

Einführung in die Programmierung mit DotNetNuke

- **Das Guestbook-Modul Template für Visual Studio**
- **Das Formular-Modul Template für Visual Studio**

Kursmaterialien

dnnWerk ist der Verbund deutschsprachiger DotNetNuke Experten. Besuchen Sie uns im Internet unter <http://www.dnnwerk.de>

Die Portalfabrik

Philipp Becker
Struckbergstrasse 17
27721 Ritterhude
Fon: +49 4292 810953
Fax: +49 4292 810243
<http://www.portalfabrik.net>

Gamma Concept GmbH

Sebastian Leupold
Zähringerstr. 43
76133 Karlsruhe
Fon: + 49 721 3545210
Fax: +49 721 3545299
<http://www.gamma-concept.de>

ITM GmbH

Benjamin Hermann
Wankelstr. 14
70563 Stuttgart
Fon: +49 711 4512922
Fax: +49 711 4512929
<http://www.itm-consulting.de>

Einführung

Das Guestbook Modul liegt als Visual Studio Template vor und zeigt exemplarisch eine Reihe von „Best Practises“ für die Programmierung von DotNetNuke Modulen. Das Template liegt als WSP Modul vor und kann sowohl in den kommerziellen Edition von Visual Studio als auch in der Express Edition verwendet werden.

Voraussetzungen:

a. Visual Studio 2005 / 2008 oder Visual Web Developer 2005 / 2008

<http://www.microsoft.com/germany/Express/download/default.aspx>

b. SQL Server 2000 / 2005 oder SQL Server 2005 Express

<http://www.microsoft.com/germany/Express/download/default.aspx>

c. DotNetNuke Starterkit

http://sourceforge.net/project/showfiles.php?group_id=77052&package_id=77939&release_id=602419

Bitte laden Sie sich für die aktuelle version von DotnetNuke das Paket „Starterkit.vsi“ herunter und installieren Sie das Paket per Doppelklick auf die heruntergeladene Datei. Visual Studio (Punkt a) muss zuvor installiert sein.

Das Starterkit installiert eine Reihe von Vorlagen, je nach verwendeter VS Edition werden u.U. nicht alle Vorlagentypen korrekt installiert. So ist es unter Visual Web Developer Express nicht möglich, die Vorlage „DotNetNuke Compiled Module“ zu installieren, weil VWD den Projekttyp „Wap Project“ noch nicht unterstützt.

Die angehängte Vorlage „**dnnWerk Modul**“ kann einfach in das Vorlagenverzeichnis von Visual Studio kopiert werden und steht damit sofort im Assistent für neue Komponenten in Visual Studio zur Verfügung.

Unter Windows Vista findet sich das Vorlagen Verzeichnis beispielhaft an der folgenden Position:

C:\Users\[IHR BENUTZERNAME]\Documents\Visual Studio 2008\Templates\ItemTemplates\Visual Web Developer

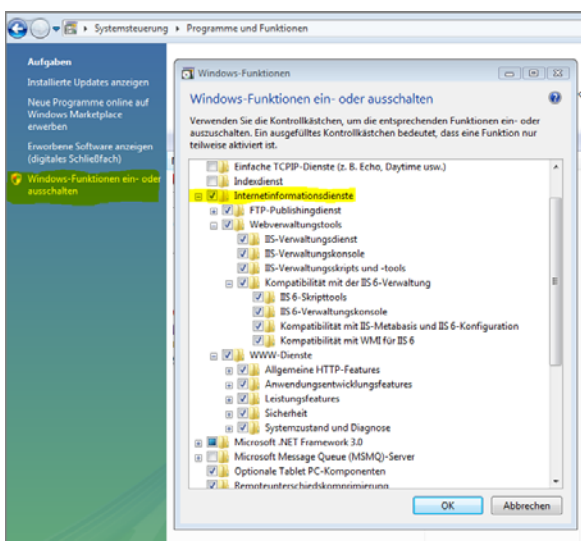
(Generell ist zu sage, dass Vorlagen unter *Eigene Dateien* zu finden sind, unter XP ist damit der Pfad entsprechend anzupassen, ähnlich anzupassen ist der Pfad entsprechend für Visual Studio 2005)

Einrichtung der Entwicklungsumgebung

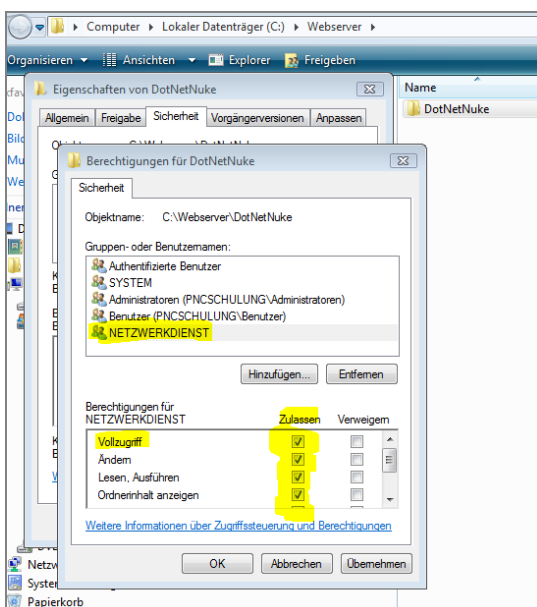
Die Einrichtung der Entwicklungsumgebung ist hier für den Betrieb unter Windows Vista Business beschrieben und muss in Details ggf. für den Betrieb unter Windows XP Professional angepasst werden. Die grundsätzliche Vorgehensweise ist jedoch identisch.

Achtung! Unter Windows Vista muss Visual Studio als Administrator ausgeführt werden! Hierfür muss das Programm u.U. über die rechte Maustaste (Als Administrator ausführen) gestartet werden

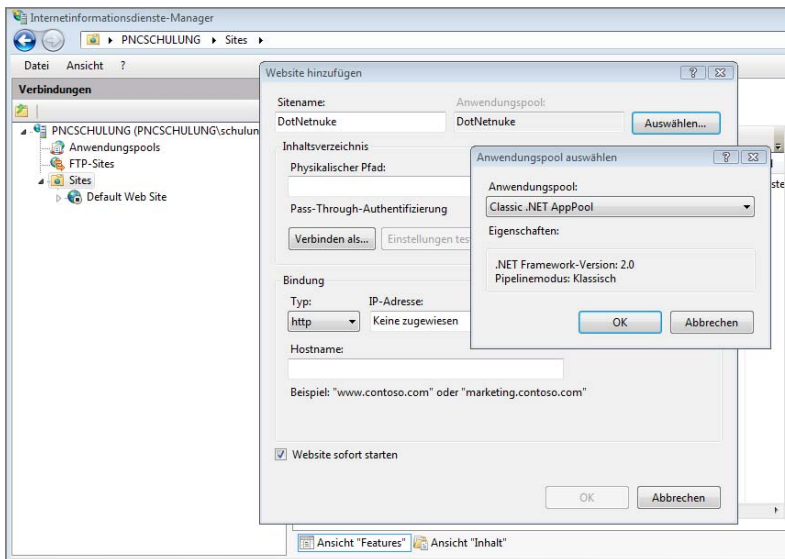
1. **Internet Information Services (IIS) Webserver** in Vista aktivieren und einrichten. Bitte beachten, dass wirklich alle Subfeatures (inkl. der IIS 6 Kompatibilität) aktiviert sind.



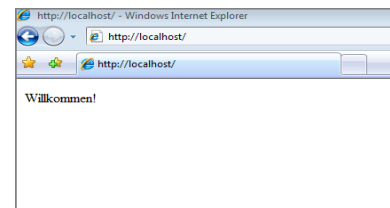
2. Ein neues Verzeichnis anlegen, z.B. C:\Webserver\DotNetnuke. In den Sicherheitseinstellungen für das neue Verzeichnis dem Konto **Netzwerkdienst Vollzugriff** gewähren:



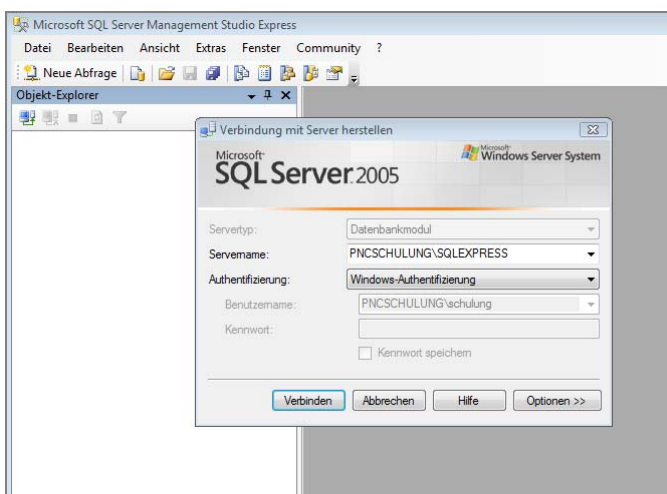
- In der IIS Verwaltungskonsole (Suche > IIS Manager) eine neue Webseite einrichten. Meiner Erfahrung nach ist es einfacher zunächst den klassischen Anwendungspool Modus zu nutzen; als physikalisches Verzeichnis nutzen wir das zuvor erstellte Verzeichnis.



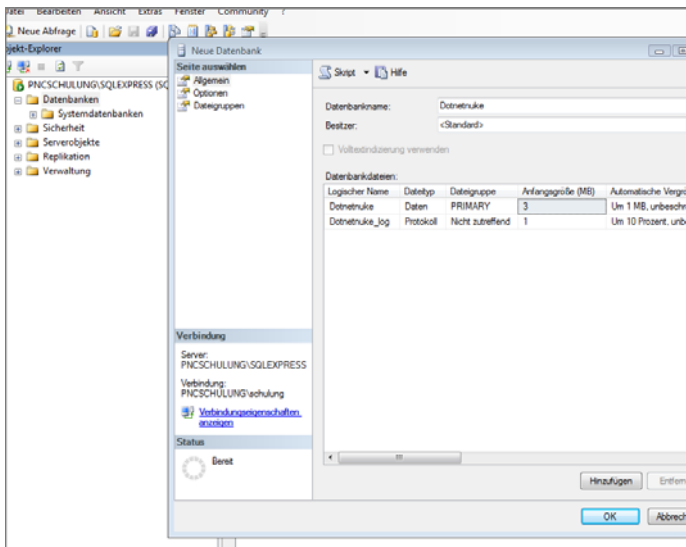
- Ich persönlich teste ganz gerne, ob die Webseite auch wirklich funktioniert. Hierzu erstelle ich mir in dem gerade erstellen Verzeichnis eine einfache Textdatei mit dem Inhalt „Willkommen“ und speichere diese als default.htm ab. Im Browser sollte der URL Aufruf von <http://localhost> dann wie in der Grafik aussehen. Bei Erfolg die Datei wieder löschen!



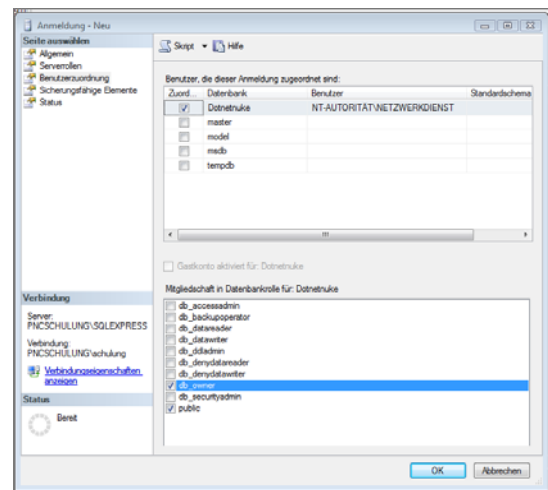
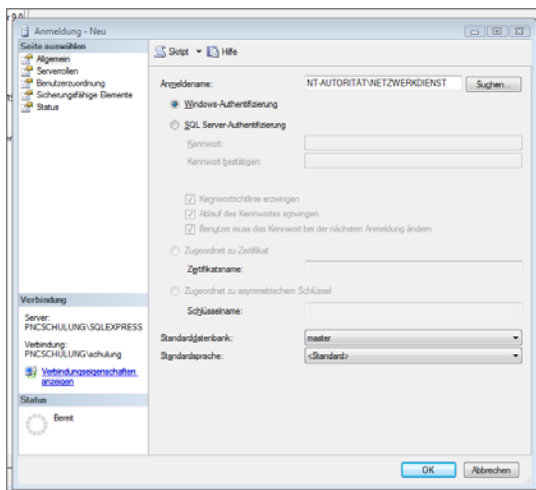
- Gemäß o.g. Links **Visual Web Developer Express**, **SQL Server Express** und das **DotNetNuke StarterKit** vollständig installieren
- Das SQL Server Management Studio öffnen, standardmäßig sollte man sich mit der Instanz `Rechnername\SQLExpress` verbinden können.



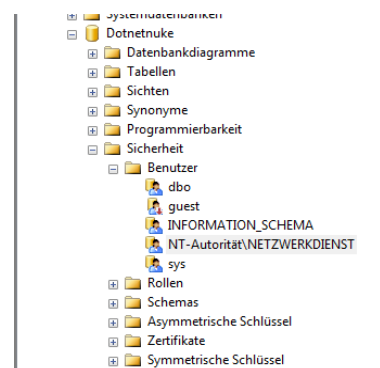
7. Eine neue Datenbank erstellen, wir geben dieser den Namen **DotNetNuke** und übernehmen alle vorgeschlagenen Standardwerte.



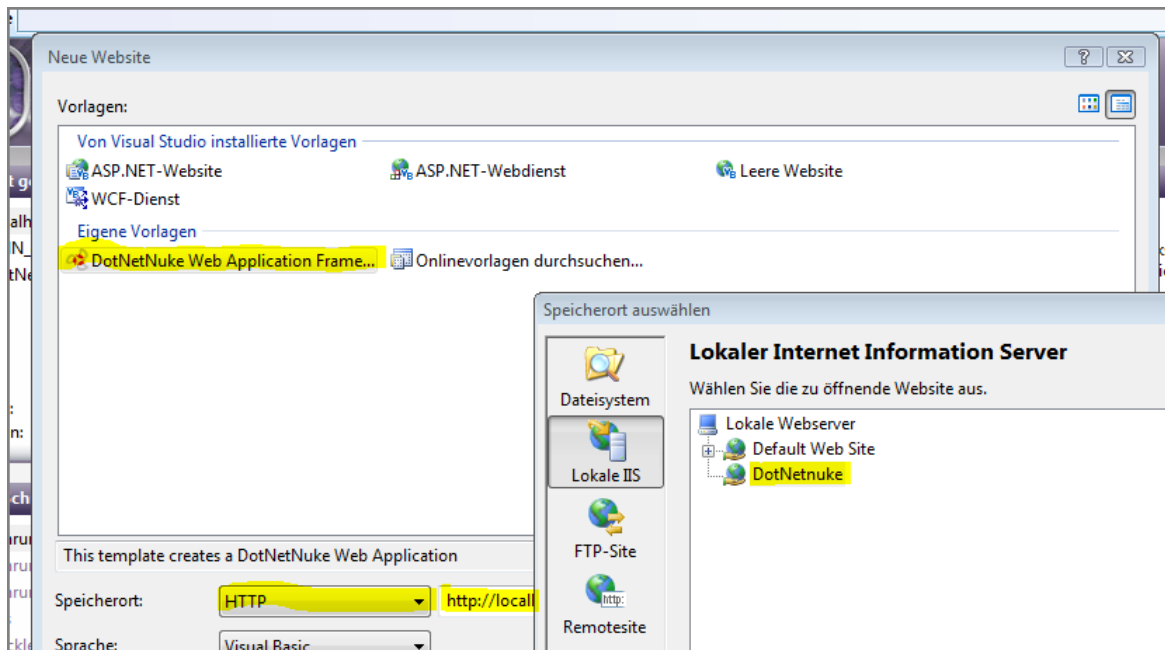
8. Auf Datenbankserver Ebene erstellen wir einen neuen Login für das Windowskonto **Netzwerkdienst** und geben diesem über den Menüpunkt **Benutzerzuordnung** die **dbo Berechtigung** für die zuvor erstellte Datenbank. Den Benutzer muss man evtl. händisch in der Form NT-AUTORITÄT\NETZWERKDIENT eintragen, da der Benutzername RECHNERNAME\NETZWERKDIENT manchmal nicht zugeordnet werden kann, NT-AUTORITÄT sollte aber immer funktionieren.



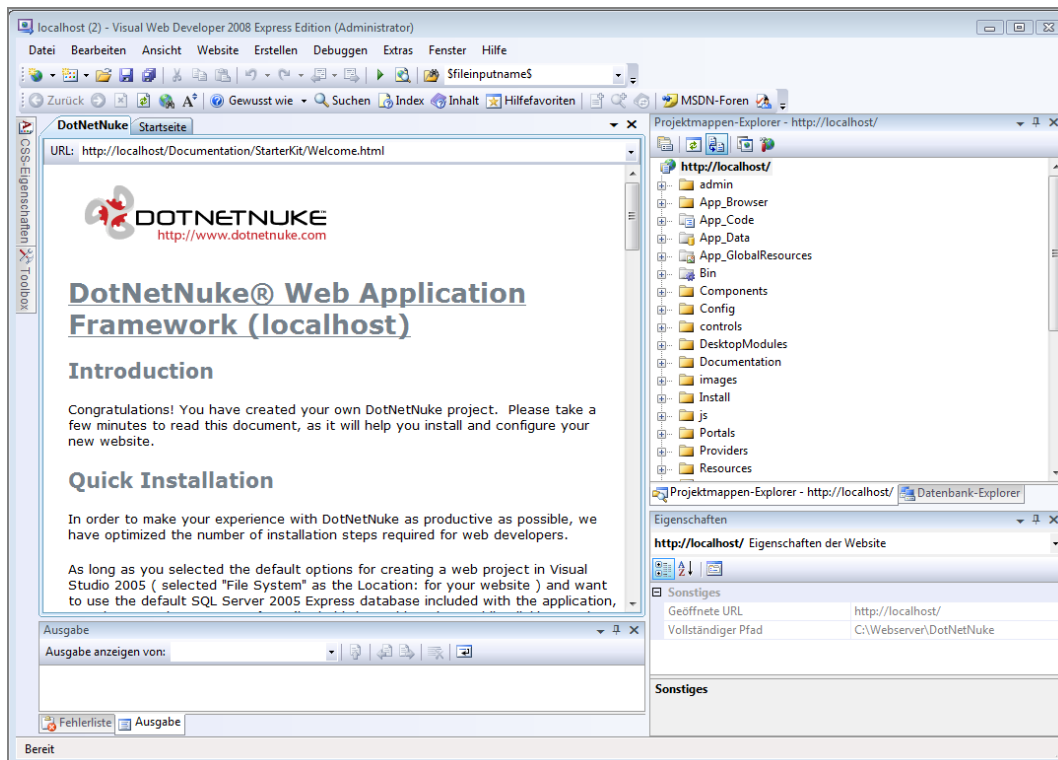
Die Datenbank Sicherheit sollte sich am Ende wie folgt darstellen:



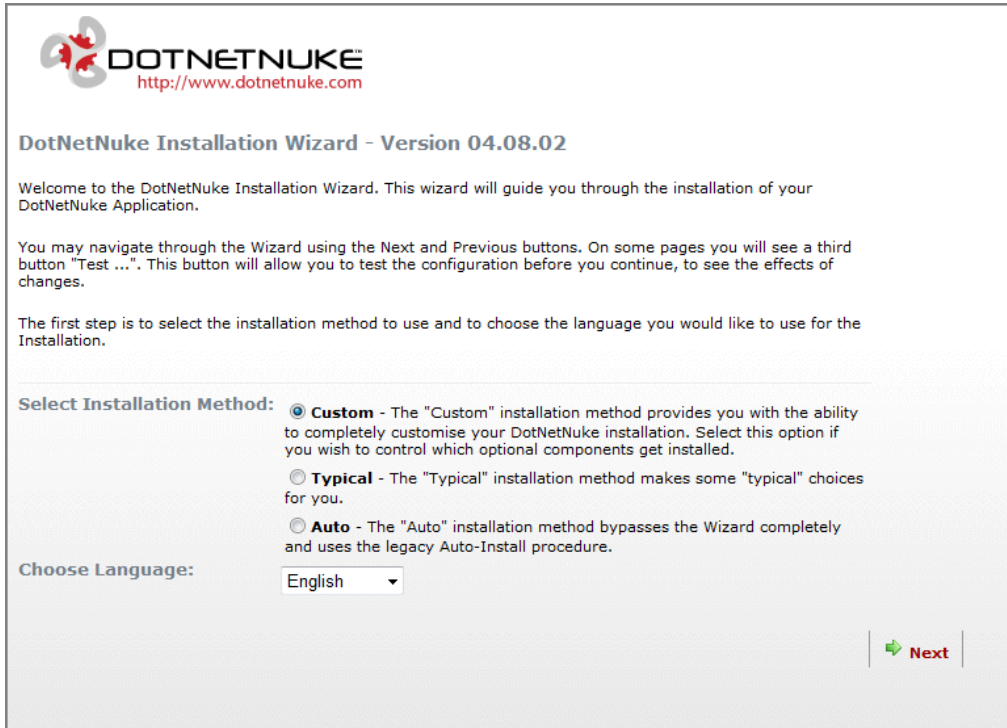
9. Jetzt öffnen wir die erstellte Webseite in Visual Studio. Dazu starten wir Visual Studio und gehen im Menü **Datei** auf **Neue Webseite**. Es öffnet sich der Assistent für neue Webseite, dieser sollte an dieser Stelle **DotNetNuke** als Vorlage für die neue Seite anbieten. Als Speicherort wählen wir **HTTP**, über den Klick auf **Durchsuchen** neben dem URL sollten wir die gerade erstellte Webseite im IIS auswählen können.




Der Klick auf **OK** kopiert jetzt alle notwendigen Dateien aus der Vorlage in das zuvor erstellte Verzeichnis der IIS Seite. Auf Nachfragen müssen wir den Assistenten u.U. anweisen, an der gefragten Stelle eine neue Webseite zu erstellen. Der Assistent sollte mit folgendem Bild enden:



10. Wir starten jetzt unsere neue DotNetNuke Seite um die die finale Einrichtung abzuschließen. Wir erledigen das über den Menüpunkt **Debuggen > Starten ohne debuggen**. Es sollte ein Build Prozess starten, der einige Zeit benötigen wird, da es sich hierbei um den ersten Aufruf der Seite handelt. Am Ende sollte im Browser der Installationsassistent von Dotnetnuke aufgerufen werden (Es kann passieren, dass nur unsere default.htm angezeigt wird, die wir vorhin zum Testen erstellt haben, in diesem Fall diese Datei einfach löschen und den Buildvorgang nochmals auslösen)



 **DOTNETNUKE™**
http://www.dotnetnuke.com

DotNetNuke Installation Wizard - Version 04.08.02

Welcome to the DotNetNuke Installation Wizard. This wizard will guide you through the installation of your DotNetNuke Application.

You may navigate through the Wizard using the Next and Previous buttons. On some pages you will see a third button "Test ...". This button will allow you to test the configuration before you continue, to see the effects of changes.

The first step is to select the installation method to use and to choose the language you would like to use for the Installation.

Select Installation Method:

- Custom** - The "Custom" installation method provides you with the ability to completely customise your DotNetNuke installation. Select this option if you wish to control which optional components get installed.
- Typical** - The "Typical" installation method makes some "typical" choices for you.
- Auto** - The "Auto" installation method bypasses the Wizard completely and uses the legacy Auto-Install procedure.

Choose Language: English

[Next](#)

11. Wir schalten zunächst auf die Sprache **deutsch** um, wählen **Individuell** als Installationsmethode aus und klicken dann auf **Weiter**

12. Im nächsten Schritt müssen wir unbedingt zunächst die Berechtigungen prüfen lassen. Sollten hier Fehler auftreten, hat die Berechtigungserteilung auf Ordner Ebene nicht funktioniert und muss geprüft werden (siehe Punkt 2)



 **DOTNETNUKE**
http://www.dotnetnuke.com

Dateizugriffsrechte werden geprüft

Das System erlaubt es, Dateien für Inhalte und installierbare Erweiterungen hochzuladen. Diese Funktionen erfordern erweiterte Berechtigungen, so dass die Anwendung in der Lage ist, Ordner und Dateien zu erstellen und zu löschen.

Diese Seite prüft die aktuellen Dateizugriffsrechte, um sicherzustellen, dass diese Funktionen problemlos ausgeführt werden können.

Zusammenfassung der Dateiberechtigungen:

- Ordner anlegen
- Datei erstellen
- Datei löschen
- Ordner löschen

Ihre Web-Site hat die Berechtigungsprüfung erfolgreich absolviert.

[Berechtigungen testen](#) | [zurück](#) | [weiter](#)

13. Der folgende Schritt ist sehr wichtig, hier legen wir fest, wie sich DNN mit der Datenbank verbinden soll, wir übernehmen hier alle Einstellungen entsprechend der Grafik:

DOTNETNUKE
http://www.dotnetnuke.com

Datenbankverbindung konfigurieren

Sie können auf dieser Seite die Datenbank-Einstellungen aktualisieren. Wenn Sie das Portalsystem bei einem Hosting Provider mit separater Datenbank installieren, hat dieser Ihnen die entsprechenden Zugangsdaten zur Verfügung gestellt.

Es gibt zwei Optionen für SQL-Server 2005: SQL-Server 2005 unterstützt Datenbanken und Datenbankdateien, i.d.R. werden Sie hier Datenbanken bevorzugen. Bei SQL-Server 2005 Express ist die Einrichtung von Datenbankdateien besonders einfach.

Datenbank wählen: SQL-Server 2005 Express Datei **SQL-Server 2000/2005 Datenbank**

Server: Geben Sie den Namen oder die IP-Adresse des Rechners an, auf dem der Datenbank-Server installiert ist. (Bei Verwendung von Oracle geben Sie die Datenquelle (SID) an)

Datenbank: Geben Sie den Datenbanknamen ein

Integrierte Sicherheit: Aktivieren Sie diese Option, wenn Sie sich über "integrierte Sicherheit" am SQL-Server mit dem Windows Benutzerkonto anmelden wollen. Wenn Sie SQL-Server Express verwenden, werden Sie diese Option vermutlich benötigen. Wenn Sie explizit einen Benutzernamen mit Kennwort für die Anmeldung am SQL-Server erhalten oder eingerichtet haben, deaktivieren Sie diese Option.

Als DB-Owner ausführen: Aktivieren Sie diese Option, wenn Sie auf die Datenbank als Eigentümer (DBOwner) zugreifen. Wenn die Option nicht aktiv ist, führen Sie sie als angegebener Benutzer aus.

Objekt-Präfix: Sie können bei Bedarf einen "Präfix" angeben, der allen Objektnamen vorangestellt wird - dadurch wird sichergestellt, dass es zu keinen Namenskonflikten kommt, wenn weitere Anwendungen die Datenbank mitverwenden.

Verbindung hergestellt

[Datenbankverbindung testen](#) | [zurück](#) | [weiter](#)

Sollte hier der Verbindungstest scheitern, kann es nicht weitergehen. In diesem Fall bitte zurück zu Punkt 6 und das Anlegen und Freigeben der Datenbank nochmal prüfen.

14. In den folgenden Schritten werden wir nach den zu erstellenden Zugangskonten gefragt, das System fragt ab, ob zusätzliche Module installiert werden sollen und bietet an, optionale Dienste, Sprachpakete und Layouts zu installieren. Wir übernehmen hier jeweils die Standardwerte und sollten am Schluss folgende Seite erreichen:

DOTNETNUKE
http://www.dotnetnuke.com

Herzlichen Glückwunsch

Sie haben DotNetNuke erfolgreich installiert.

Weitere Unterstützung erhalten Sie bei der deutschsprachigen DotNetNuke User-Group.

[Beendet \(Web-Site aufrufen\)](#)

15. Ein Klick auf **Webseite aufrufen** sollte uns direkt in das neue DNN Portal bringen:

The screenshot shows the DotNetNuke (DNN) portal home page. At the top left is the logo for DOTNETNUKE with the URL <http://www.dotnetnuke.com>. Below the logo is a navigation bar with "Home" and "Web Site" tabs, and a search box. The date "Monday, June 09, 2008" and the breadcrumb "...: Home :..." are displayed. On the right side of the navigation bar are links for "Register" and "Login".

The main content area is divided into several sections:

- Need Help?**: A section with a graphic of three silhouettes and the text "Get Support." with a small icon below it.
- Welcome To DotNetNuke®**: A central section containing:
 - Text: "If your site was installed automatically, a couple of default user accounts were created which allow you to access the various site management features offered by the application. In this case, please take note of the following user account information:"
 - Administrator Login:**
 - Username: admin
 - Password: dnnadmin
 - Text: "* The Administrator account is a privileged user account which allows you to manage various aspects of your site. Once you login as the Administrator, an Admin menu option will be visible in the site navigation. Please note that the first time you login to the site using this account, you will be prompted to enter a new password for security purposes."
 - Host Login:**
 - Username: host
 - Password: dnnhost
 - Text: "* The Host account is a highly privileged user account which allows you to manage various aspects of your hosting environment. Once you login as the Host, a Host menu option will be visible in the site navigation. Please note that the first time you login to the site using this account, you will be prompted to enter a new password for security purposes."
 - Small icons for RSS and a globe.
- DotNetNuke Books**: A section at the bottom of the central area with a small icon.
- Links**: A sidebar on the left containing:
 - DotNetNuke® Website
 - Project Downloads
 - Resource Directory
 - Discussion Forum
 - Project Blogs
 - Guided Tour
 - Language Packs
- Sponsors**: A sidebar on the right containing logos for:
 - Microsoft ASP.net
 - MaximumASP (server hosting, server choice)
 - telerik (deliver more than expected)
 - ExactTarget.
 - easyCGI (ADVANCED HOSTING EASY SOLUTIONS)
 - CrystalTech (DotNetNuke Hosting)

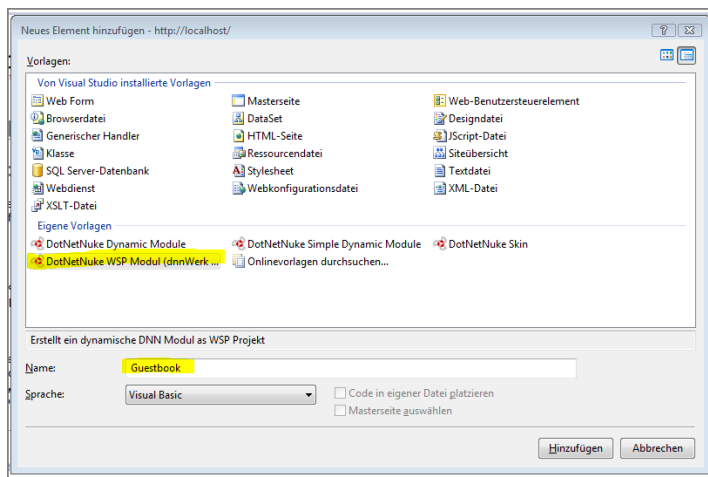
16. Wir kopieren jetzt das beigefügt Template in das Visual Studio eigene Vorlagen Verzeichnis. Wir finden dies allgemein unter Eigene Dateien in einem von Visual Studio angelegten Verzeichnis für Item Templates. Unter Windows Vista und Visual Web Developer Express 2008 lautet der Pfad bspw.

C:\Users\[BENUTZERNAME]\Documents\Visual Studio 2008\Templates\ItemTemplates\Visual Web Developer

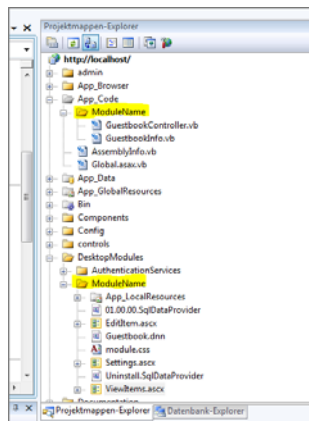
17. Wieder zurück in Visual Studio können wir jetzt ein eigenes Modul erstellen, indem wir im Projektmappenexplorer
- den Stammknoten **localhost** markieren
 - den Menüpunkt **Datei > Neue Datei** auswählen

Es sollte sich wiederum ein Assistent öffnen, der die vom Starterkit bereitgestellten, bzw. das von uns bereitgestellte Template anbietet.

Als Modulname wählen wir einen Namen für unser neues Modul aus, dieser Name wird bei Erstellung des Moduls in den Klassen und Dateien entsprechend verwendet. Wenn wir ein Gästebuchmodul erstellen wollten, würden wir also ggf. z.B. **Guestbook** hier eintragen:



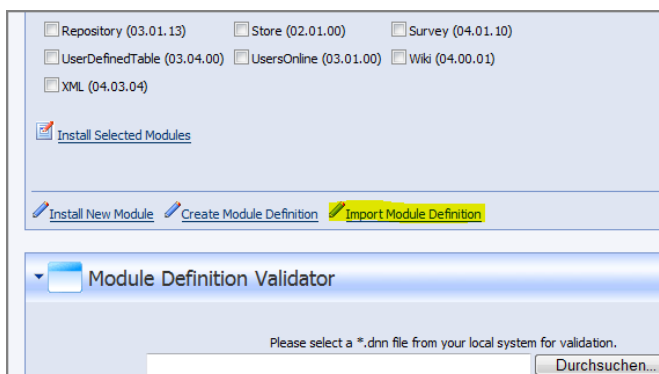
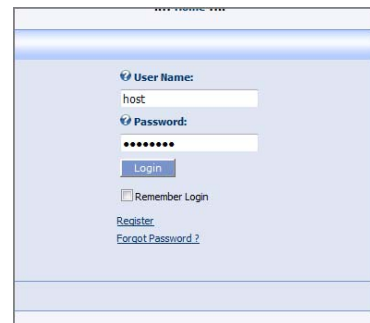
18. Visual Studio erstellt nun für uns das neue Modul, unglücklicherweise werden aber die Verzeichnisnamen nicht korrekt angelegt, wir müssen daher im ersten Schritt die beiden Verzeichnisse „Modulname“ in „YourCompany.Guestbook“ umbenennen:



Hinweis: Das Modul aus dem Template läuft im Namespace **YourCompany.Modules.ModuleName**. Selbstverständlich ist YourCompany in einem echten Modul durch den eigenen Firmennamen zu ersetzen, um keine Konflikte mit Modulen anderer Hersteller zu verursachen.

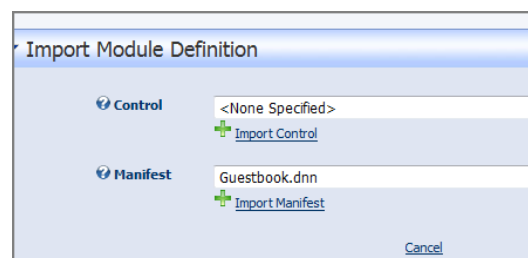
Es empfiehlt sich daher, in allen durch das template erstellten Dateien den Begriff „YourCompany“ durch einen eigenen Begriff zu ersetzen. Die Funktion „Suchen und ersetzen“ in Visual Studio erledigt das zuverlässig.

19. Wenn wir uns jetzt in das Portal einloggen (wir verwenden den im Installationsassistenten erstellten Hostzugang), können wir unser neues Modul ein DNN registrieren und verwenden.



Wir wechseln im ersten Schritt nach Host > Moduledefinitionen und klicken weiter unten auf der Seite auf „Import Moduledefinition“

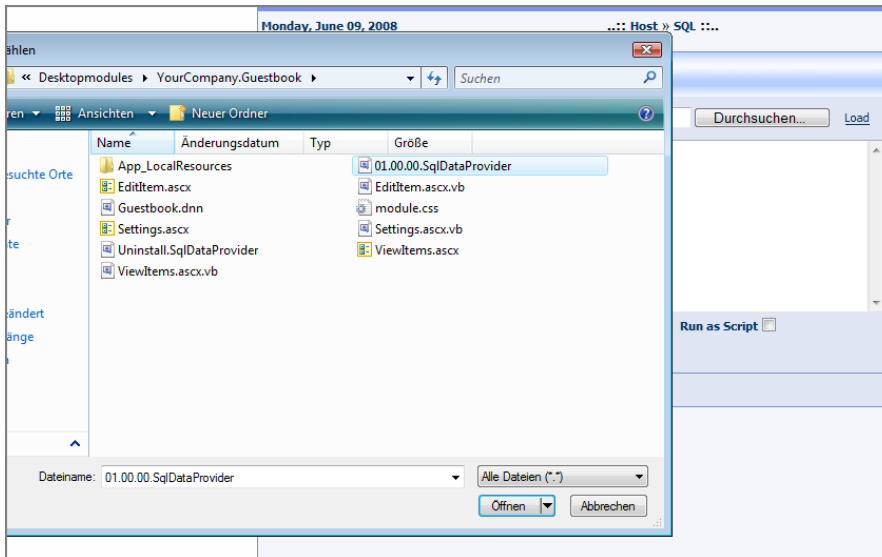
Es wird u.a. unser Guestbook Modul angeboten, wir klicken hier auf „Import Manifest“...



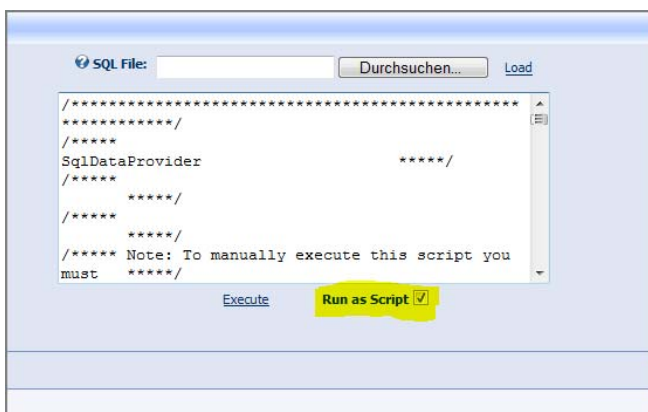
	Feed Explorer	Allows users to browse RSS feeds using
	Google AdSense	Allows you to create Google AdSense ad
	Guestbook	A Guestbook module
	Links	This module renders a list of hyperlinks. edit the Links data stored in the SQL da
	MarketShare	The DotNetNuke MarketShare affiliate p commission on gross product sales throu
	Search Input	The Search Input module provides the a

...und stellen fest, dass unser Modul jetzt registriert ist.

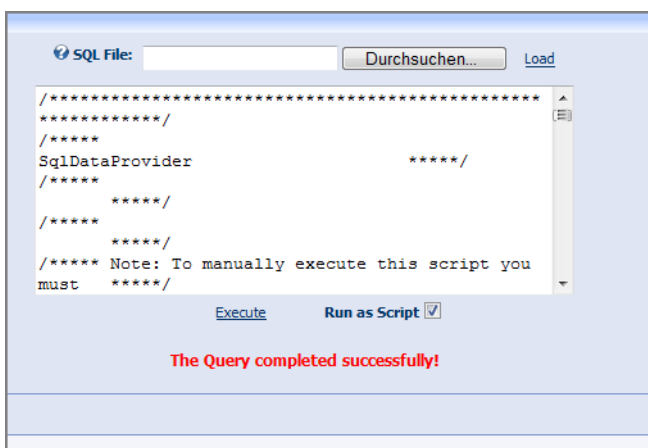
20. Im nächsten Schritt nehmen wir die für den Betrieb unseres Moduls notwendigen Datenbankerweiterungen vor. Hierfür wechseln wir im Portal nach Host > SQL und wählen die Datei 01.00.00.SqlDataProvider aus dem Modulverzeichnis als SQL Script Datei aus.



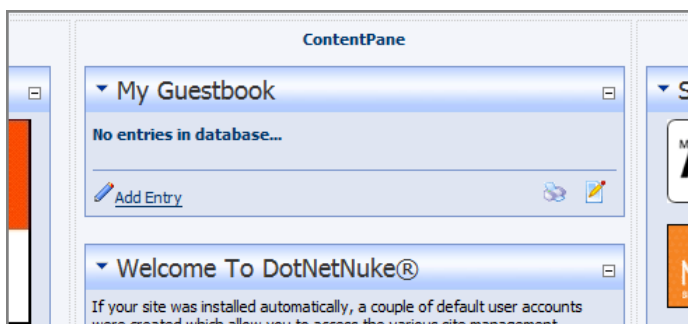
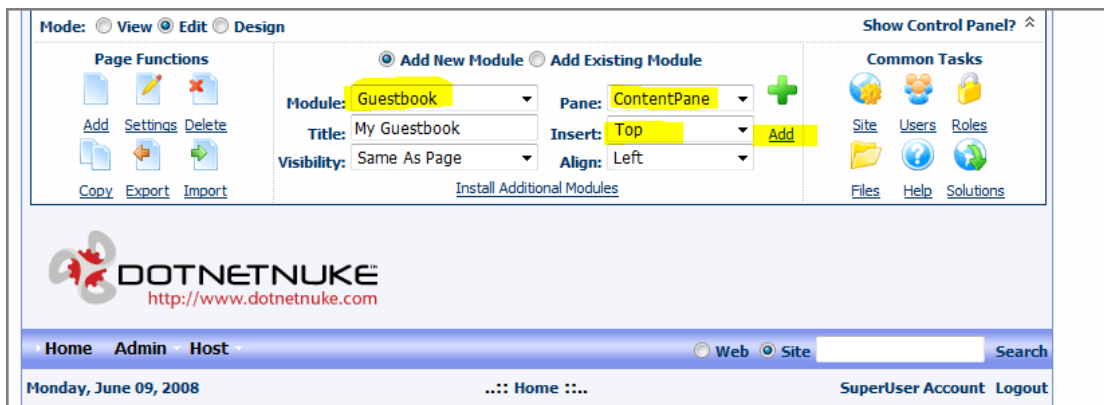
Wir klicken auf „Load“ und laden damit das SQL Script in das Ausführungsfenster.



Wichtig ist hier der Haken „Run as Script“, der dafür sorgt, dass wirklich das ganze Skript abgearbeitet wird. Ein Klick auf **Execute** sollte das Script ausführen und folgendes Resultat liefern:



Wir haben damit alle Vorbereitungen für den Betrieb unseres Moduls abgeschlossen und sollten damit das Modul jetzt verwenden können. Wir gehen dazu auf die Startseite unseres Portals und fügen das Modul über das Controlpanel oben auf der Seite hinzu.



Voilà, unser Guestbook Modul ist erfolgreich auf der Seite platziert.

Im nächsten Kapitel sehen wir uns detaillierter an, was das Template im Einzelnen für uns angelegt hat und was wir mit dem Modul nunmehr machen können.

Inhalt des Templates *dnnWerk Datenbank Modul (WSP)*

Das eingangs erwähnte und in dieser Anleitung verwendete Template erstellt folgende Verzeichnisse und Dateien, deren Bedeutung hier kurz erläutert sein soll.

Hinweis: Visual Studio legt das Modul in zwei Unterverzeichnissen an, die zunächst manuell umbenannt werden müssen! Die im Template enthaltene Manifestdatei registriert das Modul im Verzeichnis

YourCompany.[Name des Moduls]

Es muss also zunächst die Verzeichnisse „Modulname“ in YourCompany.[Name des Moduls] umbenannt werden!

U.g. Ausführungen gehen davon aus, dass dieser Schritt erledigt ist und dass das Modul unter dem Namen Guestbook angelegt worden ist!

Es gibt im Wesentlichen zwei entscheidende Unterverzeichnisse in DotnetNuke, in denen Code für eigene Module platziert werden kann, namentlich Desktopmodules und App_Code. Daher legt Visual Studio bzw. unser Template auch genau dort seine Dateien ab.

1. Desktopmodules/YourCompany.Guestbook

Die Präsentations- und Konfigurationsebene unseres Moduls

a. Guestbook.dnn

Die Manifestdatei wird von DNN verwendet um die u.g. Dateien in DNN zu registrieren und deren Funktion zu definieren. Die Datei wird benötigt, um ein Modul in DNN zu installieren.

b. EditItem.ascx

c. EditItem.ascx.vb

[abgeleitet von Dotnetnuke.Entities.Modules.PortalModuleBase]

Das Formular Control zum Erstellen und Bearbeiten von Einträgen. Dieses Control wird vom Manifest als Edit Control registriert, so dass nur Benutzer mit Edit Berechtigung Einträge Bearbeiten bzw. Erstellen dürfen.

d. Settings.ascx

e. Settings.ascx.vb

[abgeleitet von Dotnetnuke.Entities.Modules.ModuleSettingsBase]

Das Einstellungsformular wird als Settings Control registriert, so dass wir Moduleinstellungen in DNN hinterlegen können. Settings Controls stehen ebenfalls nur Benutzern mit Edit Berechtigung zur Verfügung.

f. ViewItems.ascx

g. ViewItems.ascx.vb

[abgeleitet von Dotnetnuke.Entities.Modules.PortalModuleBase]

Das Hauptcontrol wird als View Control registriert, so dass Benutzer mit View Berechtigung auf das Modul Einträge aus der Datenbank angezeigt bekommen können.

h. module.css

Die Stylesheetdatei eines Moduls kann verwendet werden, um modulspezifische CSS Selektoren zu hinterlegen, die nur von dem Modul verwendet werden sollen.

i. 01.00.00.SqlDataProvider

Die SQL Scriptdatei für die Version 1 des zu installierenden Moduls. Je nach zu installierender Modul Version (definiert im Manifest, s.o.) wird diese Scriptdatei ausgeführt. Wird bspw. eine Version 01.00.01 eines Moduls installiert prüft DNN während der Installation, ob eine Scriptdatei mit dem Namen 01.00.00.SqlDataProvider mitgeliefert ist. Ggf. wird diese dann ausgeführt.

SQLdataProvider Dateien enthalten Transact SQL Code mit dem einzigen Unterschied zu echtem T-SQL Code, dass DNN den Text nach {ObjectQualifier} und {DatabaseOwner} Token parst und diese durch die in der web.config eingetragenen Werte ersetzt.

j. Uninstall.SqlDataProvider

Die SQL Scriptdatei, die ausgeführt wird, wenn ein Modul deinstalliert wird. Hier können alle SQL Befehle hinterlegt werden, die notwendig sind, um die erstellten Datenbank Objekte wieder zu entfernen.

2. App_Code/ YourCompany.Guestbook

Die Geschäftslogik unseres Moduls. In der vorliegenden Implementierung gerne auch als DAL++ bezeichnet, weil wir hier Geschäfts- und Datenbankebene zusammengefasst haben.

a. GuestbookInfo.vb

Die Info Objekt Klasse für unser Guestbook. Datenbank Entities wie z.B. ein Gästebucheintrag werden im DNN Framework idealerweise in einer eigenen Info Klasse abgebildet. Unsere vorliegende Klasse implementiert darüber hinaus noch eine Schnittstelle, über die unsere Controller Klasse die Info Klasse mit Einträgen befüllen kann (s.u.)

b. GuestbookController.vb

Die Controller Klasse für unser Modul, die Methoden bereitstellt, über die wir Datenbankeinträge abrufen, erstellen, löschen und bearbeiten können. Die Methoden akzeptieren in der Regel Info Objekte (s.o.) als Parameter.

Inhalt des Templates *dnnWerk Formular Modul (WSP)*

Das ebenfalls beigefügte Formular Modul enthält lediglich ein einziges einfaches View Control, über das eine kurze E-Mail Nachricht an einen beliebigen Empfänger verschickt werden kann.

Hinweis: Visual Studio legt das Modul in einem Unterverzeichnissen an, das zunächst manuell umbenannt werden muss! Die im Template enthaltene Manifestdatei registriert das Modul im Verzeichnis

YourCompany.[Name des Moduls]

Es muss also zunächst die Verzeichnisse „Modulename“ in YourCompany.[Name des Moduls] umbenannt werden!

U.g. Ausführungen gehen davon aus, dass dieser Schritt erledigt ist und dass das Modul unter dem Namen Mailform angelegt worden ist!

Es gibt im Wesentlichen zwei entscheidende Unterverzeichnisse in DotnetNuke, in denen Code für eigene Module platziert werden kann, namentlich Desktopmodules und App_Code. Daher legt Visual Studio bzw. unser Template auch genau dort seine Dateien ab.

1. Desktopmodules/YourCompany.Guestbook

Die Präsentations- und Konfigurationsebene unseres Moduls

a. Mailform.dnn

Die Manifestdatei wird von DNN verwendet um die u.g. Dateien in DNN zu registrieren und deren Funktion zu definieren. Die Datei wird benötigt, um ein Modul in DNN zu installieren.

c. View.ascx

d. View.ascx.vb

[abgeleitet von Dotnetnuke.Entities.Modules.PortalModuleBase]

Das Hauptcontrol wird als View Control registriert, so dass Benutzer mit View Berechtigung auf das Modul Nachrichten verschicken können. Unangemeldete Besucher bekommen das Formular generell nicht angezeigt.

Weiterführende Literatur

Coreteam Mitglied Michael Washington hat eine Reihe von Tutorials veröffentlicht, die sich mit Modulen unter VWD befassen.

<http://www.adefwebserver.com/DotNetNukeHELP/>

Ex-Coreteam Mitglied Scott McCulloch hat zahlreiche Module veröffentlicht, die sich als vollständiges Abonnement inkl. Quellcode gegen äußerst geringes Entgelt käuflich erwerben lassen.

<http://www.ventrian.com>

DotnetNuke-Gründer Shaun Walker hat vor langer Zeit einen ausführlichen Artikel über das Starterkit verfasst

<http://forums.asp.net/p/937685/1114393.aspx>

Community-Mitglied Michael Tobisch hat in der deutschen DNN Usergroup eine Blogreihe zum Thema Programmierung in DNN gestartet:

<http://www.dnn-usergroup.de/Community/Blogs/tabid/80/articleType/ArticleView/articleId/85/Default.aspx>

Letzte Aktualisierung: Philipp Becker, dnnWerk

Datum: 09.06.2008