

AllTheOtherTests

JDO 2.0 TCK Assertions to be Tested

These are the assertions, in addition to those for metadata ([XMLMetadata](#)) and query (QueryTests), that need to be tested to complete the JDO 2.0 TCK. See the spec for more information on each assertion. If you wish to volunteer, enter a JIRA issue if one has not already been created and add your initials to the Who column. Discuss the issue with comments on JIRA.

Spreadsheet Tab	Number	Test	Who	JIRA	
LifeCycle	A5.1-AD1	detached.	.	.	
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="d2f2cc9a-e853-4eb6-a6bb-32bfb4c94547"><ac: plain-text-body><![CDATA[	LifeCycle	A5.1-AD2	[A detached instance]... and any attempt to access unloaded fields, whether for read or write, is denied	.	]]></ac: plain-text-body></ac: structured-macro>
LifeCycle	A5.1-AD3	A detached instance maintains its persistent identity and the identity can be obtained by an observer.	.	.	
LifeCycle	A5.1-AD4	A detached instance allows changes to be made to loaded fields, and tracks those changes detached.	.	.	
LifeCycle	A5.1-AD5	Detached instances never observe transaction boundaries.	.	.	
LifeCycle	A5.8-AD1	JDO instances that have been removed from their persistence manager and have not been modified are detached-clean.	.	.	
LifeCycle	A5.8-AD2	Detached-clean instances transition to detached-dirty if a loaded field is modified.	.	.	
LifeCycle	A5.8-AD3	Attempts to change their state via any of the persistence manager methods (e.g. makePersistent, deletePersistent, etc.) throw JDOUserException.	.	.	
LifeCycle	A5.9-AD1	the instance is detached in the persistent-dirty state while the PersistenceManager flag DetachedDirty is true;	.	.	
LifeCycle	A5.9-AD2	a field of the detached instance is explicitly modified by the application. Dirty-detached instances do not change their life cycle state.	.	.	

<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="743d112f- 1c24-4fc9-8b89- 7a4cc7f5a691"><ac: plain-text-body><! [CDATA[	PC	A7.5-1	Persistence[('Capable jdoNewInstance(State))' Manager sm); A7.5-1 [This method creates a new instance of the class of the instance.]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="06006917- 8fff-4087-8036- 27492f35240e"><ac: plain-text-body><! [CDATA[	PC	A7.5-2	A7.5-2 [If the class is abstract, null is returned.]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="e20d79ca- 78a8-4ef1-a113- cf75dc3d2fbf"><ac: plain-text-body><! [CDATA[	PC	A7.5-3	Persistence[('Capable jdoNewInstance(State))' Manager sm, Object oid); A7.5-3 [This method creates a new instance of the class of the instance, and copies key field values from the oid parameter instance.]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="f0b9af03- 94e5-415d-af9b- 701048746f43"><ac: plain-text-body><! [CDATA[	PC	A7.12-1	public Object jdoNewObjecttdInstance(); A7.12-1 [This method creates a new instance of the class used for JDO identity.]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>

<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="b98b5a80- 6c23-4a79-bfff- f4d28152c5b8"><ac: plain-text-body><![CDATA[	PC	A7.12-2	public Object jdoNewObjectInstanceId();A7.12-2 [If the class has been enhanced for datastore identity], or A7.12-3 [if the class is abstract, null is returned.]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="b928e6c6- 4353-4878-9622- 71b2bde66ce7"><ac: plain-text-body><![CDATA[	PC	A7.12-3	public Object jdoNewObjectInstanceId(); or A7.12-3 [if the class is abstract, null is returned.]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="56c47d95- 303d-44e9-aaac- b4e3876c031b"><ac: plain-text-body><![CDATA[	PC	A7.12-20	Public Object jdoNewObjectInstanceId(String str); A7.12-20 [This method creates a new instance of the class used for JDO identity, using the String constructor of the object id class. It is intended only for application identity.]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="95f7c443- 58ad-44ca-85a7- 4535aff0da75"><ac: plain-text-body><![CDATA[	PC	A7.12-21	A7.12-21 [If the class has been enhanced for datastore identity],	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>

<ac:structured-macro ac:name="unnmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="6b5bab4b- 3332-4181-97f9- 53d23d8d7c69"><ac: plain-text-body><![CDATA[	PC	A7.12-22	A7.12-22 [if the class is abstract], null is returned.	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unnmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="1cae61a5- c747-430c-a551- 738af84bb683"><ac: plain-text-body><![CDATA[	PC	A7.12-40	A7.12-40 [Object{{{`Id`}}Field} Supplier: the field values are fetched and used to construct the single field identity instance]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unnmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="f4302e79- 7543-46ad-9233- 2ce650489847"><ac: plain-text-body><![CDATA[	PC	A7.12-41	A7.12-41 [Number: the parameter key must be an instance of the key type or, for primitive key types, the wrapper of the key type; the key is passed as a parameter to the single field identity constructor].	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unnmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="95837ecf- 40ba-4b39-b5bc- 312e4932c121"><ac: plain-text-body><![CDATA[	PC	A7.12-42	A7.12-42 [String: the String is parsed to a value of the appropriate type and the value is used to construct the single field identity instance. For Object`Identity, the String is decomposed into two parts using ":" as a delimiter. The first part is the class name, the second is the String representation of the value of the class]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>

<ac:structured-macro ac:name="unnmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="d771e399-df5c-4203-85e7-fa7a96d914d5"><ac: plain-text-body><![CDATA[	PC	A7.12-43	A7.12-43 [Object: for Object`Identity, the key type must be assignable from the parameter key]	.	.	]]></a c: pl ai n-te xt-bo dy></a c: str uc tu re d-m ac ro>
<ac:structured-macro ac:name="unnmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="d06df43f-c996-4834-8b64-c629ec755b29"><ac: plain-text-body><![CDATA[	PC	A7.13-1	A7.13-1 [void jdoReplaceDetachedState(); This method calls the State{[' Manager with the current detached state instance as a parameter and replaces the current detached state instance with the value provided by the State]} Manager.]	.	.	]]></a c: pl ai n-te xt-bo dy></a c: str uc tu re d-m ac ro>
<ac:structured-macro ac:name="unnmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="219eec1e-1020-45f8-8337-32af78f82aec"><ac: plain-text-body><![CDATA[	J D O H el per	A8.3-11	A8.3-11 [static Object[] getObjectIds (Object[] pcs); static Collection getObjectIds (Collection pcs); These methods return the JDO identities of the parameter instances. For each instance in the param-eter, the getObjectId method is called. They return one identity instance for each persistence-capable instance in the parameter. The order of iteration of the returned Collection exactly matches the order of iteration of the parameter Collection].	.	J D O- 188	]]></a c: pl ai n-te xt-bo dy></a c: str uc tu re d-m ac ro>
<ac:structured-macro ac:name="unnmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="5e8cad05-2314-4a8a-9218-af8cfb19c0f9"><ac: plain-text-body><![CDATA[	J D O H el per	A8.4-1	A8.4-1 [static Object getVersion (Object pc); This method returns the JDO version of the instance for persistent and detached instances. It returns null if the instance is transient or null or if its class is not persistence-capable.]	.	.	]]></a c: pl ai n-te xt-bo dy></a c: str uc tu re d-m ac ro>

<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="25d5436c-526d-4f4c-9f0c-2ac7ec6467be"><ac: plain-text-body><![CDATA[	J D O H e l p e r	A8.5.6-1	A8.5.6-1 [static boolean isDetached (Object pc); Instances that have been detached return true. The method returns false if the instance is transient or null or if its class is not detachable.]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="aaeb60d0-d286-4a26-98f1-29fd761aa9c9"><ac: plain-text-body><![CDATA[	J D O H e l p e r	A8.6-4	If there are any exceptions while trying to construct the Properties instance or to call the static method, then either A8.6-4 [JDO{{ Fatal}}User`Exception] or	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="4f1a3734-789d-466b-beac-c3670f1c5e20"><ac: plain-text-body><![CDATA[	J D O H e l p e r	A8.6-5	[JDO{{ Fatal}}Internal`Exception is thrown], depending on whether the exception is due to the user or the implementation.	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="3f92c5a3-afe0-49b5-b1d0-fca5486144f3"><ac: plain-text-body><![CDATA[	J D O H e l p e r	A8.6-6	A8.6-6 [If the class named by the javax.jdo.Persistence{{ Manager}}FactoryClass property cannot be found, or is not accessible to the user, then JDOFatalUser`Exception is thrown.]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>

<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="db0f4073-ddf4-4c7e-bb36-237e78e4067a"><ac: plain-text-body><![CDATA[	J D O H el per	A8.6-7	A8.6-7 [If there is no public static implementation of the getPersistenceManagerFactory(Properties) method, then JDO({Fatal}}Internal`Exception is thrown.]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="b14d1c9d-990f-4f8c-ba70-cda341f268ab"><ac: plain-text-body><![CDATA[	P M F	A11.4-10	A11.4-10 [boolean isClosed(); Return true if this Persistence({Manager}}Factory is closed; and false otherwise.]	.	J D O- 189	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="7b2fa937-6330-4cc6-9a54-3ccc10aafc93"><ac: plain-text-body><![CDATA[	P M F	A11.8-1	A11.8-1 [To obtain a reference to the cache manager, the getDataStoreCache() method of Persistence({Manager}}Factory is used. DataStore`C ache getDataStoreCache(); If there is no second-level cache, the returned instance silently does nothing.]	.	J D O- 190	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="db02b175-47ed-4d00-a688-9eceadc7a682"><ac: plain-text-body><![CDATA[	P M F	A11.9-1	A11.9-1 [void addInstanceLifecycleListener (Instance({Lifecycle}}){Listen-er listener, Class[] classes); This PersistenceManagerFactory method adds the listener to the list of instance lifecycle event listeners set as the initial listeners for each Persistence({Manager created by this Per-sistence}Manager Factory. The classes parameter identifies all of the classes of interest. If the classes parameter is specified as null, events for all persistent classes and interfaces are gen-erated. If the classes specified have persistence-capable subclasses, all such subclasses are regis-tered implicitly. The listener will be called for each event for which it implements the corresponding listener inter-face.]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>

<ac:structured-macro ac:name="unnmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="3e06cf3d-4b7d-4d70-9b06-f16467672db3"><ac: plain-text-body><![CDATA[	PM	A11.9-2	A11.9-2 [void removeInstanceLifecycleListener (Instance(['Lifecycle])Listener listener); This PersistenceManager([Factory method removes the listener from the list of event listeners set as the initial listeners for each PersistenceManager created by this PersistenceManager Factory.]			]]></ac:plain-text-body></ac:structured-macro>
<ac:structured-macro ac:name="unnmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="8aa4e451-572e-4574-9280-1c8a016f5b73"><ac: plain-text-body><![CDATA[	PM	A12.5.1-2	A12.5.1-2 (PART 1) [If evictAll with no parameters is called, then all persistent-clean instances are evicted (they transition to hollow).] A12.5.1-2 (PART 2), A12.5.1-3 (PART 2), A12.5.1-4 (PART 2) [For each persistent-clean and persistent-nontransactional instance that the JDO PersistenceManager evicts, it calls the jdoPreClear method on each instance, if the class of the instance implements Instance] Callbacks clears persistent fields on each instance (sets the value of the field to its Java default value); changes the state of instances to hollow or persistent-nontransactional (cannot distinguish between these two states) this is not directly testable..] See spreadsheet for more info.			]]></ac:plain-text-body></ac:structured-macro>
<ac:structured-macro ac:name="unnmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="ffe9f473-8b33-4643-83e2-091840e697e5"><ac: plain-text-body><![CDATA[	PM	A12.5.4-2	Extent getExtent (Class persistenceCapable, boolean subclasses); Extent getExtent (Class persistenceCapable); A12.5.4-2, A12.5.4-3 [The getExtent method returns an Extent that contains all of the instances in the parameter class or interface, and if the subclasses flag is true, all of the instances of the parameter class and its subclasses.] The method with no subclasses parameter is treated as equivalent to getExtent (persistenceCapable, true).]See spreadsheet for more info.			]]></ac:plain-text-body></ac:structured-macro>
<ac:structured-macro ac:name="unnmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="4c64f2d6-234a-499e-acd1-75e1d931521f"><ac: plain-text-body><![CDATA[	PM	A12.7.1-1	A12.7.1-1 [/** Add the fetchgroup to the set of active fetch groups. Duplicate names will be removed.*/ Fetch([Plan addGroup(String fetchGroupName); /** Remove the fetch group from the set active fetch groups.*/ Fetch]]Plan removeGroup(String fetchGroupName); /** Remove all active groups, including the default fetch group.*/ FetchPlan clearGroups(); /** Return an immutable Set of the names of all active fetch groups.*/ Set getGroups(); /** Set a Collection of group names to replace the current groups. Duplicate names will be removed.*/ FetchPlan setGroups(Collection fetchGroupNames); /** Set an array of group names to replace the current groups. Duplicate names will be removed.*/ FetchPlan setGroups(String[] fetchGroupNames); /** Set a single group to replace the current groups.*/ Fetch Plan setGroup(String fetchGroupName);]]			]]></ac:plain-text-body></ac:structured-macro>

<ac:structured-macro ac:name="unnmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="8aaf3739- 9c30-48b6-8681- 9a93af4320e6"><ac: plain-text-body><![CDATA[	PM	A12.7.1-2	A12.7.1-2 [/** Set the fetch size for large result set support. */ Fetch`Plan setFetchSize(int fetchSize); /** Return the fetch size; 0 if not set; -1 for greedy fetching. */ int getFetchSize();]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unnmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="652b097e- 0283-4520-befc- eb02150e6302"><ac: plain-text-body><![CDATA[	PM	A12.7.1-3	A12.7.1-3 [/** Set detachment options */ Fetch`Plan setDetachmentOptions(int options); /** Return the detachment options */ int getDetachmentOptions();]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unnmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="ce7060d5- 4cef-4600-bd56- 6ad24c8eaff8"><ac: plain-text-body><![CDATA[	PM	A12.7-3	A12.7-3 [FetchPlan getFetchPlan(); This method retrieves the FetchPlan associated with the PersistenceManager. It always returns the identical instance for the same PersistenceManager]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unnmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="8f89427a- 8401-42b8-b8a2- 441eb8bde0ef"><ac: plain-text-body><![CDATA[	PM	A12.8-2	A12.8-2 [void flush(); This method flushes all dirty, new, and deleted instances to the datastore. It has no effect if a transaction is not active. If a datastore transaction is active, this method synchronizes the cache with the datastore and reports any exceptions. If an optimistic transaction is active, this method obtains a datastore connection and synchronizes the cache with the datastore using this connection. The connection obtained by this method is held until the end of the transaction..]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>

<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="d000fd6-9f9a-4ca9-a4bf-365b47886ed0"><ac: plain-text-body><![CDATA[	PM	12.5.6-18	12.5.6-18 [Object getObjectById (Class cls, Object key); The getObjectById method attempts to find an instance in the cache with the derived JDO identity. The key parameter is either the string representation of the object id, or is an object repre-sentation of a single field identity key.]]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="1b8d742d-a03-4e98-8b53-a523ef5230ef"><ac: plain-text-body><![CDATA[	PM	12.5.6-17	12.5.6-17 [Collection getObjectsById (Collection oids); Object[] getObjectsById (Object[] oids); Collection getObjectsById (Collection oids, boolean validate); Object[] getObjectsById (Object[] oids, boolean validate); The getObjectsById method attempts to find instances in the cache with the specified JDO identities. The elements of the oids parameter object might have been returned by earlier calls to getObjectId or getTransactionalObjectId, or might have been constructed by the application. If a method with no validate parameter is used, the method behaves exactly as the corresponding method with the validate flag set to true. If the Object[] form of the method is used, the returned objects correspond by position with the object ids in the oids parameter. If the Collection form of the method is used, the iterator over the returned Collection returns instances in the same order as the oids returned by an iterator over the parameter Collection. The cardinality of the return value is the same as the cardinality of the oids parameter.]	J D O- 141	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="6fbf5ba1-70b9-44ab-ae9b-bf081dd2c35d"><ac: plain-text-body><![CDATA[	PM	A12.14-1	A12.14-1 [The Persistence{{ Manager provides a method to retrieve a Sequence by name. Sequence getSequence(String name); If the named sequence does not exist, JDOUser}} Exception is thrown.]]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac: schema-version="1" ac: macro-id="f1de7315-9dd5-497d-a130-9db1b740fb08"><ac: plain-text-body><![CDATA[	PM	A12.14-2	A12.14-2 [String getName(); This method returns the fully qualified name of the Sequence.]]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>

<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="97700c4d- f84d-46af-9dd1- acba3f1413e8"><ac: plain-text-body><! [CDATA[	PM	A12.14-3	A12.14-3 [Object next(); This method returns the next sequence value object. The sequence might be protected by transac-tional semantics, in which case the sequence value object will be reused if the transaction in which the sequence value object was obtained rolls back.]]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="8b818b6a- 271b-4b3c-95b8- ccb233a1987"><ac: plain-text-body><! [CDATA[	PM	A12.14-4	A12.14-4 [Object current(); This method returns the current sequence value object if it is available. It is intended to return a se-quence value object previously used.]]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="820f8026- 9438-41d3-b32b- aa627cbf8df8"><ac: plain-text-body><! [CDATA[	PM	A12.14-5	A12.14-5 [long nextValue(); This method returns the next sequence value as a long if it is available and able to be converted to a number].]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="7f264bc5- a2a2-4cc4-8fd8- 1e19568f3247"><ac: plain-text-body><! [CDATA[	PM	A12.14-6	A12.14-6 [long currentValue(); This method returns the current sequence value as a long if it is available and able to be converted to a number.]]	.	.	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>

<ac:structured-macro ac:name="unmigrated- wiki-markup" ac: schema-version="1" ac: macro-id="f6945a37- 0288-47fb-97ac- 4804f6abf946"><ac: plain-text-body><![CDATA[	PM	A12.16-1	A12.16-1 [JDOConnection getDataStoreConnection(); If this method is called while a datastore transaction is active, the object returned will be enlisted in the current transaction. If called in an optimistic transaction before flush has been called, or outside an active transaction, the object returned will not be enlisted in any transaction.]]	.	J D O- 191	]] > < /a c: pl ai n- te xt- bo dy > < /a c: str uc tu re d- m ac ro>
---	----	----------	--	---	---------------------	--