DTP in Your Language

John Graham
Eclipse Data Tools Platform (DTP) Project
PMC Chair
Overview

- DTP 1.5 (Europa) summary
- DTP 1.5.1 update
- DTP 1.6 (Ganymede) plans
- DTP usage in language environments
  - Java
  - Ruby
  - C
- Conclusion
DTP 1.5

- With Europa, following DTP 1.0 in December 2006
- Re-organized features to make it easier
  - Users: Choose data sources you work with
  - Extenders: Choose data sources and frameworks
- Enablement
  - Sybase SQL Anywhere (ASA) support
  - HSQLDB
  - ODA Web Services Driver
- Many other features & bug fixes
- See http://www.eclipse.org/datatools/nn/nn_dtp1.5.html
DTP 1.5.1

- Released two weeks ago, September 27 with Europa
- Statistics
  - 94 bugs resolved
  - 34 builds
  - 3 months of work
  - Last minute rush: BZ204632
- DTP 1.5.1 is now the current DTP release
  - DTP 1.5 is superceded
  - DTP 1.6 is not ready yet (but see next…)
- (DTP 1.5.2 in February, 2008)
DTP 1.6

- Next major release, with Ganymede (Eclipse 3.4)
- Major feature work underway
  - SQL Query Builder
  - Refactoring of some Enablement plug-ins
- Nightly builds available, first milestone (M3) in early November
- Project documentation
- Still time for contributions and ideas!
  - Yes, it happens! BZ197806 (MaxDB)
  - And you too…
Demo

DTP Europa
DTP in Your Language

- Use case
  - Make database programming easier in Eclipse by using DTP
  - Multi-language support
  - Extensibility
- Today, we’ll consider DTP with
  - Java
  - Ruby
  - C
- Concentrating on connection code in this demo
Demo

DTP in Your Language
Challenges

- DTP uses Java for database access
  - Natural, because plug-ins are written in Java
  - But, biases the design toward Java
- Ruby
  - Using DBI for example
  - Need to map to database by vendor
- C
  - Higher level of abstraction than individual drivers?
  - Database-specific code required
- ➔ Need a second tier of extensibility?
Challenges, cont’

- Integration is at surface
  - IDE tools in Eclipse are various and not easy to extend
  - Nice features, like drag-and-drop not handled
- How to handle drivers in each language (Java easy)
- Connection code doesn’t react to changes
  - No automatic update or link by reference
  - No refactoring support
  - Full solution would require far more infrastructure than currently exists in any Eclipse language IDE
- Is there hope at all for a generic solution?
Summary

- Seems like a nice idea, but faces serious challenges
- Could have one-off solutions, but this invites duplication
- Similar, but not as severe, as Enablement
  - If DTP doesn’t work with your database it is useless to you
  - If DTP doesn’t work directly with your language IDE, it is a shame, but not a disaster (?)
  - Needs similar “enablement” support, and we have a project structure for this
- Potential area of future development
Questions?

Thank you!