Windows Phone Mango编程实践

***Windows Phone Mango Programming Practice***

第二篇 Silverlight交互篇

# 计划操作(Scheduled Actions)

## 计划操作（Scheduled Actions）

应用程序可以使用预定操作执行任务，即使在主应用程序未在前台运行。此功能的两个子类： 计划通知（包括提醒和警告）和计划任务（定期的资源密集型任务在后台执行）。Windows Phone Mango的计划通知功能允许应用程序创建有关警告和提醒的可配置的计划表；计划任务功能允许注册后台代理去执行计划的任务。

### 计划通知

计划通知在预定的时间在手机屏幕中弹出信息提醒和警告的对话框，类似于Windows Phone内置的通知。对话框的内容为自定义的文本信息，并允许用户取消和推迟通知。如果用户点击通知，与通知相关联的应用程序将启动。计划通知可以配置为启动一次或多次重复执行的计划，请注意计划通知的时间表精确到分。

计划通知有两种类型：警告（Alarm）和提醒（Reminder）。报警允许指定通知程序启动时播放的语音文件；提醒则声明与通知相关联的URI指向应用程序中的页面，包括查询字符串参数。用户点击提醒后，与此通知相关的应用程序将启动并显示声明的URI页面。

### 计划任务

计划任务允许应用程序执行后台代理程序，执行条件是主程序未激活。与计划通知不同的是，计划任务只能选择两种类型执行，即PeriodicTask和ResourceIntensiveTask。

PeriodicTask定期执行，但是执行时间短，且限制使用处理周期和内存等系统资源。此类型适合快速任务，比如检查启用位置功能的Web服务的用户数，或者缓冲小量的数据。

ResourceIntensiveTask不定期执行，在设备处于资源充沛的情况下执行，比如设备处于Wi-Fi网络连接状态并且设备采用外接电源供电。此类型的任务在允许使用充足的设备资源时，可以运行更长的时间处理大量的数据，即此类型的任务执行时间是弹性的。

GeoCoordinateWatcher是值得关注的计划任务，后台代理支持此API用于获得设备的地理位置坐标。需要注意的是此API获得地理位置坐标并不是当前的实际值，而是缓存中的位置坐标值，缓存中的位置坐标由设备每十五分钟更新一次。

## 动手实践——提醒（Reminders）

本节动手实践重点讲述使用Reminder和ScheduledActionService类创建Windows Phone的计划通知——提醒的方法。我们将创建提醒的显示列表、添加列表和提醒导航的页面，在提醒导航的页面中我们将传递的字符串数据作为查询操作的关键词，显示查询的结果。

### 创建提醒的显示列表

1. 本例的工程是在**Visual Studio**使用**Silverlight for Windows Phone**模板创建**Windows Phone Application**工程，命名为**ReminderSample**。
2. 在MainPage.xaml中创建用户界面，命名为ReminderListBox的ListBox以数据绑定的形式显示提醒列表。

Silverlight Project: ReminderSample File: MainPage.xaml

<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">

<TextBlock Text="you have no reminders registered" Name="EmptyTextBlock" Visibility="Collapsed"/>

<ListBox Name="ReminderListBox">

<ListBox.ItemTemplate>

<DataTemplate>

<Grid Background="Transparent" Margin="0,0,0,30">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="380"/>

<ColumnDefinition Width="50"/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<StackPanel Orientation="Vertical">

<TextBlock Text="{Binding Title}" TextWrapping="NoWrap" Foreground="{StaticResource PhoneAccentBrush}" FontWeight="Bold"/>

<TextBlock Text="{Binding Content}" TextWrapping="Wrap" Foreground="{StaticResource PhoneAccentBrush}"/>

<StackPanel Orientation="Horizontal">

<TextBlock Text="begin "/>

<TextBlock Text="{Binding BeginTime}" HorizontalAlignment="Right"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="expiration "/>

<TextBlock Text="{Binding ExpirationTime}" HorizontalAlignment="Right"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="recurrence "/>

<TextBlock Text="{Binding RecurrenceType}" HorizontalAlignment="Right"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="is scheduled? "/>

<TextBlock Text="{Binding IsScheduled}" HorizontalAlignment="Right"/>

</StackPanel>

</StackPanel>

</Grid>

<Grid Grid.Column="1">

<Button Tag="{Binding Name}" Click="deleteButton\_Click" Content="X" BorderBrush="Red" Background="Red" Foreground="{StaticResource PhoneBackgroundBrush}" VerticalAlignment="Top" BorderThickness="0" Width="50" Padding="0,0,0,0"></Button>

</Grid>

</Grid>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

</Grid>

1. 在 MainPage.xaml中创建应用程序栏ApplicationBar。在应用程序栏中增加导航到添加提醒的页面导航按钮。

Silverlight Project: ReminderSample File: MainPage.xaml

<phone:PhoneApplicationPage.ApplicationBar>

<shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">

<shell:ApplicationBarIconButton IconUri="/Images/add.png" Text="Add" Click="ApplicationBarAddButton\_Click"/>

</shell:ApplicationBar>

</phone:PhoneApplicationPage.ApplicationBar>

1. 在MainPage.xaml.cs中添加Microsoft.Phone.Scheduler 的引用。

Silverlight Project: ReminderSample File: MainPage.xaml.cs

using Microsoft.Phone.Scheduler;

1. 添加类型为 IEnumerable<Reminder>的变量reminders. <Reminder>声明Enumerable对象包含的是Reminde类的对象。

Silverlight Project: ReminderSample File: MainPage.xaml.cs

public partial class MainPage : PhoneApplicationPage

{

IEnumerable<Reminder> reminders;

……

}

1. 创建 **ResetItemsList**方法，此方法调用**[GetActions<T>()](http://msdn.microsoft.com/en-us/library/hh220821(v=VS.92).aspx)**检索与使用[**ScheduledActionService**](http://www.microsofttranslator.com/bv.aspx?from=&to=zh-CHS&a=http%3A%2F%2Fmsdn.microsoft.com%2Fen-us%2Flibrary%2Fmicrosoft.phone.scheduler.scheduledactionservice(v%3DVS.92).aspx)注册的应用程序的提醒对象。当列表中包含一项或多项提醒时，隐藏"no reminders"的TextBox，更新ReminderListBox的数据源，在页面上显示提醒列表。 **ResetItemsList**方法在OnNavigatedTo方法和删除提醒deleteButton\_Click的方法中被调用。

Silverlight Project: ReminderSample File: MainPage.xaml.cs

private void ResetItemsList()

{

// Use GetActions to retrieve all of the scheduled actions

// stored for this application. The type <Reminder> is specified

// to retrieve only Reminder objects.

reminders = ScheduledActionService.GetActions<Reminder>();

// If there are 1 or more reminders, hide the "no reminders"

// TextBlock. IF there are zero reminders, show the TextBlock.

if (reminders.Count<Reminder>() > 0)

{

EmptyTextBlock.Visibility = Visibility.Collapsed;

}

else

{

EmptyTextBlock.Visibility = Visibility.Visible;

}

// Update the ReminderListBox with the list of reminders.

// A full MVVM implementation can automate this step.

ReminderListBox.ItemsSource = reminders;

}

1. 重载**OnNavigatedTo(NavigationEventArgs)**方法。**OnNavigatedTo(NavigationEventArgs)**方法在导航到该页面时被调用，包括应用程序第一次启动。在此方法中调用ResetItemsList 方法更新ReminderListBox 的数据源。

Silverlight Project: ReminderSample File: MainPage.xaml.cs

protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)

{

base.OnNavigatedTo(e);

// Reset the ReminderListBox items when the page is navigated to.

ResetItemsList();

}

1. 实现在 XAML 中创建的删除按钮的事件处理程序。调用ScheduledActionService的Remove方法删除提醒。提醒被删除后，调用ResetItemsList方法更新列表。

Silverlight Project: ReminderSample File: MainPage.xaml.cs

private void deleteButton\_Click(object sender, RoutedEventArgs e)

{

// The scheduled action name is stored in the Tag property

// of the delete button for each reminder.

string name = (string)((Button)sender).Tag;

// Call Remove to unregister the scheduled action with the service.

ScheduledActionService.Remove(name);

// Reset the ReminderListBox items

ResetItemsList();

}

1. 应用程序栏按钮事件处理程序，实现导航至AddReminder.xaml页面添加提醒。

Silverlight Project: ReminderSample File: MainPage.xaml.cs

private void ApplicationBarAddButton\_Click(object sender, EventArgs e)

{

// Navigate to the AddReminder page when the add button is clicked.

NavigationService.Navigate(new Uri("/AddReminder.xaml", UriKind.RelativeOrAbsolute));

}

### 创建添加提醒的页面

创建添加新的提醒的页面，在此页面必须能够允许用户选择日期和时间，创建新的提醒。此示例使用DatePicker和TimePicker是Windows Phone Silverlight Toolkit 中包含的控件。示例代码中已经包含Windows Phone Silverlight Toolkit的引用。

系统中的每个提醒名称必须是唯一的，应用程序可以采用任何形式命名，本例中采用系统生成的唯一的Guid命名提醒。

计划通知——提醒可以设置重复执行，在本例中判断每年、月末、每月、每周、每日的提醒重复执行的选项，如果设置了就设定重复执行的周期，否则设置重复执行的属性为空。

创建提醒对应的导航连接并传递参数，调用ScheduledActionService的Add方法创建新的提醒Reminder。

Silverlight Project: ReminderSample File: AddReminder.xaml.cs

private void ApplicationBarSaveButton\_Click(object sender, EventArgs e)

{

// Generate a unique name for the new reminder. You can choose a

// name that is meaningful for your app, or just use a GUID.

String name = System.Guid.NewGuid().ToString();

// Get the begin time for the reminder by combining the DatePicker

// value and the TimePicker value.

DateTime date = (DateTime)beginDatePicker.Value;

DateTime time = (DateTime)beginTimePicker.Value;

DateTime beginTime = date + time.TimeOfDay;

// Make sure that the begin time has not already passed.

if (beginTime < DateTime.Now)

{

MessageBox.Show("the begin date must be in the future.");

return;

}

// Get the expiration time for the reminder

date = (DateTime)expirationDatePicker.Value;

time = (DateTime)expirationTimePicker.Value;

DateTime expirationTime = date + time.TimeOfDay;

// Make sure that the expiration time is after the begin time.

if (expirationTime < beginTime)

{

MessageBox.Show("expiration time must be after the begin time.");

return;

}

// Determine which recurrence radio button is checked.

RecurrenceInterval recurrence = RecurrenceInterval.None;

if (dailyRadioButton.IsChecked == true)

{

recurrence = RecurrenceInterval.Daily;

}

else if (weeklyRadioButton.IsChecked == true)

{

recurrence = RecurrenceInterval.Weekly;

}

else if (monthlyRadioButton.IsChecked == true)

{

recurrence = RecurrenceInterval.Monthly;

}

else if (endOfMonthRadioButton.IsChecked == true)

{

recurrence = RecurrenceInterval.EndOfMonth;

}

else if (yearlyRadioButton.IsChecked == true)

{

recurrence = RecurrenceInterval.Yearly;

}

// Create a Uri for the page that will be launched if the user

// taps on the reminder. Use query string parameters to pass

// content to the page that is launched.

string param1Value = param1TextBox.Text;

string param2Value = param2TextBox.Text;

string queryString = "";

if (param1Value != "" && param2Value != "")

{

queryString = "?param1=" + param1Value + "&param2=" + param2Value;

}

else if(param1Value != "" || param2Value != "")

{

queryString = (param1Value!=null) ? "?param1="+param1Value : "?param2="+param2Value;

}

Uri navigationUri = new Uri("/ShowParams.xaml" + queryString, UriKind.Relative);

Reminder reminder = new Reminder(name);

reminder.Title = titleTextBox.Text;

reminder.Content = contentTextBox.Text;

reminder.BeginTime = beginTime;

reminder.ExpirationTime = expirationTime;

reminder.RecurrenceType = recurrence;

reminder.NavigationUri = navigationUri;

// Register the reminder with the system.

ScheduledActionService.Add(reminder);

// Navigate back to the main reminder list page.

NavigationService.GoBack();

}

### 创建提醒启动页面

重写ShowParams.xaml.cs的基类PhoneApplicationPage的OnNavigatedTo(NavigationEventArgs)方法。计划通知——提醒导致此页面时将调用此方法。在此方法中，使用NavigationContext类的查询字符串值获得传递的参数值。

Silverlight Project: ReminderSample File: ShowParams.xaml.cs

// Implement the OnNavigatedTo method and use NavigationContext.QueryString

// to get the parameter values passed by the reminder.

protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)

{

base.OnNavigatedTo(e);

string param1Value = "";

string param2Value = "";

NavigationContext.QueryString.TryGetValue("param1", out param1Value);

NavigationContext.QueryString.TryGetValue("param2", out param2Value);

param1TextBlock.Text = param1Value;

param2TextBlock.Text = param2Value;

}

### 调试应用程序

按F5运行应用程序，或者点击Start Debugging按钮运行，如图10-1 Start Debugging。



图10-1 Start Debugging

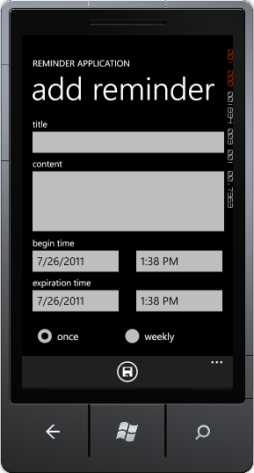
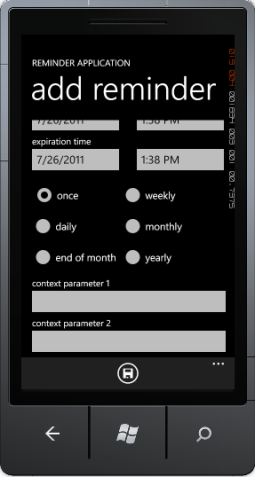
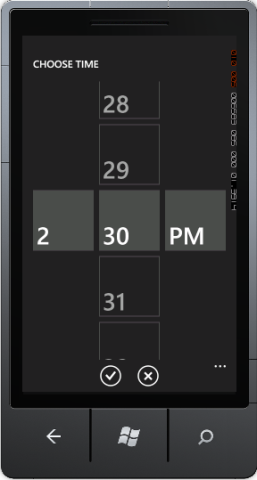
   

图10-2 添加提醒

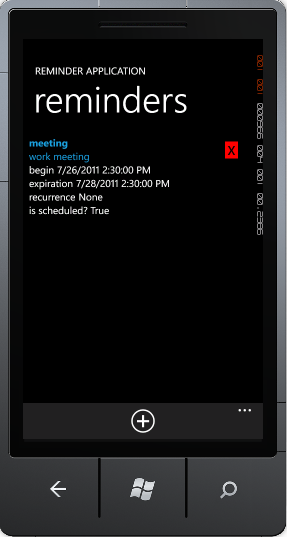
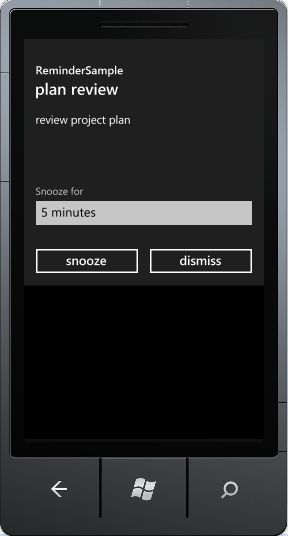
 

图10-3 动手实践——提醒的运行结果