Winform 分页控件 使用说明书 V0.1

序号	修改人	修改日期	修改后版本	修改说明
1	伍华聪	2012-6-25	V0. 1	初稿

Ħ	킢
日	X

1.	引言.		2
1.1.	背景		2
1.2.	编写目	的	2
1.3.	参考资	そ料	2
1.4.	术语与	5缩写	3
2.	WINFC	DRM 分页控件	3
2.1.	控件特	产性	3
2.2.	使用效	文果介绍	3
2.3.	控件使	更用	8
	2.3.1.	添加到 Visual Studio 的工具箱	8
	2.3.2.	控件相关菜单	9
	2.3.3.	分页控件表头中文转义及显示字段控制	10
	2.3.4.	每页显示记录大小	11
	2.3.5.	表头全选操作	12
	2.3.6.	简单数据绑定	13
	2.3.7.	如何自定义数据列表的宽度。	15
	2.3.8.	如何实现多表关联的分页数据查询	17
	2.3.9.	如何实现附加语句查询	18
2.4.	分页控	2件使用注意事项	19

1. 引言

1.1.背景

在Winform程序开发中,分页是永恒的话题,因为需要显示的数据总是很多很多,分页 展示在程序性能和数据查看感官方面得到很好的平衡,是一种良好的编程习惯和UI设计。 Winform中的分页控件可能没有Asp.net世界中的分页控件那么丰富多彩,不过也有不少的 分页控件可以采用,各个人的可能都有一些不同的东西,一些好的东西。就我而言,我希望 控件能够尽可能的多一些功能,耦合性低一些,例如我不想是基于存储过程的,因为我很多 程序需要使用Access作为数据库,一般来说,我还希望有导出Excel数据的功能,还有打 印预览功能,由于我的数据源表头,如实体类集合、表格内容绑定的时候,表头是英文的, 我需要变为中文的,其他的功能有则更好。一句话:良好封装、功能丰富,使用简便。

基于这个思想,我在Winform 开发框架中引入一个Winform 分页控件模块,这个控件能 满足我绝大多数的分页应用及界面要求。该控件已经在我的共享软件中大量使用并得到升 华,如Winform 开发框架、WCF 开发框架、酒店管理系统、备件仓库管理系统、送水管理系 统软件、病人资料管理软件,以及其他为单位或个人定做的各种业务系统等等。

在软件开发过程中,为了节省开发时间,提高开发效率,统一用户处理界面,尽可能使 用成熟、功能强大的分页控件,这款 Winform 环境下的分页控件,集成了数据分页、内容提 示、数据打印、数据导出、表头中文转义等很多功能,由于集成性很好,省却很多功夫,专 注客户的业务及变化即可,否则一项表头的中文转换就够呛,还不说数据的分页,由于整合 性、一致性、稳定性等特点,客户使用感觉比较好。

1.2. 编写目的

本文档主要介绍《Winform 分页控件》的特性以及如何在 Winform 业务系统中进行使用。

1.3. 参考资料

序号	名称	版本/日期	来源

第 2 页 共 21 页

1	《Winform开发框架-架构设计说明书.doc》	内部
2		内部
3		内部
4		内部

1.4. 术语与缩写

1 在本文件中出现的"系统"一词,除非特别说明,均适用于《WCF开发框架》、《Winform 开发框架》。

2 在本文安装.NET框架中,除非特别说明,均指.NET 2.0框架。

2. Winform 分页控件

2.1. 控件特性

本 Winform 分页控件具有下面几个特点:

1)本分页控件主要是实现 Winform 下方便快捷、功能集成、高效友好的数据分页,分 页控件和具体的数据库无关,支持各种数据库的分页解决方案,如 Access、Sqlite、 SqlServer、MySql、Oracle 等等。

2) 支持文件复制或者对压缩包进行解压,适合更多琐碎程序集的整体升级。

这件集成除了基本的分页外,还有其他功能,如导出当前页、导出全部页、打印列表、以及相关功能操作的菜单(只要实现了相关的接口,则呈现相同的菜单)。

4)另外还有一些小地方,也是很常用关键的地方,就是间隔行的颜色变化设置,表头的中文化,表字段顺序可调整,行提示内容,是否显示行号、页面显示记录大小等基本界面功能。

5) 支持表头勾选操作,方便一些特殊界面处理,需要勾选而不是鼠标框选操作。

2.2. 使用效果介绍

1) 控件使用整体界面截图。

◎ 广州爱启迪技术有限公司-深田	日之星送水管理系统网络赢	[已注册]				
系统(S) 基本信息管理(B) 运行(信息(26) 辅助工具(27) 窗口	コピン 数据维护(型)	帮助(出)			
🗊 导航条 🌘 来电记录 🕘	送货记录 🚯 水票管理	📻 库存管理 📓	产品管理 ಿ 客户的	管理 🚰订单管理 📔	🗙 退出系统 商店 🕽	总店 🔹
工具窗口 早 X	来电记录 送货记录	水票管理 库存管理	P 产品管理 客户管	理 订单管理 操作人	、员登录记录 基本(i	i息管理
销售管理	产 日夕称	÷ 0 * 1				·
二 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一	/ #-31	广晶尖雪		97 规格型号	· /	m2 5
<u>一</u> 坐中间总自建	生产)間	▲ 😅 备注信息		查询	新建	
<u> </u>	() 但旦	立口米利 立	日夕物 辛日期	성 호민필드	进华谷	(建设价) 责任价
基本信息管理	\$75 {777770202-78			8 / 4925	7,0000	
2	{415E26B4-CF	承装 水	新建(图)		6,0000	8,0000 8,0000
1	{B1092865-44	尔水机	编辑选定项(图)	YR5-25L-21	77.0000	85,0000 85,000
权限管理	{68950E26-4E	尔水机	删除选定项 (2)	YR5-13L-01	35.0000	39.0000 39.0000
<u> </u>			刷新列表 (B)			
峰作品祭录は表			打印列表(2)			
球作员显示记录						
财务管理			共 4 条记录,每页 50	条,共1页 [[]]		号出当前贝」 【号出全部页
售后管理						.::
当前日期:2009年7月7日星期二 农历市	己丑[牛]年闰五月十五 小暑	当前用户:管理员([admin] 未接来电: 本:	地号码:61640502 当前そ	· 录商店: 总店	

2) 分页控件支持选定指定的列内容进行打印操作。

备件: 图	编码	备件名称 供货商		备件属类 ▼ 材质		-	备件类别 来 源	-	规格型号 库 位	
(使用1	立立	▼ 111注信思		所偶库房		•	所属部门	•	1	個
•	 編号 5280 5281 5282 5283 5285 5286 5287 5288 5289 5289 5289 	项目编号 1140101000010 1140101000017 1140101000021 1140101000023 1140101000088 114010100021 1140101000221 1140101000267 1140101000291	项目名利 打印設 内六角虫 待打 内六角虫 第 外六角虫 9 外六角虫 9	政 政 印印列 電台 電台 原目場合 原目指定存在 以留号 別格型号 別特所属 資子 原作 東京 自位 自伯 東京	2 全选 E	打印方式 ④ 全部行 适应页面宽度 页面抬头 广州爱启迪校表 备件信息报表	○ 法中行 ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	康 音	i什属类 准件(国产) 准件(国产) 准件(国产) 准件(国产) 准件(国产) 准件(国产) 准件(国产) 准件(国产) 准件(国产) 進件(国产) 進件(国产) 進件(国产) 進件(国产) 進件(国产) 進什(国产)	备件类别 日常备件 日常备件 日常备件 日常备件 日常备件 日常备件 日常备件 日常备件
	5290	1140101000319	♪♪六角螺 外六角螺	缸	Ψ.	确定	(0) 取消(C)	村	准件 (国产) 准件 (国产)	日常畜件
	5292	1140101000333	外六角螺栓	376			30*80 8.8%	村	准件 (国产)	日常备件
	5293	1140101000364	内六角螺栓	92			36*60 10.9级	标	准件(国产)	日常备件

3) 导出 Excel 是基本的功能,本控件支持当前页导出,全部页导出两种模式。导出的 Excel 数据也还是比较好看的,不是一般的格式哦。该分页控件整合了优秀的 Aspose.Cell 控件来进行 Excel 数据的导出,速度非常快,而且默认表头冻结,非常方便。

第4页共21页

6	0		□) ~ (? •) =		111	[兼容模]	स्ट) - Microsoft Excel				• ×	
	C	л	始插入页面	而布局 公式	数据	审阅	视图 加载项			۲	- • •	ĸ
		E	11 🔫 💽	f _x								¥
		Α	В	С	D	E	F	G	Н		J	
	1	编号	项目编号	项目名称	供货商	图号	规格型号	材质	备件属类	备件类别	单位	-
	5	5283	1140101000023	外六角螺栓			5*30 8.8级		标准件 (国产)	日常备件	ZHI	
	6	5284	1140101000068	外六角螺栓			10*15 8.8级		标准件(国产)	日常备件	ZHI	
	7	5285	1140101000171	外六角螺栓			16*210 8.8级		标准件(国产)	日常备件	ZHI	
	8	5286	1140101000221	外六角螺栓			20*180 8.8级		标准件 (国产)	日常备件	ZHI	
1	9	5287	1140101000226	外六角螺栓			20*220 8.8级		标准件 (国产)	日常备件	ZHI	
1	10	5288	1140101000267	外六角螺栓			24*35 8.8级		标准件 (国产)	日常备件	ZHI	
1	11	5289	1140101000291	外六角螺栓			24*210 8.8级		标准件 (国产)	日常备件	ZHI	
1	12	5290	1140101000319	外六角螺栓			27*170 8.8级		标准件 (国产)	日常备件	ZHI	
1	13	5291	1140101000325	外六角螺栓			30*40 8.8级		标准件 (国产)	日常备件	ZHI	
1	1/1	5292	1140101000333	<u> </u>			30*80 8.843		标准件 (国产)	日堂备件	7HI	*
Ľ		► • •	Sheetl 💋									
Ľ	就第	ā								9	•	

4) 报表打印界面。控件一项功能,也是集实用功能之所成,打印当前列表内容,如下图所示,该内容会保存用户在每个列表数据中的信息,打印不同的表头内容,如下图所示。总体该功能上就是我们一般报表所需要的功能。其中报表打印预览可以设置报表标题,打印的列也可以设定,有一些字段的汇总功能,而且这样的报表基本上不需要额外的代码就能实现(相对分页控件来说)。

	打印預	览												X
8	• ۹			关闭 (C)									页创	1 🗘
														-
							某某有	限公司-	- 收银明细道	间报表			2	200
	房间号	宾客编号	宾客姓名	收款时间	收款方式	现金	储值卡	代金劵	国内信用卡	国际信用卡	收银类别	操作员	备注信息	
	BD102	2	小邱	2009-12-14 19:15:35	现金	500.0000	0.0000	0.0000	0.0000	0.0000	押金收银	管理员	入住登记时交付押金	
	BD102	2	小邱	2009-12-14 19:16:19	现金	- 298. 5000	0.0000	0.0000	0.0000	0.0000	结账收银	管理员	[结账时未退押金]已结财 号:ZD20091214001	<u>ж</u>
	合计:					201.5000	0.0000	0.0000	0.0000	0.0000				
														~
<							1111							>

打印预览									X
i 🗋 🗳	宿放(Z) 🔻	4 1 /6	9 🕨 🔰 🗙 关	闭(C)					
									^
			广	州爱启迪技术	有限公司一	当前庠存査	询统计报表011年	7月3日10:45	:
	编号	项目编号	项目名称	单价	库存量	库存金额	备件属类	备件类别	
	6576	1140203000144	内六角螺栓	25. 0000	7	175. 0000	标准件(国产)	日常备件	
	6577	1140203000148	内六角螺栓	25. 0000	48	1200. 0000	标准件 (国产)	日常备件	
	6578	1140203000170	内六角螺栓	25. 0000	7	175.0000	标准件 (国产)	日常备件	
	6579	1140203000188	内六角螺栓	25. 0000	200	5000. 0000	标准件 (国产)	日常备件	
	6580	1140203000172	内六角螺栓	25.0000	337	8425.0000	标准件 (国产)	日常备件	
	6581	1140203000173	内六角螺栓	25.0000	319	7975. 0000	标准件 (国产)	日常备件	
	6582	1140203000175	内六角螺栓	3. 2300	135	436. 0500	标准件 (国产)	日常备件	
	6583	1140203000176	内六角螺栓	25. 0000	102	2550. 0000	标准件 (国产)	日常备件	
			1. N. A. 1=14	1		1	1-000 - 1-000		•

5)列内容自动提示。由于我们设置了中文表头,另外一项便利的功能就是,当鼠标停放在 某一行的时候,出现改行内容的信息提示,这样可以方便用户了解一些详细的信息, 如下所示。

	编号	客户编号	客户名称	客户类型	客户地区	客户单位	客户地址	电话1
	879e4aeb-bd0	B146	测试146	机关客户	武侯区			
	f0c86052-80e	A72	小邱	家庭客户	武侯区			1375187
	{C3E0012D-E8	A2	伍华聪	单位客户	锦江区	某公司	广州市五山路	
•	{C3E0012D-E8	A1	伍华聪	家庭客户	青羊区	某公司	广州市五山路	1357027
	18649881-3f9	数据基本信息: 編号:{[3][10120-8842-4677-9	880-185002878334}	侯区			
	f91a7782-bbb	客户编号:		000 11000211200Aj	侯区			
	39700a5e-342	客尸名称:1 客户类型:3	立毕聪 家庭客户		侯区			
	Зf31c99c-03Ъ	客户地区: ³ 家白魚位・1	青羊区 第八司		侯区			
		客户地址:) 电话1:135 电话3: 电话3: 电话3: 电话5:616 开户日期: 分店DI:{1 备注: 更新日期:;	¹¹ 升市五山路 70271034 40502 2008-04-19 22:24: 5F7E83D-D824-4e2b 2009-03-22 9:18:5	24 -8CB1-49ZD6B0B148 2	¥F}			
<		ш						>
			共 8 条记录,	毎页 50 条,共 1			导出当前页 导出	全部页

6) 支持表头全选操作。表头全选操作只需要一行代码即可实现。

this.winGridViewPager1.ShowCheckBox = true;

客户	白名称			客户类	型		▼ 客.	户地区		-	联系电话		
客户	编号			客户地	tut 🗌 🗌		容	户单位			备注信息		
开户	日期	<mark>2011</mark> 年	=12月30日	■▼ 至 201	1年12月30日		闺用	获取勾选数量	查询		新建	显示未	订货客户
		V	客户编号	客户名称	客户类型	客户地区	客户单位	客户地址	电话 1	电话 <mark>2</mark>	电话3	电话4	电话5
1		V	A2	伍华聪	单位客户	锦江区	某公司	广州市五	13570271	61640502			
2		1	A1	伍华聪	家庭客户	青羊区	广东省广	广东省广	13570271				6164050
3		1	B148	测试148	机关客户	武侯区	某公司	广州市五					
4		V	B147	测试147	机关客户	武侯区							
5		1	B146	测试146	机关客户	武侯区							
6		1	B108	测试108	机关客户	武侯区							
7		1	B107	测试107	机关客户	武侯区							
8		1	B106	测试106	机关客户	武侯区							
9		1	B105	测试105	机关客户	武侯区							
10		V	B104	测试104	机关客户	武侯区							
10		¥	8104	<u> </u> 测试104	机天备尸	武侠区							Þ

7) 支持基于 dotNetbar 控件界面的分页。由于本人随笔《Winform 开发框架之 Office Ribbon 界面》引入了基于 DotNetbar 的 Winform 开发框架,因此在此基础上利用 DotNetBar 的界面效果特性,修改了原来的分页控件,提供基于 DotNetbar 控件效果的分页控件支持,效果如下所示。

🦉 客户管理 🗕 🗖 🗙										
客,	P:	2称	客户类	型	▶ 客户地		▶ 联系	民电话		
客,	户翁	扁号	客户地	址	客户单	自位	备社	注信息		
ŦЯ	⇒E	期 2011-08-26	⊻ 至 2011-	-08-26	自用		查询 新	建 显示未	订货客户	
	4	客户编号	客户名称	客户类型	客户地区	客户单位	客户地址	电话1	电话2	
	1	A2	伍华聪	单位客户	锦江区	某公司	广州市五山路	13570271034	61640502	
	2	A1	伍华聪	家庭客户	青羊区	某公司	广州市五山路	13570271034		
Þ	3	B148	测试148	机关客户	武侯区	某公司	广州市五山路			_
	4	B147	测试147	机关客户	武侯区					
	5	B146	测试146	机关客户	武侯区					
	6	B108	测试108	机关客户	武侯区					
	7	B107	测试107	机关客户	武侯区					
	8	B106	测试106	机关客户	武侯区					
	9	B105	测试105	机关客户	武侯区					-
•		-								
		共 71 条记录,每	页 3O 条,共 3 页	τ I<		> 导出当前:	页 导出全部页			

8) 分页控件除了支持传统效果、DotNetBar 效果外,还是支持目前应用很广泛的 DevExpress 界面控件,这几种模式的控件使用属性及方法 99%以上是一致的。

客户	名称		客户	类型		▶ 客	户地区		*	联系电话		
客户	编号		客户提	也址		酒	F户单位			备注信息		
开户	日期 2011	-08-26	⊻ 至 20	11-08-26	<u> </u>	启用		查	询	新建	显示未证	丁货客户
_	客户编号	客户名称	客户类型	客户地区	客户单位	客户地址	电话1	电话2	电话3	电话4	电话5	开户日期
1	A2	伍华聪	单位客户	锦江区	某公司	广州市五	13570271	61640502				2008-04
2	A1	伍华聪	家庭客户	青羊区	某公司	广州市五	13570271				61640502	2008-04
З	B148	测试148	机关客户	武侯区	某公司	广州市五						2008-08
4	B147	测试147	机关客户	武侯区								2008-08
5	B146	测试146	机关客户	武侯区								2008-08
6	B108	测试108	机关客户	武侯区								2008-08
										_		•

2.3. 控件使用

2.3.1. 添加到 Visual Studio 的工具箱

 在 Visual Studio 开发环境的工具箱中,添加一个分页控件(可以其他名称)的项目, 然后选择 WHC.Pager.WinControl.dll(DevExpress 版本的为 WHC.Pager.WinControlDx.dll)文件,导入分页控件的工具箱图标,如下所示。

其中 WinGridViewPager 和 WinGridView 两个控件就是我们常用到的分页控件,两者用 法几乎一致,界面效果不同在于 WinGridViewPager 具有分页工具条,而 WinGridView 是提 供显示所有内容,没有工具条。

工具箱
᠃ WHC.WareHouseMis.DotUI 组件
⊞ 所有 Windows 窗体
⊞ 公共控件
団 容器
⊞ 菜单和工具栏
団 数据
∃ 组件
⊞ 打印
∃ 对话框
⊞ 报表
• Visual Basic PowerPacks
□ 分页控件
指针
🏶 WinGridViewPager
🎲 WinGridView
🐺 WinListViewPager
🎲 Pager

2) 拖动到指定的窗体中。

3) 在代码引用相关的代码实现动态调用。

2.3.2. 控件相关菜单

在窗体加载实现中添加分页控件的实现代码,以 On 开始的是相关操作的实现事件,如 OnPageChanged 表示分页控件页面发生变化的时候,需要实现的事件处理(这个是必须的), 另外 AppendedMenu 是可以在分页控件自带菜单上增加的菜单,如下所示。注意,只要实现 相关的事件处理,那么对应的上下文菜单将会出现,默认几个标准的上下文菜单如下所示, 包含新建、编辑选定项、删除选定项、打印列表、刷新列表菜单。

注意,为了使分页控件能够显示总数,并记住当前的分页,那么在 OnPageChanged 实现中需要修改分页控件的这两个属性。

编号	项目编号		项目名称	当前库在
4838	YG12204010)093	0型圈	16
4837	YG12204010	092	0-刑閥	22
4836	¥G122	±12±10	n	
4835	YG122	新建(1)	り (字)面(に)	22
4834	YG1 <mark>2</mark> 2			6
4833	YG1 <mark>2</mark> 2	刷新列	(E) 表(R)	20
4832	YG1 <mark>2</mark> 2		±(n)	3
4831	YG122	打印列	表(P)	50
4830	YG1220400	1085	1127.9858	5

默认菜单需要实现相关的事件,才会显示出来,如新建菜单,需要实现事件 OnAddNew 的事件处理,"编辑选定项"菜单需要实现 OnEditSelected 事件处理,删除菜单需要实现 OnDeleteSelected 事件处理。

C#代码
<pre>this.winGridViewPager1.OnAddNew += new EventHandler(winGridViewPager1_OnAddNew);</pre>
<pre>this.winGridViewPager1.OnEditSelected += new EventHandler(winGridViewPager1_OnEditSelected);</pre>
<pre>this.winGridViewPager1.OnDeleteSelected += new EventHandler(winGridViewPager1_OnDeleteSelect</pre>
ed);

如果需要添加自己的菜单,则指定 AppendedMenu 对象即可。

客户类型	客户	地区	客户单位	客户地址	电话
机关客户	新建	订单(O)			
家庭客户	购买	水票(T)			51
单位客户		±)7些寿			[
家庭客户	室白:	不り 5日 开白时间	r/~(r) 1		1 0
机关客户		/1/ #JIH	,		
机关客户	新建((N)	-		
机关客户	编辑	选定坝(E	:)		
机关客户	売 売 開除)	选定项(D))		
	刷新	列表(R)			
	打印	列表(P)			

标准的分页控件初始化代码如下所示:

C#代码

```
public FrmCustomer()
 {
 InitializeComponent();
 if (!this.DesignMode)
 {
 this.winGridViewPager1_OnPageChanged += new EventHandler(winGridViewPager1_OnPageChanged);
 this.winGridViewPager1_OnStartExport += new EventHandler(winGridViewPager1_OnStartExport);
this.winGridViewPager1.OnEditSelected += new EventHandler(winGridViewPager1 OnEditSelected);
 this.winGridViewPager1.OnDeleteSelected +=
    new EventHandler(winGridViewPager1_OnDeleteSelected);
    this.winGridViewPager1.OnRefresh += new EventHandler(winGridViewPager1_OnRefresh);
     this.winGridViewPager1.OnAddNew += new EventHandler(winGridViewPager1_OnAddNew);
     this.winGridViewPager1.AppendedMenu = this.contextMenuStrip1;
     this.winGridViewPager1.ShowLineNumber = true;//显示行号
     this.winGridViewPager1.PagerInfo.PageSize = 30;//页面大小
     this.winGridViewPager1.EventRowBackColor = Color.LightCyan;//间隔颜色
     this.winGridViewPager1.ShowCheckBox = true;//全选操作
 }
    }
```

2.3.3. 分页控件表头中文转义及显示字段控制

控件表头的中文转义是通过函数 AddColumnAlias 为字段添加别名进行转义的,如下所示。

C#代码
private void BindData()
{
#region 添加别名解析
this.winGridViewPager1.AddColumnAlias("ID", "编号");
this.winGridViewPager1.AddColumnAlias("Number", "客户编号");
this.winGridViewPager1.AddColumnAlias("Name", "客户名称");
this.winGridViewPager1.AddColumnAlias("Type", "客户类型");
this.winGridViewPager1.AddColumnAlias("Area", "客户地区");
this.winGridViewPager1.AddColumnAlias("Company", "客户单位");
this.winGridViewPager1.AddColumnAlias("Address", "客户地址");
this.winGridViewPager1.AddColumnAlias("Telephone1", "电话1");
this.winGridViewPager1.AddColumnAlias("Telephone2", "电话 2");
this.winGridViewPager1.AddColumnAlias("Telephone3", "电话 3");
this.winGridViewPager1.AddColumnAlias("Telephone4", "电话 4");
this.winGridViewPager1.AddColumnAlias("Telephone5", "电话 5");
this.winGridViewPager1.AddColumnAlias("CreateDate", "开户日期");
this.winGridViewPager1.AddColumnAlias("Shop_ID", "分店 ID");
this.winGridViewPager1.AddColumnAlias("Note", "备注");
this.winGridViewPager1.AddColumnAlias("LastUpdated", "更新日期");
#endregion
<pre>string where = GetSearchSql();</pre>
this.winGridViewPagerl.DataSource = BLLFactory <customer>.Instance.Find(where, this.winGr</customer>
<pre>idViewPager1.PagerInto);</pre>
<pre>this.winGridViewPager1.dataGridView1.Refresh();</pre>
}

上面默认的操作时显示数据源里面所有的字段信息,如果需要控制显示的字段内容及顺

序,那么使用 DisplayColumns 属性即可。这个属性指定字段的顺序就是控件表头字段的显示顺序。

C#代码

this.winGridViewPager1.DisplayColumns =

"ID,ItemNo,ItemName,Manufacture,MapNo,Specification,StockQuantity,AlarmQuantity,WareHouse"; 控件报表标题如下所示。

C#代码

this.winGridViewPager1.PrintTitle = Portal.gc.gAppUnit + " -- " + "备件信息报表";

2.3.4. 每页显示记录大小

分页控件可以通过属性设置每页显示多少条记录,如下代码所示:

第 11页 共 21页

C#代码 this.winGridViewPager1.PagerInfo.PageSize = 30;//页面大小 this.winGridViewPager1.ShowLineNumber = true;//显示行号 this.winGridViewPager1.EventRowBackColor = Color.LightCyan;//间隔颜色

2.3.5. 表头全选操作

表头全选操作,只需要设置一个属性 ShowCheckBox=True 即可显示。各个版本(传统、DotNetBar、DevExpress 版本均一致)。

C#代码

this.winGridViewPager1.ShowCheckBox = true;



这样的简单调用就能实现控件表头全选的操作,不需要另外其他额外的操作,是不是很 好,简洁呢?这其实就是我追求的分页控件表头全选的最优方法了。

最后,您可能还有一个疑问,就是我们全选或者部分选择,我要知道我选择的是那些行, 该怎么操作呢?放心,我这个分页控件也已经为你考虑了,呵呵。

通过调用下面的代码即实现。



2.3.6. 简单数据绑定

使用简单操作加载数据方式绑定分页控件,而不是基于我 Database2Sharp 自动生成的整个 EnterpriseLibary 架构的实现方式。

C#代码

{

private void BindData()

```
#region 添加别名解析
```

```
this.winGridViewPager1.AddColumnAlias("ID", "编号");
this.winGridViewPager1.AddColumnAlias("Number", "客户编号");
this.winGridViewPager1.AddColumnAlias("Name", "客户名称");
this.winGridViewPager1.AddColumnAlias("Type", "客户类型");
this.winGridViewPager1.AddColumnAlias("Area", "客户地区");
this.winGridViewPager1.AddColumnAlias("Company", "客户单位");
this.winGridViewPager1.AddColumnAlias("Address", "客户地址");
this.winGridViewPager1.AddColumnAlias("Telephone1", "电话 1");
this.winGridViewPager1.AddColumnAlias("Telephone2", "电话 2");
this.winGridViewPager1.AddColumnAlias("Telephone3", "电话 3");
this.winGridViewPager1.AddColumnAlias("Telephone4", "电话 4");
this.winGridViewPager1.AddColumnAlias("Telephone5", "电话 5");
this.winGridViewPager1.AddColumnAlias("CreateDate", "开户日期");
this.winGridViewPager1.AddColumnAlias("Shop_ID", "分店 ID");
this.winGridViewPager1.AddColumnAlias("Note", "备注");
this.winGridViewPager1.AddColumnAlias("LastUpdated", "更新日期");
#endregion
```

```
string where = GetSearchSql();
if (this.radNormal.Checked) //采用框架加载
{
this.winGridViewPager1.DataSource =
BLLFactory<Customer>.Instance.Find(where, this.winGridViewPager1.PagerInfo);
```

} else {

```
//直接加载数据
```

```
DataTable dt = DirectLoadData(where, this.winGridViewPager1.PagerInfo);
this.winGridViewPager1.DataSource = dt.DefaultView;
```

}

this.winGridViewPager1.dataGridView1.Refresh();

C#代码

```
/// <summary>
/// 通过自己组装分页语句
/// </summary>
private DataTable DirectLoadData(string where, PagerInfo pagerInfo)
{
     DataTable dt = null;
     PagerHelper
                   helper
                                         PagerHelper("All_Customer",
                                                                         "*",
                                                                                "ID",
                                                                                         pagerInfo.PageSize,
                             =
                                  new
pagerInfo.CurrenetPageIndex, true, where);
    string countSql = helper.GetPagingSql(DatabaseType.SqlServer, true);
    string dataSql = helper.GetPagingSql(DatabaseType.SqlServer, false);
    string value = SqlValueList(countSql);
    pagerInfo.RecordCount = Convert.ToInt32(value);
    dt = SqlTable(dataSql);
    return dt;
}
```

2.3.7. 如何自定义数据列表的宽度。

传统界面和 DotNetBar 控件的,通过 dataGridView1.DataBindingComplete 的事件处理,实现 自定义宽度设置;而 DevExpress 控件样式的分页控件,则通过 DataSourceChanged 事件进行 处理。

```
C#代码
public partial class FrmItemDetail : BaseDock
{
    public FrmItemDetail()
    {
        InitializeComponent();
        InitDictItem();
        this.winGridViewPager1.OnPageChanged +=
                        new EventHandler(winGridViewPager1_OnPageChanged);
        this.winGridViewPager1_OnStartExport += new EventHandler(winGridViewPager1_OnStartExport);
        this.winGridViewPager1.OnEditSelected += new EventHandler(winGridViewPager1_OnEditSelected);
        this.winGridViewPager1.OnAddNew += new EventHandler(winGridViewPager1_OnAddNew);
        this.winGridViewPager1.OnDeleteSelected +=
                        new EventHandler(winGridViewPager1_OnDeleteSelected);
        this.winGridViewPager1_OnRefresh += new EventHandler(winGridViewPager1_OnRefresh);
        this.winGridViewPager1.AppendedMenu = this.contextMenuStrip1;
        this.winGridViewPager1.ShowLineNumber = true;
        this.winGridViewPager1.dataGridView1.DataBindingComplete +=
                   new DataGridViewBindingCompleteEventHandler(dataGridView1_DataBindingComplete);
    }
    /// <summary>
    /// 绑定数据后,分配各列的宽度
    /// </summary>
    void dataGridView1_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)
    {
        if (this.winGridViewPager1.dataGridView1.Columns.Count > 0)
        {
             this.winGridViewPager1.dataGridView1.Columns["ID"].Width = 100;
             this.winGridViewPager1.dataGridView1.Columns["ItemNo"].Width = 100;
             this.winGridViewPager1.dataGridView1.Columns["ItemName"].Width = 100;
             this.winGridViewPager1.dataGridView1.Columns["Manufacture"].Width = 100;
             this.winGridViewPager1.dataGridView1.Columns["MapNo"].Width = 100;
             this.winGridViewPager1.dataGridView1.Columns["Specification"].Width = 100;
        }
    }
```

如果是 DevExpress 控件样式的的分页控件,要实现自定义宽度,需要下面的操作。



2.3.8. 如何实现多表关联的分页数据查询

PagerHelper 是分页控件里面内置的一个用于生成多种数据库不同的分页语句的类库。

```
C#代码
public DataTable SearchParkUser(string condition, PagerInfo pagerInfo)
{
     DataTable dt = null;
     string selectSql = string.Format(@"select u.*, p.current_space_number, to_char(p.sample_time, 'yyyy-mm-dd
hh24:mi:ss') as sample_time from device_user u
    left join device_parking p on u.park_id = p.park_id order by u.id desc ");
     condition += string.Format(" AND status=0 ");
    PagerHelper helper = new PagerHelper(selectSql, "*", "id",
                      pagerInfo.PageSize, pagerInfo.CurrenetPageIndex, true, condition);
     string countSql = helper.GetPagingSql(DatabaseType.Oracle, true);
     string dataSql = helper.GetPagingSql(DatabaseType.Oracle, false);
     string value = SqlValueList(countSql);
     pagerInfo.RecordCount = Convert.ToInt32(value);
     dt = SqlTable(dataSql);
     return dt;
```

2.3.9. 如何实现附加语句查询

SearchCondition 控件是公用类库里面的对象,功能是快速构造相关的查询语句,支持多种数据库语句生成。

```
C#代码
private string GetSearchSql()
{
    SearchCondition condition = new SearchCondition();
    condition.AddCondition("COMPANY_CODE", this.txtCompanyCode.Text, SqlOperator.Like)
        .AddCondition("COMPANY_NAME", this.txtCompanyName.Text, SqlOperator.Like)
        .AddCondition("LICENSE", this.txtLicense.Text, SqlOperator.Like)
        .AddCondition("PARK_NAME", this.txtParkName.Text, SqlOperator.Like)
        .AddCondition("PARK_ADDR", this.txtParkAddr.Text, SqlOperator.Like)
        .AddCondition("STATUS", this.txtStatus.Text, SqlOperator.Equal)
        .AddCondition("AREA_COUNTRY", this.txtAreaCountry.Text, SqlOperator.Like);
    string where = condition.BuildConditionSql(DatabaseType.Oracle);
    if (this.chkSpaceUp.Checked)
    {
        where += " AND CURRENT_SPACE_NUMBER >= PARK_SPACE_NUMBER";
    }
    if (this.chkLatLong.Checked)
    {
        where += "AND (LATITUDE is null or LATITUDE=0)";
    }
    if (chkSpaceNumber.Checked)
    {
        where += " AND CURRENT_SPACE_NUMBER is not null ";
    }
    if (this.chkHourOnline.Checked)
    {
        where += "AND SPACE_LAST_UPDATE >= sysdate - 1/24";
    }
    if (this.chkContact.Checked)
    {
        where += " AND (CONTACT is null OR CONTACT_TELEPHONE is null)";
    }
    return where.Replace("Where", "");
```

2.4. 分页控件使用注意事项

特别说明:如果在 VS2010 中打开含有分页控件的窗体,出现错误,请大家使用 DotNetReactor 混淆的时候,不要勾选 Anti ILDASM 项目,或者不要使用混淆也可以。

.NET Reactor v4.5.0.0 - Full Ver	sion [yoza[UpK]] *NEW*	- D (X)
File Actions Tools Pr	otection Presets Help	
Tasks Image: Constraint of the system Image: Constrait of the system	Files Settings License Manager .NET Inspector Main Assembly Image: Comparison of the set of the	Open Add Remove See
	To protect more than one assembly at the same time add further assemblies to a	this list.
NecroBit is a powerful protection technology to stop any decompiling.	Quick Settings Q NecroBit Q Anti ILDASM Q Anti ILDASM Q Anti Tampering Q String Encryption Q Control Flow Obfuscation	le pt Resources

在实际项目使用过程中,发现 VS2010 和 VS2008 不同,如果不能 ILDASM,那么会出现异常,正常打包项目好像都会受影响的,不能刷新 dll 的依赖项,因此对于一些项目控件,尽量不用勾选上面的那两个 Anti项目。