# Working with the CloudTran 2.0 Release

This release contains the CloudTran 2.0 executables together with documentation and sample programs.

New major features in this release are Multi-Version Concurrency Control (MVCC) support and the Replicator (transactional replication between remote data centers).

Coherence 12.1.2 was released during final testing of CloudTran 2.0 and so is not included in this build.

The local tests described here are specifically for the LLAPI, both the MVCC and the Pessimistic Locking (PL) versions and for TopLink Grid.

To perform realistic Replicator testing requires a multiple machine environment. If you would like further information on Replicator testing then please contact us.

Download the distribution from

www.CloudTran.com/downloads/CloudTran-Coherence/CloudTran2.0.zip

and unpack it.

**Directories** Underneath the top-level directory ( "`CloudTran...`" ) you will see:

| | |
|---|---|
| docs/ | The CloudTran Guide is at `CloudTranCoherenceGuide/index.html`<br>See Chapter 6 for a description the LLAPI (Low-Level API)<br>See Javadocs for details of the LLAPI.<br>See Chapter 7 for the Replicator. |
| examples/ | This is an Eclipse workspace holding the new CloudTran code and a number of code examples and unit tests |
| MySQL/ | Has a database with sample data. |

**Examples** The examples directory holds the following Eclipse projects:

| | |
|---|---|
| LLAPI_MVCC_UnitTests/ | Low-level API MVCC Unit Tests |
| LLAPI_PL_UnitTests/ | Low-level API Pessimistic Locking Tests |
| CTPerformanceTest/ | CloudTran performance Test |
| ChildActivities/ | Top Link Grid Testing |
| Jars/ | Holding project containing jars |

**Running**

To run any Test, do the following steps:

1. This download does not contain **`coherence.jar`** for Coherence 3.7.1. If you do not have it available, you must download it from

   http://www.oracle.com/technetwork/middleware/coherence/downloads/index.html

   Then copy `coherence.jar` into the directory

   examples/Jars/lib

   If you don't do this, there will be a build error when you start the first application in Eclipse.

2. If you plan to run the LLAPI_PL's Test 40 or the TLG Tests, you need to start the database. Start MySQL using MySQL/**startMySQL**.bat

   `MySQL/startMySQL.bat` starts MySQL; `MySQL/stopMySQL.bat` stops it.

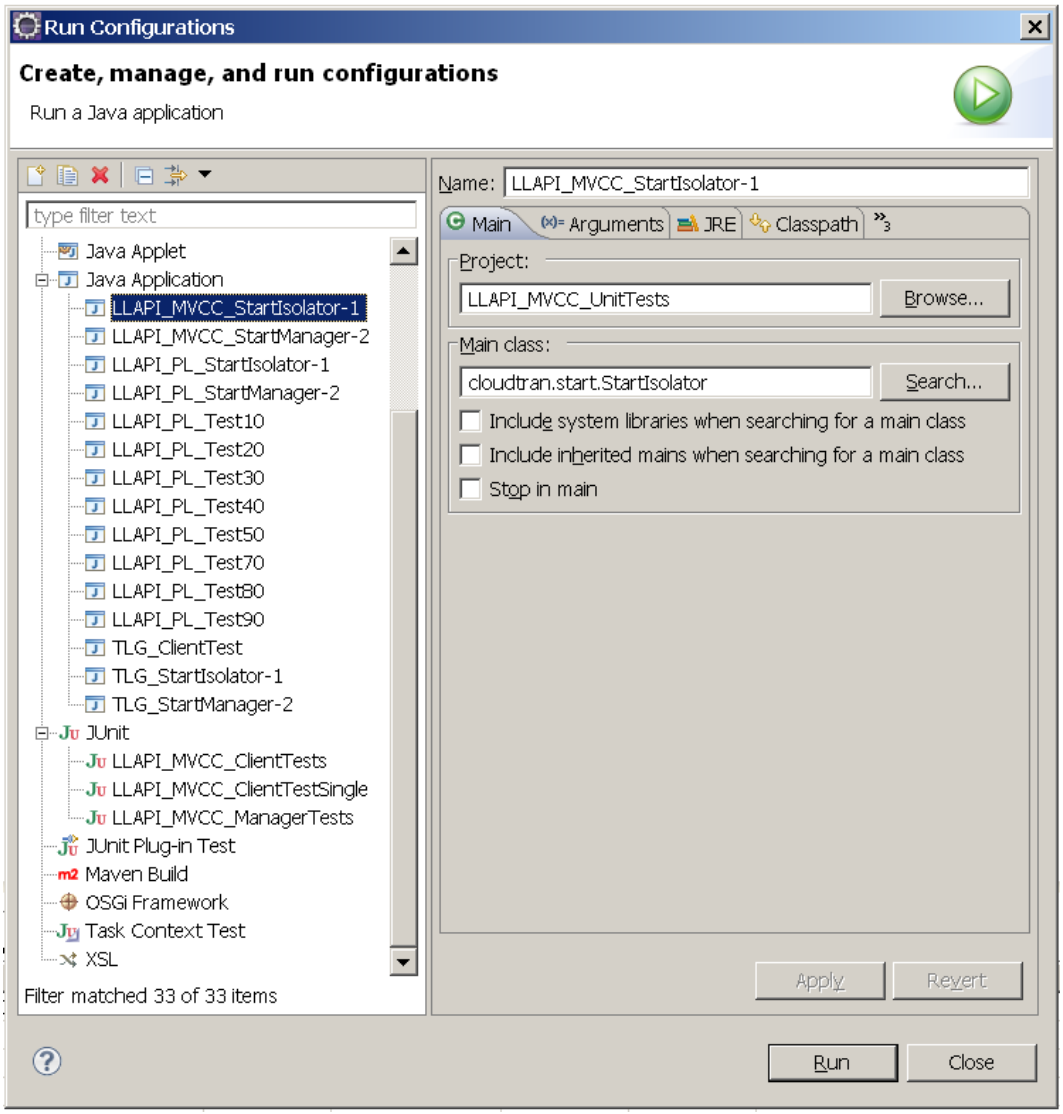
   The database is configured to use port 3306. Make sure that `startMySQL` doesn't exit prematurely – which it will do if you have MySQL already running on port 3306.

3. Start Eclipse. The "examples/" directory is the Eclipse workspace – so put that directory in when you start Eclipse.

   If you got step 1 wrong, you'll get 4 errors complaining about /Jars/lib/coherence.jar being missing. You can remedy this within Eclipse by pasting the coherence.jar file (using the OS' "Copy" facility) into the Jars/lib directory.

**Run Configurations**

There are a number of run configurations. Most are 'Java Applications'; the MVCC tests are in JUnit launches:



**LLAPI_MVCC Tests**

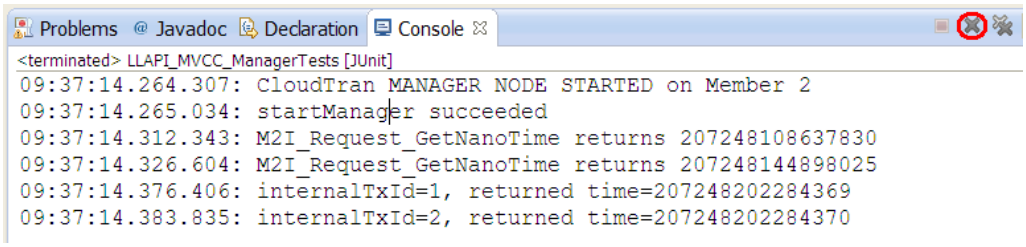
1. Start the "LLAPI_MVCC_StartIsolator-1" configuration first. Wait till you see the console message

   CloudTran ISOLATOR node started

   before continuing (about 30 seconds). Then leave this running – it is required for the LLAPI_MVCC* test configurations.

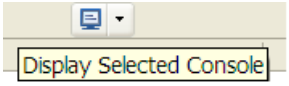
2. There is a manager test for MVCC timestamping. If you want to run that

   - start the LLAPI_MVCC_ManagerTests configuration. This should run through to conclusion using a special transaction manager and then terminate:




   *Sometimes output to the console window appears to gets lost. This is because Eclipse switches the windows as output is written.*

   *To look at the results of the run in the Eclipse console window, make sure you select the console window for the Test in question – not the Isolator or Manager – using this button*




   *If the console window doesn't match the expected output in the ReadMe.txt, check the test*_console.txt files in UnitTests/log.*

   - you can remove the process using the highlighted button on Console Window

3. To run other MVCC tests, you must start the "LLAPI_MVCC_StartManager-1" configuration. Wait till you see the console message

   CloudTran MANAGER node started

4. Now you can run the other LLAPI_MVCC* tests – these are in the JUnit area of the Run Configurations. The programs are in the LLAPI_MVCC_UnitTests Eclipse project, with the source in the `src/test/java` directory. The tests are as follows:

| | |
|---|---|
| LLAPI_MVCC_ClientTests | - Starting a transaction using Spring annotations |
| LLAPI_MVCC_ClientTestSingle | - Run a single test - selected in the 'test class' |

   **Note**: The Test_PutRemoveEP_MVCC_Spring needs to be run using the LLAPI_MVCC_ClientTestSingle. It will not work with the LLAPI_MVCC_ClientTests.

   Test output goes into the `LLAPI_MVCC_UnitTests/logs/` directory and also the console window. There are two types of output for each application – a '.txt' file and a '.log' file. The '.txt' file is more interesting for now – the .log file is mostly used for detailed trace.

**LLAPI_PL Testing**

The LLAPI_PL tests consist of a set of unit tests for the LLAPI Pessimistic Locking aspects of CloudTran.

1. If an Isolator or manager is still running from the MVCC tests, shut it down.
2. Start the "LLAPI_PL_StartIsolator-1" configuration first, then "LLAPI_PL_StartManager-2". Wait till you see the console message

   CloudTran ISOLATOR *[MANAGER]* node started

   before continuing (about 30 seconds for the Isolator, 2-3 for the Manager). Then leave them running – they are required for the LLAPI_PL* configurations.

   Normally, you won't have to restart these applications. (But if you run LLAPI_PL_Test40 without having the database started, you will have to bounce the Manager after starting MySQL.)

   > *If you run in Debug (rather than Run) mode, Spring throws internal exceptions for a file not found, which will be caught by the Eclipse debugger. There are 2 exceptions thrown in LLAPI_PL_StartIsolator-1 and LLAPI_PL_StartManager-1 and 6 in the example projects. Just hit F8 when this happens.*

3. Now you can run the different Test* configurations in the "Run/ Run configurations..." dialog. These use Spring for configuration.

   The programs are in the LLAPI_PL_UnitTest project in packages named like **ctTxSpringTests.test***. Each test has a ReadMe.txt file that describes what is being tested and the expected outcome. The tests are as follows:

   LLAPI_PL_Test10 - Starting a transaction using Spring annotations
   LLAPI_PL_Test20 - Transactional cache.put()/remove()
   LLAPI_PL_Test30 - Transactional EntryProcessor
   LLAPI_PL_Test40 - User-written persistence
   LLAPI_PL_Test50 - Use ManagerEvents via the ManagerEventListener for persistence
   LLAPI_PL_Test70 - Distributed transactions
   LLAPI_PL_Test80 –MapTriggers on transactional NamedCaches
   LLAPI_PL_Test90 - Shows a lock conflict

   Test output goes into the `LLAPI_PL_UnitTest/logs/` directory and also the console window. There are two types of output for each application – a '.txt' file and a '.log' file. The '.txt' file is more interesting for now – the .log file is mostly used for detailed trace.

**Replicator Testing**

The Replicator tests consist of the Replicator test environment and a set of unit tests for the SSD (LocalStore) of the Replicator. To run the Replicator tests

1. To run the Replicator Tests

| | |
|---|---|
| "Replicator_DC1_StartIsolator-1" | starts Data Centre 1 Isolator |
| "Replicator_DC1_StartIsolator-2_Replicator" | Starts Data Centre 1 Replicator |
| "Replicator_DC1_StartManager-1" | Starts Data Centre 1 Manager |
| "Replicator_DC1_StartManager-2" | Starts a second Manager (optional) |
| "Replicator_DC2_StartIsolator-1" | Starts Data Centre 1 Replicator |

| | |
|---|---|
| "Replicator_DC2_StartIsolator-1" | Starts Data Centre 2 Isolator |
| "Replicator_DC2_StartManager-1" | Starts Data Centre 2 Manager |
| "Replicator_DC2_StartManager-2" | Starts a second Manager (optional) |
| "Replicator_DC2_StartManager-1" | Starts Data Centre 2 Manager |

Wait until all the consoles are running before starting the client

| | |
|---|---|
| "Replicator_StartReplicatorTest" | Starts the Client |

**2.** To Run the SSD Local Unit Tests run the following under the JUnit test which can be found in the "Run/ Run configurations..." dialog.

| | |
|---|---|
| "Replicator_SSD_Local_Store" | Starts SSD Local Store Unit tests |

**Top-Link Grid Testing**

The Top-Link Grid test is a full application. The Application documentation can be found in the CloudTran Coherence Guide in the Example Application chapter. This application will require the database to be started. Details of this are described in the "**Running**" section. Once the database is started run the Isolator, Manager and Client as described below

**1.** Start the "TLG_StartIsolator-1" configuration first, then "TLG_StartManager-2". Wait till you see the console message

CloudTran ISOLATOR *[MANAGER]* node started

before continuing (about 30 seconds for the Isolator, 2-3 for the Manager). Then leave them running – they are required for the TLG ClientLLAPI_MVCC* configurations.

**2.** To start the client for the TLG run the configuration "TLG_ClientTest"

August 2013