



TierDeveloper™
Rapidly Develop Your Middle-Tier Objects!

Fast Track Guide for .NET

Learn installing and configuring TierDeveloper followed by developing and deploying .NET components with IIS by a never been so easy process. Copyright © AlachiSoft All rights reserved

Table of Contents

About Alachisoft.....	iii
AlachiSoft Headquarters.....	iii
Customer Support.....	iii
Introduction.....	iv
About this Guide.....	v
1 Prerequisite of TierDeveloper.....	1
1.1 Prerequisite Environments.....	1
1.1.1 Build Environment.....	1
1.1.2 Production Environment.....	1
2 Getting started with TierDeveloper.....	1
2.1 System Requirements.....	1
2.2 Download TierDeveloper.....	2
2.3 Installing TierDeveloper.....	2
3 Typical Process of Using TierDeveloper.....	6
3.1 Create or Use Sample Project.....	6
3.2 Generate and Build Source Code.....	14
3.3 Generate Web Application.....	14
3.4 Generate Documentation.....	14
3.5 Test .NET Components with Web Application.....	14
4 Additional Resources.....	15

About Alachisoft

AlachiSoft, a California based company located in Bay Area, is in the business of providing software development productivity tools. The first tool in this line is "TierDeveloper" that helps developers of Enterprise Applications cut down their development time and cost, through its object-to-relational mapping and code generating capabilities. TierDeveloper has been designed to become a Rapid Application Development (RAD) environment for customers who want to design and develop Enterprise systems in Microsoft technologies. AlachiSoft is also a current partner of Visual Studio Industry Partner Program.

AlachiSoft Headquarters

AlachiSoft

231 Market Place, Suite 522
San Ramon, CA 94583

<http://www.alachisoft.com>

Customer Support

Technical Support: support@alachisoft.com
US Toll Free: 1-(800) 253-8195
Intl: 1-(925) 236-3830
Fax: 1-(925) 886-2417
Email: sales@alachisoft.com

You might just wish to tell us how much you liked TierDeveloper or have ideas about how this whole process could be improved, so you can e-mail us at:
support@alachisoft.com

Introduction

TierDeveloper is an object-to-relational mapping and code generation tool that generate .NET business and data objects, ASP.NET apps, and Windows Forms apps. TierDeveloper gives the user an opportunity to define Data Objects that are mapped to one or more database tables that help to extract information as required in the shortest possible time. You can define your Data Objects using existing database schemas, generate Data Object source code, build/deploy the Data Objects and test the Data Objects with an Application, which is readily generated.

To gain full advantage of TierDeveloper, it is required to be familiar with the .NET Architecture, programming concepts of C#/VB.NET, IIS and MTS, Oracle/DB2/SQL Server Environment and Microsoft Access and above.

About This Guide

Welcome to the Fast Track Guide for .NET edition of TierDeveloper. You will find that TierDeveloper is not only easy to use but it is also a vital and powerful code-generating tool for visually creating professional Data Objects.

The purpose of this guide is to help you get started with TierDeveloper with utmost ease. It is designed to provide you with guidance for having the right configurations and environment for working with TierDeveloper. Moreover, it guides you through setting up TierDeveloper and then takes you through a cycle of creating a project. It not only allows you to generate code, using TierDeveloper but provides a mean to evaluate it very quickly as well.

If you still come across any problem then please report to the support team and we will get back to you as soon as possible.

1 Prerequisite Environments

As a prerequisite to install TierDeveloper, you need to have the following working environments and settings configured on your system as explained below.

In order to install/use TierDeveloper, you need to have configurations and environment according to the extent to which you are to utilize the capabilities of TierDeveloper. You can either deploy the applications in a production environment. Further requirements for these two options are skimmed below.

1.1 Build Environment

This is where you install TierDeveloper and use it to generate source code. You need connectivity to database, if you are going to create a new TierDeveloper project. However, if you are only using sample project, you can technically get by without a database and still generate source code. Nevertheless, you have to modify the Sample Project Settings to specify your database server connectivity information anyway. Here is what you need in this environment:

Compulsory:

- Windows NT4.0/Windows 2000 or up.
- Database connectivity provider such as OLEDB.
- .NET FrameWork.

Optional:

- Microsoft Visual Studio .NET 2003/2005 (C# and VB.NET)
- Visual Studio Integration Module required only if VS .NET is installed.
- Microsoft Internet Information Server (IIS).

1.1 Production Environment

This is where you deploy the final application for use. Here you do not require the installation of TierDeveloper. The requirements are:

- Windows NT4.0/Windows 2000 or up.
- .NET FrameWork.

2 Getting Started with TierDeveloper

2.1 System Requirements

TierDeveloper requires the following system requirements:

- Windows NT 4.0/Windows 2000 or above
- 64 MB RAM (minimum)

- 15 MB hard disk space

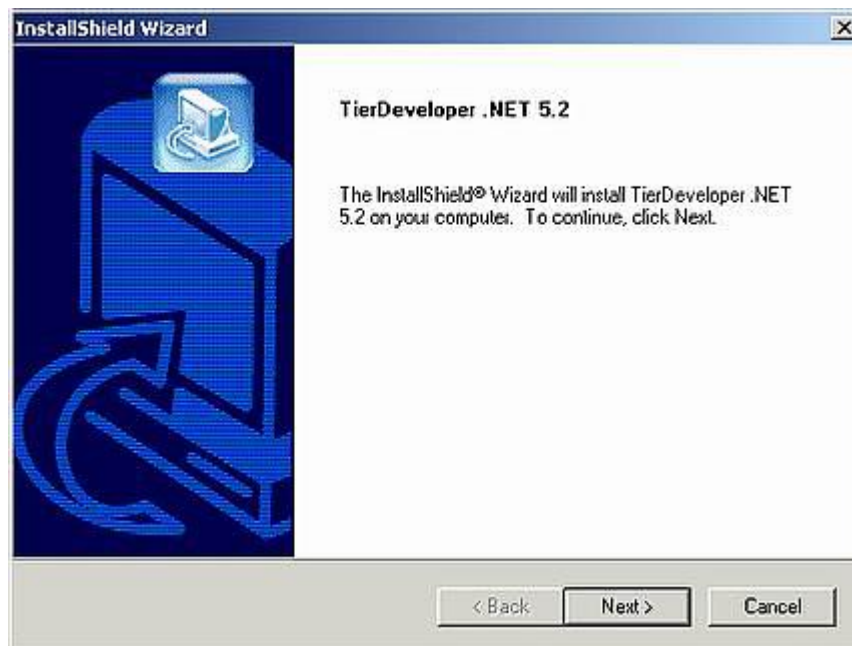
2.2 Download TierDeveloper

TierDeveloper is available as TierDeveloper .NET Enterprise Edition (for both VB.NET and C#), which can be downloaded from <http://www.alachisoft.com/download.html>. You are asked to fill a license agreement form. After accepting the agreement, you can choose TierDeveloper Enterprise (C# and VB.NET) to download its exe. The evaluation key for it is sent at the e-mail address provided in the license agreement form.

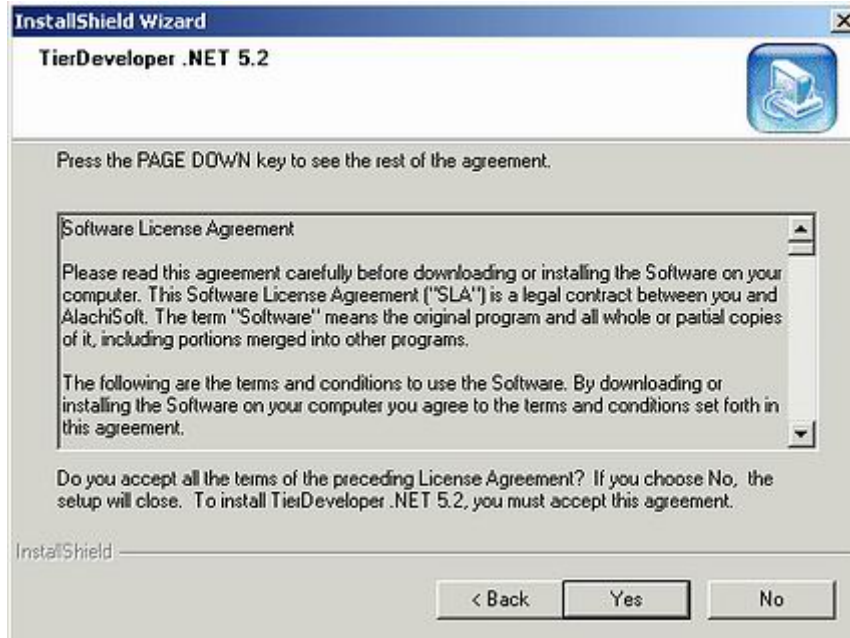
2.3 Installing TierDeveloper

TierDeveloper installation is very simple, consisting of a single executable file. This installation includes samples and documentation. For installing TierDeveloper on your system:

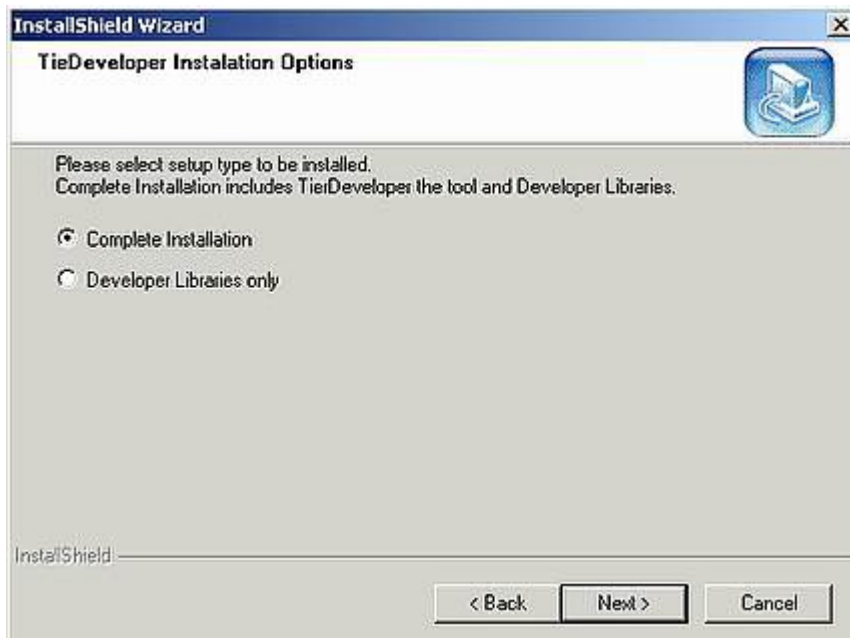
- 1- Run "TierDevNetEnt.exe" file and click "Next".



- 2- In the "Software License Agreement" dialog, read the text and click "Yes" if you agree to the terms. (Clicking "No" exits the installation procedure.)



3- The next dialog is "TierDeveloper Installation Options".

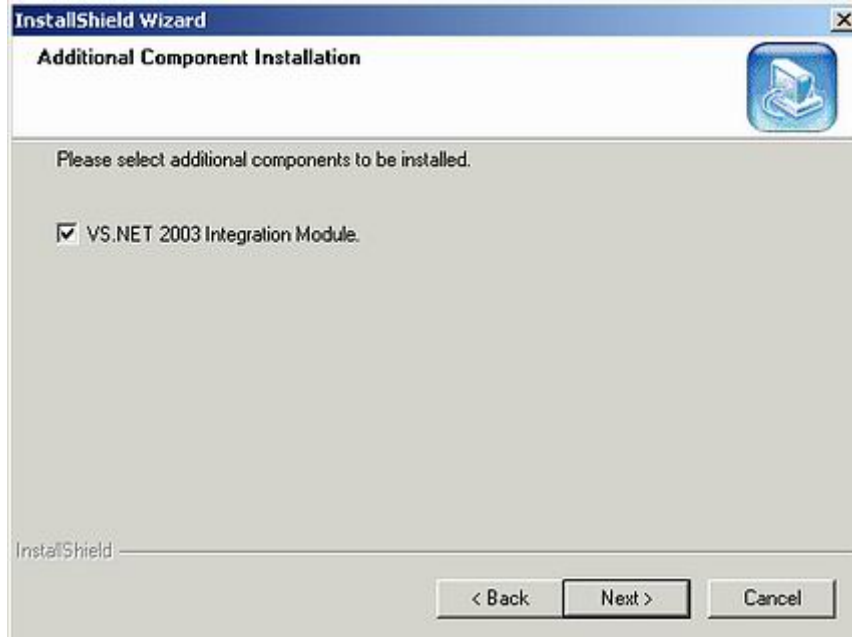


- **Complete Installation:** TierDeveloper along with its libraries is installed.
- **Developer Libraries only:** Only Libraries are installed.

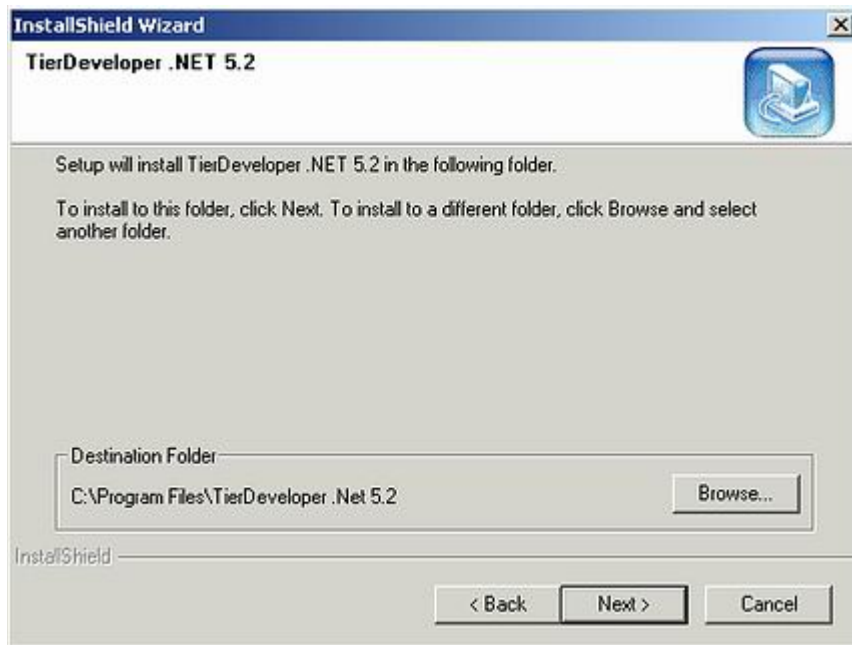
After making your choice, click next to move on.

4- In the "Additional Component Installation" window, select the "TierDeveloper VS.NET Integration Module" option if you want to integrate TierDeveloper with VS .Net IDE. You can integrate TierDeveloper with VS.NET 2005 as well as VS.NET

2003.



5- The next dialog allows you to choose the location where you wish to install TierDeveloper. You can browse to choose a location other than the default.

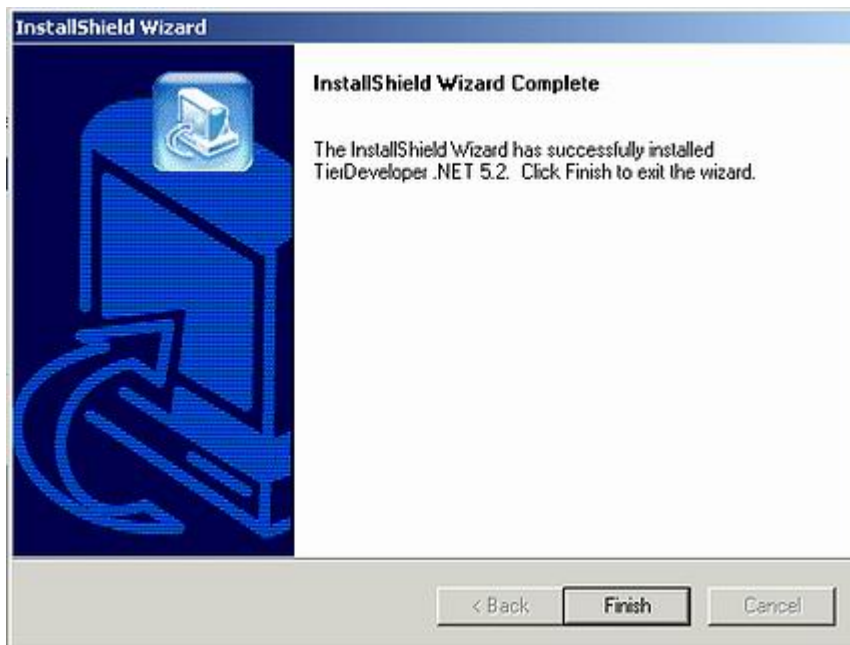


6- Next dialog allows you to choose the folder name based on which an icon is placed in the Programs Folder. You can type a different name as well besides the default.



7- Upon clicking "Next", TierDeveloper installation starts and the status is shown as percentage installed.

8- In the next window, click "Finish" to exit the installation procedure.



TierDeveloper is bundled with a sample project that helps you to evaluate the software in a shorter time. The sample projects are located in the "Samples\Projects" folder in the TierDeveloper's root directory.

NOTE: If you have downloaded the trial version of TierDeveloper then you receive an

evaluation license key via email. While on the other hand if you have purchased TierDeveloper, you get another license key that allows you to use TierDeveloper with its complete functionality un-locked.

3 Typical Process of Using TierDeveloper

Now that you are familiar with the prerequisites that you require in order to use TierDeveloper, here is an overview of a typical process you undergo for development with the help of TierDeveloper. The following steps take you through a typical process of generating source code, compiling and deploying it. Then you can generate a test application to verify the auto-generated components.

3.1 Create or Use Sample Project

In order to do any object-to-relational (OR) mapping, you have to create first a TierDeveloper project. Either you can start by creating a new project or you can use sample project files .tier provided with TierDeveloper.

1- Go to "File-> New" or press "CTRL + N" to start a new project.

2- When the new project option is invoked, a dialog starting the wizard is opened as shown in Figure 1. Specify the project specific information.

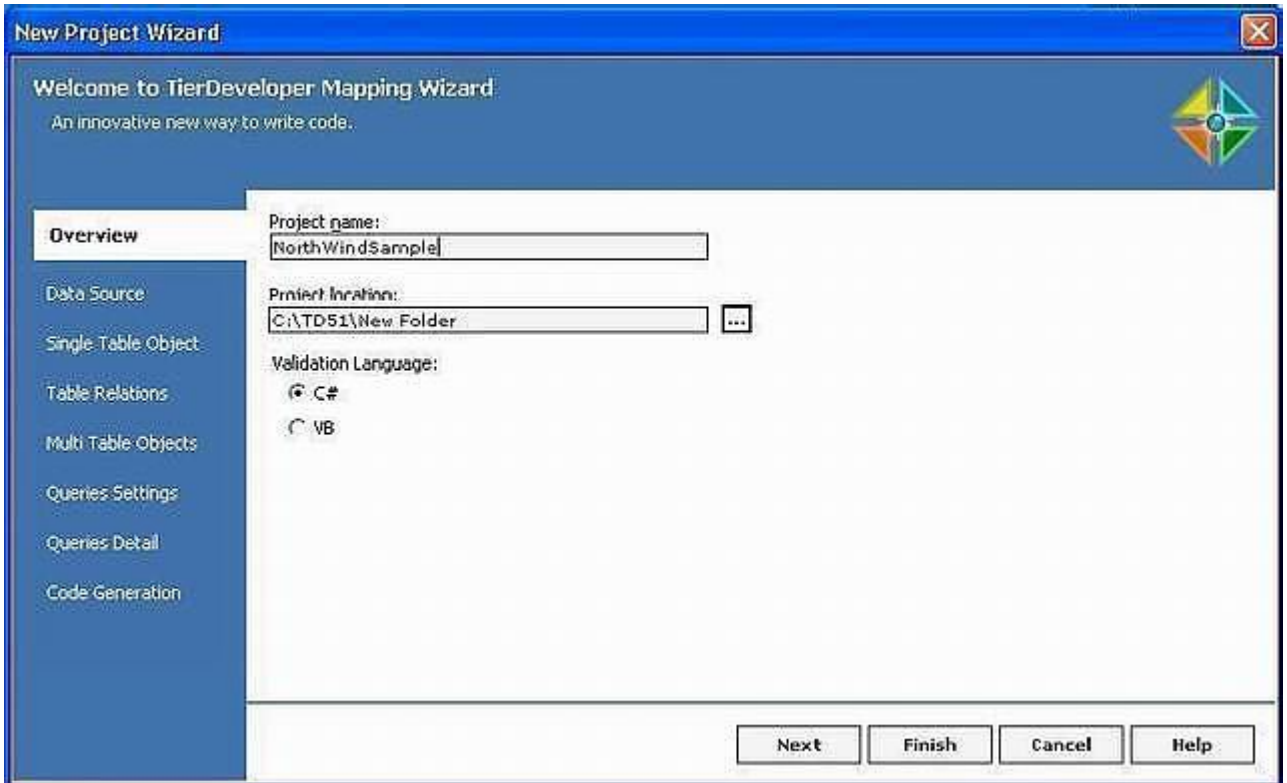


Figure 1: New Project Dialog

You need to specify the following information:

- **Project Name:** Choose a name for the new project.
- **Project Location:** It corresponds to the target path where project file is saved.
- **Validation Language:** Select the target language for your components. This is either C#.NET or VB.NET. You can change component language even after you have created the project.

Once you have provided the mentioned fields for the new project and press **Next** to proceed or select **Finish** to create a project with an empty data source

3- On the third step of the wizard, specify the database connection information. This information is specific to the data source type you select from **Data Source Type**. **Figure 2** displays the database connection information for SQL Server. You need to specify the following information required for the database connection.

Data Source Type: Represents the database types supported by TierDeveloper i.e. Oracle, DB2 and SQL Server.

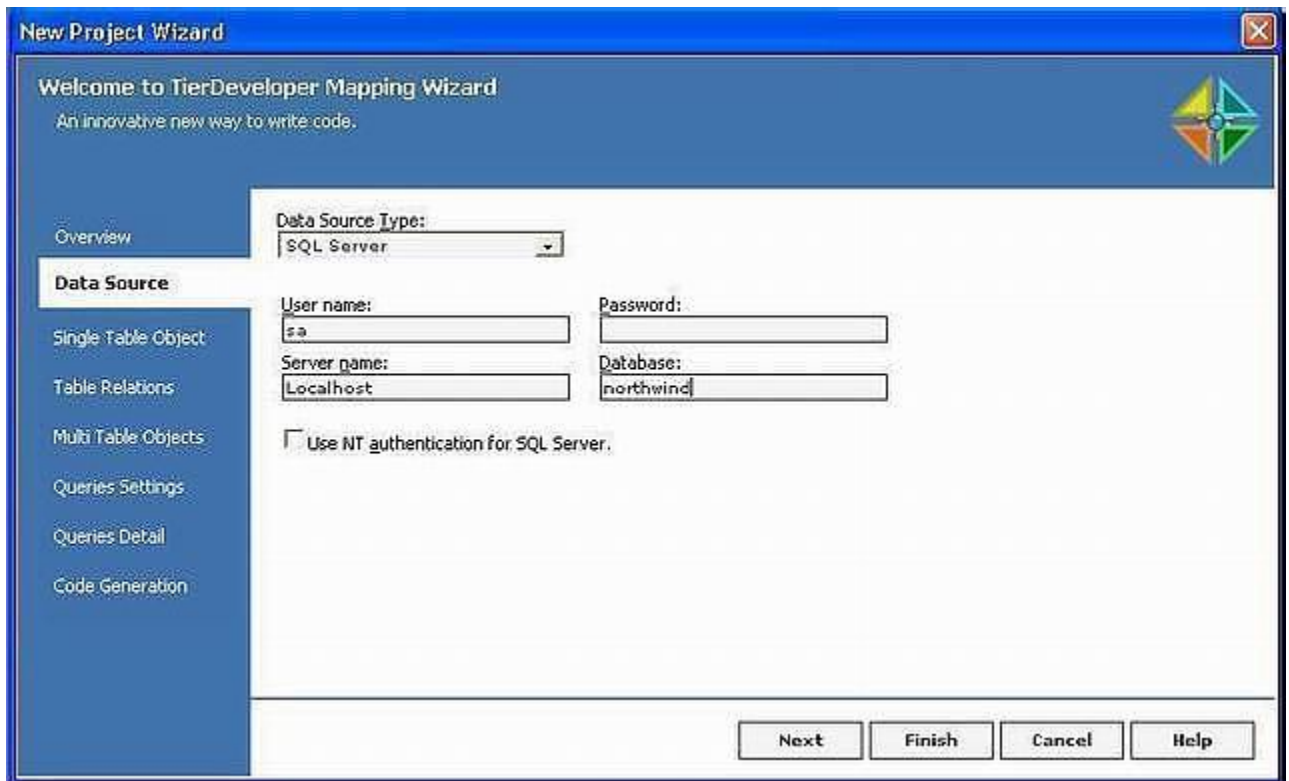


Figure 2: Database Connection Information for SQL Server

For SQL Server you have to provide following information.

- **User Name:** For SQL Server Authentication mode, a valid user name that has access rights for the target database.
- **Password:** It corresponds to the password (if any) for the user.

- **Server Name:** It corresponds to the server name on which the database resides.
- **Database Name:** It represents the target database name.
- **Use NT authentication for SQL Server:** if you select this option, you do not have to specify **User Name** and **Password** to access the database as your NT specific user name and password are used to access the database.

Select **Finish** to create an empty data source project or select **Next** to proceed further.

Note: You can navigate to previous page from **Overview** link from the left pane.

4- Once you have successfully specified the project and database information, you need to acquire the schema information. Click the **Acquire** button on the **Single Table Object** and wait until TierDeveloper displays a list of tables/views and their default mapping with Objects. In case there is some error with the connection, TierDeveloper displays an error message instead of tables/views list. In that case check and change your settings on Data Source page.

After you are able to get the list of tables/views you can select or unselect any of the mappings as shown in **Figure 3**. Each Object here is mapped to a single table/view and gives you the option of specifying which of the standard four methods (i.e. insert, delete, load, and update) you want to include in this Object. You can later edit these Objects and add further dynamic behavior. A Table and a View is distinguished by "T" (for a Table) and "V" (for a View) respectively appearing before the table/view name.

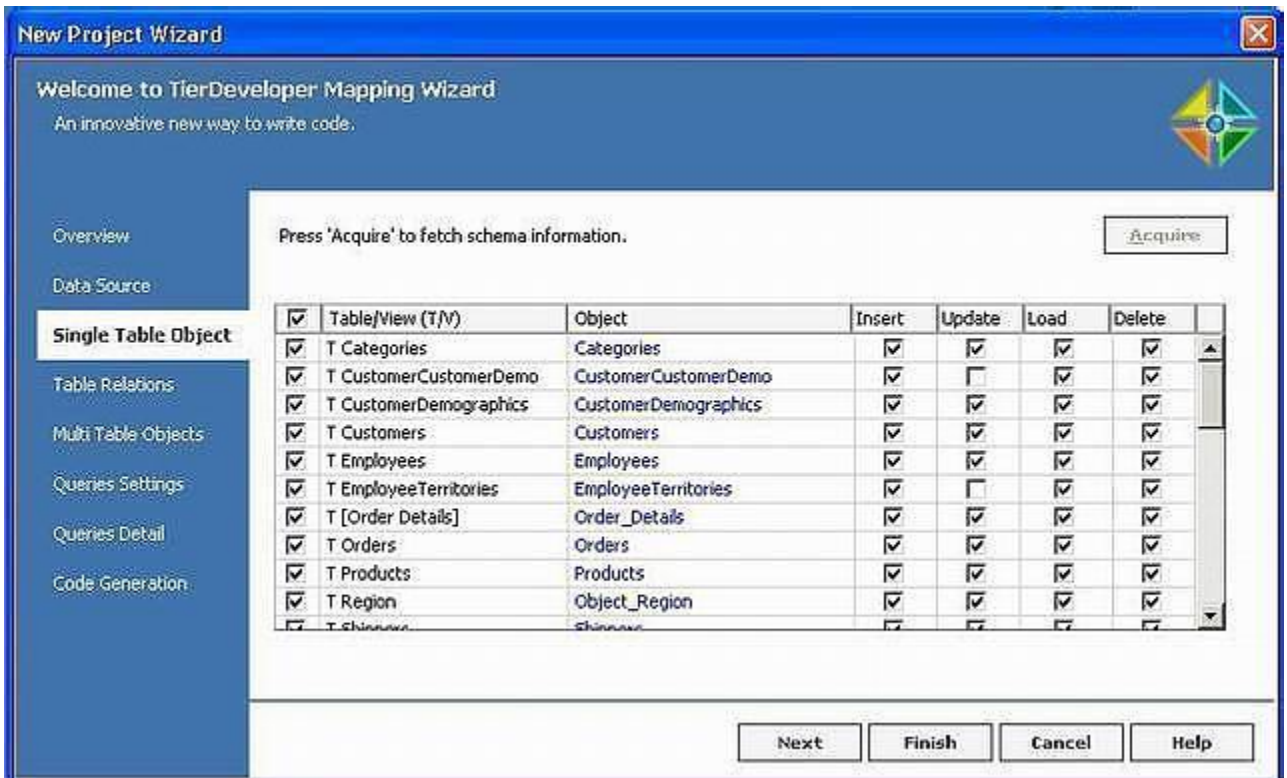


Figure 3: New project Wizard with table selection

Now you can click **Finish** to create a project with a data source but no behavior defined for the mapped Objects. Click **Next** to move to the next page for creating Relationships between the mapped Objects.

5- The **Table Relations** page displays all the Relationships automatically generated between the mapped Objects on the basis of the information in the schema. Select the ones that you want to generate.

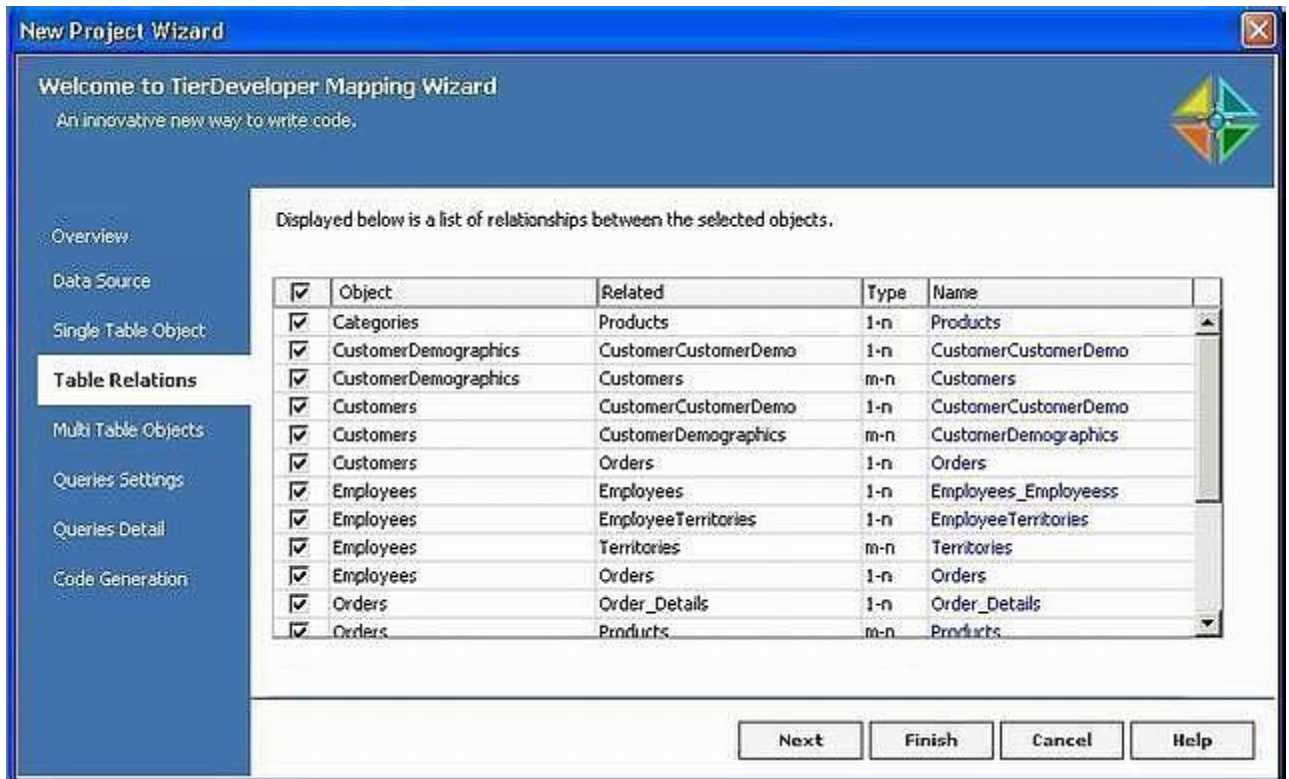


Figure 4: Relationships between the mapped Objects.

- **Object:** Names of the mapped Objects.
- **Related:** The corresponding Object related.
- **Type:** Type of the Relationship.
- **Name:** Name of the relationship generated. It is editable and can be renamed.

Click **Finish** to create a project with the functionality defined until now or Click **Next** to move to the Multi Table Objects page.

6- For creating a Multi Table Object, you can select tables for it. When you select it, a **Table Selection Dialog** is opened. View Figure 5a and 5b. After selecting at least two table, a Multi table Object is mapped and appears. The name and the table joins can be edited.

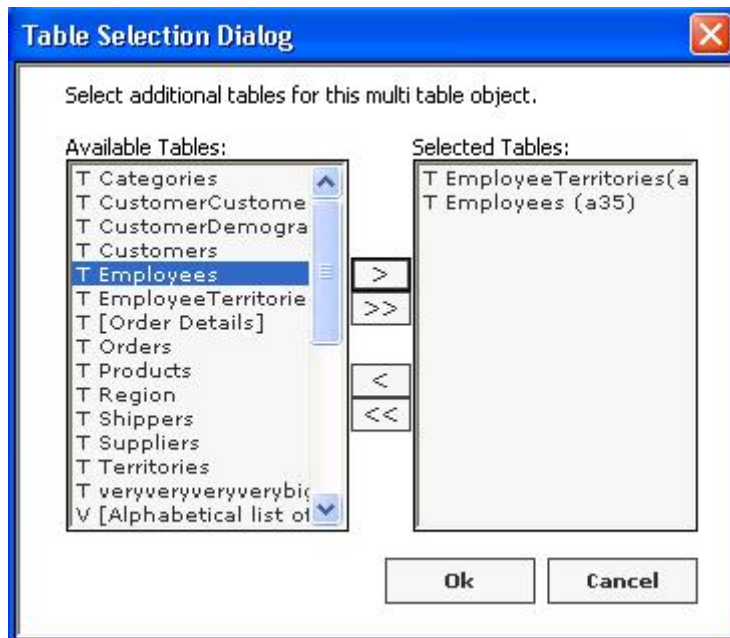


Figure 5a: Selecting tables for a multi table Object

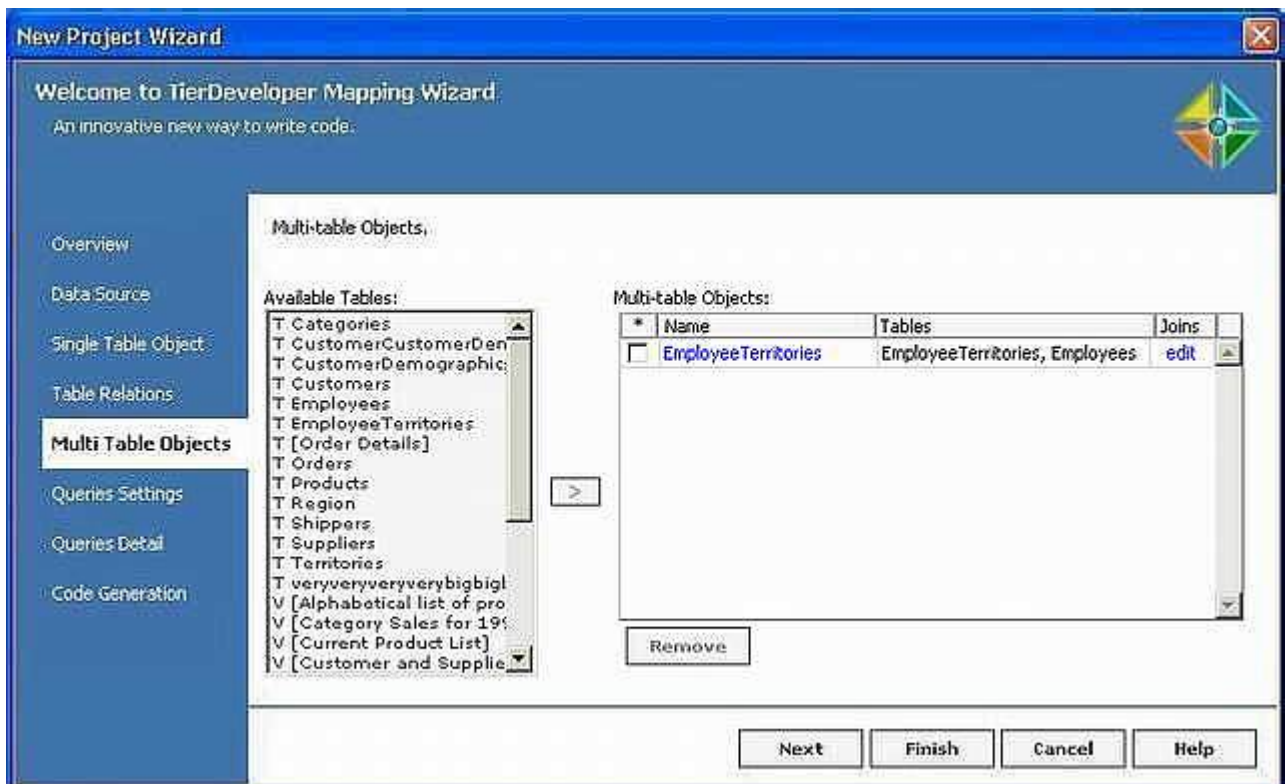


Figure 5b: Multi table Object

- **Available Tables:** Tables and Views from the data source, which can be used to form a Multi table Object.

- **Name (Multi Table Objects):** Name of the Multi table Object. It is editable and you can change its name.
- **Tables (Multi Table Objects):** Names of the tables forming the Multi table Object.
- **Joins (Multi Table Objects):** The basis on which the two tables are bound. It is editable.
- **Remove:** Removes the selected Multi table Object.

You can click **Finish** to create the project with the functionality defined till now or you can click **Next** to move on to the generating Queries automatically for the mapped Objects.

7- On the **Query Settings** page, for each mapped Object, you can choose to generate two types of Queries.

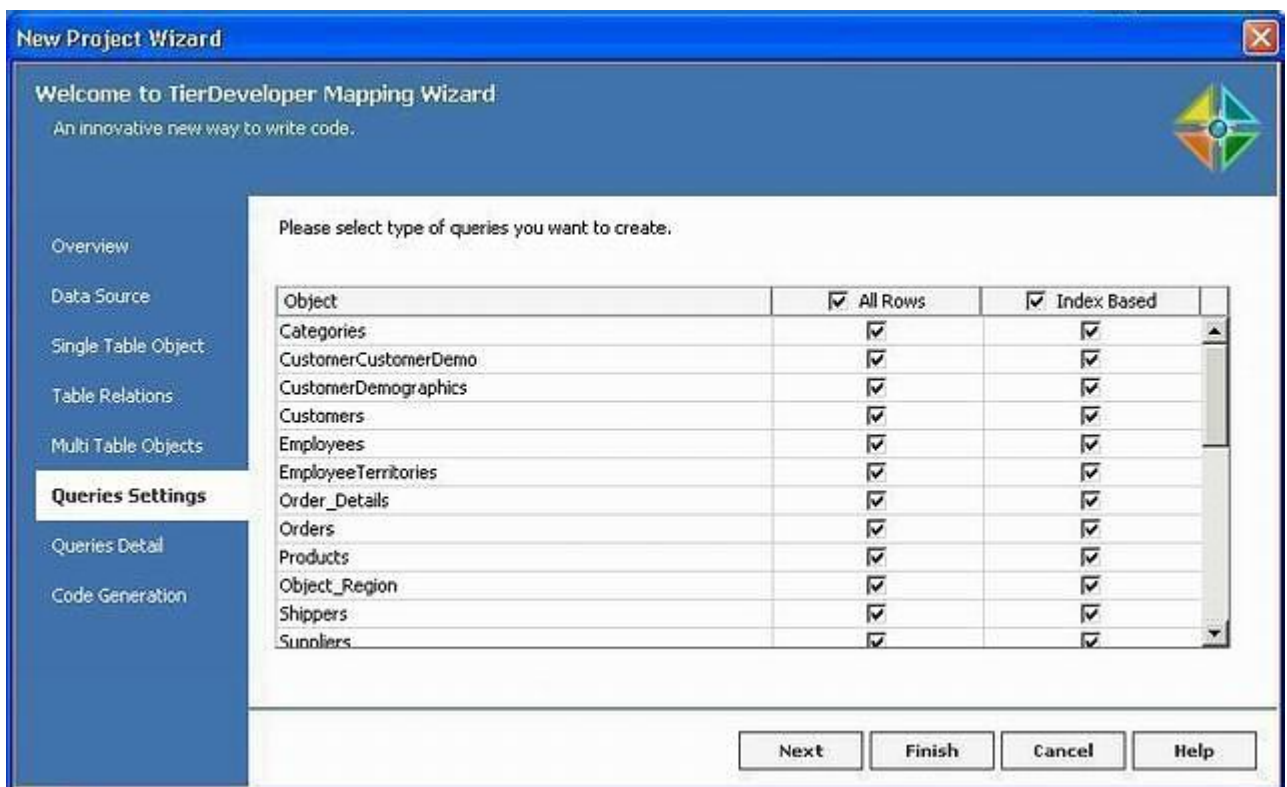


Figure 6: Selection of types of Queries

- **Object:** Names of the mapped Objects.
- **All Rows:** Generates a query which results in all the records returned for the corresponding Object.
- **Index Based:** Generates Queries based on some criteria.

You can click **Finish** to create the project or click **Next** to add further functionality.

8- On **Query Details**, the Queries for the mapped Object are generated depending upon the type chosen on the previous page. From the generated queries, you can further select the ones you which to generate.

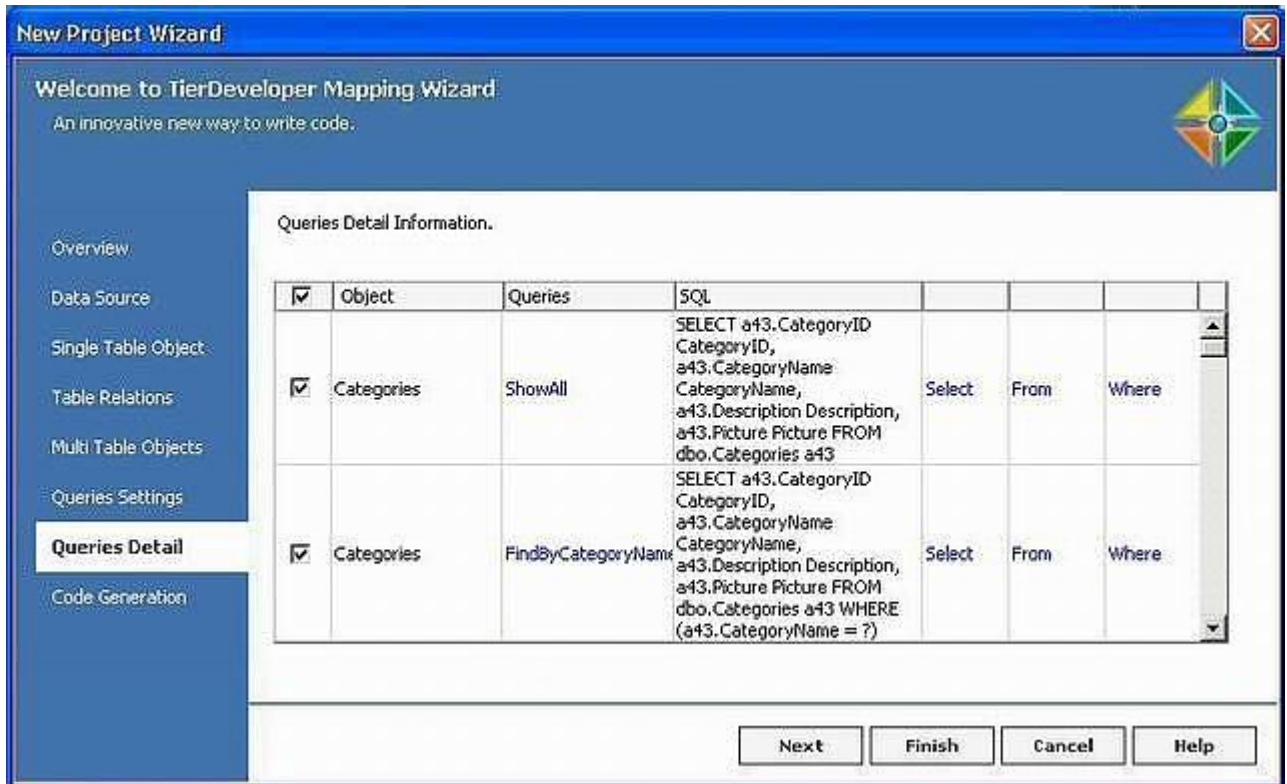


Figure 7: Selection of Queries

- **Object:** Names of the mapped Objects.
- **Queries:** Name of the generated Query. It is editable and can be renamed
- **SQL:** The SQL statement generate for the Query.
- **Select:** The Select clause of the SQL statement. It is editable and can be change according to your need. You can select columns to be displayed.
- **From:** The From clause of the SQL statement. It can be edited from here.
- **Where:** The Where clause of the SQL statement. It is editable.

You can click **Finish** to create the project or click **Next** to add further functionality.

9- The last step is for the code settings. You can specify where and what code to generate for your project. Refer to **Figure 8**.

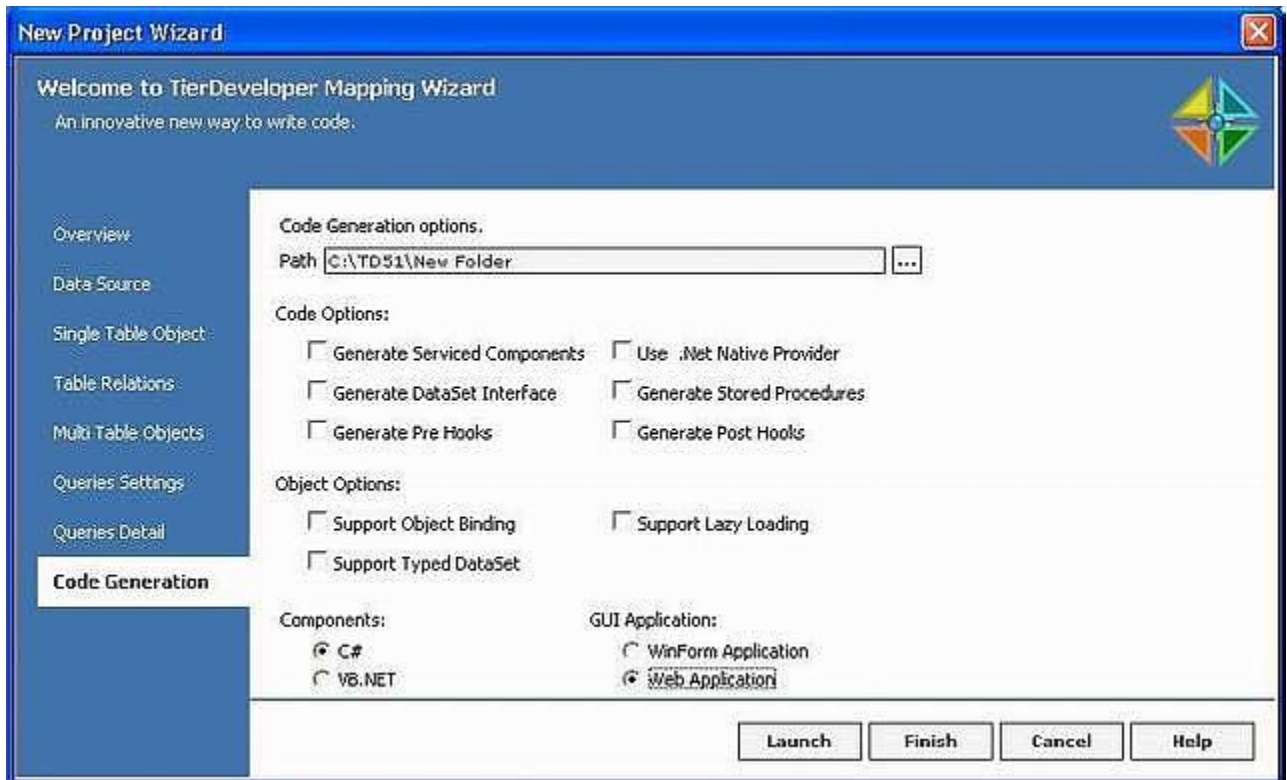


Figure 8: The settings for code generation

- **Path:** The location where the code is generated.
- **Generate Serviced Components:** This feature enables you to use the generated serviced components via private assemblies.
- **Use .NET native provider:** By default, the generated .NET component uses OLE DB library for database access. However if the user checks this option, the generated component code makes use of the .NET native provider in the .NET framework.
- **Generate DataSet Interface:** To generate dataset interfaces for all the Objects.
- **Generate Stored Procedures:** Generates the stored procedure for the project.
- **Generate PreHooks:** Generates PreHooks for all methods.
- **Generate PostHooks:** Generates PostHooks for all methods.
- **Support Object Binding:** To enable support for Object binding for each Object. Collection Objects implement `IBindingList` and `ITypeedList` while the info Object implements `IEditable Object`.
- **Support Lazy Loading:** To enable support for lazy loading for each Object i.e. loading child records on demand.
- **Support Typed DataSet:** To enable support for Typed DataSet for each Object.
- **Components (C#):** To generate the code components using C# as the component language.
- **Components (VB.NET):** To generate the code components using VB.NET as the component language.
- **WinForm Application:** To generate a WinForm application.
- **Web Application:** To generate a Web application.

Select Web Application and now you can click **Launch** to launch the application selected or click **Finish** to open the project in the TierDeveloper environment.

3.2 Generate and Build Source Code

Select "Project -> C# Components -> Generate" menu-item to generate the components in C# and "Project -> C# Components -> Build" to build the code (Similar can be done with VB.NET).

3.3 Generate Web Application

Selecting "Project --> Web Application --> Generate" and "Project-- > Web Application -->Build" results in Web App code generation followed by the build process.

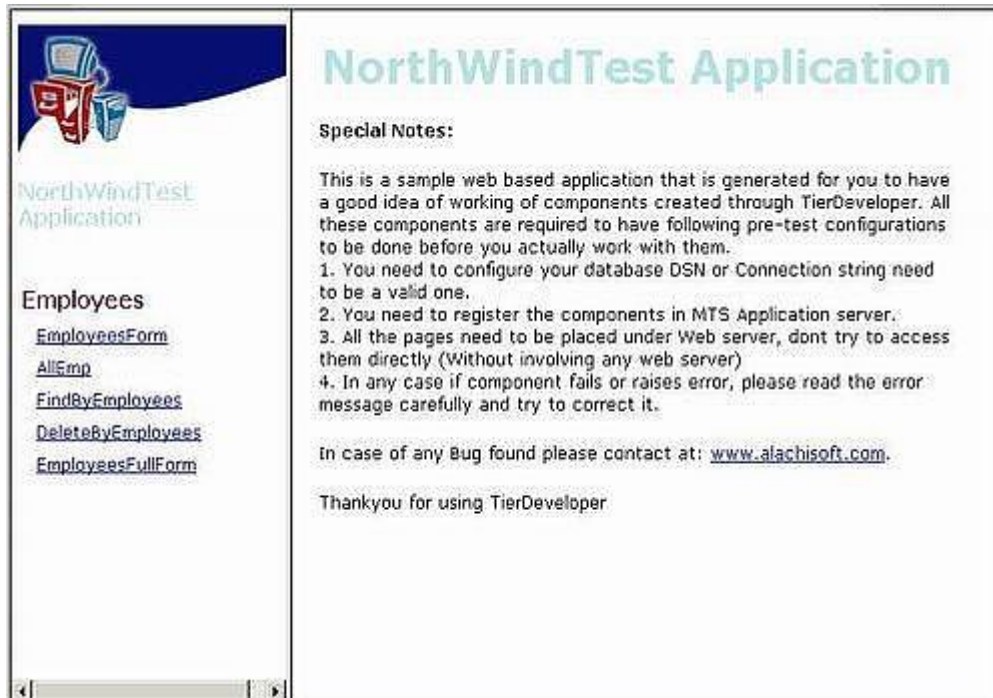
After building the code for the Web Application, it is auto-deployed and then can be launched through "Projects --> Web Application --> Launch". It is opened in a browser.

3.4 Generate Documentation

To generate documentation for the created project, select "Project -> Documentation -> Generate". The Documentation of the application is generated in the HTML format. A folder with the HTML pages corresponding to each Object, its details and classes is generated and are created in a sub-folder called "Docs" under target path.

3.5 Test .NET Components with Web Application

Now you are ready to run the web application and test the generated components. As mentioned before, you can launch the Web app and utilize its functionality.



Web Application

Congratulations! TierDeveloper has completed its job. Similar to the web application, you can also generate and build WinForm application and WebServices which can be tested through remoting WinForm Apps.

4 Additional Resources

Resources for TierDeveloper are as follows:

- 1- To have a quick start in using TierDeveloper, you can benefit from "*NorthWind Sample Code*".
- 2- To have an idea of how to migrate to the latest version of TierDeveloper, refer to "*5.2 Migration Guide*".
- 3- TierDeveloper's "*.NET Programmer's Guide*".
- 4- Understand code generated by TierDeveloper through "*Generated Code Samples*".
- 5- For information on .Net Technology - see www.microsoft.com/net.