

Designing for Direct Manipulation

User interaction with pen and touch input differs from that of traditional pointing devices, like the mouse, in the following ways:

- Moving the cursor involves moving the hand an equivalent distance. Interfaces that require frequent cross-screen movements can slow user interaction and cause fatigue.
- The hand that holds the tablet pen or touches the screen can obscure screen areas that are below it. The user encounters difficulties if the areas that are obscured convey important information.
- It can be harder to point to and tap small objects.
- The user may inadvertently move the pen when double tapping or when pressing the tablet pen button.

Monitor and Respond to Pen Interaction

The digitizer on a Tablet PC provides detailed information about pen activity. The resolution of the digitizer in a Tablet PC screen is greater than that of the display. The high resolution of Tablet PC digitizers enables the system to smoothly and naturally capture handwriting, sketching, and other pen input. Pen positioning information is reported as tablet units. A tablet unit measures 0.01 millimeters.

Certain pen movements (or combinations thereof) are recognized as system gestures. A system gesture is a pen movement that corresponds to one of the basic user interaction events in a graphical user interface, such as the following:

- Clicking
- Double-clicking
- Dragging

Windows Vista reports system gestures through mouse messages, such as [WM_MOUSEMOVE](#) and through pen programming interfaces, such as [RealTimeStylus](#). For a complete list of system gestures, see the reference topic about the [SystemGesture](#) enumeration.

When translating pen movements to system gestures, the system accounts for user preferences, such as allowed travel distance within a double-tap, and press-and-hold delays for right-clicking. The user can set these preferences in Tablet PC Settings in Control Panel. As the system reports system gestures through mouse messages, it maps the high-resolution position data from the digitizer to the lower-resolution pixel coordinates that are used with mouse events. Tablet PC applications can also monitor the high-resolution stylus data through interfaces such as [RealTimeStylus](#).

Applications can monitor system gestures and other pen activity by using one of three methods:

- By monitoring mouse messages, such as [WM_MOUSEMOVE](#) and [WM_LBUTTONDOWN](#), or the equivalent mouse events in the Microsoft .NET Framework.
- By using the events that are provided with ink collecting objects, such as [InkCollector](#). In addition to reporting system gestures, these objects expose rich interfaces that enable ink collection, editing, and storage.
- By implementing a real-time stylus plug-in, such as the [IStylusSyncPlugin](#) and [IStylusAsyncPlugin](#) interfaces. These plug-in interfaces offer finer control than the ink collecting objects offer, which are designed for simpler, drop-in implementations of Tablet PC functionality.

Applications should not mix these methodologies.

Reduce the Need for Cursor Travel

With pen and touch input, hand movements are mapped directly to the cursor, so a cross-screen movement might involve 20+ centimeters (cm) of hand movement, compared to just 2-3 cm when using a mouse. The user can become fatigued by a user interface that requires the hand to repeatedly span the full screen distance to access commands and select information.

When designing for pen or touch input, look for ways to reduce the travel distance that's required to activate features and tools, such as the following:

- Place frequently used functions on toolbars on the application frame. When you provide quick, one-tap access to functions that are commonly used in mobile environments, users can avoid the potential difficulty of selecting commands from drop-down menus.
- Provide thoughtfully designed context menus that display a short list of command choices that are appropriate for the current selection and context. Also, consider creating a radial menu, a context menu that lays out menu items in a circle surrounding the cursor location. Radial menus can be navigated quickly and easily by using the tablet pen or a fingertip. For more information about context and radial menus, see [Using Context Menus](#).
- Support gestures, which provide quick access to application functions. You can support the application commands that are accessible by using pen flicks, and provide additional command shortcuts by implementing application gestures. For more information, see [Integrating Application Gestures](#).

Make Interface Elements Large Enough

Small objects are harder to target with the tablet pen and touch. If the object that the user is trying to tap appears to be small or difficult to differentiate from nearby objects, the user is forced to slow down and focus closely to tap the object. Don't force the user to engage in precise, minute pen movements to target screen elements; it's frustrating and, with repetition, exhausting. Size your interface elements large enough so that they are easy to tap.

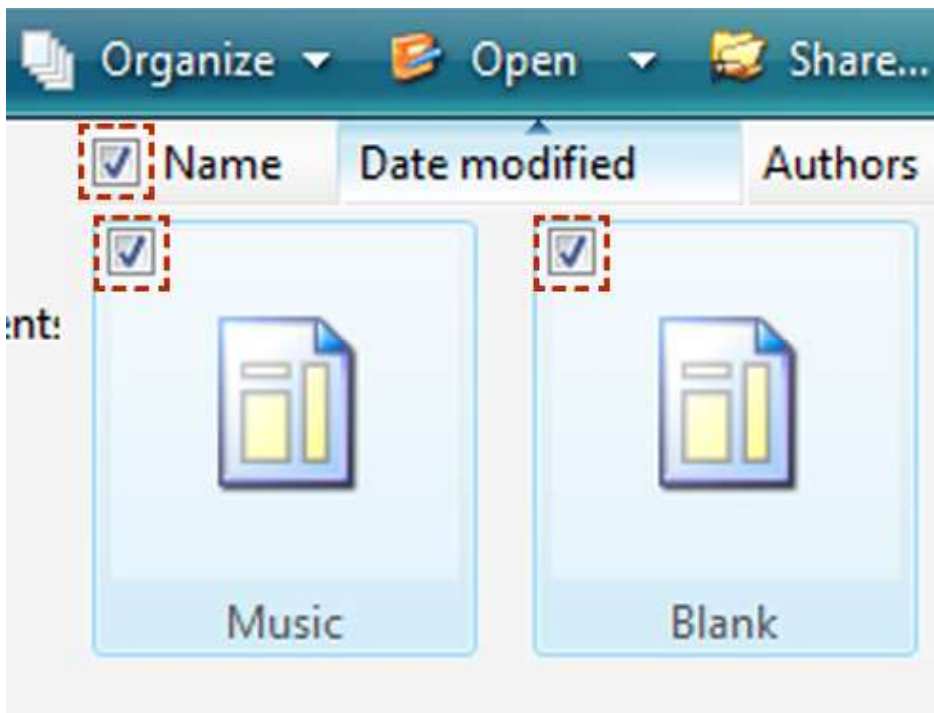
To accurately predict the size of screen elements on various types of mobile PC displays, you must implement a dpi-aware application. For more information, see [Pixel Density and Usability](#).

Follow these guidelines when sizing interactive screen elements.

Pointing method	Minimum size
Pen	2.5 mm
Touch (using touch pointer)	2.5 mm
Touch (no touch pointer)	9 mm

Increase the Hotspot Size

In addition to creating sufficiently large interface elements, you can also increase the size of the area around the element that users can tap. By making the hotspot larger than the visible element, you enable the user to select it with less precision. By combining a larger hotspot with hover feedback, you can make the element much more accessible. For example, consider the visual area and hotspot area of the following selection handle.



Provide Optimal Spacing Between Controls

As mentioned, targeting is more difficult when using a pen or finger as the pointing device. Users sometimes tap outside the intended target. When interactive controls are placed very close together but are not actually touching, users may click on the inactive space between the controls. Because clicking the inactive space has no result or visual feedback, the user is often uncertain about what went wrong.

To address this issue, the target regions of interactive controls should either be touching, or have at least 5 pixels of space between them. To avoid having interactive controls visually touching, you can increase the hotspot size. The targeting problem is addressed when the hotspots of the adjacent controls follow the spacing guidelines, eliminating imperceptible "dead regions" between interactive controls.

Correct:

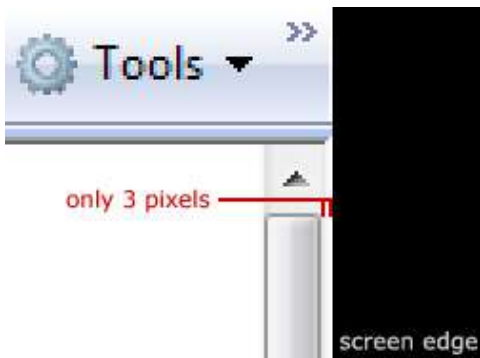
In these two examples, the target regions are correctly positioned. In the first screen shot, from Windows Photo Gallery places, the target regions are five pixels apart. In the second screen shot, from Microsoft Office Word 2007, the target regions touch.





Incorrect:

In the following screen shot from Microsoft Internet Explorer, the inactive space to the right of the scrollbar is especially difficult to target because it's near the edge of the screen.



Implement Hover Feedback

The hand can obscure the screen, so it can be difficult to tell whether the cursor is over a target. Provide some type of visual feedback that confirms when the cursor hovers over a control or hotspot. This type of feedback also helps users navigate the interface more confidently in low-contrast situations when glare makes it harder to see the screen.

Hover feedback can include visual changes in the cursor or the element hovered over. Hover feedback is helpful for mouse users as well. You'll see hover feedback implemented on Tablet PCs for several of the Windows common controls, including check boxes, option buttons, and spin boxes.

Allow for Involuntary Hand Movements

In Windows Vista, mouse messages are generated from tablet pen events, such as a tap or double-tap on the screen, or the user holding the pen stationary. The conversion of pen movements to mouse messages is determined by a set of

rules that are designed to account for the involuntary hand movements that are typical with pen input. The tablet digitizer can register movements as small as 1/100 of a millimeter. The pen is likely to slip some distance across the screen even if the user intends to tap a very specific point on the screen.

In Tablet PC Settings in Control Panel, the pen user can specify settings for double-tap, hover, and the press-and-hold action that substitutes for using a right mouse button. These settings provide flexibility to accommodate involuntary hand movements. The system uses these settings when interpreting pen movements and converting them to mouse messages and system events.

By using appropriate sizing in your forms and interface elements, and by employing helpful behavior such as "snap-to," you can better accommodate the pen user whose hands may tremble or who may hold her Tablet PC in such a way as to make precise targeting difficult. A good example of snap-to behavior can be found in the forms editor in Microsoft Visual Studio 2005.

Account for Handedness

Handedness refers to whether the user is right-handed or left-handed. Be sure to test your application with both left-handed and right-handed users. On Tablet PCs, handedness is an issue because the hand blocks some region of the screen. Which part of the screen is blocked depends on which hand is holding the pen or touching the screen.

Windows Vista includes a handedness setting that affects the positioning of menus. By default, handedness is set to right-handed in Tablet PC Settings in Control Panel. You can test for this setting by calling the [GetSystemMetrics](#) function with the SM_MENUDROPALIGNMENT flag. A return value of nonzero indicates right-aligned menus (for a right-handed user). A return value of zero indicates left-aligned menus (for a left-handed user).

Be aware of the probable hand location when you arrange information and controls in your application. For left-handed users, information on the right side of the screen is easier to view. The opposite is true for right-handed users.

[Send comments about this topic to Microsoft](#)

Build date: 2/8/2011

Community Additions

© 2016 Microsoft