

FineUI

第三方组件及其应用分析设计

代 劲 赵显静 王 梅 何雨虹 著

清华大学出版社



FineUI 第三方组件及其应用分析设计

代 劲 赵显静 王 梅 何雨虹 著

清华大学出版社
北 京

内 容 简 介

第三方组件是程序开发工具的重要补充,能够弥补开发平台在应用功能上的不足或缺陷。

FineUI 作为 ASP.NET 开发平台的第三方组件,历经 8 年 120 多个版本的积累及沉淀,目前,在应用上已经非常成熟。

FineUI 具有丰富的组件库,拥有原生的 Ajax 支持和出色的 UI 效果。通过使用 FineUI,Web 程序设计人员无须拥有 JavaScript 和 ExtJS 的基础知识,就可以快速地开发出功能强大的 Web 应用;不仅能够极大地减少开发时间、降低开发成本,而且维护升级方便,具有良好的兼容性能。

本书围绕 FineUI,创建了一个基于 ASP.NET 平台的在线应用项目。通过针对该项目的分析、设计、代码实现、测试与部署等环节,使读者能够快速地掌握 FineUI 程序设计的技巧,更高效地进行项目设计与开发。

本书既适合具有一定 ASP.NET 基础知识的人员阅读,也可作为高等学校计算机相关专业 ASP.NET 开发平台课程的辅助书籍。此外,本书还可供程序员和编程爱好者作为实际工作中的参考手册。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

FineUI 第三方组件及其应用分析设计/代劲等著. —北京:清华大学出版社,2016
ISBN 978-7-302-44296-7

I. ①F… II. ①代… III. ①网页制作工具 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2016)第 157033 号

责任编辑:吴艳华

封面设计:杨玉兰

责任校对:宋延清

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:虎彩印艺股份有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:24.25 字 数:594 千字

版 次:2016 年 8 月第 1 版 印 次:2016 年 8 月第 1 次印刷

定 价:49.00 元

产品编号:069804-01

前 言

第三方组件是针对软件开发平台在应用功能上的不足或缺陷，由软件开发平台编制方以外的其他组织或个人开发的相关组件。第三方组件是程序设计开发的重要补充，能够提供更为丰富的 UI 设计控件、更为强大的功能组件。

在 ASP.NET 程序开发中，常用的第三方组件很多，如 Spread、Developer Express、DevComponents、FineUI、NeatUpload、MyXls、Amcharts 等，为 ASP.NET 应用程序的开发提供了非常重要的补充与完善。

FineUI 是一款由国内程序员开发的 ASP.NET 第三方组件，历经 8 年 120 多个版本的积累及沉淀，目前在应用上已经非常成熟。

FineUI 具有以下一些特点。

基于 C#和 ASP.NET 2.0 开发：组件的命名、属性、方法、事件与原生的 ASP.NET 组件保持一致，开发人员无须学习更多的代码基础知识。

丰富的组件库及应用主题：FineUI 具有超过 50 个专业的 ASP.NET 组件，提供 4 种主题 UI 方案，同时，支持自定义主题来满足项目的个性化需求。在此基础上，不仅能快速实现业务逻辑，而且能够使页面设计及页面展现形式更加丰富多彩。

轻量级的数据传输：FineUI 采用透明的 Ajax 数据更新模式，不需要任何额外的代码，即可实现页面的局部更新。此外，独创的 XState 页面数据保存及传输机制，极大地减少了数据传输量。

跨浏览器支持：FineUI 使用 C#代替传统的 JavaScript 开发，不需要 JavaScript 代码就能完成真正的 Web 2.0 应用，不仅提高了开发效率，而且还可以确保在所有主流的浏览器下流畅地运行、维护和升级。

丰富的示例教程且开源免费：FineUI 提供了超过 100 个示例和丰富的在线教程。此外，可以免费应用于任何非商业或者商业性的项目。

本书从 ASP.NET 项目开发入手，围绕需求分析、功能设计、数据库设计等方面，从项目分析设计、代码实现、测试与部署等环节展开，将 FineUI 第三方组件的学习融入项目的具体实现中。通过对本书的学习，读者将熟练掌握 FineUI 第三方组件在 ASP.NET 项目开发中的应用，提高快速开发 Web 程序的能力。

本书在撰写过程中，得到了重庆邮电大学“软件工程专业卓越工程师”项目、重庆重邮信科(集团)股份有限公司“文峰大学英语云端学习平台”孵化项目资金的支持，促使本书能尽快地展现给广大读者朋友。限于作者本身的学识水平及应用设计能力，本书难免存在不妥或值得商榷之处，敬请广大读者批评指正。

本书适合具有一定 ASP.NET 基础知识的人员阅读，可作为高等学校计算机相关专业 ASP.NET 开发平台课程的辅助书籍。此外，本书也可供程序员和编程爱好者作为实际工作中的参考手册。

目 录

第一篇 初识 FineUI

第 1 章 ASP.NET 常用的第三方组件.....3	2.3 FineUI 使用示例..... 14
1.1 Developer Express3	2.3.1 从当前窗口弹出对话框..... 14
1.2 Ribbon Designer.....5	2.3.2 从顶层窗口弹出对话框..... 15
1.3 FineUI7	2.3.3 FineUI 综合示例..... 16
第 2 章 FineUI 介绍.....10	2.4 常用的 FineUI 组件..... 18
2.1 初识 FineUI10	2.4.1 表单组件..... 18
2.1.1 FineUI 发展历史10	2.4.2 表单布局..... 32
2.1.2 FineUI 的构成10	2.4.3 表格组件..... 40
2.1.3 FineUI 的运行机制11	2.4.4 窗体控件..... 80
2.2 FineUI 的配置与运行11	2.4.5 内联框架..... 91

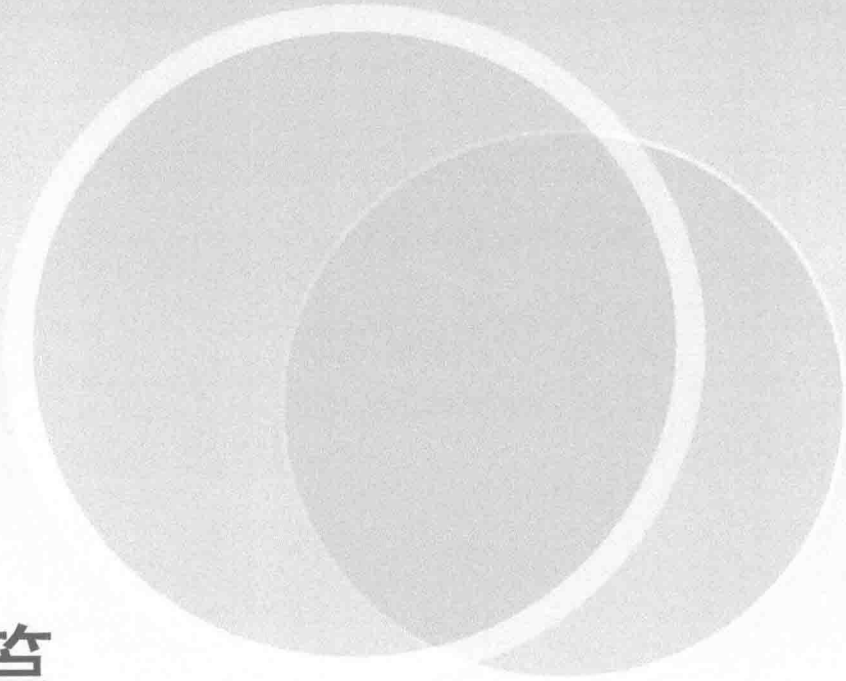
第二篇 FineUI 项目实战

第 3 章 项目需求分析101	5.3.2 自定义函数的设计 141
3.1 项目背景介绍101	5.3.3 存储过程的设计 145
3.2 系统功能需求101	5.4 数据库的保护 148
第 4 章 总体设计114	5.4.1 数据库备份和还原 148
4.1 系统目标114	5.4.2 故障恢复 150
4.2 开发环境114	第 6 章 代码实现..... 152
4.3 网站功能结构115	6.1 用户管理 152
4.3.1 普通游客功能115	6.1.1 用户添加 152
4.3.2 学生用户功能115	6.1.2 用户管理 159
4.3.3 教师用户功能117	6.2 用户登录 173
4.3.4 管理员用户功能121	6.3 作业管理 176
4.4 业务流程125	6.3.1 作业发布 176
第 5 章 数据库设计129	6.3.2 作业维护 194
5.1 概念设计130	6.3.3 作业详情 196
5.2 逻辑结构设计133	6.3.4 作业批改 204
5.3 物理结构设计135	6.4 文件管理 213
5.3.1 系统的表结构135	6.5 题库管理 216
	6.5.1 频道管理 216

6.5.2 领域管理.....	221	6.7 新闻管理.....	268
6.5.3 题库管理.....	223	6.8 自主学习.....	277
6.6 数据统计.....	249	6.8.1 学习计划管理.....	277
6.6.1 用户使用清单统计.....	249	6.8.2 完成学习计划.....	294
6.6.2 作业安排统计.....	252	6.9 讨论与交流.....	310
6.6.3 作业成绩清单.....	253		
6.6.4 作业情况分析.....	254	第 7 章 项目测试与部署	331
6.6.5 部门题库统计.....	254	7.1 项目的测试.....	331
6.6.6 题库清单.....	255	7.1.1 测试的重要性.....	331
6.6.7 教师题库统计.....	256	7.1.2 测试的基本概念.....	332
6.6.8 自主式学习统计.....	257	7.1.3 测试计划.....	332
6.6.9 自主式学习详细清单统计.....	258	7.1.4 测试结果分析.....	345
6.6.10 按学校用户使用统计.....	258	7.2 项目发布与部署.....	345
6.6.11 按部门用户使用统计.....	259	7.2.1 在 Windows 7 中配置 IIS 7....	345
6.6.12 用户使用清单.....	260	7.2.2 发布网站.....	346
6.6.13 按用户使用统计.....	261	7.2.3 部署网站.....	347
6.6.14 图表统计.....	261		

第三篇 FineUI 拾遗

第 8 章 FineUI 组件索引	353	10.3 在 Web.config 中配置 FineUI.....	369
8.1 FineUI 的表单组件.....	353	10.3.1 设置 configurations.....	369
8.2 FineUI 的表格组件.....	355	10.3.2 设置 system.web.....	370
8.3 其他组件.....	355	10.3.3 常见问题列表.....	370
第 9 章 NeatUpload 第三方组件	357	第 11 章 FineUI 支持的 Ajax 属性	373
9.1 NeatUpload 组件的介绍.....	357	第 12 章 ViewState 与 XState	376
9.2 NeatUpload 组件的安装及使用.....	357	12.1 ViewState 在 ASP.NET 控件中的	
9.2.1 获取 NeatUpload 组件.....	357	作用.....	376
9.2.2 配置 NeatUpload 组件.....	358	12.2 FineUI 早期使用 ViewState 的	
9.2.3 NeatUpload 组件的使用.....	359	情况.....	377
第 10 章 Web.config 的详细配置	361	12.3 FineUI 中的 XState 机制.....	380
10.1 Web.config 文件介绍.....	361	参考文献	382
10.2 Web.config 配置详解.....	361		



第一篇

初识 FineUI

FineUI 是一组基于 ExtJS 的 ASP.NET 第三方专业组件库，拥有原生的 Ajax 支持和华丽的 UI 效果。借助于 FineUI，Web 程序设计人员无须拥有 JavaScript 和 ExtJS 的基础知识，即可快速开发出功能强大的 Web 2.0 应用。

使用 FineUI 不仅能够极大地减少开发时间，降低开发成本，而且维护升级方便，具有良好的兼容性，在绝大部分主流浏览器下均能流畅运行。

第 1 章 ASP.NET 常用的第三方组件

本章要点：

- 第三方组件的概念。
- 第三方组件的优势。
- 常用的第三方组件。

第三方组件，从通常的意义上来说，是针对某种软件在应用功能上的不足或缺陷，而由软件编制方以外的其他组织或个人开发的相关组件。这种组件不是官方提供的，也不是由软件使用者编写的组件。用成熟的第三方组件进行开发，会提高开发的效率，减少出错的可能。作为对软件开发的支撑，组件是用户可与之交互以输入或操作数据的对象，通常出现在对话框中，或出现在工具栏上。

第三方组件的范围很广。简单的第三方组件有按钮、文本框等，复杂的第三方组件可以包括报表组件、表格组件和文字处理组件等，也会有一些组件套包，是由一系列组件组成的，通常包括表格、报表、图表、菜单、工具栏、数据输入验证等，特点是包含的组件数量比较多。

当然，使用第三方组件也存在一定风险：通常会有版本上的问题，或是要遵循某个开发协议，甚至会有 Bug。如果出现 Bug，会由于没有源代码，无法排除。即使有源代码，因为组件比较庞大，所以排除 Bug 的难度也会比较高。

在 ASP.NET 程序开发中，常用的第三方组件有很多，如 Spread、Developer Express、DevComponents、DotNetBar 等。本章主要介绍三种常用的第三方组件：Developer Express、Ribbon Designer 和 FineUI。

书中还用到了 NeatUpload、MyXls、Amcharts 等第三方组件，本章不再赘述。

1.1 Developer Express

Developer Express(以下简称 Dev)是 Visual Studio 开发平台中自带的组件。下载地址为 <https://www.visualstudio.com/en-us/>，最新版本为 Developer Express .NET v8.2。

1. 什么是 Developer Express

Dev 是 Visual Studio 2012(以下简称 VS2012)自带的 DataGridView 控件重写版本的第三方组件。Dev 包括较多优秀产品，有套包，也有子组件。Dev 产品不仅功能丰富、应用简便，而且界面华丽，更可方便地定制，对于 .NET 编程人员来说是个不错的选择。它的菜单栏组件更具代表性，完全可以替代开发环境提供的基本组件，让编写的程序或软件更显专业性。它还提供完善的帮助系统，资料详尽，可以快速入手。

Developer Express 的界面如图 1.1 所示。

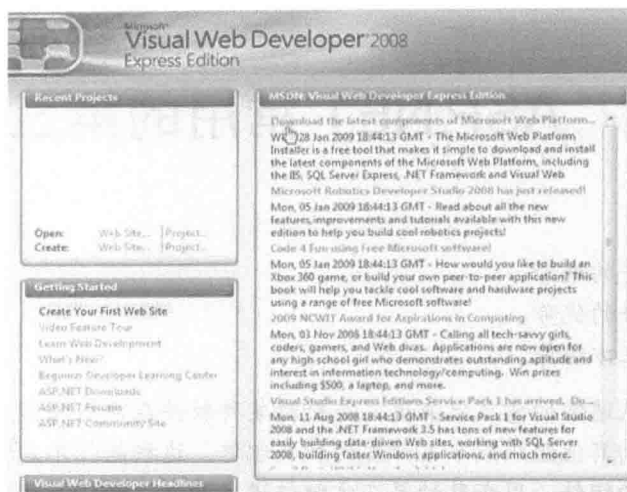


图 1.1 Developer Express 的界面

2. Developer Express 的优势

(1) 提供精美的界面设计功能。

Dev 界面精美，而且提供强大的自定义功能，用于 UI 设计时，展示效果很华丽。

(2) 内嵌功能强大。

对于很多传统 WinForm 项目难以展示的功能，Dev 都可以实现，特别是在图表展示方面，可以实现丰富多彩的效果，如图 1.2 和 1.3 中所展示的网站。

(3) 产品免费。

该产品只需要开发者授权(部署授权免费)便可部署。一个授权(Developer License)允许一个工程师安装使用，开发出来的程序可免费部署，不再收取任何费用。

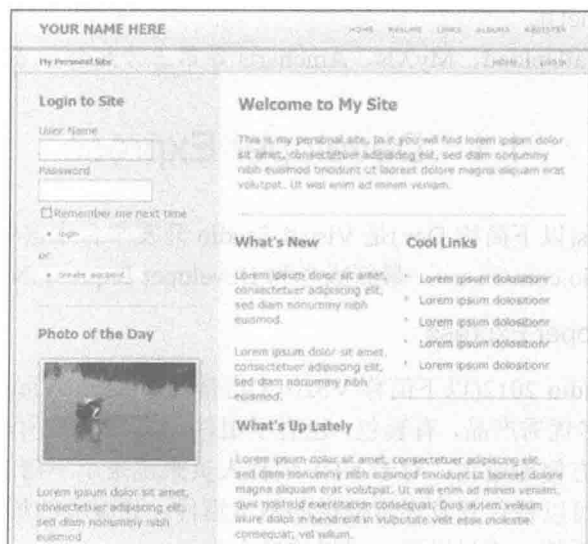


图 1.2 用 Developer Express 做的个人网站



图 1.3 用 Developer Express 做的社团网站

(4) 高效和实用。

Dev 拥有大量的示例和帮助文档，开发者能够快速上手。在国内，Dev 拥有大量的用户群体。

此外，Dev 设计开发资料比较完善，交流方便，除了界面美观外，最大的优势就是使用简单，可缩短开发周期，节约项目成本。

除此而外，Dev 在国内最早流行，相较其他组件更为成熟稳定，兼容性更好。

1.2 Ribbon Designer

Ribbon Designer 第三方组件由白俄罗斯的一家公司开发，用于帮助用户对其 SharePoint 和 Office 365 解决方案提供完全的自定义框架。

下载地址为 <http://www.bcgsoft.com/doc/RibbonDesigner.htm>。

最新版本为 Ribbon Designer v1.6.257。

1. 什么是 Ribbon

Ribbon 即功能区，是一种以面板及标签页为架构的使用者界面(User Interface)，最早出现在 Microsoft Office 2007/2010/2013 的 Word、Excel 和 PowerPoint 等组件中。

在仪表板设计器中，功能区包含一些用于创建、编辑和导出仪表板及其元素的上下文工具，它是一个收藏了命令按钮和图示的面板，把命令组织成一组“标签”，每一组包含了相关的命令，每一个应用程序都有一个不同的标签组，展示了程序所提供的功能。

在每个标签里，各种相关的选项被组合在一起。

如图 1.4 和 1.5 所示为典型的 Ribbon 界面。

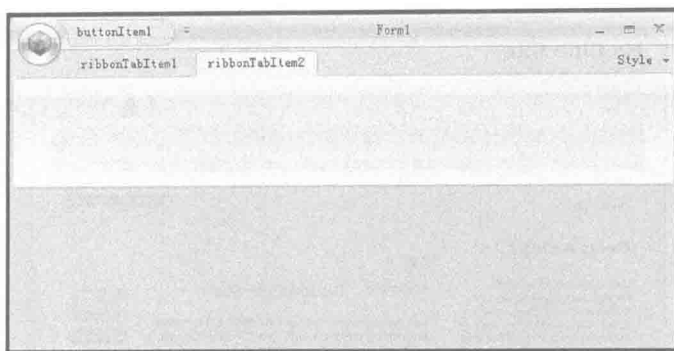


图 1.4 典型的 Ribbon 界面(白)

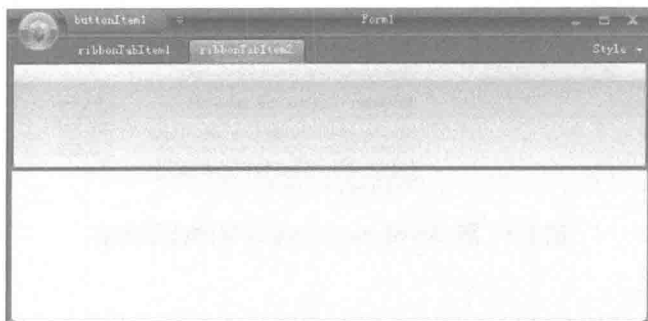


图 1.5 典型的 Ribbon 界面(黑)

2. Ribbon 的优势

与传统的菜单式用户界面相比，Ribbon 界面的优势主要体现在如下几个方面。

(1) 使用方便。

所有功能有组织地集中存放，不再需要查找级联菜单、工具栏等，还能更好地在每个应用程序中组织命令，提供足够显示更多命令的空间，如图 1.6 所示。

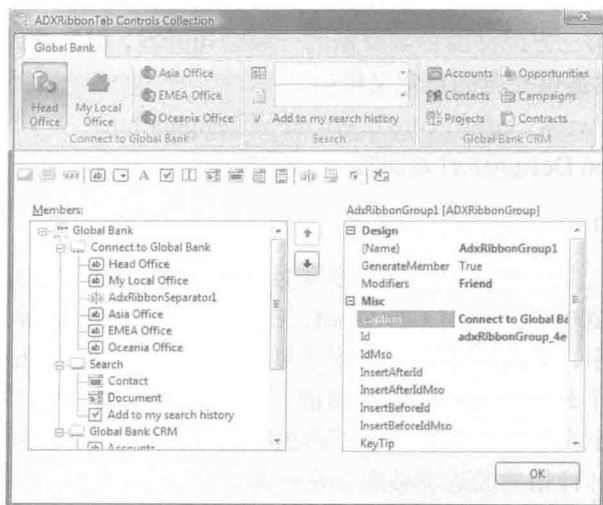


图 1.6 Ribbon 工具栏

(2) 效率高。

丰富的命令布局，可以帮助用户更容易找到重要的、常用的功能。可以显示图示，对命令的效果进行预览，例如改变文本的格式等，如图 1.7 所示。

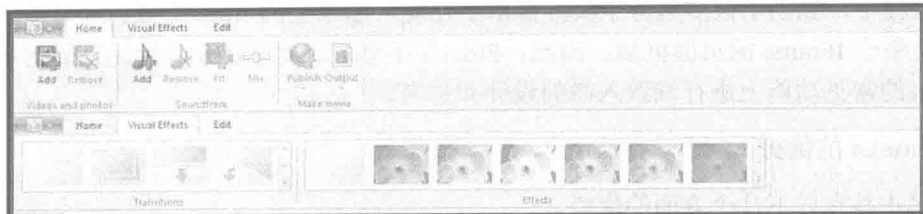


图 1.7 显示图示

(3) 更加适合触摸屏操作。

虽然从菜单式界面到 Ribbon 界面有一个比较漫长的熟悉过程，但事实是，Ribbon 界面正在被越来越多的人接受。相应地，越来越多的软件开发商开始抛弃传统的菜单式界面，转而采用 Ribbon 界面。

1.3 FineUI

FineUI 是由中国程序员三生石上和 Ringo Ding 创建并维护的，提供完整的 Web 2.0 开发设计解决方案。

下载地址为 <http://www.fineui.com/>。

最新版本为 FineUI(开源版) v4.2.1 和 FineUI(专业版) v2.8.0。

1. 什么是 FineUI

FineUI 是一组基于 ExtJS 的专业 ASP.NET 组件库，拥有原生的 Ajax 支持和华丽的 UI 效果，如图 1.8 所示。

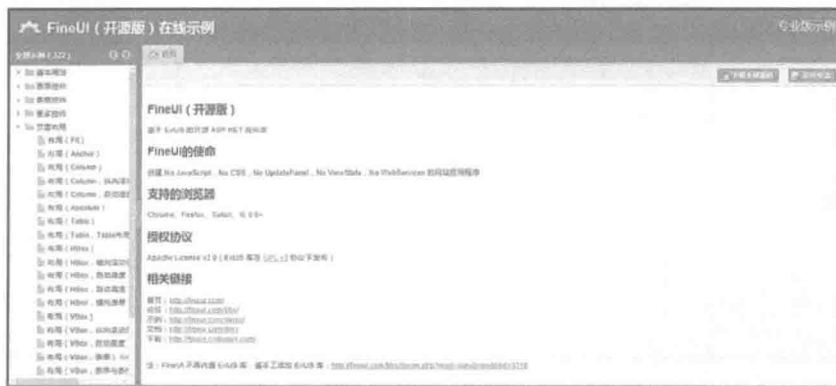


图 1.8 FineUI(开源版)在线示例

FineUI 的设计目的，就是让 ASP.NET 开发人员在不需要了解 JavaScript 和 ExtJS 的情况下，就能快速开发出漂亮的 Web 2.0 应用。因此，每个组件的编写都尽量符合 ASP.NET

的开发习惯。从组件和属性的命名，到复杂的 Ajax 交互，都让传统的 ASP.NET 开发人员感到非常便捷。

在开发模式上，FineUI 以管理项目为驱动，组件的开发优先级都是服务于项目需要的。在设计过程中，独创了很多服务于实际需求的功能，如原生的 Ajax 支持、轻量级的数据传输，以及基于 IFrame 的页面框架。因此，FineUI 不会追求对 ExtJS 进行完全的封装，但是会在 80% 的常见功能上进行细致入微的设计和思考。

2. FineUI 的优势

FineUI 具有以下几个方面的优势。

(1) 快速的项目开发。

基于 C# 和 ASP.NET 2.0 开发模式，不需要 JavaScript 代码就能完成真正的 Web 2.0 应用。不仅能够极大地减少开发时间和降低开发成本，而且维护升级方便。

(2) 基于现有的 ASP.NET 知识。

虽然 FineUI 是基于 ExtJS 组件的，但是，不需要用户事先了解 ExtJS 的知识。

FineUI 严格遵循 ASP.NET 的命名习惯和开发框架完成，组件的命名、属性、方法、事件与原生的 ASP.NET 组件保持一致。

(3) 丰富的组件库。

FineUI 提供了超过 50 个专业的 ASP.NET 组件，不仅能帮助开发者快速实现业务逻辑，而且会让页面更加生动和华丽。

(4) 跨浏览器支持。

FineUI 使用 C# 代替传统的 JavaScript 开发，不仅能提高开发效率，而且可以确保在所有主流的浏览器下流畅运行。

(5) 多主题支持。

FineUI 不仅内置 3 种官方主题，而且通过示例，提供了 4 种扩展主题，同时，还允许开发者自定义自己的主题，来满足项目的个性化需求。如图 1.9 所示。



图 1.9 官方主题

(6) 页面无刷新。

FineUI 所有的页面回发都不会导致整个页面重新渲染，而只有部分页面得到更新。而这个 Ajax 过程对开发人员完全透明，可以像往常一样，在服务器端改变组件属性，而这个

修改会直接更新反映到前台页面，而不需要任何额外的代码。

(7) 轻量级的数据传输。

FineUI 独创的 XState 页面数据保存及传输机制，极大地减少了数据传输量。

(8) 丰富的示例教程。

超过 100 个示例和丰富的在线教程，让开发者可以全面掌握 FineUI 的各种用法。

(9) 开源免费。

FineUI 可以免费应用于任何非商业或者商业性的项目，采用对商业应用友好的 Apache License 2.0 协议。

第 2 章 FineUI 介绍

本章要点:

- 了解 FineUI。
- FineUI 的配置和运行。
- 常用的 FineUI 组件。

2.1 初识 FineUI

FineUI 是使用比较广泛的 ASP.NET 第三方专业组件,它具有使用方便、开发效率高、界面丰富的特点,不仅能够极大地减少开发时间,降低开发成本,而且维护升级方便,在绝大部分主流浏览器下均可保障良好的兼容性能。

2.1.1 FineUI 发展历史

FineUI 由若干规模较小的第三方组件产生,并且不断完善。发展历程如下。

(1) 2008 年 4 月 11 日,第一个版本的 ExtAspNet(FineUI 的曾用名)诞生,其中只包含 7 个简单的表单组件。

(2) 2009 年 3 月 25 日,ExtAspNet(FineUI)发布了该项目的最后一个版本 v1.3.1,此时,这个版本已经被应用于多个企业的实际项目,并产生了很好的经济效益。

(3) 2012 年 11 月 18 日,ExtAspNet 正式改名为 FineUI,并发布 v3.2.0。

(4) 2012 年 12 月, FineUI 发布首款商业软件 FineUI.Design,为 FineUI 提供 VS 的设计支持。

(5) 2013 年 10 月 26 日, FineUI 推广上线,商业软件 FineUI.Design 可以开始被购买。

(6) 2014 年 7 月 21 日,发布 FineUI v4.1.0 正式版,这是 FineUI(开源版)的最稳定版本,同时,也是专业版的兼容版本。

(7) 2015 年 10 月 22 日,同时发布了 FineUI(开源版) v4.2.1 和 FineUI(专业版) v2.8.0。

2.1.2 FineUI 的构成

FineUI 是将 ExtJS 和 ASP.NET 进行深度整合后的产物。通过两者的优化组合,具有以下显著优点:大量地基于 JavaScript 和 CSS,却并不需要程序员编写其代码;拥有原生的 Ajax 支持,却并不需要依赖于 ASP.NET 的 UpdatePanel;可获得快速的 ASP.NET 开发模式,却并不需要 ViewState 的参与。

学习 FineUI 只需要有 ASP.NET 的开发基础,不需要事先学习 ExtJS,原因在于 FineUI 已经是一个深度封装的组件库,绝大部分操作均实现了模块化与可视化。

2.1.3 FineUI 的运行机制

在传统 ASP.NET 项目开发中，使用 ViewState 机制在回发过程中保存组件数据。

ViewState 是 ASP.NET 的一个非常重要的特性，用来在页面回发过程中维护组件的服务器端状态，这样，我们就能方便地在按钮的单击事件处理函数中编写如下代码了：

```
Label11.Text = TextBox1.Text;
```

对 ViewState 的描述可以简单概括如下。

(1) 页面第一次加载时，服务器组件把自身的属性保存在页面上的一个隐藏字段 VIEWSTATE 中。

(2) 页面PostBack时,表单提交的同时, __VIEWSTATE 的值也被提交到后台,ASP.NET 会从这个隐藏字段中恢复组件属性。

在传统的网站开发中，ViewState 的这一工作原理无可厚非，并且极大地简化了程序员的工作量。但是，随着 Ajax 的兴起，特别是在通过 JavaScript 渲染的客户端界面中，ViewState 的实现显得捉襟见肘。

FineUI 弃用了 ViewState，转而采用自己实现的 XState 机制：在不放弃页面回发中 C# 操作服务器组件便利的同时，拒绝了任何形式数据传输的浪费。

XState 是 FineUI 进行页面保存及数据传输的重要机制。目前, XState 已经在 FineUI 组件内部完全取代了原生的 ViewState, 避免了重复数据的传输, 极大地减少了网络流量。

2.2 FineUI 的配置与运行

FineUI 的官方下载地址是 <http://fineui.com/>。在安装 FineUI 之前，首先要确认已经安装了 Visual Studio。FineUI 的配置步骤如下。

(1) 将 Visual Studio 打开(以 2012 版为例),新建一个 ASP.NET Web 应用程序项目,如图 2.1 所示。



图 2.1 创建 ASP.NET Web 应用程序项目