

## **Bits & PCs - Designing Icons for Software Interfaces**

Richard Hodgkinson (IBM United Kingdom Ltd) & Denise Bell (Tivoli Systems).

*When creating icons for computer software there are many factors that the designer must consider from the standpoint of both the user and the software. In addition there are many valuable resources currently available to assist the icon designer. This paper addresses many of these considerations and identifies some of these resources that icon designers can draw upon to assist them in creating successful icons.*

With the increasing availability of graphics packages and dedicated icon editors anyone can easily design icons. However, if you require new or unique icons and do not have visual design skills available, seek out a professional visual designer. Their unique skills will make a remarkable difference to your product, provide input from an external and broader viewpoint, and where many icons are required, release your skills to address other problems.

When looking for a visual designer, if you are unable to find one who has already designed icons, seek one who is familiar with designing small objects such as company logotypes or postage stamps. It is important that the designer understands both the potential and limitations of the technology.

### **WHERE TO BEGIN**

Start the design process by carefully selecting the appropriate visual metaphors to use. The icon designer must understand the technical environment where the product is used, the working environment of the user and the key functions that the user will associate with each icon.

The first environment to consider is the platform (check the icons of the operating system, e.g. Windows, and complimentary products which the user will encounter:

- Can existing metaphors or objects be re-used?
- What are the technical requirements?
- What is the visual style?

Consider the working environment of the user....this should influence the metaphor

set to be used. For instance, take the concept of information storage. If the user works in an office then a file cabinet or folder might be appropriate metaphors, whilst for a person, such as a Data Processing system administrator familiar with host or mainframe computing concepts and terminology, the use of a stack of disks could provide a more appropriate solution.

Be very conscious of cultural considerations if you are designing for an international market. For instance, using the symbolism of hands and gestures. An upright, open-palmed hand facing the viewer can mean "STOP!". In Greece it can mean "Go to Hell!", whereas in sign language it can mean "Listen!" or "You look at this". Similarly the "thumbs up" gesture can mean "OK!", but in some countries it can be have a derogatory meaning, especially when accompanied by the words "sit on this!". In Eastern cultures, presentation or associations with the soles of feet or palms of hands are insulting. Similarly, body parts (eyes, ears, etc) and animals can have negative or religious connotations and should be avoided.

Unless icons can easily be replaced by sets for local use, do not include text in icons, since translation could be necessary.

Having established these requirements you will need a description of the object or function that the icon must represent to the user. Talk to as many people as possible who are involved with the development and requirements of the product so as to identify the main function or functions that the user will associate with this icon. Ask what the user will most frequently use this icon to access or initiate. You will often get varying views, but this process will provide valuable insights, information and ideas. Other valuable sources are product specifications and glossaries.

Icons on the interface can represent both objects and functions. In an object-orientated interface, object (or noun) icons are fairly straightforward to present as visual metaphors since they inevitably exist in the real world. There are fairly well established conventions for common office objects such as files, documents, mail, printers, etc. Part 2: Object icons, of

ISO/IEC 11581 - Icon Symbols and Functions (1) has standardised a core set of 19 of these objects. Function (or verb) icons are less easy to represent visually. Often we represent these by association with the object upon which they interact, or by the object resulting from their action.

In addition to the six part ISO/IEC 11581, there are a number of other valuable International Standards for symbols on office equipment such as ISO 7000, ISO/IEC 9995 part 7, IEC 4196, etc. Also, in preparation within the ISO/IEC JTC1 (Information Technology)/SC 35 (User Interfaces), is a standard that will consolidate all the existing standards of (hardware) symbols for office equipment into a single standard (ISO/IEC 13251), when this is published it should also be consulted.

There are other useful sources of ideas. Books are available on icon design (The Icon Book by William Horton (2)), collections of symbols (such as Henry Dreyfuss's Symbols Source Book (3)), many interface design guides provide reference information.

Both IBM (4) and Microsoft (5) also provide guidance information on their World Wide Web sites. Available free of charge from IBM is a collection of Windows® icon clip art, providing both a consistent set of common metaphors to use when building icons for Windows '95®, 98® and NT® applications. The core of this collection is a set of icons that conform to ISO/IEC 11581 parts 1, 2 and 6 (providing they reflect the correct functions). The collection and guideline can be downloaded from [http://www-3.ibm.com/ibm/easy/eou\\_ext.nsf/Publish/648](http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/648) [Editor's note 2005/01/21: this URL is not available.]

Another useful source is road traffic control signage, but be careful to use only those which are used internationally (if your product is for an international market and substitute national icon sets are not provided). Do not misuse any safety symbols that are used in dangerous or life-threatening situations.

It may be more appropriate to use an object with a historical association to the system object, rather than represent the real object itself....a church bell is

commonly used to signify an alarm, or a steam train to represent a railway crossing....these are no longer the real objects in use, but are far more memorable and have real associations to the user than say a circuit that makes an alarm sound or a modern locomotive that is visually similar to a tramcar or bus.

Use abstract geometric symbols only as a last resort, there are few conventions as to their meaning (e.g. flowcharting, safety/road signs) and no widespread established guidelines as to their use.

## **DEVELOPING DESIGNS**

From your information gathering exercise you will frequently end up with descriptions that amount to a paragraph or more of text, and different developers can give varying views of the purpose of a single function!

Next "distil" this information down to the visual equivalent of a single phrase or short sentence to end up with no more than three component objects for your icon.

Many objects represented as icons on the screen interface provide a number of functions and to attempt to create an icon which visually "describes" all of them would be very complex and confuse the user. With care, sometimes more than three components can be accommodated, but constantly review the final size of the icon on the interface to ensure that the image is discernible.

These components will in turn influence the design of any smaller or "mini" icons you will need to provide for other resolution or implementation variants, and which will require a further "distillation" of the concept. Ensure that you always provide the differing cell size variants that the operating system or platform requires. If a specific cell size is not provided, the software will automatically create the missing sizes...frequently not with the result that the designer would appreciate!

Begin the design process on paper by writing out this phrase and drawing candidate objects to represent the nouns, together with any ideas that have been proposed by others. Include anything, however bizarre or flippant that comes to mind...if you run out of ideas, work on

another icon....do something else....break your locked thought stream and when you return, you'll approach the problem from a different perspective with fresh ideas.

Keep these sketches small as there are limits to what you can build and do not destroy them as they can often be a source of valuable alternative ideas that you can return to later. It is not uncommon to have to rebuild a "finished" icon set due to functionality being removed from the product or combined into single objects during development. This can necessitate redesigning already completed icons to reflect changes elsewhere.

### **BUILD THOSE IDEAS**

Transfer your ideas from your sketches using a dedicated icon editor that provides a constantly updated final size model so you can see the "true" icon as you build it. If you can, build several icons side-by-side, as this will enable the development of several ideas in parallel.

So as to maintain adequate sizes when several elements are present, you may decide to overlap them. This is called "cascading" and is typically done by arranging them diagonally within the cell from a top corner. To maintain the identity and emphasis of these elements great care must be exercised as to which parts are concealed or revealed, the direction that the objects face, and which are to have priority or emphasis (i.e. which one has to be on top of the stack?).

When cascading these elements avoid visual "clumping" which results where two or more parallel lines from differing objects combine and result in the visual equivalent on an ink blot...overlay these lines as a single pixel thickness, common line by moving one of the objects closer by a pixel width.

Similarly, avoid "run-ons" where a prominent line in one object coincides with another line in a second and can visually continue through them both resulting in a, possibly undesirable, visual linking of two objects. This can be avoided by offsetting one of the objects.

To maintain the individual identity of overlaid objects consider the use of toning down or "greying" the rearmost to give an impression of distance between them. Alternatively, the use of single, pale grey

pixels on the rearmost object, where they "touch", will further aid separation and identification.

Consider the overall silhouetted shape of the icon. It has long been recognised in typography that the visual variety of upper and lower case characters in words aids recognition over uppercase (capital) only words. Apply this concept to your icons so that unless there is a specific purpose in maintaining regular outlines (such as in a set where only a single element changes the shape or colour to show a change in condition or status) they vary visually. This will be of further value if the operating system uses the remaining (transparent) area in the cell to show selection status.

Consistency is a critical element of icon design. Having established a family of metaphorical objects together with visual hierarchies, a set of colours, etc., do not arbitrarily change these. Where large sets of icons are required, don't create them piecemeal, step back and consider the entire set:

- Where should there be commonality?
- Where can objects be reused?
- What are the rules for combining them?

Where should relationships exist? Often you will need to rework already completed icons as you progress through the set and encounter new variations of your object set. It's not unusual for additional icons to be requested when your designs are well advanced, usually resulting in some degree of rework of "completed" icons. Consistency is the glue that binds this visual language together, speeds the users understanding and recall, saves you time and effort and your customer money.

### **PRESENTING DESIGN PROPOSALS**

Produce only a single, resolution variant draft icon. A 32x32, 4 bit (16 colour) one is suitable for most draft and testing purposes.

If the image request consists primarily of 16x16 images, use this as the starting size for your designs.

Consider how you will review your design proposals. On most computers this cell size is suitable, but note that today's workstation screens run at 1600x1200,

1280x1024, 1024x768, 800x600 or 640x480 pixels on a variety of physical screen sizes. Note that some display devices may require larger cell sizes, such as an icon cell size of 48x48, pixels. It is important not to build the full resolution variant sets until your customer reviews and initial tests are completed successfully as you could end up wasting a great deal of time at this stage. If possible, build the full "oven-ready" sets only when you have reached agreement on the draft concepts.

During your design process, you will probably create more than one design concept for each icon. Choose the design that YOU consider most meets the functional description provided and be prepared to describe the merits of this design and why it is appropriate. If this draft is rejected, then be prepared to present another of your alternative design concepts. It is not unusual for your customer to reveal additional requirements at this stage and which could invalidate your original design, but could make your earlier ideas more appropriate!

### **COMPLETING THE ICON SETS**

When the initial concepts and designs are approved and have been tested, you will need to complete the different cell size variants and possibly conduct further tests.

Larger sizes (40x40 and 48x48) can be visually richer and contain more detail, but in smaller sizes (24x24, 20x20 and 16x16) do not try to accommodate all the detail or objects from the larger sizes. In the interests of clarity you will need to decide which are the essential elements and omit the others.

One simple rule to follow is to try to represent 1 or 2 objects in the smallest sized icons (16x16 and 20x20) then include additional detail and/or objects as the size increases.

### **USING COLOUR**

Use colour with care. Strong colour is a very powerful means of attention getting and should be reserved specifically for that purpose, e.g. to show a change in status or an alarm condition. For larger icons, use pastel colours and grey shades. However, for smaller sizes (20x20 & 16x16) stronger outlines and a higher ratio

of stronger colours will compensate to the loss of visibility caused by the reduced cell size.

### **COLOUR PALETTES**

The number of colours available on a system is determined by the "bit depth", which is the number of bits of information used to represent a single pixel on the monitor. Standard display monitors that are available these days have 8 bits per pixel (256 colours); 16 bits provides for thousands of colours (65,536) and 24 bits, common on newer systems, provide for millions of colours (16,777,216). The specific colours available on a system are determined by the way in which the target platform allocates colours. Available colours might differ from application to application. (Ref. Java Look and Feel Design Guidelines)

VGA (640x480 pixel) colour displays use the default, 4 bit (16 colour) palette. Changing the co-ordinates of these colours to provide alternatives for this resolution is ineffective since the system will automatically reset them to its default colour palette. Additional colours can be created by mechanically "dithering" (i.e. chequer patterning) the default colours, but be careful, this works best when colours of similar tones are used. Colours of contrasting tones will result in the dither patterns becoming more visible, especially on liquid crystal (LCD), flat panel displays used on many laptop computers. However, cathode ray tube (CRT) displays are more forgiving for dithered patterns, since the phosphor dots still provide some colour spread.

For displays that can display more than 256 colours and above, this form of colour mixing is unnecessary, but you should be prepared to visually match the colours between whichever colour palettes and display resolutions are supported.

On the UNIX systems using the Common Desktop Environment serious colour shifting can occur when non-compatible colours are used. This can also cause usability and application problems.

NOTE: Use the full 256 and above colour palettes for your icons only where it adds real benefits as this number of colour positions will significantly increase the

storage requirements on your product, especially where large numbers of icons are involved.

## LEGAL CONSIDERATIONS

Icons are copyright artistic works of their owners/authors, though the copyright may be freely licensed or never asserted and copyright registration (which is only possible in the USA) may be omitted. Patents are not capable of protecting icons, except again in the US where a special type of patent - a Design Patent, has been used by some companies to try to protect icons.

Although it is often requested, never use product marks and/or logos as icons, or included in icons. These images are not supposed to be arbitrarily re-sized and legal approval must be verified before this type of image can be used.

For icons that IBM designs, individual Certificates of Originality have been completed to record their appearance, creation date and designer. All of the icons in the IBM clip art collection have been certified in this way and are free of third party copyright constraints. IBM does not surrender copyright but licenses others to make copies - under the agreement on the Ease of Use web site.

When re-using existing designs always check that they are not protected.

## OPERATING SYSTEM TECHNICAL REQUIREMENTS

NOTE: At the time of writing, the following are the current known technical and visual requirements. Currently there is no universal file format or colour palette that can be used across all platforms. The .gif format appears to be the most versatile. The web safe colour palette of 216 colours seems to be the most stable.

### Java

<i>Cell size (pixels):</i>	<i>Colours:</i>	<i>Where used:</i>
16x16	16 or 256	Task bar, title bar, menus, tree views
24x24	16 or 256	Tool bar, and buttons
32x32	16 or 256	Desk top, open windows

### Comments:

- Glyphs centred visually within cell
- Various 'looks' are provided with Java. The default set includes the Java/"Metal" look, the Windows look and the CDE/UNIX look. There is also a Macintosh look and feel available via download from the Java/Sun site.
- Default 16 colour palette: Black, white, light and dark red, magenta, light and dark green, tan, two greys, cyan, medium and dark blue, purple and teal, plus transparent. Most of the time Java applications can handle 256 colours in the images.
- File types: .gif = for most instances (always save as "non-interlaced" .gifs); .jpg's are also supported and work well with photo-realistic images where protecting the image quality is important.
- Icon editor: Microangelo Studio (6), Photoshop, Paint Shop Pro.

### Microsoft Windows '95® & NT®

<i>Cell size (pixels):</i>	<i>Colours:</i>	<i>Where used:</i>
16x16	16	Task bar, title bar, menus, details & small icon views
32x32	16	Desk top, open windows & large icon views
48x48	256	Explorer

### Comments:

- Glyphs centred visually within cell
- Most "real world" objects rendered in 3D, i.e. in Isometric projection. The main exceptions are stationary objects such as documents/files and folders.
- Selection state not applied to cell background (therefore glyph could occupy more cell space than an OS/2 icon).
- Default 16 colour palette: Black, white, light and dark red, magenta, light and dark green, yellow, tan, two greys, cyan, medium and dark blue, purple and teal, plus transparent.
- File types: .ico = desktop icons, .bmp = desktop (inanimate) and toolbar icons, .cur = pointer icons.
- Be careful creating large number of 256 colour icons as this can significantly increase the systems storage requirements.
- Icon editor: Microsoft Developer Studio (Visual C++ Developers Studio)

### UNIX/AIX(CDE Desktop)/LINUX

Cell size (pixels):	Colours:	Where used:
16x16	16	- File manager & application manager (low, medium & high resolution) - Sub panels (low resolution)
24x24	16	Toolbars in applications
32x32	16	- File manager & application manager (low, medium & high resolution). - Front panel (low resolution), sub panels (medium & high resolution) - Minimised windows (low resolution)
48x48	16	Front panels and minimised windows (medium & high resolution)

#### Comments:

- Glyphs aligned to left side, touching base and sides of cell.
- Default colour palette comprises: eight colours (black, white, red, green, blue, yellow, cyan, magenta), eight greys, and eight dynamic colours.
- File types: .xpm = coloured, .xbm = black & white, \_m.bm = black & white mask
- Icon editor: Dticon editor, GIMP image editing tool available for all UNIX platforms.

### IBM OS/2® (Warp Merlin)

Cell size (pixels):	Colours:	Where used:
16x16	16	Title bar, details views and templates at 640x480 only
20x20	16	Title bar, details views and templates at higher resolutions
32x32	16	Desk top, and windows at 640x480 only
40x40	16	Desk top, and windows at higher resolutions

#### Comments:

- Glyphs visually centred horizontally in cells
- For OS/2 Warp Merlin, almost all “real world” objects rendered in 3D, i.e. in isometric projection. The main exception is the document/file object (folders are 3D and can “open”).
- Glyphs located at base of cell with set margins (2 pixels at 32x32 and 3 pixels at 40x40)
- Adequate transparent areas required surrounding glyphs to show “selection” status
- 16 colour palette comprises: black, white, 3 greys, dark and light blue, yellow, light and dark red, magenta, cyan, teal, bright green, tan and purple plus screen (transparent) and inverse screen.
- VGA = 640x480 pixels at 16 colours (4 bits per pixel)
- SVGA = 800x600 pixels at 256 colours (8 bits per pixel)
- XGA & 8514A = 1024x768 pixels at 256 colours (8 bits per pixel)
- File types: .ico = desktop icons, .bmp = desktop (inanimate) and toolbar icons, .ptr = pointer icons
- Icon editor: IBM OS/2 Icon Editor

### Apple Macintosh NOTE: needs to be updated or removed

Cell size (pixels):	Colours:	Where used:
Small 16x16	Monochrome, 16 & 256 colours	?
Large 32x32	Monochrome, 16 & 256 colours	?

#### Comments:

- Glyphs are centred visually within the cell
- The new Aqua (Mac OS X) allows for desktop icons to be as large as 128x128 with photo quality
- Default 16 colour palette:
- File types: .icns files
- Icon editor : ResEdit,
- Other resources:  
Icon Machine:  
<http://www.dathorc.com/iconmachine.html>  
CP Icon Edit  
<http://www.ksoft.net/>  
The Icon Factory:  
<http://www.iconfactory.com/>

## REFERENCES

### Books.

- (2) *The Icon Book*, by William Horton, John Wiley & Sons, ISBN 0-471-59900-X
- (3) *Symbol Sourcebook*, by Henry Dreyfuss, McGraw-Hill, ISBN 0-07-017837-2
- *The Windows Interface Guidelines for Software Design*, Microsoft Press, ISBN 1-55615-679-0
- *Global Interface Design*, by Tony Fernandez. *AP Professional*, ISBN 0-12-253790-4
- *Macintosh Human Interface Guidelines*, by Apple Computer Inc, Addison-Wesley Publishing Company, ISBN 0-201-62216-5
- *Visual Interface Design for Windows*, by Virginia Howlett. Wiley Computer Publishing, ISBN 0-471-13419-8
- *The Java Look and Feel Design Guidelines*, copyright Sun Microsystems Inc. Addison-Wesley ISBN 0-201-61585-1
- *Danny Goodman's Macintosh Handbook*, Bantam Books, ISBN 0-201-62216-5
- *The Art of Human-Computer Interface Design*, edited by Brenda Laurel, Addison-Wesley Publishing Company, ISBN 0-201-51797-3
- *The Common Desktop Environment Users Guide*, copyright 1994, IBM, HP, SUN and Novel. Hewlett-Packard order number #B1171-90101

### ISO standards (available from the BSi)

- ISO/IEC 11581-1:2000, Information technology - User system interfaces and symbols - Icon symbols and functions, Part 1: Icons - General
- ISO/IEC 11581-2:2000, Information technology - User system interfaces and symbols - Icon symbols and functions, Part 2: Object icons
- ISO/IEC 11581-3:2000, Information technology - User system interfaces and symbols - Icon symbols and functions, Part 3: Pointer icons
- ISO/IEC 11581-6:1999, Information technology - User system interfaces and symbols - Icon symbols and functions, Part 6: Action icons

### URLs

- (4) IBM Ease of Use (and icon clip art, look under "Downloads") [http://www-3.ibm.com/ibm/easy/eou\\_ext.nsf/publish/5](http://www-3.ibm.com/ibm/easy/eou_ext.nsf/publish/5)

58 [Editor's note 2005/01/21: this URL is not available.]

- (5) Microsoft User Experience and UI Design Resources <http://msdn.microsoft.com/ui/default.asp>
- Apple Mac OS X (Aqua) <http://www.apple.com/macosex/technologies/aqua.html>
- ISO/IEC Joint Technical Committee 1 - Information Technology <http://www.jtc1.org> [Editor's note 2005/01/21: this URL is not available.]
- (6) Microangelo (icon editor) available from: <http://www.impactsoft.com>
- British Standards Institution (BSi) <http://www.bsi.org.uk/>
- Linux icon guidelines: <http://developer.kde.org/documentation/standards/kde/icon-style.html>
- Linux Information: <http://slashdot.org/search.pl?query=Linux+UI+design+guidelines>
- Java Look and Feel Guidelines online version: <http://java.sun.com/products/jlf/dg/index.html> [Editor's note 2005/01/21: this URL is not available.]

### CONTACTS:

Editor's note 2005/01/21: Richard Hodgkinson is no longer at IBM

*Richard Hodgkinson (IBM) & Denise Bell (Tivoli Systems) 2000-09-21*