

org.apache.openjpa.persistence

Class EntityManagerImpl

[java.lang.Object](#)

└─ org.apache.openjpa.persistence.EntityManagerImpl

All Implemented Interfaces:

[EntityManager](#), [EntityTransaction](#), [Connection](#), [LocalTransaction](#), [CallbackModes](#), [AutoClear](#), [AutoDetach](#), [ConnectionRetainModes](#), [DetachState](#), [FindCallbacks](#), [OpCallbacks](#), [RestoreState](#), [Closeable](#), [OpenJPAEntityManager](#)

```
public class EntityManagerImpl
extends Object
implements OpenJPAEntityManager, FindCallbacks, OpCallbacks
```

Implementation of [EntityManager](#) interface.

Author:

Patrick Linskey, Abe White

Field Summary

Fields inherited from interface org.apache.openjpa.kernel.[ConnectionRetainModes](#)

[CONN_RETAIN_ALWAYS](#), [CONN_RETAIN_DEMAND](#), [CONN_RETAIN_TRANS](#)

Fields inherited from interface org.apache.openjpa.kernel.[DetachState](#)

[DETACH_ALL](#), [DETACH_FGS](#), [DETACH_LOADED](#)

Fields inherited from interface org.apache.openjpa.kernel.[RestoreState](#)

[RESTORE_ALL](#), [RESTORE_IMMUTABLE](#), [RESTORE_NONE](#)

Fields inherited from interface org.apache.openjpa.kernel.[AutoDetach](#)

[DETACH_CLOSE](#), [DETACH_COMMIT](#), [DETACH_NONTXREAD](#), [DETACH_ROLLBACK](#)

Fields inherited from interface org.apache.openjpa.kernel.[AutoClear](#)

[CLEAR_ALL](#), [CLEAR_DATASTORE](#)

Fields inherited from interface org.apache.openjpa.event.[CallbackModes](#)

[CALLBACK_FAIL_FAST](#), [CALLBACK_IGNORE](#), [CALLBACK_LOG](#), [CALLBACK_RETHROW](#), [CALLBACK_ROLLBACK](#)

Fields inherited from interface org.apache.openjpa.kernel.[OpCallbacks](#)

[ACT_CASCADE](#), [ACT_NONE](#), [ACT_RUN](#), [OP_ATTACH](#), [OP_DELETE](#), [OP_DETACH](#), [OP_EVICT](#), [OP_LOCK](#), [OP_NONTRANSACTIONAL](#), [OP_PERSIST](#), [OP_REFRESH](#), [OP_RELEASE](#), [OP_RETRIEVE](#), [OP_TRANSACTIONAL](#)

Constructor Summary

[EntityManagerImpl](#)([EntityManagerFactoryImpl](#) factory, [Broker](#) broker)
Constructor; supply factory and delegate.

Method Summary

void	addLifecycleListener (Object listener, Class ... classes) Register a listener for lifecycle-related events on the specified classes.
void	addTransactionListener (Object listener) Register a listener for transaction-related events.
void	begin ()
void	beginStore () Begins a store transaction if one isn't already started.
boolean	cancelAll () Cancel all pending data store statements.
void	clear ()
void	close ()
void	commit ()
void	commitAndResume () Issue a commit and then start a new transaction.
boolean	contains (Object entity)
boolean	containsAll (Collection entities) Whether the given objects are managed.
boolean	containsAll (Object ... entities) Whether the given objects are managed.
<T> Extent <T>	createExtent (Class <T> cls, boolean subclasses) Return an extent of the given class, optionally including subclasses.
<T> T	createInstance (Class <T> cls) Create a new instance of type cls.
Interaction	createInteraction ()
OpenJPAQuery	createNamedQuery (String name)
OpenJPAQuery	createNativeQuery (String query)
OpenJPAQuery	createNativeQuery (String query, Class cls)
OpenJPAQuery	createNativeQuery (String query, String mappingName)

OpenJPAQuery	createQuery (Query query) Create a new query from the given one.
OpenJPAQuery	createQuery (String query)
OpenJPAQuery	createQuery (String language, String query) Create a new query in the given language.
<T> T	detach (T entity) Detach the specified object from the entity manager.
Collection	detachAll (Collection entities) Detach the specified objects from the entity manager.
Object []	detachAll (Object ... entities) Detach the specified objects from the entity manager.
void	dirty (Object o, String field) Make the named field of the given object dirty.
void	dirtyClass (Class cls) Mark the given class as dirty within the current transaction.
boolean	equals (Object other)
void	evict (Object entity) Evict the given object.
void	evictAll () Evict all clean objects.
void	evictAll (Class cls) Evict all persistent-clean and persistent-nontransactional instances in the extent of the given class (including subclasses).
void	evictAll (Collection entities) Evict the given objects.
void	evictAll (Extent extent) Evict all persistent-clean and persistent-nontransactional instances in the given Extent .
void	evictAll (Object ... entities) Evict the given objects.
<T> T	find (Class <T> cls, Object oid)
<T> Collection <T>	findAll (Class <T> cls, Collection oids) Return the objects with the given oids.
<T> T[]	findAll (Class <T> cls, Object ... oids) Return the objects with the given oids.
<T> T	findCached (Class <T> cls, Object oid) Return the cached instance for the given oid/object, or null if not cached.
void	flush ()
int	getAutoClear () Whether to clear state when entering a transaction.
int	getAutoDetach () Bit flags marked in AutoDetach which indicate when persistent managed objects should be automatically detached in-place.

Broker	getBroker() Broker delegate.
ClassLoader	getClassLoader() Return the current thread's class loader at the time this entity manager was obtained from the factory.
OpenJPAConfiguration	getConfiguration() Return the configuration associated with this entity manager.
Object	getConnection() Return the connection in use by the entity manager, or a new connection if none.
String	getConnectionPassword() Return the connection password.
int	getConnectionRetainMode() Return the connection retain mode for this entity manager.
String	getConnectionUserName() Return the connection user name.
Object	getDelegate()
int	getDetachState() Detach mode constant to determine which fields are part of the detached graph.
Collection	getDirtyObjects() Return a set of current dirty instances.
OpenJPAEntityManagerFactory	getEntityManagerFactory() Return the factory that produced this entity manager.
boolean	getEvictFromStoreCache() Whether to also evict an object from the store cache when it is evicted through this entity manager.
FetchPlan	getFetchPlan() Return the (mutable) fetch plan for loading objects from this entity manager.
Generator	getFieldGenerator(Class forClass, String fieldName) Returns a Generator for the generated values of the specified type, or null if the field is not generated.
FlushModeType	getFlushMode()
Generator	getIdGenerator(Class forClass) Returns a Generator for the datastore identity values of the specified type, or null if the type is unmanaged or its identity cannot be represented by a sequence.
boolean	getIgnoreChanges() Whether to take into account changes in the transaction when executing a query or iterating an extent.
int	getLifecycleListenerCallbackMode() The CallbackModes flags for handling lifecycle listener exceptions.
LocalTransaction	getLocalTransaction()
LockModeType	getLockMode(Object entity) Return the lock mode of the given instance, or null if not locked.

Collection	getManagedObjects() Return a set of all managed instances.
ManagedRuntime	getManagedRuntime() Return the managed runtime in use.
ConnectionMetaData	getMetaData()
boolean	getMultithreaded() Whether the entity manager or its managed instances are used in a multithreaded environment.
Generator	getNamedGenerator(String name) Return the named generator defined in the metadata.
boolean	getNontransactionalRead() Whether to allow nontransactional access to persistent state.
boolean	getNontransactionalWrite() Whether to allow nontransactional changes to persistent state.
Object	getObjectId(Object o) Return the oid of the given instance.
Class	getObjectIdClass(Class cls) Return the application identity class the given persistent class uses for object ids, or null if not a type that uses application identity.
boolean	getOptimistic() Whether to use optimistic transactional semantics.
boolean	getOrderDirtyObjects() Whether dirty objects will be returned in the order they were dirtied.
Collection	getPendingTransactionalObjects() Return a set of instances which will become transactional upon the next transaction.
Collection<Class>	getPersistedClasses() Return the set of classes that have been made persistent in the current transaction.
boolean	getPopulateStoreCache() Whether objects accessed during this transaction will be added to the store cache.
<T> T	getReference(Class<T> cls, Object oid)
Collection<Class>	getRemovedClasses() Return the set of classes that have been deleted in the current transaction.
int	getRestoreState() Whether to restore an object's original state on rollback.
ResultSetInfo	getResultSetInfo()
boolean	getRetainState() Whether objects retain their persistent state on transaction commit.
Throwable	getRollbackCause() Returns the Throwable that caused the transaction to be marked for rollback.

boolean	getRollbackOnly() Return whether the current transaction has been marked for rollback.
boolean	getSyncWithManagedTransactions() Whether to check for a global transaction upon every managed, non-transactional operation.
EntityTransaction	getTransaction()
Collection	getTransactionalObjects() Return a set of current transaction instances.
int	getTransactionListenerCallbackMode() The CallbackModes flags for handling transaction listener exceptions.
Collection < Class >	getUpdatedClasses() Return the set of classes for objects that have been modified in the current transaction.
Object	getUserObject(Object key) Get the value for the specified key from the map of user objects.
Object	getVersion(Object o) Returns the current version indicator for o.
int	hashCode()
boolean	isActive()
boolean	isDetached(Object entity) Returns true if pc is a detached object (one that can be reattached to a EntityManager via a call to EntityManager.merge(I)); otherwise returns false.
boolean	isDirty(Object o) Return whether the given object is dirty.
boolean	isLargeTransaction() Whether memory usage is reduced during this transaction at the expense of possibly more aggressive data cache evictions.
boolean	isManaged() Whether this entity manager is using managed transactions.
boolean	isNewlyPersistent(Object o) Return whether the given object was made persistent in the current transaction.
boolean	isOpen()
boolean	isPersistent(Object o) Return whether the given object is persistent.
boolean	isRemoved(Object o) Return whether the given object is deleted.
boolean	isStoreActive() Whether a store transaction is active.
boolean	isTransactional(Object o) Return whether the given object is transactional.
void	joinTransaction()

void	lock (Object entity) Ensure that the given instance is locked at the current lock level, as set in the FetchPlan for the entity manager.
void	lock (Object entity, LockModeType mode)
void	lock (Object entity, LockModeType mode, int timeout) Ensure that the given instance is locked at the given lock level.
void	lockAll (Collection entities) Ensure that the given instances are locked at the current lock level, as set in the FetchPlan for the entity manager.
void	lockAll (Collection entities, LockModeType mode, int timeout) Ensure that the given instances are locked at the given lock level.
void	lockAll (Object ... entities) Ensure that the given instances are locked at the current lock level, as set in the FetchPlan for the entity manager.
void	lockAll (Object [] entities, LockModeType mode, int timeout) Ensure that the given instances are locked at the given lock level.
<T> T	merge (T entity)
Collection	mergeAll (Collection entities) Merge the specified detached objects into the entity manager.
Object []	mergeAll (Object ... entities) Merge the specified objects into the entity manager.
void	nontransactional (Object entity) Make the given object nontransactional.
void	nontransactionalAll (Collection objs) Make the given objects nontransactional.
void	nontransactionalAll (Object [] objs) Make the given objects nontransactional.
void	persist (Object entity)
void	persistAll (Collection entities) Persist the given objects.
void	persistAll (Object ... entities) Persist the given objects.
void	preFlush () Run pre-flush actions on transactional objects, including persistence-by-reachability, inverse relationship management, deletion of dependent instances, and instance callbacks.
int	processArgument (int op, Object obj, OpenJPAStateManager sm) Process operation argument.
Object	processArgument (Object arg) Process find argument.
Object	processReturn (Object oid, OpenJPAStateManager sm) Process operation return value.
Object	putUserObject (Object key, Object val) Put the specified key-value pair into the map of user objects.

void	<code>refresh</code> (<code>Object</code> entity)
void	<code>refreshAll</code> () Refresh all transactional objects.
void	<code>refreshAll</code> (<code>Collection</code> entities) Refresh the state of the given objects.
void	<code>refreshAll</code> (<code>Object</code> ... entities) Refresh the state of the given objects.
void	<code>release</code> (<code>Object</code> entity) Release the given object from management.
void	<code>releaseAll</code> (<code>Collection</code> entities) Release the given objects from management.
void	<code>releaseAll</code> (<code>Object</code> ... entities) Release the given object from management.
void	<code>releaseSavepoint</code> () Release the last set savepoint and any resources associated with it.
void	<code>releaseSavepoint</code> (<code>String</code> name) Release the savepoint and any resources associated with it.
void	<code>remove</code> (<code>Object</code> entity)
void	<code>removeAll</code> (<code>Collection</code> entities) Delete the given persistent objects.
void	<code>removeAll</code> (<code>Object</code> ... entities) Delete the given persistent objects.
void	<code>removeLifecycleListener</code> (<code>Object</code> listener) Remove a listener for lifecycle-related events.
void	<code>removeTransactionListener</code> (<code>Object</code> listener) Remove a listener for transaction-related events.
void	<code>retrieve</code> (<code>Object</code> entity) Immediately load the given object's persistent fields.
void	<code>retrieveAll</code> (<code>Collection</code> entities) Retrieve the persistent state of the given objects.
void	<code>retrieveAll</code> (<code>Object</code> ... entities) Retrieve the persistent state of the given objects.
void	<code>rollback</code> ()
void	<code>rollbackAndResume</code> () Issue a rollback and then start a new transaction.
void	<code>rollbackToSavepoint</code> () Rollback the current transaction to the last savepoint.
void	<code>rollbackToSavepoint</code> (<code>String</code> name) Rollback the current transaction to the given savepoint name.
void	<code>setAutoClear</code> (int val) Whether to clear state when entering a transaction.
void	<code>setAutoDetach</code> (int flags) Bit flags marked in <code>AutoDetach</code> which indicate when persistent managed objects should be automatically detached in-place.

void	<code>setAutoDetach</code> (int flag, boolean on) Bit flags marked in <code>AutoDetach</code> which indicate when persistent managed objects should be automatically detached in-place.
void	<code>setDetachState</code> (int mode) Detach mode constant to determine which fields are part of the detached graph.
void	<code>setEvictFromStoreCache</code> (boolean evict) Whether to also evict an object from the store cache when it is evicted through this entity manager.
void	<code>setFlushMode</code> (<code>FlushModeType</code> flushMode)
void	<code>setIgnoreChanges</code> (boolean val) Whether to take into account changes in the transaction when executing a query or iterating an extent.
void	<code>setLargeTransaction</code> (boolean largeTransaction) If a large number of objects will be created, modified, or deleted during this transaction setting this option to true will reduce memory usage if you perform periodic flushes.
void	<code>setLifecycleListenerCallbackMode</code> (int mode) The <code>CallbackModes</code> flags for handling lifecycle listener exceptions.
void	<code>setMultithreaded</code> (boolean multithreaded) Whether the entity manager or its managed instances are used in a multithreaded environment.
void	<code>setNontransactionalRead</code> (boolean val) Whether to allow nontransactional access to persistent state.
void	<code>setNontransactionalWrite</code> (boolean val) Whether to allow nontransactional changes to persistent state.
void	<code>setOptimistic</code> (boolean val) Whether to use optimistic transactional semantics.
void	<code>setOrderDirtyObjects</code> (boolean order) Whether dirty objects will be returned in the order they were dirtied.
void	<code>setPopulateStoreCache</code> (boolean cache) Whether to populate the store cache with objects used by this transaction.
void	<code>setRestoreState</code> (int val) Whether to restore an object's original state on rollback.
void	<code>setRetainState</code> (boolean val) Whether objects retain their persistent state on transaction commit.
void	<code>setRollbackOnly</code> () Mark the current transaction for rollback.
void	<code>setRollbackOnly</code> (<code>Throwable</code> cause) Mark the current transaction for rollback with the specified cause of the rollback.
void	<code>setSavepoint</code> (<code>String</code> name) Set a transactional savepoint where operations after this savepoint will be rolled back.
void	<code>setSyncWithManagedTransactions</code> (boolean sync) Whether to check for a global transaction upon every managed, non-transactional operation.

void	setTransactionListenerCallbackMode (int mode) The CallbackModes flags for handling transaction listener exceptions.
void	transactional (Object entity, boolean updateVersion) Make the given object transactional.
void	transactionalAll (Collection objs, boolean updateVersion) Make the given objects transactional.
void	transactionalAll (Object [] objs, boolean updateVersion) Make the given objects transactional.
void	validateChanges () Validate the changes made in this transaction, reporting any optimistic violations, constraint violations, etc.

Methods inherited from class java.lang.[Object](#)

[clone](#), [finalize](#), [getClass](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

EntityManagerImpl

```
public EntityManagerImpl(EntityManagerFactoryImpl factory,
                        Broker broker)
```

Constructor; supply factory and delegate.

Method Detail

getBroker

```
public Broker getBroker()
```

Broker delegate.

getMetaData

```
public ConnectionMetaData getMetaData()
    throws ResourceException
```

Specified by:

[getMetaData](#) in interface [Connection](#)

Throws:

[ResourceException](#)

createInteraction

```
public Interaction createInteraction()
    throws ResourceException
```

Specified by:

[createInteraction](#) in interface [Connection](#)

Throws:

[ResourceException](#)

getLocalTransaction

```
public LocalTransaction getLocalTransaction()  
                                throws ResourceException
```

Specified by:

[getLocalTransaction](#) in interface [Connection](#)

Throws:

[ResourceException](#)

getResultSetInfo

```
public ResultSetInfo getResultSetInfo()  
                                throws ResourceException
```

Specified by:

[getResultSetInfo](#) in interface [Connection](#)

Throws:

[ResourceException](#)

getEntityManagerFactory

```
public OpenJPAEntityManagerFactory getEntityManagerFactory()
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the factory that produced this entity manager.

Specified by:

[getEntityManagerFactory](#) in interface [OpenJPAEntityManager](#)

getConfiguration

```
public OpenJPAConfiguration getConfiguration()
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the configuration associated with this entity manager.

Specified by:

[getConfiguration](#) in interface [OpenJPAEntityManager](#)

getFetchPlan

```
public FetchPlan getFetchPlan()
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the (mutable) fetch plan for loading objects from this entity manager.

Specified by:

[getFetchPlan](#) in interface [OpenJPAEntityManager](#)

getConnectionRetainMode

```
public int getConnectionRetainMode()
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the connection retain mode for this entity manager.

Specified by:

[getConnectionRetainMode](#) in interface [OpenJPAEntityManager](#)

See Also:

[ConnectionRetainModes](#)

isManaged

```
public boolean isManaged()
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether this entity manager is using managed transactions.

Specified by:

[isManaged](#) in interface [OpenJPAEntityManager](#)

getManagedRuntime

```
public ManagedRuntime getManagedRuntime()
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the managed runtime in use.

Specified by:

[getManagedRuntime](#) in interface [OpenJPAEntityManager](#)

getSyncWithManagedTransactions

```
public boolean getSyncWithManagedTransactions()
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to check for a global transaction upon every managed, non-transactional operation. Defaults to false.

Specified by:

[getSyncWithManagedTransactions](#) in interface [OpenJPAEntityManager](#)

setSyncWithManagedTransactions

```
public void setSyncWithManagedTransactions(boolean sync)
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to check for a global transaction upon every managed, non-transactional operation. Defaults to false.

Specified by:

[setSyncWithManagedTransactions](#) in interface [OpenJPAEntityManager](#)

getClassLoader

```
public ClassLoader getClassLoader()
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the current thread's class loader at the time this entity manager was obtained from the factory.

Specified by:

[getClassLoader](#) in interface [OpenJPAEntityManager](#)

getConnectionUserName

```
public String getConnectionUserName()
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the connection user name.

Specified by:

[getConnectionUserName](#) in interface [OpenJPAEntityManager](#)

getConnectionPassword

```
public String getConnectionPassword()
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the connection password.

Specified by:

[getConnectionPassword](#) in interface [OpenJPAEntityManager](#)

getMultithreaded

```
public boolean getMultithreaded()
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether the entity manager or its managed instances are used in a multithreaded environment.

Specified by:

[getMultithreaded](#) in interface [OpenJPAEntityManager](#)

setMultithreaded

```
public void setMultithreaded(boolean multithreaded)
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether the entity manager or its managed instances are used in a multithreaded environment.

Specified by:

[setMultithreaded](#) in interface [OpenJPAEntityManager](#)

getIgnoreChanges

```
public boolean getIgnoreChanges()
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to take into account changes in the transaction when executing a query or iterating an extent.

Specified by:

[getIgnoreChanges](#) in interface [OpenJPAEntityManager](#)

setIgnoreChanges

```
public void setIgnoreChanges(boolean val)
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to take into account changes in the transaction when executing a query or iterating an extent.

Specified by:

[setIgnoreChanges](#) in interface [OpenJPAEntityManager](#)

getNontransactionalRead

```
public boolean getNontransactionalRead()
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to allow nontransactional access to persistent state.

Specified by:

[getNontransactionalRead](#) in interface [OpenJPAEntityManager](#)

setNontransactionalRead

```
public void setNontransactionalRead(boolean val)
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to allow nontransactional access to persistent state.

Specified by:

[setNontransactionalRead](#) in interface [OpenJPAEntityManager](#)

getNontransactionalWrite

```
public boolean getNontransactionalWrite()
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to allow nontransactional changes to persistent state.

Specified by:

[getNontransactionalWrite](#) in interface [OpenJPAEntityManager](#)

setNontransactionalWrite

```
public void setNontransactionalWrite(boolean val)
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to allow nontransactional changes to persistent state.

Specified by:

[setNontransactionalWrite](#) in interface [OpenJPAEntityManager](#)

getOptimistic

```
public boolean getOptimistic()
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to use optimistic transactional semantics.

Specified by:

[getOptimistic](#) in interface [OpenJPAEntityManager](#)

setOptimistic

```
public void setOptimistic(boolean val)
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to use optimistic transactional semantics.

Specified by:

[setOptimistic](#) in interface [OpenJPAEntityManager](#)

getRestoreState

```
public int getRestoreState()
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to restore an object's original state on rollback.

Specified by:

[getRestoreState](#) in interface [OpenJPAEntityManager](#)

See Also:

[RestoreState](#)

setRestoreState

```
public void setRestoreState(int val)
```

Description copied from interface: [OpenJPAEntityManager](#)
Whether to restore an object's original state on rollback.

Specified by:

[setRestoreState](#) in interface [OpenJPAEntityManager](#)

See Also:

[RestoreState](#)

getRetainState

```
public boolean getRetainState()
```

Description copied from interface: [OpenJPAEntityManager](#)
Whether objects retain their persistent state on transaction commit.

Specified by:

[getRetainState](#) in interface [OpenJPAEntityManager](#)

setRetainState

```
public void setRetainState(boolean val)
```

Description copied from interface: [OpenJPAEntityManager](#)
Whether objects retain their persistent state on transaction commit.

Specified by:

[setRetainState](#) in interface [OpenJPAEntityManager](#)

getAutoClear

```
public int getAutoClear()
```

Description copied from interface: [OpenJPAEntityManager](#)
Whether to clear state when entering a transaction.

Specified by:

[getAutoClear](#) in interface [OpenJPAEntityManager](#)

See Also:

[AutoClear](#)

setAutoClear

```
public void setAutoClear(int val)
```

Description copied from interface: [OpenJPAEntityManager](#)
Whether to clear state when entering a transaction.

Specified by:

[setAutoClear](#) in interface [OpenJPAEntityManager](#)

See Also:

[AutoClear](#)

getDetachState

```
public int getDetachState()
```

Description copied from interface: [OpenJPAEntityManager](#)

Detach mode constant to determine which fields are part of the detached graph.

Specified by:

[getDetachState](#) in interface [OpenJPAEntityManager](#)

See Also:

[DetachState](#)

setDetachState

```
public void setDetachState(int mode)
```

Description copied from interface: [OpenJPAEntityManager](#)

Detach mode constant to determine which fields are part of the detached graph.

Specified by:

[setDetachState](#) in interface [OpenJPAEntityManager](#)

See Also:

[DetachState](#)

getAutoDetach

```
public int getAutoDetach()
```

Description copied from interface: [OpenJPAEntityManager](#)

Bit flags marked in [AutoDetach](#) which indicate when persistent managed objects should be automatically detached in-place.

Specified by:

[getAutoDetach](#) in interface [OpenJPAEntityManager](#)

setAutoDetach

```
public void setAutoDetach(int flags)
```

Description copied from interface: [OpenJPAEntityManager](#)

Bit flags marked in [AutoDetach](#) which indicate when persistent managed objects should be automatically detached in-place.

Specified by:

[setAutoDetach](#) in interface [OpenJPAEntityManager](#)

setAutoDetach

```
public void setAutoDetach(int flag,  
                          boolean on)
```

Description copied from interface: [OpenJPAEntityManager](#)

Bit flags marked in [AutoDetach](#) which indicate when persistent managed objects should be automatically detached in-place.

Specified by:

[setAutoDetach](#) in interface [OpenJPAEntityManager](#)

getEvictFromStoreCache

```
public boolean getEvictFromStoreCache()
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to also evict an object from the store cache when it is evicted through this entity manager.

Specified by:

[getEvictFromStoreCache](#) in interface [OpenJPAEntityManager](#)

setEvictFromStoreCache

```
public void setEvictFromStoreCache(boolean evict)
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to also evict an object from the store cache when it is evicted through this entity manager.

Specified by:

[setEvictFromStoreCache](#) in interface [OpenJPAEntityManager](#)

getPopulateStoreCache

```
public boolean getPopulateStoreCache()
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether objects accessed during this transaction will be added to the store cache. Defaults to true.

Specified by:

[getPopulateStoreCache](#) in interface [OpenJPAEntityManager](#)

setPopulateStoreCache

```
public void setPopulateStoreCache(boolean cache)
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether to populate the store cache with objects used by this transaction. Defaults to true.

Specified by:

[setPopulateStoreCache](#) in interface [OpenJPAEntityManager](#)

isLargeTransaction

public boolean **isLargeTransaction**()

Description copied from interface: [OpenJPAEntityManager](#)

Whether memory usage is reduced during this transaction at the expense of possibly more aggressive data cache evictions.

Specified by:

[isLargeTransaction](#) in interface [OpenJPAEntityManager](#)

setLargeTransaction

public void **setLargeTransaction**(boolean largeTransaction)

Description copied from interface: [OpenJPAEntityManager](#)

If a large number of objects will be created, modified, or deleted during this transaction setting this option to true will reduce memory usage if you perform periodic flushes.

Specified by:

[setLargeTransaction](#) in interface [OpenJPAEntityManager](#)

getUserObject

public [Object](#) **getUserObject**([Object](#) key)

Description copied from interface: [OpenJPAEntityManager](#)

Get the value for the specified key from the map of user objects.

Specified by:

[getUserObject](#) in interface [OpenJPAEntityManager](#)

putUserObject

public [Object](#) **putUserObject**([Object](#) key,
 [Object](#) val)

Description copied from interface: [OpenJPAEntityManager](#)

Put the specified key-value pair into the map of user objects. Use a value of null to remove the key.

Specified by:

[putUserObject](#) in interface [OpenJPAEntityManager](#)

addTransactionListener

public void **addTransactionListener**([Object](#) listener)

Description copied from interface: [OpenJPAEntityManager](#)

Register a listener for transaction-related events.

Specified by:

[addTransactionListener](#) in interface [OpenJPAEntityManager](#)

removeTransactionListener

```
public void removeTransactionListener(Object listener)
```

Description copied from interface: [OpenJPAEntityManager](#)

Remove a listener for transaction-related events.

Specified by:

[removeTransactionListener](#) in interface [OpenJPAEntityManager](#)

getTransactionListenerCallbackMode

```
public int getTransactionListenerCallbackMode()
```

Description copied from interface: [OpenJPAEntityManager](#)

The [CallbackModes](#) flags for handling transaction listener exceptions.

Specified by:

[getTransactionListenerCallbackMode](#) in interface [OpenJPAEntityManager](#)

setTransactionListenerCallbackMode

```
public void setTransactionListenerCallbackMode(int mode)
```

Description copied from interface: [OpenJPAEntityManager](#)

The [CallbackModes](#) flags for handling transaction listener exceptions.

Specified by:

[setTransactionListenerCallbackMode](#) in interface [OpenJPAEntityManager](#)

addLifecycleListener

```
public void addLifecycleListener(Object listener,  
                                Class... classes)
```

Description copied from interface: [OpenJPAEntityManager](#)

Register a listener for lifecycle-related events on the specified classes. If the classes are null, all events will be propagated to the listener.

Specified by:

[addLifecycleListener](#) in interface [OpenJPAEntityManager](#)

removeLifecycleListener

```
public void removeLifecycleListener(Object listener)
```

Description copied from interface: [OpenJPAEntityManager](#)

Remove a listener for lifecycle-related events.

Specified by:

[removeLifecycleListener](#) in interface [OpenJPAEntityManager](#)

getLifecycleListenerCallbackMode

```
public int getLifecycleListenerCallbackMode()
```

Description copied from interface: [OpenJPAEntityManager](#)

The [CallbackModes](#) flags for handling lifecycle listener exceptions.

Specified by:

[getLifecycleListenerCallbackMode](#) in interface [OpenJPAEntityManager](#)

setLifecycleListenerCallbackMode

```
public void setLifecycleListenerCallbackMode(int mode)
```

Description copied from interface: [OpenJPAEntityManager](#)

The [CallbackModes](#) flags for handling lifecycle listener exceptions.

Specified by:

[setLifecycleListenerCallbackMode](#) in interface [OpenJPAEntityManager](#)

getReference

```
public <T> T getReference(Class<T> cls,  
                           Object oid)
```

Specified by:

[getReference](#) in interface [EntityManager](#)

find

```
public <T> T find(Class<T> cls,  
                  Object oid)
```

Specified by:

[find](#) in interface [EntityManager](#)

findAll

```
public <T> T[] findAll(Class<T> cls,  
                       Object... oids)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the objects with the given oids.

Specified by:

[findAll](#) in interface [OpenJPAEntityManager](#)

oids - the oids of the objects to return

Returns:

the objects that were looked up, in the same order as the oids parameter.

See Also:

[EntityManager.find\(Class, Object\)](#)

findAll

```
public <T> Collection<T> findAll(Class<T> cls,  
                                Collection oids)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the objects with the given oids.

Specified by:

[findAll](#) in interface [OpenJPAEntityManager](#)

oids - the oids of the objects to return

Returns:

the objects that were looked up, in the same order as the oids parameter.

See Also:

[EntityManager.find\(Class, Object\)](#)

findCached

```
public <T> T findCached(Class<T> cls,  
                       Object oid)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the cached instance for the given oid/object, or null if not cached.

Specified by:

[findCached](#) in interface [OpenJPAEntityManager](#)

oid - the object's id

Returns:

the cached object, or null if not cached

getObjectIdClass

```
public Class getObjectIdClass(Class cls)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the application identity class the given persistent class uses for object ids, or null if not a type that uses application identity.

Specified by:

[getObjectIdClass](#) in interface [OpenJPAEntityManager](#)

getTransaction

```
public EntityTransaction getTransaction()
```

Specified by:

[getTransaction](#) in interface [EntityManager](#)

joinTransaction

```
public void joinTransaction()
```

Specified by:

[joinTransaction](#) in interface [EntityManager](#)

begin

```
public void begin()
```

Specified by:

[begin](#) in interface [EntityTransaction](#)

Specified by:

[begin](#) in interface [LocalTransaction](#)

commit

```
public void commit()
```

Specified by:

[commit](#) in interface [EntityTransaction](#)

Specified by:

[commit](#) in interface [LocalTransaction](#)

rollback

```
public void rollback()
```

Specified by:

[rollback](#) in interface [EntityTransaction](#)

Specified by:

[rollback](#) in interface [LocalTransaction](#)

commitAndResume

```
public void commitAndResume()
```

Description copied from interface: [OpenJPAEntityManager](#)

Issue a commit and then start a new transaction. This is identical to:

```
manager.commit (); manager.begin ();
```

except that the entity manager's internal atomic lock is utilized, so this method can be safely executed from multiple threads.

Specified by:

[commitAndResume](#) in interface [OpenJPAEntityManager](#)

See Also:

[EntityTransaction.commit\(\)](#), [EntityTransaction.begin\(\)](#)

rollbackAndResume

```
public void rollbackAndResume()
```

Description copied from interface: [OpenJPAEntityManager](#)

Issue a rollback and then start a new transaction. This is identical to:

```
manager.rollback (); manager.begin ();
```

except that the entity manager's internal atomic lock is utilized, so this method can be safely executed from multiple threads.

Specified by:

[rollbackAndResume](#) in interface [OpenJPAEntityManager](#)

See Also:

[EntityTransaction.rollback\(\)](#), [EntityTransaction.begin\(\)](#)

getRollbackCause

```
public Throwable getRollbackCause()
```

Description copied from interface: [OpenJPAEntityManager](#)

Returns the Throwable that caused the transaction to be marked for rollback.

Specified by:

[getRollbackCause](#) in interface [OpenJPAEntityManager](#)

Returns:

the Throwable, or null if none was given

getRollbackOnly

```
public boolean getRollbackOnly()
```

Description copied from interface: [OpenJPAEntityManager](#)

Return whether the current transaction has been marked for rollback.

Specified by:

[getRollbackOnly](#) in interface [EntityTransaction](#)

Specified by:

[getRollbackOnly](#) in interface [OpenJPAEntityManager](#)

setRollbackOnly

```
public void setRollbackOnly()
```

Description copied from interface: [OpenJPAEntityManager](#)

Mark the current transaction for rollback.

Specified by:

[setRollbackOnly](#) in interface [EntityTransaction](#)

Specified by:

[setRollbackOnly](#) in interface [OpenJPAEntityManager](#)

setRollbackOnly

```
public void setRollbackOnly(Throwable cause)
```

Description copied from interface: [OpenJPAEntityManager](#)

Mark the current transaction for rollback with the specified cause of the rollback.

Specified by:

[setRollbackOnly](#) in interface [OpenJPAEntityManager](#)

setSavepoint

```
public void setSavepoint(String name)
```

Description copied from interface: [OpenJPAEntityManager](#)

Set a transactional savepoint where operations after this savepoint will be rolled back.

Specified by:

[setSavepoint](#) in interface [OpenJPAEntityManager](#)

rollbackToSavepoint

```
public void rollbackToSavepoint()
```

Description copied from interface: [OpenJPAEntityManager](#)

Rollback the current transaction to the last savepoint. Savepoints set after this one will become invalid.

Specified by:

[rollbackToSavepoint](#) in interface [OpenJPAEntityManager](#)

rollbackToSavepoint

```
public void rollbackToSavepoint(String name)
```

Description copied from interface: [OpenJPAEntityManager](#)

Rollback the current transaction to the given savepoint name. Savepoints set after this one will become invalid.

Specified by:

[rollbackToSavepoint](#) in interface [OpenJPAEntityManager](#)

releaseSavepoint

```
public void releaseSavepoint()
```

Description copied from interface: [OpenJPAEntityManager](#)

Release the last set savepoint and any resources associated with it. The given savepoint and any set after it will become invalid.

Specified by:

[releaseSavepoint](#) in interface [OpenJPAEntityManager](#)

releaseSavepoint

```
public void releaseSavepoint(String name)
```

Description copied from interface: [OpenJPAEntityManager](#)

Release the savepoint and any resources associated with it. The given savepoint and any set after it will become invalid.

Specified by:

[releaseSavepoint](#) in interface [OpenJPAEntityManager](#)

flush

```
public void flush()
```

Specified by:

[flush](#) in interface [EntityManager](#)

preFlush

```
public void preFlush()
```

Description copied from interface: [OpenJPAEntityManager](#)

Run pre-flush actions on transactional objects, including persistence-by-reachability, inverse relationship management, deletion of dependent instances, and instance callbacks. Transaction listeners are not invoked.

Specified by:

[preFlush](#) in interface [OpenJPAEntityManager](#)

validateChanges

```
public void validateChanges()
```

Description copied from interface: [OpenJPAEntityManager](#)

Validate the changes made in this transaction, reporting any optimistic violations, constraint violations, etc. In a datastore transaction or a flushed optimistic transaction, this method will act just like [EntityManager.flush\(\)](#). In an optimistic transaction that has not yet begun a datastore-level transaction,

however, it will only report exceptions that would occur on flush, without retaining any datastore resources.

Specified by:

[validateChanges](#) in interface [OpenJPAEntityManager](#)

isActive

```
public boolean isActive()
```

Specified by:

[isActive](#) in interface [EntityTransaction](#)

isStoreActive

```
public boolean isStoreActive()
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether a store transaction is active.

Specified by:

[isStoreActive](#) in interface [OpenJPAEntityManager](#)

beginStore

```
public void beginStore()
```

Description copied from interface: [OpenJPAEntityManager](#)

Begins a store transaction if one isn't already started. The entity manager must already be in a logical transaction.

Specified by:

[beginStore](#) in interface [OpenJPAEntityManager](#)

contains

```
public boolean contains(Object entity)
```

Specified by:

[contains](#) in interface [EntityManager](#)

containsAll

```
public boolean containsAll(Object... entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Whether the given objects are managed.

Specified by:

[containsAll](#) in interface [OpenJPAEntityManager](#)

containsAll

public boolean **containsAll**([Collection](#) entities)

Description copied from interface: [OpenJPAEntityManager](#)

Whether the given objects are managed.

Specified by:

[containsAll](#) in interface [OpenJPAEntityManager](#)

persist

public void **persist**([Object](#) entity)

Specified by:

[persist](#) in interface [EntityManager](#)

persistAll

public void **persistAll**([Object](#)... entities)

Description copied from interface: [OpenJPAEntityManager](#)

Persist the given objects.

Specified by:

[persistAll](#) in interface [OpenJPAEntityManager](#)

persistAll

public void **persistAll**([Collection](#) entities)

Description copied from interface: [OpenJPAEntityManager](#)

Persist the given objects.

Specified by:

[persistAll](#) in interface [OpenJPAEntityManager](#)

remove

public void **remove**([Object](#) entity)

Specified by:

[remove](#) in interface [EntityManager](#)

removeAll

```
public void removeAll(Object... entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Delete the given persistent objects.

Specified by:

[removeAll](#) in interface [OpenJPAEntityManager](#)

removeAll

```
public void removeAll(Collection entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Delete the given persistent objects.

Specified by:

[removeAll](#) in interface [OpenJPAEntityManager](#)

release

```
public void release(Object entity)
```

Description copied from interface: [OpenJPAEntityManager](#)

Release the given object from management. This operation is not recursive.

Specified by:

[release](#) in interface [OpenJPAEntityManager](#)

releaseAll

```
public void releaseAll(Collection entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Release the given objects from management. This operation is not recursive.

Specified by:

[releaseAll](#) in interface [OpenJPAEntityManager](#)

releaseAll

```
public void releaseAll(Object... entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Release the given object from management. This operation is not recursive.

Specified by:

[releaseAll](#) in interface [OpenJPAEntityManager](#)

refresh

```
public void refresh(Object entity)
```

Specified by:

[refresh](#) in interface [EntityManager](#)

refreshAll

```
public void refreshAll()
```

Description copied from interface: [OpenJPAEntityManager](#)

Refresh all transactional objects.

Specified by:

[refreshAll](#) in interface [OpenJPAEntityManager](#)

refreshAll

```
public void refreshAll(Collection entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Refresh the state of the given objects.

Specified by:

[refreshAll](#) in interface [OpenJPAEntityManager](#)

refreshAll

```
public void refreshAll(Object... entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Refresh the state of the given objects.

Specified by:

[refreshAll](#) in interface [OpenJPAEntityManager](#)

retrieve

```
public void retrieve(Object entity)
```

Description copied from interface: [OpenJPAEntityManager](#)

Immediately load the given object's persistent fields. One might use this action to make sure that an instance's fields are loaded before transitioning it to transient. Note that this action is not recursive. Any related objects that are loaded will not necessarily have their fields loaded.

Specified by:

[retrieve](#) in interface [OpenJPAEntityManager](#)

retrieveAll

```
public void retrieveAll(Collection entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Retrieve the persistent state of the given objects.

Specified by:

[retrieveAll](#) in interface [OpenJPAEntityManager](#)

See Also:

[OpenJPAEntityManager.retrieve\(java.lang.Object\)](#)

retrieveAll

```
public void retrieveAll(Object... entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Retrieve the persistent state of the given objects.

Specified by:

[retrieveAll](#) in interface [OpenJPAEntityManager](#)

See Also:

[OpenJPAEntityManager.retrieve\(java.lang.Object\)](#)

evict

```
public void evict(Object entity)
```

Description copied from interface: [OpenJPAEntityManager](#)

Evict the given object.

Specified by:

[evict](#) in interface [OpenJPAEntityManager](#)

evictAll

```
public void evictAll(Collection entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Evict the given objects.

Specified by:

[evictAll](#) in interface [OpenJPAEntityManager](#)

evictAll

```
public void evictAll(Object... entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Evict the given objects.

Specified by:

[evictAll](#) in interface [OpenJPAEntityManager](#)

evictAll

```
public void evictAll()
```

Description copied from interface: [OpenJPAEntityManager](#)
Evict all clean objects.

Specified by:
[evictAll](#) in interface [OpenJPAEntityManager](#)

evictAll

```
public void evictAll(Class cls)
```

Description copied from interface: [OpenJPAEntityManager](#)
Evict all persistent-clean and persistent-nontransactional instances in the extent of the given class (including subclasses).

Specified by:
[evictAll](#) in interface [OpenJPAEntityManager](#)

evictAll

```
public void evictAll(Extent extent)
```

Description copied from interface: [OpenJPAEntityManager](#)
Evict all persistent-clean and persistent-nontransactional instances in the given [Extent](#).

Specified by:
[evictAll](#) in interface [OpenJPAEntityManager](#)

detach

```
public <T> T detach(T entity)
```

Description copied from interface: [OpenJPAEntityManager](#)
Detach the specified object from the entity manager.

Specified by:
[detach](#) in interface [OpenJPAEntityManager](#)

Parameters:
entity - the instance to detach

Returns:
the detached instance

detachAll

```
public Object[] detachAll(Object... entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Detach the specified objects from the entity manager.

Specified by:

[detachAll](#) in interface [OpenJPAEntityManager](#)

Parameters:

entities - the instances to detach

Returns:

the detached instances

detachAll

```
public Collection detachAll(Collection entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Detach the specified objects from the entity manager.

Specified by:

[detachAll](#) in interface [OpenJPAEntityManager](#)

Parameters:

entities - the instances to detach

Returns:

the detached instances

merge

```
public <T> T merge(T entity)
```

Specified by:

[merge](#) in interface [EntityManager](#)

mergeAll

```
public Object[] mergeAll(Object... entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Merge the specified objects into the entity manager.

Specified by:

[mergeAll](#) in interface [OpenJPAEntityManager](#)

Parameters:

entities - instances to import

Returns:

the re-attached instances

mergeAll

```
public Collection mergeAll(Collection entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Merge the specified detached objects into the entity manager.

Specified by:

[mergeAll](#) in interface [OpenJPAEntityManager](#)

Parameters:

entities - Collection of instances to import

Returns:

the re-attached instances

transactional

```
public void transactional(Object entity,  
                        boolean updateVersion)
```

Description copied from interface: [OpenJPAEntityManager](#)

Make the given object transactional.

Specified by:

[transactional](#) in interface [OpenJPAEntityManager](#)

Parameters:

entity - instance to make transactional

updateVersion - if true, the instance's version will be incremented at the next flush

transactionalAll

```
public void transactionalAll(Collection objs,  
                           boolean updateVersion)
```

Description copied from interface: [OpenJPAEntityManager](#)

Make the given objects transactional.

Specified by:

[transactionalAll](#) in interface [OpenJPAEntityManager](#)

Parameters:

objs - instances to make transactional

updateVersion - if true, the instance's version will be incremented at the next flush

transactionalAll

```
public void transactionalAll(Object[] objs,  
                           boolean updateVersion)
```

Description copied from interface: [OpenJPAEntityManager](#)

Make the given objects transactional.

Specified by:

[transactionalAll](#) in interface [OpenJPAEntityManager](#)

Parameters:

objs - instances to make transactional

updateVersion - if true, the instance's version will be incremented at the next flush

nontransactional

```
public void nontransactional(Object entity)
```

Description copied from interface: [OpenJPAEntityManager](#)

Make the given object nontransactional.

Specified by:

[nontransactional](#) in interface [OpenJPAEntityManager](#)

nontransactionalAll

```
public void nontransactionalAll(Collection objs)
```

Description copied from interface: [OpenJPAEntityManager](#)

Make the given objects nontransactional.

Specified by:

[nontransactionalAll](#) in interface [OpenJPAEntityManager](#)

nontransactionalAll

```
public void nontransactionalAll(Object[] objs)
```

Description copied from interface: [OpenJPAEntityManager](#)

Make the given objects nontransactional.

Specified by:

[nontransactionalAll](#) in interface [OpenJPAEntityManager](#)

getNamedGenerator

```
public Generator getNamedGenerator(String name)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the named generator defined in the metadata.

Specified by:

[getNamedGenerator](#) in interface [OpenJPAEntityManager](#)

getIdGenerator

```
public Generator getIdGenerator(Class forClass)
```

Description copied from interface: [OpenJPAEntityManager](#)

Returns a [Generator](#) for the datastore identity values of the specified type, or null if the type is unmanaged or its identity cannot be represented by a sequence.

Specified by:

[getIdGenerator](#) in interface [OpenJPAEntityManager](#)

getFieldGenerator

```
public Generator getFieldGenerator(Class forClass,  
                                   String fieldName)
```

Description copied from interface: [OpenJPAEntityManager](#)

Returns a [Generator](#) for the generated values of the specified type, or null if the field is not generated.

Specified by:

[getFieldGenerator](#) in interface [OpenJPAEntityManager](#)

createExtent

```
public <T> Extent<T> createExtent(Class<T> cls,  
                                  boolean subclasses)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return an extent of the given class, optionally including subclasses.

Specified by:

[createExtent](#) in interface [OpenJPAEntityManager](#)

createQuery

```
public OpenJPAQuery createQuery(String query)
```

Specified by:

[createQuery](#) in interface [EntityManager](#)

Specified by:

[createQuery](#) in interface [OpenJPAEntityManager](#)

createQuery

```
public OpenJPAQuery createQuery(String language,  
                                String query)
```

Description copied from interface: [OpenJPAEntityManager](#)

Create a new query in the given language.

Specified by:

[createQuery](#) in interface [OpenJPAEntityManager](#)

createQuery

```
public OpenJPAQuery createQuery(Query query)
```

Description copied from interface: [OpenJPAEntityManager](#)

Create a new query from the given one.

Specified by:

[createQuery](#) in interface [OpenJPAEntityManager](#)

createNamedQuery

public [OpenJPAQuery](#) createNamedQuery([String](#) name)

Specified by:

[createNamedQuery](#) in interface [EntityManager](#)

Specified by:

[createNamedQuery](#) in interface [OpenJPAEntityManager](#)

createNativeQuery

public [OpenJPAQuery](#) createNativeQuery([String](#) query)

Specified by:

[createNativeQuery](#) in interface [EntityManager](#)

Specified by:

[createNativeQuery](#) in interface [OpenJPAEntityManager](#)

createNativeQuery

public [OpenJPAQuery](#) createNativeQuery([String](#) query,
[Class](#) cls)

Specified by:

[createNativeQuery](#) in interface [EntityManager](#)

Specified by:

[createNativeQuery](#) in interface [OpenJPAEntityManager](#)

createNativeQuery

public [OpenJPAQuery](#) createNativeQuery([String](#) query,
[String](#) mappingName)

Specified by:

[createNativeQuery](#) in interface [EntityManager](#)

Specified by:

[createNativeQuery](#) in interface [OpenJPAEntityManager](#)

setFlushMode

public void setFlushMode([FlushModeType](#) flushMode)

Specified by:

[setFlushMode](#) in interface [EntityManager](#)

getFlushMode

public [FlushModeType](#) getFlushMode()

Specified by:

[getFlushMode](#) in interface [EntityManager](#)

clear

```
public void clear()
```

Specified by:

[clear](#) in interface [EntityManager](#)

getDelegate

```
public Object getDelegate()
```

Specified by:

[getDelegate](#) in interface [EntityManager](#)

getLockMode

```
public LockModeType getLockMode(Object entity)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the lock mode of the given instance, or null if not locked.

Specified by:

[getLockMode](#) in interface [OpenJPAEntityManager](#)

lock

```
public void lock(Object entity,  
                LockModeType mode)
```

Specified by:

[lock](#) in interface [EntityManager](#)

lock

```
public void lock(Object entity)
```

Description copied from interface: [OpenJPAEntityManager](#)

Ensure that the given instance is locked at the current lock level, as set in the [FetchPlan](#) for the entity manager.

Specified by:

[lock](#) in interface [OpenJPAEntityManager](#)

lock

```
public void lock(Object entity,  
                LockModeType mode,  
                int timeout)
```

Description copied from interface: [OpenJPAEntityManager](#)

Ensure that the given instance is locked at the given lock level.

Specified by:

[lock](#) in interface [OpenJPAEntityManager](#)

Parameters:

entity - the object to lock

mode - the lock level to use

timeout - the number of milliseconds to wait for the lock before giving up, or -1 for no limit

lockAll

```
public void lockAll(Collection entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Ensure that the given instances are locked at the current lock level, as set in the [FetchPlan](#) for the entity manager.

Specified by:

[lockAll](#) in interface [OpenJPAEntityManager](#)

lockAll

```
public void lockAll(Collection entities,  
                   LockModeType mode,  
                   int timeout)
```

Description copied from interface: [OpenJPAEntityManager](#)

Ensure that the given instances are locked at the given lock level.

Specified by:

[lockAll](#) in interface [OpenJPAEntityManager](#)

Parameters:

entities - the objects to lock

mode - the lock level to use

timeout - the number of milliseconds to wait for the lock before giving up, or -1 for no limit

lockAll

```
public void lockAll(Object... entities)
```

Description copied from interface: [OpenJPAEntityManager](#)

Ensure that the given instances are locked at the current lock level, as set in the [FetchPlan](#) for the entity manager.

Specified by:

[lockAll](#) in interface [OpenJPAEntityManager](#)

lockAll

```
public void lockAll(Object[] entities,  
                  LockModeType mode,  
                  int timeout)
```

Description copied from interface: [OpenJPAEntityManager](#)

Ensure that the given instances are locked at the given lock level.

Specified by:

[lockAll](#) in interface [OpenJPAEntityManager](#)

Parameters:

entities - the objects to lock

mode - the lock level to use

timeout - the number of milliseconds to wait for the lock before giving up, or -1 for no limit

cancelAll

```
public boolean cancelAll()
```

Description copied from interface: [OpenJPAEntityManager](#)

Cancel all pending data store statements. If statements are cancelled while a flush is in progress, the transaction rollback only flag will be set.

Specified by:

[cancelAll](#) in interface [OpenJPAEntityManager](#)

Returns:

true if any statements were cancelled, false otherwise

getConnection

```
public Object getConnection()
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the connection in use by the entity manager, or a new connection if none.

Specified by:

[getConnection](#) in interface [OpenJPAEntityManager](#)

getManagedObjects

```
public Collection getManagedObjects()
```

Description copied from interface: [OpenJPAEntityManager](#)

Return a set of all managed instances.

Specified by:

[getManagedObjects](#) in interface [OpenJPAEntityManager](#)

getTransactionalObjects

public [Collection](#) **getTransactionalObjects()**

Description copied from interface: [OpenJPAEntityManager](#)

Return a set of current transaction instances.

Specified by:

[getTransactionalObjects](#) in interface [OpenJPAEntityManager](#)

getPendingTransactionalObjects

public [Collection](#) **getPendingTransactionalObjects()**

Description copied from interface: [OpenJPAEntityManager](#)

Return a set of instances which will become transactional upon the next transaction.

Specified by:

[getPendingTransactionalObjects](#) in interface [OpenJPAEntityManager](#)

getDirtyObjects

public [Collection](#) **getDirtyObjects()**

Description copied from interface: [OpenJPAEntityManager](#)

Return a set of current dirty instances.

Specified by:

[getDirtyObjects](#) in interface [OpenJPAEntityManager](#)

getOrderDirtyObjects

public boolean **getOrderDirtyObjects()**

Description copied from interface: [OpenJPAEntityManager](#)

Whether dirty objects will be returned in the order they were dirtied. Default is determined by the store manager.

Specified by:

[getOrderDirtyObjects](#) in interface [OpenJPAEntityManager](#)

setOrderDirtyObjects

public void **setOrderDirtyObjects**(boolean order)

Description copied from interface: [OpenJPAEntityManager](#)

Whether dirty objects will be returned in the order they were dirtied. Default is determined by the store manager.

Specified by:

[setOrderDirtyObjects](#) in interface [OpenJPAEntityManager](#)

dirtyClass

public void **dirtyClass**([Class](#) cls)

Description copied from interface: [OpenJPAEntityManager](#)

Mark the given class as dirty within the current transaction.

Specified by:

[dirtyClass](#) in interface [OpenJPAEntityManager](#)

getPersistedClasses

public [Collection](#)<[Class](#)> **getPersistedClasses**()

Description copied from interface: [OpenJPAEntityManager](#)

Return the set of classes that have been made persistent in the current transaction.

Specified by:

[getPersistedClasses](#) in interface [OpenJPAEntityManager](#)

getUpdatedClasses

public [Collection](#)<[Class](#)> **getUpdatedClasses**()

Description copied from interface: [OpenJPAEntityManager](#)

Return the set of classes for objects that have been modified in the current transaction.

Specified by:

[getUpdatedClasses](#) in interface [OpenJPAEntityManager](#)

getRemovedClasses

public [Collection](#)<[Class](#)> **getRemovedClasses**()

Description copied from interface: [OpenJPAEntityManager](#)

Return the set of classes that have been deleted in the current transaction.

Specified by:

[getRemovedClasses](#) in interface [OpenJPAEntityManager](#)

createInstance

public <T> T **createInstance**([Class](#)<T> cls)

Description copied from interface: [OpenJPAEntityManager](#)

Create a new instance of type cls. If cls is an interface or an abstract class whose abstract methods follow the JavaBeans convention, this method will create a concrete implementation according to the metadata that defines the class. If cls is a non-final concrete type that has metadata but does not implement [PersistenceCapable](#), this method will create a subclass of the type that does implement [PersistenceCapable](#), following the property-based persistent attribute access rules, or will raise an

exception if the class does not meet the requirements for subclassing. Otherwise, this will return an instance of the specified class.

Specified by:

[createInstance](#) in interface [OpenJPAEntityManager](#)

close

```
public void close()
```

Specified by:

[close](#) in interface [EntityManager](#)

Specified by:

[close](#) in interface [Connection](#)

Specified by:

[close](#) in interface [Closeable](#)

isOpen

```
public boolean isOpen()
```

Specified by:

[isOpen](#) in interface [EntityManager](#)

dirty

```
public void dirty(Object o,  
                 String field)
```

Description copied from interface: [OpenJPAEntityManager](#)

Make the named field of the given object dirty.

Specified by:

[dirty](#) in interface [OpenJPAEntityManager](#)

getObjectId

```
public Object getObjectId(Object o)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return the oid of the given instance.

Specified by:

[getObjectId](#) in interface [OpenJPAEntityManager](#)

isDirty

```
public boolean isDirty(Object o)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return whether the given object is dirty.

Specified by:

[isDirty](#) in interface [OpenJPAEntityManager](#)

isTransactional

```
public boolean isTransactional(Object o)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return whether the given object is transactional.

Specified by:

[isTransactional](#) in interface [OpenJPAEntityManager](#)

isPersistent

```
public boolean isPersistent(Object o)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return whether the given object is persistent.

Specified by:

[isPersistent](#) in interface [OpenJPAEntityManager](#)

isNewlyPersistent

```
public boolean isNewlyPersistent(Object o)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return whether the given object was made persistent in the current transaction.

Specified by:

[isNewlyPersistent](#) in interface [OpenJPAEntityManager](#)

isRemoved

```
public boolean isRemoved(Object o)
```

Description copied from interface: [OpenJPAEntityManager](#)

Return whether the given object is deleted.

Specified by:

[isRemoved](#) in interface [OpenJPAEntityManager](#)

isDetached

```
public boolean isDetached(Object entity)
```

Description copied from interface: [OpenJPAEntityManager](#)

Returns true if pc is a detached object (one that can be reattached to a [EntityManager](#) via a call to [EntityManager.merge\(T\)](#)); otherwise returns false.

Specified by:

[isDetached](#) in interface [OpenJPAEntityManager](#)

getVersion

```
public Object getVersion(Object o)
```

Description copied from interface: [OpenJPAEntityManager](#)

Returns the current version indicator for o.

Specified by:

[getVersion](#) in interface [OpenJPAEntityManager](#)

processArgument

```
public Object processArgument(Object arg)
```

Description copied from interface: [FindCallbacks](#)

Process find argument. Throw proper [OpenJPAException](#) for illegal value.

Specified by:

[processArgument](#) in interface [FindCallbacks](#)

Returns:

the id to look up, or null to ignore this argument

processReturn

```
public Object processReturn(Object oid,  
                           OpenJPASStateManager sm)
```

Description copied from interface: [FindCallbacks](#)

Process operation return value.

Specified by:

[processReturn](#) in interface [FindCallbacks](#)

Returns:

the object to return

processArgument

```
public int processArgument(int op,  
                           Object obj,  
                           OpenJPASStateManager sm)
```

Description copied from interface: [OpCallbacks](#)

Process operation argument. Throw proper [OpenJPAException](#) for illegal value.

Specified by:

[processArgument](#) in interface [OpCallbacks](#)

Parameters:

op - the operation constant

obj - the object passed to the operation

sm - the argument's state manager, or null if none

Returns:

the action to take on the argument

hashCode

```
public int hashCode()
```

Overrides:

[hashCode](#) in class [Object](#)

equals

```
public boolean equals(Object other)
```

Overrides:

[equals](#) in class [Object](#)

[Overview](#) [Package](#) **[Class](#)** [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

Copyright © 2006-2007 [Apache Software Foundation](#). All Rights Reserved.