

# PL/P Parrot

PGCon 2010, Ottawa  
May 21, 2010

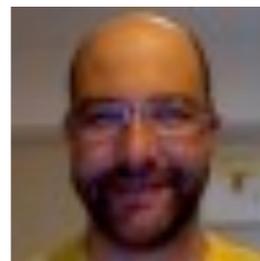
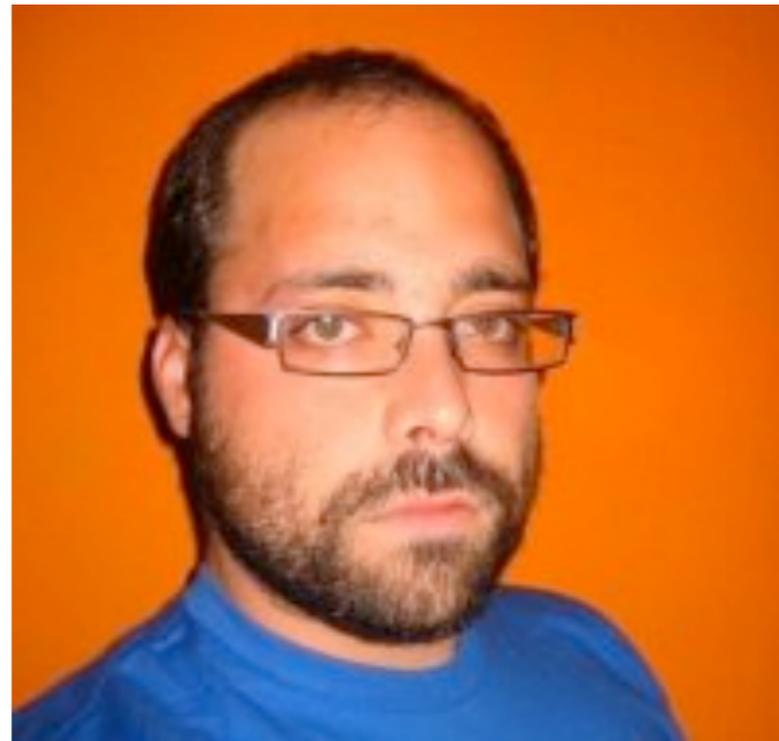
Copyright © 2010  
David Fetter [david.fetter@pgexperts.com](mailto:david.fetter@pgexperts.com)  
All Rights Reserved





**Um, WTF?!?!?**

# People To Blame



# Parrot

# Parrot

- *A Virtual Machine for Dynamic Languages*

# Parrot

- A Virtual Machine for Dynamic Languages
- Register-based

# Parrot

- A Virtual Machine for Dynamic Languages
- Register-based
- Really, really `#&^%#@%^` cool

# Parrot

- A Virtual Machine for Dynamic Languages
- Register-based
- Really, really `#&^%#@%^` cool
  - Pluggable!

# Parrot

- A Virtual Machine for Dynamic Languages
- Register-based
- Really, really `#&^%#@%^` cool
  - Pluggable!
  - Interoperable!

# Parrot

- A Virtual Machine for Dynamic Languages
- Register-based
- Really, really `#&^%#@%^` cool
  - Pluggable!
  - Interoperable!
  - Dynamic!

# PostgreSQL

# PostgreSQL

- If you gotta ask,

# PostgreSQL

- If you gotta ask,
- you ain't never gonna know.

# PostgreSQL

- If you gotta ask,
- you ain't never gonna know.
- (apologies to Louis Armstrong)

# Why?

# Why?

- Writing, Debugging and Maintaining PLs is a

# Why?

- Writing, Debugging and Maintaing PLs is a
- Pain

# Why?

- Writing, Debugging and Maintaing PLs is a
- Pain
- In

# Why?

- Writing, Debugging and Maintaining PLs is a
- Pain
- In
- The

# Why?

- Writing, Debugging and Maintaining PLs is a
- Pain
- In
- The
- As^Wneck

# Why?

# Why?

- PL

# Why?

- PL

- Toolkit

# Why?

# Why?

- Don't

# Why?

- Don't
- Repeat

# Why?

- Don't
- Repeat
- Yourself!

# Why?

- Write in PL/Perl6
- Call from PL/Python

# What

- **First things first:**
- **PL/PIR**



# <EMBED>



# Parrot Embed In Flux

# Parrot Embed In Flux

- Parrot trunk!

# Parrot Embed In Flux

- Parrot trunk!
- PostgreSQL master!

**</EMBED>**

<HLL>



- HLL API not implemented yet, so:

- HLL API not implemented yet, so:
  - PL/PIR (mostly ok)

- HLL API not implemented yet, so:
  - PL/PIR (mostly ok)
  - PL/PIRU (rough edges)

- HLL API not implemented yet, so:
  - PL/PIR (mostly ok)
  - PL/PIRU (rough edges)
- OBTW

- HLL API not implemented yet, so:
  - PL/PIR (mostly ok)
  - PL/PIRU (rough edges)
- OBTW
- PL/Specification

</HLL>

# <MARSHAL>



# PostgreSQL Data Types

# PostgreSQL Data Types

- We support basic data types!

# PostgreSQL Data Types

- We support basic data types!
- Working on time types.

# PostgreSQL Data Types

- We support basic data types!
- Working on time types.
- Parrot needs bytea/time object...

**</MARSHAL>**

# What It Looks Like

```
CREATE FUNCTION test_float_add(float)
RETURNS float
LANGUAGE plparrot
AS $$
    .param num x
    x += 5
    .return(x)
$$;
```

# What It Looks Like

```
CREATE FUNCTION text_concat(text)
RETURNS text
LANGUAGE plparrot
AS $$
    $S1 = 'PL/'
    $S2 = 'Parrot'
    $S3 = $S1 . $S2
    .return($S3)
$$;
```

# Access Control

# Access Control

- **HAVE:**

# Access Control

- **HAVE:**
  - **Filesystem access: DENIED!**

# Access Control

# Access Control

- WANT

# Access Control

- WANT
  - Configurable Controls

# Access Control

- WANT
  - Configurable Controls
  - Network Access Control

# Access Control

- WANT
  - Configurable Controls
  - Network Access Control
  - TESTS

# What We Have Now

# What We Have Now

- An actual PL, PIRU

# What We Have Now

- An actual PL, PIRU
- TAP tests via pgTap

# What We Have Now

- An actual PL, PIRU
- TAP tests via pgTap
- A Git Repo

# What We Have Now

- An actual PL, PIRU
- TAP tests via pgTap
- A Git Repo
- An Issue Tracker

# What We Have Now

- An actual PL, PIRU
- TAP tests via pgTap
- A Git Repo
- An Issue Tracker
- #plparrot on Freenode

# What We Have Now

- An actual PL, PIRU
- TAP tests via pgTap
- A Git Repo
- An Issue Tracker
- #plparrot on Freenode
- Enthusiasm!

# What We'll Do Soon

# What We'll Do Soon

- BETTAR Argument Passing

# What We'll Do Soon

- BETTAR Argument Passing
- MOAR Data Type Marshalling

# What We'll Do Soon

- BETTAR Argument Passing
- MOAR Data Type Marshalling
- TESTS. LOTS OF TESTS

# What We Want

# What We Want

- Sandboxing

# What We Want

- Sandboxing
- HLLs

# What We Want

- Sandboxing
- HLLs
- **MOAR DEVELOPERZ**

# What We Want

- Sandboxing
- HLLs
- MOAR DEVELOPERZ
- Users!

# Where's The Code

<http://github.com/leto/plparrot>

# Thanks, eh!

PGCon 2010, Ottawa  
May 21, 2010

Copyright © 2010  
David Fetter [david.fetter@pgexperts.com](mailto:david.fetter@pgexperts.com)  
All Rights Reserved

**PGX**  
POSTGRES  
EXPERTS, INC.