Releasing Extensions on PGXN

David E. Wheeler
PostgreSQL Experts, Inc.

PGCon
May 20, 2011

Problem Solved

Problem Solved

- Solved a database problem

Problem Solved

- Solved a database problem
- Want to share

Problem Solved

- Solved a database problem
- Want to share
- Open source it

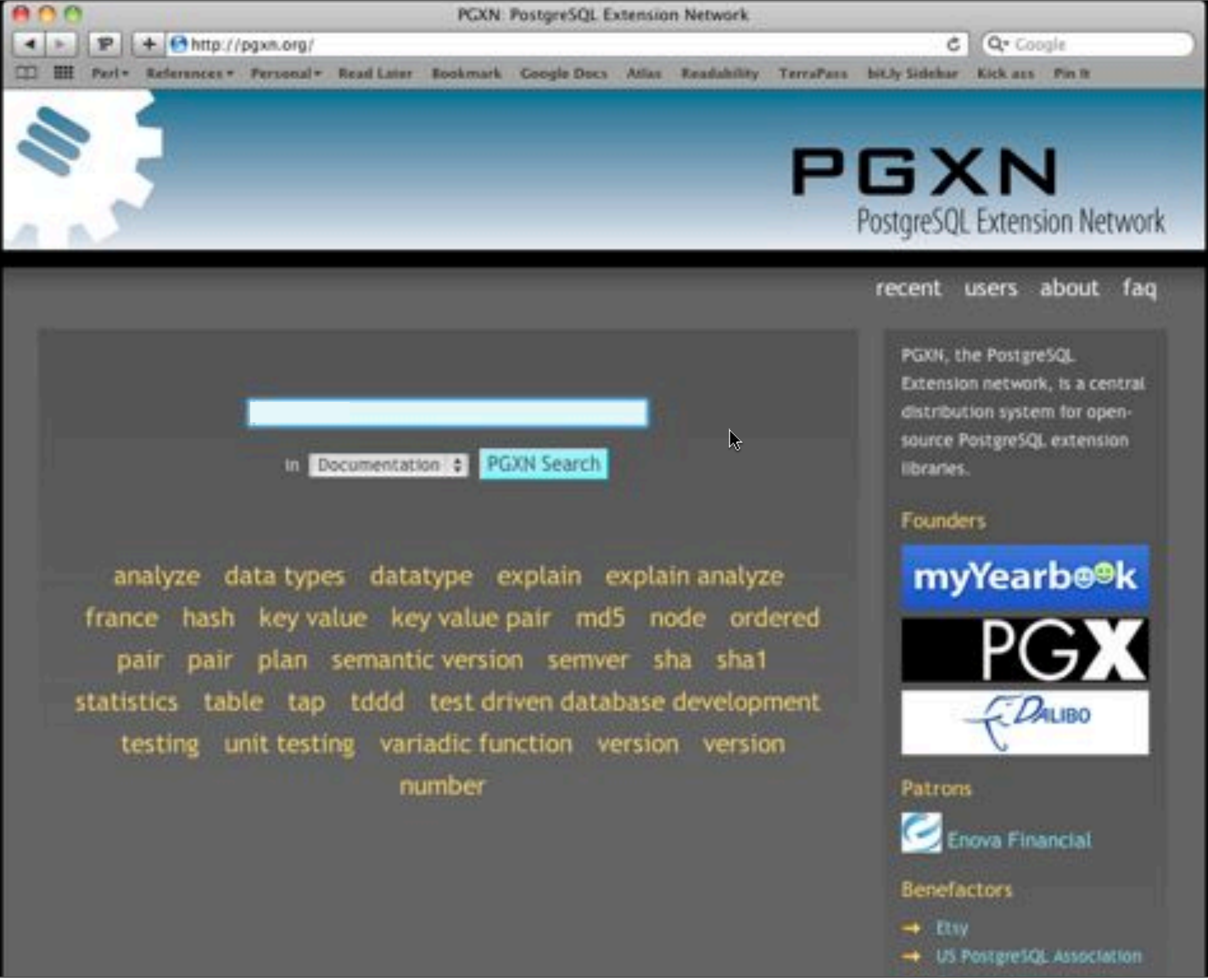
Problem Solved

- Solved a database problem
- Want to share
- Open source it
- Where to distribute

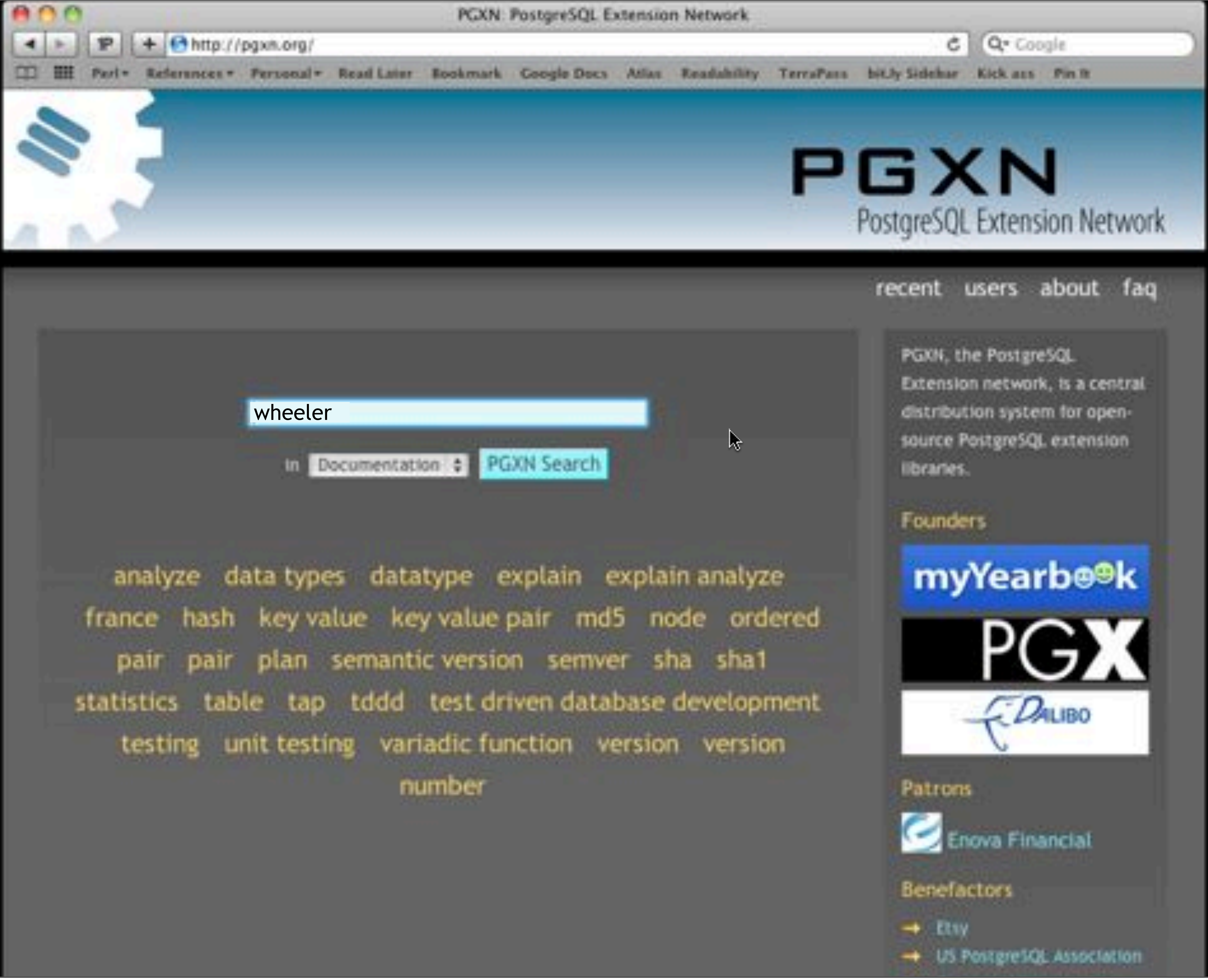
PGXN

PGXN

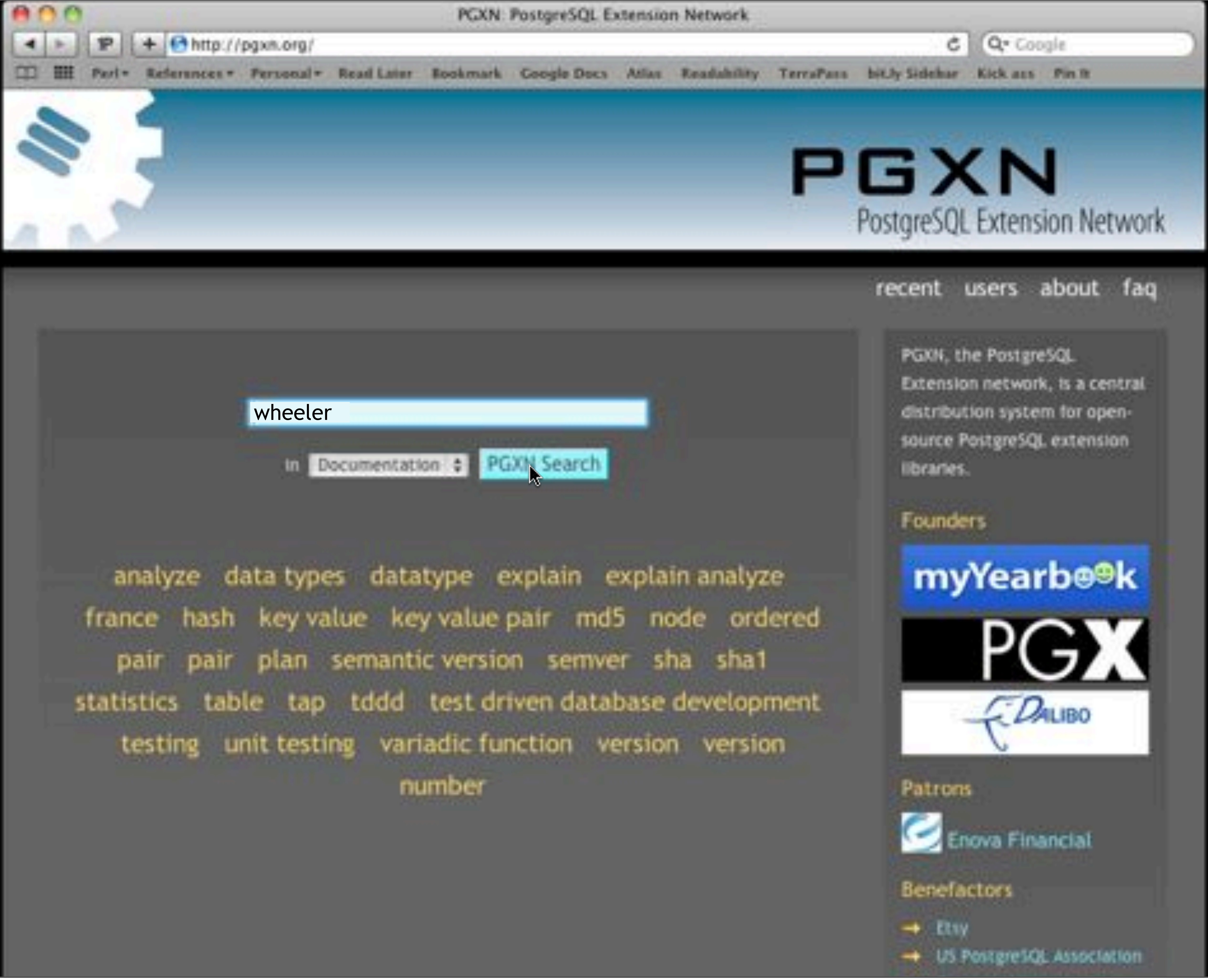
"The PostgreSQL Extension Network is a central distribution system for open-source PostgreSQL extension libraries and utilities."



PGXN, the PostgreSQL Extension network, is a central distribution system for open-source PostgreSQL extension libraries.


PGXN: PostgreSQL Extension Network
http://pgxn.org/
Perl References Personal Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass Pin It
PGXN
PostgreSQL Extension Network
recent users about faq
wheeler
in Documentation PGXN Search
analyze data types datatype explain explain analyze france hash key value key value pair md5 node ordered pair pair plan semantic version semver sha sha1 statistics table tap tddd test driven database development testing unit testing variadic function version version number
Founders
myYearbook
PGX
Dalibo
Patrons
Enova Financial
Benefactors
Etsy
US PostgreSQL Association

PGXN, the PostgreSQL Extension network, is a central distribution system for open-source PostgreSQL extension libraries.



PGXN: PostgreSQL Extension Network

http://pgxn.org/

PGXN
PostgreSQL Extension Network

recent users about faq

wheeler

in Documentation PGXN Search

analyze data types datatype explain explain analyze france hash key value key value pair md5 node ordered pair pair plan semantic version semver sha sha1 statistics table tap tddd test driven database development testing unit testing variadic function version version number

PGXN, the PostgreSQL Extension network, is a central distribution system for open-source PostgreSQL extension libraries.

Founders

myYearbook

PGX

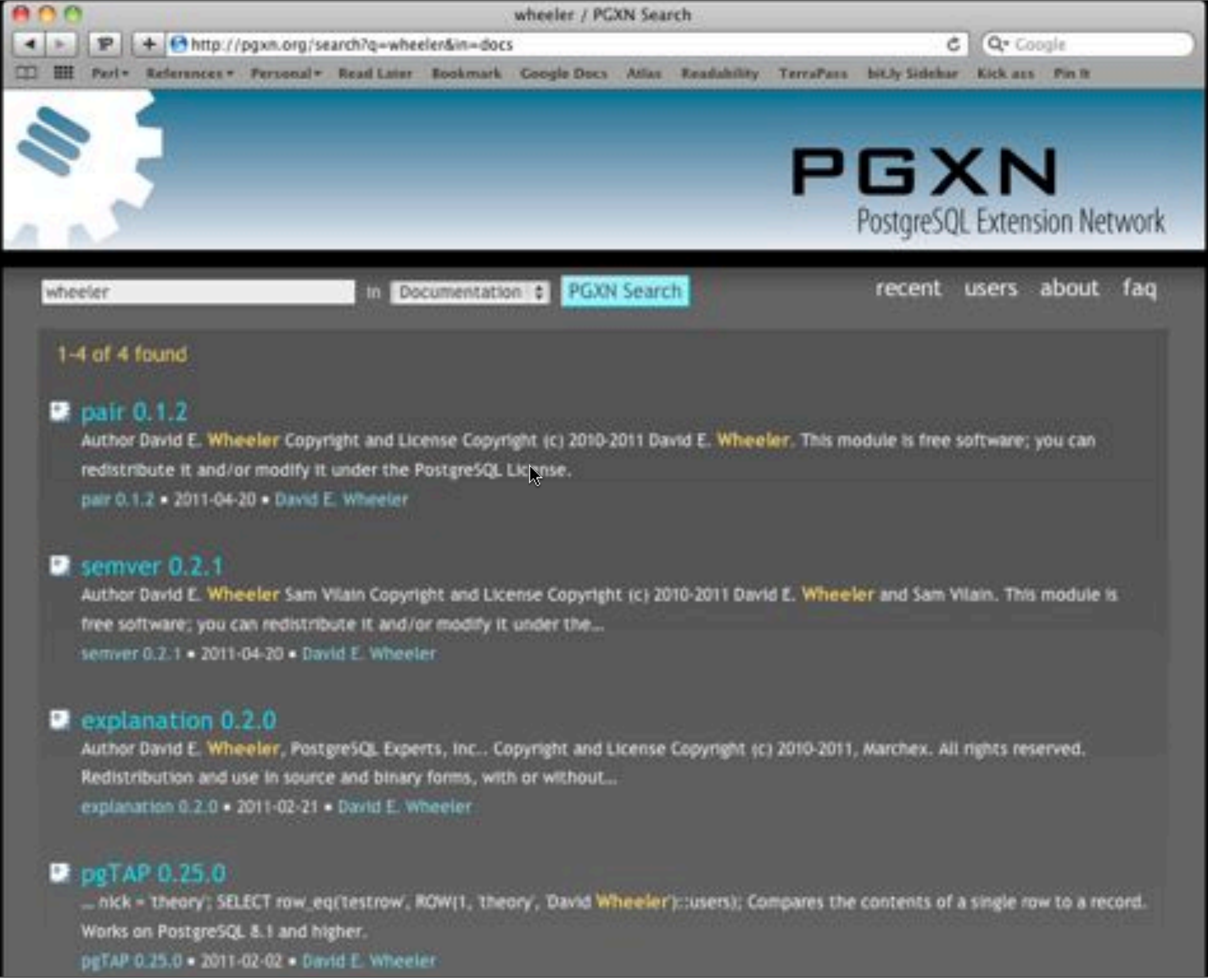
Dalibo

Patrons

Enova Financial

Benefactors

→ Etsy
→ US PostgreSQL Association



1-4 of 4 found

pair 0.1.2

Author David E. Wheeler Copyright and License Copyright (c) 2010-2011 David E. Wheeler. This module is free software; you can redistribute it and/or modify it under the PostgreSQL License.

pair 0.1.2 • 2011-04-20 • David E. Wheeler

semver 0.2.1

Author David E. Wheeler Sam Vilain Copyright and License Copyright (c) 2010-2011 David E. Wheeler and Sam Vilain. This module is free software; you can redistribute it and/or modify it under the...

semver 0.2.1 • 2011-04-20 • David E. Wheeler

explanation 0.2.0

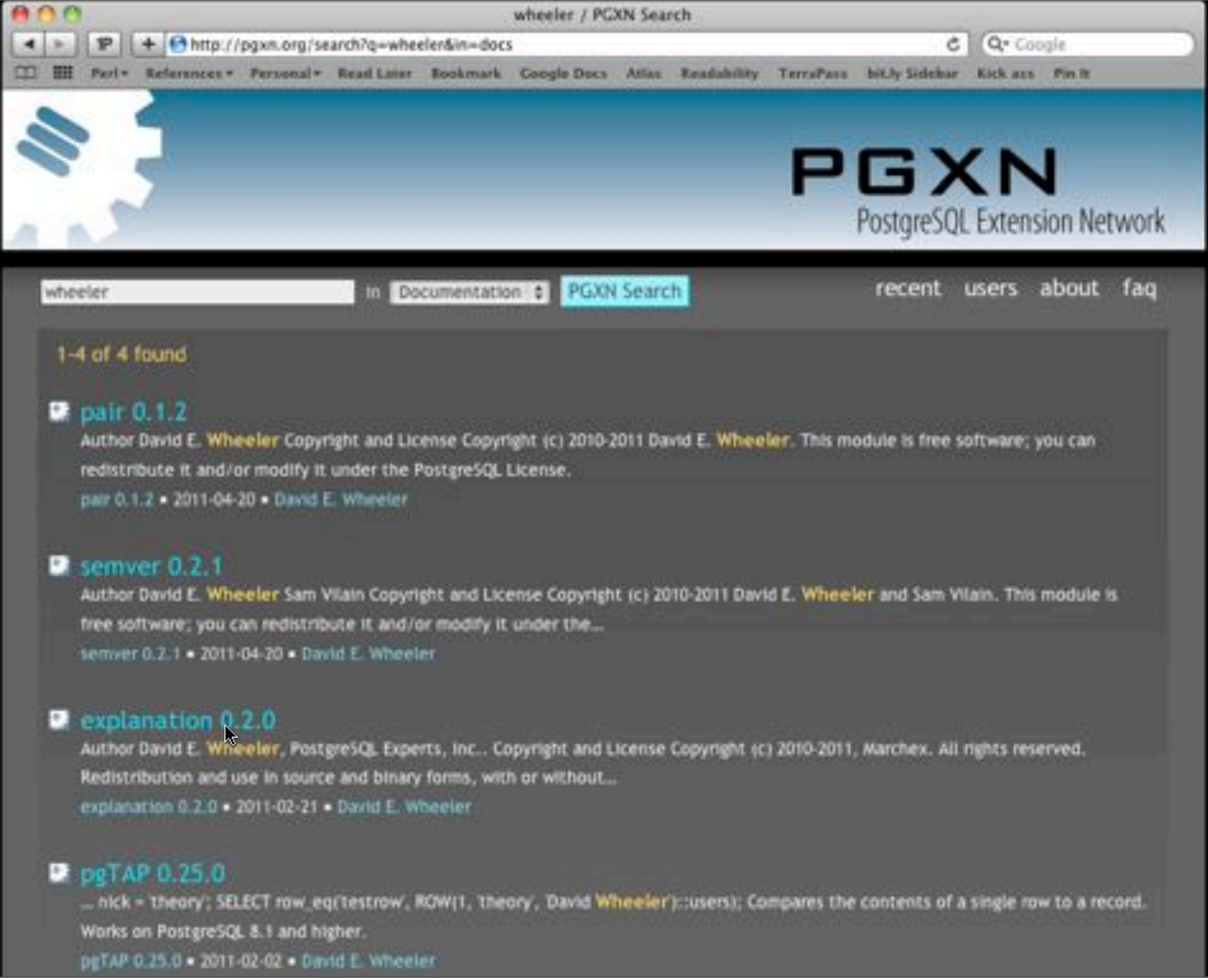
Author David E. Wheeler, PostgreSQL Experts, Inc.. Copyright and License Copyright (c) 2010-2011, Marchex. All rights reserved. Redistribution and use in source and binary forms, with or without...

explanation 0.2.0 • 2011-02-21 • David E. Wheeler

pgTAP 0.25.0

... nick = 'theory'; SELECT row_eq('testrow', ROW(1, 'theory', 'David Wheeler')::users); Compares the contents of a single row to a record. Works on PostgreSQL 8.1 and higher.

pgTAP 0.25.0 • 2011-02-02 • David E. Wheeler


wheeler / PGXN Search
http://pgxn.org/search?q=wheeler&in=docs
Perl References Personal Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass Pin It
PGXN
PostgreSQL Extension Network
wheeler
in
Documentation
PGXN Search
recent
users
about
faq

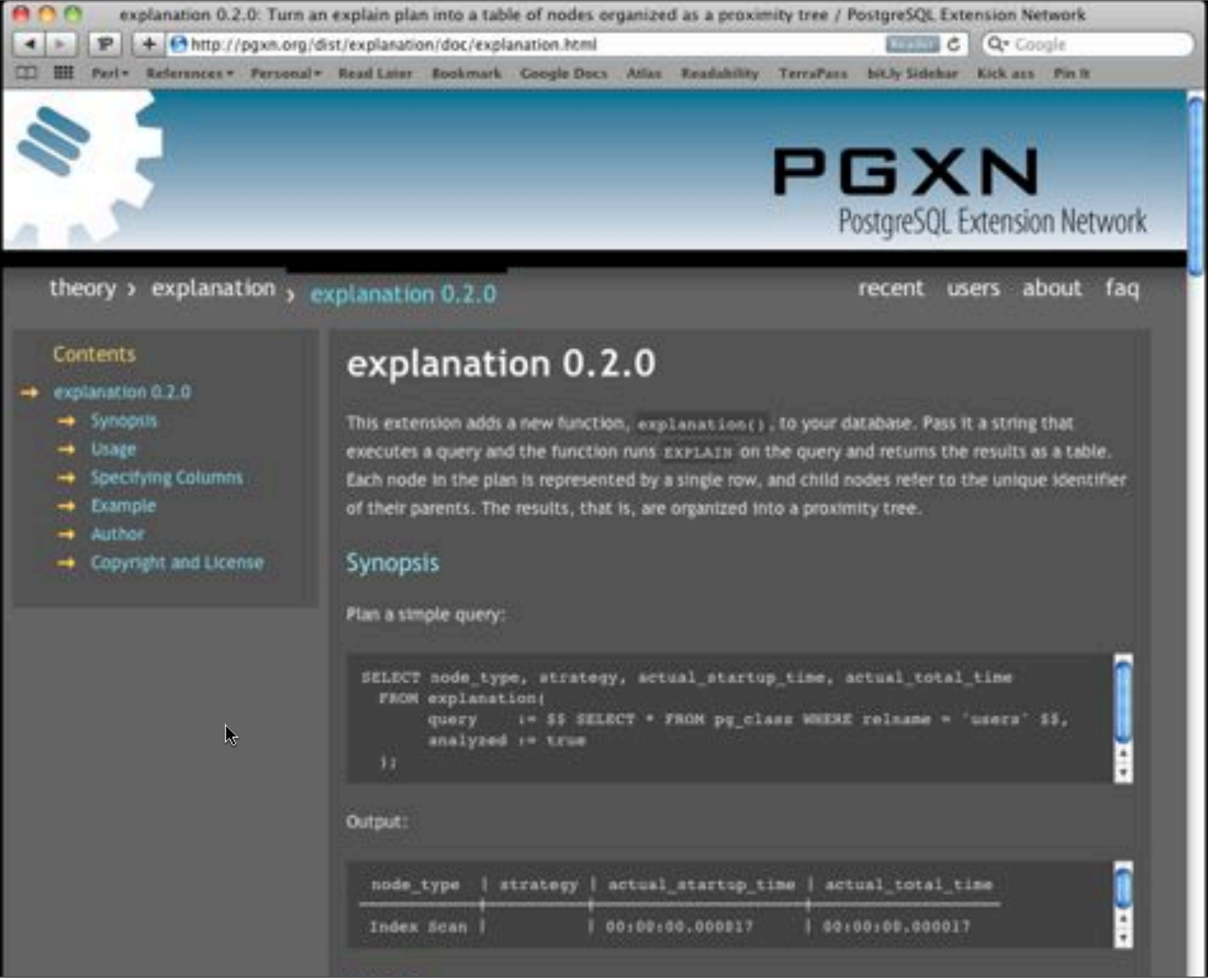
1-4 of 4 found

pair 0.1.2
Author David E. Wheeler Copyright and License Copyright (c) 2010-2011 David E. Wheeler. This module is free software; you can redistribute it and/or modify it under the PostgreSQL License.
pair 0.1.2 • 2011-04-20 • David E. Wheeler

semver 0.2.1
Author David E. Wheeler Sam Vilain Copyright and License Copyright (c) 2010-2011 David E. Wheeler and Sam Vilain. This module is free software; you can redistribute it and/or modify it under the...
semver 0.2.1 • 2011-04-20 • David E. Wheeler

explanation 0.2.0
Author David E. Wheeler, PostgreSQL Experts, Inc.. Copyright and License Copyright (c) 2010-2011, Marchex. All rights reserved. Redistribution and use in source and binary forms, with or without...
explanation 0.2.0 • 2011-02-21 • David E. Wheeler

pgTAP 0.25.0
... nick = 'theory'; SELECT row_eq(testrow, ROW(1, 'theory', 'David Wheeler')::users); Compares the contents of a single row to a record. Works on PostgreSQL 8.1 and higher.
pgTAP 0.25.0 • 2011-02-02 • David E. Wheeler


explanation 0.2.0: Turn an explain plan into a table of nodes organized as a proximity tree / PostgreSQL Extension Network
http://pgxn.org/dist/explanation/doc/explanation.html
PGXN
PostgreSQL Extension Network
theory › explanation › explanation 0.2.0
recent users about faq

explanation 0.2.0

This extension adds a new function, `explanation()`, to your database. Pass it a string that executes a query and the function runs `EXPLAIN` on the query and returns the results as a table. Each node in the plan is represented by a single row, and child nodes refer to the unique identifier of their parents. The results, that is, are organized into a proximity tree.

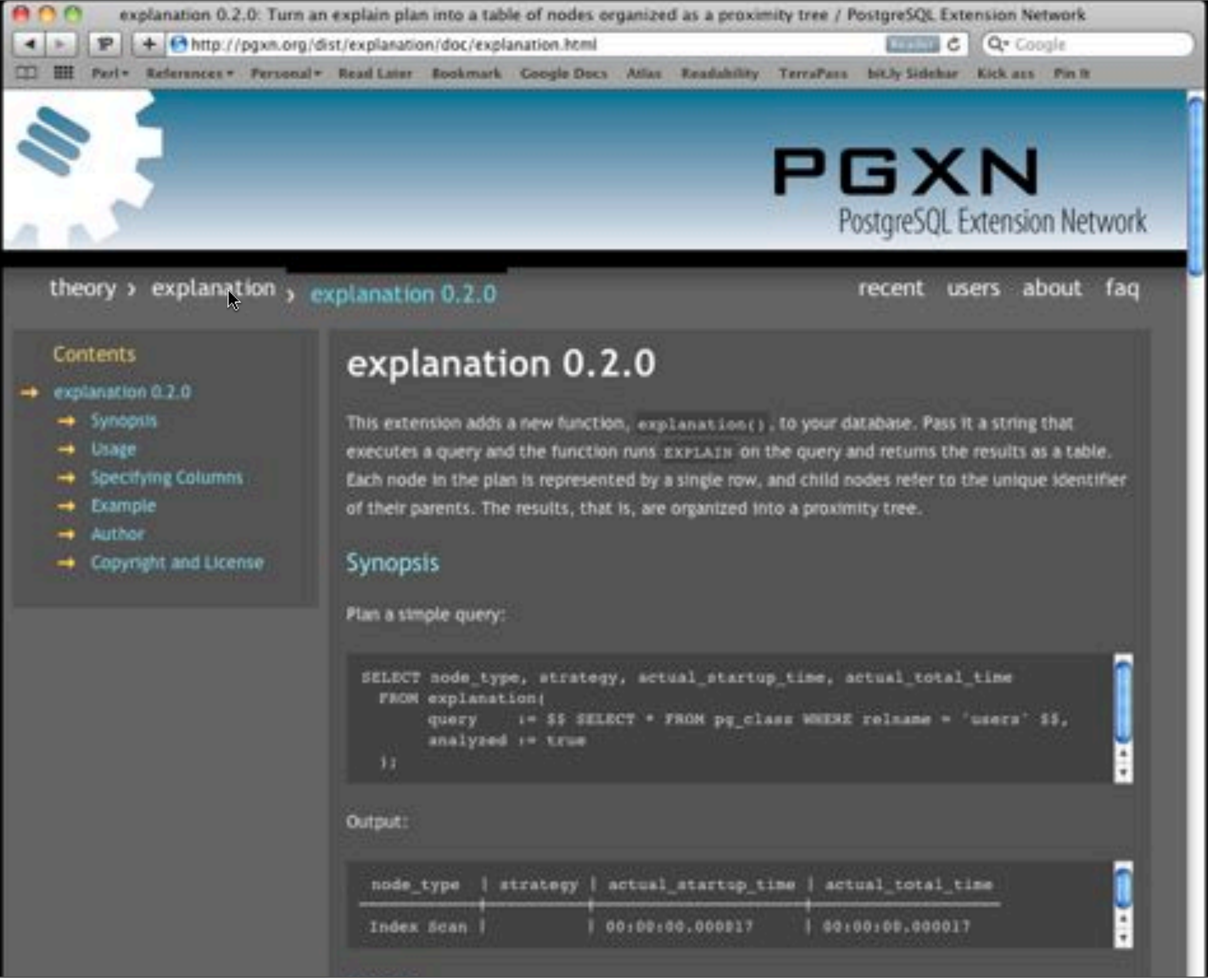
Synopsis

Plan a simple query:

```
SELECT node_type, strategy, actual_startup_time, actual_total_time
  FROM explanation(
       query    := $$ SELECT * FROM pg_class WHERE relname = 'users' $$,
       analyzed := true
  );
```

Output:

```
 node_type  | strategy | actual_startup_time | actual_total_time
------------+----------+---------------------+-------------------
 Index Scan |          | 00:00:00.000017     | 00:00:00.000017
```


explanation 0.2.0: Turn an explain plan into a table of nodes organized as a proximity tree / PostgreSQL Extension Network
http://pgxn.org/dist/explanation/doc/explanation.html
PGXN
PostgreSQL Extension Network
theory › explanation › explanation 0.2.0
recent users about faq

explanation 0.2.0

This extension adds a new function, `explanation()`, to your database. Pass it a string that executes a query and the function runs `EXPLAIN` on the query and returns the results as a table. Each node in the plan is represented by a single row, and child nodes refer to the unique identifier of their parents. The results, that is, are organized into a proximity tree.

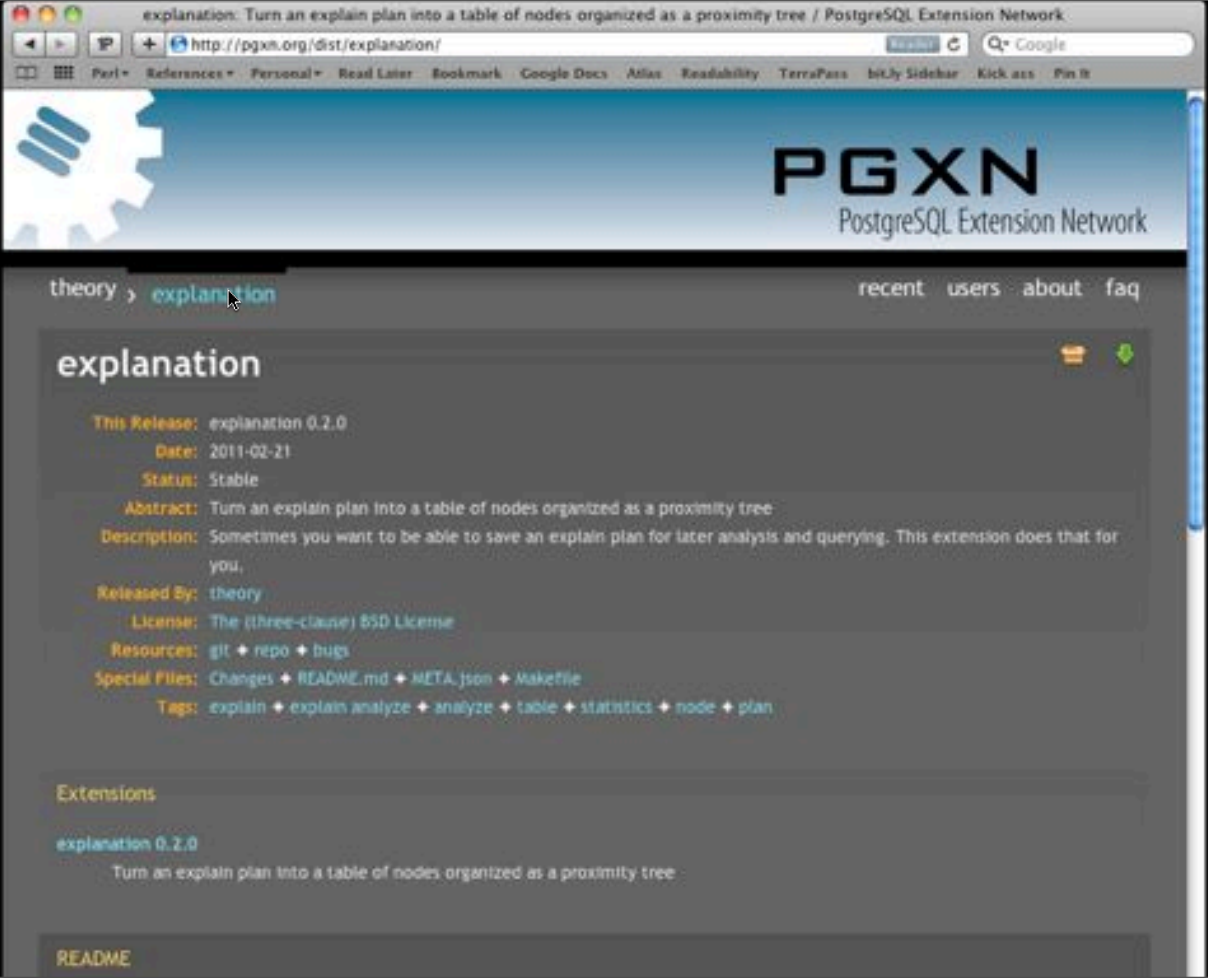
Synopsis

Plan a simple query:

```
SELECT node_type, strategy, actual_startup_time, actual_total_time
  FROM explanation(
       query    := $$ SELECT * FROM pg_class WHERE relname = 'users' $$,
       analyzed := true
  );
```

Output:

```
 node_type  | strategy | actual_startup_time | actual_total_time
------------+----------+---------------------+-------------------
 Index Scan |          | 00:00:00.000017     | 00:00:00.000017
```


explanation: Turn an explain plan into a table of nodes organized as a proximity tree / PostgreSQL Extension Network
http://pgxn.org/dist/explanation/
Perl References Personal Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass Pin It
PGXN
PostgreSQL Extension Network
theory › explanation
recent users about faq

explanation

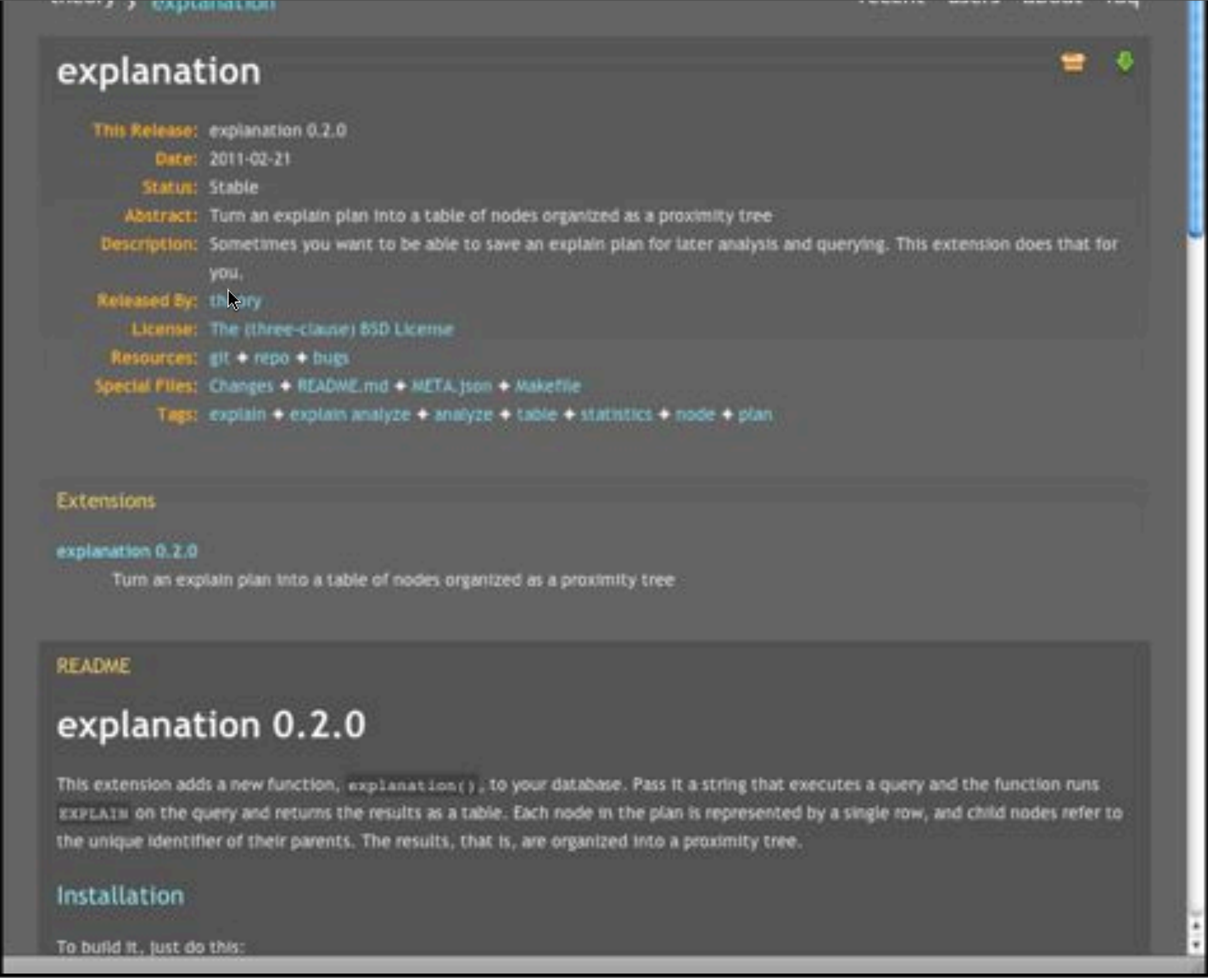
This Release:	explanation 0.2.0
Date:	2011-02-21
Status:	Stable
Abstract:	Turn an explain plan into a table of nodes organized as a proximity tree
Description:	Sometimes you want to be able to save an explain plan for later analysis and querying. This extension does that for you.
Released By:	theory
License:	The (three-clause) BSD License
Resources:	git ✦ repo ✦ bugs
Special Files:	Changes ✦ README.md ✦ META.json ✦ Makefile
Tags:	explain ✦ explain analyze ✦ analyze ✦ table ✦ statistics ✦ node ✦ plan

Extensions

explanation 0.2.0

Turn an explain plan into a table of nodes organized as a proximity tree

README



theory › explanation

explanation

This Release:	explanation 0.2.0
Date:	2011-02-21
Status:	Stable
Abstract:	Turn an explain plan into a table of nodes organized as a proximity tree
Description:	Sometimes you want to be able to save an explain plan for later analysis and querying. This extension does that for you.
Released By:	theory
License:	The (three-clause) BSD License
Resources:	git ✦ repo ✦ bugs
Special Files:	Changes ✦ README.md ✦ META.json ✦ Makefile
Tags:	explain ✦ explain analyze ✦ analyze ✦ table ✦ statistics ✦ node ✦ plan

Extensions

explanation 0.2.0

Turn an explain plan into a table of nodes organized as a proximity tree

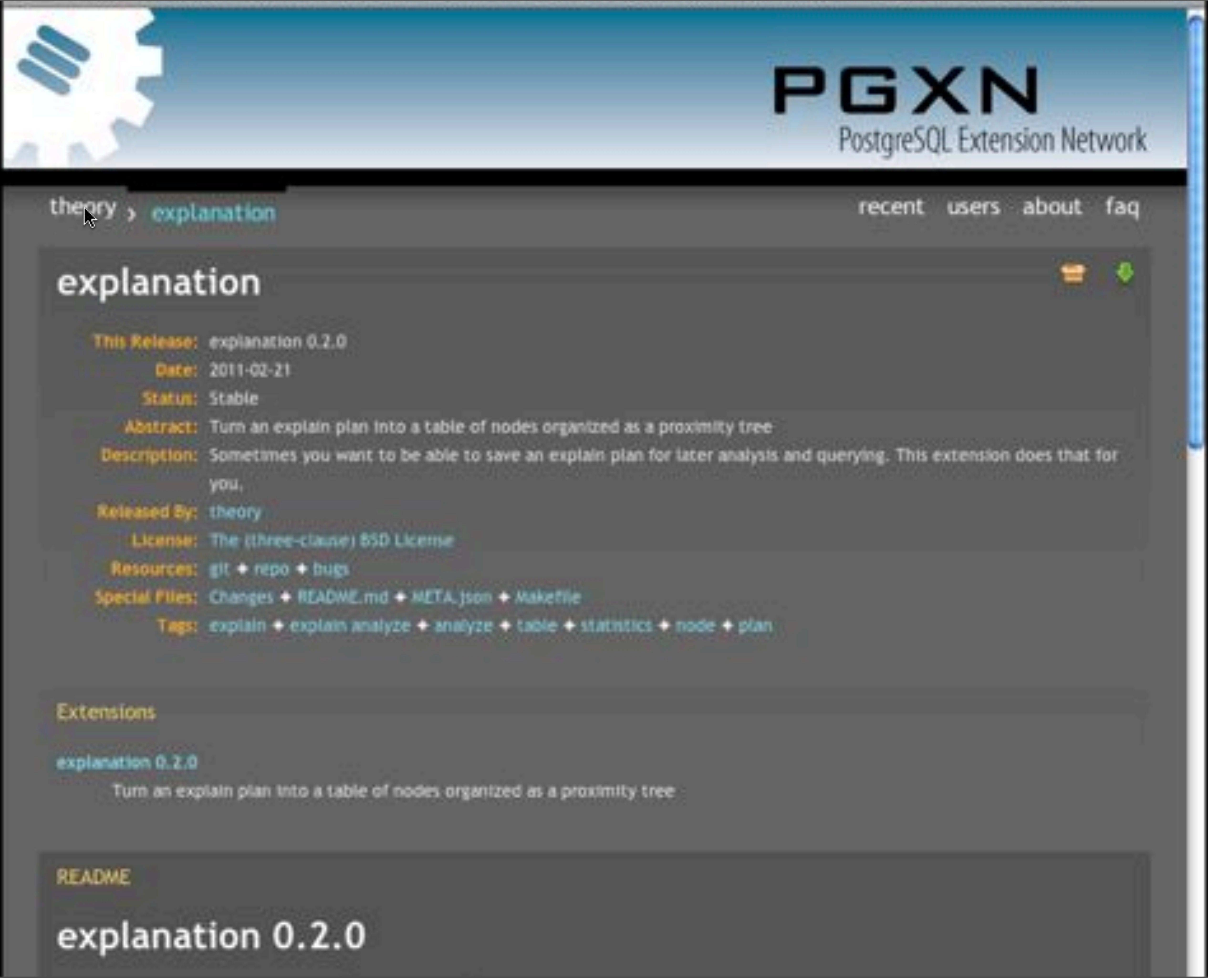
README

explanation 0.2.0

This extension adds a new function, `explanation()`, to your database. Pass it a string that executes a query and the function runs `EXPLAIN` on the query and returns the results as a table. Each node in the plan is represented by a single row, and child nodes refer to the unique identifier of their parents. The results, that is, are organized into a proximity tree.

Installation

To build it, just do this:



PGXN
PostgreSQL Extension Network

theory › explanation   recent   users   about   faq

explanation

This Release:	explanation 0.2.0
Date:	2011-02-21
Status:	Stable
Abstract:	Turn an explain plan into a table of nodes organized as a proximity tree
Description:	Sometimes you want to be able to save an explain plan for later analysis and querying. This extension does that for you.
Released By:	theory
License:	The (three-clause) BSD License
Resources:	git ✦ repo ✦ bugs
Special Files:	Changes ✦ README.md ✦ META.json ✦ Makefile
Tags:	explain ✦ explain analyze ✦ analyze ✦ table ✦ statistics ✦ node ✦ plan

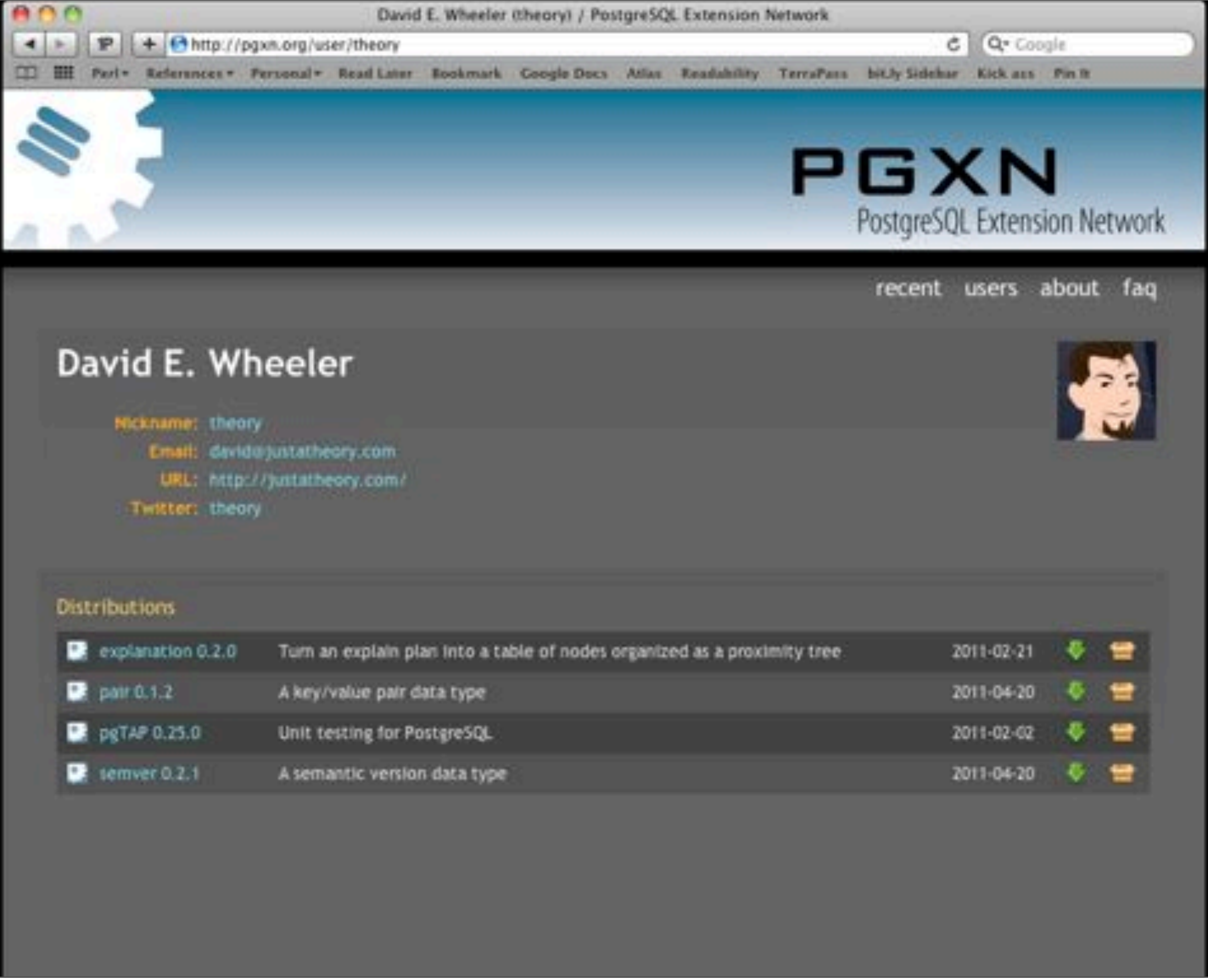
Extensions

explanation 0.2.0

Turn an explain plan into a table of nodes organized as a proximity tree

README

explanation 0.2.0



David E. Wheeler (theory) / PostgreSQL Extension Network

http://pgxn.org/user/theory

Perl References Personal Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass Pin It
PGXN
PostgreSQL Extension Network
recent users about faq

David E. Wheeler

Nickname:	theory
Email:	david@justatheory.com
URL:	http://justatheory.com/
Twitter:	theory

Distributions

explanation 0.2.0	Turn an explain plan into a table of nodes organized as a proximity tree	2011-02-21
pair 0.1.2	A key/value pair data type	2011-04-20
pgTAP 0.25.0	Unit testing for PostgreSQL	2011-02-02
semver 0.2.1	A semantic version data type	2011-04-20

Your Solution

Your Solution

- You’ve solved a problem

Your Solution

- You've solved a problem
- Using database objects

Your Solution

- You've solved a problem
- Using database objects
- Packaged like contrib

Your Solution

- You've solved a problem
- Using database objects
- Packaged like contrib
- Want to open-source it

Your Solution

- You’ve solved a problem
- Using database objects
- Packaged like contrib
- Want to open-source it
- How to distribute on PGXN?

Your Solution

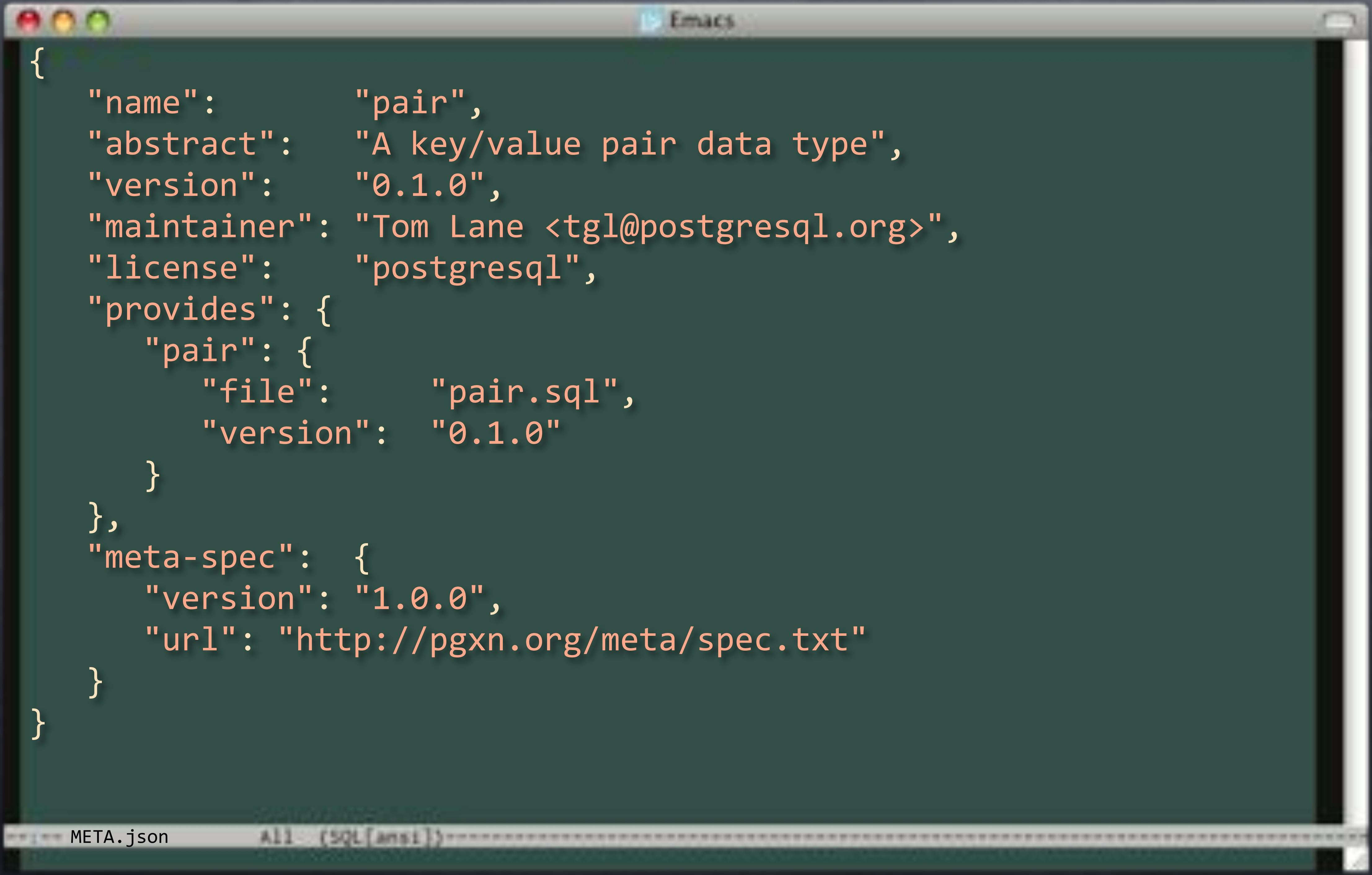
- You've solved a problem
- Using database objects
- Packaged like contrib
- Want to open-source it
- How to distribute on PGXN?
- Just one thing:

Your Solution

- You’ve solved a problem
- Using database objects
- Packaged like contrib
- Want to open-source it
- How to distribute on PGXN?
- Just one thing:
- META.json

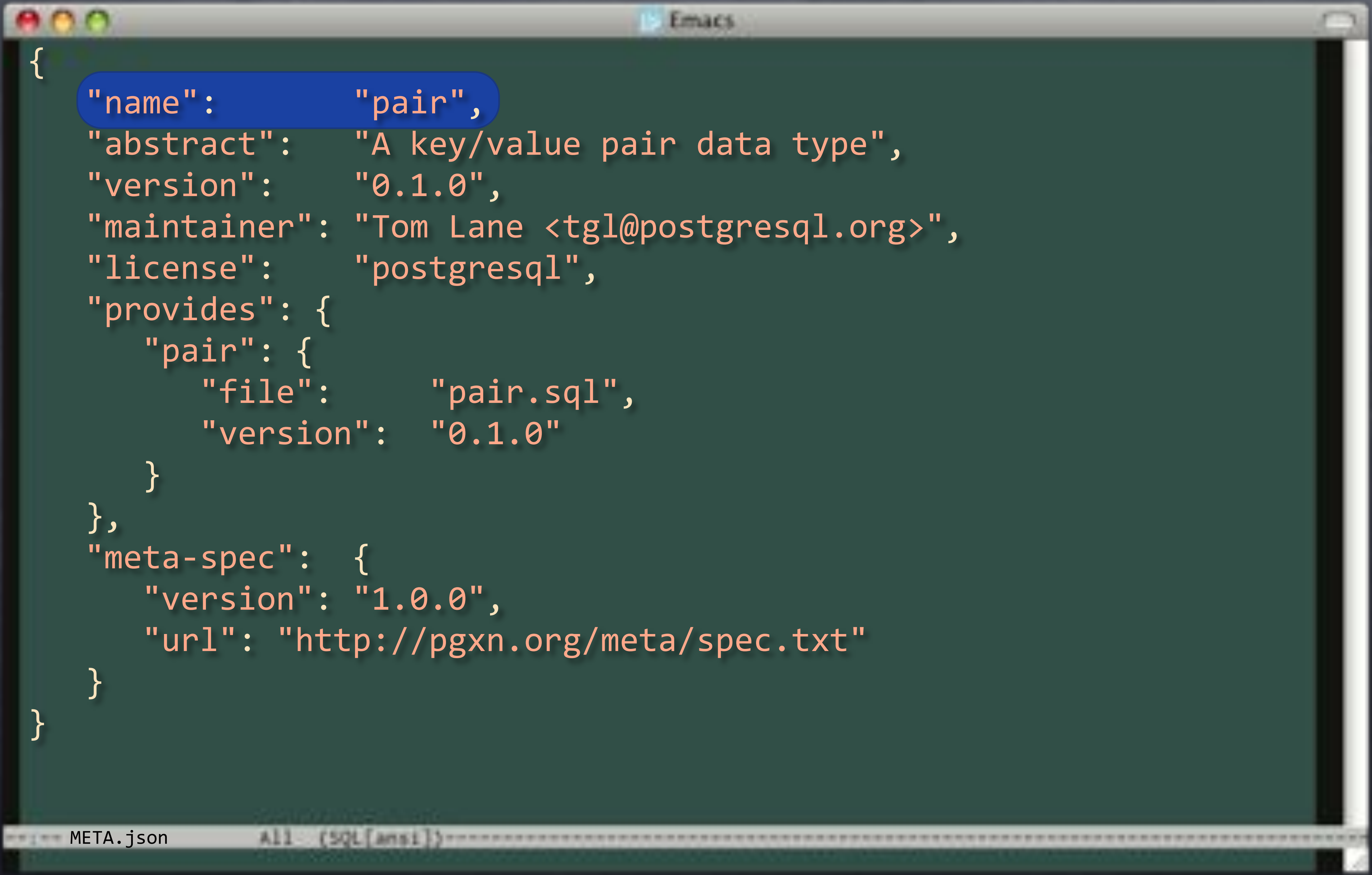
META.json

META.json



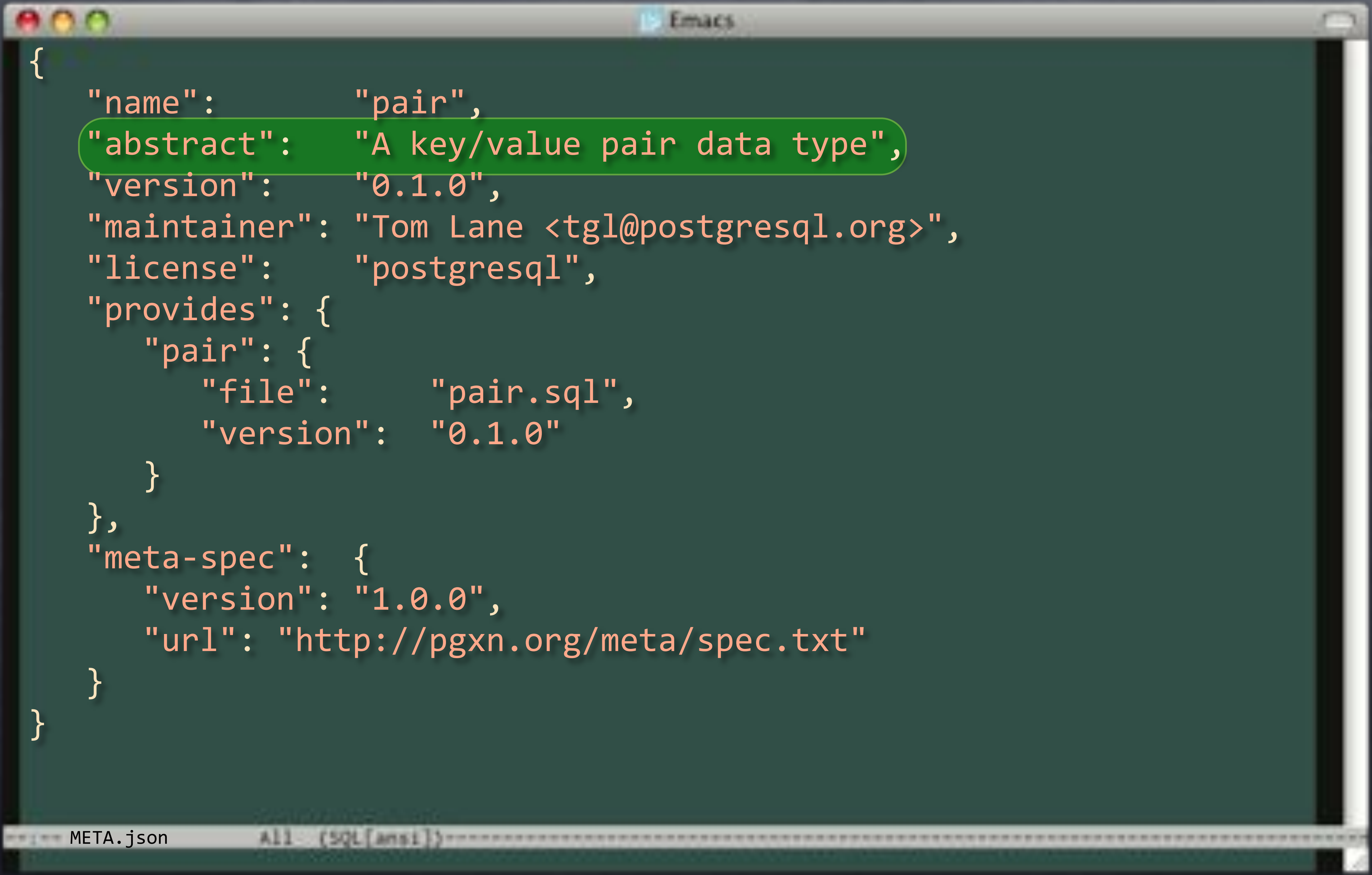
```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.0",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

META.json



```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.0",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "provides": {
      "pair": {
         "file":      "pair.sql",
         "version":   "0.1.0"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

META.json

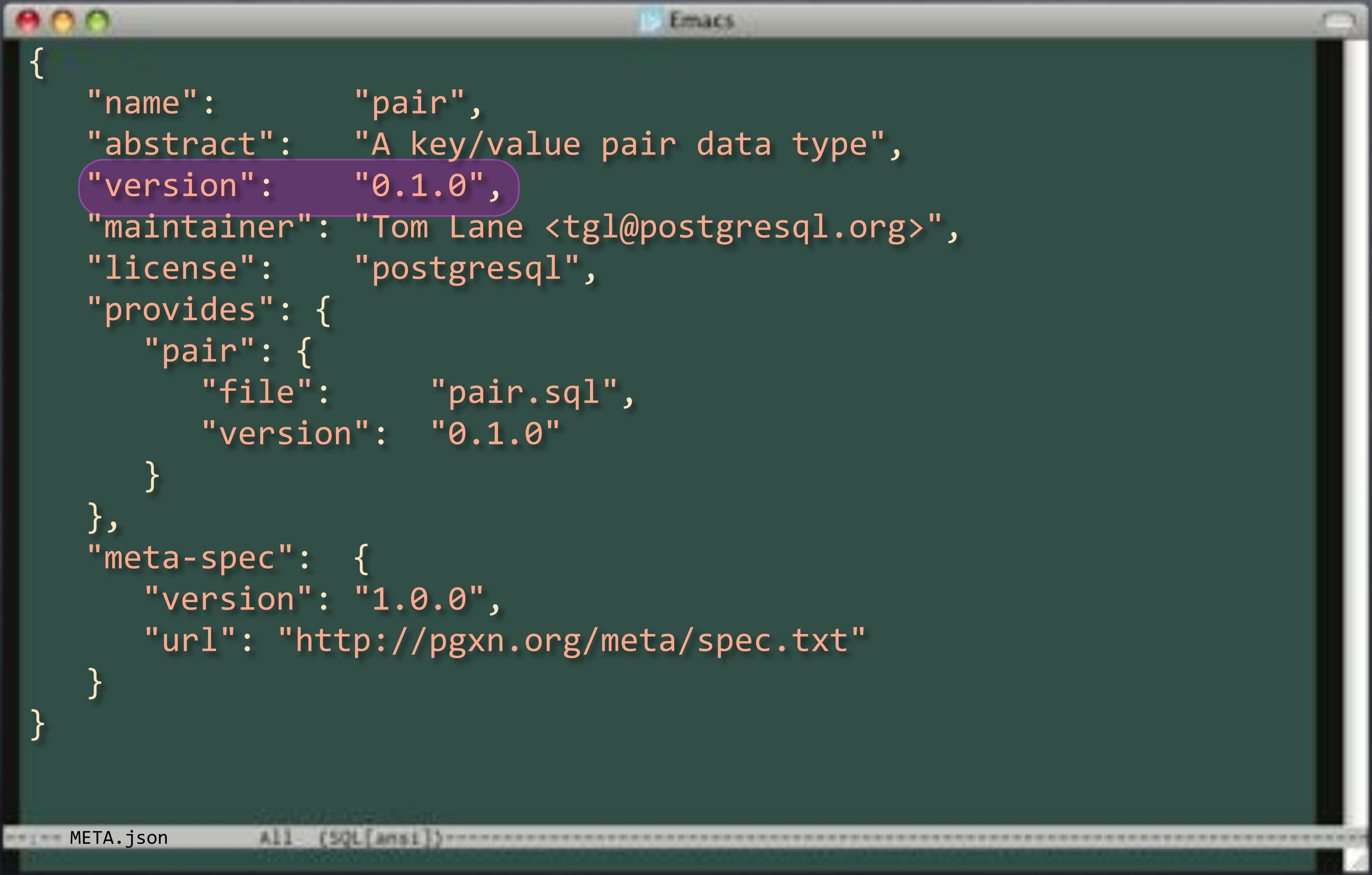


Emacs

```
{
   "name":       "pair",
   "abstract":   "A key/value pair data type",
   "version":    "0.1.0",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":    "postgresql",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

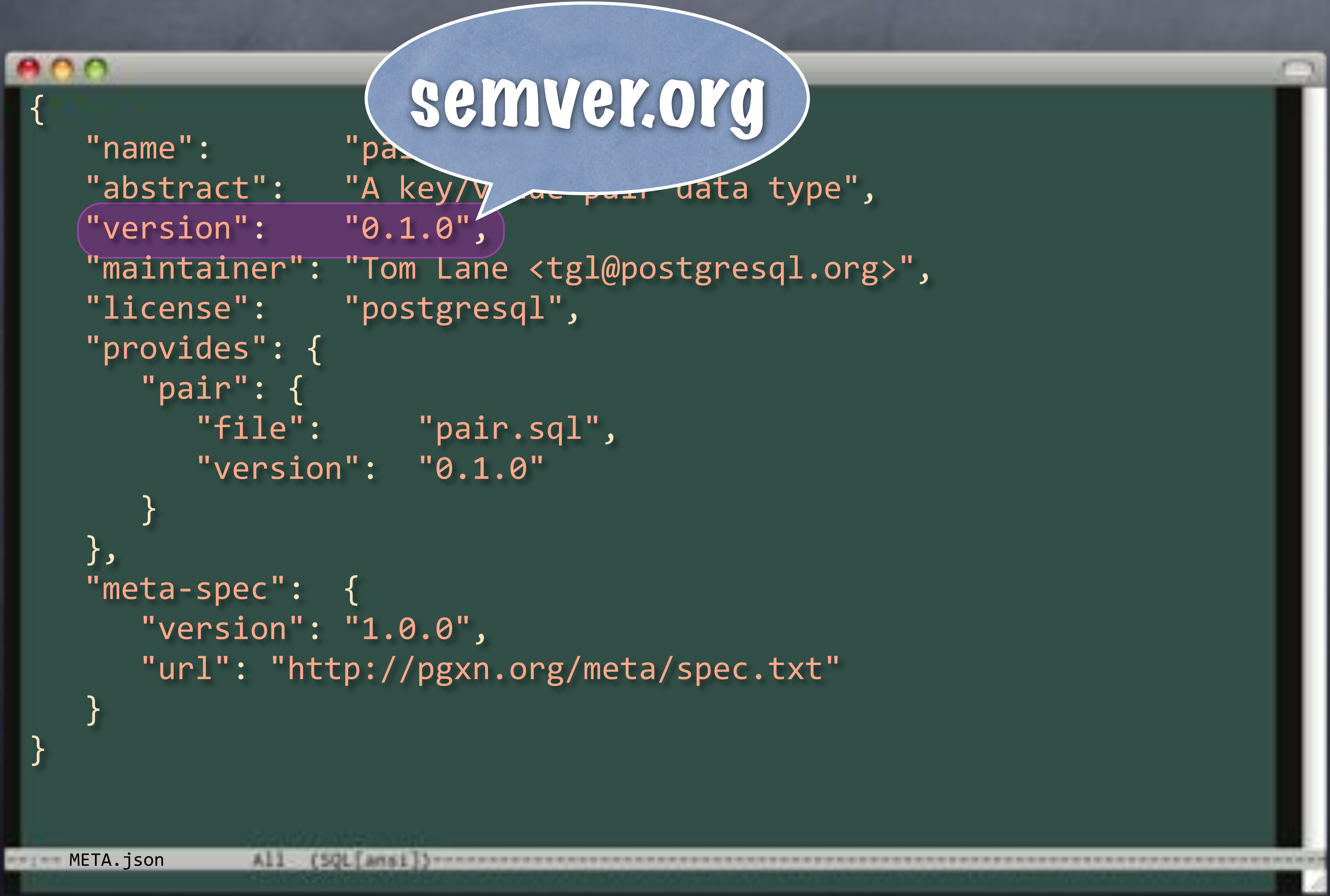
```
--:--  META.json        All   (SQL[ansi])------------------------------------------------
```

META.json



```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.0",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

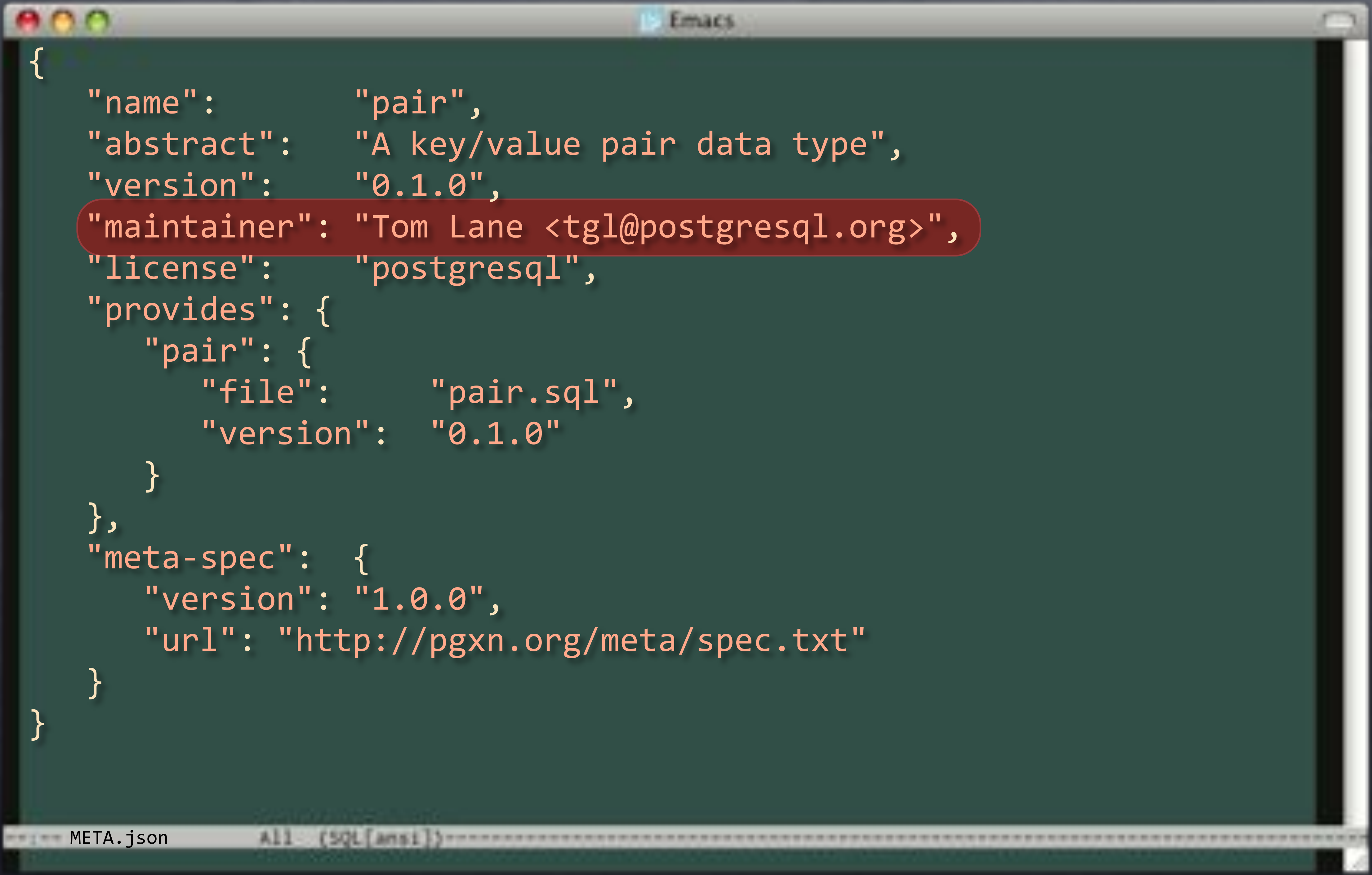
META.json



```
{
   "name":        "pa
   "abstract":    "A key/v     ue pair data type",
   "version":     "0.1.0",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

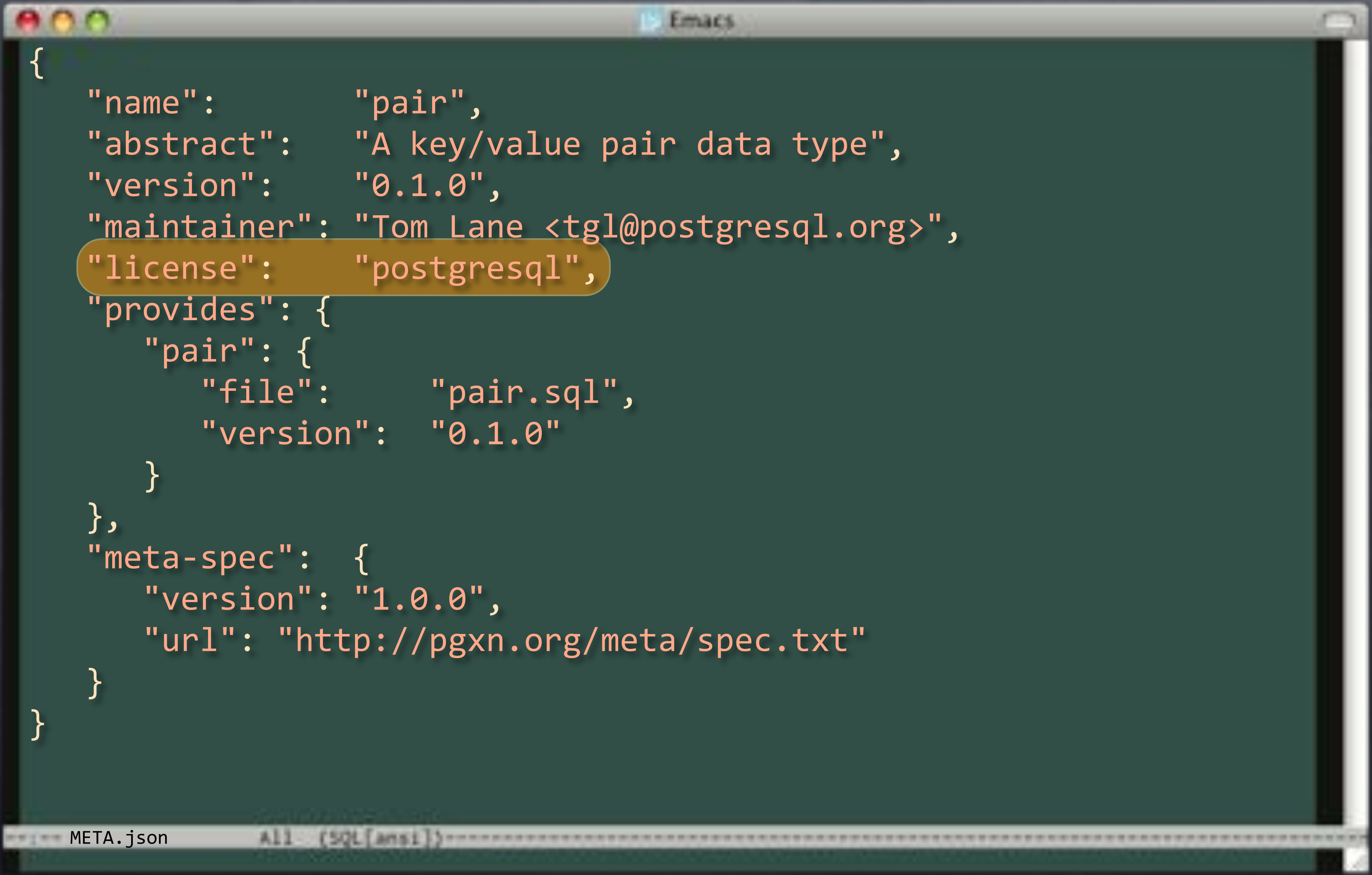
```
--:-- META.json        All  (SQL[ansi])-----------------------------------------------
```

META.json



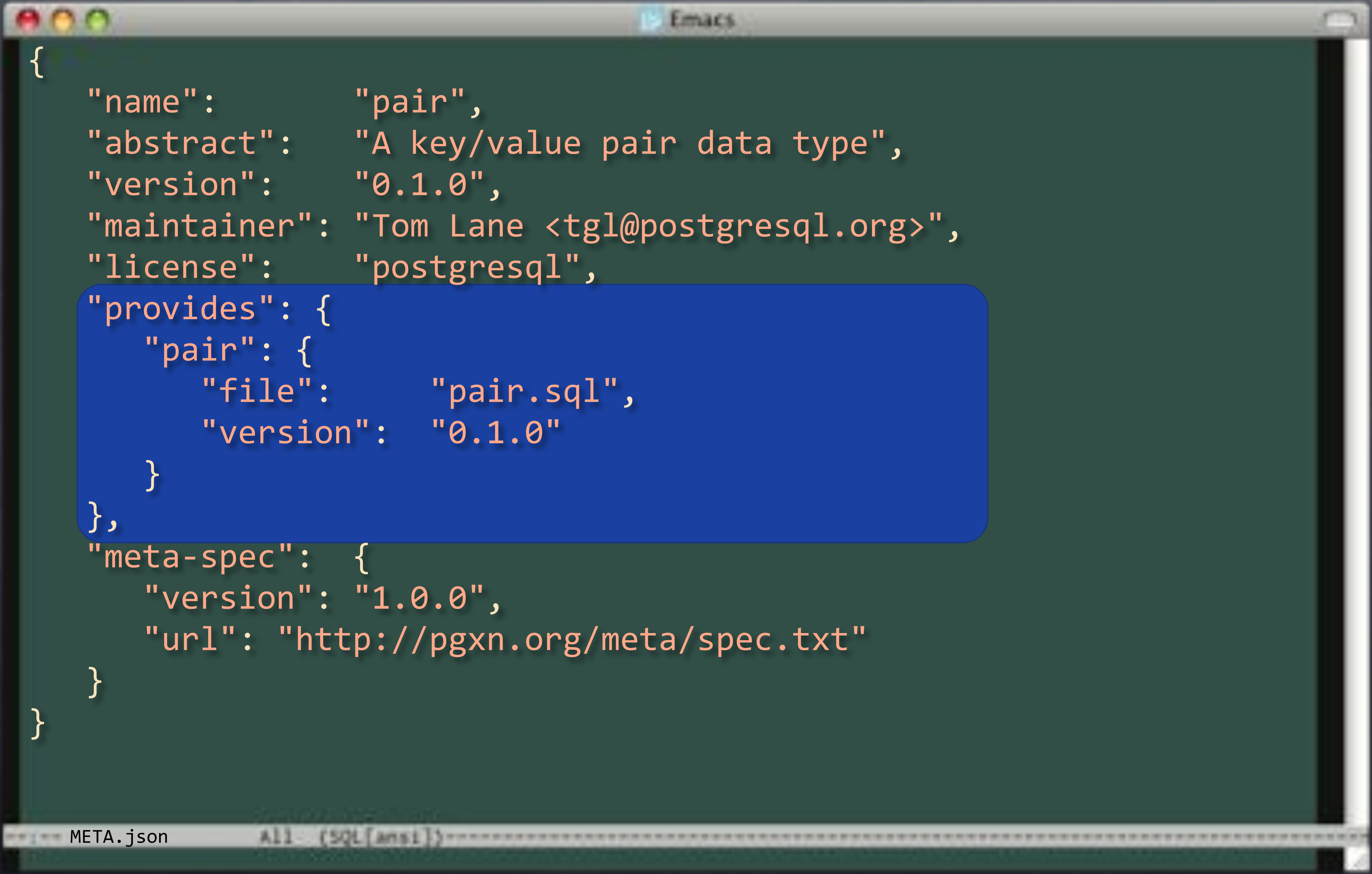
```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.0",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "provides": {
      "pair": {
         "file":      "pair.sql",
         "version":  "0.1.0"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

META.json



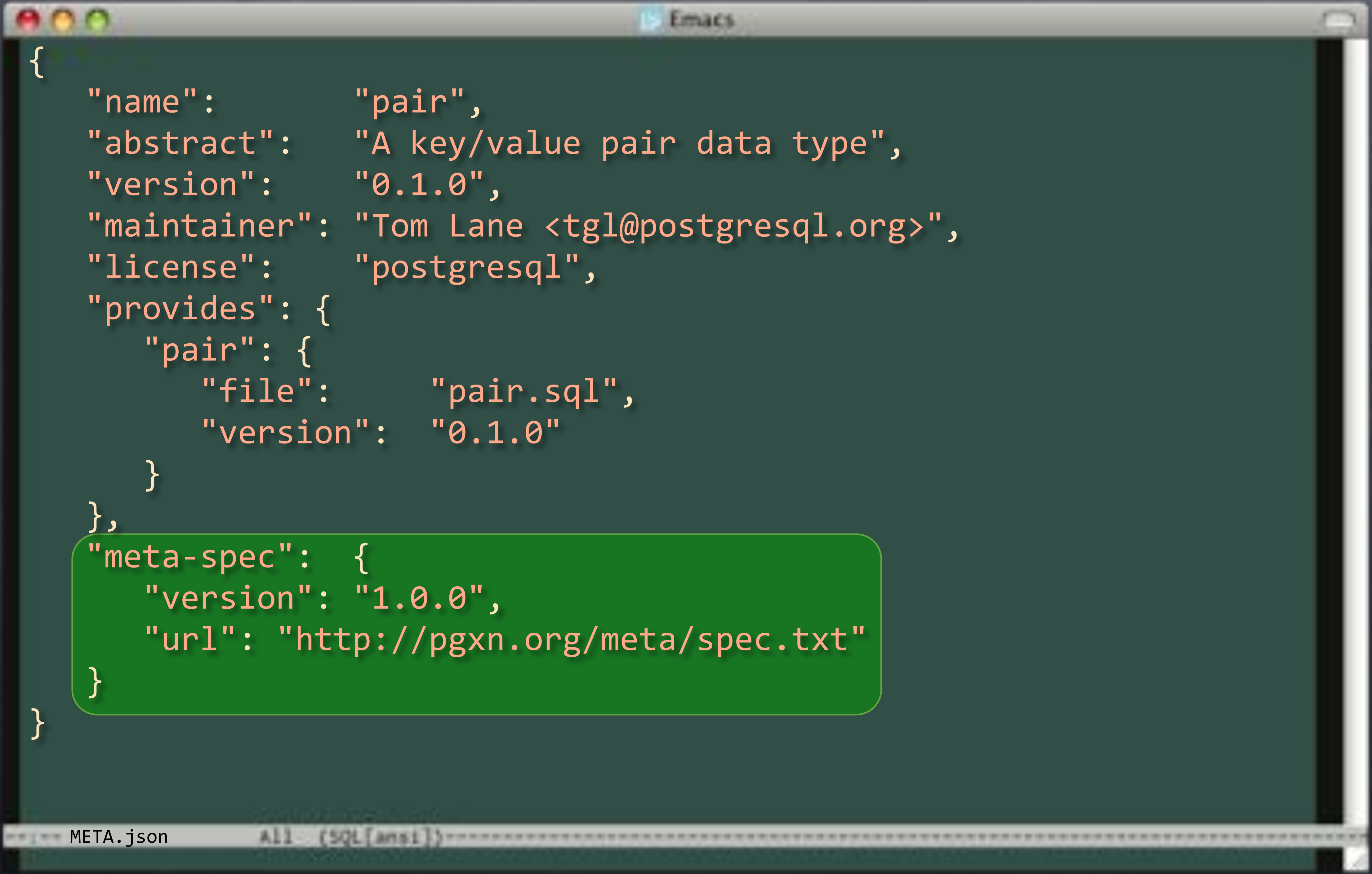
```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.0",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "provides": {
      "pair": {
         "file":      "pair.sql",
         "version":  "0.1.0"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

META.json



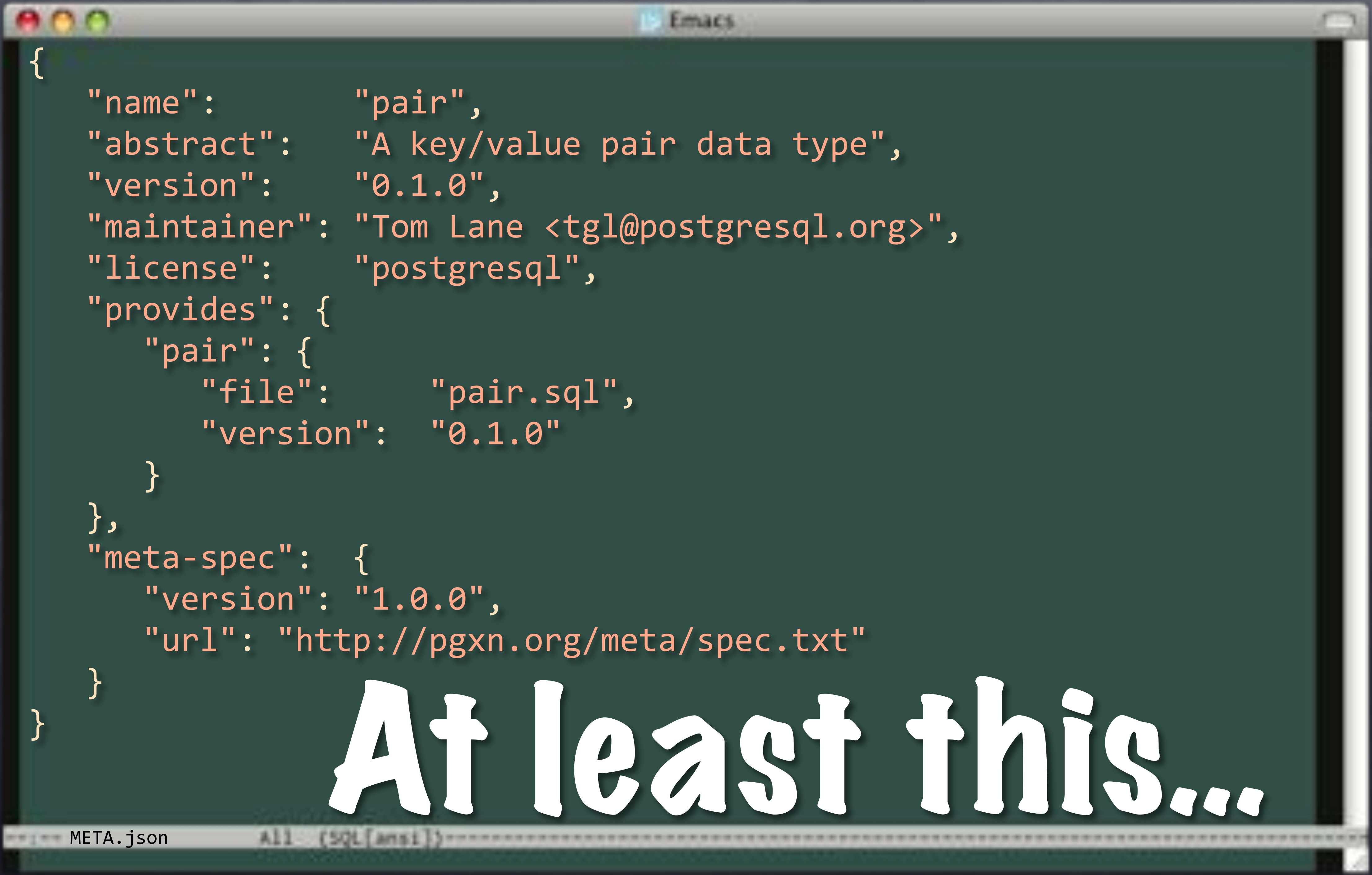
```
{
   "name":         "pair",
   "abstract":     "A key/value pair data type",
   "version":      "0.1.0",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":      "postgresql",
   "provides": {
      "pair": {
         "file":       "pair.sql",
         "version":   "0.1.0"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

META.json



```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.0",
   "maintainer":  "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

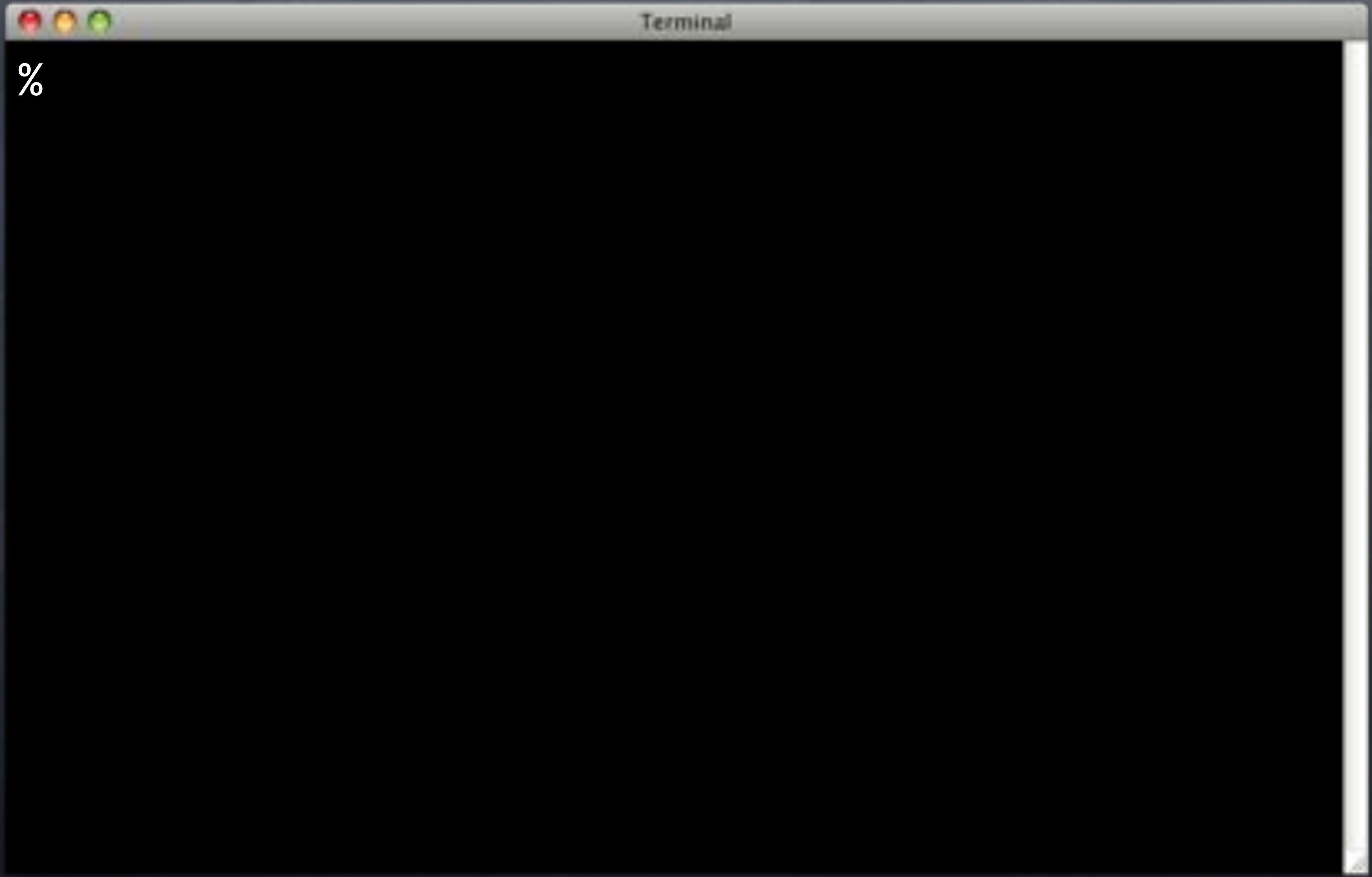
META.json



```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.0",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

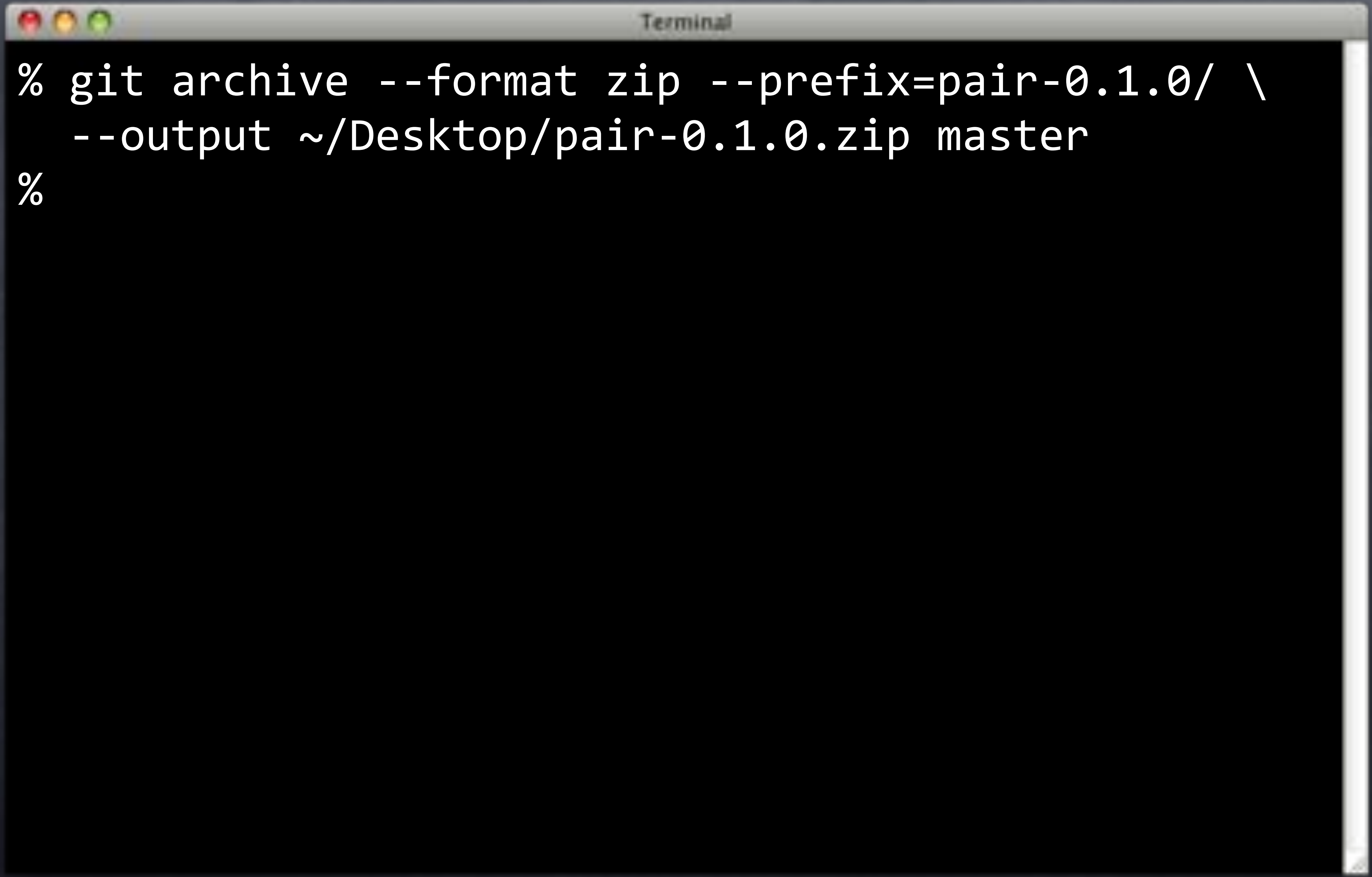
At least this...

Package it Up!



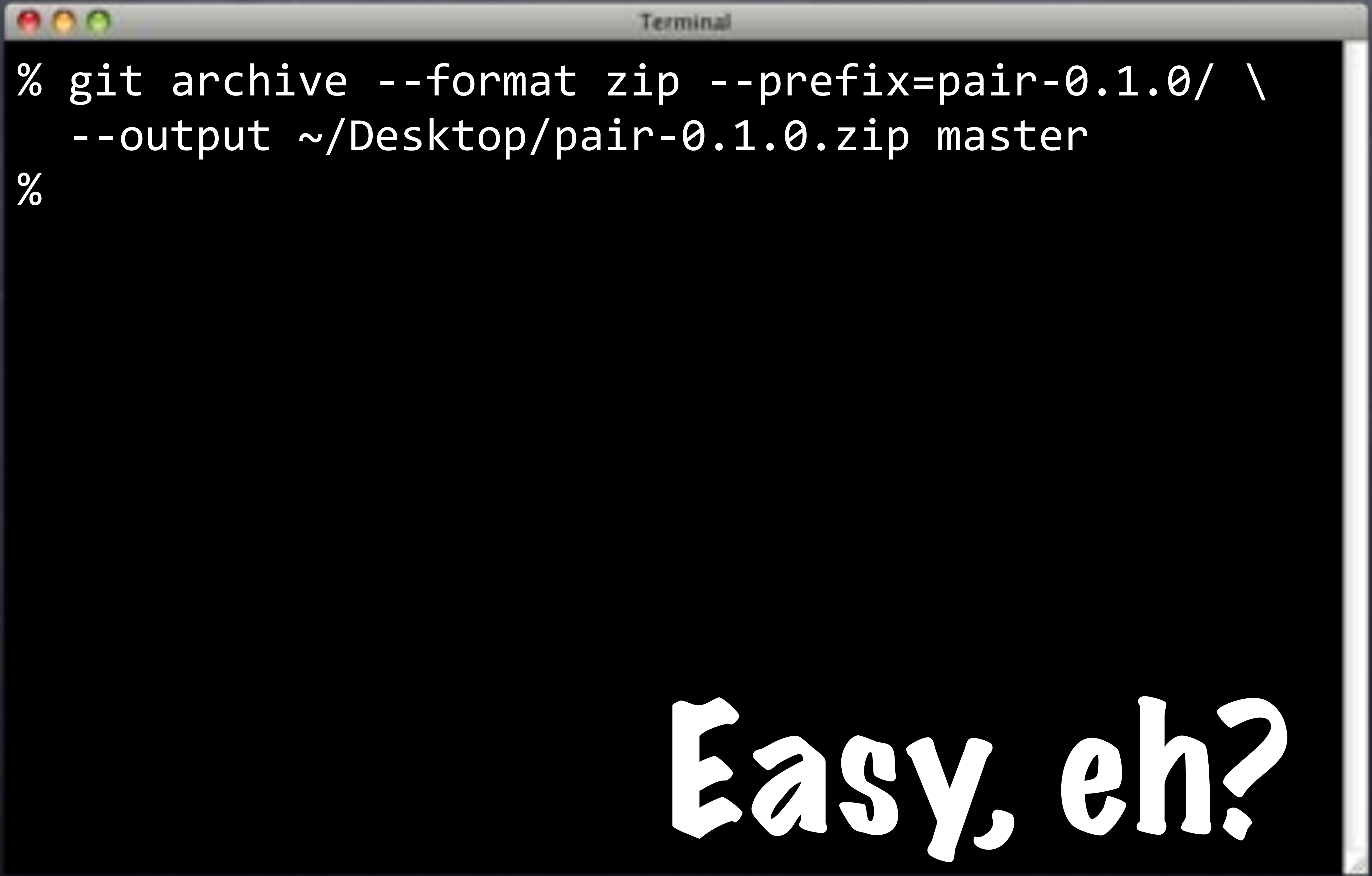
```
%
```

Package it Up!



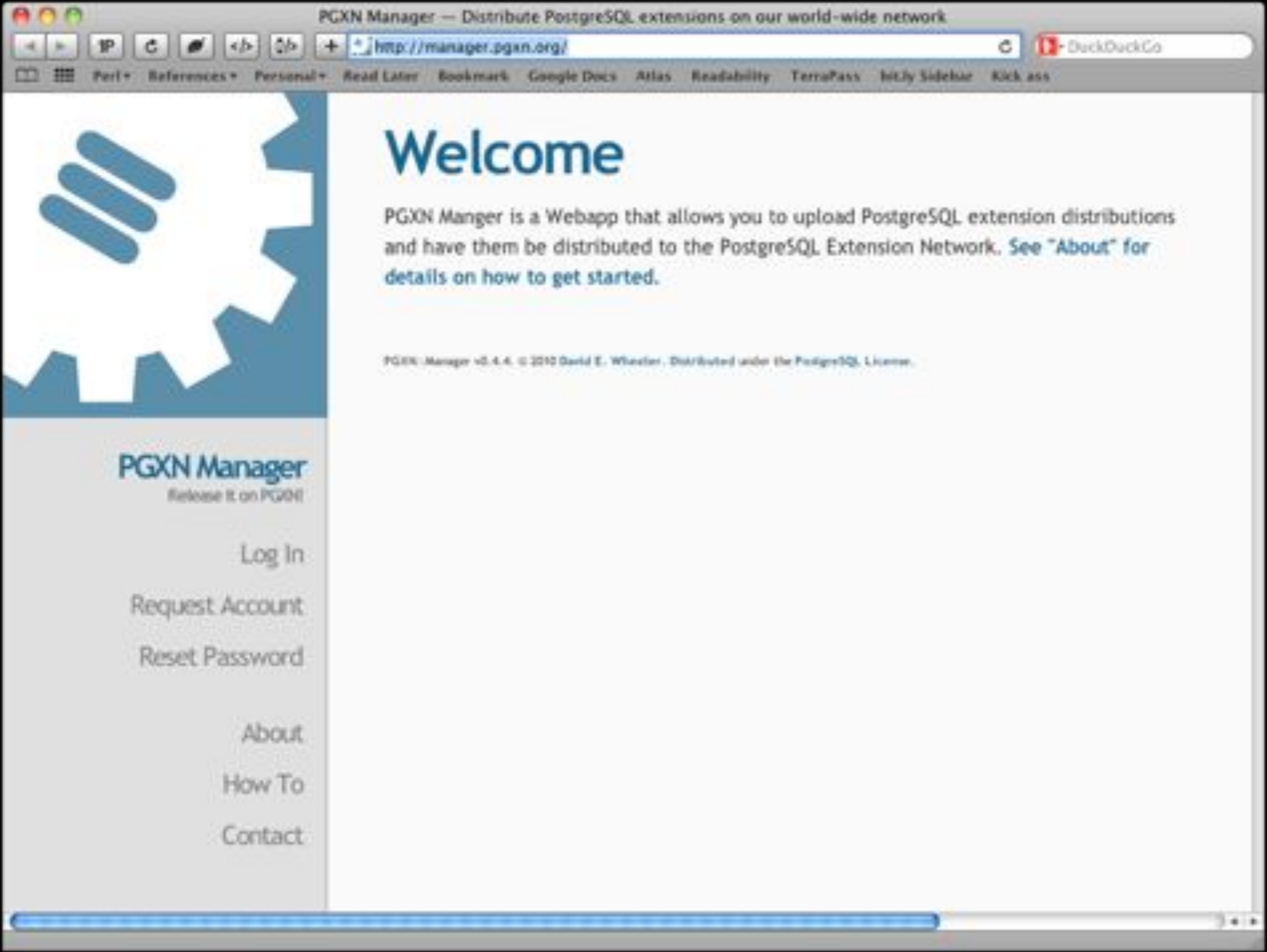
```
% git archive --format zip --prefix=pair-0.1.0/ \
  --output ~/Desktop/pair-0.1.0.zip master
%
```

Package it Up!



```
% git archive --format zip --prefix=pair-0.1.0/ \
  --output ~/Desktop/pair-0.1.0.zip master
%
```

Easy, eh?

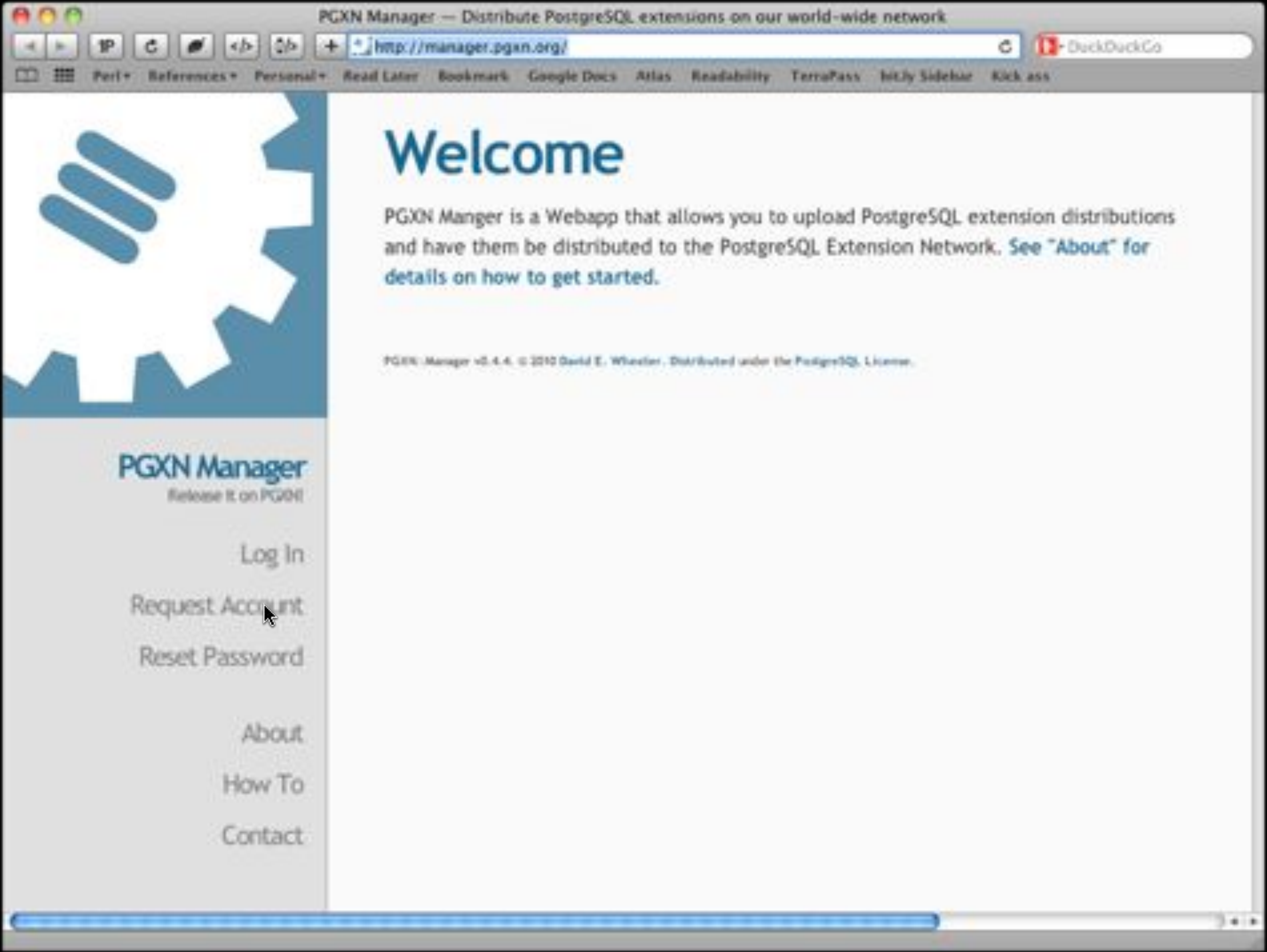

PGXN Manager — Distribute PostgreSQL extensions on our world-wide network

http://manager.pgxn.org/

Welcome

PGXN Manger is a Webapp that allows you to upload PostgreSQL extension distributions and have them be distributed to the PostgreSQL Extension Network. See "About" for details on how to get started.

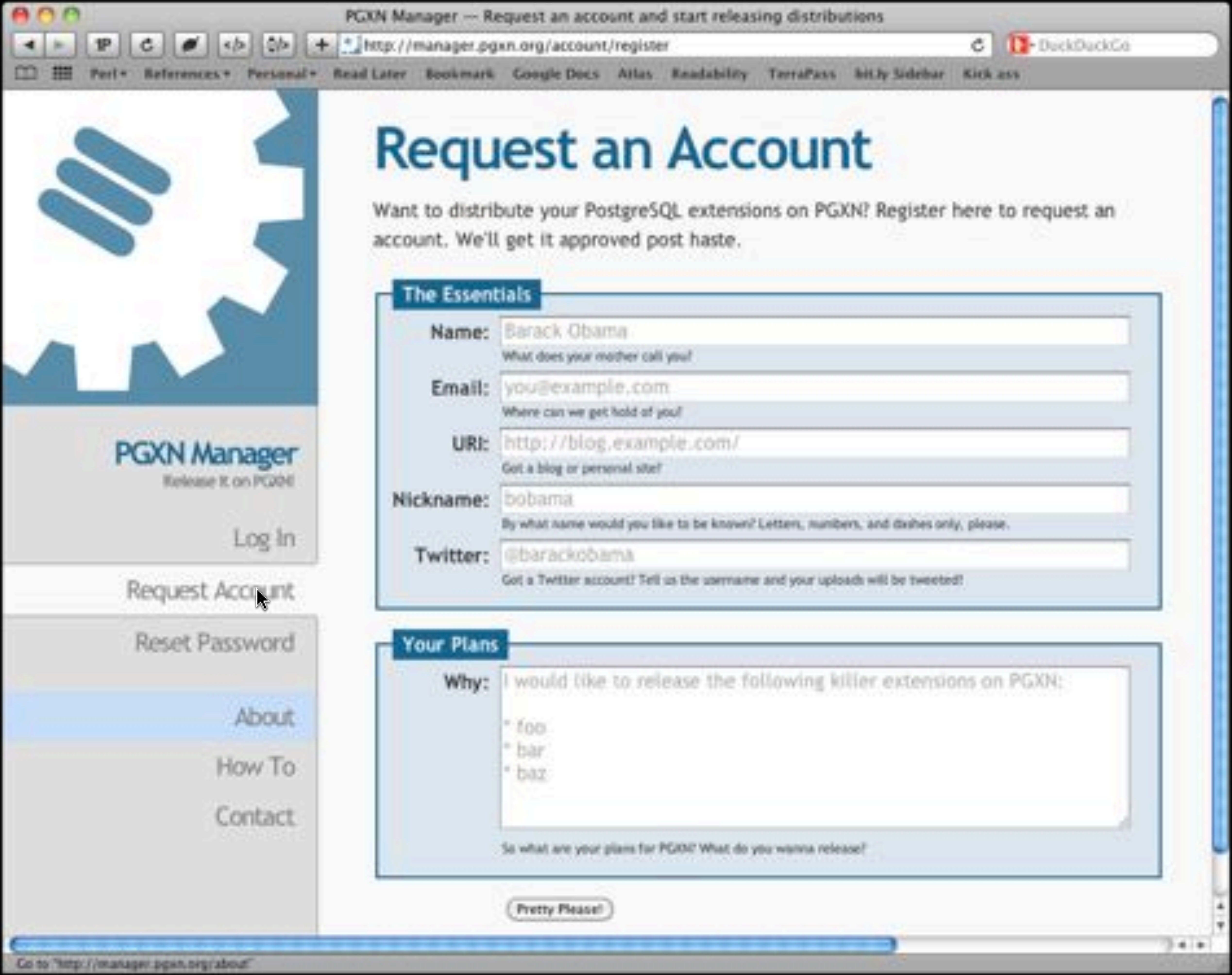
PGXN::Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.


PGXN Manager — Distribute PostgreSQL extensions on our world-wide network
http://manager.pgxn.org/
DuckDuckGo
Perl ▾ References ▾ Personal ▾ Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass
PGXN Manager
Release It on PGXN!
Log In
Request Account
Reset Password
About
How To
Contact

Welcome

PGXN Manger is a Webapp that allows you to upload PostgreSQL extension distributions and have them be distributed to the PostgreSQL Extension Network. See "About" for details on how to get started.

PGXN::Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.



PGXN Manager — Request an account and start releasing distributions

http://manager.pgxn.org/account/register

Request an Account

Want to distribute your PostgreSQL extensions on PGXN? Register here to request an account. We'll get it approved post haste.

The Essentials

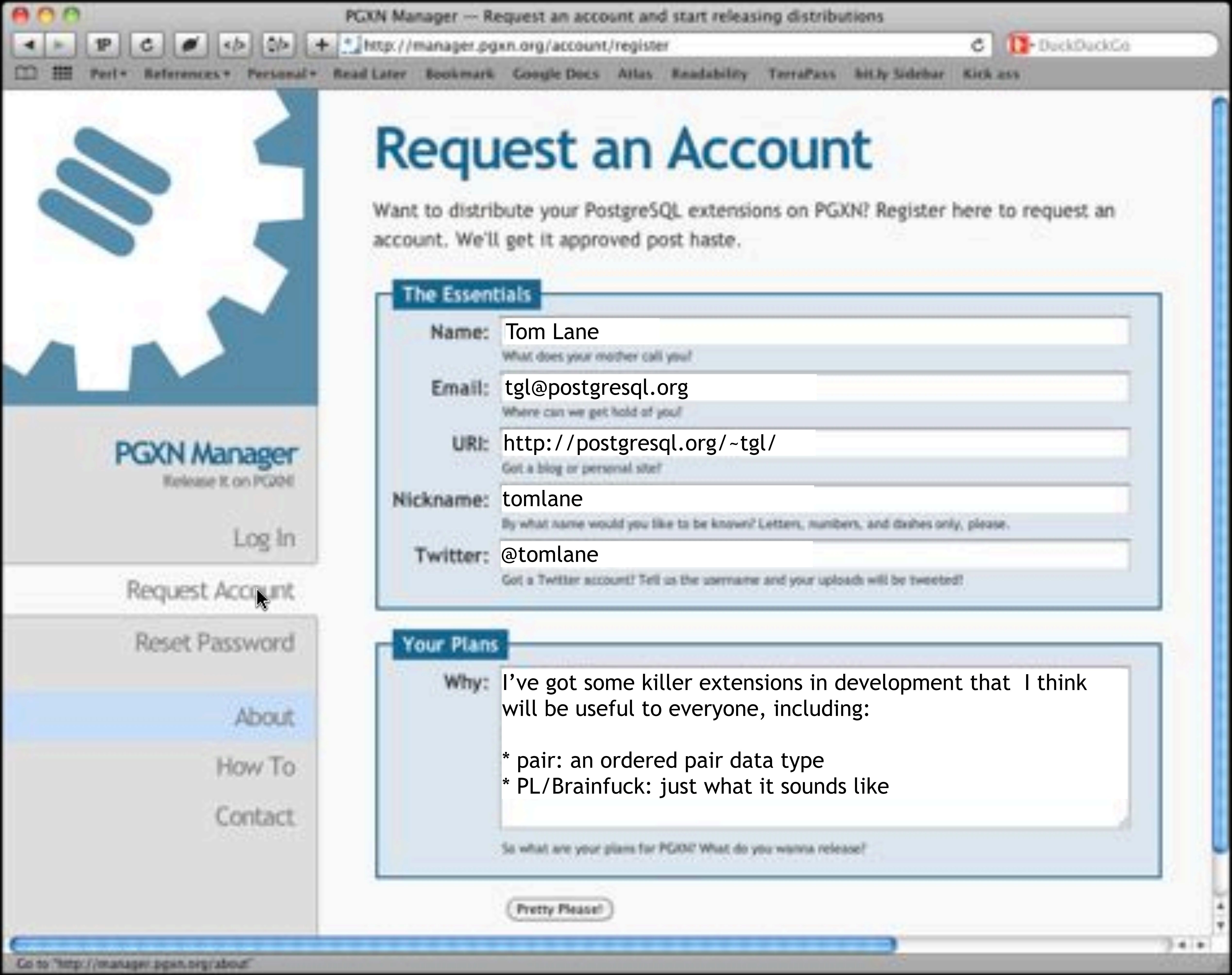
Name:	Barack Obama What does your mother call you?
Email:	you@example.com Where can we get hold of you?
URI:	http://blog.example.com/ Got a blog or personal site?
Nickname:	bobama By what name would you like to be known? Letters, numbers, and dashes only, please.
Twitter:	@barackobama Got a Twitter account? Tell us the username and your uploads will be tweeted!

Your Plans

Why: I would like to release the following killer extensions on PGXN:

* foo
* bar
* baz

So what are your plans for PGXN? What do you wanna release?



PGXN Manager — Request an account and start releasing distributions
http://manager.pgxn.org/account/register
PGXN Manager
Release It on PGXN!
Log In
Request Account
Reset Password
About
How To
Contact
Pretty Please!

Request an Account

Want to distribute your PostgreSQL extensions on PGXN? Register here to request an account. We'll get it approved post haste.

The Essentials

Name: Tom Lane
What does your mother call you?

Email: tgl@postgresql.org
Where can we get hold of you?

URI: http://postgresql.org/~tgl/
Got a blog or personal site?

Nickname: tomlane
By what name would you like to be known? Letters, numbers, and dashes only, please.

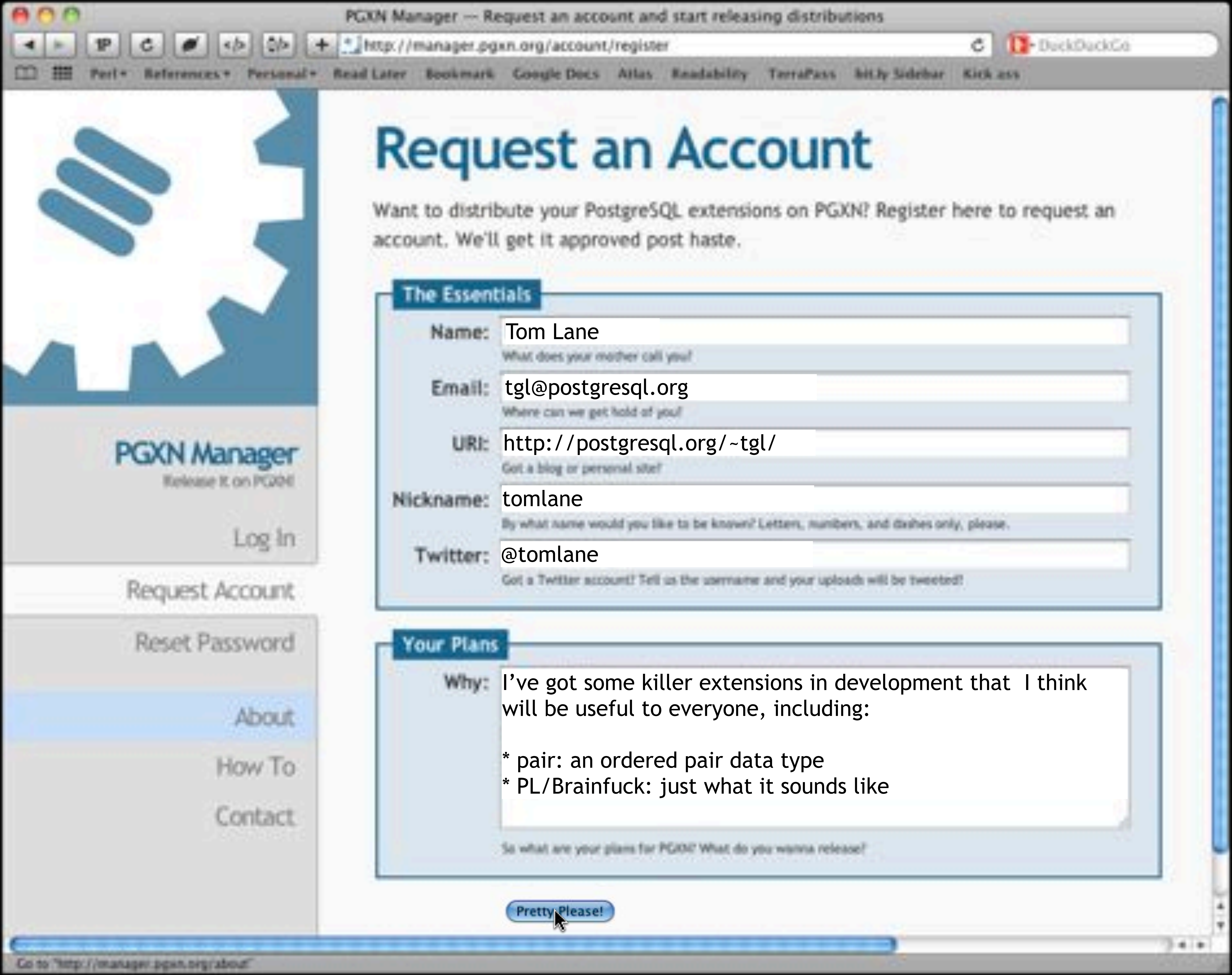
Twitter: @tomlane
Got a Twitter account? Tell us the username and your uploads will be tweeted!

Your Plans

Why: I've got some killer extensions in development that I think will be useful to everyone, including:

* pair: an ordered pair data type
* PL/Brainfuck: just what it sounds like

So what are your plans for PGXN? What do you wanna release?


PGXN Manager — Request an account and start releasing distributions
http://manager.pgxn.org/account/register
PGXN Manager
Release It on PGXN!
Log In
Request Account
Reset Password
About
How To
Contact
Pretty Please!

Request an Account

Want to distribute your PostgreSQL extensions on PGXN? Register here to request an account. We'll get it approved post haste.

The Essentials

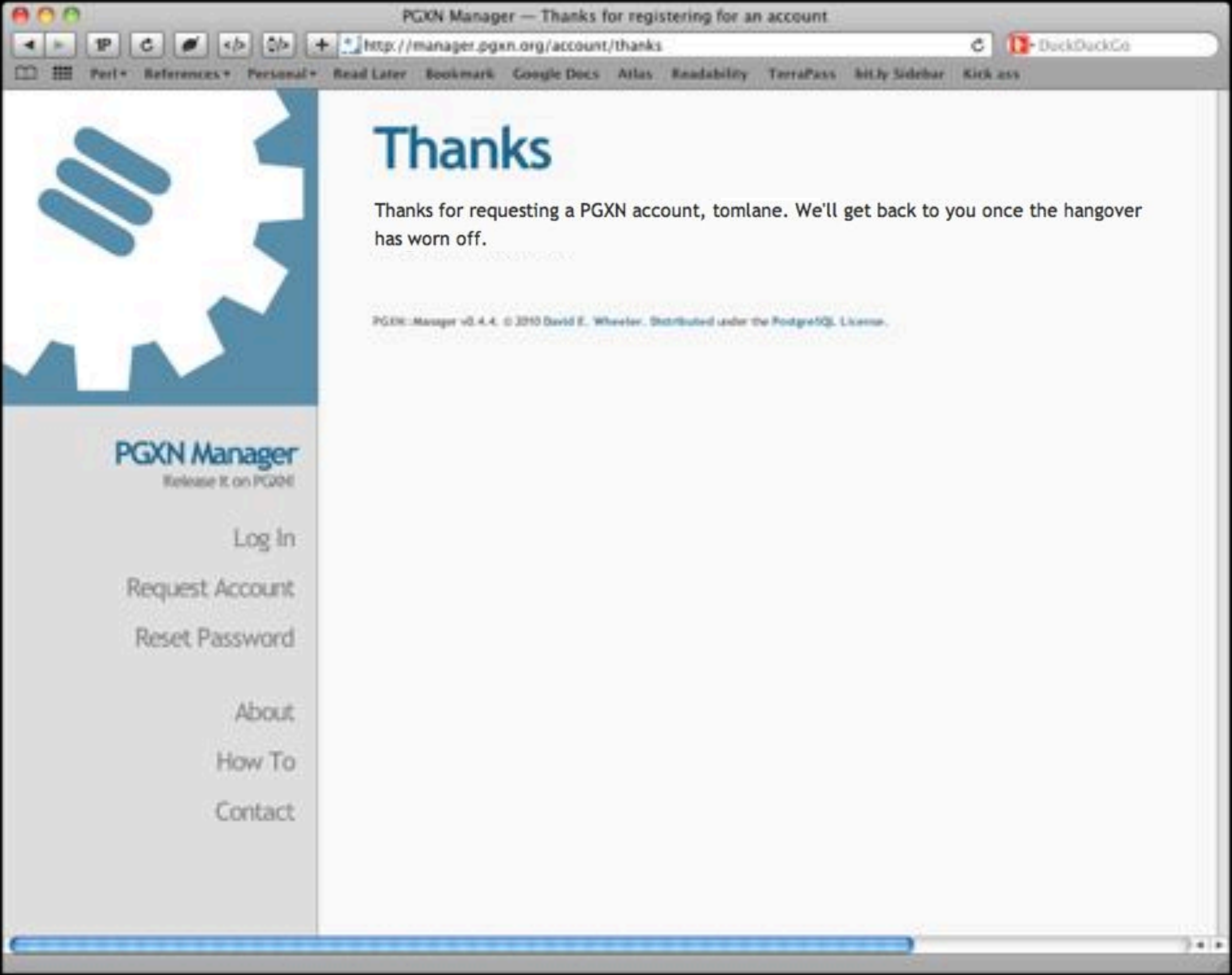
Name:	Tom Lane What does your mother call you?
Email:	tgl@postgresql.org Where can we get hold of you?
URI:	http://postgresql.org/~tgl/ Got a blog or personal site?
Nickname:	tomlane By what name would you like to be known? Letters, numbers, and dashes only, please.
Twitter:	@tomlane Got a Twitter account? Tell us the username and your uploads will be tweeted!

Your Plans

Why: I’ve got some killer extensions in development that I think will be useful to everyone, including:

* pair: an ordered pair data type
* PL/Brainfuck: just what it sounds like

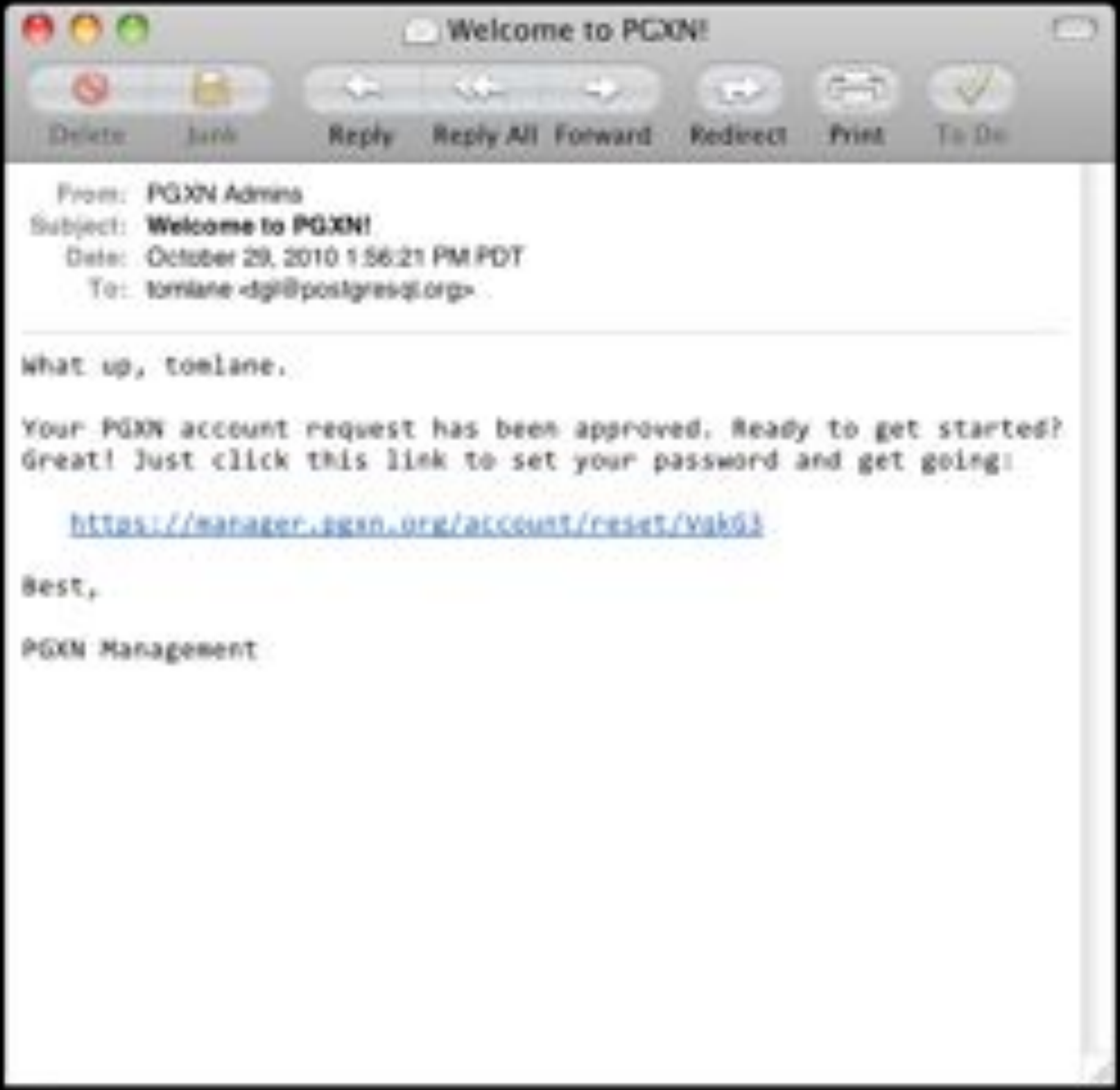
So what are your plans for PGXN? What do you wanna release?


PGXN Manager — Thanks for registering for an account
http://manager.pgxn.org/account/thanks
DuckDuckGo
Perl▾ References▾ Personal▾ Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass
PGXN Manager
Release It on PGXN!
Log In
Request Account
Reset Password
About
How To
Contact

Thanks

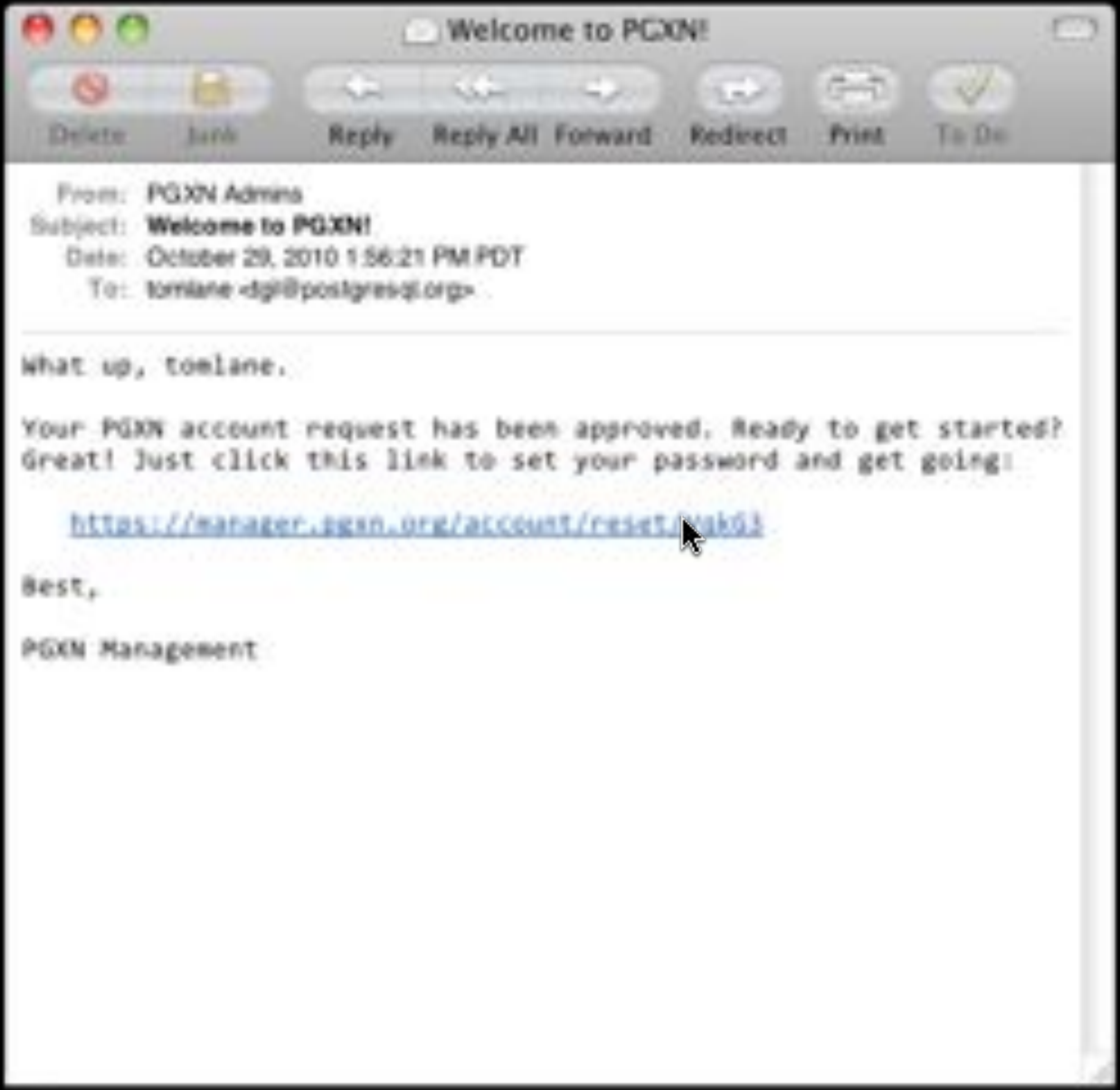
Thanks for requesting a PGXN account, tomlane. We'll get back to you once the hangover has worn off.

PGXN::Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.



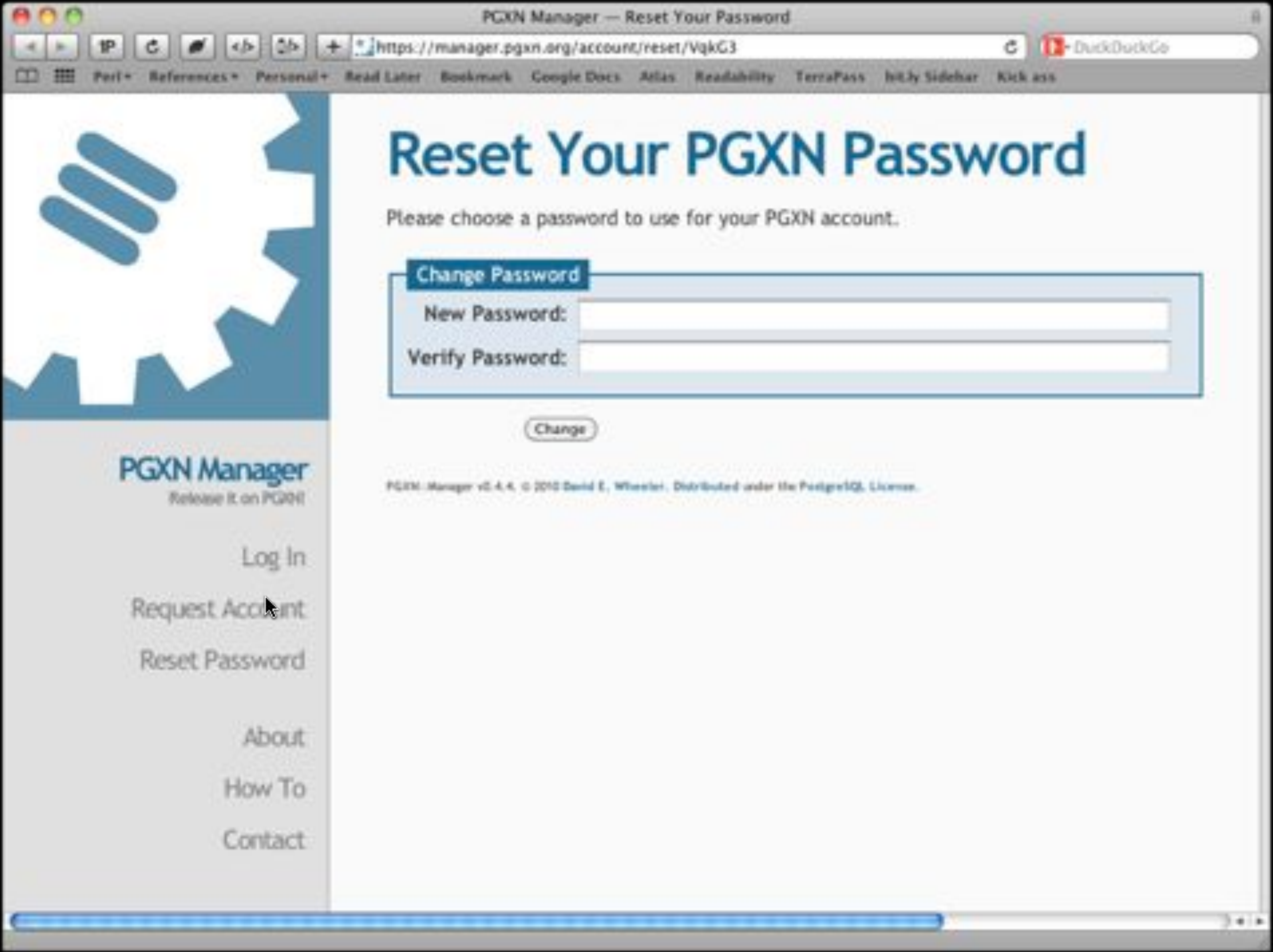
From: PGXN Admins
Subject: **Welcome to PGXN!**
Date: October 29, 2010 1:56:21 PM PDT
To: tomlane <dgl@postgresql.org>

```
What up, tomlane.

Your PGXN account request has been approved. Ready to get started?
Great! Just click this link to set your password and get going:

  https://manager.pgxn.org/account/reset/Vqk63

Best,

PGXN Management
```



From: PGXN Admins
Subject: **Welcome to PGXN!**
Date: October 29, 2010 1:56:21 PM PDT
To: tomlane <dgl@postgresql.org>

```
What up, tomlane.

Your PGXN account request has been approved. Ready to get started?
Great! Just click this link to set your password and get going:

  https://manager.pgxn.org/account/reset/wgk63

Best,

PGXN Management
```


PGXN Manager — Reset Your Password
https://manager.pgxn.org/account/reset/VqkG3
DuckDuckGo
Perl References Personal Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass
PGXN Manager
Release it on PGXN!
Log In
Request Account
Reset Password
About
How To
Contact

Reset Your PGXN Password

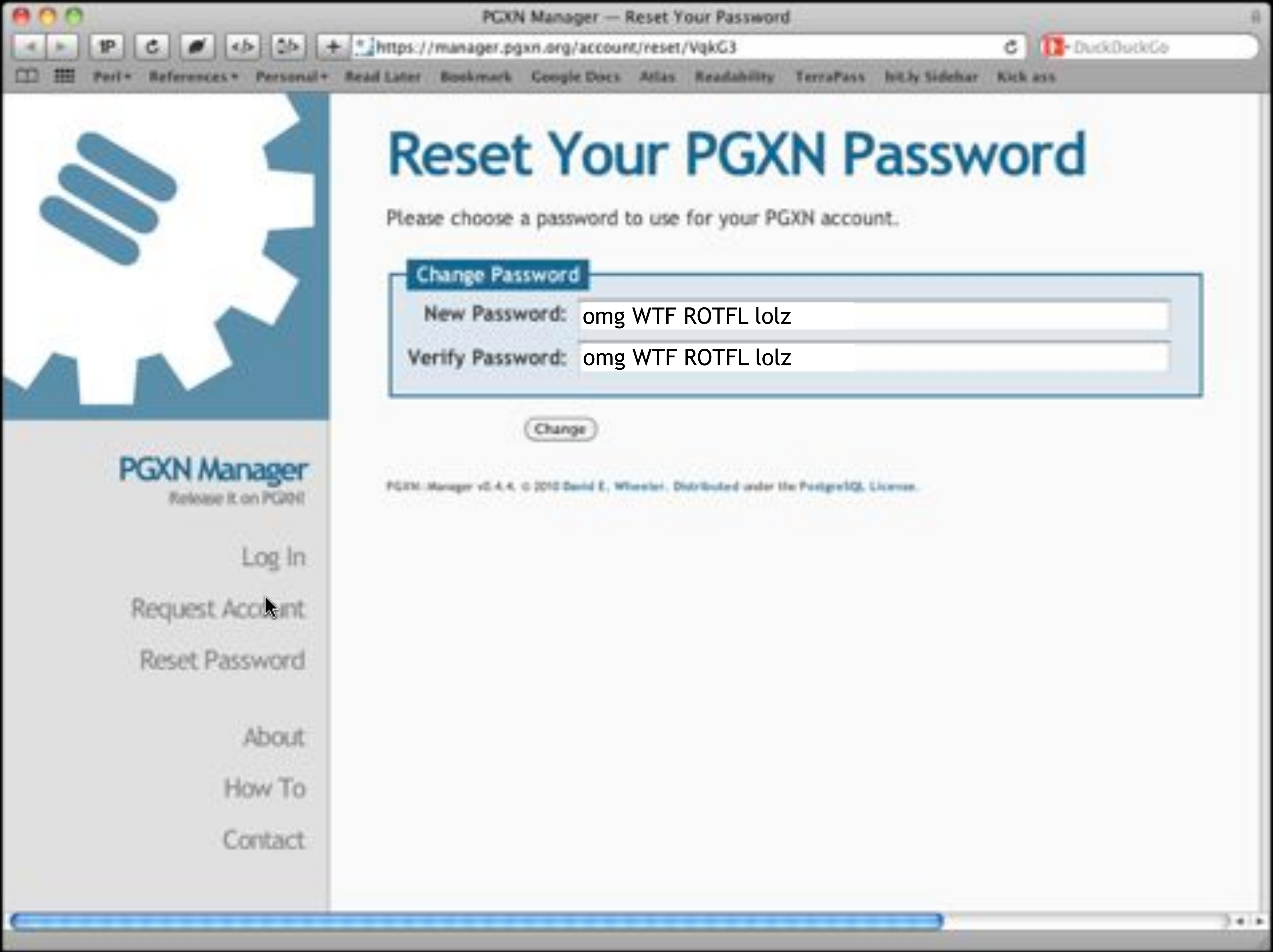
Please choose a password to use for your PGXN account.

Change Password

New Password:	
Verify Password:	

Change

PGXN Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.



PGXN Manager — Reset Your Password

https://manager.pgxn.org/account/reset/VqkG3

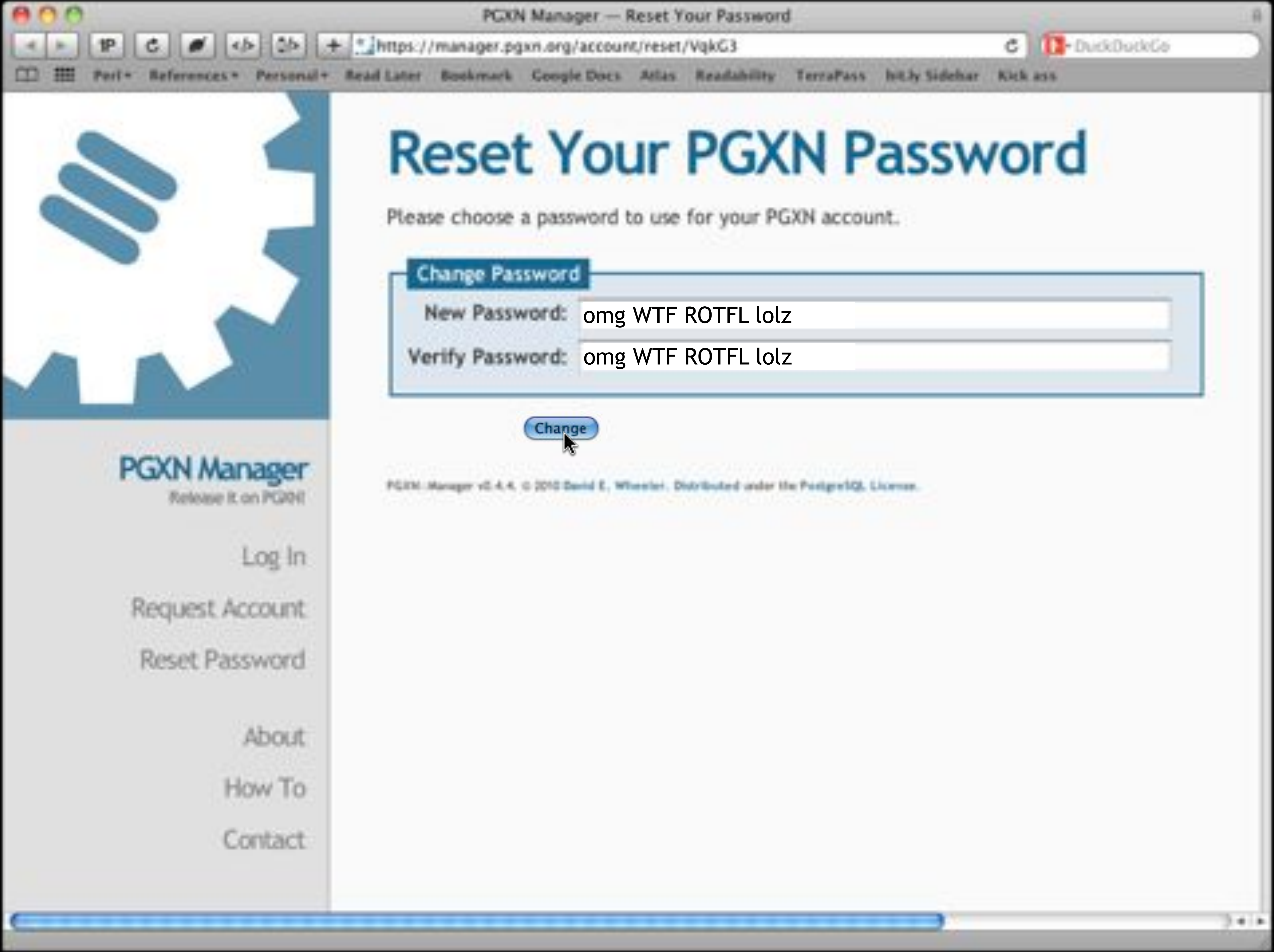
Reset Your PGXN Password

Please choose a password to use for your PGXN account.

Change Password

New Password:	omg WTF ROTFL lolz
Verify Password:	omg WTF ROTFL lolz

PGXN Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.


PGXN Manager — Reset Your Password

https://manager.pgxn.org/account/reset/VqkG3

Reset Your PGXN Password

Please choose a password to use for your PGXN account.

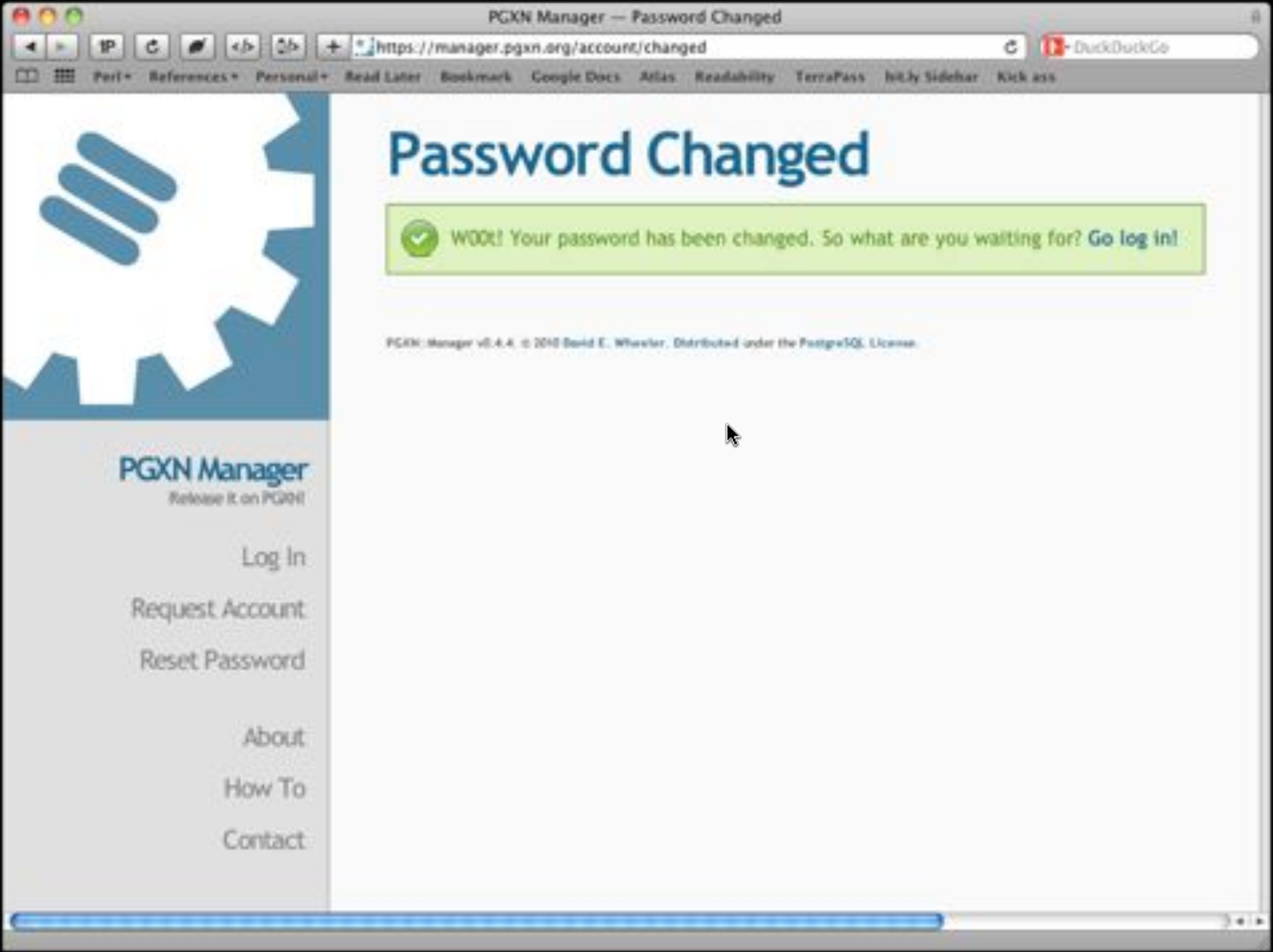
Change Password

New Password: omg WTF ROTFL lolz

Verify Password: omg WTF ROTFL lolz

Change

PGXN Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.

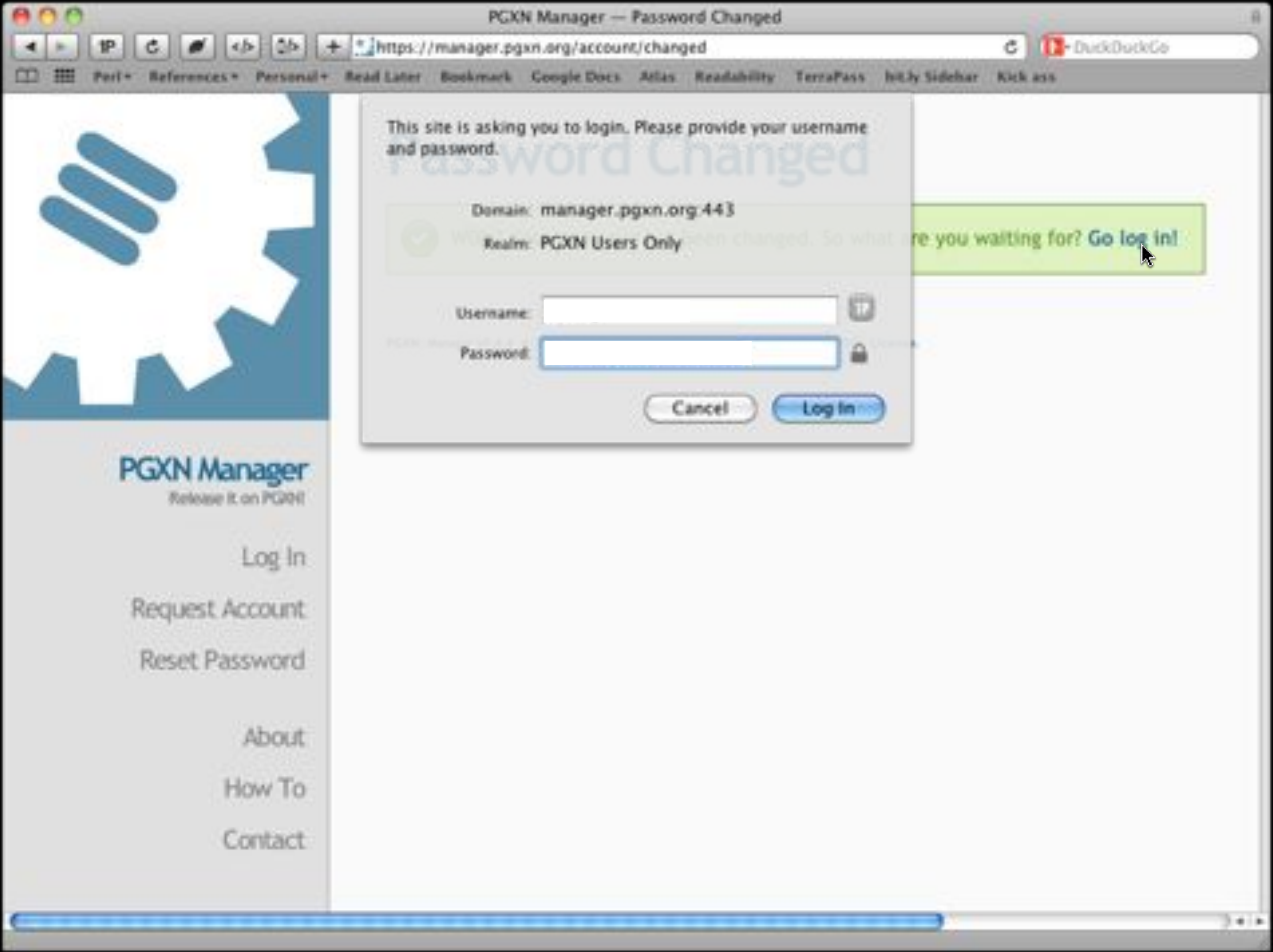


Password Changed

WOOt! Your password has been changed. So what are you waiting for? **Go log in!**

PGXN::Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.

PGXN Manager
Release it on PGXN!



PGXN Manager — Password Changed

https://manager.pgxn.org/account/changed

This site is asking you to login. Please provide your username and password.

Domain: manager.pgxn.org:443

Realm: PGXN Users Only

Username:

Password:

Cancel | Log In

...re you waiting for? Go log in!

PGXN Manager

Release it on PGXN!

Log In

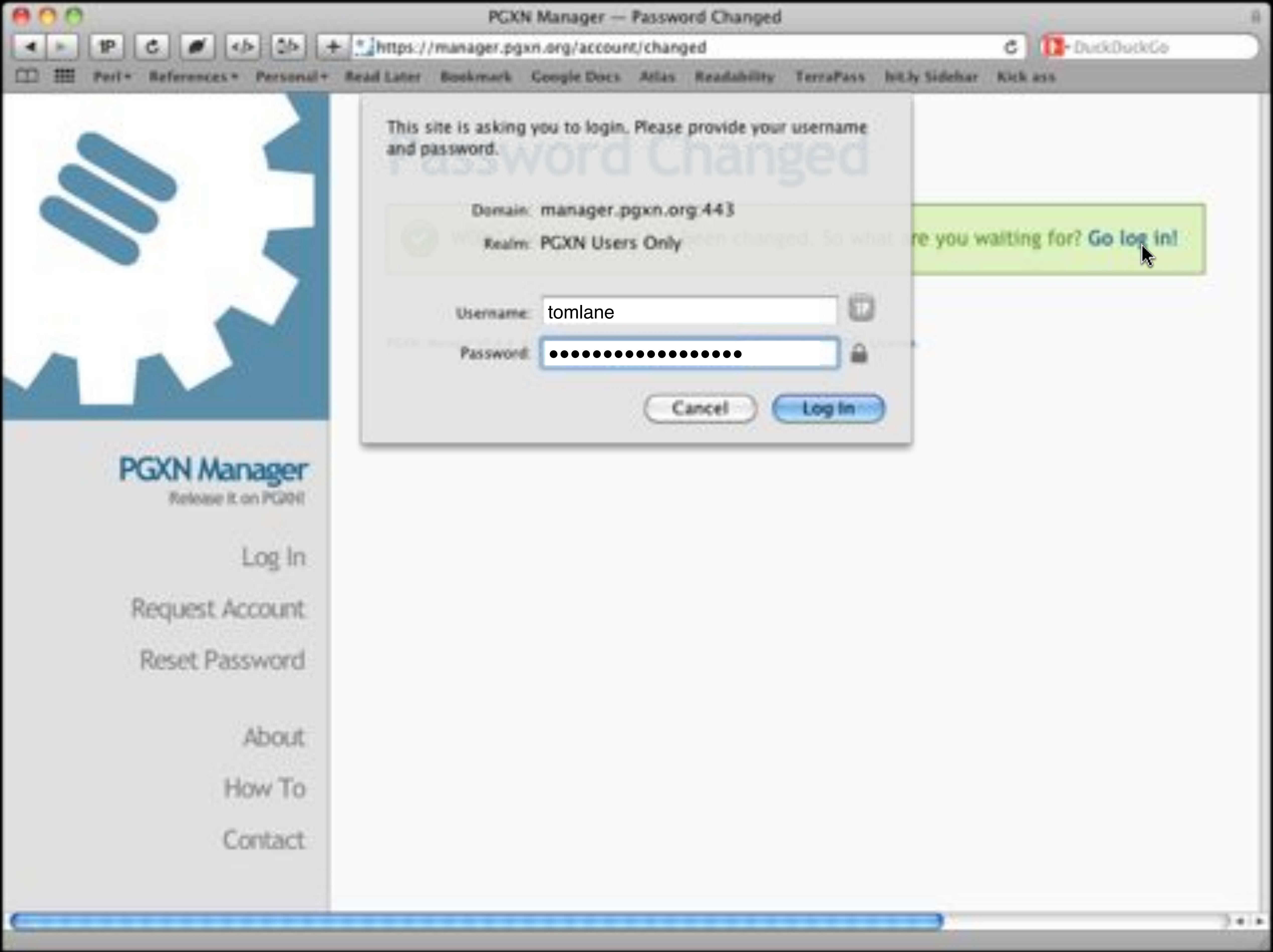
Request Account

Reset Password

About

How To

Contact



PGXN Manager — Password Changed

https://manager.pgxn.org/account/changed

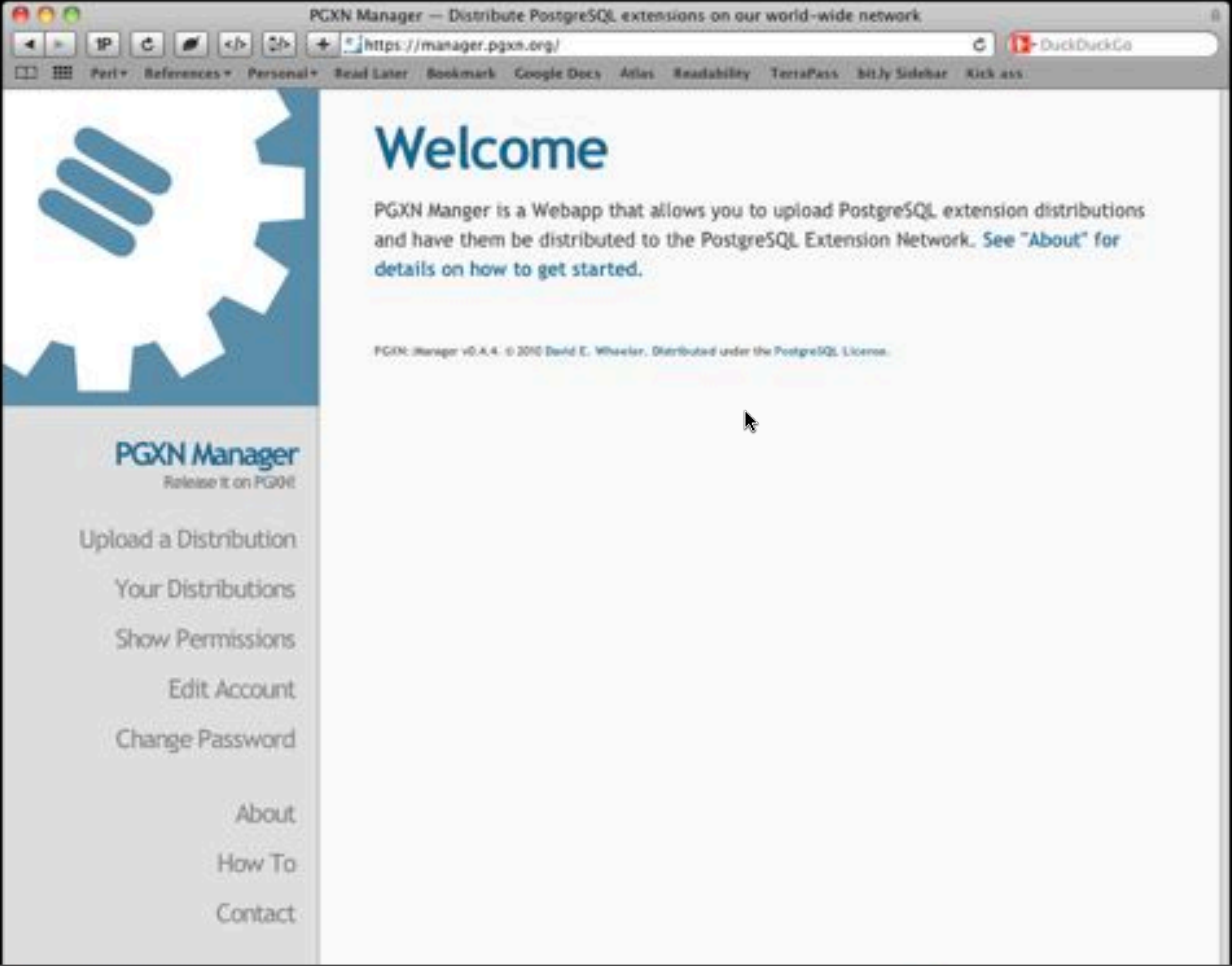
This site is asking you to login. Please provide your username and password.

Domain: manager.pgxn.org:443

Realm: PGXN Users Only

Username: tomlane

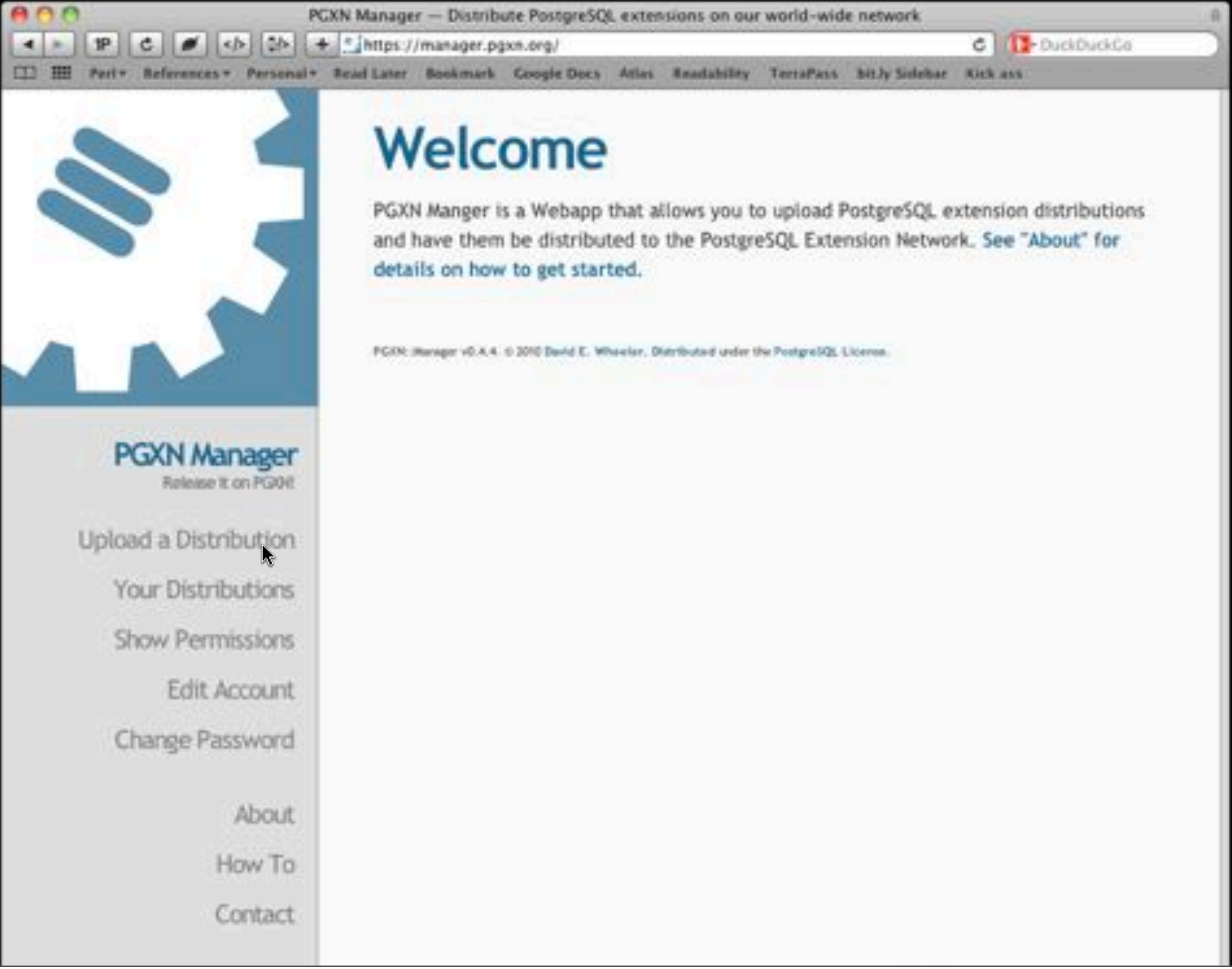
Password: ••••••••••••••••••



Welcome

PGXN Manger is a Webapp that allows you to upload PostgreSQL extension distributions and have them be distributed to the PostgreSQL Extension Network. See "About" for details on how to get started.

PGXN::Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.


PGXN Manager — Distribute PostgreSQL extensions on our world-wide network
https://manager.pgxn.org/
DuckDuckGo
Perl▾ References▾ Personal▾ Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass

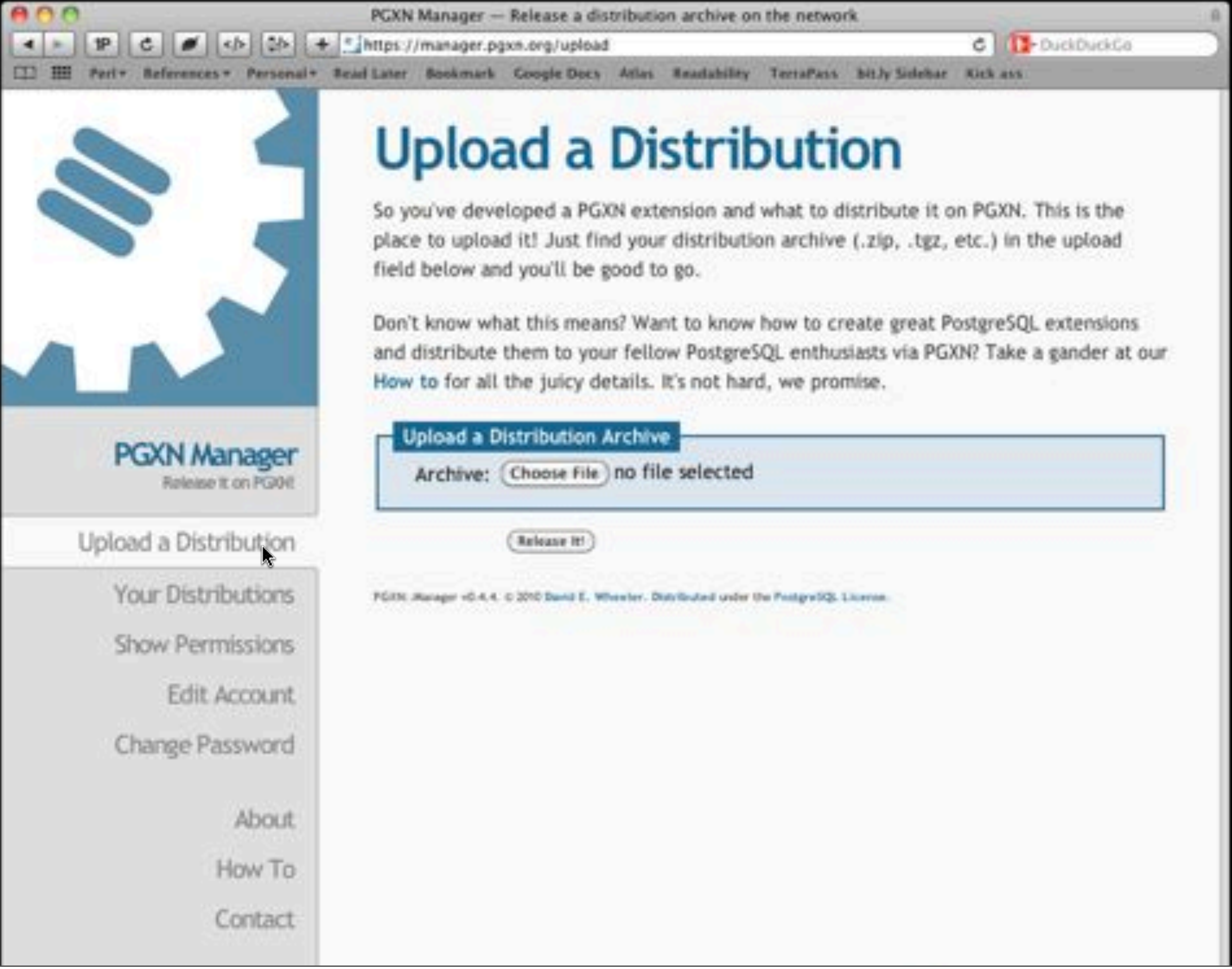
Welcome

PGXN Manger is a Webapp that allows you to upload PostgreSQL extension distributions and have them be distributed to the PostgreSQL Extension Network. See "About" for details on how to get started.

PGXN::Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.

PGXN Manager
Release It on PGXN!

- Upload a Distribution
- Your Distributions
- Show Permissions
- Edit Account
- Change Password
- About
- How To
- Contact


PGXN Manager — Release a distribution archive on the network
https://manager.pgxn.org/upload
DuckDuckGo
Perl▾ References▾ Personal▾ Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass
PGXN Manager
Release it on PGXN!
Upload a Distribution
Your Distributions
Show Permissions
Edit Account
Change Password
About
How To
Contact

Upload a Distribution

So you've developed a PGXN extension and what to distribute it on PGXN. This is the place to upload it! Just find your distribution archive (.zip, .tgz, etc.) in the upload field below and you'll be good to go.

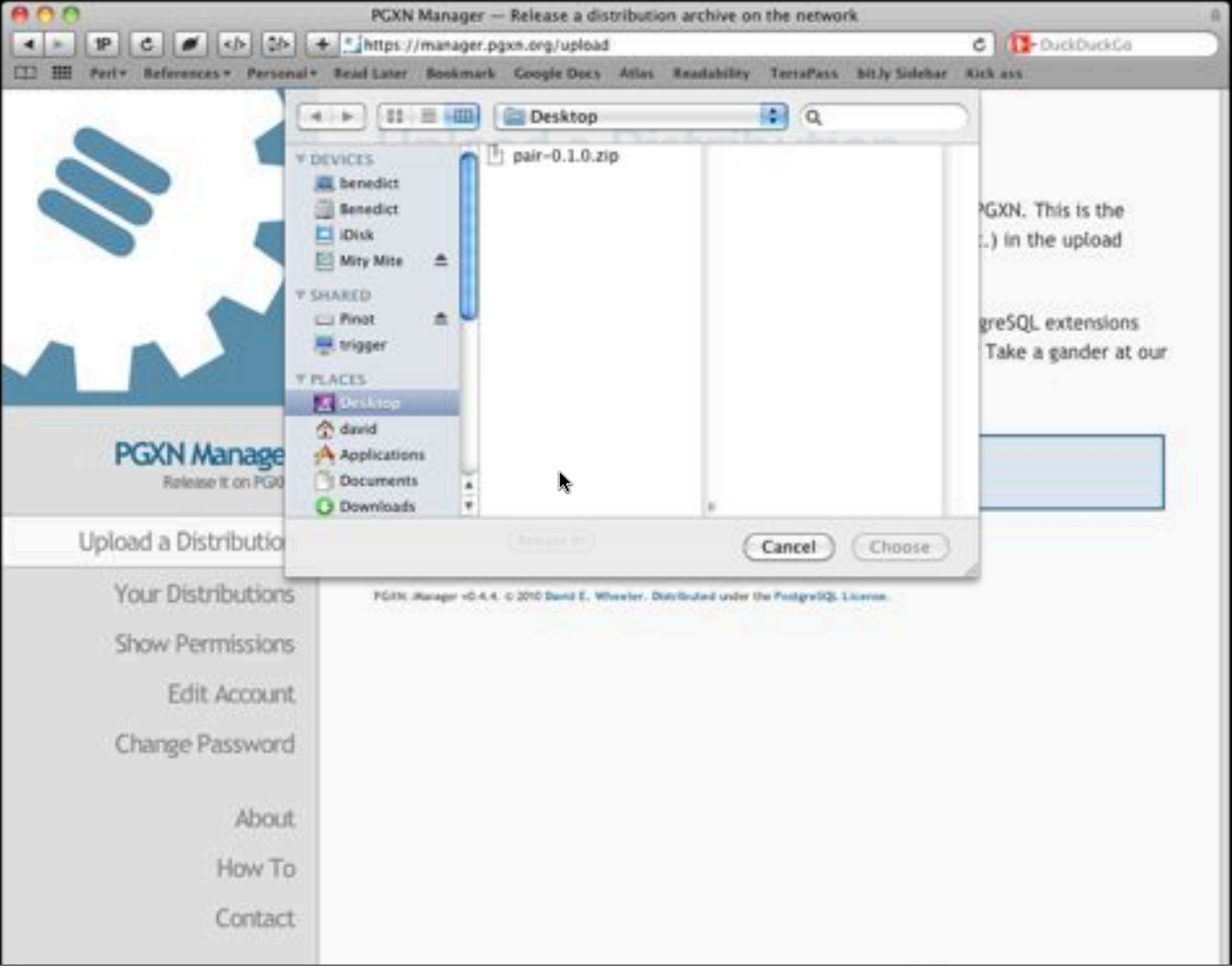
Don't know what this means? Want to know how to create great PostgreSQL extensions and distribute them to your fellow PostgreSQL enthusiasts via PGXN? Take a gander at our How to for all the juicy details. It's not hard, we promise.

Upload a Distribution Archive

Archive: Choose File no file selected

Release it!

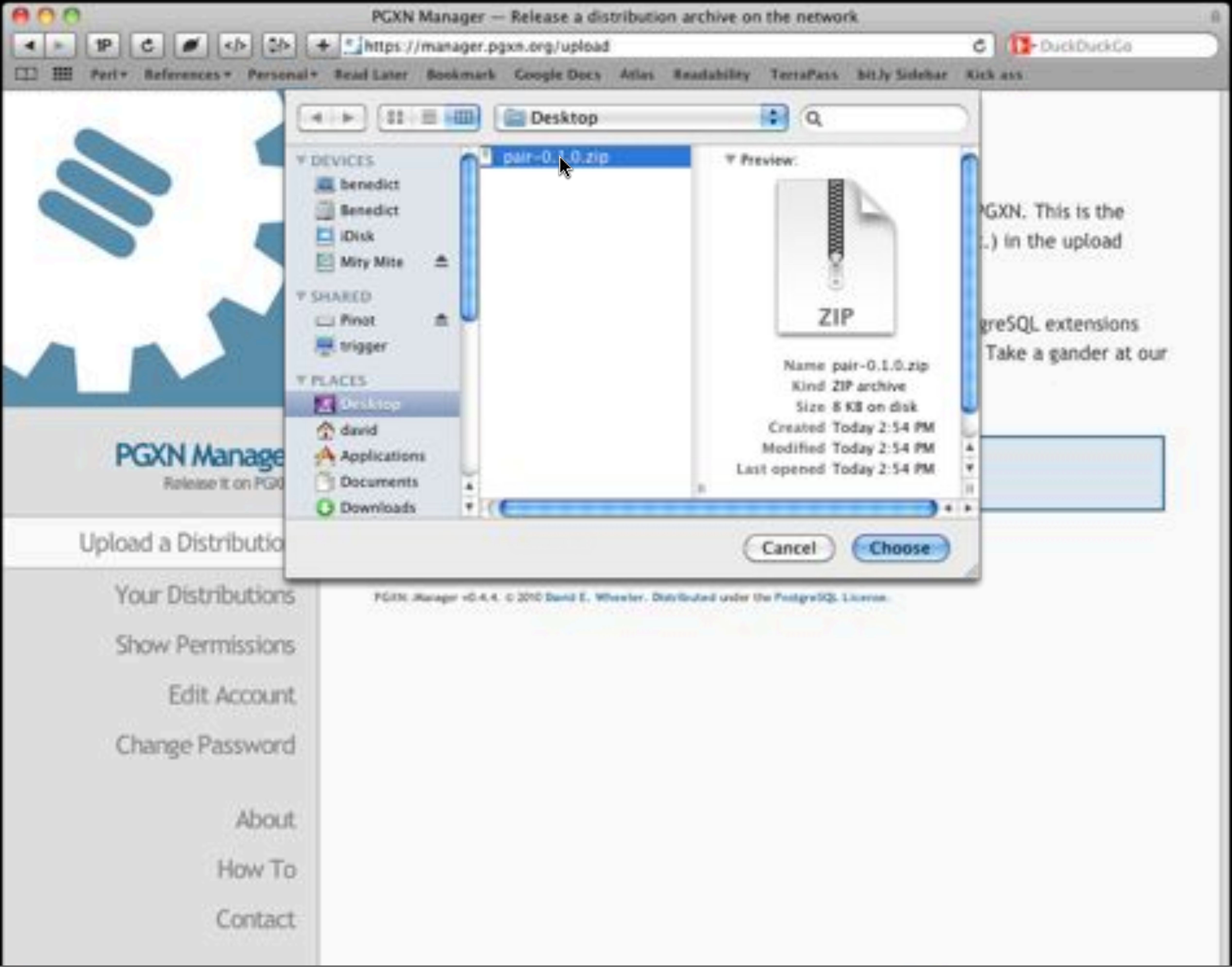
PGXN Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.



PGXN. This is the
.) in the upload

greSQL extensions
Take a gander at our

PGXN Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.


PGXN Manager — Release a distribution archive on the network

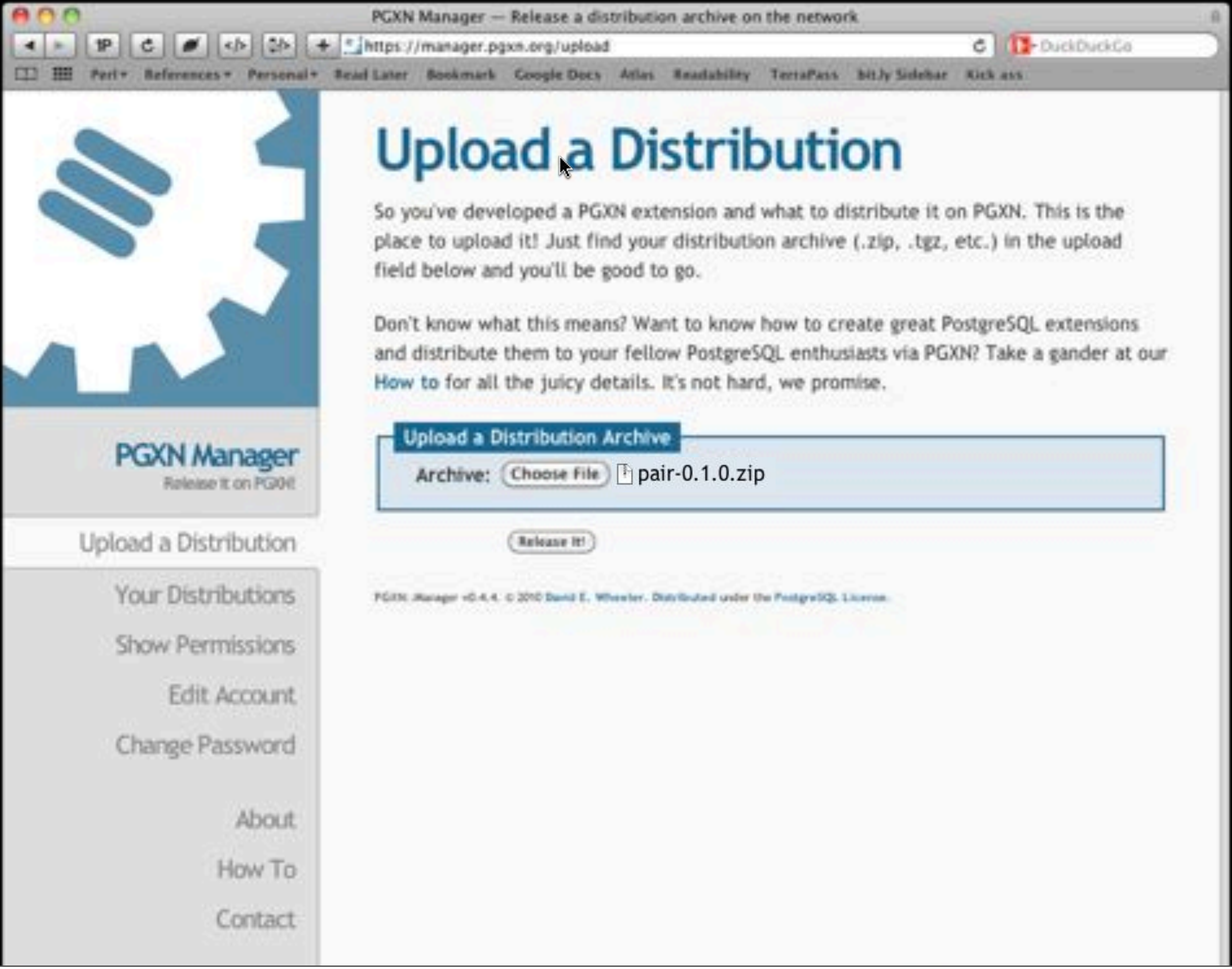
https://manager.pgxn.org/upload

Name	pair-0.1.0.zip
Kind	ZIP archive
Size	8 KB on disk
Created	Today 2:54 PM
Modified	Today 2:54 PM
Last opened	Today 2:54 PM

GXN. This is the .) in the upload

greSQL extensions Take a gander at our

PGXN::Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.


PGXN Manager — Release a distribution archive on the network
https://manager.pgxn.org/upload
DuckDuckGo
Perl▾ References▾ Personal▾ Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass
PGXN Manager
Release It on PGXN!
Upload a Distribution
Your Distributions
Show Permissions
Edit Account
Change Password
About
How To
Contact

Upload a Distribution

So you've developed a PGXN extension and what to distribute it on PGXN. This is the place to upload it! Just find your distribution archive (.zip, .tgz, etc.) in the upload field below and you'll be good to go.

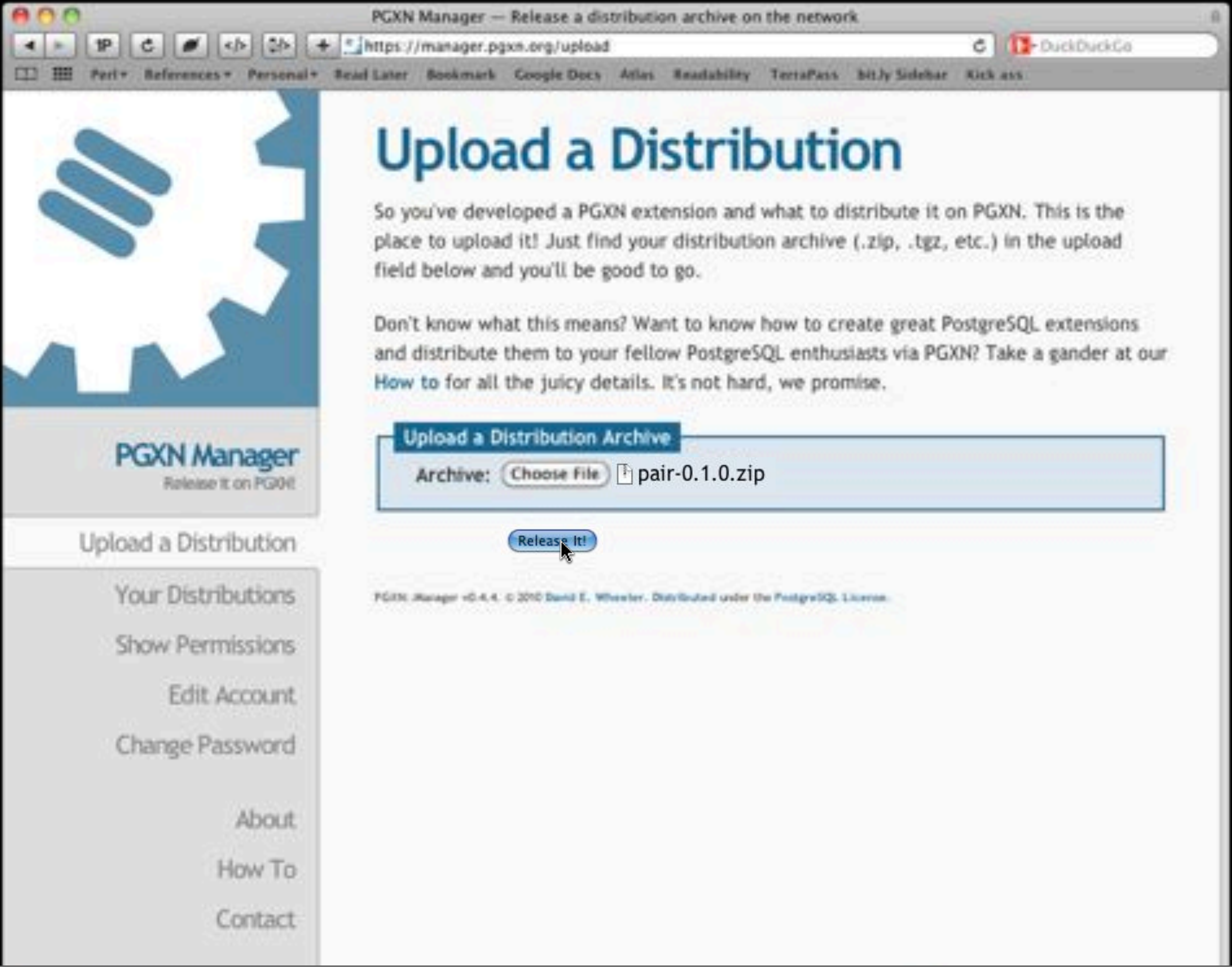
Don't know what this means? Want to know how to create great PostgreSQL extensions and distribute them to your fellow PostgreSQL enthusiasts via PGXN? Take a gander at our How to for all the juicy details. It's not hard, we promise.

Upload a Distribution Archive

Archive: Choose File pair-0.1.0.zip

Release it!

PGXN Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.


PGXN Manager — Release a distribution archive on the network
https://manager.pgxn.org/upload
DuckDuckGo
Perl▾ References▾ Personal▾ Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass
PGXN Manager
Release it on PGXN!
Upload a Distribution
Your Distributions
Show Permissions
Edit Account
Change Password
About
How To
Contact

Upload a Distribution

So you've developed a PGXN extension and what to distribute it on PGXN. This is the place to upload it! Just find your distribution archive (.zip, .tgz, etc.) in the upload field below and you'll be good to go.

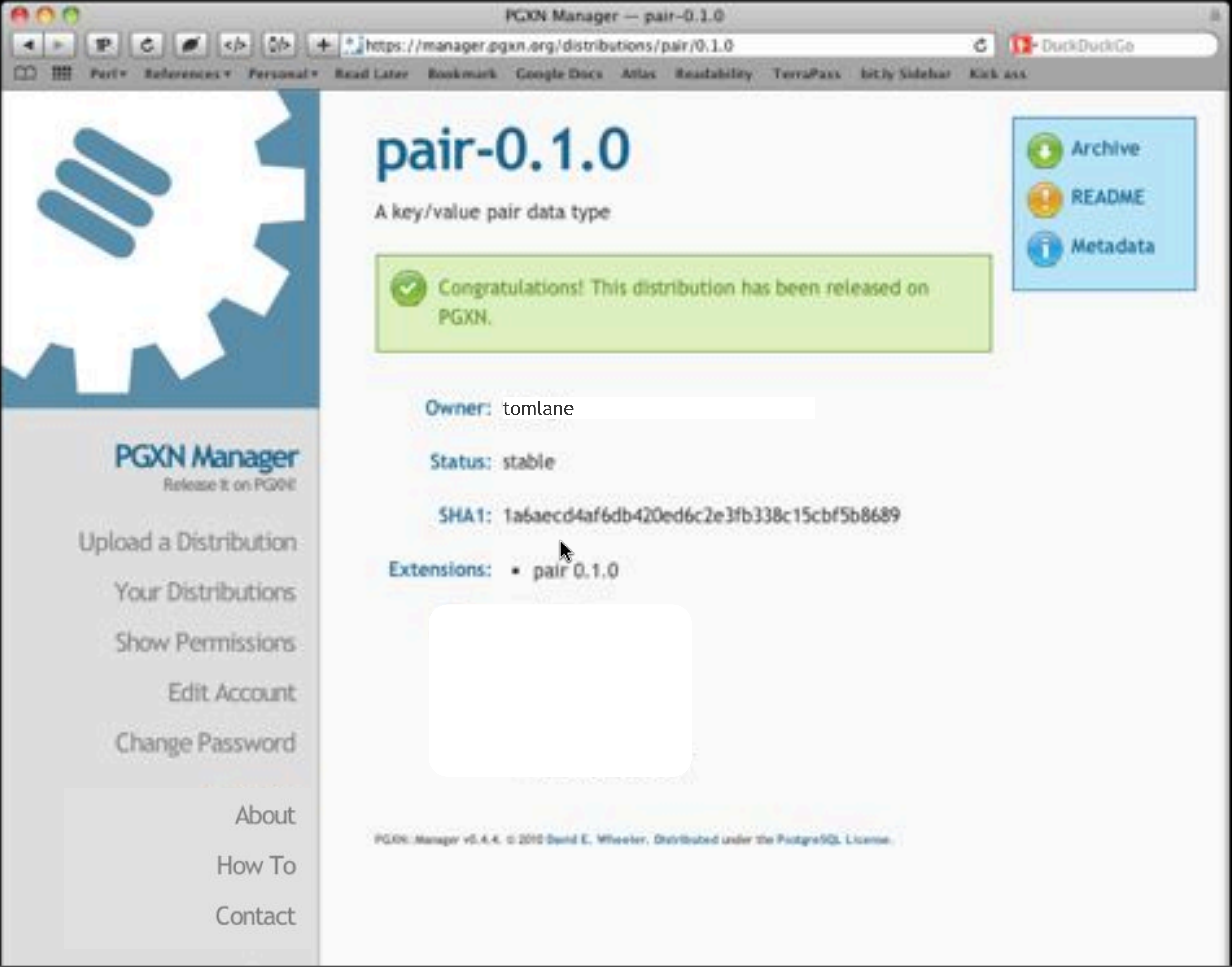
Don't know what this means? Want to know how to create great PostgreSQL extensions and distribute them to your fellow PostgreSQL enthusiasts via PGXN? Take a gander at our How to for all the juicy details. It's not hard, we promise.

Upload a Distribution Archive

Archive: Choose File pair-0.1.0.zip

Release It!

PGXN Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.



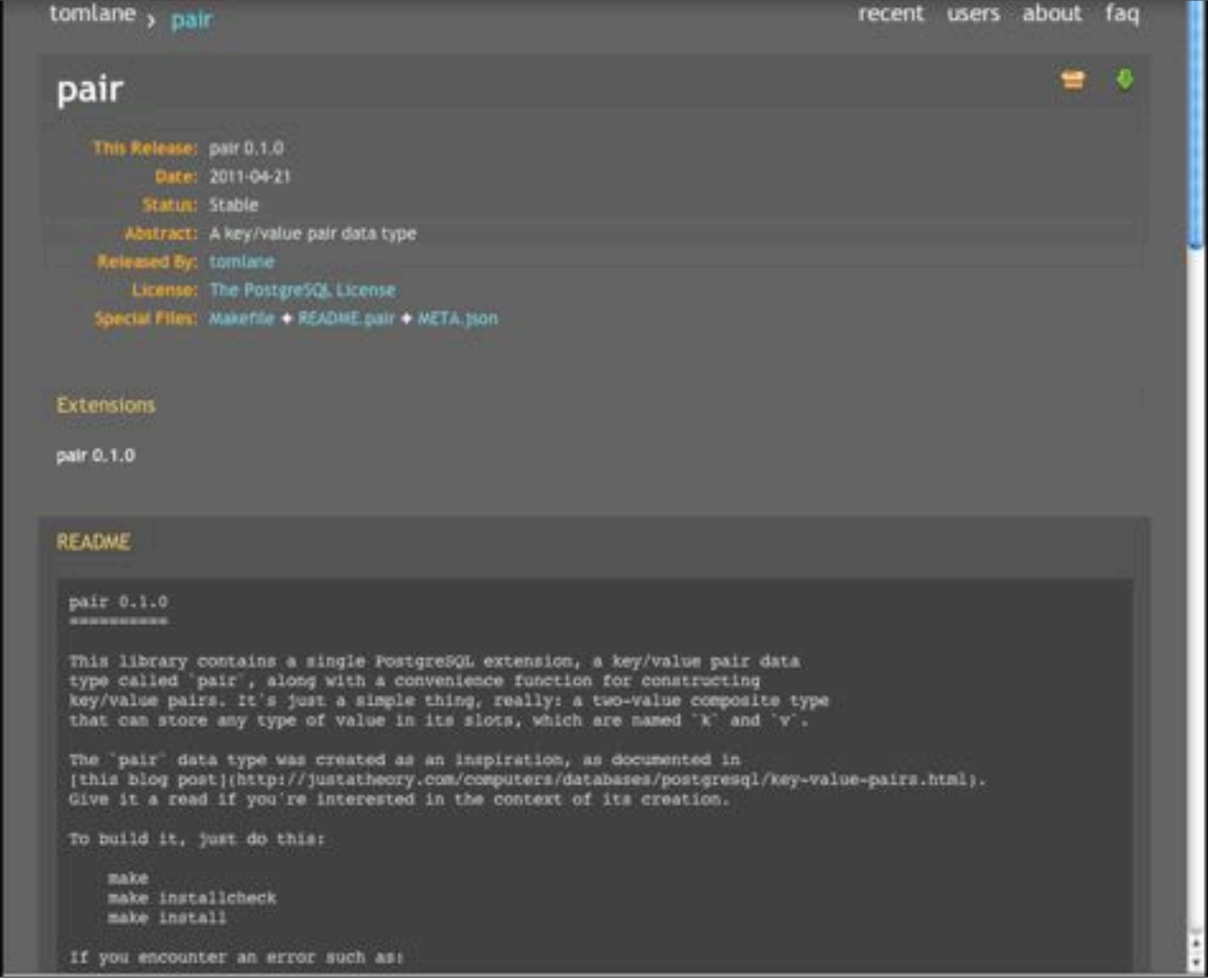
pair-0.1.0

A key/value pair data type

Congratulations! This distribution has been released on PGXN.

Owner:	tomlane
Status:	stable
SHA1:	1a6aecd4af6db420ed6c2e3fb338c15cbf5b8689
Extensions:	• pair 0.1.0

PGXN::Manager v0.4.4. © 2010 David E. Wheeler. Distributed under the PostgreSQL License.



tomlane › pair

recent users about faq

pair

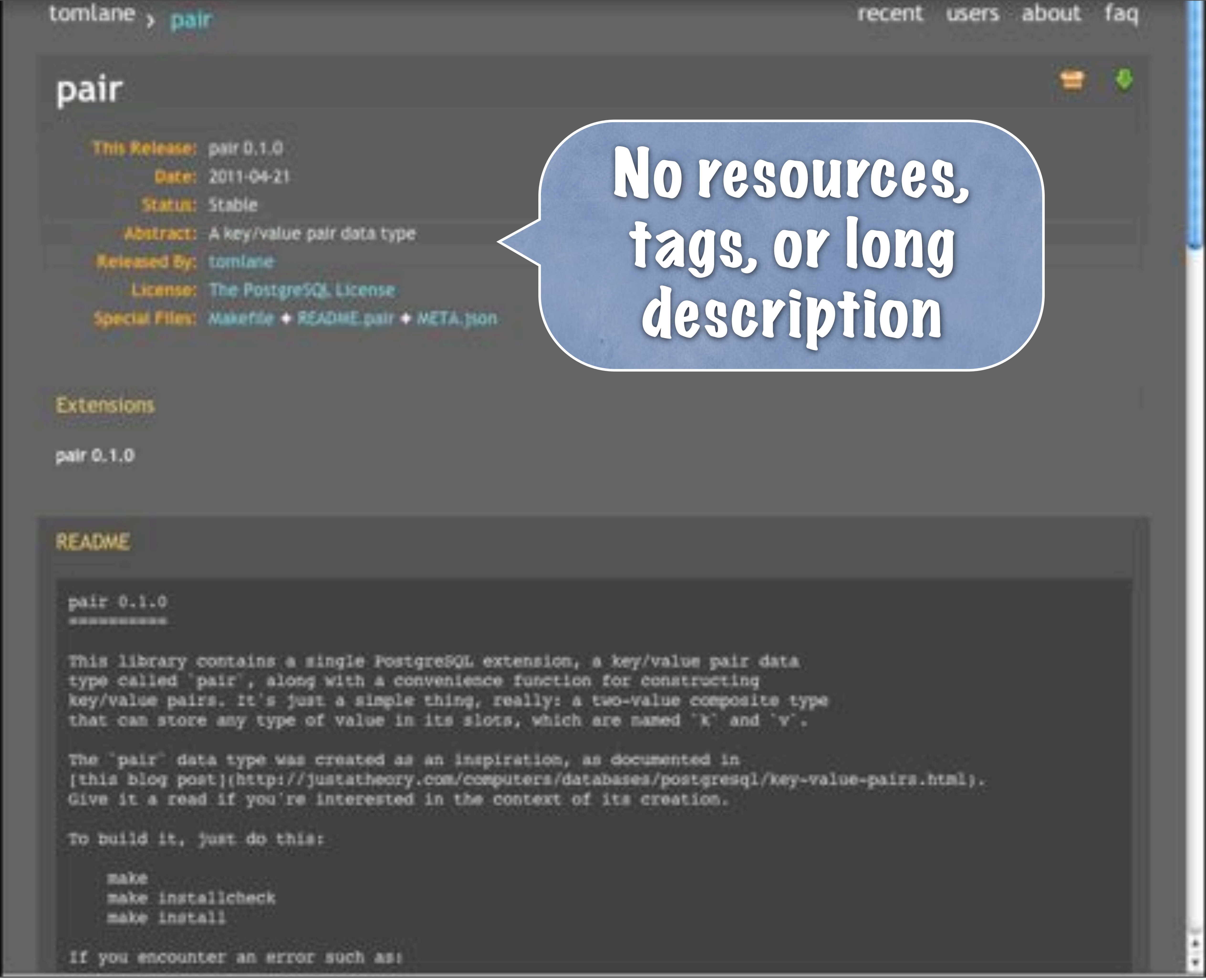
This Release: pair 0.1.0
Date: 2011-04-21
Status: Stable
Abstract: A key/value pair data type
Released By: tomlane
License: The PostgreSQL License
Special Files: Makefile ✦ README.pair ✦ META.json

Extensions

pair 0.1.0

README

```
pair 0.1.0
==========

This library contains a single PostgreSQL extension, a key/value pair data
type called `pair`, along with a convenience function for constructing
key/value pairs. It's just a simple thing, really: a two-value composite type
that can store any type of value in its slots, which are named `k` and `v`.

The `pair` data type was created as an inspiration, as documented in
[this blog post](http://justatheory.com/computers/databases/postgresql/key-value-pairs.html).
Give it a read if you're interested in the context of its creation.

To build it, just do this:

    make
    make installcheck
    make install

If you encounter an error such as:
```



pair

This Release: pair 0.1.0
Date: 2011-04-21
Status: Stable
Abstract: A key/value pair data type
Released By: tomlane
License: The PostgreSQL License
Special Files: Makefile ✦ README.pair ✦ META.json

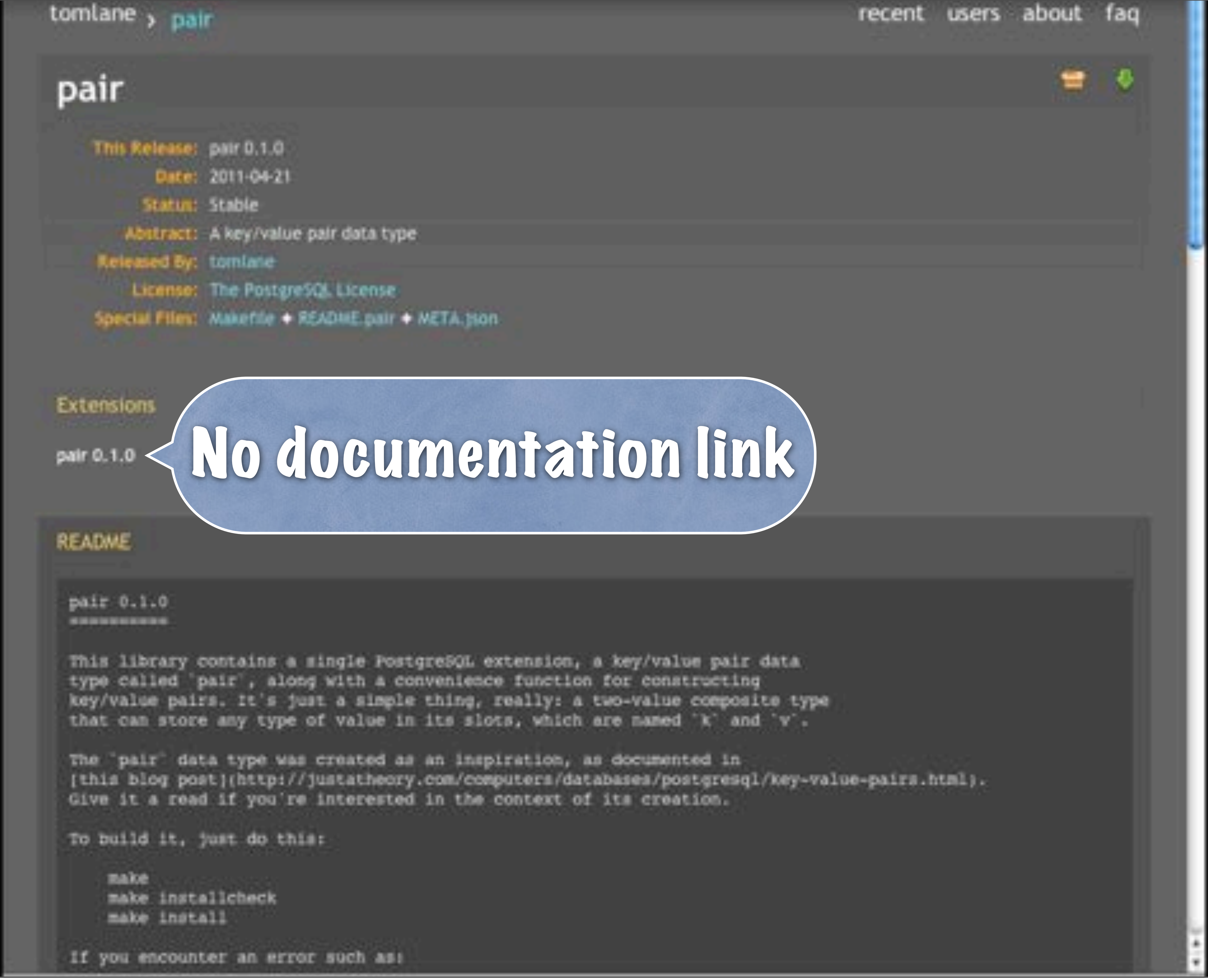
Extensions

pair 0.1.0

README

```
pair 0.1.0
==========

This library contains a single PostgreSQL extension, a key/value pair data
type called `pair`, along with a convenience function for constructing
key/value pairs. It's just a simple thing, really: a two-value composite type
that can store any type of value in its slots, which are named `k` and `v`.

The `pair` data type was created as an inspiration, as documented in
[this blog post](http://justatheory.com/computers/databases/postgresql/key-value-pairs.html).
Give it a read if you're interested in the context of its creation.

To build it, just do this:

    make
    make installcheck
    make install

If you encounter an error such as:
```

No resources, tags, or long description



pair

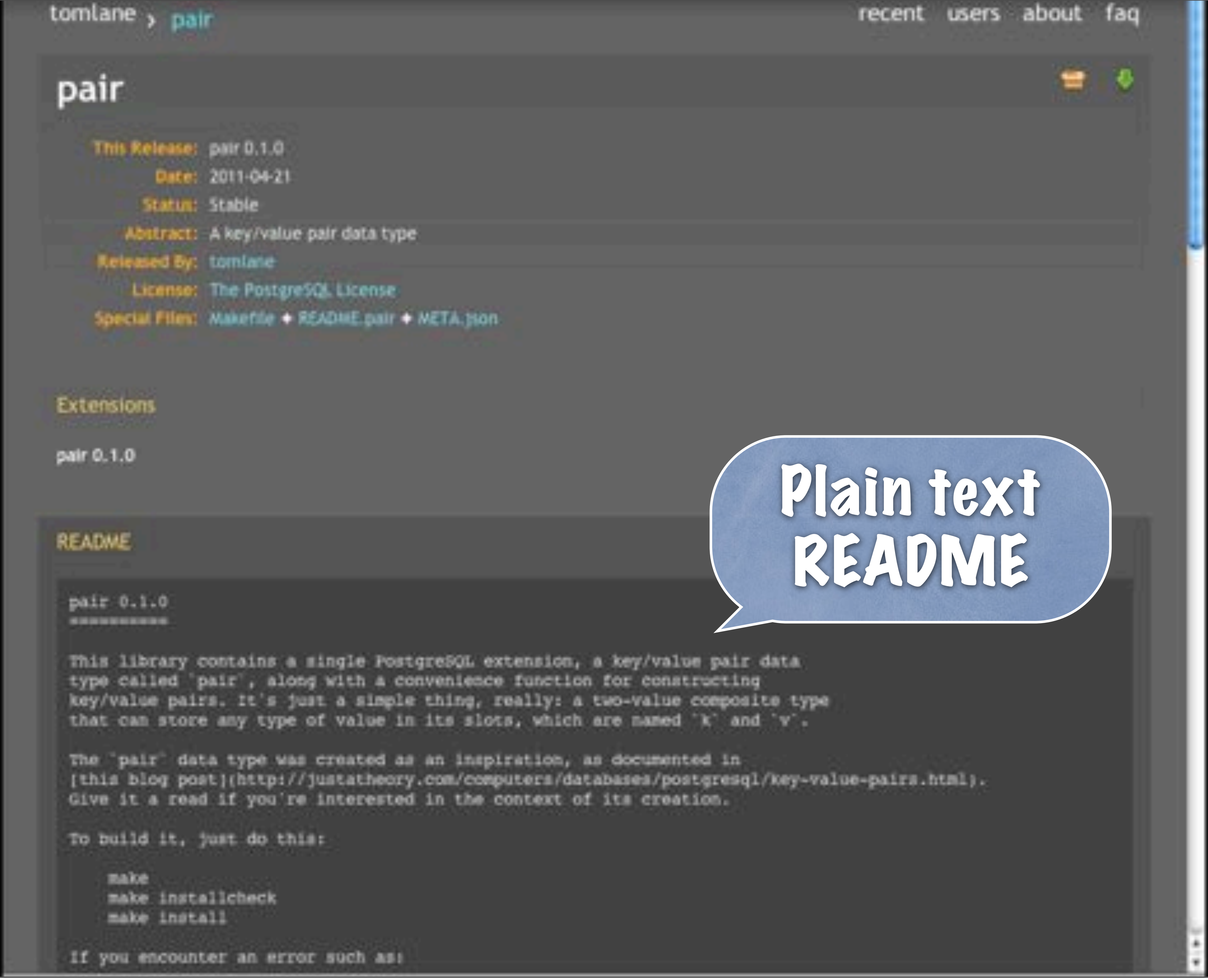
This Release: pair 0.1.0
Date: 2011-04-21
Status: Stable
Abstract: A key/value pair data type
Released By: tomlane
License: The PostgreSQL License
Special Files: Makefile ✦ README.pair ✦ META.json

Extensions

pair 0.1.0

README

```
pair 0.1.0
==========

This library contains a single PostgreSQL extension, a key/value pair data
type called `pair`, along with a convenience function for constructing
key/value pairs. It's just a simple thing, really: a two-value composite type
that can store any type of value in its slots, which are named `k` and `v`.

The `pair` data type was created as an inspiration, as documented in
[this blog post](http://justatheory.com/computers/databases/postgresql/key-value-pairs.html).
Give it a read if you're interested in the context of its creation.

To build it, just do this:

    make
    make installcheck
    make install

If you encounter an error such as:
```


tomlane › pair

recent users about faq

pair

This Release:	pair 0.1.0
Date:	2011-04-21
Status:	Stable
Abstract:	A key/value pair data type
Released By:	tomlane
License:	The PostgreSQL License
Special Files:	Makefile ◆ README.pair ◆ META.json

Extensions

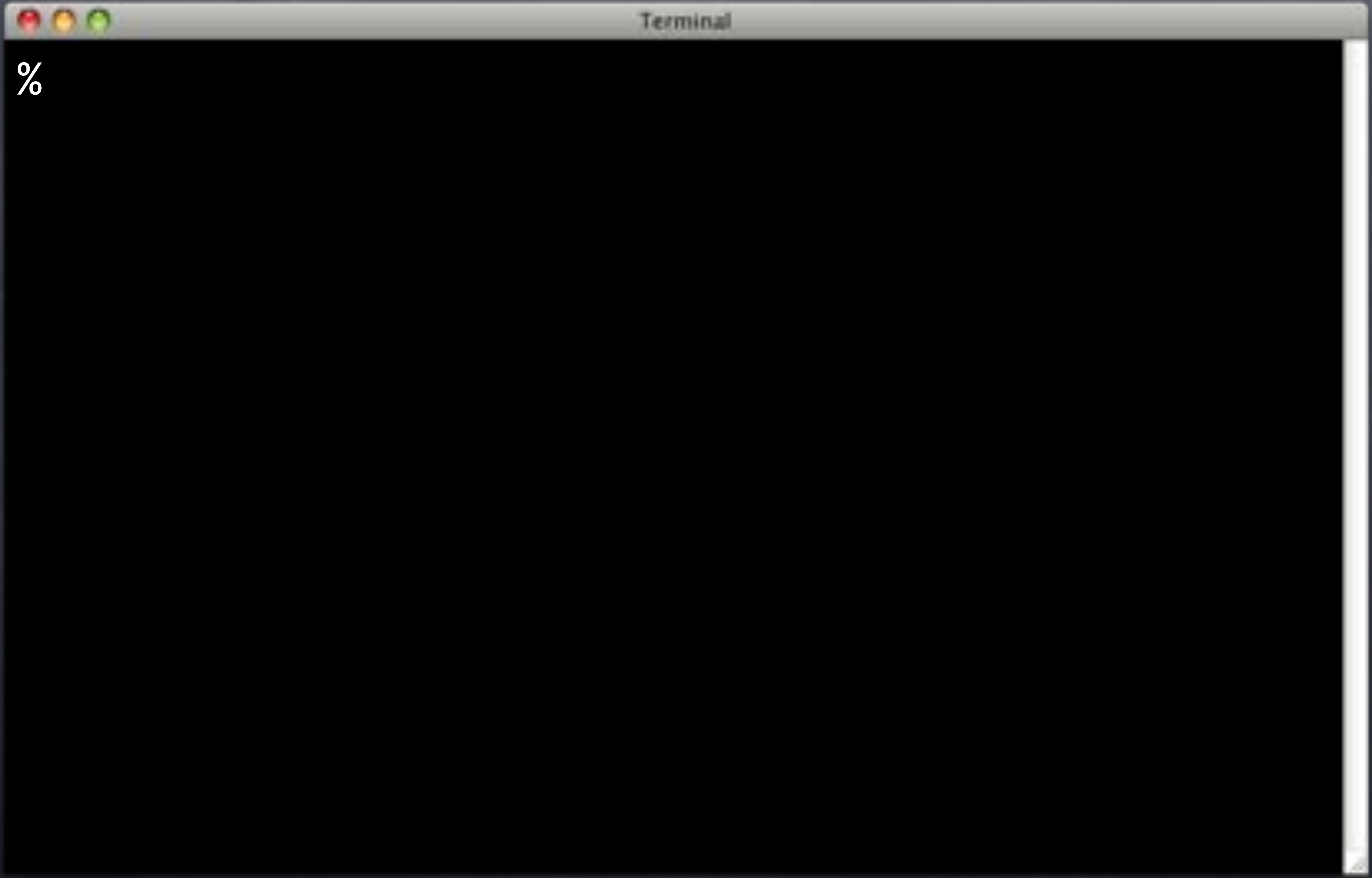
pair 0.1.0

README

```
pair 0.1.0
==========

This library contains a single PostgreSQL extension, a key/value pair data
type called `pair`, along with a convenience function for constructing
key/value pairs. It's just a simple thing, really: a two-value composite type
that can store any type of value in its slots, which are named `k` and `v`.

The `pair` data type was created as an inspiration, as documented in
[this blog post](http://justatheory.com/computers/databases/postgresql/key-value-pairs.html).
Give it a read if you're interested in the context of its creation.

To build it, just do this:

    make
    make installcheck
    make install

If you encounter an error such as:
```

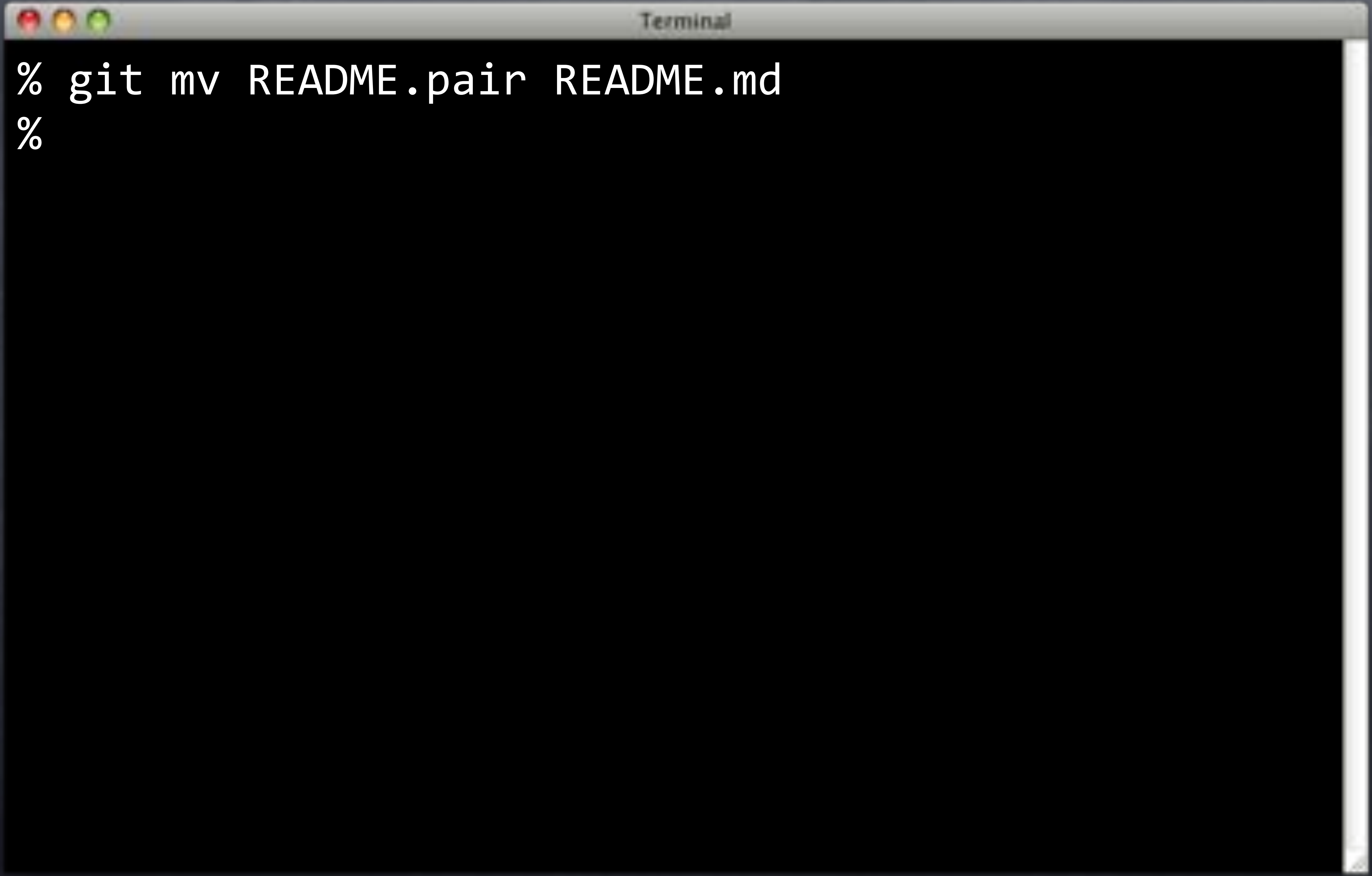
Plain text README

Add README Extension



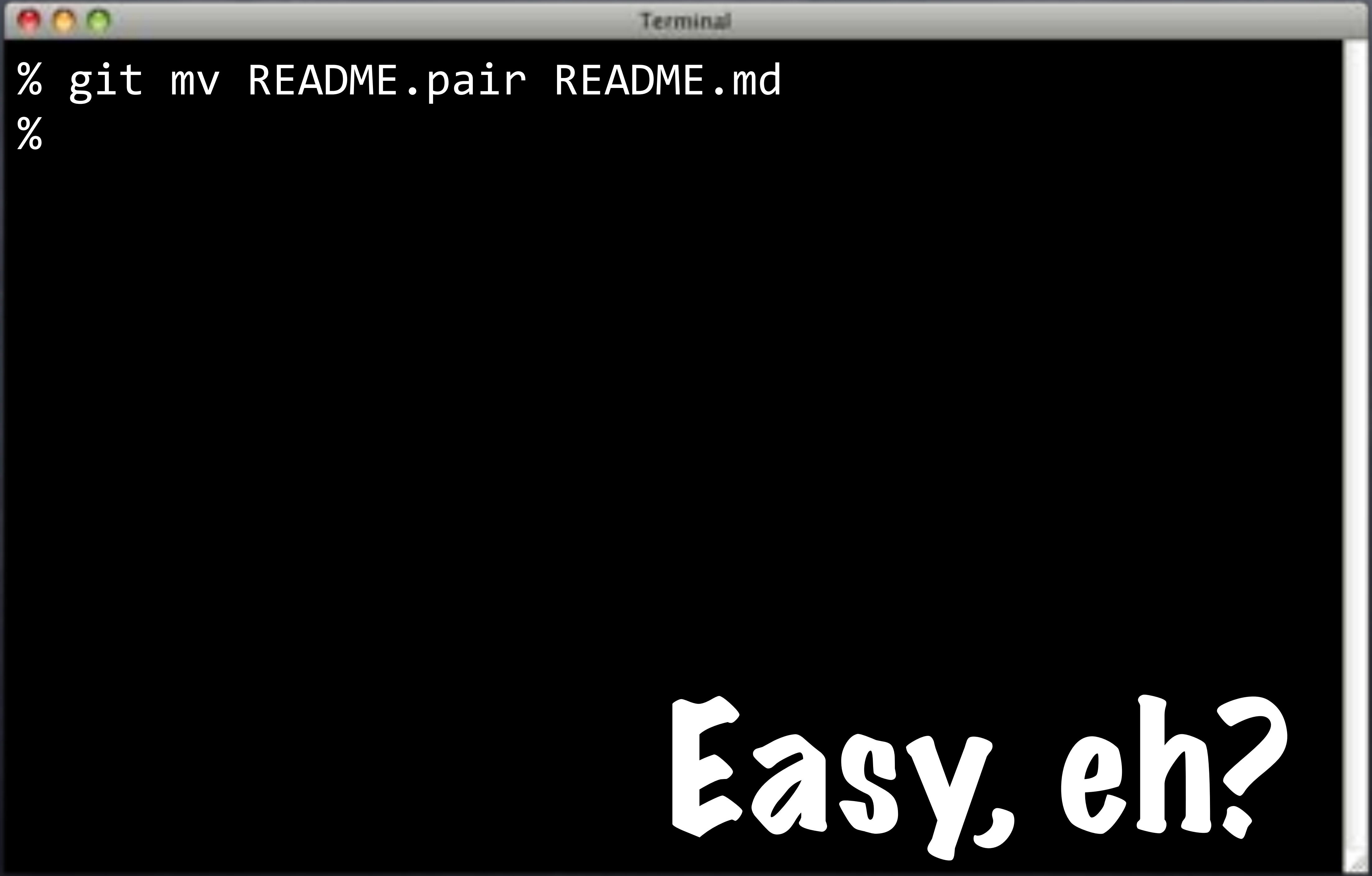
```
%
```

Add README Extension



```
% git mv README.pair README.md
%
```

Add README Extension



```
% git mv README.pair README.md
%
```

Easy, eh?

PGXN Markup

PGXN Markup

- HTML
- Markdown
- MultiMarkdown
- Pod
- Textile
- Trac
- MediaWiki

PGXN Markup

- HTML
- Markdown
- MultiMarkdown
- Pod
- Textile
- Trac
- MediaWiki

- Text::Markup (fork me!)

PGXN Markup

- HTML
- Markdown
- MultiMarkdown
- Pod
- Textile
- Trac
- MediaWiki

- Text::Markup (fork me!)
- Write some Docs!

Add Documentation

Add Documentation

- Use any supported markup

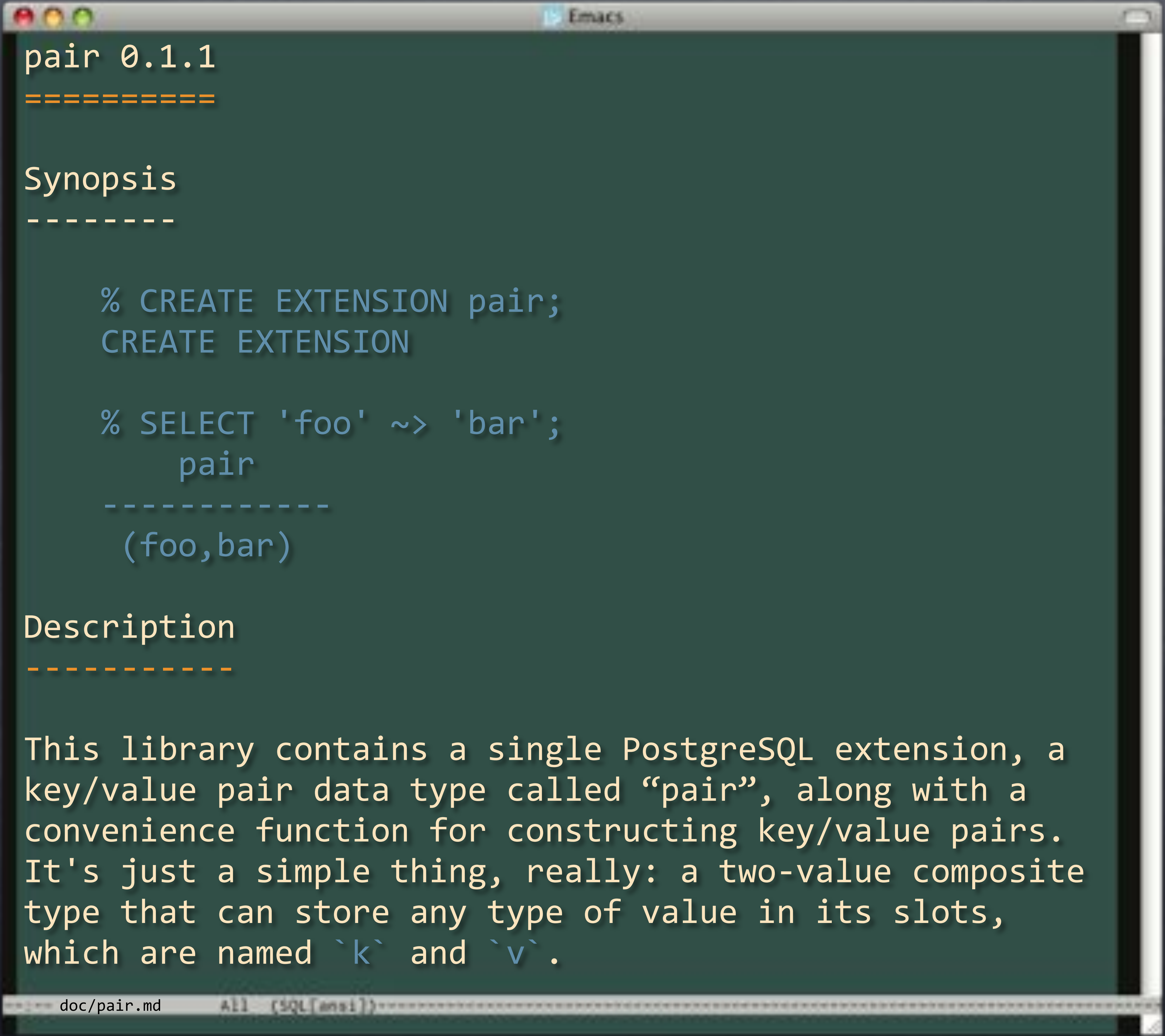
Add Documentation

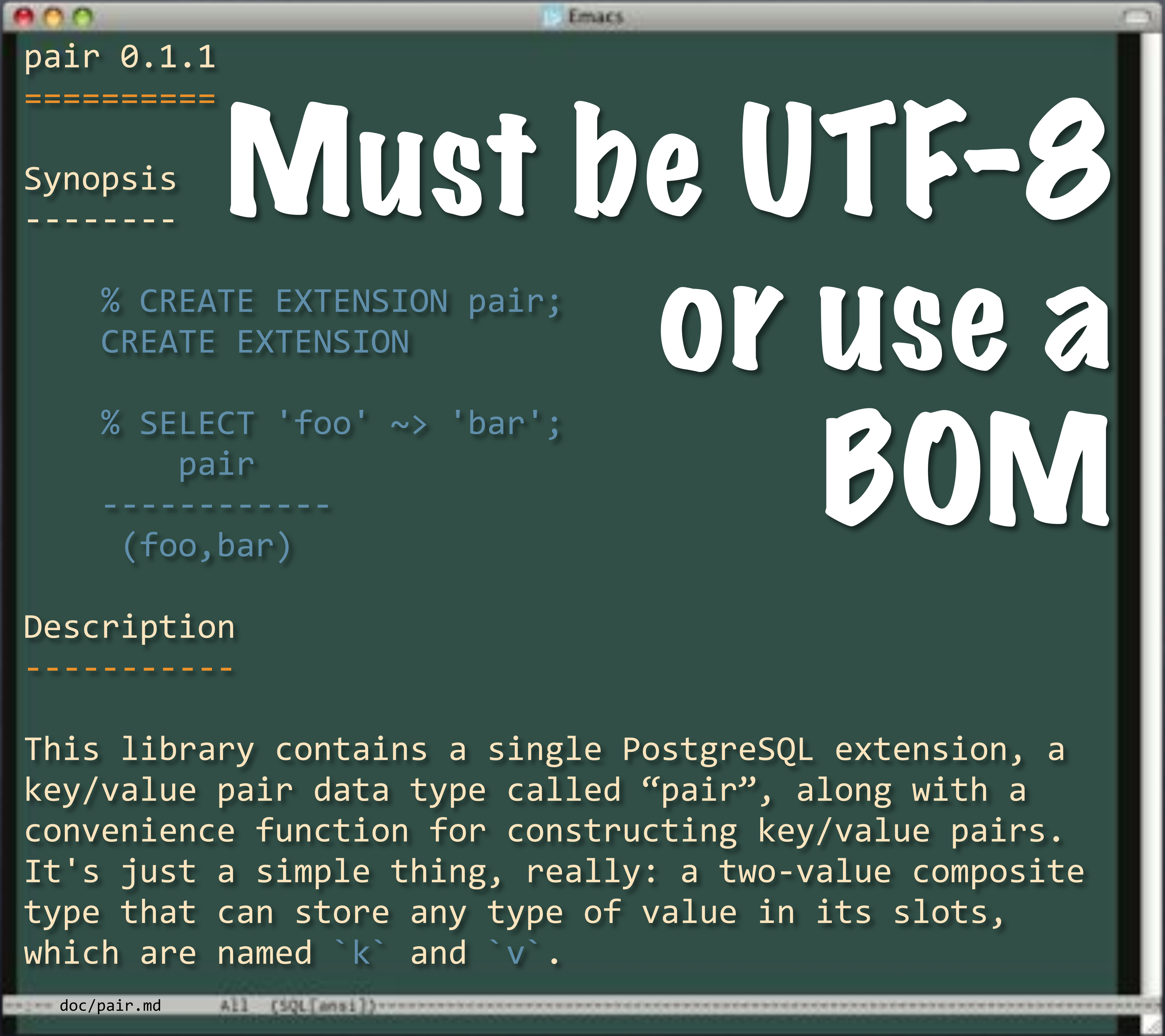
- Use any supported markup
- Put wherever you like

Add Documentation

- Use any supported markup
- Put wherever you like
- Recommend doc/

Emacs
--:-- doc/pair.md All (SQL[ansi])

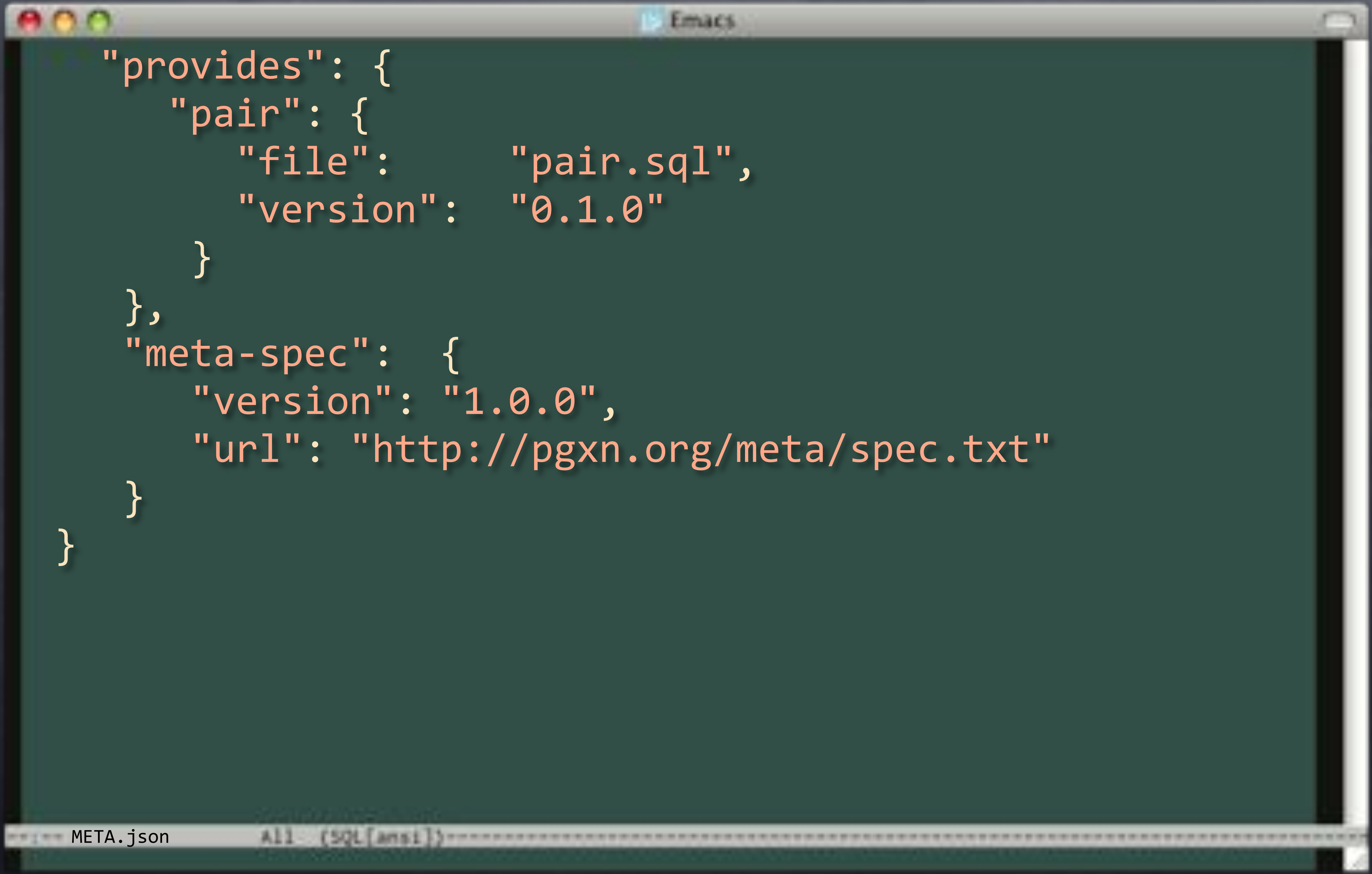

```
pair 0.1.1
==========

Synopsis
--------

    % CREATE EXTENSION pair;
    CREATE EXTENSION

    % SELECT 'foo' ~> 'bar';
        pair
    ------------
     (foo,bar)

Description
-----------

This library contains a single PostgreSQL extension, a
key/value pair data type called “pair”, along with a
convenience function for constructing key/value pairs.
It's just a simple thing, really: a two-value composite
type that can store any type of value in its slots,
which are named `k` and `v`.
```



Must be UTF-8 or use a BOM

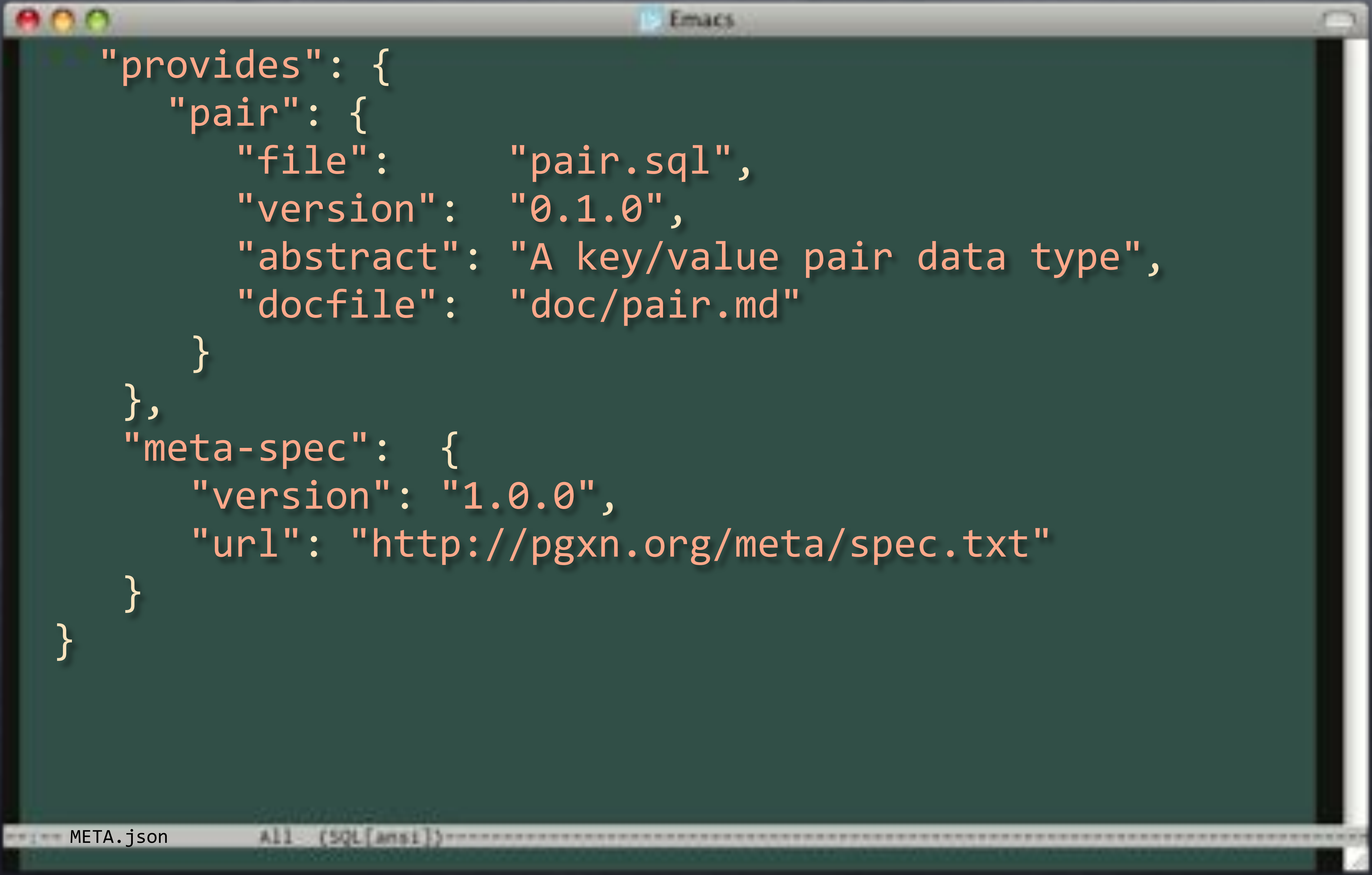
```
pair 0.1.1
==========

Synopsis
--------

    % CREATE EXTENSION pair;
    CREATE EXTENSION

    % SELECT 'foo' ~> 'bar';
        pair
    ------------
     (foo,bar)

Description
-----------

This library contains a single PostgreSQL extension, a
key/value pair data type called “pair”, along with a
convenience function for constructing key/value pairs.
It's just a simple thing, really: a two-value composite
type that can store any type of value in its slots,
which are named `k` and `v`.
```

Improve Provides



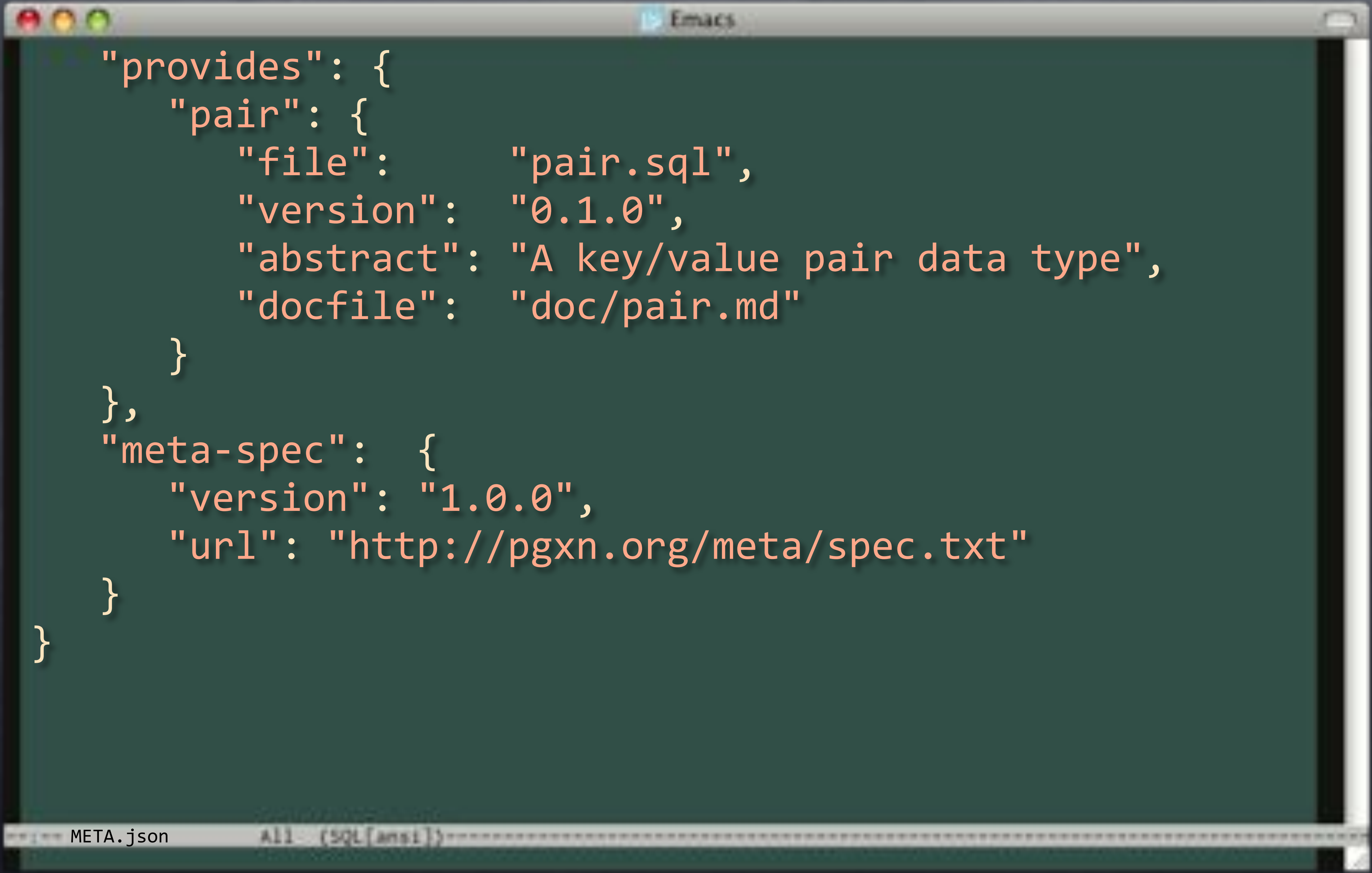
```
  "provides": {
     "pair": {
        "file":      "pair.sql",
        "version":   "0.1.0"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

Improve Provides



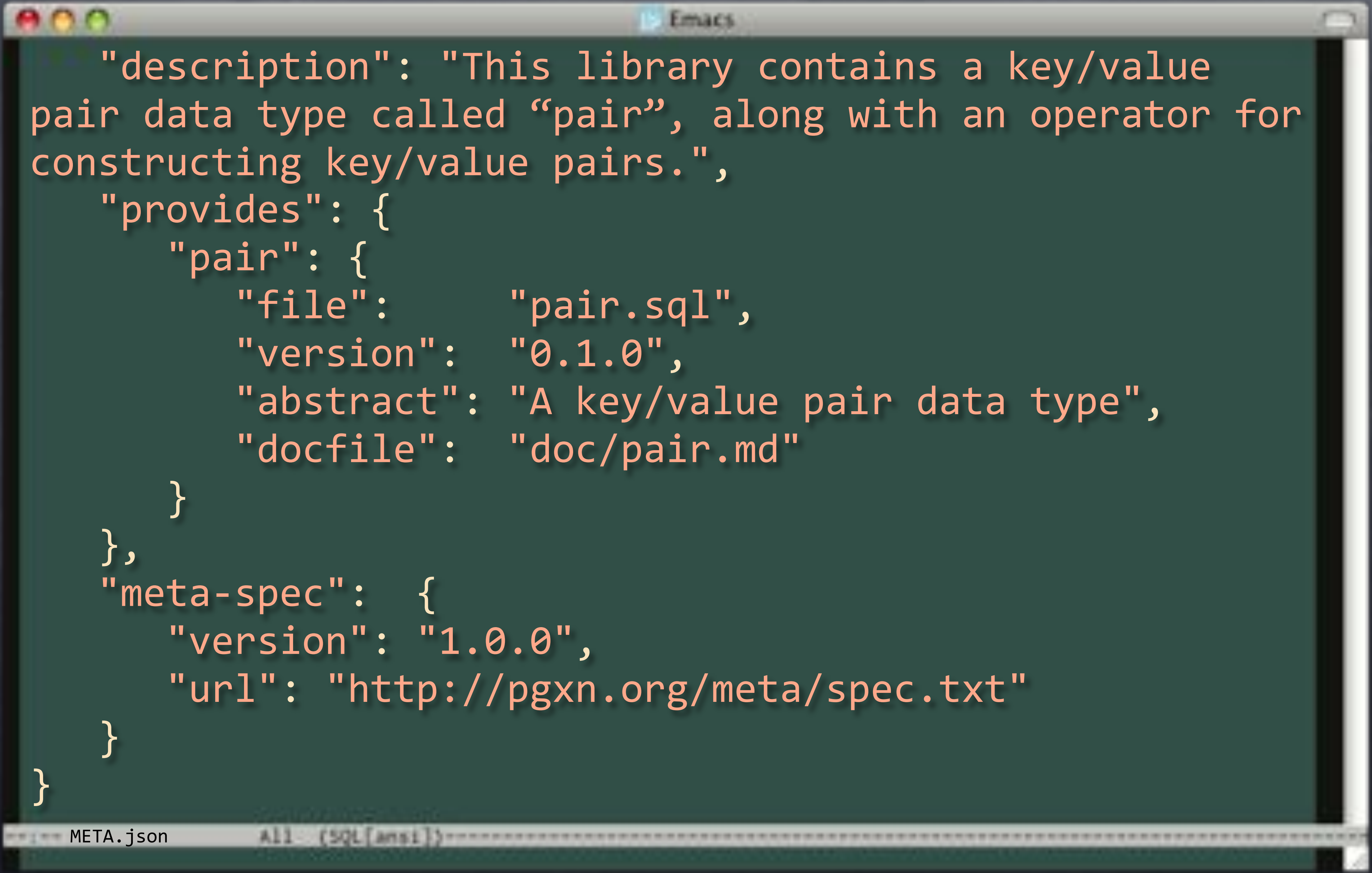
```
 "provides": {
    "pair": {
       "file":     "pair.sql",
       "version":  "0.1.0",
       "abstract": "A key/value pair data type",
       "docfile":  "doc/pair.md"
     }
  },
  "meta-spec":  {
     "version": "1.0.0",
     "url": "http://pgxn.org/meta/spec.txt"
  }
}
```

Add Description



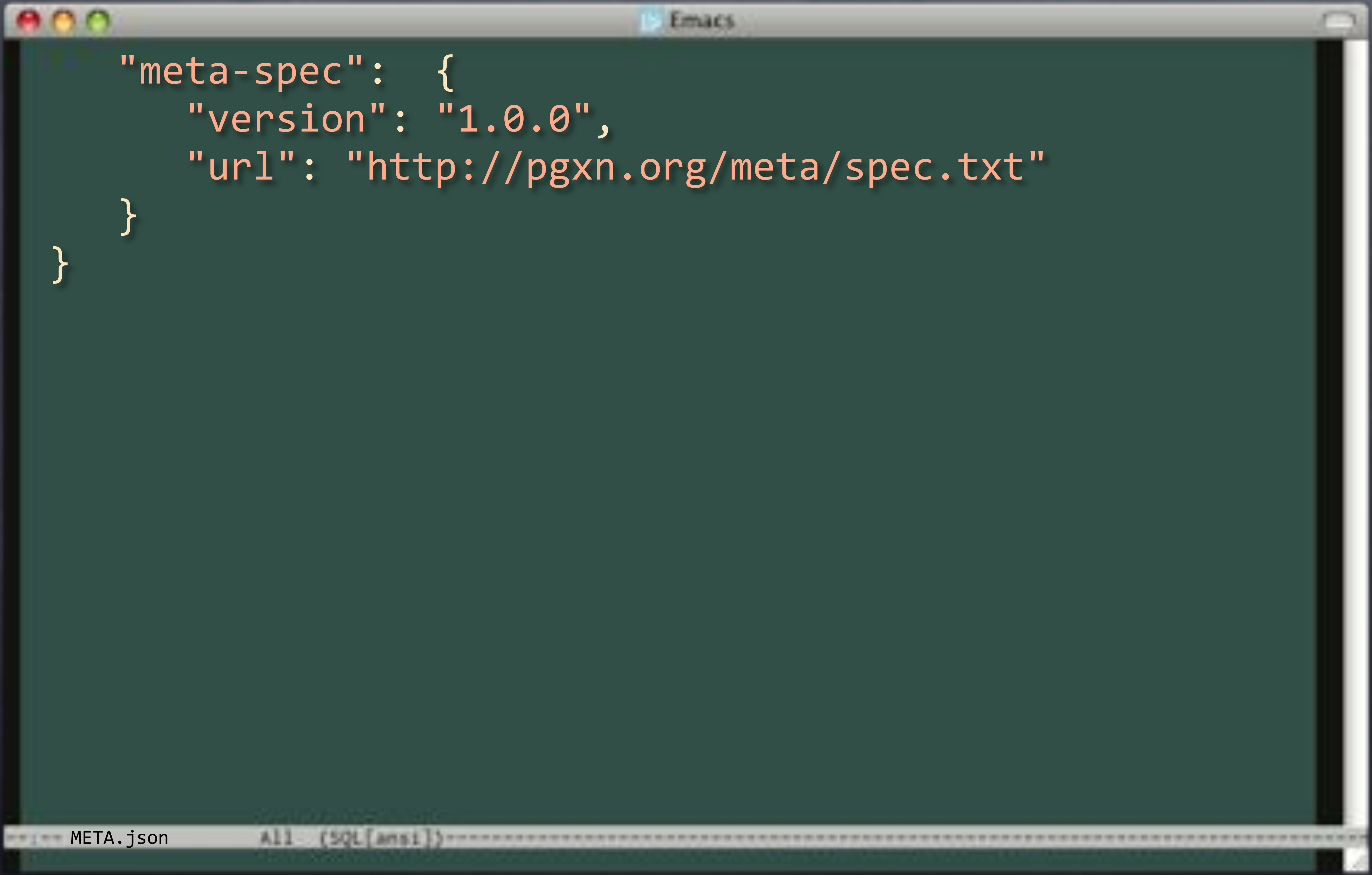
```
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0",
         "abstract": "A key/value pair data type",
         "docfile":  "doc/pair.md"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

Add Description



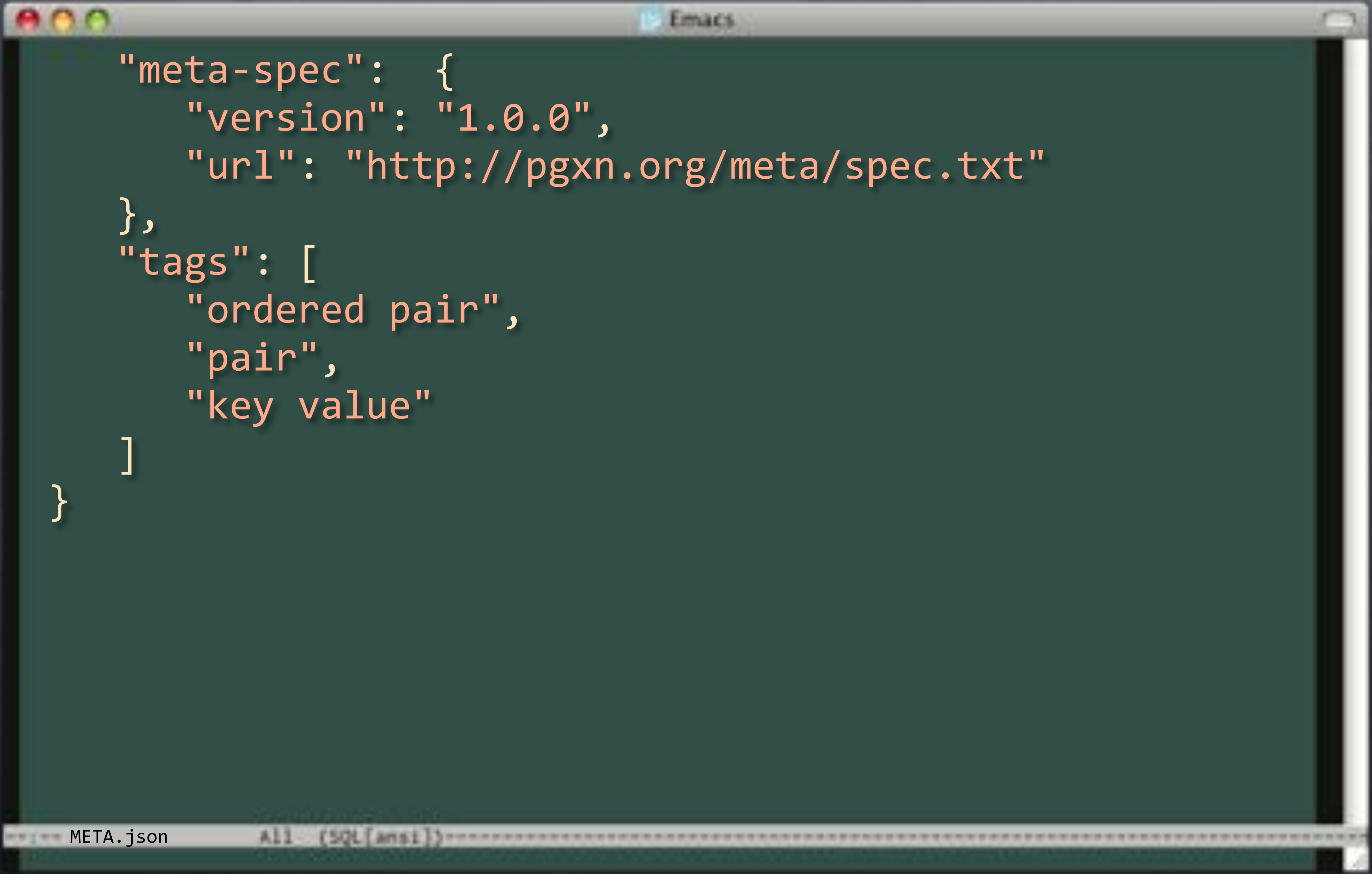
```
   "description": "This library contains a key/value
pair data type called “pair”, along with an operator for
constructing key/value pairs.",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0",
         "abstract": "A key/value pair data type",
         "docfile":  "doc/pair.md"
      }
   },
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

Add Tags



```
   "meta-spec":  {
      "version": "1.0.0",
      "url": "http://pgxn.org/meta/spec.txt"
   }
}
```

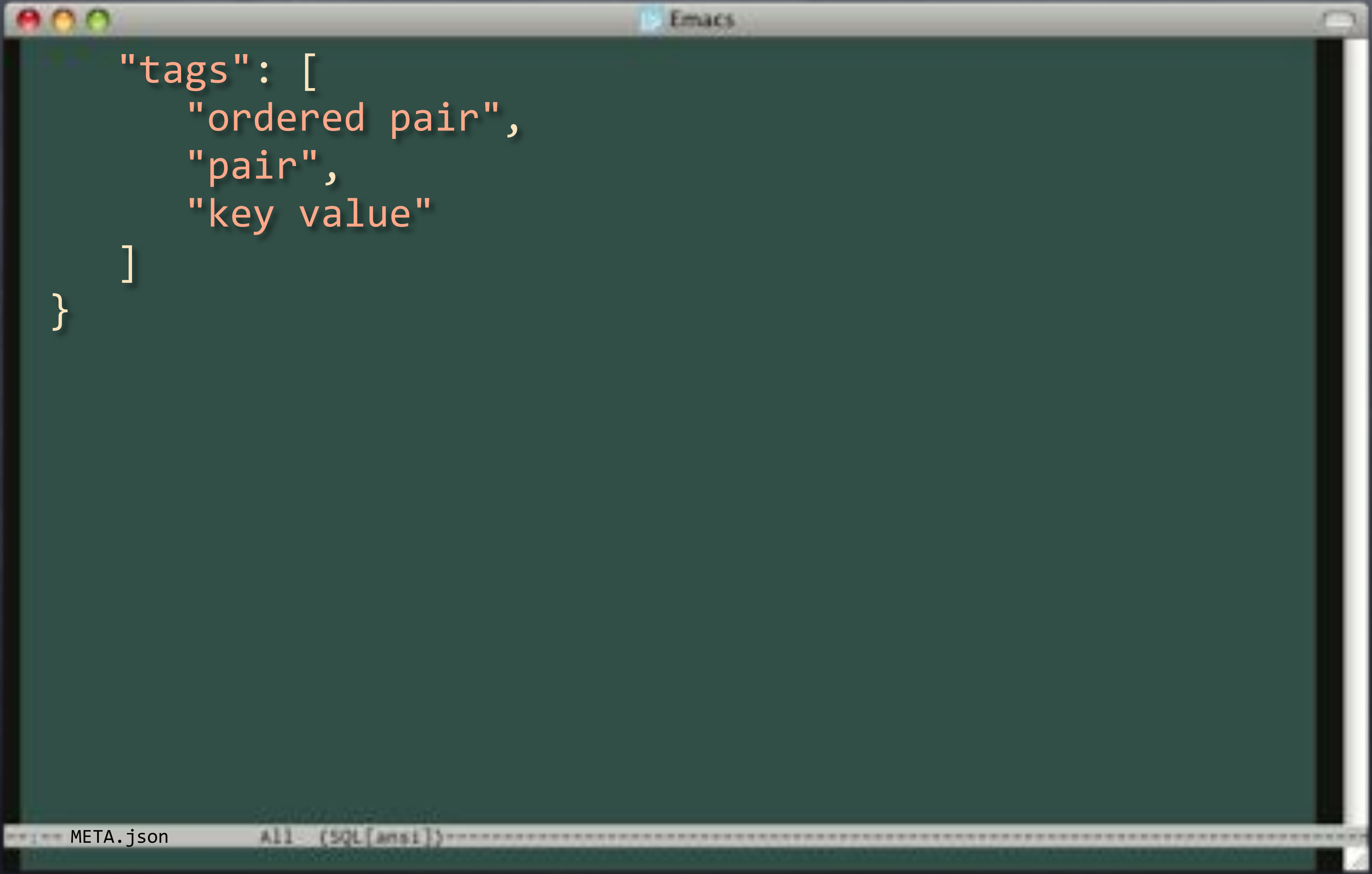
Add Tags


Emacs

```
    "meta-spec":  {
       "version": "1.0.0",
       "url": "http://pgxn.org/meta/spec.txt"
    },
    "tags": [
       "ordered pair",
       "pair",
       "key value"
    ]
}
```

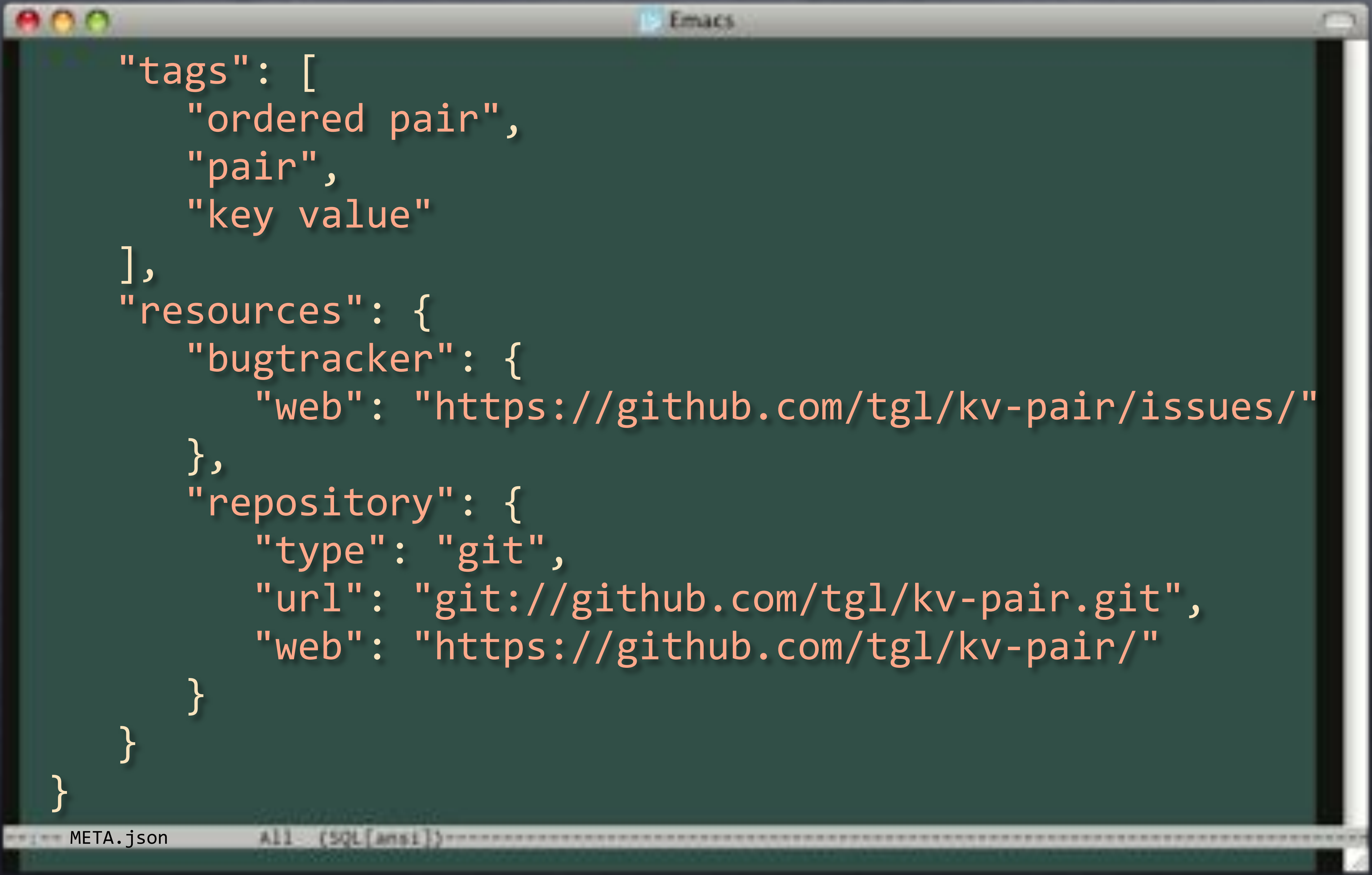
```
--:--  META.json      All   (SQL[ansi])------------------------------------------------
```

Add Resources



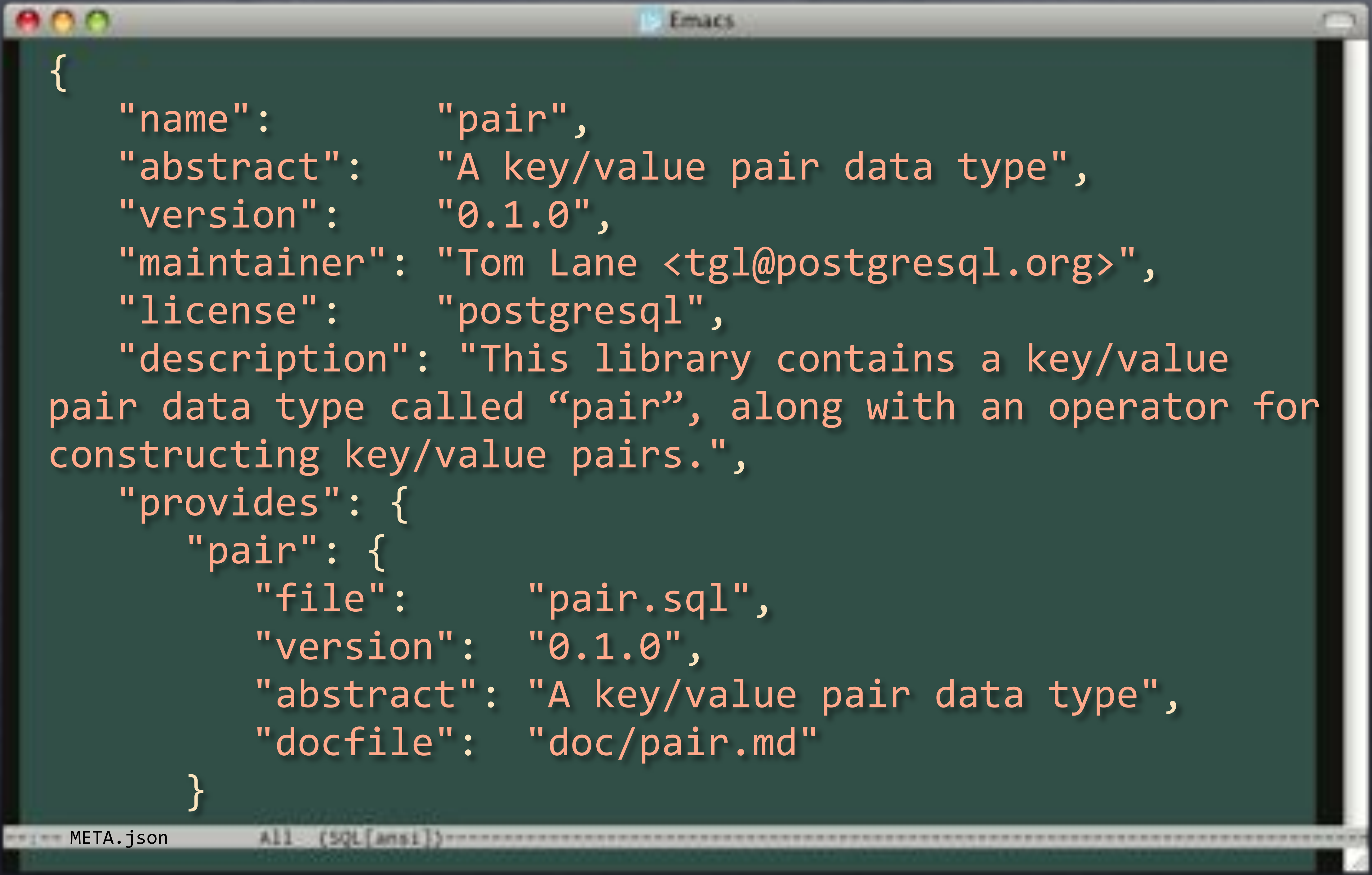
```
    "tags": [
       "ordered pair",
       "pair",
       "key value"
    ]
}
```

Add Resources



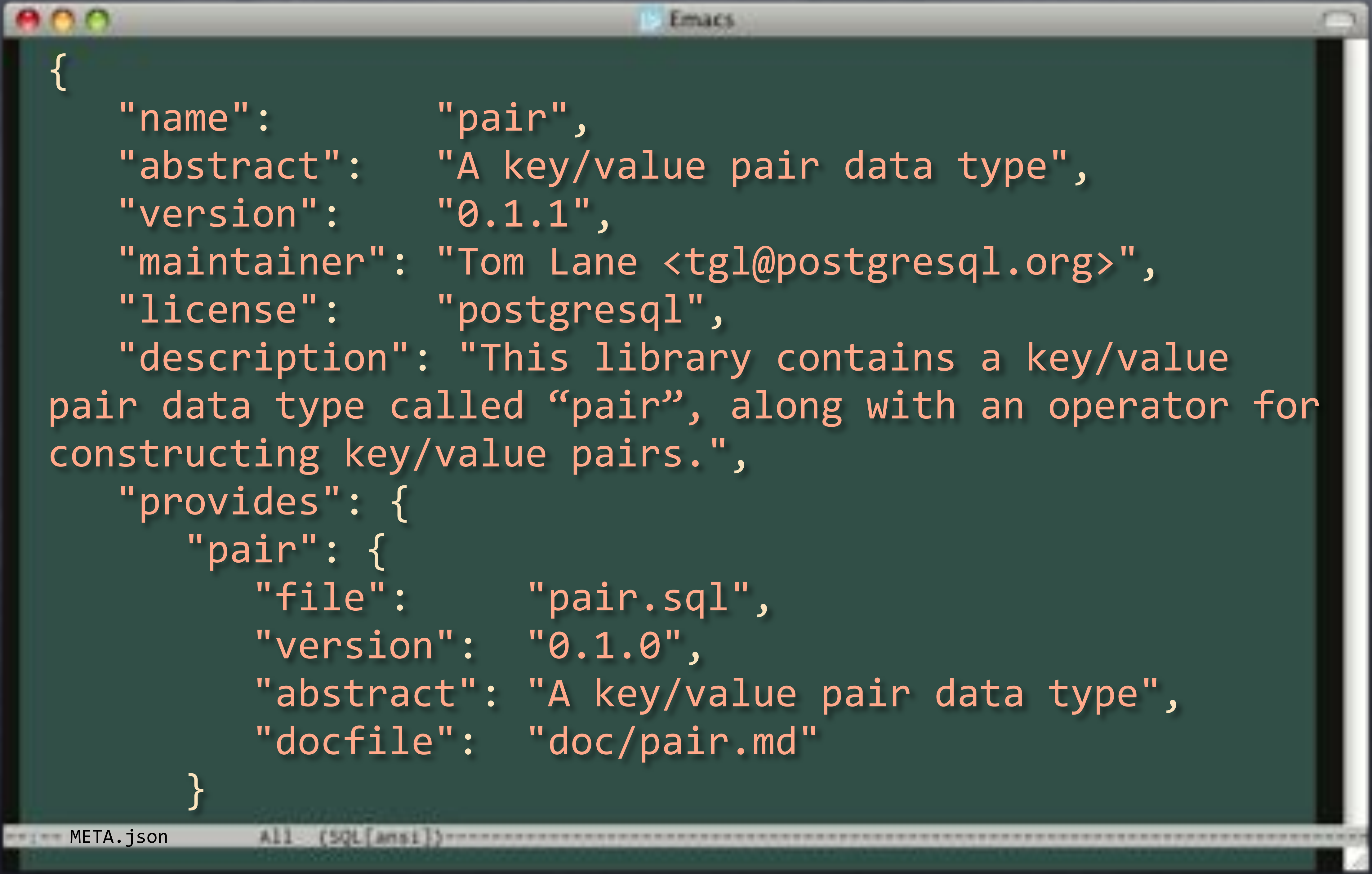
```
    "tags": [
       "ordered pair",
       "pair",
       "key value"
    ],
    "resources": {
       "bugtracker": {
          "web": "https://github.com/tgl/kv-pair/issues/"
       },
       "repository": {
          "type": "git",
          "url": "git://github.com/tgl/kv-pair.git",
          "web": "https://github.com/tgl/kv-pair/"
       }
    }
}
```

Update Version



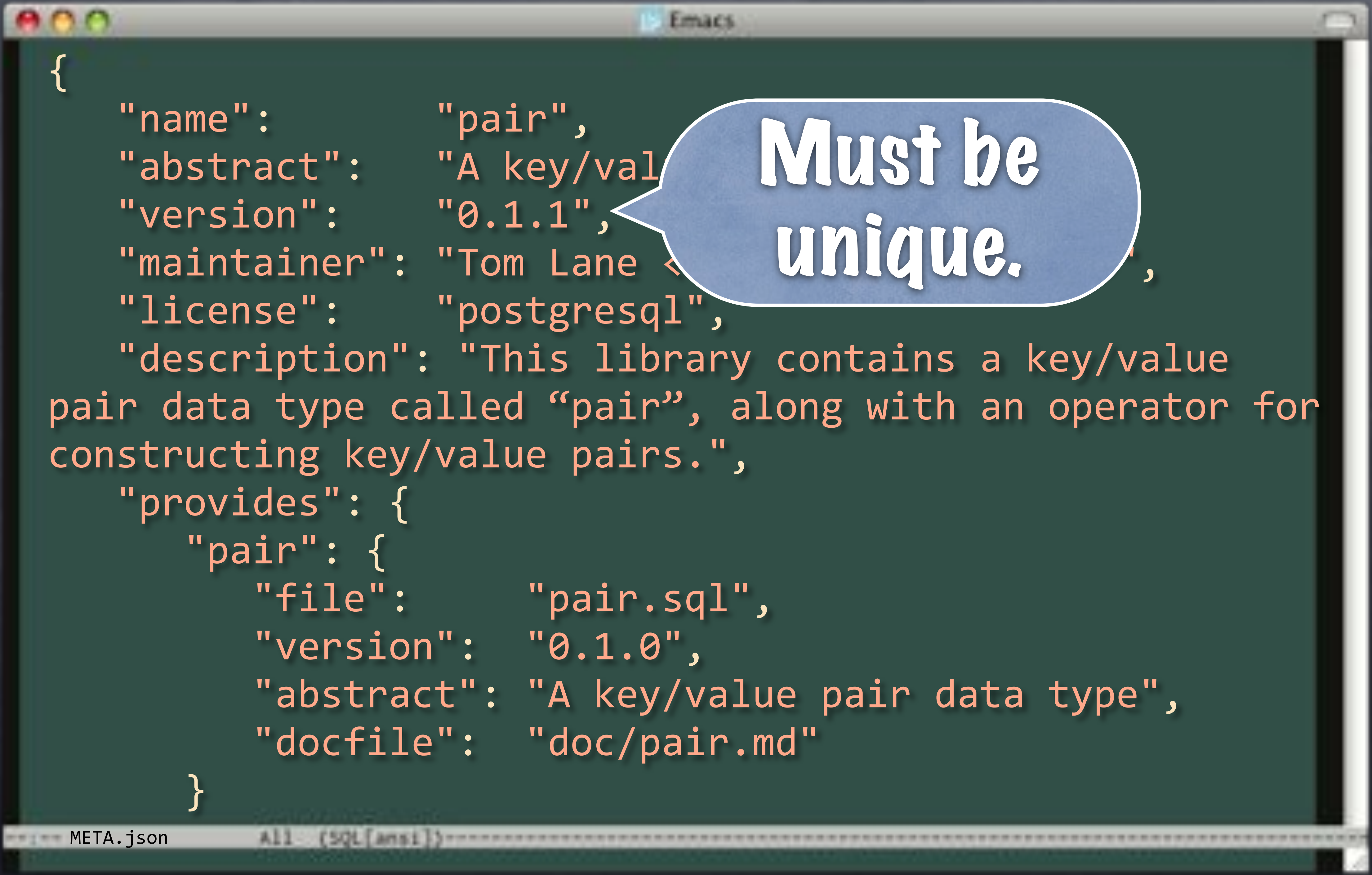
```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.0",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "description": "This library contains a key/value
pair data type called “pair”, along with an operator for
constructing key/value pairs.",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0",
         "abstract": "A key/value pair data type",
         "docfile":  "doc/pair.md"
      }
```

Update Version



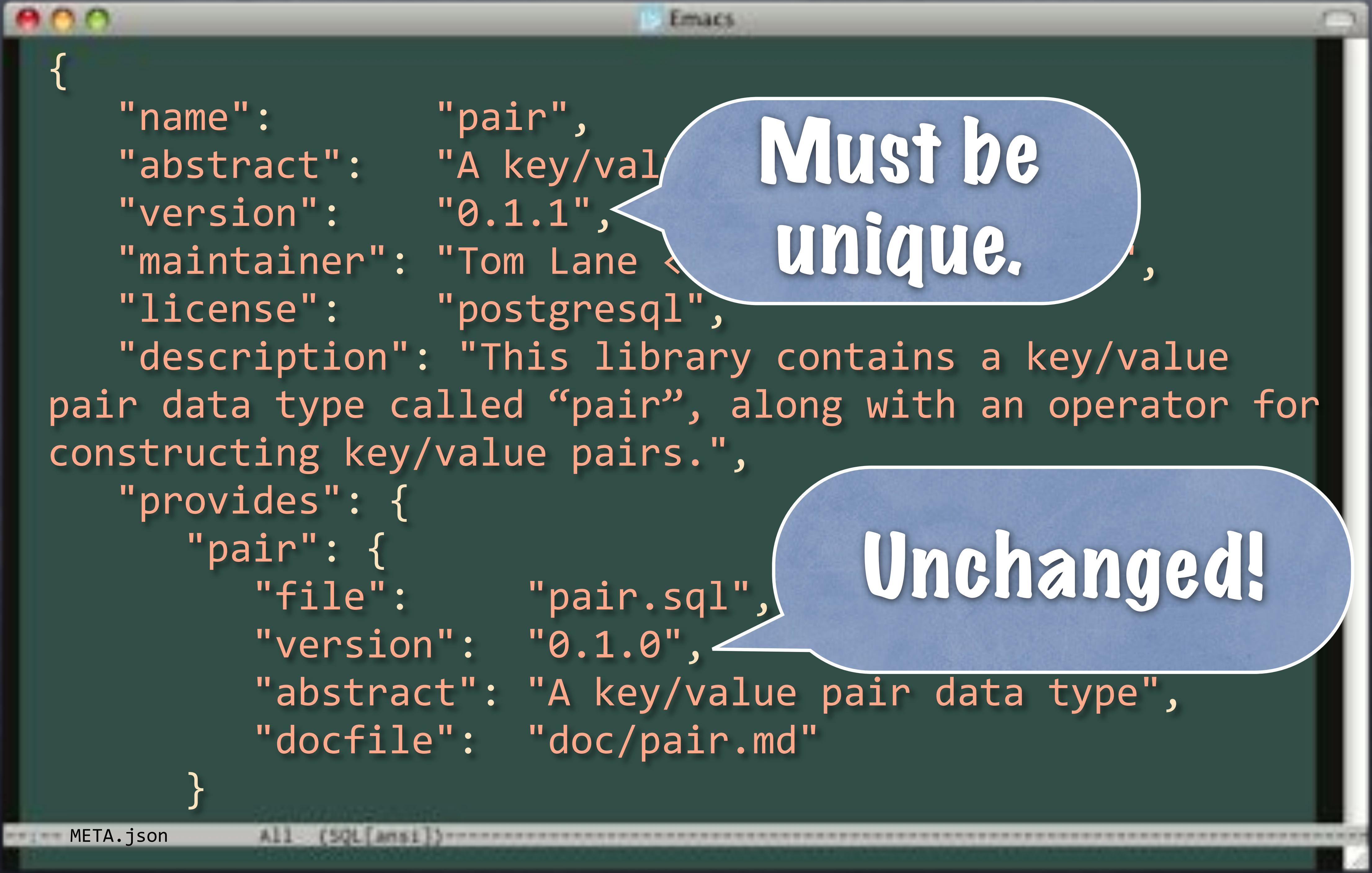
```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.1",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "description": "This library contains a key/value
pair data type called “pair”, along with an operator for
constructing key/value pairs.",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0",
         "abstract": "A key/value pair data type",
         "docfile":  "doc/pair.md"
      }
```

Update Version



```
{
   "name":       "pair",
   "abstract":   "A key/val
   "version":    "0.1.1",
   "maintainer": "Tom Lane <                          ",
   "license":    "postgresql",
   "description": "This library contains a key/value
pair data type called “pair”, along with an operator for
constructing key/value pairs.",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0",
         "abstract": "A key/value pair data type",
         "docfile":  "doc/pair.md"
      }
```

Update Version

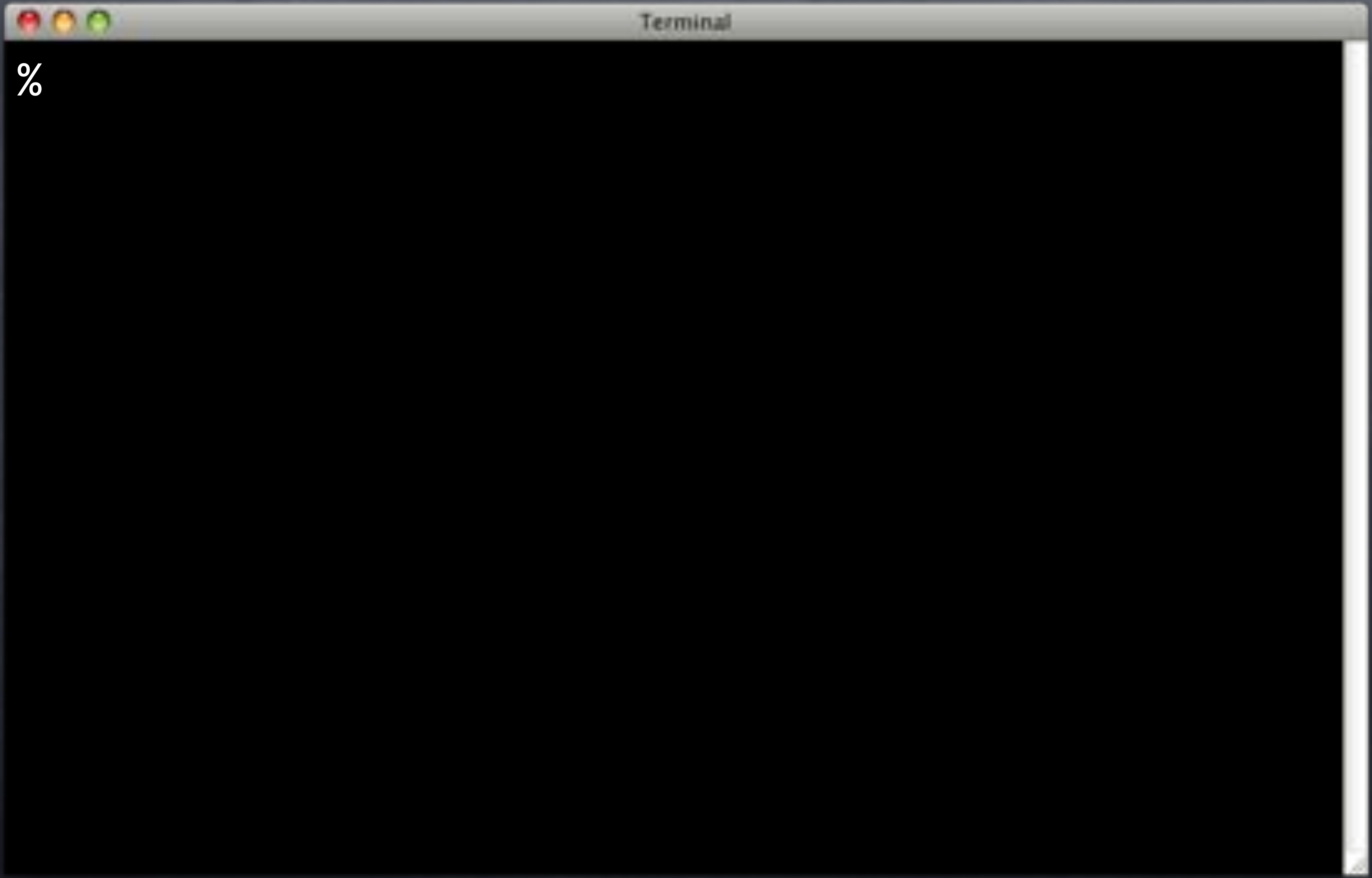


```
{
   "name":       "pair",
   "abstract":   "A key/val
   "version":    "0.1.1",
   "maintainer": "Tom Lane <
   "license":    "postgresql",
   "description": "This library contains a key/value
pair data type called “pair”, along with an operator for
constructing key/value pairs.",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0",
         "abstract": "A key/value pair data type",
         "docfile":  "doc/pair.md"
      }
```

PGXN Meta Spec

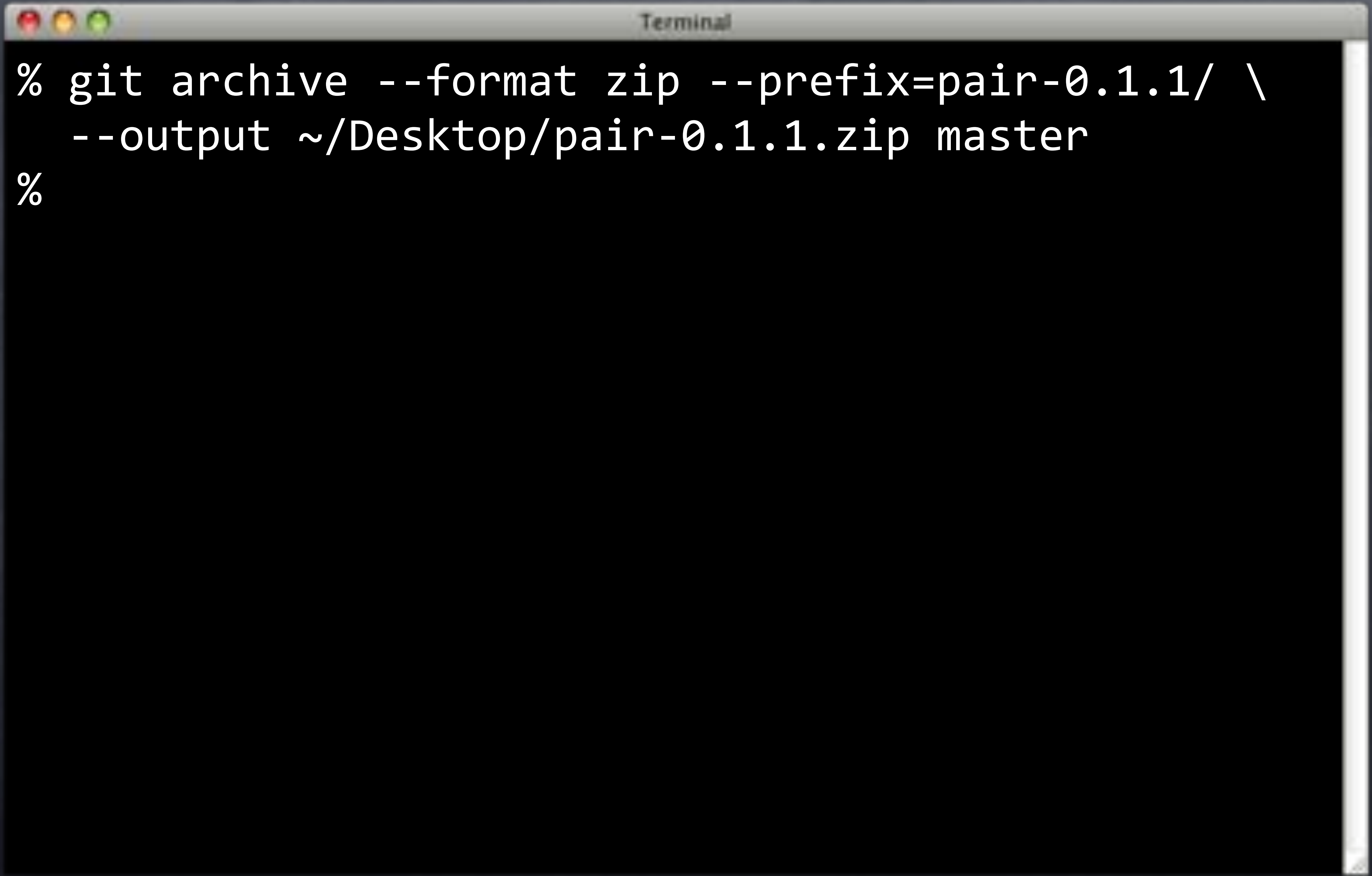
http://pgxn.org/spec/

Package it up



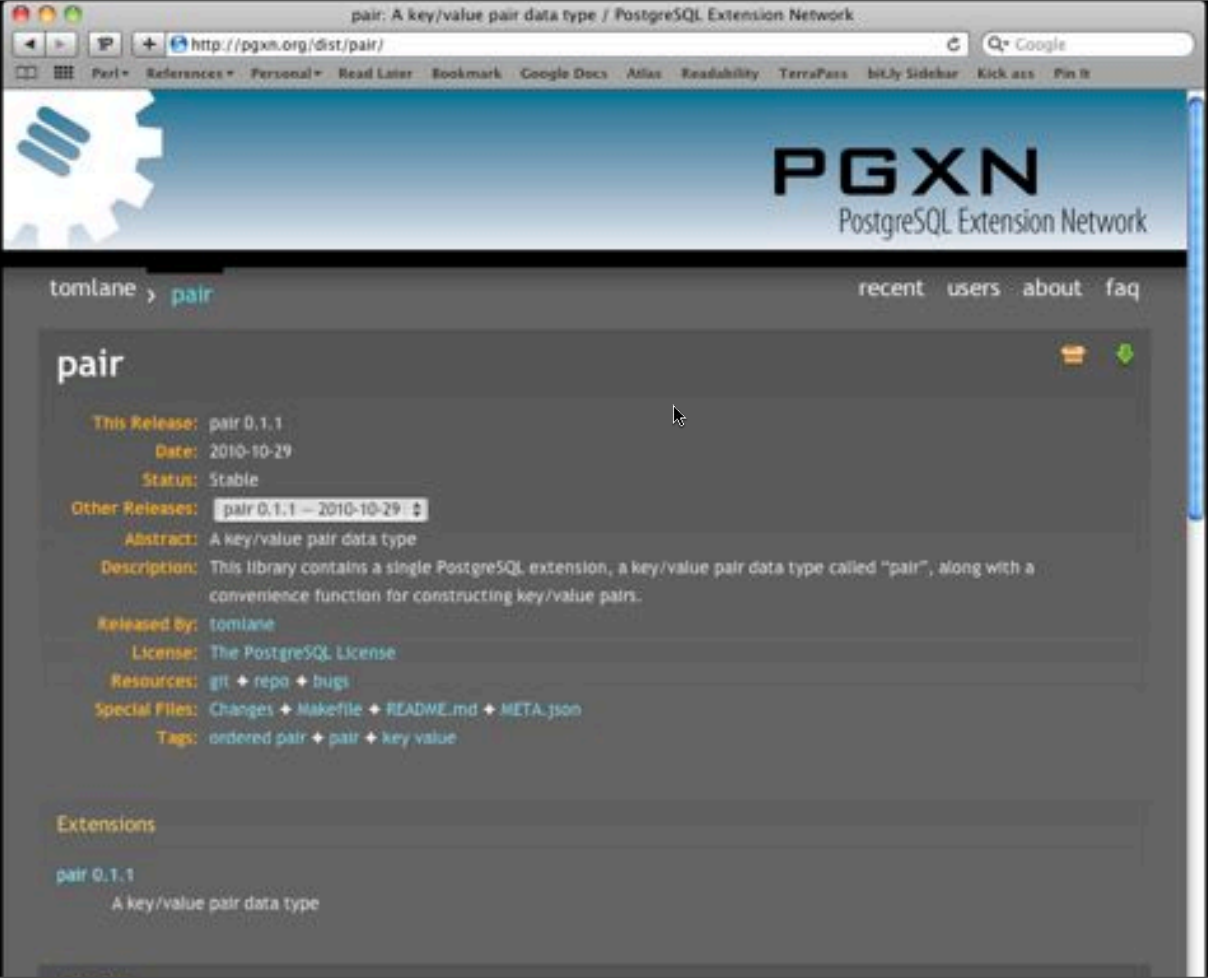
```
%
```

Package it up



```
% git archive --format zip --prefix=pair-0.1.1/ \
  --output ~/Desktop/pair-0.1.1.zip master
%
```

Release it

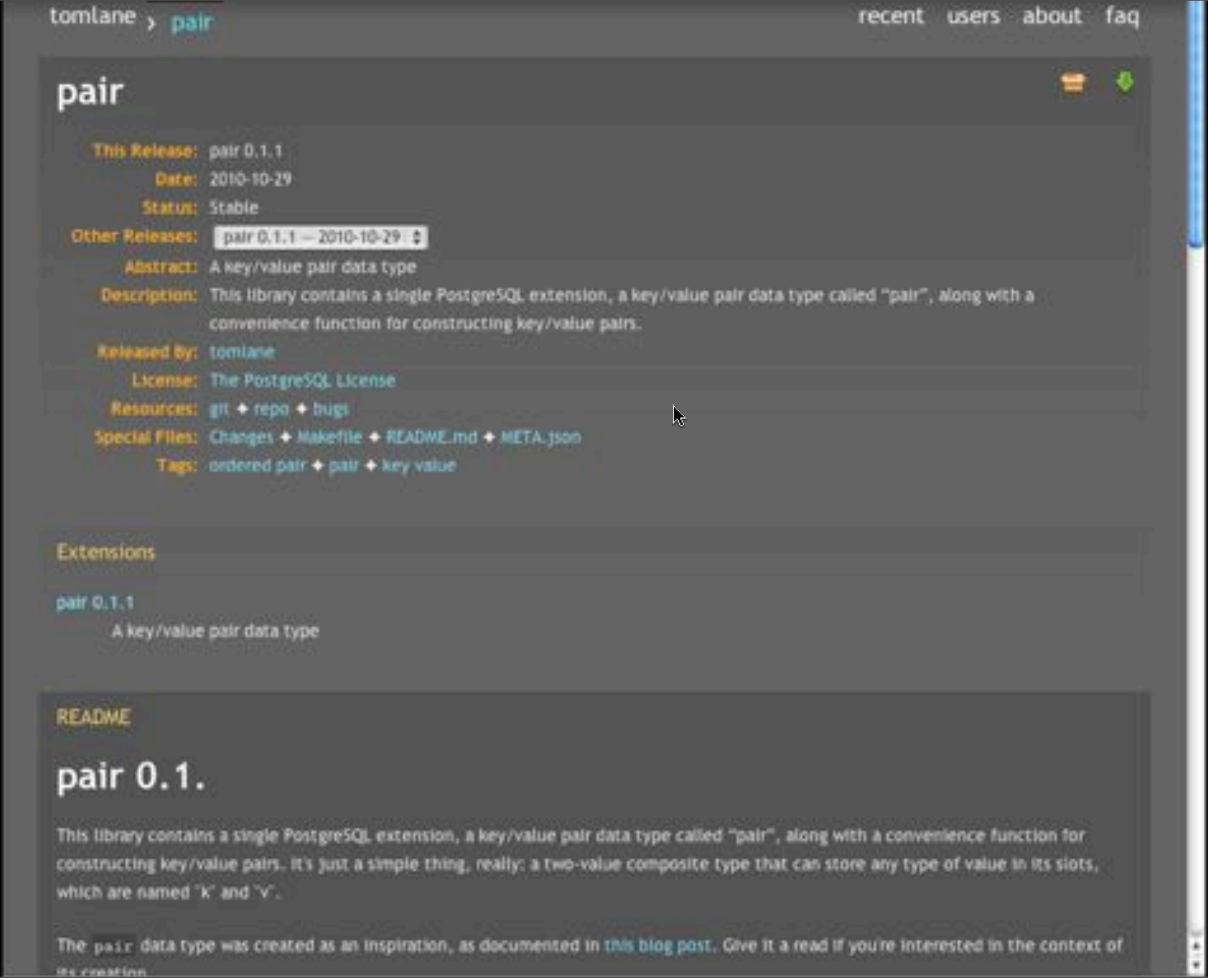

pair: A key/value pair data type / PostgreSQL Extension Network
http://pgxn.org/dist/pair/
Perl References Personal Read Later Bookmark Google Docs Atlas Readability TerraPass bit.ly Sidebar Kick ass Pin It
PGXN
PostgreSQL Extension Network
tomlane › pair
recent users about faq

pair

This Release: pair 0.1.1
Date: 2010-10-29
Status: Stable
Other Releases: pair 0.1.1 — 2010-10-29
Abstract: A key/value pair data type
Description: This library contains a single PostgreSQL extension, a key/value pair data type called "pair", along with a convenience function for constructing key/value pairs.
Released By: tomlane
License: The PostgreSQL License
Resources: git ✦ repo ✦ bugs
Special Files: Changes ✦ Makefile ✦ README.md ✦ META.json
Tags: ordered pair ✦ pair ✦ key value

Extensions

pair 0.1.1
A key/value pair data type



pair

This Release: pair 0.1.1
Date: 2010-10-29
Status: Stable
Other Releases: pair 0.1.1 — 2010-10-29
Abstract: A key/value pair data type
Description: This library contains a single PostgreSQL extension, a key/value pair data type called "pair", along with a convenience function for constructing key/value pairs.
Released by: tomlane
License: The PostgreSQL License
Resources: git ✦ repo ✦ bugs
Special Files: Changes ✦ Makefile ✦ README.md ✦ META.json
Tags: ordered pair ✦ pair ✦ key value

Extensions

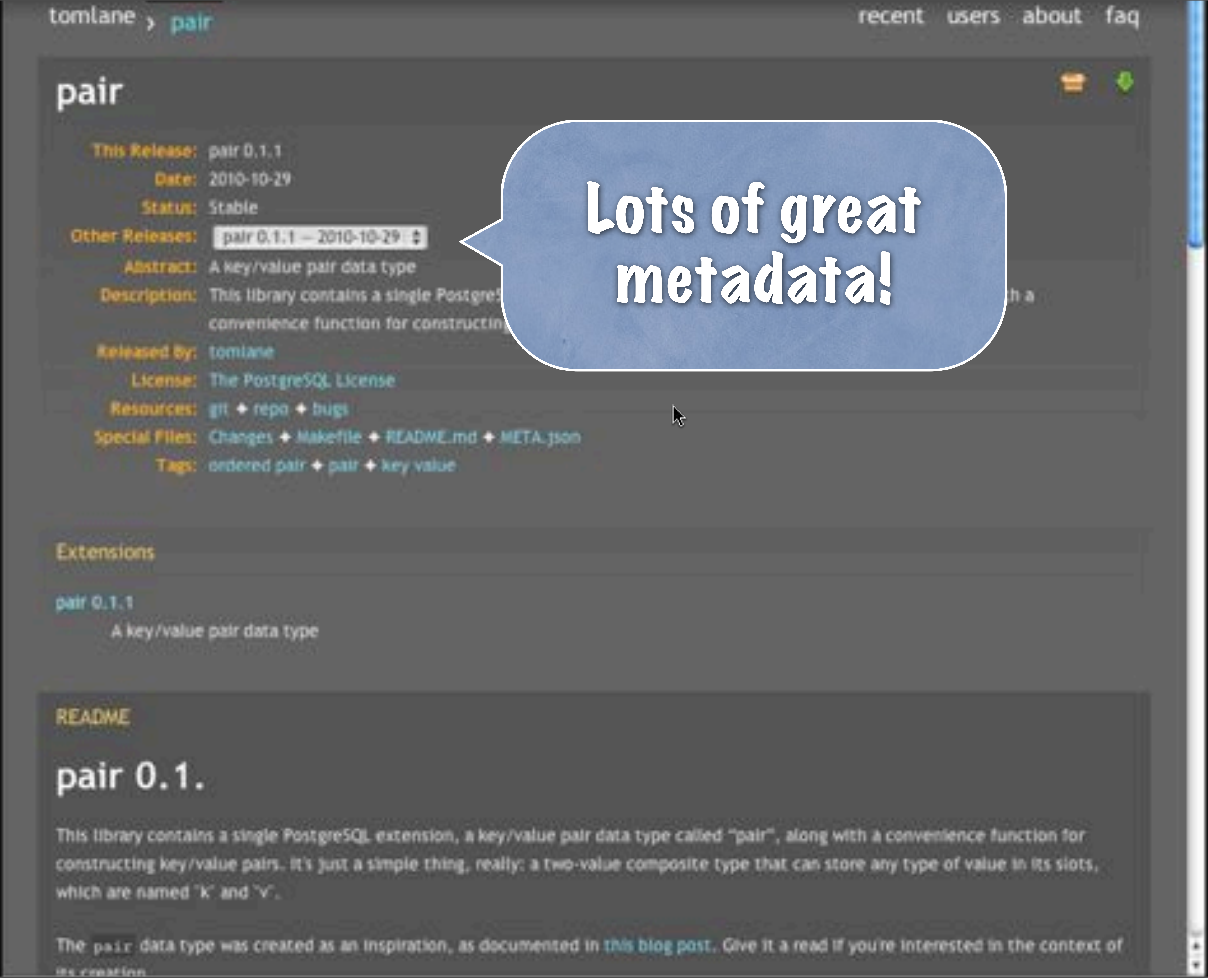
pair 0.1.1
A key/value pair data type

README

pair 0.1.

This library contains a single PostgreSQL extension, a key/value pair data type called "pair", along with a convenience function for constructing key/value pairs. It's just a simple thing, really: a two-value composite type that can store any type of value in its slots, which are named "k" and "v".

The `pair` data type was created as an inspiration, as documented in this blog post. Give it a read if you're interested in the context of



tomlane › pair　　recent　users　about　faq

pair

This Release: pair 0.1.1
Date: 2010-10-29
Status: Stable
Other Releases: pair 0.1.1 — 2010-10-29
Abstract: A key/value pair data type
Description: This library contains a single Postgre … th a convenience function for constructin
Released by: tomlane
License: The PostgreSQL License
Resources: git ✦ repo ✦ bugs
Special Files: Changes ✦ Makefile ✦ README.md ✦ META.json
Tags: ordered pair ✦ pair ✦ key value

Extensions

pair 0.1.1
A key/value pair data type

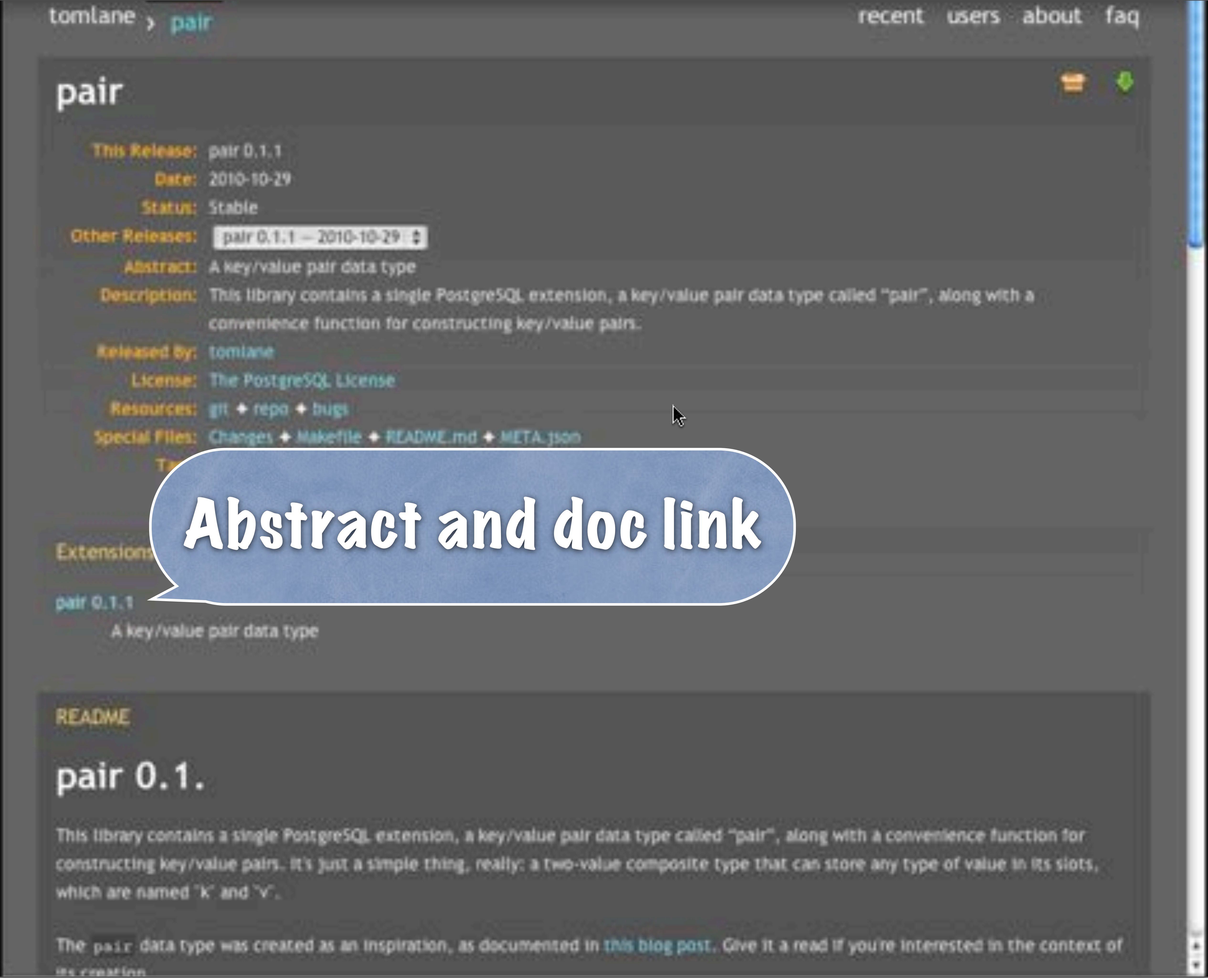
README

pair 0.1.

This library contains a single PostgreSQL extension, a key/value pair data type called "pair", along with a convenience function for constructing key/value pairs. It's just a simple thing, really: a two-value composite type that can store any type of value in its slots, which are named "k" and "v".

The `pair` data type was created as an inspiration, as documented in this blog post. Give it a read if you're interested in the context of its creation

Lots of great metadata!



tomlane › pair

recent users about faq

pair

This Release: pair 0.1.1
Date: 2010-10-29
Status: Stable
Other Releases: pair 0.1.1 — 2010-10-29
Abstract: A key/value pair data type
Description: This library contains a single PostgreSQL extension, a key/value pair data type called "pair", along with a convenience function for constructing key/value pairs.
Released By: tomlane
License: The PostgreSQL License
Resources: git ✦ repo ✦ bugs
Special Files: Changes ✦ Makefile ✦ README.md ✦ META.json

Extensions

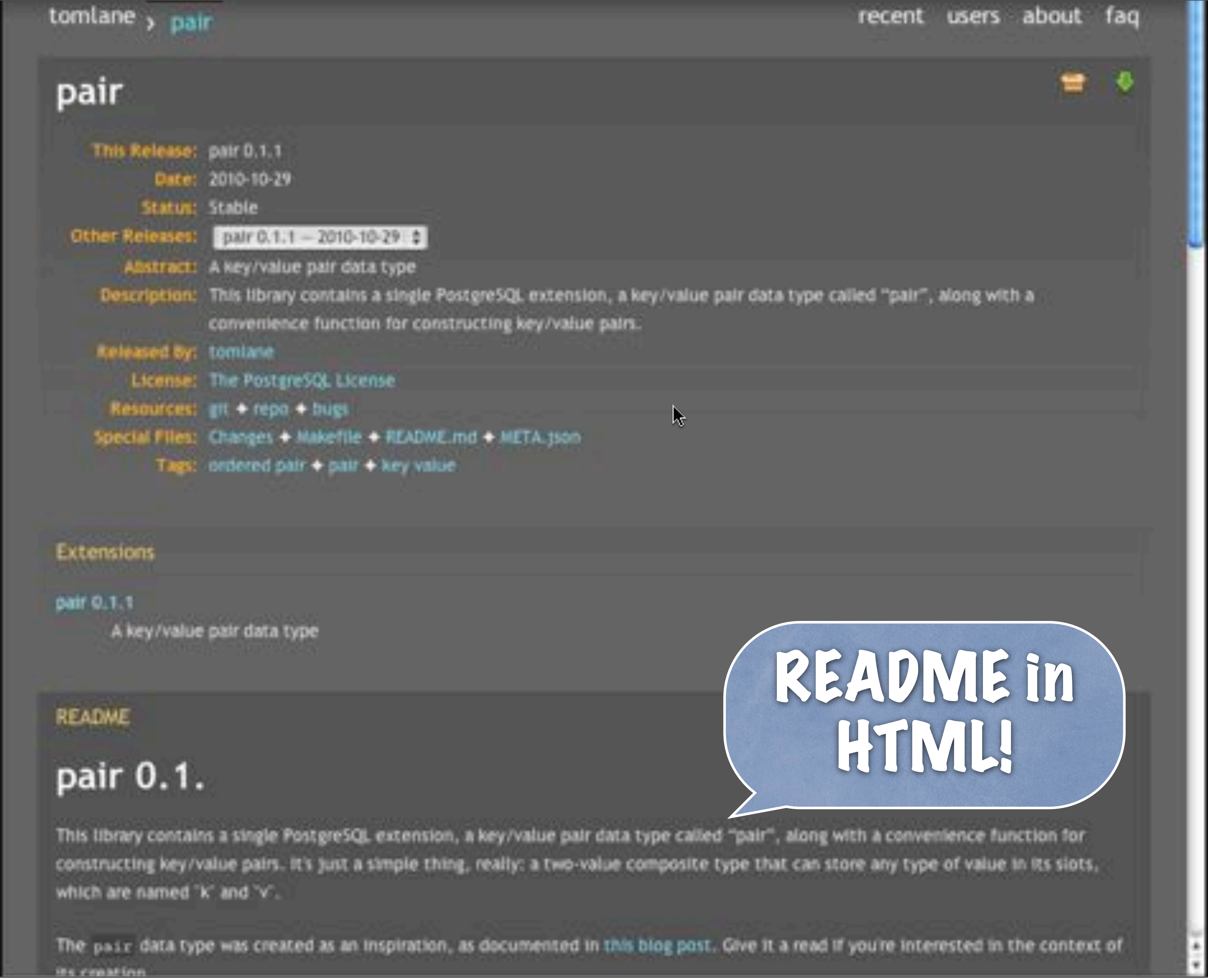
pair 0.1.1
A key/value pair data type

README

pair 0.1.

This library contains a single PostgreSQL extension, a key/value pair data type called "pair", along with a convenience function for constructing key/value pairs. It's just a simple thing, really: a two-value composite type that can store any type of value in its slots, which are named "k" and "v".

The `pair` data type was created as an inspiration, as documented in this blog post. Give it a read if you're interested in the context of its creation.



tomlane › pair recent users about faq

pair

This Release: pair 0.1.1
Date: 2010-10-29
Status: Stable
Other Releases: pair 0.1.1 — 2010-10-29
Abstract: A key/value pair data type
Description: This library contains a single PostgreSQL extension, a key/value pair data type called "pair", along with a convenience function for constructing key/value pairs.
Released by: tomlane
License: The PostgreSQL License
Resources: git ✦ repo ✦ bugs
Special Files: Changes ✦ Makefile ✦ README.md ✦ META.json
Tags: ordered pair ✦ pair ✦ key value

Extensions

pair 0.1.1
A key/value pair data type

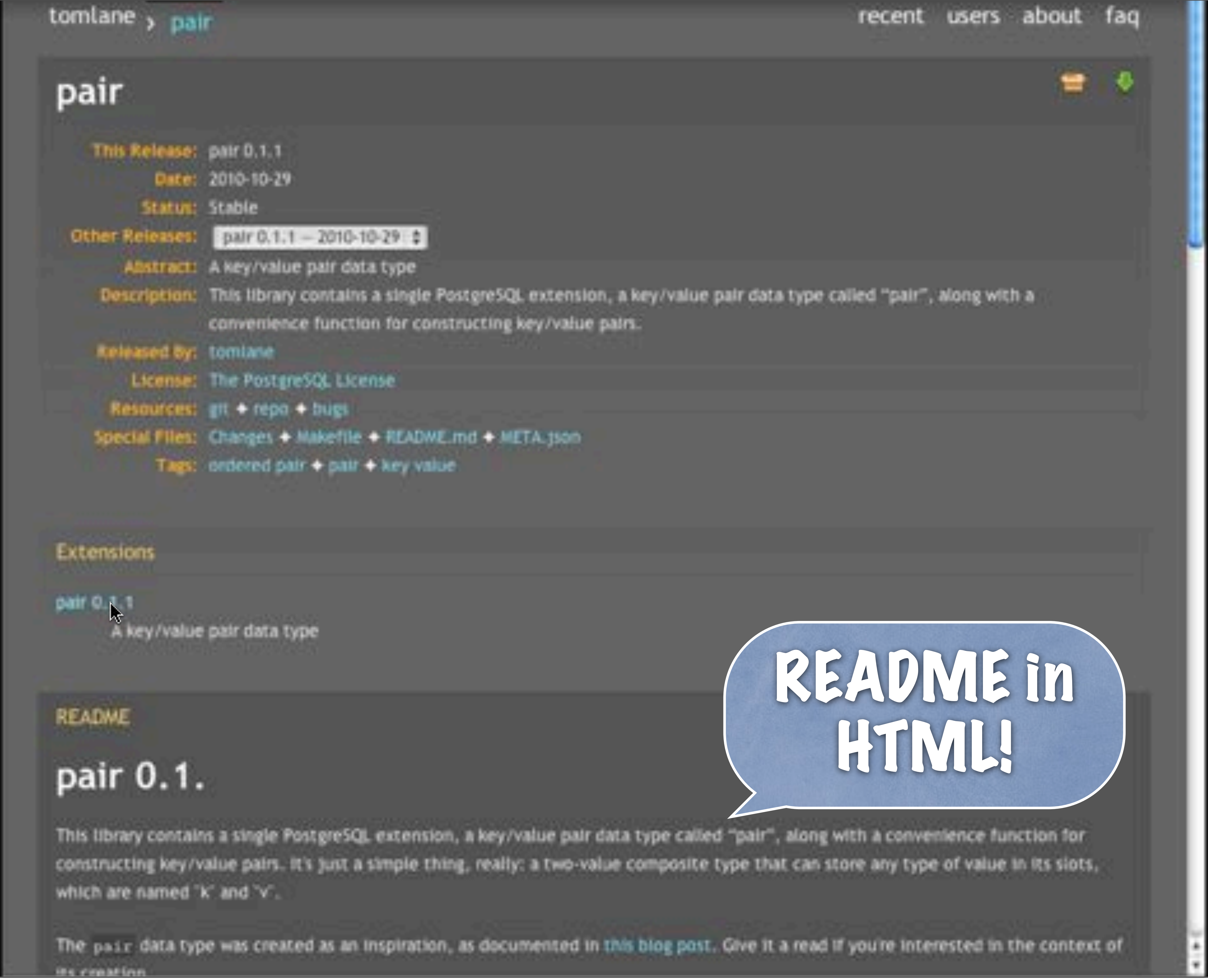
README

pair 0.1.

This library contains a single PostgreSQL extension, a key/value pair data type called "pair", along with a convenience function for constructing key/value pairs. It's just a simple thing, really: a two-value composite type that can store any type of value in its slots, which are named "k" and "v".

The `pair` data type was created as an inspiration, as documented in this blog post. Give it a read if you're interested in the context of its creation

README in HTML!


tomlane › pair

recent users about faq

pair

This Release:	pair 0.1.1
Date:	2010-10-29
Status:	Stable
Other Releases:	pair 0.1.1 — 2010-10-29
Abstract:	A key/value pair data type
Description:	This library contains a single PostgreSQL extension, a key/value pair data type called "pair", along with a convenience function for constructing key/value pairs.
Released by:	tomlane
License:	The PostgreSQL License
Resources:	git ✦ repo ✦ bugs
Special Files:	Changes ✦ Makefile ✦ README.md ✦ META.json
Tags:	ordered pair ✦ pair ✦ key value

Extensions

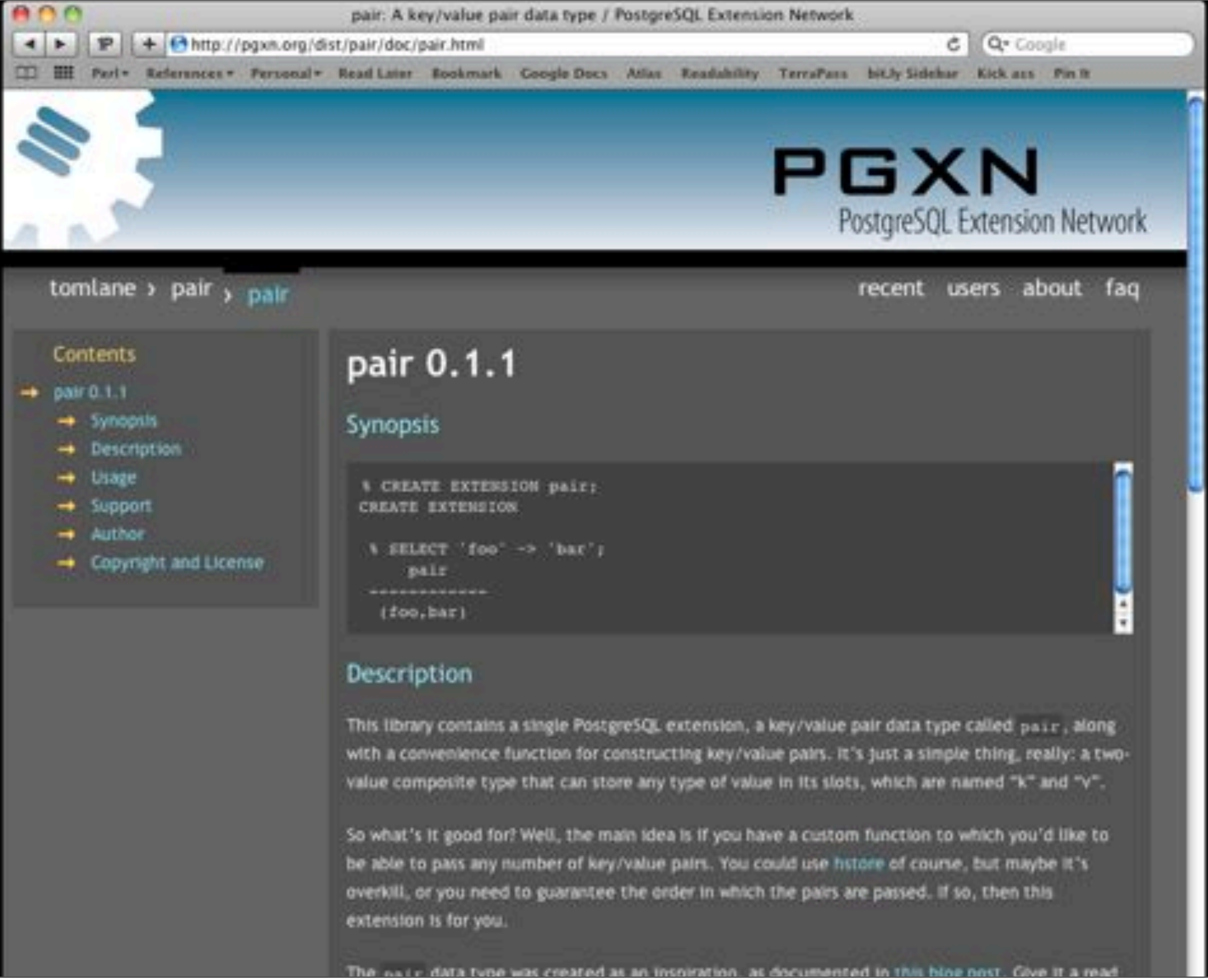
pair 0.1.1
A key/value pair data type

README

pair 0.1.

This library contains a single PostgreSQL extension, a key/value pair data type called "pair", along with a convenience function for constructing key/value pairs. It's just a simple thing, really: a two-value composite type that can store any type of value in its slots, which are named "k" and "v".

The `pair` data type was created as an inspiration, as documented in this blog post. Give it a read if you're interested in the context of its creation.


pair: A key/value pair data type / PostgreSQL Extension Network
http://pgxn.org/dist/pair/doc/pair.html
PGXN
PostgreSQL Extension Network
tomlane › pair › pair
recent users about faq

pair 0.1.1

Synopsis

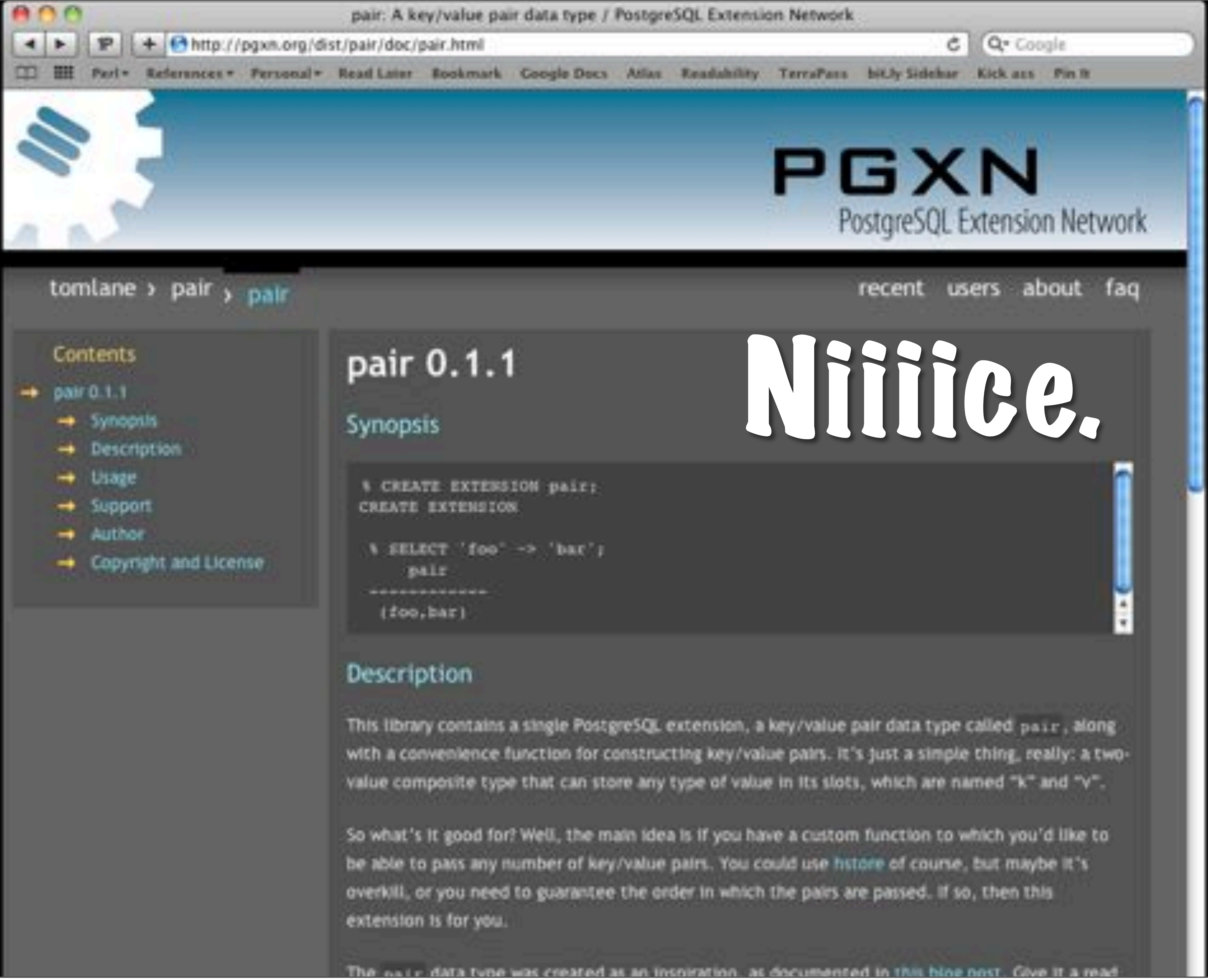
```
% CREATE EXTENSION pair;
CREATE EXTENSION

 % SELECT 'foo' -> 'bar';
     pair
 ------------
  (foo,bar)
```

Description

This library contains a single PostgreSQL extension, a key/value pair data type called `pair`, along with a convenience function for constructing key/value pairs. It's just a simple thing, really: a two-value composite type that can store any type of value in its slots, which are named "k" and "v".

So what's it good for? Well, the main idea is if you have a custom function to which you'd like to be able to pass any number of key/value pairs. You could use hstore of course, but maybe it's overkill, or you need to guarantee the order in which the pairs are passed. If so, then this extension is for you.

The `pair` data type was created as an inspiration, as documented in this blog post. Give it a read


pair: A key/value pair data type / PostgreSQL Extension Network
http://pgxn.org/dist/pair/doc/pair.html
PGXN
PostgreSQL Extension Network
tomlane › pair › pair
recent users about faq
Contents
pair 0.1.1
Synopsis
Description
Usage
Support
Author
Copyright and License

Niiiice.

pair 0.1.1

Synopsis

```
% CREATE EXTENSION pair;
CREATE EXTENSION

 % SELECT 'foo' -> 'bar';
     pair
 ------------
  (foo,bar)
```

Description

This library contains a single PostgreSQL extension, a key/value pair data type called `pair`, along with a convenience function for constructing key/value pairs. It's just a simple thing, really: a two-value composite type that can store any type of value in its slots, which are named "k" and "v".

So what's it good for? Well, the main idea is if you have a custom function to which you'd like to be able to pass any number of key/value pairs. You could use hstore of course, but maybe it's overkill, or you need to guarantee the order in which the pairs are passed. If so, then this extension is for you.

What Else?

What Else?

- Recommended file layout

What Else?

- Recommended file layout
- Standard Makefile format

What Else?

- Recommended file layout
- Standard Makefile format
- 9.1 CREATE EXTENSION support

What Else?

- Recommended file layout
- Standard Makefile format
- 9.1 CREATE EXTENSION support
  - With compatibility!

File Recommendations

File Recommendations

- SQL source in sql/

File Recommendations

- SQL source in sql/
- C source in src/

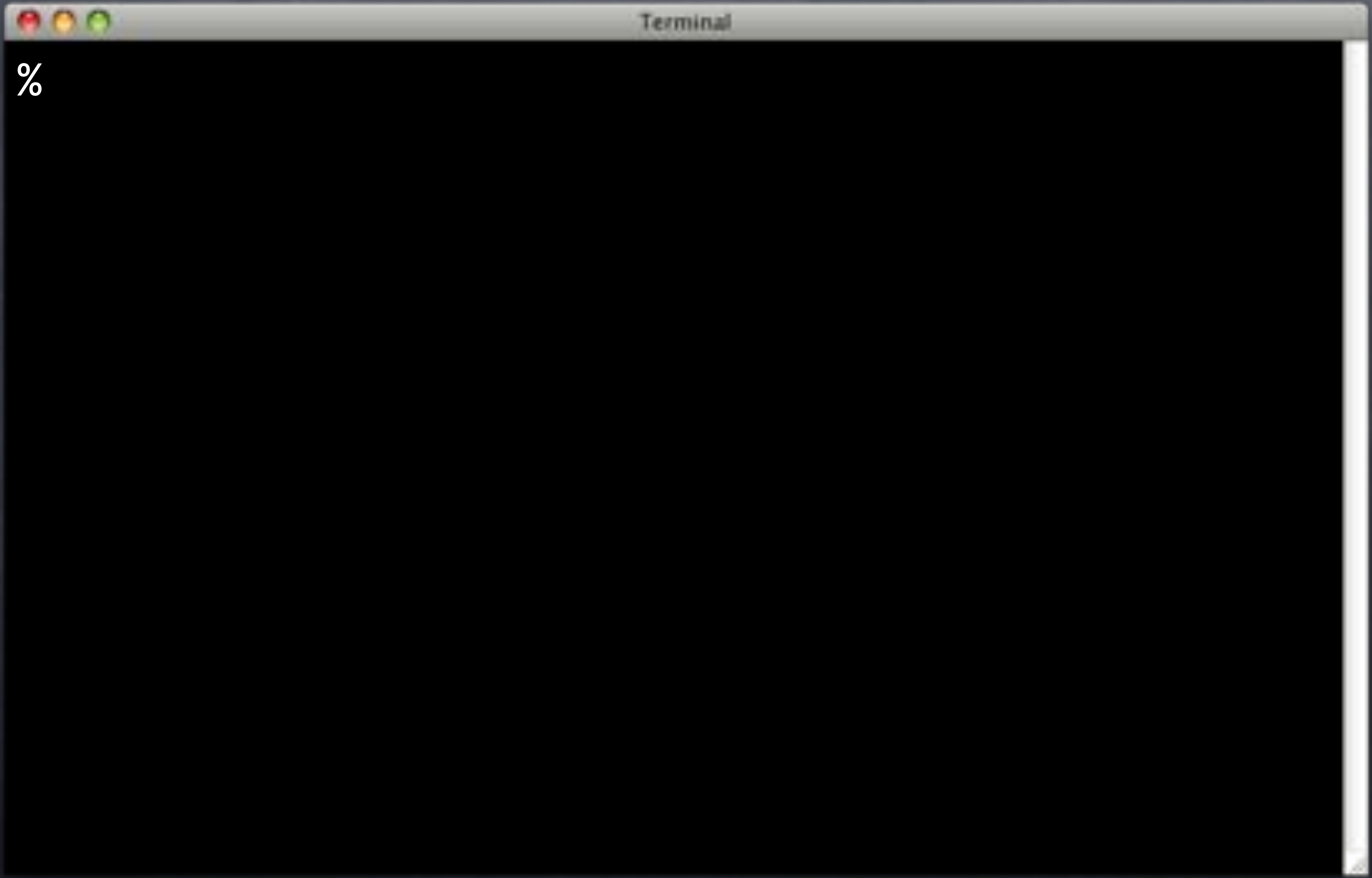
File Recommendations

- SQL source in sql/
- C source in src/
- Tests in test/

File Recommendations

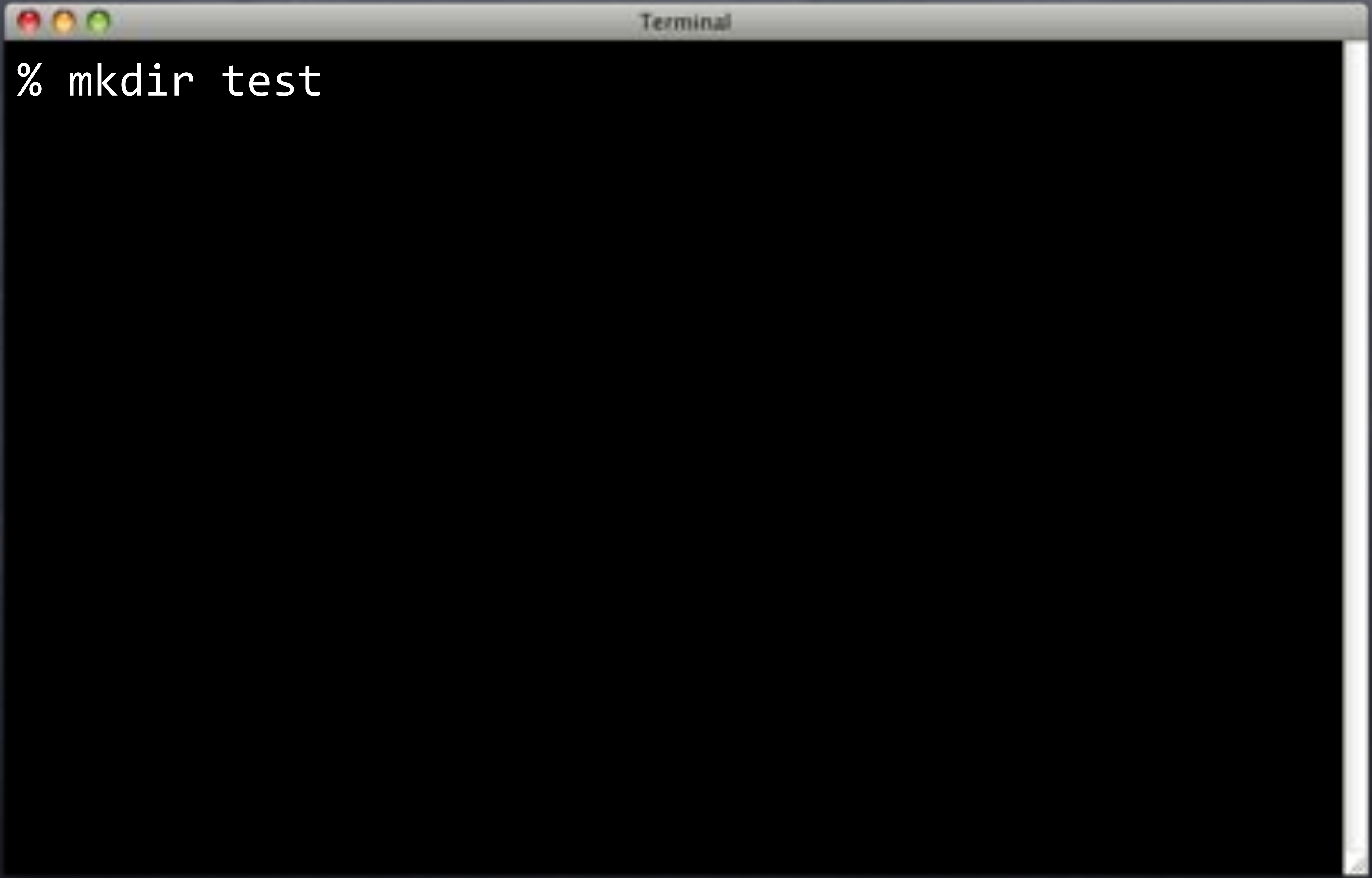
- SQL source in sql/
- C source in src/
- Tests in test/
- Documentation in doc/

Rearrange



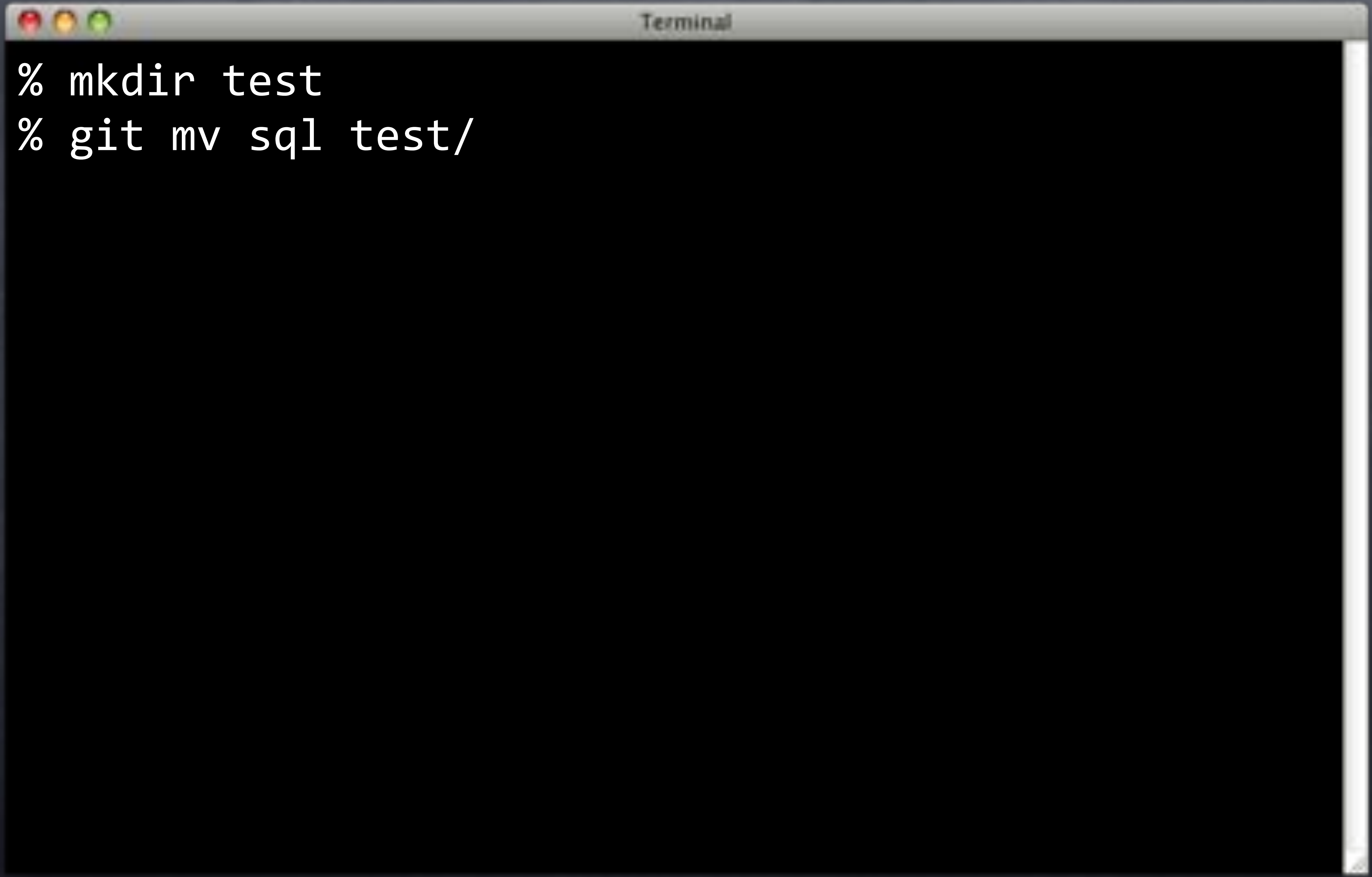
```
%
```

Rearrange



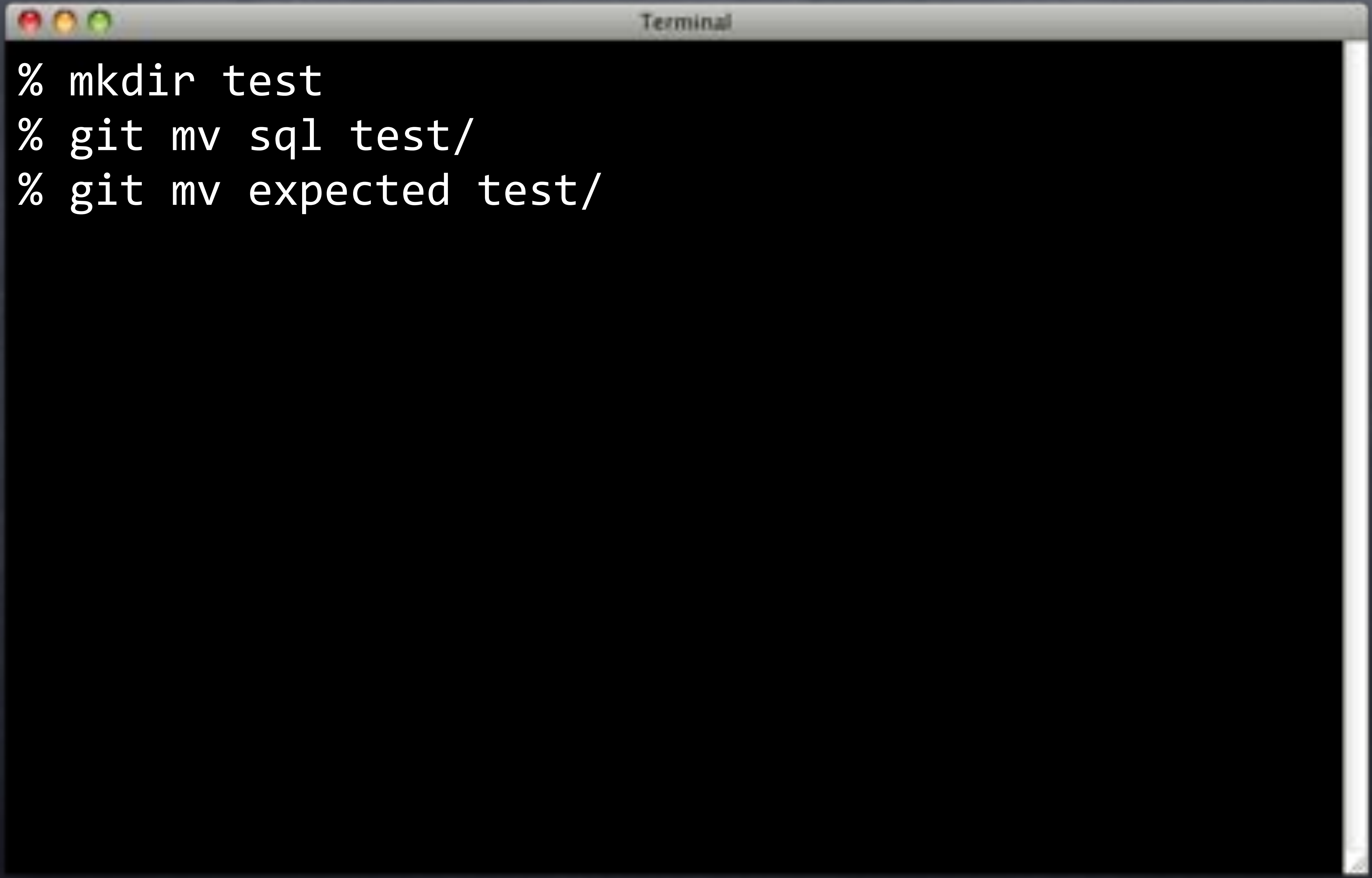
```
% mkdir test
```

Rearrange



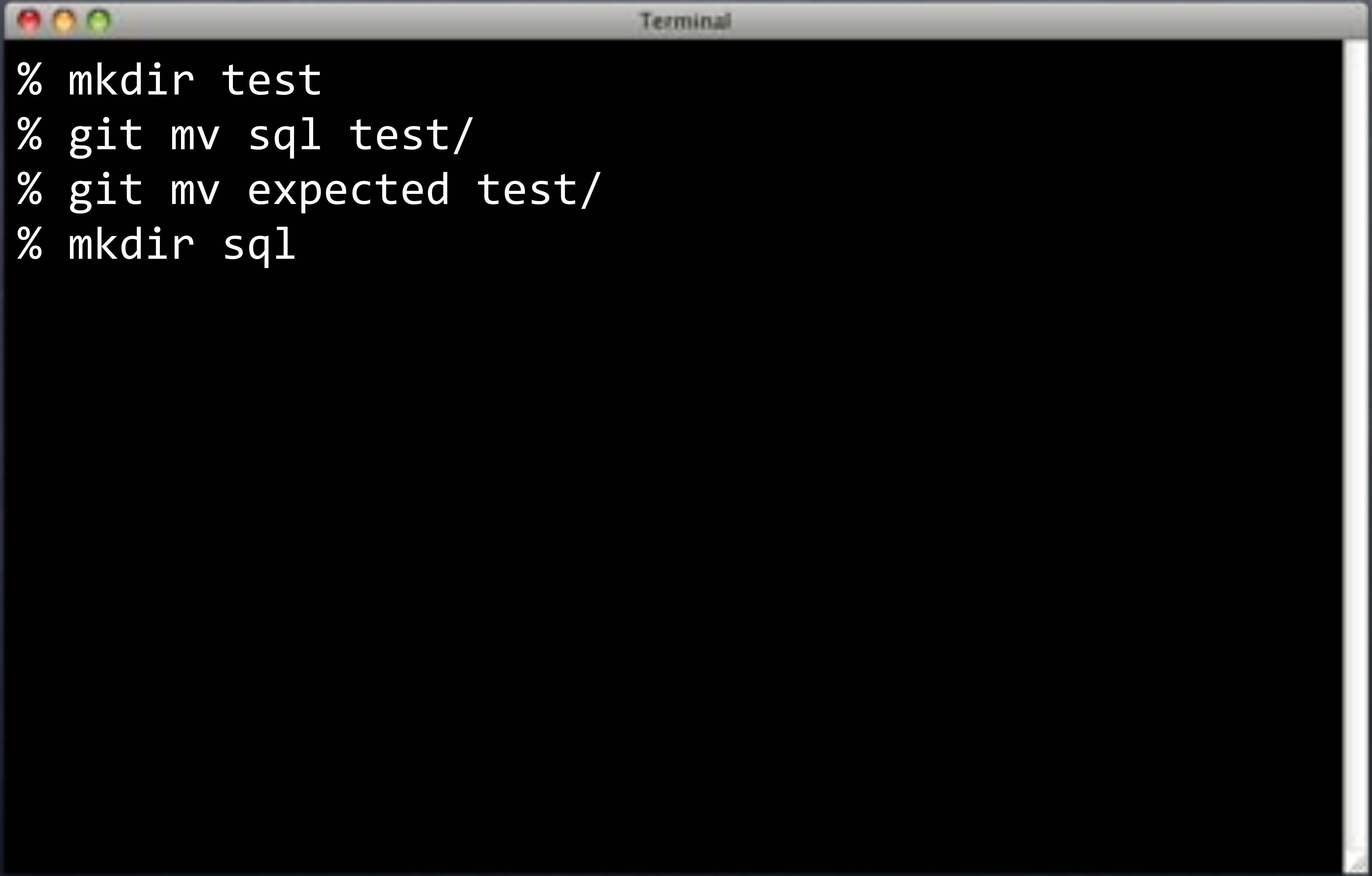
```
% mkdir test
% git mv sql test/
```

Rearrange


Terminal

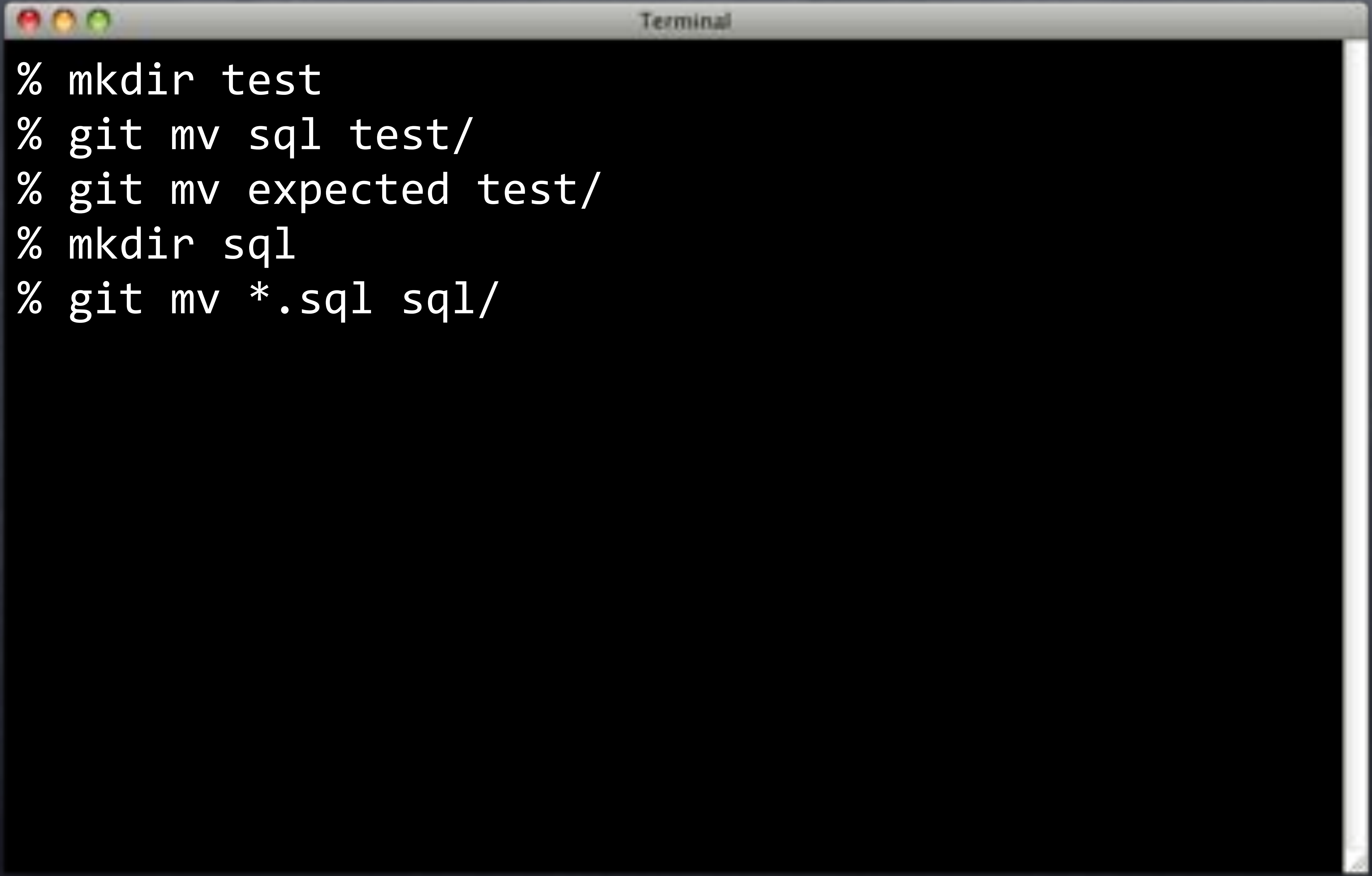
```
% mkdir test
% git mv sql test/
% git mv expected test/
```

Rearrange



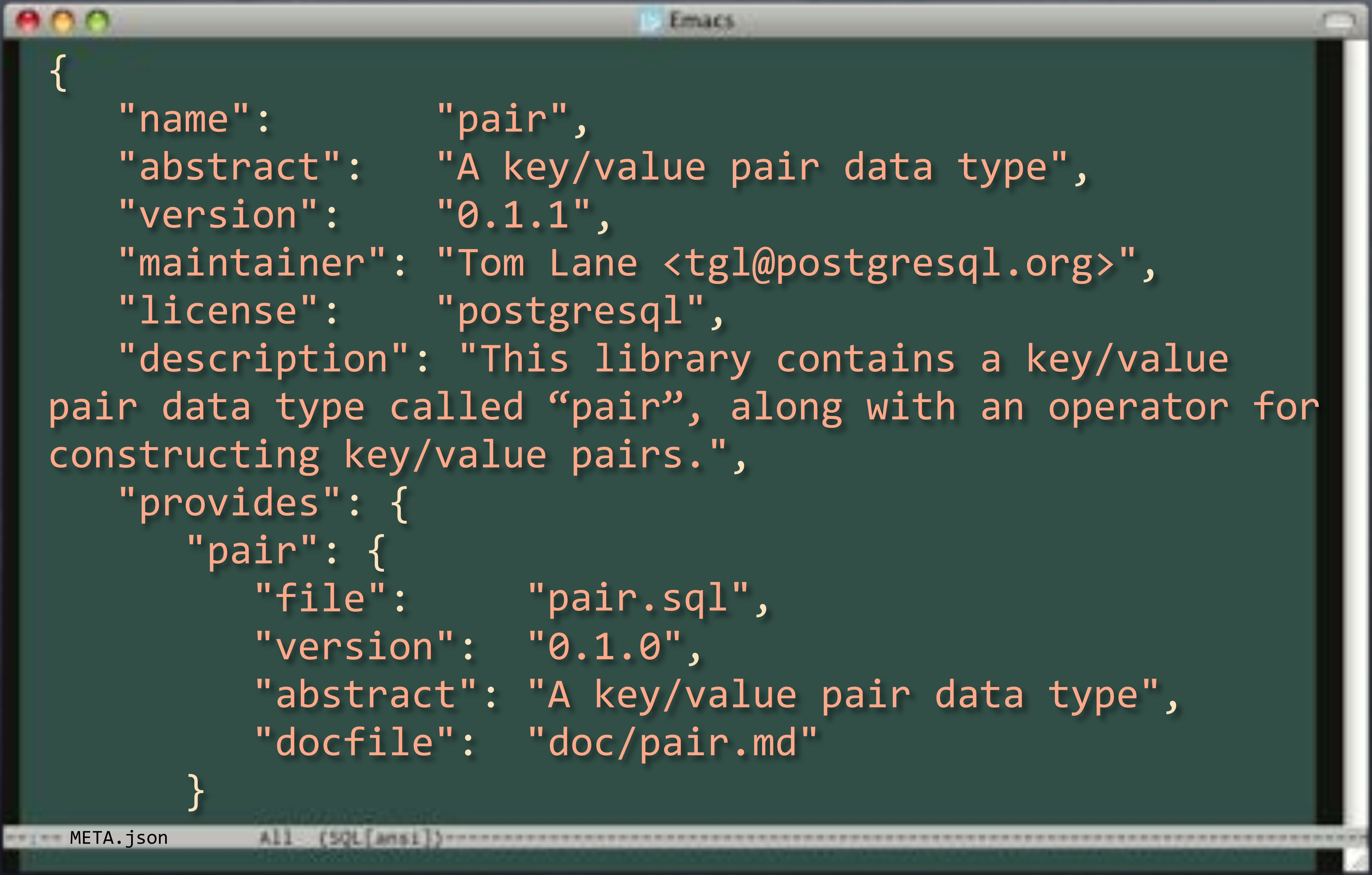
```
% mkdir test
% git mv sql test/
% git mv expected test/
% mkdir sql
```

Rearrange



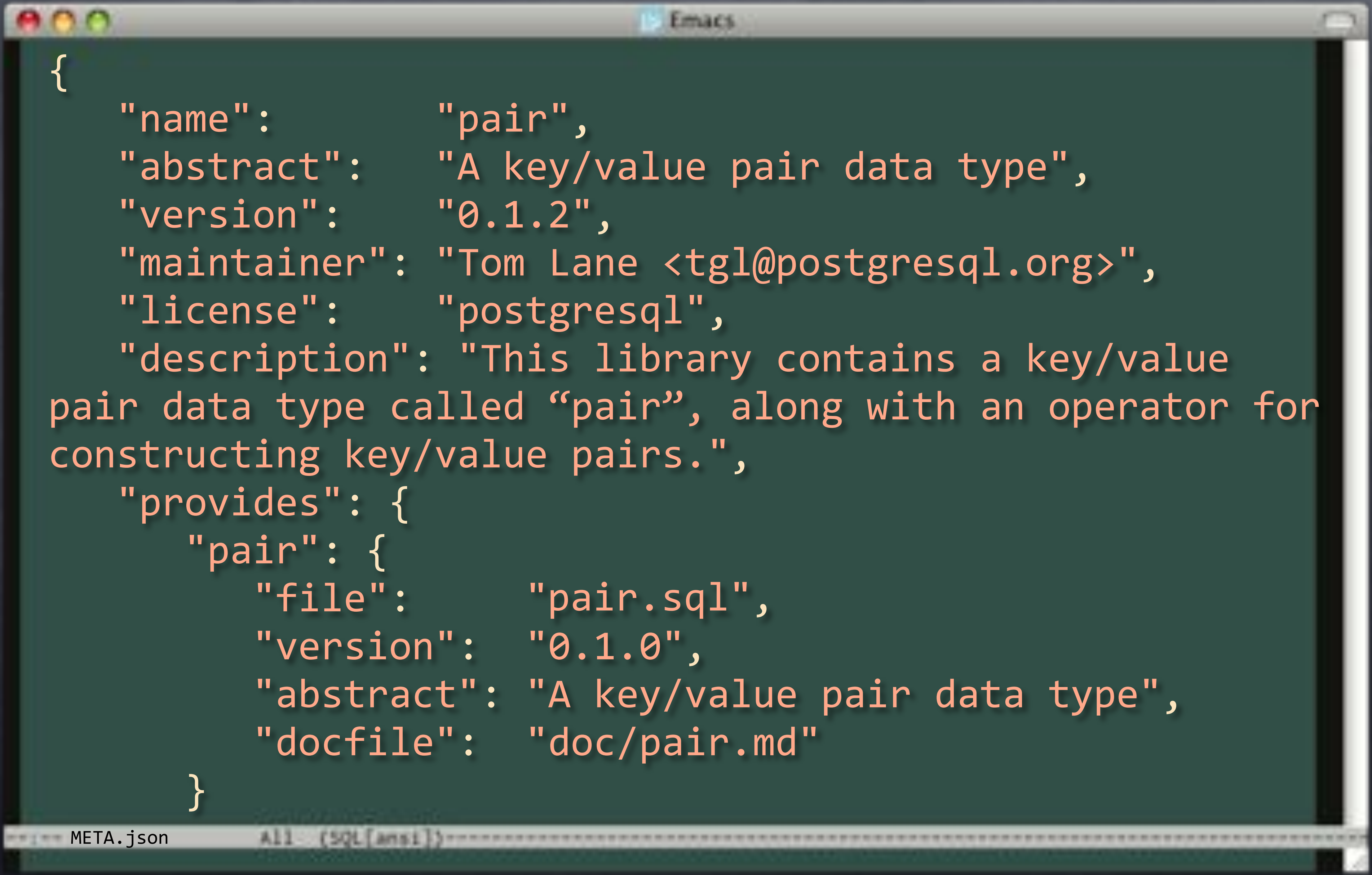
```
% mkdir test
% git mv sql test/
% git mv expected test/
% mkdir sql
% git mv *.sql sql/
```

Update Path



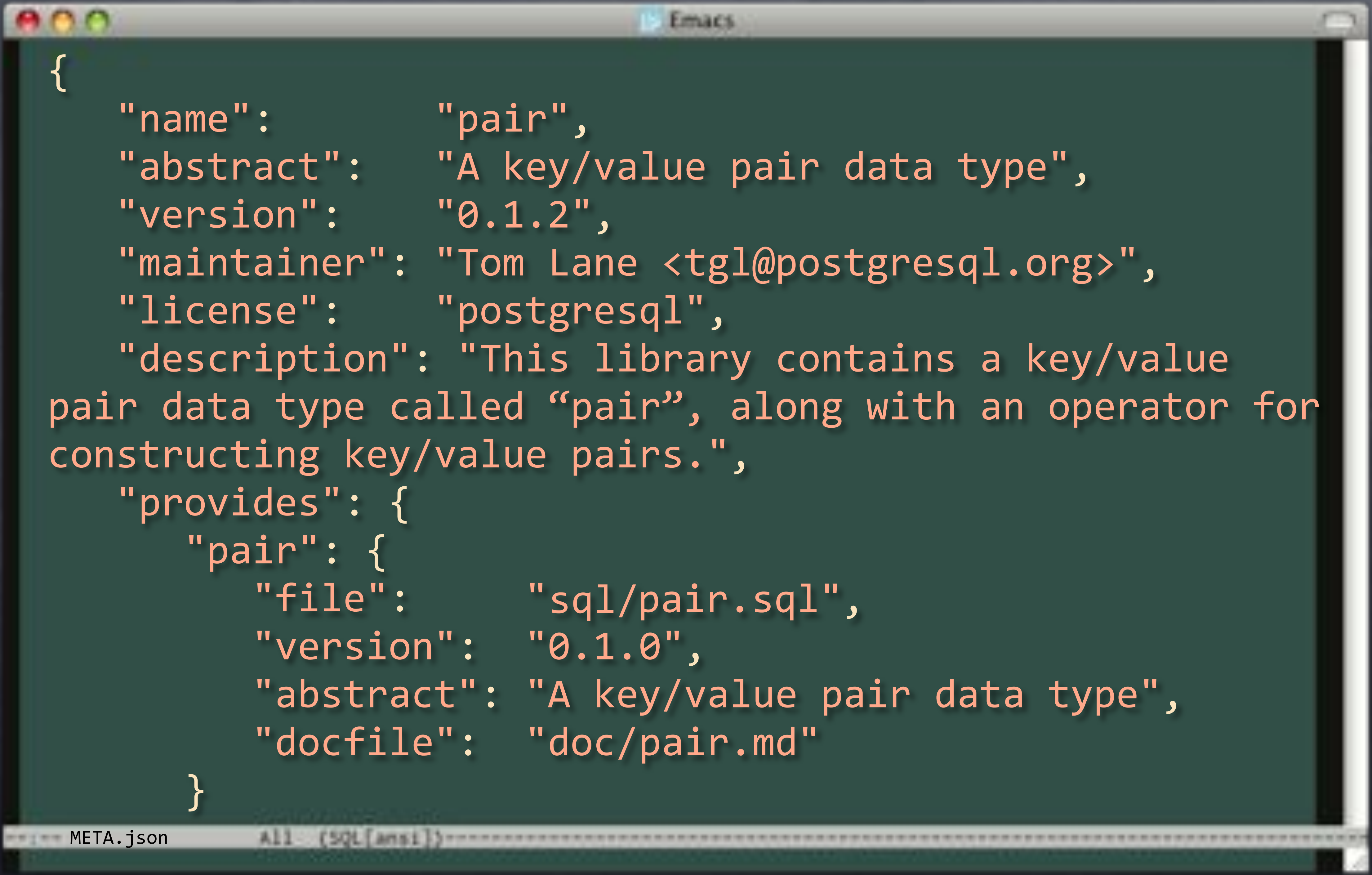
```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.1",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "description": "This library contains a key/value
pair data type called “pair”, along with an operator for
constructing key/value pairs.",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0",
         "abstract": "A key/value pair data type",
         "docfile":  "doc/pair.md"
      }
```

Update Path



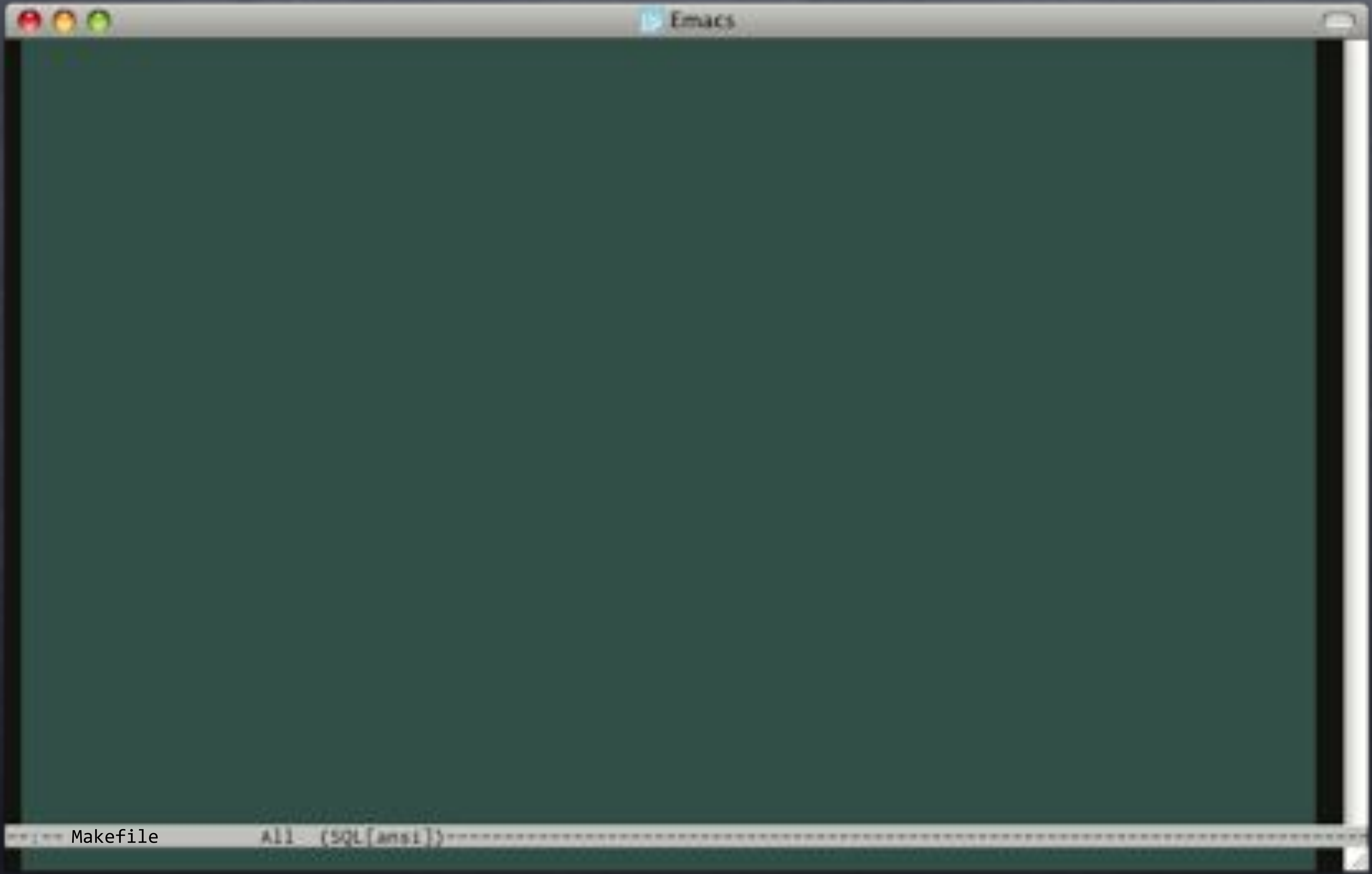
```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.2",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "description": "This library contains a key/value
pair data type called “pair”, along with an operator for
constructing key/value pairs.",
   "provides": {
      "pair": {
         "file":     "pair.sql",
         "version":  "0.1.0",
         "abstract": "A key/value pair data type",
         "docfile":  "doc/pair.md"
      }
```

Update Path

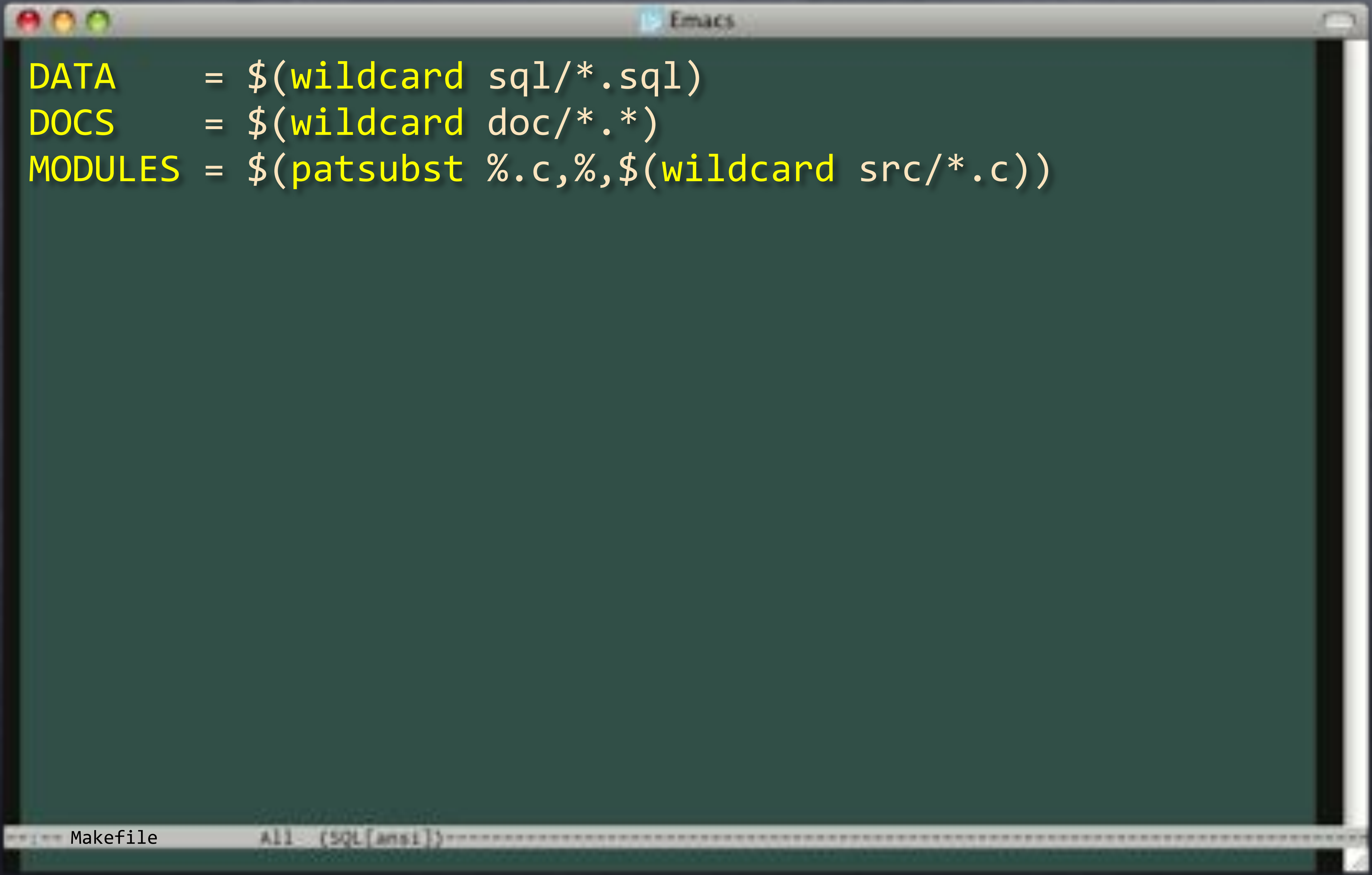


```
{
   "name":        "pair",
   "abstract":    "A key/value pair data type",
   "version":     "0.1.2",
   "maintainer": "Tom Lane <tgl@postgresql.org>",
   "license":     "postgresql",
   "description": "This library contains a key/value
pair data type called “pair”, along with an operator for
constructing key/value pairs.",
   "provides": {
      "pair": {
         "file":     "sql/pair.sql",
         "version":  "0.1.0",
         "abstract": "A key/value pair data type",
         "docfile":  "doc/pair.md"
      }
```

Makefile

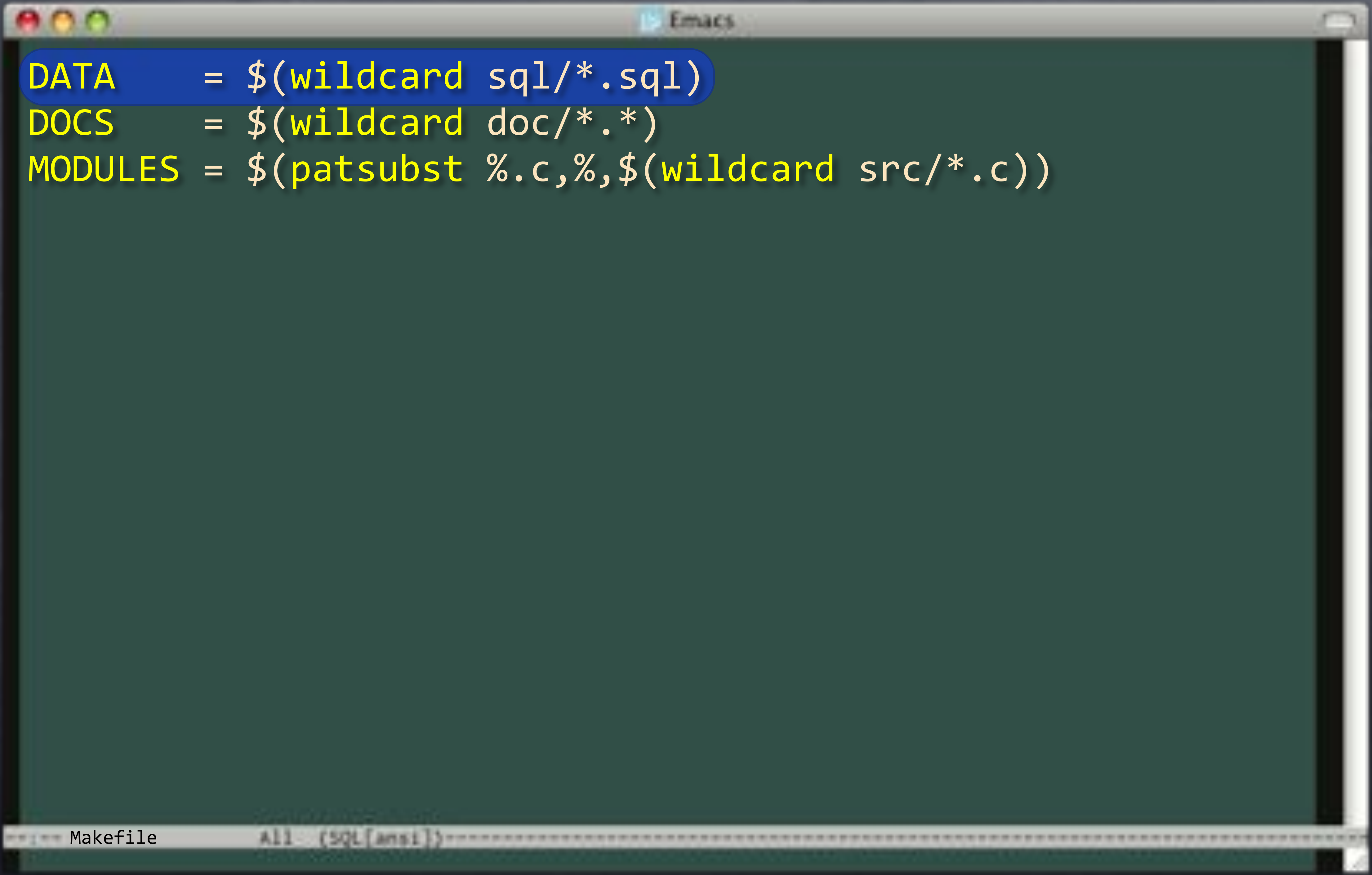


Makefile



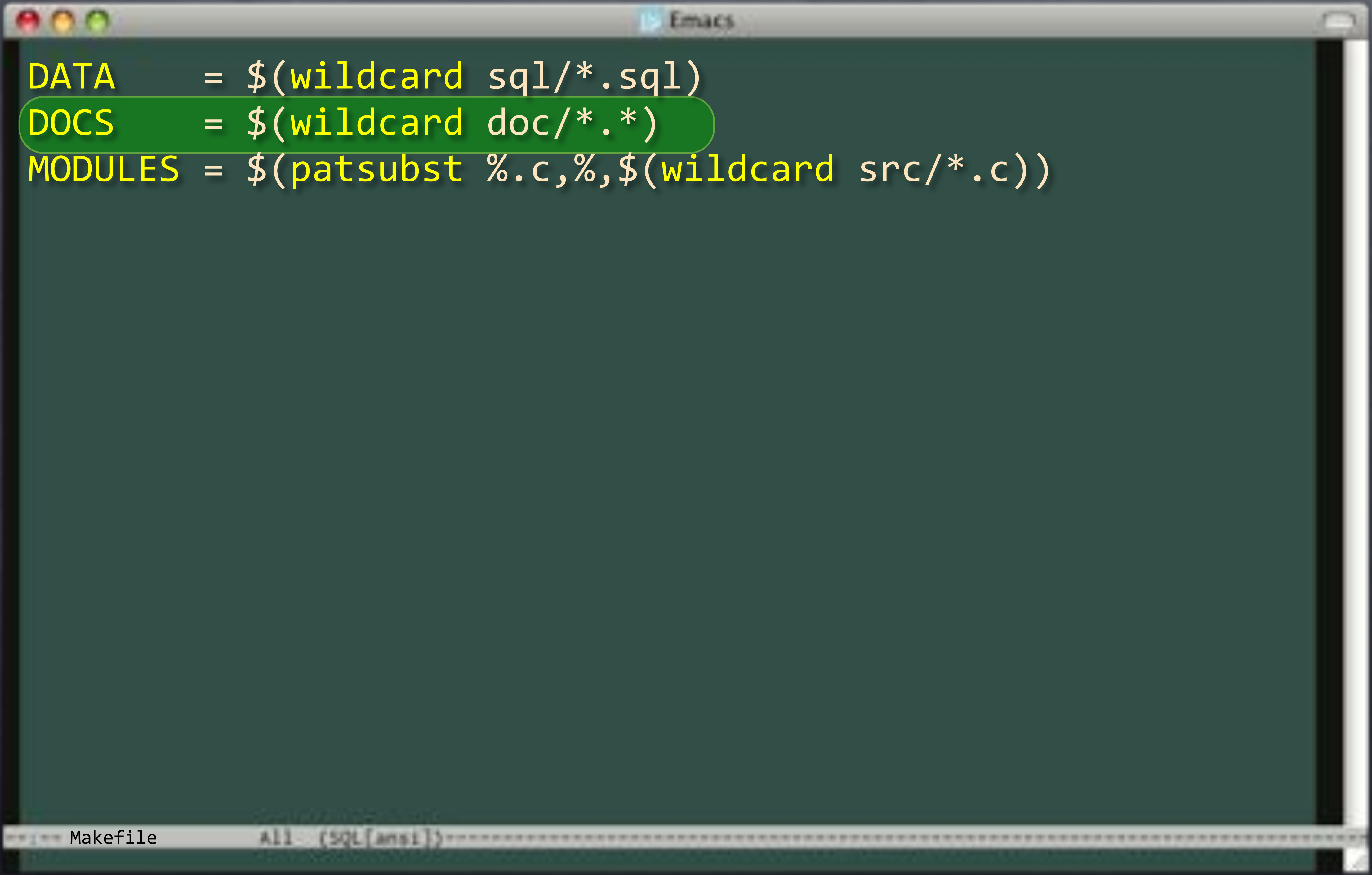
```
DATA    = $(wildcard sql/*.sql)
DOCS    = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))
```

Makefile



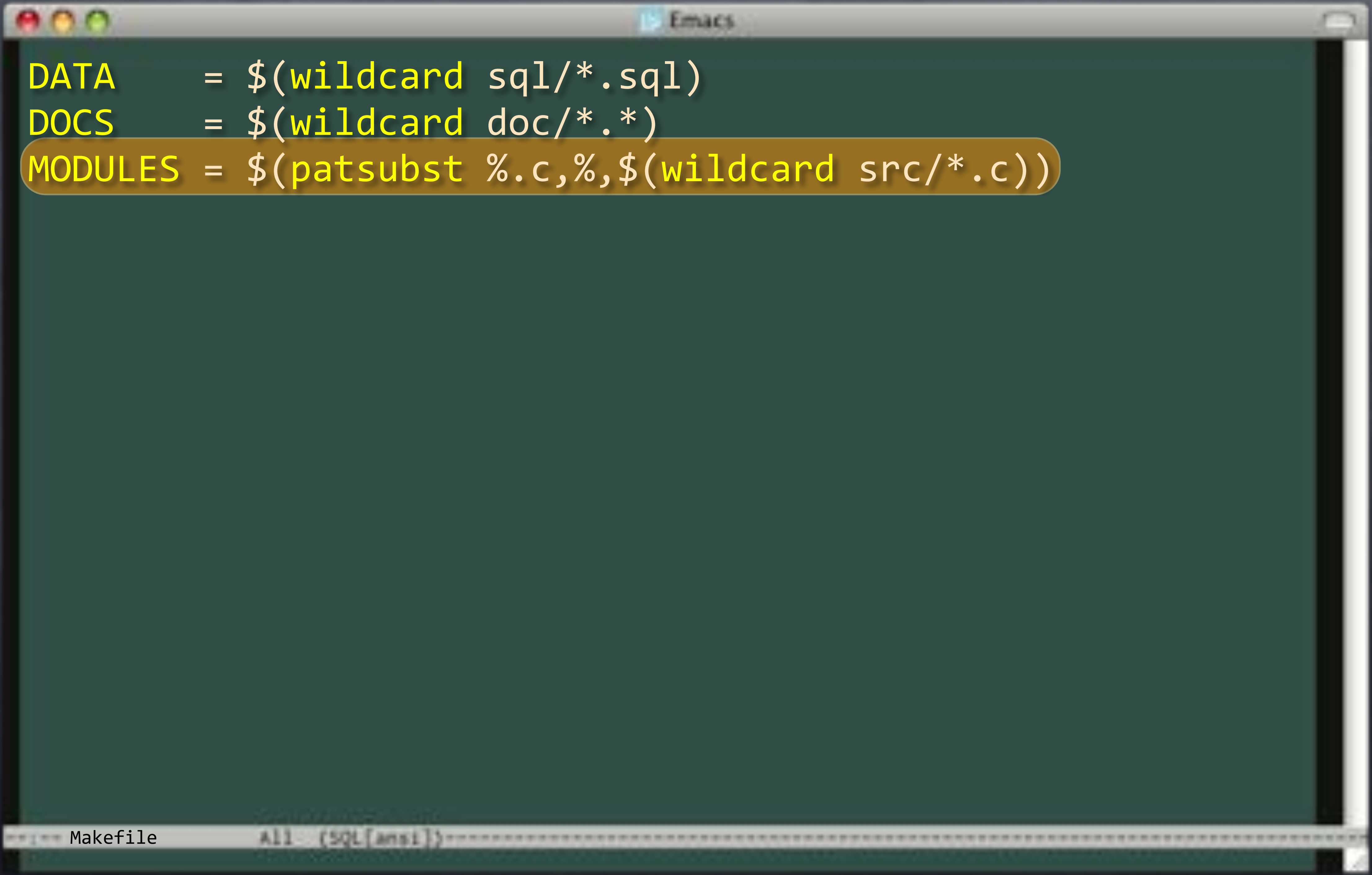
```
DATA    = $(wildcard sql/*.sql)
DOCS    = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))
```

Makefile



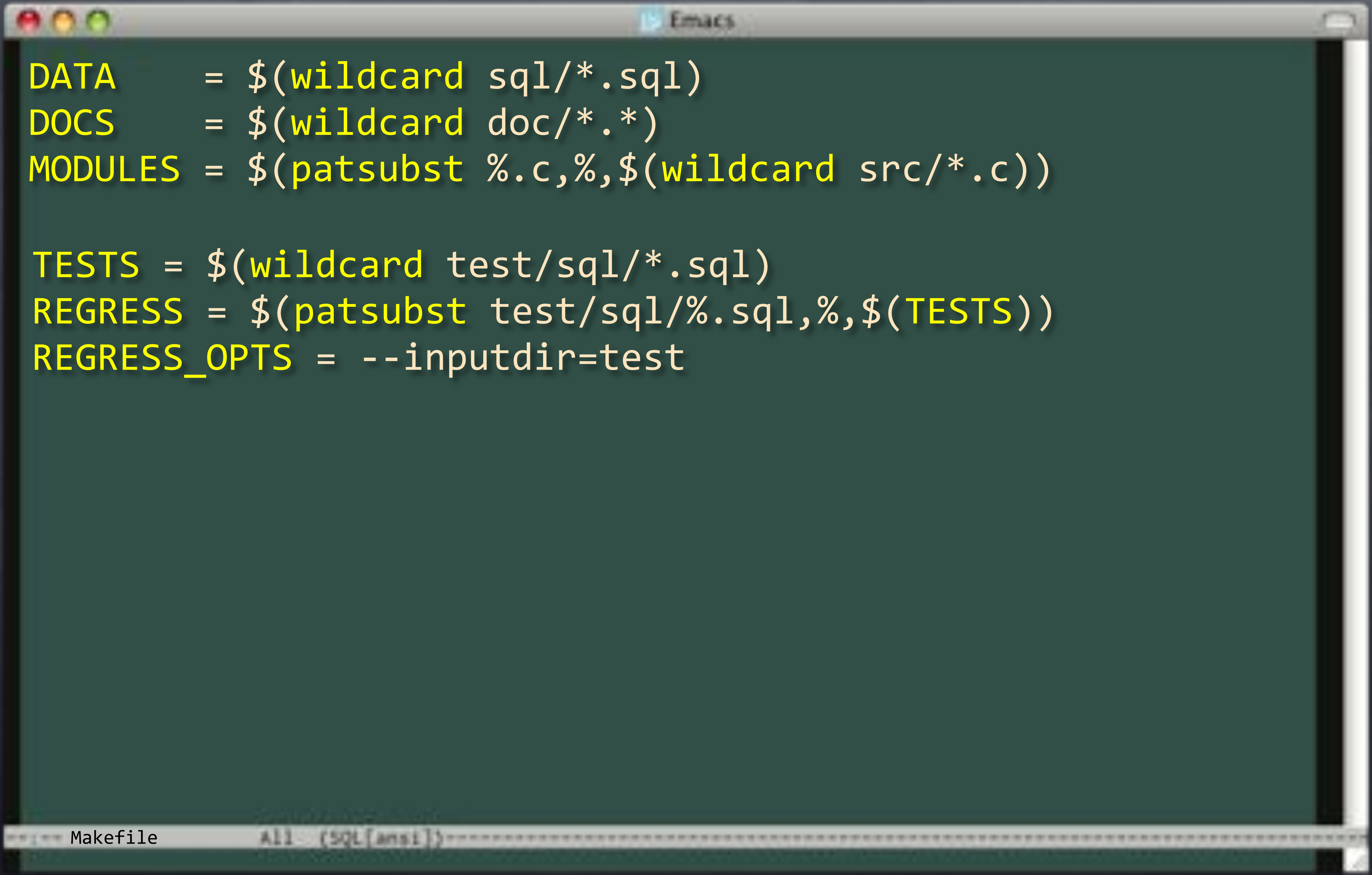
```
DATA     = $(wildcard sql/*.sql)
DOCS     = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))
```

Makefile



```
DATA     = $(wildcard sql/*.sql)
DOCS     = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))
```

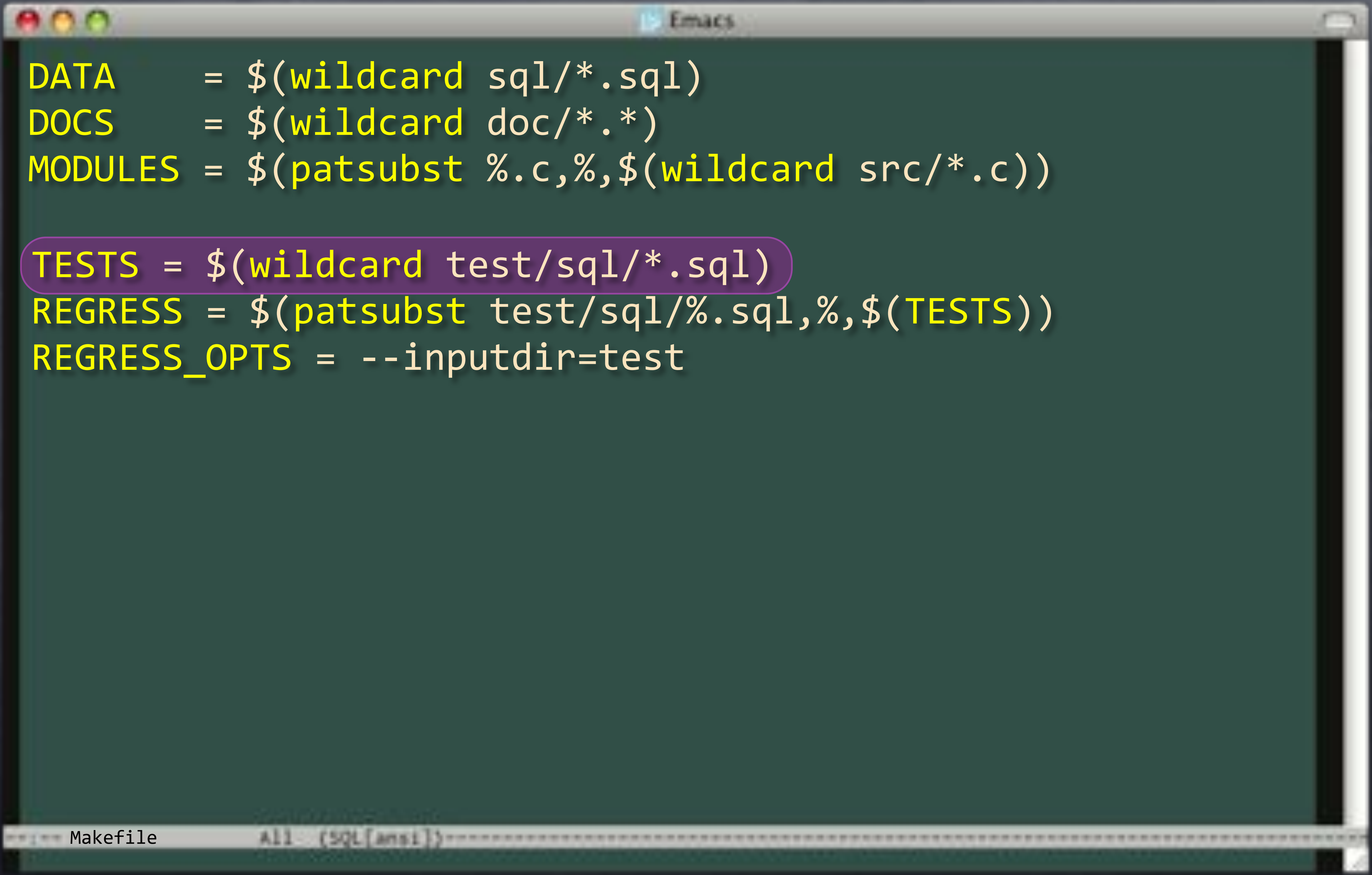
Makefile


Emacs

```
DATA     = $(wildcard sql/*.sql)
DOCS     = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))

TESTS = $(wildcard test/sql/*.sql)
REGRESS = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test
```

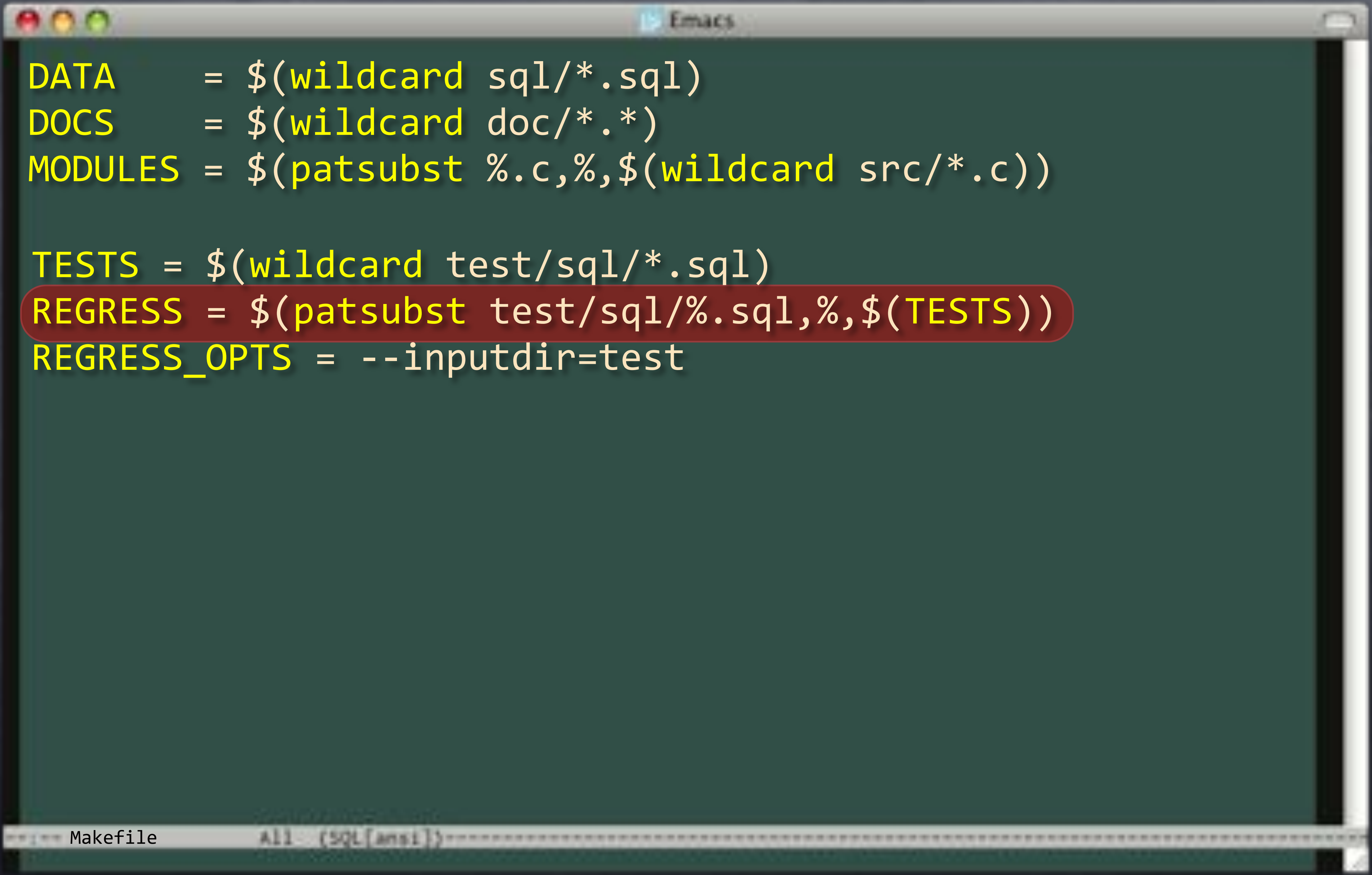
```
--:-- Makefile      All   (SQL[ansi])--------------------------------------------
```

Makefile



```
DATA     = $(wildcard sql/*.sql)
DOCS     = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))

TESTS = $(wildcard test/sql/*.sql)
REGRESS = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test
```

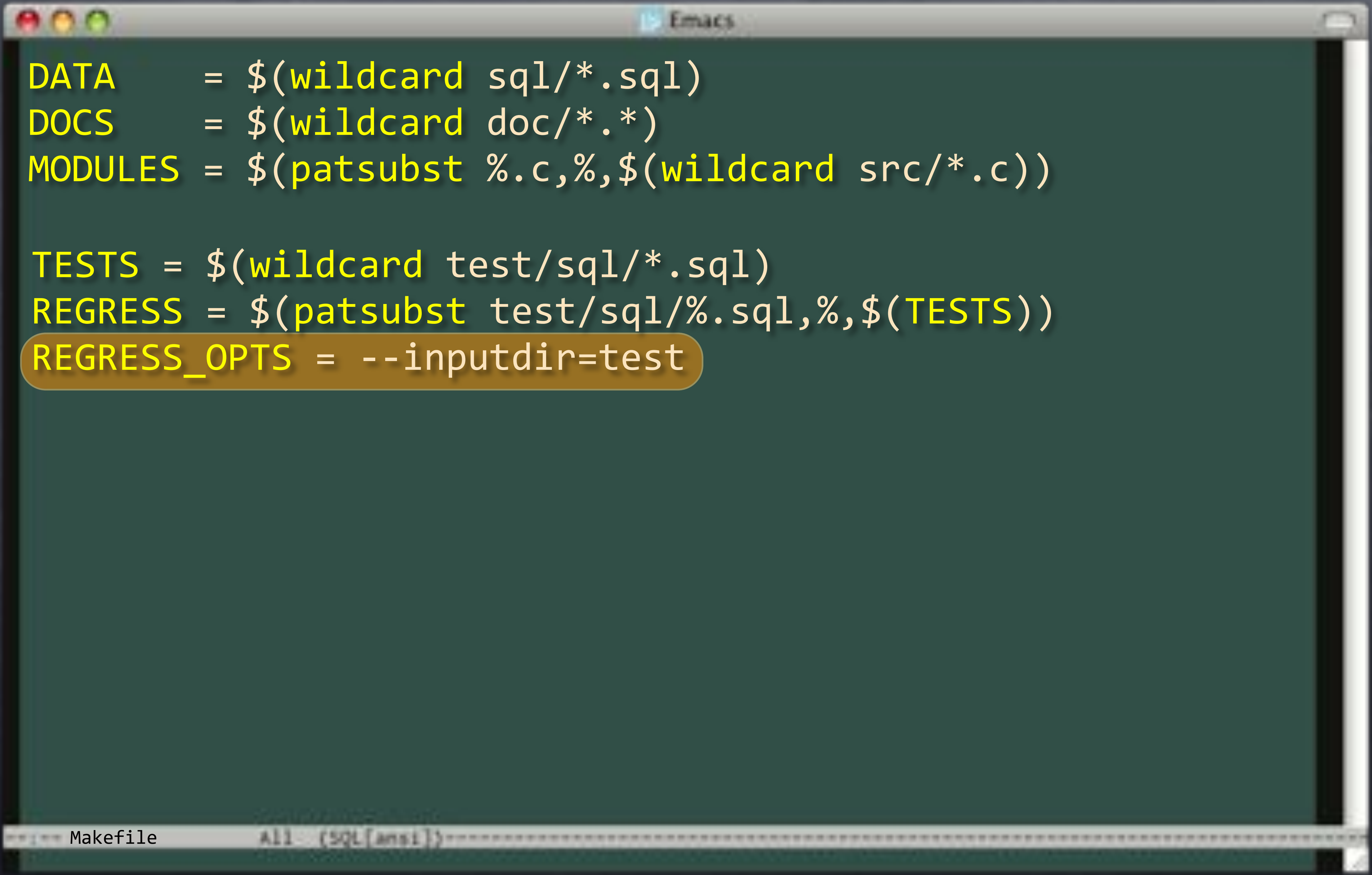
Makefile



Emacs

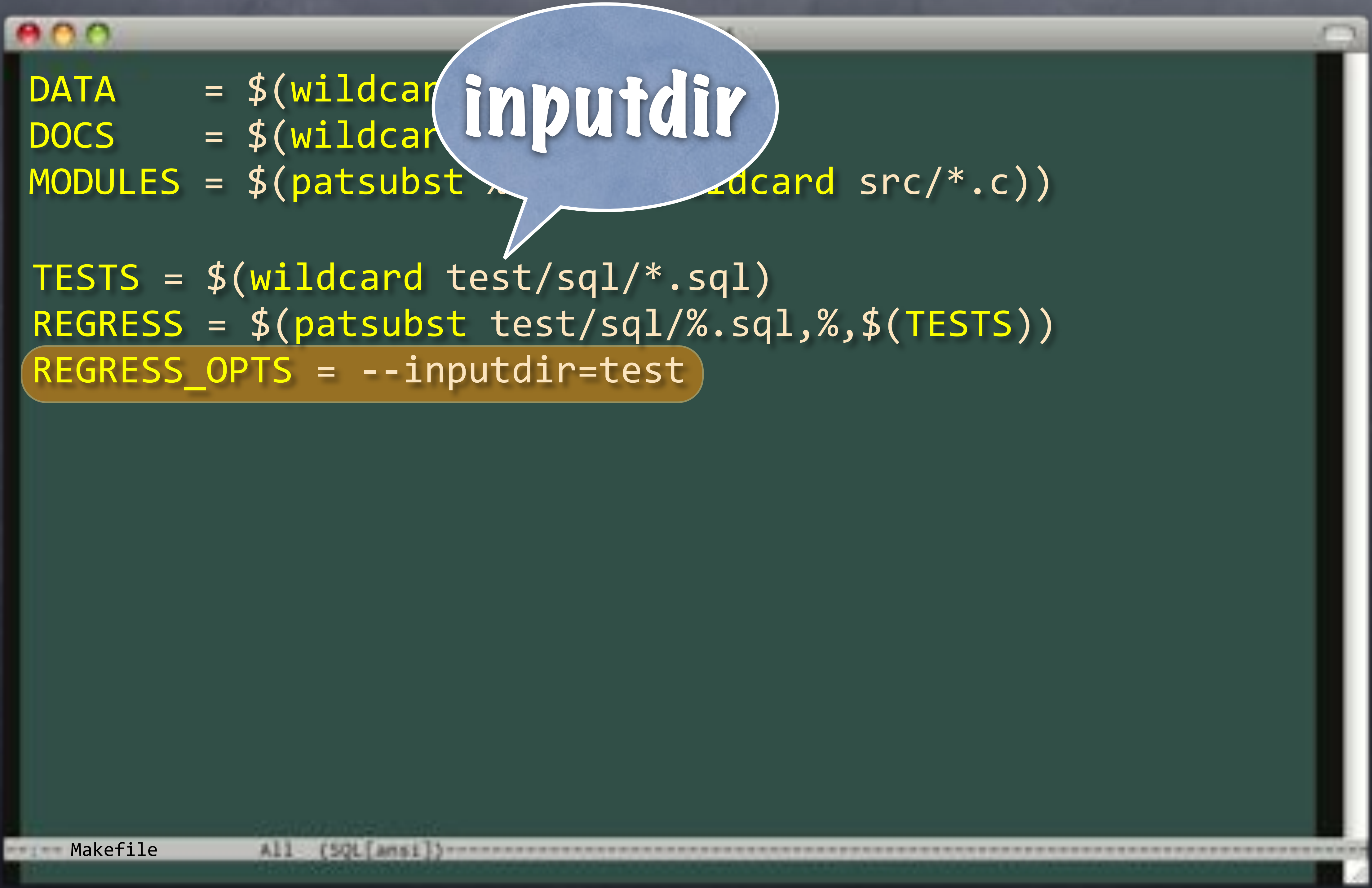
```
DATA     = $(wildcard sql/*.sql)
DOCS     = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))

TESTS = $(wildcard test/sql/*.sql)
REGRESS = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test
```

Makefile



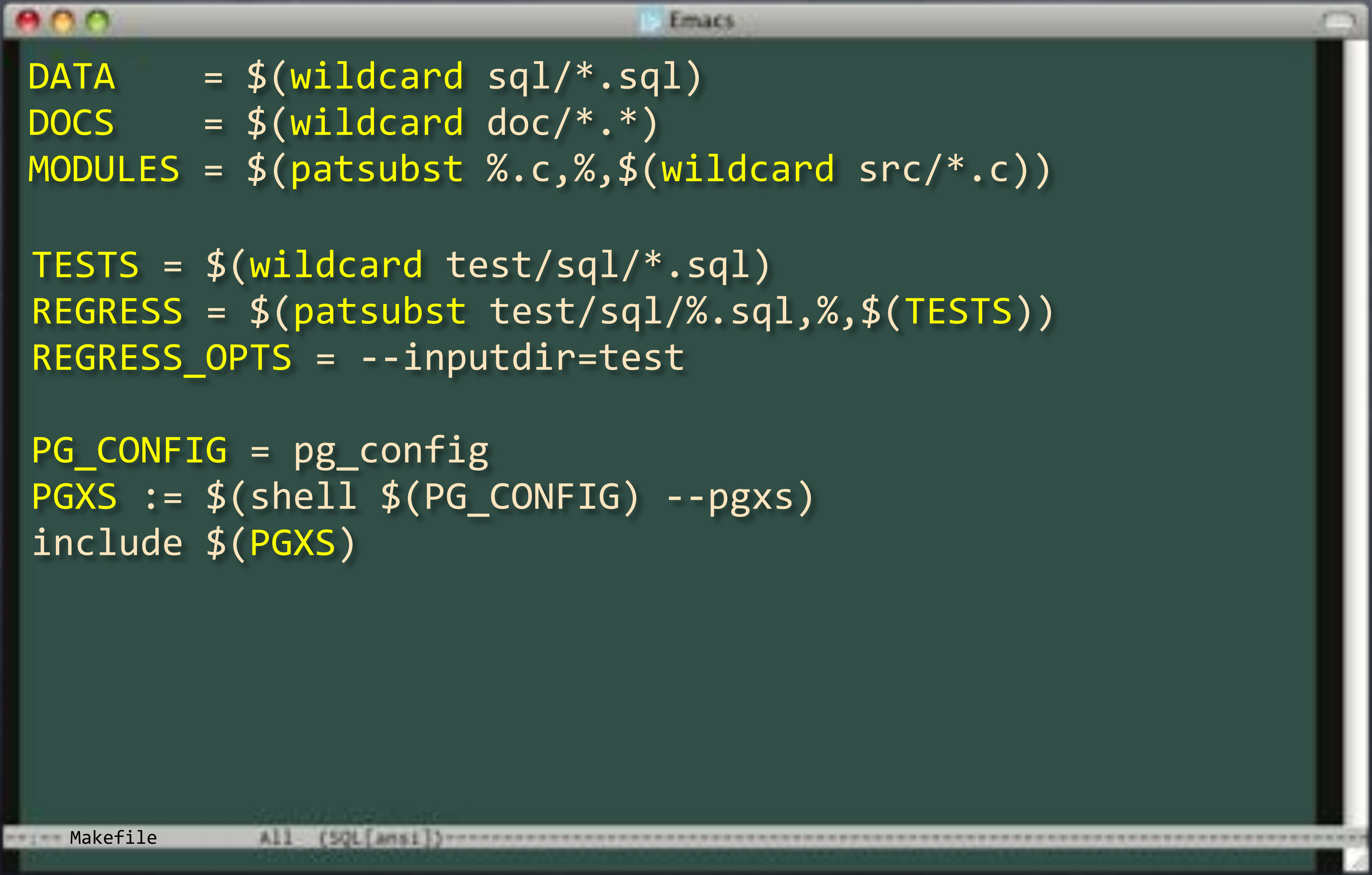
```
DATA     = $(wildcard sql/*.sql)
DOCS     = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))

TESTS = $(wildcard test/sql/*.sql)
REGRESS = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test
```

Makefile



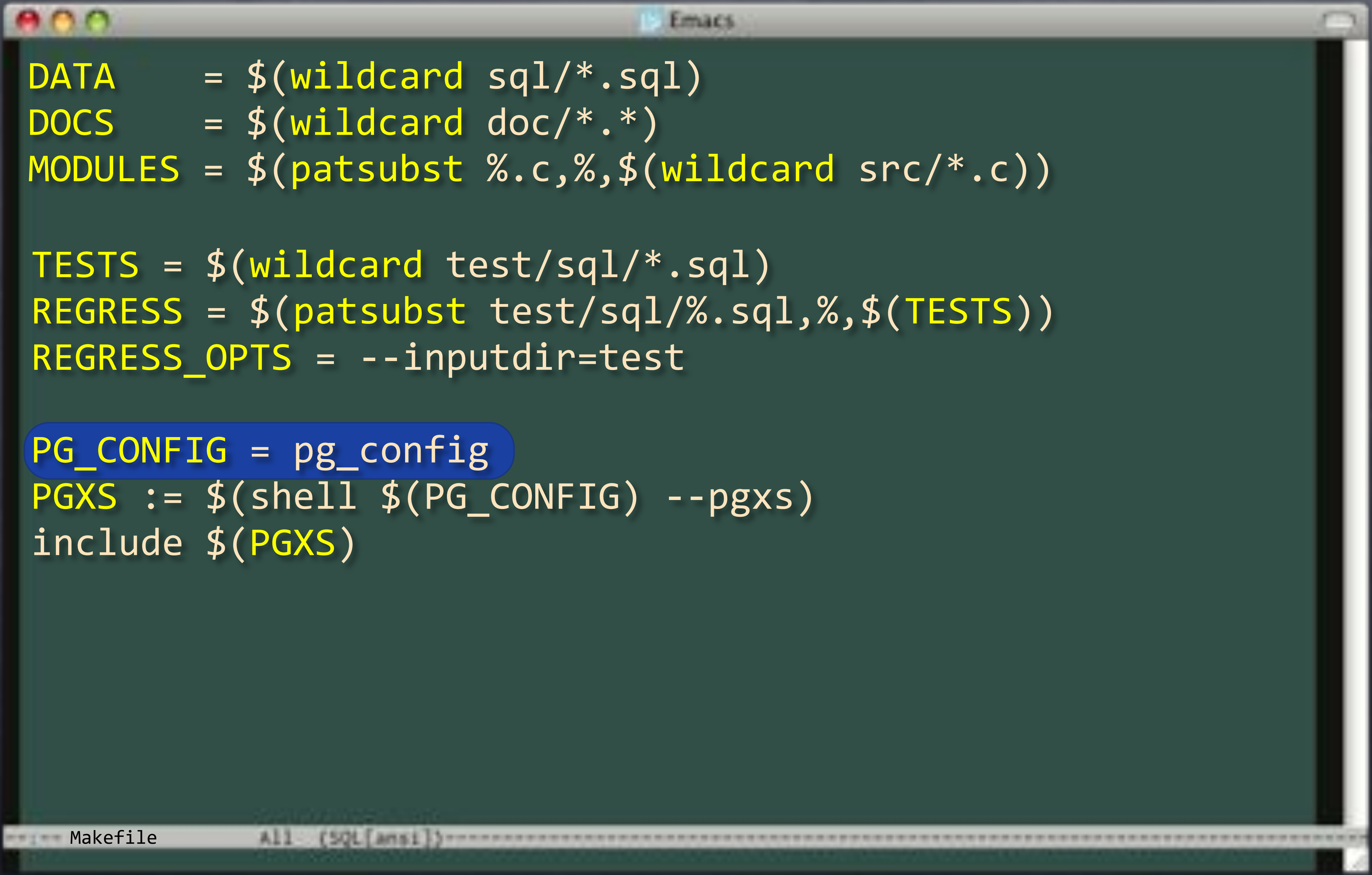
```
DATA     = $(wildcar
DOCS     = $(wildcar
MODULES = $(patsubst %                idcard src/*.c))

TESTS = $(wildcard test/sql/*.sql)
REGRESS = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test
```

Makefile



```
DATA     = $(wildcard sql/*.sql)
DOCS     = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))

TESTS = $(wildcard test/sql/*.sql)
REGRESS = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test

PG_CONFIG = pg_config
PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

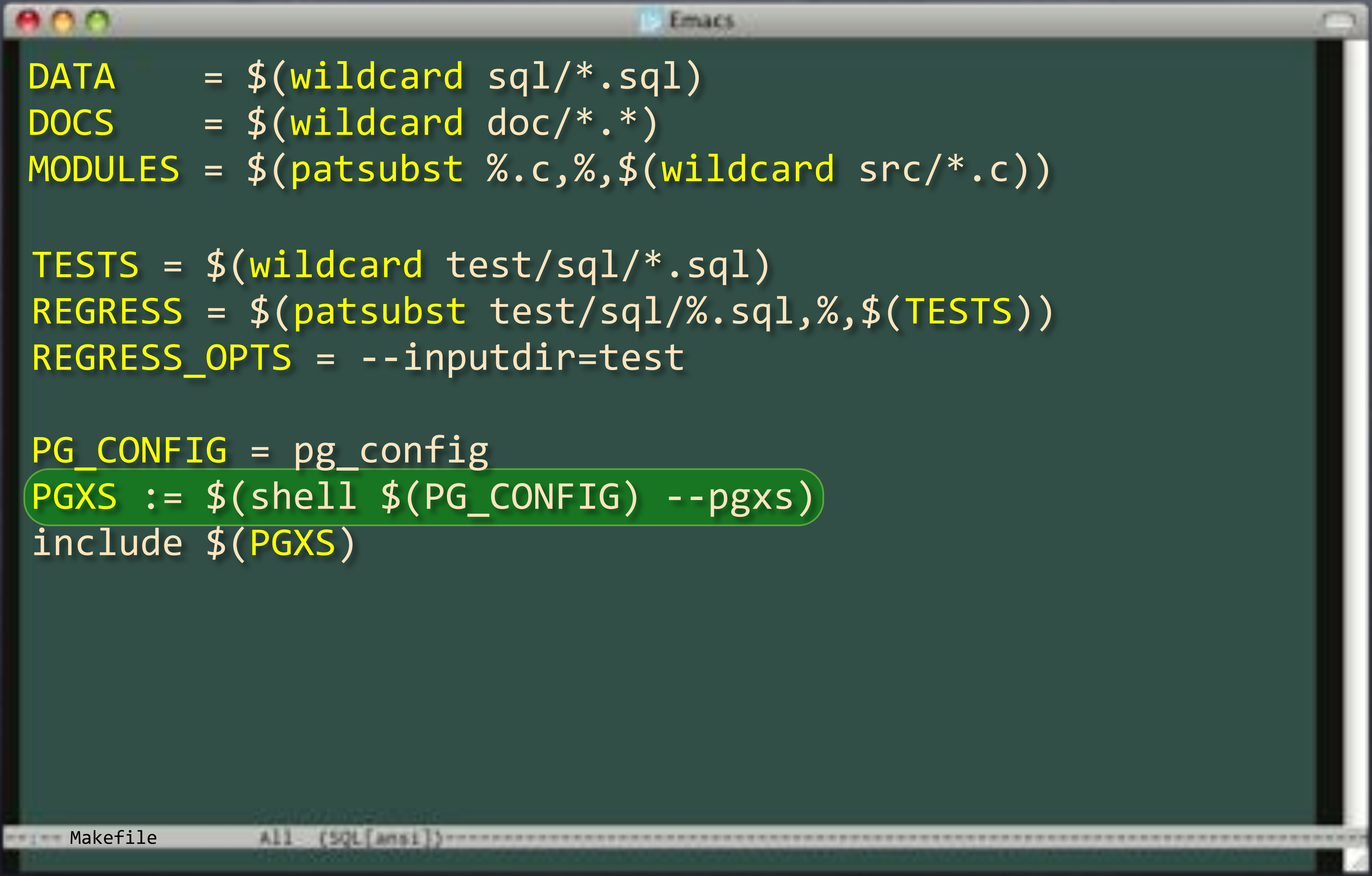
Makefile


Emacs

```
DATA    = $(wildcard sql/*.sql)
DOCS    = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))

TESTS = $(wildcard test/sql/*.sql)
REGRESS = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test

PG_CONFIG = pg_config
PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

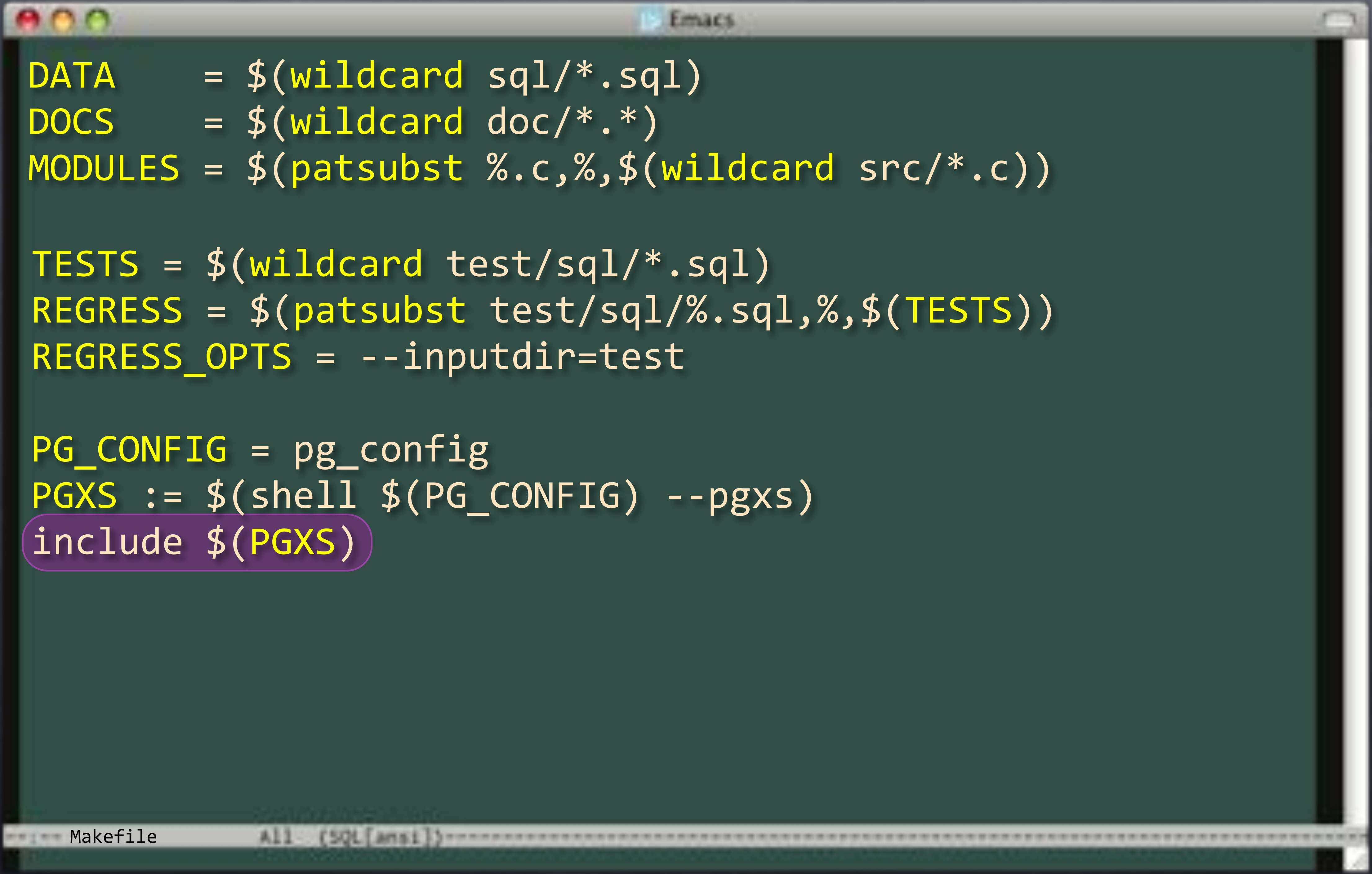
```
--:-- Makefile        All  (SQL[ansi])--------------------------------------------
```

Makefile



```
DATA    = $(wildcard sql/*.sql)
DOCS    = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))

TESTS = $(wildcard test/sql/*.sql)
REGRESS = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test

PG_CONFIG = pg_config
PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

Makefile



```
DATA    = $(wildcard sql/*.sql)
DOCS    = $(wildcard doc/*.*)
MODULES = $(patsubst %.c,%,$(wildcard src/*.c))

TESTS = $(wildcard test/sql/*.sql)
REGRESS = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test

PG_CONFIG = pg_config
PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

CREATE EXTENSION

CREATE EXTENSION

- New in 9.1

CREATE EXTENSION

- New in 9.1
- Add extension to a DB with

CREATE EXTENSION

- New in 9.1
- Add extension to a DB with
  - CREATE EXTENSION pair;

CREATE EXTENSION

- New in 9.1
- Add extension to a DB with
  - CREATE EXTENSION pair;
- No need to run SQL script in psql

Packaging Extensions Needs

Packaging Extensions Needs

- Control file

Packaging Extensions Needs

- Control file
- Migration from unpackaged

Packaging Extensions Needs

- Control file
- Migration from unpackaged
  - pair--unpackaged--0.1.0.sql

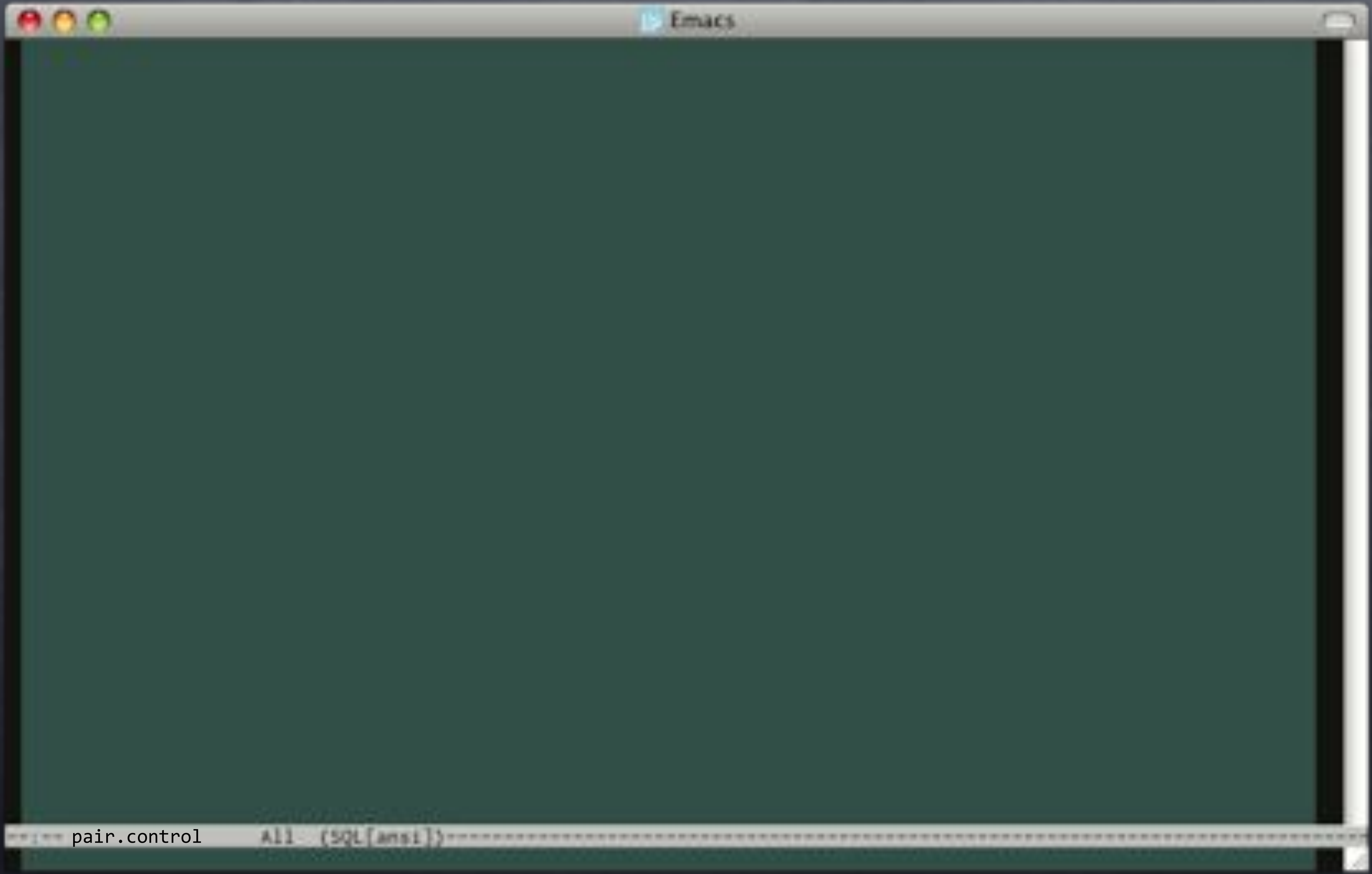
Packaging Extensions Needs

- Control file
- Migration from unpackaged
  - pair--unpackaged--0.1.0.sql
- Properly-named SQL script

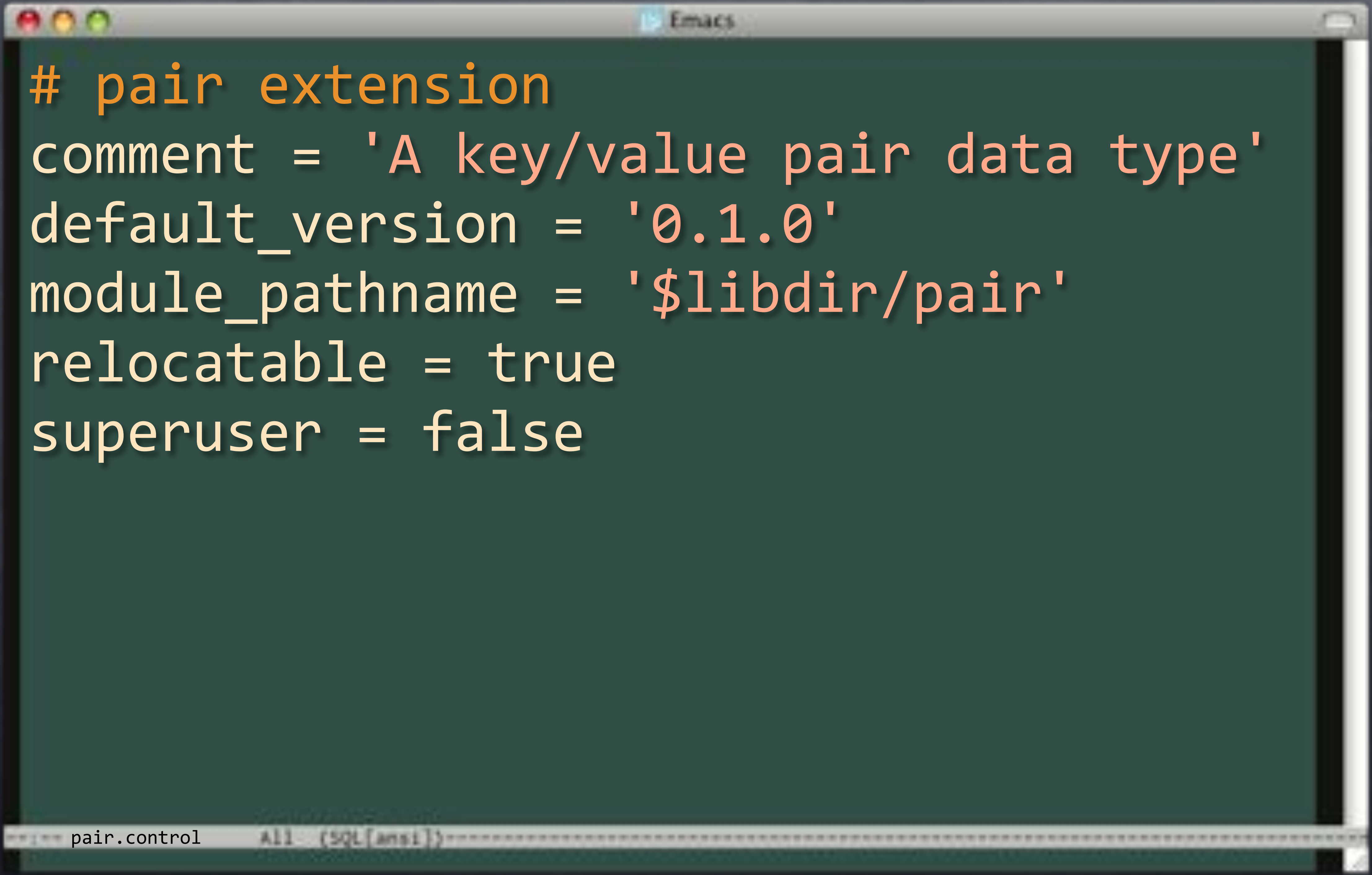
Packaging Extensions Needs

- Control file
- Migration from unpackaged
  - pair--unpackaged--0.1.0.sql
- Properly-named SQL script
  - pair--0.1.0.sql

Create the control file

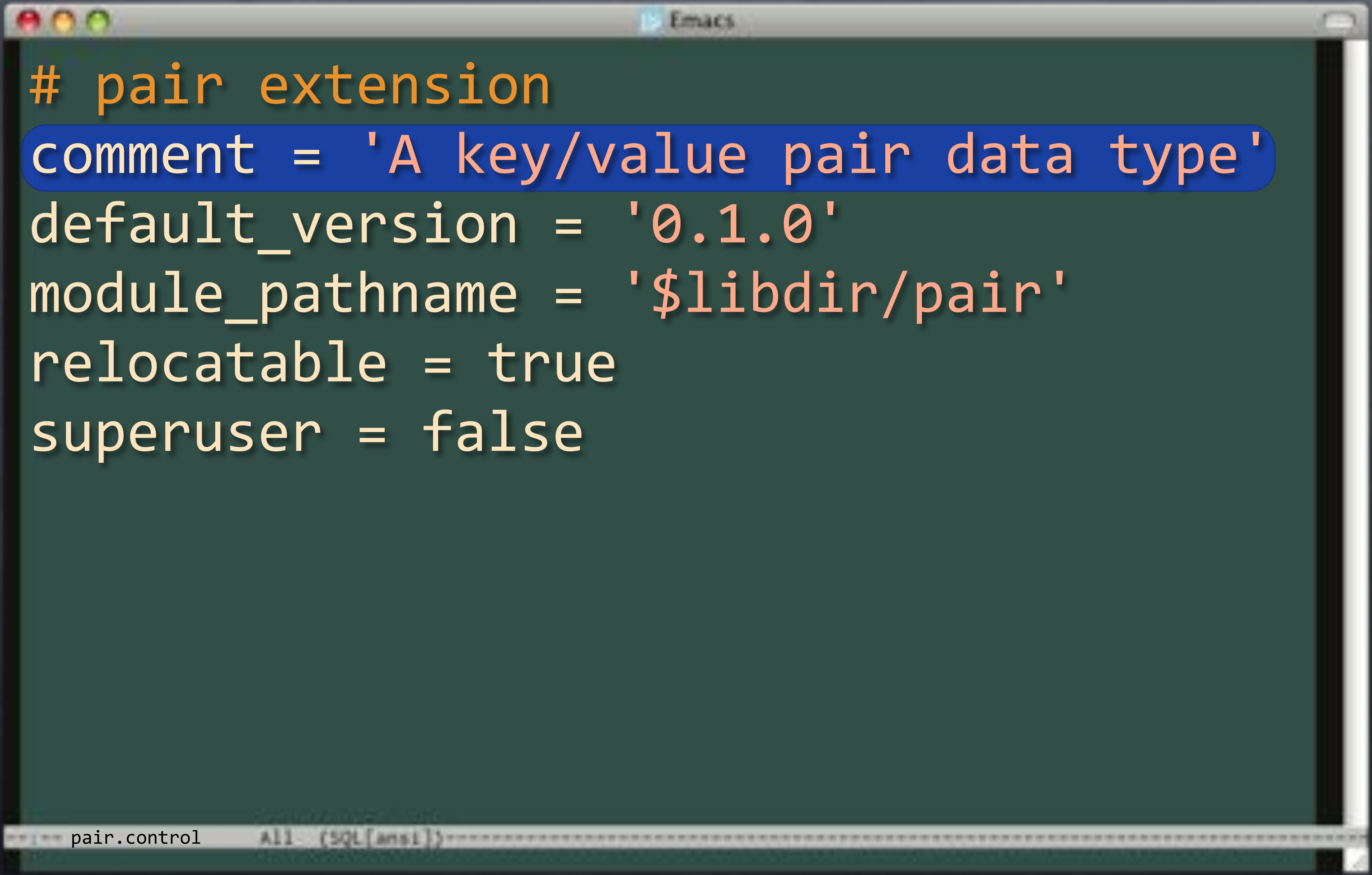


Create the control file



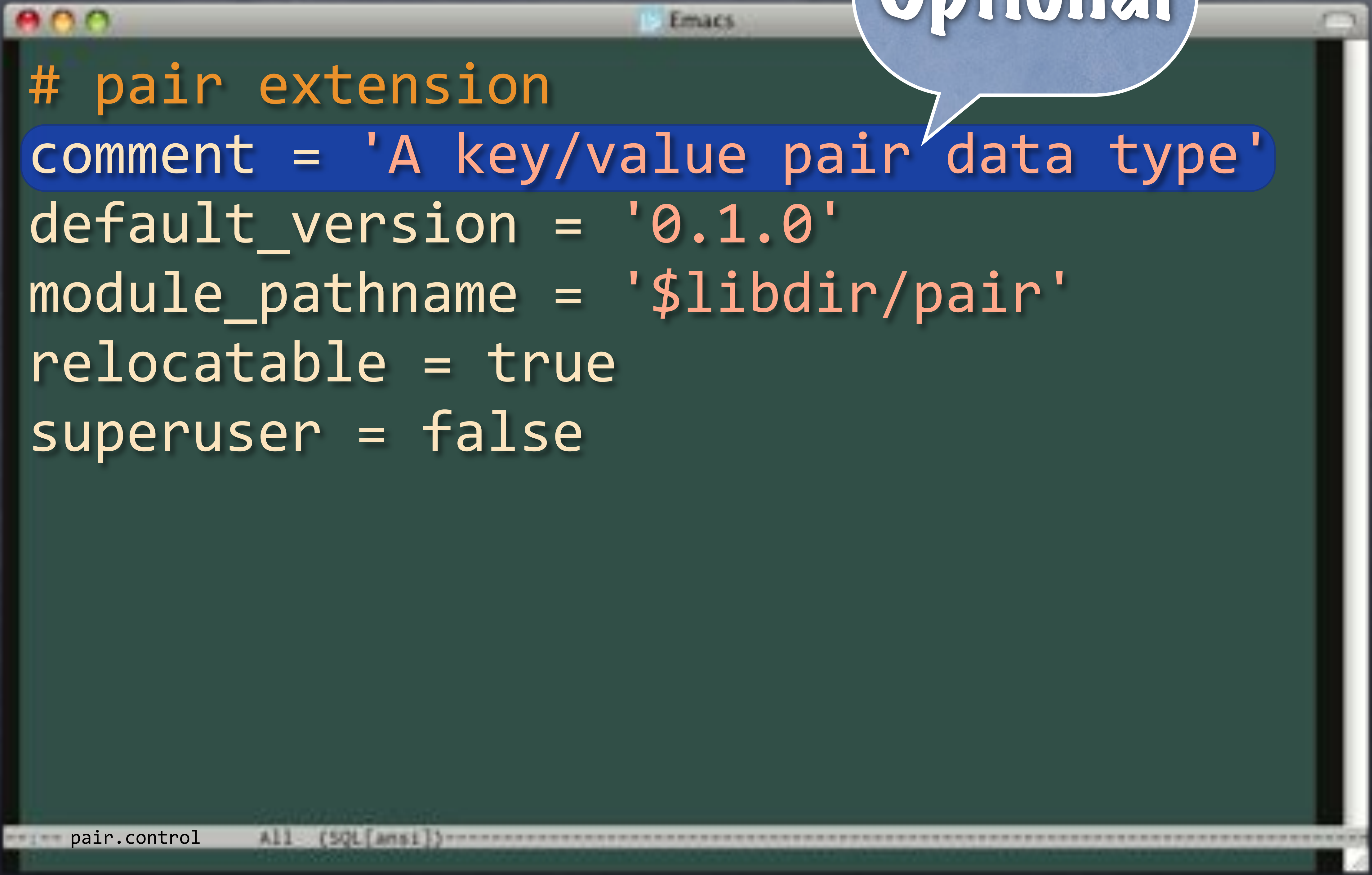
```
# pair extension
comment = 'A key/value pair data type'
default_version = '0.1.0'
module_pathname = '$libdir/pair'
relocatable = true
superuser = false
```

Create the control file



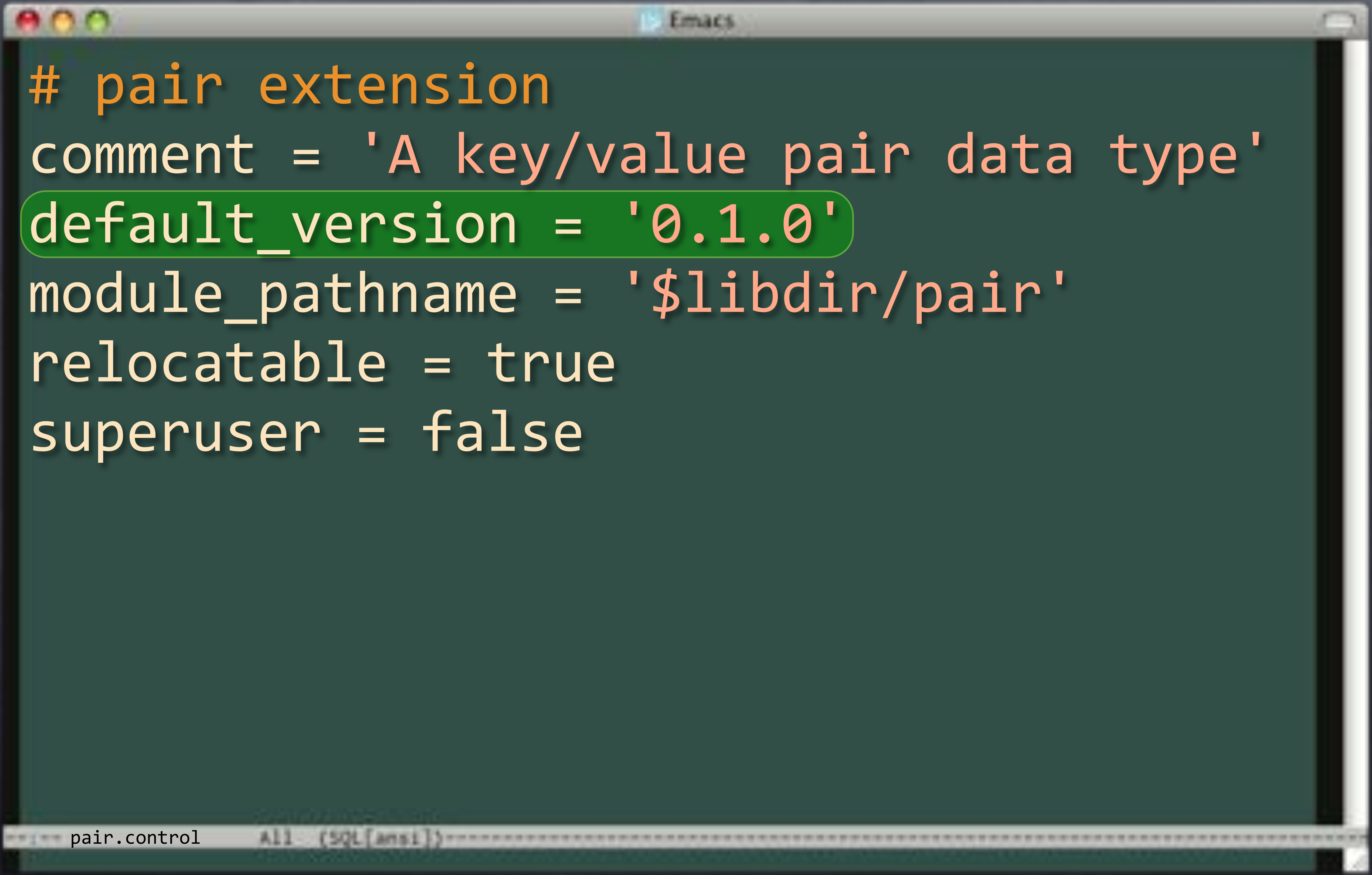
```
# pair extension
comment = 'A key/value pair data type'
default_version = '0.1.0'
module_pathname = '$libdir/pair'
relocatable = true
superuser = false
```

Create the control file



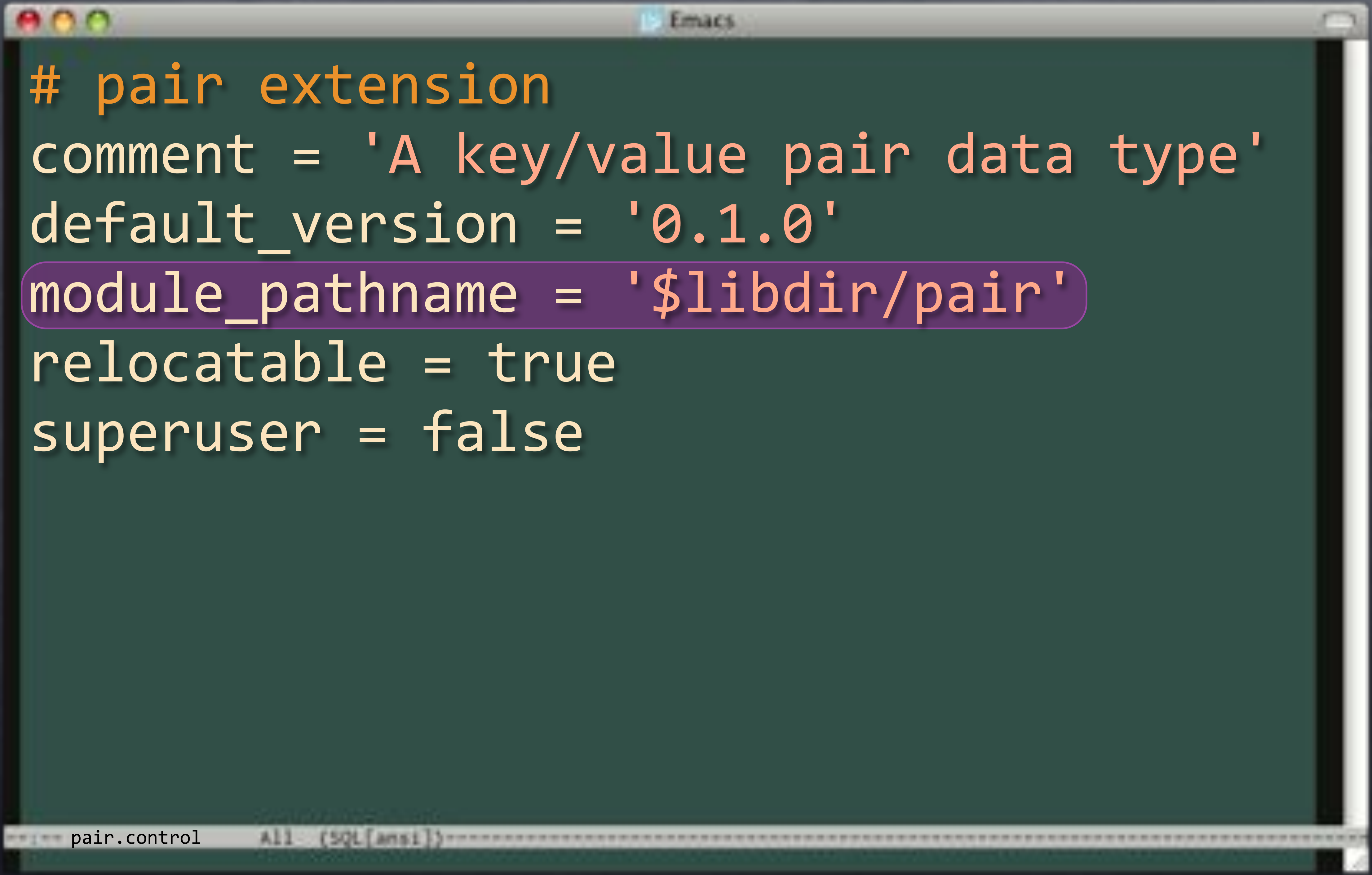
```
# pair extension
comment = 'A key/value pair data type'
default_version = '0.1.0'
module_pathname = '$libdir/pair'
relocatable = true
superuser = false
```

Create the control file



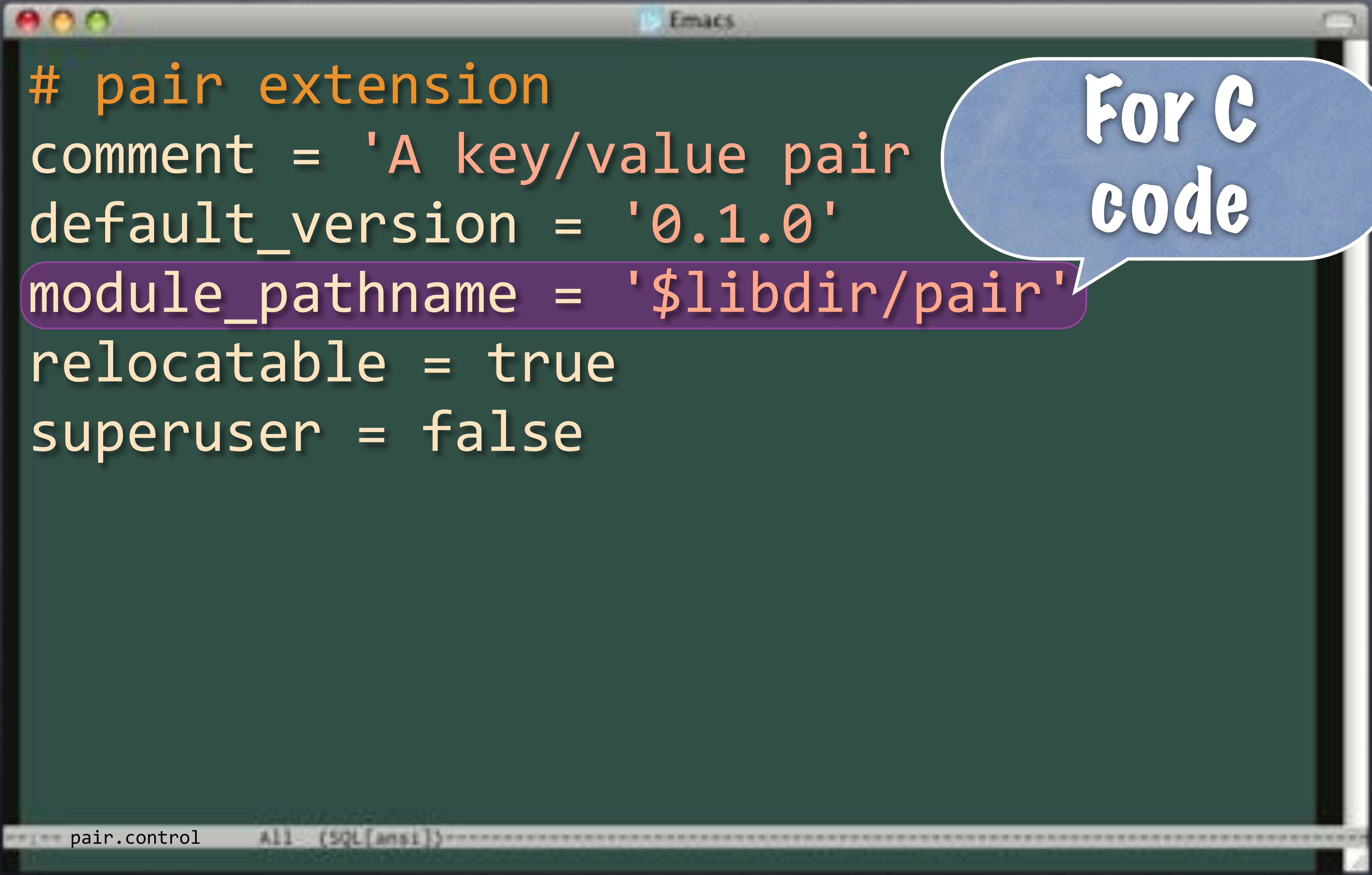
```
# pair extension
comment = 'A key/value pair data type'
default_version = '0.1.0'
module_pathname = '$libdir/pair'
relocatable = true
superuser = false
```

Create the control file



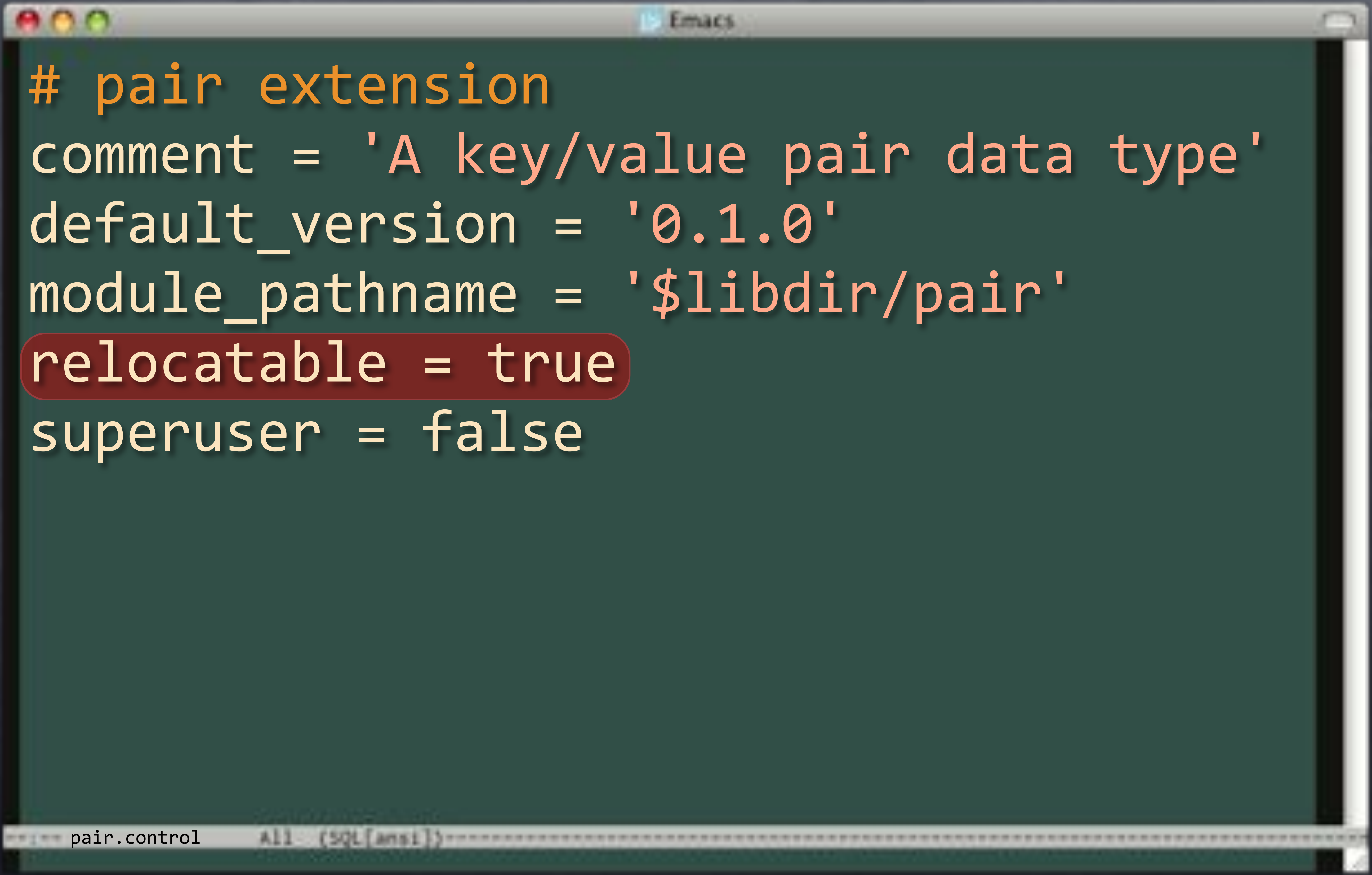
```
# pair extension
comment = 'A key/value pair data type'
default_version = '0.1.0'
module_pathname = '$libdir/pair'
relocatable = true
superuser = false
```

Create the control file



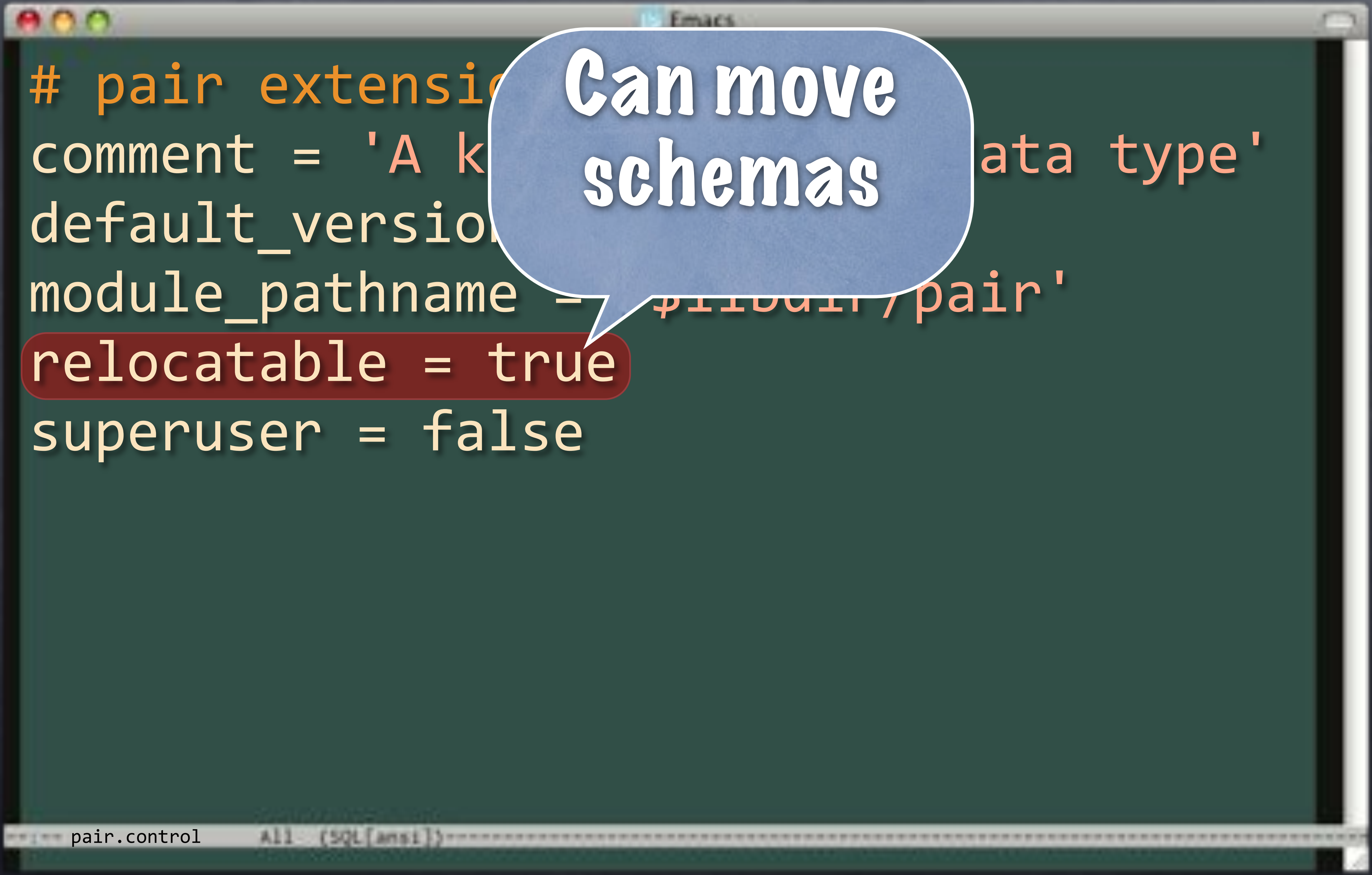
```
# pair extension
comment = 'A key/value pair
default_version = '0.1.0'
module_pathname = '$libdir/pair'
relocatable = true
superuser = false
```

Create the control file



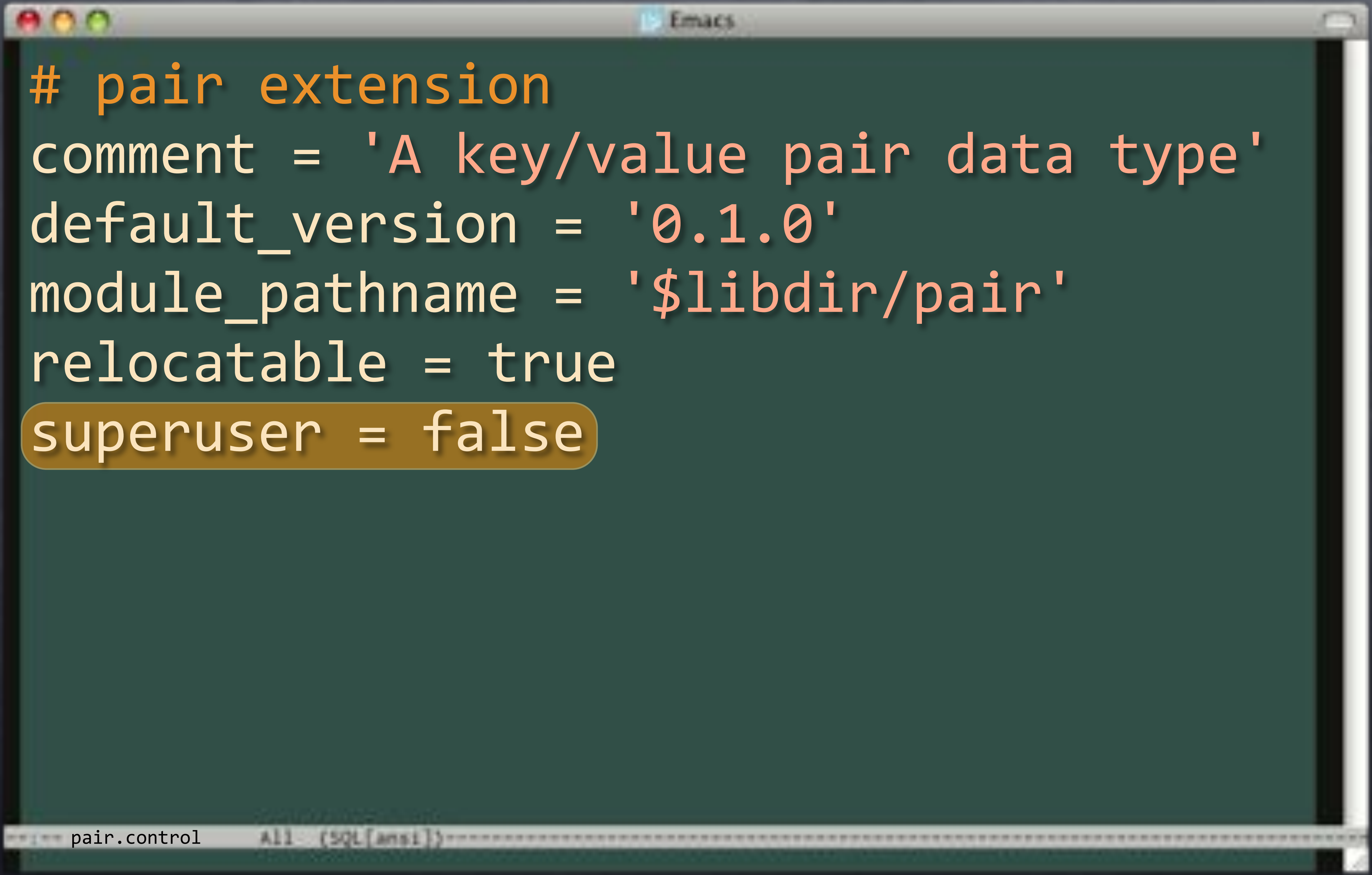
```
# pair extension
comment = 'A key/value pair data type'
default_version = '0.1.0'
module_pathname = '$libdir/pair'
relocatable = true
superuser = false
```

Create the control file



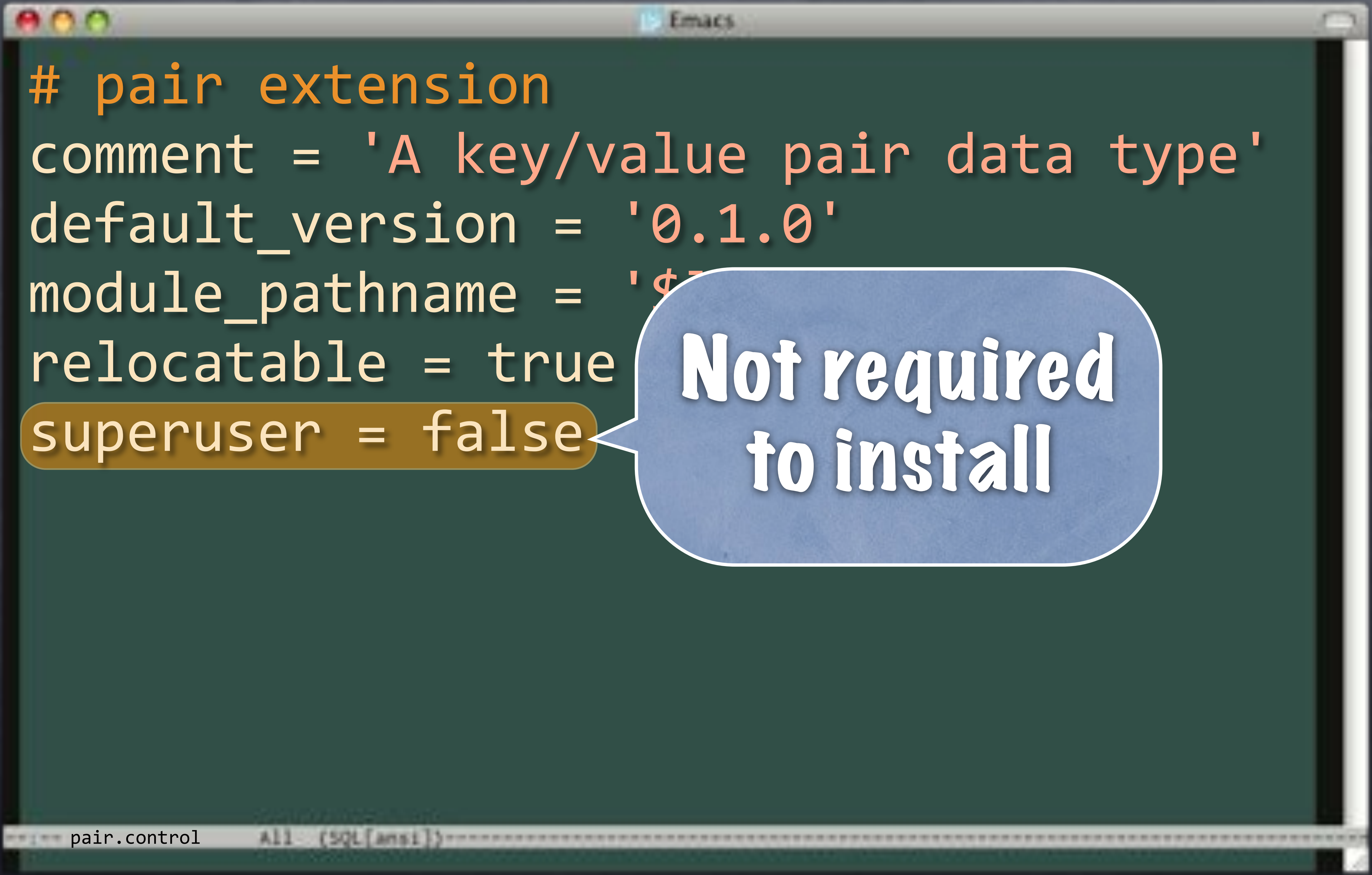
```
# pair extensi
comment = 'A k                ata type'
default_versio
module_pathname =             /pair'
relocatable = true
superuser = false
```

Create the control file



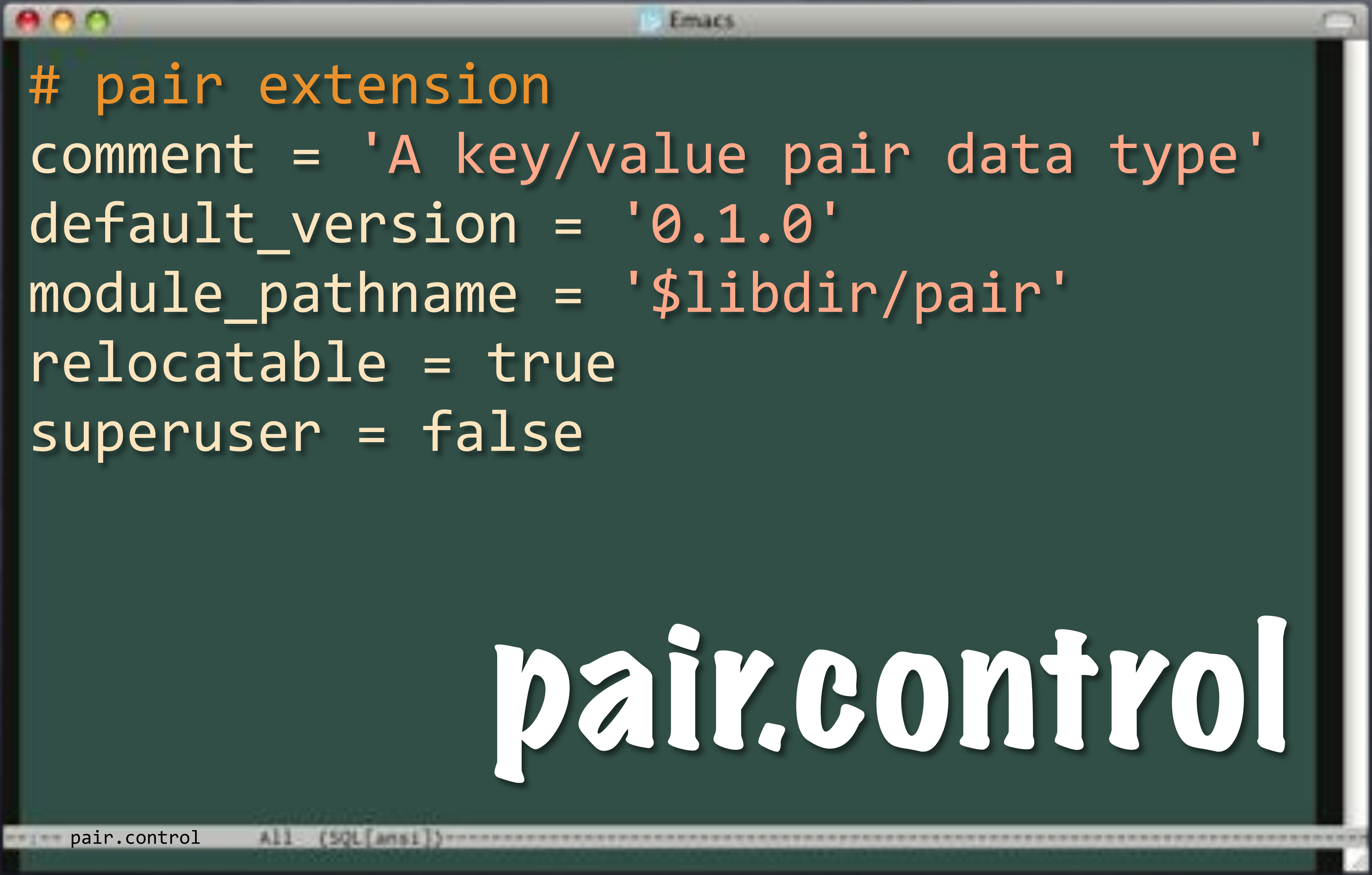
```
# pair extension
comment = 'A key/value pair data type'
default_version = '0.1.0'
module_pathname = '$libdir/pair'
relocatable = true
superuser = false
```

Create the control file



```
# pair extension
comment = 'A key/value pair data type'
default_version = '0.1.0'
module_pathname = '$
relocatable = true
superuser = false
```

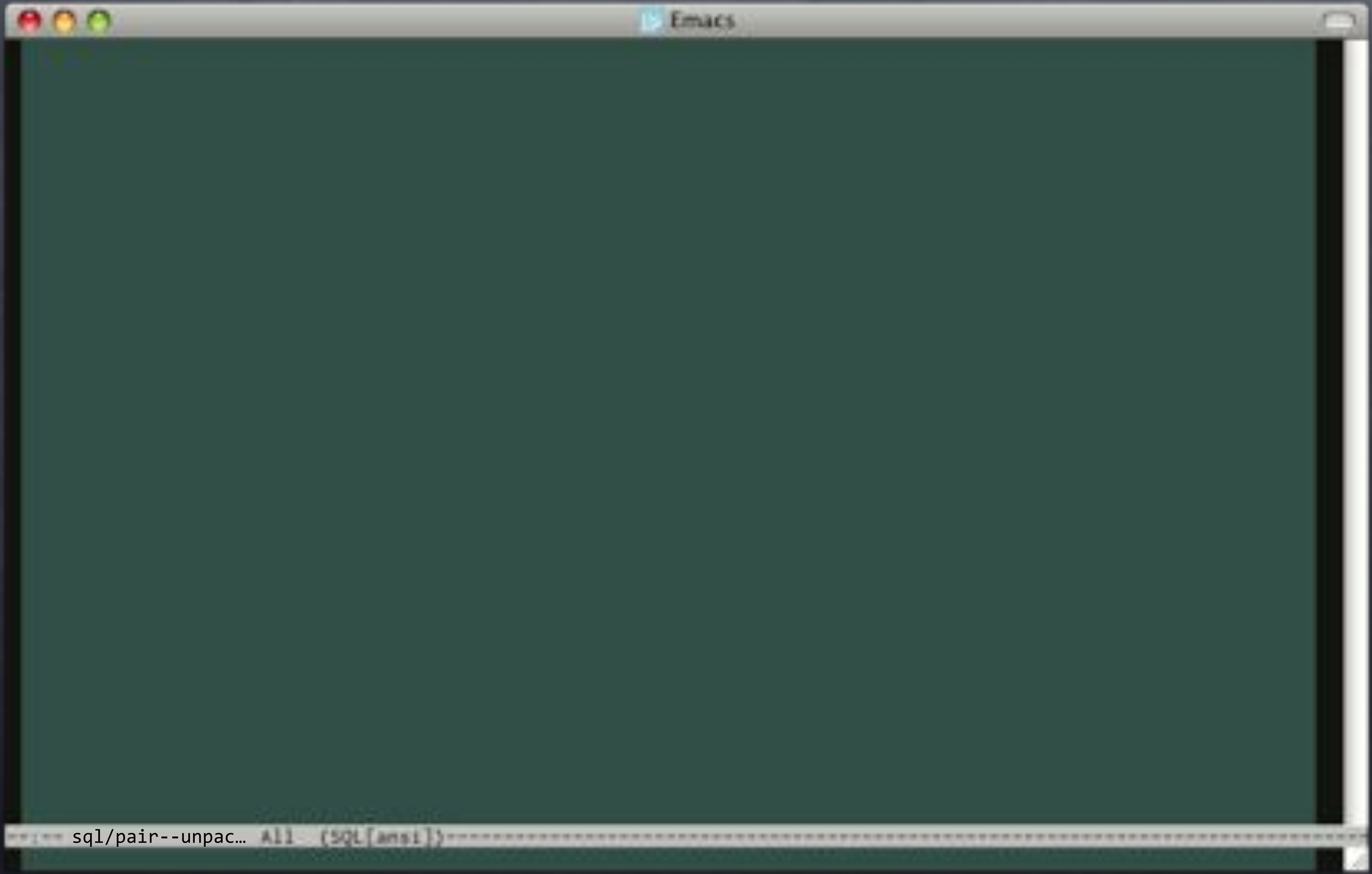
Create the control file



```
# pair extension
comment = 'A key/value pair data type'
default_version = '0.1.0'
module_pathname = '$libdir/pair'
relocatable = true
superuser = false
```

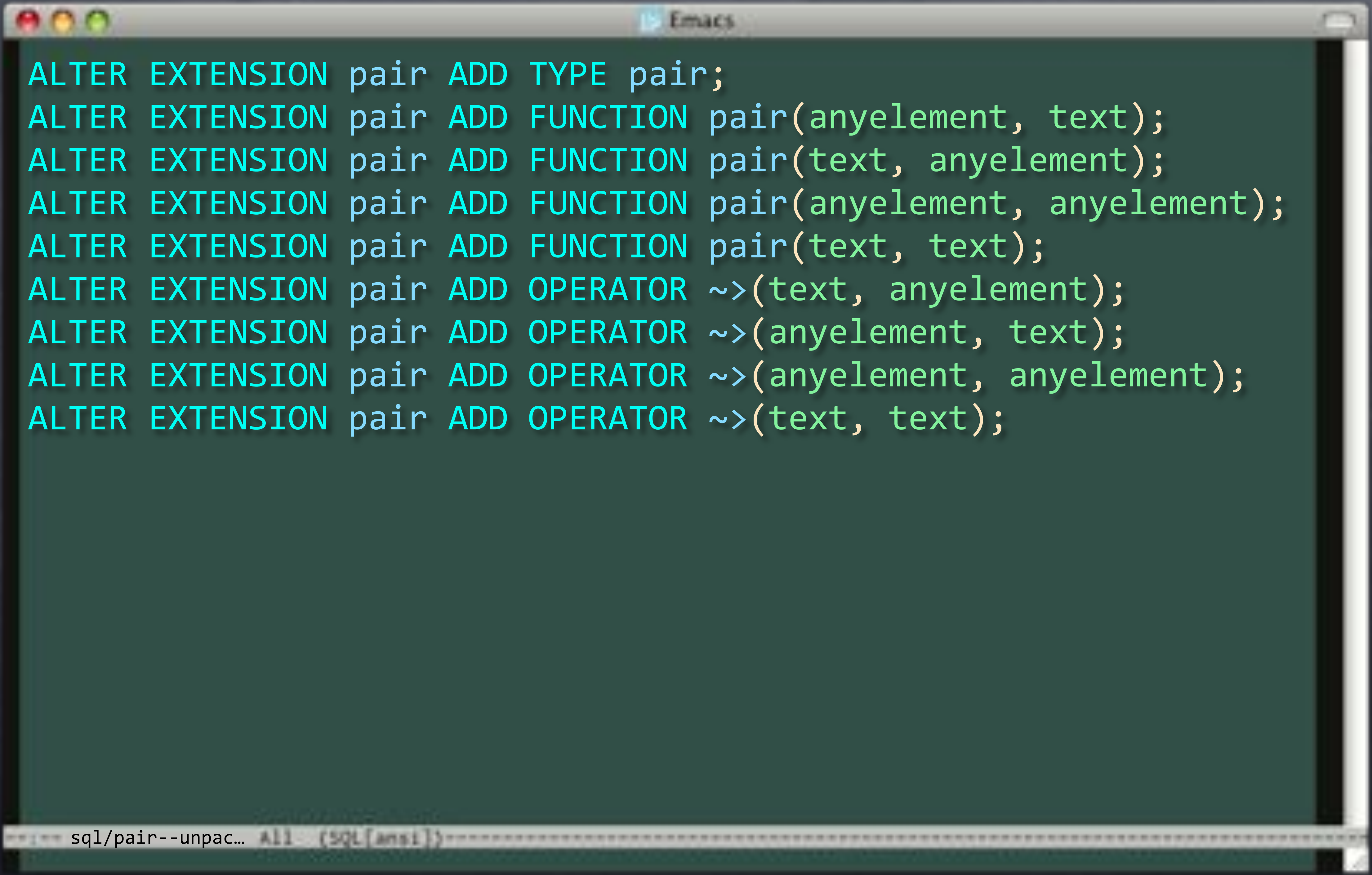
pair.control

Migration from Unpackaged



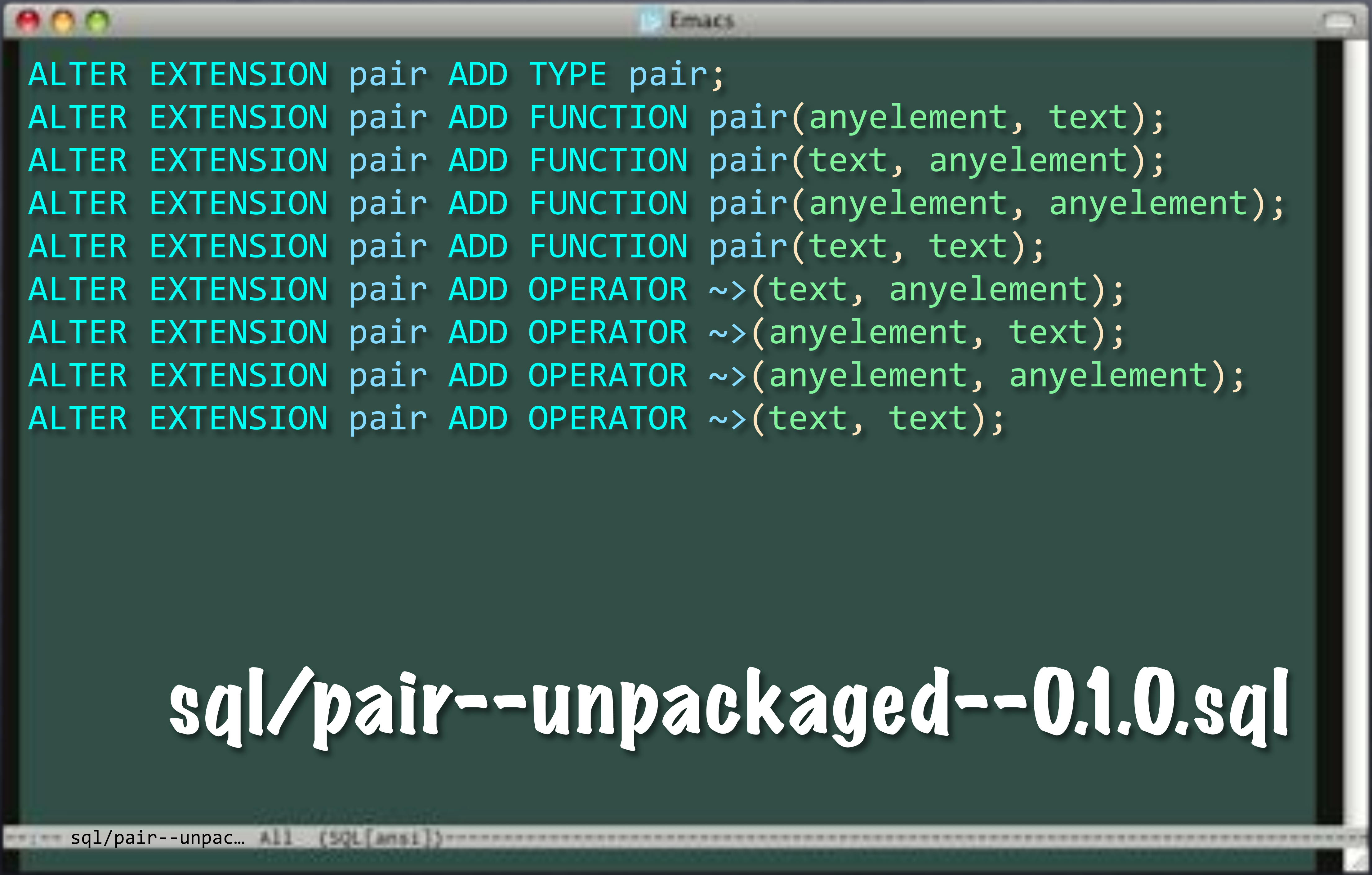
```
--:-- sql/pair--unpac…  All  (SQL[ansi])-------------------------------------------
```

Migration from Unpackaged



```sql
ALTER EXTENSION pair ADD TYPE pair;
ALTER EXTENSION pair ADD FUNCTION pair(anyelement, text);
ALTER EXTENSION pair ADD FUNCTION pair(text, anyelement);
ALTER EXTENSION pair ADD FUNCTION pair(anyelement, anyelement);
ALTER EXTENSION pair ADD FUNCTION pair(text, text);
ALTER EXTENSION pair ADD OPERATOR ~>(text, anyelement);
ALTER EXTENSION pair ADD OPERATOR ~>(anyelement, text);
ALTER EXTENSION pair ADD OPERATOR ~>(anyelement, anyelement);
ALTER EXTENSION pair ADD OPERATOR ~>(text, text);
```

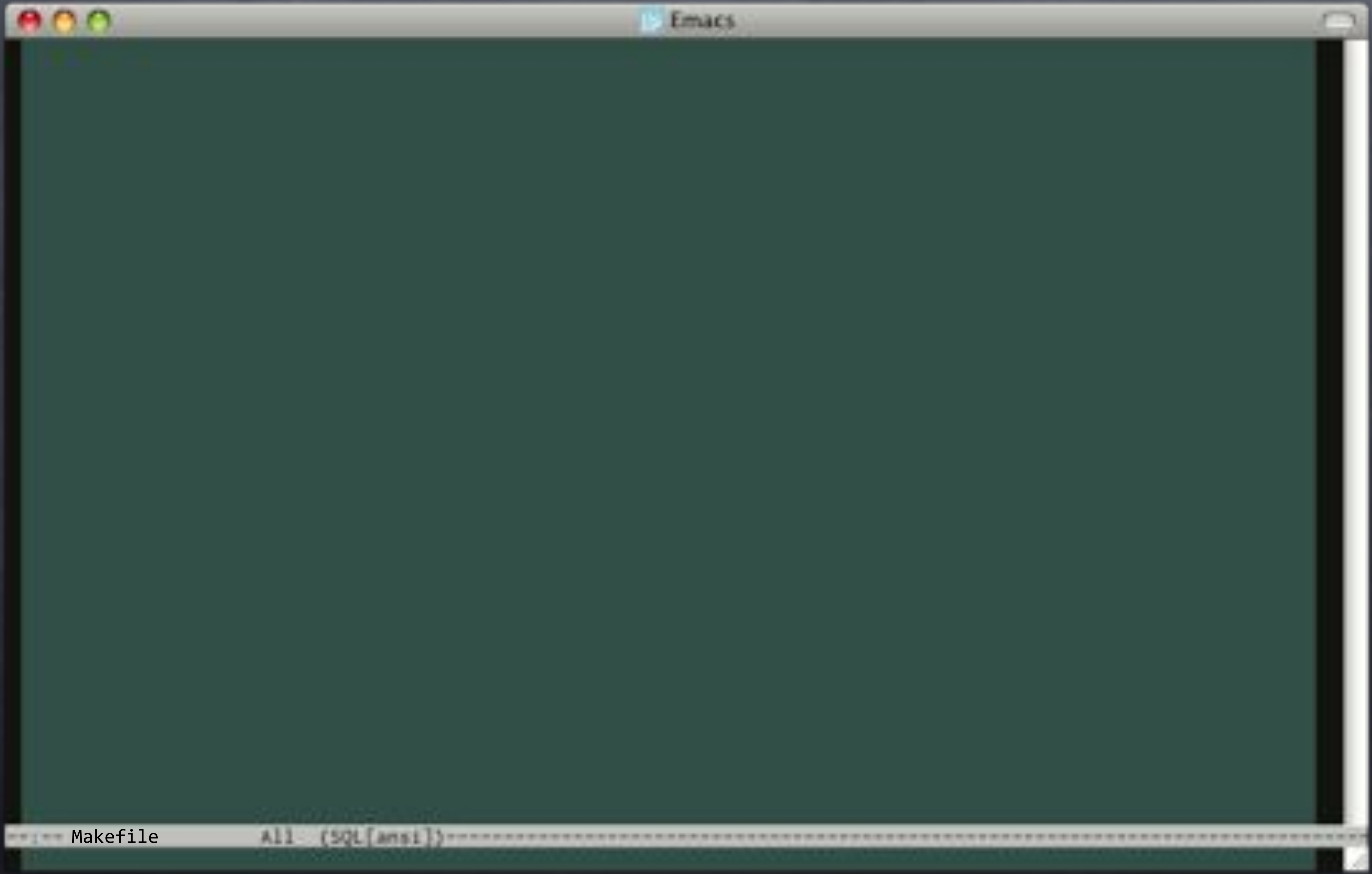
Migration from Unpackaged



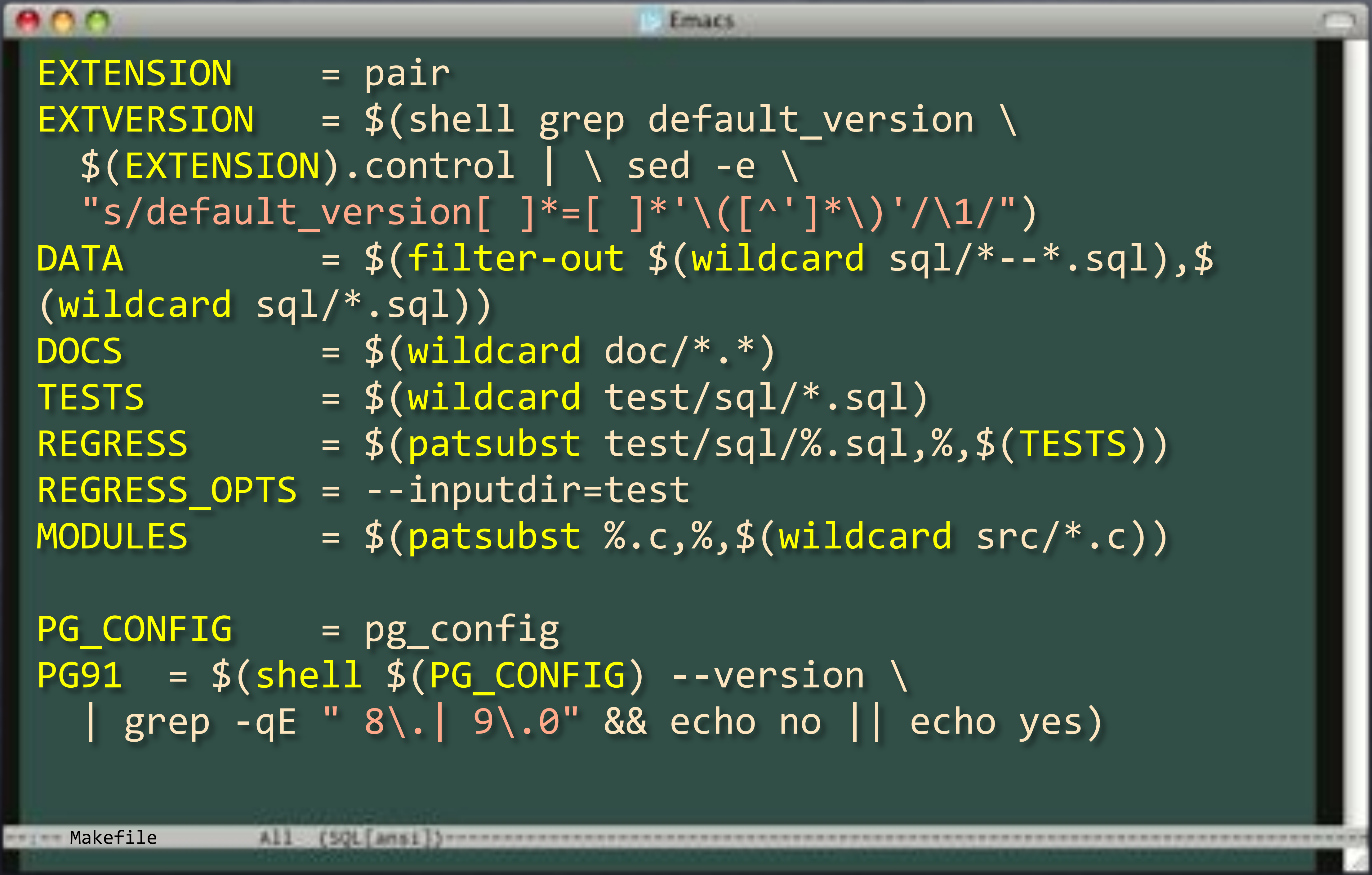
```
ALTER EXTENSION pair ADD TYPE pair;
ALTER EXTENSION pair ADD FUNCTION pair(anyelement, text);
ALTER EXTENSION pair ADD FUNCTION pair(text, anyelement);
ALTER EXTENSION pair ADD FUNCTION pair(anyelement, anyelement);
ALTER EXTENSION pair ADD FUNCTION pair(text, text);
ALTER EXTENSION pair ADD OPERATOR ~>(text, anyelement);
ALTER EXTENSION pair ADD OPERATOR ~>(anyelement, text);
ALTER EXTENSION pair ADD OPERATOR ~>(anyelement, anyelement);
ALTER EXTENSION pair ADD OPERATOR ~>(text, text);
```

sql/pair--unpackaged--0.1.0.sql

Update the Makefile

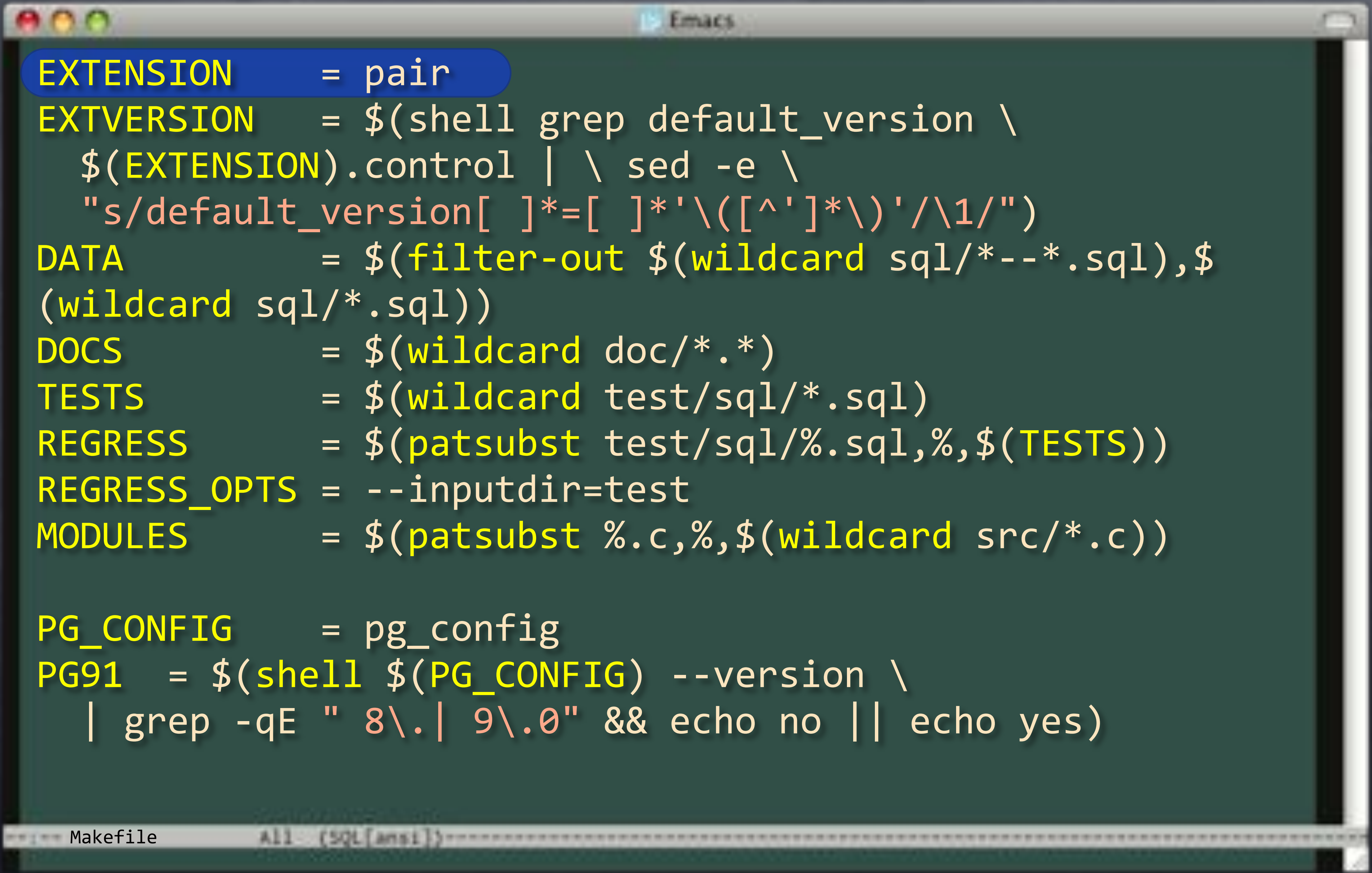


Update the Makefile



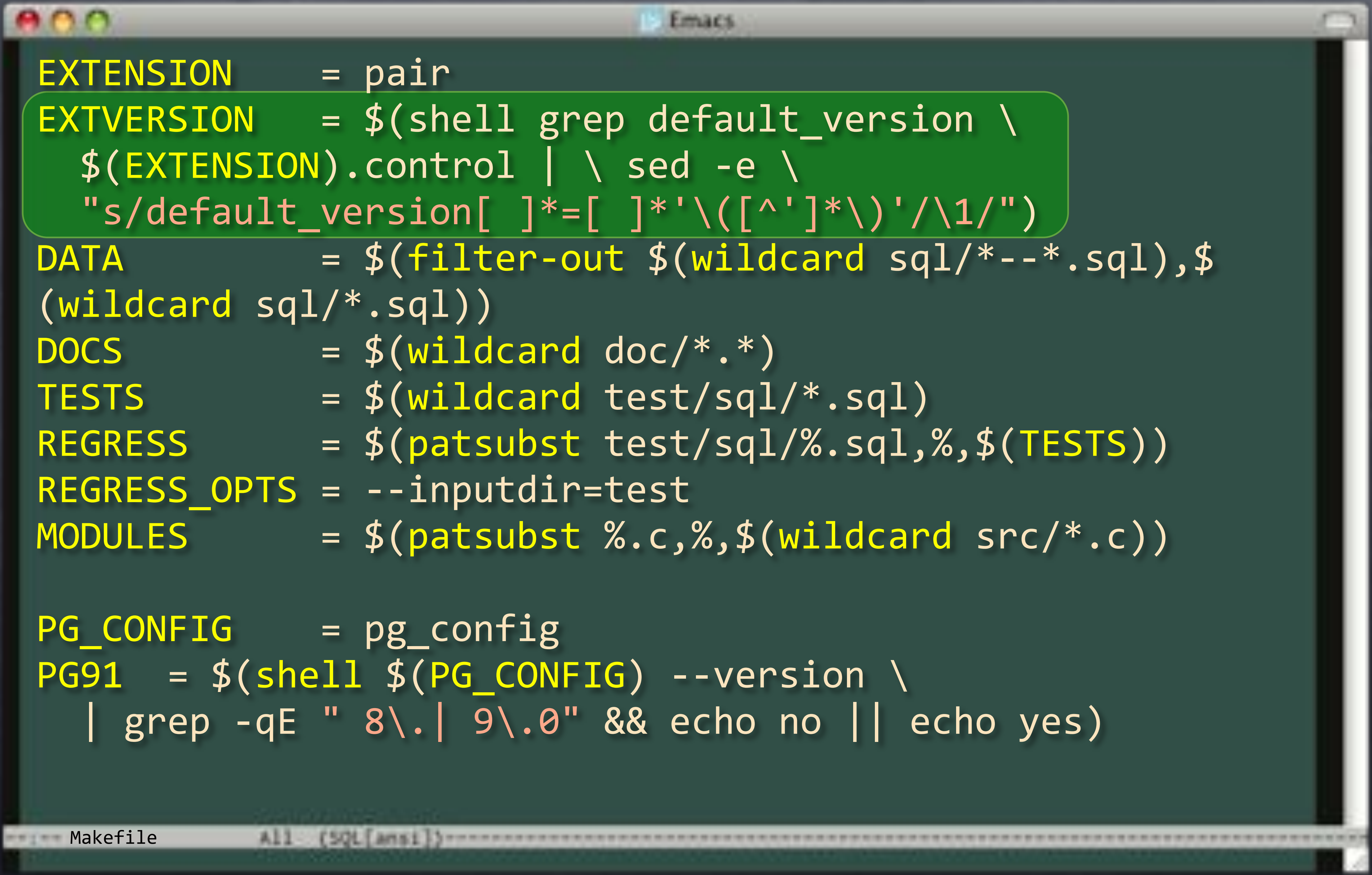
```
EXTENSION    = pair
EXTVERSION   = $(shell grep default_version \
  $(EXTENSION).control | \ sed -e \
  "s/default_version[ ]*=[ ]*'\([^']*\)'/\1/")
DATA         = $(filter-out $(wildcard sql/*--*.sql),$
(wildcard sql/*.sql))
DOCS         = $(wildcard doc/*.*)
TESTS        = $(wildcard test/sql/*.sql)
REGRESS      = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test
MODULES      = $(patsubst %.c,%,$(wildcard src/*.c))

PG_CONFIG    = pg_config
PG91  = $(shell $(PG_CONFIG) --version \
  | grep -qE " 8\.| 9\.0" && echo no || echo yes)
```

Update the Makefile



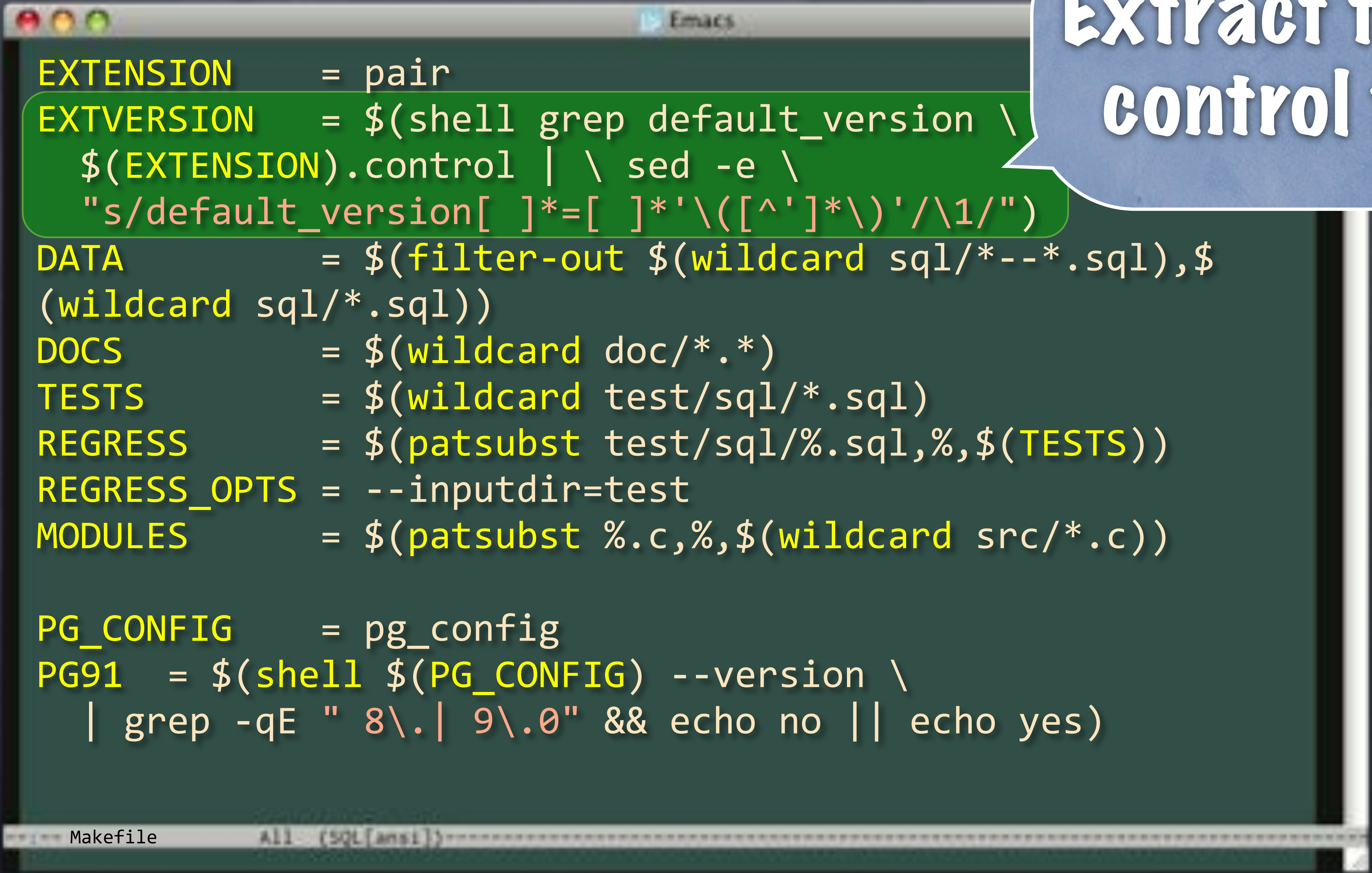
```
EXTENSION    = pair
EXTVERSION   = $(shell grep default_version \
  $(EXTENSION).control | \ sed -e \
  "s/default_version[ ]*=[ ]*'\([^']*\)'/\1/")
DATA         = $(filter-out $(wildcard sql/*--*.sql),$
(wildcard sql/*.sql))
DOCS         = $(wildcard doc/*.*)
TESTS        = $(wildcard test/sql/*.sql)
REGRESS      = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test
MODULES      = $(patsubst %.c,%,$(wildcard src/*.c))

PG_CONFIG    = pg_config
PG91  = $(shell $(PG_CONFIG) --version \
  | grep -qE " 8\.| 9\.0" && echo no || echo yes)
```

Update the Makefile



```
EXTENSION    = pair
EXTVERSION   = $(shell grep default_version \
  $(EXTENSION).control | \ sed -e \
  "s/default_version[ ]*=[ ]*'\([^']*\)'/\1/")
DATA         = $(filter-out $(wildcard sql/*--*.sql),$
(wildcard sql/*.sql))
DOCS         = $(wildcard doc/*.*)
TESTS        = $(wildcard test/sql/*.sql)
REGRESS      = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test
MODULES      = $(patsubst %.c,%,$(wildcard src/*.c))

PG_CONFIG    = pg_config
PG91  = $(shell $(PG_CONFIG) --version \
  | grep -qE " 8\.| 9\.0" && echo no || echo yes)
```

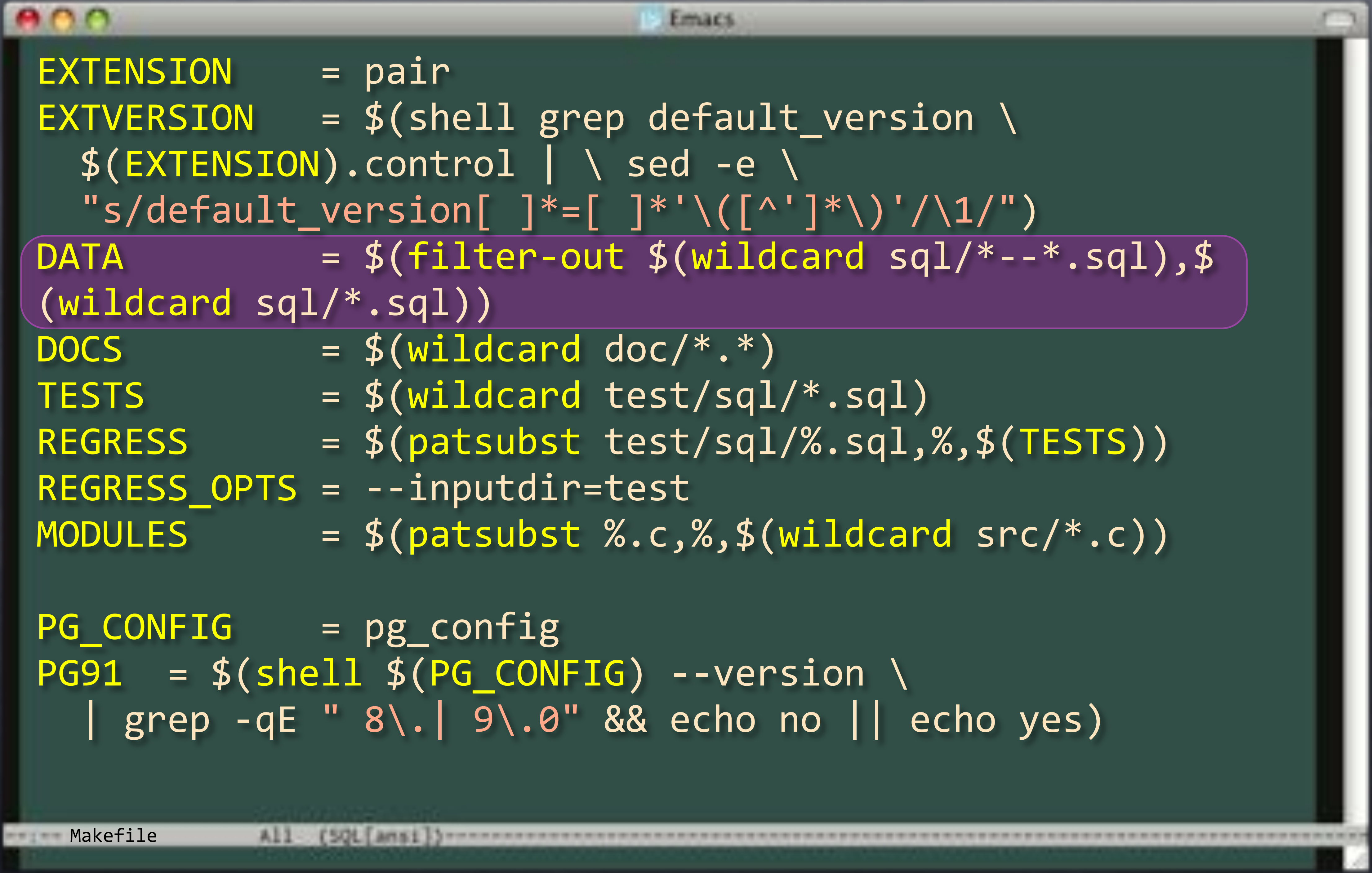
Update the Makefile



```
EXTENSION    = pair
EXTVERSION   = $(shell grep default_version \
  $(EXTENSION).control | \ sed -e \
  "s/default_version[ ]*=[ ]*'\([^']*\)'/\1/")
DATA         = $(filter-out $(wildcard sql/*--*.sql),$
(wildcard sql/*.sql))
DOCS         = $(wildcard doc/*.*)
TESTS        = $(wildcard test/sql/*.sql)
REGRESS      = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test
MODULES      = $(patsubst %.c,%,$(wildcard src/*.c))

PG_CONFIG    = pg_config
PG91  = $(shell $(PG_CONFIG) --version \
  | grep -qE " 8\.| 9\.0" && echo no || echo yes)
```

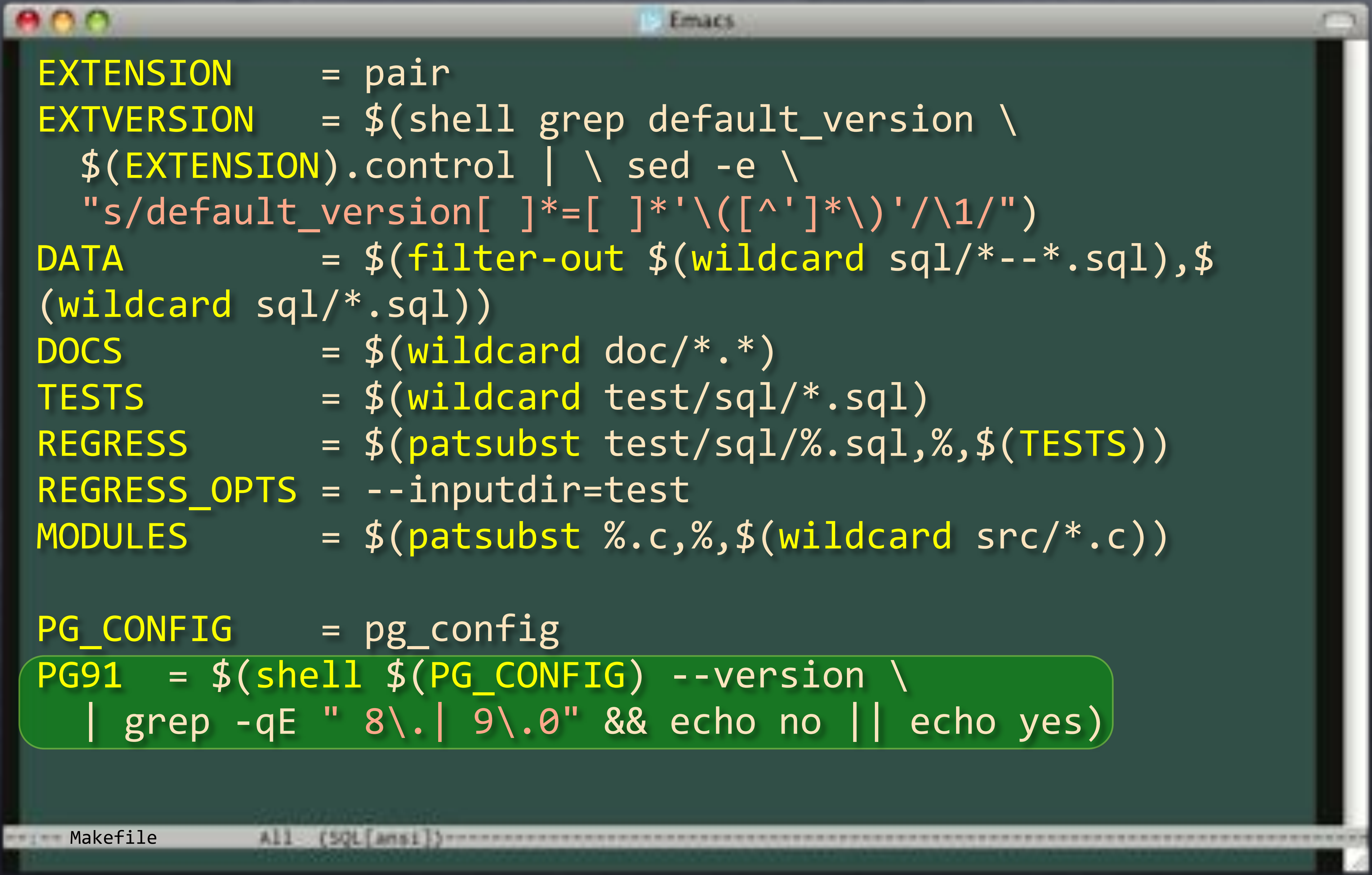
Extract from control file

Update the Makefile



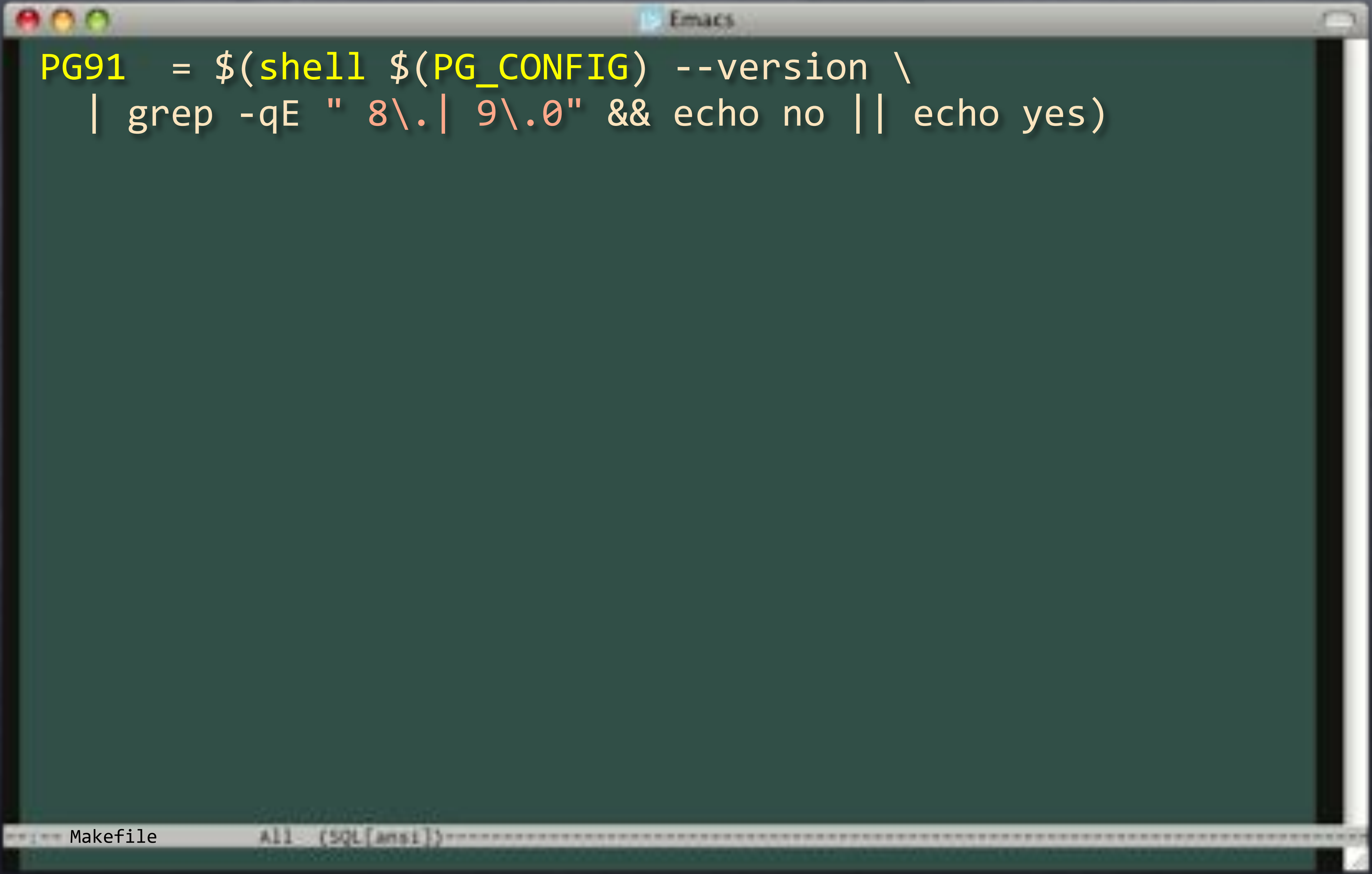
```
EXTENSION    = pair
EXTVERSION   = $(shell grep default_version \
  $(EXTENSION).control | \ sed -e \
  "s/default_version[ ]*=[ ]*'\([^']*\)'/\1/")
DATA         = $(filter-out $(wildcard sql/*--*.sql),$
(wildcard sql/*.sql))
DOCS         = $(wildcard doc/*.*)
TESTS        = $(wildcard test/sql/*.sql)
REGRESS      = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test
MODULES      = $(patsubst %.c,%,$(wildcard src/*.c))

PG_CONFIG    = pg_config
PG91   = $(shell $(PG_CONFIG) --version \
  | grep -qE " 8\.| 9\.0" && echo no || echo yes)
```

Update the Makefile


Emacs

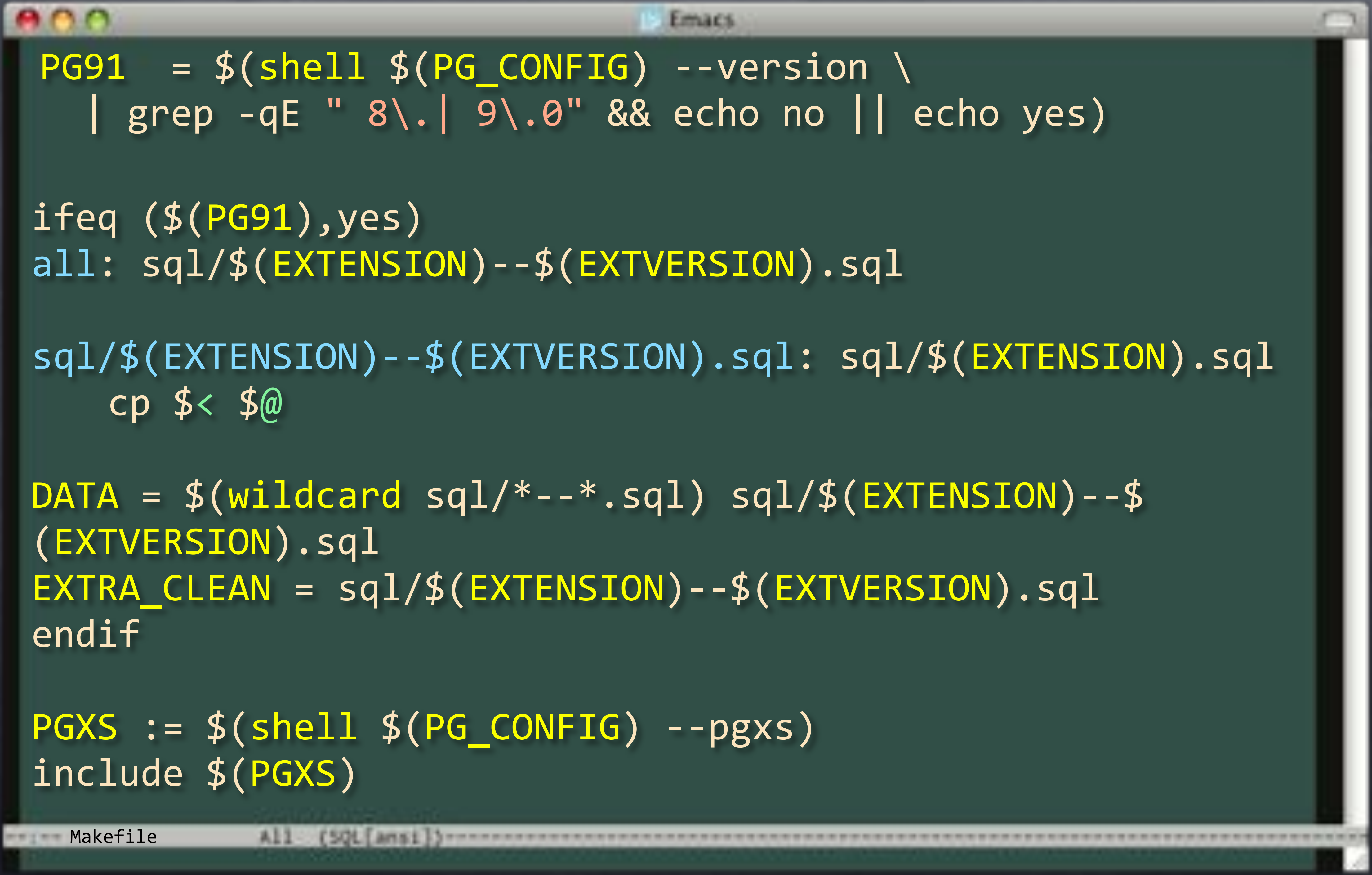
```
EXTENSION    = pair
EXTVERSION   = $(shell grep default_version \
  $(EXTENSION).control | \ sed -e \
  "s/default_version[ ]*=[ ]*'\([^']*\)'/\1/")
DATA         = $(filter-out $(wildcard sql/*--*.sql),$
(wildcard sql/*.sql))
DOCS         = $(wildcard doc/*.*)
TESTS        = $(wildcard test/sql/*.sql)
REGRESS      = $(patsubst test/sql/%.sql,%,$(TESTS))
REGRESS_OPTS = --inputdir=test
MODULES      = $(patsubst %.c,%,$(wildcard src/*.c))

PG_CONFIG    = pg_config
PG91   = $(shell $(PG_CONFIG) --version \
  | grep -qE " 8\.| 9\.0" && echo no || echo yes)
```

Update the Makefile



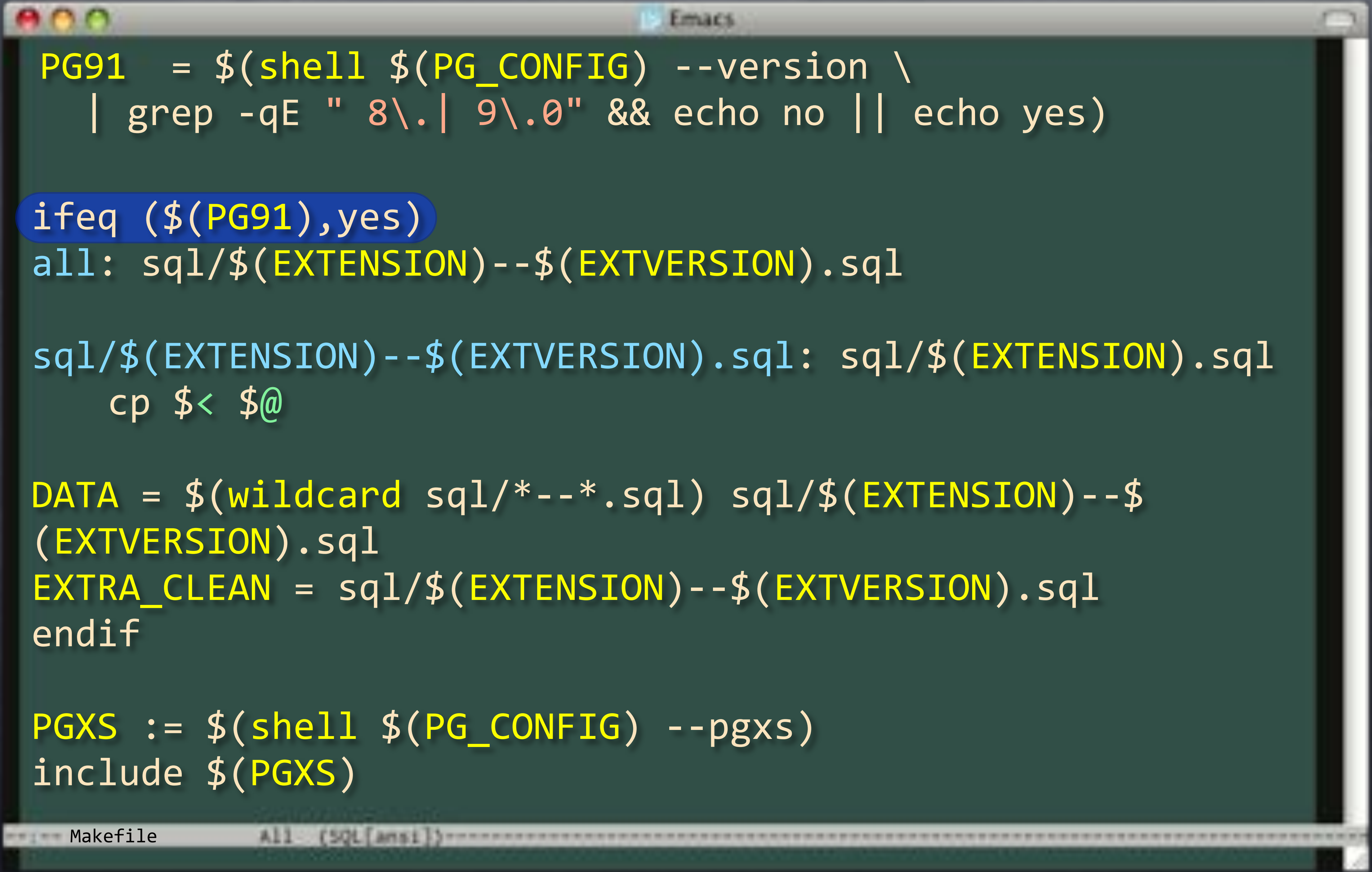
```
PG91  = $(shell $(PG_CONFIG) --version \
  | grep -qE " 8\.| 9\.0" && echo no || echo yes)
```

Update the Makefile



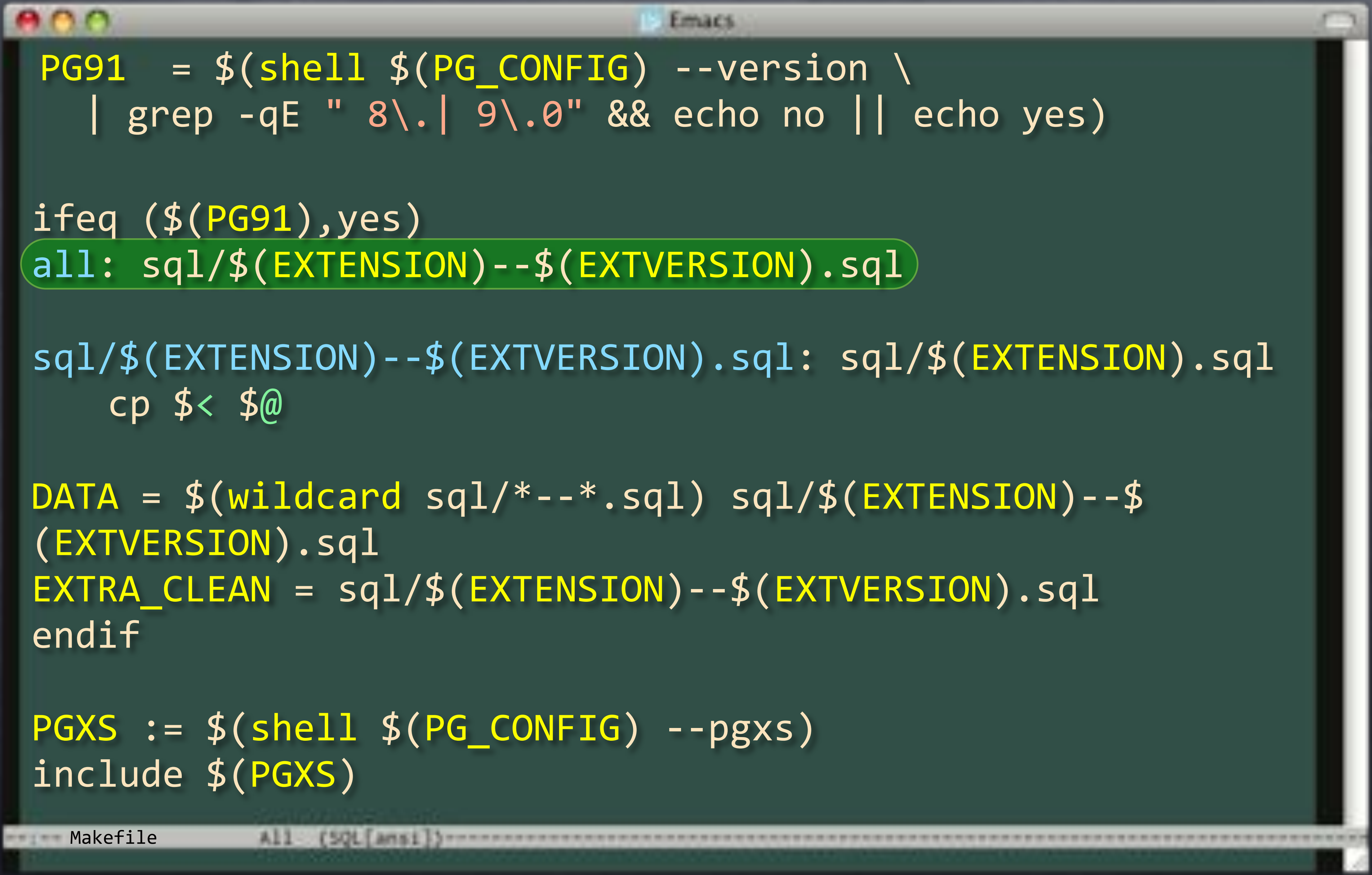
```
PG91   = $(shell $(PG_CONFIG) --version \
   | grep -qE " 8\.| 9\.0" && echo no || echo yes)

ifeq ($(PG91),yes)
all: sql/$(EXTENSION)--$(EXTVERSION).sql

sql/$(EXTENSION)--$(EXTVERSION).sql: sql/$(EXTENSION).sql
	cp $< $@

DATA = $(wildcard sql/*--*.sql) sql/$(EXTENSION)--$
(EXTVERSION).sql
EXTRA_CLEAN = sql/$(EXTENSION)--$(EXTVERSION).sql
endif

PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

Update the Makefile



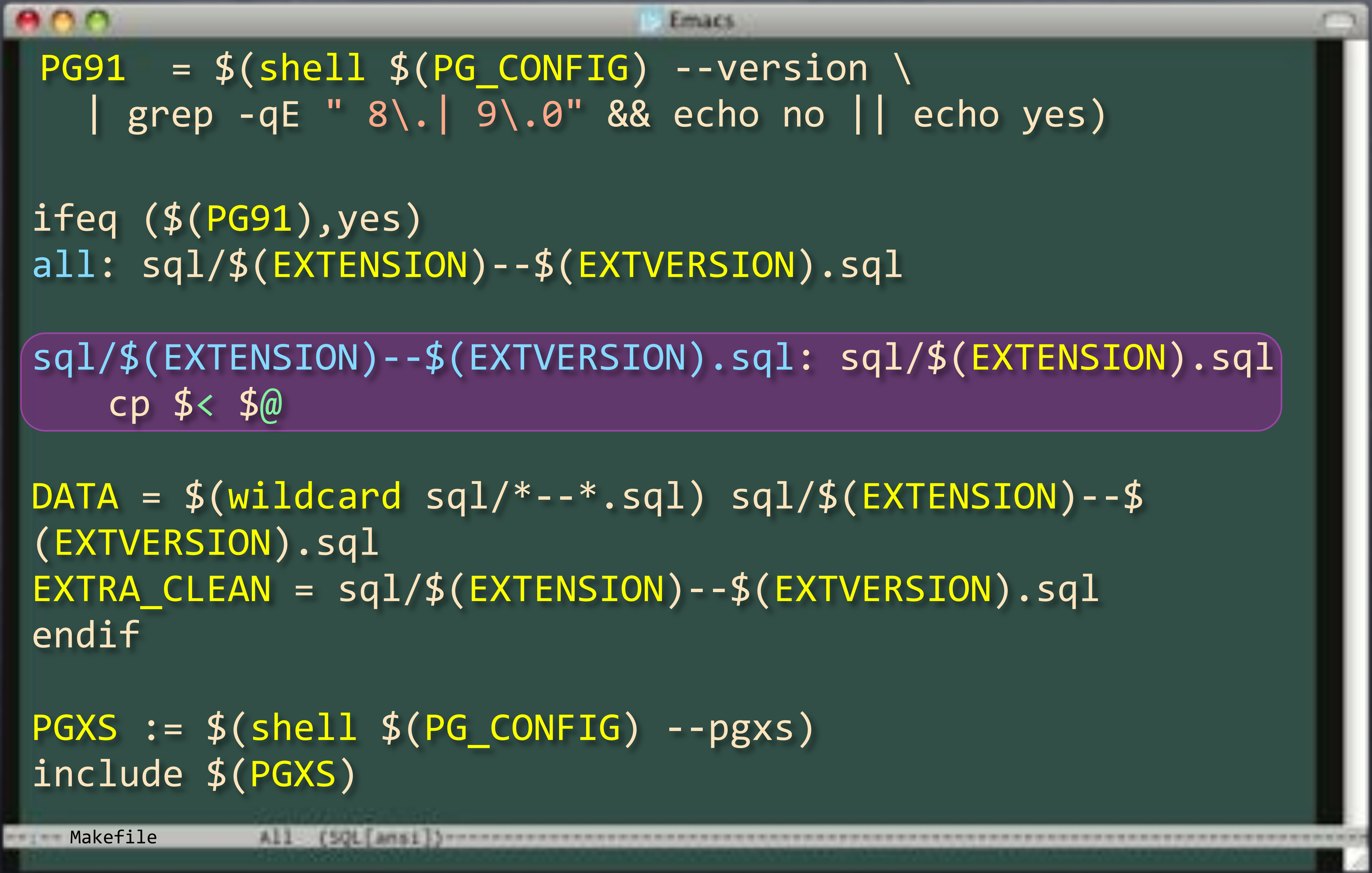
```
PG91   = $(shell $(PG_CONFIG) --version \
   | grep -qE " 8\.| 9\.0" && echo no || echo yes)

ifeq ($(PG91),yes)
all: sql/$(EXTENSION)--$(EXTVERSION).sql

sql/$(EXTENSION)--$(EXTVERSION).sql: sql/$(EXTENSION).sql
	cp $< $@

DATA = $(wildcard sql/*--*.sql) sql/$(EXTENSION)--$
(EXTVERSION).sql
EXTRA_CLEAN = sql/$(EXTENSION)--$(EXTVERSION).sql
endif

PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

Update the Makefile



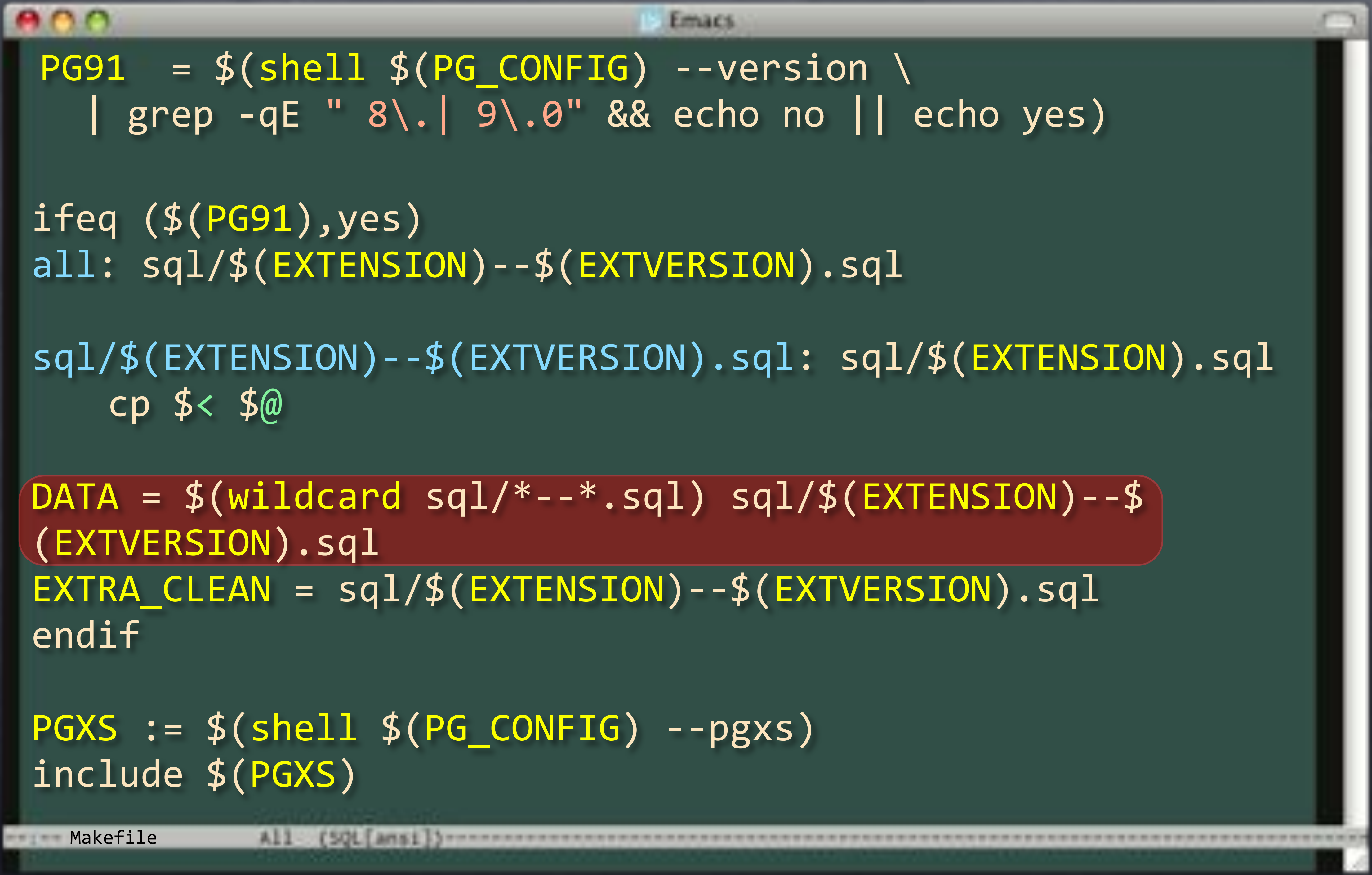
```
PG91   = $(shell $(PG_CONFIG) --version \
   | grep -qE " 8\.| 9\.0" && echo no || echo yes)

ifeq ($(PG91),yes)
all: sql/$(EXTENSION)--$(EXTVERSION).sql

sql/$(EXTENSION)--$(EXTVERSION).sql: sql/$(EXTENSION).sql
	cp $< $@

DATA = $(wildcard sql/*--*.sql) sql/$(EXTENSION)--$
(EXTVERSION).sql
EXTRA_CLEAN = sql/$(EXTENSION)--$(EXTVERSION).sql
endif

PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

Update the Makefile



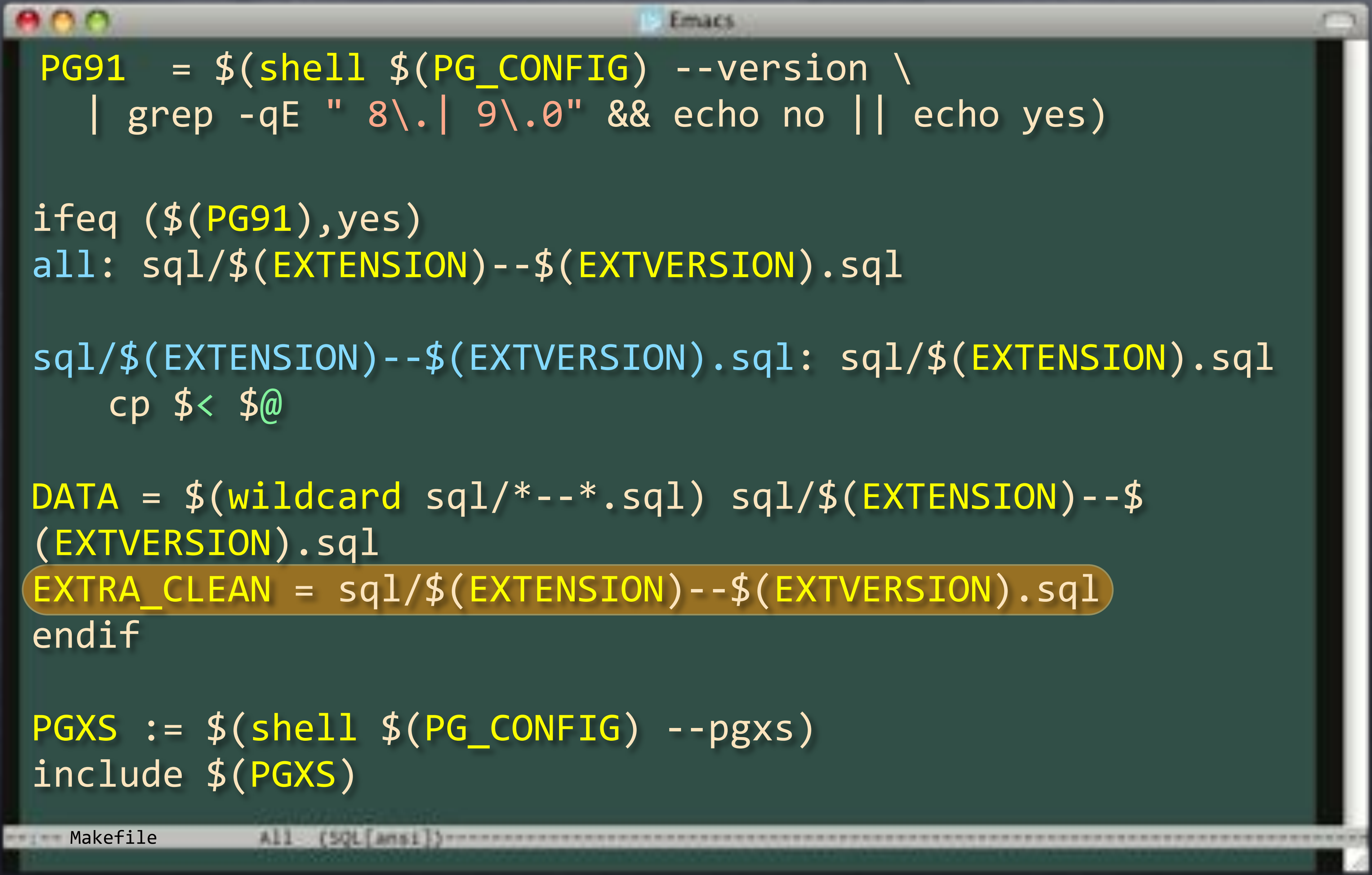
```
PG91   = $(shell $(PG_CONFIG) --version \
   | grep -qE " 8\.| 9\.0" && echo no || echo yes)

ifeq ($(PG91),yes)
all: sql/$(EXTENSION)--$(EXTVERSION).sql

sql/$(EXTENSION)--$(EXTVERSION).sql: sql/$(EXTENSION).sql
	cp $< $@

DATA = $(wildcard sql/*--*.sql) sql/$(EXTENSION)--$
(EXTVERSION).sql
EXTRA_CLEAN = sql/$(EXTENSION)--$(EXTVERSION).sql
endif

PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

Update the Makefile



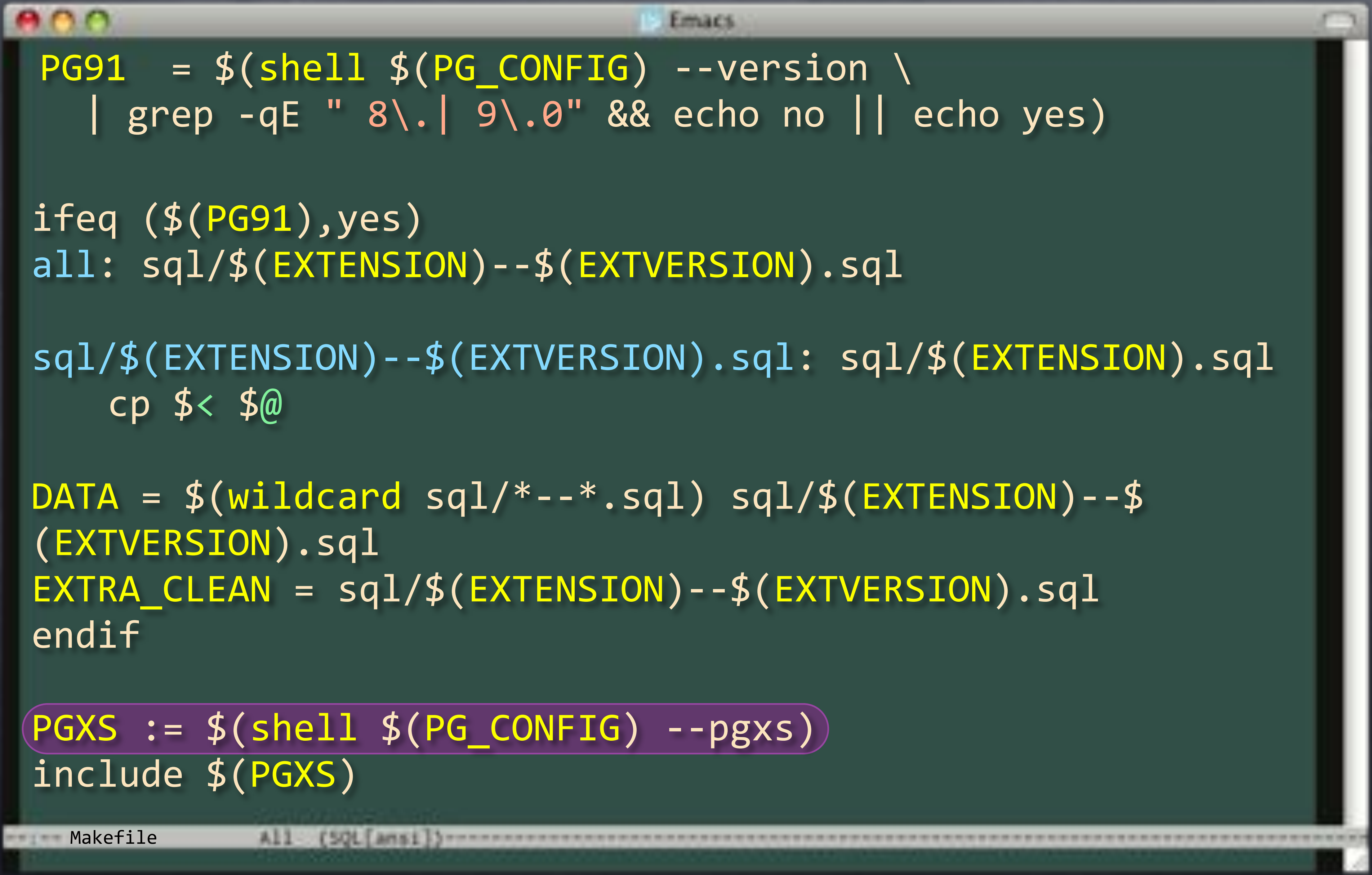
```
PG91   = $(shell $(PG_CONFIG) --version \
   | grep -qE " 8\.| 9\.0" && echo no || echo yes)

ifeq ($(PG91),yes)
all: sql/$(EXTENSION)--$(EXTVERSION).sql

sql/$(EXTENSION)--$(EXTVERSION).sql: sql/$(EXTENSION).sql
	cp $< $@

DATA = $(wildcard sql/*--*.sql) sql/$(EXTENSION)--$
(EXTVERSION).sql
EXTRA_CLEAN = sql/$(EXTENSION)--$(EXTVERSION).sql
endif

PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

Update the Makefile



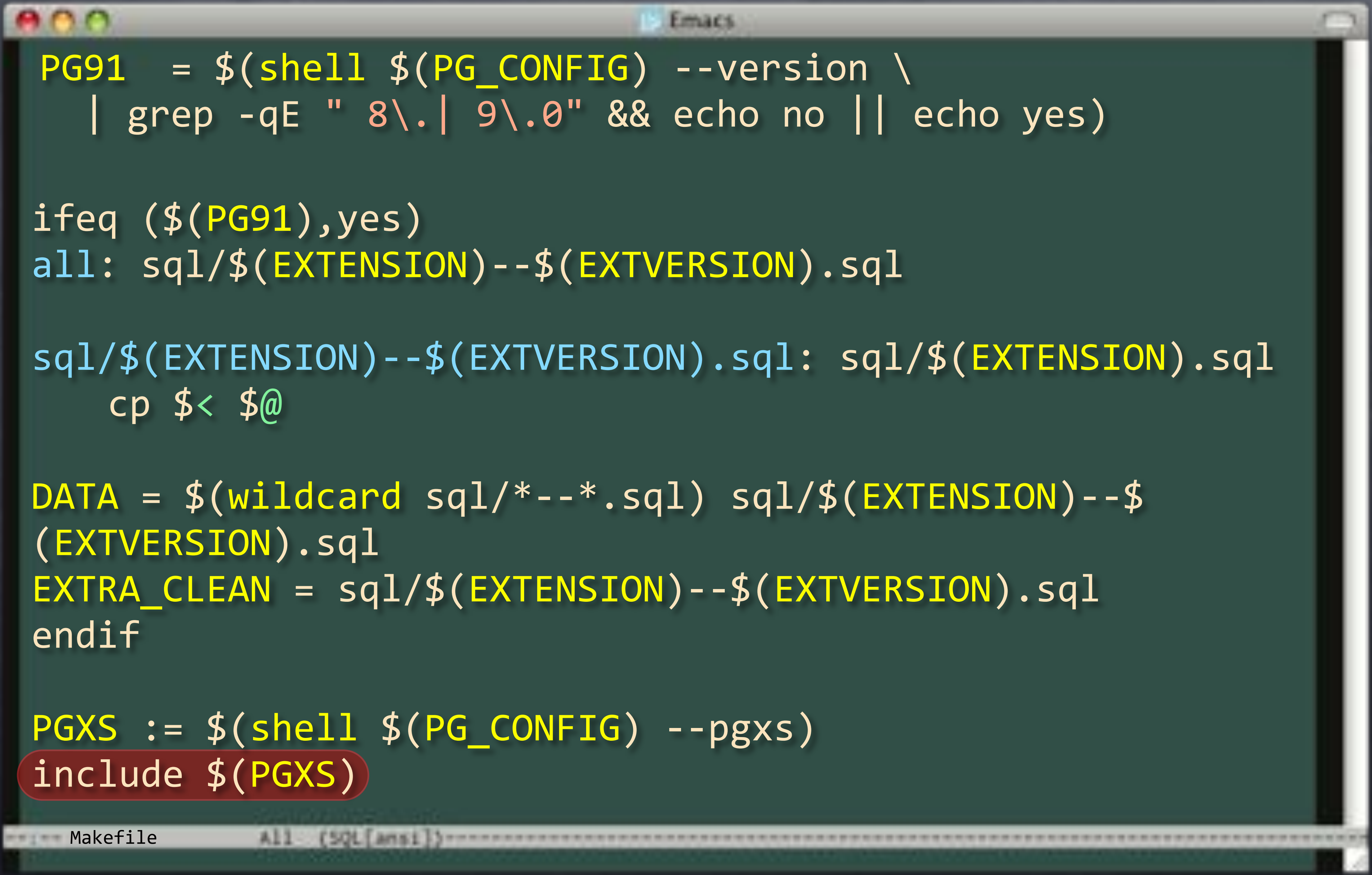
```
PG91   = $(shell $(PG_CONFIG) --version \
   | grep -qE " 8\.| 9\.0" && echo no || echo yes)

ifeq ($(PG91),yes)
all: sql/$(EXTENSION)--$(EXTVERSION).sql

sql/$(EXTENSION)--$(EXTVERSION).sql: sql/$(EXTENSION).sql
	cp $< $@

DATA = $(wildcard sql/*--*.sql) sql/$(EXTENSION)--$
(EXTVERSION).sql
EXTRA_CLEAN = sql/$(EXTENSION)--$(EXTVERSION).sql
endif

PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

Update the Makefile



```
PG91   = $(shell $(PG_CONFIG) --version \
   | grep -qE " 8\.| 9\.0" && echo no || echo yes)

ifeq ($(PG91),yes)
all: sql/$(EXTENSION)--$(EXTVERSION).sql

sql/$(EXTENSION)--$(EXTVERSION).sql: sql/$(EXTENSION).sql
	cp $< $@

DATA = $(wildcard sql/*--*.sql) sql/$(EXTENSION)--$
(EXTVERSION).sql
EXTRA_CLEAN = sql/$(EXTENSION)--$(EXTVERSION).sql
endif

PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

Update the Makefile



```
PG91   = $(shell $(PG_CONFIG) --version \
  | grep -qE " 8\.| 9\.0" && echo no || echo yes)

ifeq ($(PG91),yes)
all: sql/$(EXTENSION)--$(EXTVERSION).sql

sql/$(EXTENSION)--$(EXTVERSION).sql: sql/$(EXTENSION).sql
	cp $< $@

DATA = $(wildcard sql/*--*.sql) sql/$(EXTENSION)--$
(EXTVERSION).sql
EXTRA_CLEAN = sql/$(EXTENSION)--$(EXTVERSION).sql
endif

PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

Or Forget It

Or Forget It

- Copy Makefile

Or Forget It

- Copy Makefile
- Edit first line

Or Forget It

- Copy Makefile
- Edit first line
  - EXTENSION=pair

Or Forget It

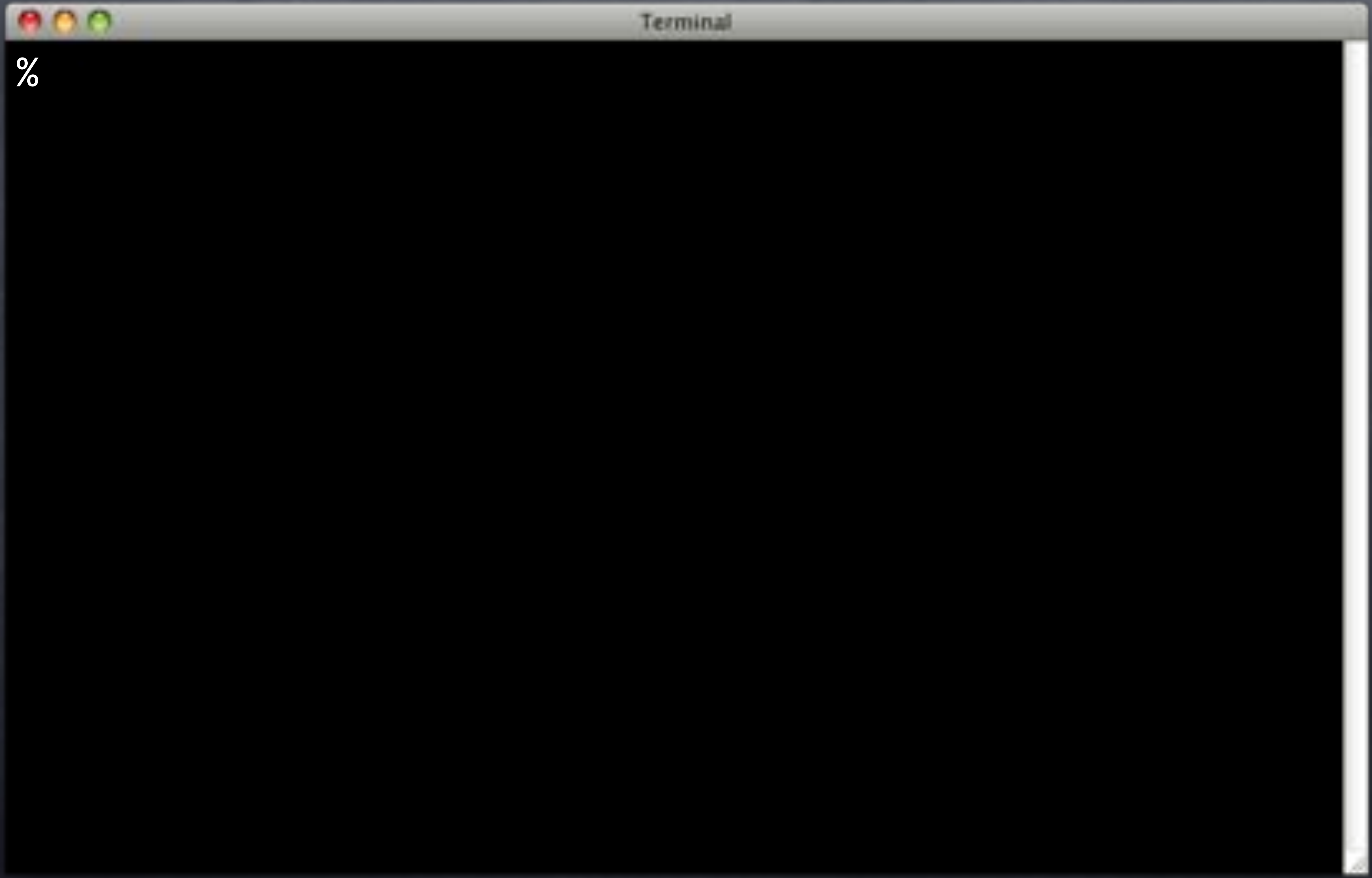
- Copy Makefile
- Edit first line
  - EXTENSION=pair
- Ignore the rest

Or Forget It

- Copy Makefile
- Edit first line
  - EXTENSION=pair
- Ignore the rest

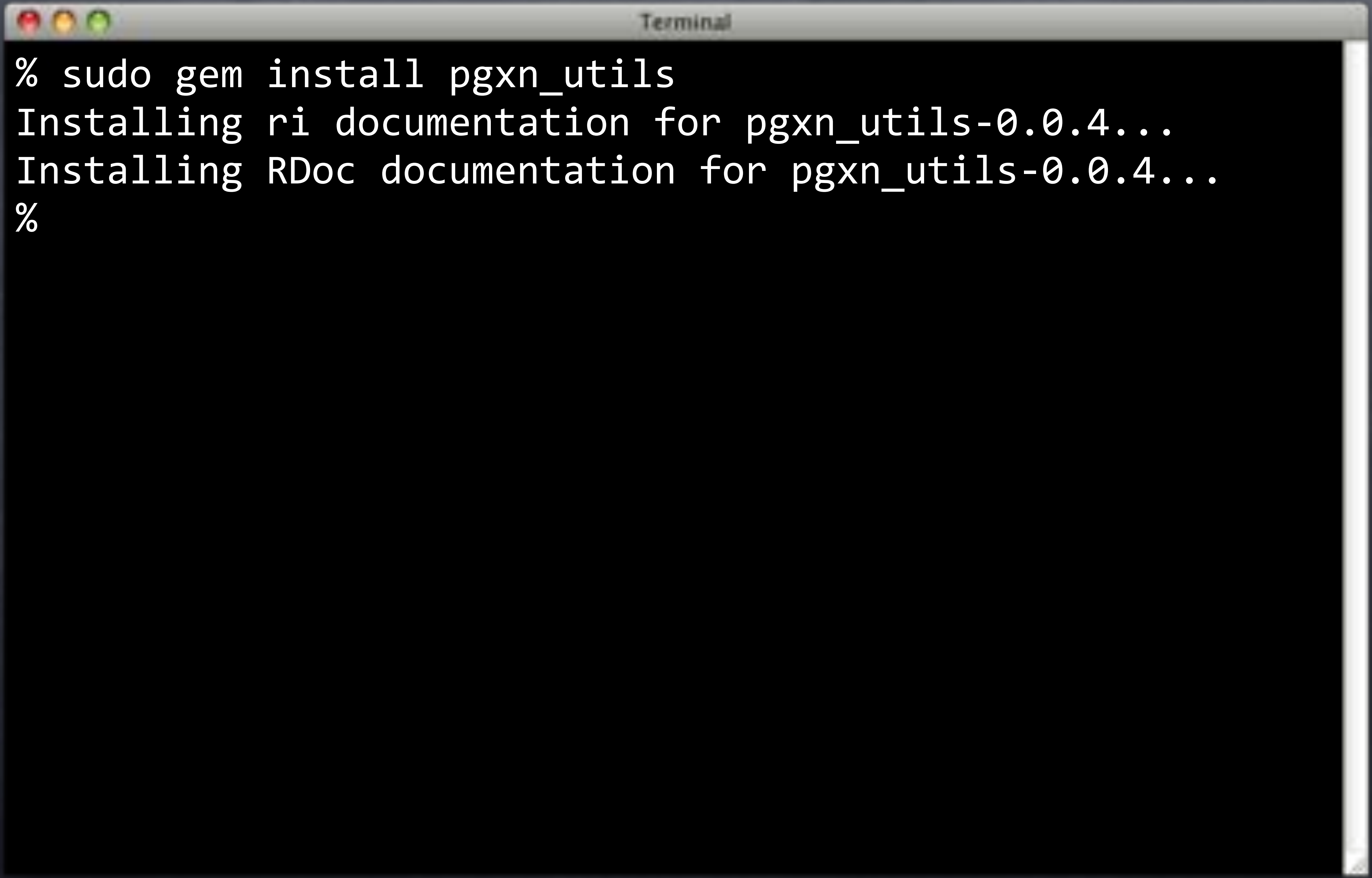
Better still...

Skeleton in the Closet



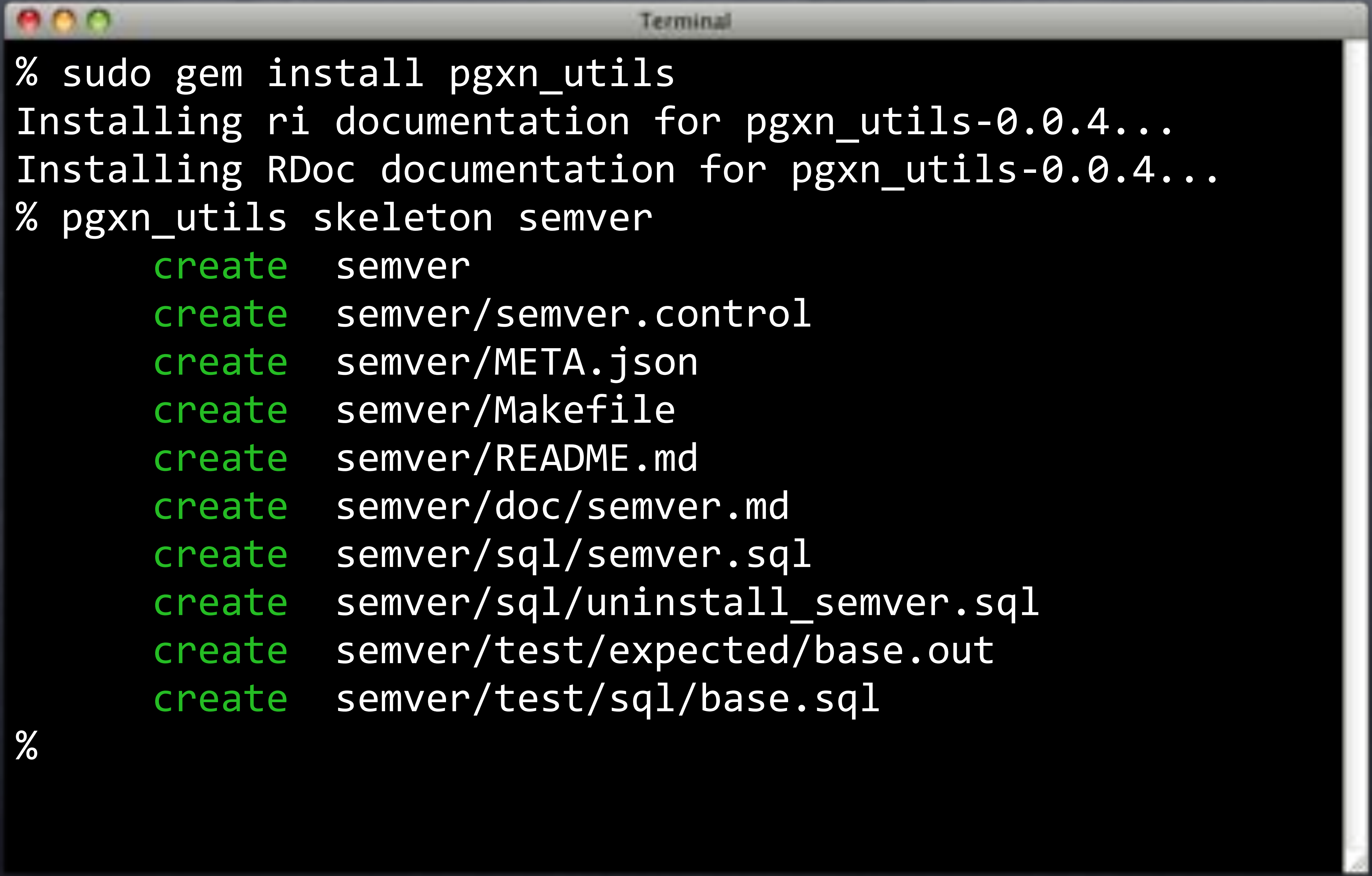
```
%
```

Skeleton in the Closet



```
% sudo gem install pgxn_utils
Installing ri documentation for pgxn_utils-0.0.4...
Installing RDoc documentation for pgxn_utils-0.0.4...
%
```

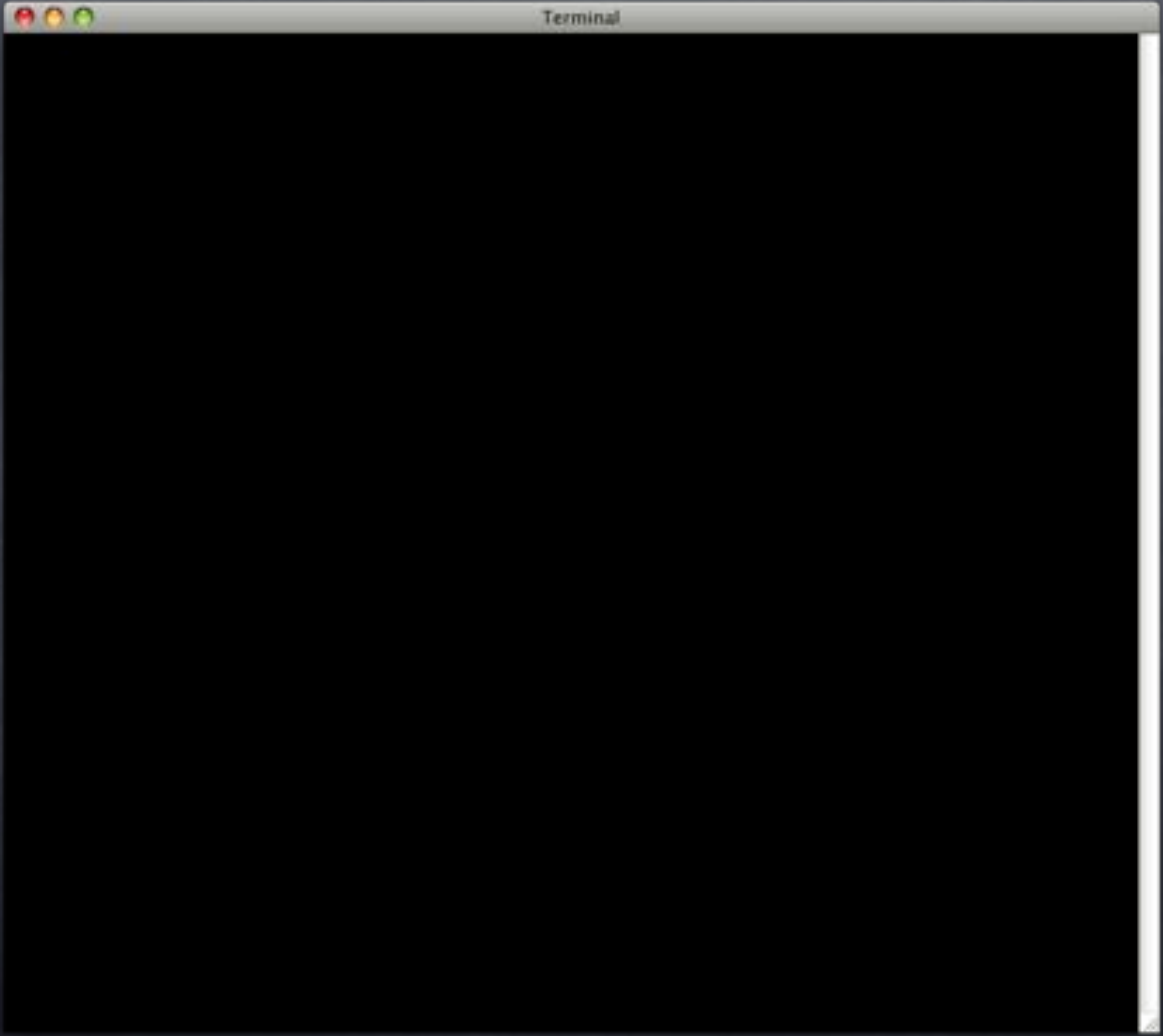
Skeleton in the Closet

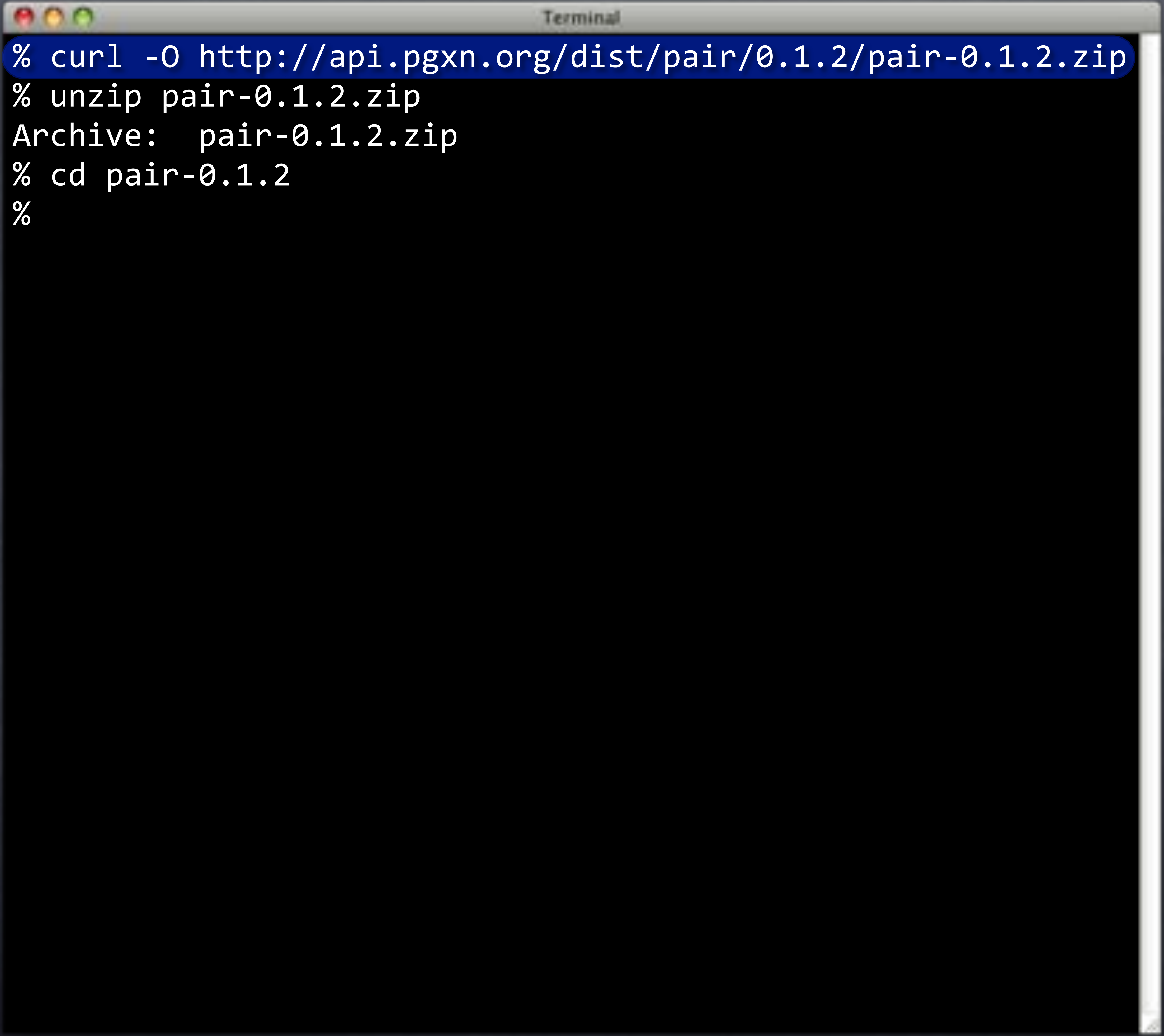


Terminal

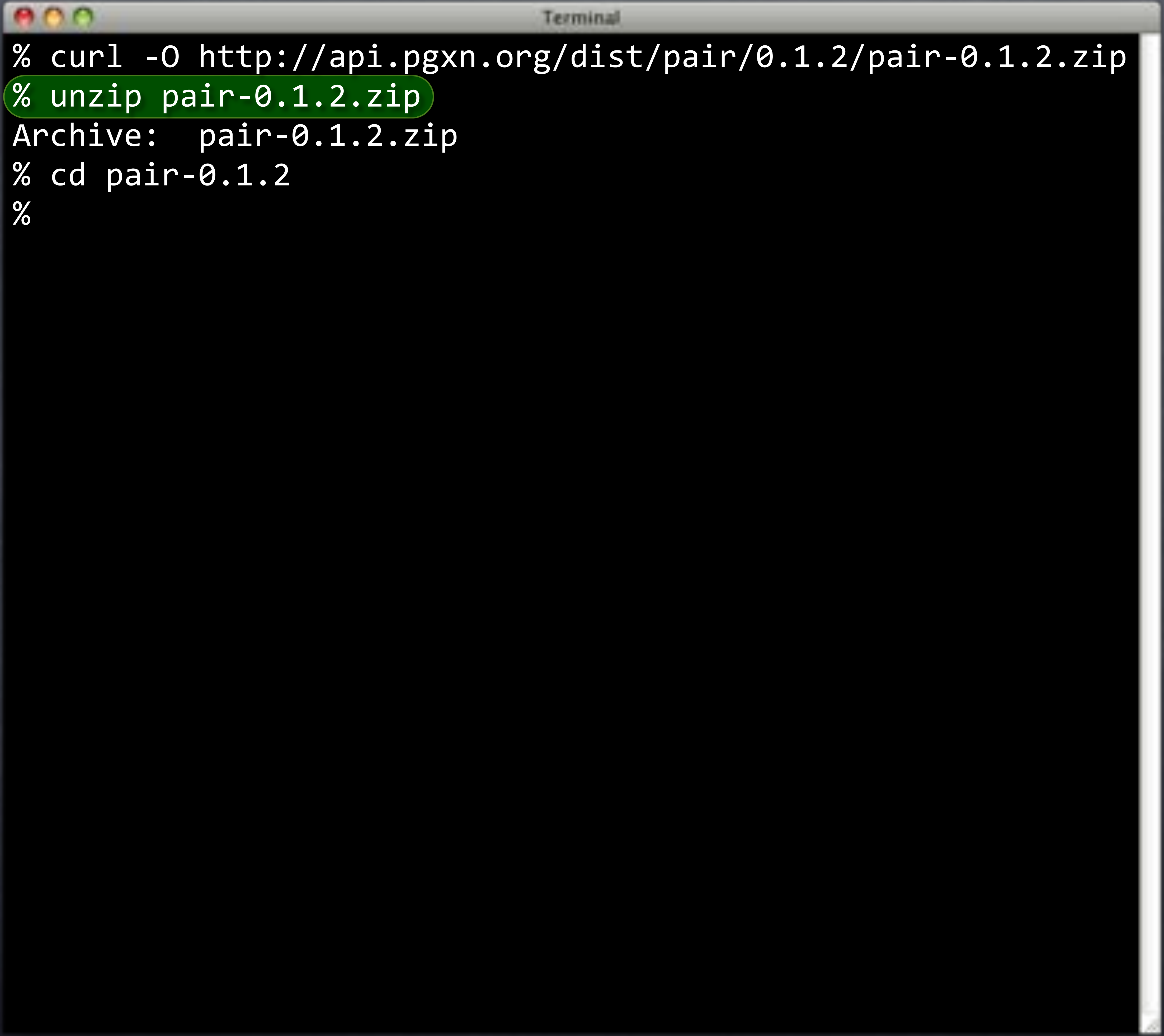
```
% sudo gem install pgxn_utils
Installing ri documentation for pgxn_utils-0.0.4...
Installing RDoc documentation for pgxn_utils-0.0.4...
% pgxn_utils skeleton semver
      create  semver
      create  semver/semver.control
      create  semver/META.json
      create  semver/Makefile
      create  semver/README.md
      create  semver/doc/semver.md
      create  semver/sql/semver.sql
      create  semver/sql/uninstall_semver.sql
      create  semver/test/expected/base.out
      create  semver/test/sql/base.sql
%
```

Thank you
Dickson S. Guedes

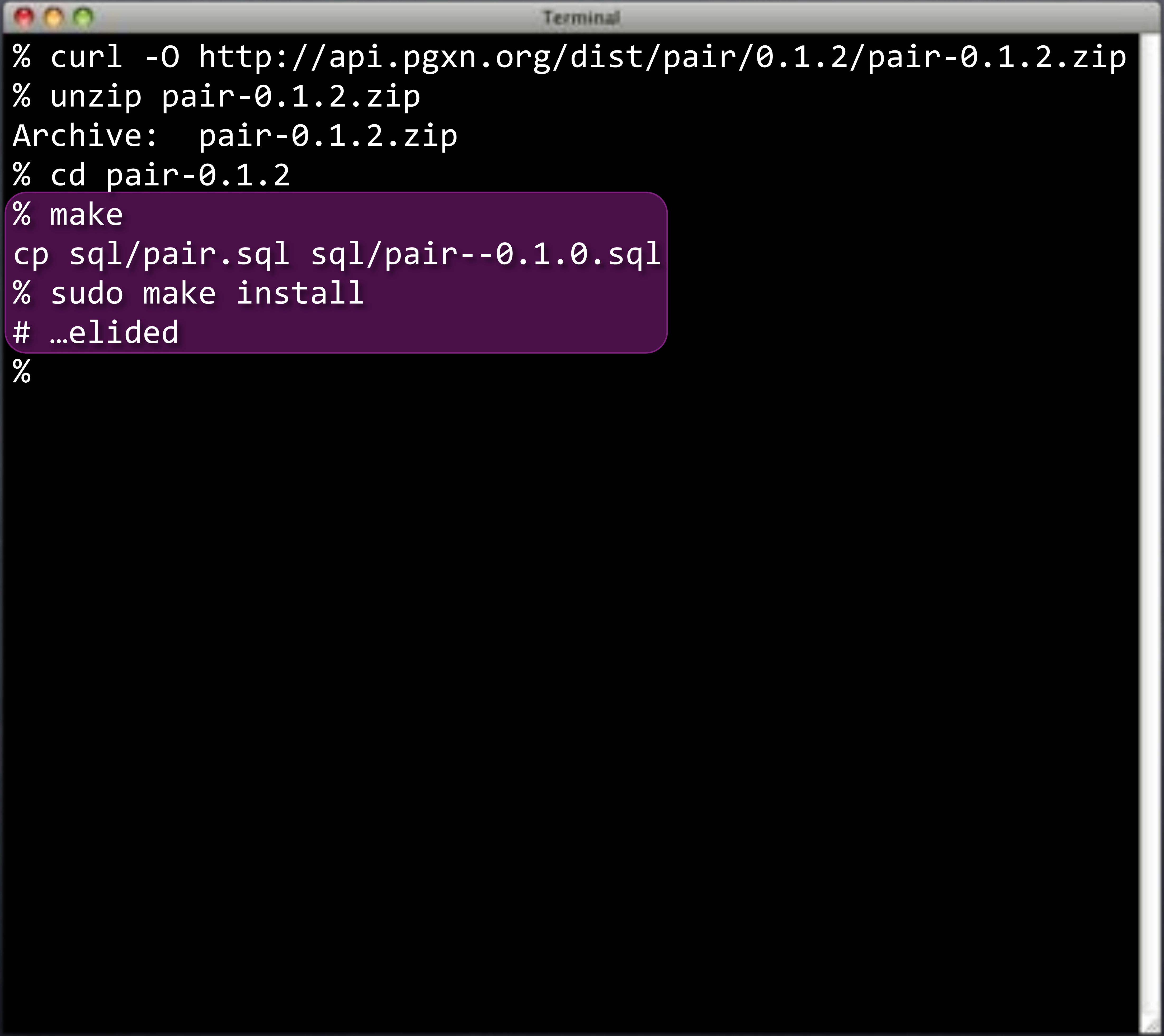

Terminal



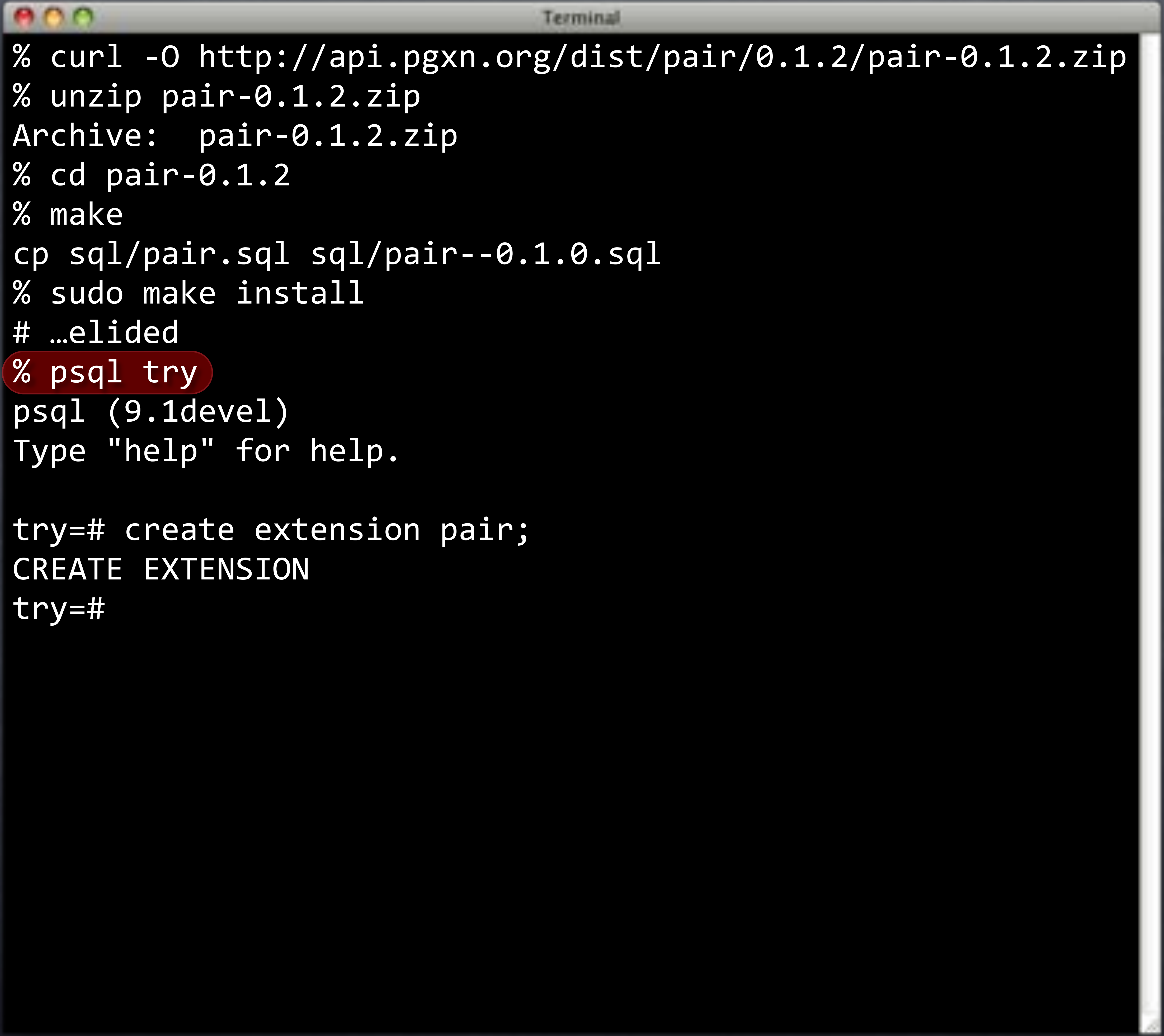
```
% curl -O http://api.pgxn.org/dist/pair/0.1.2/pair-0.1.2.zip
% unzip pair-0.1.2.zip
Archive:  pair-0.1.2.zip
% cd pair-0.1.2
%
```



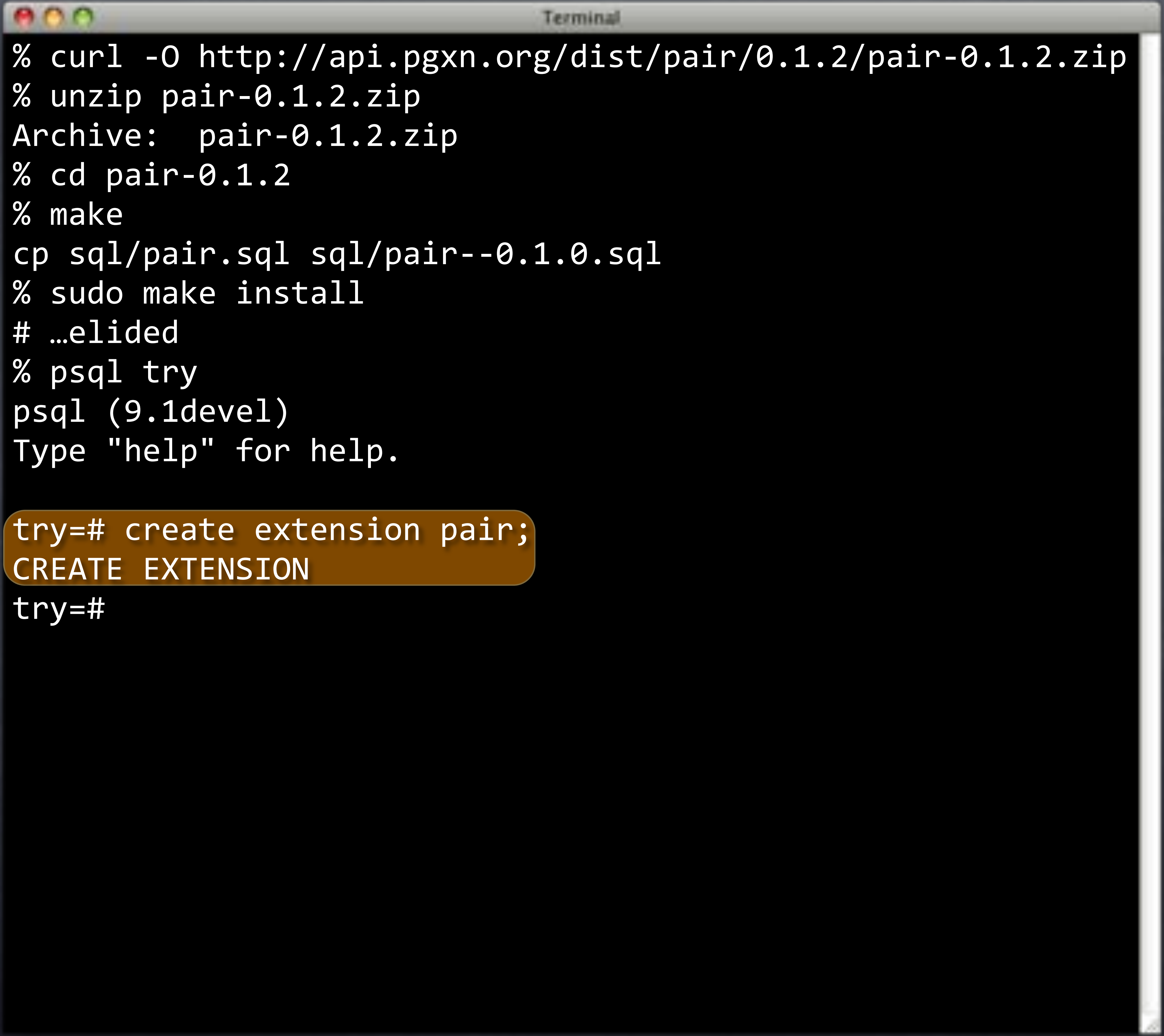
```
% curl -O http://api.pgxn.org/dist/pair/0.1.2/pair-0.1.2.zip
% unzip pair-0.1.2.zip
Archive:  pair-0.1.2.zip
% cd pair-0.1.2
%
```



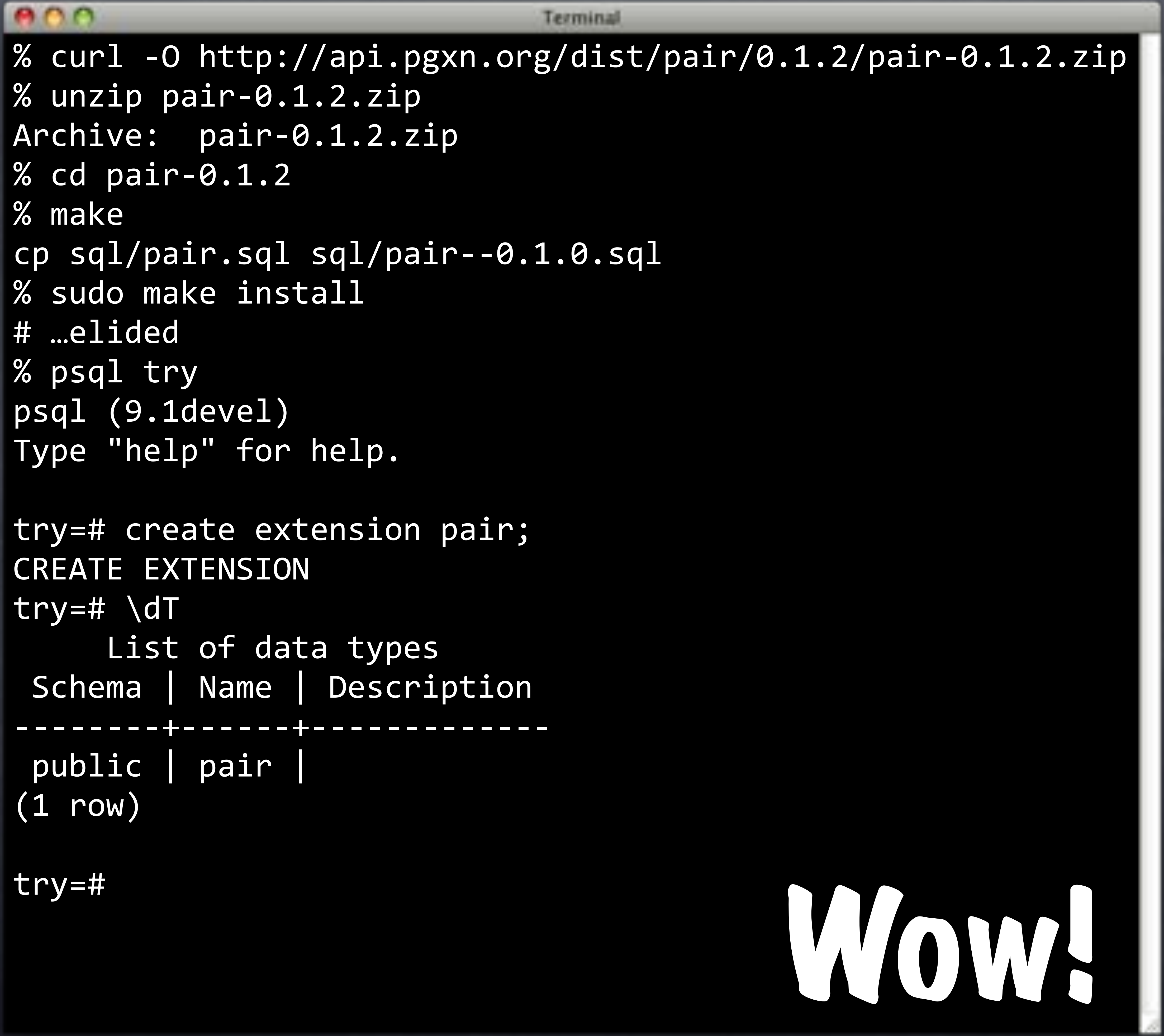
```
% curl -O http://api.pgxn.org/dist/pair/0.1.2/pair-0.1.2.zip
% unzip pair-0.1.2.zip
Archive:  pair-0.1.2.zip
% cd pair-0.1.2
% make
cp sql/pair.sql sql/pair--0.1.0.sql
% sudo make install
# …elided
%
```



```
% curl -O http://api.pgxn.org/dist/pair/0.1.2/pair-0.1.2.zip
% unzip pair-0.1.2.zip
Archive:  pair-0.1.2.zip
% cd pair-0.1.2
% make
cp sql/pair.sql sql/pair--0.1.0.sql
% sudo make install
# …elided
% psql try
psql (9.1devel)
Type "help" for help.

try=# create extension pair;
CREATE EXTENSION
try=#
```

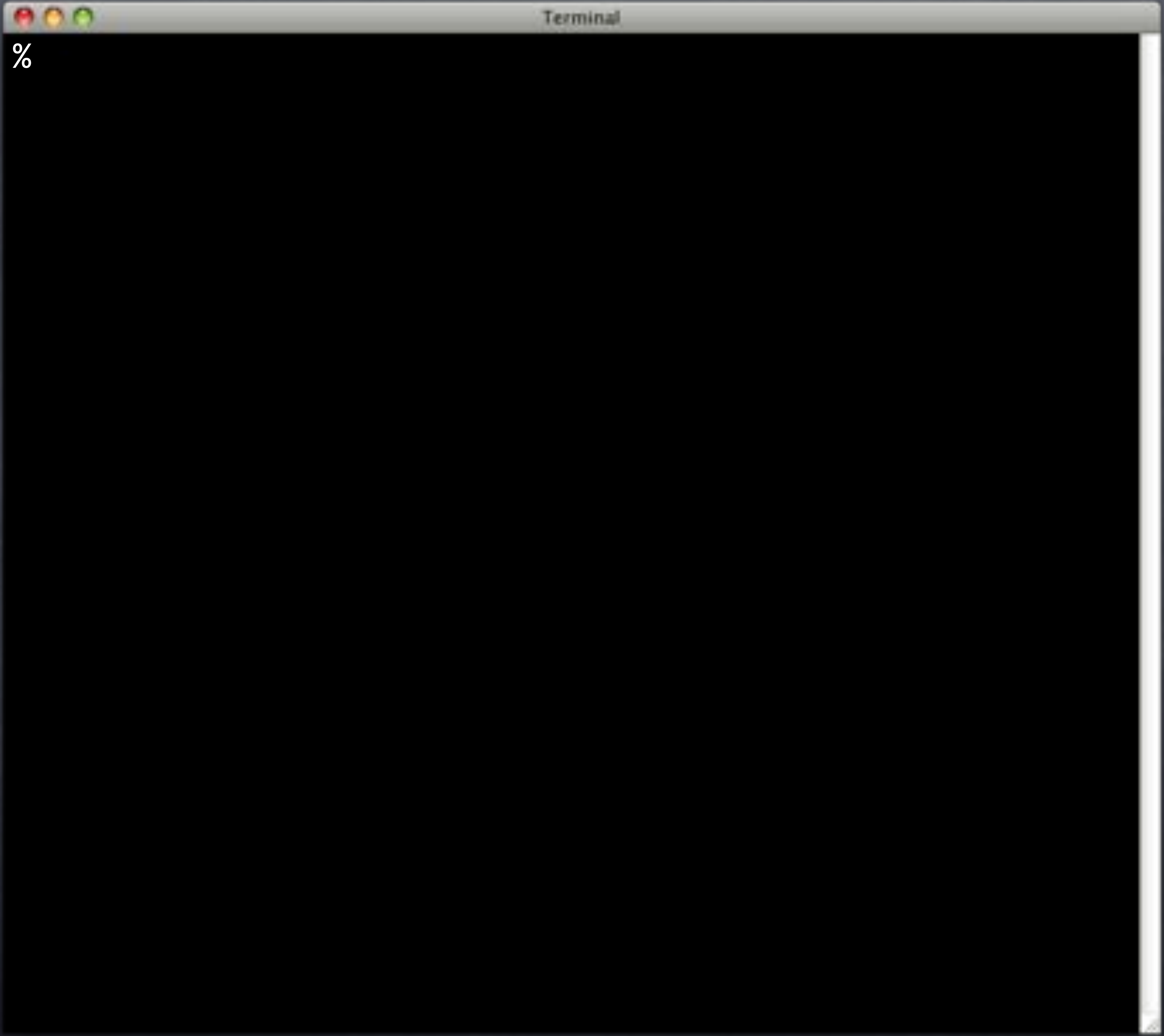

Terminal

```
% curl -O http://api.pgxn.org/dist/pair/0.1.2/pair-0.1.2.zip
% unzip pair-0.1.2.zip
Archive:  pair-0.1.2.zip
% cd pair-0.1.2
% make
cp sql/pair.sql sql/pair--0.1.0.sql
% sudo make install
# …elided
% psql try
psql (9.1devel)
Type "help" for help.

try=# create extension pair;
CREATE EXTENSION
try=#
```

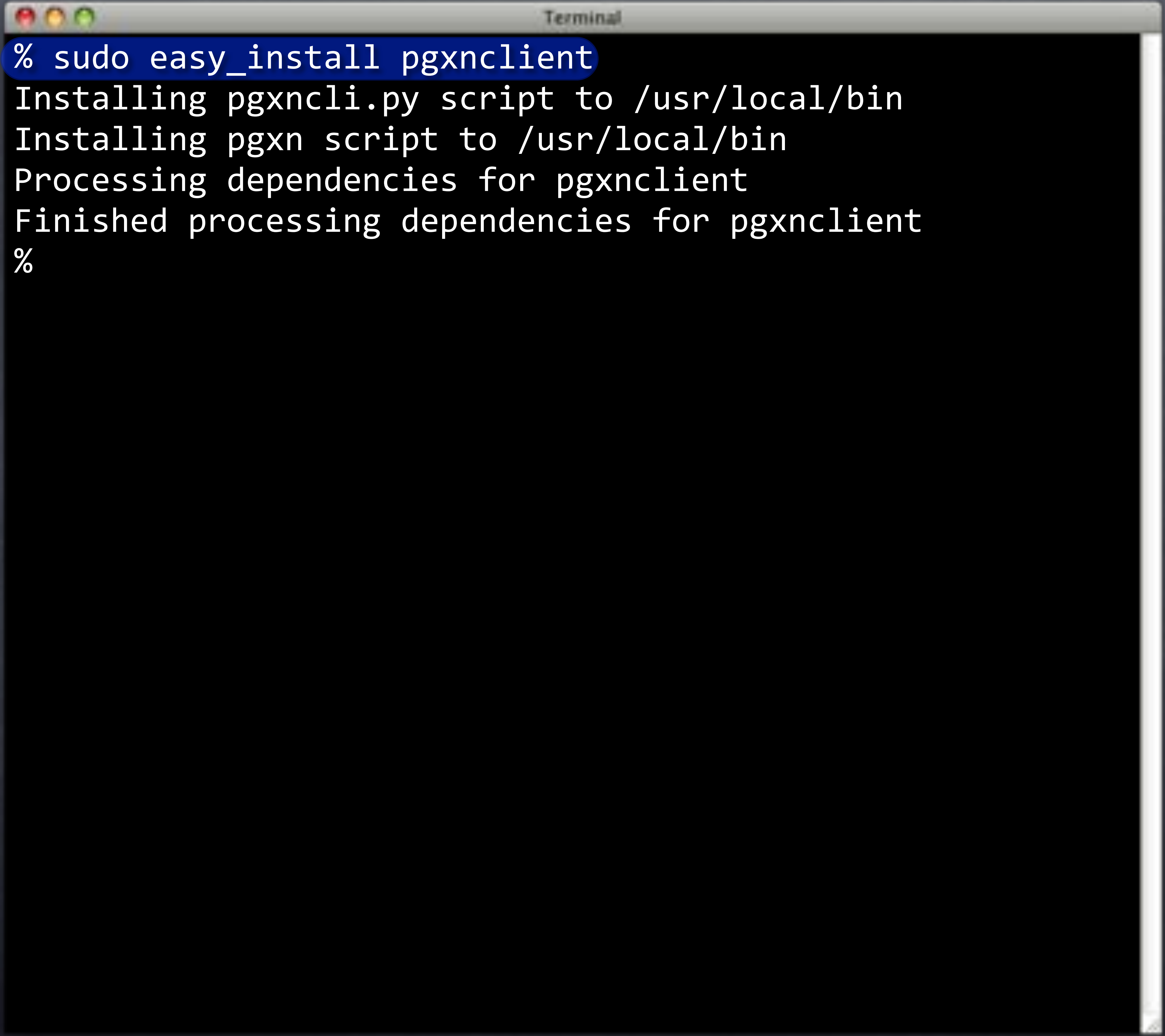


```
% curl -O http://api.pgxn.org/dist/pair/0.1.2/pair-0.1.2.zip
% unzip pair-0.1.2.zip
Archive:  pair-0.1.2.zip
% cd pair-0.1.2
% make
cp sql/pair.sql sql/pair--0.1.0.sql
% sudo make install
# …elided
% psql try
psql (9.1devel)
Type "help" for help.

try=# create extension pair;
CREATE EXTENSION
try=# \dT
     List of data types
 Schema | Name | Description
--------+------+-------------
 public | pair |
(1 row)

try=#
```

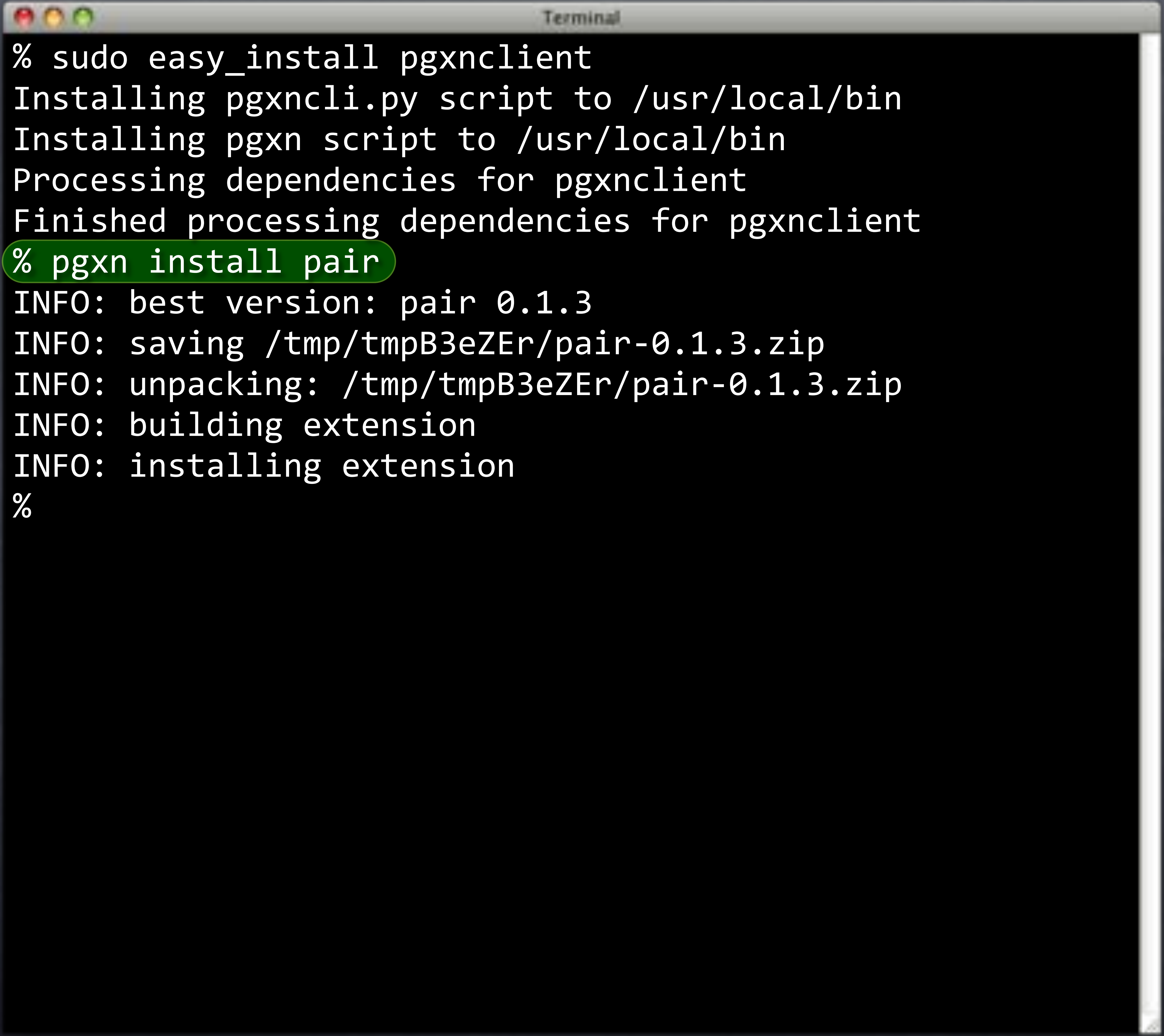
Wow!


Terminal

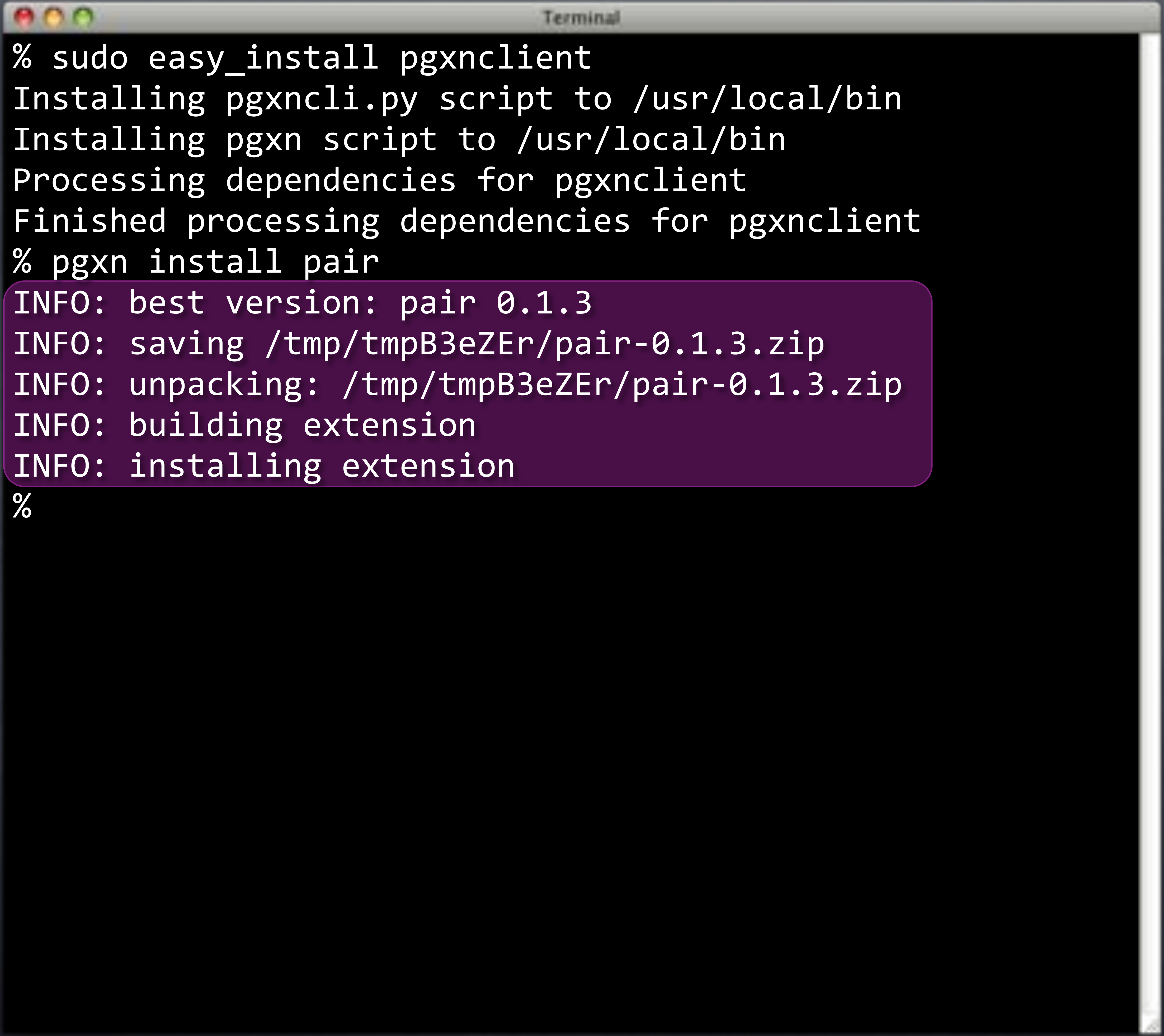
```
%
```


Terminal

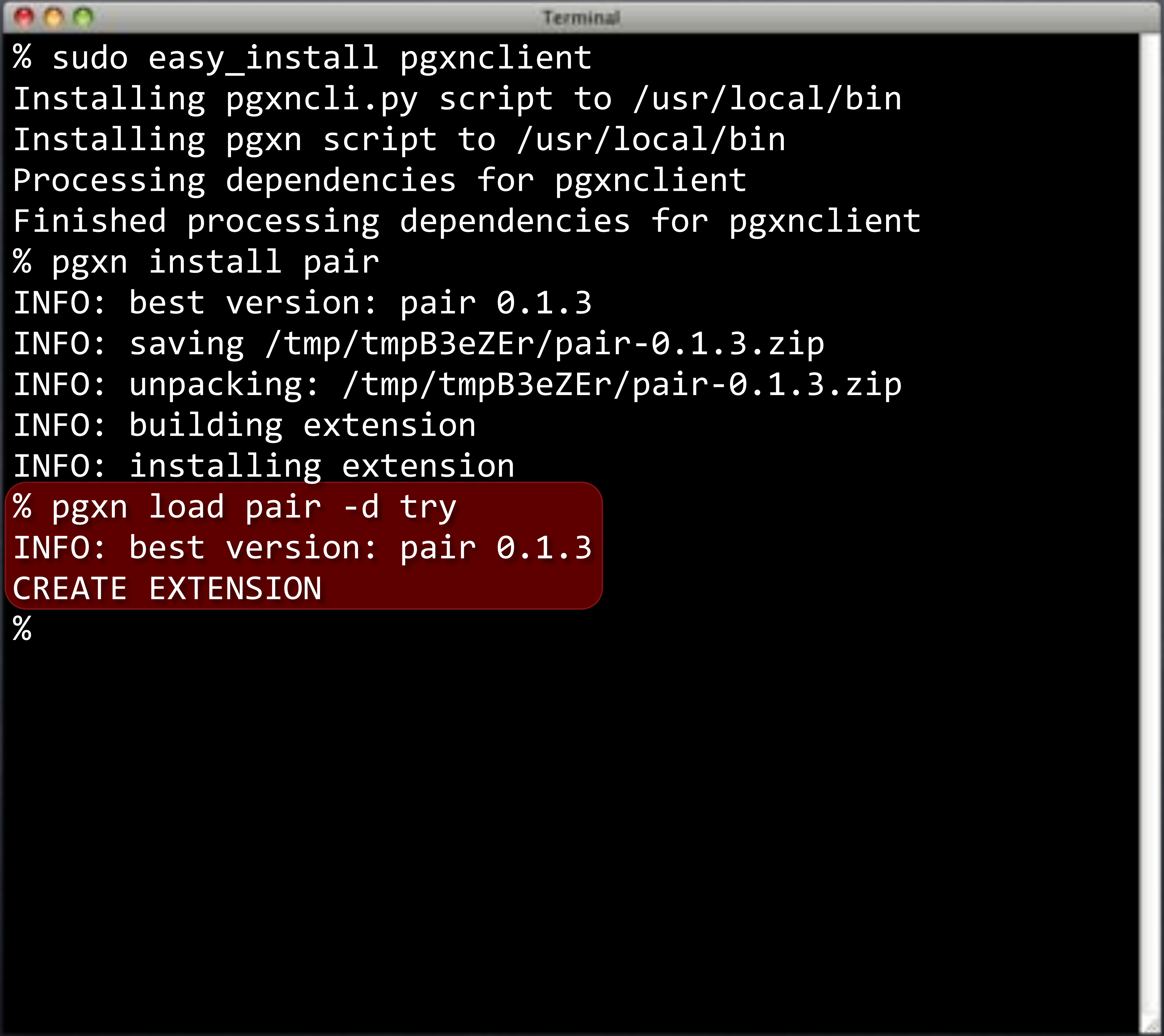
```
% sudo easy_install pgxnclient
Installing pgxncli.py script to /usr/local/bin
Installing pgxn script to /usr/local/bin
Processing dependencies for pgxnclient
Finished processing dependencies for pgxnclient
%
```



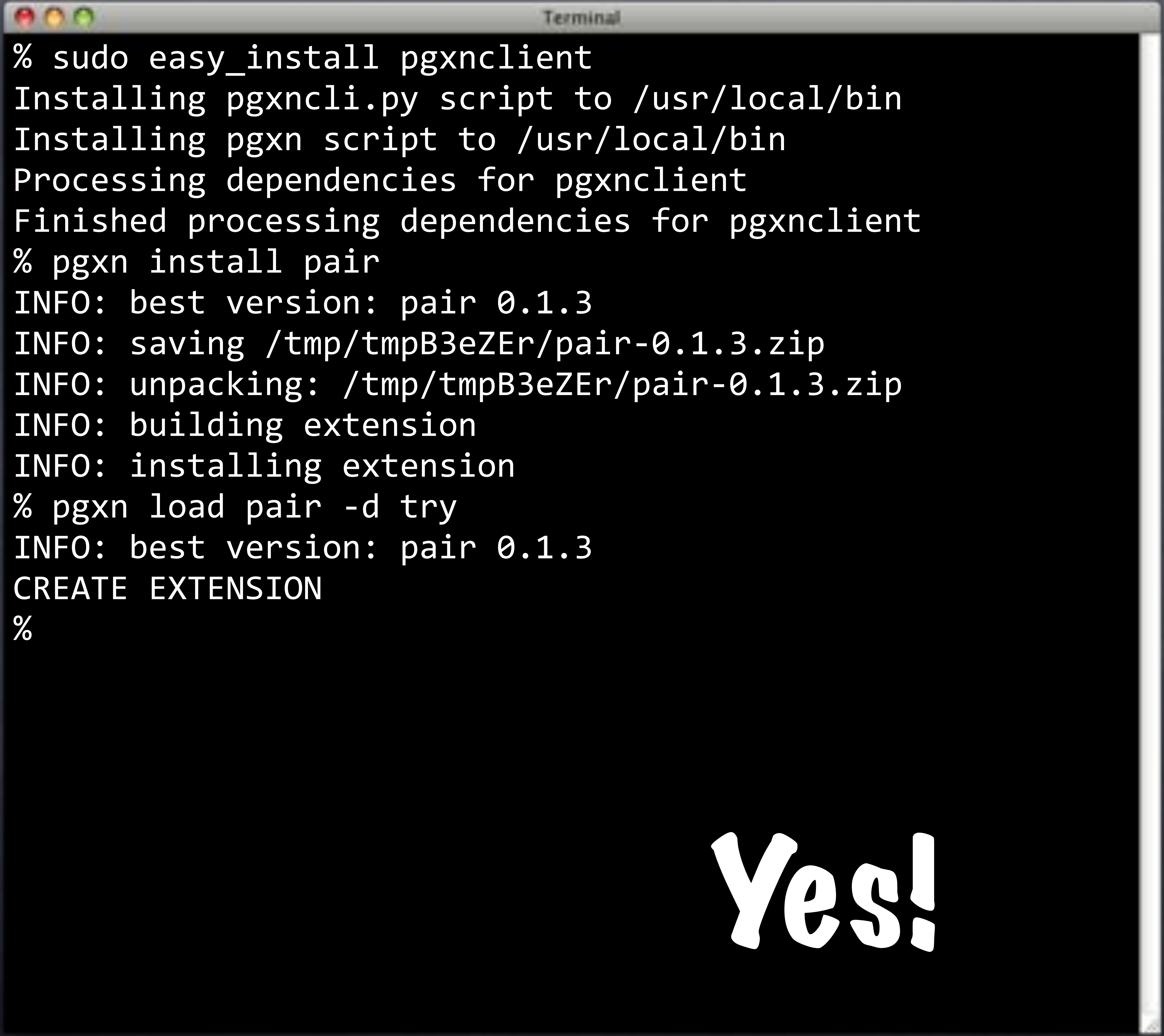
```
% sudo easy_install pgxnclient
Installing pgxncli.py script to /usr/local/bin
Installing pgxn script to /usr/local/bin
Processing dependencies for pgxnclient
Finished processing dependencies for pgxnclient
% pgxn install pair
INFO: best version: pair 0.1.3
INFO: saving /tmp/tmpB3eZEr/pair-0.1.3.zip
INFO: unpacking: /tmp/tmpB3eZEr/pair-0.1.3.zip
INFO: building extension
INFO: installing extension
%
```



```
% sudo easy_install pgxnclient
Installing pgxncli.py script to /usr/local/bin
Installing pgxn script to /usr/local/bin
Processing dependencies for pgxnclient
Finished processing dependencies for pgxnclient
% pgxn install pair
INFO: best version: pair 0.1.3
INFO: saving /tmp/tmpB3eZEr/pair-0.1.3.zip
INFO: unpacking: /tmp/tmpB3eZEr/pair-0.1.3.zip
INFO: building extension
INFO: installing extension
%
```



```
% sudo easy_install pgxnclient
Installing pgxncli.py script to /usr/local/bin
Installing pgxn script to /usr/local/bin
Processing dependencies for pgxnclient
Finished processing dependencies for pgxnclient
% pgxn install pair
INFO: best version: pair 0.1.3
INFO: saving /tmp/tmpB3eZEr/pair-0.1.3.zip
INFO: unpacking: /tmp/tmpB3eZEr/pair-0.1.3.zip
INFO: building extension
INFO: installing extension
% pgxn load pair -d try
INFO: best version: pair 0.1.3
CREATE EXTENSION
%
```



```
% sudo easy_install pgxnclient
Installing pgxncli.py script to /usr/local/bin
Installing pgxn script to /usr/local/bin
Processing dependencies for pgxnclient
Finished processing dependencies for pgxnclient
% pgxn install pair
INFO: best version: pair 0.1.3
INFO: saving /tmp/tmpB3eZEr/pair-0.1.3.zip
INFO: unpacking: /tmp/tmpB3eZEr/pair-0.1.3.zip
INFO: building extension
INFO: installing extension
% pgxn load pair -d try
INFO: best version: pair 0.1.3
CREATE EXTENSION
%
```

Yes!

Client Plans

Client Plans

- Git-like dispatch

Client Plans

- Git-like dispatch
  - pgxn foo => pgxn-foo

Client Plans

- Git-like dispatch
  - pgxn foo => pgxn-foo
- pgxn_utils adapted

Client Plans

- Git-like dispatch
    - pgxn foo => pgxn-foo
- pgxn_utils adapted
    - pgxn skeleton semver

Client Plans

- Git-like dispatch
  - pgxn foo => pgxn-foo
- pgxn_utils adapted
  - pgxn skeleton semver
- META.json validator under development

Client Plans

- Git-like dispatch
  - pgxn foo => pgxn-foo
- pgxn_utils adapted
  - pgxn skeleton semver
- META.json validator under development
  - pgxn validate-meta

Client Plans

- Git-like dispatch
  - pgxn foo => pgxn-foo
- pgxn_utils adapted
  - pgxn skeleton semver
- META.json validator under development
  - pgxn validate-meta
- New dispatcher next week

Client Plans

- Git-like dispatch
  - pgxn foo => pgxn-foo
- pgxn_utils adapted
  - pgxn skeleton semver
- META.json validator under development
  - pgxn validate-meta
- New dispatcher next week
- Write some utilities!

Thank you
Daniele Varrazzo

Resources

Resources

- Site: http://pgxn.org/

Resources

- Site: http://pgxn.org/
- Twitter: http://twitter.com/pgxn

Resources

- Site: http://pgxn.org/
- Twitter: http://twitter.com/pgxn
- Blog: http://blog.pgxn.org/

Resources

- Site: http://pgxn.org/
- Twitter: http://twitter.com/pgxn
- Blog: http://blog.pgxn.org/
- Release: http://manager.pgxn.org/

Resources

- Site: http://pgxn.org/
- Twitter: http://twitter.com/pgxn
- Blog: http://blog.pgxn.org/
- Release: http://manager.pgxn.org/
- API: http://github.com/pgxn/pgxn-api/wiki/

Resources

- Site: http://pgxn.org/
- Twitter: http://twitter.com/pgxn
- Blog: http://blog.pgxn.org/
- Release: http://manager.pgxn.org/
- API: http://github.com/pgxn/pgxn-api/wiki/
- Group: http://groups.google.com/group/pgxn-users

Resources

- Site: http://pgxn.org/
- Twitter: http://twitter.com/pgxn
- Blog: http://blog.pgxn.org/
- Release: http://manager.pgxn.org/
- API: http://github.com/pgxn/pgxn-api/wiki/
- Group: http://groups.google.com/group/pgxn-users
- Donate: http://fundraising.pgxn.org/

Resources

- Site: http://pgxn.org/
- Twitter: http://twitter.com/pgxn
- Blog: http://blog.pgxn.org/
- Release: http://manager.pgxn.org/
- API: http://github.com/pgxn/pgxn-api/wiki/
- Group: http://groups.google.com/group/pgxn-users
- Donate: http://fundraising.pgxn.org/

Thank you.

Releasing Extensions on PGXN

David E. Wheeler
PostgreSQL Experts, Inc.