



August 28, 2020



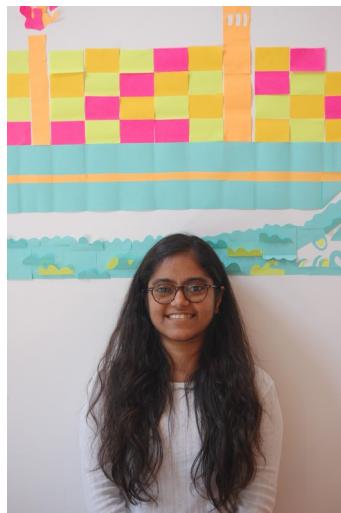
YottaDB Update

K.S. Bhaskar
bhaskar@yottadb.com

- Founded in 2017 by three 20+ year veterans of GT.M
 - Unlock value in free / open-source GT.M code base
 - Exploit full potential across markets & applications
- Organic growth to team of ten in 2020
- Multiple releases & product extensions, 100% FOSS
 - Built around core hierarchical NoSQL database



100% organic growth



Plus others
not shown

We love dogs ...

YottaDB

We're
dogmatic
about
FOSS

... and also
great
software and
documentation

- A mature, high performance, hierarchical key-value NoSQL database whose code base scales up to mission-critical applications like large real-time core-banking and electronic health records, and also scales down to run on platforms like the Raspberry Pi Zero, as well as everything in-between.
- *Rock Solid. Lightning Fast. Secure. Pick any three.*

100% Free / Open Source



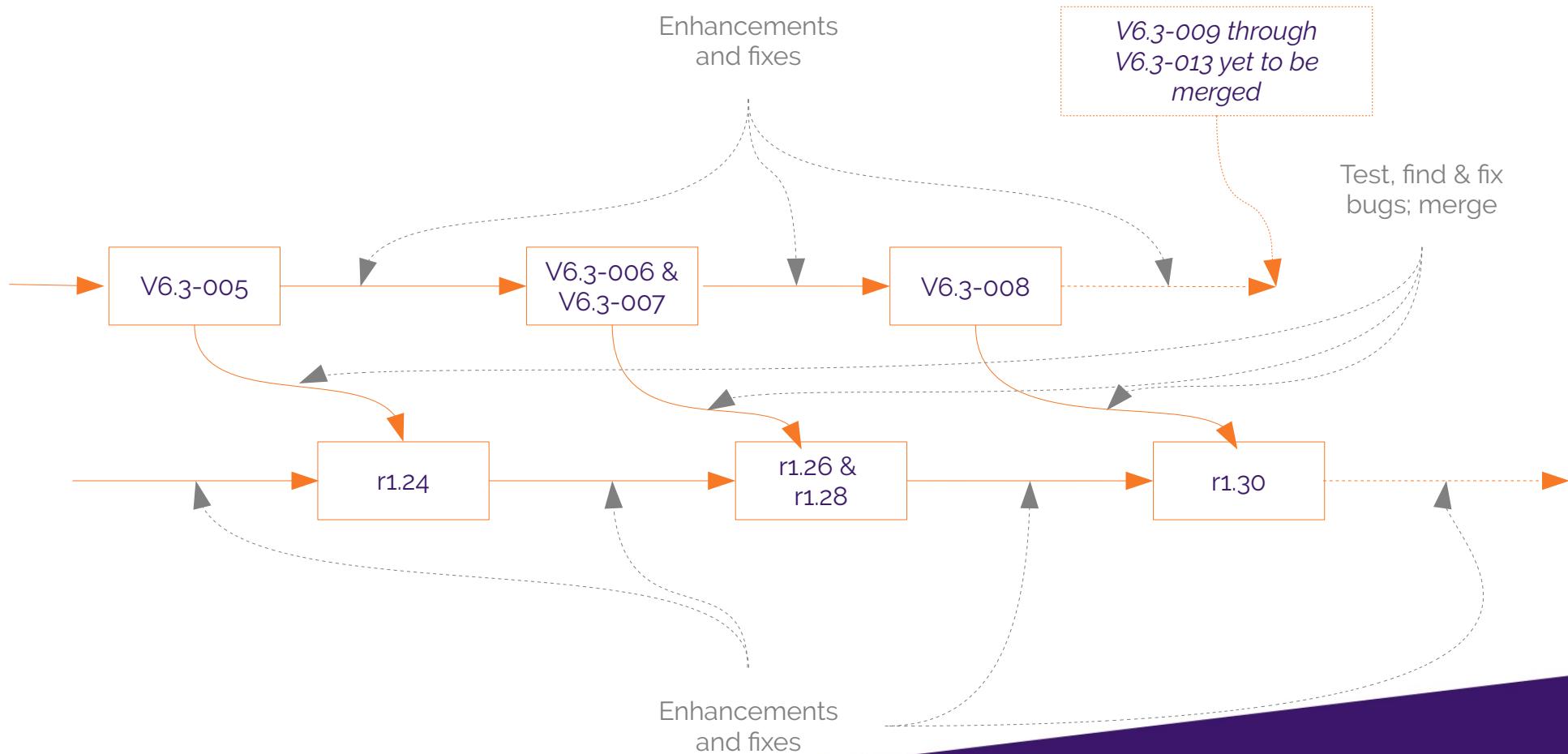
- 100% FOSS project, not just FOSS software
- <https://gitlab.com/YottaDB>
- <https://hub.docker.com/u/yottadb>

Business Update



(discuss verbally)

Compatibility with Upstream



Issues · YottaDB / DB / YDB · GitLab - Mozilla Firefox

Issues · YottaDB / DB / YDB · GitLab

Search

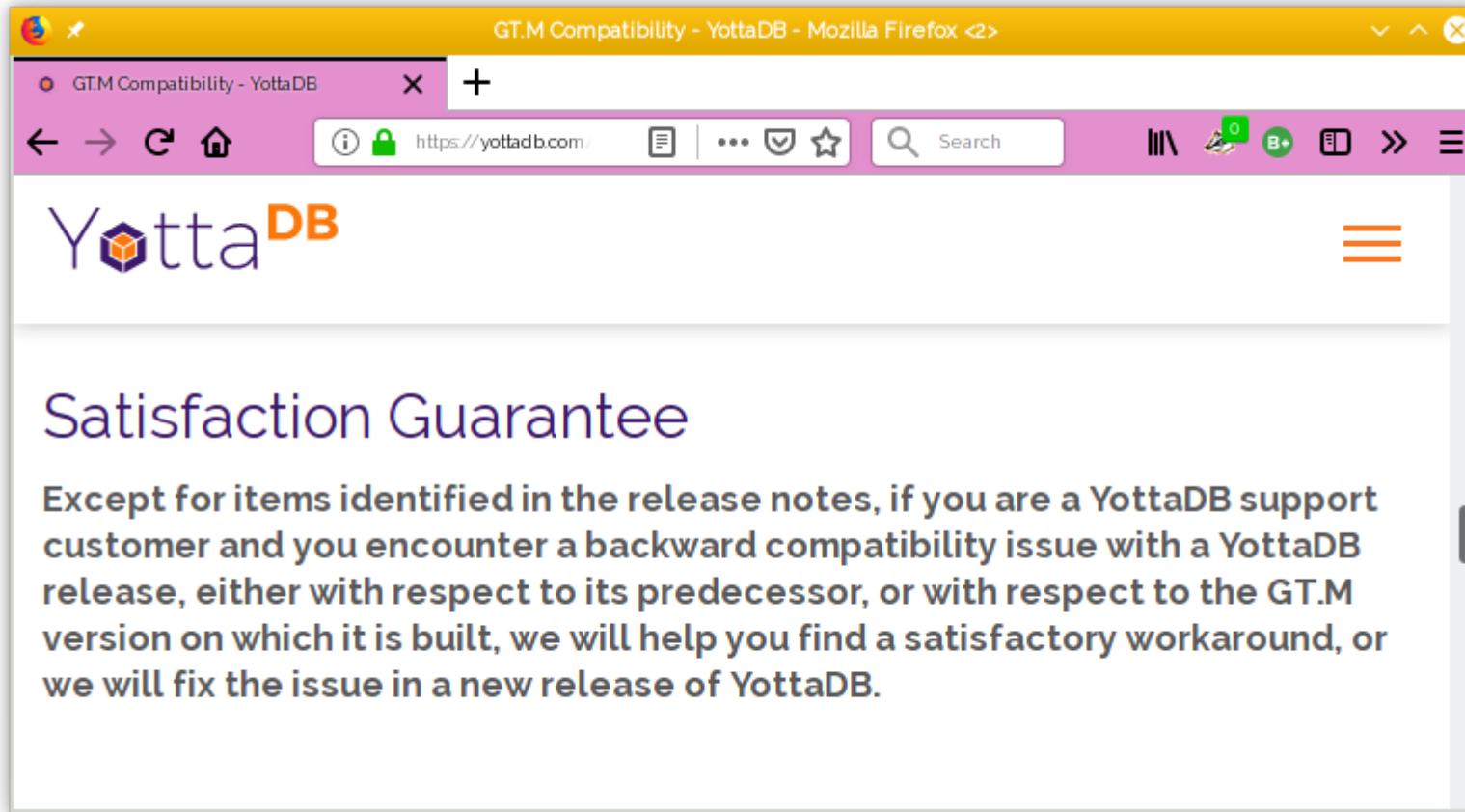
Projects Groups Snippets Help Sign in / Register

Open 15 Closed 100 All 115

Recent searches Label = ~"upstream issue"

Last updated

\$ZSYSLOG breaks formatting of certain strings passed in #551 · opened 5 months ago by Tomas Morstein	CLOSED	20
r1.32 upstream issue	updated 21 hours ago	
Processes accessing YottaDB terminate at a safe point if they are sent a termination signal #560 · opened 5 months ago by Narayanan Iyer	CLOSED	14
r1.30 misfeature upstream issue	updated 5 days ago	
\$ZTDELIM returns piece separator for KILL and ZKILL triggers #596 · opened 1 month ago by Narayanan Iyer	CLOSED	1
r1.30 enhancement upstream issue	updated 5 days ago	
HANG 0 interrupted by MUPIP INTRPT returns forthwith #567 · opened 4 months ago by Narayanan Iyer	CLOSED	2
r1.30 bug upstream issue	updated 2 weeks ago	
SILENT^%RSEL restores value of \$!O #525 · opened 6 months ago by Sam Habel	CLOSED	12
r1.30 bug upstream issue	updated 2 weeks ago	



GT.M Compatibility - YottaDB - Mozilla Firefox <2>

GT.M Compatibility - YottaDB

https://yottadb.com/

Satisfaction Guarantee

Except for items identified in the release notes, if you are a YottaDB support customer and you encounter a backward compatibility issue with a YottaDB release, either with respect to its predecessor, or with respect to the GT.M version on which it is built, we will help you find a satisfactory workaround, or we will fix the issue in a new release of YottaDB.

- r1.26 – June 27, 2019 – major release
 - Fully Supported functions for multi-threaded applications
 - `ydb_env_set` and `ydb_env_unset`
 - Enhancements and fixes inherited from GT.M V6.3-006 and V6.3-007
 - <https://gitlab.com/YottaDB/DB/YDB/-/releases#r126>

- r1.26 – June 27, 2019 – major release
- r1.28 – September 11, 2019 – major release for Go API
 - Go wrapper (API) supported for production use
 - Minor enhancements, e.g.,
 - \$fnumber() formatting code ". "
 - new \$test
 - <https://gitlab.com/YottaDB/DB/YDB/-/releases#r128>

Releases, 2020



- r1.30 – August 14, 2020 – major release
 - Many, many, enhancements & fixes
 - Functionality required for Rust wrapper and Octo®
 - GT.M V6.3-008 merged
 - <https://gitlab.com/YottaDB/DB/YDB/-/releases#r130>

Octo is a registered trademark of YottaDB LLC

r1.30 · Tags · YottaDB / DB / YDB · GitLab - Chromium <2>

r1.30 · Tags · YottaDB / DB / YDB · [Paused](#)

gitlab.com/YottaDB/DB/YDB/-/tags/r1.30

Projects Groups More + 13 6 ? :

The most important new functionality, required by Octo to distinguish between the empty string ("") and the absence of data (NULL), is the intrinsic special variable, \$ZYSQLNULL, which can be used as a local variable subscript or node value. [\(484\)](#)

Other highlights include:

- The `ydbinstall` / `ydbinstall.sh` scripts include `--enclugin`, `--octo`, and `--posix` command line options to install respectively, the encryption plugin, Octo, and the POSIX plugin. [\(458\)](#), [\(457\)](#), [\(521\)](#)
- The `%YDBPROCSTUCKEXEC` routine provides a standard action that can be invoked by `ydb_proctuckexec`. [\(579\)](#)
- The call to `$ZJOBEXAM()` allows for specification of the data to dumped. So, for example, the dumping of local variables can be suppressed unless required, as local variables can contain confidential information. [\(482\)](#)
- `yottadb -version` provides a detailed report on the YottaDB build. [\(595\)](#)
- For faster numeric base conversion, `$ZCONVERT()` converts between decimal and hexadecimal. [\(485\)](#)
- `$ZYHASH()` returns the 128-bit MurmurHash3 hash of a string. [\(390\)](#)
- Simple API functions use nanosecond timers internally. [\(388\)](#)
- A new implementation of `$RANDOM()`. [\(384\)](#)
- To facilitate migration to YottaDB from big endian versions of the upstream code base, YottaDB can automatically convert global directories from big to little endian formats. Note that all YottaDB platforms are little endian. [\(524\)](#)
- CentOS 8 and Ubuntu 20.04 LTS on x86_64 are Supported platforms with their own binary distributions.

Yottadb r1.30 is upward compatible with YottaDB r1.28, as well as with GT.M V6.3-008.

Persistence is About Data, not Languages



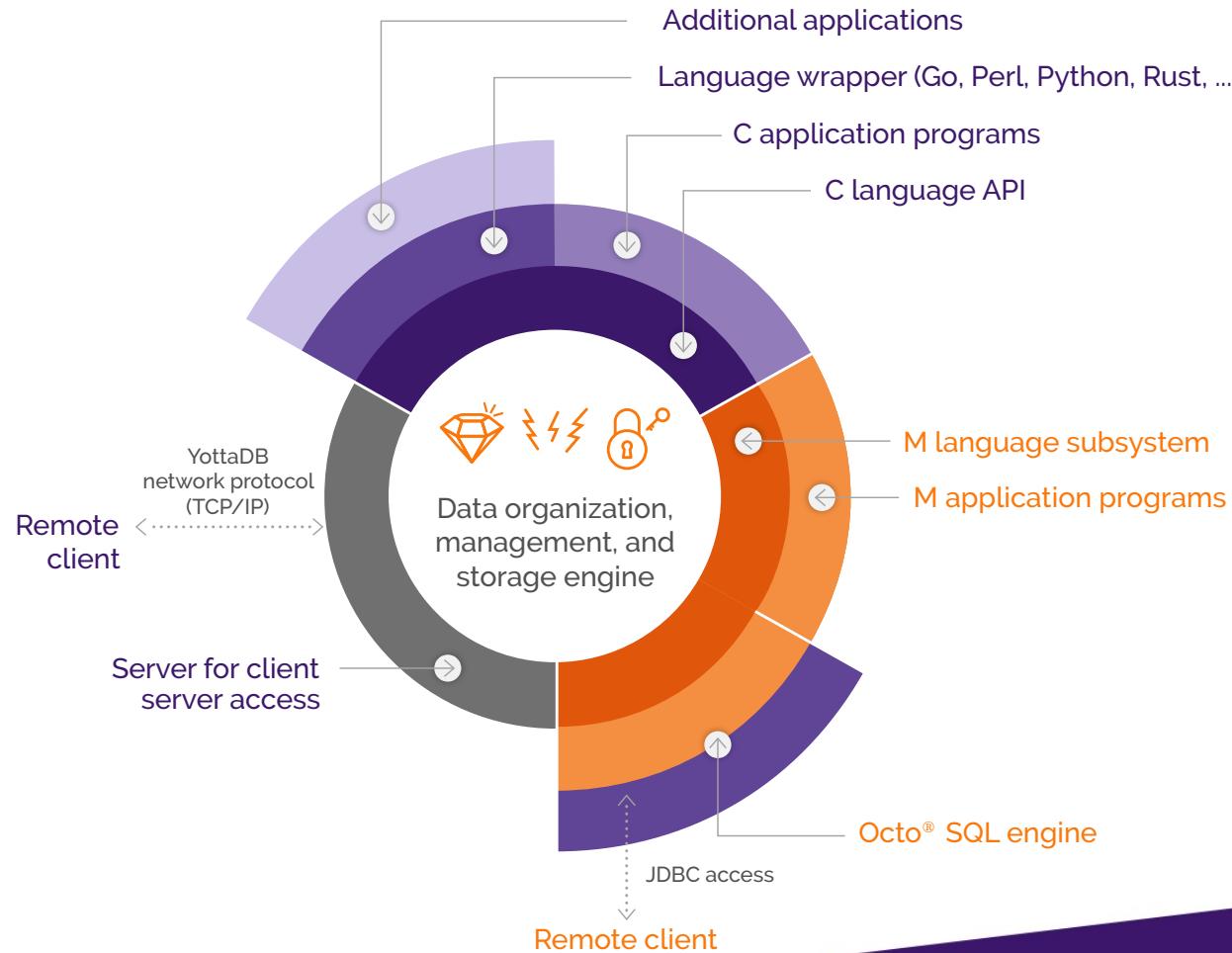
- Languages are like religions
 - I'll change mine if you'll change yours
 - (But you go first...)
- YottaDB – language agnostic access to data
 - Any language, any platform, anywhere
 - (At least, that's the goal...)
- More languages = more opportunities (hopefully)

Key-Value Model is Universal



- M local/global variables
- XML
- JSON
- And extends to other data models
 - <http://mgateway.com/docs/universalNoSQL.pdf>
- Including relational (Octo)

YottaDB Data-Centric Architecture



Programming in C – MultiLanguage Programmers Guide documentation - Mozilla Firefox

Programming in C – MultiLanguag x +

Search https://docs.yottadb.com/MultiLar ... Search

Index

Main YottaDB Documentation Page

Multi-Language Programmer's Guide

Programming in M

Programming in C

- Symbolic Constants
 - Data Structures & Type Definitions
 - Macros
- Simple API
 - Comprehensive API
- Utility Functions
 - Calling M Routines

Programming in Go

Programming Notes (Avoiding Common Pitfalls)

- `ydb_data_s()` / `ydb_data_st()`
- `ydb_delete_s()` / `ydb_delete_st()`
- `ydb_delete_excl_s()` / `ydb_delete_excl_st()`
- `ydb_get_s()` / `ydb_get_st()`
- `ydb_incr_s()` / `ydb_incr_st()`
- `ydb_lock_s()` / `ydb_lock_st()`
- `ydb_lock_decr_s()` / `ydb_lock_decr_st()`
- `ydb_lock_incr_s()` / `ydb_lock_incr_st()`
- `ydb_node_next_s()` / `ydb_node_next_st()`
- `ydb_node_previous_s()` / `ydb_node_previous_st()`
- `ydb_set_s()` / `ydb_set_st()`
- `ydb_str2zwr_s()` / `ydb_str2zwr_st()`
- `ydb_subscript_next_s()` / `ydb_subscript_next_st()`
- `ydb_subscript_previous_s()` / `ydb_subscript_previous_st()`
- `ydb_tp_s()` / `ydb_tp_st()`
- `ydb_zwr2str_s()` / `ydb_zwr2str_st()`

Hello world – C



```
#include "libyottadb.h"
int main()
{
    ydb_buffer_t    lang[1], value, varname;
    YDB_LITERAL_TO_BUFFER("^hello", &varname);
    YDB_LITERAL_TO_BUFFER("C", &lang[0]);
    YDB_LITERAL_TO_BUFFER("Hello, world!", &value)
    return ydb_set_s(&varname, 1, &lang[0], &value);
}
```

Go

A screenshot of a Mozilla Firefox browser window. The title bar says "yottadb package · pkg.go.dev - Mozilla Firefox". The address bar shows the URL "https://pkg.go.dev/lang.yottadb.cc". The page content is for the "yottadb" package, version v1.1.0, published on Aug 14, 2020. It includes sections for Overview, YottaDB Quick Start, and a command-line interface example.

yottadb package · pkg.go.dev - Mozilla Firefox

Black Lives Matter Support the Equal Justice Initiative

lang.yottadb.com/go/yottadb

Package yottadb v1.1.0 Latest

Published: Aug 14, 2020 | Licenses: [AGPL-3.0](#), [AGPL-3.0](#) | Module: lang.yottadb.com/go/yottadb

Doc Overview Subdirectories Versions Imports Imported By Licenses

• Overview Examples Constants ↗ Functions ↗ Types

Overview

Package yottadb provides a Go wrapper for YottaDB - a mature, high performance, transactional NoSQL engine with proven speed and stability.

YottaDB Quick Start

Before starting, consider reading the introduction to YottaDB's data model at <https://docs.yottadb.com/MultiLangProgGuide/MultiLangProgGuide.html#concepts>

The YottaDB Go wrapper requires a minimum YottaDB version of r1.30 and is tested with a minimum Go version of 1.13. If the Go packages on your operating system are older, and the Go wrapper does not work, please obtain and install a newer Go implementation.

This quickstart assumes that YottaDB has already been installed as described at <https://yottadb.com/product/get-started/>.

After installing YottaDB, install the Go wrapper:

YottaDB

Hello world – Go



```
package main
import (
    "lang.yottadb.com/go/yottadb"
)
func main() {
    defer yottadb.Exit()
    _ = yottadb.SetValE(yottadb.NOTTP, nil, "สวัสดิ์ชาวโลก",
        "^hello", []string{"Go"})
}
```

M – Of course



Hello world – M



```
sayhelloM  
    set ^hello("M")="Приветствую, мир!"  
quit
```

YottaDB - Perl extension for accessing YottaDB - metacpan.org - Mozilla Firefox

YottaDB - Perl extension for acc

Testers (0/0/0)
Kwalitee
% Coverage
License: unknown

ACTIVITY

24 month

TOOLS

Download (74.53Kb)
MetaCPAN Explorer
Permissions
Subscribe to distribution
Install Instructions

Search distribution
grep distribution
Jump to version
Diff with version

PERMALINKS

This version
Latest version

\$data = y_data \$var [, @subs]

The *y_data* function returns in `$data`:

- 0 - no value and no subtree
- 1 - has a value but no subtree
- 10 - no value but a subtree
- 11 - a value and a subtree exists

y_killall ()

The *y_killall* function kills all local variables.

y_kill_excl [\$var0 , \$var1 [...]]

The *y_kill_excl* function deletes all local variables except the specified one(s). *y_kill_excl* without arguments is the same as *y_killall*.

y_kill_node \$var [, @subs]

Deletes a node but not a subtree.

y_kill_tree \$var [, @subs]

Deletes a node and all subtrees.

y_set \$var, [@subs,] \$value

Sets the variable to `$value`

Thank you,
Stefan Traby!

Hello world – Perl



```
#! /usr/bin/perl
use YottaDB qw(:all);
y_set "^hello", "Perl", "Grüß Gott Welt";
```

The screenshot shows a Mozilla Firefox window displaying the Rust documentation for the `yottadb` crate on GitLab. The page title is "yottadb - Rust - Mozilla Firefox <2>". The left sidebar lists various documentation sections: Re-exports, Modules, Macros, Constants, Functions, and a section titled "Crates" which contains a link to "yottadb". The main content area starts with a large YottaDB logo and a brief introduction: "YottaDB is a NoSQL Database suitable for high-performance systems. YottaDB runs in-process, like SQLite, with no need for a daemon. This crate is a Rust wrapper around the C implementation of YottaDB." It then states that there are three major APIs: `craw`, `simple_api`, and `context_api`. A note at the bottom says: "The context_api is recommended for normal use, but the others".

[-] YottaDB is a NoSQL Database suitable for high-performance systems.

YottaDB runs in-process, like SQLite, with no need for a daemon. This crate is a Rust wrapper around the C implementation of YottaDB.

There are three major APIs:

- `craw`, the FFI bindings generated directly by bindgen. These are not recommended for normal use, but are available in case the other APIs are missing functionality.
- `simple_api`, a wrapper around the `craw` API which handles resizing buffers and various other recoverable errors. The simple API also provides a `YDBError` struct so that errors are returned as `Result` instead of an error code.
- `context_api`, which is a wrapper around the `simple_api` that stores the current `tptoken` and an error buffer so you don't have to keep track of them yourself. The reason the `context_api` is necessary is because this crate binds to the threaded version of YottaDB, which requires a `tptoken` and `err_buffer`. See [transaction processing](#) for more details on transactions and `tptokens`.

The context_api is recommended for normal use, but the others

Hello world – Rust



```
#![allow(non_snake_case)]
use yottadb::simple_api::Key;
use yottadb::craw::YDB_NOTTP;

fn main() {
    let err_buffer = Vec::new();
    let mut hello = Key::new("^hello", &["Rust"]);
    hello.set_st(YDB_NOTTP, err_buffer,
                 " こんにちは世界 ".as_bytes()).unwrap();
}
```

Hello world – C, Go, M, Perl, Rust



```
$ ./sayhelloC
$ ./sayhelloGo
$ yottadb -run sayhelloM
$ ./sayhelloPerl.pl
$ ./sayhelloRust
$ mupip extract -format=zwr -label="Hello" -select=hello -stdout
Hello UTF-8
16-JAN-2020 22:18:16 ZWR
^hello("C")="Hello, world!"
^hello("Go")="สวัสดีชาวโลก"
^hello("M")="Приветствую, мир!"
^hello("Perl")="Grüß Gott Welt"
^hello("Rust")="こんにちは世界"
%YDB-I-RECORDSTAT, ^hello:          Key cnt: 5  max subsc len: 13  max rec len: 36  max node len: 44
%YDB-I-RECORDSTAT, TOTAL:          Key cnt: 5  max subsc len: 13  max rec len: 36  max node len: 44
$
```

Database updates
from multiple
languages

Python – Coming soon



<https://gitlab.com/YottaDB/Lang/YDBPython>

Thank you,
Peter Goss!

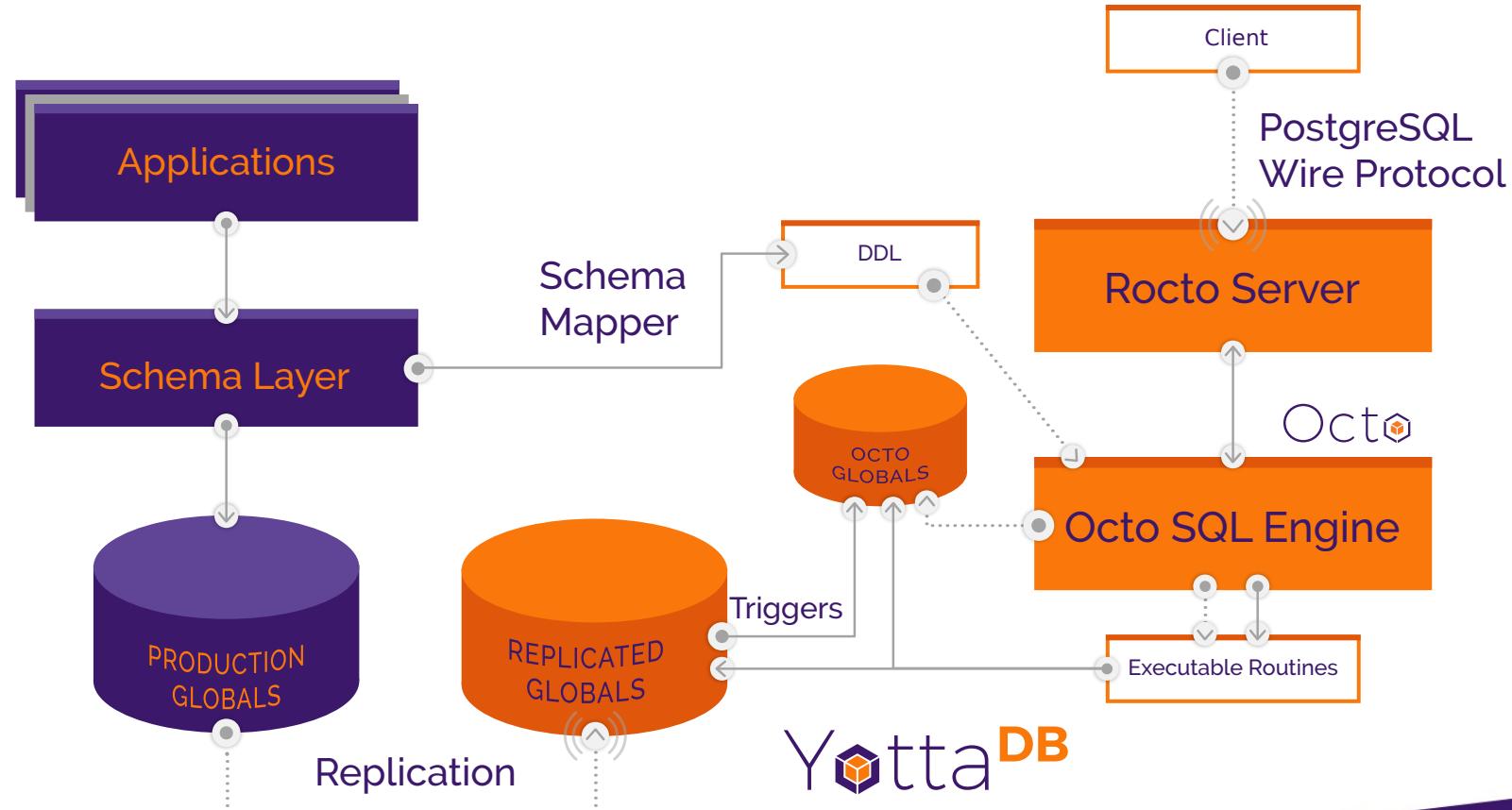
Octo

- Octo is a SQL database engine that whose tables are stored in global variables (YottaDB hierarchical key-value nodes)



Octo Production Configuration

YottaDB



Octo Status & Road Map



- <https://gitlab.com/YottaDB/DBMS/YDBOcto>
- Initial production release Any Day Now™
 - VistA Fileman to DDL mapping tool
 - Read-only access (SQL-92 plus required extensions)
 - PostgreSQL compatible JDBC client support
- To follow: read-write access, other RDMS compatibility

YottaDB – Not Just Languages & Octo



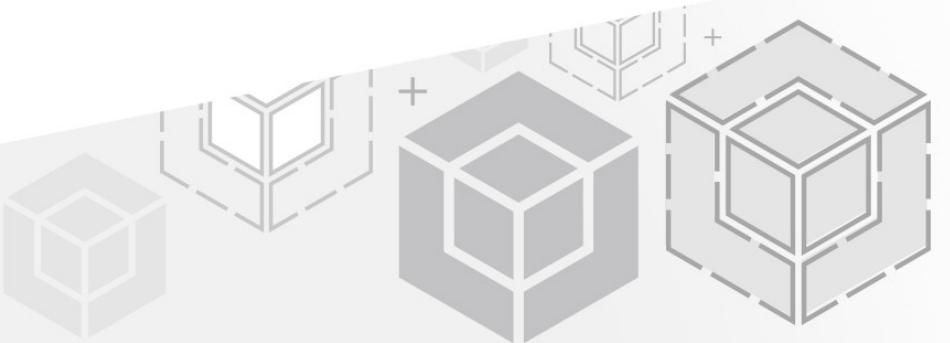
- Linux on ARM (e.g., Raspberry Pi)
 - 32-bit: Raspbian on ARM v6 & v7
 - 64-bit: Ubuntu on ARM v8
- Deployable in Docker containers
- New Acculturation Guide
- And more...

Reaching Out to New Users



- All Things Open, Raleigh, NC (Oct 2019)
- FOSDEM, Brussels, Belgium (Feb 2020)

Currently on hold because of pandemic



Yotta**DB**

Thank You!

K.S. Bhaskar
bhaskar@yottadb.com