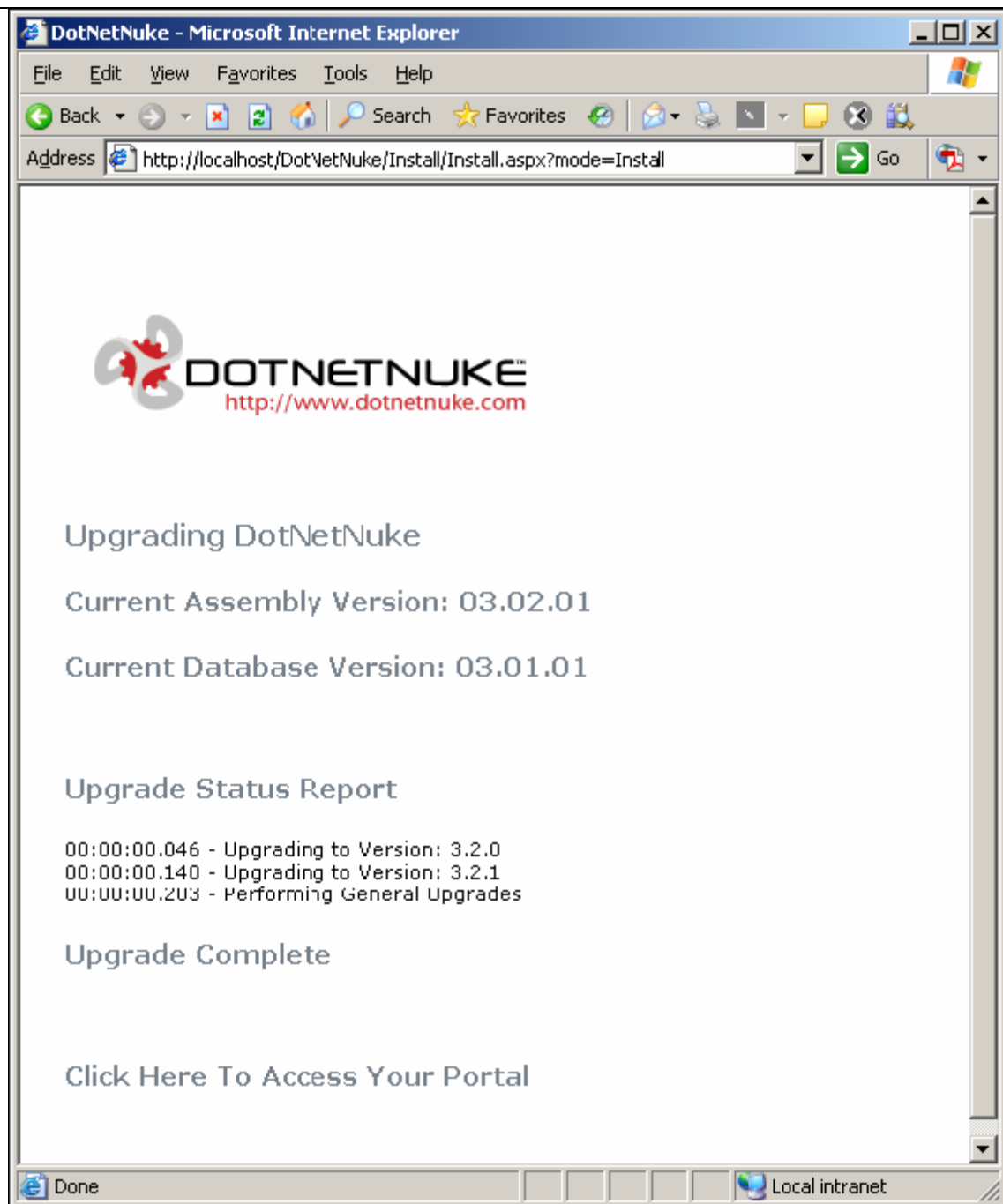
- ✧ 备份你现有的 web.config 文件（例如：web.backup.resources）。对于备份的重要性，我们已经强调的不能再强调了。如果你丢失了你的 MachineKeys 的初始，那么你所有的用户将都不能登陆。
- ✧ 在确认你已经安全的备份了你的 web.config 文件之后，将 release.config 重命名为 web.config
- ✧ 把新的 web.config 中的下列键值替换成你的备份文件中的值
 - SiteSqlServer
 - MachineValidationKey
 - MachineDecryptionKey
 - InstallationDate(可能 web.config 文件中没有，如果没有就从 web.backup.resources 加到 web.config)
- ✧ 修改 web.config 中的其它内容，使之支持附加的 providers 等。

当然，你也可以升级你前一版本的 web.config(前面备份的)，把 release.config 文件中的新的配置项移到你的 web.config 中。不同版本的 DotNetNuke 引入的新 provider 节是不同的，同时依赖你要升级的版本，你要把下列配置项从 release.config 移动到 web.config：

	引入版本	从下列版本（或更早版本）升级
Navigation Provider 节	3.2.0	3.1.1
Caching Provider Section 节	3.1.0	3.0.13
DBLoggingProvider 节	3.1.0	3.0.13

现在你可以浏览你的站点激活更新过程。在你确认在新的 web.config 文件中的 MachineKeys 和你备份旧文件中的相同之前，千万不要浏览你的站点。

当你浏览你的站点时，你会得到如下所示的更新报告：



我们看看升级的过程

- ✧ 安装程序报告将被安装的新的 DotNetNuke 版本（这里是：3.2.1）
- ✧ 安装程序报告当前的版本（这里是：3.1.1）
- ✧ 安装程序报告一系列被执行的增加脚本
- ✧ 安装程序报告主升级正被执行。
- ✧ 在这例子里没有的是，如果安装文件夹中包括任何可选资源（模块，皮肤，容器），它们将在升级的后期被安装。

安装 DotNetNuke4.x

概述

DotNetNuke4.0 是 DotNetNuke 第一个基于 ASP.NET2.0 的版本。虽然 3.2 版本可以运行于 ASP.NET2，但是开发需要 Visual Studio 2003 和 .NET1.1。

如同在许多 ASP.NET 相关的社区中讨论的，.NET2.0（和 Visual Studio 2005）为开发和运行网站引入了一个不同的范例。我们的主要问题在于，在 .NET2.0/Visual Studio2005 中，地址不再是一个 web 工程。.NET2 把一个网站的虚拟目录下的所有文件作为网站的一部分。

另外，Visual Studio 2005(VS2005)提供了更先进的工程和项目模板系统，在 Express 产品系列中也支持，也就是 Visual Web Developer Express(VWD)。这一特征允许我们开发一个 DotNetNuke Starter Kit，它能简化 DotNetNuke 的站点的创建。

本章主要描述这一版本在安装过程的不同点。如果你需要某个主题信息没有提到，你可以回顾一下上一章“安装 DotNetNuke3.x”的内容

使用哪个安装包？

DotNetNuke4 有四个包

- ✧ DotNetNuke_X.Y.Z_Install.zip—安装包—它包含了 ascx,aspx 等文件，也包含了已编译的程序集（dll 文件），但不包含 provider 和 httpModules 的 VisualBasic 代码文件。
- ✧ DotNetNuke_X.Y.Z_Source.zip—源码包—它包含了所有的 DotNetNuke 工程的所有核心文件和 Visual Basic 源码文件。
- ✧ DotNetNuke_X.Y.Z_StarterKit.vsi—Visual Studio 2005 安装包—它包含一个工程模板（基于安装包），一个模块项目模板（包括 Visual Basic.NET 的 C#），和一个皮肤项目模板。支持类库，providers 和 HttpModules 以预编译形式提供（dll）

要决定使用哪个包，你应该首先确定你的开发环境。因为 Visual Web Developer Express(VWD)不支持除网站工程这外的其它工程类型。因此，如果你希望在 VWD 而不是 Visual

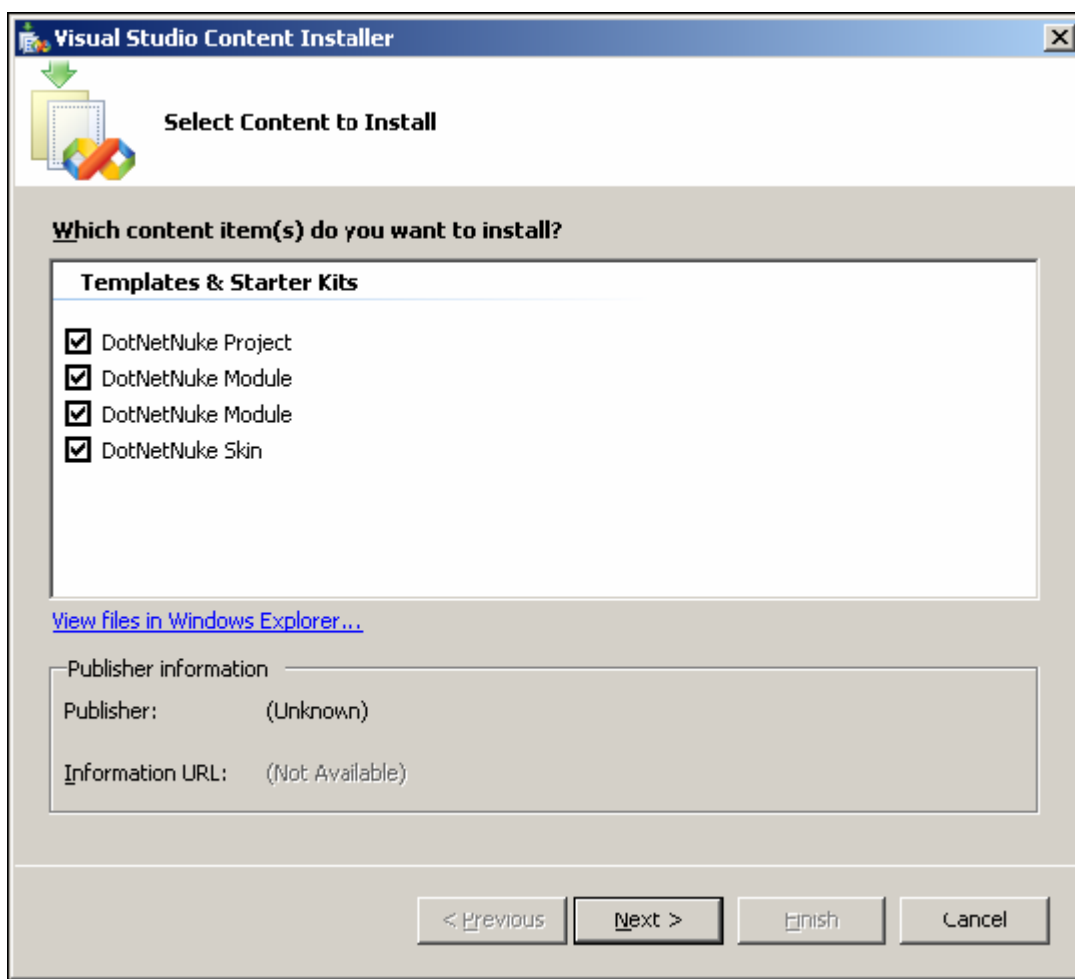
Studio2005(VS2005)中工作，那么你将不能使用源码包。

使用 Starter Kit

1. 安装 Starter Kit

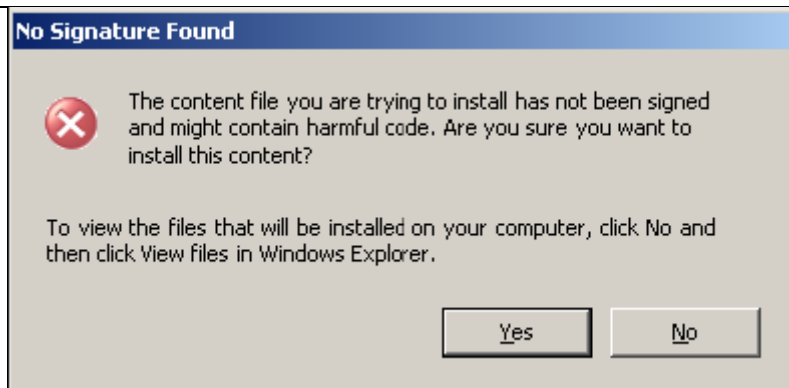
要想使用 Starter Kit，首先要安装它。如果你已经安装了 VS2005 或 VWD，你才能安装它。

要安装 Starter Kit, 在 Windows 资源管理器中双击这个文件。Visual Studio 内容安装将启动，如下图：

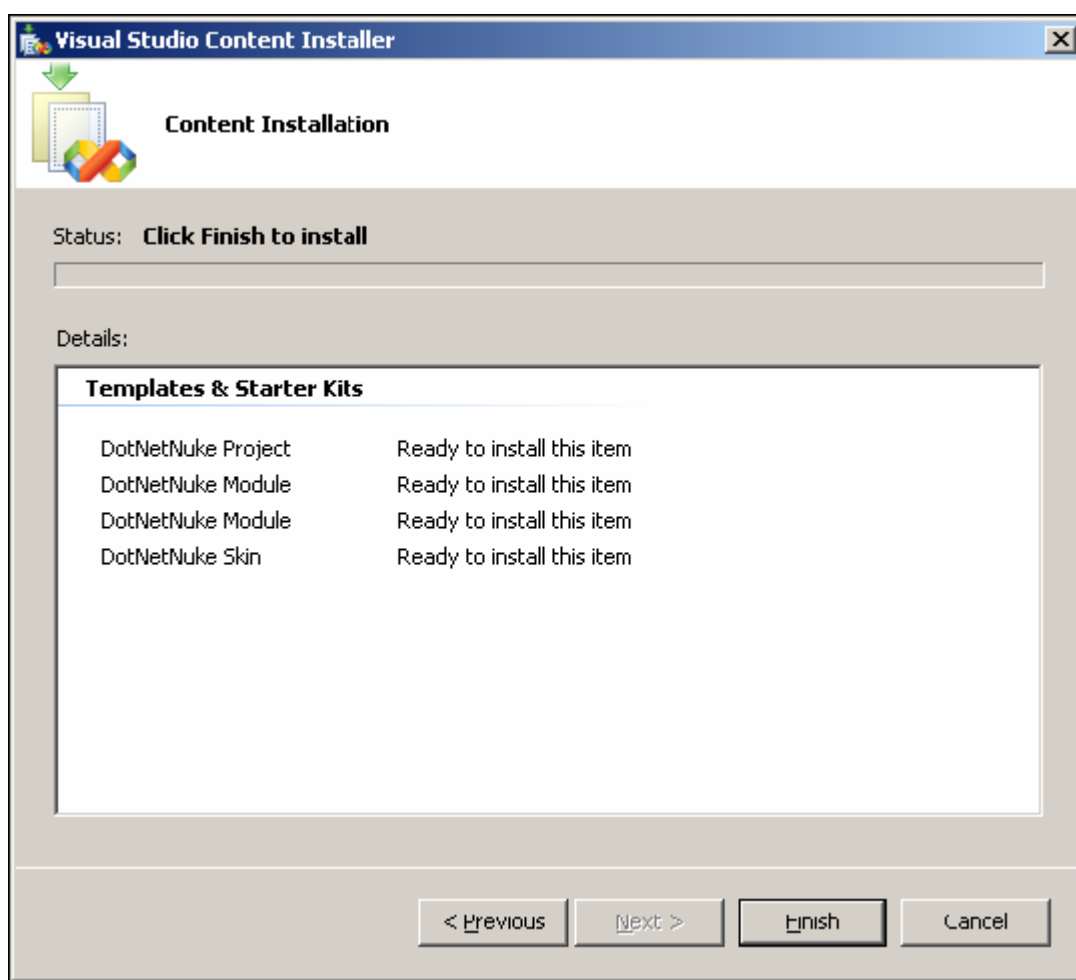


选择你要安装的模板—默认全部选择—并且点击 Next—有两个 DotNetNuke 模板（一个是 VB 一个是 C#）

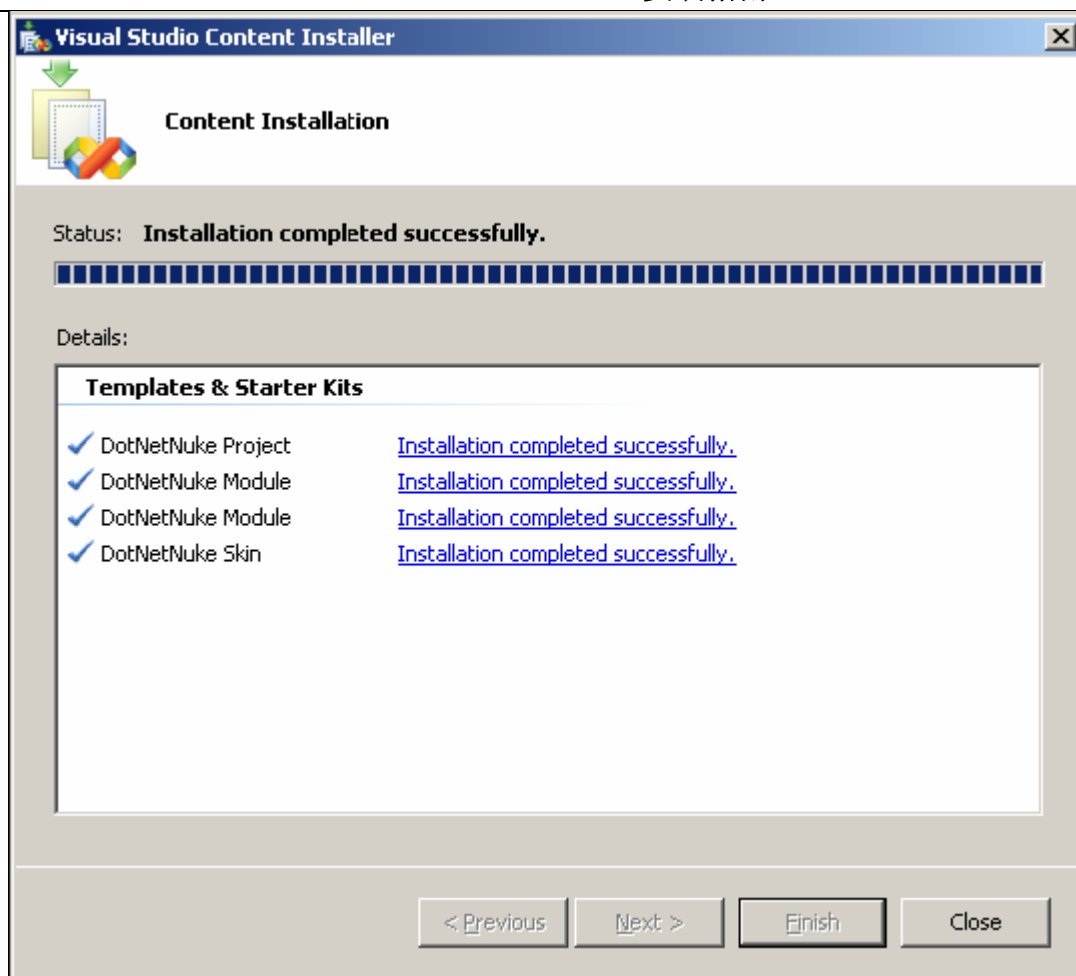
由于这个安装包没有注册数据许可证，所以你将看到下面的警告：



选择 Yes 到下一屏幕

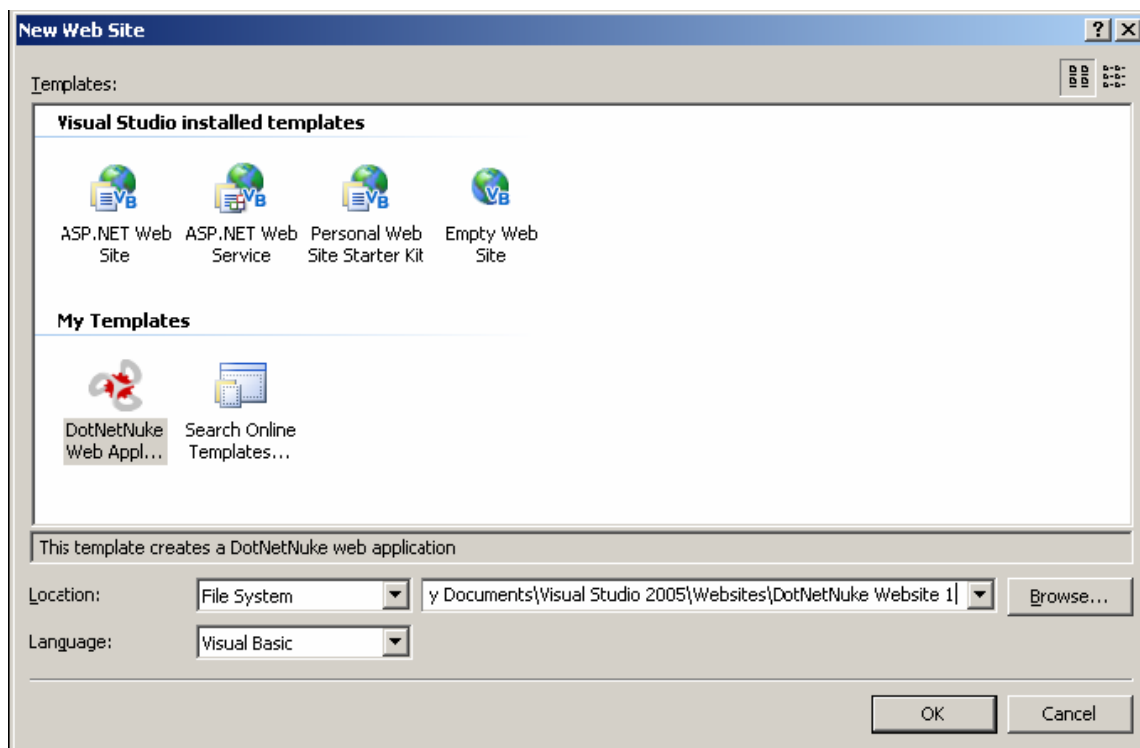


最后点击完成安装模板



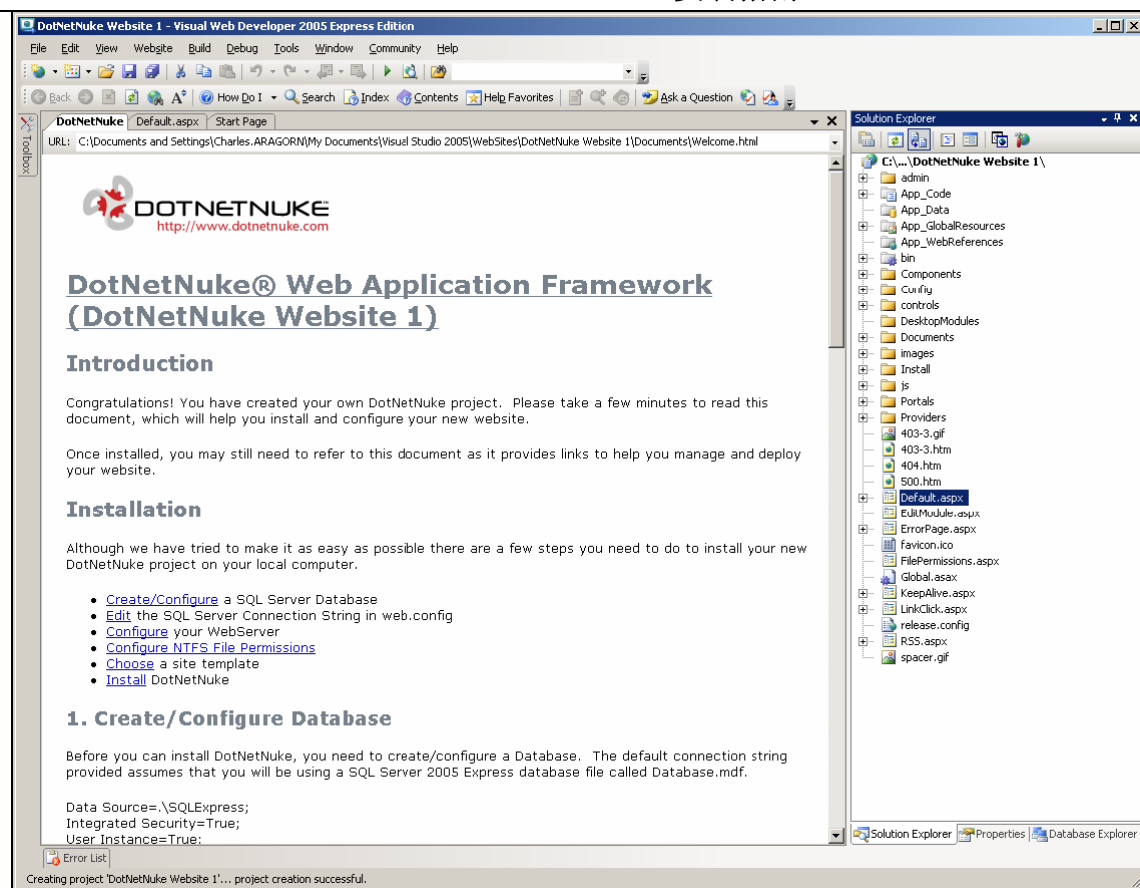
2. 从工程模板创建一个 DotNetNuke 站点

一旦你安装了模板，你就可以创建一个新的 DotNetNuke 工程了。在 VS2005 或 VWD 中从文件菜单中选择 New Website...



选择 DotNetNuke 模板，在 Location 中选择 File System 或 HTTP，然后在文本框中输入名字。上面的例子中，我们选择 File System 选项，这样将使用 VS2005/VWD 在基于 WebServer 的文件系统中创建，命名站点为“DotNetNuke Website 1”

点击 OK 后，将在你选择的文件夹创建一个 web 站点，并且在内部浏览器中自动打开“welcome.html”文件



Welcome.html 介绍如何在你机器上完成安装

使用或安装源码包

1. 解压包

安装 DotNetNuke3.x 时，无论你使用哪个包（安装包或源码包），你都要把它解压到应用程序运行目录。

然而，如果你决定使用源码包，我们看到的所有类库支持工程文件夹结构和我们以前的认识略有不同。

如果你现在只想用安装包，你只需要照 3.x 版本包的方法。

如果你想现在只使用安装包，将来再使用源码包，请按下面的过程操作

❖ 创建一个应用程序运行的目录（例如： C:\DotNetNuke_2）

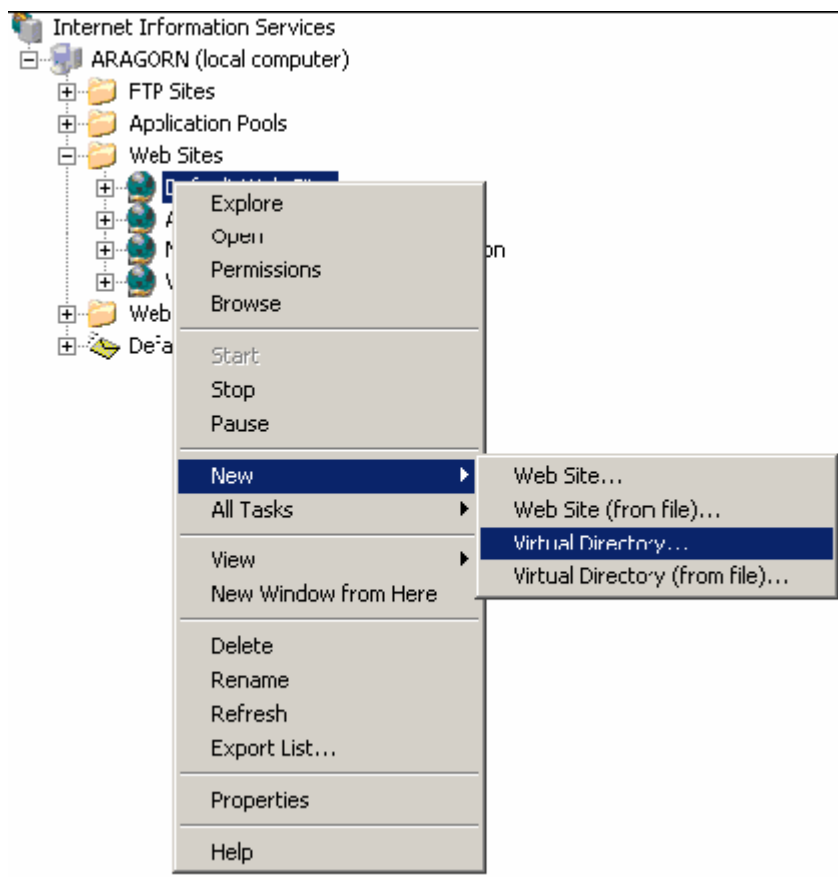
- ✧ 如果你使用安装包
- 创建一个名为 Website 的子目录
 - 把安装包解压解压到你刚创建的子目录。

如果使用源码包，那么把源码包解压到创建的第一个文件夹（C:\DotNetNuke_2），子文件夹将自动创建。

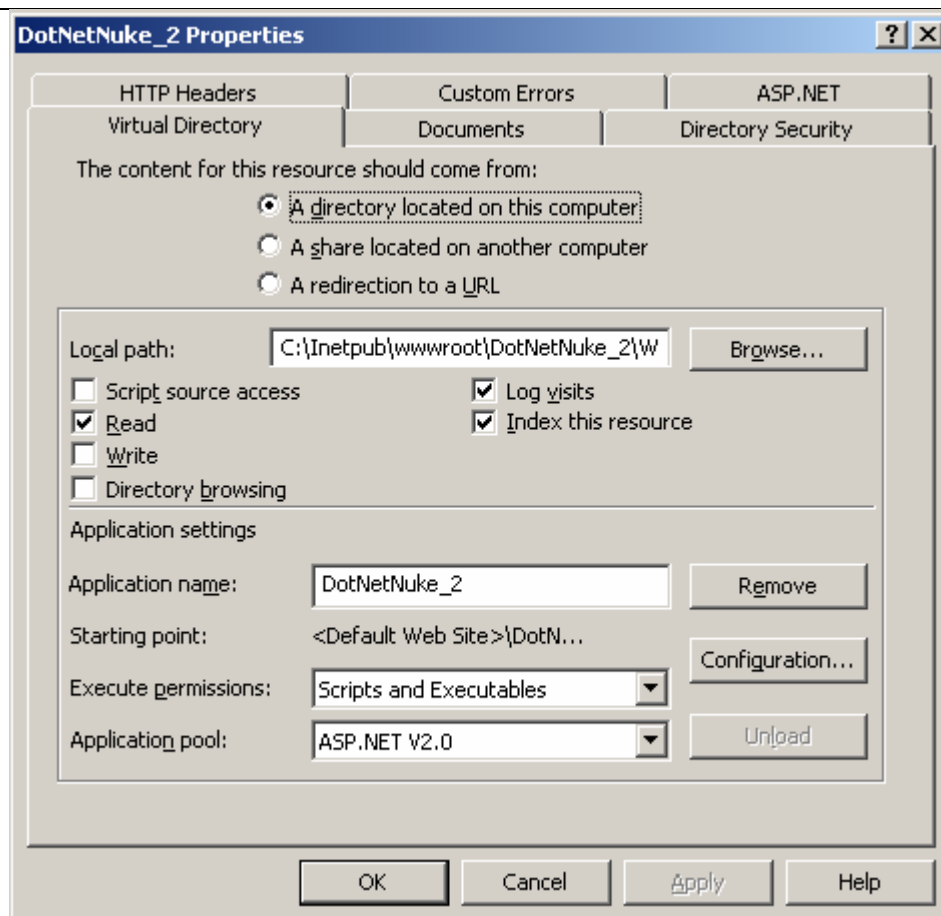
2. 配置 IIS（Internet 信息服务器）

在本地 intranet 配置，在 IIS 中创建一个名为 DotNetNuke_2r 虚拟目录，指向上面提到的相应的物理目录

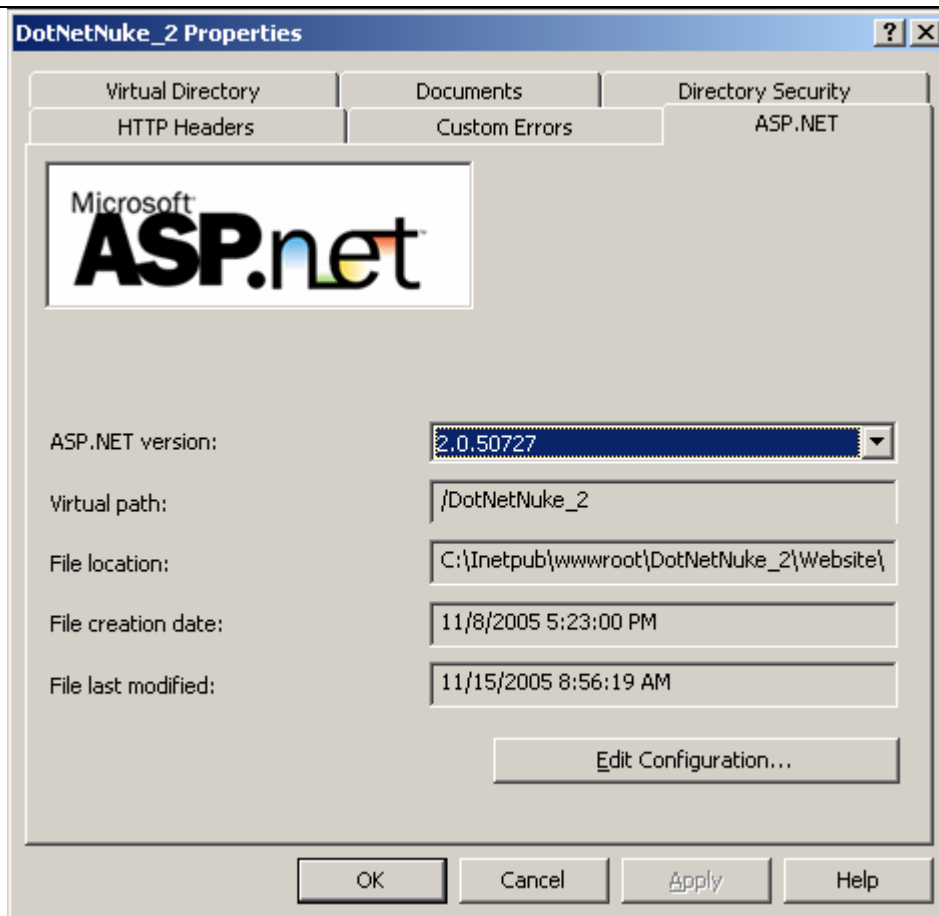
这个目录具体是哪个目录根据上面提到的过程而定。如果你使用安装包，就把 IIS 目录指向你解压安装包的位置。如果你使用源码包，把 IIS 目录指向\Website 子目录。



接着将启动一个向导来指导你配置你的虚拟目录。在你使用向导创建虚拟目录这后，你可以在虚拟目录上右击选择属性，然后对虚拟目录进行编辑。



对于 DotNetNuke4.x 而言，你必须把虚拟目录配置成.NET2.0



配置 web.config

虽然在 DotNetNuke4.x 中使用的 web.config 和 DotNetNuke3.x 中的略有不同，但是这些不同大部分不由于 .NET2 造成的，而不是 DotNetNuke4.x 造成的。

因此，这里我们只专注于 3.x 和 4.x 版本这间的差别内容。对于 DotNetNuke 中特殊的 web.config 设置，请参考“安装 DotNetNuke 3.x 版本”一章。

要 3.x 中，要做的第一件事就是，把 **release.config** 重命名为 **web.config**（如果你需要在 Medium Trust 方式下运行，那么重命名 **development.config** 而不是 **release.config**）

在 3.x 和 4.x 的 web.config 文件中，主要要三点不同

1. 除了要有 3.x 版本在<appSettings>中的 connectionString 设置这外，在 DotNetNuke4.x 中还需要在<connectionString>节设置相同的连接字符串。

```
<connectionStrings>
  <!-- Connection String for SQL Server 2005 Express -->
  <add
    name="SiteSqlServer"
    connectionString="Data Source=.\SQLEXPRESS; Integrated Security=True;
User Instance=True; AttachDBFilename=|DataDirectory|Database.mdf;"
    providerName="System.Data.SqlClient" />
</connectionStrings>
```

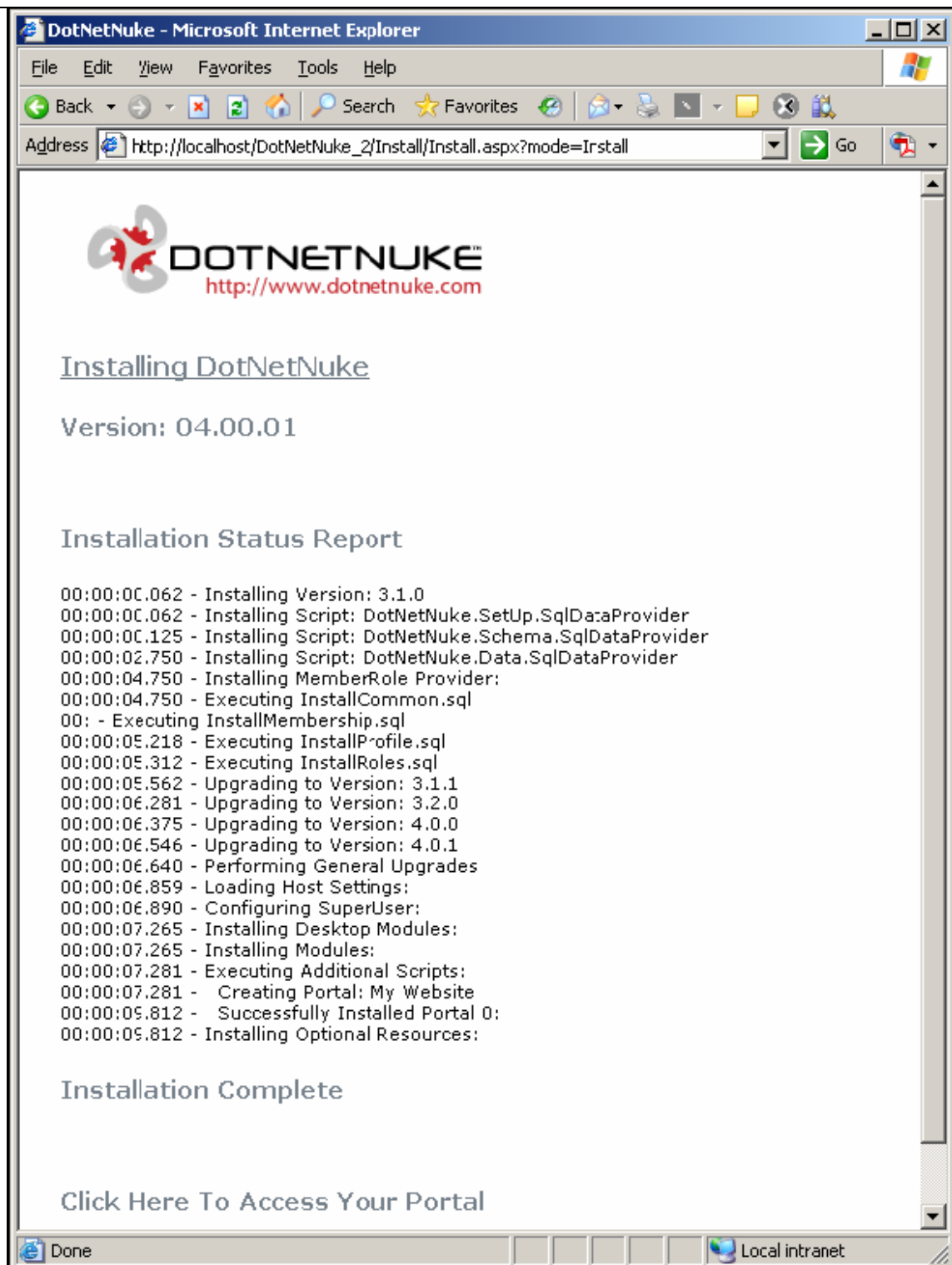
注意，connectionString 的名字“SiteSqlServer”，和在<appSettings>中的键中使用的名字相同，并且<appSettings>中的设置也要支持 legacy 模式

2. 以前在<appSettings>配置节中的 machine key 设置 — MachineValidationKey，MachineDecryptionKey 和 MachineValidationMode — 现在位于<system.web>中的<machineKey>中。
3. 成员权限设置本身没有变化，但是它的位置从原来的<memberRolesPrototype>移动到<system.web>节点。如果想更多修改这一设置，请阅读“安装 DotNetNuke3.x”一章中的详细介绍。

```
<machineKeyvalidationKey="F9D1A2D3E1D3E2F7B3D9F90FF3965ABDAC304902"
decryptionKey="F9D1A2D3E1D3E2F7B3D9F90FF3965ABDAC304902F8D923AC"
decryption="3DES"
validation="SHA1"/>
```

安装 DotNetNuke

我们已经配置了 web.config 文件和安装模板文件，现在准备安装 DotNetNuke4.x。DotNetNuke4.x 的安装过程和 3.x 版本的过程相同。浏览你配置的站点的虚拟目录，安装就会执行，如下所示（注意，在下面的示例中，因为在 Install/Module 文件夹中没有模块的 zip 文件，所以没有自动安装模块）



升级到 DotNetNuke4.x

备份你的站点

在升级之前，如果可以，最好备份整个站点。至少备份 web.config 文件

准备升级

在你读过上一章之后，你会知道，工作于源码版本时，DotNetNuke4.x 引入了新的文件夹结构。这样的结果是，30 个类库工程不能在网站的主文件夹和其子文件夹下运行。

这种引入给那些想从 3.x 版本的源码开发安装升级到 4.x 的源码开发安装的人带来了一定的麻烦。

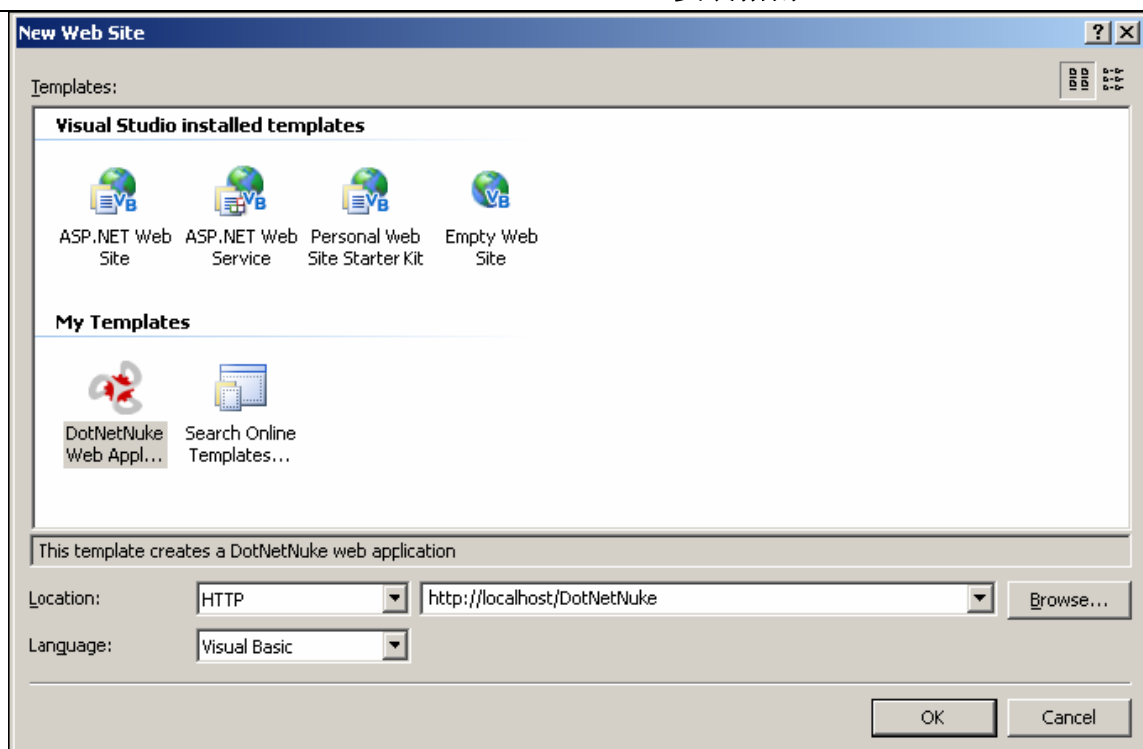
如果你从 3.x 的源码开发安装升级到 4.x 的安装，或从 3.x 的安装版本升级到 4.x 安装版本，你只需要照常规升级过程操作就可以了。

把 3. x 安装版升级到 4. x 安装版。

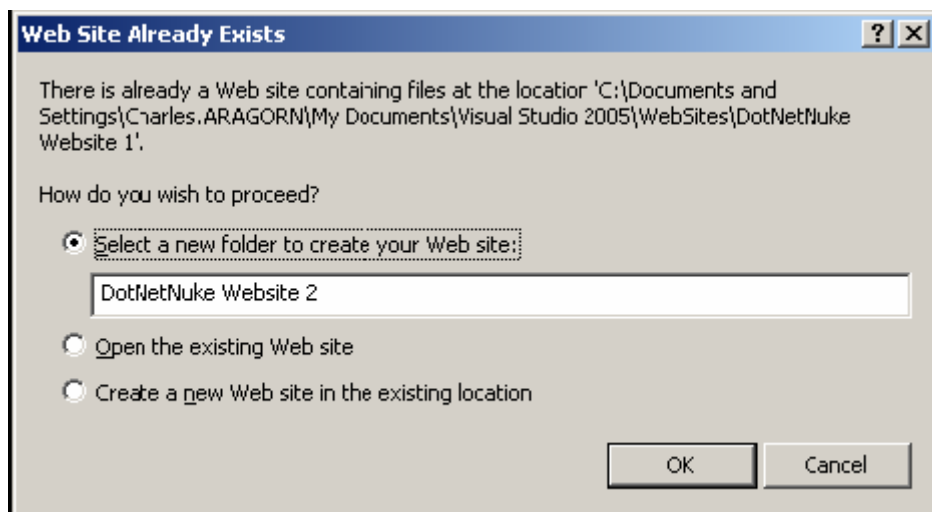
如果你不在乎类库源码工程，你可以升级到 4.x 的安装包。

一旦你备份了你的 web.config 文件，你有两种方式覆盖你现有的 3.x 版本，执行安装。

1. 解压 Install.zip 包覆盖你现有的版本。
在 VS2005 或 VWD 中选择“New Website...”依照上一章描述的过程操作，中间要选择你现存的站点的位置。



由于站点已经存在，新站点向导（New Web Site Wizard）将提示你当前位置已存在站点，并提供三个选项：



这里我们选择最后一个选项：Create a new Web site in the existing location。

完成这些工作作之后，你就可以准备为 4.x 版本配置你的 web.config 文件了。

把 3. x 版本升级到 4.0 源码版

如果你想升级到 4.0 和源码版本，情况就会有一点棘手。

因此，我们建议，与其用某种方式覆盖你的“旧”站点，不如按照上一章介绍创建一个新的 DotNetNuke4.x 站点，然后把你旧站点的自定义内容移动到这个新站点，这此内容包括：模块，皮肤，和 Portal 内容，包括 web.config 文件。

然后在 IIS 中把虚拟目录的改为指向 DotNetNuke4.x 版本站点。

这是将你的开发从 3.x 转移到 4.x 的最安全和最简单的方式，完成之后，就可以配置你的 web.config 文件了。

配置升级到 4.x 版本的 web.config

升级你的 web.config 文件最容易的方法是，把你修改过的设置拷贝到 release.config 文件—依照下面的过程：

- ✧ 备份原来的 web.config 文件（例如：web.backup.resources）。备份的重要性已经强调的不能在强调了。如果你丢失了原始听 MachineKeys，那么，你所有的用户将不能登陆。
- ✧ 在你确认安全的备份了你的 web.config 文件之后，把 release.config 改名为 web.config
- ✧ 使用备份中的内容替换新的 web.config 文件中的下列键：

SiteSqlServer

InstallationData（release.config 中没有这个键，所以我们要把 web.backup.resource 中的键值加到新的 web.config 中）

- ✧ 把旧的 web.config 中的的<connectionStrings>内容拷贝过来。

```
<add
name="SiteSqlServer"
connectionString="???????"
providerName="System.Data.SqlClient" />
```

- ✧ 设置以前在 appSettings 节点的的 machineKey 元素的值：

```
<machineKeyvalidationKey="F9D1A2D3E1D3E2F7B3D9F90FF3965ABDAC304902"
decryptionKey="F9D1A2D3E1D3E2F7B3D9F90FF3965ABDAC304902F8D923AC"
decryption="3DES"
validation="SHA1" />
```

“validationKey”属性值在 MachineValidationKey 设置中，decryptionKey 属性的值应该在 MachineDecrypitonKey 设置中。

- ✧ 继续配置 web.config，以支持其它的 provider

升级 DotNetNuke

现在你可以浏览你的站点，激活升级过程。在你确认新的 web.config 和你备份的的文件的 MachineKeys 的值相同之前，一定不要浏览你的站点。



在你浏览站点后，将出现一个类似“升级到 3.x 版”一章的简单升级报告。

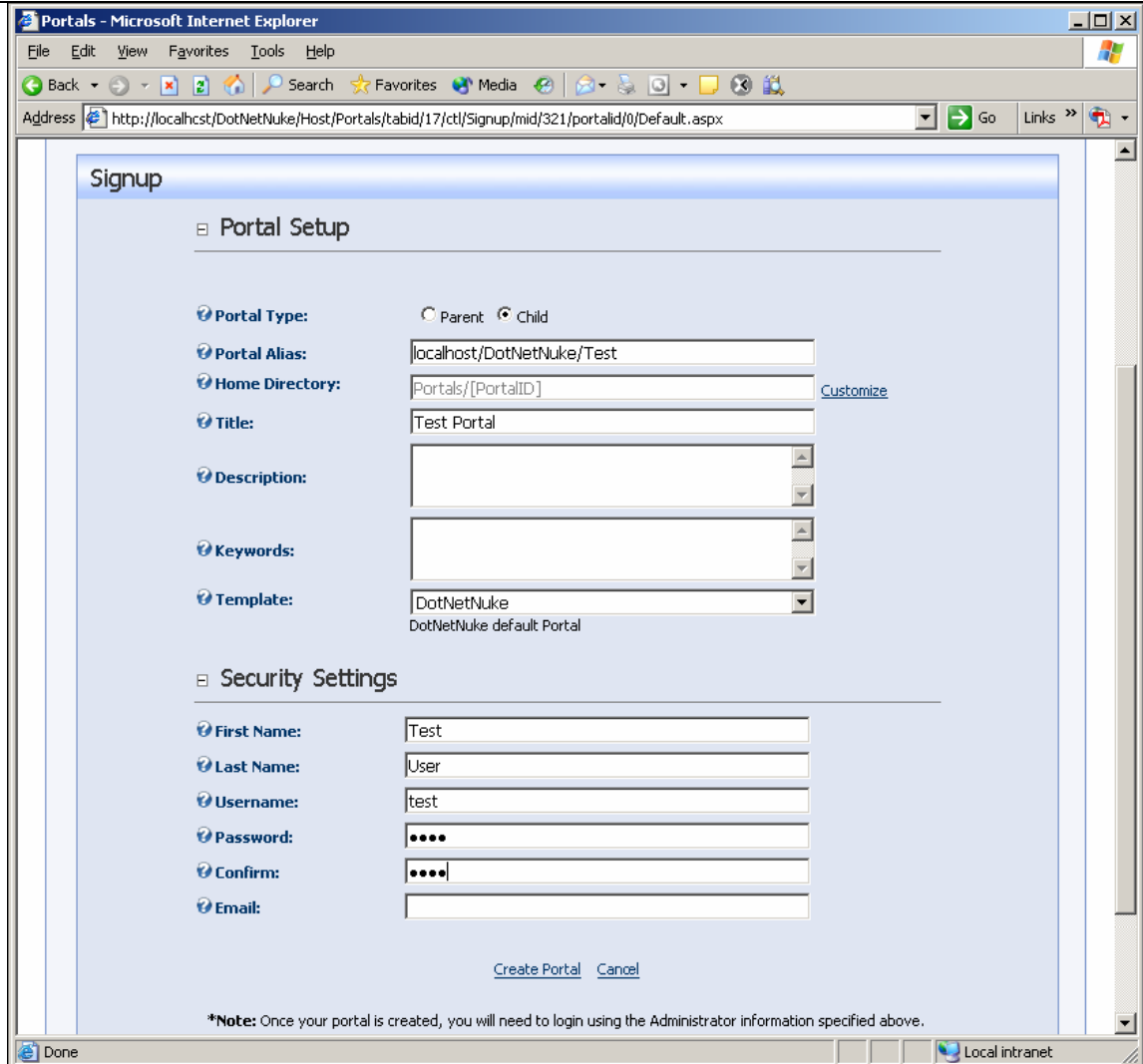
安装可选资源和模块

象语言包，皮肤，保护性程序集这样的可选资源，也可以在在安装过程中被自动安装。在 /Install 文件夹下面有六个文件夹：**Container**, **Language**, **Module**, **Portal**, **Skin** 和 **Template**。不同的资源的 zip 包（后面将详细描述）被放置在相应的文件夹，安装程序将自动安装它们。

从 3.1 版本以后，核心 Desktop 模板默认被放在 /Install/Module 文件夹中，因此它们会被自动安装。

安装附加 Portals

在 DotNetNuke 被安装后，有两种方法添加附加的 portal。第一种，也是最显而易见的方法是，以 Host 登陆，从 Host 菜单中选择 Portals，然后输入这个 Portal 的相关信息。



第二种方法可以通过一次操作，安装多个 portal

这种方法就是在 DotNetNuke 安装过程中，模仿 portals 的安装。有一个模板文件：portals.resource 可以被放在/Install/Portal 文件夹下。除了这个文件仅被处理<portals>和<portal>节点之外，它和安装模板很类似。

Portals.resource 文件示例

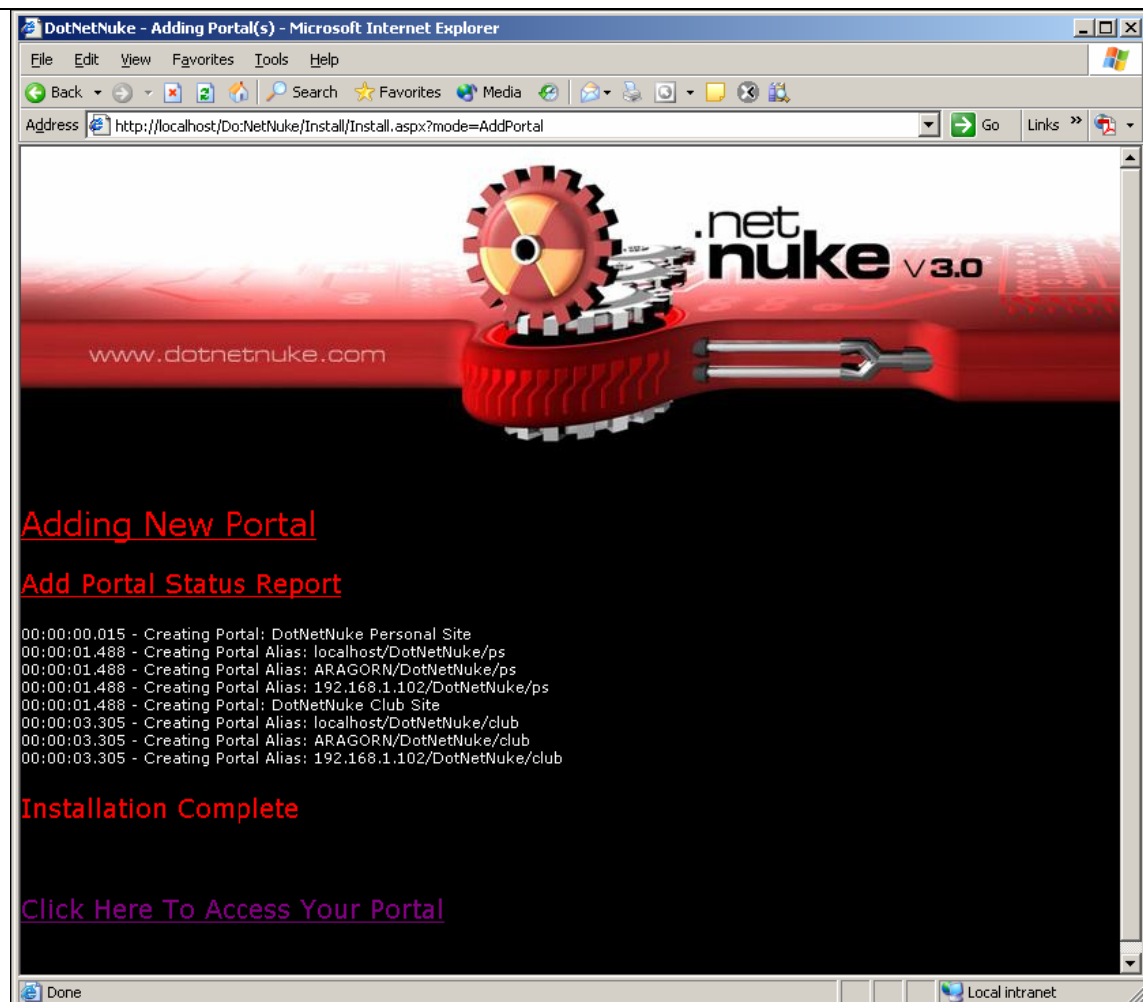
```
<dotnetnuke>
  <portals>
    <portal>
      <portalname>DotNetNuke Personal Site</portalname>
      <administrator>
        <firstname>Administrator</firstname>
        <lastname>Account</lastname>
        <username>admin</username>
        <password>admin</password>
        <email></email>
      </administrator>
      <description>Default Personal Site</description>
    </portal>
  </portals>
</dotnetnuke>
```

```
<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>
```

当这个文件被保存到 Install/Portal 文件夹之后，文件中定义的附加 portals 的安装有两种方法启动。

1. 使用“Install Resources”这个 scheduled 事件。
2. 使用下面的 Url 手动浏览安装页面 “Install/Install.aspx?mode=AddPortal”

这时，安装页面将提示回馈信息，如下所示：



安装附加资源

同样有两种方式安装附加资源，象皮肤，模块（打包成保护性程序集）和语言包。

通过程序中相应的 Admin/Host 菜单。每一种资源都可以通过其相应的用户界面元素来安装。比如如模块就可以通过 Host/Module Definitions 菜单来安装。

把安装包放到/Install 目录下的相应文件夹中。在/Install 目录下和一系列的子文件夹。把 zip 文件放到相应的文件夹下面，通过浏览 “**Install/Install.aspx?mode=InstallResources**” 来安装。也可以通过资源安装 Scheduled 事件安装。如果使用安装页面安装，会出现安装状态简单报告：



在被安装这前，资源需要被打包成 zip 文件。本文主要是描述安装，因此不描述如何打包资源。我们将在其它文档中描述。

下面是各种资源的安装方法。

模块

- ✧ 通过程序 Host/Module Definitions 用户界面，从 Action 菜单或控件的链接按钮中，选择“Upload New Module”
- ✧ 通过安装程序把 zip 文件放到/Install/Module
- ✧ 模块的开发和打包在“DotNetNuke Module 开发指南中”叙述

皮肤/容器

- ✧ 在 Host 或 Admin 菜单中的 Skins 菜单项，打开程序界面，选择“Upload Container”或“Upload Skin”
- ✧ 通过安装程序把 zip 文件放入/Install/Skin 或/Install/Container 目录
- ✧ 皮肤/容器的开发和打包的说明在“DotNetNuke 皮肤”文档中

语言包

- ✧ 通过 Host/Languages 打开程序界面，选择“Upload Language Pack”

✧ 使用安装程序把 zip 文件放入/Install/Language 目录

出错情况

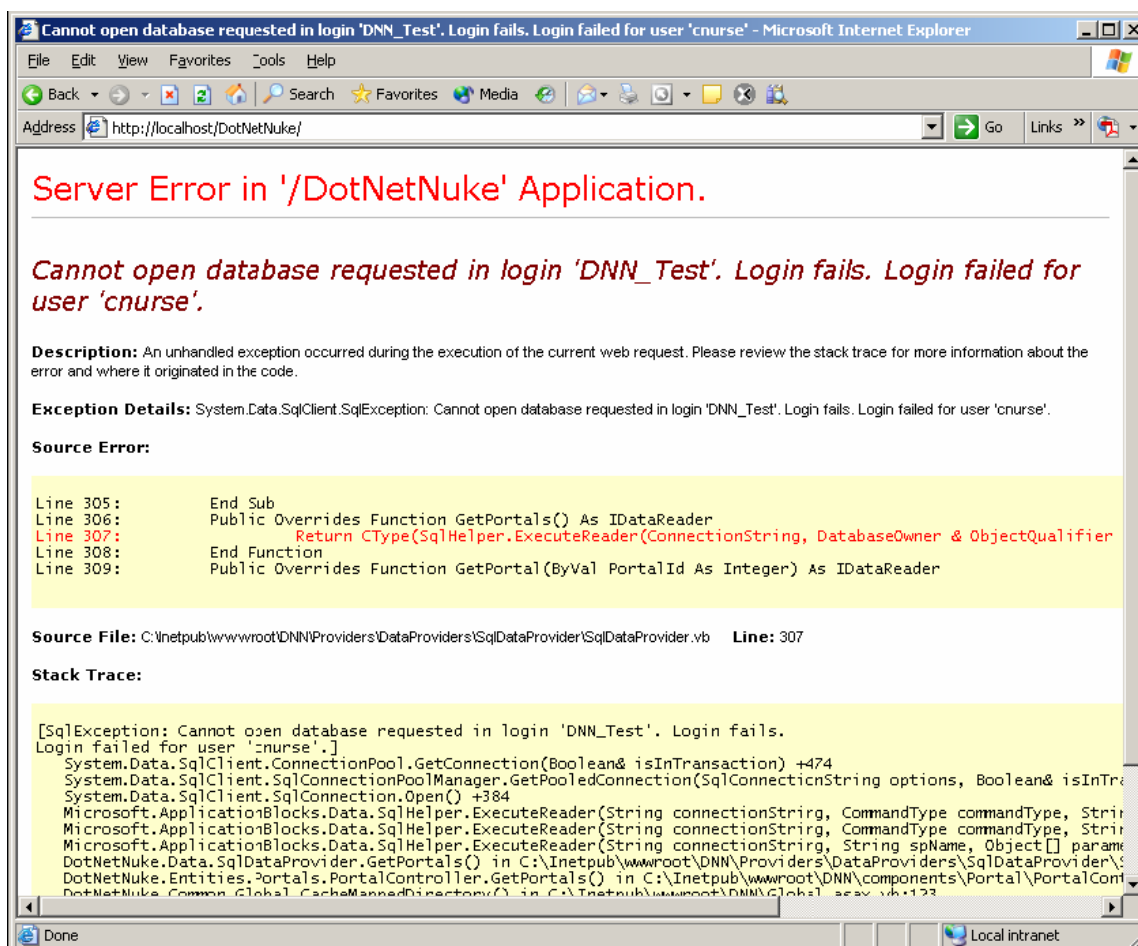
这一节我们将对一些常见的失败或出错的情况进行解答。

错误：不能连接到 connectionString 中指定的 SqlDataProvider 数据库(Could not connect to database specified in connectionString for SqlDataProvider.)



这个错误是由于不正确的连接字符串引起的一本例中是由于数据库拼写错了，应该是 DNN_Test 而不是 DNN_Teest

不能打开登陆请求的数据库 “{ Database}”。登陆失败。使用 “{ user }” 用户登陆失败。(Cannot open database requested in login '{Database}'. Login fails. Login failed for user '{user}')



这是由于在连接字符串的提供的用户不能登陆的数据库造成的一这里是因为，在 SQLServer 数据库中还没有创建“cnurse”这个用户。另外还可能是因为信任连接字符串不正确(如输入的密码不正确)

找不到存储过程“dbo.GetPortals” (Could not find stored procedure 'dbo.GetPortals')

这是因为 {**objectqualifier**} 不正确。在 **dnn.config** 中存储的数据库版本和程序集/应用程序的版本一致，但是由于 {**objectqualifier**} 不正确，应用程序试图执行一个未知的存储过程。这里，被调用的存储过程显示，{**objectqualifier**} 定义成空字符串，但是数据库是使用“dnn_”创建的。

还有可能是这样的原因，你打算进行全新的安装（新的数据库）到一个已经存在一个 DotNetNuke 安装的目录。这种情况下 **dnn.config** 也报告数据库版本和程序集版本符合，但实际情况不是这样，因为我们使用新的数据库。要解决这个问题，把 **dnn.config** 文件从 **Install** 目录下删除就行了。

这种情况实际是很类似的，这两种情况都是 **dnn.config** 报告数据库版本和程序集的版本一致，而实际上不是。第一种情况是 {**objectqualifier**} 不正确，第二种数据库实际上是空的。

附录一 DotNetNuke 安装模板

前面提到过，使用模板文件控制安装使这一过程非常清楚。在 **web.config** 的“InstallTemplate”设置中定义。你即可以编辑默认文件 **DotNetNuke.install.resource**，也可以自己创建一个文件，然后编辑“InstallTemplate”设置。安装模板文件是一个位于 **Intall** 目录下的 **xml** 文件。

下面对这个文件中的节点/属性提供了解释。附件中包含一个打包进发布中的模板的拷贝。

<dotnetnuke>节点

<dotnetnuke>节点是模板的根节点。所有其它节点都是<dotnetnuke>的子节点。它有 7 个直接子节点

✧ <description>子节点

提供模板的描述

✧ <scripts>子节点

定义创建数据库的脚本文件（这些文件必须放在 **web.config** 中默认 **Data provider** 的“providerPath”属性指定的位置），稍后说明

✧ <version>子节点

<scripts>文件创建的数据库的版本。一般是当前版本，但如果少于当前版本，就将运行相应的升级脚本，使数据库版本=应用程序版本。例如，如果模板的脚本是 3.0.9 程序集版本是 3.0.12，那么<scripts>中定义脚本依照 3.0.10, 3.0.11 和 3.0.12 运行。

✧ <superuser>子节点

关于 superuser(host 帐户)的信息，后面再讨论

✧ <settings>子节点

应用程序默认的 HostSettings—该节点的子节点对应数据库中的 HostSettings 表，在安装以后可以通过 Host/Host Settings 菜单管理它。

✧ <desktopmodules>子节点

要安装的 desktop modules—稍后讨论

✧ <portals>子节点

将被安装的一个或多个 portal—稍后讨论

<scripts>节点

<scripts>节点定义创建数据库的脚本。它可以包含任意多个<script>子节点

✧ <script name="xxx">子节点

每个<script>子节点定义一个将被执行的脚本。name 属性用来定义脚本文件（如，用来做什么）。元素的值是脚本文件的文件名。

<superuser>节点

<superuser>节点配置默认的超级用户（或 host 帐户）。它有 7 个直接子节点

✧ <firstname>子节点

超级用户的名（first name）

✧ <lastname>子节点

超级用户的姓（last name）

✧ <username>子节点

超级用户的用户名（name）

✧ <password>子节点

超级用户的密码—注意密码必须和上面<memberrolesprototype>设置的相吻合。

✧ <email>子节点

超级用户的 email 地址

✧ <locale>子节点

超级用户的位置 (culture)

✧ <timezone>子节点

超级用户的时区 (GMT 时间)

<portals> 节点

<portal>节点配置默认 portal。它包括一系列的<portal>节点，每个节点有 8 个直接子节点。

✧ <portalname>子节点

Portal 的名字(显示在标题栏)

✧ <administrator>子节点

配置使用此 portal 的管理员用户—结构和<superuser>相同—如上

✧ <description>子节点

Portal 的描述 (用于 Html 页的 META 标签)

✧ <keywords>子节点

Portal 的关键词 (用于 Html 页的 META 标签)

✧ <templatefile>子节点

用于 portal 的模板文件的文件名 (必须位于 Portals/_default 目录下)

✧ <homedirectory>子节点

设置 portal 的主页目录 (一般留空—默认是 Portals/[PortalId])

✧ <portalaliases><portalalias>子节点

Portal 的别名 (一般留空—默认为安装运行的位置)

✧ <ischild>子节点

必须为空 (以后使用)

模板示例

```
<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</FileExtensions>
<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>
<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/>
<portals>
<portal>
<portalname>DotNetNuke Default Portal</portalname>
<administrator>
<firstname>Administrator</firstname>
<lastname>Account</lastname>
<username>admin</username>
</administrator>
</portal>
</portals>
</desktopmodules>
</dotnetnuke>
```

```
<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>
```

其它信息

DotNetNuke 门户应用程序构架一直都在被修订和改进。要得到最新版本的软件和文档，请访问 DotNetNuke 的官方网站：

<http://www.dotnetnuke.com>

下面的的几个网站提供 DotNetNuke 技术和与相关概念的帮助信息。

DotNetNuke 社区

<http://forums.asp.net/90/ShowForum.aspx>

Microsoft. ASP.Net

<http://www.asp.net>

Open Source

<http://www.opensource.org/>

W3C Cascading Style Sheets, level 1

错误和遗漏

如果你发现本文档的任何错误和遗漏，[请发邮件到marketing@dotnetnuke.com](mailto:marketing@dotnetnuke.com)。请提供文档的标题，错误或遗漏所在的页码和其相关信息，以便帮助我们修正错误。

附录 A：文档历史

版本	最新更新	作者	修改
1.0.0	2005 年 3 月	Charles Nurse	最初的 3.0 安装指南
1.0.1	2005 年 6 月	Charles Nurse	更新到 3.1.0 版本
1.0.2	2005 年 8 月 16 日	Shaun Walker	使用新模板
1.0.3	2005 年 11 月 22 日	Charles Nurse	更新到 3.2/4.0 版本