

# Eclipse Persistence Services Project

## Creation Review

### Contents:

1. Introduction
2. Scope
3. Out of Scope
4. Code Contribution
5. Mentors
6. Initial Participants
7. Community Support
8. Roadmap

## 1. Introduction

The Eclipse Persistence Services Project (EclipseLink) incubator project's goal is to provide an extensible framework that will enable Java developers to interact with relational databases, XML, and Enterprise Information Systems (EIS). EclipseLink will provide support for a number of persistence standards including the Java Persistence API (JPA), Java API for XML Binding (JAXB), Java Connector Architecture (JCA), and Service Data Objects (SDO).

### 1.1. Project Naming

**Descriptive Name:** Eclipse **Persistence Services** Project

**Acronym:** EPS

**Nickname:** EclipseLink

**URL:** [www.eclipse.org/eps](http://www.eclipse.org/eps)

**Root Package:** org.eclipse.persistence

**Newsgroup:** eclipse.technology.eps

**Mailing list:** eps-dev@eclipse.org

### 1.2. Links

- Creation Review Comments/Votes: [https://bugs.eclipse.org/bugs/show\\_bug.cgi?id=188722](https://bugs.eclipse.org/bugs/show_bug.cgi?id=188722)
- Project Proposal: <http://www.eclipse.org/proposals/eclipselink/>

## 2. Scope

This project will provide persistence services in an extensible framework focusing on performance, productivity, and flexibility. Delivering these services in a standards based approach will also offer consumers the greatest flexibility and enable greater integration potential. The following persistence services and functional areas are defined to be within the initial scope of this project:

## **2.1. Object-Relational**

The Object-Relational Mapping (ORM) service with support for the Java Persistence API (JPA) enables mapping a Java domain model to a relational schema using metadata provided in XML and/or annotations. Applications using this service will have access to the mapping, query framework, transaction support, object caching (including clustering support), advanced database specific capabilities, and many performance tuning and management options. Each of these functional areas are flexible and extensible to address complex application requirements.

## **2.2. Object-XML**

The Object-XML Mapping (OXM) service with support for Java Architecture for XML Binding (JAXB) enables mapping of a Java domain model to an XML Schema (XSD) using metadata provided in XML and/or annotations. This service supports the generation of mappings and XSD from an annotated model, generation of the domain model and mapping from a provided XSD, or meet-in-the-middle mapping with a provided domain model and XSD. The application interface enables the efficient marshalling of the domain model objects into an XML document and un-marshalling of XML documents into domain model objects via DOM or SAX.

This service also provides the necessary infrastructure for other services leveraging XML binding.

## **2.3. Service Data Objects (SDO)**

A Service Data Object (SDO) implementation will be provided. Developers will be able to generate dynamic and static SDO models from an XSD and use these within their application. In addition to the XML binding capabilities defined in the specification additional flexibility is provided by the Object-XML service. This approach provides a solution for applications requiring less coupling to the data structures being accessed and modified. Support for wrapping a Java object (i.e. JPA entity) within a DataObject will be provided to enable the usage of existing domain models with SDO.

A Data Access Service (DAS) will be provided that brings together the Java Persistence API (JPA) with SDO.

## **2.4. Database Web Services (DBWS)**

This service will provide a simple and declarative solution for defining and generating a web service exposing database operations. Using the core Object-Relational and Object-XML capabilities, consumers can leverage default XML representations of their relational data or completely customize the shape of their XML documents.

## **2.5. Enterprise Information Systems (EIS)**

The EIS persistence service enables the usage of data stores through Java Connector

Architecture (JCA) resource adapters. Using XML metadata the interactions and their exchanged data are configured and mapped onto a domain model. The interactions data can be mapped from either the Common Client interface (CCI) or using XML schemas. This usage is intended for non-relational data stores where no JDBC or SQL access is provided.

## **2.6. Ant Tasks and Scripts**

Ant tasks and other scripts to automate schema and/or code compilation/generation and runtime artifact generation are in scope. For example, Ant tasks or scripts to automate both JAXB and SDO schema compilation and schema generation. These Ant tasks or scripts will simplify adoption by making it easy to incorporate EclipseLink into project build processes.

## **2.7. OSGi**

Guidelines will be developed in conjunction with the expert group for persistence usage within OSGi-based applications to access all types of enterprise data through existing standard APIs. The combination of standardized persistence with the OSGi services framework will bring increased portability and improved modularity to applications and confirm the value proposition of merging these two technologies.

## **2.8. Extensibility**

The existing initial code base provides many extensibility points and pluggable policies. One of the goals of this project will be to continue to expose this extensibility as well as expose OSGi extension points.

Examples of extensions that can be built on top of this project or integrated into include:

- Adding support for additional data sources
  - Relation Database Platforms (SQL dialect, stored procedures, custom operators, and custom data types)
  - EIS adapters
- Adding support for additional containers
- Adding additional data services and standards implementations

## **3. Out of Scope**

*The following items are not currently planned as part of the initial EclipseLink project:*

### **3.1. Design Time Tools**

This is a runtime project and will not include any graphical design time tooling. The project team will work with other Eclipse projects to facilitate the delivery of the design time functionality.

### 3.2. EJB 2 CMP and BMP

This project is will deliver persistence services that can be used within any Java EE or SE container. Therefore Enterprise Java Beans (EJB) Container Managed Persistence (CMP) and Bean Managed Persistence (BMP) support is not within scope.

### 4. Code Contribution

This project will be seeded with an initial code contribution based on the Oracle TopLink 11g code stream. The functionality is already publicly available in a technology preview ([available here](#)). Following this initial contribution this project will operate with no implied or explicit dependencies on the Oracle TopLink product.

During the proposal feedback process there were many questions from existing Oracle TopLink and TopLink Essentials users concerning what functionality of Oracle TopLink would be contributed to this project. The intent is to contribute all functionality from the current development stream that falls within the scope defined in this document.

*Note: The Oracle TopLink product will use and extend the output of this project only for the purposes of better supporting the usage of EclipseLink within the Oracle Application Server and SOA Suite.*

For clarity purposes the following functionality will not be contribute to this project:

- Graphical user interfaces (Mapping Workbench)
- EJB 2.1 CMP and BMP support
- OracleAS specific integration support which require binary dependencies
- Current DBWS functionality
- Any deprecated functionality

### 5. Mentors

**Neil Hauge** (Oracle Corporation), Project Lead for Eclipse Dali and WTP PMC Member

**Wayne Beaton** (Eclipse Foundation), PMC member for Technology project

### 6. Initial Participants

Along with the initial code contribution for EclipseLink, Oracle is committed to providing the necessary resources required to ensure that EclipseLink is a viable and active project. Other participants are welcome and being actively sought. It is hoped that committers from many of the interested parties will join the project during incubation as they participate in the EclipseLink

community and begin to adopt the various EclipseLink services in their products and projects.

Peter Krogh, Oracle	Co-Project Lead, Committer	Peter is currently the development manager for Oracle TopLink focussing on object-relational persistence and JPA. Peter has a long history with TopLink in both development management and core development. Peter is also a committer on TopLink Essentials.
Douglas Clarke, Oracle	Co-Project Lead, Committer	Currently the product manager for Oracle TopLink with a focus on requirements gathering and prioritization. Doug has a long history with object-relational persistence and TopLink in both product development and customer consulting and has driven many of the advanced extensions to the framework.
Shaun Smith, Oracle	Ecosystem Development, Committer	Shaun is a product manager for Oracle TopLink focussing on the Object-XML (JAXB) and SDO persistence services. Shaun has a background in consulting using TopLink. Shaun is also the co-lead for the Eclipse Dali project and is a committer on the EMFT Teneo project.
David Twelves, Oracle	Committer, Architecture Council	David is a development manager for Oracle TopLink, focussing on Object-XML related components including JAXB, SDO, EIS, and DBWS. He also leads initiatives targetting improvements to performance.
Mike Keith, Oracle	Committer, Architecture Council	Mike is an architect for Oracle TopLink and Java EE 5. He is also the co-specification lead of EJB 3.0 and a member of the Java EE 5 expert group. Mike will have a principal role with the architecture council ensuring consistency across persistence services as well as integration into other Eclipse and non-Eclipse technologies.
Gordon Yorke, Oracle	Committer, Architecture Council	Gordon is a developer and technical lead on Oracle TopLink and TopLink Essentials (GlassFish). Gordon's development focus is on the core object-relational persistence and JPA implementation. Gordon is the primary developer responsible for the transaction framework.
Tom Ware, Oracle	Committer, Architecture Council	Tom is a developer on Oracle TopLink, and project lead on TopLink Essentials. Tom focusses on the core object-relational persistence and JPA implementation. Tom is the primary developer responsible for JPA deployment.

Blaise Doughan, Oracle	Committer, Architecture Council	Blaise is the team lead and technical architect for Object-XML related components. He is a member of the SDO and JAXB specification expert groups and has focussed his efforts to ensure that TopLink is compliant in these areas.
James Sutherland, Oracle	Committer, Architecture Council	James is a lead architect for the TopLink product and has a long history of development on this product. He is responsible for overseeing and reviewing all architectural changes. He is currently the primary developer for analysing and implementing a wide range of performance improvements.
Mike Norman, Oracle	Committer, Architecture Council	Mike is the lead architect for the new TopLink DBWS functionality. He has a long history with TopLink, previously working on customer engagements in a consultancy role before moving to development.
Andrei Ilitchev, Oracle	Committer	Andrei is a developer on Oracle TopLink and committer on TopLink Essentials. Andrei is the primary developer responsible for the query framework.
Guy Pelletier, Oracle	Committer	Guy is a developer on Oracle TopLink and committer on TopLink Essentials. Guy is the primary developer responsible for the object-relational and JPA metadata.
Chris Delahunt, Oracle	Committer	Chris is a developer on Oracle TopLink and committer on TopLink Essentials. Chris is the primary developer responsible for the object building life cycle.
Kyle Chen, Oracle	Committer	Kyle is a developer on Oracle TopLink and committer on TopLink Essentials. Kyle is currently focussing on customer reported issues addressing functionality and usability.
Denise Smith, Oracle	Committer	Denise is a developer on Oracle TopLink responsible for implementing and delivering an SDO 2.1-compliant product. She is the primary developer responsible for handling customer escalations relating to SDO. She is also a key contributor to the core OXM component.
Rick Barkhouse, Oracle	Committer	Rick is a developer for Oracle TopLink and is currently the primary developer for delivering the designtime component of the BPEL Database adapter, which is built on core TopLink.
Matt MacIvor, Oracle	Committer	Matt is a developer for Oracle TopLink, responsible for delivering a JAXB 2.0-

		compliant product based on TopLink OXM. He owns JAXB 2.0 annotation support and also contributes to the SDO component.
David McCann, Oracle	Committer	Dave is a developer for Oracle TopLink, responsible for delivering a JAXB 2.0-compliant product based on TopLink OXM. He also contributes to the SDO component. Dave is the primary developer responsible for handling customer issues relating to the TopLink EIS component.
Michael O'Brien, Oracle	Committer	Michael is a developer on Oracle TopLink responsible for implementing and delivering an SDO 2.1-compliant product. He is the primary developer responsible for implementing SDO ChangeSummary.
David Minsky, Oracle	Committer	David is a developer on Oracle TopLink focussing on customer reported issues addressing functionality and usability.
Peter Purich, Oracle	Committer	Peter is a technical writer for Oracle TopLink who will be contributing documentation content for EclipseLink.
Liza Rekadze, Oracle	Committer	Liza is a technical writer for Oracle TopLink who will be contributing documentation content for EclipseLink.
Rick Sapir, Oracle	Committer	Rick is a technical writer for Oracle TopLink who will be contributing documentation content for EclipseLink.

## 7. Community Support

During the proposal phase, many parties, both within the Eclipse Ecosystem and without, expressed interest in the proposed EclipseLink project.

### 7.1. Eclipse Projects

This project will deliver key persistence infrastructure that any Eclipse project can use for database persistence, XML binding, SDO usage, or EIS persistence. The following projects have already been identified as candidates for consuming/integrating/extending the delivered technology.

Eclipse Project	Interest
<b>EMF Project</b> Ed Merks (IBM)	<p>The EMFT <a href="#">Teneo</a> project provides tools for the generation of object-relational mappings targeting both the RedHat(TM)/JBoss(TM) Hibernate(TM) product and JPOX JDO. Support for JPA based mapping is not currently available in Teneo. Availability of the EclipseLink JPA runtime would allow for a complete and standards based persistence solution for EMF 'out of the box'.</p> <p>Currently, the EMFT Teneo project has Shaun Smith (Oracle) and Stephan Eberle (Bosch) as committers. They have already built an EMF persistence solution using Oracle TopLink and will participate in the contribution of the</p>

	EclipseLink based solution to the EMF project.
<b>Maya Project</b> (incubation)	The Maya Project requires a persistence technology to store its configuration information in a data store. By leveraging EclipseLink, Maya will be able to accomplish this with any relational database as well as optionally support XML configuration.
<b>Dali Project</b>	The Dali project provides JPA tooling support within WTP. This project is focussed on providing JPA compliant tooling as well as enabling vendor extensibility for custom extensions. Specific support for EclipseLink will be available as a Dali extension.
<b>Equinox Project</b>	One of the goals of EclipseLink is the development of a set of persistence service patterns for OSGi based applications. The EclipseLink project will work with the OSGi expert group to define these patterns.  EclipseLink will be an exemplary OSGi runtime framework that should be compatible with any compliant OSGi implementation.
<b><u>SOA Project</u></b> (proposed)	The proposed SOA project will provide a runtime framework for application developers. This project has some potential intersection points with the EclipseLink project concerning SDO, DAS, and general persistence and data binding. A discussion has been initiated to delve into how these projects may relate.

## 7.2. Interested Parties

Party	Interest
Genuitec (MyEclipse)	MyEclipse includes design time support for JPA and is interested in supporting the JPA implementation from EclipseLink as well as other persistence services offered by this project.  "Genuitec is looking forward to working with the project to adopt and deliver an enterprise ready, open source, standards-based, data binding system all wrapped in a commercial-friendly license (EPL)." - Todd Williams, Genuitec
TNI Eliane Fourgeau	TNI-Software, a Premium Member of AUTOSAR (AUTomotive Open System ARchitecture), has been developing AUTOSAR Builder, an open, Eclipse based Tool Suite for AUTOSAR. AUTOSAR Builder is an Eclipse based platform which proposes a number of open-interface plug-ins corresponding to different AUTOSAR development process phases. These plug-ins produce standard descriptions and information based on the AUTOSAR methodology while facilitating the contribution of each actor realizing a specific phase of the automotive EE development process.  TNI-Software is interested in the use of EclipseLink for persisting AUTOSAR models.
Innoopract/RAP Joachen Krause	Innoopract is a software and service company that helps developers and corporations make the most of their investment in development tools and platforms. The company is a founding member of the Eclipse foundation and bases its products and services on Eclipse's leading development platform.  Innoopract is interested in the development of frameworks that will extend



	the Eclipse runtime platform.
TmaxSoft Wonsoek Kim	Wonsoek of TmaxSoft is currently a committer on the TopLink Essentials project in GlassFish. TmaxSoft has expressed interest in this project and will hopefully join as full committer(s) as soon as is feasible.
Interface21	Interface21 are the principal developers of the Apache-licensed Spring Framework. Currently, they include support for JPA and ship multiple implementations, including TopLink Essentials.  Interface21 is interested in the integration of leading open source solutions within the Spring Framework.
Andrejus Baranovskis	Andrejus has been an active user of Oracle TopLink and involved with extending the framework for advanced Oracle database types and queries. Andrejus is active in his blogging of this work ( <a href="http://andrejusb.blogspot.com/">http://andrejusb.blogspot.com/</a> ). Andrejus is very interested in contributing to this project and becoming a committer.
Stephan Eberle (Bosch)	Stephan is a committer on Teneo working on providing EMF persistence with JPA. EclipseLink would provide an EPL licensed JPA provider for use in Teneo.

## 8. Roadmap

The following is the proposed roadmap for the project. Based on community feedback and additional contributions, this schedule may require adjustment.

May 30th, 2007

Project Creation Review

TBD (Requires project provisioning)

Initial contribution of code

0.1 - July 30th, 2007

This will be the first milestone focused on providing a functional preview with test cases for the initial code contribution. The goal of this preview is to allow interested parties and projects to evaluate the technology and assist with integration/extension discussions.

The project will function under an agile development process with short iterations rolling up into regular milestone deliverables 6-8 weeks apart. Beyond the initial milestone, the incubation project will focus on addressing the intersection with other Eclipse projects as well as delivering functionality in the areas of:

- JAXB 2.0 compliance
- SDO 2.1 compliance
- Data Access Service (DAS)
- DBWS using WS-\* standards

During the initial incubation phase, the milestone release schedule with specific functionality will be discussed and published.