



Understanding Free Software & Open Source Licensing

May 30
eLiberatica 2008
Bucharest, Romania



Warnings and Disclosure



Warnings and Disclosure

1. I am not a lawyer



Warnings and Disclosure

1. I am not a lawyer
2. Law is not code



Warnings and Disclosure

1. I am not a lawyer
2. Law is not code
3. Solely my opinions

1

Foo

Associates



—
Foo
Associates
—



Free Software

Foundation License

Compliance Lab

Volunteer since 2004

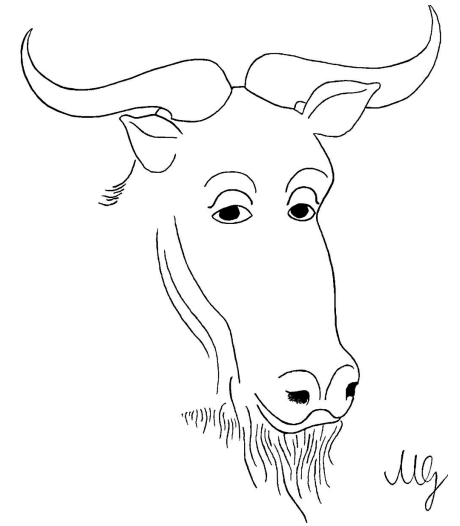
—
Foo
Associates
—





Foo

Associates



Mozilla Foundation

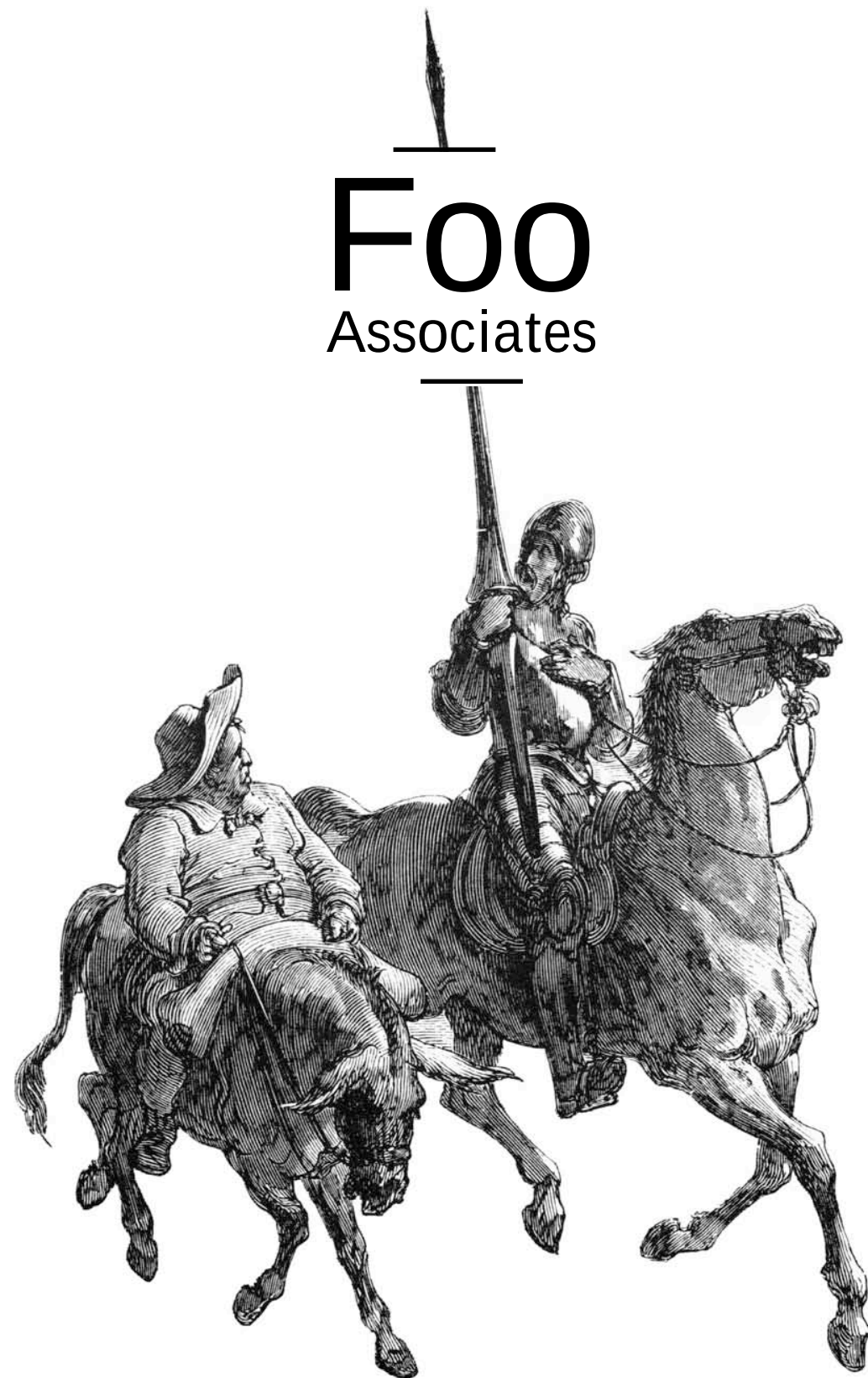
Paid Staff Member

Since 2005

Foo
Associates



mozilla
FOUNDATION

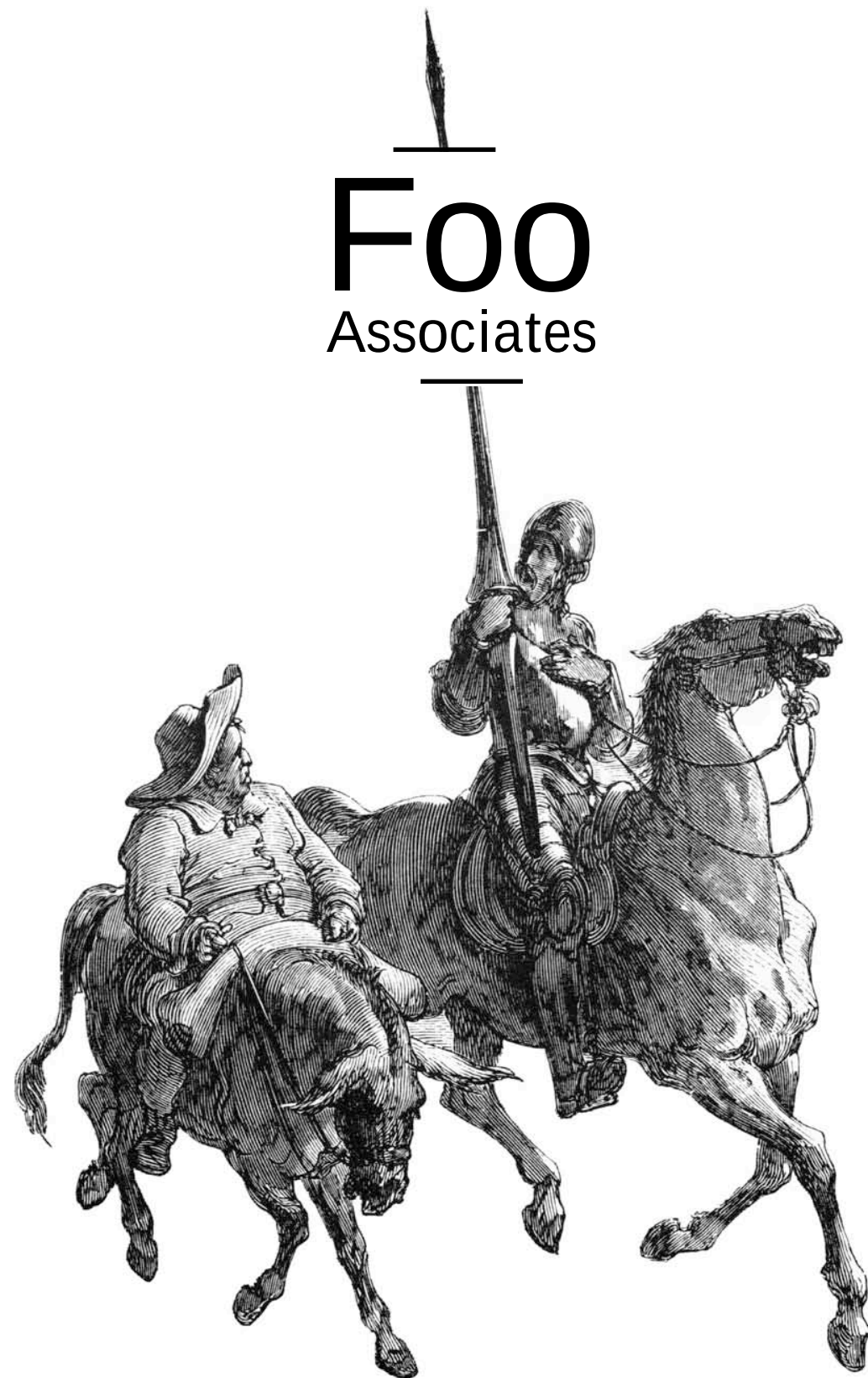


Foo
Associates



mozilla
FOUNDATION

Open Source Initiative
Board Observer since
January



Foo
Associates



mozilla
FOUNDATION



open source



Mozilla Manifesto

<http://mozilla.org/about/mozilla-manifesto.html>

Maximize the global public and individual benefits of the Internet.



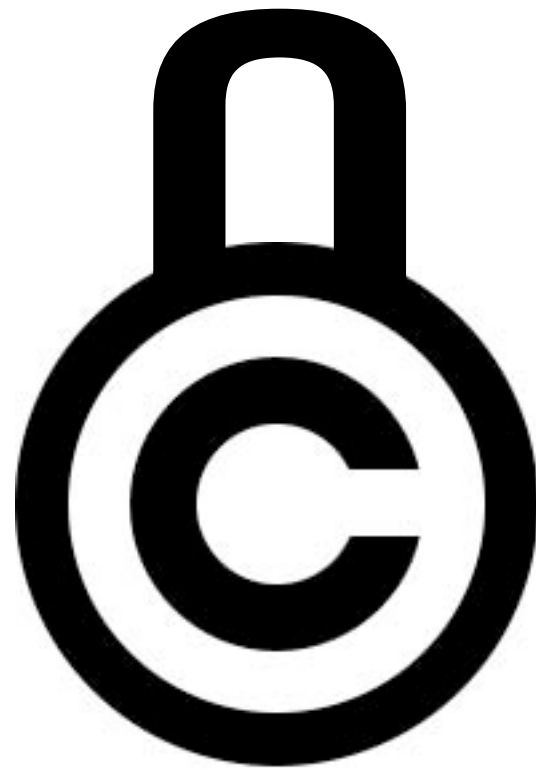
Foundations

The Roots of Free
Software & Open
Source Licensing



Copyright

Automatically
granted to creators
by the state



Copyright

Automatically
granted to creators
by the state



The Big Idea?

You own your
software.



If you can't open it...

you don't own it.



Proprietary Software

Your software
owns you.



Four Freedoms



Four Freedoms

1. Run



Four Freedoms

1. Run
2. Study



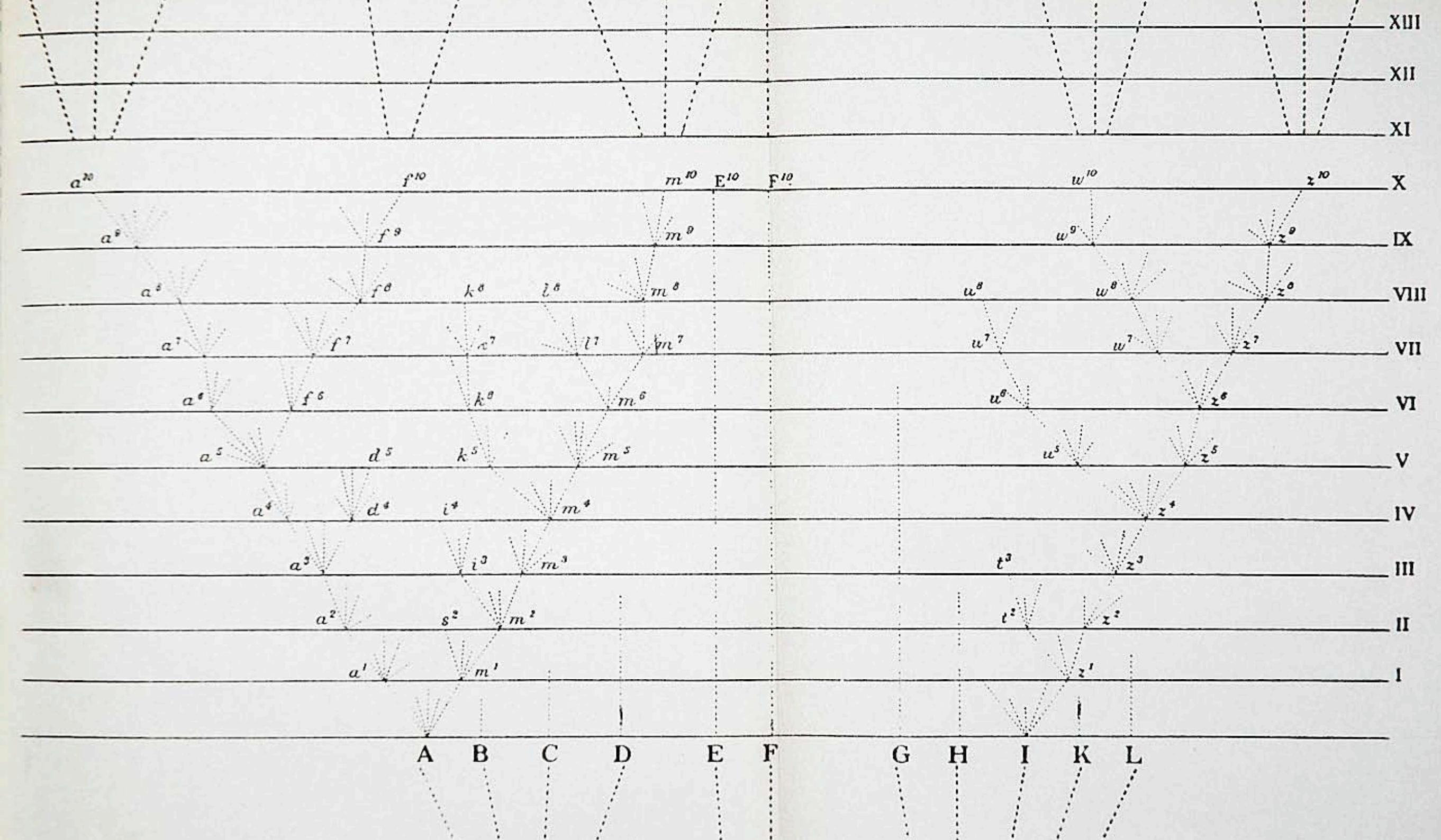
Four Freedoms

1. Run
2. Study
3. Modify



Four Freedoms

1. Run
2. Study
3. Modify
4. Share



License Types



Permissive

The sky is the limit

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



BSD License



BSD License

Keep license notices



BSD License

Keep license notices
No warranty



BSD License

Keep license notices
No warranty
Don't use our name



Mac OS X



Mac OS X

BSD



Mac OS X

BSD
Mach



Mac OS X

BSD
Mach
NeXTStep



Copyleft

Share ...
but Share-alike

Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foobassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFIGFILE = "ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```


Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foobassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFIGFILE = "ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

```
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFIGFILE = "ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
```


Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foobassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFIGFILE = "ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

```
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFIGFILE = "ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/";
```


Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foobassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

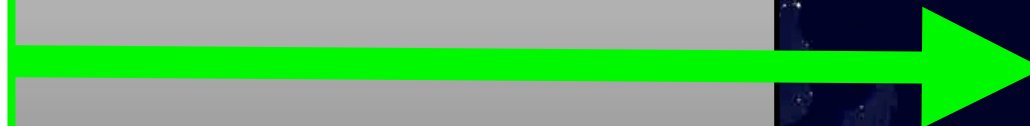
our $CONFIGFILE = "ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

```
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFIGFILE = "ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/";
```



Legal Entity

Copyright

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foaassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

```
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;
```

Copyright

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foaassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding->get_encoding();
};
$encoding ||= "utf-8";

binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();
    $args{owner} = "me";
```


Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";
binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();

    $args{owner} = "me";

    do_login() or die("Bad username/password -- edit $CONFFILE and try
again.");

    @_ = @ARGV;
    gen_head();
    &list_tasks;
}

sub gen_head {
    print <<EOD
    <style type="text/css">
```

```
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/
nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";
binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();

    $args{owner} = "me";

    do_login() or die("Bad username/password -- edit $CONFFILE and try
again.");

    @_ = @ARGV;
    gen_head();
    &list_tasks;
}

sub gen_head {
    print <<EOD
    <style type="text/css">
```


Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

```
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/
nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";

binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();

    $args{owner} = "me";

    do_login() or die("Bad username/password -- edit $CONFFILE and try
again.");

    @_ = @ARGV;
    gen_head();
    &list_tasks;
}

sub gen_head {
    print <<EOD
    <style type="text/css">
```

Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foaassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foaassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";

binmode STDOUT, ":",encoding($encoding);

main();

sub main {
    setup_config();

    $args{owner} = "me";

    do_login() or die("Bad username/password -- edit
$CONFFILE and try again.");

    @_ = @ARGV;

    gen_head();
    &list_tasks;
}

sub gen_head {
    print <<EOS
<style type="text/css">
```

```
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/
nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";

binmode STDOUT, ":",encoding($encoding);

main();

sub main {
    setup_config();

    $args{owner} = "me";

    do_login() or die("Bad username/password -- edit $CONFFILE and try
again.");

    @_ = @ARGV;

    gen_head();
    &list_tasks;
}

sub gen_head {
    print <<EOS
<style type="text/css">
```


Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foaassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foaassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";

binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();
    $args{owner} = "me";

    do_login() or die("Bad username/password -- edit
$CONFFILE and try again.");

    @_ = @ARGV;
    gen_head();
    &list_tasks;
}

sub gen_head {
    print <<EOS
<style type="text/css">
```

```
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/
nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";

binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();
    $args{owner} = "me";

    do_login() or die("Bad username/password -- edit $CONFFILE and try
again.");

    @_ = @ARGV;
    gen_head();
    &list_tasks;
}

sub gen_head {
    print <<EOS
<style type="text/css">
```

Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foaassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foaassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";

binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();

    $args{owner} = "me";

    do_login() or die("Bad username/password -- edit
$CONFFILE and try again.");

    @_ = @ARGV;

    gen_head();
    &list_tasks;
}

sub gen_head {
    print <<EOD
<style type="text/css">
```

```
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/
nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";

binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();

    $args{owner} = "me";

    do_login() or die("Bad username/password -- edit $CONFFILE and try
again.");

    @_ = @ARGV;

    gen_head();
    &list_tasks;
}

sub gen_head {
    print <<EOD
<style type="text/css">
```

Incompatible

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foaassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```


Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foobassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foobassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Incompatible

```
=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foobassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
```


Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

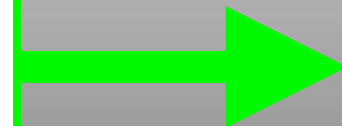
=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```



Incompatible

```
=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@foassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
```



Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Incompatible

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";
binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();

    $args{owner} = "me";
```

Incompatible

```
=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
```


Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Incompatible

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";
binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();

    $args{owner} = "me";
```

Incompatible

```
=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
```



Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Incompatible

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";
binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();

    $args{owner} = "me";
```

Incompatible

```
=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

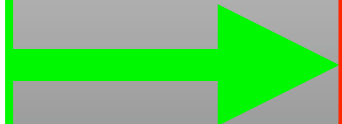
use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
```



Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Incompatible

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";
binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();

    $args{owner} = "me";
```

Incompatible

```
=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
```

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Incompatible

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";
binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();

    $args{owner} = "me";
```

Incompatible

```
=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
```

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```


Legal Entity

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Incompatible

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
    $encoding = Term::Encoding::get_encoding();
};
$encoding ||= "utf-8";
binmode STDOUT, ":encoding($encoding)";

main();

sub main {
    setup_config();

    $args{owner} = "me";
```

Incompatible

```
=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
```

Copyleft

```
#!/usr/bin/env perl
use strict;
use warnings;

=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;
```

Incompatible

```
=head1 DESCRIPTION
Hacked version of Best Practical's Hiveminder todo.pl script
Generates reports from Hiveminder - meant to be safe-ish to
run as a cgi script
Hacked up by zak@fooassociates.com
=cut

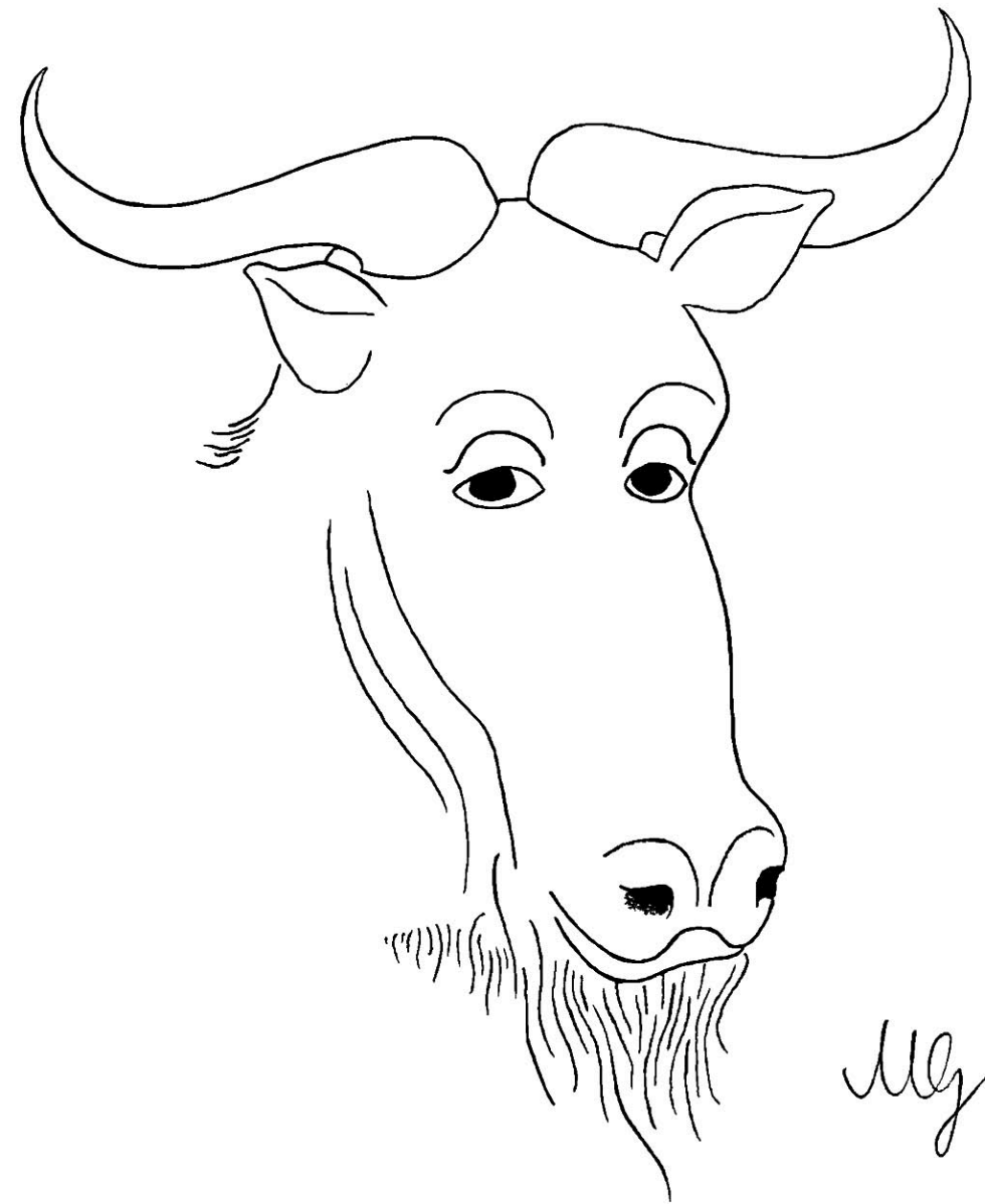
use Encode ();
use YAML ();
use LWP::UserAgent;
use Number::RecordLocator;
use Email::Address;
use Fcntl qw(:mode);
use File::Temp;

our $CONFFILE = "$ENV{HOME}/.hiveminder";
our %config = ();
our $ua = LWP::UserAgent->new;
our $locator = Number::RecordLocator->new();
our $default_query = "not/complete/starts/before/tomorrow/
accepted/but_first/nothing";
our %args;

$ua->cookie_jar({});

# Load the user's proxy settings from %ENV
$ua->env_proxy;

my $encoding;
eval {
    require Term::Encoding;
```



GNU General Public License

Private Freedom.
Public Responsibility.



Google

Proprietary services
built with GPL'd
software.



MySQL

Dual-licensing:
Share-alike or pay.

Time?

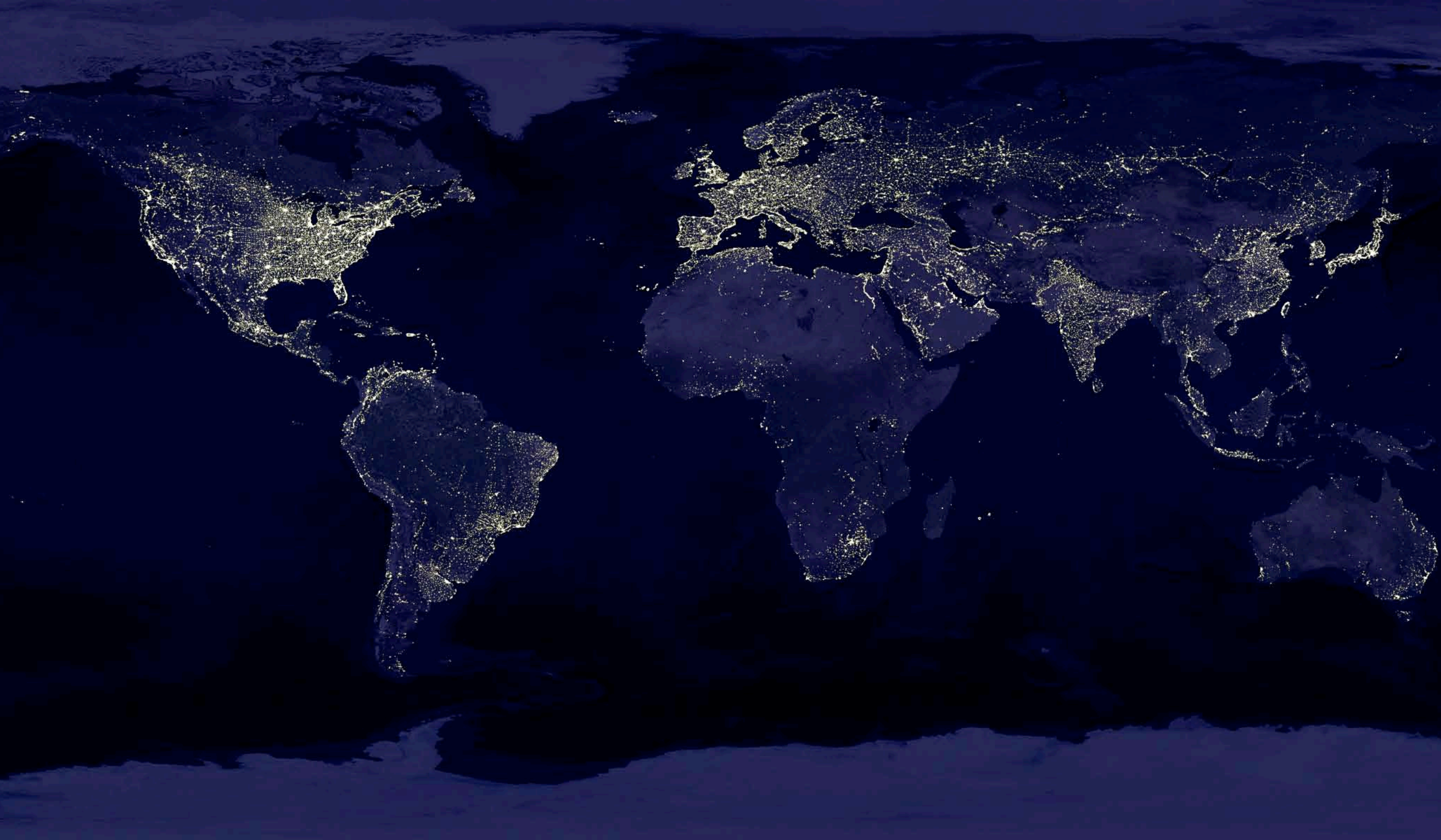
I don't want to be late!





1984 called

It wants its software market
back.



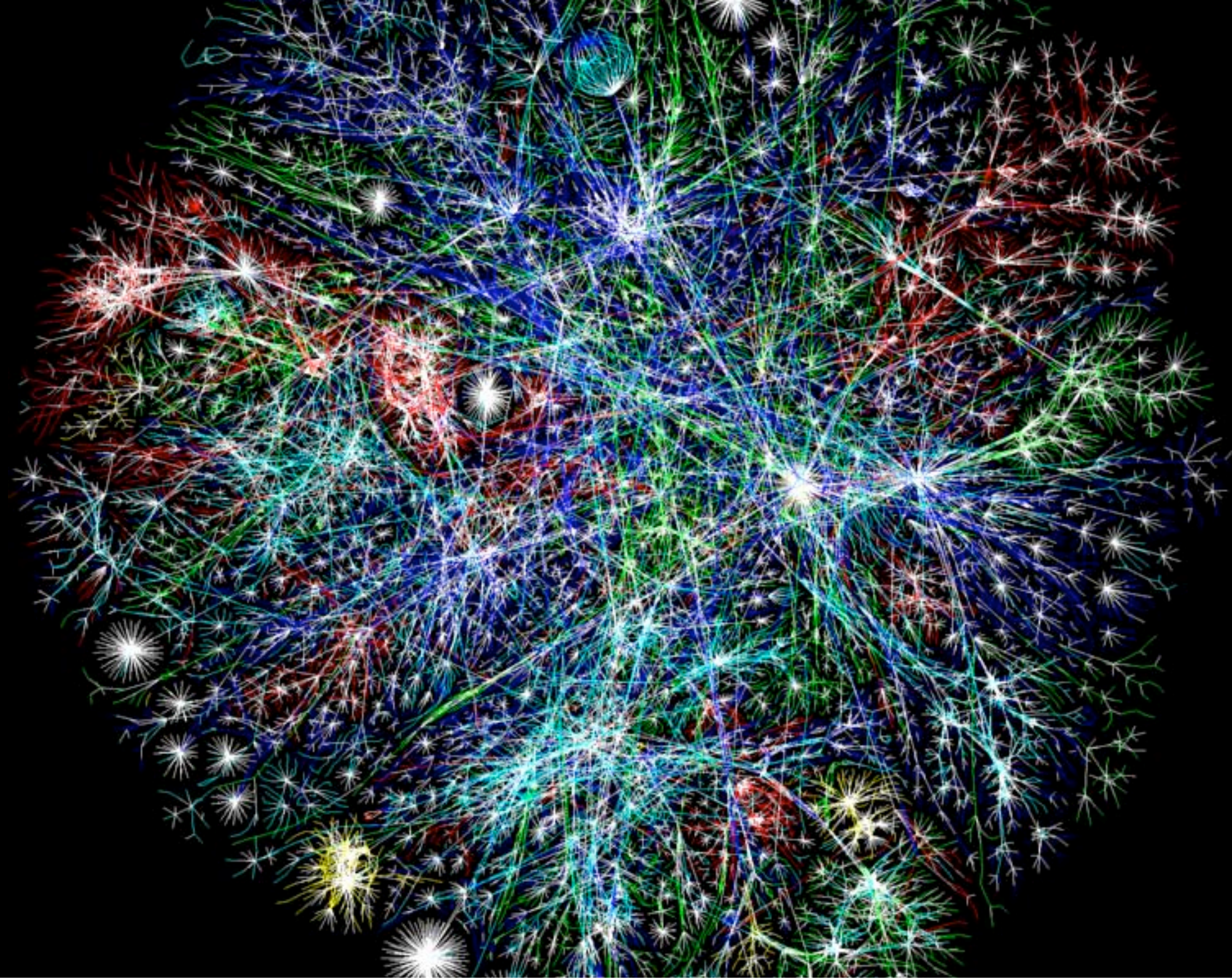
The Internet

1969



PC Revolution

1980's



The World Wide Web

1991



Questions?

zak@greant.com



License; Disclaimer of Warranty and Liability

This Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

THIS WORK IS OFFERED AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL I BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT THE USE OF THE WORK, EVEN IF I HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.