

# **DNN KickStart**

**THE DOTNETNUKE MODULE GENERATOR**  
**[WWW.DNNKICKSTART.COM](http://www.DNNKickStart.com)**

## Contents

What is DNN KickStart.....	4
Features.....	5
FAQ .....	6
Screenshots.....	7
Installation.....	8
Starting the installation .....	8
Requirements.....	8
Run the installation.....	8
Introduction .....	13
Overview tree.....	14
Log.....	15
Menu.....	16
Options .....	17
Default options .....	17
Folders.....	18
Primary code output folder .....	18
Project folder .....	19
Module definitions folder .....	19
WebProject folder.....	19
Database .....	20
Module.....	21
Libraries .....	21
How to create a module .....	23
Module meta data .....	26
Company.....	26
Module name.....	27
Description .....	27
Module version .....	28
Module scope.....	28
Fields .....	29
Sequence .....	29
Field name .....	29
Field type.....	29
Label .....	30
Length .....	30
Can be empty.....	30
Identity.....	30
Precision .....	30
Scale .....	30
Field lookup.....	31
Lookup source.....	31
Lookup text.....	31
Lookup value.....	31

Hidden.....	31
Help text.....	31
Required message.....	31
Default value.....	31
True text.....	31
False text.....	32
Can be sorted.....	32
Sort order.....	32
Format string.....	32
Custom group.....	32
Validation message.....	32
Validation text.....	32
Controls.....	33
Single record.....	34
Default web control.....	35
Output.....	36
Export.....	37
Export package.....	37
Export code.....	37
Export to DNN Web.....	37
Export as WebProject.....	37

## What is DNN KickStart

### **Create your own DotNetNuke modules**

Users of DotNetNuke can create their own modules and applications with DNN KickStart.

### **No programming experience needed**

You do not need any knowledge about programming, databases or other inside information. Just define the data fields you want to store in your application in DotNetNuke and DNN KickStart will create the module for you.

### **Instant ready**

This module is ready to use instantly. You can upload this module on the normal way you upload any DotNetNuke module. After uploading you can use your module on the page wherever and whenever you want.

## Features

DNN KickStart features are

- All code is generated, inclusive SQL etc to upload your module.
- Source code generated is complete, and can be modified if needed
- Define fields in a field entry form to create your module from.
- Source code is available.
- The program only needs DOT NET 2
- It is delivered with installer.
- Modules generated are tested for DNN 4 only.
- Works now for MS SQL Database (all versions).
- Create edit screens with data validation.
- Includes all data types you can use in the database
- Includes field validation for screens
- Label your own fields.
- You can generate VB.NET and C#
- You can also generate WebProjects for compiled Module delivery.
- Optional: Select tables from your database to create a DNN 4 module.
- Runs on Windows XP, 2000 and Vista

## FAQ

Here you can find the most asked question about DNN KickStart.

Q: Do I need to have DNN installed locally to use this program?

A: No, you do not need DNN locally to use this program

Q: Can I change my module after it is generated?

A: Yes you can. The sources code is created by the DNN KickStart and can be modified after generation.

Q: Does this program also imports my table data into the module?

A: No, it only creates a module from you table definition

Q: Do you have a C# version

A: Yes we do!. The code generated is compatible with the DNN Starter Kit module and generated as VB.NET 2005 code

Q: Can I access also other SQL databases beside the DNN Database?

A: Of course, this is one of the reasons to use this tool. All SQL Database can be used for Module generation.

Q: Can I create a module without a database?

A: Yes, with the latest release you can create your module, and you do not need any development environment or database anymore

Q: Do I need Visual Studio to edit my generated source code?

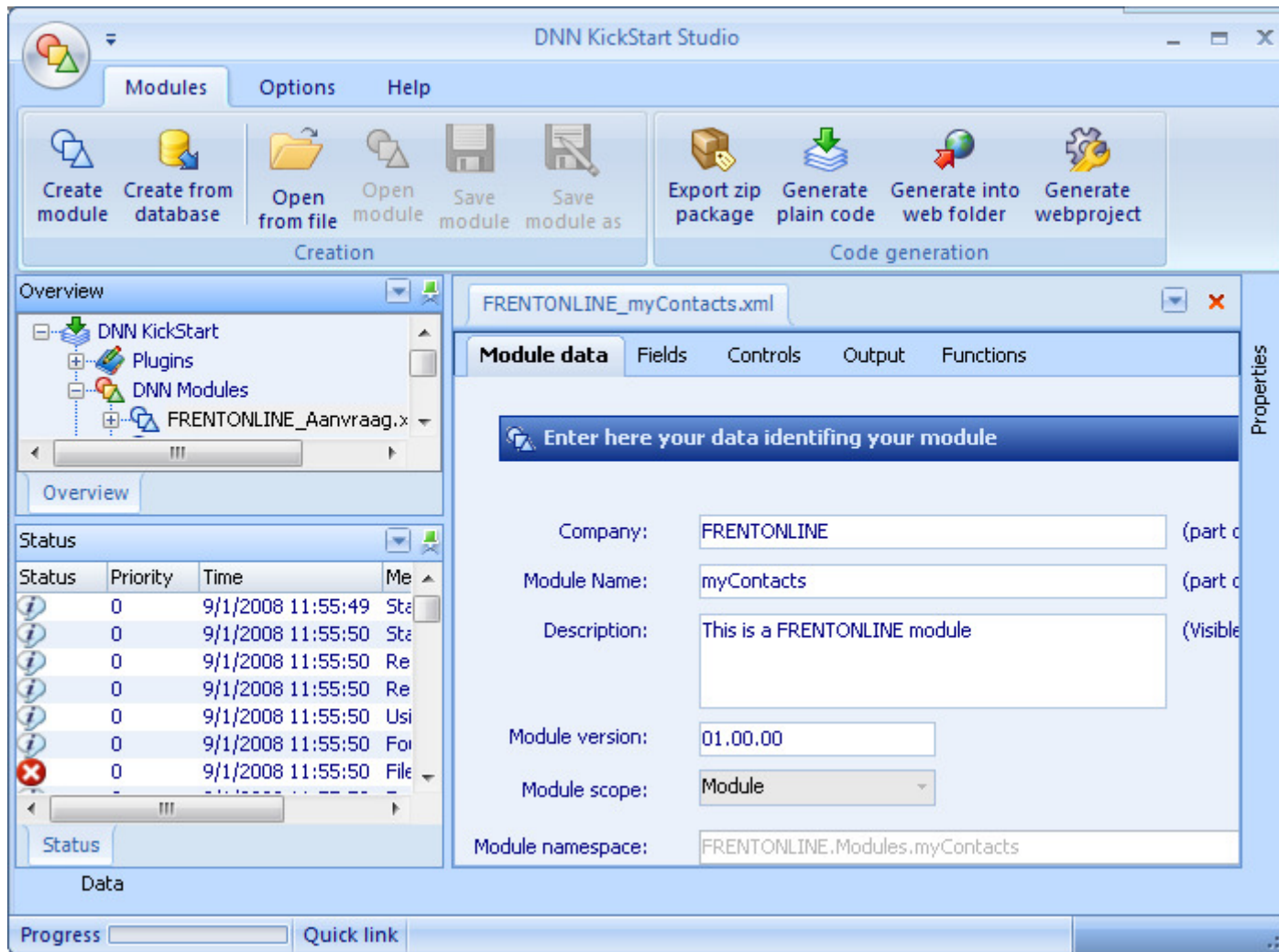
A: No, you do not need any development environment. The module is instant ready. But you are able to load the module code in Visual Studio 2005 or Web Developer 2005 Express to add, change or remove functionality.

Q: Can I deliver my module as PA (compiled version)?

A: Yes you can. Modules can be exported as Web Projects and compiled into PA (DLL's)

## Screenshots

Main overview screen DNN Kickstart



## Installation

### Starting the installation

---

You can start the installation by running the program **DNNKickStartSetup.exe**. This will start the installation wizard that will guide you through the installation of DNN KickStart.

### Requirements

---

- You should install the program with a user account with **Administrator** rights.
- About 20MB disk space is needed to install the program.
- You need to have DotNet 2.0 installed.
- The program needs Windows XP, Windows 2000 or Windows 2003 and Vista.

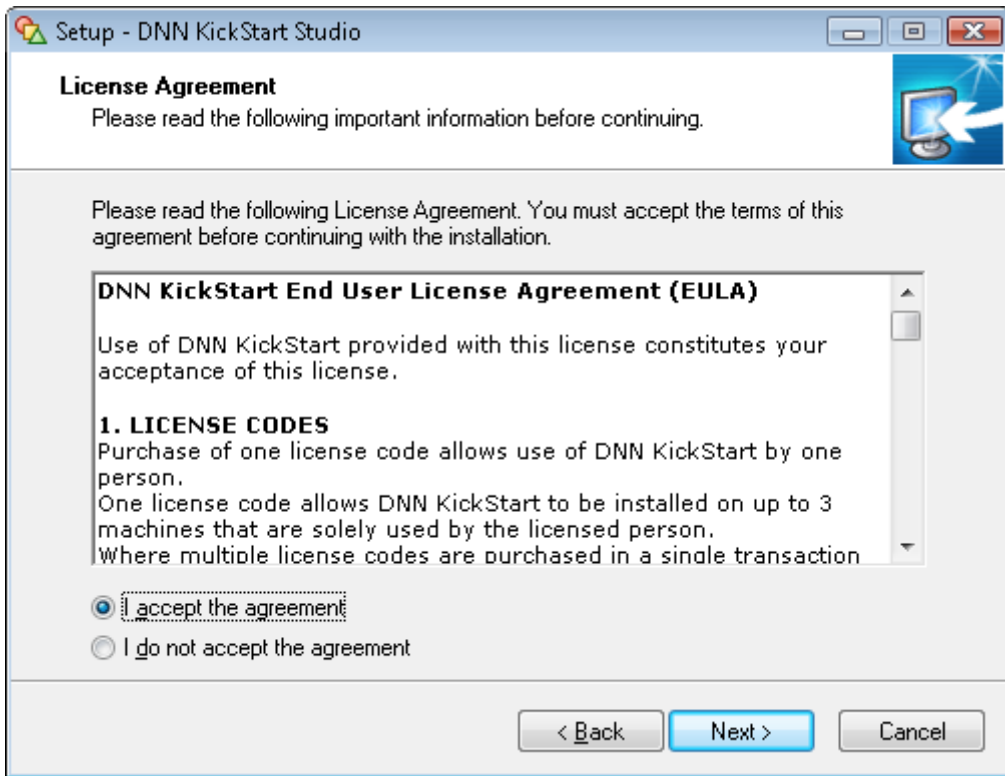
### Run the installation

---

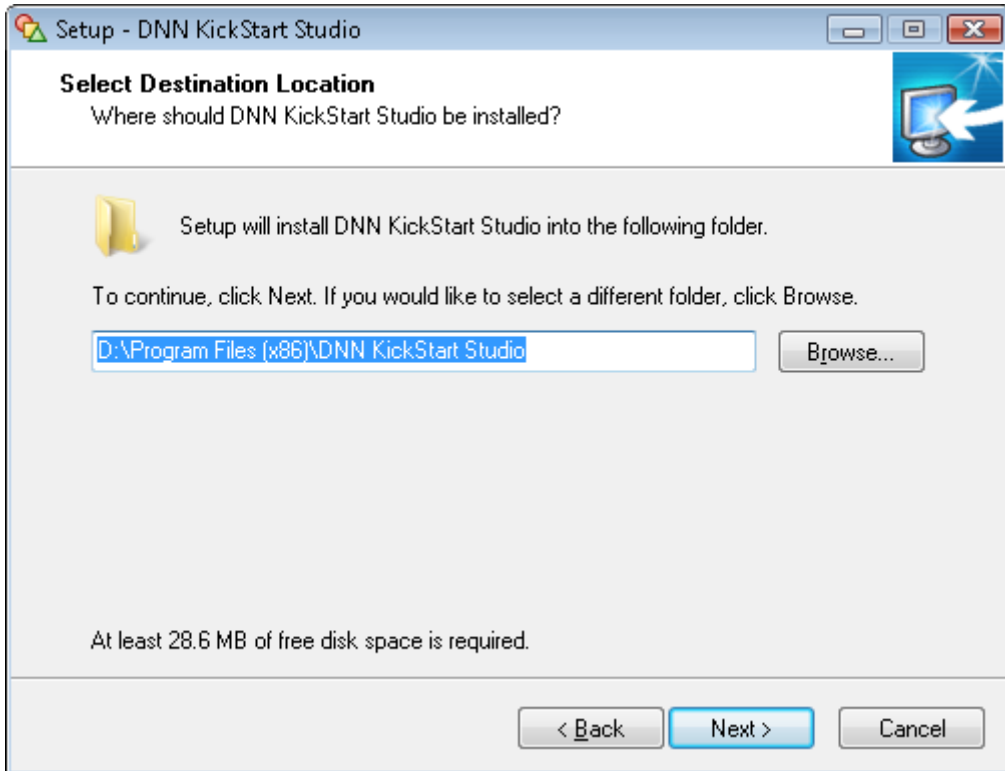


Click Next to continue

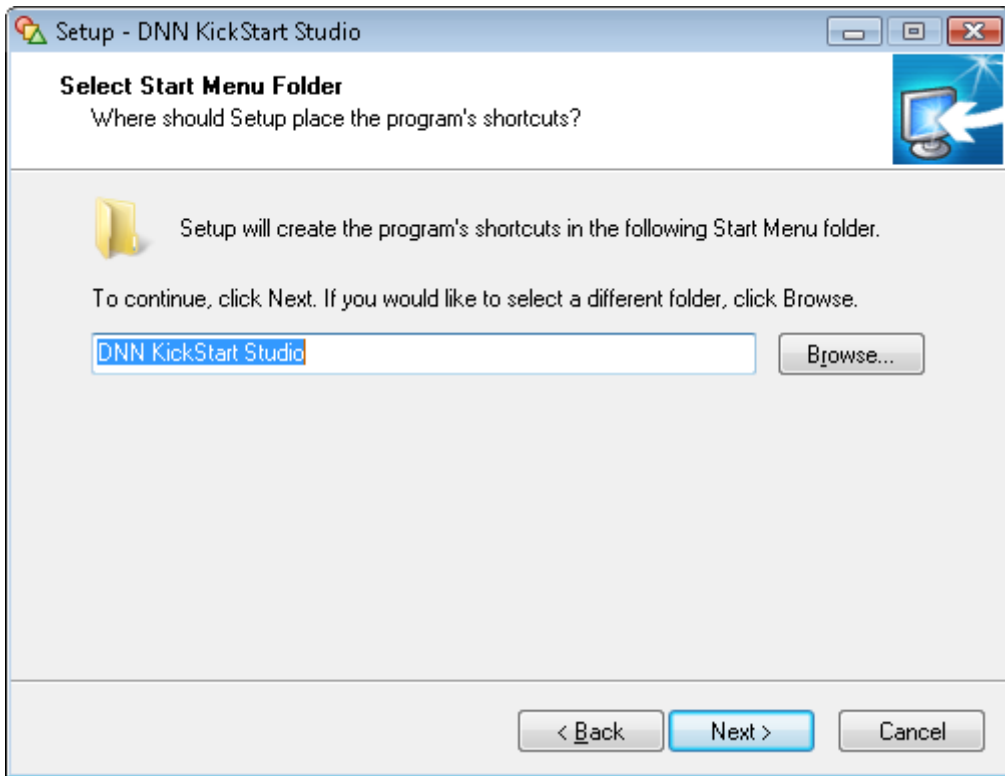




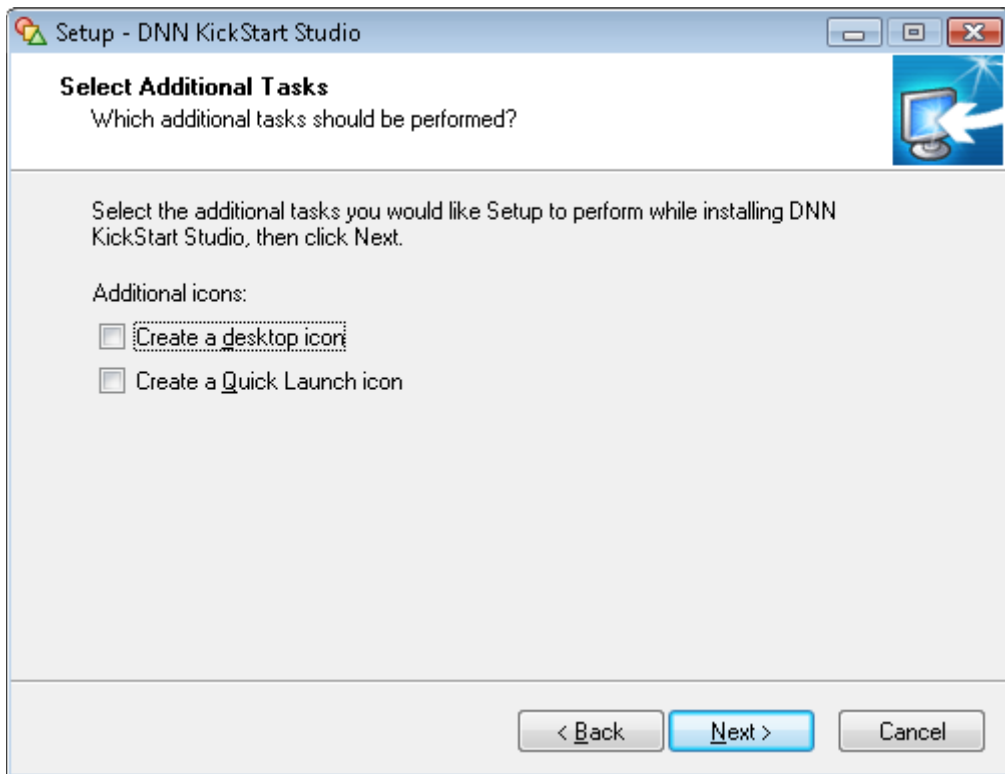
Accept the agreement.



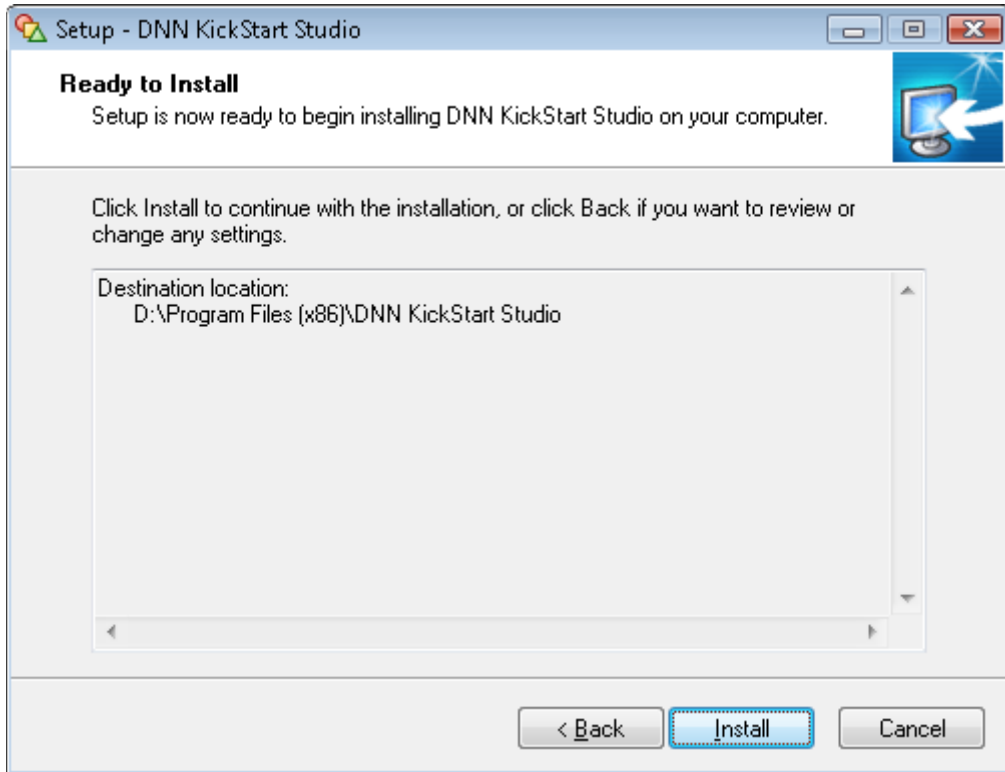
Select your folder you want to install to.



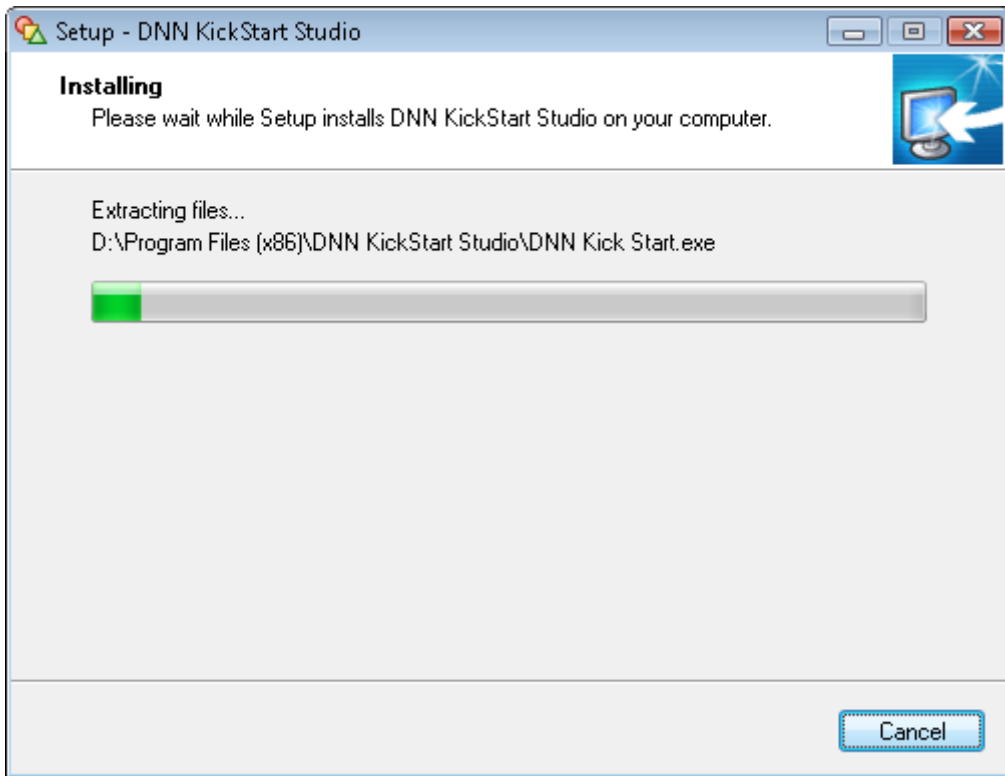
Enter the Start menu folder you want the program installed to.  
Click Next to continue.



You can create additional icons, like desktop and quick launch icons.  
Click Next to continue.



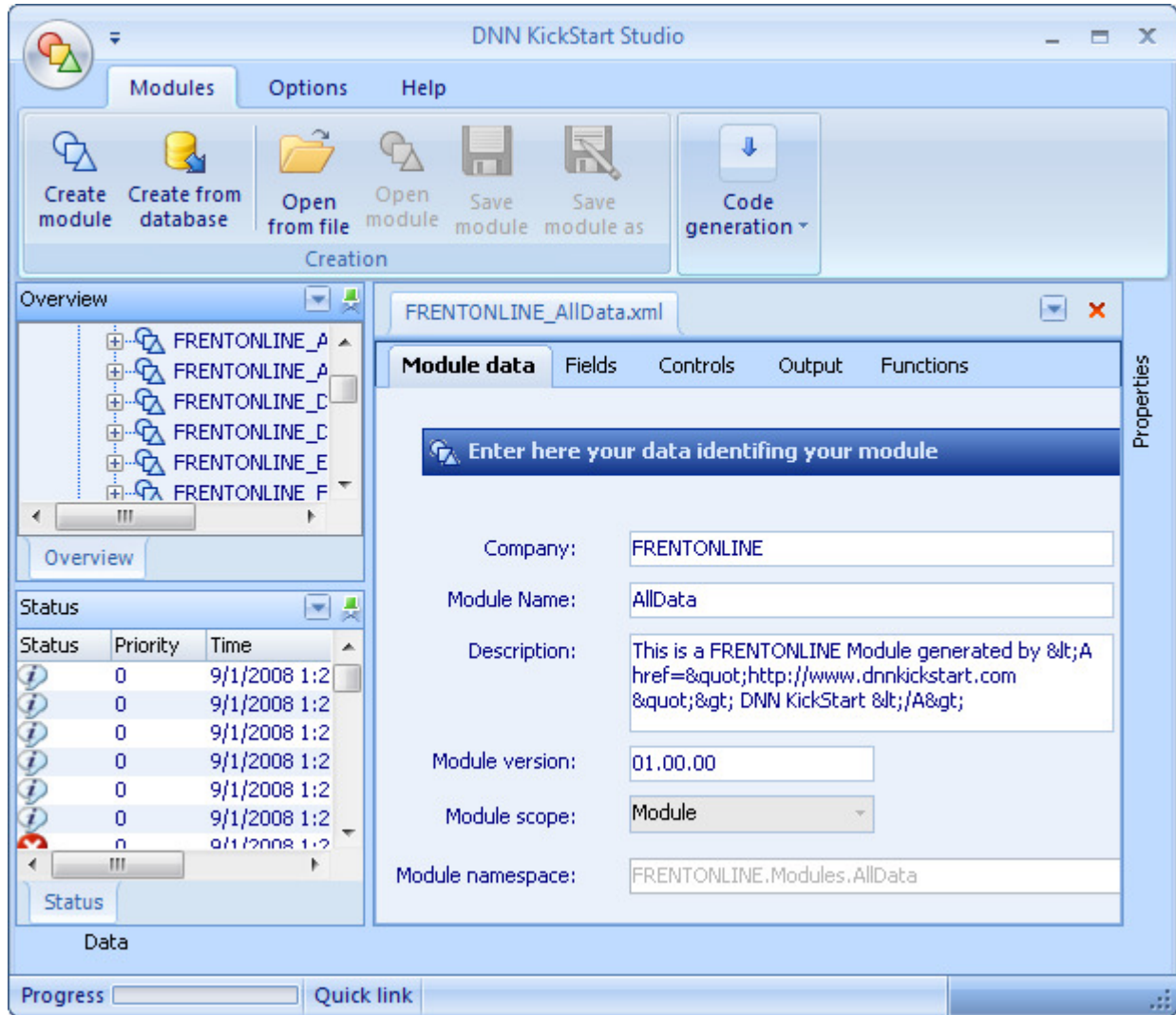
Click on Install to start the installation.



After this the program is installed and you can use it immediately by selecting Launch program.

## Introduction

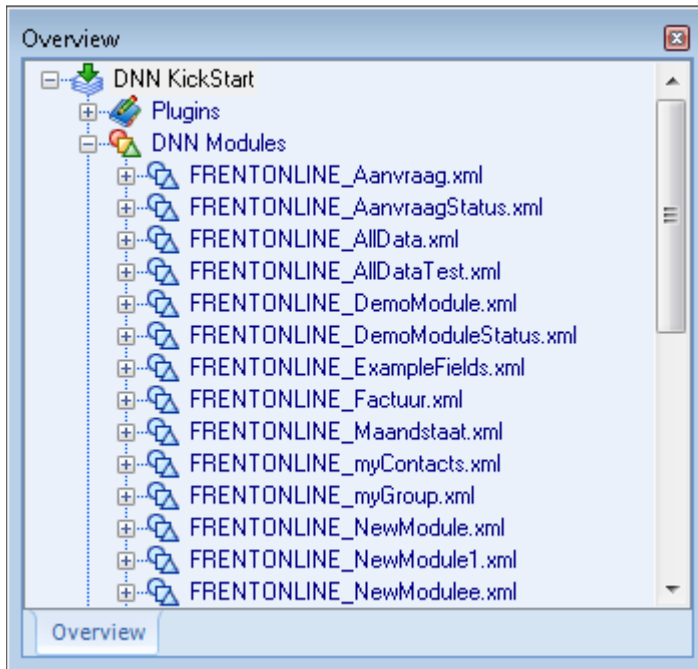
After launching the program you will be shown a screen that is the main user interface. This consists in Overview tree, log, menu and toolbar.



## Overview tree

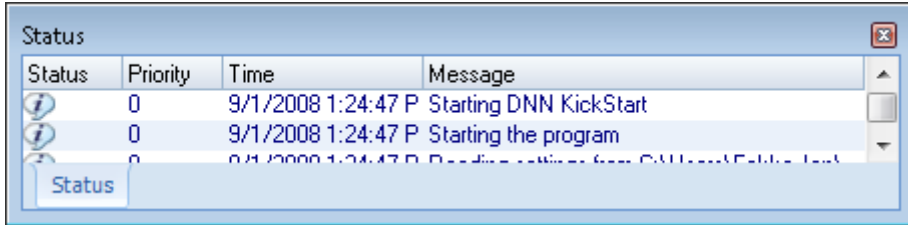
The overview tree displays all your stored module definition files. Default a definition file is installed called FRENTONLINE\_NewModule.xml.

The Overview tree presents the files as stored in the Module definitions folder under the menu Tools, Options.



## Log

In the log viewer you can see what kind of events are fired within the DNN KickStart Application. This can be useful for informational purposes.



## Menu

The menu present options that can be used to create modify or generate DotNetNuke modules.

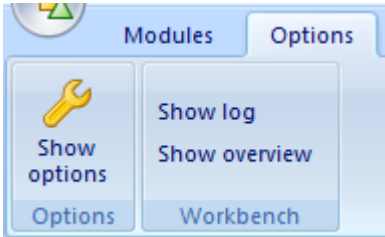




## Options

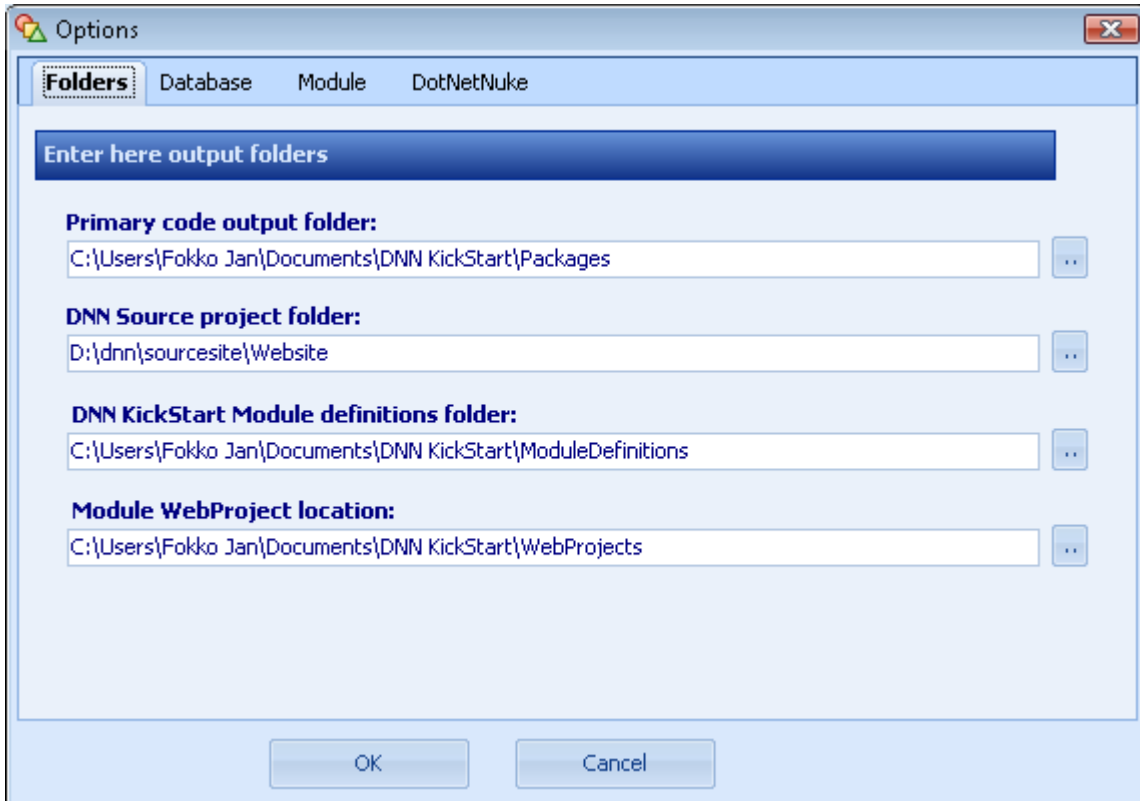
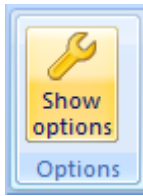
### Default options

Before starting creating modules, it is useful to set the default options for DNN KickStart. This save a lot of time when creating a new module, as module default will be set to your defaults.



## Folders

With Options you can set your default folders.



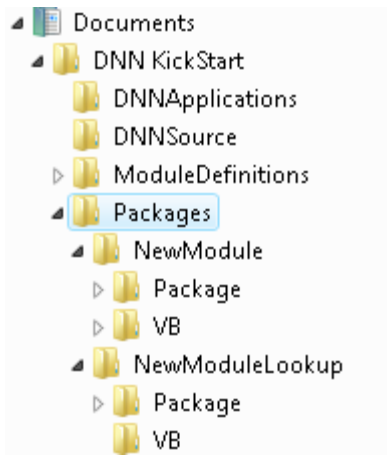
You can set a couple of default folders:

- Primary code output folder
- DNN Source project folder
- DNN Kickstart Module definitions folder
- Module WebProject location

### Primary code output folder

The output folder is the folder that contains the code when you choose Generate package. In the folder a subset of folders will be created for each generated module. These folders are:

- **Package** folder: <Output folder> \ Module name \ Package
- **VB** Code folder: <Output folder> \ Module name \ VB
- **CSharp** Code folder: <Output folder> \ Module name \ CSharp



### **VB Folder**

In the VB Code folder all files are generated for your module if you have chosen VB as language.

### **CSharp Folder**

The CSharp folder is equal to the VB folder, except that it contains the CSharp files when chosen to generate as CSharp module.

### **Package Folder**

The files will all be zipped into the Package folder into two zip files: one zip file contains the up loadable module; the other is an identical zip file for archiving purposes.

### **Project folder**

---

The project folder is the folder that contains the DotNetNuke folder. This folder will be used to create your module files in when you choose Export to DNN Web. The correct files will be placed into the **App\_Code** folder and **DesktopModule** folder.

Normally you choose for the project folder is: **<DotNetNuke Source> \ Website**

### **Module definitions folder**

---

The module definitions folder is the folder where you module files are stored. These are files only used by DNN KickStart. You never need them in your DotNetNuke module. These files contain all the definitions needed for DNN KickStart to create a module. All these files located here are also shown in the Overview tree in the Main screen.

Default the folder is **<DNN KickStart> \ Module definitions**

### **WebProject folder**

---


The web project folder is the folder where web projects will be generated. This will be the case if you select Export to WebProject.

A sub folder with the module name will be generated in the WebProject folder for each Module.

Default the folder is **<DNN KickStart> \ WebProject**

## Database

When generating modules from a database, it is useful to fill the database defaults. In this way you do not have to fill in these values every time.



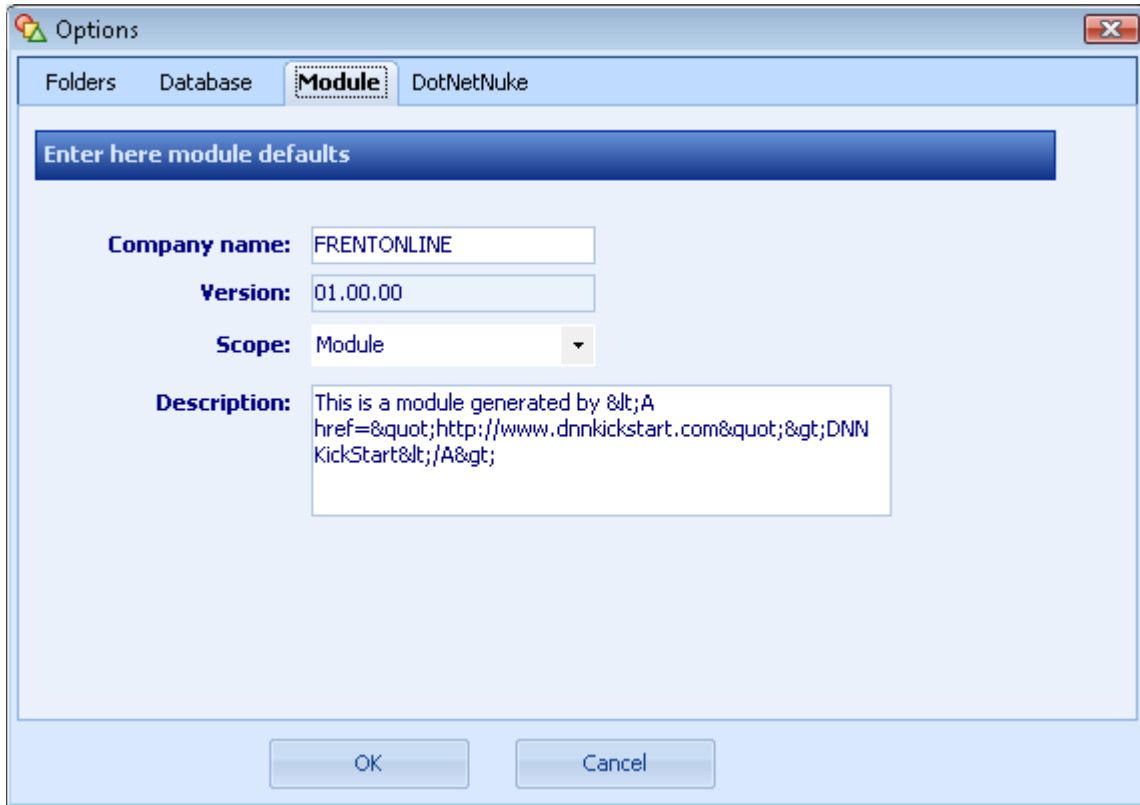
The image shows a screenshot of a Windows-style dialog box titled "Options". The "Database" tab is selected, and the "Module" is set to "DotNetNuke". A blue banner at the top of the dialog says "Enter here your database defaults". Below this, there are four input fields: "Database:" with the value "DotNetNuke", "Catalog:" with the value "DotNetNuke", "User:" which is empty, and "Password:" which is empty. At the bottom of the dialog are "OK" and "Cancel" buttons.

You can use the following defaults:

- **Database**, The database (host \ instance name) from your SQL server
- **Catalog**, the catalog you use
- **User**, the default user you use to logon
- **Password**, the password for the default user

## Module

The module defaults are used when creating a new module. Your new module will contain the following values you set here as default.



The screenshot shows a Windows-style dialog box titled "Options" with a close button in the top right corner. The dialog has four tabs: "Folders", "Database", "Module" (which is selected and highlighted), and "DotNetNuke". Below the tabs is a blue header bar with the text "Enter here module defaults". The main area contains four fields: "Company name:" with a text box containing "FRENTONLINE"; "Version:" with a text box containing "01.00.00"; "Scope:" with a dropdown menu showing "Module"; and "Description:" with a text box containing the text: "This is a module generated by &lt;A href='\"http://www.dnnkickstart.com\"'&gt;DNN KickStart&lt;/A&gt;". At the bottom of the dialog are two buttons: "OK" and "Cancel".

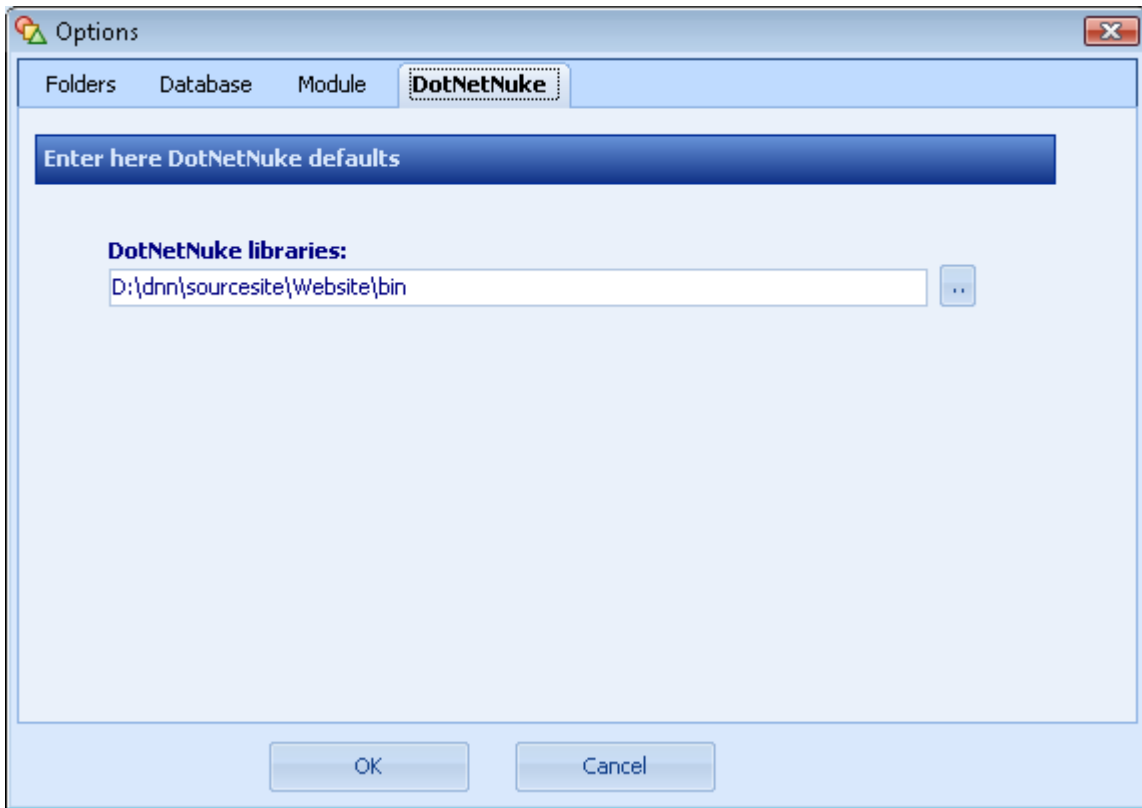
You can use the following defaults:

- **Company name**, the name of your company
- **Version**, the version of the module
- **Scope**, the scope of the module
- **Description**, the description of your module

## Libraries

The libraries folder is the folder where the compiled DotNetNuke libraries are. This is needed only when creating a WebProject. The Visual Basic project needs a reference to the DotNetNuke libraries, so this folder will be included in the WebProject.

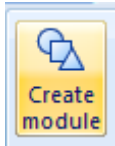
The most important DLL, DotNetNuke.DLL is located in the DotNetNuke Libraries folder.



## How to create a module

In this section we explain how to create your first module.

First we start by creating a new Module. Choose **Create Module** to create a new Module.



You will be shown the Module screen.

This is the screen that displays the data for your module.

A screenshot of a web application window titled "new:". The window has a tabbed interface with "Module data" selected, and other tabs for "Fields", "Controls", "Output", and "Functions". A blue banner at the top of the form area says "Enter here your data identifying your module". Below this, there are several input fields: "Company:" with the value "FRENTONLINE", "Module Name:" with "NewModule", "Description:" with a text area containing "This is a module generated by &lt;A href='\">http://www.dnnkickstart.com&quot;&gt;DNN KickStart&lt;/A&gt;", "Module version:" with "01.00.00", "Module scope:" with a dropdown menu set to "Module", and "Module namespace:" with "FRENTONLINE.Modules.NewModule".

Enter the data for your module, like Company, Modulename etc.

See [Module meta data](#)

### Module data

Fill the following values for **Module data**:

Company: *YourCompanyName*

Module name: *YourModuleName*

Description: *The description of your module*

Module version: default: *01.00.0*

Module scope: default: *Module*

### Fields

Fill the fields section with your data fields

new:					
Module data					
Fields					
Controls					
Output					
Functions					
Details	Sequence	Field Name	Field type	Label	
▶	0	MyID	Identifier	My ID	
	0	Name	Text	Name	
	0	ModuleId	DNN Modul	ModuleI	
	0	PortalId	DNN Portal	PortalId	
	0	UserId	DNN UserI	UserId	
*					

Add the fields you need.

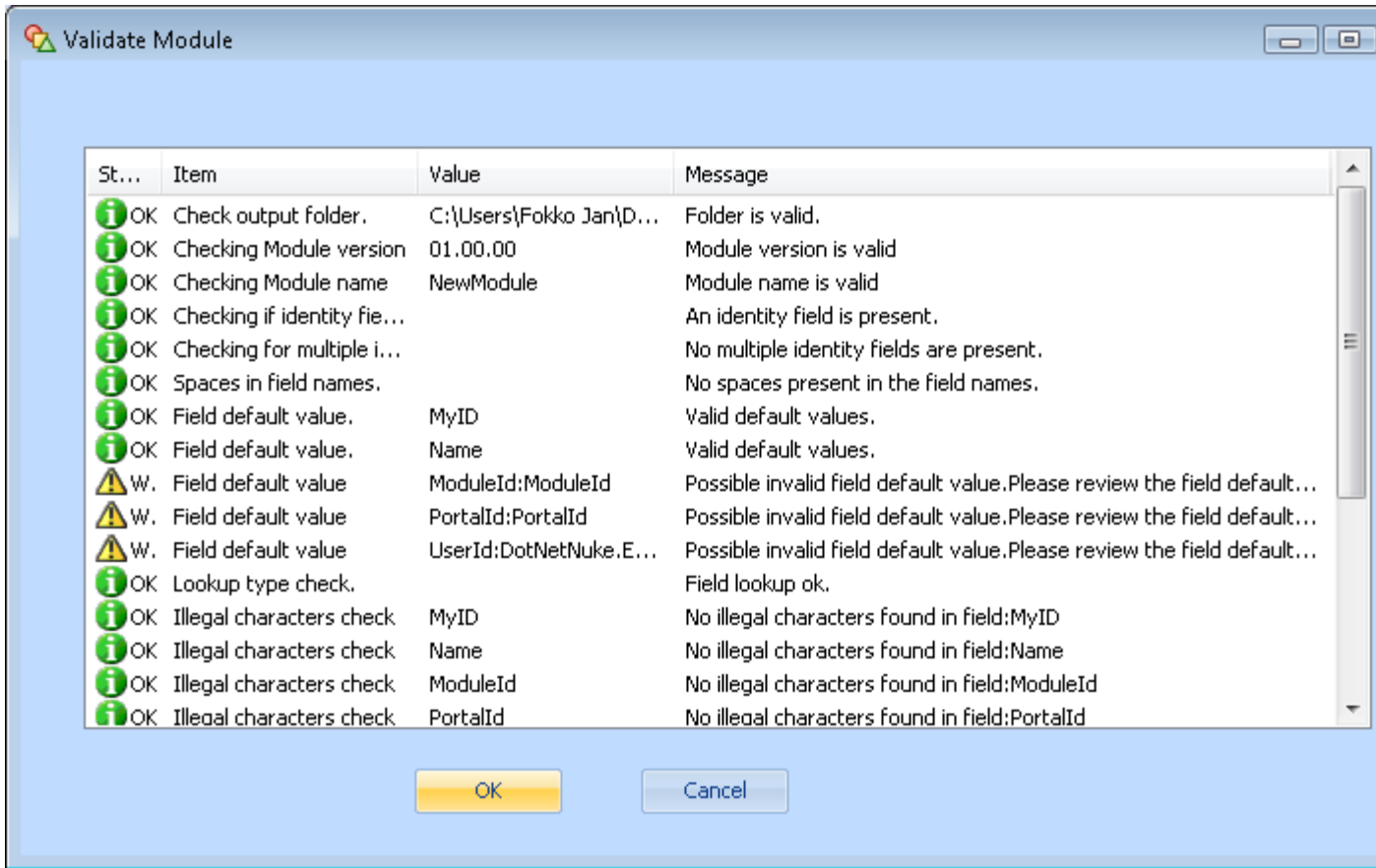
**Export the package**

Export the module into a zip package ready for upload.

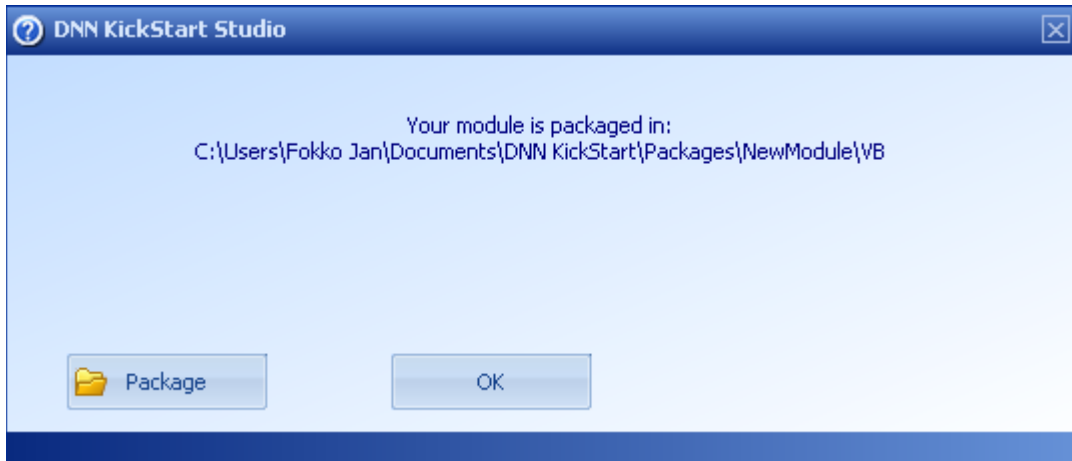


You will receive the following message:





Click OK to continue.



Your package is ready for deployment in a zip file.  
 This zip file can be uploaded into your DotNetNuke Website.  
 That's all there is!

## Module meta data

The first tab shows the module meta data. These are the major meta data fields:

- Company
- Module name
- Description
- Module version
- Module scope

The screenshot shows a web-based interface for configuring module metadata. At the top, there are tabs for 'Module data', 'Fields', 'Controls', 'Output', and 'Functions'. The 'Module data' tab is selected. Below the tabs is a blue header bar with the text 'Enter here your data identifying your module'. The main area contains several input fields:

- Company:** A text box containing 'FRENTONLINE' with a tooltip '(part of the na...)'.
- Module Name:** A text box containing 'NewModule' with a tooltip '(part of the na...)'.
- Description:** A text box containing 'This is a module generated by &lt;A href=&quot;http://www.dnnkickstart.com&quot;&g t;DNN KickStart&lt;/A&gt;' with a tooltip '(Visible in Modu...)'.
- Module version:** A text box containing '01.00.00'.
- Module scope:** A dropdown menu with 'Module' selected.
- Module namespace:** A text box containing 'FRENTONLINE.Modules.NewModule'.

## Company

This is the name of the company that delivers the module. Most of the cases the name will be your company.

### Function

- The company field is used in the namespace of your code, to create unique namespaces.

```
NewModuleInfo.vb Start Page
Imports System
Imports System.Configuration
Imports System.Data

Namespace FRENTONLINE.Modules.NewModule
    Public Class NewModuleInfo
```

- It is also used as pre string for your table and stored procedures names.

```
create procedure dbo.[FRENTONLINE_GetNewModule]
```

```
    @ModuleId int,  
    @MyID int
```

```
as
```

### Restrictions

The company field has the following restrictions:

- May not contain spaces  
Cannot be longer as 128 characters

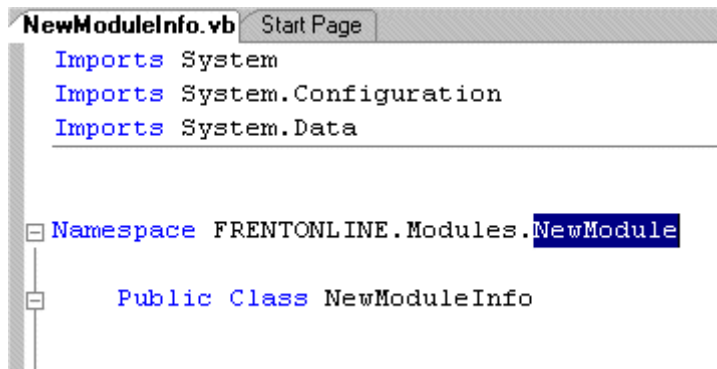
## Module name

---

This is the name of the module.

### Function

- The module name field is used in the namespace of your code, to create unique namespaces.



```
NewModuleInfo.vb Start Page  
Imports System  
Imports System.Configuration  
Imports System.Data  
  
Namespace FRENTONLINE.Modules.NewModule  
  
    Public Class NewModuleInfo
```

- It is used as a qualifier in the stored procedures names and table creation

```
if not exists (select * from dbo.sysobjects where  
BEGIN  
    CREATE TABLE dbo.[FRENTONLINE_NewModule]  
    (  
        [ModuleId] [int] NOT NULL,  
        [MyID] int IDENTITY(1,1) ,  
        [Comment] char(8)  
    )
```

### Restrictions

The module name field has the following restrictions:

- May not contain spaces  
Cannot be longer as 128 characters

## Description

---

This is the description of the module, as it will be shown in the module list.

### Function

- The description is only used in the module overview list in DotNetNuke

### Restrictions

The description field has the following restrictions:

Cannot be longer as 128 characters

## Module version

---

### Description

This is the version of the module, as it will be shown in the module list.

### Function

- The description is used in the module overview list in DotNetNuke
- The module version will be checked and used to run the SQL code to create tables and procedures

Name	Size
01.00.00.SqlDataProvider	12 Ki
01.00.00.SqlDataProvider.Plain.sql	5 Ki

### Restrictions

The module version field has the following restrictions:

- It has to comply to the format xx.xx.xx where x stands for a number value

Default the version is 01.00.00

## Module scope

---

### Description

This is the scope of the module.

### Function

- The scope is used to display and edit data only within a certain scope
- The scope can be module, portal or user
- If the scope is **module**, data will be stored for every module instance on every page separately. If you place a module on a tab, this module stores its data for this module only.
- If the scope is **portal**, data will be stored once for the portal. Every module instance on every page shows the data as it is shown in other modules. If you place a module on a tab, this module shows the same data wherever in the portal.
- If the scope is **user**, data will be stored once for the portal **user**. Every module instance on every page shows the data as it is shown in other modules for the logged on user. If you place a module on a tab, this module shows the same data wherever in the portal.

### Restrictions

The module scope has the following restrictions:

It can be either portal, user or module

---

DNN KickStart Manual

Page 28

## Fields

The second tab shows the module field data. These are fields that are used within the module.

Module data	Fields	Controls	Output	Functions
Details	Sequence	Field Name	Field type	Label
	0	MyID	Identifier	My ID
▶	0	Name	Text	Name
	0	ModuleId	DNN Modul	ModuleId
	0	PortalId	DNN Portal	PortalId
	0	UserId	DNN UserI	UserId

The number of entries for the field data will be expanded for all field properties. With DNN KickStart the base of your module consists of fields. These fields can be entered manually or imported from a database table.

---

### Sequence

#### Description

The sequence of field as it is displayed in the screen

#### Function

This field is only used in DNN KickStart. However you can use this field to apply the sequence of the fields as they are shown on the screen. By sorting on sequence number you choose, you determine the sequence of fields shown on the screen.

#### Restrictions

You can use any positive number you want. Sequence numbers do not need to be incremental. They can be the same, or you can skip numbers if you want.

---

### Field name

#### Description

The field name as stored in the table

#### Function

This field is the field name as it would be stored in the database.

#### Restrictions

The normal MS SQL restrictions count for field names. However you cannot use spaces in the field name. (You can with field labels instead).

---

### Field type

#### Description

Data category of the field

#### Function

This indicates the data category of the field. Based upon the data category the date type, restrictions and formatting will be determined.

You can choose the following field types:

- Custom, indicates that you use your **own** data types and formatting
- Date, indicates that you use **date** type and formatting
- Text, indicates that you use **text** type and formatting
- Identifier, indicates this field is the identifier field
- Image, indicates that you use **image** type and formatting (not used for now)

- Lookup, indicates that you use **lookup** type and formatting
- Money, indicates that you use **money** type and formatting
- Time, indicates that you use **time** type and formatting
- Yes/No, indicates that you use **Boolean** type and formatting

### **Restrictions**

You are restricted to list above

## **Label**

---

### **Description**

The label as shown in the screens for the field

### **Function**

This is the label as is shown in the edit and view screens. The labels will be added to the resource file. You can copy the resource file to create Multilanguage screens

### **Restrictions**

No restrictions

## **Length**

---

### **Description**

The length of the field

### **Function**

The length you indicate will be used by settings the length of the fields in the database. Also the edit box for the field has this length restriction.

### **Restrictions**

No restrictions

## **Can be empty**

---

### **Description**

Indicates if the field can contain null values

### **Function**

If this property is enabled, you can store null values in this field

### **Restrictions**

Can be enabled or not enabled

## **Identity**

---

### **Description**

Indicates that the field is used as the unique identifier (Auto numbering)

### **Function**

Only one field can be the identifier field. This is the field that indicates the unique id of a record

### **Restrictions**

Only one field can be the identity field.

## **Precision**

---

### **Description**

Only used for numeric fields

## **Scale**

---

### **Description**

Only used for numeric fields

## Field lookup

---

### Description

Indicates if the field is a lookup (combo list), either None or Lookup

### Function

If set to Lookup, this field will create a combo list in the edit screen to retrieve data from another data source (table). The code to connect and retrieve the data will be generated.

You can refer to default DNN tables, but also refer to your own created DNN KickStart modules.

### Restrictions

If set to Lookup the lookup source must be also exist in your DotNetNuke portal on delivery of the module.

## Lookup source

---

### Description

The table that contains the lookup values

### Function

This is the table the lookup data is retrieved from

### Restrictions

The table needs to be present in the DNN portal on delivery of the module.

## Lookup text

---

### Description

The text shown in the lookup up

## Lookup value

---

### Description

The value stored in the field to be lookup up

## Hidden

---

### Description

Indicates that SQL and other code is generated for the field, but is not shown in the screens

## Help text

---

### Description

Custom help text to be shown for this field if user press help icon

## Required message

---

### Description

The message shown if this is a required field and the user did not enter a value in the edit screen.

## Default value

---

### Description

The default value to be stored in the field

## True text

---

### Description

Text to be shown if the field is a Boolean and contents is true in the view screen

## **False text**

---

### **Description**

Text to be shown if the field is a Boolean and contents is false in the view screen

## **Can be sorted**

---

### **Description**

Indicates if the field can be sorted if in a data grid (not used now)

## **Sort order**

---

### **Description**

Indicates the sorted order if the field can be sorted (not used now)

## **Format string**

---

### **Description**

Sets the string for formatting your data

## **Custom group**

---

### **Description**

Is used to sort your data in DNN KickStart. Will not be used in the generated code

## **Validation message**

---

### **Description**

Sets the validation error message the user gets when entering invalid data

## **Validation text**

---

### **Description**

Set the validation text the user receives when entering data



## **Controls**

The control data can be set to indicate what control behavior the screens must have.

## Single record

The single record option is added to show only one record you can view and one record you can add. This can be useful in combination with the scope **user** to create personal pages with data that can be used as user settings.



The screenshot shows a web interface with a navigation bar at the top containing tabs for 'Module data', 'Fields', 'Controls', 'Output', and 'Functions'. The 'Functions' tab is active. Below the navigation bar is a blue header with a play icon and the text 'Enter here your functions used for your module'. The main content area is light blue and contains a list of six checkboxes, each followed by a label:

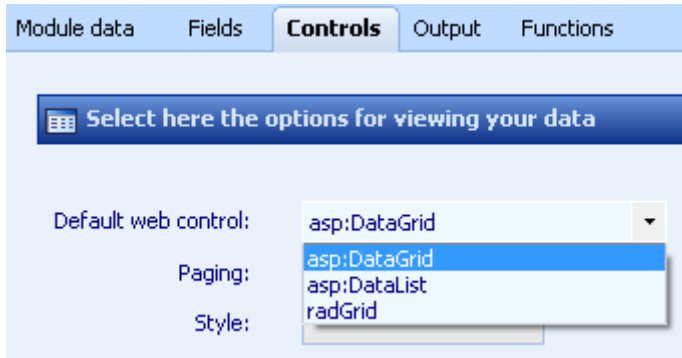
- Searchable
- Portable
- Upgradable
- Only show and edit a single record
- Generate initial record if no records are present
- Create Export to Excel button

If selected only one record can be added and viewed

## Default web control

The default Web Control can be used to select the control visible for viewing. You can select the following controls:

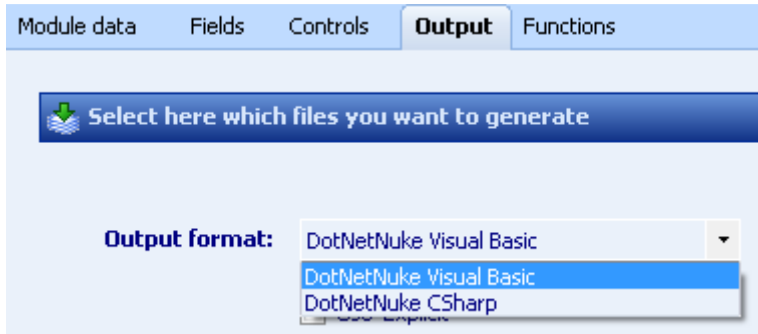
- asp:DataGrid
- asp:DataList



If you select the asp:DataGrid, the data will be displayed in a datagrid. If you select asp:DataList, the data will be shown in a list.

## Output

The output tab indicates what kind of output needs to be generated. You can select to generate Visual Basic code or C# code.



All screens, code and items will be generated in the language you choose.

## Export

### Export package

---

This command is used to create a package for your definition. The package will be generated in the folder that is chosen in the default export folder.

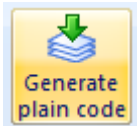
The package generated is immediate available and ready for upload to your DNN portal.



### Export code

---

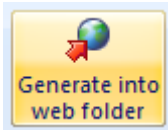
This command is used to create a code for your definition. The code will be generated in the folder that is chosen in the default export folder.



### Export to DNN Web

---

This command is used to create a code for your definition into a DNN Website. The code will be generated in the folder that is chosen in the default project folder.



### Export as WebProject

---

This command is used to create a code for your definition into a WebProject. The code will be generated in the folder that is chosen in the default Web project folder.

With this you can create a PA of you module.

