| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TransactionalWriter.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/sql/rowset/spi/SyncResolver.html)   [**NEXT CLASS**](http://docs.google.com/javax/sql/rowset/spi/XmlReader.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/rowset/spi/TransactionalWriter.html)    [**NO FRAMES**](http://docs.google.com/TransactionalWriter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **javax.sql.rowset.spi**

Interface TransactionalWriter

**All Superinterfaces:** [RowSetWriter](http://docs.google.com/javax/sql/RowSetWriter.html)

public interface **TransactionalWriter**extends [RowSetWriter](http://docs.google.com/javax/sql/RowSetWriter.html)

A specialized interface that facilitates an extension of the standard SyncProvider abstract class so that it has finer grained transaction control.

If one or more disconnected RowSet objects are particating in a global transaction, they may wish to coordinate their synchronization commits to preserve data integrity and reduce the number of sychronization exceptions. If this is the case, an application should set the CachedRowSet constant COMMIT\_ON\_ACCEPT\_CHANGES to false and use the commit and rollback methods defined in this interface to manage transaction boundaries.

| **Method Summary** | |
| --- | --- |
| void | [**commit**](http://docs.google.com/javax/sql/rowset/spi/TransactionalWriter.html#commit())()            Makes permanent all changes that have been performed by the acceptChanges method since the last call to either the commit or rollback methods. |
| void | [**rollback**](http://docs.google.com/javax/sql/rowset/spi/TransactionalWriter.html#rollback())()            Undoes all changes made in the current transaction. |
| void | [**rollback**](http://docs.google.com/javax/sql/rowset/spi/TransactionalWriter.html#rollback(java.sql.Savepoint))([Savepoint](http://docs.google.com/java/sql/Savepoint.html) s)            Undoes all changes made in the current transaction made prior to the given Savepoint object. |

| **Methods inherited from interface javax.sql.**[**RowSetWriter**](http://docs.google.com/javax/sql/RowSetWriter.html) |
| --- |
| [writeData](http://docs.google.com/javax/sql/RowSetWriter.html#writeData(javax.sql.RowSetInternal)) |

| **Method Detail** |
| --- |

### commit

void **commit**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Makes permanent all changes that have been performed by the acceptChanges method since the last call to either the commit or rollback methods. This method should be used only when auto-commit mode has been disabled.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or the Connection object within this CachedRowSet object is in auto-commit mode

### rollback

void **rollback**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Undoes all changes made in the current transaction. This method should be used only when auto-commit mode has been disabled.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or the Connection object within this CachedRowSet object is in auto-commit mode

### rollback

void **rollback**([Savepoint](http://docs.google.com/java/sql/Savepoint.html) s)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Undoes all changes made in the current transaction made prior to the given Savepoint object. This method should be used only when auto-commit mode has been disabled.

**Parameters:**s - a Savepoint object marking a savepoint in the current transaction. All changes made before *s* was set will be undone. All changes made after *s* was set will be made permanent. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or the Connection object within this CachedRowSet object is in auto-commit mode

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TransactionalWriter.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/sql/rowset/spi/SyncResolver.html)   [**NEXT CLASS**](http://docs.google.com/javax/sql/rowset/spi/XmlReader.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/rowset/spi/TransactionalWriter.html)    [**NO FRAMES**](http://docs.google.com/TransactionalWriter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).