



Calipso

Community Edition

Installation Manual

for Calipso Community Edition v.2.1.2

Table of Contents

1. Introduction.....	3
1.1. Prerequisites.....	3
1.1.1. Java Development Kit.....	3
1.1.2. Database.....	3
1.1.3. Servlet Container.....	3
1.1.4. Build Environment.....	3
1.1.4.1. Apache Maven.....	4
1.1.4.2. Apache Ant.....	4
1.1.4.3. Apache Subversion.....	4
2. Building the Community Version.....	5
2.1. Code Checkout.....	5
2.1.1. Checkout using the command line.....	5
2.1.2. Checkout using Eclipse.....	5
2.2. Building the calipso-war project.....	6
3. Custom Configuration.....	7
3.1. Calipso Home.....	7
3.1.1. Edit the "calipso-init.properties" file.....	7
3.1.2. Set a Servlet context parameter.....	7
3.1.3. Set a System / JVM parameter.....	7
3.2. Email Settings.....	7
3.3. PDF Fonts.....	8
3.4. Custom Database Configuration.....	8
3.4.1. MySQL.....	8
3.4.2. Oracle.....	9
3.4.3. Microsoft SQL Server.....	9
3.4.4. Using a JNDI Datasource.....	10
3.5. Custom Authentication Configuration.....	10
3.5.1. Configuring LDAP Authentication.....	10
3.5.2. Integrating with CAS for Single Sign On.....	10
3.5.2.1. Changes to web.xml.....	10
3.5.2.2. Changes to applicationContext-acegi-cas.xml.....	11
4. Community Support.....	11

1. Introduction

This document is an installation manual for the Community Edition of the Calipso platform. It provides a guide and information for building and deploying the Community Edition on your system.

This document does not apply to the Enterprise Edition. To configure the Enterprise Edition WAR archive or to use the Enterprise Edition Installer please follow the Installation Manual included in your distribution.

1.1. Prerequisites

To install and run Calipso Community Edition, you will need:

1. A Java Development Kit (JDK), version 1.5 or higher.
2. A relational database like MySQL and the appropriate JDBC driver
3. A Servlet Container like Apache Tomcat 5.5 or higher.
4. Apache Maven 2.1 and Apache Ant 1.7 or higher.

1.1.1. Java Development Kit

Calipso requires the Java Runtime Environment version 5 or higher. You can check the availability and version of Java in your system by opening a command line prompt (MS-DOS for windows or terminal for Unix/Linux). In your command line prompt type and execute the following:

```
java -version
```

If Java is installed, you should see a message like:

```
java version "1.5.0_13"  
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_13-b05)  
Java HotSpot(TM) Client VM (build 1.5.0_13-121, mixed mode)
```

If the version in the first line starts with "1.5" or greater, you can install Calipso. Otherwise, you can download Java from here: <http://www.java.com/download>

1.1.2. Database

Calipso requires an RDBMS to run. You only need to create the database, Calipso will create the tables when it starts for the first time. MySQL, Oracle, MSSQL and PostgreSQL are excellent choices. You will also need a proper JDBC driver for your RDBMS.

1.1.3. Servlet Container

Calipso is designed so that you can be up and running by just deploying the WAR file into your existing servlet 2.4 compliant container.

For example, if you have Apache Tomcat 5.5.X, you can either copy the WAR file into the [tomcat.home]/webapps folder or choose to upload the WAR file through the Tomcat console.

1.1.4. Build Environment

To build and deploy the Calipso Community Edition you will need Apache Maven version 2.2 or higher and Apache Ant installed.

1.1.4.1. Apache Maven

To check for Maven in your command line prompt type and execute the following:

```
mvn -version
```

If Maven is installed, you should see a message like:

```
Apache Maven 2.2.1 (rdebian-1)
Java version: 1.6.0_20
Java home: /usr/lib/jvm/jdk1.6.0_20/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux" version: "2.6.32-25-generic" arch: "i386" Family:
"unix"
```

Make sure the Maven version in the first line starts with 2.1 or greater.

1.1.4.2. Apache Ant

To check for Apache Ant in your command line prompt type and execute the following:

```
ant -version
```

If Ant is installed, you should see a message like:

```
Apache Ant version 1.7.1 compiled on September 8 2010
```

Make sure the Ant version is 1.7 or greater.

1.1.4.3. Apache Subversion

To check for Subversion in your command line prompt type and execute the following:

```
svn --version --quiet
```

If Subversion is installed, you should see just the version:

```
1.6.6
```

You can also use your preferred IDE to check out the code, for example Eclipse IDE or Netbeans.

2. Building the Community Version

2.1. Code Checkout

To build the Community Edition, you need to obtain the source code from the public repository. You can check out the code using SVN from the command line or your favorite IDE. For more information please see the following paragraphs and the public repository documentation over at google code:

<http://code.google.com/p/calipso/source>

2.1.1. Checkout using the command line

Use this command to anonymously check out the latest project source code:

```
svn checkout http://calipso.googlecode.com/svn/trunk/calipso-war
```

2.1.2. Checkout using Eclipse

Alternatively, you can use Eclipse or your favorite IDE to checkout the code.

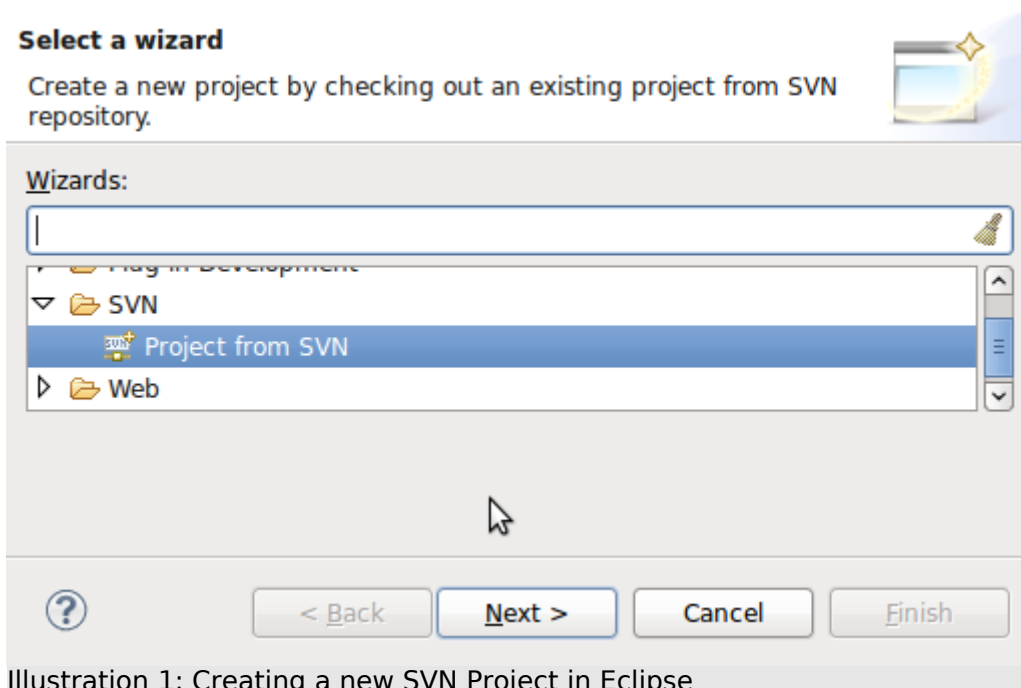
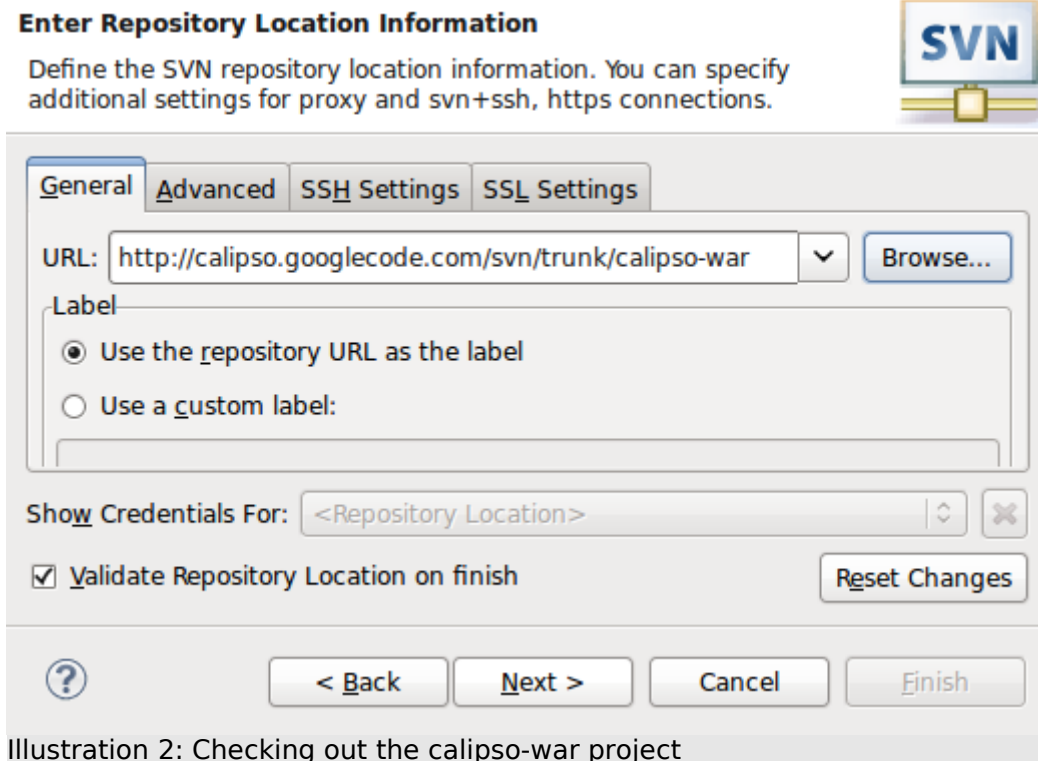


Illustration 1: Creating a new SVN Project in Eclipse



2.2. Building the calipso-war project

When you check out the project a folder calipso-war is created, unless you have chosen to create the project using a different name. Open the calipso-war/README.txt file and follow the instructions:

1. Create a database in your RDBMS named "calipso"
2. Add the appropriate JDBC driver in your servlet container's shared libraries, for example mysql-connector-java-5.1.12-bin.jar in TOMCAT_HOME/shared/lib
3. Create the calipso home directory in your home, e.g. /home/username/calipso
4. Copy the file calipso-war/calipso.properties in calipso home and edit it according to your environment, see "PDF Fonts"
5. Edit README.txt accordingly and RENAME it to "build.properties"
6. Open a console and navigate to the project directory:

```
cd /path/to/calipso-war
```

7. Run the following command to add required jars to your local Maven repo:

```
ant mvn-add-jars
```

8. Run the following command to generate build classpaths for Ant:

```
mvn antprops:generate
```

9. If you use Eclipse, configure the project with the following, then refresh the project:

```
mvn org.apache.maven.plugins:maven-eclipse-plugin:2.6:eclipse
```

10. Build and deploy the project:

```
ant tomcat-reload
```

The latter will suffice to re-deploy the application if needed.

3. Custom Configuration

3.1. Calipso Home

Calipso saves database information and uploaded attachments into a directory on the server. This directory is logically called "calipso.home". On startup, Calipso tries to detect the value of "calipso.home", and checks the following configuration options in the order listed below so, to configure the location of calipso.home you can do one of the following:

3.1.1. Edit the "calipso-init.properties" file

/WEB-INF/classes/calipso-init.properties file and customize the location of "calipso.home".

Note that you can choose to deploy Calipso as an exploded-war and it is not mandatory that you re-package (zip) the WAR file after editing "calipso-init.properties".

3.1.2. Set a Servlet context parameter

For example you can specify the calipso.home using the Tomcat administration console or as follows using a <Context> parameter.

```
<Context docBase="${catalina.home}/calipso/calipso.war">
  <Parameter name="calipso.home" value="C:/data/calipso_home"/>
</Context>
```

3.1.3. Set a System / JVM parameter

If you can easily set a system property called "calipso.home" within the environment of your application server then you don't need to edit the WAR file.

Please look at the contents of "start.bat" to see an example of how the appropriate JVM (Java Virtual Machine) parameters can be set when deploying Calipso.

For example, if you are using Apache Tomcat 5.5.X, you can set an environment variable called JAVA_OPTS before invoking the startup script (startup.bat / startup.sh).

Here is an example for Tomcat 5.5.X on Windows:

```
set JAVA_OPTS=-Dcalipso.home=C:/data/calipso_home -Dfile.encoding=UTF-8
```

If "calipso.home" has not been configured in any of the above ways, Calipso will create a folder called ".calipso" in the user home directory and proceed to use that as "calipso.home".

3.2. Email Settings

Calipso needs an SMTP server for e-mail notifications and password recovery.

Users who have the System Administrator role will be able to access the Calipso configuration settings screen from a link on the "OPTIONS" screen. Note that you will need to provide values for the "mail.from" and the "calipso.url.base" properties also, for things like the hyperlinks within e-mail to work properly.

Calipso also supports mail servers that require authentication or secure connections using SSL. You can refer the descriptions of the available configuration property settings available from the settings screen as shown below.

If you have trouble getting e-mail to work, one of the things to watch out for is whether there is any firewall blocking the communication between Calipso and the mail server - for example you may have a firewall running on the machine where Calipso is installed.

It is very common for mail servers to be configured to prevent mail-relay requests from unknown applications or IP addresses so you may need to check with your mail server administrator.

The Calipso settings screen where you can configure an e-mail / SMTP server Calipso also can lookup and use a JNDI mail session (`javax.mail.Session`) if required.

If you provide a JNDI name, Calipso will ignore the SMTP server details provided.

3.3. PDF Fonts

Calipso allows you to customise PDF export (e.g. for assets) using templates. Templates produce well-formed XML or HTML markup and CSS that is then converted to PDF by the XHTMLRenderer library (also known as Flying Saucer). Calipso will try to embed fonts used by CSS in the produced PDF files to ensure it displays properly on any platform.

To allow Calipso to embed the fonts, those must be available in the `calipso.home/fonts` folder. For example, the following CSS rule:

```
*{font-family: arial;}
```

Assumes the arial font is available in the `calipso.home/fonts/ARIALUNI.TTF` file. Calipso will use UTF-8 encoding and left-to-right text flow by default.

3.4. Custom Database Configuration

You can configure the database that Calipso uses by editing the "calipso.properties" file that Calipso expects within the "calipso.home" folder. Note that if Calipso does not find a "calipso.properties" file in the expected location, Calipso creates a fresh one which is pre-configured for HSQLDB.

The contents of this file are as follows:

```
database.driver=org.hsqldb.jdbcDriver
database.url=jdbc:hsqldb:file:${calipso.home}/db/calipso
database.username=sa
database.password=
hibernate.dialect=org.hibernate.dialect.HSQLDialect
hibernate.show_sql=false
```

You can change the database values and the hibernate dialect to match the settings of a database that you have already.

When Calipso starts, it will connect to the database and create the schema if one does not already exist.

3.4.1. MySQL

If you edit the "calipso.properties" file, ensure that there are no trailing spaces in the entries shown above (especially when cutting and pasting from somewhere). This can save you a lot of frustration!

Don't forget to make the database driver available for the web-application. You can get one from <http://dev.mysql.com/downloads/connector/j/>. The database driver/connector is called something like `mysql-connector-java-x.y.z-bin.jar`

If you are using the bundled Jetty web-app server, this is as simple as copying the `mysql-connector-java-x.y.z-bin.jar` file into the "lib" folder along with the jetty jar files.

If you are using Tomcat, you can either place the `mysql-connector-java-x.y.z-bin.jar` file under "shared/lib" if you are using version 5.x or under "lib" if you are using version 6.x or within the "/WEB-INF/lib" folder of the "calipso" (exploded) WAR itself.

Given below is a sample calipso.properties file for MySQL.

Using the default "root" user is obviously not advisable from a security point of view and you should try and have a separate user for the database used by Calipso.

```
database.driver=com.mysql.jdbc.Driver
database.url=jdbc:mysql://localhost/calipso
database.username=root
database.password=
hibernate.dialect=org.hibernate.dialect.MySQLDialect
hibernate.show_sql=false
```

For MySQL, do not use the "autoReconnect=true" parameter in the MySQL URL, this is not recommended by the MySQL team.

The Apache DBCP connection pool used internally by Calipso is configured to refresh stale database connections.

The default query to test idle database connections is "SELECT 1". This has been reported to not work for databases such as Derby and Progress because they consider this invalid SQL.

You can customize the validation query by adding a "database.validationQuery" property in "calipso.properties".

If you set the validation query to an empty string, this effectively disables the periodic testing of idle database connections that Apache DBCP performs.

Note 1 The database Calipso, should be first created by user before the first deploy. It is strongly recommended that on database creation, the value of the default character set should be 'utf8' and the value of default collate should be 'utf8_general_ci'.

3.4.2. Oracle

Sample calipso.properties file for Oracle.

```
database.driver=oracle.jdbc.driver.OracleDriver
database.url=jdbc:oracle:thin:@localhost:1521:xe
database.username=calipso
database.password=calipso
hibernate.dialect=org.hibernate.dialect.Oracle9Dialect
hibernate.show_sql=false
database.validationQuery=SELECT 1 FROM DUAL
```

3.4.3. Microsoft SQL Server

```
database.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
database.url=jdbc:sqlserver://localhost:1433;databaseName=calipso;
database.username=sa
database.password=
hibernate.dialect=org.hibernate.dialect.SQLServerDialect
hibernate.show_sql=true
```

Note 2 If you are using the Express edition, go to the Configuration Manager, choose "Protocols for SQLEXPRESS" and double-click (or focus and press enter) on "TCP-IP". Go to the "IP Addresses" tab, scroll down to the "IPALL" section and set the value of TCP Port to 1433 if it is empty.

3.4.4. Using a JNDI Datasource

You can provide a JNDI datasource name as an additional property called "database.datasource.jndiname" in the "calipso.properties" file. Calipso will ignore the other database properties in this case.

3.5. Custom Authentication Configuration

3.5.1. Configuring LDAP Authentication

A couple of extra entries in "calipso.properties" are required in order to configure LDAP authentication.

Calipso will attempt LDAP authentication if an "ldap.url" config entry is present.

```
ldap.url=ldap://myldaphost
ldap.searchBase=DC=foo,DC=bar,DC=org
```

The following additional entry is recommended if you are using Active Directory.

This switches on an LDAP optimization that is specific to Active Directory.

```
ldap.activeDirectoryDomain=MYDOMAIN
```

The benefit of LDAP authentication is that users can sign-on to Calipso by using their existing LDAP credentials (username and password). At the moment, LDAP support is limited to authentication and does not cover authorization. Users still have to be created by an administrator and mapped to the relevant Spaces using the Calipso administration screens before they can start using Calipso.

3.5.2. Integrating with CAS for Single Sign On

In order to integrate Calipso with CAS you have to edit a couple of XML files within the "WEB-INF" folder ('web.xml' and 'applicationContext-acegi-cas.xml').

The changes that you need to make are described below.

3.5.2.1. Changes to web.xml

- Change "applicationContext-acegi.xml" to "applicationContext-acegi-cas.xml". This appears towards the start of the file in the 'contextConfigLocation' context-param.
- Uncomment the filter and filter-mapping XML for the 'acegi' FilterToBeanProxy
- Uncomment the servlet and servlet-mapping XML for the 'casProxy' ProxyTicketReceptor. You should also uncomment the associated context-param called 'edu.yale.its.tp.cas.proxyUrl' and customize the value for your environment.

3.5.2.2. Changes to applicationContext-acegi-cas.xml

In the XML corresponding to bean id "casProxyTicketValidator" - edit the following property values to suit your environment:

- casValidate

- proxyCallbackUrl
- trustStore
- loginUrl
- logoutUrl

For more details and troubleshooting, you can refer to the Acegi documentation on CAS integration.

4. Community Support

Community Edition users are welcome to submit support requests, bug reports, fixes and feature requests at the project issue tracker:

<http://code.google.com/p/calipso/issues/list>