Data Tools for Rich Clients

John Graham
Eclipse Data Tools Platform PMC Chair
Staff Software Engineer
Sybase, Inc.

From Proposal to Project

- Sybase proposes DTP top-level project in February, 2005
- EclipseCon 2005: Community building
  - Contributions from Sybase
  - Contributions from WTP/IBM
  - Contributions from BIRT/Actuate
  - Contributions from objectNation
- Consolidation exercise during spring 2005
- Creation review in June, 2005
- Initial project activities during summer, 2005
  - Planning
  - Initial code contributions

Initial Milestones

- Initial DTP plan
  - Two milestones in 5Q4
  - Two milestone in 6Q1
  - Release 0.7 just before EclipseCon 2006
- Callisto project
  - DTP joins 9 other Eclipse projects in this effort
  - Sets another DTP release in June 2006
  - Sets the tone for much of DTP activity during 6Q1 and 6Q2

What does DTP Provide?

- Following Eclipse spirit
  - Extensible frameworks
  - Exemplary tool
- Emphasis on frameworks
  - Vendor neutral
  - Extensible
- Projects
  - Model Base: EMF-based domain components (e.g. SQL)
  - Connectivity: Core connection frameworks and tools
  - SQL Dev Tools: Frameworks and tools for SQL

Model Base Project

- Uses the Eclipse Modeling Framework (EMF)
- Model: Think *EMF* not *Data Modeling*
- Provides DTP domain models
  - SQL
  - Database Definition
  - SQL Query
  - SQL XML Query
- Generic support for standards
- Specialize for vendor specific support

Connectivity Project

- Frameworks
  - Driver management
  - Connection management
  - Open Data Access (ODA)
- Tools
  - Data Source Explorer (DSE)
  - ODA flat-file and design UI

SQL Development Tools Project

- Frameworks
  - Routine editor
  - Routine debugger
  - Execution plan
  - SQL Query Parser
- Tools
  - SQL editor
  - Results view

Download Types

- Supports Eclipse 3.1 and Eclipse 3.2 platforms
- Requires appropriate versions of
  - EMF
  - GEF (for SQL Dev)
- Eclipse 3.1 requires additional plug-ins for compatibility
- DTP is available in two main packages
  - Binaries: Everything required to run DTP
  - SDK: Binaries, source code and extender documentation

Component Structure

- SQL Dev requires
  - Connectivity
  - Model Base
- Connectivity requires
  - Model Base
- Model Base requires
  - EMF

DTP in RCP

- RCP is a *range*, not a *choice*
  - The base platform is very simple
  - Can add any number of plug-ins
  - Results in Eclipse SDK functionality
  - Add more plug-ins…
- Can DTP be used in RCP?
  - Yes!
  - But… it depends on what you want

Dependencies: Model Base

- Uses EMF, so you need at least that
  - Brings in *org.eclipse.core.resources*
  - Brings in file system support
- You'll need
  - 12 plug-ins as dependents
  - To get the SQL Model
- But
  - Only 2 additional dependencies
  - To get the other 4 Model Base Plug-ins

Dependencies: Basic Connectivity

- Driver and Connection Management Frameworks
  - 9 dependents and 2 framework plug-ins
- Data Source Explorer (DSE)
  - Adds 5 dependents and the DSE plug-in
  - Crosses the line into “IDE”
- Database Connectivity
  - Adds 13 dependents + 7 DTP plug-ins
- Open Data Access (ODA)
  - 10 dependents and 2 ODA for basic framework
  - 12 dependents and 5 ODA for design time support

Dependencies: SQL Dev Tools

- SQL Development Tools
- Designed for IDE use in 0.7
- Depends on
  - Models
  - Connectivity
- Perhaps isolate components (SQL Editor) later
- SQL Query Parser
  - Depends only on Model Base
  - Can be used outside IDE environment

Demonstration & Code Walk-through

Advanced Topics

- Connection Environments
  - Occasionally connected
  - Location dependency
- Data Security
  - What does DTP store?
  - How to protect user data
- Extending DTP
  - Model specializations
  - Extension points
  - Internal classes (gasp!)

So What?

- There are lots of other choices
  - Commercial
  - Open source
  - Even Eclipse plug-ins
- A lot of them have more functionality than DTP
- A lot of them have wider database support than DTP
- Most have a longer history than DTP
- Some are very good, and have a strong reputation

DTP Advantages

- Open source under EPL
- Based on frameworks
  - Built for extensibility
  - Samples of how to extend
- Built for Eclipse
  - Completely integrated with Eclipse
  - RCP enabled
  - Following Eclipse best practices
  - Working closely with rest of Eclipse ecosystem
- DTP is not just about databases
- Support from major vendors

Future Directions

- Incremental feature upgrades between 0.7 and 1.0
- Post 1.0
  - Visual Query Builder
  - Extensible SQL Parser framework
- Community interests
  - Database refactoring
  - Administrative tools
  - …

Call to Action

- Connection Profiles for specific targets
  - Generic, simple interface to implement
  - Extend the reach of DTP
  - Extend the reach of Eclipse tools
  - Work with DTP committers on it
- Proposals for addition components
  - Driven by community interest
  - Build a community
- Use and comment on DTP
  - Bugs (!)
  - Enhancements

Community Connections

- Web: www.eclipse.org/datatools
- Newsgroup at eclipse.org
- Mailing lists
  - PMC
  - DEV
  - Projects
- Various meetings
- Conferences
- Me: john.graham@sybase.com

Questions?

