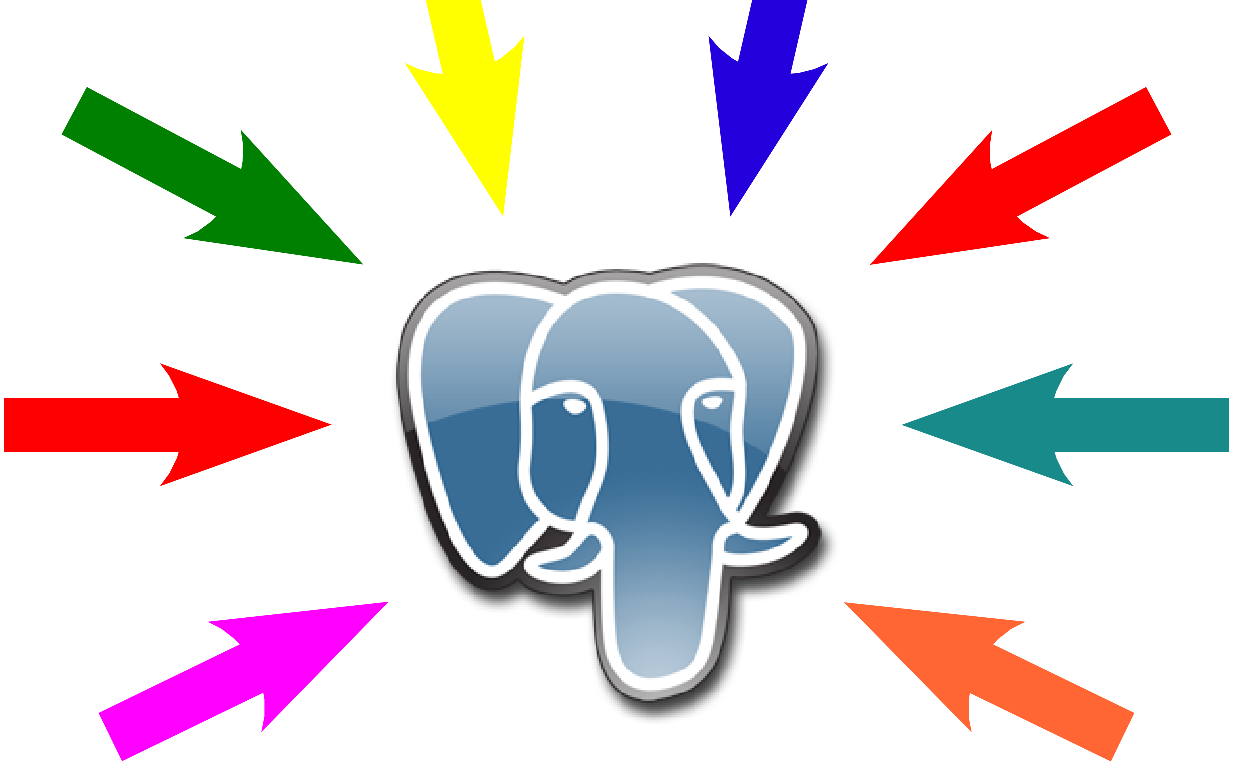
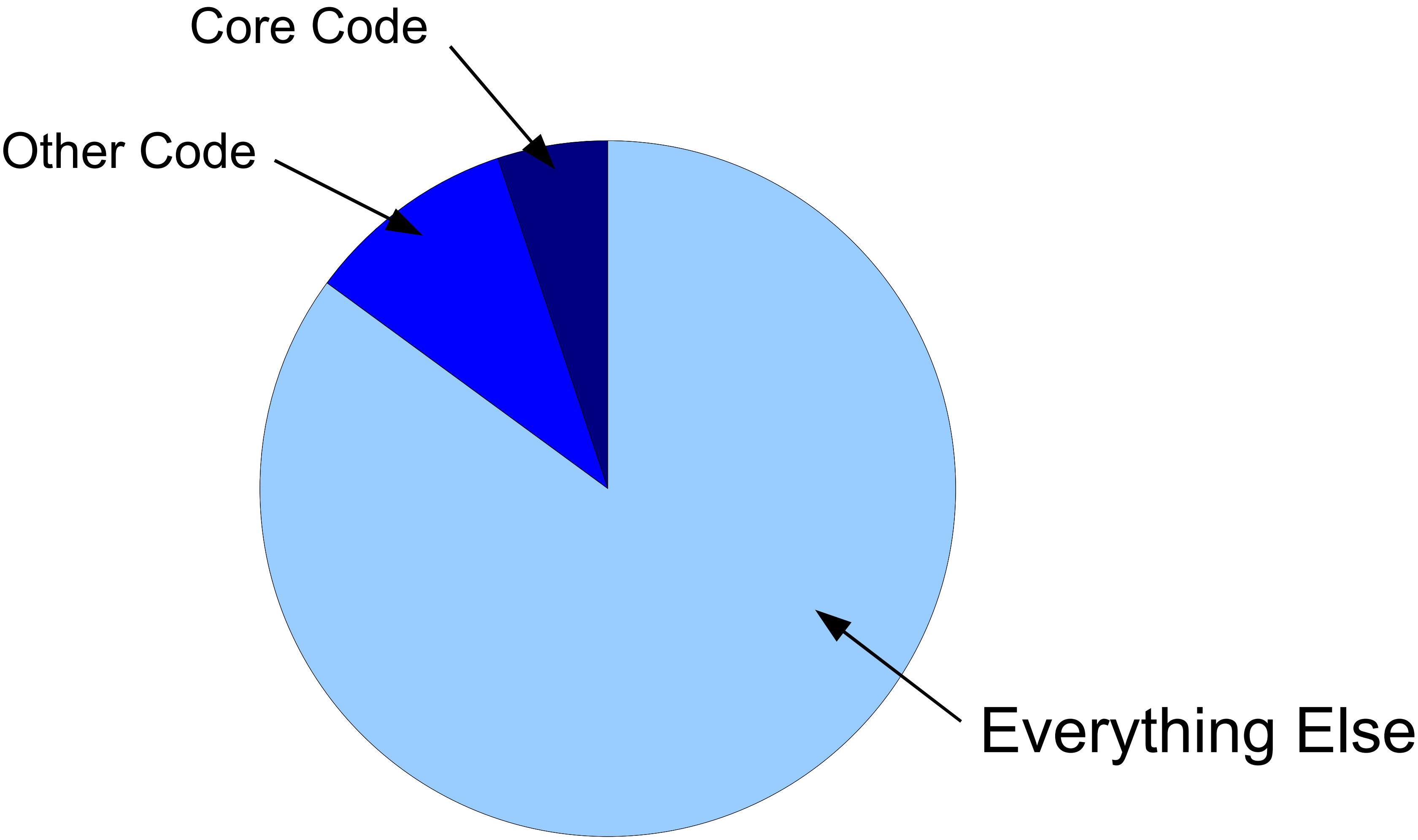


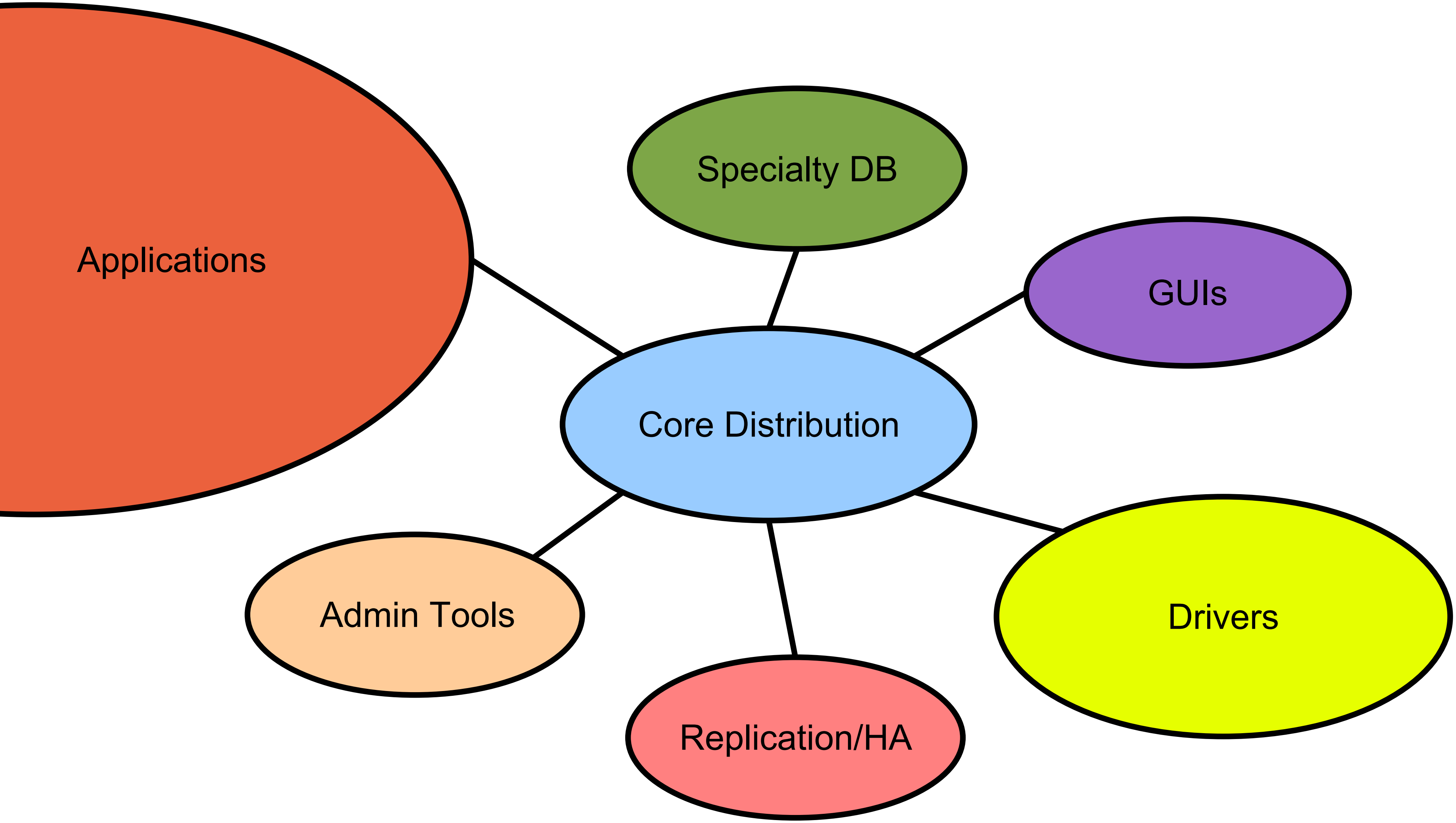
How to contribute to PostgreSQL

or, 50 Ways To Love Your Project

Contribution Pie



PostgreSQL Constellation



The 50 Ways

Code

1. Contribute Code to the Core

(more on this later)

2. Review Code

we need this more than new code!

3. Squash Bugs

survey the mailing lists,
and squash away
pgsql-bugs and all others

4. Contribute to an Add-In

find something on pgFoundry or /contrib you care about

5. Become a Driver Maintainer

DBD::Pg, PDO, JDBC, ODBC, libpqxx, psychoPg2, ecpg, etc.

Non-Code

Packaging

6. Create/Maintain an OS Package

   SuSE, Solaris, OpenBSD, AIX, Windows

7. Create/Maintain Packages for Add-ins

   Or just help OS package maintainers pick add-ins.

Applications

8. Port a popular application to PostgreSQL

    or just help maintain one

9. Create a new PostgreSQL application

    show people how to use cool PG features

9. Improve an existing PostgreSQL application

    make it #1 in its area and people will use the DB

10. Get your company to open-source an App

    have they done something cool with Postgres?

Translation

do you know a non-English language? PostgreSQL needs your help!

11. Translate the main docs
12. Translate add-in docs
13. Translate the website
14. Translate questions by non-English speakers

    when they need help not available in their language

PostgreSQL Needs Writers

15. Write main docs

join pgsql-docs -- learn SGML!

16. Write add-in docs

some have no docs at all

17. Write a HOWTO

solved a problem on your own? show others how

developer wiki, TechDocs, O'Reilly, etc.

18. Write a porting guide

MySQL, Oracle, Informix, SQL Server

19. Write promotional content

flyers, website info, press releases

20. Write a case study

we have many companies waiting to do case studies

pgsql-advocacy

20. Write a magazine article

php|arch, LinuxMag, LWN, Dr. Dobb's, Oreillynet, etc.

21. Write a book

APress, NoStarch, SAMS, etc.

education books needed most

22. Just Blog It

We ♥ Designers!

Are you talented with graphics? We could use your help. Heck, I could use your help.

23. Contribute to the website design
    pgsql-www list

24. Design advocacy materials
    flyers, brochures: pgsql-advocacy list

25. Create logos & Graphics
    t-shirts and banners too

User Groups

local user groups help grow the PostgreSQL community

26. Lead a PostgreSQL User Group

    arrange monthly meetings

27. Present at someone else's UG

    Linux UG, sysadmins, Perl, Rails, MSDN, etc.

Conferences

28. Speak at a technology conference
    hundreds every year
    keep track & submit a paper
    ask Funds Group for help with travel

29. Work in PostgreSQL booth
    sign-up sheet for the OSCON booth
    in the back of the room.

30. Organize a PostgreSQL mini-conference
    like this one!

Infrastructure

Sysadmins wanted!

31. Help maintain & improve the postgresql.org infrastructure

    hundreds of volunteer-hours needed

32. Donate a server or virtual server

33. Organize a PostgreSQL mini-conference

    like this one!

At Work

34. Persuade your company to do more projects with PostgreSQL

35. Get your company to provide a testimonial
    and get on the postgresql.org home page

36. Convince your company to do a case study
    we need more highly technical ones

Help other users

If you got help as a newbie, you can return the favor.

37. On mailing lists *(-newbie, -SQL, -general)*
38. On IRC
39. On other projects' mailing lists
40. In other languages

You can never have too much

Testing

41. Test new patches

try them out and see what breaks

42. Run a pgbuildfarm instance

www.pgbuildfarm.org

43. Help improve the pgbuildfarm

we want it to test more things

44. Do performance tests

compare versions & settings

45. Check standards/spec compliance

SQL, ODBC, JDBC, APIs

46. Test integration

does the new version work with all drivers & tools?

47. Test applications

**try porting yours to the beta,
or popular OSS applications**

Education

48. Get your school / alma mater to include PostgreSQL in the cirriculum

49. Participate in Summer of Code
    student, mentor, recruiter, admin

50. Money

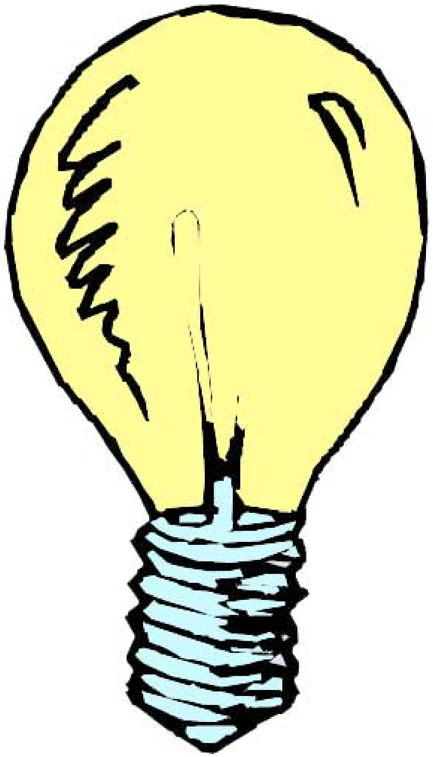
donations gladly accepted through SPI

So, you still wanna write some code?

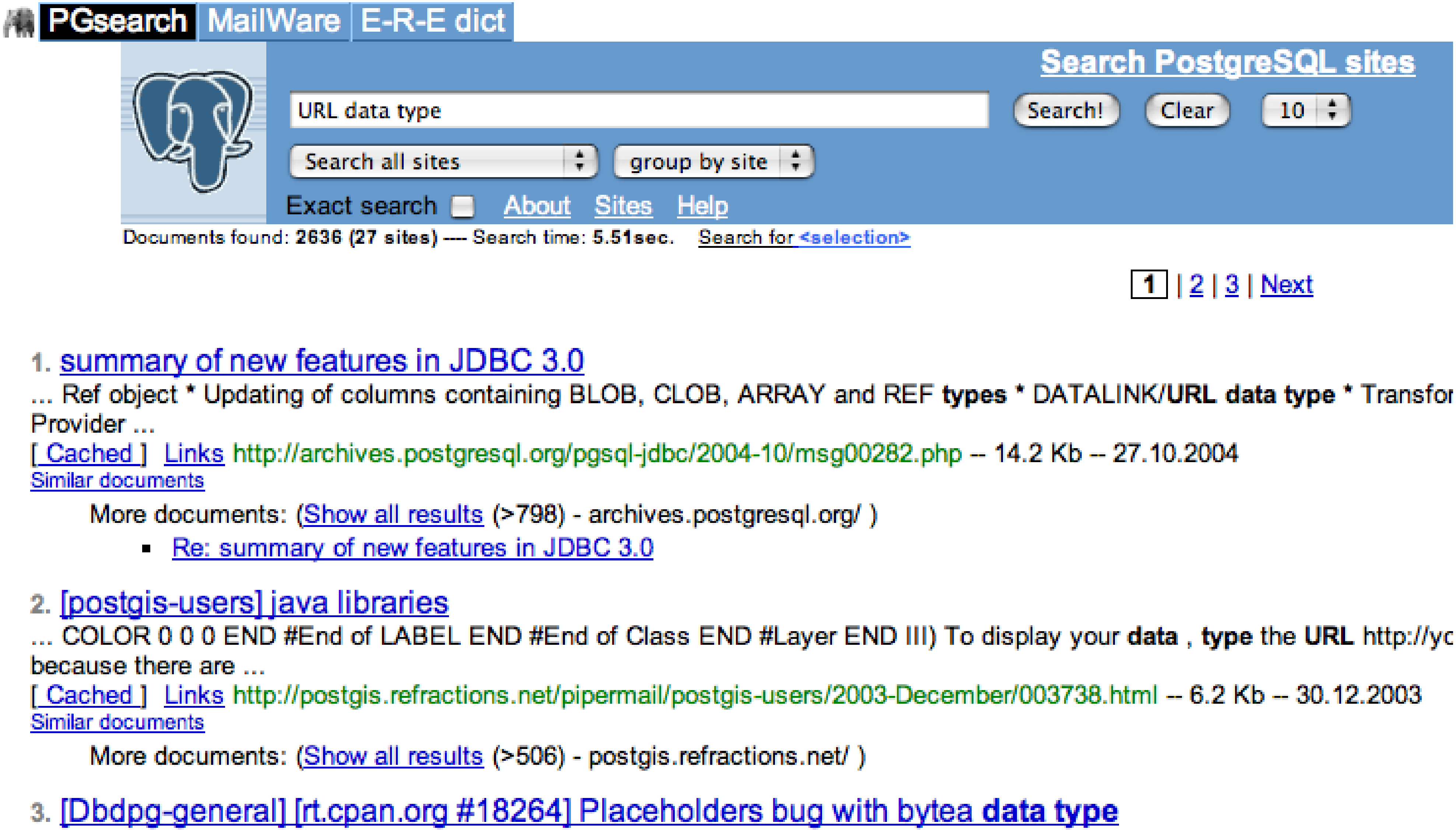
Life of a Patch

a contribution success story in examples

You have an idea



You research if we've discussed it before



URL data type

Search all sites

group by site

Exact search □

Documents found: **2636 (27 sites)** ---- Search time: **5.51sec.** Search for <selection>

1 | 2 | 3 | Next

1. summary of new features in JDBC 3.0
... Ref object * Updating of columns containing BLOB, CLOB, ARRAY and REF **types** * DATALINK/**URL data type** * Transfor
Provider ...
[Cached] Links http://archives.postgresql.org/pgsql-jdbc/2004-10/msg00282.php -- 14.2 Kb -- 27.10.2004
Similar documents

More documents: (Show all results (>798) - archives.postgresql.org/)

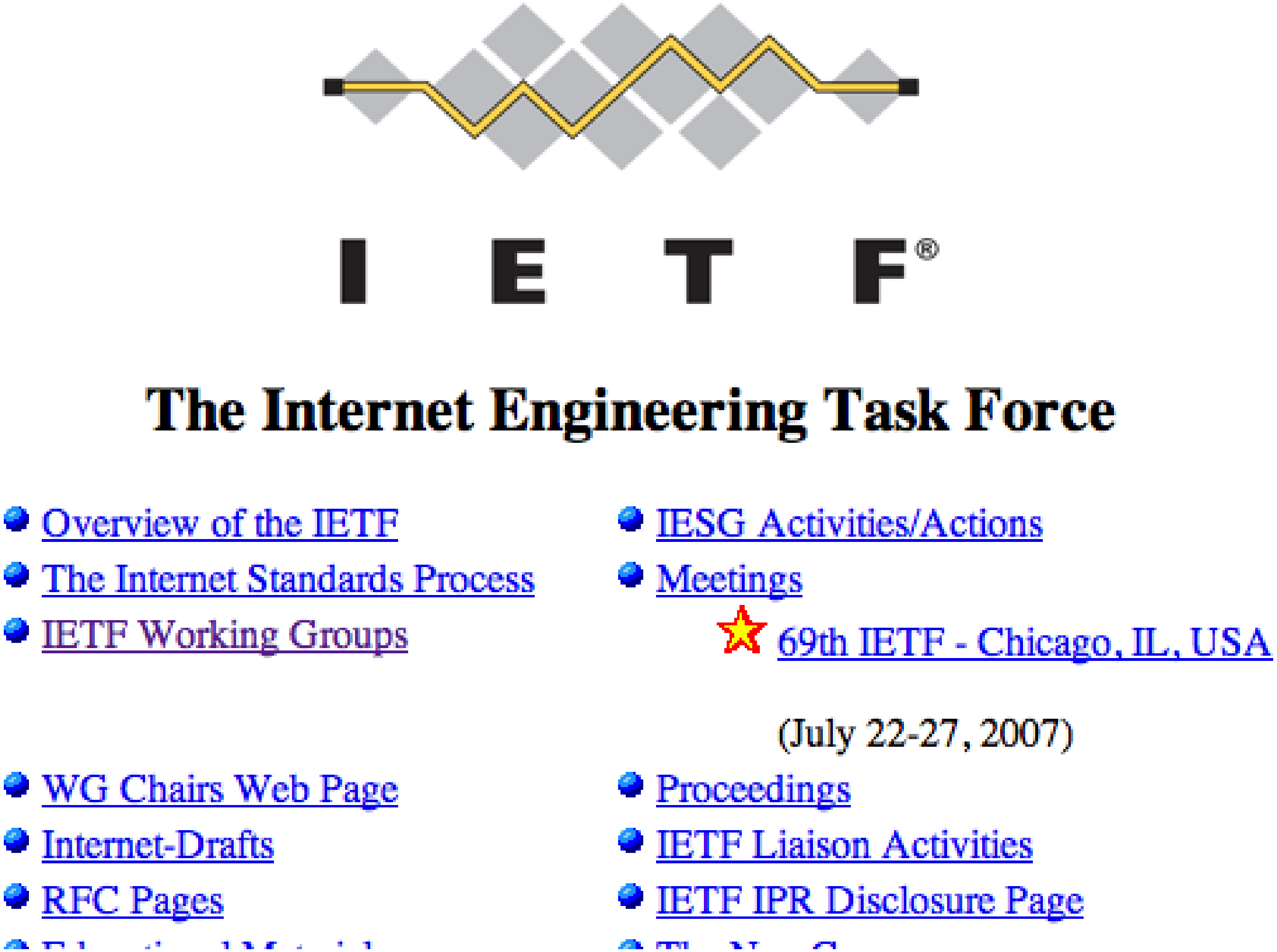
- Re: summary of new features in JDBC 3.0

2. [postgis-users] java libraries
... COLOR 0 0 0 END #End of LABEL END #End of Class END #Layer END III) To display your **data** , **type** the **URL** http://yc
because there are ...
[Cached] Links http://postgis.refractions.net/pipermail/postgis-users/2003-December/003738.html -- 6.2 Kb -- 30.12.2003
Similar documents

More documents: (Show all results (>506) - postgis.refractions.net/)

3. [Dbdpg-general] [rt.cpan.org #18264] Placeholders bug with bytea **data type**

You research applicable standards



The Internet Engineering Task Force

- Overview of the IETF
- The Internet Standards Process
- IETF Working Groups
- WG Chairs Web Page
- Internet-Drafts
- RFC Pages
- IESG Activities/Actions
- Meetings
  - 69th IETF - Chicago, IL, USA

    (July 22-27, 2007)
- Proceedings
- IETF Liaison Activities
- IETF IPR Disclosure Page

You survey other users

From: **"Raymond O'Donnell" <rod (at) iol (dot) ie>**
To: pgsql-general (at) postgresql (dot) org
Subject: Male/female
Date: Fri, 08 Dec 2006 15:23:11 -0000

Just wondering.....how do list member represent gender when storing details of people in a database?

I've done it two ways:

* A bool column, with the understanding that true/false represents one gender or the other.

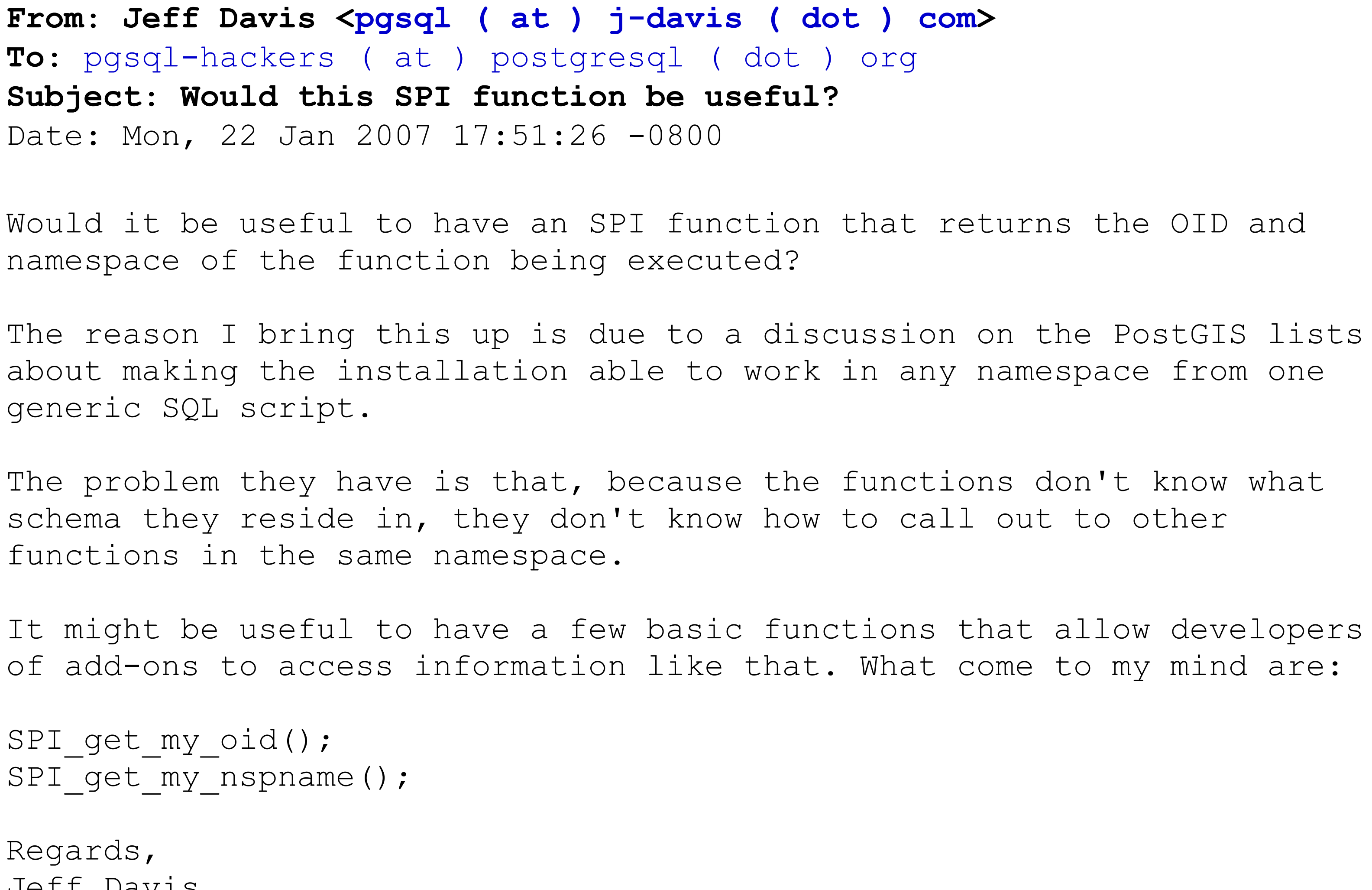
* Create a domain, something like:

```
CREATE DOMAIN gender_domain
  AS character varying(7)
  NOT NULL
   CONSTRAINT gender_domain_check CHECK ((((VALUE)::text =
'male'::text) OR ((VALUE)::text = 'Female'::text)))
```

I personally prefer the second, as it's self-documenting...is there any other/better way of doing it?

You propose your idea on pgsql-hackers

(or other developer list)



From: Jeff Davis <pgsql (at) j-davis (dot) com>
To: pgsql-hackers (at) postgresql (dot) org
Subject: Would this SPI function be useful?
Date: Mon, 22 Jan 2007 17:51:26 -0800

Would it be useful to have an SPI function that returns the OID and
namespace of the function being executed?

The reason I bring this up is due to a discussion on the PostGIS lists
about making the installation able to work in any namespace from one
generic SQL script.

The problem they have is that, because the functions don't know what
schema they reside in, they don't know how to call out to other
functions in the same namespace.

It might be useful to have a few basic functions that allow developers
of add-ons to access information like that. What come to my mind are:

SPI_get_my_oid();
SPI_get_my_nspname();

Regards,
Jeff Davis

Tom Lane says your idea sucks

From: Tom Lane <tgl (at) sss (dot) pgh (dot) pa (dot) us>
To: Newbie hacker
Subject: Re: New data type idea
Date: Wed, 06 Jun 2007 19:41:26 -0400

Search the archives of this list on “URL data type” and you'll see a host of reasons why it's a really bad idea. You should think about these things more before you post them. Besides, who needs this?

            regards, tom lane

No, not really

From: Tom Lane <tgl (at) sss (dot) pgh (dot) pa (dot) us>
To: Newbie hacker
Subject: Re: New data type idea
Date: Wed, 06 Jun 2007 19:41:26 -0400

I'm not sure that you could apply the complex rules for URLs to a data type constraint without it being a huge performance drag. Did you have something in mind for this?

regards, tom lane

If you're lucky, you get a lot of feedback

From: **Peter Eisentraut <peter_e (at) gmx (dot) net>**
To: pgsql-hackers (at) postgresql (dot) org
Subject: **Re: [PATCHES] Enums patch v2**
Date: Tue, 19 Dec 2006 10:58:48 +0100

```
Heikki Linnakangas wrote:
> I'm sorry I missed the original discussions, but I have to ask: Why
> do we want enums in core? The only potential advantage I can see over
> using a look-up table and FK references is performance.

The difference is that foreign-key-referenced data is part of your data
whereas enums would be part of the type system used to model the data.
```

From: **Andrew Sullivan <ajs (at) crankycanuck (dot) ca>**
To: **Andrew Dunstan <andrew (at) dunslane (dot) net>**
Subject: **Re: Idea for new data type**
Date: Tue, 6 Feb 2007 11:46:24 -0500

```
On Tue, Feb 06, 2007 at 11:43:24AM -0500, Andrew Dunstan wrote:
> Andrew Sullivan wrote:
> >
> >As a data point, some time ago (7.2 days) I used to do this as a
> >matter of completeness, and never had a collision.
>
> The point I at least have been trying to make is that extensions
```

From: **Neil Conway <neilc (at) samurai (dot) com>**
To: **Martijn van Oosterhout <kleptog (at) svana (dot) org>**
Subject: **Re: Asynchronous I/O Support**
Date: Sun, 15 Oct 2006 14:26:12 -0400

```
On Sun, 2006-10-15 at 19:56 +0200, Martijn van Oosterhout wrote:
> Sure, I even implemented it once. Didn't get any faster.

Did you just do something akin to s/read/aio_read/ etc., or something
more ambitious? I think that really taking advantage of the ability to
have multiple I/O requests outstanding would take some leg work.

> Maybe one of these days I should work out if my current system supports
> it, and give it another go...
```

Even negative feedback is useful.

If you don't, find out who should be interested.

From: **"Dragan Zubac" <zubacdragan (at) gmail (dot) com>**
To: Andrew Sullivan
Subject: URL Data type
Date: Tue, 26 Sep 2007 22:39:18 +0200

Andrew:

Given the kind of data you work with, would you find a URL data type helpful? Can you check over my idea below?

...

Then you go off and develop a specification

```
CREATE TABLE tabname (
     ...
     ...
  ) PARTITION BY
  HASH(expr)
 | RANGE(expr)
 | LIST(expr)
 [PARTITIONS num_partitions] /* will apply to HASH
only for now*/
 [PARTITION partition_name CHECK(...),
  PARTITION partition_name CHECK(...)
  ...
 ];
```

You post the specification

From: NikhilS <nikkhils (at) gmail (dot) com>
To: pgsql-hackers (at) postgresql (dot) org
Subject: Auto creation of Partitions
Date: Tue, 6 Mar 2007 17:27:50 +0530

Hi,

This is to get feedback to meet the following TODO:
Simplify ability to create partitioned tables
This would allow creation of partitioned tables without requiring creation of rules for INSERT/UPDATE/DELETE, and constraints for rapid partition selection. Options could include range and hash partition selection.

There was some discussion on the pgsql mailing lists, which lead to the above TODO:
http://archives.postgresql.org/pgsql-hackers/2006-09/msg00189.php
http://archives.postgresql.org/pgsql-hackers/2006-08/msg01874.php

We can have the following syntax to support auto creation of partitions in Postgresql:

```
 CREATE TABLE tabname (
     ...
     ...
  ) PARTITION BY
```

You get more feedback

From: **Peter Eisentraut <peter_e (at) gmx (dot) net>**
To: pgsql-hackers (at) postgresql (dot) org
Subject: **Re: [PATCHES] Enums patch v2**
Date: Tue, 19 Dec 2006 10:58:48 +0100

```
Heikki Linnakangas wrote:
> I'm sorry I missed the original discussions, but I have to ask: Why
> do we want enums in core? The only potential advantage I can see over
> using a look-up table and FK references is performance.

The difference is that foreign-key-referenced data is part of your data
whereas enums would be part of the type system used to model the data.
```

From: **Andrew Sullivan <ajs (at) crankycanuck (dot) ca>**
To: **Andrew Dunstan <andrew (at) dunslane (dot) net>**
Subject: **Re: Idea for new data type**
Date: Tue, 6 Feb 2007 11:46:24 -0500

```
On Tue, Feb 06, 2007 at 11:43:24AM -0500, Andrew Dunstan wrote:
> Andrew Sullivan wrote:
> >
> >As a data point, some time ago (7.2 days) I used to do this as a
> >matter of completeness, and never had a collision.
>
> The point I at least have been trying to make is that extensions
```

From: **Neil Conway <neilc (at) samurai (dot) com>**
To: **Martijn van Oosterhout <kleptog (at) svana (dot) org>**
Subject: **Re: Asynchronous I/O Support**
Date: Sun, 15 Oct 2006 14:26:12 -0400

```
On Sun, 2006-10-15 at 19:56 +0200, Martijn van Oosterhout wrote:
> Sure, I even implemented it once. Didn't get any faster.

Did you just do something akin to s/read/aio_read/ etc., or something
more ambitious? I think that really taking advantage of the ability to
have multiple I/O requests outstanding would take some leg work.

> Maybe one of these days I should work out if my current system supports
> it, and give it another go...
```

If you don't, remind people until you do.

From: NikhilS <nikkhils (at) gmail (dot) com>
To: pgsql-hackers (at) postgresql (dot) org
Subject: Re: **Auto creation of Partitions**
Date: Tue, 16 Mar 2007 17:27:50 +0530

```
Hey!  I haven't seen any comments on the spec I
    posted 10 days ago.  This is on the TODO
    list; how about a little feedback?
```

You do several drafts of the specification.

```
    table customers
          |
          |
  split @ name >= 'N'
   /               \
  /                 \
part1              part2
```

```
For such a modulo node, the executor would have to
start multiple index scans, i.e.:
                         table sample
                               |
                               |
                         'id' modulo 4
                       /    |    |      \
                      /     |    |       \
                  part1  part2  part3  part4
```

```
A combination of the two would look like:
```

```
            table invoices
                  |
                  |
          split @ id >= 50000
           /                \
          /                  \
 hash(id) modulo 3          part4
    /    |     \
   /     |      \
part1   part2   part3
```

The specification gets done, or the discussion bogs down.

Follow-Ups:

From: Tom Lane

From: Martijn van Oosterhout

References:

From: NikhilS

From: Markus Schiltknecht

From: Simon Riggs

From: Gregory Stark

From: Markus Schiltknecht

From: Gregory Stark

From: Markus Schiltknecht

From: Simon Riggs

From: Joshua D. Drake

From: Markus Schiltknecht

From: Martijn van Oosterhout

From: Luke Lonergan

From: Simon Riggs

From: Shane Ambler

From: NikhilS

From: Simon Riggs

You write some code.

```
+ void
+ DefinePartitions(CreateStmt *stmt)
+ {
+   CreateStmt     *childStmt;
+   RangeVar       inr;
+   Oid            childOid;
+   PartitionAttrs *partAttr;
+
+   if (stmt->partAttr == NULL)
+       return;
+
+   Assert(IsA(stmt->partAttr, PartitionAttrs));
+   partAttr = (PartitionAttrs *)(stmt->partAttr);
+
+   /*
+    * All the partitions will inherit from the parent,
set the parent in the
+    * inhRelations structure
```

You create a proof-of-concept patch, and post it.

Auto Partitioning Patch - WIP version 1

From: NikhilS <nikkhils (at) gmail (dot) com>
To: pgsql-patches (at) postgresql (dot) org
Subject: Auto Partitioning Patch - WIP version 1
Date: Fri, 30 Mar 2007 12:28:02 +0530

Hi,

Please find attached the WIP version 1 of the auto partitioning patch. There was discussion on this a while back on -hackers at:

You get more feedback.

From: **Peter Eisentraut <peter_e (at) gmx (dot) net>**
To: pgsql-hackers (at) postgresql (dot) org
Subject: **Re: [PATCHES] Enums patch v2**
Date: Tue, 19 Dec 2006 10:58:48 +0100

```
Heikki Linnakangas wrote:
> I'm sorry I missed the original discussions, but I have to ask: Why
> do we want enums in core? The only potential advantage I can see over
> using a look-up table and FK references is performance.

The difference is that foreign-key-referenced data is part of your data
whereas enums would be part of the type system used to model the data.
```

From: **Andrew Sullivan <ajs (at) crankycanuck (dot) ca>**
To: **Andrew Dunstan <andrew (at) dunslane (dot) net>**
Subject: **Re: Idea for new data type**
Date: Tue, 6 Feb 2007 11:46:24 -0500

```
On Tue, Feb 06, 2007 at 11:43:24AM -0500, Andrew Dunstan wrote:
> Andrew Sullivan wrote:
> >
> >As a data point, some time ago (7.2 days) I used to do this as a
> >matter of completeness, and never had a collision.
>
> The point I at least have been trying to make is that extensions
```

From: **Neil Conway <neilc (at) samurai (dot) com>**
To: **Martijn van Oosterhout <kleptog (at) svana (dot) org>**
Subject: **Re: Asynchronous I/O Support**
Date: Sun, 15 Oct 2006 14:26:12 -0400

```
On Sun, 2006-10-15 at 19:56 +0200, Martijn van Oosterhout wrote:
> Sure, I even implemented it once. Didn't get any faster.

Did you just do something akin to s/read/aio_read/ etc., or something
more ambitious? I think that really taking advantage of the ability to
have multiple I/O requests outstanding would take some leg work.

> Maybe one of these days I should work out if my current system supports
> it, and give it another go...
```

You develop a beta patch.

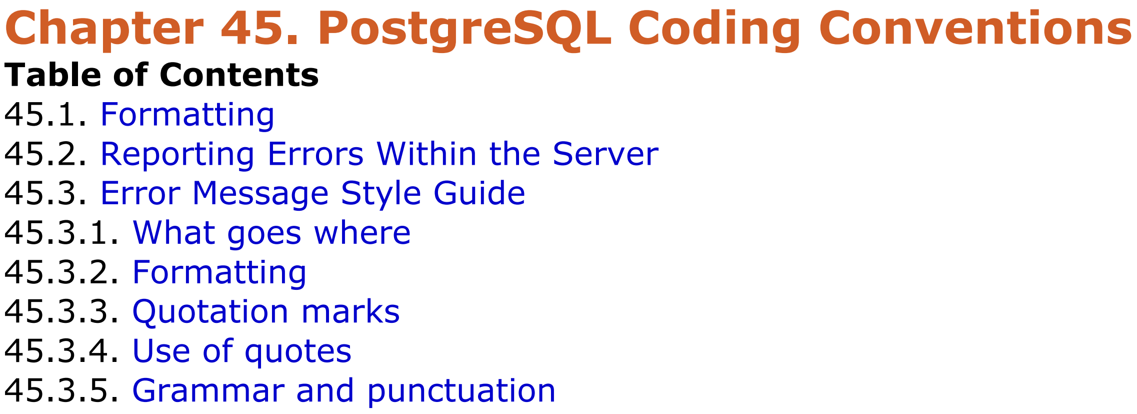
Auto Partitioning Patch - WIP version 6

From: NikhilS <nikkhils (at) gmail (dot) com>
To: pgsql-patches (at) postgresql (dot) org
Subject: Auto Partitioning Patch - WIP version 6
Date: Fri, 30 Mar 2007 12:28:02 +0530

Hi,

Please find attached the WIP version 6 of the auto partitioning patch. I've fixed the race condition pointed out with 5 ...

You apply PostgreSQL format and commenting standards.



Chapter 45. PostgreSQL Coding Conventions

Table of Contents

45.1. Formatting
45.2. Reporting Errors Within the Server
45.3. Error Message Style Guide
45.3.1. What goes where
45.3.2. Formatting
45.3.3. Quotation marks
45.3.4. Use of quotes
45.3.5. Grammar and punctuation

```
+   * ----------------------------------------------------------------------------
+   *    DefinePartitions
+   *    Create new partitions. They end up inheriting from the parent
+   *    relation.
+   *    Once they have been created, rules need to be assigned to the parent to
+   *    provide the UPDATEs/INSERTs/DELETEs to percolate down to the children
+   *        Callers expect this function to end with CommandCounterIncrement if it
+   *        makes any changes.
+   * ----------------------------------------------------------------------------
+   */
```

You submit to pgsql-patches.

From: Magnus Hagander <magnus (at) hagander (dot) net>
To: pgsql-patches <pgsql-patches (at) postgresql (dot) org>
Subject: SSPI authentication - patch
Date: Wed, 18 Jul 2007 12:16:42 +0200

Attached is the patch to support SSPI authentication in libpq. With this
patch, I can authenticate windows clients against a linux server using
kerberos *without* reqiring setting up MIT kerberos on the windows side.
Protocol has not changed at all.

A hacker reviews your patch.

From: Tom Lane <tgl (at) sss (dot) pgh (dot) pa (dot) us>
To: Magnus Hagander <magnus (at) hagander (dot) net>
Subject: Re: SSPI authentication - patch
Date: Fri, 20 Jul 2007 12:47:35 -0400

```
Magnus Hagander <magnus ( at ) hagander ( dot ) net> writes:
> Stephen Frost wrote:
>> That's true, but if we used upper-case with something NEW (SSPI) while
>> keeping it the same for the OLD (KRB5, and I'd vote GSSAPI) then we're
>> not breaking backwards compatibility while also catering to the masses.
>> I guess I don't see too many people using SSPI w/ an MIT KDC, and it
>> wasn't possible previously anyway.
>>
>> What do you think?

> Hmm. It makes the default a lot less clear, and opens up for confusion.
> So I'm not so sure I like it :-)
```

You revise & debug

From: Magnus Hagander <magnus (at) hagander (dot) net>
To: pgsql-patches <pgsql-patches (at) postgresql (dot) org>
Subject: Re: SSPI authentication - patch
Date: Thu, 19 Jul 2007 13:53:15 +0200

On Wed, Jul 18, 2007 at 12:16:42PM +0200, Magnus Hagander wrote:

> Attached is the patch to support SSPI authentication in libpq. With this
> patch, I can authenticate windows clients against a linux server using
> kerberos *without* reqiring setting up MIT kerberos on the windows side.
> Protocol has not changed at all.

Here's an updated version of this patch. This version has full SSPI support in the server as well, so I can do both kerberos and NTLM between two windows machines using the negotiate method.

Eventually your patch is accepted in the queue

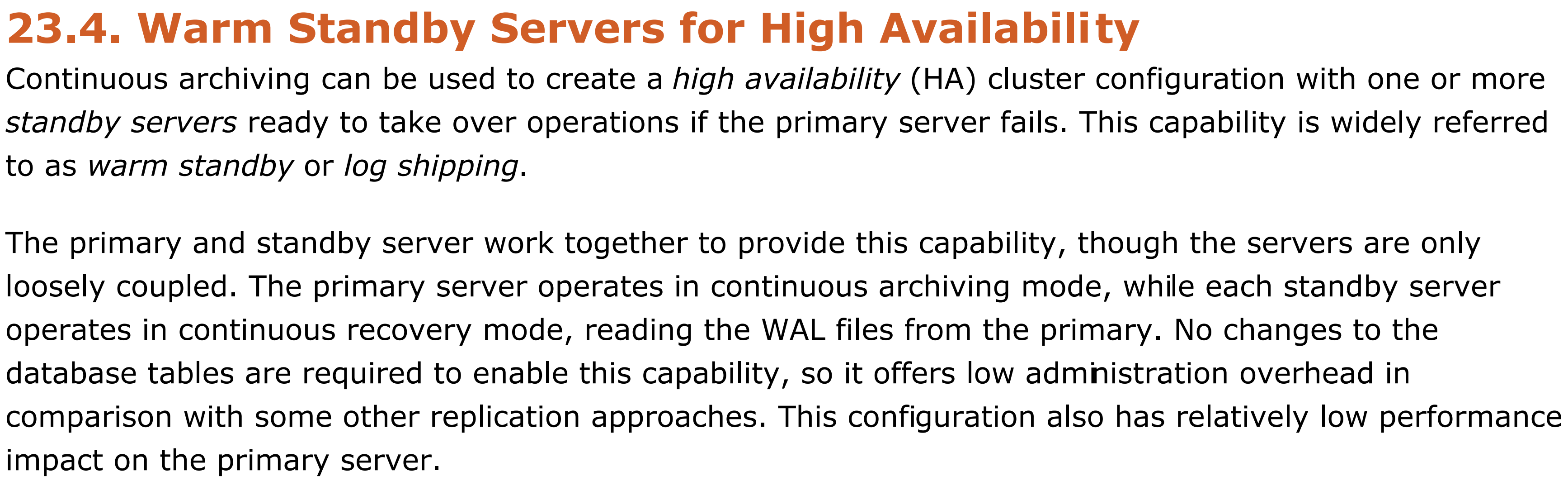
From: Bruce Momjian <bruce (at) momjian (dot) us>
To: Simon Riggs <simon (at) 2ndquadrant (dot) com>
Subject: Re: Synchronous Commit Doc Patch
Date: Tue, 17 Jul 2007 01:09:25 -0400 (EDT)

Your patch has been added to the PostgreSQL unapplied patches list at:

http://momjian.postgresql.org/cgi-bin/pgpatches

It will be applied as soon as one of the PostgreSQL committers reviews
and approves it.

You write the documentation.



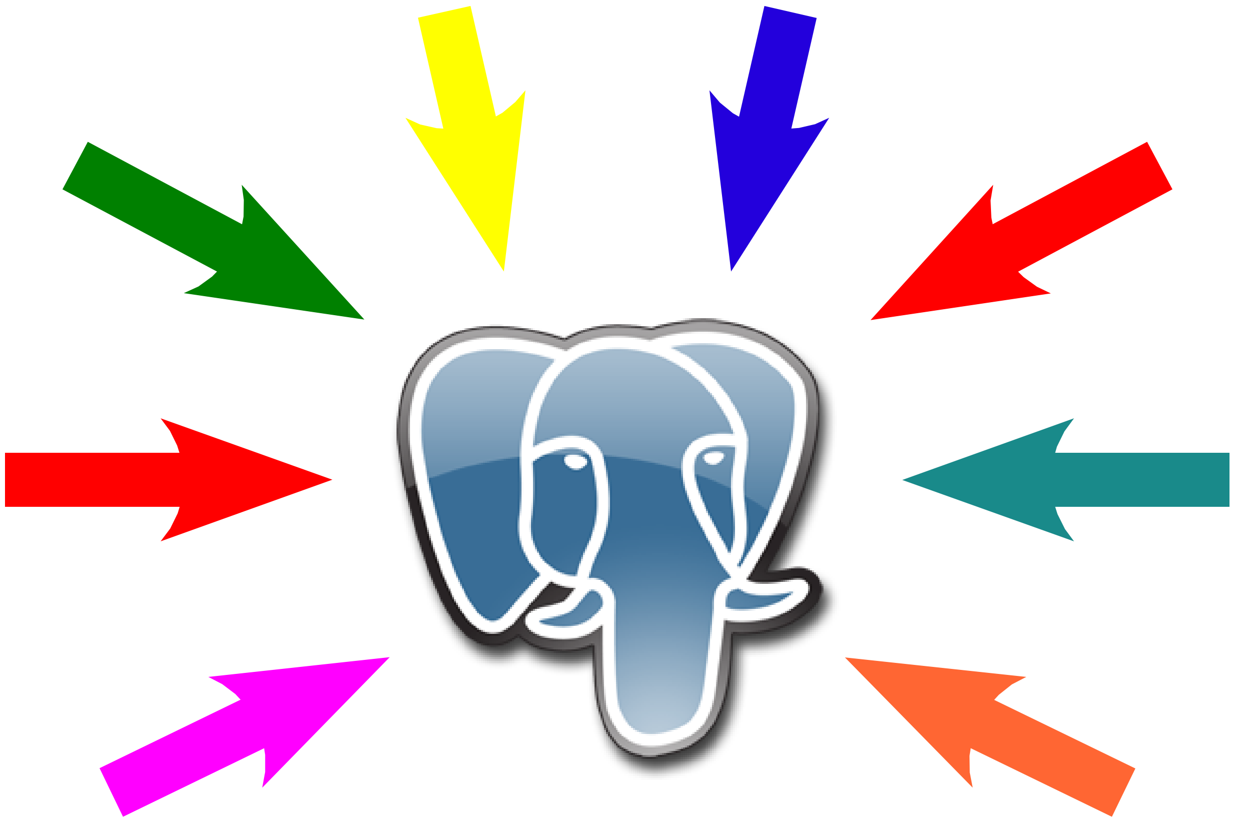
23.4. Warm Standby Servers for High Availability

Continuous archiving can be used to create a *high availability* (HA) cluster configuration with one or more *standby servers* ready to take over operations if the primary server fails. This capability is widely referred to as *warm standby* or *log shipping*.

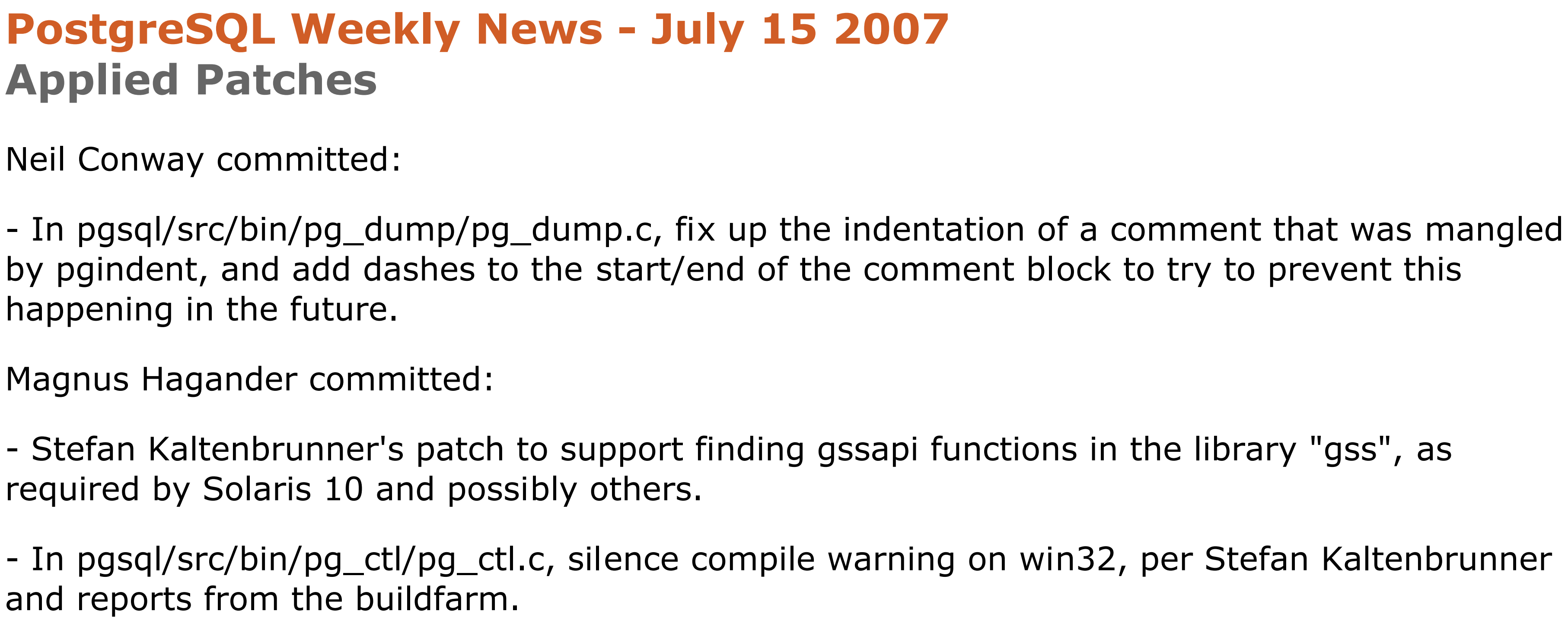
The primary and standby server work together to provide this capability, though the servers are only loosely coupled. The primary server operates in continuous archiving mode, while each standby server operates in continuous recovery mode, reading the WAL files from the primary. No changes to the database tables are required to enable this capability, so it offers low administration overhead in comparison with some other replication approaches. This configuration also has relatively low performance impact on the primary server.

Feature freeze hits

You are called back to help make your patch integrate with others.



Your patch is applied by a committer.



PostgreSQL Weekly News - July 15 2007

Applied Patches

Neil Conway committed:

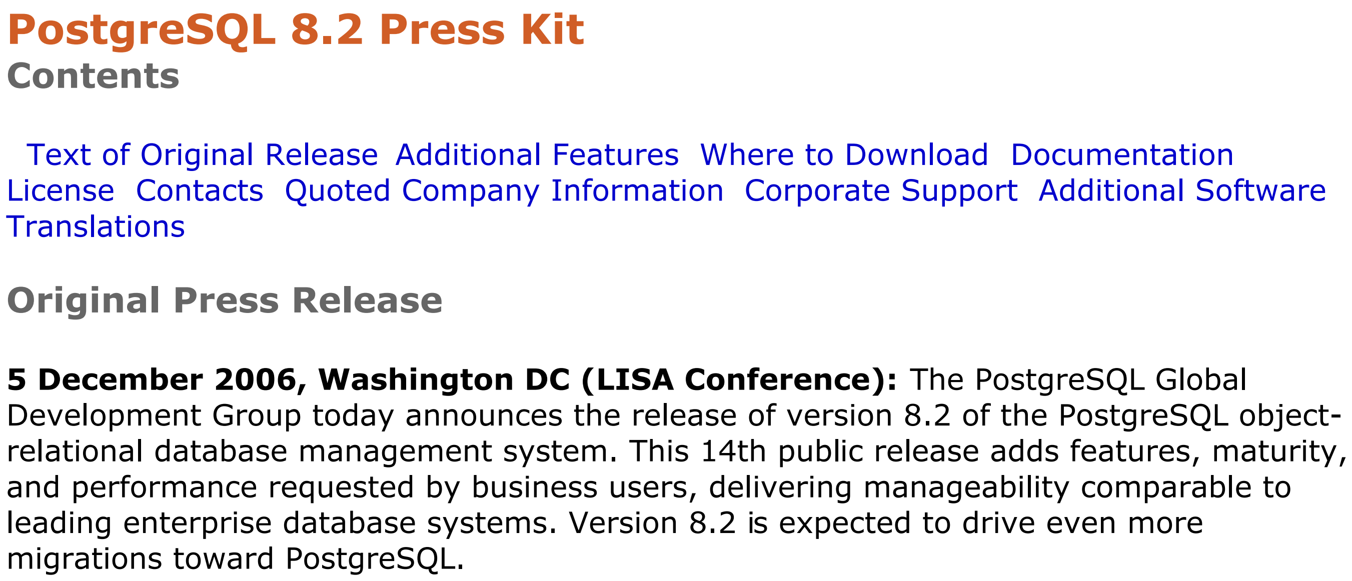
- In pgsql/src/bin/pg_dump/pg_dump.c, fix up the indentation of a comment that was mangled by pgindent, and add dashes to the start/end of the comment block to try to prevent this happening in the future.

Magnus Hagander committed:

- Stefan Kaltenbrunner's patch to support finding gssapi functions in the library "gss", as required by Solaris 10 and possibly others.

- In pgsql/src/bin/pg_ctl/pg_ctl.c, silence compile warning on win32, per Stefan Kaltenbrunner and reports from the buildfarm.

PostgreSQL does a release.



PostgreSQL 8.2 Press Kit

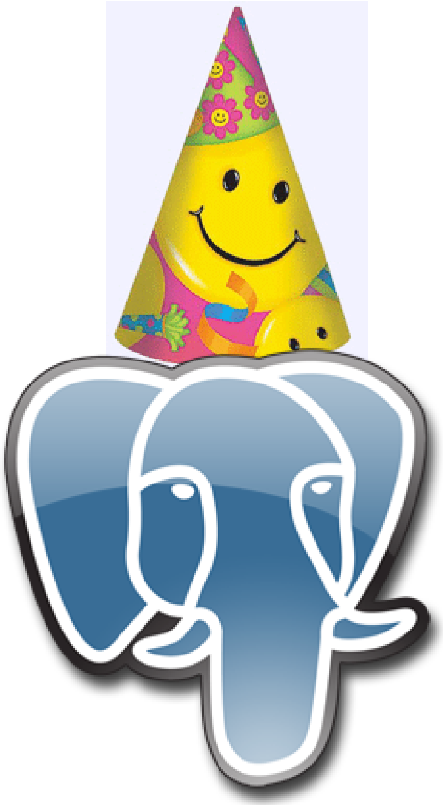
Contents

Text of Original Release Additional Features Where to Download Documentation License Contacts Quoted Company Information Corporate Support Additional Software Translations

Original Press Release

5 December 2006, Washington DC (LISA Conference): The PostgreSQL Global Development Group today announces the release of version 8.2 of the PostgreSQL object-relational database management system. This 14th public release adds features, maturity, and performance requested by business users, delivering manageability comparable to leading enterprise database systems. Version 8.2 is expected to drive even more migrations toward PostgreSQL.

Your patch is now a feature.



Now you get to maintain it for the rest of your life!

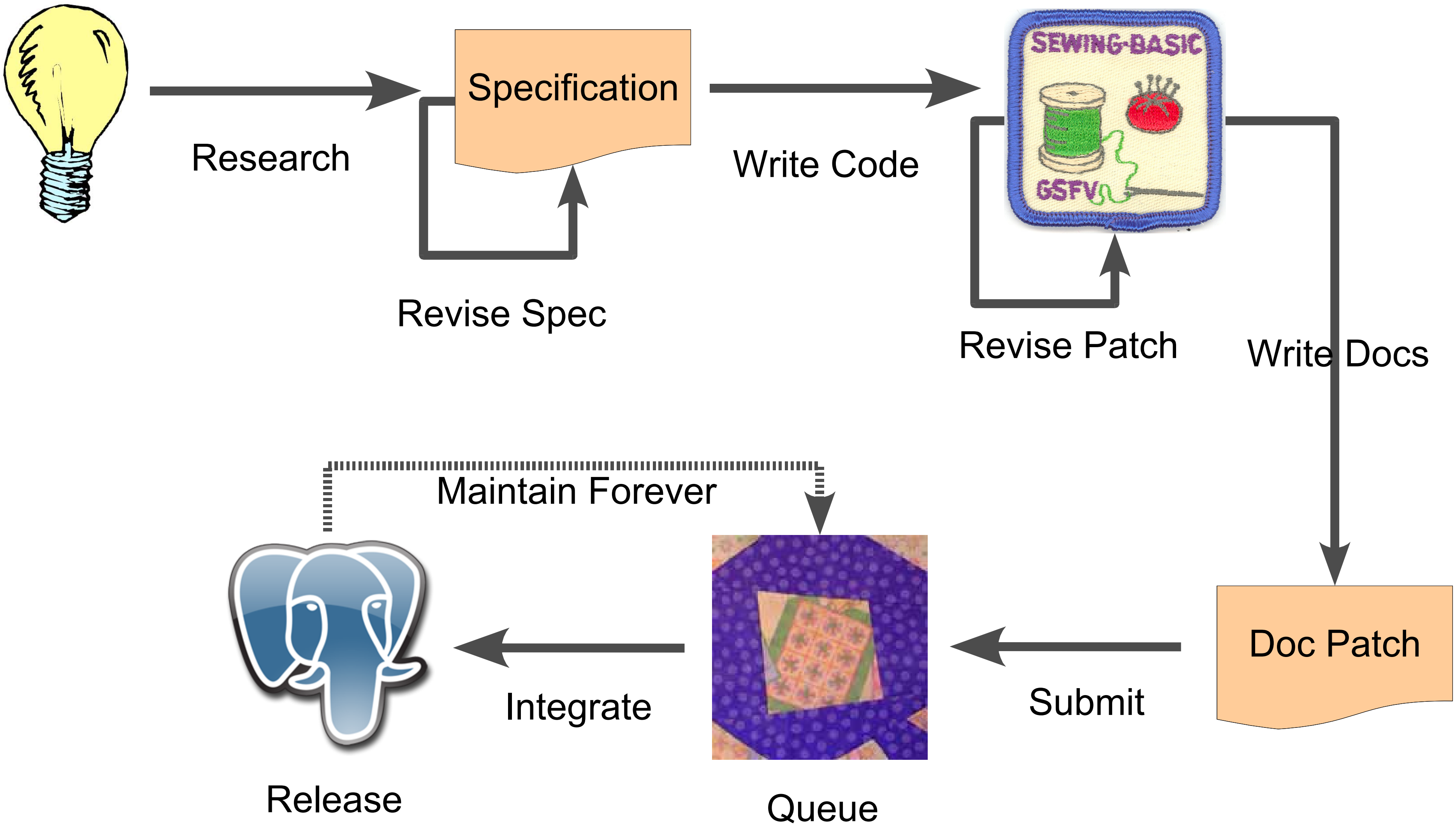
From: Joe Conway <mail (at) joeconway (dot) com>
To: Robert Treat <xzilla (at) users (dot) sourceforge (dot) net>
Subject: Re: dblink connection security
Date: Sun, 01 Jul 2007 14:12:09 -0700

Robert Treat wrote:
Joe Conway <mail (at) joeconway (dot) com> writes:
Well certainly dbi-link has the exact same issue.
dbi-link only works in plperlu, so you've already decided your superuser only.

How so -- it is fundamentally no different than dblink, which is C language (also untrusted).

I think the issue is that once the superuser creates said functions, usage of the functions is automatically granted to PUBLIC, no? Being an untrusted language just means that it takes a superuser to create the functions using that language, not to use the functions themselves.
Joe

Patch Lifecycle



Research

Specification

Revise Spec

Write Code

Revise Patch

Write Docs

Doc Patch

Submit

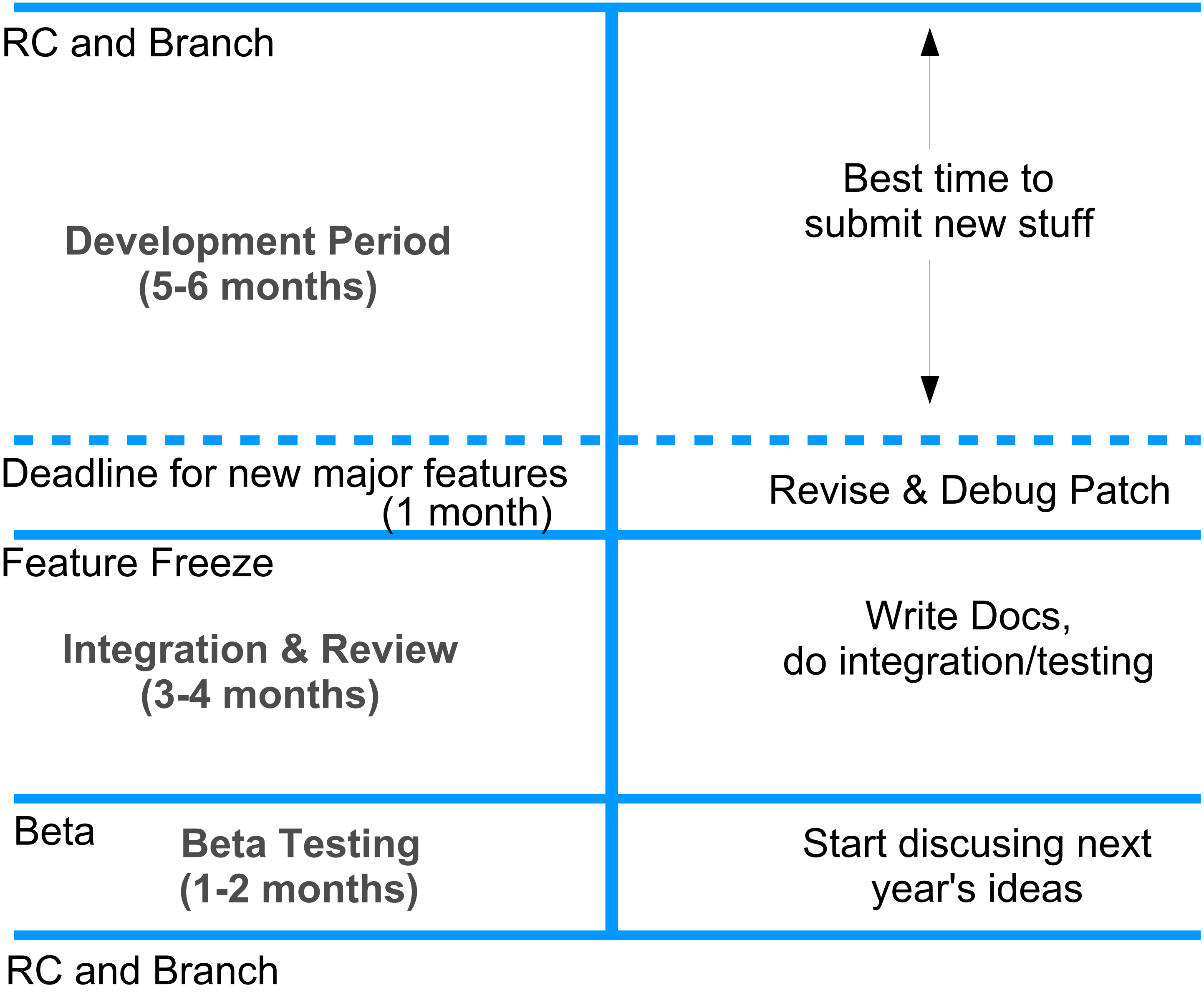
Queue

Integrate

Release

Maintain Forever

A PostgreSQL Year



RC and Branch

Development Period (5-6 months)

Best time to submit new stuff

Deadline for new major features (1 month)

Revise & Debug Patch

Feature Freeze

Integration & Review (3-4 months)

Write Docs, do integration/testing

Beta

Beta Testing (1-2 months)

Start discusing next year's ideas

RC and Branch

Other tips on submitting

- Don't get discouraged.
  - Be prepared to argue.
  - One hacker rejecting your idea doesn't mean everyone does.
  - Committers (esp. Tom Lane) are often more concerned about maintainability than cool stuff.
- Be flexible: you *will* have to make changes.
  - Corporate coding standards are generally lower than the project's.

Other tips on submitting

- Don't use the wrong arguments
  - “MySQL/Oracle does it this way.”
  - “Based on this hot academic trend.”
- Some things make a patch harder to accept
  - New syntax
  - Backwards compatibility issues
  - High code counts
- Don't get discouraged.

Now, go write some code.

or contribute in some easier way

Contact Information

- Josh Berkus
  - www.powerpostgresql.com
  - josh@postgresql.org
  - blogs.ittoolbox.com/ database/soup
- PostgreSQL on Solaris
  - www.sun.com/ postgresql

- Upcoming events
  - LinuxWorldExpo San Francisco, August 6-9
  - Sun TechDays Milan, Rome -- This September