Disclaimer

This document is for informational purposes only and is subject to change at any time without notice. The information in this document is proprietary to Actian and no part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of Actian.

This document is not intended to be binding upon Actian to any particular course of business, pricing, product strategy, and/or development. Actian assumes no responsibility for errors or omissions in this document. Actian shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. Actian does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.
Under the Hood Guided Tour of Zen Embedded DB 13.1

Desmond Tan
Director Product Engineering
PSQL Installations

October 2008 through February 2016
The Internet of Things connects the world around us...

- **Smart Scales:** Track health in outpatients
- **Connected car:** Tracks location, status of car parts
- **Container Tracking:** End-to-end tracking, prevent tampering
- **HealthCare:** Monitor patients at home
- **Vending Machine:** Stock reporting, temperature, shelf life
- **Smart Meter:** Track and control usage
- **Mobile:** Mobile payments
- **Smart Deliveries:** Track parcel, monitor and open garage door remotely on arrival
- **Building Security:** Facial recognition, remote notification
- **Heating and Air Conditioning:** Maximum efficiency using weather predictions and remote control
What we learnt in 2016

1. These “Things” are small, therefore if a database is needed, it needs to be small and embeddable and zero administration.

2. The devices tend to be design using the ARM architecture or running OSes complied using the ARM architecture because of (1).

3. These devices are almost incapable of handling relational data structure (SQL) and the data handled will be in various structures.

4. IoT application developers are most likely into rapid development and like programming languages like Python.
• Support Windows IoT Core on x86 and arm32
• Support Windows Nano Server
• Raspbian on arm32
• Support Linux arm 32 and arm64
• Support Docker for Windows

• Additional SQL syntax
• Reporting Engine
• Schema export in PCC
• Domain authorization for secured databases
• LIKE operator in Btrieve extended ops

• SQL can use ISR’s in addition to ACS’s
• JDBC 4
• Btrieve API 2 for more languages
• Cache pre-load & purge for all engines
New Platforms Support

- For 32-bit single-board computers
  - Windows IoT Core x86
  - Windows IoT Core arm32 (armhf)
  - Raspbian arm32 (armhf)
- Equivalent to Server Engine, so that local storage is always available
- Up to 10 user count
- Windows Nano Server
- Supports NanoServer 2016 for use in Windows containers
- Linux on arm 32 and arm 64
- Supports Docker for Linux
Docker for Windows

- PSQL supports NanoServer 2016 for use in Windows containers
  - Can also use Windows Server Core and silent MSI installs
- You can install PSQL Client as part of containerizing an app
- You can install a PSQL Engine and then add a license layer as part of deploying turnkey systems
  - We recommend that data files be external to the engine container

FROM microsoft/nanoserver
RUN Powershell -Command Expand-Archive PSQL.zip
RUN PSQL\Install-PSQL.ps1 Client

Additional SQL syntax

- SELECT ... LIMIT 20 OFFSET 100
  - An alternative to TOP; goes at the end of the SELECT, after ORDER BY
  - ADO.NET 4.3 provider supports skip() operation
- ORDER BY in expression subqueries
  - Top-level queries and table subqueries already supported
- DROP ... IF EXISTS
  - Works for all database objects, such as views and procedures
Reporting Engine

- SQL engine that proxies for a given Storage Engine
- Intended for read-mostly workloads
- No local storage, so no license required
- When connected to the Reporting Engine, you can do all SQL operations within a database
- Cannot manage databases themselves; use the Storage Engine
- Additional performance counters give view of cache behavior
- Also acts as a Btrieve cache engine
Schema export in PCC

- Export now dumps all metadata to a script
  - Tables, views, stored procedures, UDFs, triggers, security
  - Export logs metadata problems of original database
- Import uses the script to recreate the database, and optionally, empty data files
  - Import logs issues with re-creating metadata
- Validation compares the new database to the original to find differences
- Can be used to migrate from v1 to v2 metadata
Schema export in PCC
Schema export in PCC

[5:02:01 PM] [INFO] Starting Schema Export...
[5:02:03 PM] [INFO] Starting Tables...
[5:02:05 PM] [WARN] One or more tables use the collation sequence 'UPPER.ALT'. This file must be copied to the import database location before importing.
[5:02:05 PM] [DETAILS] SET TRUENULLCREATE = OFF; CREATE TABLE "ALUVAT" USING 'ALUVAT.DAT' ("LupaNumero" VARCHAR(40) CASE, "AsiakasNumero" INTEGER, "Sukunimi" VARCHAR(40), "Etunimet" VARCHAR(40), "Toimintatapa" SMALLINT, "TehtaanMerkkiJaMalli" VARCHAR(40), "Kaliiperi" VARCHAR(10), "Varaaja" VARCHAR(9), "Syntymaika" VARCHAR(20), "Garjanro" VARCHAR(40), "Voimassa" DATE, "Antopvm" DATE, "Yrtanomaisein" VARCHAR(40) COLLATE 'UPPER.ALT', "Suljettu" DATE, "VoimassaTolstaiseksi" LOGICAL, "KauppiasLupa" LOGICAL, "varalla" VARCHAR(97) ); SET TRUENULLCREATE = ON;
[5:02:05 PM] [WARN] Column ASALETU.Tuote has the same offset as another column that is not type BIT
[5:02:05 PM] [DETAILS] SET TRUENULLCREATE = OFF; CREATE TABLE "ASALETU" USING 'ASALET.DAT' ("AsiakasNumero" INTEGER, "Ryhmatoikko" LSTRING(11), "Tuote" LSTRING(19) /* Column ASALETU.Tuote has the same offset as another column that is not type BIT */, "RyhmaNumero" CHAR(5), "Hinta" DOUBLE, "Prosentteina" LOGICAL, "Alkaa" CHAR(8), "Loppuu" CHAR(8), "Hacm" LSTRING(21), "Tyyppi" UTINYINT, "Maara" DOUBLE, "AsiakasTyyp" UTINYINT, "KplAlo" LOGICAL, "OstoHinta" REAL, "KplAtensa" LOGICAL, "Vainatkilla" LOGICAL, "VaraBoolean" LOGICAL, "HintaAlv0" DOUBLE, "varalla" VARCHAR(8), "AlkaaKlo" TIME, "LoppuuKlo" TIME, "Varaalla" CHAR(20) ); SET TRUENULLCREATE = ON;
[5:02:05 PM] [WARN] Column ASCOLLECTOR.Kuittinro has an incorrect size of 0
[5:02:05 PM] [DETAILS] SET TRUENULLCREATE = OFF; CREATE TABLE "ASCOLLECTOR" USING 'ASCOLLECTOR.DAT' ("Asiakas" INTEGER, "Varausid" VARCHAR(40) CASE, "Lisasyklo" DATE CASE, "PeruttuPvm" DATE, "peruttuKlo" DATE, "Lahde" VARCHAR(9), "Kuittinro" CHAR(0) /* Column ASCOLLECTOR.Kuittinro has an incorrect size of 0 */ ); SET TRUENULLCREATE = ON;
[5:02:05 PM] [INFO] Comp[5:05 PM] INFO] Completed Tables
[5:02:05 PM] [INFO] Starting Procedures...
[5:02:06 PM] [INFO] Completed StoredProcedure...
[5:02:06 PM] [INFO] Starting Views...
[5:02:06 PM] [INFO] Completed Views
[5:02:06 PM] [INFO] Starting Triggers...
[5:02:07 PM] [INFO] Completed Triggers
[5:02:07 PM] [INFO] Schema Export Completed
Domain authentication for secured databases

- New method for securing databases
- PSQL has Master user and Groups (roles)
- PSQL grants rights to groups
- AD has Users and group membership
Domain authentication for secured databases
<table>
<thead>
<tr>
<th>JDBC 4.0</th>
<th>LIKE operator for Btrieve extended ops</th>
<th>SQL can use ISR’s in addition to ACS’s</th>
</tr>
</thead>
</table>
| • Frameworks can query JDBC 4.0 metadata about driver  
  • Compiled for Java 6, so no 4.1 support  
  • .jar file name unchanged | • Btrieve extended ops now support a LIKE comparison operator  
  • Behavior is similar to the SQL LIKE operator | • COLLATE can now take ISR’s, including Unicode ICU collations  
  • CREATE TABLE t1 (c1 CHAR(20) COLLATE 'u54-msft_enus_0')  
  • SELECT * FROM t2 ORDER BY c2 COLLATE 'PVSW_ENUS01252_1' |
New Btrieve API for more languages

- All the Btrieve operations, with focused entry points
- Compatible with SWIG, so you can generate bindings in popular scripting languages

```python
sys.path.append("C:\\Program Files\\Actian\\PSQL\\bin")
import btrievePython

myFileName = "myData.btr"

myFileHandle = btrievePython.BtrieveFile()
rc = btrieveClient.FileOpen(
    myFileHandle,
    myFileName,
    None, # no owner name
    btrievePython.Btrieve.OPEN_MODE_NORMAL)

keyFormat = "<B"
integerValue = int(sys.argv[1])
key = struct.pack(keyFormat, integerValue)
recordFormat = "<BHd60s"
record = struct.pack(recordFormat, 0, 0, 0, b"
")

rc = myFileHandle.RecordRetrieve(btrievePython.Btrieve.COMPARISON_EQUAL,
                                btrievePython.Btrieve.INDEX_1, key, record)

record = struct.unpack(recordFormat, record)
print("record is :", record[0], record[1], record[2],
     (re.sub("b\\\", ",", (re.sub(r'\\.*', ",", str(record[3]))))))

rc = btrieveClient.FileClose(myFileHandle)
rc = btrieveClient.Reset()
```
Cache pre-load & purge for all engines

- Warm-up” or clean out the PSQL engine’s runtime cache
- `butil -cache D://data//popularFile.mkd`
- `butil -purge D://data//occasionalFile.mkd`
- Affect client runtime cache when executed on a Reporting Engine machine or on a Client with Cache Engine enabled
Thank you!
Lunch - Terrace Restaurant
Time: 11:45-12:45
#hybriddataconference