



## **MDG Link for Visual Studio.NET User Guide**

*Welcome to the MDG Link for Visual Studio.NET. The MDG Link for Visual Studio.NET enables you to work simultaneously with both Enterprise Architect and Visual Studio.NET and merge the changes with minimal effort.*



# MDG Link for Visual Studio.NET User Guide

© 1998 - 2008 Sparx Systems Pty Ltd

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: May 2008

## **Publisher**

*Sparx Systems*

## **Managing Editor**

*Geoffrey Sparks*

## **Technical Editors**

*Alistair Leslie-Hughes*

## **Special thanks to:**

*All the people who have contributed suggestions, examples, bug reports and assistance in the development of MDG Link for Visual Studio.NET. The task of developing and maintaining this tool has been greatly enhanced by their contribution.*

# Table of Contents

Foreword	1
<b>Introduction</b>	<b>2</b>
Welcome .....	2
Copyright Notice .....	4
Software Product License Agreement .....	5
Acknowledgement of Trademarks .....	8
Support .....	9
System Requirements .....	10
Order MDG Link for Visual Studio.NET .....	11
<b>Getting Started</b>	<b>12</b>
Register the MDG Link for Visual Studio.NET .....	12
Set Up the MDG Link .....	14
Create a Link to a Visual Studio Project .....	15
Merge for the First Time .....	17
<b>Perform Tasks with MDG Link for Visual Studio.NET</b>	<b>18</b>
<b>Add-In Options from the Project Browser</b> .....	<b>18</b>
Add-In Menu Items .....	18
Locate Elements .....	18
Edit Classes .....	18
Edit Operations .....	19
Edit Attributes .....	20
<b>Build Project</b> .....	<b>22</b>
Build and Run a Project .....	22
Build Dialog Options .....	23
Build Project Errors .....	23
<b>Classes</b> .....	<b>25</b>
Create Class .....	26
Edit Class .....	28
Edit Class, Switch to Visual Studio .NET .....	29
Edit Class Attributes and Operations .....	30
Add Inheritance to Classes .....	31
Add Class and Find Association Links .....	32
<b>Code</b> .....	<b>34</b>
Edit Code .....	34
Add Code Comments .....	35
<b>Diagrams</b> .....	<b>37</b>
Format a Diagram .....	37
<b>Round Trip Engineering</b> .....	<b>38</b>
Merge Options .....	38
Merge Project Dialog Options .....	38
Forward Engineering .....	40
Forward Engineer from a Class.....	40
Forward Engineer with a Merge.....	41

Assign Classes to Files for Export.....	42
<b>Reverse Engineering .....</b>	<b>42</b>
Reverse Engineer from a Visual Studio.NET Source Class.....	43
Reverse Engineer with a Merge.....	43
<b>Synchronize Code and Model .....</b>	<b>44</b>
<b>Import and Synchronize TFS Work Items .....</b>	<b>45</b>
Maintain Work Items .....	46
Manage Mapped Fields .....	48
<b>Index .....</b>	<b>50</b>

# Foreword

MDG Link for Visual Studio.NET provides a light-weight bridge between Enterprise Architect and Visual Studio.NET.

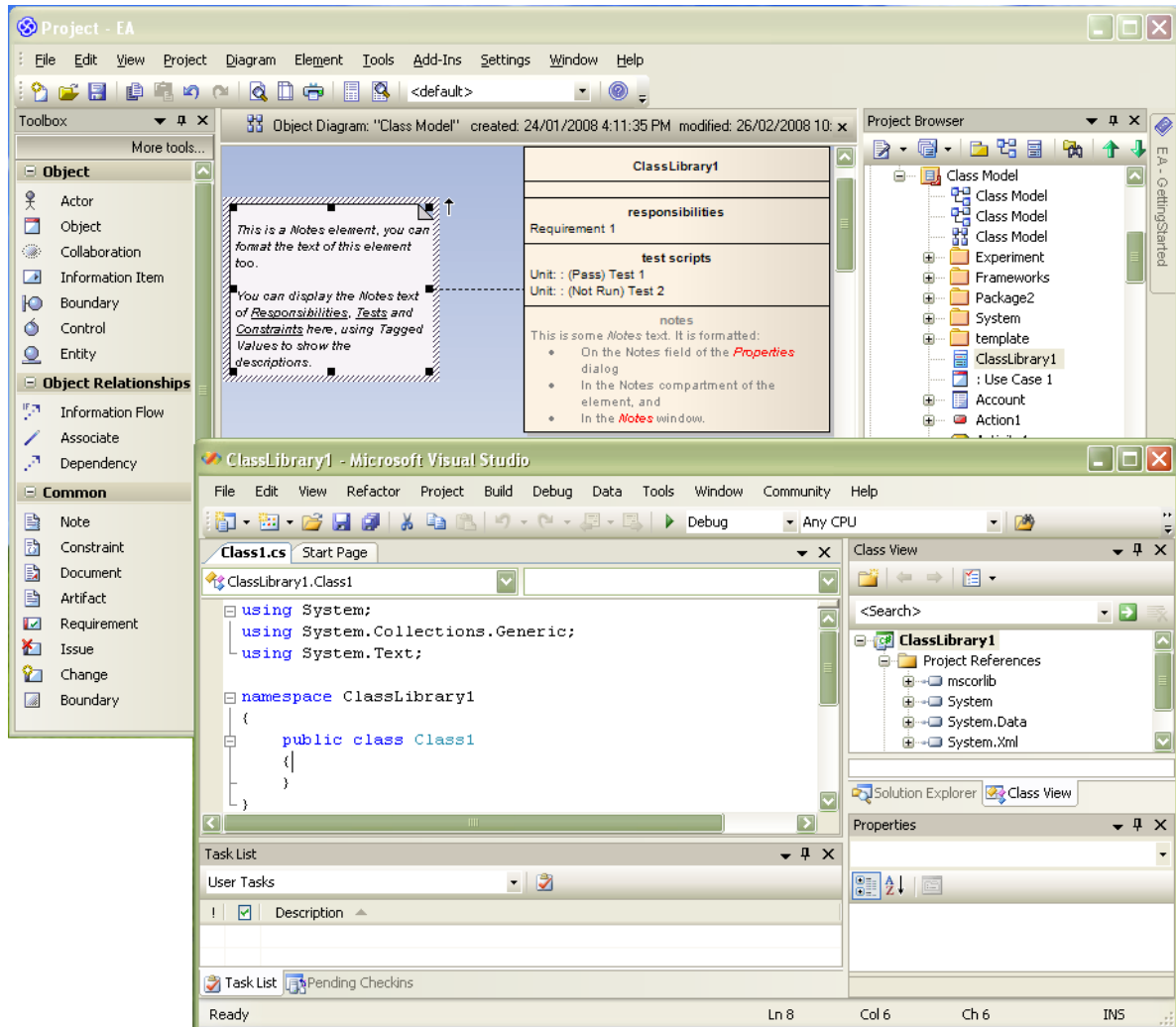
## 1.1 Welcome



Welcome to the Model Driven Generation Link™ for Visual Studio.NET<sup>®</sup>. The MDG Link for Visual Studio.NET is designed to enable you to work simultaneously with both Enterprise Architect and Visual Studio.NET, and merge the changes with minimal effort. The MDG Link for Visual Studio.NET works with both the Professional and Corporate editions of Enterprise Architect, and provides a tight integration between Enterprise Architect and Visual Studio, enabling you to either create UML in Enterprise Architect or generate UML from Visual Studio.NET.

### **MDG Link for Visual Studio.NET has the following features:**

- Provides a simple, easy to use connection between Enterprise Architect models and Visual Studio.NET projects.
- Enables you to merge an entire project simply and easily.
- Provides support for different development configurations.
- Prompts you with the proposed merge before changes are written.



To get started now, see [Getting Started](#)<sup>14</sup>.

### See Also

- [License Agreement](#)<sup>5</sup>
- [Copyright Notice](#)<sup>4</sup>

## 1.2 Copyright Notice

### Copyright © 1998 - 2008 Sparx Systems Pty. Ltd. All rights reserved.

The software contains proprietary information of Sparx Systems Pty Ltd. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. Please read the [license agreement](#)<sup>[54]</sup> for full details.

Due to continued product development, this information may change without notice. The information and intellectual property contained herein is confidential between Sparx Systems and the client and remains the exclusive property of Sparx Systems. If you find any problems in the documentation, please report them to us in writing. Sparx Systems does not warrant that this document is error-free. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Sparx Systems. Licensed users are granted the right to print a single hardcopy of the user manual per licensed copy of the software, but may not sell, distribute or otherwise dispose of the hardcopy without written consent of Sparx Systems.

#### **Sparx Systems Pty. Ltd.**

7 Curtis St,  
Creswick, Victoria 3363,  
AUSTRALIA

Phone: +61 (3) 5345 1140

Fax: +61 (3) 5345 1104

Support Email: [support@sparxsystems.com](mailto:support@sparxsystems.com)

Sales Email: [sales@sparxsystems.com](mailto:sales@sparxsystems.com)

Website: [www.sparxsystems.com](http://www.sparxsystems.com)



## 1.3 Software Product License Agreement

### MDG Link for Visual Studio.NET

#### Copyright (C) 1998-2008 Sparx Systems Pty Ltd. All Rights Reserved

**IMPORTANT- READ CAREFULLY:** This End User License Agreement ("EULA") is a legal agreement between YOU as Licensee and SPARX for the SOFTWARE PRODUCT identified above. By installing, copying, or otherwise using the SOFTWARE PRODUCT, YOU agree to be bound by the terms of this EULA. If YOU do not agree to the terms of this EULA, promptly return the unused SOFTWARE PRODUCT to the place of purchase for a full refund.

The copyright in the SOFTWARE PRODUCT and its documentation is owned by Sparx Systems Pty Ltd A.B.N 38 085 034 546. Subject to the terms of this EULA, YOU are granted a non-exclusive right for the duration of the EULA to use the SOFTWARE PRODUCT. YOU do not acquire ownership of copyright or other intellectual property rights in any part of the SOFTWARE PRODUCT by virtue of this EULA.

Your use of this software indicates your acceptance of this EULA and warranty.

### DEFINITIONS

In this End User License Agreement, unless the contrary intention appears:

- "ACADEMIC EDITION" means an edition of the Software Product purchased for educational purposes at an academic discount price.
- "EULA" means this End User License Agreement
- "SPARX" means Sparx Systems Pty Ltd A.B.N 38 085 034 546
- "Licensee" means YOU, or the organization (if any) on whose behalf YOU are taking the EULA.
- "Registered Edition of MDG Link for Visual Studio.NET" means the edition of the SOFTWARE PRODUCT which is available for purchase from the web site: <[http://www.sparxsystems.com/ea\\_purchase.htm](http://www.sparxsystems.com/ea_purchase.htm)>.
- "SOFTWARE PRODUCT" or "SOFTWARE" means MDG Link for Visual Studio.NET, which includes computer software and associated media and printed materials, and may include online or electronic documentation.
- "Support Services" means email based support provided by SPARX, including advice on usage of MDG Link for Visual Studio.NET, investigation of bugs, fixes, repairs of models if and when appropriate and general product support.
- "SPARX support engineers" means employees of SPARX who provide on-line support services.
- "Trial edition of MDG Link for Visual Studio.NET" means the edition of the SOFTWARE PRODUCT which is available free of charge for evaluation purposes for a period of 30 days.

### GRANT OF LICENSE

In accordance with the terms of this EULA YOU are granted the following rights:

- a) To install and use one copy of the SOFTWARE PRODUCT or, in its place, any prior version for the same operating system, on a single computer. As the primary user of the computer on which the SOFTWARE PRODUCT is installed, YOU may make a second copy for your exclusive use on either a home or portable computer.
- b) To store or install a copy of the SOFTWARE PRODUCT on a storage device, such as a network server, used only to install or run the SOFTWARE PRODUCT over an internal network. If YOU want to increase the number of users entitled to concurrently access the SOFTWARE PRODUCT, YOU must notify SPARX and agree to pay an additional fee.
- c) To make copies of the SOFTWARE PRODUCT for backup and archival purposes.

### EVALUATION LICENSE

The Trial version of MDG Link for Visual Studio.NET is not free software. Subject to the terms of this agreement, YOU are hereby licensed to use this software for evaluation purposes without charge for a period of 30 days.

Upon expiration of the 30 days, the Software Product must be removed from the computer. Unregistered use of MDG Link for Visual Studio.NET after the 30-day evaluation period is in violation of Australian, U.S. and international copyright laws.

SPARX may extend the evaluation period on request and at their discretion.

If YOU choose to use this software after the 30 day evaluation period a license must be purchased (as described at [http://www.sparxsystems.com/ea\\_purchase.htm](http://www.sparxsystems.com/ea_purchase.htm)). Upon payment of the license fee, YOU will be

sent details on where to download the registered edition of MDG Link for Visual Studio.NET and will be provided with a suitable software 'key' by email.

## ASSIGNMENT

YOU may only assign all your rights and obligations under this EULA to another party if YOU supply to the transferee a copy of this EULA and all other documentation including proof of ownership. Your license is then terminated.

## TERMINATION

Without prejudice to any other rights, SPARX may terminate this EULA if YOU fail to comply with the terms and conditions. Upon termination YOU or YOUR representative shall destroy all copies of the SOFTWARE PRODUCT and all of its component parts or otherwise return or dispose of such material in the manner directed by SPARX.

## WARRANTIES AND LIABILITY

### WARRANTIES

SPARX warrants that the SOFTWARE PRODUCT will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt, and any Support Services provided by SPARX shall be substantially as described in applicable written materials provided to YOU by SPARX, and SPARX support engineers will make commercially reasonable efforts to solve any problems associated with the SOFTWARE PRODUCT.

### EXCLUSIONS

To the maximum extent permitted by law, SPARX excludes, for itself and for any supplier of software incorporated in the SOFTWARE PRODUCT, all liability for all claims, expenses, losses, damages and costs made against or incurred or suffered by YOU directly or indirectly (including without limitation lost costs, profits and data) arising out of:

- YOUR use or misuse of the SOFTWARE PRODUCT
- YOUR inability to use or obtain access to the SOFTWARE PRODUCT
- Negligence of SPARX or its employees, contractors or agents, or of any supplier of software incorporated in the SOFTWARE PRODUCT, in connection with the performance of SPARX' obligations under this EULA, or
- Termination of this EULA by either party for any reason.

### LIMITATION

The SOFTWARE PRODUCT and any documentation are provided "AS IS" and all warranties whether express, implied, statutory or otherwise, relating in any way to the subject matter of this EULA or to this EULA generally, including without limitation, warranties as to: quality, fitness; merchantability; correctness; accuracy; reliability; correspondence with any description or sample, meeting your or any other requirements; uninterrupted use; compliance with any relevant legislation and being error or virus free are excluded. Where any legislation implies in this EULA any term, and that legislation avoids or prohibits provisions in a contract excluding or modifying such a term, such term shall be deemed to be included in this EULA. However, the liability of SPARX for any breach of such term shall if permitted by legislation be limited, at SPARX's option to any one or more of the following upon return of the SOFTWARE PRODUCT and a copy of the receipt:

- If the breach relates to the SOFTWARE PRODUCT:
  - the replacement of the SOFTWARE PRODUCT or the supply of an equivalent SOFTWARE PRODUCT
  - the repair of such SOFTWARE PRODUCT
  - the payment of the cost of replacing the SOFTWARE PRODUCT or of acquiring an equivalent SOFTWARE PRODUCT, or
  - the payment of the cost of having the SOFTWARE PRODUCT repaired.
- If the breach relates to services in relation to the SOFTWARE PRODUCT:
  - the supplying of the services again, or
  - the payment of the cost of having the services supplied again.

## TRADEMARKS

All names of products and companies used in this EULA, the SOFTWARE PRODUCT, or the enclosed documentation may be trademarks of their corresponding owners. Their use in this EULA is intended to be in compliance with the respective guidelines and licenses.

Windows, Windows NT, Windows ME, Windows XP and Windows 2000 are trademarks of Microsoft.

**GOVERNING LAW**

This agreement shall be construed in accordance with the laws of the Commonwealth of AUSTRALIA.

## 1.4 Acknowledgement of Trademarks

### Trademarks of Microsoft

- Microsoft Visual Studio.NET
- Windows®.

### Registered Trademarks of OMG

- CORBA®
- the OMG Object Management Group logo
- The Information Brokerage®
- CORBA Academy®
- IIOB®
- XMI®.

### Trademarks of OMG

- OMG™
- Object Management Group™
- The CORBA logo
- ORB™
- Object Request Broker™
- The CORBA Academy design
- OMG Interface Definition Language™
- IDL™
- CORBAservices™
- CORBAfacilities™
- CORBAMED™
- CORBAnet™
- Unified Modeling Language™
- UML™
- The UML Cube logo
- MOF™
- CWM™
- Model Driven Architecture™
- MDA™
- OMG Model Driven Architecture™
- OMG MDA™.

---

## 1.5 Support

Technical support for MDG Link for Visual Studio.NET is available to registered users. Responses to support queries are sent by email. Sparx Systems endeavors to provide a rapid response to all product-related questions or concerns.

Registered users can lodge a support request, by visiting:  
[http://www.sparxsystems.com/registered/reg\\_support.html](http://www.sparxsystems.com/registered/reg_support.html).

Trial users can contact Sparx Systems with questions regarding their evaluation at:  
[support@sparxsystems.com](mailto:support@sparxsystems.com).

An online user forum is also available for your questions and perusal, at  
<http://www.sparxsystems.com/cgi-bin/yabb/YaBB.cgi>.

## 1.6 System Requirements

The following software must be installed to use the MDG Link for Visual Studio.NET:

### Operating System

- Windows XP Professional (SP2 or later)
- Windows XP Home Edition (SP2 or later)
- Windows XP Media Center Edition (SP2 or later)
- Windows XP Tablet PC Edition (SP2 or later)
- Windows 2000 Professional (SP5 or later required for installation)

### Enterprise Architect

- Enterprise Architect Version 7.1 Professional Edition, or
- Enterprise Architect Version 7.1 Corporate Edition

### Visual Studio

- Visual Studio.NET version 2003 or higher

### Other

- Sun JDK 1.5

## **1.7 Order MDG Link for Visual Studio.NET**

MDG Link for Visual Studio.NET is designed, built and published by Sparx Systems and is available from [Sparx Systems](#).

The latest information on pricing and purchasing is available at: [Sparx Systems Purchase/Pricing Website](#).

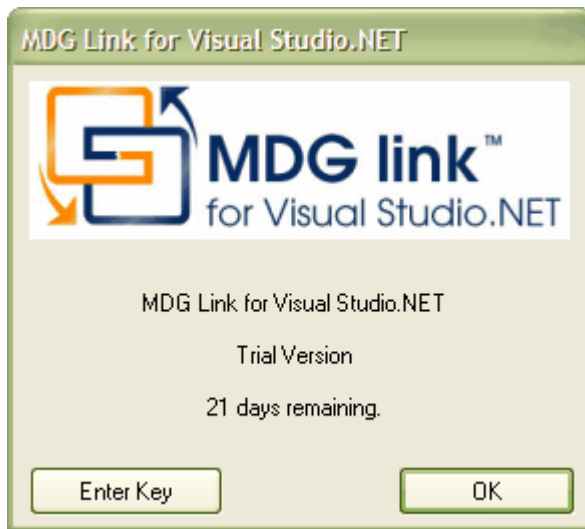
For more information, contact [sales@sparxsystems.com](mailto:sales@sparxsystems.com).

## 2.1 Register the MDG Link for Visual Studio.NET

To activate the MDG Link for Visual Studio.NET, follow the steps below:

1. Purchase one or more licenses. When you have paid for a licensed version of the MDG Link for Visual Studio.NET, you receive (via email or other suitable means):
  - a license key or keys
  - the address of the web site from which to download the full version.
2. Save the license key and download the latest full install package from the address supplied.
3. Run the setup program to install the full version.

If this is the first time that the MDG Link for Visual Studio.NET has been installed, the **MDG Link for Visual Studio.NET** dialog prompts you to enter the registration key or to continue the trial.

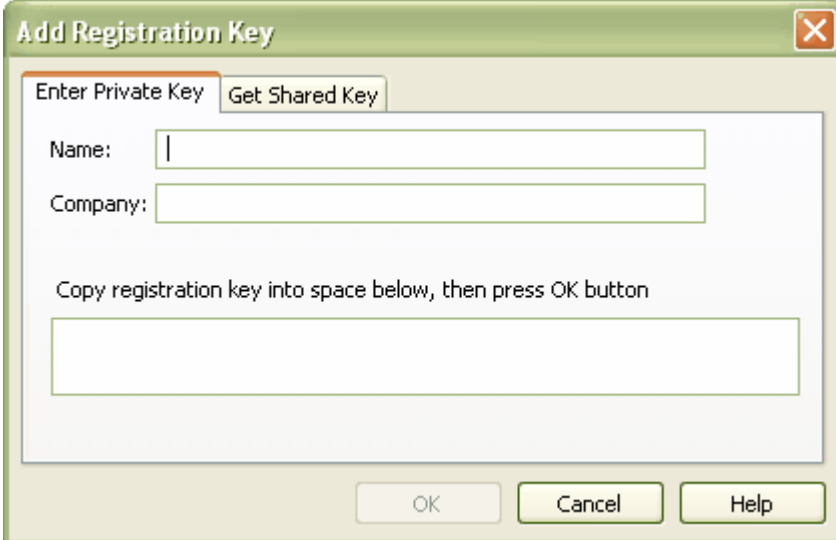


4. To enter the new key, click on the **Enter Key** button. The **License Management** dialog displays.



5. Click on the **Add Key** button. The **Add Registration Key** dialog displays.





**Add Registration Key**

Enter Private Key | Get Shared Key

Name:

Company:

Copy registration key into space below, then press OK button

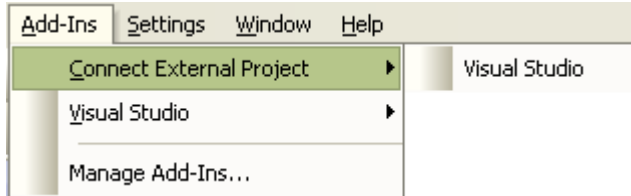
OK Cancel Help

6. Copy the license key from the email and paste it into the **Copy registration key ...** field.
7. Click on the **OK** button. The full version of the MDG Link for Visual Studio.NET is available for use with your version of Enterprise Architect.

## 2.2 Set Up the MDG Link

Before the MDG Link for Visual Studio .NET can perform its main operations, an Enterprise Architect package must be configured to link to a particular Visual Studio.NET project. For more information on how to create a link to a Visual Studio project go to the [Create a link to a Visual Studio Project](#)<sup>[15]</sup> topic.

Once the install program has been run, MDG Link for Visual Studio.NET should be accessible through the **Add-Ins** option on the Enterprise Architect menu bar, as shown below.

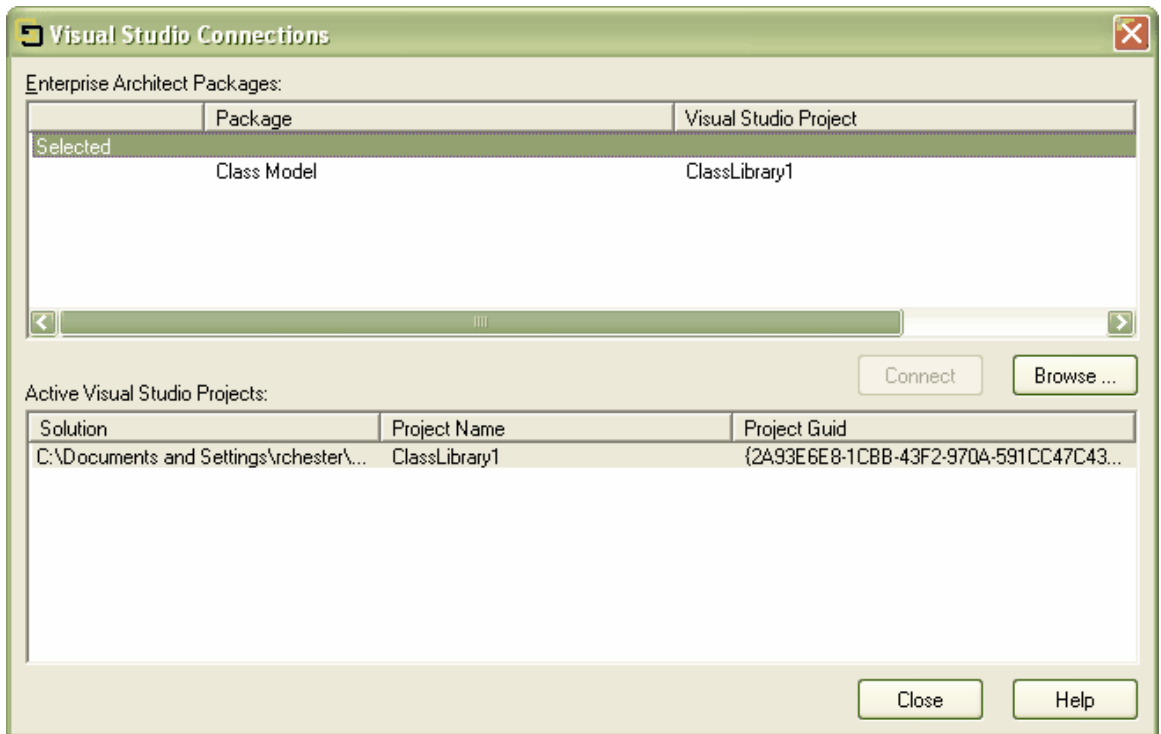


If this menu doesn't appear, check the [System Requirements](#)<sup>[10]</sup>.

## 2.3 Create a Link to a Visual Studio Project

To link an Enterprise Architect package to a particular Visual Studio.NET project, follow the steps below:

1. From Visual Studio.NET, open the solution containing the project to link to.
2. Ensure that the project is the active project within the solution.
3. Open an Enterprise Architect model and in the **Project Browser** select the package that is to represent the Visual Studio project.
4. Right-click on the package to display its context menu, and select the **Add-In | Connect External Project | Visual Studio** menu option. The **Visual Studio Connections** dialog displays.



The **Visual Studio Connections** dialog enables you to review and configure connections to Visual Studio.Net from this project.

Field	Description
<b>Enterprise Architect Packages</b>	Shows the Enterprise Architect packages in the current model that are connected to Visual Studio.net projects.
<b>Selected</b>	If the currently selected package in the Enterprise Architect <b>Project Browser</b> has a new screen.
<b>Active Visual Studio Projects</b>	The Visual Studio.NET projects that you can connect to.

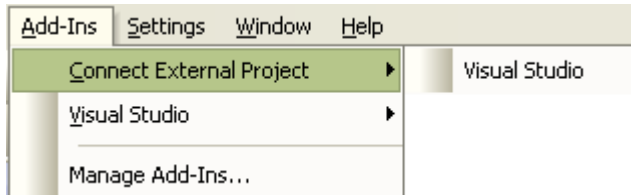
Button	Description
<b>Connect</b>	Connect the Enterprise Architect package to the selected Visual Studio project.
<b>Browse</b>	Locate and select a Visual Studio solution via Windows Explorer. Once selected, the solution and its projects appear in the <b>Active Visual Studio Projects</b> list.

**Note:**

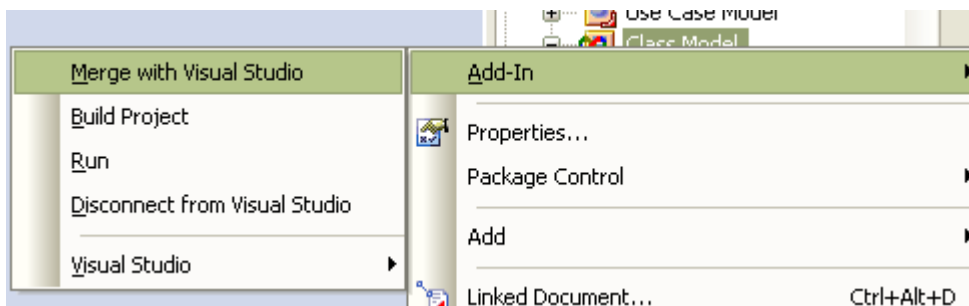
If you are using an Enterprise Architect model that has already been configured on another machine, you must still tell the MDG Link where the solution lies. To do this, open the solution through Visual Studio.NET and click on any of the menu items.

## 2.4 Merge for the First Time

Merging for the first time enables you to reverse engineer code from a Visual Studio.NET project or to generate code from an Enterprise Architect model into a Visual Studio.NET project. Merging the model is a simple task once [a link has been created](#)<sup>[15]</sup> to Visual Studio .NET; you can then perform a merge from the **Project Browser** or from the **Add-Ins | Connect External Project | Visual Studio** menu option on the Enterprise Architect menu bar.



To perform a merge from the **Project Browser**, select an item from the hierarchy and right-click on the connected package. The context menu displays for the item. Select the **Add-In | Merge with Visual Studio** menu option.



For more information on the options that are available for merging see the [Merge Project Dialog Options](#)<sup>[38]</sup> topic.

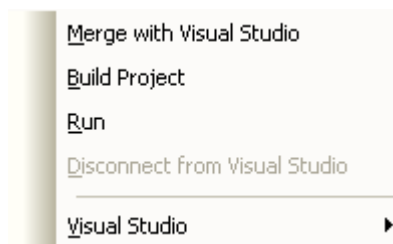
### 3.1 Add-In Options from the Project Browser

The **Project Browser** enables you to navigate through the Enterprise Architect project space. It displays packages, diagrams, elements and element properties.

You can drag and drop elements between folders, or even drop elements from the **Project Browser** directly into the current diagram. With the MDG Link for Visual Studio.NET additional functionality is given to the **Project Browser**. This includes the ability to access the **Add-In** menu, locate Class diagrams and provide the direct link to editing both Classes and methods in Visual Studio.NET.

#### 3.1.1 Add-In Menu Items

To access the **Add-In** menu from the **Project Browser**, right-click on an object to display the context menu. The **Add-In** menu option is the first entry; select it, to display the following submenu:



The **Add-In** menu offers the following options:

Menu Option	Description
<b>Merge with Visual Studio</b>	Displays the <a href="#">Merge Project</a> [38] dialog to provide merging options.
<b>Build Project</b>	<a href="#">Builds</a> [22] the current project.
<b>Run</b>	Runs the project.
<b>Disconnect from Visual Studio</b>	Disconnects the Enterprise Architect package from the Visual Studio .Net project to free that package so that it can be connected to other solutions.
<b>Visual Studio</b>	Provides two further options: <ul style="list-style-type: none"> <li>• Display the <a href="#">Visual Studio Connections</a> [15] dialog with details of the Visual Studio.NET connections</li> <li>• Import and synchronize work items from <a href="#">Team Foundation Server</a> [45].</li> </ul>

#### 3.1.2 Locate Elements

Locating an element in the **Project Browser** can be a difficult task, especially when the size of a package has increased to include many Classes.

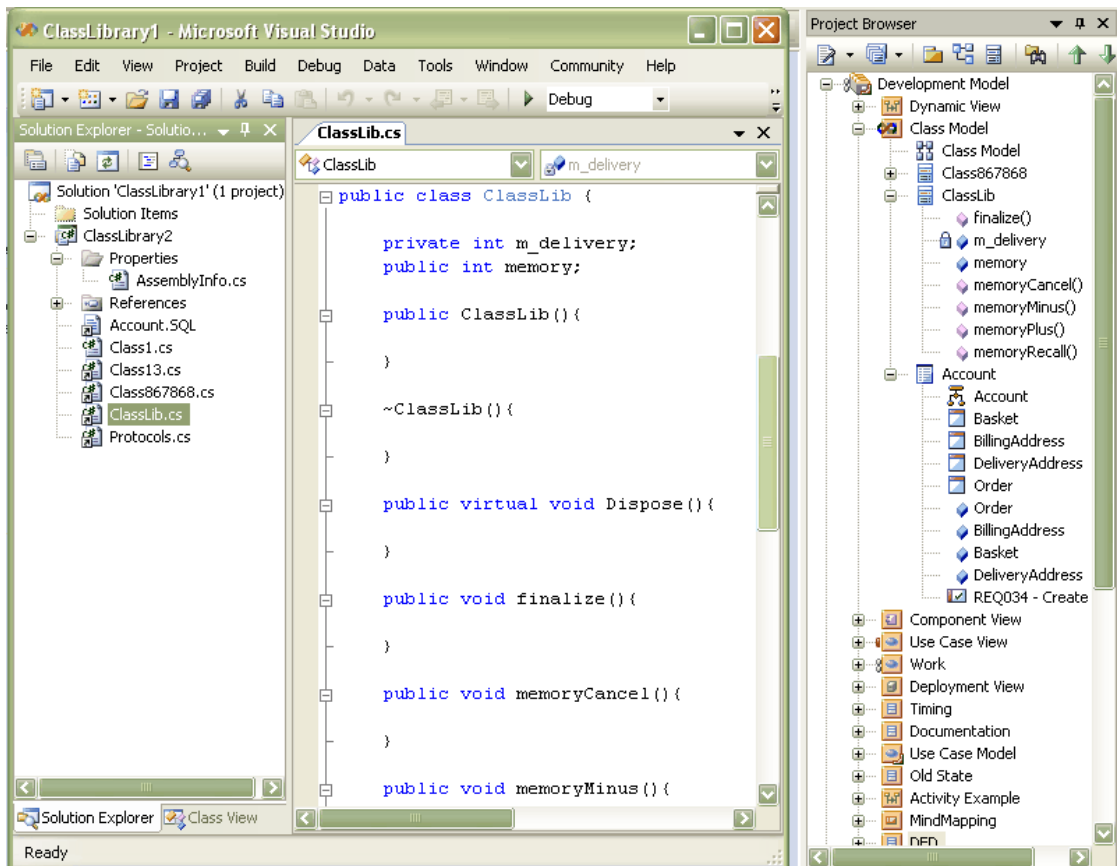
To locate a Class in the **Project Browser**, right-click on the Class in a diagram to display its context menu. Then select the **Find | In Project Browser** menu option. The Class is highlighted in the **Project Browser**. Alternatively select the Class in the diagram and press **[Alt]+[G]**.

#### 3.1.3 Edit Classes

The **Project Browser** enables you to easily access the details of a Class, so that you can edit the Class properties directly from Enterprise Architect or in Visual Studio.NET. To select a specific Class follow the steps below:

1. In the **Project Browser** navigate to the package containing the required Class.

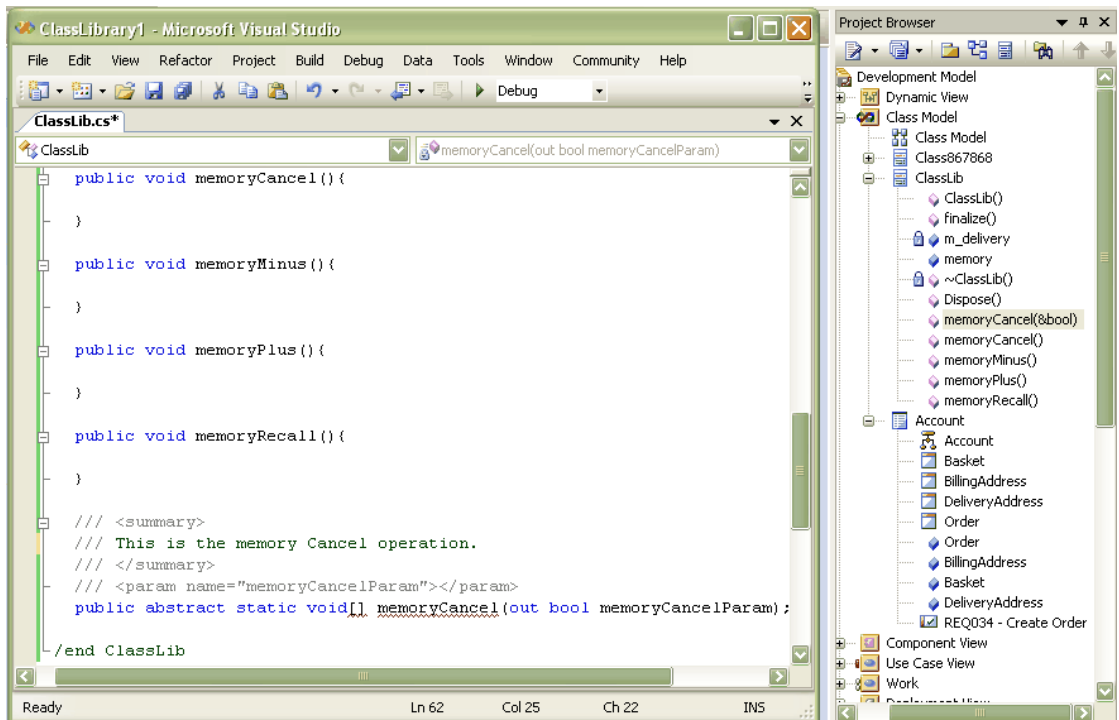
2. Expand the details of the Class by clicking on the + symbol next to the Class name
3. Right-click on the Class to display its context menu:
  - Select the **Properties** option to access the Class **Properties** dialog in Enterprise Architect.
  - Select the **View Source Code** option to edit the Class code in Visual Studio.NET.



### 3.1.4 Edit Operations

The **Project Browser** enables you to easily access the operations of a Class, so that you can edit the operations directly from Enterprise Architect or in Visual Studio.NET. To select a specific operation follow the steps below:

1. In the **Project Browser**, locate and expand the details of the required Class.
2. Locate the required operation and right-click on it to display its context menu:
  - Select the **Operation Properties** option to display the operation **Properties** dialog within Enterprise Architect.
  - Select the **View Source Code** option to edit the operation in Visual Studio .NET; this displays the operation in the code.

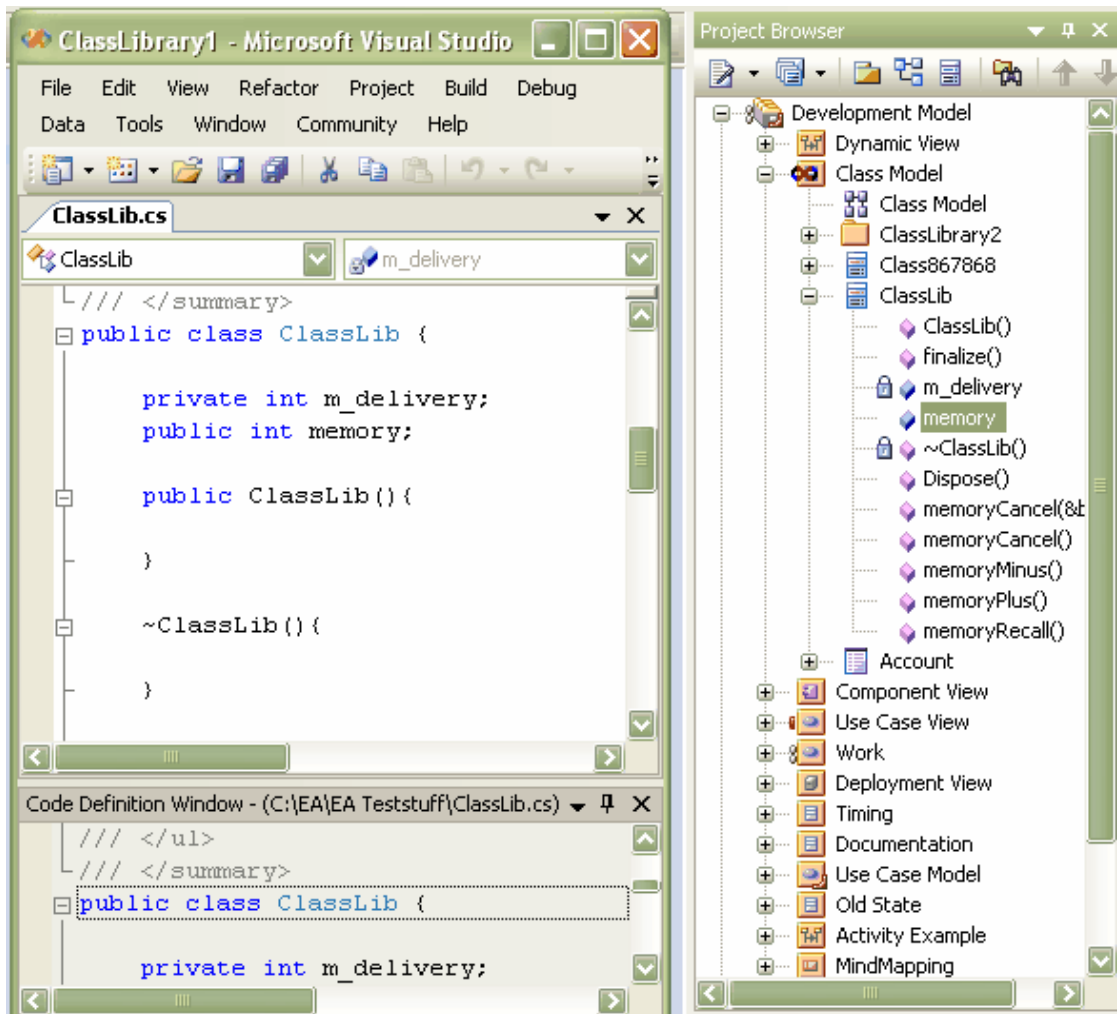


### 3.1.5 Edit Attributes

The **Project Browser** enables you to easily access the attributes of a Class, so that you can edit the attributes directly from Enterprise Architect or in Visual Studio.NET. To select a specific attribute follow the steps below:

1. In the **Project Browser**, locate and expand the details of the required Class.
2. Locate the required attribute and right-click on it to display its context menu:
  - Select the **Attribute Properties** option to display the attribute **Properties** dialog within Enterprise Architect.
  - Select the **View Source Code** option to edit the attribute in Visual Studio .NET; this displays the attribute in the code.



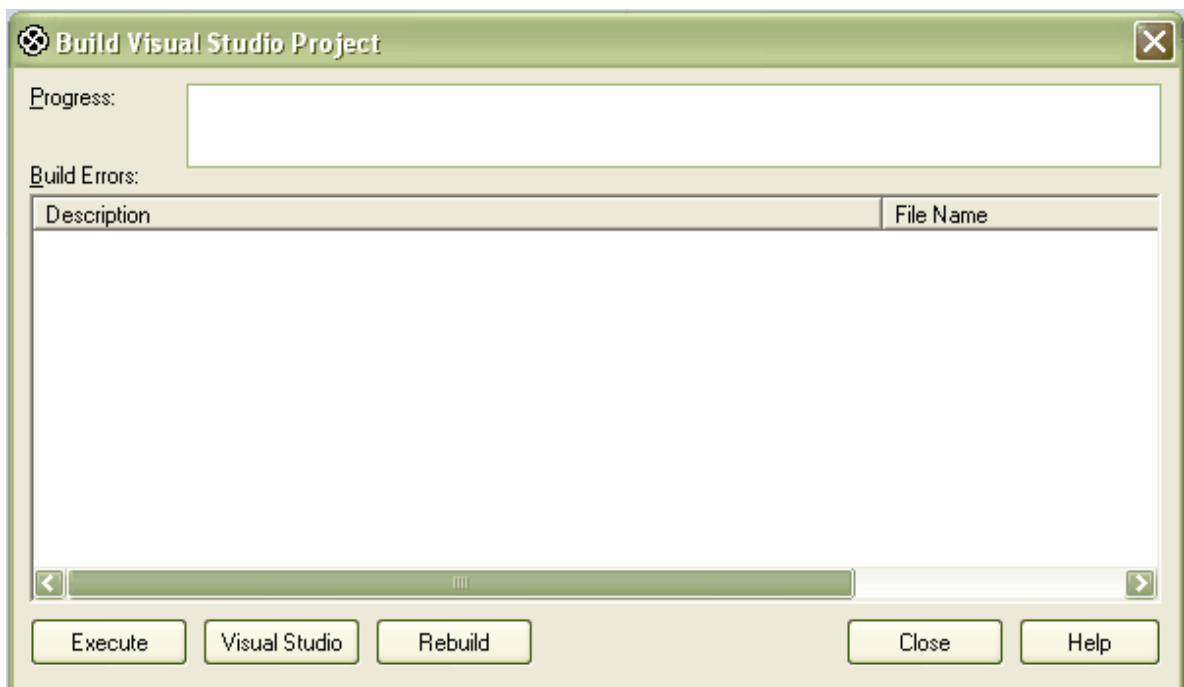


## 3.2 Build Project

It is possible to build and execute a Visual Studio .NET project from within Enterprise Architect. Building the project from within Enterprise Architect enables you to make changes to the code from the model and to determine if the changes to the code have been successful. Selecting the **Build Project** option gives you the choice of building the project and executing the project.

### 3.2.1 Build and Run a Project

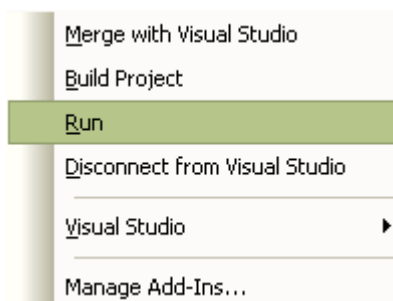
To build a Visual Studio.NET project from within Enterprise Architect, select the **Add-Ins | Build Project** menu option. The **Build Visual Studio Project** dialog displays:



When the build is successful, the **Progress** field displays the message *Build Successful*. If any errors have been encountered, the errors are listed in the **Build Errors** panel. For more information relating to build errors, see the [Build Project Errors](#) <sup>[23]</sup> topic.

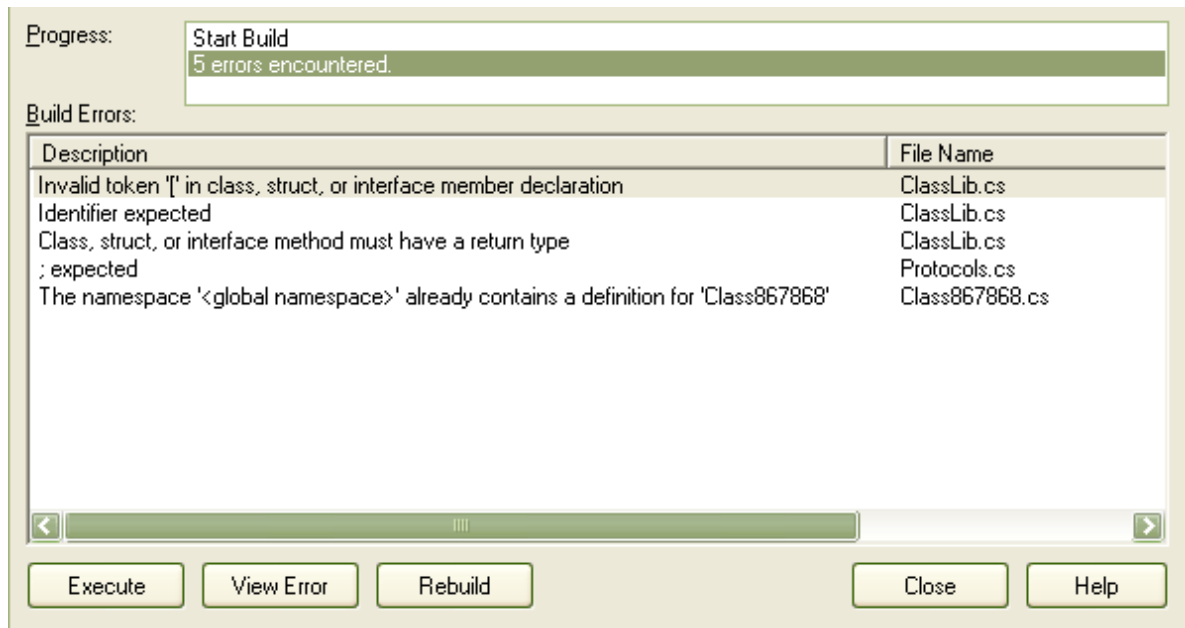
To execute the project immediately after the build, click on the **Execute** button. For the other options available on this dialog, see the [Build Dialog Options](#) <sup>[23]</sup> topic.

To execute a project from Enterprise Architect, select the **Add-Ins | Run** menu option.



### 3.2.2 Build Dialog Options

The **Build <projectname>** dialog enables you to build and execute a Visual Studio.NET project from within Enterprise Architect.

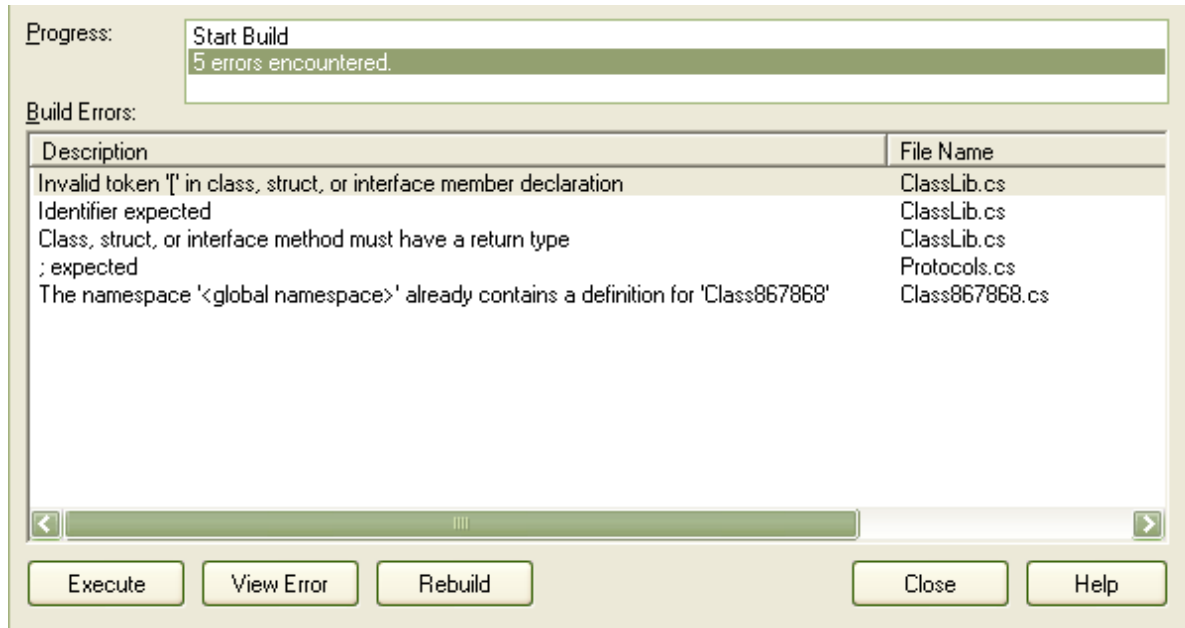


Field	Description
<b>Progress</b>	Displays the current status of the build process.
<b>Build Errors</b>	Displays information on any errors that have occurred during a build. This information includes an error description and the filename associated with the error.

Button	Description
<b>Execute</b>	Executes the project.
<b>View Error</b>	Displays the line of code with the error, in Visual Studio.NET (this button is available only when the <a href="#">Build has encountered errors</a> <sup>[23]</sup> ).
<b>Visual Studio</b>	Switches to Visual Studio.NET.
<b>Rebuild</b>	Rebuilds the project.
<b>Close</b>	Closes the <b>Build &lt;projectname&gt;</b> dialog.
<b>Help</b>	Opens the Help topic for this operation.

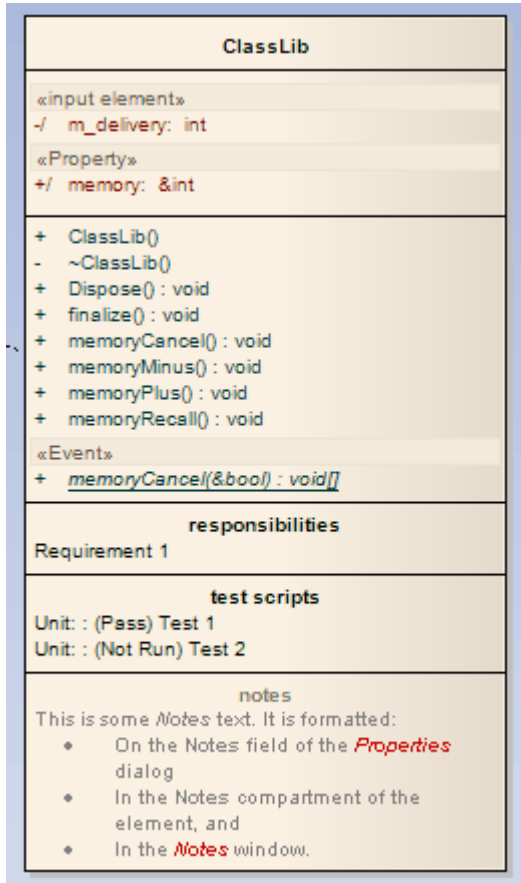
### 3.2.3 Build Project Errors

When building a project, if errors have been encountered they are listed in the **Build Errors** panel. This panel shows a description of the error and the name of the Class (in the **File name** column). To inspect the error in Visual Studio.NET, click on the Class name in the **Build Errors** panel and click on the **View Error** button; alternatively, double-click on the Class name.



### 3.3 Classes

The MDG Link for Visual Studio.NET enables flexible [creation](#)<sup>[26]</sup>, [editing](#)<sup>[28]</sup> and UML modeling of Class diagrams. In UML a Class is represented by a rectangle with at least three separate compartments. The upper compartment shows the name of the Class and, if it has one, the stereotype of the Class. The middle compartment displays the attributes of the Class, and the final compartment details the methods or operations that are available for the Class. An example of a UML Class is shown below.



The C# code that corresponds to this Enterprise Architect Class displays in Visual Studio.Net for this Class as in the diagram below.

```

ClassLib.cs
ClassLib
/// <summary>
/// This is some <i>Notes</i><b> </b>text. It is formatted:
/// <ul>
/// <li>On the <b>Notes </b>field of the <font
/// color="#ff0000"><i>Properties</i></font> dialog</li>
/// <li>In the <b>Notes </b>compartment of the element, and </li>
/// <li>In the <font color="#ff0000"><i>Notes</i></font> window.</li>
/// </ul>
/// </summary>
public class ClassLib {

    private int m_delivery;
    public int memory;

    public ClassLib()...

    ~ClassLib()...

    public virtual void Dispose()...

    public void finalize()...

    public void memoryCancel()...

    public void memoryMinus()...

    public void memoryPlus()...

    public void memoryRecall()...

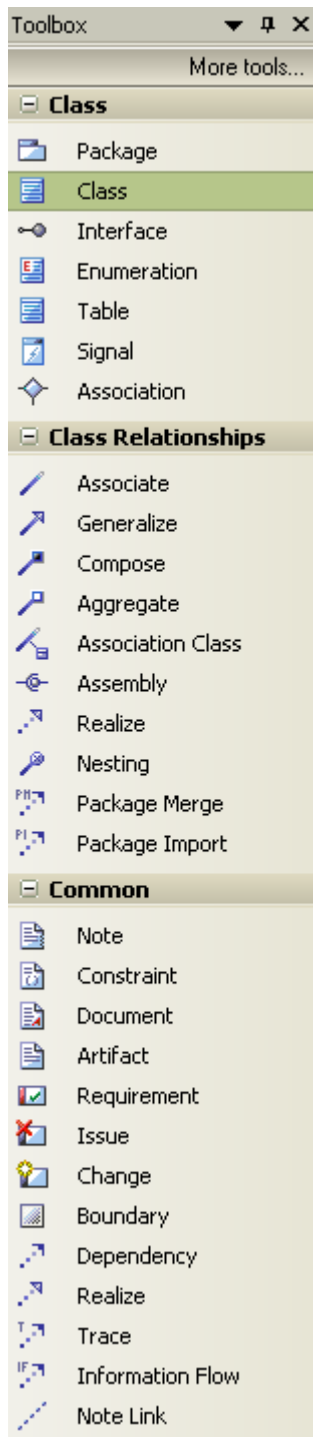
    /// <summary>
    /// This is the memoryCancel operation.
    /// </summary>
    /// <param name="memoryCancelParam"></param>
    public abstract static void memoryCancel(out bool memoryCancelParam);

} //end ClassLib

```

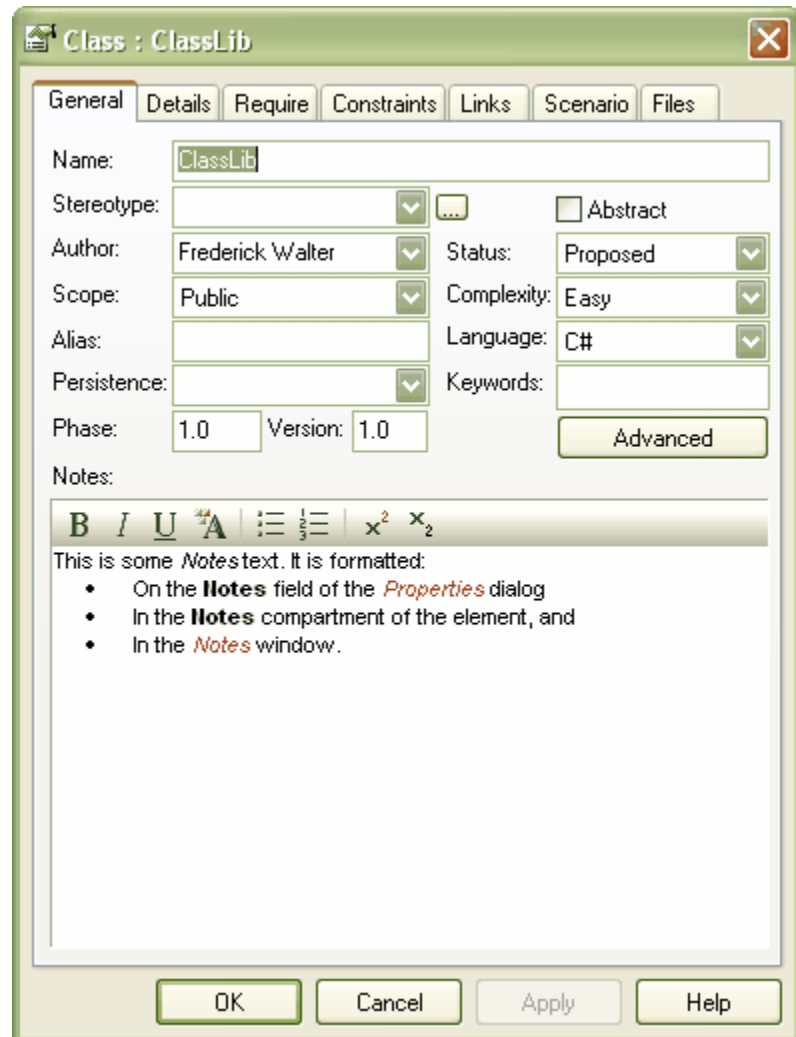
### 3.3.1 Create Class

With the MDG Link for Visual Studio.NET it is possible to create a Class either in Visual Studio.NET or in Enterprise Architect. To create a Class in Enterprise Architect select the Enterprise Architect UML **Toolbox** and click on the **More tools | Class** menu option. The **Toolbox** pages for a Class diagram display.



In the **Class** page, click on the Class element icon and drag it onto a diagram from the current package.

When you release the mouse button, the Class **Properties** dialog displays (if not, right click on the element in the diagram and select the **Properties** menu option). Use this to set the properties of the Class.



This dialog offers a range of options. The following options are available on the **General** tab:

Field	Description
<b>Name</b>	The name of the Class
<b>Stereotype</b>	Stereotypes are based on certain existing types or Classes in the meta model. They can extend the semantics, but not the structure of pre-existing types and Classes.
<b>Abstract</b>	Select to set the Class as an abstract Class, deselect to set the Class as a concrete Class
<b>Author</b>	The name of the user who created the Class.
<b>Status</b>	Flags the status of the Class.

Field	Description
Scope	Determines the visibility of the Class ( <b>public, private, protected</b> or <b>package</b> ).
Complexity	Used for project estimation ( <b>easy, medium, hard</b> ).
Persistence	The persistence that is associated with the Class, either <b>Persistent</b> or <b>Transient</b> .
Language	Determines or displays the programming language of the Class.
Alias	Defines an alternate display name for the object.
Keywords	A free text area that can be filtered in Use Case metrics and search dialogs - typically used for items such as keywords or context information.
Phase	Indicates the phase this element is to be implemented in; for example: 1, 1.1, 2.0.
Version	The version of the Class.

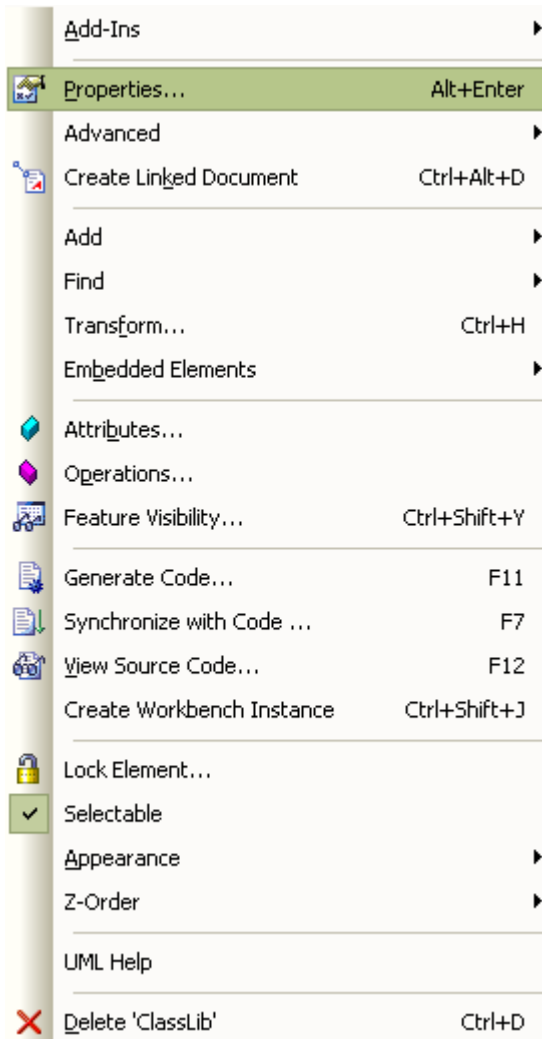
### 3.3.2 Edit Class

With the MDG Link for Visual Studio.NET you can edit the Class from within Enterprise Architect or from Visual Studio.NET. When editing the Class in Enterprise Architect you can add and delete both attributes and operations as well as define inheritance, Class dependencies and uses. For more information on adding inheritance to Classes see the [Add Inheritance to Classes](#)<sup>[31]</sup> topic.

To access the Class in Enterprise Architect and perform editing in Visual Studio.NET see the [Edit Class, Switch to Visual Studio .NET](#)<sup>[29]</sup> topic. To edit the properties of a Class in Enterprise Architect follow the steps below:

1. Select the Class to modify from either a Class diagram or the **Project Browser**.
2. Right-click on the Class. The context menu displays.





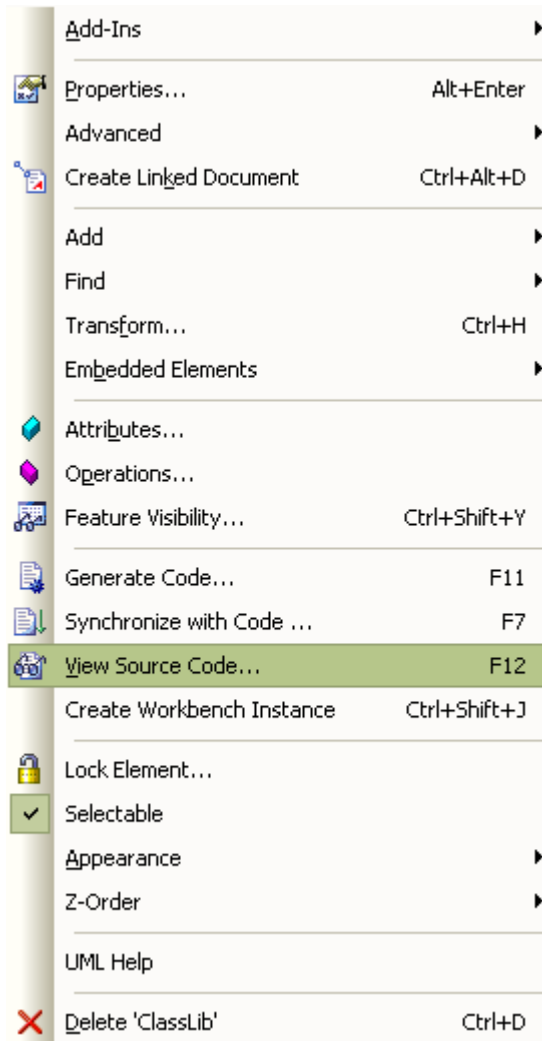
3. Select the **Properties** menu item or press **[Alt]+[Enter]** to display the **Class Properties** dialog, which has a series of options as detailed in the [Create Class](#)<sup>[27]</sup> topic.

The **Class Properties** dialog also gives you access to the Class attributes and operations. To edit these items see the [Edit Class Attributes and Operations](#)<sup>[30]</sup> topic.

### 3.3.3 Edit Class, Switch to Visual Studio .NET

To edit a Class in Visual Studio.NET from Enterprise Architect, follow the steps below:

1. Select the Class from a diagram.
2. Right-click on the Class. The context menu displays.



3. Select the **View Source Code** menu option, or alternatively press **[F12]**. Visual Studio.NET opens, displaying the Class at the start of the code ready for editing.

### 3.3.4 Edit Class Attributes and Operations

It is possible to edit the attributes and operations of a Class from within Enterprise Architect.

1. Access the Class **Properties** dialog, following the steps outlined in the [Edit Class](#) <sup>28</sup> topic. Click on the **Detail** tab. The dialog displays as shown below. .

The screenshot shows the 'Details' tab of a class editor. It includes the following elements:

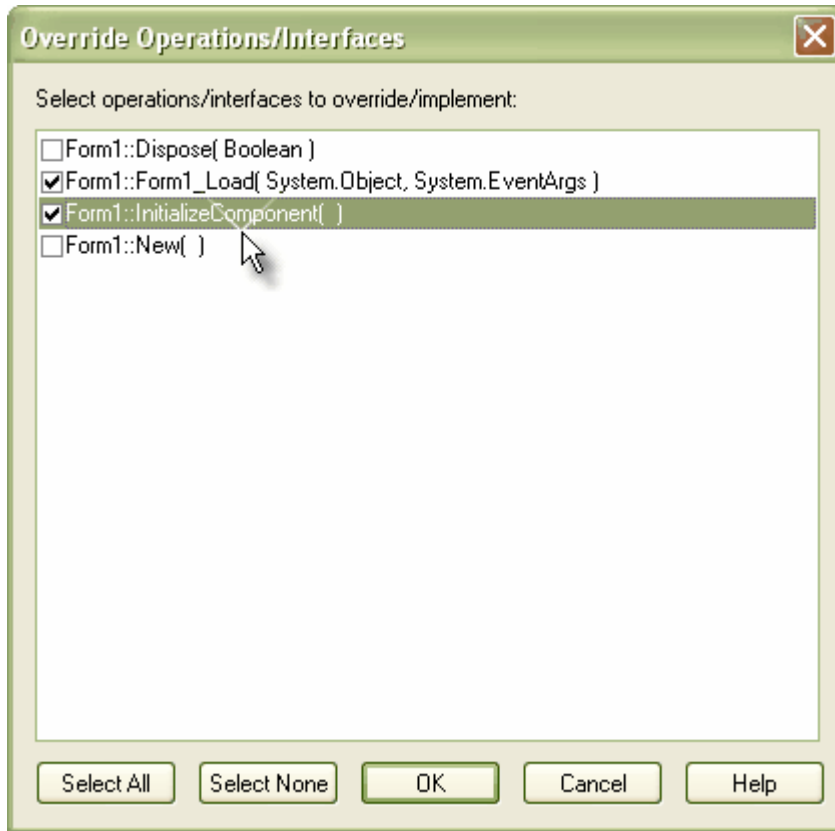
- Cardinality:** A dropdown menu.
- Visibility:** A dropdown menu.
- Buttons:** 'Attributes...', 'Operations...', and 'Collection Classes...'.
- Concurrency:** A group box containing four radio buttons: 'Sequential', 'Guarded', 'Active', and 'Synchronous'.
- Templates:** A section with a 'Type' dropdown, 'Add', 'Edit', and 'Delete' buttons, and a table with columns 'Parameter', 'Type', and 'Default'.
- Arguments:** A text input field.
- Footer:** 'OK', 'Cancel', 'Apply', and 'Help' buttons.

2. To set the attributes of a Class click on the **Attributes** button; for more information on the options related to editing attributes go to the [Attributes](#) topic.
3. To set the operations of a Class click on the **Operations** button; for more information on the options related to editing operations go to the [Operations](#) topic.

### 3.3.5 Add Inheritance to Classes

Adding inheritance between Classes in the MDG Link for Visual Studio.NET from Enterprise Architect is a simple procedure. To achieve this follow the steps below.

1. On a diagram, locate the Class or Classes involved in the operation.
2. Select the *Generalize* connection from the **Toolbox**.
3. Connect the child Class to the parent Class.
4. Select the **Element | Advanced | Overrides & Implementations** menu option. The **Override Operations/Interfaces** dialog displays.
5. Select the operations/interfaces to override/implement.

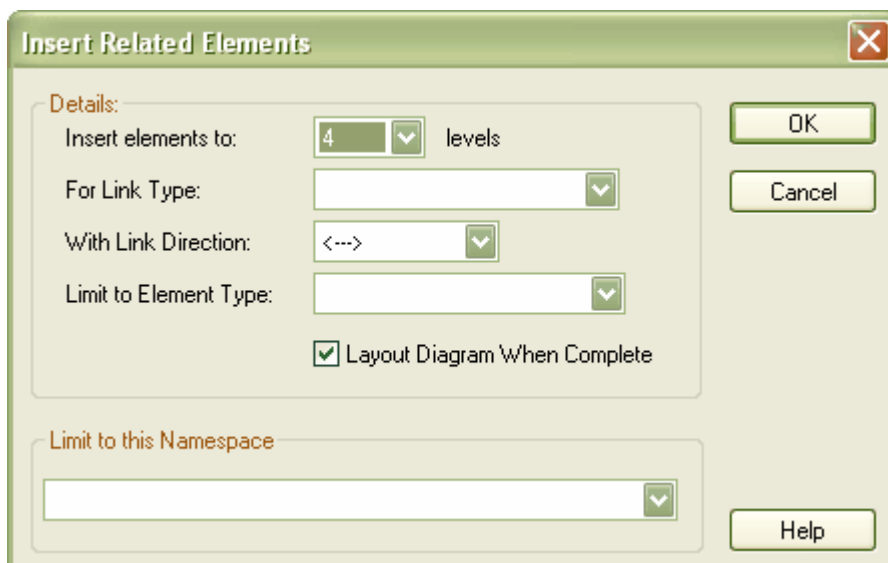


- To update the model to the source code follow the steps outlined in [Synchronize Code and Model.](#)<sup>[44]</sup>

### 3.3.6 Add Class and Find Association Links

One of the powerful options available in MDG Link for Visual Studio.NET is the ability to add one Class to a diagram and to find the relationships between Classes that link to the original Class. To achieve this follow the steps below.

- Create a new diagram and in the **Project Browser** find the Class that you are interested in.
- Drag this Class onto the diagram workspace and paste it as a *Simple Link*.
- Right-click on the Class to display its context menu, and select the **Add | Related Elements** menu option. The **Insert Related Elements** dialog displays:



Use the options on this dialog to bring the related Classes into the diagram, giving you a picture of the relationships between the original Class and other related Classes.

For more information relating to the options of this dialog see the [Insert Related Elements](#) topic.

## 3.4 Code

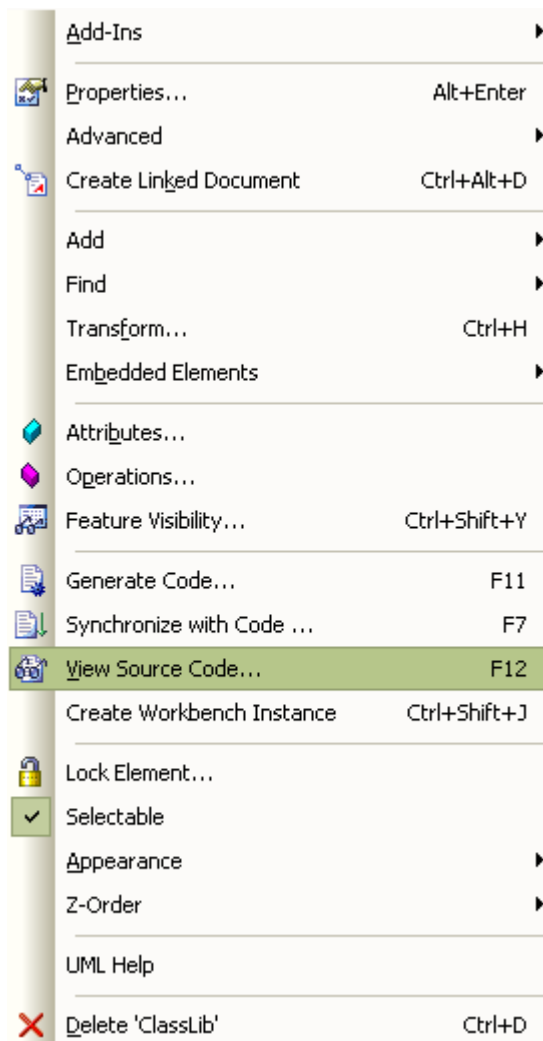
The MDG Link for Visual Studio.NET enables flexible [creation](#)<sup>[26]</sup>, [editing](#)<sup>[28]</sup> and UML modeling of Class diagrams.

### 3.4.1 Edit Code

The MDG Link for Visual Studio.NET adds extra functionality to the code generation abilities of Enterprise Architect. In addition to [generation of code](#)<sup>[40]</sup> (forward engineering) and [synchronization of code](#)<sup>[44]</sup> (reverse engineering) the MDG Link for Visual Studio.NET enables you to quickly edit the source code in Visual Studio.NET.

To edit code follow the steps below:

1. In the **Diagram View**, right-click on the Class to edit. The context menu displays



2. Select the **View Source Code** menu item or press **[F12]** to open Visual Studio.NET to edit the Class code.

(You can also edit from the **Project Browser** by right-clicking the required item (which can be a Class, attribute or operation) to display its context menu, then pressing **[F12]** or selecting the **View Source Code** menu option to edit the code.)

```

ClassLib.cs
ClassLib

public class ClassLib {

    private int m_delivery;
    public int memory;

    public ClassLib() {...}

    ~ClassLib() {...}

    public virtual void Dispose() {...}

    public void finalize() {...}

    public void memoryCancel() {...}

    public void memoryMinus() {...}

    public void memoryPlus() {...}

    public void memoryRecall() {...}

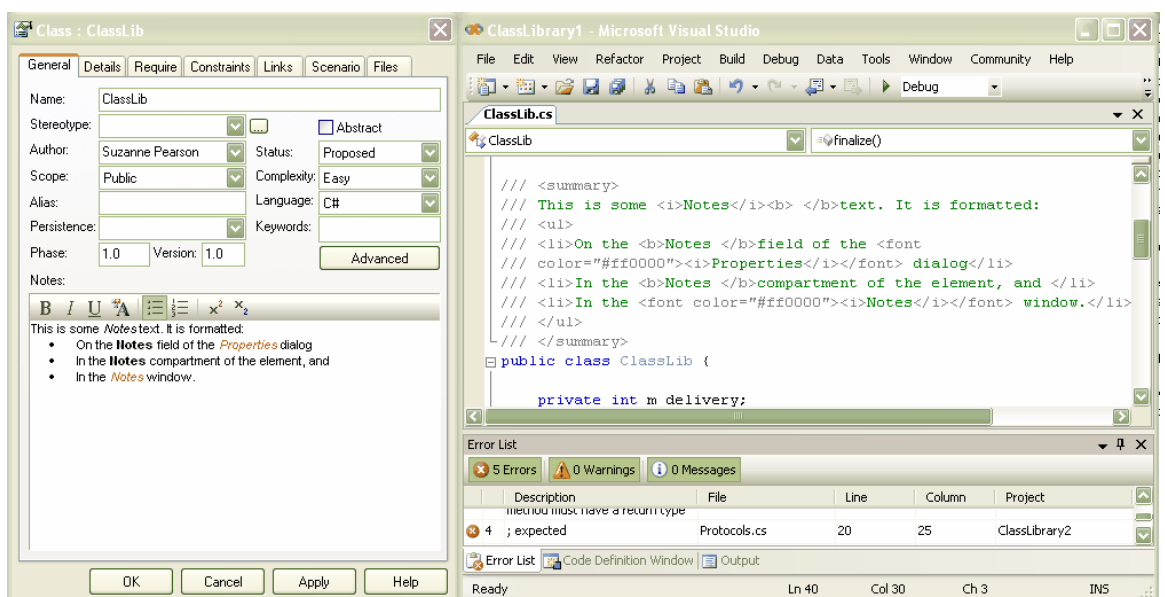
    /// <summary>
    /// This is the memoryCancel operation.
    /// </summary>
    /// <param name="memoryCancelParam"></param>
    public abstract static void[] memoryCancel(out bool memoryCancelParam);
} //end ClassLib

```

### 3.4.2 Add Code Comments

To add comments to code from Enterprise Architect, follow the steps below:

1. On a diagram right-click on the Class or, in the **Project Browser**, right-click on the Class or method. Select the **Properties** context menu option. The appropriate **Properties** dialog displays.
2. In the **Note** field, type the comments.



3. Click on the **Apply** and **OK** buttons, then click on the Class or method and press either **[F7]** (Synchronize with code) or **[F11]** (Generate code) to update the code with the changes.

**Note:**

Comments in the Enterprise Architect **Notes** fields can be formatted using Rich Text Notes formatting commands. These commands display in-text in the comment code in Visual Studio .NET.



## 3.5 Diagrams

UML Diagrams are collections of project elements laid out and inter-connected as required. Enterprise Architect supports several kinds of UML diagrams as well as custom extensions. For full details, see [UML Diagrams](#).

### 3.5.1 Format a Diagram

Formatting a UML Class diagram does not change the functionality of your Classes, but instead creates a more readable diagram. A facility is provided by Enterprise Architect to layout diagrams automatically. This creates a reasonable tree based structure from the Class diagram elements and relationships in a diagram. Owing to the complexity of many Class diagrams, the results might require some manual 'tweaking'.

To format your UML Class diagram, follow the steps below:

1. Select a diagram.
2. Select the **Diagram | Layout Diagram** menu option

For more information on the options for laying out a UML Class diagram, see the [Layout a Diagram](#) topic.

## 3.6 Round Trip Engineering

The MDG Link for Visual Studio.NET round-trip engineering process enables you to model your application in UML 2.1 notation, then generate ([forward engineer](#)<sup>[40]</sup>) the code elements to Visual Studio.NET based on the model, perform modifications and implement the code as necessary, and then [reverse engineer](#)<sup>[42]</sup> that code back into the Enterprise Architect model.

This results in consistency between the model and the external code base, and can be achieved with a merge at the touch of a button. The MDG Link for Visual Studio.NET also enables you to merge the project. The merge options include the option to both forward and reverse engineer Classes at the same time to completely [synchronize the code and the model](#)<sup>[44]</sup>.

### 3.6.1 Merge Options

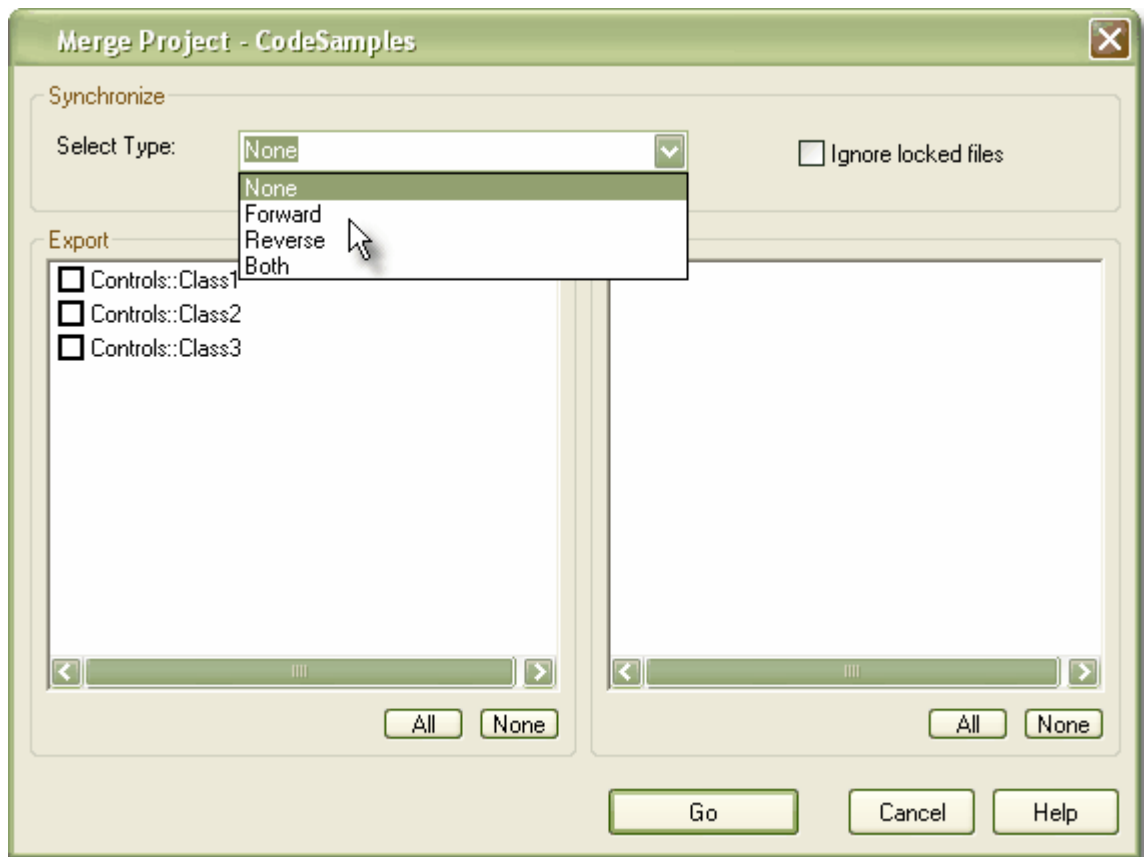
Merging enables you to reverse engineer code from a Visual Studio.NET project, or to generate code from an Enterprise Architect model into a Visual Studio.NET project. Merging interrupts the normal processes involved in forward and reverse engineering, enabling a greater level of control than is available in the standard versions of Enterprise Architect. Performing a merge enables you to:

- [Choose the filename](#)<sup>[42]</sup> for new Classes created in Enterprise Architect, to assign more than one Class to the same file name
- [Export](#)<sup>[41]</sup> selected Classes, to export code only on selected Classes
- [Import](#)<sup>[43]</sup> selected Classes, to import code only on selected Classes
- [Synchronize the Model](#)<sup>[44]</sup> and the source code in one simple step; a synchronized merge forward engineers the model from Enterprise Architect into Visual Studio.NET and then reverse engineers the code from Visual Studio.NET into the Enterprise Architect model in one simple step, enabling the model and the code to accurately represent each other
- Optionally ignore locked files.

### 3.6.2 Merge Project Dialog Options

To perform a merge you select the **Add-Ins | Merge with Visual Studio** menu option. The **Merge Project** dialog displays.

The **Merge Project** dialog enables you to connect to and disconnect from a single Visual Studio.Net project.



Field	Description
<b>Select Type</b>	<p>Click on the drop-down arrow and select one of the four options for merging.</p> <ul style="list-style-type: none"> <li>• <b>None</b> - select to not perform synchronization</li> <li>• <b>Forward</b> - select to generate code from the Enterprise Architect model into Visual Studio.NET</li> <li>• <b>Reverse</b> - select to bring code out of Visual Studio.NET into the Enterprise Architect model.</li> <li>• <b>Both</b> - select to perform the operation of forward engineering and then the operation of reverse engineering, which fully synchronizes the model and the code.</li> </ul> <p><b>Note:</b></p> <p>Synchronization only applies to Classes that appear in both the model and the source code in Visual Studio.NET.</p>
<b>Ignore Locked files</b>	Select to ignore locked files in the import or export.
<b>Export</b>	<p>Lists Classes that are present only within the model and that are not currently included in the code in Visual Studio.NET.</p> <p>You select Classes from this list for export; if a Class is not selected it is not included in the export.</p> <p>Click on the <b>All</b> button to select all of the Classes in the list. Click on the <b>None</b> button to clear all selections in the list.</p>
<b>Import</b>	<p>Lists Classes that are present only within the code in Visual Studio.NET and that are not currently included in the model.</p> <p>You select Classes from this list for import. If a Class is not selected it is not included in the import.</p>

Field	Description
	Click on the <b>All</b> button to select all of the Classes in the list. Click on the <b>None</b> button to clear all selections in the list.
Buttons	Description
<b>Go</b>	Executes the merge.
<b>Cancel</b>	Cancels the operation.
<b>Help</b>	Opens the help file.

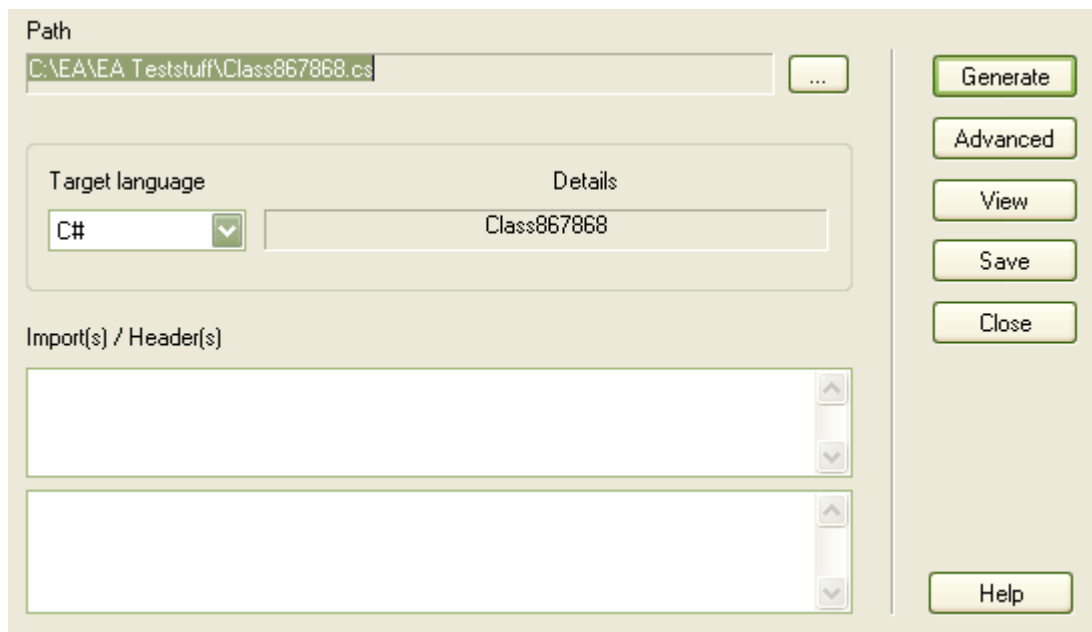
### 3.6.3 Forward Engineering

Code Generation (forward engineering) generates and updates code from the UML model and places it into Visual Studio.NET. When used to generate a Class created purely in Enterprise Architect, the code that is created in Visual Studio.NET consists of constructors and destructors as well as *get* and *set* methods. This leaves generation of the business operations of the code up to you. When updating, or synchronizing, existing code from the model, you update the data that differs between the code and model; you do not regenerate all the code.

You can generate code in several ways with the Enterprise Architect MDG Link for Visual Studio.NET, using the **Merge** option on the **Add-Ins** menu as well as from the context menu of a Class.

#### 3.6.3.1 Forward Engineer from a Class

The **Generate Code** dialog enables you to control how your source code is generated. Normally you access this dialog from the context menu of a single Class or interface. Right-click on the Class or interface and select the **Generate Code** context menu option. Alternatively, select the Class or interface and press **[F11]**.



This dialog enables you to define:

- The **Path** where the source is to be generated. Click on the [...] (Browse) button to display a file browser dialog, which defaults to the path of the current Visual Studio.NET Solution.
- The **Target language** for generation: select the language to generate - this becomes the permanent option for that Class, so change it back if you only want to do one pass in another language.
- **Advanced** settings: note that the settings you make here only apply to the current Class.
- **Import** statements: two areas in which to define import statements (or #includes, #defines and macros in Visual C++). For Visual C++ this area is placed in the header file.
- **Generate**: click on this button to generate your source code; a **Progress** dialog displays messages as the

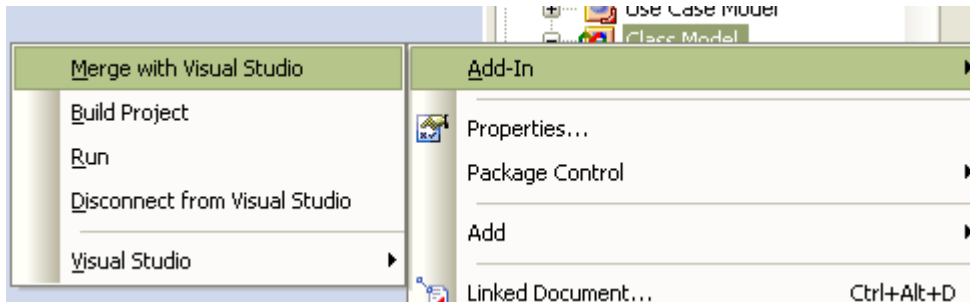
generation proceeds.

- **View:** click on this button to view the generated source code in Visual Studio.NET.

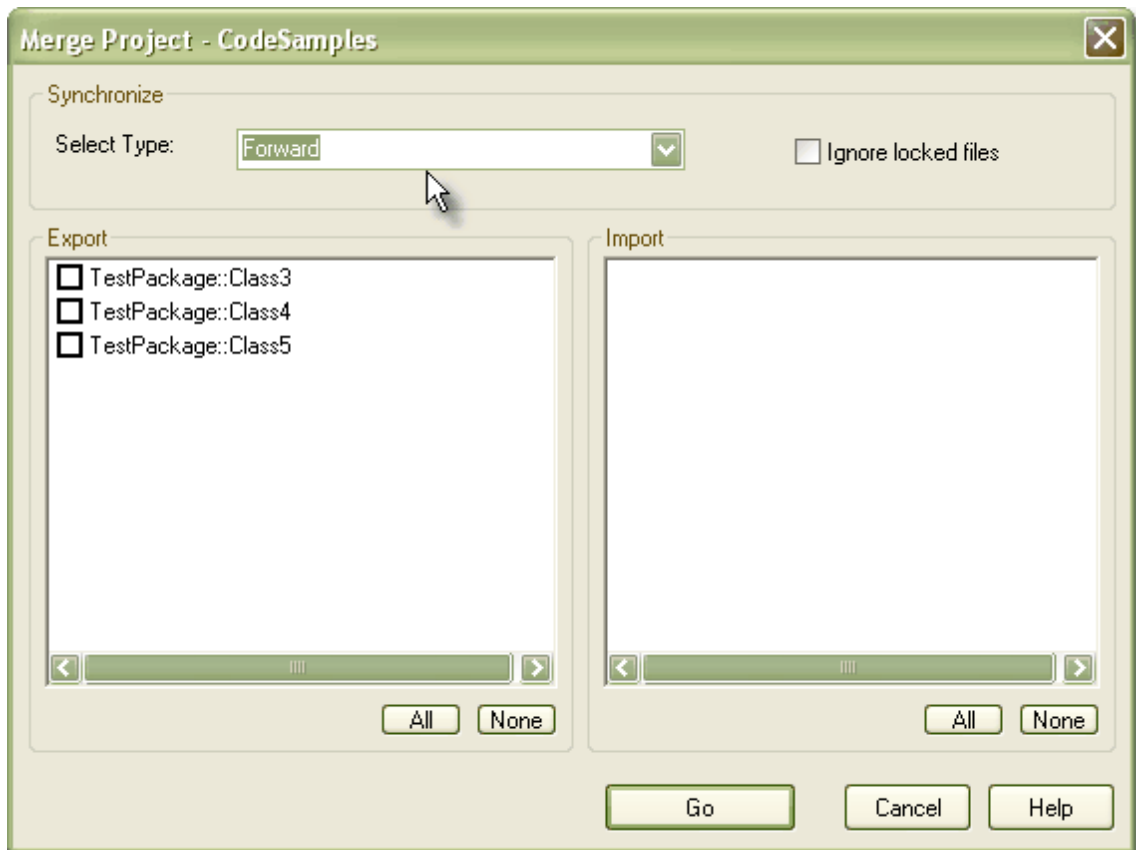
### 3.6.3.2 Forward Engineer with a Merge

To generate code with a merge once [a link has been created](#)<sup>[15]</sup> to a Visual Studio.NET Project, follow the steps below.

1. Select the **Add-In | Merge with Visual Studio** context menu option.



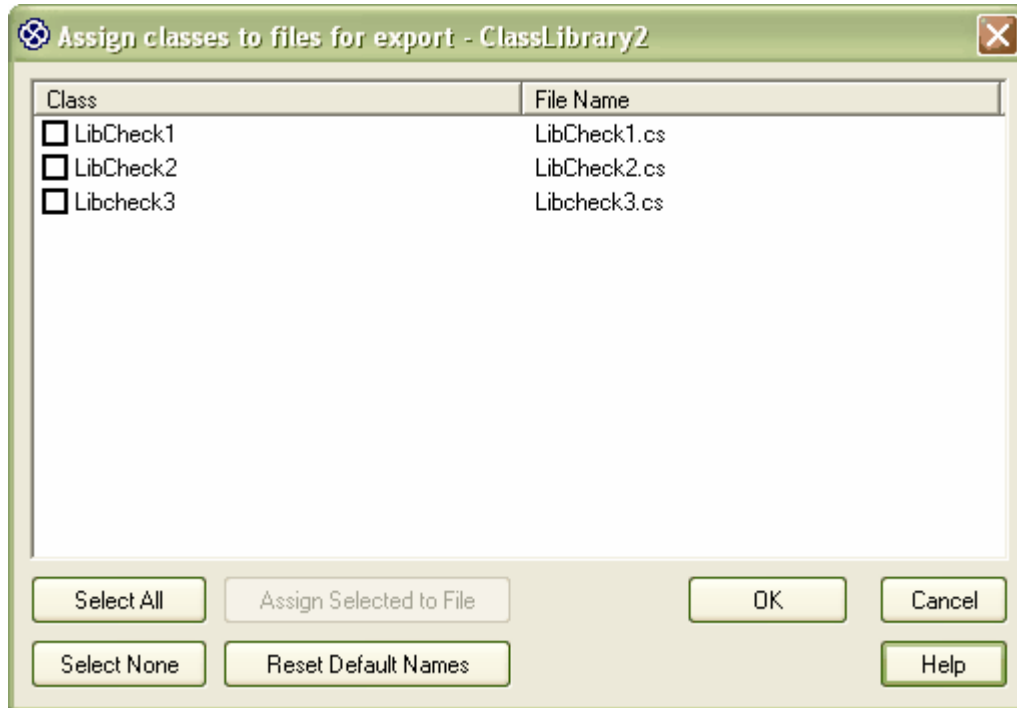
The **Merge Project** dialog displays.



2. In the **Synchronize** panel of the dialog, select **Forward** to update Classes contained in the code from corresponding elements contained in the model.
3. The items in the **Export** section apply to elements that currently exist in the model but do not exist in the code. Select the checkbox against each Class to export into the code, or click on the **All** button to select all the Classes.
4. Click on the **Go** button to forward-engineer the code. If the forward engineer includes new Classes, Enterprise Architect prompts you to assign a filename for the new Classes. For more information on assigning new Classes see the [Assign Classes to Files for Export](#)<sup>[42]</sup> topic.

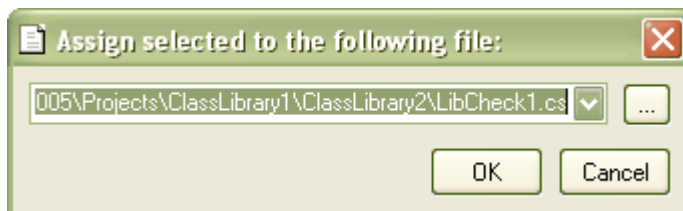
### 3.6.3.3 Assign Classes to Files for Export

When a new Class is created in Enterprise Architect and a merge is performed, you can assign the filename for the Classes to allow multiple Classes to be assigned to the same filename. The **Assign classes to files for export** dialog enables you to select the Classes to add into files.



To assign a filename to a Class follow the steps below:

1. Click on the Class to assign a filename to, or press **[Ctrl]** and click on a number of Classes.
2. To assign the file name click on the **Assign Selected to File** button. Enterprise Architect prompts you for a file path for the Class to be saved.



3. Overtyping the path name, or using the **[...]** (Browse) button to search for a file location. Click on the **OK** button to assign the file.
4. To cancel the assigned filename and return to the previous filename, click on the **Reset Default Names** button.
5. When you have finished assigning the filenames, click on the **OK** button. The **Merge Progress** window displays, with status messages as the merge proceeds.

If a Class is not assigned, a **Save As** dialog displays to prompt you for a file location.

### 3.6.4 Reverse Engineering

Reverse engineering creates or updates the UML model from the Visual Studio.NET source code. This action enables you to convert a legacy system to a model to examine the architecture of the existing code. When updating, or synchronizing, an existing model from code, you update the data that differs between the code and model; you do not recreate the entire model.

You can reverse engineer code to a model in several ways with the MDG Link for Visual Studio.NET, in Enterprise Architect from a Merge operation or from the context menu of a Class.

### 3.6.4.1 Reverse Engineer from a Visual Studio.NET Source Class

To reverse engineer (import Visual Studio .NET source code into) an existing model Class, follow the steps below:

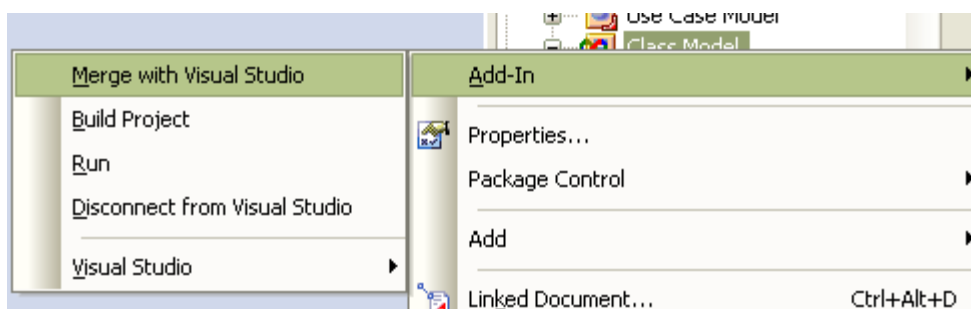
1. In the diagram or the **Project Browser** window, right-click on the Class to be updated from the code. The context menu displays.
2. Select the **Synchronize with code** menu option, or press **[F7]**.

Enterprise Architect automatically updates the model Class from the Visual Studio .NET code.

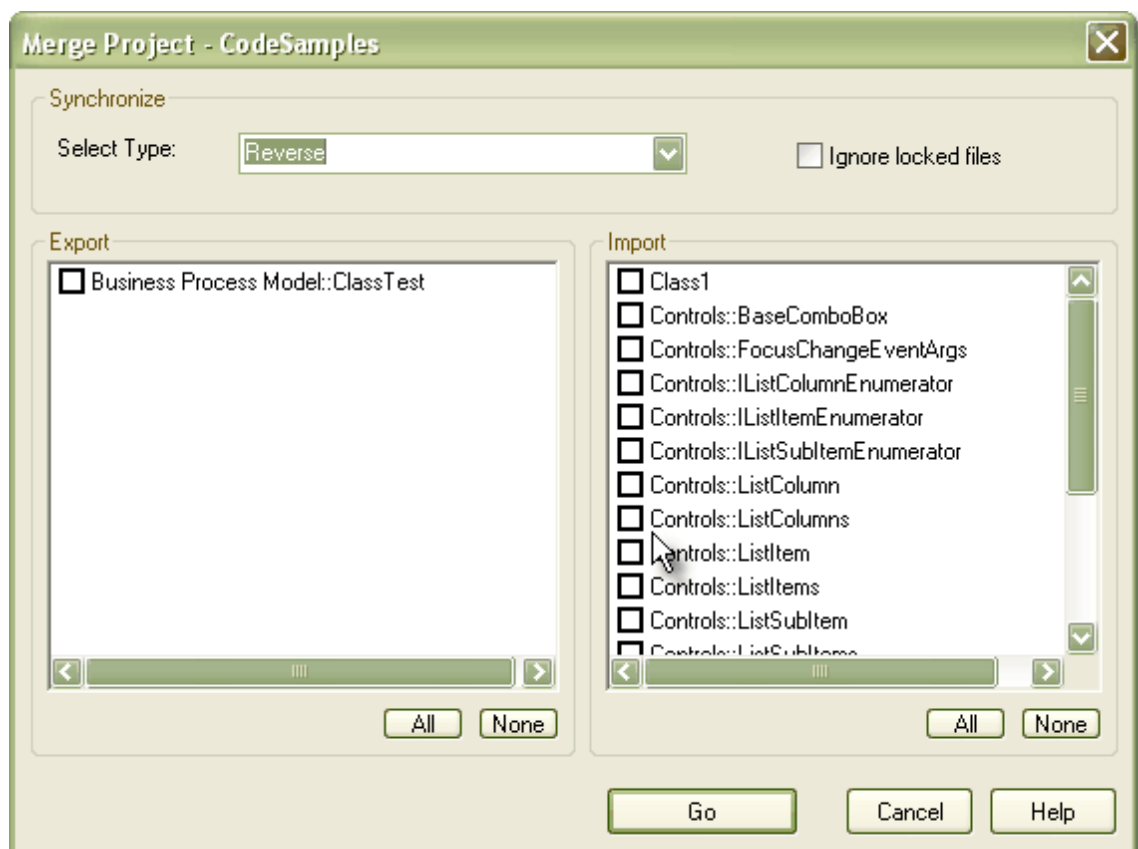
### 3.6.4.2 Reverse Engineer with a Merge

To generate code with a merge once [a link has been created](#) <sup>151</sup> to a Visual Studio.NET Project, follow the steps below:

1. Right-click on the required package for the linked Visual Studio.NET project.
2. Select the **Add-In | Merge with Visual Studio** menu option.



The **Merge Project** dialog displays:



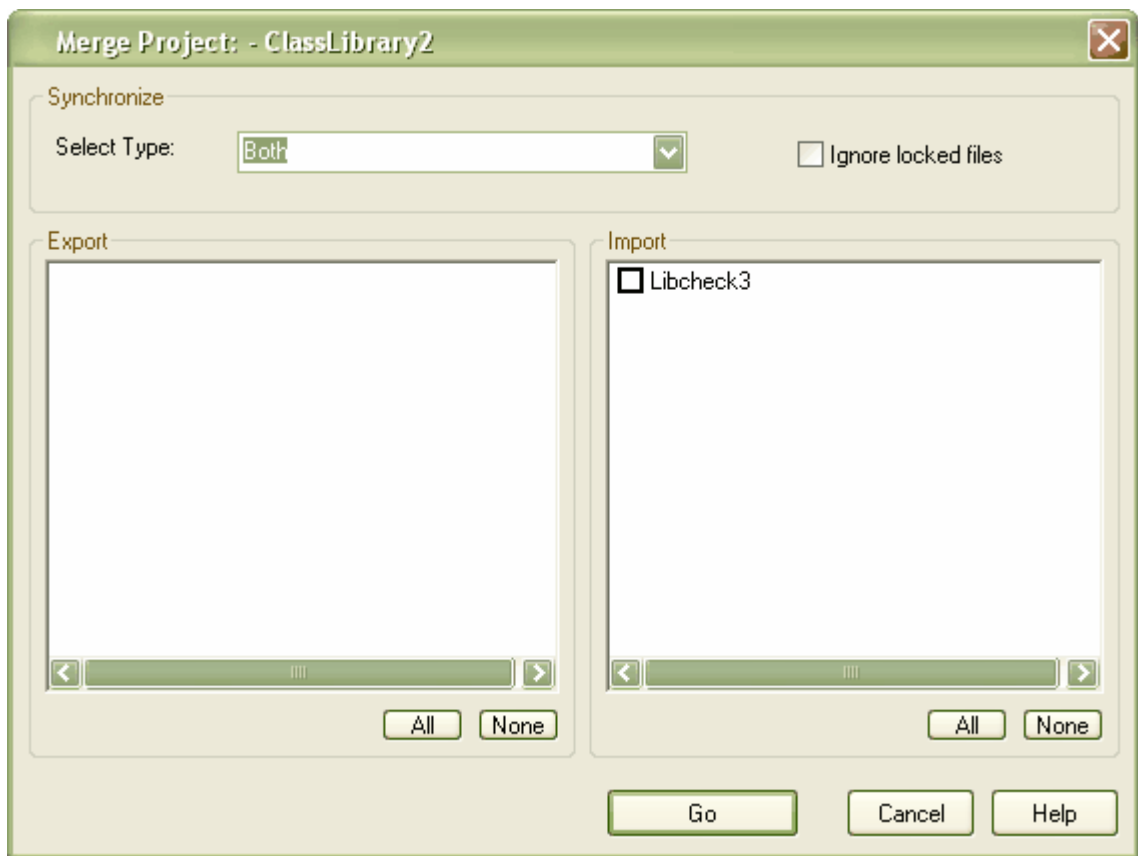
3. In the **Select Type** field click on the drop-down arrow and select **Reverse**, to update elements contained in the model from Classes contained in the code.

4. The items in the **Import** section are elements that currently exist in the model but do not exist in the code. Select the checkbox against each element to be imported into the code, or click on the **All** button to select all of the elements.
5. Click on the **Go** button to reverse-engineer the code. A **Merge Status** window displays, showing a series of status messages as the merge progresses. This window automatically closes when the merge is complete.

### 3.6.5 Synchronize Code and Model

Synchronizing the code with the model is a simple task once [a link has been created](#)<sup>[15]</sup> to a Visual Studio.NET Project. You can perform a merge at any time by following the steps below:

1. Right-click on the connected package in the Enterprise Architect **Project Browser** window. The context menu for the package displays.
2. Select the **Add-In | Merge with Visual Studio** menu option. The **Merge Project** dialog displays.



3. In the **Select Type** field, click on the drop-down arrow and select **Both**.
4. If there are new Classes listed in the **Export** and/or **Import** panels, select the checkbox against each Class to add to the model or code, as required.
5. Click on the **Go** button.

If you selected new Classes to be exported from the model to code, the **Assign classes to files for export** dialog displays. [Assign the Classes](#)<sup>[42]</sup> as appropriate.

6. The **Merge Progress** screen displays, showing messages as the forward engineering (export) and reverse engineering (import) are performed. This screen automatically closes when the merge is complete.



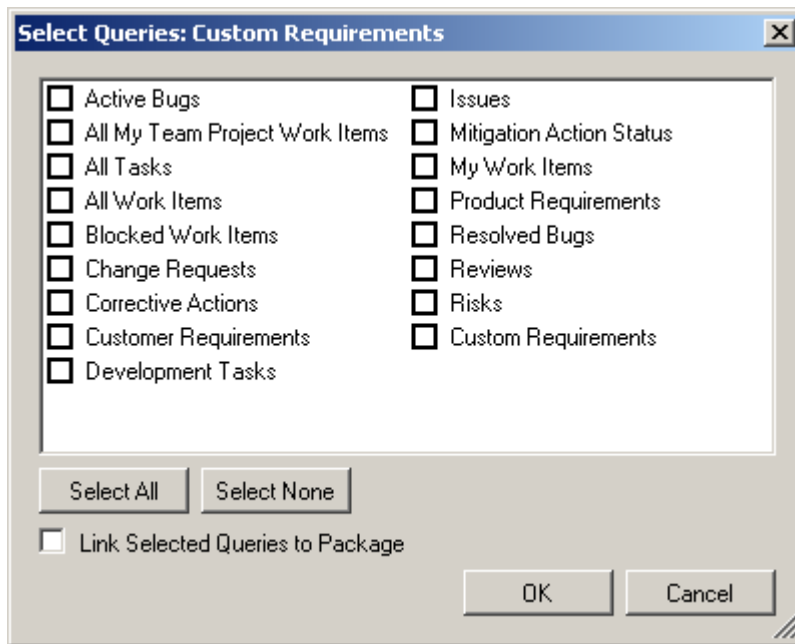
## 3.7 Import and Synchronize TFS Work Items

MDG Link for Visual Studio .NET enables you to import and synchronize work items from Team Foundation Server. It also enables you to [add, view, edit and unlink work items](#)<sup>[46]</sup> that have been linked against a UML element, and to [map work item fields](#)<sup>[48]</sup> against Enterprise Architect elements and element properties.

### Import Work Items

To import work items, follow the steps below:

1. In the **Project Explorer**, right-click on the required package and select the **Add-In | Visual Studio | Team Foundation Server | Import Work Items** context menu option. The **Select Queries** dialog displays.



2. Click on the checkbox against each query for which to import work items from Team Foundation Server.
3. Click on the **Link Selected Queries to Package** checkbox.
4. Click on the **OK** button. The work items in the selected queries are imported and, if the **Link Selected Queries to Package** checkbox is selected, the queries are stored against the package.

You can return to this dialog to add and remove queries for a package.

### Synchronize With TFS

When a package has stored queries against it, new elements are automatically added and existing elements updated, based on each query.

#### Note:

Elements are never deleted.

To manage the mapped fields for a work item linked against a UML element, right-click on the required package in the **Project Explorer** and select the **Add-In | Visual Studio | Team Foundation Server | Manage Mapped Fields** context menu option. The **Configure Field Mappings** dialog displays.

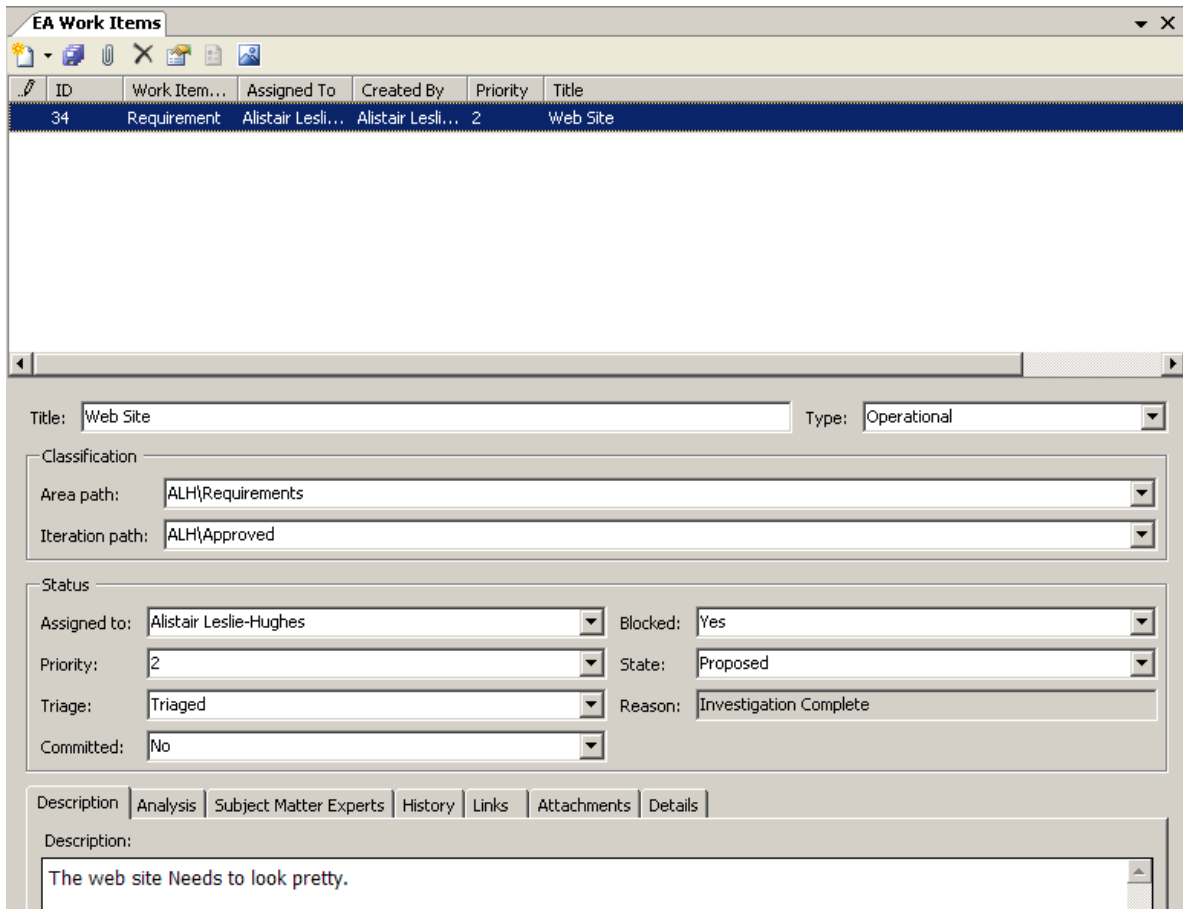
Field/Button	Description
<b>Work Item Type</b>	Click on the drop-down arrow and select the type of work item to configure.
<b>Don't create Elements for this work item type</b>	Select this checkbox to prevent MDG Link for Visual Studio .NET from creating an Enterprise Architect element if the query returns a work item of the specified type.
<b>UML Element</b>	If MDG Link for Visual Studio .NET is to create an element, click on the drop-down arrow and select the type of element to create.
<b>Element Stereotype</b>	If required, click on the drop-down arrow and select the stereotype to apply to the element.
<b>Work Item Field</b>	If required, click on a work item field against which to configure an Enterprise Architect property.
<b>Configure</b>	Click on this button to select an Enterprise Architect property to configure against the selected work item field.
<b>Clear</b>	Clear the currently-selected mapping.
<b>Default</b>	Apply the default mapping for this work item type.

### 3.7.1 Maintain Work Items

The **EA Work Items** tab enables you to add and update work items against a selected element, configure the connection to the Team Foundation Server, and connect to or disconnect from the server.

To display the **EA Work Items** tab, right-click on the required element in the **Project Browser** and select the

Add-In | Visual Studio | Team Foundation Server | Maintain Work Items context menu option.

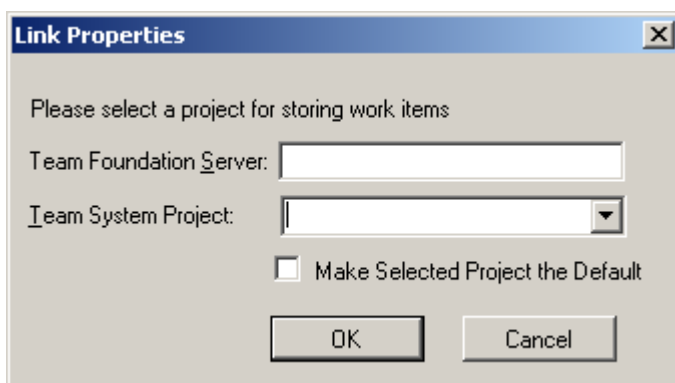


The EA Work Items tab has a toolbar. The functions provided by the toolbar icons are, from left to right:

- **Add New Work Item** - create a new work item linked to the currently-selected element
- **Save** - Save all changes to the current work item (selected from the panel underneath the toolbar)
- **Link** - Link one or more work items to the selected element
- **Remove** - unlink selected work items from the selected element
- **Properties** - Define the current Team Foundation Server connection properties; displays the [Link Properties](#) dialog
- **Connect** - Connect to the Team Foundation Server
- **Disconnect** - Disconnect from the Team Foundation Server.

### Connection Properties

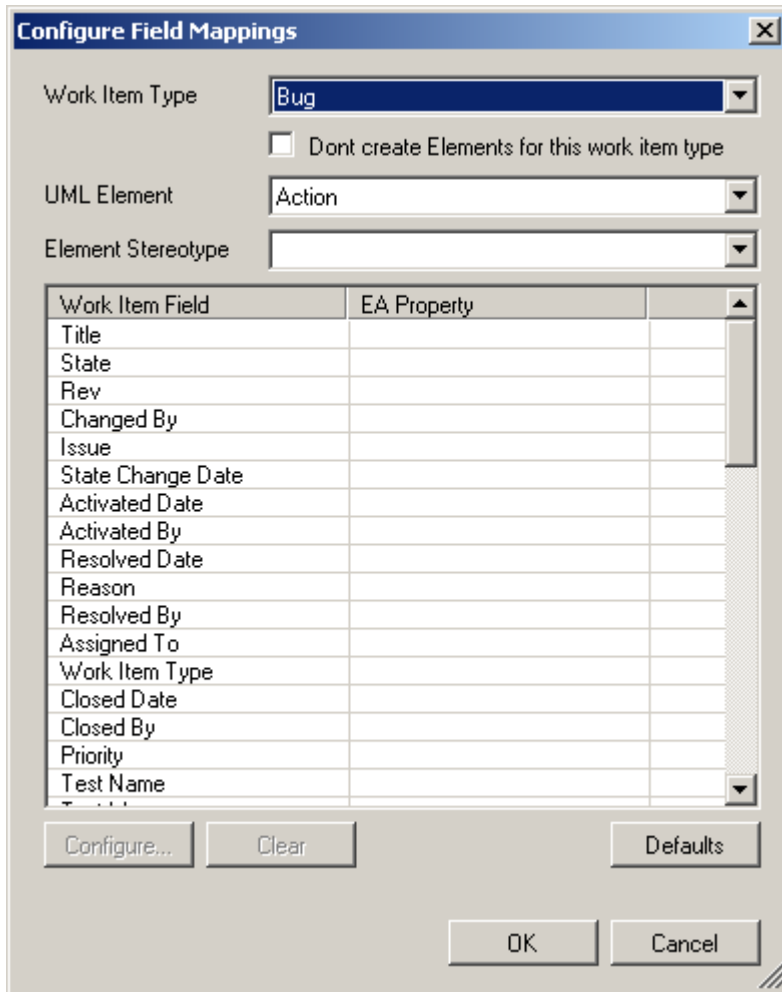
The Link Properties dialog enables you to configure the connection to the Team Foundation Server.



1. In the **Team Foundation Server** field, type the name of the server to connect to.
2. In the **Team System Project** field, type or select the name of the project to connect to.
3. Select the **Make Selected Project the Default** checkbox to make the specified project the default.
4. Click on the **OK** button.

### 3.7.2 Manage Mapped Fields

To manage the mapped fields for a work item linked against a UML element, right-click on the required package in the **Project Explorer** and select the **Add-In | Visual Studio | Team Foundation Server | Manage Mapped Fields** context menu option.



Field/Button	Description
<b>Work Item Type</b>	Click on the drop-down arrow and select the type of work item to configure.
<b>Don't create Elements for this work item type</b>	Select this checkbox to prevent MDG Link for Visual Studio .NET from creating an Enterprise Architect element if the query returns a work item of the specified type.
<b>UML Element</b>	If MDG Link for Visual Studio .NET is to create an element, click on the drop-down arrow and select the type of element to create.
<b>Element Stereotype</b>	If required, click on the drop-down arrow and select the stereotype to apply to the element.
<b>Work Item Field</b>	If required, click on a work item field against which to configure an Enterprise Architect property.

<b>Field/Button</b>	<b>Description</b>
<b>Configure</b>	Click on this button to select an Enterprise Architect property to configure against the selected work item field.
<b>Clear</b>	Clear the currently-selected mapping.
<b>Default</b>	Apply the default mapping for this work item type.

# Index

## - A -

- Acknowledgement
  - Of Trademarks 8
- Activate
  - MDG Link For Visual Studio .NET 12
- Add-In Menu
  - Access 18
  - Options 18
- Assign
  - Class To File For Export To Code 42
- Association
  - Between Classes, Find 32
- Attribute
  - Edit 30
  - Edit Code In Visual Studio .NET 20
  - Edit Properties in Enterprise Architect 20

## - B -

- Build Project
  - Dialog Options 23
  - Errors 23
  - Within Enterprise Architect 22

## - C -

- Class
  - Add Inheritance 31
  - Add Related Elements 32
  - Assign To File For Export To Code 42
  - Attributes, Edit 30
  - Code In Visual Studio .NET 25
  - Create 26
  - Edit Attributes And Operations 30
  - Edit Code In Visual Studio .NET 18
  - Edit In Enterprise Architect 28
  - Edit Properties in Enterprise Architect 18
  - Edit, Switch To Visual Studio .NET 29
  - Element In Enterprise Architect 25
  - Find Associations 32
  - Generate Code From, In Enterprise Architect 40
  - Generate Code From, Via MDG Link Merge 41
  - Locate In Project Browser 18
  - Operations, Edit 30
  - Properties 26
  - Toolbox Page 26

- Code
  - Add Comments To 35
  - Assign Class To File For Export To 42
  - Edit in Visual Studio .NET 34
  - Edit, Introduction 34
  - Generate From Class Element 40
  - Generate From Class Element, Via MDG Link Merge 41
  - Generation 40
  - Synchronize With Model 44

- Comments
  - Add To Code 35
- Compiled April 28 2008 2

- Connect
  - To TFS 46

- Connection
  - Properties 46
  - To TFS, Configure 46

- Copyright 4

- Create
  - Model From Visual Studio .NET Code 17
  - Visual Studio .NET Code From Model 17

## - D -

- Diagrams
  - Automatic Layout 37
  - Format 37
  - UML 37
- Disconnect
  - From TFS 46

## - E -

- Edit
  - Attribute Code In Visual Studio .NET 20
  - Attribute Properties in Enterprise Architect 20
  - Class Attributes 30
  - Class Code In Visual Studio .NET 18
  - Class in Enterprise Architect 28
  - Class Operations 30
  - Class Properties in Enterprise Architect 18
  - Class, Switch To Visual Studio .NET 29
  - Code, In Visual Studio .NET 34
  - Operation Code In Visual Studio .NET 19
  - Operation Properties in Enterprise Architect 19
- Engineering
  - Forward 40
  - Reverse 42
  - Round Trip 38
- Execute Project
  - Within Enterprise Architect 22

## - F -

- Forward Engineering 40
  - From Enterprise Architect 40
  - Via MDG Link Merge 41

## - G -

- Generate
  - Code From Class Element, In Enterprise Architect 40
  - Code From Class Element, Via MDG Link Merge 41

## - I -

- Import
  - TFS Work Items 45
- Inheritance
  - Add To Class 31
- Initial Merge 17

## - L -

- License 5
- Link Package
  - To Visual Studio .NET Project 15

## - M -

- Manage
  - Mapped Work Item Fields 48
- Mapped Fields
  - Manage 48
- MDG Link For Visual Studio .NET
  - Activate 12
  - Copyright 4
  - Features 2
  - License 5
  - License Keys 12
  - Link Package To Visual Studio .NET Project 15
  - Order 11
  - Register 12
  - Set Up 14
  - Support 9
  - System Requirements 10
  - Trademarks 8
  - User Forum 9
  - Welcome 2

## Merge

- Code And Model 38, 44
- Code Into Model 43
- Code To Create Model In Enterprise Architect 17
- Dialog 38
- Facilities 38
- Initial 17
- Model Into Code 41
- Model To Generate Code In Visual Studio .NET 17
- Options 38

## Model

- Generation 42
- Synchronize With Code 44

## - N -

- Navigation
  - Of Project 18

## - O -

- ocate
  - Element In Project Browser 18
- Operation
  - Edit 30
  - Edit Code In Visual Studio .NET 19
  - Edit Properties in Enterprise Architect 19
- Options
  - Merge 38
  - Project Browser, Add In Options 18
- Order
  - MDG Link For Visual Studio .NET 11

## - P -

- Project
  - Build Dialog Options 23
  - Build Errors 23
  - Build Within Enterprise Architect 22
  - Execute Within Enterprise Architect 22

## - R -

- Register
  - MDG Link For Visual Studio .NET 12
- Related Elements
  - Add To Class 32
- Reverse Engineering
  - Introduction 42

Reverse Engineering	
Model From Code	43
Single Class	43
Round Trip Engineering	38
Synchronize Model With Code	44

## - S -

Set Up	
MDG Link For Visual Studio .NET	14
Software Product License Agreement	5
Support	9
Synchronize	
Class With Code	43
Code And Model	44
Code With Model	40
Forward	40
Model From Code	43
Model With Code	42
Reverse	42
TFS Work Items	45
System Requirements	10

## - T -

Team Foundation Server	
Configure Connection	46
Connect To	46
Disconnect From	46
Import Work Items	45
Maintain Work Items	46
Menu Option	18
Synchronize Work Items	45
TFS	
Configure Connection	46
Connect To	46
Disconnect From	46
Import Work Items	45
Maintain Work Items	46
Menu Option	18
Synchronize Work Items	45
Trademarks	8

## - U -

UML Diagrams	37
--------------	----

## - V -

Visual Studio .NET Project	
Link Package To	15

## - W -

Welcome	2
Work Item	
Manage Mapped Fields	48
Map Fields To Enterprise Architect Properties	48
Work Items	
Add New	46
Import	45
Link To Element	46
Remove From Element	46
Synchronize	45
TFS	45
TFS, Maintain	46





MDG Link for Visual Studio.NET User Guide

[www.sparxsystems.com](http://www.sparxsystems.com)