C#报表工具的设计与实现



学	院 <u>计</u> 算	算机科学与技术
专	业 <u>计算</u>	算机科学与技术
年 绪	级	2007 级
姓	名	孙富强
指导教	师	张坤龙
2011年	6月	15 日

摘要

报表工具是一种能够对数据进行分析处理,并在报表中动态展示数据的软件。如何快速有效地生成多种多样的报表一直都是现代企业信息系统中的一个重 要课题。

论文在分析了报表工具的需求之后,设计了报表工具的数据库模块、绘图模块、排版模块、读写 XML 模块、打印模块和界面模块。并对服务器、数据库连接,数据绑定、利用鼠标键盘动态绘制图形、界面对齐、大小调整、读写 XML、打印等功能进行了实现。

很多报表在实际应用中只需要在已有报表的基础上稍加修改,而不用重新设计。为了实现这种灵活性,论文的报表工具采用 XML 格式来存储报表格式。

关键词:报表工具;报表设计;报表打印;报表模版;C#;

ABSTRACT

Reporting tool is a software to analyze and process the data, and dynamic display of data in the report. How quickly and efficiently generate a variety of reports has been the modern enterprise information system, particularly in the management information system is an important issue.

After pepers analysis the demand of the reporting tool, it designs a database reporting tool modules, graphics modules, layout module, the read and write XML modules, print modules and interface modules. And the server, database connectivity, data binding, using mouse and keyboard dynamic rendering graphics, interfaces align, resize, read and write XML, printing and other functions were realized.

Many reports in practical applications only need to have been slightly modified on the basis of the report, rather than re-design. To achieve this flexibility, the paper's reporting tools in XML format to store the report format.

Key words: reporting tools; report design; report printing; report templates; C#;

笄	夏一章	绪论1
	1.1	课题背景1
	1.2	研究意义4
	1.3	研究内容4
	1.4	论文结构5
笄	三章	C#报表工具需求分析6
	2.1	用例分析6
	2.2	功能需求6
	2.3	执行流程分析7
	2.4	软硬件需求7
笌	三章	C#报表工具设计9
	3.1	数据模块设计9
	3.2	绘图模块设计9
	3.3	排版模块设计10
	3.4	读写 XML 模块设计11
	3.5	打印模块设计12
	3.6	用户界面设计13
笌	医四章	C#报表工具实现17
	41	数据模块实现17
	1.1	

4.3	排版模块实现	23
4.4	读写 XML 模块实现	24
4.5	打印模块实现	26
4.6	制作安装文件	28
第五章	章 总结和展望	29
参考文	〔献	
外文资	 彩料	
中文译	文	
致谢		

第一章 绪论

1.1 课题背景

1.1.1 相关概念

报表就是用表格、图表等格式来动态显示数据¹¹。人们利用计算机的数据处 理和界面设计能力来生成、展示报表。报表的主要特点是数据动态化,格式多样 化,并且实现报表数据和报表格式的完全分离,用户可以只修改数据,或者只修 改格式。

1.1.1.1 报表系统

报表系统^[2]由四个部分组成:报表设计器、报表引擎、报表服务器、报表 客户端。各部分组成见图 **1-1** 所示。



图 1-1 报表系统

一个完整的报表系统应该能够在报表设计器中设计报表,包括报表的绘制, 排版等;应该能过在报表引擎中进行运算,数据导入与处理,添加图片,改变格 式,对报表进行分页;应该能过在报表服务器中进行缓存服务,日志服务,调度 服务,监控服务,调度服务等;能够在客户端通过和服务器端交互得到满意的报 表。

1.1.1.2 报表工具

报表工具^[3]:是用来帮助用户展现自己输入的数据,更多时候是将数据库中的数据,以客户想要的方式展现出来。报表工具可以认为是报表系统的一部分,比如在大型企业或政府机关中,上级要监管、考核下级,通常的做法是要求下级定时保送一些报表,然后对这些报表进行分析统计。一个大型企业或政府机关需要一套报表系统,而每一级从报表系统中制作报表,相当于每一级有一个自己的报表工具。所以说报表工具不适合大型企业或政府机关。他只能应对不太复杂多变,对数据处理能力不是很强的情况。

报表工具不同于普通的 Excel 表格,首先报表工具是和数据库相连的,可以 动态刷新数据,比如当在数据库中加一项数据,Excel 中的数据不会有变化,但 是报表工具中的数据会自动更新;其次,报表工具可以通过不同的数据权限得到 不同的报表,而不用像 Excel 做几张不同的表格;最后所有用户可以通过报表直 接看到数据更新,而不用每个人发一份 Excel。

1.1.2 报表工具的发展及现状

1.1.2.1 报表工具的发展

由于中国在信息产业领域起步晚,所以中国在计算机相关技术上面的发展主要是借鉴和学习国外的先进技术,然后消化、吸收、改进。作为国内报表的启蒙者——水晶报表^[4]从 1988 年开始开发以来,93 年随着微软的 VB 一起发行,随着 VB 的流行,它几乎在一夜之间成为报表软件业的标准。但是水晶报表无法很好的满足中国式报表的要求,比如多数据源关联交叉表是非常常见的一种报表样式。水晶报表解决方案是,通过写复杂视图或子报表来解决问题。但是,如果遇到复杂需求,写一个复杂视图就是一件困难的事情,同时还要面对视图性能的挑战。使用子报表解决,可能在格式上有麻烦,也有可能在计算上有困难(主子报表之间往往只传递参数,无法直接引用数据进行公式计算)。这促进了国内报表工具如雨后春笋般得出现在报表市场中,成为现今不可或缺的中坚力量。

中国报表工具的主要代表有 FineReport 和 RAQ。FineReport 是由几个硅谷 回国的技术人员创办的,是一款集数据展示(报表)和数据录入(表单)功能于一 身,用于快速构建企业信息系统的报表工具。RAQ 是本土从做校园管理软件转型 的报表工具。它们的主要特点如下。

- ▶ 报表设计器简单高效。该报表采用类 Excel 界面设计,简洁易懂,轻松 上手。通过拖拽控件就可以绘制一个表格。
- 报表的展示可以动态变化。报表支持多数据源,来自不同结构数据源的数据,仍可以被用户使用,建立需要的报表模版。
- ▶ 报表打印多样话。支持 pdf、applet、flash 等打印格式。

- ▶ 兼容 Excel 文件直接导入,可拖拽数据集
- ▶ 具有丰富的图表
- 可以设置报表定时任务,可以设定事件段,自定义的频率生成报表。并 对生成的文件入库,进行检索管理
- ▶ 统一的通用报表应用平台,彻底解决业务数据不能并行使用的问题
- 支持自定义规则数据分组。表报不单单可以像数据库中默认的把数据相同的分为一组,还可以自定义分组,这样可以更灵活,准确的进行数据分析处理。
- 可以对数据进行双向操作。传统报表只能将数据导入到报表,不能回写 ,相当于一种数据展示的作用,而现在可以把报表中的数据回写到数据 库中。
- ➤ 二次开发能力强。报表工具设计采用 java 语言,为二次开发预留了很多接口。

1.1.2.2 报表工具的现状

报表软件经过几十年的发展,特别是最近几年,企业和政府都开始认识到了 报表工具的重要作用,报表工具得到了蓬勃的发展,现在的报表工具虽然不能满 足所有要求,但是每一样要求几乎都有某款报表工具把它做到完美。下面介绍一 下主流报表做的好的方面:

- ▶ 拥有 Dreamweaver 一般的操作界面。只有有一个好的界面才能有一个好的人机交互,才能得到客户的承认,并乐于使用这款报表工具。
- ▶ 直接导入 Excel, html。不单单可以从数据库导入数据,还可以从现有报表中导入数据。
- 所见即所得。又是一项以用户为本的举措,用户不但可以直接把看到的 控件拖拽到报表设计器上,还可以快速直接的把数据源设置给特定的控件。
- ▶ 可以实现多个数据库的连接,把多个数据源导入报表中,大大提高了表报的数据处理分析能力。
- ▶ 跨平台的结构。报表工具为外界提供了 GUI 接口和可嵌入式的接口,以 实现跨平台服务。
- ▶ 实现系统化。报表系统可以看成多个报表工具的集合,分配给大型企业中的不同级别的部门,避免了每一个部门生成一种报表格式,数据每个人都能看的问题。
- ▶ 支持多种打印格式。

1.2 研究意义

报表作为一种信息组织和分析的有力手段,是企业信息系统的重要组成部分 之一,在各行各业中有着广泛的应用,如何快速有效地生成多种多样的报表一直 都是现代企业信息系统特别是管理信息系统中的一个重要课题。

在计算机出现以前,人们每天利用纸和笔来记录数据,然后隔一段时间会对 记录的数据进行分析处理。在计算机出现以后人们在Excel等工具上记录数据, 然后每隔一段时间对数据进行处理。这样就会有很多问题摆在我们面前,比如说 要改动数据的话,报表格式也必须跟着进行改动;数据和报表格式不分离,不便 于对数据进行分析处理;报表间的数据迁移也很困难。于是人们就像能不能开发 一种数据和报表格式分开,又能相互融合,可以任意设计报表格式且报表格式灵 活多变,还能对报表进行排版,并能对数据进行分析、处理和导出的软件呢。他 们想要的软件就是报表工具

报表工具能够把企业或政府的数据以自己设计的表格的形式直观展现出来, 而且报表工具不仅仅是数据展示,还能对数据进行分析处理。这样决策者可以从 这些分析过直观的数据看到不用报表工具很难看到的趋势、问题。从而制定更加 有效的决策,使公司的状态一直处于自己的掌握之中,让公司一直处于最好的运 行态。

目前,虽然市场上已经有了一些报表生成工具,但它们普遍存在自定义性差、 对需求变化适应能力弱的问题,不能很好地满足用户的需求。

设计和实现一个报表工具实现报表模板设计、报表生成、报表打印、和报表保存功能对各行各业加快信息化建设进程有着重大的意义。

1.3 研究内容

为了设计一个完美的报表工具,我们需要对一下内容进行研究:

- ▶ 研究国内外报表发展现状以及发展趋势,熟悉典型报表的结构、功能和操作方法,为设计和实现自己的报表工具做准备。
- 在有了一个报表工具模型的基础上学习所需的基础知识 c#^[5]、.net^[6]、 windows forms、visual studio2008 等。
- 在设计报表之前可以选择以前设计过的报表模型,并在模型上更改出需要的报表,当然也可以从空白页开始设计。
- 能够动态的设计报表,绘制报表的基本组成结构,如标签、表格、直线等,并在绘制过程中给予用户反馈。
- 能够对绘制的报表进行对齐,调整大小等排版功能。要求能够使用报表 上自带的控件进行排版,也可以用鼠标直接拖动,还可以用键盘控制, 要求可以使用正常 office 下的快捷键。

- ▶ 能够从数据库中把数据导入报表中,并可动态选择要导入的数据表。
- ▶ 能够对报表中的数据进行求和,排序等操作。
- ▶ 能过查看并更改控件的属性。
- ▶ 能过用把报表中的控件、数据源用 XML 格式保存,以便于 传输、存储、 查看^[7]。
- ▶ 能够对报表进行打印,可以打印预览,对打印页面进行设置,还可以对 打印机进行设置。
- ▶ 制作安装程序。安装后开发的报表工具能够在桌面上添加自己的图标, 在开始菜单中添加自己的菜单项。

1.4 论文结构

本论文从报表工具的背景、研究意义和研究内容出发,逐步讨论了报表生成 工具的设计于实现。

第二章从用例需求、功能需求、软硬件需求三个方面对报表生成工具进行了 需求分析。

第三章从数据模块设计、绘图模块设计、排版模块设计、读写 XML 模块设计、打印模块设计五个方面对报表生成工具进行了设计。

第四章从数据模块实现、绘图模块实现、排版模块实现、读写 XML 模块实现、打印模块设计、制作安装文件等六个方面对报表生成工具进行了实现。

第五章对本次毕业设计的总结和未来工作的展望

第二章 C#报表工具需求分析

本章阐述了报表生成工具的用户需求。分别从用例分析、功能需求、软硬件 需求等方面描述该报表生成工具应满足的要求。通过本章,读者可以对报表生 成器有一个初步的认识,并可以清楚报表生成器是一个什么样工具。

2.1 用例分析

用例图采用参与者和用例作为基本元素,以不同的视角展现系统的功能性需求。用例图是了解系统的第一个关口,我们通过用例图得知一个系统将会做什么,用例图如图 2-1 所示。



图 2-1 报表工具用例图

本报表工具的参与者是报表设计者,他把需要的数据从数据库中导出放入所用的控件中。报表设计者可以选择不同的报表样式,这可以满足不同行业的报表 设计者的设计需求。报表设计者还可以添加绘制控件,并且对报表排版,最后对 报表就行打印。

2.2 功能需求

报表生成工具需要实现的主要功能描述如下:

- 在设计报表之前可以选择以前设计过的报表模型,并在模型上更改出需要的报表,当然也可以从空白页开始设计。
- 能够动态的设计报表,绘制报表的基本组成结构,如标签、表格、直线等,并在绘制过程中给予用户反馈。
- 能够对绘制的报表进行对齐,调整大小等排版功能。要求能够使用报表 上自带的控件进行排版,也可以用鼠标直接拖动,还可以用键盘控制,

要求可以使用正常 office 下的快捷键。

- ▶ 能够从数据库中把数据导入报表中,并可动态选择要导入的数据表。
- ▶ 能够对报表中的数据进行求和,排序等操作。
- ▶ 能过查看并更改控件的属性。
- ▶ 能过用把报表中的控件、数据源用 XML 格式保存,以便于 传输、存储、 查看。
- ▶ 能够对报表进行打印,可以打印预览,对打印页面进行设置,还可以对 打印机进行设置。
- ▶ 制作安装程序。安装后开发的报表工具能够在桌面上添加自己的图标, 在开始菜单中添加自己的菜单项。

2.3 执行流程分析

本报表工具的执行流程应该是先在报表设计器中把报表的格式设计好,报表 格式包括报表绘制和报表排版。然后利用报表引擎对导入到报表中的数据进行处 理分析,然后打印,输出,保存。具体过程如图 2-2 所示。



图 2-2 报表工具执行流程

2.4 软硬件需求

本报表生成工具的软硬件需求如表 2-1 所示。

名称描述	具体内容	
运行硬件环境	普通 pc	
开发系统	WINDOWS XP/ VISTA/7	
开发环境	Microsoft Visual Studio 2008	
	.Net Framework3.5	
开发语言	C#	
构架模型	c/s	

表 2-1 软硬件需求

之所以选择 c#作为开发语言,是为了使用.Net Framework3.5^[8]的开发环境。 NET Framework 主要包含一个非常大的代码库, 里面封装了特别多的组件,如 打印需要的 PrintDocument、PrintPreviewDialog、PrintDialog 大大降低了报 表打印的开发难度,直接对 PropertyGrid 的引用也使控件的属性控制变得非常 容易。

第三章 C#报表工具设计

3.1 数据模块设计

3.1.1 服务器,数据库连接

创建一个连接服务器界面,在上面可动态输入选择需要连接服务器的名称、登录名、密码、数据库类型、数据库名称来登录特定服务器中的特定数据 库。

3.1.2 报表数据绑定

本报表工具是主从表绑定,主表绑定标签,从表绑定表格。本绑定利用 bindingsource 组件^[9], bsMain 是主表 bindingsource, bsGrid 是表格 bindingsource。把数据集的数据导入 bindingsource,数据集的数据来自"报表数 据选择"这个界面。我们通过 SQL 语句把数据库中的所有表名加入数据集绑定 到 ComboBox 上,用户可以在 ComboBox 的下拉字段中看到所有的列表的名称。这样用户可以直接选择所需的报表数据绑定到 bsGrid。

3.2 绘图模块设计

3.2.1 封装控件

按照需求分析,本报表工具应该像 visual studio 一样拥有自己的控件,可以像制作 windows forms 一样设计实现想要的报表。但本报表工具还要保证简单 易用且功能齐全,所以需要自己封装控件。画一个报表需要的控件及其功能如表 3-1 所示。

控件	功能
标签	绘制一个矩形,可以在上面添加文字,也可以从数据库中导入
矩形	绘制一个矩形框
直线	绘制一条直线,方向可调
检查框	比标签多了一个选中框
表格	可以从数据库中导入数据到表格中
图形	可以在图形空间中添加图形

表 3-1 控件名称及功能表

本报表生成器还应该封装一个属性窗口,以便于对控件的属性进行修改,显示。

9

3.2.2 添加、处理控件

如何像 visual studio 一样动态的绘制控件, 鼠标在每台起时绘制一个虚的矩 形框, 抬起时在虚矩形框处画出一个控件, 这就要用到鼠标事件。同时, 为了提 高本报表的易用性, 我们还要加入键盘事件, 鼠标键盘事件。其中快捷键的设计 如表 3-2 所示。

表 3-2 快捷键

快捷键	功能
F4	关闭窗口
Ctrl+P	打印
Ctrl+b	预览
Ctrl+s	保存
Ctrl+z	撤销(可以撤销时)
Ctrl+y	重做(可以重做时
Ctrl+x	剪切(选中控件集合中有控件)
Ctrl+c	复制(选中控件集合中有控件)
Ctrl+v	粘贴(粘贴控件集合中有控件)

3.3 排版模块设计

我们要有 windows forms 上的常用的排版功能,具体功能如表 3-3 所示。

	表 3-3	报表工	具的	排版	功能
--	-------	-----	----	----	----

名称	功能
左对齐	控件左侧和标准控件左侧对齐
居中对齐	控件中间(水平方向)和标准控件中间对齐
右对齐	控件右侧和标准控件右侧对齐
顶部对齐	控件顶部和标准控件顶部对齐
中间对齐	控件中间(垂直方向)和标准控件中间对齐
底部对齐	控件底部和标准控件底部对齐
使宽度相同	使控件的宽度和标准控件相同
使高度相同	使控件的高度和标准控件相同
使大小相同	使控件的大小和标准控件相同
使水平间距相等	选中控件之间的水平距离都相等
使垂直间距相等	选中控件之间的垂直距离都相等
水平相邻垂直相邻	控件之间在 x 方向上相邻
垂直相邻	控件之间在 y 方向上相邻
水平居左	控件位于页面最左边
水平居中	控件位于页面正中间
水平居右	控件位于页面最右边

有了这些排版功能,用户在对报表排版时不用自己手动鼠标校准控件的位置。可以很准确的确定控件该出现的位置,大大简化了绘制报表的难度。

3.4 读写 XML 模块设计

现在主流报表工具的数据存储和输出都有 XML^[10]格式。这是因为 XML 格 式有其特殊的优点。首先报表内容的数据大多是存在数据库中的,而报表本身的 框架是以 XML 格式存储的,这就可以发挥 XML 存储小型数据方便的优势,而 忽略大量数据 IO 瓶颈的劣势; XML 通用性强,可以在任何平台上使用,便于传 输;利于二次开发。

本报表上的控件也都是以XML格式存储的,打印也需要XML格式的文件。 由于我们的控件只涉及到自己封装的六个,属性也是常见的。所以读写XML相 对容易,只需要考虑这六种控件和相关属性就行。XML 记录的内容如表 3-4 所 示。

11

控件	记录属性		
标签	标签名、位置、大小、背景色、边框颜色、文字位置、位子内容、字		
	体		
直线	直线名、位置、大小		
矩形	矩形名、位置、大小		
检查框	名字、默认检查、位置、大小、背景色、文字、是否选中检查框		
表格——自身	位置、大小、边框风格、背景色、是否允许有空列、记录选择项、是		
	否允许删除、是否允许更新、是否允许添加新列等		
表格——表格内	表头、数据字段、可见性、背景色、文字位置等		
图片	名称、位置、大小、边框类型、图片路径		

表 3-4 XML 文件控件属性

XML 节点具体结构设计如图 3-1 所示。

1	K?xml version="1.0" encoding="UTF-8"?>
2	<report ″="" 属性=" ″属性="></report>
3	<controls></controls>
4	<控件种类 Name=
5	<控件种类 Name="控件名" 属性=:: "属性=:: " 属性=:: " 漏性=:: " 漏性=:: ") 属性=:: ") />
6	<cltruedbgrid 1"<="" name="″表名″" td="" ″="" 属性=" ″</td></tr><tr><td>7</td><td></td></tr><tr><td>8</td><td><pre><ClDataColumn Tag="></cltruedbgrid>
9	<cldatacolumn "="" "属性="" tag="2" 属性=""></cldatacolumn>
1	0
1	1 Shifti was mished and a metal of a planet of a pl
1	2 〈经件种类 Name= 经件名 属性= 属性= 属性= 属性= 属性= />
1	3
14	4

图 3-1 XML 格式设计

图 3-1XML 文件中的数字是我为了描述方便加进去的,实际的 XML 文件中没 有那一列数字。第1行是 XML 序言,每一个 XML 文件都有序言,xml version="1.0"^[11] 用来告诉解析器和浏览器,这个文件应该按照 1.0 版本的 XML 规则进行解析。 encoding = "utf-8"^[11]表示此 xml 文件采用 utf-8 的编码格式。第2行用来描述一 个报表文件的名称、大小、位置等属性。第三行是开始描述控件。第4、5、12 行是描述非表格类控件的 XML 节点。第6、7、8、9、10、11 描述的是表格控 件,其中第6 行是表格的各种属性,8、9 是描述表格内容的属性,一行对应表 格中的一行数据。第10、11、13、14 是对应描述的结束标志,这是 XML 格式 的规定。

3.5 打印模块设计

打印需要调用 windows 自带的几个 API 函数,我们先介绍一下 API 函数^[12]。

DllImport("winspool.Drv", EntryPoint="AddForm", SetLastError=true, CharSet=CharSet.Unicode,

ExactSpelling=false, CallingConvention=CallingConvention. StdCall) kernel32.dll——定位参数,指定了包含要导入方法的动态链接库名称。 EntryPoint——指示要调用的 DLL 入口点的名称或序号。

SetLastError——指示被调用方在从属性化方法返回之前是否调用 Win32 API 函数。

ExactSpelling——控制 System. Runtime. InteropServices 字段是否使公 共语言运行库在非托管 DLL 中搜索入口点名称,而不使用指定的入口点名称。

CallingConvention: 指示入口点的调用约定。

本次打印需要的 API 函数有如表 3-5 所示。

EntryPoint	API 函数涵义
OpenPrinter	用于打开指定的打印机
ClosePrinter	关闭指定的打印机
AddForm	增加打印格式
DeleteForm	删除打印格式
GetLastError	获取最后一个错误
CreateDC	创建打印输出格式
ResetDC	重新设置打印的输出格式
DeleteDC	删除打印机的输出格式
	EntryPoint OpenPrinter ClosePrinter AddForm DeleteForm GetLastError CreateDC ResetDC DeleteDC

表 3-5 API 函数表

3.6 用户界面设计

3.6.1 主界面

报表工具是一种应用性很强的软件,需要良好的人机交互。良好的人机交互 需要"以用户为本"的思维方式,把界面做的简单,清晰。大胆的除去用不到的 功能,把最常用的功能放在最明显的地方。设计报表的主界面如图 3-2 所示。

图 3-2 报表工具主窗口设计图

工具栏没有使用常用的菜单栏,因为菜单栏不能使用户一眼就看到功能,也 不知道自己想要的控件在哪个菜单下。由于我们只显示常用的功能,所以我们可 以用工具条列出每一个功能,这样可以使用户很快熟悉本报表生成工具。

3.6.2 封装控件

此报表生成器自定义了六种控件。其中 CheckBox 也就是检查框控件, 是.net 的 System.Windows.Forms.CheckBox 类型; Label 是标签控件, 是.net 的 System.Windows.Forms.Label; Line 是线段控件, 是 XReport.ReportLine 类型; Rectangle 是矩形控件, 是 XReport. ReportRectangle 类型; PictureBox 是图形控 件, 是.net 的 System.Windows.Forms. PictureBox 类型; TrueDBGrid 是表格控 件, 是C1.Win.C1TrueDBGrid.C1TrueDBGrid 类型。这六种控件每一个对应一个 工具条上的按钮^[13]。

3.6.3 属性窗口

除了上述六种控件外,本报表生成器还封装了一个属性窗口,之所以不用 VS 自带的属性窗口是因为 VS 上面的属性很多我们是用不到的,而且上面的属 性是英文的。所以我们要建立一个属性很少,并且是中文的属性窗口^[14]。属性 窗口设计如图 3-3 所示。

	分类方法
属性分类	

图 3-3 属性窗口设计图

属性窗口可以按分类排序,比如尺寸类包括高读宽度水平位置等;也可以按 属性的字母顺序排列。

上述六个控件有很多属性。把这些属性设置成枚举型的,用户就可以直接 在下拉菜单中选择。具体内容如表 3-6 所示。

属性	枚举值					
边界	有、无					
可见性	可见、不可见					
锁定	锁定、不锁定					
页码	不是页码、数字页号、数字页码、中文页号、中文页码、全局					
	数字页号、全局数字页码、全局中文页号、全局中文页码					
标签文字排列	居左上、居左中、居左下、居上中、居正中、居下中、居右					
	上、居右中、居右下					
选中	是、否					
表格文字排列	居左、居中、居右					
直线方向	水平、垂直					
自动显示滚动条	是、否					
显示列标题	是、否					
显示网格线	是、否					
允许调整行高	是、否					
打印页合计	是、否					
显示格式	不格式化、数量格式、单价格式、金额格式、日期格式、日期					
	时间格式、时间格式、百分数					
图形缩放	正常、缩放、自动、居中					

表 3-6 属性窗口属性设计表

除了定义上述属性外,六个控件中还要封装自己的属性。每一个控件还把 自己的属性分组,这是通过[Description("设置文字"), Category("文字")]attribute (特性)实现的。每一个控件里还实现了中英文转换和属性保存到内存中 Schema。然后把属性放到属性窗口的属性集中。属性窗口继承了 Form,并利用 了.net 的 PropertyGrid^[15]:提供用于浏览对象属性的用户界面。并在里面设置了 属性集,以便和外界交互。

16

第四章 C#报表工具实现

4.1 数据模块实现

4.1.1 连接数据库服务器

利用.net 自带的控件 TextBox,把上面输入的字符串存起来,实现了动态输入选择需要连接服务器的名称、登录名、密码、数据库类型、数据库名称来登录特定服务器中的特定数据库。同时,为了简化用户操作,我在服务器名称、登录名、密码处设立了默认输入,在数据库类型、和名称处设立了下拉可选输入。服务器、数据库连接实现如图 4-1 所示。

服务器名称/IP:	SUNFUQIANG-PC
登录名:	sa
密码:	123456
SQL类型:	
数据库:	-
	び 连接 🚺 取消
连接到Sever	
	d

图 4-1 连接数据库服务器

4.1.2 报表数据绑定

本报表工具是主从表绑定,主表绑定标签,从表绑定表格。本绑定利用 bindingsource 组件, bsMain 是主表 bindingsource, bsGrid 是表格 bindingsource。把数据集的数据导入 bindingsource,数据集的数据来自"报表数 据选择"这个界面。我们通过 SQL 语句把数据库中的所有表名加入数据集绑定 到 ComboBox 上,用户可以在 ComboBox 的下拉字段中看到所有的列表的名称。这样用户可以直接选择所需的报表数据绑定到 bsGrid。数据导入如图 4-2 所示。

	■ 报表数据选择
请选择所需报表	请选择所需报表
学生信息	学生信息 ▼
确定	课程成绩 学生成绩 销售订单 ³³³ 测试表 学院管理信息
(a) 数据导入下拉框	(b) 数据导入下拉框拉开后

图 4-2 数据导入表

4.2 绘图模块实现

报表工具的绘图模块和排版模块都是在的主界面下实现的,报表工具主界面 如图 4-3 所示。

新建编辑 礼	咽图项目 生成	格式	绘						
件属性窗口		4 ×		.123	.4567	89 10	1112	13 14	. 15 16.
			:						
类型	文字标签	*	i		20	11全国硕士	主统一者	针试	
其他				5. .					
页码	不是页码		Ż						
设计	71		÷			分女	: 1		
钡定控件	个领定		3			准有	TIE		
数据子段 数据字段	(穴)								
気は子校	(土)		1						
外观	1 14-010		5		考生编号	10056100010655	0		100
背景色	White		:		考生姓名	孙富强			
边界	有		6		्र / में आ	日日白小一	LL TH	Ð	
前景色	Black	=	:		证件失望	居民矛份证	作力	カ	STRA ASS
文字排列方式	居左中		Ť		证件号码	21050419880707	0295		
文字	RDAMIT		:		学习工作单位	天津大学			
父子 行物	后氏身切脏		Ê			(1005() 5 %	n/s		
可见性	តណ្		ġ		报考单位 (代码)	(10056) 大泽力	(乎		
字体					报考专业 (代码)	(081200) 计算	机科学与技	*	
〕字体	楷体, 12pt		10						
			1						
2/*			11						
F F F									

4.2.1 添加控件

利用 new 函数,新生成一个控件。其中 CheckBox 也就是检查框控件,是.net 的 System.Windows.Forms.CheckBox 类型; Label 是标签控件,是.net 的 System.Windows.Forms.Label; Line 是线段控件,是 Junxian.XReport.ReportLine

类型; Rectangle 是矩形控件, 是 Junxian.XReport. ReportRectangle 类型; PictureBox 是图形控件, 是.net 的 System.Windows.Forms. PictureBox 类型; TrueDBGrid 是 表格控件, 是 C1.Win.C1TrueDBGrid. C1TrueDBGrid 类型。

在 Add_Control 方法里判断添加的事哪个控件,这里用一个 string 类型变量 dragcontrol 标记选取的是哪个控件。这里以 Label 为例,说明控件的添加过程。 OnLabel 事件在点击标签按钮时被触发,这里有一个小问题,就是要考虑 dragcontrol 的值,如果 dragcontrol== "Label"说明已经点过了,再点一次的话应 该就取消添加控件操作,既把 dragcontrol 返回之前的值,否则设 dragcontrol = "Label"。

Add_Control 是由 page 上的鼠标事件触发的,当鼠标抬起时开始添加控件。 控件的起点坐标、长度和宽度在鼠标事件中也有记录,我将在鼠标事件中阐述。 在这里还要调用 CreateLabel 方法,这个方法是用来控制控件越界,初始化控件 名字、颜色、文字、起点坐标、长宽、边框风格、位置等属性。该方法还负责把 控件添加到控件集合和已选控件集合。

添加的控件需要先存写到内存中,保存时再存入写入硬盘,控件的名字、位置、大小、背景色、边框色、文字等属性均被序列化成 XML 格式,并存入 XML 文件格式集合 xcSchema 中。

4.2.2 鼠标键盘事件

本报表软件的很多功能都是由鼠标、键盘事件触发的。如添加控件、拖动 改变控件大小,拖动改变控件位置。鼠标键盘事件中的如何绘制控件、控件周 围的小四边形,如何改变控件大小,拖动控件改变大小,拖动控件位置将在后 文介绍。

4.2.2.1 page 上的鼠标事件

page 是 form 上的一个 panel (容器),在它上面有三种鼠标事件, Panel_MouseDown, Panel_MouseMove, Panel_MouseUp^[16]。

Panel_MouseDown 事件利用自带的参数 e,确定了鼠标此时的横纵坐标。然后根据控件集合 alControl 中记录的控件位置判断自己的鼠标形状。如果此时的鼠标形状等于一个设定的字符串"none"时,说明此时鼠标位于 page 的空白处(此处没有控件),而且此前没有点击标签的控件。那么这时应该清空选中控件集合 alControl。如果此时的形状是八种(上下左右和两条对角线)双向对角线图标,说明鼠标位于某选中控件周边的八个小四边形上。

Panel_MouseMove 事件也有参数 e,这里要考虑边界问题,如果越界,坐标 位置设为界面的最大宽度和高度值。这里的参数 e 是动态记录的,每移动一 下, e 变一次。我们可以根据这里的 e 值和 Panel_MouseDown 里的 e 值确定本次

19

移动产生矩形的宽高。再利用画图函数,我们就实现了根据鼠标移动动态的画 一个虚线矩形。

Panel_MouseUp,当抬起的鼠标右键时,出现一个弹出菜单,上面有显示 隐藏元素、打印预览、打印、页面设置、打印设置、剪切、复制、粘贴等功能, 具体实现如图4-4所示。当选中控件集合为空时剪切、复制选项是灰色的,当系 统剪切板为空时,粘贴建也是灰色的。Panel_MouseUp 的两项重要功能就是提 起鼠标时根据鼠标的形状判断是选中控件,还是拉伸控件。根据 dragcontrol 的 值判断是否添加控件。

	显示隐藏元素
à,	打印预览
6	打印
	页面设置
Â	打印设置
¥	剪切
	复制
12	粘贴

图 4-4 右键菜单

添加控件已经在上面介绍了,这里讲一下拉伸控件和选中控件。拉伸控件 事件其实是在 Panel_MouseDown 事件时就已经确定的了。根据鼠标形状确定特 定的拉定方向,大小变化时只取拉伸方向的偏移量,拉伸后新位置、大小要保 存到内存,属性窗口要更新。选中控件鼠标始终是默认形状的,我们把选中控 件分为两部分:一部分是画的矩形虚框四个边压中的控件,一部分是画的矩形 虚框内部的控件。矩形虚框的坐标和大小已经被记录。page 上的所有控件的大 小和位置也是被记录的,他们被记录在 Controls 里,这是.net 自带的功能。

最后画出选中,添加,或拉伸的控件,并在控件周边画出八个小四边形。

4.2.2.2 Control 上的鼠标事件

control_MouseDown 是要和键盘事件(Shift 或 Ctrl)配合使用的。

首先讨论不按键键盘键时的鼠标操作:

确定标准控件。标准控件就是标杆控件,我们在给报表排版时会用到这个 控件,其它的控件都要根据这个控件来排版,比如左对齐,使高度相同,是水 平距离相等等。如果当前选中控件集合包括当前鼠标点击的控件并且控件集合 只有一个控件,那么肯定没有标准控件,正常画一个控件,并在周围画八个白 色为填充色,黑色为边框的小四边形就行;如果控件集合个数大于一,并且原 来有标准控件,则把标准控件变为正常控件,把现在的控件变为标准控件,如 果没有标准控件,那么设当前控件就是标准控件。标准控件周围的八个小四边 形是黑色填充色,白色边框。在这里还要确定控件是否在锁定控件里,如果是 还要在周围加上锁定框。

如果选中控件不在控件集中,那么清空控件集,然后把选中控件加入控件 集中,设标准控件为空。这里也要判断是否是锁定控件。

当鼠标按下时, Shift 或 Ctrl 已经按下:

如果选中的控件已经在控件集合中,那么把这个控件从选中控件集合中移除,清空 page 上这个区域的所有控件,再在 page 上画上 page 上的所有控件, 然后再给选中控件集合里的控件周围加上八个小四边形。并设控件集合中的第 一个为标准控件。

如果选中的控件不在选中控件集合中,如果原来的选中控件集合为空,直 接把这个控件周围加上八个小四边形,如果原来的选中集合不为空,把标准控 件变为正常控件,把此控件设为标准控件,并画相应的八个小四边形。

这里还要计算出控件四个脚的极值,这是为 control_MouseMove 中的移动 控件做准备。当然最后要把选中控件加到选中控件集合中。

control_MouseUp 事件处理的都是一些基本内容,像根据此时鼠标位置得到 一些有关的位置,从新确定控件位置,更新 page,更新属性窗口,在选中控件 周围画八个小四边形。

control_MouseMove 这里根据鼠标的位置变化鼠标的形状,如果在控件上, 鼠标就变为十字形。这里有一种控件是特殊的,就是表格控件。当鼠标位于表 格控件的第一行位置时鼠标形状不变,位于其它行时才会变化。这是因为位于 第一行时,把鼠标放于列之间时会变为左右双向箭头,可以改变列宽度。如果 鼠标左键按下时,控件被重画,重画操作将在下面介绍。

4.2.2.3 Form 上的键盘事件

Report_KeyDown 当你按下键盘上的 Delete 键时,会删除选中的控件,删除 功能将在下面介绍。还可以通过上下左右键将控件向特定方向移动。如果 Ctrl 键和上下左右键同时使用,左键和上键是向左边和上边缩小控件,下键和右键 是向右边扩大控件。

同时按下 Ctrl 键和 A 键,可以选中所有的控件。原理就是把选中控件集合 先清空,然后把 page 的 controls 集合中的控件在重新绘制一遍,并在周围加上 八个小四边形。

Report_KeyUp,可以用来促发上面介绍的快捷键。

Ctrl 和 Shift 配合上下左右键使用时,当键抬起时需要刷新,重新绘制矩形,绘制周围的八个小四边形。同样上下键单独使用也要刷新,重新绘制矩

形,绘制周围的八个小四边形。

4.2.2.4 鼠标形状的变化

对于控件集合中的每一个控件, 鼠标位于控件周围的小矩形上时, 鼠标会 根据小矩形未知的不同而变成不同的方向的双向箭头。当推拽标记 dragcontrol 不是 "Arrow"时, 鼠标形状变为十字线光标。

4.2.2.5 控件四个脚的极值和最小高度和最小宽度

首先把起点横坐标的最小值设为 page 的最大宽度,起点的纵坐标的最小值 设为 page 的最大高度;起点和终点横坐标及纵坐标的最大值设为 0,最小宽度 和高度设为 page 的最大宽度和最大高度。然后根据 slRec 中存储的控件的位置 和大小(slRec 是 SortedList 类型的,它存放控件名称和一个记录矩形横纵坐标 和宽高的实例),确定每个控件的起点和终点的横纵坐标。跟上面设定的初始值 做比较,最大的取其中的大值,最小的取其中的小值。其中代码含义如表4-1所示。

表 4-1	控件极值

代码	含义
cx1min	所有控件起点中横坐标的最小值
cx1max	所有控件终点中横坐标的最大值
cx2max	所有控件终点中横坐标的最大值
cy1min	所有控件起点中纵坐标的最小值
cy1max	所有控件起点中纵坐标的最大值
cy2max	所有控件终点中纵坐标的最大值
Cwmin	所有控件宽度的最小值
Chmin	所有控件高度的最小值

4.2.3 图形绘制

4.2.3.1 绘制控件边界八个小矩形

定义一个画矩形函数, DrawRec(int x1,int y1,Color backColor,Color borderColor), 其中 x1, y1 分别是矩形的横纵坐标, backColor 是背景色, borderColor 是边框颜色。定义一个 SolidBrush, SolidBrush 的颜色定义为背景色; 再定义一个 Pen, Pen 的颜色定义为边框颜色, 粗细定义为 1.然后利用 GDI+ 绘图 FillRectangle 填充一个边宽为 6 矩形, 用 DrawRectangle 画一个矩形边长为 6 的矩形。然后在空间的左上顶点、上中点、右上顶点、左中点、右中点、坐下 顶点、下中点、右下顶点分别调用这个绘图函数。控件选中前后对比如图 4-4 所



图 4-4 选中控件前后对比实现图

4.2.3.2 画动态矩形框

定义一个重新绘制矩形函数,以动态跟踪矩形框的变化 ReiniRec(int adx,int ady,int adwidth,int adheight), adx 和 ady 分别是矩形横纵坐标的增加值, adwidth 和 adheight 分别是高度和高度的增长值。利用自己定义的 DrawRectangle (这个 和上面的 DrawRectangle 不同,上面的是 GDI+自带的方法),此方法调用了.net 自带的绘制可逆框架的函数 DrawReversibleFrame,实现了矩形框的动态变化。 锁定矩形框也用到 ControlPaint 类,只不过是用 DrawLockedFrame 方法。

4.2.3.3 拉伸动态矩形

动态拉伸要联系上面鼠标键盘事件中的鼠标形状变化一起使用。被分为上下左右四个方向进行考虑,凡是跟上方向有关的(左上顶点、上中点、右上顶点)都算在上的方向上。当 addy<0 时,向上,并且要考虑越界问题,比如如果 addy 和原控件最小纵坐标之和小于六,这时只能拖动到离边界为六的距离,因为当选中控件时要画边长为六的小矩形;当 addy>0 时,向下,这里也要考虑越界问题,比如如果控件最小高度减去 addy 小于八,这时只能拖拽到高度为八时,因为边框的粗细是 1,小四边形的边长是 6,如果高度是 8,选中控件时无法画中间的小矩形。左右下方向同理。

4.3 排版模块实现

4.3.1 跟标准控件对齐

在设计模块中我们已经列出了所有的排版功能,这里通过介绍左对齐和顶部 对齐来说明左对齐、居中对齐、右对齐、顶部对齐、中间对齐、底部对齐这六种 对齐方式是如何实现的。

左对齐: 首先要备份控件以便后面的重做, 重做将在后面介绍。这个控件 不能是锁定控件。然后确定标准控件的横坐标, 控件起点的横纵坐标。这里要 考虑越界问题, 标准控件的起点横坐标加上控件宽度必须小于 page 宽度, 否则 不能对齐。左对齐其实就是把原控件去掉,然后在对齐后的位置重新画了一个 矩形。这里定义了一个函数 ReiniSIRec(string cname, int adx, int ady, int adwidth, int adheight), cname 是控件名称, adx 和 ady 分别是这个控件横坐标和纵坐标的 增加/减小值, adwidth和 adheight分别是宽度和高度的增加/减小值。根据这些条 件可以获得变化后控件的位置、大小,并画新控件。左对齐只需令 adx 等于标 准控件起点横坐标减去当前控件起点横坐标。居中对齐、右对齐也是至更改 adx 即可。顶部对齐、中间对齐、底部对齐更改 ady 即可。

4.3.2 比照标准控件调整大小

这里介绍使宽度相同、使高度相同、使大小相同三种排版方式。这些功能和 上面对齐功能一样需要备份、考虑越界,只不过这里改变 ReiniSlRec 中的参数 不同。宽度改变 adwidth,高度改变 adheight,自然大小就同时要改变 adwidth 和 adheight。

4.3.3 调整选中控件之间的距离

这里介绍了使水平间距相等、使垂直间距相等、水平相邻、垂直相邻等四种 排版方式的实现方法。主要介绍使水平距离相等和水平相邻。水平距离相等首先 把控件按 x 坐标排序,然后计算控件之间的距离和,然后将总距离平分。距离 之和为起点横坐标最小值减去起点横坐标最大值,再减去不是拥有最小和最大 横坐标的控件的宽度,最后减去拥有最小横坐标控件的宽度,把控件之和除以 控件数减一,便得到控件之间的平均距离。之后把控件起点 x 坐标加上控件宽 度加上控件平均距离,便得到控件起点横左边,纵坐标不变,宽高不变。之后 按 x 排列顺序重新绘制所有除起始控件和终止控件的控件。使垂直间距相等同 理。

水平相邻也要把控件按 x 坐标排序,这里是一个一个控件画,前一个控件的起点横坐标加上控件宽度减去 1,是第二个控件的起点的横坐标,利用 ReiniSlRec 函数只需改变横坐标。

4.3.4 页面对齐

这里是最常见的三种对齐方式:水平居左、水平居中、水平居右。水平居左 就是利用 ReiniSlRec 函数,把用 6 减去每个控件起点的横坐标作为 adx。居中和 居右的原理都是一样的。

4.4 读写 XML 模块实现

4.4.1 读 XML

这里先介绍一种存储结构, xcSchema 是 xmlComponentSchema^[17]类型的。

xmlComponentSchema 是单据、查询、报表中控件定义的基类,这里定义了所 有控件的公共属性,控件集合。xcSchema 是一整张报表里所有控件的 XML 表 示,存放了所有控件的所有信息。报表中的新建的所有控件都是先存入内存中 的 xcSchema 结构,燃用利用递归函数遍历 xcSchema 中的所有节点的所有信息 存入磁盘中。XML 文件在浏览器中查看如图 4-5 所示。 <ReportLine Name="ReportLine1" Location="89, 60" Size="631, 18" /> <Label Name="Label1" Location="307,18" Size="212,39" BackColor="White" BorderStyle="None" TextAlign="MiddleLeft" Text="天津大学学生档案" Font="宋体, 18pt" /> <Label Name="Label2" Location="6,180" Size="100,23" BackColor="White" BorderStyle="FixedSingle" TextAlign="MiddleLeft" Text="性别" /> <Label Name="Label4" Location="6,203" Size="100,21" BackColor="White" BorderStyle="FixedSingle" TextAlign="MiddleLeft" Text="年龄" /> <Label Name="Label11" Location="106,180" Size="378,23" BackColor="White" BorderStyle="FixedSingle" TextAlign="MiddleLeft" Text="男" /> <Label Name='Label12' Location='106,156' Size='378,24' BackColor='White' BorderStyle='FixedSingle' TextAlign='MiddleLeft' Text='张三' /> <a href="label13" Location="107,203"
Size="0,0" BackColor="White" BorderStyle="FixedSingle" TextAlign="MiddleLeft" Text="文字" /> <Label Name="Label14" Location="106,203" Size="378,23" BackColor="White" BorderStyle="FixedSingle" TextAlign="MiddleLeft" Text="22" /> <Label Name="Label21" Location="106, 203" Size="82,5" BackColor="White" BorderStyle="FixedSingle" TextAlign="MiddleLeft" Text="文字" /> <Label Name="Label21" /> <Label Name="Label22" Size="0,0" /> Location="6,156" Size="100,24" BackColor="White" BorderStyle="FixedSingle" TextAlign="MiddleLeft" Text="姓名" /> C1TrueDBGrid Name="C1TrueDBGrid1" Location="83,405" Size="412,178" BorderStyle="FixedSingle" BackColor="White" FlatStyle="Flat" EmptyRows="True" RecordSelectors="False" AllowDelete="False" AllowUpdate="False" AllowAddNew='False" AllowColSelect="False" MultiSelect="None" MarqueeStyle="FloatingEditor" RowDivider.Color="198, 195, 198" WrapCellPointer="True" DataSource="" ColumnFooters="True" PageFooter="true"> - <Columns> <C1DataColumn Tag="1" Caption="学院" DataField="学院" DisplayColumn.Visible="True" DisplayColumn.AllowSizing="True" DisplayColumn.HeadingStyle.ForeColor='ControlText' DisplayColumn.HeadingStyle.BackColor='White' DisplayColumn.HeadingStyle.HorizontalAlignment='Center' DisplayColumn.Style.ForeColor="WindowText" DisplayColumn.Style.BackColor="Window" DisplayColumn.FooterStyle.ForeColor="ControlText" DisplayColumn.FooterStyle.BackColor="White" /> <C1DataColumn Tag="2" Caption="专业" DataField="专业" DisplayColumn.Visible="True" DisplayColumn.AllowSizing="True" DisplayColumn.HeadingStyle.ForeColor="ControlText" DisplayColumn.HeadingStyle.BackColor="White" DisplayColumn.HeadingStyle.HorizontalAlignment="Center" DisplayColumn.Style.ForeColor="WindowText" DisplayColumn.Style.BackColor="Window" DisplayColumn.FooterStyle.ForeColor="ControlText" DisplayColumn.FooterStyle.BackColor="White" /> <C1DataColumn Tag="3" Caption="负责人" DataField="负责人" DisplayColumn.Visible="True" DisplayColumn.AllowSizing="True" DisplayColumn.HeadingStyle.ForeColor="ControlText" DisplayColumn.HeadingStyle.BackColor="White" DisplayColumn.HeadingStyle.HorizontalAlignment="Center" DisplayColumn.Style.ForeColor="WindowText" DisplayColumn.Style.BackColor="Window" DisplayColumn.FooterStyle.ForeColor="ControlText" DisplayColumn.FooterStyle.BackColor="White" /> <C1DataColumn Tag="4" Caption="联系电话" DataField="联系电话" DisplayColumn.Visible="True" DisplayColumn.AllowSizing="True" DisplayColumn.HeadingStyle.ForeColor="ControlText" DisplayColumn.HeadingStyle.BackColor="White" DisplayColumn.HeadingStyle.HorizontalAlignment="Center" DisplayColumn.Style.ForeColor="WindowText" DisplayColumn.Style.BackColor="Window" DisplayColumn.FooterStyle.ForeColor="ControlText" DisplayColumn.FooterStyle.BackColor="White" /> </Columns> </C1TrueDBGrid> <ReportRectangle Name="ReportRectangle1" Location="535, 245" Size="86, 62" /> checkBox Name="CheckBox1" AutoCheck="False" Location="371, 266" Size="69, 62" FlatStyle="Flat" BackColor="White" Text="good" Checked="true" /> />PictureBox Name="PictureBox1" Location="566,388" Size="66,76" BorderStyle="FixedSingle" Image="E:\picture\Inch photos\OO截图未命名.png" /> 图 4-5 XML 文件实现图

4.4.2 写 XML

读 XML 就是用递归函数 LoadXmlComponent(object obj, ArrayList childCotrols, Junxian.XReport.XmlComponentSchema xcSchemaParent)遍历 XML 中的所有节点,并把所有节点的所有结构读入内存中的 XmlComponentSchema 结构。

Obj——当前节点 childCotrols——所有子节点 xcSchemaParent——节点存入的位置

4.5 打印模块实现

4.5.1 打印页面设置

打印的很多部分都要用到 PrintDocument 这个类,它的 VS 定义如下:定义 在从 Windows 窗体应用程序进行打印时,将输出发送到打印机的可重用对象。 我们可以把它理解成一张可以重复放要打印对象的纸。

打印对话框是一个新类,它继承了 VS 自带的窗体类。并在上面添加了纸张 大小、来源、长度、宽度、方向、四个方向的边距、添加/删除纸张格式等属 性。

这里要考虑的一个问题就是单位换算。.net 内部中采用的是英制的计量单 位英寸,而我们在页面设置中用的是公制计量单位厘米,所以我们在此要进行 一下换算。页面设置如图 4-6 所示。

页面设置				×
纸张				
大小: 🗛	•	高度:	29.6926	厘米
来源: 自动选择	•	宽度:	21.0058	厘米
纸张方向	页边距 (厘米)			
◎ 纵 向	左: 2.54	右:	2.54	
◎ 橫 向	上: 2.54	下:	2.54	
添加	删涂	确认	Į.	则消

图 4-6 页面设置对话框

4.5.2 打印设置

这里再介绍一个.net 自带的类 PrintDialog: 允许用户从 Windows 窗体应用 程序中选择一台打印机,并选择文档中要打印的部分。这里也要用到 PrintDocument 这个类,主要是用到这个类中的 PrinterSettings: 获取或设置对文 档进行打印的打印机。打印设置如图 4-7 所示。

打印		100	1.81	×
-打印机- 名称 (8) 状态: 类型: 位置:	: Microsoft 准备就绪 Microsoft) XPSPort:	XPS Document Writer XPS Document Writer		属性(2)
备注:				打印到文件(L)
打印范围	E		份数	
全部	(A)		份数(C):	1
 页码》 页码》 ① 选定》 	范围(G) 从(F) 范围(S)	: 0 到(T): 0	11 22	33 自动分页 (0)
				确定 取消

图 4-7 打印设置

4.5.3 打印预览

实现打印预览当然要用到.net 自带的 PrintPreviewDialog 类。首先实例 化这个类, 然后实例化一个 PrintDocument, 把这个实例化的 PrintDocument 作 为要打印的文档, 然后显示实例化的 PrintPreviewDialog 就可以了。打印预览 效果如图 4-8 所示

📲 打印预罚	5		1						x
⊜ ∕ -		···· 88	🎬 关闭(C)					页(P)	1 🌲
Г]	•
					al de	يلى ريك			
			20	11全固领士	±鈗_	有试			
				15 H	• -				E
				准考	цĿ				
			考生稿号	10056100010655	90		1		
			石田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田	476位 居民身份征	性別	3			
			证件号码	21050419880707	0295		ana yake		
			学习工作单位	天津大学					
			报考单位 (代码)	(10056) 关注:	t.≄				
			报考专业 (代码)	(081200) 対第	机科学与	技术			
									-

图 4-8 打印预览实现图

打印预览包括查看多页打印的效果,打印放大等功能。

4.6 制作安装文件

打开 VS 编辑器,点击新建项目,选择:其他项目类型->安装与部署->安装项目,然后点确定.

双击"应用程序文件夹"在右边的空白处右击,选择添加->文件,将可执行文件导入,这时跟可执行文件有关的动态生成库文件都会被导入.右击可执行文件, 创建快捷方式,然后把快捷方式分别复制到左边的"用户的'程序'菜单"和"用户 桌面"中.这样安装程序安装安装完成后会在"开始->所有程序"和"桌面"上生 成程序的快捷方式.

右 击 左 边 的 " 应 用 程 序 文 件 夹 " 打 开 属 性 对 话 框:将 属 性 中 的 "defaultlocation"的路径中的"[manufacturer]"去掉,不然的话做好的安装程 序默认安装目录会是"c:\programm file\你的用户名\安装解决方案名称";

在打开的系统必备页中,选中.NET framework2.0选择项选上以后,在生成的 安装文件包中包含.netframework组件;

点击"生成->生成解决方案",现在打开解决方案文件夹下的 debug 文件夹, 就可以看到生成的安装文件了.

第五章 总结和展望

本论文主要讨论了报表生成器的需求、设计和实现的过程。在毕设的初始阶 段我对报表都没有一个总体的概念,不知道报表究竟是做什么的,报表系统、报 表生成器和报表工具的区别有哪些。后来经过自己查资料,和老师同学讨论,逐 步对报表工具的结构、基本功能和作用有了初步的了解,清楚了自己要做的大概 是一个什么东西,大概应该有哪些功能。

开学后我收到了指导老师的任务书,经过对老师任务书的认真分析,我对报 表工具有了更深刻的认识,并对报表工具进行了需求分析、基本设计,对毕设的 进度时间安排进行了规划。让我对这次毕设有了一个总体的认识,清除了自己的 努力方向。

接着便是对报表工具的实现过程,自己动手后会发现一些问题,做了之后也 会对报表工具有更深的理解,所以回头看自己的需求分析和设计,会觉得有一些 需要改进的地方,经过自己的探索和老师的指导,我又加了一些改变了一些设计。

最后的报表工具分为报表设计器,报表引擎,底层数据存储,输出四个部分,其中报表设计器能够对报表进行灵活的绘制,排版;报表引擎能够对报表中的数据进行分析计算,分页,打印等处理;底层数据存储包括数据库存储和 XML 存储;输出包括 XML 格式,word 格式等。

当然本报表工具还有许多需要完善的地方,比如本报表工具设计的简洁易用,很多不常用的功能没有设计,所以本报表工具只适合中小型企业,不能胜任大型国企和政府的工作。

在表报设计器上,我的控件只有六种,而且都是矩形的,显然不能胜满足中 国的复杂报表要求。

在所有模块中,可以说报表引擎模块的发展空间是最大的,单拿我有的计算 模块而言,我的计算模块只有求和和排序等常见功能。而一个综合的报表系统是 需要复杂的数据计算的,这里还要用到编译原理的知识。而且本报表工具还有好 多引擎都没有,或是不完善。

在输出模块中,我们用的是现在常用的既 windows 自带的打印模式,已经基本上可以满足绝大多数用户的需求。我们还可以用 XML 进行输出,后续开发应该包括 Exce1、pdf 等多种输出方式。

在存储模块,我们用了 XML 和数据库进行对数据的存储,我们以后争取可以 直接读取 Excel 等文件中数据,还能把不同数据库中的数据交互使用。

综上所述,我做的报表工具应该被改写成一个功能齐全的,适合大型企业或 政府应用的报表系统。这个报表系统也是由报表设计器,报表引擎,报表服务器 ,报表客户端组成。服务器有大型公司的财务部负责管理,为每个部门分配可享 用的数据,每个部门可以在客户端从服务器端获取他需要的报表功能。也就是服 务器为他量身定做的报表工具。客户端根据自己的权限级别获取数据库中的数 据。

参考文献

- [1] 尤永涛,孙学波. 基于.Net组件技术的报表生成器的设计和实现 [J]. 辽宁科技大学,2008. 3.17
- [2] 财务报表分析 马丁·弗里德森,费尔南多·阿尔瓦雷斯著 = Financial statement analysisapractitioner's guide Martin Fridson, Fernando Alvarez 朱丽译 eng 北京 中国人民 大学出版社 2010
- [3] 陈新林,张双武.我国报表系统研究综述[J]. 计算机系统应用,2007.6第 12期
- [4] 财务报表分析 (美)利奥波德·A.伯恩斯坦(Leopold A.Bernstein),约翰·J.维欧德(John J. Wild)著 许秉岩,张海燕译 北京 北京大学出版社 2001
- [5] Michael McMillan 著; 吕秀锋, 崔睿 译. 数据结构与算法: C#语言描述[M]北京: 人民 邮电出版社, 2009.5.
- [6] .NET 3.5 programming : problem-design-solution / Chris Love Love, Chris. Indianapolis, IN : Wiley Pub., c2010.
- [7] System architecture with XML / Berthold Daum, Udo Merten Daum, Berthold, 1949-Amsterdam, [Netherlands] ; San Francisco, CA : Morgan Kaufmann Publishers, c2003.
- [8] Working with Microsoft Visual studio 2005 / Craig Skibo, Marc Young, Brian Johnson Skibo, Craig. Shanghai : Shanghai World Publishing Corporation, 2008.
- [9] Christian Nagel, Bill Evjen, Jay Glynn 等著; 李铭 翻译; 黄静 审校. C#高级编程(第6版)
 [M]. 北京:清华大学出版社, 2008.10
- [10] 清宏计算机工作室. XML 编程起步[M]. 北京: 机械工业出版社, 2002.
- [11] Even more excellent HTML with XML, XHTML, and Javascript / Timothy T. Gottleber, Timothy N. Trainor Gottleber, Timothy T. Boston, Mass. : McGraw-Hill, c2003
- [12] Professional Twitter development with examples in .NET 3.5 / Daniel Crenna , Daniel. Indianapolis, IN : Wiley, c2009.
- [13] 陈俊先,高阳..Net 软件设计新思维——像搭积木一样搭建软件 [M]. 北京: 电子工业 出版社,2009.09
- [14]Windows forms 编程实战 = Windows forms in action (美) Erik Brown 著 徐波等译 eng 布 朗 (Brown, Erik E.) 著 北京 机械工业出版社 2008
- [15] Head first C# = 深入浅出 C# / Andrew Stellman, Jennifer Greene Stellman, Andrew. 南京: 东南大学出版社, 2009.
- [16] Programming in C#.Net / Julia Case Bradley, Anita C. Millspaugh Bradley, Julia Case. Boston : McGraw-Hill, c2004.
- [17] XML.NET 编程指南 = XML.NET developer's guide (美) Adam Sills, Mesbah Ahmed
 等著 战晓苏等译 eng 西尔斯, A. (Sills, Adam) 著北京 电子工业出版社 2003

致 谢

本次毕业设计过程中得到了很多老师、同学的关心、指导和帮助。

首先感谢张坤龙老师,张老师在我准备考研初试和复试时给予了我充足的时间和鼓励。在之后的毕业设计过程张老师不但中给予在科研方面的宝贵指导和建议,还交给我一些做学问的道理,让我在专研时能够有一颗平静的心。

感谢王银鹤和胡俊同学,和我一起讨论毕设中遇到的问题,并能在关键的地 方给予我一些的建议和讲解。

感谢张祯同学给予我的鼓励,让我在最困惑的时候坚持走下去,没有半途而 废。

最后感谢所有帮助过我的人,我能过顺利完成毕设和你们的帮助、支持、关 心和鼓励是分不开的。