

Kapitel 43

Reporting Services programmieren

In diesem Kapitel:

Möglichkeiten der Integration	1322
Die Schnittstellen URL und Webservice	1322
Das ReportViewer-Control	1348
Abschlussbemerkungen	1358

Die SQL Server Reporting Services stellen Schnittstellen zur Verfügung, mit denen die Administration des Berichtsservers und den dort veröffentlichten Berichten in eigene Applikationen integriert werden können. Auch die Berichtsanzeige oder das Rendern eines Berichts in verschiedene Formate wird von diesen Schnittstellen unterstützt.

In diesem Kapitel werden Sie einige der Integrationsmöglichkeiten kennen lernen, wobei der Fokus auf der Berichtsgenerierung liegt.

Möglichkeiten der Integration

Die bereits mit den Reporting Services 2000 verfügbaren Schnittstellen, die ermöglichen, den Berichtsserver und dessen Berichte über eine URL oder über einen Webservice anzusteuern bzw. zu programmieren, sind auch im SQL Server 2005 weiterhin verfügbar.

Bei beiden Varianten handelt es sich nicht um eine wirkliche Integration. Dies wird am Beispiel der Berichtsgenerierung deutlich. Der Zugriff auf einen Bericht via URL kann diesen zwar öffnen bzw. in ein vorher definiertes Format rendern und speichern, doch dazu wird immer der Browser gestartet. Für die Berichtsanzeige des Online-Berichts wird der Browser ebenso benötigt, wie für das Öffnen eines gerenderten Berichts immer die jeweilige Software gestartet werden muss. So muss für die Anzeige eines in das Excel-Format gerenderten Berichts zunächst Excel gestartet werden.

Dies gilt auch für den Zugriff über den Webservice. Hier wird für den Render-Vorgang zwar kein Browser benötigt, trotzdem müssen die gerenderten Berichte mit den jeweiligen Programmen geöffnet werden. Eine Berichtsanzeige wie über die URL-Schnittstelle ist hier erst gar nicht möglich.

Bei beiden Varianten findet die Berichtsgenerierung wie die Berichtsanzeige außerhalb der eigentlichen Applikation statt. Um die Berichte nun innerhalb der Applikation anzuzeigen, steht das *ReportViewer*-Control zur Verfügung, das mit Visual Studio 2005 ausgeliefert wird. Mit diesem Control erhalten Sie eine Berichtsanzeige, die die Berichte innerhalb Ihrer Applikation generiert und anzeigt. Und noch mehr: Sie können mit dem *ReportViewer*-Control auch Berichte generieren und anzeigen, die nicht auf dem Berichtsserver gespeichert sind.

Die Implementation des *ReportViewer*-Controls wird im Abschnitt »Das ReportViewer-Control« beschrieben. Doch zunächst ein Blick auf die Möglichkeiten der Zugriffe über die URL- bzw. Webservice-Schnittstellen.

Die Schnittstellen URL und Webservice

Auf den folgenden Seiten werden Sie beide Schnittstellen anhand eines Beispiels kennen lernen. Ergänzend dazu erhalten Sie einen Einblick in die Möglichkeiten, die beide Schnittstellen für die Administration des Berichtsservers zur Verfügung stellen.

Die Applikation Druckdialog

Als Beispiel werden Sie einen kleinen Druckdialog entwickeln, mit dem Sie den Bericht *Kundenliste* – der im vorherigen Kapitel mit dem Berichts-Generator erstellt wurde – entweder über die URL- oder über die Webservices-Schnittstelle in verschiedene Formate rendern können.

Der Druckdialog wird als Windows-Form in Visual Basic.NET entwickelt und beinhaltet lediglich zwei Listenfelder und eine Schaltfläche. Mit dem ersten Listenfeld wird das Format des Berichts ausgewählt und mit dem zweiten die Variante des Zugriffs bestimmt. Der Render-Vorgang wird über die Schaltfläche gestartet.

Die notwendigen Schritte zur Entwicklung des Windows-Forms werden im nächsten Abschnitt kurz beschrieben; die eigentliche Programmierung der URL- bzw. Webservice-Funktionen folgt im Anschluss daran.

CD-ROM

Sie finden die nachfolgenden Listings im Verzeichnis *Projekte\Kapitel 43 - Berichtsprogrammierung*.

Starten Sie Visual Studio und legen Sie ein neues Projekt vom Typ *Visual Basic* mit der Vorlage *Windows-Anwendung* an. Geben Sie dabei dem Projekt die Bezeichnung »Druckdialog«.¹ Nachdem das Projekt angelegt wurde, ändern Sie als erstes den Dateinamen der erzeugten Visual Basic-Datei *Form1.vb* in »Druckdialog.vb«. Den Titel des Dialogs *Kundenliste drucken* tragen Sie im Textfeld *Text* in den Eigenschaften des Windows-Forms ein.

Für die beiden Listenfelder ziehen Sie zwei Controls vom Typ *Combobox* aus der Toolbox in das Windows-Form. In der jeweiligen Eigenschaft (*Name*) der beiden Listenfelder geben Sie dem ersten Listenfeld die Bezeichnung »cboFormat« und dem zweiten »cboVariante«. Um in den Listenfeldern nur die Werte der Liste auswählen zu können, ändern Sie die Eigenschaft *DropDownStyle* beider Listenfelder auf *DropDownList*.

Die Listenfelder müssen noch um eine Beschriftung ergänzt werden. Dazu ziehen Sie zwei *Label*-Controls aus der Toolbox in das Windows-Form, von denen Sie eines in der Nähe des ersten Listenfelds und das andere in der Nähe des zweiten Listenfelds platzieren. Dem Label für das Listenfeld *cboFormat* geben Sie in der Eigenschaft (*Name*) die Bezeichnung »lblFormat« und in der Eigenschaft *Text* als Ausgabewert den Eintrag »Format:«. Beim zweiten Label tragen Sie unter der Eigenschaft (*Name*) »lblVariante« und unter der Eigenschaft *Text* »Variante:« ein.

Nun müssen Sie die Listenfelder noch mit Daten füllen. Dazu markieren Sie das jeweilige Listenfeld und klicken auf den Link *Einträge bearbeiten*, den Sie im Fuß des *Eigenschaften*-Fensters sehen. Im darauf folgenden Dialog können Sie die Einträge des Listenfelds erfassen. Für das Listenfeld *cboFormat* geben Sie die Werte »CSV«, »Excel«, »HTML«, »Image as TIFF«, »Image as JPEG«, »Online-Bericht«, »PDF« und »XML« jeweils in eine neue Zeile ein. Das zweite Listenfeld ergänzen Sie um die Einträge »per URL« und »per Webservice«.

¹ Die hier verwendeten Bezeichnungen des Windows-Forms, sowie der Controls und Subroutinen sind natürlich nicht zwingend, erleichtern aber die Nachvollziehbarkeit des Beispiels.

Der Druckdialog soll mit einer Vorauswahl zum Erstellen eines Online-Berichts über die URL-Schnittstelle gestartet werden. Für diese Vorauswahl müssen den Listenfeldern Standardwerte zugewiesen werden. Dazu wechseln Sie mit einem Doppelklick auf eine freie Fläche im Windows-Form in den Source-Editor und geben dort das Listing 43.1 ein.

```
Private Sub Druckdialog_Load(ByVal sender As System.Object,
                             ByVal e As System.EventArgs) Handles MyBase.Load
    Try
        cboFormat.SelectedItem = "Online-Bericht"
        cboVariante.SelectedItem = "per URL"
    Catch ex As Exception
        MsgBox(Err.Number.ToString & " - " & Err.Description)
    End Try
End Sub
```

Listing 43.1 Standardwerte für die Listenfelder

Jetzt fehlt nur noch die Schaltfläche, über die die Berichtsgenerierung gestartet werden soll. Ziehen Sie ein Control vom Typ *Button* aus der Toolbox in das Windows-Form und ändern Sie dessen Eigenschaften (*Name*) in »cmdStart« und *Text* in »Bericht erstellen«. Die eigentliche Verarbeitung der Schaltfläche wird in einer Subroutine entwickelt. Dazu müssen Sie mit einem Doppelklick auf die Schaltfläche zu dieser Subroutine wechseln. Dort tragen Sie das Listing 43.2 ein.

```
Private Sub cmdStart_Click(ByVal sender As System.Object, _
                           ByVal e As System.EventArgs) Handles cmdStart.Click
    Try
        If cboVariante.SelectedItem = "per URL" Then
            sStartPerUrl(cboFormat.SelectedItem)
        ElseIf cboVariante.SelectedItem = "per Webservice" Then
            sStartPerWebservice(cboFormat.SelectedItem)
        End If
    Catch ex As Exception
        MsgBox("Bei der Auswahl der Variante ist ein Fehler aufgetreten." & _
              (Chr(13) & Chr(10)) & "Wiederholen Sie die Auswahl.")
    End Try
End Sub
```

Listing 43.2 Auswahl der Varianten

Je nach ausgewählter Variante wird in eine weitere Subroutine verzweigt, in der der Zugriff über die jeweilige Schnittstelle codiert ist. Legen Sie die beiden Subroutinen aus Listing 43.3 an.

```
Private Sub sStartPerUrl(ByVal strFormat As String)
End Sub
Private Sub sStartPerWebservice(ByVal strFormat As String)
End Sub
```

Listing 43.3 Die Subroutinen *sStartPerUrl* und *sStartPerWebservice*

Die Subroutine *sStartPerUrl* steuert die Berichtsgenerierung über die URL-Schnittstelle. Dazu mehr im Abschnitt »Die Schnittstelle URL«. Die Steuerung über den Webservice ist Inhalt der Subroutine *sStartPerWebservice* und wird im Abschnitt »Die Schnittstelle Webservice« beschrieben.

Bevor Sie die beiden Subroutinen nun mit dem entsprechenden Code ergänzen, sollten Sie die bisher entwickelte Funktionalität des Windows-Forms testen. Starten Sie das Windows-Form über den Menüpunkt *Debuggen/Debuggen starten*.



Abbildung 43.1 Das Windows-Form *Druckdialog*

Soweit, so gut. Im nächsten Abschnitt werden Sie den Druckdialog um die Funktion der Berichtsgenerierung über die URL-Schnittstelle erweitern.

Die Schnittstelle URL

Den Zugriff auf einen Bericht über die URL-Schnittstelle ist im Grunde genommen nichts Neues, denn Sie nutzen diese Technik bereits im Berichts-Manager. Dort werden die jeweiligen Berichte des Berichtsservers mit deren Bezeichnung aufgelistet. Diese Berichtsbezeichnung ist nichts anderes als ein Link, der auf den Speicherort des Berichts innerhalb des Berichtsservers verweist. Über diesen Link öffnen Sie den Bericht in der Berichtsanzeige.

Dasselbe Prinzip gilt für die Integration der URL-Schnittstelle in Ihrer Applikation. Sie müssen lediglich den Link zum Bericht an eine Anwendung – in diesem Fall an einen Browser – übergeben und der Bericht wird in der Berichtsanzeige geöffnet.

Es stehen mehrere Parameter zur Verfügung, mit denen die Berichtsgenerierung beeinflusst werden kann. Ob der Bericht nun in ein anderes Format gerendert, die Konfiguration der Berichtsanzeige modifiziert oder Berichtsparameter übergeben werden sollen – für alle diese Optionen stehen Parameter zur Verfügung. Damit diese Optionen bei der Berichtsgenerierung umgesetzt werden, muss der Link mit den entsprechenden Parametern erweitert werden.

Unter anderem steuern Sie auch die verschiedenen Rendervorgänge mit Parametern. Diese werden Sie nun in der Subroutine *sStartPerUrl* nutzen.

Bericht öffnen

Das Beispiel *Druckdialog* soll die Generierung des Berichts *Kundenliste* in verschiedene Formate ermöglichen. Nach dem Start des Dialogs ist das Format *Online-Bericht* bereits selektiert. Anhand dieser Auswahl soll der Bericht in der Berichtsanzeige – die Sie bereits aus dem Berichts-Manager kennen – geöffnet werden.

Um dies zu erreichen, muss in der Subroutine *sStartPerUrl* der Link zum Bericht *Kundenliste* aufgerufen werden. Durch diesen Aufruf wird der Browser gestartet und der Bericht als Online-Bericht für die Berichtsanzeige generiert. Ergänzen Sie dazu die Subroutine mit den Zeilen aus Listing 43.4.

```
Private Sub sStartPerUrl(ByVal strFormat As String)

    Dim strUrl As String = "http://localhost/Reportserver?/Bestellungen/Kundenliste"

    System.Diagnostics.Process.Start(strUrl)

End Sub
```

Listing 43.4 Online-Bericht per URL starten

Der Link besteht aus zwei Teilen. Der erste Teil verweist auf den Berichtsserver², während der zweite Teil, der durch ein Fragezeichen eingeleitet wird, den Link zum jeweiligen Bericht enthält. Somit wird über den Link *http://localhost/Reportserver?/Bestellungen/Kundenliste* der Bericht *Kundenliste* aus dem Verzeichnis *Bestellungen* vom Berichtsserver *Reportserver*, der auf dem Webserver *localhost* zur Verfügung gestellt wird, gestartet. Der Link wird an die Methode *System.Diagnostics.Process.Start(strUrl)* übergeben, die zunächst den Browser startet und anschließend die Berichtsanzeige öffnet.

Um das Öffnen des Berichts über die URL-Schnittstelle zu testen, starten Sie den Druckdialog wieder über den Menüeintrag *Debuggen/Debuggen starten* und klicken Sie auf die Schaltfläche *Bericht erstellen*. Der Browser wird mit dem übergebenen Link geöffnet.

² Den Pfad zu dem Berichtsserver »http://localhost/Reportserver« müssen Sie ggf. an Ihre Installation anpassen.

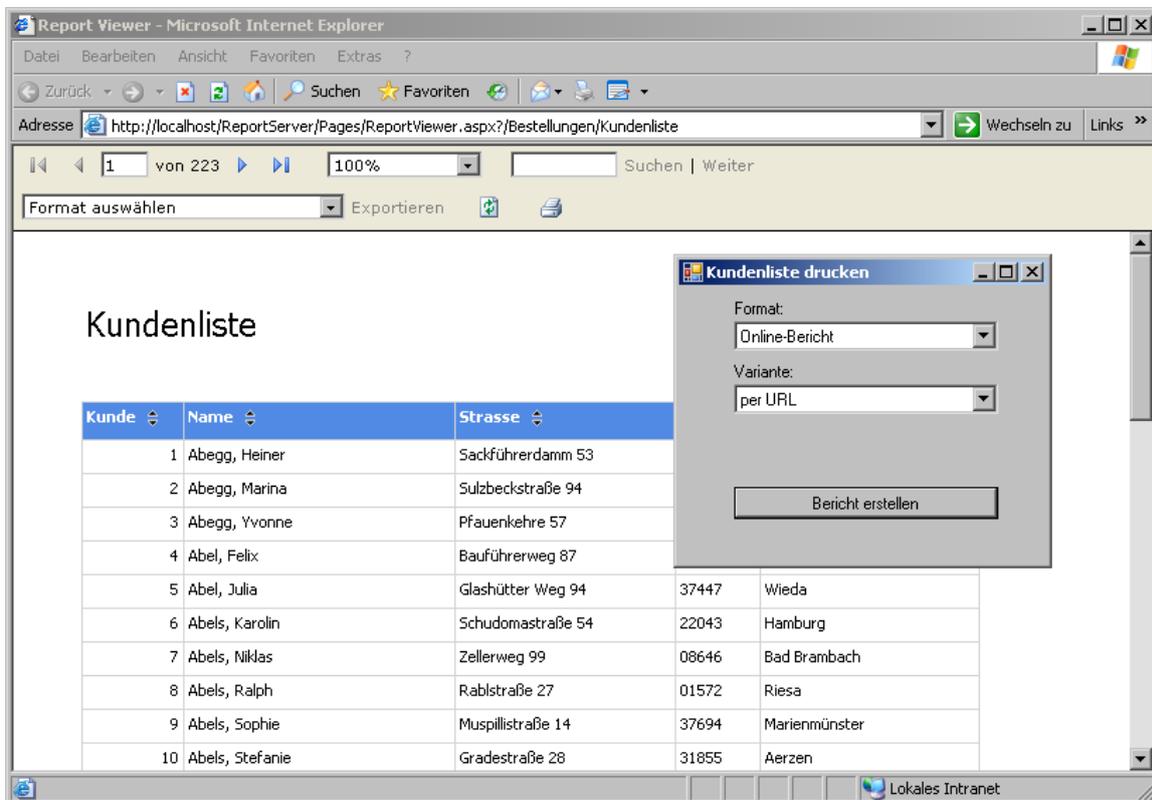


Abbildung 43.2 Der Online-Bericht über den Druckdialog

In der Berichtsanzeige sehen Sie nun den Bericht *Kundenliste* als Online-Bericht.

Der Druckdialog bietet aber noch weitere Formate an, in die der Bericht ausgegeben werden kann. Im nächsten Abschnitt werden Sie den Link in der Subroutine um die jeweiligen Parameter für die verschiedenen Formate erweitern.

Bericht rendern

Sofern im Link lediglich der Pfad des Berichtsservers und des Berichts angegeben ist, wird der Bericht als Online-Bericht gerendert. Nun kann in der Berichtsanzeige der Bericht über die Funktion *Exportieren* zwar in andere Formate gerendert werden, aber dieser Umweg ist nicht notwendig.

Durch den Parameter *rs:Format=<FormatBezeichnung>* wird der Bericht direkt im angegebenen Format generiert. Dieser Parameter wird am Ende des Links angehängt und mit dem »&<-Zeichen eingeleitet. Der folgende Link liefert den Bericht *Kundenliste* im Excel-Format.

```
http://localhost/Reportserver?/Bestellungen/Kundenliste&rs:Format=Excel
```

Da das Listenfeld mehrere Ausgabeformate zur Auswahl stellt, bietet sich im Sourcecode eine *CASE*-Anweisung an, anhand der die für die Auswahl notwendigen Formate gesetzt werden. In Listing 43.5 sehen Sie die erweiterte Subroutine.

```
Private Sub sStartPerUrl(ByVal strFormat As String)
    'Startet den Bericht über den URL-Zugriff

    Dim strUrl As String = "http://localhost/Reportserver?/Bestellungen/Kundenliste"

    Select Case strFormat
        Case "CSV"
            strUrl = strUrl & "&rs:Format=CSV"
        Case "Excel"
            strUrl = strUrl & "&rs:Format=Excel"
        Case "HTML"
            strUrl = strUrl & "&rs:Format=HTML4.0"
        Case "PDF"
            strUrl = strUrl & "&rs:Format=PDF"
        Case "Online-Bericht"
            'Für den Start des Reportviewers wird kein Format benötigt.
        Case "XML"
            strUrl = strUrl & "&rs:Format=XML"
    End Select
    System.Diagnostics.Process.Start(strUrl)
End Sub
```

Listing 43.5 Rendern per URL

Mit dieser Änderung können Sie den Bericht nun in den Formaten *CSV*, *Excel*, *HTML*, *PDF* und *XML* erzeugen oder als Online-Bericht in der Berichtsanzeige anzeigen lassen.

Öffnen Sie den Druckdialog erneut über den Menübefehl *Debuggen/Debuggen starten* und wählen Sie das Format *Excel* aus dem Listenfeld *Format* aus. Wieder wird zunächst der Browser mit dem entsprechenden Link gestartet – aber nun mit dem Unterschied, dass zusätzlich der Dialog *Dateidownload* geöffnet wird.



Abbildung 43.3 Berichts-
ausgabe im Excel-Format

Der Dialog bietet Ihnen – neben der Möglichkeit den Vorgang abubrechen – die Optionen, die Datei zu speichern oder direkt zu öffnen.

HINWEIS Beim direkten Öffnen der Datei wird diese nicht automatisch gespeichert. Wenn Sie die Datei also ohne vorheriges Speichern wieder schließen, wird diese gelöscht.

Das Listenfeld *Format* bietet neben den Formaten, die bereits in der *CASE*-Anweisung berücksichtigt wurden, noch die beiden Formate *Image als TIFF* und *Image als JPEG* an. Für das Rendern von Bildern gibt es einen kleinen aber feinen Unterschied.

Bericht in Bildformate rendern

Durch den Parameter *rs:Format=IMAGE* wird ein Bild immer im TIFF-Format gerendert. Sofern das Bild aber in einem anderen Format – beispielsweise als JPEG – erzeugt werden soll, muss der Parameter um einen weiteren Parameter ergänzt werden.

Mit diesem Parameter werden Informationen über die Renderingweiterungen an den Berichtsserver übergeben. Durch die Ergänzung *&rc:OutputFormat=JPEG* wird der Bericht als Bild im JPEG-Format erstellt. Auch hier wird der zusätzliche Parameter durch das »&«-Zeichen eingeleitet.

Der komplette Link für die Ausgabe des Berichts im JPEG-Format lautet somit

```
http://localhost/Reportserver?/Bestellungen/Kundenliste&rs:Format=IMAGE&rc:OutputFormat=JPEG
```

Nun soll der Druckdialog die Ausgabe sowohl in das TIFF- wie auch in das JPEG-Format unterstützen. Dazu muss die *CASE*-Anweisung um zwei Einträge erweitert werden. Während für die JPEG-Ausgabe der Link mit den beiden Parametern *rs:Format=IMAGE&rc:OutputFormat=JPEG* ergänzt werden muss, reicht für die Ausgabe im TIFF-Format die Angabe des Parameters *rs:Format=IMAGE*. Den kompletten Inhalt der Subroutine sehen Sie in Listing 43.6.

```
Private Sub sStartPerUrl(ByVal strFormat As String)
    'Startet den Bericht über den URL-Zugriff

    Dim strUrl As String = "http://localhost/Reportserver?/Bestellungen/Kundenliste"

    Try

        Select Case strFormat
            Case "CSV"
                strUrl = strUrl & "&rs:Format=CSV"
            Case "Excel"
                strUrl = strUrl & "&rs:Format=Excel"
            Case "HTML"
                strUrl = strUrl & "&rs:Format=HTML4.0"
            Case "Image als JPEG"
                'Alle Bildformate außer TIFF benötigen den Parameter OutputFormat
                strUrl = strUrl & "&rs:Format=IMAGE&rc:OutputFormat=JPEG"
            Case "Image als TIFF"
                strUrl = strUrl & "&rs:Format=IMAGE"
            Case "PDF"
                strUrl = strUrl & "&rs:Format=PDF"
        End Select
    End Try
End Sub
```

```

        strUrl = strUrl & "&rs:Format=PDF"
    Case "Online-Bericht"
        'Für den Start des Reportviewers wird kein Format benötigt.
    Case "XML"
        strUrl = strUrl & "&rs:Format=XML"
    End Select
    If strUrl = "" Then
        MsgBox("Der angegebene Link ist nicht korrekt.")
    Else
        System.Diagnostics.Process.Start(strUrl)
    End If

Catch ex As Exception
    MsgBox(Err.Number.ToString & " - " & Err.Description)
End Try

End Sub

```

Listing 43.6 Die Subroutine sStartPerUrl

Die Programmierung des Zugriffs über die URL-Schnittstelle wäre hiermit abgeschlossen. Bevor Sie nun mit der Webservice-Schnittstelle weitermachen, folgt noch ein kurzer Einblick auf die weiteren Parameter, die die URL-Schnittstelle zur Steuerung des Berichtsservers und der Berichte anbietet.

Parameter der URL-Schnittstelle

Die URL-Schnittstelle stellt zwei verschiedene Parametertypen zur Verfügung, über die der Berichtsserver und die Berichte gesteuert werden können. Diese Parametertypen sind durch die Präfixe *rs* und *rc* gekennzeichnet.

Das Präfix *rs* steht für *ReportServer* und bietet Parameter und Befehle zur Steuerung des Berichtsservers, während das Präfix *rc* für *ReportControl* steht und Parameter und Befehle für die Steuerung der Berichtsgenerierung anbietet.

Mit diesen Parametern – die sich grob in vier Gruppen einteilen lassen – können die Berichte, der Berichtsserver, die Berichtsanzeige und die Anmeldung gesteuert werden.

Parametertyp *rc* und die Renderingweiterungen

Mit dem Parameter *rc:OutputFormat=JPEG* haben Sie bereits eine der Renderingweiterungen kennen gelernt, die Sie über den Parametertyp *rc* steuern können. Es stehen Ihnen für die verschiedenen Formate vielfältige Erweiterungsmöglichkeiten zur Verfügung.

So können Sie beispielsweise für die Ausgabe in das CSV-Format mit dem Parameter *rc:FieldDelimiter* das Zeichen für die Spaltentrennung bestimmen oder mit *rc:NoHeader* die Ausgabe der Kopfzeilen verhindern.

Mit den Parametern *rc:MarginLeft*, *rc:MarginRight*, *rc:MarginBottom* und *rc:MarginTop* sowie *rc:PageHeight* und *rc:PageWidth* bestimmen Sie die Seitenränder und Seitengröße bei der Ausgabe der Formate *Image* und *PDF*.

Alle verfügbaren Renderingweiterungen können hier nicht beschrieben werden, da sie für dieses Buch zu umfangreich sind. Die Renderingweiterungen sind in der Online-Hilfe der Reporting Servi-

ces ausführlich beschrieben. Sie finden diese unter dem Suchbegriff *device information settings [Reporting Services]*.

Parametertyp rc und die Steuerung der Berichtsanzeige

Neben den Rendering-erweiterungen stellt der Parametertyp *rc* auch Parameter zur Steuerung der Berichtsanzeige zur Verfügung. Diese sehen sie in Tabelle 43.1.

Parameter	Beschreibung
rc:Toolbar	Blendet die Symbolleiste der Berichtsanzeige ein bzw. aus.
rc:Parameters	Blendet die Parameterauswahl der Berichtsparameter ein bzw. aus.
rc:DocMap	Blendet die DocumentMap ein bzw. aus.
rc:DocMapId	Generiert den Bericht und verzweigt direkt zu den Daten der angegebenen Id.
rc:Zoom	Bestimmt den Zoomfaktor zur Ausgabe des Berichts.
rc:Section	Generiert den Bericht und verzweigt direkt zu der im Parameter angegebenen Seite. Wird eine Zahl angegeben, die höher als die Gesamtseitenzahl ist, wird die letzte Seite angezeigt.
rc:FindString	Generiert den Bericht, sucht den im Parameter angegebenen Begriff und verzweigt direkt zur ersten Seite, in der der Begriff gefunden wurde.
rc:StartFind	Bestimmt, ab welcher Seite der Begriff aus <i>rc:FindString</i> gesucht werden soll.
rc:EndFind	Bestimmt, bis zu welcher Seite der Begriff aus <i>rc:FindString</i> gesucht werden soll.
rc:FailbackPage	Bestimmt die Seite, die aktiviert werden soll, wenn die unter <i>rc:DocMapId</i> oder <i>rc:StartFind</i> angegebenen Werte nicht gefunden wurden.
rc:Stylesheet	Bestimmt das Stylesheet, das zur Darstellung des Berichts in der Berichtsanzeige verwendet werden soll. Dabei muss es sich um eine gültige CSS-Datei handeln, die im Verzeichnis <i>Styles</i> des Installationsverzeichnis des Berichtsservers gespeichert sein muss.

Tabelle 43.1 Steuerung der Berichtsanzeige

Parametertyp rs und die Steuerung des Berichtsservers

Der Parametertyp *rs* bietet Parameter zur Steuerung des Berichtsservers an. Neben dem Ihnen bereits bekannten Parameter *rs:Format* stehen noch die Parameter *rs:Command*, *rs:ParameterLanguage* und *rs:Snapshot* zur Verfügung.

Mit dem Parameter *rs:Command* können Sie die Eigenschaften von Elementen des Berichtsservers – wie Berichte, Datenquellen, Ressourcen und weitere – auslesen und in bestimmte Formate rendern. So erzeugt der Befehl *GetDataSourceContents*, den Sie über den Parameter *rs:Command* an den Berichtsserver übergeben, eine XML-Datei, die die Eigenschaften der im Link angegebenen Datenquelle beinhaltet. Der Link

```
http://localhost/reportserver?/Datenquellen/Bestellungen
&rs:Command=GetDataSourceContents
```

liefert Ihnen also die Eigenschaften der Datenquelle *Bestellungen* in einer XML-Datei.

HINWEIS

Um die nachfolgenden Beispiele zu testen, öffnen Sie Ihren Browser und geben Sie dort den jeweiligen Link direkt ein.

Der Befehl *GetResourceContents* rendert die angegebene Ressource in eine HTML-Seite. Folgender Link würde die Ressource *Firmenlogo* – das Logo als Bitmap – in einer HTML-Seite darstellen.

```
http://localhost/reportserver?/Bestellungen/Firmenlogo
&rs:Command=GetResourceContents
```

Den Befehl *Render* haben Sie bereits mehrmals ausgeführt. Dies ist der Standardbefehl, der immer dann genutzt wird, wenn der Parameter *rs:command* nicht im Link angegeben wird.

Alle untergeordneten Elemente eines Berichtserver-Elements werden über den Befehl *ListChildren* in einer HTML-Seite ausgegeben. Mit dem folgenden Link erhalten Sie eine HTML-Seite mit allen Elementen des Ordners *Bestellungen*.

```
http://localhost/reportserver?/Bestellungen&rs:Command=ListChildren
```

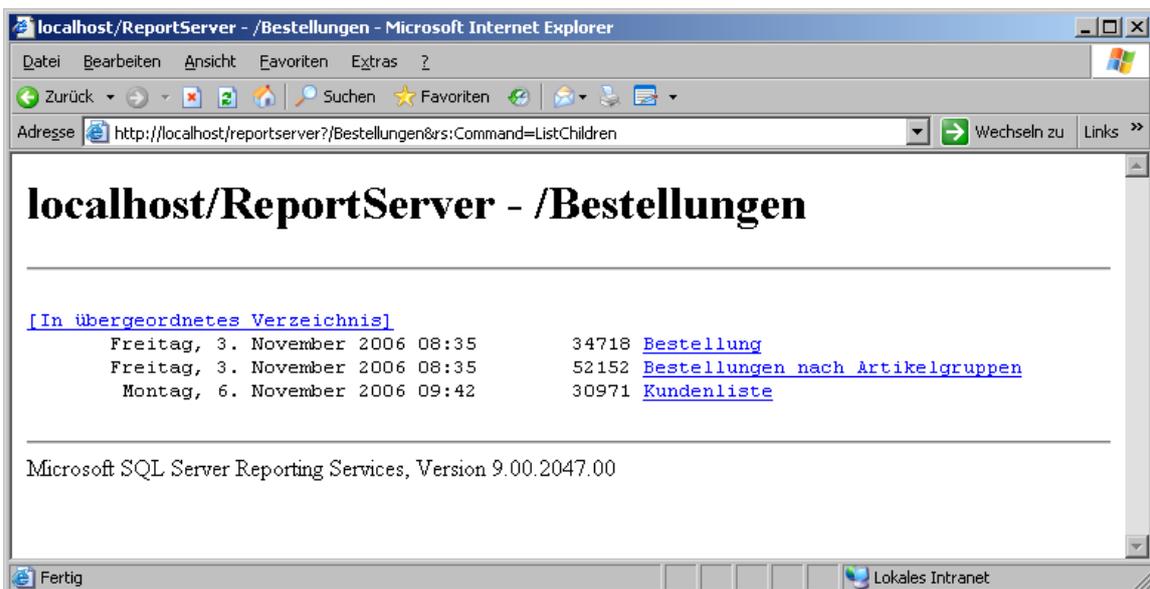


Abbildung 43.3 Der Parameter `rs:command=ListChildren`

Mit dem Parameter *rs:ParameterLanguage* steuern Sie die länderspezifischen Formate der übergebenen Berichtparameter. Diese werden normalerweise mit der länderspezifischen Einstellung des Browser übergeben. Gerade bei länderspezifischen Einstellungen wie dem Datumsformat kann dies aber zu Problemen führen. Als Beispiel sollen über folgenden Link alle Bestellungen vom 09.06.2005 bis zum 12.09.2005 ausgegeben werden.