



FIDELITY NATIONAL
INFORMATION SERVICES

PIPTM Infrastructure Project

High-End Database Application Development Platform

GT.M+PIP Technology Exchange
October 16, 2008

Pat Kelly
patrick.kelly@fnis.com

Agenda

- Introduction
- Architecture
- Demonstration
- Considerations
- Questions



Introduction

- PIP is the “framework” for FIS Profile
 - Real-time core processing application for financial institutions around the world.
- Application development platform based on GT.M
 - High-performance, ACID, 24x7 DB capabilities
- PIP provides RDB capabilities for MDB
 - Layer of abstraction
- Abstraction provides flexibility
 - All Profile V7.* application code is “database independent”

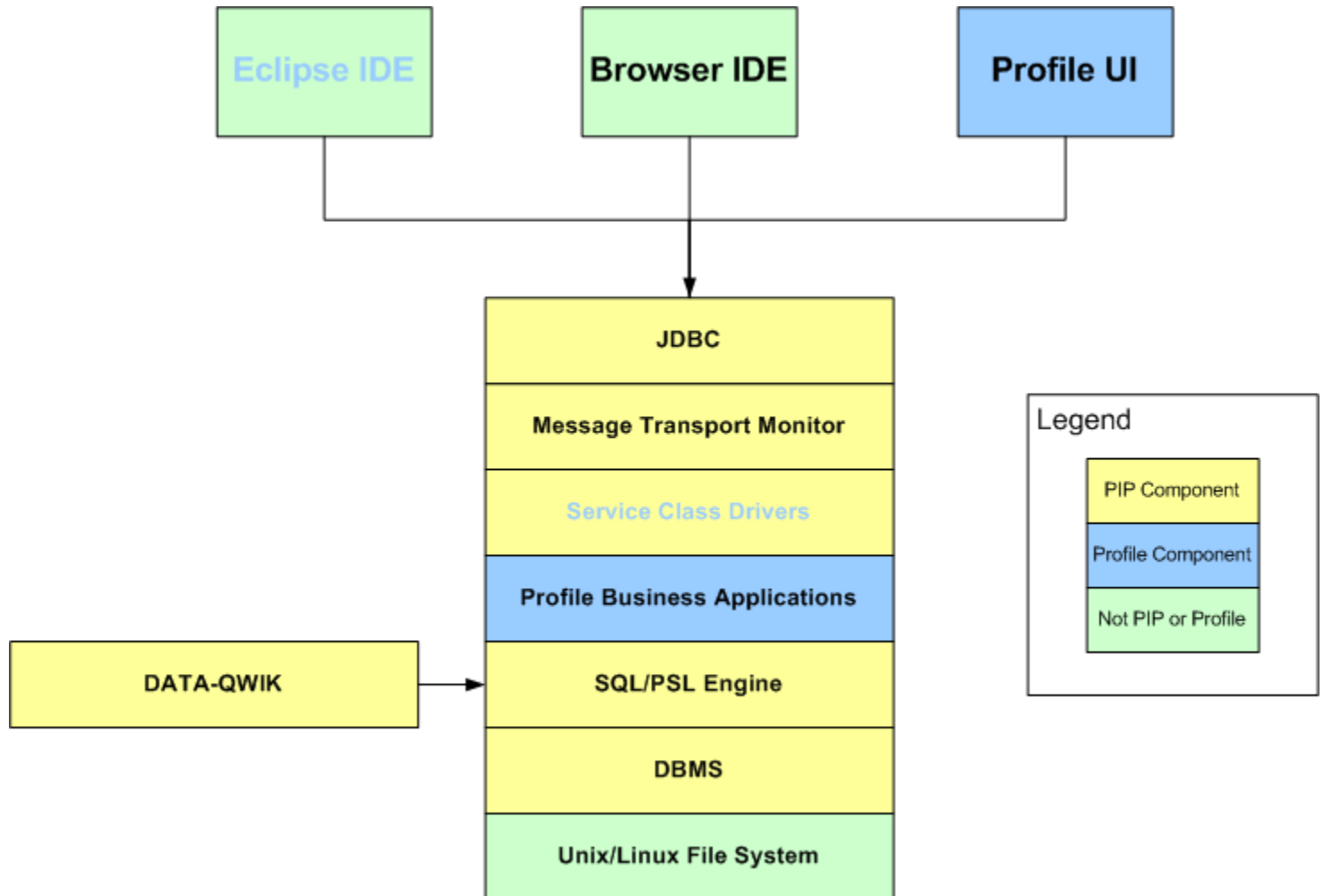


What's in the Box?

- Schema definition & mapping tools
 - More than just global mapping
- Support for multiple program types
 - Triggers, batch definitions, procedure definitions
- Screen and report generators
 - Presentation-based coding
- Profile Scripting Language (PSL) Compiler
- JDBC Interface & Eclipse Plug-ins



Profile & PIP Architecture



DATA-QWIK

- PIP's character-based interface
- Provides menu options and native UI for creating and maintaining PIP artifacts
 - Schema: Tables, Indices, Journal and FK definitions
 - Process: Triggers, Procedures and Batch Definitions
 - Presentation: Screens and Reports
- Self-defining
- Extensible, time-saving, and flexible tool to easily construct and manage application code

DATA-QWIK (Cont.)

- Table mappings for DATA-QWIK elements
 - Tables: DBTBL1*
 - Triggers: DBTBL7*
 - Indices: DBTBL8
 - Journals: DBTBL9*
 - Procedures: DBTBL25*
 - Batch Definitions: DBTBL33*
 - Screens: DBTBL2*
 - Reports: DBTBL5*

Profile Scripting Language (PSL)

- Object-oriented version of GT.M (M++)
- Enables transition from legacy Mumps applications to modern coding paradigms (OO, RDB)
- PSL Compiler provides abstraction between business logic and underlying DBMS
 - PSL procedure compiles into M routine which is then compiled (linked) by GT.M into native machine code
- PSL may be compiled for different DBMS (GT.M and Oracle currently supported)

PSL (Cont.)

- PSL

```
type RecordCRCD crcd = Db.getRecord("CRCD","CO='SCA',CRCD='USD'")
```

- Compiled M for GT.M

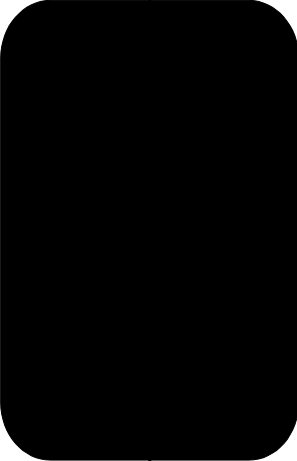
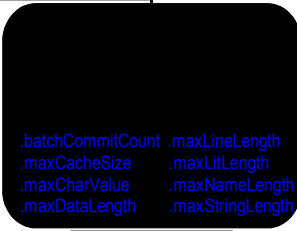
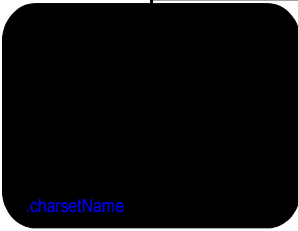
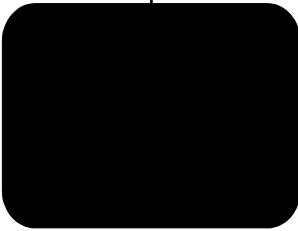
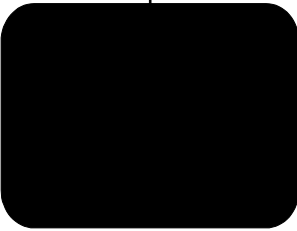
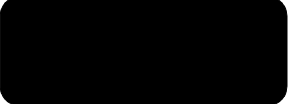
```
N crcd S crcd=$$vDb12("SCA","USD")
...
vDb12(v1,v2) ; voXN = Db.getRecord(CRCD,,0)
;
N crcd
S crcd=$G(^CRCD(v1,v2))
I crcd="", $D(^CRCD(v1,v2))
I $T S $ZE="0," _ $ZPOS_ , %PSL-E-RECNOFL,,CRCD", $SEC=", U1001,"
Q crcd
```

- Compiled M for Oracle

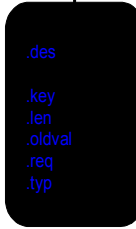
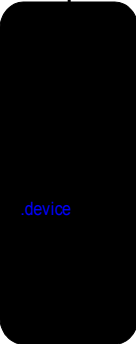
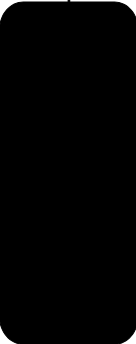
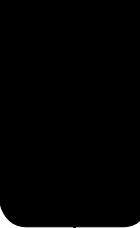
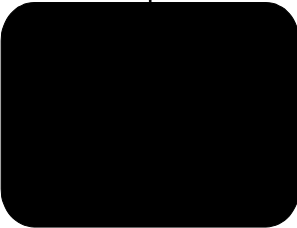
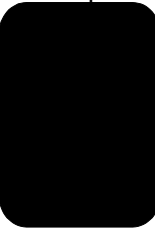
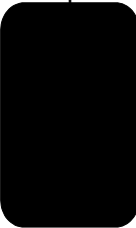
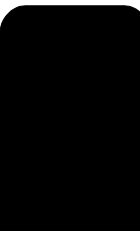
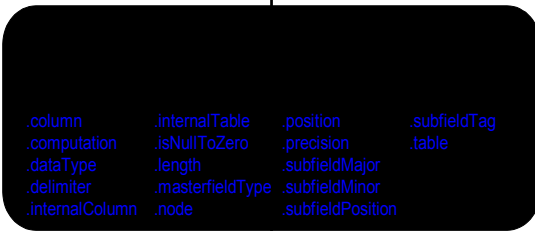
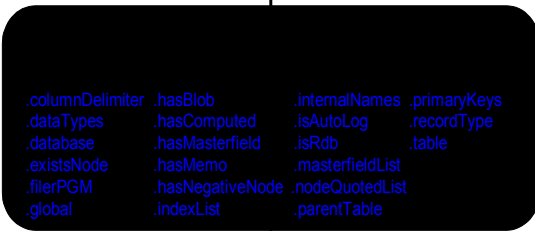
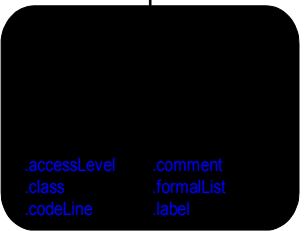
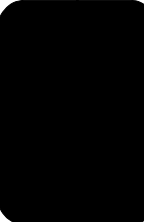
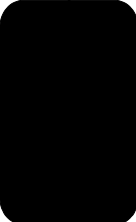
```
N crcd S crcd=$$vDb12("SCA","USD")
...
vDb12(v1,v2) ; voXN = Db.getRecord(CRCD,,0)
;
N crcd N vData,vEr,vRm
S vEr=$$EXECSP^%DBAPI(0,"T_CRCD$SEL_ALL",v1_$C(1)_v2_$C(1),2,$C(124)...
I vEr=100
I $T S $ZE="0," _ $ZPOS_ , %PSL-E-RECNOFL,,CRCD", $SEC=", U1001,"
Q crcd
```

PSL Class Hierarchy Profile 7.0 and higher

Classname
.method()
.property
.readOnlyProperty



Number
.byte()
.char()
.fnumber()
.random()
.roundCur()
.roundDec()
.toString()
.zero()



Demonstrations

- DATA-QWIK and Eclipse Plug-in examples



Considerations

- Separation from Profile is not complete
 - Code and data in PIP that belongs in Profile and vice-versa
 - Documentation not separated
- More stand-alone testing needed
 - PIP tested thoroughly via both GT.M and Profile but not by itself
- Security
 - PIP is a development environment
 - Application security should be provided by the app
- Skillsets required
 - Mumps, PSL (OO), SQL, RDB, O/S
 - Expert-friendly

Questions?

- Thanks for attending!

