University of Zurich UZH

OLAT Installation Guide 7.2

The Essential Guide to Deploying OLAT

Preface

Since its inception over ten years ago OLAT (http://www.olat.org) has become one of the leading providers of Learning Management System software. It is the strategic learning platform for the University of Zürich, Switzerland, which continues to support the project. It has also been adopted by numerous public and private organizations worldwide. OLAT has proven itself to be a reliable, secure and scalable platform capable of supporting services that are used by tens of thousands of students every year.

This guide describes the installation of OLAT 7.2 and assumes that you are either performing an upgrade from 7.1 or are installing OLAT for the first time. Though we refer to the specifically to the 7.2.0.0 release you should be aware that patches do exist and we recommend that you always use the latest patch available. Throughout this publication we implicitly assume that the installation is being performed on a Linux host and accordingly some commands may need to be adapted for your target environment.

OLAT is a Java based web application that employs both database and filesystem for persistent storage needs. OLAT recommends the MySQL database which requires that you use the InnoDB storage engine. The application itself is packaged as a Web Archive (WAR) and deployment merely requires that it be placed in an appropriate web container such as Apache Tomcat. If OLAT is deployed in a load balanced topology then it is also necessary to install a JMS middleware component such as ActiveMQ. Optionally it is possible to integrate an XMPP instant messaging for which OLAT uses the Openfire server.

In this publication we refer to a Tomcat instance as a "node" and we differentiate between simple single node installations and more complex multi-node (i.e., load balanced) topologies. The latter are intended for large institutions with heavy loads and have been successfully deployed at the University of Zürich and elsewhere for the past several years.

Depending on your production environment it is likely that additional infrastructure components are necessary to provide load balancing, proxies, monitoring and other services. Please consult with your local system administration for guidance on such matters.

The OLAT Team.

Zürich, 10.06.2011

OLAT Demo Installer is a simple "single-click" installation for those wishing to get an initial impression of the application and to try out new features.

The OLAT Demo Installer is a simple means of installing OLAT in order to get a general first impression of the application and to evaluate new and existing features but is not intended for production use. The Demo Installer is shipped with a preconfigured Tomcat container and embedded HSQL database.

After downloading either double click on the JAR or invoke it via the following command:

```
%java –jar OLAT-LMS-7.2.0.0-demo-installer.jar
```

Install OLAT using the Wizard by accepting the license terms and selecting a directory in which the application and its infrastructure is to be installed. The Wizard may be terminated once the web application has successfully started.   OLAT can be accessed via the following URL:

```
                http://localhost:8080/olat/dmz
```

and will continue to operate until the process itself has been terminated.

Details of several pre-configured users (based on standard OLAT roles) can be found on the login page. Any of these users can be altered or removed by the administrator account.

Please note that since the Demo Installer does not reflect the OLAT reference production environment it is subjected to less rigorous testing and should be considered neither definitive nor suitable for anything other than evaluation purposes.

To remove the application, simply delete the entire contents of the installation directory.

Before installing OLAT you should ensure that you have an appropriate hosting environment commensurate with your needs. The reference environment requires a Tomcat 6.x web container (for which a Java SDK 1.6 must be installed) together with a MySQL 5.x database configured with an InnoDB storage engine. We refer you to the relevant product documentation for further information.

Begin your installation by downloading the OLAT Web Archive (WAR) file from the following location:

http://repo.olat.org/nexus/content/repositories/releases/org/olat/OLAT-LMS/OLAT-7.2.0.0

MySQL Database Setup

The first step in your installation is to log into your MySQL server as the "root" user and create an "olat" database and user "olat" substituting the details as appropriate in the following commands (see MySQL documentation for further information).

```
%mysql –u root –p –e "create database olat"
%mysql –u root –p olat –e "grant all privileges on olat.* to
              'olat'@'localhost' identified by 'olat'"
```

To create the database objects it is necessary to unpack the setupDatabase.sql script located within the WAR package by issuing the following command

```
%jar xf OLAT-LMS-7.2.0.0.war
     WEB-INF/classes/database/mysql/setupDatabase.sql
```

The script can be deployed using the following command:

```
%mysql –u olat –p olat < /path/to/setupDatabase.sql
```

Tomat Web Container Setup

The Tomcat container configuration, server.xml, requires that the context be made non-reloadable

```
<Context path="/olat" docBase="/path/to/olat/webapp"
    debug="0" reloadable="false" >
```

and that the default connector handle UTF-8 URI encoding

```
<Connector port="8080" minProcessors="2" maxProcessors="100"
      URIEncoding="utf-8" debug="0"/>
```

In addition it is advisable to configure your `log4j.xml` according to your needs for which a very simple template can be extracted from the WAR file using the following command:

```
%jar xf OLAT-LMS-7.2.0.0.war WEB-INF/classes/log4j.xml
```

Moreover your Tomcat runtime may require the following settings:

- `-Dlog4j.configuration`: pointing to the location of your `log4j.xml` configuration referenced via the `file:/` protocol.
- `-Djava.net.preferIPv4Stack=true`: forces use of IPv4 instead of IPv6 in order to avoid specific integration issues that sometimes arise in the context of 64bit hosts.
- `-Xss256k`: reduces the maximum thread stack allocation which can be necessary on heavily loaded systems.
- `-Xms/-Xmx`: heap configuration settings which should be set to c. 2-3G for heavily loaded systems.
- `-Djava.io.tmpdir`: determines the temporary directory to be used by the Tomcat process and should be configured in a manner appropriate to your needs.

OLAT Application Configuration

It remains merely to configure OLAT by setting the appropriate entries in the `olat.local.properties` file which should be placed on the classpath of the container (e.g., in the `lib` directory of a Tomcat installation.)   Any settings you apply here will override those set in the `olat.properties` file which is packaged in the WAR and which can be extracted for reference using the following command:

```
%jar xf OLAT-LMS-7.2.0.0.war
      WEB-INF/classes/serverconfig/olat.properties
```

The properties are self-explanatory and well-documented in the `olat.properties` file. As a minimum, however, you should consider setting the following in your `olat.local.properties`:

- `userdata.dir`: the root directory for all file based resources (e.g., course materials)
- `smtp.*`: SMTP mail host and in particular your support email addresses
- `db.name,port,user,pass`: Database host and user credentials configuration

There may be other properties (e.g., server port) that you would also like to alter and further details can be found in the default `olat.properties` file.

Deployment

Once the database, container and OLAT application `olat.local.properties` properties have been configured, it remains to deploy the OLAT WAR into the `webapps` directory of the Tomcat container in the usual fashion and to start up the application.

You may access the application via the following URL

<div align="center">

http://localhost:8080/olat

</div>

and log in using either the administrator account ("administrator" with password "olat")

The simple single node topology is adequate for most purposes and should be considered the default installation topology for the majority of providers. For large institutions (catering for the needs of thousands of students) it may, however, be necessary to deploy OLAT over several Tomcat instances fronted by a load balancer. In this configuration an additional ActiveMQ JMS broker component is required in order to support the inter-node communications that ensure synchronicity of access to underlying resources shared by the nodes. In addition if nodes are not collocated then it is also necessary to employ a distributed filesystem (e.g., NFS) accessible by all participating nodes.

All nodes in such a configuration should have the following properties set in their respective `olat.local.properties` files:

- `cluster.mode=Cluster`: enables multi-node features amongst the nodes
- `node.id`: each node must be assigned a unique numeric identifier starting with "1" and increasingly monotonically
- `*.jms.broker.url`: these settings reflect on your ActiveMQ configuration
- `hibernate.caching.cluster.class`: enables the use of the JBoss (distributed) tree cache within the cluster. See the sample `treecache.xml` configuration which is packaged within the WAR and note in particular the multicast network setting.
- `jmx.rmi.port`: must be configured to a unique port for each node in the topology.

In such installations it is necessary to assign specific duties to "backend" nodes into which perhaps ordinary users cannot log in (e.g., owing to proxy or load balancer configurations.) For example the search index generation may be delegated to such a node by enabling it only on that node. In particular some services may only be enabled on a single node and must be explicitly disabled on the other nodes:

- `cluster.singleton.services=enabled:` activates specific services (e.g., notifications, course logging etc.) that are not considered multi-node safe
- `search.service=enabled`: enables the fulltext search service on the designated node

In addition OLAT recommends that in order to optimally support live backup processes a distributed filesystem with snapshot capability be used and that the MySQL database be placed into a replicated database topology. We refer you to the relevant product documentation for further details.

OLAT can be integrated with the Openfire 3.x XMPP server to provide instant messaging services within the application. We refer to the Openfire product documentation on all matters relating to the installation and configuration of this server.

The first step to enabling instant messaging support in OLAT is to configure the `instantMessaging.*` properties in your `olat.local.properties` file. You can extract the default properties file from the OLAT WAR by issuing the following command:

```
%jar xf OLAT-LMS-7.2.0.0.war
     WEB-INF/classes/serverconfig/olat.properties
```

and use these as a template for your configuration.

Next it is necessary to install the OLAT Openfire plugin which can be downloaded from the following URL:

```
http://repo.olat.org/content/repositories/releases/org/olat/plugins/ola
t-openfire-plugin/7.2
```

You will need access to the Openfire administrator account and should refer to the Openfire documentation for further information on installing plugins.

On restart of the application a "buddy list" chat icon should be visible in the upper top right pane of the application.