

开发框架- 系统功能介绍白皮书

V3.0

目 录

| | | |
|-----------|--------------------------------------|----------|
| 1. | 引言 | 2 |
| 1.1. | 背景..... | 2 |
| 1.2. | 编写目的..... | 2 |
| 1.3. | 参考资料..... | 2 |
| 1.4. | 术语与缩写 | 3 |
| 2. | 开发框架的重要特性总结 | 3 |
| 2.1.1. | 支持插件化应用开发，能够动态配置及加载开发的插件..... | 4 |
| 2.1.2. | 菜单支持动态配置，可根据用户权限进行动态控制显示..... | 6 |
| 2.1.3. | 集成各种独立开发好的模块..... | 7 |
| 2.1.4. | 可重复使用的系统基础模块..... | 13 |
| 2.1.5. | 代码生成工具 Database2Sharp 的整合 | 16 |
| 2.1.6. | 通用的 Excel 数据导入导出..... | 18 |
| 2.1.7. | 自定义 Excel 自定义模板报表的生成 | 20 |
| 2.1.8. | 强大通用的统计图表模块，数据统计更方便..... | 22 |
| 2.1.9. | 专业美观 RDLC 设计报表，方便打印浏览和 Excel 导出..... | 27 |
| 2.1.10. | 参数配置管理，可以配置系统的各种运行参数 | 29 |
| 2.1.11. | 基于多数据库的数据查询模块和通用高级查询模块 | 31 |
| 2.1.12. | 框架提供基于多种数据库的整合 | 35 |
| 2.1.13. | 框架界面基类也进行统一封装，使用更方便，效果更统一..... | 36 |
| 2.1.14. | 框架提供基础性的支持 | 37 |
| 2.1.15. | Web API 接口框架的支持 | 38 |
| 2.1.16. | 特性总结 | 42 |

1. 引言

1.1. 背景

《Winform 开发框架》、《WCF 开发框架》和《混合式开发框架》三个框架是我们经过多年的项目积累，吸收众多框架产品客户的宝贵意见，反复提炼优化而成的。

《Winform 开发框架》用于传统的数据库通讯获取数据，这种方式获取数据，方便快捷，可以用于常规的业务系统的场景，用于**单机版软件**或者基于**局域网内**的业务系统软件。

《WCF 开发框架》采用了 WCF 技术的分布式开发模式，系统能够通过远程的 WCF 服务获取数据，而不用直接和数据库相连，提高数据的安全性和可维护性，适用于**互联网**、**局域网**环境下的业务系统的搭建，是一种稳定、安全的框架应用。

Web API 是一种应用接口框架，它能够构建 HTTP 服务以支撑更广泛的客户端（包括浏览器，手机和平板电脑等移动设备）的框架，**ASP.NET Web API** 是一种用于在 .NET Framework 上构建 RESTful 应用程序的理想平台。

《混合式开发框架》是指混合了传统《Winform 开发框架》、《WCF 开发框架》和 Web API 接口框架的特点，可以在三者之间自由切换，统一了系统界面层对业务服务的调用模式，所有组件模块均实现三种方式的调用，是一种弹性化非常好的框架应用，既可用于**单机版软件**或者基于**局域网内**的应用软件，也可以用于分布式技术的**互联网环境**应用。

1.2. 编写目的

本文档主要介绍《Winform 开发框架》、《WCF 开发框架》和《混合式开发框架》中系统的功能特性，这三种框架除了它们获取数据的方式不同外，他们具体大多数相同的特性，本文档主要着重介绍这三个基于插件化应用框架的各种相同特性。

1.3. 参考资料

| 序号 | 名称 | 版本/日期 | 来源 |
|----|-----------------------------|-------|----|
| 1 | 《Winform 开发框架-架构设计说明书.doc》 | | 内部 |
| 2 | 《Winform 开发框架-数据库设计说明书.doc》 | | 内部 |

| | | | |
|---|--------------------------|--|----|
| 3 | 《WCF 开发框架-系统部署使用说明书.doc》 | | 内部 |
| 4 | 《开发框架-系统数据库还原操作说明.doc》 | | 内部 |
| 5 | 《混合式开发框架主体框架项目说明.doc》 | | 内部 |
| 6 | 《权限管理系统功能介绍白皮书.doc》 | | 内部 |

1.4. 术语与缩写

- 1 在本文件中出现的“开发框架”一词，除非特别说明，适用于《Winform开发框架》、《WCF开发框架》和《混合式开发框架》。
- 2 在本文安装.NET框架中，除非特别说明，均指.NET 4.0框架。
- 3 RDLC，它的全称是Report Definition Language Client-Side，原来是微软基于SQL Server 报表服务中经提供了一种被称为报表定义语言（Report Definition Language, RDL）的改进版本。

2. 开发框架的重要特性总结

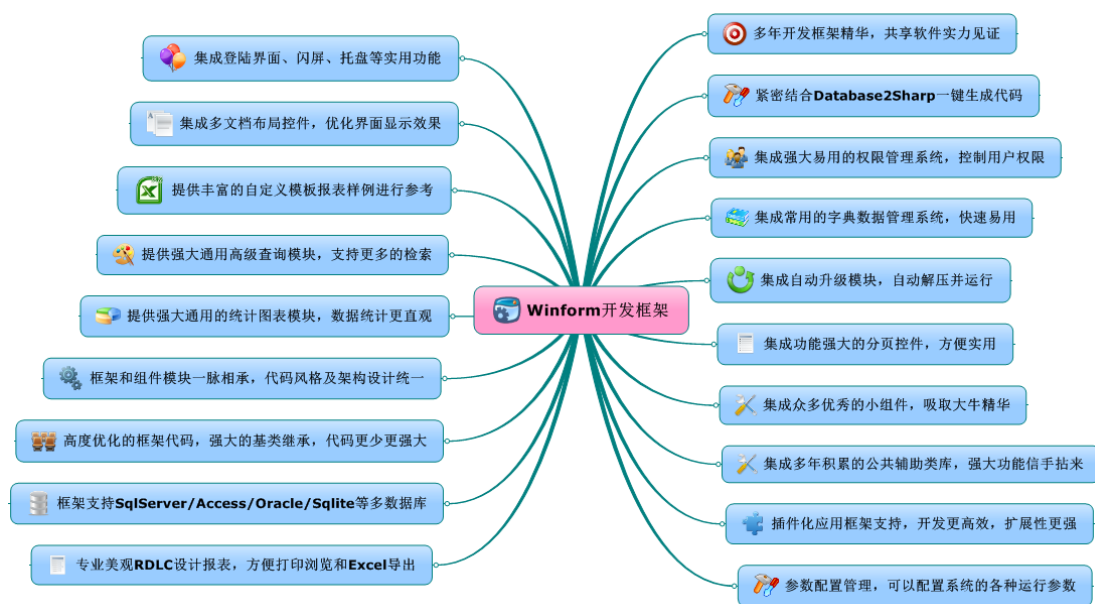
从事 Winform 开发框架的研究和推广，也做了好几个年头了，从最初的项目雏形到目前各种重要特性的加入完善，是经过了很多项目的总结归纳和升华，有些则是根据客户需要或者应用前景的需要进行的完善，整个 Winform 开发框架具有很好的通用性和借鉴性，本文从该 Winform 开发框架进行概括总结，力求把各个重要的特性进行一些详细的说明，使大家了解整个 Winform 开发框架的面貌和特性。

Winform 开发框架总体性概括有：支持插件化应用开发，能够动态配置及加载开发的插件；菜单支持动态配置，可根据用户权限进行动态控制显示；集成各种独立开发好的模块，如权限管理系统、字典管理、附件管理、通讯录管理、人员管理模块等，自动更新模块等，实现系统的权限控制、字典管理、软件支持远程自动更新等功能；可重复使用的系统基础模块，Winform 分页控件和.NET 开发公用类库等；整个框架通过与代码生成工具 Database2Sharp 进行配合，能够一键生成整体性框架代码，Winform 系统界面代码，开发更高效。

整个 Winform 开发框架，提供了很多常见业务系统提供的功能，包括 Excel 数据导入导出，能快速导入自定义模块的 Excel 数据，快速导出列表数据；自定义 Excel 自定义模板报

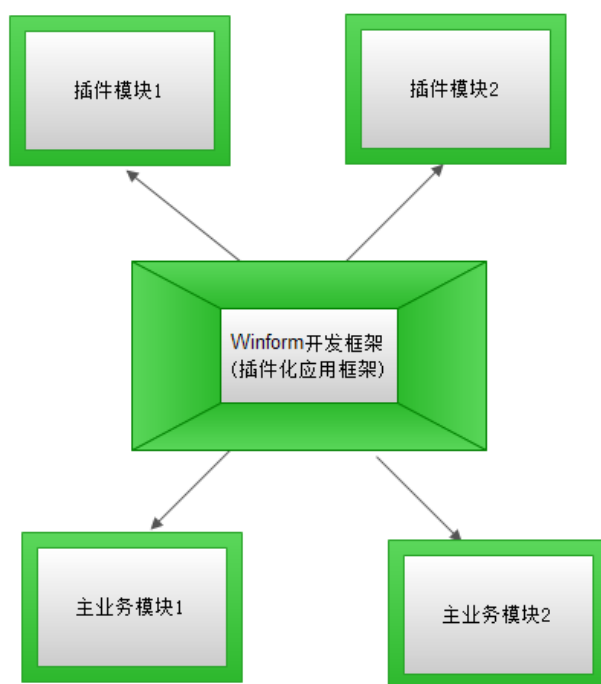
表的生成，生成各种复杂的 Excel 报表；强大通用的统计图表模块，数据统计更方便；基于 RDLC 的报表模块，能制作出美观专业的报表；基于多数据的数据查询模块和通用高级查询模块，查询数据更方便；框架提供基于多种数据库（Sqlserver/Oracle/MySql/SQLite /Access/国产达梦数据库)的整合，能够手动配置数据库类型，通用模块内置支持多种数据库切换；框架提供了高度封装的数据访问基类，开发代码更少更高效；框架界面基类也进行统一封装，使用更方便，效果更统一；框架提供基础性的支持，包括集成登陆、闪屏、托盘功能，以及为插件模块提供登陆用户信息和系统信息。

Winform 开发框架的主要功能概览如下图所示。



2.1.1. 支持插件化应用开发，能够动态配置及加载开发的插件

框架实现基于插件化的应用，如下所示。



支持插件化应用的开发框架能给程序带来无穷的生命力，也是目前很多系统、程序追求的重要方向之一，插件化的模块，在遵循一定的接口标准的基础上，可以实现快速集成，也就是所谓的热插拔操作，可以无限对已经开发好系统进行扩展，而且不会影响已有的功能，不再需要的模块，通过修改配置移除即可。

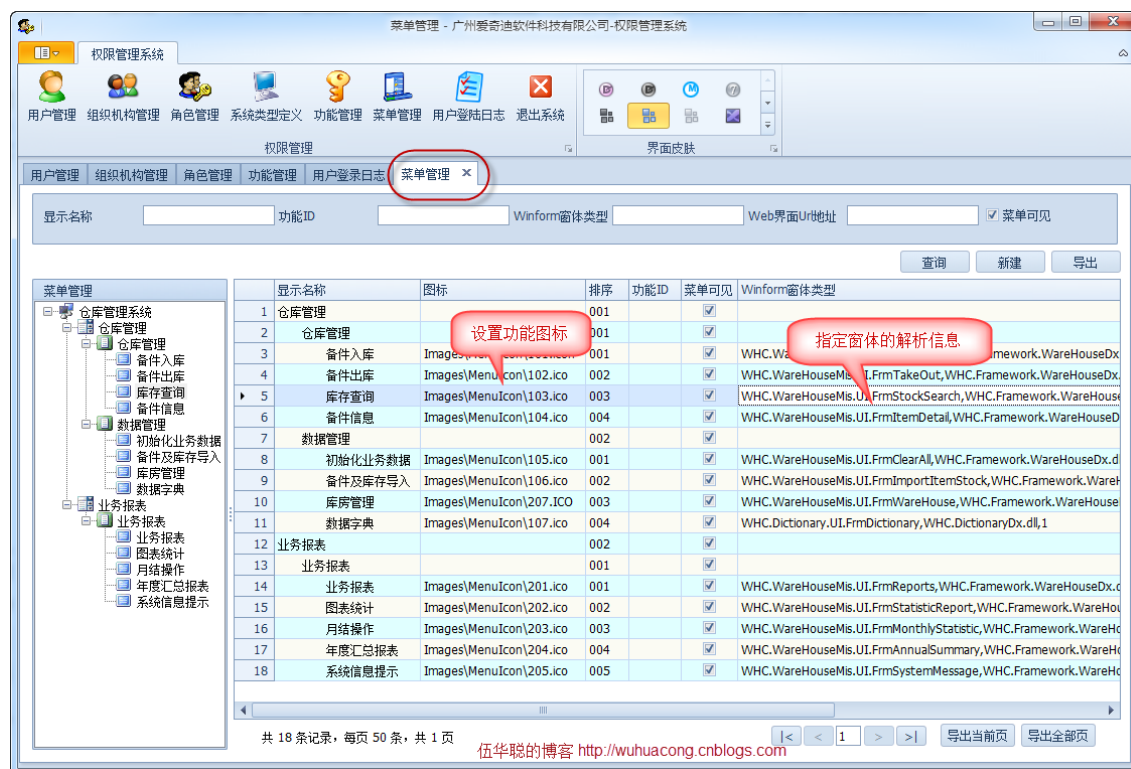
插件化应用也是目前框架和系统开发的主流方向，从开发企业的产品管理角度来看，第一可以把控好已有的框架产品，第二可以把内部的人员分配到不同的业务模块中，让他们遵循一定的接口即可快速开发，从而提高团队对业务模块的横向切割和快速开发的效率，更好、

统一、高效完成企业化应用框架的搭建和使用。

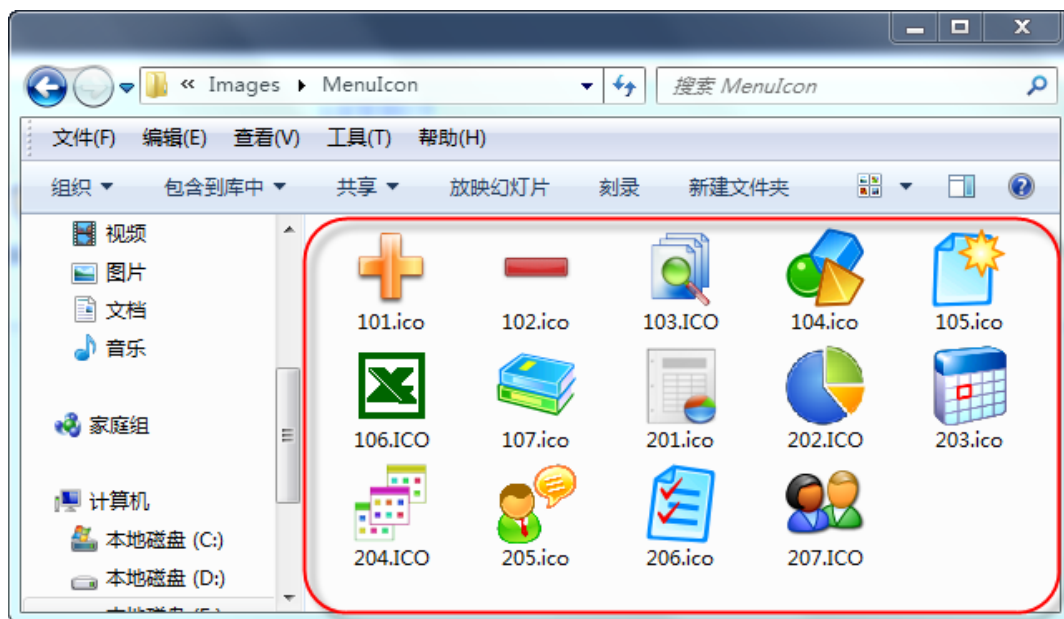
2.1.2. 菜单支持动态配置，可根据用户权限进行动态控制显示

一般 Winform 程序中，菜单都是直接写在主程序中的，这样对于修改和维护菜单的相关信息，可能有所不便，而且对于插件化应用，新增菜单是常见的事情，这样就需要对菜单进行动态配置管理了，通过后台菜单的配置和权限的指定，能够实现菜单的动态加载和权限验证。

因此菜单也是权限分配的一部分，为了有效管理菜单资源，我们把菜单放到权限管理系统中进行管理控制。



设定好菜单的图标和 Winform 窗体类型，就能在 Winform 框架中动态构建菜单信息了。菜单的图表放置在系统的相对目录下即可。



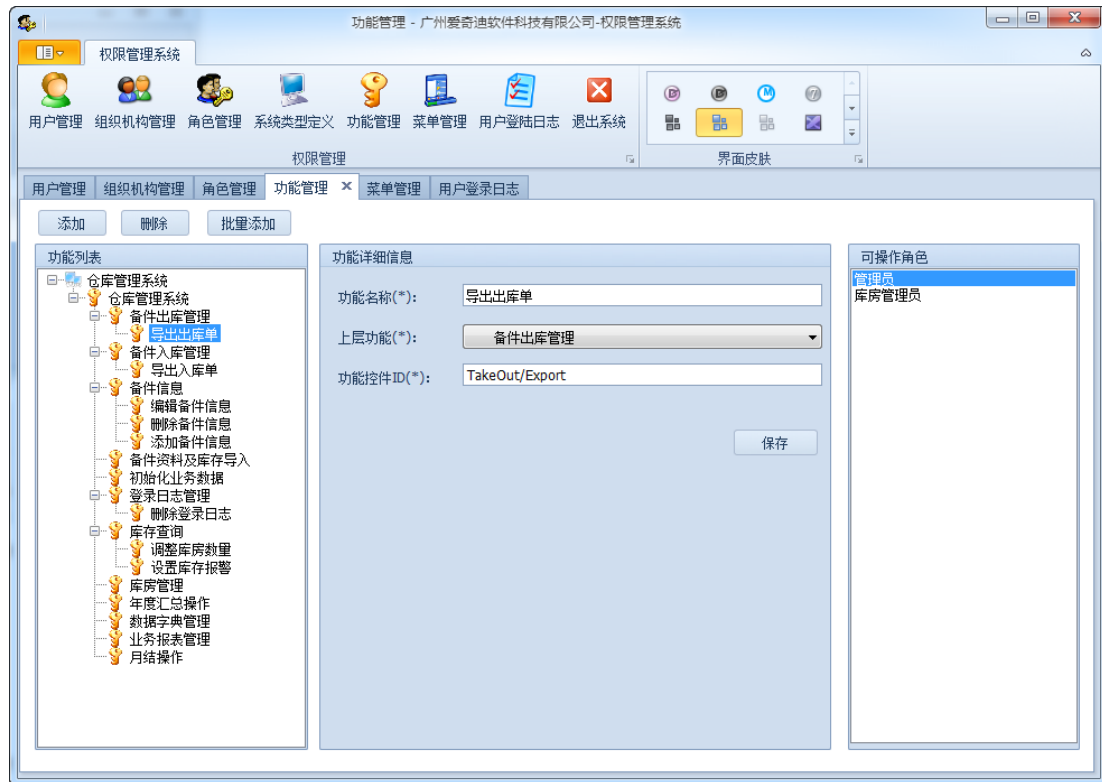
2.1.3. 集成各种独立开发好的模块

如权限管理系统、字典管理模块等，自动更新模块等。

1) 权限管理模块

权限的分配和管理，基本上是每个业务系统需要考虑的东西，而这些常用的东西，在整个 Winform 开发框架中，把它作为一个独立的模块，既方便维护管理，也留有接口很容易进行集成，这样能够在项目中反复进行使用。

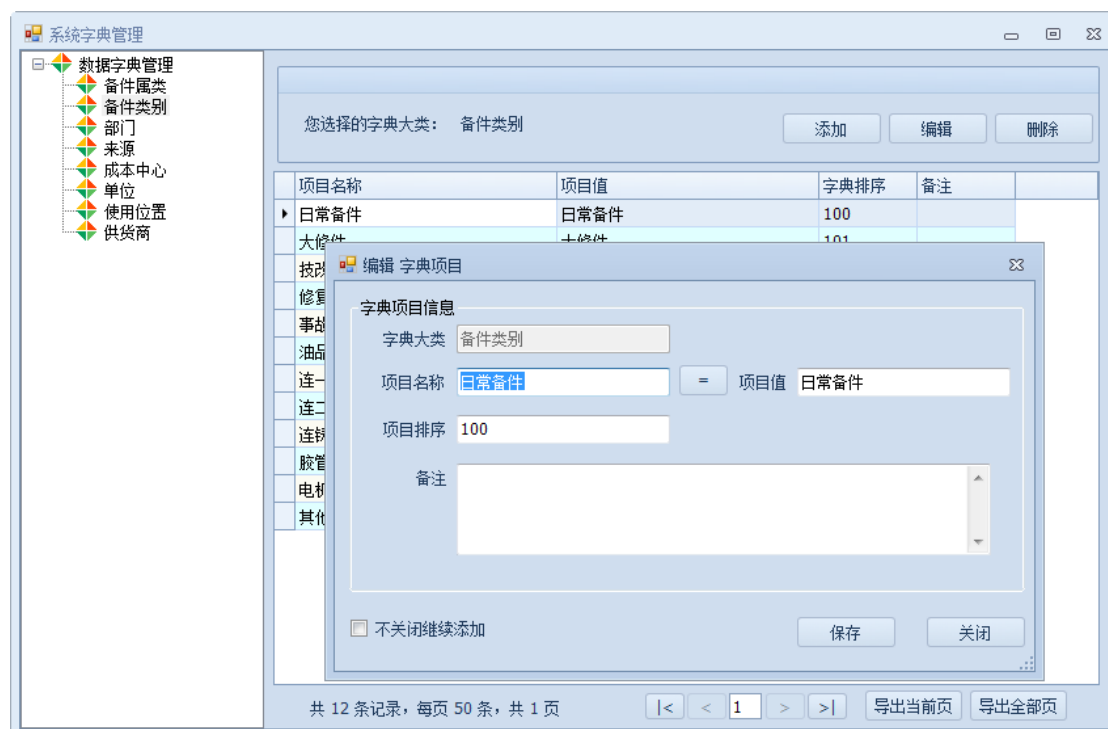
权限系统需要考虑的系统用户、组织机构、角色、权限定义及分配、菜单管理、用户登录日志等相关信息。对于每新增一个系统，我们只需要在权限管理系统中增加一个系统类型定义，以及相关的功能、菜单数据即可，非常方便管理。



2) 字典管理模块

另外，除了权限的管理，字典管理也是常规业务模块的一个总要功能，因此字典管理模块在 Winform 开发框架中，也是一个独立的模块进行开发管理，在使用的时候进行集成即可。

字典的分类可以按多级进行分类管理，可以拖拉进入新的分类，非常方便。

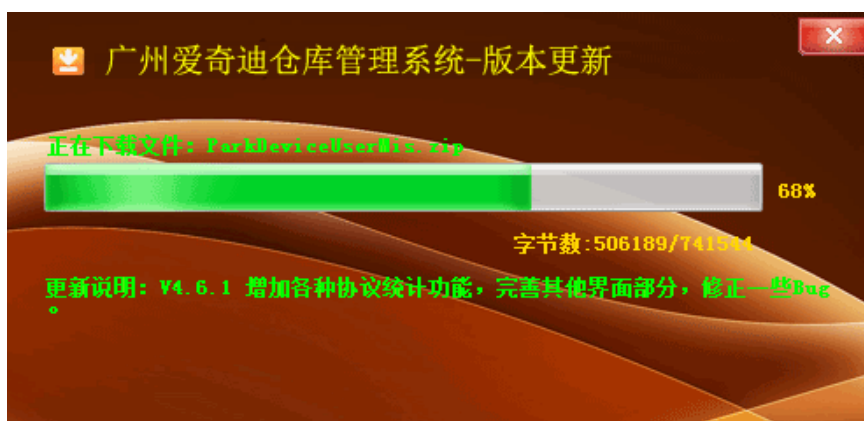


3) 自动更新模块

一般情况下，为了有效管理应用程序的发布及更新，我们需要软件支持远程自动更新功能；在 Winform 开发框架中再引入一个自动升级更新的通用模块，这个自动升级的通用模块除了具备一般的功能外，可以通过配置程序标题、升级路径方式等方式，实现通用化的升级操作，其他需要升级的程序，拿过来就可以使用，经过项目的整合及优化，已经具备较好的应用前景了。

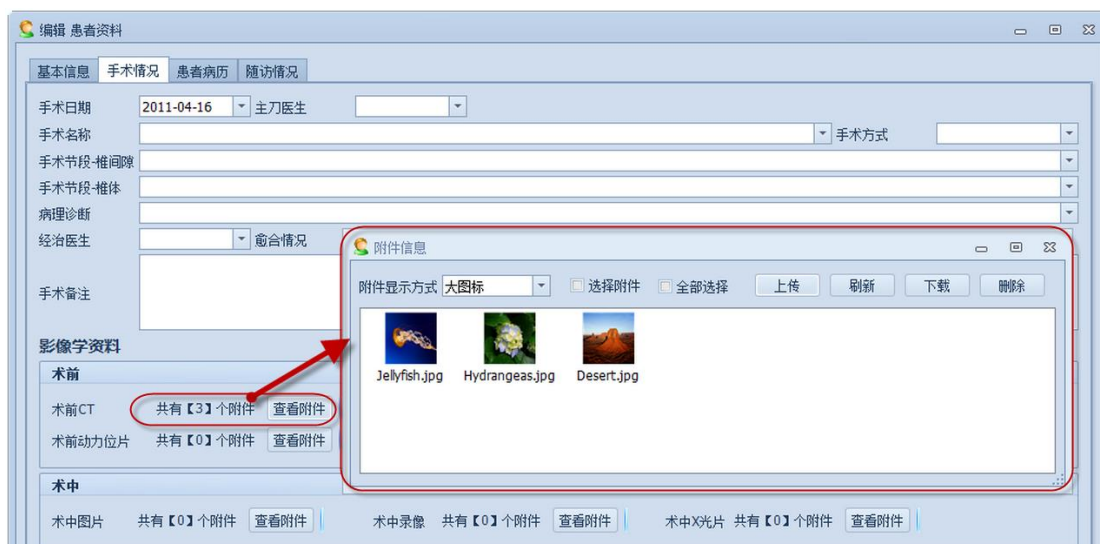
本通用自动更新模块具有下面几个特点：

- a) 程序标题可配置，更新路径可配置，适用于各种需要更新的项目整合。
- b) 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。
- c) 支持文件更新后自动启动，或者带参数的启动方式。

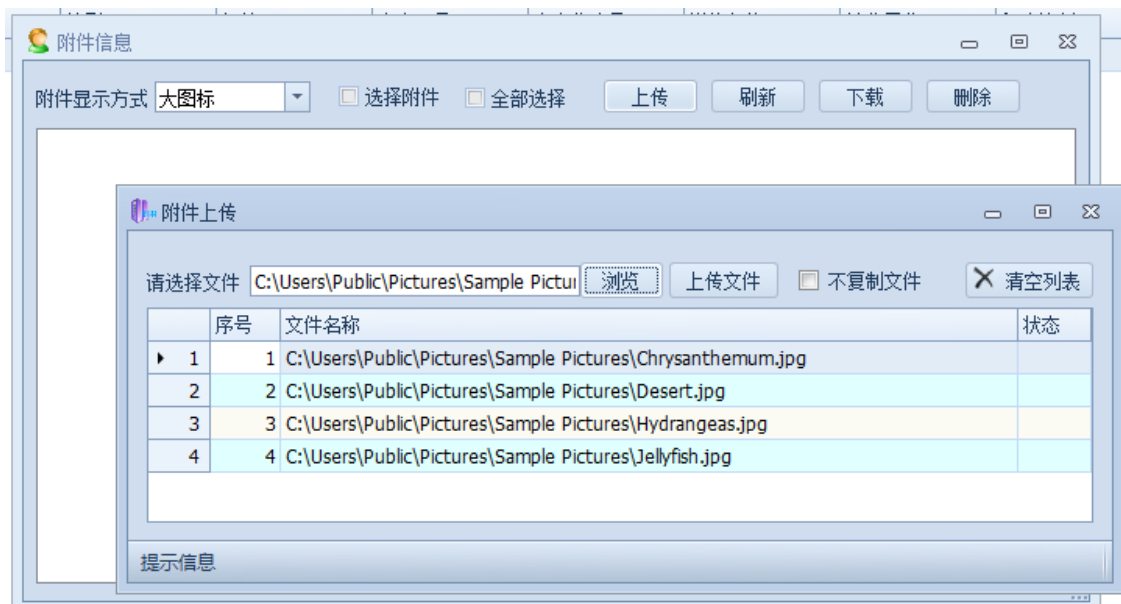


4) 附件管理模块

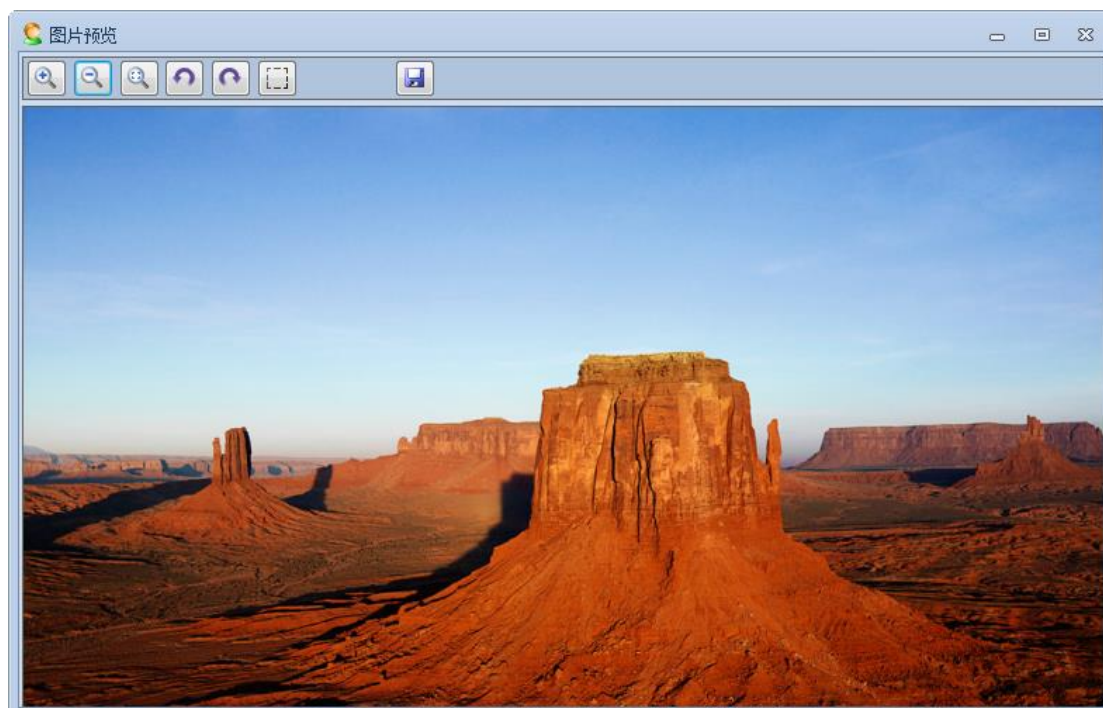
在很多常规的管理系统里面，都可能附件管理的需求。在我做的一系列信息管理系统里面，都希望能很好的管理图片、文档等附件资料，并且可以把它们分门别类的进行存储，需要的时候可以预览查看，那么这个功能我们可以把它独立做成一个附件管理模块，这样可以在各个系统中集成即可。



附件管理的上传界面如下所示，可以从本地选择多个文件一并上传，每个文件上传成功后立即显示状态。

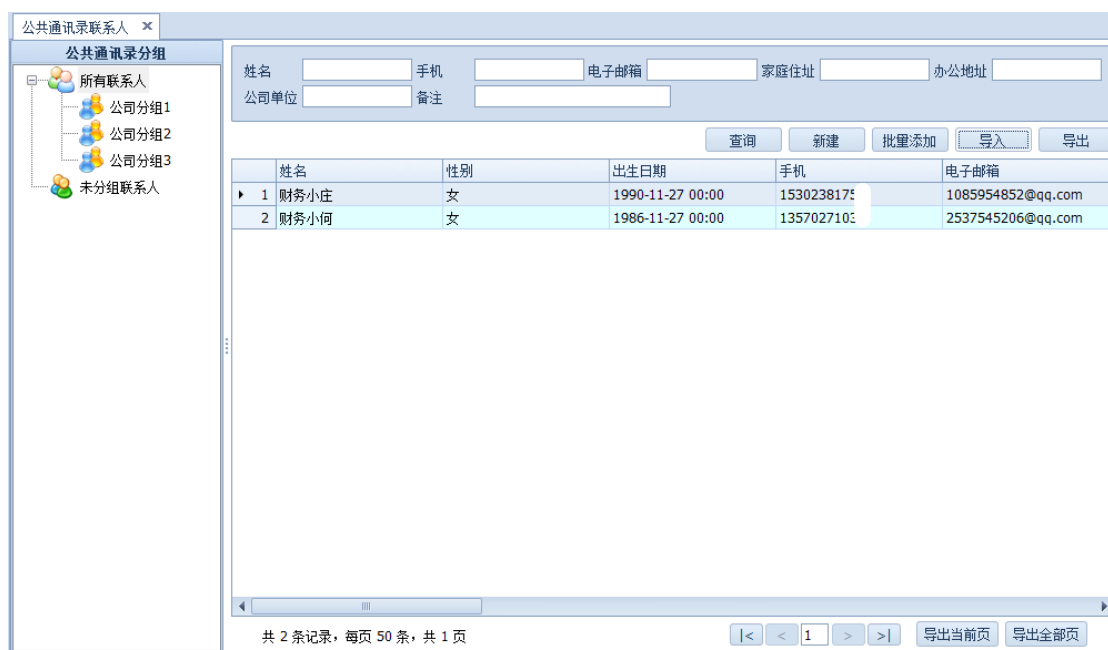


图片双击后可以预览，如果其他文件，那么可以下载后打开。

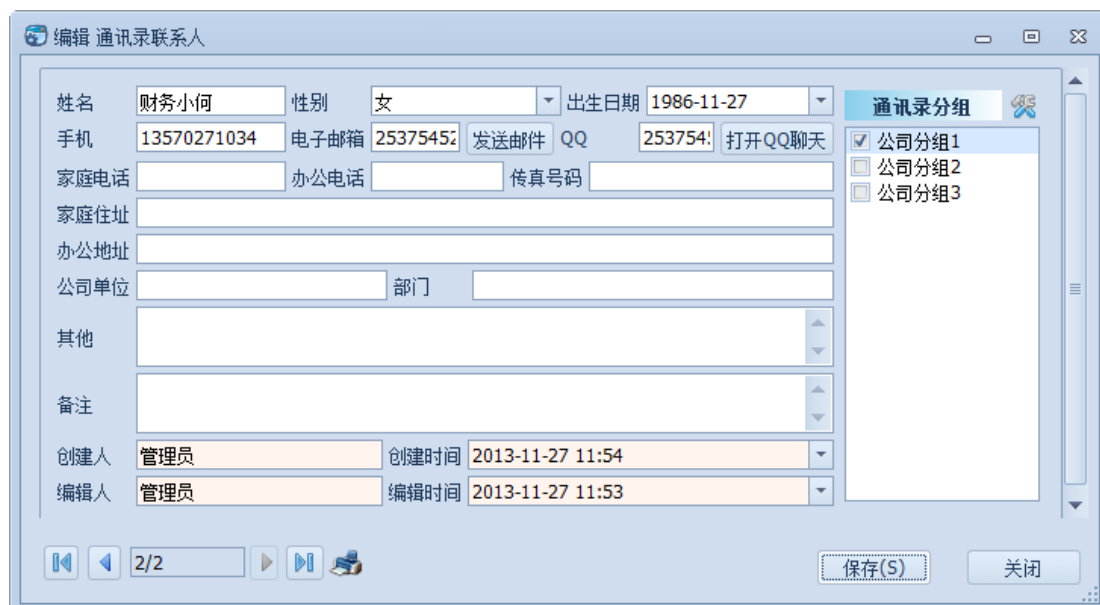


5) 通讯录管理模块

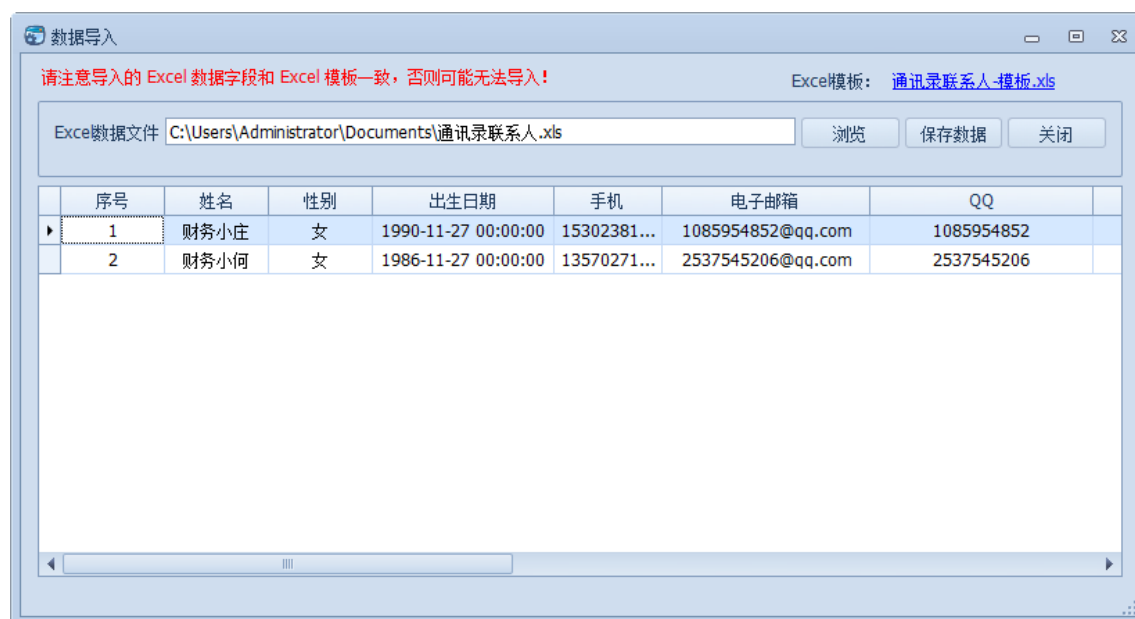
通讯录管理，包含公共通讯录和个人通讯录两个模块，公共通讯录主要是用于公司内部的会员通讯录管理，个人通讯录则是当前用户专属的通讯录数据。



公共通讯录的编辑界面和个人通讯录的界面类似，只是通讯录分组也改为公共通讯录分组列表而已，如下所示，另外通讯录的分组也可以通过单击图标按钮进入管理分类操作。

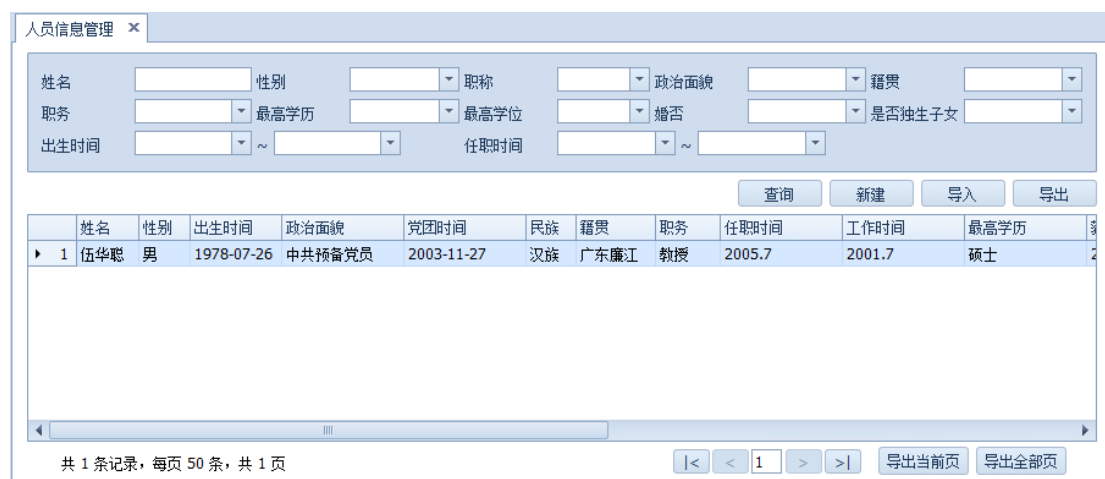


为了方便通讯录的添加，可以进行 Excel 数据的批量导入操作。



6) 人员管理模块

人员管理模块管理企业员工的相关信息，包括基本信息、学习及职称信息、履历情况、家庭及个人信息、个人文件资料等信息。



人员基本信息的编辑界面如下所示。



2.1.4. 可重复使用的系统基础模块

包括 Winform 分页控件和.NET 开发公用类库等可重复使用的模块。

1) Winform 分页控件

为了提高数据的获取及显示效率，一般情况下，我们都需要对数据进行分页获取和显示，在网络化环境，这种方式获取数据特别重要，在 Winform 里面，没有现成的分页控件可以使用，因此出于这个考虑，把数据显示、数据导出、数据打印等众多功能集成一起，更加方便使用。分页控件完美支持列表数据的分页，可以调整显示列及顺序，是否显示复选框，字段别名，奇偶列的颜色变化，列信息提示和数据复制等等众多实用细致的功能的。

列表的数据绑定及显示代码，通过代码生成工具，可以一键生成所需的界面代码，开发效率飞一般的提升。

集成功能强大的分页控件

多种界面样式，集成导出Excel、打印预览、

按需取数据，高效快速、多数据库支持、

可显示信息提示、行号、中文别名、

集成可扩展菜单及事件、奇偶行颜色变化、

动态设置显示列、打印列、

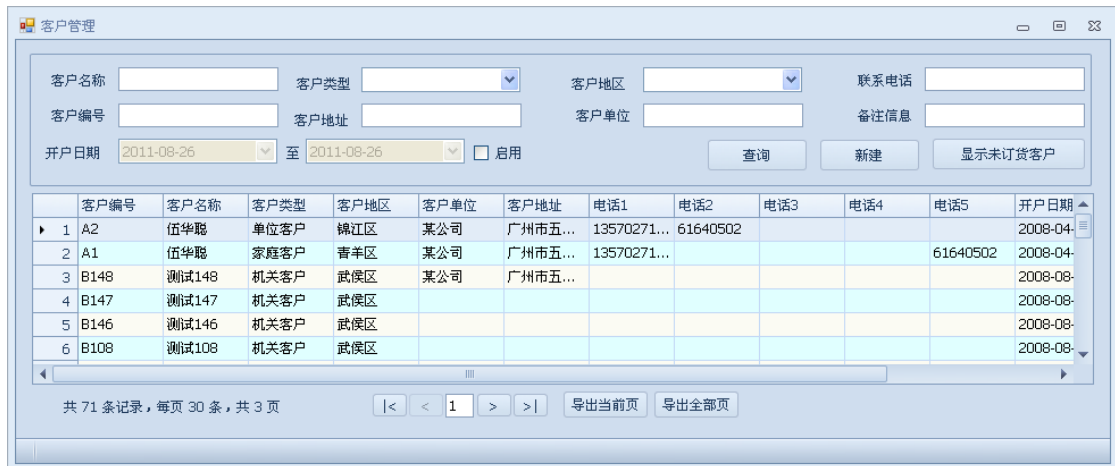
可选择显示、隐藏指定的列、指定顺序。

更多特性等您来发现！

The image displays a Winform application interface with several components:

- Product Management Table:** A table with columns: 编号 (ID), 产品类型 (Product Type), 产品名称 (Product Name), 产品规格 (Product Spec), 产品型号 (Product Model), 进货价 (Purchase Price), 建议价 (Suggested Price), 零售价 (Retail Price). A context menu is open over the table, listing options: 新建 (New), 编辑选定项 (Edit Selected), 删除选定项 (Delete Selected), 刷新列表 (Refresh List), and 打印列表 (Print List).
- Customer Management Form:** A form titled "客户管理" (Customer Management) with fields for 客户名称 (Customer Name), 客户类型 (Customer Type), 客户地址 (Customer Address), 联系电话 (Contact Number), and 备注信息 (Remarks). It includes a "设置列的可见性" (Set Column Visibility) panel with checkboxes for 编号 (ID), 客户名称 (Customer Name), 客户类型 (Customer Type), 客户地区 (Customer Region), 客户单位 (Customer Unit), 客户地址 (Customer Address), 电话1 (Phone 1), 电话2 (Phone 2), 电话3 (Phone 3), 电话4 (Phone 4), 电话5 (Phone 5), 开户日期 (Opening Date), 分机号 (Extension), 备注 (Remarks), and 更新日期 (Update Date). There are also "全选/全不选" (Select All/Deselect All) and "反选" (Invert) options.
- Excel Spreadsheet:** A Microsoft Excel window showing a table with columns: 编号 (ID), 项目编号 (Project ID), 项目名称 (Project Name), 供货商 (Supplier), 图号 (Drawing No.), 规格型号 (Spec Model), 材质 (Material), 备件图类 (Part Drawing Class), 备件类别 (Part Category), 单位 (Unit).
- Print Preview Window:** A window titled "打印预览" (Print Preview) showing a table of customer information.

| 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 |
|------|------|------|------|------|------|-----------|
| ... | ... | ... | ... | ... | 广州市五 | 135702710 |



2) .NET 公用类库

俗话说，一个好汉十个帮，众人拾柴火焰高等都说明一个道理，有更多的资源，更丰富的积累，都是助你走向成功，走向顶峰的推动力。就我们开发者而言，其中技巧的积累、资源的积累，就是类似一个个好汉、一根根好柴，是我们能够进行高效开发的保证和推动力。这些类库是我从事多年软件开发，逐渐提炼和发现的一些闪光点或者好片段，有些是吸收别人的优秀的东西，有些是自己逐步提炼的精华，以前，在网络上看到一些开源的项目，总会先看看其是否有封装良好、功能独立的辅助类库，发现好的辅助类库，总是欣喜若狂好一阵子，学习中逐步积累，研究中逐渐提炼，多年过后，略有小成，终为今天所介绍的辅助类库集合。

这些辅助类库平时也并不是所有的都会用得上，不过一些常用的，几乎各个项目就会用到，类库涉及面非常广，能够为我们开发节省很多时间，并且我们也可以根据自己的需要进行扩充完善，形成自己的类库集合。博客公用类库在线帮助文档列表：

- [厚积薄发，丰富的公用类库积累，助你高效进行系统开发（1）----开篇总结](#)
- [厚积薄发，丰富的公用类库积累，助你高效进行系统开发（2）----常用操作](#)
- [厚积薄发，丰富的公用类库积累，助你高效进行系统开发（3）----数据库相关操作](#)
- [厚积薄发，丰富的公用类库积累，助你高效进行系统开发（4）----CSV、Excel、INI 文件、独立存储等文件相关](#)
- [厚积薄发，丰富的公用类库积累，助你高效进行系统开发（5）----热键、多线程、窗体动画冻结等窗体操作](#)
- [厚积薄发，丰富的公用类库积累，助你高效进行系统开发（6）----全屏截图、图标获取、图片打印、页面预览截屏、图片复杂操作等](#)
- [厚积薄发，丰富的公用类库积累，助你高效进行系统开发（7）-----声音播放、硬件信息、键盘模拟及钩子、鼠标模拟及钩子等设备相关](#)
- [厚积薄发，丰富的公用类库积累，助你高效进行系统开发（8）----非对称加密、BASE64 加密、MD5](#)

[等常用加密处理](#)

[厚积薄发，丰富的公用类库积累，助你高效进行系统开发（9）----各种常用辅助类](#)

[厚积薄发，丰富的公用类库积累，助你高效进行系统开发（10）---各种线程同步的集合类](#)

[厚积薄发，丰富的公用类库积累，助你高效进行系统开发（11）---各种线程相关操作类](#)

[厚积薄发，丰富的公用类库积累，助你高效进行系统开发（12）---网络相关操作辅助类](#)

2.1.5. 代码生成工具 Database2Sharp 的整合

整个框架通过与代码生成工具 Database2Sharp 进行配合，能够一键生成整体性框架代码，Winform 系统界面代码，开发更高效。

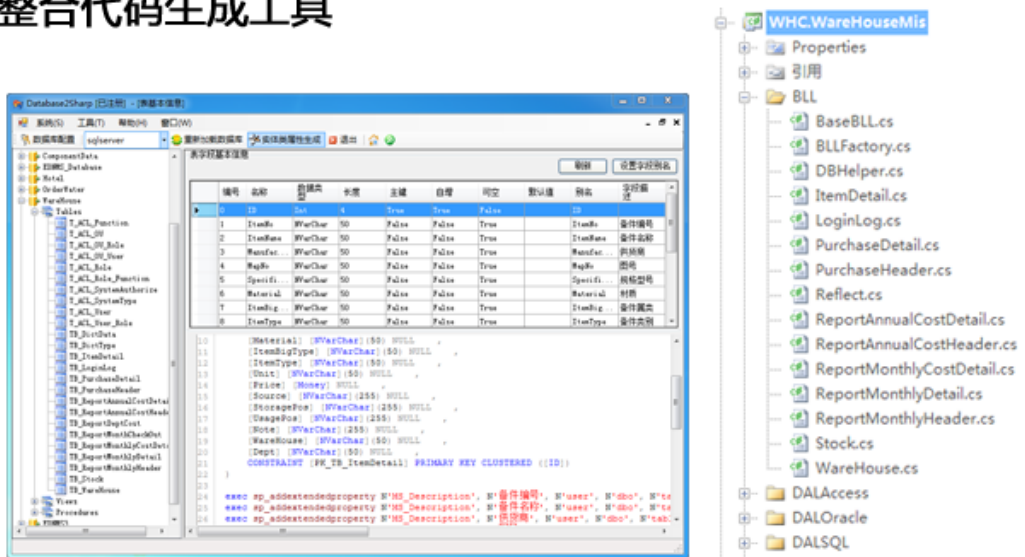
代码生成工具Database2Sharp

| | |
|---|--|
|  <p>多数据库支持 支持Oracle、SQLServer等5种数据库，可浏览表数据。内置代码生成处理逻辑，不同数据库生成不同的数据访问层代码。</p> |  <p>解决方案代码一键生成 一键生成界面代码、业务代码、数据访问、实体类等各层代码，生成即可编译通过。</p> |
|  <p>数据元数据及对象关系 具有完备的数据库各种元数据信息应用，方便在模板引擎中使用数据库对象及关系。</p> |  <p>模板可定制性 可以根据需要自定义代码生成模板，快速使用模板引擎生成所需代码。</p> |
|  <p>Enterprise架构代码生成 最优的框架架构代码生成，历经多年项目开发总结及提炼，完备、丰富的基类封装，操作更方便。</p> |  <p>Web界面代码生成 可快速生成Web界面层代码，生成即可编译运行，极大提高开发效率。</p> |
|  <p>Winform界面代码生成 提供Winform界面代码级界面后台逻辑代码生成，开发Winform更方便。界面可以生成传统、DevExpress、DotNetBar等项目代码。</p> |  <p>集成优秀控件 生成代码集成各种良好的控件。集成分页控件、公用类库、权限管理、字典管理等模块。</p> |
|  <p>数据库文档生成 可快速生成数据库文档，直接可用于数据库设计文档。</p> |  <p>服务支持 实时网上的专业问题解答，技术支持。</p> |

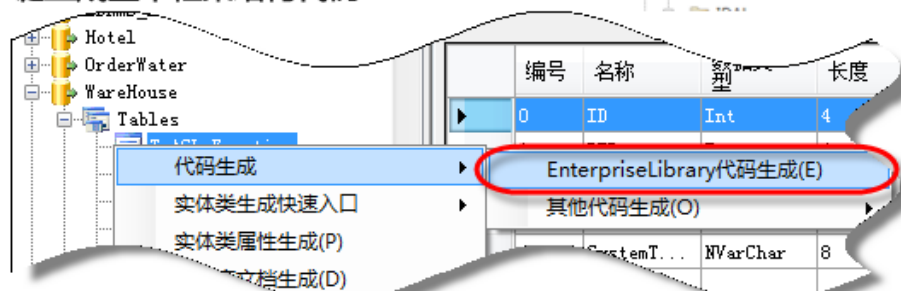
在整个 Winform 开发框架中， Database2Sharp 生成出来的代码体现了非常完美的整合性，能够无缝接入开发的框架系统中，无论是常规的业务逻辑和数据访问层代码，以及一些列表、编辑界面的 Winform 界面代码，都能快速生成，稍作调整即可满足业务模块的需要。

Database2Sharp 是一个简单点击几次鼠标就能完成一周代码量的代码生成工具，效率惊人、友好体贴，真正的开发好伴侣。提供了对 SqlServer 2000/2005/2008、Oracle、Mysql、Access、SQLite、国产达梦数据库等的支持；可以生成各种架构代码（包括 Winform 开发框架、WCF 开发框架、混合式开发框架等，以及 Winform 界面代码，并且和 Winform 开发框架完美整合，体现出更高的开发效率。

整合代码生成工具



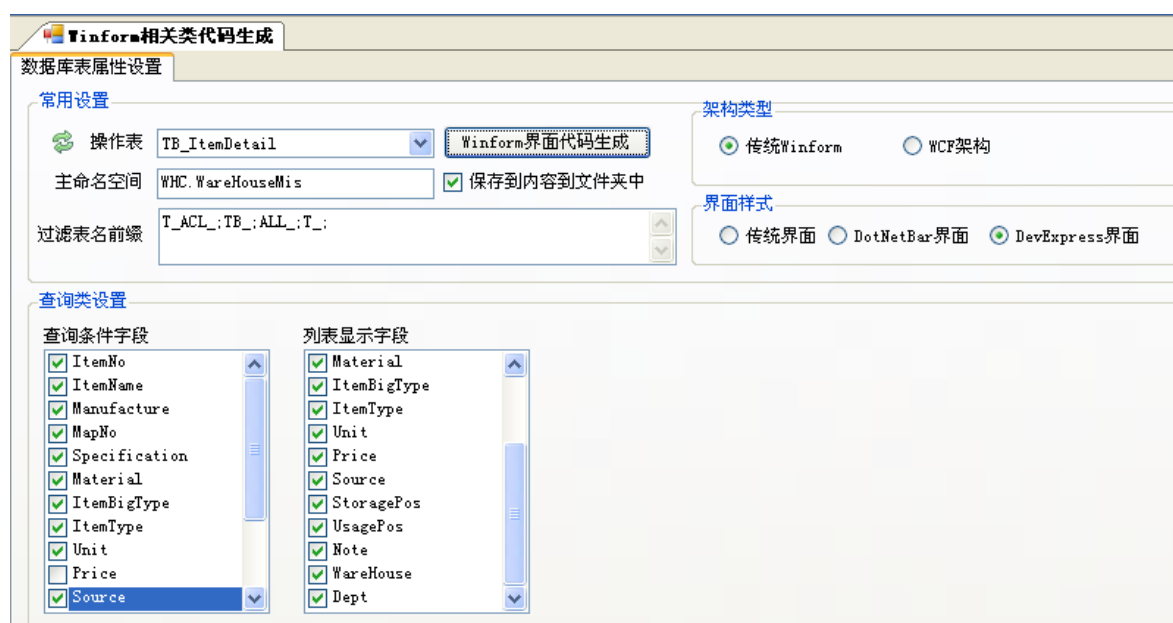
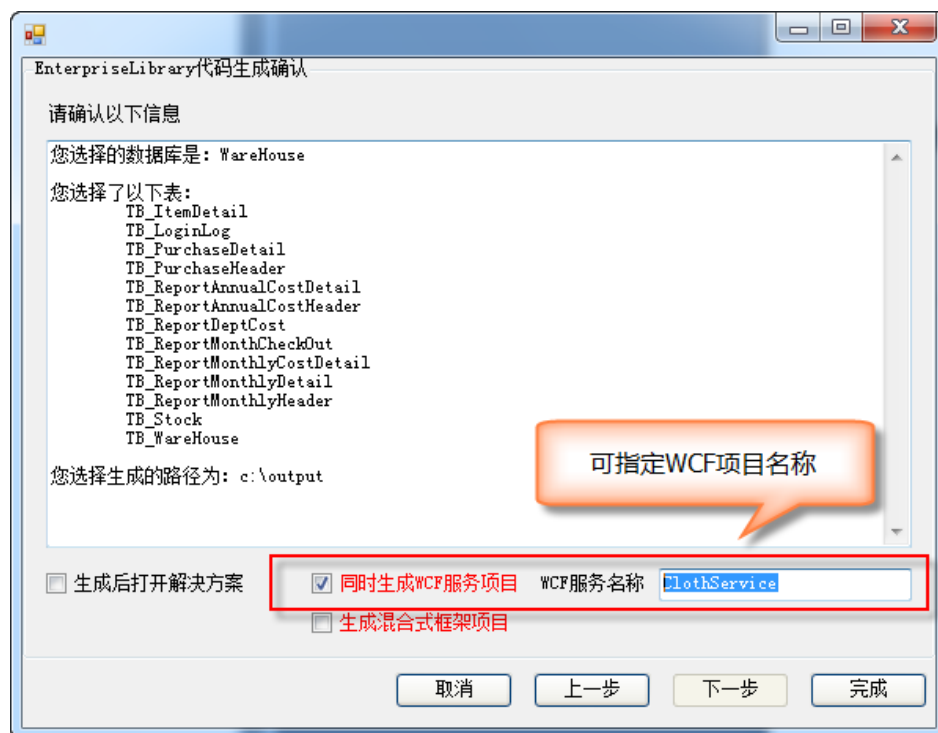
一键生成整个框架结构代码



界面代码快速生成



在开发框架中，代码生成工具 Database2Sharp 可以根据需要生成各种不同框架的代码，极大提高了开发效率和企业内部的编码统一。各种框架的界面代码，也可以通过 Database2Sharp 里面的“Winform 界面代码生成”功能进行快速生成，直接复制到项目中使用即可。



2.1.6. 通用的 Excel 数据导入导出

通用的 Excel 数据导入导出功能，能快速导入自定义模块的 Excel 数据，快速导出列表数据。

由于一般的业务系统，经常性的数据导入时很正常的业务需求，因为毕竟使用 Excel 来操作数据也很方便，或者由于系统之间的数据交换需要，我们需要提供一个入口给客户导入

所需要的数据。但是导入数据的时候，不同的业务数据对应不同的 Excel 文件，很难做到统一，但如果是每个业务模型，都创建一个不同的导入界面来操作 Excel 数据，又会觉得可能某种程度上重复劳动，增加开发及维护成本。

那么有无一种介于两者之间的方法，来实现效率的最优化，并且能够统一利用好一个导入的界面呢，在开发领域，只要能想到的，一般也能做到，由于工作的需要，在我的 Winform 开发框架中引入了一个通用的数据导入模块，来实现这个既是统一，又是变化的业务需求。

Winform 框架提供的个通用的 Excel 数据导入导出机制，通过代码生成工具 Database2Sharp 自动生成的代码，就包含了如何使用这个通用导入模块的相关代码以及该模块的导出数据的代码，我们要做的就是系统运行起来，导出一些数据作为某个模块的 Excel 模板即可。下面的功能按钮就是使用代码生成工具自动生成的界面包含的按钮。



通用数据导入功能，包含下面几个方面的内容。



下面就是一个实际生成的功能模块，其导入界面的运行效果。



在最底的状态栏里面，但我们保存数据的时候，会调用后台线程进行数据保存，并显示数据导入的进度状态，由于是采用后台线程处理，不会阻塞当前的界面，在多文档的 [Winform 开发框架](#) 界面中，可以切换到其他业务界面进行其他处理，不影响整体界面操作。

2.1.7. 自定义 Excel 自定义模板报表的生成

本模块用于生成各种复杂的 Excel 报表。很多情况下，我们需要生成比较专业的 Excel 模块，因此自定义模板报表就是一种很好的方案，Winform 开发框架提供了多种自定义 Excel 报表的生成。

支持各种自定义报表生成

自定义模板报表实例是你编制报表最好的老师

2010年1月库房结存月报表

| 项 目 | 上月结存 | | 本月入库 | | 本月出库 | | 本月结存 | |
|----------|------|----|------|----|------|----|------|----|
| | 数量 | 金额 | 数量 | 金额 | 数量 | 金额 | 数量 | 金额 |
| 标准件（国产） | | | | | | | | |
| 非标准件（国产） | | | | | | | | |
| 标准件（进口） | | | | | | | | |
| 非标准件（进口） | | | | | | | | |
| 合计 | | | | | | | | |
| 日常备件 | | | | | | | | |
| 大修件 | | | | | | | | |
| 技改件 | | | | | | | | |
| 修复件 | | | | | | | | |
| 事故件 | | | | | | | | |
| 油品 | | | | | | | | |
| 送一钢板 | | | | | | | | |
| 送二钢板 | | | | | | | | |
| 送特殊零件 | | | | | | | | |
| 取寄零件 | | | | | | | | |
| 其他零件 | | | | | | | | |

复杂表格、函数自定义
动态生成行列、组合列
变量定义、行列计算



使用普通的二维表，虽然能满足大多数的情况，不过在一般的业务中，自定义模板的报表根据贴近实际，符合客户的要求，虽然自定义模板的报表，比普通的二维报表复杂一些，不过利用 Aspose.Cell 控件，并在预设模板中预设变量，可以生成很复杂的报表。

具体的自定义模板报表可以参考下我总结的两篇文章。

[使用 Aspose.Cell 控件实现 Excel 高难度报表的生成（一）](#)

[使用 Aspose.Cell 控件实现 Excel 高难度报表的生成（二）](#)

例如生成一个标准的出库单，这个表单有表头信息，列表数据信息，并非一个普通的二维表，而且这种格式比较固定，因此很适合自定义模块报表的生成操作。

出库单

领用部门：（公章） 成本中心： 部门： 库房编号： 出库日期：

| 领用 | 序 | 备件编号 | 备件名称 | 规格型号 | 领用数 | 单位 | 参考单价 |
|----|---|------|------|------|-----|----|------|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

库管员： 库管章：

其他设计模板如下所示：

实际生成的报表如下所示：

| 项 目 | | 上月结存 | | 本月入库 | | 本月出库 | | 本月结存 | |
|-------|--|------|---------|------|---------|------|---------|------|---------|
| | | 数量 | 金额 | 数量 | 金额 | 数量 | 金额 | 数量 | 金额 |
| 备件属类1 | | 10 | 1000.00 | 20 | 2000.00 | 30 | 300.00 | 40 | 4000.00 |
| 备件属类2 | | 11 | 1100.00 | 21 | 2100.00 | 31 | 3100.00 | 41 | 4100.00 |
| 备件属类3 | | 12 | 1200.00 | 22 | 2200.00 | 32 | 3200.00 | 42 | 4200.00 |
| 合计 | | 33 | 3300 | 63 | 6300 | 93 | 6600 | 123 | 12300 |
| 备件属类1 | | 10 | 1000.00 | 20 | 2000.00 | 30 | 300.00 | 40 | 4000.00 |
| 备件属类2 | | 11 | 1100.00 | 21 | 2100.00 | 31 | 3100.00 | 41 | 4100.00 |
| 备件属类3 | | 12 | 1200.00 | 22 | 2200.00 | 32 | 3200.00 | 42 | 4200.00 |
| 合计 | | 33 | 3300 | 63 | 6300 | 93 | 6600 | 123 | 12300 |

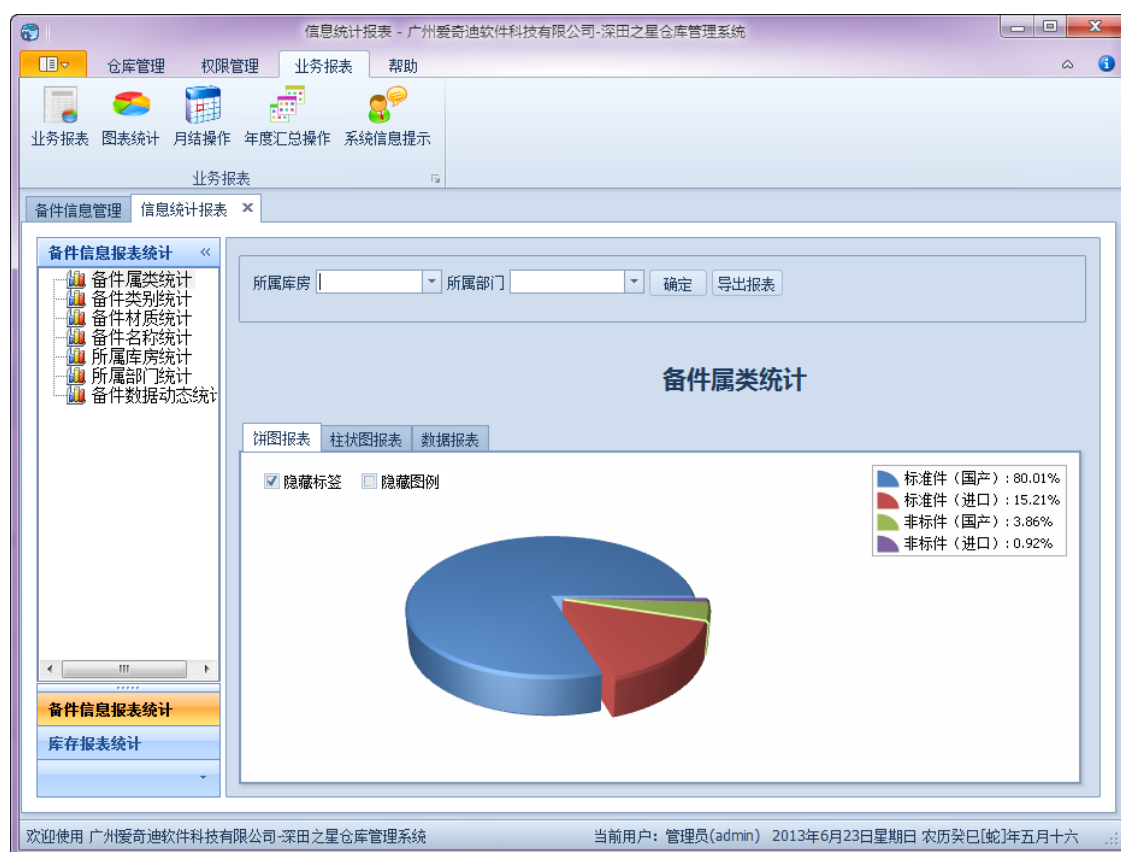
2.1.8. 强大通用的统计图表模块，数据统计更方便

统计图表在很多项目都可能用到，集成到框架中，更方便大家对一些图表项目的设计理解以及功能的重用。在一般的传统的框架中，可以采用 ZedGraph 开源控件或者微软自带

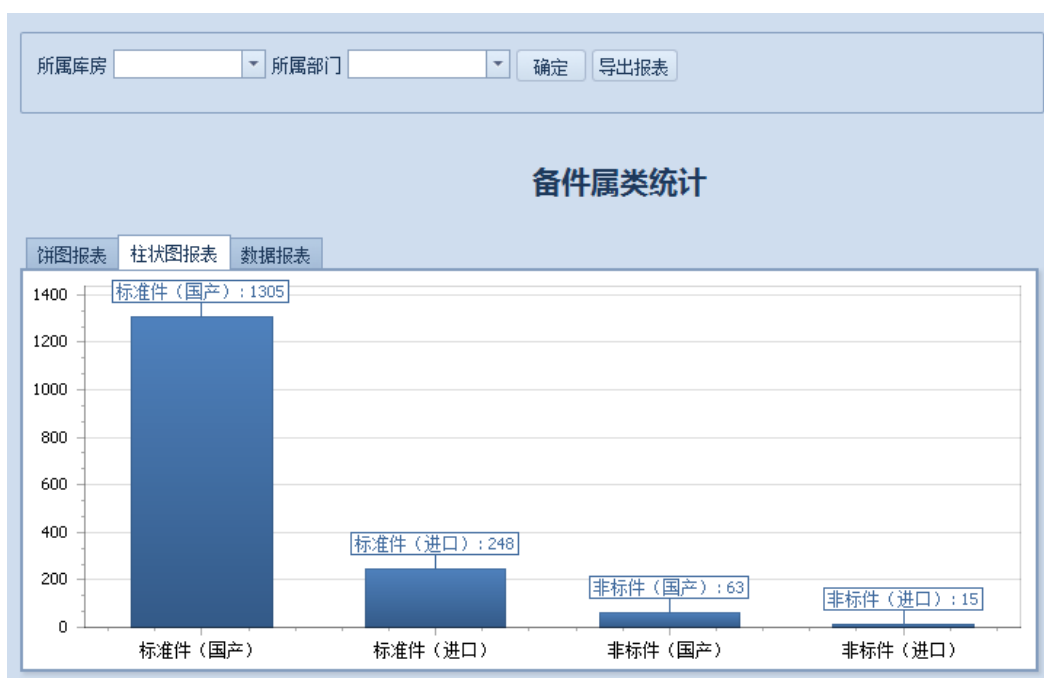
的 MSChart 进行图表设计，DevExpress 控件套件有自己的图表控件，这里主要介绍基于 DevExpress 控件的图表控件进行图表设计。

1) 普通统计图表

这里指的普通统计图表，只是对表某一项目进行单一的统计，可以从饼状图、柱状图的图表中体现这些项目各自所占的比例和数值，在 Winform 框架中的普通统计图表模块中，包括了饼状图、柱状图和数据表格，这样更方便对数据进行全面分析和查看。整个模块是可以重用的，指定字段属性就可以比较合理的展现出不同分类项目的统计效果了，具体效果图如下所示。



上面的统计图表中，还包含了下面两个功能模块，如下所示。



所属库房 所属部门 确定 导出报表

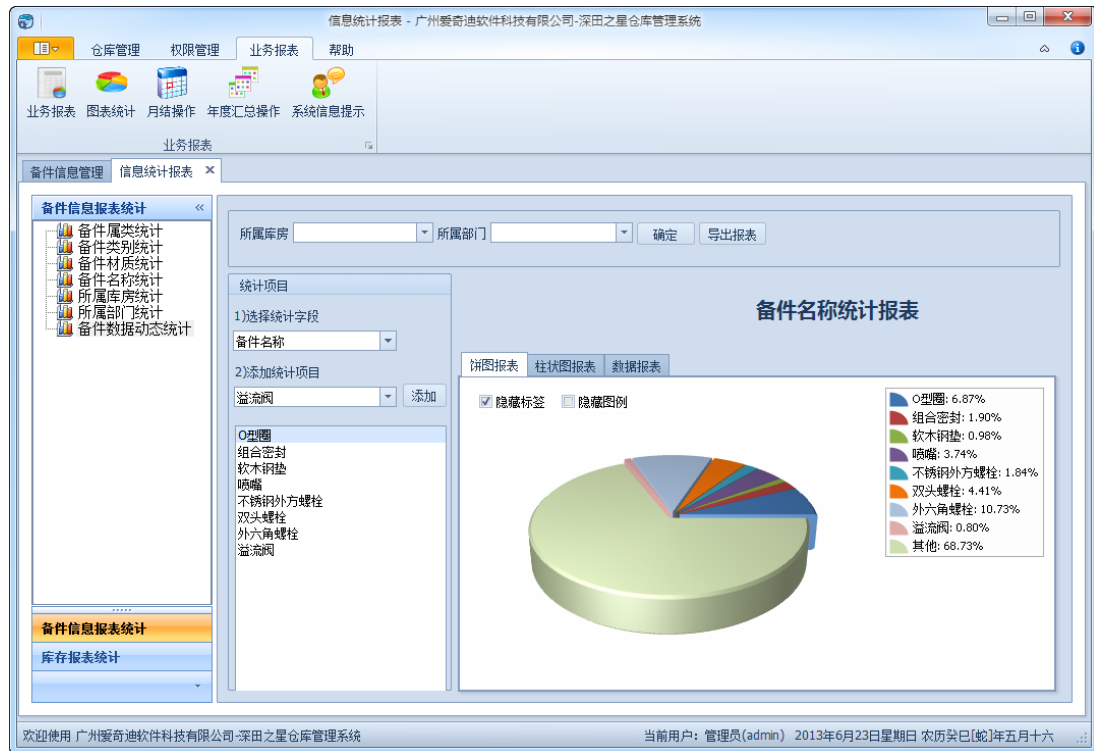
备件属类统计

饼图报表 柱状图报表 数据报表

| 统计项目 | 统计值 |
|-----------|------|
| ▶ 标准件(国产) | 1305 |
| 标准件(进口) | 248 |
| 非标件(国产) | 63 |
| 非标件(进口) | 15 |

2) 动态项目统计图表

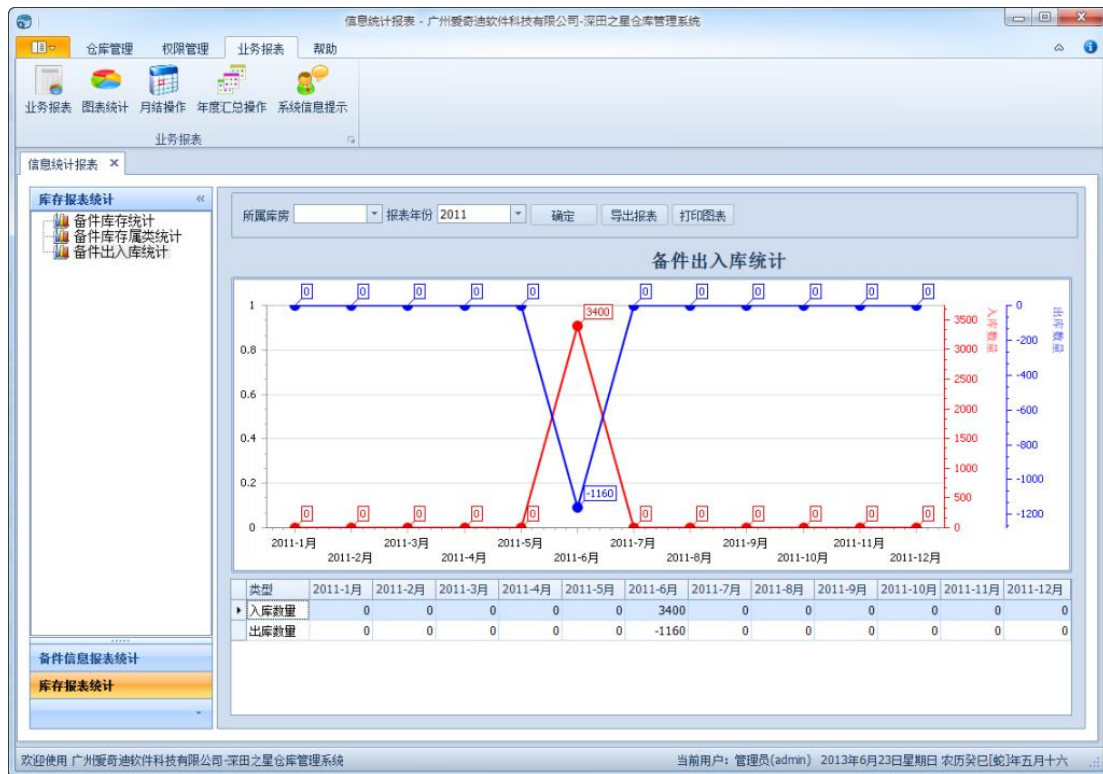
有时候，对于表里面的数据，可能要对不同类型的内容进行动态的统计，以确定他们各自的比例情况，那么这些动态项目的统计图表就比较合适了，例如，对于病人资料的管理，可能需要统计各种病种所占的比例或者各种职业类型的犯病率，这些不太确定的统计项目，就需要一个能够支持动态项目的统计图表进行支撑，对于本 Winform 框架，为了较好呈现这个类型报表的意义，我选择了对备件类型所占的比例进行一个统计分析，得到下面的统计图表，如下所示。



上面的图表统计，除了能够根据一些条件进行限定查询范围外，还可以对一些预设的统计字段进行动态选取，然后根据字段里面的各种内容（统计项目）进行统计，这样就可以比较有效的统计出各种类型的数值和比例了。

3) 多重坐标对比统计图表模块

在 Winform 框架里面，可以对某一年各月份的出入库数量进行分析，得到下面的统计图。



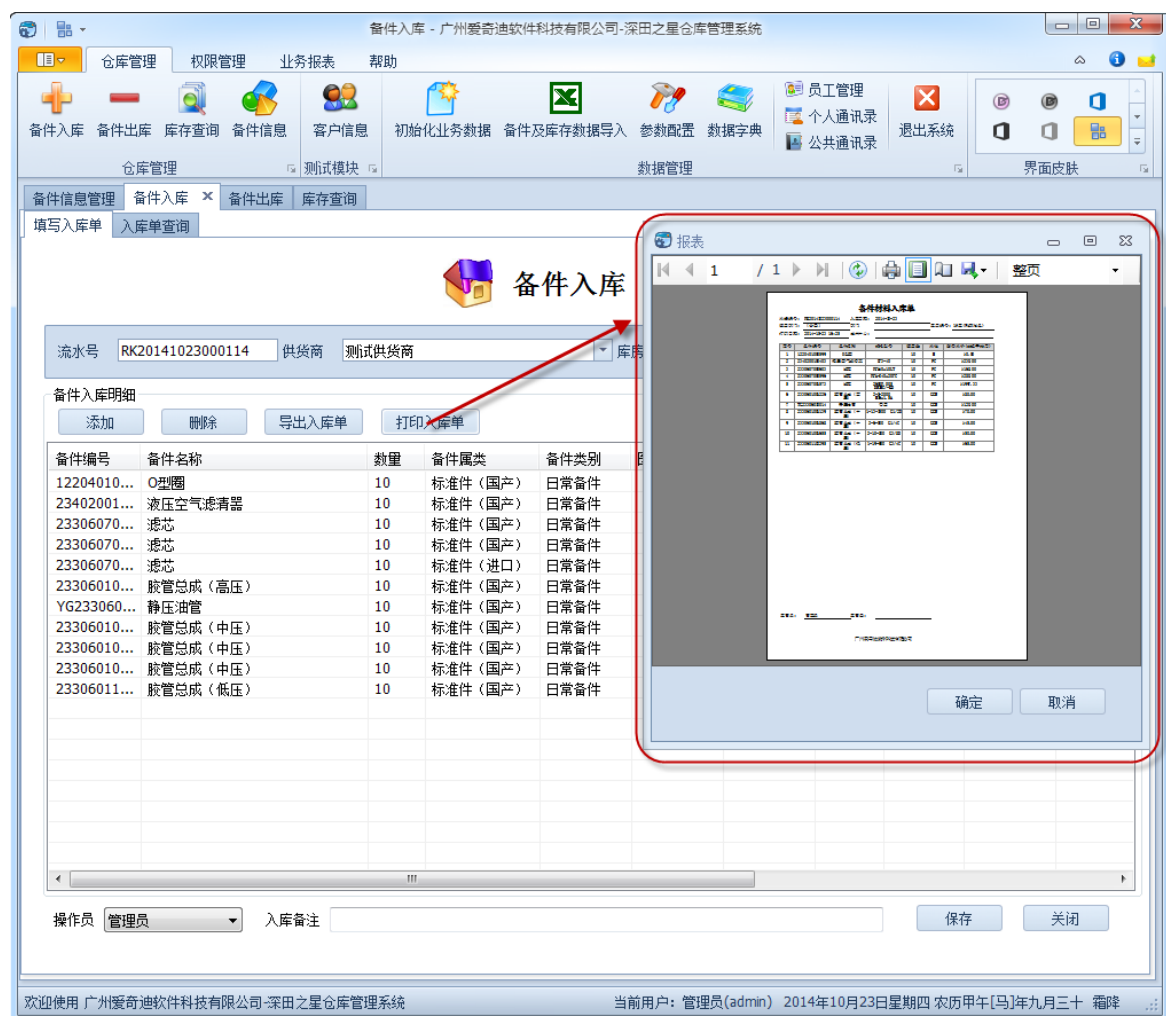
以上数据不多，展现可能不太好看，下面我给出我另一个软件系统的界面，其中对病人的出入院记录进行一个统计对比分析，统计报表如下所示。



2.1.9. 专业美观 RDLC 设计报表，方便打印浏览和 Excel 导出

RDLC，它的全称是 Report Definition Language Client-Side，原来是微软基于 SQL Server 报表服务中提供的一种被称为报表定义语言（Report Definition Language, RDL）的改进版本，增加了客户端处理报表功能，在 WinForm 和 WebForm 中均可以使用这种报表。RDLC 报表展现是基于一个定义的报表文件的，一般是以 rdlc 后缀名的报表文件。

本 Winform 框架，集成了 RDLC 通用报表，在入库单，出库单模块里面，均提供了报表的演示模块，方便学习使用。



报表

1 / 1

100%

备件材料入库单

单据编号: RK20141023000114 入库日期: 2014-10-23

领用部门: (公章) 部门: _____ 库房编号: 1#库(轧钢准备)

打印日期: 2014-10-23 16:25 成本中心: _____

| 序号 | 备件编号 | 备件名称 | 规格型号 | 领用数 | 单位 | 参考单价(加权平均后) |
|----|---------------|----------|------------------------|-----|-----|-------------|
| 1 | 1220401000599 | O型圈 | | 10 | M | ¥0.00 |
| 2 | 2340200100403 | 液压空气滤清器 | EF3-40 | 10 | PC | ¥230.00 |
| 3 | 2330607000603 | 滤芯 | RFA40*10LY | 10 | PC | ¥168.00 |
| 4 | 2330607000596 | 滤芯 | RFA-240*20FC | 10 | PC | ¥280.00 |
| 5 | 2330607001573 | 滤芯 | 2600R 005 BN3HC/-KB | 10 | PC | ¥1997.33 |
| 6 | 2330601001226 | 胶管总成(高压) | 2-8-2000 M16*1.5A | 10 | GEN | ¥80.00 |
| 7 | YG23306010014 | 静压油管 | 专用 | 10 | GEN | ¥120.00 |
| 8 | 2330601001139 | 胶管总成(中压) | 1-13-1500 G1/2D | 10 | GEN | ¥70.00 |
| 9 | 2330601001868 | 胶管总成(中压) | 2-6-500 G1/4C | 10 | GEN | ¥45.00 |
| 10 | 2330601001655 | 胶管总成(中压) | 2-10-800 G3/8D | 10 | GEN | ¥50.00 |
| 11 | 2330601102295 | 胶管总成(低压) | 1-19-600 G3/4C | 10 | GEN | ¥65.00 |

确定 取消

导出的 Excel 效果如下所示:

WHC.WareHouseMis.PurchaseReport.xls [兼容模式] - Excel

文件 开始 插入 页面布局 公式 数据 审阅 视图 加载测试 团队 登录

P14 : GEN

| 序号 | 备件编号 | 备件名称 | 规格型号 | 领用数 | 单位 | 参考单价(加权平均后) |
|----|---------------|----------|------------------------|-----|-----|-------------|
| 1 | 1220401000599 | O型圈 | | 10 | M | ¥0.00 |
| 2 | 2340200100403 | 液压空气滤清器 | EF3-40 | 10 | PC | ¥230.00 |
| 3 | 2330607000603 | 滤芯 | RFA40*10LY | 10 | PC | ¥168.00 |
| 4 | 2330607000596 | 滤芯 | RFA-240*20FC | 10 | PC | ¥280.00 |
| 5 | 2330607001573 | 滤芯 | 2600R 005 BN3HC/-KB | 10 | PC | ¥1997.33 |
| 6 | 2330601001226 | 胶管总成(高压) | 2-8-2000 M16*1.5A | 10 | GEN | ¥80.00 |
| 7 | YG23306010014 | 静压油管 | 专用 | 10 | GEN | ¥120.00 |
| 8 | 2330601001139 | 胶管总成(中压) | 1-13-1500 G1/2D | 10 | GEN | ¥70.00 |
| 9 | 2330601001868 | 胶管总成(中压) | 2-6-500 G1/4C | 10 | GEN | ¥45.00 |
| 10 | 2330601001655 | 胶管总成(中压) | 2-10-800 G3/8D | 10 | GEN | ¥50.00 |
| 11 | 2330601102295 | 胶管总成(低压) | 1-19-600 G3/4C | 10 | GEN | ¥65.00 |

库管员: 管理员 库管章: _____

报导入出的Excel格式

WHC.WareHouseMis.PurchaseReport

就绪

2.1.10. 参数配置管理，可以配置系统的各种运行参数

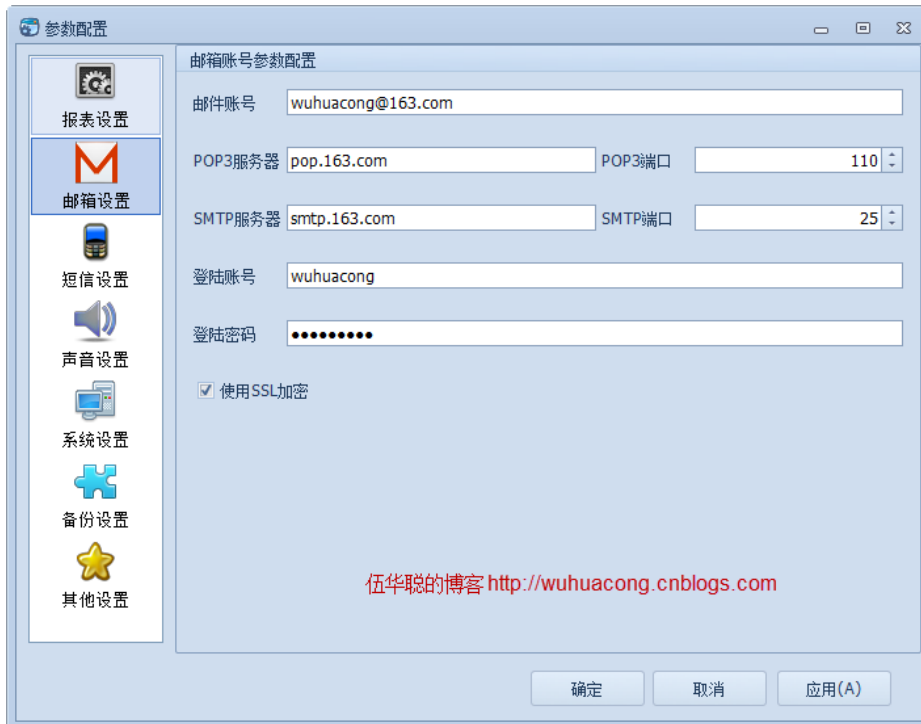
在 Winform 程序里面，很多时候，需要保存一些参数配置信息，有些是可能需要存储在文件即可，有些则希望保存在数据库里面，这样可以随时随地都能够获取自己的配置信息了。结合 FireFoxDialog 参数配置界面组件和 SettingsProvider.net 技术，可以实现较为美观、灵活的 Winform 程序参数配置管理。

1) SettingsProvider.net 功能介绍

这个是在 GitHub 上的一款参数配置组件，能够基于普通配置文件、ProgramData 目录文件、独立存储区文件等方式的配置文件存储，它主要是基于 Json 格式进行的配置保存，因此我们还可以把它存储到数据库，虽然官方没有提供例子，但是我们很容易通过扩展实现这个功能，因此在框架里面，我扩展了这个组件，增加了数据库保存的模块，并在字典管理模块里面增加一个表：TB_UserParameter，用来存储相关的参数数据。

2) 参数配置功能界面截图

基于上述的 SettingsProvider.net 的组件，我们可以结合我前面介绍的 FireFoxDialog 界面效果，实现较好的参数配置管理功能，如下界面所示。



我们可以分别把不同的参数放到不同的存储介质里面去，如一些常用的，可以配置到本地目录文件里面去，一些和个人信息相关的内容，我们可以把它放到数据库里面去，这样可以在

各个客户端使用都不需要重新配置，非常方便。

框架提供的参数配置的例子，针对性包含了两个，一个是基于本地文件参数存储，一个是基于数据库文件的参数存储。

使用 XML 方式的参数存储代码如下所示：

```
public partial class PageReport : PropertyPage
{
    private SettingsProvider settings;
    private ISettingsStorage store;

    public PageReport()
    {
        InitializeComponent();

        if(!this.DesignMode)
        {
            // PortableStorage: 在运行程序目录创建一个 setting 的文件记录
            参数数据
            store = new PortableStorage();
            settings = new SettingsProvider(store);
        }
    }
}
```

注意：**PortableStorage** 是在运行程序目录创建一个 **setting** 的文件记录参数数据，运行后，最终会在目录下生成一个类似“ReportParameter.settings”的文件，它的内容格式如下所示。

```
[{"Key": "CarSendReportFile", "Value": "\\WHC. CarDispatch. CarSendBill2. rd1c\\"}]
```

使用数据库存储类：数据库的存储类 **DatabaseStorage**，内容将保存在数据库里面，而且我们通过用户的标识 **Create** 进行路径的针对性处理，可以使每个用户的配置文件不同。

```
public partial class PageEmail : PropertyPage
{
    private SettingsProvider settings;
    private ISettingsStorage store;

    public PageEmail()
    {
        InitializeComponent();

        if (!this.DesignMode)
```



```
    {  
        //DatabaseStorage: 在数据库里面，以指定用户标识保存参数数据  
        string creator = Portal.gc.LoginUserInfo.Name;  
        store = new DatabaseStorage(creator);  
        settings = new SettingsProvider(store);  
    }  
}
```

2.1.11. 基于多数据库的数据查询模块和通用高级查询模块

基于多数据库的数据查询模块和通用高级查询模块，查询数据更方便。

在我的 Winform 开发框架中，使用了一个查询辅助类 SearchCondition 来实现查询条件的获取和转化，这个辅助类内置了对多种数据库条件的分析处理，因此能够很好生成所需要的数据查询条件，正确高效获取所需的数据进行显示。

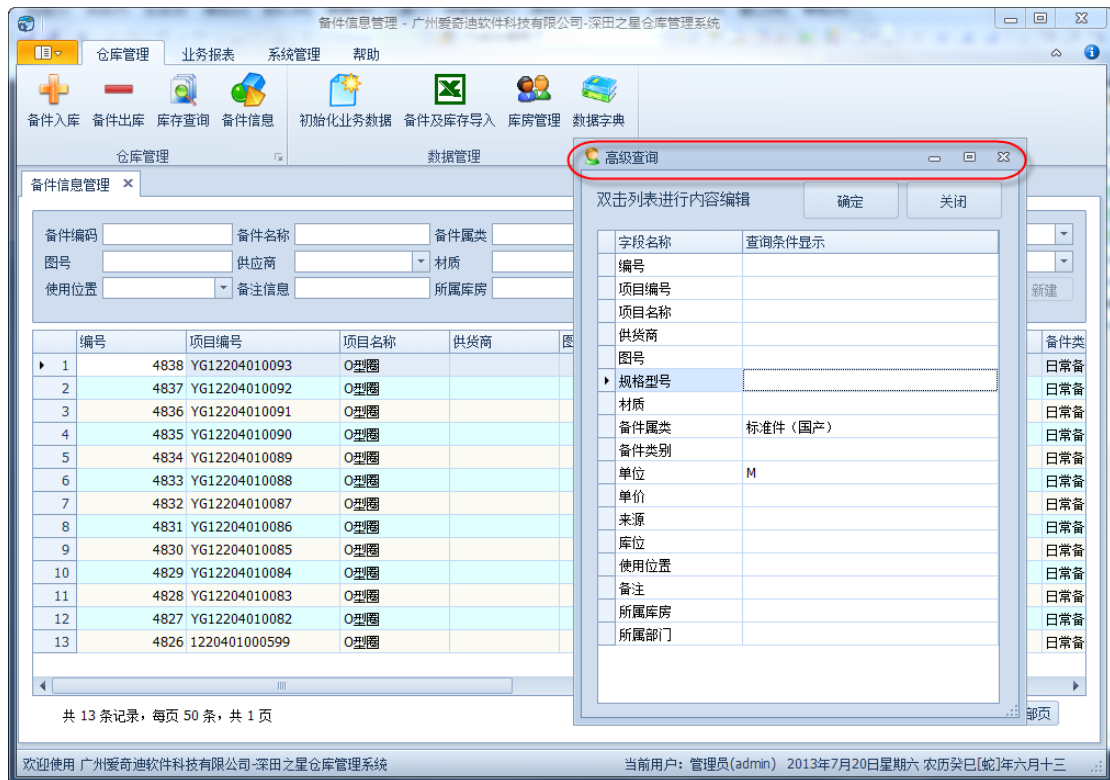
```
/// <summary>  
/// 根据查询条件构造查询语句  
/// </summary>  
private string GetConditionSql()  
{  
    //如果存在高级查询对象信息，则使用高级查询条件，否则使用主表条件查询  
    SearchCondition condition = advanceCondition;  
    if (condition == null)  
    {  
        condition = new SearchCondition();  
        condition.AddCondition("ItemName", this.txtName.Text, SqlOperator.Like)  
            .AddCondition("ItemBigType", this.txtBigType.Text, SqlOperator.Like)  
            .AddCondition("ItemType", this.txtItemType.Text, SqlOperator.Like)  
            .AddCondition("Specification", this.cmbSpecNumber.Text, SqlOperator.Like)  
            .AddCondition("MapNo", this.txtMapNo.Text, SqlOperator.Like)  
            .AddCondition("Material", this.txtMaterial.Text, SqlOperator.Like)  
            .AddCondition("Source", this.txtSource.Text, SqlOperator.Like)  
            .AddCondition("Note", this.txtNote.Text, SqlOperator.Like)  
            .AddCondition("Manufacture", this.txtManufacture.Text, SqlOperator.Like)  
            .AddCondition("ItemNo", this.txtItemNo.Text, SqlOperator.LikeStartAt)  
            .AddCondition("WareHouse", this.txtWareHouse.Text, SqlOperator.Like)  
            .AddCondition("Dept", this.txtDept.Text, SqlOperator.Like)  
            .AddCondition("UsagePos", this.txtUsagePos.Text, SqlOperator.Like)  
            .AddCondition("StoragePos", this.txtStoragePos.Text, SqlOperator.Like);  
    }  
}
```

```
    }

    string where = condition.BuildConditionSql().Replace("where", "");
    return where;
}

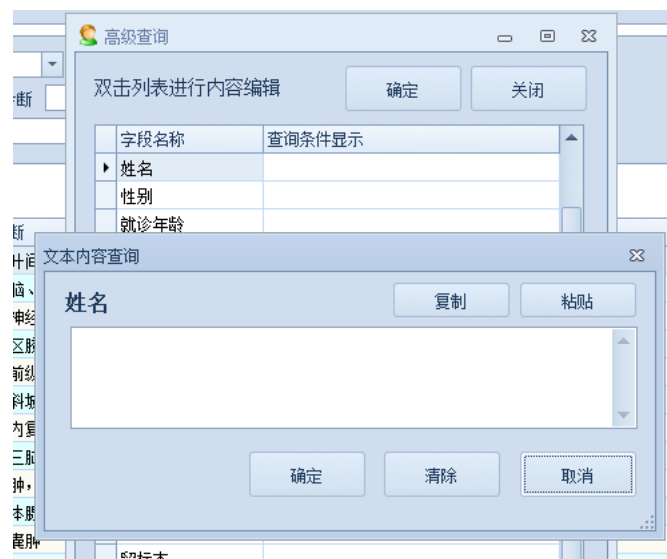
/// <summary>
/// sql 的查询符号
/// </summary>
public enum SqlOperator
{
    [Description("Like 模糊查询")]
    Like,
    [Description("Not Like 模糊查询")]
    NotLike,
    [Description("Like 开始匹配模糊查询, 如 Like 'ABC%')]
    LikeStartAt,
    [Description("= 等于号")]
    Equal,
    [Description("<> (≠) 不等于号")]
    NotEqual,
    [Description("> 大于号")]
    MoreThan,
    [Description("<小于号")]
    LessThan,
    [Description(">=大于或等于号 ")]
    MoreThanOrEqual,
    [Description("<= 小于或等于号")]
    LessThanOrEqual,
    [Description("在某个字符串值中")]
    In
}
}
```

另外，一个好的数据查询，一般有一个强大的高级查询模块，这个模块在很多程序中都很常见，也是给客户扩展查询的一个很好的补充，由于我一直希望我的 Winform 开发框架能够精益求精，所以做了这个通用高级查询模块，希望对今后我自己所有的项目以及框架本身，都能高效的使用。

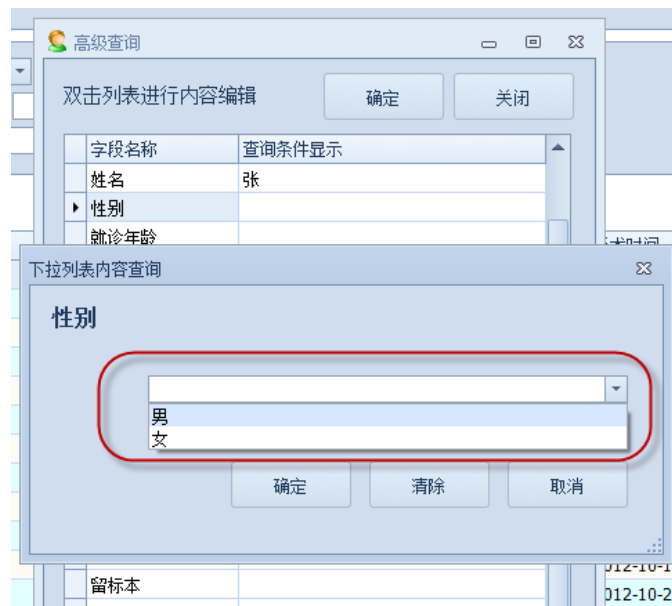


在介绍输入条件的时候，我们注意到，查询输入，基本上可以分为几类：其一是常规的文本类型，使用文本框替代即可；其二是下拉列表类型，用户从列表下面选择内容；其三是日期类型，需要用户指定开始日期和结束日期；其四是数字类型，需要用户指定起始和结束的数值。

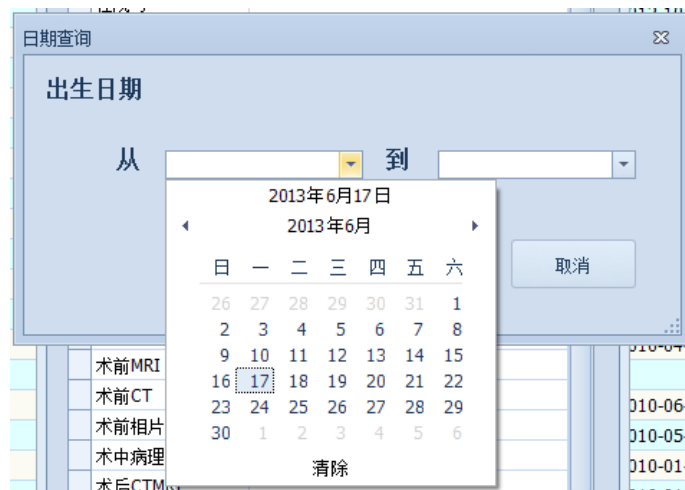
1) 常规的文本类型条件输入界面：



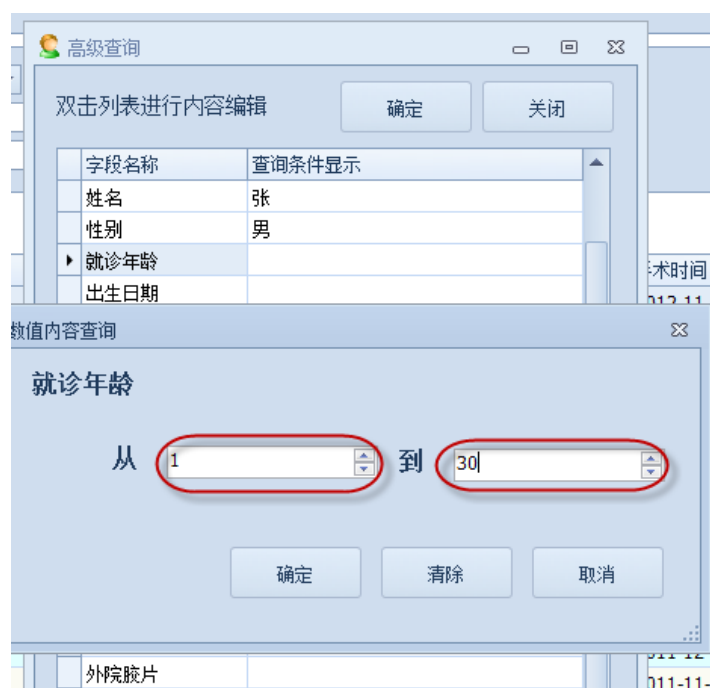
2) 下拉列表类型条件输入界面：



3) 日期类型条件输入界面:



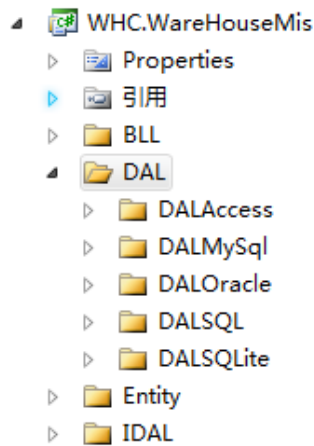
4) 数字类型条件输入界面:



2.1.12. 框架提供基于多种数据库的整合

框架提供基于多种数据库（Sqlserver/Oracle/Mysql/Sqlite/Access/达梦数据库)的整合。虽然我们在实际项目中，一般采用一种数据库进行处理，但是不同的项目，采用的数据库类型可能不同，本 Winform 开发框架为了方便演示和扩展的需要，内置支持了 Sqlserver/Oracle/Mysql/Sqlite/Access/达梦数据库，更多的数据库，也可以通过扩展数据库访问基类的方式进行更多数据库的支持。

Winform 开发框架里面的所有模块，如用到了数据存储的，如权限管理管理模块、通用数据字典管理模块，均内置支持这几种数据库的整合支持。整个 Winform 开发框架的数据库访问，能够手动配置数据库类型，对于同一种数据库，也可以把数据存储分开存储，如业务数据存储在一个数据库，权限管理控制存储在另外一个数据库这种方式。

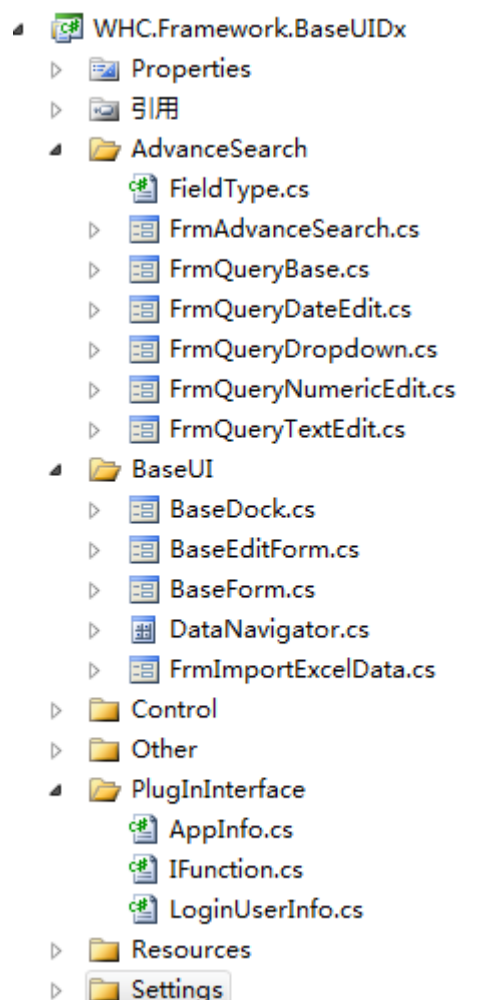


Winform 开发框架提供多种数据库支持，数据访问基类依然很精简，因为我们利用的数据库访问模块是 EnterpriseLibrary，把数据库抽象化，并且我把所有数据库通用操作放在了一个超级基类上，具体的数据库基类只需要实现变化的部分即可。业务访问类则使用泛型进行封装处理。

因此，Winform 开发框架提供了高度封装的数据访问基类，开发代码更少更高效。

2.1.13. 框架界面基类也进行统一封装，使用更方便，效果更统一

为了更好开发常用界面模块，Winform 开发框架把一些公用的界面模块，统一放置在了一个 BaseUI 的项目中，把其中的通用高级查询、通用数据导入模块、常见处理界面基类，插件接口等模块放在一起，根据易于管理和使用。这样开发的模块，重用很多常规的界面，开发效率更快，使用更方便，效果更统一了。



2.1.14. 框架提供基础性的支持

框架提供基础性的支持，包括集成登陆、闪屏、托盘功能，以及为插件模块提供登陆用户信息和系统信息。

整个 Winform 开发框架，是有一个框架启动模块进行集中处理的，系统启动后，用户登录处理后，通过动态加载菜单和插件模块，并在主体框架界面中进行展示，提供权限控制和登录用户信息等方面的框架支持。另外框架支持闪屏图片的动态配置，系统激活热键和托盘缩小灯功能。

登录、闪屏、热键、托盘

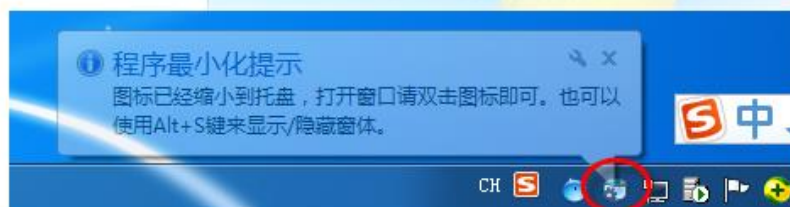


细节决定成败，
细节也是影响客户体验的重要环节
Winform框架为你托起欢呼的喝彩！

大方得体的登录界面



Alt+S的热键操作，一键隐藏/显示
最小化提示及托盘图标管理，更显专业



闪屏减少等待的乏味，提高产品体验

由于系统登录后，框架本身存储了用户登录和权限信息，框架动态加载某个模块后，会把用户信息和权限控制信息，注入到模块的界面基类中，因此插件的界面模块只要是继承了BaseUI的界面基类，就能够获取到用户信息和权限控制信息了。

2.1.15. Web API 接口框架的支持

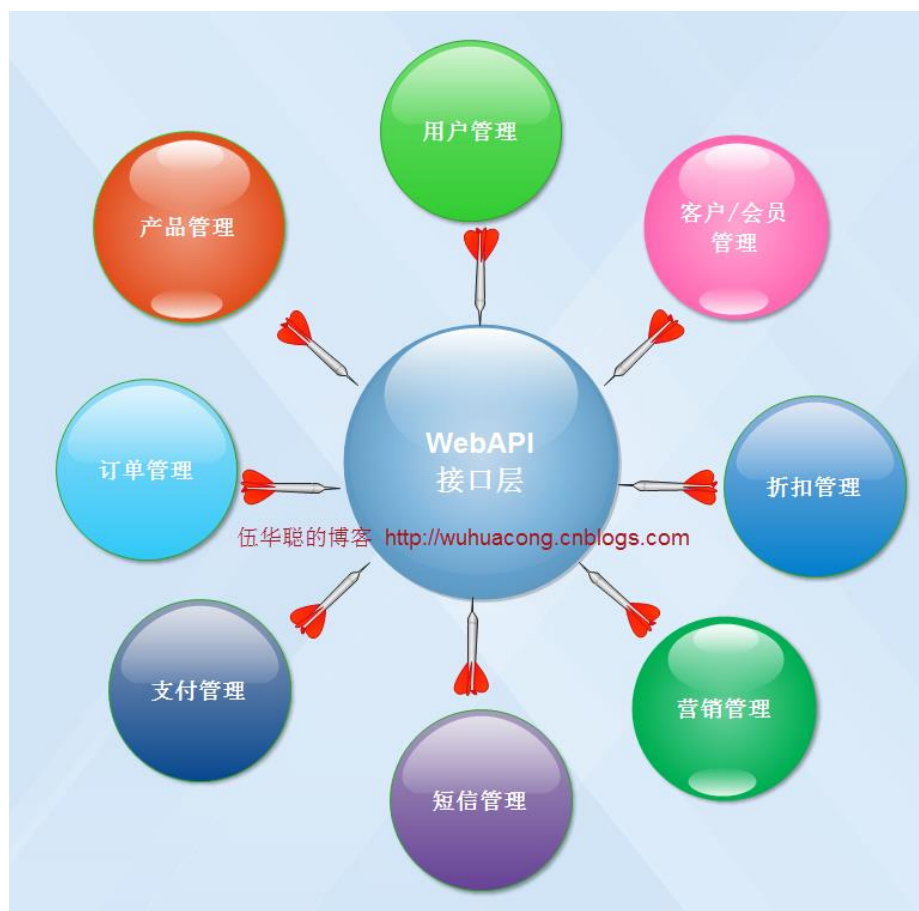
Web API 是一种应用接口框架，它能够构建 HTTP 服务以支撑更广泛的客户端（包括浏览器，手机和平板电脑等移动设备）的框架，ASP.NET Web API 是一种用于在 .NET Framework 上构建 RESTful 应用程序的理想平台。在目前发达的应用场景下，我们往往需要接入 Winform 客户端、APP 程序、网站程序、以及目前热火朝天的微信应用等，这些数

据应该可以由同一个服务提供, 这个就是我们所需要构建的 Web API 平台。由于 Web API 层作为一个公共的接口层, 我们就很好保证了各个界面应用层的数据一致性。

通过上面的了解, 我们可以知道, 所有外部的应用, 其实都可以基于一个相同的 Web API 核心开展的, 如下图所示。

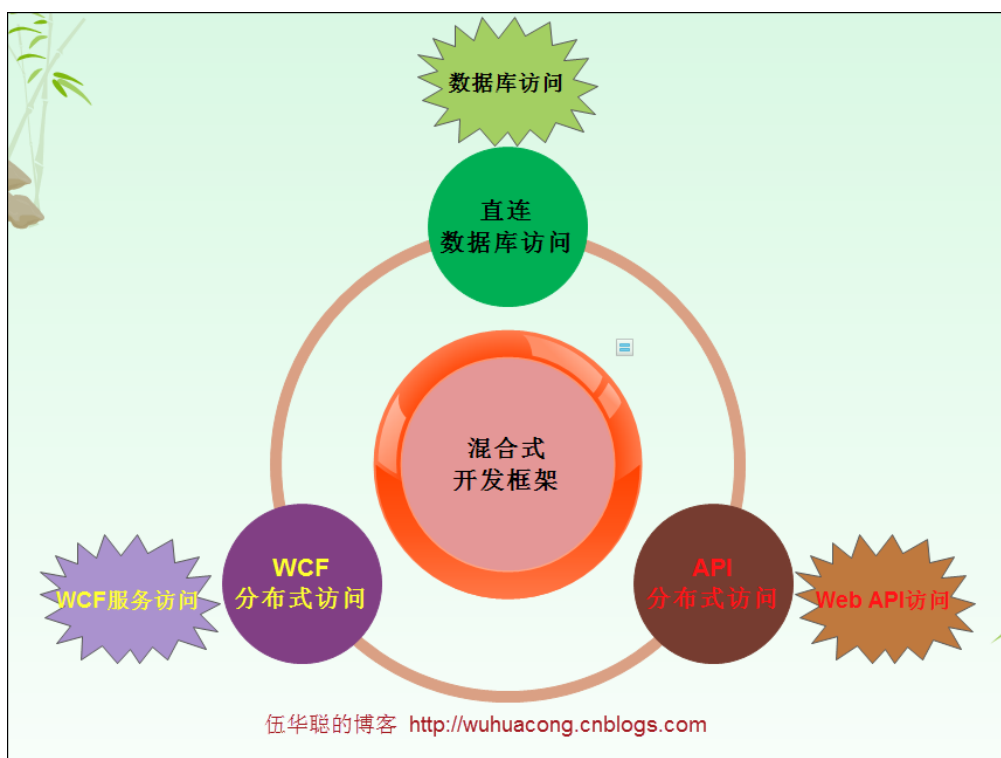


在当前大平台, 大应用的背景下, 可以基于一个整体的平台, 构建很多应用生态链, 这样就可以把 Web API 作为核心层, 可以在上面开发我们各种企业业务应用了。

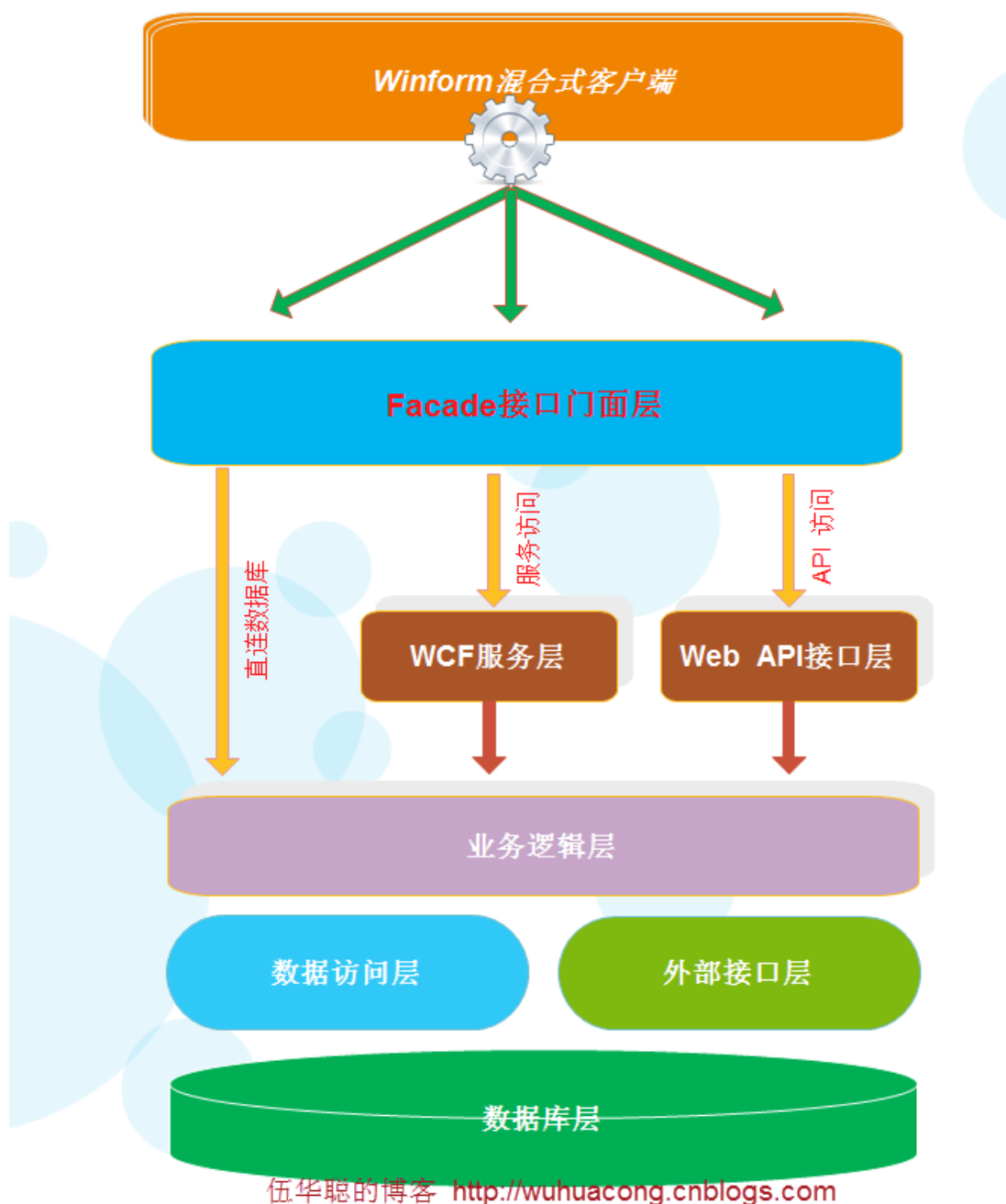


在 Winform 混合式框架的界面里面，我们除了可以利用直接访问数据库方式，以及采用访问分布式 WCF 服务的方式接入，还可以使得它能够访问 Web API 的数据服务，从而构建成一个适应性更加广泛、功能更加强大的**混合式开发框架**模式；对于 Web API，由于它提供的是一种无状态的接口访问，而且往往 Web API 一般为了多种客户端接入的需要，可能需要发布在公网上进行访问，因此我们需要更加注重 Web API 接口层的安全性。

除了直连数据库访问的传统模式，WCF 分布式访问的 WCF 服务访问模式，还可以接入 API 分布式访问的 Web API 接口模式，他们的关系构成了一个完整的 Winform 应用体系，如下图所示。



混合式框架的实现细节，就是通过一个类似开关模式的配置模块，确定是采用直接访问数据库方式，还是访问 WCF 服务的方式，它们两者是统一到一个 Facade 接口门面层上，如果考虑到 Web API 层，基于混合式的架构，也就是在这个 Facade 接口门面层上增加多一个 Web API 的接口的封装成即可。具体整个框架的架构图如下所示。



2.1.16. 特性总结

Winform 框架，本身就是为了能够快速开发一个高效、稳定、美观大方、扩展性强的应用软件系统。因此我在自己十多年的共享软件开发生涯以及公司项目开发中，不断思考，精雕细琢，对很多重要的特性都进行了归纳和升华，吸收项目中好的闪光点，借鉴一些好的软件开发思路，力求把软件做的更好；在开发效率方面，除了开发一些常规通用的模块、在模块内部又充分考虑继承、重用的规则，还对大幅度提高效率的代码生成工具，根据 Winform

开发框架的实现思路和特点，进行了完善优化，使得无论在业务代码生成，还是在界面代码生成方面，均能把开发效率发挥到极致，希望整个 Winform 开发框架能够持续发挥它的魅力和吸引力，为更多的人带来希望，体验开发的乐趣。

对于《Winform 开发框架》、《WCF 开发框架》和《混合式开发框架》这三种框架，他们各自又有自己的特点，如《WCF 开发框架》和《混合式开发框架》这两者利用了 WCF 的技术，因此它们可以对公用服务和业务服务进行分开配置和处理，可以对 WCF 服务进行加密传输，体现了松耦合的应用集成；而《Winform 开发框架》则相对于其他两种框架，更易于理解；《混合式开发框架》则有着一统天下的特征，因为它是基于接口的调用，而非针对具体的业务实现，实现了各种模块的有效统一。