Problems with converting Mac Eudora 6 from CFM to Mach-O

I spent a couple of weeks starting May 2004 attempting to convert Eudora from a CFM executable to Mach-O (using CodeWarrior). I did not succeed. This is what I discovered along the way:

1) Eudora uses many libraries that are CFM. Each of these would have to be replaced, since we can't link using CFM libraries if we are trying to output a Mach-O executable. Here's a (partial) list, with comments about each one:

* LDAP - we link against an ancient LDAP library, that has not been built from the sources we have since 1999 (or so). Mac OS X ships with the successor to this library, OpenLDAP already installed, and we should use that instead. This will require some work.

* Spell Check - we use a custom spell check library, from WinterTree Software. We have the source to this library, (although it, too has not been built for several years) so it could be ported to Mach-O. Alternately, we could use the spelling routines provided as part of Mac OS X, enabling us to share dictionaries with other programs.

* Kerberos - The MIT kerberos libraries are available on Mac OS X as Mach-O shared libraries, and we will need to convert Eudora to use those.

* Hesiod - the hesiod library we use (and do not have source for) is CFM only. There are various open source Hesiod implementations, though.

2) Source code changes. These fall into two categories:

* Bringing code to Mach-O. As Apple has upgraded Mac OS X, they have changed the interfaces or behaviors of some of the OS calls. These changes have not been ported back to the CarbonLib (CFM) libraries that Eudora links against. When we move to Mach-O, and start linking against Carbon.Framework, then we will probably run into a few cases. However, we don't know how many there will be.

* Removing bridging code. There are some technologies on Mac OS X that are Mach-O only, and Eudora contains bridging code to deal with them. This code can be removed. Two examples of this code are (a) The Address Book integration, and (b) OpenSSL support.

3) Plugins will be a problem. This can be dealt with either via rewriting the plugin code in Eudora to load non-native plugins, or by requiring plugin vendors to rebuild their plugins.

4) This does not address updating Eudora to remove usages of deprecated technologies in Mac OS X (<http://developer.apple.com/documentation/Carbon/Conceptual/newtocarbon/LegacyInterfaces/chapter_4_section_1.html>). See the document "Mac Eudora Alternatives" for a more complete discussion of these issues.