

# 用于创建 SharePoint 应用程序的技术（第二部分，共两部分）

## 目录

- [简介](#)
- [重要 SharePoint 构建基块](#)
- [SharePoint 开发人员构建基块的完整列表](#)
- [使用开发人员仪表板查看性能指标](#)
- [SharePoint 2010 辅助功能准则](#)
- [SharePoint 2010 开发工具](#)
- [设置 SharePoint 2010 开发环境](#)
- [SharePoint 2010 和应用程序生命周期管理 \(ALM\)](#)
- [结论](#)
- [其他资源](#)

## 简介

在本系列文章的上一篇文章中，我对 SharePoint 应用程序进行了定义，介绍了几个人们认为很有价值的 SharePoint 功能，讨论了开发人员方案，并提供了几种方案的示例。在本文中，我将讨论 SharePoint 开发人员构建基块。作为一名开发人员，您在为 Microsoft SharePoint 产品和技术生成应用程序时实际上开发的是什么呢？

## 重要 SharePoint 构建基块

尽管下面的表以字母顺序显示，但并非所有构建基块对您来说都具有相同的重要性。本节列出了几个因重要性较高而应单独标注的构建基块。

- 首先，您必须了解[服务器对象模型](#)。您在生成 Web 部件、包含自定义操作的工作流和事件接收器时会使用此编程接口。托管客户端对象模型基于服务器对象模型。您了解的大部分关于服务器对象模型的内容都适用于客户端对象模型。
- [Web 部件](#)是扩展 SharePoint 最常用的方法。学习生成 Web 部件是开始着手 SharePoint 开发的好方法。

- 利用[工作流](#)，您可以编纂和标准化业务流程，而且工作流也是实现某些方案的重要工具之一。如果使用代码扩展工作流，则应使用[服务器对象模型](#)。
- Microsoft [Business Connectivity Services \(BCS\)](#) 使 SharePoint 部署外部的数据可用于 SharePoint 应用程序。
- 了解协作应用程序标记语言 ([协作应用程序标记语言 \(CAML\)](#)) 和 [LINQ to SharePoint](#) 对于生成良好的 SharePoint 应用程序至关重要。
- [文档库](#)和[列表](#)不仅是 Microsoft SharePoint Foundation 2010 的最终用户功能，还是开发人员的编程资源。通常，您在设计软件系统时，必须存储用于驱动应用程序行为的配置信息。您可以在隐藏列表中存储这类信息，并在必要时限制用户访问。当您必须让非开发人员可以维护应用程序配置时，这种方法最适用。另一种方法是在一个隐藏的、访问受限的文档库中存储配置 XML 文件。

## SharePoint 开发人员构建基块的完整列表

下表列出了可用于构建 SharePoint 应用程序的 SharePoint 开发人员构建基块。

### SharePoint 2010 开发人员构建基块

构建基块	说明	SharePoint Foundation	SharePoint Server Standard	SharePoint Server Enterprise
<a href="#">Access Services</a>	向 SharePoint 网站发布 Microsoft Access 数据库			X
<a href="#">管理</a>	以编程方式管理 SharePoint Foundation 2010 和 Microsoft SharePoint Server 2010	X	X	X
<a href="#">通知</a>	发送电子邮件或短信	X	X	X

ASP.NET 应用程序页	使用 Microsoft ASP.NET 代码隐藏模型构建网页	X	X	X
Business Connectivity Services	将外部数据引入 SharePoint 部署	X	X	X
Business Connectivity Services 客户端 API	从客户端计算机访问 Business Connectivity Services 数据	X	X	X
协作应用程序标记语言 (CAML)	以 XML 形式表示的声明性查询语言	X	X	X
列类型和字段类型	为列表和库中的数据创建域定义	X	X	X
内容类型	定义列表项的原型	X	X	X
自定义操作	创建链接、工具栏按钮和菜单项	X	X	X
自定义视图	定义列表和文档库的视图	X	X	X
仪表板和 KPI	报告、图表和 KPI 的 Web 部件			X
文档库	任意类型的文档或文件的存储	X	X	X
文档元数据	高级文档分类		X	X
ECMAScript 客户端对象模型	从 ECMAScript (JavaScript、JScript) 访问 SharePoint 数据	X	X	X
事件接收器	提供更新列表或库时的行为	X	X	X
Excel Services	加载、计算和显示 Microsoft Excel 工作簿			X
Excel Services ECMAScript 接口	从 ECMAScript 访问 Excel Web Access Web 部件控件			X
Excel Services REST API	通过 URL 直接访问 Excel 工作簿文档部件或元素			X

Excel Services 用户定义函数	编写在服务器端电子表格中使用的高性能函数			X
Excel Web Services	使用 Web 服务以编程方式访问 Excel Services			X
功能	部署单元	X	X	X
运行状况规则	编纂在大型安装中监视运行状况的方法	X	X	X
InfoPath Forms Services	使 Microsoft InfoPath 2010 表单能够在浏览器中填写			X
LINQ to SharePoint	使用 LINQ 来查询 SharePoint 列表和文档库	X	X	X
列表	用户可编辑的数据存储	X	X	X
托管客户端对象模型	从客户端计算机访问 SharePoint 数据	X	X	X
移动页、控件和适配器	将 SharePoint 数据引入移动电话		X	X
OData REST API	使用 OData 建议的标准提供对 SharePoint 数据的 REST 访问	X	X	X
Office 客户端托管外接程序	增强 Office 客户端功能，使它能使用 SharePoint 数据和文档	X	X	X
Open XML SDK	将 Open XML 用于文档生成和转换	X	X	X
PerformancePoint Services	构建仪表盘、记分卡、报告和关键性能指标			X
PowerPivot	创建商业智能解决方案			X
沙盒解决方案	沙盒基础结构用于隔离和关闭错误的代码，以防止它降低服务器场的性能	X	X	X
搜索自定义	支持搜索各种内容，包括外部内容类型。		X	X

Secure Store Service	安全地访问旧版系统		X	X
安全性	确保敏感信息仅对授权用户可用	X	X	X
服务器对象模型	用于访问数据和内容的基于 .NET Framework 的服务器端 API	X	X	X
服务器功能区	在 SharePoint 中自定义功能区	X	X	X
服务应用程序体系结构	创建可由多个 SharePoint 应用程序使用的共享服务	X	X	X
SharePoint 对话框框架	用于创建对话框和其他客户端用户界面元素的 ECMAScript 编程接口	X	X	X
SharePoint 主题	将自定义品牌信息应用于网站或网站集	X	X	X
SharePoint Web Services	通过 Web 服务访问 SharePoint 数据	X	X	X
Silverlight Web 部件	创建外观更佳、使用更方便的用户界面	X	X	X
网站定义	创建、配置和设置 SharePoint 网站	X	X	X
网站页面	用户可自定义的页面，可节约资源	X	X	X
计时器作业	在特定时间或以特定间隔执行代码	X	X	X
变体	针对不同的文化、市场和语言定制网站		X	X
Visio Services	共享和查看 Microsoft Visio 绘图			X
Visual Studio SharePoint 工具扩展性	扩展 Microsoft Visual Studio 2010 以使您能够更快地创建自定义解决方案	X	X	X
基于 Web 的分布式创作和版本管理 (WebDAV)	有效地上载和下载文件	X	X	X
Web 部件	利用 Web 部件和 Web 部件页，用户可以自	X	X	X

	定义其用户界面。生成 Web 部件是扩展 SharePoint 最常用的方法。			
Web 服务	用于生成可扩展的分布式松散结合 SharePoint 应用程序的关键组件。	X	X	X
Windows Identity Foundation	根据受信任的提供程序提供基于声明的标识服务	X	X	X
Word Automation Services	提供将文档在各种文件格式之间转换的服务		X	X
工作流	使用任务和表单将业务流程自动化	X	X	X

### Access Services

可以使用 Internet 浏览器或 Access 客户端在 SharePoint Server 2010 中使用 Access Services 来编辑、更新和创建链接的 Access 2010 数据库。可以使用 Microsoft SQL Server Reporting Services 生成格式化的报告。

操作方法视频: [Creating Web Databases with Access 2010 and Access Services](#) (该链接可能指向英文页面)

视频: [Access 演示: Access Services 和 Web 数据库的 Access 2010 演示](#) (该链接可能指向英文页面)

白皮书: [使用 Microsoft Access Services 提高 Microsoft Access 2010 数据库应用程序的应用范围和可管理性](#) (该链接可能指向英文页面)

门户网站: [Access 2010 开发](#) (该链接可能指向英文页面)

SDK: [将 Access Services 与 SQL Reporting Services 配合使用: 安装 SQL Server 2008 R2 Reporting Services 外接程序 \(SharePoint Server 2010\)](#)

### 管理

[Microsoft.SharePoint.Administration](#) 命名空间的类为开发人员提供了以编程方式部署、迁移、维护和管理其 SharePoint 网站和基于 SharePoint 的解决方案的工具。

SDK: [SharePoint Foundation 管理](#)

SDK: **Microsoft.SharePoint.Administration**

SDK: [Microsoft SharePoint Foundation 的管理对象模型](#)

## 通知

通知提供了用于自动向 SharePoint 用户发送电子邮件或短信通知结果的框架。可以将通知附到列表或列表项。SharePoint Foundation 2010 提供了一个针对常用方案的通知模板库。

SDK: [构造块: 通知](#)

SDK: [SharePoint Foundation 中的通知](#)

SDK: [通知概述](#)

SDK: [Alert 对象模型](#)

## ASP.NET 应用程序页

若要构建具有特定应用程序逻辑 (使用 ASP.NET 代码隐藏模型通过 Microsoft Visual C# 或 Microsoft Visual Basic 编写) 的自定义用户界面, 可以使用 ASP.NET 应用程序页 (也称为 .aspx 页), 这是一种特殊的 ASP.NET 页。

您的 .aspx 页必须与 ASP.NET 母版页集成。利用母版页和 .aspx 页之间的关系, 您可以在各个 SharePoint 网站和网站集中提供一致的用户界面。

因为它们是可包含隐藏代码且经过预编译的传统 .aspx 页, 所以它们具有相对于 SharePoint 网页的性能优势。

您不能使用应用程序页来承载 Web 部件区域或 Web 部件。

.aspx 页的特征之一是提供了多个放置 Microsoft Silverlight 控件的位置的选项。对于某些用户, 可以只将 Silverlight 界面提供给网站的大型部件。但是, 网站维护人员可以使用默认 SharePoint 用户界面。

SDK: [母版页](#)

SDK: [网站和网页](#)

SDK: [SharePoint 页面类型](#)

MSDN: [Creating Application Pages for SharePoint](#)

SDK: [构建基块: 网页和用户界面](#)

SDK: [SharePoint Foundation 中级联样式表类的用法](#)

## Business Connectivity Services

**Business Connectivity Services** 提供了基础结构，以便您可以定义某些方法（使用特定原型），它还将数据引入到 SharePoint 中，以便用户可以直接从 SharePoint 查看和修改数据。外部表被称为 *外部内容类型* 或 *实体*。您可以将外部内容类型作为实体使用（如实体/关系理论中所述）。

**Business Connectivity Services** 提供了三个基本的机制，可以用于将外部数据引入到 SharePoint 中：

- 您可以连接到 SQL 数据库。默认情况下支持 SQL Server。只要多做尝试，就可以连接到 Oracle 或其他各种数据库服务器。
- 您可以使用某些 Web 服务，这些服务用于公开遵循方法原型的特定模式的方法。
- 您可以编写 .NET 托管代码（C# 或 Visual Basic）以连接到任意类型的数据。有两种方法可让您通过代码进行扩展：
  - 您可以编写 .NET 程序集连接器，让用户可以对预定义架构创建外部内容类型。该架构不能在不进行重新编译和重新部署的情况下做出更改。
  - 如果在包含自己的架构机制和定义的基础系统上构建，则可以编写一个自定义连接器，该连接器使用基础架构定义，并会让用户为基础架构中的任何表创建外部内容类型。这当然不只是编写 .NET 程序集连接器而已。

外部列表和外部数据列是内置用户界面功能的一部分，您可以通过它显示和维护外部内容类型中的数据。您可以使用 Web 接口、Microsoft SharePoint Designer 2010 或 Visual Studio 2010 创建外部列表。

您可以对搜索引擎公开外部内容类型，这样用户便可以在这些类型中查找数据。默认情况下将在配置文件页中显示数据。您可以自定义该页面或自定义用户在单击搜索结果中的项目时会发生的情况。例如，可以使用非 SharePoint 网站来显示和维护项目。

SDK: [Business Connectivity Services 概述](#)

SDK: [Microsoft Business Connectivity Services](#)

SDK: [Business Connectivity Services: 操作方法和演练](#)

SDK: [使用 .NET 程序集连接器与编写自定义连接器之间的差异](#)

SDK: [如何: 在 SharePoint 中创建外部列表](#)

SDK [如何: 基于 SQL Server 表创建外部内容类型](#)

SDK [如何: 使用 Business Connectivity Services 连接到 Oracle 数据库](#)

SDK [如何: 创建包含外部数据列的 SharePoint 列表](#)

操作方法视频: [创建可在外部数据源中读取和写入数据的 SharePoint 2010 Web 部件](#)

演练: [Creating an External List in SharePoint by Using Business Data](#)

视频和动手实验: [使用 SharePoint 2010 中的 Business Connectivity Services 访问外部数据](#) (该链接可能指向英文页面)

博客: [使用 Visual Studio 2010 创建 SharePoint 2010 外部列表的演练](#) (该链接可能指向英文页面)

博客: [使用 Business Connectivity Services 在 SharePoint 2010 中搜索外部数据](#) (该链接可能指向英文页面)

### **Business Connectivity Services 客户端 API**

通过使用 Business Connectivity Services 客户端 API, 可以创建具有针对外部内容类型的创建、读取、更新和删除 (CRUD) 功能的应用程序。在构建 Microsoft Office 托管外接程序时, 一个显示用户界面的简便方法是使用任务窗格或操作窗格。这允许用户继续在其文档或电子表格中工作的同时与 Business Connectivity Services 数据进行交互。

Business Connectivity Services 客户端 API 为客户端应用程序提供了从 Business Connectivity Services 访问数据的方法。此外, 您还可以配置 Business Connectivity Services, 以便使数据脱机。当客户端计算机再次联机时, 如果有必要, 所做的更改将使用冲突解决通过 Business Connectivity Services 进行同步。Business Connectivity Services 客户端 API 最常在 [Office 2010 客户端托管外接程序](#) 中使用, 尽管它可以通过任何基于 .NET Framework 的应用程序使用。

您可以使数据脱机, 以使用户能够继续检查和修改缓存中的数据。如果某个脱机用户对数据所做的修改与其他用户所做的修改相冲突, 则可以编写代码来解决这些冲突, 并向用户提供适当的选项。

SDK: [使用 Business Connectivity Services 创建基于代码的高级解决方案](#)

SDK: [如何: 使用内容控件在 Word 中显示外部数据](#)

博客: [Business Connectivity Services 基于代码的高级解决方案的概述](#) (该链接可能指向英文页面)

### **协作应用程序标记语言 (CAML)**

协作应用程序标记语言 (CAML) 是基于 XML 的查询语言, 您可以利用它指定筛选器、排序顺序和其他内容。您可以准确地指定要检索每个项目中的哪些字段, 这样您便可以控制查询结果的大小。

尽管在某些情况下可以使用 LINQ to SharePoint 代替 CAML 进行查询, 但 CAML 在 SharePoint 2010 中仍是相关的。您在使用[托管客户端对象模型](#)进行查询时将使用 CAML。在使用托管客户端对象模型时, 如果使用 LINQ 筛选列表项而不是使用 CAML, 则基础系统中的筛选将在所有项目返回之后发生。这可能导致在服务器上使用的 CPU 和内存资源高于必需的量。

SDK: [协作应用程序标记语言核心架构](#)

SDK: [协作应用程序标记语言 \(CAML\) 简介](#)

## 列类型和字段类型

利用列类型和字段类型，您可以为 SharePoint 列表中的标准化字段和列定义大量元数据。您可以定义能在整个网站中使用的特定数据类型的网站栏（在网站栏库中）。这类似于在数据库架构设计中定义域。这样您就可以确保多个列表中的列使用同一值空间。

SDK: [自定义字段类型](#)

SDK: [列](#)

## 内容类型

您可以定义作为某列表项的原型的内容类型。如果您对列表使用了内容类型，则可以定义该列表只包含一个内容类型的项目，或者定义该列表可包含多个内容类型中的一种内容类型的项目。

SDK: [内容类型](#)

SDK: [内容类型简介](#)

## 自定义操作

自定义操作表示一个链接、工具栏按钮、菜单项或者可添加到出现在用户界面中的工具栏或菜单的任意控件。可以使用功能定义文件中的自定义操作元素来定义自定义操作。可以将自定义操作绑定到列表类型、内容类型、文件类型或编程标识符 (ProgID)。

SDK: [自定义操作](#)

SDK: [如何：使用自定义操作修改用户界面](#)

## 自定义视图

为 SharePoint 列表定义自定义视图是工具包中的一个重要工具。您可以通过编程方式为特定用户或方案定义自定义视图，以便数据以最简便的形式显示。自定义视图的常见用法是定义导出到电子表格以进行其他处理的数据的格式。

### 注释：

此上下文中的 *视图* 不同于 SQL Server 中的视图。在 SQL Server 中，视图是将视图定义为若干个表的数据库构造。在此上下文中，视图是显示在页面中的可见表，以便用户可以看到 SharePoint 列表中的特定数据子集。

Office Online: [创建或更改视图](#)

SDK: [SPViewCollection](#)

SDK: [SP.View Class](#) (该链接可能指向英文页面)

SDK: [SP.ViewCollection Class](#) (该链接可能指向英文页面)

## 仪表板和 KPI

仪表板 是信息工作者的一个解决方案，可将个人、团队、企业或外部信息结合在一起，并提供对分析和协作工具的单击访问。仪表板可将公司数据资源的整合视图显示在单一桌面上。这样用户便能方便地访问关键业务信息。

入门: [商业智能中心简介](#)

Office Online: [仪表板入门](#)

Office Online: [KPI 和状态指示符](#)

SDK: [使用 PerformancePoint Services 的开发方案](#)

## 文档库

SharePoint 文档库是 SharePoint 列表的特殊形式。这些文档库具有保留主要和次要版本的文档等功能，可以根据身份限制对文档的访问。您可以定义任意数量的列，这些列包含有关库中的每个文档的辅助信息。

文档库的一个非常重要的功能就是您可以在文档中将栏式数据绑定到元数据。当从库中签出文档时，会将元数据复制到该文档，以便用户可在文档中维护元数据。当将文档签入回文档库中时，更新的元数据将会复制回文档库中项目的字段中。

文档库可包含任意类型的文件，还可以包含嵌套文件夹。这使得文档库与文件系统类似。

SDK: [构建基块: 文件和文档](#)

## 文档元数据

文档元数据是 SharePoint 为文档库中的每个文档存储的自定义数据。您的用户可使用 SharePoint Web 用户界面或 Microsoft Word 指定此数据。您可以在用户向文档库添加文档时以编程方式在事件处理程序中设置此数据，然后用户可以使用此数据筛选和查找文档。此外，您还可以在 [SharePoint 工作流](#)中访问此元数据，并根据元数据更改工作流行为。一般而言，应使用 [内容类型](#)定义文档元数据。工作流可以与此数据交互。

文档信息面板是在客户端应用程序中显示的窗体。它们包含文档元数据的字段。利用文档信息面板，用户可以在需要时输入有关文件的重要元数据而不必退出客户端应用程序。

SDK: [元数据导航和筛选](#)

SDK: [文档信息面板概述](#)

## ECMAScript 客户端对象模型

ECMAScript 客户端对象模型是可通过 ECMAScript 使用的托管客户端对象模型的变体。您可以使用相同的方法和数据类型，并且适用于相同的编程模型。与托管客户端对象模型一样，您也可以控制向服务器发送以及从服务器发送网络流量的时间。

SDK: [托管客户端对象模型](#)

SDK: [托管对象模型和 ECMAScript 对象模型的区别](#)

## 事件接收器

利用事件接收器，您可以编写事件处理程序，这些程序将在用户在 SharePoint 文档库或列表中添加、删除或修改项目时以及其他时间进行调用。您可以将事件处理程序部署为完全信任服务器服务器场解决方案或沙盒解决方案。

视频: [如何: 在 Visual Studio 2010 中创建用于 SharePoint 2010 的事件接收器 \(该链接可能指向英文页面\)](#)

SDK: [构造块: 事件处理](#)

SDK: [SharePoint Foundation 2010 中的事件](#)

SDK: [SharePoint Foundation 事件简介](#)

SDK: [Walkthrough: Add Feature Event Receivers](#)

SDK: [SPEventReceiverDefinition](#)

## Excel Services

利用 Excel Services，用户可以在 SharePoint Server 2010 上加载、计算和显示 Excel 工作簿。Excel Services 有四个主要编程接口。

- **Excel Web Access。**可以添加到 SharePoint Server 2010 中的任意 Web 部件页的查看器页和 Excel Services Web 部件。Excel Web Access 将在网页上呈现活动 Excel 工作簿（也就是说，为活动 Excel 工作簿创建 HTML），从而允许用户与这些工作簿交互。
- **Excel Web Services。**提供编程访问。
- **JavaScript 对象模型。**允许您自动化、自定义和驱动 Excel Web Access 控件，并帮助构建更引人注目的集成解决方案。
- **具象状态传输 (REST) API。**允许通过 URL 直接访问工作簿文档部件。

若要自定义电子表格，可以通过创建 [Excel Services 用户定义函数扩展](#) Excel Calculation Services。

Office Online: [Excel Services](#)

SDK: [Excel Web Access](#)

SDK: [Excel Web Services](#)

SDK: [Excel Services ECMAScript \(JavaScript、JScript\)](#)

SDK: [Excel Services REST API](#)

SDK: [Getting Started with Excel Services](#) (该链接可能指向英文页面)

SDK: [Excel Services 开发路线图](#)

### **Excel Services ECMAScript 接口**

利用 Excel Services 中的 ECMAScript 对象模型，您可以编写用于自动化和自定义页面上的 Excel Web Access Web 部件控件以及与该控件进行交互的代码。通过使用 ECMAScript 对象模型，您可以构建混合和其他与 Excel Web Access Web 部件控件交互的集成解决方案。您还可以利用它向工作簿添加更多功能。

SDK: [Excel Services ECMAScript \(JavaScript、JScript\)](#)

### **Excel Services REST API**

Excel Services REST API 通过 URL 直接向工作簿文档部件或元素提供编程访问权限。

SDK: [Excel Services REST API](#)

### **Excel Services 用户定义函数**

利用 Excel Services 用户定义函数 (UDF)，您可以使用 Microsoft Visual C 或 Microsoft Visual C++ 语言编写在服务器端电子表格中使用的高性能函数。

SDK: [Excel Services 用户定义函数](#)

直观操作方法: [Writing Asynchronous User-Defined Functions in Excel 2010](#) (该链接可能指向英文页面)

直观操作方法: [Creating Managed-Code UDFs for Excel Services](#) (该链接可能指向英文页面)

### **Excel Web Services**

可以使用 Excel Web Services 以编程方式访问 Excel Services。

SDK: [Excel Web Services](#)

## 功能

功能是 SharePoint 的部署单元。在大型服务器场中部署功能可确保该功能中的项目安装在所有服务器上。功能可包含模板、页面、列表定义、事件处理程序、工作流、ECMAScript 源文件和级联样式表文件。功能还可以包含功能接收器，这些接收器是安装、卸载、激活、禁用或升级功能时运行的事件处理程序。

SDK: [使用功能](#)

SDK: [功能升级概述](#)

SDK: [Walkthrough: Add Feature Event Receivers](#)

## 运行状况规则

SharePoint 运行状况分析器通过应用运行状况规则来监视整个服务器场中的 SharePoint Foundation 部署的运行状况。运行状况规则是可执行代码，它部署在服务器场级别的已编译程序集中，并注册到 SharePoint 运行状况分析器。在注册规则时，会创建计时器作业来运行这些规则。与规则对应的项目将添加到“运行状况分析器规则”列表中，该列表位于管理中心的“监视”部分的“复查分析器规则”下。

SDK: [构建基块: 运行状况规则](#)

SDK: [SharePoint 运行状况分析器](#)

## InfoPath Forms Services

InfoPath Forms Services 是一项服务器技术，用户可以利用它在 Web 浏览器中填写表单。通过在中心位置管理表单，您可以简化业务流程并改善与客户、合作伙伴和供应商的关系。

开发人员中心: [InfoPath 开发人员中心](#) (该链接可能指向英文页面)

Office Online: [InfoPath 2010 入门](#)

TechNet: [规划 InfoPath Forms Services \(SharePoint Server 2010\)](#)

## LINQ to SharePoint

LINQ to SharePoint 是一个 LINQ 查询提供程序，利用它，您可以比使用 CAML 时更自然地编写查询。当使用 SharePoint Server 对象模型时，此程序最有用。

LINQ to SharePoint 的一个重要特征是，它使用某种方法生成代码，使您的代码使用强类型对象模型来访问查询返回的数据。

对于对函数编程感兴趣的开发人员，LINQ to SharePoint 将充当执行纯函数变换的数据源。

SDK: [使用 LINQ to SharePoint 管理数据](#)

视频和动手实验: [使用服务器端 API 访问 SharePoint 2010 数据](#) (该链接可能指向英文页面)

博客: [使用 C# 3.0 中的函数编程技术编写查询](#) (该链接可能指向英文页面)

## 列表

SharePoint 网站中的自定义列表提供了存储数据的位置。使用 SharePoint 列表来包含以编程方式访问的数据对开发人员来说是一个非常重要的构建基块。在 [将 SharePoint 应用程序与数据库应用程序进行比较](#) 一节中，我介绍了如何对部分数据库使用 SharePoint 列表。

SDK: [构造块: 列表和文档库](#)

SDK: [SharePoint 列表数据模型](#)

## 托管客户端对象模型

托管客户端对象模型提供了从基于 .NET Framework 的外部应用程序访问 SharePoint 数据的权限。此 API 也称为客户端对象模型 (CSOM)。

托管客户端对象模型有三个基本用途:

- **托管 .NET-connected 应用程序。**可以通过大多数基于 .NET Framework 的应用程序使用托管客户端对象模型。这包括大量在用户计算机上运行的客户端应用程序，或者作为基于 .NET Framework 的服务器应用程序一部分的客户端应用程序。您可以使用托管客户端对象模型编写在客户端计算机上运行并从 SharePoint Server 提取信息的 Windows Presentation Foundation (WPF) 不动产管理应用程序。
- **Silverlight Web 部件。** Silverlight 控件中的代码可使用客户端对象模型。
- **HTML Web 部件。**您可以使用 ECMAScript 版本的客户端对象模型。它具有完全相同的语义，除非您使用 ECMAScript 编程接口。

### 重要信息:

当使用托管客户端对象模型来查询列表中的项目时，应总是使用 CAML 来构建查询，而不是使用 LINQ。如果使用 LINQ，则性能会受到列表中的项目总数的影响。相反，当使用服务器对象模型时，建议使用 LINQ to SharePoint 而不使用 CAML。

客户端代码的常见活动是在 SharePoint 文档库中上传和下载文档和文件。在托管客户端模型中，[OpenBinaryDirect](#) 方法和 [SaveBinaryDirect](#) 方法使用基于 Web 的分布式创作和版本管理

(WebDAV) 来上载和下载文件。如果没有生成您自己的自定义 Windows Communication Foundation (WCF)，那么 WebDAV 就是最有效的上载和下载文件的方法。

SDK: [托管客户端对象模型](#)

技术文章: [使用 SharePoint Foundation 2010 托管客户端对象模型](#)

技术文章: [将 SharePoint Foundation 2010 托管客户端对象模型用于 Open XML SDK 2.0](#)

视频和动手实验: [使用客户端 API 访问 SharePoint 2010 数据 \(该链接可能指向英文页面\)](#)

### 移动页、控件和适配器

SharePoint Foundation 2010 提供了与您在台式计算机和便携计算机上看到的页面和控件有所不同的移动页和移动控件。移动浏览器可通过移动 Web 部件适配器访问某些 Web 部件。

SDK: [构建基块: 移动页、控件和适配器](#)

### OData REST API

SharePoint 2010 使用 OData 协议 ([该链接可能指向英文页面](#)) 将数据作为可由 URI 寻址的资源发布。您可以使用 GET、PUT、POST 和 DELETE 这些标准 HTTP 谓词来访问和更改数据。OData 使用实体数据模型的实体关系约定将资源作为通过关联相关的实体公开。

SDK: [SharePoint Foundation REST 接口](#)

### Office 客户端托管外接程序

Office 客户端托管外接程序提供了扩展 Office 客户端应用程序并将它们更深入地与 SharePoint Foundation 和 SharePoint Server 集成的方法。生成 Office 客户端托管外接程序是一种使 SharePoint 应用程序的功能可供发现并方便用户使用的有效方法。SharePoint 数据与 Microsoft Word 文档或 Excel 电子表格的深度集成可使应用程序中只有重要数据的一个版本，从而有助于防止基于陈旧数据进行决策。

您可以通过 [Business Connectivity Services 客户端 API](#) 访问 SharePoint 外部内容类型，还可以使用[托管客户端对象模型](#)访问存储在 SharePoint 列表和文档中的数据和文档。

开发人员中心: [Office 开发人员中心](#)

开发人员中心: [使用 Visual Studio 进行 Office 开发 \(该链接可能指向英文页面\)](#)

视频: [使用 Office 2010 进行开发 \(该链接可能指向英文页面\)](#)

### Open XML SDK

大多数 SharePoint 用户都会广泛使用文档库，文档库中的大多数文档都以 Open XML 文件格式存储。在 Open XML 出现以前，编写直接访问文档内容的代码很困难，但自从有了 [Open XML SDK](#)，您就拥有了查询和转换 Open XML 文档的服务器强化工具。从 SharePoint 应用程序使用 Open XML 开辟了新的方案。您可以生成许多文档并使它们在各种格式之间自动转换，还可以查询现有文档的内容并将这些内容和数据以新颖有趣的方式提供给用户。例如，产品设计团队可针对规范使用特定的文档模板，规范的每个部分可能遵循用于报告规范状态的约定。您然后可以编写应用程序来处理项目的所有规范，并生成显示规范编写过程的总体状态的管理报告。

通过将 Open XML SDK 的强大功能与 Word Services 的强大功能结合，可以实现很多性能优越的服务器端方案。

在 Open XML 出现以前，客户有时候会尝试使用 Office 自动化解解决服务器端文档处理需求。在服务器上自动化 Microsoft Office 应用程序会带来一些已知的问题。解决方法是使用 SharePoint 2010 和 Open XML SDK。

SharePoint 2010 中生成文档的常见模式是将它们生成到 SharePoint 文档库中。然后，如有必要，可以使用 [Word Automation Services](#) 来更新该文档中的字段或导入 altChunk 内容。

开发中心: [Open XML 格式资源中心](#) (该链接可能指向英文页面)

外部链接: [ECMA 376-2](#) (该链接可能指向英文页面)

外部链接: [ISO/IEC 29500-2](#) (该链接可能指向英文页面)

KB 文章: [服务器端 Office 自动化注意事项](#)

博客: [Eric White 的博客](#) (该链接可能指向英文页面)

博客: [Brian Jones 的博客](#) (该链接可能指向英文页面)

博客: [如何对文档程序集使用 altChunk](#) (该链接可能指向英文页面)

## PerformancePoint Services

SharePoint Server 2010 中的 PerformancePoint Services 是一项性能管理服务，可用于监视和分析您的业务。它提供了用于构建仪表盘、记分卡、报告和关键性能指标 (KPI) 的工具。集成分析有助于员工从监视信息快速转换到分析信息，并适时在整个组织内共享信息。

SDK: [使用 PerformancePoint Services 的开发方案](#)

## PowerPivot

PowerPivot 允许用户创建引人注目的自助服务商业智能解决方案，支持对用户生成的商业智能解决方案的共享和协作，并允许 IT 组织通过基于 Microsoft SQL Server 2008 R2 的管理工具提高操作效率。

MSDN: [PowerPivot Overview](#)

外部: [PowerPivot.com](#) (该链接可能指向英文页面)

## 沙盒解决方案

当 SharePoint 用户必须内置专用的解决方案时, 沙盒解决方案是一种解决可伸缩性问题的方法。如果每位开发人员都知道所有最佳做法, 并严格遵循这些做法, 而且编写的代码总是很完美, 就不会存在这个问题了。但是, 实际上代码并不完美, 而且使用服务器场管理员权限运行的代码中的 Bug 会降低服务器场的性能。通过使用沙盒解决方案, 基础结构会关注代码中引发的异常并监视内存和 CPU 使用率等。行为异常的代码将停止运行, 而服务器场不会停止运行。

SDK: [沙盒解决方案](#)

SDK 如何: [创建沙盒 workflow 操作](#)

视频和动手实验: [SharePoint 2010 中的 Web 部件的沙盒解决方案](#) (该链接可能指向英文页面)

## 搜索自定义

您可以自定义搜索, 以便为新的不同类型的内容编制索引。这包括[外部内容类型](#)。

SDK: [SharePoint 企业级搜索](#)

博客: [使用 Business Connectivity Services 在 SharePoint 2010 中搜索外部数据](#) (该链接可能指向英文页面)

## Secure Store Service

Secure Store Service 是一种声明感知授权服务, 用于提供凭据 (如帐户名称和密码) 的存储和映射。利用它, 您可以安全地存储用于提供凭据的数据, 这些凭据是连接到外部系统和将自身与特定标识或标识组关联所需的。

SDK: [Secure Store Service](#)

SDK: 如何: [使用 Secure Store Service 连接到外部系统](#)

## 安全性

SharePoint Foundation 2010、基础 ASP.NET 和 .NET Framework 的安全功能有助于防止对数据进行未经授权更改 (无论有意还是无意), 并帮助确保敏感信息仅可供授权用户查看。

SDK: [SharePoint Foundation 安全性](#)

## 服务器对象模型

利用 SharePoint 服务器端对象模型，您可以编写程序以访问 SharePoint 列表和文档库、网站集、网站和其他资源。尽管该对象模型非常大，但并不难。

如果使用 SharePoint 进行开发，情况就可能比较复杂。您可能要开发 Web 部件、事件接收器或作为工作流的一部分运行的代码。在所有这些情况下，您都可以编写使用服务器对象模型的代码。这意味着您的编辑/编译/执行周期会更加复杂。在每次编译时，您可能都必须重新安装 SharePoint 功能、循环使用 Internet Information Services 或连接到其他过程以进行调试

。在学习服务器对象模型的工作方式的过程中就指望解决这些问题是不可取的。相反，在我学习服务器对象模型时，我生成了使用服务器对象模型的小型控制台应用程序，并直接在服务器上执行这些控制台应用程序。在编写更复杂的代码（最终构成 Web 部件）时，我对代码进行了分解，以便使用服务器对象模型的代码被分隔在应用程序中的一个或多个类中。我还将使用 Open XML SDK 的代码分解到它自己的一组类中。这样就加快了 Open XML 代码和服务器对象模型代码的编辑/编译/执行周期。然后，在开发完所有组成部件之后，就可以轻松地生成和部署使用已调试代码的 Web 部件。

使用服务器对象模型的代码可以使用服务器场管理员权限运行或作为沙盒解决方案运行。通过使用沙盒解决方案，您可以访问可用于通过服务器场管理员权限运行的代码的 API 子集。

您可以将服务器对象模型分为三个层次结构：

- [Microsoft SharePoint Foundation 的内容层次结构](#) 讨论了用于表示数据的可发布项（如列表项）的各种类，还讨论了用于表示数据的嵌套容器的类，如列表、内容数据库、网站、网站集和称为“Web 应用程序”的网站集分组。
- [Microsoft SharePoint Foundation 的物理对象层次结构](#) 讨论了用于表示物理实体的各种类（如服务器和文件）以及这些实体的容器（如服务器场和文件夹）。
- [Microsoft SharePoint Foundation 的服务层次结构](#) 讨论了用于表示 Web 服务、Windows 服务和其他类型的服务以及服务实例的类。

SDK: [服务器和网站体系结构：对象模型概述](#)

SDK: [基本 SharePoint 编程任务](#)

SDK: [使用 SharePoint Foundation 服务器端对象模型](#)

SDK: [基本对象模型任务](#)

SDK: [数字资产管理编程模型](#)

SDK: [文档管理编程模型](#)

SDK: [记录管理编程模型](#)

SDK: [元数据和分类编程模型](#)

SDK: [发布编程模型](#)

SDK: [沙盒解决方案](#)

视频和动手实验: [使用服务器端 API 访问 SharePoint 2010 数据](#) (该链接可能指向英文页面)

## 服务器功能区

通过自定义 SharePoint 2010 服务器功能区, 您可以为 SharePoint 应用程序提供直观且功能强大的用户界面。自定义功能区背后的关键点是使用户可以发现您的自定义 SharePoint 应用程序的功能。自定义功能区可与自定义 Office 客户端应用程序的功能区相提并论。

SDK: [构造块: 服务器功能区](#)

SDK: [SharePoint Foundation 中的服务器功能区](#)

SDK: [服务器功能区的体系结构](#)

SDK: [自定义服务器功能区](#)

视频和动手实验: [为 SharePoint 2010 创建对话框和功能控件](#) (该链接可能指向英文页面)

## 服务应用程序体系结构

利用服务应用程序体系结构, 您可以创建可由多个 SharePoint 应用程序使用的可靠且可扩展的共享服务。

SDK: [服务应用程序框架](#)

SDK: [服务应用程序框架体系结构](#)

视频: [SharePoint 2010 服务应用程序体系结构](#) (该链接可能指向英文页面)

视频: [创建自定义服务应用程序](#) (该链接可能指向英文页面)

## SharePoint 对话框框架

在生成 Web 部件或应用程序页时, 可能需要显示各种对话框。

SDK: [Walkthrough: Customizing the PWA Ribbon and Accessing the JS Grid](#) (该链接可能指向英文页面)

视频: [SharePoint 2010 UI 功能](#) (该链接可能指向英文页面)

博客: [如何通过功能区按钮显示 SharePoint 对话框并获取选定的项目上下文](#) (该链接可能指向英文页面)

视频和动手实验: [为 SharePoint 2010 创建对话框和功能区控件 \(该链接可能指向英文页面\)](#)

## SharePoint 主题

主题对于创建 SharePoint 网站品牌很重要, 它能提供快速简便的方法在 SharePoint 2010 中对网站应用颜色和字体。主题的更改将导致大多数页面元素 (如背景图像、文本、超链接) 的颜色发生更改。对某些页面元素 (如标题) 使用的字体也会发生更改。

TechNet: [主题概述 \(SharePoint Server 2010\)](#)

## SharePoint Web Services

SharePoint 长久以来一直提供一组 Web 服务来实现客户端与 SharePoint 部署的交互。这些服务中有一些是专门针对数据检索的, 而另一些包含了某些数据检索服务。这些提供数据的 Web 服务的 API 仍受支持, 以实现向后兼容以及与 Web 服务客户端的互操作。出于性能方面的原因, 也为了方便起见, 我们建议使用[客户端对象模型](#)或 [OData REST API](#) 来查询 SharePoint 数据。

SDK: [SharePoint Foundation 2010 Web 服务](#)

SDK: [WebSvcLists](#)

SDK: [WebSvcWebs](#)

SDK: [WebSvcSites](#)

SDK: [WebSvcSiteData](#)

SDK: [WebSvcDspSts](#)

互操作性文档: [SharePoint Products and Technologies Protocols \(该链接可能指向英文页面\)](#)

不应使用 Web 服务, 而应参阅以下内容:

SDK: [SharePoint Foundation REST 接口](#)

SDK: [托管客户端对象模型](#)

技术文章: [使用 SharePoint Foundation 2010 托管客户端对象模型](#)

技术文章: [将 SharePoint Foundation 2010 托管客户端对象模型用于 Open XML SDK 2.0](#)

视频和动手实验: [使用客户端 API 访问 SharePoint 2010 数据 \(该链接可能指向英文页面\)](#)

## Silverlight Web 部件

Silverlight Web 部件是构建最佳用户界面的重要工具。有时候除了 Silverlight, 其他工具都不起作用。

代码示例: [SharePoint Foundation 2010 代码示例](#)

视频和动手实验: [为 SharePoint 2010 解决方案创建 Silverlight 用户界面](#) (该链接可能指向英文页面)

## 网站定义

网站定义是一系列文件,其中包含存储在所有前端 Web 服务器上的 XML 配置文件。网站定义提供了网站外观、网站包含的列表、网站的默认导航结构等设置的基本蓝图。

SDK: [网站类型: Web 模板和网站定义](#)

## 网站页面

网站页面提供了有效的方法(针对存储和 CPU 时间)来支持成千上万的网页,其中很多网页是相同的,而且有一些网页是自定义的。与传统的 ASP.NET 应用程序不同,服务器上的每个网站页面不必存在实际的 .aspx 页面。SharePoint 会将任何未自定义的网站页面视为虚拟文件,这样便只有少数文件会出现在多个位置。这是 SharePoint 提供可伸缩性的多个方面中的一个。如果没有此功能,大型网站就必须包含极多的已编译 .aspx 页面。管理如此多的页面将很困难。

SDK: [网站和网页](#)

SDK: [SharePoint 页面类型](#)

博客: [使用 Visual Studio 2010 通过代码隐藏创建 SharePoint 网站页面](#) (该链接可能指向英文页面)

## 计时器作业

与 cron 作业一样,计时器作业也是在特定时间运行的代码。应用程序可能必须定期处理 SharePoint 列表中的数据,还可能要修改外部系统中的数据。您可以为应用程序编写计时器作业,使它在每天凌晨 1 点运行。

SDK: [Timer job reference \(SharePoint Server 2010\)](#) (该链接可能指向英文页面)

## 变体

利用变体,您可以针对不同的文化、市场和语言自定义网站。这是一种便于向不同受众提供不同内容的功能。变体有助于根据用户的语言提供不同的内容。此外,变体还可以根据最终用户设备或您确定的任何决定性差异(如品牌或用户专业知识的水平)提供不同的内容。

SDK: [变体和多语言网站](#)

博客: [SharePoint 2010 中的变体](#) (该链接可能指向英文页面)

## Visio Services

Visio Services 是一种服务应用程序，允许用户共享和查看 Microsoft Visio Web 绘图。该服务也使数据连接 Visio 2010 Web 绘图能够从各种数据源刷新和更新。

SDK: [Visio Services 概述 \(SharePoint Server 2010\)](#)

## Visual Studio SharePoint 工具扩展性

Visual Studio 2010 中的 SharePoint 工具能满足很多应用程序开发方案的要求。但是，您可能会发现，在某些情况下，这些工具不能提供您或其他开发人员所需的功能。在这些情况下，您可以扩展 SharePoint 工具以创建需要的功能。

SDK: [Visual Studio SharePoint Tools Extensibility](#)

视频: [Visual Studio 2010 中的 SharePoint 2010 开发工具入门](#)

## 基于 Web 的分布式创作和版本管理 (WebDAV)

客户端代码的常见活动是上载和下载文档和文件。您可能必须将文档从 SharePoint 文档库检索到客户端计算机，或将文档从客户端计算机上载到 SharePoint 文档库。您可以直接使用 WebDAV。但还有一种简单的使用 WebDAV 的方法，那就是使用托管客户端对象模型。客户端对象模型中的 **OpenBinaryDirect** 方法和 **SaveBinaryDirect** 方法使用 WebDAV 上载和下载文件。在没有生成您自己的自定义 WCF 服务的情况下，WebDAV 是最有效的上载和下载文件的方式。请注意，WebDAV 只能访问文件，而不能访问数据。

外部链接: [RFC 4918 \(该链接可能指向英文页面\)](#)

## Web 部件

SharePoint 基于 ASP.NET。它继承了 ASP.NET 的几个重要方面，其中包括 Web 部件页和 Web 部件的概念。表示 SharePoint Web 部件的类派生自相应的 ASP.NET 类。

SharePoint 开发人员生成 SharePoint Web 部件所耗的时间比生成任何其他类别的所耗的时间都多（可能比生成所有其他类别所耗的总时间还多）。如今，用户比以前更期待漂亮、直观且功能强大的用户界面，开发 Web 部件通常是最有效的方法。

Web 部件能与其他 Web 部件通信。例如，您可以将它们链接在一起，以便在维护数据时启用父/子表单/列表用户界面。

AJAX Web 部件是一种特殊类别。当您的最佳用户界面需要它时，便可以利用 AJAX 体系结构。

Silverlight Web 部件是另一种特殊类别。在某些情况下，Silverlight 是最合适的。就个人而言，我认为 SharePoint 中的 Silverlight Web 部件是将来的流行趋势。

您可以使用 ASP.NET 代码隐藏模型在沙盒或完全信任模式下生成 Web 部件。

在客户端，Web 部件可以通过托管客户端对象模型、OData REST API 或 Web 服务与 SharePoint 服务器通信。WCF 是 Silverlight Web 部件中的一个选项。

尽管您可以生成自定义 Web 部件，但 SharePoint 2010 还是提供了很多对于设计自定义解决方案很有用的 Web 部件。其中比较常用的一些 SharePoint 内置 Web 部件包括以下内容。

### 常用内置 Web 部件

Web 部件	链接
列表视图 Web 部件	SDK: <a href="#">列表视图</a> 博客: <a href="#">SharePoint 2010 列表视图博客系列文章</a> (该链接可能指向英文页面) Office Online: <a href="#">关于列表视图 Web 部件</a> (该链接可能指向英文页面)
图像 Web 部件	Office Online: <a href="#">图像 Web 部件</a>
网站用户 Web 部件	Office Online: <a href="#">网站用户 Web 部件</a>
图表 Web 部件	Office Online: <a href="#">图表 Web 部件简介</a>

SDK: [SharePoint Foundation 中的 Web 部件](#)

SDK: [Microsoft.SharePoint.WebPartPages](#)

SDK: [构造块: Web 部件](#)

SDK: [SharePoint 2010 中的连接 Web 部件](#)

SDK: [JS 网格控件](#)

SDK: [如何: 创建基本 JS 网格](#)

视频和动手实验: [SharePoint 2010 中的 Web 部件构建入门](#) (该链接可能指向英文页面)

视频和动手实验: [SharePoint 2010 中的 Web 部件开发的构建基块](#) (该链接可能指向英文页面)

视频和动手实验: [为 SharePoint 2010 解决方案创建 Silverlight 用户界面](#) (该链接可能指向英文页面)

视频和动手实验: [SharePoint 2010 中的 Web 部件的沙盒解决方案](#) (该链接可能指向英文页面)

视频: [创建可在外部数据源中读取和写入数据的 SharePoint 2010 Web 部件](#)

视频: [SharePoint 2010 中的高级 Web 部件开发 \(该链接可能指向英文页面\)](#)

SDK: [SharePoint Foundation 中级联样式表类的用法](#)

代码示例: [SharePoint Foundation 2010 代码示例](#)

## Web 服务

在生成大型分布式应用程序时,最佳做法是使用 ASMX Web 服务或 WCF 服务将它们松散地结合起来。您在生成大型 SharePoint 应用程序时也必须应用这种方法。

作为示例,运筹学通常要求部署复杂的数学优化程序来得出对某些问题的最佳或接近最佳的解决方案。这应在专用硬件上进行部署并通过 Web 服务公开。

在某些情况下,您可能希望生成使用经过最大程度扩展的高级服务器端功能的客户端应用程序,还可能希望最大程度地减小多余的网络流量。在这些情况下,最佳解决方案是在 SharePoint Server 上承载 WCF 服务。此服务中的代码可通过完全服务器场权限使用服务器对象模型,并为客户端应用程序提供必要的功能。根据需要,当生成这些类型的服务时,您必须执行攻击面分析和威胁建模。

开发人员中心: [WCF 开发人员中心: \(该链接可能指向英文页面\)](#)

博客: [WCF Web 服务构建入门 \(该链接可能指向英文页面\)](#)

## Windows Identity Foundation

SharePoint Foundation 2010 和 SharePoint Server 2010 的基于声明的标识模型基于 Windows Identity Foundation (WIF) 构建。基于声明的标识的功能包括:

- 基于 Windows 系统和非基于 Windows 系统的用户身份验证。
- 多种身份验证类型。
- 更强的实时身份验证。
- 更广泛的主体类型。
- 应用程序之间的用户标识委派。

当您构建声明感知应用程序时,用户会以一组声明的形式向您的应用程序提供标识。其中的一个声明可能是用户名,另一个声明可能是电子邮件地址。原因是外部标识系统会被配置为在每次收到请求时向您的应用程序提供所有需要的关于用户的信息,并提供您的应用程序收到的标识数据是来自受信任的源的加密保证。在此模型下,单一登录更容易实现。

开发人员中心: [SharePoint 2010 安全资源中心 \(该链接可能指向英文页面\)](#)

开发人员中心: [Windows Identity Foundation](#) (该链接可能指向英文页面)

博客: [使用 SharePoint 2010 外部内容类型形式的启用声明的 WCF Web 服务](#) (该链接可能指向英文页面)

### Word Automation Services

Open XML SDK 中有一些任务难以执行,如重新分页、转换为其他文档格式(如 PDF)或者更新字段代码或文档中的目录、字段和其他动态内容。Word Automation Services 是 SharePoint 2010 的一种功能,可在这些情况下发挥作用。它是一种共享服务,能提供文档到其他格式的无人参与服务器端转换的功能,以及其他一些重要功能。从一开始,它就设计为在服务器上工作并能以可靠且可预测的方式处理多个文档。

SDK: [Word Automation Services](#)

技术文章: [使用 SharePoint 2010 Word Automation Services 进行开发](#)

### 工作流

利用 SharePoint 工作流,您可以编纂业务流程。工作流逻辑可以非常简单,也可以非常复杂。最终用户可使用工作流来生成无代码业务解决方案。您可以使用一些在适当的时间调用的 C# 代码来改进工作流。您可以将工作流部署为完全信任服务器服务器场解决方案或部署为沙盒解决方案。工作流使用的示例包括声明处理、内容审批或采购流程。

SDK: [SharePoint Foundation 中的工作流](#)

SDK: [Microsoft.SharePoint.Workflow](#)

SDK: [演练: 结合使用 SharePoint 完全信任工作流活动与 Business Connectivity Services](#)

视频和动手实验: [使用 SharePoint 2010 工作流开发业务流程](#) (该链接可能指向英文页面)

## 使用开发人员仪表板查看性能指标

开发 SharePoint 应用程序可能非常复杂。如果页面或 Web 部件性能不佳,则难以确定延迟或资源消耗的原因。SharePoint 2010 提供了一个开发人员仪表板,可用于查看与正在加载的页面有关的性能指标。

您可以通过运行以下 Stsadm 命令打开开发人员仪表板:

**stsadm -o setproperty -pn developer-dashboard -pv OnDemand**

SDK: [ASP.NET 开发人员的 SharePoint 2010 开发](#) (该链接可能指向英文页面)

SDK: [使用开发人员仪表板](#)

## SharePoint 2010 辅助功能准则

Microsoft 采用了 SharePoint 2010 Web 内容辅助功能准则 2.0 (WCAG 2.0)，并将目标级别设置为 AA。WCAG 2.0 于 2008 年 12 月 11 日成为 W3C 的推荐内容，其中阐明了用户对精心构建的可访问网站的期望，并规定了这样的网站中应部署的技术。

外部链接: [Web 内容辅助功能准则 2.0](#) (该链接可能指向英文页面)

博客: [可访问性和 SharePoint 2010](#) (该链接可能指向英文页面)

## SharePoint 2010 开发工具

有很多工具可用于生成 SharePoint 2010 应用程序。

### SharePoint 开发工具

工具	说明
Visual Studio 2010	Visual Studio 2010 是 SharePoint 2010 专业开发人员的主要工具。它包含预配置项目类型，可尽量简化为您的方案编写代码的过程。
SharePoint Designer 2010	利用 SharePoint Designer 2010，您可以创建包含常见方案的无代码 SharePoint 解决方案，如协作网站、Web 发布、业务线数据集成、商业智能解决方案和人员参与的工作流。您可以使用 SharePoint Designer 2010 开始 SharePoint 开发项目。
Visio 2010	利用 Visio 2010，您和行业专家可以通过拖放操作创建 SharePoint 工作流。您然后将这些工作流导入 SharePoint Designer 2010 中以进行进一步编辑，然后在 SharePoint 2010 中执行它们。

SDK: [SharePoint Foundation 开发工具](#)

MSDN: [SharePoint Development in Visual Studio](#)

MSDN 杂志: [用于 SharePoint 开发的 Visual Studio 2010 工具](#)

视频: [Visual Studio 2010 中的 SharePoint 2010 开发工具入门](#)

下载: [Microsoft SharePoint Designer 2010 \(32 位\)](#)

博客: [Visio Premium 2010 中的 SharePoint 工作流创作](#) (该链接可能指向英文页面)

博客: [Visual Studio SharePoint 开发博客](#) (该链接可能指向英文页面)

## 设置 SharePoint 2010 开发环境

有几种方法可以设置 SharePoint 2010 开发环境。

- 我最常用的方法是使用 Windows Server 2008 R2, 并通过 Hyper-V 设置我的开发计算机。在安装每个程序后, 我会立即拍摄快照。这样我就能在必要时恢复到之前的干净状态, 但根据我的经验, 这样的情况很少发生。若要获得最佳体验, 请在主机操作系统所在驱动器之外的驱动器上放置虚拟硬盘驱动器。
- 您可以在运行 Windows Server 2008 (或 Windows Server 2008 R2) 的计算机上进行安装, 该工具直接在服务器上运行, 而不是在 Hyper-V 虚拟计算机中运行。这种方法可能很有效, 但会失去恢复到某个快照的能力。
- 您可以在运行 Windows 7 64 位或 Windows Vista 64 位的计算机上进行安装。某些开发人员就职的公司会有政策阻止他们运行服务器操作系统, 因此, 这可能就是一个不错的解决方案。此外, 如果我要开始一个要耗时数月的大型开发项目, 则会在便携式计算机上安装 SharePoint 2010, 以便能在适当的时间进行开发和方便地使用最终用户操作系统。

SDK: [在 Windows Vista、Windows 7 和 Windows Server 2008 上设置 SharePoint 2010 开发环境](#).

博客文章: [SharePoint 开发环境](#) (该链接可能指向英文页面)

## SharePoint 2010 和应用程序生命周期管理 (ALM)

应用程序生命周期管理 (ALM) 是通过支持和跟踪工作团队之间和工作团队内部的协作的工具所进行的软件工程各方面的协作 - 包括业务和技术要求的提出和沟通、代码设计和体系结构、项目跟踪、变更管理、编码、测试、调试和发布管理。

资源中心: [应用程序生命周期管理资源中心](#) (该链接可能指向英文页面)

视频和动手实验: [SharePoint 2010 的开发生命周期](#) (该链接可能指向英文页面)

视频: [使用 Visual Studio 2010 改进整个生命周期的软件开发](#) (该链接可能指向英文页面)

## 结论

由于有了 SharePoint Foundation 2010 和 SharePoint Server 2010, SharePoint 已演变为一个全面的开发平台。尽管很多 SharePoint 应用程序都十分简单, 但如果使用多个构建基块协同工作,

SharePoint 应用程序就会变得非常复杂。您必须仔细设计您的应用程序，以确保应用程序可靠且具有扩展性，并确保应用程序的用户界面满足用户的需求。

## 其他资源

代码示例: [SharePoint Server 2010 代码示例](#)

代码示例: [SharePoint Foundation 2010 代码示例](#)

SDK: [SharePoint Server 的最佳做法](#)

文章: [SharePoint 2010 技术文章](#)

视频: [SharePoint 2010 视频 | 2009 年 SharePoint 会议](#) (该链接可能指向英文页面)

视频和动手实验: [开发人员需要了解的关于 SharePoint 2010 的内容](#) (该链接可能指向英文页面)

社区: [MSIW \(Microsoft Information Workers Network\)](#)

博客: [SharePoint 团队博客](#) (该链接可能指向英文页面)

博客: [Microsoft SharePoint 开发人员文档团队博客](#) (该链接可能指向英文页面)

MSDN: [开发人员入门培训](#)

MSDN: [开发人员高级培训](#) (该链接可能指向英文页面)

培训: [Microsoft 培训和 SharePoint 2010 认证](#) (该链接可能指向英文页面)