Evolution of an Open Source Project

The LogicMail Story

Derek Konigsberg
octo@logicprobe.org
http://logicmail.sf.net/

Why did I start the project?

I just got a BlackBerry, and I wanted an E-Mail client!

But didn't you say it was a BlackBerry?

These were the included options...

- BlackBerry Enterprise Services (BES)
  - Requires expensive server-side software that only works with MS Exchange and Lotus Notes
- BlackBerry Internet Services (BIS)
  - Periodic INBOX polling and forwarding
    - No synchronization, worked on 15 minute cycles
    - Only did the INBOX
  - Could have BIS-specific account
    - Immediate push
    - Any synchronization required forwarding kludges
  - Some sort of MS Outlook connector
    - Need I say more?

But I just wanted an E-Mail client!

So what exactly can you run on a BlackBerry?

- BlackBerry is a Java-based mobile platform
  - J2ME, MIDP-2.0, CLDC-1.1, to be specific
- It'll run two kinds of Java applications:
  - Applications written to the BlackBerry API
  - Applications written to a more general J2ME API

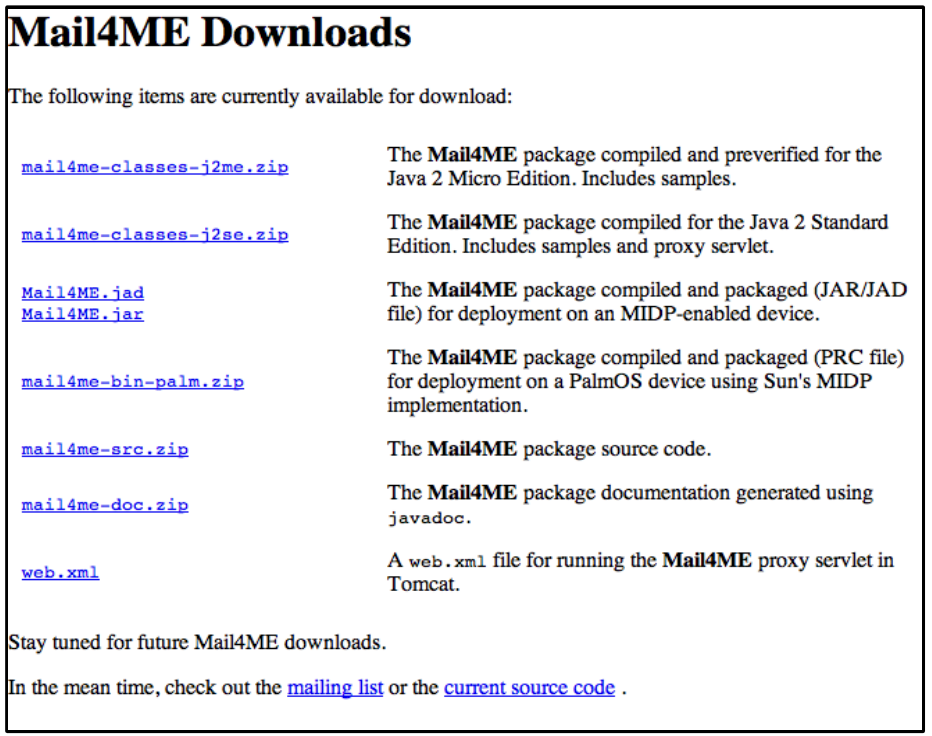
This must have been done before! Let's see what my options are:

- Mail4ME
  - Defunct project
  - Not very usable
  - At least it was open-source
- iCJKMail
  - Simplistic UI
  - Commercial, required some sort of subscription
  - I don't remember why, but I wasn't impressed
- ReqWireless Email Viewer
  - Seemed very promising
  - Commercial, company lost interest
  - Product vanished from availability

I guess I'll just have to write it myself...

Now to get started

- RIM has a complete BlackBerry SDK
- Mail4ME has some reusable code
- This should really be open-source



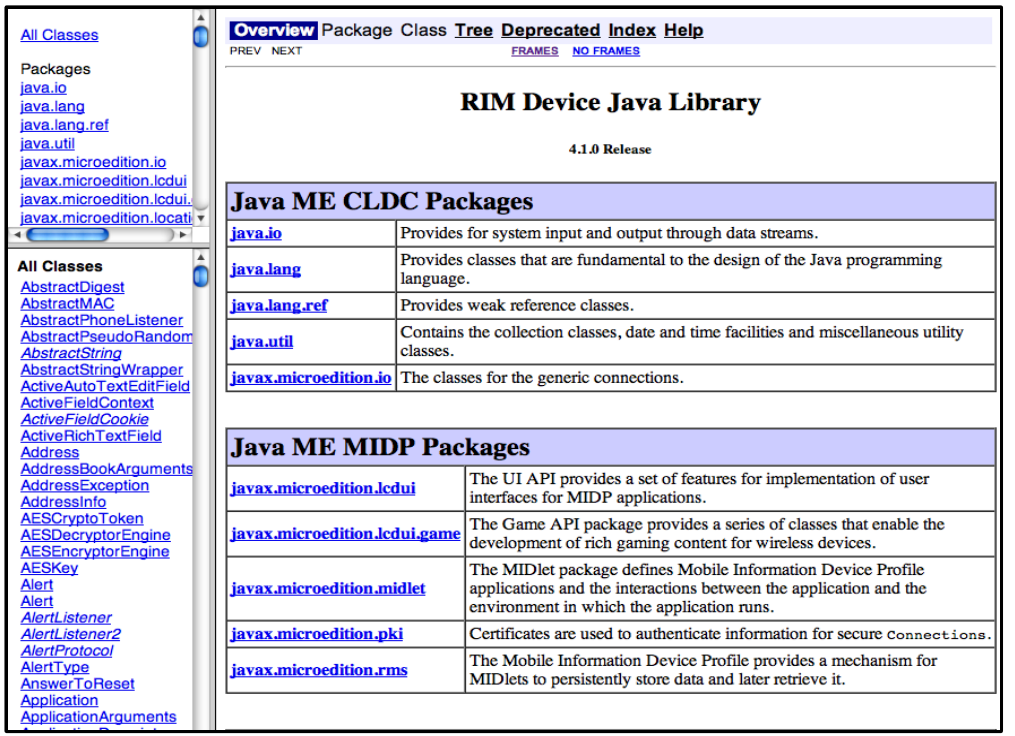
Mail4ME Downloads

The following items are currently available for download:

mail4me-classes-j2me.zip	The **Mail4ME** package compiled and preverified for the Java 2 Micro Edition. Includes samples.
mail4me-classes-j2se.zip	The **Mail4ME** package compiled for the Java 2 Standard Edition. Includes samples and proxy servlet.
Mail4ME.jad Mail4ME.jar	The **Mail4ME** package compiled and packaged (JAR/JAD file) for deployment on an MIDP-enabled device.
mail4me-bin-palm.zip	The **Mail4ME** package compiled and packaged (PRC file) for deployment on a PalmOS device using Sun's MIDP implementation.
mail4me-src.zip	The **Mail4ME** package source code.
mail4me-doc.zip	The **Mail4ME** package documentation generated using `javadoc`.
web.xml	A `web.xml` file for running the **Mail4ME** proxy servlet in Tomcat.

Stay tuned for future Mail4ME downloads.

In the mean time, check out the mailing list or the current source code .



All Classes

Packages
java.io
java.lang
java.lang.ref
java.util
javax.microedition.io
javax.microedition.lcdui
javax.microedition.lcdui.
javax.microedition.locati

All Classes
AbstractDigest
AbstractMAC
AbstractPhoneListener
AbstractPseudoRandom
AbstractString
AbstractStringWrapper
ActiveAutoTextEditField
ActiveFieldContext
ActiveFieldCookie
ActiveRichTextField
Address
AddressBookArguments
AddressException
AddressInfo
AESCryptoToken
AESDecryptorEngine
AESEncryptorEngine
AESKey
Alert
Alert
AlertListener
AlertListener2
AlertProtocol
AlertType
AnswerToReset
Application
ApplicationArguments

Overview Package Class Tree Deprecated Index Help
PREV NEXT FRAMES NO FRAMES

RIM Device Java Library

4.1.0 Release

Java ME CLDC Packages	
java.io	Provides for system input and output through data streams.
java.lang	Provides classes that are fundamental to the design of the Java programming language.
java.lang.ref	Provides weak reference classes.
java.util	Contains the collection classes, date and time facilities and miscellaneous utility classes.
javax.microedition.io	The classes for the generic connections.

Java ME MIDP Packages	
javax.microedition.lcdui	The UI API provides a set of features for implementation of user interfaces for MIDP applications.
javax.microedition.lcdui.game	The Game API package provides a series of classes that enable the development of rich gaming content for wireless devices.
javax.microedition.midlet	The MIDlet package defines Mobile Information Device Profile applications and the interactions between the application and the environment in which the application runs.
javax.microedition.pki	Certificates are used to authenticate information for secure `Connections`.
javax.microedition.rms	The Mobile Information Device Profile provides a mechanism for MIDlets to persistently store data and later retrieve it.

Why open-source?

- Hardly any open-source software exists for this platform
- Everyone wants to nickel-and-dime you for software
- I really don't want to invest the necessary time and resources to go commercial
- Perhaps I can prove that yes, you can do OSS on the BlackBerry

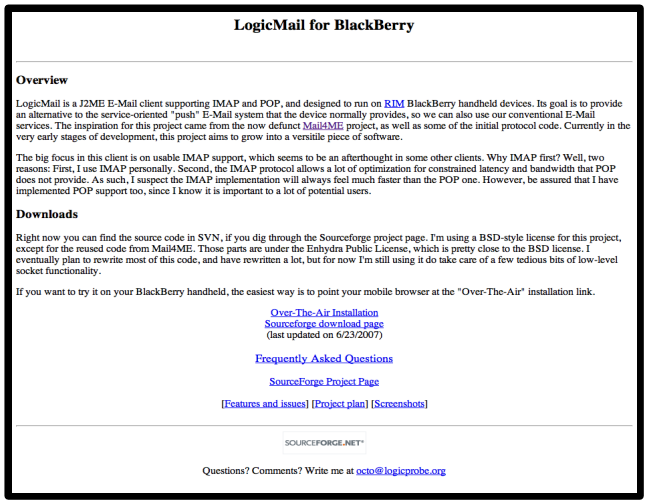
But which license?

VS

I chose the BSD license, though this is always open for debate

And the coding begins...

I created a project on SourceForge



LogicMail for BlackBerry

Overview

LogicMail is a J2ME E-Mail client supporting IMAP and POP, and designed to run on RIM BlackBerry handheld devices. Its goal is to provide an alternative to the service-oriented "push" E-Mail system that the device normally provides, so we can also use our conventional E-Mail services. The inspiration for this project came from the now defunct Mail4ME project, as well as some of the initial protocol code. Currently in the very early stages of development, this project aims to grow into a versitile piece of software.

The big focus in this client is on usable IMAP support, which seems to be an afterthought in some other clients. Why IMAP first? Well, two reasons: First, I use IMAP personally. Second, the IMAP protocol allows a lot of optimization for constrained latency and bandwidth that POP does not provide. As such, I suspect the IMAP implementation will always feel much faster than the POP one. However, be assured that I have implemented POP support too, since I know it is important to a lot of potential users.

Downloads

Right now you can find the source code in SVN, if you dig through the Sourceforge project page. I'm using a BSD-style license for this project, except for the reused code from Mail4ME. Those parts are under the Enhydra Public License, which is pretty close to the BSD license. I eventually plan to rewrite most of this code, and have rewritten a lot, but for now I'm still using it do take care of a few tedious bits of low-level socket functionality.

If you want to try it on your BlackBerry handheld, the easiest way is to point your mobile browser at the "Over-The-Air" installation link.

Over-The-Air Installation
Sourceforge download page
(last updated on 6/23/2007)

Frequently Asked Questions

SourceForge Project Page

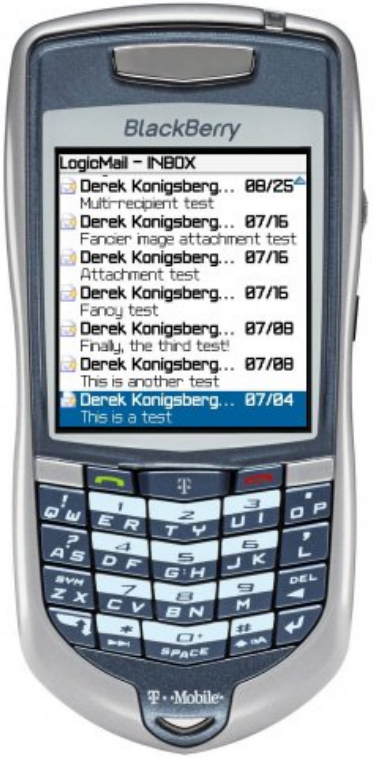
[Features and issues] [Project plan] [Screenshots]

SOURCEFORGE.NET

Questions? Comments? Write me at octo@logicprobe.org

I threw up a very basic-looking website

Once I had something that something worked,
I posted it as a development snapshot



If you post it, the users will find it...

- They're going to have questions
- They're going to run into bugs
- They're going to want more features

You'll get plenty of E-Mails just like this one:

From paolo@XXXX.net Tue Sep 5 09:29:48 2006
Date: Tue, 5 Sep 2006 06:30:20 -0700 (PDT)
From: Paolo XXXX <paolo@XXXX.net>
To: octo@logicprobe.org
Subject: About LogicMail for BlackBerry

Hello,

had a look to your LogicMail for BlackBerry software.

I have a blackberry, enterprise edition and I'd like to be able to read my personal email with that.
It's surprising there are no tools available to do this, unless I miss something.
So for now I'm struggling with Opera Mini and web based email.

LogicMail might be the solution (with full POP3 and SMTP support I mean). A couple of more feature requests, essentially the ability to configure the POP3 (and SMTP in future) port number. My enterprise blackberry gives me Internet access behind the company firewall and I need to change the port number to go through the firewall.
Also, when you implement SMTP, dont forget autentication option for SMTP and the ability to set your "from address".

With that, you'd hava a loyal happy user!

-- Paolo

And you need to keep track of it all..

So you start with the FAQ

Troubleshooting

When I try to connect to my mail server, it fails with an I/O Error.

First, double-check all your configuration settings. You should especially try the "*Use MDS Proxy*" setting, which may or may not be required for your unit. If you are sure they are correct, then chances are that your BlackBerry is not properly configured for TCP networking.

Make sure that "TCP" is setup in your Blackberry's configuration (Options -> Advanced Options -> TCP). The specific settings are provider-specific, so you'll have to search the web and/or ask your provider.

To test TCP connectivity, try LogicMail again, or try any other program that uses TCP (Opera Mini is a good one for this).

I'm running BlackBerry OS vX.Y, will LogicMail work with my unit?

All LogicMail releases will work on the mentioned version of BlackBerry OS, *or any higher version*.

I'm trying to send E-Mail, but I'm getting "Problem with Sender" errors.

You need to go into the global configuration screen ("*Config*" menu option), and set the "*Full name*" field.

Why does SMTP switch to port 465 when I enable SSL?

There are two common ways you can establish a network connection using SSL:

- Connect directly to a special port that expects SSL
- Connect to a non-SSL port, then send "STARTTLS" to *switch* to SSL mode

Typically, POP and IMAP servers use the first approach. (ports 110/143 vs. ports 993/995). The same approach is supported with SMTP (port 25 vs. port 465), but is less common to see. Due to platform API limitations, however, it is impossible for LogicMail to support the second approach. Therefore, if you want to use SMTP with SSL, you have to use an SMTP server that supports direct SSL connections.

And then some information about what you've implemented thus far

Features

This project in a developmental stage, so it may not have everything one would expect from a fully functional E-Mail client. However, it is already quite feature-complete for simply checking IMAP/POP folders and messages, and sending messages via SMTP.

Since people do want to know what my plans are for this project, what features will be implemented, and what releases I have in mind, I have constructed a project release/feature plan.

Here is a somewhat disorganized list of already implemented features:

- Support for BlackBerry OS v4.1 and v4.0
  - Yes, the v4.1 build will work with v4.2.
- Global settings
- SSL (do you seriously think I'd send my password in the clear?)
- Multiple accounts
- Status popups
- Talking to an IMAP and POP mail servers
- Folder tree retrieval (IMAP)
- Folder tree cache (IMAP)
- Mailbox listing (message flags shown on IMAP servers)
- Message display
  - MIME message structure decoding
  - Display of plain text message body parts (no HTML planned at this time)
  - Decoding and display of quoted-printable encoded plain text
  - Decoding and display of Base64 encoded Unicode plain text
  - Decoding and display of Base64 encoded image attachments
  - Display of message properties, to view extra header information
- Basic SMTP support
  - Composing and sending new messages
  - Replying and forwarding messages
  - Address book access for contacts
  - SSL support only for dedicated SSL ports (i.e. 465)
  - SMTP AUTH support for PLAIN, LOGIN, and CRAM-MD5

Then someone gets carried away when you least expect it:

From rivvah@XXXX.com Mon Mar 12 22:38:42 2007
Date: Mon, 12 Mar 2007 19:38:31 -0700
From: troy XXXX <rivvah@XXXX.com>
To: Derek Konigsberg <octo@logicprobe.org>
Subject: Re: Calm down! :-)

On 3/12/07, Derek Konigsberg <octo@logicprobe.org> wrote:
> I know you're eager to file bug reports on my uber-pre-alpha code
> (that I
> released so I wouldn't hang onto it for another month) :-) Some of the
> things you mentioned are legitimate bugs, and others are discovery of
> kludges
> that I haven't yet "done right".

Fair enough -- I use the patch/bug/feature trackers on SF.net a lot, so my bad if you didn't like that. To me it was just sort of a standard course of action, rather than putting things in email. Many developers prefer just to have everything laid out in the SF.net tools so it's there when they get to it.

I won't spam the systems with anything more until you say so. :)

> As far as releases are concerned, I think all may be happer if we do set a
> series of "milestones" between now (snapshots maquerading as "0.2") and
> the
> eventual goal of 1.0. Any thoughts on feature-sets to define a roadmap?

Hmmm. *ponder* OK the below format is letter is the major item, number is a sub-item to support it. These are mainly from the IMAP point of view, I don't even use POP anymore on any of my email accounts. I'm making this up as I type it and rearrange it.

Basic Needs
==========
A) Read and Compose email
1) configure "from" email address
2) very basic signature support (global per account?)
3) reverse displayed message list (per message) -- I think that the first line (in bold) should be the Subject, and the 2nd sub-line should be the Sender email address. The current way it is right now confuses me, it's unlike other mail programs. :)

B) Copy sent email to Sent folder (must-have before replying)
1) configure "Sent" folder (IMAP, picklist of existing folders?)

C) Reply-To email, Forward email (same code almost?)
1) ability to choose "Reply to all" or "Reply to sender only"
2) reply is copied to Sent folder (configurable?)

D) Delete email (that whole IMAP Expunge thing)
1) expunge support?
2) immediate delete support?

E) Save a Draft of email
1) configure "Drafts" folder (IMAP)
2) draft email deleted when Sent
3) ability to resume Draft

E) File email
1) Copy-To folder
2) Move-To folder

Enhanced Needs (post 1.0?)
==========
F) Set IMAP flags (keywords/labels), Display keywords/labels (font colours?)
1) ability to set each font colour for each keyword/label, 1-5

G) LDAP support? (corporate address book?)

How's that look for a general outline to start with? I tried to think logically of the things I do constantly and work them out in a logical progression of code features....

-te

--
some live, some die
in the way of the samurai

Hey, isn't this starting to look like a project plan?

While we're at it, what the heck?

LogicMail Feature Backlog

* = Implemented in latest snapshot

Features for 0.3: (First "release" with build numbers)

Sending mail
- Configure "From" address (*)
- Configure message "Signature" (*)
- Message reply capability (*)
- Message forward capability (*)
- Rudimentary "Sent" folder (IMAP only)

Connection management
- Login-time password prompts (*)
- Intelligent bad password handling (*)

Features for 0.4:

Generic Mailbox Interface (GMI)
- Design interface
- Develop implementations for IMAP and POP
- Develop J2ME RMS implementation for local folders

Sending mail
- Use of configurable "Sent" folder (local RMS only)
- Use of configurable "Drafts" folder (local RMS only)

Features for 0.5:

Decouple accounts from program state
- Main screen becomes a mail folder tree
- Global folder tree becomes usable

Implement copy/move operations for GMI

Support "Sent" and "Drafts" folder anywhere in any GMI folders

Features for 0.6:

Implement RMS-backed cache for on-line GMI folders (IMAP/POP)

...

...

Unallocated features:

Support for PIPELINING on POP commands

Support for asyncronous IMAP commands

Support for IMAP IDLE

Additional authentication types
- SMTP: DIGEST-MD5, NTLM
- IMAP: LOGIN, PLAIN, CRAM-MD5, DIGEST-MD5, NTLM
- POP: LOGIN, PLAIN, CRAM-MD5, DIGEST-MD5, NTLM

Adding message sender addresses to BB address book (may require signed code)

Features requiring signed code
- GMI proxy for BlackBerry mail folders
- ActiveAutoText (highlighted links) with hooks to the rest of the BB
  - Dialing of phone numbers
  - Opening of website URLs
- Calendar integration for meeting invite E-Mails

LogicMail for BlackBerry

Overview

LogicMail is a J2ME E-Mail client supporting IMAP and POP, and designed to run on RIM BlackBerry handheld devices. Its goal is to provide an alternative to the service-oriented "push" E-Mail system that the device normally provides, so we can also use our conventional E-Mail services. The inspiration for this project came from the now defunct Mail4ME project, as well as some of the initial protocol code. Currently in the very early stages of development, this project aims to grow into a versitile piece of software.

The big focus in this client is on usable IMAP support, which seems to be an afterthought in some other clients. Why IMAP first? Well, two reasons: First, I use IMAP personally. Second, the IMAP protocol allows a lot of optimization for constrained latency and bandwidth that POP does not provide. As such, I suspect the IMAP implementation will always feel much faster than the POP one. However, be assured that I have implemented POP support too, since I know it is important to a lot of potential users.

Downloads

Right now you can find the source code in SVN, if you dig through the Sourceforge project page. I'm using a BSD-style license for this project, except for the reused code from Mail4ME. Those parts are under the Enhydra Public License, which is pretty close to the BSD license. I eventually plan to rewrite most of this code, and have rewritten a lot, but for now I'm still using it do take care of a few tedious bits of low-level socket functionality.

If you want to try it on your BlackBerry handheld, the easiest way is to point your mobile browser at the "Over-The-Air" installation link.

Over-The-Air Installation
Sourceforge download page
(last updated on 6/23/2007)

Frequently Asked Questions

SourceForge Project Page

[Features and issues] [Project plan] [Screenshots]

Questions? Comments? Write me at octo@logicprobe.org

But the site still looks ugly, is rarely updated, and SourceForge isn't helping

- They provide ticketing and forums
- They provide repository hosting and file releases
- But they don't provide much in the way of usable project management tools

And the things you need to keep track of keeps increasing...

The Trac Project

Trac is an enhanced wiki and issue tracking system for software development projects. Trac uses a minimalistic approach to web-based software project management. Our mission is to help developers write great software while staying out of the way. Trac should impose as little as possible on a team's established development process and policies.

It provides an interface to Subversion, an integrated Wiki and convenient reporting facilities.

Trac allows wiki markup in issue descriptions and commit messages, creating links and seamless references between bugs, tasks, changesets, files and wiki pages. A timeline shows all project events in order, making the acquisition of an overview of the project and tracking progress very easy.

http://trac.edgewall.org/

Ok, so what is it?

- Its a F/OSS project itself, written in Python
- I know people who work on it, and they won't shut up about it!
- It has tons of plugins!
- Its designed to do exactly what I need!

So I migrate everything I can off of SourceForge and into Trac...

Overview

LogicMail is a J2ME E-Mail client supporting IMAP and POP, and designed to run on RIM BlackBerry handheld devices. Its goal is to provide an alternative to the service-oriented "push" E-Mail system that the device normally provides, so we can also use our conventional E-Mail services. The inspiration for this project came from the now defunct Mail4ME project, as well as some of the initial protocol code.

The big focus in this client is on usable IMAP support, which seems to be an afterthought in some other clients. Why IMAP first? Well, two reasons: First, I use IMAP personally. Second, the IMAP protocol allows a lot of optimization for constrained latency and bandwidth that POP does not provide. As such, I suspect the IMAP implementation will always feel much faster than the POP one. However, be assured that I have implemented POP support too, since I know it is important to a lot of potential users.

Downloading

Right now you can find the source code in SVN, if you dig through the SourceForge project page. I'm using a BSD-style license for this project, except where noted otherwise.

If you want to try it on your BlackBerry handheld, the easiest way is to point your mobile browser at the "Over-The-Air" installation link:

- Latest release: 0.4.0
  - Over-The-Air Installation - http://logicmail.sf.net/ota
  - SourceForge Download Page
- Development snapshot builds (for BlackBerry OS v4.1 or higher) from CruiseControl:
  - Over-The-Air Installation - Latest build
  - ZIP archive - Latest build (for the Application Loader)
  - OTA installable files - Navigate to the JAD files for installation
  - ZIP archives
  - Status: Automated build execution is temporarily disabled due to build process retooling.

Resources

- Frequently Asked Questions
- Documentation
- Debugging
- Features and Issues
- Design and Development
- Screenshots
- SourceForge Project Page

Roadmap

Show already completed milestones

Update

Milestone: 0.3

Completed 9 months ago

100%

Closed tickets: 12 *Active tickets:* 0

Transition LogicMail from a series of development snapshots into versioned releases.

Milestone: 0.3.x

Completed 2 months ago

100%

Closed tickets: 8 *Active tickets:* 0

This is a milestone for tracking bug fixes to the 0.3.x baseline.

Milestone: 0.4

Completed 2 months ago

100%

Closed tickets: 17 *Active tickets:* 0

This is an interim milestone aimed at implementing logging and debugging features, as well as changes to user configuration necessary to support future versions. It will also include features that did not make it into 0.3, if they can be cleanly implemented without major structural changes to the code. In support of the user configuration changes, this milestone will also include a complete reworking of the persistent storage code used for both user configurations and data caching.

Now the pace is really picking up!

Now if only I could get the latest code out faster...

- The source repository is openly available, right?
  - Building from source is not always practical for users
  - I don't feel like cutting a "release" very often
  - Saying a bug is "fixed in the repository" is meaningless to a user

Continuous Integration with CruiseControl

- Automation of the whole build process
- Its now easy for users to download a binary of the latest and greatest code
- I'm now forced to make sure everything I commit actually compiles
- And, there is a Trac plugin!

cruisecontrol

http://cruisecontrol.sourceforge.net/

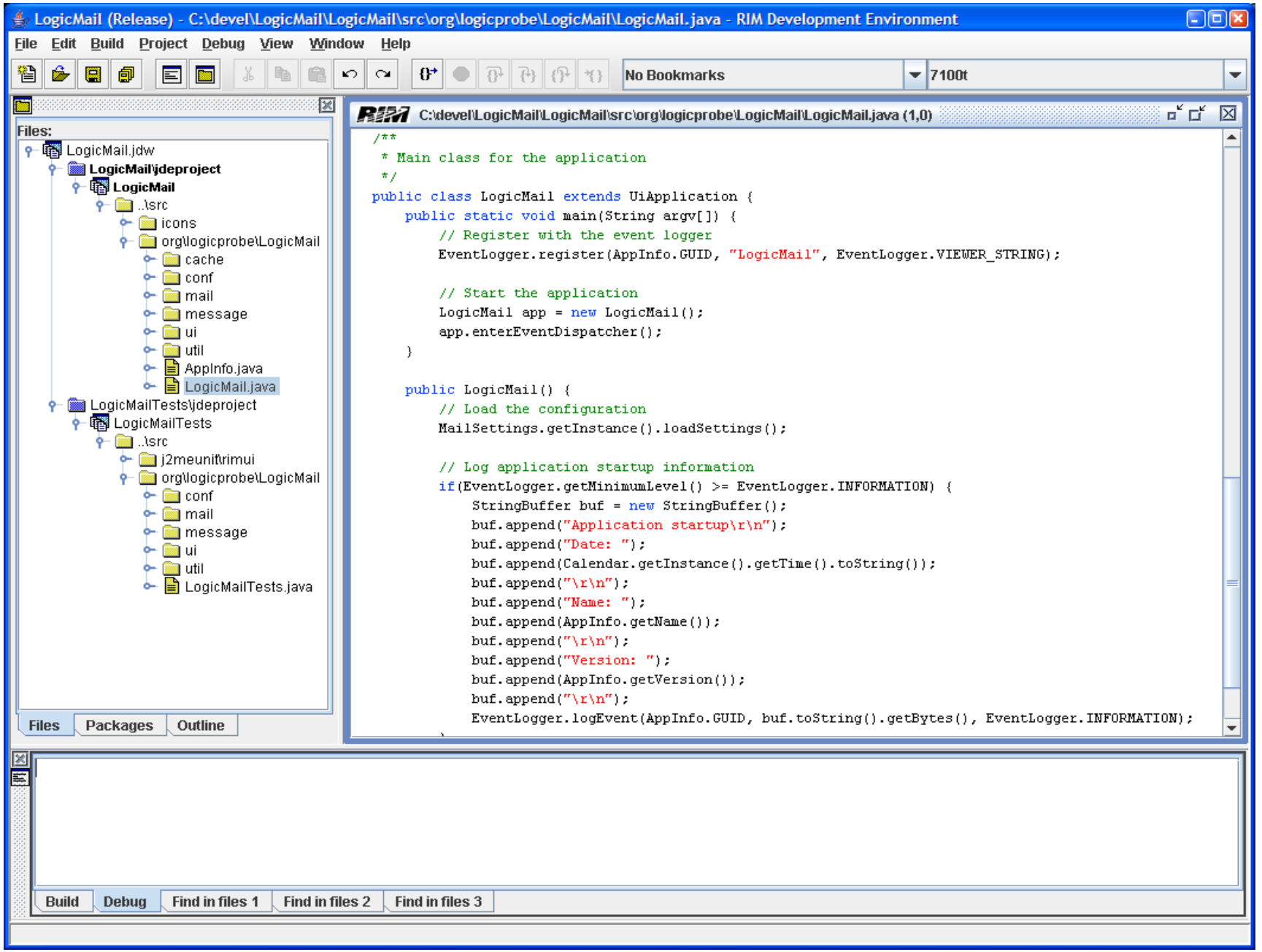
Hey, this is starting to feel like a real project!

- Users can directly submit bug reports and feature requests as tickets
- I then allocate these tickets to various project milestones
- Web content is easily improved and updated
- The userbase continues to expand

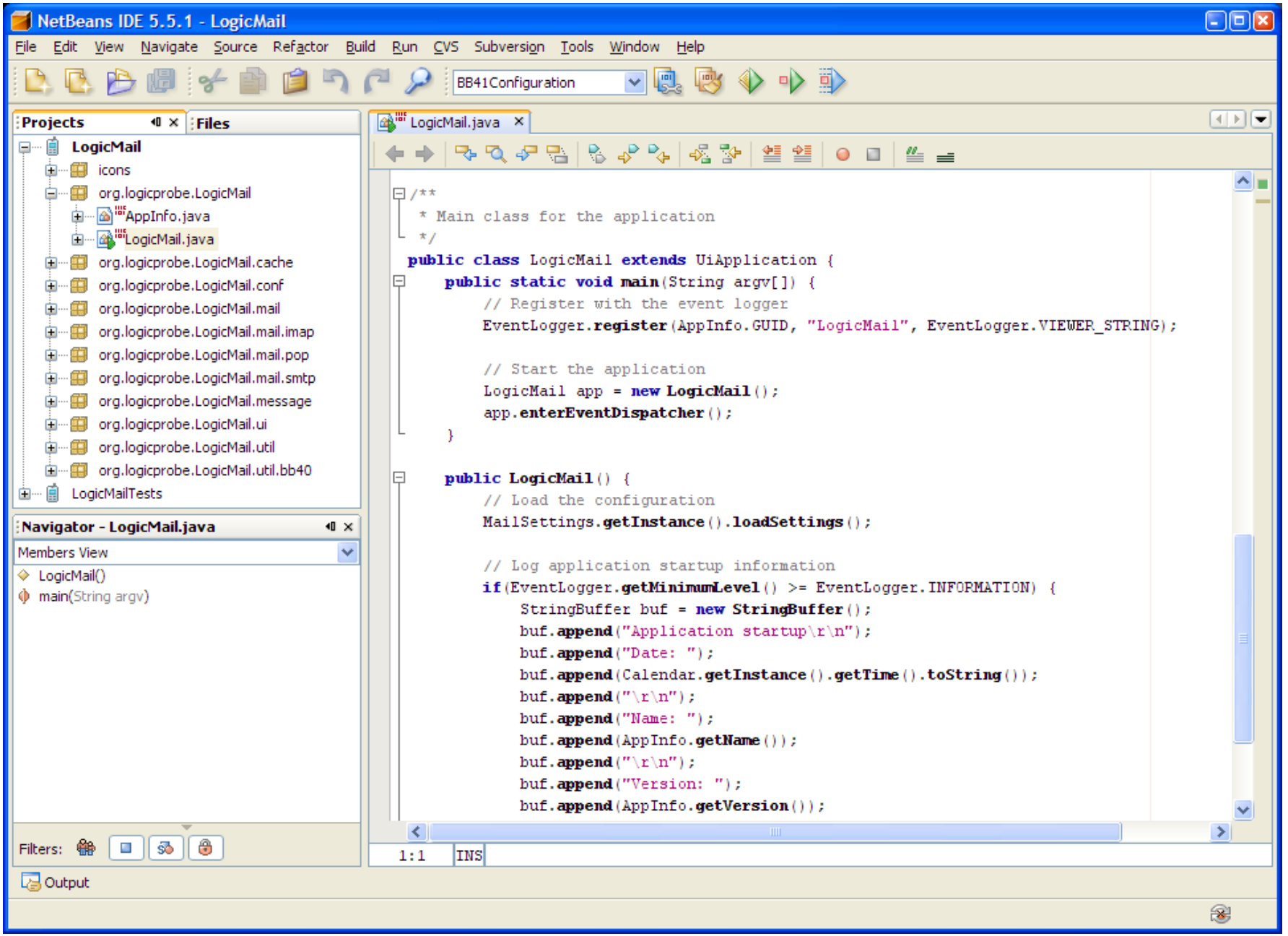
Hurdles Along the Way

- Development environment
- Getting good debugging information
- Simple-sounding feature requests
- Restricted device APIs
- Finding contributing developers

Vendor-supplied IDE

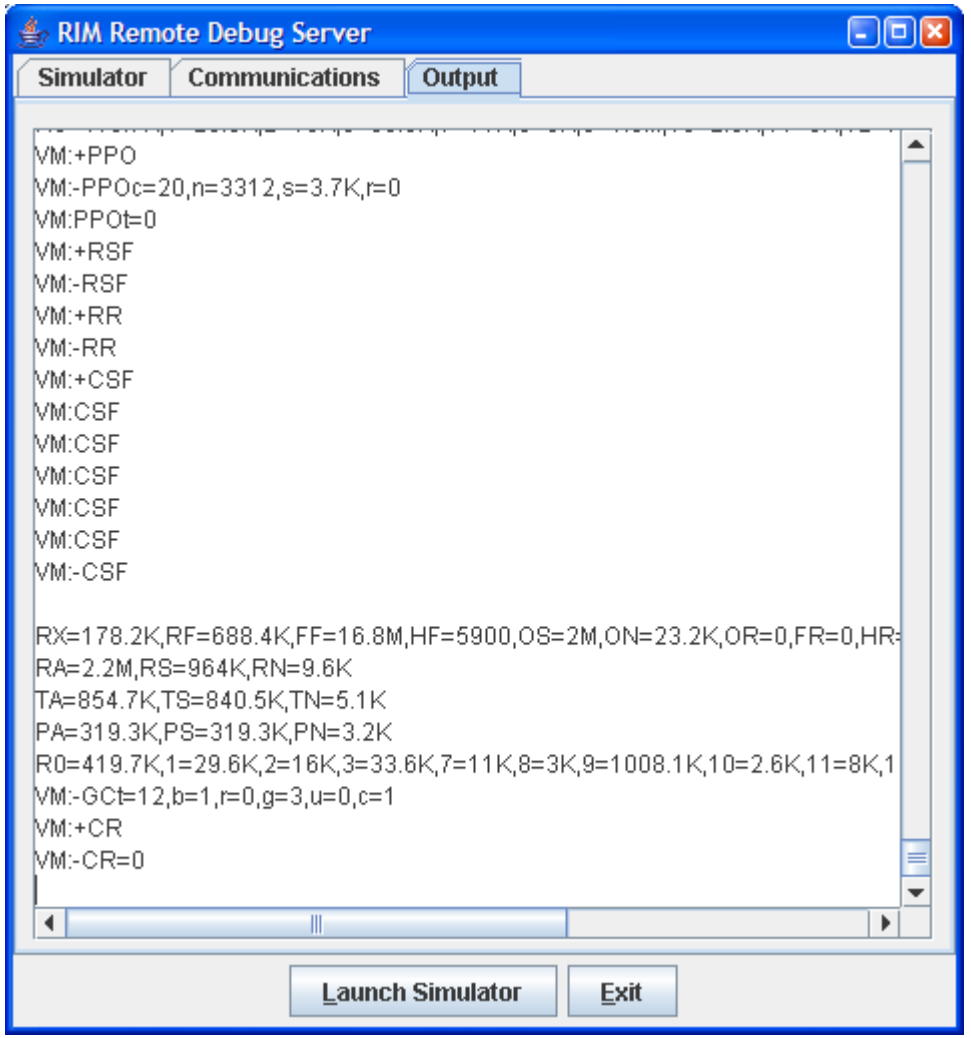


NetBeans as an alternative



Running in a simulator

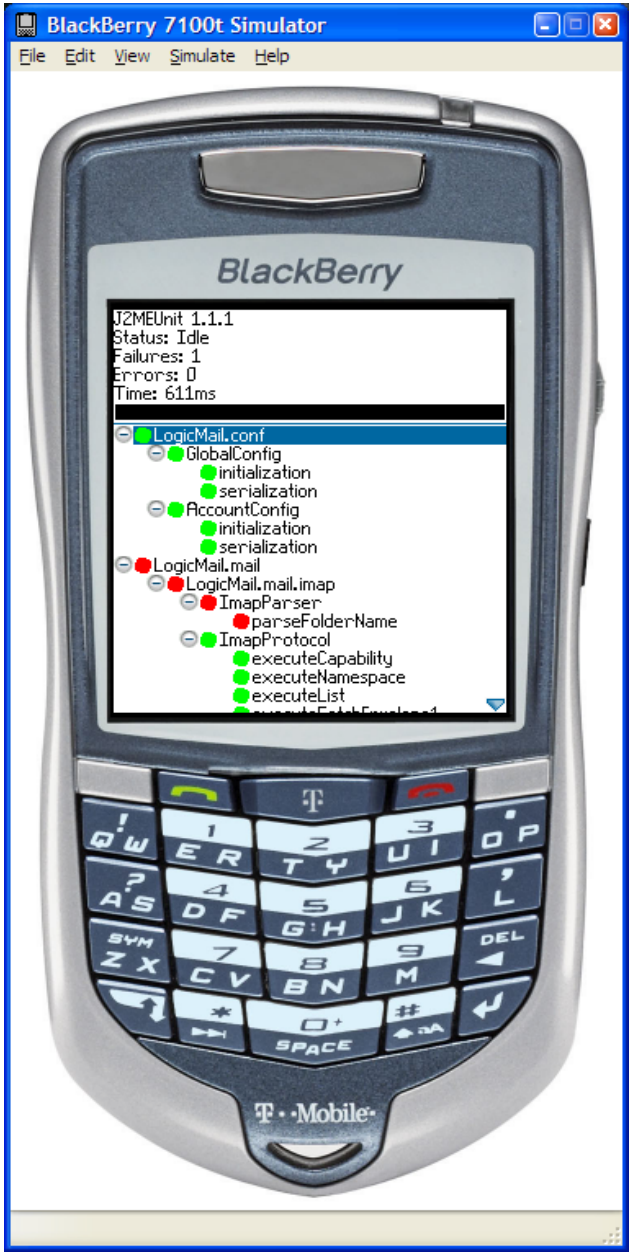
Console output goes here:



App runs here:

Unit testing is very important

- Why?
  - Application depends heavily on external inputs to function
  - Lots of complex data handling code is involved
  - Thorough testing full-up is very difficult, especially for finding bugs
  - Always useful for regression testing
- How?
  - Used the J2MEUnit framework
  - Wrote a custom UI for running tests
  - Developed test cases as I went along
  - Used bug reports to drive additional test cases



Debugging User Issues

- I can't test with every mail server out there
- Users will have issues I never encounter
- It is difficult to capture useful debugging information, especially in user-friendly ways

The Simple Feature Request

- “I'd really like it if your application could do XYZ”
  - Sometimes it is easy to just squeeze in
  - Sometimes you have to draw the line, if you actually want to reach a release milestone
  - Sometimes, you can't implement it due to technical limitations
  - And once in a while, a simple-sounding feature may take a redesign of the whole application to implement properly

Restricted APIs

- Using some parts of the BlackBerry API require your binaries to be digitally signed
- Some commonly-requested features cannot be implemented without the signed APIs
- The code signing key costs money
- Regardless of price, this causes a dilemma for an open-source project
- I've avoided these APIs thus far, but needing to use them will be inevitable

Finding Developers

- Open-source projects need users who can also become developers
- Not many developers are BlackBerry users
- Simple, yet hard-to-find, developer criteria:
  - Must be a decent Java developer
  - Must be willing to develop on Windows
  - Must be a BlackBerry user

The Path Forward

- Continue supporting the project
- Continue developing the project
- Continue interacting with the users
- Do the major redesign to better support those “simple feature requests”

Questions?