
第 24 章 广告模块设计

广告能够为网页带来很多的增色功能效果和盈利，广告模块的设计对网站来说非常重要，一个网站不可能只有一个广告或者网站的广告还需要手动增加和删除。广告模块需要随机的获取系统广告或者能够在相应的位置增加广告来实现更多广告效果。

24.1 学习要点

广告模块需要涉及到一些 ASP.NET 3.5 的基本知识，如果要仔细学习广告模块的开发，需要详细了解本书的一些章节知识，这些章节如下所示：

- ☐ ASP.NET 的网页代码模型。
- ☐ Web 窗体基本控件。
- ☐ 数据库基础。
- ☐ ADO.NET 常用对象。
- ☐ Web 窗体数据控件。
- ☐ ASP.NET 内置对象。
- ☐ 用户控件
- ☐ 自定义控件

广告模块制作的是一个或多个自定义控件，这样在多个不同的页面中就能够快速的使用控件进行广告开发，在基本了解了以上章节的知识点后，就能够熟练学习和开发此模块。

24.2 系统设计

广告系统是一个非常重要的系统，随着网站的发展，如果网站需要实现盈利，就可以通过发布和获取广告信息来得到更多的盈利。管理人员希望在后台管理中，添加广告信息，包括文字广告和图片广告，这些广告能够按照一定的顺序随机的进行展现，并且当用户访问网站时，应该能够在页面中找到相应的广告并且点击广告。

24.2.1 模块功能描述

在网站系统的发展过程中，广告投放是必不可少的一部分，因为现在的绝大部分网站都需要通过广告来盈利。不仅如此，广告还能够绝佳的展现网站现有的信息，如果网站是一个商城类型的网站，那么广告在这个时候还能够为自己的网站展现广告以达到宣传自己网站的目的。

广告的展现过程需要分几个类型的广告展现，最常见的是文字广告，文字广告是最常用也是最基本的广告类型，但是文字广告比较多时会引起用户的反感。除了文字广告还有图片广告，图片广告能够引起网站用户或者是消费者的兴趣，因为图片广告一目了然并且容易被网站用户发现和查看，如果图

片广告制作的比较精良，那么会极大的提高用户的兴趣并提高点击量。在广告模块的设计中，考虑到有不同的广告类型和展现方式，通常情况下有以下三种广告展现方式。

- ❑ 文字广告：仅向用户展现文字广告。
- ❑ 图片广告：仅向用户展现图片广告。
- ❑ 图文广告：随机的向用户展现文字或图片广告。

文字广告和图片广告都是单纯的广告形式，只向页面中展现文字或者图片，提供一个超连接即可，而图文广告能够随机的展现文字广告或图片广告，当用户将页面刷新或者缓存更新时，对用户展现的广告是不同的。

由于广告模块是网站的管理者进行发布的，开发人员可以将广告模块制作成为自定义控件，使用自定义控件可以允许管理者或开发人员进行管理或发布，对于不同的广告类型，可以考虑不同的自定义控件。而对于管理员而言，广告发布流程是比较简单的，如图 24-1 所示。

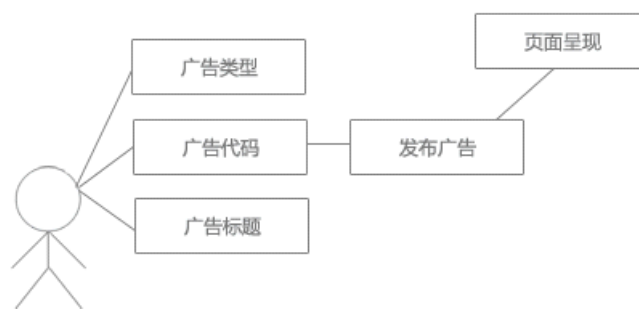


图 24-1 广告发布流程

正如图 24-1 所示，对于管理人员并无需做复杂的操作，管理人员只需要在后台编写相应的广告类型、广告代码和广告标题等广告模块需要的字段，然后进行广告的发布就能够在页面呈现了，但是对于开发人员来说，要让这个过程能够顺利的进行需要在页面呈现中进行筛选。如果是文字广告，就需要筛选出文字广告和广告说明，如果是图片广告，就需要筛选出图片和超链接，对于不同的广告类型开发人员必须筛选出不同的广告并呈现在页面中。所以对于开发人员可以选择如下两种方案进行广告模块的开发。

- ❑ 自定义控件：开发人员可以为不同的广告类型进行自定义控件的开发，对于页面编程人员可以拖动自定义控件到相应的位置进行广告的呈现，但是这样制作就有可能在页面中呈现多个相同的广告。
- ❑ 固定投放位置：开发人员可以固定投放位置，例如网站头部广告、网站底部广告和网站侧面广告，虽然这样做能够降低网站的重复广告的频率，但是这样制作无疑只能固定死网站的 HTML 代码，也显得不够灵活。

虽然网站广告投放没有最好的解决方案，通常开发人员也会通过页面的修改进行网站广告的投放，但是这里还是选择一个折中的方案，就是自定义控件的开发，自定义控件的开发可以通过编程的方法在页面中进行广告的控制和筛选，相比固定投放位置而言，在维护过程中更加的方便。从上述流程中可以基本规划几个自定义控件：

- ❑ 文字广告自定义控件：专门用于呈现文字广告自定义控件。
- ❑ 图片广告自定义控件：专门用于呈现图片广告自定义控件。
- ❑ 图文广告自定义控件：专门用于呈现图文广告自定义控件。
- ❑ 高级广告呈现控件：可以通过属性进行控制广告的呈现。

- ❑ 广告发布页面：管理员可以通过该页面进行广告发布。
- ❑ 广告呈现页面：管理员发布的广告能够在一个或多个页面进行呈现。
- ❑ 广告管理页面：管理员能够为不需要使用的广告进行管理。

通过编写多个自定义控件进行广告控制，也可以通过编写一个广告控件进行广告控制，同时管理员能够方便在后台进行广告发布并能够轻松的呈现在前台页面。

24.2.2 模块流程分析

在对业务进行了基本的划分之后，可以为模块进行基本的流程分析，包括这个模块中最基本的函数，以及这些函数在页面中是如何执行的。其中广告模块中需要开发广告发布页面，广告发布页面能够让管理员快速的发布广告，并选择广告发布的类型和位置，这样就能够轻松进行广告的发布，同时管理员也应该能够进行广告的管理，包括修改和删除，如图 24-2 所示。



图 24-2 管理员发布流程

管理员能够在广告后台发布广告、修改广告和删除广告，当对广告进行修改后可以选择对广告再次进行投放，投放完成后就能够在页面进行呈现。对于上述流程可以分别开发若干个页面进行功能整合，如图 24-3 所示。

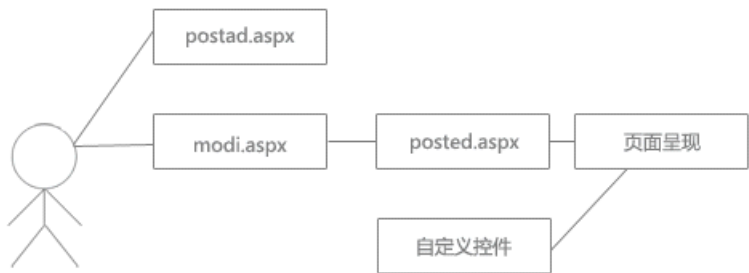


图 24-3 管理员操作页面划分

管理员在后台只需要进行广告管理等操作，而对于页面呈现，只需要从自定义控件中选取相应的数据并通过编程的方法进行整合呈现即可，而其中最重要的是如何进行页面呈现，只有灵活的将同类广告进行页面呈现才能够避免重复广告的出现。

24.3 数据库设计

对于广告模块的数据库设计可能比较的复杂，其复杂并不在数据库设计的本身上，数据库本质上就是一种存储数据的容器，而如何进行数据筛选在广告模块中是一个最为重要的过程，对于数据库的设计就需要考虑到广告模块中的数据筛选。

24.3.1 数据库设计分析

对于广告模块的数据库设计，需要加强数据条目的筛选功能，例如数据库中对广告的类型进行筛选，以选择不同类型的广告的不同呈现方式。在广告设计中，需要设计 3 个表，这三个表分别为 ads、type 和 adclass。其中 ads 表用于存放广告数据，其字段如下所示。

- ❑ 广告编号：表示广告 ID 号，为自动增长的主键。
- ❑ 发布时间：表示广告发布的时间。
- ❑ 结束时间：表示广告发布结束的时间。
- ❑ 广告名称：作为广告的标识而存在，用于表示广告的名称。
- ❑ 广告内容：作为广告的内容而存在，可以是文字也可以是 HTML 代码。
- ❑ 广告备注：作为广告的备注而存在，用于标识备注信息。
- ❑ 广告图片：作为图片广告的图片连接。
- ❑ 广告连接：作为外部连接的广告的地址。
- ❑ 广告标题：作为广告的标题。
- ❑ 广告 html：作为广告呈现的 HTML 代码，可以为 JavaScript 代码。
- ❑ 广告类型：作为广告的类型而存在，类型没描述在 type 表中。
- ❑ 聚合类型：作为广告的广告 ID 而存在，用于归纳同类广告。

其中 type 表用于存放广告的类型数据，其字段如下所示。

- ❑ 分类编号：表示广告类型的 ID 号，为自动增长的主键。
- ❑ 分类名称：表示广告类型的描述，例如文字、图片等。

其中 adclass 表示广告显示的类型，使用 ads 表的 adid 表示可以表示广告在页面中呈现的归纳，其字段如下所示。

- ❑ 聚合分类编号：表示广告类型的 ID 号，为自动增长的主键。
- ❑ 分类名称：表示广告存放类型的描述，例如头部广告、底部广告。

对于广告模块来说，其数据表比较多，为了方便维护和扩展，就必须要让一些需要长期修改的字段进行外部连接。这样就能够极大的加强数据库中数据的健壮性和低耦合性。

注意：良好的数据库设计可能需要同时创建多个表进行一个功能的描述，虽然在数据库设计时这样的方法比较麻烦，但是在维护和开发中，这样会带来很多的便利。

24.3.2 数据库表的创建

创建表可以通过 SQL Server Management Studio 视图进行创建也可以通过 SQL Server Management Studio 查询使用 SQL 语句进行创建。广告模块需要创建多个表进行广告的描述，在创建表之前首先需要创建一个 ad 数据库，数据库创建完成后就能够在数据库中创建表了。这里首先需要创建一个 ads 表，该表用于存储广告模块中的广告信息，如图 24-4 所示。

列名	数据类型	允许空
id	int	<input type="checkbox"/>
time	datetime	<input checked="" type="checkbox"/>
endtime	datetime	<input checked="" type="checkbox"/>
name	nvarchar(50)	<input checked="" type="checkbox"/>
[content]	nvarchar(MAX)	<input checked="" type="checkbox"/>
infor	nvarchar(MAX)	<input checked="" type="checkbox"/>
picture	nvarchar(500)	<input checked="" type="checkbox"/>
url	nvarchar(500)	<input checked="" type="checkbox"/>
title	nvarchar(500)	<input checked="" type="checkbox"/>
html	nvarchar(MAX)	<input checked="" type="checkbox"/>
type	int	<input checked="" type="checkbox"/>
adid	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

图 24-4 ads 表结构

正如图 23-4 所示，其中的字段意义如下所示。

- ❑ id: 表示广告的 ID 号，为自动增长的主键。
- ❑ time: 用于标识广告的开始时间
- ❑ endtime: 用于标识广告的结束时间，当时间到达该时间后，广告将不再被呈现。
- ❑ name: 用于标识广告的名称，这个名称在后台管理中可以进行辨认。
- ❑ content: 作为广告的内容而存在，管理员能够在该字段进行广告内容的编写。
- ❑ infor: 作为广告的备注而存在，管理员和管理员之间能够通过备注阅读该广告是什么广告。
- ❑ picture: 作为图片广告的图片连接。
- ❑ url: 作为外部连接的广告的地址，用户单击广告时能够跳转到相应的连接。
- ❑ title: 作为广告的标题，呈现在页面之中。
- ❑ html: 作为广告呈现的 HTML 代码，可以为 JavaScript 代码，当广告为文字广告时，将呈现 HTML。
- ❑ type: 作为广告的类型而存在，类型没描述在 type 表中。
- ❑ adid: 作为广告的广告 ID 而存在，用于归纳同类广告，一个页面可以呈现一种或多种类型的广告。

上述字段描述了相应的字段在实际应用中的意义，创建表的 SQL 语句如下所示。

```
USE [ad]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ads](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [time] [datetime] NULL,
    [endtime] [datetime] NULL,
    [name] [nvarchar](50) COLLATE Chinese_PRC_CI_AS NULL,
    [content] [nvarchar](max) COLLATE Chinese_PRC_CI_AS NULL,
    [infor] [nvarchar](max) COLLATE Chinese_PRC_CI_AS NULL,
    [picture] [nvarchar](500) COLLATE Chinese_PRC_CI_AS NULL,
    [url] [nvarchar](500) COLLATE Chinese_PRC_CI_AS NULL,
    [title] [nvarchar](500) COLLATE Chinese_PRC_CI_AS NULL,
    [html] [nvarchar](max) COLLATE Chinese_PRC_CI_AS NULL,
    [type] [int] NULL,
    [adid] [int] NULL,
    CONSTRAINT [PK_ads] PRIMARY KEY CLUSTERED
)
--创建 ads 表
```

```
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

上述代码创建了一个 ads 表用于存储广告数据，其中的 type 字段和 adid 字段都是其他表的外键，这三个表一起完成整个广告模块的数据描述，type 表创建的 SQL 语句如下所示。

```
USE [ad]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[type](
[id] [int] IDENTITY(1,1) NOT NULL,
[classname] [nvarchar](50) COLLATE Chinese_PRC_CI_AS NULL,
CONSTRAINT [PK_type] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] //创建 type 表
```

type 表用于描述广告的类型，而 adclass 表用于描述广告呈现的类型，这两个表是有区别的。type 主要描述的是广告的类型，包括图片广告、文字广告等，是系统类型，通常情况下是不会更改的。而 adclass 用于描述的是广告呈现时所需要的类型，例如头部广告和底部广告，这些广告通过 adclass 表进行筛选和整合。adclass 表创建的 SQL 语句如下所示。

```
USE [ad]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[adclass](
[id] [int] IDENTITY(1,1) NOT NULL,
[classname] [nvarchar](10) COLLATE Chinese_PRC_CI_AS NULL,
CONSTRAINT [PK_adclass] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] //创建 adclass 表
```

上述代码创建了一个 adclass 表，使用该表能够将广告进行分类并呈现到相应的页面中，可以极大程度的避免同种类型的广告的呈现。数据库的设计是非常重要的，也是在软件开发过程中一个非常重要的环节。在广告模块中，必须先规定好，以及规划好广告模块的数据库设计，否则数据库的更改会带来很多的不便，例如如果将 adclass 表和 type 表整合在 ads 表中，如果要修改一个字段的值，例如修改图片类型的广告，有可能需要更改一个或多个数据，这样就非常的不方便，也会导致数据的混乱，所以数据库设计在任何模块甚至是系统的开发过程中都是非常重要的一个环节。

24.4 界面设计

对于广告模块的界面设计，并不像前面两个模块一样对界面的要求很高，也同样没有对用户体验进行要求。但是广告模块的界面设计也并不是很简单，由于广告骄傲模块的呈现需要使用自定义控件进行 HTML 代码的呈现，其界面设计反而要求开发人员有较熟练的 HTML 编码能力。

24.4.1 发布广告界面

发布广告界面作为管理员进行广告发布的页面，这个页面无需特别复杂的呈现，因为管理员最终期望的是能够快速地进行广告的发布，而不是花哨的界面，不过虽然这样，还是需要进行一定的用户体验的开发，发布广告界面代码见光盘中源代码第 24 章\24-1\24-1\Postad.aspx 所示。

其中的代码编写了广告信息的基本控件，管理员能够填写相应的广告信息用于广告的认识。在广告发布中，还需要填写广告发布代码，以及图片连接用于高级的广告信息的呈现，示例代码见光盘中源代码\第 24 章\24-1\24-1\Postad.aspx 所示。

在该页面中，使用了若干控件，这些控件都分别为广告中的数据输入进行准备，这些控件包括 TextBox 文本框控件、日历控件和下拉菜单控件。下拉菜单作为数据绑定控件用于数据绑定，提供给管理人员选择相应的广告分类。

24.4.2 发布广告页数据源配置

在发布广告页面中使用了数据源控件进行数据源的呈现。在页面中，需要对数据源进行配置、筛选和生成能够在发布页面中进行数据选择。单击【配置数据源】按钮，选择【新建连接】选项，在新建连接窗口中进行数据源配置，如图 24-5 所示。

拖放一个数据源控件到页面，用于配置 adclass 数据连接和数据绑定，创建数据连接后，选择【将数据连接保存到 Web.config】选项，在项目里就可以使用该连接进行数据连接和绑定，如图 24-6 所示。

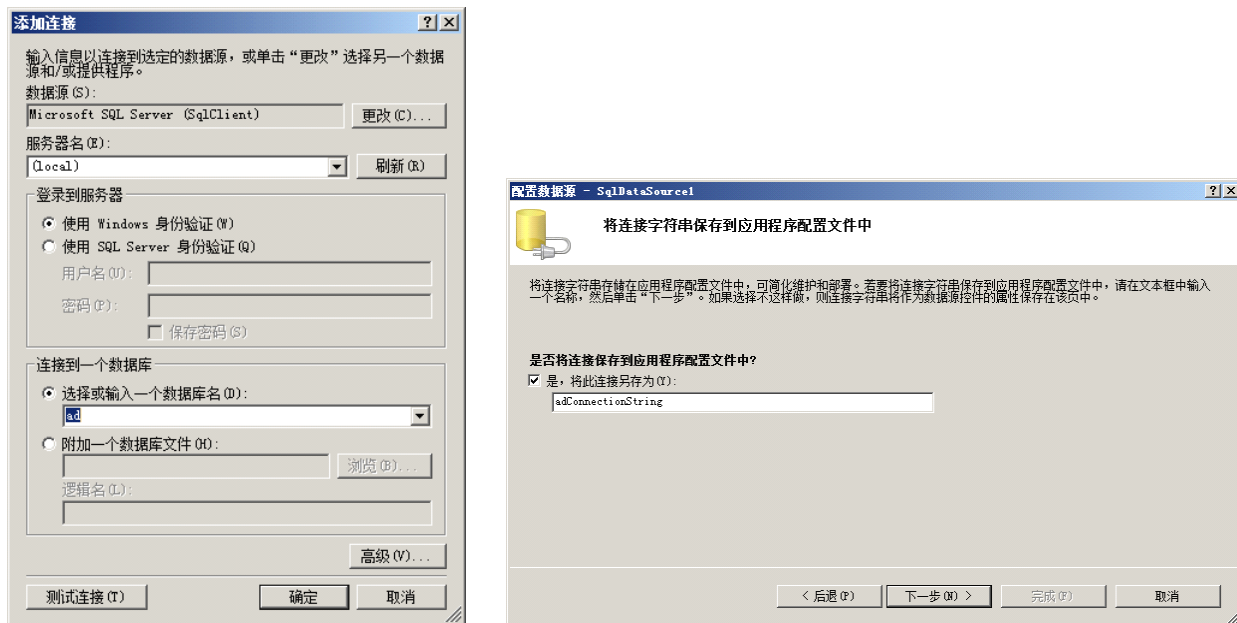


图 24-5 创建新数据连接

图 24-6 创建连接

创建连接后, 就可以自动生成 SELECT 语句填充数据绑定控件, 方便开发, 如图 24-7 所示。在完成 SELECT 语句的配置, 就可以在相应的控件中使用数据源呈现的数据, 例如在广告类型的下拉菜单中就可以使用数据源控件进行数据显示, 如图 24-8 所示。

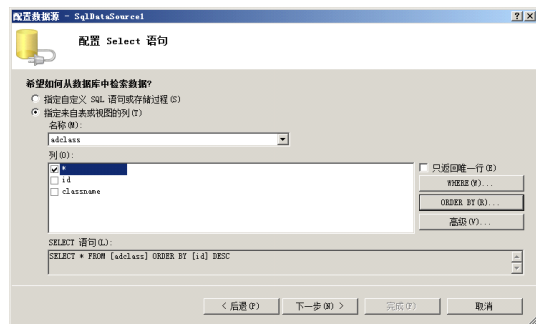


图 24-7 配置 select 语句



图 24-8 选择数据源呈现数据

在选择了数据源之后, 就可以为另一个下拉菜单进行数据源配置, 配置过程与上面的代码相同。配置完成后, 页面增加了数据源控件的代码和数据绑定控件代码, 示例代码见光盘中源代码\第 24 章\24-1\24-1\Postad.aspx 所示。

24.4.3 修改广告界面

修改广告界面同发布广告界面相同, 但是修改广告界面在加载时必须接受一个传递的参数 id 来查询相应的广告信息, 加载完成后就要填充到修改广告页面的控件中。这也就是说, 当页面加载时, 加载之后的修改广告页面应该先获取广告信息提供给管理人员修改, 修改广告界面代码见光盘中源代码\第 24 章\24-1\24-1\modi.aspx。

修改广告界面基本同添加广告界面相同, 因为修改广告界面只需要进行广告的读取和修改即可, 而广告中所需修改的字段同广告添加字段基本相同, 所以在广告修改中只需要进行字段的显示和更新就能够实现广告修改页面的制作。

24.4.4 管理广告界面

管理广告界面可以使用现有的 ASP.NET 数据源控件和 ASP.NET 数据绑定控件实现, ASP.NET 数据源控件和数据绑定控件能够快速提供数据的更新、删除等功能。由于这里使用的是自定义更新页面, 就不能够使用数据源控件本身提供的数据更新功能, 对于管理广告界面, 只需要进行数据删除操作的支持即可, 数据源示例代码见光盘中源代码\第 24 章\24-1\24-1\Manage.aspx。

上述代码配置了数据源控件的高级模式以支持数据绑定控件中的更新、删除等操作, 这里只需要使用删除操作就能够实现广告的管理, 更新操作无需使用自带的更新而使用自定义页面。单击【数据绑定】控件, 在菜单中单击【功能模块】按钮, 选择【添加新列】选项, 在【选择字段类型】选项中选择【HyperLinkFiled】选项并填写 HyperLinkFiled 类型字段中提供的相应的数据列和数据显示策略, 如图 24-9 所示。

在数据绑定控件中能够使用【更新】连接进行页面跳转功能的实现, 如图 23-10 所示, 其中就包括了系统自带的删除操作和开发人员自定义的更新操作。

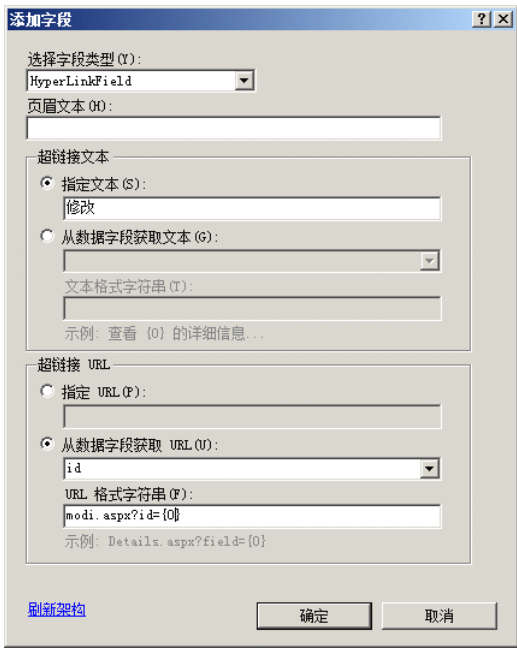


图 24-9 添加字段

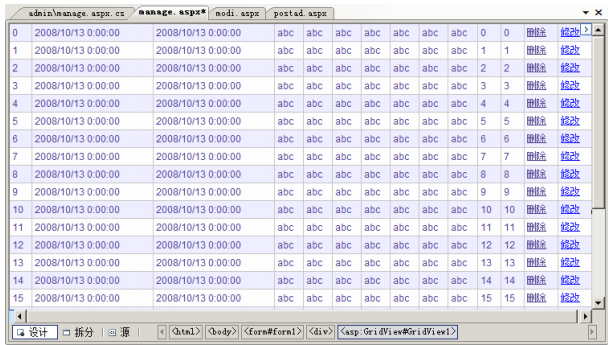


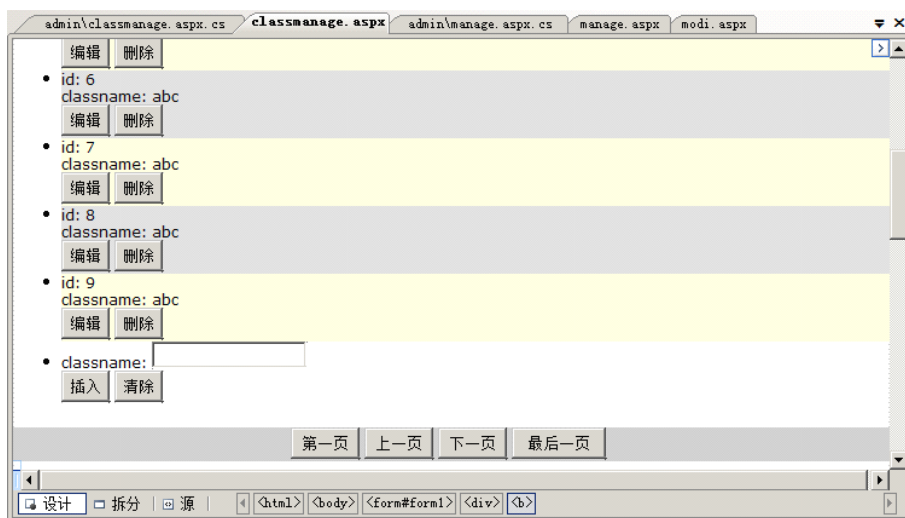
图 24-10 数据绑定控件 Grid View

其中数据绑定控件选择了自动套用格式让管理界面看上去更加的友好，管理人员能够在该界面查看相应的广告信息并且删除相应的信息，如果管理人员要修改相应的数据，可以单击【修改】按钮在自定义页面中进行广告的修改。

24.4.5 分类管理界面

分类管理界面比较的简单，因为分类管理表中的字段非常的少，所以分类管理界面就能够使用现有的控件，如 Grid View 控件进行数据插入、删除和更新，在分类管理界面中，可以直接使用控件进行操作，这样就能够在多个页面进行复杂的管理，示例代码见光盘中源代码\第 24 章\24-1\24-1\ClassManage.aspx。

上述代码使用了 ListView 控件并自动套用格式，使管理员在操作的时候更加方便和简单，ListView 控件能够直接进行数据的插入、更新和删除，更加简便的进行了数据管理，如图 24-11 所示。



与 24-11 分类管理页面效果

分类管理页面是广告模块中一个比较容易实现的模块，但在功能上却是非常重要的模块，因为在广告的分类管理是非常重要的，在自定义控件的开发过程中，可以通过广告的分类管理进行广告的筛选，以及整合，通过广告的分类可以在网站的不同页面进行不同的广告呈现，以及不同广告的筛选，避免了广告的重复。

24.5 代码实现

虽然控件为开发提供了良好的支持，但是控件毕竟样式死板、界面布局有限，而且代码实现也有限，所以很多情况下都需要使用自定义页面进行应用程序的开发，使用控件虽然能够方便和快速的进行功能开发，但是却无法避免死板的界面布局和有限的功能。

24.5.1 广告添加功能

广告添加功能可以使用 ADO.NET 进行广告添加，ADO.NET 可以执行 INSERT 语句进行数据库中的数据插入，在广告添加页面，管理员在填写完相应的项目时，可以单击按钮控件进行数据插入。在 postad.aspx 页面中，制作完成页面并双击【控件】按钮，Visual Studio 2008 能够自动生成相应的事件，开发人员可以在该事件中使用 ADO.NET 进行数据操作代码的编写，示例代码如下所示。

```
protected void Button1_Click(object sender, EventArgs e)
{
    try
    {
        SqlConnection con = new SqlConnection("Data Source=(local);Initial Catalog=ad;Integrated
                                                Security=True");           //创建连接
        con.Open();                                           //打开连接
        string strsql = "insert into ads (time,endtime,name,content,infor,picture,url,title,html,type,adid)
                        values ('" + Convert.ToDateTime(Calendar1.SelectedDate).ToString() + "','" +
                        + Convert.ToDateTime(Calendar2.SelectedDate).ToString() + "','" +
                        + TextBox1.Text + "','" + TextBox3.Text + "','" + TextBox4.Text + "','" +
                        + TextBox6.Text + "','" + TextBox7.Text + "','" + TextBox2.Text + "','" +
```

```

        TextBox5.Text + "','" + DropDownList1.Text + "','" + DropDownList2.Text + "');";
        SqlCommand cmd = new SqlCommand(strsql, con);           //创建执行
        cmd.ExecuteNonQuery();                                   //执行 SQL
        Response.Redirect("manage.aspx");                         //页面跳转
    }
    catch(Exception ee)
    {
        Response.Write(ee.ToString());                           //抛出异常
    }
}

```

广告添加过程非常的容易，正如上述代码所示，直接对数据库中的数据进行插入操作就能够插入一条新广告，对于自定义控件，可以从数据库中获取广告和筛选广告进行呈现。

24.5.2 广告修改功能

广告修改页面是广告模块中的自定义页面，这个页面使用的是控件进行组合开发，当页面被加载时，首先需要通过传递的参数进行查询，查询后填充到控件中，示例代码如下所示。

```

protected void Page_Load(object sender, EventArgs e)
{
    try
    {
        if (!IsPostBack)                                         //判断加载
        {
            if (Request.QueryString["id"] == "")                 //获取参数
            {
                Response.Redirect("manage.aspx");                 //页面跳转
            }
            SqlConnection con =
            new SqlConnection("Data Source=(local);Initial Catalog=ad;Integrated Security=True");
            con.Open();                                           //打开连接
            string strsql = "select * from ads where id=" + Request.QueryString["id"].ToString() + "";
            SqlDataAdapter da = new SqlDataAdapter(strsql, con);    //创建适配器
            DataSet ds = new DataSet();                           //创建数据集
            int count = da.Fill(ds, "table");                     //填充数据集
            if (count > 0)                                         //判断数据
            {
                TextBox1.Text = ds.Tables["table"].Rows[0]["name"].ToString(); //初始化控件
                TextBox2.Text = ds.Tables["table"].Rows[0]["title"].ToString(); //初始化控件
                TextBox3.Text = ds.Tables["table"].Rows[0]["content"].ToString(); //初始化控件
                TextBox4.Text = ds.Tables["table"].Rows[0]["infor"].ToString(); //初始化控件
                TextBox5.Text = ds.Tables["table"].Rows[0]["html"].ToString(); //初始化控件
                TextBox6.Text = ds.Tables["table"].Rows[0]["picture"].ToString(); //初始化控件
                TextBox7.Text = ds.Tables["table"].Rows[0]["url"].ToString(); //初始化控件
                Calendar1.SelectedDate = ds.Tables["table"].Rows[0]["time"].ToString(); //初始化控件
                Calendar2.SelectedDate = ds.Tables["table"].Rows[0]["endtime"].ToString();
                DropDownList1.Text = ds.Tables["table"].Rows[0]["type"].ToString(); //初始化控件
                DropDownList2.Text = ds.Tables["table"].Rows[0]["adid"].ToString(); //初始化控件
                Label1.Text = ds.Tables["table"].Rows[0]["id"].ToString(); //初始化控件
            }
        }
    }
}

```

```

        else
        {
            Response.Redirect("manage.aspx");           //页面跳转
        }
    }
}
catch
{
    Response.Redirect("manage.aspx");           //错误页跳转
}
}

```

当页面被加载时就会执行上述代码，上述代码仅仅是在数据库中查询相应的数据，并呈现在相应的控件中，这样管理员在加载页面时就能够修改现有的数据内容进行更新。当管理员更新完毕后，单击相应的事件按钮就能够执行数据更新操作，数据更新操作示例代码如下所示。

```

protected void Button1_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection("Data Source=(local);
                                           Initial Catalog=ad;Integrated Security=True");           //创建连接字符串
    con.Open();                                           //打开连接
    string strsql = "update ads set time=" + Calendar1.SelectedDate + ",endtime=" +
        Calendar2.SelectedDate + ",name=" + TextBox1.Text + ",title=" + TextBox2.Text +
        ",content=" + TextBox3.Text + ",infor=" + TextBox4.Text + ",html=" + TextBox5.Text +
        ",picture=" + TextBox6.Text + ",url=" + TextBox7.Text + ",type=" + DropDownList1.Text +
        ",adid=" + DropDownList2.Text + " where id=" + Label1.Text + """;           //更新 SQL
    SqlCommand cmd = new SqlCommand(strsql, con);           //创建执行
    cmd.ExecuteNonQuery();           //执行 SQL 语句
    Response.Redirect("manage.aspx");           //页面跳转
}

```

上述代码通过 ADO.NET 进行数据更新，从上面代码可以看出使用 ADO.NET 进行数据更新非常的简单，只需要打开数据库连接，在连接过程中使用 SqlCommand 对象执行 ExecuteNonQuery 方法进行数据的插入和删除操作就能够对数据库进行操作。

24.5.3 自定义控件的实现

在增加广告、删除广告和广告管理等页面制作完毕后，这也就意味着后台基本制作完毕，管理员可以在后台进行广告的增加和删除以及管理，也可以对广告类别进行管理。后台制作完毕后就需要在前台呈现广告，前台广告的呈现可以通过制作自定义控件进行呈现。

右击现有解决方案管理，在下拉菜单中选择【添加新项】选项，在【添加新项目】窗口中选择【自定义控件】项，这里创建一个名为 Ad 的自定义控件，如图 24-12 所示。

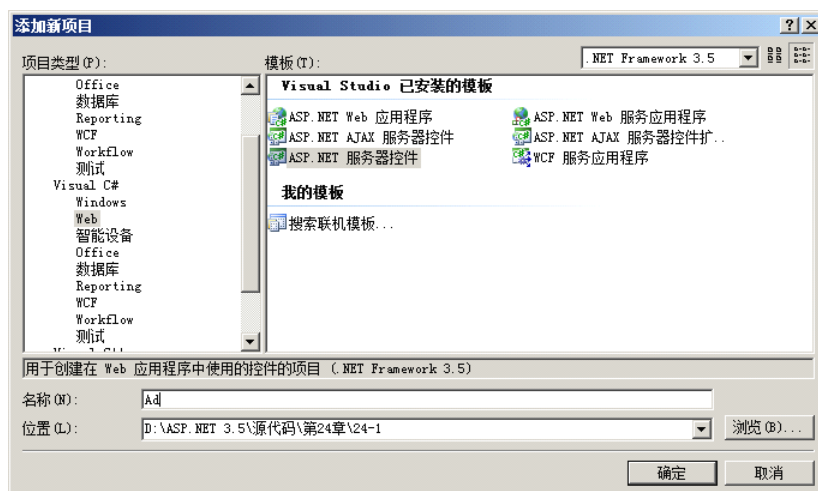


图 24-12 新增自定义控件

自定义控件用于筛选广告和呈现广告，筛选过程可以使用自定义控件的属性和方法完成，在编写完成自定义控件之后，就能够通过向页面拖动自定义控件和属性配置进行广告呈现，自定义控件属性编写示例代码如下所示。

```
[Bindable(true)]                                     //设置允许绑定
[DefaultValue("")]                                   //默认值为空
[Localizable(true)]                                  //允许本地化
public string type                                     //设置广告类型
{ get; set; }

[Bindable(true)]                                     //设置允许绑定
[DefaultValue("")]                                   //默认值为空
[Localizable(true)]                                  //允许本地化
public string adid
{ get; set; }

[Bindable(true)]                                     //设置连接字符串
[DefaultValue("Data Source=(local);Initial Catalog=ad;Integrated Security=True")]
[Localizable(true)]
public string SqlConnectionString
{ get; set; }

[Bindable(true)]                                     //设置允许绑定
[DefaultValue("")]                                   //默认值为空
[Localizable(true)]                                  //允许本地化
public bool text                                       //设置是否为文字
{ get; set; }

[Bindable(true)]                                     //设置允许绑定
[DefaultValue("")]                                   //默认值为空
[Localizable(true)]                                  //允许本地化
public string CssStyle                                //设置 CSS 样式
{ get; set; }

[Bindable(true)]                                     //设置允许绑定
[DefaultValue("")]                                   //默认值为空
[Localizable(true)]                                  //允许本地化
public string TitleCssStyle                           //设置标题 CSS
{ get; set; }
```

[Bindable(true)]	//设置允许绑定
[DefaultValue("")]	//默认值为空
[Localizable(true)]	//允许本地化
public string ContentCssStyle	//设置内容 CSS
{ get; set; }	
[Bindable(true)]	//设置允许绑定
[DefaultValue("")]	//默认值为空
[Localizable(true)]	//允许本地化
public int ShowNumber	//设置显示个数
{ get; set; }	

上述代码为自定义控件设置了属性，管理员可以使用此控件并设置属性进行控件的编写和调用相应的广告代码，自定义控件 HTML 页面实现代码如下所示。

```
protected override void RenderContents(HtmlTextWriter output)
{
    try
    {
        string constring = "Data Source=(local);Initial Catalog=ad;Integrated Security=True";
        if (SQLConnectionString != null) //获取连接字符串
        {
            constring = SQLConnectionString;
        }
        SqlConnection con = new SqlConnection(constring); //创建连接对象
        con.Open(); //打开连接
        string strsql = "select * from ads order by id desc"; //默认 SQL 语句
        if (type != null && adid != null) //筛选 SQL 语句
        {
            strsql = "select * from ads where type='" + type + "' and adid='" + adid + "' order by id desc";
        }
        else if (type != null) //筛选 SQL 语句
        {
            strsql = "select * from ads where type='" + type + "' order by id desc";
        }
        else if (adid != null) //筛选 SQL 语句
        {
            strsql = "select * from ads where adid='" + adid + "' order by id desc";
        }
        SqlDataAdapter da = new SqlDataAdapter(strsql, con); //创建适配器
        DataSet ds = new DataSet(); //创建数据集
        int count = da.Fill(ds, "table"); //填充数据集
        if (count > 0) //判断项数
        {
            if (ShowNumber < count) //判断生成条目
            {
                count = ShowNumber; //获取用户设置
            }
            StringBuilder build = new StringBuilder(); //创建 String 对象
            //开发人员可以在这里使用属性中的样式
            build.Append("<div style='padding:10px 10px 10px 10px;border:1px dashed #ccc;'>");
            for (int i = 0; i < count; i++) //遍历输出
```



```
        {
            build.Append("<div style=\"font-size:14px;border-bottom:1px dashed #ccc;\">
            <a href=\"\" + ds.Tables[\"table\"].Rows[i][\"url\"].ToString() + \"\">\" +
            ds.Tables[\"table\"].Rows[i][\"title\"].ToString() + \"</a></div>\"");
            build.Append("<div
            style=\"font-size:12px;\">\" + ds.Tables[\"table\"].Rows[i][\"content\"].ToString() +
            \"</div>\"");
            //输出 HTML
        }
        build.Append("</div>\"");
        Text = build.ToString();
        //输出 HTML
        //呈现广告内容
    }
    else
    {
        Text = "暂时没有任何投放的广告";
        //提示没广告
    }
    output.Write(Text);
    //输出 HTML
}
catch(Exception ee)
{
    Text = ee.ToString();
    //抛出异常
    output.Write(Text);
    //输出异常信息
}
}
```

上述代码通过配置不同的属性进行 SQL 语句生成，例如当用户配置了 type 属性和 adid 属性，SQL 语句就能够生成筛选 type 和 adid 属性的 SQL 语句，当开发人员使用自定义控件时，只需要在页面中拖动相应的控件并配置相应的属性即可。

在需要使用广告控件的页面中，必须要引用控件，单击需要引用的项目，单击右键，单击添加引用，在项目选项卡中选择项目或在浏览选项卡中选择相应的 DLL 文件就能够添加引用，引用添加完毕后能够通过控件拖动生成广告代码，如图 24-13 所示。



图 24-13 自定义控件

当开发人员使用自定义控件并拖动到页面时，可以配置一些基本属性就能够呈现不同的广告类型。开发人员还能够进行多次开发和扩展进行自定义控件编程，通过获取和判断数据库中的字段进行不同类型广告的呈现。

24.6 实例演示

使用广告模块能够快速的进行广告的投放和使用，开发人员还能够使用自定义控件在页面中进行自定义控件的属性配置以呈现不同的广告的显示形式。作为管理人员可以直接在后台进行广告发布，在发布之前，需要创建一些广告类型，如图 24-14 和图 24-15 所示。

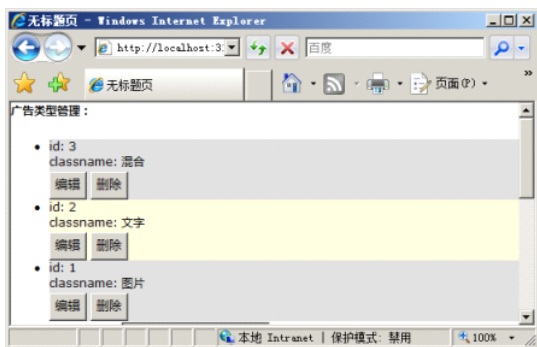


图 24-14 创建广告类型

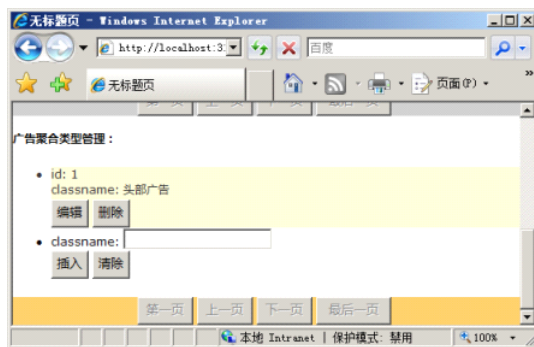


图 24-15 创建广告聚合类型

广告类型一般都不会更改，广告类型包括文字广告、图片广告和图文广告，开发人员可以配置属性显示文字广告、图片广告还是混合广告。如果显示的广告类型不同，那么呈现在相应页面的样式也不同，而创建广告聚合类型是能够避免广告的重复，例如选择头部广告、底部广告和侧边广告，可以选择一个广告只显示在头部而不显示在底部，避免了广告的重复。广告分类添加后就能够添加广告，如图 24-16 和图 24-17 所示。

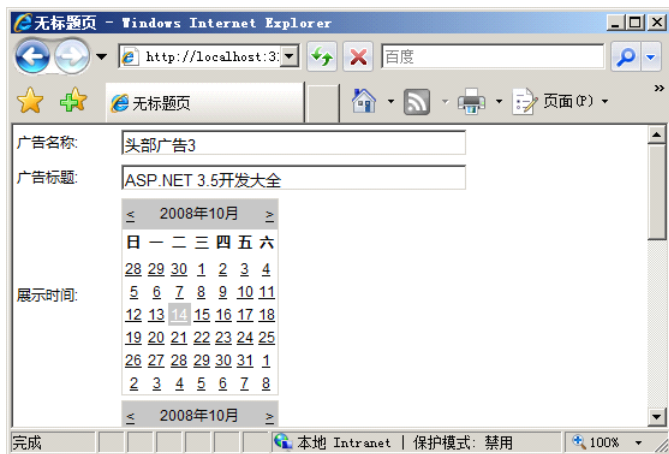


图 24-16 配置广告基本信息

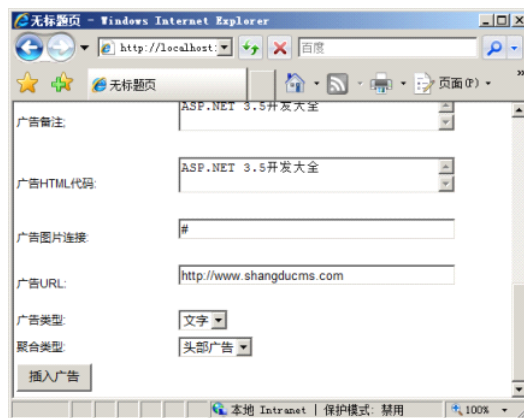


图 24-17 填写广告类型信息

插入完成后就能够在管理页面进行管理了，管理页面可以对广告进行修改、删除等操作，修改广告在自定义页面中进行修改，修改页面并没有使用现有的数据绑定控件进行修改，而是使用了自定义控件进行编程，通过使用 ADO.NET 获取数据并从数据中存放到自定义页面的控件中。

当管理员修改完成后，可以通过单击按钮进行数据更新，这时候广告就能够在各个页面被重新投放并且各个广告的风格和内容也会更改。如果开发人员为了提高广告的性能，可以在广告中使用缓存进行广告的缓存处理，限于篇幅的限制，这里就不再详细的讲解如何使用缓存进行性能提高。

在管理页面，管理员可以快速的删除广告信息，当管理员删除了广告信息后，广告信息就不会在相应的页面中进行投放，广告管理页面如图 24-18 所示。

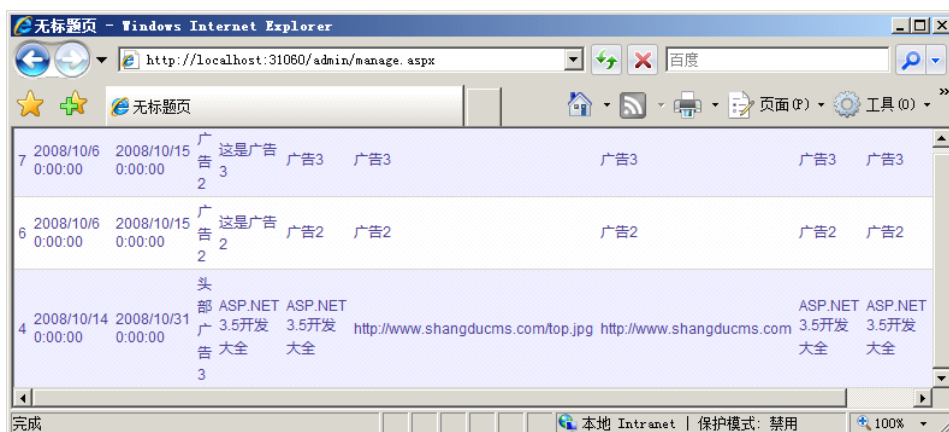


图 24-18 广告管理页面

开发人员可以选择修改和删除相应的信息以在页面呈现不同的广告内容，开发人员在相应的页面可以使用自定义控件进行广告呈现，示例代码如下所示。

```
<body>
  <form id="form1" runat="server">
    <div>
      <cc1:ads ID="ads1" runat="server" ShowNumber="10"/>           //ShowNumber 为必填项
    </div>
  </form>
</body>
```

上述代码仅仅是简单的将控件拖动到页面中，就能够呈现广告内容，这里呈现的是文字广告内容，运行后如图 24-19 所示。

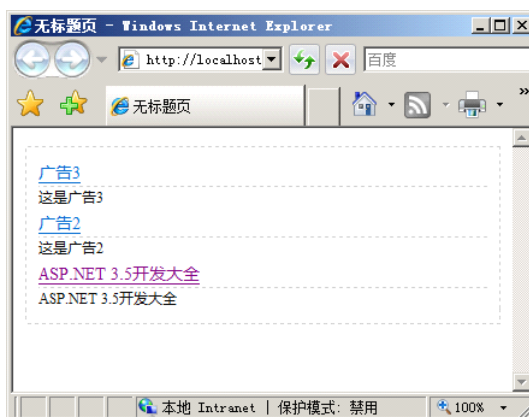


图 24-19 广告控件

当广告开发人员将广告控件拖放到页面中后，广告就能够呈现在相应的页面并提供给用户。管理人员在后台添加相应的类型的广告就能够在页面中呈现不同的广告内容。管理员能够配置自定义控件的属性进行不同的广告内容的控制，如果管理员将相应类型的广告全部删除，广告页面就会提示并没有呈现相应的广告。

由于篇幅的限制，这里的自定义控件并没有单独的制作图片广告的呈现方法，因为在图片文件的呈现和混合呈现的 HTML 代码生成中，需要筛选更多的数据或者生成更多不同种类的 HTML 代码，这里只是简单的进行了文字广告的呈现，在使用自定义控件的过程中，必须要配置相应的属性让控件能够使用数据库中的数据。

24.7 小结

本章通过编程的方法实现了广告模块的编写，广告模块是一个非常灵活和重要的模块，在网站的制作过程中，广告模块能够加强网站与用户的交互性本章在编写广告模块的过程中还巩固了：

- ❑ ASP.NET 的网页代码模型。
- ❑ Web 窗体基本控件。
- ❑ 数据库基础。
- ❑ ADO.NET 常用对象。
- ❑ Web 窗体数据控件。
- ❑ ASP.NET 内置对象。
- ❑ 用户控件
- ❑ 自定义控件

本章使用了自定义控件进行了扩展开发，自定义控件是一个难度比较高的知识点，虽然自定义控件的开发并不困难，但是对于初学者而言，自定义控件很容易让人变得混乱。自定义控件能够实现更多更强大的自定义功能，熟练的掌握自定义控件能够简化开发，提高应用程序的灵活性。