

DiaDraw E-mail ANE

Manual



© 2014 DiaDraw

Contents

I. What is DiaDraw e-mail ANE?	5
How do you use DiaDrawScreenCast.ane?	5
II. Example Code	6
Instantiate the e-mail wrapper class	6
Set up and pop up the e-mail composer:	6
Get notified when the e-mail composer closes:	7
III. API	8
NativeMailWrapper class	8
MailExtensionEvent class	9
IV. Questions?	10
V. What else is out there?	10
VI. Thank you	10

What is DiaDraw e-mail ANE?

What does it do?

DiaDrawEmail.ane is an AIR Native Extension for iOS which allows you to send e-mails with attachment from Adobe AIR.

How do you use DiaDrawScreenCast.ane?

Include **DiaDrawEmail.ane** in your Flash or Flex project.

Make an instance of the [NativeMailWrapper](#) (`com.diadraw.extensions.mail.NativeMailWrapper`) and call [isMailComposerAvailable\(\)](#) first to check whether an e-mail client has been configured on the device you are running.

If [isMailComposerAvailable\(\)](#) returns **true**, then prepare your attachment by obtaining the path to and the name of the file(s) you want to attach and call [sendMail\(\)](#). This will pop up the device's default mail composer.

Listen to [MailExtensionEvent.MAIL_COMPOSER_EVENT](#) to get notified when the user either finishes or cancels sending the e-mail.

Credit where credit is due

The first version of this ANE was based on [Piotr Koscierynski's brilliant tutorial In-app mail composer](#), which he published in November 2011.

So did the world really need yet another e-mail ANE?

In early 2012, when we first needed an e-mail ANE, Adobe and Apple's SDKs didn't quite see eye to eye when it came to orientation. The ANE would fire up the e-mail client and your app would randomly rotate when you closed it. Slightly unexpected and kind of fun, but not a plus if you wanted your app approved by Apple.

We implemented a fix for that to scratch our own itch, as we were just about to release our own app that used the e-mail ANE, [IdeaDraw](#) and badly wanted it approved and out in the App Store. So [we published our tamed version to help other developers](#) in the same situation and have been maintaining it since. And maintenance it takes... [each new iOS release seems to change the rules for view orientation](#) or ~~messes something else up~~, I mean [changes other rules](#), which is where we come in with tweaks and fixes.

To put it dramatically: we keep the light on above the trail that Piotr blazed for the rest of us... Cheers, man!

Example Code

Instantiate the e-mail wrapper class

```
import com.diadraw.extensions.mail.MailExtensionEvent;
import com.diadraw.extensions.mail.NativeMailWrapper;

private function ensureExtension():void
{
    m_mailExtension = new NativeMailWrapper();

    m_mailExtension.removeEventListener(
        MailExtensionEvent.MAIL_COMPOSER_EVENT, handleMailComposerEvent );
    m_mailExtension.addEventListener(
        MailExtensionEvent.MAIL_COMPOSER_EVENT, handleMailComposerEvent );
}
```

Set up and pop up the e-mail composer:

```
private function sendEmail(
    _attachmentPath : String,
    _fileName : String,
    _mimeType : String ) : void
{
    var subject : String = "Hey, the EasyNativeExtensions ANE shop is now open!";

    // You can use HTML in the body:
    var body : String =
        "Check it out:
        <a href=http://easynativeextensions.com/shop/>Download ANEs.
        Some free!</a>";

    // _mimeType is the internet media type of the file you are sending,
    // for example "image/png"
    var attachmentStr : String =
        _attachmentPath + "|" + _mimeType + "|" + _fileName;

    var recipients : String = "office@diadraw.com";
    var ccList : String = "support@diadraw.com";
    var bccList : String = "rada@diadraw.com"

    m_mailExtension.sendMail(
        subject,
        body,
        recipients,
        ccList,
        bccList,
        [attachmentStr],
        true );
}
```

Get notified when the e-mail composer closes:

```
private function handleMailComposerEvent( _event : MailExtensionEvent ) : void
{
    switch ( _event.composeResult )
    {
        case MailExtensionEvent.MAIL_RESULT_SENT:
            { /*The e-mail was sent*/ }
            break;

        case MailExtensionEvent.MAIL_RESULT_CANCELED:
            { /*E-mail sending was cancelled*/ }
            break;

        case MailExtensionEvent.MAIL_RESULT_SAVED:
            { /*An e-mail draft was saved for later*/ }
            break;

        case MailExtensionEvent.MAIL_RESULT_FAILED:
            { /*E-mail sending failed*/ }
            break;

        default: break;
    }
}
```

API

NativeMailWrapper class

Package `com.diadraw.extensions.mail`

`NativeMailWrapper()` constructor

```
public function NativeMailWrapper( _target : IEventDispatcher = null )
```

Creates a new instance of `NativeMailWrapper`.

`isMailComposerAvailable()` method

```
public function isMailComposerAvailable() : Boolean
```

Checks if the device's default e-mail client has been set up.

`sendMail()` method

```
public function sendMail( subject          : String = "",
                          messageBody     : String = "",
                          toRecipients    : String = "",
                          ccRecipients    : String = "",
                          bccRecipients    : String = "",
                          attachmentsData : Array = null,
                          useHtml         : Boolean = true ): void
```

Shows the native e-mail composer and populates its fields with the subject, recipients, body and attachments given.

Parameters

subject - optional; sets the e-mail subject

messageBody - optional; sets the e-mail main content. You can use HTML tags, as long as you set **useHTML** to **true**.

toRecipients - optional; comma-separated list of recipient e-mails

ccRecipients - optional; comma-separated list of Cc recipient e-mails

bccRecipients - optional; comma-separated list of Bcc recipient e-mails

attachmentData - optional, an array of strings, each of which describes one attachment, using the following format:

```
// _mimeType is the internet media type of the file you are sending,
// for example "image/png"
var attachmentStr : String =
    _attachmentPath + "|" + _mimeType + "|" + _fileName;
```

useHtml - optional; a flag that tells the mail composer to treat the string in **messageBody** as HTML, instead of plain text.

See also

[MailExtensionEvent.MAIL_COMPOSER_EVENT](#)

MailExtensionEvent class

package com.diadraw.extensions.mail

MailExtensionEvent.MAIL_COMPOSER_EVENT

composeResult type

Emitted when the user dismisses the mail composer or an error occurs.

composeResult property

composeResult : String

Contains information about the event. It can be one of the following:

MailExtensionEvent.MAIL_RESULT_SENT

The e-mail was sent successfully.

MailExtensionEvent.MAIL_RESULT_SAVED

An e-mail draft was saved.

MailExtensionEvent.MAIL_RESULT_CANCELLED

E-mail sending was cancelled.

MailExtensionEvent.MAIL_RESULT_FAILED

E-mail sending failed.

MailExtensionEvent.MAIL_COMPOSER_NOT_AVAILABLE

sendMail() was called, although isMailComposerAvailable() returned false.

Questions?

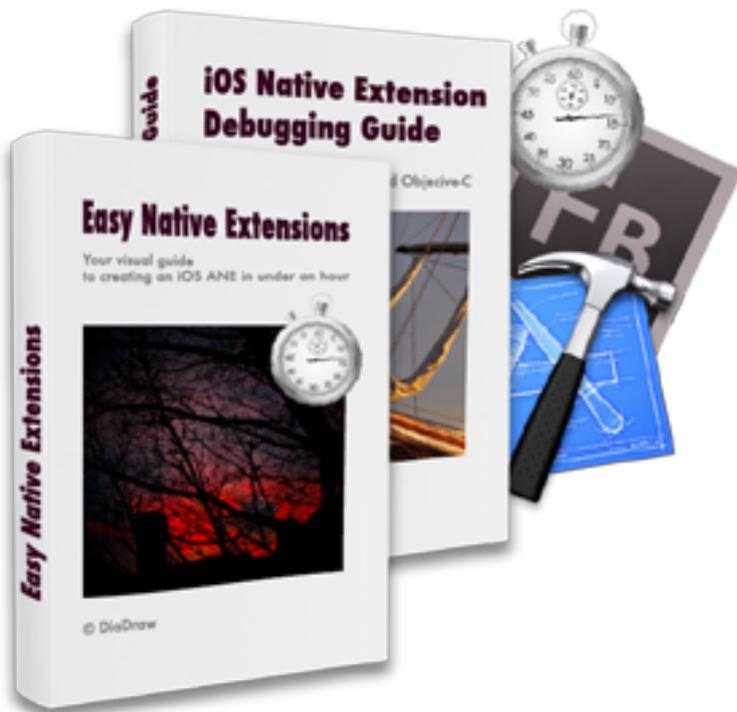
Drop us a line at support@diadraw.com if you have any questions about using **DiaDraw E-mail ANE** or this manual.

What else is out there?

Find more ANE-related information in our [tutorials](#) and [articles](#).

Have a look at [the other ANEs on our website](#).

Have you checked out our [Easy Native Extensions eBook](#) yet?



Thank you

... and happy coding from [The DiaDraw Team!](#)