ECLIPSE PERSISTENCE PLATFORM (ECLIPSELINK) FAQ

1. What is Oracle proposing in EclipseLink, the Eclipse Persistence Platform Project?

Oracle is proposing the creation of the Eclipse Persistence Platform (EclipseLink) project in Eclipse that will deliver a comprehensive persistence platform. Oracle will lead this open source project and will contribute the source and test cases of its successful Oracle TopLink product to provide a starting point for the project.

2. Why has Oracle become a Strategic Member of Eclipse?

Before becoming a Strategic Member, Oracle was leading three projects (Dali JPA Tools, JavaServer Faces Tools, BPEL) and participating in the Data Tools Platform's Enablement project. The Eclipse Persistence Platform project involves the dedication of a significant number of additional Oracle developers as well as a sizeable code contribution. The move to Strategic Member level reflects the size of the commitment Oracle is making to the Eclipse community and Oracle's interest in the success of Eclipse and the projects it is involved with.

3. Why is Oracle proposing the Persistence Platform project be developed with the Eclipse Foundation?

Through Oracle's previous participation, it has experienced firsthand Eclipse's enthusiastic, active and vibrant developer community and the healthy ecosystem of projects and companies. The Eclipse Foundation provides a vendor neutral venue that fosters the development of open source frameworks and tools that can be safely leveraged by both open source projects and commercial product companies. With the common interest the Eclipse Foundation and Oracle shares in OSGi technology, Eclipse is the ideal place to propose the Persistence Platform project.

4. What other projects are Oracle involved with at Eclipse?

Oracle currently leads three projects at Eclipse: Dali JPA Tools project, JavaServer Faces Tools projects, and the BPEL project. Oracle is also participating in the Data Tools Platform (DTP) Enablement sub-project to provide integration between DTP and the Oracle Database and has contributed support for deployment of applications to the Oracle Application Server from the Web Tools Platform (WTP).

The Dali JPA Tools project, a WTP sub-project, is building frameworks and tools for the development of applications that use JPA.

The JavaServer Faces Tools project aims to simplify development and deployment of JavaServer Faces (JSF) applications. It is a component of the Java Standard Tools (JST) WTP sub-project.

The goal of the BPEL Project is to add comprehensive support to Eclipse for the definition, authoring, editing, deploying, testing and debugging of WS-BPEL 2.0 processes. It is a sub-project of the Eclipse Technology project.

5. What capabilities will the Eclipse Persistence Platform offer?

The Eclipse Persistence Platform will deliver a proven framework with support for a number of persistence standards. It will be composed of a number of components based on a common core, each of which supports a different persistence standard or technology.

- **EclipseLink-ORM** will provide an extensible Object-Relational Mapping (ORM) framework with support for the Java Persistence API (JPA). It will provide persistence access through JPA as well as having extended persistence capabilities configured through custom annotations and XML. These extended persistence features include powerful caching (including clustered support), usage of advanced database specific capabilities, and many performance tuning and management options.
- **EclipseLink-OXM** will provide an extensible Object-XML Mapping (OXM) framework with support for the Java API for XML Binding (JAXB). It will provide serialization services through JAXB along with extended functionality to support meet in the middle mapping, advanced mappings, and critical performance optimizations.
- **EclipseLink -SDO** will provide a Service Data Object (SDO) implementation as well as the ability to represent any Java object as an SDO and leverage all of its XML binding and change tracking capabilities.
- **EclipseLink -DAS** will provide an SDO Data Access Service (DAS) that brings together SDO and JPA.
- **EclipseLink -DBWS** will provide a web services capability for developers to easily and efficiently expose their underlying relational database (stored procedures, packages, tables, and ad-hoc SQL) as web services. The metadata driven configuration will provide flexibility as well as allow default XML binding for simplicity.
- **EclipseLink -XR** will deliver key infrastructure for situations where XML is required from a relational database. The metadata driven mapping capabilities of EclipseLink-ORM and EclipseLink-OXM are both leveraged for the greatest flexibility. Using this approach to XML-Relational access enables greater transformation optimizations as well as the ability to leverage the

Eclipse Persistence Platform's shared caching functionality. While this capability is primary infrastructure for DBWS and DAS it will also be possible for consuming applications to leverage this directly.

- **EclipseLink -EIS** provides support for mapping Java POJOs onto non-relational data stores using the Java Connector Architecture (JCA) API.

6. This is a runtime project. Doesn't Eclipse only have tooling projects?

Eclipse is well known for tools but has also been providing runtimes components. The runtime projects are focused on providing frameworks and platforms on-top of the OSGi runtime, Equinox. For example, the Eclipse Rich Client Platform is widely used for building desktop applications and the Eclipse Modeling Framework (EMF) is another example of a framework that is incorporated into a running application that helps manage an application data model.

7. How will a persistence platform with multiple capabilities benefit the Eclipse community and its users?

Although relational databases are the most common storage for enterprise data, access to many other data sources and types are often required. Developers are familiar with the ORM problem space, but with the pervasiveness of XML there is also a need for a solution that also covers OXM. Applications also may need the data materialized in different structures using different standards, so in addition to POJOs, SDO is also supported.

The Eclipse Persistence Platform will also allow developers to leverage their mapping metadata across multiple persistence capabilities:

- Customers using EclipseLink-ORM will be able to leverage this metadata and domain model to easily deploy their persistence unit as a Web Service. The XML binding can be defaulted or further customized by the developer using metadata.

- Customers using EclipseLink-ORM who need to interoperate with SDO enabled components will be easily able to leverage their existing metadata and expose their entities as SDO data objects. They will also be able to leverage the Data Access Service (DAS) functionality for querying and modifying the entities.

- The EclipseLink-OXM meet in the middle mapping allows developers to leverage a single domain model and support multiple mappings to different XML representations and data stores. This flexibility will greatly simplify SOA solutions where multiple services must be exposed using the same underlying domain model and logic as well as for cases where the

ability to quickly adapt to different consumers or backend data stores is a necessity.

- The usage of EclipseLink-OXM will enable powerful meet in the middle mapping with separate XML metadata. This approach to XML binding will enable Eclipse Persistence Platform users to easily leverage their mapped domain models and the metadata to quickly expand the applications to include different data store persistence options as well as easy integration into Web Services using either the EclipseLink-DBWS solution or to plug into a JAX-WS solution within their container.

8. How will the Eclipse Persistence Platform project affect Oracle TopLink?

Oracle will continue to deliver its award winning Oracle TopLink product to customers for use within both application servers and as a stand-alone product. In addition to being a popular solution our customers use directly in their applications, Oracle TopLink is a key piece of infrastructure within the Oracle SOA Suite. Oracle will continue to deliver the product, offer support, and enhance the capabilities as demanded by our customers. Oracle TopLink will be built on the Eclipse Persistence Platform and will extend it with advanced integration into the Oracle Application Server.

9. Will Oracle TopLink offer superior capabilities to that of the Eclipse Persistence Platform?

No. The goal of contributing Oracle TopLink to the Eclipse community is to deliver a world-class persistence platform to one of the best open source communities making it freely available to all. There will not be any effort to restrict or limit the functionality or performance of the Eclipse Persistence Platform. The proposal of the Eclipse Persistence Platform is not a move to build an entry-level persistence platform but to instead to build a commercial quality platform for the Java community. The initial contribution of the Oracle TopLink product is intended to kick-start the project.

The commercial Oracle TopLink product will continue to be developed but will only contain the proprietary integration code necessary for some functional areas within the Oracle Application Server and Oracle SOA Suite. These capabilities will be minimized as much as possible and will initially include EJB 2.1 integration, OracleAS specific diagnostics including logging and the Oracle Dynamic Monitoring Service (DMS), and backwards compatible support for older Oracle TopLink metadata running on the new Eclipse Persistence Platform core. Oracle TopLink will also include the TopLink Workbench, which is a non-Eclipse based tool for Oracle specific mapping metadata

10. How will the Eclipse Persistence Platform project affect existing Oracle TopLink customers?

Existing Oracle customers will continue to benefit from their usage of Oracle TopLink. Oracle TopLink will continue to exist as it has since its first commercial release in 1994.

In future versions of Oracle TopLink, the core persistence functionality will be provided by the Eclipse Persistence Platform. Customers can expect to find their persistence capabilities grow and to see a continued focus on performance, scalability, flexibility, and developer productivity.

11. Can other companies or individuals participate in the Eclipse Persistence Platform community?

Yes. One of the principal goals of open source development at Eclipse is to encourage all interested parties and vendors to participate and contribute to the project in all areas. Oracle recognizes the value of working with as many parties as possible on open source projects and invite all interested parties to participate on the Eclipse Persistence Platform project.

12. Will the Eclipse Persistence Platform provide the Java Persistence API (JPA) implementation in the Oracle Application Server?

Yes. The Eclipse Persistence Platform will provide the core JPA functionality required by the specification as well as all of the advanced ORM capabilities Oracle TopLink customers leverage today. It will be provided as the JPA implementation through Oracle TopLink's inclusion of the Eclipse Persistence Platform and will deliver the best possible integration into the container through additional integration in Oracle TopLink with such features as integrated logging and management.

13. How does this effect TopLink Essentials and Oracle's involvement in GlassFish?

Oracle will continue to participate in the GlassFish community as the lead for the TopLink Essentials project. Delivery of a commercial quality reference implementation of JPA has been positive for the JPA community and Oracle plans to continue its support and work with the TopLink Essentials community and contributors to evolve this product based on the evolution of JPA.

14. Will consumers be able to get professional support and services for the Eclipse Persistence Platform?

Yes. With Oracle TopLink including the Eclipse Persistence Platform as the core of its functionality consumers will be able to obtain support from Oracle. In the future, other vendors may also provide consulting and support services for the open source Eclipse Persistence Platform project. Peer support will also be available from the Eclipse Persistence Platform community through the project newsgroup and mailing lists.

15. Can I still license the Oracle TopLink product?

Yes. The commercial Oracle TopLink product that offers advanced integration with the Oracle Application Server will continue to be licensed and supported by Oracle. It will continue to be available for download from the Oracle Technology Network (OTN) website subject to the OTN license.

Oracle TopLink will also continue to be bundled with the Oracle Application Server, which includes a license for Oracle TopLink. When purchasing a license for the Oracle Application Server an additional license for Oracle TopLink is not necessary as it is included.

The Eclipse Persistence Platform project will be available free of charge under the Eclipse Public License and downloadable from the Eclipse.org website.

16. Can I still get support for Oracle TopLink through Oracle Support?

Yes. Oracle TopLink customers will continue to access professional support through Oracle Support as they have in the past.

17. Does the Eclipse Java IDE provide development features for Oracle TopLink?

The Eclipse Dali JPA Tools project is building tooling for the development of applications that will run with JPA compliant persistence providers like the Eclipse Persistence Platform and Oracle TopLink. Oracle will be developing extensions to Dali that will provide specific support for advanced Eclipse Persistence Platform JPA features. This Dali extension will also support Oracle TopLink.

18. Can the Eclipse Persistence Platform be used by Spring applications?

Yes, Customers will continue to be able to leverage Spring's support for pluggable ORM and JPA providers with EclipseLink-ORM. In fact, Spring's

integration with the Eclipse Persistence Platform is likely to exceed the level in the current Spring/Oracle TopLink integration with the open source availability of the full Oracle TopLink feature set in the Eclipse Persistence Platform.

For example, the open source availability of advanced XML meta-data based Object-XML mapping functional in EclipseLink-OXM opens the door to new opportunities for Spring Web Services/Eclipse Persistence Platform integration fully compatible with Spring's POJO programming approach.

19. Will the Eclipse Persistence Platform OXM overlap with similar functionality provided by EMF?

No, they're complementary. The Eclipse Persistence Platform will provide XML and relational mapping for arbitrary POJO classes and schemas supporting meet in the middle mapping. EMF is a model driven framework that generates classes that have a well-defined XML representation. It will also be possible to combine EMF with Eclipse Persistence Platform to persist EMF generated classes into relational databases.

20. Will the Eclipse Persistence Platform SDO implementation overlap with the one provided by EMF?

No, the EMF SDO implementation is no longer being actively developed and never achieved SDO 1.0 compliance. The initial goal for the Eclipse Persistence Platform SDO implementation will be SDO 2.1 compliance and then will track the standard as it evolves.

21. Will the Oracle TopLink Workbench be contributed to the Eclipse Persistence Platform?

The Eclipse Persistence Platorm project will be focused on providing runtime capabilities. The Oracle TopLink Workbench is a design-time tool that supports the definition of Oracle TopLink Object-Relational Mapping, Object-XML Mapping, and EIS metadata and is not part of the initial contribution to the EclipseLink project. Tooling for EclipseLink-ORM is available from the Oracle lead Eclipse Dali JPA Tools project. Plans for the development of Eclipse based tooling for the other Eclipse Persistence Platform EclipseLink component frameworks have not been made at this time.

22. Are there other Fusion Middleware components that Oracle is considering open sourcing in Eclipse or other communities?

Oracle actively reviews its product portfolio along with important emerging technology trends to ensure that it is involved in key communities and, where appropriate, providing technologies like Oracle TopLink from its portfolio to help

ensure successful open source efforts. The goal of Oracle's involvement in open source is to encourage the development of communities around key technologies it believes are important for its customers. Contributions like those Oracle has provided to Eclipse, Apache, Spring and numerous other open source efforts are evidence of that commitment and such contributions and open source community participation will continue as Oracle sees the fit and opportunity.

23. How does the decision by Oracle to launch the Eclipse Persistence Platform affect the future direction of Oracle's JDeveloper?

Oracle's decision to propose the Eclipse Persistence Platform project does not affect its commitment to JDeveloper nor does it signify any changes in tooling strategy. JDeveloper continues to be Oracle's strategic IDE for building applications using the latest industry standards for Java, XML, web service, SQL and SOA. Oracle is committed to providing "productivity with choice" and offers both JDeveloper and Eclipse users a first rate development experience when building applications with Oracle technology.

24. Will JDeveloper still support the development of Oracle TopLink applications?

Yes. JDeveloper will continue to provide integrated tooling for the development of Oracle TopLink applications including those that utilize ADF with EJB 3.0 and TopLink.

25. Are you planning further enhancement for Oracle TopLink development in future versions of JDeveloper?

Yes. JDeveloper's Oracle TopLink support will continue to evolve and expand to cover new features like SDO, XR and DBWS.