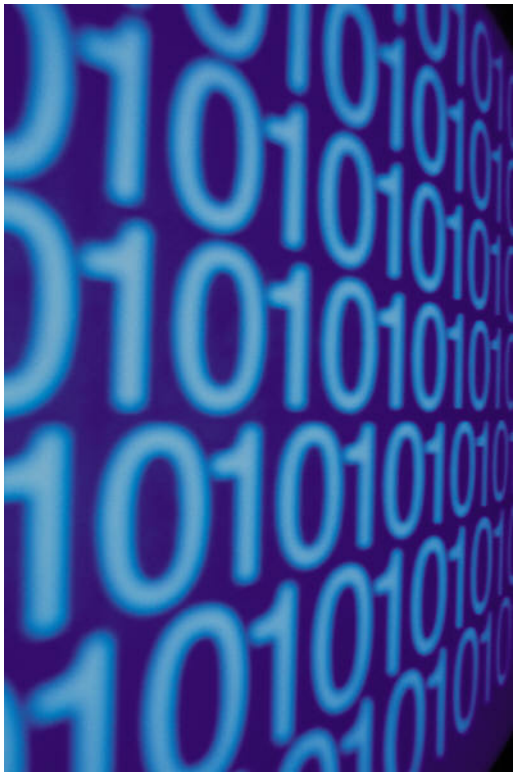
WRITING PLUGINS WITH RABBITMQ

Noah Gift and Michael Vierling

Noah's Background

- Film
- Commercial Data Center Automation Software
- Los Angeles, Atlanta, New Zealand, San Francisco
- I like programming in ANY language that is innovative

How & Why

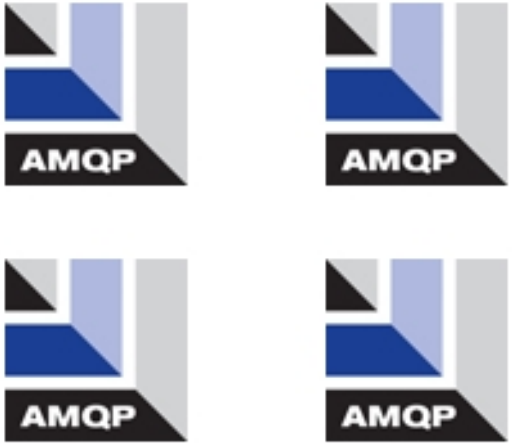


- Why AMQP?
- Why Erlang?
- Why OTP?
- Why RabbitMQ?
- How – Machinery

Why AMQP

- Industry Message Standard
- Cisco, Microsoft, Red Hat, VMWare, JPMorgan, Chase, Goldman Sachs
- Binary wire protocal

Why AMQP: Continued



- Key component in Distributed Architecture
- Enables Multi-OS, Multi-Language
- Reliable, Transactional, Flow-Control, Fault-tolerant

Downsides of AMQP

- AMQP 1.0 is almost final
- Support resources
- Upgrade glitches
- Monitoring issues
- Investment in the future

Why Erlang?

- Functional Language
- Immutable Data
- Scalable
- Reliable, Fault-tolerant

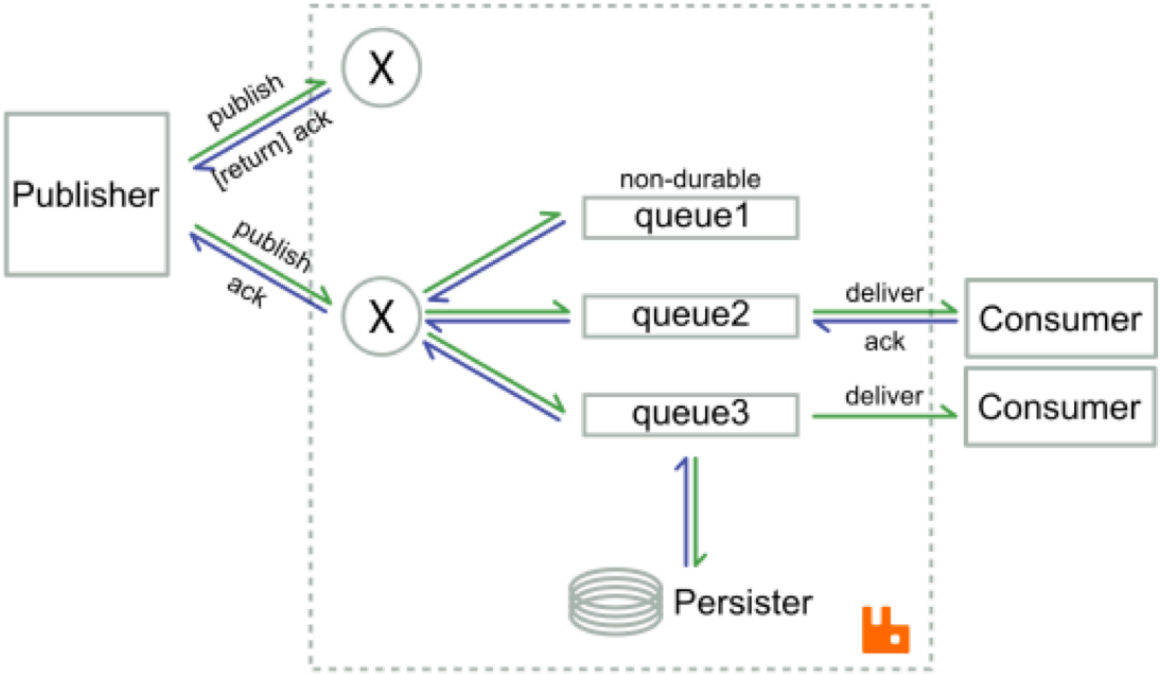
Why RabbitMQ

- Built with Erlang
- 10K messages per second – transient mode
- 3-5K messages per second – persistent mode

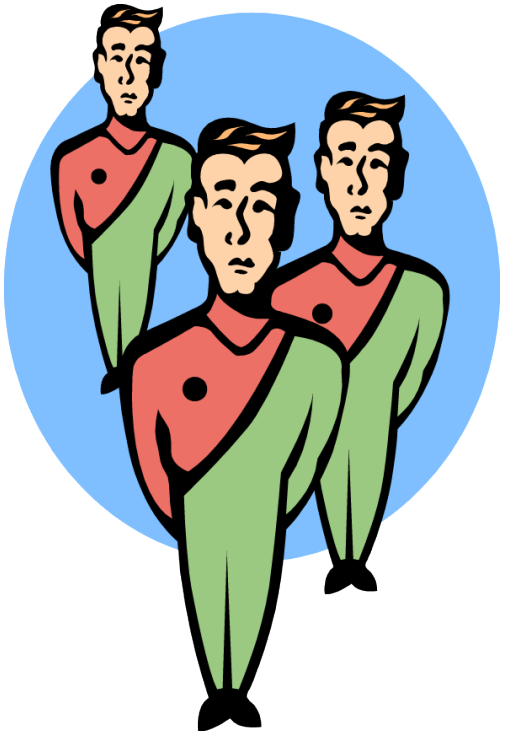
Why RabbitMQ

- Transactions
- Open Source
- Maintained by VMWare

Broker Infrastructure



Broker Infrastructure

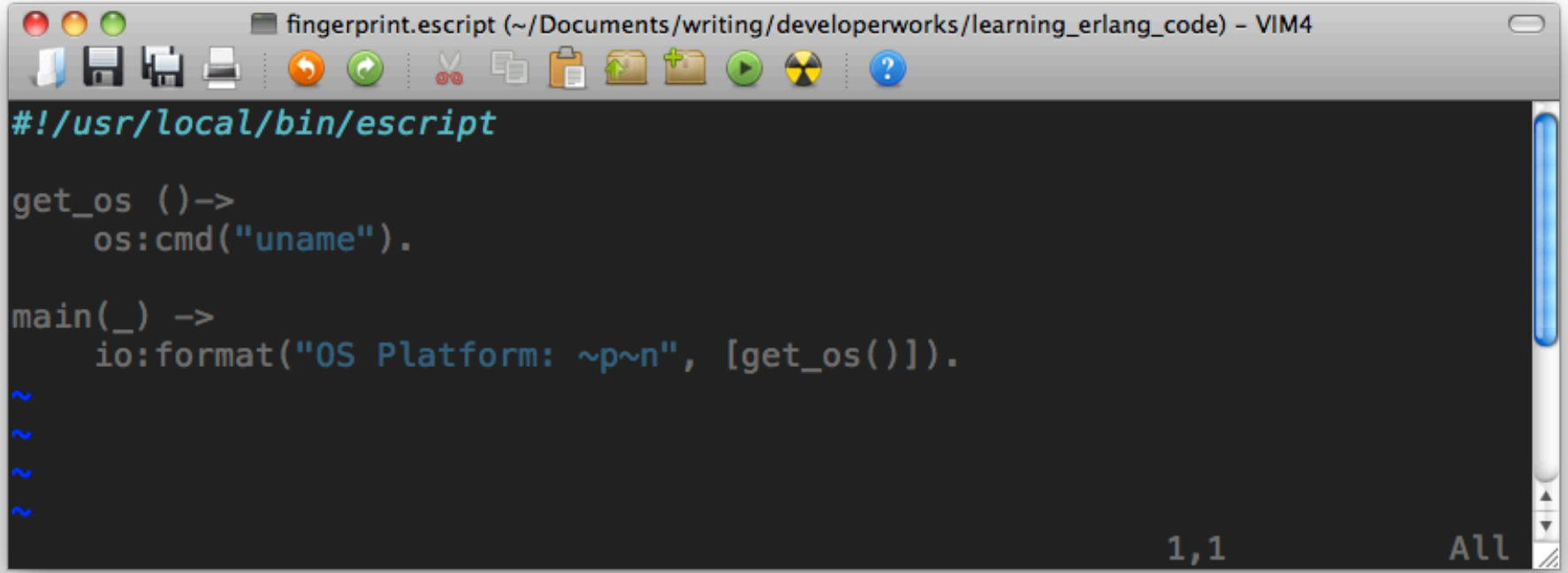


- Repeatable Install
- Reliable storage
- Redundant hardware

Broker Infrastructure: Continued

- Cluster of nodes
- Disaster Recovery
- Logging (such as Splunk) and monitoring
- Operations & Support

Broker Infrastructure: Use Escript



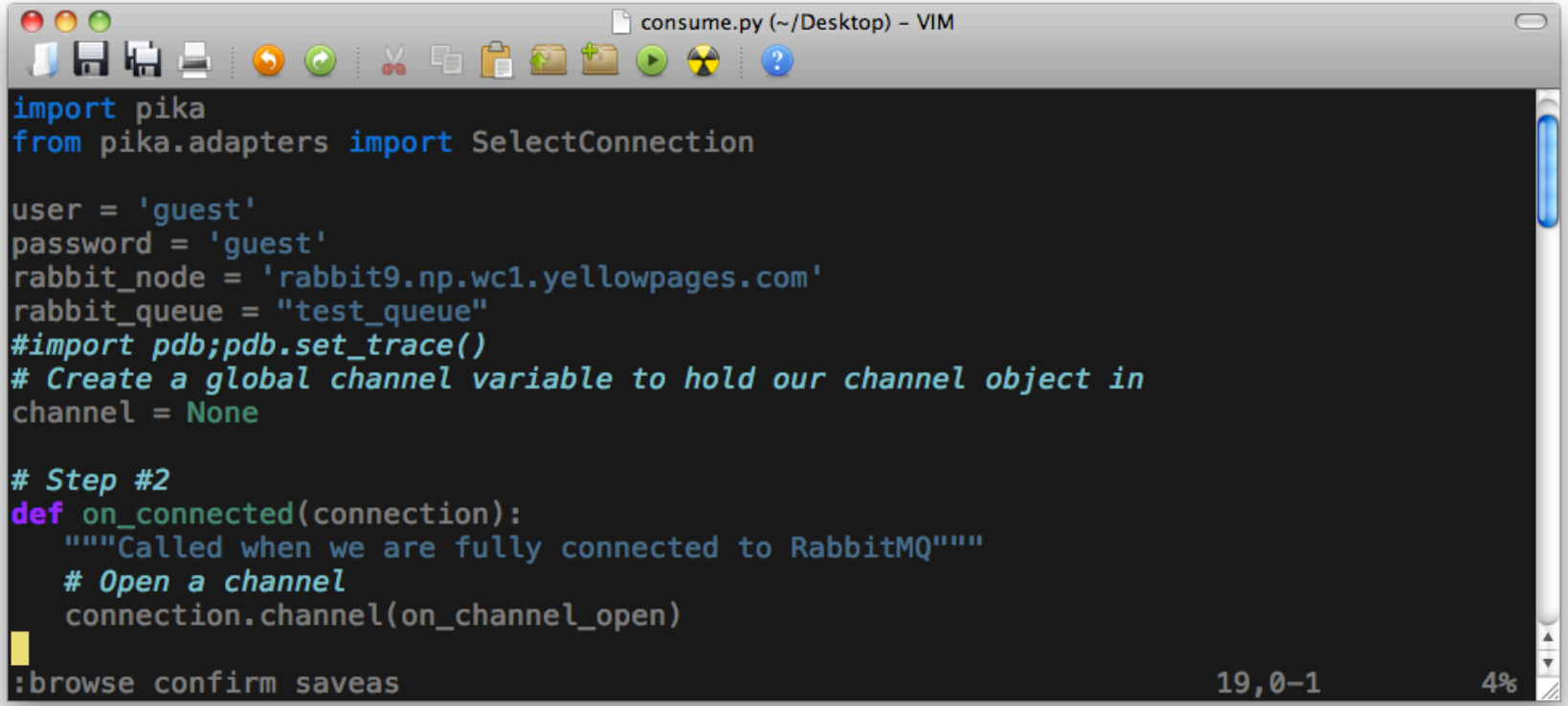
Broker Infrastructure Future?

- Integrate Rebar
- Continuous Integration Testing
- QuickCheck

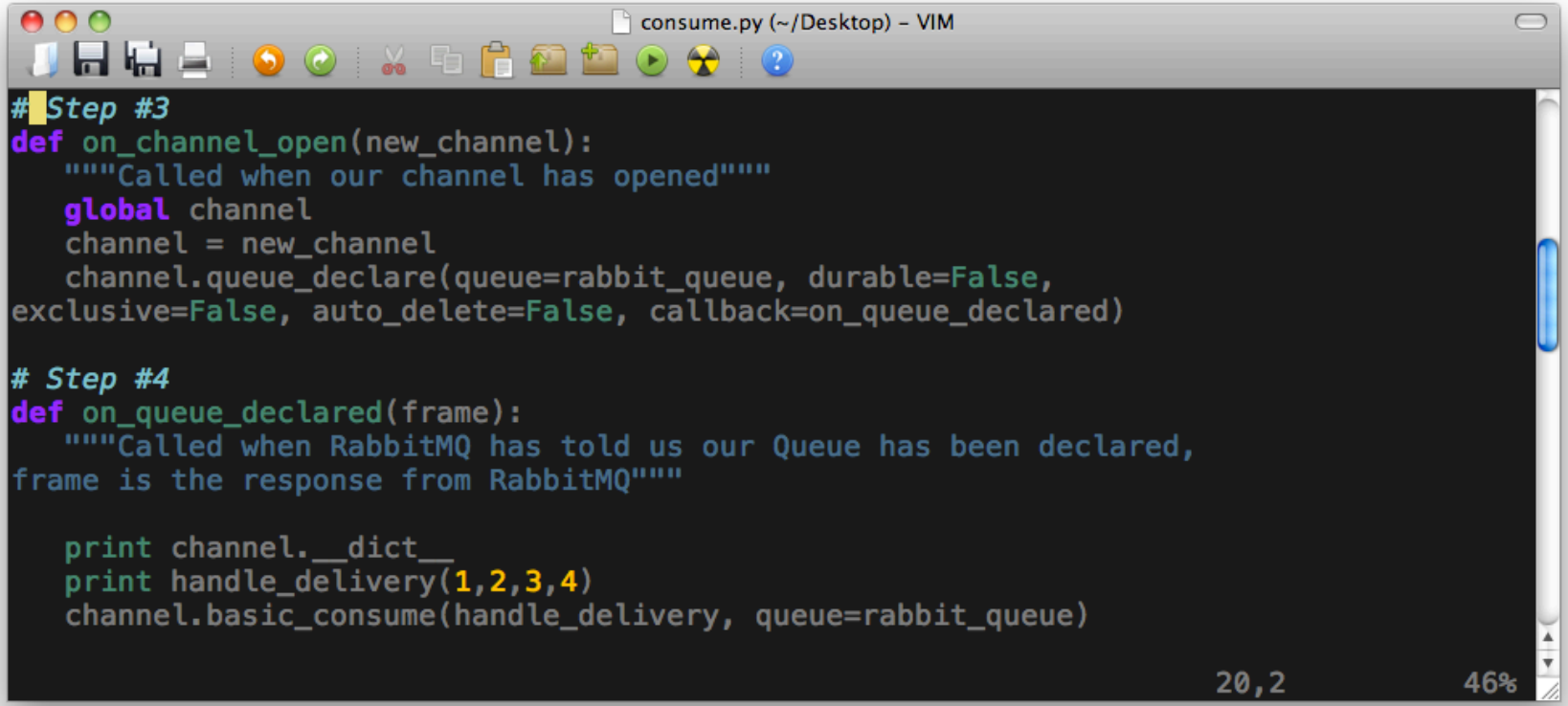
RabbitMQ Clients

- Any AMQP Clients (Apache)
- Ruby
- Python
- C#
- F# (Wrap the C# dll)

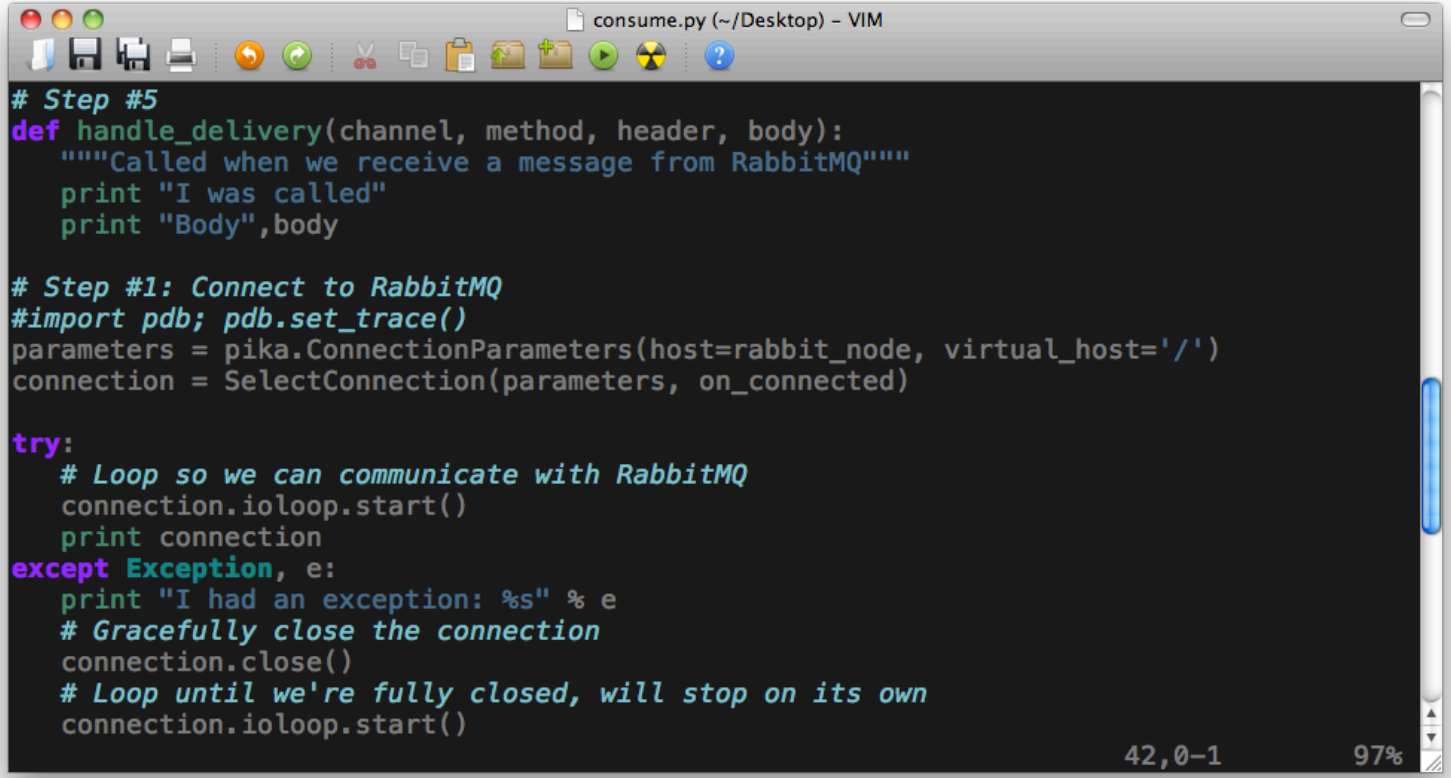
Consume.py Part: A



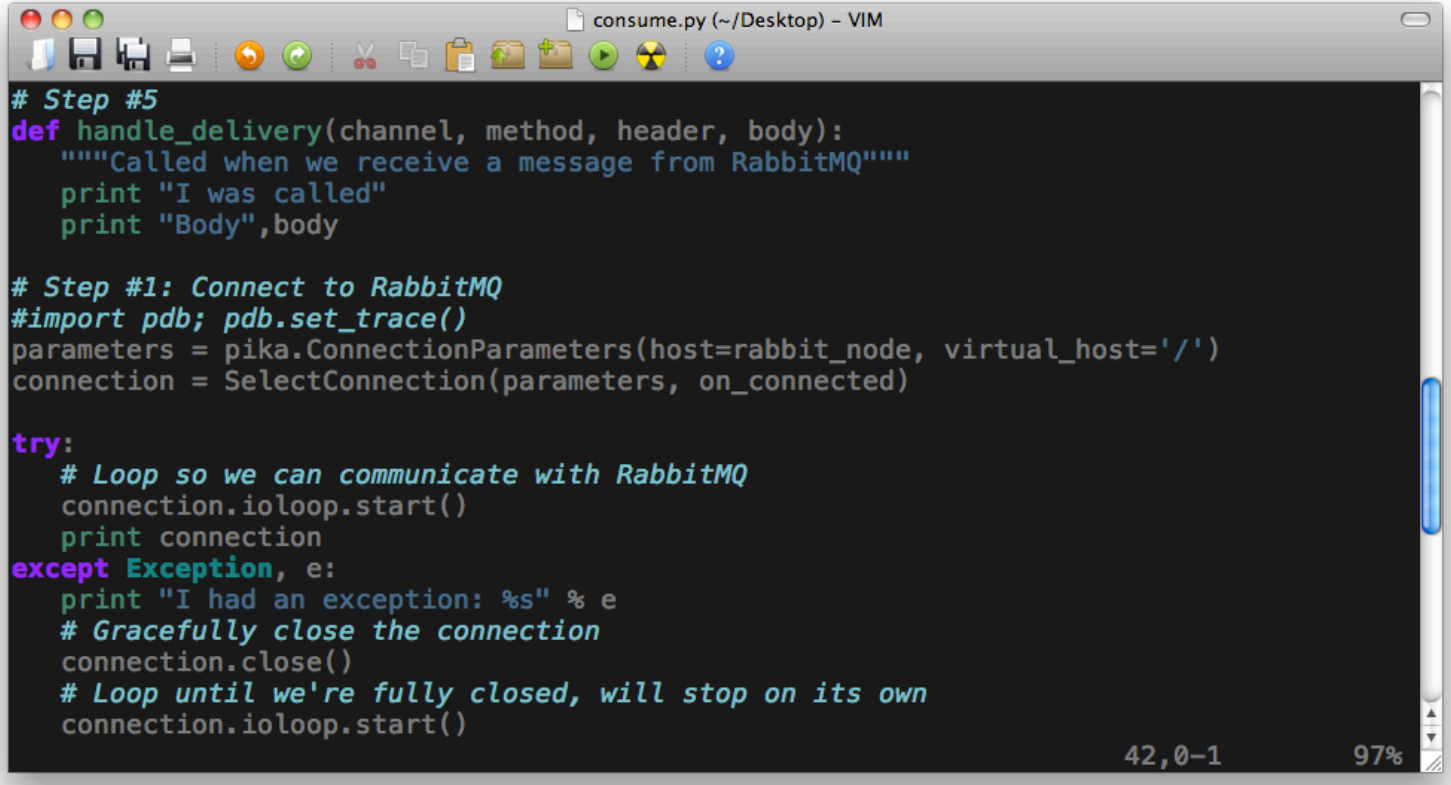
Consume.py Part: B



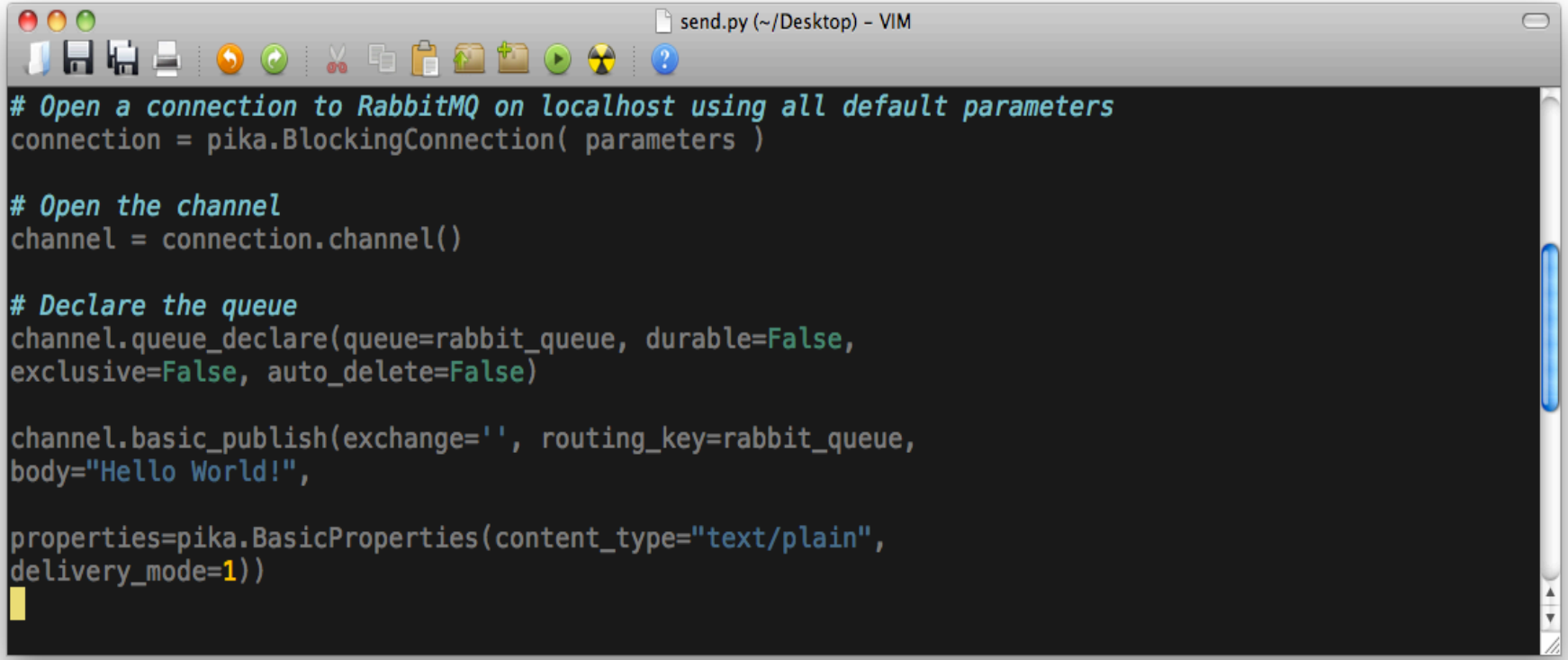
Consume.py Part: C



Send.py Part: A



Send.py Part: B



Writing RabbitMQ Plugin: WHY

- Access internal RabbitMQ functionality
- Running in same Erlang VM as broker can increase performance
- Leverage your existing infrastructure

Writing RabbitMQ Plugin: WHY NOT

- Developers don't know Erlang
- Poorly written plugin could crash broker
- You could lock yourself to internal API

Stable RabbitMQ Plugin

- Rabbitmq-management
  (replaced Alice)
- Rabbitmq-shovel
- Rabbitmq-stomp
- Rabbitmq-erlang-client

Experimental RabbitMQ Plugin Ideas

- Rabbitmq-auth-backend-ldap
- Rabbitmq-auth-mechanism-ssl
- Rabbitmq-jsonrpc-channel
- Rabbitmq-xmpp

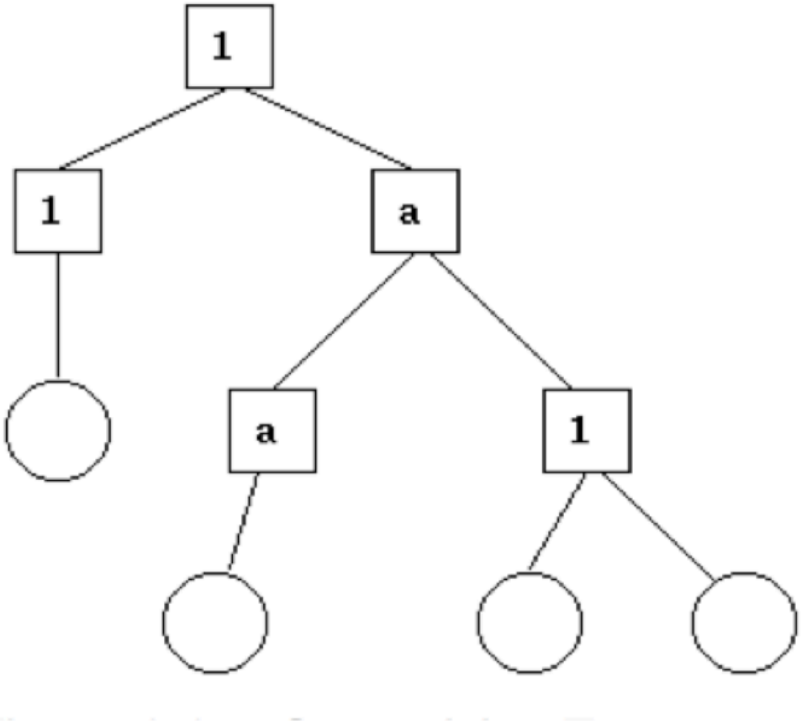
Crazy RabbitMQ Plugin Ideas

- Distributed Data Structure
- Artificial Intelligence
- Data Mining
- Create a protocol on top
- Event Processing

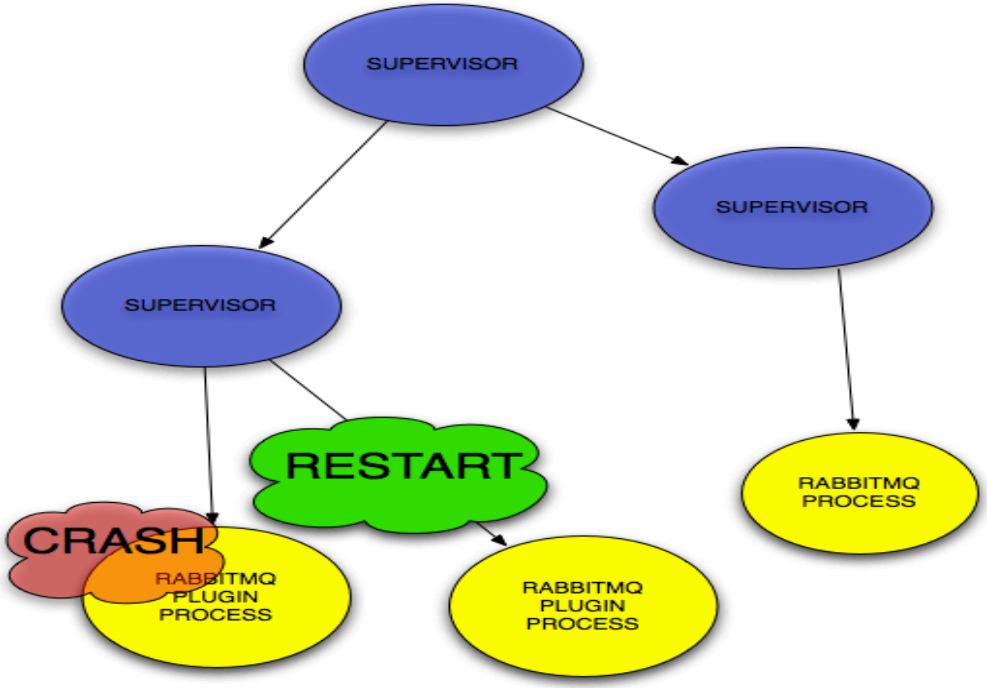
Michael Background

- Apple 5 Years
- Ascend/Lucent 4 Years
- Various Startups

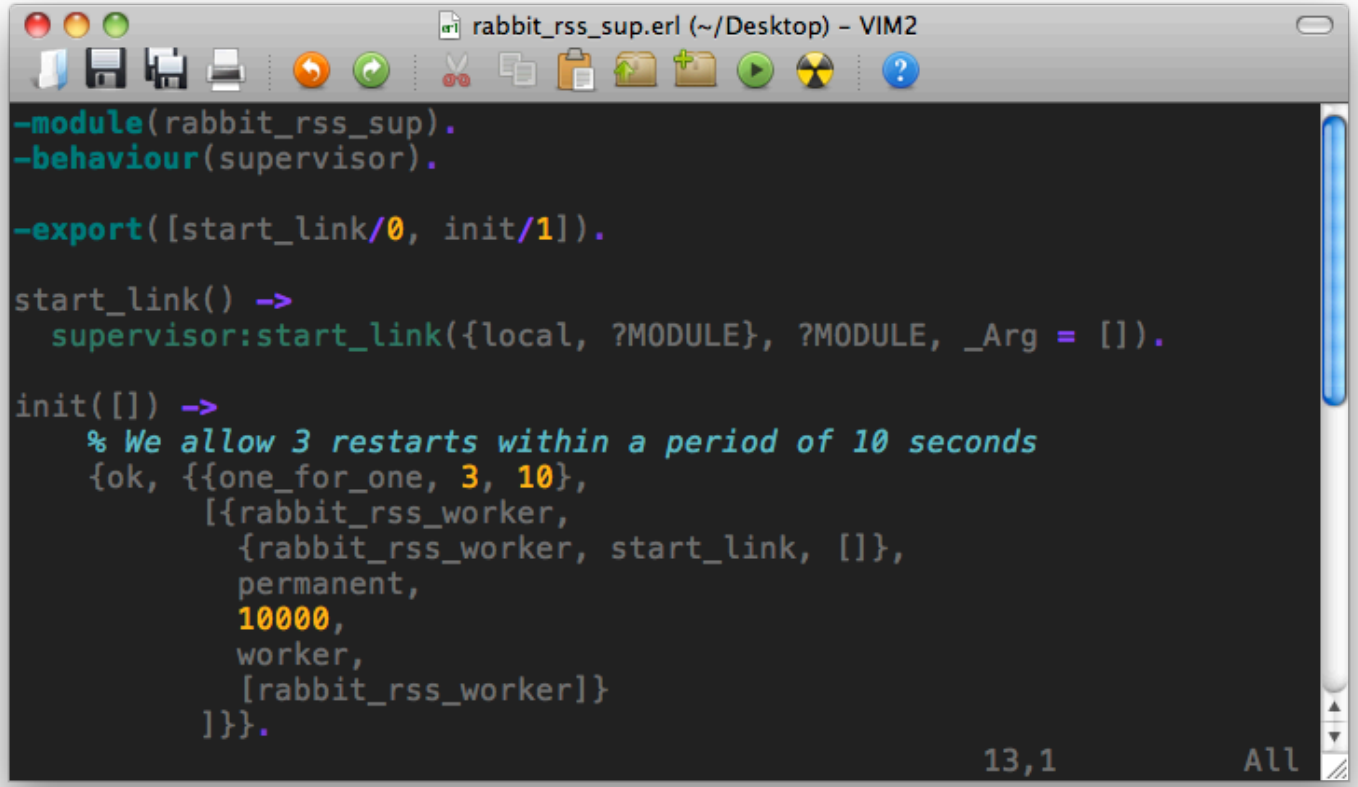
Supervision Tree



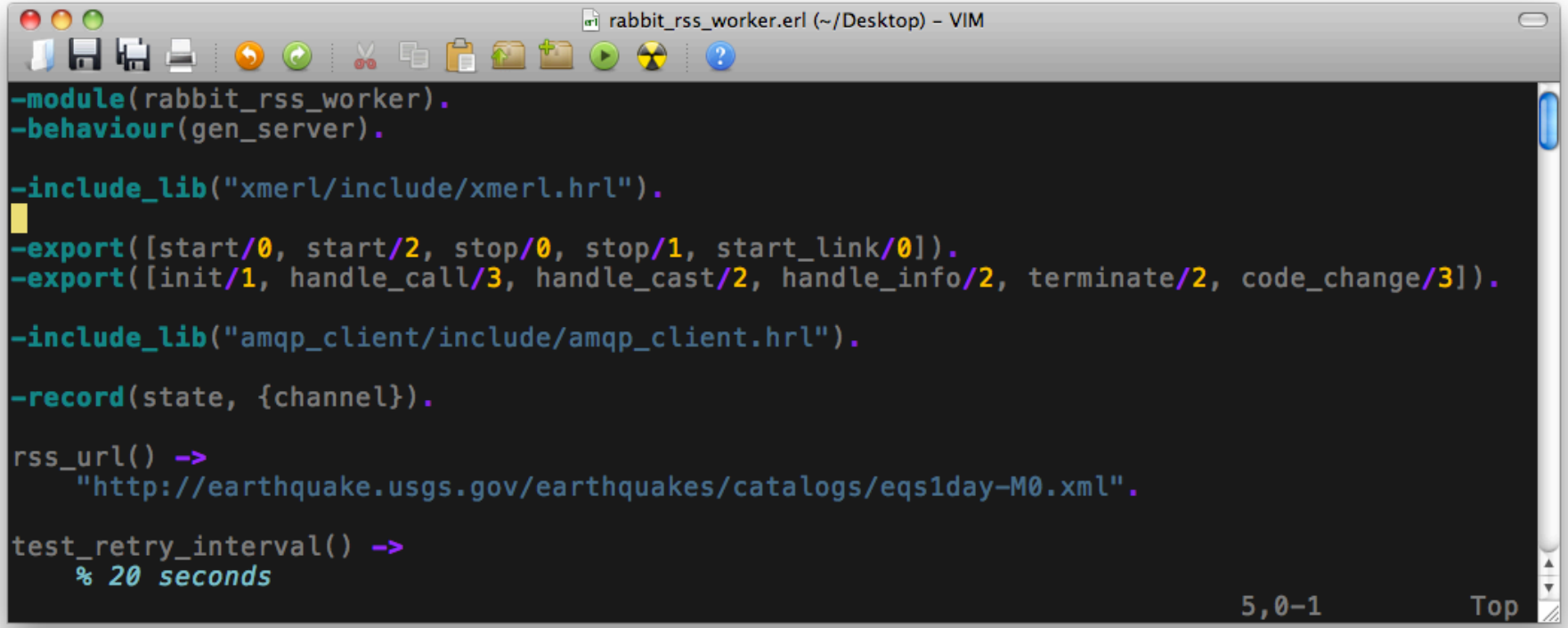
Plugin Supervisor



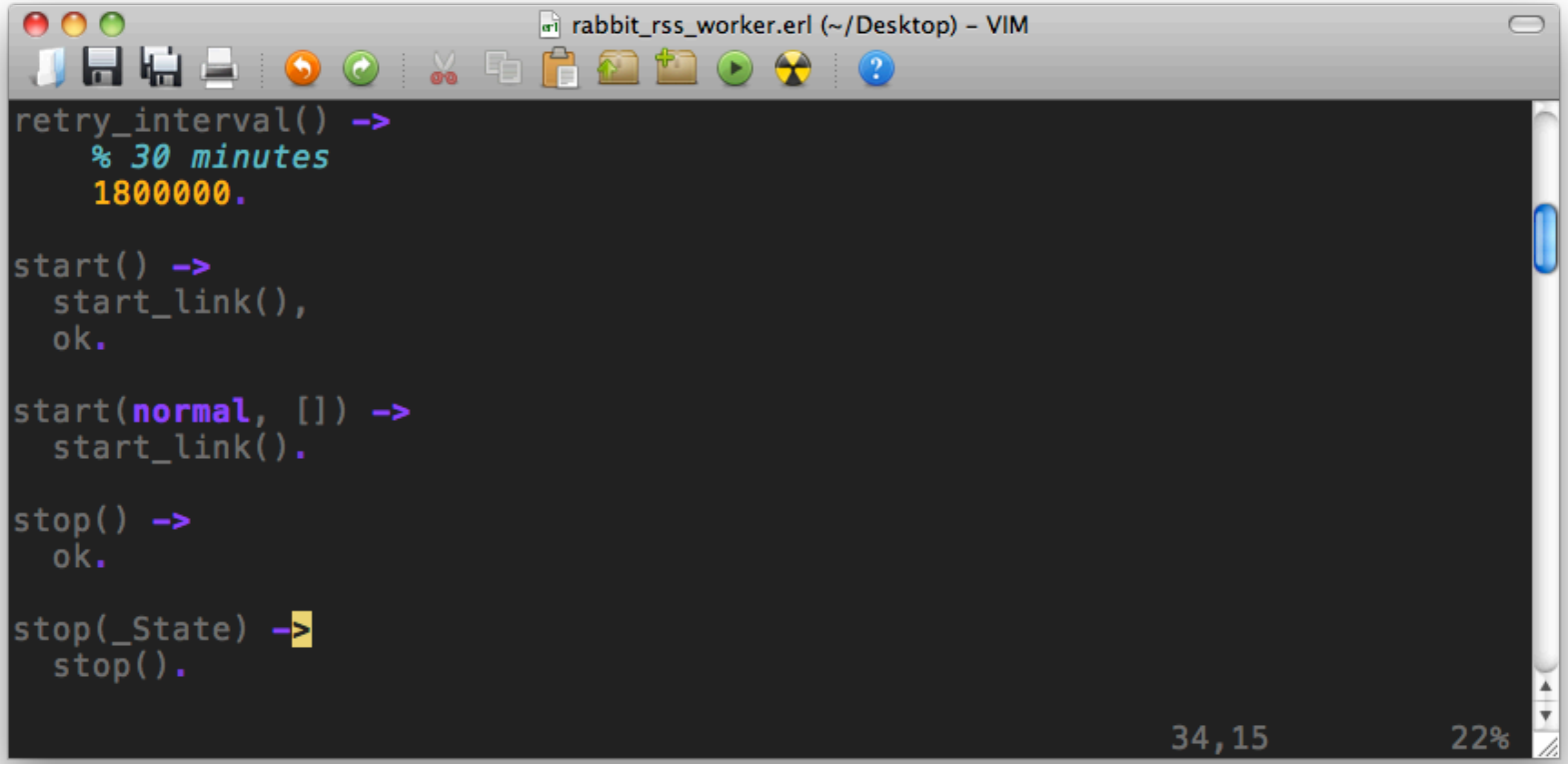
Rabbit RSS Supervisor



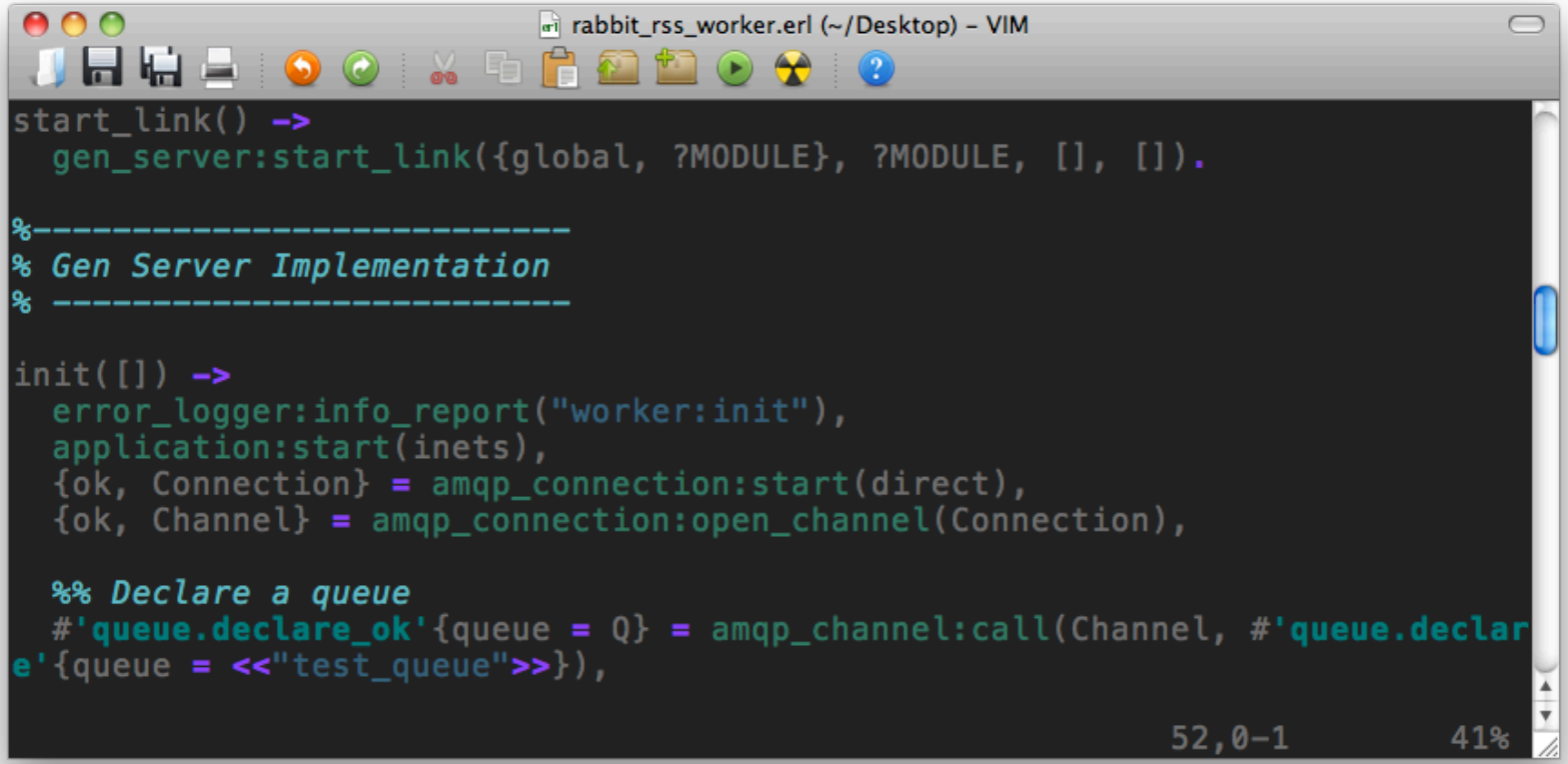
Rabbit RSS Worker: Slide A



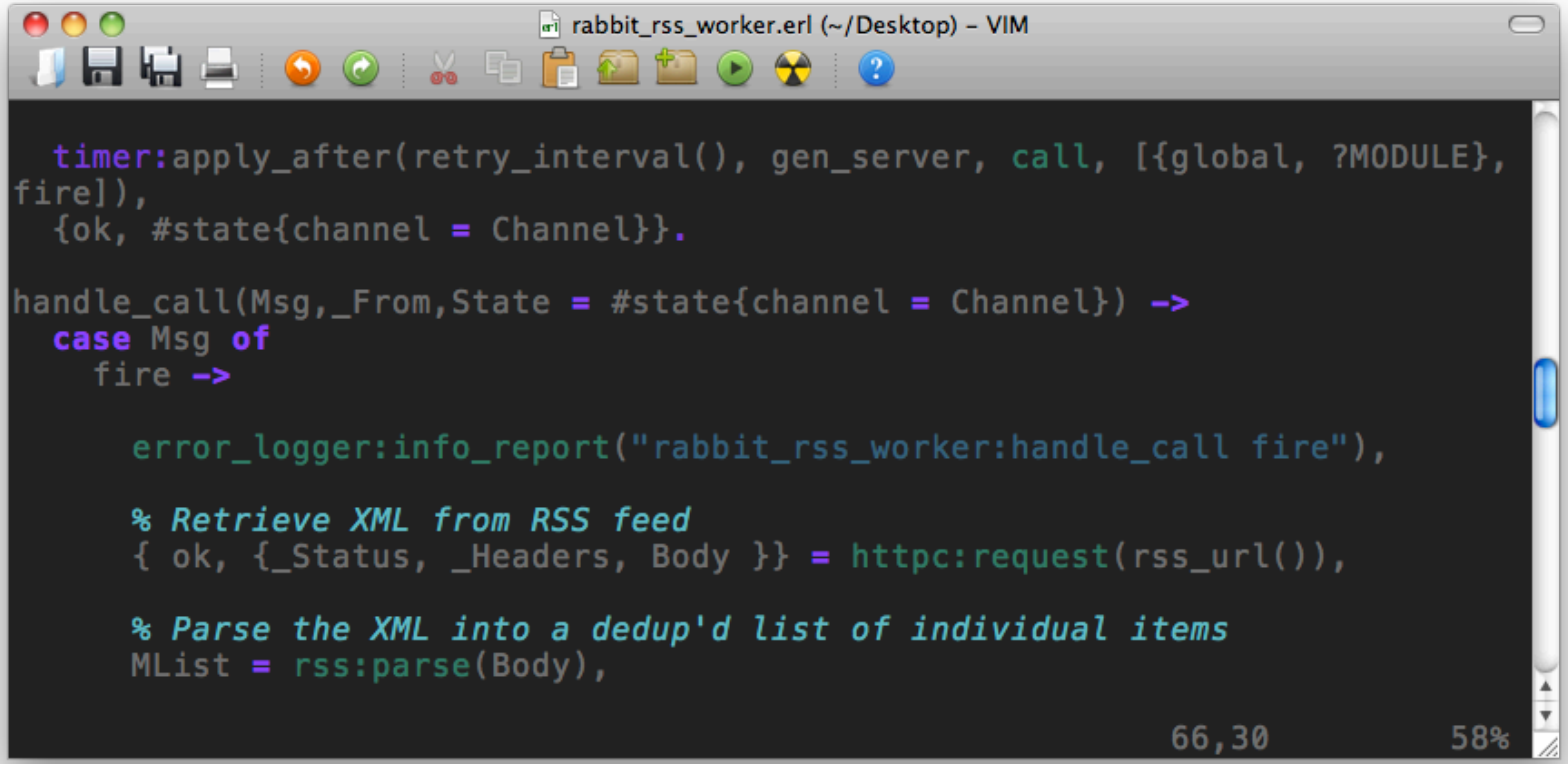
Rabbit RSS Worker: Slide B



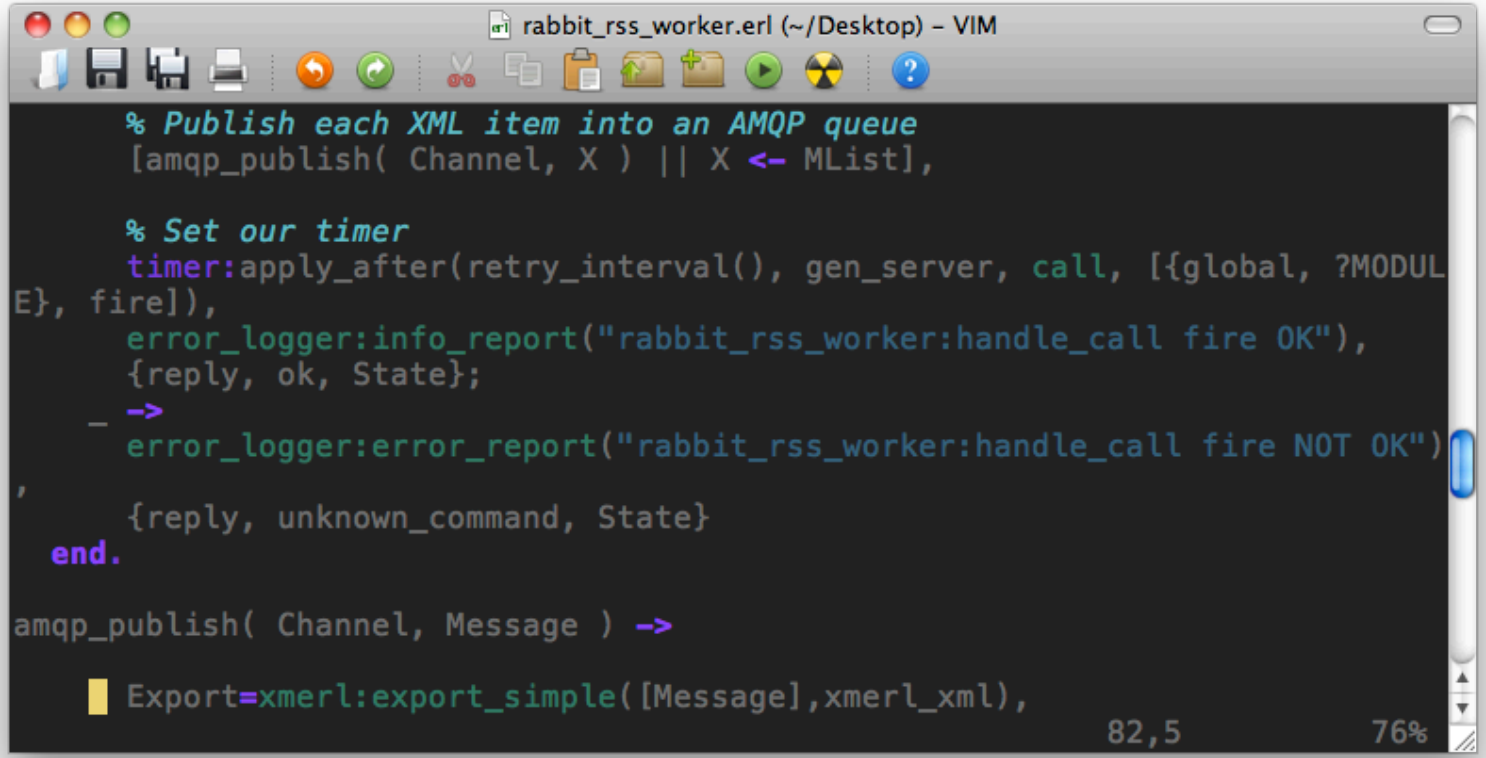
Rabbit RSS Worker: Slide C



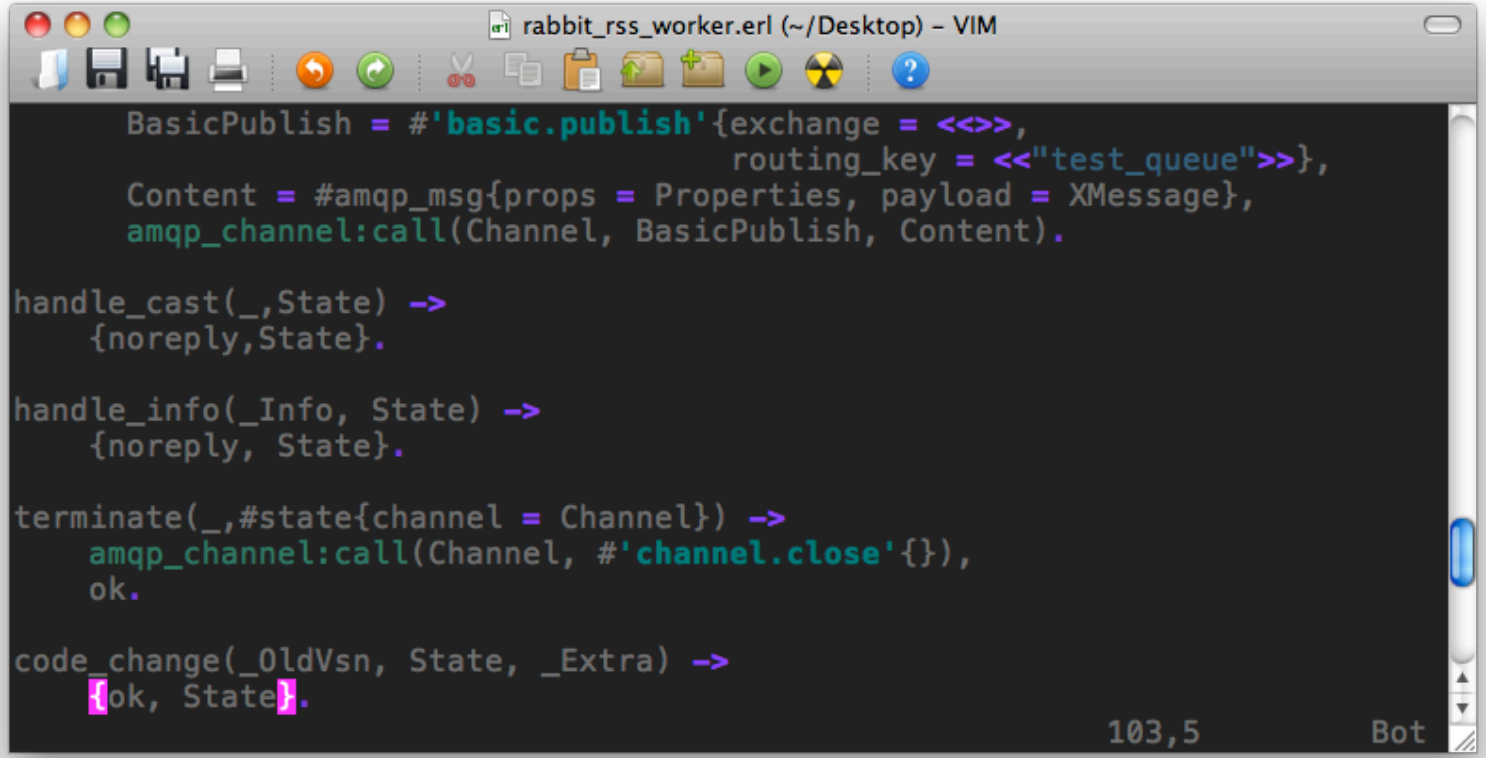
Rabbit RSS Worker: Slide D



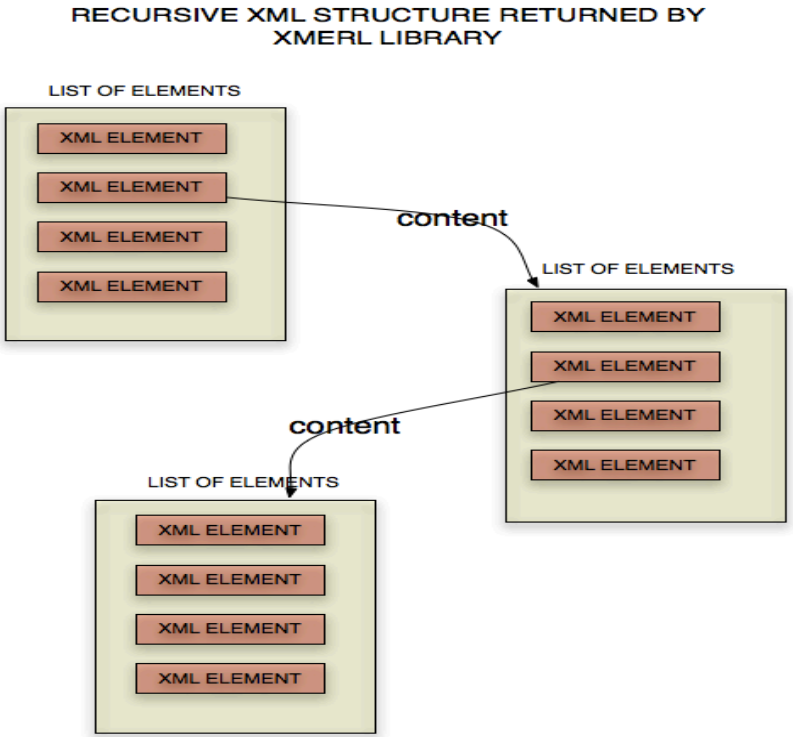
Rabbit RSS Worker: Slide E



Rabbit RSS Worker: Slide F

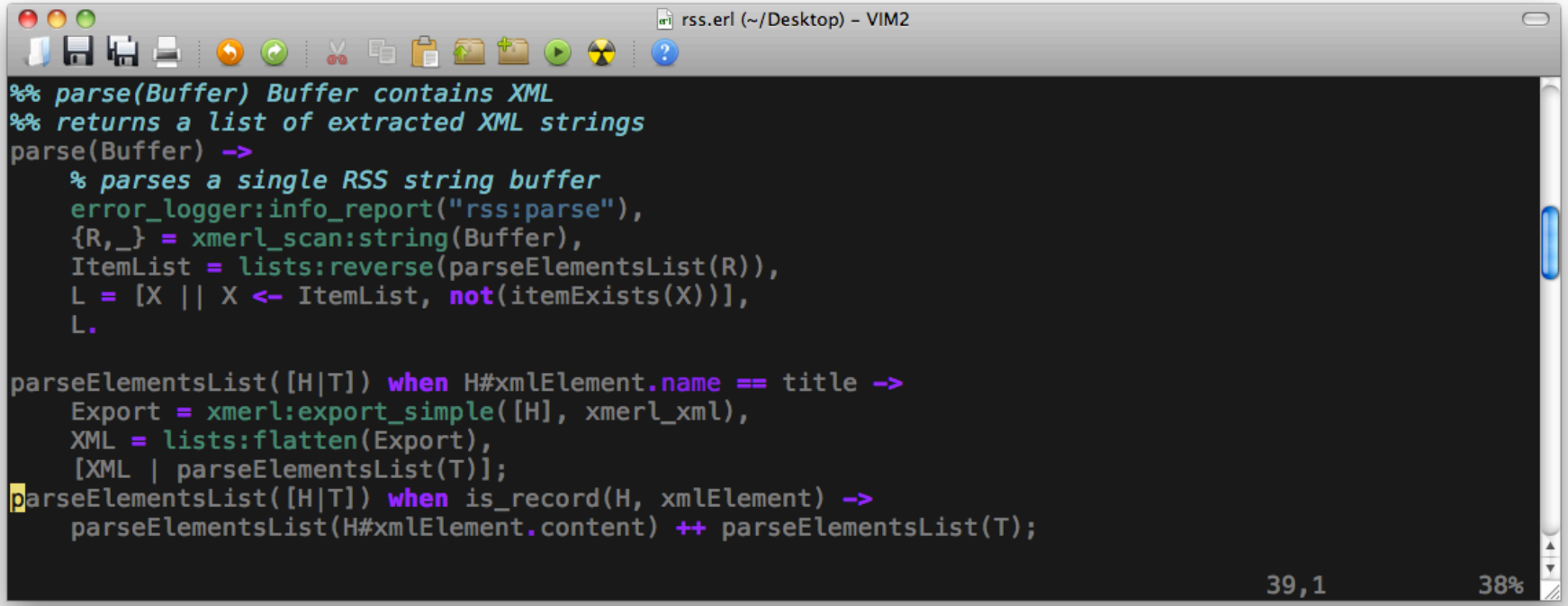


XML Parsing

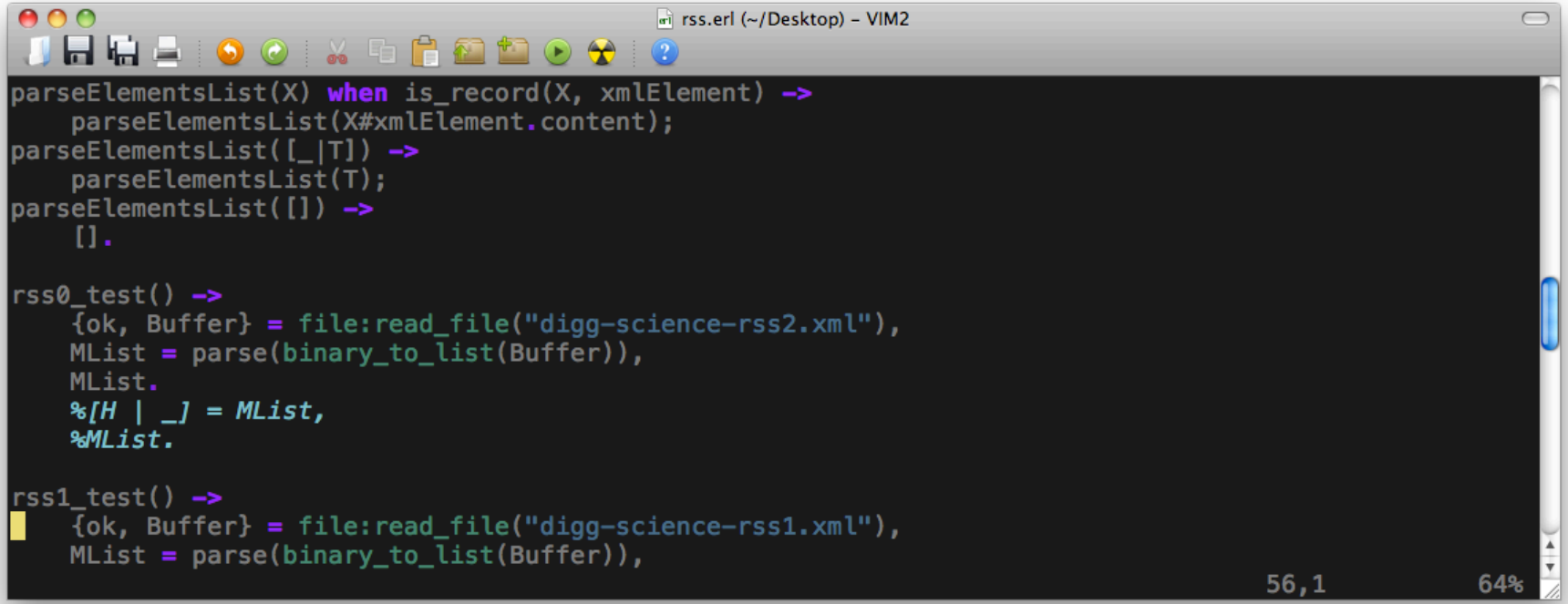

RECURSIVE XML STRUCTURE RETURNED BY XMERL LIBRARY
LIST OF ELEMENTS
XML ELEMENT
XML ELEMENT
XML ELEMENT
XML ELEMENT
content
LIST OF ELEMENTS
XML ELEMENT
XML ELEMENT
XML ELEMENT
XML ELEMENT
content
LIST OF ELEMENTS
XML ELEMENT
XML ELEMENT
XML ELEMENT
XML ELEMENT

RSS.ERL: Slide A

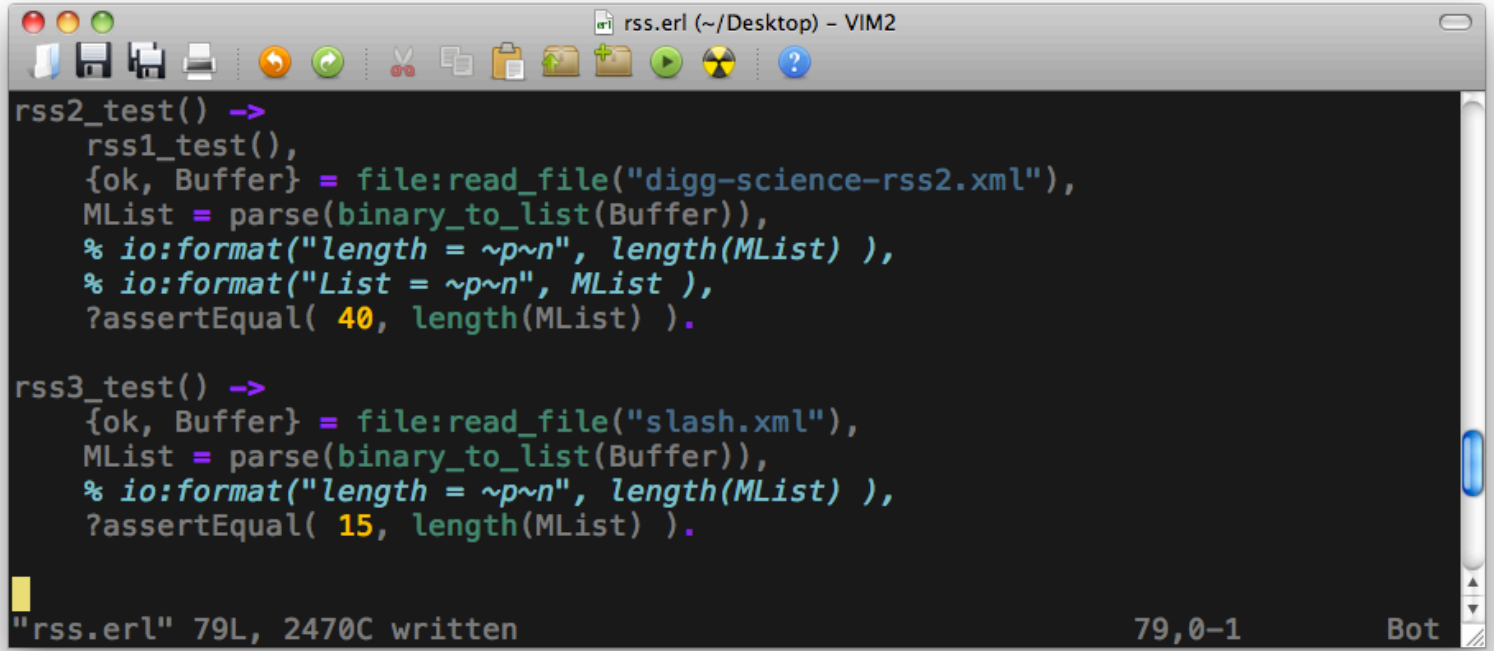
RSS.ERL: Slide B



RSS.ERL: Slide C



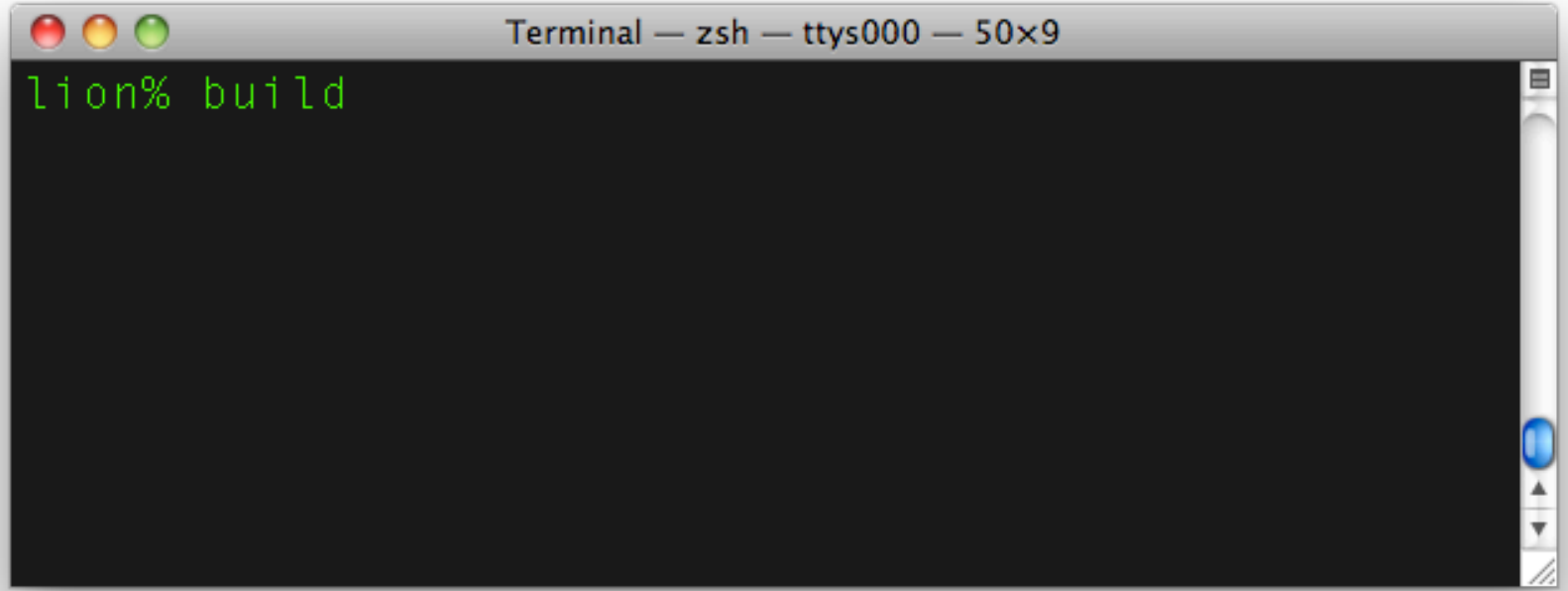
RSS.ERL: Slide D



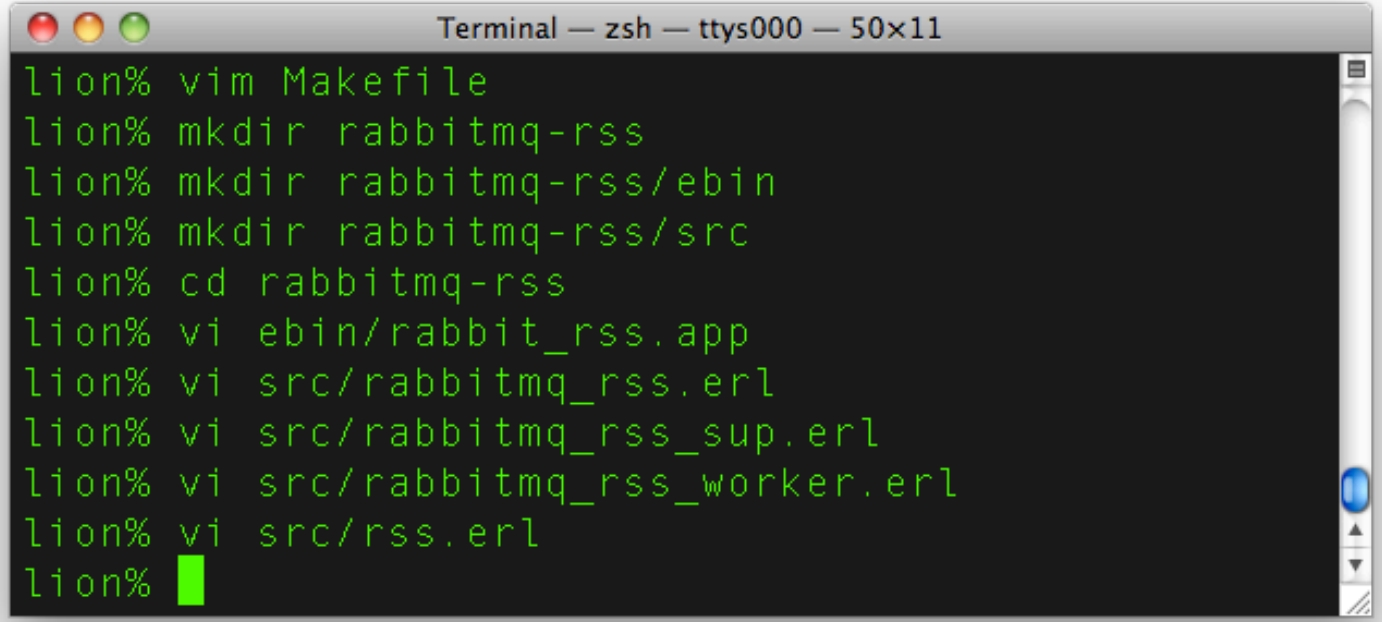
Checkout source: HG clone

Make: Be Patient….

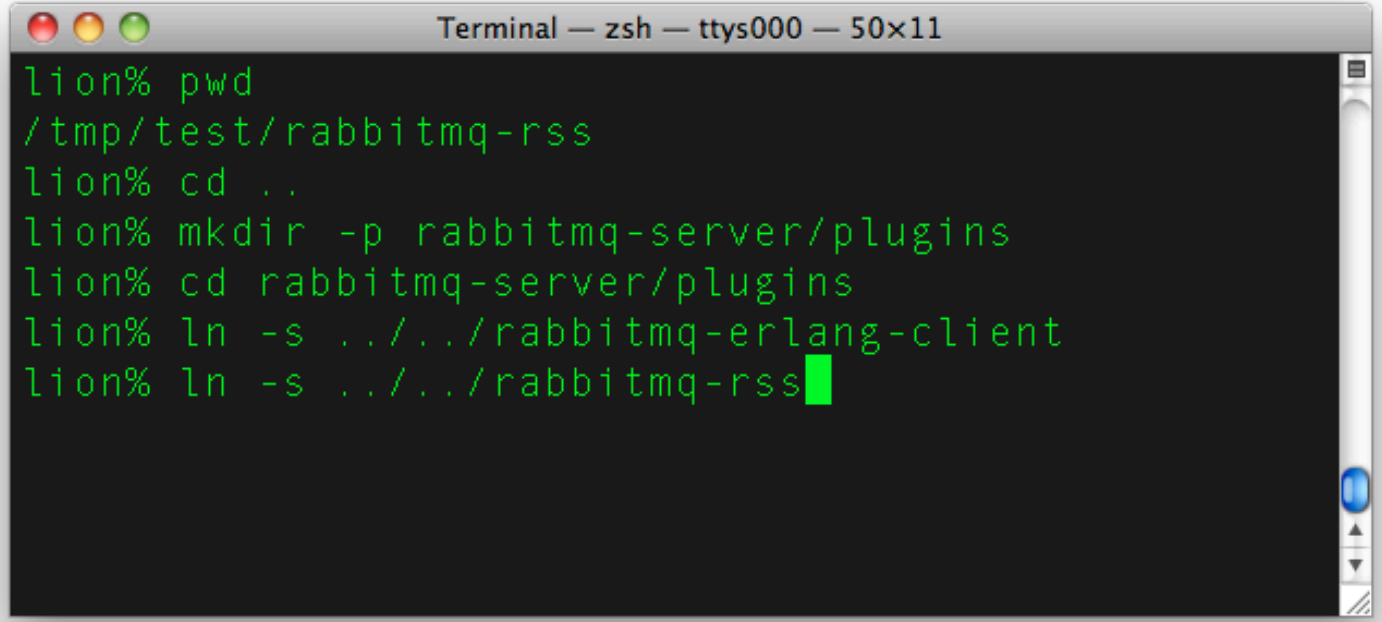
Build



Skeleton


Terminal — zsh — ttys000 — 50×11
lion% vim Makefile
lion% mkdir rabbitmq-rss
lion% mkdir rabbitmq-rss/ebin
lion% mkdir rabbitmq-rss/src
lion% cd rabbitmq-rss
lion% vi ebin/rabbit_rss.app
lion% vi src/rabbitmq_rss.erl
lion% vi src/rabbitmq_rss_sup.erl
lion% vi src/rabbitmq_rss_worker.erl
lion% vi src/rss.erl
lion%

Symlink Plugin



Run Broker

```
$ make run
(rabbit@rabbit9)1>
(rabbit@rabbit9)1>application:which_applications().

{rabbit_rss,"Embedded Rabbit RSS Reader","0.01"},
```

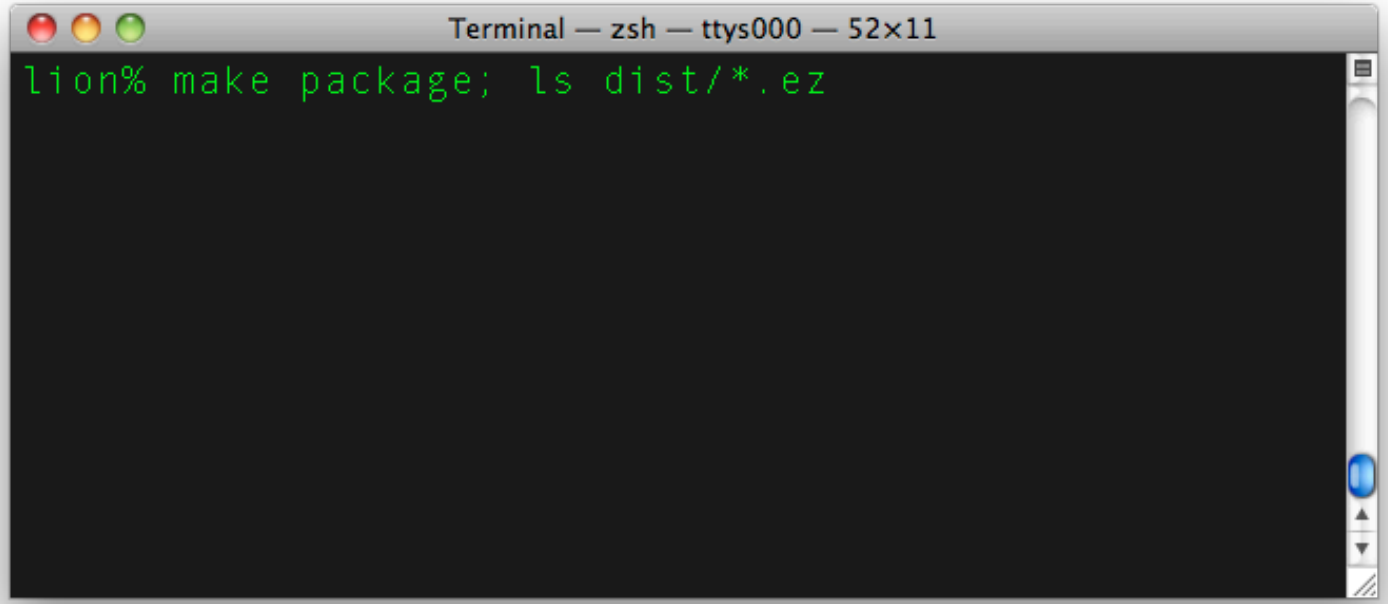
Test Broker

```
$ mkdir test
$ vi test/rabbit_rss_tests.erl
$ vi Makefile

TEST_APPS=amqp_client rabbit_rss
TEST_COMMANDS=rabbit_rss_tests:test()
START_RABBIT_IN_TESTS=true

$ make test
```

Package Plugin



Plugin Supporting Machinery

- Unit Tests
- Source Code
- Logger
- Printf
- AMQP Clients

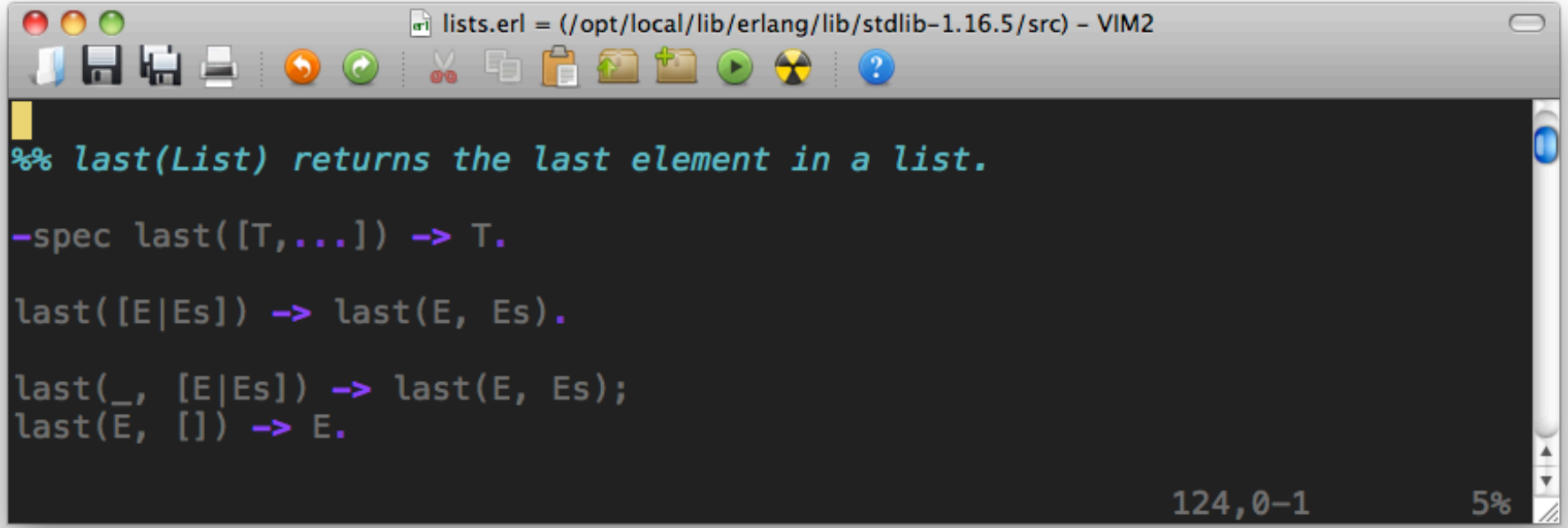
How To Build a Plugin

- RabbitMQ docs
- Source
- Examples

Use the Source!

- Documentation is hit/miss
- Google “erlang lists”
- http://www.erlang.org/doc/apps/stdlib/
- Or skip the docs and go directly to source
- ./lib/stdlib/src/lists.erl

Lists.erl


lists.erl = (/opt/local/lib/erlang/lib/stdlib-1.16.5/src) - VIM2
%% last(List) returns the last element in a list.
-spec last([T,...]) -> T.
last([E|Es]) -> last(E, Es).
last(_, [E|Es]) -> last(E, Es);
last(E, []) -> E.
124,0-1
5%

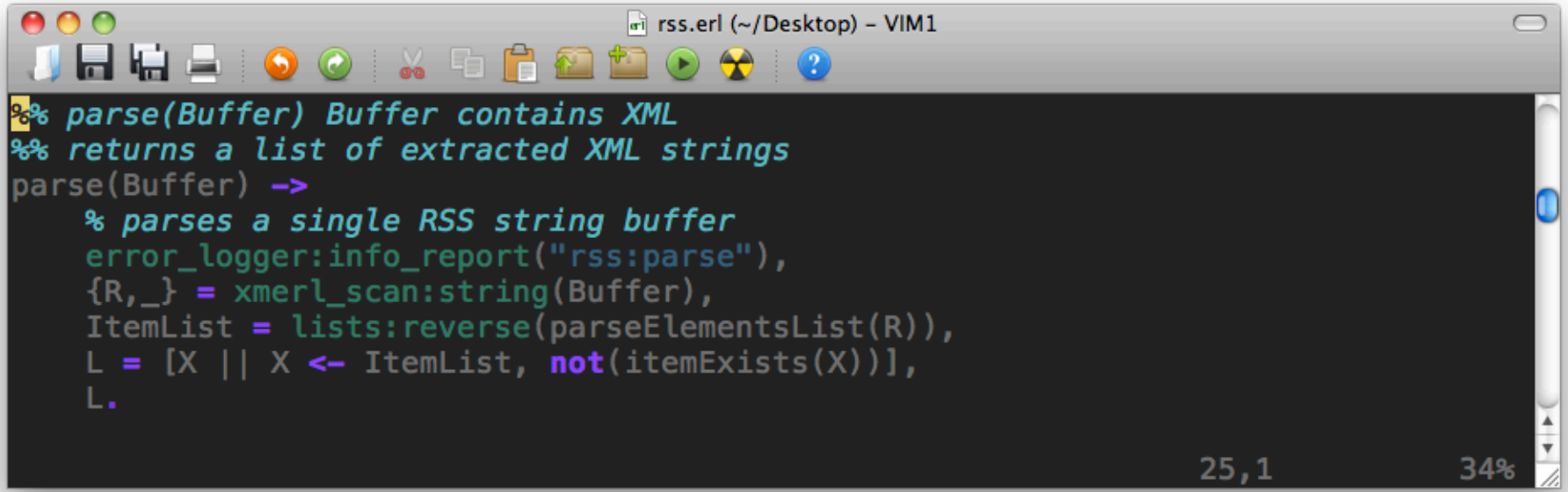
Logger

- Plugin executes in its own process
- No plugin stdout
- No plugin execution debugger

Logger: Continued

- Solution: Erlang's logger
- Output to /var/log/rabbitmq/<hostname
- Nothing to configure

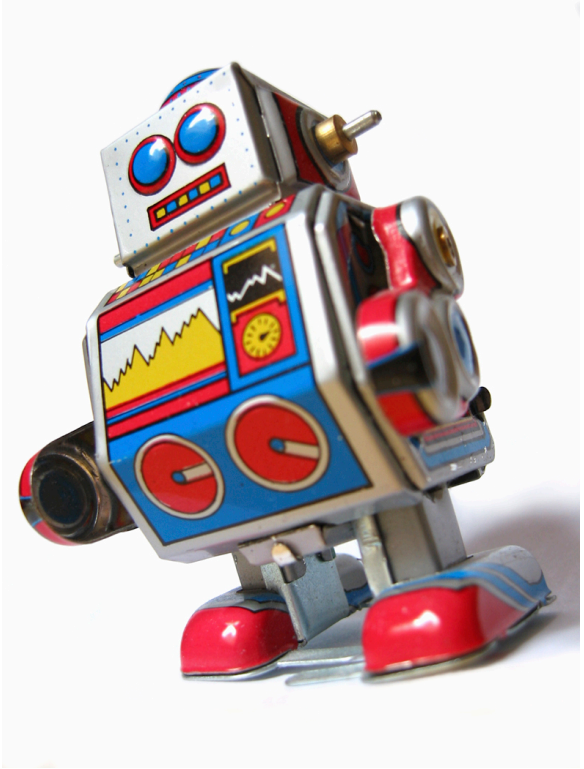
RSS.erl


rss.erl (~/Desktop) – VIM1
%% parse(Buffer) Buffer contains XML
%% returns a list of extracted XML strings
parse(Buffer) ->
% parses a single RSS string buffer
error_logger:info_report("rss:parse"),
{R,_} = xmerl_scan:string(Buffer),
ItemList = lists:reverse(parseElementsList(R)),
L = [X || X <- ItemList, not(itemExists(X))],
L.
25,1
34%

OTP Design Principles

- Supervision Tree
- Behaviors
- Applications
- Releases
- Release Handling

Machine Learning

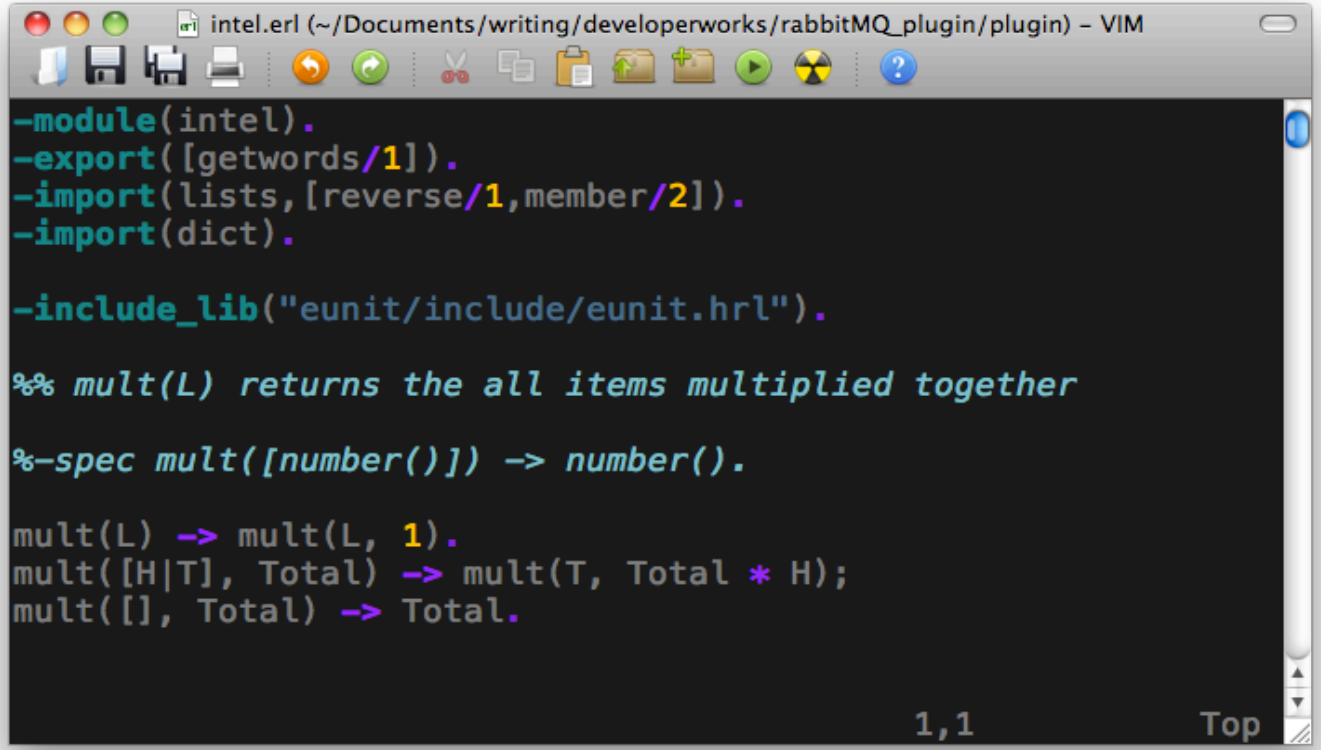


- The next step for the RSS plugin.
- A potential use case for RabbitMQ plugins.

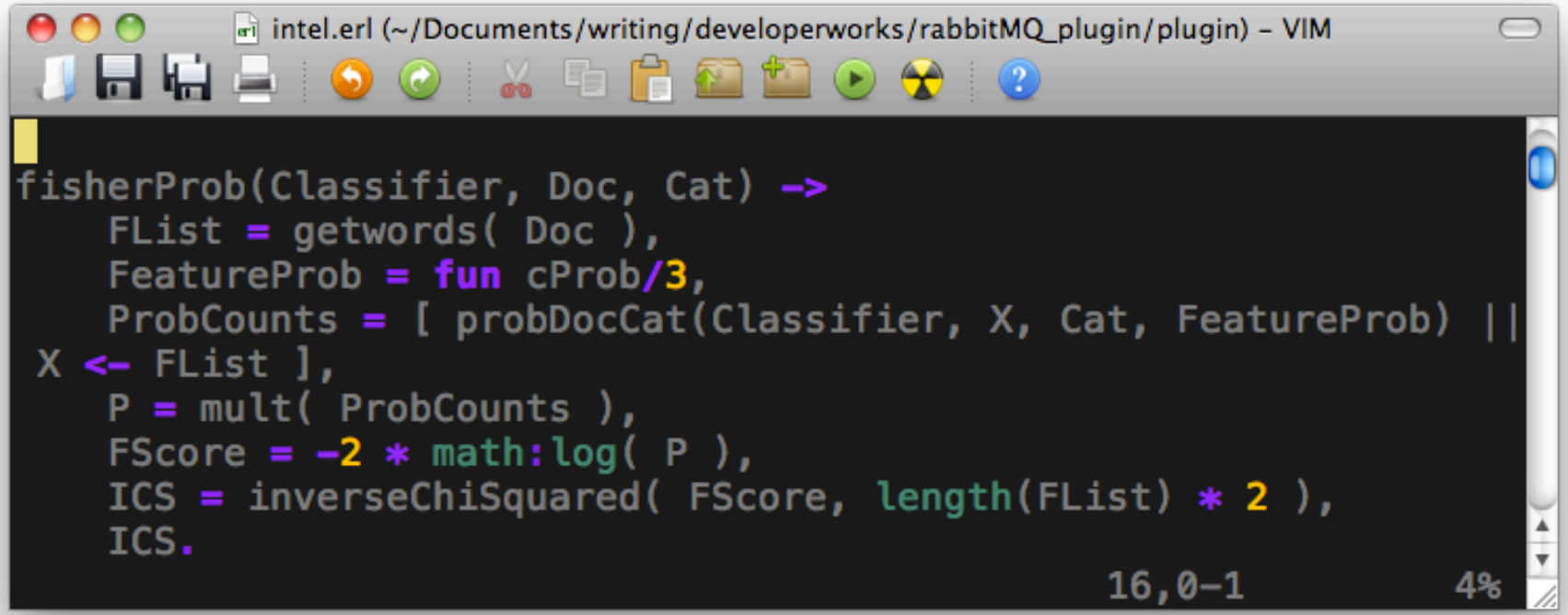
Fisher Classifier For RSS Feeds

- Data mining
- Classification

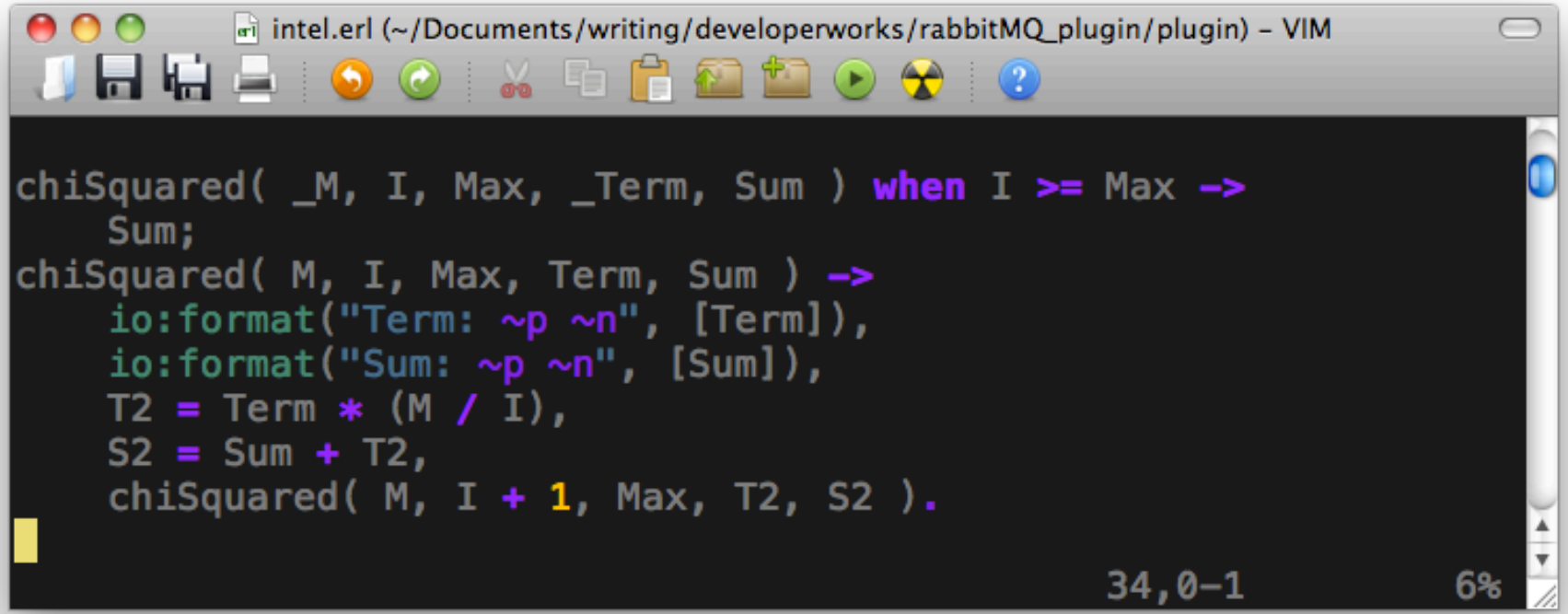
Fisher Classifier: Slide A



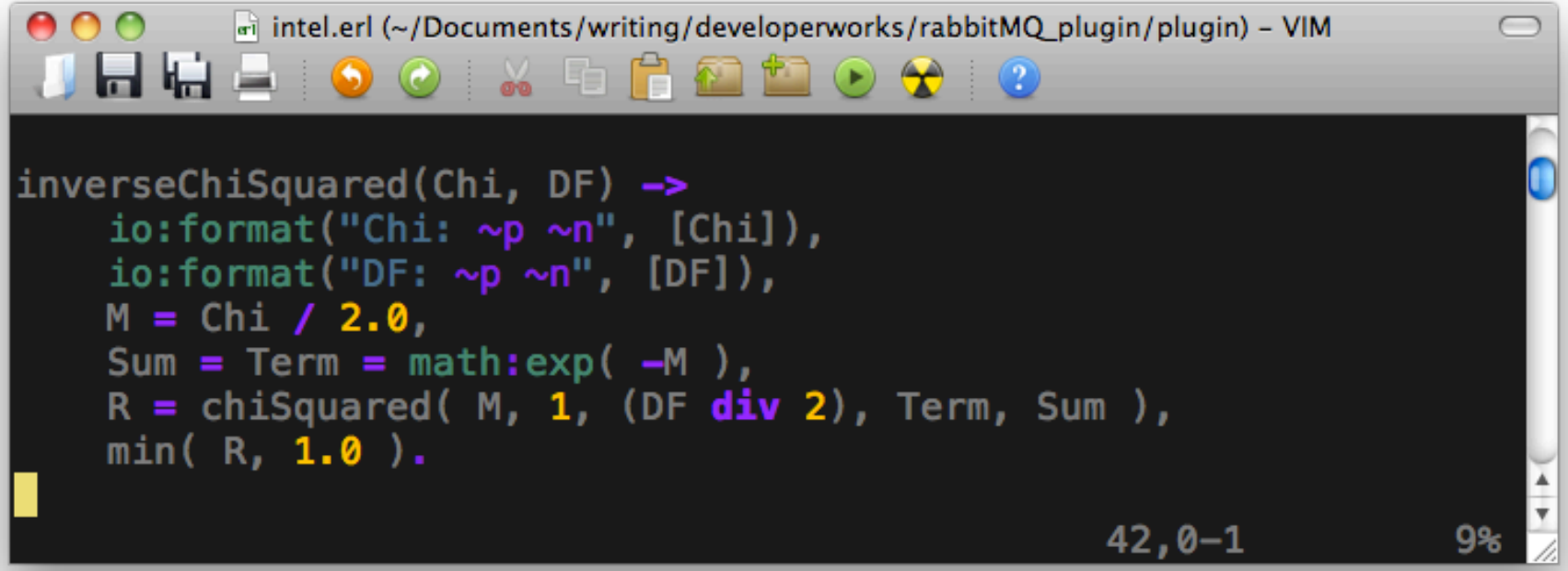
Fisher Classifier: Slide B



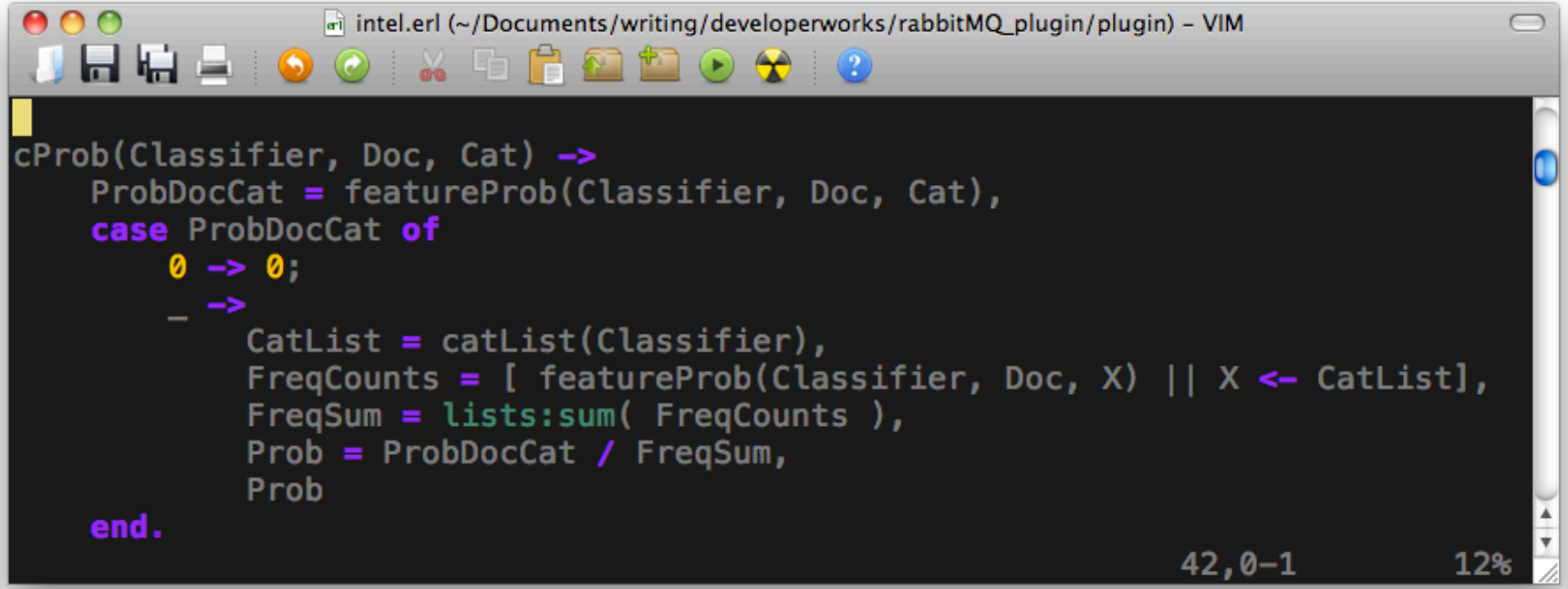
Fisher Classifier: Slide C



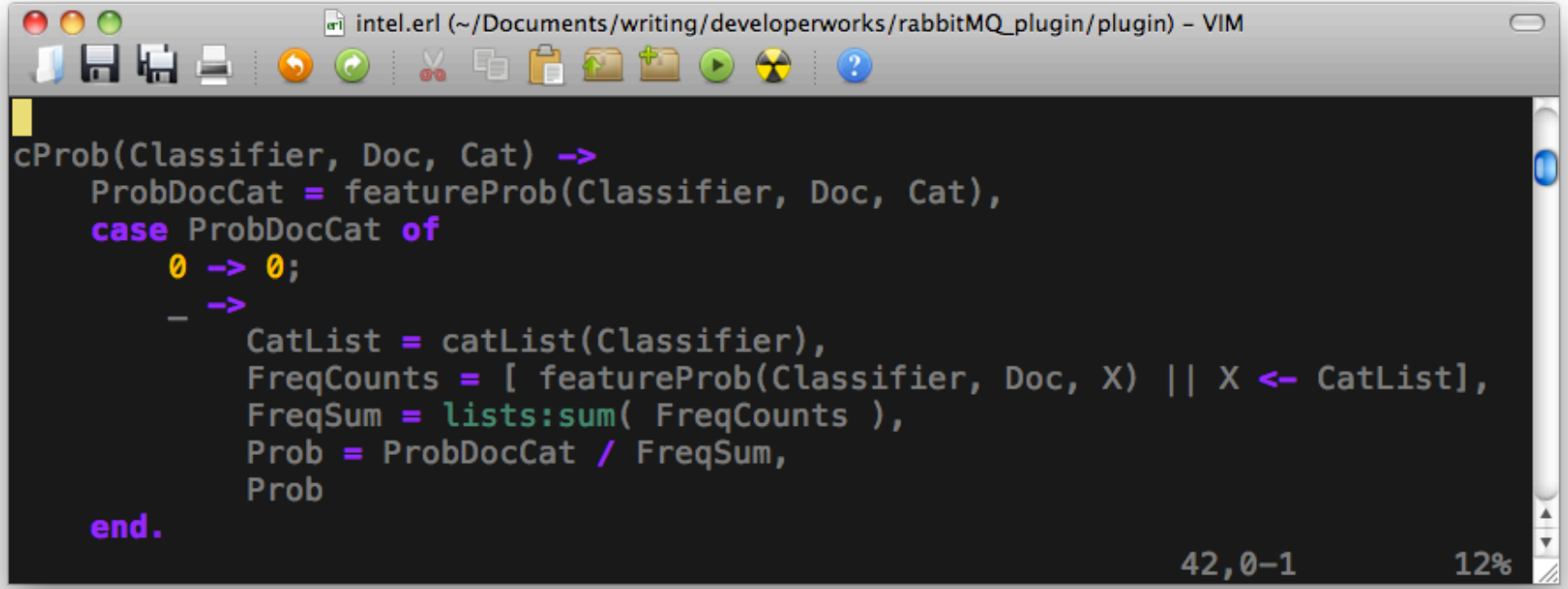
Fisher Classifier: Slide D



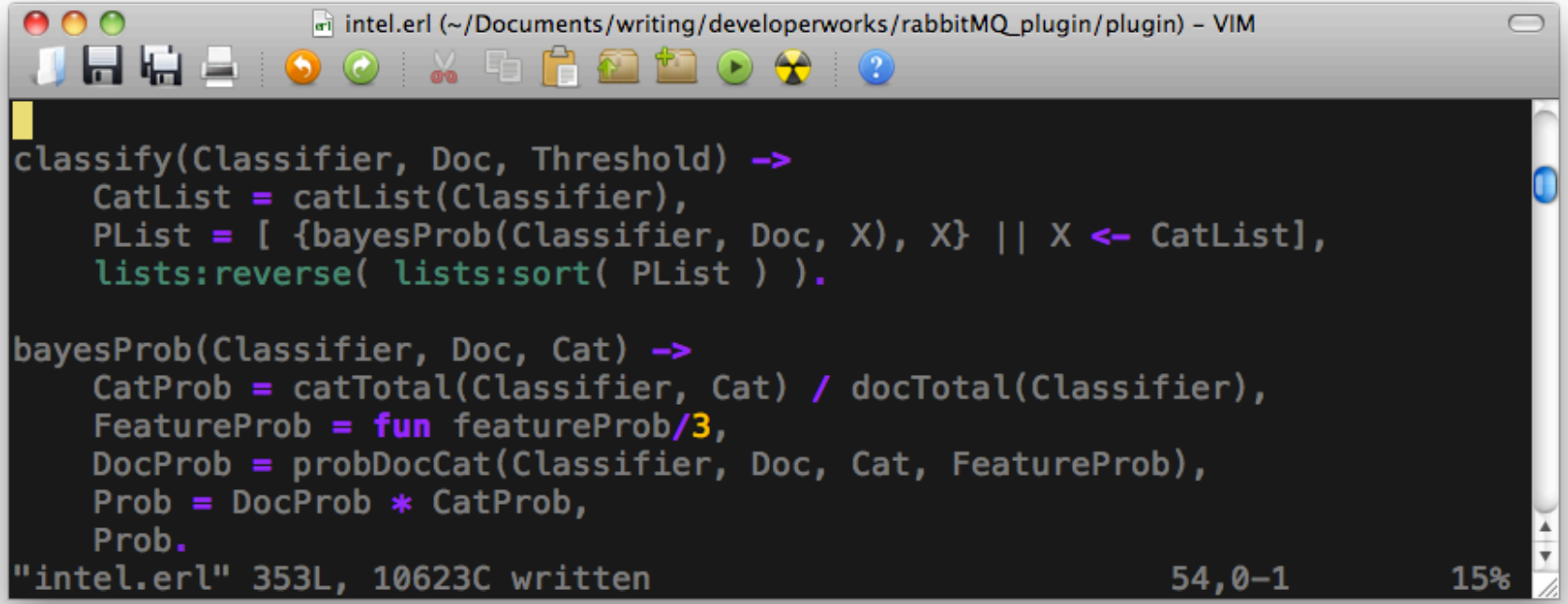
Fisher Classifier: Slide E



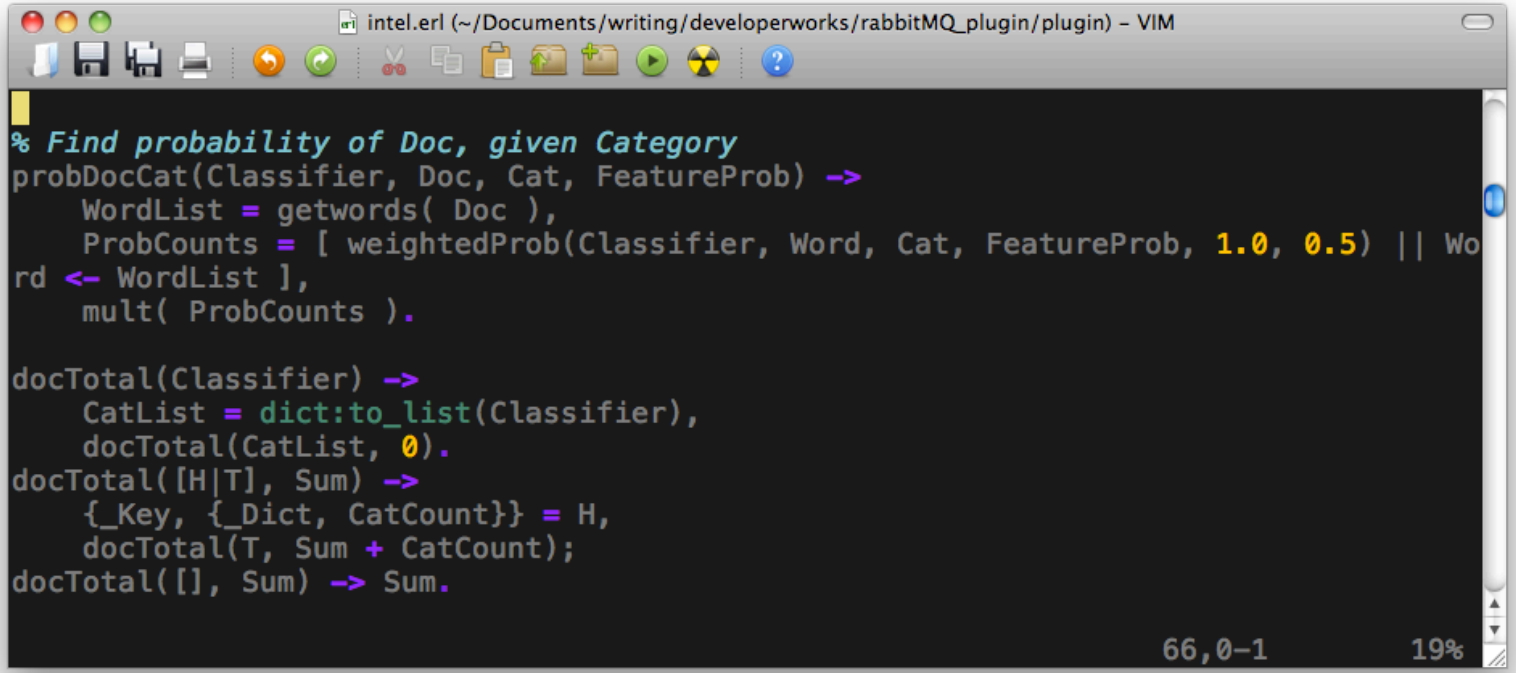
Fisher Classifier: Slide E



Fisher Classifier: Slide F



Fisher Classifier: Slide G



Conclusion

- AMQP
- Erlang
- Python
- RSS

Questions

- Code: **https://github.com/mikev/rabbitmq_rss_plugin**
- Contact: "Noah Gift" noah.gift@gmail.com, "Michael Vierling" mvierling@gmail.com