



ORACLE®

---

## **EclipseLink: The Only Persistence Platform You Will Ever Need**

Mike Keith  
Oracle Corporation

<http://www.eclipse.org/eclipselink>

ORACLE

## **About Me**

---

- Many years experience in server-side and persistence implementations
- Java and persistence architect at Oracle
- Active member of JCP expert groups, including Java EE 6, EJB 3.1 and JPA 2.0
- Co-wrote Pro EJB 3: Java Persistence API
- Contributor to OSGi, and other groups within OSOA and OASIS
- Presenter at numerous conferences and events

ORACLE

## What will you learn?

---

- **What** the Eclipse Persistence Services Project is
- **Why** this project exists, and why you should care
- **Where** the project is situated in the context of the entire Eclipse ecosystem
- **When** the project can be downloaded and used
- **How** this project can be used, and how you can benefit from it
- **Who** is involved (and how you can be too!)

ORACLE

## What is Eclipse?

---

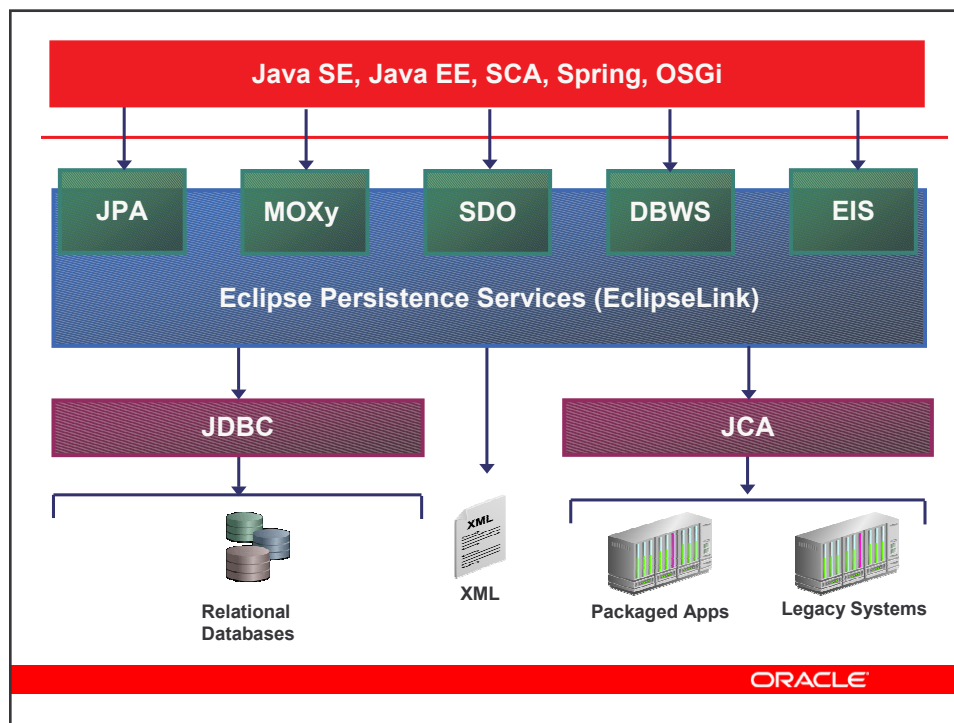
- Eclipse is an open source community
- Eclipse is more than just an IDE
  - Equinox - OSGi
  - Rich Client Platform (RCP)
  - Projects for static/dynamic languages
  - Higgins - Trust Framework
  - BIRT - Business Intelligence/Reporting
  - Embedded/Mobile projects
  - **Persistence Services Project (EclipseLink)**
  - Server-side Eclipse - Eclipse Runtime (RT)

ORACLE

# Eclipse Persistence Services

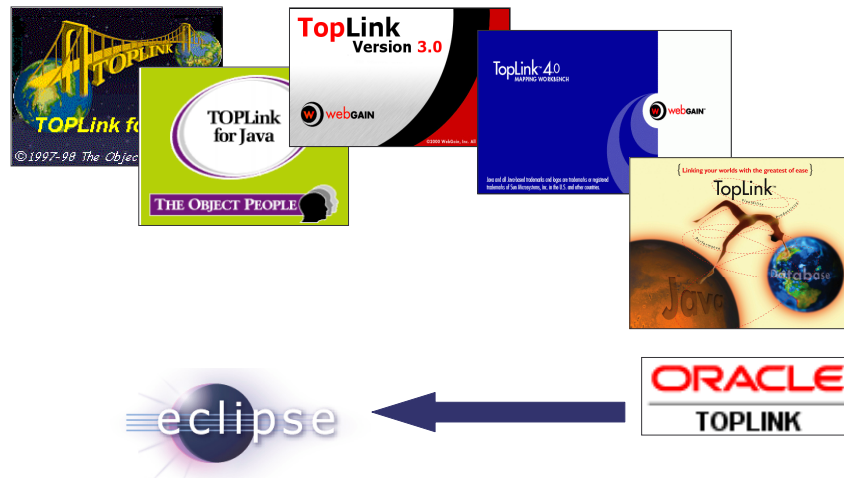
- Nicknamed “EclipseLink”
- Subproject in Eclipse RT
- Comprehensive persistence
  - Eclipse JPA: Object-Relational
  - Eclipse MOXy: Object-XML
  - Eclipse SDO: Service Data Objects
  - Eclipse DBWS: Database Web Services
  - Eclipse EIS: Non-Relational using JCA
- Defining blueprints for OSGi persistence services

ORACLE



ORACLE

# Oracle TopLink



ORACLE

## Why is this project important?

- First comprehensive open source persistence solution
  - Object-Relational and much more
- Based upon product with 12 years of commercial usage
- Shared infrastructure
  - Easily share the same domain model with multiple persistence technologies
  - Leverage metadata for multiple services
- Important part of the Eclipse Ecosystem

ORACLE

## Why Eclipse?

---

- Eclipse has a strong and vibrant community with an effective governance model
- Good reputation for quality
- Interest from within the Eclipse ecosystem
- Oracle has had a positive experience with its existing participation in Eclipse projects
  - Projects lead by Oracle: Dali, BPEL, JSF
  - Other Oracle contributions: WTP and DTP

ORACLE

## Eclipse JPA

---

- JPA 1.0 compliant implementation
- JPA 2.0 features under development
- Java EE, Java SE, Web, Spring, and OSGi
- Any JDBC/SQL compliant database
- Extensible and pluggable
- Schema generation
- Key infrastructure:
  - Caching, Locking, Query Framework, Mapping, ...
- ... plus many valuable advanced features

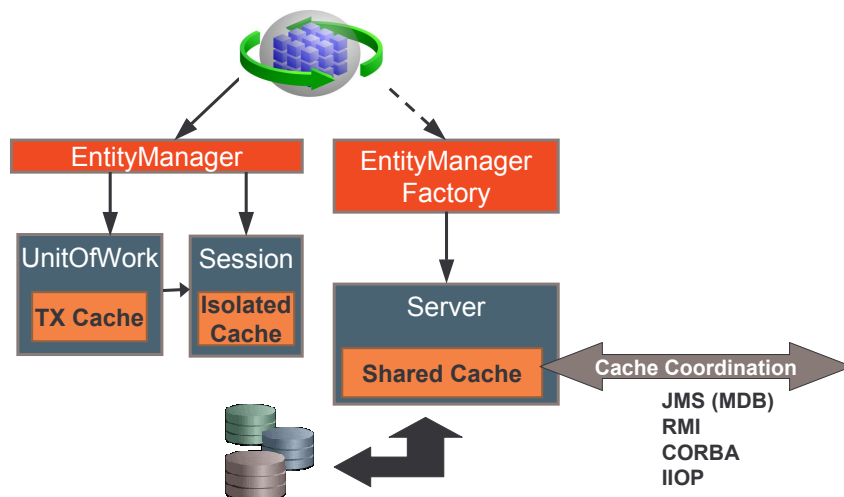
ORACLE

## EclipseLink Caching

- Entity caching
  - L2 objects shared across transactions/users
  - Coordination in a clustered deployment
- Application specific configuration
  - Cache isolation: per client (EM) or shared
  - Cache Type and Size: Weak, Soft-Weak, Full, None
  - Expiration/Invalidation
    - Time to live, Time of day, API
  - Coordination (cluster-messaging)
    - Messaging: JMS, RMI, CORBA, RMI-IIOP, ...
    - Mode: SYNC, SYNC+NEW, INVALIDATE, NONE

ORACLE

## Caching Architecture



ORACLE

## Configuring the Cache

---

- Default: objects read are cached and trusted
- Configuration by entity type important
  - Volatility of data
  - Shared usage of data
- Configuration Parameters
  - Cache isolation, type, size, expiry, coordination
  - Refreshing
    - By query (use-case) or descriptor (always)
- Locking is the only way to avoid potential data inconsistency in concurrent write scenarios

ORACLE

## Locking

---

- Prevent data corruption !!!
- Java Developers think of locking at the object level
- Databases may need to manage locking across many applications
- EclipseLink can respect and participate in locks at database level
  - Optimistic: Numeric, Timestamp, All fields, Selected fields, Changed field
  - Pessimistic: Blocking or timeout-based

ORACLE

## Query Framework

---

- Queries can be defined using
  - Entity Model: JPQL, Criteria, Query-by-example
  - Database: SQL, Stored Procedures
- Customizable
  - Locking, Cache Usage, Refreshing
  - Optimizations: Joining, Batching, parameter binding
  - Result shaping/conversions
- Static or Dynamic
  - Stored Procedure support

ORACLE

## EclipseLink JPA Extensions

---

- Extensions applied using annotations or XML
- Mappings
  - @PrivateOwned, @JoinFetch
  - @Converter, @TypeConverter, @ObjectTypeConverter
- @Cache
  - type, size, isolated, expiry, refresh, cache usage, coordination
  - Cache usage and refresh query hints
- @NamedStoredProcedureQuery
  - IN / OUT / INOUT parameters, multiple cursor results

ORACLE

## EclipseLink JPA Extensions

---

- Locking
  - Non-intrusive policies @OptimisticLocking
  - Pessimistic query hints
- JDBC Connection Pooling
- Logging: Diagnostics, SQL, Debugging
- Weaving for lazy fetch and change tracking
  - Dynamic and Static
- Customization
  - Entity Descriptor: @Customizer, @ReadOnly
  - Session Customizer

ORACLE

## Mapping Extensions

---

```
@Entity
@Cache(type=SOFT_WEAK, coordinationType=SEND_OBJECT_CHANGES)
@OptimisticLocking(type=CHANGED_COLUMNS)
@Converter(name="money", converterClass=MoneyConverter.class)
public class Employee {
    @Id
    private int id;

    private String name;

    @OneToMany(mappedBy="owner")
    @PrivateOwned
    private List<PhoneNumbers> phones;

    @Convert("money")
    private Money salary;
    ...
}
```

ORACLE

## Database Platform

---

- Native SQL (dialect) support with custom operators
- Stored Procedure & Function
- Extensible Advanced Data Types support (Struct)
- Database Security
  - Oracle DB's VPD/OLS and Proxy Authentication
- Configurable value return from write
- Supported platforms (default = Auto)
  - MySQL, Oracle, Derby, DB2, Sybase, SQLServer, TimesTen, PostgreSQL, SQLAnywhere, HSQL, Informix, ...

ORACLE

## Server Platform

---

- Simplified configuration and mediator for host container environment
- Enables
  - Direct JTA integration
  - Data Source/JDBC connection un-wrapping
  - JMX MBean deployment
  - Logging integration
- Current Server Platforms
  - SunAS/GlassFish, OracleAS/OC4J, WLS, WAS, JBoss

ORACLE

## Performance and Tuning

---

- Highly configurable and tunable
  - Principle: minimize and optimize database calls
  - Enable application specific tuning
- Flexibility allows efficient business models and relational schemas to be used
- Leverages underlying performance tuning features
  - Java, JDBC and the underlying database technology

ORACLE

## EclipseLink JPA Configuration

---

- JPA (portable)
  - persistence.xml with EclipseLink properties
  - Mapping: Annotations and/or orm.xml
  - Query hints
- EclipseLink
  - Sessions Configuration (sessions.xml)
  - Mapping using XML or Code
- Eclipse JPA
  - JPA + EclipseLink configurations options
  - EclipseLink annotations

ORACLE

## Eclipse MOXy

---

- Provides complete Object-XML mapping
  - Allows developers to work with XML as objects
  - Efficiently produce and consume XML
  - Document Preservation
- Supports Object-XML standard - JAXB
  - Provides additional flexibility to allow complete control on how objects are mapped



ORACLE

## Eclipse MOXy Benefits

---

- Rich set of mappings providing complete control and flexibility to map objects to any XSD
  - Direct, composite object, composite collection, inheritance, positional, path, transformation ....
- Development Approaches
  - Model + Annotations → XSD
  - XSD → Model + Annotations
  - Model + Mappings(Annotations or XML)
- Supports any JAXP compliant parser
  - SAX, DOM, StAX
- Visual Mapping support using Workbench

ORACLE

## Eclipse MOXy: JAXB

---

```
JAXBContext ctx =
    JAXBContext.newInstance(classes);
Marshaller marshaller = ctx.createMarshaller();

Customer customer = new Customer();
customer.setFirstName("William");
customer.setLastName("Gibson");

marshaller.marshal(customer, System.out);
```

In jaxb.properties file:

```
javax.xml.bind.context.factory =
    org.eclipse.persistence.jaxb.JAXBContextFactory;
```

ORACLE

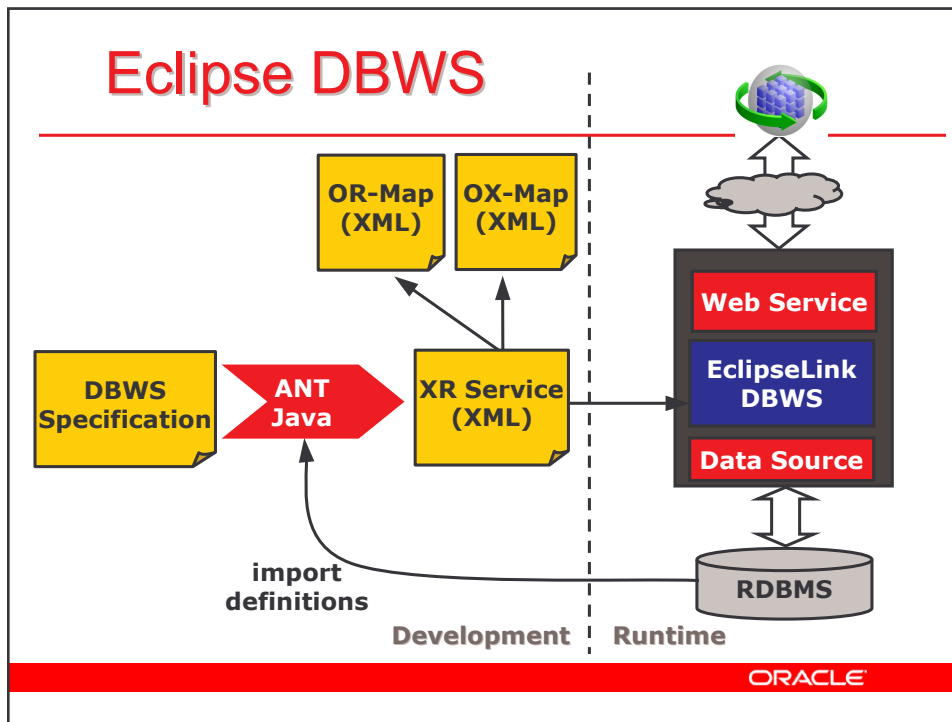
## Eclipse DBWS

---

- Simplified and efficient access to relational data through Web Services
- Relational-to-XML data translation
- Minimal configuration with dev utilities to retrieve metadata and generate/package Web Service
- Developers can fully customize the database access and XML mapping of the data
- Ideal for usage within SOA/SCA

ORACLE

## Eclipse DBWS



## Eclipse SDO

- What can you do?
  - Marshall/Unmarshall objects to/from XML
  - Define Types/Properties programmatically or derive from XSD
  - Generate JavaBean classes from XSD
  - Advanced mapping support for greater flexibility
- Why would you use it?
  - Schema/Structure unknown at compile time
  - Declarative metadata based tools/frameworks
  - XML-centric applications, need open content support
  - Dynamic content user interfaces

## Eclipse EIS

---

- Provide persistence support for non-relational data stores using Java EE Connector Architecture (JCA)
- Mapping interaction inputs and outputs to persistent domain model
  - XML mapping leveraging Eclipse MOXy
  - Common Client Interface (CCI) mapping
- Visual mapping Workbench support
- Out of the box support for:
  - MQSeries, OracleAQ, Sun JCA, XML Files

ORACLE

## EclipseLink and OSGi

---

- Work with OSGi EEG (Enterprise Expert Group) to define OSGi persistence services blueprint
- EclipseLink obtainable as OSGi bundles
- Show through examples how to leverage within an OSGi solution
- Address technical challenges as a community

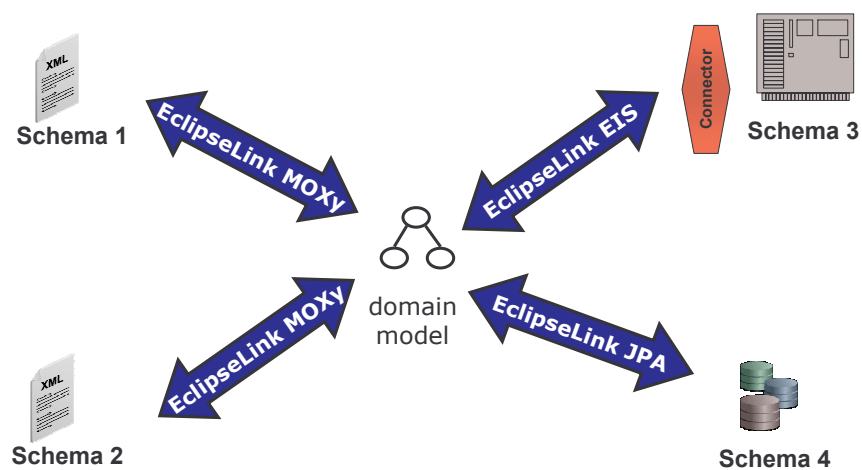
ORACLE

## Combining Services

- Metadata based approach allows the same domain model to be mapped with multiple persistence services
  - Supports usage within Web Services / SOA / SCA
  - Domain model can be shared between persistence services (JPA, MOXy, EIS)
  - Transformations are bidirectional:
    - Unmarshall XML to objects and then persist
    - Marshall persistent objects to XML

ORACLE

## Common Domain Model



ORACLE

## EclipseLink and Spring

---

- Eclipse JPA
  - Container integration
  - Java SE Persistence API
  - DAO Template approach
- EclipseLink Native ORM Template
- Eclipse MOXy
  - Direct, Spring WS, Spring Remoting, ...
- and many more possibilities...
  - Spring Batch, Spring DM, ...

ORACLE

## Eclipse JPA and Spring

---

```
@Repository
@Transactional
public class EntityManagerClinic implements Clinic {

    @PersistenceContext
    private EntityManager em;

    public Collection<Owner> findOwners(String lastName)
        throws DataAccessException {
        Query query =
            em.createNamedQuery("Employee.findOwners");
        query.setParameter("lastName", lastName + "%");
        return query.getResultList();
    }
    ...
}
```

ORACLE

## Part of the Eclipse Ecosystem

---

- Provide an Eclipse persistence solution easily consumable by any project
  - Storage of metadata in RDBMS, XML, EIS
  - XML Messaging infrastructure
- Eclipse Projects
  - Dali JPA Tooling Project
  - Teneo to use EclipseLink for EMF model persistence
  - Maya for storage of deployment configuration
  - SOA Project for EclipseLink SDO
  - Eclipse Runtime Project

ORACLE

## Where are we going?

---

- 1.1.0 just released
- Monthly milestones
- Continued Spring framework support
- Specifications: JPA 2.0, JAXB 2.0, SDO 2.1
- JSR 317 - JPA 2.0 Reference Implementation
- OSGi EEG - JPA Reference Implementation
- Data Access Service (DAS) - SDO with JPA
- Simplified DataMap Access and Dynamic Persistence

ORACLE

## How can you get involved?

---

- Users
  - Try out 1.1.0 release
  - File bug reports and feature requests
- Contributors
  - Contribute to roadmap discussions
  - Bug fixes
- Committers
  - Very interested in growing committer base

ORACLE

## EclipseLink Summary

---

- First comprehensive Open Source Persistence solution
  - Eclipse JPA: Object-Relational
  - Eclipse MOXy: Object-XML
  - Eclipse SDO: Service Data Objects
  - Eclipse DBWS: Database Web Services
  - Eclipse EIS: Non-Relational using JCA
- Mature and full featured
- Much of JPA 2.0 functionality is available

ORACLE

## More Information

---

Project:

[www.eclipse.org/eclipselink](http://www.eclipse.org/eclipselink)

Newsgroup:

[eclipse.technology.eclipselink](mailto:eclipse.technology.eclipselink)

Wiki:

[wiki.eclipse.org/index.php/EclipseLink](http://wiki.eclipse.org/index.php/EclipseLink)

Committer Blog:

[eclipselink.blogspot.com](http://eclipselink.blogspot.com)