



Microsoft

Visual Basic .net

Entdecken Sie die Welt der Entwicklung

Guided Tour



© 2002 Microsoft Corporation. Alle Rechte vorbehalten.

Bei diesem Dokument handelt es sich um ein vorläufiges Dokument, das bis zur endgültigen Handelsausgabe der hierin beschriebenen Software wesentlichen Änderungen unterliegen kann.

Die in diesem Dokument enthaltenen Informationen stellen die behandelten Themen aus der Sicht der Microsoft Corporation zum Zeitpunkt der Veröffentlichung dar. Da Microsoft auf sich ändernde Marktanforderungen reagieren muss, stellt dies keine Verpflichtung seitens Microsoft dar, und Microsoft kann die Richtigkeit der hier dargelegten Informationen nach dem Zeitpunkt der Veröffentlichung nicht garantieren.

Dieses Dokument dient nur zu Informationszwecken. MICROSOFT SCHLIESST FÜR DIESES DOKUMENT JEDE GEWÄHRLEISTUNG AUS, SEI SIE AUSDRÜCKLICH ODER KONKLUDENT.

Die Benutzer/innen sind verpflichtet, sich an alle anwendbaren Urheberrechtsgesetze zu halten. Unabhängig von der Anwendbarkeit der entsprechenden Urheberrechtsgesetze darf ohne ausdrückliche schriftliche Erlaubnis der Microsoft Corporation kein Teil dieses Dokuments für irgendwelche Zwecke vervielfältigt oder in einem Datenempfangssystem gespeichert, darin eingelesen oder übertragen werden, unabhängig davon, auf welche Art und Weise oder mit welchen Mitteln (elektronisch, mechanisch, durch Fotokopieren, Aufzeichnen usw.) dies geschieht.

Microsoft kann Inhaber von Patenten oder Patentanträgen, Marken, Urheberrechten oder anderen gewerblichen Schutzrechten sein, die den Inhalt dieses Dokuments betreffen. Die Bereitstellung dieses Dokuments erteilt keinerlei Lizenzrechte an diesen Patenten, Marken, Urheberrechten oder anderem geistigen Eigentum, ausgenommen, dies wurde explizit durch einen schriftlich festgehaltenen Lizenzvertrag mit der Microsoft Corporation vereinbart.

Microsoft, Visual Studio, das Visual Studio-Logo, Windows, IntelliSense, Visual Basic, Visual C++, Visual C#, Visual J#, FrontPage, MSDN, ActiveX, Active Accessibility, Windows NT, JScript, Visio, Visual SourceSafe, BizTalk, Visual InterDev, Visual FoxPro, Win32, das Windows-Logo und das Internet Explorer-Logo sind entweder eingetragene Marken oder Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Weitere in diesem Dokument aufgeführte Produkt- und Firmennamen können geschützte Marken ihrer jeweiligen Inhaber sein.

Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA

ENDBENUTZER-LIZENZVERTRAG FÜR MICROSOFT-SOFTWARE

MICROSOFT VISUAL BASIC. NET - FÜHRUNG

WICHTIG - BITTE SORGFÄLTIG LESEN: Dieser Endbenutzer-Lizenzvertrag ("EULA") ist ein rechtsgültiger Vertrag zwischen Ihnen (entweder als natürlicher oder als juristische Person) und Microsoft Corporation für das oben bezeichnete Microsoft-Softwareprodukt, das Computersoftware sowie möglicherweise dazugehörige Medien, gedruckte Materialien, Dokumentation im "Online"- oder elektronischen Format und internetbasierte Dienste umfasst ("Produkt"). Diesem Produkt liegt möglicherweise eine Ergänzungsvereinbarung oder ein Nachtrag zu diesem EULA bei. INDEM SIE DAS PRODUKT INSTALLIEREN, KOPIEREN ODER ANDERWEITIG VERWENDEN, ERKLÄREN SIE SICH DAMIT EINVERSTANDEN, DURCH DIE BESTIMMUNGEN DIESES EULAS GEBUNDEN ZU SEIN. FALLS SIE SICH DAMIT NICHT EINVERSTANDEN ERKLÄREN, SIND SIE NICHT BERECHTIGT, DAS PRODUKT ZU INSTALLIEREN ODER ZU VERWENDEN.

SOFTWAREPRODUKT-LIZENZ

1. LIZENZGEWÄHRUNGEN. Microsoft gewährt Ihnen unter der Voraussetzung, dass Sie alle Bestimmungen dieses EULAs einhalten, die in diesem EULA beschriebenen Rechte.
 - 1.1. *Allgemeine Lizenzgewährung.* Microsoft gewährt Ihnen als natürlicher Person eine persönliche, nicht ausschließliche Lizenz, Kopien des Produkts zum Entwurf, zur Entwicklung und zum Testen Ihrer Softwareprodukte anzufertigen und zu verwenden, vorausgesetzt, Sie sind die einzige natürliche Person, die das Produkt verwendet.
Wenn Sie eine juristische Person sind, gewährt Ihnen Microsoft eine persönliche, nicht ausschließliche Lizenz, Kopien des Produkts anzufertigen und zu verwenden, vorausgesetzt, dass Sie für jede natürliche Person, die das Produkt innerhalb Ihrer Organisation verwendet, eine gesonderte und gültige Lizenz für jede dieser natürlichen Personen erworben haben.
 - 1.2. *Dokumentation.* Sie sind berechtigt, eine unbegrenzte Anzahl von Kopien der Dokumentation anzufertigen und zu verwenden, vorausgesetzt, diese Kopien werden nur für persönliche Zwecke verwendet und sollen nicht (in gedruckter oder elektronischer Form) außerhalb Ihrer Geschäftsräume neu veröffentlicht oder verteilt werden.
 - 1.3. *Komponenten.* Komponenten, die Sie als Teil des Produkts erhalten, enthalten möglicherweise einen gesonderten Endbenutzer-Lizenzvertrag (jeweils als "Komponenten-EULA" bezeichnet); bei Unstimmigkeiten zwischen diesem EULA und einem Komponenten-EULA haben die Bestimmungen dieses EULAs Vorrang.
2. BESCHREIBUNG ANDERER RECHTE UND EINSCHRÄNKUNGEN
 - 2.1. Nicht zum Weiterverkauf bestimmte Software. Falls das Produkt als "Nicht zum Weiterverkauf bestimmt" oder "NFR" (Not For Resale) gekennzeichnet ist, sind Sie nicht berechtigt, das Produkt weiterzuverkaufen oder auf andere Weise für einen Gegenwert zu übertragen.
 - 2.2. Einschränkungen im Hinblick auf Zurückentwicklung (Reverse Engineering), Dekompilierung und Disassemblierung. Sie sind nicht berechtigt, das Produkt zurückzuentwickeln, zu dekompileieren oder zu disassemblieren, es sei denn, dass (und nur insoweit) das anwendbare Recht dies ungeachtet dieser Einschränkung ausdrücklich gestattet.
 - 2.3. Vermietung. Sie sind nicht berechtigt, das Produkt zu vermieten, zu verleasen oder zu verleihen.
 - 2.4. Marken. Dieses EULA gewährt Ihnen keinerlei Rechte in Verbindung mit Marken oder Dienstleistungsmarken von Microsoft.
 - 2.5. Zustimmung zur Nutzung von Daten. Sie stimmen zu, dass Microsoft und deren verbundene Unternehmen berechtigt sind, die technischen Daten, die im Rahmen der Ihnen bereitgestellten Produktsupportleistungen (sofern vorhanden) in Bezug auf das Produkt gewonnen werden, zu sammeln und zu benutzen. Microsoft ist nur berechtigt, solche Daten zur Verbesserung ihrer Produkte oder zur Bereitstellung von benutzerdefinierten Dienstleistungen oder Technologien an Sie zu verwenden, und verpflichtet sich, diese Informationen ausschließlich anonym offen zu legen.
 - 2.6. Softwareübertragung. Der erste Benutzer des Produkts darf dieses EULA und Produkt einmalig dauerhaft nur direkt an einen Endbenutzer übertragen. Diese Übertragung muss das ganze Produkt umfassen (einschließlich aller Komponenten, der Medien und gedruckten Materialien, aller Updates, dieses EULAs und gegebenenfalls des Certificate of Authenticity (Echtheitszertifikat)). Eine solche Übertragung darf nicht als Kommission oder irgendeine andere indirekte Übertragung erfolgen. Der Empfänger einer solchen einmaligen Übertragung muss sich mit den Bestimmungen dieses EULAs einverstanden erklären, einschließlich der Verpflichtung, dieses EULA und Produkt nicht weiter zu übertragen.
 - 2.7. Trennung von Komponenten. Das Produkt wird als einheitliches Produkt lizenziert. Sie sind nicht berechtigt, seine Komponenten für die Verwendung durch mehr als einen Benutzer voneinander zu trennen.
 - 2.8. Kündigung. Unbeschadet sonstiger Rechte ist Microsoft berechtigt, dieses EULA zu kündigen, sofern Sie gegen die Bestimmungen dieses EULAs verstoßen. In einem solchen Fall sind Sie verpflichtet, sämtliche Kopien des Produkts und alle seine Komponenten zu vernichten.
3. AUSSCHLUSS DER GEWÄHRLEISTUNG. Im größtmöglichen durch das anwendbare Recht gestatteten Umfang stellen Microsoft und deren Lieferanten das Produkt und gegebenenfalls Supportleistungen wie besprochen und ohne Garantie auf Fehlerfreiheit zur Verfügung. Sie schließen hiermit alle anderen Gewährleistungen und Garantien aus, seien sie ausdrücklich, konkludent oder gesetzlich, einschließlich, aber nicht beschränkt auf (sofern vorhanden) jede konkludente Gewährleistung, Garantie oder Pflicht der Handelsüblichkeit, Eignung für einen bestimmten Zweck, Zuverlässigkeit oder Verfügbarkeit, Genauigkeit oder Vollständigkeit von Antworten, Ergebnisse, fachmännische Bemühungen, Virenfreiheit und Sorgfalt - alles in Bezug auf das Produkt - sowie durch das Produkt oder anderweitig aus der Verwendung des Produkts entstehende bereitgestellte oder nicht erbrachte Support- oder andere Leistungen, Informationen, Software und dazugehörigen Inhalt. Es wird auch jede Gewährleistung oder Garantie für Eigentum, ungestörte Nutzung, ungestörten Besitz, Übereinstimmung mit der Beschreibung und Nichtverletzung von Rechten Dritter in Bezug auf das Produkt ausgeschlossen.
Wenn Sie das Produkt in DEUTSCHLAND erworben haben, findet der vorstehende Ausschluss der Gewährleistung auf Sie keine Anwendung. Hinsichtlich etwaiger gesetzlicher Gewährleistungsansprüche wenden Sie sich bitte an Ihren Händler.
4. AUSSCHLUSS VON FOLGE-, ZUFÄLLIGEN UND BESTIMMTEN ANDEREN SCHÄDEN. Im größtmöglichen durch das anwendbare Recht gestatteten Umfang sind Microsoft oder deren Lieferanten in keinem Fall haftbar für irgendwelche speziellen, Folge-, zufälligen, strafrechtlichen, indirekten oder anderen Schäden welcher Art auch immer (einschließlich, aber nicht beschränkt auf Schäden aus entgangenem Gewinn, Verlust von geschäftlichen, vertraulichen oder sonstigen Informationen, Geschäftsunterbrechung, Personenschäden, Verlust von Privatsphäre, Verletzung von Vertragspflichten (einschließlich Pflichten nach Treu und Glauben oder Sorgfaltspflichten), Fahrlässigkeit und andere finanzielle oder sonstige Verluste), die aus der Verwendung des Produkts, aus bereitgestellten oder nicht erbrachten Support- oder anderen Leistungen, Informationen, Software und dazugehörigem Inhalt durch das Produkt oder anderweitig aus der Verwendung des Produkts oder anderweitig aus oder in Verbindung mit einer Bestimmung dieses EULAs resultieren oder in irgendeinem Zusammenhang damit stehen, selbst im Falle von Verschulden, unerlaubten Handlungen (einschließlich Fahrlässigkeit), verschuldensunabhängiger Haftung, Vertragsbruch oder Verletzung der Garantie von/durch Microsoft oder deren Lieferanten, und selbst wenn Microsoft oder deren Lieferanten auf die Möglichkeit solcher Schäden hingewiesen worden sind.

5. **BESCHRÄNKUNG DER HAFTUNG UND VON ANSPRÜCHEN.** Ungeachtet aller Schäden, die Sie aus welchen Gründen auch immer erleiden mögen (einschließlich, aber nicht beschränkt auf alle oben angesprochenen Schäden sowie alle direkten oder allgemeinen Schäden) ist die gesamte Haftung von Microsoft und deren Lieferanten unter allen Bestimmungen dieses EULAs und Ihr ausschließlicher Anspruch für alles Vorstehende beschränkt auf: den tatsächlich von Ihnen für das Produkt gezahlten Betrag oder US-\$ 5,00, je nachdem welcher Betrag höher ist. Die vorstehenden Beschränkungen und Ausschlüsse (einschließlich der vorstehenden Abschnitte 3 und 4) gelten im größtmöglichen durch das anwendbare Recht gestatteten Umfang, auch wenn ein Anspruch seinen wesentlichen Zweck verfehlt.

Wenn Sie das Produkt in DEUTSCHLAND erworben haben, finden die Beschränkungen in den vorstehenden Absätzen 4 und 5 auf Sie keine Anwendung. Stattdessen gilt Folgendes: Microsoft haftet in Fällen von Vorsatz, grober Fahrlässigkeit, bei Verletzung von Leben, Körper oder der Gesundheit sowie bei Ansprüchen nach dem Produkthaftungsgesetz nach den gesetzlichen Vorschriften. Eine weitergehende Haftung von Microsoft besteht nicht. Dies gilt für alle Haftungstatbestände einschließlich unerlaubter Handlung.

6. **LINKS ZU SITES VON DRITTANBIETERN.** Möglicherweise können Sie durch die Verwendung des Produkts zu verknüpften Sites von Drittanbietern gelangen. Die Sites von Drittanbietern stehen nicht unter der Kontrolle von Microsoft, und Microsoft ist nicht für den Inhalt der Sites von Drittanbietern, für irgendwelche in den Sites von Drittanbietern enthaltenen Links oder für Änderungen oder Updates der Sites von Drittanbietern verantwortlich. Microsoft ist nicht für Webcasting oder eine andere Form der Übertragung verantwortlich, die Sie von Sites von Drittanbietern empfangen. Microsoft stellt Ihnen diese Links zu Sites von Drittanbietern nur gefälligkeitshalber zur Verfügung. Aus der Aufnahme eines Links kann keine Billigung der jeweiligen Site eines Drittanbieters durch Microsoft abgeleitet werden.
7. **AUSFUHRBESCHRÄNKUNGEN.** Sie erkennen an, dass das Produkt aus den USA stammt und dem Exportrecht der USA unterliegt. Sie erklären sich damit einverstanden, alle anwendbaren internationalen und nationalen Gesetze einzuhalten, die für das Produkt gelten, einschließlich der Regelungen der USA zur Exportkontrolle (US Export Administration Regulations) sowie Beschränkungen im Hinblick auf Endbenutzer, Endbenutzung und Bestimmungsort, die von der Regierung der USA und anderen Regierungen erlassen wurden. Weitere Informationen finden Sie unter <http://www.microsoft.com/exporting/>.
8. **ANWENDBARES RECHT.** Wenn Sie dieses Produkt in den USA erworben haben, unterliegt dieses EULA den Gesetzen des Staates Washington. Wenn Sie dieses Produkt außerhalb der USA erworben haben, gilt möglicherweise das örtlich anwendbare Recht.
9. **GESAMTER VERTRAG.** Dieses EULA (einschließlich aller Nachträge oder Ergänzungsvereinbarungen zu diesem EULA, die im Lieferumfang des Produkts enthalten sind) stellt den vollständigen Vertrag zwischen Ihnen und Microsoft in Bezug auf das Produkt und die Supportleistungen (sofern vorhanden) dar. Es hat Vorrang vor allen vorherigen oder gleichzeitigen mündlichen oder schriftlichen Mitteilungen, Vorschlägen und Zusicherungen in Bezug auf das Produkt oder jeden anderen Gegenstand dieses EULAs. Soweit Bestimmungen einer Microsoft-Richtlinie oder eines Microsoft-Programms für Supportleistungen den Bestimmungen dieses EULAs widersprechen, haben die Bestimmungen dieses EULAs Vorrang.
10. Dieses Produkt ist durch Urheberrechtsgesetze und durch andere Gesetze und Abkommen über geistiges Eigentum geschützt. Microsoft oder deren Lieferanten gehören Eigentum, Urheberrecht und andere gewerbliche Schutzrechte an dem Produkt. Das Produkt wird lizenziert, nicht verkauft.

INHALT

| | |
|--|-----------|
| ÜBERSICHT..... | 1 |
| Microsoft Visual Basic .NET Guided Tour | 1 |
| INSTALLATIONSANWEISUNGEN..... | 3 |
| Installationsanweisungen | 3 |
| Installation der Guided Tour | 3 |
| TEIL 1: KURZE TOUR DURCH DIE ANWENDUNG..... | 4 |
| Kurze Tour durch die Anwendung | 4 |
| Zusammenfassung | 8 |
| TEIL 2: ERSTELLEN EINES AUSSAGEKRÄFTIGEN WINDOWS FORMULARS..... | 9 |
| Erstellen eines aussagekräftigen Windows Formulars | 9 |
| TEIL 3: ABRUFEN VON AUFGABENDATEN | 18 |
| Abrufen von Aufgabendaten | 18 |
| TEIL 4: HINZUFÜGEN VON BENUTZERDEFINIERTEN DIAGRAMMSTEUERELEMENTEN..... | 26 |
| Hinzufügen von benutzerdefinierten Diagrammsteuerelementen | 26 |
| TEIL 5: ANPASSEN DES DATAGRIDS | 35 |
| Anpassen des DataGrids | 35 |
| TEIL 6: ERSTELLEN UND VERWENDEN EINES XML-WEBDIENSTES..... | 41 |
| Erstellen und Verwenden eines XML-Webdienstes | 41 |
| TEIL 7: VERWENDEN DER UNBEAUF SICHTIGTEN BEREITSTELLUNG | 59 |
| Verwenden der unbeaufsichtigten Bereitstellung | 59 |
| TEIL 8: VERWENDEN VON CRYSTAL REPORTS (OPTIONAL) | 65 |
| Verwenden von Crystal Reports (Optional) | 65 |
| TEIL 9: VERWENDEN VON ACTIVEX-KOMPONENTEN (OPTIONAL).... | 70 |
| Verwenden von ActiveX-Komponenten (optional) | 70 |
| ANHÄNGE | 75 |
| Anhang 1: Erste Schritte | 75 |
| 1.1 Verwandte Infodateien | 75 |
| 1.2 Systemanforderungen | 76 |

| | |
|--|----|
| Anhang 2: Installation | 77 |
| 2.1 Installationsvorbereitung für Visual Studio .NET | 77 |
| So deinstallieren Sie Visual Studio .NET Beta 2 oder höhere vorab veröffentlichte Versionen | 77 |
| So deinstallieren Sie Visual Studio .NET 7.0 Beta 1 | 78 |
| So deinstallieren Sie Visual Studio 7.0 PDC Technology Preview | 78 |
| 2.2 Installieren von Microsoft® Visual Studio .NET | 79 |
| So installieren Sie Visual Studio .NET | 79 |
| So registrieren Sie Visual Studio .NET | 79 |
| Anhang 3: Aufrüsten auf SQL Server | 80 |
| 3.1 Aktualisieren der Access-Datenbank auf SQL Server | 80 |
| Hinzufügen einer ASPNET-Benutzeranmeldung | 84 |
| Einstellen der Zugriffsberechtigungen für die Datenbank | 89 |
| Anpassung der Anwendung an die neue SQL-Datenbank | 90 |
| Erstellen und Ausführen | 92 |

ÜBERSICHT

Microsoft Visual Basic .NET Guided Tour

Willkommen bei Microsoft® Visual Basic® .NET Guided Tour. Entdecken Sie hier die von Microsoft entwickelte Visual Basic .NET-Sprache. Mit Visual Basic .NET macht die schnelle Anwendungsentwicklung einen weiteren Schritt nach vorn. VB-Entwickler können ihre bereits erworbenen Fähigkeiten voll ausnutzen und gleichzeitig sämtliche Programmieraufgaben mit Hilfe von vertrauten und intuitiven Paradigmen lösen.

Mit Visual Basic .NET können VB-Entwickler:

- Rich-Desktop-Anwendungen mit Windows Forms erstellen
- weitreichende und mobile Webanwendungen erstellen
- unsichtbare Geschäftslogikkomponenten entwickeln, einschließlich XML-Webdiensten

Im Laufe dieser Tour werden Sie feststellen, wie einfach es mit Visual Basic .NET ist, umfassende, unternehmenskritische Lösungen zu erstellen. Der Schwerpunkt dieser Tour liegt auf der Erstellung und Integration der drei Schlüsselkomponenten, aus denen sich die .NET-Plattform der nächsten Generation von Microsoft zusammensetzt: einer Unmenge an **Clientlösungen**, **serverbasierten Softwarekomponenten** und **XML-Webdiensten**. Während der Tour werden Sie auch die leistungsstarke unbeaufsichtigte Bereitstellung (No-Touch-Deployment) von .NET zum Bereitstellen Ihrer Visual Basic .NET Windows Form-Anwendung verwenden.

Visual Basic .NET Guided Tour enthält die folgenden Abschnitte:

Teil 1: Kurze Tour durch die Anwendung

Teil 2: Erstellen eines aussagekräftigen Windows Formulars

Teil 3: Abrufen von Aufgabendaten

Teil 4: Hinzufügen von benutzerdefinierten Diagrammsteuerelementen

Teil 5: Anpassen des DataGrids

Teil 6: Erstellen und Verwenden eines XML-Webdienstes

Teil 7: Verwenden der unbeaufsichtigten Bereitstellung

Die folgenden Abschnitte sind optional:

Teil 8: Verwenden von Crystal Reports

Teil 9: Verwenden von ActiveX-Komponenten

Während der Tour werden Task Management-Informationen in einer lokalen Microsoft Access-Datenbank gespeichert. Anhang 3 dieses Dokuments enthält detaillierte Anweisungen, wie Sie die Anwendung, die Sie während der Tour erstellen, auf Microsoft SQL Server aktualisieren.

Weitere Informationen zu Visual Basic .NET finden Sie im Visual Basic .NET Reviewers Guide unter: <http://msdn.microsoft.com/vbasic/>.

Installationshinweise

Zum Ausführen dieser Guided Tour müssen Sie IIS und eine freigegebene Version von Microsoft Windows® 2000 oder Windows XP und die RTM-Version von Visual Studio .NET installiert haben. Microsoft SQL Server 2000 und Microsoft Excel sind für die *optionalen* Teile der Tour erforderlich.

Diese Demoversion wurde nur unter Windows 2000 und Windows XP Professional getestet.

Anweisungen zur Installation von Visual Studio .NET finden Sie im Anhang und/oder in der Infodatei auf CD 1 von Visual Studio .NET. Aktuelle Informationen finden Sie auf folgender Website: <http://msdn.microsoft.com/>

INSTALLATIONS- ANWEISUNGEN

Installationsanweisungen

Informationen zur Installation von Visual Studio .NET finden Sie in der Infodatei auf CD 1 von Visual Studio .NET bzw. in Anhang A dieses Dokuments.

Installation der Guided Tour

Die Guided Tour funktioniert nur mit Visual Studio .NET ordnungsgemäß. Daneben muss Microsoft Internet Information Server installiert sein und ausgeführt werden. Die Einstellungen der Demoversion verlangen, dass Sie sich als Administrator anmelden. Wenn Sie nicht als Administrator angemeldet sind, müssen Sie zumindest Mitglied der Gruppe **Administratoren** sein.

- a. Kopieren Sie die Datei **Visual Basic .NET Guided Tour.msi** auf Ihren Desktop.
- b. Führen Sie die MSI-Datei aus (doppelklicken Sie auf die Datei).
- c. Klicken Sie auf **Weiter** (Willkommen beim Setup-Assistenten von Visual Studio .NET Guided Tour).
- d. Klicken Sie auf **Weiter** (Installation bestätigen).
- e. Klicken Sie auf **Schließen** (Installation beendet).

TEIL 1: KURZE TOUR DURCH DIE ANWENDUNG

Kurze Tour durch die Anwendung

Übersicht über Teil 1 ▶

Willkommen bei Visual Basic .NET Guided Tour. Diese Tour stellt die drei Hauptbereiche der Visual Basic .NET-Entwicklerplattform vor: die Windows Forms-Entwicklung, das Erstellen und Verwenden von XML-Webdiensten sowie die unbeaufsichtigte Bereitstellung.

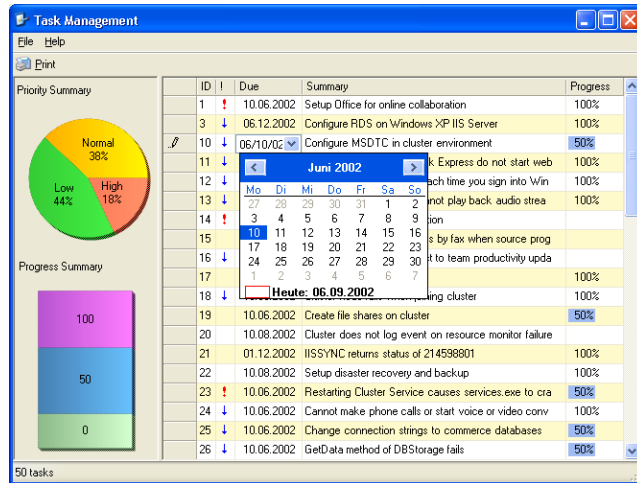
Bei dieser Tour nutzen Sie Visual Basic .NET, um eine Task Management-Anwendung zu erstellen, mit der Sie Softwarefeatureanforderungen, Defekte von benutzerdefinierter Software und Änderungsanforderungen im Allgemeinen verfolgen können. Sie erstellen schnell eine robuste Windows Forms-Anwendung mit den neuen Features zum Verankern und Andocken, und anschließend rufen Sie Änderungsdaten aus einer lokalen Microsoft Access-Datenbank ab. Daraufhin passen Sie die Darstellung der Anwendung mit einem benutzerdefinierten Diagrammsteuerelement an und erstellen einen XML-Webdienst, um die Änderungsverwaltungsdaten abzurufen. Außerdem ermöglichen Sie Benutzern der Anwendung, auf die Änderungsdaten von einem beliebigen PC aus zuzugreifen. Und schließlich wird gezeigt, wie einfach es ist, die Anwendung in einem Intranet oder Extranet mit der unbeaufsichtigten Bereitstellung bereitzustellen.

Zunächst betrachten wir die fertige Task Management-Anwendung, indem wir die aktuellen Aufgaben abrufen und Zusammenfassungsinformationen anzeigen.

Teil 1: Kurze Tour durch die Anwendung

◀ Schritt 1 von 4 Task Management-Anwendung ausführen ▶

- a. Öffnen Sie die [Task Management-Anwendung](#).
(Klicken Sie auf die Schaltfläche **Öffnen**, um die Anwendung zu starten.)



Teil 1: Kurze Tour durch die Anwendung

◀ Schritt 2 von 4 Aufgabenzusammenfassungsdaten anzeigen ▶

Im DataGrid sind alle Aufgaben aufgelistet. Für jede Aufgabe werden eine eindeutige Aufgaben-ID, die Priorität, das Fälligkeitsdatum, eine Zusammenfassung und der Fortschritt aufgezeichnet.

| | ID | ! | Due | Summary | Progress |
|---|----|---|------------|--|----------|
| ▶ | 1 | ! | 10.06.2002 | Setup Office for online collaboration | 100% |
| | 3 | ↓ | 10.06.2002 | Configure RDS on Windows XP IIS Server | 100% |
| | 10 | ↓ | 10.06.2002 | Configure MSDTC in cluster environment | 50% |
| | 11 | ↓ | 10.06.2002 | Internet shortcuts in Outlook Express do not start web | 100% |
| | 12 | ↓ | 10.08.2002 | Prompt asks for upgrade each time you sign into Win | 100% |
| | 13 | ↓ | 10.08.2002 | Windows Media Player cannot play back audio strea | 100% |
| | 14 | ! | 04.08.2002 | Troubleshoot MTA connection | |
| | 15 | | 01.10.2002 | Unable to send attachments by fax when source prog | |
| | 16 | ↓ | 01.10.2002 | Configure server to connect to team productivity upda | |
| | 17 | | 10.08.2002 | Install Mail Server | 100% |
| | 18 | ↓ | 10.08.2002 | Cluster node fails when joining cluster | 100% |
| | 19 | | 10.06.2002 | Create file shares on cluster | 50% |
| | 20 | | 10.08.2002 | Cluster does not log event on resource monitor failure | |
| | 21 | | 01.12.2002 | ISSYNC returns status of 214598801 | 100% |
| | 22 | | 10.08.2002 | Setup disaster recovery and backup | 100% |
| | 23 | ! | 10.06.2002 | Restarting Cluster Service causes services.exe to cra | 50% |
| | 24 | ↓ | 10.06.2002 | Cannot make phone calls or start voice or video conv | 100% |
| | 25 | ↓ | 10.06.2002 | Change connection strings to commerce databases | 50% |
| | 26 | ↓ | 10.06.2002 | GetData method of DBStorage fails | 50% |

- Klicken Sie auf eine andere Spaltenüberschrift, um die Liste der Aufgaben nach dieser Spalte zu sortieren.

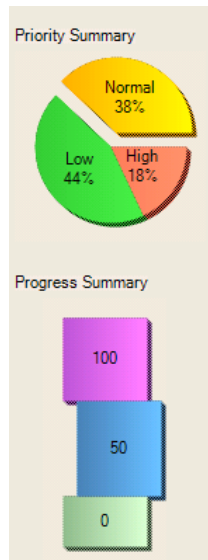
Mit Hilfe der vertikalen Bildlaufleiste können Sie die gesamte Liste der Aufgaben anzeigen.

Teil 1: Kurze Tour durch die Anwendung

◀ Schritt 3 von 4 Aufgabenzusammenfassungsdaten anzeigen ▶

Links im DataGrid wird eine Übersicht über die in der Datenbank enthaltenen Aufgaben nach Priorität und Fortschritt sortiert angezeigt.

- Klicken Sie auf ein Kreissegment im Kreisdiagramm, um dieses Segment hervorzuheben.



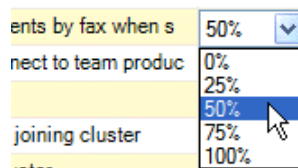
Klicken Sie auf einen Balken im Balkendiagramm, um diesen Balken hervorzuheben.

Teil 1: Kurze Tour durch die Anwendung


◀ Schritt 4 von 4 Aufgabendaten bearbeiten ▶

Änderungen an Aufgaben werden in der Datenbank gespeichert. Änderungen der Spalten für Priorität und Fortschritt spiegeln sich in den Übersichtsdiagrammen wider. Das DataGrid wird so angepasst, dass der Fortschritt als Balkendiagramm und die Priorität als Symbol angezeigt wird.

- Sie können den Fortschritt einer Aufgabe aktualisieren, indem Sie für jede Aufgabe auf den Fortschrittsbalken klicken. Daraufhin wird ein Kombinationsfeld angezeigt.
- Ändern Sie die Priorität, und drücken Sie die **EINGABETASTE**, oder klicken Sie auf eine andere Aufgabe.



| | |
|---------------------|------|
| ents by fax when s | 50% |
| nect to team produc | 0% |
| | 25% |
| | 50% |
| joining cluster | 75% |
| nter | 100% |

- Schließen Sie das Formular, indem Sie auf das Symbol  in der Ecke oben rechts klicken oder indem Sie im Menü **File** auf **Exit** klicken. (Oder drücken Sie **ALT+F4**.)

- **Hinweis**

- Die Diagramme werden automatisch aktualisiert und spiegeln somit die aktualisierte Prioritätsübersicht wider.

Zusammenfassung

Hiermit ist die kurze Tour durch die Task Management-Anwendung beendet. Nun sollten Sie die Anwendung mit Visual Basic .NET und XML-Webdiensten erstellen.

TEIL 2: ERSTELLEN EINES AUSSAGEKRÄFTIGEN WINDOWS FORMULARS

Erstellen eines aussagekräftigen Windows Formulars

Übersicht über Teil 2 ▶

Windows Forms ist das Rich Client-Formularpaket der nächsten Generation von Microsoft, das weit über die bisherigen Grenzen von Desktopanwendungen hinausgeht. Windows Forms gehört zum Lieferumfang von .NET Framework und steht allen Visual Basic .NET-Entwicklern zur Verfügung, die mit ihrem bisherigen Wissen leistungsstarke, zukunftsorientierte Desktopanwendungen erstellen möchten.

Wir möchten eine Windows Form-Anwendung mit Visual Basic .NET erstellen. Zunächst fügen wir dem Formular die grundlegenden Benutzeroberflächenelemente hinzu. Mit dem neuen Windows Forms-Andockfeature wird die Größe der Benutzeroberflächenelemente, die in das Formular eingefügt werden, automatisch angepasst, wenn die Größe des Formulars geändert wird, so dass keine manuellen Anpassungen vorgenommen werden müssen.

Die Visual Studio .NET-Projektmappe und das Visual Basic .NET Windows Form-Projekt sind an dieser Stelle bereits vorhanden, ebenso andere Dateien, die zu einem späteren Zeitpunkt in der Guided Tour verwendet werden.

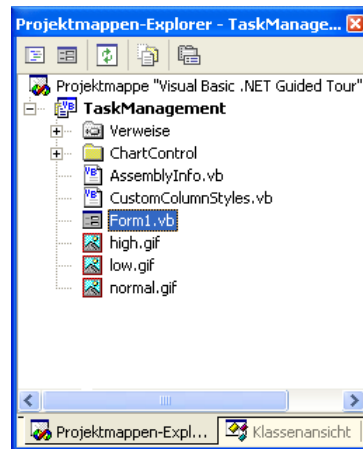
Teil 2: Erstellen eines aussagekräftigen Windows Formulars

◀ Schritt 1 von 6 Dem Hauptmenü Elemente hinzufügen ▶

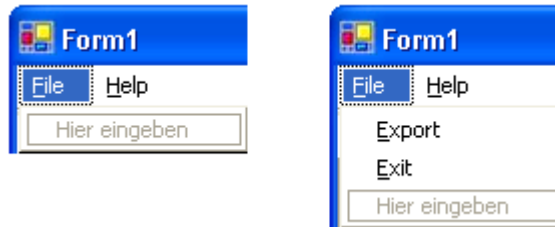
Das Formular enthält bereits ein Hauptmenü, diesem müssen jedoch die Menüelemente **Export** und **Exit** hinzugefügt werden. Wenn der Benutzer das Menüelement **Exit** wählt, wird das Formular geschlossen. Das Menüelement **Export** wird zu einem späteren Zeitpunkt eingebunden.

Visual Studio .NET enthält einen neuen Menüeditor zum Erstellen von Menüs. Ein neues Menüelement kann ganz einfach hinzugefügt werden, indem Sie an die Stelle klicken, an der das neue Element eingefügt werden soll, und anschließend einen Namen für das Menüelement eingeben.

- Zunächst muss **Form1** im Designer angezeigt werden. Klicken Sie im Menü **Ansicht** auf **Projektmappen-Explorer**. (Oder drücken Sie **STRG+R**.)
- Doppelklicken Sie im Projektmappen-Explorer auf **Form1.vb**.



- c. Klicken Sie im Formular auf das Menü mit der Beschriftung **File**.
- d. Klicken Sie anschließend unter dem Menübefehl **File** in das Feld, in dem **Hier eingeben** steht. Geben Sie **&Export** ein, und drücken Sie die **EINGABETASTE**. Dieses Menüelement wird zu einem späteren Zeitpunkt eingebunden.





- **Hinweis**
 - Das **kaufmännische UND-Zeichen** vor dem Buchstaben e legt **ALT+E** als Tastenkombination für dieses Menüelement fest.
 - Klicken Sie unterhalb des neuen Menüelements **Export**, geben Sie **&Exit** ein, und drücken Sie die **EINGABETASTE**.
 - Doppelklicken Sie anschließend auf das Menüelement **Exit**, um den Ereignishandler für dieses Menüelement hinzuzufügen. Wir fügen den Code hinzu, um das Formular zu schließen.

Geben Sie **Me.Close** ein, um den Code hinzuzufügen, mit dem das Formular geschlossen wird, wenn das Menüelement **Exit** gewählt wird.

Teil 2: Erstellen eines aussagekräftigen Windows Formulars

◀ Schritt 2 von 6 Eine Symbolleiste hinzufügen ▶

Visual Studio .NET enthält das neue ToolBar-Steuerelement.

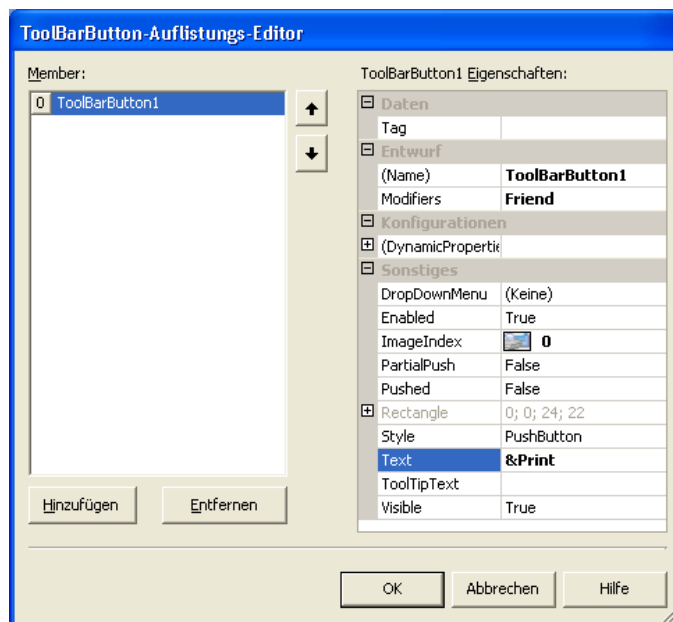
- a. Klicken Sie im Menü **Ansicht** auf **Designer**.
(Oder drücken Sie **UMSCHALT+F7**.)
 - b. Klicken Sie im Menü **Ansicht** auf **Toolbox**.
(Oder drücken Sie **STRG+ALT+X**.)
 - c. Klicken Sie auf die Registerkarte **Windows Forms**.
 - d. Doppelklicken Sie auf das Symbol **ToolBar**, um es dem Formular hinzuzufügen.
 - e. Drücken Sie **F4**, um die Eigenschaften von **ToolBar** anzuzeigen.
 - f. Ändern Sie die Eigenschaft **Appearance** in **Flat**.
 - g. Ändern Sie die Eigenschaft **TextAlign** in **Right**.
 - h. Klicken Sie auf die Eigenschaft **ImageList**, und wählen Sie **ImageList1**.
- **Hinweis**
 - Sie können die Eigenschaften im Eigenschaftfenster alphabetisch sortieren, um sie besser durchsuchen zu können.
Klicken Sie dazu auf die Schaltfläche .
 - Wenn Sie auf die Schaltfläche  klicken, werden die Aufgaben erneut nach Kategorie sortiert angezeigt.

Teil 2: Erstellen eines aussagekräftigen Windows Formulars

◀ Schritt 3 von 6 Der Symbolleiste Schaltflächen hinzufügen ▶

Mit dem neuen ToolbarButton-Auflistungs-Editor können Sie der Werkzeugleiste Schaltflächen hinzufügen.

- Klicken Sie auf die Eigenschaft **Buttons**, und klicken Sie auf die Schaltfläche **...**.
- Klicken Sie auf die Schaltfläche **Hinzufügen**, um eine neue Symbolleisten-schaltfläche mit der Bezeichnung ToolBarButton1 hinzuzufügen.
- Ändern Sie die Eigenschaft **Text** in der Eigenschaftenspalte in **&Print** für die neue Schaltfläche.
- Setzen Sie die Eigenschaft **ImageIndex** auf das erste Bild (Element 0).
- Klicken Sie auf **OK**.



Teil 2: Erstellen eines aussagekräftigen Windows Formulars

◀ Schritt 4 von 6 Dem Formular eine Statusleiste hinzufügen ▶

Jetzt können Sie dem Formular eine Statusleiste hinzufügen.

- a. Klicken Sie im Menü **Ansicht** auf **Toolbox**.
(Oder drücken Sie **STRG+ALT+X**.)
- b. Klicken Sie auf die Registerkarte **Windows Forms**.
- c. Doppelklicken Sie auf das Symbol **StatusBar**, um es dem Formular hinzuzufügen.
(Möglicherweise müssen Sie in der Toolbox einen Bildlauf nach unten durchführen, um das Symbol **StatusBar** anzuzeigen.)
- d. An dieser Stelle sollten wir die Überschrift des Formulars ändern. Klicken Sie auf die Titelleiste des Formulars, und drücken Sie **F4**. Ändern Sie die Eigenschaft **Text** in **Task Management**.

Teil 2: Erstellen eines aussagekräftigen Windows Formulars

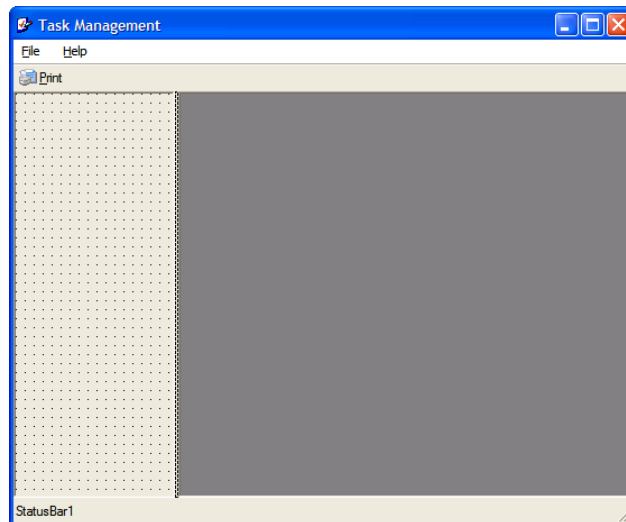
◀ **Schritt 5 von 6** Dem Formular ein Panel- und ein Splitter-Steuerelement hinzufügen ▶

Wir richten das Formular so ein, dass der Benutzer die Größe des Formulars mit einem Splitter-Steuerelement einfach ändern kann. Dieses Steuerelement ist neu im Windows Forms-Designer.

Wir verwenden das neue Windows Forms-Andockfeature, mit der die Größe der Steuerelemente automatisch geändert wird, wenn die Größe des Formulars geändert wird. Bisher war für diese Funktion viel benutzerdefinierter Code zum Ändern der Größe erforderlich. Jetzt müssen Sie nur noch die Andockeigenschaft (Dock) für jedes Steuerelement festlegen. Es ist kein Code erforderlich.

In unserer Anwendung ist der Bereich links vom Splitter-Steuerelement für Übersichtsinformationen reserviert, die wir zu einem späteren Zeitpunkt hinzufügen werden. Der Bereich rechts vom Splitter enthält ein DataGridView-Steuerelement.

Das fertige Formular sieht so aus:




- Klicken Sie im Menü **Ansicht** auf **Toolbox**. (Oder drücken Sie **STRG+ALT+X**.)
- Klicken Sie auf die Registerkarte **Windows Forms**.
- Doppelklicken Sie auf das Symbol **Panel**, und fügen Sie dem Formular ein **Panel**-Steuerelement hinzu. In dieses Panel werden wir zu einem späteren Zeitpunkt während der Tour benutzerdefinierte Diagrammsteuerelemente hinzufügen.
- Jetzt verankern wir dieses Panel mit dem Andockfeature auf der linken Seite des Formulars. Drücken Sie **F4**, um die Eigenschaften des Panels anzuzeigen. Klicken Sie auf die Eigenschaft **Dock**, und geben Sie den Buchstaben **L** ein, um die Eigenschaft von **None** in **Left** zu ändern.
- Ändern Sie die Eigenschaft **BorderStyle** des Panels in **Fixed3D**.

-
- f. Ändern Sie die Größe des **Panel**-Steuerelements so, dass es die linke Seite des Formulars ausfüllt.
 - g. Ziehen Sie anschließend ein **Splitter**-Steuerelement in die rechte Seite des Panels.
 - h. Ziehen Sie das **DataGrid** aus der Toolbox in das Formular rechts neben das **Splitter**-Steuerelement.
 - i. Drücken Sie **F4**, um den Eigenschaftenbereich für das DataGrid anzuzeigen, und setzen Sie die Eigenschaft **Dock** auf **Fill**, indem Sie **F** drücken.
 - j. Setzen Sie die Eigenschaft **CaptionVisible** auf **False**.

Teil 2: Erstellen eines aussagekräftigen Windows Formulars

◀ Schritt 6 von 6 Die Anwendung testen ▶

Mit dem neuen Windows Forms-Andockfeature wird sichergestellt, dass die Größe des DataGrids und des Panels automatisch geändert werden, wenn die Größe des Formulars geändert wird. Testen Sie dies, indem Sie die Anwendung ausführen.

- a. Klicken Sie im Menü Debuggen auf Starten ohne Debuggen.
 - b. Ändern Sie die Größe des Formulars. Daraufhin ändert sich ebenfalls die Größe des Panels, des Splitters und des DataGrids. Ziehen Sie den Splitter nach rechts und nach links, um jeweils das DataGrid bzw. das Panel-Steurelement zu vergrößern.
 - c. Schließen Sie das Formular, indem Sie auf das Symbol  in der Ecke oben rechts klicken oder indem Sie im Menü File auf Exit klicken. (Oder drücken Sie **ALT+F4**.)
- **Wichtig**
 - Sie können die Tastenkombination STRG+F5 drücken, um das Projekt zu erstellen und auszuführen. Stellen Sie sicher, dass das Toolfenster für das Guided Tour-Add-In nicht den Fokus besitzt, wenn Sie **STRG+F5** verwenden. Wenn Sie **STRG+F5** drücken, während das Guided Tour-Toolfenster den Fokus hat, wechselt es zu Teil 1. (In diesem Fall müssen Sie Schritt für Schritt wieder zu diesem Punkt navigieren.)

TEIL 3: ABRUFEN VON AUFGABENDATEN

Abrufen von Aufgabendaten

Übersicht über Teil 3 ▶

Nachdem wir nun die Benutzeroberfläche für die Anwendung Task Management erstellt haben, sollten wir die Aufgabendaten anzeigen.

Mit Microsoft Visual Basic .NET können Sie die leistungsstarken Klassen des .NET Framework nutzen. An dieser Stelle verwenden wir das XML-basierte Microsoft ADO.NET-Programmiermodell, um Aufgabeninformationen aus der Datenbank abzurufen und die Datenbank mit den vom Benutzer vorgenommenen Änderungen zu aktualisieren.

Teil 3: Abrufen von Aufgabendaten

◀ **Schritt 1 von 7** Eine Verbindung zur lokalen Access-Datenbank herstellen ▶

Die Aufgabeninformationen sind in einer lokalen Microsoft Access-Datenbank gespeichert. Wir erstellen ein OleDbConnection-Objekt, das auf die lokale Access-Datenbank verweist.

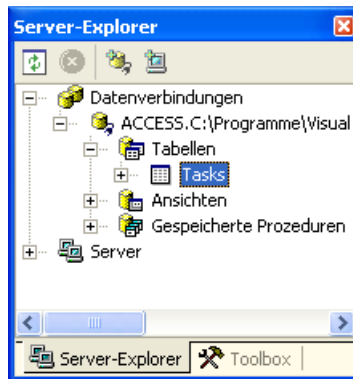
- a. Klicken Sie im Menü **Ansicht** auf **Server-Explorer**.
(Oder drücken Sie **STRG+ALT+S**.)
 - b. Klicken Sie im Menü **Extras** auf **Mit Datenbank verbinden**. Daraufhin wird das Dialogfeld **Datenverknüpfungseigenschaften** angezeigt.
 - c. Klicken Sie auf die Registerkarte **Provider**, und wählen Sie in der Liste der Provider **Microsoft Jet 4.0 OLE DB Provider**.
 - d. Klicken Sie auf **Weiter >>**.
 - e. Klicken Sie auf **...**, und suchen Sie in der Liste nach der Datei **Tasks.mdb**.
(Diese Datei dürfte sich im Ordner C:\Programme\Visual Basic .NET Guided Tour\Projects\TaskManagement\Database befinden.)
 - f. Markieren Sie diese Datei, und klicken Sie auf **Öffnen** (oder doppelklicken Sie auf die Datei **Tasks.mdb**).
 - g. Klicken Sie auf **OK**.
- **Auf Microsoft SQL Server aktualisieren**
 - Sie können von der Access-Datenbank einfach auf Microsoft SQL Server 2000 aktualisieren, indem Sie die Schritte in [Anhang 3](#) ausführen.

Teil 3: Abrufen von Aufgabendaten

◀ Schritt 2 von 7 Dem Formular die Aufgabentabelle hinzufügen ▶

Da wir eine Verbindung zur lokalen Access-Datenbank hergestellt haben, können wir ein OleDbDataAdapter-Objekt erstellen, indem wir die benötigte Tabelle in das Formular ziehen. Die SQL-Anweisungen SELECT, INSERT, UPDATE und DELETE, die zum Ändern von Aufgabendaten erforderlich sind, werden automatisch erstellt.

- Klicken Sie im Menü **Ansicht** auf **Server-Explorer**. (Oder drücken Sie **STRG+ALT+S**.)
- Erweitern Sie den Knoten **Datenverbindungen**, und suchen Sie die Tabelle **Tasks**. (Diese wird unter dem Element **Tabellen** mit der soeben erstellten Verbindung angezeigt.)
- Ziehen Sie das Element **Tasks** in das Formular.



- **Optional**

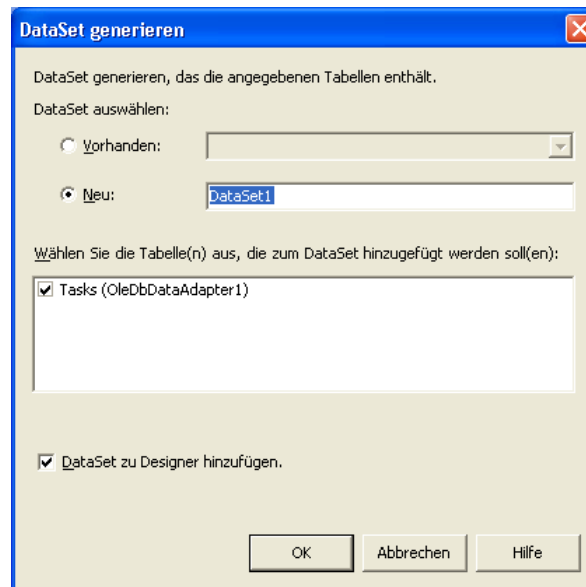
- Sie können eine Vorschau der Tabelle Tasks anzeigen, indem Sie auf das Tasks-Symbol doppelklicken. Auf diese Weise können Sie die Daten in der Tabelle schnell überprüfen oder ändern. Schließen Sie dieses Fenster durch Drücken von **STRG+F4**, bevor Sie zum nächsten Schritt weitergehen.

Teil 3: Abrufen von Aufgabendaten

◀ Schritt 3 von 7 Das Dataset erstellen ▶

Visual Studio .NET generiert automatisch Datenbankschemainformationen durch die Erstellung einer Klasse, die von der ADO.NET-Dataset-Klasse abgeleitet wird. Die leistungsstarke DataSet-Klasse enthält Schemainformationen und die Daten für eine oder mehrere Tabellen. Außerdem kann die DataSet-Klasse Angaben zu Beziehungen zwischen den Tabellen enthalten. In diesem Schritt wird eine neue DataSet-Klasse unter Verwendung des OleDbDataAdapter-Objekts erstellt, das wir im vorherigen Schritt erstellt haben.

- Wählen Sie das Objekt **OleDbDataAdapter1**. Visual Studio wählt standardmäßig beide Objekte aus. Sie können die Auswahl von OleDbConnection1 rückgängig machen, indem Sie in einen leeren Bereich der Designoberfläche klicken oder die **STRG**-Taste gedrückt halten und auf das Verbindungsobjekt klicken.
- Klicken Sie mit der rechten Maustaste auf das Symbol OleDbDataAdapter1, und wählen Sie **DataSet generieren**.
- Ersetzen Sie im Textfeld **Neu** den Text **DataSet1** durch den neuen Namen **TaskSchema**.
- Klicken Sie auf **OK**.



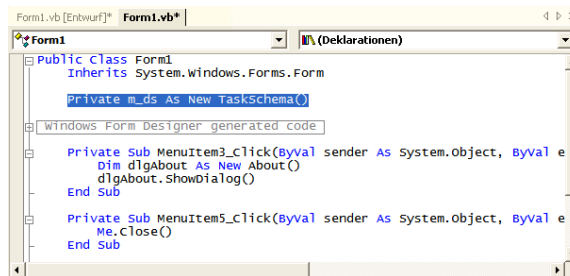
- Visual Studio .NET erstellt automatisch eine XML Schema Definition-Datei mit dem Namen **TaskSchema.xsd**.
- Doppelklicken Sie im Fenster **Projektmappen-Explorer** auf die Datei **TaskSchema.xsd**. Daraufhin wird die Datei **Task Schema Definition** angezeigt. Blättern Sie bis zum Ende der Tabelle, um sich zu vergewissern, dass die **TaskID**, der primäre Schlüssel für die Tabelle, dem Schlüsselssymbol zugeordnet ist.
- Schließen Sie dieses Fenster (**STRG+F4**), bevor Sie zum nächsten Schritt weitergehen.

Teil 3: Abrufen von Aufgabendaten

◀ Schritt 4 von 7 Die DataSet Member-Variable deklarieren ▶

Das neue DataSet muss mit Aufgabendaten gefüllt werden, und es muss an das DataGrid in unserem Formular gebunden werden.

- Klicken Sie im Menü **Ansicht** auf **Code**. (Oder drücken Sie **F7**.)
- Positionieren Sie den Cursor *über* dem Feld **Windows Form Designer generated code**.
- Klicken Sie in der Toolbox auf die Registerkarte **Visual Basic .NET Guided Tour**, und doppelklicken Sie auf **3.4 DataSet Member Variable**, um die Deklaration für die DataSet Member-Variable einzufügen.



```
Form1.vb [Entwurf]* Form1.vb*
Form1
  Inherits System.Windows.Forms.Form
  Private m_ds As New TaskSchema()
  Windows Form Designer generated code
  Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e As EventArgs)
    Dim dlgAbout As New About()
    dlgAbout.ShowDialog()
  End Sub
  Private Sub MenuItem5_Click(ByVal sender As System.Object, ByVal e As EventArgs)
    Me.Close()
  End Sub
```

- **Hinweis**

- Eines der neuen Features von Visual Basic .NET ist, dass Variablen in einer einzigen Anweisung deklariert und initialisiert werden können.

Teil 3: Abrufen von Aufgabendaten

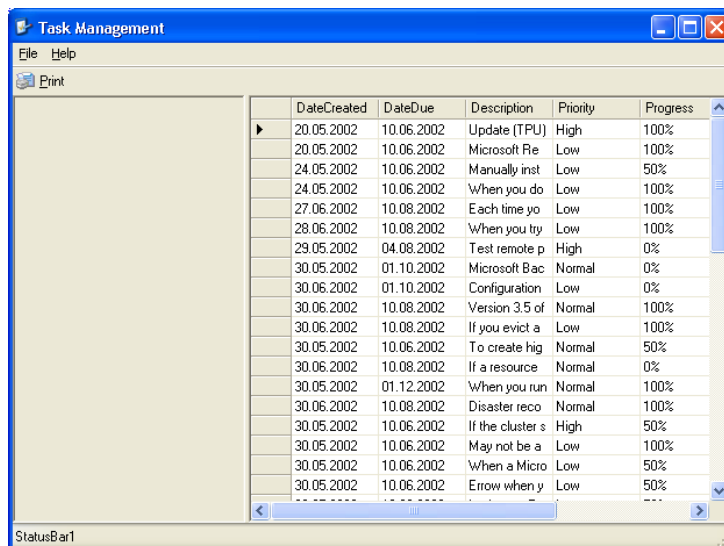
◀ Schritt 5 von 7 Das DataGrid mit Aufgabendaten auffüllen ▶


Mit diesem Code werden die Aufgabendaten aus der lokalen Access-Datenbank abgerufen und in das ADO.NET-DataSet eingefügt. Anschließend binden wir das DataSet in das DataGrid ein, um das DataGrid mit den Aufgabendaten zu füllen. Dieser Schritt wird beim Laden des Formulars ausgeführt.

- Fügen Sie einen Handler für das Ereignis zum Laden des Formulars hinzu, indem Sie in der Objektliste den Eintrag (**Basisklassenergebnisse**) (links oben im Code-Fenster) und in der Ereignisliste das Ereignis **Load** (rechts oben) auswählen. Der Cursor muss *innerhalb* der Methode **Form1_Load** positioniert werden.
- Doppelklicken Sie auf **3.5 Populate DataGrid**, um das DataSet an das DataGrid zu binden und das DataSet mit Aufgabendaten aufzufüllen.

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Load
    OleDbDataAdapter1.Fill(m_ds, "Tasks")
    DataGrid1.DataSource = m_ds.Tasks
End Sub
```

- Testen Sie das Formular. Klicken Sie im Menü **Debuggen** auf **Starten ohne Debuggen**. (Oder drücken Sie **STRG+F5**.)



- Schließen Sie das Formular, indem Sie auf das Symbol  in der Ecke oben rechts klicken oder indem Sie im Menü **File** auf **Exit** klicken. (Oder drücken Sie **ALT+F4**.)

Teil 3: Abrufen von Aufgabendaten

◀ Schritt 6 von 7 Die Statusleiste verwenden ▶

Über die Statusleiste können Übersichtsinformationen angezeigt werden. Wir können die Gesamtzahl der Aufgaben in der Datenbank anzeigen, indem wir eine Eigenschaft des DataSet verwenden.

- Positionieren Sie den Cursor am *Anfang* der Zeile `End Sub` für die Methode **Form1_Load**.
- Doppelklicken Sie auf **3.6 Use the Status Bar**, um die Anzahl der Aufgaben in der Statusleiste anzugeben.

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    OleDbDataAdapter1.Fill(m_ds, "tasks")
    DataGridView1.DataSource = m_ds.Tasks
    StatusBar1.Text = m_ds.Tasks.Rows.Count & " tasks"
End Sub
```


Teil 3: Abrufen von Aufgabendaten

◀ Schritt 7 von 7 Die Datenbank aktualisieren ▶

Das DataGrid kann zum Bearbeiten von Daten verwendet werden. Sie können im DataGrid Änderungen an den Aufgabendaten vornehmen und anschließend das Objekt **OleDbDataAdapter** anweisen, die Datenbank zu aktualisieren. Dazu sind nur einige Zeilen Code erforderlich, da das Objekt **OleDbDataAdapter** eine SQL-Anweisung für die Aktualisierung der Aufgabendaten erstellt hat.

- Fügen Sie für das Ereignis **CurrentCellChanged** einen Ereignishandler hinzu. Wählen Sie in der Objektliste den Eintrag **DataGrid1** und in der Ereignisliste das Ereignis **CurrentCellChanged**. Der Cursor muss *innerhalb* der Methode **DataGrid1_CurrentCellChanged** positioniert werden.
- Doppelklicken Sie auf **3.7 Update Database**, um den Datenadapter anzuweisen, die Datenbank mit unserem DataSet zu aktualisieren.

```
Private Sub DataGrid1_CurrentCellChanged(ByVal sender As Object,
    If m_ds.HasChanges Then
        OleDbDataAdapter1.Update(m_ds, "Tasks")
    End If
End Sub
```

- Testen Sie das Formular. Klicken Sie im Menü **Debuggen** auf **Starten ohne Debuggen**. (Oder drücken Sie **STRG+F5**.)
- Hinweis**
 - Ändern Sie die Priorität der ersten Aufgabe aus High in Low, und klicken Sie anschließend in die nächste Zeile. Das DataSet markiert die Zeile intern als geändert. Das DataGrid-Ereignis **CurrentCellChanged** wird ausgelöst, und die Datenbank wird mit der von Ihnen vorgenommenen Änderung aktualisiert.
 - Schließen Sie das Formular, indem Sie auf das Symbol  in der Ecke oben rechts klicken oder indem Sie im Menü **File** auf **Exit** klicken. (Oder drücken Sie **ALT+F4**.)

TEIL 4: HINZUFÜGEN VON BENUTZERDEFINIERTEN DIAGRAMM- STEUERELEMENTEN

Hinzufügen von benutzerdefinierten Diagrammsteuerelementen

Übersicht über Teil 4 ▶

Visual Basic .NET bietet eine bislang einzigartige Flexibilität beim Erstellen von benutzerdefinierten Steuerelementen. Das Erweitern von bereits vorhandenen Benutzersteuerelementen und Windows Forms-Steuerelementen ist einfach, und Sie können eigene Steuerelemente mit benutzerdefinierter Benutzeroberfläche entwerfen.

Benutzersteuerelemente sind ein großer Fortschritt gegenüber den ActiveX-Steuerelementen in Visual Basic 6.0. Die bei ActiveX-Steuerelementen auftretenden Bereitstellungsprobleme spielen hier keine Rolle mehr, da .NET die Koexistenz von unterschiedlichen Versionen, XCOPY-Bereitstellung und automatisches Downloaden aus dem Web von windowsbasierten Anwendungen unterstützt. Sie können vorhandenen Code übernehmen, da alle ActiveX-Steuerelemente auch in Visual Basic .NET-Anwendungen verwendet werden können.

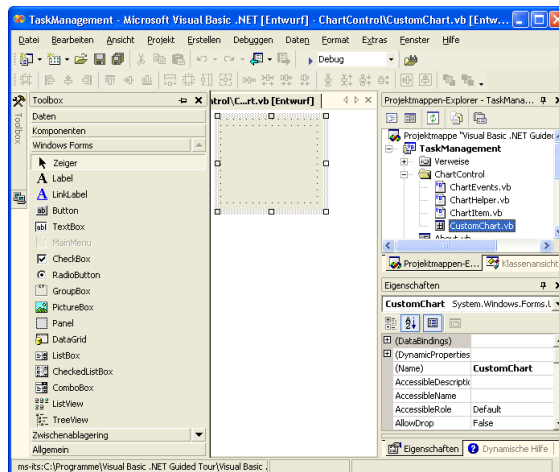
In diesem Teil der Tour wird demonstriert, wie einfach die Verwendung von benutzerdefinierten Steuerelementen in Ihrer Anwendung ist, indem ein Diagrammsteuerelement verwendet wird, das in Visual Basic .NET geschrieben wurde. Das Diagrammsteuerelement wird zum Anzeigen von Prioritäts- und Fortschrittsinformationen im linken Fensterbereich verwendet. Da die Steuerelemente die Datenbindung unterstützen, müssen Sie nur vier Codezeilen schreiben, um zwei interaktive Diagramme zu erhalten, die automatisch neu aufgefüllt werden, wenn sich die zu Grunde liegenden Daten ändern.

Teil 4: Hinzufügen von benutzerdefinierten Diagrammsteuerelementen

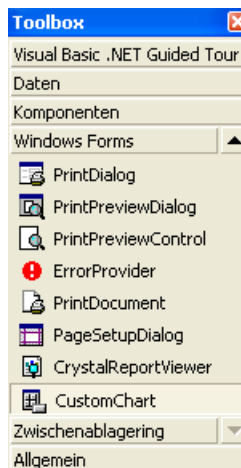
◀ Schritt 1 von 6 Der Toolbox Diagrammsteuerelemente hinzufügen ▶

Es gibt mehrere Möglichkeiten, der Toolbox Benutzersteuerelemente hinzuzufügen. In diesem Schritt wird dargestellt, wie Sie dies erreichen, indem Sie das Benutzersteuerelement in Visual Studio öffnen.

- Öffnen Sie das Fenster **Projektmappen-Explorer**, indem Sie im Menü **Ansicht** die Option **Projektmappen-Explorer** wählen. (Oder drücken Sie **STRG+R**.)
- Erweitern Sie im Projektmappen-Explorer den Ordner **ChartControl**, und doppelklicken Sie anschließend auf **CustomChart.vb**, um das Benutzersteuerelement zu öffnen.



- Visual Studio erkennt, dass es sich hierbei um ein Benutzersteuerelement handelt, und fügt der Toolbox ein Symbol hinzu, wie unten dargestellt. Schließen Sie das Fenster **ChartControl.vb**, indem Sie **STRG+F4** drücken.



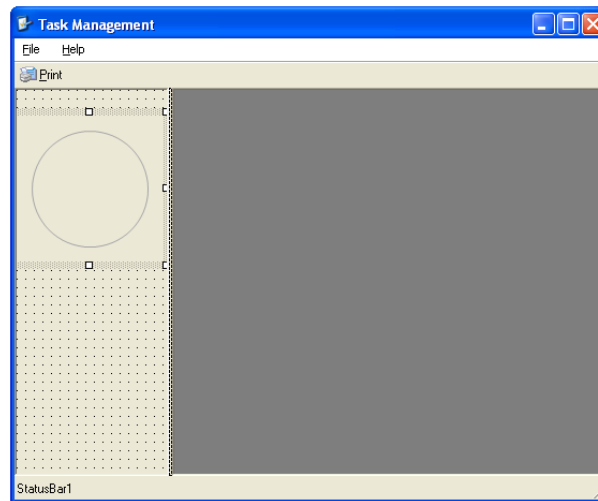
- Hinweis**
 - Sie können der Toolbox auch Symbole hinzufügen, indem Sie im Menü **Extras** die Option **Toolbox anpassen** wählen (oder **STRG+T** drücken) und zu der Assembly navigieren, die das Steuerelement enthält.

Teil 4: Hinzufügen von benutzerdefinierten Diagrammsteuerelementen

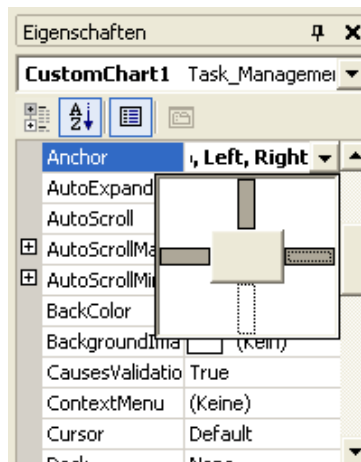
◀ Schritt 2 von 6 Kreisdiagramm für die Priorität hinzufügen ▶

Fügen Sie ein benutzerdefiniertes Diagrammsteuerelement hinzu, und konfigurieren Sie es als Kreisdiagramm, das Werte und Prozentsätze anzeigt.

- Öffnen Sie Form1, indem Sie im Projektmappen-Explorer auf **Form1.vb** doppelklicken.
- Klicken Sie im Menü **Ansicht** auf **Toolbox**. (Oder drücken Sie **STRG+ALT+X**.)
- Klicken Sie in der Toolbox auf die Registerkarte **Windows Forms**.
- Ziehen Sie das Steuerelement **CustomChart** in den linken Fensterbereich von **Form1**. Stellen Sie Größe und Position des Steuerelements wie unten gezeigt ein.

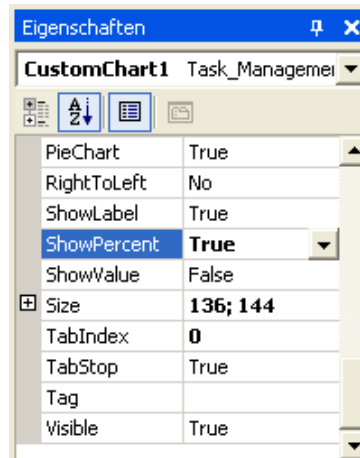


- Drücken Sie **F4**, um die Eigenschaften des Steuerelements anzuzeigen. Klicken Sie auf die Eigenschaft **Anchor**, und setzen Sie den Wert auf **Top, Left, Right**.



- **Hinweis**

- Die Verankerung ist eine neue Funktion von Visual Basic .NET und ermöglicht die Erstellung größenveränderlicher Formulare ohne das Schreiben von Code. Die Verankerungseigenschaft (Anchor) legt fest, welche Seiten an das Container-Steuerelement gebunden werden. Wenn z. B. die rechte Seite verankert werden soll, bleibt die rechte Seite mit dem Container verbunden, wenn die Größe verändert wird.
- f. Doppelklicken Sie auf die Eigenschaft **ShowPercent**, um den Wert von False in **True** zu ändern.

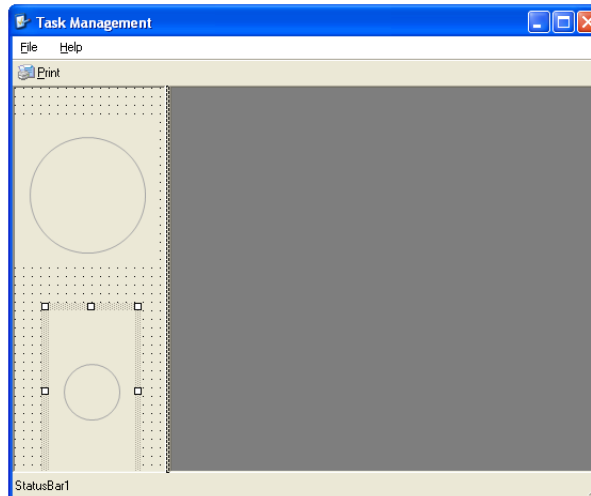


Teil 4: Hinzufügen von benutzerdefinierten Diagrammsteuerelementen

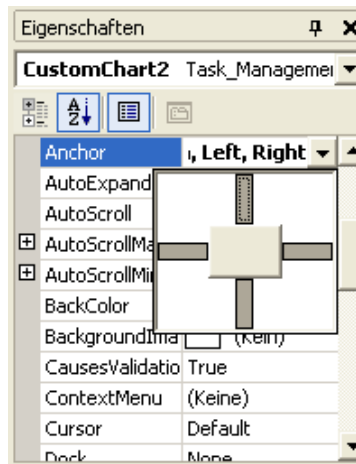
◀ Schritt 3 von 6 Kreisdiagramm für den Fortschritt hinzufügen ▶

Fügen Sie ein weiteres benutzerdefiniertes Diagrammsteuerelement hinzu, und konfigurieren Sie es als Balkendiagramm, das Werte anzeigt.

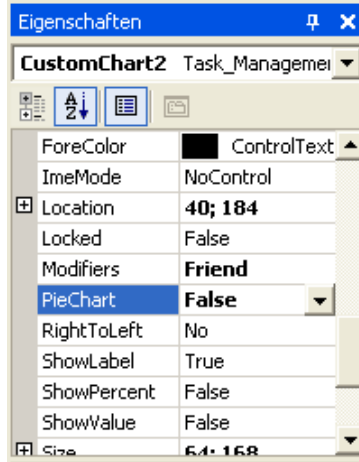
- Ziehen Sie das Steuerelement **CustomChart** in den linken Fensterbereich von **Form1** unter das Kreisdiagramm. Stellen Sie Größe und Position des Steuerelements wie unten gezeigt ein.



- Drücken Sie **F4**, um die Eigenschaften des Steuerelements anzuzeigen. Klicken Sie auf die Eigenschaft **Anchor**, und setzen Sie den Wert auf **Top, Bottom, Left, Right**.



- c. Doppelklicken Sie auf die Eigenschaft **PieChart**, um den Wert von True in **False** zu ändern.

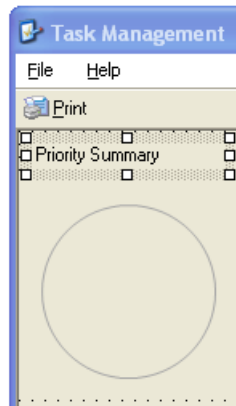


Teil 4: Hinzufügen von benutzerdefinierten Diagrammsteuerelementen

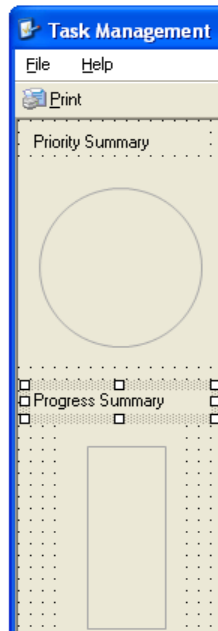
◀ Schritt 4 von 6 Beschriftungen über den Diagrammsteuerelementen hinzufügen ▶

Fügen Sie Beschriftungen über den Diagrammen hinzu, um anzuzeigen, welche Informationen die Diagramme enthalten.

- Ziehen Sie ein **Label**-Steuerelement in den linken Fensterbereich, und legen Sie es wie unten gezeigt über dem Kreisdiagramm ab.



- Drücken Sie **F4**, um die Eigenschaften des Steuerelements anzuzeigen. Setzen Sie die Eigenschaft **Text** auf **Priority Summary**.
- Ziehen Sie ein weiteres **Label**-Steuerelement in den linken Fensterbereich, und legen Sie es wie unten gezeigt über dem Balkendiagramm ab.



- Drücken Sie **F4**, um die Eigenschaften des Steuerelements anzuzeigen. Setzen Sie die Eigenschaft **Text** auf **Progress Summary**.

Teil 4: Hinzufügen von benutzerdefinierten Diagrammsteuerelementen

◀ Schritt 5 von 6 Daten an die Diagrammsteuerelemente binden ▶

Durch das Binden von Daten werden die Diagrammsteuerelemente aktualisiert, wenn die zu Grunde liegenden Daten geändert werden, ohne dass zusätzlicher Code geschrieben werden muss. Wir werden das Kreisdiagramm an das Datenbankfeld **Priority** und das Balkendiagramm an das Feld **Progress** binden.

- Doppelklicken Sie auf den Titel **Task Management** in **Form1**. Dadurch wird der Code für die Sub-Prozedur **Form1_Load** angezeigt.
- Positionieren Sie den Cursor *links* neben der Anweisung **End Sub** für die Prozedur **Form1_Load**.
- Klicken Sie auf die Registerkarte **Visual Basic .NET Guided Tour**, und doppelklicken Sie auf **4.5 Data Bind Chart Controls**, um den Code einzufügen, der die Diagrammsteuerelemente bindet.

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e
    OleDbDataAdapter1.Fill(m_ds, "Tasks")
    DataGridView1.DataSource = m_ds.Tasks

    StatusBar1.Text = m_ds.Tasks.Rows.Count & " tasks"

    CustomChart1.DataSource = m_ds.Tasks
    CustomChart1.DataMember = "Priority"

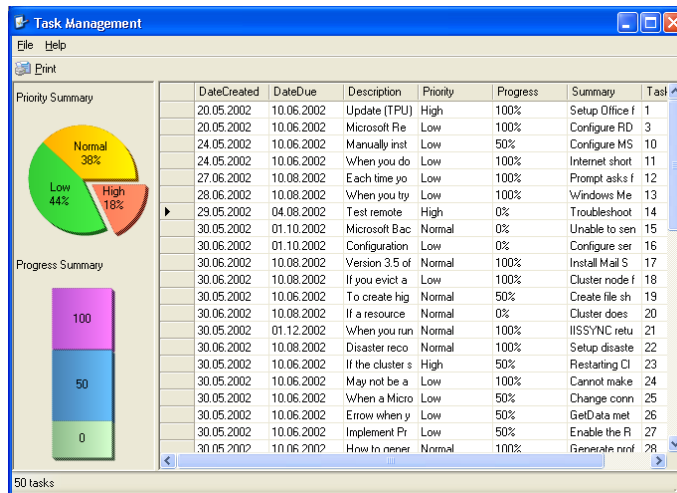
    CustomChart2.DataSource = m_ds.Tasks
    CustomChart2.DataMember = "Progress"
End Sub
```


Teil 4: Hinzufügen von benutzerdefinierten Diagrammsteuerelementen

◀ Schritt 6 von 6 Erstellen und Ausführen ▶

Die Erstellung der benutzerdefinierten Diagrammsteuerelemente ist jetzt beendet, und wie Sie gesehen haben, wurde nur ganz wenig Code benötigt. Dieser Schritt zeigt, wie die Diagrammsteuerelemente in der Anwendung funktionieren.

- Klicken Sie im Menü **Debuggen** auf **Starten ohne Debuggen**, um die Anwendung auszuführen. (Oder drücken Sie **STRG+F5**.)



- Ändern Sie die Größe des Formulars, und verschieben Sie den Splitter. Beachten Sie, wie sich die Größe der Diagramme an den verfügbaren Platz im linken Fensterbereich anpasst, da wir die Anchor-Eigenschaft aktiviert haben.
- Klicken Sie auf verschiedene Elemente in den Diagrammen. Die markierten Elemente werden jeweils hervorgehoben.
- Ändern Sie einige Daten im DataGrid. Ändern Sie zum Beispiel die Priorität einer Zeile in der Spalte **Priority** von **Low** in **High**. Das Kreisdiagramm wird sofort aktualisiert, da es an das zu Grunde liegende DataSet gebunden ist.
- Schließen Sie das Formular, indem Sie auf das Symbol  in der Ecke oben rechts klicken oder indem Sie im Menü **File** auf **Exit** klicken. (Oder drücken Sie **ALT+F4**.)

TEIL 5: ANPASSEN DES DATAGRIDS

Anpassen des DataGrids

Übersicht über Teil 5?

Mit Visual Basic .NET ist es problemlos möglich, alle Bereiche unserer Anwendung anzupassen. In unserem DataGrid können wir die Aufgabenpriorität mit Hilfe von Bildern anstelle von Text darstellen, indem wir eine eigene benutzerdefinierte DataGrid-Spalte hinzufügen. Auf die gleiche Weise können wir auch den Fortschritt einer Aufgabe als einfaches Fortschrittsdiagramm anzeigen.

Da Visual Basic .NET eine vollständig objektorientierte Sprache ist, können wir durch Vererbung von der `DataGridColumnStyle`-Basisklasse eine eigene `DataGridColumnStyle`-Klasse erstellen.

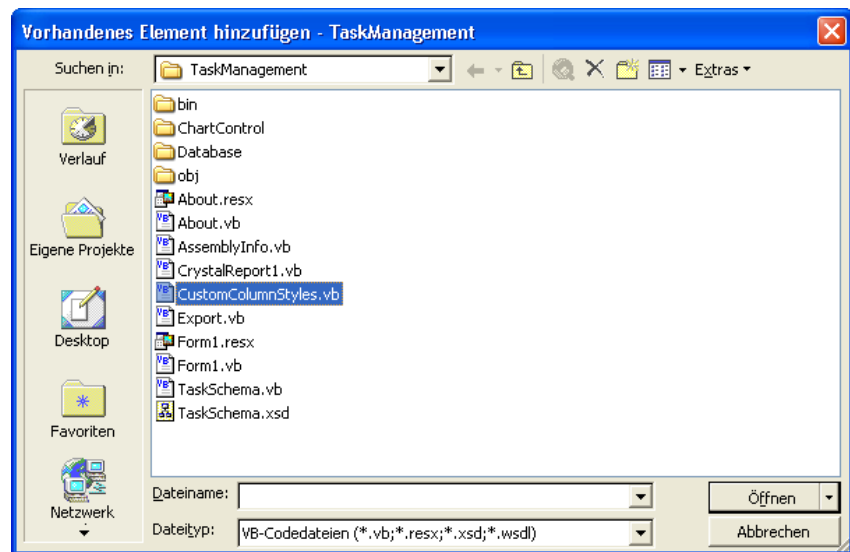
In diesem Teil werden die benutzerdefinierten DataGrid-Spalten, die durch Verwenden der Vererbung in der Datei `CustomColumnStyles.vb` implementiert wurden, zum Anpassen des DataGrids verwendet.

Teil 5: Anpassen des DataGrids

◀ Schritt 1 von 5 Die benutzerdefinierten Rasterstile hinzufügen ▶

Zunächst müssen wir unserem Projekt die Datei mit den benutzerdefinierten DataGridViewColumnStyle-Klassen hinzufügen.

- Klicken Sie im Menü **Ansicht** auf **Projektmappen-Explorer**.
(Oder drücken Sie **STRG+R**.)
- Klicken Sie im Menü **Projekt** auf **Vorhandenes Element hinzufügen**.
(Oder drücken Sie **STRG+D**.)
- Doppelklicken Sie in der Dateiliste auf **CustomColumnStyles.vb**.
(Die Datei **CustomColumnStyles.vb** befindet sich im Ordner **C:\Programme\Visual Basic .NET Guided Tour\Projects\TaskManagement**).



Teil 5: Anpassen des DataGrids

◀ Schritt 2 von 5 Tabellenstile für das DataGrid erstellen ▶

Die Spalten in einem DataGrid können mit Hilfe eines Tabellenstils definiert werden. Ein Tabellenstil enthält eine oder mehrere GridColumnStyle-Definitionen. Mit Hilfe der Klassen, die wir dem Projekt im vorherigen Schritt hinzugefügt haben, weisen wir unserem DataGrid einen neuen Tabellenstil hinzu, der die benutzerdefinierten DataGridViewColumnStyles verwendet.

- Klicken Sie im Projektmappen-Explorer auf **Form1.vb**.
- Klicken Sie im Menü **Ansicht** auf **Code**. (Oder drücken Sie **F7**.)
- Setzen Sie den Cursor in der Methode **Form1_Load** an den *Anfang* der Zeile `StatusBar1.Text = ...`
- Klicken Sie in der Toolbox auf die Registerkarte **Visual Basic .NET Guided Tour**, und doppelklicken Sie auf **5.2 Create DataGrid TableStyle**, um den Code einzufügen, der zum Zuweisen des neuen Tabellenstils zum DataGrid erforderlich ist.

```
datagrid1.TableStyles.Add(CustomDataGridViewStyles.CreateTableStyle())  
StatusBar1.Text = m_ds.Tasks.Count & " tasks"
```

• Hinweis

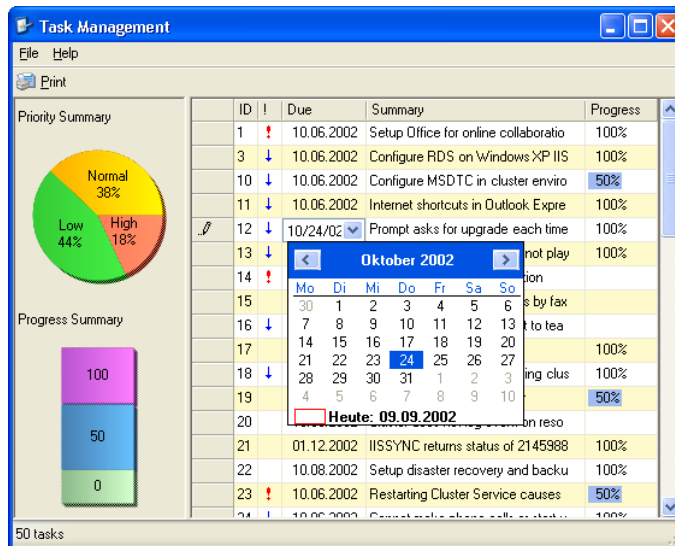
- Zum Erstellen des Tabellenstils wird die Methode **CreateTableStyle** in der Klasse **CustomDataGridViewStyles** aufgerufen. Beachten Sie, dass wir die Klasse nicht deklarieren und keine Instanz der Klasse erstellen mussten, bevor wir sie verwendet haben. Die Methode **CreateTableStyle** wird als freigegebene Methode deklariert (**Shared**). Freigegebene Elemente werden nicht einer bestimmten Instanz einer Klasse zugeordnet. Auf diese kann zugegriffen werden, indem sie mit dem entsprechenden Klassennamen gekennzeichnet werden.

Teil 5: Anpassen des DataGrids


◀ Schritt 3 von 5 Das benutzerdefinierte DataGrid testen ▶

Führen Sie die Task Management-Anwendung aus, um das benutzerdefinierte DataGrid anzuzeigen.

- a. Klicken Sie im Menü **Debuggen** auf **Starten ohne Debuggen**. (Oder drücken Sie **STRG+F5**.)



• Hinweis

- Die Priorität der Aufgaben wird als Symbol dargestellt (↓ oder !). Wenn kein Symbol angezeigt wird, liegt keine Priorität vor (der Wert ist Normal). Der Fortschritt der Aufgaben wird als Fortschrittsbalken angezeigt. Außerdem bieten die Spalten **Priority**, **Due** und **Progress** zusätzliche Bearbeitungsfunktionen, denn Sie können die Steuerelemente **ComboBox** und **DateTimePicker** wie oben dargestellt verwenden.
- b. Schließen Sie das Formular, indem Sie auf das Symbol  in der Ecke oben rechts klicken oder indem Sie im Menü **File** auf **Exit** klicken. (Oder drücken Sie **ALT+F4**.)

Teil 5: Anpassen des DataGrids

◀ Schritt 4 von 5 Die Größe der Spalte Summary automatisch anpassen ▶

Wenn die Größe des Formulars geändert wird, ändert sich auch die Größe des DataGrids, da die Eigenschaft **Dock** auf **Fill** gesetzt ist. Es wäre jedoch besser, wenn die Größe der Spalte **Summary** angepasst würde, wenn sich die Breite des Formulars und des DataGrids ändert.

Im Ereignis **SizeChanged** des DataGrids kann die Größe der Spalte **Summary** entsprechend der neuen Größe des DataGrids angepasst werden.

- Fügen Sie für das Ereignis **SizeChanged** des DataGrids einen Ereignishandler hinzu. Wählen Sie in der Objektliste den Eintrag **DataGrid1** und in der Ereignisliste das Ereignis **SizeChanged**. Der Cursor muss *innerhalb* der Methode **DataGrid1_SizeChanged** positioniert werden.
- Doppelklicken Sie in der Toolbox auf **5.4 SizeChanged Event**, um den Code für den Ereignishandler hinzuzufügen.


```
Private Sub DataGrid1_SizeChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles DataGrid1.SizeChanged
    Try
        ' Determine width of all columns
        Dim colTotal As Integer
        Dim col As DataGridViewColumnStyle
        For Each col In DataGrid1.TableStyles(0).GridColumnStyles
            colTotal += col.Width
        Next

        ' Resize Summary column (column 3)
        DataGrid1.TableStyles(0).GridColumnStyles(3).Width += _
            DataGrid1.Width - colTotal - _
            DataGrid1.RowHeaderWidth - _
            SystemInformation.VerticalScrollBarWidth - _
            (SystemInformation.BorderSize.Width * 2)
    Catch
    End Try
End Sub
```

Teil 5: Anpassen des DataGrids

◀ Schritt 5 von 5 Die Anwendung testen ▶

Führen Sie die Task Management-Anwendung aus, um zu überprüfen, wie sich die Größe der Spalte **Summary** ändert, wenn sich die Größe des Formulars ändert.

- a. Klicken Sie im Menü **Debuggen** auf **Starten ohne Debuggen**. (Oder drücken Sie **STRG+F5**.)
 - b. Ändern Sie die Größe des Formulars.
- **Hinweis**
 - Die Größe der Spalte **Summary** ändert sich jetzt im DataGrid, wenn sich die Größe des Formulars ändert.
- c. Schließen Sie das Formular, indem Sie auf das Symbol  in der Ecke oben rechts klicken oder indem Sie im Menü **File** auf **Exit** klicken. (Oder drücken Sie **ALT+F4**.)

TEIL 6: ERSTELLEN UND VERWENDEN EINES XML-WEBDIENSTES

Erstellen und Verwenden eines XML-Webdienstes

Übersicht über Teil 6 ▶

Bisher haben Sie erfahren, dass das Erstellen umfassender Windows-basierter Anwendungen mit Visual Basic®.NET sehr einfach ist. In diesem Teil der Tour wird Ihnen gezeigt, mit welcher Leichtigkeit ein XML-Webdienst für den Datenzugriff generiert werden kann.

Bei Verwendung eines XML-Webdienstes für den Datenzugriff werden die Zugriffsdetails nicht unmittelbar in der Windows-Anwendung angezeigt. Sie können daher die im Hintergrund liegende Datenbank frei nach Bedarf ändern (z. B. von Access auf SQL Server wechseln), ohne dabei auf die Client-Anwendung zuzugreifen.

Zuerst wird zum Erstellen des XML-Webdienstes die RAD-Funktionalität (Rapid Application Development; Schnelle Anwendungsentwicklung) von Visual Studio.NET verwendet. Die Erstellung des Dienstes ist genau so einfach wie die Erstellung einer Klasse in Visual Basic 6.0. Die neue Projektvorlage für XML-Webdienste erzeugt die zu Grunde liegende Infrastruktur, so dass Sie sich auf die Einzelheiten Ihrer Vorgehensweise konzentrieren können. Dieser Vorgang ähnelt dem Erstellen einer Funktion in Visual Basic 6.0, mit dem Unterschied, dass die Funktion nun im Web für zahlreiche Anwendungen zur Verfügung gestellt werden kann.

Danach ändern wir die Task Management-Anwendung, um den Dienst für den Datenzugriff nutzen zu können. Der XML-Webdienst und die Windows-Anwendungsprojekte sind Teil einer Projektmappe und verwenden zum Entwickeln beider Projekte dieselbe IDE. Alle diese hilfreichen Funktionen stehen beim Erstellen von XML-Webdiensten zur Verfügung.

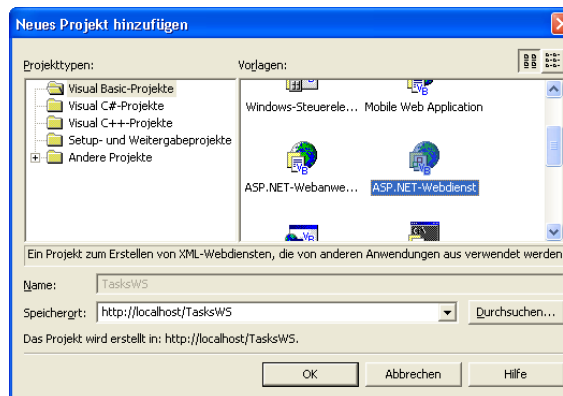
Das Debuggen ist ähnlich einfach. Die Projekte integrieren sich nahtlos und können beide in derselben IDE gedebuggt werden.

Am Ende dieses Teils der Guided Tour haben Sie den notwendigen Einblick in die Bedeutung und Leistung von XML-Webdiensten gewonnen. Außerdem wissen Sie, wie schnell und einfach Entwickler mit Hilfe von Microsoft Visual Studio .NET Webdienste erstellen und diese im Internet und in Windows-basierten Anwendungen integrieren können.

◀ **Schritt 1 von 14** XML-Webdienst erstellen ▶

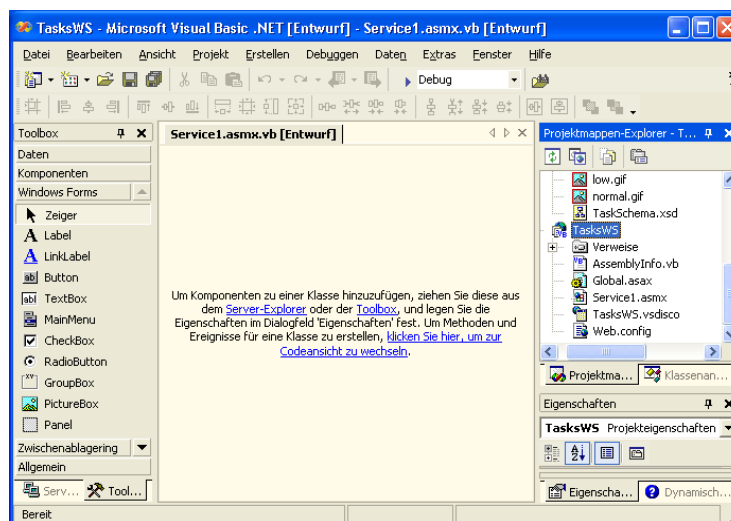
Zuerst wird ein neuer XML-Webdienst erstellt und in die Projektmappe integriert. Die Projektmappe umfasst die Windows-Anwendung Task Management und die Webservice1-Projekte.

- Öffnen Sie das Fenster **Projektmappen-Explorer**, indem Sie im Menü **Ansicht** auf die Option **Projektmappen-Explorer** klicken. (Oder drücken Sie **STRG+R**.)
- Klicken Sie am oberen Fensterrand auf **Projektmappe 'Visual Basic .NET Guided Tour'**.
- Klicken Sie im Menü **Datei** auf **Projekt hinzufügen** und dann auf **Neues Projekt**.
- Klicken Sie in der Liste **Projekttypen** auf **Visual Basic-Projekte**. Klicken Sie in der Liste **Vorlagen** auf das Symbol **ASP.NET-Webdienst**. Geben Sie als Verzeichnis **http://localhost/TaskWS** ein. Klicken Sie auf **OK**, um das Dialogfeld zu schließen.



• **Hinweis**

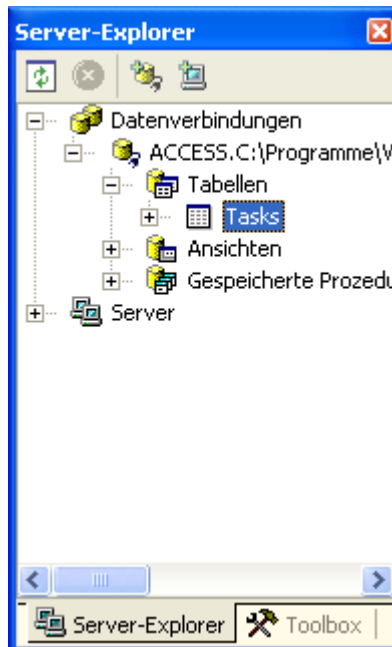
- Damit wird das TaskWS-Projekt in die Projektmappe integriert und **Service1** in Visual Studio.NET geöffnet.



◀ **Schritt 2 von 14** Datenzugriffsobjekte erstellen ▶

Verwenden Sie zum Erstellen der Datenzugriffsobjekte die RAD-Funktionen von Visual Studio.NET, indem Sie die Objekte einfach per Drag & Drop ziehen und ablegen.

- a. Öffnen Sie das Fenster **Server-Explorer**, indem Sie im Menü **Ansicht** auf die Option **Server-Explorer** klicken. (Oder drücken Sie **STRG+ALT+S**.)
- b. Erweitern Sie das Verzeichnis **Datenverbindungen**, und suchen Sie in der Tabelle **Tasks** nach der im vorigen Abschnitt der Tour erstellten Verbindung (Datenverbindungen » ACCESS-Datenbank » Tabellen).



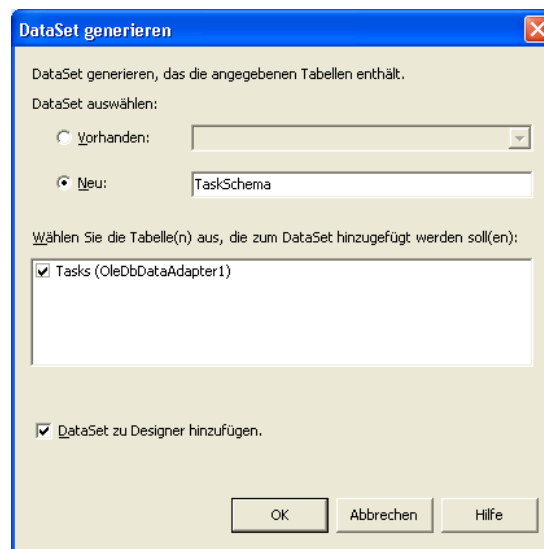
- c. Ziehen Sie die Tabelle **Tasks** auf die Entwurfsoberfläche. Visual Studio erstellt auf Grundlage der Tabelle **Tasks** zwei Datenobjekte, **OleDbConnection1** und **OleDbDataAdapter1**, und fügt diese in das Komponentenfach ein.

• **Hinweis**

- Sie verwenden dieselbe lokale Access-Datenbank wie in den vorigen Teilen der Guided Tour. Im Unterschied dazu befindet sich aber nun zwischen der Task Management-Anwendung und den Daten ein XML-Webdienst. Die Datenbank kann sowohl auf einem Netzwerkserver als auch als SQL Server-Datenbank angelegt werden.

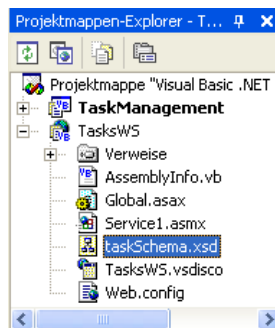
Verwenden Sie die RAD-Funktionen von Visual Studio. NET, um ein stark typisiertes DataSet zu erzeugen.

- Wählen Sie das Objekt **OleDbDataAdapter1** aus. Visual Studio wählt standardmäßig beide Datenobjekte aus. Sie können die Auswahl von OleDbConnection1 aufheben, indem Sie in der Entwurfsoberfläche auf eine leere Fläche klicken oder mit gedrückter **STRG**-Taste auf das Verbindungsobjekt klicken.
- Klicken Sie mit der rechten Maustaste auf das Objekt **OleDbDataAdapter1**, und wählen Sie die Option **DataSet generieren**.
- Benennen Sie das Standarddataset **DataSet1** in **TaskSchema** um. Klicken Sie auf **OK**, um das Dialogfeld zu schließen.



• **Hinweis**

- Das typisierte DataSet **TaskSchema.xsd** wurde dem Projekt hinzugefügt. Damit können Benutzer des XML-Webdienstes mit Hilfe von benutzerfreundlichen Namen und den stark typisierten Variablen auf die Daten zugreifen. Dank der Visual Studio. NET-Funktionen IntelliSense und der Codevervollständigung erreichen Sie eine verbesserte Lesbarkeit des Codes und höhere Entwicklungsgeschwindigkeiten.



Die Methoden für XML-Webdienste lassen sich genau so einfach erstellen wie die Funktionen in den Anwendungen von Visual Basic 6.0. Als Nächstes werden die Methoden **GetTasks** und **UpdateTasks** für diesen Dienst implementiert. **GetTasks** gibt in der aufrufenden Anwendung ein DataSet zurück, und **UpdateTasks** aktualisiert die zu Grunde liegende Datenbank anhand des DataSets, das als Argument übergeben wurde.

- Klicken Sie im Menü **Ansicht** auf **Code**. (Oder drücken Sie **F7**.)
- Positionieren Sie den Cursor *links* neben der **End Class**-Anweisung im unteren Teil der Datei.
- Klicken Sie im Menü **Ansicht** auf **Toolbox**. (Oder drücken Sie **STRG+ALT+X**.)
- Klicken Sie auf die Registerkarte **Visual Basic .NET Guided Tour**, und doppelklicken Sie dann auf **6.4 Web Service Methods**, um den Code einzufügen.

```

Service1.asmx.vb [Entwurf]* Service1.asmx.vb*
Service1
UpdateTasks
' <WebMethod()> Public Function HelloWorld() As String
'     HelloWorld = "Hello World"
' End Function

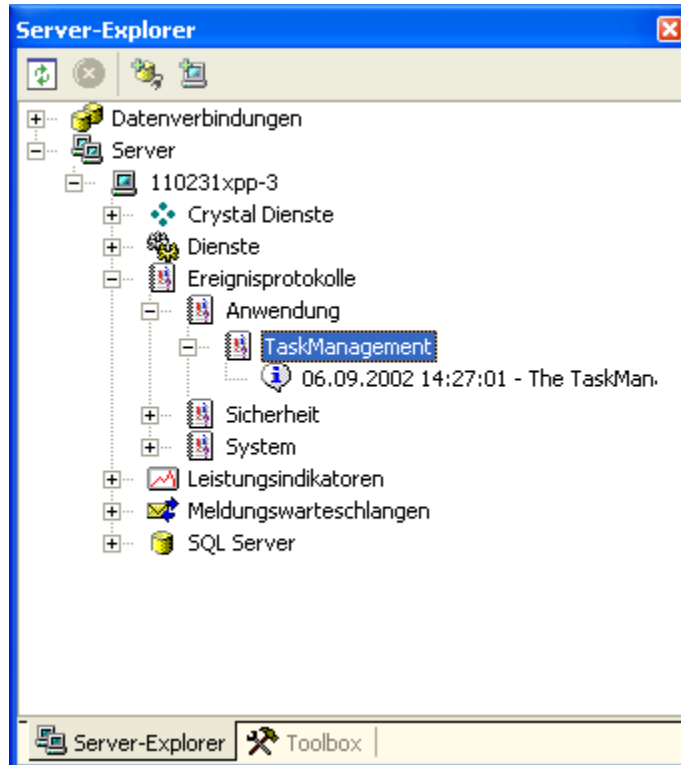
<WebMethod()> Public Function GetTasks() As taskSchema
    Dim ds As New taskSchema()
    Try
        OleDbDataAdapter1.Fill(ds)
        PerformanceCounter1.Increment()
    Catch ex As Exception
        EventLog.WriteEntry("TaskManagement", ex.Message)
    End Try
    Return ds
End Function

<WebMethod()> Public Function UpdateTasks(ByVal ds As tas
    Dim updatedDS As New taskSchema()
    Try
        OleDbDataAdapter1.Update(ds)
        OleDbDataAdapter1.Fill(updatedDS)
    Catch ex As Exception
        EventLog.WriteEntry("TaskManagement", ex.Message)
    End Try
    Return updatedDS
End Function
End Class
    
```

- **Hinweis**
 - Visual Basic .NET unterstützt die **Return**-Anweisung und die strukturierte Ausnahmebehandlung. Das **WebMethod**-Attribut legt Ihre Methode im Web offen.
 - Im Beispielcode wird das **PerformanceCounter1**-Objekt verwendet, das noch nicht erstellt wurde. Visual Studio kennzeichnet die Stelle mit einer blauen Wellenlinie. Neben den Hervorhebungen durch Wellenlinien enthält Visual Studio .NET zahlreiche weitere Verbesserungen des Code-Editors, z. B. verbesserte IntelliSense-Funktionen, intelligente Auflistung von Code zur besseren Lesbarkeit und einen Hintergrundcompiler für Benachrichtigungen über Syntaxfehler in Echtzeit. Mit Hilfe von Visual Studio .NET können Sie schneller und effektiver Programmcode erstellen.

- **Hinweis**

- Der Dienst schreibt unter Verwendung einer benutzerdefinierten Quelle in das Ereignisprotokoll. Sie könnten direkt in das Ereignisprotokoll in Visual Basic 6.0 schreiben, aber Sie müssten dann alle API-Aufrufe deklarieren und aufrufen, um eine vollständige Kontrolle über die Protokollierung zu erlangen. In Visual Basic .NET werden jetzt viele Vorgänge unterstützt, die Sie zuvor in Visual Basic 6.0 mit API-Aufrufen ausführen mussten.
- Das Ereignisprotokoll können Sie direkt aus Visual Studio heraus im Server-Explorer-Fenster anzeigen. Das Ereignisprotokoll für Task Management sieht in diesem Fall wie folgt aus:



◀ **Schritt 5 von 14** Einen neuen Leistungsindikator hinzufügen ▶

Wenn Sie Visual Studio .NET verwenden, sind RAD-Serverfunktionen, Komponenten und serverseitige Ressourcen wie Leistungsindikatoren und Meldungswarteschlangen in programmgesteuerte Schnittstellen eingebunden, wodurch sie ganz einfach programmiert werden können.

In diesem Schritt erstellen Sie einen benutzerdefinierten Leistungsindikator, der aufzeichnet, wie oft pro Sekunde auf die neue **GetTasks**-Webdienstmethode zugegriffen wird.

- Klicken Sie im Menü **Ansicht** auf **Server-Explorer**. (Oder drücken Sie **STRG+ALT+S**.)
- Erweitern Sie die Server-Struktur, und suchen Sie den Eintrag **Leistungsindikatoren** (Server » *Computername* » Leistungsindikatoren).
- Klicken Sie mit der rechten Maustaste auf **Leistungsindikatoren**, und klicken Sie anschließend auf **Neue Kategorie erstellen**.
- Geben Sie im Feld **Kategorienname** den Namen **TaskManagement** ein.
- Klicken Sie auf die Schaltfläche **Neu**.
- Ändern Sie den Eintrag unter **Typ** von **NumberOfItems32** in **RateOfCountsPerSecond32**. In diesem Beispiel wird der Standardname **CounterName** übernommen. Klicken Sie auf **OK**.

Leistungsindikatoren-Generator

Definieren Sie eine neue Kategorie oder eine Liste von Zählern, die die Kategorie enthalten soll, oder bearbeiten Sie die Werte einer vorhandenen Kategorie.

OK
Abbrechen
Hilfe

Kategorienname:
TaskManagement

Kategoriebeschreibung:
Keine

Zählerlisten-Generator

Um einen neuen Zähler hinzuzufügen, klicken Sie auf die Schaltfläche "Neu", und geben Sie dann die Details an.

Zähler

Name:
CounterName

Typ:
RateOfCountsPerSecond32

Zählerbeschreibung:
Keine

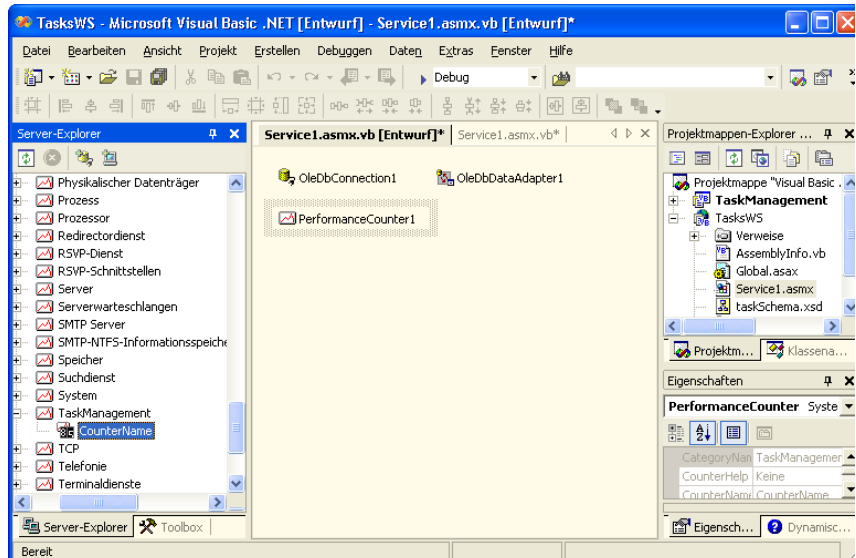
Zähler:
CounterName

Neu
Löschen

◀ **Schritt 6 von 14** Den Leistungsindikator verwenden ▶

Als Nächstes wird der Leistungsindikator aktualisiert, wenn die **GetTasks**-Methode aufgerufen wird. Zur Aktualisierung des Leistungsindikators wird nur eine Codezeile benötigt. Vor Visual Studio .NET waren für derart einfache Schritte unzählige Codezeilen notwendig.

- Doppelklicken Sie im **Projektmappen-Explorer** auf **Service1.asmx**.
- Suchen Sie im **Server-Explorer** den neuen Leistungsindikator **TaskManagement**. Erweitern Sie die Struktur (Server » Computername » Leistungsindikatoren » TaskManagement » CounterName).
- Ziehen Sie das Symbol **CounterName** in die Entwurfsoberfläche.
- Drücken Sie **F4**, um die Eigenschaften des **PerformanceCounter1**-Objekts anzuzeigen. Doppelklicken Sie auf die **ReadOnly**-Eigenschaft, um sie in **False** zu ändern.

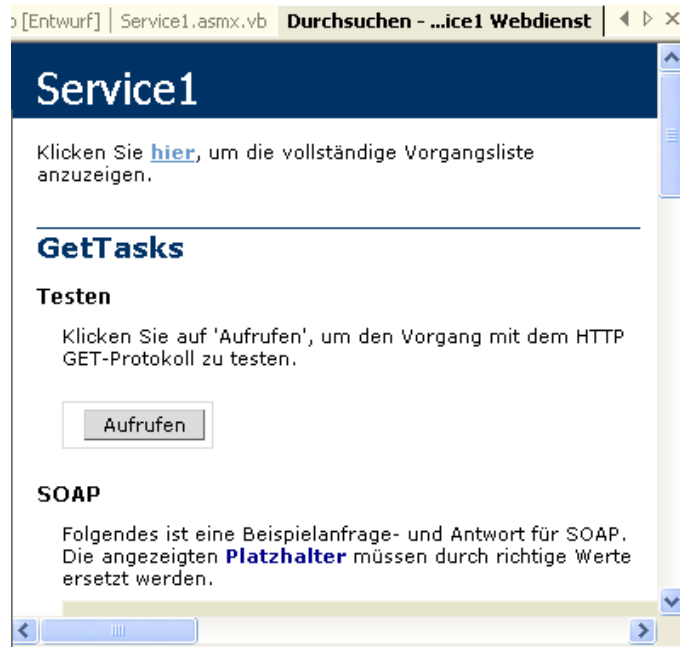


Sie können XML-Webdienste mit Hilfe der Visual Studio.NET-Umgebung testen und debuggen. In diesem Abschnitt wird die **GetTasks**-Dienstmethode getestet.

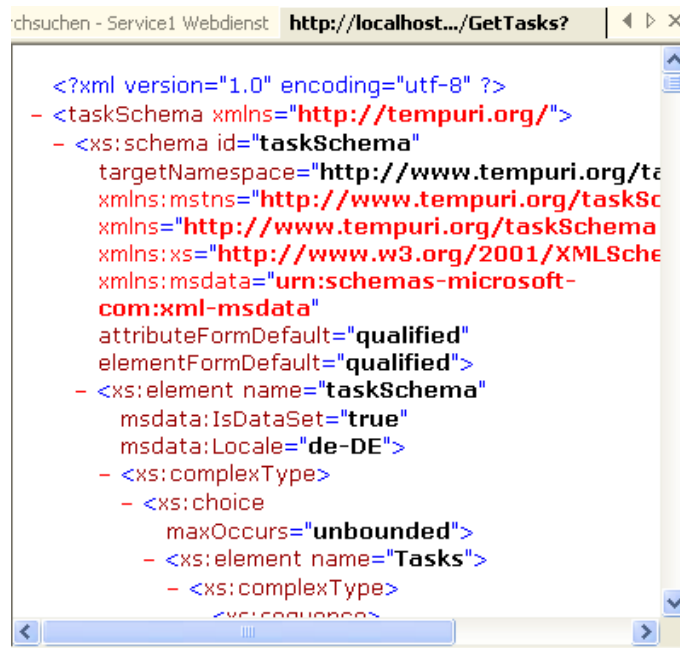
- a. Wählen Sie im **Projektmappen-Explorer** den Eintrag **Service1.asmx**. Klicken Sie im Menü **Datei** auf **Erstellen und durchsuchen**.



- **Hinweis**
 - Beim Öffnen einer .asmx-Datei in einem Browser wird eine Service-Hilfeseite ausgegeben. Auf dieser Seite können Sie eine detaillierte Beschreibung des Dienstes anzeigen und Servermethoden aufrufen.
- b. Klicken Sie auf die Verknüpfung **GetTasks**, um die nachfolgende Seite aufzurufen.



- c. Klicken Sie dort auf die Schaltfläche **Aufrufen**. Der erzeugte XML-Code wird im Browser angezeigt. Führen Sie auf der Seite einen Bildlauf nach unten durch, und zeigen Sie die Informationen zum Schema und zum Tabelleninhalt an.

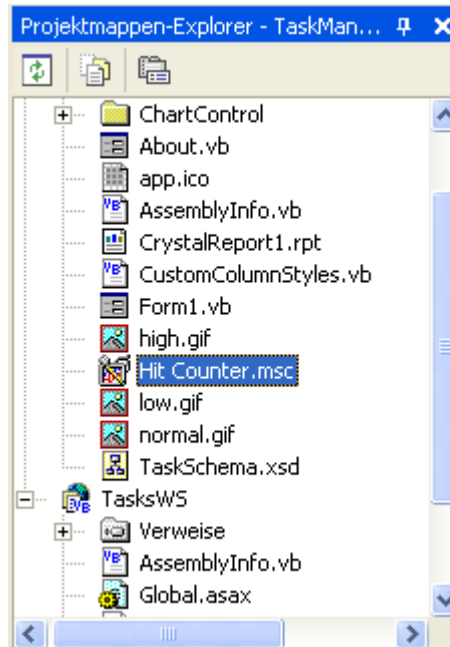



- d. Schließen Sie die Seite mit dem Ergebniscode noch nicht, da sie im nächsten Schritt wieder verwendet wird.

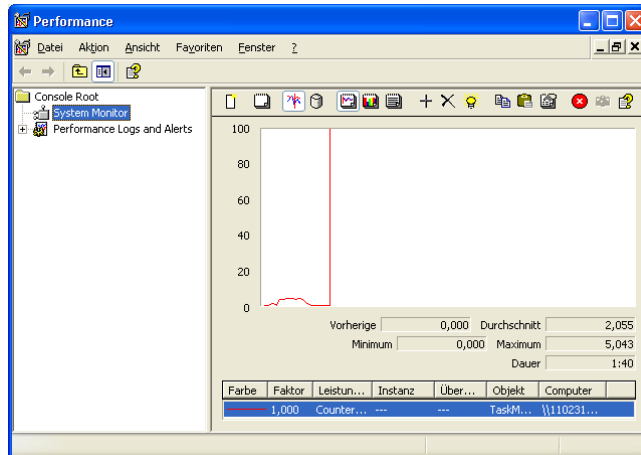
◀ **Schritt 8 von 14** Den Leistungsindikator testen ▶

Wir werden nun den Leistungsindikator in Aktion zeigen, indem wir beobachten, wie der Indikator hochzählt, wenn wir den XML-Webdienst verwenden.

- a. Doppelklicken Sie im **Projektmappen-Explorer** unter dem TaskManagement-Projekt auf **Hit Counter.msc**. Damit wird die Systemmonitor-Anwendung geöffnet.



- b. Kehren Sie zu Visual Studio .NET zurück, und klicken Sie mehrmals auf das Symbol **Aktualisieren**  (in der Visual Studio Web-Symboleiste), um den Webdienst erneut abzufragen.
- c. Wechseln Sie in die Systemmonitor-Anwendung. Sie sollten nun sehen können, wie sich der Leistungsindikator auf und ab bewegt und dadurch anzeigt, wie viele Abfragen diese Webdienstmethode pro Sekunde bearbeitet.



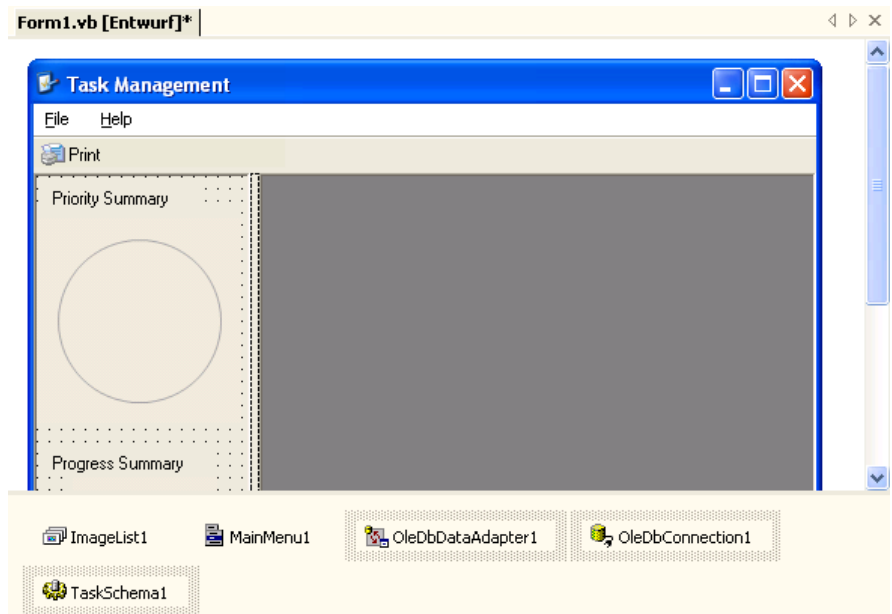
- d. Schließen Sie die Systemmonitor-Anwendung. Wenn Sie gefragt werden, ob die vorgenommenen Änderungen gespeichert werden sollen, klicken Sie auf **Nein**.
- e. Klicken Sie in Visual Studio im Menü **Fenster** auf **Alle Dokumente schließen**. Wenn Sie gefragt werden, ob die vorgenommenen Änderungen gespeichert werden sollen, klicken Sie auf **Ja**.

Teil 6: Erstellen und Verwenden eines XML-Webdienstes

◀ Schritt 9 von 14 Den XML-Webdienst nutzen ▶

Die Verwendung der XML-Webdienste ist genau so einfach wie das Erstellen. In den folgenden Schritten werden die lokalen Datenzugriffsobjekte durch Aufrufe des XML-Webdienstes ersetzt. Zuerst werden die lokalen Datenzugriffsobjekte aus der TaskManagement-Anwendung gelöscht.

- Doppelklicken Sie im **Projektmappen-Explorer** auf **Form1.vb**, das sich im **TaskManagement**-Projekt befindet.
- Löschen Sie die Objekte **OleDbDataAdapter1**, **OleDbConnection1** und **TaskSchema1** aus dem Komponentenfach.

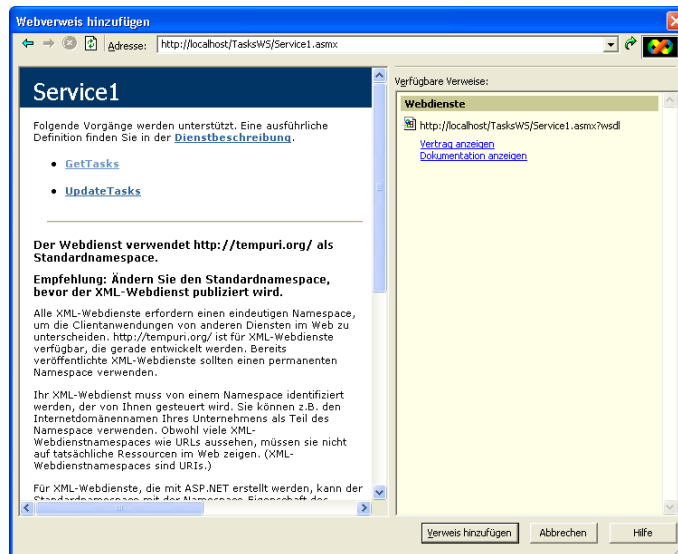


Teil 6: Erstellen und Verwenden eines XML-Webdienstes

◀ Schritt 10 von 14 Einen Webverweis hinzufügen ▶

In diesem Schritt wird dem XML-Webdienst ein Verweis hinzugefügt. Visual Studio downloadet die Dienstbeschreibung und erzeugt eine Proxy-Klasse. Dadurch können Sie den Remoteserver so einfach wie jede beliebige Klasse verwenden.

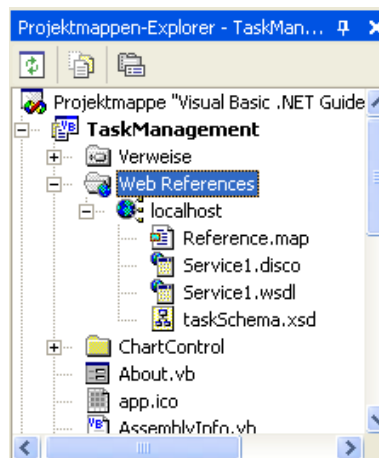
- Wählen Sie im **Projektmappen-Explorer** das Projekt **TaskManagement**. Klicken Sie im Menü **Projekt** auf **Webverweis hinzufügen**.
- Geben Sie als Adresse **http://localhost/TaskWS/Service1.asmx** ein, und drücken Sie die **Eingabetaste**.



- Klicken Sie auf die Schaltfläche **Verweis hinzufügen**.

• Hinweis

- Visual Studio erstellt im Ordner **Web References** unterschiedliche Dateien, die Ihnen die Interaktion mit dem Remotedienst ermöglichen.

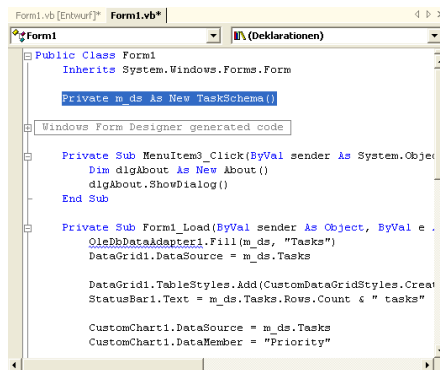


Teil 6: Erstellen und Verwenden eines XML-Webdienstes

◀ Schritt 11 von 14 Webdienstobjekte deklarieren ▶

In den nächsten drei Schritten werden die Aufrufe des alten **DataAdapter**-Objekts durch die Aufrufe des neuen XML-Webdienstes ersetzt. Dazu deklarieren wir zuerst den Dienst und die Datenschemaobjekte.

- Doppelklicken Sie im **Projektmappen-Explorer** auf **Form1.vb**, um Form1 zu öffnen.
- Klicken Sie im Menü **Ansicht** auf **Code**. (Oder drücken Sie **F7**.)
- Markieren Sie im oberen Teil der Datei die Variablendeklaration **m_ds** entsprechend der unteren Abbildung.



```
Public Class Form1
    Inherits System.Windows.Forms.Form

    Private m_ds As New TaskSchema()

    Windows Form Designer generated code

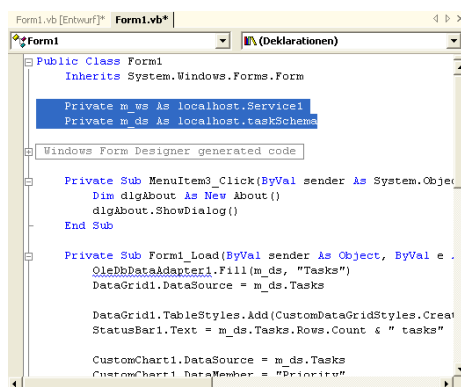
    Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e As EventArgs)
        Dim dlgAbout As New About()
        dlgAbout.ShowDialog()
    End Sub

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs)
        OleDbDataAdapter1.Fill(m_ds, "Tasks")
        DataGridView1.DataSource = m_ds.Tasks

        DataGridView1.TableStyles.Add(CustomDataGridViewStyles.Create)
        StatusBar1.Text = m_ds.Tasks.Rows.Count & " tasks"

        CustomChart1.DataSource = m_ds.Tasks
        CustomChart1.DataMember = "Priority"
    End Sub
End Class
```

- Klicken Sie im Menü **Ansicht** auf **Toolbox**. (Oder drücken Sie **STRG+ALT+X**.)
- Klicken Sie auf die Registerkarte **Visual Basic .NET Guided Tour**, und doppelklicken Sie dann auf **6.11 Declare Web Service Objects**, um den Code einzufügen. Der betreffende Teil des Codes sollte dann wie folgt dargestellt werden.



```
Public Class Form1
    Inherits System.Windows.Forms.Form

    Private m_ws As localhost.Service1
    Private m_ds As localhost.taskSchema

    Windows Form Designer generated code

    Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e As EventArgs)
        Dim dlgAbout As New About()
        dlgAbout.ShowDialog()
    End Sub

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs)
        OleDbDataAdapter1.Fill(m_ds, "Tasks")
        DataGridView1.DataSource = m_ds.Tasks

        DataGridView1.TableStyles.Add(CustomDataGridViewStyles.Create)
        StatusBar1.Text = m_ds.Tasks.Rows.Count & " tasks"

        CustomChart1.DataSource = m_ds.Tasks
        CustomChart1.DataMember = "Priority"
    End Sub
End Class
```

- **Hinweis**
 - Die blauen Markierungen weisen darauf hin, dass OleDbDataAdapter nicht mehr verfügbar ist. Daher ersetzen wir diesen Code im nächsten Schritt.

Als Nächstes wird der Code geändert, der die Aufgabenliste aus der Datenbank abrufen. Anstelle des lokalen **DataAdapter**-Objekts wird die XML-Webdienstmethode **GetTasks** aufgerufen.

- Führen Sie einen Bildlauf bis zur Sub-Prozedur **Form1_Load** durch.
- Markieren Sie wie in der folgenden Abbildung die erste Zeile, in der **OleDbDataAdapter1.Fill** aufgerufen wird.

```

Form1.vb [Entwurf]* Form1.vb*
Form1
Public Class Form1
    Inherits System.Windows.Forms.Form

    Private m_ws As localhost.Service1
    Private m_ds As localhost.taskSchema

    Windows Form Designer generated code

    Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem3.Click
        Dim dlgAbout As New About()
        dlgAbout.ShowDialog()
    End Sub

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs) Handles MyBase.Load
        OleDbDataAdapter1.Fill(m_ds, "Tasks")
        DataGridView1.DataSource = m_ds.Tasks

        DataGridView1.TableStyles.Add(CustomDataGridViewStyles.Create)
        StatusBar1.Text = m_ds.Tasks.Rows.Count & " tasks"

        CustomChart1.DataSource = m_ds.Tasks
        CustomChart1.DataMember = "Priority"
    End Sub
End Class
    
```

- Klicken Sie unter **Toolbox** auf die Registerkarte **Visual Basic .NET Guided Tour**, und doppelklicken Sie dann auf **6.12 Call GetTasks Service Method**, um den Code einzufügen. Der betreffende Teil des Codes sollte dann wie folgt aussehen.

```

Form1.vb [Entwurf]* Form1.vb*
Form1
Windows Form Designer generated code

    Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem3.Click
        Dim dlgAbout As New About()
        dlgAbout.ShowDialog()
    End Sub

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs) Handles MyBase.Load
        Cursor = Cursors.WaitCursor
        m_ws = New localhost.Service1()
        m_ds = m_ws.GetTasks()
        Cursor = Cursors.Default

        DataGridView1.DataSource = m_ds.Tasks

        DataGridView1.TableStyles.Add(CustomDataGridViewStyles.Create)
        StatusBar1.Text = m_ds.Tasks.Rows.Count & " tasks"

        CustomChart1.DataSource = m_ds.Tasks
        CustomChart1.DataMember = "Priority"

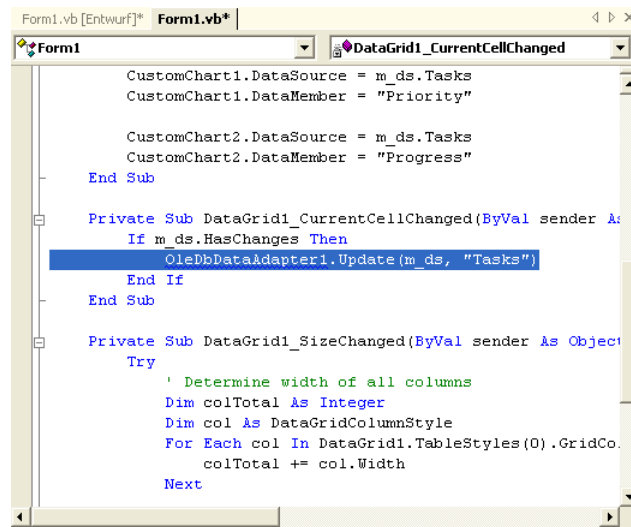
        CustomChart2.DataSource = m_ds.Tasks
    End Sub
End Class
    
```

Teil 6: Erstellen und Verwenden eines XML-Webdienstes

◀ Schritt 13 von 14 UpdateTasks-Dienstmethode aufrufen ▶

Im letzten Schritt wird anstelle der lokalen **DataAdapter**-Aktualisierungsmethode die **UpdateTasks**-Dienstmethode eingesetzt.

- Führen Sie einen Bildlauf zur Sub-Prozedur **DataGrid1_CurrentCellChanged** durch.
- Markieren Sie wie in der folgenden Abbildung den Aufruf **OleDbDataAdapter1.Update**.



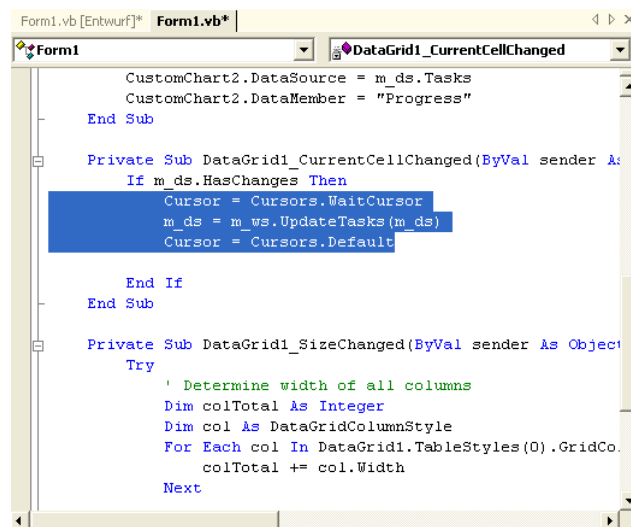
```
Form1.vb [Entwurf]* Form1.vb*
Form1
CustomChart1.DataSource = m_ds.Tasks
CustomChart1.DataMember = "Priority"

CustomChart2.DataSource = m_ds.Tasks
CustomChart2.DataMember = "Progress"
End Sub

Private Sub DataGrid1_CurrentCellChanged(ByVal sender As Object, ByVal e As EventArgs)
    If m_ds.HasChanges Then
        OleDbDataAdapter1.Update(m_ds, "Tasks")
    End If
End Sub

Private Sub DataGrid1_SizeChanged(ByVal sender As Object, ByVal e As EventArgs)
    Try
        ' Determine width of all columns
        Dim colTotal As Integer
        Dim col As DataGridViewColumnStyle
        For Each col In DataGridView1.TableStyles(0).GridColumnStyles
            colTotal += col.Width
        Next
    End Try
End Sub
```

- Klicken Sie unter **Toolbox** auf die Registerkarte **Visual Basic .NET Guided Tour**, und doppelklicken Sie dann auf **6.13 Call UpdateTasks Service Method**, um den Code einzufügen. Der betreffende Teil des Codes sollte dann wie folgt aussehen.



```
Form1.vb [Entwurf]* Form1.vb*
Form1
CustomChart2.DataSource = m_ds.Tasks
CustomChart2.DataMember = "Progress"
End Sub


Private Sub DataGrid1_CurrentCellChanged(ByVal sender As Object, ByVal e As EventArgs)
    If m_ds.HasChanges Then
        Cursor = Cursors.WaitCursor
        m_ds = m_ws.UpdateTasks(m_ds)
        Cursor = Cursors.Default
    End If
End Sub

Private Sub DataGrid1_SizeChanged(ByVal sender As Object, ByVal e As EventArgs)
    Try
        ' Determine width of all columns
        Dim colTotal As Integer
        Dim col As DataGridViewColumnStyle
        For Each col In DataGridView1.TableStyles(0).GridColumnStyles
            colTotal += col.Width
        Next
    End Try
End Sub
```

Teil 6: Erstellen und Verwenden eines XML-Webdienstes

◀ Schritt 14 von 14 Erstellen und Ausführen ▶

Erstellen Sie die Anwendung, und führen Sie sie aus. Die Anwendung verwendet den XML-Webdienst zum Abrufen und Aktualisieren von Aufgaben.

- a. Klicken Sie im Menü **Debuggen** auf **Starten ohne Debuggen**, um die Anwendung auszuführen. (Oder drücken Sie **STRG+F5**.)
- b. Die Anwendung sieht genau so aus, der Unterschied liegt aber darin, dass das DataGrid über das Web mit Daten gefüllt wird. Ändern Sie den **Status** einer Aufgabe von 50 % zu 100 %, und drücken Sie die **Eingabetaste**. Damit wird ein weiterer Aufruf des XML-Webdienstes ausgelöst, und die zu Grunde liegende Datenbank wird aktualisiert.
- c. Schließen Sie das Formular über die Schaltfläche  in der oberen rechten Ecke des Fensters. (Oder drücken Sie **ALT+F4**.)

TEIL 7: VERWENDEN DER UNBEAUFICHTIGTEN BEREITSTELLUNG

Verwenden der unbeaufsichtigten Bereitstellung

Übersicht über Teil 7 ▶

Visual Basic .NET bietet eine Möglichkeit zur Installation und Ausführung der Windows Form-Anwendungen. Dazu müssen Sie einfach nur einen URL in einem Webbrowser eingeben. In der Task Management-Anwendung werden Benutzer von der unbeaufsichtigten Bereitstellungsfunktion Gebrauch machen, um die Anwendung aus dem Intranet oder Extranet auszuführen. Dadurch führen Benutzer stets die aktuellste Version der Anwendung aus. Sie brauchen dafür nur die aktualisierte Datei TaskManagement.exe auf der Website ablegen.

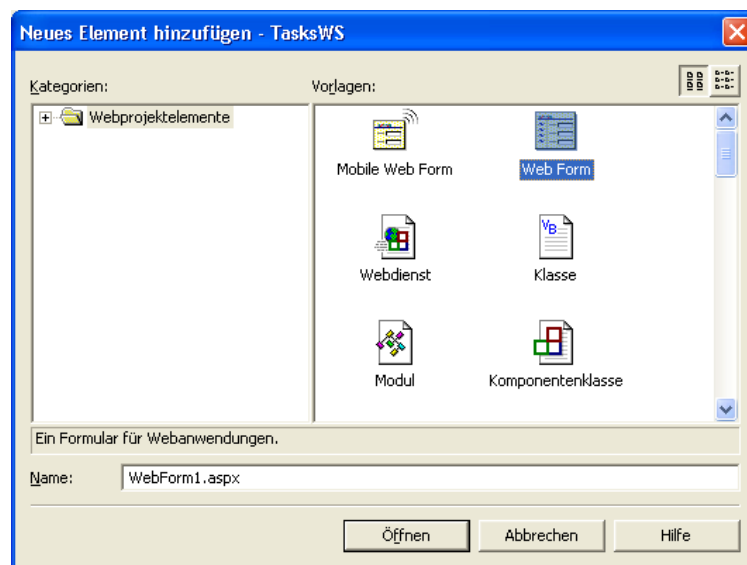
Im letzten Teil der Guided Tour erstellen Sie ein ASP.NET Web Form, positionieren darin einen Hyperlink zu der Task Management-Anwendung und führen dann die Anwendung in Ihrem Internetbrowser aus.

Teil 7: Verwenden der unbeaufsichtigten Bereitstellung

◀ Schritt 1 von 5 Web Form erstellen ▶

In diesem Schritt wird ein Web Form in das **TasksWS**-Webdienstprojekt eingefügt.

- Klicken Sie im Menü **Ansicht** auf **Projektmappen-Explorer**.
(Oder drücken Sie **STRG+R**.)
- Klicken Sie auf das Webdienstprojekt **TasksWS**.
- Klicken Sie im Menü **Projekt** auf **Web Form hinzufügen**.
- Daraufhin wird das Fenster **Neues Element hinzufügen** angezeigt.
Den Namen **WebForm1.aspx** können Sie übernehmen.
- Klicken Sie auf **Öffnen**, um das neue Web Form in das Projekt einzufügen.
Den Namen **WebForm1.aspx** können Sie übernehmen.

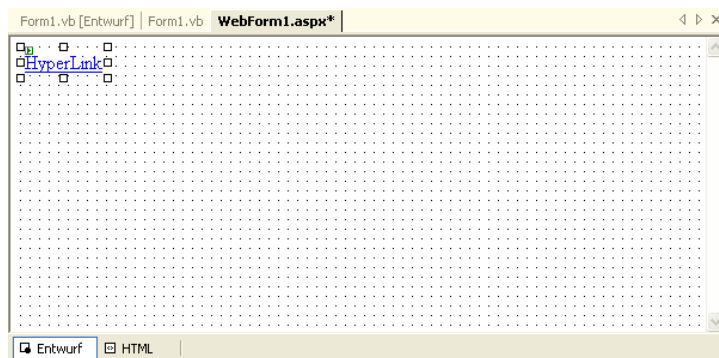


Teil 7: Verwenden der unbeaufsichtigten Bereitstellung

◀ Schritt 2 von 5 Einen Hyperlink in die Anwendung einfügen ▶

In diesem Schritt wird ein Hyperlinksteuerelement in das Formular eingefügt, das auf die Windows Form-Anwendung TaskManagement.exe in Visual Basic .NET verweist.

- Klicken Sie im Menü **Ansicht** auf **Toolbox**. (Oder drücken Sie **STRG+ALT+X**.)
- Klicken Sie in der Toolbox auf die Registerkarte **Web Forms**.
- Doppelklicken Sie auf das Symbol **HyperLink**, um das Steuerelement in das Web Form zu übernehmen.



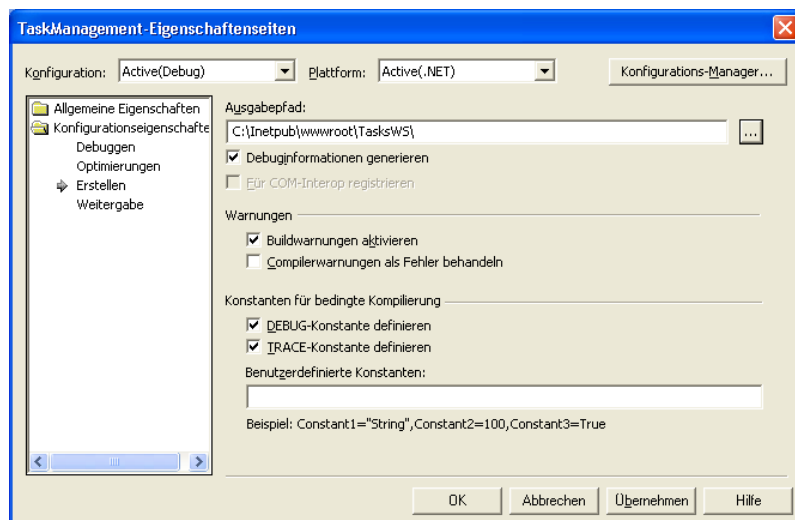
- Drücken Sie **F4**, um die Eigenschaften des Hyperlinks anzuzeigen.
- Ändern Sie die Eigenschaft **Text** in **Task Management Application**.
- Ändern Sie die Eigenschaft **NavigateURL** in **TaskManagement.exe** (ohne Leerzeichen).

Teil 7: Verwenden der unbeaufsichtigten Bereitstellung

◀ Schritt 3 von 5 Build-Position ändern ▶

Nachdem im letzten Schritt ein Hyperlink in der Anwendung erstellt wurde, muss jetzt der richtige Speicherort der Task Management-Programmdateien gewährleistet werden. Dazu werden die Build-Eigenschaften des TaskManagement-Projekts geändert.

- Klicken Sie im Menü **Ansicht** auf **Projektmappen-Explorer**. (Oder drücken Sie **STRG+R**.)
- Klicken Sie auf das Projekt **TaskManagement**.
- Wählen Sie aus dem Menü **Projekt** die Option **Eigenschaften**.
- Klicken Sie dann auf **Konfigurationseigenschaften**.
- Klicken Sie auf **Erstellen**.
- Klicken Sie anschließend auf **...**, um unter **Ausgabepfad** den Ordner festzulegen, der das soeben erstellte Web Form enthält. (Der Pfad sollte **c:\inetpub\wwwroot\TasksWS** lauten.)
- Klicken Sie auf **OK**, um das Fenster **TaskManagement-Eigenschaftenseiten** zu schließen.
- Klicken Sie im Menü **Erstellen** auf **Projektmappe erstellen**. (Oder drücken Sie **STRG+UMSCHALT+B**.)

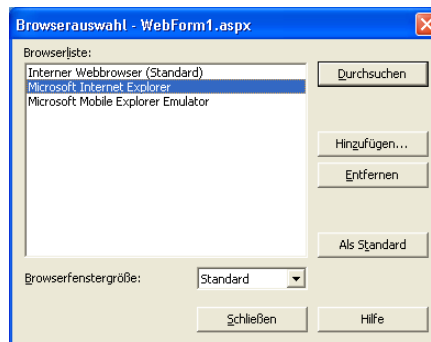


Teil 7: Verwenden der unbeaufsichtigten Bereitstellung

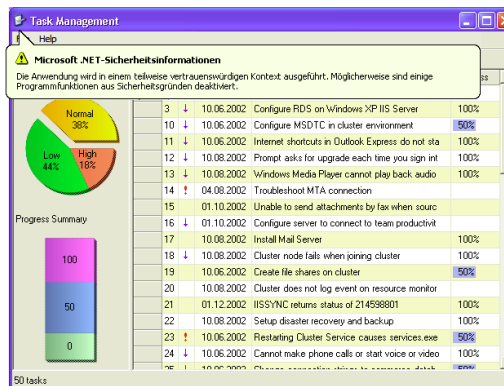
◀ Schritt 4 von 5 Die Anwendung aus einem Browser ausführen ▶

Die unbeaufsichtigte Bereitstellung ist ganz einfach zu verwenden. Navigieren Sie zu dem neuen Web Form, und klicken Sie auf den Hyperlink, um die Task Management-Anwendung auszuführen.

- Klicken Sie im Menü **Ansicht** auf **Projektmappen-Explorer**. (Oder drücken Sie **STRG+R**.)
- Klicken Sie im Projekt **TasksWS** auf die Datei **WebForm1.aspx**.
- Wählen Sie dann aus dem Menü **Datei** die Option **Browserauswahl**.



- Wählen Sie aus der Browserliste **Microsoft Internet Explorer**, und klicken Sie dann auf **Durchsuchen**.
- Klicken Sie auf die Verknüpfung **Task Management Application**, um die Anwendung zu laden.



• Hinweis

- Die unbeaufsichtigte Bereitstellung ist nicht nur einfach zu verwenden, sondern auch sicher. Standardmäßig dürfen in der Task Management-Anwendung keine Aktionen ausgeführt werden, die für den Computer des Benutzers ein Risiko darstellen. So eine Aktion ist z. B. das Ändern eines Aufgabenstatus. Dieser Vorgang ist nicht zulässig. Die Anwendung muss vollständig vertrauenswürdig sein, bevor Standardfunktionen durch benutzerdefinierte Funktionen ersetzt werden können. Benutzer können im Ergebnis stets die aktuellste Version der Anwendung ausführen und haben dabei immer die volle Kontrolle darüber, welche Vorgänge die Anwendung auf ihrem System ausführen kann und welche nicht.

Teil 7: Verwenden der unbeaufsichtigten Bereitstellung

◀ Schritt 5 von 5 Zusammenfassung ▶

Die Visual Basic .NET Guided Tour umfasst drei Schlüsselgebiete der Visual Basic .NET-Entwickleraufgaben: Entwicklung von Windows Forms, Erstellen und Verwenden von XML-Webdiensten und die unbeaufsichtigte Bereitstellung.

In der Guided Tour haben Sie die Verankerungs- und Andockfunktionen verwendet, damit die Größe der Anwendung automatisch ohne benutzerdefinierten Code angepasst werden kann. Danach haben Sie aus einer lokalen Access-Datenbank mit Hilfe nur weniger Codezeilen Aufgabeninformationen abgefragt und aktualisiert. Als Nächstes haben Sie das Erscheinungsbild der Anwendung mittels benutzerdefinierter Steuerelemente und die Spalten des DataGrids mit Hilfe der neuen Vererbungsfunktion von Visual Basic .NET angepasst. Sie haben dann einen XML-Webdienst erstellt, um unabhängig vom jeweiligen Speicherort stets auf Ihre Daten zugreifen zu können. Letztlich haben Sie erfahren, wie einfach die neueste Version der Anwendung mit der unbeaufsichtigten Bereitstellung an die Benutzer verteilt werden kann.

Alle diese Schlüsselgebiete und andere Erweiterungen unterstützen Entwickler bei den dringendsten Aufgaben, die sie täglich lösen müssen. Visual Basic .NET ermöglicht Ihnen somit, Software so optimal wie möglich zu entwickeln.

TEIL 8: VERWENDEN VON CRYSTAL REPORTS (OPTIONAL)

Verwenden von Crystal Reports (Optional)

Übersicht über Teil 8 ▶

Dieser Teil ist optional und kann mit der unbeaufsichtigten Bereitstellungsfunktion verwendet werden, wenn Sie die Sicherheitseinstellungen so ändern, dass die Programmdatei TaskManagement.exe auf Ihrem Computer vollständig vertrauenswürdig ist. Standardmäßig schützt .NET den Benutzer, indem Steuerelemente von Drittanbietern bei der unbeaufsichtigten Bereitstellung einer Anwendung nur dann zugelassen werden, wenn sie für den Benutzer vertrauenswürdig sind.

Visual Studio .NET ist eine Entwicklungsumgebung, die nahezu unbegrenzt erweitert werden kann. Mehr als 100 unabhängige Softwareanbieter (ISV; Independent Software Vendors) stellen neue Versionen ihrer Tools und Zusatzprogramme für Visual Studio .NET zur Verfügung. Zu diesen Zusatzprogrammen gehört auch eine spezielle Version von Crystal Reports, die direkt in die IDE von Visual Studio .NET integriert ist.

Crystal Reports spielt eine wichtige Rolle im Anwendungsentwicklungsprozess, da es Entwicklern eine produktive, integrierte RAD-Umgebung zur Erstellung interaktiver relationaler Datenberichte bietet. Diese Berichte können für alle .NET-Anwendungstypen erstellt werden, einschließlich XML-Webdienste, webbasierte Lösungen und Windows-basierte Anwendungen, die auf Desktopcomputern ebenso wie auf mobilen Geräten ausgeführt werden.

In diesem Schritt wird die Anwendung um neue Berichtsfunktionen erweitert. Mit dem neuen Bericht können Benutzer ihre Aufgaben in Adobe Acrobat sowie im Word- und Excel-Format anzeigen und ausgeben sowie in diese Formate exportieren.

Teil 8: Verwenden von Crystal Reports (Optional)

◀ Schritt 1 von 4 Das neue Windows Form hinzufügen ▶

Dem Projekt wird in diesem Schritt ein zweites Windows Formular hinzugefügt, das den Crystal Report enthält, der im nächsten Schritt erstellt wird.

- Klicken Sie im **Projektmappen-Explorer** auf das Projekt **TaskManagement**. Klicken Sie im Menü **Projekt** auf **Windows Form hinzufügen...**
- Das Fenster mit den Vorlagensymbolen wird angezeigt, in dem das Windows Form-Symbol markiert ist. Klicken Sie auf **Öffnen** (der Standardname **Form2.vb** kann übernommen werden).

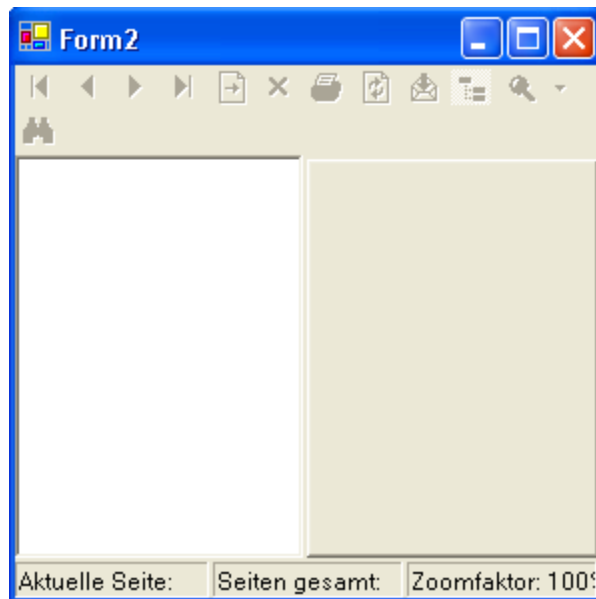


Teil 8: Verwenden von Crystal Reports (Optional)

◀ **Schritt 2 von 4** Ein Report Viewer-Steuerelement in das Formular einfügen ▶

Jetzt können wir das Crystal Reports Viewer-Steuerelement verwenden.

- Klicken Sie im Menü **Ansicht** auf **Toolbox**. (Oder drücken Sie **STRG+ALT+X**.)
- Klicken Sie in der Toolbox auf die Registerkarte **Windows Forms**.
- Doppelklicken Sie auf das Steuerelement **CrystalReportViewer**, um es in die Entwurfsoberfläche des Formulars einzufügen.
- Drücken Sie auf **F4**, um die Eigenschaften des Steuerelements anzuzeigen. Klicken Sie auf die **Dock**-Eigenschaft, und drücken Sie **F**, um als Wert **Fill** anzugeben.

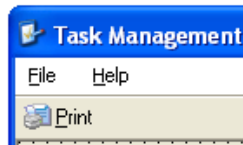


Teil 8: Verwenden von Crystal Reports (Optional)

◀ Schritt 3 von 4 Das Berichtsformular anzeigen ▶

In diesem Schritt wird das neue Formular mit dem Crystal Report Viewer-Steuerelement angezeigt.

- Doppelklicken Sie im **Projektmappen-Explorer** auf **Form1.vb**.
- Doppelklicken Sie auf das Steuerelement für Symbolleisten (enthält die Schaltfläche **Print**). Damit wird die Codedatei geöffnet, in der für das **ToolBar1**-Steuerelement das **ButtonClick**-Ereignis verarbeitet wird.



- Klicken Sie auf die Registerkarte **Visual Basic .NET Guided Tour**, und doppelklicken Sie dann auf **8.3 Display Report Form**, um den Code einzufügen, der das Berichtsformular aufruft.

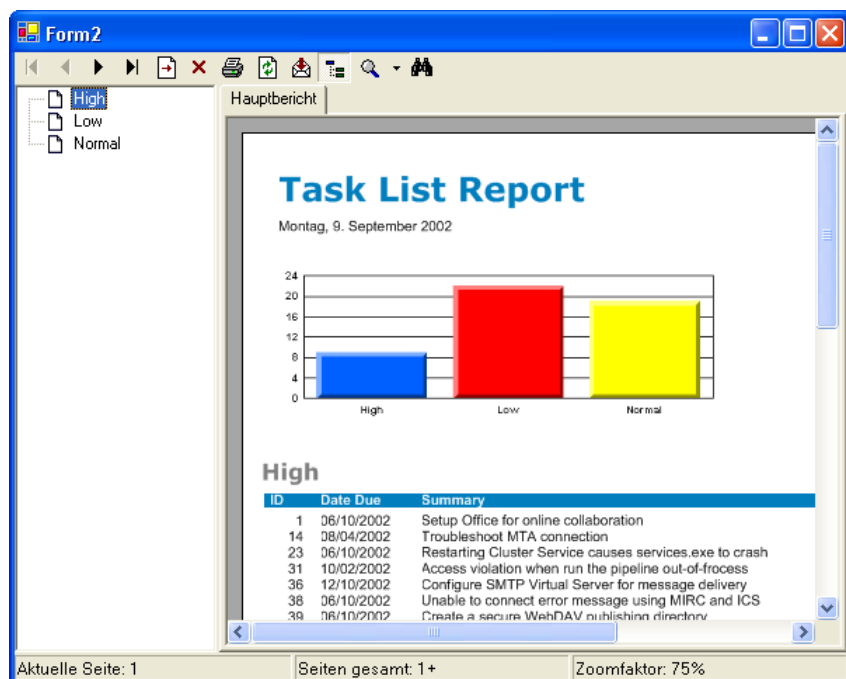
```
Private Sub ToolBar1_ButtonClick(ByVal sender As Object, ByVal e As EventArgs)
    Select Case (ToolBar1.Buttons.IndexOf(e.Button))
        Case 0
            Dim form As Form2 = New Form2()
            Dim report As CrystalReport1 = New CrystalReport1()
            report.SetDataSource(m_ds.Tasks)
            form.CrystalReportViewer1.ReportSource = report
            form.Show()
    End Select
End Sub
```



Teil 8: Verwenden von Crystal Reports (Optional)

◀ Schritt 4 von 4 Den Bericht erstellen und anzeigen ▶

Die Leistungsfähigkeit von Crystal Reports lässt sich ganz einfach in Ihre .NET-Anwendungen integrieren. Lassen Sie uns einen Blick auf den Bericht werfen, indem wir die Anwendung erstellen und ausführen.

- Klicken Sie im Menü **Debuggen** auf **Starten ohne Debuggen**, um die Anwendung auszuführen. (Oder drücken Sie **STRG+F5**.)
- Klicken Sie in der Symbolleiste auf die Schaltfläche **Drucken**, um das Berichtsformular anzuzeigen.
- Passen Sie die Größe des Formularfensters so an, dass Sie den Bericht besser überblicken können. Beachten Sie, wie sich die Größe des Report Viewer-Steuerelements ändert, um sich an die Größe des Formulars anzupassen.



- Klicken Sie auf die Schaltfläche  zum Exportieren des Berichts. Sie können den Bericht in Adobe Acrobat, Excel, Word und als RTF-Dokument exportieren.
- Klicken Sie auf die Schaltfläche **Abbrechen**, um das Exportdialogfeld zu schließen.
- Schließen Sie das Formular über die Schaltfläche  in der oberen rechten Ecke des Fensters. (Oder drücken Sie **ALT+F4**.)

- **Hinweis**

- Sie können den Bericht auch drucken, durchsuchen und vergrößern.

TEIL 9: VERWENDEN VON ACTIVEX-KOMPONENTEN (OPTIONAL)

Verwenden von ActiveX-Komponenten (optional)

Übersicht über Teil 9 ▶

Dieser Teil ist optional und erfordert Microsoft Excel 2000 oder 2002.

Die Verwendung von ActiveX-Komponenten in Visual Basic .NET gestaltet sich unkompliziert. Jede beliebige ActiveX-Komponente kann durch Hinzufügen eines entsprechenden Verweises in Ihrer .NET-Anwendung verwendet werden.

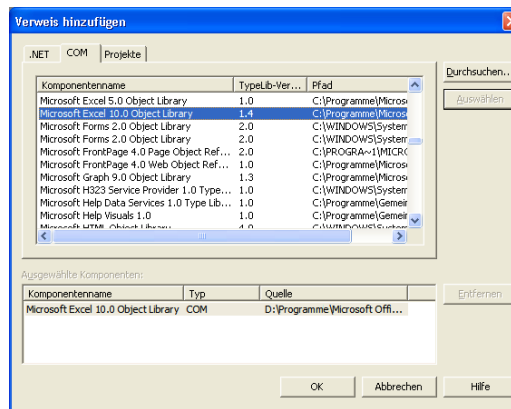
In diesem Teil wird die Aufgabenliste mit Hilfe der Microsoft Excel ActiveX-Komponente in eine Excel-Tabelle exportiert.

Teil 9: Verwenden von ActiveX-Komponenten (Optional)

◀ Schritt 1 von 4 Einen Verweis auf die Komponente hinzufügen ▶

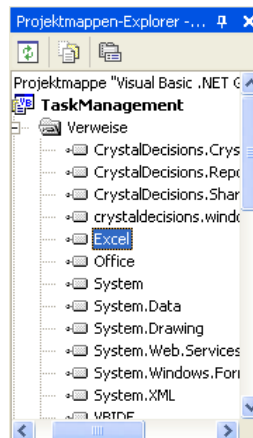
Zu der gewünschten ActiveX-Komponente muss zunächst ein Verweis hinzugefügt werden. In diesem Fall wird ein Verweis auf die Microsoft Excel-Objektbibliothek eingefügt.

- Klicken Sie im Menü **Ansicht** auf **Projektmappen-Explorer**. (Oder drücken Sie **STRG+R**.)
- Klicken Sie auf das Projekt **TaskManagement**.
- Klicken Sie im Menü **Projekt** auf **Verweis hinzufügen**.
- Klicken Sie auf die Registerkarte **COM**.
- Wählen Sie aus der Liste den Eintrag **Microsoft Excel 10.0 Object Library** aus. (Unter Umständen verfügen Sie über eine frühere Version und/oder mehrere Versionen der Objektbibliothek. Wählen Sie die aktuellste Version aus).
- Klicken Sie auf **Auswählen**.
- Klicken Sie auf **OK**.



• Hinweis

- Die Verweise auf die ActiveX-Komponenten von Excel wurden zu dem Projekt hinzugefügt und werden in der Verweisliste im Fenster **Projektmappen-Explorer** angezeigt.

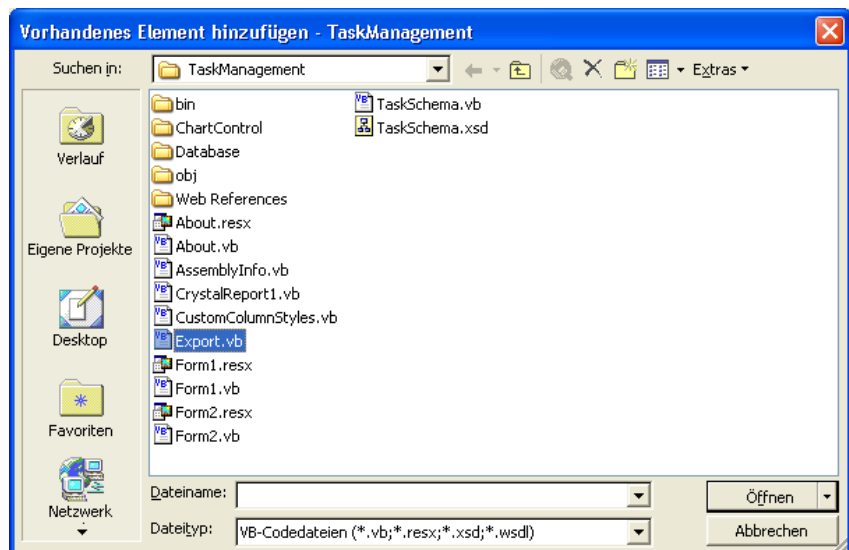


Teil 9: Verwenden von ActiveX-Komponenten (Optional)

◀ Schritt 2 von 4 Exportcode hinzufügen ▶

Als Nächstes müssen wir dem Projekt die Datei hinzufügen, die die benutzerdefinierte Microsoft Excel-Exportklasse enthält.

- Klicken Sie im Menü **Ansicht** auf **Projektmappen-Explorer**.
(Oder drücken Sie **STRG+R**.)
- Klicken Sie im Menü **Projekt** auf **Vorhandenes Element hinzufügen**.
(Oder drücken Sie **STRG+D**.)
- Doppelklicken Sie in der Liste der Dateien auf **Export.vb**.
(Die Datei **Export.vb** befindet sich im Ordner **C:\Programme\Visual Basic .NET Guided Tour\Projects\TaskManagement**).



Teil 9: Verwenden von ActiveX-Komponenten (Optional)

◀ Schritt 3 von 4 Exportfunktion implementieren ▶

In diesem Schritt fügen Sie für das Menüelement **Export** einen Ereignishandler hinzu.


- Klicken Sie im Menü **Ansicht** auf **Projektmappen-Explorer**. (Oder drücken Sie **STRG+R**.)
- Doppelklicken Sie im **Projektmappen-Explorer** auf **Form1.vb**.
- Klicken Sie im Formular **Task Management** im Hauptmenü auf **File**.
- Doppelklicken Sie auf **Export**, um dem Menüelement **Export** einen Ereignishandler hinzuzufügen. Der Cursor sollte dabei im Ereignishandler `MenuItem4_Click` positioniert werden.
- Klicken Sie in der Toolbox auf die Registerkarte **Visual Basic .NET Guided Tour**, und doppelklicken Sie auf **9.3 Export Tasks**, um den Code einzufügen, der zum Exportieren der Aufgabendaten in Microsoft Excel erforderlich ist.

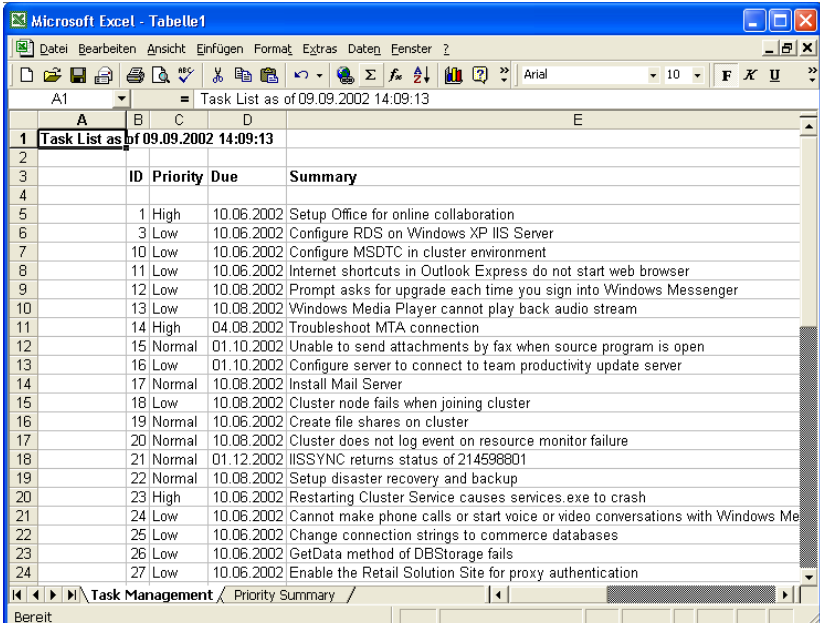
```
Private Sub MenuItem4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    If Not ExportExcel(CType(mdsr.Tasks, System.Data.DataTable)) Then
        MessageBox.Show("Microsoft Excel is not installed on your system. The Export
        MenuItem4.Enabled = False
    End If
End Sub
```

Teil 9: Verwenden von ActiveX-Komponenten (Optional)

◀ Schritt 4 von 4 Anwendung testen ▶

Führen Sie die Task Management-Anwendung aus, und exportieren Sie die Aufgabendaten in Microsoft Excel.

- Klicken Sie im Menü **Debuggen** auf **Starten ohne Debuggen**. (Oder drücken Sie **STRG+F5**.)
- Klicken Sie im Menü **File** auf **Export**.
- Es wurden zwei Arbeitsblätter erstellt: eines für die Aufgabenliste und das zweite für ein Kreisdiagramm.
- Schließen Sie das Formular, indem Sie auf das Symbol  in der Ecke oben rechts klicken oder indem Sie im Menü File auf **Exit** klicken. (Oder drücken Sie **ALT+F4**.)



The screenshot shows a Microsoft Excel window titled "Microsoft Excel - Tabelle1". The active sheet is "Task List as of 09.09.2002 14:09:13". The table contains the following data:

| ID | Priority | Due | Summary |
|----|----------|------------|---|
| 1 | High | 10.06.2002 | Setup Office for online collaboration |
| 3 | Low | 10.06.2002 | Configure RDS on Windows XP IIS Server |
| 10 | Low | 10.06.2002 | Configure MSDTC in cluster environment |
| 11 | Low | 10.06.2002 | Internet shortcuts in Outlook Express do not start web browser |
| 12 | Low | 10.08.2002 | Prompt asks for upgrade each time you sign into Windows Messenger |
| 13 | Low | 10.08.2002 | Windows Media Player cannot play back audio stream |
| 14 | High | 04.08.2002 | Troubleshoot MTA connection |
| 15 | Normal | 01.10.2002 | Unable to send attachments by fax when source program is open |
| 16 | Low | 01.10.2002 | Configure server to connect to team productivity update server |
| 17 | Normal | 10.08.2002 | Install Mail Server |
| 18 | Low | 10.08.2002 | Cluster node fails when joining cluster |
| 19 | Normal | 10.06.2002 | Create file shares on cluster |
| 20 | Normal | 10.08.2002 | Cluster does not log event on resource monitor failure |
| 21 | Normal | 01.12.2002 | IISYNC returns status of 214598801 |
| 22 | Normal | 10.08.2002 | Setup disaster recovery and backup |
| 23 | High | 10.06.2002 | Restarting Cluster Service causes services.exe to crash |
| 24 | Low | 10.06.2002 | Cannot make phone calls or start voice or video conversations with Windows Me |
| 25 | Low | 10.06.2002 | Change connection strings to commerce databases |
| 26 | Low | 10.06.2002 | GetData method of DBStorage fails |
| 27 | Low | 10.06.2002 | Enable the Retail Solution Site for proxy authentication |

1.1 Verwandte Infodateien

Visual Studio .NET enthält neben dieser Infodatei für Visual Studio .NET weitere Infodokumente mit Informationen zu bestimmten Produktbereichen. In der folgenden Liste sind die Namen verwandter Infodokumente sowie deren Speicherort auf den mitgelieferten Medien aufgeführt.

| Inhalt | Dateiname | Speicherort |
|---|-----------------|--|
| Microsoft Visual Studio .NET Windows Component Update | Moreinfo.htm | Auf der CD Visual Studio .NET Windows Component Update im Verzeichnis Help oder auf der DVD im Verzeichnis wcu\Help |
| Infodatei für Setup-Administratormodus von Microsoft Visual Studio .NET | adminreadme.htm | Auf CD 1 oder der DVD für Visual Studio .NET im Verzeichnis Setup |
| .NET Framework SDK | nicht verfügbar | http://go.microsoft.com/fwlink/?LinkId=501 |
| VSIP & VSA SDK | Readme.htm | CD VSIP SDK im Stammverzeichnis |
| VSS 6.0c | Readmess.htm | Auf der CD Visual SourceSafe® 6.0c im Stammverzeichnis |
| Beispiel für die Visual Basic-Version von Duwamish 7.0 | readme.htm | Standardverzeichnis: [Buchstabe des Laufwerks, auf dem Visual Studio .Net installiert ist]: \\Programme\Microsoft Visual Studio .NET\ Enterprise Samples\ Duwamish 7.0 VB |
| Beispiel für die C#-Version von Duwamish 7.0 | readme.htm | Standardverzeichnis: [Buchstabe des Laufwerks, auf dem Visual Studio .Net installiert ist]: \\Programme\Microsoft Visual Studio .NET\ Enterprise Samples\ Duwamish 7.0 CS |
| Beispiel für Fitch and Mather 7.0 | ReadMe.html | Standardverzeichnis: [Buchstabe des Laufwerks, auf dem Visual Studio .Net installiert ist]: \\Programme\Microsoft Visual Studio .NET\Enterprise Samples\FMStocks7 |



| | | |
|-------------------------------|------------|---|
| Application Center Test (ACT) | ReadMe.htm | [Buchstabe des Laufwerks, auf dem Visual Studio .Net installiert ist]:\Programme\Microsoft ACT\ |
| Tools für Studierende | README.HTM | CD Tools für Studierende im Stammverzeichnis |
| Fakultätstools | README.HTM | CD Fakultätstools im Stammverzeichnis |

1.2 Systemanforderungen

Systemanforderungen für die Installation verschiedener Versionen von Visual Studio .NET

| Visual Studio .NET | | | | |
|---|--|----------------------|--------------|----------|
| | Enterprise Architect | Enterprise Developer | Professional | Academic |
| Prozessor | PC mit einem Prozessor der Pentium II-Klasse, 450 MHz (empfohlen wird: Pentium III-Klasse, 600 MHz) | | | |
| RAM¹ | <ul style="list-style-type: none"> Windows NT[®] 4.0 Workstation – 64 MB, Windows NT 4.0 Server – 160 MB (empfohlen wird: 96 MB für Workstation, 192 MB für Server) Windows 2000 Professional – 96 MB; Windows 2000 Server – 192 MB (empfohlen wird: 128 MB für Professional, 256 MB für Server) Windows XP Professional – 160 MB (empfohlen wird: 192 MB) Windows XP Home – 96 MB (empfohlen wird: 160 MB) | | | |
| Verfügbare Festplattenspeicher² | 600 MB auf dem Systemlaufwerk, 3 GB Installationslaufwerk | | | |
| Betriebssystem^{3, 4} | Windows 2000 [®] , Windows XP und Windows NT 4.0 | | | |
| CD-ROM- oder DVD-ROM-Laufwerk⁵ | Erforderlich | | | |
| Grafikkarte | 800 x 600, 256 Farben (High Color 16-Bit empfohlen) | | | |
| Maus | Microsoft-Maus oder kompatibles Zeigegerät | | | |

Anhang 2: Installation

2.1 Installationsvorbereitung für Visual Studio .NET

Die aktuellsten Installationsanweisungen für Visual Studio .NET finden Sie unter <http://support.microsoft.com/support/kb/articles/Q312/7/79.asp>.

- Vor der Installation von Visual Studio .NET sollten Sie sicherstellen, dass die Systemanforderungen erfüllt sind und Sie die unten stehenden Anweisungen für die Installationsvorbereitung von Visual Studio .NET befolgen.
- Wenn Sie Microsoft Office XP Developer verwenden möchten, müssen Sie diese Anwendung vor .NET Framework SDK und dem MSDN®-Entwicklerprogramm für Visual Studio .NET installieren. Andernfalls ist die Dokumentation für Office Developer Help nicht verfügbar.
- Wenn Sie als aktives Gebietsschema Ihres Betriebssystems Türkisch oder Aserbaidschanisch festlegen, lesen Sie vor der Installation bitte die Infodatei für .NET Framework SDK unter <http://go.microsoft.com/fwlink/?LinkId=501>. In Version 1 von .NET Framework sind bekannte Probleme zu den Gebietsschemas Türkisch und Aserbaidschanisch aufgeführt.

Warnung: Wenn Sie auf Ihrem Computer vorab veröffentlichte Versionen von .NET Framework oder Visual Studio (Release Candidate, Beta 2, Beta 1 oder PDC Technology Preview) installiert haben, müssen Sie folgende Schritte zum Deinstallieren dieser Versionen ausführen.

So deinstallieren Sie Visual Studio .NET Beta 2 oder höhere vorab veröffentlichte Versionen

1. Klicken Sie im Menü **Start** auf **Einstellungen** und anschließend auf **Systemsteuerung**.
2. Klicken Sie auf **Software** und dann auf **Programme ändern oder entfernen**.
3. Wählen Sie (sofern vorhanden) **Visual Studio .NET** aus, und klicken Sie auf **Ändern/Entfernen**. Fahren Sie andernfalls mit Schritt 7 fort.
4. Klicken Sie auf die Verknüpfung 2, **Visual Studio .NET**.
5. Klicken Sie zum Starten des Deinstallationsvorgangs auf **Visual Studio .NET deinstallieren**.
6. Kehren Sie nach dem Beenden der Deinstallation von Visual Studio .NET Beta 2 zum Fenster **Software** zurück.
7. Wählen Sie **Microsoft .NET Framework** aus, und klicken Sie anschließend auf die Schaltfläche **Ändern/Entfernen**.
8. Klicken Sie auf **Ja**, um den Deinstallationsvorgang fortzusetzen.
9. Kehren Sie nach der Deinstallation von Microsoft .NET Framework zum Desktop zurück.

10. Klicken Sie mit der rechten Maustaste auf **Arbeitsplatz**, und klicken Sie im Kontextmenü auf **Verwalten**.
11. Erweitern Sie den Knoten **Dienste und Anwendungen**, und wählen Sie **Dienste** aus.
12. Doppelklicken Sie auf **.NET Framework Support Service**.
13. **Hinweis** Wenn der Dienst nicht aufgeführt ist, fahren Sie mit Schritt 14 fort.
14. Ändern Sie den **Starttyp** in **Deaktiviert**.
15. Starten Sie den Computer neu.

So deinstallieren Sie Visual Studio .NET 7.0 Beta 1

1. Klicken Sie im Menü **Start** auf **Einstellungen** und anschließend auf **Systemsteuerung**.
2. Klicken Sie auf **Software** und dann auf **Programme ändern oder entfernen**.
3. Wählen Sie **Visual Studio .NET 7.0 Enterprise** aus, und klicken Sie anschließend auf **Ändern/Entfernen**.
4. Klicken Sie auf die Verknüpfung 2, **Visual Studio .NET**.
5. Klicken Sie zum Starten des Deinstallationsvorgangs auf **Visual Studio .NET 7.0 Enterprise deinstallieren**.
6. Kehren Sie nach dem Beenden der Deinstallation von Visual Studio .NET 7.0 Beta 1 zum Fenster **Software** zurück.
7. Wählen Sie **Microsoft .NET Framework SDK** aus, und klicken Sie anschließend auf die Schaltfläche **Ändern/Entfernen**.
8. Klicken Sie auf **Ja**, um den Deinstallationsvorgang fortzusetzen.

So deinstallieren Sie Visual Studio 7.0 PDC Technology Preview

1. Klicken Sie im Menü **Start** auf **Ausführen**.
2. Geben Sie im Feld **Ausführen** Folgendes ein:
`msiexec.exe /x {33334F07-0C6B-11D3-A51D-00A0C9231BA1} NOVSUI=1`
3. Klicken Sie nach Aufforderung auf **Ja**, um mit der Deinstallation von Visual Studio .NET PDC Technology Preview fortzufahren.
4. Geben Sie im Feld **Ausführen** Folgendes ein:
`msiexec.exe /x {62EDCF3C-69F4-11D3-A521-00A0C9231BA1}`

5. Klicken Sie nach Aufforderung auf **Ja**, um mit der Deinstallation des Vorabversionsurladers für Visual Studio .NET PDC Technology Preview fortzufahren.
6. Klicken Sie im Menü **Start** auf **Einstellungen** und anschließend auf **Systemsteuerung**.
7. Klicken Sie auf **Software** und dann auf **Programme ändern oder entfernen**.
8. Wählen Sie **Microsoft NGWS SDK**, und klicken Sie anschließend auf **Ändern/Entfernen**.
9. Klicken Sie auf **Ja**, um den Deinstallationsvorgang fortzusetzen.
10. Entfernen Sie sämtliche Dateien aus
n:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Microsoft Help.
Mit diesem Schritt wird die Dokumentation der PDC-Version deinstalliert.

2.2 Installieren von Microsoft® Visual Studio .NET

So installieren Sie Visual Studio .NET

1. Schließen Sie alle geöffneten Anwendungen, um zusätzliche Neustarts während des Setup-Vorgangs zu vermeiden.
2. Legen Sie die CD mit der Aufschrift **Visual Studio .NET CD1** oder **DVD** ein.

Autorun startet die Datei **Setup.exe**. Wenn Autorun deaktiviert ist, führen Sie **Setup.exe** im Stammverzeichnis der Installations-CD oder -DVD aus.

Das Setupprogramm durchsucht die Festplatte nach installierten Komponenten. Wenn bei diesem Vorgang festgestellt wird, dass die Aktualisierung einer Systemkomponente erforderlich ist, führen Sie Schritt 1: Windows Component Update installieren aus, der im Dialogfeld **Installation** verfügbar ist. Wählen Sie Schritt 1, um die Systemkomponenten zu aktualisieren. Wenn keine Aktualisierung von Komponenten erforderlich ist, dann ist diese Option nicht verfügbar. Nach dem Aktualisieren der Systemkomponenten wird im Dialogfeld **Installation** Schritt 2 aktiviert: Visual Studio .NET installieren. Wählen Sie Schritt 2, um Visual Studio .NET zu installieren.

Hinweis: Wenn während des Installationsvorgangs ein Virenschutzprogramm, z. B. Norton AntiVirus 2001, aktiviert ist, werden Warnungen angezeigt, da bei der Installation Skripts ausgeführt werden, die auf das Dateisystemobjekt zugreifen. Die Skripts können gefahrlos ausgeführt werden.

So registrieren Sie Visual Studio .NET

Öffnen Sie im Internet die Registrierungsseite für Visual Studio .NET unter <http://register.microsoft.com/regsys/regsys.asp?wizid=4289&lcid=1033>.

Anhang 3: Aufrüsten auf SQL Server

Zusammenfassung: In diesem Anhang werden die Schritte zum Aktualisieren der Visual Basic .NET Guided Tour beschrieben, um von der Access-Datenbank auf eine SQL Server-Datenbank umzusteigen. Dabei werden folgende Themen behandelt:

- Aktualisieren der Access-Datenbank auf SQL Server
- Hinzufügen einer ASPNET-Benutzeranmeldung
- Einstellen der Zugriffsberechtigungen für die Datenbank
- Anpassung der Anwendung an die neue SQL-Datenbank

3.1 Aktualisieren der Access-Datenbank auf SQL Server

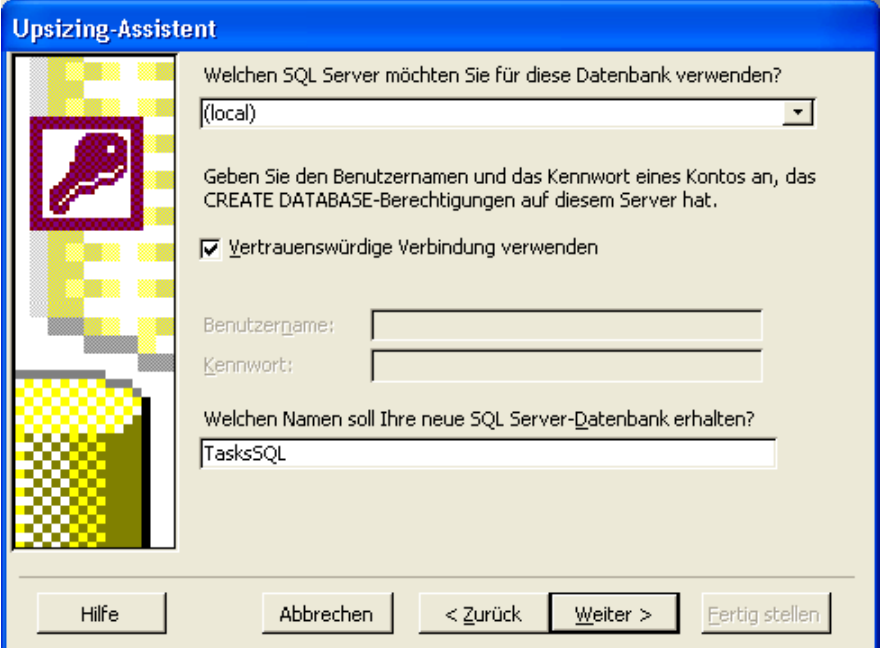
Mit dem Upsizing-Assistent von Access können Sie die Access-Datenbank am unkompliziertesten auf SQL Server aktualisieren. Anhand der folgenden Schritte werden Sie durch diesen Vorgang geleitet.

1. **Öffnen der Access-Datenbankdatei**
Öffnen Sie die Aufgabendatenbank in Access, indem Sie das Verzeichnis C:\Programme\Visual Basic .NET Guided Tour\Projects\TaskManagement\Database\ suchen und auf die Datei Tasks.mdb doppelklicken.
2. **Ausführen des Upsizing-Assistenten**
Klicken Sie im Menü Extras auf **Datenbank-Dienstprogramme** und dann auf **Upsizing-Assistent**. Damit wird der Upsizing-Assistent wie in der unteren Abbildung angezeigt. Übernehmen Sie die Standardoption **Neue Datenbank erstellen**, und klicken Sie auf die Schaltfläche **Weiter**. Der Upsizing-Assistent ist nicht standardmäßig installiert, so dass Access gegebenenfalls zur Installation des Assistenten auffordert.



3. Auswählen von SQL Server

Wählen Sie die SQL Server-Datenbank aus, die als neue Datenbank verwendet werden soll. Wählen Sie **(local)**, oder geben Sie einen Punkt ein, wenn die SQL Server-Datenbank in Ihrem System installiert ist. Aktivieren Sie die Option **Vertrauenswürdige Verbindung verwenden** zur Anmeldung. Der Standardname der Datenbank **TasksSQL** kann übernommen werden. Klicken Sie auf die Schaltfläche **Weiter**, um mit dem nächsten Schritt fortzufahren.



Upsizing-Assistent

Welchen SQL Server möchten Sie für diese Datenbank verwenden?
(local)

Geben Sie den Benutzernamen und das Kennwort eines Kontos an, das CREATE DATABASE-Berechtigungen auf diesem Server hat.

Vertrauenswürdige Verbindung verwenden

Benutzername:

Kennwort:

Welchen Namen soll Ihre neue SQL Server-Datenbank erhalten?
TasksSQL

Hilfe Abbrechen < Zurück Weiter > Fertig stellen

4. Auswählen von Tabellen

Dieser Schritt muss nicht weiter erklärt werden, da die Datenbank nur eine Tabelle enthält. Doppelklicken Sie auf **Tasks**, um diese in die Liste **Nach SQL Server exportieren** zu verschieben. Klicken Sie auf die Schaltfläche **Weiter**.

The screenshot shows the 'Upsizing-Assistent' dialog box with the title 'Upsizing-Assistent'. The main question is 'Welche Tabellen möchten Sie nach SQL Server exportieren?'. There are two list boxes: 'Vorhandene Tabellen:' (empty) and 'Nach SQL Server exportieren:' (containing 'Tasks'). Between them are navigation buttons: '>', '>>', '<', and '<<'. Below the lists is a text field 'Ausgewählte Tabelle:' containing 'Tasks'. At the bottom are buttons: 'Hilfe', 'Abbrechen', '< Zurück', 'Weiter >', and 'Fertig stellen'.

5. Tabellenattribute

Übernehmen Sie alle Standardattribute, und klicken Sie auf **Weiter**, um mit dem nächsten Schritt fortzufahren.

The screenshot shows the 'Upsizing-Assistent' dialog box with the title 'Upsizing-Assistent'. The main question is 'Der Upsizing-Assistent kann zusätzlich zu den Daten Tabellenattribute exportieren.'. Below this is the question 'Welche Tabellenattribute möchten Sie einbeziehen?'. There are four checked checkboxes: 'Indizes', 'Standardwerte', 'Gültigkeitsregeln', and 'Tabellenbeziehungen'. There are also two radio buttons: 'Mit DRI' (selected) and 'Mit Trigger'. Below this is the question 'Welche Datenoptionen möchten Sie aufnehmen?'. There is a dropdown menu for 'Timestamp-Feld in Tabellen einfügen?' with the value 'Ja, der Assistent soll entscheiden'. There is also an unchecked checkbox for 'Nur Tabellenstruktur erstellen, keine Daten.'. At the bottom are buttons: 'Hilfe', 'Abbrechen', '< Zurück', 'Weiter >', and 'Fertig stellen'.

6. ADP-Anwendung

Wir möchten keine Anwendung erstellen. Wählen Sie darum die Option **Keine Anwendungsänderungen**, und klicken Sie dann auf **Weiter**.

Upsizing-Assistent

Der Upsizing-Assistent kann die bestehende Anwendung ändern oder eine neue Anwendung erstellen, so dass sie die SQL Server-Tabellen benutzt.

Welche Anwendungsänderungen möchten Sie durchführen?

Eine neue Access-Client/Server-Anwendung erstellen.

ADP-Dateiname:

SQL Server-Tabellen in die bestehende Anwendung einbinden.

Keine Anwendungsänderungen.


Kennwort und Benutzer-ID speichern.

7. Fertigstellen des Assistenten

Der Assistent hat nun alle erforderlichen Daten erfasst, um die SQL Server-Datenbank zu erstellen. Klicken Sie auf die Schaltfläche **Fertig stellen**, um die SQL Server-Datenbank **TasksSQL** zu erstellen.

Upsizing-Assistent

Dies sind alle Antworten, die der Upsizing-Assistent zum Erstellen Ihrer Datenbank benötigt.

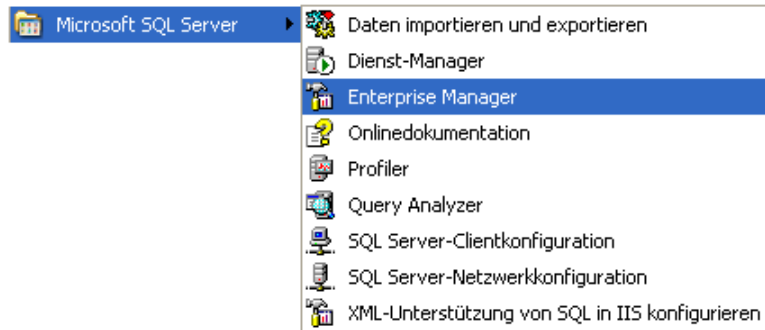


Hinzufügen einer ASPNET-Benutzeranmeldung

Jetzt verfügen Sie über eine SQL Server-Datenbank. Im nächsten Schritt weisen Sie dem ASPNET-Prozess die entsprechenden Zugriffsberechtigungen für die Datenbank zu. Zuerst müssen Sie für SQL Server die ASPNET-Benutzeranmeldung erstellen. Wenn für den Server bereits eine ASPNET-Benutzereinstellung vorhanden ist, können Sie diesen Schritt auch überspringen.

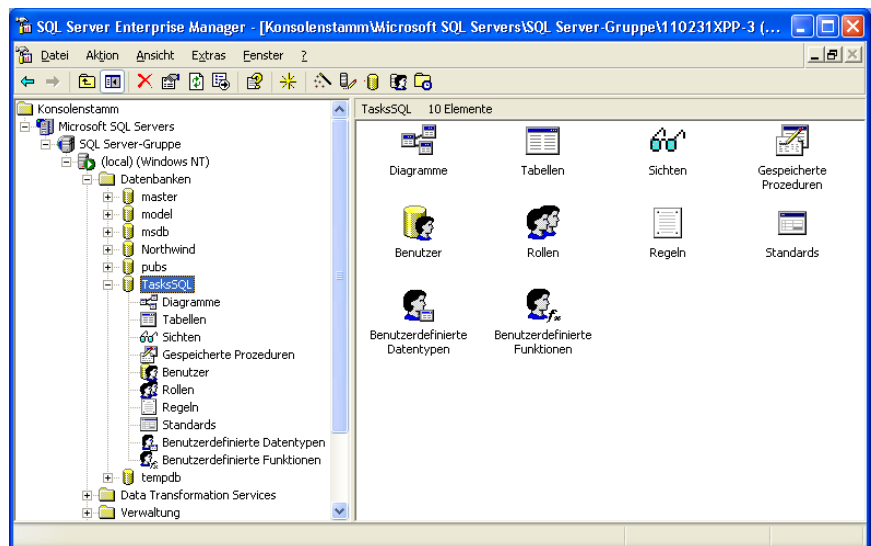
1. Ausführen von SQL Server Enterprise Manager

Führen Sie diesen Schritt aus, indem Sie wie in der unten stehenden Abbildung im Startmenü von Microsoft SQL Server auf **Enterprise Manager** klicken.



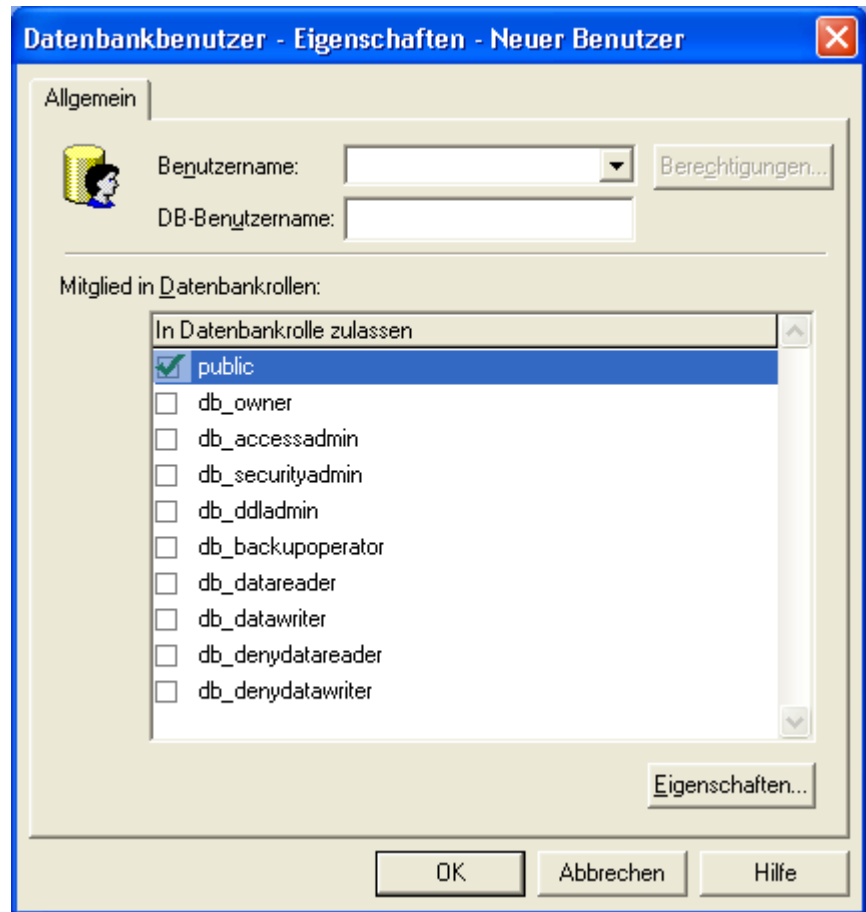
2. Navigieren zur TasksSQL-Datenbank

Der entsprechende Pfad lautet (Konsolenstamm » Microsoft SQL Servers » SQL Server-Gruppe » *Systemname* » Datenbanken » TasksSQL).



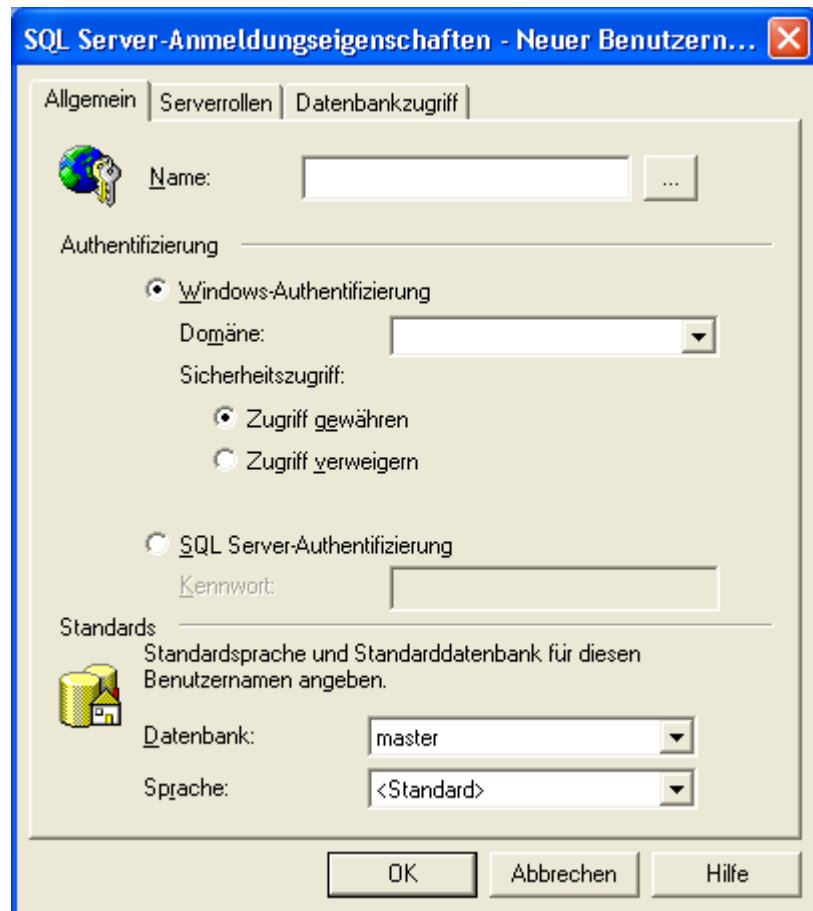
3. Anzeigen des Dialogfelds „Benutzereigenschaften“

Klicken Sie mit der rechten Maustaste auf das Element **Benutzer**, und wählen Sie **Neuer Datenbankbenutzer...** Damit wird das Eigenschaftendialogfeld für Datenbankbenutzer angezeigt.




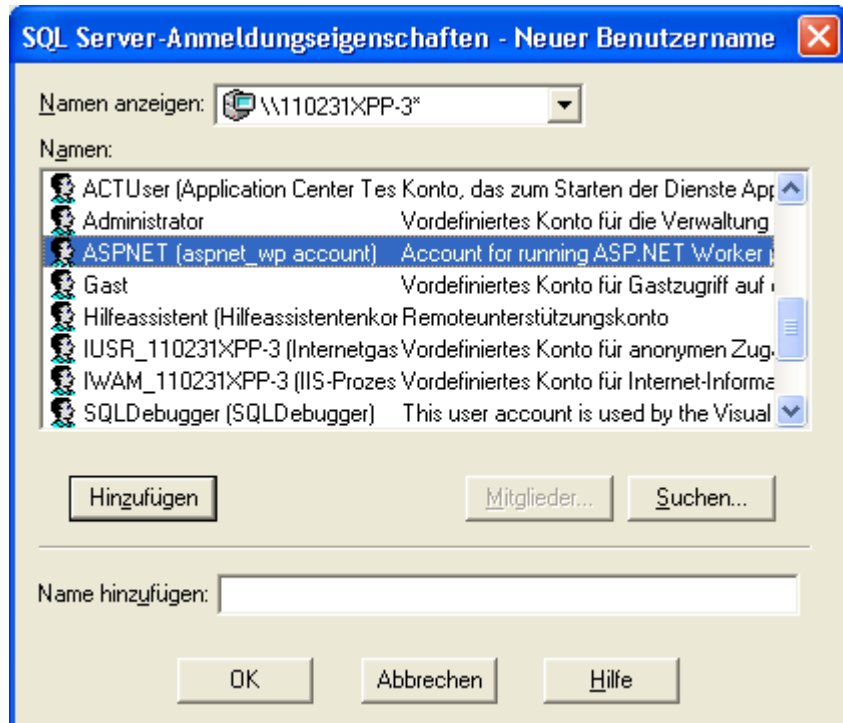
4. Erstellen von neuen Benutzern

Wählen Sie in der Dropdownliste **Benutzername** den Eintrag <neu>, um das Dialogfeld **SQL Server-Anmeldungseigenschaften** anzuzeigen.



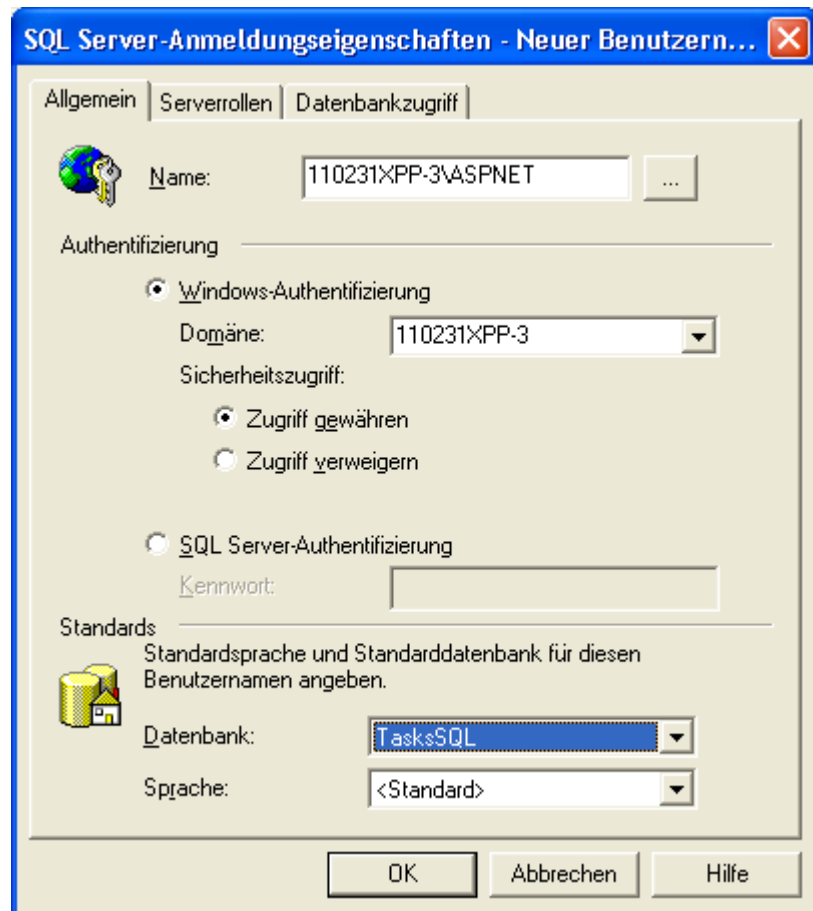
5. Auswählen eines ASPNET-Benutzers

Klicken Sie neben dem Feld **Name** auf die Schaltfläche , um eine Liste der Benutzernamen anzuzeigen. Wählen Sie **ASPNET (aspnet_wp account)**, und klicken Sie auf die Schaltfläche **Hinzufügen**. Klicken Sie auf **OK**, um den Namen zu übernehmen.



6. Angeben der Datenbank

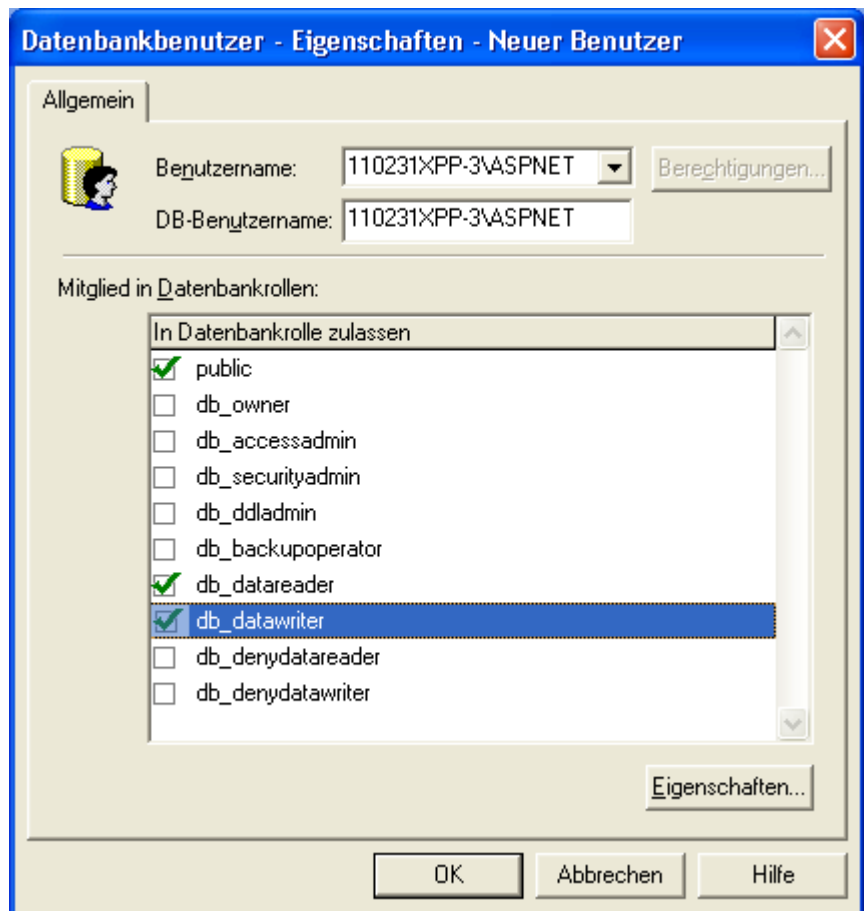
Wählen Sie aus der Dropdownliste **Datenbank** den Eintrag **TasksSQL**, und klicken Sie auf **OK**.



Einstellen der Zugriffsberechtigungen für die Datenbank

Jetzt haben Sie in SQL Server die Anmeldung für den ASPNET-Benutzer eingerichtet. Im folgenden Schritt wird die Berechtigung für den Zugriff auf die TasksSQL-Datenbank gewährt.

- Wählen Sie aus der Dropdownliste **Benutzername** den Eintrag **Systemname\ASPNET**. Aktivieren Sie zusätzlich zu **public** die Datenbankrollen **db_datareader** und **db_datawriter**. Klicken Sie auf die Schaltfläche **OK**.



Anpassung der Anwendung an die neue SQL-Datenbank

Wir haben die Access-Datenbank auf SQL Server aufgerüstet und die ASPNET-Anmeldung mit den richtigen Datenbank-Zugriffsberechtigungen eingerichtet. Im letzten Schritt wird der XML-Webdienst so angepasst, dass anstelle der Access- die SQL Server-Datenbank verwendet wird.

1. **Öffnen der Projektmappe**

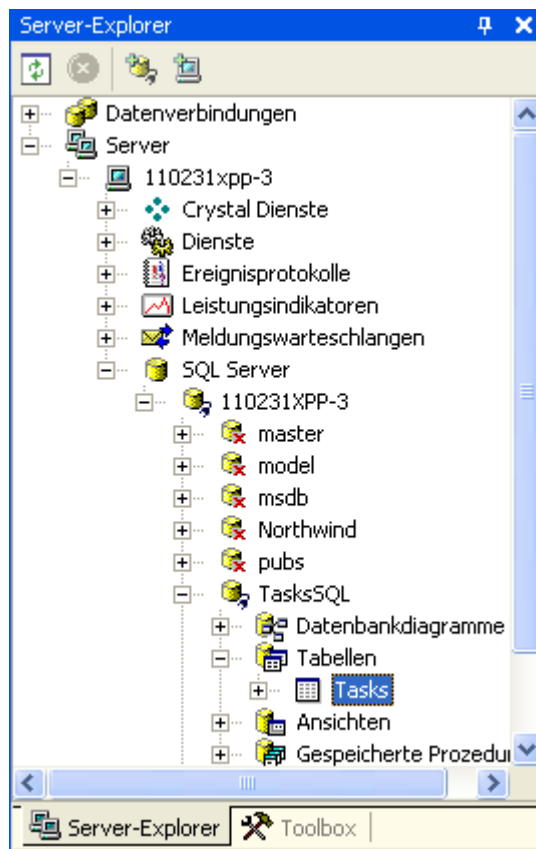
Öffnen Sie die Projektmappe **Guided Tour** in Visual Studio. Öffnen Sie im Webdienstprojekt **TasksWS** die Datei **Service1.aspx**.

2. **Löschen der OleDb-Datenobjekte**

Löschen Sie die vorhandenen Datenobjekte **OleDbDataAdapter1** und **OleDbConnection1**.

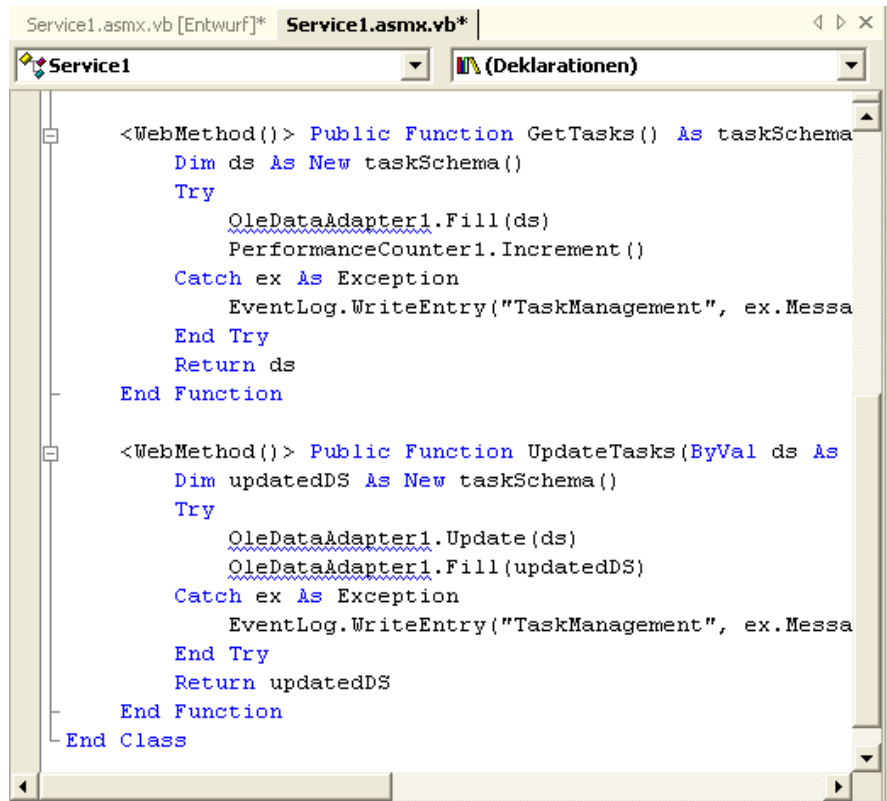
3. **Erstellen neuer SQL-Datenobjekte**

Öffnen Sie den **Server-Explorer**, und navigieren Sie nach unten zu der SQL Server-Datenbank **TasksSQL**. Der entsprechende Pfad lautet (Server » *Systemname* » SQL Server » *Systemname* » TasksSQL). Erweitern Sie die Datenbank, und navigieren Sie zu der Tabelle **Tasks**. Ziehen Sie die Tabelle in die Entwurfsoberfläche von **Service1.aspx**. Visual Studio erstellt zwei SQL-Datenobjekte; **SqlDataAdapter1** und **SqlConnection1**.



4. Verwenden von neuen SQL-Datenobjekten

Drücken Sie auf F7, um den Code anzuzeigen. Beachten Sie, dass das **OleDbDataAdapter1**-Objekt mit blauen Wellenlinien versehen ist, da das Objekt nicht mehr deklariert ist.



```
<WebMethod()> Public Function GetTasks() As taskSchema
    Dim ds As New taskSchema()
    Try
        OleDbDataAdapter1.Fill(ds)
        PerformanceCounter1.Increment()
    Catch ex As Exception
        EventLog.WriteEntry("TaskManagement", ex.Message)
    End Try
    Return ds
End Function

<WebMethod()> Public Function UpdateTasks(ByVal ds As TaskSchema) As TaskSchema
    Dim updatedDS As New TaskSchema()
    Try
        OleDbDataAdapter1.Update(ds)
        OleDbDataAdapter1.Fill(updatedDS)
    Catch ex As Exception
        EventLog.WriteEntry("TaskManagement", ex.Message)
    End Try
    Return updatedDS
End Function
End Class
```

Ersetzen Sie wie in der unten stehenden Abbildung die drei Instanzen des **OleDbDataAdapter1**-Objekts durch **SqlDataAdapter1**.

```
<WebMethod()> Public Function GetTasks() As TaskSchema
    Dim ds As New TaskSchema()
    SqlDataAdapter1.Fill(ds)
    PerformanceCounter1.Increment()
    Return ds
End Function
```

```
<WebMethod()> Public Function UpdateTasks(ByVal ds As TaskSchema) As TaskSchema
    Dim updatedDS As New TaskSchema()
    SqlDataAdapter1.Update(ds)
    SqlDataAdapter1.Fill(updatedDS)
    Return updatedDS
End Function
```

Erstellen und Ausführen

Erstellen Sie die Anwendung neu, und führen Sie sie aus. Die Anwendung weist äußerlich keine Unterschiede auf, das TaskManagement-Projekt wurde faktisch nicht verändert, greift nun aber statt auf eine Access-Datenbank auf eine SQL Server-Datenbank zu. Dazu waren nur einige wenige Änderungen des XML-Webdienstes erforderlich.

