CurrentDevelopment

This page list the work planned for the JDO 2.1 Maintenance Release. The work may involve updating the specification, changing the api, and/or adding tests or modifying tests in the TCK.

The task to adapt the JDO sources to the new apache license policy is completed, see FixingLicenseHeader for details.

Number	Task	Spec changes	API changes	TCK changes
1	In 5.4.1, Compound Identity should be updated to reflect that for key fields of reference types, the type of the key field is the reference type in the class but the oid of the reference type in the oid class.	done		done
2	In Table 2: State Transitions, the transition for a transient-dirty instance during commit with DetachAllOnCommit = true should be to transient-clean. Also, state changes need to be added for detach methods and serialization. JDO-456	done		done
3	In 5.5.8 and 5.5.9, detachCopy should be removed from the list of methods that throw exceptions if applied to detached-clean or detached-dirty instances.	done		
4	In 6.4.3, change "Portable JDO applications must not depend on whether instances of these classes are treated as SCOs or FCOs. " to "Portable JDO applications must not depend on SCO or FCO uniquing behavior, nor on the storage mechanism in the datastore. Portable applications may use the same instance of these classes as field values in any persistence-capable class instance." JDO-397	done		
5	In section 7.5, change "If the class is abstract, null is returned." to "If the class is abstract, JDOFatalInternalException is thrown".	done		
6	In section 7.5, add public byte fetchByteField(int fieldNumber); to ObjectIdFieldSupplier.	done		
7	In section 7.5, add public byte storeByteField(int fieldNumber, byte value); to ObjectIdFieldConsumer.	done		
8	In Chapter 8, add after class JDOHelper {public JDOHelper(); For some usage patterns, an instance of JDOHelper on which to invoke methods is preferable to the use of static methods. For this purpose, a public constructor is provided. JDO-444	done	done	
9	In Chapter 11, add a convenience method that returns a PersistenceManager proxy that can be used in web and ejb containers to dynamically bind to the transaction-associated PersistenceManager. JDO-445	done	done	
10	In Chapter 9, add section on managing date formatting for ObjectIdentity constructors. public synchronized void registerDateFormat(java.text.DateFormat df);	done		
11	In 9.4, add method to retrieve persistence-capable classes that have been registered. public java.util.Collection getRegisteredClasses();	done		
12	In 9.5, add method to verify that the class is authorized to be a state manager. public static void checkAuthorizedStateManagerClass(Class smClass);	done		
13	In 9.5, add method to register multiple state manager classes. public static void registerAuthorizedStateManagerClasses(java.util.Collection smClasses) throws SecurityException;	done		
14	In Chapter 11, add properties for configuring PersistenceManagerFactory that are consistent with JPA specification of TransactionType and Persistence Unit Name. JDO-449	done	done	
15	In Chapter 12, specify the behavior of PersistenceManager if it extends Serializable and writeObject is called.	JDO-455		x
16	In 12.6.6, clarify that a JDOUserException will be thrown when invoking newInstance: if a class is not persistence-capable, or does not declare a public no-args constructor; if an interface is not persistence-capable or declares methods that are not defined as persistent properties; if an abstract class is not persistence-capable or declares abstract methods that are not defined as persistent properties. JDO-453 JDO-487	done		done
17	In 12.6.8, section heading Explicit Detach, the sentence "If the parameter instance is detached, then JDOUserException is thrown. " should be removed.	done		
18	In 12.6.8, add a note that serialization for storage using the serialized, serialized-element, serialized-key, or serialized-value metadata attributes does not create a detached instance.	done		
19	In 12.6.8, clarify the behavior of instances during serialization both with and without an active transaction. JDO-456	done		done
20	In 12.7.5, change public int setMaxFetchDepth(); to public int getMaxFetchDepth();	done		
21	In 12.7.5, specify that getFetchGroups returns a read-only copy of the active Fetch Groups.	done		done
22	In 12.7.6, p. 127, change "A recursion-depth of 0 will fetch the whole graph of instances reachable from this field." to "A recursion-depth of -1 will fetch the whole graph of instances reachable from this field."	done		
23	In 12.7.6 p. 129, change fetch-depth to recursion-depth in the example.	done		
24	In 14, add subqueries to permit e.g. select from Employee where this.salary > (select avg(salary) from Employee) JDO-446	JDO-569	done	done
25	In 14.6.2 p. 159, the sectionheading "Methods" is not marked as a sectionheading.	done		
26	Add to 14.6.9: Projected SCOs are never owned, projected FCOs are always managed. Modifying unowned SCOs never has an effect on the database. Modifying FCOs no matter how you get them always has an effect if the tx commits.	done		done
27	In 15.3, p. 187, the third paragraph, beginning "The field on the other side" and ending "in the next transaction", is duplicated and will be removed.	done		
28	In 15.3, add text to describe updating the other side of relationships where this side is deleted. This maintains referential integrity for delete as well as update. JDO-521	done		done

2 9	Add to 17.1.11 JDOUserCallbackException extends javax.jdo.JDOUserException	done	JD O- 447	
3 0	In Chapter 18, add an xml element to specify the fetch plan to use for a query.		JD O- 448	JD O- 448
3 1	In 18.15.1, change "(e.g. a field of type Object can specify field-type="Integer")." to "(e.g. an element of type Object can specify element-type="Integer").	done		
3 2	In Chapter 18, add to .jdo metadata: <!ATTLIST property field-type CDATA #IMPLIED>	done		
3 3	Add to 18.15.1 "The default for dependent-element is false."	done		
3 4	Add to 18.15.2 "The default for dependent-key is false."	done		
3 5	Add to 18.15.2 "The default for dependent-value is false."	done		
3 6	Change the last paragraph of 21.6 from "For Detachable classes, the results of restoring a serialized persistent instance graph is a graph of interconnected detached instances that might be attached via the attachCopy methods." to "For Detachable classes, the results of restoring a serialized persistent instance graph is a graph of interconnected detached instances that might be attached via the makePersistent methods."	done		
3 7	Change 21.13 from "Some methods require a non-null state manager. In these cases, if the jdoStateManager is null, then IllegalStateException is thrown." to "Some methods require a non-null state manager. In these cases, if the doStateManager is null, then JDOFatalInternalException is thrown."	done		
3 8	Change 21.21.7 Generated jdoGetManagedFieldCount sample implementation to avoid using the jdoFieldNames field that might be initialized after it is used during initialization of a subclass. (See ChangeLog for details.)	done		
3 9	Add to Chapter 23, constants defined in the JDOPermission class. (See ChangeLog for details.)	done		
4 0	Add methods to `PersistenceManager` and `PersistenceManagerFactory` to allow a user to get the current time and date from the server. Allow the `Persistence ManagerFactory` to be configured statically with `TimeZone` and `Locale` in case the information is not available from the server itself. JDO-462	done	done	
4 1	Add to 6.3 a section requiring support for enum types, including collections of enum types. JDO-546	done		done
4 2	Change in 12.6 signatures of some PersistenceManager methods to be generic. Note that these changes are source compatible with existing application programs. (See ChangeLog for details.) JDO-450 JDO-542 JDO-538	done	done	done
4 3	Add to 15.1 a paragraph describing that mapping an enum to a fixed-precision numeric type uses the ordinal() value for storage; mapping to a character column type (CHAR, VARCHAR, etc.) uses the name() value for storage; mapping to any other column type is an error. JDO-546	done		done
4 4	Provide in a new chapter a set of annotations that map directly to xml elements as an alternative to using xml metadata. Describe how jdo implementations can use either JDO annotations or JPA annotations to provide metadata. JDO-451 details JDO-503	done		JD O- 564
4 5	Provide interfaces that extend both JDO and JPA in order to make it easier to migrate applications from JDO to JPA. JDO-452	done	done	
4 6	Add method evictAll taking a class instance to PersistenceManager interface. JDO-535	done	done	
4 7	Add new test case for `JDODetachedFieldAccessException`.	done	done	JD O- 563