

# Windows 8 guide 设计指南翻译

Windows  
Developer  
Preview

Using  
the Design  
Checklist

Design  
Metro  
Style Apps

Guidelines  
and Checklist  
for Tiles

Guidelines  
and Checklist  
for Splash  
Screen

Guidelines  
and Checklist  
for App Bars

Guidelines  
for Snapped  
and Fill Views

What are  
Metro  
Style Apps

Designing  
for Different  
form Factors

Designing  
for Touch

Designing  
for Globalization  
and Localization

Designing  
for your Brand

Designing  
for your Quality

Creating and  
Managing  
Secondary Tiles  
...

# 目录

《Windows Developer Preview Guide - Windows开发者预览指南》	1
《What 's Metro Style App - 什么是Metro风格的应用？》	35
《Design Metro Style App - 设计Metro风格的应用》	39
《Designing for different form factors - 为不同尺寸设备而设计》	49
《Designing for touch - 为触摸而设计》	57
《Design for your brand - 为品牌而设计》	67
《Design for your quality - 为质量而设计》	69
《Design for globalization and localization - 为全球化及本地化而设计》	71
《Using the design checklist – 使用设计检查表》	75
《All About Tile - 磁贴设计规范与检查表》	85
《secondary tile - 二级磁贴设计规范与检查表》	89
《Guidelines and checklist for splash screen - 启动画面设计规范与检查表》	93
《Guidelines and checklist for app bars - 应用栏—设计规范与检查表》	97
《Guidelines for snapped and fill views - 辅屏与全屏—设计规范与检查表》	101

# Windows 开发者 预览

Windows系统的再构想	01	打造量身定制的网络	22
全新而漂亮的用户界面	03	针对Internet Explore10而开发	22
Metro风格	03	Live Connect开发平台	23
一切从此开始	05	Windows 推送提醒服务	23
触摸键盘	05	联系人选择器	23
一个Windows--几种形态	06	使用最现代流行的工具和架构进行开发	23
应用之力	07	Visual Studio 2011, Express 和Expression Blend在Windows 8系统下完美运行	23
你的应用如影随形	09	用户体验数据	24
应用可以协同工作	09	创建于广泛用户基础的Windows应用商店之中	24
时刻在线	10	最大化你的收入	25
开箱即得的优秀应用	10	使用应用内购买 ( In APP Purchase ) 选项	25
IE10:为触摸设计的网页浏览	10	在世界各地贩售	25
触屏浏览, 不仅仅是在触屏设备上进行浏览	11	一条贩售应用更好的途径	25
以你的站点为中心	11	技术基础, 设备,及安全	27
体验美观的网页	11	技术基础	28
Windows商店	11	系统要求	28
桌面	13	在ARM, X86及X64架构下运行的应用	28
兼容性	15	连通性	29
为高级用户做的更多	15	无线网络和移动宽带	29
轻松管理文件	15	设备	29
任务管理器	15	Metro风格的设备应用	29
多显示器选项	16	媒介驱动	29
一键重置	16	安全	29
Windows更新	16	Metro 风格的应用运行于一个更安全的环境中	29
应用开发与Windows 8	17	密码系统的简化	30
Windows 运行库和应用模型	19	Smart screen筛选过滤	30
可供广泛选择的应用开发语言	19	Windows 安全防护	30
同现有架构的兼容性	19	安全启动	30
使用扩展架构建立控件库	19	Windows 8与商业	31
外壳集中扩展	19	无处不在的移动性和连通性	32
为你的应用添加Metro风格	20	Windows To Go	33
用带有适应性布局来创造沉浸感的用户界面	20	DirectAccess	33
图形	20	灵活地获取文件信息	33
使用DirectX创建沉浸式的游戏	20	数据安全和管理的	33
媒体和传感器	21	应用锁	33
整合音频和视频	21	比特锁	33
利用传感器来建立为用户量身定制的体验	21	测量启动	34
	21	改善的活动目录激活	34
	21	客户端 Hyper-V	34
		流畅性部署	34

# Windows

如今超过三分之二的PC是移动设备，比如笔记本、上网本、平板电脑。几乎每一台PC都有无线连接的能力。屏幕的尺寸也是多种多样：从10寸之内到壁挂式的监视屏再到数个高清屏组。存储能力已经从M级增长到T级，且正步入云端。多核处理和强大的图形显示技术也广泛普及。拥有丰富的功能的触屏手机彼此汇集，改变了我们对计算的观念。社会化技术，云服务以及通过各类应用获取网络服务等爆炸式的发展，给人们新的强大的连接和共享方法。现代运算更关注人和应用而不再是操作系统。这些宏观格局上的变化促进了Windows系统的诸多重大变革。

# 再构想

但是，重新设想Windows并不意味着从头开始。Windows7的基础特性，像速度、可靠性、安全性和兼容性使之成为微软有史以来最好的操作系统。正在设计中的下一版本的Windows（代号Win8）正是基于Win7的那些显著的优点上设计搭建的，而且将提供更丰富的安全特性，更快的启动，更长的电池待机时间且适应更加广泛的设备和芯片组。Win8用一个新的为触屏优化的界面去扩展这些基本特性。Win8也提供这样的一个平台，让人能创造基于现代的web标准全新一代的全屏应用，并且通过新的Windows商店供人们使用。我们已经重新设计了windows的体验，并且构建了你所喜欢的各种东西。

# 全新而漂亮的 用户界面

## Metro风格

Windows8引入了一种新的Metro风格的界面。它显示对你而言那些重要的信息，形式简单，并能自由控制。这种界面布局个性化，排版简洁，动画清晰，让人同个人电脑之间的互动流畅且直观。新的界面为触屏构建。通过触摸，你可以迅速而自然的做你想做的事情。新的“开始”屏幕，将你所有的应用直接放在你眼前，触手可及。你不需要记住他们放在哪。不论你是要输入，点击，或者滑动，优化了的Windows8让导航更加容易，不论使用手指还是使用键盘和鼠标，你都能在操作系统中四处移动时不费吹灰之力。



你的应用和内容在主屏上一瞥既得

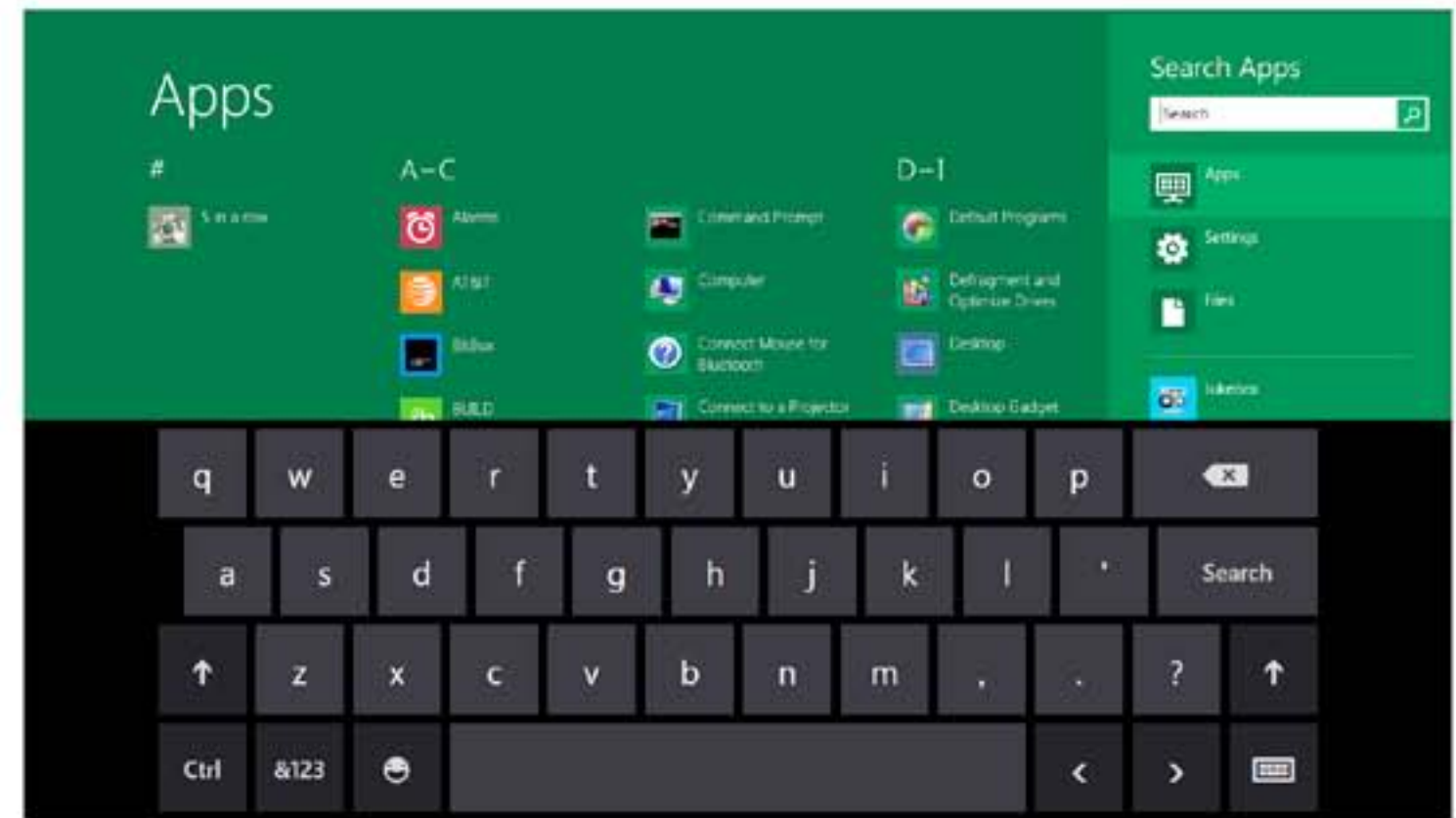
## 一切从此开始

新的开始界面给你一个快捷的方式让你和应用及内容进行交互。磁贴中的信息是活动的，在任何时候都显示给你正在发生的变化。你关心的信息——天气、朋友的动态更新、在奈飞排行榜上的排名靠前的电影、你的下一个日程安排——全在那里一目了然。你可以自定义你的开始界面，给你最喜欢的事情放置快捷方式，像网页、播放列表、或者相册。你可以决定哪些东西出现在主屏上以及你要如何对他们进行分组。通过在各应用分组之间增加“减速带”来优化使用效率，这样在开始界面中移动时不会产生多余的前后滑动来快速定位。

## 触摸式键盘

win8有两个触摸键盘提供终极的移动性和灵活性。一个经过改良的带有大尺寸按钮的全尺寸触摸键盘和一个独一无二的拇指键盘，让你更简单的使用你的计算机。无论你是否坐着还是在走动，当键盘和主机分离时，你都能更舒适和自然的用拇指输入。

Windows 8下的触摸键盘包含了多种功能，能提高打字速度和输入准确性。改进过的键盘技术可以减少不正确的击键并自动更正拼写错误，而不降低输入速度。为了输入更具效率，在输入的过程中，Windows会在屏幕上显示推荐词，只需要轻轻一点进行选择。



按需出现的触摸式键盘

键盘适应不同的语言。在触屏电脑，键盘的布局会自动适应你的选择的系统语言。你的输入和语言设置会自动适应到整个计算机，而不光光是某一个程序。

## 一个Windows——几种形态

对ARM的芯片组、触摸设备和传感器的支持让Windows 8完美的运行在你所选的各类设备上，像全天电池待命的10寸平板、超轻薄的笔记本、和功能强大的27寸高清屏的一体机。无论你用什么设备，Windows内核的改进使得启动时间更快、应用管理更高效、内存占用更低。Windows 8非常强大且响应迅速。

# 应用之力

为Windows 8设计的Metro风格的应用美观、直观和并以你的体验为中心。他们有沉浸感、充满整个屏幕降低干扰。这些应用能够适应许多形状构成因子和屏幕分辨率，像平板或者大型显示器，也能运行在x86、x64和arm的平台上。应用协调工作，使得搜索、共享和应用间发送内容变的很容易。当联网的时，你的应用主动激活并显示最新的内容给你，让你一目了然。

Windows 8提供了日常生活最重要的各类应用，包括一个触摸浏览器。Windows商店提供一切你期望得到的应用。你会找到你要的应用。你可以放心他们的安全性因为我们会对他们进行审核。你将享受到浏览、下载、购买和试用（如果有提供）的灵活性。

你的应用可以在任何安装Windows 8的电脑上试用——台式机、笔记本、平板——不管是你的、你朋友的还是其他家庭成员的。通过一个连接帐号，你就能下载你的应用且不论在哪都可以使用他们。



## 你的应用如影随形

当你在另外一台装有Windows 8的电脑上登录你的微软联机帐号，你的metro风格应用和设置就伴随你，如同使用自己的电脑一样。你也将登录到所有你要登录过的站点。你的连接帐户更像一个便携的个人化的电脑，出现在你使用的任何安装了Windows 8的设备上。不管你在哪，你总是能从你上次里离开的任务状态中开始。

## 应用可以协同工作

在Windows 8里，应用程序相互之间可以通信。所以，如果你要在邮件里面发照片，而这些照片来自与不同的地方比如facebook、flickr或者来自你的硬盘，你只需要很简单的挑出来进行发送。Windows 8将提供一个统一的界面来显示这些信息。



从一个地方选取你要发送和分享的文件

## 时刻在线

Metro风格的应用甚至在电脑待机时仍然能够运行和保持更新。当你连接到互联网，你将收到信息更新、邮件、ip电话、即时通信消息而且能够在背景播放在线音乐或者别的内容。当windows结束待机，你的应用不需要额外的时间来准备。

## 开箱既得的优秀应用

不管时娱乐、还是完成工作、亦或是保持联系，或者提醒备忘，你将拥有你所需要的重要的应用。他们丰富而美观，汇集了你所有的在线内容，并以全屏方式运行，没有任何东西会跳出并干扰你正在做的事情。

## IE10:为触摸设计的网页浏览

Metro风格的ie提供一种快速流畅的触屏优先浏览体验，让你浏览所有站点都更像应用程序。依靠更快的性能、领先的安全保护、更多的硬件加速和对html5站点的支持，ie10允许开发商源源不断的开发出更丰富更美观的站点来。



快速、流畅和直觉化的触摸浏览

触屏化的浏览，不仅仅是在触屏设备上浏览

Internet Explorer 10为触摸浏览而优化，为速度和流畅而设计。基于手势的导航是直观和快速的。你可以平移，缩放和自然导航。当你需要的时候，导航条为导航任务统一的提供触屏友好的控件。Ie10的触屏体验拥有windows系统所有的优点，所以你可以享有分屏应用并快速感受到搜索和共享的魅力。Ie10所提供的浏览体验让网页浏览无缝衔接，如同一切都在存在于Windows 8 的设备上

以你的站点为中心

Internet Explorer 10的Metro风格的界面更加干净和无感知，这样你就能专注于你最喜爱的站点。导航条仅仅在你需要的时候出现，以便让你的页面占有更大的空间。一个在你需要的时候才出现的触摸键盘被智能的整合到浏览器里。像在win7下的ie9,你可以将你喜爱的站点钉在新的windows开始界面上。这使得他们像应用一样，因为你可以轻轻一点就进入网站。如果你喜欢的站点已有一个metro风格的应用，ie10会向你发出提醒，如果你愿意你就能切换过去。

体验更美观的网页

通过利用Windows和你的设备硬件的优势，ie10继续ie9的在加载页面速度上继续发力让你的网站如同瞬间打开。Ie9领先的硬件加速允许网页开发者和设计师去构建更丰富和沉浸的体验。由于现实世界安全和隐私威胁仍然困扰者现代网络，IE10继续成为一个更安全和值得信赖的浏览器。基于Windows的安全平台和世界领先的云端信誉系统Smartscreen，ie10将帮助你更安全的浏览网页。

## Windows商店

新的Windows商店有很多你所熟悉的应用和其他你所喜闻乐见的应用。从激动人心的新游戏到你已经了解并信赖的Windows上面效率型的工具，找到他们都非常的简单。

你可以浏览并比较上千个应用——我们通过对相似的应用进行分组，高亮经常下载的应用和显示其他用户是如何对他们所购买的应用进行评分，以此来让你的选择更加容易。对于许多应用，如果你还有一点顾虑，你可以在购买之前进行试用。一旦你作出决定，就可以安心购买，要知道这些应用都是已经被筛选和病毒检查。你可以把到的应用安装到最多5台Windows 8的个人电脑上。

# 桌面

在x86和x64的电脑上，你惯用的应用，像微软的office和excel、Cad / cam和adobe photoshop，他们仍然是像以前一样在桌面运行。当你要用到这些应用时，你可以从开始屏幕进入。仅通过点击然后进入。Windows在桌面打开你的应用，你可以用触摸或者键盘和鼠标进行操作。

## 兼容性

在x86和x64电脑上，Windows 8支持Windows 7桌面应用程序和设备，所以你不必要妥协或者放弃你惯用的东西。在这些电脑上，你现有的基于Windows 7的应用仍然能够工作。

## 为高级用户做的更多

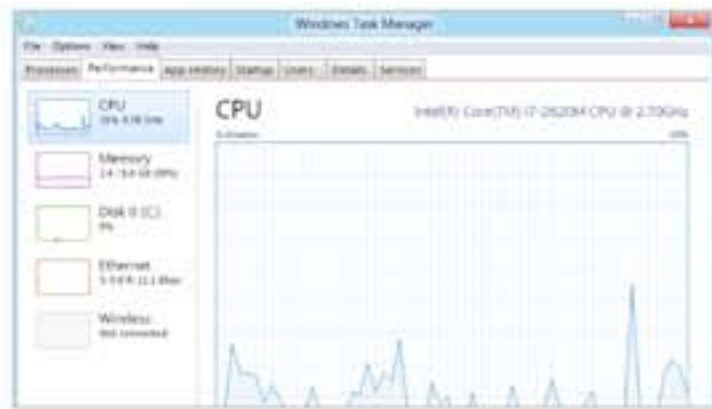
如果你想挑战你的电脑体验极限，我们为一些高级用户常用的功能做了相应的投入。我们重新设计了Windows文件浏览和任务管理器去极大地提高工作效率同时也开发了新的，灵活的多监视器的设置方案。定制新的弹性的多显示起设置。

### 轻松管理文件

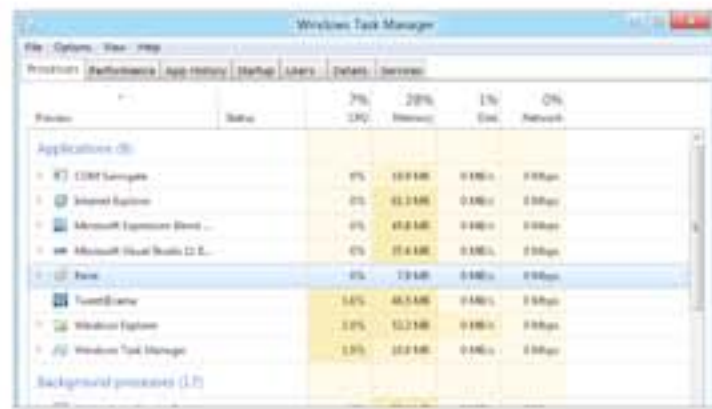
我们已经加强了Windows的文件管理，使文件的导航和常用文件管理任务更加直接简明。磁贴控件管理了组织管理器丰富的功能，预先显示隐藏的特性，也有如ISO磁盘图像的新控件，以及和单窗口移动或者拷贝。我们带回了一个散失多年的最爱：向上按钮。我们还引入了新的增强拷贝体验，这样你就可以在一个地方查看和管理所有的文件。

### 任务管理器

新的任务管理器提供了易于使用的集成面板用来监视和控制你个人电脑，因此你总是知道你的电脑的状态。信息用带有颜色的磁贴表现，让你注意到消耗最多资源的条目都是哪些。。利用摘要图表和处理过程中、应用和历史纪录的详细信息，你可以保持对系统效率的实时跟踪。用多监视器的选项，你可以看到同每一个进程相连接的服务。



任务管理器显示了你电脑的运行状态



进程标签卡显示了电脑资源的使用情况

## 多显示器选项

多监视器的使用比以前更具灵活性。新的选项允许你把开始界面单独显示在一个监视器，把桌面显示另一个。显示不同的桌面背景在每一个监视器上，或者放你的图片于两者之间。也设有选项可以让每一个监视器中显示复制的或者独立的任务栏。



多显示器现在可以作为一个更大工作空间

### 一键重置

没有什么比电脑故障花费数小时去重置系统到初始状态或者花钱请人帮忙更让人沮丧的事了。Windows 8提供了一系列方法让你重载电脑到初始状态。你可以做一次最基础的重置，这能让你非常简便的从新开始，而你的文档、帐号、个人设置和任何从微软商店购买的应用都能得到保留。

Windows 8也提供一些新的，高级的设置项。你可以做完全的复位，重置你的电脑到你最初购买时的状态。你也可以建一个复位磁盘，这对投入时间去优化和设置他们电脑的高级用户来就是一个福音。从usb密钥是一个备份重置你的电脑的拇指大的驱动器体型小，省空间，当你的硬盘损坏或者需要更换的时候，又是非常方便的备份工具。

## Windows 更新

没人喜欢当他们正在做一些重要的事情的时候被打断。同时，保持电脑时时最新又是很重要的。

Windows 8能自动处理并让电脑保持最新，在不打扰你的情况下包揽了绝大部分的维护行为。如果你安装时开启了windows的更新，windows会在电脑空闲的时间或者在你指定的时间去安装那些更新。倘若更新后需要重启，有一个宽限期，允许你挑选一个方便的时间去重启。因此，你会很少看到有关维护，更新或者重启的各种提示。你将不会由于被打断从而降低工作效率，而你的电脑也会持续运行良好。

# 应用开发 与Windows 8

伟大的应用始于伟大的开发者。Windows 8给你提供一个平台和工具，构建让你的顾客可以专注于他们自己重视的任务的丰富应用体验。应用是Windows8体验的中心。它们通过动态和不断更新的内容而显得生动而富有生机。你的顾客会沉浸于你的全屏，Metro风格的应用中，这会让他们更加专注于他们关注的内容而不是操作系统的界面。

登录并通过Windows应用商店来出售你的应用，意味着你可以接触到广大通过Windows系统来访的遍及全球的客户。你可以取用一系列丰富的新工具和服务来更快速和更经济地开发和部署你的应用。当然，你也可以通过你所选择的商业平台接触并融入到新的客户群中去。

通过Windows 8你可以凭借你现有的技能和代码资产来为你顾客创建更好的体验。

- 网络开发人员可以使用HTML5, CSS3和JavaScript的技能来搭建本地Windows应用程序。
- .NET开发人员能够使用XAML, C#, 和Visual Basic来建立漂亮的Metro风格的应用。
- 游戏开发人员可以使用强大的DirectX 11.1来营造华丽且令人投入的游戏体验。
- 驱动开发人员可以使用新的，整合过的Microsoft Visual Studio 开发环境来提高生产效率。

使用Windows 8，你就可以开始想像，创建并开始贩售新一代的伟大应用。

## Windows 运行库和应用模型

Windows 运行库是一组直接顺向的API，用以建立Metro风格的应用。WinRT让你所建立的应用在你的客户眼中看起来更好更直观。WinRT 的API有多种语言可供开发者选择，包括JavaScript, C++, C#, 和 Visual Basic. 为Metro风格应用准备的新的Windows软件开发工具包含了一个带有传统Win 32，控件模型，.NET Framework API的子集。同时HTML5和CSS3 的API也可被Metro风格的应用开发者所使用。

## 可供广泛选择的应用开发语言

为了建立起Windows8的电脑中最重要的部分——完全本地化以及强有力应用，你可以使用网页编码的技术或者所熟悉的Windows编码技术来进行开发。使用Javascript并借用HTML5和CSS3的优势来搭建用户界面的Windows Metro风格的应用，可以依据JS来实现其应用逻辑。使用C++, C#, 或者Visual Basic (用 XAML Markup)来生成用户界面的Windows Metro风格的应用，则用C++, C#, 或者Visual Basic来实现其应用逻辑。游戏开发者能够充分利用图形硬件使用C++ 和 DirectX 11.1 创建Metro风格的游戏，或者使用HTML5或XAML开发休闲游戏。

## 同现有架构的兼容性

WinRT 下的API能够接受来自于其他微软程序框架的访问，比如像.NET框架或者Windows C 运行库。桌面应用能够在代码中包含关键字的方法来使用WinRT的功能。这些功能含有应用程序二进制接口代理 (ABI)，能够让Windows的API被其他编程语言所访问。这个二进制接口被展示元数据组合资料中(.winmd 文件)，这样可以让静态语言（如C#）或者动态语言(如JavaScript)来理解WinRT API的结构。

## 使用扩展架构建立控件库

新的Windows8应用模型有本地化扩展功能，通过这些扩展功能你可以建立供自己复用的控件库。最开始你可以利用C++, C#, 或者Visual Basic来建立自定义的控件。然后你可以运用JavaScript或者其他支持语言从你的应用中调用这些控件。

## 外壳集中扩展

当创建了一个Metro风格的应用，你便得到了高度整合后的新Windows8的各类特性和功能，包括应用程序栏，边缘，动态磁贴，以及同其他应用之间的联系。你的应用将加入到触摸优先的界面以及现代的用户体验的行列之中。当应用运行的时，你的应用将会成为注意的焦点而操作系统的窗口和框架将退隐至背景。

## 为你的应用添加Metro风格

通过使用Windows8的控件，你的应用可以得到可预见的为设备量身定做的Metro风格的界面。这些控件被设计成可同时应用于触摸设备和鼠标键盘设备。默认情况下，你的应用将表达出用户所熟知的用户体验的Windows特征。以下是这些你可以利用的控件：

**标准控件**：这包括一些你需要呈现，加入，操控数据和内容。控件包，如浏览，文本，格式，重叠，媒体（音频和视频），内容，收藏和基础等。

**集成控件**：这些控件辅助设计师创建丰富统一的，且适用于触摸操作的内容体验。这些控件包含植入式的拖拽释放控制，而且能够让你通过使用样式和模板来自定义显示模式。范例有简单的列表，栅格视图，分组栅格视图，翻转视图以及Semantic缩放（内容会随着UI的放大缩小而改变显示的内容）。

**内部控件**：这些控件可从windows的JavaScript库中找到，他们也提供了非CSS3盒状控件的限制，这样你可以在界面设计中提供更多的扩展性能或者融合你个人的品牌到你的顾客体验中去。

## 用带有适应性布局来创造沉浸感的用户界面

Windows 8 为你提供了多种创意性的选择方案可以动态的适应应用中的变化：调整屏幕显示的大小，改变方向，和用CSS3营造多种不同的显示特性。这些特征和功能让你开发的Metro风格的应用为客户提供一种流畅的，更自然的使用体验。以下是一些示例。

**动画**：用HTML5和CSS3实体化Metro风格，并创建流畅的，动态体验及元素。还可以利用综合的生动且独特，又被用户所熟悉的预定义动画。

**三维转换**：添加流畅的视觉体验，如在打开和关闭界面上带有透视变化及翻转元素。在以往，用户必须用本地代码来实现这些效果，但现在你只需要运用HTML5和CSS3就可以创建。

**灵活多变的盒状布局**：创建灵活多变的各类容器，它们能够按比例填满任何多余的由HTML5布局的空间。这样更有利于设计师创建关键的应用控件，如工具栏或者和导航元素。

**栅格布局**：在栅格结构上的单元中定位并设定内容元素的大小，以这样的方式定义出固定大小，小分块的结构或者自动单元。

**多列布局**：通过多个平行等宽（高）的HTML5内容列来模仿报纸和杂志布局。

## 图形

Windows 8 蕴含了DirectX的游戏开发能力。这样你就拥有了多重创建Metro风格的游戏应用的选择。你可以写出全屏的，流畅的非Chrome游戏，用HTML5，Javascript以及CSS3写出不闪烁的动作。

又比如，用C#和Visual Basic控制XAML写出华丽的2D以及轻便的3D的游戏。你也可用C++及HLSL配合DirectX 11.1充分利用现代图形硬件的处理能力来开发游戏。

### 使用DirectX创建沉浸式的游戏

新的windows 8对图形栈进行了更好的融合，让Direct2D, Direct3D和DirectComputer组件能够很好地同时使用而且不像以前那样需要很多的复制资源。诸如DirectmXMath, XAudio2和XInput 这些以往只在XNA中的功能，现在也已经能够使用了。为了最大地优化视频和游戏体验，你可以使用DirectX 11.1为你的应用带来三维环绕立体的体验

## 媒体和传感器

充分利用现代硬件设备建立沉浸式体验对Windows8来说并不难。你可以使用支持多种主流的编码解码的声明式API来整合音频和视频内容。传感器的支持可以让你使用强大的Win32 API或者非常易用的WinRT组件来建立适应于用户使用情景的体验。

### 整合音频和视频

Windows中Metro风格的应用可以通过JavaScript来调用<Audio>和<Video>两个标签，这样你可以把一小段音频或者视频植入到一条HTML语句中。同样的，XAML开发者可以调用<MediaElement>标签，这是一个强大的媒体整合控制标签。一系列的编码 / 解码都得到支持，如：h.263, h.264, AAC以及MP3等。强化的媒体基础为桌面应用开发人员改进了兼容性和功能上的表现。

Metro风格的应用在WinRT中同样可以获取Windows.Media的命名空间，用于捕获和转译媒体，或使用“播放至”（PlayTo）来实现转译媒体为流音频，视频，或者图片到到联网的设备上。通过照相机捕捉支持，你可以创建捕获控制，通过一个预览窗口，定义用于媒体捕捉的解析度和编解码等等控制，而无需调用各种复杂的API。对于应用来说，只需要一个基础的照相机捕捉而不需要其他的控制，CameraCaptureUI类只需要一个简单的调用方法就能够捕捉一张照片或者视频，而Windows为这个调用方法提供了对话框，你无需为捕捉操作另自行创建任何界面。

### 利用传感器为用户量身定制各类体验

Windows8的传感平台是一个端对端的解决方案。它为各个硬件生产商，类驱动的开发人员，非Win32 API的开发者，以及Windows Runtime 组件的开发者提供了指导手册。

Win32的API可以让你无需写大段代码而整合从传感器读取的数据到你的应用程序上。你能够“发现”可利用的传感器，配准它们并让它们接收事件提醒，以及控制硬件所需发出提醒事件的时间间隔。通过API所支持的传感器的种类几乎是没有限制的。只要是传感数据是通过人机界面设备类驱动发布的，你就能控制使用这些传感器。

WinRT组件能让你快速获取七大传感器的输入：加速器，倾斜计，陀螺测试仪，罗盘仪，环境光，定位仪以及简单的方向定位设备（不包含地理位置的定位）。定位传感器融合一系列物理传感器的数据来提供四元数和旋转矩阵做为输出。游戏开发者能够使用这些组件在多种游戏风格中来控制用户的视角和方向。

## 打造量身定制的网络

Windows 8是一个建立应用的非常好的平台。它也是一个建立美观，快速及高安全网站的领先平台。充分利用了Skydive, Hotmail和Windows Live Messenger的功能，网络服务带来了增强的用户体验。通知机制让你的应用持续的增加价值，甚至当这些应用没有运行时同样如此。

### 面向Internet Explore10而开发

Internet Explore 10是微软为开发者提供的最让人激动的浏览器。通过支持现代的各类标准，如HTML5, CSS3,以及SVG，也包括硬件加速图形和编译的JavaScript，Internet Explore为你建立了下一代的网络体验。

Internet Explore 10持续挑战网络不如本地应用的理念，把硬件加速的HTML5置于核心位置，让开发者为下一代的网络体验创造了可能性。

微软继续履行着IE 10支持HTML和Windows8的承诺，提供一个更强大的平台来推动网络向前发展。Internet Explore提供了基于标准的网络平台，能完全在桌面和新的Metro风格的IE 10浏览器中共同操作，让开发人员能够写相同的HTML5的代码同时拥有两种体验。

IE10的发布，开发人员能够得到更多HTML5方面的支持，让它们能够创建丰富的视觉效果和硬件加速的CSS变革后的动态效果，动画，3D变形，带有CSS3的文字阴影，渐变和SVG滤镜特效。由于对CSS3栅格，弹性盒状布局，多列，figure标签，CSS3区域和文字截断等方面的支持，以及HTML5的表单和输入控制，让创建日益复杂的页面布局也为可能。

IE10的平台同样提供了本地存储方面的支持，如IndexedDB, 应用缓存，异步加载代码，多线程解决方案Web Workers, 严格模式代码支持，Web Sockets网页应用程序中客户端和同服务器端之间的非HTTP通信机制，HTML5 文件API, 拖拽释放功能，历史，以及沙盒模型。这些特性及功能提供开发者、一种更加丰富，更贴近标准，更多程序模型，用来创建现代的网络体验。

微软对HTML5的投入远远超过只是浏览器之于Windows系统的投入。通过使用JavaScript引入Windows Metro风格的应用，使用HTML5和JavaScript建立起来的令人惊奇的网站可以通过网络开发者利用Windows所蕴含的各种能力，快速而简单的转换成为更加丰富的应用软件。通过熟悉的HTML5编码模型和工具组，开发者们能够的开始建立更加漂亮的互动网站和拥有更好性能和更高安全的Metro风格的应用。有了Internet Explore 10和Windows, HTML5将持续为开发者们打开重新构想网络的窗口。

Live Connect提供了一系列控件和API可以让应用程序整合各个微软的连接账户为单独的登录账户（SSO），让用户能够访问SkyDrive, Hotmail和Messenger上的信息。应用可以识别用户并藉由SSO个性化体验，提供访问用户的联系人和Hotmail日历，或者上传照片，文档以及其他内容到SkyDrive上。

### Windows 推送提醒服务

Windows 8中的应用是随着活动和不断更新的内容共同呼吸着的，即便是这些应用并没有在运行。有了Windows推送提醒服务（WNS），应用就能接收来自于你的网站的安全信息，并把它们发送到你应用上的活动磁贴或者生成一条提醒给用户。例如，你能使用WNS在多人在线游戏中通知用户轮到他们了。或者创建一个餐厅点评应用，它能够显示关于最新开张的人气餐厅的通知或者提醒你有人在你最近一条评论中做了回复。你的应用无需打开才能持续添加数据，这样做同样增强了你的客户设备的电池寿命。你可以利用WNS服务所提供的新功能来使用Windows应用商店，自动为你的应用提供ID并更加容易的给你的应用发送提醒和通知。

### 联系人选择器

联系人选择器集成了直接从Windows系统中选择联系人的功能。这就让任何一个应用都能参与并成为一个联系人地提供者，于是其他应用也能够读取存在你的应用中的联系人数据。用户可以选择是否让一个应用分享这些数据或者自己控制这些数据。统一的联系人数据能够使更好的应用被创建，从而让你的用户能够更加容易的找到他们最在乎的人，和他们一起生活娱乐，交谈并联络感情。

## 使用最现代流行的工具和架构进行开发

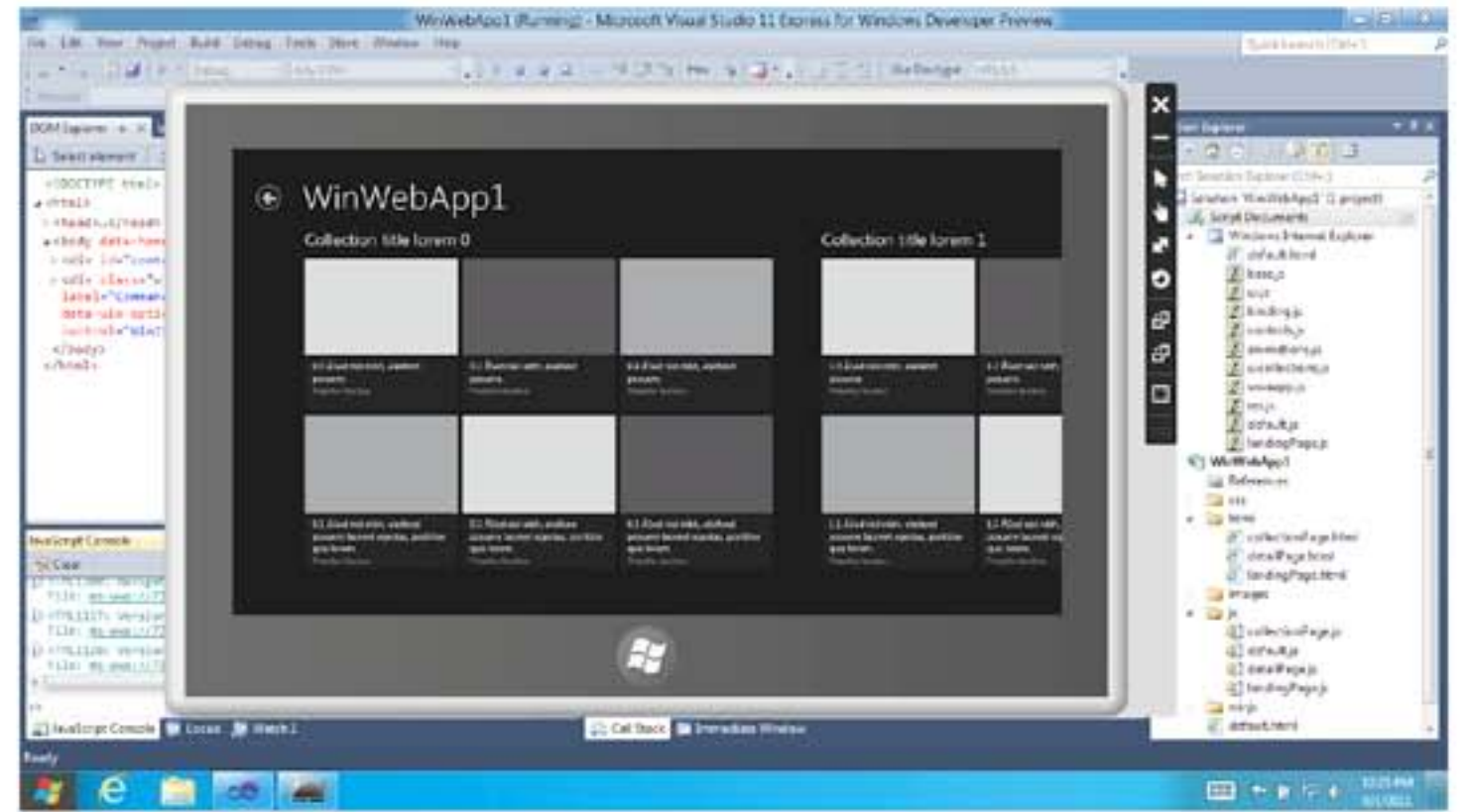
新的Windows 8 SDK和Windows Dev Center 提供了你所需要的一切来创建你的应用。Windows 8 SDK中含有免费版的Microsoft Visual Studio 2011 Express 和Microsoft Expression Blend, 同样也包含了最新的工具，API, 编译器，调试器，示范应用，文档，模版，教程和专家指导。使用SDK工具可以得到完整的端对端的解决方案。

当你在开发应用同应用，操作系统，硬件之间的复杂交互时，Metro风格应用的SDK还另外提供了用以支持这些应用开发场景的功能。除此之外，在评估和配置工具组中（ADK）的工具测量系统性能来帮助确认你开发的硬件和软件都有最好的品质。

### Visual Studio 2011, Express 和Expression Blend在Windows 8系统下完美运行

Microsoft Visual Studio 2011 Express是强大的Visual Studio整合开发环境（IDE）免费轻量级版本 Visual Studio Express同时打包了Expression Blend, 这是一个视觉设计工具，可以帮助你使用HTML5和XAML。Expression Blend即能让你使用Windows控件又是一个勾勒用户界面的整合工具。为了帮助你用DirectX和HLSL来编写游戏，Visual Studio Express有了一个新的图形创作和调适IDE。

如果你开发设备驱动，Visual Studio Express现在集成了一整套的工具和界面帮助你进行编写，创建，编译，签名认证，测试，调适和确认驱动，同时还配有工具来移植你现有的驱动文件到的Visual Studio Express项目和解决方案的文件中。



### Visual Studio Express帮助你开发Metro风格的应用

#### 用户体验数据

Windows Dev Center提供了一个仪表盘窗口，里面包含了很多通过观察记录成功应用得来的改进你应用的方法。具体可以查看下载量，受益，使用情况，应用交易，用户打分，市场趋势，崩溃和死机数据等多种报告。

Windows 8的开发者能够得到这些报告和展示基于失败事件和以触发原因为中心的用户体验数据，这些数据则显示了他们的产品在整个市场上的可靠性如何，多久就会发生崩溃和当机的事件，在整个系统生态中（公司和个人信息不会泄露）和其他应用相比是怎么样。这些新的报告统一且一致并可行动参考，这样你可以快速有效的找到问题的所在。

## 创建于广泛用户基础的Windows应用商店之中

Windows 8 和Windows应用商店让亿万用户轻松的找到，试用和购买有用的，高质量的来自于世界各地的应用。Windows8的生态系统创建了一个容易的散布，升级还有通过你开发的应用获取酬劳的环境。



## 最大化你的收入

通过Windows应用商店的营收分享方法，你的应用从最开始就拥有带来更强收益的潜力，而且之后可以通过你提供的升级，新功能和应用内产生的购买行为带来更多的价值。当你在Windows应用商店中提交了一个应用，你就可以特别说明该应用是免费版本，或者一个完整的应用的试用版本，或者是由广告赞助的免费版本，或者是付费版本。

### 使用应用内购买 ( In APP Purchase ) 选项

有了应用内购买选择，你可以设定你的客户购买特定内容，或者新的功能或者游戏中锁定的关卡。你可以放置购买选项，并设定当地货币单位，把他们安排在你的应用中最合适客户的位置。

### 在世界各地贩售

在Windows应用商店的提交系统中，你可以选择特定的国家或者区域来贩售你的应用，而且你且只能给你的应用设置唯一的一个价格，这个价格会被转换成其他国家的货币单位。

### 一条贩售应用更好的途径

Windows 8和Windows应用商店提供了一个灵活的途径来帮助你的应用推向市场。商店用户通过一个应用的政策简明和直接的。应用商店能够在必要时移除应用或者提供多种途径给顾客来发现并安装你的应用。Windows 应用商店同时拥有多种工具来检测购买率，盈利和应用质量。

# 技术基础, 设备,及安全

Windows 8是构建于windows 7坚实的基础上，拥有诸多在功能，安全，隐私和系统稳定性方面的改良。而那些让消费者，开发人员以及IT领域的专业人士所喜爱Windows7的一切都予以保留，甚至更加优秀。对内核的改良改进了系统的响应，安全及性能。为驱动开发所进行的驱动模块和工具组链的改进改善了系统的稳定性和可靠性。因此Windows可以运行在ARM的设备（主要指手机和平板电脑）上，也包括X86和X64的设备。你将从这些革新的安全功能上受益，而你的应用在Windows8系统上会更快的运行。

## 技术基础 系统要求

Windows8 能在支持Windows Vista和Windows 7的相同硬件设备上表现优异。

1 千兆赫或者更快的32位处理器

1 GB 的内存(RAM) (32位) 或者 / 2GB ( 64位 )

16 GB的可用硬盘空间 ( 32位 ) / 20GB(64位)

支持Direct X 9的图形设备 带有WDDM (Windows显示驱动模型) 1.0 或以上的驱动

当然，要使用触摸输入的话还需要一块支持多点触控的屏幕。

## 在ARM, X86及X64架构下运行的应用

除了运行在X86 和X64的设备上，windows 8在ARM设备上表现出更加优秀的扩展能力。按照你所使用的开发技术，只需一次写好你的Metro 风格的应用，它们就能够在任何一个所支持的架构下运行。而你同样能够利用你的客户的个人终端的特有性能。

# Windows 8 与商业

为了帮助公司能够释放他们全部的工作潜能。Windows 8把用户放在每一块体验的中心。就是说，当保证同人和信息间个性化和无缝连接的同时，也能够通过植入反间谍，强认证和数据编码的方式来保护设备同数据库之间流动敏感数据。

Windows 8系统运行于当前的硬件设备。通过融合了一系列简化兼容性测试和开发的工具，Windows 8能够在你当前的客户端管理架构上，包括顺应性和安全流程等方面运作。它能帮助专业的IT人员让他们的系统运行的更加安全流畅。

## 各地漫游及联通

有了Windows 8，商业用户能够无缝连接到到各种各样的内容，设备，以及工作当中的重要伙伴。有了Windows To Go技术（Windows To Go，是Windows 8 的一项新功能，通过这一功能，用户可以在USB存储设备（比如U盘）中运行带有企业应用、数据和设置的Windows 8。）用户可以不用随身携带电脑和移动设备就能拥有增值的工作漫游和连续性的体验。一个通过编码的USB设备，能够安全的启动用户的工作桌面，并任何地方任意一台装有Windows 8的联网的电脑来获取公司的资源。

DirectAccess能够内网的远程用户安全的获取各类资源，如SharePoint站点，一些列的商业应用工具，分享的文件等，这些都无需让用户自动启动连接会公司网络。Windows 8同样运用可计量的连接服务来提醒用户关于付费数据的使用情况，来优化用户和各类应用所耗费的成本及带宽。

企业IT专业人员可以提供给用户一个带有Windows8副本的可启动的USB存储设备，其中包含有行业所需应用，数据和设置等。当用户完成工作并注销后，他们只需简单的移除这个USB设备，而不会遗留下任何数据或者信息的痕迹。

### DirectAccess

Windows 8对DirectAccess做了一系列的重要的改进，重点集中在简化网络架构的需求，降低开发成本以及提高可管理的能力。在Windows8中的DirectAccess更好的支持了现有的架构，比如non-IPv6的服务器资源以及简化小型组织的部署而无需另行构建PKI架构（公钥加密和数字签名服务的系统）。

### 灵活地获取文件信息

IT专业人员能够创建和维护一种动态的，按照一定结构的文件，文件夹，以及共享文档的访问政策。这些都是基于可配置的条件来建立的，比如用户角色，所在部门，国家，或者地区，被访问数据的敏感程度，访问该数据所用设备的健康程度等。

## 数据安全和管理的

各行业能够更好的保护行业数据，从数据创建开始，到存储和访问的生命周期都得到保护。运行Windows8的个人电脑，无需用户的操作，值得信赖的启动程序能够自动修护驱动和强化政策，从而让个人电脑能够更好的防护间谍软件。IT专业人员们也可以要求新的应用在严格限制的环境下运行，从而减少用户当机的时间以及敏感数据的丢失。

数据加密同样得到了改进。Windows8启用了一种始终运行的数据加密机制，这种机制运行于后台，即使是当设备丢失的时候其存储的数据都是安全的。Windows8提供了一组非常简便的工具给IT专业人员来确认或者禁止在商业用的电脑上运行的某些应用程序。

Windows8简化了访问控制管理来帮助确保在公司架构下不同的人能够访问到恰当和合适的的数据。

### 应用锁

应用锁包含了新的功能从而能够同时管理桌面和Metro Style的应用程序。IT专业人员能够穿件安全政策来允许或者拒绝特定人或人群触发的特定应用程序。

### 比特锁

Windows 8对比特锁提供了一系列丰富的改进，从而帮助行业减少非授权的数据访问带来的风险。对于Windows8的行业用户来说，比特锁比当前仅用来支持快速，硬件加密驱动等软件加密方式更好的性能。比特锁经对有数据的分区进行加密编码，然后随着空闲空间的使用而逐加密。通过清除键来对硬件驱动进行预编译及标准化的引脚复位让用户体验得到了改善。

## 测量启动

在安全启动之外，Windows进一步对启动程序进行确认。开始进程现在则经过了数字签名，保护和测量，然后被存贮在TPM芯片中来防止恶意程序工具包和间谍软件的感染。对基于TPM的系统而言，在启动的进程中，Windows8 将展开一些列综合性的测量链任务，被成为测量启动，它主要用来对启动进程进行确认，并防护恶意程序工具包和其他间谍软件的侵害。

### 改善的活动目录激活

Windows8能够在活动目录里用户身份对使用Windows系统的电脑进行激活，而无需另外系统设施。Windows服务器同时也包含了新的容积激活角色来快速有效的配置服务器，并把它作为关键管理服务器主机来同时激活Window 7 和 Windows8的客户端。

活动目录激活让Windows8系统在企业环境下被激活：通过使用当前活动目录的执行结果和当前IT专员们所用于认证的同类Windows产品激活技术。

### 客户端 Hyper-V

Microsoft® Client Hyper-V®是一个灵活可变的，强有力的以及高性能的客户端虚拟技术。它让IT专员和开发人员可以在他们的Windows8的电脑上同时运行多个操作系统的事件。通过整合硬件来降低成本并同时通过Hyper-V 服务器上的虚拟机的兼容性改进了效率。

### 流畅性部署

Windows8提供了一系列简便易用且强大的工具用来简化图形创建，测试，部署和激活。Windows 8让远程安装和部署变得非常容易并且高度安全。IT专业人员能让计算机在部署的同时进行预加密，并让标准用户在第一次使用的时候创建脚注。他们同时也能够在无需另行建立基础架构，仅通过目录服务中的用户身份来保证计算机容积激活的安全。

IT专业人员也能够对所安装软件的影响进行图形分析来增强系统图像的表现力。植入式的，带有高性能的客户端虚拟技术让虚拟化向前跃进了一大步：IT专业人员们可以在同一台机器上进行开发，测试以及运行多个操作系统。

最后，IT专业人员现在能够在单一的图形中进行多语言部署，让面向全球发布的内容更加容易创建和管理。

# 什么是Metro 风格的应用？

Metro 风格的应用就是为用户需求，为之所运行的设备，为触摸互动以及Windows 用户界面量身订制的。Windows系统帮助你同你的用户之间互动，帮助用户同你的应用互动。阅读这篇文章能够了解Metro风格的应用以及开发上好应用的各项基础。

## 为你的用户量身订做的应用

在使用Metro风格的应用时，用户能够体验到：

**沉浸感** Metro风格的应用会填充至整个屏幕，让用户充分沉浸在你营造的体验之中。在全屏显示的应用中，用户能够完全被他们做喜爱的内容所吸引。我们让操作系统的界面为你的内容腾出空间，你的应用将会成为整个Windows体验的中心。充分利用这些空间，但不要过度堆积。让你的用户专注在他们想要做的事情上。

**融入感和生动感** Metro风格的应用会通过用户所感兴趣的内容或在意的人来抓住他们的注意力。动态的磁贴更新可以让用户轻轻一瞥便能被你的应用所吸引。开始界面让你的应用得以充分施展所能。应用磁贴的活力来自于状态和动作的更新，这样做能够鼓励用户融入到你的应用中去。当你在设计你的应用磁贴时，你需要：

- 高亮你的品牌。应用磁贴是用视觉化的方式来定义你的品牌和用户群的机会，它应该非常独特且有吸引力。
- 展示你的用户最感兴趣的信息和动作。你得让你的用户不断的回访你的磁贴，查看更新，并点击进入。你要用那些更新来把用户拉回到应用本身。你对那些需要被展示的信息和行为考虑的越深入，用户就越有可能融入进来。

想要查阅更多关于设计和创建应用磁贴的信息，请见磁贴参考和一览表 以及 提醒参考和一览表

**有联系的** 你的用户可以在他们的社会圈子里分享内容，或发送内容到所连接的设备上。更多内容见查找，分享和联接

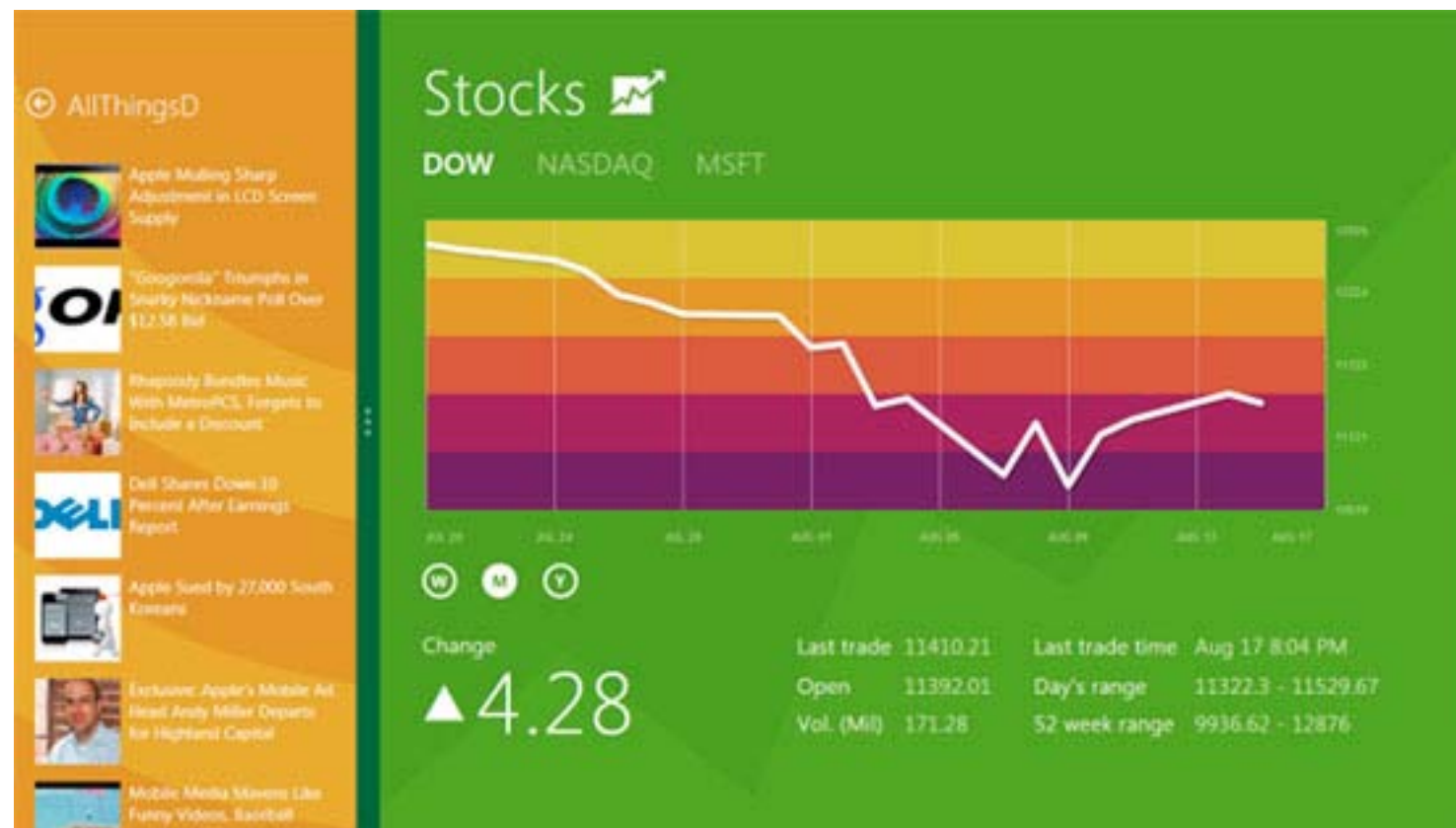
**互动和触摸优先** 你若把应用为触摸和物理控制设计，Windows会免费为你提供键盘，鼠标和图形缩放的支持。更多请查看为触摸而设计以及响应用户的互动。

**更棒的多视图和多外观设置** 应用可使用支持“流畅性”布局的控件，这样内容可以自然流动并根据用户所选择的设备进行自动调试空间。此平台还提供了自动缩放功能，从而在多分辨率和多屏幕方向的情景下设计更加容易。更多请见为不同的外观设置而设计。

**更鼓舞人心** 你的用户能够安全的下载，安装，使用，更新或者卸载你的应用，而无需对Windows或其他应用做任何不可逆的改动。

## 为用户的多任务而设计

尽管在全屏环境下，你的用户完全沉浸在应用中，Windows任然能够为用户提供了多任务的可能



• **辅屏视图** 你的用户喜欢一次做几件事情！他们想要在使用你的应用时同时和他人聊天，上网，看电影或者整理他们的个人税务。所以当贴靠或者取消贴靠视图时，让你的辅屏视图更有用并始终维持整体情境。

• **填充视图** 你的用户可能把其他应用贴靠在辅屏视图中，而把你的应用安排在填充视图中。要考虑到水平方向减小的空间来调整应用来适应该视图。

## 为你量身订做的开发者体验

开发者也同样获益良多：

**你选择的开发语言** 你的使用让你最习惯的开发语言来创建你的应用：JavaScript搭配HTML5 和CSS3 ，或者C++/C#/VB的XAML，了解更多Windows Runtime和你可以选择的开发语言，请参见Windows Runtime .

**新的部署方法** 你的用户可以安全而简便的安装你的应用。 移除应用也如此，而且不会随着时间的推移而降低系统的运行效率。

• **Metro风格的应用封装** 封装你的应用成为一个Metro 风格的应用包用于部署。所有Metro风格的应用都是通过每个用户单位来安装的。他们可以安装，更新以及无痕卸载；而且他们也不会对操作系统的造成不可逆的设置或改动。

## • Windows应用商店 (未发布)

**新的界面和触摸为中心的输入方法** 运用那些为达成用户目标的控件来创建你的应用。这样做可以支持用户直接对界面元素进行控制。

**本地化** 一旦成功开发好了应用，你就可以把它们用多种语言部署在多个地方。不仅如此，Windows 开发者预览 (WDP) 提供了一个API，能够让你的应用根据用户的语言和文化偏好产生响应—如日期，时间，数字，以及货币单位等—这对你来说几乎不用费什么功夫。而这样做的最终结果就是能够提供给各个地方的用户如同量身订做般的同其文化保持一致的体验。

**开发者工具** 使用免费版本的Microsoft Expression Blend和为Windows开发者预览 (WDP) 而做的Microsoft Visual Studio 11 Express，你就能够获得一个帮助你设计，开发，封装，测试，以及部署应用的开发环境和各类模板。

Build date: 9/16/2011

# 设计Metro风格的应用

当你开始规划Metro风格应用时，多思考你想带给用户怎样的体验而不是你想包含怎样的功能特性。如果你还没有开始，你也许希望了解更多关于Metro风格应用的内容在什么是Metro风格应用。



## 1. 决定你的用户体验目标

比如你想做一个照片应用。那么，你会思考用户处理，保存，分享他们的照片的原因，然后你意识到他们想要安全的保管照片，并通过照片和他人建立联系，重现记忆。这些就是你希望这个应用最擅长的事情，你将在剩下的设计过程中用这些体验目标来指导你。

常用方法：头脑风暴，关联图，思维导图

**你的应用是怎样的？** 从一个宽泛的概念开始并列所有你希望你的应用可以帮用户做的事情。比如，你想做一个帮人们规划旅行的应用。以下是一些你可能写在餐巾纸背面的点子：

- 获取一次行程中所有地点的地图并随声携带
- 当你在某一城市时，找出当地正在发生的特殊事件
- 让旅友们创建并分享自己的必看景点和必参加活动列表
- 让部分驴友们编辑整理自己的所有照片并与家人朋友分享
- 根据机票价格获取推荐目的地
- 找到一份关于我目的地附近的饭店，商场和活动的统一列表

**你的应用擅长什么？** 后退一步看看之前列出的点子，有没有哪个场景能即刻引起你的注意。试着去裁剪那份列表直到只剩下一个你最希望专注的场景。在这一过程中，你可能会划掉许多好点子，但是对他们说“不”对于把一个场景做到极致是至关重要的。

在你选择了一个场景之后，接着决定该怎样只用一句话解释把他给普通人。例如

- 我的应用擅长帮助朋友们为集体出行创建行程计划。



- 我的锻炼应用善于帮助人们跟踪记录自己的锻炼进度并和朋友们互相分享成果。
- 我的百货应用善于帮助家庭协调管理每周的杂货采购，这样他们就再不会忘记买某样东西或者买同样的东西。

这就是你给应用做出的“擅长”声明，在你创建自己的应用过程中，他能引导你权衡设计，做出决策。关注于那些你想要带来亮点的用户场景，并小心不要把它变成一份功能列表。他应该是关于你的用户能做什么，而不是你的应用能做什么。

## 2. 决定你希望带给用户怎样的活动或流程

用户为了达到他们的目标而与你的应用产生的一整套相关交互就是一个流程。每个流程都应该与你的“擅长”声明紧密关联，并帮助他们达成你想要的用户场景。优秀应用的流程应该是需要最少的交互并且易于学习的。

方法：

- 概述流程：先做什么事，接下来是什么？
- 流程故事版：用户怎样通过一个个界面去完成整个流程？
- 原型：用快速原型来测试流程。

**用户能做什么？** 例如，擅长帮助朋友们为集体出行创建行程计划的旅行应用。我们来列出关键的流程：

- 创建一个包含基本信息的行程
- 邀请朋友加入行程
- 加入朋友的行程
- 查看其他旅行者推荐的日程安排
- 在行程中添加目的地和活动
- 编辑和评论朋友添加的目的地和活动
- 分享旅程以便于家人朋友关注

### 3. 决定如何展示你的应用

就是要确保基本元素的正确。既然你已经知道你的应用擅长做什么并且你想出了实现他的流程，那么你可以开始考虑软件的基石了。



**应该怎样组织你的界面内容？**大多数应用内容可以被划分为某种群组或层级的形式。在所有内容中，你所选择的最高分组层级应该与你的“擅长”声明的核心相符。

以旅行应用为例，有许多种方式来分组旅程路线。如果应用关注的是发现有趣的目的地，那么你应该根据兴趣来分组，像是冒险，阳光假期，浪漫度假。但是，如果应用的关注点是和朋友们做旅行计划，那么根据社交圈子来组织内容，像是家人，大学同学，同事，才更为合理一些。

选择怎样组织内容会帮助你决定在你的应用中需要怎样的页面或视图。Microsoft Visual Studio 11 Express for Windows Developer Preview中的项目模板提供了可以满足多数内容需要的通用应用布局样式。

我们来看一下“分享旅程以便于家人朋友关注”的流程。我们假设用户已经创建了一个旅程。分享旅行路线需要这样一些步骤：

1. 用户开启应用并看到自己创建的旅程列表。
2. 用户点选她想要分享的旅程。

3. 旅程的详细信息出现在屏幕上。
4. 用户通过某个界面来发起分享。
5. 用户选择或输入了她想要分享的朋友的点子邮件地址或名字。
6. 用户使用某界面完成分享。
7. 应用将她刚才分享的朋友列表更新到了旅程详细信息中。

在这一过程中，你开始明白你需要创造那些界面，以及那些需要考虑的额外细节（像是为还没有使用你的应用的朋友准备一份标准的邮件模板）。你也可以开始去掉那些多余的步骤。例如，也许在开始分享前，用户其实并不需要看到旅程的详细信息。流程越简洁，容易使用。参考选择正确的界面来获取关于怎样使用不同的界面的更多细节。

**如何组织指令？**使用流程步骤的大纲来标示出你需要设计的潜在指令。接着考虑那些指令应该出现在应用的哪些地方。

- **总是尝试利用内容。**一旦可行就应该让用户直接操作应用画布上的内容，而不是添加作用在内容上的指令。例如，在旅行应用中，让用户通过拖放列表中的活动来重新整理行程安排，而不是选中一个活动然后使用上下指令按钮。
- **如果你不能利用内容，在界面上放置指令。**
  - 在应用栏上：你应该把最常用的指令放到应用栏上，在用户呼出他们之前，他们通常会隐藏起来。
  - 在应用画布上：如果用户所处的页面或视图有着单一目的，你应该在画布上直接提供针对那一目的的指令。指令数量应该比较少。
  - 在情景菜单中：对于剪切板动作（例如剪切，复制和粘贴）或者作用于那些不能直接选中的内容上的指令（比如在地图中的一个地点上添加图钉），你可以使用情景菜单。

**决定你应用中的每一个视图的版面。** Windows Developer Preview支持横向视图，纵向视图，分屏视图和填充视图。用户可以在任何时候将你的应用放置于其中之一，而你希望在每一个视图中你的应用看起来用起来都很棒。这意味着你需要为每个视图的界面元素规划布局，并将你的布局实现映射到每一个视图状态中。这么做之后，你的应用就可以流畅的切换来满足用户的需求和偏好。

更多信息，参见视图支持，选择一种布局和分屏，填充模式规范。

## 4. 决定你能使用怎样的应用和窗口特性来实现你的流程设计

你怎样使自己的应用脱颖而出？如何让你的应用在Windows Developer Preview中看起来很棒？

既然你已经知道你的用户想要什么，也知道该怎样帮助他们实现目标，现在你可以看看你工具箱中的各种工具了。探索Windows平台并把功能特性和应用的需求联系起来。确保每一功能都符合用户体验规范。

常用方法：

- 平台调研：找出平台能提供的特性以及你要如何使用他们。
- 关联图：把你的流程和功能关联起来。
- 原型：测试功能以确保他们能满足你的需求。

**应用协议** 你的应用可以加入应用协议以使广泛的跨应用，跨功能用户流程成为可能。

- **搜索** 使用户在系统的任何位置搜索你的应用内容，包括从其他应用内搜索，反之你也可以搜索其他应用的内容。
- **分享** 使用户从你的应用分享内容给其他人，以及接受来自他人的分享内容。
- **播放至** 使用户在他们的家庭网络中享受从你的应用传输到其他设备上的音频，视频或图像。
- **设置** 统一的整合所有设置并使用户通过他们以及熟悉的通常机制配置你的应用。
- **文件选取器和选取器扩展** 使用户通过本地文件系统，连接的存储设备，家庭组甚至其他应用读取和存储文件到你的应用中。你也可以提供一个文件选取器扩展来让其他应用读取你的应用内容。

更多信息，参见 Windows 应用契约或文件选取器。



**不同视图和形式要素** Windows Developer Preview将你的应用带到台前并使用户掌握主导。你希望你的应用界面在任何设备，任何方向，任何用户想要使用他的环境中都熠熠生辉。了解更多关于为据不同形式要素设计。

**触控优先** Windows Developer Preview 提供了一种独特的触控体验而不是简单的模拟鼠标功能。例如，语意缩放就是一种用于在大量内容中进行导航的针对触控优化的操作。用户可以平移或滚动内容分类，然后放大进入那些分类中去查看越来越详尽的信息。相较于传统的导航和设计布局（例如标签），你可以利用这一特性来更触觉化，可视化和信息化的呈现你的内容。当然，你可以利用许多触控交互，比如旋转，平移，轻扫，交叉滑动等。了解更多关于触控特性。

**迷人且新鲜的** 确保你的应用让人感到新鲜并用这些标准体验来吸引用户。

- **动画** 用我们的动画库使你的应用让人感到快速和流畅。帮助用户理解情景的变化并把体验和视觉过度联系起来。了解更多关于赋予界面生气。
- **推送提醒** 使用户通过烤面包提醒来了解时效性内容和个人相关内容，并使他们重回到你的应用中即使应用已被关闭。了解更多关于提醒。
- **内容磁贴** 在“开始”屏幕上推广你应用中的有趣内容和深层链接，并使用户启动应用直接进入特定的页面或视图。
- **应用磁贴** 提供最新或相关更新来吸引用户回到你的应用。下一部分是更多关于这一点的内容。

**个性化**

- **设置** 使用户通过保存应用设置来保留他们想要的体验。统一的整合所有设置并使用户通过他们以及熟悉的通常机制配置你的应用。
- **漫游** 使用漫游和应用生命周期来保持设置和状态，以使你的应用在任何地方都易于使用，不论是家用电脑，工作桌面还是个人平板。
- **用户磁贴** 使你的应用更加个人化。允许用户读取他们的用户图块图像，或者让他们从应用中设置内容作为整个Windows中的个人磁贴。

**设备性能** 确保你的应用可以充分发挥现今设备的性能。

- **接近型手势** 在可能有多名用户聚集的地方（多人游戏）让用户通过自然的轻拍手势来连接设备，提示体验。了解更多关于接近型手势。
- **摄像头** 连接用户的内置或外接摄像头来聊天，会议，录制视频博客，拍摄图片，记录身边的世界或者任何你的应用擅长的事情。
- **加速计和其他传感器** 今天的设备拥有许多传感器。你的应用可以根据环境光来调暗或调亮显示，或是在用户旋转设备时重新调整界面，亦或是对任何的物理运动做出反应。了解更多关于传感器。
- **地理位置** 使用来自网络数据或位置传感器的地理位置信息来帮助用户外出，在地图上定位或是获取关于附近的人，活动和目的地的通知。了解更多关于地理位置。
- **短信息服务**
- **打印**

我们再回到之前的旅行应用案例。为了能很好的帮助朋友们合作为集体出行创建行程计划，有可以利用那些特性中，仅举几个例子：

- 分享：用户们将即将到来的旅行和自己的安排分析到多个社交圈子，让家人朋友也能感受到旅行前的兴奋。
- 分屏和填充视图：屏幕的一边是旅行应用，另一边是浏览器用来研究和预订服务。
- 搜索：用户从其他人的共享或公开旅程中搜索并找到活动或目的地，并把他们添加到自己的旅行中。
- 语义缩放：用户浏览行程的时间线并放大查看更详细的信息。
- 通知：当旅游公司更新了他们的行程安排时用户会得到通知。
- 设置：用户根据自己的偏好配置应用，例如哪个行程要使用通知或是哪个社交圈子被允许搜索自己的行程计划。
- 语义缩放：用户浏览行程的时间线并放大查看更详细的信息。
- 用户图块：用户选择当与朋友进行分享时自己希望显示的图片。

正如你所见的，Windows Developer Preview帮助你创造能带给用户快乐的全新的，吸引人的体验。探索开发Metro风格应用和不同Metro风格应用来获取更多概念。

## 5. 创造美好的第一印象

思考一下，当用户第一次打开你的应用，你要传达给用户的信息是什么。回顾一下你的擅长声明。即使你没有机会直接亲身告诉用户你的应用擅长做什么，你还是能通过第一印象把信息传递给用户。利用以下这些：

**磁贴和启动画面** 图块是应用程序的脸。在开始屏幕上那众多的应用中，怎样让用户想要打开你的应用？确保图块强调了你的应用品牌并展示了应用最擅长做的。利用图块通知让你的应用令人感觉新鲜，把用户一次又一次的带回到你的应用中。

**启动画面** 启动画面是另一个绝佳的品牌时刻，利用它去传达应用的个性。

**第一次启动** 在用户注册你的服务，登陆自己的账号，添加自己的内容之前他们会看到什么？在向用户请求信息之前试着展现出应用的价值。考虑展示样本内容给用户尝试，让他们在犯错前理解应用的目的。

**登录页面** 登陆页是每一次用户启动应用来到的地方。这里的内容应该有明确的焦点，并及时的展示应用所关注的内容。这个页面只要做好一件事，用户会探索应用的其他部分。登陆页要解决的是干扰，而不是发现性。

## 6. 根据规范和要求验证你的设计

在你深入到实际开发之前，你应当验证你的设计或原型来避免之后的重做。每个特性都有一套用户体验规范来帮助你打磨你的应用，以及一套你需要满足的商店要求来在Windows商店贩卖你的应用（目前不可用）。你可以使用 Windows App Certification Kit测试商店要求的技术符合性。

利用设计清单使你关注在重要的特性上。

Build date: 9/15/2011

## Designing for different form factors

# 为不同尺寸的设备而设计

备注：form factor是个短语, 设备外观设计配置，通常指尺寸大小。

本文系根据个人理解进行翻译，并在部分内容上有所改动，如有疑问请与我联系。

RTX：jrlin 或 mail：limjr@qq.com

[This documentation is preliminary and is subject to change.]

[此文并非正式文档，后期有调整的可能。]

Metro style apps are at home on desktops, laptops, and slates. Your users can move between different devices, change the screen orientation, or shut everything off and on, and your Metro style app moves, changes, and reacts with them.

Metro风格的App会被应用在家中的台式电脑、笔记本以及平板电脑上，用户在多个不同的设备间切换、改变屏幕的方向、或者关闭/打开任何其他程序，Metro风格的App都会随之移动、改变从而适应这些变化。

## Touch, mouse, and keyboard input

### 触摸、鼠标及键盘输入

When you design your app for touch input and manipulation, you get support for mouse and keyboard input for free. Your users can switch from one put method to another and not miss a beat of the app experience. Plug a keyboard into a slate? No problem. Your app adapts to your users' choices. See Responding to user interaction for more information.

当你的App是按触摸互动设计了，就已经默认支持鼠标和键盘了。这样，用户可以在不同的输入方式之间切换，而不会因为输入方式影响该应用的应有体验。想在平板电脑中加上一个键盘？没问题，你所设计的应用应当支持用户选择。查看更多关于“用户交互的反馈”的信息。

## Device capabilities

### 设备的性能

Great apps take full advantage of the devices they run on. Windows Developer Preview has built-in support for these device capabilities:

好的应用应当完全发挥所在设备的优势，Windows开发者预览版本已经内置了对这些设备能力的支持。

**Accelerometers and other sensors** Devices come with a number of sensors nowadays. Your app can dim or brighten the display based on ambient light, or re-flow the UI if the user turns the display, or react to any physical movement. Learn more about sensors.

**加速器及其他传感设备** 当前的设备已经配备了许多传感设备。你的应用可以根据当前环境的光线强弱而调整自身显示屏的明暗，可以在用户转动设备时调整界面流布局方向，可以针对其他物理的变化做出反应。学习更多关于传感器

**Geolocation** Use geolocation information from standard web data or from geolocation sensors to help your users get around, locate their position on a map, or get notices about nearby people, activities, destinations. Learn more about geolocation.

**地理位置** 通过标准web数据或地理位置传感器所获得的地理位置信息，可以帮助用户：了解周围的情况、在地图上定位、获得附近人/活动/目的地的信息。学习更多关于地理位置

**Cameras** Connect your users to their built-in or plugged in cameras for chatting and conferencing, recording vlogs, taking profile pics, documenting the world around them, or whatever activity your app is great at.

**相机** 用户通过设备上自带（或外接）的相机来聊天、视频会议、记录视频博客、拍摄资料照片、记录他们周围的世界，以及其他任何你的App可以支持的事情。

**Proximity gestures** Let your users connect devices with the human gesture of a 'tap' to light up experiences where you expect multiple users to be physically nearby (multi-player games). Learn more about proximity gestures.

**邻近手势** 当有多个用户设备在邻近范围内，可让用户通过简单手势激活体验。（例如，通过“轻拍/轻敲/轻叩”等人体手势将邻近的多台设备建立连接，从而达到多人游戏等目的）。了解更多关于邻近手势

When planning your app's features, consider the devices your app might run on. Are some device capabilities mandatory for your app to work well? Or can you get by without some? You have to declare which capabilities your app supports in your app manifest, but in the app itself, you can create fallbacks for optional capabilities. For example, let's say a travel mapping app lets users track their travels on a map, tag sites, include journal comments, send to social media, and add photos or videos from their trip. Geolocation would be a mandatory capability, but camera support could be optional. If the device doesn't have a camera, users could upload videos or photos taken on another device.

筹划App的功能时，你需要考虑App的运行设备：是否设备的某些性能是你的App正常运行的必要条件？或者说如果没有某些性能，你的应用会怎么样？你必须声明App的功能列表中哪些功能是需要设备性能支持的。性能支持分为：必要性能、可选性能。针对可选性能，当设备无法提供可选性能时，你的App可以创建备选方案。例如，有一款旅行地图的App，可以让用户在地图上追踪旅行的轨迹、标记地点、记录旅途点评等，然后结合图片和视频到发布到社交媒体当中。地理定位是该App的一个必要设备性能，相机是可选性能的。如果设备不支持相机，则用户可以选择上传该设备中存储的视频和照片。

Great apps cover all their bases.

一款优秀的应用，应该适配其所运行设备的所有情况。

## Fluid, multiple views of your app

### 流动布局，多样的应用视图

Windows Developer Preview puts users in charge. You want your app UI to shine on any device, in any orientation, in whatever circumstance the user decides to use it. When users change the orientation of their monitor or mobile device, your app gracefully re-flows in response. When you design your app UI with fluid views, you get this behavior for free. Windows takes care of the rest.

Windows开发者预览版 将所有的控制权都交给了用户。不论用户在任何环境中、在任何设备上、以任何方向使用你的App，你App的UI都必须出色地适配。当用户调整显示器或移动设备的方向时，你App的UI界面应该优雅流畅地完成对应的一系列变化/切换。当你App的UI是按照流式布局的方式设计的，那以上提及的完美适配、流畅变化都会自动完成，因为Windows会为剩下的事情做好处理。

#### Landscape 横向视图

Design for landscape first so your app looks good on all form factors.  
首先设计App的横向视图，使其在任何尺寸的设备上看起来都是没有问题的。

#### Portrait 纵向视图

But remember some devices rotate! So optimize the layout of your content when in portrait view and retain functionality whenever possible.

但是，请记住一些设备是会旋转的！所以，请尽量优化在纵向视图下App内容的布局，并尽可能的保持功能完整性（尽可能的保留功能）。

The key to looking good in these views (as well as in snapped, fill, and fullscreen views) is defining layouts for the app in each view. When you plan ahead for each view, your app UI reflows automatically when a different view is triggered on the device. See [Supporting multiple views](#) and [Choosing a layout](#) for more information.

在各种视图模式下（包括分屏、填充以及全屏）都能良好展示的关键是：针对每个视图模式定义App的布局设计。当设备触发不同视图模式切换时，你App的UI也将自动切换至对应的布局设计。查看关于支持多样视图和选择一个布局的更多信息。



## Built-in graphical scaling

### 内置的图形缩放

If users can access your app on multiple form factors, does that mean you have to design a different UI for every potential screen size Windows works on? That's a lot of different screen sizes! The answer is, not necessarily. Built-in scaling means your app and content always look great, whether on a small 7" device or on a big 30" monitor. You just need to use a fluid layout and make sure the graphics in your app look good when scaled. For more info, see [Guidelines for scaling](#).

如果用户可以在各种尺寸的设备上运行你的App，是否意味着你需要为每一个可能的屏幕尺寸设计不同的UI界面？事实上存在大量不同尺寸的屏幕尺寸！答案是没必要。内置缩放意味着无论在小型7寸的设备或者大型的30寸的显示器中，你的App和内容总是可以完美展现，前提是你的App使用流视图布局设计并确保在你App中所使用的图片经过缩放依然可以正常查看。了解更多内容，查看[缩放指南](#)。

## Roaming data

### 数据漫游

What if your users go from their work desktop to their slate at home? Their files, app state, and app preferences go home with them. They can pick up right where they left off, across different machines and user sessions. Learn more about roaming and [Managing the application lifecycle](#).

如果用户从办公室的台式机转移到家中的平板电脑，会如何？用户的文件、App状态以及App的偏好设置都会自动同步到家中的平板电脑，用户可以通过任何设备（不同的硬件及软件环境）继续之前的操作。了解更多关于漫游和管理应用生命周期的内容。

Designing for touch

# 为触摸而设计

本文系根据个人理解进行翻译，并在部分内容上有所改动，如有疑问请与我联系。  
RTX : jrlin 或 mail : limjr@qq.com

[This documentation is preliminary and is subject to change.]

[此文并非正式文档，后期有调整的可能。]

This topic describes the new touch interactions for Windows Developer Preview and provides high-level guidelines for adding touch support to custom UI.

本文介绍关于Windows开发者预览版本中出现的新触摸交互方式，并为自定义UI支持触摸交互提供了概括性指南。

## Standard touch interactions

### 触摸交互标准

The following describes the standard touch-related terms used in Windows Developer Preview.

以下介绍了在Windows开发者预览版中的标准触摸交互术语。

#### **Important 重要**

To avoid confusing users, custom interactions you create should not duplicate or redefine existing, standard interactions.

为了避免造成用户迷惑，用户的自定义交互不应当重复或者重新定义已经存在的标准交互方式。



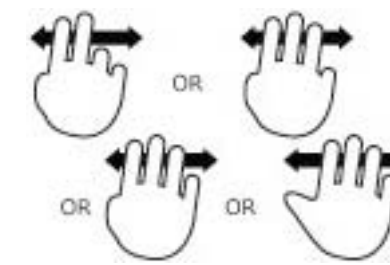
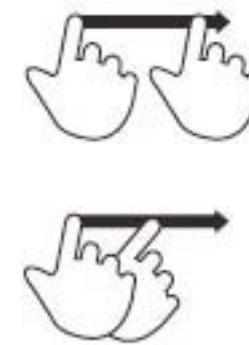
**Tap** One finger is placed down briefly to set focus, and then lifted up without sliding. Used to activate a control, select a word, position the caret, and so on.

**单击** 用一个手指触碰并在短时间内松开，触碰期间手指不在屏幕上滑动。用来激活控件、选择文字、光标定位等等。



**Hold, or press and hold** One finger is placed down without moving. Used to learn about the user interface (tooltip or self-revealing interaction) or access a context menu.

**长按，持续选中** 将一只手指放下并且不移动。用来了解界面（气泡提示或者自我说明类的交互）以及调用快捷菜单。



**Drag** One or more fingers are placed on a moveable UI element and move in the same direction. The fingers are not lifted until the action is complete. Used to pan, move an object, or select an object.

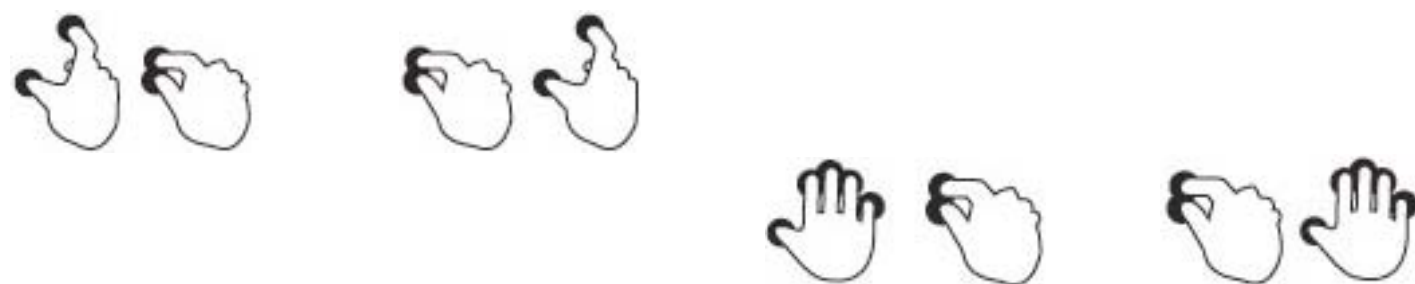
**拖拽** 将一只（或多只）手指置于可移动的UI元素上并且朝同一个方向移动，动作没有完成之前手指不离开。用来平移，移动对象或者选择对象。

**Cross-slide** One finger is placed on an object and dragged at a right angle to the panning direction. Used to select an object without invoking a command, or to move it to a new location (drag-and-drop).

**交叉滑动** 将一只手指放在一个对象上，然后向垂直于界面滚动方向滑动。（备注：panning是指在touch屏幕中向上下（或左后）滑动翻页这样的动作）用来在不调用命令的情况下选择一个对象，或者将对象移动到一个新的位置（拖动并放下）。

**Pan** One or more fingers are placed on a scrollable UI surface and move together across the surface. Used to scroll, or pan, across the app UI.

**滚动** 将一只（或多只）手指放在一个可滚动的UI页面上，然后在UI表面上移动。用来完成整个App UI界面或内容的滚动/翻页。



**Pinch** Two or more fingers are moved closer together or farther apart. Used to zoom in or out, optically or semantically.

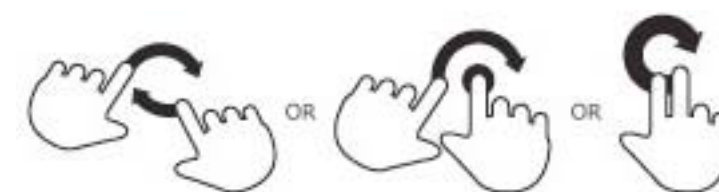
**捏放** 两只（或两只以上）手指在屏幕上朝不同方向移动，使指间距离缩小或变大。用作光学或语义上的缩放。

**Optical zoom** Two or more fingers are placed on a magnifiable UI surface and are moved farther apart or closer together. Used to adjust the magnification level and to obtain a more detailed view of the content.

**光学缩放** 两只（或两只以上）手指放在可放大的UI表面上，然后各个手指朝不同方向移动，使指间距离缩小或变大。用来调整物理放大程度以查看关于内容视图的更多细节。

**Semantic zoom** Two or more fingers are placed on a structured collection of UI elements and are moved farther apart or closer together. Used to drill down into different views of a collection of structured content to access different or more detailed information about categories and items in the collection.

**语义缩放** 两只（或两只以上）手指放在一系列的UI元素集合上，然后各个手指朝不同方向移动，使指间距离缩小或变大。用作内容集合在不同层级视图间的切换，以获得关于内容的更多细节信息。



**Rotate** Two or more fingers are moved in a clockwise or counter-clockwise arc. Used to rotate an object or content.

**旋转** 两只（或两只以上）手指以顺时针或者逆时针的弧度移动。用作旋转对象或者内容。

**Note 注意**

The drag, pinch, and rotate manipulations are performed with any number of touch points and can be compounded (executed simultaneously).

拖拽、捏放以及旋转可被任何数量的触点执行，并且可以复合（同时执行）。

## General guidelines for touch

### 触摸通用指南

If you use the standard Windows controls, you get touch, mouse, and keyboard support for free. If, however, you need to create your own UI or custom controls, you'll want to follow these guidelines.

如果你使用了Windows标准控件，你将自动获得触摸、鼠标以及键盘输入的支持。然而，如果你需要创造你独有的UI或者自定义控件，你需要遵循以下指南。

**Provide visual feedback** Instant visual feedback during interactions make users feel more engaged, confident, and in control.

**提供可视的反馈** 交互过程中的即时视觉反馈可以使用户更积极，自信以及有控制感。

**Avoid visual obstruction** When users interact with your UI using their hands and fingers, their view of the UI can be obstructed. This is called occlusion. You should consider the impact of occlusion when you lay out your UI.

**避免视觉阻碍** 当用户用手或者手指对应用的UI进行交互时，他们的视线可能被阻挡。这就是所谓的闭塞。你应当在设计UI时考虑这种闭塞的影响。

**Support hand motions** The physiology of hands and fingers and the casualness of touch interaction can result in interactions that are less precise, so you should design for sloppy touch interaction. For example, snap-points can make it easy to pan through content and stop at desired locations. Directional rails can make it easy to pan vertically or horizontally even when the hand moves in a slight arc.

**支持手势动作** 手和手指的生理结构以及触摸时的漫不经心导致了触摸交互是缺乏精准的，所以设计时应当为可能并不精准的触摸动作考虑。比如对齐点可以使平移内容并在希望的位置停下变得简单。方向线可以使水平或者垂直的平移变得简单，即便移动过程带有轻微的弧度。

**Support standard interactions** Where it makes sense for your app, you should support the standard interactions described earlier.

**支持标准交互** 为使你的App更易上手，你应当使用之前介绍的标准触摸交互。

**Avoid duplicating interactions** To avoid confusing users, you shouldn't create new interactions that accomplish the same goal as a standard interaction.

**避免重复交互** 为了避免用户迷惑，你不应当另外创造与标准交互所完成目标一样的新交互。

# Touch principles for Windows Developer Preview

## 交互规范（适用于Windows开发者预览版）

Touch introduces new opportunities and constraints around the user experience that you want to design. The following list of principles provides a basis for developing a user interface for touch interaction.

触摸交互对用户体验设计而言，既引入了新的机会，也带来了相应的规则（限制）。随后的将为你研发用户触摸交互提供基础概念。

**Natural and intuitive** Is the user interface clear, instinctive, and easy to use?

**天然直观** 用户界面是否清晰、直观并易于使用？

How does the lack of support for hover affect my design?

Are interactions accomplished through simple manipulations, such as taps and drags?

Does the design accommodate touch contacts that have shape and geometry?

Does the design take advantage of touch interactions?

缺乏悬停交互的支持对我的设计有什么影响？

交互是否都是通过简单的操作完成，比如轻拍或者拖拽？

触点的设计是否有统一的轮廓或者形状？

设计是否发挥了触摸交互的优势？

**Confident and connected** Is the user interface optimized for touch?

**准确关联** 是否已针对触摸交互优化了用户界面？

Does the design encourage users to touch without consequence or uncertainty?

Is fingertip and hand occlusion minimized?

Is an immediate response provided for every touch?

设计是否做到“明白、准确”，让用户放心大胆操作，而不需要去猜想可能出现的结果？

手指和手对操作的阻碍是否最小化了？

每次触摸操作是否都有提供即时反馈？

**Direct and engaging** Does the user interface offer a smooth and responsive touch experience?

**迅速响应** 用户界面所提供的触摸体验是否流畅且总能即时反馈？

Is the application aware and responsive?

Does the application provide instant gratification?

Can users manipulate content directly, rather than through controls?

Are physics and inertia used to engage the user?

App是否易懂且反应灵敏？

App是否能让人有即时满足感？

用户是否可以直接操作内容，而不是通过控件？

物理学以及惯性特性是否被使用以吸引用户？

# 为你的品牌而设计

试想一下，当用户在应用商店中看到你的Metro风格的应用时，当用户在开始屏幕上看到该应用时，当用户第一次运行该应用时，当用户第一次使用该应用时，该传达给用户什么样的讯息？抓住并利用这些机会运用应用的视觉设计来定义及推广你的品牌。



## 磁贴

“磁贴”是你的应用的脸面。在充斥着大量的应用的用户开始屏幕中，是什么激励用户想去运行你的应用？请确保“磁贴”强调了应用的品牌并且展示了其优越之处。

## 启动画面

虽然启动画面出现的时间很短，但启动画面仍然是在你的应用准备运行前极佳的品牌时刻——好好的利用它来表达应用的个性和特征。

## 初次运行

用户注册你提供的服务，登录帐号，以及添加用户自己的内容时，他们会看到什么呢？尝试在向用户索取信息之前，先证明你应用的价值，保证任何出现的图像都能巩固你的品牌。

## 登录页面（着陆页）

登录页面是用户每次运行应用后最先见到的页面。这里的内容应当有一个清晰的重点并作为应用定位的直接展示。

## 字形和图标

如果你的应用有独树一帜的审美风格，可以考虑为UI控件创造一套自定义字体和图标强化这种审美风格。

## 图形化

一张图片胜过千言万语。如果你决定使用这种形象化的表现，那么创造整体的视觉表现和观感对一个成功的应用是至关重要的。最佳的做法是使用简洁的视觉表征。把应用程序的本质提炼成为一个简单的想法或一张图片。

以下是一些例子：

- 若你的应用是关于烹饪的，考虑下是什么食物类型。如果是针对甜点的，那么从你的食谱当中挑选一张蛋糕的图片就是一个不错的选择。仅仅一张。一个蛋糕的画面会比满桌的甜点有更有戏剧性的视觉表现。
- 如果你的应用是关于纸牌游戏的，不需要展示整桌的纸牌。一张纸牌就可以表达一整副纸牌。更进一步的想，什么是这些纸牌中可能最具代表性的？Joker就是一个很不错的代表。

提示：不要使用用户界面的截屏来当作“磁贴”。小尺寸下，界面的细节是不容易引起注意的。在全屏尺寸的情况下，截屏可以当成启动画面或在类似场景中使用。

裁剪图像让图像的主题位于左侧，并引导视觉从左到右延伸。这样做给右侧创造了一个含蓄的空间，从而巧妙地鼓励用户追随他们的天性去学习和探索。

## Windows应用商店中的推广艺术

此部分暂未公布

# 为质量而设计

Metro风格应用的质量要通过几个测试过程的评估。本主题描述了由Windows程序认证组件 ( Windows App Certification Kit ) 所执行的测试。通过这些测试，你所设计的应用将会在质量和可靠性方面有所改进。

Windows程序认证组件 ( Windows App Certification Kit ) 测试以下几个类别。当使用该组件测试完你的应用后，你将获得关于该应用详细的测试报告。提前了解Windows程序认证组件为什么而测试，在设计应用时，请在这些方面多留心。

- 应用失效
- 应用清单
- Windows 安全特性
- 支持的平台APIs
- 启动性能
- 中止执行
- 应用程序包生效
- .NET版本检查

## 应用失效

应用绝对不能崩溃或意外中止。诸如崩溃或意外中止等应用失效情况都会扰乱或挫败客户。消除此类失效会改进应用的稳定性和可靠性，而总体来说，也给客户带来了更好的体验。

## 应用清单

应用清单必须包含所有需要的属性

## Windows安全特性

应用必须使用Windows安全特性。

## 支持的平台API

应用使用的API必须只能来自Metro风格应用的Windows软件开发工具组（SDK）。

## 启动性能

应用应该在营造一个快速且即时响应的启动体验时，而只消耗的一定且合理的系统资源（CPU，文件IO以及内存等），从而用户可以在未打开的应用之间可以快速切换并执行多任务。

## 中止执行

应用应当在消耗大量的系统资源（CPU，文件IO以及内存等）的同时也能有一个快速且响应及时的启动体验，并且保证这些资源能被有效释放。

## 应用程序包生效

如同在应用包清单概要中所解释的，应用清单应当必须在resources.pri文件中定义有效的资源。

## .NET版本检查

应用必须使用.NET4.5或者更新的版本

# 为全球化和本地化而设计

设计你的Metro风格的应用，使之可以在全球范围内使用。在最初设计应用的时候遵循这些指南，之后你可以针对不同的本土和市场做些改造。

许多应用开发者仅考虑他们自己的语言和文化而创建应用。当应用变得成功并且在其他语言和市场中也有需求时，改造这个应用就会很难。文本和图像源可能直接由代码指定，这让转换成别的语言及为其他文化做准备变的很难。以下过程可以让应用第一次被设计时，其国际化和本土化的过程就变的很简单。

- **国际化:** 设计和开发应用并让其在多个不同的地方使用和运行的进程。设计你的应用特性，并让该应用可以支持对多个地方，而编写应用的代码在任何地方都能同样运行。

- **本地化:** 针对特定市场改造一个应用的进程。包括翻译用户的界面，调整弹出对话框尺寸，自定义特性，以及测试结果，以保证该应用在目标市场中正常运作。

## 全球化和本地化的编码指南

实践	说明
把资源同代码和标签中分离出来	这样做的好处： <ul style="list-style-type: none"><li>• 不需要影响到资源内容而轻易管理维护代码</li><li>• 资源内容可以被维护，本地化以及自定义修改而不会给代码带来任何错误</li><li>• 本地化可以交给第三方执行，不论是机器翻译还是语言专家翻译，而不需要学习代码知识</li></ul>
隔离可本地化的资源	为本地化读者提供必要的标注 在代码和标签中添加引用资源 决定哪些图片需要本地化并把他们放进带有语言标签的文件夹中 在源文件中维护字符串资源
使用合适的字符串	<ul style="list-style-type: none"><li>• 在源文件中内嵌字符串，从代码中分离出来</li><li>• 允许出现比英文字符更长的语言字符</li><li>• 使用完整一致的字符串资源，以便更易本地化</li><li>• 避免使用口语字符或者难以翻译的字符串</li></ul>
使用合适的图片	<ul style="list-style-type: none"><li>• 可能的情况下，使用所有文化中都传达同一种含义的图片</li><li>• 分离需要本地化的图片，比如包含需要翻译文字的图片或者需要镜像的图片</li></ul>
为不同的分辨率调整图片	每一个图片都准备以一个典型96dpi设备的原始尺寸做三个版本，100%，140% 和180%

## 全球化和本地化的内容指南

- 为本地化使用合适的字符串。
- 分离需要本地化的图片。
- 在文本和图片被本地化时要留意文化差异和政治问题。

## 恰当地使用字符串

不把字符串直接放入代码中，而是把他们内嵌于所支持区域中的源文件中，而你或者购买商户的译者可以把他们翻译成为支持其他特定市场所用，而不用转换整个应用程序的代码。

- 增加字符串的空间

为了适应更多本地化的语言，确保设计的布局能够允许比容纳英文字符更多地空间。一些语言需要大约多于英文40%的空间。例如，打开控件的多行显示支持以及缩小文字显示能够为单个字符串提供更多的显示空间

- 避免运行时串联的字符串

考虑一下这串代码：“the{0} could not be synchronized.({0}不能被同步)”。变量{0}可以被各种词汇代替，比如一个约会事项，任务或者文档。这个例子也许仅适用于英语的语言，但对其他语言来说可能就不适合。另外很多在英文中的名词不具备性别特征，但其他多种语言则具备这样的特征，因此在合适的地方就需要做出相应的调整。更好的方式是使用完整的字符串以便可后期单独本地化。

- 使用一致的字符串

使用单一字符串并传达在同一产品中统一使用的使用的概念。把这个字符串放到一个源文件中并只需本地化一次，从而减少呈现过程中产生多种变化。

## 不恰当地使用字符串

不要不恰当的使用字符串，从而难以被翻译而不适应其他地方市场

- 不要暴露过多的字符串去本地化

本地化具体的字符串，而不是标签，参考下面的例子：

过度本地化的字符串	正确本地化的字符串
<code>&lt;link&gt;terms of use &lt;/link&gt;</code>	terms of use
<code>&lt;link&gt;privacy policy&lt;/link&gt;</code>	privacy policy

把上面例子中的<link>标签也放在源文件中将会表示该标签本身也会被本地化，而导致该标签失效。而真正需要被本地化的应该是字符本身。

- 避免使用缺乏充分指导信息的口语和隐喻

语言往往是针对某一特定族群的，比如文化和年龄，因此翻译的难度很大，因为只有那个族群的人才会使用哪个语言。同样的，对某一个人的来说很有道理的隐喻对另外一个人而言可能完全没有意义。例如，一只“蓝色的鸟 (bluebird)”表达了滑雪文化的一些具体的表征，但不懂此文化的人便难以理解这层含义。如果你采用了一些非正式的声音或者语气，确保在翻译后有对其真正意义上的具体解释。

- 避免使用技术术语，缩写或者首字母缩写词

这些词汇对非技术读者来说根本无法理解，而且难以被翻译。在日常生活的对话中，他们不会使用这些词汇。这些过重的语言通常被用来描述软件或者硬件的各种错误，而且出现在恰当的时候。在这里，我们鼓励你排除这些词汇并重新改写这些字符串。

- 避免使用模糊的词汇

词汇的意义常常是对内容敏感的。例如，字符串“text”和“fax”既可以当成名词使用也可以当成动词使用，这取决于他们使用的上下文环境。如果某些地方比较模糊，为本地化的读者们注上哪些词是名词，哪些词是动词，形容词。

- 不要使用名词堆叠

一个名词堆叠是用两三名词同时使用并传达另外一个名词的意思的用法。比如“Network adapter card plugin”。很难解释这个词的具体意思是什么 -- 是一个网络适配器卡的插件还是一个网络适配器的卡片插件，而且这个词也很难本地化。考虑到在界面上如何呈现这些字符并改写他们，或者提供一些注释给本地化的读者们，指出哪些词是组合在一起的，从而表达其准确的含义。

# 使用设计检查表

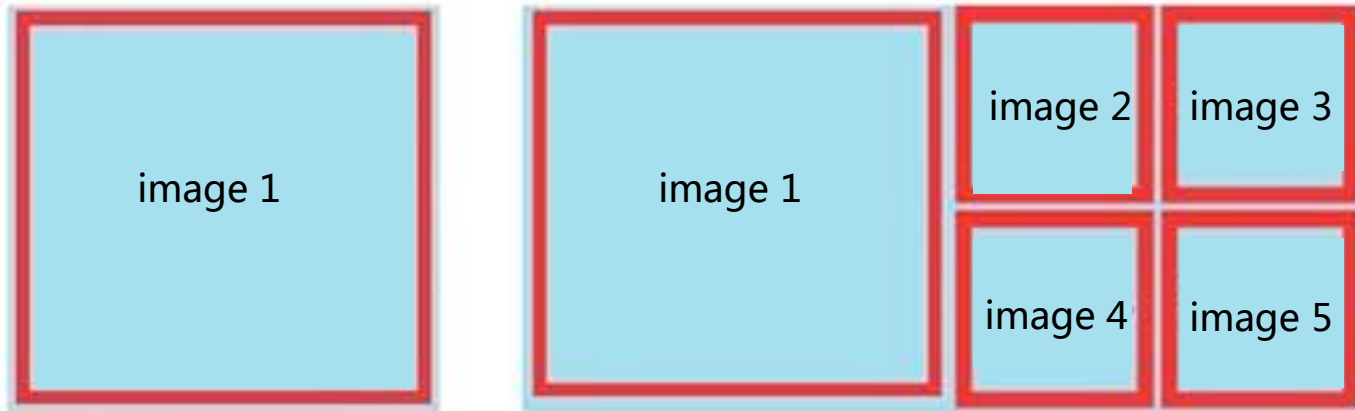
[这是一个初步的文档并且很可能被修改。]

你可以使用这个不太完善但能够覆盖你所需要信息的检查表，开发一个Metro风格应用程序。

## 基础知识

这些设计基础知识适合所有应用：

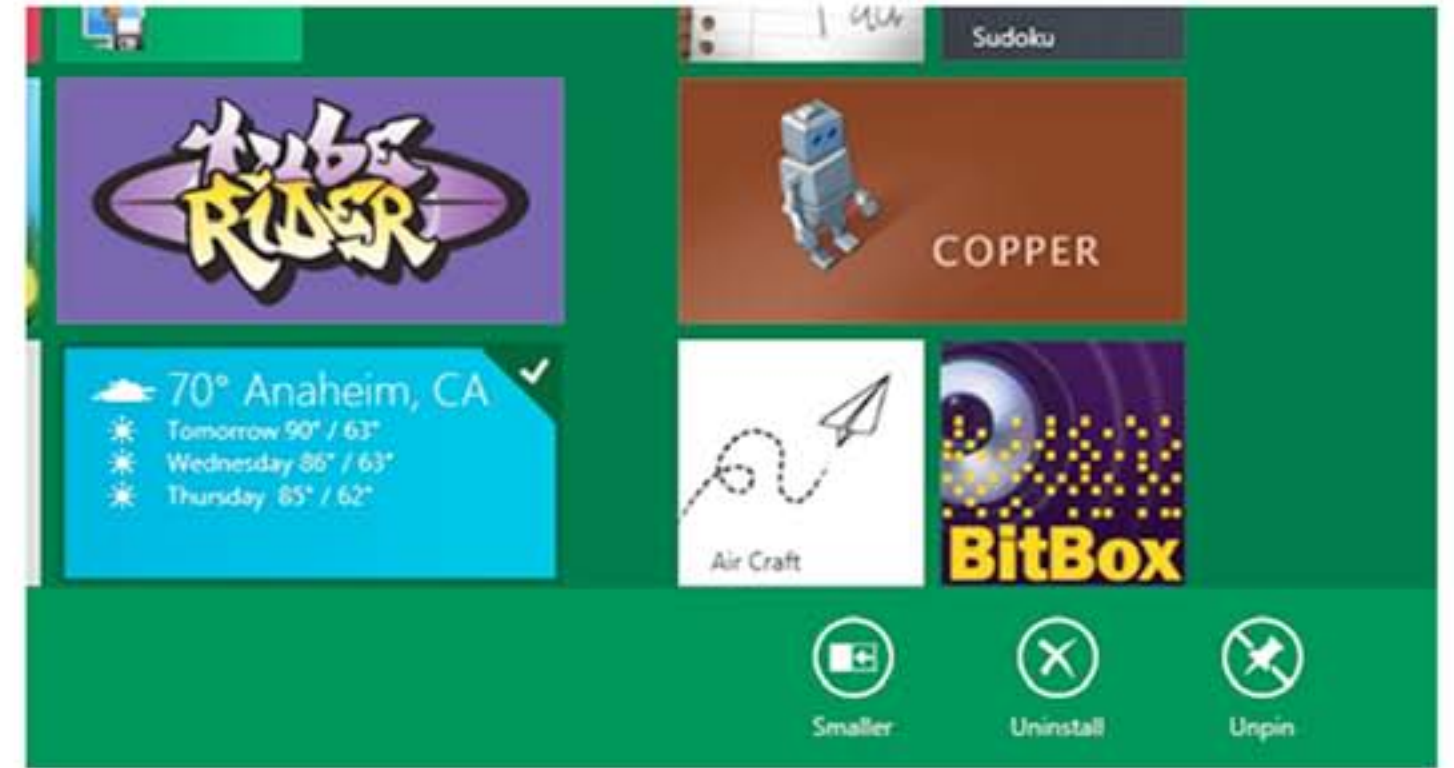
- 应用“磁帖”：使用应用“磁帖”来吸引、激励用户使用你的应用，给人以持续更新并很贴切的感觉。



- 启动画面：这是一个可以宣传品牌的时刻。使用颜色与图形化的logo来加强用户对品牌的印象。



- 生命周期管理：你的应用可以被开启或关闭退出屏幕，并且当在后台不再被使用时，程序将会被终止。保存和还原状态来尽可能的保持上下文。
- 应用栏：把应用程序的命令放在应用栏里，方便用户找到。



## 视图与形式要素

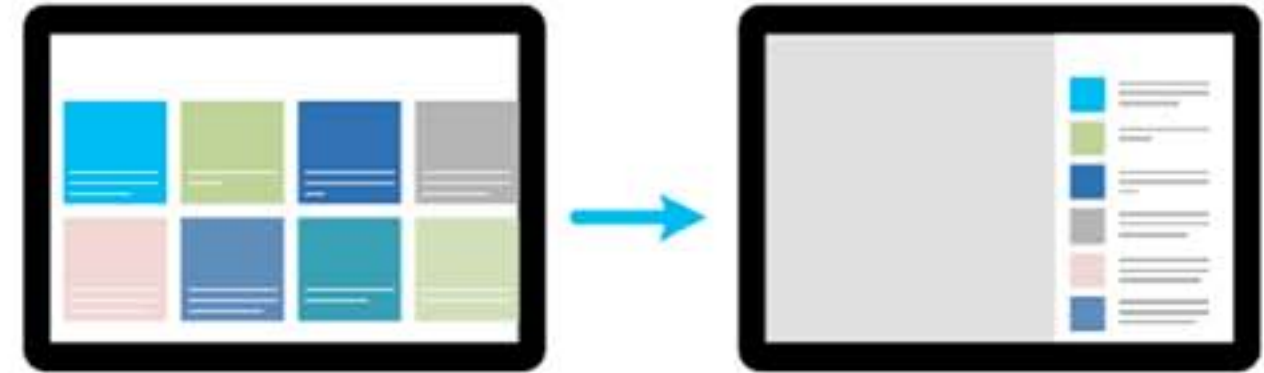
以下特性可以帮助你创建丰富的任意形式与视图的用户体验：



- **全屏**：先为横屏体验设计，你的应用可以满足所有的形式要素。



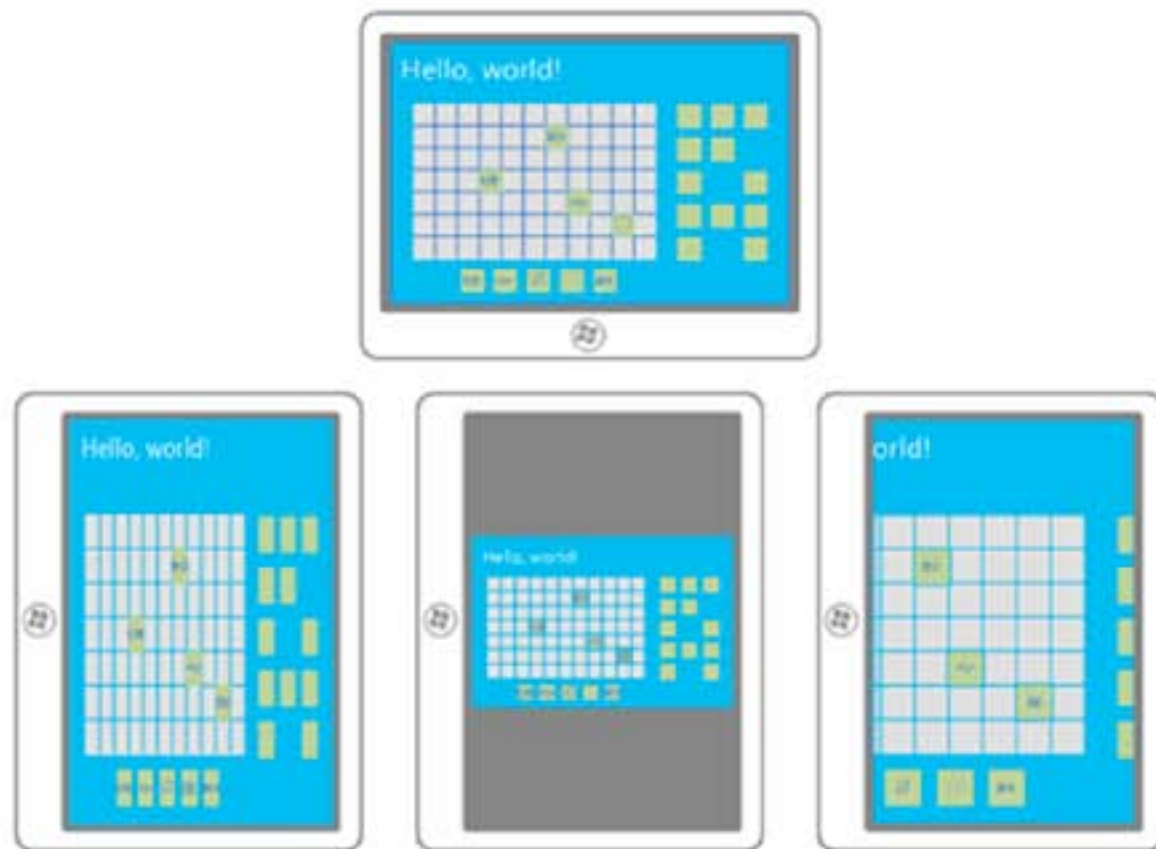
- **辅屏**：为多任务而设计。在使用某应用的同时，用户还想聊天、浏览网页、看电影甚至缴税等，因此在辅屏与非辅屏视图之间切换时，确保你的辅屏试图是有用的保持上下文的连贯。



- **填充视图**：当你的应用采用填充视图，是为了减少横向版面的移动幅度。



- **纵屏**：屏幕是可旋转的，所以为更高更宽的显示并且保留功能而优化内容的布局。



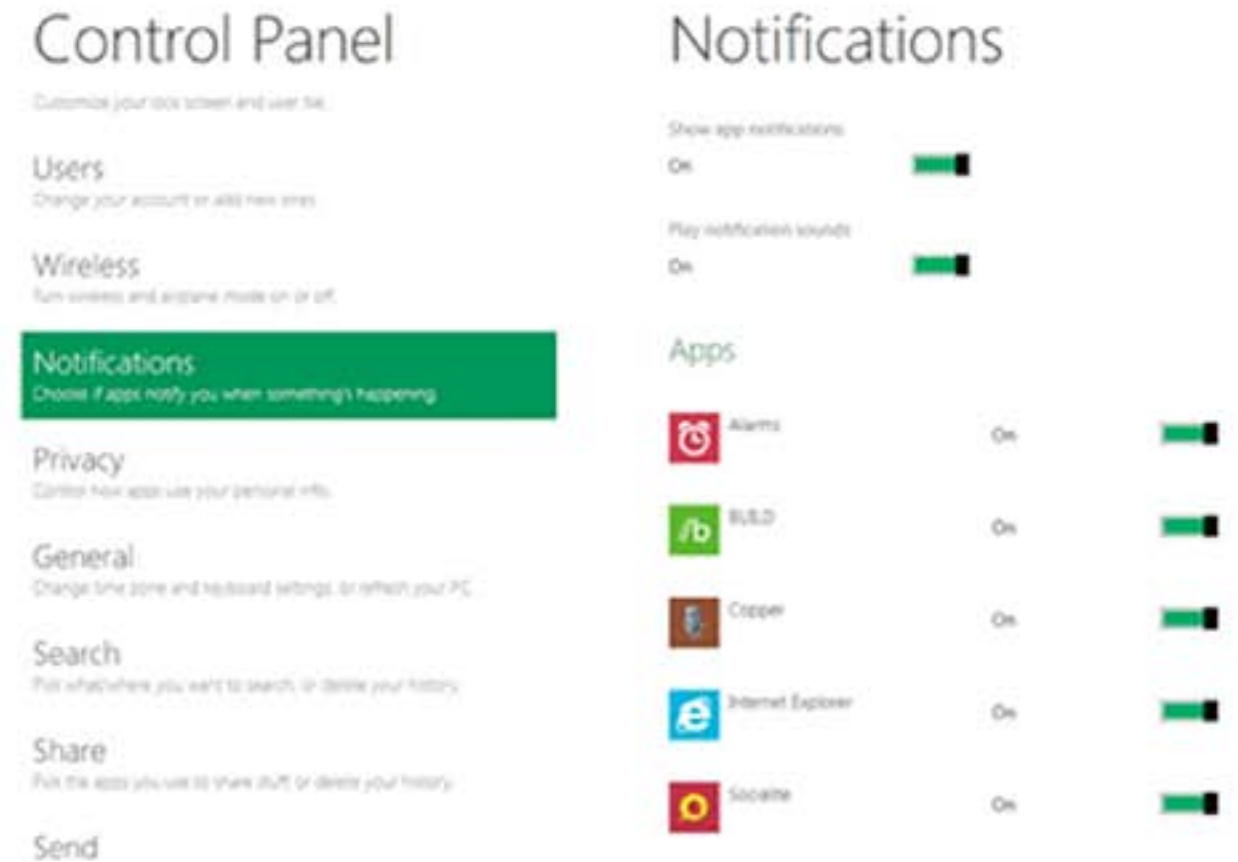
- **缩放**：采用流畅的布局，并且保证在应用中使用的图像在缩放后可以很好的呈现。Windows系统会缩放你的应用确保与物理尺寸保持一致，而忽略设备的分辨率。



### Windows应用协议

这些特性可以帮助用户采用更有意义的方式处理内容：

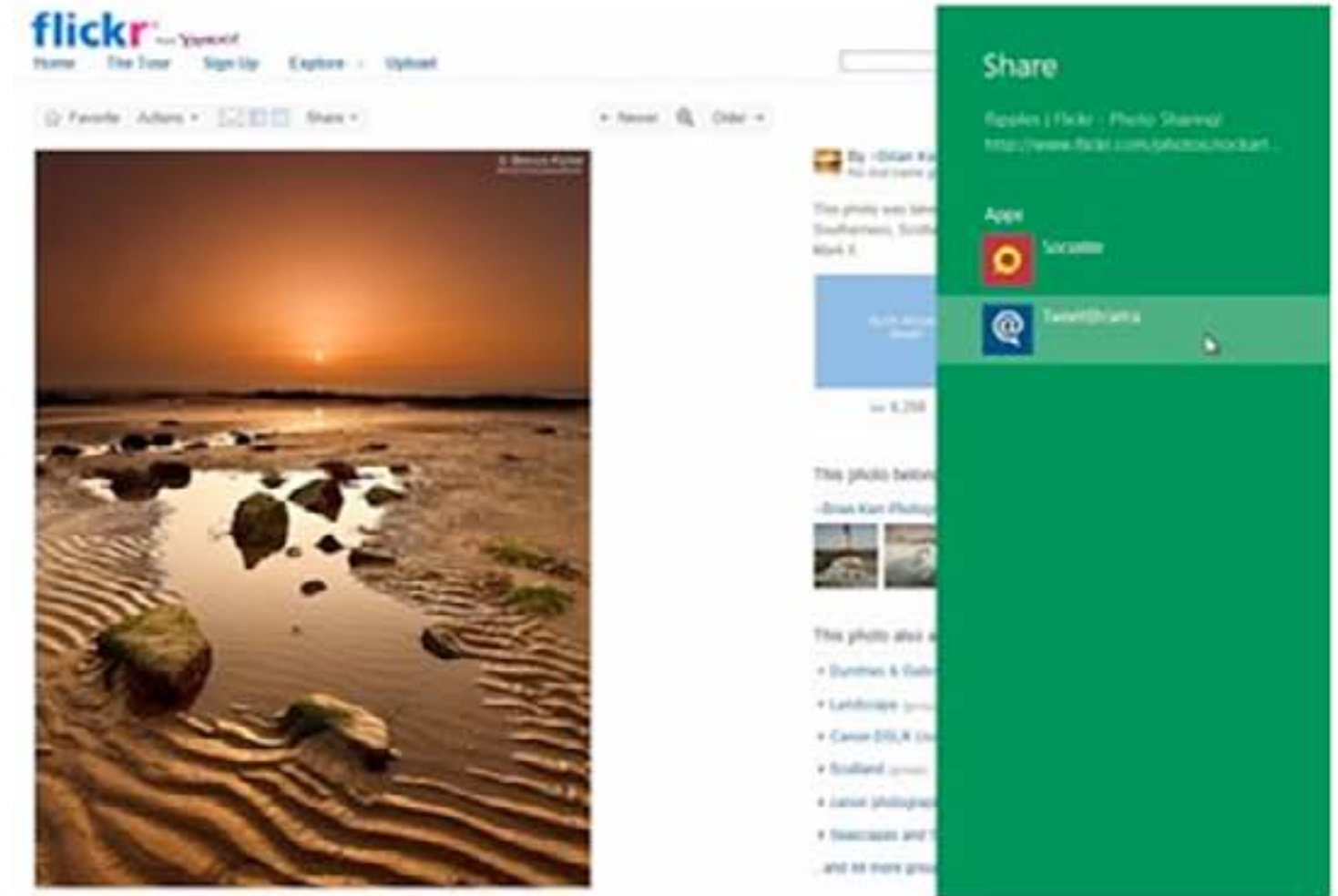
- **设置**：将应用所有的设置项放在一个界面内，并且让用户以他们熟悉的常规的方式来配置应用。



- **搜索**：用户可以在系统的任何地方快速的查找应用里的内容。



- **分享**：用户可以把自己应用里的内容分享给其他人，也可以收到他人的分享，并可以把内容呈现给用户。

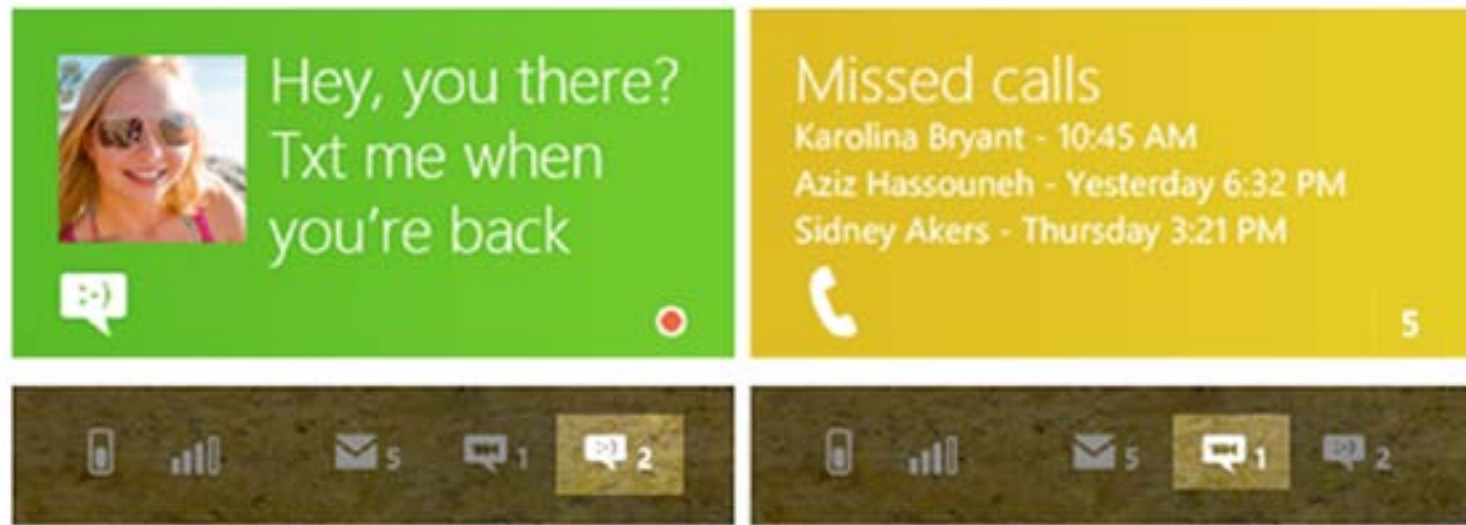


更多信息，请查看[windows应用合同协议](#)和[搜索、分享与连接](#)。

## 让你的应用更强大

以下特性让你的应用脱颖而出：

- **通知**：当用户不在应用的前台时，通过“推送”式通知，让用户感知到需要及时处理或他相关的内容，来邀请他回到该应用。
- **内容“磁帖”**：在开始屏推送有趣的，深度连接来自应用程序的内容，并且通过“磁帖”直观的开启应用。



- **语义缩放**：帮助用户浏览大量的内容。通过放大或缩小，你的应用可以呈现不通种类的信息给用户。
- **文件选择器与选择器扩展**：用户可以从本地PC、已连接的存储设备、家庭组及其他的应用程序加载到你的应用程序或加载回去。你还可以提供拓展的文件选择器，来允许其他应用程序加载你的应用里的文件。
- **动画**：通过动画让你的应用感觉是快速而流畅的。帮助用户理解上下文的变化，并符合视觉转换的体验。
- **接近手势**：用户使用一个“单击”的操作就可以连接两个或多个设备。当希望多个用户加入时，例如多人游戏。这种手势让应用的体验很抢眼。
- **用户“磁帖”**：通过加载用户的图像或者在应用里设置过的内容，放在windows个人“磁帖”上，使你的应用更加个人化。
- **漫游**：通过设置持续漫游状态，用户可以方便从家庭电脑、办公PC到个人桌面等任何位置使用你的应用。

请在[什么是Metro风格应用？](#)里了解更多关于是什么使得Metro风格应用与桌面应用不同。

## 测试

当你已经在测试原型的时候，请使用Windows App Certification Kit ( windows应用程序认证套件 ) 来测试你可能遇不到的任何技术要求。

# 磁贴

## —设计规范与检查表

以下主题介绍了为你的应用程序创建磁贴的最佳实践方法。

- 用户体验设计规范
  - 如何选择磁贴的尺寸
  - 适当使用磁贴的情况
  - 不适当使用磁贴的情况
- 适当使用提示标记情况
- 不适当使用提示标记情况
- 适当使用默认磁贴情况
- 不适当使用默认磁贴情况
- 其他的设计思考

## • 用户体验设计规范

### • 如何选择磁贴尺寸

- 如果你的应用程序不是使用通知发送更新的信息给用户，应使用正方形。
- 如果你的应用程序只支持基于场景的简短摘要式通知，应使用正方形并带有提示标记。例如，短信服务的应用的磁贴将会只显示收到新消息的条数。
- 如果你的应用程序发送的动态消息不需要显示在开始屏幕上，应使用方形尺寸。例如，支付存根应用程序。
- 只有当你的应用有跟新和有趣的内容要显示给用户，并且这些通知是频繁更新的（至少每周一次），应使用长方形。宽“磁贴”里的内容要新鲜而活跃。

### • 适当使用磁贴的情况

- 磁贴的设计师应该试图为应用创造精致的磁贴，呈现新的、定制的及引人入胜的内容，用户将在“开视屏幕”查看到，并且引导用户开启应用。
- 对于一套应用，应为套件中每一个独特的应用创建一个磁贴。

### • 不适当使用磁贴的情况

- 在一个应用里，不要创建多个磁贴来打开子程序。每个唯一的程序应该只有一个磁贴。相反，对于某些场景要考虑二级的磁贴是否是更好的选择。
- 不要用应用程序里附加的或辅助磁贴扰乱用户的开始屏幕。只有当程序真的特殊需要为每个单独的核心应用创建磁贴时，才允许创建多个。
- 不要为应用程序的配置或故障排除创建磁贴。那种功能应该在应用的设置提供给用户。

### • 适当使用提示标记的情况

- 当数字足够小到在你的场景里是有意义的时候可以使用数字。如果您的提示总是显示50或者更大时，考虑使用其他系统符号。要考虑避免数字麻痹，统计的数字应该从上次退出开始算起，而不是绝对的数值。例如：显示从上次退出应用后一共的未接来电数，要比显示从安装起所有未接来电数更有意义的。
- 当数字大到没有使用意义或产生了数字麻痹，要使用一种系统符号来表明发生了新动态。例如，最新没有阅读过的大量的RSS feed统计的数字是会产生数字麻痹的。取而代之，应该使用newMessage系统符号。
- 如果数字变得没有意义，那么使用图标。例如，当磁贴为一个播放列表显示一个“暂停”的通知，那么它应该使用暂停符号，因为在这种场景下数字是没有任何意义的。
- 当数字是不确定时，使用“新通知”符号。例如，在一个社交媒体应用的磁贴提醒显示“10”，其含义是10个回复、10个新消息，10个通知，还是它们混合一起的统计呢？
- 在需要表达大量的场景下使用“新消息”符号，例如，邮件或者社交媒体，磁贴上的提示标记可能不断的更新到超过最大数字99。用户总是看到最大数字会产生数字麻痹，而且一个总保持常量的数字不能传递有用的信息。



### • 不适当使用提示标记的情况

- 不要在宽磁贴的主题内容里重复显示提示数字，因为两个实例有时会不同步。
- 如果一个符号告诉用户的内容从不改变，不要使用他。符号代表通知或者瞬间的状态，而不是任何永久的状态说明。

### • 适当使用默认磁贴的情况

使用默认磁贴上的图像来显示该应用的品种，基本上作为呈现应用标识的画布。

### • 不适当使用默认磁贴的情况

不要把默认磁贴设计得诱导用户开启应用程序，例如在磁贴上写上“点击我！”

## • 其他设计思考

- 当你考虑如何在磁贴上传达该应用的品牌信息，请选择应用程序的名字或者标识。这些已经定义在应用程序列表里，并且开发者可以选择显示哪一个。然而，一旦你已决定使用名字或者图标，为了一致性请持续使用下去。
- 如果应用的名字在磁贴上一行显示不下，请使用简称或有意义缩写。例如，Microsoft Word 2010 使用“Word”。当名字的长度超过了最大值将被截断以省略号代替。名字的最大长度为18个英文字母，并且包含特殊符号。
- 当你使用peek模板时，显示可以从框架循环的两端开始—文本完全置底或者文本完全升起—并动态向上或向下切换到其他框架。因此，确保每个peek模板框架里的内容是独立的。
- 不要把磁贴当广告用。
- 避免在磁贴上使用眼花缭乱的颜色；设计简洁、干净、优雅的磁贴将会比嘈杂吸引人的设计更会成功。
- 不要使用带文本的图片；为有文本内容需求设计一个带文字区域的模板。
- 不要依赖磁贴发送即时紧急的消息给用户。例如，一个磁贴作为新闻应用发布紧急地震疏散消息的媒介。推送通知是通知自然灾害消息的更好的媒介。
- 避免图片内容看起来像超链接、按钮或者其他控件。磁贴不支持那些元素的功能，并且整个磁贴是个单一的点击目标。

## 磁贴概览

元件可用于定义一个磁贴更新。他们详细说明模板，图形和用于自定义磁贴的文字，品牌细节特征以及语言信息。

这些元件以及它们的参数是通过文档对象模型（DOM）的控制功能来调整控制，并用以自定义磁贴的内容。

要定义一个非默认磁贴的起始内容，使用 `TileUpdateManager.getTemplateContent` 来获取可修改的模板内容。

要对一个已经存在的非默认磁贴的内容定义更新，使用 `TileNotification.content`来获取可以被编辑修改的现有磁贴的内容。

元件名	描述
binding	定义磁贴模板
image	说明在磁贴模板中所使用的图片。所提供的图片应该符合指定模板或模板内图片的大小和形状的要求
text	说明在磁贴模板所使用的文字
tile	基础磁贴元件，包含了一个visual元件
visual	包含多个binding的子元件，每个里面都定义了一个磁贴

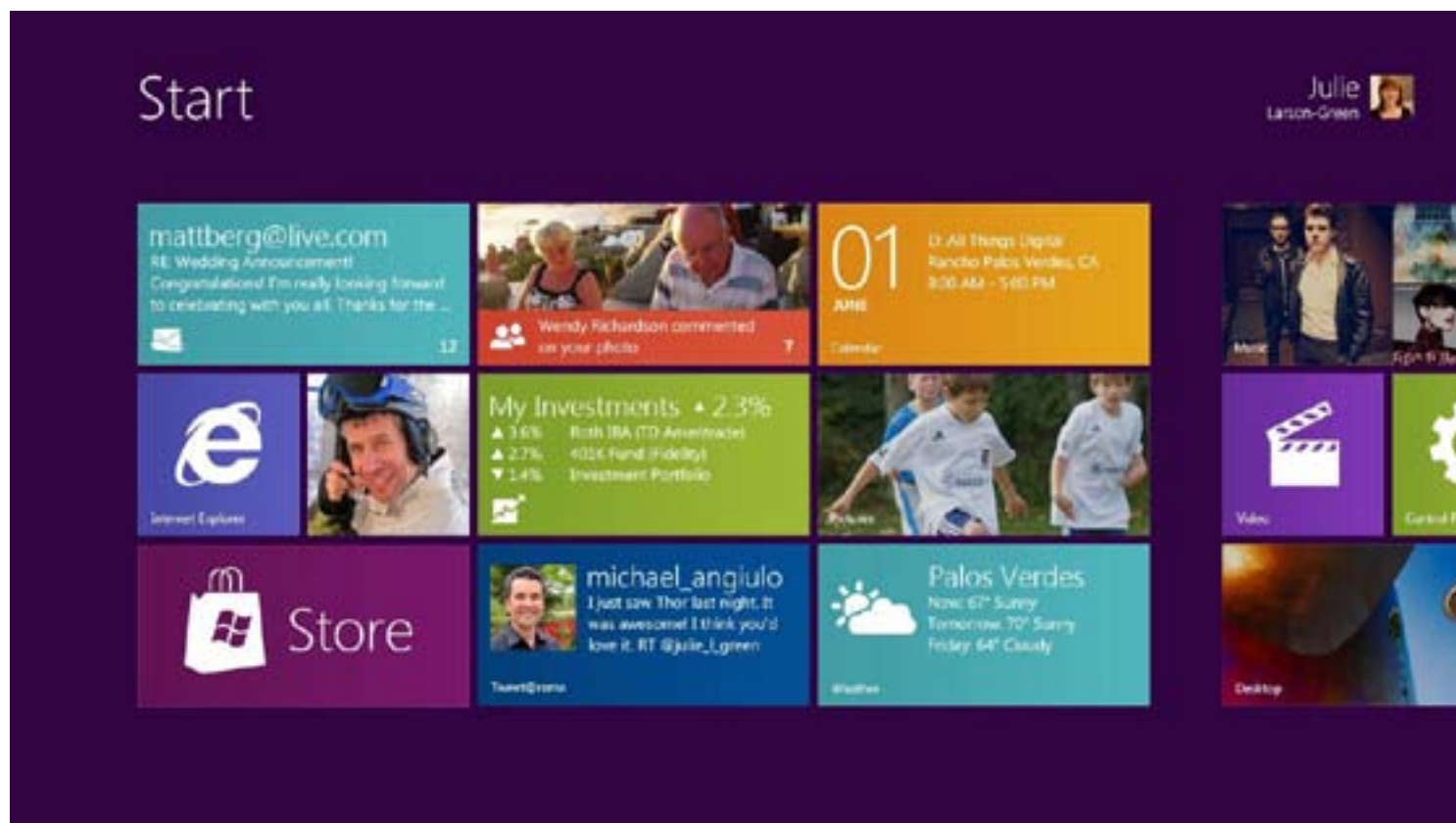
# 二级磁贴

创建并管理二级磁贴

二级磁贴设计规范与备忘录

- 用户体验设计规范
  - 适当使用二级磁贴的情况
  - 不适当使用二级磁贴的情况





二级磁贴允许用户推广有趣的内容和深链接——一个锁定应用内特定位置的参考——从Metro风格应用到开始屏幕上。二级磁贴使用户可以用播放列表，图片专辑，朋友列表和其他重要项目个性化他们的“开始屏幕”体验。

最常见的创建二级磁贴选项是在界面中的“锁定到开始屏幕”。锁定内容就是为应用创建一个二级磁贴。这个选项通常被呈现为应用栏上的一个符号。通过触摸或点击选择一个二级磁贴会进入到父级应用中，并获得集中注意力于被锁定内容或联系人的专注体验。

**只有用户可以创建二级磁贴；应用不能自动创建二级磁贴。**并且用户可以明确的删除二级磁贴，无论是在“开始屏幕”还是通过该应用程序。

## 二级磁贴和应用磁贴的关系

二级磁贴和单一的父级应用相关联。他们被锁定到“开始屏幕”上用以提供用户一个一致而高效的方式来直接启动父级应用并进入到经常使用的区域。这既可以是一个包含了经常更新内容的父级应用的一小部分，或者是一个指向应用中特定区域的深链接。

二级磁贴的场景案例包括：

- 天气应用中针对特定城市的天气更新
- 日历应用中对即将到来事件的摘要

- 社交应用中重要联系人的状态和更新
- RSS阅读器中的特定feed

任何用户想要监视的经常更新内容都是使用二级磁贴的好选择。一旦二级磁贴被锁定，用户可以通过瞥一眼磁贴获取更新并使用他来直接进入到父级应用中，专注于被锁定的内容或联系人。

二级磁贴在许多方面和应用磁贴很相似：

- 使用模板来决定布局
- 必须包换一个152x152像素的标识
- 可选择使用312x312像素大小的标识
- 可以显示提醒和提示标记
- 可以在“开始屏幕”上重排列
- 可以同时“全部程序”和“开始”中显示
- 当应用被卸载时会被自动删除

但二级磁贴和应用磁贴也有显而易见的区别：

- 用户可以在任何时候删除他们的二级磁贴而不用删除父级应用
- 二级磁贴可以在运行时被创建。应用磁贴只在程序安装时被创建
- 在添加二级磁贴之前会有飞出提示让用户确认

## 使用锁定来创建二级磁贴

开发者可以用应用栏提供在“开始屏幕”锁定和解锁内容的选项。Windows提供一个大头针符号来配合应用栏的特点，同时提供为了完成锁定动作程序应该启动的必要JavaScript代码。如果有的强制场景不使用应用栏，提供的程序实现也依然能够被应用通过元素来启动而不是应用栏。

以下是对创建二级磁贴需要采取的动作的高度概述：

1. 用户通过应用调用锁定命令，一般是通过应用栏的“锁定到开始屏幕”符号。
2. 应用提供创建二级磁贴的必要信息，包含一个独特的ID。其他信息包括磁贴背景色，字体颜色，显示名称，通知能力，标识等等。
3. Windows显示一个飞出提示显示磁贴的预览，使用户确认创建。
4. 用户确认后二级磁贴被添加到“开始屏幕”上。
5. 应用打开一个通道统一资源标识符（URI）接受来自二级磁贴的通知。

解锁流程相似：

1. 用户通过应用调用解锁命令，还是通过应用栏上的解锁符号。
2. 应用提供移除二级磁贴的必要信息。
3. Windows显示飞出提示显示将被移除磁贴的预览并请求用户确认移除。
4. 用户确认后二级磁贴从“开始屏幕”上移除。

这些步骤中的关键点是应用必须提供用于创建或移除二级磁贴的信息并让用户在动作中决定最终选择。源应用处理来自“开始屏幕”的二级磁贴激活请求。这一过程通过应用磁贴使用的标准激活机制完成。二级磁贴不能被“开始屏幕”以外的任何机制激活。

源应用还负责以下几点：

- 为二级磁贴创建一个ID
- 请求一个通道统一资源标识符来推算通知到二级磁贴
- 适当的显示锁定和解锁命令
- 处理被二级磁贴显示的深链接的激活请求

## 二级磁贴设计规范与备忘录

以下主题介绍了为你的应用程序创建二级磁贴的最佳实践方法。

### • 用户体验设计规范

注：**只有用户可以创建二级磁贴；应用不能自动创建二级磁贴。**并且用户可以明确的删除二级磁贴，无论是在“开始屏幕”还是通过该应用程序。

### • 适当使用二级磁贴的情况

- 二级磁贴像所有磁贴一样在“开始屏幕”显示作为动态的端口，显示频繁更新的新消息或内容。二级磁贴采用与其他磁贴相同的机制呈现通知或更新。为了当应用程序不在运行时更新磁贴上的动态，二级磁贴必须请求并且打开一个通道统一资源标识符（URI）和windows推送通知服务（WNS）。
- 尽管创建一个二级磁贴的决定权完全归属与用户，在app里面允许用户锁定的这个区域，可以遵循以下设计规范由开发者来决定：
  - 如果焦点内容已经被锁定了，那么应用栏应该显示可用的解锁按钮使得用户可以使用它来解除对内容的锁定。
  - 如果焦点内容是不能被锁定，那么应用栏就不应该显示锁定按钮。如果应用程序把锁定命令放在应用栏的外边，那么在应用栏里的锁定按钮应该不显示或显示为不可用。锁定按钮是否可用或显示，取决于界面和当要启用它时在哪里出现的场景。
  - 使用系统提供的锁定和解锁的符号。
  - 开发者还可以在创建二级磁贴时添加一些符合情景的互动特效。
- 应用程序应该为二级磁贴使用有意的、可重建数据的ID。针对以下原因这非常重要：
  - 当应用程序被安装在第二台计算机时，二级磁贴应该可以被用户重新获取。使用有预期的二级磁贴ID，对于帮助应用程序本身理解如何使用这些磁贴是很有意义的。
  - 在运行时，应用程序可以查询某个特定的磁贴是否存在。
  - 二级磁贴平台可以返回所有属于某个应用的二级磁贴。使用有意义的ID可以帮助一个应用检查整套的二级磁贴，并且执行适当的行为。例如，对于社交媒体应用，ID可以标识应用为哪些联系人创建了磁贴。

### • 不适当使用二级磁贴的情况

- 不要把二级磁贴当作访问不会改变的离散文件或其他静态内容的快捷方式。
- 不要把二级磁贴当作与父级应用程序交互的某虚拟命令的按钮，如某个“跳到下一曲”磁贴。

# 启动画面

## —设计规范与检查表

启动画面是所有Metro风格应用都需要的，本章的主题在于描述启动画面的目的以及如何有效的利用启动画面。

你默认的启动画面会在用户启动应用时显示，并在程序初始化资源时提供及时的反馈。当应用的第一个视图可以进行交互时，启动画面被撤掉。良好的运用启动画面可以提升用户对应用执行情况的认知。

# 你为什么需要 启动画面

你可以定制应用读取时的显示，通过制定启动画面图像和背景色，并使用启动画面API来显示更长的时间，说是当启动画面被撤掉时来提醒应用。

延长启动画面的显示时间以使你的应用完成额外的启动任务并显示附加的读取信息。例如，你的应用可能需要通过网络读取资源。你将通过获取启动图像的坐标来扩大你的启动画面，从而构建自一个模仿默认启动画面的自己的启动画面，但同样能提供给用户额外的读取信息。这种模仿默认启动画面的方式保证了在对读取进程完全控制的同时也为用户保持一个整洁，一致的读取体验。

如果你有入口动画，当启动画面撤掉时的检测会让你知道该什么时候开始入口动画。

• 恰当的使用启动画面

• 使用一张能清晰标示应用的图片

使用图片和色彩主题来清晰的标示你的应用，这样用户才有信心他们启动了正确的应用。它还帮助强化你的品牌。

• 使用透明PNG图片以求最好的视觉效果

• 使用启动画面来创造无缝的应用读取体验

如果你的应用需要额外的时间读取资源，考虑创建相同的启动画面和第一视图。一个应用的视觉读取体验始于启动画面，接着进入到应用的第一个视图（可能用于显示额外的信息），并结束于应用的登录页面。当你的启动画面和第一视图相同时，用户会感知到平滑的过渡。这不仅是从用户处提炼出的多步读取过程，还使应用对体验有完全的控制，能够进行附加的信息读取和动画。



• 使用入口动画时，利用启动画面API来侦听撤销事件，从而了解到何时开始动画。

• 启动画面的不合适使用

• 不要利用启动画面来显示广告

启动画面的目的是在应用实际读取时让用户知道他们想启动的应用正在启动。在启动画面中引入不相关的元素会降低用户认为他们启动了正确的应用的信心，并让通过一撇来辨识应用变的更加困难。

• 不要利用启动画面显示“关于”页面

启动画面不应该被用作展示版本信息或其他的应用元数据。在应用的Windows Store描述中或者在应用内显示这类信息。

# 应用栏

## —设计规范与检查表

[本文档是初步的，并且有变更的可能。]

当用Javascript制作Metro风格应用的应用栏时，遵循以下指南

# 用户体验规范

用应用栏给用户呈现导航，命令以及工具。

## 应用栏应当放置在应用中的何处

你应当把包含命令的应用栏置于屏幕的底部。当应用有全屏视图且在内容区内没有导航时，你也可以把返回按钮以及标题放在屏幕顶部的应用栏上。如果有些命令对于用户完成体验是必要的，比如购买一件商品，可以把它置于内容区内部而不是应用栏上。

## 选择正确的可解除模式

当选择可解除模式时，遵循以下设计指南：

- **当用户需要控制是否可解除时，使用人工模式**

如果你的应用要求有对不可选内容的交互，比如在内容区绘画，把轻量可解除取消而设置为人工模式。在这种模式下，应用的交互不会使应用栏被解除。无论命令被隐藏还是展示用户都能完成任务。

- **当用户需要随时可见应用栏时，使用固定模式**

如果有些命令对于用户完成体验而言时必须的，把这些命令放在固定模式应用栏上而不是内容区中。把应用栏设置为固定模式取代短暂而轻量的可解除模式。在这种模式下，工具栏是始终可见并且不能被用户解除的。

## 应用栏命令的设计指南

当把命令放到应用栏上时，遵循以下设计指南。

- **把命令放到应用栏右边**

一般来说，你应当把任何可用的，相关的应用放到应用的右边。

如果你有许多命令，并且可以在逻辑层面将其分为两部分，将其中一部分放到左边，另一部分放到右边。比如，如果你的应用可以分为总是出现的通用命令（比如主页以及视图）以及仅当内容相关联时出现的相关命令（比如裁剪和旋转），那么把通用命令当道应用栏的左侧，把场景相关的命令放到应用栏的右侧。

- **以编程方式在应用栏上展示上下文关联的命令**

如果命令只有在内容被选中时才能执行，那么当用户选中内容时以编程方式展示应用栏。把应用栏设置为固定模式，这样当用户在对内容进行操作时始终可以看见应用栏。

如果命令无论内容是否被选中都能执行，当有足够空间时把这些命令都放到应用栏上。当内容未被选中时，把应用栏隐藏了返回至轻量可解除模式。

- **当命令过多时使用菜单**

如果不能以独立按钮的形式在应用栏上放下全部的命令，把命令一起群组并把这些命令放到菜单中，这些菜单可通过应用栏上的按钮被打开。把这些命令在逻辑层面群组，比如把回复，回复全部以及转发放到应答菜单中。

不要把不相关的杂乱的命令放置在“更多”以及“高级选项”中。这种类型的普通命令使应用变的更加复杂，并且仅仅只有一小部分用户探索这个菜单。当发现你的应用需要更多额外的命令并且这些命令在逻辑层面都是不相关联的，考虑简化你的应用。

- **为分屏视图设计应用栏**

少量有限的命令可以当应用在分屏状态下适合应用栏。你可以把命令分组在一起放到菜单中或者以编程的方式适时提供额外的命令。

- **为横向的滚动设计**

当应用可横向滚动时，当心置于应用底部的应用栏会覆盖滚动条。用户可能为了使用滚动条需要点击应用使应用栏消失，或者他们可以使用鼠标滑动滚动条。

如果应用栏时固定模式，减少应用中可滚动区域的高度，滚动条始终明显地在应用栏的顶部边缘。

- **不要把决定性的命令放到应用栏上**

不要把用户完成任务所需要的必要命令放到隐藏的应用栏上。当用户为了完成任务而频繁的打开应用栏时，你应当考虑将该命令改为直接操作的，语义明显的，或者直接在内容区操作命令。

- **不要把文本剪切板命令放到应用栏中**

把剪切，复制命令放到上下文菜单中而不是应用栏上。



# 辅屏和全屏 ——设计规范与检查表

[本文档是初步的，并且有变更的可能。]

这个指南能帮助你创建一个在辅屏和全屏时外观和表现都良好的应用。

注意：辅屏只有在足够的水平分辨率下才可以使用（大于1366像素）。一个辅屏应用始终是一个标准的宽度，这个宽度以触摸独立像素来衡量，辅屏可以放置在屏幕的两侧。

# 辅屏与全屏 的用户体验规范

因为用户可以贴靠每一个应用，你应该给你的应用设计辅屏视图状态。如果你不这样做，系统将强行调整你的应用的尺寸，可能会裁切你的内容或者添加滚动条。

下面是您要遵循的一些主要原则：

• **维持状态并保持连续性**

即使在辅屏视图功能减少了，也保持状态。应该优先保持状态，而不是给用户一个更具吸引力但是断章取义的辅屏页面。

• **跨状态功能特性要对等**

记住，辅屏就像经典桌面下的Aero贴靠一样；用户期望在贴靠时能与你的应用交互。如果你能保持功能特性对等，我们建议包含一个进入功能特性的切入点和点击这个切入点能够自动取消贴靠。

• **为你的布局逻辑提供媒体查询**

这允许你的应用布局变化时作出适当反应。

• **使用应用布局接口**

这保证了你的应用在布局变化时候被通知，特别是在应用启动和从暂停状态重新启动的时候。

• **给用户控制感**

不要程序化的解除贴靠来引起用户注意。解除贴靠只有在用户尝试使用在贴靠情况下无法使用的功能时才被保留。如果你的应用所有的页面有贴靠视图，你根本不该程序化的去解除贴靠状态。

• **不要加入程序化解除贴靠的UI控件**

在应用之间的分割条时始终存在的，无论用户何时需要他都能取消贴靠。

记住，贴靠的应用是重新调整大小，它不是一个小工具或者最小化窗口。你要保持你的用户的状态，上下文和交互性。贴靠和解除贴靠不应该破坏用户的工作或者状态。



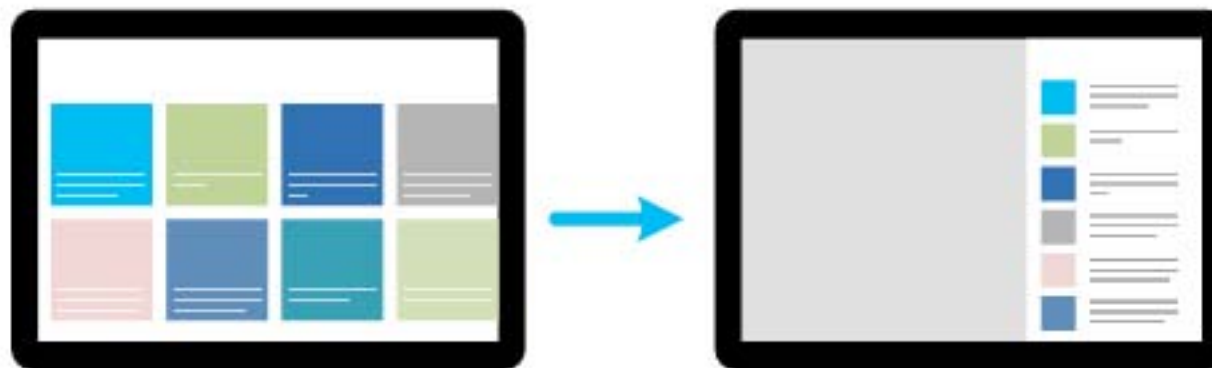
何时应该使用媒体请求，何时使用js布局变化事件？

用法：	驱动：
媒体请求	布局变化和属性是通过CSS样式特别指定： 元素尺寸 元素布局 ( inline , block ) 多种元素的可见性
JS事件	行为变化和属性不能被CSS样式特别指定： 列表视图控件的滚动方向 控件变化，如从列表按钮到单个的下拉选择控件

**贴靠布局**

**规划方向**

利用贴靠布局“又高又瘦”的宽高比的优势去垂直平移。如果你的应用在贴靠和全屏状态之间改变平移方向，请确保它在UI上是显而易见的。



**纵栏布局**

由于贴靠状态的窄宽度，我们推荐在贴靠时，过渡多栏布局到一个单栏布局。下面的例子显示了多栏布局到单栏的布局的变化。当应用贴靠时，布局元素堆叠成一个单栏的结构。

