

MỤC LỤC

Chương 1: Tổng quan về Crystal Reports .NET.....	5
1.1 Crystal Reports là gì?.....	5
1.2 Đặc điểm của Crystal Reports.....	5
Chương 2: Thiết kế một báo cáo sử dụng Report Design Environment.....	7
2.1 Các thành phần Report Designer.....	7
2.2 Các vấn đề cơ bản thiết kế báo cáo.....	8
2.2.1 Các Section của một báo cáo.....	8
2.2.2 Các đối tượng trong Field Explorer:.....	9
2.3 Thiết kế một báo cáo đơn giản:.....	10
2.3.1 Thêm một báo cáo mới vào ứng dụng.....	11
2.3.2 Chọn dữ liệu nguồn (data source).....	14
2.3.3 Thiết lập mối quan hệ giữa các bảng.....	17
2.3.4 Chọn ra các Field cho bản báo cáo.....	17
2.3.5 Gộp nhóm và Sắp xếp (Grouping - Sorting).....	19
2.3.6 Bổ sung thông tin tổng kết.....	20
2.3.7 Các tính năng phân tích.....	21
2.3.8 Biểu đồ và Đồ thị (Charting - Graphing).....	23
2.3.9 Sàng lọc thông tin trên bản báo cáo.....	24
2.3.10 Chọn kiểu định dạng cho báo cáo.....	25
2.3.11 Hoàn tất cơ bản việc thiết kế bản báo cáo.....	26
2.4 Preview bản báo cáo.....	27
2.5 Tinh chỉnh và hoàn thiện bản báo cáo.....	28
2.5.1 Định dạng các phân đoạn.....	30
2.5.2 Định dạng các Field:.....	31
2.5.3 Hiệu chỉnh Sắp xếp.....	32
2.5.4 Hiệu chỉnh Gộp nhóm.....	34
2.5.5 Vẽ khung.....	35
Chương 3 : Công thức Basic Syntax.....	38
3.1 Hộp thoại Formula Editor.....	38
3.1.1 Các thành phần chính.....	39

3.1.2 Kiểm tra lỗi cú pháp.....	40
3.2 Định dạng động sử dụng công thức.....	41
3.3 Tạo các công thức với Basic Syntax.....	44
3.3.1 Các qui ước lập trình Basic Syntax.....	44
3.3.1.1 Viết các ghi chú, diễn giải.....	44
3.3.1.2 Trình bày các câu lệnh.....	44
3.3.1.3 Tham chiếu các field báo cáo.....	45
3.3.1.4 Khai báo biến.....	45
3.3.2 Các kiểu dữ liệu.....	46
3.3.2.1 Các kiểu dữ liệu chuẩn.....	46
3.3.2.2 Kiểu dữ liệu mảng.....	46
3.3.2.3 Kiểu dữ liệu Range.....	47
3.3.3 Các toán tử đơn giản.....	49
3.3.3.1 Toán tử toán học.....	50
3.3.3.2 Toán tử Boolean.....	50
3.3.3.3 Toán tử so sánh.....	51
3.3.3.4 Toán tử chuyển đổi kiểu dữ liệu.....	51
3.3.4 Các Hàm cơ bản.....	53
3.3.4.1 Các hàm tổng kết.....	53
3.3.4.2 Các hàm xử lý chuỗi.....	53
3.3.4.3 Các hàm Date & Time	57
3.3.4.4 Các hàm toán học.....	58
3.3.4.5 Định dạng các trị kết xuất.....	59
3.3.5. Các cấu trúc điều khiển.....	60
3.3.5.1 Các cấu trúc điều kiện.....	60
3.3.5.2 Các cấu trúc lặp.....	61
3.4 Công thức chuyển đổi số thành chuỗi Tiếng Việt.....	62
Chương 4: Crystal Reports Object Model.....	66
4.1 Các bước cơ bản để tùy biến bảng báo cáo.....	66
4.2 Đối tượng ReportDocument.....	68
4.3 Đối tượng DataDefinition:.....	70

4.3.1 Tìm hiểu các lớp Sorting và Grouping:.....	70
4.3.2 Ví dụ Modify Sorting:.....	74
4.4 Đối tượng CrystalReportViewer:.....	76
Chương 5: Vấn đề kết xuất báo cáo trong trường hợp dữ liệu nguồn bị thay đổi....	80
5.1 Lớp Database:.....	81
5.2 Thay đổi thông tin kết nối với bảng dữ liệu nguồn.....	81
5.2.1 Sử dụng ReportDocument.....	83
5.2.2 Sử dụng CrystalReportViewer	83
Chương 6: Kết xuất báo cáo ra máy in, word, excel:.....	85
6.1 Xuất báo cáo thông qua Viewer.....	85
6.1.1 Ra máy in.....	85
6.1.2 Ra tập tin.....	86
6.2 Xuất báo cáo thông qua lập trình.....	87
6.2.1 Ra máy in.....	87
6.2.2 Ra tập tin.....	88
Tài liệu tham khảo.....	88

Chương 1: Tổng quan về Crystal Reports .NET

1.1 Crystal Reports là gì?

Crystal Reports là công cụ thiết kế báo cáo cho phép bạn tạo ra những báo cáo bằng cách tìm và định dạng dữ liệu từ một nguồn dữ liệu hay từ những nguồn dữ liệu khác nhau .Bên cạnh đó , Crystal Reports có một ngôn ngữ riêng để tính toán và một số tính năng khác để biến những dữ liệu thô thành những báo cáo đầy tính chuyên nghiệp.Bạn có thể tạo những báo cáo đi từ những danh sách đơn giản chỉ gồm vài cột cho đến những báo cáo phức tạp có kèm biểu đồ ,bảng và chỉ số Key Performance Indicator (KPI).Ngoài ra , Crystal Reports có một số hàm API cũng như công cụ đặc biệt được thiết kế dành cho các nhà phát triển phần mềm cho phép hội nhập các báo cáo này vào trong ứng dụng riêng của họ.

1.2 Đặc điểm của Crystal Reports

- Bạn không cần mở một ứng dụng riêng rẽ để thiết kế báo cáo khi dùng Crystal Reports
- Đối với Windows Form , Crystal Reports cho phép xem báo cáo và cung cấp tất cả các chức năng cho người sử dụng bao gồm xoáy sâu vào chi tiết ,truy xuất ,xuất khẩu...Đối với ASP.NET ,cung cấp phần lớn các chức năng trong Windows Form Viewer trong môi trường DHTML “zero client ” (Nghĩa là không client nào được nạp xuống hay cài đặt).
- Crystal Reports giúp việc truy cập dữ liệu trở nên dễ dàng hơn.
- Crystal Reports cho phép truy cập những dữ liệu nguồn khác nhau.

- Crystal Reports cho phép chia sẻ sử dụng các bảng báo cáo cũng như tạo những ứng dụng được phân phối sử dụng cho nhiều người dùng.

Chương 2: Thiết kế một báo cáo sử dụng Report Design Environment

Trong chương này ta sẽ lần lượt sử dụng Report Designer bên trong Visual Studio.NET để thiết kế một báo cáo đơn giản. Bạn sẽ thấy trong chương này bạn chỉ đơn giản là click chọn và kéo thả là có thể hoàn thành một báo cáo mà không cần phải lập trình gì cả.

2.1 Các thành phần Report Designer

Trên Crystal Reports.NET cung cấp cho bạn các cửa sổ và các thanh công cụ giúp bạn thuận tiện hơn trong khi thiết kế báo cáo như hình 2-01.

Hình 2-01: Các thành phần của Report Designer

Thanh công cụ **Crystal Reports – Main** chứa các control định dạng như Font, Size... Thanh công cụ **Crystal Reports – Insert** cho phép bạn chèn vào Summary, Group, Subreport, Chart và Picture. Hộp công cụ Crystal Reports cho phép bạn thêm vào các đối tượng Text, Line và Box.

Muốn thêm một Field vào báo cáo, thì có cửa sổ **Field Explorer**. Bạn chỉ cần bung các nhánh tương ứng trên cửa sổ này và kéo thả Field nào bạn cần lên bản báo cáo.

Cuối cùng là cửa sổ **Report Designer**, đây chính là nơi bạn thiết kế bản báo cáo, nó bao gồm các section, trên mỗi section sẽ chứa các field, các đối tượng báo cáo.

2.2 Các vấn đề cơ bản thiết kế báo cáo

2.2.1 Các Section của một báo cáo


Bảng 2-01 Các section trên báo cáo

Section báo cáo	Mô tả
Report Header	Xuất hiện trên trang đầu tiên của bản báo cáo và thường bị huỷ theo mặc nhiên .Có thể được dùng cho biết bắt đầu của một bản báo cáo mới hay dùng làm trang bìa
Report Footer	Xuất hiện vào cuối trang chót của bản báo cáo, theo mặc nhiên là cho thấy và có thể dùng tổng kết báo cáo (ghi số mẫu tin , ngày in báo cáo , tên tập tin,...)
Page Header	Xuất hiện trên đầu mỗi trang (khi sang trang) và có thể dùng ghi tựa đề báo cáo, tiêu đề các cột , số trang...
Page Footer	Xuất hiện vào cuối mỗi trang và có thể dùng in số trang ngày in báo cáo,...
Group Header	Xuất hiện trên đầu mỗi nhóm (khi có ngắt cấp bậc) và thường ghi tên nhóm mới
Group Footer	Xuất hiện vào cuối một nhóm mẫu tin , và thường được dùng để in ra tên nhóm cũ , tổng cộng của nhóm hay tổng kết
Details	Xuất hiện đối với mỗi mẫu tin thường được dùng in thông tin của cột và có thể được bung thành section các vùng mục tin lớn hơn hoặc để tạo biểu mẫu

Hình 2-02 sau đây cho thấy cách bố trí các section trên bản báo cáo

Hình 2-02 Cách bố trí section trên báo cáo

2.2.2 Các đối tượng trong Field Explorer:

- **Database Fields:** Chứa tất cả các field mà bạn đưa vào bản báo cáo, các field này có thể được lấy từ Table, View hay Stored Procedures. Các field được đưa vào bản báo cáo thì một dấu check  sẽ hiện lên bên cạnh, cho biết field này đang được sử dụng.
- **Formula Fields:** dùng để đưa các tính toán phức tạp vào báo cáo. Có 2 cú pháp phải chọn là Crystal syntax , Basic syntax.
- **Parameter Fields:** dùng để nhắc nhở người dùng nhập vào các thông tin cần thiết đối với báo cáo đang chạy. Khi bạn tạo một parameter field và đưa vào báo cáo, thì Crystal Reports .NET sẽ hiển thị một khung đối thoại mặc nhiên nhắc nhở người dùng nhập vào chi tiết mà bạn vừa nhập vào ,khi nào bản báo cáo của bạn được xem trước.
- **Running Total Field:** dùng để tính toán dựa trên giá trị các mẫu tin của một field khác.
- **Group Name Fields:** Thể hiện các nhóm đang được sử dụng trong báo cáo.

- **SQL Expression Fields:** để đảm bảo việc tính toán của bạn được hiện trên server và bạn có thể thêm nhập vào tất cả các hàm của SQL.
- **Special Fields:** thể hiện các vùng mục tin đặc biệt. Bảng 2-02 sau đây liệt kê các field đặc biệt.

Bảng 2-02 Danh sách các field đặc biệt

Tên field đặc biệt	Mô tả
Print Date	In ra ngày báo cáo được in
Print Time	In ra thời giờ khi báo cáo được in ra
Modification Date	Ngày thay đổi chót của báo cáo
Modification Time	Thời giờ thay đổi chót của báo cáo
Data Date	Ngày khi dữ liệu được đọc từ căn cứ dữ liệu
Data Time	Thời giờ khi dữ liệu được đọc từ căn cứ dữ liệu
Record Number	Con số thứ tự nội bộ của mẫu tin được gán cho tất cả các mẫu tin được trả về cho bản báo cáo
Page Number	Số thứ tự trang
Group Number	Số thứ tự nội bộ nhóm
Total Page Count	Tổng số trang đếm được
Report Title	Tựa đề báo cáo như đã được lưu trữ trong Summary Information của tập tin báo cáo
Report Comments	Chú giải sẽ được đưa vào trong Summary Information của tập tin báo cáo.
Record Selection Formula	Công thức tuyển chọn mẫu tin được sử dụng bởi báo cáo
Group Selection Formula	Công thức tuyển chọn nhóm được sử dụng bởi báo cáo
File Path and Name	Đường dẫn và tên tập tin của .rpt
File Author	Tác giả báo cáo sẽ được đưa vào trong Summary Information của tập tin báo cáo.
File Creation Date	Ngày bản báo cáo được tạo ra
Pagd N of M	Theo đây N là trang hiện hành và M là tổng số trang báo cáo

- **Unbound Fields:** gồm có 7 loại kiểu dữ liệu **Boolean**, **Currency**, **Date**, **Date Time**, **Number**, **String**, và **Timer**. Unbound dùng để tạo một bản báo cáo generic rồi dùng lập trình cho đặt nội dung các vùng mục tin này vào lúc chạy.

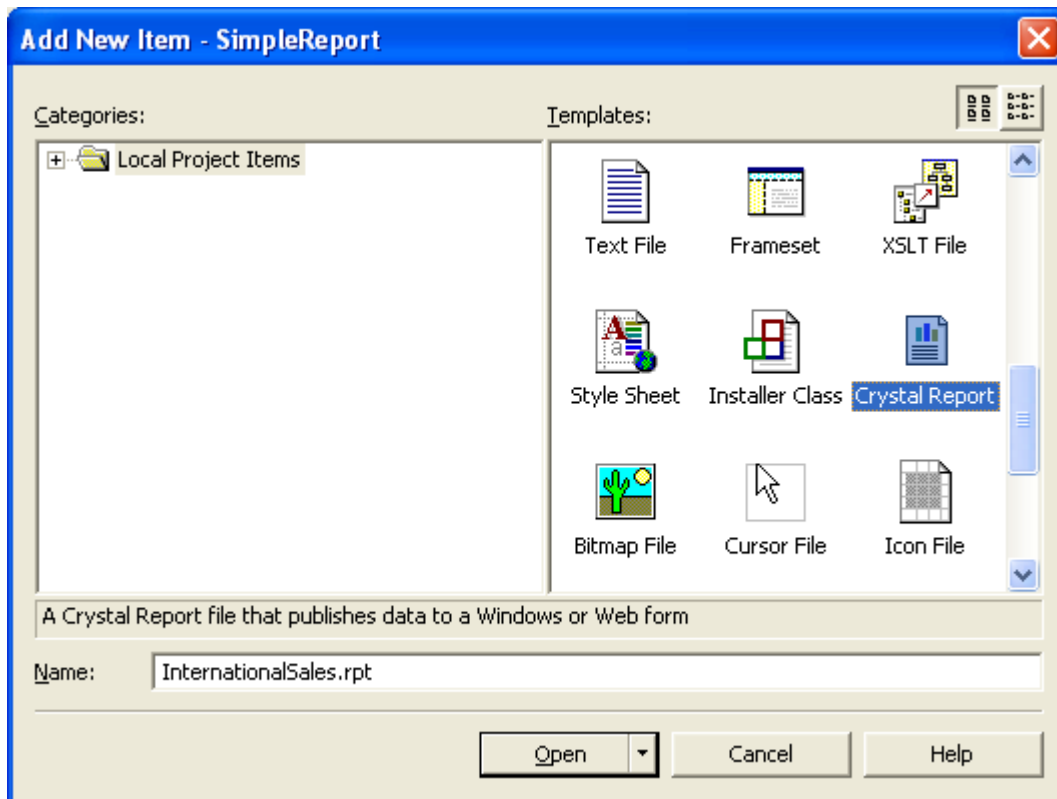
2.3 Thiết kế một báo cáo đơn giản:

Ví dụ sau đây sẽ giúp bạn tạo ra một báo cáo đơn giản, các bạn chỉ việc click và click để tạo ra một bản báo cáo mà không cần phải lập trình gì cả. Tùy theo đặc trưng của từng bản báo cáo mà bạn có thể bỏ qua một số bước không cần thiết, nhưng để giúp bạn hiểu

rõ hơn các bước nên với ví dụ này chúng ta sẽ khảo sát tất cả các bước mặc dù có một số bước chúng ta không làm gì cả mà chỉ nhấn Next để qua bước kế tiếp.

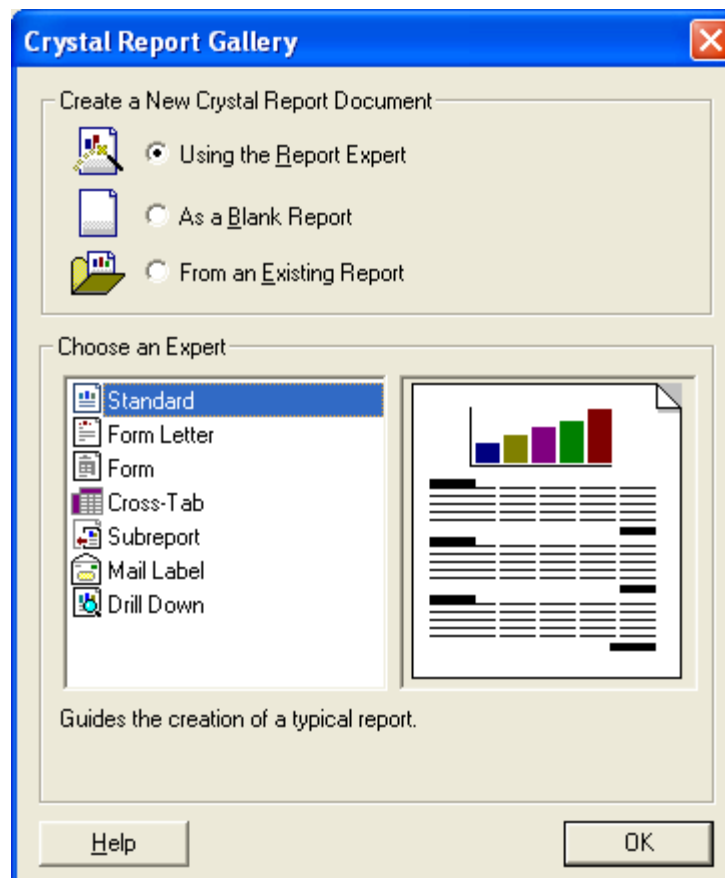
2.3.1 Thêm một báo cáo mới vào ứng dụng

Khởi động Microsoft Visual Studio .NET và tạo một project có tên SimpleReport. Sau đó, chọn **Project | Add New Item** sẽ xuất hiện hộp thoại **Add New Item** (Hình 2-03). Trên khung **Templates**, bạn chọn **Crystal Report** và đặt tên báo cáo là `InternationalSales.rpt`, rồi chọn **Open**.



Hình 2-03 Hộp thoại Add New Item

Hộp thoại **Crystal Report Gallery** sẽ hiển thị (Hình 2-04). Trên khung **Create a New Crystal Report Document** có ba lựa chọn được giải thích trong Bảng 2-03. Và trên khung **Choose an Expert** có bảy lựa chọn được giải thích trong Bảng 2-04. Bạn để các lựa chọn này theo giá trị mặc định ban đầu và nhấn **OK**.



Hình 2-04: Hộp thoại Crystal Report Gallery

Bảng 2-03: Các phương pháp tạo một Crystal Report Document mới

Phương pháp	Mô tả
Using the Report Expert	Như bạn có thể thấy trong hình 2-04 có nhiều Report Expert (Standard, Form Letter, Form, ...) mà bạn phải bước qua từng bước để tạo báo cáo. Đây là phương pháp phổ biến nhất như là điểm xuất phát cho những triển khai về sau. Lập trình viên kinh nghiệm cũng thích sử dụng phương pháp này.
As a Blank Report	Đối với lập trình viên kinh nghiệm quá quen thuộc với Report Designer thường dùng phương pháp này để tạo báo cáo rỗng và tự tay mình bổ sung từng phần vào bản báo cáo không cần sự hỗ trợ của Report Wizard. Khi mục chọn này được chọn thì các Expert khác sẽ không có sẵn.
From an Existing Report	Tạo một báo cáo mới tương tự một bản báo cáo đã có sẵn. Một lần nữa khi chọn mục này thì các Expert khác sẽ không có sẵn.

Bảng 2-04: Các loại Expert Report

Loại Expert	Mô tả
Standard	Được sử dụng thường xuyên và mang tính chung (generic) nhiều nhất. Bạn có thể sử dụng Standard Expert để tạo kiểu báo cáo kiểu hàng cột (tabular), bao gồm một số tính năng như grouping (gộp nhóm), sorting (sắp xếp) và summary (tổng kết). Standard Expert còn bao gồm khả năng thêm các hình đồ họa kiểu thống kê được gọi là chart, áp dụng một số style định sẵn và sàng lọc các mẫu tin cũng như có những tính năng phân tích cao cấp chẳng hạn TopN, BottomN (thí dụ Top10 hoặc Bottom100).
Form Letter	Bằng cách phối hợp các đối tượng văn bản và các Field CSDL, Crystal Reports.NET có thể dùng để tạo những văn thư (form letter).
Form	Dùng để tạo những bản báo cáo được thiết kế cho một biểu mẫu đặt trung (hoá đơn, bảng kê khai, ...). Bằng cách sử dụng form expert, bạn có khả năng cho nằm ẩn một hình ảnh ở dưới bản báo cáo (ví dụ như một hoá đơn trống) để có thể đưa vào đúng chỗ các field trên biểu mẫu. Từ đây, bạn có thể hoặc để yên hình ảnh và in ra biểu mẫu kèm theo các vùng mục tin lên giấy hoặc cho gỡ bỏ hình ảnh và in trực tiếp bản báo cáo lên các biểu mẫu.
Cross-Tab	Cross-tabs trong lòng Crystal Reports.NET trông giống như một bản tính (spreadsheet) với hàng cột chứa dữ liệu tổng kết. Bằng cách sử dụng Cross-tab Expert bạn có thể tạo một bản báo cáo với đối tượng cross-tab trong section report header.
Subreport	Subreport là những báo cáo con được chèn vào bản báo cáo chính. Subreport có thể không liên hệ hoặc các thông số có thể được trao qua giữa báo cáo chính và subreport để xác định nội dung cần hiển thị.
Mail Label	Crystal reports.NET hỗ trợ những báo cáo gồm nhiều cột và chức năng này làm cho ta có khả năng tạo những nhãn bì thư (mail label).

Drill Down	Khái niệm nằm đằng sau drill down (xoáy sâu vào chi tiết) là bạn cho hiển thị một summary (tổng kết) trong bản báo cáo của bạn và người sử dụng có thể xoáy vào summary để xem chi tiết hình thành tổng kết này thế nào.
------------	---

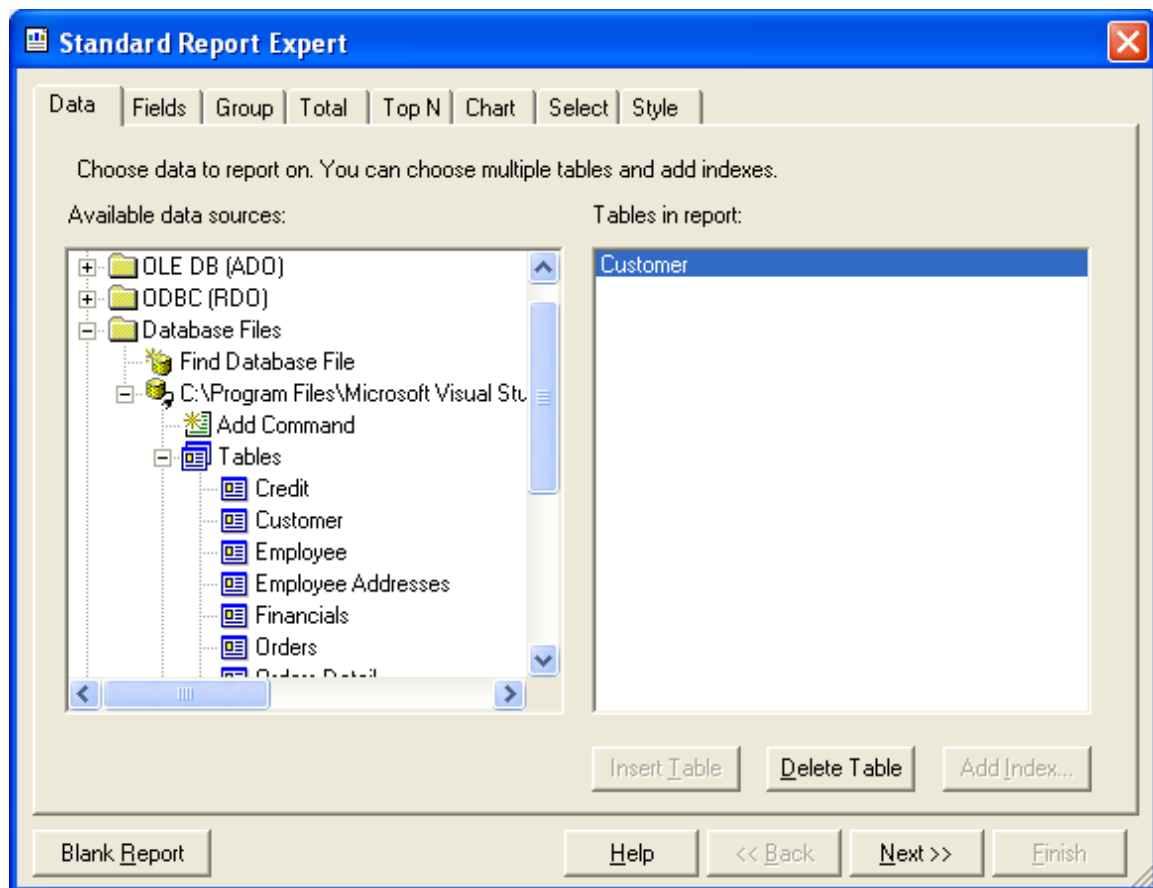
2.3.2 Chọn dữ liệu nguồn (data source)

Hộp thoại **Standard Report Expert** hiển thị lên như Hình 2-05.và mặc định sẽ focus về **Tab Data**. Khi đó bạn sẽ chọn dữ liệu nguồn thông qua **Tab Data** này, đây là bước đầu tiên cho bất kỳ loại expert nào. Có nhiều cách để chọn ra dữ liệu nguồn, Bảng 2-05 sau đây sẽ giải thích các lựa chọn trên khung **Available data source**, trong ví dụ này bạn click vào nút **Database Files** và hộp thoại **Open** mở ra ,chọn cơ sở dữ liệu `xtreme.mdb`.

Lưu ý: Cơ sở dữ liệu `xtreme.mdb` có sẵn tại thư mục *C:\ProgramFiles\Microsoft Visual Studio.NET\Crystal Reports\Samples\Database*.

Bạn bung nhánh **Tables** rồi double click lên bảng dữ liệu `Customer` để thêm vào khung **Tables in Report** bên phải. Nhấn **Next** để tiếp tục.

Lưu ý: Để bỏ chọn một bảng ra khỏi khung **Tables in Report** bạn click vào bảng đó sau đó nhấn nút **Delete Table** ở bên dưới khung **Tables in Report**.



Hình 2-05: Hộp thoại Standard Report Expert – Tab Data

Bảng 2-05: Các dữ liệu nguồn khác nhau

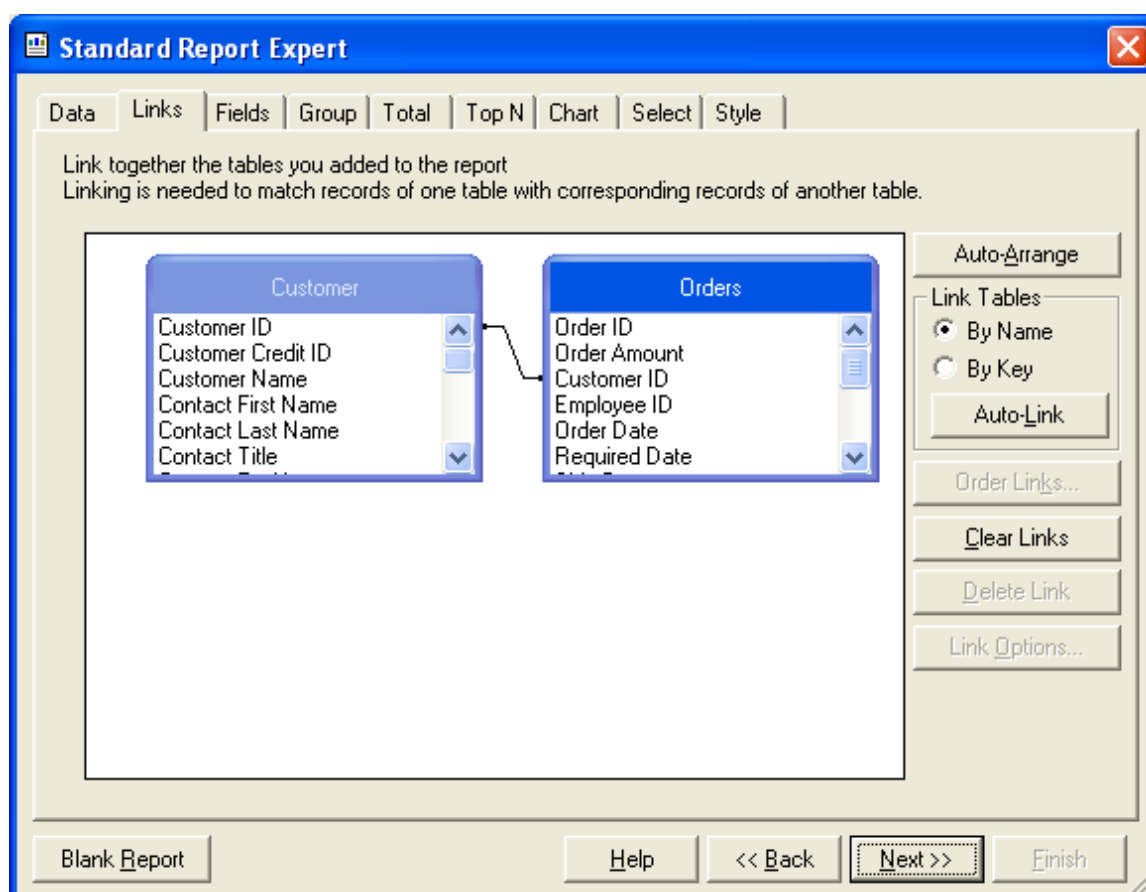
Dữ liệu nguồn	Mô tả
Project Data	Crystal Reports.NET có thể theo .NET Framework và báo cáo trực tiếp từ dataset xuất hiện trong ứng dụng của bạn. Bạn có thể thấy ADO.NET Datasets và Current Connections trong nhánh này. ADO.NET Datasets cho biết các lớp dataset được liệt kê trong cửa sổ Project Explorer. Các lớp này cho phép bạn chuyển gắn kết hầu như với bất cứ kiểu dữ liệu nào. Còn Current Connections cho phép bạn chọn bất cứ kết nối nào hiện được thiết lập trong cửa sổ Server Explorer
OLE DB (ADO)	Mục này cho phép tạo một connection string để truy cập các dữ liệu nguồn sử dụng một OLE BD driver bao gồm SQL server, Oracle, MS Jet 3.51/4.00 (Access, Excel, Paradox, Dbase , ...).

	Mục này sẽ cho hiện lên hai màn hình wizard “OLE DB Provider” và “Connection Information” yêu cầu bạn cho biết dữ liệu nguồn phải kết nối và bất cứ thông tin nào có ý nghĩa về vị trí và đăng nhập.
ODBC (RDO)	Mục này cho phép tạo một connection string để truy cập các dữ liệu nguồn sử dụng một ODBC driver cũng có 2 màn hình wizard “Datasource Selection” và “Connection information” cho phép bạn chọn một System DSN hoặc file DSN hiện hữu. Ngoài việc báo cáo từ các bảng dữ liệu, view và stored procedure , ... Crystal Reports.NET còn cho phép bạn nhập vào một lệnh SQL dùng làm cơ sở cho bản báo cáo của bạn.
Database Files	Mục này cho phép bạn chọn một trong số những dạng thức CSDL kiểu tập tin bao gồm Access, Excel, XML, TTS(loại tập tin định nghĩa của Crystal Field Definition).
Favorites	Mục này cho phép bạn chọn ra dữ liệu nguồn thường xuyên được dùng đến mà bạn đã đưa vào danh sách Favorites. Các item hiện hữu có thể được thêm vào bằng cách right-click lên chúng và chọn mục Add to Favorites . Bạn cũng có thể gỡ bỏ một item khỏi Favorites bằng cách click lên item rồi ấn <F2>.
History	Mục này cho phép bạn chọn ra dữ liệu nguồn được dùng trong các bản báo cáo khác trong cùng dự án.
More Data Sources	Mục này bao gồm : ADO.NET(XML), access/Excel(DAO) và Field Definition Only. ADO.NET(XML) cho phép tìm đọc các mẫu tin bằng cách khai báo một lối tìm về tập tin XML. Bạn cũng có thể chọn ra một ADO.NET datasets hiện hữu. Còn Access/Excel (ADO) cho phép tìm đọc các mẫu tin bằng cách dùng ADO recordset truy cập một CSDL MS Access hoặc Excel spreadsheet. Còn Field Definition Only chỉ dùng đối với Crystal Field Definition (TTX) mang tính tương thích lùi và không được triển khai mới.

2.3.3 Thiết lập mối quan hệ giữa các bảng

Nếu ở bước trước bạn chọn ra cùng lúc nhiều bản dữ liệu thì bây giờ hộp thoại **Standard Report Export** sẽ hiển thị **Tab Link** cho phép bạn khai báo các bảng dữ liệu này có những mối quan hệ nào, hình 2-06.

Lưu ý: Trong ví dụ này chúng ta chỉ sử dụng một bảng customer nên sẽ bỏ qua bước này.

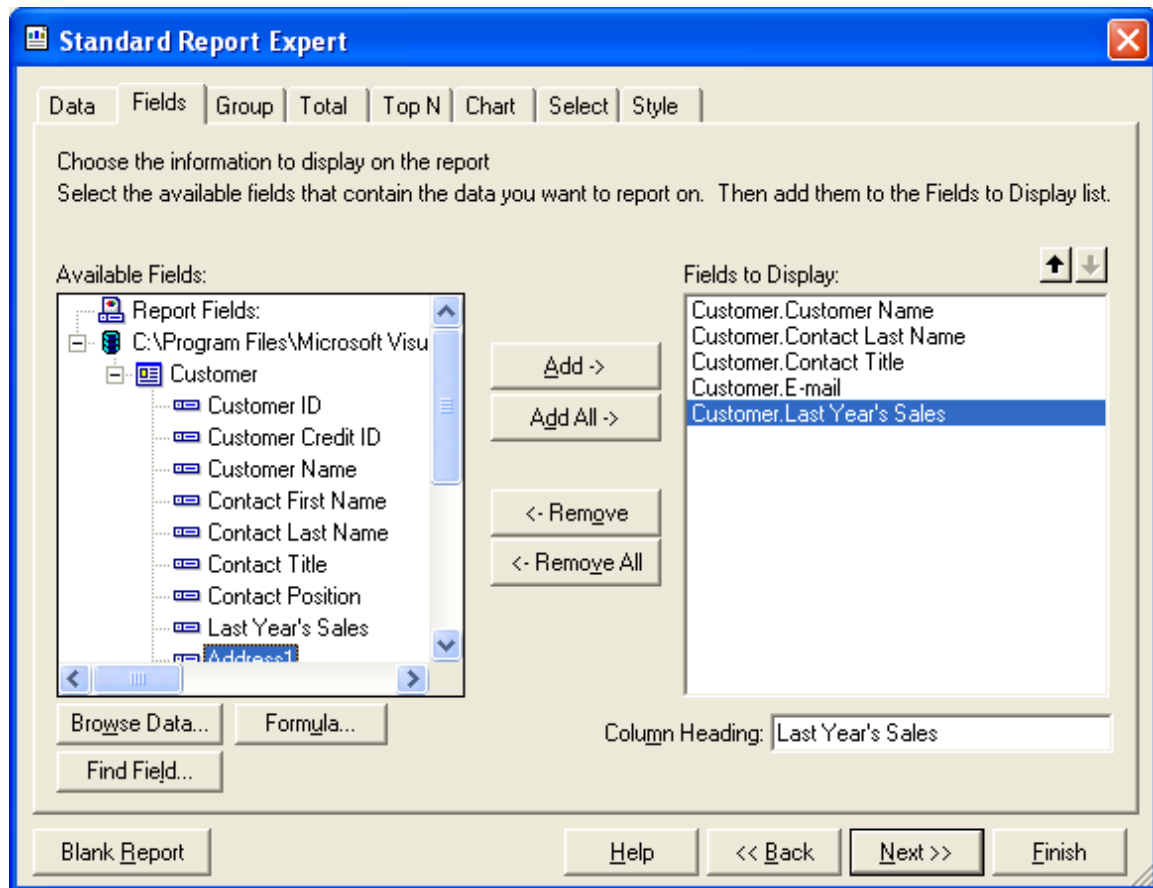


Hình 2-06: Hộp thoại Standard Report Expert – Tab Link


2.3.4 Chọn ra các Field cho bản báo cáo

Bước kế tiếp trên hộp thoại **Standard Report Export** là chọn ra các field sẽ xuất hiện lên bản báo cáo. Trên **Tab Field** (Hình 2-07), muốn sử dụng field nào bạn chọn field đó trên khung **Available Fields** rồi nhấn nút Add ➡ để chuyển field chọn qua danh sách **Field to Display** bên phải. Để tiện dụng hơn, hộp thoại **Standard Report Export** cung cấp cho bạn hai chức năng **Browse Data** và **Find Field** trên **Tab Field**. **Browse Data**

giúp bạn xem trước nội dung của Field mà bạn chọn và nếu bạn có một danh sách các Field rất dài thì **Find Field** giúp bạn tìm ra một Field nào đó trong danh sách đó.



Hình 2-07: Tab Field dùng để chọn các Field sử dụng

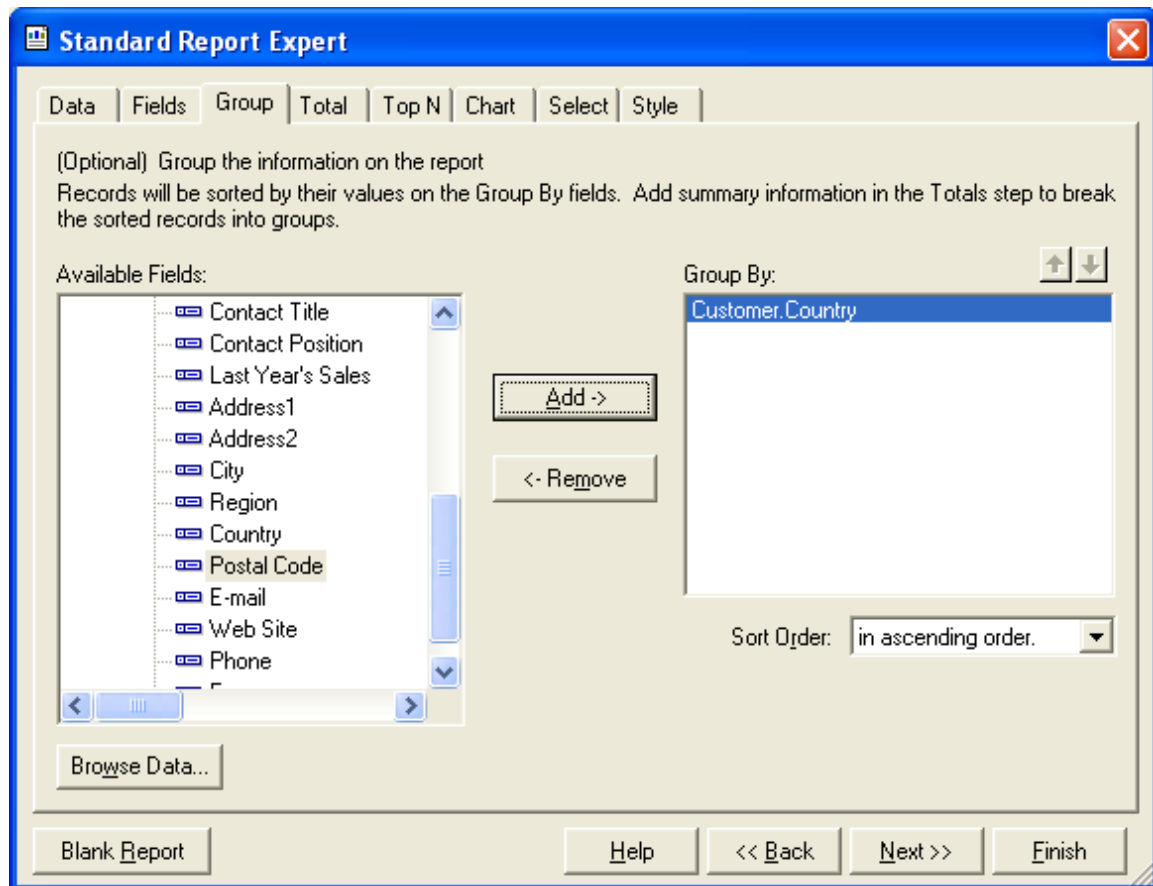
Các Field mà bạn đã chọn ra ở trên sẽ xuất hiện theo thứ tự từ trái qua phải và mỗi mẫu tin trên DataSet sẽ được in ra trên một hàng của báo cáo. Bạn có thể thay đổi thứ tự các Field bằng cách sử dụng hai nút mũi tên lên xuống  trên hộp thoại hình 2-07. Trong ví dụ này, chúng ta chọn ra các Field theo thứ tự: Customer Name, Contact Last Name, Contact Title, E-mail và Last Year's Sales.

Với mỗi Field thì sẽ có một tiêu đề cột tương ứng trên bản báo cáo (mặc định là lấy trùng tên với Field), bạn có thể thay đổi tiêu đề cột ngay tại bước này thông qua ô textbox **Column Heading** trên hình 2-07 hoặc thay đổi trực tiếp ngay trên **Report Designer** sau này.

Sau khi chọn ra các Field dữ liệu, bạn có thể nhấn Finish để làm việc với bản báo cáo trên **Report Designer**. Tuy nhiên, chúng ta sẽ qua bước kế tiếp bằng việc nhấn **Next**.

2.3.5 Gộp nhóm và Sắp xếp (Grouping - Sorting)

Tại bước này, hộp thoại **Standard Report Export** sẽ chuyển sang **Tab Group** (hình 2-08), bạn có thể chọn những Field làm cơ sở sắp xếp và tổng kết theo nhóm bằng cách click lên Field muốn chọn từ khung **Available Fields** sau đó nhấn nút **Add** để chuyển Field chọn qua khung **Group By**. Trong ví dụ này, bạn chọn Group theo `Customer.Country`.



Hình 2-08: Hộp thoại Standard Report Expert – Tab Group

Trên hộp thoại hình 2-08 cũng có hai ô mũi tên lên và xuống cho phép bạn thay đổi thứ tự các Group nếu báo cáo của bạn có nhiều hơn một Group. Ngoài ra ô combobox **Sort Oder** nằm bên dưới khung **Group By** có bốn lựa chọn cho việc sắp xếp được giải thích theo bảng 2-06 dưới đây.

Bảng 2-06 Các lựa chọn sắp xếp

Sort Oder	Mô tả
Ascending	Sắp xếp theo thứ tự tăng dần từ A-Z, 1-9, ...
Decending	Sắp xếp theo thứ tự giảm dần từ Z-A, 9-1, ...
Original	Dữ liệu sẽ có thứ tự theo thứ tự của dataset.
Specified	Dùng để tạo một custom group cho riêng bạn ¹ .

Sau khi chọn xong Field dùng để Group bạn nhấn **Next** để tiếp tục.

2.3.6 Bổ sung thông tin tổng kết

Bạn click lên Tab Total trên hộp thoại **Standard Report Export** (hình 2-09) để thêm thông tin tổng kết (summary) vào báo cáo. Các field tổng kết (summary field) sẽ chứa kết quả của phép tính toán phổ biến nhất như `sum`, `average` ...

Muốn thêm một summary field, bạn cần chọn ra một field trên khung **Available Fields** đưa vào khung **Summaried Fields** (trong ví dụ này là `Customer.Last Year's Sales`), sau đó chọn ra kiểu summary từ hộp combo box **Summary Type** như trên hình 2-09. Trong trường hợp này chúng ta chọn `sum` và check vào ô check box **Add Grand Totals**².

Có tất cả 20 tác tử summary khá phổ biến mà bạn có thể sử dụng³. Tùy theo kiểu dữ liệu của Field bạn chọn mà có thể có hoặc không một vài tác tử summary, ví dụ như bạn không thể tính trung bình (`Average`) trên một Field kiểu string.

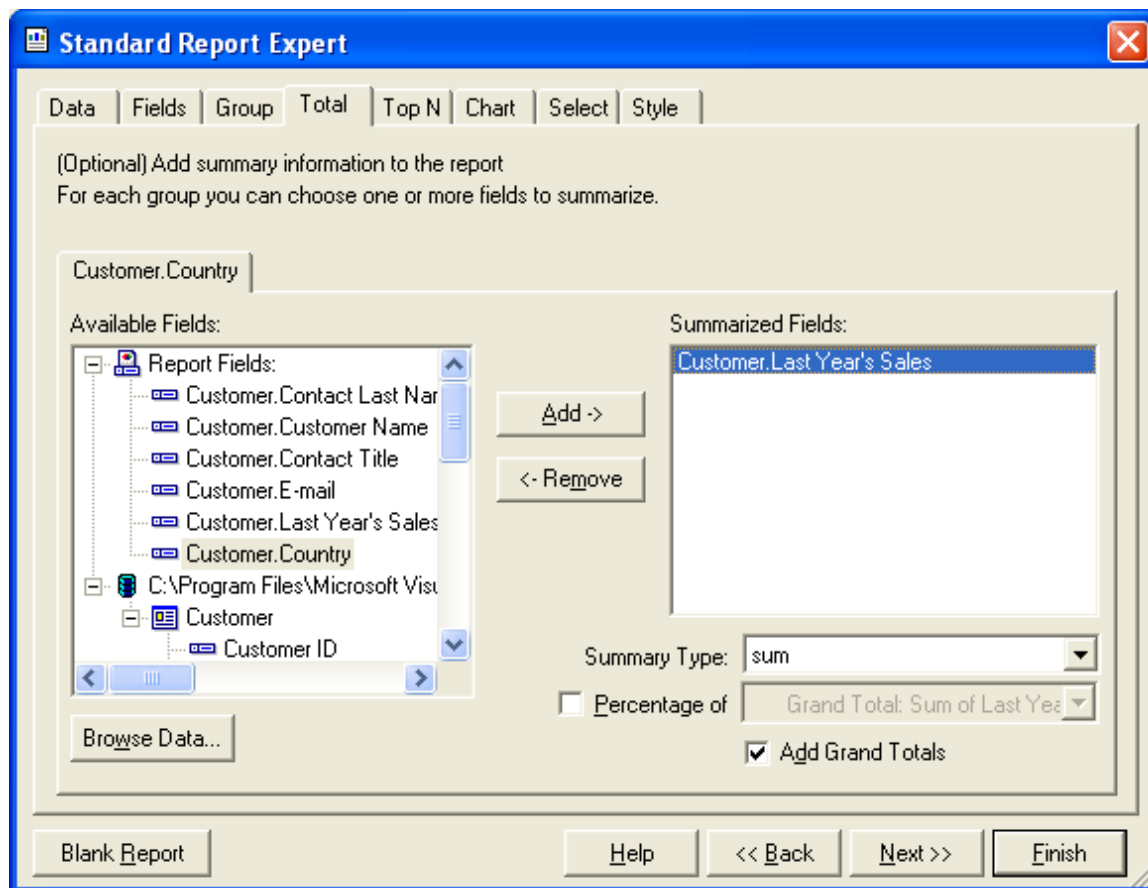
Với ví dụ này đến đây là chúng ta có thể nhấn nút **Finish** để chuyển sang **Report Designer** nhưng để giúp các bạn hiểu rõ hơn các bước còn lại nên chúng ta sẽ tiếp tục khảo sát tất cả các bước còn lại. Bạn nhấn **Next** để qua bước tiếp theo.

Lưu ý: Bạn không thể có bước này nếu báo cáo của bạn không có ít nhất một Group

¹ Specified: xem thêm trong tài liệu tham khảo “Lập trình báo cáo dùng CRYSTAL REPORT .NET và C# - Dương Quang Thiện” trang 75

² Grand Total là tổng cộng cuối cùng, còn gọi là Final Total

³ Bạn tham khảo MSDN với từ khóa SummaryOperation enumeration để biết rõ hơn.

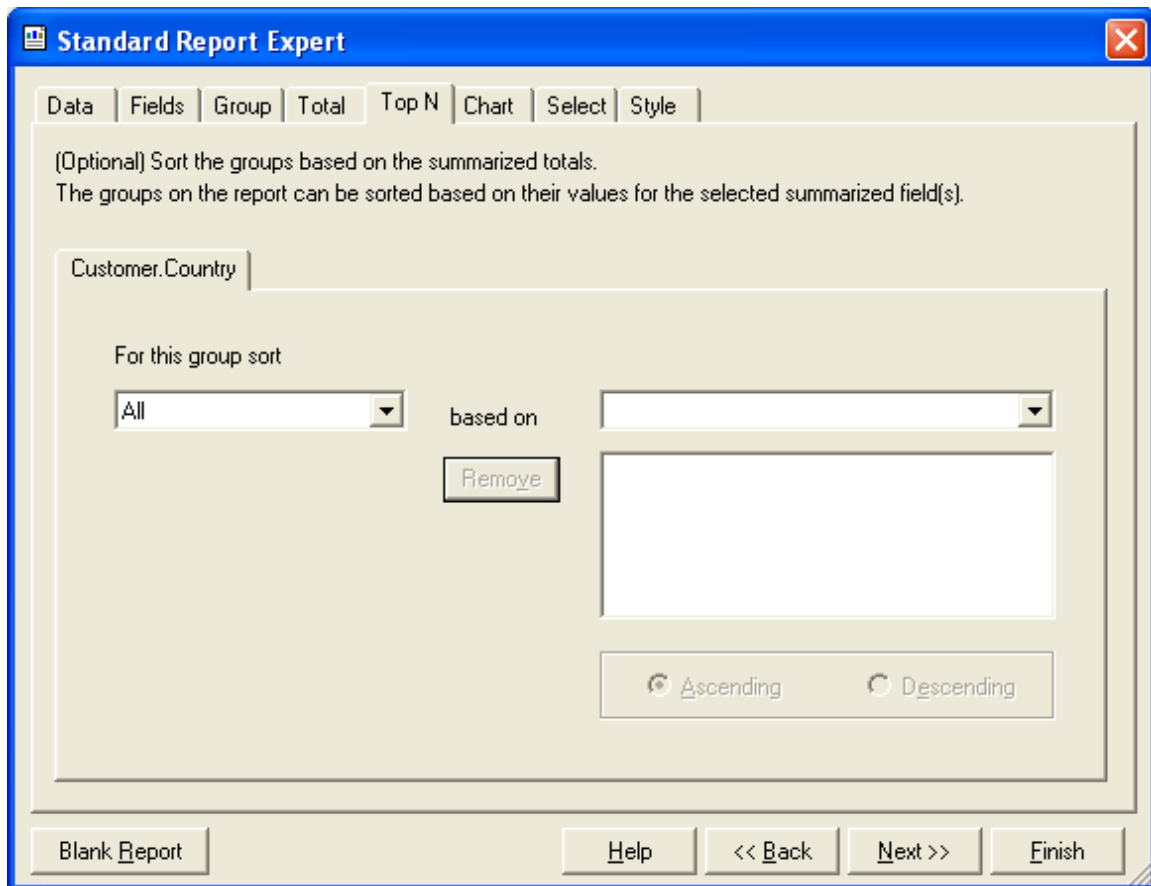


Hình 2-09: Hộp thoại Standard Report Expert – Tab Total

2.3.7 Các tính năng phân tích

Ngoài các summary phổ biến ta còn có khả năng thêm vào bản báo cáo một số chức năng phân tích giúp nhấn mạnh thông tin có thể là quang trọng. Ví dụ, thay vì liệt kê tất cả các nhân viên bán hàng trong công ty bạn có thể in ra danh sách 10 nhân viên nào thực hiện doanh thu cao nhất (để khen thưởng) hay có thể cho ra 10 nhân viên bán hàng tệ nhất trong tháng để lưu ý họ.

Trên hộp thoại **Standard Report Expert** bạn click **Tab TopN** (hình 2-10) để thêm các chức năng phân tích vào bản báo cáo. Bạn để ý có một Tab là `Customer.Country` (nếu bản báo cáo của bạn có bao nhiêu Group thì sẽ có bấy nhiêu Tab, mỗi Tab có tên tương ứng với một Group) và hộp combo **For this group sort** có năm lựa chọn được giải thích theo bảng 2-07.



Standard Report Expert

Data Fields Group Total Top N Chart Select Style

(Optional) Sort the groups based on the summarized totals.
The groups on the report can be sorted based on their values for the selected summarized field(s).

Customer.Country

For this group sort

All based on

Remove

Ascending Descending

Blank Report Help << Back Next >> Finish

Hình 2-10: Tab TopN - Sắp xếp và gộp nhóm dựa trên summarized totals.

Bảng 2-07 Các lựa chọn trên hộp combo For this group sort

Các kiểu phân tích	Mô tả
All	Hiển thị tất cả các giá trị
Top N	Bạn nhập vào một số N vào ô text box where N is . Bản báo cáo sẽ liệt kê N nhóm đầu tiên dựa trên giá trị summary field theo thứ tự giảm dần.
Bottom N	Bạn nhập vào một số N vào ô text box where N is . Bản báo cáo sẽ liệt kê N nhóm cuối cùng dựa trên giá trị summary field theo thứ tự tăng dần.
Top Percentage	Bạn nhập vào tỉ lệ N% (0-100) vào ô text box where Percentage is . Bản báo cáo sẽ liệt kê N% nhóm đầu tiên dựa trên giá trị summary field theo thứ tự giảm dần.
Bottom Percentage	Bạn nhập vào tỉ lệ N% vào ô text box where Percentage is . Bản

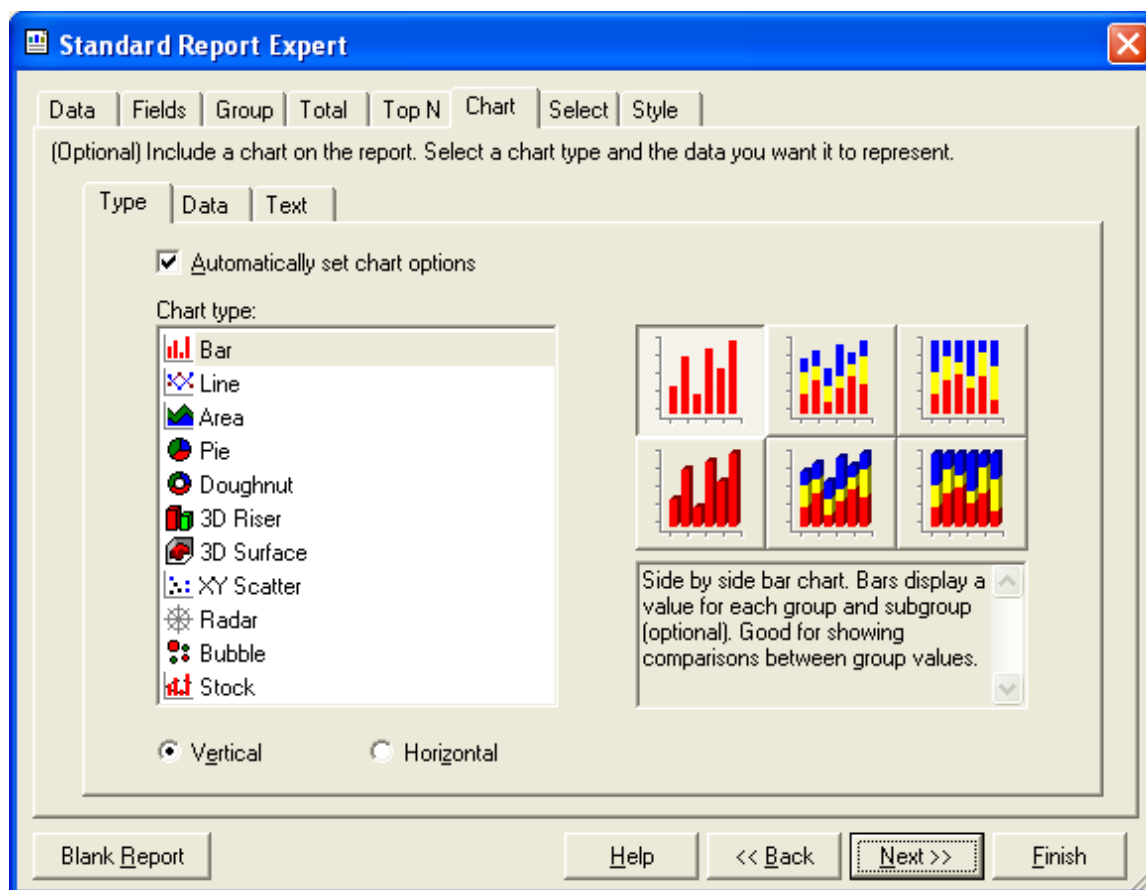
báo cáo sẽ liệt kê N% nhóm cuối cùng dựa trên giá trị summary field theo thứ tự tăng dần.

Với ví dụ này bạn để giá trị `All` trên hộp combo **For this group sort**, nhấn **Next** để tiếp tục.

Lưu ý: Tab TopN chỉ có thể thực hiện được khi bạn đã có sự thiết lập nào đó trên hai Tab Group và Tab Total .

2.3.8 Biểu đồ và Đồ thị (Charting - Graphing)

Trên hộp thoại **Standard Report Export** bạn có thể thêm một số loại Graph khác nhau vào bản báo cáo thông qua **Tab Chart**¹ (Hình 2-11). Bạn nhấn **Next** tiếp tục.



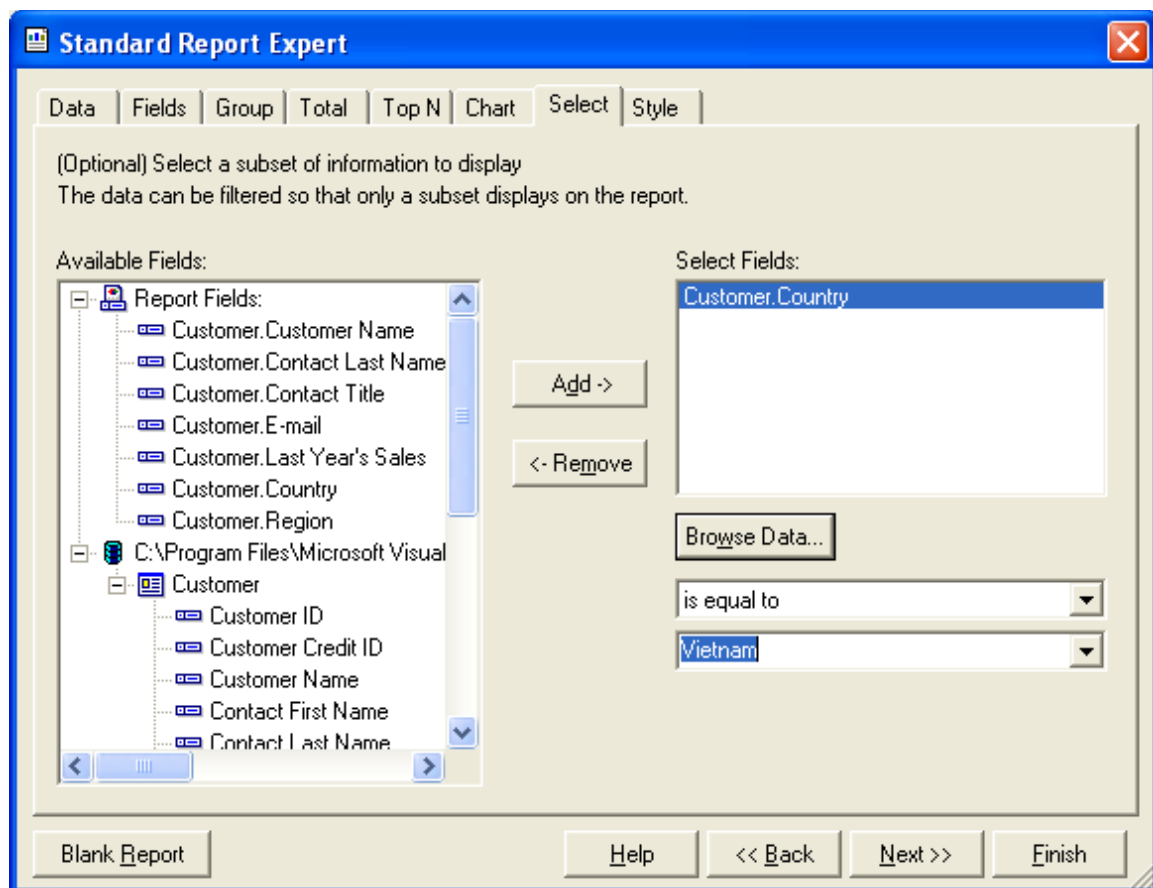
Hình 2-11: Hộp thoại Standard Report Expert – Tab Chart

¹ Muốn tạo bản báo cáo charting và graphing bạn xem thêm tài liệu tham khảo “Lập trình báo cáo dùng CRYSTAL REPORT .NET và C# - Dương Quang Thiện” Chương 8 - Vẽ biểu đồ như thế nào.

2.3.9 Sàng lọc thông tin trên bản báo cáo

Khi in báo cáo từ nhiều bảng dữ liệu khác nhau, chắc chắn là bạn không muốn xem tất cả mọi dữ liệu trên bản báo cáo. Bạn muốn chọn sử dụng những mẫu tin nào đó mà thôi dựa trên những tiêu chí nào đó. **Tab Select** (hình 2-12) sẽ giúp bạn thực hiện ý định này. Có một số tác tử cơ bản có sẵn dùng cho việc chọn lọc mẫu tin được giải thích theo bảng 2-08.

Đối với ví dụ này, ta không quan tâm việc chọn lọc mẫu tin, nghĩa là tất cả các mẫu tin sẽ được xử lý, do đó ta bỏ qua **Tab Select**, bạn nhấn **Next** để qua Tab cuối cùng.



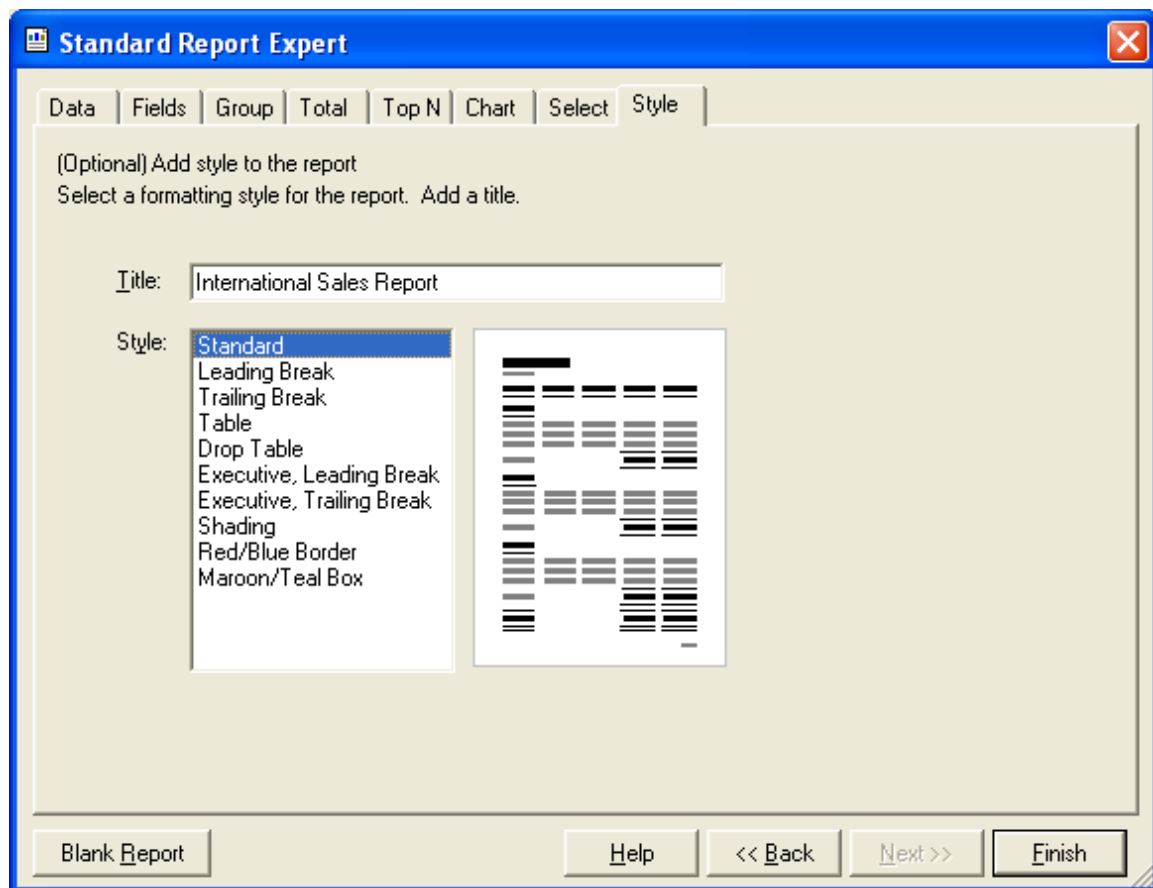
Hình 2-12: Tab Select

Bảng 2-08: Các tác tử chọn lọc cơ bản

Tác tử chọn lọc	Mô tả
Is equal to	So sánh bằng
Is not equal to	So sánh không bằng
Is less than	So sánh nhỏ hơn
Is less than or equal to	So sánh nhỏ hơn hoặc bằng
Is greater than	So sánh lớn hơn
Is greater than or equal to	So sánh lớn hơn hoặc bằng
Is one of	Thuộc một tập hợp định trước
Is between	Nằm giữa
Is not between	Không nằm giữa
Is like	Giống như (dùng tương tự like của SQL)
Is starting with	Bắt đầu bởi

2.3.10 Chọn kiểu định dạng cho báo cáo

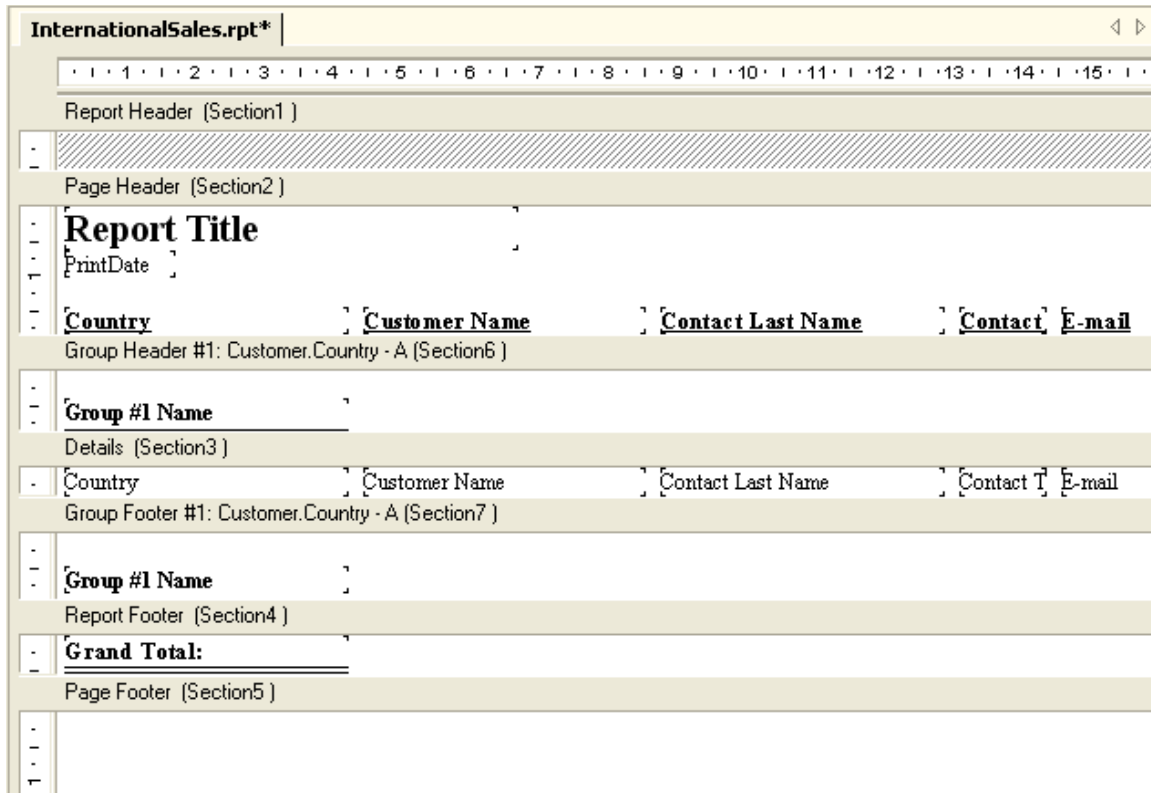
Tab cuối cùng trên **Standard Report Expert** là **Tab Style** (Hình 2-13), giúp bạn chọn lựa format cho bản báo cáo trước khi hoàn tất. Có tất cả 10 style đã được định sẵn mà bạn có thể chọn sử dụng nhưng bạn không thể thêm vào style riêng của bạn. Bạn có thể thêm tựa đề (title) của bản báo cáo vào ô **Title**. Trong trường hợp của chúng ta tựa đề của báo cáo là “International Sales Report”, thông tin này sẽ được lưu trữ trong **Summary Information** của tập tin báo cáo và một **Report Title** Field sẽ được thêm vào **Report Designer**. Bạn nhấn nút Finish để hoàn tất việc thiết kế báo cáo.



Hình 2-13: Hộp thoại Standard Report Expert – Tab Style

2.3.11 Hoàn tất cơ bản việc thiết kế bản báo cáo

Sau khi bạn chọn Finish trên hộp thoại **Standard Report Expert**, một tập tin báo cáo mang tên `InternationalSales.rpt` sẽ được tạo ra với những thiết lập mà bạn vừa mới thực hiện, như theo hình 2-14. Đến đây coi như bạn đã hoàn tất cơ bản việc thiết kế bản báo cáo.



Hình 2-14: Tập tin báo cáo InternationalSales.rpt

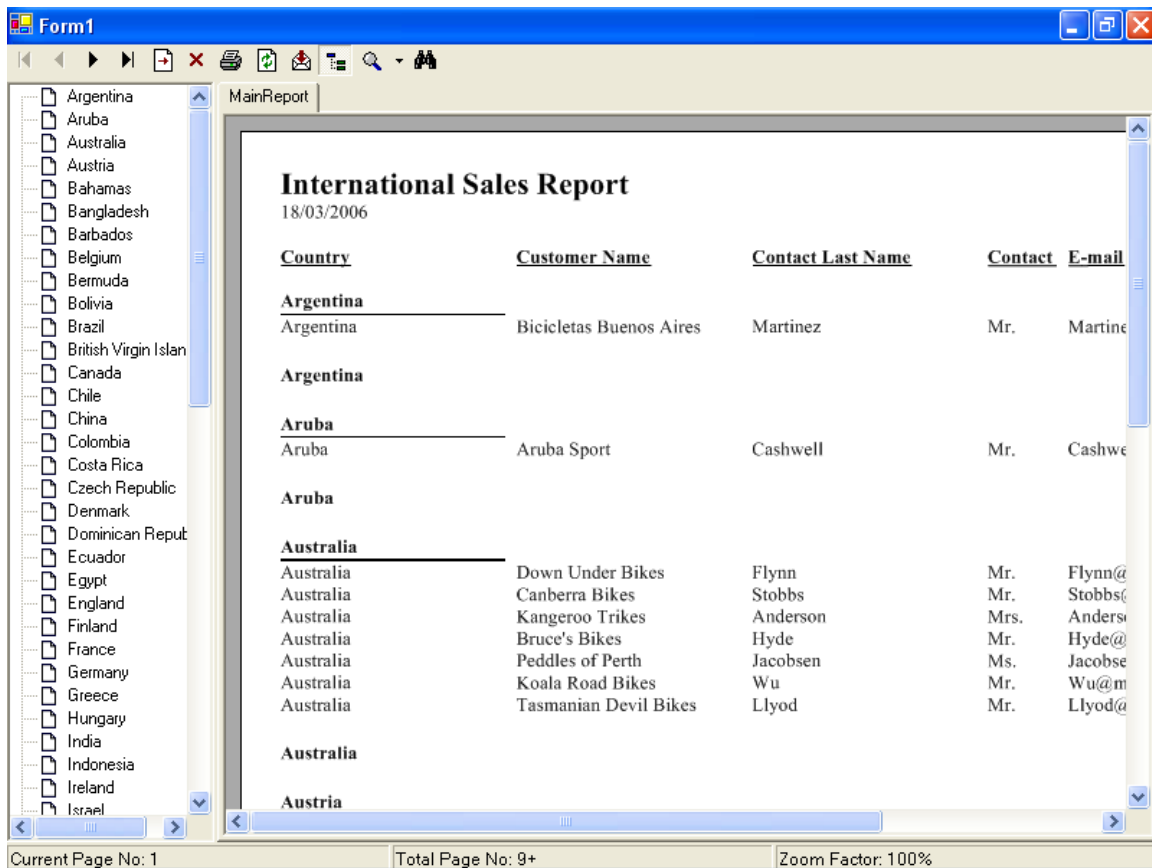
2.4 Preview bản báo cáo

Muốn in ra bản báo cáo InternationalSales.rpt bạn thêm vào Form1 ô control **CrystalReportViewer** trên Toolbox. Khi đó IDE tự động tạo ra một đối tượng `crystalReportViewer1`, bạn gán thuộc tính `Dock` của đối tượng này thành `Fill` để ô control viewer này bao phủ toàn bộ vùng Client trên `Form1`. Đồng thời bạn thay đổi thuộc tính `WindowState` của `Form1` thành `Maximized`.

Tại hàm xử lý sự kiện `Form_Load` bạn bổ sung dòng lệnh in đậm sau đây:

```
private void Form1_Load(object sender, System.EventArgs e)
{
    crystalReportViewer1.ReportSource = new InternationalSales();
}
```

Lệnh trên tạo một đối tượng `InternationalSales` (là bản báo cáo của bạn vừa thiết kế xong) và gán cho thuộc tính `ReportSource` của ô control viewer. Khi bạn cho chạy ứng dụng, nó sẽ cho thấy một viewer của bản báo cáo như hình 2-15.



Form1

MainReport

International Sales Report
18/03/2006

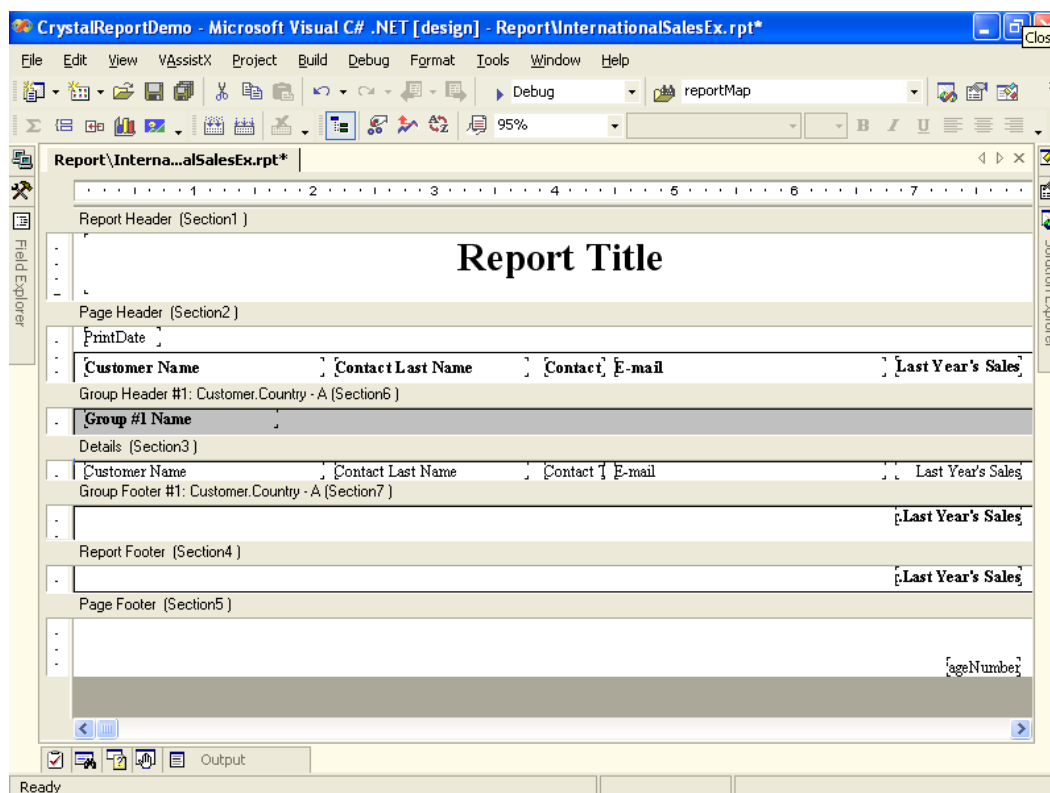
Country	Customer Name	Contact Last Name	Contact	E-mail
Argentina				
Argentina	Bicicletas Buenos Aires	Martinez	Mr.	Martine
Aruba				
Aruba	Aruba Sport	Cashwell	Mr.	Cashwe
Australia				
Australia	Down Under Bikes	Flynn	Mr.	Flynn@
Australia	Canberra Bikes	Stobbs	Mr.	Stobbs@
Australia	Kangaroo Trikes	Anderson	Mrs.	Anders
Australia	Bruce's Bikes	Hyde	Mr.	Hyde@
Australia	Peddles of Perth	Jacobsen	Ms.	Jacobse
Australia	Koala Road Bikes	Wu	Mr.	Wu@m
Australia	Tasmanian Devil Bikes	Llyod	Mr.	Llyod@
Austria				

Current Page No: 1 Total Page No: 9+ Zoom Factor: 100%

Hình 2-15: Báo cáo InternationalSales ở chế độ Preview

2.5 Tinh chỉnh và hoàn thiện bản báo cáo

Qua ví dụ trên, bạn dễ dàng tạo ra một bản báo cáo mà không cần mất nhiều thời gian, tuy nhiên báo cáo này rất thô thiển, chưa có gì là tinh tế. Phần này sẽ giúp bạn hoàn thiện bản báo cáo của mình, làm cho bản báo cáo trở nên dễ nhìn hơn bằng cách thêm vào một vài chi tiết nhỏ hay bỏ đi một số chi tiết không cần thiết. Kết quả sau khi tinh chỉnh sẽ cho ta bản báo cáo trên hình 2-15 và kết quả Preview trên hình 2-16.




Hình 2-15 File báo cáo sau khi tinh chỉnh

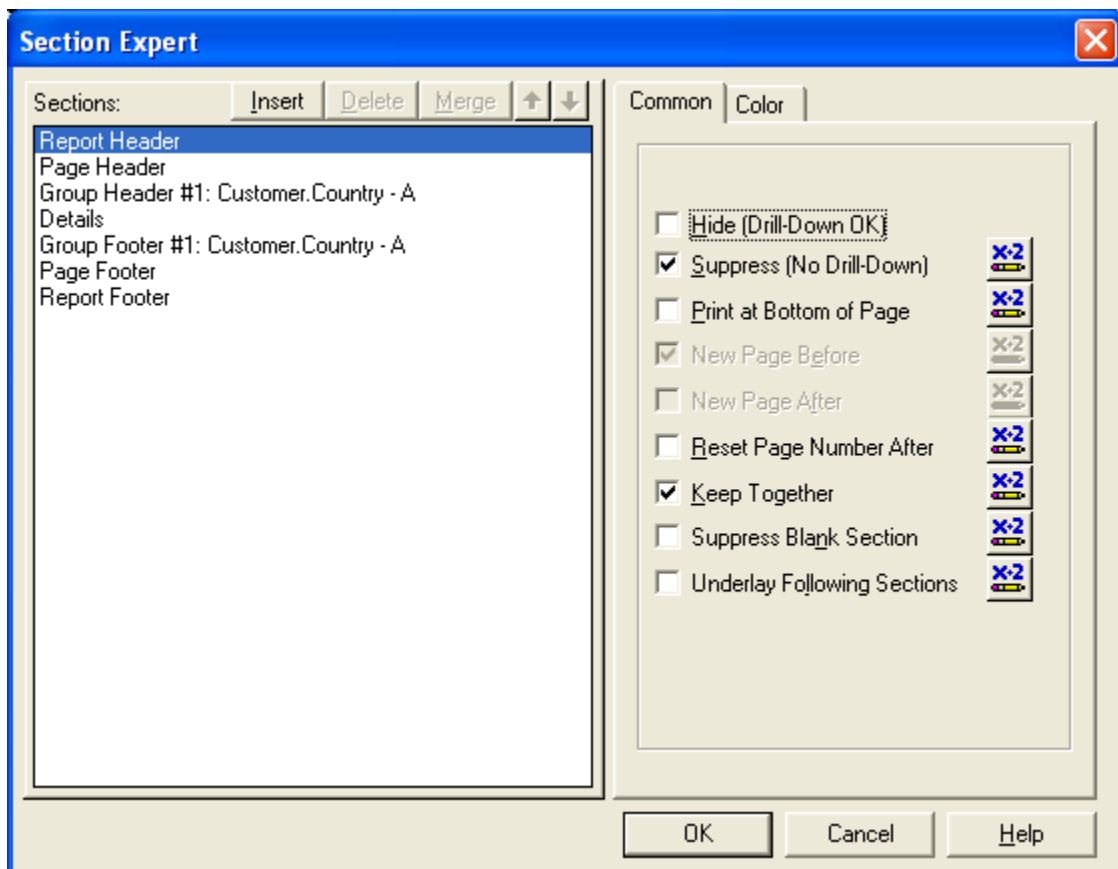


Hình 2-16: Kết quả preview của file báo cáo sau khi tinh chỉnh

2.5.1 Định dạng các phân đoạn

Xem lại hình 2-14 bạn thấy phần **Report Header** được đánh dấu chéo, điều này có nghĩa là sẽ không được hiển thị, vậy bây giờ ta sẽ định dạng lại cho **Report Header** section và xác lập màu nền cho **Group Header #1** section là màu xám như hình 2-15.

Bạn right click lên **Report Designer** và chọn **Format Section** để hiển thị hộp thoại **Section Expert** như hình 2-17. Trong khung **Sections** bên tay trái sẽ liệt kê tất cả các section trên bản báo cáo và bên tay phải có hai **Tab Common** và **Color** chứa những ô checkbox dùng để định dạng cho một section được giải thích theo bảng 2-09. Cạnh bên mỗi ô checkbox có một nút  được gọi là **Formular Button**. Nhấn lên nút này sẽ hiển thị hộp thoại **Formular Editor** cho phép bạn tạo ra một công thức, điều này sẽ được đề cập sau.



Hình 2-17: Hộp thoại Section Expert

Bảng 2-09: Các thuộc tính định dạng section

Thuộc tính định dạng	Mô tả
Hide (Drill-Down OK)	Không cho hiển thị section, nhưng cho phép Drill down vào chi tiết dữ liệu.
Suppress (No Drill-Down)	Không cho hiển thị section và cũng không cho phép Drill down vào chi tiết dữ liệu.
Print at Bottom of Page	Làm cho section bao giờ cũng in ra vào cuối trang.
New Page Before	Page break - sang trang - trước khi section được in ra.
New Page After	Page break - sang trang - sau khi section được in ra.
Reset Page number After	Cho chỉ số trang về 1 sau khi in ra section.
Keep Together	Giữ cho section in ra trên cùng 1 trang, không cho in qua trang sau.
Suppress Blank Section	Nếu không có dữ liệu trong section thì không in nó ra.
Underlay Following Sections	Xem MSDN.
Format with Multiple Columns	Xem MSDN.

Đối với bản báo cáo của chúng ta, bạn hãy uncheck hai thuộc tính **Hide (Drill-Down OK)** và **Suppress (No Drill-Down)** đối với **Report Header** section, và đối với **Group Header #1** section bạn click lên **Tab Color** và đánh dấu check vào ô **Background Color** sau đó bạn có thể chọn màu tùy ý trên hộp combobox, ở đây chúng tôi chọn là màu **Silver**.

Đến đây bạn hãy loại bỏ một số Field không cần thiết và bổ sung **Report Title** Field lên phần **Report Header** bằng cách cho bung nhánh **Special Fields** trên khung **Field Explorer** chọn **Report Title** kéo và thả vào **Report Header** section. Sau đó bạn sắp xếp lại vị trí các Field theo hình 2-15.

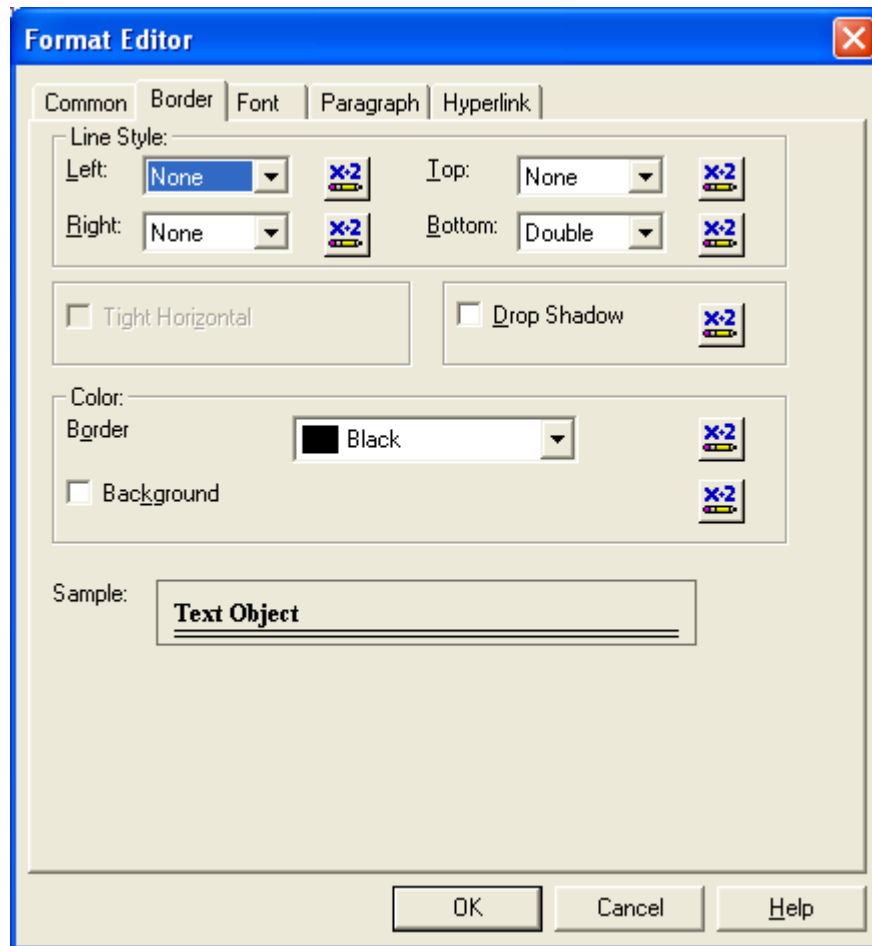
Nếu khi di chuyển các Field đến vị trí mới, bạn có thể làm cho các Field không còn thẳng hàng với nhau nữa hay kích thước không còn bằng nhau. Để khắc phục lỗi này bạn chọn các Field bị lệch trước, sau cùng bạn chọn một Field làm chuẩn right click lên Field chuẩn này rồi chọn **Align** nếu bạn muốn canh hàng - cột hoặc **Size** nếu bạn muốn canh lại kích thước.

2.5.2 Định dạng các Field:

Sau khi thiết kế báo cáo, trên **Report Deport** có một số Field dạng tổng hợp kết quả như **Grand Total** Field (hình 2-14) có định dạng mặc định trông không được đẹp mắt, bạn

có thể định lại bất kỳ Field nào mà bạn muốn bằng cách right click lên Field đó và chọn **Format** để hiển thị hộp thoại **Format Editor** như hình 2-18.

Trong trường hợp này, đối với những Field tương tự như Grand Total Field nói trên bạn click vào **Tab Border** trên hộp thoại **Format Editor**, trong khung **Line Style** bạn thiết lập giá trị **None** cho các ô control **Left**, **Top**, **Right** và **Bottom**.



Hình 2-18: Hộp thoại Format Editor

2.5.3 Hiệu chỉnh Sắp xếp

Bạn hãy để ý khi preview bản báo cáo của chúng ta hiện tại đã được gộp nhóm và sắp thứ tự theo `Customer.Contry`, tuy nhiên nội dung trong lòng mỗi nhóm lại không có thứ

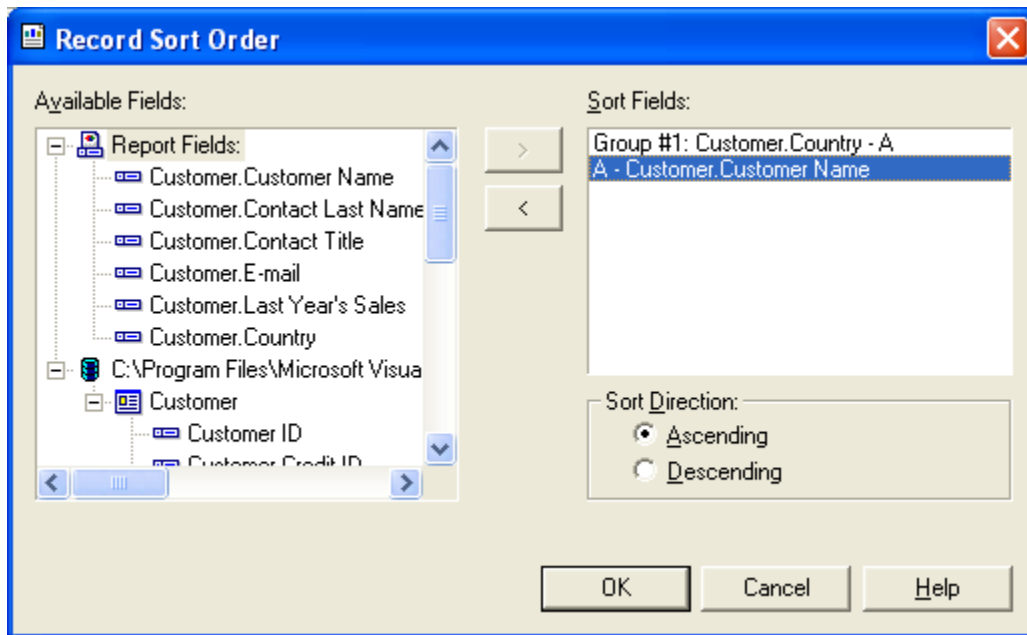
tự do đó ta sẽ sử dụng hộp thoại **Record Sort Order** để bổ xung các sắp xếp trong lòng mỗi nhóm.

Muốn hiển thị hộp thoại này bạn right click lên **Report Designer** rồi chọn **Report | Sort Record**. Hình 2-19 cho thấy bản báo cáo hiện đang được sắp xếp theo Field: `Customer.Contry`, và bạn không thể tác động gì lên Field này được vì nó được tạo ra khi ta chọn Group ở bước trước đó, muốn hiệu chỉnh Field này ta sẽ có cách làm khác.

Muốn chọn một Field để sắp xếp bạn click một item trên khung **Available Fields** bên trái rồi nhấn nút mũi tên “>” để chuyển item đó qua khung **Sort Fields** bên phải. Thứ tự bạn thêm các Field sẽ ấn định mức độ ưu tiên sắp xếp. Ngược lại nếu không muốn sắp xếp thì bạn sử dụng nút mũi tên “<” để gỡ bỏ một Field khỏi khung **Sort Fields**.

Phía dưới khung **Sort Fields** có frame **Sort Direction**, với 2 lựa chọn **Ascending** (tăng) và **Decending** (giảm) giúp bạn chọn thứ tự sắp xếp cho Field tương ứng trong khung **Sort Fields**.

Trong ví dụ này, chúng ta sẽ bổ xung sắp xếp theo Field `Customer.Name` và theo thứ tự tăng dần như hình 2-19.

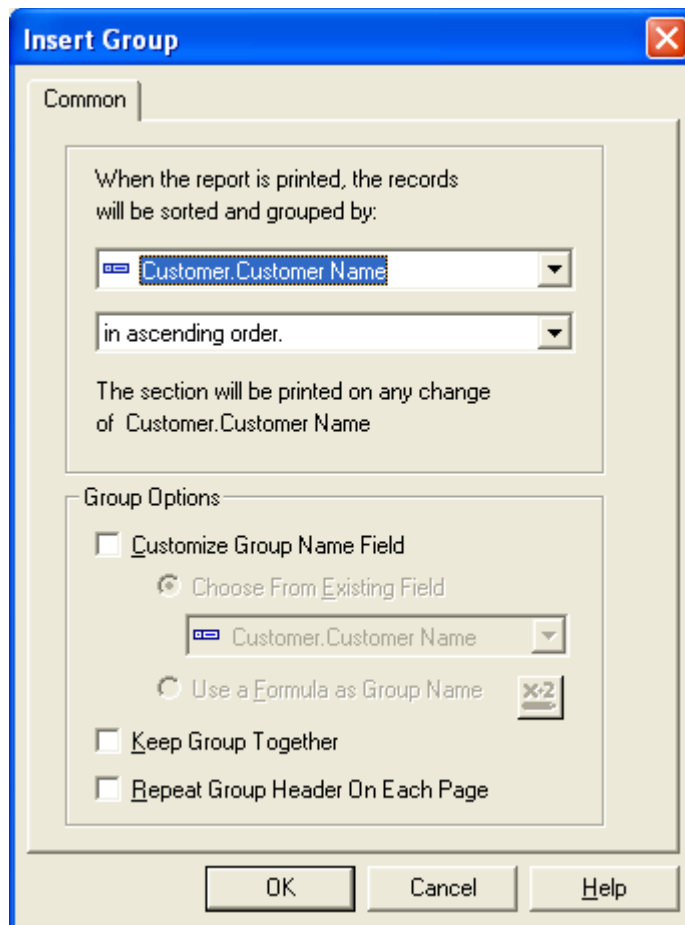


Hình 2-19: Hộp thoại Record Sort Oder

2.5.4 Hiệu chỉnh Gộp nhóm

Mặc dù bản báo cáo cũng chúng ta sẽ không có hiệu chỉnh Group nhưng phần này sẽ giúp bạn hiệu chỉnh trên các Group hiện hành cũng như bổ sung thêm Group vào bản báo cáo nếu cần thiết.

Muốn thêm một nhóm lên bản báo cáo, bạn right click lên bản báo cáo, rồi chọn **Insert | Group**. Hộp thoại **Insert Group** (hình 2-20) sẽ hiển thị với một **Tab Common** duy nhất.

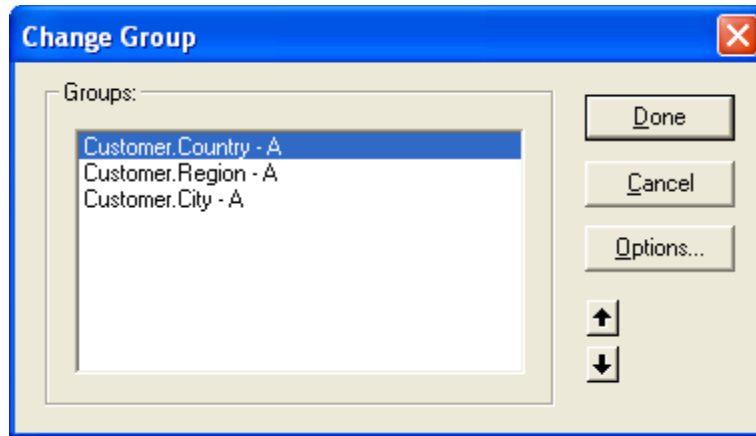


Tab Common có 2 frame. Frame đầu tiên, với ô combobox đầu tiên để xác định Field dùng để gộp nhóm và ô combobox thứ hai để xác định thứ tự sắp xếp.

Frame còn lại **Group Options** cho phép định dạng nhóm. Nếu bạn đánh dấu check vào ô **Group Together** thì nếu cả nhóm không nằm vừa trên phần còn lại của trang thì sẽ để trống và nhóm sẽ được in bắt đầu từ trang sau. Còn ô checkbox **Repeat Group On Each Page**, cho biết liệu group header có phải in lặp lại trên mỗi trang hay không.

Hình 2-20: Hộp thoại Insert Group

Khi bạn thêm một nhóm vào, thì mặc định nhóm này sẽ được nằm trong cùng so với các nhóm hiện hữu, muốn thay đổi thứ tự này cũng như thay đổi các giá trị khác của nhóm bạn right click lên bản báo cáo chọn **Report | Change Group Expert** để hiển thị hộp thoại **Change Group** như hình 2-21.

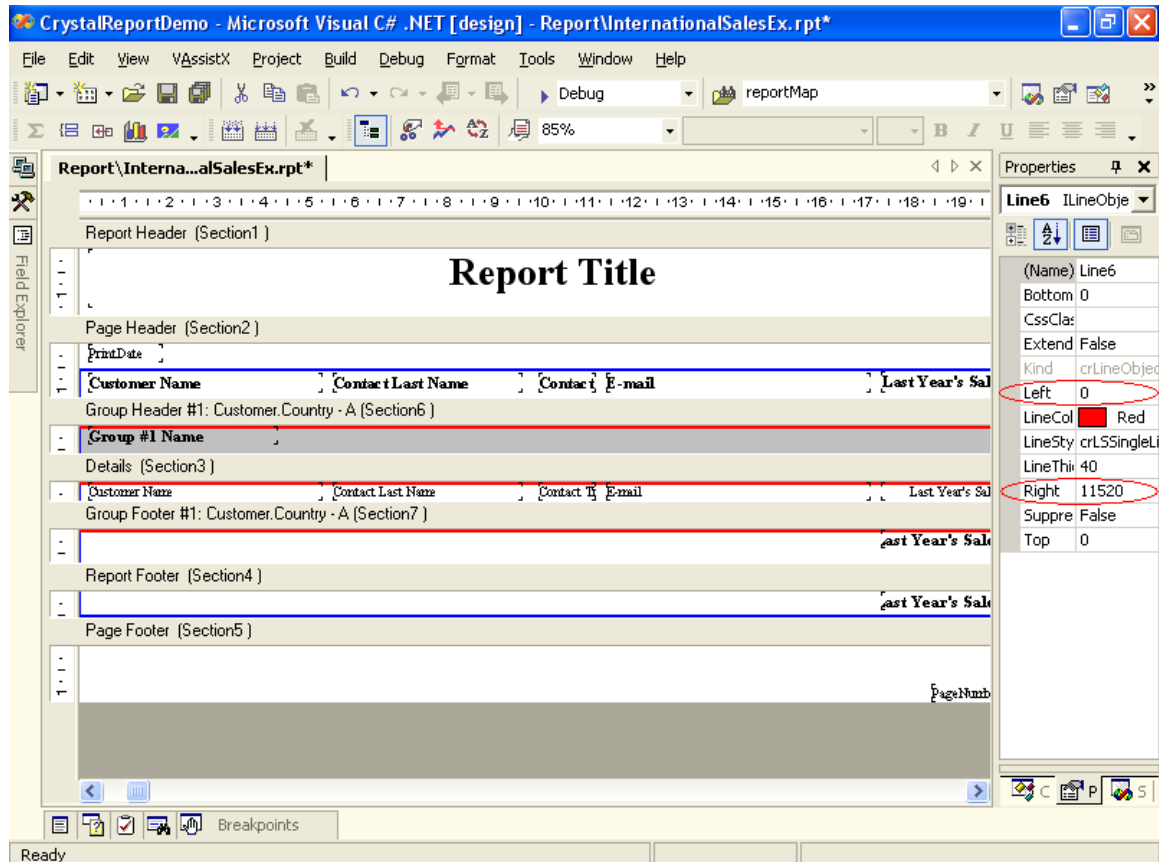


Hình 2-21: Hộp thoại Change Group

Trên hộp thoại này, các nhóm được liệt kê theo thứ tự hiện hành, bạn có thể sử dụng hai mũi tên lên xuống để sắp xếp lại thứ tự. Còn muốn thay đổi một nhóm, thì bạn nhấn nút Options để hiển thị hộp thoại Change Group Options tương tự như hộp thoại Insert Group.

2.5.5 Vẽ khung

Sau khi hoàn tất hai phần **2.5.1** và **2.5.2** thì bản báo cáo của bạn đã gần giống như trên hình 2-22. Trong phần này sẽ giúp bạn bổ xung các đối tượng Box và Line vào bản báo cáo. Đây là phần thường làm cho người mới thiết kế gặp nhiều lúng túng bởi vì nó liên quan đến nhiều section của báo cáo, đòi hỏi bạn phải hiểu rõ sự bố trí và thể hiện nội dung của các section trên bản báo cáo đã được đề cập đến trong phần **2.2.1 “Các section của một báo cáo”**.



Hình 2-22: File báo cáo sau khi tinh chỉnh
với các đối tượng Box và Line được làm nổi bật

Lưu ý: Trên hình 2-15 màu xanh da trời và màu đỏ chỉ để giúp các bạn nhìn rõ các đối tượng Box và Line, thực tế chúng có màu đen và có độ rộng là 1.

Trước hết ta sẽ bổ sung đối tượng **Box** vào bản báo cáo. Bạn right click lên **Report Designer** chọn **Insert | Box**, khi đó con trỏ chuột sẽ thay đổi thành hình dạng cây bút, bạn click và kéo để tạo ra một hình chữ nhật, bạn di chuyển và thay đổi kích thước của hình chữ nhật này sao cho nó bao tất cả các Field thuộc các section: **Page Header**, **Group Header #1**, **Details**, **Group Footer #1** và **Report Footer** ngoại trừ **PrintDate** Field nằm trong **Page Header** section như hình 2-22 (Box có màu xanh da trời). Sau đó bạn cho chạy ứng dụng để preview xem có đúng như ý bạn mong muốn hay không trước khi ta bổ xung các đối tượng **Line**. Đến đây ta vẫn chưa gặp trở ngại gì cả.

Tiếp theo ta sẽ bổ xung các đối tượng **Line** vào bản báo cáo để phân cách các mẫu tin ra thành từng hàng riêng biệt bằng cách right click lên **Report Designer** chọn **Insert | Line**. Ta sẽ bổ sung 3 đường thẳng lên bản báo cáo.

Đường thẳng thứ nhất dùng để phân cách giữa phần **Page Header** và **Group Header** như hình 2-22 (Line là đường màu đỏ), bạn chỉ cần kéo một đường thẳng sao cho nó nằm trùng với cạnh trên của vùng nội dung của **Group Header** section, bạn đừng quan tâm cạnh 2 biên trái phải của đường thẳng ta có cách giải quyết sau.

Để canh 2 biên trái và phải của đường thẳng sao cho trùng với biên của đối tượng Box có hai cách. Cách 1 bạn di chuyển con trỏ đến đầu mỗi biên và kéo để thay đổi theo ý bạn. Cách 2 bạn thay đổi giá trị 2 thuộc tính **Left** và **Right** của đường thẳng trên khung **Properties** sao cho bằng với giá trị **Left** và **Right** của đối tượng Box như hình 2-22. Sau đó bạn preview lại xem có đúng ý mình chưa.

Bạn tiếp tục bổ sung hai đường thẳng còn lại như trên hình 2-22 theo cách làm trên để hoàn chỉnh cho bản báo cáo.

Lưu ý:

Khi vẽ các đường thẳng và preview lại có thể bạn thấy không như ý mình thì lý do có thể là bạn để sai vị trí ví dụ như thay vì để trên phần top của vùng nội dung thuộc section thì bạn lại cho nằm ở bottom hay thay vì để ở section này bạn lại để ở section khác. Do đó bạn cứ thử đặt đường thẳng vào một vị trí khác và so sánh kết quả với lần preview trước đó rồi từ đó đưa ra vị trí hợp lý.

Để dễ dàng cho việc vẽ trên **Report Designer** bạn nên zoom kích thước ra khoảng 200% để định vị vị trí cho chính xác hơn.

Chương 3 : Công thức Basic Syntax

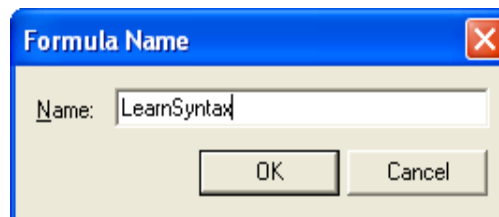
Rất ít bản báo cáo chỉ đơn thuần liệt kê thông tin - phần lớn sẽ bao gồm ít nhất một vài tính toán cơ bản, tổng kết và phần logic chương trình. Chắc hẳn bạn không muốn có một hóa đơn mà không có tính thuế GTGT và hàng tổng cộng cuối hoá đơn hoặc tổng kết doanh thu mà không có tổng doanh thu. Phần logic chương trình sẽ được đưa vào báo cáo, thông qua các công thức, tùy thuộc vào yêu cầu do người sử dụng đặt ra.

Crystal Reports cho bạn chọn lựa sử dụng hoặc Crystal Syntax hoặc Basic Syntax. Trong tài liệu này chúng tôi chỉ đề cập đến Basic Syntax vì nó tương tự như cú pháp Visual Basic.NET mặc khác giữa Crystal Syntax và Basic Syntax cũng có nhiều điểm tương đồng, cho nên nắm rõ Basic Syntax thì bạn cũng có thể dễ dàng tự tìm hiểu Crystal Syntax.

3.1 Hộp thoại *Formula Editor*

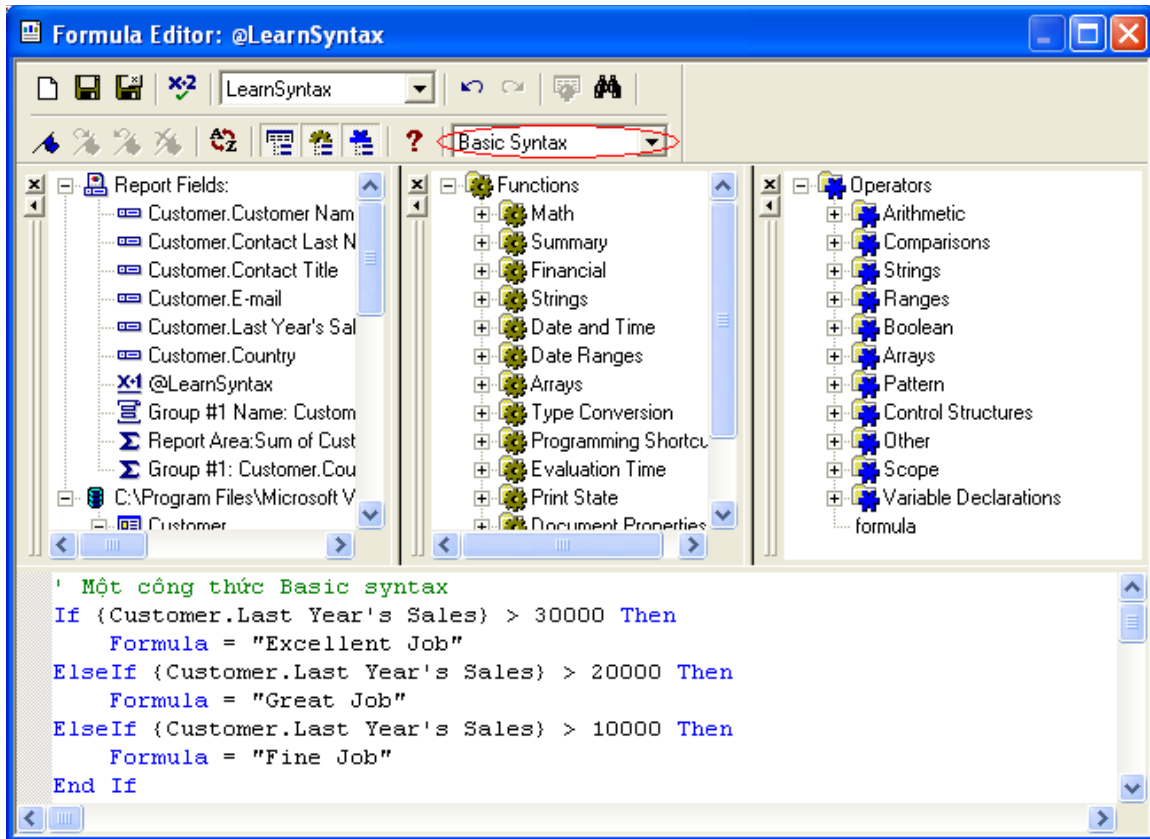
Hộp thoại Formula Editor có thể được hiển thị khi bạn muốn tạo mới một công thức hoặc hiệu chỉnh một công thức hiện có.

Để tạo mới một công thức bạn right click lên **Formula Fields** trong cửa sổ **Field Explorer** sau đó chọn **New** để hiển thị hộp thoại **Formula Name** yêu cầu bạn nhập vào tên của công thức (ví dụ như `LearnSyntax`) như hình 3-01.



Hình 3-01: Một công thức mới

Sau khi bạn nhập vào tên công thức và nhấn **OK**, hộp thoại **Formula Editor** (hình 3-02) sẽ hiển thị để bạn soạn thảo nội dung của công thức.



Hình 3-02: Hộp thoại Formula Editor

3.1.1 Các thành phần chính

Quan sát hộp thoại **Formula Editor** (hình 3-02), trên hàng đầu tiên của thanh công cụ có một ô combobox chứa chuỗi **LearnSyntax** cho biết công thức mà bạn đang soạn thảo. Nếu bạn click lên mũi tên của ô combobox thì bạn sẽ thấy danh sách các formula field hiện có trên bản báo cáo. Trên hàng thứ hai của thanh công cụ, cũng có ô combobox chứa hai giá trị là **Basic Syntax** và **Crystal Syntax** tương ứng với hai loại ngôn ngữ.

Lưu ý: Do chúng ta chỉ thực hiện với ngôn ngữ **Basic Syntax** nên bạn cần phải chọn lại ngôn ngữ này vì **Crystal Syntax** là ngôn ngữ mặc định của **Formula Editor**.


Bên dưới 2 thanh công cụ là 3 cửa sổ **Field Tree**, **Function Tree** và **Operator Tree**. Cả 3 cửa sổ này đều được thiết kế khi bạn double click lên một item nào đó trên một trong 3 cây thì item này sẽ được chèn vào cửa sổ soạn thảo bên dưới ngay tại vị trí con nháy. Như vậy bạn có thể viết phần lớn các công thức mà không cần phải nhập vào. Ba cửa sổ trên được mô tả như sau.

- Cửa sổ **Field Tree** hiển thị tất cả các field hiện có trên bản báo cáo hiện hành. Bao gồm các field trên bảng dữ liệu, các formula field, các total field, các group field và cả các cửa dữ liệu nguồn mặc dù chúng không xuất hiện trên bản báo cáo. Bạn có thể sử dụng tất cả các field được liệt kê ở đây vào trong công thức.
- Cửa sổ **Function Tree** nằm ở giữa chứa các công thức có sẵn mà phần lớn các hàm này thì tương tự như các hàm của Visual Basic .NET.
- Cửa sổ **Operator Tree** cho bạn thấy phần lớn các toán tử quen thuộc.

Cuối cùng là cửa sổ Formula, nằm ở bên dưới hộp thoại Formula Editor, là nơi bạn viết ra nội dung công thức. Các lệnh sẽ được hiển thị với những màu khác nhau: các từ khóa sẽ mang màu xanh, các chú giải sẽ có màu lục và các hằng sẽ là màu đen.

Khi bạn viết xong công thức, bạn cần kiểm tra cú pháp xem có đúng chưa. Nếu đúng thì bạn nhấn nút **Save** trên thanh công cụ để lưu lại công thức.

3.1.2 Kiểm tra lỗi cú pháp

Để kiểm tra công thức của bạn xem có lỗi cú pháp hay không, bạn có thể sử dụng nút Check có hình  trên thanh công cụ.

- Nếu không có lỗi bạn sẽ nhận được thông báo như hình 3-03(a).
- Nếu có lỗi bạn sẽ nhận được thông báo tương ứng với lỗi đầu tiên được tìm thấy, và con nháy sẽ nằm ngay vị trí gây ra lỗi này.
- Nếu bạn công thức của bạn có lỗi và bạn nhấn nút Save thì bạn sẽ nhận được thông báo như hình 3-03(b).

Hình 3-03: Một số thông báo kiểm tra cú pháp

3.2 Định dạng động sử dụng công thức

Theo bản chất các bản báo thường mang tính tĩnh, mặc dù dữ liệu in ra là có thay đổi nhưng những định dạng ban đầu thì không thay đổi. Ví dụ, nếu Field trên một cột nào đó có font chữ là Arial và màu đen thì mọi mẫu tin trong cột này đều mang phong chữ là Arial đen. Nếu bạn muốn thay đổi màu chữ (màu đỏ chẳng hạn) khi giá trị doanh thu dưới một mức báo động nào đó để người đọc báo cáo chú ý. Trong trường hợp này, làm cho báo cáo sống động hơn sẽ tăng sự hữu ích của bản báo cáo mà không gây nặng nề cho người đọc.

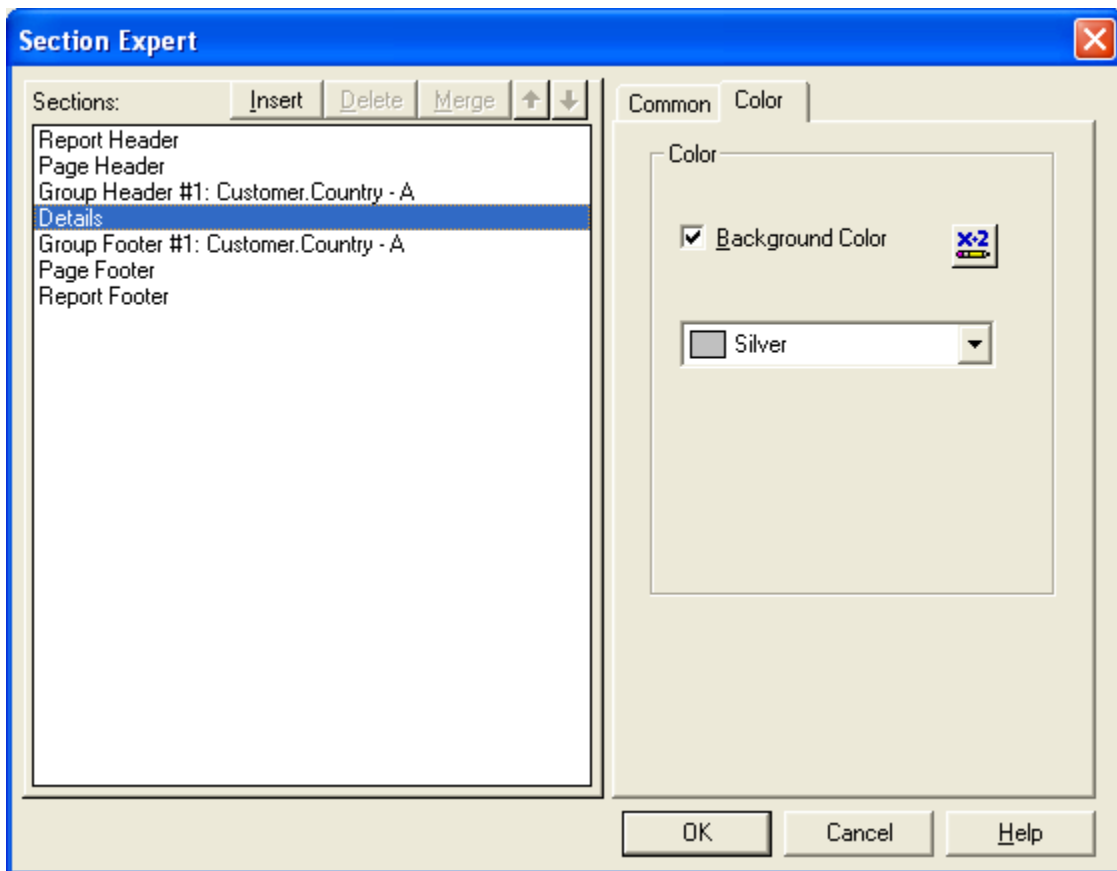
Crystal Reports cung cấp cho bạn nhiều cách khác nhau làm cho bản báo cáo sinh động hơn. Nó cho phép bạn có khả năng sử dụng công thức để thay đổi thuộc tính của các field và các section trên một bản báo cáo.

Khi thêm một công thức vào một thuộc tính, bạn phải xác định kiểu dữ liệu mà thuộc tính này sử dụng để công thức trả về kiểu dữ liệu tương ứng. Nếu bạn trả về một kiểu dữ liệu không tương thích thì Formula Editor sẽ thông báo lỗi khi bạn cố lưu trữ công thức.


Có nhiều kiểu dữ liệu, từ kiểu Boolean ứng với các thuộc tính được thể hiện dưới dạng checkbox, hoặc là các hằng số được định nghĩa sẵn trước ứng với các thuộc tính được thể hiện dưới dạng combobox, đến các kiểu dữ liệu đơn giản như string hoặc number.

Quay lại bản báo cáo `InternationalSales.rpt` đã được đề cập ở **Chương 2**, chúng ta sẽ tiếp tục bổ xung định dạng bằng công thức vào bản báo cáo này. Chúng ta sẽ định dạng màu nền cho một hàng trên bản báo cáo dựa trên giá trị `Customer.Last Year's Sales`.

Mở bản báo cáo `InternationalSales.rpt` sau đó bạn right click lên bản báo cáo và chọn `FormatSection`. Trong hộp thoại `Section Expert` bạn chọn section `Details` bên tay trái và click vào Tab `Color` bên tay phải như hình 3-04.





Hình 3-04: Hộp thoại Section Expert với Tab Color

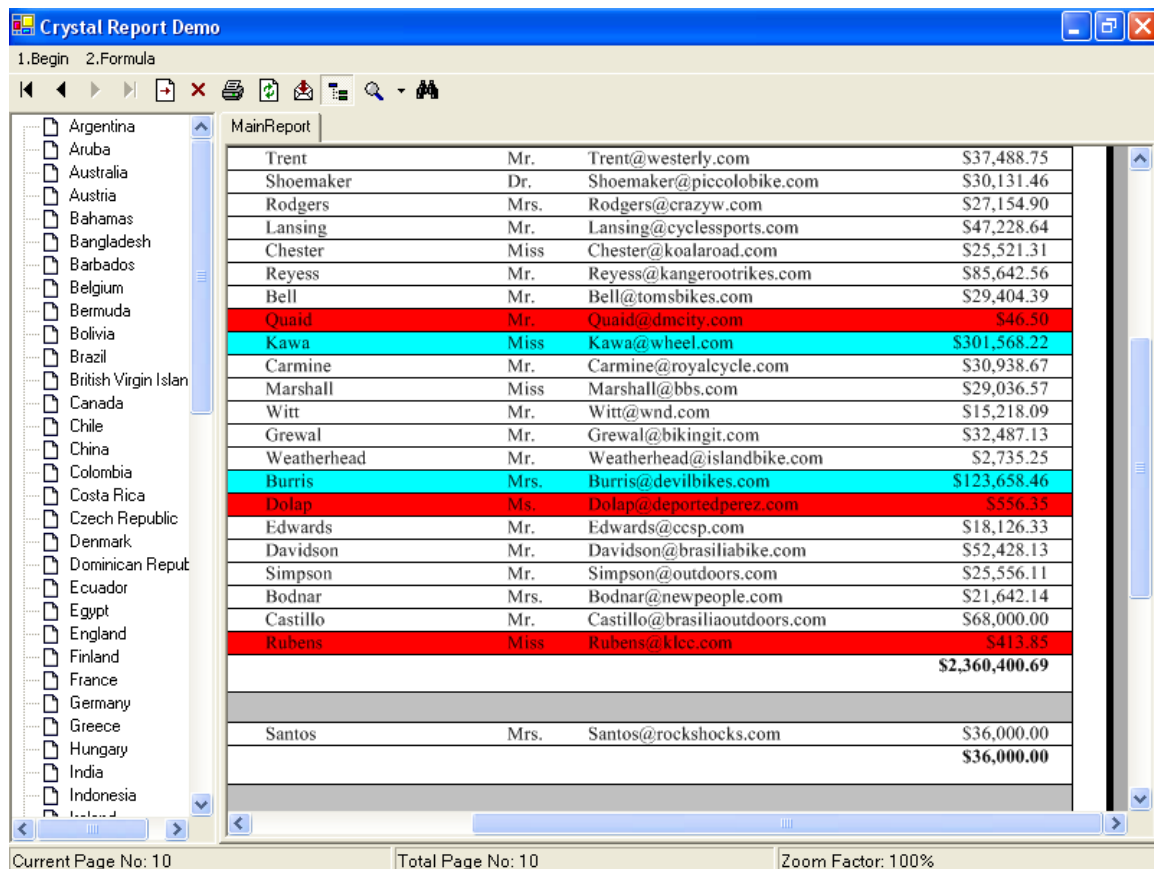
Trên hộp thoại hình 3-04, bạn thấy nút có hình  , đó là formula button nằm cạnh bên phải của mỗi thuộc tính. Bạn click lên nút formula này để hiển thị hộp thoại **Format Formula Editor** (hộp thoại này rất giống với hộp thoại **Formula Editor**) cho phép bạn đưa vào một công thức để thay đổi thuộc tính. Trên hộp thoại **Format Formula Editor** bạn thêm vào nội dung như sau, sau đó nhấn nút **Save and close** hay **Ctrl + S**:

```

If {Customer.Last Year's Sales} > 100000 Then
    Formula = crAqua
ElseIf {Customer.Last Year's Sales} < 1000 Then
    Formula = crRed
Else
    Formula = DefaultAttribute
End If

```

Giá trị DefaultAttribute là giá trị bạn gán cho thuộc tính vào lúc thiết kế. Sau khi lưu công thức, quay trở lại hộp thoại Section Expert bạn để ý thấy nút  đã thay đổi thành nút có hình  cho biết thuộc tính này có sử dụng công thức. Cho preview bản báo cáo bạn sẽ có được kết quả như trên hình 3-05.



The screenshot shows the Crystal Report Demo application. The left pane lists various countries. The main report area displays a table with customer information and sales data. The table has columns for Name, Title, Email, and Sales Amount. The data is sorted by Sales Amount in descending order. The total sales amount is \$2,360,400.69.

Name	Title	Email	Sales Amount
Trent	Mr.	Trent@westerly.com	\$37,488.75
Shoemaker	Dr.	Shoemaker@piccolobike.com	\$30,131.46
Rodgers	Mrs.	Rodgers@crazyw.com	\$27,154.90
Lansing	Mr.	Lansing@cyclessports.com	\$47,228.64
Chester	Miss	Chester@koalaroad.com	\$25,521.31
Reyess	Mr.	Reyess@kangerootrikes.com	\$85,642.56
Bell	Mr.	Bell@tomsbikes.com	\$29,404.39
Quaid	Mr.	Quaid@dmcity.com	\$46.50
Kawa	Miss	Kawa@wheel.com	\$301,568.22
Carmine	Mr.	Carmine@royalcycle.com	\$30,938.67
Marshall	Miss	Marshall@bbs.com	\$29,036.57
Witt	Mr.	Witt@wnd.com	\$15,218.09
Grewal	Mr.	Grewal@bikingit.com	\$32,487.13
Weatherhead	Mr.	Weatherhead@islandbike.com	\$2,735.25
Burris	Mrs.	Burris@devilbikes.com	\$123,658.46
Dolap	Ms.	Dolap@deportedperez.com	\$556.35
Edwards	Mr.	Edwards@ccsp.com	\$18,126.33
Davidson	Mr.	Davidson@brasiliabike.com	\$52,428.13
Simpson	Mr.	Simpson@outdoors.com	\$25,556.11
Bodnar	Mrs.	Bodnar@newpeople.com	\$21,642.14
Castillo	Mr.	Castillo@brasiliaoutdoors.com	\$68,000.00
Rubens	Miss	Rubens@klec.com	\$413.85
			\$2,360,400.69
Santos	Mrs.	Santos@rockshocks.com	\$36,000.00
			\$36,000.00

Current Page No: 10 Total Page No: 10 Zoom Factor: 100%

Hình 3-05: Kết quả preview của file báo cáo sau định dạng bằng công thức

3.3 Tạo các công thức với Basic Syntax

Basic Syntax được đưa vào từ phiên bản Crystal Reports 7.0 trở đi. Nó gần giống với Visual Basic, sử dụng cùng hàm và toán tử, nhưng có khả năng truy cập tất cả các hàm và chức năng đặc trưng của Crystal Reports.

3.3.1 Các quy ước lập trình Basic Syntax

Basic Syntax không phân biệt chữ hoa chữ thường. Cho dù công thức của bạn không cần kết xuất ra giá trị, nhưng bạn phải dùng đến biến Formula và gán cho nó một trị tùy ý.

3.3.1.1 Viết các ghi chú, diễn giải

Các ghi chú, diễn giải trong công thức được bắt đầu bằng dấu nháy đơn (‘) như ví dụ sau

```
' Công thức tính Tổng tiêu thụ
Global TotalOrders as Number
TotalOrders = TotalOrders + {Orders_Detail.Quantity}
Formula = 999 ' Do sử dụng Basic Syntax nên bắt buộc phải có dòng này
```

3.3.1.2 Trình bày các câu lệnh

Bạn có thể viết một câu lệnh trên nhiều dòng bằng cách sử dụng ký tự gạch chân “_” để liên kết giữa hai dòng. Mặc khác, trên một dòng bạn cũng có thể viết nhiều câu lệnh bằng cách sử dụng ký tự dấu hai chấm “:” để phân cách giữa hai lệnh.

```
Dim x As Number
Dim y As String
' Gán giá trị cho x = 30
x = 10 + _
    10 + 10
' 3 câu lệnh được viết trên một dòng
y = "Hello" : x = 30 : Formula = True
```

3.3.1.3 Tham chiếu các field báo cáo

Để tham chiếu đến các field cơ sở dữ liệu phải theo qui ước **{tablename.fieldname}** như ví dụ sau:

```
{Orders_Detail.Quantity} * {Orders_Detail.Unit Price}
```

Các field khác sử dụng một tiền tố (prefix) cho biết loại field đang làm việc. Parameter field sẽ có tiền tố là dấu chấm hỏi (?). Formula field có tiền tố là dấu a móc (@). Running Total Field sẽ là dấu thăng (#).

3.3.1.4 Khai báo biến

Việc khai báo biến sẽ theo dạng chuẩn sau, khai báo phạm vi rồi đến tên biến kế tiếp là từ khoá As và kiểu dữ liệu.

```
Dim var As datatype
```

Lần đầu tiên một công thức được gọi, các biến sẽ được gán về những giá trị mặc định như trong bảng 3-01.

Bảng 3-01: Các giá trị mặc định đối với kiểu dữ liệu khác nhau

Kiểu dữ liệu	Giá trị mặc định
Number	0
Boolean	False
String	""
Date	00/00/00
Time	Không có trị mặc nhiên
Datetime	Không có trị mặc nhiên

Một khi biến đã được gán một giá trị nào đó thì nó sẽ giữ nguyên các giá trị trong suốt cuộc sống của nó. Bất cứ lời gọi thực thi nào trong tương lai đối với cùng công thức sẽ không khởi gán lại giá trị mặc định cho biến. Bởi vì với Crystal Reports thì formula và report section cùng hoạt động với nhau. Các công thức thường được thêm vào report section (thường là Details section) và report section luôn được vào ra khi mẫu tin được xử lý. Nếu một biến được khởi gán lại mặc định mỗi lần một mẫu tin được xử lý thì sẽ không cách nào theo dõi việc gì xảy ra với mẫu tin cuối cùng. Nếu bạn muốn gán lại giá trị mặc định mỗi lần mẫu tin được xử lý thì bạn hãy tự làm bằng tay.

Phạm vi hoạt động của một biến được khai báo bằng 1 trong 3 từ khoá sau:

- **Local/Dim:** biến có phạm vi Local.

- **Global**: biến có phạm vi global nhưng các subreport không thể truy cập được biến này.
- **Shared**: giống như Global, nhưng các subreport có thể truy cập biến này.

Lưu ý: Một vấn đề bất thường của các biến `Global/Shared` là cho dù phạm vi của chúng cho biết là có thể sử dụng chúng trong các công thức khác, nhưng bạn vẫn phải khai báo chúng trong mỗi công thức muốn sử dụng chúng.

3.3.2 Các kiểu dữ liệu

3.3.2.1 Các kiểu dữ liệu chuẩn

Basic Syntax hỗ trợ những kiểu dữ liệu chuẩn đơn giản thường có ở trong một ngôn ngữ: `Boolean`, `Number`, `Currency`, `String`, `DateTime`, `Date` và `Time`.

Kiểu `String` được đánh chỉ số bắt đầu từ 1 (chứ không phải từ 0, các bạn quen lập trình với .NET nên lưu ý điều này). Chiều dài của một hằng chuỗi là 65.534 ký tự.

` Ví dụ gán một string constant cho một biến chuỗi`

```
Dim Var As String
```

```
Var = "Đây là một chuỗi"
```

Muốn gán một hằng kiểu `DateTime`, bạn cho bao bởi dấu `#` như sau:

```
Dim NgayLapHoaDon As DateTime
```

```
NgayLapHoaDon = #3/29/2006#
```

3.3.2.2 Kiểu dữ liệu mảng

Khác với .NET, một mảng trên Basic Syntax bắt đầu với chỉ số là 1 (thay vì 0 như thao .NET). Kích thước một mảng tối đa là 100 phần tử.

Muốn thay đổi kích thước mảng sau khi khai báo bạn dùng lệnh `ReDim`, nếu bạn muốn duy trì giá trị của các phần tử hiện có thì bạn dùng thêm từ khoá `Preserve`. Ví dụ:

```
Dim var() As datatype ` khai báo một mảng không có kích thước
Redim var(number) ` redim lại kích thước và reset tất cả các giá trị
Redim Preserve var(number) `redim kích thước và duy trì giá trị hiện có
```

Để gán giá trị ban đầu cho các phần tử trong mảng bạn sử dụng hàm `Array()` như sau:

```
Dim DaysInWeek() As String
DaysInWeek = Array("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")
```

Để truy xuất giá trị của phần tử thứ *i* bạn theo cú pháp sau `ArrayName(i)`. Để xác định xem một giá trị đã tồn tại trong mảng hay chưa, bạn sử dụng tác tử `In` bạn sẽ tránh được việc phải duyệt qua toàn bộ mảng để tìm một trị đặt biệt nào đó. Nếu giá trị cần tìm có trong mảng thì tác tử `In` sẽ trả về `True`. Xem ví dụ sau:

```
` Thí dụ đoạn code lấy danh sách zipcode của tất cả khách hàng được in
` ra và lưu vào trong mảng ZipCodes
Dim RecordCounter As Number
Dim ZipCodes(100) As Number
If Not {Customer.ZipCode} In ZipCodes Then
    RecordCounter = RecordCounter + 1
    ZipCodes(RecordCounter) = {Customer.ZipCode}
End If
```

Basic Syntax cung cấp một số hàm cho phép bạn tổng kết các giá trị trên mảng, được liệt kê theo bảng 3-02.

Bảng 3-02: Các hàm Array Summary

Hàm	Mô tả
<code>Average(array)</code>	Tính ra trung bình trên tất cả các phần tử của mảng.
<code>Count (array)</code>	Đếm xem có bao nhiêu phần tử trên mảng.
<code>DistinctCount(array)</code>	Đếm xem có bao nhiêu phần tử phân biệt.
<code>Maximum(array)</code>	Trả về phần tử có trị lớn nhất.
<code>Minimum(array)</code>	Trả về phần tử có trị nhỏ nhất.
<code>PopulaionStdDev(array)</code>	Trả về tính toán Population Standart Deviation.
<code>PopulaionVariance(array)</code>	Trả về tính toán Population Variance.
<code>StdDev(array)</code>	Trả về tính toán Standard Deviation.
<code>Variance(array)</code>	Trả về tính toán Variance.

3.3.2.3 Kiểu dữ liệu Range

Range là kiểu dữ liệu rất hữu ích nhưng lại chưa có trên .NET. Nó cho phép bạn lưu trữ một range bên trong một biến rồi dùng biến đó để kiểm tra xem một biến khác có lọt vào trong range này hay không dựa vào điểm bắt đầu và điểm kết thúc của range (điểm bắt đầu và điểm kết thúc phải thuộc cùng một kiểu dữ liệu).

Muốn khai báo một biến thuộc kiểu dữ liệu range, bạn khai báo biến theo kiểu dữ liệu chuẩn: `Boolean`, `Number`, `Currency`, `String`, `DateTime`, `Date` và `Time` rồi kết thúc bằng từ khóa `Range`.

```
Dim var As datatype Range
```

Định nghĩa một range sẽ dùng nhiều tác tử khác nhau để khai báo giá trị bắt đầu và kết thúc của range. Tác tử cơ bản nhất là `To`. Nó được đặt ở giữa giá trị bắt đầu và kết thúc. Một biến thể của tác tử `To` là sử dụng nó với một ký tự gạch dưới (underscore). Đặt underscore tại một phía của tác tử `To` cho biết giá trị của phía đó sẽ không thuộc về range. Hãy xem ví dụ sau

```
Dim NumberRange As Number Range
NumberRange = 0 To 1000 ' range: 0..1000
NumberRange = 0 To_ 1000 ' range: 0..999
NumberRange = 0 _To 1000 ' range: 1..1000
NumberRange = 0 _To_ 1000 ' range: 1..999
' Dùng trực tiếp không cần khai báo biến
If {Customer.Last Year's Sales} In (30000 To_ 100000) Then
    Formula = "Great Job"
End If
```

Basic Syntax cung cấp sẵn cho chúng ta một số Predefined Date Range để chúng ta sử dụng được liệt kê trong bảng 3-03.

Bảng 3-03: Predefined Date Range

Tên	Mô tả
<code>AllDatesToToday</code>	Từ ngày hôm nay trở về trước
<code>AllDatesToYesterday</code>	Từ ngày hôm qua trở về trước
<code>AllDatesFromToday</code>	Bắt đầu: ngày hôm nay Kết thúc: ngày cuối cùng trên vùng mục tin
<code>AllDatesFromTomorrow</code>	Bắt đầu: ngày mai Kết thúc: ngày cuối cùng trên vùng mục tin
<code>Aged0to30Days</code> <code>Aged31to60Days</code> <code>Aged61to90Days</code>	Nhóm ngày theo khối 30 ngày trước ngày hôm nay. Ngày của ngày hôm nay được bao gồm trong <code>Aged0to30Days</code>
<code>Calendar1stQtr</code> <code>Calendar2ndQtr</code> <code>Calendar3rdQtr</code>	Nhóm ngày trong khối 3 tháng .Nếu ngày đầu tiên là ngày 01 tháng 01 năm hiện hành.

Calendar4thQtr	
Calendar1stHalf	Bắt đầu: 01 tháng giêng năm hiện hành đến 30 tháng 06
Calendar2ndHalf	Kết thúc: 01 tháng 07 năm hiện hành cho đến 31 tháng 12
Last7Days	Bắt đầu: 6 ngày trước ngày hôm nay Kết thúc: Ngày hôm nay
Last4WeeksToSun	Bắt đầu: ngày thứ hai đầu tiên của 4 tuần trước ngày chủ nhật cuối cùng Kết thúc: ngày chủ nhật cuối cùng Bạn để ý: không bao gồm các ngày sau ngày chủ nhật cuối cùng cho đến ngày hôm nay
LastFullWeek	Bắt đầu: ngày chủ nhật của chọn tuần lễ cuối . Kết thúc: ngày thứ bảy của chọn tuần lễ cuối. Bạn chú ý: không bao gồm những ngày sau ngày thứ bảy đến ngày hôm nay
LastFullMonth	Bắt đầu: ngày đầu tiên của tháng cuối cùng Kết thúc: ngày cuối cùng của tháng cuối cùng
MonthToDate	Bắt đầu: ngày đầu tiên của tháng Kết thúc: ngày hôm nay
Next30Days Next31To60Days Next61To90Days Next91To365Days	Nhóm ngày theo 4 khối 30 ngày sau ngày hôm nay. Ngày của ngày hôm nay được bao gồm trong range Next30Days
Over90days	Bao gồm tất cả các ngày trước 90 ngày ,trước ngày hôm nay
WeekToDateFromSun	Bắt đầu: ngày chủ nhật cuối Kết thúc: ngày hôm nay
YearToDate	Bắt đầu: 01 tháng 01 năm hiện hành Kết thúc: ngày hôm nay

3.3.3 Các toán tử đơn giản

Bây giờ ta xem qua một vài tác tử đã được định nghĩa sẵn trên **Operator Tree** trong hộp thoại **Formula Editor** hình 3-02.

3.3.3.1 Toán tử toán học

Crystal Reports .NET hỗ trợ tất cả các toán tử toán học cơ bản (bao gồm cộng trừ nhân chia) nhưng còn hỗ trợ một số tác tử khác như trên bản 3-04 sau đây.

Bảng 3-04: Các toán tử toán học khác

Toán tử	Ký hiệu	Mô tả
Integer Divide	\	Chia lấy phần nguyên. (Ví dụ , $9 \setminus 2$ sẽ trả về kết quả là 4)
Modulus	Mod	Chia lấy phần dư. (Ví dụ, $9 \text{ Mod } 2$ sẽ trả về số dư là 1)
Negate	- ()	Cho thay đổi dấu âm dương (sign) của một con số.
Exponentiate	^	Tính số mũ, dùng tính bình phương(Ví dụ, 3^2 sẽ trả về 9)

3.3.3.2 Toán tử Boolean

Crystal Reports .NET cung cấp cho chúng ta một số toán tử Boolean trả về giá trị True hoặc False, như theo bảng 3-05 sau đây:

Bảng 3-05: Các toán tử Boolean

Tác tử Boolean	Mô tả
Not [Not x]	Đảo ngược trị - thí dụ Not (True) là False
And [x And y]	Nếu tất cả các điều kiện là True thì trả về là True. Nếu một trong các điều kiện không thỏa tiêu chí , thì trả về False.
Or [x Or y]	Trả về True nếu một trong hai điều kiện hoặc cả hai thỏa mãn tiêu chí
Xor [x Xor y]	Logical exclusion ,cho biết khi nào hai biểu thức là khác nhau.Trả về True nếu một điều kiện là True và điều kiện kia là False .Nếu cả hai biểu thức là True hoặc là False ,thì trả về False
Eqv [x Eqv y]	Logical equivalence,cho biết khi nào hai biểu thức là giống nhau.Trả về True nếu cả hai trị được so sánh là True hoặc nếu cả hai trị được so sánh là False.Nếu hai trị được so sánh khác nhau thì trả về False.
Imp [x Imp y]	Logical implication ,cho biết là nếu biểu thức đầu tiên là True thì ngầm hiểu biểu thức thứ hai cũng là True.Nếu biểu thức thứ hai cũng là True thì Imp trả về True.Còn nếu biểu thức thứ hai là False thì Imp trả về False vì nó không thỏa theo những ngầm hiểu.Ngược lại nếu biểu thức thứ nhất là False ,thì không có gì được ngầm hiểu và kết quả bao giờ cũng là True .Như vậy ,lần duy nhất Imp sẽ trả về là False là khi biểu thức thứ nhất là True và biểu thức thứ hai là False.

3.3.3.3 Toán tử so sánh

Khi so sánh hai giá trị (biến hoặc hằng) Basic Syntax những toán tử so sánh chuẩn: Equal to (=), Not equal (<>), Less than (<), Greater than (>), Less or Equal (<=) và Greater or Equal (>=).

3.3.3.4 Toán tử chuyển đổi kiểu dữ liệu

Bảng 3-06 sau đây liệt kê tất cả các hàm chuyển đổi kiểu dữ liệu. Các hàm này hoạt động tương tự như các hàm tương đương trên .NET.

Bảng 3-06: Các hàm chuyển đổi kiểu dữ liệu

Hàm chuyển đổi	Mô tả
Cbool (number) Cbool (currency)	Hàm này nhận tham số kiểu Number hoặc Currency và chuyển đổi thành trị boolean : True hoặc False. Bất cứ trị non-zero nào âm hoặc dương đều trả về True còn trị nào zero sẽ trả về False . Khi hiển thị các bản báo cáo sẽ in ra các từ “True” hoặc “False”.
Ccur (number) Ccur (string)	Hàm này nhận thông số kiểu Number hoặc String và chuyển đổi thành kiểu dữ liệu Currency. Khi chuyển đổi một string , có thể có những ký tự định dạng trong chuỗi , chẳng hạn “\$” , “,” , ... và chuỗi sẽ được chuyển đổi một cách thích hợp.
Cdbl (currency) Cdbl (string) Cdbl (boolean)	Hàm này chuyển đổi thành Number khi nhận thông số kiểu dữ liệu Currency, String hoặc Boolean . Hàm này tương đương với hàm ToNumber () .
CStr ()	Chuyển đổi các kiểu dữ liệu kiểu Number, Currency hoặc Date thành String . Tương đương với ToText () .
Cdate (string) CDate (year, month, day) CDate (Datetime)	Hàm này chuyển đổi thành một field thực sự Date <u>Ghi chú:</u> Các hàm CDate (), CTime () và CDateTime () tất cả giống nhau. Trao cho chúng một string nó sẽ chuyển đổi thành kiểu dữ liệu thích ứng. String parser đối với các hàm này rất tin vi. Chúng cho phép bạn nhập những chuỗi khác nhau chẳng hạn “May 23, 2004”, “25/25/2004” hoặc “2004, Dec 21”. Bạn cũng

	có thể trao các con số như là các thông riêng rẽ tượng trưng cho các thành phần khác nhau của một ngày hoặc thời gian.
<code>Ctime(string)</code> <code>Ctime(hour,min,sec)</code> <code>Ctime(DateTime)</code>	Hàm này chuyển đổi thành một field <code>Time</code> .Xem ghi chú trên <code>CDate()</code> .
<code>CdateTime(string)</code> <code>CdateTime(date)</code> <code>CdateTime(date, time)</code> <code>CdateTime(year, month, day)</code> <code>CdateTime(year, month, day, hour, min, sec)</code>	Hàm này chuyển đổi thành một field <code>DateTime</code> .Xem ghi chú trên <code>CDate()</code> .
<code>ToNumber(string)</code> <code>ToNumber(boolean)</code>	Hàm này nhận thông số kiểu <code>string</code> và <code>boolean</code> và chuyển thành <code>Number</code> .Hàm này tương đương với <code>CDBle()</code> .
<code>ToText()</code>	Hàm này nhận thông số kiểu <code>Number</code> , <code>Currency</code> , <code>Date</code> , <code>Time</code> hoặc <code>boolean</code> và chuyển đổi thành văn bản. Hàm này tương đương với <code>CStr()</code> .
<code>IsDate(string)</code> <code>IsTime()</code> <code>IsDateTime()</code>	Khi chuyển đổi một kiểu dữ liệu <code>Date</code> , <code>Time</code> hoặc <code>DateTime</code> bạn sẽ gặp nguy cơ lỗi xảy ra nếu chuỗi không đúng dạng thức .Các hàm này giúp bạn trắc nghiệm trước xem chuỗi có hợp lệ hay không trước khi ra lệnh chuyển đổi.Hàm sẽ trả về <code>True</code> nếu chuỗi có thể được chuyển đổi thích ứng ,bằng không sẽ trả về <code>False</code> .
<code>IsNumber (string)</code>	Khi chuyển đổi một kiểu dữ liệu <code>Number</code> bạn sẽ gặp nguy cơ lỗi xảy ra nếu chuỗi không đúng dạng thức. Hàm này giúp bạn trắc nghiệm trước xem chuỗi có hợp lệ hay không trước khi ra lệnh chuyển đổi. Hàm sẽ trả về <code>True</code> nếu chuỗi có thể được chuyển đổi thích ứng ,bằng không sẽ trả về <code>False</code> .
<code>ToWords (number)</code> <code>ToWords(number,decimals)</code>	Hàm này chuyển đổi số thành chữ giống như khi các chi phiếu hoặc các hoá đơn (Ví dụ, 101 sẽ thành “One hundred and one”) ¹

3.3.4 Các Hàm cơ bản

Bây giờ ta xem qua một hàm đã được định nghĩa sẵn trên **Functions Tree** trong hộp thoại **Formula Editor** hình 3-02.

3.3.4.1 Các hàm tổng kết

Khi bạn cho chèn một tổng kết vào bản báo cáo, một hàm tổng kết (summary function) tương ứng sẽ được dùng để tạo một field summary chuyên biệt. Các hàm tổng kết bao gồm như sau: `Sum()`, `Average()`, `Count()`, `Maximum()`, `Minimum()` ...

Các summary field thường được sử dụng theo một trong hai cách khác nhau. Cách thứ nhất, theo đó hàm tổng kết và một field đơn lẻ được tượng trưng cho grand total trên bản báo cáo.

```

\ Tính tổng tất cả các giá trị trong Customer.Sales
Sum({Customer.Sales})

```

Ngoài ra, các summary field cũng có thể được dùng trong group level. Ví dụ, nếu filed group là {Customer.Coutry} thì summary field sẽ như sau

```

\ Tính tổng tất cả các giá trị trong Customer.Sales
\ được group theo Customer.Coutry
Sum({Customer.Sales}, {Customer.Coutry})

```

3.3.4.2 Các hàm xử lý chuỗi

Khả năng thay đổi và ráp nối chuỗi là một tính năng cực mạnh của đa số ngôn ngữ lập trình và Basic Syntax cũng không ngoại lệ. Phần này phân loại các hàm xử lý chuỗi thành: phân tích chuỗi

Nhiều hàm được liệt kê sau đây thường có hai tham số sau đây: `compare` và `start`. Tham số `compare` xác định phép so sánh chuỗi là case-sensitive (không phân biệt chữ hoa chữ thường). Nếu `compare = 0`, sẽ là so sánh case-sensitive. Tham số `compare` là tùy chọn, nếu không ghi rõ ra thì mặc định sẽ là case-sensitive.

Tham số `start`, tùy chọn, cho biết hàm sẽ bắt đầu xử lý tại vị trí `start` này. Nếu không ghi rõ tham số này ra, thì mặc định là toàn bộ chuỗi sẽ được xử lý.

Sau đây là bảng 3-07 sẽ liệt kê các hàm xử lý chuỗi.

Bảng 3-07 Các hàm xử lý chuỗi

¹ Công thức chuyển đổi số thành chuỗi Tiếng Việt xem phần 3.4

Hàm	Mô tả
Các hàm đơn giản	
<code>Asc(str)</code>	Trả về trị ASCII của một ký tự <i>str</i> .
<code>Chr(val)</code>	Trả về một ký tự tương đương với một trị ASCII <i>val</i> .
<code>Len(str)</code>	Cho biết số ký tự trong một chuỗi <i>str</i> .
<code>IsNumeric(str)</code>	Trị boolean, cho biết liệu trong <i>str</i> có thể được chuyển đổi thành một Number hay không.
<code>Val(str)</code>	Hàm này trả về trị số tương đương với <i>str</i> .
<code>Space(val)</code>	Hàm này trả về một số space được khai báo như là một chuỗi đơn độc.
Các hàm chuyển đổi chuỗi	
<code>UCase(str)</code>	Chuyển đổi một chuỗi biến tất cả các ký tự thành chữ hoa.
<code>LCase(str)</code>	Chuyển đổi một chuỗi biến tất cả các ký tự thành chữ thường.
<code>StrReverse(str)</code>	Hàm này cho đảo ngược tất cả các ký tự trong chuỗi <i>str</i> .
Các hàm xén chuỗi	
<code>Trim(str)</code>	Xén bỏ các khoảng trắng (space) ở cả 2 đầu chuỗi <i>str</i> .
<code>LTrim(str)</code>	Xén bỏ các khoảng trắng (space) nằm ở bên trái chuỗi <i>str</i> .
<code>RTrim(str)</code>	Xén bỏ các khoảng trắng (space) nằm ở bên phải chuỗi <i>str</i> .
<code>Mid(str, start, length)</code>	Trả về một số ký tự nào đó bắt đầu từ một vị trí được khai báo. Các tham số <i>start</i> và <i>length</i> là tùy chọn. Nếu không khai báo <i>length</i> thì hàm sẽ trả về tất cả các ký tự bắt đầu từ ký tự đầu tiên được khai báo.
<code>Left(str, length)</code>	Trả về một số ký tự nào đó (dài do <i>length</i> chỉ định) bắt đầu từ ký tự tận cùng bên trái.
<code>Right(str, length)</code>	Trả về một số ký tự nào đó (dài do <i>length</i> chỉ định) bắt đầu từ ký tự tận cùng bên phải.
Các hàm xử lý dựa trên 2 chuỗi	
<code>InStr(start, str1, str2, compare)</code>	Hàm này xác định liệu xem chuỗi <i>str2</i> có phải là một chuỗi con trong <i>str1</i> hay không. Cả hai tham số <i>start</i> và <i>compare</i> là tùy chọn.
<code>InStrRev(start, str1, str2, compare)</code>	Hàm này tương tự như hàm <code>InStr()</code> nhưng làm ngược lại đi từ cuối ngược lên đầu.
<code>StrCmp(str1, str2, comp)</code>	So sánh 2 chuỗi <i>str1</i> và <i>str2</i> với nhau. Trị trả về dựa trên

<code>are)</code>	<p>việc so sánh 2 chuỗi với nhau:</p> <ul style="list-style-type: none"> • -1 nếu <code>str1 < str2</code> • 0 nếu <code>str1 = str2</code> • 1 nếu <code>str1 >= str2</code> <p>Tham số <code>compare</code> tùy chọn.</p>
<code>Filter (str, find, include, compare)</code>	Hàm này cho truy tìm trên một mảng các chuỗi <i>str</i> xem có chuỗi con <i>find</i> khớp hay không. <i>include</i> là một trị boolean, tùy chọn, cho biết liệu xem có trả về những chuỗi con có bao gồm <i>find</i> hay không. Nếu <i>include</i> là <code>True</code> , thì hàm trả về là mảng tập hợp con chứa <i>find</i> như là một chuỗi con. Còn nếu <i>include</i> là <code>False</code> thì hàm sẽ trả về mảng tập hợp không chứa <i>find</i> như là một chuỗi con.
<code>Replace(str, find, replace, start, count, compare)</code>	Hàm này tìm trên chuỗi <i>str</i> một chuỗi con <i>find</i> và nếu có thay thế nó bởi một chuỗi con khác <i>replace</i> . Các tham số <code>start</code> , <code>count</code> và <code>compare</code> là tùy chọn.
<code>ReplicateString(str, copies)</code>	Hàm này trả về bản sao của một chuỗi <i>str</i> .
<code>Join (list, delimiter)</code>	Hàm này trả về một chuỗi hình thành bằng cách ráp một mảng các chuỗi <i>list</i> và ngăn cách các chuỗi bởi một ký tự phân cách <i>delimiter</i> .
<code>Split(str, delimiter, count, compare)</code>	Tách một chuỗi đơn lẻ <i>str</i> thành một mảng các chuỗi dựa trên ký tự phân cách <i>delimiter</i> . Các tham số <code>count</code> , <code>compare</code> là tùy chọn.
<code>Picture(str, template)</code>	Hàm này cho định dạng các ký tự trong một chuỗi <i>str</i> lên một <i>template</i> . <i>template</i> bao gồm một loạt các ký tự 'x' dùng để ánh xạ một ký tự trên <i>str</i> lên nó. Nếu kích thước chuỗi <i>str</i> lớn hơn <i>template</i> thì các ký tự còn lại sẽ được thêm vào cuối. Nếu <i>tempate</i> có bất kỳ ký tự nào khác 'x' thì ký tự đó sẽ được giữ nguyên.

Sau đây là một số ví dụ về các hàm xử lý chuỗi

```
` Ví dụ sử dụng hàm Filter()  
Dim StringArray() As String  
Dim FilterArray() As String  
StringArray = Array ("abcd", "bcde", "cdf")  
` Kết quả là một mảng gồm các phần tử "abcd" và "bcde"  
FilterArray = Filter(StringArray, "bcd")  
` Kết quả là một mảng với chỉ một phần tử "cdf"  
` vì phần tử "cdf" không chứa chuỗi "bcd"  
FilterArray = Filter(StringArray, "bcd", False)
```

```
` Ví dụ hàm Replace() làm thay đổi các địa chỉ sử dụng cách viết tắt  
Address = "127 Nguyễn Văn Cừ Street - 05 District"  
Address = Replace(Address, "Street", "St.")  
Address = Replace(Address, "District", "Dst.")  
` Kết quả Address = "127 Nguyễn Văn Cừ St. - 05 Dst."
```

```
` Ví dụ sử dụng hàm Join() & Split()  
Dim Names As String  
Dim NameArray() As String  
Names = "Mr.Jones, Sir Alfred, Ms.Bee, Mr.Smith"  
NameArray = Split(Names, ",")  
` Kết quả: NameArray = {"Mr.Jones", "Sir Alfred", "Ms.Bee", "Mr.Smith"}  
NameArray = Filter(NameArray, "Mr." )  
` Kết quả: NameArray = {"Mr.Jones", "Mr.Smith"}  
` Ráp lại mảng với dấu phẩy làm dấu phân cách  
Names = Join(NameArray, ",")  
` Kết quả: Names = "Mr.Jones, Mr.Smith"
```

```
` Ví dụ hàm Picture() để ánh xạ một chuỗi  
Formula = Picture("yahoo", "www.xxxxx.com")  
` Kết quả là www.yahoo.com
```


3.3.4.3 Các hàm Date & Time

Vì phần lớn các bản báo cáo đều có liên quan đến thông tin ngày giờ dưới hình thức này hay hình thức khác, nên Crystal Reports cung cấp cho chúng ta một số hàm liên quan đến ngày giờ được liệt kê theo bảng 3-08 sau.

Bảng 3-08: Các hàm Date & Time

Hàm	Mô tả
<code>CurrentDate</code> <code>CurrentTime</code> <code>CurrentDateTime</code>	Hàm này trả về ngày giờ hiện hành.
<code>DateSerial(year, month, day)</code> <code>DateTime(hour, minute, second)</code>	Hàm này trả về một ngày hoặc giờ.
<code>DateAdd (interval, number, date)</code>	Hàm này tăng ngày thêm một khoảng thời gian.
<code>DateDiff (interval, startdate, enddate, firstdayofweek)</code>	Hàm này trả về sai biệt giữa hai ngày <i>firstdayofweek</i> là tùy chọn
<code>DatePart (interval, date, firstdayofweek, firstweekofyear)</code>	Hàm này trả về một con số tượng trưng khoảng cách hiện hành của một ngày.
<code>MonthName(date, abbreviate)</code>	Hàm này nhận một con số tượng trưng cho tháng và trả về chọn tên tháng hoặc trả về tháng viết tắt theo 3 mẫu tựa như thông số <i>abbreviate</i> là <code>True</code> . Theo mặc nhiên <i>abbreviate</i> là <code>False</code> trả về chọn tên tháng.
<code>Timer</code>	Hàm này trả về số giây đã đi qua kể từ nửa đêm.
<code>Weekday(date, firstdayofweek)</code>	Hàm này trả về một con số (1, 2, ...7) cho biết ngày trong tuần.
<code>WeekdayName(weekday, abbreviate, firstdayofweek)</code>	Hàm này nhận một con số (1 đến 7) tượng trưng ngày trong tuần và trả về trọn tên ngày hoặc trả về ngày viết tắt theo 3 mẫu tựa nếu thông số <i>abbreviate</i> là <code>True</code> .

Bảng 3-09: Liệt kê các chuỗi interval dùng làm tham số cho các hàm `DateAdd()`, `DateDiff()` và `DatePart()`.

Bảng 3-09: Interval String

String	Mô tả
“yyyy”	Year
“q”	Quarter – Quý
“m”	Month - Tháng
“ww”	Week (7 days) - Tuần lễ 7 ngày
“w”	Weekday – Ngày trong tuần
“y”	Day of year – Ngày trong năm
“d”	Day – Ngày
“h”	Hour-giờ
“n”	Minute – Phút
“s”	Second - Giây

3.3.4.4 Các hàm toán học

Bảng 3-10 sau đây liệt kê các hàm toán học trên Crystal Reports.

Bảng 3-10: Các hàm toán học

Hàm	Mô tả
<code>Abs (number)</code>	Trả về trị tuyệt đối.
<code>Fix (number, decimals)</code>	Trả về một con số với một số ký tự (digit) có ý nghĩa.
<code>Int (number)</code> <code>Numerator\denominator</code>	Trả về phần số nguyên của một phân số.
<code>Pi</code>	Trả về 3.14...
<code>Remainder (numerator, denominator)</code> <code>Numerator Mod denominator</code>	Trả về số dư khi chia <i>numerator</i> cho <i>denominator</i> .
<code>Round (number, decimals)</code>	Trả về một số được làm tròn với một số ký tự số có ý nghĩa.
<code>Sgn (number)</code>	Trả về một số nguyên cho biết dấu âm dương của một con số. Số trả về bằng 1 cho biết lớn hơn zero, 0 là bằng zero, còn -1 là nhỏ hơn zero
<code>Sqr (number)</code> <code>Exp (number)</code> <code>Log (number)</code>	Các hàm số học chuẩn.
<code>Cos (number)</code> <code>Sin (number)</code> <code>Tan (number)</code> <code>Atn (number)</code>	Các hàm khoa học chuẩn.

3.3.4.5 Định dạng các trị kết xuất

Muốn xây dựng một chuỗi kết xuất với tổ hợp những kiểu dữ liệu khác nhau, bạn sử dụng hàm CStr(). Hàm này nhận tham số đầu tiên là giá trị cần định dạng, và tham số thứ hai là chuỗi định dạng bao gồm một hay nhiều ký tự định dạng liệt kê trong bảng 3-11.

Bảng 3-11: Các ký tự định dạng của CStr()

Format	Mô tả
#	Dùng định dạng các con số. Nếu con số không đủ dài thì space sẽ được dùng thay vào đó. Nếu con số quá dài, so với chuỗi định dạng, thì phần số nguyên sẽ vẫn được hiển thị đầy đủ. Nhưng digit không dùng đến sau dấu phẩy số lẻ sẽ được điền bởi zero.
0	Dùng định dạng các con số. Nếu con số không đủ dài thì zero sẽ được dùng thay vào đó. Nếu con số quá dài, so với chuỗi định dạng, thì phần số nguyên sẽ vẫn được hiển thị đầy đủ. Nhưng digit không dùng đến sau dấu phẩy số lẻ sẽ được điền bởi zero.
,	Dùng định dạng các con số chỉ định dấu phân cách số ngàn.
.	Dùng định dạng các con số chỉ định dấu phân cách số lẻ.
d,M	Day, Month như là con số (không có zero trước).
dd,MM,yy	Day, Month và Year như là 2 con số (với zero đi đầu nếu thấy cần thiết).
ddd,MMM	Day, Month như là viết tắt gồm 3 mẫu tự.
dddd,MMMM.yyyy	Day, Month và Year viết ra đầy đủ.
h,m,s	Hour, Minute và Second như là con số không có zero đi đầu.
hh,mm,ss	Hour, Minute và Second như là hai con số (với zero đi đầu nếu thấy cần thiết).
HH	Cho thấy Hour sử dụng 24 giờ (military time).
t	Một ký tự tương tự AM/PM.
tt	Hai ký tự tương tự AM/PM.

Bảng 3-12 là những ví dụ cho thấy những giá trị khác nhau sau khi được định dạng.

Bảng 3-12 :Các thí dụ kết xuất CStr()

Stt	CStr()	Kết xuất
1	CStr(1234,2)	1,234.00
2	CStr(1234.567,2)	1,234.57
3	CStr(1234.567,"#")	1234
4	CStr(1234.567,"0")	1234

5	CStr(1234,"0.##")	1234.00
6	CStr(1234,"0.00")	1234.00
7	CStr(1234.567,"#.##")	1234.57
8	CStr(1234.567,"0.00")	1234.57
9	CStr(1234.567,"#####")	1234
10	CStr(1234.567,"00000")	01234
11	CStr(1234.567,"#",2)	1234
12	CStr(1234.567,"#.##",2)	1234.57
13	CStr(1234.567,"#.###",2)	1234.57
14	CStr(1234.567,"#.##",2,"")	1234.57
15	CStr(1234.567,"#.##"2,"","")	1234,57
16	CStr(1234.567,2,"","")	1.234,57
17	CStr(1234.567,"###,###.##")	1,234.57
18	CStr (#1/2/2003 04:05:06 am#, "d/M/yy H/m/s t")	1/2/03 4:5:6 A
19	CStr (#1/2/2003 04:05:06 am#, "dd/MM/yyyy HH/mm/ss tt")	01/02/2003 04:05:06 AM
20	CStr(#3:20 PM#, "HH:mm")	15:20

3.3.5. Các cấu trúc điều khiển

Crystal Report hỗ trợ một số cấu trúc điều kiện mà hầu hết các lập trình viên đều rất quen thuộc.

3.3.5.1 Các cấu trúc điều kiện

Câu lệnh If..Then

Cú pháp:

```

If <Biểu thức 1> Then
    ...
ElseIf <Biểu thức 2> Then
    ...
Else
    ...
End If
  
```

Ví dụ:

```

If {Customer.Last Year's Sales} > 30000 Then
    Formula = "Excellent Job"
ElseIf {Customer.Last Year's Sales} > 20000 Then
  
```

```
Formula = "Great Job"
ElseIf {Customer.Last Year's Sales} > 10000 Then
    Formula = "Fine Job"
End If
```

Câu lệnh Select Case

Cú pháp

```
Select Case var
    Case <Biểu thức 1>
        ...
    Case <Biểu thức 2>
        ...
    Case <Biểu thức 3>, <Biểu thức 4>
        ...
    Case Else
        ...
End Select
```

Lưu ý: câu lệnh này không có từ khoá **break**, nó tự hiểu ngầm khi so sánh biểu thức.

Ví dụ:

```
Select Case {Customer.PriorityNumber}
    Case 1, 2, 3
        Formula = "Hight priority"
    Case 4
        Formula = "Medium priority"
    Case Else
        Formula = "Low priority"
End Select
```

3.3.5.2 Các cấu trúc lặp

Vòng lặp For Next

Cú pháp

```
For var = start To end Step increment
....
Next
```

Vòng lặp While Do

Cú pháp

```
While true_condition
...
Wend
```

Các vòng lặp khác

<pre>Do While true_condition ... Loop</pre>	<pre>Do ... Loop While true_condition</pre>
<pre>Do Until false_condition ... Loop</pre>	<pre>Do ... Loop Until false_condition</pre>

- Từ khóa `While` tiếp tục lặp khi biểu thức `true_condition = True`
- Từ khóa `Until` tiếp tục lặp khi biểu thức `false_condition = False`
- Bạn có thể ngắt ngang vòng lặp `Do` bằng cách sử dụng câu lệnh `Exit Do`.

3.4 Công thức chuyển đổi số thành chuỗi Tiếng Việt

Trong thực tế các dạng report như hóa đơn, biên lai ... ngoài những giá trị ghi bằng con số còn có một phần gọi là ghi bằng chữ, công thức chúng tôi sẽ trình bày sau đây sẽ giúp các bạn thực hiện được điều này.

Công thức này gồm có 2 biến `strSourceValue` và `strUnitOfCalculate`. Trong đó `strSourceValue` lưu các con số cần chuyển đổi còn `strUnitOfCalculate` là chuỗi lưu đơn vị tính (đồng, cái, mét, kg ...). Vậy để sử dụng công thức này bạn chỉ cần 2 bước sau:

- Chuyển đổi giá trị Field mà bạn muốn chuyển đổi thành chuỗi và lưu vào biến `strSourceValue` (`strSourceValue` chỉ chứa các con số 0-9, không chứa bất kỳ ký tự khác).

- Xác định đơn vị tính của bạn bằng cách thay đổi trực tiếp giá trị biến strUnitOfCalculate.
-

Lưu ý: công thức chỉ xử lý được với giá trị tối đa là 999.999.999.999

```
' Công thức ToViewWord chuyển đổi giá trị số của một Field thành các từ Tiếng ' Việt
' Ví dụ
' Input:
'   Value = 15250005045
'   Đơn vị tính = "đồng"
' Output:
'   "Muỗi lăm ti hai trăm năm mươi triệu năm ngàn không trăm bốn mươi ' lăm đồng"
' Lưu ý: công thức chỉ xử lý được với giá trị tối đa là 999.999.999.999

Dim strSourceValue As String ' Chuỗi lưu các con số cần chuyển đổi
Dim strUnitOfCalculate As String ' Chuỗi lưu đơn vị tính (đồng, cái, _ mét, kg, ...)

' Bạn xác lập các giá trị cho công thức ở đây =====

' Nhiệm vụ của bạn ở đây là bổ sung thêm các câu lệnh để chuyển đổi Field cần sử lý về _
dạng chuỗi chỉ chứa những con số, không có các ký tự khác.
' Nếu Field có kiểu là Number thì dùng câu lệnh sau
strSourceValue = ToText({?TestNumber},"#")
strUnitOfCalculate = "đồng" ' Thay đổi đơn vị tính của bạn ở đây (mặc định là "đồng")

' Phần xử lý của công thức bạn không cần thay đổi gì cả =====

' Khai báo các biến và các hằng giá trị =====
Dim strResult As String ' Chuỗi lưu kết quả phân tích
Dim ThreeDigit As String ' Chuỗi lưu 3 chữ số

' Mảng lưu các đơn vị (const) tương ứng của 1 chữ số
Dim UnitArray1() As String
UnitArray1 = Array ("", "mười ", "trăm ")
Dim idUnit1 As Number ' idUnit1 = 1: hàng đơn vị 2: hàng chục 3: hàng trăm

' Mảng lưu các hằng đơn vị (const) tương ứng của cụm 3 chữ số
Dim UnitArray2() As String
UnitArray2 = Array ("", "ngàn ", "triệu ", "ti ")

' Mảng lưu các chữ số đơn lẻ đã được phân tích trong một cụm 3 chữ số
Dim OneWordArray(3) As String
' Stack lưu các cụm 3 chữ số đã được phân tích
Dim ThreeWordStack(4) As String
' Số lượng các phần tử có nghĩa trong ThreeWordStack
Dim nTop As Number
nTop = 1

' Bắt đầu xử lý =====
If Not IsNumeric(strSourceValue) Then ' Input không phải là giá trị số
    strResult = "Dữ liệu không hợp lệ !"
Else
    If ToNumber(strSourceValue) = 0 Then ' Giá trị Input = 0
        strResult = "Không " + strUnitOfCalculate
    Else
        If Len(strSourceValue) > 12 Then
            strResult = "Giá trị vượt quá con số 999 999 999 999!"
        Else
            'Convert chuỗi strSourceValue về dạng đầy đủ XXX XXX XXX XXX
            Dim lenght As Number
            lenght = Len(strSourceValue) Mod 3
            If lenght = 1 Then
                strSourceValue = "00" + strSourceValue
            Else 'lenght = 2
                strSourceValue = "0" + strSourceValue
            End If
        End If
    End If
End If
```

```

End If

' Bắt đầu xử lý chính việc chuyển đổi =====
While Len(strSourceValue) > 0
    ThreeDigit = Right(strSourceValue, 3) ' Lấy ra 3 chữ số cuối
    idUnit1 = 1 ' Bắt đầu là hàng đơn vị trong một cụm 3 chữ số
    Dim SingleWord As String
    Dim i As Number
    For i = 3 To 1 Step -1
        Select Case ThreeDigit(i)
            Case "1"
                SingleWord = "một " + UnitArray1(idUnit1)
                ' Con số đang xét là hàng chục
                If idUnit1 = 2 Then
                    SingleWord = "mười "
                End If
            Case "2"
                SingleWord = "hai " + UnitArray1(idUnit1)
            Case "3"
                SingleWord = "ba " + UnitArray1(idUnit1)
            Case "4"
                SingleWord = "bốn " + UnitArray1(idUnit1)
            Case "5"
                SingleWord = "năm " + UnitArray1(idUnit1)
                ' Con số đang xét là hàng đơn vị và giá trị hàng chục khác 0
                If idUnit1 = 1 And ThreeDigit(2) <> "0" Then
                    SingleWord = "lăm " + UnitArray1(idUnit1)
                End If
            Case "6"
                SingleWord = "sáu " + UnitArray1(idUnit1)
            Case "7"
                SingleWord = "bảy " + UnitArray1(idUnit1)
            Case "8"
                SingleWord = "tám " + UnitArray1(idUnit1)
            Case "9"
                SingleWord = "chín " + UnitArray1(idUnit1)
            Case Else ' Số 0
                ' Mặc định số 0 sẽ không hiển thị
                SingleWord = ""
                ' Con số đang xét là hàng chục và giá trị hàng trăm và hàng _
                đơn vị khác 0
                If idUnit1 = 2 And ThreeDigit(1) <> "0" _
                    And ThreeDigit(3) <> "0" Then
                    SingleWord = "lẻ "
                End If
                ' Con số đang xét là hàng trăm và giá trị hàng chục = 0 và _
                hàng đơn vị khác 0
                If idUnit1 = 3 And nTop = 1 And ThreeDigit(2) = "0" _
                    And ThreeDigit(3) <> "0" Then
                    SingleWord = "không trăm lẻ "
                End If
            End Select
            ' Kết thúc phân tích một chữ số trong cụm 3 chữ số
            idUnit1 = idUnit1 + 1
            ' Thêm chữ số vừa mới phân tích vào mảng
            OneWordArray(i) = SingleWord
        Next ' Kết thúc phân tích một cụm 3 chữ số của chuỗi strSourceValue

        ' Thêm cụm 3 chữ số vừa mới phân tích vào mảng
        ThreeWordStack(nTop) = OneWordArray(1) + OneWordArray(2) + _
            OneWordArray(3)

        nTop = nTop + 1
        ' Loại bỏ 3 chữ số cuối vừa mới phân tích
        strSourceValue = Left(strSourceValue, Len(strSourceValue) - 3)
    Wend ' Kết thúc phân tích chuỗi strSourceValue

    ' Lấy ra các phần tử trong ThreeWordStack theo Frist In Last Out
    nTop = nTop - 1
    For i = nTop To 1 Step -1
        If ThreeWordStack(i) <> "" Then
            strResult = strResult + ThreeWordStack(i) + UnitArray2(i)
        End If
    Next

```



```
        Next
        strResult = strResult + strUnitOfCalculate
    End If ' If Len(strSourceValue) > 12
End If ' If ToNumber(strSourceValue) = 0
End If ' If Not IsNumeric(strSourceValue)

Formula = strResult
```

Chương 4: Crystal Reports Object Model

Một tiến trình thiết kế các bản báo cáo thường bao gồm việc liên kết với các *dữ liệu nguồn* (data source) và việc bố trí những *đối tượng báo cáo* (report object) lên trên bản báo cáo theo nhiều *phân đoạn báo cáo* (report section) khác nhau. Một khi bạn đã thiết kế xong báo cáo thì dữ liệu mà nó in ra có thể thay đổi nhưng định dạng (format) của các đối tượng báo cáo sẽ được giữ nguyên mỗi khi báo cáo được hiển thị, việc này làm cho bản báo cáo trở nên gò bó. Tuy nhiên với Crystal Reports.NET thì giờ đây các bản báo cáo có thể mang tính uyển chuyển hơn làm cho bản báo cáo trở nên sinh động (dynamic) hơn. Lập trình viên .NET giờ đây có toàn quyền thâm nhập vào các thuộc tính của một bản báo cáo, các thuộc tính này đa phần là có thể bị thay đổi nếu bạn muốn. Do đó, bạn có thể làm chủ bản báo cáo bằng cách chặn lại các dữ liệu nhập vào và tùy biến lại (customize) trước khi chúng được in ra.

Lưu ý: Với Crystal Reports.NET, bạn không thể tạo mới những đối tượng đối với bản báo cáo mà chỉ có thể thay đổi gì đó trên các đối tượng hiện hữu của bản báo cáo. Ngoài ra bạn cũng không thể thay đổi các field dữ liệu đã được kết nối với cơ sở dữ liệu.

4.1 Các bước cơ bản để tùy biến bảng báo cáo

Việc tùy biến các bản báo cáo bao giờ cũng theo 3 bước sau đây:

- Khai báo một đối tượng ReportDocument.
- Thay đổi các thuộc tính.
- In hoặc Preview bản báo cáo.

Bước 1: Khai báo một đối tượng ReportDocument

Một biến đối tượng có thể là một thể hiện trực tiếp từ lớp báo cáo phát sinh khi bạn thiết kế bản báo cáo hoặc là một thể hiện của lớp **ReportDocument** (là lớp cơ sở cho các lớp báo mà bạn tạo ra) rồi sau đó bạn nạp một báo cáo cụ thể vào.

```
// Khai báo một biến đối tượng của lớp báo cáo phát sinh khi thiết kế
CrystalReport1 crMyReport = new CrystalReport1();
// Khai báo một biến đối tượng của lớp ReportDocument
ReportDocument rdMyReport = new CrystalReport1();
```

Hai biến trên là như nhau, vì chúng cùng qui chiếu về lớp CrystalReport1.

Lưu ý: CrystalReport1 là tên báo cáo mặc định mà **Report Expert** thường gán cho. Khi bạn viết các đoạn code mẫu vào ứng dụng của bạn, bạn hãy thay thế CrystalReport1 bằng tên lớp báo cáo mà bạn tạo ra.

Bước 2: Thay đổi các thuộc tính.

Như vậy, sau khi khai báo xong một biến đối tượng, tất cả các thuộc tính và phương thức của đối tượng **ReportDocument** đã sẵn sàng cho bạn sử dụng.

Trong ví dụ này, một công thức gọi là *công thức chọn lựa mẫu tin* (record selection formula) để xử lý sẽ bị thay đổi. Biến báo cáo crMyReport, nhận được trong bước 1, sẽ được sử dụng để truy cập vào đối tượng **DataDefinition** và thay đổi thuộc tính **RecordSelectionFormula** của đối tượng như sau:

```
crMyReport.DataDefinition.RecordSelectionFormula =  
"{HoaDon.NgayLapHoaDon}>#01/01/2006#";
```

Câu lệnh trên sẽ chỉ chọn xử lý những mẫu tin nào có ngày lập hoá đơn (HoaDon.NgayLapHoaDon) sau ngày 01/01/2006.

Bước 3: In hoặc Preview bản báo cáo

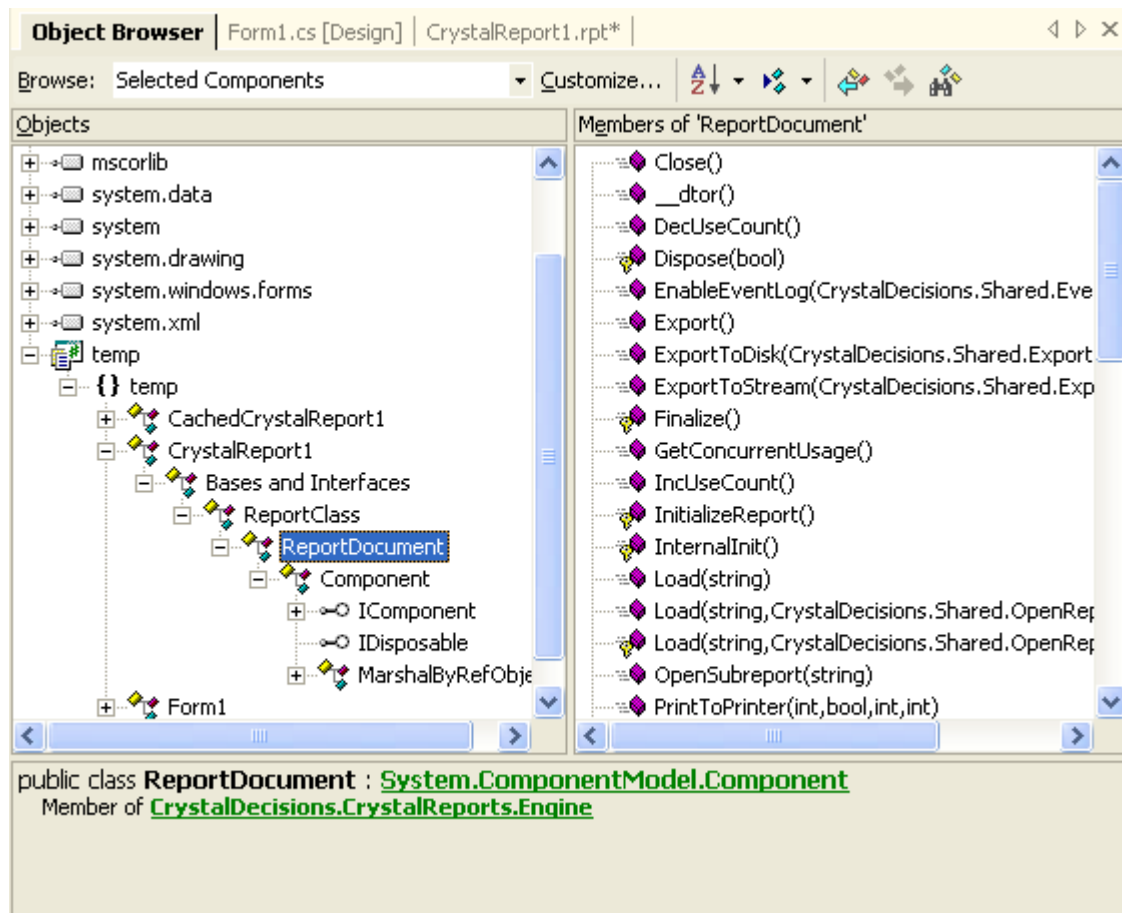
Muốn preview báo cáo bạn chỉ cần trao biến báo cáo, crMyReport cho ô control viewer. Còn muốn in ra giấy, bạn gọi hàm PrintToPrinter() của biến báo cáo.

```
// Preview bản báo cáo  
crystalReportViewer1.ReportSource = crMyReport();
```

4.2 Đối tượng ReportDocument

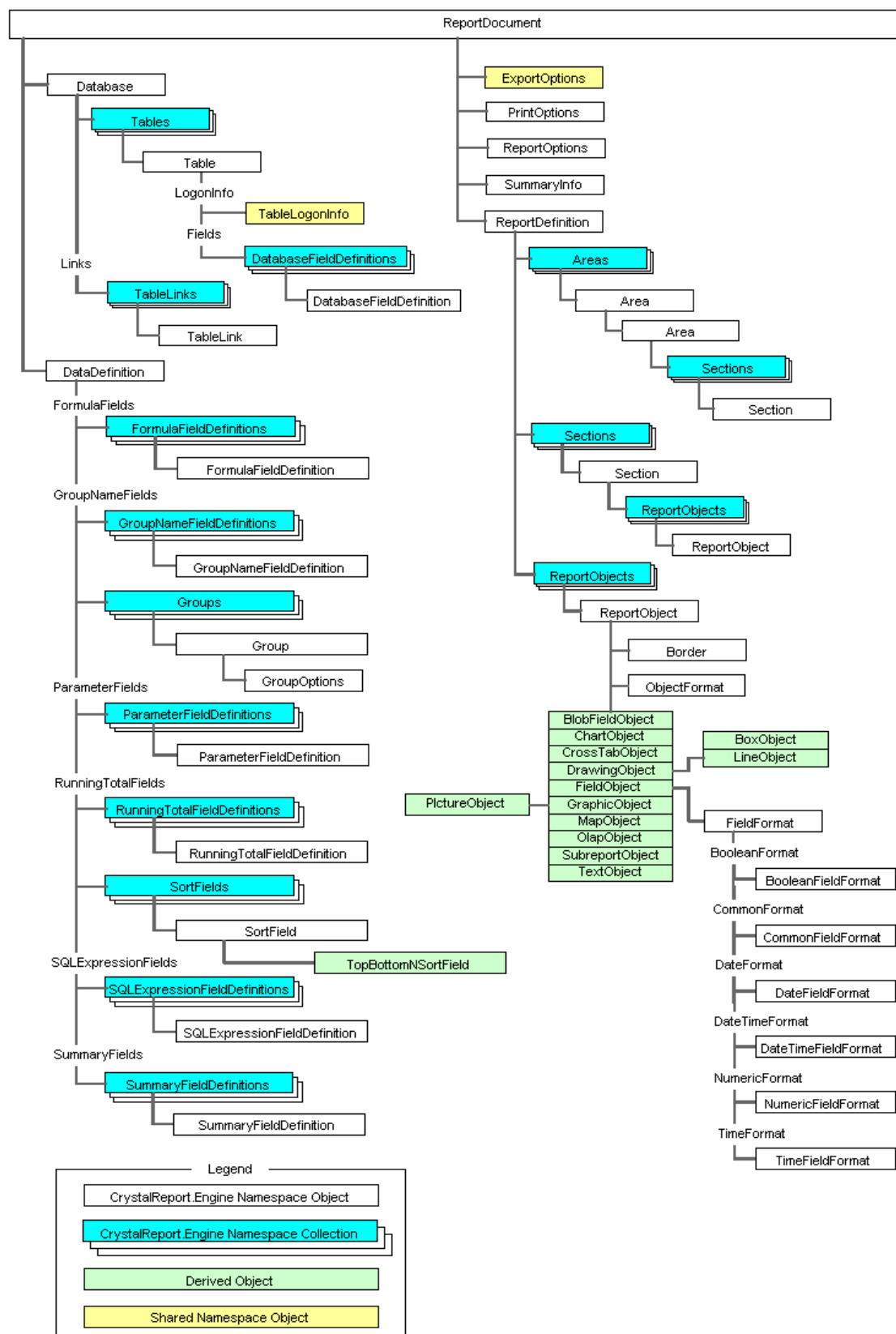
Lớp **ReportDocument** là lớp cơ sở đối với tất cả các báo cáo. Với phần lớn các thuộc tính này cho phép write, do đó bạn có thể thay đổi trị của các thuộc tính này.

Mỗi bản báo cáo là một lớp được kế thừa từ lớp **ReportDocument**. Hình 4-01 cho thấy cửa sổ Object Browser của lớp CrystalReport1. Bạn có thể thấy **ReportClass** là lớp cơ sở đối với báo cáo và **ReportDocument** là lớp cơ sở của **ReportClass**. Khung bên phải trên hình 4-01 cho liệt kê tất cả các member của lớp **ReportDocument** và khung bên dưới là namespace của lớp **ReportDocument**. (Nếu chưa có cửa sổ Object Browser bạn vào View | Object Browser hay nhấn tổ hợp phím Ctrl+Alt+J)



Hình 4-01: Object Browser View của lớp ReportDocument

Lớp **ReportDocument** còn có 7 lớp khác nhau như theo hình 4-02 và được giải thích theo bảng 4-01



Hinh 4-02: ReportDocument Object Model Diagram

Bảng 4-01: Các lớp thành phần của lớp ReportDocument

Lớp	Mô tả
DataBase	Lớp này cung cấp những thuộc tính cho phép truy xuất thông tin liên quan đến cơ sở dữ liệu trong bản báo cáo.
DataDefinition	Lớp này chứa tất cả các thông tin liên quan đến dữ liệu được xử lý dựa trên dữ liệu nguồn của báo cáo.
ExportOptions	Lớp này cung cấp những thuộc tính dùng để thiết lập các tùy chọn cho việc export một báo cáo.
PrintOptions	Lớp này cung cấp những thuộc tính và phương thức để thiết lập các tùy chọn liên quan đến in ấn.
ReportDefinition	Lớp này cho phép bạn truy xuất thông tin của các collection Areas , Sections và ReportObjects trong một báo cáo.
ReportOptions	Lớp này cho phép bạn truy cập thông tin dữ liệu liên quan đến các mục chọn báo cáo trong một báo cáo.
SummaryInfo	Lớp này cho phép bạn truy cập thông tin tóm tắt (hoặc tổng kết) trong một báo cáo.

Rất tiếc là trong phạm vi của bản báo cáo này không thể đi sâu tìm hiểu tất cả các lớp đối tượng trên, các bạn có thể tham khảo các đối tượng này dựa vào tài liệu tham khảo mà chúng tôi cung cấp.

Phần sau, chúng tôi sẽ giới thiệu lớp `DataDefinition`, theo chúng tôi nghĩ việc giới thiệu lớp đối tượng này rất có ích để các bạn có thể tùy biến bản báo cáo của mình.

4.3 Đối tượng DataDefinition:

Lớp này cung cấp cho chúng ta khả năng truy xuất đến các field như *parameter fields*, *sort fields*, *group-name fields*, *summary fields*, *running total fields*, và *SQL expression fields*. Nhờ đó ta có thể hiệu chỉnh các field này vào lúc thi hành và có thể được dùng để thay đổi dáng dấp của báo cáo. Ví dụ, bạn có thể thay đổi grouping field làm cho báo cáo được sắp xếp và tổng kết theo một cách mới.

4.3.1 Tìm hiểu các lớp Sorting và Grouping:

Trong một bản báo cáo thì tính năng sắp xếp và gộp nhóm có liên hệ chặt chẽ với nhau, nên các lớp thể hiện của chúng cũng có liên hệ mật thiết. Tất cả các lớp này đều có

thể truy xuất thông qua thuộc tính `DataDefinition` của đối tượng báo cáo. Hình 4-03 cho thấy mô hình các lớp sắp xếp và gộp nhóm. Phần lớn các lớp này có những thuộc tính thường là những hằng enum, như bạn có thể thấy trên hình 4-04 .

Sắp xếp và gộp nhóm đều dựa trên một đối tượng đó là `field`. `Field` sẽ ấn định thứ tự sắp xếp cũng như cách các mẫu tin sẽ được gộp nhóm như thế nào. Vì thế bạn phải hiểu cách chúng được tổ chức như thế nào trước khi bạn có thể làm việc với sắp xếp và gộp

nhóm.

Hình 4-03 : Mô hình đối tượng Sorting và Grouping

Hình 4-04 : Các hằng enumeration của Sorting và Grouping

Lớp **ReportDocument.DataDefinition** quản lý các nhóm bằng các collection **Groups** và **GroupNameFields** còn sắp xếp lại được quản lý bởi collection **SortFields**. Để duyệt qua các collection này bạn sử dụng chủ yếu là vòng lặp `foreach`.

Collection **SortFields** chứa các đối tượng có kiểu là **SortField**. Đối tượng **SortField** này có những thuộc tính cho biết `Field` dùng để sắp xếp, thứ tự sắp xếp tăng hay giảm (`SortDirection`), và `Field` sắp xếp là một mẫu tin hay một nhóm (`SortType`).

Collection **Groups** chứa các đối tượng **Group**, đối tượng này chỉ có duy nhất một thuộc tính `ConditionField` dùng để xác định field dùng để gộp nhóm. Điều này cho phép bạn nhanh chóng thay đổi field của `Group`.

Collection **GroupNameDefinitions** cho phép bạn thêm nhập trực tiếp vào các thuộc tính của nhóm. Tuy nhiên, collection này có một số trùng lặp vì nó cũng có một số qui chiếu về đối tượng `Group`. Trong lòng đối tượng `Group` là đối tượng `ConditionField`, mà đối tượng này lại cho bạn các thông tin đã có trong đối tượng **GroupNameDefinition** bao gồm `FormulaName`, `Name`, `Kind` và `ValueType`. Chính vì vậy để cho đơn giản chung tôi khuyên bạn nên dùng collection `Groups` để làm việc với các đối tượng nhóm.

4.3.2 Ví dụ Modify Sorting:

Ví dụ này tạo một Form như hình 4-05, dùng để liệt kê tất các Field đang được sắp xếp trên bản báo cáo vào danh sách Sort Fields và thứ tự sắp xếp tương ứng của mỗi Field. Bạn có thể thay đổi thứ tự này và cập nhật lại bản báo cáo với thứ tự mới. Form này sẽ không tự động chạy mà nó được gọi từ ứng dụng khi cần có thay đổi sự sắp xếp trên báo cáo, ứng dụng sẽ phải gán bản báo cáo vào biến thành viên có kiểu là **ReportDocument** của Form ModifySorting trước khi gọi nó thực hiện.



Hình 4-05: Hộp thoại Modify Sorting

1. Bạn tạo Form như hình 4-05 đặt tên là frmModifySorting. Bao gồm các thành phần chính như sau

- `ListBox lstSortFields;`
- `RadioButton otpAscending;`
- `RadioButton otpDescending;`
- `Button btnOK;`
- `Button btnApply;`
- `Button btnCancel;`

2. Trong ví dụ này ta sẽ sử dụng lớp **ReportDocument** và **SortDirection** nên bạn bổ sung các namespace sau vào form.

- `using CrystalDecisions.CrystalReports.Engine;`

- `using CrystalDecisions.Shared;`

Sau đó bạn khai báo một biến có kiểu là `ReportDocument`

- `public ReportDocument report;`

3. Tại sự kiện `Form_Load` ta sẽ duyệt qua collection `DataDefinition.SortField` của đối tượng `report` để lấy tên các Field được sắp xếp và Add vào `lstSortFields`.

```
private void frmModifySorting_Load(object sender, System.EventArgs e)
{
    try
    {
        foreach(SortField sortField in report.DataDefinition.SortFields)
        {
            lstSortFields.Items.Add(sortField.Field.Name.ToString());
        }
    }
    catch
    {
        MessageBox.Show("Report is null");
        this.Close();
    }
}
```

4. Tại sự kiện `SelectedIndexChanged` của đối tượng `ListBox lstSortFields` ta sẽ xác định giá trị `SortDirection` của Field đang được chọn trên danh sách và cập nhật lại trạng thái check cho 2 ô radio `otpAscending` và `otpDescending`.

```
private void lstSortFields_SelectedIndexChanged(object sender,
System.EventArgs e)
{
    // Duyệt qua các Field
    foreach(SortField sortField in report.DataDefinition.SortFields)
    {
        // Nếu đúng là Field đang được chọn trên ListBox
        if(sortField.Field.Name == lstSortFields.Text)
        {
            // Field này có thứ tự sắp xếp là AscendingOrder
            if(sortField.SortDirection == SortDirection.AscendingOrder)
            {
                otpAscending.Checked = true;
            }
            else
            {
                otpDescending.Checked = true;
            }
        }
    }
}
```

5. Tiếp theo ta sẽ thêm vào một hàm đặt tên là `Apply`, hàm này khi được gọi sẽ căn cứ vào Field nào đang được chọn trên `ListBox` và giá trị của 2 ô `otpAscending` và `otpDescending`. Từ đó cập nhật lại thuộc tính `SortDirection` của Field trong collection `SortFields`

```
private void Apply()
{
    foreach(SortField sortField in report.DataDefinition.SortFields)

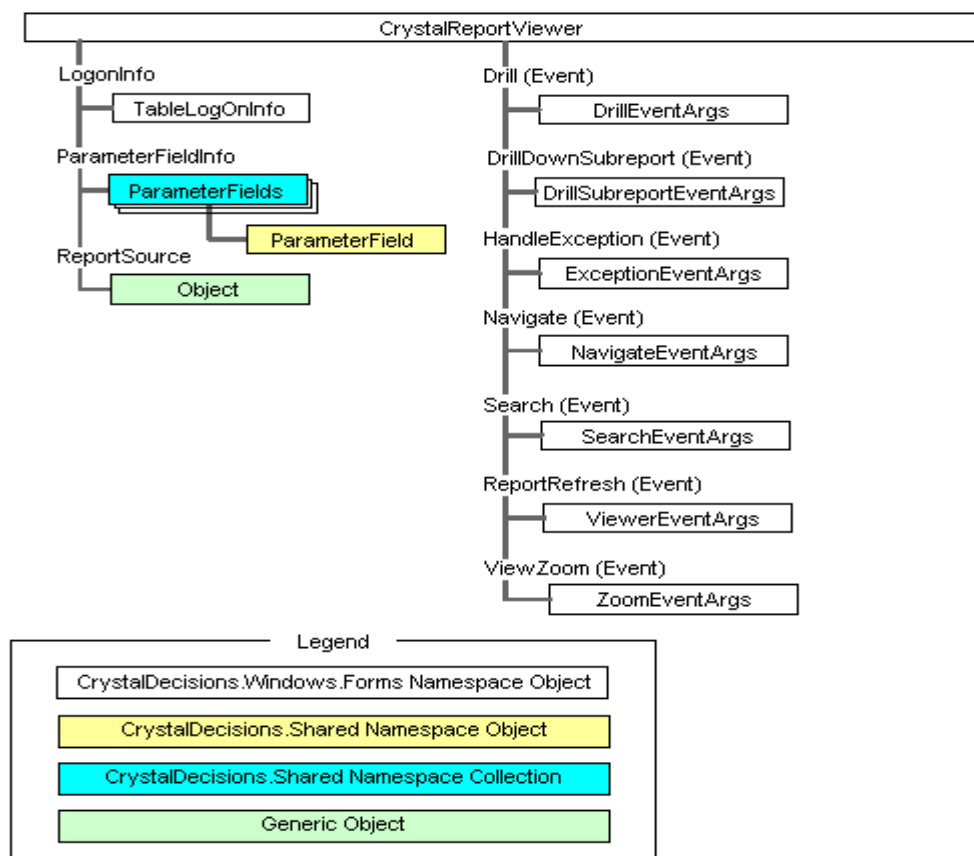
        if(sortField.Field.Name == lstSortFields.Text)
            if(otpAscending.Checked == true)
                sortField.SortDirection = SortDirection.AscendingOrder;
            else
                sortField.SortDirection = SortDirection.DescendingOrder;
}
```

6. Trong sự kiện Click của hai button OK và Apply bạn chỉ đơn giản là gọi hàm này thực hiện.

4.4 Đối tượng *CrystalReportViewer*:

Muốn preview các báo cáo phải được thông qua control **CrystaReportViewer**, Viewer này có thể được dùng để thay thế đối tượng **ReportDocument** để thay đổi các báo cáo tại thời điểm thi hành.

Hình 4-06 cho thấy mô hình đối tượng **CrystaReportViewer** và được giải thích theo bảng 4-02.



Hình 4-06: CrystalReportViewer Object Model Diagram

Bảng 4-02: Các lớp thành phần của CrystalReportViewer

Tên	Mô tả
TableLogOnInfo	Đối tượng này cung cấp các thuộc tính cho việc truy lại và thiết lập các tùy chọn kết nối table như <i>server name</i> , <i>user name</i> , <i>database name</i> , và <i>password</i> .
ParameterField	Đối tượng này cho phép bạn truy xuất và thay đổi thông tin của các tham số trong bản báo cáo.
ReportSource	Thuộc tính này dùng để thiết lập nguồn cho báo cáo.

Sau đây là ba bảng. Bảng 4-03 liệt kê các thuộc tính, bảng 4-04 liệt kê các phương thức và bảng 4-05 liệt kê các sự kiện của lớp **CrystaReportViewer**.

Bảng 4-03: Các thuộc tính của lớp CrystaReportViewer

Thuộc tính	Mô tả
------------	-------

ActiveViewIndex	Int32. Xác định giá trị index của view hiện hành trong viewer.
DisplayBackgroundEdge	Boolean. Xác định xem giữa bản báo cáo và cạnh của cửa sổ view có khoảng trống hay không.
DisplayGroupTree	Boolean. Xác định tree view được nhìn thấy hay ẩn.
DisplayToolBar	Boolean. Xác định thanh công cụ được nhìn thấy hay ẩn.
EnableDrillDown	Boolean. Xác định xem người dùng có thể drill down vào chi tiết trên báo cáo.
LogOnInfo	Lớp TableLogOnInfos . Truy xuất tập hợp TableLogOnInfos .
ParameterFieldInfo	Lớp ParameterFields . Truy xuất tập hợp parameter fields.
ReportSource	Lớp Object . Truy xuất nguồn của bản báo cáo.
RightToLeft	RightToLeft enum. Xác định xem văn bản xuất hiện từ phải qua trái hay không, chẳng hạn như dùng font chữ Do Thái hay Ả Rập.
SelectionFormula	String. Xác định công thức chọn mẫu tin cho bản báo cáo.
ShowCloseButton	Boolean. Xác định xem nút đóng một trang báo cáo có được xuất hiện trên thanh công cụ hay không.
ShowExportButton	Boolean. Xác định xem nút export có được xuất hiện trên thanh công cụ hay không.
ShowGotoPageButton	Boolean. Xác định xem nút GotoPage có được xuất hiện trên thanh công cụ hay không.
ShowGroupTreeButton	Boolean. Xác định xem nút GroupTree có được xuất hiện trên thanh công cụ hay không.
ShowPrintButton	Boolean. Xác định xem nút Print có được xuất hiện trên thanh công cụ hay không.
ShowRefreshButton	Boolean. Xác định xem nút Refresh có được xuất hiện trên thanh công cụ hay không.
ShowTextSearchButton	Boolean. Xác định xem nút TextSearch có được xuất hiện trên thanh công cụ hay không.
ShowZoomButton	Boolean. Xác định xem nút Zoom có được xuất hiện trên thanh công cụ hay không.
ViewCount	Int32. Cung cấp số lượng view trong viewer.

Bảng 4-04: Các phương thức của lớp CrystaReportViewer

Phương thức	Mô tả
CloseView	Đóng lại một tab view trong viewer.
DrillDownOnGroup	Drills down trên một group.

ExportReport	Exports bản báo cáo hiển thị trong viewer.
GetCurrentPageNumber	Lấy số trang hiện hành của bản báo cáo.
PrintReport	In bản báo cáo hiển thị trong viewer.
RefreshReport	Refresh bản báo cáo hiển thị trong viewer.
SearchForText	Tìm một mẫu văn bản nào đó trên bản báo cáo.
ShowFirstPage	Hiển thị trang đầu tiên của bản báo cáo.
ShowGroupTree	Hiển thị GroupTree trong viewer.
ShowLastPage	Hiển thị trang cuối của bản báo cáo.
ShowNextPage	Hiển thị trang kế tiếp của bản báo cáo.
ShowNthPage	Hiển thị trang thứ N trên bản báo cáo.
ShowPreviousPage	Hiển thị trang trước của bản báo cáo.
Zoom	Thay đổi độ phóng to/nhỏ của viewer.

Bảng 4-05: Các sự kiện của lớp *CrystaReportViewer*

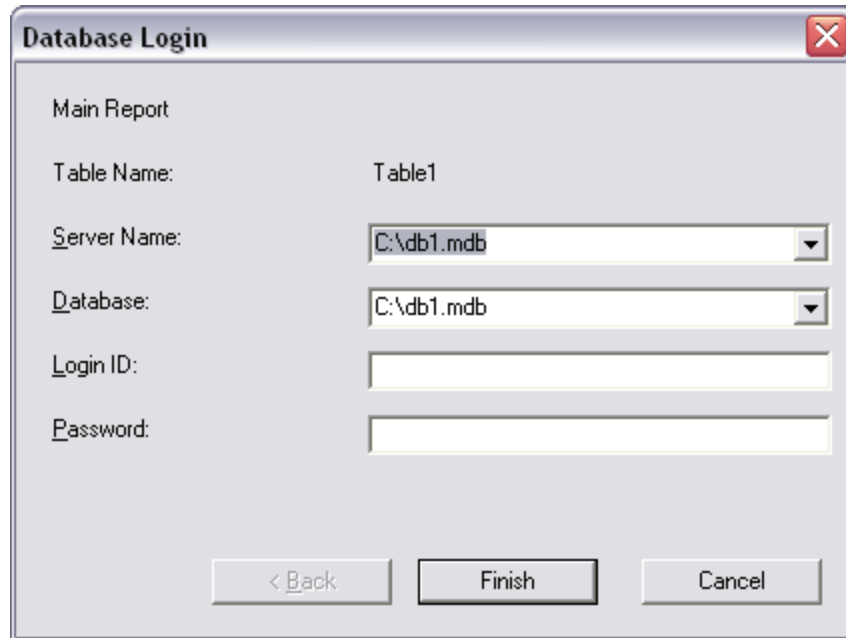
Sự kiện	Mô tả
Drill	Phát sinh khi bản báo cáo được drill down.
DrillDownSubreport	Phát sinh khi subreport được drill down.
HandleException	Phát sinh khi một biệt lệ xảy ra trong Viewer.
Navigate	Phát sinh khi người dùng di chuyển qua một trang khác trên bản báo cáo.
ReportRefresh	Phát sinh khi dữ liệu trong bản báo cáo được làm tươi.
Search	Phát sinh khi người dùng nhập vào một mẫu tìm kiếm.
ViewZoom	Phát sinh khi người dùng thay đổi độ phóng to/nhỏ.

Chương 5: Vấn đề kết xuất báo cáo trong trường hợp dữ liệu nguồn bị thay đổi

Khi tạo mới một báo cáo Crystal Reports, trình **Report Designer** (bộ thiết kế báo cáo) sẽ yêu cầu người dùng định nghĩa (hay chọn) dữ liệu nguồn. **Report Designer** sẽ tự động lưu lại các định nghĩa (bao gồm đường dẫn nếu là file access và sever, database name nếu là SQL server ...) và hiển thị các field ứng với dữ liệu nguồn được chọn hay định nghĩa trên thanh **Field Explorer**.

Khi nhúng báo cáo vào ứng dụng C#, Crystal Reports sẽ tự động lấy bảng dữ liệu nguồn đã được lưu sẵn trong báo cáo và thực hiện kết xuất báo cáo ra màn hình, máy in hay tập tin..

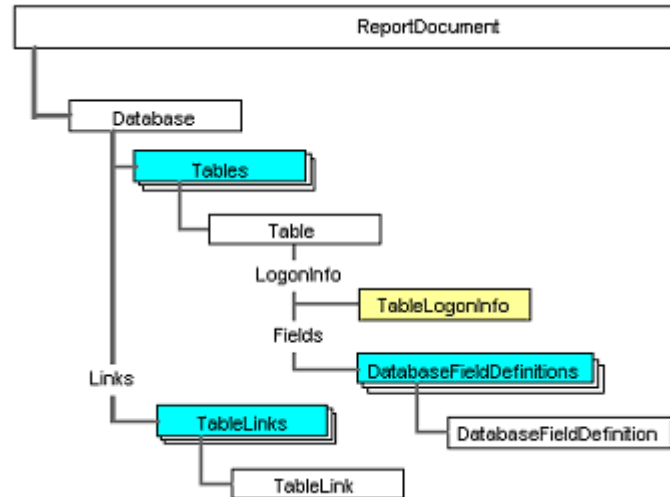
Vì vậy, khi dữ liệu nguồn bị thay đổi do report được di chuyển từ máy này sang máy khác, hay thư mục này sang thư mục khác, Crystal Reports sẽ không tìm thấy dữ liệu nguồn (do đường dẫn kết nối với access là đường dẫn vật lý hay tên server, tên database bị thay đổi). Khi đó Crystal Reports sẽ yêu cầu người dùng nhập lại các tham số để truy xuất đến dữ liệu nguồn thông qua hộp thoại **Database Login** như hình 5-01.



Hình 5-01: Hộp thoại Database Login

5.1 Lớp Database:

Lớp **ReportDocument**, như hình 5-02 bên dưới, lo quản lý các kết nối với các dữ liệu nguồn của báo cáo.



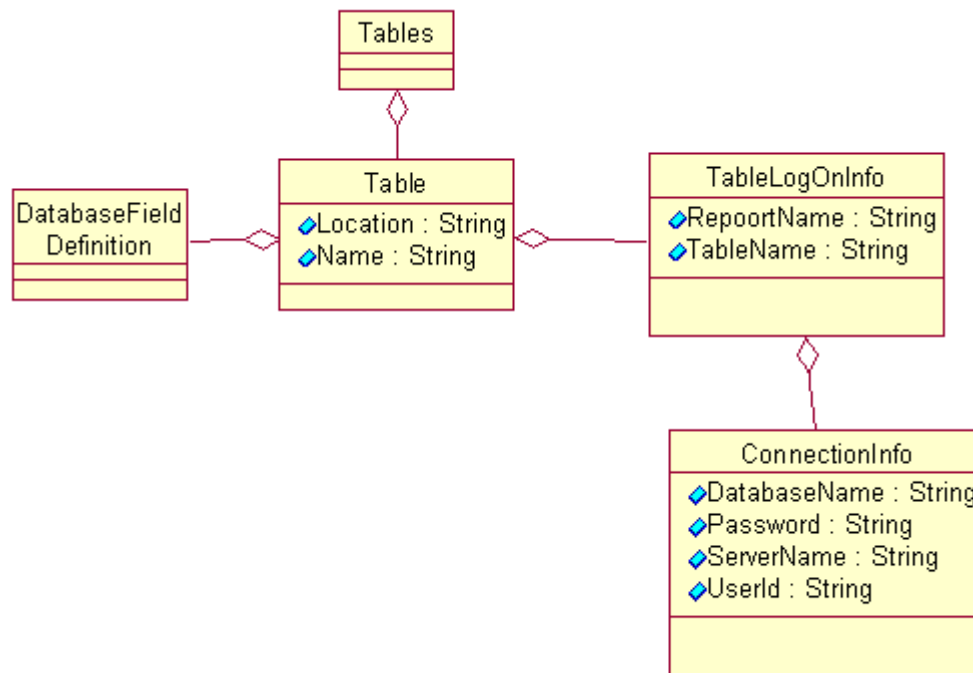
Hình 5-02: Lớp ReportDocument

Lớp **ReportDocument** có hai thuộc tính làm việc với dữ liệu nguồn là **Database** và **DataDefinition**. Thuộc tính đầu tiên **Database** là tập hợp các bảng dữ liệu(**Tables**) dùng trong báo cáo và mỗi gắn kết (**TableLinks**). Mỗi bảng dữ liệu đều có những thuộc tính dùng để đặt để thông tin có ý nghĩa đối với việc kết nối với bảng dữ liệu. Thuộc tính thứ hai là **DataDefinition**.

5.2 Thay đổi thông tin kết nối với bảng dữ liệu nguồn

Việc đăng nhập vào dữ liệu nguồn sẽ dùng đến hai lớp: **Table** và **ConnectionInfo**. Lớp **Table** là lớp chính. Ứng với mỗi table có đối tượng **TableLogOnInfo** dùng đi lấy một qui chiếu về đối tượng **ConnectionInfo**.

Đối tượng **ConnectionInfo** chứa các thông số dùng để kết nối với dữ liệu nguồn, xem hình 5-03.



Hình 5-03: Sơ đồ lớp Table

Đối tượng **ConnectionInfo** có những thông số để chỉ lại dữ liệu nguồn của report được giải thích theo bảng 5-01.

Bảng 5-01: Các thuộc tính của đối tượng ConnectionInfo

Stt	Tên thuộc tính	Mô tả
1	DatabaseName	Tên database của dữ liệu nguồn. Đối với MS Access là đường dẫn đến file .mdb
2	Password	Mật khẩu đăng nhập vào dữ liệu nguồn
3	ServerName	Tên Server chứa dữ liệu nguồn. Đối với MS Access là đường dẫn đến file .mdb
4	UserId	Tên đăng nhập người dùng

5.2.1 Sử dụng ReportDocument

Đoạn code dưới đây mô tả việc thay đổi thông tin kết nối của myReport. Nội dung thay đổi bao gồm: DatabaseName, Password, ServerName, UserId.

```
public void LoginToTables(string DatabaseName, string Password, _
    string ServerName, string UserId)
{
    MyCrystalReport myReport = new MyCrystalReport();
    CrystalDecisions.Shared.TableLogOnInfo myLogOnInfo;
    CrystalDecisions.CrystalReports.Engine.Table myTable;
    //Thay đổi logoninfo
    foreach (myTable in myReport.Database.Tables)
    {
        myLogOnInfo = myTable.LogOnInfo;
        myLogOnInfo.ConnectionInfo.DatabaseName = DatabaseName;
        myLogOnInfo.ConnectionInfo.Password = Password;
        myLogOnInfo.ConnectionInfo.ServerName = ServerName;
        myLogOnInfo.ConnectionInfo.UserId = UserId;
        myTable.ApplyLogOnInfo(myLogOnInfo);
    }
}
```

5.2.2 Sử dụng CrystalReportViewer

Việc đặt thông tin kết nối sử dụng viewer control (**CrystalReportViewer**) cũng dùng cùng đối tượng như đối tượng **ReportDocument**. Nó làm hơi khác một chút vì viewer control không có hiệu các bảng dữ liệu trong một báo cáo.

Collection **TableLogOnInfo** là đối tượng collection của viewer control. Đầu tiên, nó trống rỗng và nó cần có một đối tượng Table riêng rẽ cho mỗi bảng dữ liệu trên báo cáo. Chúng ta sẽ tạo ra một đối tượng Table mới cho mỗi bảng dữ liệu và cho đặt các thông tin kết nối của đối tượng **ConnectionInfo**. Sau đó, chúng ta dùng hàm Add() để thêm bảng dữ liệu vào collection **TableLogOnInfos**.

Đoạn code dưới đây mô tả việc kết xuất báo cáo myReport sử dụng viewer control. Đối với loại kết xuất này, chúng ta phải biết có bao nhiêu table trong dữ liệu nguồn và tên của chúng là gì. Ví dụ dưới đây sử dụng dữ liệu nguồn lấy từ 2 table 'Customer' và 'Order'

```
public void PrintPreview(string DatabaseName, string Password,
    string ServerName, string UserId)
{
    MyCrystalReport myReport = new MyCrystalReport();
    CrystalDecisions.Shared.TableLogOnInfo myLogOnInfo;
    rptViewer.LogOnInfo =
        new CrystalDecisions.Shared.TableLogOnInfos();
    //Tạo bảng Customer và đặt vào thuộc tính
    myLogOnInfo = new CrystalDecisions.Shared.TableLogOnInfo();
    myLogOnInfo.TableName = "Customer";
    myLogOnInfo.ConnectionInfo.DatabaseName = DatabaseName;
    myLogOnInfo.ConnectionInfo.ServerName = ServerName;
    myLogOnInfo.ConnectionInfo.Password = Password;
    myLogOnInfo.ConnectionInfo.UserID = UserId;
    rptViewer.LogOnInfo.Add(myLogOnInfo);
    //Tạo bảng dữ liệu Order và đặt vào thuộc tính
    myLogOnInfo = new CrystalDecisions.Shared.TableLogOnInfo();
    myLogOnInfo.TableName = "Order";
    myLogOnInfo.ConnectionInfo.DatabaseName = DatabaseName;
    myLogOnInfo.ConnectionInfo.ServerName = ServerName;
    myLogOnInfo.ConnectionInfo.Password = Password;
    myLogOnInfo.ConnectionInfo.UserID = UserId;
    rptViewer.LogOnInfo.Add(myLogOnInfo);
    //Hiển thị báo cáo
    rptViewer.ReportSource = myReport;
}
```

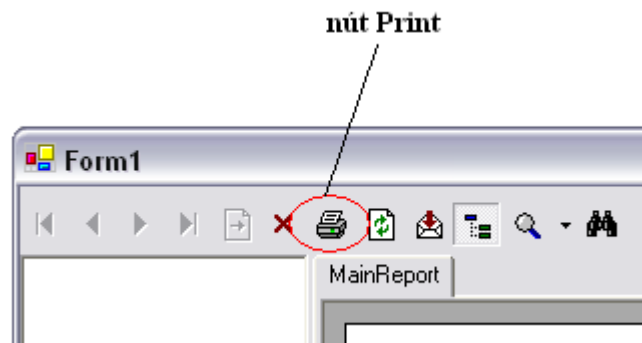
Chương 6: Kết xuất báo cáo ra máy in, word, excel:

Nhu cầu báo cáo là hiển thị thông tin dưới dạng văn bản hoàn chỉnh. Vì vậy, ngoài nhu cầu xem trên màn hình (Preview) thì còn có nhu cầu in báo cáo ra máy in, word, excel để làm tài liệu lưu trữ.

6.1 Xuất báo cáo thông qua Viewer

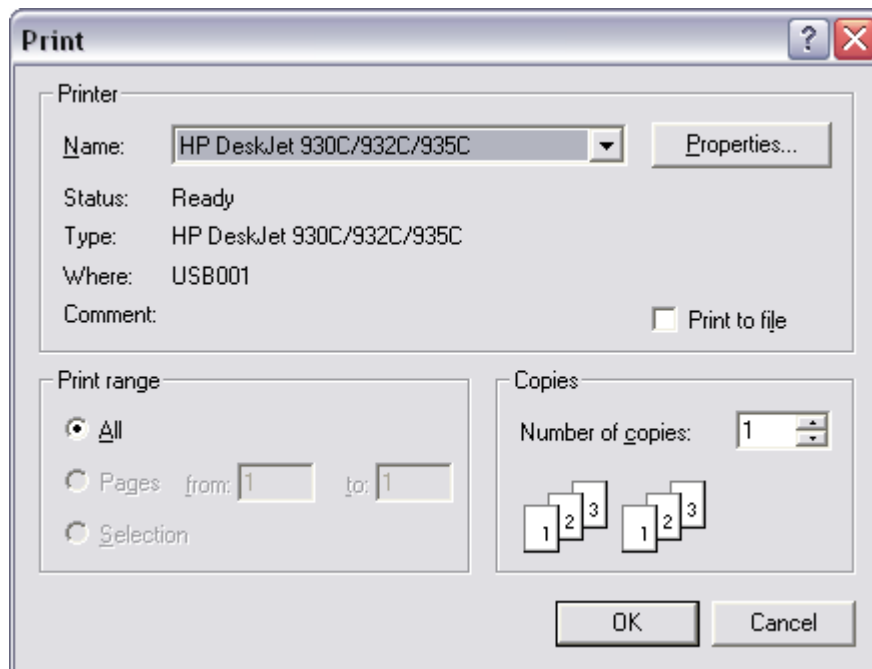
6.1.1 Ra máy in

Chọn nút print trên màn hình viewer control, như hình 6-01.



Hình 6-01: Nút Print trên màn hình Viewer

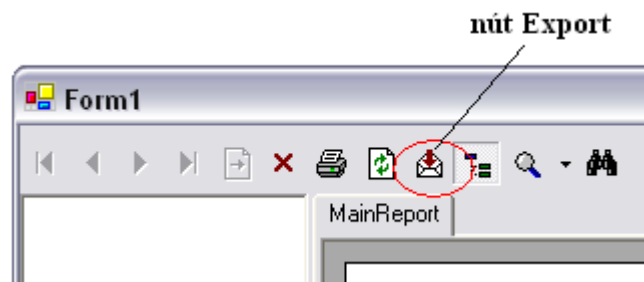
Hộp thoại **Print** (hình 6-02) hiển thị giúp bạn chọn thông số cho việc in báo cáo. Sau đó bạn nhấn nút **Ok** để tiến hành việc in ấn.



Hình 6-02: Hộp thoại Print

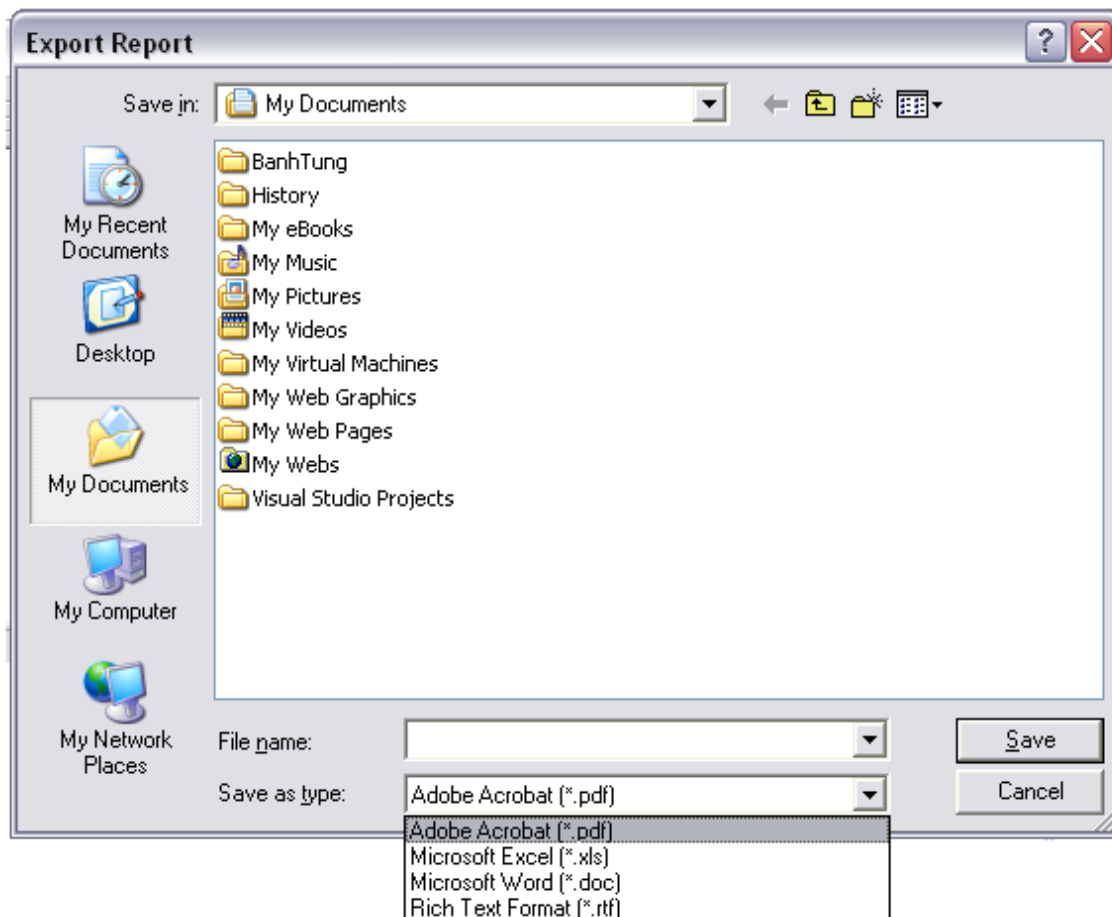
6.1.2 Ra tập tin

Nhấn nút Export trên viewer control như trên hình 6-03.



Hình 6-03: Nút Export trên màn hình Viewer

Hộp thoại **Export Report** sẽ hiển thị và yêu cầu bạn chọn dạng kết xuất ra và nhập tên tập tin kèm theo đường dẫn như hình 6-04. Sau đó bạn nhấn **OK** để tiến hành việc Export.



Hình 6-04: Hộp thoại Export Report

6.2 Xuất báo cáo thông qua lập trình

6.2.1 Ra máy in

Muốn in báo cáo ra máy in từ ứng dụng, lớp **ReportDocument** cung cấp một hàm đơn giản `PrintToPrinter` mà chúng ta có thể dùng in báo cáo từ máy in mặc nhiên. Hàm đòi hỏi 4 tham số: Số bản in, collation flag (boolean), trang bắt đầu, trang kết thúc.

Đoạn mã dưới đây cho phép tin báo cáo myReport ra máy in

```
private void Print_Button_Click(object sender, System.EventArgs e) {
    MyCrystalReport myReport = new MyCrystalReport();
    myReport.PrintToPrinter(1, true, 1, 999);
}
```

Trong trường hợp muốn làm chủ tiến trình In ấn, **ReportDocument** còn cung cấp đối tượng **PrintOptions**. Chi tiết của đối tượng này được mô tả như bảng 6-01 sau.

Bảng 6-01: Các thuộc tính của đối tượng PrintOptions

Stt	Thuộc tính	Mô tả
1	PageContentHeight	Int32. Trả về chiều cao nội dung các trang tính theo twips
2	PageContentWidth	Int32. Trả về chiều rộng nội dung các trang tính theo twips
3	PageMargins	PageMargins . trả về hoặc đặt topMargin, bottomMargin, leftMargin, rightMargin. Sử dụng hàm ApplyPageMargins để thực hiện các thay đổi này.
4	PaperOrientation	PaperOrientation . Trả về hoặc đặt chiều hướng giấy in trên máy in hiện hành. Các mục chọn là DefaultPaperOrientation, Landscape hoặc Portrait.
5	PaperSize	PaperSize . Trả về hay đặt khổ giấy in trên máy in hiện hành. Bao gồm: PaperA4, PaperLegal, PaperLetter...
6	PaperSource	PaperSource . Trả về hay đặt nguồn cung cấp giấy in trên máy in hiện hành. Hỗ trợ 13 khay giấy khác nhau gồm Auto(chọn tự động), Lower, Midder, Upper ...
7	PrinterDuplex	PrinterDuplex . Trả về hay đặt mục chọn duplex trên máy in hiện hành. Bao gồm các mục Default, Horizontal, Vertical,...
8	PrinterName	String. Trả về hay đặt tên máy in mà báo cáo dùng đến. Sẽ trả về chuỗi rỗng nếu máy in mặc nhiên được dùng.

6.2.2 Ra tập tin

Để kết xuất báo cáo ra tập tin, lớp **ReportDocument** cung cấp hàm `ExportToDisk`. Đây là một hàm rất phổ biến dùng để kết xuất ra tập tin và ghi lên đĩa. Hàm này nhận 2 tham số là `ExportFormatType` và đường dẫn tên tập tin cần kết xuất.

Các hằng enumeration `ExportFormatType` được giải thích theo bảng 6-02.

Bảng 6-02: Các hằng enumeration `ExportFormatType`

Stt	Dạng thức	File mở rộng	Enumeration Constant
1	Adobe Acrobat	.PDF	<code>PortableDocFormat</code>
2	Rich Text Format	.RTF	<code>RichText</code>
3	HTML 3.2	.HTM/.HTML	<code>HTML32</code>
4	HTML 4.0	.HTM/.HTML	<code>HTML40</code>
5	Microsoft Word	.DOC	<code>WordForWindows</code>
6	Microsoft Excel	.XLS	<code>Excel</code>

Đoạn code dưới đây kết xuất báo cáo myReport ra file Word

```
private void Export_Button_Click(object sender, System.EventArgs e) {
    MyCrystalReport myReport = new MyCrystalReport();
    myReport.ExportToDisk_
        (CrystalDecisions.Shared.ExportFormatType.WordForWindows, "D:\\Report.doc");
}
```

Tài liệu tham khảo

- MSDN 01-2004
- Lập trình báo cáo dùng Crystal Report .NET và C# - Dương Quang Thiện