List & Label 16 unter der Lupe

100 Prozent Reporting-Power

Die stetige Verarbeitung und Aufbereitung von Daten in Form von Berichten ist unabdingbar für die strategische und operative Unternehmensführung. Wie das Konstanzer Unternehmen Combit Ihnen und Ihren Anwendern mit List & Label 16 dabei unter die Arme greift, zeigt DevDorado.

List & Label hat sich in den vergangenen siebzehn Jahren als eines der bekanntesten Produkte im Reporting-Bereich etabliert. Viele gewonnene Leserwahlen sprechen für die Qualität von List & Label, das Anwendungen schnell und unkompliziert um flexible Berichte sowie Druck- und Exportfunktionalitäten erweitert. Bei Bedarf stellt List & Label auch den selbsterklärenden Berichtsgenerator für Anwender zur Verfügung.

Combit liefert für eine Vielzahl von Programmiersprachen entsprechende Bibliotheken und Steuerelemente. Das Spektrum reicht von allen Programmiersprachen des .NET Frameworks über Java und Delphi bis hin zu C++ oder Visual Basic für 32- und 64-Bit-Systeme. Auch die Präsentation im Web ist mit List & Label möglich.

Die Konstanzer haben die sechzehnte Generation der Berichtskomponenten nicht nur mit vielen neuen und nützlichen Funktionen ausgestattet, sondern konzentrieren sich mit der neuen Version auch verstärkt auf die Optimierung der Benutzerfreundlichkeit und die nahtlose Integration in Windows 7. Mit den neuen Messinstrumenten (Gauges) decken Berichte den aktuellen Geschäftsstand auf. Damit die ansprechenden Designs auch beim Export erhalten bleiben, unterstützt List & Label 16 den Export nach XHTML in Verbindung mit Cascading Style Sheets (CSS) oder in das PNG-Grafikformat. Ferner hat Combit das Barcorde-Repertoire um den Intelligent Mail Barcode (IMB) erweitert, der insbesondere in den vereinigten Staaten für Postsendungen Verwendung findet.

Nach der Installation von List & Label offeriert die Visual-Studio-Toolbox dem .NET-Entwickler insgesamt sechs Komponenten:

- ListLabel
- DataSource
- ListLabelRTFControl
- ListLabelPreviewControl
- ListLabelDocument
- ListLabelWebView

Die ListLabel-Klasse

Dreh- und Angelpunkt ist die ListLabel-Klasse, welche die Design-, Druck- und Exportfunktionen von List & Label vereint. Mit Hilfe dieser Klasse rufen Sie den Berichtsgenerator auf, binden Daten an die Berichte oder veranlassen den Druck. Die Konfiguration der Datenquelle oder der Berichtsdefinitionsdatei erfolgt im Visual Studio Designer, der Sie mit Smarttags unterstützt. Alternativ genügen wenige Codezeilen, um die Daten zu binden und den Berichtsgenerator aufzurufen:

```
ListLabel listLabel = new ListLabel();
listLabel.DataSource =
  DataAccess.Current.GetOrdersDataSet();
listLabel.Design();
```

Die Datenbindung ist eine Voraussetzung für den Aufruf des Designers, der sich immer als modaler Dialog präsentiert. List & Label verarbeitet alle Datenquellen, welche die Schnittstelle IDataProvider des Namensraums combit.ListLabel16.DataProviders implementieren. Zur Verarbeitung eines DataSets, wie im obigen Beispiel zu sehen, bedient sich List & Label intern der AdoDataProvider-Klasse. Jene Klasse unterstützt DataSet-, DataTable-, DataView- und DataViewManager-Objekte. Neben der AdoDataProvider-Klasse liefert Combit beispielsweise zur Unterstützung von Business-Objekten oder XML-Dateien die Klasse ObjectDataProvider respektive XmlDataProvider. Der Object-Data-Provider nutzt die IEnumerable-, IEnumerable<T>- und IListSource-Schnittstellen und ist demzufolge auch LINQ-kompatibel. Die Darstellung der Eigenschaften beeinflussen Sie mittels der ITypedList-Schnittstelle oder dem Browsable-Attribut.

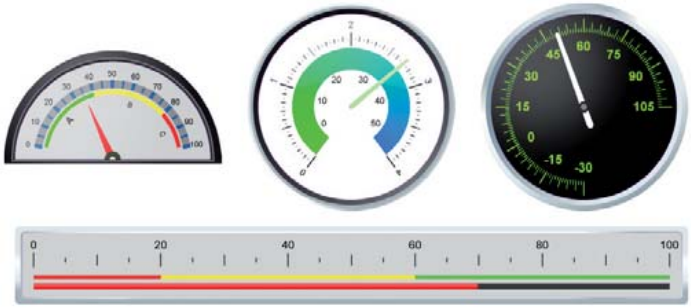
Weitere Daten-Provider erlauben Ihnen den Verbindungsaufbau zum SQL Server und zu Oracle-Datenbanken sowie die Verarbeitung nichtrelationaler Daten via OLE DB.

Datenquelle ohne Code binden

Die DataSource-Komponente konsolidiert mehrere Data-Provider und offeriert zugleich eine grafische Oberfläche zur Konfiguration der Verbindungsparameter. Der Smarttag legt alle wichtigen Einstellungsmerkmale offen. Im Eigenschaftenfenster des ListLabel-Objekts oder über dessen Smarttag wählen Sie das neu angelegte DataSource-Objekt aus. Im Anschluss finden Sie eine Liste aller Tabellen beziehungsweise Sichten der Datenbank im Editor der DataMember-Eigenschaft. Im Bericht stehen Ihnen nicht etwa nur die Datensätze der ausgewählten Tabelle zur Verfügung sondern auf Wunsch auch die über Relationen verknüpften Datensätze.

Das Rich-Text-Format-Steuerelement

Das ListLabelRTFControl kommt, nachdem Sie es auf das Formular gezogen haben, sehr unscheinbar daher, denn es ist zur Entwurfszeit nicht zu sehen. Zur Laufzeit entpuppt sich das Steuerelement als vollwertige RTF-Bearbeitungs-



Die neuen Gauges von List & Label 16

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 Echtdaten nahtlos in Ihre Anwendung. Im Lieferumfang finden Sie eine Windows-Forms- und eine Windows-Presentation-Foundation- (WPF-)Variante des Vorschausteuerelements. 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 Vorschausteuerelement 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 Vorschaudialog. 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 Vorschaudatei exportieren, um dann beispielsweise mehrere Dateien zusammenzuführen. Die resultierende Vorschaudatei 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 Vorschaudialoge 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-Vorschaudateien basiert. Infolgedessen sind Objekte dieses Typs mit den Vorschauklassen des .NET Frameworks kombinierbar.

Der ListLabelWebViewer ist eine ASP.NET-Komponente, die wie die ListLabelPreviewControl- oder ListLabelDocument-Klassen List-&-Label-Vorschaudateien 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 Echtzeitdaten. Neben einem reinen Layout- und Vorschaumodus bietet der Berichtsgenerator einen gemischten Modus, in dem die Daten mit sämtlichen Layoutfunktionen verknüpft sind. Für noch schnelleres Gestalten sorgt die neue Miniatursymbolleiste, 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

dargestellt. Der Berichtscontainer ist nur in diesem Projekttyp verfügbar. Die gebundenen Daten, etwa die Datensätze eines DataSet-Objekts, bietet der Designer in der Variablenliste als Felder an. Sie stehen nur Berichtscontainer-Elementen zur Verfügung. Hingegen handelt es sich in der List-&-Label-Terminologie bei Variablen um Daten, die sich in der Regel nur einmal pro Seite oder pro Bericht ändern und von der Anwendung „manuell" an die Berichtsverarbeitung übergeben werden. Der Zugriff auf mehrere Quelltabellen ist in den Projekttypen Etiketten und Karteikarten nicht möglich. Ist eine Datenquelle mit mehreren Tabellen an die Verarbeitungseinheit gebunden, kann lediglich eine Tabelle als Quelle fungieren.

Mit dem Etikettenprojekttyp unterstützt List & Label über 2.000 Standardetikettenvorlagen deutscher und internationaler Hersteller. Ein auf dem Etikettenprojekttyp basierender Spezialfall sind die Karteikartenprojekte, die hauptsächlich auf die Umsetzung von Berichten im Karteikartenstil, wie zum Beispiel die Darstellung aller Informationen eines Kunden, sowie von Serienbriefen abzielen.

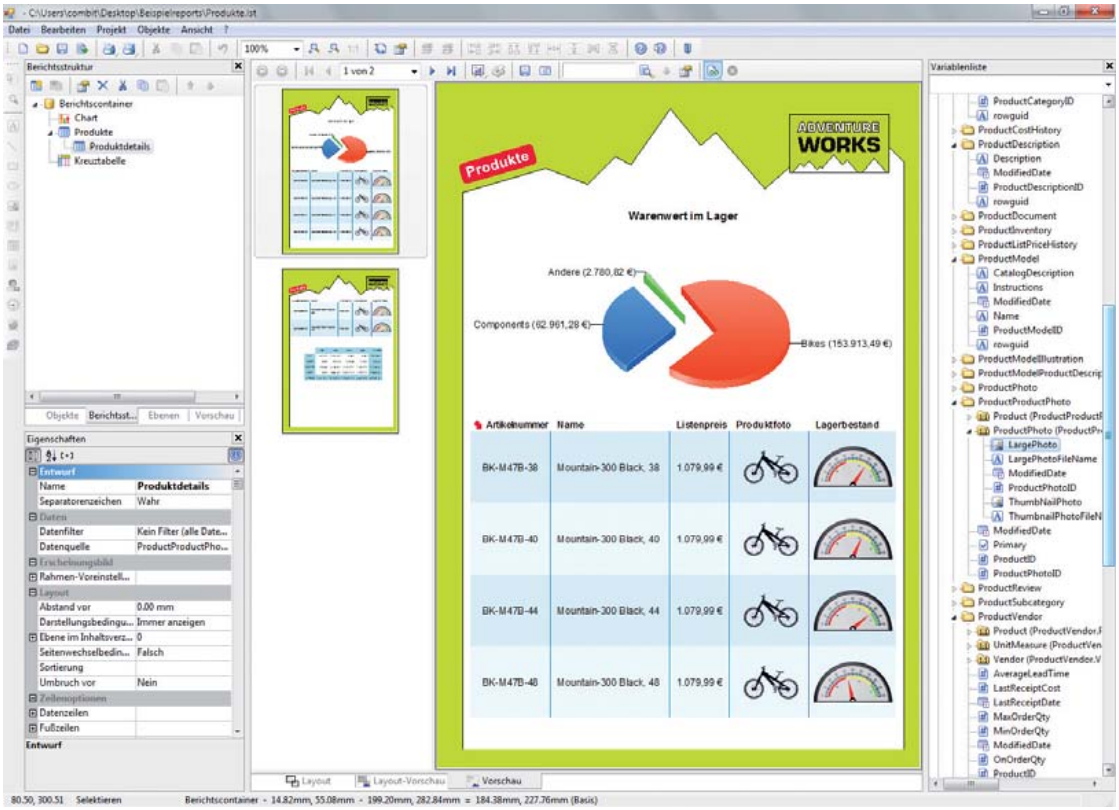
Da der Designer in den Redistributionsdateien von List & Label enthalten und über Ihre Anwendung aufrufbar ist, eignet sich der Berichtsgenerator ebenso für Ihre Anwender um bestehende Berichte an individuelle Bedürfnisse anzupassen oder gar neue Berichte auf Ihrer Datenbasis anzulegen. Die Kontrolle geht hierbei nie verloren, denn Sie können den Designer in seinem Funktionsumfang durch Ausblenden von Menüeinträgen, dem Deaktivieren von Funktionen oder dem Sperren von Berichtsobjekten anpassen. Neue Berechnungsfunktionen oder die Auslagerung von Aktionsbehandlungen in den Anwendungscode ersetzen oder erweitern das Funktionsrepertoire des Berichtsgenerators.

Jens Konerow ist Microsoft Student Partner an der Universität zu Lübeck, mehrfacher Buchautor und Entwickler bei der KEEP IT SIMPLE GmbH.

Sie erreichen ihn unter webmaster@jenskonerow.de.

Fazit

Die List-&-Label-Reporting-Tools glänzen mit einem breit gefächerten Einsatzspektrum, einem soliden Berichtsgenerator und einer sehr guten elektronischen sowie gedruckten API- und Designer-Dokumentation. Nach einer kurzen Eingewöhnungsphase an die für .NET-Framework-Entwickler etwas ungewöhnlichen Programmierschnittstelle und den leistungsstarken Designer sind die Berichte schnell in die eigene Anwendung integriert und bieten Anwendern jeglichen Komfort.



Der Berichtsgenerator im Gesamteindruck