



Windows Phone 7

UI 设计及人机交互指南

原文：[MSDN](#)

译者：[林永坚 \(Jake Lin \)](#)

[姜泳涛 \(Yongtao Jiang \)](#)

[冯峰 \(AppleSeeker \)](#)

[施炯 \(Dearsj001 \)](#)

[李卫勇](#)

[张欣\(Breakstring\)](#)

Windows Phone 7

UI设计及人机交互指南

本文档对应软件产品的预发布版本，在最终的商业版发布时，可能会发生很大的改变。本文档仅供参考，微软公司对文档中提供的信息资料不提供任何明示或暗示的保证。本文档中的 URL 和其它网站的引用，如有更改，不会另行通知。使用本文档或使用本文档导致的结果的全部风险，完全由用户承担。除非另行说明，否则本文档中所涉及的公司，组织，产品，域名，电子邮箱地址，标志，人物，地点以及所述事件均为虚构。与任何确实存在的公司，组织，产品，域名，电子邮箱地址，标志，人物，地点以及事件无任何关系，是臆想和推测出的。用户须遵守任何适用的版权法。如果版权中无限制，没有微软公司的书面许可，任何人不得复制或保存本文档中的任何内容，不可将其引入检索系统，也不可以任何形式或方式（如电子，机械，影印，录制或其他方式）发送，或用于其他目的。

微软公司具有专利权，专利申请，商标，版权或其他知识产权，包括本文档中的主题。除非获得微软公司出具的书面许可协议，本文档中所陈述的内容不会给你任何专利，商标，版权或其他知识产权的许可。

2010 年 微软公司 版权所有，违者必究。

Microsoft, Internet Explorer, Outlook, Silverlight, Windows 和 Windows Phone 为微软集团所属商标。

其它各产品的商标均为其所属公司的财产。

目录

目录

Windows Phone 7 UI 设计及人机交互指南	5
输入类型.....	5
为触摸而设计	5
支持的触摸手势.....	9
硬件按钮.....	13
键盘.....	15
传感器.....	17
用户界面框架.....	18
屏幕方向.....	18
框架和页面的构建.....	20
滚动查看器.....	21
主题.....	22
Windows Phone 应用程序控件	23
Push Button.....	25
Check Box.....	26
Radio Button	28
Hyperlink Control	29
Slider Control	30
Text Block.....	31
Text Box.....	31
List Box	32
List View Item.....	32
Progress Bar	34
Page Title.....	34
Panorama Application.....	35
Pivot Control	41
Windows Phone 系统控件	42
系统托盘和应用工具栏.....	42
上下文菜单.....	43
Windows Phone 消息通知	44
进度显示器.....	46

UI 文字注意事项	48
Windows Phone 7 语音语调	48
大小写标准化	48
标点符号	49
UI 设计注意事项	50



Windows Phone 7 UI设计及人机交互指南

输入类型

基于 Windows Phone 7 CTP 版的应用程序具有多种交互方式，包括：触摸手势、屏幕虚拟键盘、Windows Phone 的硬件按钮，以及传感器例如重力加速器。本小节将指导开发者如何在自己的应用中使用这些输入类型并讲述在开发的过程中需要考虑的相关事宜。

以下小节提供不同的输入类型的详细情况：

[为触摸而设计](#)

Windows Phone 7 CTP 的用户界面是为了触摸交互而设计的，所有的导航通过一系列手指手势的移动组合而完成。

[支持的触摸手势](#)

提供触摸手势的示例和设计上的一些考虑事宜。

[硬件按钮](#)

Windows Phone 7 CTP 版本中硬件按钮的布局。

[键盘](#)

屏幕上的虚拟键盘用于在触摸屏上输入文字。

[传感器](#)

Windows Phone 7 CTP 版中的传感器包括重力加速器。

为触摸而设计

Windows Phone 7 CTP版的用户界面是为了触摸交互而设计的，所有的导航通过一系列手指手势的移动组合而完成。为了达到你应用程序中的主要目标，提高可用性和提供直观的设计，需要对针对触摸屏的控件进行精心布置和调整尺寸。 以下各节将会为您提供一些关于如何在 Windows Phone应用程序中创建这样的体验的一些指导。 虽然本节提供关于自定义 UI 视觉元素的指导，但是我们推荐您在应用程序中使用的标准Metro主题控件。 这些控件已经为支持触摸交互并基于本指导手册做了优化调整。

备注:

也有一些例外情况，例如在游戏的开发中视觉元素的大小各尽不同。

可用的和可访问的布局

每一个可以被触摸的 UI 元素应该可以被手指方便的点击。这涉及操纵它的大小、间距、位置以及视觉效果来减少通过手指来获取目标元素的困难度。

最小点击区域大小

下面的小节中，如果你使用像素来度量，请相应的转换毫米单位为对应的分辨率和屏幕大小下的像素单位。

- ✓ 推荐的点击区域的大小（高和宽）应该最少9毫米。
- ✓ 点击区域的可视尺寸的大小（高和宽）应该是自身大小的60%~100%
- ✓ 点击区域是不可见的，所以它可以比可视元素大。但是，点击区域绝对不能比可视元素小。



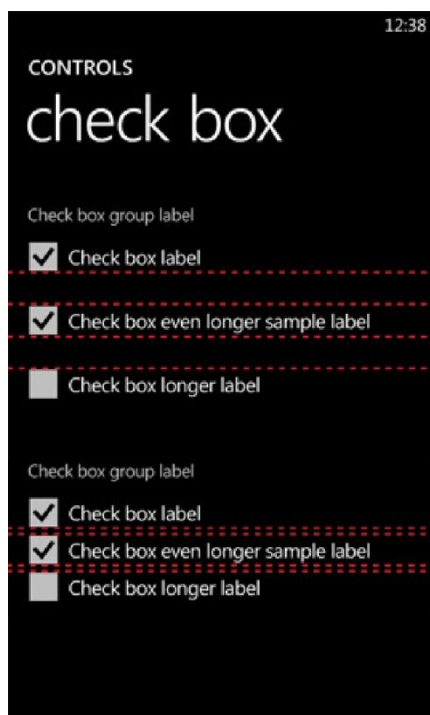
考虑较大的点击区域

以下情况需要每一个点击区域大于9毫米：

- ✓ 该UI元素经常被点击（触摸）
- ✓ 如果用户点击在错误的目标上的话后果会很严重。例如，发送出去一个没有完成的短信，或者删除一封邮件。
- ✓ 错误的点击带来的结果会让人感到沮丧。例如，导致一个对话框或者其他应用程序运行，或者用户被导航到一个他不期望的界面。
- ✓ UI元素挨着屏幕的边沿用户不便于点击。
- ✓ 该UI元素作为一系列任务的一部分，或者一个任务需要你多次点击一系列屏幕上相邻的元素。例如使用拨号面板。

备注:

屏幕上的虚拟键盘和Windows Phone® Internet Explorer® 中的超链接是个例外，因为他们在点击的时候会改变大小。



correct

At minimum, each element must be separated by a distance greater than 2mm.

incorrect

Improperly spaced UI elements will lead to touch errors.

考虑较小的点击区域

屏幕上UI元素最小的高度和宽度是7毫米。

两个UI元素或者点击目标之间的间隙必须大于2毫米且小于7毫米，例如文本或者单选框

大小的约束

对于具有特殊大小约束的 UI 元素，考虑下列一个或多个情况：

- ✓ 增大 UI 元素之间的空隙，使得适应点击区域的大小（例如 9 毫米的点击区域）。
- ✓ 如果高度被限制，可以使用长方形的控件，使得他们能够容易被点击到。如果点击区域的宽度大于 20 毫米，则其高度可以小于 7 毫米。

自然和直觉性的交互

“Metro”的设计原则要求 UI 元素真正的被数字化。但是您应当使用自然而恰当的数字化模拟，在没必要的时候，不需要完全模仿真实的交互。开发人员应该具备一种观念，那就是 UI 应该看上去和感觉上像真实世界中的那样好才行。并且，如果需要，试着来模拟真实世界的行为。

内容就是接口

- ✓ 使用应用程序的内容，例如一张照片或者一个Web页面来作为UI，并且让用户点击它进行交互。
- ✓ 再一次强调，你应该使用Windows Phone 7 CTP版中的通用Silverlight控件来保持一致性。

自然的响应

- ✓ 使用手势来模拟真实生活中的行为，例如在相片上平移手指来移动它。
- ✓ 不要使用手势作为任务的快捷方式，仅仅在需要手势的时候使用它。更多关于触摸手势的功能，参见“支持的触摸手势”一节。

简单和一致

- ✓ 在Windows Phone 7 CTP版本中，单点触摸的手势使得单手交互变得更加容易。 在需要多变并且具有复杂功能的手势的话，这里也有多点触摸。关于更多Windows Phone 7 CTP版中的手势支持，请参见“支持的触摸手势”一节。
- ✓ 所有基本或者通用操作应该可以使用一个指头完成。
- ✓ 在手机中对于手势的响应应该保持一致性。使用基于Metro主题的控件有助于保持这种一致性。

感知和响应的性能

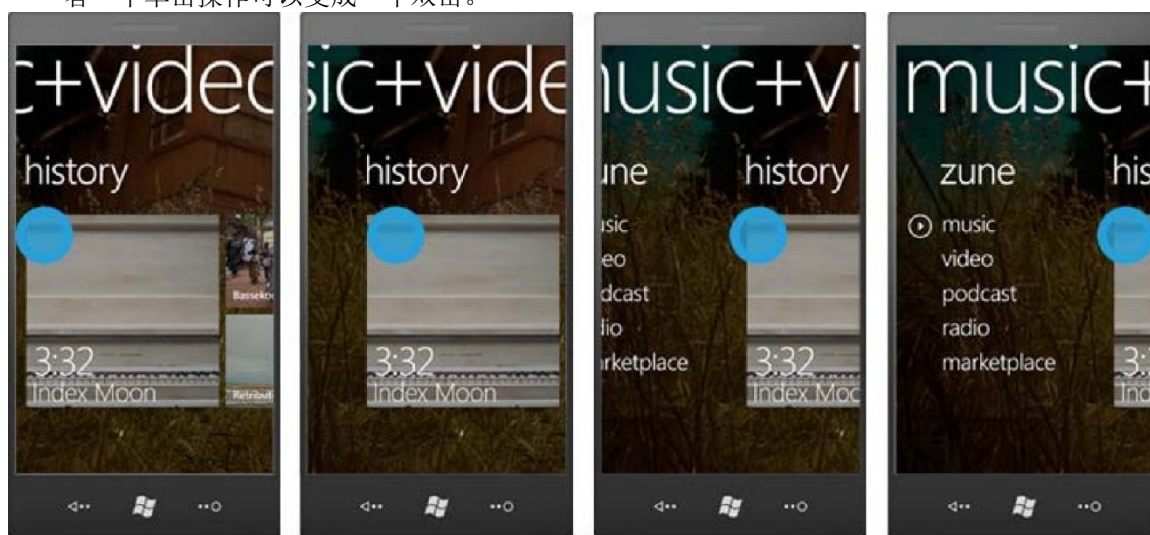
真实世界中的对象会立即响应触摸。我们强烈建议你的Windows Phone 7 CTP版应用程序的UI基于这一法则。如果UI在切换的时候有延迟或者看起来比较缓慢将会对用户体验有负面影响。

触摸的反馈

- ✓ 在UI元素上提供视觉反馈来指示操作交互。例如， Windows Phone 7 CTP版中在触摸的时候使用“tilt”动画。 而类似的照片和地图中则不使用这种效果。 同样，你也可以使用一些移动来标识元素上的交互。
- ✓ 听觉上的反馈是另外一种标识元素的交互的手段。

良好的UI响应

- ✓ UI 元素应该在触摸的时候快速响应。用户不应该被这种等待而打断他的思路、正在做的事情。所有的操作必须有一个立即且明确的结果。
- ✓ UI元素应该在手势发生的时候就响应，而不是手势之后。例如，用户快速的划动一张照片，这个移动的动作是在手势结束前就产生的。
- ✓ 手势可以转换，或者连接，及从一个手势到另一个手势。例如，一个平移手势可以转换为一个划动手势，或者一个单击操作可以变成一个双击。



gesture swipe right

Your application should respond immediately to a gesture touch. For example, there should be a smooth transition from one page with no lag time. Also, the transition should occur during the movement, not at the end.

需要消耗时间的操作

- ✓ 如果用户面临一个需要消耗时间的操作，你应该通过提供一个简单的进度条来告诉用户某些操作正在进行。有时候也需要使用内容来标识进度，例如下载的时候，在进度条上显示越来越多的内容。进度条控件用来显示在应用程序中和活动相关的一系列事件。Windows Phone 7 CTP版中所提供的进度条控件也可以被开发

人员自行定制。在需要的时候就使用进度条，而把消息通知作为最后的手段。更多信息，参见“进度条”一节。

不寻常和激动人心的体验

当使用一个 Windows Phone 应用程序时，用户应该在和手机交互的时候感到一些好奇。在你的设计中通过导航栏、浏览内容、以及激动人心的视觉效果来吸引你的用户。

探索

- ✓ 通过使用触摸手势来鼓励用户探索内容。用户应该在完成任务的时候也同样享受这个过程。

使得用户感觉满意

- ✓ 无论是单个的手势还是一系列手势的组合，他们都应该为你的程序提供一些让人爱不释手的体验。触摸使得用户看到的和交互的实际表现一致从而增强了用户感官。单点触摸，多点触摸以及多手势的组合讲给用户带来多彩的更加直观的用户体验。

支持的触摸手势

所谓触摸手势是单指或者多个指头在触摸屏上进行一些移动。点击一些 UI 元素，例如一个按钮，就是一个例子。触摸手势是用户和 Windows Phone 进行交互的主要方式。下面提供了详细的关于 Windows Phone 7 CTP 版中所支持的触摸手势的使用方式以及行为的参考。应当使用系统所提供的 Metro 主题的控件来作为触摸交互元素。再一次提醒，这些控件已经为触摸交互调整好大小。下面是在 Windows Phone 7 CTP 版中支持的单点触摸和多点触摸：

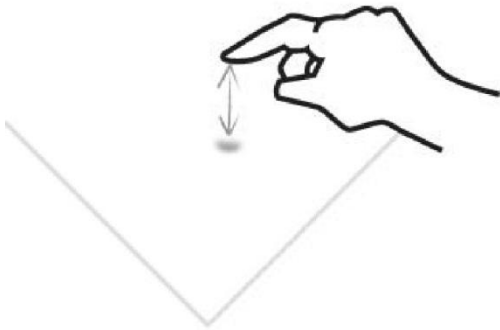
单点触摸：

- ✓ 单击
- ✓ 双击
- ✓ 平移
- ✓ 划动
- ✓ 长按

多点触摸：

- ✓ 缩放

单击



Single touch on the screen.

Finger down on a single point within a bounded area and back up within a short period of time.

单击手势由两个行为组合而成：

- ✓ 手指按下指示触摸事件开始
- ✓ 手指抬起执行该动作

在屏幕上移动则单击手势停止。

双击



Two quick taps within a bounded area.

双击操作则用来在应用程序或者控件的放大、缩小状态中改变。应用程序将会根据当前的缩放状态来自动的放大或者缩小。而放大或者缩小状态由应用程序自己所定义。

平移

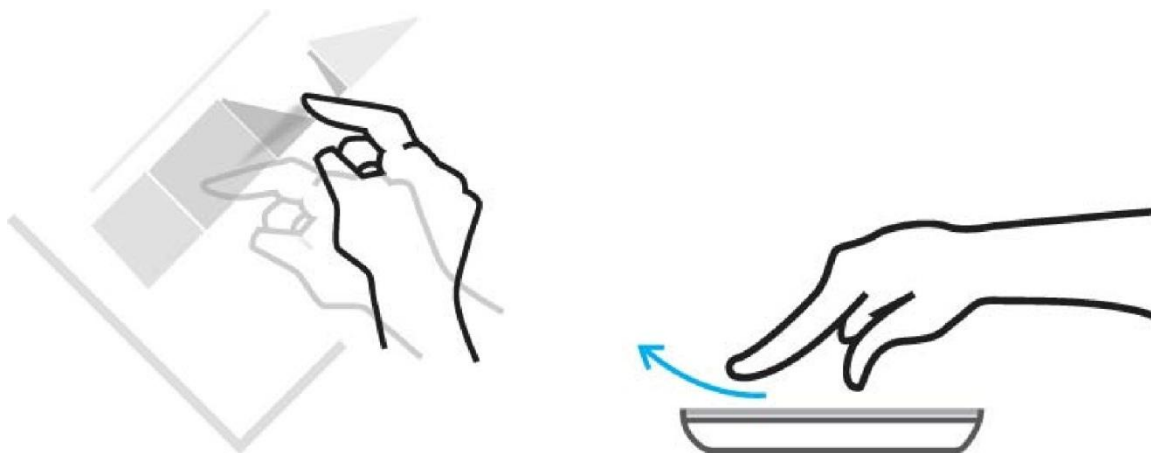


Finger down followed by finger move in a single or multiple directions. Pan ends on finger up (or when another gesture starts).

平移操作也是由两个动作组合而成：

- ✓ 内容跟随手指移动的方向而移动。控件或者应用程序可以决定支持什么方向上的平移。它可以是水平的、垂直的或者任意的角度。而且，如果内容移动到某两个状态之间时，该内容应该自行调整到最接近的一个状态。
- ✓ 平移可以移动或者改变一个特殊的元素。元素跟随手指移动，并在手指抬起后停留在新的位置。

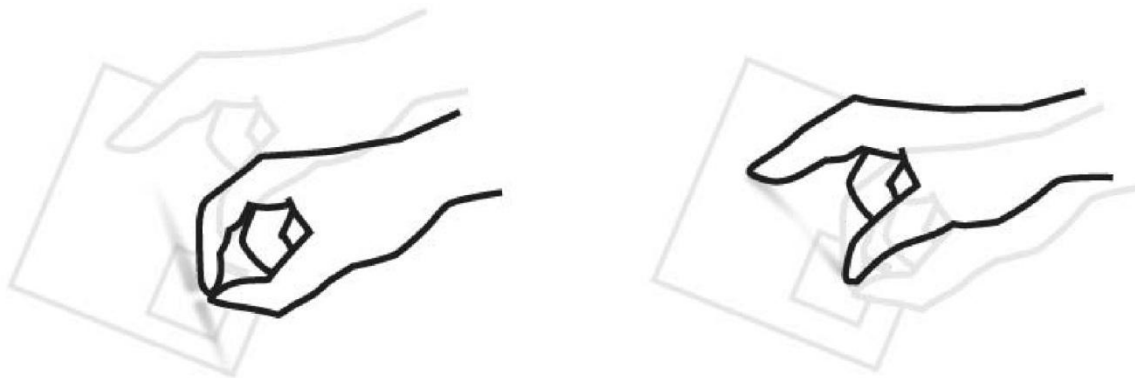
划动



Finger down followed by a quick finger move in a single direction and finger up. Flick can also follow a Pan gesture.

划动手势将内容从一个区域移动到另外一个区域。控件或者应用程序来决定支持什么样的划动方向。它可以是水平、垂直或者其他特殊的方向。如果选择垂直或者水平方向，其他方向上的移动将会转化成水平或者垂直方向上的移动。

缩放



Two fingers down followed by moving the finger closer/further apart from each other.

缩放手势在内容上提供持续的放大/缩小功能，缩放的中心点在两个手指之间。

长按



Finger down on a single point within a bounded area for a defined period of time

长按手势将会显示当前的上下文菜单或者是选项页。

触摸设计的考虑

以下内容是你设计带触摸功能的应用程序的时候需要考虑的：

- ✓ 所有的Windows Phone 7 CTP版本中的Silverlight控件支持本小节中所讨论的触摸手势。
- ✓ 必须保持一致的触摸体验。如果你建立了自定义的控件或者UI，他们也应该以同样的方式来响应手势。
- ✓ Windows Phone 7 CTP 版中的手势和 Windows 桌面系统的手势保持一致。但是也因为屏幕大小的差异和 Windows 桌面操作系统支持鼠标的缘故，有的地方会略有差异。这些差异主要位于编辑的快捷方式中，它们可能会被屏幕虚拟键盘上的功能所代替。Windows Phone 7 CTP 版本中的应用程序的手势方面的用户体验应该和 Windows 桌面系统保持一致性。
- ✓ 在Windows Phone 7 CTP版本中不能扩展手势。你只能使用本小节中所提及的手势和移动动作。

硬件按键

Windows Phone 上设置有几个硬件按键。每个按键提供一个特定功能，即可调节或影响运行的程序。

- ① 源/休眠
- ② 音量
- ③ 屏幕
- ④ 照相
- ⑤ 返回
- ⑥ 开始
- ⑦ 搜索



备注:

在应用程序中，不可扩展或重载硬件按钮的功能。

开始按钮

当用户按下该按钮，将在手机的任何页面跳转到开始菜单。应用程序运行过程中也是如此。

搜索按钮

搜索按钮能够为用户提供搜索体验，帮助用户在设备上检索所需内容。

返回按钮

返回按钮用于在一个程序或两个程序中返回（后退）。按下此按钮，应用程序则允许框架执行操作。同时，返回按钮也可以关闭屏幕键盘、菜单、对话框、导航到前一页、退出搜索操作或是切换应用程序。然而，返回按钮的主要用途是从当前页面切换到前一页面。

备注:

返回按钮与回退键不同，无法实现删除文本输入的功能。

照相按钮

从功能性来说，Windows Phone 7 CTP 照相按钮支持全按下和半按下模式。当用户全按下时，手机将会启动照相应用程序。如果用户半按下，自动调焦功能被启动。从编码角度，应用程序能够根据调用照相控制 API 启动照相功能。同时，应用程序能够决定何时何地调用该 API。

备注:

此外，应用程序无法重载该硬件照相按钮行为。

音量按钮

音量按钮的主要功能是调节音量，包括电话，音乐，收音机或正在播放的电影的音量。如果音频流是非激活状态，音量按钮则可以改变激活的声音设置。音量控件能够显示对前一个或后一个音频传输的控制。

音量按键

当有来电时，音量按键能够消除铃声。

如果正在播放有效的音频流，音量按键能够改变有效音频流的音量，但不包括系统事件声音和效果。

当设备锁定时，音量按键应仍有效且能响应操作。开发人员不能重定义此功能。

如果用户长按音量按键，将会实现反复按下该按键的动作。按住音量增强键将会不断增大音量。

电源按键

如果手机正在使用中，短按电源按键就能关闭手机的显示内容，同时也能锁定手机。如果手机处于闲置状态，短按电源按键则可以恢复手机的使用状态。

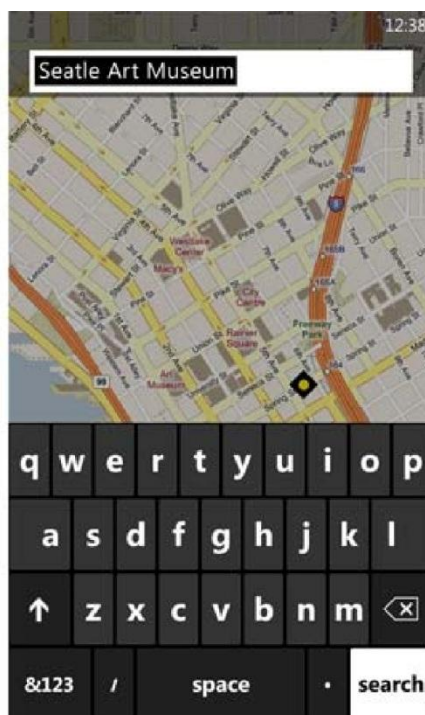
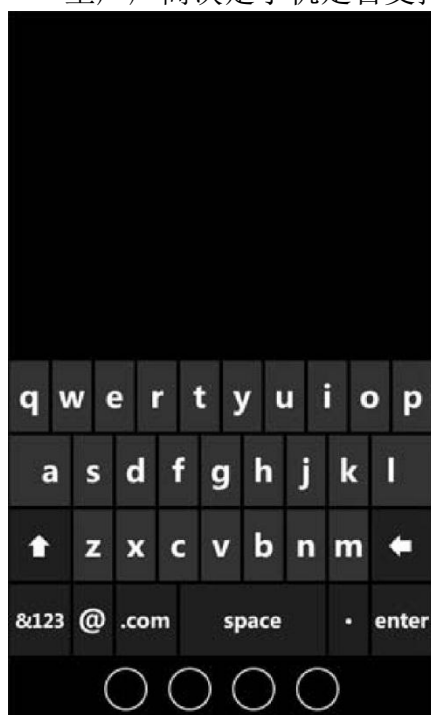
键盘

屏幕键盘

屏幕或硬件键盘，用来在 Windows Phone 上输入文本。如果编辑控件被激活，屏幕键盘将自动展开。当用户碰触编辑控件以外的地方，屏幕键盘则会关闭。如果手机本身具备硬件键盘，当使用硬件键盘时，屏幕键盘就会自动关闭。Windows Phone 7 CTP 手机具有一些输入辅助功能，如文本提示和自动修正。屏幕键盘和硬件键盘都支持该功能。

备注：

生产厂商决定手机是否支持硬件键盘。



有关屏幕键盘设计的考虑因素

当键盘在一个应用程序中展开时，将覆盖住屏幕的底部区域。

<http://www.WPMind.com>

Windows Phone 资讯观点、应用软件、开发技术交流专业站！

如果文本提示可用，其位于键盘上方，同时也能为应用程序节省大量的空间。

当键盘展开时，应用程序应能够上下滚动，以确保激活的编辑控件和插入符号可见。如果使用多行编辑控件，其中一部分可能会隐藏在屏幕键盘之后。你必须确保插入光标所在行总是可见且在屏幕键盘之上。

在一些应用程序中，用户很难碰触到文字框或编辑控件之外的控件，以关闭屏幕键盘，看到更多的内容。例如，在应用程序占用全部或大部分屏幕区域的情况下，当用户希望浏览内容而不是输入时，应用程序能够选择自动关闭屏幕键盘。例如，用户滚动应用程序区域或激活的编辑控件，以获得额外的可视区域。另一种做法就是使用一个只读视图和一个编辑视图，用户可以切换到编辑视图，应用程序展开屏幕键盘。当用户切换回只读视图时，例如按下返回按键，程序就能选择关闭屏幕键盘。

根据文字框或编辑域内容，键盘可以支持多键布局。

开发者应该在编辑区域中设定输入范围，便于选择合适的布局和启用合适的输入助手。例如，如果开发者选择 URL 输入范围，设计的键盘就应具备.com 键。想要实现这一功能，需要在开发的项目中为文字框或编辑控件设定输入范围的属性。

下列是键盘布局的一些具体内容：

键盘类型	布局
默认	标准QWERTY键盘布局
文本	带表情符号的标准ASCII键盘布局
电子邮箱地址	带.com和@键的标准键盘布局
电话号码	经典的12-键布局
网址	带.com键和自定义回车键的标准键盘布局
地图	带自定义回车键的标准键盘布局
搜索	带搜索和.com键的半透明键盘布局
SMS 地址	便于访问电话号码簿的标准键盘布局

硬件键盘

手机生产厂商决定了 Windows Phone 是否具有硬件键盘。Windows Phone 7 CTP 具有几种可用的硬件键盘，如拉出式键盘，滑出式键盘，弹出式键盘或旋转式键盘。

备注：

Windows Phone 7 CTP 只支持全字母布局，像 QWERTY, AZERTY 和 QWERTZ 键盘。但不支持 12/-20 键布局。

- ✓ 硬件键盘的可以输入字母，重读字母，数字和符号。
- ✓ 上档键可以键入大写字母。硬件键盘具有 3 个上档模式，包括 On、Off 和 Lock（大写锁定键）。
- ✓ 表情符号键带有表情符号选择器。
- ✓ 重读键用于键入重读字母。当用户按下重读键后，插入符号左边的字母就会增加一个重读。重复按下重读键可实现想要的重读。功能键（FN）加重读键可返回到上个重读。如果用户长按重读键，则会弹出重读选择器。
- ✓ 按下符号键（SYM）可启动符号选择器，从而获得键盘中不存在的字符。
- ✓ 如果用户长按 SYM 键，弹出语言选择器。FN 和 SYM 可以切换至下一种语言。

- ✓ 系统托盘输入指示器能够显示上档模式，FN 模式和当前的语言。
- ✓ 键盘按键能够被重载。用户可以长按某键或使用 FN 键访问第二功能。

有关硬件键盘设计的考虑因素

- ✓ 硬件键盘只能用来输入，不能控制 UI。
- ✓ 硬件键盘包括方向键，方向键的功能是在编辑控件内移动插入符号。但是，方向键不可用于移动焦点，滚动列表，导航地图或网页。
- ✓ 硬件键盘上可始终使用的按键如下：
 - ◆ 字母（A-Z），回车键，空格键，后退键，上档键，表情符号键，SYM，期间键和逗号键。
 - ◆ 数字（0-9）可以是主要字符或次要字符。
 - ◆ 德语、法语、意大利语和西班牙语键盘的重读键。
- ✓ 硬件键盘不支持下列按键：
 - ◆ 方向盘或任何其他导航指定硬件的按键。
 - ◆ “确认&起始”和“发送&结束”硬件的软键。
 - ◆ 删除键，插入键，控制键（CTRL），换档键，大写锁定键，上页和下页键以及退出键（ESC）。
 - ◆ 键盘中的开始键，搜索键和返回键。
- ✓ 如果启动了符号选择器，重读选择器或语言选择器，它们位于手机屏幕的底部。这些选择器会在选择完毕后自动消失，或根据预定义时间隐藏。
- ✓ 如果硬件键盘可用或被展开，则可以通过使用 API 了解应用程序。
- ✓ 应用程序能够使用 API 知道硬件键盘的可用状态，展开状态，以此定义其行为。
- ✓ 当展开硬件键盘，屏幕键盘就会关闭。如果设备具有固定的硬件键盘，则永远不会启用屏幕键盘。
- ✓ 硬件键盘可以使用输入辅助功能。

传感器

重力加速器

Windows Phone 7 CTP 的重力加速器是一个机电装置，可以测量因重力或外部电源引起的加速度。

一个 3D 运动传感器能不断提供作用于 X、Y 和 Z 平面中设备所产生的力的信息。Windows Phone 可以利用此功能为终端用户提供更为先进的用户体验。这些设计的目标是为了提供托管 API，托管 API 便于使用，也能灵活地提供自动屏幕旋转，倾斜滚动以及游戏等功能。

用户界面框架

Windows Phone 7 CTP 版用户界面的设计目标是为最终用户提供一个最佳的视觉体验。本节将教育开发者关于全屏幕模式(full screen mode)，屏幕方向(screen orientation)，框架和页面导航(frame and page navigation)以及应用程序滚动(application scrolling)等主题。

下面的主题将要在本节进行讨论：

屏幕方向 (Screen Orientation)

支持两种类型的屏幕方向，竖屏 (portrait)和横屏(landscape)。

框架和网页构造 (Frame and Page Construction)

Windows Phone 7 CTP 版的应用程序是基于 Silverlight 的网页浏览模式，在这模式下，用户可以在不同的内容页面下相互导航。

滚动查看器 (Scroll Viewer)

当应用程序的内容大于屏幕大小的时候就会发生滚动。

主题 (Themes)

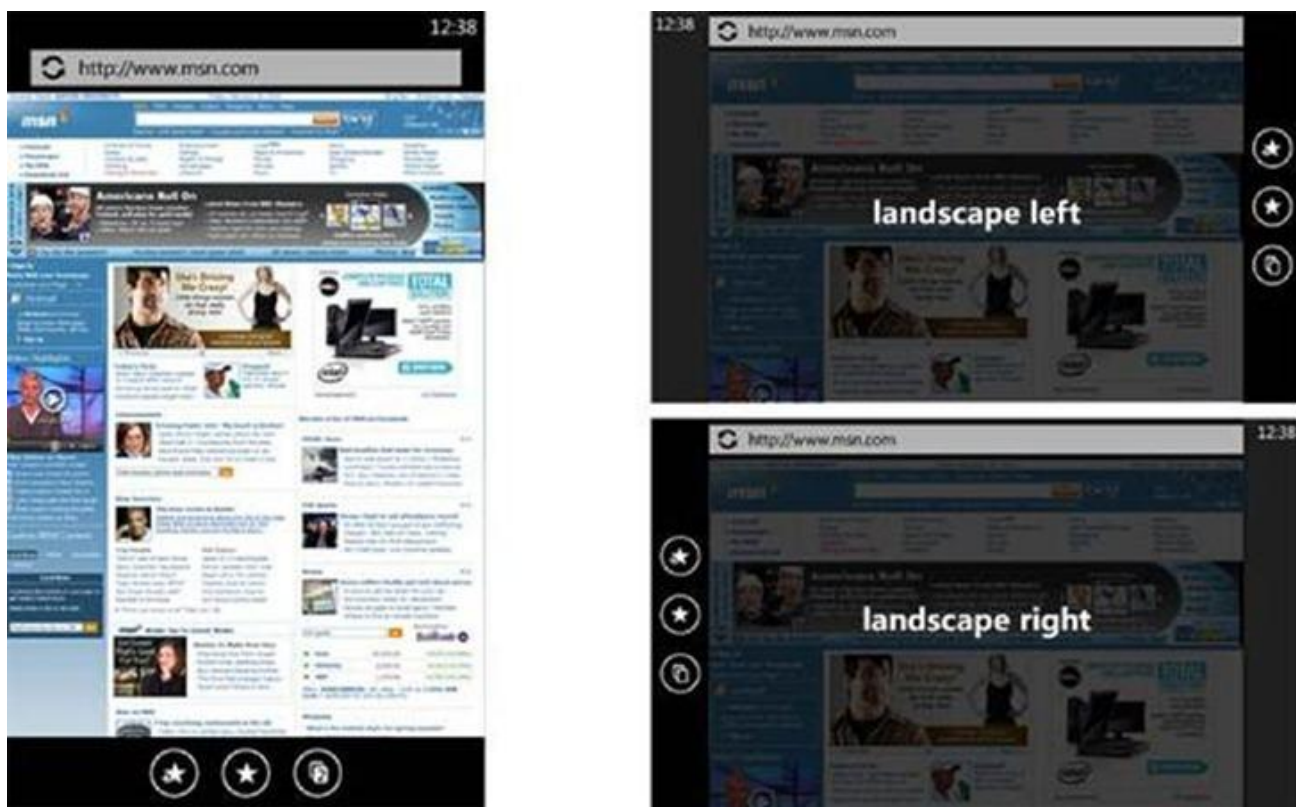
主题是用于个性化 Windows Phone 视觉元素的资源。

屏幕方向

Windows Phone 支持三种类型的屏幕方向：竖屏，向左的横屏和向右的横屏。向左横屏和向右横屏之间的区别是应用程序工具栏和系统托盘的方位不一样。

备注：

应用程序不能指定只是支持向左横屏或者向右横屏的应用程序工具栏和系统托盘的方位，需要两种都同时支持。



竖屏模式

在竖屏模式下，页面在垂直方向的时候高度要比宽度大。当设备旋转或滑出物理键盘的时候，方向会发生改变。当屏幕方向更改的时候，具有方向感知功能的应用程序也应当去响应 该方向改变。

横屏模式

在横屏模式下，页面显示的宽度大于高度。下面是一些横屏模式用途：

- 当用户使用一个横向拉出键盘进行输入时，例如编写 email 和短信的时候。
- 如果某个网页更适合横向浏览的时 候，横屏模式能为用户提供了更好体验。
- 拍照的时候。
- 查看单张照片或观看幻灯片的时候。
- 欣赏电影或 视频的时候。
- 玩游戏的时候。

程序设计时的考虑

当您设计应用程序横屏和竖屏模式的时候，需要考虑以下几方面的因素。

- 应用程序必须同时支持所有的屏幕方向。如果一个应用程序能支持横屏模式和竖屏模式，当用户翻转手机的时候，或者滑出物理键盘的时候，屏幕显示模式就会发生转换。由于 orientation 属性是只读的，通过编程的方式不能直接转换屏幕的显示模式。但是也有个间接的方法可以实现这一功能，通过对 Supported Orientations 属性的赋值可以实现。

- 当更改为横屏模式的时候，应用程序工具栏和系统托盘，应用程序菜单，用户音量界面组件，消息通知模块和系统对话框（如低电池的通知）等等将会变成横屏感知。
- 在横屏模式或者竖屏模式下的文字输入是通过物理键盘（如果可用的情况下）或者屏幕软键盘来完成。

框架和页面的构建

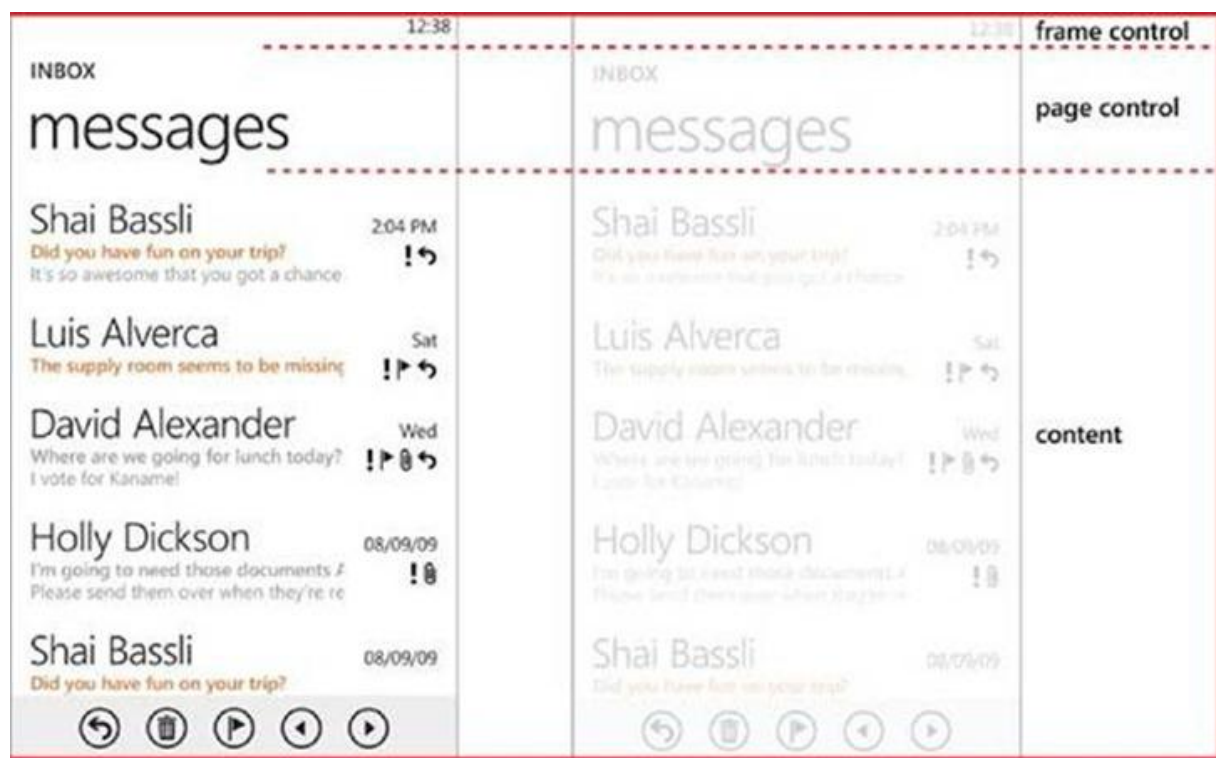
本节将详细为您讲述应用框架和页面的构建。它也将含括 Windows Phone 7 CTP 版的定制化 UI 框架的设计。

备注：

在未来版本中本节将扩展。

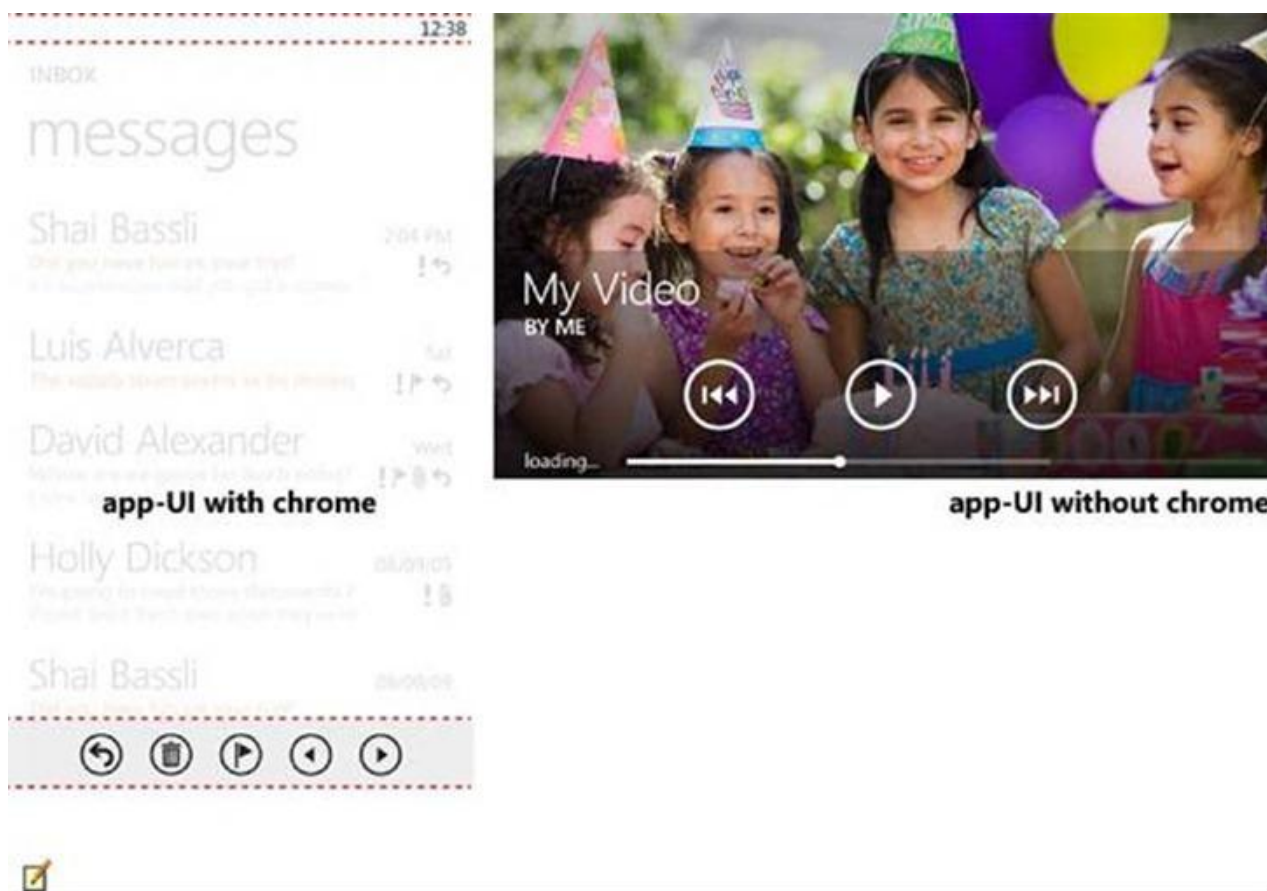
Windows Phone7 CTP 版的应用程序是基于 Silverlight 的页面浏览模式，用户可以在不同内容的屏幕上相互导向和浏览。此外，您可以点击 Windows Phone 的 **back** 硬件按钮来返回到上一个页面。这个模型的设计目标是减轻基于浏览的应用程序的开发难度，使其更自然的适合 Windows Phone 7 系列 CTP 的导航模型。

核心元素包含了称为 PhoneApplicationFrame（框架）的顶层容器控件，其可以承载 PhoneApplicationPage（页面）。在应用程序中，页面可以实现内容的分离。Windows Phone 7 系列 CTP 提供框架和页面类，以方便导航到独立的内容章节中。在应用程序中，您可以根据您的需求创建多个不同的页面，这些页面可以通过框架来进行导航。下面的图示说明一个应用程序中框架和页面的层次结构：



定制屏幕UI界面

这是一个页面内容转换的状态，用于转换正常浏览状态和全屏状态。应用程序可以选择是否包括系统托盘或应用程序工具栏。可是开发人员不应该把这些行为实现成自动改变（译注：也就是需要用户触动来改变），例如说，当在 Windows Phone 启动一个游戏的时候会打开游戏设置菜单，用户如果选择启动新游戏的时候，页面会进入全屏模式启动。



备注：

以上被认为是使用完全定制 UI 模式的最佳做法。如果应用程序只是运行全屏模式，所有视觉通知（包含来电通知）将被正常的传递到用户界面。

当您的应用程序进入全屏模式下，系统托盘栏和应用工具栏会变成不可见。开发人员可以隐藏应用程序栏或者把之显示在最表面层。可以通过对显示属性（Page.FullScreen）赋值来决定是否显示应用程序工具栏和系统托盘。

滚动查看器

当内容大于滚动查看器的边界的时候，应用程序会发生滚动。内容被包含在滚动面板下，这个滚动面板将处理页面的滚动。滚动面板会显示出滚动指示器来表示内容是否否长于或宽于页面，以及提示当前内容在整个页面的分布位置。

程序设计时的考虑

- 位于右边滚动指示器用于垂直滚动，而位于底部的滚动指示器用于水平滚动，滚动指示器只在您与页面进行交互的时候才出现的。
- 如果没有手势（gesture）运动（例如点击或者平移）事件，滚动指示器在一段时间内会消失。当有手势运动事件发生时，滚动指示器会重新出现。
- 滚动指示器标明用户在内容的方位。例如，如果用户在页面的顶部，滚动指示器显示在顶部。
- 避免在滚动查看器上使用具有很多内容的列表框控件，这可能会降低性能。

主题

主题是用于个性化的 Window Phone 视觉元素的资源。Windows Phone 7 系列 CTP 的开发人员在创建应用程序的时候从文体的角度（Stylistic standpoint）下可以保持原有设备 UI 外观以及感觉的一致性。这些样式属性包括背景颜色和重点色彩（Accent colors）。主题能确保用户界面元素以及控件呈现的一致性，防止不和谐的用户体验。

此功能的目标包括：

- 在 Windows Phone 7 CTP 版的开发人员可以构建应用程序并且直接在他们的代码中访问主题的属性。
- 开发人员可以明确地改变任何主题的属性，使其与满足他们自己的品牌要求。

使用主题的优点是保持与 Metro 设计原则的一致性和兼容性。开发人员可以直接使用默认的控件设置，而不需要更改的一些通用的属性（例如色彩风格），这些属性的风格将在运行时根据主题文件中设置而更改。在 Windows Phone 下运行的应用程序会自动根据相应的主题而修改视觉效果。此外，开发人员可以在应用程序级别上覆盖主题。例如，某些公司如果需要特显其本身强烈的色彩元素，在开发开发的应用程序的时候可以保持自身的色彩风格。开发人员可以使用自己的资源来覆盖任何主题的属性。但是，他们不能关闭主题。

备注：

主题只包含色彩。字体或者控件大小等其他元素都不能改变。



程序设计时的考虑

- 用户可以选择一个光亮或者灰暗的主题，开发人员在开发 UI 的时候应该考虑到这一点。例如，如果您在整个应用程序中的所有背景都选择为白色。这可能会影响到 LED 显示屏的电池寿命。在开发过程中，你应该永远考虑应用程序背景颜色对电池寿命的影响。
- 用户可以在您的应用程序中选择 5 个不同的重点色彩方案。如果您应用程序中支持重点色彩方案，这些色彩方案将根据用户的原先偏好设定而发生改变。这些重点色彩 方案包括包括橙色，蓝色，绿色和红色。手机的制造商可以自己定制最后一种重点色彩方案。
- 您不能修改系统级别的主题，只能修改您的应用程序级别的主题。
- 如果你明确地设置控件前景或背景的颜色，您应该同时验证在黑白方案主题中是否都可见。如果您选择的颜色是不可见的，要么选择更合适的颜色或者设定背景或前景的对比度。

Windows Phone 应用程序控件

Windows Phone 7 CTP 版的 UI 控件全部基于一个名为“Metro”的设计系统。在 Windows Phone 7 CTP 上，应用程序界面可以由一系列采用“Metro”风格的 Silverlight 控件构成。

Metro 接口呈现的视觉元素极具吸引力，而且界面整体感觉非常清爽，让用户对展现的内容一目了然。因此我们强烈推荐开发人员尽可能地采用“Metro”风格来构建应用。

尽管你的应用程序的界面可以采用其他设计方式，但如果使用推荐的“Metro”设计界面的话，那么 Windows Phone 7 的用户就可以获得一致性的用户体验，无需额外的上手学习。

Windows Phone 7 CTP 中基于“Metro”风格的 Silverlight 控件将会在本章详细讲解，至于那些在 Windows Phone 7 CTP 版本中暂缺或者不支持的控件也会逐一列出。

提示：本文只讲解 Windows Phone 7 CTP 所具备的控件集。

<http://www.WPMind.com>

Windows Phone 资讯观点、应用软件、开发技术交流专业站！

Windows Phone 7支持的控件

在 CTP 版中你只能使用 ListBox 控件，而不是 ListView 控件。

Border	Button	Canvas	Check Box	Content Control
Content Presenter	Control	Grid	Hyperlink Button	Image
In Presenter	List Box	Media Element	Multi Scale Image	Panel
Password Box	Progress Bar	Radio Button	Scroll Viewer	Slider
Stack Panel	Text Block	Text Box	Phone Application Page	List View Item
List View	Toggle Control	Phone Application Frame	User Control	

Windows Phone 7暂不支持的控件

Combo Box	Message Box	Open File Dialog	Save File Dialog	Tool Tip
			Frame (Use Phone Application Frame Class)	Page (Use Phone Application Page Class)
Calendar	Data Grid	Date Picker	Tab Control	Tree View
Gide Splitter	Label	Scroll Bar		

对于上述不支持的控件，请参照下面的 Windows Phone 7 CTP 备注信息：

- ✓ combox box、scroll viewer、tool tip 这三个控件其实是在应用程序中使用的，只不过这些控件没有在工具栏显示，使用你只能通过代码添加。
- ✓ 对于 MessageBox 控件来说，Windows Phone 7 CTP 其实是支持的，只不过在后续的版本中这个控件可能会被舍弃，但微软还没有做出最后的决定。
- ✓ Windows Phone 7 采用的是隔离存储系统，和以往的文件系统存在很大的差别，所以打开文件对话框控件和保持文件对话框控件也不被支持。
- ✓ Calendar、data grid 、date picker、grid splitter、tab 和 tree view 等控件均来源于 Silverlight 3.0，并不适合 Windows Phone 7 应用程序使用。所以你需要这些控件的话，很不幸，你必须自己开发或者使用第三方的控件

本节重点讲解 Windows Phone 7 CTP 所支持的控件以及在 WP7 应用程序的用法。

提示：在 Windows Phone 7 CTP 中，你需要为 Toggle Switch、List Box、List View Item、TextBlock 手动定义“Metor”的样式设计。更多信息请参照 Windows Phone 7 开发文档。

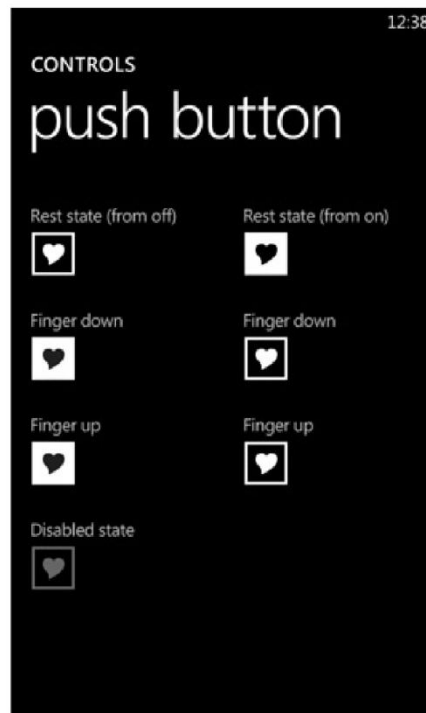
- ✓ Push Button
- ✓ Toggle Switch
- ✓ Check Box
- ✓ Radio Button
- ✓ Hyperlink Control
- ✓ Slider Control
- ✓ Text Block
- ✓ Text Box
- ✓ List Box
- ✓ List View Item

- ✓ Progress Bar
- ✓ Page Title

下面两个控件在 Windows Phone 7 CTP 版中暂不支持。但是在你的程序开发过程可能会使用到类似的用户体验，为此本节也对这两个控件有所涉及。

- ✓ Panorama Application
- ✓ Pivot Control

Push Button



Push Button 控件用来响应用户的单击操作，该控件的形状通常为矩形，你可以在 Push Button 中添加文本和图片信息。

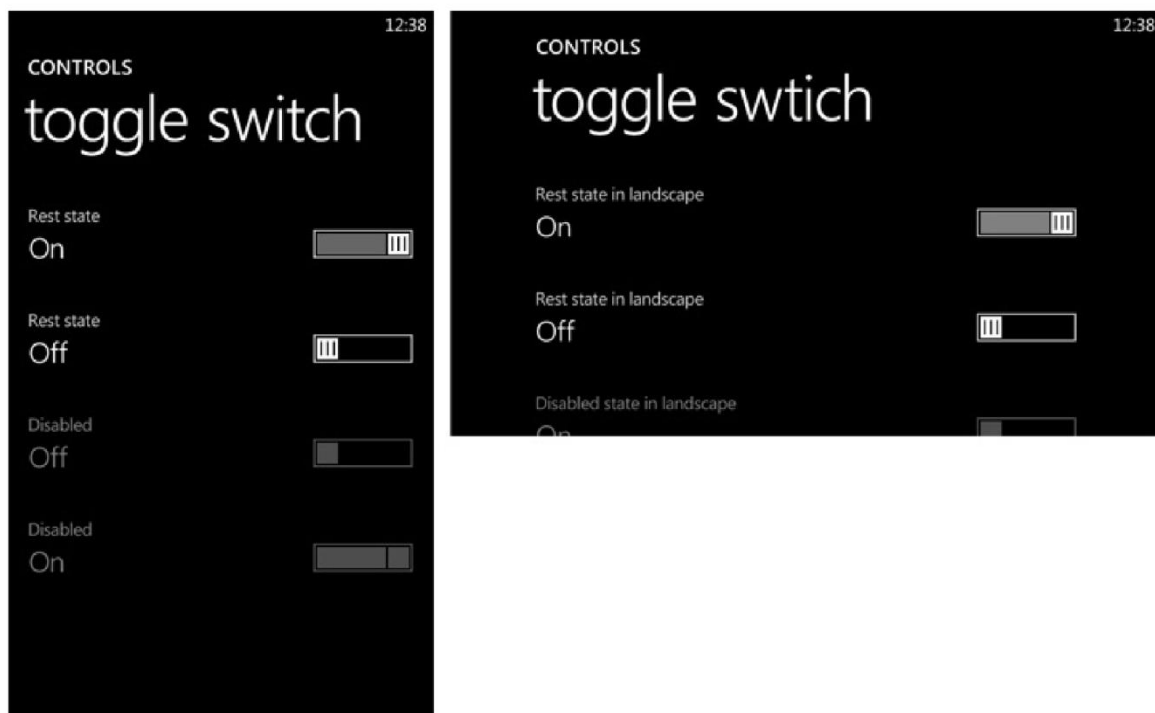
该控件的特性包括：

- ✓ Push Button 支持 rest、press、disable 状态
- ✓ Push Button 没有可视化的焦点状态

程序设计时的考虑：

- ✓ 手势支持，Push Button 支持单击手势。
- ✓ Push Button 支持自定义的样式和主题。
- ✓ Push Button 的文本信息最好不要超过两个英文单词
- ✓ Push Button 的文本最好为简明扼要的动词。

Toggle Switch



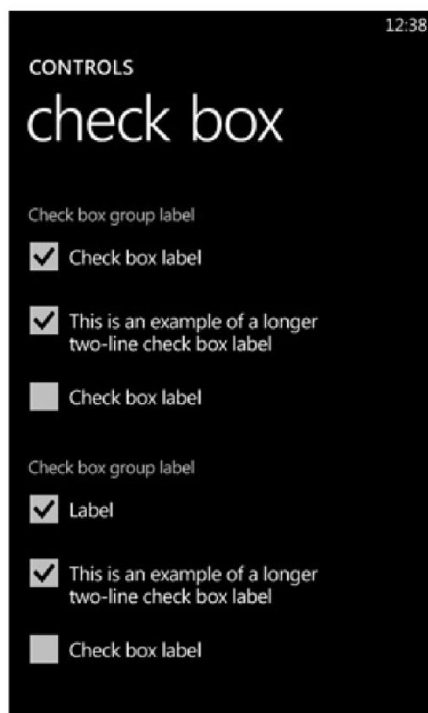
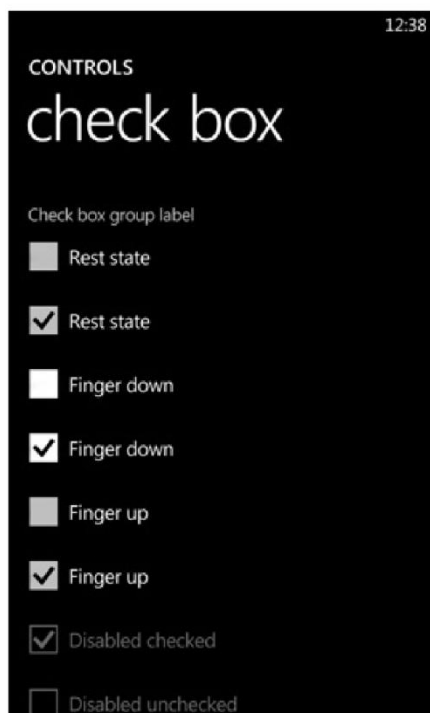
Toggle Switch 控件主旨就是二者选其一，用户可以通过点击控件的方式在 on 和 off 两个选项间进行选择操作。在 Windows Phone 7 CTP 版本中 Toggle Switch 控件已经可用。

- ✓ Toggle Switch 无论是处于 on 和 off 设置时，都支持 rest 和 disable 两个状态。
- ✓ Toggle Switch 支持自定义样式和主题。
- ✓ Toggle Switch 没有可视化的焦点状态。

程序设计时的考虑：

- ✓ 手势支持：Toggle Switch 支持单击手势。
- ✓ Toggle Switch 控件最好酌情使用。

Check Box



Check Box

控件旨在为用户在一个列表中提供多项功能。用户可以点击该控件的选择框或者文本进行选择或取消选择操作。尽管该控件支持多行的文本排列方式，但是为了设计美观，最好还是将选择提示文本保持在一到两行范围内。

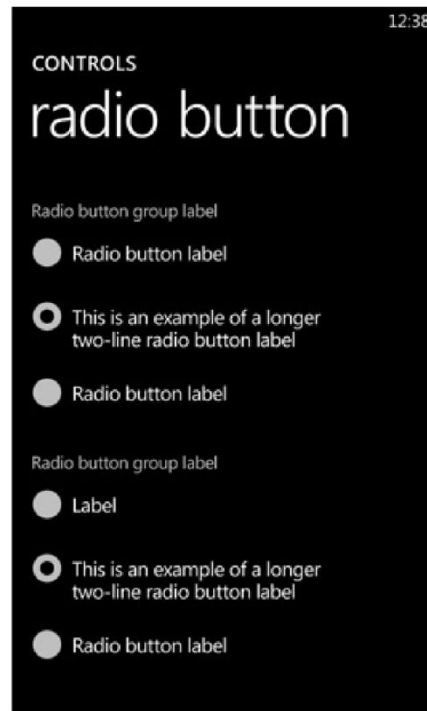
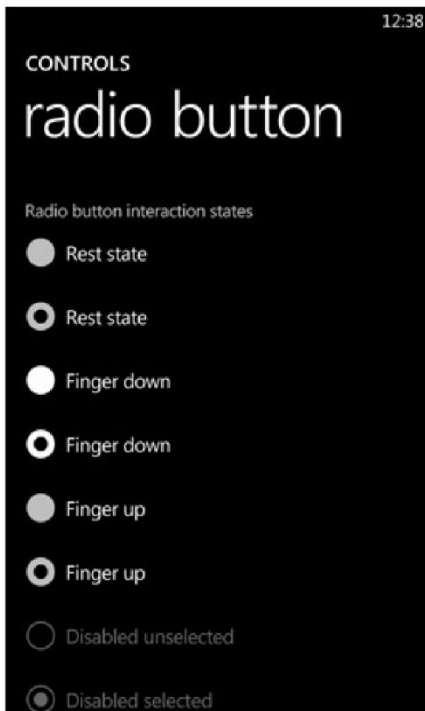
该控件的特性包括：

- ✓ Check Box 控件无论处于选择还是未选择状态，都支持 rest、press、disable 状态
- ✓ 没有可视化的焦点状态
- ✓ Check Box 控件支持主题
- ✓ Check Box 还包含一个不确定状态

程序设计时的考虑：

- ✓ 如果有多个选择，可以考虑使用 scroll viewer 控件和 Stack Panel 面板
- ✓ 手势支持：Check Box 控件支持单击手势，点击的区域可以是该控件的选择框，也可以是文本标签位置。每次点击都会使 Check Box 的当前状态方式变化，在选择和未选择间进行切换。
- ✓ Check box 控件包含一个不确定状态，用于为用户提供除选择、不选择外的第三种选择。我们不建议使用不确定状态，因为这样用户会不清楚控件到底是处于选择还是未选择的状态，导致莫名的困惑。碰到需要不确定状态的功能时，最好是使用单独的 checkboxes 控件来绑定数据源，或者使用多选择列表。如果是动态数据源的话，多选择列表相对来说更加合适。
- ✓ 如果你希望在包含标签的布局中获得丰富的支持，不推荐使用 Check Box 控件

Radio Button



radio

button 主要功能是从一个选项列表中选择一项，并且同一个组的单选按钮是互斥的。用户可以点击 radio button 的文本进行选取操作。radio button 中一次只能有一个选项被选中。

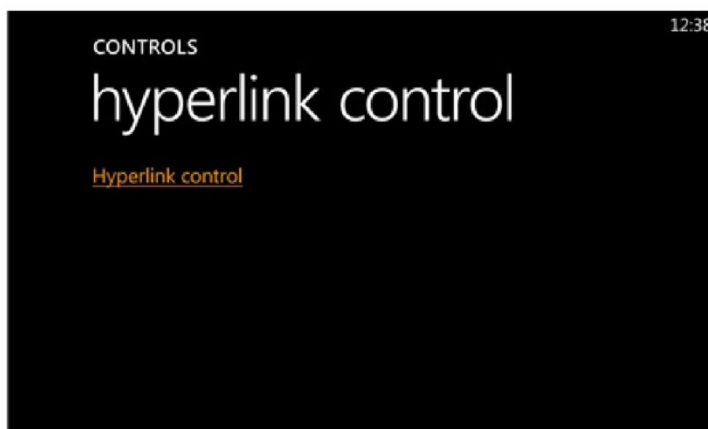
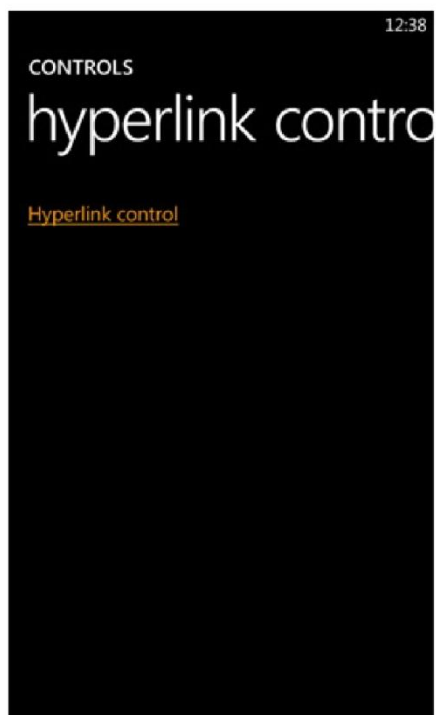
radio button 的一些特性包括：

- ✓ 处于选中或者未选中时均包含 rest, press 和 disabled 三个状态
- ✓ radio button 控件支持主题
- ✓ 没有可视化的焦点状态

程序设计时的考虑：

- ✓ 如果要同时选择多个项，那么你可以考虑使用 list box 控件的 list view 选项进行取代
- ✓ radio button 的文本可以分行显示。能而，从设计观点来看最好是将文本保持在一、两行的范围。通用，应用程序也应该为额外的文本呈现提供足够的呈现空间。
- ✓ 该控件包含一个不确定的状态，用于为用户提供除选择、不选择外的第三种选择。我们不建议使用不确定状态，因为这样用户会不清楚控件到底是处于选择还是未选择的状态，导致莫名的困惑。碰到需要不确定状态的功能时，最好是使用单独的 checkboxes 控件来绑定数据源，或者使用多选择列表。如果是动态数据源的话，多选择列表相对来说更加合适。
- ✓ 手势支持：radio button 支持在文本区域和选择栏上的 tap 手势，点击后会触发选择和未选择状态间的轮换。

Hyperlink Control



Hyperlink

控件允许你在页面中嵌入一个超链接，该控件只应用于导航功能。

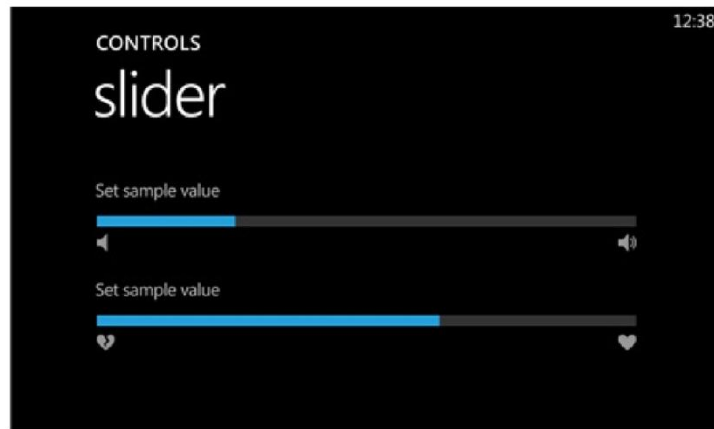
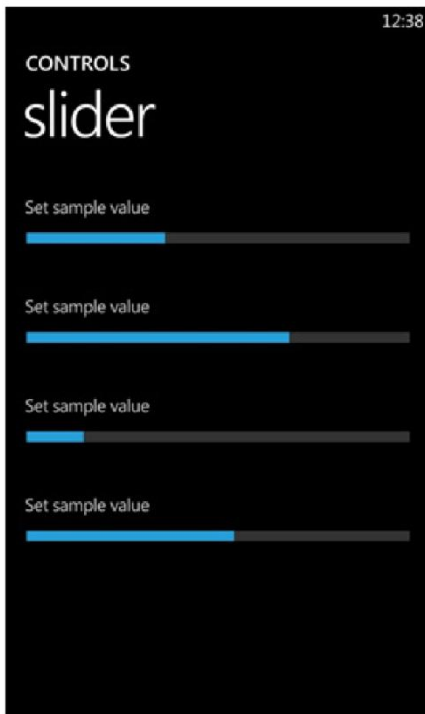
Hyperlink 控件的一些特性包括：

- ✓ 支持 **rest** 和 **press** 两种状态
- ✓ 你可以指定一个导航对象
- ✓ 没有可视化的焦点状态
- ✓ 支持主题

程序设计时的考虑

- ✓ 不要将多个 Hyperlink 控件相互邻近，否则用户很难知道那个链接是他所期望的跳转。
- ✓ 当用户希望显示或者隐藏额外的文本时，最好是采用 **push Button** 控件，而不是 Hyperlink 控件。比如说，当你希望使用 **More Details...** 或者“More Info”之类的文本时。

Slider Control

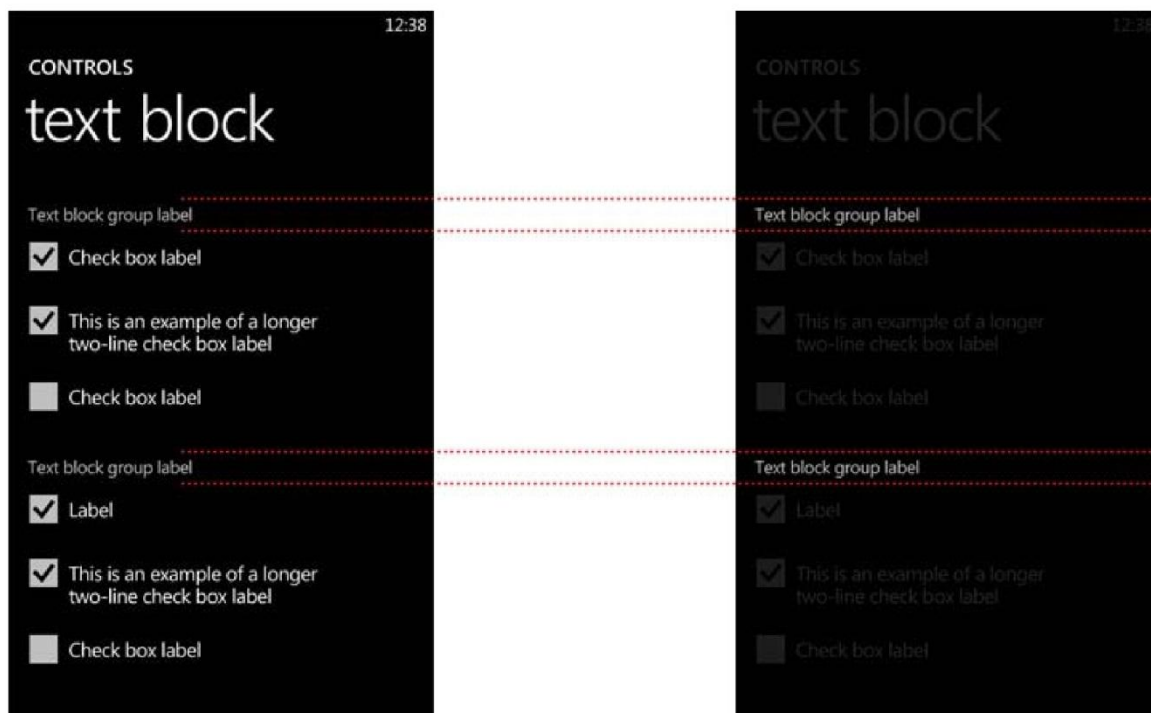


slider control 使用滑块来选取一个连续区域范围内的值。比如说，你可以通过该按钮设置音量、屏幕亮度的值。slider control 包含最小值、最大值和递增值。当 slider control 被按下时，还可以显示一个进度指示器。

程序设计时的考虑：

slider control 控件可以垂直或者水平呈现，但水平呈现比较符合我们的使用习惯。

Text Block



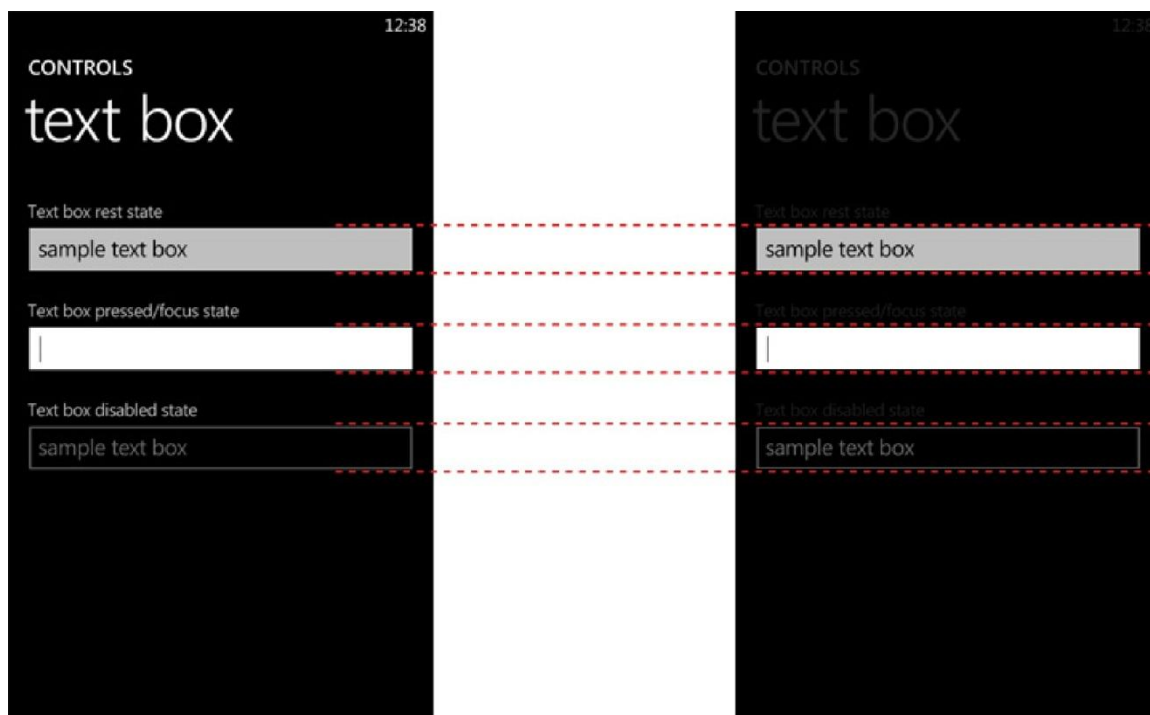
Text block 控件用于显示文本信息，我们可以使用 Text block 控件进行标签或者分组控制。Text block 控件和其关联的控件保持一致的状态。Text block 控件支持主题和自动换行

Text Box

Text box 控件允许呈现内容，同时也允许用户输入或者编辑内容。它可以设置为只读状态，但是通常情况下我们还是希望通过它来编辑信息。Text box 控件支持一行和多行显示，多行显示取决于在文本框内的文字数量的多少。

Text box 控件的特性包括：

- ✓ Text box 支持只读和编辑两种状态
- ✓ Text box 支持主题
- ✓ 当 text box 控件获得焦点时，会自动弹出软键盘提供信息输入



程序设计时的考虑

- ✓ 为了给 Text box 控件提供适合的输入方式，你需要给 Text box 控件配置 input scope 属性
- ✓ 手势支持：支持 tap 手势以便 Text box 控件处于选中状态，获取焦点
- ✓ Password text box 控件中，你可以指定密码混淆

List Box

列表框控件(List Box)包含一个数据项的集合，可以通过绑定数据源或者显示自由数据项（unbound items）来构建。列表框是一个项目控件，你也可以用来填充其他控件或者文本。比如，Windows Phone 7 CTP 列表框项目控件可以用来放置不同的表达元素和点击目标元素。

备注：

对于 CTP release 版本，在你的应用程序中必须合理地设计该控件。更多详情请参考 [Windows Phone 开发文档中的基础控件](#)（Base Controls）小节。

List View Item

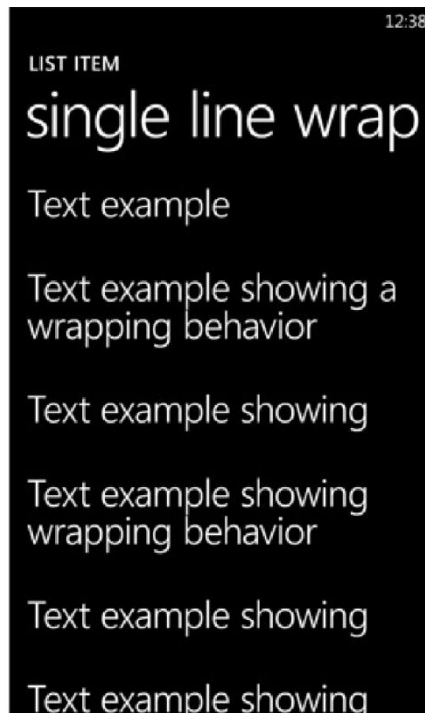
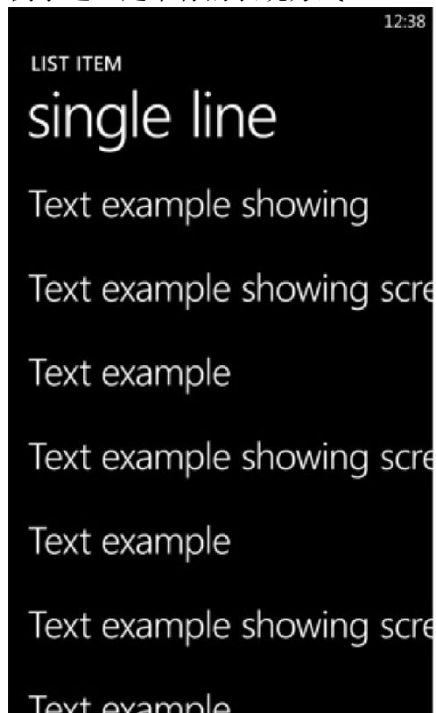
在 Windows Phone 7 CTP 中，列表控件为开发者提供了一种系统表示数据内容的方法。列表视图项(List View Item)是显示在列表中的矩形视图元素。在 Windows Phone 7 CTP 中，开发者可以使用多种风格的列表视图项。

备注：

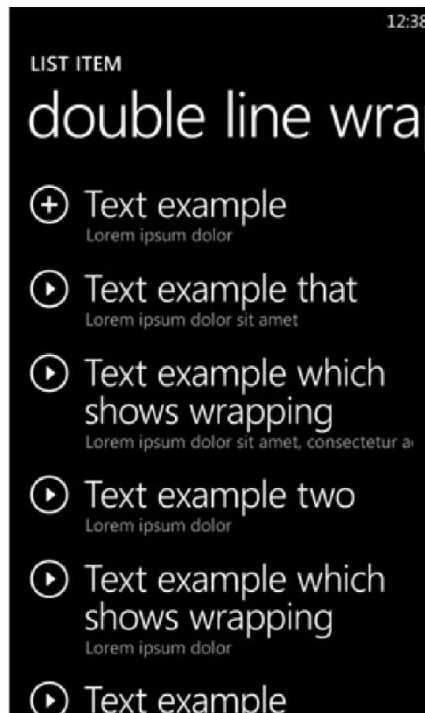
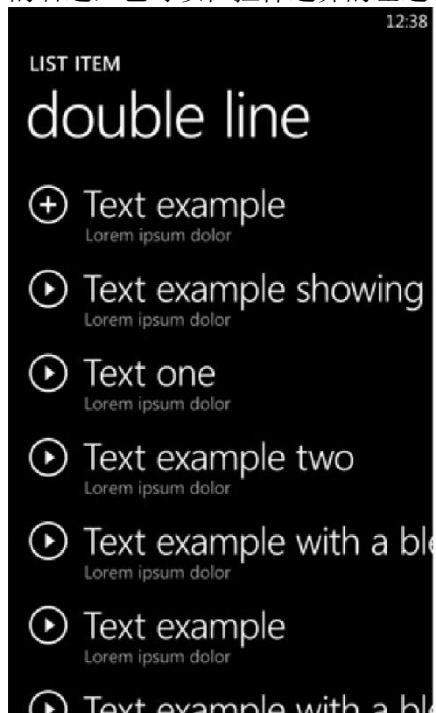
对于 CTP release 版本，在你的应用程序中必须合理地设计该控件。更多详情请参考 [Windows Phone 开发文档中的基础控件](#)（Base Controls）小节。

- ✓ 单行
- ✓ 双行
- ✓ 带图标的单行
- ✓ 带图标的双行
- ✓ 带次级点击目标的单行或双行

例子之一是单行的表现方式。



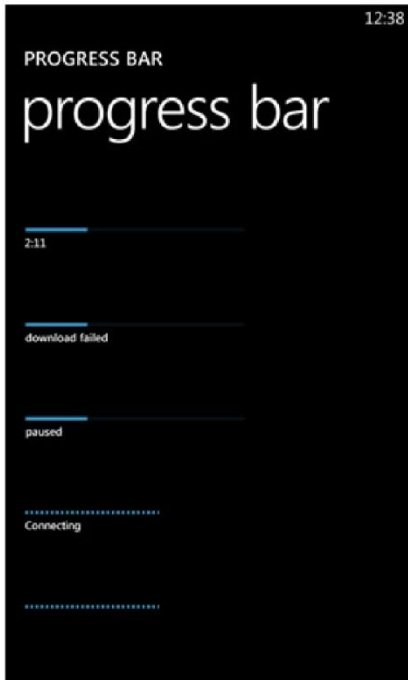
例子之二是带图标的双行格式。该控件支持具有图像占位符的多行文本，其中图像占位符可以在控件边界的右边，也可以在控件边界的左边。



列表项设计考虑

- ✓ 在使用列表框（List Box）控件时，才可以利用多点击目标。
- ✓ 使用多点击目标时，主要点击目标应大于辅助点击目标。
- ✓ 主要点击目标应该能够进一步引导到应用程序中，辅助点击目标应该完成具体的行动。

Progress Bar



进度条是一个表示某项操作进度的控件。你可以使用该控件来展示普通的进度，或者是根据一个数值改变的进度。它包括的特性有：

- ✓ 进度条支持选取框（不确定）模式。
- ✓ 进度条支持主题化。

程序设计时的考虑

- ✓ 进度条具有启用和禁用状态。当它被启用时，你可以和它进行交互，如暂停进度条。

Page Title

尽管页标题并不是一个有用的控件，在这里我们还是要讨论它。页标题控件用来清楚地显示该页内容的信息。



程序设计时的考虑

- ✓ 页标题控件不支持滚动。
- ✓ 应用程序可以选择显示或者不显示页标题。如果应用程序选择显示页标题，那么为了考虑一致性，应用程序所有页面的页标题控件应该被预留，这样，用户就不会感受到不同的窗口大小。
- ✓ 如果应用程序显示了页标题，它应该是该应用程序的名称，或者是和显示数据相关的一行描述性的文字。页标题控件不具备动作。

Panorama Application

全景视图是 Windows Phone OS 7.0 CTP 核心体验的一部分。标准应用（standard applications）受手机屏幕界限的局限，与标准应用不同，全景视图应用利用一个超出手机屏幕局限的长水平画布提供一种独特的方式来浏览控件、数据和服务。这些内在的动态应用利用分层的动画和内容，实现了层与层之间以不同速度平滑过渡，就和视差效果类似。

当前，没有一个全景应用模板或者控件是作为标准应用平台的一部分来提供的。但是，开发者们可以利用 Silverlight 来创建类似的应用体验。



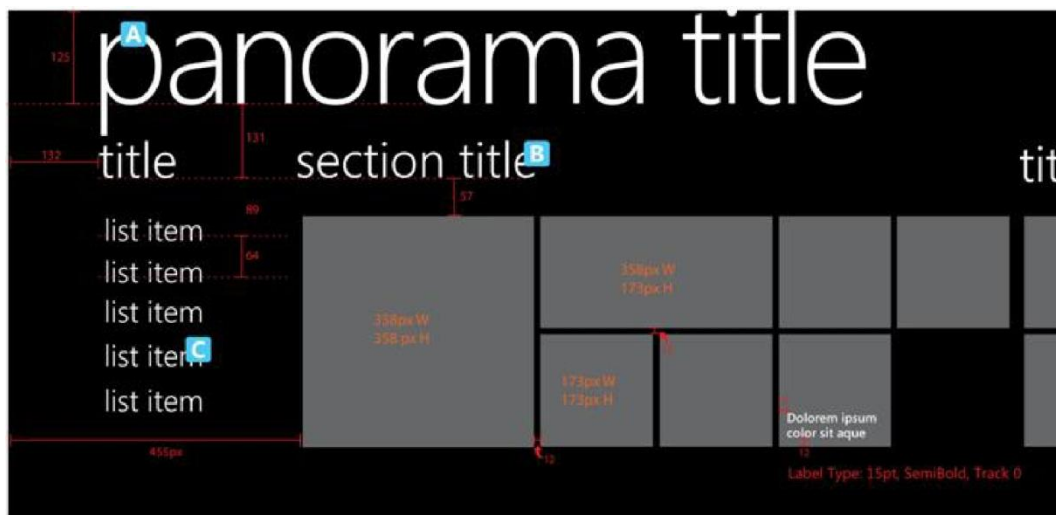
设计建议

对于自定义 Silverlight 全景应用，虽然没有官方的限制，但是 Microsoft 还是给出了一些建议。这些建议的目的是帮助你反映集成的全景视图应用，并且为人们提供更加一致的体验。



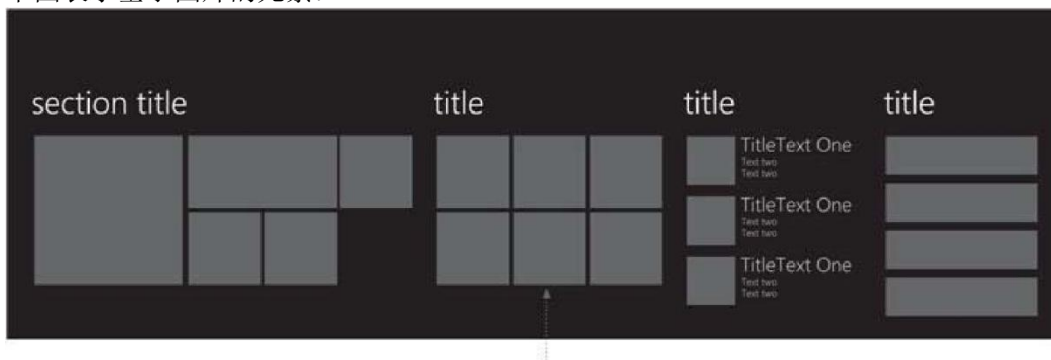
常见的UI元素（Common UI Elements）

全景应用的用户接口由 4 层类型组成：背景图片、全景标题、全景区域和全景区域标题，它们有各自独立的动作逻辑。

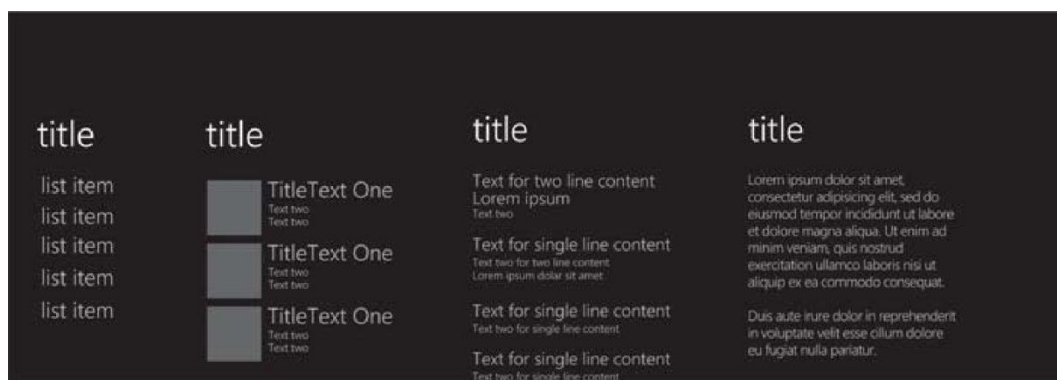


- A** Segoe UI Light 140 pt -10 Tracking
B Segoe UI Semilight 54 pt -10 Tracking
C Segoe UI Semilight 24 pt -10 Tracking

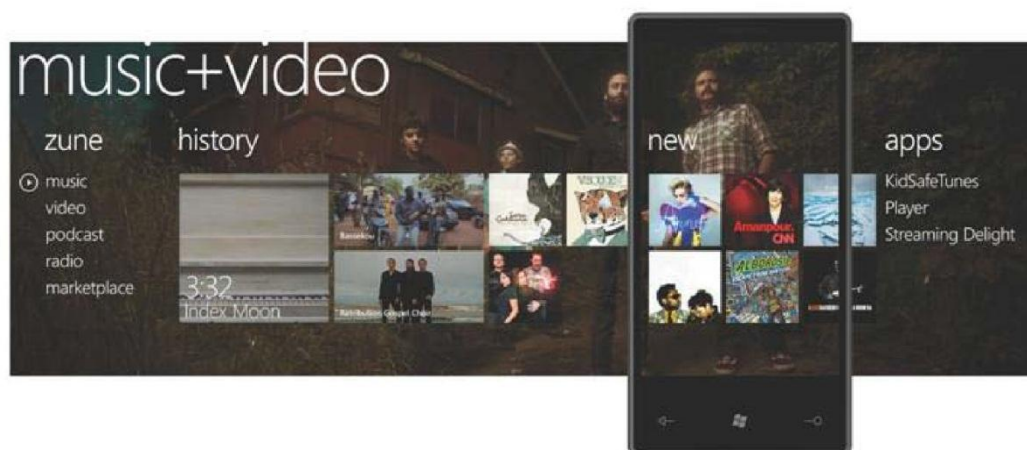
下图表示基于图片的元素：



下图表示基于文本的元素：



背景图片（Background Image）



背景图片位于全景应用的最底层，由它来给出类似于杂志的体验。背景图片通常是一张全景图，它可能是应用程序最直观的部分。如何创建一个好的应用体验，我们在设计过程中，必须牢记以下因素：

- ✓ 利用单色的背景，或者是跨度为整个全景的图片。如果你决定使用图片，从大小来考虑，你可能会使用 JPG 图片，但是 Silverlight 支持的任何 UI 图片类型都是可以接受的。
- ✓ 可以使用多个图片作为背景，但是在任一时刻，只能显示其中一张。
- ✓ 为了确保良好的程序性能，最少的加载时间，并且无需剪裁，图片大小应该在 800 x 480 和 800 x 1024 像素（高 x 宽）中选择。
- ✓ 对于一个具备 4 个全景区域的应用，使用 16 x 9 的屏幕高宽比。
- ✓ 为了提高文本的易读性，使用一个透明的黑色或者白色过滤器。
- ✓ 在动态 UI 元素上，避免使用下拉阴影效果（drop-shadow effects）。
- ✓ 使用一定比例的与平移手势相关的动作，该平移手势和顶层内容宽度与背景图片的宽度比例有关。
- ✓ 只有背景艺术出现在应用中时，才使用动画。
- ✓ 当用户的平移手势超出图片的宽度时，关闭并且返回可见区域。

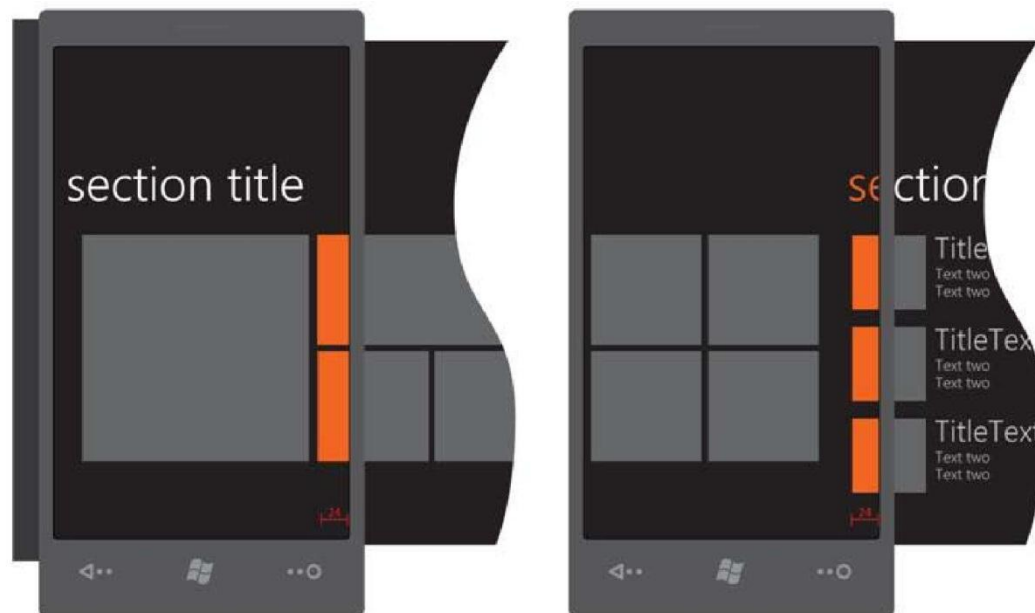
全景标题（Panorama Title）



全景标题是整个全景应用的标题。其目的是让用户识别该应用，无论是以何种方式进入应用，它都必须是可见的。下面是全景标题的设计建议：

- ✓ 使用简洁的文字或者图片，例如一个 logo 作为全景标题。使用多个 UI 元素，例如一个 logo 加文字（或者其他 UI 元素）也是可以接受的。
- ✓ 确保字体或者图片的颜色与整个背景相匹配，而且，标题的可视性不依赖于背景图片。
- ✓ 为了保持一致性体验，在 **Start** 菜单中的应用程序名称和该标题一致。
- ✓ 避免标题动画，或者动态改变标题的字体。
- ✓ 使用一定比例的动作，相对于最顶层内容来说较慢，而相对于底层图片来说较快。
- ✓ 当用户的平移手势超出图片的宽度时，关闭并且返回可见区域。

全景区域（Panorama Sections）



全景区域是全景应用的组成部分，它封装了其他控件和内容。以下是全景区域的设计建议：

- ✓ 最大化利用四个全景区域，确保全景应用的平滑性能。
- ✓ 在列表或者网格内使用垂直滚动是可以接受的，但前提是它处于全景区域内，并且不同时与水平滚动出现。
- ✓ 只要全景区域的宽度小于屏幕的宽度，垂直滚动是可以接受的。
- ✓ 支持所有自定义控件和标准控件。
- ✓ 与手指拖拽的移动比例相同。
- ✓ 当用户导向到一个新的区域时，开启屏幕动画。

- ✓ 设计全景区域的布局，使得少量的下一个全景区域可见。提供轻微的重叠，使得用户直觉地利用 Pan 手势来切换应用。
- ✓ 直到该全景区域有内容要表示时，才显示该全景区域。

全景区域标题（Panorama Section Titles）

全景区域标题是全景区域的可选部分。如果你提供标题，考虑下面的设计建议：

- ✓ 尽管可以使用图片，最好使用简洁的文本。使用多个元素，例如一个图片加文字（或者其他 UI 元素）也是可以的。
- ✓ 确保全景区域标题不依赖于背景图片。
- ✓ 避免使用标题动画，因为标题可以移动。
- ✓ 跨越整个区域，即使存在多个控件。
- ✓ 当用户导向到一个新的区域时，开启屏幕动画。

注：

根据全景区域的宽度是否大于或者小于屏幕的宽度，该全景区域的标题应该不同。如果全景区域的标题宽，它就需要水平的动画。也就是说，标题不应该停留在区域的左上方，而应该随着全景应用的移动，以不同的速度移动。在这种情况下，不应该使用垂直滚动。相反地，如果全景区域的宽度小于屏幕的宽度，标题就应该一直停留在该区域的左上方。在这种情况下，不应该使用水平滚动的动画，如果使用了垂直滚动，标题应该随着内容移动。

小图片规范

小图片是全景浏览中的主要元素。它们链接到全景以外的内容或者媒体。如下图所示，你应该使用剪裁的图像来强调一个主题，而不是一整张图片。如果图片没有文本标注，可以使用两行的文字来描述该内容。



Thumbnail without system text



Font: Segoe Semibold, 15pt, Track 0



12



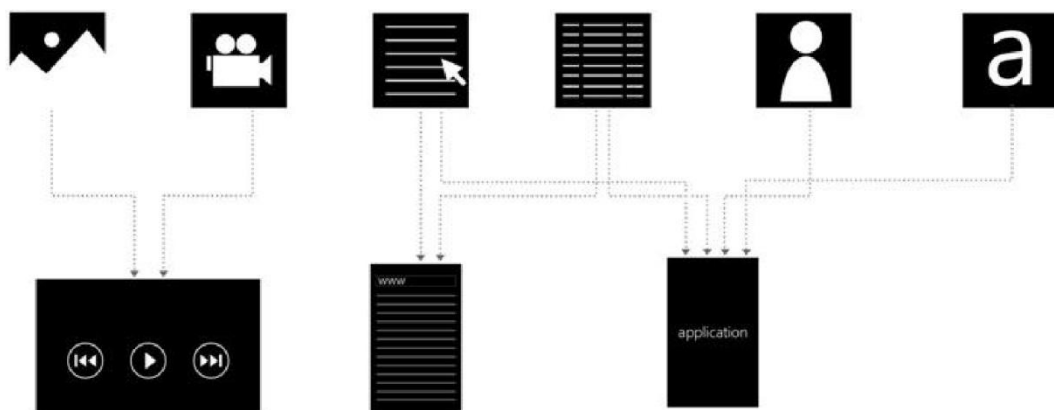
Thumbnail with text, no masking needed



Thumbnail with text + 25% black mask over picture

元素流程（Element Flow）

全景应用的元素作为更加细致体验的起点。下图描述了手机上核心全景应用的标准流程。



注:

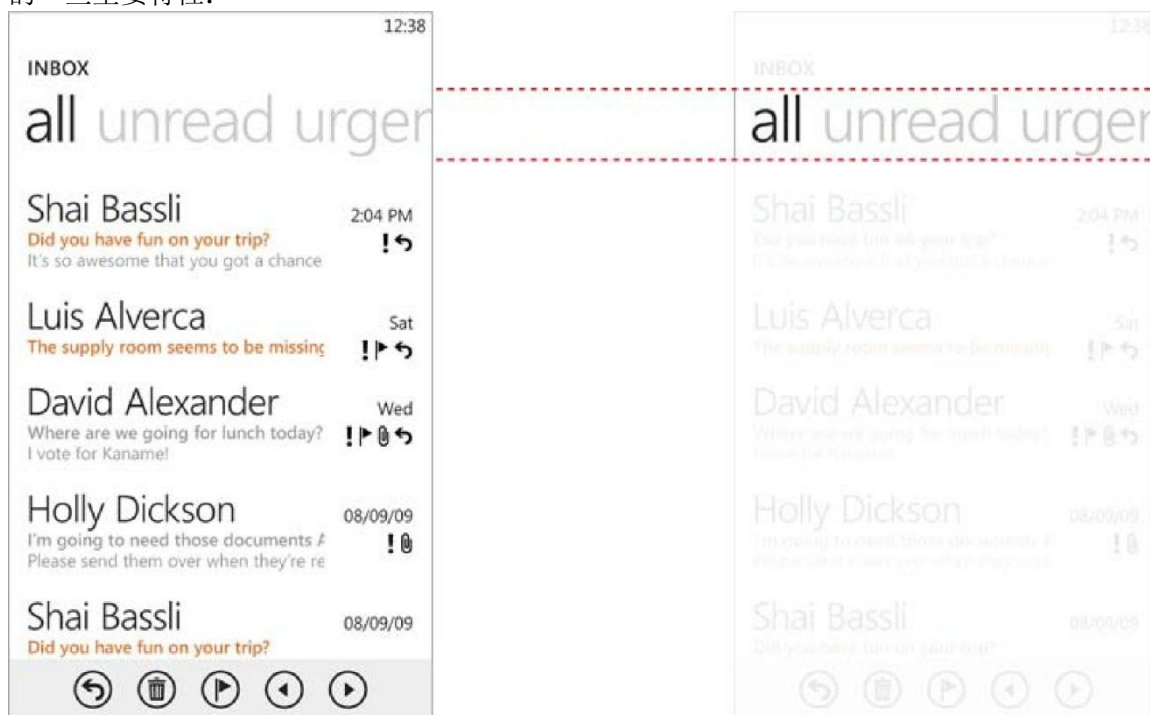
上面展示的元素流程并非指示平台的功能，而是终端用户的体验。例如，在一个全景应用中启动另一个应用程序，在终端用户看来，刚刚启动的应用程序只不过是相同全景应用的不同视图而已。

Pivot Control

枢轴（Pivot）控件提供了一种快速的方式来管理应用中的视图或页面。枢轴可以用来过滤大的数据集，浏览多个数据集，或者切换应用视图。枢轴控件水平并且相邻放置独立的视图，同时管理左侧和右侧的导航。可以通过划动或者平移手势来推动枢轴控件。

注:

在 CTP Release 版本中，该控件不可用。但是，在开发你自己的枢轴控件时，以下的设计考虑突出了它的一些主要特性：



程序设计时的考虑

- ✓ 应用程序应该使得 pivot 页最少。
- ✓ pivot 页面的内容由应用程序定义。

- ✓ pivot 页面是循环的。
- ✓ pivot 页面不能覆盖水平 pan 和水平 flick 功能，因为它与枢轴控件的交互设计相冲突。
- ✓ pivot 标题文字的长度没有限制。显示文字的数量受制于枢轴控件的宽度。
- ✓ pivot 标题高度固定，不能改变。
- ✓ 枢轴控件应该只用来显示相同类型的条目或者数据。
- ✓ 该控件不应该用来做任务流程。不同的页面应该在外观方面具备无缝切换。页面不应该极大地改变用户的活动。
- ✓ 枢轴控件应该少使用，在适当情况下才使用。
- ✓ 在用户没有可能添加信息的情况下，一个空的 pivot 页应该被删除。

Windows Phone 系统控件

这部分是针对Windows Phone 7系列CTP版本的系统控件，主要讨论如何使用这些控件以及他们对你应用程序的影响。

- ✓ 系统托盘和应用工具栏
- ✓ 上下文菜单
- ✓ Windows Phone消息通知
- ✓ 进度显示器

系统托盘和应用工具栏

Windows Phone 7 CTP版本包含2个主要的组件：系统托盘和应用工具栏

系统托盘 – 一个可以用一种简洁的方式显示系统的状态信息的指示条，预留在应用程序工作区。用户可触摸系统托盘，因为它提供了不同的更新通知，并让它们知道事情是重要的

应用工具栏 - 应用工具条提供了一个放置最常见的任务和意见的地方。应用工具栏还提供了一个单一的，可选的应用程序菜单。相同的任务和意见将以图形按钮的方式显示在工具栏上，附加在上的任务组成一个菜单。在下面的图中，系统托盘和应用酒吧代表了应用外框。



提示：用“FullScreen”属性来显示或隐藏系统托盘和应用工具栏。“Visible”属性用来设置应用工具栏是否显示，“Opacity”来设置应用工具栏的透明度

下面的图例说明了这些元素在一个应用程序中的位置：



应用工具栏的一些特性包括:

- ✓ 应用工具栏停靠在整个应用程序的底部，宽度可以从左边延伸到屏幕的最右边的边缘。
- ✓ 应用工具栏的高度是固定不变的。
- ✓ 在工具栏中的按钮会以圆圈形状从左到右排列。
- ✓ 当在应用工具栏上出现应用项时，工具栏上会有一些点来标明。你可以利用这些点，弹出应用栏菜单。
- ✓ 应用工具栏按钮是可以被禁用的。比如，在应用工具栏上有个删除按钮。为了防止不恰当的删除操作，你可以选择禁止这些按钮。你可以将他们设置为只读的。
- ✓ 用户可以通过点击非菜单区域，让应用工具栏菜单关闭。
- ✓ 在应用工具栏上最多放置 4 个子项。你应该在应用工具菜单中放一些不经常用，但又不是自动处理的任务。

程序设计时的考虑

- ✓ 应用工具栏尽量使用系统默认的主题颜色，除非你没办法，必须使用自定义颜色。因为使用自定义颜色是会影响按钮的图片质量，会引起一些特殊的菜单效果，甚至在一些显示类型上可以影响功耗。
- ✓ 可以很好的调节应用工具条的透明度，但是建议你使用这些数值(0, 0.5, 1)。如果应用工具栏的透明度小于 1 的话，整个应用工具栏就会覆盖在 UI 的上面。如果透明度设为 1，那整个页面的显示大小将会改变手势支持：你可以轻轻滑动或点击应用工具栏来弹出应用工具菜单。
- ✓ 应用工具栏上的图标必须是 48 x 48 像素。如果图标是其它尺寸，将会自动缩放到该尺寸，但这样的话图片质量就会变差。
- ✓ 图标图片应该使用白色前景色，并在透明背景下使用 Alpha 像素。应用工具栏将会根据当前样式的设置上色，彩色图标会导致此效果难以预测。
- ✓ 每一个圆圈图标按钮是由应用工具栏来绘制的，但不会包括原图标图片。
- ✓ 在你的应用程序中，主要使用图标按钮是最常用的操作。但是对于某些操作是无法用 1 个图标来表示清楚的。对于这样的情况，可以使用菜单项。如果不是必须的话，就不要占用应用工具栏的 4 个图标位置。
- ✓ 如果菜单项中的文本太长，就会导致屏幕无法显示完全。一般菜单项中的文字长度在 14 至 20 个字符之间。这是推荐的菜单项文本的最大长度。

上下文菜单

上下文菜单为用户提供了额外的方法来访问特定的功能。上下文菜单不出现滚动条的情况，推荐最多出现5个菜单项。



提示:

虽然这部分会显示系统默认版本的上下文菜单，但对于开发人员来说还是需要自己应用程序中创建自定义的上下文菜单。下面部分将提供一些新特性和帮助指导，这样可以有效的帮助你创建菜单类型。

一些特性包括:

- ✓ 上下文菜单会一直停留在屏幕上，除非用户关闭它。其它选项包括点击菜单外部区域，选择设备上的关闭按钮，或者选择了一个菜单项。同样的，一些系统事件也会关闭上下文菜单。
- ✓ 手势支持：你可以点下去，并按住一个列表项，同样可以弹出上下文菜单。



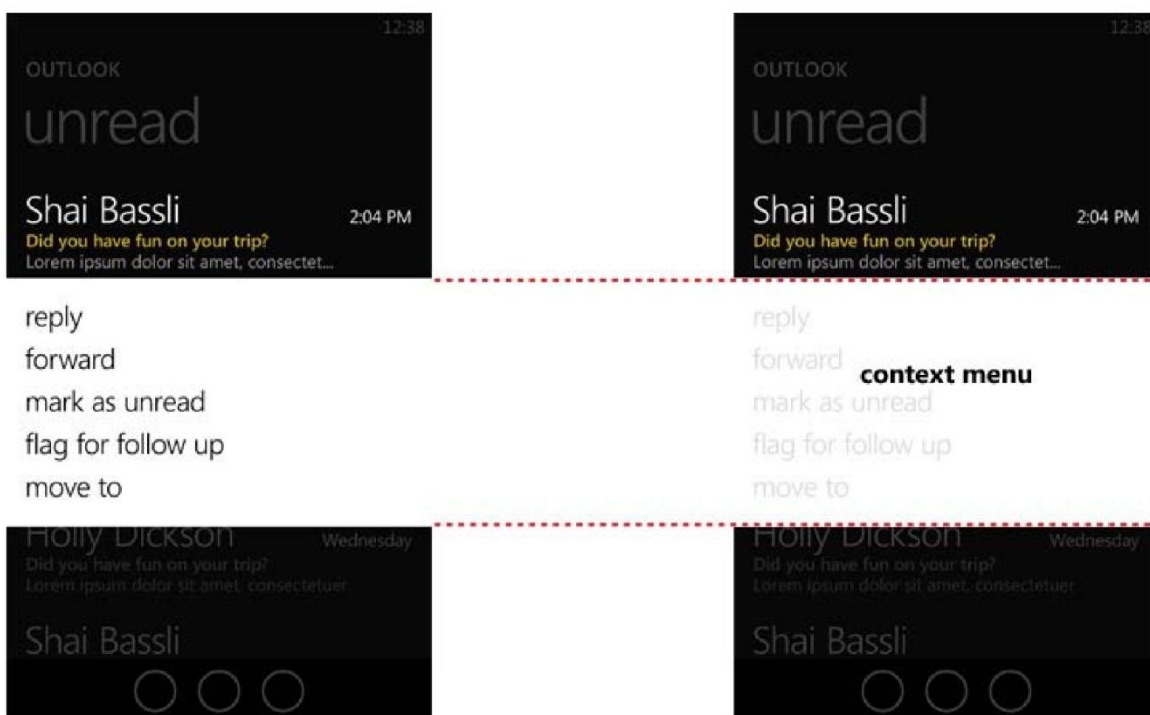
提示:

上下文菜单不是完成任务的唯一方法。它只是一种加快导航的一种方法。

Windows Phone 消息通知

在Windows Phone 7 CTP版本中，提供了几种在设备上显示通知的方法。通知类型如下:

- ✓ 感知类通知，通知用户发生的哪些改变或者哪些事件，是不会中断当前用户的行为的。他们出现在开始菜单中称之为“瓷片”的区域
- ✓ 要求采取行动的通知是全系统的通知，可以不会中断行为或中断用户现有行为要求立刻解决。这些通知最终用户接收文本信息或即时信息。
- ✓ 在应用程序通知中的行为通知，是完全受应用程序控制，并且只会应用该应用程序。



推送通知概况

对于应用程序开发，推送通知服务旨在提供一个专用的云服务，利用具有弹性和持续性的渠道推送通知到移动设备上。当云服务需要发送一个推送通知到设备上时，它会先发送一个通知请求到推送通知服务，它会通过路由通知应用程序，或到设备时作为一个提醒通知，这依赖于发送时的通知类型。在设备上的推送客户端通过通知管道接收推送通知。当管道被创建后，同时会创建一个订阅，允许云服务推送通知到这个管道里。该管道用一个URI表示，包含与订阅相关的所有信息。一旦一个应用程序接收到推送通知，它可以使用云服务协议来访问云服务以获取所需要的信息。

推送通知类型

Tiles和toast通知是2种机制，和启动云服务的发送有关，在用户应用的用户界面以外提供可操作的反馈。此外，云服务可以发送原始通知要求。根据发送的通知类型，通知将会被送到任何应用程序或系统内核中。

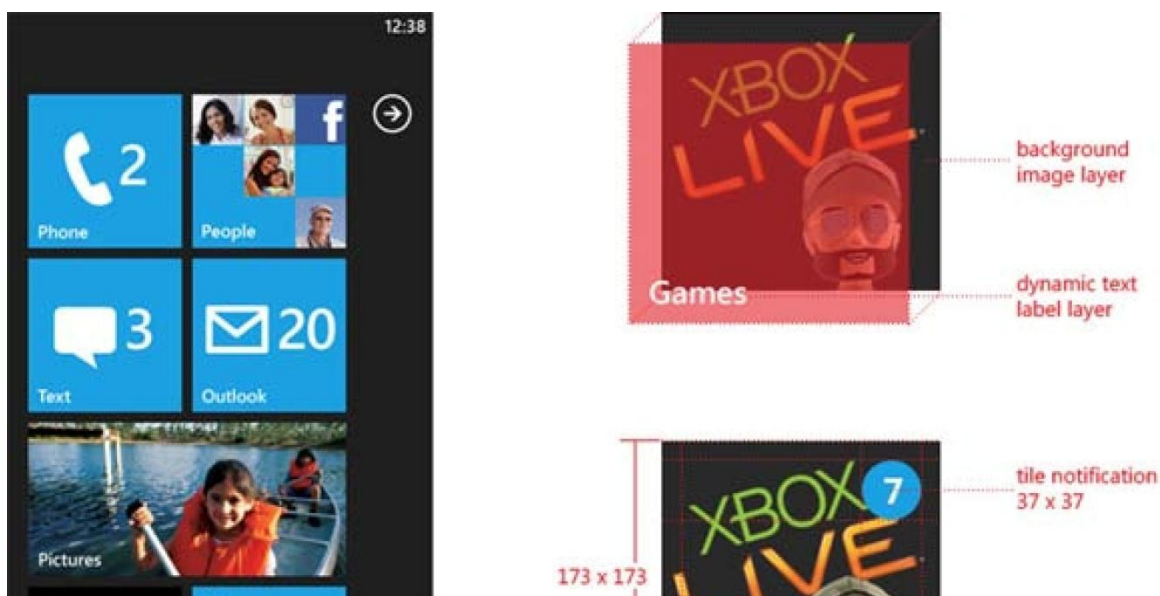
Tile通知

Tile是看得见的，在手机上的开始体验中，是一个应用程序的动态的表示或者在快速启动区的内容显示。举个例子，天气应用程序可以选择当地时间和气候条件显示在Tile中。因为云服务可以在任何时候修改Tile的外观，在这个基础上，这个机制可以用来给用户获取信息。在用户手机上的每个应用程序都可以使用一个Tile，但只有用户可以控制哪些Tile可以在快速启动区显示。但目前是没有办法让应用程序知道它的Tile是否已在快速启动区显示。

云服务可以控制Tile的背景图片，计数器以及标题属性。使用Windows Phone开发工具就能设置这些属性。声音和动画也是由该平台所控制的，而不是由应用程序本身。举个例子，如果该平台设置了任何一个Tile更新时都会有声音和动画，那不管怎么样，任何一个Tile都会生效。

Tile的背景图片可以使用本地资源（这是应用程序部署的一部分）也可以使用云服务资源。通过引用云资源，应用程序可以启动动态更新Tile的背景图片。在背景图片显示前，会启用一个场景来处理下该背景图片。在所有场景中，应用程序包都必须包含所有Tile所必需的背景图片，因为这才是性能和电池寿命的最佳方案。

下面几张图片是Tile和Tile的转换图片所表现的UI:



提示:

图片可以是JPG或者是PNG格式. Tile通知计数器在应用程序构建中是一个可选组建. 更多关于如何创建或修改Tile, 可以参考: <http://go.microsoft.com/fwlink/?LinkId=185182>.

Toast通知

云服务可以生成一个特殊的推送通知称为**Toast**通知, 它是可以叠加方式显示在用户屏幕上. 举个例子, **Toast**通知可以由即时消息客户端, 或者由其它点对点的通信应用产生. 当用户决定点击**Toast**通知, 应用程序就会执行其它行为. 云服务可以控制**Tile**的标题和子标题. **Toast**通知也同样会显示应用程序的图标 (图标必须包含在应用程序安装包中).

应该注意以下关于**Toast**的通知:

- ✓ **Toast**通知应该和时间相关联.
- ✓ 他们应该主要集中在点对点的通信上.
- ✓ 应用程序必须默认**Toast**通知是关闭的. 用户必须在应用程序UI上或者“设置”菜单中来启用**Toast**通知.

Raw通知

如果应用程序希望从云服务接收消息但又不想用推送通知的话, 那就需要不断地轮询云服务(使用 **http** 长连接池). 通过使用推送通知服务, 应用程序可以从云服务接收 **Raw** 通知, 但必须是在前台运行. 这就需要应用程序使用自定义的代码来建立并保持和云服务的连接. 如果应用程序运行在后端, **Raw** 通知将会被推送通知服务丢弃并且不会通知设备.

进度显示器

进度显示器显示与应用程序相关的操作或一系列事件的运行程序. 系统控制集成到系统托盘并且可以显示在多个应用程序页.



进度显示器可以是确定或不确定的：
确定的必须有一个开始和结束点
不确定的就必须进行一直到任务完成。

UI文字注意事项

该主题提供了 Windows Phone 7 CTP 版的 UI 界面的建议指导文本。虽然是可选的，以下的准则可以帮助你写应用程序中的文字时提供可信的、友好的、舒适的体验。如果需要，这里的许多建议都可以帮助你的 UI 界面可以和 Windows Phone 7 CTP 版本里原来的文字格式保持一致。

Windows Phone 7 语音语调

这个章节里将描述在 Windows Phone 7 CTP 版本中的语音和语调。它作为一些元素可以在不同应用之间呈现不同变化。然而，开发人员应该意识到语音语调如何呈现在 UI 文字上。同时，他们应该保持在整个应用程序中语音和语调的一致性。

声音是表达文字的个性。例如，作家的声音是他们写作中的整体性格的表现。理想情况下，Windows Phone 7 系列 CTP 版本的应用程序的语音应当真实、清晰的反应当前使用者的语言。语调是在整体的文字上呈现出例情绪如快乐或愤怒。对于应用程序而言，我们建议友好、轻松、同情的语调。千万不能使用生气或毫无语调的生硬的机械音。

在手机上使用语音和语调，就好比一个朋友协助另一个朋友来完成某件事情。这样可以帮助用户来理解出现在应用程序中的一些错误信息。开发人员不应该在出现错误时，给用户提示时提供一个僵化的、不合时宜的信息。例如，绝大多数的最终用户是不会知道 Error Code: 4560363 的。然而，比如一些信息就会丢在那儿“请在输入框中输入你的名字，这样就能到下一个页面”。应该提供用户一个简单、易懂的提示信息或行为。这样可以帮助他们理解如何解决当前的问题。

其它例子包括：

用“同步你的手机”来替代“同步电话设备”。

“在Outlook的日历中计划下明天的安排。”这又是一个不友好的文字例子。较生动的如“嘿，你必须在Outlook中添加一个明天的约会。”

虽然以上是简单的例子，它表明了使用人性化而非硬梆梆的、僵化的语音和语调的重要性。

大小写标准化

在应用程序中保持一致性是很好的做法，这样可以防止阅读体验脱节。在Windows Phone 7 CTP版本的大小写准则中在许多地方使用小写字母和全部大写的布局，并遵循下列准则：

标题大写— 在第一个到最后一个单词组成的短语之间的所有第一个字母都大写。但有一些例外是(a,an,the)定冠词，并列连词(and, but, for, not, or, so, yet)，小于或等于 4 个字母的介词(at, for, into)。比如 **Vitamins in My California Raisins**。

部分大写— 你只需要将短句的第一个单词大写。例外的情况是在句子中出现专用名词或功能、名称词时，该词必须大写。比如 **Vitamins in my California raisins**。

对于 Windows Phone 7 系列 CTP 版本来说，大小写标准化的参考标准如下：

所有小写：

- ✓ 列表控件标题
- ✓ 列表组控件标题
- ✓ 按钮上的文字或者作为命令的单词

- ✓ 列表控件里面的每一项
- ✓ 出现在搜索输入框里的说明文字
- ✓ 在句子里的超链接控件
- ✓ 应用程序里的分页标题

部分大写:

- ✓ 多选控件和单选控件的标签文字
- ✓ 进度条显示器
- ✓ 状态、通知、说明文字
- ✓ 切换开关

所有大写:

- ✓ 应用程序标题
- ✓ 日期和时间
- ✓ 上午或下午

标点符号

下面的表格将对于 UI 元素中的标点符号提供一些规范:

标点符号	使用指导
与符号 (&)	在“设置”或菜单列表中使用,比如,时间&分钟; 时钟&闹钟
冒号(:)	<ul style="list-style-type: none"> ✓ 不要在控件的标签结尾使用,比如文本框,下拉列表,进度条. ✓ 不要在文本框或下拉列表在一个句子里或当下拉列表出现在主窗口时使用 ✓ 不要在控件组标题或控件列头中使用. ✓ 使用冒号时后面可以跟数字或其它变量,比如: 碎片百分比: XX%
逗号(,)	<ul style="list-style-type: none"> ✓ 在句子中出现三个或三个以上的元素时, 元素之间可以使用逗号分隔. ✓ 在最后一个由三个或以上的元素组成的项目前始终使用逗号. 例如: "The car was red, small, and fast."
省略号(...)	<ul style="list-style-type: none"> ✓ 在进度指示器中使用省略号来表示继续进行中,比如,当用户正在下载一个文件. 即便已经有了个进度指示器了, 但你还是要使用省略号. ✓ 不要在标题中使用. ✓ 不要在按钮标签中使用
结尾符 (. ? !)	<ul style="list-style-type: none"> ✓ 仅仅在 UI 界面上的说明文字结尾出使用. ✓ 如果标题栏的说明文字结尾或按钮上结尾都不要使用. ✓ 不要在选择项或复选框最后的文字标签结尾处使用, ✓ 即使标签是一个句子. ✓ 在句子结尾用结束符后, 用空格来分隔句子, ✓ 不要使用 2 个空格. ✓ 在问题结尾用问号结束. ✓ 但在一般情况下,避免使用问号标签. ✓ 在错误提示信息或对话框的结尾可以使用问号.
圆括号 ()	如果可以的话避免出现在用户界面上, 除非你需要使用包括缩写或间断描述的话, 可以使用括号.

UI设计注意事项

- ✓ 要理解适当的利用语音语调可以创建一个更具吸引力的终端用户体验.开发人员应该精心开发应用程序,来提供清晰友好的用户界面.
- ✓ 良好定义的标准化准则将使你的应用程序一致性,放置最终用户在使用时有粗糙的体验.
- ✓ UI设计中必须使用正确的标点符号,同样,在你的应用程序中要保持一致.