

Release 1.1 Copyright ©1994 by Don Yacktman. All Rights Reserved.  
NeXTMail and Mail.app are trademarks of NeXT Computer, Inc. All Rights Reserved.

# MiscMailApp

**Inherits From:** Object

**Declared In:** <misckit/MiscMailApp.h>

## Class Description

The MiscMailApp can be thought of as a representation, or proxy, to NeXT's Mail.app. You request a MiscMailApp for a particular machine—there can be only one per machine—via the class methods **+localMailer** (for the local machine) and **+mailerForHostNamed:**. You can also use the shortcut **MISC\_MAILAPP** which is equivalent to the id of the local MiscMailApp object. Warning: do not use the **+alloc** and **-init** methods of Object—the only valid way of obtaining a MiscMailApp instance are the above two methods.

Once you have the id of a MiscMailApp instance, you can request it perform certain functions, such as opening a new compose window (via **±openWindow**) or setting the fields of the current Compose window. (**±setTo:**, **±setSubject:**, **±setCc:**, and **±setBody:**) You can deliver the current compose window with **±deliver**. The quickest way to set up messages, however, is to use the methods **-sendMailTo:subject:body:** and **-sendMailTo:subject:body:cc: andDeliver:**, both of which open a new Compose window and fill in the relevant

fields.

If you continually open Compose windows without delivering them they can eventually clutter the screen. To avoid this, you can set a *no clutter* mode in which the current Compose window is delivered before a new one is created. See `±setNoClutter:` for more information. If you need to know the host to which a MiscMailApp is connected, use the `±hostname` method.

Although the above discussion uses NeXT's Mail.app, the MiscMailApp does not have to use the actual Mail.app program provided with NEXTSTEP. Any replacement application which sets up the `MailSendDemo` Mach port is actually suitable. The MiscMailApp object will attempt to launch a mailer if this port is not set up. If a user wishes to use a different mailer, they should set the GLOBAL default `Mailer` to the name of the replacement mailer. For example, to use Mynah.app as the replacement mailer, do this:

```
dwrite GLOBAL Mailer Mynah.app
```

If the desired mailer cannot be launched (or this default is not set) then Mail.app will be launched. Note that *no* attempt to launch a mailer will occur if an application has already registered the `MailSendDemo` Mach port; the running application will be used, regardless of what the default is set to.

If the mailer is quit after the MiscMailApp has connected to it, MiscMailApp will attempt to restart it the next time it tries to send a message to the mailer. If it cannot reconnect for some reason, it will invalidate and free itself. For this reason, you should always use the `+mailerForHostNamed:` method to grab the current (and correct) instance of MiscMailApp. Don't cache the `ids` of MiscMailApp instances, since they could free themselves without warning!

## Instance Variables

```
MiscString *hostName;  
port_t mailPort;  
id theSpeaker;
```

**int currentWindow;**  
**BOOL noclutter;**

hostName	A MiscString containing the name of the host with which this MiscMailApp instance communicates.
theSpeaker	An instance of MiscMailSpeaker which is used for the communications with NeXT's Mail.app.
currentWindow	Zero if no current Compose window is available; an id of a Mail.app Compose window as returned by MiscMailSpeaker otherwise.
noclutter	Flag to determine whether or not it is OK to leave undelivered Compose windows hanging around. Default is YES.

## Method Types

Obtaining a MiscMailApp instance	+ localMailer + mailerForHostNamed:
Special features	- hostname - noclutter - setNoclutter:
Creating a message quickly	- sendMailTo:subject:body: - sendMailTo:subject:body:cc:andDeliver:
Manipulating mail messages	- deliver - deliver:

- openWindow
- openWindow:
- setBody:
- setCc:
- setSubject:
- setTo:

## Class Methods

### **localMailer**

#### **+ localMailer**

Returns a MiscMailApp instance which represents NeXT's Mail.app running on the local machine, creating and initializing it if necessary. An attempt will be made to launch Mail.app if it is not already running. Returns *nil* if no connection to Mail.app was possible.

**See also:** **+mailerForHostNamed:**

### **mailerForHostNamed:**

#### **+ mailerForHostNamed:*aHost***

Returns a MiscMailApp instance which represents a mailer application running on *aHost*, creating and initializing it if necessary. An attempt will be made to launch the application named by the GLOBAL default <sup>a</sup>Mailer°. If the default is not set or the launch attempt fails, then NeXT's Mail.app will be launched. If an application which registers the <sup>a</sup>MailSendDemo° Mach port is already running, it will be used regardless of the setting of the default. The object *aHost* should respond to the **±stringValue** method. If *aHost* is *nil* or returns a string of zero length, then the local host is assumed. Returns *nil* if no connection to Mail.app was possible.

**See also:** +localMailer

## Instance Methods

### **deliver**

- **deliver**

Delivers the current Compose window in Mail.app. Returns **self**.

**See also:** ± **deliver:**

### **deliver:**

- **deliver:sender**

Delivers the current Compose window in Mail.app. Returns **self**. This method is the same as ±**deliver** and is provided as a convenience to support target/action.

**See also:** ± **deliver**

### **hostname**

- **hostname**

Returns a MiscString containing the name of the host which is running the copy of NeXT's Mail.app that will be sent messages by this object.

**See also:** + **mailerForHostNamed:**

## **noClutter**

- (BOOL)**noClutter**

Returns the current *noClutter* setting. See the discussion of **±setNoClutter:** for details about *noClutter* mode. Returns **self**.

**See also:** **± openWindow**, **± openWindow:**, and **± setNoClutter:**

## **openWindow**

- **openWindow**

Opens a new Compose window in Mail.app. Returns **self** if all is well. If the connection to Mail.app has been lost and cannot be reestablished, then **nil** is returned. If **nil** has been returned, the MiscMailApp object has been invalidated and freed, so no further messages should be sent to it. The **+mailerForHostNamed:** method can be used to attempt to reconnect to the mailer if desired.

**See also:** **± noClutter**, **± openWindow:**, and **± setNoClutter:**

## **openWindow:**

- **openWindow:sender**

Opens a new Compose window in Mail.app. Returns **self**. This method is the same as **±openWindow** and is provided as a convenience to support target/action.

**See also:** **± noClutter**, **± openWindow**, and **± setNoClutter:**

## **sendMailTo:subject:body:**

- **sendMailTo:to**

**subject:***subject*  
**body:***body*

Makes use of the `±sendMailTo:subject:body:cc:andDeliver:` method (with `cc:nil` and `andDeliver:NO`) to create a new compose window addressed to *to* with subject *subject* and body *body*. If any of *to*, *subject*, or *body* is nil or of zero length, then those fields in Mail.app will not be set. Returns **self** upon successful completion or **nil** if unsuccessful.

**See also:** `± sendMailTo:subject:body:cc:andDeliver:`

**sendMailTo:subject:body:cc:andDeliver:**

- `sendMailTo:`*to*  
**subject:***subject*  
**body:***body*  
**cc:***cc*  
**andDeliver:**(BOOL)*flag*

This method encompasses a complete composition and delivery of a mail message. First, a new Compose window is opened up by Mail.app. Then it is addressed to *to* with subject *subject* and body *body*. Carbon copies will be sent to *cc* and then the message will be delivered if *flag* is true. If *flag* is false, the window is left open in Mail.app for the user to modify. The window will remain open until the user closes it, or, if *noclutter* is set true, until a new Compose window is opened. If any of *to*, *subject*, *body*, or *cc* is nil or of zero length, then those fields in Mail.app will not be set. Returns **self** upon successful completion or **nil** if unsuccessful.

**See also:** `± sendMailTo:subject:body:`

**setBody:**

- `setBody:`*sender*

Sets the Body: field of the current Compose window to the string returned by [**sender stringValue**]. Nothing is done if the string value is of zero length or *sender* is nil. Returns **self**.

#### **setCc:**

- **setCc:sender**

Sets the Cc: field of the current Compose window to the string returned by [**sender stringValue**]. Nothing is done if the string value is of zero length or *sender* is nil. Returns **self**.

#### **setNoclutter:**

- **setNoclutter:(BOOL)flag**

Turns *noclutter* mode on or off. When on, every time a new Compose window is requested with **±openWindow** or **±openWindow:**, the current window, if any, is first delivered before creation of the new Compose window. Returns **self**.

**See also:** **± openWindow**, **± openWindow:**, and **± noclutter**

#### **setSubject:**

- **setSubject:sender**

Sets the Subject: field of the current Compose window to the string returned by [**sender stringValue**]. Nothing is done if the string value is of zero length or *sender* is nil. Returns **self**.

#### **setTo:**

- **setTo:sender**

Sets the To: field of the current Compose window to the string returned by [**sender stringValue**]. Nothing is done if the string value is of zero length or *sender* is nil. Returns **self**.