

	HƯỚNG DẪN CODING	Mã hiệu: SOF.QP.02.M03
		Lần BH/SĐ:
		Ngày BH:

	SOẠN THẢO	XEM XÉT	PHÊ DUYỆT
Chữ ký			
Họ & tên	Lê Phát Danh		
Ngày tháng	19/06/2015		

I. MỤC ĐÍCH:

Tài liệu hướng dẫn cấu trúc Framework, cách coding và các quy định cần lưu ý khi viết code. Mục đích để nhân viên lập trình tham khảo khi thực hiện tính năng mới hoặc sửa lỗi.

II. PHẠM VI:

Áp dụng cho nhân viên lập trình sản phẩm ASoft-Accounting

III. ĐỊNH NGHĨA, TỪ VIẾT TẮT:

- SVN: công cụ quản lý tài liệu
- TL: Trưởng nhóm phát triển sản phẩm
- NV.DEV: Nhân viên lập trình
- NV.BA: Nhân viên phân tích

IV. NỘI DUNG:

1. Lưu đồ: Không

2. Thời điểm thực hiện

Theo tiến độ TL đã lên hoặc khi có yêu cầu thực hiện tính năng mới hoặc sửa lỗi.

3. Input

INPUT	KIỂM TRA
Phiếu phân tích yêu cầu SOF.QP.04.F01 đã hoàn thành	Các xử lý mô tả đúng với Tài liệu thiết kế giao diện SOF.QP.02.F01
Thông tin lỗi trên OPENERP	Thông tin mô tả rõ ràng, đầy đủ

4. Resource

RESOURCE	VAI TRÒ
NV.DEV	Thực hiện việc phát triển
NV.BA	Hỗ trợ nghiệp vụ và các xử lý trong Database

5. Output

OUTPUT	KIỂM TRA
Source code lưu trữ trên SVN	Tuân thủ các quy định từ mục [7.Môi trường phát triển] trong tài liệu này trở xuống.
Sản phẩm	Bàn giao cho NV.BA

6. Hướng dẫn thực hiện

Step 1: Chuẩn bị

CÔNG VIỆC	EXCEPTION/RISK	CHECK/GHI CHÚ
<p>NV.DEV xác định công việc sẽ làm là lỗi hay tính năng mới.</p> <ul style="list-style-type: none"> - Trường hợp là lỗi thì xem thông tin chi tiết lỗi trên OPENERP. - Trường hợp là tính năng mới, đọc tài liệu có tên tương ứng với tính năng cần phát triển trên SVN, đường dẫn: [SVN\10_DOCUMENT\11_DESIGN\Business] để nắm khái quát thông tin nghiệp vụ. 	/	<p>Tham khảo tài liệu Hướng dẫn phân tích SOF.QP.02.M02 để biết cách đọc tài liệu thiết kế chi tiết</p>

Step 2: Xác định yêu cầu

CÔNG VIỆC	EXCEPTION/RISK	CHECK/GHI CHÚ
<p>Trường hợp lỗi: Kiểm tra thông tin lỗi trên OPENERP, chỗ nào chưa rõ xác nhận lại với TL hoặc NV.BA</p> <p>Trường hợp tính năng mới: đọc hiểu tài liệu thiết kế chi tiết do NV.BA viết chỗ nào chưa rõ hoặc cần chỉnh sửa xác nhận lại với TL hoặc NV.BA</p>	/	/

Step 3: Liệt kê công việc cần làm

CÔNG VIỆC	EXCEPTION/RISK	CHECK/GHI CHÚ
<p>Lên danh sách chi tiết các công việc cụ thể để thực hiện việc sửa lỗi hoặc làm tính năng mới và các rủi ro có thể xảy ra khi phát triển</p> <p>Đánh độ ưu tiên các công việc theo thứ tự</p>	/	Việc nào quan trọng, khó, độ rủi ro cao ưu tiên làm trước

Step 4: Thực hiện

CÔNG VIỆC	EXCEPTION/RISK	CHECK/GHI CHÚ
-----------	----------------	---------------

Thực hiện coding. Tham khảo từ mục [7. Môi trường phát triển] trở xuống để biết mô hình Framework, cách coding và các quy định khi phát triển.	Trong quá trình phát triển có thể phát sinh thêm hoặc chỉnh sửa từ mục [7.Môi trường phát triển] trở xuống. Nếu có thay đổi thì báo lại với TL và cập nhật tài liệu.	/
---	---	---

Step 5: Kiểm tra

CÔNG VIỆC	EXCEPTION/RISK	CHECK/GHI CHÚ
Tự test lại các nghiệp vụ có liên quan trong quá trình phát triển, đảm bảo không để lỗi cơ bản xảy ra sau khi release.	/	/

7. Môi trường phát triển

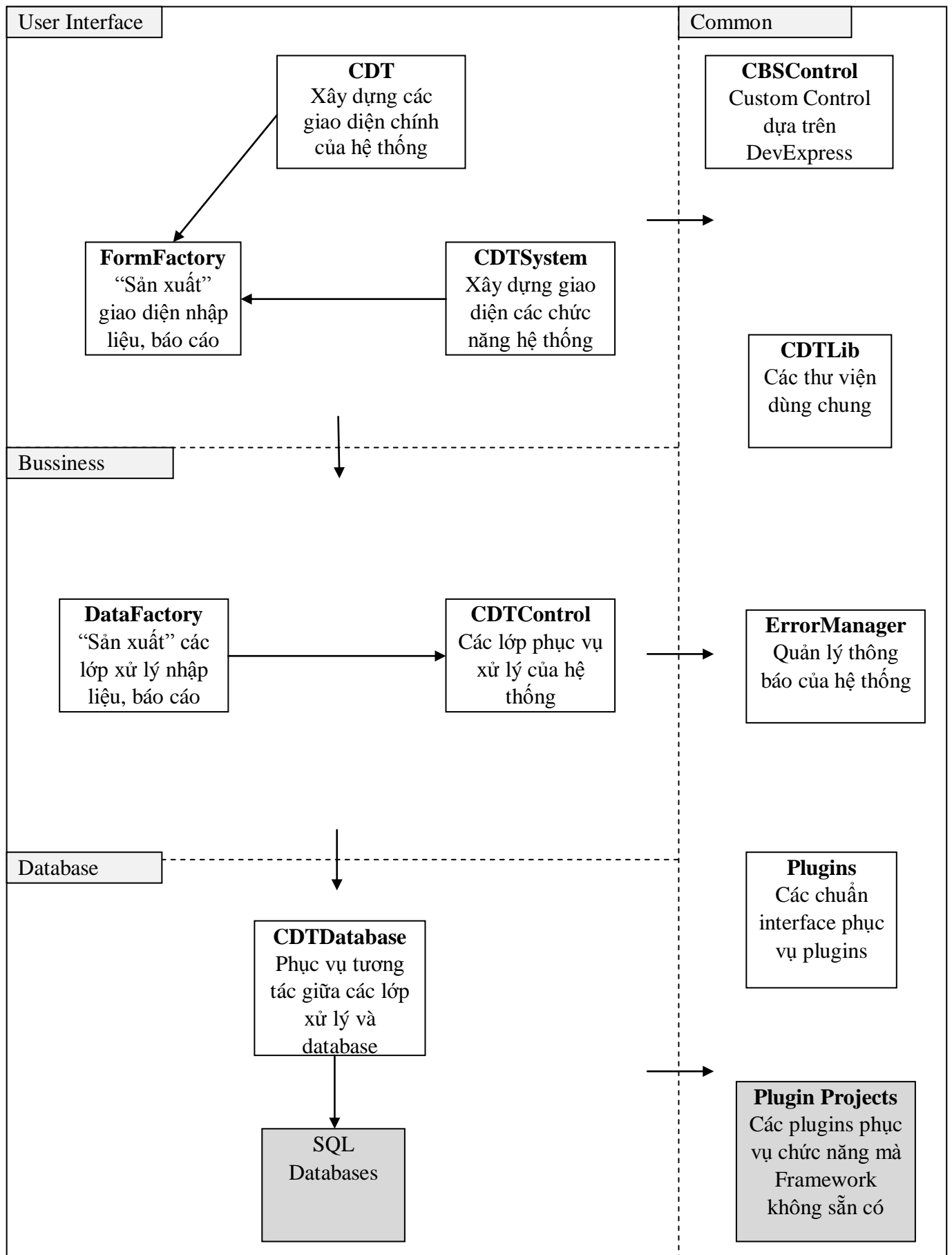
7.1 Source code

Môi trường phát triển	Microsoft Visual Studio .Net 2008
Ngôn ngữ sử dụng	C#
Công cụ sử dụng để lập trình	
Thư viện hỗ trợ	DevExpress 11.1.4.0

7.2 Database

DMBS	Microsoft SQL Server 2005 Express
Công cụ thao tác DB	SQL Server Management Studio Express (SSMSE)

7.3 Mô hình tổ chức source code



Lưu ý:

- Hai projects quan trọng thực hiện hầu hết các xử lý thông thường: FormFactory và DataFactory.
- Plugin Projects thực hiện các chức năng khác, hiện tại hệ thống cung cấp 4 interfaces sau đây:
 - o Loại IC là một chức năng hoàn toàn riêng biệt gắn thẳng vào menu chính của hệ thống
 - o Loại ICFORM là một chức năng gắn thẳng vào menu xử lý trên một giao diện nhập liệu
 - o Loại ICCONTROL là chức năng xử lý riêng cho các control trên giao diện nhập liệu (thường là event)
 - o Loại ICData là chức năng cho phép can thiệp vào xử lý trước và sau thao tác lưu trữ số liệu trên giao diện nhập liệu

7.4 Các quy định khi phát triển chương trình**1) Các module trong chương trình**

Theo cách đặt tên thường thấy của Microsoft Visual Studio .Net 2005

Riêng với các table, đặt tên theo chuẩn nghiệp vụ:

- Các danh mục: DM...
- Các bảng master: MT...
- Các bảng detail: DT...

2) Quy định cách đặt Namespace

Đặt tên theo mô tả chức năng, mỗi project tương ứng 1 namespace.

3) Quy định cách đặt tên Class | Object

Đặt tên theo mô tả chức năng

4) Các file cấu hình

- Một file cấu hình chứa thông số riêng cho mỗi máy: app.config
- Một table nằm trong database CDT chứa cấu hình chung: sysConfig

5) Quy định cách đặt mã Message

Điểm yếu hiện nay của hệ thống là Message chưa được hệ thống theo mã.

6) Quy định về cách đặt tên các loại plugins

Quy định chung là tên namespace của mỗi plugin trùng với tên của class giao tiếp của plugin đó. Class giao tiếp sẽ kế thừa Interface dùng cho loại plugin đó.

Để hệ thống dễ dàng phân biệt các plugins của mình, Assembly name quy định như sau:

- IC: Namespace + “_IC”
- ICFORM: Table + “_” + FormType + “_ICF_” + Namespace
- ICCONTROL: Table + “_” + FormType + “_ICC_” + Namespace
- ICData: Table + “_” + FormType + “_ICD_” + Namespace

Trong đó FormType là các loại form nhập liệu mà hệ thống hỗ trợ:

- Single: Sg
- SingleDt: SgDt
- MasterDetail: Md
- MasterDetailDt: MdDt

- Detail: Dt

7) Quy định update server script

- Tất cả các update liên quan đến database đều phải đặt trong thư mục trên SVN theo đường dẫn [SVN\20_DATABASE\203_UpdateDB]
- Các script liên quan đến database CDT đặt trong thư mục [SVN\20_DATABASE\203_UpdateDB\01_CDT]
- Các script liên quan đến database số liệu kế toán đặt trong thư mục [SVN\20_DATABASE\203_UpdateDB\02_STDPRO]
- Các script phải được đánh số thứ tự tăng dần, script mới nhất sẽ có số thứ tự cao nhất
Ví dụ: [SVN\20_DATABASE\203_UpdateDB\02_STDPRO\20_Update 122011]
- Script liên quan đến việc tạo hoặc chỉnh sửa table, view, store, function, trigger đặt trong thư mục trên SVN (tên thư mục có prefix từ 01 – 06)
Ví dụ:
[SVN\20_DATABASE\203_UpdateDB\01_CDT\9999_NewScripts\01_TABLES\]
hoặc
[SVN\20_DATABASE\203_UpdateDB\02_STDPRO\9999_NewScripts\01_TABLES\]
Lưu ý: trước khi tạo table mới phải kiểm tra [Not exists], không được DROP TABLE sẽ mất dữ liệu của khách hàng.
- Script liên quan đến việc cập nhật dữ liệu đặt trong thư mục trên SVN đường dẫn:
[SVN\20_DATABASE\203_UpdateDB\01_CDT\9999_NewScripts\99_DATA]
hoặc
[SVN\20_DATABASE\203_UpdateDB\02_STDPRO\9999_NewScripts\99_DATA]
- Tất cả các script update database liên quan đến thông tư 200 đều phải chứa trong thư mục trên SVN, đường dẫn:
+ CDT:
+ STDPRO:
[SVN\20_DATABASE\203_UpdateDB\02_STDPRO\9999_NewScripts\99_DATA\12_TT200.sql]
- **Không được tạo thư mục trùng tên với các thư mục update TT244, TT138**
- **Không được tạo file script có tên chứa chuỗi “TT200”**
- **Khi viết script nên tham khảo các script đã có sẵn để tránh sai sót.**

8) Quy định lưu dạng số lẻ xuống DB

Mục tiêu: Màn hình hiển thị như thế nào thì lưu xuống DB y như vậy. Thống nhất số liệu dạng số lẻ trong hệ thống.

- Các Control binding dữ liệu trên DataSet hoặc DataTable cần FormatString và Formatvalue các control đó
Ví dụ: Lưới bind data, các cột dạng số cần format định dạng khi hiển thị cũng như lưu xuống bằng thuộc tính FormatString và event ParseEditValue (*Tham khảo class LaySoLieu0328 trong plugin ToKhaiThueTNDN*)
- Khi tạo mới hoặc update số liệu bằng class CDTDatabase.Database, lưu ý làm tròn số theo thiết định định dạng số trong hệ thống.
+ Hàm UpdateDataSet, UpdateDataTable: Khi thay đổi giá trị trong DataTable cần làm tròn số trước khi lưu xuống DB. Sử dụng hàm Utils.RoundDecimalByFieldName
Ví dụ:

```
DataRow dr = _dtData.Rows[0];
dr["Tien"] =
Utils.RoundDecimalByFieldName(Utils.parseDecimal(dr["Tien"]), "Tien")
_database.UpdateDataTable(query, _dtData)
```

+ Hàm UpdateByNonQuery: Thêm function ROUND trong chuỗi query sẽ truyền vào (tham khảo class GiaNTEXT trong plugin Inventory), hoặc dùng hàm Utils.RoundDecimalByFieldName để làm tròn giá trị trước khi đưa vào chuỗi query để execute.

7.5 Case-study các chức năng trên phần mềm

Mục này sẽ hướng dẫn theo kiểu case-study các chức năng điển hình trên phần mềm. Với mỗi chức năng, sẽ đi vào mô tả nhóm chức năng chạy từ CDT, nhóm chức năng chạy bằng plugin. Cụ thể ở mỗi chức năng đều có liệt kê các dữ liệu tương ứng trong CDT và các class tương ứng trong source-code. Mục này sẽ giúp developer nắm bắt source-code, đồng thời cũng là một tài liệu tra cứu trong quá trình phát triển phần mềm này.

Lưu ý là các case-study sẽ được mô tả từ đơn giản đến phức tạp. Các case-study sau sẽ không mô tả lại chức năng đã mô tả ở case-study trước mà chỉ bổ sung các chức năng chưa mô tả. Vì vậy nên tìm hiểu từ trên xuống dưới để dễ nắm bắt.

1) Nhập danh mục và số dư

- Ví dụ thực tế: Vào số dư đầu kỳ công nợ phải trả

- Các chức năng chạy từ CDT:
 - o Tên bảng dữ liệu: OBKH.
 - o Kiểu nhập liệu của bảng: Bảng đơn nhập trên màn hình riêng. Tương ứng với lớp FrmSingle, FrmSingleDt và FormDesigner trong FormFactory, lớp DataSingle trong DataFactory.
 - o Nhập thông tin bằng cách chọn từ danh mục: Tài khoản, đối tượng, mã ngoại tệ. Tương ứng với lớp CDTLookup và GrdSingle trong FormFactory. Dữ liệu trong “Danh sách bảng dữ liệu” của CDT như sau:
 - Kiểu dữ liệu: Khóa ngoại kiểu chuỗi
 - Bảng tham chiếu: bảng danh mục
 - Trường tham chiếu: khóa chính của bảng danh mục
 - Lọc cho tham chiếu: câu điều kiện sql khi select bảng danh mục
 - Hiển thị cho tham chiếu: hiển thị thêm trường “Tên” bên cạnh trường “Mã” (ví dụ như tên đối tượng ở màn hình trên)
 - o Các thông tin chỉ hiển thị ở chế độ “Mở rộng”: Bộ phận, Hợp đồng, Khoản mục phí, Công trình. Tương ứng với lớp SmartView trong FormFactory. Dữ liệu trong “Danh sách bảng dữ liệu” của CDT như sau:
 - Hiện: true

- SmartView: false

- Các chức năng chạy bằng plugin:
 - Nếu chọn mã ngoại tệ khác với VND, hiển thị dư nợ nguyên tệ và dư có nguyên tệ. Dùng plugin All_SgDt_ICC_MaNgoaiTe2.dll: Plugin này sẽ thực thi trên tất cả các bảng nhập thông tin cho danh mục. Loại plugin là Custom Control: Add event cho control Mã ngoại tệ để xử lý. Tương ứng với plugin “MaNgoaiTe2”.

- Cập nhật lại số dư tài khoản công nợ vào bảng số dư tài khoản. Dùng plugin OBKH_Sg_ICD_OBKH.dll: chỉ thực thi trên bảng OBKH. Loại plugin là Custom Data, tương ứng với plugin “OBKH”.

2) Nhập chứng từ đơn giản

- Ví dụ thực tế: Nhập phiếu thu

Phiếu thu

*Ngày chứng từ: 28/02/2010 Mã nghiệp vụ: TKH

*Số chứng từ: CPT3-02 *Diễn giải: Thu tiền của khách hàng

*Đối tượng: KD0002 *Mã ngoại tệ: VND

*Tên đối tượng: Công ty TNHH Đông Tiến Hưng *Tài khoản nợ: 1111

Chọn dòng: Nhấp chuột vào Stt - đồng thời giữ Shift để chọn nhiều dòng; Delete: Xóa dòng chọn; F5: Sao chép dòng cuối

Stt	Phát sinh	Tài khoản có	Khách hàng	Tên khách hàng	Diễn giải
>	5 000 000	1311	KD0002	Công ty TNHH Đông Tiến Hưng	Thu tiền bán hàng
*					

Tổng tiền: 5 000 000

Báo cáo nhanh

1/1

Cơ bản Mở rộng

- Các chức năng chạy từ CDT:
 - Tên bảng dữ liệu: MT11 và DT11.
 - Kiểu nhập liệu của bảng MT11 là bảng chính; bảng DT11 là bảng phụ thuộc nhập cùng bảng chính, chỉ rõ bảng chính là bảng MT11. Tương ứng với các lớp FrmMasterDetail, FrmMasterDetailDt, FormDesigner trong FormFactory, lớp DataMasterDetail trong DataFactory.
 - Các xử lý để tự động chọn tài khoản nợ, tài khoản có sau khi chọn Mã nghiệp vụ, lấy tỷ giá mặc định sau khi chọn loại tiền ngoại tệ...
 - Tương ứng với các hàm SetValuesFormList trong các lớp FormDesigner của FormFactory và CDTData trong DataFactory.
 - Dữ liệu trong “Danh sách bảng dữ liệu” của CDT là “Công thức nhập”. Công thức này được quy ước là “Tên của trường mã”. Tên của trường cần lấy dữ liệu”.
 - Các công thức quy đổi tiền VNĐ từ tiền ngoại tệ, tính tổng tiền từ phát sinh...
 - Tương ứng với FormulaCaculator trong CDTControl.
 - Dữ liệu trong “Danh sách bảng dữ liệu” của CDT là “Công thức tính”. Công thức này hỗ trợ các toán tử thông thường, có thứ tự ưu tiên và hỗ trợ hàm sum. Quy ước là phía trước tên biến có ký tự “@”.
 - Chức năng hạch toán vào bảng sổ cái để lên báo cáo kế toán:
 - Tương ứng với lớp DataTransfer trong CDTControl.
 - Dữ liệu trong CDT:
 - Cấu hình dòng dữ liệu: Bảng tổng hợp là BLTK, bảng chính là MT11, bảng chi tiết là DT11, mỗi record tương ứng với 1 dòng hạch toán trong sổ cái.
 - Cấu hình chi tiết dòng dữ liệu: trường nào của bảng tổng hợp sẽ đi với trường nào của bảng chính và trường chi tiết để truyền

dữ liệu. Nếu cần tính toán rồi mới đưa vào một trường của bảng tổng hợp thì nhập vào “Công thức”.

- Chức năng báo cáo nhanh:
 - Tương ứng với lớp QuickReport trong FormFactory.
 - Dữ liệu trong CDT:
 - Quản lý tra cứu nhanh: Bảng số liệu là “Chi tiết phiếu thu”, Báo cáo tương ứng là Tổng hợp số dư công nợ và Sổ chi tiết công nợ.
 - Thiết lập tham số tra cứu nhanh: trường nào của tham số báo cáo sẽ đi với trường nào của dữ liệu trong phiếu chi. Nếu chưa đủ dữ liệu cho các tham số bắt buộc, sẽ xuất hiện màn hình nhập tham số trước khi lên báo cáo.

- Các chức năng chạy từ plugins:
 - Không cho lưu số liệu nếu thêm hay sửa phiếu thu thuộc về một kỳ đã khóa sổ: Dùng plugin All_MdDt_ICD_BlockData.dll. Plugin này chạy trên tất cả các màn hình nhập chứng từ. Loại plugin là CustomData và có tên là “BlockData”.

3) Nhập chứng từ phức tạp

- Ví dụ thực tế: Hóa đơn bán hàng

Hóa đơn bán hàng kiêm phiếu xuất kho

*Ngày chứng từ: 14/02/2010 *Diễn giải: Xuất bán hàng

*Số chứng từ: HDB10/002 *Mã ngoại tệ: VND

*Ngày hóa đơn: 14/02/2010 *Mã thuế: T10

*Số hóa đơn: 123234 *Tài khoản nợ: 1311

*Số seri: AK/2007N *Tài khoản thuế: 33311

*Khách hàng: KD0002 *Tên khách hàng: Công ty TNHH Đông Tiến Hưng

Chọn dòng: Nhấp chuột vào Stt - đồng thời giữ Shift để chọn nhiều dòng; Delete: Xóa dòng chọn; F5: Sao chép dòng cuối

Stt	Mã kho	Vật tư	Tên vật tư	Đơn vị tính	Số lượng	Giá bán	Thành tiền	Chiết khấu	Tài khoản doanh thu
1	K02	TP-KH-SY-001	Thành phẩm khung xe SYM 1	CA	10.	850 000.	8 500 000	333 117	5112
2	K02	TP-KH-HO-001	Thành phẩm khung xe Honda 1	CA	18.	900 000.	16 200 000	634 883	5112

Tổng tiền hàng: 24 700 000 Tổng tiền chiết khấu: 968 000

Tài khoản chiết khấu: 521 Tổng tiền thuế: 2 470 000

Tổng tiền: 26 202 000

Chức năng (...), Báo cáo nhanh, 1/1, Cơ bản, Mở rộng

- Các chức năng chạy từ CDT:
 - Tên bảng dữ liệu: MT32, DT32 và VATOUT
 - Kiểu nhập liệu của bảng MT32 là bảng chính; bảng DT32 là nhiều bảng phụ thuộc nhập cùng bảng chính, chỉ rõ bảng chính là bảng MT32; bảng VATOUT là bảng đơn nhập trên màn hình riêng (để có thể nhập liệu riêng), chỉ rõ bảng chính là bảng MT32 (để nhập liệu chung với bảng MT32 và DT32). Tương

ứng với các lớp FrmMasterDetail, FrmMasterDetailDt, FormDesigner trong FormFactory, lớp DataMultiDetail trong DataFactory.

- Các chức năng chạy từ plugins:
 - Tự động đưa thông tin vào bảng thuế GTGT đầu ra (VATOUT): các thông tin nhập ở bảng MT32 và DT32 đã đủ để lên bảng thuế GTGT. Tuy nhiên trên giao diện vẫn đưa riêng bảng VATOUT thành một tab để người dùng có thể lựa chọn điều chỉnh phần thuế chi tiết như: các mặt hàng có thuế suất khác nhau, khách hàng để báo cáo thuế khác khách hàng hạch toán công nợ... Tính năng này thực hiện bằng plugin All_MdDt_ICC_Vat.dll. Plugin này chạy trên các chứng từ có kê khai thuế GTGT. Loại plugin là Custom Data, Add Event vào table DT32 và có tên là “VAT”.
 - Xem phiếu nhập để lấy thông tin xuất hàng (chạy từ nút “Chức năng (...))”: dùng plugin All_MdDt_ICF_TraPhieuNhap.dll. Plugin này chạy trên các chứng từ cần xem phiếu nhập hàng ngay trong lúc nhập chứng từ. Loại plugin là Custom Form và có tên là “TraPhieuNhap”.

4) Nhập chi tiết của một đối tượng

- Ví dụ thực tế: Nhập định mức nguyên vật liệu của các sản phẩm

- Các chức năng chạy từ CDT:
 - Tên bảng dữ liệu: DFNVL
 - Kiểu nhập liệu của bảng là “Bảng phụ thuộc nhập riêng trên màn hình riêng”, chỉ rõ bảng chính là DMVT. Trường hợp này bảng DMVT trên giao diện chỉ đóng vai trò điều hướng để nhập chi tiết định mức cho từng sản phẩm trong bảng DMVT. Còn dữ liệu chỉ được nhập trên bảng DFNVL, màn hình nhập liệu cho DFNVL cũng chính là màn hình nhập liệu đơn (giống như màn hình nhập danh mục và số dư). Tương ứng với các lớp FrmDetail, FrmSingleDt và FormDesigner trong FormFactory; lớp DataDetail trong DataFactory.
- Các chức năng chạy bằng plugins:

- Áp giá nguyên liệu bình quân cuối kỳ vào giá định mức: sau khi định mức nguyên vật liệu bằng số lượng, nếu muốn quy về giá trị định mức, người dùng có thể nhập giá định mức hoặc dùng chức năng này để tự động lấy giá bình quân cuối kỳ áp vào giá định mức. Dùng plugin DFNVL_Dt_ICF_GiaDinhMuc.dll. Loại plugin là Custom Form, chỉ chạy trên bảng DFNVL và có tên là “GiaDinhMuc”.

5) Xử lý kết chuyển cuối kỳ

Stt	Mã chứng từ	Diễn giải	Tk Nguồn	Tk đích	Loại kết chuyển	Mã bộ phận	Mã phí	Mã vụ việc
> 0	KCCPBTP	Kết chuyển chi phí nguyên vật liệu	6211	154	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	KCCPNCTT	Kết chuyển chi phí nhân công trực tiếp	622	154	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	KCCPSXC	Kết chuyển chi phí sản xuất chung	627	154	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	KCCPNVL	Kết chuyển chi phí bán thành phẩm	6212	154	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	KCCCKTM	Kết chuyển chiết khấu thương mại	521	5112	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	KCHBTL	Kết chuyển hàng bán trả lại	531	5112	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	KCDTBH	Kết chuyển doanh thu bán hàng	511	911	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	KCDTTC	Kết chuyển doanh thu tài chính	515	911	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	KCGV	Kết chuyển giá vốn	632	911	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	KCCPTC	Kết chuyển chi phí tài chính	635	911	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	KCCPBH	Kết chuyển chi phí bán hàng				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11	KCCPQL	Kết chuyển chi phí quản lý				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12	KCLAI	Kết chuyển lãi				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13	KCLO	Kết chuyển lỗ				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Đây là một chức năng riêng biệt, loại plugin là Custom và chạy hoàn toàn độc lập trên CDT.
- Loại plugin Custom được hệ thống nhận biết khi lần đầu được đưa vào thư mục Plugins của gói phần mềm. Hệ thống sẽ đưa vào dữ liệu CDT ở “Quản lý menu”, sau đó người dùng có thể điều chỉnh chức năng này và phân quyền nó như các chức năng khác của hệ thống.
- Lưu ý khi chạy phần mềm mà không có các dll loại Custom này trong thư mục Plugins của gói phần mềm, hệ thống sẽ tự xóa bỏ chức năng của các dll đó ra khỏi dữ liệu CDT. Đặc biệt lưu ý điều này khi Debug phần mềm rồi Deploy, sẽ bị mất chức năng hoặc chức năng sẽ nằm không đúng vị trí mong muốn trên giao diện chính. Cách tốt nhất là để thư mục Plugins trong thư mục Debug giống như cấu trúc khi Deploy.
- Kiến trúc code của plugin này:
 - Có Interface là IC (mỗi loại plugin tương ứng với một Interface)
 - Dữ liệu về plugin đặt trên InfoCustom
 - Các chức năng của plugin đặt trong hàm Execute
 - Lưu ý là một plugin loại Plugin có thể chứa nhiều menu chức năng

6) Lên báo cáo

- Ví dụ thực tế: Báo cáo tổng hợp nhập xuất tồn
 - a. Chức năng nhập tham số báo cáo

- Các chức năng chạy từ CDT:
 - Nội dung báo cáo nằm ở “Danh sách báo cáo”. Mỗi báo cáo liên quan đến 1 bảng chính và bảng chi tiết. Khi chọn bảng chính và bảng chi tiết, các trường nằm trong bảng này sẽ được tự động đưa vào tham số của báo cáo đó. Alias bảng chính và bảng chi tiết dùng để phát sinh câu điều kiện (mệnh đề where) cho procedure của báo cáo.
 - Sau đó các tham số được thiết lập ở “Tham số báo cáo”
 - Một tham số có thể bắt buộc hoặc không bắt buộc nhập
 - Một tham số có thể có giá trị mặc định.
 - Nếu tham số được chọn từ danh mục, “Lọc cho báo cáo” chính là câu điều kiện khi truy vấn và hiển thị danh mục đó.
 - Một tham số được đánh dấu là “Khoảng số liệu” sẽ làm phát sinh hai tham số “Từ...” và “Đến...”. Ví dụ từ ngày đến ngày, từ mã vật tư đến mã vật tư.
 - Tùy thuộc vào giá trị của “Điều kiện đặc biệt”, sẽ có hai loại tham số
 - Tham số tự phát sinh thành câu điều kiện (mệnh đề where): CDT tự đưa giá trị của tham số vào chuỗi “@@PS”. Trong ví dụ trên Mã kho, Từ mã vật tư, Đến mã vật tư sẽ tự phát sinh vào chuỗi @@PS theo dạng “MaKho = ‘xxx’ and (MaVT between ‘yyy’ and ‘zzz’)”. Khi viết procedure cho báo cáo, chỉ việc chèn @@PS vào sau mệnh đề where là được.
 - Tham số đặc biệt được đánh dấu vào “Điều kiện đặc biệt”: Trong ví dụ trên Từ ngày, Đến ngày, Nhóm vật tư không tự phát sinh vào chuỗi @@PS mà sẽ có dạng @NgàyCT1, @NgàyCT2, @Nhom. Khi viết procedure cho báo cáo, muốn dùng tham số nào ở đâu thì đưa @NgàyCT1, @NgàyCT2, @Nhom vào chỗ đó.
 - Tương ứng với các lớp ReportFilter, FormDesigner trong FormFactory và DataReport trong DataFactory.

b. Chức năng lên báo cáo

Bảng nhóm: kéo thả một cột vào đây để nhóm số liệu

Mã vật tư	Nhóm vật tư	Tên vật tư	Tên đơn vị tính	Tồn đầu	Dư đầu	Số lượng nhập	Giá trị nhập	Số lượng xuất	Giá trị xuất	Tồn cuối	Dư cuối
CCDC		Máy vi tính	Cái	0	0	1	6 000 000	1	6 000 000	0	0
> NL-NH-PE-001		Nguyên liệu nhựa PE 1	kg	300	21 000 000	20	1 490 000	36	2 533 008.76	284	19 956 991.24
NL-TH-UV-001		Nguyên liệu thép UV 1	kg	100	5 000 000	40	2 100 000	17.6	900 390.24	122.4	6 199 609.76
NL-TH-UV-002		Nguyên liệu thép UV 2	kg	200	12 000 000	60	3 660 000	24	1 460 777	236	14 199 223
ZZZ		Tổng cộng		600	38 000 000	121	13 250 000	78.6	10 894 176	642.4	40 355 824
				1 200	76 000 000	242	26 500 000	157.2	21 788 352	1 284.8	80 711 648

F5 - Xem chi tiết F6 - Sửa mẫu Ra excel F7 - Xem Esc - Quay ra

- Các chức năng chạy từ CDT:

o Loại báo cáo:

- Bảng chi tiết (-1): Các báo cáo này là drilldown từ báo cáo khác chứ không phải là 1 báo cáo chạy độc lập.
- Bảng kê (0): Các bảng kê, bảng chi tiết, tổng hợp thông thường.
- Bảng cân đối (1): các loại bảng cân đối (nhập xuất tồn, công nợ, tài khoản...). Loại này thường có một “Bảng cấu trúc cây” kèm theo để phục vụ riêng cho xử lý của nó. Ví dụ bảng cân đối nhập xuất tồn sẽ lấy số liệu về nhập xuất tồn kho của từng vật tư. Xét trường hợp các vật tư thuộc về một nhóm nào đó và cấu trúc của nhóm vật tư là một cấu trúc cây. Yêu cầu là bảng cân đối nhập xuất tồn cần phân chia vật tư vào cấu trúc cây của nhóm vật tư và tính toán được tổng số liệu của từng nhóm vật tư. Loại báo cáo “Bảng cân đối” sẽ tự động xử lý yêu cầu trên dựa vào thông tin cung cấp ở “Bảng cấu trúc cây”.
- Bảng công thức (2): tự động xử lý các chỉ tiêu theo dạng công thức của các báo cáo tài chính, báo cáo quản trị sử dụng chỉ tiêu và công thức.
- Bảng NKCT (3): tên đúng hơn của loại này có thể là dạng Cross-Tab. Tức là báo cáo này sẽ tự động dựng các dòng số liệu thành các cột dựa vào thông tin cung cấp ở “Trường làm cột”.

- o Chức năng drilldown: được xử lý dựa vào thông tin của “Trường liên kết” và “Báo cáo cha”. Ví dụ “Bảng cân đối nhập xuất tồn” khi drilldown sẽ xem được “Sổ chi tiết vật tư”. Lúc đó “Trường liên kết” của “Bảng cân đối nhập xuất tồn” là “MaVT” còn “Báo cáo cha” của “Sổ chi tiết vật tư” là “Bảng cân đối nhập xuất tồn”.

- o Chức năng in đồ thị: được xử lý dựa vào thông tin của 3 trường đồ thị.

c. Định dạng số trên báo cáo

- Khi muốn định dạng kiểu số trên báo cáo, không được sử dụng chuỗi cố định như [### ###.###] mà phải sử dụng hàm dùng chung trong hệ thống.

Ví dụ: các control binding dữ liệu thì sử dụng script để định dạng số:

```
Label1.DataBindings[0].FormatString =
CDTControl.FormatString.GetReportFormatBinding("Tien");
```

- Hàm [GetReportFormatBinding] sẽ trả về chuỗi định dạng {0:###}
- Hàm [GetReportFormat] sẽ trả về chuỗi định dạng ###

7.6 Hướng dẫn tạo form nhập liệu, report và plugin mới

1) Tạo form nhập liệu mới

- **Bước 1:** Vào CDT → Quản lý dữ liệu → Danh sách bảng dữ liệu → F2-Thêm
Để nắm rõ chi tiết các thông tin chức năng trong danh sách bảng dữ liệu, tham khảo phần 7.5 – các Case-Study từ 1 đến 4.

Lưu ý là bước này cũng đồng thời tạo table dưới database.

- **Bước 2:** Vào CDT → Quản lý giao diện → Quản lý menu → F2-Thêm

Chọn bảng số liệu vừa mới tạo

Điều kiện phụ sẽ được đưa vào sau mệnh đề where khi truy vấn bảng số liệu

Chọn loại menu phù hợp để hiển thị đúng nơi mong muốn trên giao diện

- **Bước 3:** Mở phần mềm, chạy chức năng mới và kiểm tra.
- **Bước 4:** Viết thêm plugin để phục vụ các chức năng khác trên form mới nếu cần.
Để nắm rõ chi tiết cách viết plugin, tham khảo phần 7.5 – các Case-Study từ 1 đến 4.
- **Bước 5:** Tạo script update Database (tham khảo [Quy định update server script](#)).

2) Tạo report mới

- **Bước 1:** Vào CDT → Quản lý báo cáo → Danh sách báo cáo → F2-Thêm

Để nắm rõ chi tiết các thông tin chức năng trong danh sách báo cáo, tham khảo phần 7.5 – Case-Study 6

- **Bước 2:** Vào CDT → Quản lý báo cáo → Tham số báo cáo

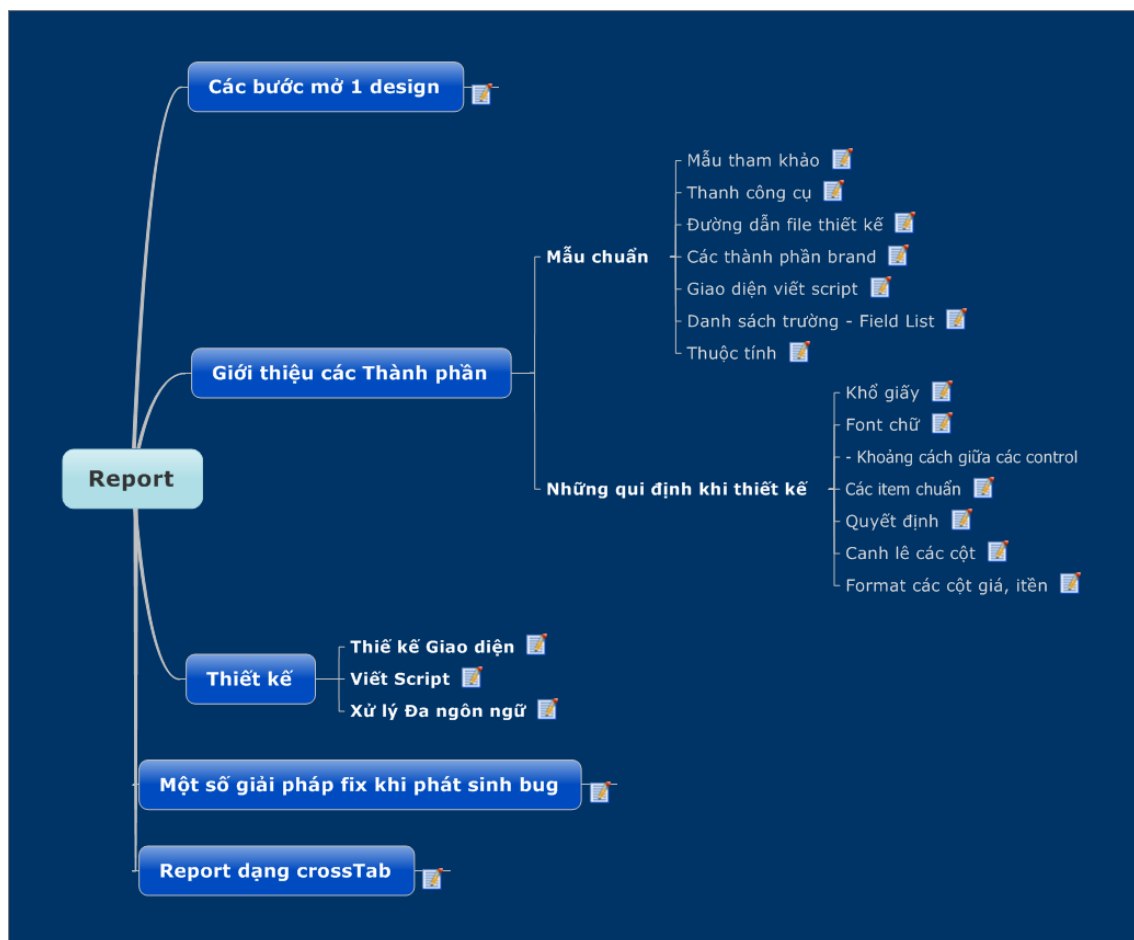
Thiết lập các tham số sẽ hiển thị để người dùng lựa chọn và phần mềm sẽ lọc dữ liệu trước khi lên báo cáo (Phần IV – Case-study 6a).

- **Bước 3:** Vào CDT → Quản lý giao diện → Quản lý menu → F2-Thêm
Chọn báo cáo vừa mới tạo
Chọn loại menu phù hợp để hiển thị đúng nơi mong muốn trên giao diện
- **Bước 4:** Mở phần mềm, chạy chức năng mới và kiểm tra.
- **Bước 5:** Tạo script update Database (tham khảo [Quy định update server script](#)).

3) Tạo plugin hoàn toàn mới

- Bước 1: Tạo mới một project trên VS.Net
- Bước 2: References với Plugins.dll
- Bước 3: Tạo mới 1 class kế thừa từ interface IC thuộc Plugins, tên namespace và tên class này phải giống nhau
- Bước 4: Đặt dữ liệu về plugin vào InfoCustom: PluginID, MenuName, MenuParent
- Bước 5: Các chức năng của plugin đặt trong hàm Execute
- Bước 6: Đặt lại tên assembly theo quy định sau: Tên class ở bước 3 + “_IC”
- Bước 7: Đưa assembly vào thư mục Plugins của gói phần mềm và chạy gói phần mềm để nạp plugin vào menu hệ thống
- Bước 8: Điều chỉnh lại thông tin menu chức năng này tại “Quản lý menu” của CDT

7.7 Hướng dẫn thiết kế một báo cáo



1) Các bước mở 1 design

- Bước 1 : Chọn menu báo cáo
- Bước 2 : Chọn điều kiện báo cáo
- Bước 3 : Kết quả báo cáo
- Bước 4 : **Sửa mẫu**
- Bước 5 : Màn hình thiết kế báo cáo.

Ví dụ : mở màn hình “Sổ Quỹ tiền mặt” : Menu Quản Lý tiền → Sổ Quỹ Tiền Mặt
→ Chọn điều kiện → F6-Sửa mẫu → File báo cáo.

2) Giới thiệu các thành phần

a. Mẫu chuẩn

- Mẫu tham khảo

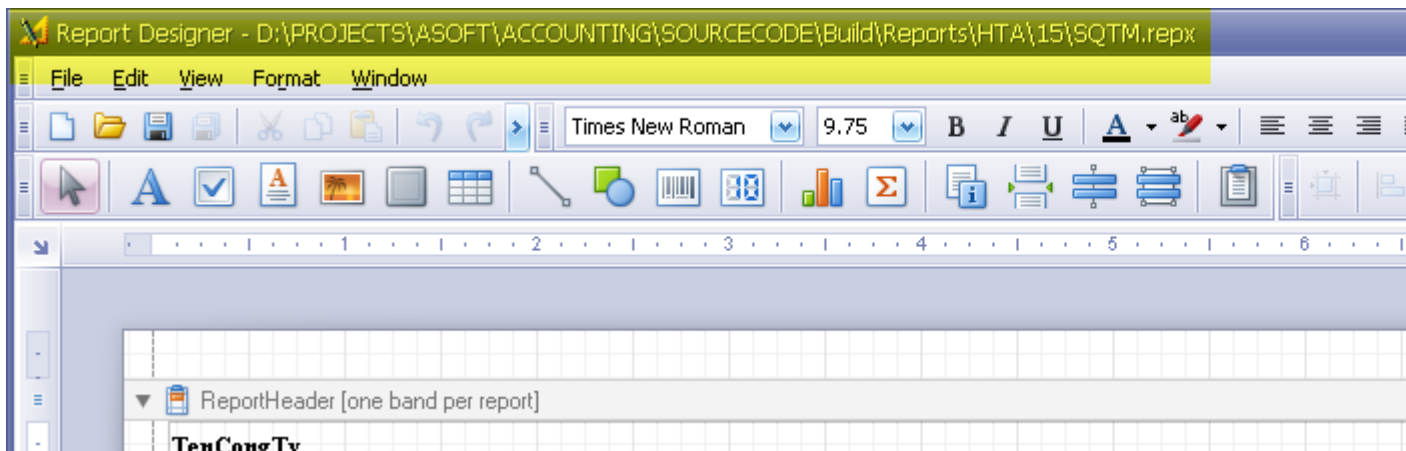
Sử dụng mẫu [Template.repx] để thiết kế (lưu file báo cáo giống tên đã cấu hình trong CDT)

- [30_SOURCECODE\304_Setup\Program\Reports\HTA\15\]
- [30_SOURCECODE\304_Setup\Program\Reports\HTA\48\]

- Thanh công cụ

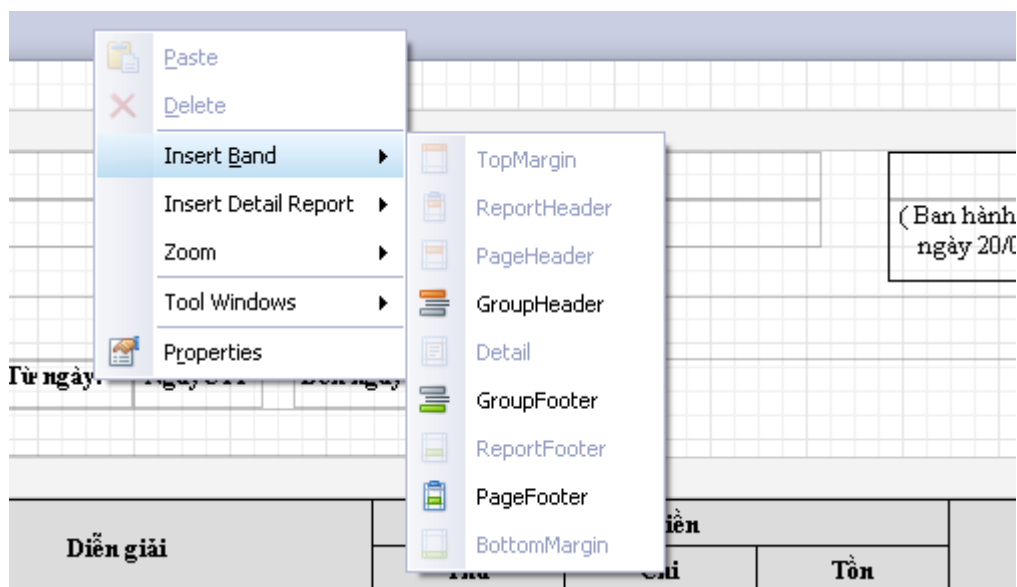


- Đường dẫn file thiết kế



- Các thành phần brand

TopMargin, ReportHeader, PageHeader, GroupHeader, Detail, GroupFooter, ReportFooter, PageFooter, BottomMargin



- Giao diện viết script

Tài khoản:

PageHeader [one band per page]				
Ngày, tháng ghi sổ	Ngày, tháng chứng từ	Số hiệu chứng từ		Diễn giải
		Thu	Chi	
[ReportData]	[ReportData]	[ReportData.S	[SoCT]	[ReportData.DIENGIAI]
ReportFooter [one band per report]				
				Tổng c
- Số này có	ToT	trang, đánh số từ trang	1	đến trang ToT
- Ngày mở sổ	10/09/2012			
Người ghi sổ		Kế toán trưởng		
(Ký, họ tên)		(Ký, họ tên)		

dataView1

Designer Preview HTML View Scripts xtraReport1 { PaperKind: A4 }

Trong version devexpress 11.2 thì script được đưa vào chung một màn hình

```

1
2 private void colPT_BeforePrint(object sender, System.Drawing.Printing.PrintEventArgs e)
3 {
4     double psNo = 0;
5     string psNoString = colPSNo.Text.Replace(" ", "");
6     if (!string.IsNullOrEmpty(psNoString))
7     {
8         psNo = double.Parse(psNoString);
9     }
10
11     if (psNo == 0)
12     {
13         colPT.Text = "";
14     }
15 }
16
17 private void colPC_BeforePrint(object sender, System.Drawing.Printing.PrintEventArgs e)
18 {
19     double psCo = 0;
20     string psCoString = colPSCo.Text.Replace(" ", "");
21     if (!string.IsNullOrEmpty(psCoString))
22     {
23         psCo = double.Parse(psCoString);
24     }
25
26     if (psCo == 0)
27     {
28         colPC.Text = "";
29     }
30 }
31
32 using CDTControl;
33 private void ReportHeader_BeforePrint(object sender, System.Drawing.Printing.PrintEventArgs e)
34 {
35     colPSNo.DataBindings[0].FormatString = FormatString.GetReportFormatBinding("PSNo");
36     colPSCo.DataBindings[0].FormatString = FormatString.GetReportFormatBinding("PSCo");
37     colPsTon.DataBindings[0].FormatString = FormatString.GetReportFormatBinding("PsTon");
38     totalSoDu.DataBindings[0].FormatString = FormatString.GetReportFormatBinding("TotalSoDu");
39
40     totalPsNo.Summary.FormatString = FormatString.GetReportFormatBinding("TotalPsNo");
41     totalPsCo.Summary.FormatString = FormatString.GetReportFormatBinding("TotalPsCo");
42 }
43
44 private void ToTalPage2_PrintOnPage(object sender, DevExpress.XtraPrint.OnPageEventArgs e)
45 {
46     ToTalPage2.Text = e.PageCount.ToString();
47 }

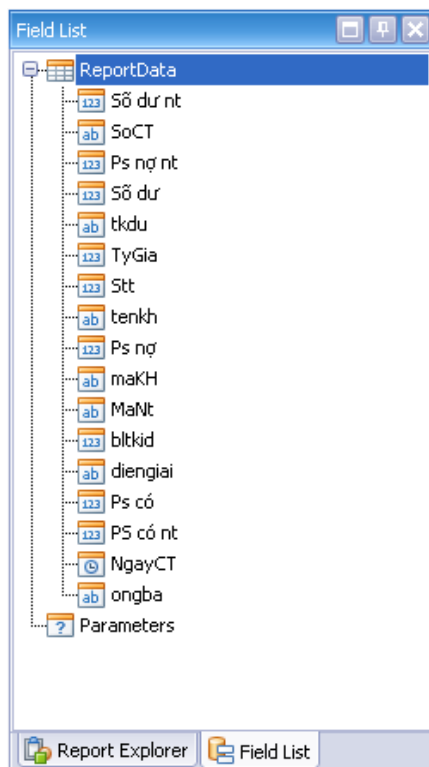
```

- Danh sách trường - Field List

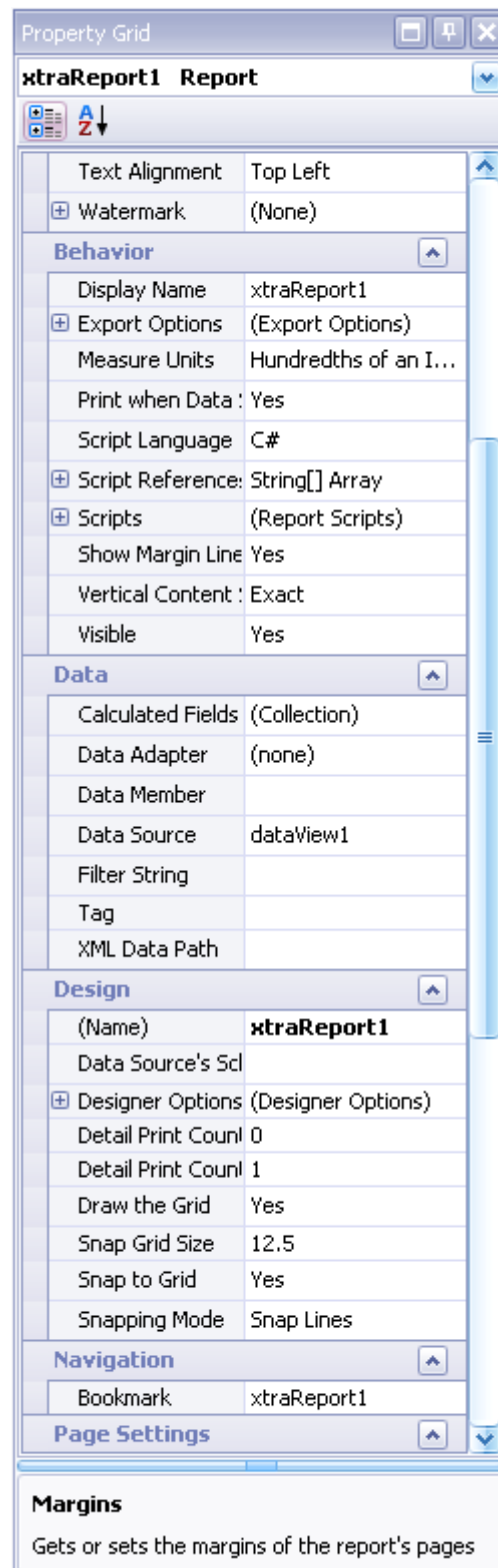
Danh sách các trường được tạo ra từ CDT

- Thuộc tính

Chọn một item trên màn hình thiết kế --> Properties



Hiện tại các parameters chưa được xài tại thời điểm cập nhật tài liệu (10/09/2012).

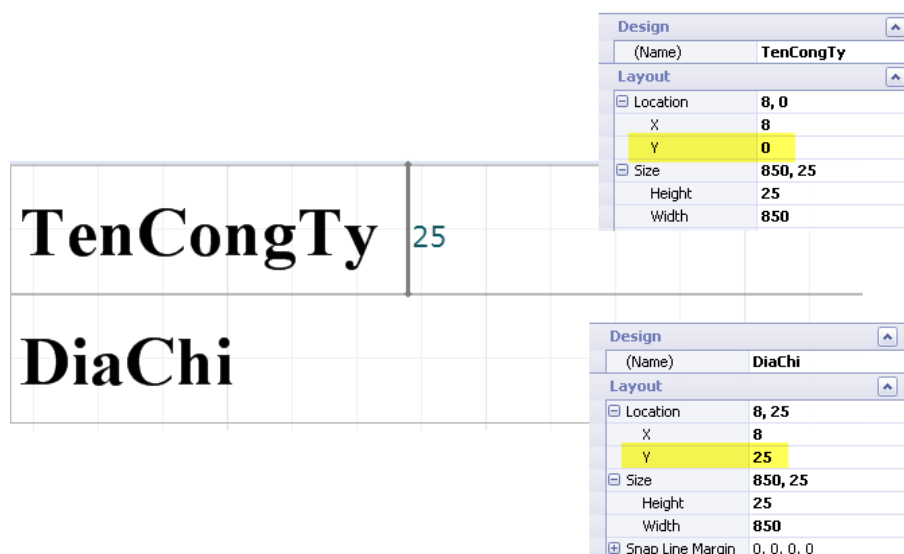


b. Những quy định khi thiết kế

- Khổ giấy

- Chuẩn chọn khổ giấy A4.
- Thay đổi khổ giấy tùy thuộc vào thiết kế :

- Chính thuộc tính Paper Kind : [A3, A4, Letter,...).
- Chọn báo cáo --> chuột phải -> Properties -> Page Settings --> Paper Kind.
- Đối với các báo cáo dạng crosstab thì tham khảo phần [\[Report dạng CrossTab\]](#).
- Margin : [Bottom, Left, Right, Top] -> [25,15,15,15]
- **Font chữ**
 - Sử dụng font Times New Roman - 9.75pt
 - Sử dụng font Times New Roman - 20pt cho title của báo cáo
 - **Bold** các header của báo cáo, các trường tổng cộng
 - **Màu** Background của các header = Gainsboro
 - **Width** của border = 1
- **Khoảng cách giữa các control**
 - **Padding** của control : 2,2,2,2
 - **Chiều cao : 25**
 - **Text Alignment** : Middle [LEFT | RIGHT | CENTER]
 - Khoảng cách giữa các control = 0



- **Các control chuẩn lấy dữ liệu từ màn hình thiết định**
- Tùy vào thiết kế thì báo cáo có một số qui định chuẩn sau đây
 - Các label cố định như Tên công ty, địa chỉ, mẫu báo cáo, **Quyết định**, Ký tên, Số trang -> **đặt tên của các label (Name) bằng với các biến constant**
 - Sử dụng các constant trong bảng : sysConfig trong CDT : `[select sysConfigID, DienGiai, DienGiai2, _Key, _Value, DbName from sysConfig where StartConfig = 1 and sysSiteID = ??? and DbName = ???)]`

- **Các quyết định (48, 15)**

- Hệ thống AsoftAccounting đang hỗ trợ 2 quyết định là : 15 và 48, vì vậy khi thiết kế phải theo các bước sau:
 - Bước 1 : Chỉnh thông số quyết định trên báo cáo (thay đổi)
 - Bước 2 : Lưu thành vào các thư mục 15, 48 tương ứng

- **Canh lề các cột**

Kiểu	Canh lề
Chuỗi	Trái
Số	Phải
Ngày	Giữa

- **Format các cột giá, tiền**

- Sử dụng code trong báo cáo để định dạng các cột giá, tiền theo cấu trúc sau:

```
private void ReportHeader_BeforePrint(object sender,
System.Drawing.Printing.PrintEventArgs e) {
colPSNo.DataBindings[0].FormatString =
FormatString.GetReportFormatBinding("Tien"); // Dòng dữ liệu
totalGia.Summary.FormatString =
FormatString.GetReportFormatBinding("DonGia"); // Summary
//.....
}
```

⇒ Tùy vào thuộc tính của control để format cho đúng

3) Thiết kế

a. Thiết kế giao diện

- Bước 1 : Kéo control và sắp xếp theo thiết kế
- Bước 2 : Kéo các trường [\[Danh sách trường - Field List\]](#) tương ứng vào control
- Bước 3 : Viết script định dạng các trường loại tiền, đơn giá. Tham khảo [Format các cột giá, tiền]

b. Viết Script

Report cho phép sử dụng các hàm trong framework, các thư viện của CDT nên người thiết kế chỉ cần using thư viện :

Ví dụ :

- Đa ngôn ngữ

```
using CDTLib;
//Sự kiện
```

```
private void xtraReport1_BeforePrint(object sender,
System.Drawing.Printing.PrintEventArgs e) {
if (Config.GetValue("Language").ToString() == "1")
{
e.DisplayText = "Communist accounts" + lblTK.Text;
}
}
```

- Số trang của báo cáo

```
private void ToTalPage2_PrintOnPage(object sender,
DevExpress.XtraReports.UI.PrintOnPageEventArgs e) {
ToTalPage2.Text = e.PageCount.ToString();
}

private void ToTalPage_PrintOnPage(object sender,
DevExpress.XtraReports.UI.PrintOnPageEventArgs e) {
ToTalPage.Text = e.PageCount.ToString();
}
```

c. Xử lý Đa ngôn ngữ

Đa ngôn ngữ cho báo cáo được cấu hình trong các phần sau:

- Bảng **Dictionary** ➔ cập nhật trong file script
\20_DATABASE\203_UpdateDB\01_CDT\9999_NewScripts\99_DATA\00\000_D
ictionary.sql
- Script của report dành cho các trường đặt biệt như text của crosstab, text của dòng
tổng cộng của dữ liệu,...

```
if (Config.GetValue("Language").ToString() == "1")
{
e.DisplayText = "Communist accounts" + lblTK.Text;
}
```

4) Một số giải pháp fix khi phát sinh bug

a. Đa ngôn ngữ của màn hình review dữ liệu

Quản lý tiền					
Số quỹ tiền mặt					
Số quỹ tiền mặt					
Bảng nhóm: kéo thả một cột vào đây để nhóm số liệu					
Số chứng từ	Ngày chứng từ	Ông bà	Đối tượng	Diễn giải	
>				Đầu kỳ	
PT12/001	01/12/2010	Bùi Tuấn Kiệt	BTK	Phiếu thu	
PT12/002	01/12/2010	Hoàng Đăng Khoa			
PT12/003	01/12/2010	Bs. Nguyễn			
PT12/004	01/12/2010	Hoàng Thị C			
PT12/005	01/12/2010	Huỳnh Văn			
PT12/006	01/12/2010	CTY TNHH A			
PT12/007	01/12/2010	CTY TNHH THIẾT BỊ Y TẾ	KH003	Phiếu thu	

Chuyển qua cột tiếng anh khi câu sql có chứa cột tiếng việt
Vi dụ : Select ongba as 'Ông bà' from bltk

➔ Thêm một trường trong bảng [sysField] với tham số (tham khảo script 69_ThemReportNKCT.sql)

- **TableName** : wReportRvCtrl
- **LabelName** : Tên cột tiếng việt
- **LabelName2** : Tên cột tiếng anh

b. Format các cột loại tiền, giá trong màn hình review dữ liệu:

Quản lý tiền										
Số quỹ tiền mặt										
Số quỹ tiền mặt										
Bảng nhóm: kéo thả một cột vào đây để nhóm số liệu										
Số chứng từ	Ngày chứng từ	Tỷ giá	Ps nợ	Ps có	Số dư	Ps nợ nt	P5 có nt	Số dư nt	Tên đối tượng	
>					3,326,026,663.8756			0		
PT12/001	01/12/2010	1	30,000	0	3,326,056,663.876	0	0	0	0 Bùi Tuấn Kiệt	
PT12/002	01/12/2010	1	23,000	0	3,326,079,663.876	0	0	0	0 Hoàng Đăng Khoa	
PT12/003	01/12/2010	1	300,000	0	3,326,379,663.876	0				
PT12/004	01/12/2010	1	150,000	0	3,326,529,663.876	0				
PT12/005	01/12/2010	1	360,000	0	3,326,889,663.876	0				
PT12/006	01/12/2010	1	260,000	0	3,327,149,663.876	0	0	0	0 CTY TNHH ABC	
PT12/007	01/12/2010	1	260,000	0	3,327,409,663.876	0	0	0	0 CTY TNHH THIẾT BỊ Y TẾ	
PT12/008	01/12/2010	1	260,000	0	3,327,669,663.876	0	0	0	0 CỬA HÀNG MỤC IN	

Format các trường này theo thiết định số của chương trình

➔ thêm một dòng dữ liệu vào bảng sysFormatString với các tham số:

- **_Key** : một trong các loại [DonGia, DonGiaNT, HeSo, SoLuong, Tien, TienNT, TConLai, SoLuongQD, DonGiaQD, DonGiaQDNT]
- **FieldName** : Tên cột trong sql báo cáo.

c. Đường gạch ngang bị đứt quãng

CÔNG TY CỔ PHẦN GIẢI PHÁP PHẦN MỀM N.H.T

35-37 Đường 65, P.Tân Phong, Q.7, TP.Hồ Chí Minh

MST : 0302986741

Điện thoại :

Số tài khoản :

Ngân hàng :

HÓA ĐƠN GIÁ TRỊ GIA TĂNG

Mẫu số: 01/GTKT - 3/L

Ký hiệu: HDDVB12/

6

Số: HDDVB12/0016

Ngày 14 Tháng 12 Năm 2010

Họ và tên người mua hàng :

Đơn vị: CTY TNHH SX-TM THÈP NGỌC PHÁT

Địa chỉ: 111 Tô Hiệu, P. Hiệp Tân, Q. Tân Phú

MST: 0201030405

Thanh toán: CK Số tài khoản:

. Ngân hàng:

STT	Tên hàng hóa dịch vụ	Đơn vị tính	Số lượng	Đơn giá	Thành tiền
1	hóa đơn dịch vụ		1	300,000	300,000

→ chỉnh thuộc tính Anchor Verically của Line = Both : tham khảo INDDVUTUIN.repx

d. Hiển thị số tiền bằng chữ

Nếu báo cáo có các trường yêu cầu đọc số thành chuỗi thì sử dụng hàm ReadVoucherMoney, ReadReportMoneyVN

A	B	C	D
rTa	[default.TenVT]	[default	[defau
and per report]			
Cộng			

- Tổng số tiền (viết bằng chữ)

- Số chứng từ gốc kèm theo

Người lập phiếu

(Ký, họ tên)

Người giao hàng

(Ký, họ tên)

- **Hàm ReadVoucherMoney()** : Đọc số tiền tùy thuộc vào tỷ giá
 - Tham số :
 - pMoney : Số tiền cần đọc
 - pDVTienTeIn : Mã tiền tệ (VND, USD, EUR,...)
 - pTenNT : Tên tiền tệ tiếng việt (VNĐ)
 - pTenNT2 : Tên tiền tệ tiếng Anh (VND)
- **Hàm ReadReportMoneyVN()** : Đọc số tiền Việt Nam
 - Tham số :
 - pMoney : Số tiền cần đọc
- **Cách sử dụng hàm ReadVoucherMoney()**
 - **Bước 1** : Tạo các control cố định để lấy tham số (đặt tên như phần in đậm)
 - **DVTienTeIn** : dữ liệu sẽ được lấy từ config. Tham khảo [...]
 - **TenNT** : Một field từ câu SQL báo cáo
 - **TenNT2** : Một field từ câu SQL báo cáo
 - **Bước 2** : Viết Script để fill dữ liệu vào control

```
using CDTLib;
using CDTControl;
using System.Data;
//Sự kiện của control cần đọc số
private void TongTien_BeforePrint(object sender,
System.Drawing.Printing.PrintEventArgs e) {
    TongTienC.Text =
    ReadNumberDynamic.ReadVoucherMoney(TongTien.Text, DVTienTeIn.Text,
    TenNT.Text, TenNT2.Text);
}
```

5) Report dạng crosstab

Tham khảo báo cáo “Nhật ký chứng từ 02”, “Nhật ký Sổ cái”.

- **Bước 1** : Cấu trúc sql phải có dạng sau

Cột cố định (có thể nhiều cột)	Cột chính	Cột giá trị
Nhân viên A	TK001	100.000
Nhân viên B	TK002	50.000
Nhân viên A	TK001	30.000

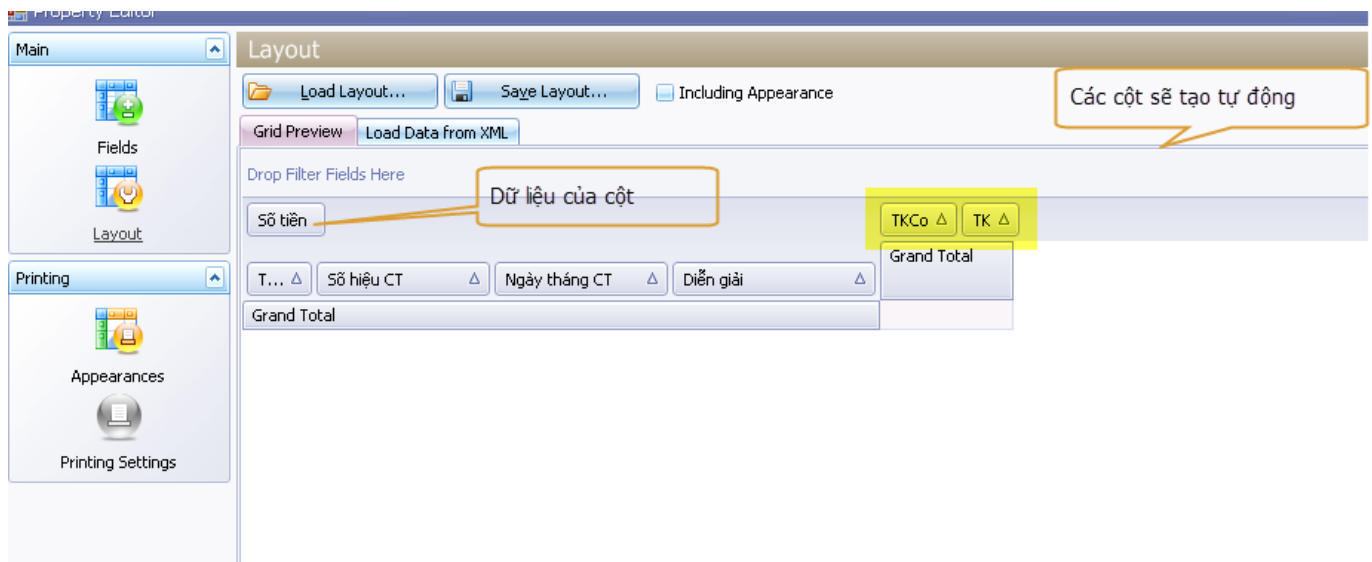
- **Bước 2** : Thiết kế các header sử dụng table và lable
- **Bước 3** : Thiết kế crosstab :



Chọn crossTab --> Run Designer.. --> cấu hình cột cho report

- **Fields:** Danh sách các filed được gen từ câu sql trong CDT
- **Layout :** Kéo/ thả các cột theo thiết kế, cố định column sẽ tạo ra cột động và dữ liệu cho cột này

Ví dụ báo cáo "Nhật ký chứng từ số 02" - NKCT02.rep



- **Bước 4 :** Chỉnh các thuộc tính của report ở Options (tham khảo)

Print Options	
Print Column Headers	Hiển thị cột báo cáo của crosstab
Print Data Headers	Hiển thị header của cột data (Số tiền)
Print Filter Header	Hiển thị cột filter
Print Horizontal Lines	Hiển thị border của lưới crosstab
Print Row Headers	Hiển thị header của dòng dữ liệu (Tkco, TK)
View Options	
Column Totals Location	Hiển thị vị trí cột tổng cộng (trước hay sau)
Group Fields in the Customization Window	Group các cột giống nhau
Show Custom Total for Single Values	Hiển thị cột tổng cộng khi chỉ có 1 cột crosstab
Show Row Totals	Hiển thị dòng tổng cộng
Show Totals for Single Values	Hiển thị dòng tổng cộng khi chỉ có 1 dòng

- **Bước 5 :** Script để tự động chỉnh trang in (do report crosstab có số cột phụ thuộc vào dữ liệu nên khổ giấy thường không cố định)

```

private void xtraReport1_BeforePrint(object sender,
System.Drawing.Printing.PrintEventArgs e) {
    int marginLeft = xtraReport1.Margins.Left;
    int marginRight = xtraReport1.Margins.Right;
    int A4Width = 550;

    int countColumn = GetCountColumnHeadersArea(); //Lấy tổng số cột sẽ
phát sinh
    if (countColumn > 5) //Tùy vào màn hình mà xác định số cột tối đa
    {
        xtraReport1.Margins = new System.Drawing.Printing.Margins( 0,0,0,0);
        xtraReport1.PaperKind = System.Drawing.Printing.PaperKind.Custom;
        xtraReport1.PageWidth=A4Width + (countColumn*94) + marginLeft +
marginRight;
        xtraReport1.Margins
= new System.Drawing.Printing.Margins( 33,marginLeft ,55 ,marginRight);
    }else{
        xtraReport1.PaperKind = System.Drawing.Printing.PaperKind.A4; // Thay
đổi khổ trang
    }
}

```

- **Bước 6** : Script để format các cột trong crosstab

```

private void pivotGrid1_PrintFieldValue(object sender,
DevExpress.XtraPivotGrid.CustomExportFieldValueEventArgs e) {
    if(e.Field != null){
        //Tên của field lúc thiết kế
        if (e.Field.Name == "fieldShiuCT1"
||e.Field.Name == "fieldDingii1")
        {
            if((e.Brick as DevExpress.XtraPrinting.TextBrick) != null)
                ((DevExpress.XtraPrinting.TextBrick)e.Brick).HorzAlignment =
DevExpress.Utils.HorzAlignment.Near;
        }
        if (e.Field.Name == "fieldStinphsinh")
        {
            if((e.Brick as DevExpress.XtraPrinting.TextBrick) != null)
                ((DevExpress.XtraPrinting.TextBrick)e.Brick).HorzAlignment =
DevExpress.Utils.HorzAlignment.Far;
        }
    }
}

```

```

if (e.Field.Name == "fieldTK1" || e.Field.Name == "fieldTKCo")
{
    if (e.IsColumn)
    {
        if((e.Brick as DevExpress.XtraPrinting.TextBrick) != null)
            ((DevExpress.XtraPrinting.TextBrick)e.Brick).Font
= new Font(e.Appearance.Font, FontStyle.Bold);
    }
}
}
}

```

- **Bước 7** : Script thay đổi text của dòng tổng cộng

```

private void pivotGrid1_FieldValueDisplayText(object sender,
DevExpress.XtraPivotGrid.PivotFieldDisplayTextEventArgs e) {
    if (e.IsColumn && e.ValueType ==
DevExpress.XtraPivotGrid.PivotGridValueType.GrandTotal)
    {
        e.DisplayText = "Cộng có TK" + lblTK.Text;
        if (Config.GetValue("Language").ToString() == "1")
        {
            e.DisplayText = "Communist accounts" + lblTK.Text;
        }
    }else if (e.ValueType ==
DevExpress.XtraPivotGrid.PivotGridValueType.GrandTotal)
    {
        e.DisplayText = "Tổng cộng";
        if (Config.GetValue("Language").ToString() == "1")
        {
            e.DisplayText = "Total";
        }
    }
}
}

```

- Chú ý các thuộc tính total của cột
 - **Name** : id của cột để xử lý trong script
 - **Totals Visibility** : Ẩn/ hiện cột tổng cộng phát sinh
 - **Min Width** : kích thước nhỏ nhất của cột

- **Width** : kích thước của cột

THEO DÕI SỬA ĐỔI TÀI LIỆU

TT	VỊ TRÍ SỬA ĐỔI	NỘI DUNG SỬA	NGÀY BAN HÀNH (Tài liệu cũ)	GHI CHÚ