

EMC[™] Common User Experience (ECUE)

Version 3.5

Mobile Applications Standard



Copyright [©] 2012-2013 EMC Corporation . All rights reserved. Published in USA.

Published September, 2013

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (https://support.emc.com). For documentation on EMC Data Domain products, go to the EMC Data Domain Support Portal (https://my.datadomain.com).

EMC Corporation Hopkinton, Massachusetts 01748-9103 1-508-435-1000 In North America 1-866-464-7381 www.EMC.com

CHAPTER 1

Mobile

This chapter contains the following topics:

٠	Mobile Applications	
٠	Design Best Practices	
٠	Launch	5
٠	Login	7
٠	About & Copyright	8
٠	iOS UI Elements	12
٠	Background Color	15
٠	Gesture & Motion Events	17
٠	Standard Buttons and Icons	
•	Mobile Dashboards	34
٠	Mobile Icon Visual Guidelines (iOS)	41

Mobile Applications

Last modified: ECUE v3.4.2, June 2013

These guidelines are intended for mobile applications which have a companion desktop or mobile application following the ECUE standards.

The guidelines are based on excerpts from the iOS Developer Library > UI Element Usage Guidelines. Specific ECUE guidelines are noted as such.

Figure 1 Example of an EMC mobile application



PDF output ECUE Mobile Application Standard (PDF, 1900KB)

4

Design Best Practices

The following guidelines are based on a Boston UPA event: Mobile Mantras: Experience Design Best Practices for Mobile.

- Mobilization is content, context and task specific.
- Mobile requires a smaller, simpler version of a desktop or web application.

During design:

- Define the use cases
- Order the use cases by the most frequent for a mobile user. Use your best guess, statistical information and usability tests to keep this order updated.
- Simplify: Do your best to make every use case successful in three clicks or less or at a
 page depth of no more than three.
- Touch/Swipe friendly

General guidelines:

- Use a placeholder (text which shows up in gray). Placeholder should remain until the user types something.- (Look to see if we can find some info on the iOs guidelines about this)
- On the iPhone, use all the available width for links, list elements, text inputs and all possible focusable elements. Avoid using columns of elements.
- On the iPad, use a consistent width for links, list elements and text inputs.
- Avoid having content that requires horizontal scroll bars.

Visual guidelines:

- Keep graphics limited
- Maintain visual consistency with your desktop or web application and brand
- Use provided ECUE branding imagery
- Use background colors to separate sections
- Only use the system font on screens, except when embedded in an image, for example the login screen

Content:

- Mobilize content, not just layout
- Search is the most important
- Personalize & localize

Launch

Last modified: ECUE v3.5, September 2013

The user launches the application by clicking the application jewel. This should be the same icon as the application desktop launch icon, and should be approved by the Mobile Go To Market Program.

To enhance the user's experience at application launch, you must provide at least one launch image. A *launch image* looks very similar to the first screen your application displays. iOS displays this image instantly when the user starts your application and until the app is fully ready to use.

Applications should use the ECUE Mobile Branding Image in conjunction with their product name to create a branding image specific to your product. The product name

should be in the MetaBold-Roman font, color #18357A. The font size is dependent on the length of the product name. It should be horizontally-center aligned.

Figure 2 ECUE mobile branding image



As soon as your app is ready for use, your app displays its first screen, replacing the launch image.

Guidelines

- Use your custom branding image (described above), which includes the application name, application logo, and ECUE Mobile Branding Image.
- Because users are likely to switch among applications frequently, you should make every effort to cut launch time to a minimum.
- Always display the gray status bar for iPhone, black status bar for iPad.

6

• We recommend that the branding imagery which is used for your companion mobile app match that of your desktop/browser UI. The Mobile Branding Imagery defined matches the ECUE v3 Branding Imagery, specified in the ECUE Branding standard.

Visual specification

Note

The recently published iOS7 guidelines have not yet been considered as part of the ECUE Mobile guidelines.

Table 1 Supplied files

File name	For use
Default.png	Standard iPhone/iPad DO include status bar region (320x480)
Default@2x.png	Standard iPhone/iPad, high resolution DO include status bar region (640x960)
Default-Portrait.png	iPad, portrait mode do not include status bar region (768x1004)
Default-Portrait@2x.png	iPad, portrait mode, high resolution do not include status bar region (1536x2008)
Default-Landscape.png	iPad, landscape mode do not include status bar region (1024x748)
Default-Landscape@2x.png	iPad, landscape mode, high resolution do not include status bar region (2048x1496)
Default-568h@2x.png	iPhone5 & iPad (5th generation) DO include status bar region

The graphics for implementation are accessed in one of two ways depending on your browser:

- Users of Internet Explorer can simply click the link to https://ecue.lss.emc.com/ styleguideshare/ECUEGraphics/v3/Mobile/Branding.
- Users of other browsers such as Firefox may need to paste the following path into Windows Explorer to access the file share \\ecue.lss.emc.com\styleguideshare \ECUEGraphics\v3\Mobile\Branding.

Login

Created: ECUE v3.4.2, June 2013

- Use gray status bar for iPhone, black for iPad.
- Field placement and behavior is not addressed in this version of the standard.

Guidelines

- Background image with small EMC logo on top right is used.
- Application name is separate image in specific color, font type & size.
- Landscape image for iPhone is not supported.
- The images below are for example only and do not imply standards for the input fields.

7

Figure 3 Login images

|--|

Visual specification

The graphics for implementation are accessed in one of two ways depending on your browser:

- Users of Internet Explorer can simply click the link to https://ecue.lss.emc.com/ styleguideshare/ECUEGraphics/v3/Mobile/Branding.
- Users of other browsers such as Firefox may need to paste the following path into Windows Explorer to access the file share \\ecue.lss.emc.com\styleguideshare \ECUEGraphics\v3\Mobile\Branding.

About & Copyright

8

Created: ECUE v3.4.1, December 2012

The About screen reminds users of the application name and version number, and should include a Copyright statement and optionally other legal notices.

Access to the About & Copyright information

In an iOS application, the application version and access to the about screen are accessed by going to the Settings menu, and tapping the EMC Application name. A settings page specific to the EMC application is displayed. This page should consist of a sub-panel titled **About this Application** which includes:

- The application name
- Access to the About screen

nd ABC 3G 02:24 PM ■	ml ABC 3G 02:24 PM Settings	util ABC 3G 02:24 PM Cettings EMC Application
Heisages Calendar Piotos Conero	Messages	Settings
	Music >	Username jamith
Contacta VauTube Stocka Miga	Video >	Password
	Notes >	Auto Synch Every 10 Minutes
		About this Application
Settigs		Version 1.14
Non Na Soleri Pod	EMC Application	About {App Name}
		Tap About

Figure 4 Accessing About information -- Native iPhone settings

Access via iPhone default Settings page:

- 1. Navigate to Settings option in iPhone home screen
- 2. Tap icon to view standard Settings page. Locate section for EMC Application
- 3. Tap section to view application settings and About information
- 4. Tap the "About [Application Name]" item to display the About screen.





Elements Text Area

- If needed, use a scroll bar.
- EMC Application Name
- Version information is one of the following formats (where build is optional):
 - V#.# (Build #)
 - V#.#.# (Build #)
 - Version #.# (Build #)
 - Version #.#.# (Build #)
- In addition to providing the version information, this screen is the appropriate place if you wish to include labels or sub-menus to display any of the following:
 - EMC Support contact information
 - Link to EMC.com or Powerlink,com
 - License information
 - Legal notices, such as 3rd party component accreditations

Optional iPhone Menu

If there is a lot of information in the text area, menu items can be added to access:

License information

• Legal notices, such as 3rd party component accreditations



Copyright message

- The copyright message should have one of these forms: © YYYY EMC Corporation. All Rights Reserved
 © XXXX-YYYY EMC Corporation. All Rights Reserved Where YYYY or XXXX-YYYY show the year or range of years of publication.
- The copyright mark must be shown as a symbol if available in that language. If the symbol is not available, use (C).
- Place the fixed copyright message in the lower left of the screen.

iOS UI Elements

Created: ECUE v3.3, June 2012

The status bar, navigation bar, tab bar, and toolbar are UI elements that have specifically defined appearances and behaviors in iOS applications. The bars provide familiar anchors to users of iOS-based devices, who are accustomed to the information they display and the types of functions they perform.

ECUE guideline: Each section below will state whether the bar is mandatory or optional per the ECUE guidelines.

Status bar

The status bar displays important information about the device and the current environment.

ECUE guideline: The status bar is required and must be present in every EMC mobile application.

Figure 5 Status bar

Appearance and behavior

The status bar always appears at the upper edge of the device screen (in all orientations) and contains information people need, such as the network connection, the time of day, and the battery charge.

ECUE guideline: On iPhone, the status bar is always gray. On iPad, the status bar is always black.

Guidelines

Although you don't use the status bar in the same way that you use other UI elements, it's important to understand its function in your app.

Consider hiding the status bar (and all other app UI) while people are actively viewing full-screen media. If you do this, be sure to allow people to retrieve the status bar (and appropriate application UI) with a single tap. Unless you have a very compelling reason to do so, it's best to avoid defining a custom gesture to redisplay the status bar because users are unlikely to discover such a gesture or remember it.

Don't create a custom status bar. Users depend on the consistency of the system-provided status bar. Although you might hide the status bar in your app, it's not appropriate to create custom UI that takes its place.

When appropriate, display the network activity indicator. The network activity indicator can appear in the status bar to show users that lengthy network access is occurring. To learn how to implement this indicator in your code, see Network Activity Indicator.

ECUE guideline: When performing network access commands, display the network activity indicator.

Navigation bar

A navigation bar enables navigation through an information hierarchy and, optionally, management of screen contents.

ECUE guideline: The navigation bar is optional, based on the application.

Two visualizations are available for the navigation bar.

- For mobile applications which:
 - use iOS4 and before and have a companion desktop or web application
 - do not have a companion desktop or web application

Figure 6 Navigation bar visualization 1



 For mobile applications which use iOS 5 and later and have a companion desktop or web application:

Figure 7 Navigation bar visualization 2



Appearance and behavior

A navigation bar appears at the upper edge of an application screen, just below the status bar.

ECUE guideline: The navigation bar must display the title of the current screen or view, centered along its length.

When navigating through a hierarchy of information, users tap the back button to the left of the title to return to the previous screen. Otherwise, users can tap content-specific controls in the navigation bar to manage the contents of the screen.

ECUE guideline: The navigation bar must match the ECUE visual specification.

On iPhone, changing the device orientation from portrait to landscape can change the height of the navigation bar automatically. On iPad, the height and translucency of a navigation bar does not change with rotation.

On iPhone, a navigation bar always displays across the full width of the screen. On iPad, a navigation bar can display within a view, such as one pane of a split view, that does not extend across the screen.

Guidelines

You can use a navigation bar to enable navigation among different views, or provide controls that manage the items in a view.

Use the title of the current view as the title of the navigation bar. When the user navigates to a new level, two things should happen:

- The bar title should change to the new level's title.
- A back button should appear to the left of the title, and it should be labeled with the previous level's title.

ECUE guideline: The system font should always be used in the navigation bar.

Consider putting a segmented control in a navigation bar at the top level of an application. This is especially useful if doing so helps to flatten your information hierarchy and makes it easier for people to find what they're looking for. If you use a segmented control in a navigation bar, be sure to choose accurate back-button titles. (For usage guidelines, see Segmented Control.).

Avoid crowding a navigation bar with additional controls, even if there appears to be enough space. The navigation bar should contain no more than a view's current title, the back button, and one control that manages the view's contents.

When a segmented control in the navigation bar, the bar should not display a title and it should not contain any controls other than the segmented control.

Use system-provided buttons according to their documented meaning:

• For more information, see Standard Buttons for Use in Toolbars and Navigation Bars.

- Future versions of this standard will include additional ECUE standard buttons for use in toolbars and navigation bars.
- If you decide to create your own navigation bar controls, see "Icons for Navigation Bars, Toolbars, and Tab Bars" for advice on how to design them. Be sure to follow the ECUE Icon Guidelines when doing so.

If appropriate, customize the appearance of navigation-bar controls. In particular, if you supply a custom background appearance for the navigation bar, you should consider supplying a coordinating appearance for the bar controls.

Make sure that a customized back button still looks like a back button. Users know that the standard back button allows them to retrace their steps through a hierarchy of information. If, for example, you create a custom back button that doesn't have a pointed side, users won't instantly know what it does.

Don't create a multisegment back button.



Creating a multisegment back button (as in the example above) causes several problems:

- The extended width of a multisegment back button does not leave room for the title of the current screen.
- There is no way to program such a multisegment button to indicate the selected state of an individual segment.
- The more segments there are, the smaller the hit region for each one, which makes it difficult for users to tap a specific one.
- Choosing which levels to display as users navigate deeper in the hierarchy is problematic.

If you think users might get lost without a multisegment back button that displays a type of breadcrumb path, it probably means that users must go too deeply into the information hierarchy to find what they need. To address this, you should flatten your information hierarchy.

On iPhone, take into account the automatic change in navigation bar height that occurs on device rotation. In particular, make sure that your custom navigation bar icons fit well in the thinner bar that appears in landscape orientation. Don't specify the height of a navigation bar programmatically; instead, you can take advantage of the UIBarMetrics constants to ensure that your content fits well.

Visual specification

The graphics for implementation are accessed in one of two ways depending on your browser:

- Users of Internet Explorer can simply click the link to https://ecue.lss.emc.com/ styleguideshare/ECUEGraphics/v3/Mobile/Bars.
- Users of other browsers such as Firefox may need to paste the following path into Windows Explorer to access the file share \\ecue.lss.emc.com\styleguideshare \ECUEGraphics\v3\Mobile\Bars.

Background Color

Created: ECUE v3.3, June 2012

We recommend any of the three colors for use as the background color of views. Use the same color consistently through an application, unless there is a reason to deviate, for example the Settings screen.

Gray background: #EBEBEB

HL. AT&T 3G	3:40 PM	🔋 76 % 🚍
Back	ProSphere	Export

Blue background: #E0EDF8



White background: #FFFFFF



Gesture & Motion Events

Created: ECUE v3.4.1, December 2012

Mobile applications respond to gestures, not clicks. Gestures are specific finger movements that operate touch devices. For example, users tap a button to activate it, flick or drag to scroll a long list, or pinch open to zoom in on an image.

Users are comfortable with standard gestures because the built-in apps use them consistently. Their experience using built-in apps gives them a familiar set of gestures that they expect to be able to use successfully in most other apps.

To promote a consistent user experience, use standard gestures consistently throughout your application.

Тар

Tapping is one of the most common and basic gestures.

To tap, users make a quick up-and-down motion with a finger, lightly striking the screen.



Examples:

• A single tap opens apps, launches controls, makes choices from menus, positions the insertion point in a note or document, and more.

When to use:

• The tap gesture should be available on charts and maps when providing drill-down capability.

iOS Event Handler:

• UITapGestureRecognizer. With the UITapGestureRecognizer, you can specify the number of taps and number of touches. For example, you could specify that two fingers are required, tapping twice with both fingers.

Double tap

A double tap consists of two quick taps. Tap an object twice in succession to execute a double-tap.



Examples:

- Zoom in or out on text, on a map, web page or photo; a second double-tap will zoom further into a map, or zoom back out of a photo or web page.
- Highlight a word ready for editing or copying.

When to use:

 The double tap gesture should be available on maps, charts, pictures and graphics when providing the ability to zoom in or out.

iOS Event Handler:

 UITapGestureRecognizer. With the UITapGestureRecognizer, you can specify the number of taps and number of touches. For example, you could specify that two fingers are required, tapping twice with both fingers.

Pan

The pan gesture is used to scroll, move or pan. To drag, users place a finger on the screen and move it in the desired direction without lifting it from the screen.



Pan is also known as Drag.

Examples:

• Pan is used to drag an element.

When to use-the pan gesture should be available when:

- Displaying historical data in charts, so you can view how the chart looked earlier in the week
- Moving to a different section of a map, so you can see other areas of the map without changing the target location

iOS Event Handler:

UIPanGestureRecognizer

Swipe

Swiping is one of the primary navigation tools.



Horizontal swipe gesture (left/right) examples include:

- Navigating between screens of apps
- Moving between photos in an album
- Moving from one screen to another within an app
- Revealing the Delete button (or other inline editing capabilities) in a table-view row A vertical swipe gesture (up/down) example includes:
- Reading text in Safari, iBooks, Newstand or elsewhere.

When to use:

• The swipe gesture should be available on a dashboard when multiple dashboards are available.

iOS Event Handler:

UISwipeGestureRecognizer

Flick

The flick gesture is used to scroll or pan quickly. To flick, users place a finger on the screen and quickly swipe it in the desired direction.



A flick is just like a swipe, only faster. Inertial scrolling is supported so that the faster or slower the finger is moved, the faster or slower the content will move.

Flick gesture examples include:

- Scrolling or panning quickly or to get to the next item in a scrollable view
- Flicking down from the status bar at the top of the Home screen to display the Notification Center - new emails, news alerts, messages, Facebook updates and so on
- Swiping or flicking up to hide the notifications once more.

When to use:

 This gesture should be available on lists of data when providing the ability to scroll quickly.

Pinch

The pinch gesture is a two-finger user gesture used to zoom in and out. To zoom in or to open something, place your thumb and index finger, pinched together on screen and spread them apart (pinch open). To zoom out, do the reverse (pinch close).



Example:

• The pinch gesture can be used to expand a graphic, for example, to zoom into a picture of a rack.

When to use:

• This gesture should be available on maps, charts, pictures, and graphics when providing the ability to zoom in or out.

iOS Event Handler: UIPinchGestureRecognizer

Long Press (Tap & Hold or Touch & Hold)

The long press gesture is used to display an information bubble, magnify content under the finger, or to perform specifications in built-in applications and features.



To long press, users touch the screen, leaving their finger motionless until the information is displayed or the action occurs.

Examples—The long press gesture is used in many scenarios:

- Use the long press gesture on an app icon to initiate it to wobble. In that state, tap
 and drag the icon to move it, or tap its Delete button to remove it from your device. Or
 if an app stops functioning or seems sluggish: double-press the Home button, tap
 and hold any of the running apps displayed in the multitasking bar and then tap an
 app's Delete button to remove it from memory.
- Use the long gesture on a word within a note to highlight it and then drag the editing handles to select surrounding words for editing. You can tap, hold, and while holding down, drag your finger to increase or decrease the selection.
- In editable or selectable text, the long press gesture is used to display a magnified view for cursor positioning
- The long press gestures can be used in a map, web page, photo or on a variety of other objects to select and move them around the screen.

When to use:

 The long press gesture should be available in a dashboard scenario to put the dashboard in edit mode. Touching and holding the view block makes the view blocks wiggle, and that means the user can dynamically (drag) the view block to another location. When in edit mode, view blocks should have an "x" to enable deletion.

iOS Event Handler: UILongPressGestureRecognizer

Rotate

Some elements can be rotated with two or more fingers. Place two fingers on the screen and make a circular gesture, clockwise or counterclockwise.



Example:

• "Undo/Reverse" on **Paper** for iPad iOS Event Handler: UIRotationGestureRecognizer

Four or Five Finger Pinch

The four- or five-finger pinch gesture quickly returns you from an application to the home screen. This accomplishes the same thing as clicking the Home button.



Start with four or five fingers outward, and then pinch them together. Depending on the speed of your pinch, you can either slowly shrink the app until it disappears into the OS home screen, or do a quick pinch so that the app disappears at the same speed as it would if you clicked the Home button normally. This gesture has no reversal option.

The 4 and 5 finger gestures are limited to the tablet.

When to use:

• We recommend not using this gesture, since it is already in use.

Four or Five Finger Horizontal Swipe

Move quickly between your most recently used apps with a four- or five-finger horizontal swipe. For example, if you're in Safari and want to switch to another open app, you can perform a four- or five-finger horizontal swipe left or right to move from one app to another; this is similar to the one-finger swipe you use to move between home screens. You can swipe only between apps that have recently been used; to see those (and which order they've been used in), pull up the multitasking bar by double-tapping the Home button or by performing the four- or five-finger vertical swipe.



As you swipe, the app that is active will follow your fingers and move off the screen in the direction you are swiping. As it slides off, the next app will begin to crawl in from the other side of the screen until you swipe far enough for it to snap to center. You can swipe quickly to jump through apps almost immediately, or drag slower to fully appreciate the animation.

The 4 and 5 finger gestures are limited to the tablet.

When to use:

• We recommend not using this gesture, since it is already in use.

Four Finger Vertical Swipe

Like a double-tap on the Home button, a four- or five-finger upward swipe will pull up the multitasking bar along the bottom of the screen. To return it, swipe downward to hide the bar (or single-tap anywhere above it).



When to use:

• We recommend not using this gesture, since it is already in use.

Motion Events

In addition to gestures, users can move the device itself to perform certain tasks. We refer to these movements as motion events. Many mobile devices (including iPhone & iPad) generate motion events when users move the device in a certain way, such as shaking it or tilting it.

To promote a consistent user experience, use motion events consistently throughout your application.

Shake



Examples of the shake gesture include:

- • To initiate an undo or redo action
- ◆ · To initiate a refresh
- • To "shuffle" in Words with Friends for iPhone and iPad

When to use:

 The shake gesture can be used to perform a refresh. This option should be implemented with caution, and if implemented, consider having a "shake to..." on/off toggle in the Settings area of the app.

Device Orientation Changes

Shift from portrait to landscape mode. When you turn the device from a vertical orientation to a horizontal one, its display shifts from portrait (vertical) to landscape (horizontal).



Only some applications support this mode shift.

When the orientation of a device changes, if the new orientation is supported, the view will update accordingly. Consider supporting alternate orientations to:

- Show or hide views that are specific to a particular orientation.
- Adjust the position or size of views based on the new orientation.
- Offer more screen real estate for content to be viewed.

Best Practices: Designers need to decide which orientations they support: portrait, landscape, upside down portrait and upside down landscape. We recommend supporting all 4 orientations.

The following examples illustrate the view type changing based on device rotation, and how a view type change and appropriate use of screen real estate positively affects the experience.

• Below is an example of how device rotation affects **Calendar** for the iPhone.

Caler	idars	All C	alen	uars		0
◀ Sun	Mon	Octo	Wed	2012 Thu	Fri	► Sat
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

Figure 8 iPhone Calendar in portrait orientation

Today	List	Day	Month	±•

Figure 9 iPhone Calendar in landscape orientation



• Below is an example of how device rotation affects **Stocks** for the iPhone.

Figure 10 iPhone Stocks in portrait orientation



Figure 11 iPhone Stocks in landscape orientation



Standard Buttons and Icons

Last modified: ECUE v3.4.2, June 2013

Mobile

To promote a consistent user experience, use iOS and ECUE standard buttons for in navigation bars and toolbars, and icons in tab bars.

Be sure to use UI element consistently, and avoid using the standard iOS and ECUE buttons and icons described in this section to represent actions other than those for which they are designed.

For more information, see iOS Human Interface Guidelines: System-Provided Buttons and Icons.

Standard Buttons for Use in Toolbars and Navigation Bars

A number of standard buttons exist for use in toolbars and navigation bars:

- Standard iOS buttons
- Standard ECUE buttons
- Bordered action buttons for use in navigation bars and toolbars

The ECUE buttons can be found on the ECUE file share at \\ecue.lss.emc.com \styleguideshare\ECUEGraphics\v3\MobileAppIcons.

The following excerpt was taken from *iOS Human Interface Guidelines: System-Provided Buttons and Icons* in September 2012.

Button	Name	Meaning
1	Action	Open an action sheet that allows users to take an application-specific action.
0	Camera	Open an action sheet that displays a photo picker in camera mode.
N	Compose	Open a new message view in edit mode.
B	Bookmarks	Show application-specific bookmarks.
d	Search	Display a search field.
+	Add	Create a new item.
	Trash	Delete current item.
Ē	Organize	Move or route an item to a destination within the application, such as a folder.
→	Reply	Send or route an item to another location.
×	Stop	Stop current process or task.
C	Refresh	Refresh contents (use only when necessary; otherwise, refresh automatically).

Table 2 Standard iOS buttons for use in toolbars and navigation bars

Table 2 Standard iOS buttons for use in toolbars and navigation bars (continued)

Button	Name	Meaning
	Play	Begin media playback or slides.
	FastForward	Fast forward through media playback or slides.
	Pause	Pause media playback or slides (note that this implies context preservation)
	Rewind	Move backwards through media playback or slides.

Table 3 Standard ECUE buttons for use in toolbars and navigation bars

Button	Name	Meaning
*	Home	Navigate to the home page of the application.
\$	Settings	Open an action sheet that allows users to view and edit application-specific settings.
	List	Display information in list format
16	Chart	Display information in chart format.
55	Heat Map	Display information in heat map format.
	Table	Display information in table format.

In addition to the buttons shown in the above tables, you can also use the iOS systemprovided Edit, Cancel, Save, Done, Redo, and Undo buttons shown in the following table to support editing or other types of content manipulation in your application.

Table 4 Bordered action buttons for use in toolbars and navigation bars

Button	Name	Meaning
Edit	Edit	Enter an editing or content-manipulation mode.
Cancel	Cancel	Exit the editing or content-manipulation mode without saving changes.
Save	Save	Save changes and, if appropriate, exit the editing or content- manipulation mode.

Table 4 Bordered action buttons for use in toolbars and navigation bars (continued)

Button	Name	Meaning		
Done	Done	Save changes, if any, and exit the current mode.		
Undo	Undo	Undo the most recent action.		
Redo	Redo	Redo the most recent undone action.		

Note

In general, there is never more than one blue button in a bar. The blue button represents the action that the user is most likely to take in the current context.

The buttons listed in the table above are suitable for both navigation bars and toolbars, and are available in the bordered style only. If you specify the plain style for one of these buttons, it is converted to the bordered style.

Standard icons for use in tab bars

A number of standard buttons exist for use in toolbars and navigation bars:

- Standard iOS buttons
- Standard ECUE buttons

Table 5 Standard iOS buttons for use in the tabs of a tab bar

Button	Name	Meaning
Ш	Bookmarks	Show application-specific bookmarks.
1	Contacts	Show contacts.
\bullet	Downloads	Show downloads.
*	Favorites	Show user-determined favorites.
*	Featured	Show content featured by the application.
Ŀ	History	Show history of user actions.
	More	Show additional tab bar items

Table 5 Standard iOS	buttons for use	in the tabs o	of a tab bar	(continued)
				(00

Button	Name	Meaning
H	MostRecent	Show the most recent item.
业	MostViewed	Show items most popular with all users.
Ŀ	Recents	Show the items accessed by the user within an application- defined time period.
Q	Search	Enter a search mode.
*	TopRated	Show the highest-ranked items, as determined by the user.

Table 6 Standard ECUE buttons for use in the tabs of a tab bar

Button	Name	Meaning
Home	Home	Navigate to the home page of the application.
Settings	Settings	Open an action sheet that allows users to view and edit application-specific settings.

Standard iOS buttons for use in table rows and other UI elements

iOS provides the buttons described in the following table for use in table rows and other elements.

Table 7 Standard buttons for use in table rows and other UI elements

Button	Name	Meaning	
•	ContactAdd	Display a people picker to add a contact to an item.	
\bigcirc	DetailDisclosure	Display a new view that contains details about the current item.	
ð	Info	Flip to the back of the view to display configuration options or more information. Note that the Info button is also available as a light-colored "i" in a dark circle.	

Mobile Dashboards

Created: ECUE v3.4.1, December 2012

A number of layout options are listed below along with guidelines for when to use the options. These are not the only supported layouts, but are suggested for the use cases we have identified at this point.

Consider the following before selecting a particular layout:

- Does the user need to drill down for more information without losing context?
- Are the KPIs (Key Performance Indicators) being shown in this dashboard?

For an overview and best practices on dashboards, refer to Certificates: Dashboards.

Grid layout

The grid layout supports a number of square or rectangular tiles placed on a grid.



When to use:

• Use the grid layout when you want to show multiple KPIs in a visual manner. This layout allows information to be delivered cleanly and visually.

Behavior:

 Tap to drill down. The number of drill down levels should be based on the application. The number should be minimized, unless having more levels adds value to the user. For example, do not limit the number of drill down levels in ITOI POC since it corresponds to the group hierarchy defined by the user.

Layout & design:

- Do not use horizontal scrolling in this layout. You can use vertical scrolling although we discourage it.
- Although currently no visual guidelines specifications exist for this layout, keep the following in mind:
 - Use colors consistently across all tiles to convey a particular state
 - · Use font, font color and font size consistently for common states and components
- Below is an example of the grid layout being used with a page indicator.



List layout (drilldown)

The list layout supports a vertical listing of information displaying values and/or charts that the user can tap to navigate to another page for more data.

Attribute1	value	>
Attribute2	value)
Backup Failures		>
Attribute3	value)
Availalble Space		,
		<i>′</i>
Attribute4	value	>

When to use:

• Use the list layout when the general purpose is to allow the user to look at specific data points or metrics one at a time.

Behavior:

• Tap an item to drill down to view another page; eventually you will get to a page with the detailed information.

Layout & design:

- Do not use horizontal scrolling in this layout. You can use vertical scrolling.
- Although no visual specifications currently exist for this layout, keep the following in mind:
 - Use colors consistently across all tiles to convey a particular state.
 - Use font, font color and font size consistently for common states and components.
 - When labels are displayed on the left and values are displayed on the right, leftalign the labels and right-align the values.
 - Use the same height for all rows.
 - Recommend right arrow treatment in all cases?
 - Below is an example of the list layout used with a:
 - Tab bar
 - Page indicator

EMC Applicati	M 🔳
	on Name
Attribute1	value 💙
Attribute2	value 🕨
Backup Failures	
	,
Attribute3	value 🕨
Available Space	
	/
Attribute4	value 🕨

Accordion (collapsible section) layout

The accordion layout supports a vertical listing of information displaying labels, values and/or charts in an accordion-like component that the user can tap to expand and retrieve more data.

Attribute1	123 GB
Attribute2	value
Backup Failures	
Attribute3	value
Availalble Space	
Attribute4	value

When to use:

• Use the accordion layout to allow drilldown to compare KPIs against each other without losing context.

Behavior:

 Tap an item to expand the area to retrieve more data. Tap the same area to collapse it. More than one area can be expanded at the same time. We recommend that only one area be expanded at a time, unless there is a need for comparison and there is appropriate real estate to do so.

Layout & design:

- Do not use horizontal scrolling in this layout. You can use vertical scrolling, particularly when one or more items are expanded.
- Although no visual guidelines currently exist for this layout, keep the following in mind:
 - Use colors consistently across all tiles to convey a particular state
 - · Use font, font color and font size consistently for common states and components
 - When labels are displayed on the left and values are displayed on the right, leftalign the labels and right-align the values.
- Below is an example of the list layout being used with a:
 - Tab bar
 - Page indicator



Examples:

• Yahoo Weather (iPhone)—In this example, only one area is expandable at a time. The user can scroll down to see all times which are followed by the remaining days of the week.





• IBM XIV (iPhone)—In this example, the user can expand as many items as they wish.





Multiple dashboards

Applications with multiple dashboards should use multiple pages, rather than tabs. Additional dashboard pages are accessed by the swipe gesture, and their visual cue is page indicators.

Multiple dashboards allow you to deliver information to users in reasonable chunks. Examples of when to use multiple dashboards are:

- When grouping by functionality (performance, compliance, alerts)
- When monitoring different states of data
- When monitoring data per object (per array)
- When users create dashboards

Behavior:

 Use a horizontal swipe to allow access to additional dashboards. Do not use the tab bar for this. use a page indicator to indicate that there are multiple dashboards. iOS also allows tapping on a circle to go directly to that page.

Editable dashboards

Not covered in this version of the guidelines.

Mobile Icon Visual Guidelines (iOS)

Created: ECUE v3.4.1, December 2012

These guidelines are intended to be used by visual designers when they create icons for the Mobile devices. The goal of these guidelines is to have consistency in icon design across all EMC mobile icons. Separate topics give additional guidelines for specific types of icon:

- Icons for navigation bars, toolbars, and tab bars on page 44
- App icon (required for all apps) (future)
- App icon for the App Store (required for all apps) (future)
- Small icon for Spotlight search results and Settings (recommended) (future)
- Web clip icon (recommended for web apps and websites) (future)

When to use

Use the system-provided buttons and icons to represent standard tasks in the apps, as much as possible. However, do not use system-provided buttons and icons to initiate actions significantly different from their standard usage. For a complete list of standard buttons and icons, and guidelines on how to use them, see "System-Provided Buttons and Icons" (iOS Human Interface Guidelines, page 145).

In addition to system-provided icons, every app needs custom icons.

- Every application needs an app icon.
- Apps should also provide an icon for iOS to display in Spotlight search results and, if necessary, in Settings (recommended).
- Some apps need custom icons to represent app-specific content, functions, or modes in navigation bars, toolbars, and tab bars.

These custom icons must meet specific criteria so that iOS can display them properly.

Style

- Beautiful, compelling icons and images are a fundamental part of the iOS user experience. Far from being merely decorative, the icons and images in your app play an essential role in communicating with users.
- For the best results, enlist the help of a professional graphic designer. An experienced graphic designer can help you develop an overall visual style for your app and apply that style to all the icons and images in it. See ECUE Icon Process for details.
- Use universal imagery that people will easily recognize. Avoid focusing on a secondary or obscure aspect of an element.
- Avoid using "greek (Lorem Ipsum) " text or wavy lines to suggest text. If you want to show text in your icon, but you don't want to draw attention to the words themselves, start with actual text and make it hard to read by shrinking it or doubling the layers.
- Create an idealized version of the subject rather than using a photo. Although it can be appropriate to use a photo (or a screenshot) in an app icon, it's often better to augment reality in an artistic way. Creating an enhanced version can help you emphasize the aspects of the subject that you want users to notice.
- If your app has a very recognizable UI, consider creating a refined representation of it, instead of using an actual screenshot of your software in the app icon. Creating an enhanced version of the UI is particularly important when users could confuse a large version of the icon with the actual interface of the app.
- Avoid using iOS interface elements in your artwork. You don't want users to confuse your icons or images with the iOS UI.
- Don't use replicas of Apple hardware products in your artwork. The symbols that represent Apple products are copyrighted and cannot be reproduced in your icons or images. In general, it's a good idea to avoid replicas of any specific devices in your artwork, because these designs change frequently and icons or images that are based on them can look dated. If no EMC-provided icon is available, you may consider using stock icons (e.g. hard disk icon).

- Don't reuse iOS app icons in your interface. It can be confusing to users to see the same icon used to mean slightly different things in multiple locations throughout the system.
- Portray real substances accurately. Icons that represent real objects should also look as though they are made of real materials and have real mass. Photo-realistic icons accurately replicate the characteristics of substances such as fabric, glass, paper, and metal, and convey an object's weight and feel.
- Use transparency when it makes sense. Transparency in an image can help depict glass or plastic, but it can be tricky to use convincingly. You should not use transparency in your app icon.

See Icons for navigation bars, toolbars, and tab bars on page 44 for specific details.

Light source & shadows

- Flat icon lighting should come from upper left. think should be silhouette effect
- Do not use drop shadows

Color & transparency

- iPhone toolbar icons should have no color.
- Use color and shadow judiciously(sensibly) to help the icon tell its story. Don't add color just to make the icon more colorful. Also, smooth gradients typically work better than sharp delineations of color.
- The standard bit depth for icons and images is 24 bits (8 bits each for red, green, and blue), plus an 8-bit alpha channel.
- You do not need to constrain your palette to web-safe colors. Although you can use alpha transparency in the icons you create for navigation bars, toolbars, and tab bars, do not use it in app icons.

lcon type	Size for iPhone 5 & iPod Touch (5th generation)	Size for high- resolution iPhone and iPod Touch	Size for iPhone and iPod Touch	Size for high- resolution iPad	Size for iPad
App icon (required for all apps)	114 x 114	114 x 114	57 x 57	144 x 144	72 x 72
App icon for the App Store (required for all apps)	1024 x 1024 (recommended)	1024 x 1024 (recommended)	512 x 512	1024 x 1024 (recommended	512 x 512
Launch image (required for all apps)	640 x 1136	640 x 960	320 x 480	1536 x 2008 (portrait) 2048 x 1496 (landscape)	768 x 1004 (portrait) 1024 x 748 (landscape)
Small icon for Spotlight search results and Settings (recommended)	58 x 58	58 x 58	29 x 29	100 x 100 (Spotlight search results) 58 x 58 (Settings)	50 x 50 (Spotlight search results) 29 x 29 (Settings)
Web clip icon (recommended for web apps and websites)	114 x 114	114 x 114	57 x 57	144 x 144	72 x 72

Size requirements

Icon type	Size for iPhone 5 & iPod Touch (5th generation)	Size for high- resolution iPhone and iPod Touch	Size for iPhone and iPod Touch	Size for high- resolution iPad	Size for iPad
Toolbar and navigation bar icon (optional)	Approximately 40 x 40	Approximately 40 x 40	Approximately 20 x 20	Approximately 40 x 40	Approximately 20 x 20
Tab bar icon (optional)	Approximately 60 x 60	Approximately 60 x 60	Approximately 30 x 30	Approximately 60 x 60	Approximately 30 x 30
Default Newstand cover icon for the App Store (required for Newstand apps)	At least 1024 pixels on the longest edge	At least 1024 pixels on the longest edge	At least 512 pixels on the longest edge	At least 1024 pixels on the longest edge	At least 512 pixels on the longest edge

Naming conventions

iOS icons that appear on regular and retina displays must follow the following naming conventions:

- MyImage .png Regular display
- MyImage@2x.png Retina display

The "MyImage" should be replaced with the name of your image. The resolution of the "@2x" file has to be double that of the ".png" file (for example 100×100 and 200×200). If you follow the above naming conventions, and add the files to your project, your app will automatically decide which file it needs for the device it is running on. This will make your app look better on Retina device.

The same naming conventions must also be used for navigation and tab bar graphics, launch graphics, buttons, etc. Launch graphics, in addition to the @2x convention, must have specific names, such as:

Names for Regular display:

- Default-Portrait.png
- Default-PortraitUpsideDown.png
- Default-Landscape.png
- Default-LandscapeLeft.png
- Default-LandscapeRight.png

Names for Retina display:

- Default-Portrait@2x.png
- Default-PortraitUpsideDown@2x.png
- Default-Landscape@2x.png
- Default-LandscapeLeft@2x.png
- Default-LandscapeRight@2x.png

Designing icons

Tips for Creating Great Artwork for the Retina Display:

- The Retina display allows you to display high-resolution versions of your art and icons. If you merely scale up your existing artwork, you miss out on the opportunity to provide the beautiful, captivating images users expect. Instead, you should rework your existing image resources to create large, higher quality versions that are:
 - Richer in texture. For example, in the high-resolution versions of the Settings and Contacts icons, the metal and paper textures are clearly visible.



• More detailed. For example, in the high-resolution versions of the Safari and Notes icons, you can see details such as the accurate contours of the continents behind the compass and the torn paper left by the previous note.



• More realistic. For example, the high-resolution versions of the Compass and Photos icons combine rich texture and fine details to create realistic portrayals of a compass and a photograph.



• Even though bar icons are simpler than app or document icons, you should consider adding details as you create high-resolution versions of them. For example, the artists tab bar icon in the Music app is a streamlined silhouette of a singer. The high-resolution version of this icon is recognizably the same icon, but includes greater detail.



- The following techniques can help you get great results as you create a high-resolution version of your artwork.
 - Start by scaling up your original artwork to 200% using the "nearest neighbor" scaling algorithm. This works well if the original artwork was not created with vector shapes and does not include layer effects. The result is a large, pixilated image on top of which you can draw matching high-resolution art. This is a good way to begin because it allows you to preserve the original layout of your design.

Note

If the original artwork was created with vector shapes, or it includes layer effects, you can use the default scaling algorithm instead of the nearest neighbor algorithm.

- Add detail and depth. Don't hesitate to draw very small elements, because the high-resolution version of your artwork allows much more room for fine details. For example, a 1-pixel dot in your original image becomes a 4-pixel dot (that is, 2 x 2 pixels) in the larger version.
- Consider softening scaled-up elements. If, for example, you have a sharp, 1-pixel dividing line in your original artwork, it might have the boldness you want when you leave it scaled up to a 2-pixel line. But for some lines and elements, you might want to soften the scaled results by feathering or even leaving the element at the smaller size.
- Consider adding blur for better results in effects such as engravings and drop shadows. For example, text engraving is typically done by shifting a duplicate image of the text by 1 pixel. Scaled up, this shift would result in an engraving width of 2 pixels, which is likely to look very sharp and unrealistic at a higher resolution. To improve this, you can leave the shift as-is (that is, at 1 pixel), but add a 1-pixel blur to soften the engraving. This still results in a 2-pixel wide engraving effect, but the outer pixel now looks more like it is only half a pixel wide, which results in a better sense of dimensionality.

Note

If you're creating resizable images to draw on a Retina display, you also need to supply high-resolution versions of your images. For example, you would also supply a solid-color 2 x 2 pixel image, or a gradient image that has a width of 2 pixels.

Evaluating icons

- Look at all sizes.
- Look at the family of icons together to evaluate family resemblance, optical balance and distinction.
- Look at icons in context to evaluate distinction, relative weights (make sure that one doesn't dominate) and visibility.
- Consider cases that may not be used now, but could be in the near future.
- Consider doing usability testing of icons.

Icons for navigation bar, toolbar, & tab bar

These guidelines are intended to be used by visual designers when they create icons for the Navigation Bar, Toolbar, and Tab bar of a mobile device.

 A navigation bar appears at the upper edge of an app screen, just below the status bar. It enables navigation through an information hierarchy and, optionally, management of screen contents.



 On an iPhone, a toolbar always appears at the bottom edge of a screen or view, but on iPad it can instead appear at the top edge. A toolbar contains controls that perform actions related to objects in the screen or view.



• A tab bar appears at the bottom edge of the screen and should be accessible from every location in the app. A tab bar gives the users the ability to switch between different subtasks, views, or modes.



Style

Before you create the art for your icon, you need to spend some time thinking about what it should convey. As you consider designs, aim for an icon that is:

- Simple and streamlined. Too many details can make an icon appear sloppy or indecipherable.
- Not easily mistaken for one of the system-provided icons. Users should be able to distinguish your custom icon from the standard icons at a glance.
- Readily understood and widely acceptable. Strive to create a symbol that most users will interpret correctly and that no users will find offensive.
- Dos and Don'ts:
 - As often as possible, you should use the system-provided buttons and icons to represent standard tasks in your app. For a complete list of standard buttons and icons, and guidelines on how to use them, see "System-Provided Buttons and lcons."
 - Avoid using images that replicate Apple products in your designs. These symbols are copyrighted and product designs can change frequently.
 - Use pure white with appropriate alpha transparency.
 - Do not include a drop shadow.
 - Use anti-aliasing.
 - If you decide to add a bevel, be sure that it is 90° (to help you do this, imagine a light source positioned at the top of the icon).

Color, shadows & transparency

All custom icons provided for a navigation bar, toolbar,or tab bar are also known as template images, because iOS uses it as a mask to create the icon you see in your app. It is not necessary to create a full-color template image.

After you've decided on the appearance of your icon, follow these guidelines as you create it:

- Use pure white with appropriate alpha transparency.
- Do not include a drop shadow.
- Use anti-aliasing.
- If you decide to add a bevel, be sure that it is 90° (to help you do this, imagine a light source positioned at the top of the icon).

Size requirements

For toolbar and navigation bar icons on iPhone, iPod touch, and iPad, create an icon in the following sizes:

- About 20 x 20 pixels
- About 40 x 40 pixels (high resolution)

For tab bar icons on iPhone, iPod touch, and iPad, create an icon in the following sizes:

- About 30 x 30 pixels
- About 60 x 60 pixels (high resolution)

Level of detail & granularity

- Embrace simplicity. Avoid cramming lots of different images into your icon. Try to use a single object that expresses the essence of your app. Start with a basic shape and add details cautiously. If an icon's content or shape is overly complex, the details can become confusing and may appear muddy at smaller sizes.
- For Retina display add detail and depth. Don't hesitate to draw very small elements, because the high-resolution version of your artwork allows much more room for fine details. For example, a 1-pixel dot in your original image becomes a 4-pixel dot (that is, 2 x 2 pixels) in the larger version.

Designing icons

- If both iPad and iPhone apps are being developed, the iPhone app should match the iPad app.
- Don't include text or culturally-specific artifact in a custom tab bar icon.
- For a tab bar icon, you can also provide a set of two fixed images, one for the unselected appearance and one for the selected appearance. If you provide fixed images, iOS does not treat them as template images and so it does not apply any standard or custom tab bar tint to them. However, iOS still draws the selection indicator image when the user taps a tab that contains a fixed-image icon.
- If necessary, include custom pressed or selected appearances for your icons . iOS automatically provides these appearances for items in navigation bars, toolbars, and tab bars, even when you specify a tint for the bar. ECUE recommends following best practices as defined by Apple.
- Give all icons in a bar a similar visual weight . Aim to balance the overall size, level of detail, and use of solid regions across all icons that can appear in a specific bar. In general, it does not look good to combine in the same bar icons that are large and blocky, and completely filled, with icons that are small, detailed, and unfilled.