TLC 2.0 Technical Installation Notes

Intended Audience

These notes are intended to be read by a technical person familiar with Java, Grails, Tomcat (or equivalent), SQL databases and your operating system. This person is referred to within the application as the System Administrator. The special System Administration options within the application are only available to the System Administrator who, through improper use of these facilities, can make the application inoperable. Obviously, therefore, the System Administrator must be a person, or persons, of considerable technical competence.

Minimum System Requirements

1. Java version 1.7 SE
2. 1.5GB of memory available to Java
3. Grails version 2.2.0
4. MySQL database server

Notes:

The project is configured to use the version of Grails shown above. If you are using a version of Grails later than that shown, you will need to run a Grails update command to bring the project up to your installed version of Grails.

If you use Java from Sun Microsystems Inc, then you should split the available memory to allow 512MB for PermGen space and 1GB to the heap with start up parameters such as: -Xms512m -Xmx1024m -XX:PermSize=256m -XX:MaxPermSize=512m. In production mode, this probably means editing the configuration files of your servlet container. In development mode, when you first want to try the system out, you can: 1) edit the 'run' settings in your IDE or 2) set your JAVA_OPTS in your environment or 3) edit the $GRAILS_HOME/bin/startGrails shell script (or .bat batch file) changing the standard JAVA_OPTS line to that required.

Although Grails uses Hibernate which should therefore allow the use of databases other than MySQL, no testing has yet been done with databases other than MySQL.

Configuration

Check the grails-app/conf/BuildConfig.groovy file within the tlc Grails project. Note that it is set to use Servlet version 3.0 (i.e. Tomcat 7) by default. The first entry in the dependencies section specifies that a MySQL connector should be downloaded and installed from Maven. This may or may not be appropriate for your installation. If you wish, you can either: 1) edit this line to something suitable for your installation or 2) remove the line altogether from BuildConfig.groovy and drop an appropriate database connector jar file in to the lib directory of the tlc Grails project.

Update the contents of the grails-app/conf/DataSource.groovy file to settings appropriate to your

database and check that the database connection pooling settings are as you require them. In particular, check that the validationQuery string is a valid SQL query statement for your database. The user that you use to connect to the database must have create tables and indexes privileges and also, optionally, the ability to perform databse adminstration operations. For development and testing purposes it is usually easiest to give the user all privileges within the target database although this might not be advisable in a production system. The database should be set up to store Unicode data (e.g. UTF-8) with an appropriate collation setting. In MySQL, the default Table Engine should be set to InnoDB.

Check through all the settings in grails-app/conf/Config.groovy. There are a number of settings specific to the TLC application which you might wish to consider and/or adjust.

The application comes as standard to use Ehcache as Hibernate's cache provider using Ehcache's default settings. You may wish to review this in light of your own requirements. To create your own Ehcache settings you may place an ehcache.xml configuration file in the grails-app/conf directory of the tlc Grails project.

Running The Application

TLC is a relatively large application and you would be best advised not to use the Grails in-memory database even for development or testing purposes. You should always connect to a 'real' database.

To run the application you must be connected to the Internet since the application will need to download its dependencies from Maven. When running the application, you will also need to be connected to the Internet if you intend to take advantage of the automatic update of foreign currency exchange rates from Google/Yahoo! that the application provides.

When running the application, ensure your browser has JavaScript enabled and, as far as possible, has its locale (language and country) set correctly.

It will take a few minutes (depending on the speed of your system) for TLC to fire up since it has a lot of tables to create and initial data to load – so be patient! Once the server is running, click on the 'Login as an existing user' option under the 'Fast Track' section of the home page. Log in as the system administrator using an id of system and password of sysadmin. You should now change the login id and password of the system administrator and, optionally, create a secondary system administrator 'just in case'.

After a couple of minutes (there is an intial start-up delay built in), have a look at the Task Queue to see that the initial tasks executed without error. This is a good indicator that all is well with your installation. Have an in-depth look around the System Administration menu options – but don't actually do anything – if this is the first time you have seen them. Additional documentation is available from the Fast Track section of the application's home page.

Post Installation

The application can run in one of two modes: 'demo' and 'live'. Demo mode allows people to set up their own companies and 'play around' with the system and also 'cleans up' after them. This can be

very useful for a development installation and as a Testing or Training installation for end users. By default, however, a new system will boot up for the first time in 'live' mode. If you wish this to be a demo system, change the 'isDemoSystem' System Setting from false to true.

If you are using MySQL, we suggest that you use the Task Definition facility in the Company Administration option of the System company to give the 'maintain' task a 'Next Scheduled Run' date and time of the current date and time since the tables already contain a significant amount of data and performance will be improved by updating the index statistics of the tables.

The application includes an 'automatic payment of suppliers' facility which, if you intend to use it, requires an additional Groovy program writing to actually pass payment information from the company to your bank(s). A heavily commented skeleton version of such a program is included in the web-app/documentation directory of the tlc application in a file called PaymentService.groovy.

The System Company

When first installed, only one company is created by the application. This is a company called System and is very special. It is used to hold 'template' data for other 'real' companies that you will create. Consequently, be very careful that you never change any data belonging to the System company unless you are absolutely certain you know what you are doing.

If you change anything about the System company without a full understanding of the implications, then all bets are off!

Hand Over

The remaining instructions are for 'live' systems as opposed to 'demo' systems.

Create the company that you intend to run on the system – remember that company System is not a 'real' company. Ensure the country, language and currency settings are correct for the new company.

Create a new user who will be the Company Administrator for the new company (your accountant?). Fill in the login id, name, email, password and confirmation, security question and answer, country and language fields. Do not alter the other fields.

Give the new user access to the new company by creating a new company user for them, selecting the new company as the company to add them to.

Assign the Company Administrator role to the new user in the new company.

You can now log out and hand the system over to the new Company Administrator for use.

