Half Pager: Testing Persistence Services for GlassFish V3

Table of Contents

1. Introduction

1.1. Project/Component Working Name

Testing Persistence Services for GlassFish V3

1.2. Name(s) and e-mail address of Document Author(s)/Supplier

Sherry Shen (sherry.shen@sun.com)

1.3. Date of This Document

Initial Document: February 06, 2009
Last Revised: October 21, 2009

2. Project Summary

2.1. Project Description

For each requirement of GFv3 persistence services, the test tasks are described below it:

- **2.1.1** *For Implementing a JPA connector for GlassFish*
  - **2.1.1.1 Testing JPA 2.0 and JPA 1.0 RI with EclipseLink**
    - EclipseLink is the default JPA provider in GlassFish.
    - All JPA1.0 and 2.0 suites available for the tests of EclipseLink
    - *appserver-sqe/pe/ejb/ejb30*, including JPA1.0 tests
    - *appserver-sqe/pe/ejb/ejb31*, including JPA integration
    - *appserver-sqe/pe/ejb/jpa20*, JPA2.0 tests
  - **2.1.1.2 Testing JPA1.0 RI with Hibernate**
    - Hibernate will be an external JPA provider from GlassFish Update Center.
    - A JPA1.0 suite is available for the sanity tests of Hibernate.
    - *appserver-sqe/pe/ejb/ejb30/war/jpaJTALookupEM*

- *appserver-sqe/pe/ejb/ejb30/war/jpaJTALookupEM*

- **2.1.2** *For Integrating JPA 2.0 (JSR 317) RI into GlassFish*

  - **2.1.2.1 Developing Use Cases for JPA2.0**
    - The use case technique is used to capture a system's behavioral requirements by detailing scenario-driven threads through the functional requirements.
    - The features in this section are database dependent and will be tested across v3 supported databases.
    - **Feature (P1): Pessimistic Locking**
      - Concurrent accesses through multi-thread Java client are used to exam the locking mechanism in pessimistic locking support.
      - Pessimistic locking in find or query for version and non version objects.
      - Locking timeout.
      - Pessmistic locking via optimistic locking
      - *appserver-sqe/pe/ejb/jpa20/war/locking*
      - *appserver-sqe/pe/ejb/jpa20/war/pessimistic*
    - **Feature (P2): Undelimited Identifiers**
      - The delimited identifiers are used for table name and column name.
      - Two different identities have the same table names in different cases.
      - The persist, update, remove, query and native query are examined.
      - *appserver-sqe/pe/ejb/jpa20/ear/delimited*

  - **2.1.2.2 Developing Basic Acceptance Tests** (BAT) **for JPA2.0**
    - The features in this section are not database dependent, BAT is planned for SQE as there are developer unit tests and CTS tests.
    - **Feature (P3): MapsId** (MappedById)
      - JPA Example. The entity *MasterCorporal* has @EmbeddedId, and a Mx1 MapsId w.r.t. the entity *Sargeant* 's id (@Id).
      - *appserver-sqe/pe/ejb/jpa20/ear/mappedbyid*
    - **Feature (P3): Derived Identifiers/ Extended Map Support**
      - JavaEE 6 Sample. The entity *Order* contains a map from entity *Item* to entity *LineItem* mapped as the OneToMany relationship, i.e.**ternary relationships** of *Order*, *Item* and *LineLitem*. The entity *LineItem* has a composite primary key consisting of lineItemNumber and order id. The order id part of the key is also mapped as ManyToOne relationship using the **derived id** feature introduced in JPA 2.0. A session bean that acts as a facade to operate on these entities. Methods createData and queryData demonstrates how to create and query data for entities using above mapping features.
      - *appserver-sqe/pe/ejb/jpa20/war/advancedmapping*
    - **Feature (P2): Criteria API**
      - JavaEE 6 Sample. The application is exact copy of the advancedmapping. The only difference being the queries are written using Criteria API.
      - *appserver-sqe/pe/ejb/jpa20/war/criteriaquery*

  - **2.1.2.3 Porting and extending GFv2 SQE JPA 1.0 Tests**
    - About 600 test cases of EJB3.0/JPA1.0 from GFv2 are ported to GFv3
      - For JPA tests, replaced TopLink Essential with EclipseLink.
      - *appserver-sqe/pe/ejb/ejb30*
    - About 65 new test cases are added for ear and war package
      - *appserver-sqe/pe/ejb/ejb30/persistence/pkgEarTest**
      - *appserver-sqe/pe/ejb/ejb30/war/**

- **2.1.3** *For Integrating Bean Validation (JSR 303) RI into GlassFish*

  - **2.1.3.1 Adding use cases for JPA using Bean Validation**
    - **Feature (P1): Validation Mode**
      - Basic tests verify JPA BeanValidation support with war or ear package, container or bean managed transaction.
      - The tests specify **validation-mode** in persistence.xml as **AUTO**, **NONE** and

        - The tests specify **validation-mode** in persistence.xml as **AUTO, NONE** and **CALLBACK**.
        - The tests persist, update and remove entity with Constraint Violation w.r.t. default group.
        - *appserver-sqe/pe/ejb/jpa20/ear/bvauto*
        - *appserver-sqe/pe/ejb/jpa20/ear/bvnone*
        - *appserver-sqe/pe/ejb/jpa20/war/bvcallbackbmt*
        - *appserver-sqe/pe/ejb/jpa20/war/bvcallbackcmt*
    - **Feature (P2): Validation Groups**
        - Advanced tests further exam the use cases with constraints for specific entity operations in AUTO mode.
        - The tests defined **validation groups** and specify the group for **pre-persist**, **pre-update** and **pre-remove** with the property of "javax.persistence.validation.group.pre-*" in persistence.xml.
        - The tests persist, update and remove entity with Constraint Violation w.r.t. specified or default groups.
        - *appserver-sqe/pe/ejb/jpa20/ear/bvgroup*

- **2.1.4** *For Porting CMP code from V2*

    - **2.1.4.1 Porting GFv2 SQE CMP Tests**

        - Around 150 test cases of EJB2.x/CMP/BMP from GFv2 are ported for GFv3.
        - Stateless, Stateful, User Exception, MixedBean,
        - ReadOnlyBean, CMP, Capture Schema, BMP, Security
        - *appserver-sqe/pe/ejb/stateless*
        - *appserver-sqe/pe/ejb/stateful*
        - *appserver-sqe/pe/ejb/cmp*
        - *appserver-sqe/pe/ejb/bmp*

    - **2.1.4.2 Executing GFv2 Development CMP/Transparent-Persistence Tests**

        - CMP/Transparent-Persistence Tests have around 730 runtime tests and 20 application server tests.
        - These tests may be executed depending on development and sqe schedule and resource.

2.2. Risks and Assumptions

Assumptions

- The spec for JSR 317 are finalized within the time line of V3 schedule
- JPA 2.0 RI will be available within the time line of V3 schedule

Risks

- Late binding features like Criteria API and Bean Validation (JSR 303) integration can affect timely delivery.

3. Interfaces

3.1. Exported Interfaces

interface : javax.persistence (ver 2.0)
stability: standard
comments : jsr 317 in public review stage

3.2. Imported Interfaces

interface : javax.naming
stability: standard

stability: standard
comments :

interface : javax.ejb
stability: standard
comments : Used by CMP

interface : com.sun.ejb
stability: standard
comments : Used by CMP

4. Reference Documents

- JSR 317
- JPA 2.0 RI implementation wiki
    - Validation, 266927
    - Pessimistic Locking, 248489
    - Undelimited Identifiers, 251908
    - MappedById, 270011
    - Derived Identifiers, 241765
    - Extended Map Support, 241410
    - Criteria API, 249218