

POP3it

Version 2.0

The Email Receiving Plug-in for Filemaker 4.x/5.x

Developed by Comm-Unity Networking Systems

Receiving email with
FileMaker has never been
so easy and flexible!



POP3it Support Email:
pop3it@comm-unity.net

POP3it Support Website:
<http://www.pop3it.com/>

Comm-Unity Networking Systems

Table of Contents

Introduction	3
Features and Uses	3
Installation and Configuration	4
A Typical Email Session	9
Issues to Think About When Using POP3it	11
Example POP3it Script	12
Extract Information Example	17
POP3it External Function Reference	20
Understanding Error Responses	26
Credits	26
Contact Information	26

POP3it

Introduction

POP3it is an exciting new plug-in from Comm-Unity Networking Systems that allows you to receive email directly into your FileMaker Pro databases. With POP3it, you can check multiple POP3 email accounts, which brings incredible new power to FileMaker Pro.

The following documentation will help you familiarize yourself with the POP3it functions, the example database included with the POP3it archive, and how to build your own POP3 email solutions. This plug-in is different than most FileMaker Pro plug-ins in the sense that it brings information into FileMaker Pro, whereas most plug-ins export information. Because of this difference, scripting POP3it is inherently different than other plug-ins.

Before we get to far into the documentation, let's take a look at the features of POP3it and some possible applications.

Features and Uses

POP3it has many features, including the following:

- Dynamically check multiple POP3 email accounts, even on different email servers.
- Automatically extract major email fields such as From, Subject, and Body.
- Receive attachments, which are placed in a directory that you can specify dynamically.
- Download entire messages into one field or split up into their respective fields.

POP3it has many potential uses. The following are a few ideas you could use POP3it in:

- Archive a mailing list in FileMaker Pro and use the Web Companion to build a web interface to that mailing list.
- Use SMTPit and POP3it together to create a complete email client using only FileMaker Pro.
- Create new records based on an email message.
- Synchronize two FileMaker Pro databases via email.

Installation and Configuration

To install the plug-in, first make sure FileMaker Pro is closed. Next, unzip the POP3it zip file on Windows, or unstuff the POP3it stuffit file on Macintosh, and then double-click the POP3it_Installer application. This will automatically place the POP3it plug-in file into the "System" folder on Windows, or the "FileMaker Extensions" folder on Macintosh, inside your FileMaker Pro 4.x or 5.x folder. If you have a previous version of POP3it, the installer will overwrite it. (Windows Users: The Installer will not run from inside an Unzipping program like WinZip. You must unzip it to a folder before running the installer.)

After you install the plug-in as described above, open FileMaker and set the default preferences. To do this, go to Edit->Preferences->Application, click on the Plug-ins tab, and double-click the POP3it plug-in.

Basics

Once the Configuration Dialog is open, click the "Basics" tab (See Figure 1) to set the following default values. **Host** is where you enter the domain name or IP address of your POP3 server. **User** is where you can enter a default User name. (The User name usually is the part of your email address before the @ symbol.) **Password** is where you can enter a default Password.

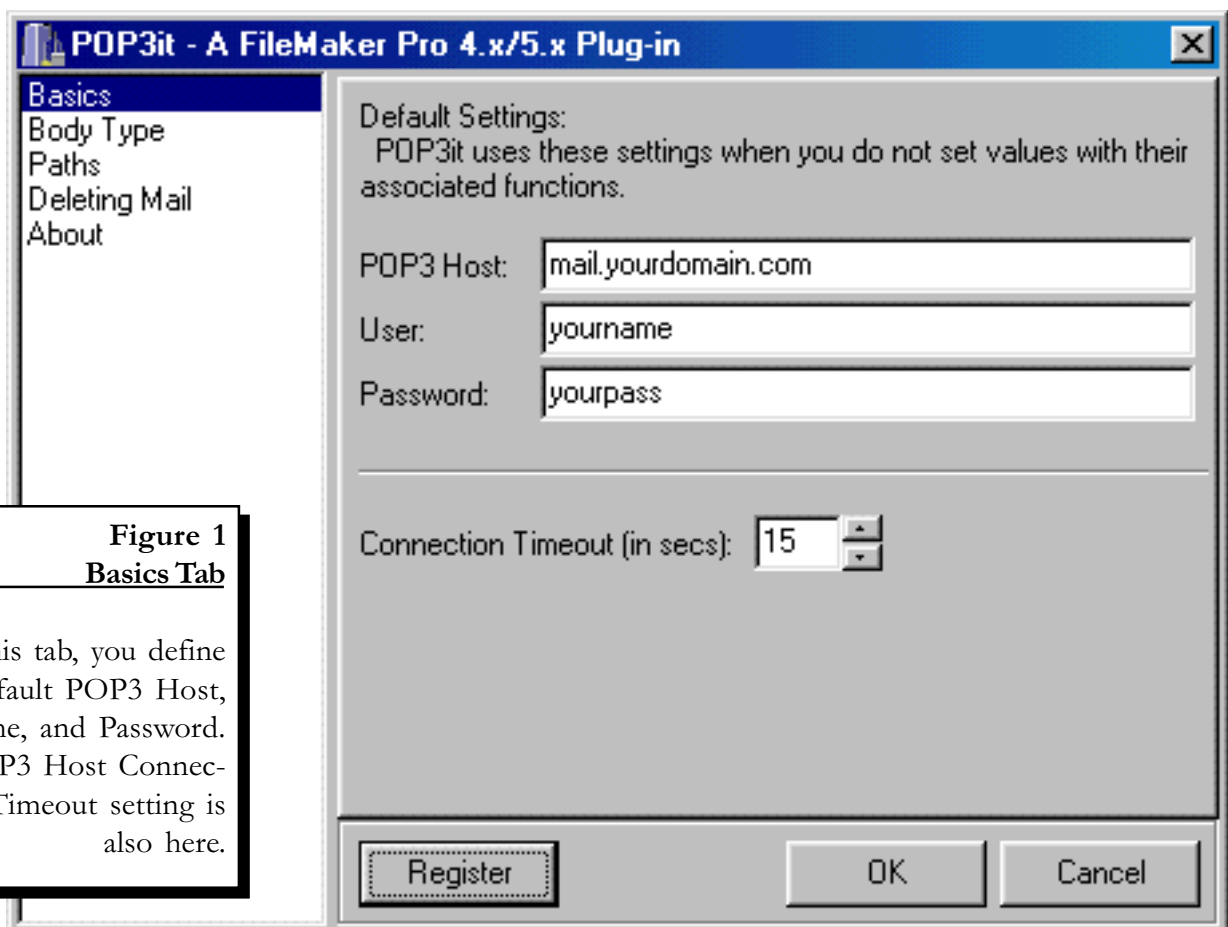


Figure 1
Basics Tab

On this tab, you define your Default POP3 Host, Username, and Password. The POP3 Host Connection Timeout setting is also here.

POP3it will refer to the default settings when you do not specifically set values in your scripts. In other words, if you do not define a User name in your POP3it related script, POP3it will use the Default User name.

POP3it waits for certain amount of time for your POP3 host to respond. By default, this is set to 15 seconds. However, you can adjust the setting to fit your needs by entering a different value in the **Connection Timeout** field.

Body Type

Emails come in three main forms. There are Text only emails, HTML only emails, and "multi-part/alternative" emails which have both a Text and an HTML part. With version 2.0 of POP3it, you can now specify which type of email you want POP3it to return when you call the "CurrBody" or "CurrAll" functions. When you do not specify which type you want when you call those functions, POP3it uses this default setting. If you select "Text", POP3it will only return a plain text body to you. If you select "HTML", POP3it will only return an HTML body to you. If you select "Both", POP3it will return a plain text body (if there is one) followed by an HTML body (if there is one). If you select "Text, HTML", POP3it will return a plain text body to you, unless there is not one, then it will return an HTML body to you. If you select "HTML, Text", POP3it will return an HTML body to you, unless there is not one,

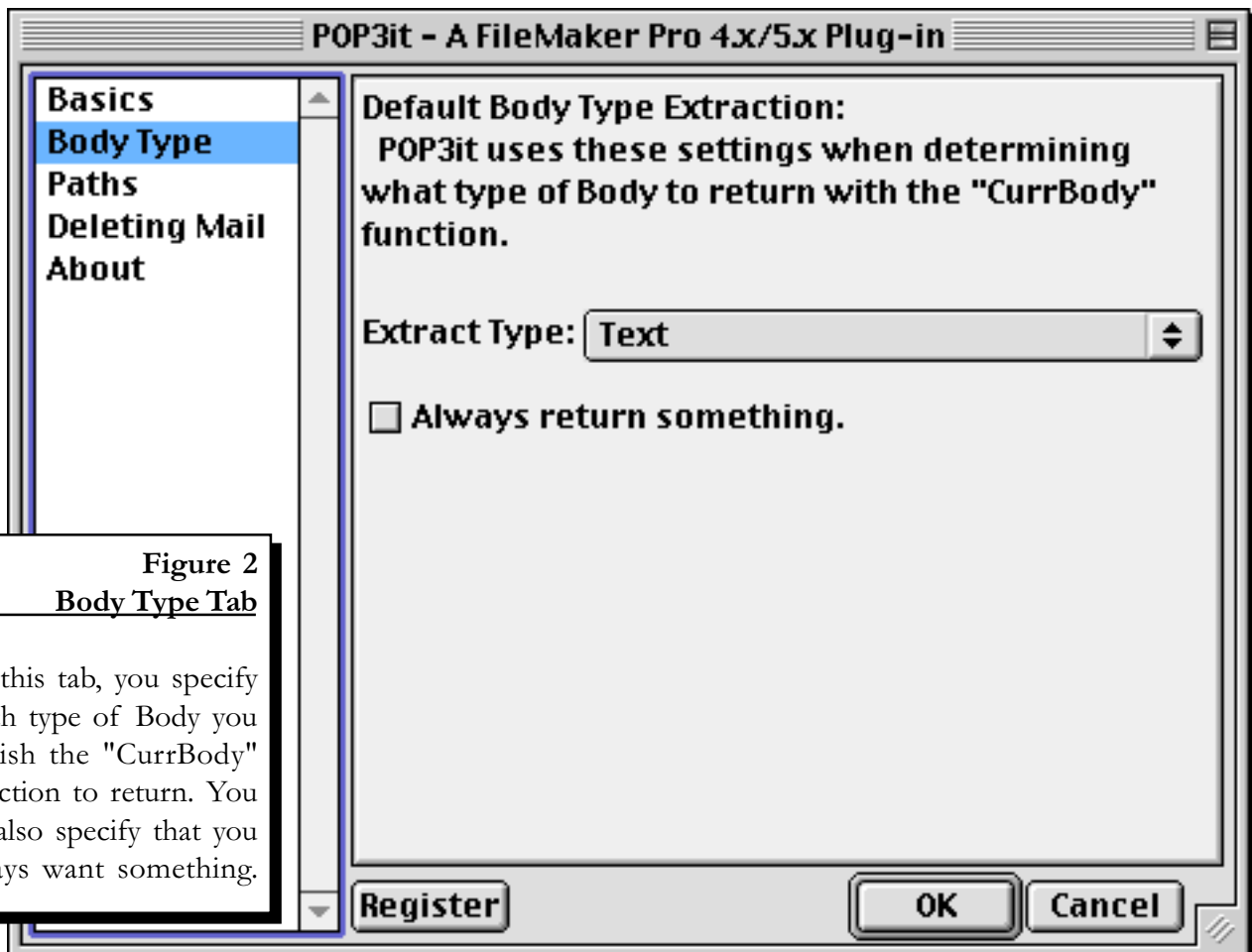


Figure 2
Body Type Tab

On this tab, you specify which type of Body you wish the "CurrBody" function to return. You can also specify that you always want something.

then it will return a plain text body to you.

It is possible for POP3it to return a blank body to you depending on the type of email and what type you ask for. For example, if you ask for "Text", and the email only has an HTML body, then POP3it will return a blank body to you. If you never want a blank body returned to you, check the **Always return something** checkbox. (Note: If there actually is nothing in the body of the email, Text or HTML, POP3it will still return a blank body because there was nothing to return.)

Paths

If you do not specify where to put Attachments in your scripts, POP3it looks at these settings. **Attachments** is where you enter the default directory for saving attachments. **HTML Export** is where you enter the default directory for saving exported HTML emails. (You can use the ... button to browse for the directory.)

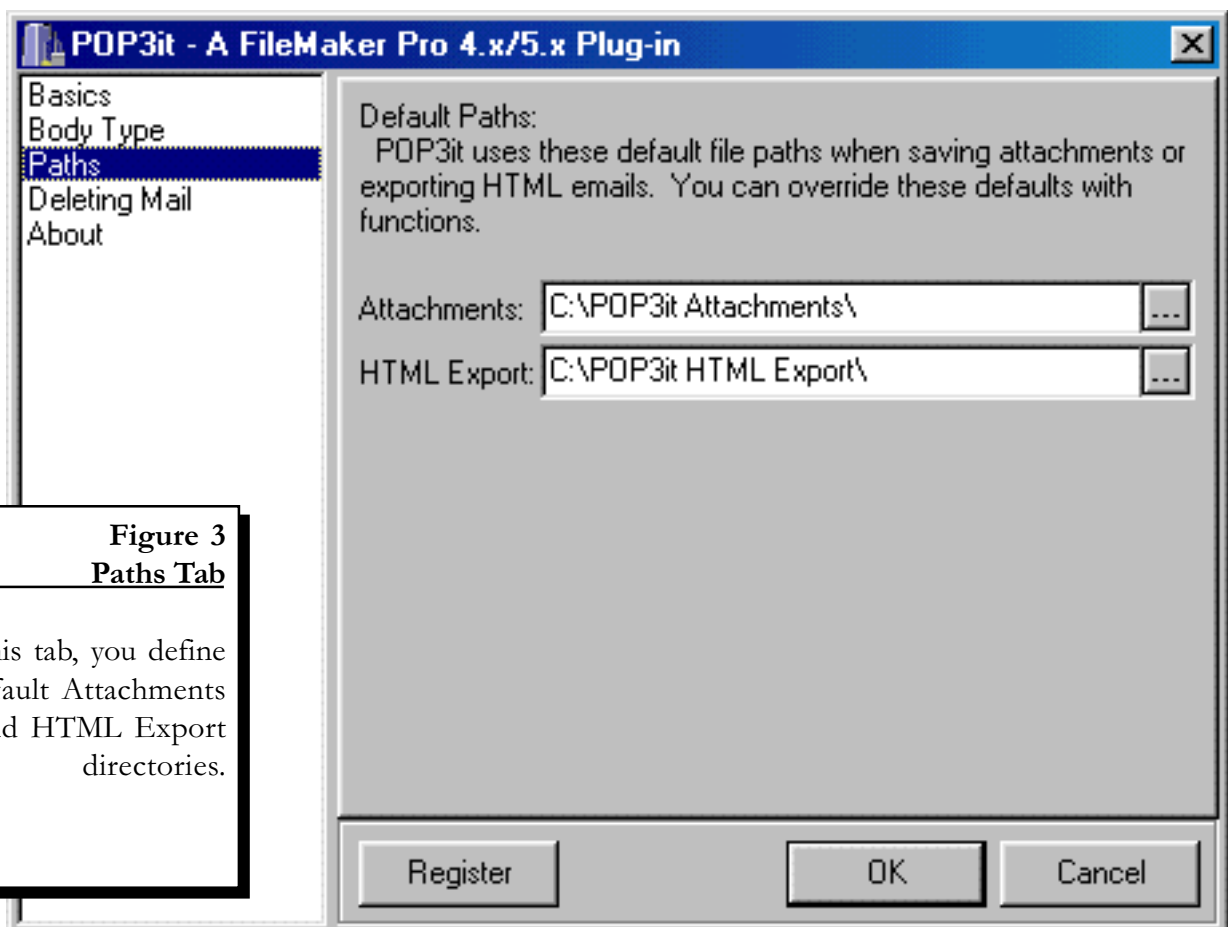


Figure 3
Paths Tab

On this tab, you define your Default Attachments and HTML Export directories.

Deleting Mail

If you want POP3it to automatically delete email for the server, check the **Automatically Delete Mail** checkbox. You also have two options on when to delete the email. Choosing **Before Acquire Next** is practically the same as calling POP3-CurrDelete before calling POP3-AcquireNext. Choosing **Before Close Connection** is practically the same as calling POP3-DeleteAll before calling POP3-CloseConnection.

About

The About Tab simply tells you what version you are currently running, and who this copy of POP3it is registered to. All software upgrades are free, so if you do not have the most recent version that is listed on our website, you can upgrade your copy simply by downloading the latest version and installing it.

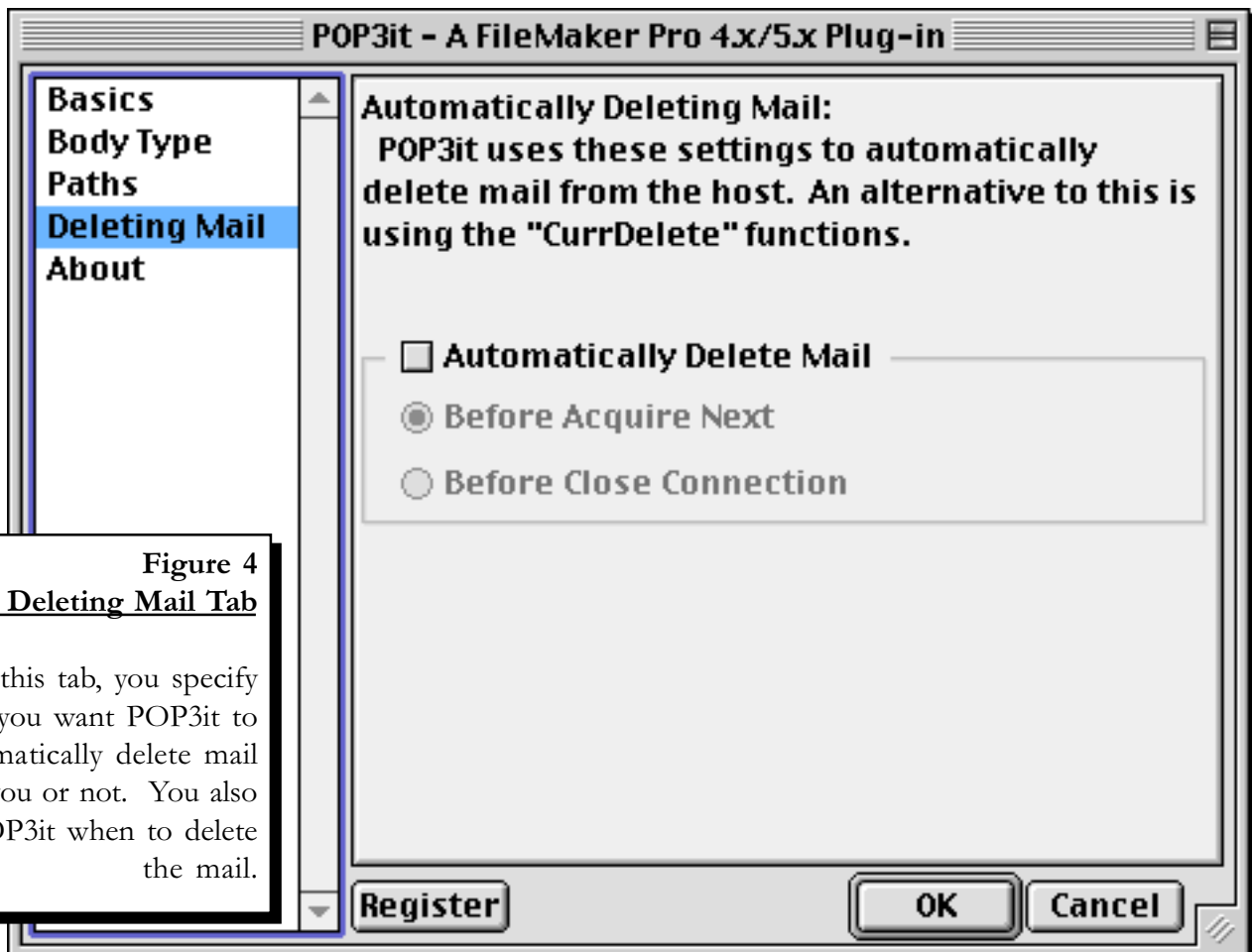


Figure 4
Deleting Mail Tab

On this tab, you specify that you want POP3it to automatically delete mail for you or not. You also tell POP3it when to delete the mail.

Registering POP3it

You can register POP3it from the Configuration Dialog once you have purchased it from our secure website or by ordering by phone. After you purchase POP3it, we will send you a serial number to register your copy by using the Register button in the Configuration Dialog. We have also decided to include a register function so that developers can easily register POP3it with their bound solutions once they have purchased a Developers license. For more information on purchasing POP3it and other exciting plug-ins, visit our website (<http://www.filemaker.comm-unity.net>) and choose the Purchase link from the tool bar on the left.

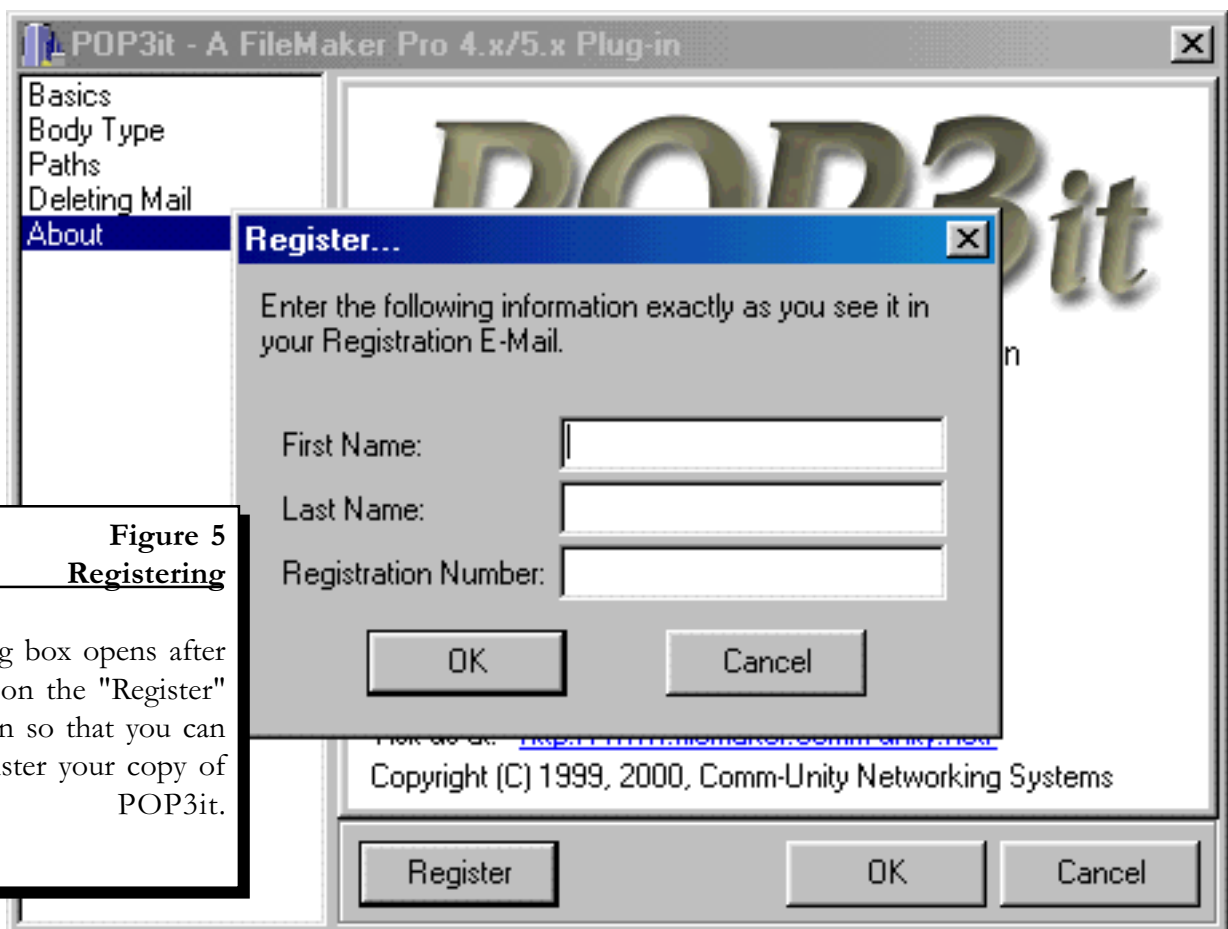


Figure 5
Registering

This dialog box opens after you click on the "Register" button so that you can register your copy of POP3it.

A Typical Email Session

Let us think for a moment about the real world equivalent of a real Post Office Box compared to the Internet world of a POP3 email box. To receive your mail from the Post Office in the real world you must first go to the Post Office. Once there you locate your mail box and use a key to open the box. At this point you take your mail, close your mail box and leave the Post Office. Every once in a while you may receive a notice that you have a package to pick up that will not fit in your box, so you may have to pick up your package before you leave the Post Office.

In the Internet world, you can compare this process almost step by step. To receive your email on the Internet, you must first open a connection with your POP3 mail server and use your username and password to open your email box. At this point, you grab all of your email including any larger attachments, and then close the connection to your POP3 server.

There are several subtle differences between checking your email and getting mail in the real world. One difference is the fact that your email will be in your box until you delete it. You will want to make sure and read the following section called "Issues to Think About When Using POP3it" to understand some of the problems that you can run into when checking email.

Now that we have a basic understanding of how the process works, let us take it a step further and explain how to check an email account with POP3it and FileMaker. Probably the best place to start is by walking through all of the steps it takes in order to get email from a POP3 mail server. For now, we will describe a simple path to receive email from your email server into FileMaker Pro, then we will give you an example script based on those steps.

Step 1: Set your username, password, and mail-host.

Use the "POP3-Host", "POP3-Username", and "POP3-Password" functions to set your username, password and mail-host. The POP3it External Function Reference at the end of this document fully describes these functions. When used properly, they can help you check multiple email accounts easily.

Step 2: Make a connection to your POP3 mail host.

To make a connection to your mail server you simply use the "POP3-OpenConnection" function with a parameter of "".

Step 3: See how many messages are waiting.

To see how many messages are waiting, we use the "POP3-AcquireFirst" function which returns how many messages are waiting and will also allow us to start working with the first message. If there are no messages the function will return a zero, which means we need to jump to Step 9 and close the POP3 connection. If there are one or more messages waiting we need to continue on with the script to get the messages.

Step 4: Enter A loop.

It is essential that we enter a loop here so that we will be able to get multiple email messages. Because "POP3-AcquireFirst" sets us up with the first email message, we are ready to start getting the information for the first message right when we enter the loop.

Step 5: Get current message information.

At this point, we can get the email message in two different ways. The first way is to use the "POP3-CurrAll" function to get the entire message and put it into one field. This function will also download any email attachments to your hard drive in the directory that you have specified either by function or on the configuration box. This way works, but it is not necessarily easy to work with. The second way is to set fields using the other "Current" functions such as "POP3-CurrMessageID", "POP3-CurrHead", "POP3-CurrDate", "POP3-CurrTo", "POP3-CurrFrom", "POP3-CurrSubject", "POP3-CurrBody", and "POP3-CurrAttach". This is somewhat easier because POP3it has extracted the major email fields already for you. All you have to do is set the appropriate fields using the above mentioned functions. This is also the best place to run sub scripts based on the email message. For example, you could check the subject of the message for a key word, and run a script based on your findings.

Step 6: Delete the current message.

Technically, you do not have to delete the current message at this time. However, it is much easier to always delete the messages so you do not have to check for duplicate message each

time you check your email account. When you are first testing your POP3it script, you may choose to skip deleting the messages until you have the script working the way you want it to, otherwise, you may have to keep sending yourself test emails.

Step 7: Check to see if there are more messages waiting. / Exit loop.

The next step is to see if there are any more messages waiting. To do this, use the "POP3-AcquireNext" function. Since "POP3-AcquireFirst" has already given us our first message, "POP3-AcquireNext" will give us the second message. Throughout the loop this function will continue to give us the next message incremented by one until there are no messages left. When there are no messages left this function will give us a value of 0 (zero), thus giving a test to break out of the loop. If "POP3-AcquireNext" does return a zero, we will need to break out of the loop and jump to Step 9 to close the POP3 connection.

Step 8: Retrieve the next message.

At this point we are at the end of the loop. If we have gotten this far, we have already retrieved one message, and there is at least one more message waiting since we made it past Step 7. So, the loop will start over with Step 5 and continue to loop until Step 7 proves that we have no more messages.

Step 9: Close the connection.

This last step is simple, but crucial. If you do not close the POP3 connection with your host you can cause problems with both POP3it and with your POP3 mail host. This is one of the pitfalls of using a POP3 host. By its nature, you have to manually open the connection and manually close it. For more information on how things can go awry and things to look out for, read the next section called "Issues to Think About When Using POP3it". With that said, to close the POP3 connection simply use the "POP3-CloseConnection" function with a parameter of "".

Issues to Think About When Using POP3it

There are several issues to think about when using POP3it to check an email account. The first major issue is to make sure that every time you open a connection with your POP3 server, that you also close that connection. This may seem quite obvious, however, if you decide to use the Paste Result script step to close your connection, and the field that you are pasting the result to is not on the current layout, FileMaker Pro will not activate the function and POP3it will not close the connection. For this reason, it is wise to either make sure that the field you are working with is on the current layout, or that you use the Set Field script step, because the Set Field script step does not require the result field be on the current layout.

You can run into another problem if you decide not to delete each message from the server when you check email. If you do not build in some kind of message-id checking, you will

receive the same email messages over and over again. To keep things simple we do not address this problem in this documentation. The problem is addressed in our Simple Mail Client solution that will be available on our website.

Another issue you may want to think about is error checking. You might consider looking at the FileMaker Pro documentation for error codes. POP3it also has several error codes that you can use. Refer to the section "Understanding Error Responses" for help with POP3it error codes.

Example POP3it Script

Included with the POP3it archive is an example FileMaker database file called "POP3it_Example". This database file has several scripts in it demonstrating how to incorporate the POP3it functions into your database. In the following pages, we will take the concept of downloading email as previously explained, and show you how that translates into a FileMaker script.

You should have a basic knowledge of FileMaker scripting to use this script, you may need to refer to the FileMaker documentation and integrated help to better understand the script functions and steps.

First of all, here are the Fields that are necessary to run the "Get Email" script that will be explained on the following pages.

Field Name	Field Type
Complete Email	Text
Result	Global(Text)
POP3 Mail Host A	Text
POP3 Password A	Text
POP3 Username A	Text

This script assumes that you have the above fields in your database. Starting on this page is the script text. Following the text will be a step by step walk through of the script which explains what each script step is and why we have used it.

Script Step	Script Parameters
Enter Browse Mode	
Freeze Window	
Go to Layout	Specify: Get Email Example
Toggle Window	Specify: Zoom
Set Field	Field Button: Result

Script Step	Script Parameters
	Specify Button: External("POP3-Host", POP3 Mail Host A) & External("POP3-Username", POP3 Username A) & External("POP3-Password", POP3 Password A)
Set Field	Field Button: Result
	Specify Button: External("POP3-OpenConnection", "")
Set Field	Field Button: Result
	Specify Button: External("POP3-AcquireFirst", "")
If	Specify Button: Result = 0 or LeftWords(Result, 1) = "ERROR"
Show Message	Specify Button: There is either no email on the server, or an error occurred. For more information, take a look at the result field. Button Captions: "OK", "", ""
Set Field	Field Button: Result
	Specify Button: Result & "" & External("POP3-CloseConnection", "")
Exit Script	
Else	
Loop	
New Record/Request	
Set Field	Field Button: Complete Email
	Specify Button: External("POP3-CurrAll", "BOTH")
Beep	
Set Field	Field Button: Result
	Specify Button: External("POP3-CurrDelete", "")
Set Field	Field Button: Result
	Specify Button: External("POP3-AcquireNext", "")
Exit Loop If	Specify Button: Result = 0 or LeftWords(Result, 1) = "ERROR"
End Loop	
End If	

Script Step	Script Parameters
Set Field	Field Button: Result
Beep	Specify Button: External("POP3-CloseConnection", "")
Show Message	Specify Button: You may have new mail
	Button Captions: "OK", "", ""

Now let us go through this script step by step.

Enter Browse Mode

This script step should be at the beginning of most, if not all, of your scripts. Placing the Enter Browse Mode script step at the beginning of your scripts can help alleviate problems when starting a script while in another mode, usually by accident.

Go to Layout	Specify: Get Email Example
--------------	-------------------------------

Second, go to the Get Email Example Layout. This is just a simple layout that holds the fields that we need to view the email that we are downloading.

Set Field	Field Button: Result
	Specify Button: External("POP3-Host", POP3 Mail Host A) & External("POP3-Username", POP3 Username A) & External("POP3-Password", POP3 Password A)

The first POP3it related step is to set up the Host, Username, and Password for the account you are going to be checking.

Set Field	Field Button: Result
	Specify Button: External("POP3-OpenConnection", "")

The next step is to open the connection. For most of the POP3it functions, we will be setting the calculation to the Result field.

Set Field	Field Button: Result
	Specify Button: External("POP3-AcquireFirst", "")

At this point, we are ready to check to see if there are any messages on the server. To do so we will want to use the External("POP3-AcquireFirst", "") function.

If	Specify Button: Result = 0 or LeftWords(Result, 1) = "ERROR"
----	--

Now we check to see what information was returned by the POP3it plug-in by doing an If statement checking the contents of the Result field. The calculation checks for two things. First, it checks to see if POP3it returned a Zero. Second, it checks to see if the first word in the Result field is the word "ERROR". If either of these cases are true, we need to inform the user of the result and close the POP3 connection.

Show Message	Specify Button: There is either no email on the server, or an error occurred. For more information, take a look at the result field. Button Captions: "OK", "", ""
--------------	--

This step is executed only if the IF statement returns true. All this step does is open an alert box telling the user that there was either an error or that no mail was on the server.

Set Field	Field Button: Result Specify Button: Result & "" & External("POP3-CloseConnection", "")
-----------	--

This step sets the result field to itself, and then closes the POP3 connection. We have taken this approach so that users can still look at the result field and know what happened in the last step as well as in this step.

Exit Script

At this point we know that there is no email or that there was an error. We have told the user, and closed the POP3 connection. So we need to exit the script because there is nothing left to do.

Else

If we make it to this step, that means that If statement has returned false. In other words, at this point, there are no errors, and there is at least one message to download. So now we need to begin a loop and download the email.

Loop

Here we begin the loop. Since we have already used the "POP3-AcquireFirst" function we are ready to begin retrieving the information for the current email.

New Record/Request

The first step of the loop is to create a new record to put the current email into.

Set Field	Field Button: Complete Email Specify Button: External("POP3-CurrAll", "BOTH")
-----------	--

Here is where we actually get the information from POP3it and put it into the current record. To make this script simple, we have chosen to use the "POP3-CurrAll" function, which will place the entire email header and message into one field, the Complete Email field. This func-

tion will also download any attachments to your hard drive. However, to make it easier to deal with in the long run, we have made several "Current" functions available: "POP3-CurrMessageID", "POP3-CurrHead", "POP3-CurrDate", "POP3-CurrTo", "POP3-CurrFrom", "POP3-CurrSubject", "POP3-CurrBody", and "POP3-CurrAttach". You can easily set several fields to their related functions. For example, if you had a field called From, you could set that field using `External("POP3-CurrFrom", "")` to get the From Email Address from the current message.

Beep

Because we wanted to keep this script as simple as possible we needed to come up with a way to help you recognize the fact the POP3it is downloading email. To help out with that, we have placed the beep script step here so that it will beep after each time that it downloads a message. There are several alternatives to using a beep. You could easily come up with a solution to go to a different layout and report what email number is being downloaded. When you use the "POP3-AcquireFirst" function at the beginning of this script, POP3it returns the number of messages waiting. Likewise, each time you use the "POP3-AcquireNext" function, POP3it returns the current message number. Using this information you can easily set up a status layout, or even a status bar.

Set Field	Field Button:
	Result
	Specify Button:
	<code>External("POP3-CurrDelete", "")</code>

At this point, we need to tell the server to set this message to be deleted on disconnect. To do this, we use the "POP3-CurrDelete" function.

Set Field	Field Button:
	Result
	Specify Button:
	<code>External("POP3-AcquireNext", "")</code>

Now that we have retrieved a message, we need to check to see if there are any more messages. If there are no messages left, POP3it will return a zero. However, if there are more messages, POP3it will return the number of the message.

Exit Loop If	Specify Button:
	<code>Result = 0 or LeftWords(Result, 1) = "ERROR"</code>

After using the "POP3-AcquireNext" function, we need to test to see what steps to take next. If there are no messages left, we need to exit the loop and close the POP3 connection. We also need to exit the loop if an error occurs.

End Loop

If there are no errors, then the script will now go back to the beginning of the loop to retrieve the current message. This process will continue until the "Exit Loop If" step proves to be true.

End If

This stop marks the end of the If/Then statement.

Set Field	Field Button: Result Specify Button: External("POP3-CloseConnection", "")
-----------	--

This is one of the most important steps of the script. To close the POP3 connection, simply use the "POP3-CloseConnection" function as show above.

Beep

This is just one last beep to get the user's attention along with the message shown in the next step.

Show Message	Specify Button: You may have new mail Button Captions: "OK", "", ""
--------------	--

This last step simply informs the user that they "may have new mail". We have used the word "may" here because we did not do any checking before hand to see how many messages there were before we started downloading messages. We took this approach so as to keep this script as simple as possible.

Extract Information Example

Here is an interesting way to extract Field data out of an email message and insert it into fields in your database. This script is included in the POP3it_Example database and is named "New Addressbook Entry Script". Say for example you wanted to email a message to an email address that would be imported into FileMaker Pro. In this message you could have the name of your fields and the value that you wanted to put in those fields. The format for this example should be as follows: Field Name, Colon, Space, Field value, Carrage Return. For example:

Name: John Doe
Address: 123 Unknown Street
City: Lost
State: Somewhere
Zip: 12345

With the "New Addressbook Entry Script" you can grab that information from the body of your email and place it into it's related field. The great thing about this script is that you can have your fields anywhere in the email message, plus you can have fields in your database layout that are not in your email message. The script is smart enough to test your field names against the information in the body of the message and only grab the information that matches. It is also smart enough to not overwrite any information in your database that is not in the message.

This script will go through all of the fields on a specific layout, examine each field name and look at the text in the body field to see if there is a value for the current field. Therefore, all the

fields that you want to use in the email message should be on the layout that you run this script on.

These are the fields that you need in order to re-create this script:

Field Name	Field Type
_Start Field	Global Text
Body	Text
Name	Text
Address	Text
City	Text
Zip	Text
Phone	Text
Fax	Text
Email	Text
Web	Text

Here is what the entire script looks like:

Script Step	Script Parameters
Enter Browse	Mode
Go to Layout	Specify: Addressbook
Go to Next Field	
Set Field	Field Button: _Start Field Specify Button: Status(CurrentFieldName)
Go to Next Field	
Loop	
If	Specify Button: PatternCount(Body, Status(CurrentFieldName))
Paste Result	Select Entire Contents: Checked Field Button: Do NOT specify. Specify Button: Middle(Body, Position(Body, Status(CurrentFieldName) & ":", 1, 1) + (Length(Status(CurrentFieldName)) + 2), Position (Body, "", Position(Body, Status(CurrentFieldName) & ":", 1, 1), 1) - (Position(Body, Status(CurrentFieldName) & ":", 1, 1) + (Length(Status(CurrentFieldName)) + 2)))
End If	
Exit Loop If	Specify Button: _Start Field = Status(CurrentFieldName)
Go To Next Field	

Script Step	Script Parameters
End Loop	
Go to Layout	Specify: Detail

There are several things to note with this script. One thing to note is the fact that you do not want to specify a field name in step seven. If you do, the script will not work at all. By not selecting a field, the script is allowed to cycle through each field during the loop. Another thing to note is how the script uses the _Start Field field to exit the script by first setting the field to the first field on the layout. To get into the first field you use the Go to Next Field script step. After setting the _Start Field up with the current field name, we then use the Go to Next Field script step again so that we can make a complete cycle through all of the fields and exit the script with the field that we started with.

Another thing to note is the use of the Paste Result field. As stated previously in this documentation, FileMaker requires that the field you are pasting into be on the current layout any time you use a paste function. For this reason, any fields that you have in your email message that are not on the current layout will be ignored.

Our last script step sends us back to the detail layout because this script is used automatically with our Check Email script if there is a subject of "New Addressbook Entry".

Though we have not gone into the complete detail of this script, you can probably understand its usability by testing it out along with the various related scripts in the POP3it_Example.fp3 database.

POP3it External Functions Reference

The following is a list of all available functions and a complete description of each. Note that there are a few functions that can have default values set in the Configuration Dialog. See the Installation and Configuration section for a complete understanding of how the Configuration Dialog works. Functions with an asterisk ("*") are new with version 2.0.

POP3-Version

This function returns the POP3it version information to the field that you are setting. The parameter should be set to "".

POP3-Register

This function allows you to register your copy of POP3it. This function takes the form of:

```
External("POP3-Register", "First Name|Last name|Serial Number").
```

In other words, the parameter consists of your first name, last name, and serial number all separated by the pipe character ("|"). ("|" is created by typing shift-backslash.) The serial number is made up of the numbers 0 through 9 and letters A through F. There are no letter O's, they are all zeros.

POP3-Host

The "POP3-Host" function is where you set the POP3 mail host for the email account that you are currently working with. This function looks like this:

```
External("POP3-Host", "mail.yourdomain.com")
```

POP3-Username

Use the "POP3-Username" function to assign the username for the email account that you are currently working with. An example of this function would be:

```
External("POP3-Username", "username")
```

Your username is the first portion of your email address, for example, if your email address were "bob47@yahoo.com", your username would be "bob47".

POP3-Password

Use this function to assign the password for the email account that you are currently working

with. An example of this function is:

```
External("POP3-Password", "mypassword")
```

POP3-AttachPath

This function allows you to dynamically set the path on your hard drive that you want to save incoming attachments in. You may want to use this function if you are checking multiple email accounts and want an attachments folder for each account. This function looks like:

```
External("POP3-AttachPath", "C:\Attachments\")
```

for Windows, or:

```
External("POP3-AttachPath", "Macintosh HD:Attachments:")
```

for Macintosh.

POP3-DlgAttachPath

The "POP3-DlgAttachPath" function does the same thing as "POP3-AttachPath", except that it presents you with a dialog so that you can select the path to save the attachments to. This function has no parameter and looks like:

```
External("POP3-DlgAttachPath", "")
```

POP3-HTMLExportPath*

This function allows you to dynamically set the path on your hard drive that you want to save HTML exports in. This function looks like:

```
External("POP3-AttachPath", "C:\HTMLExports\")
```

for Windows, or:

```
External("POP3-AttachPath", "Macintosh HD:HTMLExports:")
```

for Macintosh.

POP3-DlgHTMLExportPath*

The "POP3-DlgHTMLExportPath" function does the same thing as "POP3-HTMLExportPath", except that it presents you with a dialog so that you can select the path to

save the HTML Exports to. This function has no parameter and looks like:

```
External("POP3-DlgHTMLExportPath", "")
```

POP3-OpenConnection

This function simply opens a connection to your POP3 mail host based on the host, username and password that you have previously assigned. This function has no parameter and looks like:

```
External("POP3-OpenConnection", "").
```

POP3-AcquireFirst

The "POP3-AcquireFirst" function returns the number of email messages waiting to be downloaded at the time of your connection. If there are one or more messages waiting then you can use the "Current" functions to retrieve information for the first message. Like most of the POP3it functions, this function has no parameter and looks like:

```
External("POP3-AcquireFirst", "")
```

POP3-CurrUniqueID

This function, as do all "Current" functions, requires that you have already made a connection to your POP3 mail host and that there be at least one email message on the server. If those conditions are met, this function will give you the Unique Message ID of the current mail message. Example:

```
External("POP3-CurrUniqueID", "")
```

POP3-CurrSize

This function retrieves the size of the current email that you are working with. Example use:

```
External("POP3-CurrSize", "")
```

POP3-CurrHead

This function retrieves the email header for the current email that you are working with. Do not get this confused with "POP3-CurrHeader" which returns only one specific header. Example use:

```
External("POP3-CurrHead", "")
```

POP3-CurrDate

This function retrieves the Date from the current email message. Example:

```
External("POP3-CurrDate", "")
```

POP3-CurrTo

This function gets the To field from the current email message. Example:

```
External("POP3-CurrTo", "")
```

POP3-CurrFrom

The "POP3-CurrFrom" function retrieves the From Address for the current email message. Example:

```
External("POP3-CurrFrom", "")
```

POP3-CurrSubject

This function grabs the Subject for the current message. Example:

```
External("POP3-CurrSubject", "")
```

POP3-CurrBodyType*

This function will tell you what type of Body the current message has. It will return "TEXT" if the email only has a plain text part, "HTML" if the email only has an HTML part, or "BOTH" if the email is "multipart/alternative" and has both Text and HTML parts.

POP3-CurrBody

This function grabs the Body for the current message. New to version 2.0: you can now specify what type of body to return.

If you only want a plain Text part, use the parameter "TEXT" like so:

```
External("POP3-CurrBody", "TEXT")
```

If you only want an HTML part, use the parameter "HTML". For "multipart/alternative" emails that have both Text and HTML parts, you can use the parameter "BOTH" to get the Text part followed by the HTML part. For single part emails, "BOTH" will return which ever type is available. There are also two other parameters you can use: "Text, HTML" and "HTML, Text". The first will return a plain Text body to you if there is one. However, if there

is not a plain Text part, then it will return an HTML part. The second will return an HTML part, unless there is not one, and then it will return a plain Text part. If you do not specify a type with the parameter, POP3it uses the default body type in the Configuration Dialog.

POP3-CurrAttach

The "POP3-CurrAttach" function places the attachment in the attachment directory that you have specified by using the "POP3-AttachPath" function, "POP3-DlgAttachPath" function, or Configuration Dialog. This function also returns the path of the file to the field that you are setting. Example use:

```
External("POP3-CurrAttach", "")
```

POP3-CurrHTMLExport*

The "POP3-CurrHTMLExport" function will export the HTML part of an email to a file, complete with any inline images, into the HTML Export directory that you have specified by using the "POP3-HTMLExportPath" function, "POP3-DlgHTMLExportPath" function or Configuration Dialog. This function also returns the path of the file to the field that you are setting. Example use:

```
External("POP3-CurrHTMLExport", "")
```

POP3-CurrAll

This function retrieves the entire message into one field, including all email headers and the body of the message. It also downloads any attachments into your previously defined attachment directory. The parameter to this function is exactly like the parameter in the "CurrBody" function. Example use:

```
External("POP3-CurrAll", "BOTH")
```

POP3-CurrHeader

This function will retrieve any header out of the set of email headers by name. You can use it to extract non-standard headers such as "X-Sender" or "X-Mailer". Do not get this mixed up with "POP3-CurrHead" which retrieves the entire header of the email. Example use:

```
External("POP3-CurrHeader", "X-Mailer")
```

POP3-AcquireNext

The "POP3-AcquireNext" function retrieves the next email message for you to work with. The function will return the current email message number starting with the second message,

which is where "POP3-AcquireFirst" left off, and report each message number until no messages are left. When there are no messages left, the "POP3-AcquireNext" function returns zero. In other words, when you have a FileMaker Pro script loop you can use this function to see what the current message is. When there are no messages left on the server, this function will return a zero giving you a test to break out of the loop. This function has no parameter and looks like:

```
External("POP3-AcquireNext", "")
```

POP3-CurrDelete

This function sets the current message to be deleted on disconnect. Example use:

```
External("POP3-CurrDelete", "")
```

POP3-DeleteAll

The "POP3-DeleteAll" function sets every message in your current email session to be deleted upon disconnect. Example use:

```
External("POP3-DeleteAll", "")
```

POP3-UnDeleteAll

The "POP3-UnDeleteAll" function resets all the deletions you may have made so that none of the emails are deleted on disconnect. Example use:

```
External("POP3-UnDeleteAll", "")
```

POP3-CloseConnection

This function closes the connection with your POP3 mail host. Example use:

```
External("POP3-CloseConnection", "")
```

Understanding Error Responses

Every function in POP3it returns a response indicating the success or failure of that function. If the function is successful, it will return a response indicating that it set the value you were trying to set, or completed the task that needed to be completed, or it will return the value that you were requesting. If however, the function is not successful, it will return an Error Response. This Error Response is in the form of:

ERROR: <Function Name>: <Error Description>

For example, if you try to retrieve the subject of an email before you use AcquireFirst, the POP3-CurrSubject function will return the following Error Response:

ERROR: CurrSubject: Use AcquireFirst or AcquireNext to load a message.

Error responses always start with the word "ERROR" in all caps, followed by a colon, followed by the function that had the error, followed by a colon, followed by the actual error that occurred. You can use the various FileMaker Text Functions to extract the different parts of the Error Response for your own use.

Credits

Programming and documentation by Jake Traynham
Concept, web design, and documentation by Jesse Traynham

Contact Information

Email: POP3it@comm-unity.net
POP3it Website: <http://www.pop3it.com>
Main Website: <http://www.filemaker.comm-unity.net>
Phone: 817-560-4226

You can write us at:

Comm-Unity Networking Systems
8652 Hwy 80 West
Fort Worth, Texas 76116