



在 PowerBuilder 11.1 中部署 Smart Client 应用

本文介绍 .NET Smart Client 的组成、属性及行为。此外，我们还将讨论如何利用 Intelligent Updater 支持 PowerBuilder 11.1 的 .NET Smart Client。最后，我们将详细演示如何使用 Intelligent Updater 在 PB 11 中安装、发布和更新 .NET Windows Form 部署的 PowerBuilder 应用。

.NET Smart Client 是什么？

事实证明，应用开发中最头疼的事是“部署”。一直以来，基于浏览器的应用自始至终都能够自动为每个用户更新应用逻辑和界面。但是这种称为瘦客户端的便捷的部署应用需要付出代价：基于浏览器的用户界面缺乏丰富的功能。PowerBuilder 11.1 的新 Intelligent Updater 提供丰富的界面交互功能，使开发人员、管理员以及任何负责部署 PowerBuilder 应用的人员享受到 Web 无处不在的优势。

.NET Smart Client 具有以下特征：

- “偶链接”。也就是，它不需要必须具有持久的数据库连接。得益于其增强的 MobiLink 数据库同步支持（最初于 PowerBuilder 10.0 中引进）组件，PowerBuilder 11.1 具有偶链接的功能。
- 此外，PowerBuilder 11.1 还增加了可更新的 Web Service 源数据窗口，该窗口通过 Web Service 操作就可进行 CRUD（即 Create(创建)、Read(读取)、Update(更新)和 Delete(删除)）动作，而不需要持久的数据库连接。
- Web service 源数据窗口使 .NET Smart Client 可以通过 Web service 进行通信。作为利用 Web service 的超级集合，.NET Smart Client 还能实现对网络资源的利用。
- Smart Client 最大限度提高本地客户端资源的性能。用户对此感兴趣吗？是的，犹豫不决的用户已经从瘦客户端回归，对于那些从 1996 年以来一直在望眼欲穿地等待浏览器内的丰富的用户界面的商业用户而言尤为如此（请注意在“胖客户端”转向“富客户端”过程中的变化）。
- 最值得一提的是，.NET Smart Client 将通过 Smart Updates 提供始终最新的应用。PowerBuilder 11.1 利用 Intelligent Updater 功能实现这一点，从而使用户能够自动为所有单个用户安装、发布和更新 .NET WinForm 部署的 PowerBuilder 应用——而无需开发人员增添一行代码。

那么，.NET Smart Client 是由什么组成的？简言之，它是由任何具有一种或多种上述特征的应用组成的。这使开发部门能够确定什么样的 Smart Client 特性和技术适合您的应用。



Intelligent Updater

PowerBuilder 11.1 的 Intelligent Updater 是一种新型应用部署技术，该技术使部署 PowerBuilder .NET Windows Forms 应用就像部署 Web 应用一样简便。该功能特性是基于 .NET Framework v2.0 中的“ClickOnce”技术。Intelligent Updater 旨在解决部署和更新客户端/服务器应用以及 n 层解决方案的客户端应用的难题。

需要注意的是, Intelligent Updater 只对安装/发布/更新 .NET WinForms 部署的 PowerBuilder 应用有帮助。至于系统要求, 用户工作站只需要在系统路径中安装 .NET 可重新分发的框架。您可从以下网站下载这一框架: www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&displaylang=en。

同样, 您的开发用机器中也需要 .NET 可重分发的框架和 .NET 2.0 Framework SDK。SDK 的 %SDK%\bin 目录应位于 PATH 中, 并且需要确认该 %PBNET_HOME% (PowerBuilder 11.1 安装目录的 .\DotNET 子目录) 已在 PowerBuilder 安装期间设置为系统环境变量。最后, 还要能够访问 IIS (Internet Information Server) 5.0 或更高版本的实例。

部署模型

WinForm 通过 HTTP、FTP 或 UNC (Universal Naming Convention, 如使用 Windows “共享”资源) 向集中的 IIS 服务器发布, 并由用户从该服务器安装和更新。最初安装 PowerBuilder WinForm 应用也可选择使用 CD 或 DVD 光盘。

总体工作流程

下面, 我们概括一下 Intelligent Updater 的总体工作流程:

- 开发人员使用现有的 PowerBuilder 应用在 Windows Forms 项目中设置发布属性, 部署/编译为 .NET Windows Forms 应用, 然后发布到 IIS 的一个实例中。
- 用户浏览应用发布的 Web 页面, 该页面透明地请求部署描述文件; 然后下载、安装和选择性执行应用。
- 开发人员修改并再次发布应用。
- 用户运行应用, 这将再次请求服务器提供部署描述文件; 看到提供更新版程序; 然后下载、安装和执行更新的应用。
- 开发人员或管理员可回滚服务器上的应用版本, 这将导致用户工作站上的应用也恢复到以前的版本。
- 用户也可从控制面板中的“Add or Remove Programs”回滚到以前的版本, 或者从其工作站全部卸载应用。



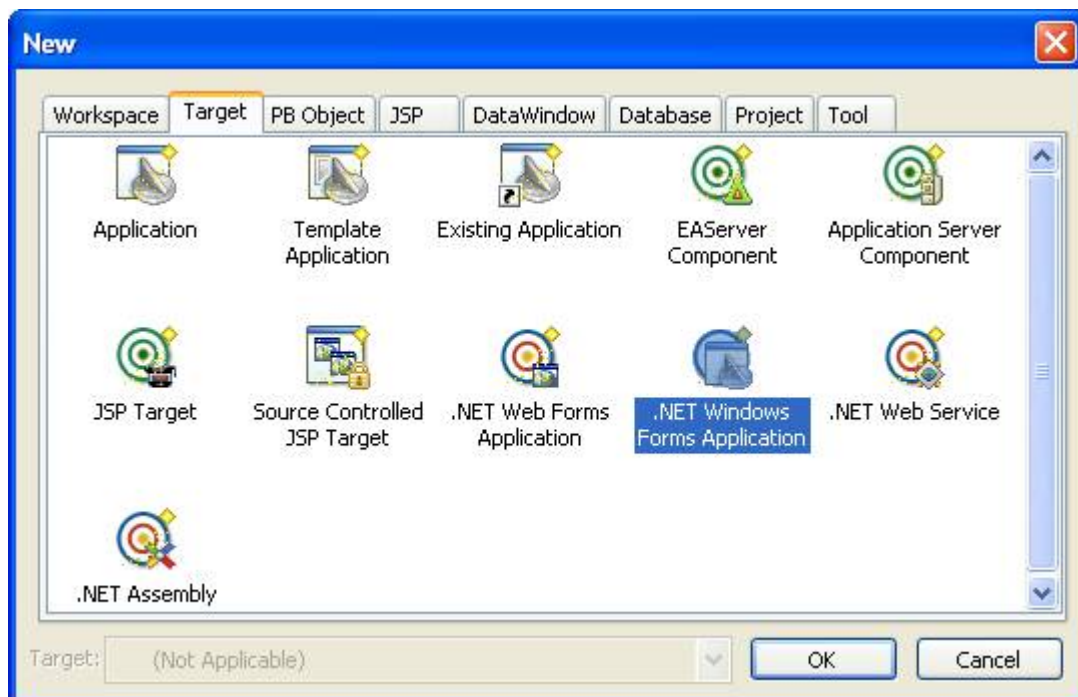
详细信息：演示

我们来具体考查这一流程，首先从现有的 PowerBuilder 应用开始。出于演示目的，我们将进行最简单的后期修改。“About”窗口的静态文本控制的背景颜色最初将是红色。我们将遵循一个完整的 RGB 进程来进行。

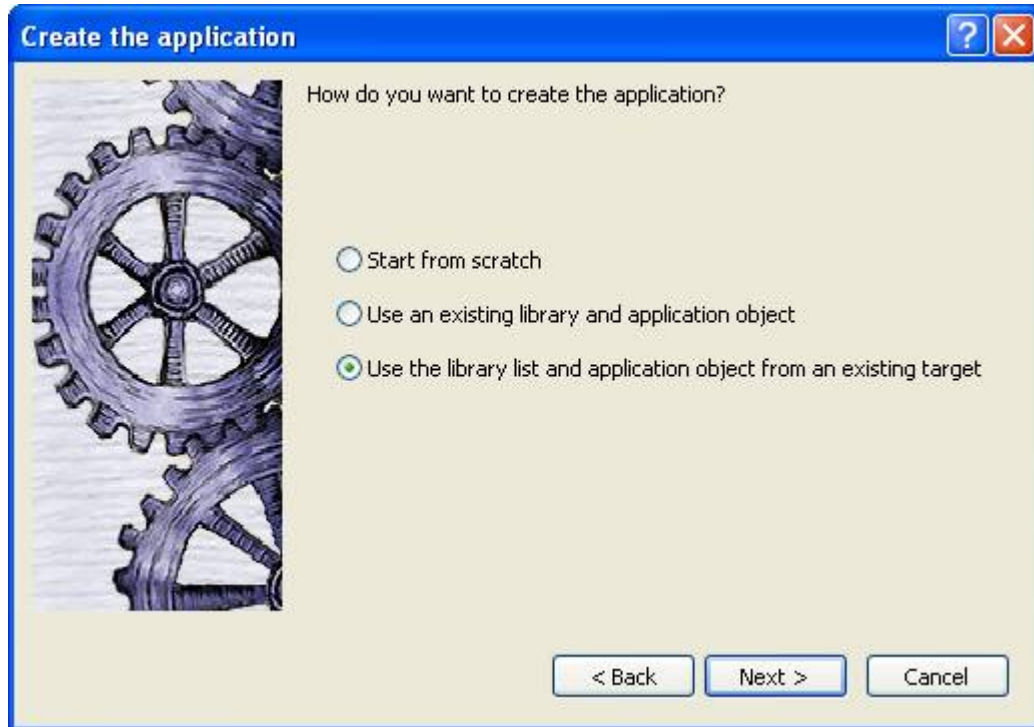
第一步，使用 PowerBuilder 11.1 将我们的应用部署为 WinForm 应用，发布该应用，并让用户下载和安装该应用。然后，我们修改应用和发布，使用户客户端自动检测新版本，并下载和安装新版本。

Intelligent Updater 的基本用法

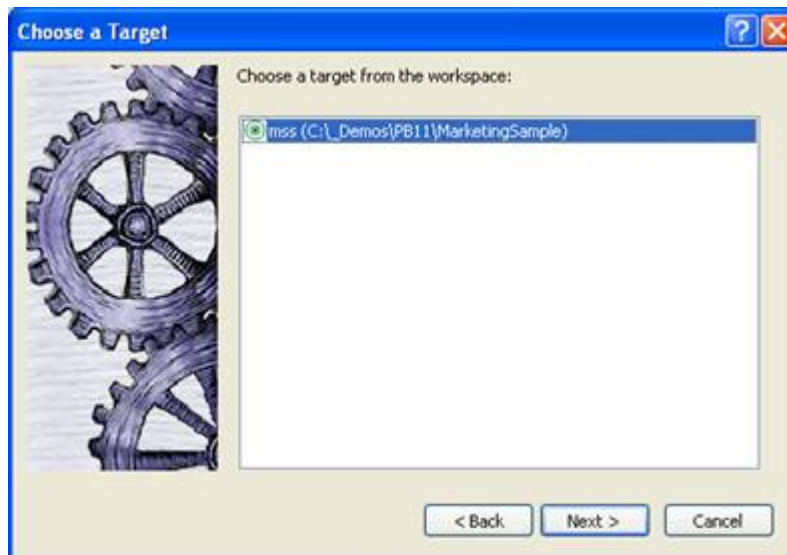
首先，我们打开一个现有 PowerBuilder 应用的工作区。在本演示中，我们使用 PowerBuilder 11.1 (MarketingSample) 提供的一个代码例子。我们需要创建一个新的 .NET Windows Forms Application 对象。

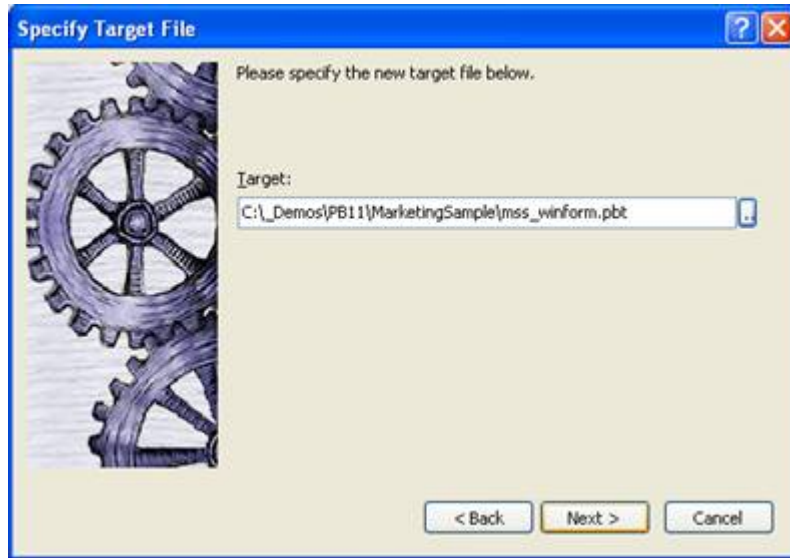


在“Create the application”向导窗口中，我们选择“Use the library list application object from an existing target”。

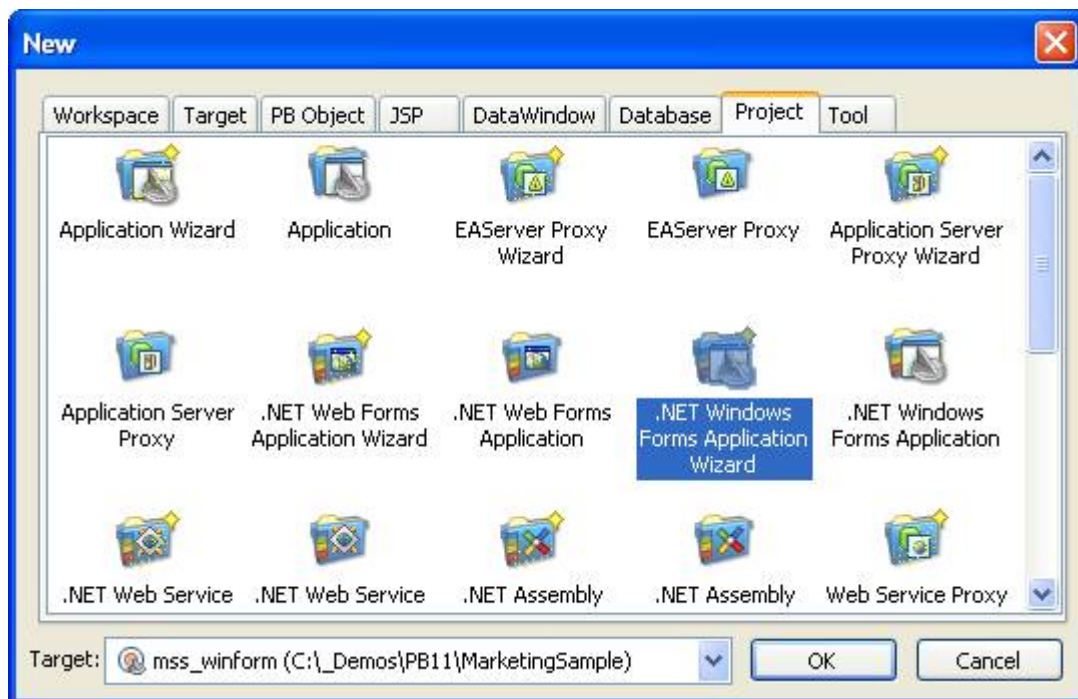


在下一个窗口中，我们选择唯一可用的对象“mss”，并在下一个窗口中接受目标文件的默认名称和位置 (*.pbt)。为了便于说明，我们将选择不用该向导创建 Project 对象，然后选择“Finish”。





接下来，我们将在新 *mss_winform* 对象上点右键，从弹出的菜单中选择“New...”，然后从“Project”标签页选择“.NET Windows Forms Application Wizard”。单击“OK”。



在初始“about”向导窗口上单击“Next>”，然后选择 *mss_main.pbl* 作为存储最终的 Project 对象的库。在下一个窗口中，接受默认的 Project 名称 *p_mss_winform*。单击“Next>”。



Specify Project Object

A project object will be built that you can use to build and deploy your application. Enter a name for the new project.

Project:

< Back Next > Cancel

在下一个窗口中，接受默认的产品名称、默认的可执行文件名称以及默认的产品版本值。单击“Next”。

Specify Application General Information

Windows Forms application general information:

Product name:

Product executable file name:

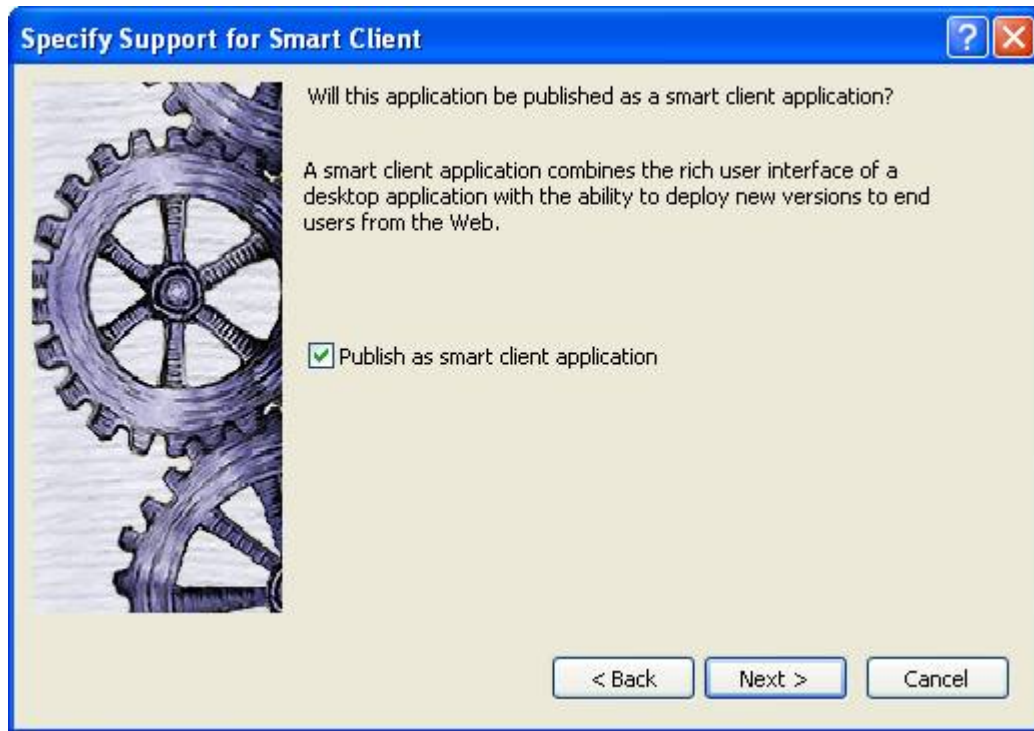
Product version:

Major	Minor	Build	Revision
1	0	0	0

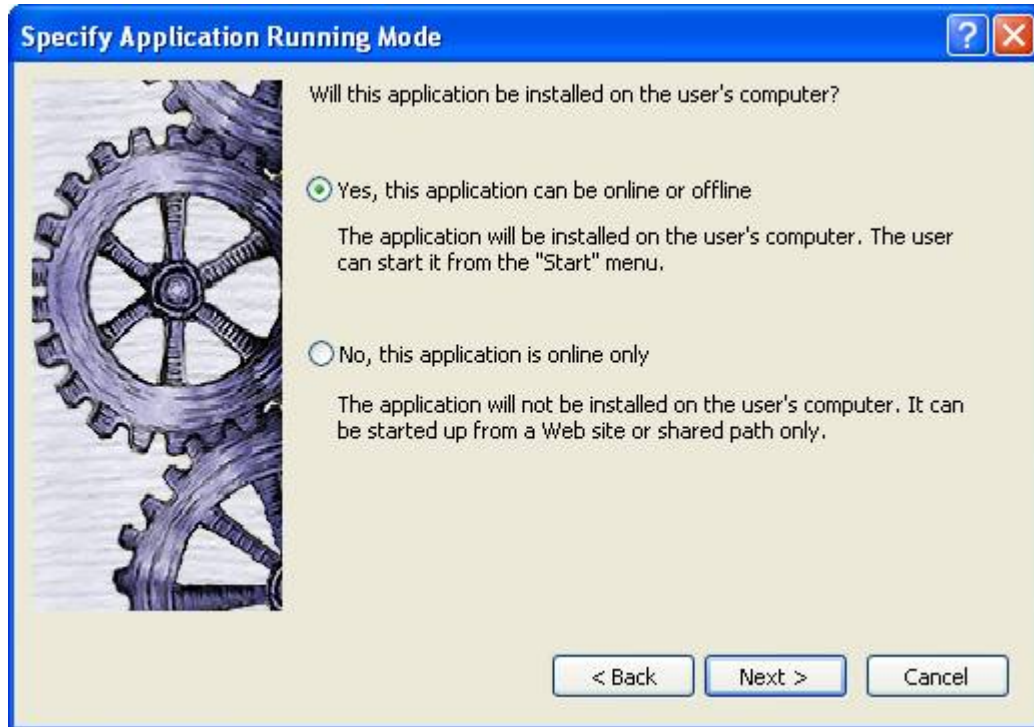
< Back Next > Cancel



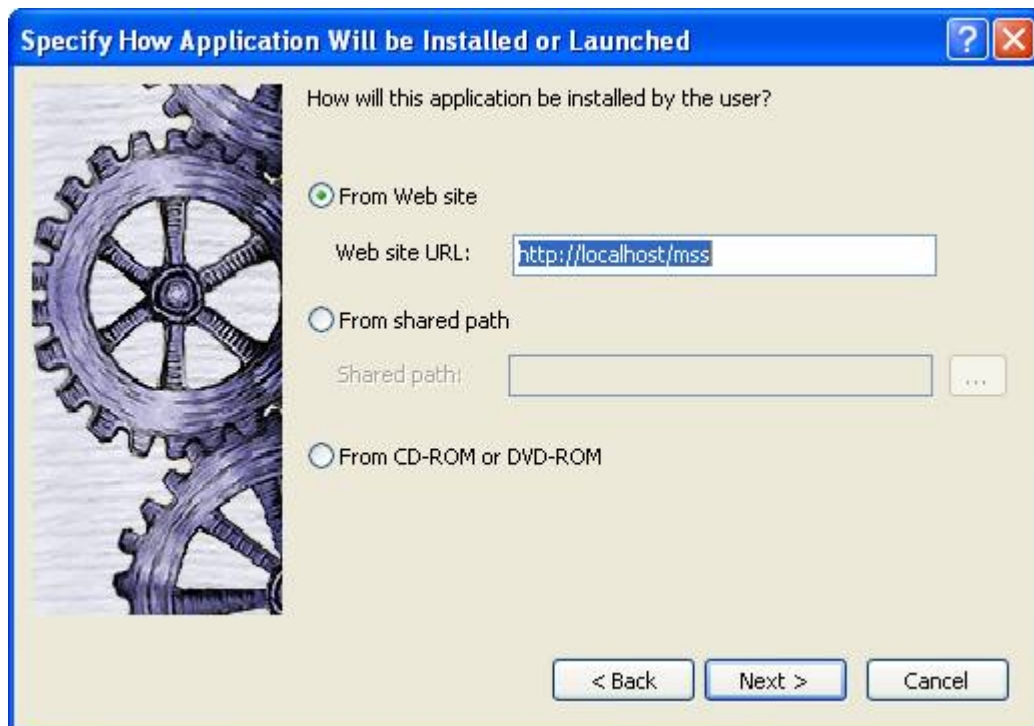
在下一个窗口中，选择支持该 WinForm 应用的安装，并通过选取“Publish as smart client application”复选框更新为 Smart Client。单击“Next>”。



选择“Specify Application Running Mode”窗口上的第一个单选按钮，允许该应用在线或脱机运行。单击“Next>”。

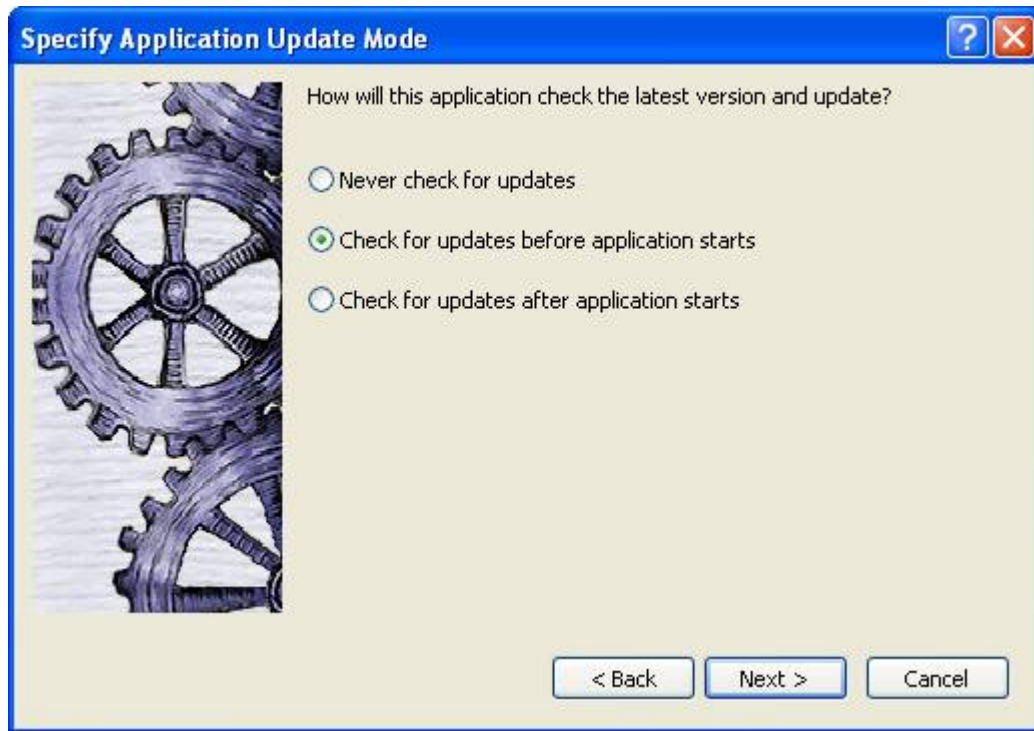


在下一个窗口中，选择用户如何安装这一应用：从网站、通过共享的 UNC 路径或者从 CD-/DVD-ROM 安装。选择默认网址 <http://localhost/mss>。单击“Next>”。





现在指定检查是否需要更新的时机。选择默认值“Check for updates before application starts”。单击“Next>”，然后单击最后的向导窗口上的“Finish”。



现在，您有了一个可重复使用的 Windows Forms Project 对象，该对象能够将 PowerBuilder 应用部署为 .NET WinForm 应用，并通过 Intelligent Updater 使用 Smart Client 技术发布该应用。

▶ 部署、运行和发布 .NET Windows Forms 应用

要将您的应用部署/编译为 .NET Windows Forms 应用，单击项目的 painter 工具栏中的“Deploy”工具条按钮。





请注意输出窗口中的输出消息。要运行 .NET WinForm 应用，单击项目的 painter 工具栏中的“Run Project”工具条按钮。



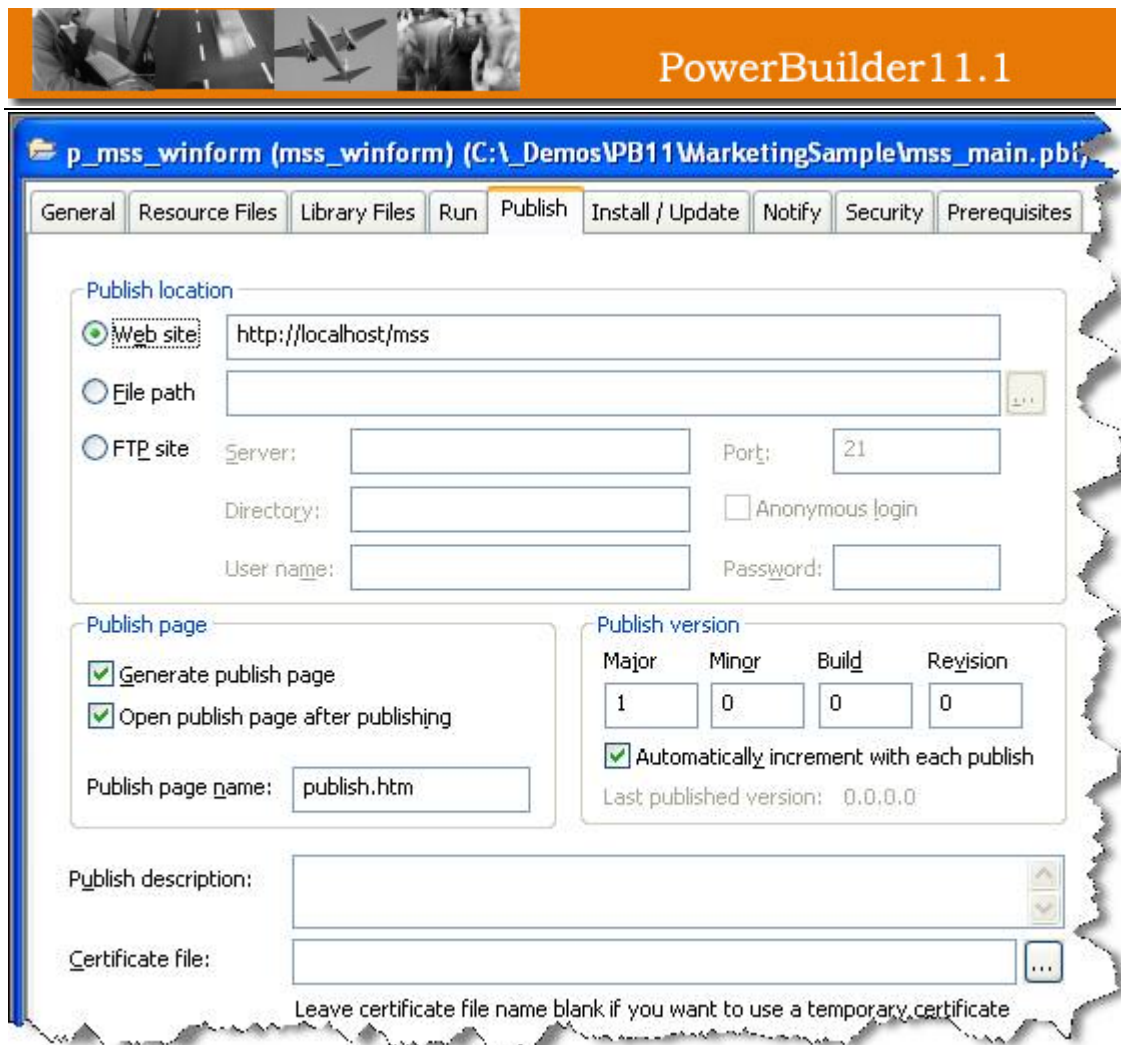
您会发现，示例的每一个功能都像是在本地操作一个 WinForm 应用。如果您在使用 Marketing 代码示例项目，那么在窗口 painter 中打开窗口类 `w_mss_about`。将静态文本控制的背景颜色改为红色，以备跟踪将使用 Intelligent Updater 发布的修改。



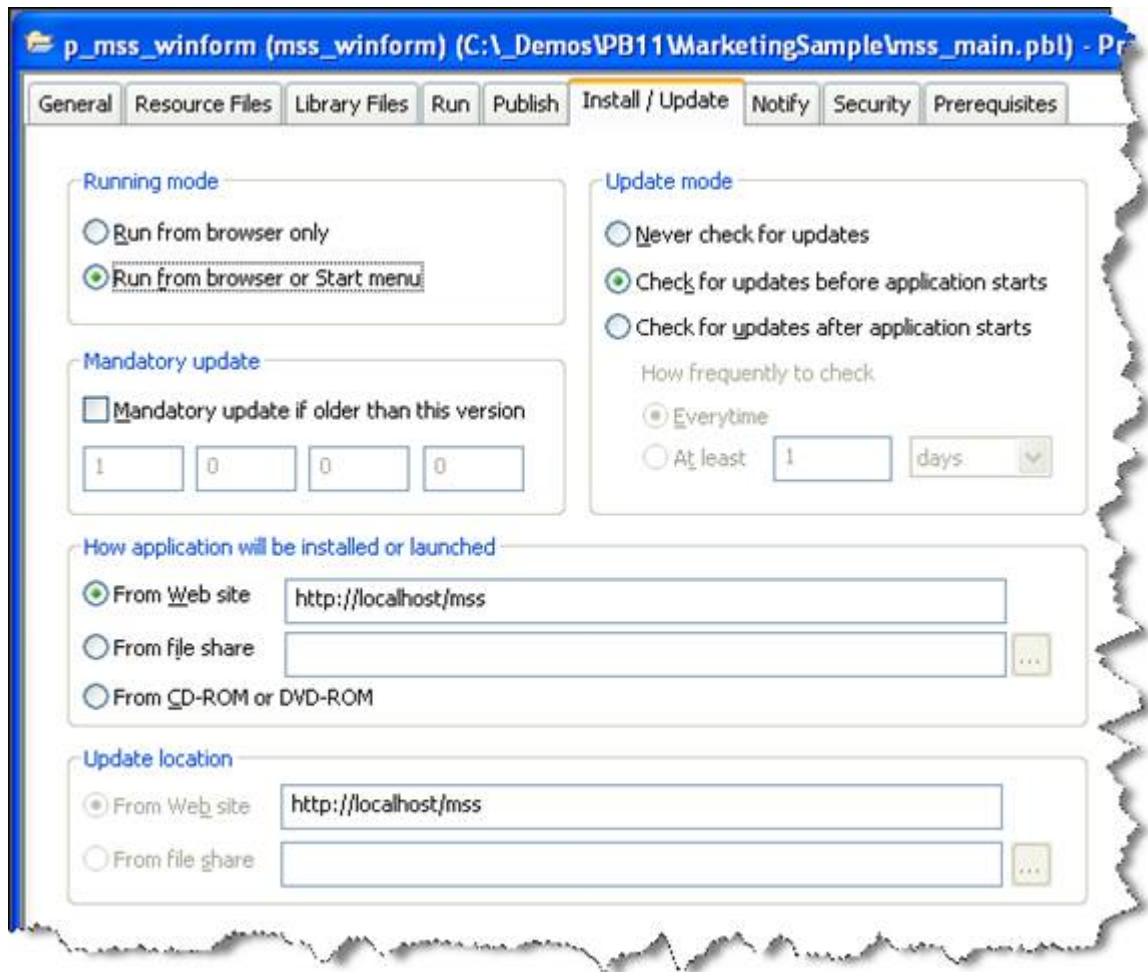
▶ Windows Form Project 对象

在 `p_mss_winform` 项目中选择 Publish 标签页。注意“Publish location”组合框和“Publish version”组合框中的值是来自“.NET Windows Forms Application Wizard”的。此外，注意“Publish”组合框中的参数的默认值。如果您愿意，PowerBuilder 将生成带有超链接的发布页面。在应用发布后，这一超链接将提供给用户。这一点对于原型开发非常有用，也很有效，它假定您用自己的方式来分发 URL，最终用户访问该 URL 来使用该解决方案的安装和更新功能。

在这一例子中需要注意的是，我们将 Certificate 文件的值留作空白。如果您目前尚不具备，应尽快申请证书，使用户确信您是合法的、已注册的应用程序发布源。



选择 Project 对象的 *Install/Update* tabpage。需要注意的是，“Update mode”组合框表明了向导中选择的值。此外，还要注意的，Install/Launch 组合框以及“Update location”组合框中的值是根据在向导中选择的位置值默认的。需要特别注意“Mandatory update”组合框，该框可防止用户使用低于规定的最低版本的旧应用。



现在开始发布我们的应用，单击项目的画板上的“Publish Project”工具条图标，注意输出窗口中的“Publish version”值。如果不指定值，浏览器将打开到默认的 *publish.htm* 页面。页面上显示的版本值与输出窗口中显示的版本值匹配。



现在，你就可以作为用户来使用发布页面。单击页面中的“Install”按钮。状态栏中显示的是发送给用户的地址。

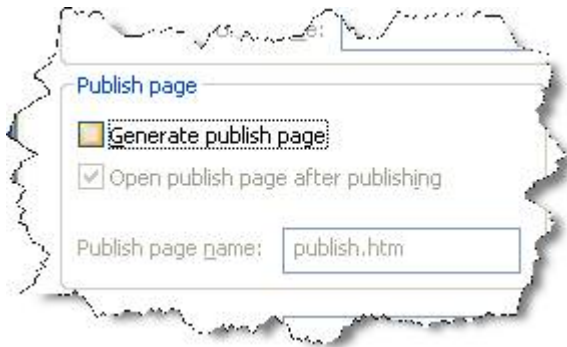




单击“Install”按钮。应用将自动下载、安装和运行。



打开 Windows Start 菜单，如果您保留了这些默认值，就应看到 Sybase, Inc. 的一个程序组，包括级联菜单项/“mss”的快捷方式。打开应用的 Help>About... 窗口，注意静态文本控制的红色背景颜色。关闭应用。如果您还未关闭，就关闭显示 *publish.htm* 的浏览器。返回到 Window Painter，然后在我们的修改序列中将静态文本控制的背景颜色改为绿色。保存并关闭窗口类。如果您愿意，可修改 *p_mss_winform* 的 Ggeneral tabpage 的公司名称和产品名称值。在 Publish tabpage 标签上，取消对“Generate publish page”复选框的选择。



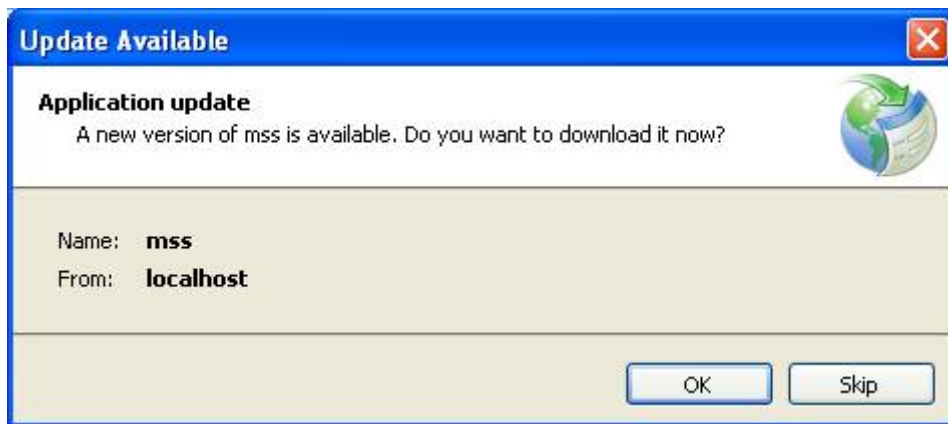
单击“Publish Project”工具条按钮。如果观察输出窗口，就会看到发布的版本值自动更新了。

```

----- Publish: Publish of p_mss_winform
Checking publish setting...
Deployed executable "C:\_Demos\PB11\MarketingSample\mss_winform_Winform
Checking project settings...
Project is being deployed to output path: C:\_Demos\PB11\MarketingSample\ms:
Generating PBD files...
Copying PBD files to output path: C:\_Demos\PB11\MarketingSample\mss_winfor
Deploy succeeded.
Publishing application to http://localhost/mss. Publish version is 1.0.0.1

```

回到用户角度，使用 Windows Start 菜单再次运行应用范例。您的应用已经透明地下载了应用的描述文件，并检测是否有可用的新版本。作为用户，您会看到提供新版本应用的通知窗口。单击 OK 下载和安装更新。



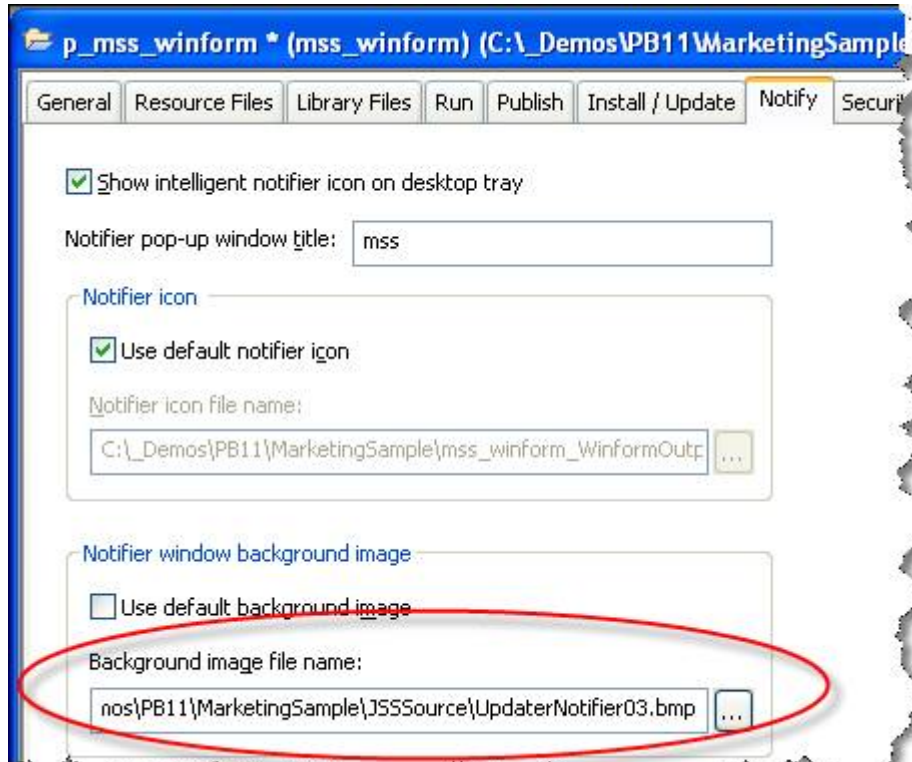
打开“About”窗口，瞧，您会看到文本的背景颜色现在变成了绿色——“G”或者我们的 RGB 序列中的另一种状态。您还将注意到，如果您过去曾使用应用的开始菜单快捷键的值，现在该选项已改变。关于应用的运行以及 PowerBuilder IDE 的情况，我们将留到下一部分讨论。

► Update Notifier 的功能

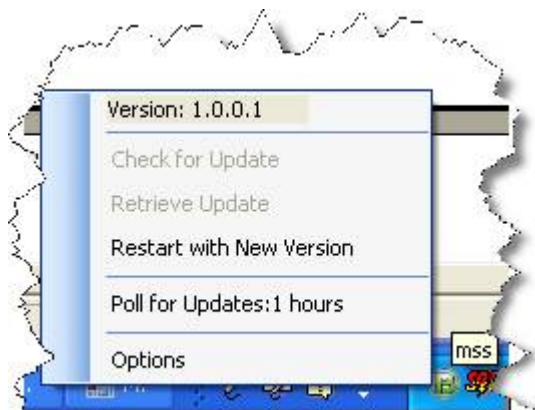
这一功能在用户端的两个地方实现。在运行时，它包括一个从系统托盘发起的弹出窗口，用于通知用户各种应用更新状态；此外，它还包括一个弹出菜单，使用户能够采取某些措施并修改与用户应用的更新相关的某些选项。



在 PowerBuilder 11.1 IDE 中，将 *w_mss_about* 上的背景颜色改为蓝色，保存并关闭 *w_mss_about*。我们还可导航到 Notifier 标签页，为 Notifier 窗口选择背景图。Notifier 图标的默认显示尺寸是 16 x 16Px，这里我们选择 UpdaterNotifier03.bmp，它的尺寸为 120*120。我们还可将 Notifier 窗口的标题改为“MSS”，再次发布应用，并且在发布完成后激活其运行实例。



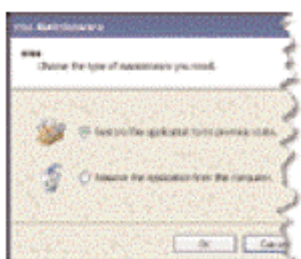
在系统托盘中，右键单击应用的图标。从弹出菜单中选择“Check for Update”。这时会出现 Notifier 窗口，通知用户有可用的更新。再次右击图标，从弹出菜单中选择“Retrieve Update”。Notifier 窗口通知更新已被下载和安装。再次单击图标，选择“Restart with New Version”，系统将重启。



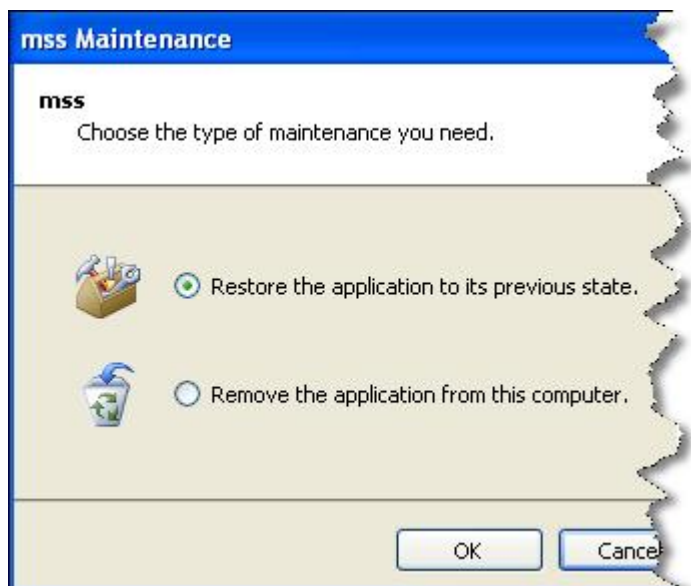
▶ 返回到以前的版本

现在我们讨论用户或管理员如何将公布的应用回滚到以前的版本。我们假定由于某种原因，目前发布的应用版本存在一些缺陷，需要使用以前的版本。用户工作站和服务器保存着应用过去的版本。

我们要执行三种任务：以用户身份返回到以前的版本；以管理员的身份返回到任何以前的版本；然后，再以用户的身份在客户端进行卸载，完全从工作站删除应用。



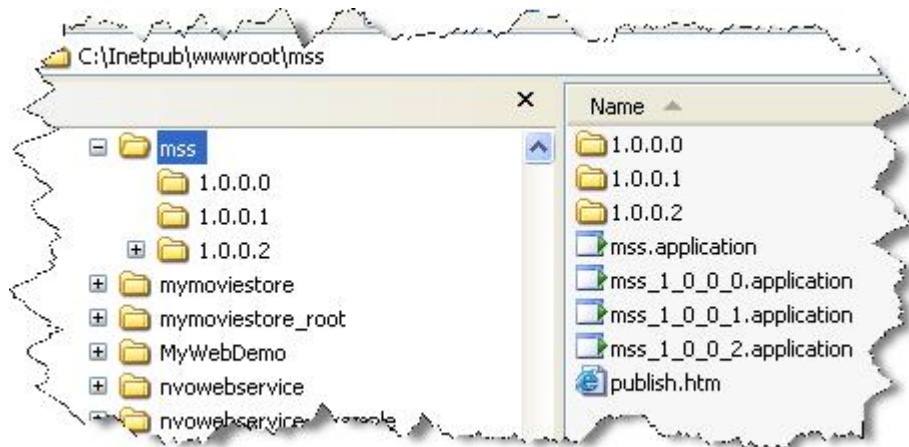
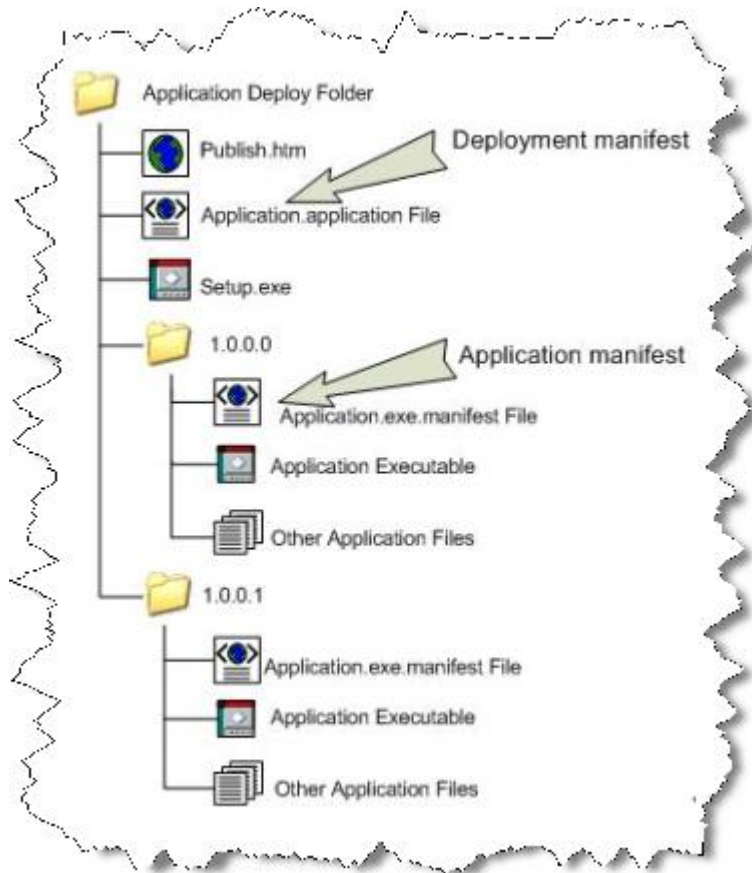
我们使用 Start 菜单打开 Settings > Control Panel。我们选择 “Add or Remove Programs”，然后向下滚动鼠标至 “Change/Remove” 按钮，选择 “Restore application to its previous state” 按钮，单击 “OK”。当显示回滚确认消息时，运行应用，打开 “About” 窗口，注意，静态文本已经恢复到绿色或者我们 RGB 序列中的另一种状态。



现在，作为管理员，我们希望将应用返回到以前的某种版本。如果 IIS 有默认的安装位置，就使用 Windows Explorer 导航到示例应用的有效目录下：C:\Inetpub\wwwroot\mss。注意



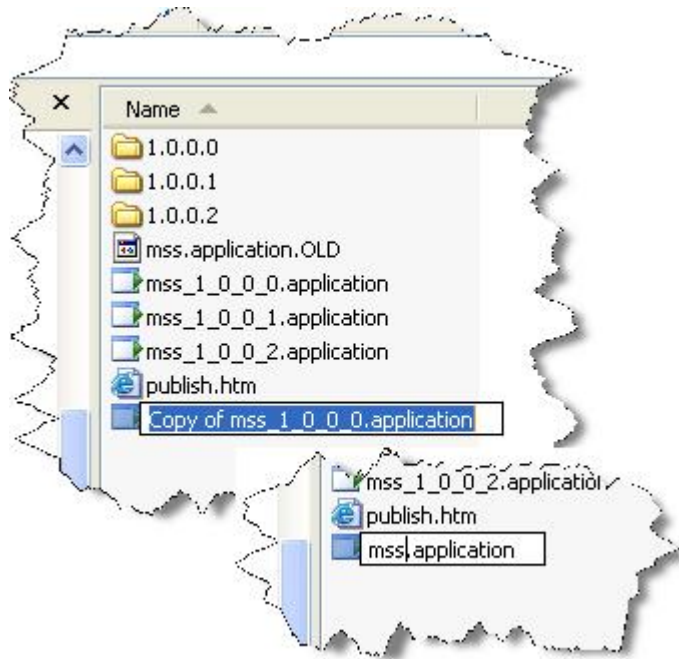
名为“1.0.0.0”、“1.0.0.1”等的文件夹。



这些文件夹包含过去及当前版本的应用。文件“mss.application”是 manifest 文件，其内容（顺便说明一下，它采用 XML）表明在保存的多个版本中哪个是向客户端工作站公布的有效版本。我们将过去的 manifest 变成现行 manifest。将“mss.application”重命名为“mss_application.OLD”。复制“mss_1_0_0_0.application”文件，并将它重命名为

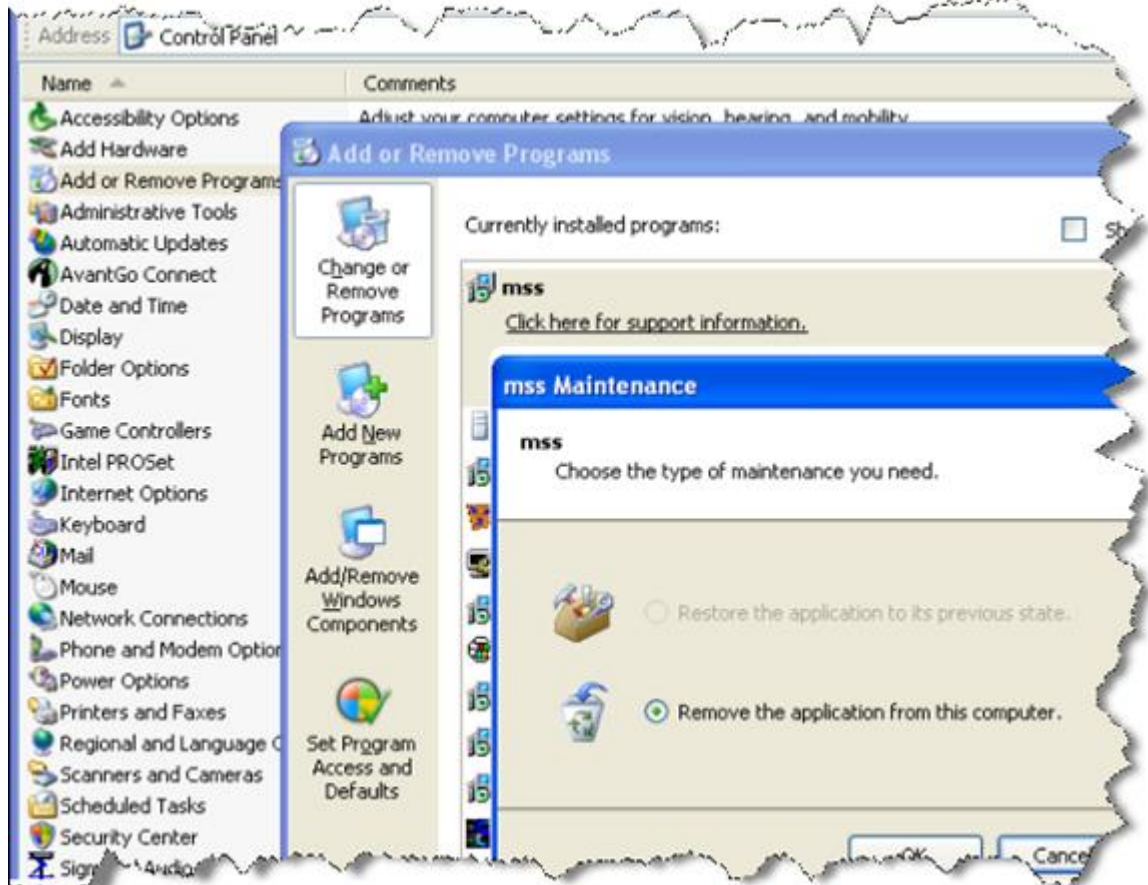


“mss.application”。这将该文件指定为当前有效的 manifest，而不是作为以前的版本。其内容执行过去的 1.0.0.0 版。



作为用户，我们从 Start 菜单运行应用，并被告知有可用的更新。接受下载，运行应用，注意现在“About”窗口显示为最初在我们的 RGB 修改序列中采用的红色值。不论您是有 5 个用户还是 5000 个用户，他们都会自动收到这一消息。

要完成这一部分，请退回到 Control Panel 的“Add or Remove Programs”部分，并以用户的身份完全从工作站卸载该应用。



► 位置、位置、位置：发布、安装和更新

我们来探究.NET Smart Client 部署的位置的概念——实际上有三个。

发布 位置是从开发人员的角度而言。开发人员（或者也可能是管理员）希望在哪里发布应用的可执行程序？开发人员可能选择的应用发布方式：通过 HTTP 或 FTP，甚至使用 Universal Naming Convention (UNC) 向 Windows 共享资源/目录发布。

与此相反，**安装** 位置是从用户角度而言。用户从哪里访问可执行程序以便进行初次安装？对于用户，可通过 HTTP 或 UNC 执行安装，如果您的部门愿意，甚至可以从 CD/DVD-ROM 安装。

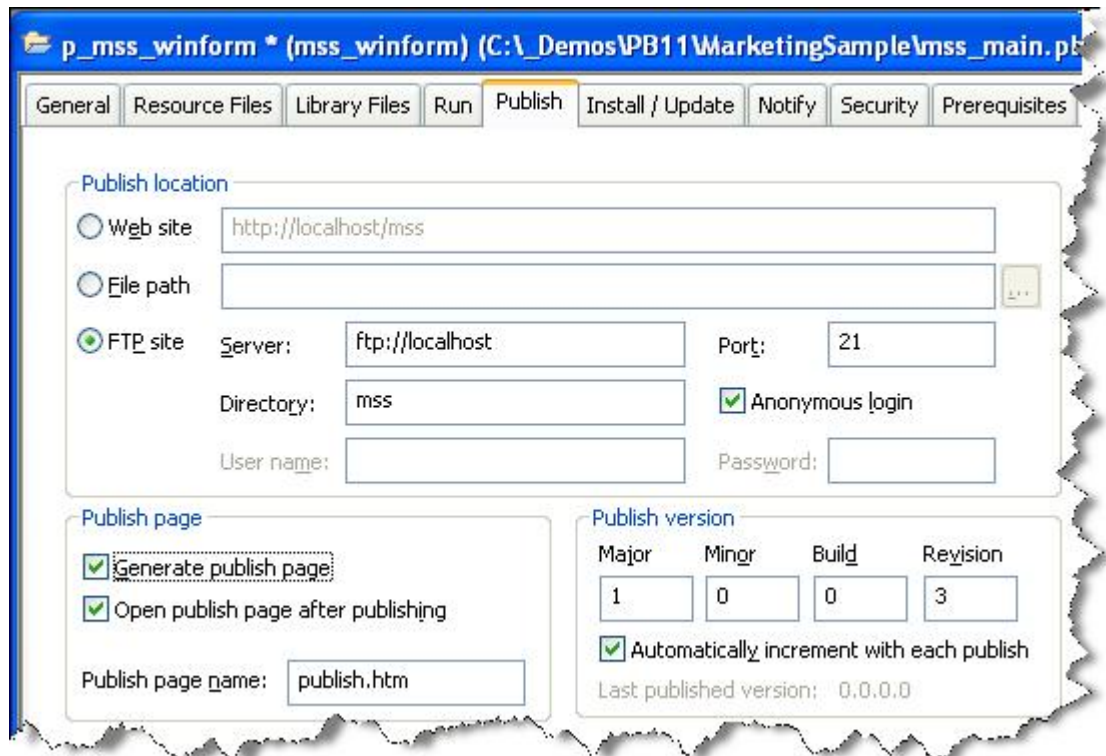
第三，**更新**位置是从用户角度而言，也可以说，实际上是从应用角度而言。应用在检查时将从哪里透明地获得 manifest 文件，然后下载应用的更新版？**更新**位置通常与**安装**位置相同。这一规则确实存在例外之处，那就是如果选择使用可移动介质 CD 或 DVD-ROM 进行安装。安装位置可以是局域网、互联网或存储介质。然而，由于显而易见的原因，Intelligent Updater 不能使用存储介质进行更新，因此更新必须通过局域网/互联网进行。



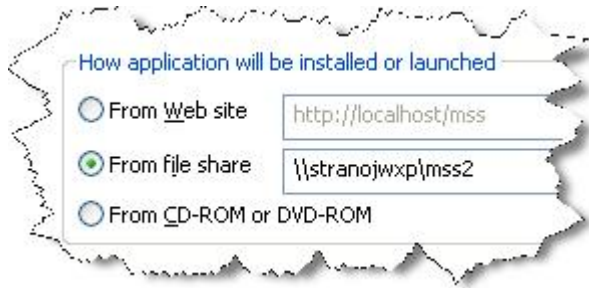
我们来看这是如何执行的。作为开发人员，我们使用 FTP 站点作为**发布**位置，并使用户通过 UNC **更新**位置访问安装程序和进行更新。为了准备该方法的下一序列，您可能需要仔细查阅以下资源：

- “How to: Create and Configure FTP Sites in IIS”
<http://msdn2.microsoft.com/en-us/library/6ws081sa.aspx>
- “How to configure file sharing in Windows XP”
<http://support.microsoft.com/kb/304040>
- “Windows XP Networking: Sharing in the Home Office”
http://www.microsoft.com/windowsxp/using/networking/expert/honeycutt_august13.msp

打开 p_mss_winform 对象，选择“Publish”标签页，并选择“FTP site”单选按钮。在 Server 文本框中输入 ftp://localhost，在 Directory 文本框中输入“mss”（实际上是该演示创建的 C:\mss 目录）。这个 FTP 的网站允许匿名登录，因此需要勾选“Anonymous login”复选框，并且确保在“Publish page”组合框中勾选了“Generate publish page”和“Open publish page after publishing”复选框。



选择“Install/Update”标签页，并在“*How application will be installed or launched*”中选择“From file share”单选按钮。并输入 \\stranojwpx\mss2（该位置也将链接到 C:\mss）。



现在将通过 FTP 发布应用到目录 (C:\mss)，该目录也被用于发布更新文件。我们需要将该共享资源命名为“mss2”，以避免名称冲突问题。该目录的共享权限只需要“Read”即可。

在 PowerBuilder 11.1 IDE 中，单击“Publish Project”工具条按钮。请注意，您的 FTP 位置在通知窗口以及输出帧中显示。发布页面应在浏览器中自动打开。浏览器地址栏中的 URL 在句法上是一个 UNC 位置。当您的鼠标光标经过“Install”按钮时，注意状态条中显示的位置是 manifest 文件的完全适用于作为 UNC 的路径。以用户身份单击 Security Warning 窗口上的“Install”按钮，如果未使用证书文件就会出现该窗口。

这里需要注意的是：如果在原型开发期间不具备有效的 TCP/IP 链接，那么 Windows 网络可能不允许激活共享目录的共享。如果这样，就连接到 LAN 或登录到无线网络；如果您在旅行中，没有用于演示的连接，那就购买一个 pocket WiFi 接入设备、路由器或集线器（当然，在生产环境假定所有点都是连通的，因此这一提示也就毫无意义了）。作为用户，您将看到应用自动执行。您可以继续试验变换发布/安装/更新的位置会有什么变化。

PowerBuilder 为您打开应用.NET 之门

PowerBuilder 不仅利用 .NET 的优势，而且还免除了您建立基础框架的艰巨工作，使您能够重点关注架构、业务规则和企业解决方案，而非低级别的基础设施构建。毕竟，这一直是 PowerBuilder 所致力于从事的工作——获得 Windows 环境中的可用资源，让您摆脱不必要的实施细节，使您成为最高效的开发人员。