为备份Windows Phone应用数据建立、使用托管的服务器

本文阐述了如何为备份Windows Phone应用数据建立、使用托管的服务器。

中文

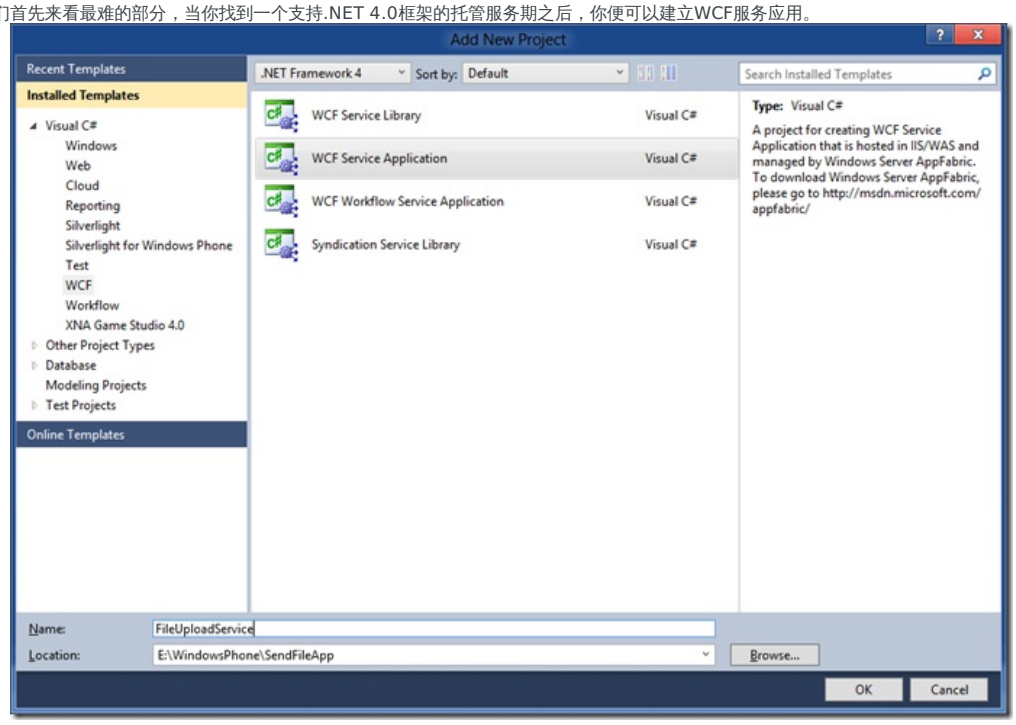
简介

将你的应用数据存储在服务器上是很有意义的（例如更换设备后你可以恢复数据以便保持访问）。一种常用的备份解决方案就是SkyDrive，但是这会在用户的SkyDrive账户上产生奇怪的文件夹。就个人而言，我发现将数据存储在个人云端或服务器上更为专业。 本文提供了步骤详尽的为Windows Phone应用建立Windows Communication Foundation (WCF)备份服务的指南，该备份托管在一个便宜的服务器上，例如godaddy.com (提笔之时，该服务要价5美金/月)。 Windows Communication Foundation是.NET框架的一部分，它为快速建立基于服务器的应用提供了一个统一的编程模型 ，这些应用以类似的方式在网络上通讯。 本文包含两种主要的备份情景：

- 在你的服务器上执行服务
- 执行客户端代码以发送你的数据

在你的服务器上执行服务

1. 我们首先来看最难的部分，当你找到一个支持.NET 4.0框架的托管服务期之后，你便可以建立WCF服务应用。



2. 将 IService1.cs 重命名为 IFileUploader.cs
3. 用以下代码替换 IFileUploader.cs中代码：

```
using System.Runtime.Serialization;
using System.ServiceModel;

namespace FileUploadService
{
    [ServiceContract]
    public interface IFileUploader
    {
        // Returns null when there is no error, otherwise it is the exception message.
        [OperationContract]
        string Upload(UploadFile uploadFile);
    }

    [DataContract]
    public class UploadFile
    {
        // Don't forget to set the variable FileUploadDirectoryWithReadWritePermission in your Web.config and set read/write perm
        // RelativeDirectory should be in the form as /Path1/Path2/ or null for the root directory.
        [DataMember]
        public string RelativeDirectory { get; set; }

        [DataMember]
        public string FileName { get; set; }

        [DataMember]
        public byte[] Content { get; set; }
```


(c) Copyright Nokia 2013 All rights reserved

```
    }
}
```

该接口仅包含Upload() 方法。Upload需要UploadFile，它包含3个属性： RelativeDirectory，FileName和Content。

5. 将 Service1.svc 重命名为 FileUploader.svc.
6. 用以下代码替换 FileUploader.svc 中代码：

```
using System;
using System.Configuration;
using System.IO;
using System.Web.Hosting;

namespace FileUploadService
{
    public class FileUploader : IFileUploader
    {
        #region IFileUploader Members

        public string Upload(UploadFile uploadFile)
        {
            string message = null;

            try
            {
                string path = HostingEnvironment.MapPath(string.Format("~/{0}", ConfigurationManager.AppSettings["FileUploadD

                if (path != null)
                {
                    if (uploadFile.RelativeDirectory != null && uploadFile.RelativeDirectory.StartsWith("/"))
                    {
                        path = string.Concat(path, uploadFile.RelativeDirectory);
                    }

                    if (!Directory.Exists(path))
                    {
                        Directory.CreateDirectory(path);
                    }

                    if (!string.IsNullOrEmpty(uploadFile.FileName))
                    {
                        using (FileStream fileStream = File.Open(Path.Combine(path, uploadFile.FileName), FileMode.Create))
                        {
                            using (var binaryWriter = new BinaryWriter(fileStream))
                            {
                                binaryWriter.Write(uploadFile.Content);
                            }
                        }
                    }
                }
            }
            catch (Exception exception)
            {
                message = exception.Message;
            }

            return message;
        }

        #endregion
    }
}
```

7. 用以下代码替换 Web.config中代码：

```
<?xml version="1.0"?>

<configuration>

    <appSettings>
        <add key="FileUploadDirectoryWithReadWritePermission" value="/backup/" />
    </appSettings>

    <system.web>
        <compilation debug="true" targetFramework="4.0" />
        <customErrors mode="Off" />
    </system.web>

    <system.serviceModel>

        <behaviors>
            <serviceBehaviors>
                <behavior name="FileUploadBehavior">
                    <serviceMetadata httpGetEnabled="true" />
                </behavior>
            </serviceBehaviors>
        </behaviors>

        <services>
            <service behaviorConfiguration="FileUploadBehavior" name="FileUploadService.FileUploader">
                <endpoint address="" name="basicHttpStream" binding="basicHttpBinding"
                bindingConfiguration="httpLargeMessageStream" contract="FileUploadService.IFileUploader" />
                <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" />
            </service>
```


(c) Copyright Nokia 2011 - All rights reserved

```
        </services>

        <bindings>
            <basicHttpBinding>
                <binding receiveTimeout="00:10:00" sendTimeout="00:10:00" openTimeout="00:10:00" closeTimeout="00:10:00"
                name="httpLargeMessageStream" maxReceivedMessageSize="2147483647" maxBufferSize="2147483647">
                    <readerQuotas maxArrayLength="2147483647" maxStringContentLength="2147483647" />
                </binding>
            </basicHttpBinding>
        </bindings>

        <serviceHostingEnvironment multipleSiteBindingsEnabled="true" />

    </system.serviceModel>

    <system.webServer>
        <modules runAllManagedModulesForAllRequests="true" />
    </system.webServer>

</configuration>
```

该文件包含两个重要部分：

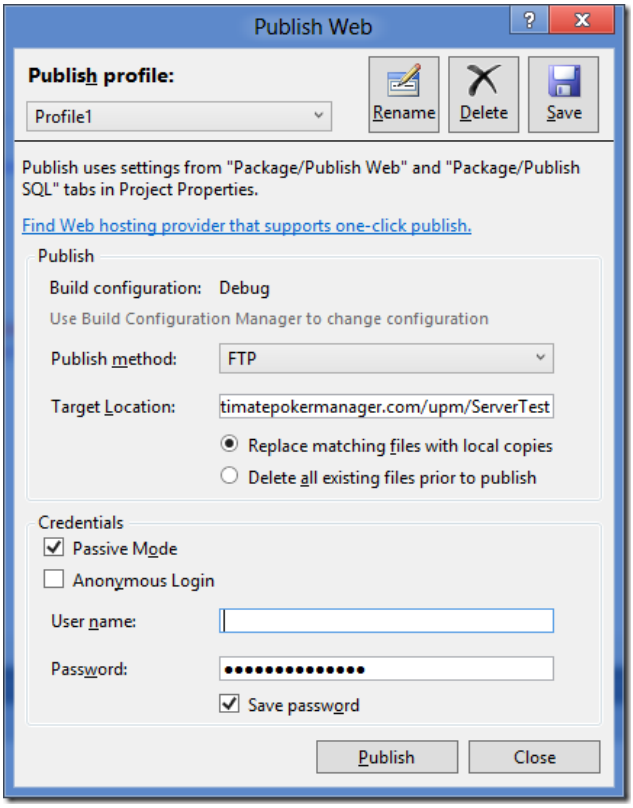
1. - FileUploadDirectoryWithReadWritePermission参数被 FileUploader.svc 文件使用，之后会被你的公共文件夹使用。

```
<appSettings>
    <add key="FileUploadDirectoryWithReadWritePermission" value="/backup/" />
</appSettings>
```

   - 该部分关于服务参数的长度：你没必要完全了解。只需让它工作就好了，如果你真的好奇，建议你阅读MSDN相关文档。

```
<bindings>
    <basicHttpBinding>
        <binding receiveTimeout="00:10:00" sendTimeout="00:10:00" openTimeout="00:10:00" closeTimeout="00:10:00"
                name="httpLargeMessageStream" maxReceivedMessageSize="2147483647" maxBufferSize="2147483647">
            <readerQuotas maxArrayLength="2147483647" maxStringContentLength="2147483647" />
        </binding>
    </basicHttpBinding>
</bindings>
```

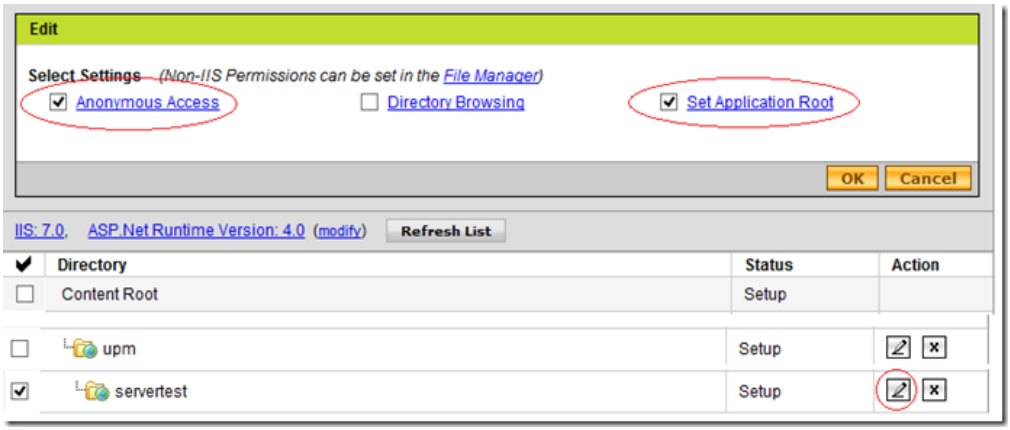
服务已经准备发布了。右击FileUploadService并选择发布。根据你的托管服务，你需要设置一下信息。 • 针对 GoDaddy： • 使用FTP方法。 • 设置目标地址，对我而言，我用ftp://ftp.ultimatepokermanager.com/upm/ServerTest • 设置证书



1.

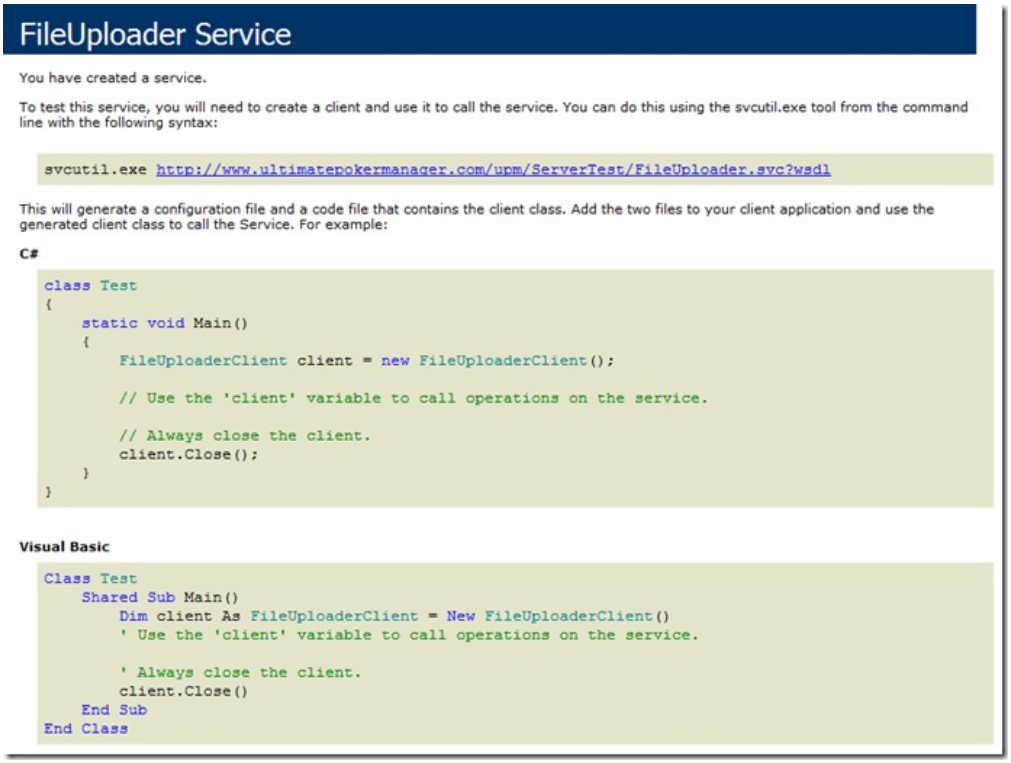
最巧妙地部分是在托管服务器上为WCF服务配置应用设置。在GoDaddy中，它位于工具部分的IIS 管理中。你需要检查匿名存取，并设置应用根目录。 upm\servertest 指向 http://www.ultimatepokermanager.com/upm/servertest#:

(c) Copyright Nokia 2011 All rights reserved



为检查你的服务是否开始运转，打开浏览器并输入服务地址，我的是http://www.ultimatepokermanager.com/upm/servertest/FileUploader.svc

如果你看到下面的图片，你就成功了！



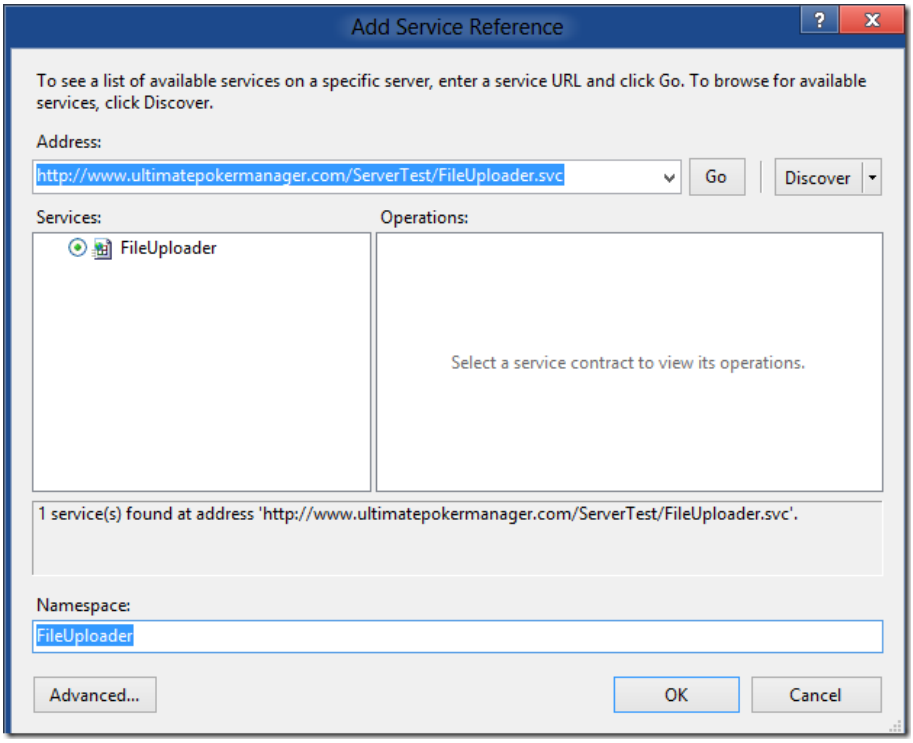
执行客户端代码以发送你的数据

本节展示了如何建立一个简单的应用，来发送一个字符串到服务器上的文件中。

(c) Copyright Nokia 2011 - All rights reserved

用常规方式建立一个Windows Phone应用

```
WCF服务客户端FileUploader 需要添加到工程中。右击Windows Phone工程并点击 Add Service Reference.
```

1. 

输入你WCF服务的地址，本例中是http://www.ultimatepokermanager.com/upm/servertest/FileUploader.svc 将命名空间重命名为 FileUploader。 点击OK。 用以下代码替换 MainPage.xaml 中代码：

```
<phone:PhoneApplicationPage x:Class="SendFileApp.MainPage"
                            xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                            xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                            xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
                            xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
                            xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                            xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
                            mc:Ignorable="d"
                            d:DesignWidth="480"
                            d:DesignHeight="800"
                            FontFamily="{StaticResource PhoneFontFamilyNormal}"
                            FontSize="{StaticResource PhoneFontSizeNormal}"
                            Foreground="{StaticResource PhoneForegroundBrush}">

    <Grid>

        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
```


(c) Copyright Nokia 2013 All rights reserved

```
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <StackPanel Margin="12,17,0,28">

            <TextBlock Text="SEND FILE"
                       Style="{StaticResource PhoneTextNormalStyle}" />

            <TextBlock Text="dashboard"
                       Margin="9,-7,0,0"
                       Style="{StaticResource PhoneTextTitle1Style}" />

        </StackPanel>

        <StackPanel Grid.Row="1"
                    Margin="12,0,12,0">

            <TextBlock Margin="12,0,0,0"
                       Text="Content" />

            <TextBox Text="abcdefg"
                     x:Name="textBoxContent" />

            <TextBlock Margin="12,0,0,0"
                       Text="File name" />

            <TextBox Text="Sample.txt"
                     x:Name="textBoxFileName" />

            <Button Content="Send File"
                    Click="ButtonSendFile" />

            <Button Content="Download File"
                    Click="ButtonDownloadFile"
                    IsEnabled="False"
                    x:Name="buttonDownloadFile"/>

        </StackPanel>

    </Grid>

</phone:PhoneApplicationPage>
```

该内容只用标准控件。 用以下代码替换 MainPage.xaml.cs中代码

```
using System;
using System.Net;
using System.Text;
using System.Windows;
using SendFileApp.FileUploader;

namespace SendFileApp
{
    public partial class MainPage
    {
        #region Constants

        // For this sample, the uploaded file can be access from my public folder
        private const string PublicServer = "http://www.ultimatepokermanager.com/upm/ServerTest/backup";

        #endregion

        #region Fields

        private string _directory;

        #endregion

        #region Constructor

        public MainPage()
        {
            InitializeComponent();
        }

        #endregion

        #region Event Handlers

        private void ButtonSendFile(object sender, RoutedEventArgs e)
        {
            byte[] content = Encoding.UTF8.GetBytes(textBoxContent.Text);
            _directory = Guid.NewGuid().ToString();

            FileUploaderClient fileUploaderClient = new FileUploaderClient();
            fileUploaderClient.UploadCompleted += UploadCompleted;

            fileUploaderClient.UploadAsync(new UploadFile { RelativeDirectory = string.Format("/{0}", _directory), FileName =
        }

        private void UploadCompleted(object sender, UploadCompletedEventArgs uploadCompletedEventArgs)
        {
            buttonDownloadFile.IsEnabled = true;
```


(c) Copyright Nokia 2011-2013. All rights reserved.

```
            if (uploadCompletedEventArgs.Result == null)
            {
                MessageBox.Show("The file has been successfully uploaded.", "Success!", MessageBoxButton.OK);
            }
            else
            {
                MessageBox.Show(string.Format("An error occured:\n\n{0}", uploadCompletedEventArgs.Result), "Error", MessageB
            }
        }

        private void ButtonDownloadFile(object sender, RoutedEventArgs e)
        {
            WebClient webClient = new WebClient();
            webClient.DownloadStringCompleted += WebClientDownloadStringCompleted;

            webClient.DownloadStringAsync(new Uri(string.Format("{0}/{1}/{2}", PublicServer, _directory, textBoxFileName.Text
        }

        private void WebClientDownloadStringCompleted(object sender, DownloadStringCompletedEventArgs downloadStringCompleted
        {
            if (downloadStringCompletedEventArgs.Error == null)
            {
                MessageBox.Show(string.Format("The content of your file is:\n\n{0}", downloadStringCompletedEventArgs.Result)
            }
            else
            {
                MessageBox.Show(string.Format("An error occured:\n{0}", downloadStringCompletedEventArgs.Error), "Error", Mes
            }
        }

        #endregion
    }
}
```

前面代码的有趣部分是：

```
private void ButtonSendFile(object sender, RoutedEventArgs e)
{
    byte[] content = Encoding.UTF8.GetBytes(textBoxContent.Text);
    _directory = Guid.NewGuid().ToString();

    FileUploaderClient fileUploaderClient = new FileUploaderClient();
    fileUploaderClient.UploadCompleted += UploadCompleted;

    fileUploaderClient.UploadAsync(new UploadFile { RelativeDirectory = string.Format("/{0}", _directory), FileName = textBoxFileNam
}
```

FileUploaderClient的用途相当简单： • 建立实例 FileUploaderClient。 • 监听 UploadCompleted事件。 • 异步上载文件。 请注意UploadFile 需要 RelativeDirectory 属性(以“/”开头)，FileName以及Content位于一个字节数组中。 一个上传文件地址实例： http://www.ultimatepokermanager.com/upm/ServerTest/backup/a5a8ae0b-50f0-4d16-90da-e83d6e1d30a3/Sample.txt。你会发现backupdirectory之前在Web.config文件中设置好了。 在本例中，我为目录名生成了一个GUID。上传的文件也是公共的，只要用户（或应用）知道目录名和文件名。如果你想加密，只需在服务上执行DownloadFile方法。若果你看了Upload方法的代码，应该不难。 我保证你会乐于看到你的web服务运行起来的！

下载

- File:SendFileApp.zip

(c) Copyright Nokia 2011 All rights reserved

(c) Copyright Nokia 2011-2013. All rights reserved.