

## Using FireMonkey to send an email with an attachment

I've decided to start posting solutions to technologies using Delphi and in particular FireMonkey with this blog. Today, I am going to share how to send emails with attachments with FireMonkey for iOS devices.

### Sending simple email (No attachment)

It is pretty straightforward to send an email without an attachment in FireMonkey. The syntax for this is as follows. In your uses section, you will need to add some ios units.

Uses

```
Macapi.ObjectiveC,  
Macapi.Helpers,  
FMX.Helpers.iOS,  
iOSapi.Foundation;
```

Then define the following procedure to email

```
procedure Email(Recipient, Subject, Content: string);  
var  
    lURL: string;  
    nsURL: NSString;  
begin  
    lURL := 'mailto:'+Recipient;  
    if (Subject<>'' ) or (Content<>'' ) then  
    begin  
        lURL := lURL+'?subject='+Subject;  
        if Content<>'' then  
            lURL := lURL+'&body='+ Content;  
        nsURL:= StrToNSStr(lURL).stringByAddingPercentEscapesUsingEncoding(NSASCIIStringEncoding);  
        lURL:= NSStrToStr(nsURL);  
  
        end;  
        SharedApplication.openURL(StrToNSURL(lURL));  
    end;
```

Now emailing is as simple as

```
Email('roywoll@gmail.com', 'subject', 'content');
```

### Sending Email (with Attachment)

Now unfortunately, this technique cannot email an attachment. But there is a way to do this in RAD Studio. Since its tricky coding, we have created a simple routine that you can call. We have provided all the code and you can download this at

[https://github.com/roywoll/wwEmailWithAttachment\\_fmx](https://github.com/roywoll/wwEmailWithAttachment_fmx)

This project contains a unit named wwEmailWithAttachment.pas which defines the method

```
procedure wwEmail(  
    Recipients: Array of String;  
    ccRecipients: Array of String;  
    bccRecipients: Array of String;  
    Subject, Content,
```

```
AttachmentPath: string;  
mimeTypeStr: string = ''); overload;
```

To use this method, add the unit `wwEmailWithAttachment` to your form's uses clause, and then call the procedure as in the following:

```
wwEmail(['roywoll@gmail.com', 'royswoll@yahoo.com'],  
        [], [], 'Subject', 'Content', fileName);
```

There is one more step required. The unit `wwEmailWithAttachment` uses the part of the iOS SDK that FireMonkey has not put into the default library. So when you compile and link your code, you will receive a link error. Fortunately, it's straightforward to add this part of the SDK to your environment. Once you have this done, then you won't have to do it again. The steps are detailed below.

You will need to add the MessageUI framework to your iOS SDK in the RAD Studio IDE.

1. Select from the IDE - Tools | Options | SDK Manager
2. Then for your 64 bit platform (and 32 bit if you like) do the following
  - a) Scroll to the bottom of your Frameworks list and select the last one
  - b) Click the add button on the right to add a new library reference and

then enter the following data for your entry

**Path on remote machine:** \$(SDKROOT)/System/Library/Frameworks  
**File Mask:** MessageUI  
**Path Type:** Leave unselected

3. Click the button *Update Local File Cache* to update your SDK
4. Click the OK Button to close the dialog

Once you have done this, you can compile the project and it should yield no linker errors.

## Android

If you wish to do this same thing with Android, it's much simpler. The provided unit also contains the same method for Android, so you can email with the same exact code in your project. In addition, you don't need to update your SDK, as it will work without any changes to your environment.