

SYBASE®



## DATA TOOLS PROJECT

THE DATA TOOLS PROJECT (DTP) PROPOSAL OF ECLIPSE  
SPONSORED BY SYBASE.

THE  
ENTERPRISE.  
UNWIRED.

PROPOSAL

## PROJECT ORGANIZATION

The Data Tools Project (DTP) is a proposed open-source Top Level Project of eclipse.org. The Charter will describe the organization of the project, roles and responsibilities of the participants, and top-level development process for the project.

## OVERVIEW

“Data Tools” is a vast domain, yet there are a fairly small number of foundational requirements when developing with or administering database systems. A developer is interested in an environment that is easy to configure, one in which the challenges of application development are due to the problem domain, not the complexity of the tools employed. Administration of databases, whether by a developer working on an application, or a DBA maintaining or monitoring a production system, should also provide a consistent, highly usable environment that works well with associated technologies.

Such an environment for data management starts with key frameworks designed both for use and extensibility. These frameworks include location and management of database drivers, and configurations for access to particular database instances. Once a connection is successfully made, the next task often is to explore the database—tables, stored procedures, triggers, and so on—making changes as required. Some of these operations might be carried out by GUI actions, others directly through commands. Users—both developers and administrators—typically will create, edit, and test SQL for these commands. Assistance in editing SQL through code completion, formatting, and dialect specialization, greatly enhances productivity. Further, the ability to execute or debug commands, both SQL and stored procedures, rounds out the rapid development process that Eclipse supports so well. Finally, bridging chasms, whether between relational structures, objects, or XML, presents challenges that data management tooling should address. To meet these needs, DTP includes a broad set of core and tooling components, enabling a diverse set of plug-in offerings specific to particular database technologies and supported by the DTP ecosystem.

## PROJECT PRINCIPLES

Founded in the spirit of open-source, community-driven principles guiding eclipse.org itself, this project will concentrate on several key ideals:

**Vendor neutrality:** We intend to provide data management frameworks and tools not biased toward any database or database vendor. Our intention is that DTP be leveraged to provide the Eclipse community with the widest range of data management choices possible. To this end, we seek community involvement in formulating key framework interfaces, so that the largest possible constituency is represented.

**Extensibility:** We recognize both the common need for data management tooling infrastructure and the desire to extend the offerings in new and innovative ways. To support these efforts, our components will be designed for, and make good use of, extensibility mechanisms supported by Eclipse.

**Community Involvement:** Success for DTP, as with other eclipse.org projects, is as much a factor of community involvement as the technical merit of its components. We strongly believe that DTP will achieve its full potential only as the result of deep and broad cooperation with the Eclipse membership-at-large. Thus, we will make every effort to accommodate collaboration, reach acceptable compromises, and provide a project management infrastructure that includes all contributors, regardless of their affiliation, location, interests, or level of involvement. Regular meetings covering all aspects of DTP, open communication channels, and equal access to process will be key areas in driving successful community involvement.

**Transparency:** As with all projects under the eclipse.org banner, key information and discussions at every level—such as requirements, design, implementation, and testing—will be easily accessible to the Eclipse membership-at-large.

**Agile development:** We will strive to incorporate into our planning process innovations that arise once a project is underway, and the feedback from our user community on our achievements to date. We think an agile planning and development process, in which progress is incremental, near-term deliverables are focused, and long-term planning is flexible, will be the best way to achieve this.

## PROJECT SCOPE

The foundation of DTP consists of three functional areas:

- Connectivity
- SQL Development
- Administration

For each of these functional areas, DTP will provide frameworks with well-defined extension points. Upon these frameworks, DTP will build exemplary, extensible tools. Our goal is to provide example implementations targeted at well-known open source runtime environments while enabling an ecosystem of complementary open source and commercial projects to create components specialized to particular databases.

Building on this foundation, DTP will also include the three projects for data-centric development:

- Object/Relational Mapping
- XML/Relational Mapping
- Extract-Transform-Load

We understand that some areas of this proposal may overlap with other efforts in eclipse.org and the community. Examples from current eclipse.org projects include the ‘rdb’ components of Web Tools, and the Extract-Transform-Load components mentioned in BIRT. From the Eclipse membership-at-large, recent work on Hibernate tooling provides one potential Object/Relational Mapping tie-in. We are very interested in working with such groups to insure a consistent, comprehensive, and best-of-breed set of data management tools in Eclipse, regardless of where they are housed.

## PROJECTS

### DATABASE CONNECTIVITY

The Database Connectivity project consists of core frameworks and components for DTP:

#### ■ Driver Management

Access to the appropriate database drivers is a prerequisite for programmatic interaction with databases. The Driver Management Framework (DMF) supplies an Eclipse preference page enabling users to create driver definitions based on supplied templates. A number of database templates are provided in the base installation, and additional templates can be added by component developers contributing to DMF extension points.

#### ■ Connection Profile Framework

The Connection Profile Framework (CPF) is the foundation upon which specific connection types are created. The connection types, called “Connection Profiles” (CP), are contributed to the CPF through extension points. Users then connect to database instances by creating and configuring a CP for that database type. Database-standard configuration parameters, such as

the connection URL, user name, and password, are provided on CP instance creation and stored as secure meta-data for the CP. Database CP allow for host connectivity checks (aka “ping”), connection, auto-connect on CP startup, and disconnect. Further, CP Extensions enable additional functionality and content to be added to a CP. For reuse of CP instance configuration, base export/export functionality is provided by CPF and surfaced in tools such as the Database Explorer (see below). Database CP then become the connection providers through which other DTP tooling accesses database instances.

#### ■ **Sample Implementations of Connection Profiles**

DTP will include a number of database CP, as a means of demonstrating, testing, and serving as examples for further CP development. While the set of database CP has yet to be determined, We will strive to include CP to open-source database offerings whenever possible. With community support, we hope that the number of CP will quickly grow to encompass a large number of databases. As described above, database-specific capabilities can be surfaced as CP extensions, allowing for specialization and presentation of differentiating database functionality directly in that database’s CP.

#### ■ **Database Explorer**

The Database Explorer (DE) is an Eclipse view housing CP instances. From this view, CP capabilities are surfaced, and database content is presented. Users can connect to database instances, browse table, stored procedures, and so on. The type and level of detail for any one instance is constrained only by the CP itself. DE also is a provider of CP instance data to clients, such as drag and drop and API calls. This allows database tools requiring connection management to interact with the DE as a mediator to CP instances.

### **SQL DEVELOPMENT TOOLS**

This SQL Development Tools (SDT) project provides a development environment similar to Eclipse JDT. It leverages the Database Connectivity project to edit, execute, and debug procedural objects (stored procedures, triggers and functions) and SQL statements. The following sections comprise the SDT:

#### ■ **SQL Editor Framework**

The SQL Editor Framework (SEF) is a text-based SQL statement and procedural object editor framework. Coupled with SQL Execution and Debugger Framework, SEF provides visual code generation tooling and grammar extensions. Features found in Eclipse code editors, such as formatting, content assist, dynamic feedback for errors and warnings, correction suggestions, and so on, will be supported.

#### ■ **SQL Execution and Debugger Framework**

The SQL Execution and Debugging Framework (SED) provides core enabling DML functionality for:

- SQL statement execution and debugging
- Procedural object execution and debugging
- Results viewing
- Results export

#### ■ **Sample Implementation for SQL Tools**

Based on SEF and SED, the sample implementation for SQL Tools will provide grammar extensions for chosen SQL dialects and execution support for selected databases. As above with the sample implementation for Connection Profiles, choices of dialect and database support will be based on open-source alternatives, and DTP members will work with the community to increase the number of specializations whenever practical.

## DATABASE ADMINISTRATION

This Database Administration project provides a framework to support generic administration functionality, which will then be tailored to specific databases. As with the other projects above, a sample implementation for selected databases will be provided. A third component of the Data Administration project is tooling for the presentation of SQL Execution plans. Thus, the Database Administration project is divided as follows:

### ■ Database Administration Framework

The Database Administration Framework (DAF) supplies generic database administration capabilities including:

- Database creation and DDL support
- Schema generation, reverse engineering and migration
- Data loading and unloading
- Performance monitoring, analysis and tuning
- Statistics collection

### ■ Sample Implementation for Database Administration

The sample implementation for database administration will enable databases supported by Connection Profiles to be administered using the interfaces and capabilities of the target system. Given the wide range of database-specific administration capabilities, it is expected that different sample implementations will vary more than those for Connection Profiles or SDT. Also, due to the large amount of functionality entailed in full database administration, there likely will be fewer choices supported at first in DTP, but, depending on community involvement, the number could increase over time.

### ■ SQL Execution Plan Inspection Tools

A final component set in the Data Administration project involves SQL Execution Plans. Vital for database tuning, execution plans need to be obtained from a database and presented to the user in a fashion that enables understanding and tuning. DTP will include generic execution plan presentation components, and extension points for specialization of execution plan acquisition and representation as necessary.

## OBJECT/RELATIONAL MAPPING

Object/Relational Mapping (ORM), is important because of the prevalence of object-oriented languages, especially Java, and the impedance in mapping object to relational structures. These factors make robust ORM tools a valuable part of modern database development. The DTP membership will strive to work with the Eclipse membership-at-large in selecting ORM base technologies and creating tools in Eclipse that are integrated with the DTP base.

## XML/RELATIONAL MAPPING

Like ORM, the prevalence and importance of XML in modern application development calls for XML/Relational Mapping (XRM) tools. Built on DTP member experience in this domain, and working with the Eclipse membership-at-large, this project will provide both key and supporting XRM components.

## EXTRACT-TRANSFORM-LOAD

Starting with the simple, but common, developer requirement to load and transfer data between databases, and moving into the complex domain of production Extract-Transform-Load (ETL) capabilities, this project will consist of core components in this domain. We expect that specialization of the extract and load endpoints will be a key area of community plug-in development, based on sample implementations. Translation capabilities will follow the framework/extension model mentioned above and will include core translation functionality in the sample implementation.