

Kapitel III

Aktualisierung der Felder mit einem Fortschritts- balken visualisieren

In diesem Kapitel:

Aktualisieren aller Felder eines Dokumentes	2
Erstellen und Einbinden eines Fortschrittsbalkens	13
Einbinden des Microsoft ProgressBar-Controls	17
Zusammenfassung	18

Dieses Kapitel zeigt Ihnen, wie Sie das Aktualisieren aller Felder in einem Dokument mit Hilfe eines Fortschrittsbalkens und des *ProgressBar*-Steuerelementes synchronisiert visualisieren können.

Im Blickpunkt:

Word-Objekte, Eigenschaften und Methoden

ActiveDocument.StoryRanges-Objekt

Range.StoryType-Eigenschaft

Range.NextStoryRange-Eigenschaft

Range.Fields-Eigenschaft

Fields.Update-Methode

Fields.Locked-Methode

ActiveDocument.Sections-Eigenschaft

Section.Headers-Eigenschaft

Section.Footers-Eigenschaft

ActiveDocument.Shapes-Objekt

Shape.TextFrame-Objekt

Benutzerformular-Steuerelemente, Ereignisse, Eigenschaften und Methoden

Label-Steuerelement

Label.With-Eigenschaft

Label.SpecialEffect-Eigenschaft

Label.MouseUp-Ereignis

Label.MouseDown-Ereignis

ListBox-Steuerelement

ListBox.AddItem-Methode

ProgressBar-Steuerelement

Aktualisieren aller Felder eines Dokumentes

In einem Word-Dokument lässt sich mittels Felder eine Vielzahl von Informationen erfassen und darstellen. Ändern sich die Informationen, die in den Feldern angezeigt werden, müssen auch die Felder aktualisiert werden. Dabei wird zwischen den Feldern unterschieden, die sich automatisch aktualisieren, wie z.B. Querverweis-Felder, und Feldern, die manuell aktualisiert werden müssen, wie das Inhaltsverzeichnis.

Es ist durchaus sinnvoll, dass nicht alle Felder immer automatisch von Word aktualisiert werden, da bei umfangreichen Dokumenten mit Querverweisen und Verknüpfungen zu Grafiken das Arbeiten ansonsten fast unmöglich würde, wenn wirklich alle Felder jedes mal überprüft werden würden.

Das Aktualisieren der Felder, die sich nicht automatisch aktualisieren, muss vom Anwender initiiert werden. Dies kann z.B. durch das Speichern und wieder Öffnen des Dokumentes erfolgen, wenn im Menü *Extras/Optionen* auf der Registerkarte *Allgemein* die Option *Automatische Verknüpfungen beim Öffnen aktualisieren* aktiviert ist. Eine andere Möglichkeit besteht darin, in die Dokumentansicht *Seitenlayout* zu wechseln. Oder Sie aktualisieren die Felder manuell mittels der Taste **[F9]** für ein einzelnes Feld bzw. für alle Felder, wenn Sie vorher das gesamte Dokument markiert haben (z.B. mittels **[Strg]+[A]**).

Leider haben alle Möglichkeiten ihre Nachteile: Entweder werden nicht alle Felder wirklich aktualisiert oder das Aktualisieren ist zu umständlich (wenn das Dokument immer erst neu geöffnet werden muss).

Die häufigste verwendete Möglichkeit, das gesamte markierte Dokument mittels **[F9]** zu aktualisieren, hat den großen Nachteil, dass dabei nur die Felder im normalen Textbereich (dem Hauptdokumentbereich) berücksichtigt werden, die auch in der Normalansicht angezeigt werden. Alle Felder in Kopf- und Fußzeilen, Fußnoten oder in Textrahmen werden mit dieser Methode nicht angesprochen und somit aktualisiert.

Das StoryRange-Objekt

Word bietet für den Zugriff auf die verschiedenen Bereiche eines Dokumentes (den Dokumentkomponenten: Kopf- und Fußzeilen, Fußnoten- und Endnotenbereich, Kommentarbereich und Hauptdokumentbereich) die StoryRanges-Eigenschaft eines Dokumentes. Diese liefert für das jeweilige Dokument eine Auflistung aller Dokumentkomponenten in Form einer StoryRanges-Auflistung zurück.



Die in diesem Abschnitt verwendeten Beispiele finden Sie auf der CD-ROM zum Buch in der Datei `\Beispiele\KapIII\BspIII_01.doc` im Modul *modStoryRanges*.

In Word 2003 stehen Ihnen insgesamt folgende Dokumentkomponentenbereiche zur Verfügung:

Tabelle III.1 Übersicht über die Komponentenbereiche (StoryRanges) eines Dokumentes

StoryRanges-Typ	Komponentenbereich	Max. Anzahl	Beschreibung
wdMainTextStory	Haupttextbereich	1 pro Dokument	Der normale Textbereich. Dieser Bereich ist immer vorhanden.
wdFootnotesStory	Fußnotenbereich	1 pro Dokument	Gehört zum Haupttextbereich. Wird erstellt, sobald Sie eine Fußnote erstellen.
wdEndnotesStory	Endnotenbereich	1 pro Dokument	Gehört zum Haupttextbereich. Wird erstellt, sobald Sie eine Endnote erstellen.
wdCommentsStory	Kommentarbereich	1 pro Dokument	Gehört zum Haupttextbereich. Wird erstellt, sobald Sie einen Kommentar erstellen.

Tabelle III.1 Übersicht über die Komponentenbereiche (*StoryRanges*) eines Dokumentes (*Fortsetzung*)

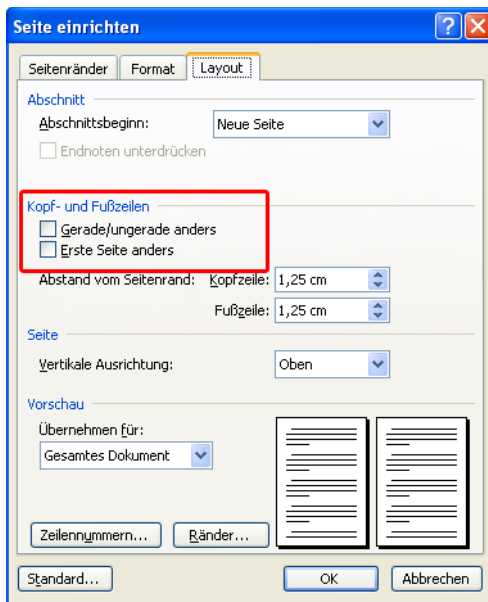
StoryRanges-Typ	Komponentenbereich	Max. Anzahl	Beschreibung
wdTextFrameStory	Textrahmenbereich	n pro StoryRange	Textrahmen gehören zu den Shape -Objekten und können in allen StoryRange -Objekten vorkommen.
wdPrimaryHeaderStory	Hauptkopfzeilenbereich/Kopfzeilenbereich ungerade Seiten	1 pro Abschnitt	Wird erstellt, sobald Sie Text in die erste Kopfzeile eintragen.
wdEvenPagesHeaderStory	Kopfzeilenbereich gerade Seiten	1 pro Abschnitt	Wird erstellt, wenn Sie das Seitenlayout auf <i>Gerade/ungerade anders</i> einstellen und Text in die Kopfzeile einer geraden Seitenzahl eintragen.
wdPrimaryFooterStory	Hauptfußzeilenbereich/Fußzeilenbereich ungerade Seiten	1 pro Abschnitt	Wird erstellt, sobald Sie Text in die erste Fußzeile eintragen.
wdEvenPagesFooterStory	Fußzeile gerade Seiten	1 pro Abschnitt	Wird erstellt, wenn Sie das Seitenlayout auf <i>Gerade/ungerade anders</i> einstellen und Text in die Fußzeile einer geraden Seitenzahl eintragen.
wdFirstPageHeaderStory	Kopfzeile erste Seite	1 pro Abschnitt	Wird erstellt, sobald Sie das Seitenlayout auf <i>Erste Seite anders</i> ändern und Text in die Kopfzeile der ersten Seiten eintragen.
wdFirstPageFooterStory	Fußzeile erste Seite	1 pro Abschnitt	Wird erstellt, sobald Sie das Seitenlayout auf <i>Erste Seite anders</i> ändern und Text in die Fußzeile der ersten Seiten eintragen
wdFootnoteSeparatorStory	Fußnoten-Trennlinienbereich	1 pro Dokument	Gehört zum Haupttextbereich. Bei neu erstellten Dokumenten existiert dieser Bereich, sobald Sie entweder eine Fußnote einfügen oder in die Kopf- oder Fußzeilenansicht wechseln.
wdFootnoteContinuationSeparatorStory	Fußnoten-Fortsetzungstrennlinienbereich	1 pro Dokument	Gehört zum Fußnotenbereich. Dieser Bereich wird erstellt, sobald Sie entweder eine Fußnote einfügen oder in die Kopf- oder Fußzeilenansicht wechseln.
wdFootnoteContinuationNoticeStory	Fußnoten-Fortsetzungshinweisbereich	1 pro Dokument	Gehört zum Fußnotenbereich. Dieser Bereich wird erstellt, sobald Sie einen Fußnoten-Fortsetzungshinweis erfassen.
wdEndnoteSeparatorStory	Endnoten-Trennlinienbereich	1 pro Dokument	Gehört zum Haupttextbereich. Bei neu erstellten Dokumenten existiert dieser Bereich, sobald Sie entweder eine Fußnote einfügen oder in die Kopf- oder Fußzeilenansicht wechseln.

Tabelle III.1 Übersicht über die Komponentenbereiche (StoryRanges) eines Dokumentes (Fortsetzung)

StoryRanges-Typ	Komponenten-bereich	Max. Anzahl	Beschreibung
wdEndnoteContinuationSeparatorStory	Endnoten-Fortsetzungstrennlinienbereich	1 pro Dokument	Gehört zum Haupttextbereich. Bei neu erstellten Dokumenten existiert dieser Bereich, sobald Sie entweder eine Fußnote einfügen oder in die Kopf- oder Fußzeilenansicht wechseln.
wdEndnoteContinuationNoticeStory	Endnoten-Fortsetzungshinweisbereich	1 pro Dokument	Gehört zum Endnotenbereich. Dieser Bereich wird erstellt, sobald Sie einen Endnoten-Fortsetzungshinweis erfassen.

Der Zugriff bzw. die Verfügbarkeit einzelner StoryRanges hängt davon ab, wie das Dokument eingerichtet ist. Speziell sind damit die Einstellungen gemeint, die Sie über den Menüpunkt *Datei/Seite einrichten* im gleichnamigen Dialogfeld *Seite einrichten* auf der Registerkarte *Layout* im Bereich *Kopf- und Fußzeilen* einstellen können (Abbildung III.1).

Abbildg. III.1 Einrichten eines Dokumentes mit unterschiedlichen Kopf- und Fußzeilen



Ein neues leeres Dokument, wie Sie es über den Menüpunkt *Datei/Neu* und dann im Aufgabenbereich über den Link *Leeres Dokument* erzeugen können, besitzt nur einen Komponentenbereich: den Haupttextbereich.

Sobald Sie das Seitenlayout auf *Gerade/ungerade anders* ändern, stehen Ihnen die StoryRange-Objekte wdPrimaryHeaderStory, wdEvenPagesHeaderStory, wdPrimaryFooterStory und wdEvenPagesFooterStory zur Verfügung. Sobald Sie in die Kopf- oder Fußzeilen Text eintragen, werden diese Bereiche im Dokument erstellt.

Wenn Sie das Seitenlayout auf *Erste Seite anders* ändern, stehen Ihnen die StoryRange-Objekte `wdFirstPageHeaderStory` und `wdFirstPageFooterStory` zur Verfügung und werden erstellt, sobald Sie in die entsprechenden Kopf-/Fußzeile Text eintragen.

Bei diesen StoryRange-Objekten handelt es sich um Dokumentkomponenten, die für jeden Abschnitt festgelegt werden können. Um nun alle Dokumentkomponenten eines bestimmten Typs in allen Abschnitten zu durchlaufen, steht Ihnen die Eigenschaft `NextStoryRange` der StoryRanges-Auflistung zur Verfügung. Mit dieser Eigenschaft können Sie prüfen, ob es weitere StoryRange-Objekte des jeweiligen Typs im Dokument gibt.

Das Beispiel in Listing III.1 zählt alle Textfelder im Hauptdokumentbereich, indem es prüft, ob die `NextStoryRange`-Eigenschaft einen Verweis auf ein weiteres Textfeld zurückliefert.

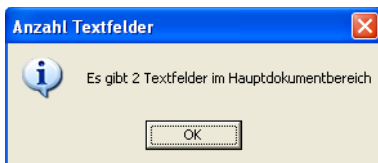
Listing III.1 Durchlaufen aller Textfelder im Hauptdokumentbereich

```
Sub subNextTextFrameStory()
    Dim rngStory As Range
    Dim intTFStory As Integer: intTFStory = 0
    For Each rngStory In ActiveDocument.StoryRanges
        If rngStory.StoryType = wdTextFrameStory Then
            intTFStory = intTFStory + 1
            While Not (rngStory.NextStoryRange Is Nothing)
                Set rngStory = rngStory.NextStoryRange
                intTFStory = intTFStory + 1
            Wend
        End If
    Next rngStory
    MsgBox "Es gibt " & intTFStory & " Textfeld" & _
        IIf(intTFStory = 1, "", "er") & " im Hauptdokumentbereich", _
        vbInformation, "Anzahl Textfelder"
End Sub
```



Wenn Sie die Prozedur `subNextTextFrameStory` auf das Beispieldokument *BspIII_02.doc* auf der CD-ROM zum Buch im Ordner `\Beispiele\KapIII` anwenden, erhalten Sie die in Abbildung III.2 gezeigte Ausgabe.

Abbildg. III.2 Anzeige der Textfeldanzahl im Dokument *BspIII_02.doc*



WICHTIG

Das Ermitteln aller Textfelder in einem Bereich liefert nur die Anzahl der eigenständigen Textfelder zurück. Miteinander verknüpfte Textfelder (wie in der Beispieldatei *BspIII_02.doc*) werden dabei nur als ein einziges Textfeld berücksichtigt.

Die Besonderheit beim Umgang mit Textfeldern wird im Abschnitt »Textfelder« behandelt.

Tabelle III.2 Übersicht über die von der *NextStoryRange*-Eigenschaft zurückgegebenen Dokumentkomponenten

Art der Dokumentkomponente	Dokumentkomponente, die von der <i>NextStoryRange</i> -Eigenschaft zurückgegeben wird
wdMainTextStory	Gibt immer Nothing zurück, da nur eine Komponente pro Dokument existiert.
wdFootnotesStory	Gibt immer Nothing zurück, da nur eine Komponente pro Dokument existiert.
wdEndnotesStory	Gibt immer Nothing zurück, da nur eine Komponente pro Dokument existiert.
wdCommentsStory	Gibt immer Nothing zurück, da nur eine Komponente pro Dokument existiert.
wdEvenPagesHeaderStory	Gibt die entsprechende Dokumentkomponente im nächsten Abschnitt zurück, sofern ein weiterer Abschnitt im Dokument existiert.
wdPrimaryHeaderStory	Gibt die entsprechende Dokumentkomponente im nächsten Abschnitt zurück, sofern ein weiterer Abschnitt im Dokument existiert.
wdEvenPagesFooterStory	Gibt die entsprechende Dokumentkomponente im nächsten Abschnitt zurück, sofern ein weiterer Abschnitt im Dokument existiert.
wdPrimaryFooterStory	Gibt die entsprechende Dokumentkomponente im nächsten Abschnitt zurück, sofern ein weiterer Abschnitt im Dokument existiert.
wdFirstPageHeaderStory	Gibt die entsprechende Dokumentkomponente im nächsten Abschnitt zurück, sofern ein weiterer Abschnitt im Dokument existiert.
wdFirstPageFooterStory	Gibt die entsprechende Dokumentkomponente im nächsten Abschnitt zurück, sofern ein weiterer Abschnitt im Dokument existiert.
wdTextFrameStory	Gibt den nächsten Textrahmen im Hauptdokumentbereich zurück, sofern ein weiterer im Dokument existiert.

Mit dieser Eigenschaft können Sie nun erstmals alle Dokumentkomponenten durchlaufen und Aktionen in den einzelnen Bereichen ausführen.

Das Listing III.2 zeigt Ihnen für das aktuelle Dokument genau an, welche StoryRanges im Dokument vorhanden sind.



Das vollständige Listing mit den notwendigen Konstanten-Deklarationen finden Sie auf der CD-ROM zum Buch in der Datei `\Beispiele\KapIII\BspIII_01.doc` im Modul `modStoryRanges`.

Listing III.2 Anzeigen aller Dokumentkomponenten eines Dokumentes

```

Sub subListStoryRanges()
    Dim rngStory As Range
    Dim strStoryName As String
    Dim strMSG As String
    For Each rngStory In ActiveDocument.StoryRanges
        Select Case rngStory.StoryType
            Case wdMainTextStory
                strStoryName = "Haupttextbereich"
            Case wdFootnotesStory
                strStoryName = "Fußnotenbereich"
        End Select
    Next rngStory
End Sub

```

Listing III.2 Anzeigen aller Dokumentkomponenten eines Dokumentes (Fortsetzung)

```

Case wdFootnoteSeparatorStory
    strStoryName = "Fußnoten-Trennlinienbereich"
Case wdFootnoteContinuationSeparatorStory
    strStoryName = "Fußnoten-Fortsetzungstrennlinienbereich"
Case wdFootnoteContinuationNoticeStory
    strStoryName = "Fußnoten-Fortsetzungshinweisbereich"
Case wdEndnotesStory
    strStoryName = "Endnotenbereich"
Case wdEndnoteSeparatorStory
    strStoryName = "Endnoten-Trennlinienbereich"
Case wdEndnoteContinuationSeparatorStory
    strStoryName = "Endnoten-Fortsetzungstrennlinienbereich"
Case wdEndnoteContinuationNoticeStory
    strStoryName = "Endnoten-Fortsetzungshinweisbereich"
Case wdCommentsStory
    strStoryName = "Kommentarbereich"
Case wdTextFrameStory
    strStoryName = "Textrahmenbereich"
Case wdEvenPagesHeaderStory
    strStoryName = "Kopfzeilenbereich gerade Seiten"
Case wdPrimaryHeaderStory
    strStoryName = "Hauptkopfzeilenbereich/Kopfzeilenbereich ungerade Seiten"
Case wdEvenPagesFooterStory
    strStoryName = "Fußzeile gerade Seiten"
Case wdPrimaryFooterStory
    strStoryName = "Hauptfußzeilenbereich/Fußzeilenbereich ungerade Seiten"
Case wdFirstPageHeaderStory
    strStoryName = "Kopfzeile erste Seite"
Case wdFirstPageFooterStory
    strStoryName = "Fußzeile erste Seite"
End Select
strMSG = strMSG & strStoryName & vbCrLf
While Not (rngStory.NextStoryRange Is Nothing)
    Set rngStory = rngStory.NextStoryRange
    Select Case rngStory.StoryType
    Case wdMainTextStory
        strStoryName = "Haupttextbereich"
    Case wdFootnotesStory
        strStoryName = "Fußnotenbereich"
    Case wdFootnoteSeparatorStory
        strStoryName = "Fußnoten-Trennlinienbereich"
    Case wdFootnoteContinuationSeparatorStory
        strStoryName = "Fußnoten-Fortsetzungstrennlinienbereich"
    Case wdFootnoteContinuationNoticeStory
        strStoryName = "Fußnoten-Fortsetzungshinweisbereich"
    Case wdEndnotesStory
        strStoryName = "Endnotenbereich"
    Case wdEndnoteSeparatorStory
        strStoryName = "Endnoten-Trennlinienbereich"
    Case wdEndnoteContinuationSeparatorStory
        strStoryName = "Endnoten-Fortsetzungstrennlinienbereich"
    Case wdEndnoteContinuationNoticeStory
        strStoryName = "Endnoten-Fortsetzungshinweisbereich"
    Case wdCommentsStory
        strStoryName = "Kommentarbereich"
    
```


Listing III.2 Anzeigen aller Dokumentkomponenten eines Dokumentes (Fortsetzung)

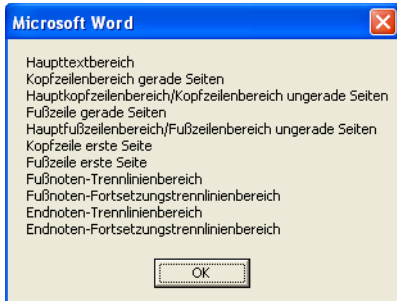
```

Case wdTextFrameStory
    strStoryName = "Textrahmenbereich"
Case wdEvenPagesHeaderStory
    strStoryName = "Kopfzeilenbereich gerade Seiten"
Case wdPrimaryHeaderStory
    strStoryName = "Hauptkopfzeilenbereich/Kopfzeilenbereich ungerade Seiten"
Case wdEvenPagesFooterStory
    strStoryName = "Fußzeile gerade Seiten"
Case wdPrimaryFooterStory
    strStoryName = "Hauptfußzeilenbereich/Fußzeilenbereich ungerade Seiten"
Case wdFirstPageHeaderStory
    strStoryName = "Kopfzeile erste Seite"
Case wdFirstPageFooterStory
    strStoryName = "Fußzeile erste Seite"
End Select
strMSG = strMSG & strStoryName & vbCrLf
Wend
Next rngStory
MsgBox strMSG
End Sub

```

In diesem Dokument, das dieses Kapitel beinhaltet, sind z.B. die in Abbildung III.3 abgebildeten Dokumentkomponenten enthalten:

Abbildg. III.3 Beispiel für die in einem Dokument enthaltenen Dokumentkomponentenbereiche



Felder in den *StoryRange*-Objekten aktualisieren

In diesem Kapitel soll ja die Aktualisierung aller Felder in einem Dokument behandelt werden, sodass wir uns die Möglichkeiten der StoryRange-Objekte für diese Aufgabe zu Nutzen machen können.

Um nun gezielt die Felder in einem Bereich anzusprechen, verwenden Sie die `Fields`-Eigenschaft des jeweiligen Bereiches. Diese Eigenschaft gehört zu den Objekten `Document`, `Range` oder `Selection` und erwartet ein entsprechendes Objekt im Aufruf. Die `Fields`-Eigenschaft liefert eine `Fields`-Auflistung aller Felder in dem angegebenen Bereich zurück. Über die `Update`-Methode der `Fields`-Eigenschaft können Sie die Felder der jeweiligen `Fields`-Auflistung aktualisieren:

```
ActiveDocument.Fields.Update
```

Wenn Sie nun als Objekt ein StoryRange-Objekt angeben, können Sie mit der Update-Methode alle Felder in diesem Bereich aktualisieren. In Listing III.3 werden als Beispiel die Hauptkopfzeilen in allen Abschnitten des aktuellen Dokumentes aktualisiert.

Listing III.3 Aktualisieren aller Felder in den Hauptkopfzeilen eines Dokumentes

```
Sub subHauptkopfzeilenfelderAktualisieren()
    Dim rngStory As Range
    For Each rngStory In ActiveDocument.StoryRanges
        If rngStory.StoryType = wdPrimaryHeaderStory Then
            rngStory.Fields.Update
            While Not (rngStory.NextStoryRange Is Nothing)
                Set rngStory = rngStory.NextStoryRange
                rngStory.Fields.Update
            Wend
        End If
    Next rngStory
End Sub
```

HINWEIS

Mit der Update-Methode werden im Haupttextbereich auch die Verzeichnis- und Indexfelder aktualisiert. Bei Inhaltsverzeichnissen werden aber dabei nur die Seitenzahlen aktualisiert; neu hinzugekommene Verzeichniseinträge werden nicht berücksichtigt. Um bei diesen Feldern ein vollständig neues Verzeichnis zu erstellen, müssen Sie gezielt auf das TablesOfContents-Objekt (oder ein anderes Verzeichnissfeld-Objekt) zugreifen und über die Update-Methode dieses Objektes das Verzeichnis erneuern:

```
ActiveDocument.TablesOfContents(1).Update
```

Textfelder in Kopf- und Fußzeilen ansprechen

Textfelder (Dokumentkomponente wdTextFrameStory) gehören zu den Shape-Objekten eines Dokumentes. Die Shapes-Eigenschaft eines Dokumentes liefert dazu eine Auflistung aller Shape-Objekte zurück, die sich im Dokument, mit Ausnahme der Kopf- und Fußzeilen, befinden. Diese Shapes-Auflistung kann Zeichnungen, Formen, Bilder, OLE-Objekte, ActiveX-Steuerelemente, Textobjekte und Legenden enthalten. Textfelder stellen dabei eine besondere Gruppe innerhalb der Shape-Objekte dar, da sie Text enthalten können, der wiederum z.B. ein Feld beinhalten kann. Zur Überprüfung von Textfeldern auf Textinhalte können Sie die HasText-Eigenschaft des Textfeldes (TextFrame-Eigenschaft des Shape-Objektes) auswerten. Diese liefert True zurück, wenn das Textfeld Text beinhaltet.

Textfelder im Haupttextbereich können über die NextStoryRange-Eigenschaft durchlaufen werden. Dabei werden miteinander verknüpfte Textfelder aber nur als ein einziges Textfeld behandelt.

Um nun auch die Textfelder in Kopf- oder Fußzeilen zu erreichen, benötigen Sie ein Objekt, das die Kopf- bzw. Fußzeile repräsentiert. Hierzu können Sie die HeaderFooter-Objekte verwenden, die in jedem Abschnitt eines Dokumentes alle Kopf- und Fußzeilen zurückliefern. Da diese Objekte nur abschnittsweise die Kopf- und Fußzeilen zurückliefern, müssen Sie das Dokument auch abschnittsweise durchlaufen: z.B. in einer For Each...Next-Anweisung über alle Abschnitte:

```
For Each oSections In Activedocument.Sections
```

Anschließend müssen Sie in jedem Abschnitt getrennt voneinander die Kopf- und Fußzeilen ansprechen. Dieses erreichen Sie über die Headers- und Footers-Auflistungen der gleichnamigen Sections-Eigenschaften Headers und Footers:

```
For Each oHeaderFooter In oSections.Headers
```

und

```
For Each oHeaderFooter In oSections.Footers
```

Nachdem Sie so die Kopf- und Fußzeilen referenziert haben und ansprechen können, müssen Sie als Nächstes alle Shapes dahingehend überprüfen, ob das Shape-Objekt ein Textfeld ist und, wenn dies der Fall ist, ob das Textfeld auch Text beinhaltet.

In Listing III.4 werden nacheinander alle Abschnitte durchlaufen und in diesen zuerst die Kopfzeilen auf Textfelder überprüft und anschließend die Fußzeilen. Für den Fall, dass die Textfelder auch Text beinhalten, werden alle evtl. vorhandenen Felder aktualisiert.

Listing III.4 Aktualisieren aller Textfelder in den Kopf- und Fußzeilen aller Abschnitte eines Dokumentes

```
Sub subKopfFußzeilenTextfelderAktualisieren()
    Dim oSections As Section
    Dim oHeaderFooter As HeaderFooter
    Dim shpHeaderFooter As Shape
    For Each oSections In ActiveDocument.Sections
        For Each oHeaderFooter In oSections.Headers
            For Each shpHeaderFooter In oHeaderFooter.Shapes
                If shpHeaderFooter.TextFrame.HasText Then
                    shpHeaderFooter.TextFrame.TextRange.Fields.Update
                End If
            Next shpHeaderFooter
        Next oHeaderFooter
        For Each oHeaderFooter In oSections.Footers
            For Each shpHeaderFooter In oHeaderFooter.Shapes
                If shpHeaderFooter.TextFrame.HasText Then
                    shpHeaderFooter.TextFrame.TextRange.Fields.Update
                End If
            Next shpHeaderFooter
        Next oHeaderFooter
    Next oSections
End Sub
```

Durchlaufen aller Dokumentkomponenten

Nachdem Sie nun in den vorherigen Abschnitten erfahren haben, wie Sie Felder in den einzelnen Dokumentkomponentenbereichen aktualisieren können und wie Sie Felder in Textfeldern aktualisieren können, die sich in den Kopf- und Fußzeilen befinden, werden diese Informationen nun so in dem Listing III.5 zusammengefasst, dass Sie alle Felder in allen Bereichen eines Dokumentes errei-

chen und aktualisieren können. Zusätzlich werden auch die besonderen Felder wie die Verzeichnissfelder berücksichtigt.

HINWEIS Sie können einzelne Felder über die Locked-Eigenschaft auch für die Aktualisierung sperren. Um diese Felder nicht unnötigerweise zu aktualisieren, werden nur die Felder, die nicht gesperrt sind, berücksichtigt.

Listing III.5 Aktualisieren aller Felder (inkl. Verzeichnis- und Indexfelder) eines Dokumentes

```
Sub subAlleFelderAktualisieren()
    Dim oDoc As Document
    Dim rngStory As Range
    Dim fldStory As Field
    Dim oSections As Section
    Dim oHeaderFooter As HeaderFooter
    Dim shpHeaderFooter As Shape
    Dim tocCont As TableOfContents
    Dim tocFig As TableOfFigures
    Dim tocAuth As TableOfAuthorities
    Set oDoc = ActiveDocument
    ' Alle Felder im Haupttextbereich, die nicht gesperrt sind, aktualisieren
    For Each rngDoc In ActiveDocument.StoryRanges
        For Each fldStory In rngDoc.Fields
            If fldStory.Locked = False Then
                fldStory.Update
            End If
        Next fldStory
        While Not (rngDoc.NextStoryRange Is Nothing)
            Set rngDoc = rngDoc.NextStoryRange
            For Each fldStory In rngDoc.Fields
                If fldStory.Locked = False Then
                    fldStory.Update
                End If
            Next fldStory
        Wend
    Next rngDoc
    ' Alle Textfelder in Shapes in allen Kopf- und Fußzeilen
    ' aller Abschnitten aktualisieren
    For Each oSections In oDoc.Sections
        ' Shapes in den Kopfzeilen überprüfen
        For Each oHeaderFooter In oSections.Headers
            For Each shpHeaderFooter In oHeaderFooter.Shapes
                If shpHeaderFooter.TextFrame.HasText Then
                    shpHeaderFooter.TextFrame.TextRange.Fields.Update
                End If
            Next shpHeaderFooter
        Next oHeaderFooter
        ' Shapes in den Fußzeilen überprüfen
        For Each oHeaderFooter In oSections.Footers
            For Each shpHeaderFooter In oHeaderFooter.Shapes
                If shpHeaderFooter.TextFrame.HasText Then
                    shpHeaderFooter.TextFrame.TextRange.Fields.Update
                End If
            Next shpHeaderFooter
        Next oHeaderFooter
    Next oSections
End Sub
```

Listing III.5 Aktualisieren aller Felder (inkl. Verzeichnis- und Indexfelder) eines Dokumentes (Fortsetzung)

```

For Each rngStory In ActiveDocument.StoryRanges
    If rngStory.StoryType = wdTextFrameStory Then
        intTFStory = intTFStory + 1
        While Not (rngStory.NextStoryRange Is Nothing)
            Set rngStory = rngStory.NextStoryRange
            intTFStory = intTFStory + 1
        Wend
    End If
Next rngStory
' Alle Rechtsgrundlagenverzeichnisse aktualisieren
For Each tocAuth In oDoc.TablesOfAuthorities
    tocAuth.Update
Next tocAuth
' Alle Abbildungsverzeichnisse aktualisieren
For Each tocFig In oDoc.TablesOfFigures
    tocFig.Update
Next tocFig
' Alle Inhaltsverzeichnisse aktualisieren
For Each tocCont In oDoc.TablesOfContents
    tocCont.Update
Next tocCont
Set oDoc = Nothing
End Sub

```

Erstellen und Einbinden eines Fortschrittsbalkens

Ein Fortschrittsbalken ist ein grafisches Element, das den aktuellen Fortschritt einer Verarbeitung in Bezug auf die Gesamtbearbeitung anzeigt. Beim Aktualisieren der Felder eines Dokumentes würde ein Fortschrittsbalken die zu einem bestimmten Zeitpunkt bereits aktualisierten Felder in Bezug zur Gesamtanzahl aller Felder anzeigen.

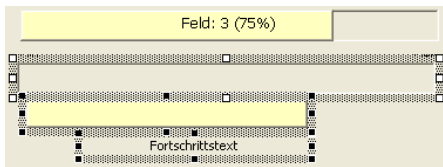
Ein Fortschrittsbalken setzt sich im Prinzip aus drei Unterelementen zusammen:

1. Dem Fortschrittsbalken-Hintergrund, der die Gesamtzahl aller Aktionen darstellt.
2. Dem Laufbalken, der die aktuelle Anzahl der bereits erfolgten Aktionen in Relation zur Gesamtzahl anzeigt.
3. Einem Anzeigetext, der die grafische Anzeige des Laufbalkens in Prozent und/oder Zahlen anzeigt.

Der Anzeigetext ist dabei nur eine weitere Ablesehilfe und kein unabdingbarer Bestandteil eines Fortschrittsbalkens.

Zusätzlich können noch Textfelder zur Anzeige der Minimum- und Maximumwerte verwendet werden.

Die Abbildung III.4 zeigt die drei Elemente anhand des Beispiels aus dem UserForm *frmFortschrittsbalkenCtrl* in der Datei *BspIII_01.doc* noch einmal nebeneinander und als Ergebnis.

Abbildg. III.4 Bestandteile eines Fortschrittsbalkens und in der Gesamtdarstellung

HINWEIS

Die Funktion *fkt_setStartPos* im UserForm *frmFortschrittsBalkenCtrl*, die im Initialize-Ereignis des UserForms aufgerufen wird, sorgt für die korrekte Positionieren der drei Bestandteile und Layouteffekte.

Um einen Fortschrittsbalken zur Laufzeit anzeigen und aktualisieren zu können, muss die Gesamtzahl aller Aktionen bekannt sein.

Beim Aktualisieren der Felder können Sie die Feldanzahl in einer Dokumentkomponente über die Count-Eigenschaft der Fields-Auflistung ermitteln. Beim Aktualisieren der Felder können Sie dann aber nicht mehr mit der *For Each...Next*-Anweisung arbeiten, da diese für das verwendete Objekt nicht immer einen Laufindex anbietet, sondern müssen mit der *For...Next*-Anweisung alle Elemente bis zum Maximumwert Count durchlaufen. Dieses ist deshalb notwendig, um den Fortschritt der Aktualisierung für jedes Feld darstellen zu können. Dieser Wert muss dann an das UserForm mit dem Fortschrittsbalken weitergegeben werden, wo mit Hilfe des Dreisatzes der Laufbalken grafisch an die Gesamtzahl und somit Breite des Fortschrittsbalkens-Hintergrundes angepasst wird.

```
lblLaufbalken.Width = lblHintergrund.Width / intGesamtzahlFelder * intAktuellesFeld
```

Zum Zurücksetzen wird dann die Breite des Laufbalkens einfach auf »0« gesetzt.

Damit die Darstellung des Fortschritts einfacher mit dem Fortschritt im Dokument synchronisiert werden kann, wird die Verarbeitung nach der Anzeige des UserForms gestartet. Da das UserForm während der gesamten Aktualisierung angezeigt bleibt, stehen die Controls auf dem UserForm auch in den Prozeduren mit ihren Eigenschaften zur Verfügung.

In Listing III.6 wird der aktuelle Aktualisierungsstand (*intFld*) der Felder im Haupttextbereich zusammen mit der Gesamtanzahl aller Felder an die Funktion *fkt_setPos* weitergegeben, in der die Breite des Laufbalkens auf dem UserForm berechnet und angepasst wird.

Listing III.6 Aktualisieren der Felder im Haupttextbereich und Weitergabe der Feldnummer an das UserForm

```
For Each rngDoc In oDoc.StoryRanges
    For intFld = 1 To rngDoc.Fields.Count
        fkt_setPos intFld, rngDoc.Fields.Count
        If rngDoc.Fields(intFld).Locked = False Then
            rngDoc.Fields(intFld).Update
        End If
    Next intFld
    While Not (rngDoc.NextStoryRange Is Nothing)
        Set rngDoc = rngDoc.NextStoryRange
        For intFld = 1 To rngDoc.Fields.Count
            fkt_setPos intFld, rngDoc.Fields.Count
```

Listing III.6 Aktualisieren der Felder im Haupttextbereich und Weitergabe der Feldnummer an das UserForm (Fortsetzung)

```

    rngDoc.Fields(intFld).Update
    Next intFld
Wend
Next rngDoc

```



In der Beispieldatei *BspIII_01.doc* auf der CD-ROM zum Buch finden Sie im Ordner *\Beispiele\KapIII* die vollständigen Prozeduren *AlleFelderMitTextfeldernAktualisieren* und *fk_t_setPos*.

Da Felder in verschiedenen Dokumentkomponenten und in mehreren Abschnitten vorkommen können, wäre die einfache Anzeige des Fortschrittsbalkens nicht informativ genug. So fehlen Informationen über die Abschnittsnummer, ob das Feld in einem Textfeld in der Kopf- oder Fußzeile ist, ob das Feld ein Verzeichnisfeld ist und wie viele Felder am Ende insgesamt aktualisiert wurden.

Da diese Informationen mit Hilfe der StoryRanges-Auflistung ermittelt werden können und zur Ermittlung der Gesamtfeldzahl eine einfache Laufvariable ausreichend ist, steht einer ansprechenden Übersicht nichts mehr im Wege.

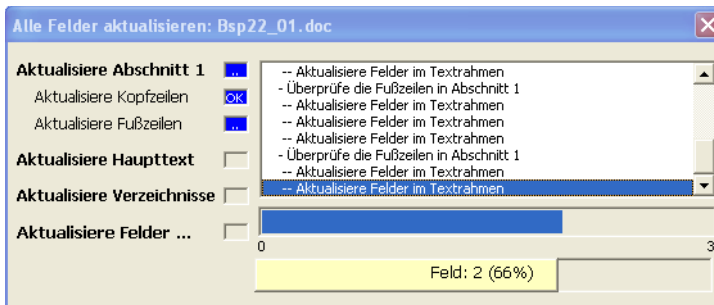
Wenn Sie in der Beispieldatei *BspIII_01.doc* im CD-ROM Ordner *\Beispiele\KapIII* auf die Schaltfläche *Felder aktualisieren* klicken, wird das UserForm *frmFortschrittsBalkenCtrl* angezeigt und die Aktualisierung durchläuft das aktuelle Dokument (Abbildung III.5).

Abbildg. III.5 Erfolgte Aktualisierung aller Felder mit Informationen über den Aktualisierungsbereich

In dem UserForm werden die Informationen über die einzelnen durchlaufenen Dokumentkomponenten an zwei Stellen angezeigt:

- In der ListBox werden alle Einträge untereinander angezeigt und können über den vertikalen Rollbalken angezeigt werden.
- In der Textform wird der Verarbeitungszustand einzelner Dokumentkomponenten und Abschnitte angezeigt.

Während der Aktualisierung werden die Dokumentkomponenten, in denen gerade Felder aktualisiert werden, mit einem blauen Kästchen markiert. Bereiche (und somit Kästchen), in denen die Aktualisierung erfolgt ist, werden mit dem Text »OK« dargestellt, während noch nicht vollständig aktualisierte Bereiche mit zwei Punkten dargestellt werden.

Abbildg. III.6 Anzeige des Aktualisierungsfortschritts in den einzelnen Dokumentkomponenten


Nach erfolgter Aktualisierung aller Felder wird der Fortschrittsbalken zu einer Schaltfläche umgewandelt: Mit einem Klick auf die Schaltfläche *Schließen* in der Titelleiste wird das UserForm geschlossen.

PROFITIPP

Mit einer geschickten Wahl der Layouteffekte von Steuerelementen auf einem UserForm und den Steuerelement-Ereignissen `MouseDown` und `MouseUp` können Sie aus einem Textfeld-Steuerelement oder einem Bezeichnungsfeld-Steuerelement eine Schaltfläche simulieren. Dazu wird in den jeweiligen Ereignissen des Steuerelementes das Aussehen durch »umkippen« des 3D-Effektes (`SpecialEffect`-Eigenschaft) erreicht (Listing III.7).

Listing III.7 Simulieren einer Schaltfläche mit einem Bezeichnungsfeld durch Änderung des Aussehens

```
Private Sub txtcounter_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, _
    ByVal x As Single, ByVal y As Single)
    Me.lblBackground.SpecialEffect = fmSpecialEffectSunken
End Sub
Private Sub txtcounter_MouseUp(ByVal Button As Integer, ByVal Shift As Integer, _
    ByVal x As Single, ByVal y As Single)
    Me.lblBackground.SpecialEffect = fmSpecialEffectRaised
    Unload Me
End Sub
```

Durch Einbinden einer entsprechenden Aktion, in diesem Beispiel das Schließen des UserForms, erreichen Sie das Verhalten einer richtigen Schaltfläche.

Wenn Sie eine komplexere Schaltfläche mit Grafik und Text simulieren möchten, müssen Sie für alle Steuerelemente, die Bestandteil der Pseudo-Schaltfläche sind, die `MouseDown`- und `MouseUp`-Ereignisse mit den Effektänderungen und weiteren evtl. auszuführenden Aktionen definieren.

Eine Alternative dazu wäre das Erstellen einer richtigen Schaltfläche zur Laufzeit, was aber auch eine Größenänderung des UserForms oder Neupositionierung von Steuerelementen auf dem UserForm bedeuten würde.

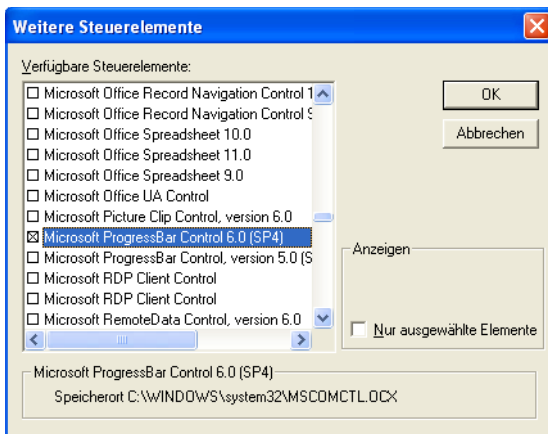
In Abbildung III.6 sehen Sie neben dem selbst gebauten Fortschrittsbalken auch eine fertige Lösung in Form des Microsoft `ProgressBar`-Controls. Dieses wird im folgenden Abschnitt »Einbinden des Microsoft `ProgressBar`-Controls« etwas näher behandelt.

Einbinden des Microsoft ProgressBar-Controls

Microsoft stellt mit dem Windows-Standardsteuerelement *MSCOMCTL.OCX* u.a. ein *ProgressBar*-Control zur Verfügung, das auf allen aktuellen Windows-Betriebssystemen verfügbar ist.

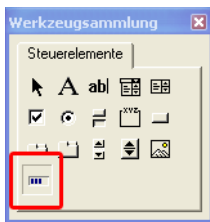
Sie können dieses Steuerelement in Ihre Werkzeugsammlung im Visual Basic-Editor einbinden, indem Sie mit der rechten Maustaste auf die Werkzeugsammlung klicken und im angezeigten Kontextmenü den Menüpunkt *Zusätzliche Steuerelemente* auswählen. In dem daraufhin angezeigten Dialogfeld suchen Sie den Eintrag *Microsoft ProgressBar Control 6.0 (SP4)*, markieren ihn und bestätigen die Auswahl über die *OK*-Schaltfläche.

Abbildg. III.7 Hinzufügen des *ProgressBar*-Controls aus der Datei *MSCOMCTL.OCX*



Anschließend steht Ihnen auf der Werkzeugsammlung das neue Steuerelement *ProgressBar* zur Verfügung (Abbildung III.8).

Abbildg. III.8 Werkzeugsammlung mit zusätzlichem Steuerelement *ProgressBar*



Sie können dieses Steuerelement nun auf ein UserForm ziehen.

Die wichtigsten Eigenschaften dieses Steuerelementes sind die Minimum- und Maximumwert-Angaben (Eigenschaften *Min* und *Max*) und der anzuzeigende Wert (*Value*-Eigenschaft).

Analog zum Fortschrittsbalken werden dem Steuerelement während der Aktualisierung der Felder diese Informationen mitgeteilt.

Da dieses Steuerelement jedoch keine Caption- oder Text-Eigenschaft zur Anzeige eines Textes besitzt und Sie kein Bezeichnungsfeld über das Steuerelement legen können (es wird immer hinter dem Steuerelement angeordnet), wird in diesem Beispiel auf die Textanzeige des Fortschritts verzichtet.

Das Ansteuern des *ProgressBar*-Steuerelementes während der Aktualisierung erfolgt ebenfalls in der Prozedur *fkt_setPos* (Listing III.8). Nach dem Zurücksetzen des Laufbalkens werden erst für das *ProgressBar*-Steuerelement *pgbctrl* die Grenzen und der aktuelle Wert und anschließend für den Laufbalken der Fortschrittsanzeige die aktuelle Breite eingestellt. Im dem den Fortschrittsbalken überlagernden Textfeld wird zusätzlich zur aktuellen Feldzahl auch der prozentuale Wert in Bezug auf die Feldgesamtzahl berechnet und angezeigt.

Listing III.8 Festlegen der Werte für das *ProgressBar*-Steuerelement und den Fortschrittsbalken

```
Function fkt_setPos(intWert As Long, intMax As Long)
With frmFortschrittsBalkenCtrl
    .lblLeiste.Width = 0
    .lblBackground.SpecialEffect = fmSpecialEffectSunken
    .lblMax.Caption = intMax
    .lblMin.Caption = "0"
    If intWert <= intMax Then
        .pgbctrl.Min = 1
        .pgbctrl.Value = intWert * 100
        .pgbctrl.Max = intMax * 100
        .lblLeiste.Width = Int(.lblBackground.Width / intMax * intWert)
        .txtcounter.Caption = "Feld: " & intWert & " (" & Int(intWert / intMax * 100) & "%)"
        DoEvents
    Else
        .pgbctrl.Min = 0
        .pgbctrl.Value = 0
        .pgbctrl.Max = intMax * 100
    End If
End With
End Function
```

Bei Erreichen des Maximalwertes, also der Anzahl aller Felder in dem jeweiligen Bereich, wird die Anzeige der *ProgressBar* wieder auf »0« zurückgesetzt.

In Abbildung III.6 sehen Sie als Ergebnis die synchrone Anzeige beider Darstellungen (*ProgressBar* und Fortschrittsbalken) untereinander mit identischen Werten.

Wenn Sie in der Beispieldatei *BspIII_01.doc* auf der Buch-CD im Ordner *\Beispiele\KapIII* auf die Schaltfläche *Felder aktualisieren* klicken, wird das UserForm *frmFortschrittsBalkenCtrl* angezeigt und die Aktualisierung wird in beiden Darstellungen (*ProgressBar* und Fortschrittsbalken) angezeigt.

Zusammenfassung

In diesem Kapitel wurde Ihnen gezeigt, aus welchen Komponenten ein Dokument besteht, wo überall Felder eingefügt werden können und wie Sie diese ansprechen können;

- Mit Hilfe der *StoryRanges*-Auflistung und der *StoryRange*-Objekte haben Sie gesehen, wie Sie die verschiedenen Dokumentkomponenten gezielt ansprechen (Seite 3 ff. und Seite 9 ff.) und Felder in diesen Bereichen aktualisieren können.

- Anschließend wurde Ihnen die Besonderheit beim Umgang mit Feldern, die innerhalb von Textfeldern in Kopf- oder Fußzeilen eingefügt sind, gezeigt (Seite 10 ff.), bevor alle Informationen zusammengefasst wurden, um alle Felder in allen Dokumentkomponenten zu durchlaufen (Seite 11 ff.).
- Anhand eines Fortschrittsbalkens (Seite 13 ff.) und des *ProgressBar*-Steuerelementes (Seite 17 ff.) wurde Ihnen gezeigt, wie Sie den Aktualisierungsfortschritt beim Aktualisieren aller Felder in einem Dokument darstellen können.

