

**Detlef Drews
Heinz Schwab**

workshop Visual Basic .NET

 **ADDISON-WESLEY**

An imprint of Pearson Education

München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam

2

Workshop: Die Visual Basic .NET-Entwicklungsumgebung

Visual Basic ist eine Programmiersprache. Die Programmierung in Visual Basic .NET erfordert nicht zwingend die Verwendung einer speziellen Programmierumgebung. Es ist durchaus möglich, mit einem normalen Texteditor wie dem Notepad von Windows und dem Compiler der .Net-Umgebung Programme zu erstellen.

Empfehlenswert ist diese Vorgehensweise allerdings höchstens für sehr kleine Konsolenanwendungen, also Programme ohne Oberfläche mit beschränktem Funktionsumfang. Für alle anderen Programme bietet Microsoft mit der Visual Studio-Entwicklungsumgebung ein hochintegriertes Werkzeug, um komfortabel Programme zu erstellen.

Sie erhalten Visual Basic .NET immer als Komplettpaket, bestehend aus der Programmiersprache selbst, den notwendigen Werkzeugen, die Sie bei der Erstellung eigener Programme unterstützen, und einer Anzahl von Zusatzprogrammen, die oft benötigte Aufgaben stark vereinfachen. All dies ist bei Visual Basic in der Visual Studio-Entwicklungsumgebung zusammengefasst.

In diesem Kapitel wird die Entwicklungsumgebung in aller Kürze vorgestellt. Die wichtigsten Oberflächenelemente und Werkzeuge werden benannt und gezeigt. Zudem wird besprochen, was ein Visual Basic-Projekt ist.

2.1 Die Oberfläche von Visual Basic .NET

Wenn Sie Visual Studio .NET starten, sehen Sie das Fenster aus Abbildung 2.1. Falls Sie Visual Studio .NET zum ersten Mal gestartet haben, werden Sie feststellen, dass eine gewisse Einarbeitungszeit benötigt wird, um alle Elemente der Oberfläche kennen zu lernen und mit ihnen arbeiten zu können.

Aber denken Sie daran, diese Entwicklungsumgebung wird Ihnen die Entwicklung Ihrer Programme auf einfachste Weise erlauben, Sie müssen sie eben nur erst kennen lernen.

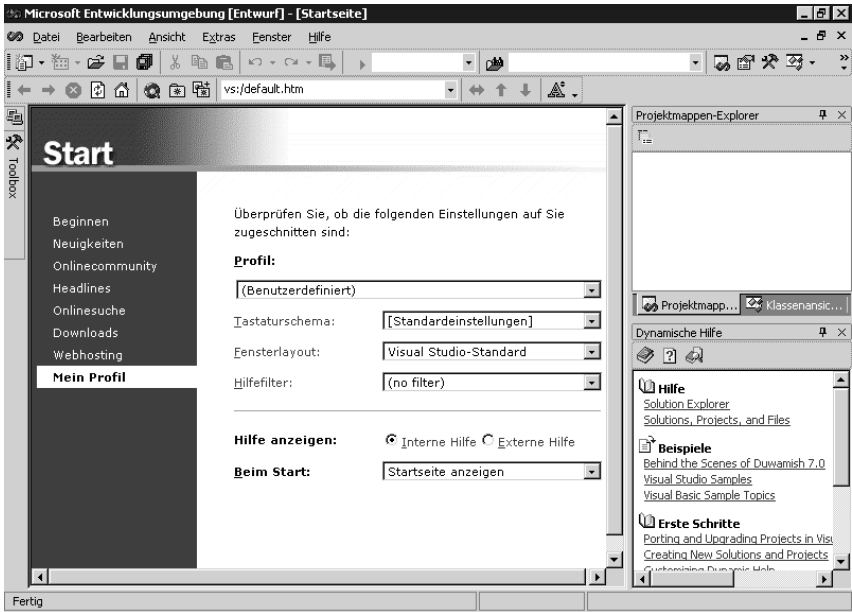
Um Ihnen einen leichten Einstieg und einen Überblick zu verschaffen, werden in den folgenden Abschnitten zunächst die wichtigsten Elemente der Oberfläche benannt und erläutert.

2.2 Die wichtigsten Menüs

Im Menü DATEI (Abbildung 2.2) finden Sie alle Funktionen, die benötigt werden, um ein Projekt oder einzelne Dateien des Projektes zu erstellen, zu laden oder zu speichern.

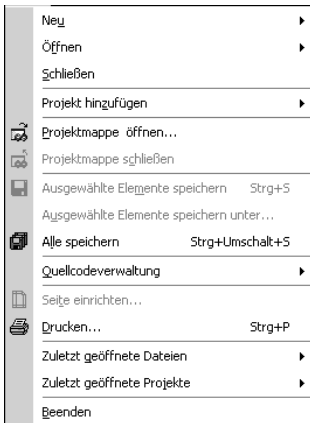
Menü Datei

Abbildung 2.1:
Die Visual Basic
.NET-Entwicklungs-
umgebung
Visual Studio



Sie können in diesem Menü Ausdrücke starten und Visual Studio beenden. Darüber hinaus wird in diesem Menü eine Liste der zuletzt geöffneten Projekte geführt, die dazu verwendet werden kann, diese Projekte zu laden, ohne dafür einen entsprechenden Dialog öffnen zu müssen.

Abbildung 2.2:
Das Menü Datei



2.2.1 Übung: Anlegen eines neuen Projekts



Legen Sie ein Standardprojekt (normale Windows-Anwendung) an und speichern Sie es.

Da dieses Projekt keinen Programmcode hat, ist es auf der beiliegenden CD nicht gespeichert.



Lösung:

Über das Menü DATEI->NEU-> PROJEKT. wird der Dialog NEUES PROJEKT (Abbildung 2.3) geöffnet.

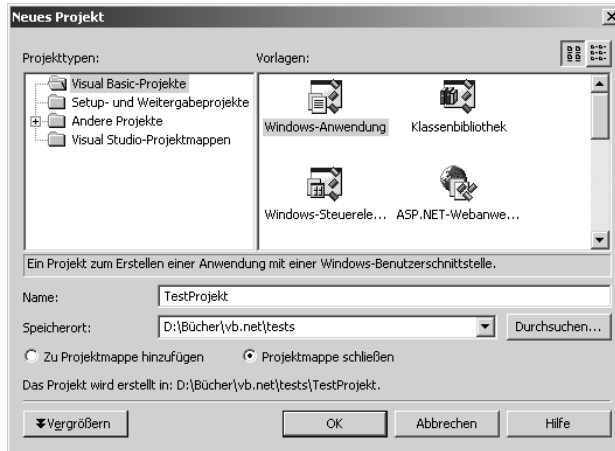


Abbildung 2.3:
Dialog Neues
Projekt

Wählen Sie den Eintrag *Windows-Anwendung* und betätigen Sie die Schaltfläche *OK*. Visual Basic erstellt jetzt automatisch ein Projekt mit allen notwendigen Projektdateien. Das Standardmodul wird automatisch geöffnet, so dass Sie sofort anfangen können mit dem Oberflächendesign.

Windows- Anwendung

Bei einer *Windows-Anwendung* wird der Projektmappe ein Formular hinzugefügt, wie Sie im Projektmappen-Explorer (Abbildung 2.4) erkennen können.

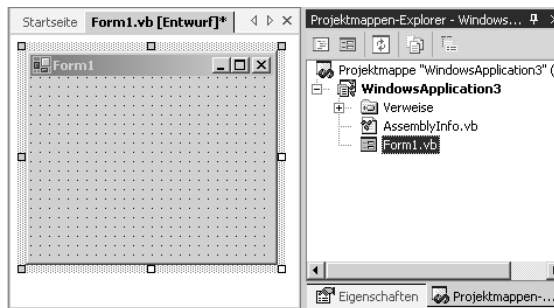


Abbildung 2.4:
Eine Standard-
Projekt wurde
angelegt

Beim Anlegen eines neuen Projektes erstellt Visual Studio sowohl eine Projektdatei mit der Endung *.vbproj* als auch eine so genannte Projektmappe mit der Dateiendung *.sln*. Sie können jedes neue Projekt in einer eigenen Projektmappe speichern oder mehrere Projekte in einer Projektmappe zusammenfassen.

Projektmappe

Projekt speichern

Um das Projekt zu speichern, ist wiederum eine Auswahl des Menüs DATEI erforderlich. Die einfachste Art ist die Auswahl des Menüpunktes ALLE SPEICHERN. Dieser sorgt dafür, dass alle dem Projekt zugehörigen Dateien gespeichert werden.

Es werden hierbei keinerlei Dialoge aufgeblendet, die eine Vergabe von Dateinamen oder Verzeichnissen erlauben würden. Es werden also für die Projektmappe selbst die Namen verwendet, die im Dialog aus Abbildung 2.3 verwendet wurden, um das Projekt anzulegen.

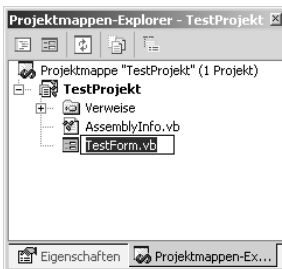


Für die Form würde der Standardname verwendet werden, der für das erste Formular eines Projekts von Visual Studio .NET mit *Form1* vorgegeben wird. Wenn Ihre Applikation nur ein Formular beinhaltet, so ist dies sicher kein Problem, wenn aber Ihre Applikation mehrere oder gar sehr viele Formulare beinhaltet, sollten Sie diese unbedingt mit sprechenden Namen speichern.

Damit Sie nicht für jedes Projekt eine Datei mit Namen *Form1* speichern, nur um diese später umzubenennen, sollten Sie vor dem ersten Aufruf des Menüpunktes ALLE SPEICHERN das Formular *Form1* umbenennen. Sie müssen hierzu lediglich im Projektmappen-Explorer den Dateinamen *Form1.vb* umbenennen, wobei die Endung *vb* nicht verändert werden sollte.

Um den Editiermodus, der in Abbildung 2.5 gezeigt wird, zu starten, benutzen Sie die rechte Maustaste über dem Eintrag *Form1.vb* im Projektmappen-Explorer. Im dadurch geöffneten Kontextmenü wählen Sie den Eintrag UMBENENNEN.

Abbildung 2.5:
Umbenennen der
Form1.vb



2.2.2 Menü Bearbeiten

Menü Bearbeiten

Das Menü BEARBEITEN (Abbildung 2.6) bietet neben den zum Standard gewordenen Punkte wie RÜCKGÄNGIG, KOPIEREN, AUSSCHNEIDEN, EINFÜGEN, SUCHEN UND ERSETZEN etc. einige neue und hilfreiche Funktionen.

Diese betreffen vor allem den Editor und dienen der Fehlerminimierung bei der Eingabe von Code. Eine nähere Beschreibung dieser Hilfsmittel finden Sie im Abschnitt über den *Editor*.

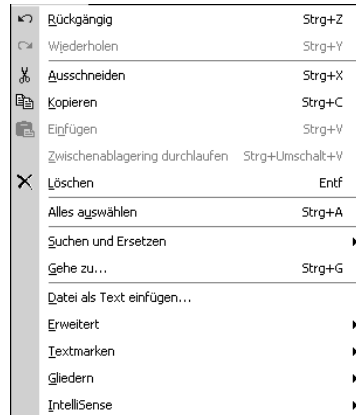


Abbildung 2.6:
Menü Bearbeiten

2.2.3 Menü Ansicht

Über die Menübefehle des Menüs ANSICHT (Abbildung 2.7) können Sie die Darstellung und Anzeige der Fenster in der Entwicklungsumgebung einstellen. Es stehen Menübefehle zum Öffnen des Codefensters oder der Objektansicht sowie zum Ein- und Ausblenden der verschiedenen Hilfsfenster zur Verfügung.

Menü Ansicht

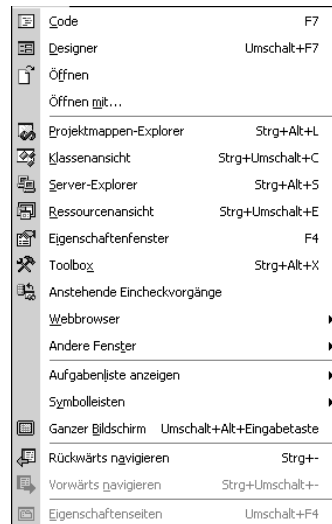


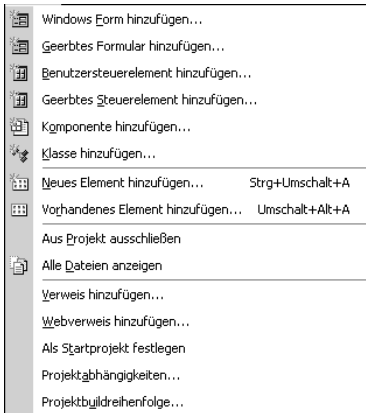
Abbildung 2.7:
Menü Ansicht

2.2.4 Menü Projekt

Über das Menü PROJEKT (Abbildung 2.8) können Sie die dem Projekt zugehörigen Dateien und Objekte verwalten. Hier können Sie neue Komponenten wie *Formulare*, *Module*, *Klassen* etc. den offenen Projekten hinzufügen oder aus ihnen entfernen.

Menü Projekt

Abbildung 2.8:
Menü Projekt



Menü-Befehl Verweise

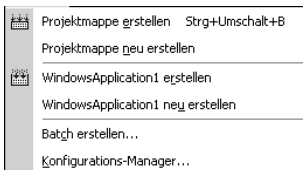
Mit dem Menübefehl VERWEISE... können Sie Bibliotheken und Anwendungen auswählen, deren Funktionen und Objekte Sie in Ihrem Projekt verwenden wollen. Es wird dann ein Verweis auf die Objektbibliothek dieser Anwendung angelegt. Nach Auswahl eines Eintrags aus dieser Liste stehen Ihnen alle Funktionen und Objekte der Bibliothek oder Anwendung in Ihrem aktuellen Projekt zur Verfügung.

2.2.5 Menü Erstellen

Menü Erstellen

Über das Menü ERSTELLEN (Abbildung 2.9) können Sie ein einzelnes Projekt oder alle Projekte einer Projektmappe erstellen, d.h. übersetzen lassen.

Abbildung 2.9:
Menü Erstellen



WÄHLEN SIE Z.B. DEN MENÜPUNKT ERSTELLEN->PROJEKTMAPPE NEU ERSTELLEN, SO WERDEN alle Projekte der aktuellen Projektmappe komplett neu übersetzt. Den Ablauf der Übersetzung können Sie dabei in einem sich zusätzlich öffnenden Ausgabefenster verfolgen (Abbildung 2.10).

2.2.6 Menü Debuggen

Menü Debuggen

Das Menü DEBUGGEN (Abbildung 2.11) dient dem Starten, Überwachen und Testen einer Anwendung. Es beinhaltet verschiedene Menüpunkte zur Prozessablaufsteuerung, die in der Regel erst nach einer Unterbrechung des Programmlaufs wirksam werden.

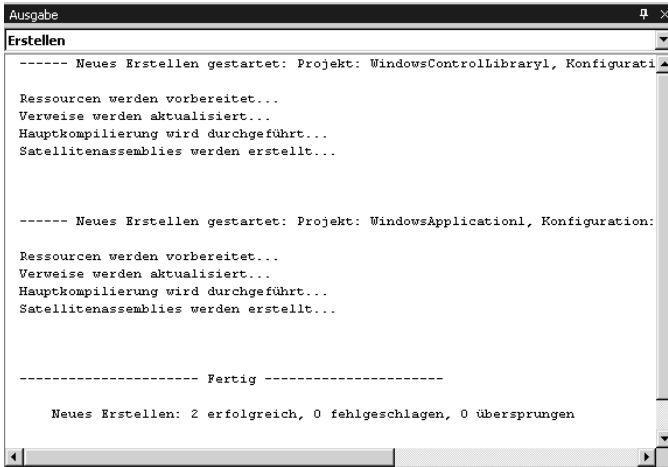


Abbildung 2.10:
Ausgaben beim Erstellen einer Projektmappe

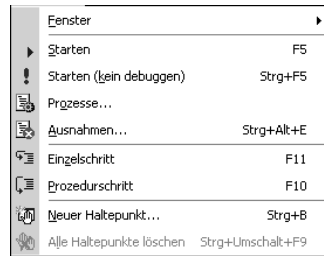


Abbildung 2.11:
Menü Debuggen

Ein wenig gewöhnungsbedürftig ist die Tatsache, dass auch der Start des Programms ohne Debug im Menü DEBUGGEN erfolgt. Sie müssen hierzu lediglich das Menü DEBUGGEN->STARTEN(KEIN DEBUGGEN) aufrufen.

2.2.7 Übung: Programm Debuggen

Laden Sie das der Aufgabe zugehörige Projekt in die Entwicklungsumgebung. Starten Sie eine Debug-Sitzung. Setzen Sie einen Haltepunkt. Arbeiten Sie im Einzelschritt-Modus und benutzen Sie die Funktionen, um Variablen zu betrachten.



Lösung:

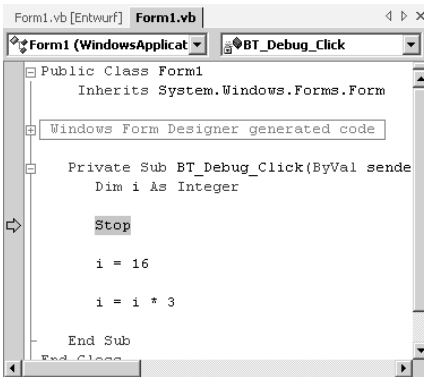
Nachdem Sie das Projekt geladen haben, können Sie es direkt starten. Benutzen Sie hierfür die Funktionstaste **F5**. Sie sehen die Oberfläche aus Abbildung 2.12.



Abbildung 2.12:
Das Programm wurde gestartet

Nachdem Sie die Schaltfläche *Starten* ausgewählt haben, öffnet sich ein Code-Fenster (Abbildung 2.13).

Abbildung 2.13:
Programmlauf
wurde durch
Stop-Anweisung
unterbrochen

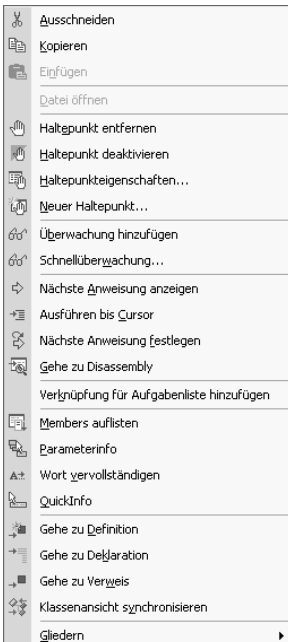


Wenn Sie ein Programm debuggen möchten, so müssen Sie den Programmlauf unterbrechen, denn nur dann haben Sie die Möglichkeit, auf den Programmlauf Einfluss zu nehmen bzw. einen aktuellen Status der Programmzustände zu erfassen.

Haltepunkt setzen

In Abbildung 2.13 wurde der Programmlauf durch eine *Stop*-Anweisung unterbrochen. Eine weitere Möglichkeit ist das Setzen eines Haltepunktes. Hierzu müssen Sie den Eingabecursor in die Zeile stellen, in welcher Sie anhalten möchten. Dann drücken Sie die Taste **[F9]**, oder Sie wählen im Kontextmenü des Editors (Abbildung 2.14) den Menüpunkt HALTEPUNKT EINFÜGEN.

Abbildung 2.14:
Das Kontextmenü
des Editors im
Debug-Modus



Sie erkennen den Haltepunkt im Code-Fenster am roten Punkt links neben der Programmzeile (Abbildung 2.15).

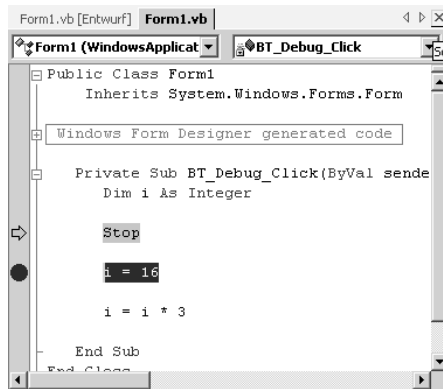


Abbildung 2.15:
Haltepunkt wurde gesetzt

Durch Betätigen der Taste **[F5]** wird der Programmablauf fortgesetzt. Der nächste Stop ist auf dem gesetzten Haltepunkt.

Über das Menü **DEBUGGEN->FENSTER** können verschiedene Hilfsfenster geöffnet werden, welche eine automatische Überwachung der lokalen Variablen oder aller Variablen des Projekts ermöglichen.

Um gezielt eine spezifische Variable zu überwachen, kann eine Schnellüberwachung durchgeführt werden, wie in Abbildung 2.16 gezeigt.

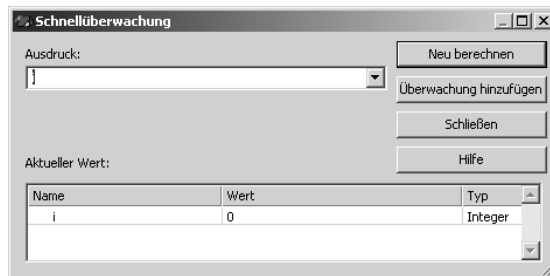
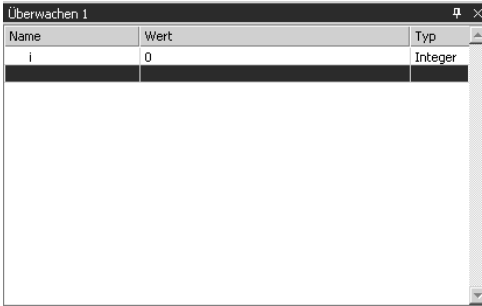


Abbildung 2.16:
Schnellüberwachung im Debugger

Die Schnellüberwachung ist geeignet, um den Inhalt einer Variablen einmalig zu überprüfen. Soll der Inhalt einer Variablen öfter überprüft werden, bietet es sich an, diese in eine dauerhafte Überwachung zu überführen. Sie können dies im Dialog **SCHNELLÜBERWACHUNG** durch Ausführen der Schaltfläche *Überwachung hinzufügen* tun oder über das Kontextmenü des Editors.

Das Überwachungsfenster in Abbildung 2.17 zeigt den Inhalt der Variablen an, wenn der Programmablauf auf dem Haltepunkt gestoppt wurde. Obwohl die Programmzeile der Variablen *i* den Wert 16 zuweist, hat die Variable im Überwachungsfenster den Wert 0.

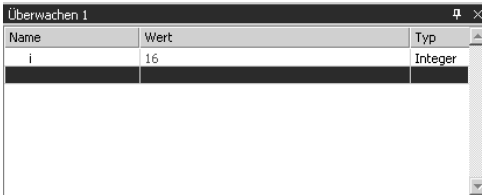
Abbildung 2.17:
Variable im Überwachungsfenster



Das Programm wird durch einen Haltepunkt gestoppt, bevor die Programmzeile, auf der sich der Haltepunkt befindet, ausgeführt wird.

Um das Programm im Einzelschrittmodus weiterzuführen, benutzen Sie die Funktionstaste **[F8]** oder den Menübefehl **DEBUGGEN->EINZELSCHRITT**. In Abbildung 2.18 können Sie sehen, dass sich der Wert der Variablen *i* im Überwachungsfenster geändert hat.

Abbildung 2.18:
Wert der Variablen nach Ausführen der Programmzeile im Einzelschrittmodus



Sie können jetzt das Programm durch mehrfaches Betätigen der Funktionstaste **[F8]** oder durch einmaliges Betätigen der Funktionstaste **[F5]** zu Ende laufen lassen.



Sie haben jetzt die Grundtechniken des Debuggens mit Visual Basic kennengelernt. Der Debugger von Visual Basic ist aber ein weitaus mächtigeres Werkzeug, als bisher gezeigt wurde. Es lohnt sich auf jeden Fall hierzu die Online-Hilfe von Visual Basic zusätzlich zu konsultieren.

2.2.8 Menü Format

Menü Format

Mit dem Menü **FORMAT** (Abbildung 2.19) erreichen Sie eine ganze Reihe von Befehlen, die es Ihnen erlauben, die Anordnung der Steuerelemente zu verändern und sie auszurichten.

Dies bezieht sich vor allem auf Gruppen von Steuerelementen. Ein einzelnes Steuerelement lässt sich mit den Menübefehlen lediglich am Raster ausrichten.

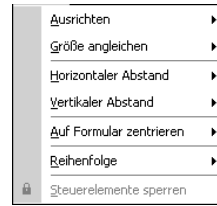


Abbildung 2.19:
Menü Format

Beachten Sie bitte, dass dieses Menü nur dann in der Menüleiste erscheint, wenn ein Oberflächendesigner aktiv ist. Befinden Sie sich momentan im Editiermodus eines Moduls, so ist das Menü FORMAT nicht vorhanden.



2.2.9 Menü Extras

Das Menü EXTRAS beinhaltet unter anderem den wichtigen Menüpunkt OPTIONEN. Dieser Menübefehl öffnet den Dialog OPTIONEN, über welchen globale Einstellungen der Entwicklungsumgebung vorgenommen werden können.

Menü Extras

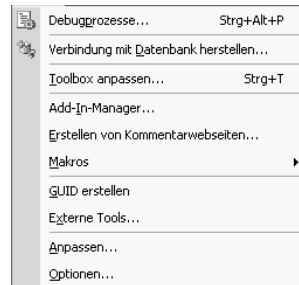


Abbildung 2.20:
Menü Extras

Des Weiteren können in diesem Menü durch Aufruf des Menüpunktes ANPASSEN die Symbolleisten, die Menüs und die zugehörigen Optionen konfiguriert werden.

Durch den *Add-In* Manager können Werkzeuge von Fremdherstellern in die Entwicklungsumgebung integriert werden, und über den Menüpunkt MAKROS kann jetzt auch in der Visual Studio-Entwicklungsumgebung mit Makros gearbeitet werden.

2.3 Symbolleisten

Die Symbolleisten stellen ein *Mittelding* zwischen Menübefehlen und Tastaturkürzeln dar. Sie ermöglichen Ihnen, einen Menübefehl mit einem Mausklick auf ein Symbol auszuführen. Diese Symbole sind, nach Themenbereichen geordnet, in so genannten Leisten untergebracht und befinden sich unterhalb der Menüleiste.

Symbolleisten

Symbolleisten als eigenständige Fenster

Jede Symbolleiste ist einzeln verschiebbar. Man bedient sich dabei eines in der linken Seite einer jeden Symbolleiste angebrachten senkrechten Balkens, der wiederum aus mehreren kleinen waagerechten Balken besteht. Damit kann man die Leisten mit der Maus umpositionieren. Wenn Sie eine Leiste außerhalb des Bereichs ziehen, in der sie liegen kann, wird sie zu einem eigenständigen Fenster. So kann man die Leisten auch als Fenster beliebig über den Bildschirm verteilen.

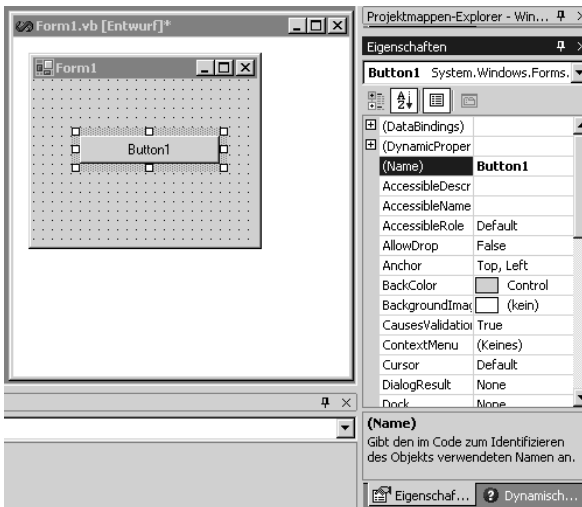
Menü Extras->Anpassen

Sie haben auch die Möglichkeit, den Inhalt der Symbolleisten anzupassen oder gar eigene Symbolleisten zu entwerfen. Hierzu dient der Menübefehl EXTRAS->ANPASSEN...

2.4 Das Eigenschaftsfenster

Im *Eigenschaftsfenster* (Abbildung 2.21) werden während der Programmentwicklung die Eigenschaften von Objekten (Formulare, Klassen, Steuerelemente) angezeigt und können dort auch verändert werden. Das *Eigenschaftsfenster* zeigt immer die Eigenschaften des aktuell markierten Objekts an.

Abbildung 2.21:
Die Eigenschaften
der Schaltfläche
Button1 im Eigen-
schaftsfenster



Falls das Eigenschaftsfenster nicht sichtbar ist, kann es über das Menü ANSICHT->EIGENSCHAFTENFENSTER, über die Taste **F4** oder über die Symbolleiste aktiviert werden.

Das Eigenschaftsfenster besteht aus den Teilen *Objekt-Listbox* und der *Eigenschaften-Liste*.

2.4.1 Objekt-Listbox

In der *Objekt-Listbox*, die sich direkt unterhalb des Fenstertitels befindet, sind alle Objekte des aktuell aktiven Moduls eingetragen. Falls nur ein Objekt markiert ist, wird der Name des markierten Objekts in der Textbox angezeigt. Mit der DropDown-Liste kann, ohne das Eigenschaftenfenster zu verlassen, auf jedes andere Objekt des aktiven Moduls gewechselt werden.

Alle Objekte des aktiven Moduls

2.4.2 Eigenschaftenliste

Die *Eigenschaftenliste* ist eine Tabelle mit zwei Spalten. Die linke Spalte enthält die Eigenschaften des markierten Objekts. Die rechte Spalte enthält die aktuellen Werte dieser Eigenschaften.

Eigenschaften und Werte

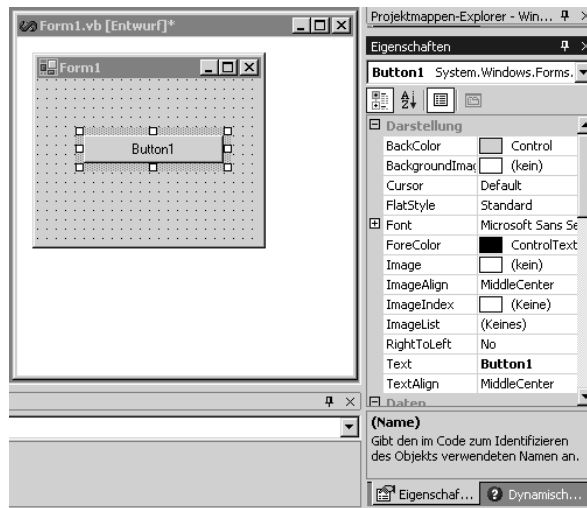


Abbildung 2.22: Eigenschaften nach Kategorien

Über die Registerkarten der Liste können die Eigenschaften alphabetisch oder, wie in Abbildung 2.22 sichtbar, nach Kategorien geordnet angezeigt werden.

Das Setzen von Eigenschaften ist in Visual Basic der halbe Weg zum fertigen Programm. Unterschätzen Sie nicht die Macht der Eigenschaften. Viele Zeilen Code meiner Programme zeigten sich im Nachhinein als überflüssig, nachdem ich die Eigenschaften der Objekte besser definierte.



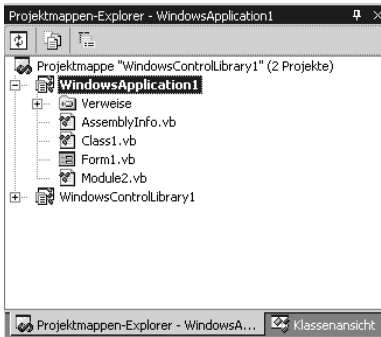
2.5 Der Projektmappen-Explorer

Der *Projektmappen-Explorer* zeigt eine Liste aller im Projekt bzw. der Projektmappe verwendeten Module. Abbildung 2.23 zeigt dabei nur eine kleine Auswahl der möglichen Projektdateien. Ob normale *Formulare* oder *Module*,

Alle Dateien des/der Projekts/Projekte

Klassen oder *Steuerelementdateien*, im Projektmappen-Explorer sind alle Projektdateien aufgelistet.

Abbildung 2.23:
Der Projektmappen-Explorer



Vom *Projektmappen-Explorer* aus kann das *Formularfenster* oder das *Code-Fenster* eines Moduls geöffnet werden. Es muss lediglich das entsprechende Modul in der Liste markiert werden. Jetzt wird entweder das Symbol *Code anzeigen* angewendet, um das zugehörige *Code-Fenster* zu öffnen, oder es wird das Symbol *Designer anzeigen* angewendet, um das zugehörige *Formularfenster* zu öffnen.

Einfach anklicken

Wird ein Modul im Projektfenster doppelt angeklickt, so wird für Formulare automatisch das *Formularfenster*, für Module automatisch das *Code-Fenster* geöffnet.

2.5.1 Kontextmenüs des Projektmappen-Explorers

Kontextmenüs

Wird über dem *Projektmappen-Explorer* die rechte Maustaste gedrückt, so öffnet sich ein Kontextmenü (Abbildung 2.24).

Alle in diesem Menü zu Verfügung stehenden Menüpunkte beziehen sich auf das markierte Projekt. Sie können hier z.B. das Projekt erstellen oder dem Projekt neue Module hinzufügen.

Das Startprojekt

Befinden sich in einer Projektmappe mehrere Projekte, so wird eines dieser Projekte im Projektmappen-Explorer fett dargestellt. Dieses Projekt wird als Startprojekt bezeichnet. Das Startprojekt wird durch Betätigen der Taste **F5** gestartet.

Als Startprojekt festlegen

In diesem Kontextmenü kann durch Aufruf des Menüpunkts **ALS STARTPROJEKT FESTLEGEN** das Startprojekt einer Projektmappe geändert werden. Sie müssen hierzu das Kontextmenü über dem Projekt öffnen, welches als Startprojekt festgelegt werden soll.

Falls Sie beim Betätigen der rechten Maustaste nicht das Menü aus Abbildung 2.24 sehen, sondern ein Kontextmenü wie in Abbildung 2.25, so befanden Sie sich wahrscheinlich mit dem Mauszeiger nicht über einem *Projekt*-Eintrag.



Abbildung 2.24:
Das Projekt-
Kontextmenü des
Projektmappen-
Explorers

In diesem Fall zeigt der Projektmappen-Explorer das Kontextmenü des jeweiligen Moduls. Das Kontextmenü eines Formulars sehen Sie in Abbildung 2.25.

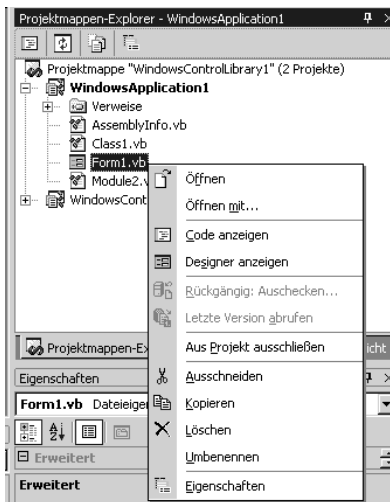


Abbildung 2.25:
Das Kontextmenü
eines Formulars

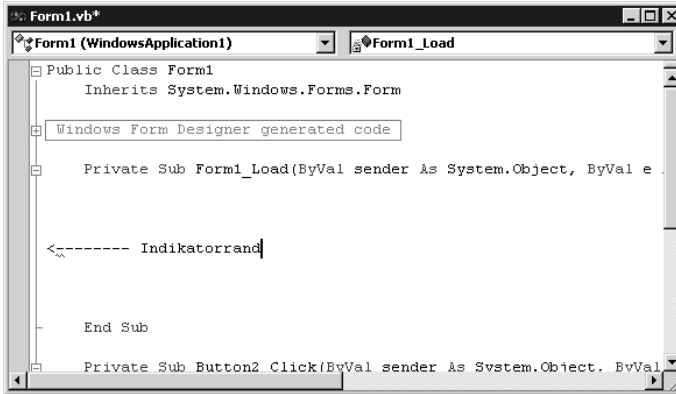
Alle Menüpunkte eines solchen Kontextmenüs beziehen sich ausschließlich auf das entsprechende Modul, über dem Sie die rechte Maustaste betätigen haben.

2.6 Das Code-Fenster

Code-Fenster = Editor

Das *Code-Fenster*, auch Editor genannt, präsentiert sich mit einigen nützlichen Eigenschaften. Zuerst wird Ihnen wohl der senkrechte Strich am linken Rand auffallen. Dies ist der so genannte Indikatorrand (Abbildung 2.26).

Abbildung 2.26:
Das Code-Fenster
mit Indikatorrand



Indikatorrand

Auf dem Indikatorrand können Lesezeichen und Haltepunkte angezeigt und gesetzt werden. Im Abschnitt über das Debuggen konnten wir dies bereits sehen.

Das Code-Fenster hat aber einige weitere Funktionen, die das Programmieren sehr erleichtern:

2.6.1 Automatische Anweisungsergänzung

**Anweisungs-
ergänzung**

Wenn Sie im Code auf ein Objekt mit einer Methode oder Eigenschaft Bezug nehmen, so wird Ihnen nach Eingabe des Punktes eine Drop-Down-Liste angezeigt, die alle Methoden und Eigenschaften des Objekts enthält (Abbildung 2.27).

Mit den Pfeiltasten oder mit der Maus wählen Sie Ihr gewünschtes Element. Wenn Sie dann die Leertaste oder die Tabulatortaste drücken, wird das gewählte Element automatisch rechts hinter dem Punkt eingetragen.

**Keine
Schreibfehler**

Dies erleichtert Ihnen die Suche nach den verfügbaren Elementen des Objekts und verhindert darüber hinaus Schreibfehler. Die *Automatische Anweisungsergänzung* kann im Dialog OPTIONEN unter TEXT-EDITOR->BASIC->ALLGEMEIN->MEMBERS AUTOMATISCH AUFLISTEN aktiviert werden.

2.6.2 Parameterinformationen

**Parameter-
informationen**

Dies ist ein weiteres Hilfsmittel, um Fehleingaben zu verhindern. Wenn im Dialog OPTIONEN in TEXT-EDITOR->BASIC->ALLGEMEIN->PARAMETERINFORMATIONEN aktiviert wurde, werden Sie beim Eingeben eines Funktionsaufrufs automatisch von den zugehörigen Parametern in Kenntnis gesetzt (Abbildung 2.28).

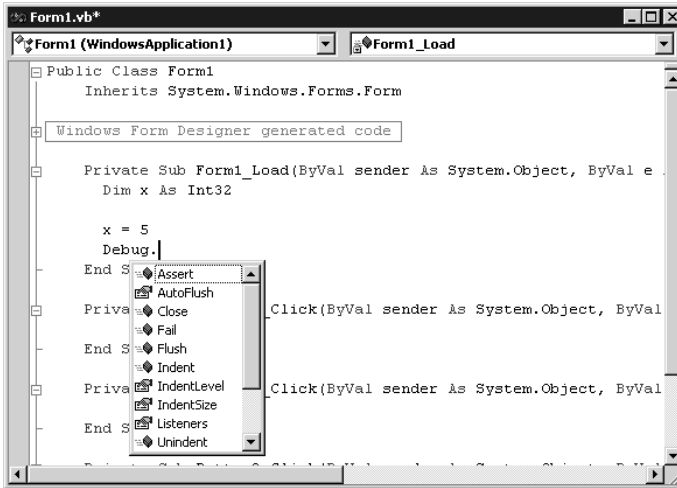


Abbildung 2.27:
Automatische
Anweisungs-
ergänzung im
Code-Fenster

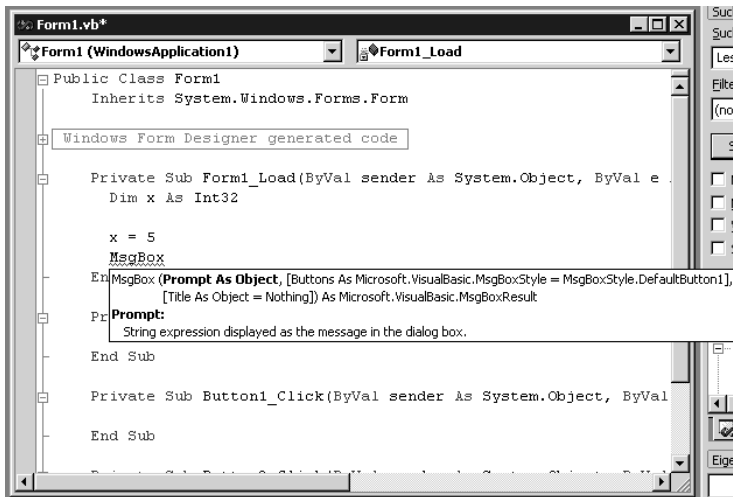


Abbildung 2.28:
Automatische
Parameterinforma-
tionen zeigt die
Parameter der
Methode MsgBox

Es wird ein Infowindow unter Ihrer aktuellen Zeile aufgeblendet, welches die Funktionssyntax anzeigt. Das erste Argument wird fett dargestellt. Wenn zu diesem ein Wert eingegeben wurde, wechselt die Markierung auf das zweite Argument und so weiter. Es kann Ihnen somit nicht mehr passieren, dass Sie zu wenig Argumente oder einen falschen Typ an eine Funktion übergeben.

All diese Hilfsmittel erreichen Sie auch über das Kontextmenü des Code-Fensters. Hiermit lässt sich die Bearbeitung über Menübefehle abkürzen.

2.7 Die Werkzeugsammlung und das Formularfenster

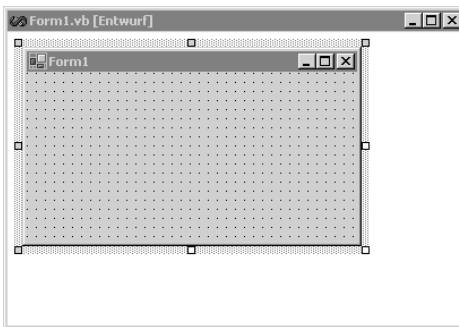
Zwei weitere, nicht wegzudenkende Elemente der Visual Basic-Entwicklungs-oberfläche sind die *Werkzeugsammlung*, auch *Toolbox* genannt, und das *Formularfenster*.

Oberflächen- design

Mit diesen zwei Werkzeugen wird im Wesentlichen die Oberfläche eines Programms erstellt. Ich habe dabei bewusst das Wort »programmiert« nicht verwendet, denn die Erstellung einer Oberfläche ist in Visual Basic nichts anderes als das Zusammenstellen von mehreren Objekten im Formularfenster.

Ein Formularfenster kann für jedes Programm-Objekt mit grafischer Oberfläche (beispielsweise eine Form) geöffnet werden (Abbildung 2.29).

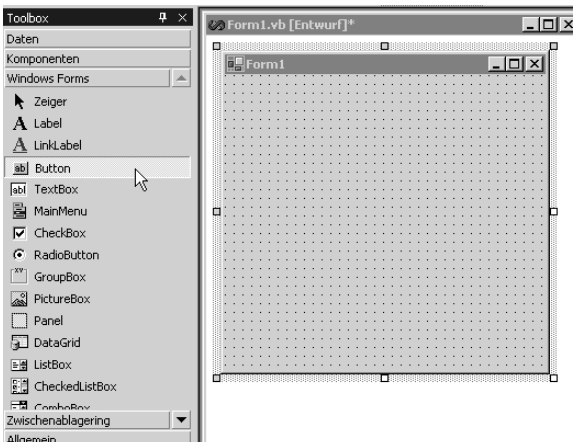
Abbildung 2.29:
Das Objekt Form1
im Objektfenster



2.7.1 Die Werkzeugsammlung – Objekte hinzufügen

Die Werkzeugsammlung ist das einzige Instrument, mit dem man neue Objekte in ein Formularfenster aufnimmt. Dabei können Sie Objekte anhand ihrer grafischen Darstellung und ihres Namens in der Werkzeugsammlung identifizieren (Abbildung 2.30).

Abbildung 2.30:
Die Toolbox



Um eine Schaltfläche auf der Form zu platzieren, wird diese, wie in Abbildung 2.30 angedeutet, in der Werkzeugsammlung doppelt angeklickt. Daraufhin wird das neue Objekt, in diesem Fall eine Schaltfläche, auf dem Formular (Objekt FORM1) platziert.

Um ein Objekt im Formularfenster an eine andere Position zu bewegen, wird es mit der Maus bei gedrückter Taste gezogen. Die Größe der Objekte kann ebenfalls per Drag and Drop geändert werden. Alle weiteren Eigenschaften werden über das Eigenschaftenfenster verändert.