Stop a slideshow

- Tap the screen.

Set slideshow settings

1. From the Home screen choose Settings > Photos.
2. To set:
   - *The length of time each slide is shown,* tap Play Each Slide For and choose a time.
   - *Transition effects when moving from photo to photo,* tap Transition and choose a transition type.
   - *Whether slideshows repeat*, turn Repeat on or off.
   - *Whether photos are shown in random order,* turn Shuffle on or off.

Play music during a slideshow

- From the Home screen choose iPod, and play a song. Then choose Photos from the Home screen and start a slideshow.

Using a Photo as Wallpaper

You see a wallpaper background picture as you unlock iPhone or when you're on a call with someone you don't have a high-resolution photo for.

Set a photo as wallpaper

1. Choose any photo and tap , then tap Use As Wallpaper.
2. Drag the photo to pan, or pinch the photo to zoom in or out, until it looks the way you want.
3. Tap Set Wallpaper.

You can also choose from several wallpaper pictures included with iPhone by choosing Settings > Wallpaper > Wallpaper from the Home screen.

Emailing a Photo

Email a photo

- Choose any photo and tap , then tap Email Photo.

iPhone must be set up for email (see "Setting Up Email Accounts" on page 46).

Sending a Photo to a Web Gallery

If you have a .Mac account, you can send photos directly from iPhone to a Web Gallery created with iPhoto '08. You can also send photos to someone else's .Mac Web Gallery if that person has enabled email contributions.

To send photos to a Web Gallery, you need to do the following:

- Set up your .Mac mail account on iPhone
- Publish an iPhoto '08 album to a .Mac Web Gallery
- Select "Allow photo uploading by email" in the Publish Settings pane of iPhoto '08

Tap Map to return to map view.

See the location of someone's address in your contacts list

- Tap in the search field, then tap Contacts and choose a contact.

To locate an address in this way, the contact must include at least one address. If the contact has more than one address, you must choose the one you want to locate. You can also find the location of an address by tapping the address directly in Contacts.

Bookmark a location

- Find a location, tap the pin that points to it, tap next to the name or description, then tap "Add to Bookmarks."

See a bookmarked location or recently viewed location

- Tap in the search field, then tap Bookmarks or Recents.

Add a location to your contacts list

- Find a location, tap the pin that points to it, tap next to the name or description, then tap Create New Contact or "Add to Existing Contact."

Getting Directions

Get directions

1. Tap Directions.
2. Enter starting and ending locations in the Start and End fields. By default, iPhone starts with your current approximate location (when available). Tap in either field and choose a location in Bookmarks (including your current approximate location and the dropped pin, when available), Recents, or Contacts.

   For example, if a friend's address is in your contacts list, you can tap Contacts and tap your friend's name instead of having to type the address.

   To reverse the directions, tap ⇅.
3. Tap Route, then do one of the following:
   - *To view directions one step at a time,* tap Start, then tap ➡ to see the next leg of the trip. Tap ⬅ to go back.
   - *To view all the directions in a list,* tap , then tap List. Tap any item in the list to see a map showing that leg of the trip.

The approximate driving time appears at the top of the screen. If traffic data is available, the driving time is adjusted accordingly.

You can also get directions by finding a location on the map, tapping the pin that points to it, tapping next to the name, then tapping Directions To Here or Directions From Here.

Conforming readers shall ignore all flags other than those at bit positions 3, 4, 5, 6, 9, 10, 11, and 12.

NOTE PDF integer objects can be interpreted as binary values in a signed twos-complement form. Since all the reserved high-order flag bits in the encryption dictionary's **P** value are required to be 1, the integer value ***P*** shall be specified as a negative integer. For example, assuming revision 2 of the security handler, the value -44 permits printing and copying but disallows modifying the contents and annotations.

Table 22 – User access permissions

Bit position	Meaning
3	*(Security handlers of revision 2)* Print the document. *(Security handlers of revision 3 or greater)* Print the document (possibly not at the highest quality level, depending on whether bit 12 is also set).
4	Modify the contents of the document by operations other than those controlled by bits 6, 9, and 11.
5	*(Security handlers of revision 2)* Copy or otherwise extract text and graphics from the document, including extracting text and graphics (in support of accessibility to users with disabilities or for other purposes). *(Security handlers of revision 3 or greater)* Copy or otherwise extract text and graphics from the document by operations other than that controlled by bit 10.
6	Add or modify text annotations, fill in interactive form fields, and, if bit 4 is also set, create or modify interactive form fields (including signature fields).
9	*(Security handlers of revision 3 or greater)* Fill in existing interactive form fields (including signature fields), even if bit 6 is clear.
10	*(Security handlers of revision 3 or greater)* Extract text and graphics (in support of accessibility to users with disabilities or for other purposes).
11	*(Security handlers of revision 3 or greater)* Assemble the document (insert, rotate, or delete pages and create bookmarks or thumbnail images), even if bit 4 is clear.
12	*(Security handlers of revision 3 or greater)* Print the document to a representation from which a faithful digital copy of the PDF content could be generated. When this bit is clear (and bit 3 is set), printing is limited to a low-level representation of the appearance, possibly of degraded quality.

7.6.3.3 Encryption Key Algorithm

As noted earlier, one function of a security handler is to generate an encryption key for use in encrypting and decrypting the contents of a document. Given a password string, the standard security handler computes an encryption key as shown in "Algorithm 2: Computing an encryption key".

Algorithm 2: Computing an encryption key

a) Pad or truncate the password string to exactly 32 bytes. If the password string is more than 32 bytes long, use only its first 32 bytes; if it is less than 32 bytes long, pad it by appending the required number of additional bytes from the beginning of the following padding string:

Table 112 – Entries in a Type 3 font dictionary (continued)

Key	Type	Value
FontBBox	rectangle	*(Required)* A rectangle (see 7.9.5, "Rectangles") expressed in the glyph coordinate system, specifying the *font bounding box*. This is the smallest rectangle enclosing the shape that would result if all of the glyphs of the font were placed with their origins coincident and then filled. If all four elements of the rectangle are zero, a conforming reader shall make no assumptions about glyph sizes based on the font bounding box. If any element is nonzero, the font bounding box shall be accurate. If any glyph's marks fall outside this bounding box, incorrect behavior may result.
FontMatrix	array	*(Required)* An array of six numbers specifying the *font matrix*, mapping glyph space to text space (see 9.2.4, "Glyph Positioning and Metrics"). NOTE A common practice is to define glyphs in terms of a 1000-unit glyph coordinate system, in which case the font matrix is [0.001 0 0 0.001 0 0].
CharProcs	dictionary	*(Required)* A dictionary in which each key shall be a glyph name and the value associated with that key shall be a content stream that constructs and paints the glyph for that character. The stream shall include as its first operator either **d0** or **d1**, followed by operators describing one or more graphics objects, which may include path, text, or image objects. See below for more details about Type 3 glyph descriptions.
Encoding	name or dictionary	*(Required)* An encoding dictionary whose **Differences** array shall specify the complete character encoding for this font (see 9.6.6, "Character Encoding").
FirstChar	integer	*(Required)* The first character code defined in the font's **Widths** array.
LastChar	integer	*(Required)* The last character code defined in the font's **Widths** array.
Widths	array	*(Required; should be an indirect reference)* An array of (**LastChar** – **FirstChar** + 1) widths, each element being the glyph width for the character code that equals **FirstChar** plus the array index. For character codes outside the range **FirstChar** to **LastChar**, the width shall be 0. These widths shall be interpreted in glyph space as specified by **FontMatrix** (unlike the widths of a Type 1 font, which are in thousandths of a unit of text space). If ***FontMatrix*** specifies a rotation, only the *horizontal* component of the transformed width shall be used. That is, the resulting displacement shall be horizontal in text space, as is the case for all simple fonts.
FontDescriptor	dictionary	*(Required in Tagged PDF documents; shall be an indirect reference)* A font descriptor describing the font's default metrics other than its glyph widths (see 9.8, "Font Descriptors").
Resources	dictionary	*(Optional but should be used; PDF 1.2)* A list of the named resources, such as fonts and images, required by the glyph descriptions in this font (see 7.8.3, "Resource Dictionaries"). If any glyph descriptions refer to named resources but this dictionary is absent, the names shall be looked up in the resource dictionary of the page on which the font is used.
ToUnicode	stream	*(Optional; PDF 1.2)* A stream containing a CMap file that maps character codes to Unicode values (see 9.10, "Extraction of Text Content").

For each character code shown by a text-showing operator that uses a Type 3 font, the conforming reader shall: