TAGGED OBJECT INTEGRATION FOR SURFACE 2.0

February 2012

CONTENTS

INTRODUCTION

Are you tasked by your customer with implementing a solution that requires table-to-object interaction on a Microsoft® Surface® device? Do you need to manage large numbers of unique identities using only Surface tags? What are the security implications of trying to authenticate by using tags? What is the difference between Microsoft Tag™, Surface tags, QR Codes, and barcodes; and how are they all used on Microsoft Surface? Read on to learn more.

PixelSense™ enables Microsoft Surface to see objects placed on the screen in the near-infrared (NIR) spectrum. This technology enables Surface to see fingers and other objects placed on the screen. It also provides for the recognition of Surface tags, marked with a specific pattern of dots.

Tags enable objects to drive the Microsoft Surface experience. Application integration of tagged objects is achieved through the Surface 2.0 SDK. The criterion for tag visibility falls into two basic categories; tag geometry and IR reflectance. The possible scenarios for tagged objects include printed paper, stickers attached to objects, and laser etched acrylic using IR reflective paint to illuminate the dot pattern of the tag. A tag consists of a specific arrangement of infrared reflective and absorbing areas.

PixelSense™ enables Surface to process raw images, but using tagged objects gives your application several advantages over processing raw images:

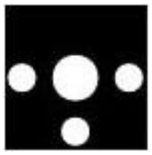
- Tagged objects are more computation-efficient. The tags are small and well-defined, so specialized code in the vision system can locate them quickly, accurately, and efficiently.
- The Microsoft Surface platform provides APIs to enable recognition of tags, so it is easy to write an application that uses tags. In contrast, an application that uses raw images must include its own image recognition algorithms.
- Each tag represents a distinct binary code value, so an application can distinguish one object from another.

DEFINITION, VALUE, AND USAGE

What are Surface tags? Why do I want to use them? How can I expect Surface tags to interact with my Surface applications? When does using a tag make sense?

Tags

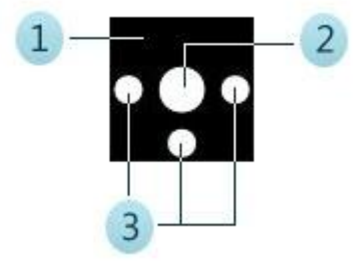
Tags store 8 bits of data (1 byte) and orientation markers. There are 256 unique tag values, ranging from 0-255.



For more information see What's New in Surface 2.0 (http://go.microsoft.com/fwlink/?LinkId=242346) and Tagged Objects and Tag Visualizations (http://go.microsoft.com/fwlink/?LinkId=242348).

Tag Geometry

Each tag that the Microsoft Surface vision system identifies has an orientation and an 8-bit tag value. The following figure illustrates the basic tag that all other tags are based on.



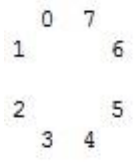
1. Infrared-absorbing or non-reflective (pass-through) background.
2. One infrared-reflecting circle (0.125-inch radius) in the center of the tag. This circle locates the tag on the Microsoft Surface screen.
3. Three infrared-reflecting circles (0.08-inch radius) located 0.28 inches from the center of the tag in each direction (left, right, and down). These "guide" circles determine the tag orientation.

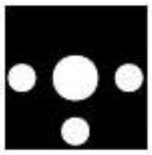
In addition, each tag contains from zero to eight data bits that define the tag value. These data bits are white circles (0.075-inch radius) that are centered in one of the following locations:

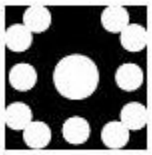
- 0.3058 inches horizontally and 0.2038 inches vertically (bits 1, 2, 5, and 6) from the center of the tag.
- 0.2038 inches horizontally and 0.3058 inches vertically (bits 0, 3, 4, and 7) from the center of the tag.

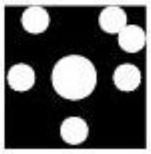
Tag Values

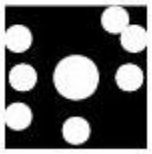
There are 256 possible unique tags that are encoded as a byte. The highest order bit (bit 7) is at the 1 o'clock position when you look at the printed side of the tag. Less significant bits are then read counter-clockwise from the 12 o'clock position. An infrared-reflective circle in a bit location represents 1. The absence of a circle represents 0 (zero). Values listed in the following illustrations are read right to left.




Tag 0x00 (00000000 bin)


Tag 0xFF (11111111 bin)


Tag 0xC1 (11000001 bin)


Tag 0xC6 (11000110 bin)

When to Use Tagged Objects

Know where people or objects are located on Surface: An application can use tags to recognize an object or distinguish between collections of objects. Applications can keep track of individual customers as they interact with Surface. For example, Surface can keep track of special playing pieces used in a game being played by multiple people. Similarly, a chess application can use multiple tags to track pieces on screen.

Start an action: An application might use a tag to initiate an action or activity. For example, placing a tagged object on the Microsoft Surface screen might open a menu or start a video.

Start or display an application: Applications can register a particular tag value. When a person places the relevant tag on the Microsoft Surface screen, it displays a menu that enables that person to start or display any application that has registered the tag. If a person selects an application by using a tagged object, the application designer can use that tag contact as the starting point for the customer's experience with that application. This feature is called object routing.

Point and orient within an application: An application might require precise movement with absolute orientation. For example, an application might enable a person to move in sequence through a series of video clips, turn the pages in a virtual book, or adjust the volume of a video or audio clip. To adjust the volume, the application might enable the customer to turn a tagged object in a circular motion, like the dial on a radio or television. (Microsoft Surface applications enable people to move and rotate an object with their fingers. But tags provide a more precise absolute location than fingers, so applications can use tagged objects as pointing devices.)

Note

We have found that the best experience for tag tracking can be achieved by tracking the tag value, instead of the tag ID.

Deliver personalized experiences: Applications can manage identity scenarios or allow access to special UI in the application. Privileged access to special interaction options is achieved through the use of a specific card, such as an assisted sales scenario. Customers are identified through unique cards.

Recall data, get results: Tag values are associated with predefined search strings that execute when the tag is presented to the screen. For example, a tag might open a Bing for Microsoft Surface predefined search result.

Security Considerations for Tagged Objects

Do not use tagged objects to access private data. Tags are not sufficiently secure to use as a method of authentication. You can use tagged objects to identify a customer, but you should require the person to enter a password, personal identification number (PIN), or other secure data for authentication purposes.

When you use tags, consider the following important security issues:

A tag does not have enough bits to be cryptographically secure. Do not use a tag as a password equivalent.

Tag bits are not encrypted when they are in memory. For example, they are not equivalent to the SecureString class in the .NET Framework (http://go.microsoft.com/fwlink/?LinkId=242350). Writing code that works with tags is like writing code that works with strings. For your application is String sufficiently secure, or would you need to use **SecureString** (http://go.microsoft.com/fwlink/?LinkId=242352)? If you would use **SecureString**, do *not* use a tag for the data.

Depending on how tags are printed, photocopiers, digital cameras, and other devices can easily capture the tags so they are easy to duplicate.

Overall, we recommend that you can use tags for identification but you should *not* use them for authentication. The contents of a tag are not more secure than a URL or an e-mail address. Authenticating with a tag would be similar to authenticating using an email address only.

Registering a Microsoft Surface Application

Every Microsoft Surface application must have an associated XML file to register the application with Surface Shell. This file enables Surface Shell to recognize the application and display it in Launcher. The XML file also includes elements for registering tagged objects that your application uses and to enable your application to use object routing. To register the application with Surface Shell, you must create a shortcut file (.lnk) that points to the application's XML file, and then put that shortcut file in the appropriate ProgramData directory. Any application that uses tagged objects should register the tags that it uses. For more information, see Registering a Microsoft Surface Application (http://go.microsoft.com/fwlink/?LinkId=242353).

Tagged Objects and Applications

Note

Always test your tags on a device made for Surface, using Data Visualizer to verify consistent value, orientation, and object ID; and Raw Image Visualizer to verify visibility (available through the Surface SDK 2.0). ***There is no substitute for testing tag recognition on a device made for Surface prior to mass production.***

When a tagged object is placed on a Microsoft Surface screen, the Surface software reads the object's tag value and checks if the tag is registered with Surface Shell. If your application is designed to be driven primarily by tagged objects, consider integrating it with the object routing feature (http://go.microsoft.com/fwlink/?LinkId=242349). Object routing enables customers to start your application without going through the Launcher. Ideally, after a person opens your application by using a tagged object, the application will continue to use the tag value in some way.

Note

Even if your application does not use object routing, you should register tags that your application uses to prevent interference from applications that do use object routing.

The Microsoft Surface software also checks whether any tag is registered for object routing. If a tag is registered for object routing, the Microsoft Surface software displays a menu that includes all of the applications that are registered for object routing with that tag.

When you design an application that works with object routing, we strongly recommend that your application use the tag value for some clear purpose after a customer opens the application from the object routing menu. A strong design will provide immediate, obvious information to the customer based on that tag. People should instantly see what options are available and how to proceed based on the tag that they placed on the screen. For example, the tag value could correspond to a particular promotion or set of personal preferences.

You can associate tagged objects with your application by using the application's XML registration file. For more information, see Registering a Microsoft Surface Application (http://go.microsoft.com/fwlink/?LinkId=242353).

Reacting to Tagged Objects and Clustered Tags

If you are creating an application based on the Presentation layer (using WPF, http://go.microsoft.com/fwlink/?LinkId=242354), your application can respond to tagged objects most easily by using the TagVisualizer control (http://go.microsoft.com/fwlink/?LinkId=242356). This control can recognize, track, and show user interface (UI) elements for tags that are placed on the control.

Note

We have found that the best experience for tag tracking can be achieved by tracking the tag value, instead of the tag ID.

Multiple or ‘clustered’ tags could be used to achieve unique numbers beyond 256 values. Clustered tags are not natively supported by the TagVisualizer control. A custom version of TagVisualizer would need to be implemented for your scenario, by listening to basic touch events.

If you want to use custom processing (or if you are writing an application based on the Core layer, typically using XNA, http://go.microsoft.com/fwlink/?LinkId=242357), your application can respond to tagged objects by watching for TouchPoint objects whose IsTagRecognized property is set to true (http://go.microsoft.com/fwlink/?LinkId=242358). When that property is true, you can use the Tag property on the contact to get information about the type and value of tag that has been recognized.

PRODUCTION, HOW DO WE DO IT?

Now that you understand the what, when, and why of Surface tags, let's talk about the various ways to implement physical tags and tagged objects to allow interaction with your Surface applications.

Tag Orientation and Placement

A Microsoft Surface application relies solely on the orientation and position of a tag (instead of the physical object that the tag is attached to), so the Microsoft Surface platform needs to know where the tag is located on the physical object the tag is attached to. The location enables the platform to infer the location of the object based on the location of the tag. Depending on the physical requirements of your tagged object, you might need to place the tag somewhere other than its center. In such a case, you should correct for the location of the tag so your application can show the UI for the tag in the appropriate position. This is especially true for UI intended to rotate as the tag rotates.

When a tag is placed on an object with a large white background, the tag should never be positioned in the center of the object. The ideal performance should be expected by positioning the tag approximately 1/4th of an inch (or more) from the edge of the white area, and away from the center of the white area.

Reflectance

Note

Always test your tags on a device made for Surface, using Data Visualizer and Raw Image Visualizer (available through the Surface SDK 2.0). ***There is no substitute for testing tag recognition on a device made for Surface prior to mass production.***

Make sure that printed tags meet the following requirements:

- The white dots should reflect light. Overly reflective materials can cause a "bloom" effect that will interfere with tag recognition.
- The dark region should absorb light. Many printers that use the CMYK (Cyan, Magenta, Yellow, and Black) format will print a color that appears black, but uses only CMY, not K or "true" black. CMY will often be transparent to the Surface vision system. The dark region of the tag needs to be printed with true black for the best results.

- A single tag on a physical object with a black or non-IR reflective background provides the best tag recognition performance for small objects, such as game pieces. It is important that no “slivers” of lighter color appear around the edges of the tag area.

- For large objects, such as promotional fliers, the best performance can be expected when the only “true” black used on the tagged side of the object is the tag itself. Any other printing on the tagged side of the object should be done with CMY only (no K), using narrow fonts with lighter tones. In certain cases, large areas of darker toned CMY will interfere with tag recognition.

High IR-Reflective/White

- White office paper
- White Mylar
- White PVC
- White Styrene
- White print labels (for example, Brother TZ-261)
- IR-reflective dielectric coating on PET film

In general, most white office paper and card stocks work well. However, some are too bright and can cause a blur in the Surface vision system. Test each solution before mass production.

Low IR-Reflectance/Black

- Laser printer black ink
- Pantone Black C
- Epolight 8771 IR absorbing ink

Choosing Materials for Objects

Place tags on heavier flat materials that help the tag maintain flush contact with the tabletop. This means that card stock will result in better performance than lighter weight office paper. Also, tags work better on physical objects that do not easily tip over while they move.

The following materials and processes have been validated and shown to produce tags that function correctly with Surface. If you use these materials, your tags should work properly.

However, *always test new tag types in their final format*; both in Data Visualizer, and with a practical use test, before you distribute them.

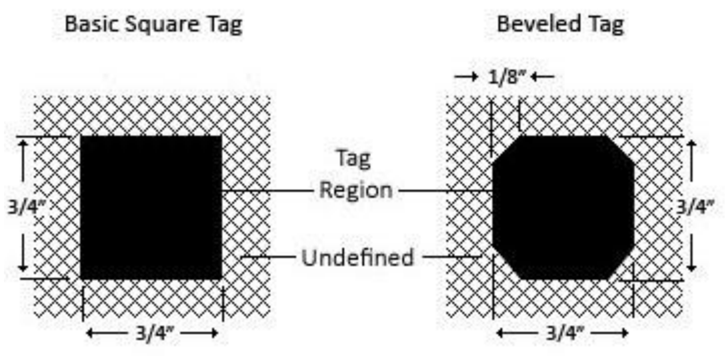
Top Coats and Protective Layers

In practice, top coats such as UV print varnish and laminates can successfully improve a tag's durability and minimally affect the optical properties of the tag. However, many types of protective layers can significantly affect optical performance. You should test any protective layer that you add to a tag to make sure it performs correctly.

Printing Tags

You can print your own tags, according to the specifications in this topic, up to the maximum of 256 unique patterns. You can also download reference files that contain the proper dot patterns (http://go.microsoft.com/fwlink/?LinkId=242359). The tags print out at the correct size when printed on a 300 DPI printer.

If you want to create and print your own tags, you have some freedom when you design and work with the physical tags. The following illustration shows the dimensions of a physical tag.



The undefined region indicates where you can extend the geometry of the tag, if you need to.

The tag region is the opaque black-and-white patterned area of the actual tag, depicted as the solid black area in the previous diagram. You can remove or cut a 0.125 × 0.125 inch triangle from each corner of the 0.75 × 0.75 inch square tag without affecting performance, if you need to affix the tag to a smaller circular object. However, the entire tag region must touch the Microsoft Surface screen.

Printers

Laser printers work very well because the black provides a good contrast for printing tags.

Ink jet printers that use a mix of cyan, magenta, and yellow (CMY) to create black generally do not work for printing tags. The three-color combination is typically invisible in the infrared spectrum. You should test all printing methods for their appearance in the NIR wavelength.

Tags that are printed by using ink jet or laser printers can wear off rapidly when they are used or handled often, due to normal wear and tear. For this reason, we recommend tags printed on vinyl stickers or embedded in objects (such as etched acrylic) for maximum durability. Your use case will dictate the most appropriate production method.

Card printers, such as the Datacard CP40 Plus, can be used to print tagged cards for events, demonstrations, or to manage identity scenarios, etc.

Common Mistakes

There are a few common mistakes when trying to create a tagged object or print tags.

Tag size too large or too small: The geometry of the tag is very specific. Printing tags a little too small, or too large, will cause the vision system to detect them as blobs, not tags. It is wise to always test a prototype of your tagged object on a device made for Surface prior to mass production. Take care to prevent unintended scaling of the white dots in your workflow from design to final output.

Tags placed on white backgrounds are less visible: In some situations, where the tag is placed on a very highly reflective surrounding background, you may need to increase the size of the dark region to 1 inch or more in order to improve tag recognition.

Some printing processes do not work well for tags: Four-color printing processes do not typically work well. What appears black in the visible spectrum (human eye) may not be black at NIR. Some whites do not reflect enough light at NIR to appear white to Microsoft Surface. Monochrome laser printer output works very well, although without a laminate the tags do degrade fairly quickly due to wear.

Tags that are flat against the Surface work the best: Some tagged items, such as brochures or cards that are warped, do not press the tag directly against the screen when laid flat. This can sometimes cause problems when Surface attempts to read the tag.

Sticker Tags

You can order sheets of stickers from a third party printer, such as GM Nameplate (http://go.microsoft.com/fwlink/?LinkId=242361). You can print or order your own stickers from any source of your choosing. Tags wear out. If you want your tags to last a long time, high quality stickers will work the best. Ink applied directly to paper with no protective coating will wear out quickly.

Transparent and Hidden Tags

Tags can be printed on clear materials that enable visible light to penetrate them, appearing almost transparent to the human eye. These tags are visible to the Surface vision system through IR contrast. Similarly, tags can be hidden in an image or pattern by printing the tag pattern with highly IR reflective ink that is not readily visible to the human eye. To successfully hide a tag in a pattern or image, the background needs to have a low IR reflectance in contrast with the tag. This can sometimes be achieved by printing the design with CMY only, and using K only on the area containing the tag.

Note

Placing clear tags on clear objects with a bright underlying UI can sometimes result in an undesirable experience. This happens when the underlying UI reflects off of the clear object and then back down on the table.

Outsourced Printing

Printing of Surface tags, including transparent tags, can be handled through a third party printer, such as GM Nameplate (http://go.microsoft.com/fwlink/?LinkId=242361). You can print or order your own stickers from any source of your choosing.

Physical Object Tags

The longest lasting tags appear as objects that integrate tag values directly on them. You can create your own, or order them built to specification from a model shop. Laser etched acrylic, using highly IR reflective paint for the dots, is a solution that works very well. Tag values can be incorporated directly into many different types of physical objects.

Barcode Support

A barcode is an optical machine-readable representation of data typically used for scanning products or tracking inventory. The most common barcode is the UPC code used for scanning products at point of sale terminals. The best barcode integration for Surface uses a dedicated external barcode scanning device that can be built into the Surface enclosure, connecting to the unit via USB. Surface does not support displaying barcodes for scanning by external devices.

TAGS DISPLAYED BY SURFACE

Surface tags are used to trigger events on a device made for Surface when they are detected through PixelSense as the tag is presented to the device made for Surface. Other tag technologies can be displayed by Surface, allowing camera and scanner equipped devices to scan those tags and interact with Surface in a variety of ways.

Microsoft® Tag™

Get the free mobile app for your phone
http://gettag.mobi

Microsoft Tags are 2D barcodes that connect people with information, entertainment, and interactive experiences in the digital world.

The Microsoft Tag is used to trigger events, such as navigating to a hyperlink, on a mobile device equipped with a camera and a software Tag scanner. An example would be the Microsoft Tags displayed on Bing for Microsoft Surface search results. For more information see Microsoft Tags (http://go.microsoft.com/fwlink/?LinkId=242360).

QR Code™

QR Code, abbreviated from Quick Response Code and a registered trademark of Denso Wave Incorporated, is similar to a Microsoft Tag. A QR Code can represent numeric or alphanumeric characters, binary, URLs, Kanji, and variable amounts of instruction directly in the QR Code. QR Codes enable software code execution without the need for redirection to an external resource. As with Microsoft Tag, QR Codes display on the Surface to be scanned by another device, triggering events on that device.

The license to the use of the QR Code stipulated by JIS (Japanese Industrial Standards) and the ISO are not necessary. The specification for QR Code has been made available for use by any person or organization. For more information see QR Codes (http://go.microsoft.com/fwlink/?LinkId=242364).