

[Contribute](#) : [Advanced Search](#) : [Directory](#) : [Forum](#) : [FAQ's](#) : [My Downloads](#) : [Links](#) : [Polls](#)



Changing the world one server at a time.

[advanced search](#)

[Welcome to AFP548](#)

Monday, March 14 2011 @ 04:18 am MDT

Sections

[Home](#)
[AFP548 Site News](#) (108/2)
[Articles](#) (291/34)
[Apple](#) (54/2)
[Extended KB](#) (3/1)
[Ask AFP548](#) (16/46)
[Reviews](#) (5/2)
[Security](#) (17/2)
[Tips](#) (133/8)
[Third Party Applications](#) (57/6)
[Odds and Ends](#) (28/1)
[Podcasts](#) (2/0)

User Functions

Username:

Password:

Don't have an account yet? Sign up as a [New User](#)
 Lost your [password](#)?

Poll

What's your favorite OpenSource management tool?

- radmind
 Puppet
 Munki

[Results](#)

38 votes | [1 comments](#)

Hosting Provided By:


powered by [macminicolo.net](#)

Exchange ActiveSync on Mac OS X Server

Friday, July 09 2010 @ 03:38 pm MDT

Contributed by: [macshome](#)

Views: 12,782

Let the headline sink in for a second there. It brings up a lot of questions doesn't it? Things like how would I do that? Why would I do that? What will it get me? These are all good questions to raise! Over this two part series we will answer all of them and gain some functionality along the way. In part one we will discuss the what, the how, the why, and get EAS mail running on Mac OS X Server along the way. 

Read on for more...

What is Exchange ActiveSync?

It's often best to begin at the beginning, so let's talk a bit about what ActiveSync is. ActiveSync came to the world as an add-on for Windows 95. There it was designed to help keep things in, well, sync. Folders, PocketPCs, even other computers could all be kept in sync with things like mail accounts, bookmarks, and music. It's actually quite a bit like MobileMe sync today. This was refined over the years and now is primarily for keeping your Windows Mobile devices in sync with your desktop. Even right now you can hop over to Microsoft.com and grab a copy for Windows 7 and start syncing.

Exchange ActiveSync (or EAS for short) is based on the same protocol as the more pedestrian ActiveSync, but is a bit more specific in it's goals. EAS is all about syncing a mobile device to an Exchange account. Typically this means mail, calendar, and notes. EAS goes a bit further and allows for the provisioning of basic policies such as passcode, screen lock, and encryption requirements. One of the most prominent EAS actions is the ability to wipe a remote device in case it is lost.

All of this is done over a fairly simple brew of HTTP and XML. Coming from the same company that gave use MAPI this is a breath of fresh air. Not so surprising is that the licensing is a bit, different. Rather than license EAS on a code basis, it is licensed on a protocol basis in the US. This means that EAS licensees such as Apple are free to implement the code anyway they see fit. What they are licensing is the right to use the protocol in general. This has had a bit of a stifling effect on some of the OSS groupware options out there, but the commercial solutions like Kerio and Gmail remain available. In fact, EAS has become one of the most available connection options out there for mobile devices, even without any MS software in the picture. This brings us to the why question...

Why use EAS?

Why would you want to sully your Mac OS X Server with Exchange ActiveSync? Apple provides a very nice mail server and even a nice push notification service on 10.6 Server. The thing is though, that push service only supports the Snow Leopard versions of Mail and iCal. What if you want push mail support for the iPhone? What happens when you need to support an Android or Windows-based device? Either you give them all a taste of 2nd tier status, or you support them the best you can with a common protocol. When it comes right down to it, EAS is the push protocol with the widest array of support.

How can I do any of this?

This is all fine and dandy, but how can I implement it and what does it get me? As with many protocols, there is an OSS solution out there. In this case, [z-push](#) is the project we are looking at. It's a simple example of EAS, implemented in PHP. Like many OSS projects, it's the open core of a commercial project ([Zafara](#)) and therein lies the rub. The protocol part of z-push is just how it talks to clients. The data comes from a backend connector that talks to the services. Currently there are backends for Zafara, IMAP, maildir, and vcard. The project maintainers put the majority of effort into the Zafara backend and understandably so. Of the remaining backends that are included in the project only the IMAP provider is directly applicable to our server. Anyone that has the skills though can create a new backend and in part 2 we will be using the iRemoteWipe backend to implement remote wipe for our devices.

Getting started

OK. Enough talk! Let's start installing things!

Observant readers will have read the previous section, put my mentions of PHP and IMAP together, and are now thinking out loud, "Hey! Apple's PHP doesn't ship with IMAP support!". This is of course correct, but it's easy enough to build that module and then tell the Apple PHP module to use it.

So make sure you have Xcode installed on a server and let's dive right in.

This is all easiest when done from the Terminal so fire it up. Now we need to download the latest IMAP source code. We can just stick this in our Downloads folder to make it logical, but you can put it anywhere you want really.

```
cd ~/Downloads
curl -O ftp://ftp.cac.washington.edu/imap/imap-2007e.tar.gz
tar -zxvf imap-2007e.tar.gz
```

Now we need to build the IMAP libraries and put them where they need to be.

```
cd imap-2007e/
make osx EXTRACFLAGS="--arch i386 -arch x86_64 -g -Os -pipe -no-cpp-precomp"
```

This will take a while and you get to watch a lot of text scroll by. With that done let's copy the modules.

```
sudo mkdir -p /usr/local/include
sudo cp c-client/*.h /usr/local/include/
sudo mkdir /usr/local/lib
sudo cp c-client/*.c /usr/local/lib/
sudo cp c-client/c-client.a /usr/local/lib/libc-client.a
```

Now we need to do the same thing with PHP. First the download the 5.3.1 code and put it somewhere logical.(I wasn't able to get this to build with the current 5.3.2 tarball.)

```
cd ~/Downloads
curl -O http://us.php.net/distributions/php-5.3.1.tar.gz
tar -zxvf php-5.3.1.tar.gz
```

With that done we can build out just the IMAP extension that we need. Again this will take a while to build.

```
cd php-5.3.1/ext/imap
phpize
./configure --with-ldap=/usr/local/imap-2007 --with-kerberos --with-ldap-ssl=/usr/
make
```

The next bit is to copy the IMAP extension to our PHP extensions folder.

```
sudo cp modules/imap.so /usr/lib/php/extensions/no-debug-non-zts-20090626/
```

Now it's time to tell PHP to use our new extension. We just need to open up the php.ini file and tell it. So first...

```
sudo vi /etc/php.ini
```

and now find the section that has some "extension=" parts in it and add the following line:

```
extension=/usr/lib/php/extensions/no-debug-non-zts-20090626/imap.so
```

Now save the file. Open up Server Admin and make sure that the PHP plugin is enabled for the Web server. Once you are sure, go ahead and restart the Web service to load the changes.

Now we have PHP IMAP functioning with our default PHP install and we can tackle the actual z-push install.

Installing z-push

This part is actually so easy it's going to blow your mind a bit. First pop over to the z-push project page and grab a copy of the latest release build.

```
cd ~/Downloads
curl -O http://download.berlios.de/z-push/z-push-1.4.tar.gz
```

Once that is done, unpack it to your web root. By default on Mac OS X Server it's in /Library/WebServer/Documents.

```
tar zxvf z-push-1.4.tar.gz -C /Library/WebServer/Documents
```

With that done we need to make a few more config changes. So let's switch over to the web root directory.

```
cd /Library/WebServer/Documents/z-push
vi config.php
```

Take a look in here and we need to change two settings. The first is to set the default timezone, the other is to tell z-push that we want to use the IMAP backend. For this first setup you can leave the other settings in here alone.

You can lookup your supported PHP timezone syntax [here](#). In my case I'm working in EST so my section looks like this. Adjust as needed for your location.

```
// Defines the default time zone
if (function_exists("date_default_timezone_set")){
    date_default_timezone_set("America/New_York");
}
```

Now scroll down and change the \$BACKEND_PROVIDER = line to read like this.

```
$BACKEND_PROVIDER = "BackendIMAP";
```

Save the file and we should be back in the z-push directory now.

Next you need to rename, or remove, the ICS backend file so things will work properly.

```
mv backend/ics.php backend/ics.php.old
```

We need to make sure that the www user can write to the 'state' folder inside z-push. It's easy just to change the ownership on it.

```
sudo chown _www:_www state
```

Now we need to make a few realm specific Apache options in here. The easy way to do this is to create a .htaccess file in the z-push directory that has the settings we need.

```
vi .htaccess
```

and make the contents just this:

```
php_flag magic_quotes_gpc off
php_flag register_globals off
php_flag magic_quotes_runtime off
php_flag short_open_tag on
```

Now fire Server Admin back up and we just need to add an alias!

Firstly I would make sure that you are doing all of this over SSL. So go ahead and enable it for the site in Server Admin. Now click on the Aliases tab for your site and add the following URL Alias.

```
Alias /Microsoft-Server-ActiveSync /Library/WebServer/Documents/z-push/index.php
```

Restart Web Services and we are all done!

As a quick and dirty test you can point a web browser on your server at at your https://127.0.0.1/Microsoft-Server-ActiveSync . Authenticate as any mail user and you should get a polite message that you should really be doing this from a mobile device. With that done we can move on to configuring an iPhone!

iPhone Exchange configuration

On an iPhone go into Settings and add a mail account. You want to select Exchange as the type of account and enter your information here. Since we don't have the Exchange 2007 auto-discovery here it will spin a second and then ask for your server name. If you are using an unsigned cert then you will also get a dialog about that. Only select that you want to sync Mail to the EAS server and you are all done. If you have mail in the account you should get a delivery noise in a few seconds.

When you want to, you can take a look back in the iPhone settings and adjust what folders you get push notifications on. If you get a lot of list mail you will probably not want to push those folders as all the incoming mail will make your iPhone sound like a pinball machine!

Wrapping up

So now you have EAS up and running against your local dovecot IMAP server on Mac OS X Server 10.6. The more adventurous of you may want to play around in the z-push/config.php file and see what happens when you point this at other, non-push enabled, IMAP servers. You can use this to access e-mail now with any ActiveSync client, so that expands the push reach of your mail to include Windows Mobile and Android devices as well. Regular IMAP clients can still use the server as they did before, and actually that's how you setup things like folder structures for your e-mail.

Remember that ActiveSync is for SYNCING things, so you can't do stuff like file mail from your inbox into other mail folders if you are in airplane mode. Any changes you make via IMAP will be almost instantly reflected on your EAS device though.

In part two we are going to take a look at remote wipe of devices and explore the possible use of EAS policy on iPhones via Mac OS X Server.

Until then, have fun and read the man pages!

2

What's Related

- [z-push](#)
- [Zafara](#)
- [here](#)
- [More by macshome](#)
- [More from Articles](#)

Story Options

- [Printable Story Format](#)

Advertising



Exchange ActiveSync on Mac OS X Server | 15 comments | [Create New Account](#)

Oldest First | Threaded | Refresh | Post a comment

The following comments are owned by whomever posted them. This site is not responsible for what they say.

Exchange ActiveSync on Mac OS X Server

Authored by: Anonymous on Friday, July 09 2010 @ 06:54 pm MDT

This seems to imply that push to iPhone works from Snow Leopard Server. I have never seen this work and this KBase says it doesn't: <http://support.apple.com/kb/HT3947>

So this is a bit off topic, but does anyone have SLS mail and calendar pushing to an iPhone?

[[Reply to This](#) | #]

- [Exchange ActiveSync on Mac OS X Server](#) - Authored by: [macshome](#) on Friday, July 09 2010 @ 07:57 pm MDT

Exchange ActiveSync on Mac OS X Server

Authored by: [philspaces](#) on Saturday, July 10 2010 @ 12:55 pm MDT

So I got the pushing to work, but cannot move messages or delete them on the phone. Is that supposed to work?? Great Article.

[[Reply to This](#) | #]

- [Exchange ActiveSync on Mac OS X Server](#) - Authored by: [macshome](#) on Saturday, July 10 2010 @ 04:01 pm MDT
 - [Exchange ActiveSync on Mac OS X Server](#) - Authored by: [donovan](#) on Thursday, July 15 2010 @ 07:01 am MDT
 - [Exchange ActiveSync on Mac OS X Server](#) - Authored by: [macshome](#) on Thursday, July 15 2010 @ 09:17 am MDT
 - [Exchange ActiveSync on Mac OS X Server](#) - Authored by: Anonymous on Thursday, July 29 2010 @ 09:16 am MDT
 - [iPhone mail delete](#) - Authored by: [macshome](#) on Wednesday, August 18 2010 @ 02:48 pm MDT

Be Careful About the License for z-push

Authored by: [gerritdewitt](#) on Wednesday, August 11 2010 @ 02:42 pm MDT

Very good article about how to implement z-push! However, I should point out that I think you need to elaborate on the EAS protocol being licensed on a protocol-basis, not a code basis:

In short, Microsoft has one or more United States patents which cover the method that is the EAS protocol. These are software patents, and, unfortunately, they're enforceable in the United States. Thus, any software, open-source or not, that implements the EAS protocol would infringe on those patents without having permission from Microsoft. Typically, that permission is obtained through a "patent license," and that's exactly what Apple did when it created its own EAS implementation in iOS and Snow Leopard.

So, in short, if you're going to be using z-push in the United States, you still need an EAS license from Microsoft. Perhaps including some information about how to obtain said license would be beneficial to the readers.

The need for the "patent license" is also mentioned in the License file that ships with z-push. It's GPLv2, but it's specifically excluded for use in the United States for this very reason. You'll also notice that Zarafa is based in Europe where method/software patents are rare.

Once again, I don't mean to be a killjoy because your article is extremely thorough. :)

--Gerrit, ACSA

[[Reply to This](#) | #]

Exchange ActiveSync on Mac OS X Server

Authored by: Anonymous on Thursday, September 02 2010 @ 08:48 pm MDT

I tried this and it kind of works (it does push), but the two main dealbreakers for me are:

1. no html mail
2. many attachments don't work

I have attempted to find solutions to the above problems but none of the fixes supplied above and in the other zpush forums have been successful. I am seriously considering using Kerio Connect. It isn't free, but it works.

Has anyone managed to get html mail and/or attachments to work reliably?

[[Reply to This](#) | #]

Sending mails

Authored by: [hetjan](#) on Wednesday, September 15 2010 @ 06:11 pm MDT

I followed these direction and they worked great, except for one small thing: Sent mails didn't get saved in the Sent folder.

To fix this on my 10.6.4 server I edited

```
/Library/WebServer/Documents/z-push/config/imap/config.php
```

```
and changed the line that reads
define('IMAP_SENTFOLDER', '');
to
define('IMAP_SENTFOLDER', 'Sent Messages');
```

The only thing I have left to fix is the move-issue; mails moved by iPhone don't get moved.

[[Reply to This](#) | #]

Exchange ActiveSync on Mac OS X Server

Authored by: Anonymous on Friday, September 17 2010 @ 06:32 am MDT

Hi macshome,

this is a very nice tutorial which works well.

But a problem occurs when starting WebDAV on the site the z-push process is executed: In this case the sync doesn't work anymore - I get an error alert in the iPhone that the Account information could not be validated. The test access with a browser works fine.

I tried a lot with different realms but didn't get it running up to now. Do you have an idea how this should be configured?

Thanks a lot!

Frank

[[Reply to This](#) | <#>]

Exchange ActiveSync on Mac OS X Server

Authored by: Anonymous on Wednesday, November 03 2010 @ 01:49 pm MDT

i would like to follow the first part of these instructions to recompile PHP on a new Snow Leopard server.

can anyone tell me whether this will work seamlessly with the existing PHP install that ships with SL? i see the author chose a different PHP version to compile...

also, i see only one extension listed on the compile instructions... i need other extensions in PHP, will i have to recompile with all those other ones or will these instructions just add the imap extension?

basically, i've never recompiled PHP before and i'd like some confidence that i am not going to disable the PHP functionality altogether.

thanks in advance for any help!

[[Reply to This](#) | <#>]

- [Exchange ActiveSync on Mac OS X Server](#) - Authored by: Anonymous on Thursday, November 04 2010 @ 08:25 am MDT

Exchange ActiveSync on Mac OS X Server

Authored by: [sap](#) on Monday, November 15 2010 @ 09:08 am MST

This is awesome!

Thank for the explanations and for your precious hints.

Now I have the problem of moving/delete mail, but most of all: there is a chance to have the calendar push events?

Thanks again.

[[Reply to This](#) | <#>]