

Kapitel 21

ActiveX-Controls und das Internet

- 21.1 Das HyperLink-Objekt 628
- 21.2 ActiveX-Controls auf Web-Seiten 629
- 21.3 Downloading, Sicherheit und Signaturen 630
- 21.4 Das Steuerelement dwBanner 646

In Kapitel 1, »ActiveX-Mythen«, habe ich versucht, ActiveX-Controls nicht mit dem Internet in Verbindung zu bringen. Und in den letzten 20 Kapiteln wurde das Internet in Hinblick auf ActiveX-Controls nur in Kapitel 16 erwähnt, wo die Einschränkungen dieser Steuerelemente besprochen wurden, sowie in Kapitel 19, wo es um asynchrone Eigenschafts-Downloads ging. Ich denke, das sollte meine Meinung mehr als deutlich machen. Die Bedeutung der ActiveX-Komponenten und Steuerelemente hat nur wenig mit dem Internet zu tun.

Aber wir können nicht ewig die Augen vor den Internet-Funktionen der ActiveX-Controls verschließen. Microsoft betont in der Werbung für Visual Basic ausdrücklich, daß man damit Steuerelemente schaffen kann, die im Internet sowie in Intranets eingesetzt werden. Und wenn Sie damit leben können, daß nicht jeder Benutzer Ihre Steuerelemente nutzen kann, dann ist Visual Basic ein ausgezeichnetes Werkzeug für die Entwicklung dieser Steuerelemente. Sie erzielen eine ausgezeichnete Performance, kurze Entwicklungszeiten, kleine Programmdateien, und mit der gelegentlichen Unterstützung durch ein Werkzeug von einem Dritthersteller ebenso viel Leistung wie Steuerelemente, die unter Verwendung anderer Sprachen entwickelt wurden, beispielsweise Visual C++. In einigen Bereichen sind die VB-Steuerelemente den mit Visual C++ erzeugten Steuerelementen sogar überlegen. Beispielsweise ist es mit VB viel einfacher, ein Steuerelement zu entwickeln, das andere Steuerelemente beinhaltet.

Dieses Kapitel konzentriert sich insbesondere auf die Aspekte von ActiveX-Steuerelementen, die mit dem Internet zu tun haben:

- Die Navigations-Fähigkeiten von Steuerelementen, wenn sie in einem Browser verwendet werden
- Die Technik, wie ein Steuerelement auf einer Web-Seite angelegt wird
- Themen, die mit dem Downloading von Steuerelementen von Web-Seiten zu tun haben

Dabei kann ich vielleicht auch gleich einige Gerüchte zerstreuen, die sich auf die Sicherheit von ActiveX-Steuerelementen beziehen.

21.1 Das HyperLink-Objekt

Das HyperLink-Objekt steht über die HyperLink-Eigenschaft des UserControl-Objekts zur Verfügung. Es stellt drei Methoden bereit: GoBack, GoForward und NavigateTo. Diese Methoden haben nur Auswirkungen, wenn sich das Steuerelement in einem Container befindet, der als Web-Browser agiert. In allen anderen Containern werden sie ignoriert.

Die Methoden GoBack und GoForward entsprechen dem Anklicken der Schaltflächen ZURÜCK und VORWÄRTS im Browser. Die Methode NavigateTo ist sehr viel leistungsfähiger. Sie nimmt drei Parameter entgegen: ein Ziel, eine Position und einen Frame.

Das Ziel ist die URL, die angesprungen werden soll. Dabei kann es sich um jede vom Container unterstützte Position handeln, lokal oder auf dem Netzwerk. Der Positionsparameter ist optional und gibt den Namen der zu ladenden Datei an. Ist er nicht angegeben, wird die Standarddatei für diese Site geladen. Der Frame-Parameter ist ebenfalls optional und gibt einen Frame an, der an der betreffenden Zielposition geladen werden soll.

Diese Navigation wird im Steuerelement `dwBanner` demonstriert, das das Click-Ereignis erkennt und auf die Web-Site von Desaware springt. Dazu verwendet es den folgenden Code:

```
Private Sub UserControl_Click()  
    On Error GoTo NoHyper  
    UserControl.HyperLink.NavigateTo "http://www.desaware.com"  
NoHyper:  
    RaiseEvent Click  
End Sub
```

Bei der Arbeit mit dem `HyperLink`-Objekt sollten Sie die Fehlersuche immer aktivieren, weil ein Fehler auftreten kann, wenn der Container die Internet-Navigation nicht unterstützt.

21.2 ActiveX-Controls auf Web-Seiten

Kapitel 15 bot Ihnen eine schnelle Einführung in HTML. Sie haben gesehen, daß sich HTML aus Tags und Text zusammensetzt. Die Tags, die in spitze Klammern eingeschlossen sind, stellen Befehle für den Browser dar, die die Operationen oder die Textanzeige vornehmen. ActiveX-Controls werden mit Hilfe des OBJECT-Tags auf einer Web-Seite plazierte, wie hier gezeigt:

```
<HTML>  
  <HEAD>  
    <TITLE>New Page</TITLE>  
  </HEAD>  
  <BODY>  
  
    <OBJECT ID="dwBanner1" WIDTH=347 HEIGHT=39  
      CLASSID="CLSID:63B8AF4F-8E9A-11D0-91BB-00AA0036005A">  
  
        <PARAM NAME="_ExtentX" VALUE="9181">  
        <PARAM NAME="_ExtentY" VALUE="1005">  
        <PARAM NAME="BackColor" VALUE="16777088">  
        <PARAM NAME="FontSize" VALUE="8.25">  
        <PARAM NAME="FontBold" VALUE="-1">  
        <PARAM NAME="FontItalic" VALUE="-1">  
        <PARAM NAME="BorderStyle" VALUE="1">  
        <PARAM NAME="ScrollText" VALUE="Visit Desaware's Web site at
```

```

        www.desaware.com for the coolest tools "> for VB developers.
        <PARAM NAME="XMargin" VALUE="20">
        <PARAM NAME="YMargin" VALUE="6">
    </OBJECT>

</BODY>
</HTML>

```

Das Feld CLASSID stellt die eindeutige ID des Steuerelements (CLSID) dar, die beim Kompilieren Ihres Steuerelements erzeugt wird. ActiveX-Steuerelemente haben, wie alle ActiveX-Objekte, diese eindeutige ID (auch als GUID bezeichnet). Beachten Sie, daß die ID aus dem obigen Listing möglicherweise nicht mit einer echten ID für das Steuerelement `dwBanner.ocx` übereinstimmt, falls das Steuerelement noch einmal für die endgültige Version kompiliert wurde, nachdem das Buch gedruckt wurde.

Der Browser erwartet, daß alle Informationen vom `<OBJECT>`-Tag bis zum `</OBJECT>`-Tag das Objekt definiert. Die Tags `<PARAM NAME >` definieren die Eigenschaftswerte, die das Steuerelement beim Erzeugen hat. Diese Werte werden vom Steuerelement während des `ReadProperties`-Ereignisses angenommen.

Wenn Ihnen das Ganze recht kompliziert erscheint, dann habe ich eine gute Nachricht für Sie. Sie werden wahrscheinlich nie in Ihrem Leben solchen HTML-Code schreiben. Microsoft bietet mehrere Werkzeuge, die diesen HTML-Text automatisch erzeugen. Das vielleicht einfachste davon ist das ActiveX Control Pad, das von der Web-Site von Microsoft heruntergeladen werden kann (momentan kostenlos). Dieses Werkzeug ermöglicht Ihnen, ein Steuerelement in eine HTML-Datei einzufügen und seine Entwurfszeit-Eigenschaften interaktiv zu bearbeiten. FrontPage 98 und Microsoft Internet Studio erlauben Ihnen, einfach ActiveX-Controls in Web-Seiten einzufügen, wobei Sie sich nie um den HTML-Code kümmern müssen, der das Ganze realisiert.

Wenn Sie Ihr Steuerelement auf einer Web-Seite ausführen möchten, müssen Sie es mit Hilfe eines Browsers testen, der es unterstützt. Diese Aussage scheint Ihnen im Moment vielleicht etwas seltsam, aber ich kann es gar nicht oft genug sagen. Es ist ganz einfach, Steuerelemente zu entwickeln, die perfekt in Visual Basic und anderen Applikations-Containern arbeiten, die aber auf einem Web-Browser dann kläglich versagen. Und das gilt insbesondere dann, wenn Sie das Extender-Objekt verwenden.

21.3 Downloading, Sicherheit und Signaturen

Der wichtigste Aspekt in Hinblick auf die Verwendung von ActiveX-Controls im Internet oder in Firmen-Intranets betrifft das Downloading von Steuerelementen von einer Web-Site auf Ihr System. Dieses Thema hat vielleicht zu den meisten Kontroversen geführt. Um die Art dieser Kontroversen zu verstehen, und ebenso, wie Microsoft versucht hat, sie für ActiveX-Controls aufzulösen, wollen wir mit der Sicherheit beginnen.

21.3.1 ActiveX-Controls und Sicherheit

So lange Web-Seiten aus einfachem Text und Grafik bestanden, war Sicherheit nicht wirklich ein Thema. Die Anzeige einer Textdatei oder eines Bildes wird sehr wahrscheinlich kein System zerstören. Das ist einer der Gründe, warum E-Mail so sicher und beliebt ist. Wenn Sie eine E-Mail von irgend jemandem erhalten, dann denken Sie nicht daran, daß es etwas sein könnte, das Ihre Festplatte beschädigen oder Ihre wertvollen Daten löschen könnte. Aber Web-Seiten, die nur einfachen Text und Grafiken enthalten, können auch sehr langweilig sein.

Das sagen die Leute wenigstens. Wenn ich das Web verwende, um Informationen zu finden, dann ist es mir ganz recht, daß Web-Seiten nur Text und ein paar Grafiken enthalten. Ich habe zwar im Büro eine T1-Leitung im Einsatz, aber zu Hause arbeite ich mit einem 28.8-k-Modem. Die großen Bitmaps, unruhigen Hintergrundbilder oder Animationen, die manche Entwickler auf ihren Web-Sites präsentieren, sind reine Zeitverschwendung. Aber ich komme vom Thema ab...

Sache ist, daß die Leute irgendwann komplexere Inhalte auf den Web-Seiten wünschten. Sie wollten Seiten sehen, die mit dem Benutzer interagieren, die scrollenden Text oder protzige Animationen anzeigen, komplexe Anzeigeeigenschaften aufweisen oder andere Operationen auf den Systemen der Benutzer ausführen. Viele dieser Eigenschaften machen es erforderlich, daß die Seite in der Lage ist, Code auf dem System auszuführen, auf der sie angezeigt werden soll. Diese Elemente werden mit dem Begriff »aktiver Inhalt« beschrieben.

Der Schlüsselsatz ist hier »Code auf dem System auszuführen, auf dem sie (die Seite) angezeigt werden soll«. Was bedeutet das? Es bedeutet, daß das System, das auf die Seite zugreift, Software herunterlädt und sie ausführt.

Wenn Sie ein ernsthafter Software-Entwickler oder Power-User sind, dann wissen Sie sehr wahrscheinlich, daß das Herunterladen von Software wohl mit das Gefährlichste ist, was Sie tun können. Nicht das eigentliche Herunterladen ist so gefährlich, sondern vielmehr die Tatsache, daß Sie möglicherweise nicht wissen, wer die Software geschrieben hat, und was sie alles in sich hat. Sie könnte Viren oder andere negative Nebeneffekte aufweisen. Das ist ein Grund, warum die meisten Entwickler am liebsten nur mit Software von vertrauenswürdigen Firmen arbeiten (auch Shareware- oder Freeware-Firmen), und vermeiden, Tausende von »kostenlosen« Programme zu verwenden, die in der Regel genau das wert sind, was man dafür bezahlt. Der andere Grund, warum ernsthafte Entwickler normalerweise keine kostenlosen Programme verwenden – falls Sie immer noch nicht überzeugt sind –, ist, daß es für diese selten Support gibt. Und nicht supportete Software hat die ungezogene Eigenheit, auf lange Sicht sehr viel mehr zu kosten, als selbst die teuerste kommerzielle Software.

Wenn Sie Web-Seiten entwickeln möchten, die aktiven Inhalt unterstützen, dann meinen Sie damit letztlich, daß Code auf der Seite abgelegt wird, der auf dem System des Endbenutzers ausgeführt werden kann.

Wenn Ihr Endbenutzer jedoch nicht gerade schwachsinnig ist, dann wird er es seinem Web-Browser nie erlauben, automatisch Software von einer Seite herunterzuladen und diese auf seinem System auszuführen. Wie können Sie Ihren Benutzern garantieren, daß die Software auf einer Web-Seite sicher ist und gefahrlos auf einem System ausgeführt werden kann. Für die Lösung dieses Problems gibt es zwei philosophische Ansätze.

21.3.2 Der Sandkasten-Ansatz

Eine Antwort wäre, die Software selbst anzuschauen. Was wäre, wenn Sie eine Sprache entwickeln würden, deren Ausführung grundsätzlich sicher wäre? Eine solche Sprache müßte die folgenden Eigenschaften haben:

- Sie darf nicht in der Lage sein, direkte Aufrufe von Betriebssystemfunktionen auszuführen, weil Betriebssystemfunktionen in der Lage sind, ein System zu beschädigen.
- Sie darf nicht in der Lage sein, auf die Platte zu schreiben, oder nur unter streng kontrollierten Bedingungen.
- Sie darf nicht beliebig auf den Speicher zugreifen. Das bedeutet, sie sollte nicht in der Lage sein, Speicherzeiger direkt zu manipulieren, weil das dazu führen könnte, daß andere Applikationen oder gar das Betriebssystem gestört werden.

Die Sprache würde möglicherweise zu Code für die jeweiligen Zielmaschinen kompiliert. Statt dessen müßte sie interpretiert werden (vielleicht auf ganz ähnliche Weise wie Visual Basic zu Pseudocode, oder P-Code, kompiliert wird, der von der VB-Laufzeitumgebung interpretiert wird). Das bedeutet gleichzeitig, daß Software, die in dieser Sprache geschrieben ist, auf jedem System ausgeführt werden könnte, das den P-Code-Interpreter unterstützt. Sie wäre nicht auf ein einziges Betriebssystem oder einen CPU-Typ beschränkt.

Ein Programm, das in dieser Sprache geschrieben ist, wäre natürlich im Vergleich zu einer typischen Anwendung, die vollen Zugriff auf alle Funktionen ihres Zielsystems hat, wesentlich eingeschränkt. Es wäre jedoch auch sicher in seiner Ausführung, weil jede Möglichkeit, das Zielsystem zu stören, ausgeschaltet wurde. Sie können sich das Programm so vorstellen, als würde es in einem Sandkasten auf dem System ausgeführt – einem sicheren Bereich, wo es sich nach Belieben verhalten kann, ohne das restliche System zu stören.

Die Programmiersprache Java ist eine solche Sprache. Wenn ein Java-Applet auf einer Web-Seite erscheint, wird es vom Web-Browser erkannt und an eine virtuelle Java-Maschine geschickt, die den Java-Code interpretiert und ausführt. Der Code ist plattformunabhängig, weil es die virtuelle Java-Maschine für fast jedes Betriebssystem und jede Plattform gibt. Die Sprache Java unterstützt keine Zeiger, sie kann also den Speicherinhalt Ihres Systems nicht zerstören. Die virtuelle Maschine erlaubt keine unsicheren Operationen und auch keinen direkten Zugriff auf das Betriebssystem.

Wenn Sie also ein Java-Applet auf einer Web-Seite sehen, brauchen Sie grundsätzlich nichts zu befürchten, wenn Sie es herunterladen.

Beachten Sie jedoch, daß ich kein Sicherheitsexperte in Sachen Internet bin. Ich bin mir sicher, daß irgendeiner von Ihnen mehr als glücklich sein wird, Sicherheitslöcher in diesem Ansatz bloßzustellen. Und das ist gewissermaßen genau der Punkt. Alle Sicherheitslücken in diesem Ansatz sind Lücken, die vielleicht korrigiert werden können und sollten. Und grundsätzlich ist dieser Ansatz nicht schlecht. Die meisten Leute denken nicht zweimal darüber nach, ob sie es ihren Browsern erlauben sollen, Java-Applets auf ihrem System auszuführen. Außerdem ist es ganz wichtig, zwischen Java der Skripting-Sprache, die auf einer Web-Seite ausgeführt wird, und der allgemeinen Sprache Java, die genutzt werden kann, um alle anderen Komponenten zu entwickeln, unter anderem auch COM-Komponenten, zu unterscheiden. Das eine ist eine relativ sichere Skripting-Sprache, das andere ist so gefährlich wie jede andere Sprache.

21.3.3 Der »Vertrau mir!«-Ansatz

Microsoft verfolgt mit den ActiveX-Controls einen ganz anderen Ansatz. Man hat beschlossen, daß der aktive Inhalt vollen Zugriff auf alle Funktionen des zugrundeliegenden Systems haben sollte.

Wenn ich Microsofts Aktionen, die ich in Kapitel 2, »ActiveX: Eine historische (aber auch technische) Betrachtung«, beschrieben habe, richtig interpretiere, dann hatte Microsoft gar keine andere Wahl. Man entschied, die OLE-Steuerelemente für das Internet zu erweitern und der Technologie einen neuen Namen zu geben, ActiveX. Aber die COM- und die ActiveX-Technologie waren nie für den Sandkasten vorgesehen, und existierende OLE-Steuerelemente wurden in Visual C++ geschrieben, was alles andere als eine sichere Sprache ist. Die Implementierung von OLE-Steuerelementen basierte auf einem extensiven Zugriff auf zugrundeliegende Systemaufrufe. OLE selbst war also Teil des Betriebssystems geworden.

Sie sollten immer daran denken, daß jedes ActiveX-Steuerelement ein richtiges Programm ist. Es hat vollen Zugriff auf alle Ressourcen und die gesamte Funktionalität eines Systems, eingeschränkt nur durch die Sicherheitseinstellungen auf dem System selbst.

Dadurch wird ein ActiveX-Control außerordentlich flexibel und leistungsfähig. Sie können in einem Steuerelement fast alles machen, was auch in einer Anwendung möglich ist. Damit ist es aber gleichzeitig extrem gefährlich. Sie könnten ein ActiveX-Control schreiben, das nach dem Herunterladen die Platte oder die Registrierungseinstellungen des Benutzers löscht, so daß dieses System nicht mehr genutzt und sogar nicht mehr gebootet werden kann. Ich warte sogar irgendwie auf die Meldung, daß es Computerviren geben wird, die auf ActiveX-Controls basieren, und die den Virus über heruntergeladene Steuerelemente verbreiten. Microsoft wußte das und verfolgte deshalb einen anderen Ansatz.

Es stimmt, daß ein cleverer Computer-Benutzer nie erlauben würde, daß beliebige Software heruntergeladen und auf seinem System ausgeführt wird. Aber die Computer-Benutzer erhalten ständig irgendwelche Software. Sie kaufen sie in Geschäften und laden sie von vertrauenswürdigen Firmen oder anderen bekannten Quellen herunter, von denen man weiß, daß sie sicheren und zuverlässigen Code produzieren. Weil ActiveX-Controls nicht daraufhin entwickelt wurden, sicher zu sein, werden drei Dinge wichtig:

- Es muß eine Möglichkeit geben, daß die Autoren von Steuerelementen den Benutzern mitteilen, ob ihr Steuerelement sicher herunterzuladen ist.
- Die Benutzer müssen sicherstellen können, daß ein Steuerelement von einer zuverlässigen, bekannten und auch sonst vertrauenswürdigen Firma oder einem entsprechenden Entwickler stammt.
- Es muß möglich sein, sicherzustellen, daß das Steuerelement nicht von einem Dritten verändert wurde, nachdem der Entwickler es veröffentlicht hat.

21.3.4 Sicherheit ≠ Signatur

Es gibt unendlich viele Informationen über die Bereitstellung von ActiveX-Controls im Internet. Da ist zum einen die Information, die in Visual Basic enthalten ist, und die beschreibt, wie der Setup-Assistent (jetzt auch als Paket- und Weitergabe-Assistent bezeichnet) benutzt wird. Es gibt den Internet Software Development Kit (SDK) (zuvor als ActiveX Software Development Kit bezeichnet), der auf der Web-Site von Microsoft zur Verfügung steht. Es gibt zahlreiche Artikel in Magazinen und Büchern. Dieses Thema kann sehr schnell sehr kompliziert werden, insbesondere, wenn Sie anfangen, sich mit allen Variationen zu beschäftigen, die es für die Installation und das Herunterladen von Steuerelementen gibt.

Ich habe Tage damit verbracht, mich durch diese Informationen zu kämpfen und sie zu verstehen. Ich habe fast alle der vorgestellten Konzepte verstanden (zumindest bilde ich mir das ein), aber irgendwie konnte ich mir einfach nicht vorstellen, wie man das alles machen sollte, wovon da gesprochen wurde. Mit anderen Worten, ein Artikel hat in aller Ausführlichkeit erklärt, wie wichtig es sei, ein Steuerelement als »Sicher für das Skripting« oder »Sicher für die Initialisierung« zu markieren, hat aber mit keinem Wort erwähnt, wie man das macht. Die Autoren sprachen über die Vorteile, Code zu signieren, ohne jedoch zu erklären, wie man ein Steuerelement signiert.

Nach zahlreichen Experimenten habe ich schließlich herausgefunden, worum es ging. Ich möchte hier nicht auf jeden Aspekt des Downloadings und der Sicherheit eingehen, sondern so klar, kurz und genau wie möglich erklären, wie man Steuerelemente entwickelt, die sicher von Web-Seiten heruntergeladen werden können.

Zunächst müssen Sie verstehen, daß hier drei völlig verschiedene Themen angesprochen werden sollen:

- Wie Sie Ihr Steuerelement sicher machen
- Wie Sie Ihr Steuerelement als sicher markieren
- Wie Sie Ihr Steuerelement kennzeichnen (signieren)

Wie Sie Ihr Steuerelement sicher machen

Als Autor von Steuerelementen machen Sie Ihr Steuerelement sicher, so daß es das System eines Endbenutzers nicht beschädigen kann. Es gibt zwei Arten von Sicherheit, die hier in Betracht zu ziehen sind:

- Ist es sicher, Ihr Steuerelement in einen Container zu laden, egal welche Einstellungen für die Eigenschaften vorgegeben waren?
- Ist es sicher, Ihr Steuerelement in einer Anwendung zu verwenden, unabhängig davon, welche Methoden des Steuerelements aufgerufen werden, und welche Eigenschaften zugewiesen werden?

Der erste Aspekt wird als »Sicher für die Initialisierung« bezeichnet. Ihr Steuerelement ist sicher für die Initialisierung, wenn es während des `ReadProperties`-Ereignisses beliebige Eigenschaften annehmen kann, und das System nicht beschädigt. Wenn Ihr Steuerelement beispielsweise einen Eigenschaftswert hat, der Einträge der Registrierung ändern oder das Dateisystem manipulieren kann, indem bei der Initialisierung eine bestimmte Eigenschaft gesetzt wird, ist dieses Steuerelement nicht sicher. Sie wissen, daß die Web-Autoren beliebige Einträge in das `<PARAM>`-Tag in HTML vornehmen können. Sie können also nicht voraussetzen, daß die gelesenen Eigenschaftswerte von Ihrem eigenen Steuerelement geschrieben wurden. Ihr Steuerelement muß in der Lage sein, alles zu verarbeiten, was ihm begegnet.

Der zweite Sicherheitstyp ist das sogenannte »Sicher für das Skripting«. Neben den ActiveX-Controls-Objekten können Web-Seiten auch andere Inhalte beispielsweise in Form von Java-Applets oder VBScript-Applets (zwei verschiedene Web-Skriptingsprachen) haben. Wenn Ihr Steuerelement alle möglichen Methodenaufrufe und Eigenschaftswerte korrekt verarbeiten kann, ist es sicher für das Skripting. Testen Sie Ihr Steuerelement sorgfältig. Angenommen, Sie haben eine Methode, die Informationen aus dem Steuerelement unter einem bestimmten Pfad speichern kann. Das muß nicht unbedingt unsicher sein, aber was macht Ihr Steuerelement, wenn der Pfadname gleich dem Namen eines Systemgeräts ist? Der falsche Methodenparameter könnte den Betrieb dieses Geräts stören. Oder was ist, wenn Sie eine Eigenschaft haben, die erlaubt, einen Registrierungsschlüssel zu setzen? Wenn der Registrierungsschlüssel in dem Steuerelement fest vorgegeben ist, dann kann das sicher sein. Handelt es sich jedoch dabei um einen Parameter für eine Eigenschaft des Steuerelements, könnte ein boshafter Web-Skript-Autor mit Hilfe Ihres Steuerelements wüste Zerstörungen verursachen.

Beachten Sie, daß sich die Sicherheit, wie sie im Kontext eines Internet-Steuerelements verstanden wird, davon unterscheidet, ein Steuerelement sicher für

Anwendungsentwickler zu machen. Es gibt viele Situationen, wo es durchaus wünschenswert wäre, daß Ihre ActiveX-Steuerelemente eine Funktionalität unterstützen, die auf einer Web-Seite nicht sicher wäre.

Beispielsweise bietet StorageTools von Desaware ein ActiveX-Steuerelement zur Manipulation der Registrierung, mit dessen Hilfe die VB-Entwickler die Registrierung so einfach wie ihr Dateisystem abändern können. Das Steuerelement ist sicher in der Hinsicht, daß es korrekt arbeitet, aber weil es den Entwicklern vollen Zugriff auf die Systemregistrierung gibt, braucht man wohl nicht zu sagen, daß die Entwickler es auch nutzen könnten, um unerlaubte Informationen in die Registrierung zu schreiben, wenn sie das denn wollten. Aber wenn die Entwickler ihre Anwendungen verkaufen, die dieses Registrierungs-Steuerelement verwenden, versichern sie ihren Kunden, daß das Programm sicher ausgeführt werden kann, es gibt also kein Problem. Das Registrierungs-Steuerelement ist jedoch unsicher, weil es möglich ist, es systemschädigend einzusetzen. Das Steuerelement ist also zwar sicher für Anwendungsentwickler, aber nicht als sicher für das Skripting gekennzeichnet.

Später in diesem Kapitel werden Sie sehen, daß ein Steuerelement zwei Modi unterstützen kann: Einen sicheren Modus für die Verwendung auf Web-Seiten und einen unsicheren Modus, in dem sie nur von Anwendungsentwicklern verwendet werden.

Wie Sie Ihr Steuerelement als sicher markieren

Nachdem Sie Ihr Steuerelement sicher für die Initialisierung und/oder sicher für das Skripting gemacht haben, müssen Sie im nächsten Schritt dem Endbenutzer mitteilen, welche Sicherheitsstufe Ihr Steuerelement unterstützt. Das wird später in diesem Kapitel noch detailliert erklärt. Hier sollten Sie vorerst nur wissen, daß es zwei Möglichkeiten gibt, ein Steuerelement als sicher zu markieren. Die eine ist, der Systemregistrierung Informationen hinzuzufügen, die anzeigen, daß das Steuerelement sicher ist. Die andere ist, dem Steuerelement selbst eine Funktionalität mitzugeben, mit deren Hilfe die Browser-Software einen sicheren Einsatz fordern, und feststellen kann, ob das Steuerelement sicher ist.

Wenn der Browser das Steuerelement herunterlädt, probiert er beide Methoden aus, um zu prüfen, ob das Steuerelement sicher ist. Was er mit dieser Information macht, ist von der Browser-Konfiguration abhängig. Mehr dazu später.

Wie Sie Ihr Steuerelement signieren

Fast jeder Artikel, in dem es um die Sicherheit von ActiveX-Controls geht, erwähnt auch die Code-Signaturen. Man könnte dadurch leicht den Eindruck gewinnen, die Code-Signatur hätte etwas mit der Code-Sicherheit zu tun. Tatsächlich aber handelt es sich dabei um zwei völlig verschiedene Dinge.

Früher in diesem Kapitel habe ich erklärt, daß es zwei Bedingungen gibt, die ActiveX-Controls erfüllen müssen, um eine zulässige Komponente für den Download aus dem Internet darzustellen.

- Es muß eine Möglichkeit für die Autoren von Steuerelementen geben, den Benutzern mitzuteilen, ob ihr Steuerelement sicher herunterzuladen ist.
- Die Benutzer müssen sicherstellen können, daß ein Steuerelement von einer zuverlässigen, bekannten und auch sonst vertrauenswürdigen Firma oder einem entsprechenden Entwickler stammen.

Sie haben bereits gelesen, daß es eine Möglichkeit für die Autoren gibt, ein Steuerelement als sicher zu markieren, aber Sie haben noch nicht erfahren, wie das geht. Die Wahrheit ist jedoch, daß dies gar keine Lösung ist. Sehen Sie, niemand kann ein Steuerelement als sicher markieren. Bei der Markierung eines Steuerelements als sicher gibt es nichts, was zuverlässig sicherstellt, daß das Steuerelement wirklich sicher ist. Damit kommen wir zu einer interessanten Frage: Woher wissen Sie, daß ein als sicher markiertes Steuerelement wirklich sicher zu installieren und auszuführen ist? Die Antwort ist: Sie können es nicht wissen. Sie müssen darauf vertrauen, daß der Autor des Steuerelements Ihnen die Wahrheit sagt und außerdem auch in der Lage ist, ein sicheres Steuerelement zu erstellen.

Ihre erste Reaktion könnte sein »Oha, jetzt haben wir ein Problem!«. Aber überlegen Sie doch – stehen Sie nicht genau derselben Situation gegenüber, wenn Sie Software auf Ihrem System installieren? Woher wissen Sie, daß das Programm, das Sie von Lotus oder Borland oder IBM gekauft haben, Ihr System nicht beschädigt? (Ich erwähne Microsoft hier nicht; schließlich ist es Ihr System.) Sie wissen es, weil Sie ihnen und ihren Programmierern im Großen und Ganzen vertrauen. Sie wissen, daß Sie jemanden anrufen können, der Ihnen weiterhilft, falls ein Problem auftritt.

Die Software-Industrie baut auf Vertrauenswürdigkeit und Verantwortungsbeußsein auf. Ich möchte hier betonen, daß dieses Vertrauen durchaus auch auf viele Händler von Shareware oder Freeware ausgeweitet werden kann. Es gibt viele kleine Firmen, einzelne Software-Entwickler und Berater, die Software schreiben, die ebenso gut und zum Teil besser ist, als die großen kommerziellen Pakete. Aber nichtsdestotrotz gilt: Bevor Sie Software von solchen Entwicklern verwenden, sollten Sie wissen, wer sie sind, entweder individuell oder dem Ruf nach. Auch hier spielt Vertrauen die größte Rolle.

So lange Sie nur Steuerelemente von Sites herunterladen, denen Sie vertrauen, haben Sie keine Probleme. Sie können den Web-Autoren dieser Site vertrauen, daß sie Ihr System nicht über ein Steuerelement zerstören, das Sie heruntergeladen haben. Es ist dabei auch nicht so wichtig, ob das Steuerelement als sicher markiert ist, denn damit ein Steuerelement Unheil anrichten kann, muß es nicht nur böswillige Operationen erlauben, sondern der Web-Autor muß unter Verwendung von Eigenschaftswerten oder Skripts auch wirklich eine böswillige Operation für das Steuerelement programmieren.

Im Web gelangt man jedoch häufig fast zwangsläufig an Stellen, die man nicht kennt und denen man also auch nicht vertrauen sollte. Als Seiten-Autor sollten Sie immer davon ausgehen, daß Ihre Site für einige Ihrer Besucher ebenfalls in

diese Kategorie fällt. Als Benutzer können Sie die Seiten gefahrlos ansehen und Steuerelemente aus unbekannten Sites herunterladen, wenn Sie wissen, daß diese Steuerelemente als sicher markiert sind und aus einer Quelle stammen, der Sie vertrauen können.

Wem Sie vertrauen wollen, bleibt Ihnen überlassen. Die Code-Signatur stellt Ihnen nur einen Mechanismus bereit, sicherzustellen, daß die Quelle des Steuerelements wirklich das ist, was sie vorgibt zu sein. Wenn Sie beispielsweise ein Steuerelement erhalten, das in seiner Versionsinformation behauptet, daß es von Desaware entwickelt wurde, dann kann das vielleicht stimmen. Es könnte aber auch eine Fälschung sein – jemand hat vielleicht die Quellinformationen des Steuerelements geändert. Wenn das Steuerelement jedoch die Code-Signatur benutzt, um damit sicherzustellen, daß es von Desaware stammt, dann beweist ein digitales, in das Steuerelement eingebettetes Zertifikat, daß es zweifelsfrei wirklich von Desaware entwickelt und veröffentlicht wurde, und daß sich niemand daran zu schaffen gemacht hat. Dieses Zertifikat wird von zahlreichen Firmen verliehen, die sich darauf spezialisiert haben, Software-Entwickler mit verschlüsselten Zertifikaten zu versorgen (wie später in diesem Kapitel noch beschrieben wird). Ob Sie den Steuerelementen von Desaware vertrauen oder nicht, bleibt natürlich Ihre Entscheidung.

21.3.5 Veröffentlichen und Herunterladen von ActiveX-Controls

Jetzt haben Sie erfahren, daß es zwei separate Technologien gibt, eine, mit der Steuerelemente als sicher für die Initialisierung oder das Skripting markiert werden, und eine, mit der garantiert werden kann, daß die Steuerelemente wirklich von der Person oder Firma erzeugt wurden, wie vom Autor behauptet. Später werden wir genauer untersuchen, wie man Steuerelemente als sicher markiert und sie signiert. Für den Moment wollen wir betrachten, wie sich diese beiden Technologien in den Downloading-Prozeß aus der Perspektive des Endbenutzers, des Site-Entwicklers und des Steuerelement-Autors einfügen.

Sicherheit des Endbenutzers

Jeder Browser, der als Container für ActiveX-Steuerelemente dienen kann, verwendet unterschiedliche Sicherheitseinstellungen. Der Microsoft Internet Explorer, der momentan die meiste Unterstützung für ActiveX-Steuerelemente bietet, hat drei Einstellungen, die der Benutzer in Hinblick auf die Sicherheit konfigurieren kann:

- **HOCH** – Mit dieser Einstellung können nur Steuerelemente ausgeführt werden, die mit einem Zertifikat von einer der autorisierenden Firmen signiert sind, die der Benutzer anerkennt, und die von einer vom Benutzer anerkannten Firma entwickelt wurden. Es gibt ein eigenes Dialogfeld im Microsoft Explorer, wo Sie angeben können, welche zertifikat-ausstellende Firmen Sie akzeptieren wollen. Beim Herunterladen des Steuerelements wird das Zertifikat angezeigt. Sie können wählen, ob Sie das Steuerelement übernehmen möchten, und ob Sie

die Firma, die dieses Steuerelement entwickelt hat, in Ihre Liste der vertrauenswürdigen Steuerelement-Autoren aufnehmen möchten.

- MITTEL – Mit dieser Einstellung können unsichere und nicht signierte Steuerelemente heruntergeladen werden. Der Explorer zeigt ein Zertifikat an, das den Autor des Steuerelements angibt, und ebenso, ob das Steuerelement sicher zur Initialisierung und zum Skripting ist. Anschließend haben Sie die Möglichkeit, das Steuerelement auszuführen oder nicht.
- NIEDRIG – Mit dieser Einstellung werden Steuerelemente automatisch heruntergeladen und ausgeführt. Diese Einstellung wäre denkbar für ein Firmen-Intranet, das keinen Zugriff auf außenstehende Sites bietet. Andernfalls sollten Sie diese Einstellung nicht verwenden.

Der Internet Explorer bietet auch die Möglichkeit, die Sicherheitseinstellungen anzupassen und anzugeben, wie der Browser auf bestimmte Situationen reagieren soll.

Das digitale Zertifikat beinhaltet einen Hypertext-Link zu der Firma, die das Zertifikat ausgestellt hat. Damit können Sie prüfen, ob das Zertifikat noch gültig ist. Der Explorer ermöglicht Ihnen außerdem, zukünftige Warnungen für Komponenten von bestimmten Herstellern oder solchen, die digitale Zertifikate von bestimmten Autorisierungsfirmen haben, zu unterdrücken.

Der Site-Entwickler und/oder Steuerelement-Autor

Es ist trivial, einer Web-Seite ein automatisches Downloading hinzuzufügen. Sie brauchen dem <OBJECT>-Tag einfach nur das CODEBASE-Attribut hinzuzufügen:

```
<OBJECT ID="dwBanner1" WIDTH=324 HEIGHT=49
CLASSID="CLSID:63B8AF4F-8E9A-11D0-91BB-00AA0036005A"
CODEBASE="dwBanner.CAB#version=1,0,0,15">
  <PARAM NAME="_ExtentX" VALUE="8573">
  <PARAM NAME="_ExtentY" VALUE="1296">
  <PARAM NAME="BackColor" VALUE="16777088">
  <PARAM NAME="BorderStyle" VALUE="1">
  <PARAM NAME="ScrollIncrement" VALUE="3">
  <PARAM NAME="ScrollText" VALUE="Visit Desaware's Web site at
    http://www.desaware.com - The coolest tools
    for VB/VBA developers. ">
  <PARAM NAME="XMargin" VALUE="20">
  <PARAM NAME="YMargin" VALUE="6">
</OBJECT>
```

Das CODEBASE-Attribut teilt dem Browser den Namen der Datei mit, die ermittelt werden soll, und die das Objekt und seine Version enthält. Das schöne daran ist: wenn das System bereits ein Objekt mit der angegebenen CLSID enthält, mit der aktuellen Version oder einer neueren, verwendet der Browser automatisch das bereits existierende Objekt, statt eine neue Kopie davon herunterzuladen.

Die .CAB-Datei ist eine komprimierte Cabinet-Datei in einem von Microsoft vorgegebenen Format. Beachten Sie, daß Sie auch andere Dateitypen herunterladen können.

Sie könnten beispielsweise den Namen der .OCX-Datei direkt in der Zeile angeben. Das Problem dabei ist, daß dabei mögliche Abhängigkeiten nicht berücksichtigt werden. Für die korrekte Ausführung der meisten Steuerelemente (und alle VB-autorisierten Steuerelemente) sind zusätzliche DLLs erforderlich. Die .CAB-Datei enthält eine Datei mit Informationen zur Installation, mit der Dateinamenerweiterung .INF, die die Liste der Abhängigkeiten enthält, und ebenso, wo die zusätzlichen Dateien zu finden sind. Sie könnten die CODEBASE-Zeilenreferenz direkt in die .INF-Datei aufnehmen, aber Sie können eine .INF-Datei nicht signieren, der Benutzer könnte also die Quelle der Datei nicht überprüfen.

Die .CAB-Datei kann von einem Site-Entwickler erzeugt werden, wird aber in der Regel vom Steuerelement-Autor angelegt. Eine .CAB-Datei kann (und sollte) ihre eigene digitale Signatur haben, die angibt, ob das Herunterladen sicher ist. Sie sollten auch Ihre Steuerelemente signieren. Was zählt, ist, daß die in der CODE-BASE-Zeile angegebene Datei signiert ist.

Die meisten VB-Programmierer verwenden den Setup-Assistenten von Visual Basic oder ein Installationswerkzeug von einem Drittanbieter, um die .CAB-Dateien für ihre Steuerelemente zu erzeugen. Außerdem erstellen sie eine Schablone für das <OBJECT>-Tag für das Steuerelement, die den HTML-Seiten hinzugefügt werden kann.

Ich werde hier nicht detailliert auf die Verwendung des Setup-Assistenten von Visual Basic eingehen. Die Dokumentation von Visual Basic ist ausreichend, und der Assistent ist einfach in der Anwendung. Es gibt jedoch einige Aspekte, die Sie berücksichtigen sollten, wenn Sie den Assistenten einsetzen. Sie beziehen sich sowohl auf die Sicherheit als auch auf die Signatur eines Steuerelements und werden in den folgenden Abschnitten genauer beleuchtet.

Die .CAB-Auswahl

Die .INF-Datei in einer .CAB-Datei kann zusätzliche Dateien auflisten, die für die Ausführung Ihres Steuerelements erforderlich sind. Wenn Sie ein Steuerelement entwickeln, das über das Internet heruntergeladen werden kann, geben Sie in der Regel an, daß das Steuerelement von der Web-Site von Microsoft oder von Ihrer eigenen heruntergeladen werden kann. Ich empfehle Ihnen die von Microsoft. Warum sollten Sie die Bandbreite Ihres Servers durch den Download von den großen VB-Laufzeit-Support-Dateien belasten?

Sie können gegebenenfalls auch noch weitere Dateien in der .CAB-Datei angeben, aber das erhöht die Download-Zeit für die Datei. Dieser Ansatz ist in Intranets gebräuchlich, wo die Download-Zeiten weniger wichtig sind.

Der Internet Client SDK, den Sie direkt von Microsoft erhalten (sehen Sie auf der Web-Site nach) beinhaltet Werkzeuge zur Anpassung von Sites sowie eine detail-

lierte Erklärung des Dateiformats .INF. Sie können .CAB-Dateien erzeugen, die mehrere Betriebssysteme unterstützen. Die ursprüngliche .INF-Datei enthält eine Liste der .CAB- und Betriebssystemabhängigkeiten. Der Browser lädt dann die für das Zielsystem benötigte .CAB-Datei. Das führt jedoch dazu, daß zwei unterschiedliche Zertifikate angezeigt werden (eines für jede .CAB-Datei).

21.3.6 Steuerelemente als sicher markieren

Es gibt zwei Ansätze, ein Steuerelement als sicher zu markieren.

Der Registrierungs-Ansatz

Dieser Ansatz besteht darin, Einträge in der Registrierung vorzunehmen, die ein bestimmtes Steuerelement als sicher kennzeichnen. In Kapitel 6 (also vor einer ganzen Ewigkeit) haben Sie erfahren, wie eine ActiveX-DLL registriert wird. Weil ein ActiveX-Steuerelement einfach ein spezieller Typ von ActiveX-DLL ist, sollte nicht erstaunlich sein, daß es auf dieselbe Weise registriert wird. Unter der Klassen-ID Ihres Steuerelements gibt es einen Schlüssel, Implemented Categories, der zusätzliche Informationen über das Steuerelement enthält. Zwei neue Unterschlüssel wurden für diesen Schlüssel definiert, um den Sicherheitsstatus eines Steuerelements anzuzeigen. Wenn das Steuerelement sicher initialisiert werden kann, lautet dieser Unterschlüssel:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\your control's CLSID\
Implemented Categories\{7DD95802-9882-11CF-9FA9-00AA006C42C4}"
```

Wenn für das Steuerelement ein sicheres Skripting (Programmierung) möglich ist, lautet der Unterschlüssel:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\your control's CLSID\
Implemented Categories\{7DD95801-9882-11CF-9FA9-00AA006C42C4}"
```

Der Setup-Assistent von Visual Basic erlaubt Ihnen zu spezifizieren, daß Ihr Steuerelement sicher ist. Er nimmt Einträge in die .INF-Datei für das Steuerelement vor, das die angegebenen Sicherheitseinstellungen in die Registrierung eingetragen hat. Es muß wohl nicht gesagt werden, daß Sie diese Registrierungs-Einträge keinen Steuerelementen hinzufügen sollten, von denen Sie nicht genau wissen, ob sie sicher sind.

Der IObjectSafety-Ansatz

Der Registrierungs-Ansatz hat zwei Nachteile.

- Immer wenn geprüft werden muß, ob ein herunterzuladendes Steuerelement sicher ist, muß in der Registrierung nachgesehen werden.
- Steuerelemente, die diesen Ansatz nutzen, müssen immer sicher sein. Sie haben nicht die Möglichkeit, unsichere Operationen zu unterstützen, ohne das Vertrauen eines anderen Steuerelement-Autors zu verletzen, der seinen Code als sicher markiert.

`IObjecSafety` ist eine Standardschnittstelle, über die jedes Steuerelement dem System mitteilen kann, daß es sicher ist. Sie ermöglicht einem Steuerelement, sowohl sichere als auch unsichere Operationsmodi zu unterstützen, abhängig davon, wie der Container es gerade braucht.

Sie können `IObjecSafety` implementieren, indem Sie einen Verweis auf die `IObjecSafety`-Typbibliothek aufnehmen, die sich auf der MSDN-CD-ROM von Visual Studio im Beispiele-Verzeichnis VB98/`IOBJSafe` befindet, oder in dem entsprechenden Verzeichnis auf der CD mit den Beispielanwendungen von VB, falls Sie nur Visual Basic gekauft haben.

Wenn Sie diese Schnittstelle implementieren, sehen Sie zwei Funktionen, `GetInterfaceSafetyOptions` und `SetInterfaceSafetyOptions`.

Ein Container, der eine sichere Operation erzwingt, beispielsweise ein Internet-Browser, ruft die Funktion `SetInterfaceSafetyOptions` auf, um zu fordern, daß das Steuerelement in den sicheren Modus übergeht. Das erfolgt unmittelbar nach der Initialisierung des Steuerelements und vor dem `ReadProperties`-Ereignis. Der Parameter `dwOptionsSetMask` setzt sich aus Bits zusammen, die definieren, welche Art Sicherheit gefordert wird (sicher für die Initialisierung oder sicher für das Skripting). Der Parameter `dwEnableOptions` zeigt den gewünschten Sicherheitswert für jeden Typ an. Ein Container könnte beispielsweise fordern, daß Ihr Steuerelement nur für die Initialisierung sicher ist, dann verzichtet er auf die Sicherheit beim Skripting.

Die Funktion `GetInterfaceSafetyOptions` erlaubt dem Container, die aktuelle Einstellung des Steuerelements zu ermitteln. Der Parameter `pIID` wird von VB-Programmierern in der Regel nicht verwendet, wird aber für zukünftige Situationen bereitgestellt, wo Sie vielleicht möchten, daß das Steuerelement für bestimmte Schnittstellen sicher sein soll. Eine detailliertere Erklärung dieser Funktionen, ihrer Verwendung und möglicher Parameterwerte finden Sie in der Beispielanwendung `Iobjsafe` von Visual Basic. Eine einfachere Lösung, Ihrer Komponente die `IObjecSafety`-Schnittstelle hinzuzufügen, finden Sie in `Spy-Works` von Desaware.

Vom Standpunkt des Steuerelement-Autors aus gesehen ist wichtig, daß man den vom Container geforderten Sicherheitsstatus ermitteln und den aktuellen Status in einer Variablen auf Modulebene ablegen kann. Damit ist es möglich, gefährliche Operationen zu deaktivieren, wenn gefordert wird, daß das Steuerelement sicher sein muß. Das ist ideal für die Anpassung existierender Steuerelemente für die Verwendung im Internet und erspart Ihnen, eine Internet-sichere Version und eine zweite für die Verwendung in Applikationen vorzuhalten.

Als Nebeneffekt eliminiert dieser Ansatz zusätzlich erforderliche Zugriffe auf die Registrierung. Der Container lädt das Steuerelement. Wenn er sieht, daß das Steuerelement sicher ist, initialisiert er es und führt es auf der Web-Seite aus. Andernfalls beendet er das Steuerelement.

Steuerelemente signieren

Der Sinn einer Code-Signatur ist, zu beweisen, daß ein Steuerelement aus einer bestimmten Quelle stammt. Um zu verstehen, wie das funktioniert, brauchen Sie ein paar Informationen über die sogenannte Verschlüsselung mit öffentlichem Schlüssel.

Die Verschlüsselung erfolgt, indem Text unter Verwendung eines Codier-Algorithmus und eines Schlüssels durcheinandergewürfelt wird. Bei der normalen Verschlüsselung brauchen der Sender und der Empfänger denselben Schlüssel. Der Sender verschlüsselt die Nachricht unter Verwendung seiner Kopie des Schlüssels. Die Nachricht kann nur von jemandem gelesen werden, der denselben Schlüssel besitzt.

Die Verschlüsselung mit öffentlichem Schlüssel verwendet zwei unterschiedliche Schlüssel, einen privaten Schlüssel und einen öffentlichen Schlüssel. Wenn eine Person eine Nachricht mit ihrem privaten Schlüssel verschlüsselt, kann jeder, der den öffentlichen Schlüssel besitzt, diese Nachricht lesen. Nachrichten, die mit dem öffentlichen Schlüssel verschlüsselt sind, können nur von jemandem gelesen werden, der den privaten Schlüssel besitzt. Wie funktioniert das? Ich habe keine Ahnung. Bei der überwältigenden Geschwindigkeit, mit der sich die Technologie ständig ändert, muß man hier und da Mut zur Lücke besitzen. Für mich ist es momentan ausreichend zu wissen, daß die Verschlüsselung mit öffentlichem Schlüssel so funktioniert, wie man es von ihr behauptet. Weitere Informationen über dieses Thema finden Sie auf der Web-Site von Verisign (www.verisign.com).

Wie funktioniert die Verschlüsselung mit öffentlichem Schlüssel in diesem Fall? Gut, wenn Sie ein Dokument unter Verwendung eines privaten Schlüssels erzeugen, das nur mit einem öffentlichen Schlüssel entschlüsselt werden kann, beweist eine erfolgreiche Entschlüsselung mit dem öffentlichen Schlüssel, daß das Dokument mit Hilfe eines bestimmten privaten Schlüssels angelegt wurde. Wenn Sie den öffentlichen Schlüssel von Desaware verwenden, um ein Dokument zu entschlüsseln, das behauptet, von Desaware zu sein, und es wird erfolgreich entschlüsselt, wissen Sie, daß es wirklich von Desaware stammt. Dabei wird natürlich vorausgesetzt, daß Desaware seinen privaten Schlüssel geheim hält.

Jetzt werden wir die Schritte beschreiben, wie Sie und Ihre Endbenutzer die Signatur von ActiveX-Steuerelementen implementieren.

Ein Zertifikat erhalten

Als erstes wenden Sie sich an eine Firma, die digitale Zertifikate erteilt. Als Autor von VB-Steuerelementen werden Sie vermutlich Verisign Inc. (www.verisign.com) wählen. Wenn Sie dies lesen, hat Microsoft möglicherweise schon Abkommen mit anderen Firmen getroffen, deshalb sollten Sie als erstes die Web-Site von Microsoft besuchen und die neuesten Informationen über die Signatur von ActiveX-Steuerelementen lesen.

Egal an welche Organisation Sie sich gewandt haben, sie wird Ihnen eine Möglichkeit bereitstellen, wie Sie Ihr digitales Zertifikat online erhalten. Sie werden nach grundlegenden Identifizierungsinformationen gefragt. Verisign bietet zwei Arten von Zertifikaten an, eines für einzelne Software-Entwickler, und eines für Firmen. Das für die einzelnen Software-Entwickler kostet 20\$ pro Jahr (zum Zeitpunkt der Drucklegung dieses Buches). Das Firmen-Zertifikat kostet 400\$. Auf der Web-Site von Verisign werden die Schritte beschrieben, wie geprüft wird, wer Sie sind. Beispielsweise wird die Dunn&Bradstreet-Geschäftsnummer überprüft. Dabei soll sichergestellt werden, daß sich ein Endbenutzer bei einem Firmen-Zertifikat wirklich darauf verlassen kann, daß es sich bei der genannten Firma wirklich um eine Firma handelt.

Außerdem teilen Sie der Organisation neben den anderen Informationen auch ein privates Paßwort mit, das genutzt wird, um einen privaten und einen öffentlichen Schlüssel zu erzeugen. Der private Schlüssel wird in einer Datei mit der Erweiterung `.PVK` abgelegt. Sie müssen diesen privaten Schlüssel geheimhalten. Der öffentliche Schlüssel wird zu der ausstellenden Organisation geschickt.

Anschließend überprüft die ausstellende Organisation Ihre Applikation und nimmt bestimmte Schritte vor, um sicherzugehen, daß Sie auch der sind, der Sie zu sein behaupten. Sie erzeugt ein digitales Zertifikat, das Ihnen per E-Mail zugesickt wird. Dieses Zertifikat enthält Ihren öffentlichen Schlüssel, der wiederum unter Verwendung ihres privaten Schlüssels verschlüsselt ist. Das bedeutet, niemand kann den öffentlichen Schlüssel der Organisation dazu verwenden, den öffentlichen Schlüssel Ihrer Firma zu ermitteln.

Betrachten Sie das Beispiel des `dwBanner`-Steuerelements, das Sie gleich kennenlernen werden. Angenommen, Verisign ist die ausstellende Organisation und Sie haben ihren öffentlichen Schlüssel. (Der Microsoft Explorer hat diesen Schlüssel und nutzt ihn, um ActiveX-Steuerelemente zu überprüfen.) Wenn Sie ein von Verisign erteiltes Zertifikat erfolgreich entschlüsseln können, wissen Sie, daß es wirklich von dieser Firma stammt. Wenn Sie den öffentlichen Schlüssel von Desaware aus diesem Zertifikat ermitteln, haben Sie Verisigns Garantie, daß der Schlüssel von Desaware stammt, weil Verisign ihn von Desaware erhalten hat, und sich die Mühe gemacht hat zu überprüfen, daß die Firma wirklich die ist, die sie vorgibt zu sein. Jetzt haben Sie nicht nur einen gültigen Schlüssel von Desaware, sondern auch eine gewisse Garantie von Verisign in Hinblick auf die Legitimierung der Firma. Wenn Sie eine Firma, die das Screening von Verisign übergibt, für Software-Downloads als zuverlässig erachten, könnten Sie dem Explorer mitteilen, daß Sie automatisch alle Steuerelemente akzeptieren, die mit einem ihrer Zertifikate signiert sind.

Ein Steuerelement signieren

Als Autor erhalten Sie von Verisign ein digitales Zertifikat, das die Erweiterung `.SPC` hat. Um ein Steuerelement zu signieren, verwenden Sie das Programm `SignCode.exe`, das Sie von Microsoft erhalten, und das Teil des Internet Client SDK ist. Dieses Programm nimmt Ihren privaten Schlüssel und das Zertifikat ent-

gegen. Es führt zuerst einen Verschlüsselungsalgorithmus aus, um eine Zwischenversion der Datei zu erzeugen, die Sie signieren möchten. Daraus ergibt sich eine große Zahl, ähnlich einer Prüfsumme, außer daß die Wahrscheinlichkeit, daß zwei Dateien denselben Wert erzeugen, extrem gering ist. Dieser Wert wird dann mit Ihrem privaten Schlüssel verschlüsselt. Dieser Wert und die Zertifikat-Information werden der Datei hinzugefügt. Sie können .OCX-Dateien und .CAB-Dateien signieren.

Wir führen das dwBanner-Beispiel fort, um zu sehen, was mit einem signierten Steuerelement passiert, nachdem der Browser es heruntergeladen hat. Der Browser extrahiert das Zertifikat und sieht, daß es von Verisign ausgestellt wurde. Er entschlüsselt das Zertifikat mit dem öffentlichen Schlüssel von Verisign und erhält den öffentlichen Schlüssel von Desaware. Anschließend führt er denselben Verschlüsselungs-Algorithmus für die Datei aus, der auch während der Signierung ausgeführt wurde. Damit erhält er einen neuen Prüfwert, der mit dem im Zertifikat abgelegten Wert übereinstimmen muß. Um den im Zertifikat gespeicherten Schlüssel zu sehen, muß der Browser jedoch den öffentlichen Schlüssel von Desaware benutzen. Wenn der entschlüsselte Wert mit dem soeben durch den Browser erzeugten Wert übereinstimmt, wissen Sie mehrere Dinge:

- Daß das Steuerelement wirklich von Desaware entwickelt wurde
- Daß Desaware als Firma existiert, was die Bestätigung von Verisign garantiert
- Daß das Steuerelement nicht verändert wurde, seit Desaware es signiert hat

Dieser letzte Teil ist sehr wichtig. Wenn irgend jemand versucht, sich an einer Datei zu schaffen zu machen, unterscheidet sich der vom Browser für die Datei ermittelte Wert von dem im Zertifikat abgelegten Wert. Diese Verwendung der Code-Signatur hat nichts mit dem Internet zu tun. Sie können beliebige Steuerelemente signieren und mit dem Programm ChkTrust.exe (auch Teil des Microsoft ActiveX SDK) ausführen, um sicherzustellen, daß ein Steuerelement nicht verändert wurde.

Fazit

Hier folgt eine kurze Zusammenfassung der Schritte, die erforderlich sind, um Ihre Steuerelemente zu signieren:

- Lesen Sie unter www.microsoft.com die Liste der Organisationen nach, die Zertifikate für ActiveX-Steuerelemente ausstellen. Sie sollten unbedingt ein Zertifikat für ActiveX-Steuerelemente anfordern – es gibt noch viele andere Arten von Zertifikaten und digitalen Signaturen!
- Fordern Sie auf der Web-Site der betreffenden Organisation ein Zertifikat an. Sie erhalten eine .SPC-Datei, die den privaten Schlüssel enthält. Sie erhalten eine .SPC-Datei mit dem Zertifikat, das Sie mit dem privaten Schlüssel aus der .PVK-Datei entschlüsseln, die Sie während des Applikationsprozesses erzeugt haben.

- Installieren Sie das Programm `SignCode.exe` von Microsoft (im ActiveX SDK).
- Signieren Sie Ihre `.OCX`-Datei und die `.CAB`-Datei, die mit dem Setup-Assistenten von Visual Basic oder einem anderen Installationswerkzeug erzeugt werden, mit dem Programm `SignCode`.

21.4 Das Steuerelement *dwBanner*

Das Steuerelement *dwBanner* wurde insbesondere für das Internet entwickelt. Es handelt sich bei dieser Beispielapplikation um eine einfache rollende Titelleiste oder ein Marquee, eine frühe Ausgabe des *dwBanner*-Steuerelements, das (mit vollständigem Quellcode) in ActiveX Gallimaufry von Desaware enthalten ist. Gallimaufry ist eine neue Sammlung VB-erzeugter Steuerelemente mit Quellcode, die sowohl praktisch als auch lehrreich sind. Das Steuerelement ist, soweit ich weiß, sicher zu initialisieren und sicher für das Skripting. Die kompilierte Version auf Ihrer CD-ROM ist signiert. Beachten Sie, daß jede Version, die Sie selbst kompilieren, nicht mit dem Zertifikat von Desaware signiert ist.

Es gibt zwei HTML-Seiten in den Beispielverzeichnissen, die Sie nutzen können, um das Steuerelement zu laden. Die Datei `dwBanner.htm` im Beispielverzeichnis `ch21` zeigt das Steuerelement auf einer Web-Seite, wenn Sie bereits eine Version des Steuerelements registriert haben.

Sie können das Steuerelement mit Hilfe des Programms `regsvr32` registrieren, aber bei dieser Vorgehensweise wird das Steuerelement nicht als sicher markiert. Die Datei `dwBanner.htm` im Verzeichnis `ch21\swsetup` installiert und registriert das Steuerelement unter Verwendung der `.CAB`-Datei in diesem Verzeichnis. Diese Datei ist signiert und markiert das Steuerelement in der Registrierung als sicher.

Das Steuerelement *dwBanner* demonstriert viele der Techniken, die Sie in früheren Kapiteln bereits kennengelernt haben, und zeigt die Schritte, die Sie unternehmen sollten, wenn Sie ein kommerzielles Steuerelement entwickeln.

In diesem Abschnitt werden wir den Code schrittweise durchlaufen. Die charakteristischen Eigenschaften des Steuerelements werden offensichtlich, wenn Sie die Code-Beschreibung lesen. Beachten Sie jedoch, daß dieses Steuerelement einige Windows-API-Techniken benutzt. Alle hier verwendeten API-Funktionen sind in meinem Buch *Dan Appleman's Visual Basic 5.0 Programmer's Guide to the Win32 API* beschrieben, deshalb werde ich sie hier nicht im Detail erklären.

Die API-Deklarationen findet man im Standardmodul `dwBanner.bas`, das in Listing 21.1 gezeigt ist.

```
' Desaware's ActiveX Gallimaufry
' Einfaches Steuerelement für rollende Titelleisten
' Copyright (c) 1997 by Desaware Inc. All Rights Reserved
```

```
Option Explicit

Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

' Format-Flags für DrawText()
Public Const DT_TOP = &H0
Public Const DT_LEFT = &H0
Public Const DT_CENTER = &H1
Public Const DT_RIGHT = &H2
Public Const DT_VCENTER = &H4
Public Const DT_BOTTOM = &H8
Public Const DT_WORDBREAK = &H10
Public Const DT_SINGLELINE = &H20
Public Const DT_EXPANDTABS = &H40
Public Const DT_TABSTOP = &H80
Public Const DT_NOCLIP = &H100
Public Const DT_EXTERNALLEADING = &H200
Public Const DT_CALCRECT = &H400
Public Const DT_NOPREFIX = &H800
Public Const DT_INTERNAL = &H1000

' API-Funktionen
Declare Function GetClientRect Lib "user32" (ByVal hWnd As Long,
lpRect _
    As RECT) As Long
Declare Function CreateRectRgnIndirect Lib "gdi32" (lpRect As RECT)
As Long
Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long)
As Long
Declare Function DrawText Lib "user32" Alias "DrawTextA" (ByVal hdc
As _
    Long, ByVal lpStr As String, ByVal nCount As Long, lpRect As RECT,
ByVal _
    wFormat As Long) As Long
Declare Function SelectClipRgn Lib "gdi32" (ByVal hdc As Long,
ByVal hRgn _
    As Long) As Long
```

Listing. 21.1: Datei dwBanner.bas

Die Funktionen werden bei ihrem Auftreten kurz beschrieben. Listing 21.2 listet die Variablendeklarationen für das Steuerelement auf.

```
dwBanner.ct1

' Desaware's ActiveX Gallimaufry
' Einfaches Steuerelement für rollende Titelleisten
' Copyright (c) 1997 by Desaware Inc. All Rights Reserved

' dwBanner.ocx
' Desaware's ActiveX Gallimaufry
' Copyright © 1997 by Desaware Inc. All Rights Reserved

Option Explicit
'Standard-Eigenschaftswerte:
Const m_def_ScrollEnabled = True
Const m_def_ScrollIncrement = 2
Const m_def_AutoSizeFont = True

'Eigenschaftsvariablen:
Dim m_ScrollEnabled As Boolean
Dim m_ScrollText As String
Dim m_ScrollIncrement As Integer
Dim m_XMargin As Integer ' Mindestwert horizontaler Rand
Dim m_YMargin As Integer ' Mindestwert vertikaler Rand
Dim m_ReplaceText As String ' Anzeigestring zur Laufzeit
Dim m_AutoSizeFont As Boolean ' Automatisches Anpassen der Schrift

' Internal variables
Dim m_TextClippingRegion As Long ' Clipping-Bereich für den Text
Dim m_TextUpper As Integer ' Obere Position für die Textausgabe
Dim m_TextTotalWidth As Long ' Gesamtlänge des Texts
Dim m_CurrentXOffset As Long ' Aktueller Offset nach links für die An-
zeige
Dim Cliprc As RECT
Dim m_Initializing As Boolean ' Eigenschaften werden gelesen

Public Enum dwBorderStyle
    dwNone = 0
    dwSingle = 1
End Enum

'Ereignisdeklarationen:
Event Click() 'MappingInfo=UserControl,UserControl,-1,Click
Event Db1Click() 'MappingInfo=UserControl,UserControl,-1,Db1Click
Event MouseDown(Button As Integer, Shift As Integer, X As Single, _
Y As Single)
'MappingInfo=UserControl,UserControl,-1,MouseDown
```

```
Event MouseMove(Button As Integer, Shift As Integer, X As Single, _
Y As Single)
'MappingInfo=UserControl,UserControl,-1,MouseMove
Event MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
'MappingInfo=UserControl,UserControl,-1,MouseUp
Event CycleComplete() ' Indicates that a cycle has completed

'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!
'MappingInfo=UserControl,UserControl,-1,BackColor
Public Property Get BackColor() As OLE_COLOR
    BackColor = UserControl.BackColor
End Property

Public Property Let BackColor(ByVal New_BackColor As OLE_COLOR)
    UserControl.BackColor() = New_BackColor
    PropertyChanged "BackColor"
    UserControl.Refresh
End Property

'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!
'MappingInfo=UserControl,UserControl,-1,ForeColor
Public Property Get ForeColor() As OLE_COLOR
    ForeColor = UserControl.ForeColor
End Property

Public Property Let ForeColor(ByVal New_ForeColor As OLE_COLOR)
    UserControl.ForeColor() = New_ForeColor
    PropertyChanged "ForeColor"
    UserControl.Refresh
End Property

'ACHTUNG! DIE FOLGENDEN AUSKOMMENTIERTEN ZEILEN NICHT VERSCHIEBEN
'ODER VERÄNDERN!!!
'MappingInfo=UserControl,UserControl,-1,Enabled
Public Property Get Enabled() As Boolean
    Enabled = UserControl.Enabled
End Property

Public Property Let Enabled(ByVal New_Enabled As Boolean)
    UserControl.Enabled() = New_Enabled
    PropertyChanged "Enabled"
End Property

'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
```

```
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!
'MappingInfo=UserControl,UserControl,-1,Font
Public Property Get Font() As Font
    If Ambient.UserMode Then
        Err.Raise 393
    End If
    Set Font = UserControl.Font
End Property

Public Property Set Font(ByVal New_Font As Font)
    If (Not m_Initializing) And Ambient.UserMode Then
        Err.Raise 382
    End If
    Set UserControl.Font = New_Font
    PropertyChanged "Font"
    If Not m_Initializing Then CalculateTextMetrics
End Property

' Diese Version unterstützt keine Transparenz
'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!
'MappingInfo=UserControl,UserControl,-1,BackStyle
'Public Property Get BackStyle() As Integer
'    BackStyle = UserControl.BackStyle
'End Property

'Public Property Let BackStyle(ByVal New_BackStyle As Integer)
'    UserControl.BackStyle() = New_BackStyle
'    PropertyChanged "BackStyle"
'End Property

'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!
'MappingInfo=UserControl,UserControl,-1,BorderStyle

Public Property Get BorderStyle() As dwBorderStyle
    BorderStyle = UserControl.BorderStyle
End Property

Public Property Let BorderStyle(ByVal New_BorderStyle As dwBorderStyle)
    If New_BorderStyle > 1 Or New_BorderStyle < 0 Then
        Err.Raise 380
    End If
    UserControl.BorderStyle() = New_BorderStyle
    PropertyChanged "BorderStyle"
    If Not m_Initializing Then CalculateTextMetrics
End Property
```



```
Public Property Get FontUnderline() As Boolean
    FontUnderline = UserControl.FontUnderline
End Property

Public Property Let FontUnderline(ByVal New_FontUnderline As Boolean)
    UserControl.FontUnderline() = New_FontUnderline
    PropertyChanged "FontUnderline"
    If Not m_Initializing Then CalculateTextMetrics
End Property

'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!
'MappingInfo=UserControl,UserControl,-1,FontSize
Public Property Get FontSize() As Single
    FontSize = UserControl.FontSize
End Property

Public Property Let FontSize(ByVal New_FontSize As Single)
    UserControl.FontSize() = New_FontSize
    PropertyChanged "FontSize"
    If Not m_Initializing Then CalculateTextMetrics
End Property

'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!
'MappingInfo=UserControl,UserControl,-1,FontName
Public Property Get FontName() As String
    FontName = UserControl.FontName
End Property

Public Property Let FontName(ByVal New_FontName As String)
    UserControl.FontName() = New_FontName
    PropertyChanged "FontName"
    If Not m_Initializing Then CalculateTextMetrics
End Property

'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!
'MappingInfo=UserControl,UserControl,-1,FontItalic
Public Property Get FontItalic() As Boolean
    FontItalic = UserControl.FontItalic
End Property

Public Property Let FontItalic(ByVal New_FontItalic As Boolean)
    UserControl.FontItalic() = New_FontItalic
    PropertyChanged "FontItalic"
    If Not m_Initializing Then CalculateTextMetrics
End Property
```

```
'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!
'MappingInfo=UserControl,UserControl,-1,FontBold
Public Property Get FontBold() As Boolean
    FontBold = UserControl.FontBold
End Property

Public Property Let FontBold(ByVal New_FontBold As Boolean)
    UserControl.FontBold() = New_FontBold
    PropertyChanged "FontBold"
    If Not m_Initializing Then CalculateTextMetrics
End Property

Public Property Get XMargin() As Integer
    XMargin = m_XMargin
End Property

Public Property Let XMargin(ByVal vNewValue As Integer)
    If vNewValue >= UserControl.ScaleWidth Then
        ' Invalid property value
        Err.Raise 380
    End If
    m_XMargin = vNewValue
    PropertyChanged "XMargin"
    If Not m_Initializing Then CalculateTextMetrics
End Property

Public Property Get YMargin() As Integer
    YMargin = m_YMargin
End Property

Public Property Let YMargin(ByVal vNewValue As Integer)
    If vNewValue >= UserControl.ScaleHeight Then
        ' Invalid property value
        Err.Raise 380
    End If
    m_YMargin = vNewValue
    PropertyChanged "YMargin"
    If Not m_Initializing Then CalculateTextMetrics
End Property

'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!
'MappingInfo=Timer1,Timer1,-1,Interval
Public Property Get Interval() As Long
    Interval = Timer1.Interval
End Property
```

```
Public Property Let Interval(ByVal New_Interval As Long)
    Timer1.Interval() = New_Interval
    PropertyChanged "Interval"
End Property

Public Property Get ScrollEnabled() As Boolean
    ScrollEnabled = m_ScrollEnabled
End Property

Public Property Let ScrollEnabled(ByVal New_ScrollEnabled As Boolean)
    m_ScrollEnabled = New_ScrollEnabled
    PropertyChanged "ScrollEnabled"
    If Ambient.UserMode Then Timer1.Enabled = ScrollEnabled
End Property

Public Property Get ScrollIncrement() As Integer
    ScrollIncrement = m_ScrollIncrement
End Property

Public Property Let ScrollIncrement(ByVal vNewValue As Integer)
    If vNewValue > 20 Then
        Err.Raise 380
    End If
    m_ScrollIncrement = vNewValue
    PropertyChanged "ScrollIncrement"
End Property

Public Property Get ScrollText() As String
    ScrollText = m_ScrollText
End Property

Public Property Let ScrollText(ByVal New_Text As String)
    m_ScrollText = New_Text & " "
    PropertyChanged "Text"
    If Not m_Initializing Then CalculateTextMetrics
    UserControl.Refresh
End Property

Public Property Get ReplaceText() As String
    If Not Ambient.UserMode Then
        Err.Raise 393
    End If
    ReplaceText = m_ReplaceText
End Property
```

```

Public Property Let ReplaceText(ByVal New_Text As String)
    If Not Ambient.UserMode Then
        Err.Raise 382
    End If
    m_ReplaceText = New_Text
    PropertyChanged "ReplaceText"
End Property

Public Property Get AutoSizeFont() As Boolean
    AutoSizeFont = m_AutoSizeFont
End Property

Public Property Let AutoSizeFont(ByVal vNewValue As Boolean)
    m_AutoSizeFont = vNewValue
    PropertyChanged "AutoSizeFont"
    If Not m_Initializing Then CalculateTextMetrics
End Property

'Eigenschaften für die Benutzersteuerung initialisieren
Private Sub UserControl_InitProperties()
    m_ScrollEnabled = m_def_ScrollEnabled
    m_ScrollIncrement = m_def_ScrollIncrement
    m_AutoSizeFont = m_def_AutoSizeFont
    On Error GoTo InitNoExtender
    ' Standardtext auf den vom Benutzer zugewiesenen Namen setzen
    m_ScrollText = Ambient.DisplayName & " "
    'Set Font = Ambient.Font
    PropertyChanged "ScrollText"
InitNoExtender:
End Sub

'Eigenschaftswerte aus dem Speicher laden
Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
    m_Initializing = True
    UserControl.BackColor = PropBag.ReadProperty("BackColor", &H8000000F)
    UserControl.ForeColor = PropBag.ReadProperty("ForeColor", &H80000012)
    UserControl.Enabled = PropBag.ReadProperty("Enabled", True)
    'Set Font = PropBag.ReadProperty("Font")
    FontName = PropBag.ReadProperty("FontName", "Arial")
    FontSize = PropBag.ReadProperty("FontSize", 8)
    FontBold = PropBag.ReadProperty("FontBold", False)
    FontItalic = PropBag.ReadProperty("FontItalic", False)
    ' UserControl.BackStyle = PropBag.ReadProperty("BackStyle", 1)
    UserControl.BorderStyle = PropBag.ReadProperty("BorderStyle", 0)
    Timer1.Interval = PropBag.ReadProperty("Interval", 100)
    m_ScrollEnabled = PropBag.ReadProperty("ScrollEnabled", _
    m_def_ScrollEnabled)

```

```

m_ScrollIncrement = PropBag.ReadProperty("ScrollIncrement", _
m_def_ScrollIncrement)
m_ScrollText = PropBag.ReadProperty("ScrollText", "")
m_XMargin = PropBag.ReadProperty("XMargin", 0)
m_YMargin = PropBag.ReadProperty("YMargin", 0)
m_AutoSizeFont = PropBag.ReadProperty("AutoSizeFont", True)
m_Initializing = False
End Sub

'Eigenschaftswerte in den Speicher schreiben
Private Sub UserControl_WriteProperties(PropBag As PropertyBag)

    Call PropBag.WriteProperty("BackColor", UserControl.BackColor, &H8000000F)
    Call PropBag.WriteProperty("ForeColor", UserControl.ForeColor, &H80000012)
    Call PropBag.WriteProperty("Enabled", UserControl.Enabled, True)
    'Call PropBag.WriteProperty("Font", Font)
    Call PropBag.WriteProperty("FontName", FontName, "Arial")
    Call PropBag.WriteProperty("FontSize", FontSize, 8)
    Call PropBag.WriteProperty("FontBold", FontBold, False)
    Call PropBag.WriteProperty("FontItalic", FontItalic, False)
    ' Call PropBag.WriteProperty("BackStyle", UserControl.BackStyle, 1)
    Call PropBag.WriteProperty("BorderStyle", UserControl.BorderStyle, 0)
    Call PropBag.WriteProperty("Interval", Timer1.Interval, 100)
    Call PropBag.WriteProperty("ScrollEnabled", m_ScrollEnabled, _
m_def_ScrollEnabled)
    Call PropBag.WriteProperty("ScrollIncrement", m_ScrollIncrement, _
m_def_ScrollIncrement)
    Call PropBag.WriteProperty("ScrollText", m_ScrollText, "")
    Call PropBag.WriteProperty("XMargin", m_XMargin, 0)
    Call PropBag.WriteProperty("YMargin", m_YMargin, 0)
    Call PropBag.WriteProperty("AutoSizeFont", m_AutoSizeFont, True)
End Sub

```

Listing. 21.2: Variablen-, Eigenschafts- und Ereignisdeklarationen aus der Datei

Nun wollen wir die Eigenschaften des Steuerelements genauer betrachten. Die folgenden Eigenschaften stellen einfache Zuweisungen von Standardeigenschaften für das UserControl-Objekt dar:

- BackColor
- ForeColor
- Enabled
- BorderStyle

Ihre Implementierung sollte Ihnen bereits vertraut sein.

Die `Font`-Eigenschaft wird etwas anders behandelt, als Sie es vielleicht gewohnt sind. Die Standardschrift ist die Schrift, die der Autor des Steuerelements zur Entwurfszeit für das `UserControl`-Objekt festgesetzt hat. Die in der Umgebung festgelegte Schrift wird in diesem Fall überhaupt nicht verwendet. Einzelne Einstellungen für das Aussehen der Schrift werden als `FontBold`, `FontItalic` und `FontSize` ausgedrückt. Sie werden als separate Eigenschaften angegeben, um die Programmierung des Steuerelements einfacher zu machen – sie erscheinen als Eigenschaften des Steuerelements selbst, statt als Eigenschaften eines internen Objekts. Dadurch werden auch Fehler mit früheren Versionen des ActiveX Control Pad vermieden, das nicht in der Lage war, `Font`-Objekte in einer HTML-Seite zu speichern.

Der Entwickler hat zur Entwurfszeit Zugriff auf das `Font`-Objekt, er kann also die Einstellungen zentral im Dialogfeld `SCHRIFTART` ändern. Zur Laufzeit kann auf das `Font`-Objekt nicht zugegriffen werden. Wenn Sie dies zur Laufzeit vornehmen möchten, bräuchten Sie ein internes `StdFont`-Objekt und müßten das `Font-Changed`-Ereignis erkennen, wie in früheren Beispielen dieses Buchs bereits gezeigt. Das muß so sein, weil das Steuerelement unmittelbar auf Änderungen der Schrift reagieren muß. Änderungen der Schrift rufen die folgende Zeile auf:

```
If Not m_Initializing Then CalculateTextMetrics
```

Die Funktion `CalculateTextMetric` berechnet die Größe der Schrift und die Ränder, um sicherzugehen, daß der Text korrekt angezeigt wird. Außerdem startet sie die `Scrolling`-Operation neu. Wenn versucht wird, eine Schriftgröße zu setzen, die zu groß für die Anzeige ist, wird die Schriftgröße automatisch verkleinert. In so einer Situation wird kein Fehler aufgeworfen.

Die folgenden Eigenschaften beziehen sich auf die Operation des Banner-Steuerelements:

- `Interval` – bestimmt das Zeitintervall zwischen `Scrolling`-Operationen. Ist der `Interval`-Eigenschaft eines zugehörigen `Timer`-Steuerelements zugeordnet.
- `ScrollIncrement` – Legt fest, um wie viele Pixel der Text nach jedem Zeitintervall nach links gescrollt wird.
- `XMargin` – Der doppelte Rand zwischen der linken und der rechten Kante des Steuerelements und wo der Text erscheint.
- `YMargin` – Der doppelte Rand zwischen der oberen und der unteren Kante des Steuerelements und wo der Text erscheint.
- `ScrollEnabled` – Wird genutzt, um die `Scrolling`-Operation zu starten oder zu beenden. Ist der `Enabled`-Eigenschaft eines zugehörigen `Timer`-Steuerelements zugeordnet.
- `ScrollText` – Der zu scrollende Text. Der Text wird fortlaufend gescrollt. Wenn die Eigenschaft gesetzt wird, wird dem Text automatisch ein zusätzli-

ches Leerzeichen hinzugefügt, um sicherzustellen, daß zwischen dem Ende des Strings und dem neuen Anfang ein Abstand liegt. Sie können zusätzliche Strings hinzufügen, um den Abstand zu vergrößern.

- **ReplaceText** – Ein neuer String, der den aktuell gescrollten String ersetzt. Wenn diese Eigenschaft gesetzt wird, wird der aktuelle String vollständig aus dem Banner geschoben und der `ScrollText`-Eigenschaft wird automatisch der `ReplaceText`-String zugewiesen.
- **AutoSizeFont** – Wenn diese Eigenschaft gleich `True` ist, wird die Schriftgröße automatisch so angepaßt, daß sie die im Steuerelement verfügbare Höhe ausfüllt (die Gesamthöhe des Steuerelements minus dem Wert von `YMargin`).

Listing 21.3 zeigt, wie die verschiedenen Ereignisse und Methoden für das Steuerelement implementiert werden.

'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!

```
'MappingInfo=UserControl,UserControl,-1,Refresh
Public Sub Refresh()
    UserControl.Refresh
End Sub

Private Sub Timer1_Timer()
    Call DrawTheText(UserControl.hdc)
End Sub

Private Sub UserControl_Click()
    On Error GoTo NoHyper
    UserControl.Hyperlink.NavigateTo "http://www.desaware.com"
NoHyper:
    RaiseEvent Click
End Sub

Private Sub UserControl_DblClick()
    RaiseEvent DblClick
End Sub

Private Sub UserControl_MouseDown(Button As Integer, Shift As Integer, X _
As Single, Y As Single)
    RaiseEvent MouseDown(Button, Shift, X, Y)
End Sub

Private Sub UserControl_MouseMove(Button As Integer, Shift As Integer, X _
As Single, Y As Single)
    RaiseEvent MouseMove(Button, Shift, X, Y)
End Sub
```

```

Private Sub UserControl_MouseUp(Button As Integer, Shift As Integer, X _
As Single, Y As Single)
    RaiseEvent MouseUp(Button, Shift, X, Y)
End Sub

'ACHTUNG! DIE IM FOLGENDEN AUSKOMMENTIERTEN ZEILEN DÜRFEN
'NICHT ENTFERNT ODER VERÄNDERT WERDEN!
'MappingInfo=UserControl,UserControl,-1,hWnd
Public Property Get hWnd() As Long
    hWnd = UserControl.hWnd
End Property

Private Sub UserControl_Resize()
    CalculateTextMetrics
End Sub

Private Sub UserControl_Paint()
    Call DrawTheText(UserControl.hdc)
End Sub

Private Sub UserControl_Show()
    CalculateTextMetrics
End Sub

```

*Listing. 21.3: Ereignisse und Methoden des Steuerelements *dwBanner**

Die meisten Ereignisse und Methoden sind ganz einfach. Refresh bezieht sich direkt auf das UserControl-Objekt. Das Paint-Ereignis und das Timer-Ereignis rufen die Funktion DrawTheText auf. Diese Funktion nimmt einen Gerätekontext als Parameter entgegen, um zukünftige Erweiterungen zu erlauben. Die Funktion CalculateTextMetrics wird zum ersten Mal aufgerufen, nachdem das Steuerelement-Fenster während des Show-Ereignisses auf dem Container platziert wurde. Damit wird sichergestellt, daß die fensterabhängigen Berechnungen korrekt sind.

Das Click-Ereignis bringt den Browser zur Web-Site von Desaware, wenn das Steuerelement in einem Browser enthalten ist. Diese Zeile wurde nur für Demonstrationszwecke aufgenommen. In der kommerziellen Version dieses Steuerelements wird das Click-Ereignis ganz normal ausgelöst. Der Entwickler kann dem Click-Ereignis in seiner Skripting-Sprache beliebige Operationen zuweisen. Listing 21.4 zeigt die Funktionen, die das Scrolling implementieren.

```

'-----
'
' Private Funktionen, die intern genutzt werden
' Berechnung der Variablen zur Textausgabe
Private Sub CalculateTextMetrics()
    Dim TxtHeight As Integer

```



```

InitializeClippingRegion
If m_TextClippingRegion = 0 Then
    ' Text kann nicht gezeichnet werden; Beenden
    Exit Sub
End If
TxtHeight = UserControl.TextHeight(m_ScrollText)
If TxtHeight > Cliprc.Bottom - Cliprc.Top Or m_AutoSizeFont Then
    ' Schrift zur Laufzeit zu groß -> verkleinern
    If Ambient.UserMode Then Call AdjustFontSize
    TxtHeight = UserControl.TextHeight(m_ScrollText)
End If
m_TextUpper = Cliprc.Top + (Cliprc.Bottom - Cliprc.Top - TxtHeight) \ 2

m_TextTotalWidth = UserControl.TextWidth(m_ScrollText)
If Ambient.UserMode Then
    m_CurrentXOffset = -(Cliprc.Right - Cliprc.Left)
    ' Start off to the right
    Timer1.Enabled = m_ScrollEnabled
End If
UserControl.Refresh
End Sub

' Clipping-Bereich initialisieren
Private Sub InitializeClippingRegion()
    If m_TextClippingRegion <> 0 Then
        ' Existierenden Bereich löschen
        Call DeleteObject(m_TextClippingRegion)
    End If
    Call GetClientRect(UserControl.hWnd, Cliprc)
    ' Rechteck verkleinern
    Cliprc.Left = Cliprc.Left + m_Xmargin \ 2
    Cliprc.Right = Cliprc.Right - m_Xmargin \ 2
    Cliprc.Top = Cliprc.Top + m_Ymargin \ 2
    Cliprc.Bottom = Cliprc.Bottom - m_Ymargin \ 2
    If Cliprc.Right > Cliprc.Left And Cliprc.Bottom > Cliprc.Top Then
        ' Zeichenbereich ist erlaubt
        m_TextClippingRegion = CreateRectRgnIndirect(Cliprc)
    End If
End Sub

' Schriftgröße an die maximal erlaubte Höhe anpassen
' Wenn RetrieveOnly True ist, wird nur die größtmögliche Schriftgröße
' zurückgegeben
Private Function AdjustFontSize(Optional ByVal RetrieveOnly As _
Boolean = False)

```

```

Dim targetheight&
Dim targetpoints&
targetheight = (Cliprc.Bottom - Cliprc.Top - 1) * Screen.TwipsPerPixelY
targetpoints = targetheight / 20
UserControl.Font.Size = targetpoints
If targetpoints < 8 Then
    ' Windows-Schriftauswahl-Trick
    UserControl.Font.Name = UserControl.Font.Name
    UserControl.Font.Size = targetpoints
End If

End Function

' Verwendet den hdc, weil zukünftige Versionen Transparenz
' erlauben
Private Function DrawTheText(ByVal hdc As Long) As Long
    Dim drawrc As RECT
    Static ReplaceMarker As Long
    Static ReplacePending As Boolean
    If m_TextClippingRegion = 0 Then Exit Function
    If m_TextTotalWidth = 0 Then
        Exit Function
    End If
    Call SelectClipRgn(hdc, m_TextClippingRegion)
    Call GetClientRect(UserControl.hWnd, drawrc)
    drawrc.Top = m_TextUpper
    drawrc.Left = Cliprc.Left - m_CurrentXOffset
    Do
        Call DrawText(hdc, m_ScrollText, -1, drawrc, DT_SINGLELINE _
            Or DT_NOPREFIX)
        If Not Ambient.UserMode Then Exit Do
        ' Schleife beenden, wenn der Text unterwegs ist
        drawrc.Left = drawrc.Left + m_TextTotalWidth
        If ReplacePending And drawrc.Left > ReplaceMarker Then
            ' Sehr viele Leerzeichen anfügen, um Resttexte zu löschen
            Call DrawText(hdc, Space$(Len(m_ScrollText)), -1, drawrc, _
                DT_SINGLELINE Or DT_NOPREFIX)
            Exit Do ' Es steht ein Ersatz an
        End If
    Loop While drawrc.Left < drawrc.Right
    If m_ReplaceText <> "" And Not ReplacePending Then
        ' Ersatztext initialisieren
        ReplaceMarker = drawrc.Left - 1
        ReplacePending = True
    End If
    Call SelectClipRgn(hdc, 0) ' Clipping-Bereich löschen
    If Ambient.UserMode Then

```

```
' Das Scrolling erfolgt nur zur Laufzeit
m_CurrentXOffset = m_CurrentXOffset + m_ScrollIncrement
If ReplacePending Then ReplaceMarker = ReplaceMarker - _
m_ScrollIncrement
If m_CurrentXOffset > m_TextTotalWidth Then
    m_CurrentXOffset = 0
    RaiseEvent CycleComplete
    If ReplacePending And ReplaceMarker <= Cliprc.Left + _
        m_ScrollIncrement Then
        ReplaceMarker = 0
        ReplacePending = False
        m_ScrollText = m_ReplaceText
        m_ReplaceText = ""
        CalculateTextMetrics
    End If
End If
End If
End Function

Public Sub AboutBox()
    On Error GoTo AlreadyVisible
    frmAboutBox.Show vbModal
    Set frmAboutBox = Nothing
AlreadyVisible:
End Sub
```

Listing. 21.4: Der Code für die Implementierung des Steuerelements

Die Funktion `CalculateTextMetrics` initialisiert zunächst den Clipping-Bereich für das Steuerelement. Ein Clipping-Bereich maskiert den Teil des Steuerelements, in den wirklich gezeichnet werden kann. Dieser Mechanismus bewirkt, daß die Ränder in Kraft treten. Der Clipping-Bereich wird auf den Bereich gesetzt, in dem der Text erscheinen soll. Die Zeichenfunktion zeichnet den Text auf den gesamten Bereich des Steuerelements, wobei es Windows überlassen wird, den Text abzuschneiden, der außerhalb des Clipping-Bereichs liegt.

Anschließend wird, falls nötig, mit der Funktion `AdjustFontSize`, die Schriftgröße angepaßt. Diese Funktion berechnet die gewünschte Schriftgröße basierend auf der aktuellen Bildschirmauflösung, die in Twips angegeben ist. Es gibt 1440 Twips pro Inch, und 72 Twips pro Punkt. Ist die Schrift größer als 8 Punkte, wird die Größe übergeben und der Schriftname zusammen mit der Größe gesetzt. Dies ist ein Trick, der Windows hilft, die bestmögliche Schrift für kleinere Größen zu wählen.

Die Gesamtbreite des gescrollten Texts wird berechnet, und die Anfangsposition für den Text wird ganz rechts im Steuerelement gesetzt.

Die eigentliche Arbeit erledigt die Funktion `DrawTheText`. Bei jedem Aufruf scrollt sie den Text um so viele Pixel nach links, wie in der Eigenschaft `ScrollIncrement` angegeben. In einer Schleife wird der Text immer wieder gezeichnet, falls im Steuerelement mehr als eine Kopie des Texts Platz haben. Diese Routine ist auch für die erneute Ausgabe des Texts zuständig, falls die `ReplaceText`-Eigenschaft gesetzt ist. Zur Entwurfszeit wird nur eine einzige Kopie der `ScrollText`-Eigenschaft gezeichnet, und das Scrolling ist deaktiviert.

Wie Sie sehen, ändert das Steuerelement nichts an der Registrierung, dem Container, dem Dateisystem oder anderen Fenstern im System. Ich kann nicht erkennen, daß der Aufruf der Methoden oder Eigenschaften des Steuerelements etwas solches verursachen könnte, weder bei der Initialisierung noch zur Entwurfszeit. Dieses Steuerelement kann also ohne weiteres als sicher markiert werden.

Das Steuerelement implementiert außerdem eine Eigenschaftsseite und ein Info-Feld. Den Code für diese Elemente finden Sie auf der CD-ROM zum Buch. Außerdem gibt es dort ein Programm, das die Verwendung des Steuerelements demonstriert.

Dieses Steuerelement kann noch erweitert werden, beispielsweise unter Verwendung der Funktion `ScrollWindow`, um die Scroll-Leistung zu verbessern. Das wird aber erst in einer späteren Version des Steuerelements der Fall sein.

21.4.1 Fazit

`dwBanner` ist das vielleicht einzige Beispiel für ein vollständiges Steuerelement in diesem Buch. Es erfüllt alle Anforderungen, die für ein kommerziell nutzbares ActiveX-Steuerelement gelten, außer daß das Testen der hier vorgestellten Version etwas zu kurz gekommen ist, und daß keine Online-Hilfe bereitgestellt wird.

Hier ein kurzer Überblick über die wichtigsten Punkte, die Sie bei der Entwicklung eines ActiveX-Steuerelements berücksichtigen sollten. Ich habe versucht, sie in einer logischen Reihenfolge anzugeben, die aber nur einen Anhaltspunkt darstellen soll:

- Definition der Benutzeroberfläche. Wenn Ihr Steuerelement nicht gerade zur Laufzeit unsichtbar ist, hat es eine Benutzeroberfläche. Vergessen Sie nicht, neben der Maus-Oberfläche auch eine Tastatur-Oberfläche zu berücksichtigen, insbesondere für grafikintensive Steuerelemente.
- Definition der Programmschnittstelle. Bevor Sie die erste Codezeile schreiben, sollten Sie die Eigenschaften, Methoden und Ereignisse des Steuerelements bestimmen, und festlegen, wie sie interagieren sollen.
- Implementieren der Eigenschaftsprozeduren. Denken Sie daran, die Eigenschaftstypen von `Variants` in spezielle Datentypen zu ändern, es sei denn, Sie brauchen unbedingt `Variants`. `Variants` sind langsamer und weniger effizient als andere Datentypen und bedingen eine aufwendigere Fehlerprüfung.

- Dem UserControl-Objekt und den zugehörigen Steuerelementen Eigenschaften zuweisen. Einige der öffentlichen Eigenschaften und Methoden Ihres Steuerelements verweisen einfach auf Eigenschaften und Methoden des UserControl-Objekts oder der zugehörigen Steuerelemente. Beachten Sie, daß ein öffentliches Element Ihres Steuerelements auf mehrere der zugehörigen Steuerelemente verweisen kann.
- Eigenschaften auswerten. Wirft Fehler auf, wenn versucht wird, eine Eigenschaft auf einen nicht erlaubten Wert zu setzen. Beachten Sie, daß Visual Basic keine automatische Überprüfung für Aufzählungstypen bereitstellt; Sie müssen Ihre Werte selbst überwachen.
- Laufzeit/Entwurfszeit-Charakteristika für Eigenschaften und Methoden einfügen. Wirft Fehler auf, wenn versucht wird, eine Eigenschaft zu einem ungültigen Zeitpunkt zu setzen. Versuchen Sie beispielsweise, eine Eigenschaft zur Laufzeit zu setzen, die zur Laufzeit nur gelesen werden soll.
- Implementieren der Initialisierung und Persistenz von Eigenschaften. Prüft alle Standardwerte für Eigenschaften.
- Definieren der Eigenschaftsseiten. Sie sollten Eigenschaftsseiten für alle Eigenschaften Ihres Steuerelements implementieren, weil Sie nicht voraussetzen können, daß ein Container einen Eigenschafts-Browser bereitstellt, der die Eigenschaften Ihres Steuerelements setzt.
- Implementierung der Ereignisse. Beachten Sie, daß Sie den öffentlichen Ereignissen Ihres Steuerelements UserControl-Ereignisse oder Ereignisse aus zugehörigen Steuerelementen zuordnen können.
- Zuweisung von Beschreibungen, Standard-Prozedur-IDs und Eigenschaftsseiten für Eigenschaften, Methoden und Ereignissen. Die Beschreibungen, die Sie setzen, erscheinen in dem Objekt-Browser für Ihr Steuerelement.
- Standard-Prozedur-IDs (auch als Dispatch-IDs bezeichnet) werden verwendet, um den Container anzuweisen, ein eigenschaftsspezifisches Verhalten zu implementieren. Unter Visual Basic ist das vor allem für die Enabled-Eigenschaft kritisch.
- Falls nötig, Code zum Zeichnen implementieren. Steuerelemente, die sich vollständig aus Standard-Steuerelementen zusammensetzen, benötigen möglicherweise keine zusätzlichen Zeichenoperationen, während für benutzerdefinierte Steuerelemente sehr viel Aufwand für das Zeichnen erforderlich ist.
- Anlegen und Einrichten einer Symbolleisten-Bitmap.
- Anlegen des Info-Felds.
- Anlegen einer Hilfedatei und Zuweisen der Hilfekontexte.

- Testen, testen, testen!
- Die Funktionalität auf Ziel-Containern überprüfen. Das ist vor allem wichtig, wenn Sie das `Extender`-Objekt benutzen, das sich in jedem Container anders verhält. Visual Basic selbst ist einer der komplexesten Container, den Sie verwenden. Die meisten anderen Container bieten nicht so viel Funktionalität. Sorgen Sie dafür, daß Fehler Ihres Steuerelements verarbeitet werden, falls eine Container-Funktionalität fehlt.

Damit beenden wir unsere Beschreibung von ActiveX-Controls für das Internet. Wir kehren in den Teilen IV und V zu den Internet- und Intranet-basierten Komponenten zurück, wo wir über ActiveX-Dokumente und IIS-Applikationen sprechen. Aber nachdem Sie nun alle Voraussetzungen für die Entwicklung von Steuerelementen haben, ist es an der Zeit, sich in die Stratosphäre zu katapultieren und einige der komplexeren Techniken kennenzulernen, die in Ihre Steuerelemente eingebaut werden können.