

## Chương 7:

# BIỂU ĐỒ VÀ CÁC THÀNH PHẦN TRỰC QUAN

- ❖ *Tổng quan.*
- ❖ *Thêm biểu đồ vào báo biểu.*
- ❖ *Định dạng nâng cao.*
- ❖ *Tóm tắt.*

## 1. TỔNG QUAN

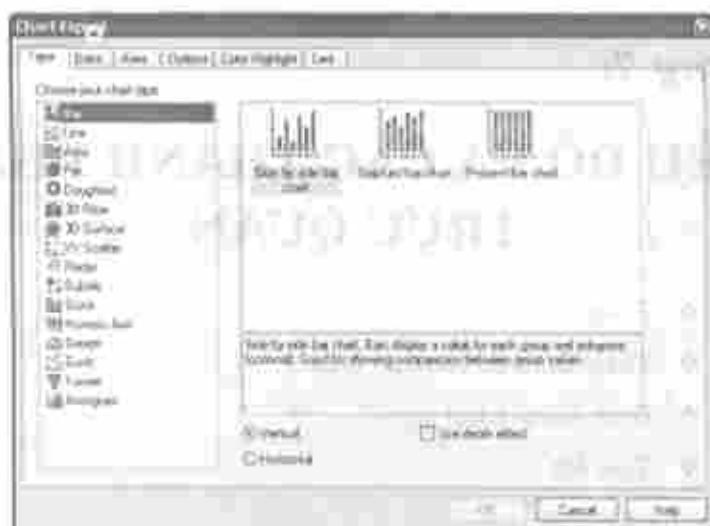
Dữ liệu ngày càng bùng nổ trong các tổ chức, công ty, xí nghiệp. Phần lớn những ứng dụng thương mại và các hệ thống thông tin đều liên quan đến cơ sở dữ liệu (CSDL). Một tổ chức có thể có thu thập hàng triệu mẫu tin để đánh giá phân tích thống kê. Crystal Reports là một giải pháp tất yếu để giải quyết vấn đề này, mỗi một báo biểu có thể được hình thành từ hàng triệu dòng thông tin theo một khuôn dạng gọn gàng và sáng sủa.

Bạn đã xem qua một số cách mà Crystal Reports thể hiện thông tin trên báo biểu, như tổng gộp số liệu, báo biểu phân cấp, tổng chạy, báo biểu ma trận Cross-Tab. Tất cả những hình thức trình bày dữ liệu này đều dưới dạng văn bản và con số.

Trong chương này, chúng ta sẽ khảo sát cách trình bày dữ liệu trực quan khác dưới dạng trực quan bằng hình ảnh thông qua biểu đồ và đồ thị. Với biểu đồ, bạn có thể đúc kết hàng triệu dòng dữ liệu vào trong một hình mô tả dễ nhận biết và phân tích hơn (hình 7.1).

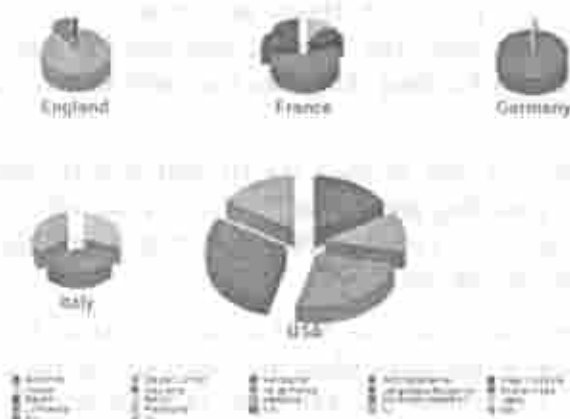
Bộ biểu đồ mà Crystal cung cấp có thể phục vụ cho đủ mọi lĩnh vực thương mại, khoa học, nghiên cứu, phân tích với hơn 40 mẫu dựng sẵn để bạn chọn lựa. Những biểu đồ này có thể được tùy biến, chỉnh sửa theo ý bạn để có thể đạt được kết quả như mong muốn.

Biết cách sử dụng biểu đồ, báo biểu của bạn có thể truyền đạt thông tin cô đọng và ấn tượng về dữ liệu như trong hình 7.2.



Hình 7.1: Các dạng biểu đồ Crystal Reports hỗ trợ.

#### Sales by Region



Hình 7.2: Các biểu đồ gắn vào báo biểu.

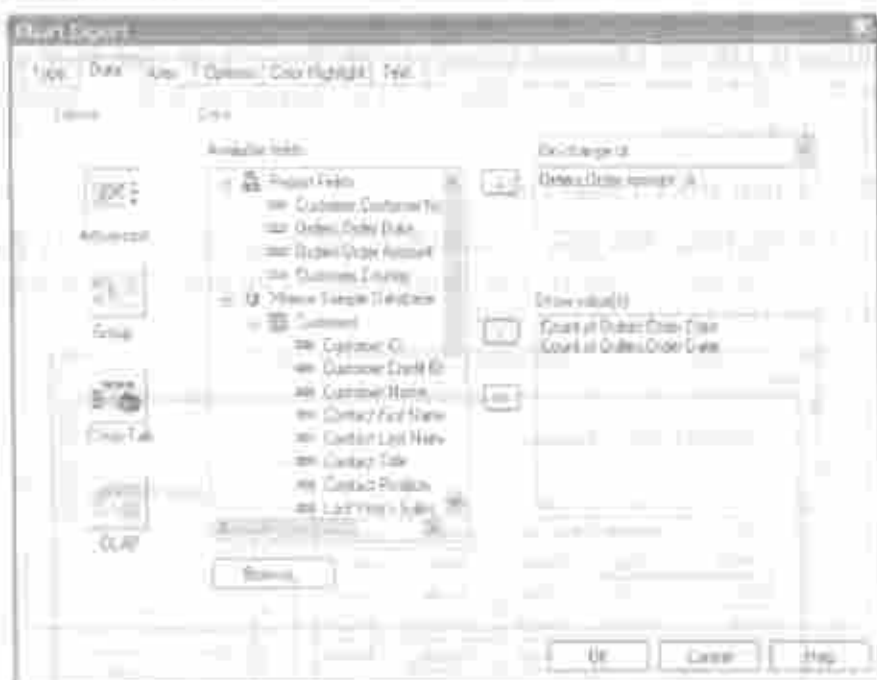
Trong chương này, chúng ta sẽ xem xét chi tiết từng biểu đồ, bắt đầu từ các biểu đồ thư viện mà Crystal cung cấp sẵn.

Các báo biểu tạo trong chương này chứa trong thư mục Project trên đĩa CD ROM. Bạn có thể thực hành các bước theo chỉ dẫn tạo báo biểu

trong giáo trình hoặc có thể chép báo biểu từ CD ROM vào trong ổ đĩa cứng của bạn rồi sửa lại.

## 2. THÊM BIỂU ĐỒ VÀO BÁO BIỂU

Crystal Reports hỗ trợ nhiều dạng biểu đồ khác nhau như biểu đồ cột, biểu đồ tròn, biểu đồ đường thẳng. Các loại biểu đồ mà bạn gặp thường rơi vào một trong bốn kiểu sau: Advanced, Group, Cross-Tab và OLAP; như hình 7.3.



Hình 7.3: Các dạng biểu đồ Crystal Report.

### 2.1. Biểu đồ Advanced (Biểu đồ cấp cao)

Biểu đồ Advanced làm việc giống như các biểu đồ mà bạn đã từng học ở trường: bạn vẽ một biểu đồ hoặc đồ thị dựa vào giá trị  $x$  và  $y$ .

### 2.2. Biểu đồ Group (Biểu đồ nhóm)

Biểu đồ nhóm thường được sử dụng trong các báo biểu, có thể được sử dụng bất cứ nơi nào mà bạn có thể chèn một nhóm vào trong báo biểu và tạo một Field tổng gộp dựa vào nhóm đó (Sum, Average, v.v.). Một

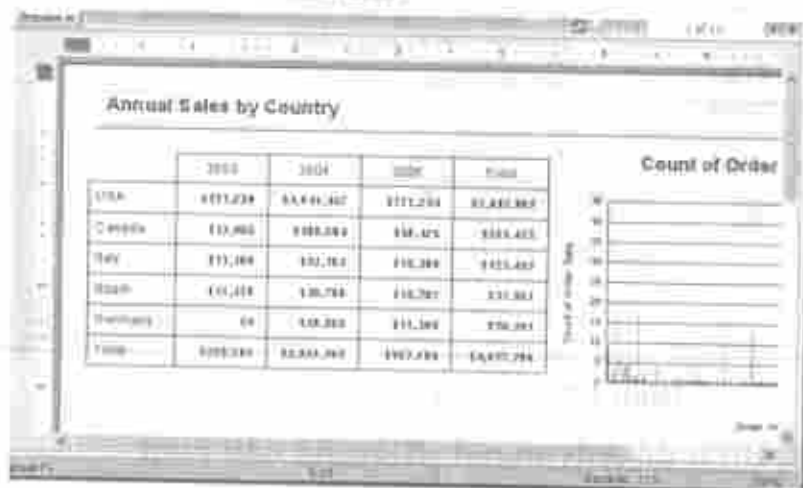
biểu đồ nhóm có thể xuất hiện một lần, đại diện dữ liệu trong toàn bộ báo biểu hoặc bạn có thể trình bày một biểu đồ cho mỗi nhóm.

Các biểu đồ nhóm có thể được sử dụng cho khả năng Drill-Through (đuyệt sâu thông tin), bạn có thể bắt đầu với một đồ thị ở mức cao nhất của dữ liệu (toàn bộ báo biểu) và xem dữ liệu phân cấp qua các nhóm khác. Đồ thị sẽ được trình bày theo mỗi nhóm, lượng thông tin sẽ dần hiện ra chi tiết khi bạn lần sâu xuống theo các trục của biểu đồ. Bạn có thể tham khảo ví dụ của báo biểu Drill-Through trong thư mục Project trên đĩa CD-ROM.

Nên nhớ rằng sự khác nhau giữa những biểu đồ Advanced và biểu đồ nhóm là biểu đồ Advanced không yêu cầu Group hoặc một Field tổng gộp có trong báo biểu. Bình thường, biểu đồ Advanced được sử dụng với các Field công thức.

### 2.3. Biểu đồ Cross-Tab

Dữ liệu trong báo biểu Cross-Tab có thể dùng để tạo biểu đồ theo hai trục tương ứng với giá trị của dòng và cột. Ví dụ hình 7.4 là một báo biểu Cross-Tab và biểu đồ bên cạnh.



Hình 7.4: Lưới Cross-tab và biểu đồ.

### 2.4. Biểu đồ khung lưới OLAP

Biểu đồ khung lưới OLAP tương tự như Cross-Tab, chỉ khác là OLAP trình bày dữ liệu đa hướng, trái ngược với Cross-Tab chỉ thể hiện dữ liệu quan hệ. Ngoài những sự khác biệt đó, chúng xử lý hoàn toàn giống nhau và chia sẻ tất cả các thuộc tính định dạng với nhau.

## 2.5. Thêm biểu đồ Advanced vào báo biểu

Biểu đồ Advanced được vẽ dựa vào giá trị x và y. Trong ví dụ sau chúng ta sẽ tạo một biểu đồ Advanced sử dụng báo biểu Order Listing từ CD ROM.

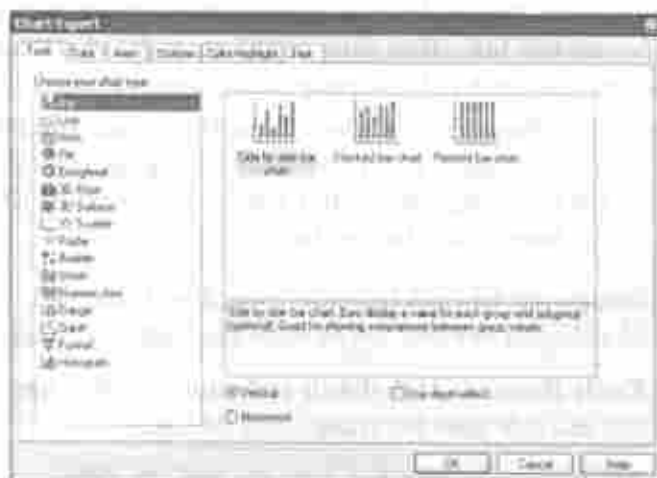
1. Mở Crystal Reports và mở báo biểu Orderlisting.rpt từ CD-ROM, như hình 7.5.



Item	Description	Price
10000000	Lamborghini	50000
10000001	Ferrari	45000
10000002	Rolls Royce Phantom	60000
10000003	Mercedes-Benz S-Class	48000
10000004	Rolls Royce Phantom	60000
10000005	Rolls Royce Phantom	60000
10000006	Rolls Royce Phantom	60000
10000007	Rolls Royce Phantom	60000
10000008	Rolls Royce Phantom	60000
10000009	Rolls Royce Phantom	60000
10000010	Rolls Royce Phantom	60000
10000011	Rolls Royce Phantom	60000
10000012	Rolls Royce Phantom	60000
10000013	Rolls Royce Phantom	60000
10000014	Rolls Royce Phantom	60000
10000015	Rolls Royce Phantom	60000
10000016	Rolls Royce Phantom	60000
10000017	Rolls Royce Phantom	60000
10000018	Rolls Royce Phantom	60000
10000019	Rolls Royce Phantom	60000
10000020	Rolls Royce Phantom	60000

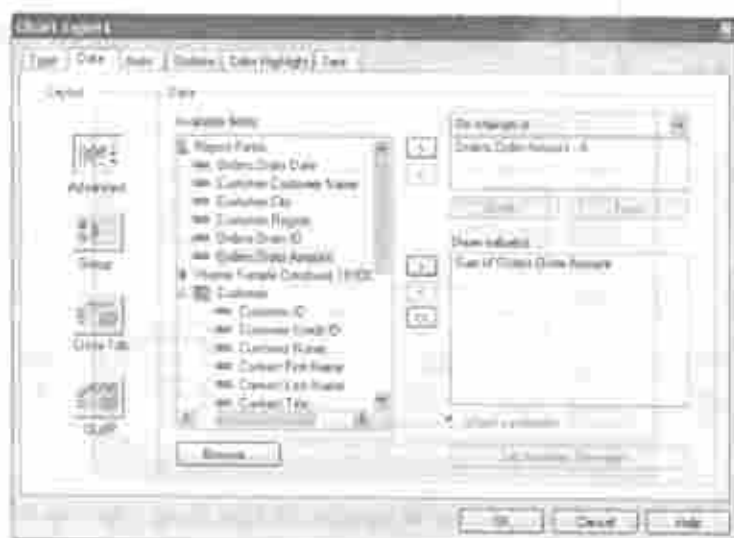
Hình 7.5: Báo biểu đặt hàng là nguồn dữ liệu cung cấp cho biểu đồ Advanced.

2. Chuyển báo biểu của bạn sang Tab Design, sau đó click Inset > Chart. Đặt biểu đồ vào mục Report Header và hộp thoại Chart Expert sẽ hiển thị như hình 7.6.



Hình 7.6: Hộp thoại chọn kiểu biểu đồ.

- Click vào Tab Type, chọn kiểu biểu đồ là Bar và sau đó click vào biểu tượng **Side by Side Chart** trong khung bên phải.
- Ở phía trên hộp thoại click vào tab Data. Hộp thoại sẽ có dạng như hình 7.7.



Hình 7.7: Thiết lập cho biểu đồ Advanced.

Trong ví dụ này, chúng ta sẽ tạo một biểu đồ Advanced hiển thị danh sách đơn đặt hàng được sắp xếp theo ngày đặt hàng. Biểu đồ sẽ hiển thị trong phần Report Header của báo biểu và sẽ xuất hiện trong trang đầu tiên của báo biểu. Bạn cũng có thể thay đổi vị trí xuất hiện của nó, chẳng hạn có thể đặt báo biểu trong phần Report Footer.

- Đầu tiên, click chọn **Field Orders OrderDate**, sau đó click vào dấu mũi tên (>) để di chuyển **Field OrderDate** sang khung **On change of**, đây là các giá trị tương đương như giá trị trực hoành. Khi làm việc với giá trị trực hoành này, bạn có ba tùy chọn:
  - On change of:** Khi giá trị chứa trong Field thay đổi, thành phần biểu đồ mới như bar hay pie sẽ được cập nhật và vẽ lại.
  - For Each Record:** Một thành phần biểu đồ bar hay pie mới sẽ được tạo cho mỗi mẫu tin trong cơ sở dữ liệu.
  - For All Records:** Một bar hay pie mới sẽ được tạo cho tất cả những mẫu tin trong cơ sở dữ liệu.



**Hình 7.8:** Báo biểu hoàn tất với biểu đồ *advanced* được thêm vào.

- Do biểu đồ trong ví dụ này là biểu đồ **Bar** để hiển thị Field Order Date, bạn sử dụng danh sách số xuống để chọn On change of.
- Tiếp theo, click chọn Field Orders.Order Amount, sau đó click vào mũi tên (<) phía dưới nhãn Show Values để di chuyển Field Orders.Order Amount vào danh sách, nó tương đương giá trị y trong biểu đồ của bạn.

**Ghi chú:** Lưu ý rằng, bạn có thể chọn nhiều giá trị x trong đó thị *Advanced*. Điều này cho phép bạn tạo các đồ thị phức tạp, đòi hỏi nhiều giá trị trên trục x, nhưng không phải đồ thị nào cũng hỗ trợ đa giá trị x.

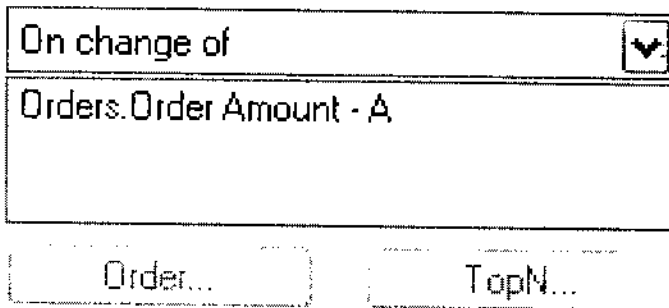
- Click vào tab Text để đặt tiêu đề cho biểu đồ của bạn. Như với các kiểu đồ thị khác, Crystal Reports tự tạo các tiêu đề cho biểu đồ, nhưng bạn có thể thay đổi và định dạng lại các tiêu đề cho biểu đồ theo ý mình.
- Để ghi đè lên các giá trị mặc định, bỏ chọn ở Auto Text và nhập vào tên tiêu đề cho biểu đồ.

10. Để thay đổi các định dạng thuộc tính liên quan đến văn bản, khi tiêu đề được hiển thị, bạn chọn các thành phần cần thay đổi trong danh sách bên góc dưới của hộp thoại. Click vào nút Font để thay đổi những thuộc tính này.
11. Click OK trong hộp thoại Chart Expert để chèn biểu đồ Advanced vào báo biểu của mình (hình 7.8).

### **2.6. Thay đổi cách sắp xếp**

Không may, biểu đồ mà chúng ta vừa tạo ra không thể hiển thị tất cả dữ liệu của báo biểu như ở dạng số và văn bản, lý do là có quá nhiều cột Bar hoặc dữ liệu nguồn. Tổ chức lại dữ liệu biểu đồ là điều cần thiết và chúng ta có thể thay đổi trật tự sắp xếp của biểu đồ để làm cho nó trở nên dễ đọc hơn.

Trên Tab Data của hộp thoại Chart Expert, bạn chú ý đến nút sắp xếp (Order) như trong hình 7.9. Nút này bị mờ đi cho đến khi bạn chọn On change of hoặc chọn xong Field làm giá trị cho trục x.

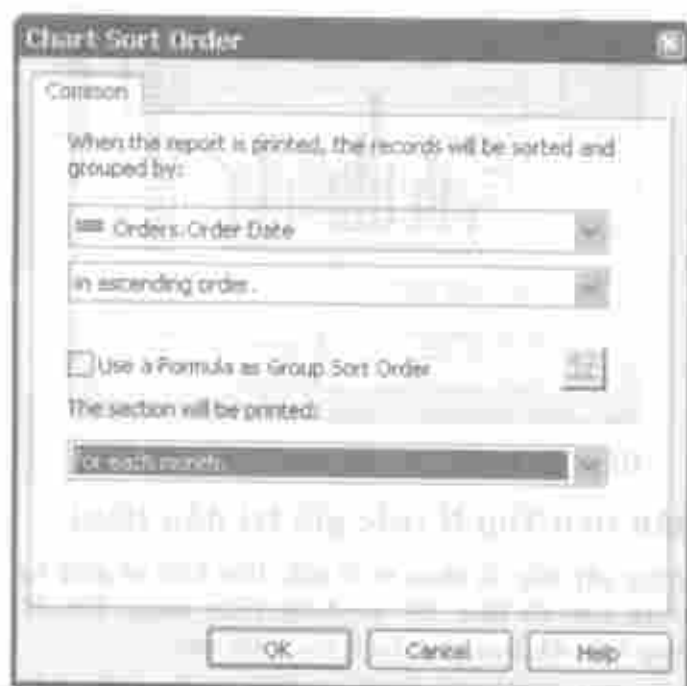


**Hình 7.9:** Chọn dữ liệu cho biểu đồ.

Bạn có thể sử dụng nút này để sắp xếp và tổng kết dữ liệu xuất hiện trong biểu đồ, hãy thực hiện các bước như sau:

1. Từ báo biểu Order Listing chúng ta đã thực hiện ở trên, click phải vào báo biểu Advanced mà chúng ta vừa tạo và chọn Chart Expert từ menu tắt để mở hộp thoại Chart Expert.
2. Click vào tab Data.
3. Xác định Field Orders.Order.Date trong hộp bên dưới danh sách On change of và click chọn nó.
4. Click vào nút Order để mở hộp thoại Sort Order như hình 7.10.





Hình 7.10: Tùy chọn sắp xếp biểu đồ.

Có bốn tùy chọn sắp xếp:

- **Ascending:** Sắp xếp tăng từ A to Z, 0 to 9.
- **Descending:** Sắp xếp giảm Z to A, 9 to 0.
- **Specified:** Sắp xếp theo thứ tự riêng.
- **Original:** Giữ nguyên thứ tự mặc định của dữ liệu.

5. Mặc định của tùy chọn sắp xếp là tăng dần.

Do ta vừa tạo biểu đồ Advanced trên Field ngày tháng (Order Date), một ô danh sách thứ ba xuất hiện với các tùy chọn nhóm theo khoảng thời gian như each day, week, month, v.v.

6. Để biểu đồ dễ đọc hơn, bạn chọn khoảng thời gian là For each Month và click OK.

7. Click OK lần nữa từ hộp thoại Chart Expert để trở về Report Designer. Bây giờ báo biểu của bạn sẽ trông giống như hình 7.11.



Hình 7.11: Báo biểu với nhóm là từng tháng.

## 2.7. Phân tích Top N (các giá trị đầu tiên)

Khả năng sắp xếp và chọn ra N mẫu tiêu biểu sẽ giúp bạn xác định được xu hướng của dữ liệu. Để áp dụng chức năng Top N / Sort all Analysis trong biểu đồ, bạn thực hiện các bước sau:

1. Click phải vào biểu đồ hiện hành của bạn và chọn Chart Expert.
2. Trong hộp thoại Chart Expert, click tab Data.
3. Tìm Field đúng sắp xếp để chọn ra N mẫu tin trong danh sách On Change Of, click chọn nó.
4. Khi Field đã được chọn, nút Top N sẽ có hiệu lực. Click vào nút này để mở hộp thoại Top N / Sort.
  - **Top N:** Nhập vào giá trị N để xác định số giá trị đầu tiên cần lấy dựa trên Field giá trị y mà bạn chọn.
  - **Bottom N:** Nhập vào giá trị N để xác định số giá trị cuối cùng cần lấy dựa trên Field giá trị y mà bạn chọn.
  - **Sort All:** Chọn tùy chọn này để sắp xếp tất cả dữ liệu hoặc là tăng hoặc là giảm dựa vào giá trị y.

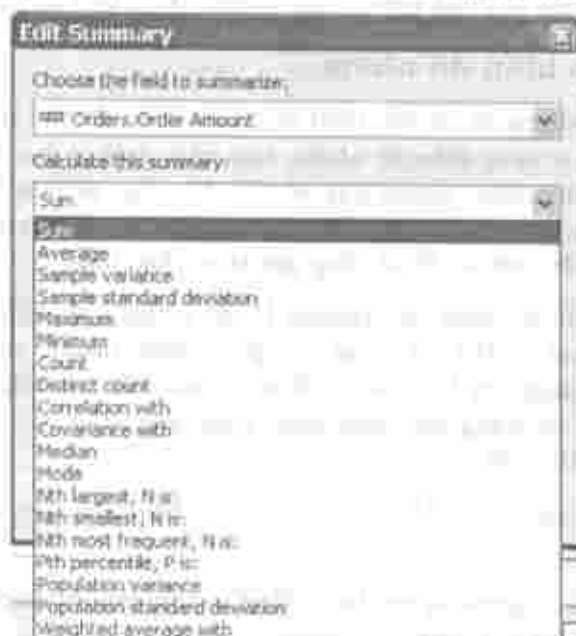
Với Top N và Bottom N, bạn cũng có thể xác định cách các giá trị N thể hiện. Khi phân tích báo biểu với Top N / Bottom N, bạn đã loại bỏ những giá trị không thỏa mãn bằng cách giữ chúng trong một nhóm khác.

5. Khi đã hoàn thành việc thiết lập tùy chọn Top N / Sorting, click OK để chấp nhận thay đổi của bạn và trở về môi trường thiết kế. Đồ thị của bạn bây giờ sẽ được cập nhật theo những tùy chọn mà bạn đã chỉ định.

## 2.8. Thay đổi toán tử tổng gộp

Một đặc tính khác của báo biểu mà chúng ta có thể sử dụng trong biểu đồ là toán tử tổng gộp đã học qua trong chương 4. Mặc định, một biểu đồ kiểu Advanced sẽ sử dụng một toán tử tổng gộp Sum, nhưng bạn có thể thay đổi toán tử này bằng những toán tử tổng gộp mà Crystal Reports hỗ trợ. Để thay đổi toán tử tổng gộp, bạn thực hiện các bước sau:

1. Click phải vào biểu đồ hiện hành của bạn và chọn Chart Expert.
2. Click vào Field tổng gộp mà bạn muốn thay đổi phép tính.
3. Kích vào nút Set Summary Operation, hộp thoại như hình 7.12 sẽ hiển thị. Bạn có thể chọn các toán tử tính tổng gộp từ danh sách.



Hình 7.12: Toán tử Summary có thể sử dụng trong biểu đồ.

- **Sum:** Tính tổng, sử dụng cho kiểu dữ liệu Numeric, Currency.
- **Average:** Tính trung bình cộng, sử dụng cho kiểu dữ liệu Numeric, Currency.

- **Minimum:** Xác định giá trị nhỏ nhất của Field trong cơ sở dữ liệu, sử dụng cho các kiểu dữ liệu Numeric, Currency, String và Date.
  - **Maximum:** Xác định giá trị lớn nhất của Field trong cơ sở dữ liệu, sử dụng cho các kiểu dữ liệu Numeric, Currency, String, Date
  - **Count:** Đếm những giá trị của Field trong cơ sở dữ liệu, sử dụng cho tất cả các kiểu dữ liệu.
  - **Distinct Count:** Tương tự như Count, trừ đi các giá trị giống nhau (chỉ đếm một lần).
4. Click OK để chấp nhận thay đổi kiểu tổng gộp của bạn và trở về hộp thoại Chart Expert.
  5. Khi đã hoàn thành việc chỉnh sửa biểu đồ của, click OK để chấp nhận thay đổi của bạn trở về môi trường thiết kế báo biểu.

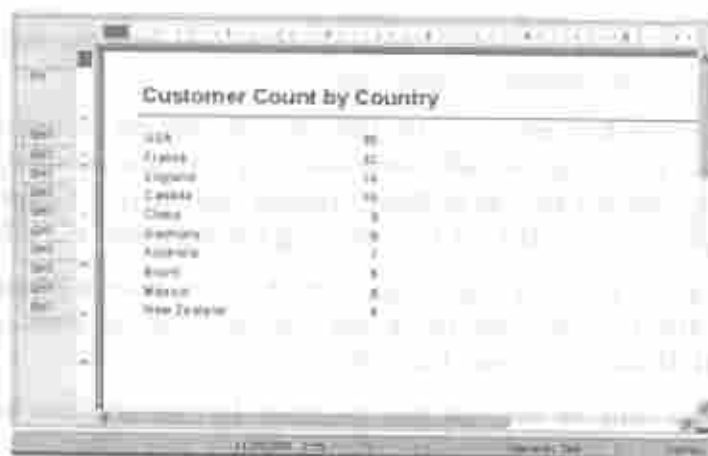
## 2.9. Chèn biểu đồ nhóm

Bây giờ chúng ta sẽ tìm hiểu một kiểu biểu đồ phổ biến khác: biểu đồ nhóm. Để sử dụng biểu đồ nhóm, bạn cần phải có hai điều kiện sau: Đầu tiên là một nhóm, nhóm này có thể được tạo từ một Field trong cơ sở dữ liệu, một công thức, v.v... Yêu cầu thứ hai cho một biểu đồ nhóm là bạn phải sắp xếp một số Field tổng gộp trong báo biểu.

Mặc dù những field này thường là các nhóm tính tổng trên field số, nhưng bạn cũng có thể sử dụng các phép đếm Count, tính giá trị lớn nhất hay nhỏ nhất với Max, Min... Một lần nữa, để cho công việc dễ dàng hơn, chúng ta sử dụng báo biểu mẫu Customercount.rpt trong thư mục Project trên CD ROM.

Để chèn một biểu đồ nhóm vào trong báo biểu này, bạn thực hiện các bước sau:

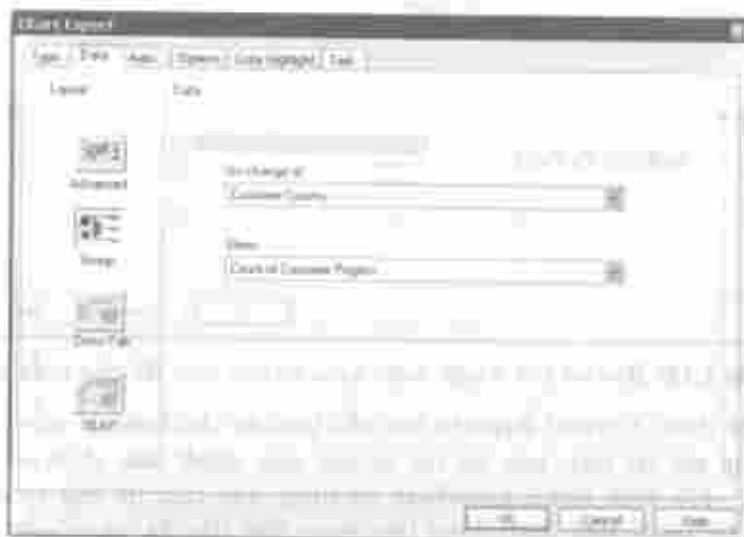
1. Mở Crystal Reports và mở báo biểu **Customercount.rpt** từ CD ROM, như hình 7.13.
2. Chuyển báo biểu của bạn sang Tab Design, click Insert | Chart và click vào mục Report Header trong biểu đồ của bạn. Biểu đồ sẽ tự động xuất hiện với các thiết đặt mặc định của nó. Để thay đổi những thiết đặt này, click phải vào biểu đồ và chọn Chart Expert.



**Hình 7.13:** Báo biểu sử dụng với biểu đồ nhóm.

Bước đầu tiên trong việc tùy biến một biểu đồ nhóm là lựa chọn một kiểu biểu đồ mà bạn muốn tạo.

3. Click vào tab Type và chọn một kiểu biểu đồ từ danh sách bên trái, sau đó click vào hình để xác định kiểu biểu đồ mà bạn muốn tạo.
4. Bước tiếp theo là chọn nguồn dữ liệu sẽ đưa vào. Click vào tab Data để mở hộp thoại như hình 7.14. Click vào biểu tượng Group bên phải của hộp thoại.



**Hình 7.14:** Bạn có thể chọn nguồn dữ liệu cho biểu đồ nhóm.

*Ghi chú: Nếu không chèn một nhóm vào trong báo biểu của bạn, thì tùy chọn này không thể thực hiện.*

- Bạn cần xác định nơi muốn tạo biểu đồ mới. Nếu chỉ có một nhóm và một tổng gộp trong báo biểu của mình, công việc này dễ dàng, bạn chỉ việc chọn nơi đặt biểu đồ là Report Header hoặc Report Footer. Nếu có nhiều nhóm, bạn có thể đặt biểu đồ trong Group Header của mỗi nhóm.
- Trong phần giữa của hộp thoại, bạn chọn On-change of and Show Value. Hai tùy chọn này tương ứng với các nhóm và Field tổng gộp mà bạn đã chèn vào trong báo biểu.
- Bước cuối cùng để tạo một biểu đồ nhóm là thiết lập tiêu đề (tức là Title, Label...) sẽ xuất hiện trong biểu đồ của bạn. Click tab Text để hiển thị hộp thoại như hình 7.15.



**Hình 7.15:** Bạn có thể nhập hoặc hiệu chỉnh tiêu đề cho biểu đồ.

Mặc định, Crystal Reports tạo sẵn tiêu đề cho biểu đồ của bạn. Để sửa những giá trị này, bạn bỏ tất cả các mục đánh dấu chọn của các ô checkbox phía dưới nhãn Auto-Text và nhập vào tiêu đề của riêng bạn. Để thay đổi thuộc tính định dạng liên quan đến tiêu đề, chọn một mục từ danh sách phía cuối góc phải hộp thoại và click vào nút Font để thay đổi thuộc tính.

8. Bây giờ click OK, các tùy chọn biểu đồ mới của bạn được chấp nhận và thêm vào trong báo biểu.

## 2.10. Chèn thêm biểu đồ Cross-Tab

Bạn có thể chèn một biểu đồ dựa trên dữ liệu kết xuất của bảng báo biểu Cross-Tab. Trong ví dụ sau, chúng ta sẽ xây dựng biểu đồ cho báo biểu Crosstab.rpt nằm trong thư mục Project của đĩa CD ROM. Để thêm một biểu đồ Cross-Tab vào báo biểu, bạn thực hiện các bước sau:

- Mở Crystal Reports và mở báo biểu Crosstab.rpt từ CD ROM.
  - Xem trước nội dung báo biểu để chắc chắn rằng dữ liệu kết xuất đầy đủ ở trên các dòng, cột và có cả cột tổng cộng.
  - Chuyển sang tab Design của báo biểu, click Insert > Chart, nhấp vào nơi chèn kiểu biểu đồ của hộp thoại Chart Expert.

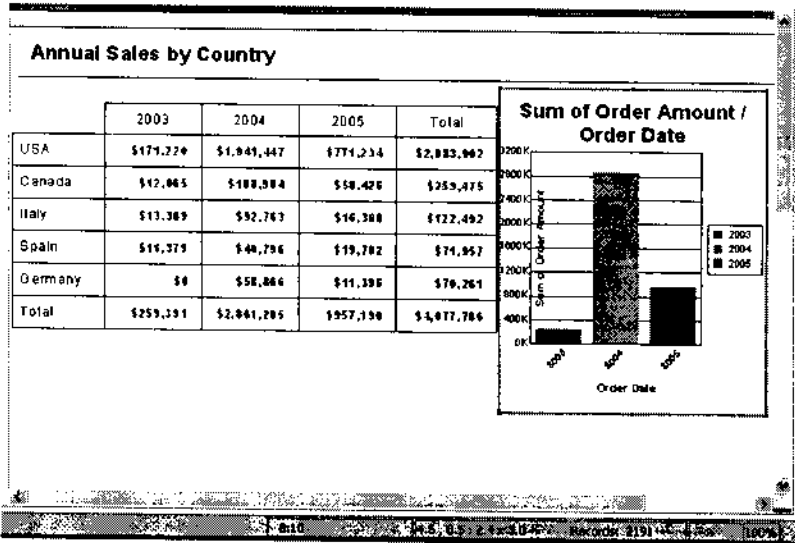
**Ghi chú:** Crystal Reports tự động thiết lập các thuộc tính biểu đồ mặc định để giảm bớt một số yêu cầu thiết lập biểu đồ. Nếu không muốn sử dụng các giá trị thiết lập mặc định này, bạn có thể bỏ nút check Automatically Set Chart Options trong hộp thoại Chart Expert. Những tab bổ sung này sẽ xuất hiện trong hộp thoại Chart Expert.



Hình 7.18: Tùy chọn biểu đồ Cross-Tab.

2. Click vào tab Data để tiến hành các bước tiếp theo của hộp thoại Chart Expert. Trên hộp thoại Chart Expert, Click vào kiểu Cross-Tab và chọn cách sắp xếp biểu đồ của bạn. Nếu báo biểu của bạn chỉ có một Cross-Tab, bạn có thể đặt biểu đồ trong Report Header hoặc Report Footer.
3. Click vào Tab Text để tiến đến bước cuối cùng trong hộp thoại Chart Export, và chọn tiêu đề cho biểu đồ cùng các định dạng cho đối tượng.
4. Khi đã hoàn thành việc sửa tiêu đề của biểu đồ, click OK quay về báo biểu.

Báo biểu của bạn bây giờ sẽ có hình dạng như hình 7.17.



**Hình 7.17:** Hoàn tất báo biểu với Cross-Tab và biểu đồ bên cạnh.

### 2.11. Di chuyển biểu đồ

Biểu đồ có thể được đặt hầu như mọi nơi trong báo biểu của bạn. Để thay đổi vị trí một biểu đồ bên trong một phân đoạn section, click vào biểu đồ, sau đó kéo và thả biểu đồ đến một vị trí mới.

Để di chuyển một biểu đồ giữa những section trong báo biểu (ví dụ, giữa Page Header và Group Header), bạn sử dụng hộp thoại Chart Expert theo các bước sau:

1. Click phải vào biểu đồ, chọn Chart Expert, sau đó click vào tab Data.
2. Sử dụng ô danh sách phía trên hộp thoại, chọn section mới cho biểu đồ.



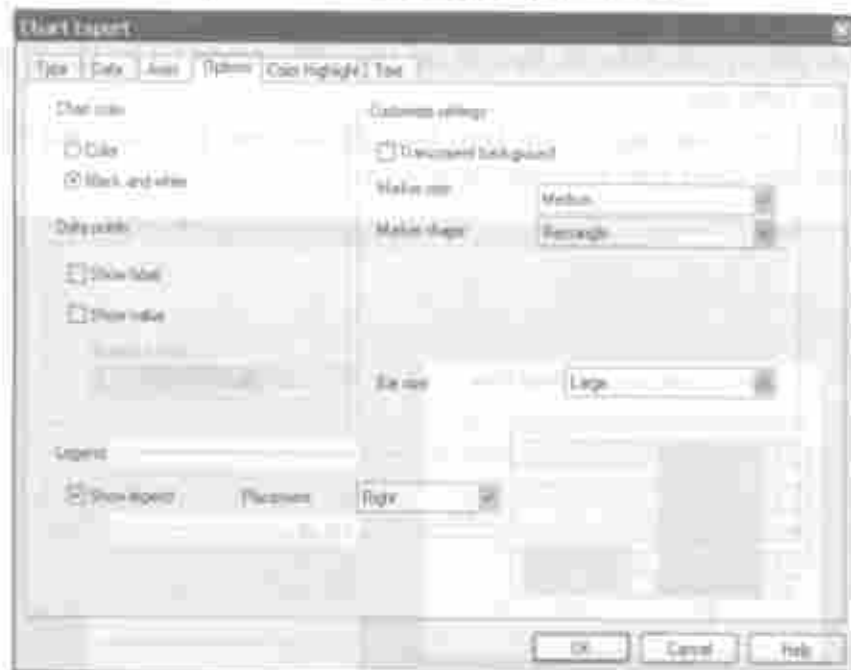
- Click OK để chấp nhận sự thay đổi. Biểu đồ của bạn bây giờ sẽ xuất hiện trong section mới mà bạn đã chọn.

## 2.12. Thay đổi biểu đồ màu thành biểu đồ trắng và đen

Mặc định, biểu đồ trong Crystal Reports được vẽ với các màu sắc khác nhau. Tuy nhiên, bạn có thể chọn chỉ sử dụng màu nền trắng và đen để kết quả ra máy in rõ ràng hơn.

Để chuyển một biểu đồ sang màu trắng đen, bạn thực hiện các bước sau:

- Click phải trên biểu đồ mà bạn muốn đổi màu và click vào Format / Chart Expert từ menu tắt.
- Click vào tab Option để mở hộp thoại như hình 7.18.



**Hình 7.18:** Một số tùy chọn biểu đồ có sẵn để bạn có thể thao tác trong Expert.

- Click vào nút radio Black and White, sau đó click OK để chấp nhận thay đổi của bạn. Biểu đồ của bạn bây giờ sẽ thay đổi chỉ có màu đen và xám.

### 2.13. Hiện thị chú giải

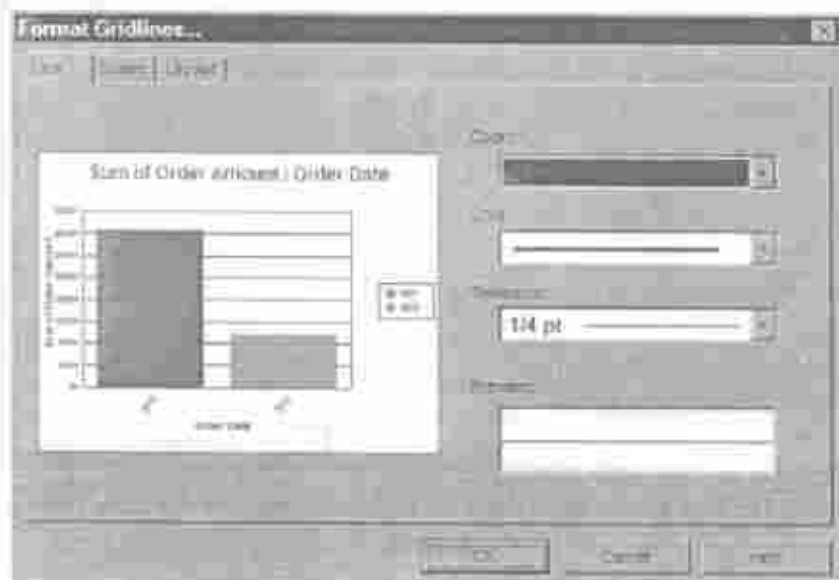
Crystal Reports tạo ra chú giải cho biểu đồ. Để thao tác với các chú giải trong Chart Expert, bạn thực hiện theo các bước sau:

1. Click phải trên biểu đồ mà bạn muốn làm việc và click vào **Format | Chart Expert** từ menu tắt.
2. Click vào **Tab Type** trên hộp thoại Chart Expert, bỏ chọn mục **Automatically Set Chart Options** và click vào **Tab Option**.
3. Định vị mục **Legend** và chọn checkbox **Show Legend** để đặt lên chú giải.
4. Sử dụng danh sách số xuống kế bên để chọn nơi đặt chú giải (**Right**, **left** hoặc **bottom**).
5. Click **OK** để chấp nhận thay đổi của bạn.

### 2.14. Điều khiển ô lưới và tỷ lệ đo

Crystal kẻ các khung lưới giúp bạn có thể dễ dàng xác định tọa độ của các giá trị trên biểu đồ. Bạn có thể thay đổi các giá trị về khung lưới theo các bước sau:

1. Click phải vào một ô lưới trên biểu đồ và chọn **Format | Grid Line**. Một hộp thoại xuất hiện như hình 7.19.



Hình 7.19: Bạn có thể thiết lập định dạng lưới bằng cách sử dụng các tùy chọn được hỗ trợ.

2. Click chọn Tab Scale hoặc Layout.
3. Trên Tab Layout, chọn Show Gridlines để hiển thị khung lưới.
4. Click vào tab Scale để đặt tùy chọn cho tỷ lệ chia trên trục.
5. Khi đã hoàn thành việc thiết lập tùy chọn lưới và tỷ lệ, click OK để chấp nhận thay đổi của bạn.

### **2.15. Thay đổi kiểu biểu đồ**

Bạn có thể áp dụng một số kiểu biểu đồ khác nhau cho báo biểu của mình. Ngoài những kiểu biểu đồ chuẩn như Bar, Pie bạn còn có nhiều kiểu biểu đồ chuyên dùng cho thống kê như biểu đồ rada và biểu đồ nổi bọt (bubble graphic), biểu đồ mảng ba chiều (three-dimensional) và nhiều loại biểu đồ khác. Để thay đổi kiểu biểu đồ, bạn thực hiện theo các bước sau:

1. Click phải vào biểu đồ bạn muốn thay đổi, chọn Load Template từ menu tắt.
2. Từ Tab Gallery, chọn một kiểu biểu đồ mới bằng cách click vào biểu tượng các biểu đồ trong danh sách có sẵn.
3. Click OK để chấp nhận thay đổi của bạn và trở về báo biểu.

Ngoài những tùy chọn cơ bản, Crystal Reports còn có một số định dạng nâng cao dành cho biểu đồ và dữ liệu. Các thuộc tính định dạng này bạn có thể sử dụng bằng cách click phải vào một biểu đồ và chọn Chart Options.

### **2.16. Chỉnh kích thước Font và màu cho biểu đồ**

Một trong các tác vụ định dạng thông dụng nhất bạn sẽ dùng đến là thay đổi Font, kích thước và màu sắc cho các thành phần biểu đồ. Để thực hiện những thay đổi này, bạn làm theo các bước sau:

1. Tìm thành phần biểu đồ mà bạn muốn thay đổi, click phải vào biểu đồ và chọn Format XXX để mở hộp thoại Formatting.
2. Sử dụng hộp thoại như hình 7.20, chọn Font, kích thước, kiểu và màu cho đối tượng.
3. Click OK để trở về báo biểu.

### 3. TÓM TẮT

Chương này chúng ta đã học về kỹ thuật xây dựng biểu đồ trình diễn số liệu báo biểu dưới dạng hình ảnh. Bất chấp dù bạn có 10 hoặc 10.000 mẫu tin, biểu đồ vẫn là cách đơn giản nhất để tổng hợp và hiển thị thông tin trong báo biểu khiến người dùng dễ nhận biết và phân tích nhất. Trong chương sau, chúng ta sẽ bắt đầu tìm hiểu về những kỹ thuật tích hợp báo biểu với các ứng dụng Windows và Web.

## Chương 8:

# TÍCH HỢP BÁO BIỂU

- ❖ *Tổng quan.*
- ❖ *Chọn phương pháp tích hợp.*
- ❖ *Chọn nguồn dữ liệu.*
- ❖ *Chọn cấu trúc báo biểu.*
- ❖ *Tóm tắt.*

## 1. TỔNG QUAN

Bạn đã biết cách thiết kế các loại báo biểu, phần tiếp theo của các chương trong giáo trình này sẽ hướng dẫn bạn cách tích hợp báo biểu vào những loại ứng dụng khác nhau. Báo biểu của bạn có thể tích hợp với ứng dụng Windows, Web và Java. Chương này chủ yếu giới thiệu với bạn các phương pháp tích hợp, lựa chọn nguồn dữ liệu cho báo biểu, những kiến trúc tích hợp nào được xem là tốt nhất.

## 2. CHỌN PHƯƠNG PHÁP TÍCH HỢP

Thường thì các nhà phát triển ứng dụng tích hợp các báo biểu vào trong chương trình dựa vào nguồn dữ liệu là cơ sở dữ liệu mà chương trình sử dụng, chương trình cung cấp cho người dùng các mục chọn menu để chọn loại báo biểu cần xem. Hình 8.1 là mô hình chung nhất khi phát triển ứng dụng có sử dụng báo biểu. Thông thường người dùng sẽ chọn tên báo biểu cần xem từ menu, trình Viewer sẽ hiển thị nội dung báo biểu (View Only). Với một số báo biểu, người dùng có thể nhập vào tham số thông qua hộp thoại (View with Parameters). Với những ứng dụng cao cấp hơn, việc tích hợp còn cho phép người dùng tương tác, lọc, drill-down thông tin trên báo biểu.

### 2.1. Report Viewer

Viewer là một công cụ để hiển thị nội dung báo biểu. Ví dụ, để xem báo biểu từ một ứng dụng Windows được tạo với công cụ phát triển Visual Studio 2005, trước tiên bạn kéo điều khiển ReportViewer vào cửa

số Form và nhập dòng mã sau để gán báo biểu cho Viewer hiển thị nội dung báo biểu sales.rpt:

```
CrystalReportViewer1.ReportSource="c:\sales.rpt"
```



**Hình 8.1:** Các tùy chọn tích hợp báo biểu phổ biến.

Người sử dụng có thể thực thi và xem trước (Preview) nội dung báo biểu trên Viewer như hình 8.2.



**Hình 8.2:** Report Viewer chuẩn cho các ứng dụng Windows.

Mỗi môi trường phát triển ứng dụng khác nhau sẽ có thành phần Viewer với một tập lệnh phong phú theo mô hình đối tượng, cho phép bạn điều khiển diện mạo và cách hiển thị nội dung báo biểu. Một Viewer thường bao gồm thanh công cụ với các nút lệnh cho phép nhảy trang, in, lọc, lưu file.

## 2.2. Sử dụng Report Viewer với tham số

Kiểu tích hợp kế tiếp được sử dụng khá rộng rãi là truyền tham số cho báo biểu. Trong trường hợp này, người phát triển ứng dụng sẽ tạo ra các báo biểu từ cơ sở dữ liệu với tham số được dùng để lọc nội dung báo biểu theo yêu cầu của người dùng cuối.

Hình 8.3 là một ví dụ của ứng dụng Windows hiển thị báo biểu trong Viewer với tham số cho phép người dùng nhập từ bên ngoài vào để làm điều kiện lọc cho báo biểu.



Hình 8.3: Báo biểu với tham số trong ứng dụng Windows.

## 2.3. Tích hợp nâng cao

Khi người phát triển ứng dụng mong muốn điều khiển báo biểu ở mức sâu hơn, họ có thể bỏ qua Viewer hoặc chỉ dùng một phần chức năng của Viewer. Họ thường sử dụng các hàm lập trình API mà Crystal cung cấp để tạo báo biểu động, thực hiện những công việc đặc thù như nhúng Viewer vào một phần ứng dụng nào đó.

Trong cả ba phương pháp tích hợp trên thì phương pháp xem báo biểu với tham số chiếm đến 70%, phương pháp chỉ đơn thuần xem nội dung báo biểu mà không cần tham số chiếm 20% và phần còn lại là phương pháp tích hợp nâng cao không dùng đến Viewer (chiếm 10%).

Tại sao xem một báo biểu với tham số lại sử dụng nhiều như vậy? Lý do là dùng tham số bạn có thể linh động tạo ra nhiều báo biểu khác nhau. Nội dung báo biểu có thể thay đổi theo yêu cầu của người dùng cuối. Phương pháp không dùng Viewer ít được sử dụng vì không phải nhà





<i>Teradata v2R5, R6</i>		X		
<i>Microsoft SQL Server 2003 SP3</i>		X	X	X
<i>Microsoft SQL Server 7.0 SP4</i>		X		X
<i>MySQL 4.0</i>		X	X	
<i>Microsoft Access 2000</i>	X	X		X
<i>Microsoft Access 2003</i>	X	X		X
<i>Microsoft Excel 2000</i>	X	X		X
<i>Microsoft Excel 2004</i>	X	X		X
<i>Microsoft Exchange</i>	X			
<i>ADO.Net</i>	X			
<i>XML</i>		X		
<i>All Other ODBC Drivers</i>		X		
<i>All Other JDBC Drivers</i>			X	

### **3.2. Dữ liệu ứng dụng**

Nguồn dữ liệu phổ biến khác cho các báo biểu là dữ liệu bên trong các ứng dụng. Thường ứng dụng sẽ sử dụng những tập dữ liệu phức tạp có cấu trúc đặc thù riêng của mình. Chẳng hạn bạn có thể xây dựng một cấu trúc file để lưu địa chỉ khách hàng của riêng mình và nhờ Crystal đọc dữ liệu và in nội dung danh sách địa chỉ ra báo biểu. Không phải nguồn dữ liệu của ứng dụng nào cũng mở và có thể truy xuất được.

### **3.3. XML và những dịch vụ Web XML**

Một nguồn dữ liệu nữa có thể dùng cho báo biểu là file XML và các dịch vụ Web XML. Crystal Reports XI bao gồm một bộ điều khiển XML linh hoạt có thể sử dụng để tạo báo biểu từ file XML cục bộ và trên máy chủ Web Server hay thông qua các dịch vụ XML Web. Phương pháp này mở ra một khả năng để làm việc với dữ liệu cả bên trong và bên ngoài tổ chức cũng như tạo khả năng giao tiếp cho các chương trình chưa được Crystal hỗ trợ truy xuất nguồn dữ liệu bên trong ứng dụng.

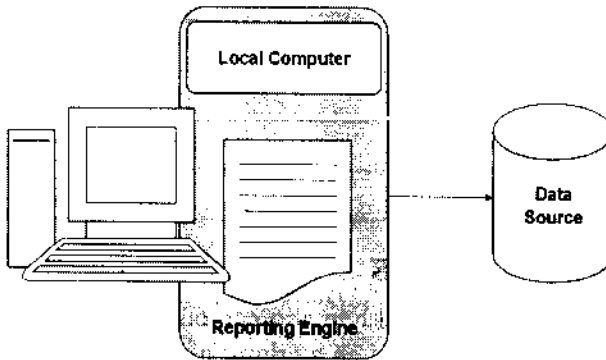
Nếu phải chọn giữa một trong số những phương pháp truy cập dữ liệu, tốt nhất nên chọn ODBC hoặc JDBC, OLE-DB, vì đây là phương pháp dễ sử dụng và phổ biến nhất.

## 4. CHỌN KIẾN TRÚC BÁO BIỂU

Mục đích chung nhất của hầu hết những nhà phát triển là tạo ra các ứng dụng có thể thay đổi dễ dàng (scalable) và truy cập bởi nhiều người sử dụng. Để làm được điều này bạn phải xây dựng kiến trúc của ứng dụng sao cho phù hợp. Dưới đây là những kiến trúc ứng dụng thường sử dụng các báo biểu của Crystal nhất.

### 4.1. Ứng dụng Thick-Client

Ứng dụng Thick-Client là những ứng dụng truyền thống chạy cục bộ trên máy PC của người sử dụng như hình 8.5.



Hình 8.5: Một ứng dụng Thick-Client.

Đa số người phát triển và người sử dụng đều quen thuộc với kiểu ứng dụng này do hầu hết các ứng dụng của Windows là Thick-Client. Mặc dù phần lớn những ứng dụng này chạy độc lập trên máy đơn nhưng ứng dụng Thick-Client đôi khi cũng được xem là loại ứng dụng khách-chủ (Client-Server) nếu chúng lưu cơ sở dữ liệu trên máy chủ thay vì trên máy cục bộ.

Ngay các báo biểu của Crystal Reports đa phần được thiết kế chạy cục bộ phục vụ cho ứng dụng Thick-Client, một số ứng dụng thương mại sử dụng kiến trúc này cũng đơn giản và rất dễ tích hợp.

Một trong những thuận lợi khi phân phối báo biểu theo kiến trúc này là mọi yêu cầu xem báo biểu đều được thiết lập cục bộ. Thuận lợi khác là người sử dụng có thể xem và tương tác với báo biểu trên máy cục bộ không yêu cầu mạng hay kết nối với nguồn dữ liệu ở máy chủ.

Sự bất lợi của kiểu kiến trúc này cũng là do các báo biểu thực thi cực bộ cho nên nếu có thay đổi thì người sử dụng sẽ phải chờ các nhà phát triển gửi đến phiên bản cập nhật của báo biểu. Ngoài ra, việc phân phối một ứng dụng Thick-Client yêu cầu phân phối kèm theo một số thành phần của Crystal Reports, thường thì cài đặt những phiên bản Crystal khác nhau có thể dẫn đến những lỗi không tương thích về phiên bản.

Các ứng dụng Thick-Client tuy có những giới hạn nhưng thế mạnh là nó cho phép tạo ra các báo biểu nhúng trực tiếp vào các ứng dụng. Hãy chọn một kiểu kiến trúc này nếu ứng dụng của bạn được thiết kế chủ yếu để chạy trên máy cực bộ đơn giản hóa các tác vụ đơn giản của người sử dụng.

## 4.2. Ứng dụng Single-Tier

Ứng dụng Single-Tier có thêm tầng xử lý phía máy chủ (thường là Web Server). Tất cả công việc xử lý được đặt tại trung tâm máy chủ (xem hình 8.6).



Hình 8.6: Ứng dụng Single-Tier.

Kiểu ứng dụng báo biểu này dần dần phát triển thành một tập hợp những công cụ mà Crystal Reports dùng hỗ trợ cho các ứng dụng Web. Nhưng một khuyết điểm trong mô hình Single-Tier truyền thống là tất cả công việc xử lý dồn trên máy chủ Web Server. Vì lý do này, các ứng dụng báo biểu có thể gây quá tải cho trình chủ Web Server do Web Server phải lo xử lý những tác vụ truy xuất tài nguyên khác. Để khắc phục, Crystal đưa ra mô hình ứng dụng báo biểu đa tầng.

## 4.3. Ứng dụng Multitiered (đa tầng)

Hình 8.7 là mô hình phát triển một ứng dụng đa tầng.



Hình 8.7: Kiến trúc đa tầng.

Trong một ứng dụng đa tầng, lời xử lý tính toán cho báo biểu tập trung trên một máy chủ chuyên dụng (Application Server). Crystal Reports Server và BusinessObjects Enterprise là bộ khung điều khiển mô hình ứng dụng này.

Các báo biểu tập trung trên một máy chủ chịu sự phân quyền, kiểm tra tính an toàn bảo mật, chia sẻ dữ liệu và khai thác sức mạnh tính toán của máy chủ. Ứng dụng Web hay những ứng dụng máy khách sẽ gửi yêu cầu (tham số, tên báo biểu...) đến máy chủ. Máy chủ sẽ xử lý yêu cầu và trả về kết quả.

Tóm lại các ứng dụng Windows truyền thống thường sử dụng mô hình Thick-Client. Với môi trường ứng dụng Web, đa số những nhà phát triển sẽ chọn mô hình Single-Tier. Mô hình này có thể làm tăng giá thành khi phải mua thêm máy chủ và phần mềm bổ sung. Khi ứng dụng của bạn mở rộng trên quy mô nhiều người dùng, mô hình đa tầng Multitiered sẽ phù hợp hơn cả vì nó bảo đảm tính bảo mật, chia sẻ thông tin và khai thác sức mạnh xử lý của máy chủ.

## 5. TÓM TẮT

Mục đích của chương này là đưa ra các giải pháp để bạn chọn lựa khi thiết kế ứng dụng cần tích hợp báo biểu của Crystal. Không có mô hình hay kiến trúc nào tốt nhất cả, tùy theo mục đích và quy mô của ứng dụng mà bạn chọn giải pháp thích hợp. Trong các chương tiếp theo, chúng ta sẽ từng bước cụ thể hóa quá trình tích hợp báo biểu vào các loại ứng dụng Windows, .NET, Web và Java. Bắt đầu bằng những ứng dụng truyền thống Windows, chương sau sẽ hướng dẫn bạn cách sử dụng và liên kết báo biểu với ứng dụng .NET.

## Chương 9:

# TÍCH HỢP BÁO BIỂU VÀO ỨNG DỤNG WINDOWS

- ◆ *Tổng quan.*
- ◆ *Làm việc với mô hình Report View Object.*
- ◆ *Làm việc với mô hình Report Document Object.*
- ◆ *Tóm tắt.*

## 1. TỔNG QUAN

Trong những năm qua các ứng dụng của Window hầu như thống trị trên thị trường phần mềm máy tính, chúng bao gồm các ứng cá nhân, ứng dụng doanh nghiệp từ trình xử lý văn bản, bảng tính, cơ sở dữ liệu, quản lý khách hàng, tài chính, kế toán, thống kê...

Hầu hết các ứng dụng doanh nghiệp đều cần quản lý một số lượng lớn dữ liệu. Một yêu cầu chung cho những kiểu ứng dụng này là khả năng trình bày thông tin sáng sủa, ngắn gọn - đây cũng là mục tiêu mà các chương trình cần đến Crystal Reports.

Một trong những công ty nhận ra nhu cầu sử dụng công cụ Crystal Reports là Microsoft. Crystal Reports được phát triển trong Microsoft từ Crystal Reports 2.0 và được gắn vào Visual Basic 3.0 từ năm 1993. Trên mỗi phiên bản, Crystal Reports được phát triển song hành cùng với môi trường lập trình ứng dụng của Microsoft, phiên bản mới nhất mà Microsoft tích hợp đầy đủ các chức năng của Crystal Reports là Visual Studio 2005.

Chương này sẽ hướng dẫn bạn cách tích hợp Crystal Reports vào trong những ứng dụng của Window được tạo ra từ môi trường Visual Studio .NET và .NET Framework.

Giả thiết rằng trong chương này bạn đã cài đặt Visual Studio .NET 2003 (hoặc Visual Studio 2005) cùng Crystal Reports XI Developers Edition. Bạn có thể soạn thảo các báo biểu nào trực tiếp trong môi trường Visual Studio IDE. Nếu thích sử dụng môi trường Crystal riêng để

thiết kế báo biểu độc lập bạn click phải vào báo biểu trong dự án và chọn Open With để mở hộp thoại như hình 9.1. Chọn Crw32. Đây là chương trình Crystal Reports XI.



Hình 9.1: Tùy chọn Open With.

Tất cả các ứng dụng mẫu và báo biểu của chương này được chứa trong CD ROM.

Với Crystal Reports XI, bạn có thể sử dụng hai mô hình đối tượng chính để tích hợp báo biểu vào trong ứng dụng .NET. Mô hình thứ nhất là mô hình đối tượng *CrystalReportViewer*. Mô hình này cung cấp các thuộc tính, phương thức và sự kiện để xem báo biểu và thực hiện các thao tác yêu cầu đơn giản của khách hàng như duyệt trang, in ấn. Mô hình thứ hai là mô hình đối tượng *ReportDocument*. Mô hình này cung cấp một tập hợp các thuộc tính, phương thức và sự kiện mà bạn có thể sử dụng để làm việc trực tiếp với báo biểu.

Quyết định sử dụng mô hình đối tượng nào là tùy vào kiểu ứng dụng mà bạn muốn tạo. Trong chương 8, chúng ta đã xem qua những phương pháp tích hợp khác nhau. Với những ví dụ đơn giản, mô hình đối tượng *CrystalReportViewer* là chọn lựa tốt nhất do nó cung cấp các tùy chọn mà bạn có thể xem trước (Preview), in và lưu dữ liệu báo biểu thành file... Bạn cũng có thể chuyển tham số cho báo biểu thông qua mô hình đối tượng này cũng như đăng nhập tự động vào cơ sở dữ liệu.

Nếu cần tích hợp sâu hơn, chẳng hạn chương trình cần chuyển tập dữ liệu đặc thù cho báo biểu qua ADO hoặc ADO.NET, in kết quả ra một máy in riêng biệt nào đó, điều khiển Field công thức và định dạng báo biểu theo cách của bạn, khi đó bạn nên sử dụng mô hình đối tượng *ReportDocument*.

## 2. LÀM VIỆC VỚI MÔ HÌNH REPORT VIEW OBJECT

Để sử dụng mô hình đối tượng *CrystalReportViewer* bạn cần làm hai việc. Đầu tiên là từ cửa sổ công cụ điều khiển .NET, bạn kéo và thả đối tượng *CrystalReportViewer* vào trong cửa sổ Form của ứng dụng; công việc thứ hai là viết mã cho Viewer thông qua các thuộc tính, phương thức và sự kiện liên quan đến mô hình đối tượng để điều khiển báo biểu và Viewer.

### 2.1. Xem trước báo biểu

Bạn chỉ cần một dòng mã duy nhất để xem báo biểu từ ứng dụng. Để xem báo biểu trong ứng dụng, tất cả những gì bạn cần là kéo *CrystalReportViewer* lên trên form và đặt thuộc tính *ReportSource* trở vào file báo biểu mà bạn thiết kế trong Crystal hay trong dự án .NET. Dưới đây là các bước để hiển thị một báo biểu trong ứng dụng .NET:

1. Mở Visual Studio.NET 2003 (hoặc Visual Studio 2005).
2. Click File | New | Project.
3. Từ cửa sổ New Project, click vào mục Visual Studio Project, và sau đó chọn Windows Application.
4. Nhập tên dự án, click OK. Trong ví dụ này, chúng ta đặt tên dự án là VB\_ViewReport. Một dự án mới được mở ra cùng với cửa sổ form mẫu.
5. Click View | Toolbox và kéo *CrystalReportViewer* từ cửa sổ Toolbox vào form, đặt thuộc tính Dock của *CrystalReportViewer* là Fill.
6. Tiếp theo, Double click vào form để viết mã cho sự kiện Form Load như sau:

```
CrystalReportViewer1.ReportSource="C:\Roster.rpt"
```

Báo biểu Roster.rpt được chứa trong thư mục Project trên CD ROM. Bạn có thể sửa đổi vị trí của dự án phụ thuộc vào nơi bạn lưu trữ file này.

7. Click Debug | Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Cửa sổ Form sẽ hiển thị cùng với điều khiển *CrystalReportViewer* mà bạn đã thêm vào Form như hình 9.2.



Hình 9.2: Viewer hiển thị trong cửa sổ Form.

Theo mặc định, điều khiển Report Viewer được hiển thị với một thanh công cụ ở đỉnh bao gồm các nút sau:

- **Export:** Khi click vào nút **Export**, xuất hiện hộp thoại **Save**, bạn có thể kết xuất báo biểu sang định dạng **Word**, **Excel**, **PDF**, v.v..
- **Print:** Mở hộp thoại in.
- **Refresh:** Làm mới lại cơ sở dữ liệu.
- **Group Tree:** Hiển thị hoặc ẩn cây phân nhóm phía bên trái báo biểu.
- **Page Navigation Buttons:** Cung cấp các nút di chuyển, duyệt trang.
- **Find:** Mở hộp thoại **Find** nhập văn bản để tìm kiếm thông tin trong báo biểu.
- **Zoom:** Điều khiển phóng to, thu nhỏ nội dung báo biểu.

## 2.2. Sửa đổi diện mạo công cụ Viewer

Công cụ Viewer cung cấp một số thuộc tính cho phép bạn thay đổi diện mạo của nó.

Thuộc tính	Ý nghĩa	Kiểu
<code>DisplayToolBar</code>	Hiển thị thanh công cụ Viewer.	Boolean
<code>DisplayBackGroundEdge</code>	Hiển thị khung viền quanh Viewer.	Boolean



<code>DisplaygroupTree</code>	Hiển thị cấu trúc cây phân nhóm phía trái khung Viewer.	<code>Boolean</code>
-------------------------------	---	----------------------

Ví dụ nếu muốn ẩn cây phân nhóm hoặc thanh công cụ, bạn có thể thêm đoạn mã sau vào Form Load:

```
CrystalReportViewer1.DisplayGroupTree=False
CrystalReportViewer1.DisplayToolBar=False
```

Nếu sử dụng mã này để tắt cây phân nhóm và thanh công cụ, báo biểu của bạn bây giờ sẽ trông như hình 9.3.



**Hình 9.3:** Report Viewer loại bỏ cây phân nhóm và thanh công cụ.

Bạn cũng có thể điều khiển các thành phần riêng lẻ trên thanh công cụ. Ví dụ, bạn có thể không muốn hiển thị nút in, khi đó người sử dụng sẽ không thể in nội dung báo biểu từ công cụ Viewer. Những thuộc tính sau có thể dùng để điều khiển các nút trên thanh công cụ:

- ❖ `ShowCloseButton`.
- ❖ `ShowExportButton`.
- ❖ `ShowGotoPageButton`.
- ❖ `ShowGroupTreeButton`.
- ❖ `ShowPageNavigationButtons`.

- ❖ ShowPrintButton.
- ❖ ShowRefreshButton.
- ❖ ShowTextSearchButton.

Tất cả các thuộc tính là Boolean, để dấu nút in bạn có thể viết mã như sau:

```
CrystalReportViewer1.ShowPrintButton=False
```

Khi xem báo biểu nút in Print sẽ không còn xuất hiện trên công cụ Viewer nữa (hình 9.4).



Hình 9.4: Báo biểu công cụ xem thiếu nút Print.

### 2.3. In và kết xuất dữ liệu

Mặc dù chính Report Viewer cung cấp nút in Print để người sử dụng có thể click vào đó in báo biểu, nhưng bạn có thể chủ động in bằng cách gọi phương thức PrintReport của Viewer. Mục in lúc này có thể được tùy biến để trong menu chính.

Trong ví dụ này, chúng ta sẽ thêm Report Viewer vào trong form, sau đó tạo một mục menu tên là Print Report gắn trên menu File của chương trình. Khi người sử dụng click File|Print Report, một hộp thoại in chuẩn của Windows sẽ xuất hiện và cho phép người sử dụng chọn để in nội dung báo biểu. Để tạo ứng dụng này, thực hiện các bước sau:

1. Click Start | Visual Studio.NET 2003.
2. Click File | New | Project.
3. Từ cửa sổ New Project, click vào mục Visual Studio Project và sau đó chọn Windows Application.
4. Nhập tên cho dự án và click OK. Trong ví dụ này, chúng ta đặt tên cho dự án là VB\_ViewReport.
5. Khi dự án mới xuất hiện với cửa sổ form trống, click View|Toolbox và chọn công cụ điều khiển MainMenu từ cửa sổ ToolBox.

Kéo công cụ điều khiển MainMenu vào trong Form. Trong ô Type Here, gõ vào: File. Một ô xuất hiện ngay bên dưới, gõ vào: Print Report, như hình 9.5.



Hình 9.5: Thêm menu vào cửa sổ chương trình.

6. Tiếp theo, chọn thành phần điều khiển *CrystalReportViewer* trong cửa sổ công cụ và kéo nó vào Form. Đặt thuộc tính *Dock* cho *CrystalReportViewer* là *Fill*.

7. Bạn viết mã cho sự kiện Form Load nạp báo biểu như sau:

```
CrystalReportViewer1.ReportSource="d:\roster.rpt"
```

8. Trên Form, double click vào tùy chọn Print Report trên menu chính và nhập vào đoạn mã sau để hiển thị hộp thoại in báo biểu:

```
CrystalReportViewer1.PrintReport
```

9. Click Debug | Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Form sẽ hiển thị báo biểu, khi click File | Print Report, hộp thoại in chuẩn của Windows xuất hiện (hình 9.6).



Hình 9.6: Màn hình PrintReport.

10. Lưu đề án với tên Visual Basic\_PrintExportReport; chúng ta sẽ tiếp tục đề án này trong phần tiếp theo.

Ngoài việc in báo biểu, bạn cũng có thể kết xuất số liệu báo biểu sử dụng phương pháp *ExportReport*. Phương pháp này sẽ mở một hộp thoại Save As chuẩn của Windows, cho phép bạn lựa chọn dạng kết xuất dữ liệu cũng như nhập vào tên file và xác định vị trí cho file kết xuất dữ liệu. Để lưu dữ liệu của báo biểu sang định dạng khác, bạn thực hiện các bước sau:

1. Click Start | Visual Studio.NET 2003 (hoặc Visual Studio 2005).
2. Click File | Open | Project. Mở dự án Visual Basic\_PrintExportReport mà bạn đang làm việc trong phần trước.
3. Dưới tùy chọn Print Report, trong hộp thoại Type Here gõ vào một mục menu mới mang tên Export Report.
4. Tiếp theo, double click tùy chọn Export Report để viết mã cho mục chọn trên menu này như sau:

```
CrystalReportViewer1.ExportReport()
```

5. Click Debug | Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Form xuất hiện và trình bày báo biểu. Khi click File | Export Report, hộp thoại Save As chuẩn của Windows xuất hiện.
6. Nhập vào tên file và click nút Save để lưu file.

Mô hình đối tượng CrystalReportViewer cung cấp các kiểu kết xuất dữ liệu sau:

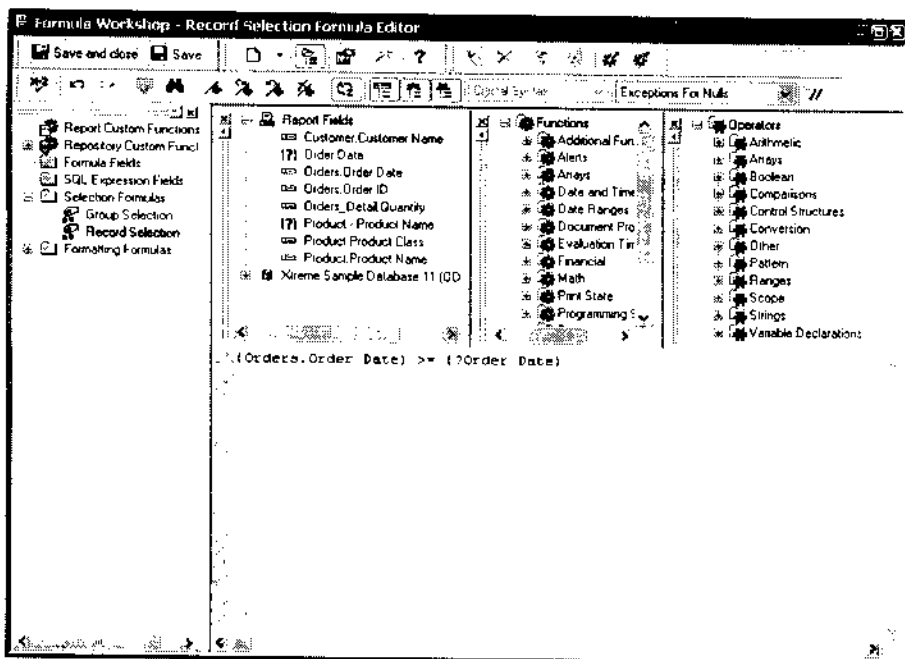
- Crystal Reports (.rpt)*
- Adobe Acrobat (.pdf)*
- Microsoft Excel 97 - 2000 (.xls)*
- Microsoft Excel 92 - 2000 Data Only (.xls)*
- Microsoft Word (.doc)*
- Microsoft Word - Editable (.rtf)*
- Rich Text Format (.rtf)*

## 2.4. Thay đổi điều kiện lọc mẫu tin của báo biểu

Như bạn đã học, báo biểu Crystal cho phép bạn viết công thức lọc mẫu tin thông qua Record Selection. Ví dụ:

```
{Customer.Country}="USA"
```

Công thức lọc mẫu tin trên sẽ lọc và hiển thị những mẫu tin có mã quốc gia Country là USA. Bạn có thể xem các điều kiện lọc đang sử dụng trong báo biểu bằng cách chọn Report | Selection Formulas | Record trong môi trường Crystal Reports. Hộp thoại Record Selection Formula Editor sẽ xuất hiện như hình 9.7.



Hình 9.7: Hộp thoại Record Selection Formula Editor.

Bất kỳ điều kiện lọc mẫu tin nào mà bạn tạo ra trong công cụ này đều lưu trữ cùng với báo biểu. Khi tích hợp các báo biểu vào trong ứng dụng, bạn có hai thuộc tính khác nhau để đặt điều kiện lọc cho mẫu tin. Thuộc tính thứ nhất, *SelectionFormula*, có thể được sử dụng để tìm hoặc đặt công thức lọc mẫu tin cho báo biểu. Thuộc tính thứ hai, *ViewTimeSelectionFormula*, được sử dụng để kết nối với một công thức lọc báo biểu dựa trên công thức lọc đã tồn tại. Ví dụ, nếu báo biểu đã có một điều kiện lọc mẫu tin:

```
{Customer.Country}="USA"
```

bạn sử dụng thuộc tính *ViewTimeSelectionFormula* để thêm điều kiện lọc:

```
{Customer.Country}="NSW"
```

kết quả công thức lọc mẫu tin sẽ trở thành:

```
{Customer.Country}="USA" and {Customer.Country}="NSW"
```

Ví dụ dưới đây sẽ hướng dẫn bạn cách thiết lập công thức lọc mẫu tin khi đưa báo biểu vào ứng dụng:

1. Từ menu Start, mở Visual Studio .NET 2003 (hoặc Visual Studio 2005).
2. Click File | New | Project.
3. Từ cửa sổ New Project, click mục Visual Studio Project và chọn Windows Application.
4. Nhập vào tên dự án, click OK. Trong ví dụ này, chúng ta đặt tên dự án là VB\_RecordSelection.
5. Một dự án mới xuất hiện cùng cửa sổ Form trống. Click View | Toolbox và kéo thành phần điều khiển *CrystalReportViewer* vào trong Form. Đặt thuộc tính *Dock* của *CrystalReportViewer* là *Fill*.
6. Viết mã cho sự kiện Form Load để nạp báo biểu *Dynselect.RPT*. Báo biểu mẫu *Dynselect.RPT* được chứa trong thư mục *Project* trên CD ROM. Bạn có thể sửa đổi vị trí file cho phù hợp với đường dẫn.

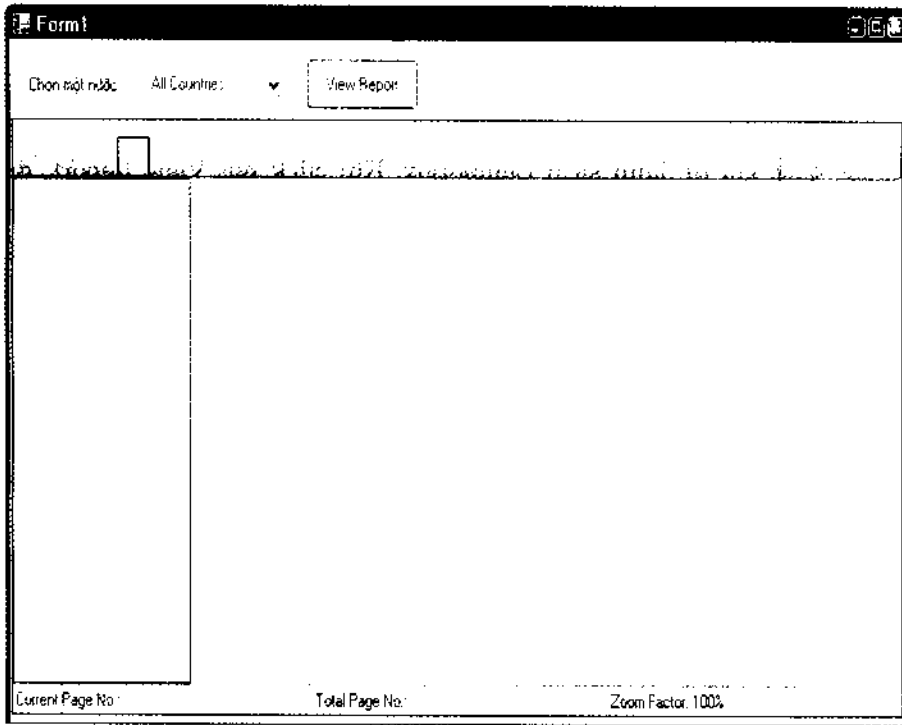
```
CrystalReportViewer1.ReportSource= "d:\dynselect.rpt"  
CrystalReportViewer1.DisplayGroupTree=False
```

7. Tiếp theo, bạn kéo thêm vào Form một đối tượng nhãn Label, một đối tượng ComboBox, và một đối tượng Button đặt phía trên Viewer như hình 9.8:



Hình 9.8: Các thành phần dùng để lọc dữ liệu báo biểu.

8. Thay đổi thuộc tính Text của Label là *Select a Country*. Và của Combo Box là *All Countries*, của Button là *View Report*. Giao diện Form bây giờ sẽ giống như hình 9.9.



**Hình 9.9:** Form được hoàn thành.

9. Bây giờ chúng ta cần nhập vào danh sách một số nước để người dùng lựa chọn. Bạn chọn thuộc tính Items của ComboBox và nhập vào các tên nước sau:

- ◆ USA
- ◆ Australia
- ◆ Canada
- ◆ Mexico

10. Tiếp theo chúng ta cần viết mã cho nút lệnh View Report để tạo công thức lọc mẫu tin và chuyển nó vào trong báo biểu. Bạn Double click vào nút View Report để viết mã cho sự kiện Click của nút nhấn như sau:

```

If ComboBox1.Text ="All Countries" Then
    Crystal ReportsViewer1.SelectionFormula=""
Else
    CrystalReportViewer1.SelectionFormula="
        {Customer.Country} "= " & ComboBox1.Text & "
    
```

```
End If
```

```
Crystal ReportsViewer1.RefreshReport ()
```

11. Cuối cùng, click Debug | Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Form xuất hiện cùng với nội dung báo biểu. Bạn có thể chọn các quốc gia từ danh sách ComboBox. Khi click nút View Report, điều kiện lọc mẫu tin sẽ được áp dụng và báo biểu sẽ được cập nhật lại.

Đây là một ví dụ đơn giản áp dụng điều kiện lọc mẫu tin cho báo biểu. Bạn có thể tạo ra các công thức lọc phức tạp hơn theo yêu cầu của mình.

## 2.5. Làm việc với các Field tham số

Với báo biểu trong Crystal Reports, bạn có thể truyền tham số động và tham số tĩnh. Khi thực thi báo biểu, Crystal sẽ hiển thị hộp thoại để bạn chọn hay nhập vào các giá trị cho tham số (hình 9.10).



Hình 9.10: Hộp thoại truyền tham số trong Crystal Reports XI.

Tuy nhiên, nếu muốn tùy biến hộp thoại nhập giá trị tham số theo cách riêng của ứng dụng, bạn sẽ chuyển giá trị tham số cho Report Viewer thông qua lớp *ParameterField*. Ngoài ra, còn một lớp khác mang tên *ParameterFields* (chú ý số nhiều có 's'), là tập hợp chứa tất cả các Field tham số trong báo biểu.

Để làm việc với các Field tham số, trước hết chúng ta sẽ xem làm cách nào có thể đọc các thông tin về Field tham số trong báo biểu.



### 2.5.1. Đọc thông tin từ Field tham số

Ngoài việc đặt giá trị cho Field tham số, bạn có thể đọc thông tin về các Field tham số trong báo biểu. Những thông tin cơ bản về Field tham số gồm:

- *Name*: Tên của tham số.
- *ReportParametersType*: Kiểu của tham số (String, Date, v.v...).
- *ParameterValueType*: Kiểu giá trị được yêu cầu bởi tham số (String, Date, v.v...).
- *PromptText*: Văn bản được hiển thị nhắc nhở người sử dụng nhập vào giá trị cho tham số.
- *ReportName*: Tên của báo biểu chính hoặc báo biểu con nơi tham số xuất hiện. Nếu nó xuất hiện trong phần chính của Report thì thuộc tính này sẽ được để trống.
- *HasCurrentValue*: Cho biết tham số này có một giá trị hay không.
- *MinimumValue*: Trả về giá trị nhỏ nhất sẽ được tham số chấp nhận.
- *MaximumValue*: Trả về giá trị lớn nhất sẽ được tham số chấp nhận.
- *EditMask*: Mặt nạ nhập liệu cho tham số.

Để tìm hiểu các thuộc tính này làm việc như thế nào, trong ví dụ tiếp theo của chúng ta sẽ lần lượt sử dụng chúng:

1. Mở Visual Studio .NET 2003 (hoặc Visual Studio 2005).
2. Click File | New | Project.
3. Từ cửa sổ New Project, click vào mục Visual Studio Project và chọn Windows Application.
4. Nhập tên và xác định đường dẫn chứa dự án và click OK. Trong ví dụ này chúng ta đặt tên dự án là VB\_Parameters.
5. Một dự án mới sẽ xuất hiện cùng cửa sổ Form trống. Click View | Toolbox và kéo thành phần điều khiển *CrystalReportViewer* vào Form. Đặt thuộc tính Dock cho *CrystalReportViewer* là Fill.
6. Viết mã cho sự kiện Form Load để nạp báo biểu Paraminfo.RPT. Báo biểu mẫu Paraminfo.RPT được chứa trong thư mục Project trên CD ROM. Bạn có thể sửa đổi vị trí file cho phù hợp với đường dẫn.

```
CrystalReportViewer1.ReportSource="d:\Paraminfo.rpt"
CrystalReportViewer1.DisplayGroupTree=False
```

7. Để kiểm tra báo biểu nạp đúng đắn, click Debug | Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Form xuất hiện và hiển thị báo biểu. Đóng cửa sổ quay trở về môi trường Visual Studio.
8. Tiếp theo chúng ta có thể khai báo các biến tham số *ParameterField*. Bạn viết thêm đoạn mã sau cho sự kiện Form Load:

```
CrystalReportViewer1.ReportSource="d:\Paraminfo.rpt"
CrystalReportViewer1.DisplayGroupTree=False
```

*'Phần mã viết thêm*

```
Dim ParameterFields As
    CrystalDecisions.Shared.ParameterFields
Dim ParameterField As
    CrystalDecisions.Shared.ParameterField

ParameterFields =
    CrystalReportViewer1.ParameterFieldInfo

Dim DisplayText
For Each ParameterField In ParameterFields
    DisplayText = " Parameter Name: " &
        ParameterField.Name & Chr(13)
    DisplayText = DisplayText & " Parameter Type: " &
        ParameterField.ReportParameterType.ToString
        & Chr(13)
    DisplayText = DisplayText &
        " Parameter Value Type: " &
        ParameterField.ParameterValueType.ToString &
        Chr(13)
    DisplayText = DisplayText & " Prompting Text: " &
        ParameterField.PromptText & Chr(13)
    DisplayText = DisplayText & " Report Name: " &
        ParameterField.ReportName & Chr(13)
    DisplayText = DisplayText & " Has Current Value: " &
        ParameterField.HasCurrentValue & Chr(13)
    DisplayText = DisplayText & " Minimum Value: " &
        ParameterField.MinimumValue & Chr(13)
    DisplayText = DisplayText & " Maximum Value: " &
        ParameterField.MaximumValue & Chr(13)
    DisplayText = DisplayText & " Edit Mask: " &
        ParameterField.EditMask & Chr(13)
    MsgBox(DisplayText)
Next
```

9. Click **Debug | Start** hoặc nhấn phím **F5** để thực thi lại ứng dụng. Một hộp thoại thông báo sẽ được hiển thị cho biết thông tin về các Field tham số lưu trong báo biểu.
10. Click **OK** để tiếp tục, nhập vào giá trị cho tham số và click **OK** để xem báo biểu. Lưu dự án với tên **VB\_Parameter**.

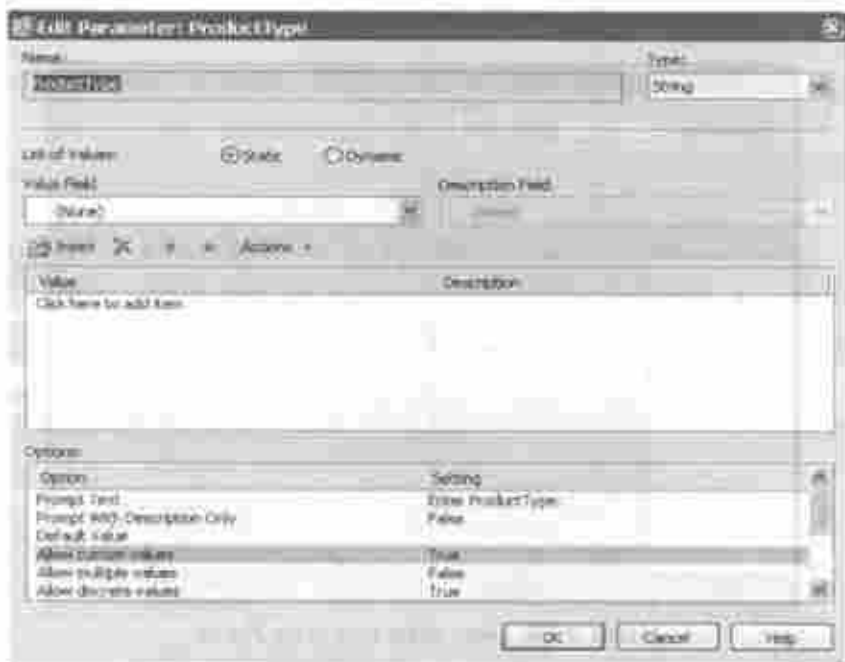
Trong phần tiếp theo chúng ta sẽ học cách đọc các giá trị mặc định của tham số lưu trong báo biểu và gán giá trị mới cho Field tham số.

### 2.5.2. Đọc và đặt các giá trị mới cho field tham số

Khi tạo các Field tham số trong Crystal Reports, bạn có thể tạo một danh sách giá trị tính cho phép người sử dụng chọn. Danh sách này được gọi là *giá trị mặc định* cho tham số.

Ví dụ, nếu có một tham số **State**, bạn có thể tạo giá trị mặc định bao gồm chuỗi diễn giải **New York** và một giá trị kèm theo là **NY**. Hình 9.11 là cửa sổ tạo các giá trị mặc định cho tham số trong Crystal Reports.

Khi người sử dụng được yêu cầu chọn hay nhập giá trị cho tham số, danh sách các giá trị mặc định này sẽ hiển thị như hình 9.12.



Hình 9.11: Đặt các giá trị mặc định cho tham số.



```
CrystalReportViewer1.ReportSource="d: \Dynparm.rpt"
CrystalReportViewer1.DisplayGroupTree=False
ParameterFields =
```

```
    CrystalReportViewer1.ParameterFieldInfo
```

```
    ParameterField =ParameterFields(0)
    Label1.Text= ParameterField.PromptText
```

```
For I As Integer= 0 to
```

```
    (ParameterField.DafaultValues.Count-1 )
    ComboBox1.Item.Add(ParameterField.
```

```
    DefaultValues.Item(i). Description)
```

```
Next
```

```
ParameterDiscreteValue=New
CrystalDecisions.Shared.ParameterDiscreteValue
ParameterDiscreteValue.Value=""
```

```
ParameterField.CurrentValues.Add(
    ParameterDiscreteValue)
```

4. Double click vào nút View Report trong Form và nhập đoạn mã như sau cho sự kiện Click của nút nhấn:

```
CrystalReportViewer1.ReportSource="d: \ Dynparm.rpt"
ParameterFields = CrystalReportViewer1.Parameter
FieldInfoParameterField =ParameterFields(0)
```

```
    ParameterDiscreteValue=New
CrystalDecisions.Shared.ParameterDiscreteValue
ParameterDiscreteValue.Value=ComboBox1.SelectedItem
ParameterField.CurrentValues.Add(
    ParameterDiscreteValue)
```

5. Click Debug |Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Click vào ComboBox để chọn giá trị cho tham số định nghĩa sẵn trong báo biểu (hình 9.14).
6. Chọn một giá trị và click vào nút View Report. Bây giờ báo biểu sẽ được cập nhật lại theo giá trị tham số mà bạn đã chọn.



**Hình 9.14:** Form với danh sách giá trị mặc định đọc được từ báo biểu.

7. Thay vì đọc các giá trị mặc định từ báo biểu, bạn có thể tự xây dựng các giá trị mặc định này trong ứng dụng của mình, sau đó gọi phương thức là `CurrentValues.Add` để chuyển giá trị tham số cho báo biểu thực thi.

*Ghi chú:* Một thủ thuật khác, khi nào làm việc với báo biểu mà báo biểu đó có chứa báo biểu con, trong báo biểu con yêu cầu truyền tham số, bạn có thể chuyển giá trị tham số cho báo biểu con như sau:

```
ParameterField = ParameterField(0, "SubReport1")
```

### 2.5.3. Thiết đặt giá trị phạm vi cho Field tham số

Một kiểu Field tham số phổ biến khác là tham số phạm vi. Tham số phạm vi được định nghĩa bởi một giá trị căn trên và cận dưới. Field tham số phạm vi thường được sử dụng với kiểu Date hoặc kiểu Number.

Ví dụ, bạn có thể tạo ra một tham số phạm vi cho phép bạn chọn những khách hàng với doanh số đạt được trong khoảng \$10000 đến \$20000. Hoặc bạn có thể tạo ra một tham số phạm vi cho phép nhập vào ngày bắt đầu và ngày kết thúc, như vậy bạn có thể chọn tất cả mẫu tin nằm trong khoảng giữa của hai ngày này (hình 9.15).



**Hình 9.15:** Tham số phạm vi theo kiểu ngày tháng gồm ngày bắt đầu và ngày kết thúc.

Bạn có thể đặt giá trị cho kiểu tham số này tương tự những tham số thông thường. Tuy nhiên, thay vì tạo ra những giá trị riêng biệt đơn giản, bạn sẽ chuyển cho báo biểu giá trị đầu và giá trị cuối của tham số. Trong ví dụ tiếp theo, chúng ta sẽ tạo ra một giao diện thân thuộc cho phép người sử dụng nhập vào ngày bắt đầu và ngày kết thúc và sau đó chuyển giá trị này vào trong báo biểu. Để tạo ứng dụng này, bạn thực hiện các bước sau:

1. Mở Visual Studio, Click File | New | Project.
2. Từ cửa sổ New Project, click vào mục Visual Studio Project và chọn Windows Application.
3. Nhập tên và xác định đường dẫn chứa dự án và click OK. Trong ví dụ này chúng ta đặt tên dự án là VB\_RangeParam.
4. Một dự án mới sẽ được mở cùng với cửa sổ Form trống. Click View | ToolBox và kéo điều khiển *CrystalReportViewer* vào Form. Đặt thuộc tính Dock cho *CrystalReportViewer* là Fill.
5. Viết mã cho sự kiện Form Load để nạp báo biểu Rangeparam.RPT. Báo biểu mẫu Rangeparam.RPT được chứa trong thư mục Project trên CD ROM. Bạn có thể sửa đổi vị trí file cho phù hợp với đường dẫn.

```
CrystalReportViewer1.ReportSouce="d:\ RangeParam.rpt"  
CrystalReportViewer1.DisplayGroupTree=False
```

6. Click Debug | Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Form xuất hiện cùng với nội dung báo biểu. Đóng cửa sổ Form và trở về cửa sổ Visual Studio.
7. Tiếp theo, thêm hai TextBox và ba nút Button vào Form. Gán thuộc tính Text cho 2 TextBox là "" và thuộc tính Text của hai nút Button là "...". Nút Button thứ ba đặt tên là View Report. Thêm hai nhãn Label trước hai TextBox là Start Date và End Date. Bây giờ giao diện Form sẽ trông giống như hình 9.16.
8. Để chọn ngày tháng một cách dễ dàng, bạn kéo điều khiển *MonthCalendar* trên cửa sổ công cụ ToolBox đặt gần các nút nhấn. Khi người dùng kích vào nút nhấn, đối tượng lịch này sẽ hiển thị cho phép bạn chọn ngày tháng.
9. Thay đổi thuộc tính Visible của hai điều khiển *MonthCalendar* là False.



Hình 9.16: Giao diện Form hoàn tất.

10. Tiếp theo, nhập đoạn mã sau cho sự kiện Click của nút nhấn:

```
MonthCalendar1.Show()
```

11. Nhấp đôi chuột vào điều khiển *MonthCalendar* để mở sự kiện *DateSelected* khi người sử dụng chọn lựa ngày tháng. Trong sự kiện này chúng ta sẽ chuyển giá trị ngày tháng được chọn cho ô *TextBox*.

```
Private Sub MonthCalendar1_DateSelected(  
    ByVal sender As System.Object,  
    ByVal e As DateRangeEventArgs)  
    Handles MonthCalendar1.DateSelected  
    TextBox1.Text = MonthCalendar1.DateSelected  
    MonthCalendar1.Hide()  
End Sub
```

12. Tiếp theo bạn khai báo và nhập vào đoạn mã dưới đây trong lớp Form:

```
Dim ParameterFields As  
    CrystalDecisions.Shared.ParameterFields  
Dim ParameterField As  
    CrystalDecisions.Shared.ParameterField  
Dim ParameterRangeValue As  
    CrystalDecisions.Shared.ParameterRangeValues
```

13. Mở lại sự kiện *Form Load*, viết mã cho sự kiện này để thiết đặt giá trị *Field* tham số cho báo biểu:

```
CrystalReportViewer1.  
    ReportSource="d:\RangeParm.rpt"  
CrystalReportViewer1.DisplayGroupTree=False
```



```
ParameterFields =  
    CrystalReportViewer1.ParameterFieldInfo  
  
ParameterField =ParameterFields(0)  
ParameterRangeValue =New  
CrystalDecision.Shared.ParameterRangeValue  
ParameterRangeValue.StartValue="1/1/1"  
ParameterRangeValue.EndValue="1/1/1"  
  
ParameterRangeValue.LowerBoundType=CrystalDecision.  
    [Shared].RangeBoundType.BoundInclusive  
ParameterRangeValue.UpperBoundType=CrystalDecision.  
    [Shared].RangeBoundType.BoundInclusive  
ParameterField.CurrentValues.Add(ParameterRangeValue)
```

14. Double click vào nút View Report và nhập đoạn mã tiếp theo để thực thi khi người sử dụng click nút nhấn. Đoạn mã này hầu như không khác gì với đoạn mã sử dụng trong sự kiện Form Load.

```
CrystalReportViewer1.ReportSource="d:\RangeParm.rpt"  
CrystalReportViewer1.DisplayGroupTree=False  
  
ParameterFields =  
    CrystalReportViewer1.ParameterFieldInfo  
ParameterField =ParameterFields(0)  
  
ParameterRangeValue =New  
    CrystalDecision.Shared.ParameterRangeValue  
  
ParameterRangeValue.StartValue=TextBox1.Text  
ParameterRangeValue.EndValue=TextBox2.Text  
  
ParameterRangeValue.LowerBoundType=CrystalDecision.  
    [Shared].RangeBoundType.BoundInclusive  
ParameterRangeValue.UpperBoundType=CrystalDecision.  
    [Shared].RangeBoundType.BoundInclusive  
ParameterField.CurrentValues.Add(ParameterRangeValue)
```

15. Click Debug | Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Form xuất hiện, bạn có thể nhập vào ngày bắt đầu và ngày kết thúc.

Click vào nút (...) để mở khung lịch biểu cho phép bạn chọn ngày tháng thông qua MonthCalendar, click vào nút View Report để chuyển giá trị này cho báo biểu và xem nội dung báo biểu đã được lọc lại.

## 2.6. Đăng nhập cơ sở dữ liệu

Hầu hết các hệ cơ sở dữ liệu lớn đều có chế độ bảo mật yêu cầu bạn cung cấp thông tin đăng nhập trước khi cho phép báo biểu truy xuất nguồn dữ liệu. Thông thường khi bạn thực thi báo biểu, Crystal sẽ hiển thị hộp thoại yêu cầu bạn nhập vào tên và mật khẩu dùng cho đăng nhập vào cơ sở dữ liệu như hình 9.17. Để người dùng không phải nhập thông tin đăng nhập này mỗi khi thực thi báo biểu, bạn gán thông tin đăng nhập sẵn cho Viewer thông qua lớp LogonInfo như sau:

```
Dim LogonInfo as CrystalDecisions.Shared.TableLogOnInfo
```

```
CrystalReportViewer1.LogOnInfo=New
```

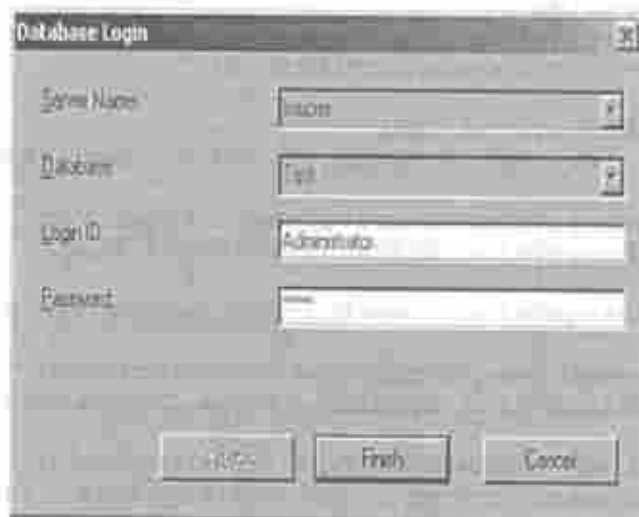
```
CrystalDecisions.Shared.TableLogOnInfos
```

```
LogOnInfo.TableName="Products"
```

```
LogOnInfo.ConnectionInfo.UserID="Admin"
```

```
LogOnInfo.ConnectionInfo.Password=""
```

```
CrystalReportViewer1.LogOnInfo.Add(LogOnInfo)
```



Hình 9.17: Nhập vào tên và mật khẩu để đăng nhập vào nguồn cơ sở dữ liệu.

Bạn phải chỉ định rõ bảng dữ liệu nào muốn truy xuất cùng với tên đăng nhập và mật khẩu. Bạn không thể áp dụng một lần thiết lập thông tin đăng nhập cho mọi bảng. Muốn thực hiện điều này bạn phải dùng mô hình đối tượng ReportDocument thay cho CrystalReportViewer. Mô hình đối tượng ReportDocument cho phép duyệt qua tất cả các bảng trong báo biểu và thiết lập các thông số đăng nhập cần thiết.

## **3. LÀM VIỆC VỚI MÔ HÌNH ĐỐI TƯỢNG REPORTDOCUMENT**

Mô hình đối tượng ReportDocument cung cấp các phương thức và thuộc tính cho phép bạn điều khiển ở mức thấp tất cả thông tin có trong cấu trúc của báo biểu.

### **3.1. Chọn phương thức ràng buộc**

Các báo biểu mà chúng ta làm việc cho đến bây giờ thường sử dụng mô hình đối tượng Viewer chỉ định file báo biểu thông qua thuộc tính ReportSource:

```
Crystal ReportsViewer1.ReportSource="c:\myreport.rpt"
```

Tuy nhiên, đây không phải là cách duy nhất mà bạn có thể dùng để truy cập báo biểu từ ứng dụng (mặc dù nó được xem là cách dễ nhất). Khi tích hợp Crystal Reports vào trong ứng dụng, bạn có thể chọn đặt báo biểu độc lập bên ngoài ứng dụng hoặc nhúng chúng vào trong cùng ứng dụng. Lợi ích của việc lưu báo biểu ngoài ứng dụng là ứng dụng có thể thêm nhiều báo biểu mới hoặc việc sửa đổi những báo biểu đã tồn tại. Nhúng báo biểu trực tiếp vào ứng dụng làm cho ứng dụng chạy nhanh hơn và không cho phép người dùng thay đổi những báo biểu quan trọng.

Để kết nối tới một báo biểu được lưu bên ngoài ứng dụng, bạn thực hiện các bước sau:

1. Mở Visual Studio, click chọn File | New | Project.
2. Từ cửa sổ New Project, click vào mục Visual Studio Project và chọn Windows Application.
3. Nhập tên cho dự án và click OK. Trong ví dụ này chúng ta đặt tên dự án là VB\_ExternalBind. Một dự án mới sẽ xuất hiện cùng cửa sổ Form trống.

4. Bạn thêm dòng sau vào đầu lớp Form, khai báo sử dụng thư viện Crystal:

```
Imports CrystalDecisions.CrystalReport.Engine
```

5. Tiếp theo, kéo điều khiển *CrystalReportViewer* vào trong Form và đặt thuộc tính Dock là Fill.
6. Double click vào Form để viết mã cho sự kiện Form Load như sau:

```
Dim ExternalBindReport as ReportDocument =  
    New ReportDocument  
ExternalBindReport.Load ("C:\binding.rpt")  
CrystalReportViewer1.ReportSource=ExternalBindReport
```

7. Click Debug | Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng, báo biểu sẽ hiển thị mặc dù chúng không nằm trong ứng dụng, bạn có thể đặt báo biểu ở bất kỳ đâu và chỉ cần chỉ đường dẫn để Crystal nạp báo biểu vào ứng dụng.

Để kết nối với một báo biểu lưu trữ bên trong ứng dụng, bạn thực hiện các bước sau:

1. Mở Visual Studio, click chọn File | New | Project.
2. Từ cửa sổ New Project, click vào mục Visual Studio Project và chọn Windows Application.
3. Nhập tên cho dự án và click OK. Trong ví dụ này chúng ta đặt tên dự án là VB\_EmbeddedBind. Một dự án mới sẽ xuất hiện cùng cửa sổ Form trống.
4. Đầu tiên chúng ta cần chỉ rõ báo biểu muốn làm việc, để thêm một báo biểu vào trong dự án, chọn Project | Add Existing Item. Bạn hãy chọn một báo biểu mẫu. Ví dụ ta chọn báo biểu binding.rpt chứa trong thư mục Projects trên đĩa CD ROM.
5. Từ thanh công cụ, kéo biểu tượng ReportDocument lên Form. Hộp thoại sẽ xuất hiện, nó sẽ cho phép bạn chọn các báo biểu hiện có trong dự án. Chọn một báo biểu và click OK.
6. Tiếp theo, viết mã cho sự kiện Form Load của Form như sau:

```
Dim EmbeddedBindReport as New Binding
```

7. Bây giờ bạn có thể làm việc và điều khiển báo biểu này thông qua thuộc tính, phương thức của EmbeddedBindReport trong mô hình đối

tượng `ReportDocument`. Để xem báo biểu, bạn kéo điều khiển `CrystalReportViewer` vào trong `Form`, tiếp theo nhập thêm đoạn mã sau vào `Form Load`:

```
CrystalReportViewer1.ReportSource= EmbeddedBindReport
```

8. Sau cùng, click `Debug | Start` hoặc nhấn phím `F5` để bắt đầu thực thi ứng dụng. Báo biểu của bạn giờ đây được biên dịch và kết hợp vào chung với ứng dụng khi thực thi, `Viewer` sẽ đọc báo biểu từ bên trong ứng dụng và thực thi tương tự như trường hợp báo biểu được lưu bên ngoài. Bạn không cần chép file báo biểu theo chương trình khi cài đặt ứng dụng cho người dùng cuối.

### 3.2. In báo biểu

Mặc dù công cụ `Viewer` trong `Crystal Reports` cho phép xem trước báo biểu và sau đó nhấn vào nút `Print` để in báo biểu, nhưng sẽ có lúc bạn cần in trực tiếp mà không phải hiển thị `Viewer`. Sử dụng mô hình đối tượng `ReportDocument`, bạn có thể in trực tiếp kết xuất của báo biểu ra máy in. Bạn còn có thể điều khiển chọn máy in, số trang in, v.v... Trong phần tiếp theo, chúng ta sẽ tìm hiểu kỹ thuật in bằng mô hình đối tượng `ReportDocument`.

Phương pháp dễ nhất để in báo biểu là gọi phương thức `PrintToPrinter`. Phương thức này có thể được sử dụng để in báo biểu ra một máy in mặc định. Phương thức này sử dụng thêm các thuộc tính sau:

- ◆ `Number of Copies` (số bản copy).
- ◆ `Start Page` (trang bắt đầu).
- ◆ `End Page` (trang kết thúc).

Ví dụ dưới đây sẽ sử dụng phương thức này để in báo biểu ra máy in mặc định:

1. Mở `Visual Studio`, click chọn `File | New | Project`.
2. Từ cửa sổ `New Project`, click vào mục `Visual Studio Project` và chọn `Windows Application`.
3. Nhập tên cho dự án và click `OK`. Trong ví dụ này chúng ta đặt tên dự án là `VB_PrintReport`. Một dự án mới sẽ xuất hiện cùng cửa sổ `Form` trống.

- Đầu tiên cần chỉ rõ báo biểu chúng ta muốn làm việc. Để thêm một báo biểu vào dự án, chọn Project | Add Existing Item và chọn file báo biểu có sẵn. Trong ví dụ này ta chọn báo biểu mẫu InvoiceReport.rpt trong thư mục Projects trên đĩa CD ROM.
- Viết mã cho sự kiện Form Load như sau:

```
Dim InvoiceReport as New InvoiceReport ()
```

- Từ cửa sổ ToolBox, kéo một nút Button vào Form.
- Viết mã cho sự kiện Click của nút nhấn như sau:

```
InvoiceReport.PrintToPrinter(1, True, 0, 0)
```

- Cuối cùng, click Debug | Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Báo biểu bây giờ sẽ được in trực tiếp ra máy in mặc định mà không cần hiển thị Viewer.
- Nếu muốn in ra một máy in khác (trường hợp máy tính kết nối với nhiều máy in) bạn có thể chỉ định thêm thông số máy in như sau:

```
With InvoiceReport.PrintOptions
```

```
.PrinterName= "Canon Pixma iP1000"
```

```
.PaperSize=CrystalDecisions.
```

```
[Shared].PaperSize.PaperA4
```

```
.PaperOrientation = CrystalDecisions.[Shared].
```

```
PaperOrientation.Landscape
```

```
End with
```

Đoạn mã trên yêu cầu chương trình in báo biểu ra máy in Canon trên khổ giấy A4 theo hướng in ngang.

### 3.3. Xuất dữ liệu báo biểu

Bạn có thể kết xuất nội dung báo biểu ra file theo các định dạng khác. Quá trình này gọi là xuất dữ liệu (export).

Có hai phương thức kết xuất khác nhau - thứ nhất bạn gọi phương thức *ExportToDisk* và chỉ rõ dạng file muốn kết xuất. Thứ hai, là phương thức *Export* sử dụng phức tạp hơn với các tùy chọn kết xuất mở rộng.

Đối tượng *ReportDocument* hỗ trợ những dạng kết xuất sau:

<b>Định dạng</b>	<b>Dạng kết xuất</b>
<i>Adobe Acrobat PDF</i>	<i>PortableDocFormat</i>
<i>Crystal Reports</i>	<i>CrystalReport</i>
<i>HTML 3.2</i>	<i>HTML32</i>
<i>HTML4.0</i>	<i>HTML40</i>
<i>Microsoft Excel</i>	<i>ExcelRecord</i>
<i>Microsoft Word</i>	<i>WordForWindows</i>
<i>No Format</i>	<i>NoFormat</i>
<i>Rich Text</i>	<i>RichText</i>
<i>Rich Text (Editable)</i>	<i>EditableRTF</i>
<i>Text</i>	<i>Text</i>
<i>Text(CharacterSeparated)</i>	<i>CharacterSeparatedValues</i>
<i>Text (Tab Separated)</i>	<i>TabSeparatedText</i>

Để sử dụng phương thức *ExportToDisk*, tất cả những gì bạn cần làm là chỉ định dạng kết xuất và đường dẫn file cần lưu. Để sử dụng phương thức *ExportToDisk*, bạn thực hiện các bước sau:

1. Mở Visual Studio, click chọn File | New | Project.
2. Từ cửa sổ New Project, click vào mục Visual Studio Project và chọn Windows Application.
3. Nhập tên cho dự án và click OK. Trong ví dụ này chúng ta đặt tên dự án là VB\_SimpleExportReport. Một dự án mới sẽ xuất hiện cùng cửa sổ Form trống.
4. Đầu tiên bạn cần chỉ định báo biểu chúng ta muốn làm việc. Để thêm một báo biểu vào trong dự án, chọn Project | Add Existing Item và trở đến file báo biểu cần sử dụng. Trong ví dụ này chúng ta sử dụng báo biểu mẫu *ExportReport.rpt* chứa trong thư mục Projects trên đĩa CD ROM.
5. Từ cửa sổ công cụ Toolbox, kéo đối tượng *ReportDocument* vào Form – Visual Studio sẽ mở một hộp thoại với danh sách các báo biểu đang có trong dự án. Hãy chọn báo biểu chúng ta vừa thêm vào.

6. Bạn viết mã cho sự kiện Form Load như sau:

```
Dim ReportToExport as New exportReport ()
ReportToExport.ExportToDisk
(CrystalDescisions.[Shared].ExportFormatType.Excel,
"c:\simpleexport.xls")
```

7. Click Debug / Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Báo biểu bây giờ sẽ được kết xuất ra file simpleexport.xls theo định dạng Excel.
8. Phương thức ExportToDisk sử dụng nhanh và khá dễ, nhưng nếu bạn muốn điều khiển quá trình xử lý kết xuất chi tiết hơn thì phải sử dụng phương thức *Export*. Phương thức *Export* cho phép bạn chọn thiết bị mà file sẽ kết xuất.

Nơi lưu trữ	Vị trí
<i>Disk</i>	<i>DiskFile</i>
<i>Exchange Folder</i>	<i>ExchangeFolder</i>
<i>Email (MAPI)</i>	<i>MicrosoftMail</i>
<i>No Destination</i>	<i>NoDestination</i>

Dưới đây là ví dụ kết xuất báo biểu sang dạng PDF và lưu nó vào ổ đĩa cứng bằng phương thức *Export*:

- Mở Visual Studio, click chọn File | New | Project.
- Từ cửa sổ New Project, click vào mục Visual Studio Project và chọn Windows Application.
- Nhập tên cho dự án và click OK. Trong ví dụ này chúng ta đặt tên dự án là VB\_ExportReport. Một dự án mới sẽ xuất hiện cùng cửa sổ Form trống.
- Đầu tiên bạn cần chỉ định báo biểu chúng ta muốn làm việc. Để thêm một báo biểu vào trong dự án, chọn Project | Add Existing Item và trỏ đến file báo biểu cần sử dụng. Trong ví dụ này chúng ta sử dụng báo biểu mẫu ExportReport.rpt chứa trong thư mục Projects trên đĩa CD ROM.
- Từ cửa sổ công cụ Toolbox, kéo đối tượng ReportDocument vào Form - Visual Studio sẽ mở một hộp thoại với danh sách các báo biểu đang có trong dự án. Hãy chọn báo biểu chúng ta vừa thêm vào.



6. Bạn viết mã cho sự kiện Form Load như sau:

```
Dim ReportToExport As New ReportToExport
    ReportToExport.Load()
Dim ExportOptions As New
    CrystalDescisions.Shared.ExportOptions
Dim DiskFileDestinationOptions As New
    CrystalDescisions.Shared.

    DiskFileDestinationOptions
Dim FormatTypeOptions As New
    CrystalDescisions.Shared.

PdfRtfWordFormatOptions
ExportOptions = ReportToExport.ExportOptions

With FormatTypeOptions
    .FirstpageNumber=1
    .LastPageNumber=10
End with

With DiskFileDestinationOptions
    .DiskFileName="c:\advancedexport.pdf"
End with

With ExportOptions
    .ExportDestinationType=CrystalDescisions.

Shared.ExportFormatType.PortableDocFormat
    .ExportDestinationOptions=

    DiskFileDestinationOptions
    .FormatOptions=FormatTypeOptions
End with

ReportToExport.Export()
```

7. Sau cùng, click Debug! Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Báo biểu sẽ được kết xuất và lưu vào ổ đĩa cứng C. Theo cách tương tự bạn có thể kết xuất chuyển báo biểu sang Mail hay thư mục Mail Server của Microsoft Exchange.

### 3.4. Làm việc với các tham số

Mô hình *ReportDocument* cung cấp khả năng làm việc với các Field tham số trong báo biểu tương tự như với mô hình Report Viewer.

Trong ví dụ sau tiếp theo, chúng ta sẽ học cách đọc các tham số trong báo biểu bằng mô hình ReportDocument. Chương trình sẽ hiển thị các giá trị mặc định và chuỗi mô tả liên quan đến các Field tham số có trong báo biểu. Người sử dụng có thể chọn giá trị cho các Field tham số và chương trình sẽ chuyển các giá trị này ngược lại cho báo biểu.

1. Mở Visual Studio, click chọn File | New | Project.
2. Từ cửa sổ New Project, click vào mục Visual Studio Project và chọn Windows Application.
3. Nhập tên cho dự án và click OK. Trong ví dụ này chúng ta đặt tên dự án là VB\_ParameterValues. Một dự án mới sẽ xuất hiện cùng cửa sổ Form trống.
4. Đầu tiên bạn cần chỉ định báo biểu chúng ta muốn làm việc. Để thêm một báo biểu vào trong dự án, chọn Project | Add Existing Item và trở đến file báo biểu cần sử dụng. Trong ví dụ này chúng ta sử dụng báo biểu mẫu Citysales.rpt chứa trong thư mục Projects trên đĩa CD ROM.
5. Từ cửa sổ công cụ Toolbox, kéo đối tượng ReportDocument vào Form – Visual Studio sẽ mở một hộp thoại với danh sách các báo biểu đang có trong dự án. Hãy chọn báo biểu chúng ta vừa thêm vào.
6. Tiếp theo, thêm 5 đối tượng nhãn Label, ComboBox lên Form để hiển thị các giá trị của tham số trong báo biểu.
7. Thêm tiếp vào Form nút Button đặt thuộc tính Text của nút là View Report.
8. Cuối cùng, thêm vào Form điều khiển Crystal Reports Viewer. Bây giờ bạn có thể bắt đầu viết mã cho Form.
9. Double click vào Form và khai báo các biến trong lớp Form như sau:

```
Dim ParameterFieldDefinitions As
    CrystalDecisions.CrystalReports.Engine.
        ParameterFieldDefinitions
Dim ParameterFieldDefinition As
    CrystalDecisions.CrystalReports.Engine.
        ParameterFieldDefinition
Dim defaultParameterValues As
    CrystalDecisions.Shared.ParameterValues
Dim defaultParameterValue As
    CrystalDecisions.Shared.ParameterValue
```

```

Dim currentParameterValues As
    CrystalDecisions.Shared.ParameterValues
Dim currentParameterValue As
    CrystalDecisions.Shared.ParameterValue
Dim ParameterDiscreteValue As
    CrystalDecisions.Shared.ParameterDiscreteValue
Dim ParameterDiscreteValues As
    CrystalDecisions.Shared.ParameterDiscreteValue
Dim ReportDocumentwithParameters As
    CrystalDecisions.CrystalReports.Engine.
        ReportDocument

```

10. Mã của sự kiện Form Load sẽ được cài đặt như sau để duyệt qua danh sách các tham số có sẵn trong báo biểu:

```

ParameterFieldDefinitions =
    ReportDocumentwithParameters.
    DataDefinition.ParameterFields
For i As Integer= 0 To
    ParameterFieldDefinitions.Count - 1
    ParameterFieldDefinition =
        ParameterFieldDefinitions(i)
    defaultParameterValues =
        ParameterFieldDefinition.DefaultValues
    If i = 0 Then
        Label1.Text =
        ParameterFieldDefinition.PromptText.ToString
        For Each defaultParameterValue In
            defaultParameterValues
            ParameterDiscreteValues =
                CType(defaultParameterValue,
                CrystalDecisions.Shared.ParameterValue)
            ComboBox1.Items.Add(
                ParameterDiscreteValues.Value.ToString())
        Next
    End If
    If i = 1 Then
        Label2.Text = ParameterFieldDefinition.
            PromptText.ToString
        For Each defaultParameterValue In
            defaultParameterValues
            ParameterDiscreteValue =
                CType(defaultParameterValue,
                CrystalDecisions.Shared.ParameterDiscreteValue)
            ComboBox2.Items.Add(
                ParameterDiscreteValue.Value.ToString())
        Next
    End If

```

```

End If
If i = 2 Then
  Label3.Text =
ParameterFieldDefinition.PromptText.ToString
  For Each defaultParameterValue In
    defaultParameterValues
    ParameterDiscreteValue =
      CType(defaultParameterValue,
        CrystalDecisions.Shared.
        ParameterDiscreteValue)
    ComboBox3.Items.Add(
      ParameterDiscreteValue.Value.ToString())
  Next
End If
If i = 3 Then
  Label4.Text = ParameterFieldDefinition.
    PromptText.ToString
  For Each defaultParameterValue In
    defaultParameterValues
    ParameterDiscreteValue =
      CType(defaultParameterValue,
        CrystalDecisions.Shared.
        ParameterDiscreteValue)
    ComboBox4.Items.Add(
      ParameterDiscreteValue.Value.ToString())
  Next
End If
If i = 4 Then
  Label5.Text = ParameterFieldDefinition.
    PromptText.ToString
  For Each defaultParameterValue In
    defaultParameterValues
    ParameterDiscreteValue =
      CType(defaultParameterValue,
        CrystalDecisions.Shared.
        ParameterDiscreteValue)
    ComboBox5.Items.Add(
      ParameterDiscreteValue.Value.ToString())
  Next
End If
Next

```

11. Tiếp theo, chúng ta viết mã cho sự kiện Click của nút nhấn để chuyển giá trị mà người dùng chọn cho tham số trong báo biểu:

```

ParameterFieldDefinitions =
ReportDocument.DataDefinition.ParameterFields

```

```
Dim i As Integer
For i = 0 To ParameterFieldDefinitions.Count - 1
    ParameterFieldDefinition =
        ParameterFieldDefinitions(i)
    currentParameterValues =
        ParameterFieldDefinition.CurrentValues
    Dim ParameterDiscreteValue As
    CrystalDecisions.Shared.ParameterDiscreteValue
        = New CrystalDecisions.Shared.
            ParameterDiscreteValue
    If i = 0 Then
        ParameterDiscreteValue.Value =
            ComboBox1.SelectedItem
        currentParameterValues.Clear()
        currentParameterValues.Add(
            ParameterDiscreteValue)
        ParameterFieldDefinition.
            ApplyCurrentValues(currentParameterValues)
    End If
    If i = 1 Then
        ParameterDiscreteValue.Value =
            ComboBox2.SelectedItem
        currentParameterValues.Clear()
        currentParameterValues.Add(
            ParameterDiscreteValue)
        ParameterFieldDefinition.
            ApplyCurrentValues(currentParameterValues)
    End If
    If i = 2 Then
        ParameterDiscreteValue.Value =
            ComboBox3.SelectedItem
        currentParameterValues.Clear()
        currentParameterValues.Add(
            ParameterDiscreteValue)
        ParameterFieldDefinition.
            ApplyCurrentValues(currentParameterValues)
    End If
    If i = 3 Then
        ParameterDiscreteValue.Value =
            ComboBox4.SelectedItem
        currentParameterValues.Clear()
        currentParameterValues.Add(
            ParameterDiscreteValue)
        ParameterFieldDefinition.
            ApplyCurrentValues(currentParameterValues)
    End If
```

```
If i = 4 Then
    ParameterDiscreteValue.Value =
        ComboBox5.SelectedItem
    currentParameterValues.Clear()
    currentParameterValues.Add(
        ParameterDiscreteValue)
    ParameterFieldDefinition.
        ApplyCurrentValues(currentParameterValues)
End If
Next
CrystalReportViewer1.ReportSource = ReportDocument
```

12. Mọi thứ đã sẵn sàng, click Debug | Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Các nhãn sẽ hiển thị chuỗi mô tả tham số còn các đối tượng ComboBox sẽ hiển thị giá trị tham số. Chọn giá trị cho tham số và click vào nút View Report, báo biểu sẽ hiển thị trong Viewer với nội dung lọc theo giá trị do bạn chọn.

### 3.5. Đăng nhập tự động vào cơ sở dữ liệu

Trong phần tìm hiểu về mô hình ReportViewer, chúng ta đã đề cập đến cơ chế đăng nhập tự động vào nguồn cơ sở dữ liệu. Mô hình ReportViewer không cho phép tự động đăng nhập một lần. Bạn phải chỉ định tường minh tên bảng cùng với thông tin đăng nhập trong lớp LogonInfo. Điều này có nghĩa là nếu bạn có nhiều bảng liên kết trong cùng một báo biểu thì bạn phải chỉ định thông tin đăng nhập nhiều lần. Mô hình đối tượng ReportDocument cho phép chúng ta duyệt qua tất cả các nguồn dữ liệu khác nhau đang sử dụng trong báo biểu và xác định thông số đăng nhập chỉ một lần.

Trong dự án mẫu tiếp theo, chúng ta sẽ duyệt qua tất cả các bảng tồn tại bên trong một báo biểu và thiết lập thông tin kết nối cho chúng.

1. Mở Visual Studio, click chọn File | New | Project.
2. Từ cửa sổ New Project, click vào mục Visual Studio Project và chọn Windows Application.
3. Nhập tên cho dự án và click OK. Trong ví dụ này chúng ta đặt tên dự án là VB\_SecureDatasource. Một dự án mới sẽ xuất hiện cùng cửa sổ Form trống.
4. Đầu tiên bạn cần chỉ định báo biểu chúng ta muốn làm việc. Để thêm một báo biểu vào trong dự án, chọn Project | Add Existing Item và trở đến file báo biểu cần sử dụng. Trong ví dụ này chúng ta sử dụng

báo biểu mẫu SecureReport.rpt chứa trong thư mục Projects trên đĩa CD ROM.

5. Từ cửa sổ công cụ ToolBox, kéo đối tượng ReportDocument vào Form – Visual Studio sẽ mở một hộp thoại với danh sách các báo biểu đang có trong dự án. Hãy chọn báo biểu chúng ta vừa thêm vào.
6. Cũng từ cửa sổ công cụ ToolBox, kéo đối tượng CrystalReportViewer vào trong Form và đặt thuộc tính Docking là Full Screen.
7. Tiếp theo, double click vào Form để viết mã cho sự kiện Form Load của Form liên kết đến cơ sở dữ liệu Northwind chuẩn của SQL Server như sau:

```
Dim SecureReport As SecureReport
```

```
Dim tables As  
    CrystalDecisions.CrystalReports.Engine.Table
```

```
Dim table As  
    CrystalDecisions.CrystalReports.Engine.Table
```

```
Dim tableLogonInfo As  
    CrystalDecisions.Shared.TableLogOnInfo
```

```
Dim LogonInfos As  
    CrystalDecisions.Shared.TableLogOnInfo
```

```
Dim LogonInfo As  
    CrystalDecisions.Shared.TableLogOnInfo
```

```
Dim ConnectionInfo As  
    CrystalDecisions.Shared.ConnectionInfo =
```

```
New  
    CrystalDecisions.Shared.ConnectionInfo
```

```
ConnectionInfo.ServerName="(local)"  
ConnectionInfo.DatabaseName="Northwind"  
ConnectionInfo.UserID="Sa"  
ConnectionInfo.Password="Admin"  
table =SecureReport.Database.Table
```

```
For each table in tables  
    tableLogonInfo=table.LogOnInfo  
    tableLogonInfo.ConnectionInfo=ConnectionInfo  
    table.ApplyLogonInfo(tableLogonInfo)
```

Next

```
CrystalReportViewer1.ReportSource=SecureReport
```

8. Sau khi hoàn thành, click Debug | Start hoặc nhấn phím F5 để bắt đầu thực thi ứng dụng. Cơ sở dữ liệu sẽ được kết nối, thông tin đăng nhập sẽ được gán cho mỗi bảng. Báo biểu sẽ thực thi mà không bị lỗi đang nhập như trước đó.

## 4. TÓM TẮT

Chương này đã hướng dẫn bạn cách tích hợp báo biểu vào các ứng dụng .NET của môi trường lập trình ứng dụng Visual Studio. Bạn đã biết cách sử dụng hai mô hình ReportViewer và ReportDocument cùng với kỹ thuật chuyển tham số bằng mã lệnh chương trình cho hai mô hình này. Chương tiếp theo chúng ta sẽ học các tích hợp báo biểu Crystal Reports vào một loại ứng dụng khá phổ biến trong thế giới Internet ngày nay, đó là ứng dụng Web.



## Chương 10:

# TÍCH HỢP BÁO BIỂU VÀO ỨNG DỤNG WEB

- ❖ *Tổng quan.*
- ❖ *Tiếp cận ứng dụng Web.*
- ❖ *Làm việc với Report View Object Model.*
- ❖ *Tóm tắt.*

## 1. TỔNG QUAN

Khi Internet trở nên thông dụng, mọi người nhận ra ngay đây là một công cụ mạnh kết nối con người với thông tin; các nhà phát triển ứng dụng lại nhìn thấy một khả năng dùng trình duyệt Web (Web Browse) để chạy ứng dụng ở mọi nơi mà không cần cài đặt bất cứ thứ gì lên máy cục bộ.

Khi giới thiệu công nghệ Active Server Pages (ASP), Microsoft đã tạo nên một công cụ thiết kế Web cho các nhà phát triển, giúp họ có thể xây dựng các trang Web động thay cho mã HTML tĩnh. Năm 2002, Microsoft xây dựng phát triển nền .NET Framework cùng phiên bản ASP.NET (bản nâng cấp của ASP). Phiên bản này chạy trong môi trường CLR (Common Language Runtime) của .NET và hỗ trợ rất nhiều ngôn ngữ lập trình. Là một đối tác thân thiết của Microsoft, Crystal Reports đã được tích hợp vào Visual Studio .NET từ phiên bản 2002 trở đi, giờ đây báo biểu có thể được xem và tương tác từ ứng dụng Web ASP.NET. Chương này sẽ hướng dẫn bạn cách tích hợp Crystal Reports vào ứng dụng ASP.NET được tạo và sử dụng bởi Visual Studio 2005 và .NET Framework.

Tất cả các ví dụ ứng dụng và báo biểu mẫu của chương này đều nằm trên đĩa CD ROM kèm theo giáo trình.

## 2. TIẾP CẬN ỨNG DỤNG WEB

Giống như với các ứng dụng Windows, đã được trình bày trong chương 9, bạn có thể dùng hai mô hình đối tượng sau đây để tích hợp báo biểu vào trong ứng dụng Web ASP.NET của mình. Mô hình đối tượng thứ

nhất là CrystalReportViewer, đối tượng này sẽ cung cấp tất cả các thuộc tính, phương thức và sự kiện mà bạn cần khi xem báo biểu. ReportViewer trên Web có thể gần giống với ReportViewer của các ứng dụng Windows, tuy một số thuộc tính hay phương thức của đối tượng có thể khác nhau, nhưng về mặt ý nghĩa thì chúng hoàn toàn giống nhau và thực hiện cùng một mục đích: cho phép xem báo biểu từ các ứng dụng của bạn và cho phép thay đổi các kết xuất báo biểu. Để có thể xử lý báo biểu về mọi mặt và toàn quyền điều khiển báo biểu, bạn cần sử dụng mô hình đối tượng thứ hai là ReportDocument.

**Ghi chú:** Mô hình đối tượng ReportDocument trên Web này giống với mô hình đối tượng ReportDocument mà chúng ta đã học trong chương trước. Bạn sẽ thấy một số nét giống nhau giữa chúng ở cuối chương nhưng đa phần là rất nhiều thuộc tính khác nhau. Hầu hết sự khác nhau nếu có là do ứng dụng Windows và ứng dụng Web có cách xử lý khác biệt và đó là lý do Crystal sử dụng hai mô hình đối tượng ReportDocument cho cùng một mục đích.

Sử dụng mô hình đối tượng nào là tùy vào kiểu ứng dụng Web mà bạn phát triển. Trong chương 8, chúng ta đã xem qua các kiểu tích hợp mà bạn có thể sử dụng trong ứng dụng của mình như xem và hiển thị nội dung báo biểu, truyền biến tham số cho báo biểu. Đối với các ứng dụng Web đơn giản, mô hình đối tượng CrystalReportView là sự lựa chọn tốt nhất, do nó cung cấp tất cả các lựa chọn cơ bản cần thiết để bạn có thể xem (Preview), thiết lập các mẫu tin sẽ được lọc bên trong ứng dụng của mình. Bạn có thể xem hình dạng của báo biểu trên trang Web Form như hình 10.1.



Hình 10.1: ReportViewer của Crystal Reports trên Web.

Chú ý rằng, đối với những người đã quen sử dụng các ứng dụng Windows, khi xem báo biểu trên Web sẽ có một chút khác biệt nhỏ. Thanh công cụ Viewer phía trên và phía dưới màn hình không thể neo được trên cửa sổ Web Form. Tuy nhiên, Viewer này có thể chia sẻ các chức năng không khác gì Windows ReportViewer, kể cả việc truyền biến tham số cho Viewer cũng như đăng ký login vào CSDL theo cách thông thường của ứng dụng Windows.

Nếu có thêm các nhu cầu tích hợp khác, như kết nối dữ liệu thông qua ADO hoặc ADO .NET, in ra máy in hoặc kết xuất ra các định dạng file chuẩn, điều khiển công thức tính của các trường (Field) hay khuôn dạng báo biểu, bạn cần sử dụng đến mô hình đối tượng ReportDocument.

Trong chương này chúng ta sẽ xem xét kiểu đối tượng CrystalReportsViewer thông qua việc chạy và thử các chức năng có sẵn. Phần cuối chương, chúng ta sẽ tìm hiểu cách sử dụng đối tượng ReportDocument.

Bạn không thể dùng lẫn lộn hai đối tượng này, nếu sau khi xem qua đối tượng CrystalReportViewer và nhận thấy chưa đáp ứng được nhu cầu, bạn có thể phải chuyển sang sử dụng đối tượng ReportDocument còn đối tượng CrystalReportViewer chỉ dùng để xem kết quả của báo biểu.

## **3. LÀM VIỆC VỚI CRYSTALREPORTVIEWER**

ReportViewer là điều khiển (Control) của .NET. Bạn có thể nắm kéo và thả nó vào trang Web Form, sau đó viết mã thực thi cho các sự kiện của Control. Các đoạn mã này sẽ điều khiển và qui định cách thức thể hiện giao diện của ReportViewer.

### **3.1. Xem báo biểu trên Web**

Để xem báo biểu trong ứng dụng ASP.NET, bạn chỉ cần viết một dòng mã. Kéo CrystalReportViewer vào Web Form của bạn và thiết lập thuộc tính ReportSource trỏ đến file báo biểu (nếu đã thiết lập thuộc tính này trong cửa sổ Properties thì bạn cũng chẳng phải viết thêm dòng mã nào nữa cả).

Sau khi thêm CrystalReportViewer vào Web Form và thiết lập thuộc tính ReportSource, bạn biên dịch và chạy ứng dụng Web Form của mình trong trình duyệt. ReportViewer sẽ hiển thị nội dung báo biểu ngay bên trong trình duyệt Web, hãy xem chúng ta thực hiện điều này như thế nào:

1. Từ Start menu chọn và mở Visual Studio 2005.
2. Nhấn chọn File | New | New Web Site.



Hình 10.2: Báo biểu hiển thị trên Web.

3. Nhập vào tên dự án và chọn nơi cất giữ dự án, sau đó chọn OK. Ví dụ: dự án này được đặt tên là VB\_WebViewReport.
4. Một dự án Web mới sẽ được tạo ra với một Web Form trống. Bạn chọn View | Toolbox, nắm kéo CrystalReportViewer vào trang Web Form.
5. Double-click vào Form để vào phần viết mã cho trang, bạn thêm mã vào thủ tục Form\_Load nạp file báo biểu như sau:
 

```
CrystalReportViewer1.ReportSource
      = "d:\Projects\websales.rpt"
```
6. Cuối cùng, chọn Debug > Start hoặc nhấn phím F5 để bắt đầu chạy ứng dụng. Cửa sổ trình duyệt sẽ được mở ra để thực thi Web Form và Viewer mà bạn thêm vào (hình 10.2).

Theo mặc định, Viewer hiển thị thanh công cụ ở phía trên trình Preview. Thanh công cụ này bao gồm một số nút như sau:

- Export: Dùng để xuất báo biểu ra các định dạng World, Excel, PDF thông qua một hộp thoại xuất.
- Print: Dùng mở hộp thoại in.
- Group Tree: Hiển thị hoặc ẩn nhóm cây thư mục bên cạnh trái của bản báo biểu.

- ◆ Page Navigation Button: Cung cấp nút điều hướng cho các trang báo biểu.
- ◆ Find: Dùng mở hộp thoại tìm kiếm cho phép nhập chuỗi để tìm trong báo biểu.
- ◆ Zoom: Chức năng phóng đại.

Ngoài ra bạn có thể tùy biến cách hiển thị của Viewer như chúng ta sẽ xem xét trong phần sau.

### 3.2. Thay đổi cách thức hiển thị của Viewer

Viewer cung cấp rất nhiều thuộc tính để bạn thay đổi diện mạo của nó. Nhóm thuộc tính đầu tiên xử lý các diện mạo bên ngoài của Viewer; còn nhóm thuộc tính thứ hai sẽ xử lý các biểu tượng Icon xuất hiện trên thanh công cụ Viewer. Các thiết lập ban đầu của thuộc tính được thể hiện trong bảng sau:

Phương thức	Mô tả	Kiểu
DisplayToolbar	Các Control hiển thị phía trên thanh công cụ của Viewer.	Boolean
DisplayBottomToolbar	Các Control hiển thị phía dưới thanh công cụ của Viewer.	Boolean
DisplayGroupTree	Control hiển thị Group Tree trên cạnh trái của Viewer.	Boolean
PageToTreeRation	Cách mà Control Group Tree sẽ xuất hiện trong trang báo biểu.	Integer

Nếu bạn không muốn hiển thị thanh công cụ của Viewer và điều chỉnh cây phân nhóm (Group Tree) nhỏ hơn, bạn có thể thêm đoạn mã sau vào trong Form\_Load:

```
CrystalReportViewer1.DisplayToolbar = False
CrystalReportViewer1.DisplayBottomToolbar = False
CrystalReportViewer1.PageToTreeRadio = 8
```

Nếu bạn sử dụng đoạn mã trên để tắt thanh công cụ và thay đổi tỉ lệ Ration, Viewer sẽ hiển thị báo biểu như hình 10.3.



Hình 10.3: Viewer báo biểu đã loại bỏ thanh công cụ với kích thước Group Tree tách rời.

Bằng cách điều chỉnh các biểu tượng trên thanh công cụ, bạn có thể điều chỉnh cách thức tương tác của người sử dụng với báo biểu. Ví dụ: Nếu bạn không muốn cho hiển thị nút in, người sử dụng sẽ không thể nào in báo biểu từ Viewer. Bạn có thể thay đổi các thuộc tính sau đây để làm xuất hiện hay giấu đi các nút của thanh công cụ:

- HasCrystalLogo.
- HasDrillUpButton.
- HasExportButton.
- HasPageNavigationButtons.
- HasPrint.Button.
- HasRefreshButton.
- HasSearchButton.
- HasToggleGroupTreeButton.
- HasViewList.
- HasZoomFactorList.

Tất cả những thuộc tính trên đều có kiểu Boolean, để có thể ẩn đi nút in ấn, bạn có thể viết mã như sau:

```
CrystalReportViewer1.HasPrintButton = False
```

Khi xem báo biểu, Viewer sẽ ẩn nút in và hiển thị như hình 10.4 sau đây.



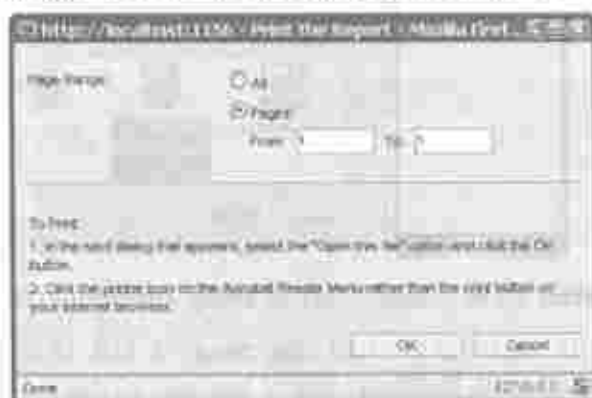
Hình 10.4: Viewer báo biểu đã loại bỏ nút in ấn.

Ngoài các thuộc tính thay đổi các nút nhấn trên thanh công cụ, bạn còn có thể sử dụng các thuộc tính để định dạng nội dung của khung Group Tree xuất hiện bên cạnh trái của cửa sổ Viewer. Group Tree giúp người sử dụng di chuyển qua các mục của báo biểu được dễ dàng hơn, nhưng cũng sẽ có đôi lúc bạn muốn thay đổi cách thể hiện của nó. Để làm điều này bạn có thể sử dụng thuộc tính GroupTreeStyle. Chẳng hạn như để thay đổi font hiển thị của Group Tree bạn viết lệnh sau:

```
CrystalReportViewer1.GroupTreeStyle.  
Font.Name = "Arial"
```

### 3.3. Chọn lựa chế độ in ấn

Một trong những tính năng mới trong Crystal Reports XI là có khả năng in trực tiếp báo biểu từ trang Web bằng Viewer. Tính năng in này có thể thực hiện theo 2 cách. Cách thứ nhất là sử dụng file PDF (Adobe® Acrobat®) như một file trung gian, với chức năng in ấn được cung cấp thông qua Adobe Acrobat Reader. Kỹ thuật này yêu cầu Adobe Acrobat Reader đã được cài đặt trên máy tính của người sử dụng từ trước. Khi người sử dụng nhấp vào nút in ấn, một hộp thoại sẽ hiển thị như hình 10.5 hướng dẫn bạn cách in báo biểu ra Acrobat Reader.



Hình 10.5: Hộp thoại in báo biểu.

Do hầu hết người sử dụng đều đã quen với file PDF, đây là phương thức in ấn mặc định và không đòi hỏi phải thêm bất kỳ phần mềm nào vào hoặc cài đặt các thành phần (component) khác.

Phương thức in ấn thứ hai là thông qua ActiveX control từ Internet Explorer, thành phần ActiveX này sẽ in báo biểu trực tiếp ra máy in của bạn. Khi thành phần này được kích hoạt và người sử dụng nhấp vào nút in trên Viewer của báo biểu, ActiveX control tự động tải về (download) và tự cài đặt vào môi trường trình duyệt. Một cửa sổ trung gian nhỏ giống như hình 10.6 hiển thị và hộp thoại của số máy in chuẩn được mở ra. Trong hộp thoại này người sử dụng có thể chọn lựa và định nghĩa máy in (chỉ xuất dành cho trình duyệt Internet Explorer).

Bạn có thể điều chỉnh cách thức in thông qua thuộc tính **PrintMode** của Viewer. Chúng ta có thể sử dụng 2 chế độ in sau đây:

- CrystalDecisions.Web.PrintMode.Pdf.
- CrystalDecisions.Web.PrintMode.ActiveX.

Để thiết lập chế độ in với ActiveX và in trực tiếp ra máy in, bạn viết mã như sau:

```
CrystalReportViewer1.PrintMode =
CrystalDecisions.Web.PrintMode.ActiveX
```



Hình 10.6: ActiveX máy in trong khi thực hiện.

Nếu bạn là người phát triển ứng dụng và biết chắc ứng dụng của mình chạy trên Internet Explorer, khi đó bạn nên sử dụng ActiveX control.

### 3.4. Hiệu chỉnh việc lọc các mẫu tin báo biểu

Record Selection là quá trình xử lý của Crystal Reports dùng để lọc các mẫu tin báo biểu. Công việc này được thực hiện thông qua đoạn mã sau đây:



```
{Customer.Country} = "USA"
```

Công thức Record-Selection được viết bởi cú pháp Crystal và có thể tạo ra trong quá trình thiết kế báo biểu hoặc ngay thời điểm xem báo biểu. Bạn có thể xem công thức Record Selection hiện hành đang sử dụng cho báo biểu bằng cách mở báo biểu trong Crystal Reports và chọn **Report > Selection Formulas > Record** để mở trình biên soạn công thức lọc Record Selection.

Bất kỳ công thức lọc Record Selection nào mà bạn xây dựng trong quá trình biên soạn này cũng được lưu vào trong báo biểu. Khi tích hợp các báo biểu vào ứng dụng, bạn có thể sử dụng hai thuộc tính để thiết lập Record Selection khi báo biểu đang chạy. Thuộc tính đầu tiên là SelectionFormula (công thức chọn lọc), có thể được sử dụng để lấy hoặc thiết lập công thức lọc mẫu tin. Thuộc tính thứ hai là ViewTimeSelectionFormula, được sử dụng để gắn thêm công thức lọc vào công thức lọc Record Selection đã tồn tại trong báo biểu.

Ví dụ: Nếu báo biểu đã tồn tại công thức Record Selection sau:

```
{Customer.Country} = "USA"
```

và bạn đã sử dụng thuộc tính ViewTimeSelectionFormula để thiết lập công thức Record Selection để lọc thêm một thành phố khác như sau:

```
{Customer.Country} = "NSW"
```

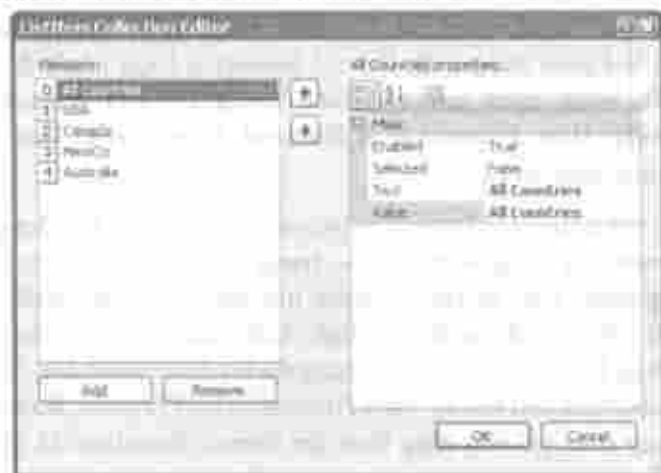
kết quả công thức Record Selection sẽ trở thành:

```
{Customer.Country} = "USA"  
And {Customer.Country} = "NSW"
```

Chúng ta hãy xem cách ứng dụng ViewTimeSelectionFormula như sau:

1. Từ Start Menu mở Visual Studio 2005 hay phiên bản cao hơn.
2. Chọn **File > New > Web Site**.
3. Nhập tên và chọn vị trí để lưu dự án (Project) của bạn vào và nhấp OK. Trong ví dụ này, chúng ta đặt tên cho dự án là VB\_WebRecordSelection.
4. Một dự án mới sẽ được tạo cùng với trang Web form trống có tên là WebForm1.aspx. Chọn **View > Toolbox** và đặt vào Web Form đối tượng ComboBox cùng với nút nhấn Button.

5. Bước kế tiếp, hiệu chỉnh thuộc tính *Items* của *ComboBox* như hình 10.7. Hãy nhập các mục sau vào hộp thoại:



Hình 10.7: Trình soạn thảo danh sách.

- All Countries.
  - USA.
  - Australia.
  - Canada.
  - Mexico.
6. Thuộc tính *Text* và *Value* của mỗi mục chọn là giống nhau.
7. Phần tiếp theo, thay đổi thuộc tính *Text* của nút *Button* thành *View Report*. *Form* của bạn bây giờ sẽ trông giống như hình minh họa 10.8.



Hình 10.8: Hoàn thành *Form*.

8. Từ *Form* này chúng ta sẽ cho phép người dùng chọn lựa giá trị từ danh sách số xuống *ComboBox* và sau đó click vào nút *View Report* để di chuyển qua bước kế tiếp của *Form*, nơi mà giá trị này sẽ được sử dụng để tạo và thiết lập công thức lọc cho báo biểu. Double-click nút nhấn *View Report* và nhập vào đoạn mã sau:

```
Server.Transfer("WebForm2.aspx", True)
```

9. Tiếp theo, chúng ta cần xây dựng Form thứ hai, Form này sẽ thiết lập công thức lọc mẫu tin, tính toán và sẽ hiển thị báo biểu. Để xây dựng Form này, click **Project > Add Web Form** và sử dụng luôn tên form chuẩn là **WebForm2.aspx**, click **OK**.
10. Double-click Form này trong cửa sổ Solution Explorer để mở cửa sổ thiết kế và kéo 2 Field nhãn (label) từ thanh công cụ Toolbar vào phần phía trên của Form.
11. Tìm điều khiển CrystalReportViewer và kéo nó lên phần giữa của Form.
12. Khi đã đặt Viewer vào đúng vị trí, double-click vào Form để viết mã cho sự kiện Form\_Load như sau:

```
Dim RecordSelectFormula as String
Dim PassedValue as String
PassedValue = Request.Form("DropDownList1")
RecordSelectFormula = "{Customer.Country} = '" &
                        PassedValue & "'"
Label1.Text = "You selected to run the report for " &
              PassedValue
Label2.Text = "The Resulting Record Selection Formula is " &
              RecordSelectFormula
CrystalReportViewer1.ReportSource =
              "d:\Projects\dynselect.rpt"
CrystalReportViewer1.SelectionFormula =
              RecordSelectFormula
```

13. Hiện giờ tất cả những gì còn lại là chạy ứng dụng và xem việc lọc báo biểu của bạn diễn ra như thế nào. Click **Debug > Start** hoặc nhấn phím **F5** để chạy chương trình ứng dụng của bạn. Cửa sổ trình duyệt sẽ hiển thị và ứng dụng Web của bạn sẽ trông như hình minh họa 10.9.
14. Chọn giá trị từ hộp danh sách xổ xuống và click nút nhấn **View Report** để tiến hành thực thi qua trang thứ hai, hiển thị nội dung báo biểu đã áp dụng công thức Record Selection.

Đây chỉ là một ví dụ đơn giản về cách mà mẫu tin được chọn lọc bằng Record Selection. Bạn cần tạo một công thức lọc mẫu tin phức tạp hơn theo nhu cầu của riêng mình.



Hình 10.9: Hoàn tất ứng dụng đơn giản.

### 3.5. Làm việc với Field tham số

Các Field tham số luôn là vấn đề khi vận hành trong các phiên bản Crystal Reports trước đây. Các hộp thoại mặc định yêu cầu nhập giá trị biến rất khó chịu và không dễ sử dụng. Hầu hết các nhà phát triển ứng dụng đều thích tạo lại các hộp thoại nhập giá trị biến tham số hơn là sử dụng hộp thoại mặc định do Crystal cung cấp. Với Crystal Reports XI, những hộp thoại này đã có thêm nhiều cách chọn lựa hơn và có thể sử dụng cả hai loại biến (Parameter) động và phân tầng (Cascading). Bạn có thể đọc giá trị từ một nguồn dữ liệu và sau đó chọn giá trị trả về cho biến tham số trong báo biểu của mình.

Các Field tham số mở rộng cùng hộp thoại nhập tham số bằng giao diện Web (như hình minh họa 10.10) có thể giúp cho các nhà phát triển tạo các báo biểu đơn giản nhanh chóng.



Hình 10.10: Hộp thoại biến tham số mặc định Crystal Reports trên Web.

Nếu muốn kiểm soát hộp thoại tùy biến linh động hơn, hoặc nếu bạn muốn tạo những biến tham số của riêng mình, bạn cần truyền các

giá trị biến tham số cho Viewer và báo biểu thông qua sử dụng lớp `ParameterField`. Lớp này chứa các thuộc tính cho phép đọc và đặt các giá trị thông tin cho biến tham số khi báo biểu thực thi. Ngoài ra, lưu ý Crystal còn cung cấp thêm lớp `ParameterFields` (chú ý có s), là tập hợp lưu giữ tất cả các Field tham số trong báo biểu của bạn.

Để làm việc với các Field tham số, bạn cần phải học cách thức đọc giá trị thông tin từ các Field tham số trong báo biểu của mình.

### 3.6. Đọc và thiết lập các giá trị rời rạc cho biến tham số

Khi xây dựng các Field tham số trong Crystal Reports, bạn có thể tạo các giá trị chứa trong danh sách tính cho phép người sử dụng chọn lựa khi báo biểu thực thi. Danh sách này cũng có thể chứa giá trị được chọn mặc định mà báo biểu sử dụng.

Để đọc các giá trị biến tham số mặc định của báo biểu, bạn sử dụng thuộc tính `DefaultValues`. Trong ví dụ sau, báo biểu của chúng ta chỉ có một biến tham số, trước hết chúng ta sẽ đọc và gán giá trị cho biến tham số này.

1. Mở Visual Studio 2005.
2. Click **File > New > Web Site**.
3. Nhập vào tên dự án, chọn vị trí lưu dự án của bạn và click **OK**. Trong ví dụ này, chúng ta đặt tên cho dự án là `VB_WebRecordSelection`.
4. Dự án mới sẽ được tạo với một Web Form trống mang tên `WebForm1.aspx`. Click chọn **View > Toolbox** và kéo vào Form một đối tượng nhãn `Label`, cùng một danh sách xổ xuống `Dropdown` và nút nhấn `Button`.
5. Double-click vào Form bạn khai báo biến ngay phía bên dưới định nghĩa lớp **`Inherits System.Web.UI.Page`** như sau:

```
Dim ParameterFields As  
    CrystalDecisions.Shared.ParameterFields  
Dim ParameterField As  
    CrystalDecisions.Shared.ParameterField  
Dim ParameterDiscreteValue As  
    CrystalDecisions.Shared.ParameterDiscreteValue
```

6. Trong thủ tục xử lý sự kiện `Form_Load`, bạn cài đặt cách hiển thị báo biểu và danh sách tham số mặc định như sau:

```

CrystalReportViewer1.ReportSource =
    "D:\projects\DYNPARAM.rpt"
CrystalReportViewer1.DisplayGroupTree = False
ParameterFields =
    CrystalReportViewer1.ParameterFieldInfo
ParameterField = ParameterFields(0)
Label1.Text = ParameterField.PromptText

Dim i As Integer
For i = 0 To (ParameterField.DefaultValues.Count - 1)
    DropDownList1.Items.Add(
        ParameterField.DefaultValues.Item(i).
            Description)
Next
ParameterDiscreteValue = New
CrystalDecisions.Shared.ParameterDiscreteValue
ParameterDiscreteValue.Value = " "
ParameterField.CurrentValues.Add(
    ParameterDiscreteValue)

```

7. Tiếp đến, chuyển trở về cửa sổ Design của Web Form và đổi tên thuộc tính Text trên nút nhấn thành **View Report**.
8. Double-click vào nút nhấn View Report và nhập vào đoạn mã sau:

```
Server.Transfer("WebForm2.aspx", True)
```

9. Từ cửa sổ Project Explorer, Double-click vào tập in WebForm2.aspx và thêm vào 2 nhãn (label) từ cửa sổ công cụ Toolbox giống như thêm Crystal Reports Viewer control vào vậy.
10. Double-click vào Form mới để viết mã, trước hết bạn khai báo biến ngay bên dưới dòng **Inherits System.Web.UI.Page** như sau:

```

Dim ParameterFields As
    CrystalDecisions.Shared.ParameterFields
Dim ParameterField As
    CrystalDecisions.Shared.ParameterField
Dim ParameterDiscreteValue As
    CrystalDecisions.Shared.ParameterDiscreteValue

```

11. Cài đặt sự kiện Form Load cho Form như sau:

```

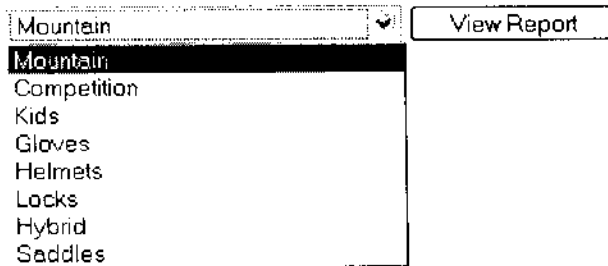
CrystalReportViewer1.ReportSource =
    "d:\projects\DYNPARAM.rpt"
Dim PassedValue

```

```
PassedValue = Request.Form("DropDownList1")
ParameterFields =
    CrystalReportViewer1.ParameterFieldInfo
ParameterField = ParameterFields(0)
ParameterDiscreteValue = New
    CrystalDecisions.Shared.ParameterDiscreteValue
ParameterDiscreteValue.Value = PassedValue
ParameterField.CurrentValues.Add(ParameterDiscreteValue)
```

12. Click vào **Debug > Start** hoặc nhấn phím **F5** để bắt đầu chạy ứng dụng của bạn. Hộp ComboBox ở phía trên cùng Form sẽ đưa các giá trị mặc định vào trong hộp DropDown của báo biểu như hình minh họa 10.11.

Enter ProductType:

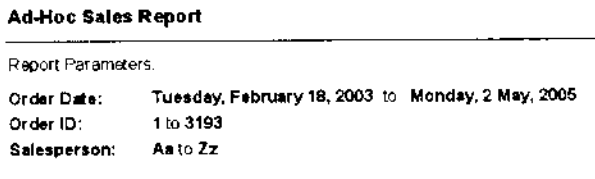


**Hình 10.11:** Form với giá trị mặc định chứa trong danh sách xổ xuống.

13. Chọn một giá trị bất kỳ và click vào nút View Report. Báo biểu sẽ được lọc lại theo giá trị biến tham số mà bạn vừa chọn

### 3.6.1. Đặt giá trị phạm vi cho biến tham số

Bạn có thể thiết lập biến tham số cho phép nhập một dãy các giá trị trong khoảng nào đó giống như báo biểu được minh họa trong hình 10.12.



**Hình 10.12:** Một kiểu báo biểu với biến phạm vi.

Biến tham số này có thể nhận cả hai giá trị: giá trị bắt đầu và giá trị kết thúc.

Đoạn mã sau đây sẽ minh họa cách sử dụng biến tham số truyền theo dạng mảng:

```
CrystalReportViewer1.ReportSource =
    "d:\projects\RANGEPARAM.rpt"
ParameterFields =
    CrystalReportViewer1.ParameterFieldInfo
ParameterField = ParameterFields(0)
ParameterRangeValue = New
    CrystalDecisions.Shared.ParameterRangeValue
ParameterRangeValue.StartValue = "12/31/2004"
ParameterRangeValue.EndValue = "01/01/2005"

ParameterField.CurrentValues.Add(ParameterRangeValue)
```

### 3.7. Cách thức đăng nhập CSDL

Để đăng nhập đến nguồn dữ liệu được đảm bảo hơn từ báo biểu của ứng dụng Web, bạn cần tạo đối tượng LogonInfo và sau đó chỉ rõ tên bảng dữ liệu, tên người sử dụng (user name) và mật khẩu (password). Crystal Reports phiên bản XI này cho phép bạn xử lý đăng nhập bằng mã lệnh chương trình đối với mỗi bảng dữ liệu bạn muốn truy xuất:

```
Dim LogonInfo As CrystalDecisions.Shared.TableLogOnInfo
CrystalReportViewer1.LogOnInfo = New
    CrystalDecisions.Shared.TableLogOnInfos
LogonInfo = New CrystalDecisions.Shared.TableLogOnInfo
LogonInfo.TableName = "Products"
LogonInfo.ConnectionInfo.UserID = "Admin"
LogonInfo.ConnectionInfo.Password = ""
CrystalReportViewer1.LogOnInfo.Add(LogonInfo)
```

Mô hình đối tượng ReportDocument thường dùng cơ chế này để duyệt qua tất cả các bảng dữ liệu trong báo biểu.

## 4. TÓM TẮT

ASP.NET là nền tảng phổ biến dùng tạo nên các ứng dụng Web mạnh mẽ và giờ đây Crystal Reports đã trở thành một phần vận hành công việc trình bày báo biểu và kết xuất từ các ứng dụng Web. ASP.NET là công nghệ của Microsoft dành cho nền Windows. Nếu bạn sử dụng công nghệ Java thì chương sau sẽ hướng dẫn bạn cách tích hợp Crystal Reports vào trong ứng dụng JSP (Java Server Pages).



## Chương 11:

# TÍCH HỢP BÁO BIỂU VÀO ỨNG DỤNG JAVA

- ❖ *Tổng quan.*
- ❖ *Tạo báo biểu sử dụng với ứng dụng JSP.*
- ❖ *Làm việc với các thành phần báo biểu Java.*
- ❖ *Làm việc với các Field tham số.*
- ❖ *Tóm tắt.*

## 1. TỔNG QUAN

Crystal Reports đã trở nên quen thuộc với các nhà phát triển ứng dụng theo dòng sản phẩm Microsoft, nhưng nó ít được các chuyên gia ứng dụng Java biết đến. Đó là do Crystal Reports thường kết hợp với các sản phẩm của bộ công cụ Visual Studio. Tuy nhiên, trong vài năm trở lại đây, BusinessObjects đã giúp một số nhà cung cấp công cụ tích hợp phiên bản OEM của Crystal Reports với thành phần Java Report (Java Report Component) vào bộ công cụ phát triển Java như: BEA Weblogic Workshop, Borland® JBuilder, IBM Rational Application Developer, IBM WebSphere® Studio.

Việc mở rộng phạm vi này đã khiến Crystal Reports có khả năng hỗ trợ 100% mã Java. JRC (Java Report Component) được giới thiệu từ phiên bản Crystal Reports 10 đã tạo cơ hội cho các chuyên viên phát triển Java sử dụng Crystal Reports nhiều hơn trong ứng dụng của họ. JRC cung cấp các công cụ thiết lập mạnh mẽ cho việc tích hợp báo biểu Report vào ứng dụng của bạn với khả năng viết mã ít nhất.

**Ghi chú:** Do JRC thuần Java, nó có thể sử dụng với các nền phát triển Java và các Server ứng dụng như WebLogic, WebSphere và Tomcat.

Trong chương này chúng ta sẽ xem xét đến việc sử dụng JRC để thêm báo biểu Report vào ứng dụng JSP.

## 2. TẠO BÁO BIỂU SỬ DỤNG VỚI ỨNG DỤNG JSP

Phần sau sẽ hướng dẫn bạn kết nối nguồn dữ liệu để tạo ra báo biểu trong các ứng dụng JSP.

### 2.1. Tìm hiểu JDBC

Khi làm việc với Crystal Reports và Java, có hai cách khác nhau mà bạn có thể sử dụng để xuất dữ liệu trong báo biểu. Cách thứ nhất là sử dụng trình JDBC™ để kết nối trực tiếp vào các nguồn dữ liệu. Cách thứ hai là dùng cơ chế cầu nối ODBC-JDBC.

Đối với cả hai cách này, trước tiên bạn cài đặt và cấu hình JDBC driver theo yêu cầu của nhà sản xuất đưa ra. Tiếp đến, bạn cần phải cập nhật file CRConfig.XML, nằm trong thư mục C:\Program Files\Common Files\Business Objects\3.0\Java. Trong file XML này có nhiều cấu hình thiết lập khác nhau và bạn có thể tham khảo trong bảng dưới đây:

Nội dung	Mô tả	Ví dụ
JavaDir	Đường dẫn đến thư viện thực thi Java.	C:\Program Files\Java\jre1.5.0_02\bin
ClassPath	Đường dẫn trỏ đến file JAR chứa trình JDBC Driver.	
JDBCURL	Chuỗi URL mà JDBC driver sử dụng.	Jdbc:oracle:oci8:scott/tiger@myhost
JDBCUserName	Tên đăng nhập.	Scott
JDBCClassName	Lớp JDBC driver.	com.oracle.jdbc.oracle.oci8

Dưới đây là nội dung của file CRConfig.XML

```
<?xml version="1.0" encoding="utf-8"?>
<CrystalReportEngine-configuration>
  <reportlocation>../..</reportlocation>
  <timeout>10</timeout>
```

```

<ExternalFunctionLibraryClassNames>
    <classname> </classname>
    <classname> </classname>
</ExternalFunctionLibraryClassNames>

<keycode>XXXXXX-XXXXX-XXXX-XXXX</keycode>
<Javasever-configuration>
<DataDriverCommon>
<JavaDir>C:\Program Files\Java\jre1.5.0_02\bin</JavaDir>
  <Classpath>C:\Program Files\Common Files\Business
  Objects\3.0\java\lib\crlovmanifest.jar;C:\Program Files\Common
  Files\Business Objects\3.0\java\lib\CRLOVEexternal.jar;C:\Program
  Files\Common Files\Business
  Objects\3.0\java\lib\CRDBJavaServerCommon.jar;C:\Program
  Files\Common Files\Business
  Objects\3.0\java\lib\CRDBJavaServer.jar;C:\Program
  Files\Common Files\Business
  Objects\3.0\java\lib\CRDBJDBCServer.jar;C:\Program
  Files\Common Files\Business
  Objects\3.0\java\lib\CRDBXMLServer.jar;C:\Program
  Files\Common Files\Business
  Objects\3.0\java\lib\CRDBJavaBeansServer.jar;C:\Program
  Files\Common Files\Business
  Objects\3.0\java\lib\external\CRDBXMLExternal.jar;C:\Program
  Files\Common Files\Business
  Objects\3.0\java\lib\external\log4j.jar;C:\Program Files\Common
  Files\Business Objects\3.0\java\lib\cecore.jar;C:\Program
  Files\Common Files\Business
  Objects\3.0\java\lib\celib.jar;C:\Program Files\Commcn
  Files\Business Objects\3.0\java\lib\ebus405.jar;C:\Program
  Files\Common Files\Business
  Objects\3.0\java\lib\corbaidl.jar;C:\Program Files\Common
  Files\Business
  Objects\3.0\java\lib\external\freessl201.jar;C:\Program
  Files\Common Files\Business
  Objects\3.0\java\lib\external\asn1.jar;C:\Program Files\Common
  Files\Business Objects\3.0\java\lib\external\certj.jar;C:\Program
  Files\Common Files\Business
  Objects\3.0\java\lib\external\jsafe.jar;C:\Program Files\Common
  Files\Business Objects\3.0\java\lib\external\sslj.jar; C:\Program
  
```

```
Files\Microsoft SQL Server 2000 Driver for JDBC\lib\msbase.jar;
C:\Program Files\Microsoft SQL Server 2000 Driver for
JDBC\lib\mssqlserver.jar; C:\Program Files\Microsoft SQL Server
2000 Driver for JDBC\lib\msutil.jar;${CLASSPATH}</Classpath>
  <IORFileLocation>${TEMP}</IORFileLocation>
  <JavaServerTimeout>1800</JavaServerTimeout>
  <JVMMaxHeap>64000000</JVMMaxHeap>
  <JVMMinHeap>32000000</JVMMinHeap>
  <NumberOfThreads>100</NumberOfThreads>
</DataDriverCommon>
```

<JDBC>

```
<CacheRowsetSize>100</CacheRowsetSize>
<JDBCURL>jdbc:microsoft:sqlserver://sqlserver01:1443
</JDBCURL>
  <JDBCClassName>
    com.microsoft.jdbc.sqlserver.SQLServerDriver
  </JDBCClassName>
<JDBCUserName>sa</JDBCUserName>
<JNDIURL></JNDIURL>
<JNDIConnectionFactory></JNDIConnectionFactory>
<JNDIInitContext>|</JNDIInitContext>
<JNDIUserName>weblogic</JNDIUserName>
<GenericJDBCdriver>
  <Option>No</Option>
  <DatabaseStructure>catalogs,tables</DatabaseStructure>
  <StoredProcType>Standard</StoredProcType>
  <LogonStyle>SQLServer</LogonStyle>
</GenericJDBCdriver>
```

</JDBC>

<XML>

```
<CacheRowsetSize>100</CacheRowsetSize>
<PreReadNBytes>4096</PreReadNBytes>
<XMLLocalURL></XMLLocalURL>
<SchemaLocalURL></SchemaLocalURL>
<XMLHttpURL></XMLHttpURL>
<SchemaHttpURL></SchemaHttpURL>
```

</XML>

```

<JavaBeans>
  <CacheRowSetSize>100</CacheRowSetSize>
  <JavaBeansClassPath></JavaBeansClassPath>
</JavaBeans>
</Jaserverconfiguration>
</CrystalReportEngineconfiguration>

```

Với cấu hình driver này, bước kế tiếp là chỉ định JDBC driver trong Crystal Reports. Để tìm hiểu cách vận hành, chúng ta sẽ tạo một báo biểu dựa trên nguồn dữ liệu JDBC. Trong phần này chúng ta dùng CSDL SQL Server của Microsoft. Để tạo báo biểu từ nguồn dữ liệu JDBC bạn thực hiện các bước sau:

1. Từ Menu Start mở Crystal Reports XI.
2. Nhấp File > New > Blank Report, cửa sổ Database Expert sẽ hiển thị.
3. Double-click vào mục **Create New Connection**.
4. Double-click vào mục **JDBC (JNDI)**. Hộp thoại tùy biến JDBC driver sẽ hiển thị như hình 11.1.



Hình 11.1: JDBC driver.

- Nhập đường dẫn kết nối vào ô **Connection URL** và lớp truy xuất CSDL vào **Database Classname**, sử dụng các giá trị giống như bạn đã nhập vào trong file **CRConfig.XML**, sau đó click **Next**.
- Sử dụng hộp thoại kết nối như hình 11.2 để nhập tên người dùng (**User Name**) và mật khẩu (**Password**) cho CSDL của bạn.



Hình 11.2: Sử dụng JDBC đang nhập CSDL.

- Khi hoàn tất, click vào nút **Finish** để kết nối đến nguồn dữ liệu của bạn. Một danh sách các bảng dữ liệu có sẵn trong CSDL sẽ xuất hiện bên dưới tên nguồn dữ liệu.

Chọn các bảng dữ liệu (table), các trường (Field) như cách thông thường mà bạn hay dùng để tạo Report trong môi trường Microsoft.

## 2.2. Làm việc với nguồn dữ liệu XML

Crystal Reports cung cấp trình điều khiển XML driver, cho phép bạn truy xuất dữ liệu lưu trong file XML tĩnh hoặc dữ liệu cung cấp từ các dịch vụ Web dạng XML. Để tạo một báo biểu từ file XML bạn thực hiện qua các bước sau:

1. Từ Menu Start mở Crystal Reports XI.
2. Nhấp File > New > Bank Report, cửa sổ Database Expert sẽ hiển thị.
3. Mở rộng mục Create New Connection và ở bên trong mục này mở rộng tiếp mục XML. Hộp thoại nguồn dữ liệu XML sẽ hiển thị như hình 11.3.



Hình 11.3: Driver dữ liệu XML.

4. Nhấp vào nút ">" trong hộp thoại để duyệt tìm file XML dữ liệu. Trong ví dụ này bạn có thể sử dụng file CUSTOMERS.XML nằm trên đĩa CD-ROM kèm giáo trình. Sau khi chọn file, click vào nút Finish để trở lại hộp thoại DataBase Expert.
5. File dữ liệu XML bây giờ sẽ xuất hiện bên dưới mục XML. Bạn có thể tạo báo biểu theo như cách tạo một báo biểu thông thường mà chúng ta vẫn làm (chọn trường, thêm nhóm, sắp xếp...).

Crystal Reports sẽ xem file XML như là một bảng dữ liệu và lấy ra kiểu dữ liệu thích hợp cho mỗi trường nó nhận diện được.

## 3. LÀM VIỆC VỚI CÁC THÀNH PHẦN BÁO BIỂU JAVA

JRC (Java Reporting Component) là thành phần giúp tích hợp các báo biểu vào trong các ứng dụng Java. Trong phần tiếp theo sau, chúng ta sẽ học cách cấu hình và sử dụng JRC để hiển thị nội dung báo biểu.

### 3.1. Cấu hình môi trường phát triển ứng dụng

Điều đầu tiên khi bắt đầu làm việc với JRC là cấu hình môi trường phát triển và ứng dụng. Bạn cần thực hiện qua các bước sau:

1. Chép thư mục CrystalReportViewer11 vào thư mục ứng dụng Web.
2. Chép các file thư viện JAR của JRC vào thư mục **WEB-INF\lib**.
3. Chép file **Crystal-tag-ReportViewer.tld** vào thư mục **WEB-INF\lib**.
4. Chép file CRConfig.XML vào thư mục **WEB-INF\classes**.
5. Sửa đổi file **Web.XML**.

#### **Bước 1: Kiểm tra cấu trúc của thư mục ứng dụng Web:**

Bảo đảm thư mục ứng dụng Web của bạn có thư mục con **WEB-INF** và trong thư mục **WEB-INF** này tạo 2 thư mục con **\lib** và **\classes**. Hầu hết các môi trường JDE đều tạo cấu trúc thư mục này tự động khi bạn tạo mới dự án (Project) JSP.

#### **Bước 2: Chép thư mục CrystalReportViewer11 vào thư mục ứng dụng Web:**

Thư mục này chứa tất cả file cần thiết cho trình xem báo biểu. Bạn có thể tìm thấy thư mục này theo đường dẫn **C:\Program Files\Business Objects\Common\3.0\CrystalReportViewers11**.

#### **Bước 3: Chép thư mục Java Report Component JAR vào thư mục WEB-INF\lib:**

Các file JAR cho JRC nằm trong **C:\Program Files\Business Objects\Common\3.0\java\lib**. Bạn cần chép tất cả các file này vào thư mục con **\lib** của ứng dụng Web. Kiểm tra rằng bạn đã chép tất cả file JAR có trong thư mục con để đảm bảo bạn có tất cả các thành phần của nhà cung cấp thứ ba mà JRE (Java Runtime Environment) cần.



**Bước 4: Chép Crystal-tags-ReportViewer.tld vào thư mục WEB-INF\lib của bạn:**

File này được yêu cầu mỗi khi bạn sử dụng trình xem tag (viewer tag) Crystal Reports. Bạn có thể tìm thấy file theo đường dẫn C:\Program Files\Business Objects\Common\3.0\java\lib>taglib. Để có thêm thông tin về cách sử dụng Report viewer tags, bạn xem thêm tài liệu Java SDK đi kèm theo với Crystal Reports XI.

**Bước 5: Chép file CRConfig.XML vào thư mục WEB-INF\classes:**

File này chứa cấu hình tùy biến cho Report engine, hãy chép nó vào thư mục con \classes.

**Bước 6: Sửa đổi file Web.xml của bạn:**

Bước cuối cùng là sửa đổi file Web.xml của ứng dụng. Bạn sẽ cần thêm vào đoạn cấu hình sau:

```
<context-param>
  <Param-name>crystal_image_uri</Param-name>
  <Param-value>crystalreportviewers11</Param-value>
</context-param>
```

Và nếu có dự định sử dụng thư viện Crystal Viewer tag, bạn cần thêm vào đoạn cấu hình sau:

```
<taglib>
  <taglib-url>
    /crystal-tags-reportviewer.tld
  </taglib-url>
  <taglib-location>
    /WEB-INF/crystal-tags-reportviewer.tld
  </taglib-location>
</taglib>
```

Chỉ khi nào đã hoàn tất những bước này, bạn mới có thể sử dụng tích hợp báo biểu với JRC vào ứng dụng JSP của mình.

### 3.2. Xem báo biểu trên trang Web

Để xem một báo biểu bằng JRC, bạn sử dụng lớp `CrystalReportViewer` do JRC cung cấp. Tương tự với thành phần `CrystalReportViewer` của Visual Studio, thành phần `CrystalReportViewer` của JRC cũng chứa tất cả các thuộc tính và phương thức dùng cho việc hiển thị báo biểu và điều khiển nội dung báo biểu.

Trước hết, để có thể xem Report từ trang JSP, bạn cần tạo một tham chiếu lớp ở đầu trang JSP như sau:

```
<%@page import=  
    "com.crystaldecisions.report.web.viewer.*"%>  
<%@page import ="com.crystaldecisions.report.sdk.*"%>
```

Sau đó tạo một đối tượng thuộc lớp `ReportClientDocument` và mở file báo biểu:

```
String MyReportName = "WEBSALES.rpt";  
ReportClientDocument myReportClientDocument =  
    new ReportClientDocument();  
myReportClientDocument.Open(MyReportName, 0);
```

Bước kế tiếp để xây dựng nội dung báo biểu là lấy về đối tượng `ReportSource` bằng đoạn mã sau:

```
Object MyReportSource;  
MyReportSource =
```

```
myReportClientDocument.getReportSource();
```

Tiếp đến là đặt nội dung báo biểu vào Viewer để hiển thị:

```
CrystalReportViewer myCrystalReportViewer =  
    new CrystalReportViewer();  
myCrystalReportViewer.SetReportSource(  
    MyReportSource);  
myCrystalReportViewer.setOwnPage(true);  
myCrystalReportViewer.setOwnForm(true);
```

Cuối cùng, chúng ta cần làm tươi báo biểu để đọc dữ liệu sau cùng nhất và dùng phương thức `processHttpRequest` để xử lý yêu cầu hiển thị báo biểu.

```
myCrystalReportViewer.refresh();  
myCrystalReportViewer.processHttpRequest(request,
```

```
response, getservletconfig().getServletContext(),
null);
```

Khi biên dịch và chạy trang ứng dụng, Report của bạn sẽ được hiển thị bằng trình duyệt HTML cung cấp bởi Crystal Reports XI và sẽ trông như hình 11.4.



Hình 11.4: Xem báo biểu trên HTML.

Nếu bạn không thể biên dịch và chạy ứng dụng, hãy kiểm tra lại các bước và xem Report của bạn nằm đúng vị trí đường dẫn hay chưa. Dưới đây là toàn bộ mã nguồn của trang JSP.

Viewreport1.JSP

```
<%@Page import="
    "com.crystaldecisions.report.web.viewer.*"%>
<%@Page import="com.crystaldecisions.report.sdk.*"%>
<%
String MyReportName = "WEBSALES.rpt";
ReportClientDocument myReportClientDocument = new
CrystalReportDocument();
myReportClientDocument.open(MyReportName, 0);

Object MyReportSource;
MyReportSource =
myReportClientDocument.getReportSource();

CrystalReportViewer myCrystalReportViewer = new
CrystalReportViewer();
myCrystalReportViewer.setReportSource(
    MyReportSource);
```

```

myCrystalReportViewer.SetOwnPage(true);
myCrystalReportViewer.SetOwnForm(true);

myCrystalreportViewer.refresh();
myCrystalreportViewer.processHttpRequest(request,
response, getServletConfig().getServletContext(), null);
%>

```

## 4. LÀM VIỆC VỚI CÁC FIELD THAM SỐ

Các Field tham số trong Crystal Reports có thể được dùng để hiển thị thông tin trên báo biểu, lọc nội dung... Bạn có thể dùng JRC để truyền tham số khi báo biểu thực thi bằng cách sử dụng lớp *ParameterField*.

Bước đầu tiên là tạo một tập hợp chứa các Field tham số bằng mã chương trình như sau:

```

// Tạo tập hợp .
Fields MyFields = new Fields();

// Tạo một giá trị biến mới và thiết lập nó là "USA"
String MyParameterValue = "USA"

```

Tiếp theo là tạo Field tham số và đưa nó vào tập hợp

```

// Tạo đối tượng Field tham số
ParameterField MyParameterField = new ParameterField();
MyParameterField.setReportName(MyReportName);
Values MyParameterValues = new Values();

// Tạo giá trị Field tham số
ParameterFieldDiscreteValue
    MyParameterFieldDiscreteValue =
        new ParameterFieldDiscreteValue();

// Thiết lập tên của biến và giá trị biến
MyParameterField.setName("?EnterCountry");
MyParameterFieldDiscreteValue.setValue(
    MyParameterValue);
MyParameterValues.add(MyParameterFieldDiscreteValue);
MyParameterField.setCurrentValues(MyParameterValues);
MyFields.add(MyParameterField);

```

Phần còn lại, khi thực thi báo biểu tham số sẽ lọc ra tất cả các mẫu tin có giá trị mã quốc gia là "USA".

## 5. ĐĂNG NHẬP TỰ ĐỘNG VÀO CSDL

Khi bạn tạo một báo biểu từ CSDL hoặc từ một nguồn dữ liệu khác, bạn phải nhập tên người dùng (username) và mật khẩu (password) để truy cập vào CSDL hoặc nguồn dữ liệu đó. Khi báo biểu chạy trong ứng dụng Web, để tránh người dùng phải nhập vào thông tin đăng nhập, bạn sử dụng đối tượng *ConnectionInfo* để cung cấp thông tin truy cập CSDL trước khi xem báo biểu. Trong ví dụ sau, tên tài khoản đăng nhập là "sa" và tên mật khẩu "admin" được truyền qua đối tượng *ConnectionInfo* sẽ khiến quá trình đăng nhập tự động, người sử dụng sẽ không còn bị nhắc nhở nhập giá trị kết nối này khi xem báo biểu.

```
// Thiết lập thông tin kết nối
ConnectionInfos MyConnectionInfos = new
    ConnectionInfos();
ConnectionInfo MyConnectionInfo = new ConnectionInfo();

// Thiết lập tên và mật khẩu CSDL
MyConnectionInfo.SetUserName("sa");
MyConnectionInfo.SetPassword("admin");
MyConnectionInfos.add(MyConnectionInfo);
```

Dưới đây là mã trang JSP thực hiện cơ chế kết nối tự động hiển thị thông tin báo biểu:

```
SETDB.JSP
<%@ Page import="com.crystaldecisions.report.sdk.*"%>
<%@ Page import=
    "com.crystaldecisions.sdk.occa.
        report.reportsource.*"%>
<%@ Page import=
    "com.crystaldecisions.sdk.occa.report.lib.*"%>
<%@ Page import=
    "com.crystaldecisions.sdk.occa.report.data.*"%>
<%

    // Mở tài liệu Report
    String MyReportName = "setdb.rpt"
    ReportClientDocument MyReportClientDocument = new
        ReportClientDocument();
    MyReportClientDocument.open(MyReportDocument);

    // Lấy Report
    Object MyReportSource;
    MyReportSource =
        MyReportClientDocument.getReportSource();
```

```
// Thiết lập thông tin kết nối
ConnectionInfos MyConnectionInfos = new
    ConnectionInfos();
ConnectionInfo MyConnectionInfo = new ConnectionInfo();

// Thiết lập tên và mật khẩu CSDL
MyConnectionInfo.setUsername("sa");
MyConnectionInfo.setPassword("admin");
MyConnectionInfos.add(MyConnectionInfo);

// Tạo viewer và thiết lập report source cho Report
CrystalReportViewer MyCrystalReportViewer = new
CrystalReportViewer();
MyCrystalreportViewer.setReportSource(MyReportSource);

// Thiết lập tùy biến viewer
MyCrystalReportViewer.setOwnpage(true);
MyCrystalReportViewer.setOwnForm(true);
MyCrystalReportViewer.setDatabaseLogonInfos(
    MyConnectionInfos);

// Làm tươi Report
MyCrystalreportViewer.refresh();
MyCrystalreportViewer.processHttpRequest(
    request, response,
getServletConfig().getServletContext(), null);
%>
```

## 6. TÓM TẮT

Java Server Pages (JSP) là công cụ phát triển ứng dụng Web mạnh mẽ. Giờ đây với sự hỗ trợ của Crystal Reports, bạn có thể nhúng mã xử lý báo biểu hoàn toàn bằng Java, tận dụng sức mạnh của Crystal vốn là lợi thế của các nhà phát triển trên nền Microsoft. Trong chương tiếp theo, chúng ta sẽ học cách tích hợp báo biểu vào ứng dụng cao cấp sử dụng BusinessObjects Enterprise XI.

## Chương 12:

# BUSINESSOBJECTS ENTERPRISE

- ❖ *Tổng quan.*
- ❖ *Tìm hiểu kiến trúc Report Server.*
- ❖ *Cài đặt BusinessObjects.*
- ❖ *Làm việc xử lý báo biểu với đường dẫn URL.*
- ❖ *Xem báo biểu.*
- ❖ *Tóm tắt.*

## 1. TỔNG QUAN

Trong một ứng dụng lớn có hàng ngàn báo biểu và truy xuất bởi nhiều người với các quyền truy cập khác nhau, bạn chắc chắn sẽ cần đến kiến trúc BusinessObjects Enterprise. Đây là kiến trúc quản lý báo biểu phía máy chủ Server hỗ trợ lập trình phân tán cho những hệ thống ứng dụng lớn. BusinessObjects Enterprise cung cấp:

- ◆ Kiến trúc mạnh mẽ cho việc đăng nhập và xử lý Report trên nhiều máy chủ Server.
- ◆ Kỹ thuật lưu trữ Report và khả năng xem trên trình duyệt Web.
- ◆ Engine lập kế hoạch chạy báo biểu định kỳ linh động theo một khoảng thời gian và ngày cụ thể.
- ◆ Tùy biến phân tán, chuyển tải, lưu trữ báo biểu theo email, ra máy in, đĩa (disk), FTP.
- ◆ Khả năng xuất báo biểu ra nhiều định dạng khác nhau như: Adobe Acrobat Reader, Excel, Word, RTF, Text.
- ◆ Quản lý bảo mật báo biểu cho phép nhiều người truy cập.

## 2. TÌM HIỂU KIẾN TRÚC SERVER

BusinessObjects Enterprise cung cấp một kiến trúc đa Server, mỗi Server xử lý những tác vụ hoàn toàn độc lập. Hình 12.1 là cấu trúc cơ bản của BusinessObjects Enterprise.



Hình 12.1: Kiến trúc cơ bản của BusinessObjects Enterprise.

Máy chủ trung tâm gọi là CMS (Central Management Server) CMS tiếp nhận xử lý các yêu cầu gửi đến từ các máy chủ Web Server như các tác vụ đăng nhập, xác thực quyền xem báo biểu, lập lịch thực thi Report. Các báo biểu Report muốn sử dụng trong môi trường BusinessObjects đều phải được lưu vào kho chứa (quá trình này gọi là xuất bản hay Publishing) theo cách sau: trong môi trường thiết kế Crystal Reports, chọn **File > Save As** và nhập vào biểu tượng Enterprise để đăng nhập vào BusinessObjects Enterprise và lưu Report vào thư mục riêng. Bạn cũng có thể sử dụng công cụ Crystal Management Console, hoặc Web Publishing Wizard của Crystal để xuất bản file Report lưu vào kho chứa của BusinessObjects.

Sau khi xuất bản Report, một mục nhập chứa thông tin báo biểu sẽ được lưu vào hệ thống CSDL của CMS và báo biểu được sao chép một bản vào máy chủ lưu trữ file mang tên FRS (File Repository Server). FRS có thể phân làm hai loại, máy chủ FRS "nguồn" (hay input) sẽ chứa tất cả các mẫu template và máy chủ khác gọi là FRS "kết xuất" (hay output) sẽ lưu nội dung báo biểu kết quả sau khi thực thi. Các Report template chứa trên FRS nguồn có thể lập lịch biểu để chạy thực thi cho ra các Report lưu vào FRS "kết quả" chờ người dùng yêu cầu hiển thị.

## 3. CÀI ĐẶT BUSINESSOBJECTS ENTERPRISE XI

Để tiến hành cài đặt BusinessObjects Enterprise XI, chạy file *Setup.exe* từ đĩa CD-ROM. Cửa sổ chào mừng hiển thị như hình minh họa 12.2.



Chọn **Install**, trình Wizard sẽ hướng dẫn bạn qua các bước cài đặt, bạn chỉ việc nhấn nút **Next** cho đến khi quá trình cài đặt hoàn tất.

Theo mặc định, BusinessObjects Enterprise XI được cài đặt với quyền quản trị (Administrator) và mật khẩu trống.



Hình 12.2: Giao diện cài đặt BusinessObjects Enterprise XI.

Bạn có thể kiểm tra quá trình cài đặt diễn ra thành công hay không sau đó bằng cách sử dụng Central Management Console. Để gọi trình này bạn mở *menu Start* và click vào **Programs > BusinessObjects Enterprise XI > .NET Administrator Launchpad**.



Hình 12.3: Crystal Management Console.

Một trang Web sẽ hiển thị. Bạn có thể đăng nhập vào trong Crystal Management Console bằng tài khoản Administrator chỉ định trong quá trình cài đặt. Sau khi đăng nhập bạn có thể quản lý người dùng (User), nhóm người dùng (Group), thư mục (Folder), xuất bản báo biểu Report của mình lưu vào kho chứa của BusinessObjects Enterprise.

## 4. XỬ LÝ BÁO BIỂU VỚI ĐƯỜNG DẪN URL

Hầu hết các lệnh và yêu cầu phục vụ bạn gửi đến BusinessObjects Server thông qua sử dụng địa chỉ Web URL. Bạn có thể truyền các lệnh, tham số trên địa chỉ URL yêu cầu BusinessObjects thực hiện các công việc như:

- Đăng nhập vào BusinessObjects Enterprise.
- Xem và thực thi báo biểu Report.
- Chọn Viewer để hiển thị nội dung báo biểu (ActiveX, DHTML, Java).
- Truyền biến cho các Field tham số.
- Thiết lập công thức lọc mẫu tin.
- Chuyển đổi định dạng báo biểu Report.

### 4.1. Xem báo biểu

Muốn xem báo biểu nào đó trong BusinessObjects bạn phải biết mã số ID của nó. Mã số này được cấp khi lần đầu tiên bạn xuất bản (Publishing) báo biểu đưa vào kho lưu trữ của BusinessObjects.

Để tìm mã ID của Report, bạn mở trang Central Management Console thực hiện theo các bước sau:

1. Từ Start menu chọn **Programs > BusinessObjects Enterprise 11 > BusinessObjects Enterprise > .NET Administrator Launchpad**.
2. Đăng nhập vào hệ thống với tên user là Administrator và mật khẩu để trống.
3. Click vào biểu tượng thư mục (Folders) và tìm đến vị trí thư mục nơi báo biểu được lưu trữ. Trong ví dụ này, chúng ta sẽ sử dụng một trong các Report mẫu mà BusinessObjects Enterprise cung cấp. Bạn chọn vị trí **Report Samples > General Business**, sau đó chọn Report **World Sales**.

4. Click vào Report World Sales để mở trang thuộc tính của nó và nhấn nút Preview để xem nội dung báo biểu Report.



Hình 12.4: Xem bản biểu World Sales từ URL.

5. Hãy nhìn vào đường dẫn URL trên thanh Address Bar của trình duyệt Web, tại đây bạn sẽ thấy mã số ID của báo biểu do BusinessObject sinh ra. Ví dụ, trên máy chủ của chúng tôi mã số ID là 844 nhưng nó có thể khác trên Server của bạn.

Để xem một báo biểu Report sử dụng địa chỉ URL, bạn gán ID của Report theo cú pháp sau:

*<http://servername/businessobjects/viewerpt.cwr?id=Mã số ID>*

Ví dụ:

*<http://servername/businessobjects/viewerpt.cwr?id=844>*

## 4.2. Đăng nhập vào BusinessObjects Enterprise

Mỗi khi sử dụng địa chỉ Web URL để xem báo biểu Report, bạn sẽ bị nhắc nhở nhập vào tài khoản và mật khẩu thông qua trang Log On như hình 12.5. Muốn đăng nhập tự động, bạn cần sử dụng thêm các tham số sau trên địa chỉ URL:

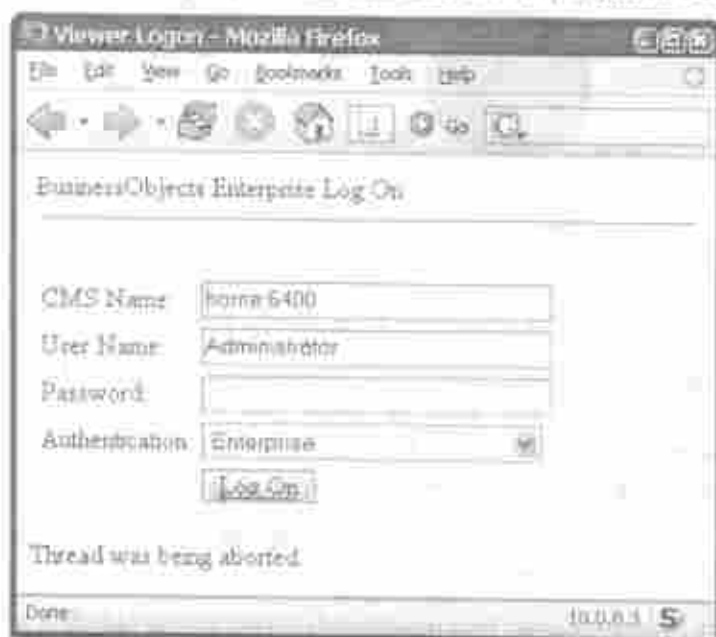
- APSUSER
- APSPASSWORD

Các tham số phân cách nhau bằng dấu "&". Ví dụ để mở báo biểu có mã số ID là 844 theo tên đăng nhập (username) là Administrator, mật khẩu (password) là admin123, bạn tạo địa chỉ URL như sau:

`http://servername/businessobjects/viewerpt.cwr?id=844&aspuser=admini  
strator&asppassword=admin123`

### 4.3. Chọn lựa Viewer

BusinessObjects Enterprise cung cấp một số trình xem nội dung báo biểu trên Web. Theo mặc định trình xem báo biểu mặc định là DHTML.



Hình 12.5: DHTML Viewer.

Trình xem báo biểu DHTML có lợi thế là xem được trên hầu hết các trình duyệt mà không cần cài thêm bất kỳ dạng hỗ trợ plug-in nào. Nó cung cấp phần lớn các chức năng mà người xem Report cần đến (Ví dụ: hộp danh sách số xuống, tìm kiếm trong báo biểu...).

Cách xem báo biểu thứ hai trong BusinessObjects Enterprise là sử dụng Viewer dạng điều khiển ActiveX nhưng trong trang Web như hình 12.6.



Hình 12.6: ActiveX Viewer.

Trình Viewer ActiveX chỉ làm việc với trình duyệt Internet Explorer. Nó có giao diện và thao tác hoàn toàn giống với Viewer của ứng dụng Windows Desktop.

Cách xem báo biểu cuối cùng là sử dụng Applet của Java như hình 12.7.



Hình 12.7: Java Viewer.

Trình Viewer dạng Applet cũng tương tự ActiveX nhưng sử dụng mã Java và được nhiều trình duyệt ủng hộ hơn ActiveX.

Cả hai dạng Viewer ActiveX và Applet Java đều yêu cầu cài thêm hoặc download phần hỗ trợ Plug-in.

Để chỉ định trình Viewer nào sẽ sử dụng khi hiển thị nội dung báo biểu, bạn sử dụng các tham số INIT trên đường dẫn URL với các giá trị như sau:

<b>Kiểu thẩm định</b>	<b>Biến giá trị</b>
DHTML Viewer	HTML
ActiveX Viewer	ACTX
Applet Java	JAVA

Ví dụ, để xem báo biểu Report bằng Viewer ActiveX, bạn truyền tham số trên địa chỉ như sau:

*<http://servername/businessobjects/viewrpt.cwr?id=844&init=actx>*

#### **4.4. Chuyển đổi định dạng báo biểu**

Bạn có thể yêu cầu chuyển đổi nội dung báo biểu thành các định dạng khác nhau như Excel, PDF, HTML. Bạn sử dụng biến tham số CMD với giá trị CMD = Export cùng với biến *export\_FMT* cho biết các định dạng chuyển đổi theo theo các giá trị sau:

<b>Định dạng kết xuất</b>	<b>Giá trị biến</b>
Crystal Reports	U2FCR:0
Microsoft Excel	U2FXL:3
Microsoft Excel (Data Only)	U2FXL:4
Microsoft Word	U2FWORDW:0
Rich Text	U2DRTF:0
Plain Text	U2FTEXT:1

Ví dụ, để chuyển đổi Report của bạn sang dạng Rich Text, địa chỉ URL sẽ được truyền tham số như sau:

[http://servername/businessobjects/viewrpt.cwr?id=844&cmd=export&export\\_fmt=U2DRTF:0](http://servername/businessobjects/viewrpt.cwr?id=844&cmd=export&export_fmt=U2DRTF:0)

Ngoài ra, bạn có thể chỉ định các trang thông qua biến tham số *export\_OPT*. Ví dụ, để chuyển đổi trang 1-5 của báo biểu sang dạng RTF bạn xây dựng URL như sau:

[http://servername/businessobjects/viewrpt.cwr?id=844&cmd=export&export\\_fmt=U2DRTF:0&EXPORT\\_OPT=\[1-5\]](http://servername/businessobjects/viewrpt.cwr?id=844&cmd=export&export_fmt=U2DRTF:0&EXPORT_OPT=[1-5])

#### 4.5. Làm việc với các Field tham số

Bạn có thể truyền các Field tham số của báo biểu trên đường dẫn URL thông qua chuỗi **PROMPTEX** kết hợp với tên Field tham số của báo biểu. Ví dụ, nếu bạn có một báo biểu Report chứa Field tham số *EnterCountry*, bạn có thể truyền giá trị của *EnterCountry* trên đường dẫn URL bằng cách đặt nối tên tham số và chuỗi PROMPTEX với dấu gạch ngang (*PROMPTEX-EnterCountry*) như sau:

[http://servername/businessobjects/viewrpt.cwr?id=844&promptex-entercountry="Australia"](http://servername/businessobjects/viewrpt.cwr?id=844&promptex-entercountry=)

Trong ví dụ này chúng ta truyền giá trị chuỗi cho Field tham số, tuy nhiên bạn hoàn toàn có thể truyền các kiểu giá trị khác, chẳng hạn nếu muốn truyền dạng số bạn có thể đặt giá trị trực tiếp không bọc trong dấu nháy như sau:

<http://servername/businessobjects/viewrpt.cwr?id=844&promptex-number=1500>

Bạn cũng có thể truyền biến dạng ngày bằng cách chỉ định hàm DATE theo khuôn dạng DATE(YYYY,MM,DD). Ví dụ để chỉ định ngày cho Field tham số mang tên *OrderDate*, địa chỉ URL sẽ được xây dựng như sau:

[http://servername/businessobjects/viewrpt.cwr?id=844&promptex-orderdate="Date\(2006,01,01\)"](http://servername/businessobjects/viewrpt.cwr?id=844&promptex-orderdate=)

Nếu Report của bạn chứa nhiều Field tham số, bạn có thể phân cách các tham số bằng ký hiệu (&) như sau:

[http://servername/businessobjects/viewrpt.cwr?id=844&promptex-orderdate="Date\(2006,01,01\)"&promptex-ordercount=10000](http://servername/businessobjects/viewrpt.cwr?id=844&promptex-orderdate=)

Để truyền dãy các giá trị cho một biến tham số bạn phân cách các giá trị bằng dấu phẩy (.). Ví dụ, chúng ta có thể truyền các giá trị *Australia*, *USA* và *Canada* cho biến tham số *EnterCountry* như sau:

```
http://servername/businessobjects/viewrpt.cwr?id=844&promptx-  
entrycountry="Australia","USA","Canada"
```

Với các giá trị liên tục bạn có thể chỉ định giá trị biên trên và biên dưới trong cặp ngoặc vuông như sau:

```
http://servername/businessobjects/viewrpt.cwr?id=844&promptx-  
orderamount=[10000-20000]
```

Nếu không lấy giá trị 10000 và 20000 bạn sử dụng dấu ngoặc đơn thay cho ngoặc vuông như sau:

```
http://servername/businessobjects/viewrpt.cwr?id=844&promptx-  
orderamount=(10000-20000)
```

Còn nếu lấy một giá trị biên trên/hay dưới bạn sử dụng dấu ngoặc vuông đặt phía giá trị muốn lấy, chẳng hạn để lấy giá trị 1000:

```
http://servername/businessobjects/viewrpt.cwr?id=844&promptx-  
orderamount=[10000-20000)
```

## 4.6. Lọc mẫu tin

Một chức năng hoạt động khác của URL là có thể thiết lập các giá trị lọc mẫu tin. Công thức lọc trong của báo biểu Report được dùng để trả về chỉ những mẫu tin thỏa điều kiện yêu cầu.

Bạn dùng biến lệnh SF để truyền công thức lọc này cho địa chỉ URL. Chẳng hạn nếu muốn lọc tất cả những khách hàng thuộc quốc gia *Australia* bạn có thể truyền công thức SF trên URL như sau:

```
http://servername/businessobjects/viewrpt.cwr?id=844&sf={Customer.  
Country}="Australia"
```

Bạn cũng có thể xây dựng một công thức lọc phức tạp hơn như ví dụ sau:

```
http://servername/businessobjects/viewrpt.cwr?id=844&sf =  
{Customer.Region} = "New South Wales" and {Customer.City} = "Sydney"  
and {Customer.Country} = "Australia" and {Orders.Order Amount} >  
$10000.00 and {Orders.Order Date} > DateTime(2005, 01, 01, 00, 00, 00)
```



#### 4.7. Làm việc với CSDL.

Hầu hết các báo biểu Report đều dựa trên nguồn dữ liệu bảo mật, mỗi khi đọc nguồn dữ liệu này người sử dụng phải nhập vào tên (username) và mật khẩu (password) để xác thực quyền truy xuất (CSDL). Bạn có thể chỉ định trước các thông tin đăng nhập để không phải chỉ định các thông tin này trên địa chỉ URL. Để làm điều này, bạn sử dụng Central Management Console và thực hiện theo các bước sau:

1. Từ *Start Menu*, click vào **Programs > BusinessObjects 11 > BusinessObjects Enterprise > .NET Administration Launchpad**.
2. Click vào đường link đến **Central Management Console**.
3. Đăng nhập vào **Central Management Console** với tên người dùng là **Administrator** và mật khẩu để trống.
4. Click vào biểu tượng Folder và di chuyển đến Folder nơi mà Report của bạn được lưu giữ.
5. Click vào tên Report để mở trang thuộc tính của nó, như hình 12.8.
6. Click vào tab **Process** sau đó click vào link **Database**. Trang thuộc tính Database sẽ hiển thị:



Hình 12.8: Trang thuộc tính báo biểu.

7. Nhập thông tin đăng nhập username và mật khẩu (password) vào để sử dụng cho Report này.
8. Cuộn xuống dưới cho đến cuối trang và chọn đánh dấu nút *Use Same Database Logon as When Report is run.*
9. Click vào nút *Update* ở phía cuối trang để lưu lại các thiết lập cho báo biểu.

Lúc này mỗi khi thực thi báo biểu bạn không còn cần phải nhập vào thông tin đăng nhập CSDL nữa.

## 5. TÓM TẮT

Kiến trúc BusinessObjects Enterprise phục vụ cho các ứng dụng lớn với hàng trăm báo biểu, khả năng quản lý Report mạnh mẽ, phân chia trên các máy chủ an toàn và bảo mật cao. Môi trường BusinessObjects cung cấp hầu như tất cả những cơ chế phân quyền, bảo vệ Report, xử lý các Report phức tạp với khối lượng dữ liệu kết xuất lớn, thực thi Report theo lịch định kỳ, điều mà những chương trình bình thường khó thực hiện được. Bạn chỉ chuyên tâm vào phát triển nội dung của báo biểu theo nghiệp vụ của mình, phần còn lại hãy để cho BusinessObjects lo liệu. Ngoài ra, bạn còn có thể lập trình và sử dụng các hàm API tương tác với BusinessObjects, nếu quan tâm đến vấn đề này, hãy tham khảo giáo trình “Visual Basic 2005 - Tập 4, quyển 1 - Crystal Reports Developer” do nhà sách Minh Khai đã xuất bản.

# Sử dụng

## CRYSTAL REPORTS XI

---

Thực hiện liên doanh: Công ty TNHH Minh Khai S.G  
E-mail: mk.book@minhkhai.com.vn - Website: www.minhkhai.com.vn

### Tổng phát hành

- ❖ Nhà sách Minh Khai: 249 Nguyễn Thị Minh Khai - Quận 1 - TP.HCM  
ĐT: (08) 9.250.590 - 9.250.591 - Fax: (08) 9.257.837
- ❖ Nhà sách Minh Châu: Nhà 30 - Ngõ 22 - Tạ Quang Bửu - Bách Khoa - Hà Nội  
ĐT: (04) 8.692.785 - Fax: (04) 8.683.995

### Đại lý các khu vực

- ❖ Nhà sách Huy Hoàng: 95 Núi Trúc - Kim Mã - Ba Đình - Hà Nội  
ĐT: (04) 7.365.859
- ❖ Cty có phần sách thiết bị trường học Đà Nẵng: 78 Bạch Đằng - Đà Nẵng  
ĐT: 0511.337100
- ❖ Nhà sách Chánh Trí: 116A Nguyễn Chí Thanh - Đà Nẵng  
ĐT: 0511.820129
- ❖ Cty phát hành sách Khánh Hòa:
  - Nhà sách Ponagar: 73 Thống Nhất - Nha Trang - Khánh Hòa  
ĐT: 058.822636
  - Siêu thị sách Tân Tiến: 11 Lê Thành Phương - Nha Trang - Khánh Hòa  
ĐT: 058.827303
- ❖ Nhà sách Năm Hiền: 79/6 Xô Viết Nghệ Tĩnh - TP. Cần Thơ  
ĐT: 071. 821668

---

### ■ SỬ DỤNG CRYSTAL REPORTS XI

*(Tin học và đời sống)*

Chịu trách nhiệm xuất bản: Quang Thắng

Biên tập nội dung: Thảo Nguyễn

Sửa bản in: Minh Như

Biên: Hữu Nghĩa

In 4.000 cuốn, có kèm CD bài tập, khổ 16x24cm tại

Xí nghiệp In Machinco

21 Bùi Thị Xuân, Q.1, TP.Hồ Chí Minh

Số đăng ký kế hoạch xuất bản:

06-2006/CXB/102-1.10/PĐ

Cục xuất bản ký ngày 14 tháng 10 năm 2005

In xong và nộp lưu chiểu tháng 12 năm 2006

# Sử dụng Crystal Reports XI

- Học Crystal Reports từ kinh nghiệm của chuyên viên thiết kế, cách tạo báo biểu trong ứng dụng .NET và Java.
- Cung cấp một cách rõ ràng, dễ hiểu kinh nghiệm và kiến thức thiết kế ứng dụng liên quan đến báo biểu.
- Trình bày các chức năng cấp cao của Crystal Reports mà bạn cần như: Sub-Report, Cross-tab, Charts và biến tham số động..
- Giới thiệu các cách triển khai và tích hợp trên các nền ứng dụng Visual Studio 2005, Windows Forms, ASP.NET, JSP và BusinessObjects Enterprise.
- Kèm theo sách là một CD-ROM với tất cả các đoạn mã và các báo biểu ví dụ có trong sách.



*Minh Khai*



Giá: 69.000 đ