Windows* Store App Features and Differentiators

Abstract

This article highlights the key features of Microsoft Windows* 8 operating system. It provides a summary of the differentiating features of Windows Store apps running on Intel® Core™ and Intel® Atom™ processors. One objective of this article is to serve as an introductory reference for application developers new to Windows 8.

Overview

Officially released in October 2012, Windows 8 revolutionizes the user interface over other versions of Windows. Some of the new compelling features introduced include:

- The Start Screen
- The ability to run across many different types of hardware devices, from tablets based on Intel Atom processors to Ultrabook™ systems with Intel Core processors
- Optimized for Touch while providing full keyboard and mouse support
- Apps from the Windows Store

On Windows 8, Windows Store apps can incorporate the following new UI/UX features, which we will describe in more detail later in this document:

- Live tiles
- Charms
- "Lock Screen" update
- App bar
- Snapped and fill view
- Semantic zoom

For application developers, these features can be implemented with the Visual Studio* 2012 and Windows Runtime APIs using familiar programming languages such as Visual C#*, JavaScript*, or Visual C++*.

Live Tiles

On the Windows 8 Start screen, installed applications are depicted using icons or "Tiles." Touching a tile or clicking on it with the mouse, launches the application.

On the Start screen, tiles can be presented in two sizes, a square (small) and a rectangle (large/wide). For applications that support both tile sizes, users can choose to display either. Tiles can either display a static image/icon or a dynamic "live" image that is updated via notifications from the application represented by the tile.

Tiles present rich content from the apps even when they are not the active application. When a user is on the Start screen, all applications that present a Live Tile, update the content while running in the background. For examples, a weather app can display your current local temperatures, and a finance app can show the current market snapshot.

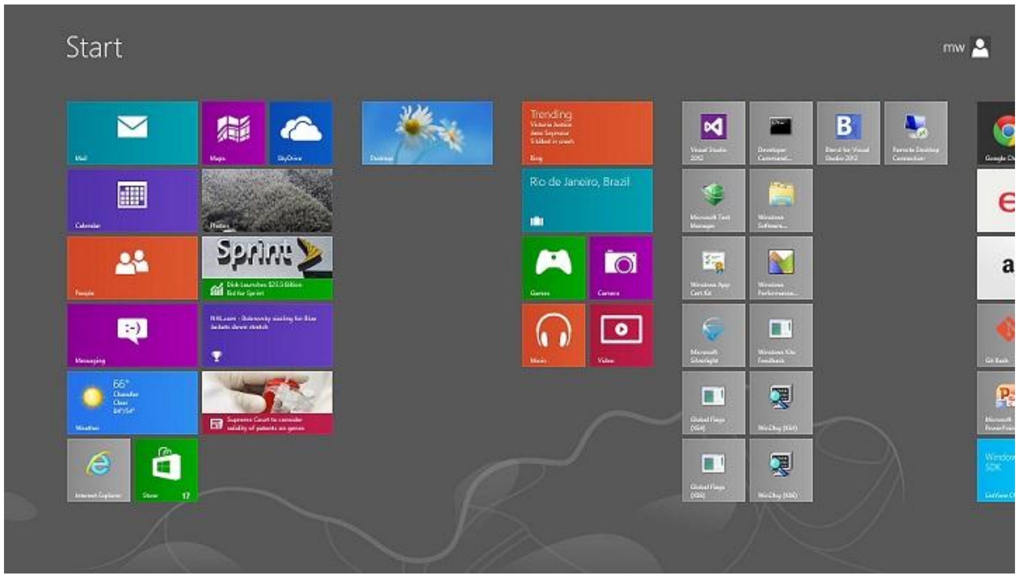

Figure 1 Tiles in Windows* 8 Start screen.

For application developers, a compelling live tile makes your app stand out on the Windows Start screen. Windows Runtime provides `Windows.UI.Notifications` APIs and a catalog of tile templates to implement tile updates. The following tutorials can help you get started:

- [Sending a tile update]()

- [The tile template catalog]()

Charms

On Windows 8, at anytime and anywhere, if you swipe from the right edge into the screen or mouse point on the upper right corner or lower right corner, a vertical bar will appear on the right side of the screen showing additional functions, i.e., charms. They include Search, Share, Start, Devices, Settings, and more. Charms can be customized to include extra behaviors inside a specific app beyond what is available from the Start screen.

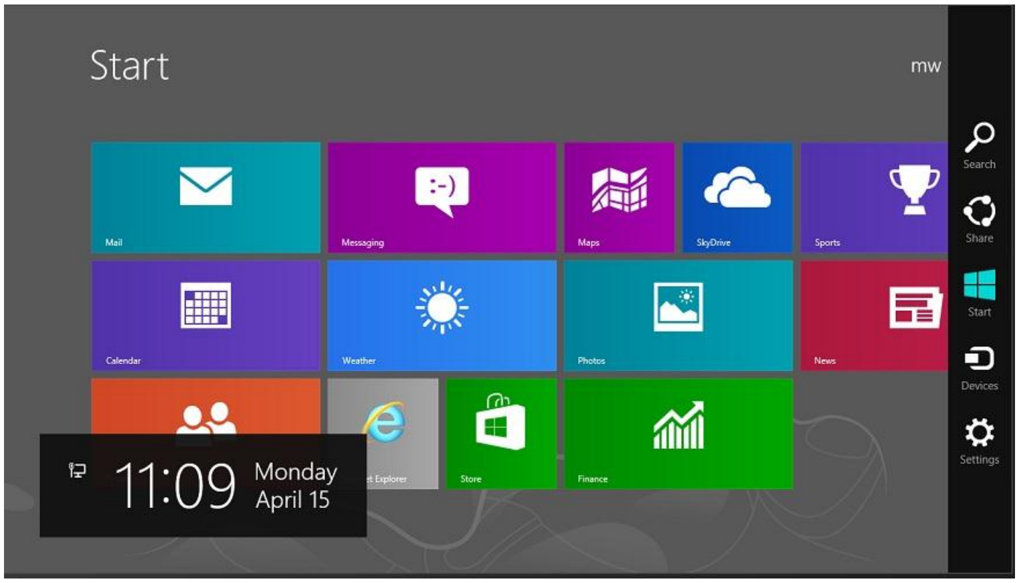

Figure 2 Charms appear on the right-side of the screen when the right edge swipe gesture is used.

We will discuss the functionality of each charm below.

Search

With the Search charm, you can search for apps, files, and other items on your system locally. You can also search for things on the Internet using a specific app or service, such as Bing*.

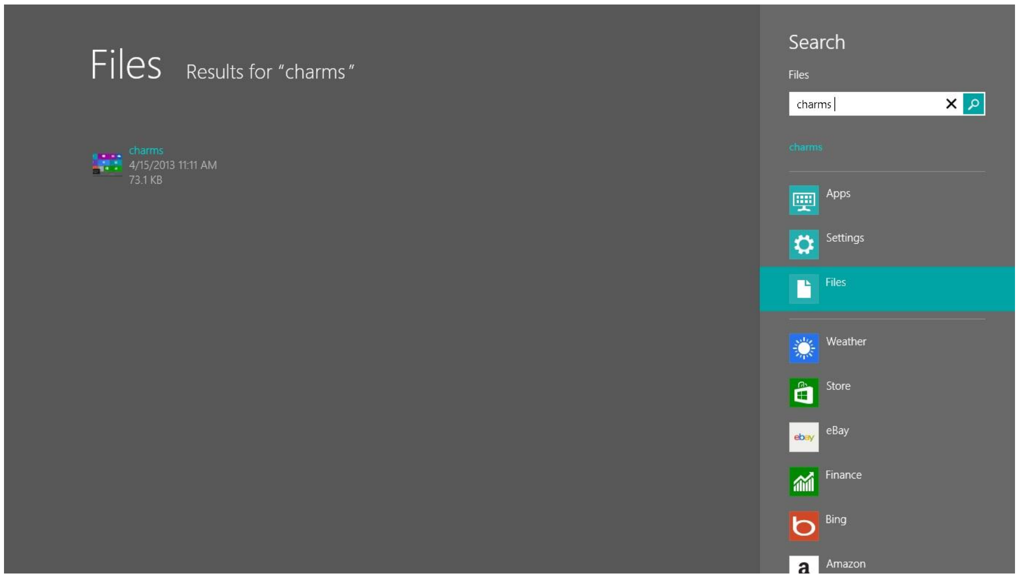

Figure 3 The Search charm.

In Windows Store apps, you can let users search within your app by adding the Search contract. The following tutorial will get you started:

- [Adding search to an app]()

Share

Within a specific app, you can quickly share files, photos, web links, or other information with other people by using the Share charm. It will display a list of apps or services you can share with.

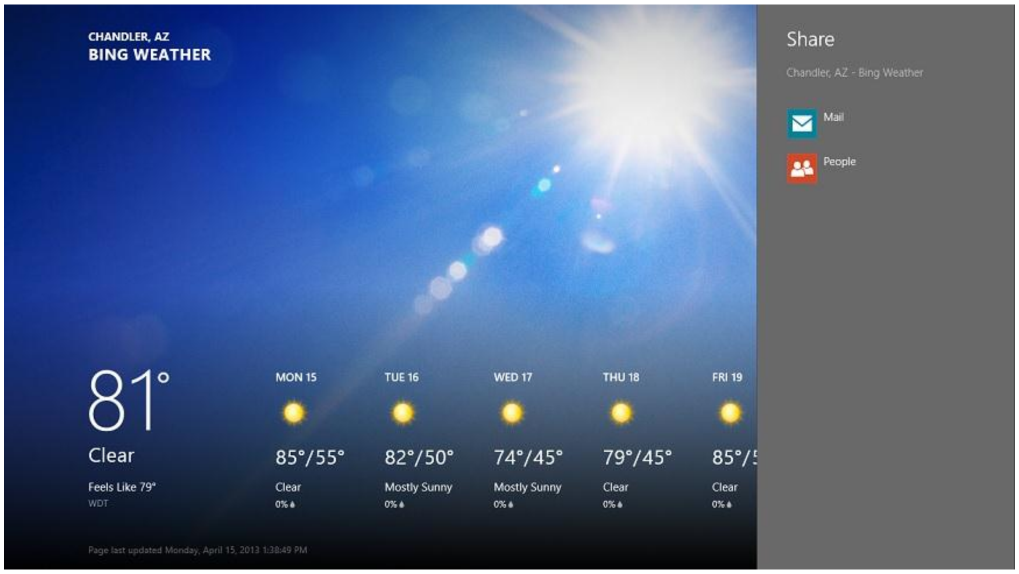

Figure 4 The Share charm under Bing* Weather let you share the current local weather through email and other social networking apps

The `Windows.ApplicationModel.DataTransfer` namespace includes the basic classes and APIs you need to enable sharing in your Windows Store apps. The following tutorial has more information:

- Sharing content

Start

The Start charm allows you to quickly go back to the Start screen from any app. If you are already in the Start screen, pressing or clicking the Start charm will go back to the last app you used.

Devices

The Devices charm is mainly used to set up connections with external devices, such as printers, displays, or wireless TVs.

Settings

You can use the Settings charm to personalize your PC and customize a specific app.

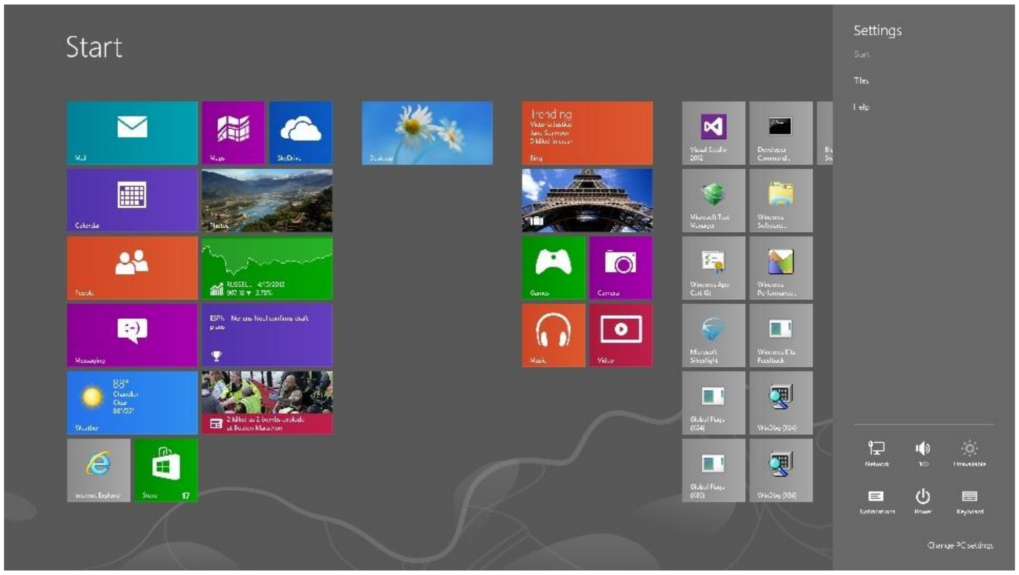

Figure 5 The Settings charm allows you to personalize the PC

Windows Store app developers can use `Windows.UI.ApplicationSettings` class to implement the Settings contract and to add app settings:

- [Add app settings]()

In summary, we discussed the usages of charms in Windows 8 and provided links on how to implement charms in Windows Store apps.

App Bar

Most desktop operating systems users are familiar with the Menu bar and the Tool bar that allow access to various functions/actions supported by the application. Typically, with current desktop operating system apps, even though the Menu and Tool bars are not needed all the time while interacting with the app, the control bars permanently occupy the application's UI real estate and can be distracting.

In Windows 8 Store apps, to embrace the immersive and full-screen design fundamentals, the menu options and commands are no longer permanently displayed while the user interacts with the app, but are instead part of the App bar. They no longer permanently occupy the app's valuable UI real estate. Both the app bar and charms, discussed in the previous section, are "UI on demand." They only show up when the user requests them.

App bars can be at the top of the screen, at the bottom of the screen, or both. The user can swipe into the screen from either the top edge or the bottom edge, or right click within the screen to display the app bars. Within a screen, the user swipes or right clicks to select items, the app bar responds by showing options and commands valid for the currently selected item(s).

While charms expose standard functionality and operation supported by all the installed apps, application specific items can be included in the app bar based on the app's context. They can be different across apps, across the screens within an app, and across different items selected on a screen.

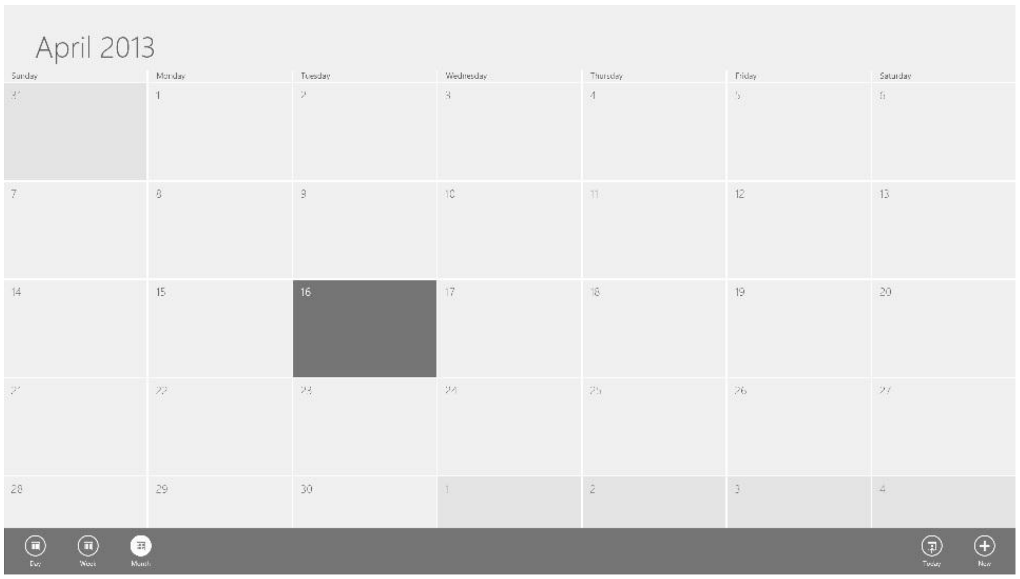

Figure 6 App bar for a Calendar app

Windows Store app developers can use XMAL's AppBar control to easily add an app bar. Please refer to the following tutorial for more information:

- Adding app bars

Lock screen apps

By default, the Windows 8 lock screen always shows some basic system information, such as date, time, network status, and battery level. When the device is in a locked state, Windows 8 also allows up to seven apps to run in the background and display Badges and Toasts on the lock screen. In addition, one of those apps is allowed to show its latest tile notification text. Users can configure which apps show their status and notifications from the Settings charm / Change PC settings /Personalize / Lock screen.

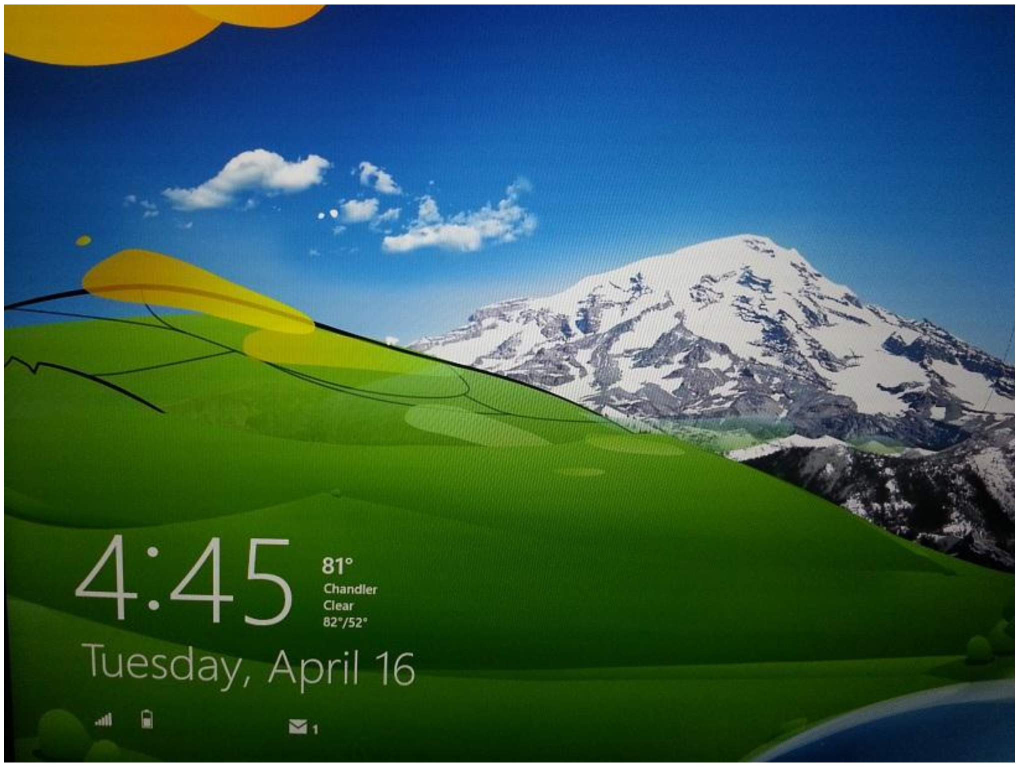

Figure 7 A lock screen shows the Email app and the weather app's status and notifications. The weather app shows the tile notification text.

As Windows Store app developers, you can enable your apps to show tile and badge updates on the lock screen by following the tutorial:

- [Showing tile and badge updates on the lock screen]()

Snapped and Fill Views

For devices with a horizontal resolution of 1366 relative pixels or greater, Windows 8 allows users to interact with two applications simultaneously. Both applications run side by side on the screen, where one application occupies ¾ of the screen (Fill View), and the other occupies the remaining ¼ of the screen (Snapped View). The snapped view app can be on either the left or right side of the screen while the Fill View occupies the remainder of the screen along with the divider. The divider can be dragged to the left or the right to switch the application from Filled View to Snapped View and vice versa.

Snapped view mode can be invoked via a "swipe and hold" gesture with a finger or mouse pointer by swiping from the top of the screen towards either the left or right side of the screen until a divider appears. The app can then be "dropped" in the smaller snapped region by releasing the mouse or by lifting the finger off of the screen.

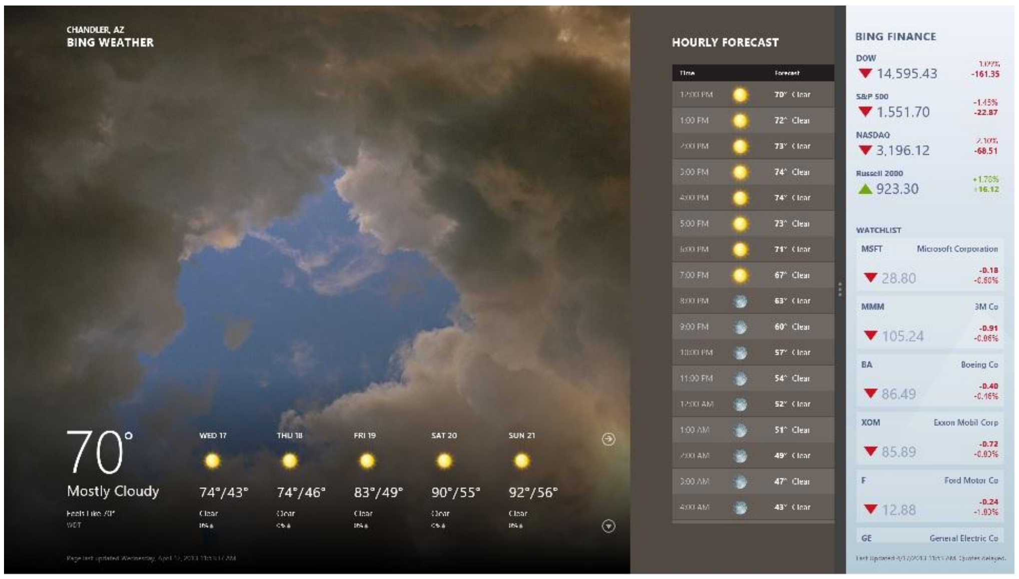

Figure 8 The Bing* Finance app is running in the snapped view mode, while the Bing Weather app is running in the fill view mode

To make the Windows Store apps behave properly when running in the snapped view mode, you must adhere to the following guidelines:

- Guidelines for snapped and fill views

Semantic Zoom

People who have experiences with touch-screen devices that support multi-touch are familiar with using the pinch gestures to zoom in and zoom out to enlarge or shrink an image, a web page, or a map view. Semantic Zoom extends the zoom concept in Windows Store apps by allowing zoom to operate on other data.

To invoke the semantic zoom, the user uses the pinch gesture, or Ctrl key while scrolling the mouse scroll wheel.

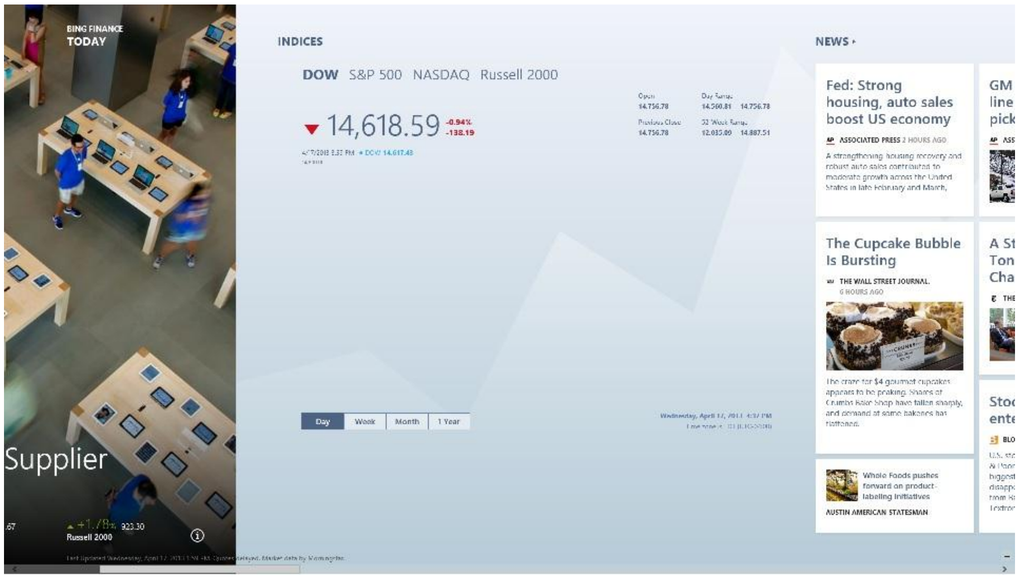
BING FINANCE
TODAY
Supplier
Russell 2000
INDICES
DOW S&P 500 NASDAQ Russell 2000
14,618.59
-0.94%
-138.19
Day Week Month 1 Year
NEWS
Fed: Strong housing, auto sales boost US economy
ASSOCIATED PRESS 2 HOURS AGO
The Cupcake Bubble Is Bursting
THE WALL STREET JOURNAL
Whole Foods pushes forward on product-labeling initiatives
AUSTIN AMERICAN STATESMAN

Figure 9 The zoomed-in mode (normal mode) of the Bing* Finance app.

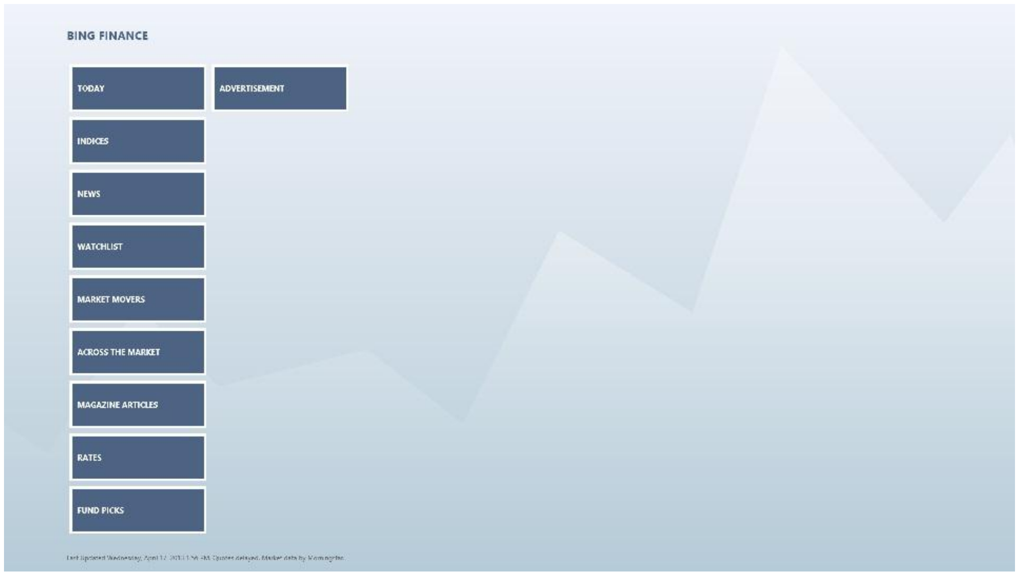

Figure 10 The zoomed out mode of the Bing* Finance app.

To support semantic zoom, the application should organize and present data or information in two distinct modes: one low-level (or zoomed-in) mode that displays the normal data or information, and one high-level (or zoomed-out) mode that displays a summarized view with data or information grouped and categorized. From the zoomed-out mode, users can quickly jump to the detailed section they want to access.

An example of semantic zoom is an app that primarily displays a list of the customers. The zoomed-out mode would show a list of states customers reside in.

To support Semantic Zoom in the Windows Store apps, follow these guidelines:

- Guidelines for Semantic Zoom

Development Tools

To develop Window Store apps, application developers can use Visual Studio 2012 and the languages they may have learned and used before: Visual C#, Visual C++, HTML5/CSS, JavaScript, Visual Basic*, etc. Visual Studio 2012 is also integrated with the popular and powerful design tools such as Blend.

- Develop Windows Store apps using Visual Studio 2012

Improved User Experiences on Intel® Architecture-based Devices

On Intel® Architecture-based systems, application developers can utilize some features available on Intel® processors. They can use WiDi in Windows 8 Desktop apps to provide a premium user experience and unique usage models:

- Download Intel® WiDi Extensions SDK

Developers can also integrate Perceptual Computing SDK features such as gesture recognition, voice recognition, and augmented reality to create new app experiences:

- Intel® Perceptual Computing SDK

Summary

In this article, we discussed the differentiating features in Windows 8 and Windows Store apps. We also provided links to tutorials on how to use these features.

References and Resources

Comment [LP1]: All of the bullet formatting is "off." IDZ please fix it.

- Resources for iOS* developers
  http://msdn.microsoft.com/en-us/library/windows/apps/jj680134

- Porting iOS Apps to Windows 8 (1) : Introducing Windows 8 Platform to iOS App Developers
  http://blogs.msdn.com/b/win8devsupport/archive/2012/11/14/porting-ios-apps-to-windows-8-1-introducing-windows-8-platform-to-ios-app-developers.aspx

- Porting iOS Apps to Windows 8 (2) : User Interface
  http://blogs.msdn.com/b/win8devsupport/archive/2012/11/16/porting-ios-apps-to-windows-8-2-user-interface.aspx

- Case studies (Windows Store apps) - Some good guidelines for UI design
  http://msdn.microsoft.com/en-us/library/windows/apps/hh770549.aspx

- A good starting point for developers to become familiar with the tools and features of programming Windows Store apps:
  - Getting started with Windows Store apps (Windows)
    http://msdn.microsoft.com/en-us/library/windows/apps/br211386.aspx

- Some links that dig deeper into the programming details of Windows Store apps:
  - Developing Windows Store apps (Windows)
    http://msdn.microsoft.com/en-us/library/windows/apps/xaml/BR229566
  - Learn to build Windows Store apps
    http://msdn.microsoft.com/library/windows/apps/
  - Introduction to creating Windows Store apps using XAML – BUILD video
    http://channel9.msdn.com/Events/Build/2012/3-116

- A summary page with multiple links on articles with details about writing multimedia Windows Store apps:
  - Adding multimedia (Windows Store apps using C#/VB/C++ and XAML) (Windows)
    http://msdn.microsoft.com/en-us/library/windows/apps/xaml/hh465134.aspx
  - Designing UX for apps
    http://msdn.microsoft.com/library/windows/apps/hh779072/
  - Quickstart: Touch input (Windows Store apps using C#/VB/C++ and XAML)
    http://msdn.microsoft.com/en-us/library/windows/apps/xaml/hh465387.aspx

  - Quickstart: Creating a media player app (Windows Store apps using C#/VB/C++ and XAML)
  - http://msdn.microsoft.com/en-us/library/windows/apps/xaml/hh986967.aspx

- Various Intel® Developer Zone articles about programming for Windows 8 written by the Intel team:
  - Windows* Style app development: Using Window Runtime from C#
    http://software.intel.com/en-us/articles/windows-8-metro-style-app-development-using-winrt-from-c

  - Enabling Touch in Windows 8* Style UI Apps with C#
    http://software.intel.com/en-us/articles/enabling-touch-in-windows-8-metro-style-apps-with-c

  - Porting iOS Apps to Windows 8 - Overview
    http://software.intel.com/en-us/articles/porting-ios-apps-to-windows-8-overview

  - Porting App Life Cycle, Settings and Search features from iOS to Windows 8
    http://software.intel.com/en-us/articles/porting-app-life-cycle-settings-and-search-features-from-ios-to-windows-8

  - Porting the User Interface from an iOS Basic App to Windows* 8
    http://software.intel.com/en-us/articles/porting-the-user-interface-from-an-ios-basic-app-to-windows-8

  - Porting Advanced User Interfaces from iOS to Windows* 8
    http://software.intel.com/en-us/articles/porting-advanced-user-interfaces-from-ios-to-windows-8

  - Porting iOS Custom Charts and Graphs to Windows* 8 UI
    http://software.intel.com/en-us/articles/porting-ios-custom-charts-and-graphs-to-windows-8-ui

- In addition to the articles above, samples of Windows Store Apps are a great resource, and you can find some here:
  http://code.msdn.microsoft.com/windowsapps/

- If you have some previous exposure to XAML, this article addresses some of the differences with programming for Windows Store apps.

Porting Silverlight* or WPF XAML/code to a Windows Store app (Windows)

http://msdn.microsoft.com/en-us/library/windows/apps/xaml/br229571.aspx

- BUILD videos are another great resource that are easy to search and find what you are looking for:
  http://channel9.msdn.com/Events/Build

- The last recommendation is to sign up for the developer forums on the Windows 8 site. If any technical questions or API problems are encountered, the question might have already been discussed/answered there, or it is easy to start a new thread to get the question answered:
  http://msdn.microsoft.com/en-US/windows/apps/br229515/

- Other Reference articles and blogs:
  http://apimappingsite.cloudapp.net/library?source=iOS#iOS-iOS

  http://msdn.microsoft.com/en-us/library/windows/apps/hh974578.aspx

  http://blog.bignerdranch.com/1010-windows-8-for-ios-developer/

About the Author

Miao Wei is a software engineer in the Intel Software and Services Group. He is currently working on scale-enabling projects for Intel Atom processors.

Notices

Optimization Notice