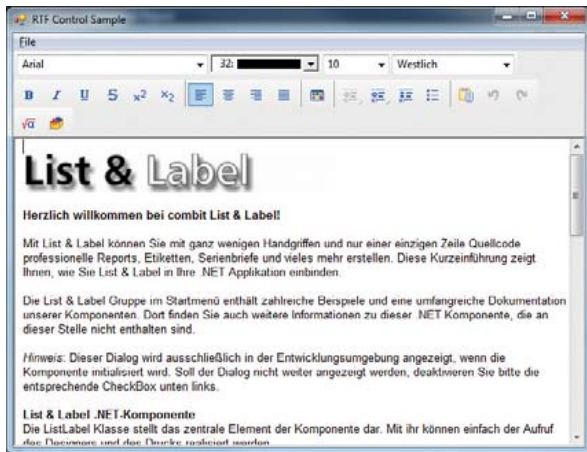


komponente, die über eine Symbolleiste zur Formatierung verfügt und mit Formeln umgehen kann. Vorausgesetzt, Sie weisen dem Control ein ListLabel-Objekt zu und setzen den Projekttyp des ListLabel-Objekts auf Card:

```
listLabel1.AutoProjectType =
    LlProject.Card;
listLabelRTFControl1.ParentComponent =
    listLabel1;
```

Die einzelnen Menübefehle lassen sich durch Aufruf der ProhibitAction-Methode ausblenden. Zur Selektion der Schaltfläche erwartet die Methode einen Wert der LLRTFEditorAction-Enumeration. Dieses Control eignet sich insbesondere im Zusammenspiel mit von Ihnen vorgefertigten Berichten, bei denen die Anwender eingeschränkte Gestaltungsmöglichkeiten erhalten sollen.



Mit dem RTF-Steuerelement bestimmt der Benutzer den Inhalt

Dokumentvorschau, Druck und Webpräsentation

Mittels des ListLabelPreviewControl-Steuerelements integrieren Sie die Berichtsvorschau mit Echtdateien nahtlos in Ihre Anwendung. Im Lieferumfang finden Sie eine Windows-Forms- und eine Windows-Presentation-Foundation- (WPF-)Variante des Vorschau-Steuerelements. Ebenso wie der RTF-Editor ist das Steuerelement an ein ListLabel-Objekt gebunden, über das Sie die Ausgabeparameter steuern. Folgender Code zeigt beispielsweise die Vorschau eines Berichts an:

```
listLabel1.DataSource = DataAccess.Current.GetOrdersDataSet();
listLabel1.AutoDestination = LlPrintMode.PreviewControl;
listLabel1.PreviewControl = listLabelPreviewControl1;
listLabel1.AutoDesignerPreview = true;
listLabel1.AutoProjectFile = Path.Combine(sAppDir, "Orders.lst");
listLabel1.AutoProjectType = LlProject.List;
listLabel1.AutoShowPrintOptions = false;
listLabel1.AutoShowSelectFile = false;
listLabel1.Print();
```

Nachdem die Datenquelle angebunden ist, werden die Ausgabe auf das Vorschau-Steuerelement umgelenkt, der Projekttyp und die Berichtsdefinitionsdatei zugewiesen und die Projektauswahl deaktiviert, um zu verhindern, dass der Anwender die Möglichkeit hat, selbst eine Berichtsdefinitionsdatei auszuwählen. Anschließend veranlasst die Print-Methode die Ausgabe des Berichts. Eine Alternative zum Steuerelement ist ein eigenständiger Vorschau-dialog. Setzen Sie hierfür die AutoDestination-Eigenschaft auf den Wert LlPrintMode.Preview.

Die Vorschau erlaubt eine Navigation durch alle Berichtsseiten anhand von Thumbnails und unterstützt sowohl den Druck als auch den Export des Berichts. Mit Version 16 hat Combit der Vorschau auch eine Suchfunktion spendiert. Auf Wunsch kann das Steuerelement auf das Wesentliche reduziert werden – die Anzeige des Berichts – und die Befehle zur Navigation, zum Druck oder zum Export in eigene Menüs auslagern.

Die Darstellung des Berichts als Vorschau basiert intern auf einem Format, welches das Layout aus der Berichtsdefinitionsdatei mit den gebundenen und ungebundenen Daten verknüpft. Sie können den Bericht in diesem Format auch als Vorschau-datei exportieren, um dann beispielsweise mehrere

Dateien zusammenzuführen. Die resultierende Vorschau-datei ist wiederum druck- und exportfähig:

```
//.ctor-Parameter: Dateipfad, ReadOnly
PreviewFile cover = new
    PreviewFile(Path.Combine(
        sDictionary, "deckblatt.ll", false);
PreviewFile report = new
    PreviewFile(Path.Combine(
        sDictionary, "bericht.ll", true);

//Bericht an Deckblatt anhängen
cover.Append(report);

//Gesamtbericht produzieren
cover.Print();

//Als PDF exportieren
cover.ConvertTo(Path.Combine(
    sDictionary, "Report.pdf"));
```

Möchten Sie vollständig auf die von Combit gelieferten Vorschau-dialoge und -komponenten verzichten, bietet sich die ListLabelDocument-Klasse als Alternative an. Die Klasse erbt von PrintDocument und repräsentiert ein erzeugtes Druckprojekt, das auf den List-&-Label-Vorschau-dateien basiert. Infolgedessen sind Objekte dieses Typs mit den Vorschau-klassen des .NET Frameworks kombinierbar.

Der ListLabelWebView ist eine ASP.NET-Komponente, die wie die ListLabelPreviewControl- oder ListLabelDocument-Klassen List-&-Label-Vorschau-dateien verarbeitet und im Browser anzeigt.

Der Designer

Mit dem autonomen Designer, den Sie wahlweise zur Entwurfszeit aus Visual Studio oder zur Laufzeit aus Ihrer Anwendung aufrufen, visualisieren Sie gebundene und ungebundene Daten, fügen weitere Designelemente wie Charts, Barcodes oder OLE-Objekte ein und kontrollieren Ihre Arbeit direkt in der Vorschau mit den Echtzeit-dateien. Neben einem reinen Layout- und Vorschau-modus bietet der Berichtsgenerator einen gemischten Modus, in dem die Daten mit sämtlichen Layoutfunktionen verknüpft sind. Für noch schnelleres Gestalten sorgt die neue Miniatur-symbolleiste, die sich bei selektierten Objekten einblendet und zur Kontrolle von Schriftart und -größe oder des Absatzstils dient.



Minisymbolleiste zur Schnellformatierung

Denkbare Einsatzgebiete für List-&-Label-Berichte reichen vom Erzeugen einfacher Briefe oder Listen über den Druck von Barcodes und Etiketten bis hin zu Formularen mit interaktiven Inhalten. Alle diese Berichte basieren auf einem der folgenden drei Projekttypen:

- Listen
- Etiketten
- Karteikarten

Der Listenprojekttyp ist der gängige Einstiegspunkt für alle Anforderungsfälle, die eine Liste oder eine tabellarische Darstellungsform fordern. Listen, Tabellen und Kreuztabellen werden im Designer durch das Berichtscontainer-Element

