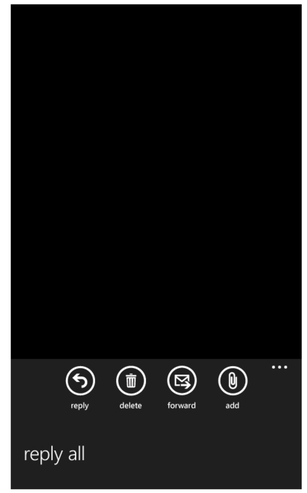
application bar & application menu

The application bar provides a place to promote the most common tasks and views of the application. For less common tasks, applications can include a single menu launched from the application bar. An Application Menu is not required, but if you have one it is associated with the application bar.

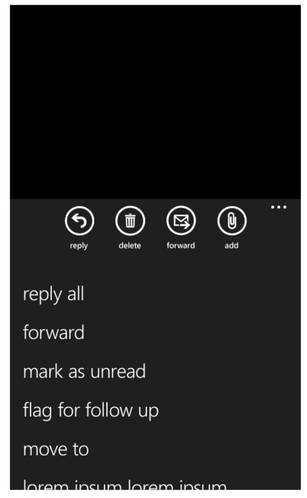
COMMON CONTROLS | **APPLICATION BAR & APPLICATION MENU** | OVERVIEW



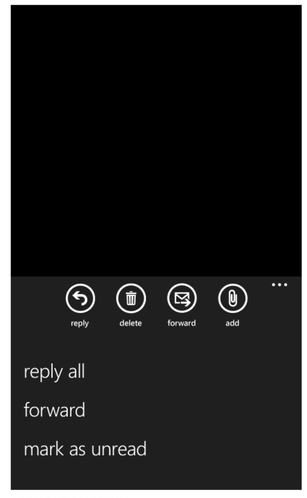
APPLICATION BAR
Shown with icon buttons



APPLICATION MENU
Showing menu sized for 1 overflow task



APPLICATION MENU
Showing maximum viewable tasks in scrollable region



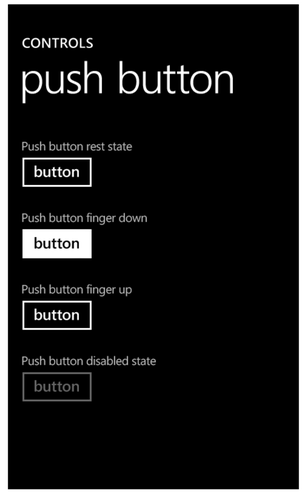
APPLICATION MENU
Showing menu sized for 3 overflow tasks

buttons

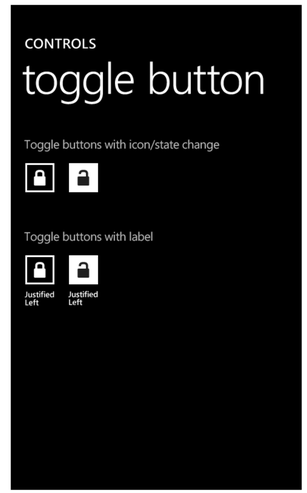
A push button causes an application to perform an action when the user taps on it.

Button styles
1. Standalone Push Buttons
2. Toggle Buttons
3. Application Bar Icon Buttons
4. Dialog Buttons
5. Command Buttons

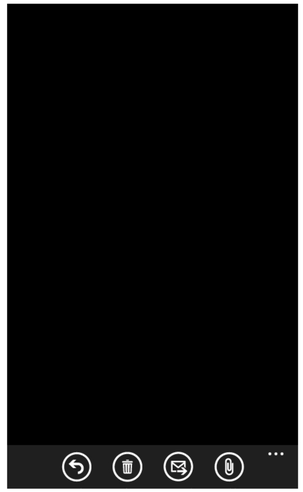
COMMON CONTROLS | **BUTTONS** | OVERVIEW



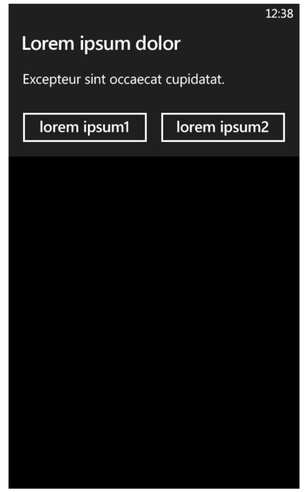
1. STANDALONE PUSH BUTTONS
There is a minimum width for finger accessibility



2. TOGGLE BUTTONS
Optional image states, labels and sticky behavior



3. APPLICATION BAR ICON BUTTONS
Must be clear, simple and familiar



4. DIALOG BUTTONS
1-2 fixed size buttons with primary action on left

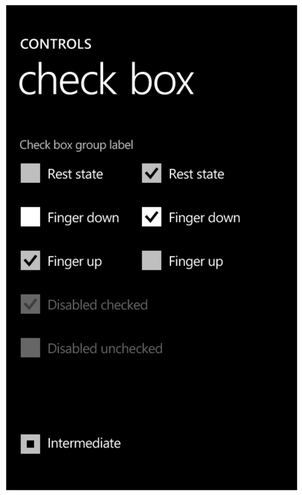


5. COMMAND BUTTONS
Used only for verbose explanation of the action/task

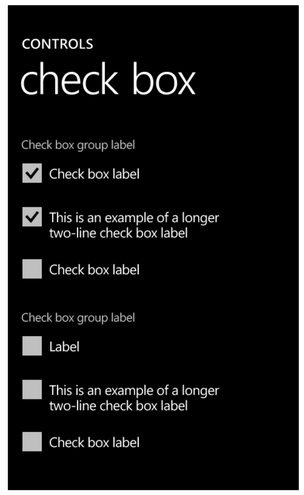
check box

You can use the check box control in groups to display multiple choices from which the user can select one or more. The check box control is similar to the radio button control in that each is used to indicate a selection that is made by the user. They differ in that only one radio button in a group can be selected at a time. With the check box control, however, any number of check boxes may be selected. Check box labels can wrap to two lines of text.

COMMON CONTROLS | **CHECK BOX** | OVERVIEW



CHECK BOX
Touch interaction states



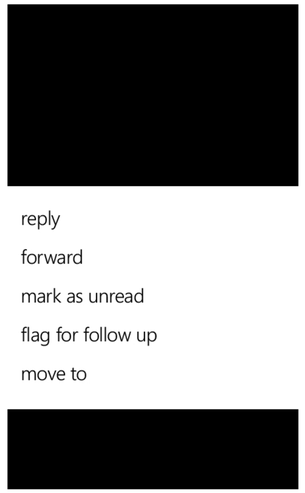
CHECK BOX LAYOUT
Shown with group labels to create hierarchy

context menu

The context menu offers an additional way for the user to access specific tasks. They are intended to be a way to accelerate a user's workflow, and should never be the only way to accomplish a specific task. The context menu should be used to express secondary or advanced actions to the user, never primary tasks, and may have no more than five items.

This control is not shipped as a part the Windows Phone Developer Tools. The templates and instructions are provided as-is to assist designers and developers to match, visually and in interaction, a similar control that appears in Windows Phone OS 7.0.

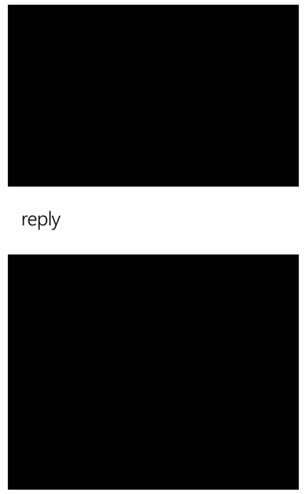
COMMON CONTROLS | **CONTEXT MENU** | OVERVIEW



CONTEXT MENU
Shown at largest size (5 items)



CONTEXT MENU
Shown at medium size (3 items)



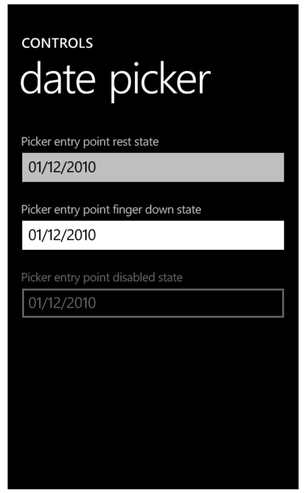
CONTEXT MENU
Shown at smallest size (1 item)

date picker

A date picker is used to select a specific date. To access the date picker, the user taps on a button which contains a date preselected by the application. Date picker selects only one date.

This control is not shipped as a part the Windows Phone Developer Tools. The templates and instructions are provided as-is to assist designers and developers to match, visually and in interaction, a similar control that appears in Windows Phone OS 7.0.

COMMON CONTROLS | **DATE PICKER** | OVERVIEW



DATE PICKER
Entry points and touch interaction states



DATE PICKER
Landing screen



DATE PICKER
Showing expanded month column

dialog boxes

A dialog box is a partial screen UI and is a way of communicating important information within a particular application to users. Dialog boxes should rarely be used for system wide interruptions.

Application Design Considerations
- Dialog boxes will cover up the top portion of the application UI
- Dialog boxes have a minimum and maximum height
- Dialog boxes can display 0-2 buttons
- All dialog boxes are cancellable by the back button
- Avoid nesting dialog boxes

Dialog Boxes may contain:
- Check box
- Radio button
- Link control
- Push button
- Toggle button

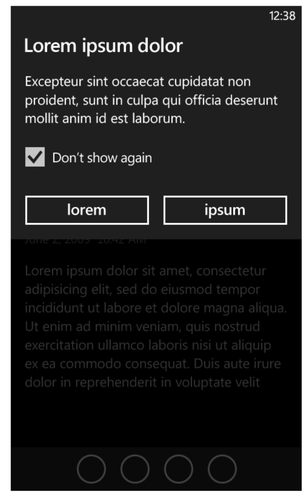
COMMON CONTROLS | **DIALOG BOXES** | OVERVIEW



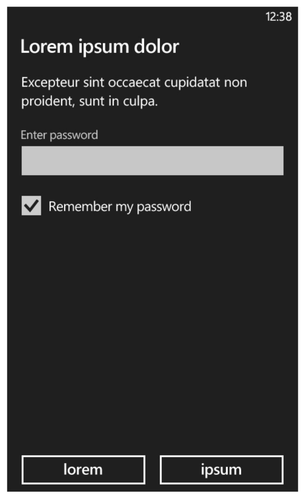
MESSAGE BOX
Shown at minimum height with all elements



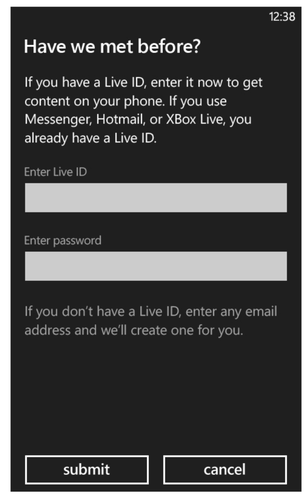
MESSAGE BOX
Showing wrapped, double-line header text



MESSAGE BOX
Shown with inline check box control



DIALOG WITH TEXT INPUT
Showing a password entry field and check box



CUSTOM DIALOG
Robust dialog with login/password fields



DIALOG WITH COMMAND BUTTONS
Command buttons handle verbose text

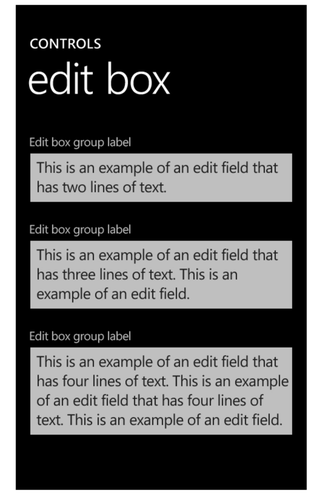
edit control

An edit control accepts user input. When an edit control is activated, the caret is shown and the on-screen keyboard is deployed. The edit control becomes active upon user tap or activation by the application. Applications should avoid activating the edit control by default when a user navigates to the page. Exceptions are allowed, such as a phone number field activated when user navigates from another page that prompted user to update their phone number.

Use a label above the edit control when the user needs to disambiguate between multiple edit controls.

Use placeholder text when instructional text is useful and there is no need for disambiguation that requires labels.

COMMON CONTROLS | **EDIT CONTROL** | OVERVIEW



MULTI-LINE EDIT BOX
Can be a fixed size or grow dynamically



EDIT BOX
Touch interaction states

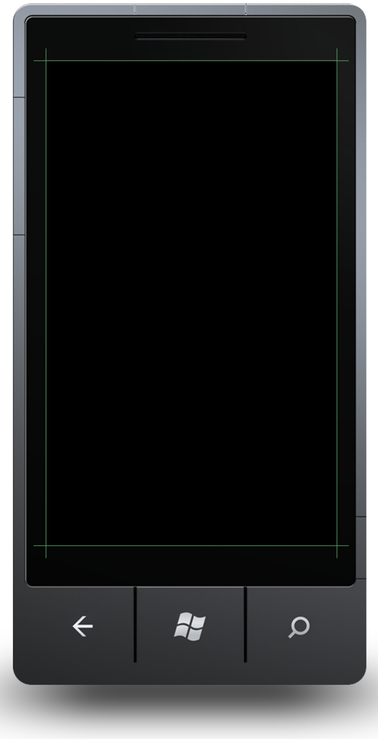


CHAT/SMS/MMS BUBBLE
Custom style with additional PNG bitmap "tail"

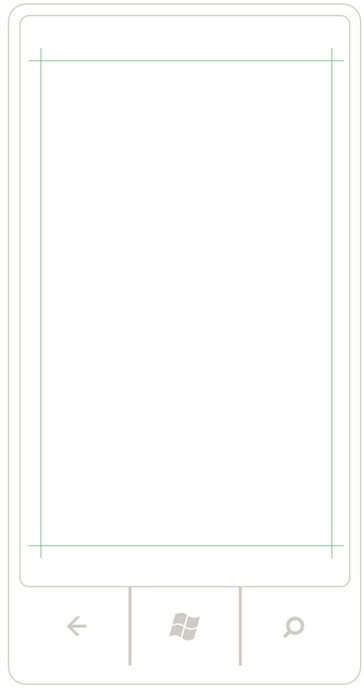
hardware

Start is located in the center of the device, below the display
Back is located to the left of start
Search is located to thte right of start

FRAMEWORK | **HARDWARE** | OVERVIEW



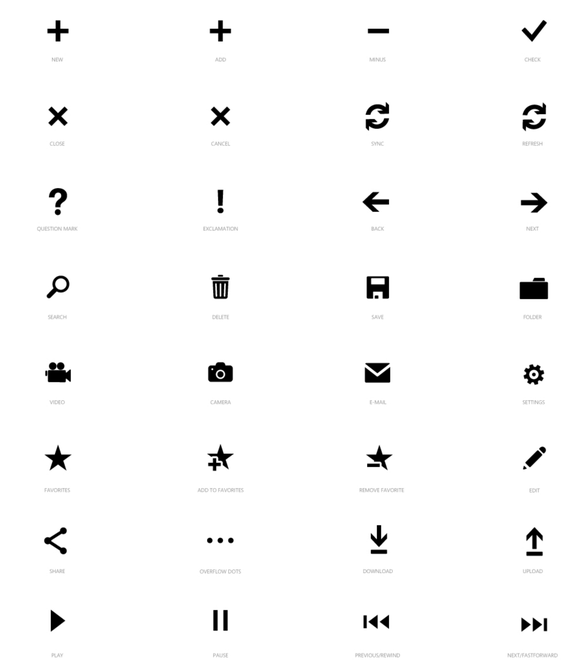
HARDWARE MOCKUP
Rendered style

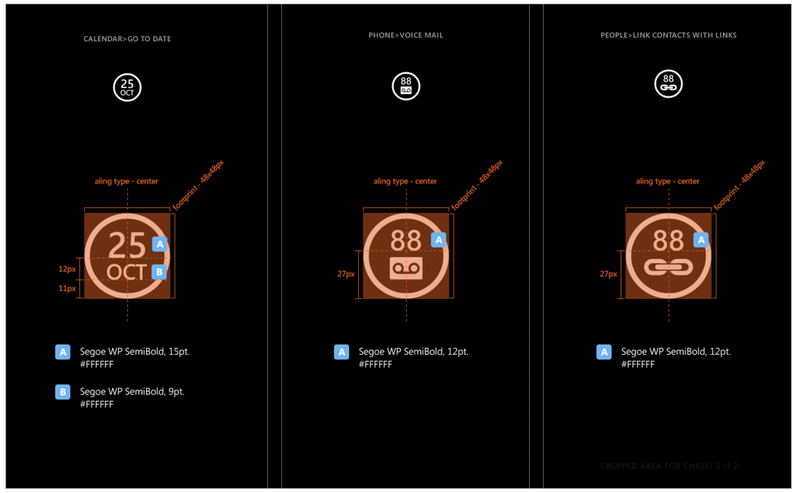


HARDWARE MOCKUP
Wireframe style

icons

Icons should be clear and understandable, using real-world metaphors that are familiar to users. The best icons are monochromatic with simple geometry.

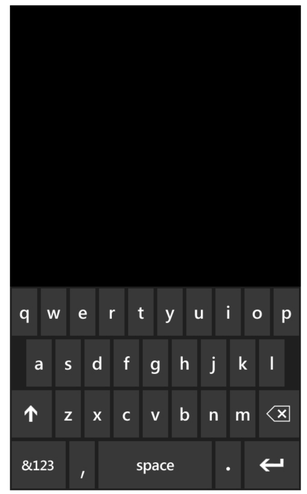




keyboard & input

The on-screen keyboard is used to input text on a touch screen and is deployed automatically when an edit control becomes active. The on-screen keyboard closes automatically when the user taps outside of the edit control or presses the back button. There are seven different on-screen keyboards available, depending upon input scope.

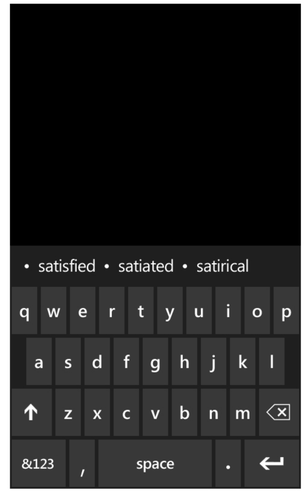
UX FRAMEWORK | **KEYBOARDS & INPUT** | OVERVIEW



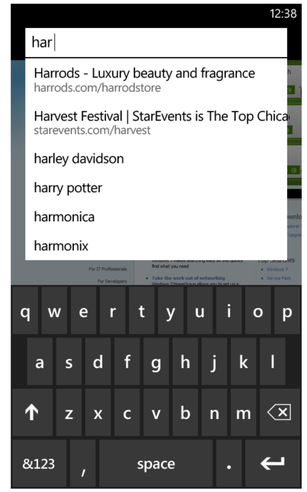
DEFAULT KEYBOARD
Shown as lowercase English



DEFAULT KEYBOARD
Shown as lowercase English in light theme



CANDIDATE WINDOW
Used to expose word suggestions for faster typing



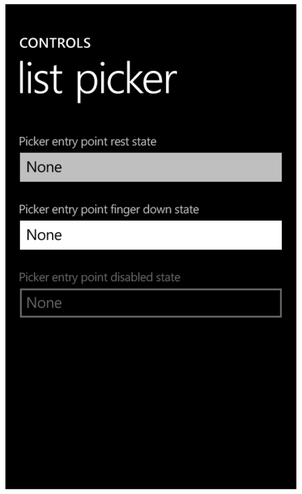
AUTOSUGGEST WINDOW
Offers suggestions as characters are typed

list picker

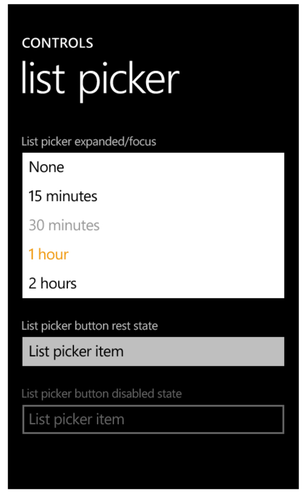
A single inline list picker lets the user pick a single item from a list of potential options. The user launches the picker from one of the entry points. Once the picker expands, the user taps on an item to select it. The list picker is then closed and takes the user back to the previous screen. An inline list picker can have up to 5 items.

This control is not shipped as a part the Windows Phone Developer Tools. The templates and instructions are provided as-is to assist designers and developers to match, visually and in interaction, a similar control that appears in Windows Phone OS 7.0.

COMMON CONTROLS | **LIST PICKER** | OVERVIEW



INLINE LIST PICKER
Entry points and touch interaction states



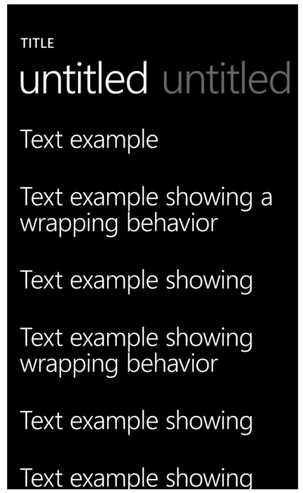
INLINE LIST PICKER
Shown expanded with disabled and preselected items

list view

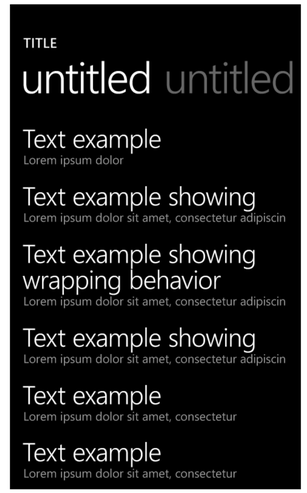
The list view displays set of list items. The list view presents views of data and images. It does this by giving application developers the ability to display this data in a vertical list or a grid. There are a set of standard layouts for the list view, however applications may opt to provide a richer presentation for some scenarios by using a custom layout. When this control is used, the list view typically covers the majority of the screen.

This control is not shipped as a part the Windows Phone Developer Tools. The templates and instructions are provided as-is to assist designers and developers to match, visually and in interaction, a similar control that appears in Windows Phone OS 7.0.

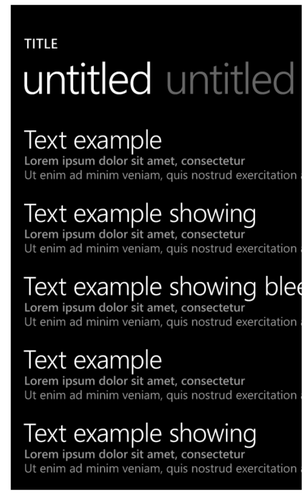
COMMON CONTROLS | **LIST VIEW** | OVERVIEW



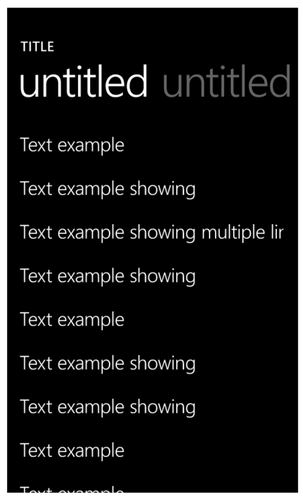
LIST VIEW
Showing a single line list with wrapping



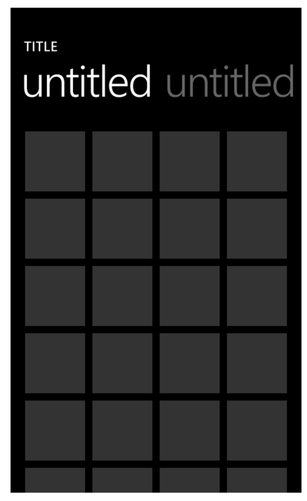
LIST VIEW
Showing a double line list with wrapping



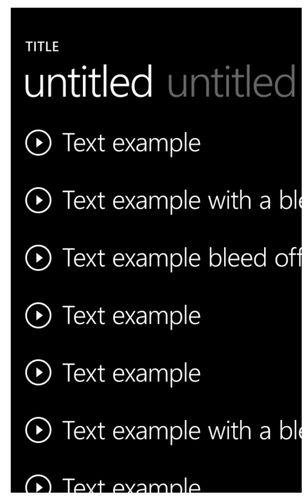
LIST VIEW
Showing a triple line list with no wrapping



LIST VIEW
Showing small text size with wrapping



GRID VIEW
Shown as small size grid - 4 columns



LIST VIEW
Shown with preview icon

panorama

Microsoft will provide design templates, guidance and controls around building panoramas at a later date. Microsoft recommends that designers and developers wait to implement panoramas until official guidance is available.

This control is not shipped as a part the Windows Phone Developer Tools. The templates and instructions are provided as-is to assist designers and developers to match, visually and in interaction, a similar control that appears in Windows Phone OS 7.0.

COMMON CONTROLS | **PANORAMA** | OVERVIEW

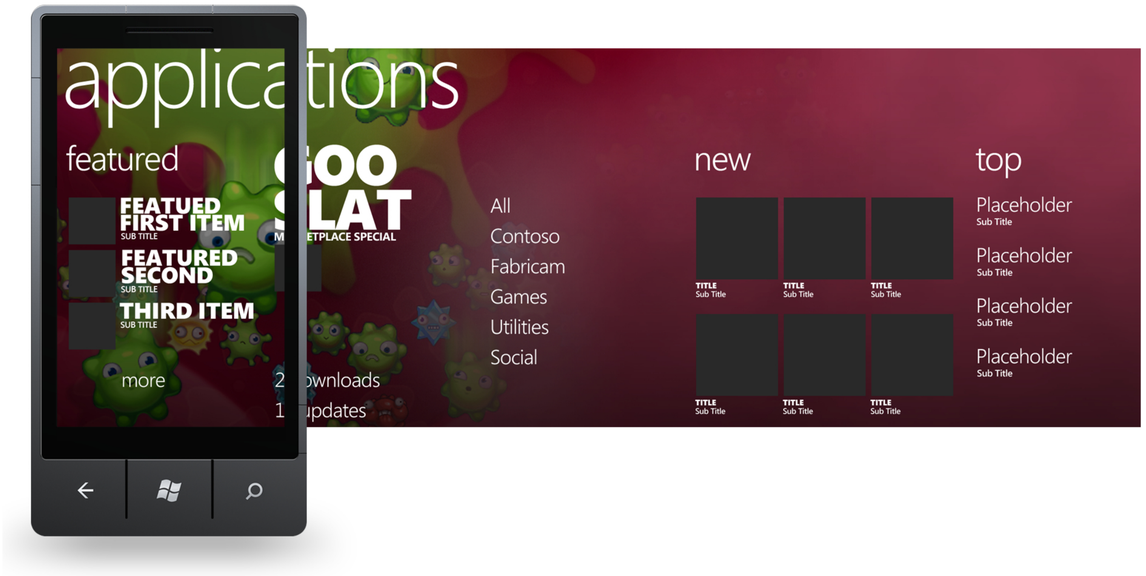


panorama backgrounds

Panorama background images are optional. If you use one, it should be selected with care to ensure that is does not degrade the ability of the user to discriminate touch targets from the image.

This control is not shipped as a part the Windows Phone Developer Tools. The templates and instructions are provided as-is to assist designers and developers to match, visually and in interaction, a similar control that appears in Windows Phone OS 7.0.

COMMON CONTROLS | **PANORAMA BACKGROUNDS** | OVERVIEW

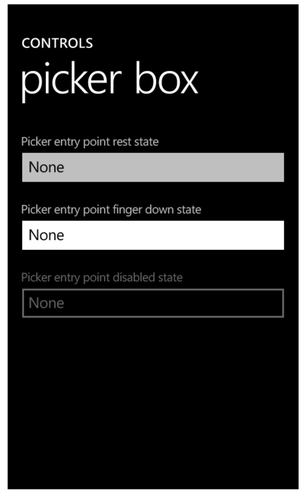


picker box

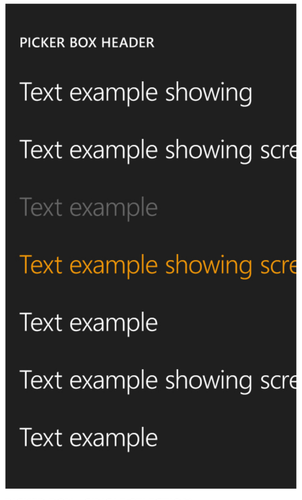
Picker box is a versatile control that can be used to select either a single item or multiple items. Picker box can be launched from a push button, a menu item, or application bar. There are three main types of this control: a single item select, multi item select, and single and multi item select with preview.

This control is not shipped as a part the Windows Phone Developer Tools. The templates and instructions are provided as-is to assist designers and developers to match, visually and in interaction, a similar control that appears in Windows Phone OS 7.0.

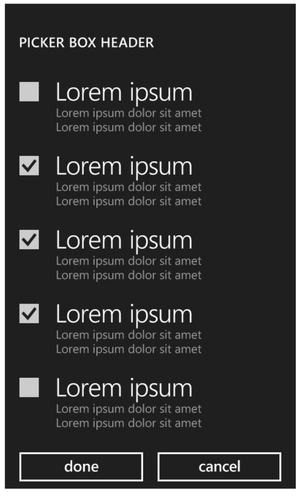
COMMON CONTROLS | **PICKER BOX** | OVERVIEW



PICKER BOX
Entry points and touch interaction states



PICKER BOX - SINGLE SELECTION
Shown with disabled and preselected items



PICKER BOX - MULTI SELECT
Requires a done and cancel button

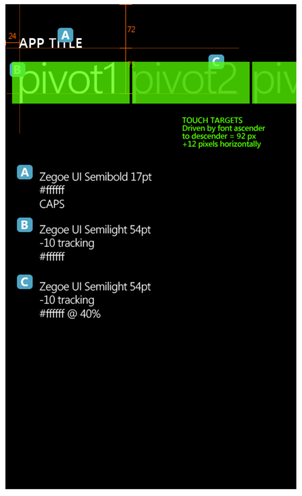
pivot control

A pivot control provides a quick way to manage views within the application. Pivot controls can be used for filtering large datasets, viewing multiple data sets, switching application views, or related content. The pivot control places them horizontally next to each other, and manages the left/right navigation (flick/pan) of the individual pages. Gesturing left and right on the page advances the pivot experience regardless of what controls are on the page.

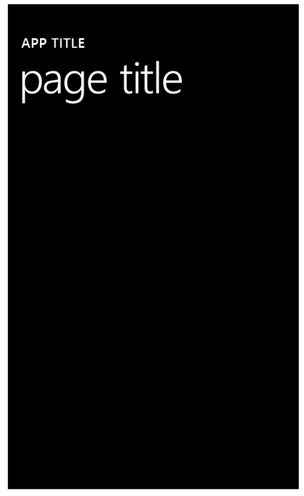
The visual structure can also be used in a static manner where multiple views aren't exposed.

This control is not shipped as a part the Windows Phone Developer Tools. The templates and instructions are provided as-is to assist designers and developers to match, visually and in interaction, a similar control that appears in Windows Phone OS 7.0.

COMMON CONTROLS | **PIVOT CONTROL** | OVERVIEW



PIVOT CONTROL
Shown with application title and dynamic pivot control



PIVOT STYLE HEADER
Shown with application title and page sub-title

progress indicators

Applications should use their content as the primary indicator of progress whenever possible. The progress indicator shows in-application activity related to an activity or series of events. Progress indicators may be custom built into the application design, however a set of standard progress indicators are recommended for general usage.

The wait indicator is an exceptional case. It is a blocking visual animation typically reserved for cases where the application cannot continue until an event is satisfied. Scenarios must minimize any waiting or break in the user task flow as much as possible.

The progress indicator may be either determinate or indeterminate:
- Determinate has a beginning and end point.
- Indeterminate continues until the task is finished. Indeterminate may appear as either a bar or wait indicator.

Progress indicator placement and behavior:
- The progress indicator overlays the application UI, flush with the top of the screen.
- The progress indicator is associated with the current application; when the user switches to a different application that is not actively using the progress indicator, the progress indicator will not be shown.
- Direct interaction is not possible with the progress indicator; applications must present their own UI to halt an operation.
- Users must be able to interact with the application while the progress indicators are active.
- Wait indicators should rarely ever be used as they completely block user interaction.

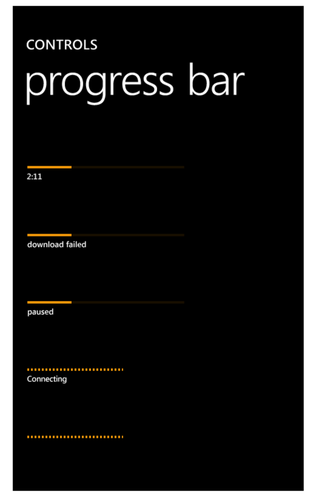
COMMON CONTROLS | **PROGRESS BARS** | OVERVIEW



PROGRESS INDICATOR
Showing determinate progress without label



PROGRESS INDICATOR
Showing indeterminate progress without label



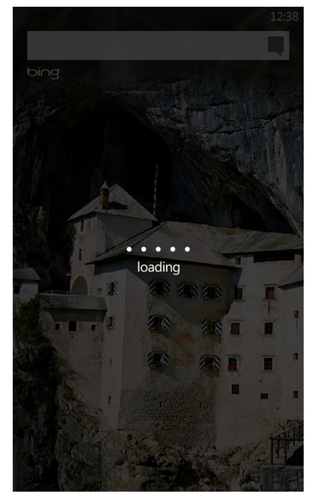
INLINE PROGRESS BARS
Used in association with specific content on the page



PROGRESS INDICATOR
Showing determinate progress with label



PROGRESS INDICATOR
Showing determinate progress with label



WAIT CURSOR
Wait cursor takes over and dims the screen

quick jump grid

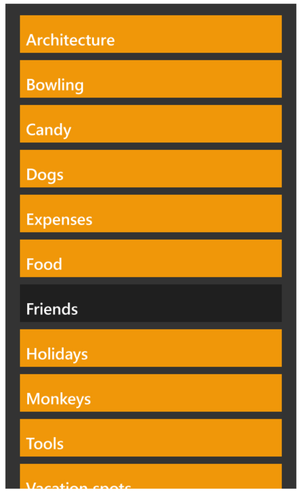
The quick jump grid is a way to help the user scroll through long lists. The quick jump grid can be an alphabetical list, a set of group headers or dates. The quick jump grid overlays the list, an item is selected, and the user is then taken back to the long list at the point they selected.

This control is not shipped as a part the Windows Phone Developer Tools. The templates and instructions are provided as-is to assist designers and developers to match, visually and in interaction, a similar control that appears in Windows Phone OS 7.0.

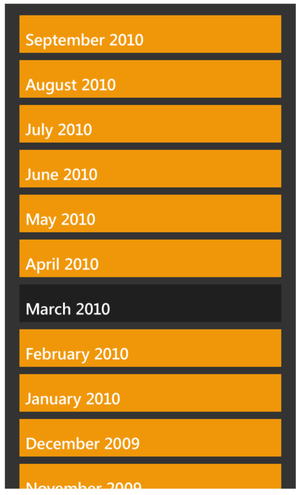
COMMON CONTROLS | **QUICK JUMP GRID** | OVERVIEW



QUICK JUMP GRID
Showing alphabetical list



QUICK JUMP GRID
Showing group header list

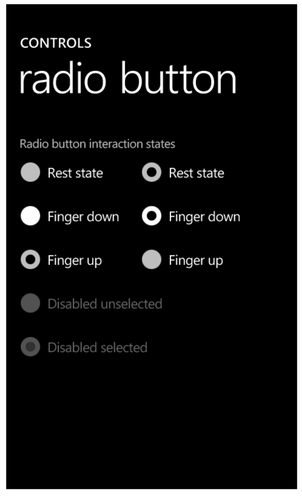


QUICK JUMP GRID
Showing dates list

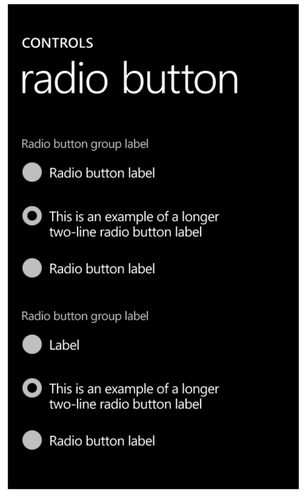
radio button

A radio button is used to represent a set of related but mutually exclusive choices. Only one option is selected at a time. The user can tap on the associated radio button label text to select the radio button.

COMMON CONTROLS | **RADIO BUTTON** | OVERVIEW



RADIO BUTTON
Touch interaction states



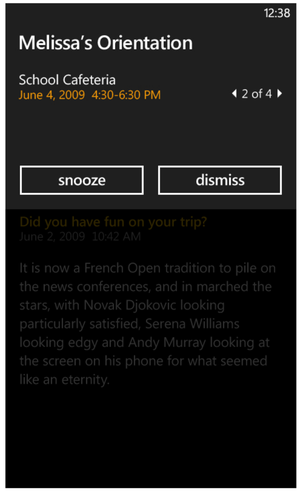
RADIO BUTTON LAYOUT
Shown with group labels to create hierarchy

reminders

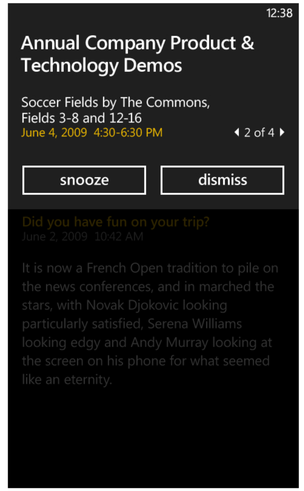
Reminders are used to display time-sensitive appointments, tasks, or alarms. Reminders are modal in nature and users must take action to dismiss them. Applications should provide enough information for a reminder to be descriptive and useful.

Application Design Considerations
- Reminders display over the application, covering the top portion of the application UI
- Users may snooze, dismiss, or open a reminder item
- Applications should create reminders with a call back mechanism to view the reminder details
- Snoozing a reminder uses a snooze time relative to the event. Snooze times are controlled by the system.
- Content which does not fit in the reminder UI is truncated and is not scrollable

COMMON CONTROLS | **REMINDERS** | OVERVIEW



REMINDER
Shown with single line text strings



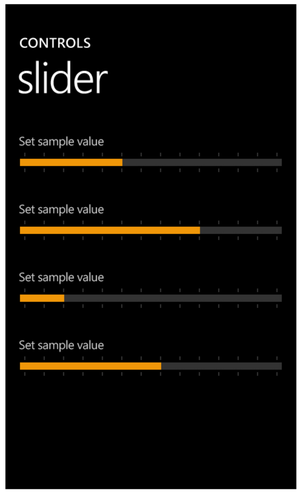
REMINDER
Shown with double line wrapped text

slider

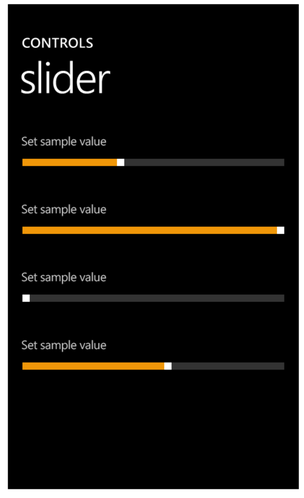
Sliders are used for setting a value in a continuous range of data (e.g. volume or brightness). Sliders are not used for discrete choices, such as days of the week, letters in the alphabet, or profiles.

Characteristics of the slider control
- Applications can use either a horizontal or vertical slider, but horizontal use is recommended
- Only the extremes of a range are described in the labels of a slider, for example, low/high, dark/light
- Icons can be used in sliders
- On a single page with stacked sliders, keep the elements the same, text labels and icon labels cannot be combined
- Tick marks are optional
- The setting can be represented by the thumb mark, or by a colored bar

COMMON CONTROLS | **SLIDER** | OVERVIEW



SLIDER CONTROL
Shown with tick marks and snap points



SLIDER CONTROL
Shown with active thumb/grabber

start & app list

Start is the beginning of the user's experience when they unlock their phone and displays application tiles, easily recognizable visual shortcuts for an application or its content. Users can set tiles in an arbitrary location on the phone start experience. Other than pre-installed application tiles, only the user can pin tiles to start. Double-width tiles are only available to Microsoft, phone manufacturers and mobile operators.

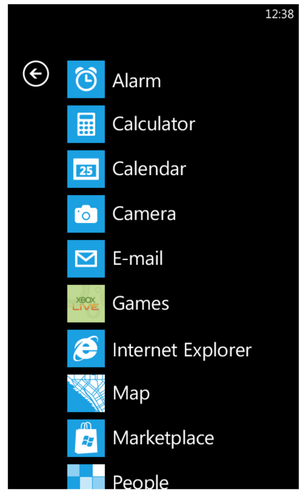
Tiles can communicate information to the user by displaying an optional counter that uses the system font, updating developer-provided tile background images, or displaying an optional title that uses the system font and is a fixed size and color. Counter, background image and title updates are controlled using the tile notification service. The accent color for the counter is always the accent color the user has selected. Counter display is optional.

The app list is a list of all the applications installed on the phone

UX FRAMEWORK | **START & APP LIST** | OVERVIEW



START SCREEN QUICK LAUNCH
Example showing default tiles



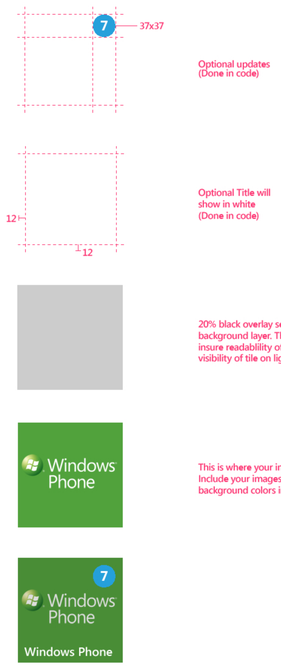
APPS LIST
Example showing default applications

start tiles

Recommendations

- Try and be flat, not glossy. Do not apply lens flares.
- Limit the use of words. When your icon does not represent or differentiate your application use the title to name it.
- Keep it simple. Try and think of one concept that represents your application.
- Set the expectation of your application by its visual representation. The experience will feel more comfortable to the user when your tile feels as one with your application.
- Do not use white as your background color. This will make your tile disappear within the light theme, and make it impossibile to see your title, if you choose to use it.
- Do not try an use rounded corners within your backgrounds. They will always be applied over a square.
- The tile title can be displayed without using tile notifications.
- If you use multiple tile images, they should be visually consistent with each other and have a recognizable theme or style.
- Developers cannot change the color, font, font color or size of the counter display.
- Be conservative in tile notification use – excessive use can negatively impact battery charge.

UX FRAMEWORK | **START TILES** | OVERVIEW

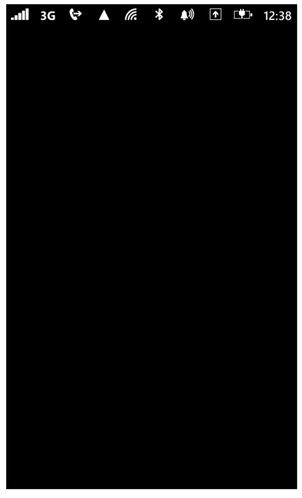


TILE TEMPLATE

status bar

The status bar displays system-level status information in a simple and clean manner, reserved in the application workspace (status bar real-estate on screen is reserved for the system). The user may interact with the status bar, as it updates to provide different notifications and keep them aware of things that are important (roaming, signal strength, battery). It's visible throughout the device, and is displayed in all application UI (with exception of full-screen).

UX FRAMEWORK | **STATUS BAR** | OVERVIEW



STATUS BAR
Shown with all slots

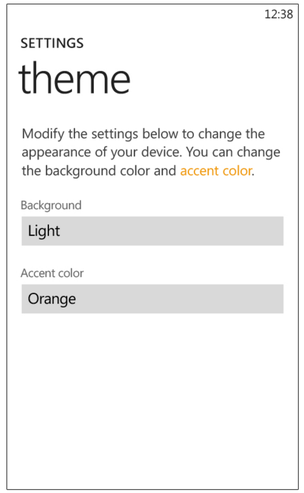


STATUS BAR
Showing slot designations

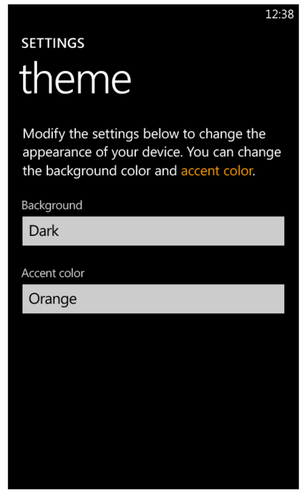
theme

The Windows Phone architecture allows for an application to respond to a user theme setting. In the theme control panel in settings, the user can set their background preference to either light or dark and their accent color to any of ten colors. Only applications which have implemented common controls will respond to changes in the system theme. Custom controls will not respond to theme changes unless they have been programed to do so.

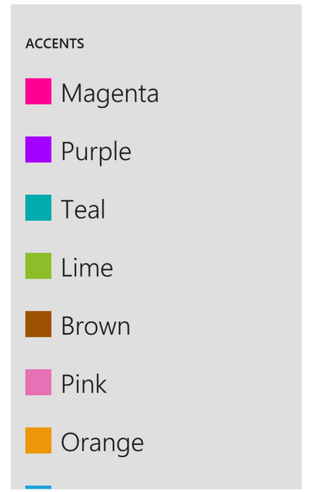
UX FRAMEWORK | **THEME** | OVERVIEW



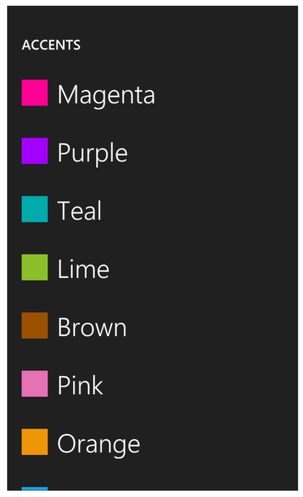
THEME SETINGS
Shown in light theme



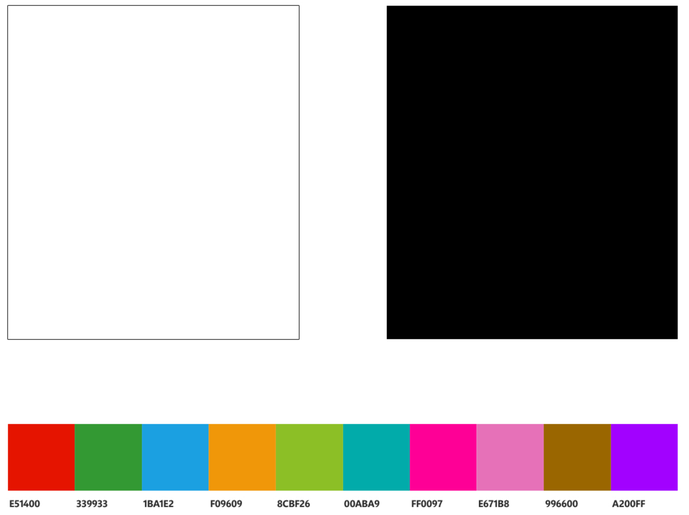
THEME SETTINGS
Shown in dark theme



ACCENT COLOR PICKER
Shown in light theme



ACCENT COLOR PICKER
Shown in dark theme

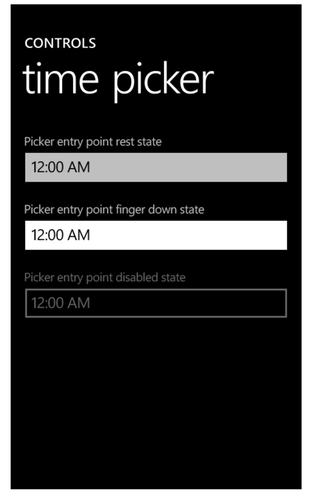


time picker

A time picker is used to select a specific time. To access the time picker, the user taps on a button which is prefilled with the appropriate time for the application. Upon button tap, time picker opens and contains "Done" and "Cancel" buttons. Once in time picker, the user uses the flick and pan gesture up and down to scroll the hours and minutes. Each digit (10s and 1s) of minutes values are selected individually . Either the 12 or 24 Hour clock format is used, respecting users' system preferences.

This control is not shipped as a part the Windows Phone Developer Tools. The templates and instructions are provided as-is to assist designers and developers to match, visually and in interaction, a similar control that appears in Windows Phone OS 7.0.

COMMON CONTROLS | **TIME PICKER** | OVERVIEW



TIME PICKER
Entry points and touch interaction states



TIME PICKER
12 hour format landing screen



TIME PICKER
Showing expanded hours column

toast notifications

Toast notifications are used to promote awareness of a new event to the user and request action from the user in an unobtrusive and friendly manner. Toast notifications should be used sparingly, and are tied to personal events or an unexpected situation. Applications should be conservative in displaying toast notifications in order to avoid annoying the user.

- Users may interact with the toast notifications by tapping on it. The toast will then launch the notifying application.
- Toast notifications are global, showing up in all applications
- Toast notifications come in from the top and overlay application content
- The height of the toast notifications is fixed
- Applications should design UI to accommodate the height of the toast notifications
- Toast notifications time-out and disappear completely after a period of time

COMMON CONTROLS | **TOAST NOTIFICATIONS** | OVERVIEW



TOAST NOTIFICATION
Showing SMS application notification



TOAST NOTIFICATION
Showing system message from Wi-Fi

toggle switch

A toggle switch is used to indicate whether a particular condition is on or off. Toggle switch can only be used for on/off. Toggle switch is intended to add an element of delight and should be used sparingly. The toggle switch presents to the user a binary set of mutually exclusive choices. The control shows only one of the two possible choices at any given time, and provides visual awareness to the user of this. Tapping on the control toggles between the two possible choices.

This control is not shipped as a part the Windows Phone Developer Tools. The templates and instructions are provided as-is to assist designers and developers to match, visually and in interaction, a similar control that appears in Windows Phone OS 7.0.

COMMON CONTROLS | **TOGGLE SWITCH** | OVERVIEW



TOGGLE SWITCH
Touch interaction states



TOGGLE SWITCH
Shown in wi-fi settings screen

Windows Phone 7

©2010 Microsoft Corporation. All rights reserved.
Microsoft and Windows Phone are trademarks of the Microsoft group of companies.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Schedules and features contained in this document are subject to change.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.