Jalios JCMS 9.0

Installation and Operation Manual

© 2018 Jalios
v1.0.3
1 INTRODUCTION .................................................................................................................. 6
2 PREREQUISITES ................................................................................................................... 7
3 JCMS 9.0 ARCHITECTURE .................................................................................................. 8
   3.1 GENERAL ARCHITECTURE ......................................................................................... 8
   3.2 DATA STORAGE ............................................................................................................. 8
4 CONFIGURING THE INSTALLATION ENVIRONMENT .................................................. 10
   4.1 OPERATING SYSTEM ................................................................................................. 10
   4.2 JAVA VIRTUAL MACHINE (JVM) .............................................................................. 10
   4.3 CONFIGURING THE DATABASE ............................................................................... 10
   4.4 CHARACTER ENCODING .......................................................................................... 10
   4.5 FAQS ............................................................................................................................ 10
5 INSTALLING JCMS ON THE APPLICATION SERVER .................................................. 11
   5.1 GENERAL PRINCIPLES ............................................................................................. 11
   5.2 INSTALLATION ON APACHE TOMCAT 7.0 ............................................................. 11
   5.3 INSTALLATION ON IBM WEBSPHERE APPLICATION SERVER 8.5.5 ..................... 15
   5.4 INSTALLATION ON ORACLE WEBLOGIC SERVER 12C ........................................ 27
   5.5 INSTALLATION ON JBOSS EAP 5.2 ........................................................................ 30
6 STARTING AND CONFIGURING JCMS ........................................................................ 32
   6.1 STARTING THE APPLICATION .................................................................................... 32
   6.2 INITIAL CONFIGURATION OF THE APPLICATION .................................................. 33
   6.3 CONFIGURING THE DATABASE ............................................................................... 34
   6.4 CONFIGURING E-MAIL .............................................................................................. 35
   6.5 CONFIGURING LDAP ................................................................................................. 36
   6.6 CONFIGURING THE FORWARD PROXY ................................................................. 38
   6.7 CONFIGURING THE REVERSE PROXY ................................................................. 39
   6.8 CONFIGURING THE FORMAT OF DISPLAY URLs ................................................. 39
   6.9 CONFIGURING THE CONNECTION POOL .............................................................. 40
   6.10 CONFIGURING REPLICA BY JSYNC ...................................................................... 41
   6.11 INSTALLING AN ADD-PACK .................................................................................... 42
   6.12 CONFIGURING THE FRONTAL WEB SERVER ....................................................... 42
   6.13 INSTALLING JALIOS OFFICE LAUNCHER .............................................................. 42

March 2018
6.14 Adding Plugins ......................................................................................................................... 43

7 Operating Procedures .................................................................................................................. 44
  7.1 Starting and Stopping JCMS ...................................................................................................... 44
  7.2 Monitoring .................................................................................................................................. 45
  7.3 Data Integrity Check .................................................................................................................... 46
  7.4 Event Logs .................................................................................................................................... 47
  7.5 Stopping Writes .......................................................................................................................... 48
  7.6 Index Manager ............................................................................................................................ 49
  7.7 Cache Management ..................................................................................................................... 49
  7.8 Store Cleaner .............................................................................................................................. 49
  7.9 Backup/Restoration Procedures ................................................................................................. 51

8 Update and Deployment ................................................................................................................. 53
  8.1 General Procedure ...................................................................................................................... 53
  8.2 Developing and Preparing the war File ....................................................................................... 53
  8.3 Deploying the war on the Server .............................................................................................. 54
  8.4 Remarks ....................................................................................................................................... 55

9 Equipment and Configuration ........................................................................................................ 56
  9.1 Configuring the Server ............................................................................................................... 56
  9.2 Configuring the Workstations .................................................................................................... 58
This document describes the architecture, installation and configuration of Jalios JCMS 9.0.

The detailed installation procedures are described for various certified application servers.

The main operating procedures of the product are described, along with the update and deployment procedures.
2 Prerequisites

The prerequisite components described in this section are necessary for correct operation of Jalios JCMS 9.0. In the rest of this document, it is assumed that these have been installed and tested. See §9 for a detailed description of the hardware and software configurations.

Mandatory components:

- Application server: Java EE (Tomcat, Oracle WebLogic Server or IBM WebSphere)
- SMTP Host

Optional components:

- External Relational Database Management System (RDBMS): MySQL, Oracle, PostgreSQL, Microsoft SQL Server, IBM DB2
- Web server: Apache, IHS, etc.
- SSO server: CASE, RSA ClearTrust, etc.
- POP3 or IMAP server
3 JCMS 9.0 Architecture

3.1 General architecture

JCMS 9.0 is a Java EE 5 web application. It runs on an application server. Several instances of JCMS 9.0 can run on the same application server. The diagram below illustrates the general architecture of a JCMS 9.0 application.

![General architecture of JCMS 9.0](image)

Figure 1 - General architecture of JCMS 9.0

3.2 Data Storage

JCMS 9.0 incorporates a dual data storage mechanism. The data are shared between JStore and a relational database JcmsDB. Structure data (workspaces, groups, categories, portlets), members and content are managed in JStore. Technical data, archives, preferences and user content (forum, review, comment, survey, etc.) are stored in JcmsDB.

3.2.1 JStore

JStore is an object database whose persistence is assured by journalizing (logging) all changes. When the application starts, all objects are loaded into memory then the operations contained in the file store.xml are applied. During operation, when an object is created, modified or deleted, the operation describing this change is added to the journal. Only attributes of the objects are loaded into memory and stored in store.xml. This does not therefore concern files uploaded to the server, presentation templates (JSP), style sheets, etc. Two metrics are therefore associated with JStore: memory volume and loading time.

The memory volume depends on the number of live objects to load and their average size. The size of an object depends essentially on the volume of text it contains (typically a JCMS object without text field occupies less than 1 KB). JStore is based on the Java APIs in which the character strings are stored in 2-byte Unicode. The memory volume occupied is therefore a little more than twice the text volume. For example, an A4 page of text representing 3 KB of raw text on disk will occupy 7 KB (including the index) in JStore. A JCMS site managing 50,000 content items of this type will therefore require about 400 MB.
As a general rule of thumb:

- small objects (documentary sheet, category, portlet, member, etc.) occupy 3 KB of memory on average; 300,000 objects of this sort occupy about 1 GB;

- big objects (content items containing text fields) occupy 10 KB on average; 100,000 objects of this sort occupy about 1 GB.

The loading time depends on the number of operations logged in the journal store.xml. Creation operations are the longest to process. The loading time is proportional to the number and size of objects. For example, one journal of 50,000 objects each representing one A4 page of raw text (3 KB on disk) loads in 1 minute on a PC running at 2 GHz with 2 GB of RAM and 7200 rpm disks.

### 3.2.2 JcmsDB

JCMS 9.0 stores some of its data in a relational database, JcmsDB, managed by a RDBMS.

The choice of this RDBMS is highly structuring decision that must be made when creating the architecture. Jalios does not supply database conversion tools, so it is difficult to change this choice later on.

JCMS 9.0 is delivered with an embedded RDBMS called Derby. This is quite suitable for development and testing environments, but it must not be used in a production environment.

The creation of the JcmsDB base and the operating procedures (notably regular backups) are the responsibility of RDBMS Administrator. In an environment using Derby, JCMS handles the database creation and backups.

At the very first start-up, JCMS generates in JcmsDB all the necessary tables and indexes, regardless of the chosen RDBMS. This structure can evolve over time with the creation and modification of publication types and addition of plugins.
4 Configuring the Installation Environment

4.1 Operating System

Under Unix it is recommended to set the limit on the number of simultaneously open files to a value of at least 4096. Do not exceed the limit defined at kernel level (/proc/sys/fs/file-max), otherwise change this limit too.

Check the limits in a user shell that serves to launch the application server with the `ulimit` command.

Configure your system limits (/etc/limits or /etc/security/limits.conf).

On Linux, make sure that JCMS is installed on an ext3 file system.

4.2 Java Virtual Machine (JVM)

The application server must be configured to run with Oracle Java SE 6 (HotSpot) or IBM's 1.6 for WebSphere.

It is necessary to:

1. Increase the default value of the PermGenSpace via the attributes `-XX:PermSize` and `-XX:MaxPermSize`. It is recommended to set both these attributes to 256 MB.

2. Configure the memory allocated to the JVM. The allocation can be configured by specifying the parameters: `-Xmx` (maximum stack size) and `-Xms` (initial stack size). For JCMS it is recommended to set both these parameters to the same value in order to avoid reallocations when restarting the webapp.

Details of the configuration are shown in §5 when installing JCMS on the application server.

4.3 Configuring the Database

No configuration is necessary if you decide to use Derby, the JCMS integrated database manager. If you use an external RDBMS you must first install JCMS with Derby then follow the procedure described in §6.3.

4.4 Character Encoding

JCMS 9.0 uses UTF-8 character encoding. Therefore to edit JCMS files it is very important to use a text editor that supports UTF-8.

Configure your text editor to use UTF-8 without BOM (Byte Order Mark).

4.5 FAQs

For other questions on the environment configuration, consult the FAQs on Jalios Community (http://community.jalios.com/faq).
5 Installing JCMS on the Application Server

5.1 General Principles

JCMS 9.0 is a war file:

```
jcms-9.0.0.war
```

Installing JCMS consists in loading this file on the application server. It is indispensable that during this deployment the war file be decompressed since the application reads and writes data in its deployment folder.

Note: if the war file was downloaded using Internet Explorer, a .zip extension may have been added. Remove this extension before proceeding with the installation.

During the deployment, choose a context name associated with the application. This name is used to build the URL to access the site. In the installation procedures described below, we will use the context name jcms9.

To guarantee the security of your platform and be sure you always have latest corrective updates, we recommend that you systematically use the latest maintenance version of the application server (keeping the same major version indicated).

5.2 Installation on Apache Tomcat 7.0

Download the latest maintenance version of Tomcat 7.0 from http://tomcat.apache.org/

5.2.1 Configuring the Application Server

If JCMS is installed as a root webapp (ROOT/ folder), there may be access conflicts between resources located in folders bearing the same name in JCMS 9.0 and other webapps. In this case, the other webapps' folders must be renamed in order to ensure non-conflictual access in the presence of JCMS. For example, this is the case with the Tomcat administration webapp and the JCMS admin/ folder. Edit the file TOMCAT_DIR/conf/Catalina/.../admin.xml and modify the access path of the webapp by changing the path attribute of the Context tag to a different value, for example /adminTomcat.

Encoding UTF-8

To correctly support UTF-8 encoding, add `URIEncoding="UTF-8"` in all the connectors used in the server configuration (conf/server.xml).

Example with the HTTP connector:

```
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443"
    URIEncoding="UTF-8"
 />
```

Adding tools.jar

To enable JCMS 9.0 to generate and compile the classes of the types, the library tools.jar (containing the javac compiler) must be added to Tomcat.
Open the file `TOMCAT_DIR/bin/setclasspath.sh` and position at the every end of the file. Add the following lines:

```bash
# Jalios: tools.jar required to run JCMS
# Set standard CLASSPATH
CLASSPATH="$JAVA_HOME"/lib/tools.jar
```

**Under Windows:**

When Tomcat is started under Windows, change the `classpath` used by the service. To do this, use the graphic utility available in the Windows task bar (installed when service mode is selected). Modify the `classpath` (in the Java tab) by adding the file's path `C:\{jdk_path}\lib\tools.jar`, preceded by a semicolon (`;`).

If you start Tomcat under Windows using the launch script, open the file `TOMCAT_DIR\bin\setclasspath.bat` and position at line 75 before the line `goto end`

Add the following lines:

```bash
rem Jalios: tools.jar required to run JCMS
rem Set standard CLASSPATH
set "CLASSPATH=%JAVA_HOME%\lib\tools.jar"
```

**Configuring the PermGenSpace and the memory allocation**

It is necessary to increase the `PermGenSpace` and to configure the JVM memory allocation. Do this by setting the attributes via the environment variable `JAVA_OPTS`, either by adding the line below in the Tomcat launch scripts (`catalina.sh` and `catalina.bat`) or by setting the environment variable before executing the launch scripts.

With the Oracle JVM it is recommended to increase the `PermGenSpace`. Do this by setting attributes via the `JAVA_OPTS` environment variable in the Tomcat launch scripts.

**Under Unix:**

```bash
JAVA_OPTS="-XX:PermSize=256m -XX:MaxPermSize=256m -Xmx1024m -Xms1024m"
```

**Under Windows:**

```bash
set JAVA_OPTS=-XX:PermSize=256m -XX:MaxPermSize=256m -Xmx1024m -Xms1024m
```

If Tomcat is executed as a windows service, edit the properties via Programs > Apache Tomcat > Configure Tomcat, then in the Java tab:

- Set the value 1024 in the “Initial memory pool” and “Maximum memory pool” fields
- Add the `PermGenSpace` options in the "Java Options" field, after the existing values, specifying one option per line without spaces
  ```bash
  -XX:PermSize=256m
  -XX:MaxPermSize=256m
  ```
Configuring session cookies in HttpOnly

To prevent cookie theft via JavaScript, configure Tomcat such that the Java EE session cookie (JSESSIONID) sets the HttpOnly attribute.

To do this, modify the Tomcat default context configuration by adding the attribute `useHttpOnly="true"` in the `Context` tag of the file `conf/context.xml`:

```xml
<Context [...] useHttpOnly="true">
    [...] 
</Context>
```

Configuration in production mode

Tomcat is delivered with development mode activated. To improve the performance, modify the Tomcat file `conf/web.xml` to set the development parameter to false on the JspServlet:

```xml
<servlet>
    <servlet-name>jsp</servlet-name>
    <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
    [...] 
    <init-param>
        <param-name>development</param-name>
        <param-value>false</param-value>
    </init-param>
    [...] 
</servlet>
```
5.2.2 Deploying the Application

1. Stop Tomcat, if necessary,
   Under Windows in service mode: Programs > Apache Tomcat > Stop Tomcat;

2. Access the folder <TOMCAT_DIR>/webapps

3. Create a folder jcms9

4. Decompress the file jcms-9.0.0.war in this folder

5. Start Tomcat
   Under Windows in service mode: Programs > Apache Tomcat > Start Tomcat;

6. JCMS is now ready to run.
5.3 **Installation on IBM WebSphere Application Server 8.5.5**

**Prerequisite:** if you use the JVM J9 1.6 supplied by IBM, you must use version SR11 or later.

### 5.3.1 Deploying the Application

1. Start WebSphere: *Programs > IBM WebSphere > Application Server V8.5 > Profiles > Default > Start the server*

2. Open the Administrative Console: *Programs > IBM WebSphere > Application Server V8.5 > Profiles > Default > Administrative Console*

3. In the **Applications** menu click **New application**

4. Click on **New Enterprise Application**
5. Select **Local file system** then click **Browse**.... Find the right folder to select `jcms-9.0.0.war`, then click **Next**.

6. Select **Shortcut - Request only if additional information is required**, then click **Next**
7. Check the choice *Enable class reloading* and enter 2 in *Reload interval in seconds*. Click *Next*.

8. Click *Next*.
9. Click Next.

10. In the field Context root, enter the access URI of the webapp (e.g. /jcms9). Click Next.
11. Verify the summary, then click *Finish*.

12. After the installation, click *Save*.
5.3.2 Configuring Shared Libraries

WebSphere must now be configured to make it use the Derby and Jackson libraries version delivered with JCMS 9.0.

1. To do this, in the menu Environment > Shared library, choose Cell scope then click New.

2. Enter JcmsLibs in Name and in Class path enter the access paths to the following libraries supplied with JCMS;
   - WEB-INF/lib/derbynet.jar
   - WEB-INF/lib/jackson-core-lgpl-1.9.12.jar
   - WEB-INF/lib/jackson-mapper-lgpl-1.9.12.jar

   Note: the paths must be absolute, must not contain spaces and, under Windows, the backslash character "\" must be doubled. Enter one path per line. Check the box Use an isolated class loader for this shared library. Click OK.

   Note: if JCMS has already been started a first time in your WebSphere application server without this configuration, to ensure that the shared libraries are taken into account this server must be completely restarted (restarting the JCMS is not enough)

3. Save the configuration.

4. Open the menu Applications > Application types > WebSphere enterprise applications, then click on the JCMS application

5. In the Reference section, choose the option Shared library references.

6. Check the box to select the JCMS application and click Reference shared library, then click OK.
7. Select the *JemsLibs* shared library created in the previous stage in the left-hand column and add it to the selected libraries using the right-arrow button.

8. Click OK on the next two screens.

9. Save the configuration.
5.3.3 Configuring the Application Server

Configuring the Servlets Filters

WebSphere must now be configured so that it invokes the JCMS Servlets Filters in all circumstances:

1. Go to the menu Servers > Servers types > WebSphere application servers and select your server.

2. In the section Web Container Settings > Web Container Settings, choose the Web Container option.

3. Click Custom properties then New. Enter com.ibm.ws.webcontainer.invokeFiltersCompatibility in Name and true in Value. Click OK then Save.
Configuring the PermGenSpace and the memory allocation

It is necessary to increase the PermGenSpace (256 MB recommended) and to configure the JVM memory allocation (1024 MB recommended).

1. In the Administrative Console: Servers > Servers types > WebSphere application servers, click the server to configure.

2. In the Server Infrastructure section, open Process and Java Management > Process definition.

3. Finally, in the Additional properties section click Java virtual machine
   - Enter 1024 the fields Initial stack size and Maximum stack size
   - Enter the following parameters in the field Generic JVM arguments
     -XX:PermSize=256m -XX:MaxPermSize=256m

4. Click OK and save the configuration

See the WebSphere documentation for more details:
Configuring session cookie security

To prevent cookie theft via JavaScript, configure WebSphere such that the Java EE session cookie (JSESSIONID) sets the `HttpOnly` attribute.

To do this, set the following property to protect all cookies sent by the application:

```java
com.ibm.ws.webcontainer.httpOnlyCookies=*;
```

You will find the configuration procedure and the WebSphere versions required in this security warning:

http://www-01.ibm.com/support/docview.wss?uid=swg1PK98436

The detailed documentation of this property is available at the following address (search for "httpOnlyCookies"):


Importing certificates

If your JCMS application must make HTTPS requests to a remote web server, the remote server's intermediate certificates must be imported.

This is the case of the default JCMS 9 webapp which contains an RSS Portlet to Jalios news feeds on https://www.jalios.com/

If you do not do this, the following message may appear in the JCMS and WebSphere logs:

```
javax.net.ssl.SSLHandshakeException: com.ibm.jsse2.util.j: PKIX path building failed: java.security.cert.CertPathBuilderException: PKIXCertPathBuilderImpl could not build a valid CertPath.;
```

1. In the Administrative Console: Security -> SSL Certificate and Key Management -> Key and Certificate Stores -> NodeDefaultTrustStore
2. In the Additional properties section, click the Signatory Certificates link. Click the Extract a port button.

3. Enter the information on the remote site:
The remote site's domain name (e.g. www.jalios.com)
The port, generally 443
Choose a nickname, for example 'jalioscom'
Click Apply.
5.3.4 Starting the Application

To start the webapp, go to the menu Applications > Application types > WebSphere enterprise applications. Check the box next to jcms-9.0.0.war, then click Start.

JCMS is now ready to run.
5.4 Installation on Oracle WebLogic Server 12c

1. Decompress the war in a dedicated folder: jcms9

2. Move the following library in the newly decompressed JCMS webapp:

   WEB-INF/lib/xml-apis.jar → WEB-INF/lib-disabled/xml-apis.jar

   The presence of the folder lib-disabled with this library is required to allow the
   operation script to execute correctly (see §8 Update and Deployment)

3. Open the Administration Console. In the Deployments section click Install.

4. Select the jcms9 folder created in step 1, then click Next.
5. Select *Install this deployment as an application*, then click *Next*.

6. Leave the default choices, then click *Finish*. 
7. Wait for the deployment to complete and the application to start.

8. JCMS is now ready to run.

9. **Configure the PermGenSpace and the memory allocation**
   
   It is necessary to increase the *PermGenSpace* (256 MB recommended) and configure the JVM memory allocation (1024 MB recommended). To do this, refer to the Oracle WebLogic documentation:
   
   [http://docs.oracle.com/middleware/1213/wls/PERFM/jvm_tuning.htm](http://docs.oracle.com/middleware/1213/wls/PERFM/jvm_tuning.htm)
5.5 Installation on JBoss EAP 5.2

1. Configure JBoss EAP with correction of the issue [JBPAPP-8065]:
   To do this, add the following parameter in the parameters of the JVM starting JBoss:
   
   ```
   -Dorg.jboss.net.protocol.file.useURI=false
   ```

   Under Windows, open the file `jboss-eap-5.2\jboss-as\bin\run.conf.bat` and add the following line:
   ```
   set "JAVA_OPTS=%JAVA_OPTS% -Dorg.jboss.net.protocol.file.useURI=false"
   ```

   Under Unix, modify the file `jboss-eap-5.2\jboss-as\bin\run.conf` and add the parameter in the existing JAVA_OPTS.

2. Open the JBoss Administrative Console (http://<server>:<port>/admin-console/).

3. Open the web applications management interface.

4. Click *Add a new resource*.

5. Force the "Deploy Exploded".

6. Click *Continue*.

7. Wait for the installation to complete and the JCMS application to start.
8. The application is now correctly installed and configured.

With JBoss EAP 5.2, the first startup of the application will not work.
The two following operations specific to this application server must be executed:

- Move the two following libraries into the deployed JCMS webapp:
  - WEB-INF/lib/xercesImpl.jar ➔ WEB-INF/lib-disabled/xercesImpl.jar
  - WEB-INF/lib/xml-apis.jar ➔ WEB-INF/lib-disabled/xml-apis.jar

  The presence of the folder lib-disabled with these libraries is required to allow the
  operation script to execute correctly (see § 8 Update and Deployment).

- Start twice (at the initial installation and after each type modification or plugin
  installation).

Configuring the PermGenSpace and the memory allocation

It is necessary to increase the PermGenSpace (256 MB recommended) and to configure the JVM
memory allocation (1024 MB recommended). To do this refer to the JBoss documentation:


6 Starting and Configuring JCMS

6.1 Starting the Application

Once JCMS 9.0 is installed and the application server is running, start a browser and access the JCMS Status Page:

http://localhost:<port>/jcms9/admin/status.jsp

The list below shows the default ports of the different application servers:

- Apache Tomcat: 8080
- IBM WebSphere: 9080
- Oracle WebLogic: 7001
- JBoss: 8080

![Figure 2 - JCMS 9.0 Status Page](image)

Check that the site has started correctly (the JCMS Infos table should appear below the Application Server Info table).
If no error is displayed, click the Front office link to access the JCMS default home page:

![JCMS Default Home Page](image)

**Figure 3 - JCMS 9.0 Default Home Page**

This page includes an identification panel Log on as Administrator (login: admin, password: admin).

### 6.2 Initial Configuration of the Application

This section describes the main properties to be set for an initial configuration. For more details, see the JCMS User Manual on Jalios Community ([http://community.jalios.com/documentations/manuels](http://community.jalios.com/documentations/manuels)).

Once you are authenticated as Administrator, click the icon then the link to the Admin Area. In the Operation section, click the Properties link.

![Properties Editor](image)

**Figure 4 - JCMS 9.0 Properties Editor**
Site tab > Site info

- **Site Name**: enter the site's name.
- **URL**: enter the webapp's URL (e.g. [http://localhost:8080/jcms9/](http://localhost:8080/jcms9/) for a local site).

Site tab > Languages

- **Main language**: choose the site's main language. Note: JCMS does not convert content when the main language changes. It is therefore recommended to choose the main language before inputting content, then never change it.
- **Site language**: select the languages used on the site. The multi-language options are activated as soon as two languages are in use. The order of the languages will be respected in the front-office interfaces.

### 6.3 Configuring the Database

JCMS is pre-configured to use Derby, its integrated RDBMS.

To use an external RDBMS, follow this procedure:

- Download the plugin corresponding to your RDBMS from the Jalios support site:
  - [http://community.jalios.com/plugin](http://community.jalios.com/plugin)
- Install this plugin in JCMS then restart JCMS
- On the RDBMS:
  - Create a database named, for example, *jcmsdb*
    - The base must be encoded in UTF-8.
  - Create a user named, for example, *jcms*. This user must have all the rights on the *jcmsdb* base.
- On JCMS:
  - Declaration via a JDBC URL
    - In the *Database* tab select your RDBMS in the listbox.
    - Indicate the JDBC connection URL and the account/password of the user *jcms*.
    - Click *Check connection*
  - Declaration via a DataSource
    - Indicate the DataSource
    - Click *Check connection*
    - Attention: if the DataSource is used, the connection pool (generally c3p0) is systematically deactivated. The pool must be defined in the DataSource declaration.
o Restart JCMS.

o At the first start-up, JCMS creates in the MyJcmsDB base all the tables and indexes necessary for its operation.

o Verify the correct configuration in Admin Area > Monitoring > Database Info

6.4 Configuring E-mail

6.4.1 Configuring Outgoing E-mail

JCMS communicates with the e-mail messaging server in order to send e-mails to users. These mails may be sent by various JCMS functions – workflows, change notifications, publications sent by mail, alerts for administrators – and by certain modules (Newsletter, Forum, ESN, etc.).

E-mail tab > Outgoing e-mail

- **Activated:** determines whether sending of e-mail is activated.
- **SMTP Host:** address of the SMTP server.
- **Default e-mail:** the site’s default e-mail address.

6.4.2 Configuring Incoming E-mail

JCMS can also receive e-mail. This communication channel to JCMS can be used to archive e-mails or to trigger actions on receipt of an e-mail (e.g. reply to a forum discussion, change one’s ESN status, etc.) The configuration of incoming e-mail requires one or more mailboxes dedicated to JCMS.

JCMS is compatible with POP3 and IMAP servers. These must be configured to redirect all e-mails *@example.com to a dedicated mailbox. This redirection is necessary to support dynamic e-mail addresses.

Example:

- dbforum@example.com => jcms@example.com
- archive+124df6s5@example.com => jcms@example.com
- space1@example.com => jcms@example.com

JCMS analyses:

- the **from** attribute to determine the member sending the e-mail,
- the **to, cc, bcc** attributes to determine the task to be carried out.

The address prefix and suffix are used when sending an e-mail that necessitates a reply by JCMS. For better homogeneity these values should be consistent with the other addresses declared in JCMS (administrator, workspace, default address).

Incoming e-mails whose address corresponds to the regular expression of the archives will be archived in JcmsDB. It is possible to prevent any mail being archived by defining archiving rights:

- The sender must have a member account in JCMS (e.g. **from:member@example.com**).
E-mail tab > Incoming e-mail

- **Enabled**: determines whether receiving of e-mail is activated.
- **Replies prefix**: label placed in the prefix of e-mail addresses for replies.
- **Replies suffix**: suffix added to incoming e-mail addresses for replies.
- **E-mail account**: e-mail account name;
  - **Activated**: determines whether the account is active;
  - **Server**: messaging server address;
  - **Protocol**: consultation protocol of POP3 or IMAP mails;
  - **Identifier**: e-mail account login;
  - **Password**: e-mail account password;
  - **Mailbox**: the POP3 protocol uses INBOX by default. IMAP accounts can specify subfolders.
  - **Planning**: determines the consultation frequency of the e-mail account. This can be defined either as a period in minutes or by sequencing in CRON format.
  - **Author**: author of e-mails stored in JCMS.
  - **Workspace**: reception workspace of e-mails stored in JCMS.
  - **Expunge**: determines whether e-mails are removed from the e-mail server once they have been read.

6.5 Configuring LDAP

6.5.1 Principles

JCMS manages users, their groups and their rights autonomously. However, JCMS can be coupled with an LDAP (or LDAPS) directory to authenticate members and recover their information (name, e-mail, etc.). When a user tries to log in, JCMS interrogates the LDAP directory. If the authentication is confirmed, JCMS creates an account for this person at the first access then carries out the authentication. If the LDAP authentication fails, JCMS performs an authentication using its own members base.

JCMS completes the integration with the LDAP directories by synchronizing the LDAP groups and the users composing them. Since groups management is specific to each LDAP server, JCMS adapts to the directory schema. Pre-configurations are proposed for leading LDAPS server (ActiveDirectory, OpenLDAP, Sun Directory).

Since every organization has a specific LDAP schema, the LDAP configuration settings require knowledge of:
Starting and Configuring JCMS

- the DN (Distinguish Name) model of the users and groups,
- the groups model (posixGroup, groupOfUniqueNames, etc.),
- the fields containing users’ information (login, name, forename, e-mail, function, etc.).

6.5.2 Creating a Technical User in the LDAP

A technical user must be created to enable JCMS to connect to the LDAP directory. Create the following user, for example: uid=jcms,ou=People,dc=example,dc=com, with the password of your choice.

This user must have search and consultation rights in the LDAP directory. Write rights are not necessary since JCMS never modifies the LDAP directory.

6.5.3 Setting JCMS Properties

In the LDAP tab of the Admin Area > Operation > Properties Editor, set the following fields:

**Configuration**

- **Enabled**: check Yes;
- **Hostname**: name of the machine hosting the LDAP server;
- **Port**: port on which the server is listening (LDAP: 389, LDAPS: 636);
- **SSL**: check No unless you use an LDAPS server;
- **Login (DN)**: technical user ID (e.g. uid=jcms,ou=People,dc=example,dc=com);
- **Password**: the JCMS password of the technical user defined above.

Click Check connection… to check that JCMS has managed to contact the LDAP server and authenticate with the technical user.

**LDAP Schema**

Choose an LDAP schema model to pre-fill the Users and Groups fields

**Users**

- **User search DN**: base DN used to search for users in the LDAP (e.g. ou=People,dc=example,dc=com).
- **User search filter**: format of the query used to search for a user. The string \{0\} represents the value input by the user in the Login field. This format depends on the LDAP schema. On ActiveDirectory, the format is generally sAMAccountName={0}; on Sun Directory or OpenLDAP the format is generally uid={0}.
- **User objectClass**: the object class of your LDAP schema that characterizes a user (e.g. person, posixAccount, inetOrgPerson, organizationalPerson, etc.).
Starting and Configuring JCMS

- **Synchronize users**: activates the synchronization to automatically create and update members using by the LDAP information. Deactivating this synchronization prevents users who do not exist in JCMS from logging onto JCMS. The synchronization is automatically deactivated during the recovery of a Work Copy.

- **Login, Name, First name, …**: for each JCMS member field define the corresponding fields in the LDAP directory (if there is one).

**Groups**

- **Group search DN**: base DN used to search for groups in the LDAP (e.g. ou=Groups,dc=example,dc=com).

- **Group search filter**: query used to search for a group. The string \{0\} represents the DN of the LDAP entry (user or sub-group) in which the groups must be sought. This query depends on the LDAP schema. On ActiveDirectory, Oracle Directory Server or OpenLDAP, the format is generally \(&\{objectClass=group\}\{member\{0\}\}).

- **Group objectClass**: the object class of the LDAP schema that characterizes a group (e.g. group, posixGroup, groupOfNames, groupOfUniqueNames, etc.).

- **Synchronize groups**: activates the synchronization to automatically create and update the groups of a member by using the LDAP information. If this property is deactivated, new LDAP users are created in the default group. The synchronization is automatically deactivated if the synchronization of users is deactivated.

- **Name**: the LDAP field containing the group name (e.g. cn)

- **Members**: the LDAP field defining the membership of a member or a sub-group (e.g. member, memberUid, memberOf, etc.).

For more details about the implementation, see the article *Configuring JCMS with OpenLDAP (LDAP/LDAPS)* on Jalios Community (http://community.jalios.com/howto/ldap).

### 6.6 Configuring the Forward Proxy

#### 6.6.1 Principles

Some JCMS functionalities (RSS syndication, replication, etc.) perform HTTP queries that may require the use of the enterprise proxy server. If this is the case, complete the **Forward Proxy** section on the **Proxy** tab.

#### 6.6.2 Configuring JCMS Properties

- **Host**: the server address

- **Port**: the port on which the server is listening

- **Login**: the user account ID to connect to the proxy server, if necessary

- **Password**: the password to the proxy server, if necessary
- *Not for:* the addresses for which the proxy must not be used. In particular, indicate the addresses of the replicas in order that JSync synchronization requests are not directed to the proxy.

### 6.7 Configuring the Reverse Proxy

#### 6.7.1 Principles

A reverse proxy is an HTTP (HTTPS) server that receives queries and dispatches them to other HTTP servers. This type of architecture enables the real servers to be hidden from HTTP clients (they know only the reverse proxy). Reverse proxy architectures are typically used to secure a cluster of HTTP servers or to enable a homogeneous access URL to heterogeneous HTTP services, or for load balancing purposes.

JCMS must be configured explicitly to operate behind a reverse proxy: it must produce hypertext links and HTTP redirections for the reverse proxy URL instead of the URL for its own HTTP server.

It is possible to declare several reverse proxies.

#### 6.7.2 Configuring JCMS Properties

In the *Proxy* tab, enter the following fields:

- *Enabled:* check Yes;
- *Rev. Proxy IP addresses:* only queries coming from this address will be handled in reverse proxy mode
- *External Context Path:* path of the site when it is accessed from the outside world (e.g. /cms)
- *External Base URL:* the base URL of the JCMS site when it is accessed from the outside world.
- *External SSL Base URL:* the base URL of the JCMS site when it is accessed from the outside world in HTTPS.

### 6.8 Configuring the Format of Display URLs

In order to improve the referencing of JCMS sites by search engines, the URLs associated with data display (publications, categories, members, etc.) are now built to contain the title of the item to be displayed. The reason for this is that search engines increase the ranking of a page according to the words present in its URL.

In JCMS 9.0 front-office URLs take the form:

```
```

The URL prefix (/jcms/) and the format of the text part (/title-of-the-item) can be controlled by parameters. It is possible for example to add the name of a category, the date, etc. These configurations are made using the properties `descriptive-url.*` whose default values are found in the file `WEB-INF/jalios/jcms.prop` and which can be personalized in the file `WEB-INF/data/custom.prop` or in the plugin file `plugin.prop` of your main plugin.
If the URL prefix must be modified, this must be done before producing publications, since certain publications have Text Area and Rich Text Area fields which may contain URLs with this prefix.

To change the URL prefix, carry out the two following operations:

1. Add the property `descriptive-urls.prefix` in the file `WEB-INF/data/custom.prop` using the required prefix, for example:

   ```
   descriptive-urls.prefix: myprefix
   ```

2. Edit the file `WEB-INF/web.xml` and replace all the occurrences of

   ```
   <url-pattern>/jcms/*</url-pattern>
   ```

   by:

   ```
   <url-pattern>/myprefix/*</url-pattern>
   ```

### 6.9 Configuring the Connection Pool

It is recommended to activate the JDBC connection pool on JCMS production sites. The connection pool increases performance, reduces consumption of resources on the base and guarantees better robustness in the event of disconnection from the base.

Hibernate can be configured with different connection pools (C3P0, DBCP, etc.). C3P0 is a standard feature of JCMS. It can be configured by modifying the following properties in `WEB-INF/data/custom.prop`

Use precisely chosen values according to the site's expected load:

```
# C3P0 properties
hibernate.cfg.common.prop.hibernate.c3p0.max_size:
hibernate.cfg.common.prop.hibernate.c3p0.min_size:
hibernate.cfg.common.prop.hibernate.c3p0.acquire_increment:
hibernate.cfg.common.prop.hibernate.c3p0.max_statements:
hibernate.cfg.common.prop.hibernate.c3p0.idle_test_period:
hibernate.cfg.common.prop.hibernate.c3p0.timeout:
```

In addition to these properties, it is possible to define others for a more advanced configuration. However, they must be declared in the file `WEB-INF/classes/c3p0.properties`.

**Note:** the properties defined in `custom.prop` overwrite those defined in `c3p0.properties` and their name is different.

It is recommended to set the connection timeout in milliseconds (by default there is no timeout) via the property `c3p0.checkoutTimeout`. For example, to limit the connection delay to 5 seconds:

```
c3p0.checkoutTimeout: 5000
```

For more details about C3P0 configuration, consult the documentation:

6.10 Configuring Replication by JSync

JCMS incorporates the JSync data replication protocol that assures data synchronization between several JCMS instances. JSync makes it possible to use high-availability architectures (load balancing and redundancy) and decentralized architectures (several JCMS instances spread over an extranet). These different architectures are described in the JCMS Technical White Paper (http://community.jalios.com/documentations/livresblancs).

JSync is an optimistic replication protocol of epidemic propagation type; it enables very flexible topologies. In a replicated architecture every instance of JCMS is called a replica. The replicas are organized in a group with a leader. Each replica modifies its data autonomously then notifies these changes to its leader which in turn propagates them to the other replicas of the group in order to assure global data consistency. A replica can disconnect itself, diverge then reconnect later to propagate it’s the latest changes. JCMS supports two ways of propagating changes:

- automatic propagation (distribution as soon as the new data are stabilized);
- manual propagation (distribution at the Administrator’s request).

Data conflicts can appear after divergence periods. For example, a given data item may have been modified on one replica but deleted on another. During propagation of new data, JSync automatically resolves this type of conflict. Deletions have priority; concurrent updates are reordered homogeneously. Inter-object conflicts are not handled (for example, deletion of a category on a replica and addition of a content item referencing this category on another replica). In minimize conflicts it is recommended to keep the divergence periods short. Ideally, contributors working on linked data should be grouped on the same replica or on replicas configured for immediate propagation. Non-contributing readers, on the other hand, can be spread over all the replicas of the group.

The use of JSync is incompatible with Derby, the JCMS 9.0 internal database. If you wish to use JSync you must therefore implement an external RDBMS. The leader and replicas must connect to the same database. To assure high availability it is recommended to employ a clustered database.

![Figure 5 – Example of a high-availability configuration for JCMS 9.0](image-url)

For more details about the implementation of replication in the case of load balancing, see the article Implementation of a high-availability JCMS site with the JSync replication protocol on the Jalios support site (http://community.jalios.com/howto/jsync). A second article is also available for the Oracle WebLogic environment: Implementation of a high-availability JCMS site with Oracle WebLogic (http://community.jalios.com/howto/jsyncwls).
6.11 Installing an Add-Pack

The standard JCMS 9.0 delivery configuration provides for management of 1 workspace, 10 collaborative spaces\(^1\), 100 members and 2 languages. If you need more, you must acquire and install an add-pack.

An add-pack takes the form of an encrypted key that must be declared via the property channel.add-pack. This parameter must be added to the file WEB-INF/data/custom.prop.

To verify that the add-pack is taken into account, consult Admin Area > Site Info. The add-pack name appears in Server Info/JCMS Version.

6.12 Configuring the Frontal Web Server

JCMS 9.0 does not require a specific frontal web server. It can use the one provided by the Java EE application server. However, a frontal such as Apache HTTPD or IHS has some advantages:

- optimized performance:
  - server side: static files are served directly by the web frontal which significantly lightens the load on the Java EE server;
  - client side: HTTP cache headers are activated on the static resources in order to avoid clients reloading resources for every request,
- high availability by implementing load balancing and/or failover mechanisms.

The article "Configure Apache, mod_jk and Tomcat for JCMS" describes the implementation of an Apache HTTPD frontal for Tomcat and JCMS:
http://community.jalios.com/howto/apache

The article "Implementation of a high availability JCMS site with the JSync replication protocol" describes the installation of several JCMS application synchronized using JSync behind an Apache HTTPD frontal:
http://community.jalios.com/howto/jsync

JCMS 9.0 is certified for the Apache HTTPD 2.2 web server using the mod_jk 1.2 connector.

Make sure you respect the following rules:

- The dynamic resources generated by JCMS (jsp and servlet) and the resources protected by access rights in JCMS (upload folder/) must imperatively be handled by the Java EE server. Refer to the above article for an exhaustive list of these dynamic resources.
  - KeepAlive must be activated on the frontal server.

6.13 Installing Jalios Office Launcher

If you wish to be able to trigger WebDAV editing using a browser other than MS Internet Explorer, you must install the Jalios Office Launcher component on the user workstations concerned.

On JCMS, edit the file WEB-INF/data/custom.prop and set the parameter channel.webdav.custom-protocol.enabled to true, then restart JCMS.

\(^1\) Requires the Collaborative Spaces plugin.
Starting and Configuring JCMS

Note: once this property is set, all contributors must use Jalios Office Launcher.

You can download Jalios Office Launcher from Jalios Community (http://community.jalios.com/tools/jaliosofficelauncher)

On the user workstation, quit all Microsoft Office applications (Word, PowerPoint, Excel, etc.). Decompact the Jalios Office Launcher archive file, then run the setup.exe program.

On Windows Vista machines, if you access your JCMS site in HTTP (rather than HTTPS), you must launch the file vista-webdav.reg also contained in the archive file, then restart the workstation.

6.14 Adding plugins

JCMS incorporates an extension mechanism based on modules known as plugins. JCMS plugins add new functionalities (e.g. new portlets, blog, podcast, SSO logon, etc.). Such plugins may be proposed by Jalios or by other providers (publishers, integrators, users, etc.). Each plugin has its own usage license.

JCMS plugins are available on Jalios Community (http://community.jalios.com/plugin/)

A plugin takes the form of a ZIP file. A Plugin Manager is provided to handle the installation, configuration and removal of plugins (Admin Area > Operation > Plugin Manager).

To add a plugin, in the Upload a Plugin box select the plugin ZIP file using the Browse… button, check the Automatic deploy after upload check-box then click the Upload a Plugin button. JCMS must be restarted once the plugin is installed.

To configure a plugin, click on its name in the plugins list and select the Administration tab.

To delete a plugin, click on its name in the plugins list, then click Delete the plugin.
### 7 Operating Procedures

#### 7.1 Starting and Stopping JCMS

Starting and stopping JCMS consists simply in starting or stopping the webapp. When JSync is used, it is recommended to stop the leader's webapp last.

##### 7.1.1 Apache Tomcat

Tomcat does not provide for starting or stopping a particular webapp; you must start or stop the entire server.

##### 7.1.2 IBM WebSphere Application Server

The Administrative Console of the WebSphere Application Server is used to start and stop JCMS:

**Starting**

Open the WebSphere Application Server console: Programs > IBM WebSphere > Application Server v6 > Profiles > Default > Administrative Console.

Go to Applications > Enterprise applications. Check the box next to jcms9, then click Start. The first start-up may be rather long since the files have to be compiled.

**Stopping**

Open the Administrative Console of the WebSphere Application Server: Programs > IBM WebSphere > Application server v6 > Profiles > Default > Administrative Console.

Go to Applications > Enterprise applications. Check the box next to jcms9, then click Stop.

##### 7.1.3 Oracle WebLogic Server

The Oracle WebLogic Server Administration Console is used to start and stop JCMS.

**Starting**

Select jcms9-ear in the applications list. Open the Start drop-down menu and choose Servicing all requests.

**Stopping**

Select jcms9-ear in the applications list, then click Stop.
7.2 Monitoring

JCMS 9.0 provides an interface to monitor the application. In addition to the Memory Monitor, this presents, for a given period, the site restarts, site accesses (queries and sessions) and the number of objects in the store. All this information is now persistent and is no longer lost during restarts. The persistence window is parameter-defined via the property monitoring.history-day.

![Monitoring Interface](image)

Figure 6 - Monitoring

Note: the indicated memory consumptions are based on information supplied by the JDK APIs and are generally overestimated. These figures represent the totality of the memory presently consumed (part of this memory will be released later). Moreover, if several webapps run on the same JVM, the consumption displayed reflects the cumulated consumption of all these webapps.

In view of the memory management scheme in the JVM, based on a garbage collector, it is quite normal that the consumption regularly rises (up to 90%) and falls. If on the other hand, the consumption never falls below 80%, the memory allocated to the JVM should be increased.
7.3 **Data Integrity Check**

It is possible at any time to perform a data integrity check. This control lists any data that do not respect the constraints assigned to them (e.g. mandatory fields, duplicates, specific controls, etc.).

To launch the integrity check: **Admin Area > Monitoring > Data Integrity Check.** A report is generated during the first access. Since the calculation requires a lot of processor resources, this report is cached (in memory). To update it just click the **Refresh** button.

![Data Integrity Check](image)

**Figure 7 - Data Integrity Check**
7.4 Event Logs

JCMS 9.0 uses the library log4j to log events. The logs are saved in the folder <WEBAPP_DIR>/WEB-INF/data/logs/.

The last 50 events can be consulted in the Admin Area (Monitoring > Event logs).

For more details about event logging configuration, see the practical sheet on Jalios Community: *JCMS 5: Logging of events with log4j* (http://community.jalios.com/howto/log4j).

![Event log](image-url)
7.5 Stopping Writes

For some maintenance operations it may be necessary to prevent users from modifying their content momentarily. To do this, click the *Disable data write...* button in the Admin Area. You are invited to enter a message explaining the stoppage. During the writing stoppage:

- All writes in *JStore* and *JcmsDB* are stopped.
- Writes associated with *JSync* replication remain active (you can use the Properties Editor to deactivate them).
- A message indicating the stoppage of writes scrolls in the Admin Area.
- Workspaces become inaccessible; the maintenance message is displayed instead.
- The Admin Area is active but it is not possible to update members, groups and workflows.
- All contribution interfaces in the front-office disappear.
- The publication Track Reader is inactive.

To enable writing again, click the *Enable data write...* button.
7.6 **Index Manager**

When JCMS is running no reindexing is necessary. The indexes are fed automatically during creations, updates, deletions and synchronizations of data. Similarly, the Search Spell Suggest module updates itself regularly. However, reindexing is necessary after a deployment, a migration or an external modification of the store.

To reindex the publications: **Admin Area > Operation > Index Manager** and click the **Reindex All Publication...** button.

Note: index optimization and indexing are tasks that consume a lot of processor resources and execution time. However, the site and search functions are still active throughout these operations.

![Index Manager](image)

Figure 9 - Index Manager

7.7 **Cache Management**

JCMS 9.0 portal pages can contain caches defined for each portlet. The Cache Manager (**Admin Area > Operation > Cache Manager**) can be used to consult and empty them.

7.8 **Store Cleaner**

Structure data (category, groups, member, portlet, etc.) and content is stored in *JStore*. With each write operation (creation, update, deletion), *JStore* writes a line representing this operation in the file *store.xml*, so this file grows progressively over time. This file is read only in three cases:

1. when JCMS is started,
Operating Procedures

2. during access to the version history of a publication (partial reading and caching),

3. during synchronization by JSync (partial reading).

If the file becomes big, for example more than 100 MB, and there are a large number of updates and deletions (Admin Area > Monitoring > Store Info), it may be necessary to clean the store.

**Note: the Store Cleaner must not be used on a site in production.** Further, if JSync replication is activated, you must execute the following procedure first:

1. Stop the writes on the production webapp(s) and synchronize the replicas, if any.

2. Make a complete copy of the site (using DeployManager).

3. Click **Store Cleaner** on the Store Info page.

4. Redeploy the cleaned store on all the production webapps.

To start the cleaning, you must define the cleaning scope, in other words the portion of **store.xml** that will be cleaned, and the cleaning rules to apply. Once these choices are made, click the **Clean and restart...** button. The current file **store.xml** is copied (to the folder WEB-INF/data) and replaced by the cleaned store, then the site restarts. Verify in the traces that the start-up with this new store has functioned correctly.

---

![Figure 10 - Store Cleaner](image-url)
7.9 Backup/Restoration Procedures

7.9.1 Automatic Store Backup

JCMS incorporates a system to back up JStore regularly. The objective is simply to provide a minimal store backup to cover extreme incidents such as a defective external backup. This backup system is by default activated, but it can be deactivated (Properties Editor > Advanced tab).

Note: this system does not replace a system for regular, total backup of the JCMS site: store, types, workflows, documents, properties, etc.

The automatic backup involves three parameters:

1. Backup planning (store.backup.schedule): every day at 03:00h by default.
2. Number of backups to preserve (store.backup.max): 10 backups by default.

7.9.2 Automatic Backup of JcmsDB (Derby)

JCMS incorporates a system to perform regular backups of the JcmsDB database managed by Derby. Note: JCMS does not handle backups if you use an external RDBMS.

Backup of the JcmsDB database is activated by default, but it can be deactivated (via the property derby.backup.enabled). The entire folder WEB-INF/derby/jcmsdb is saved in the form of a zip file. To restore one of the archives, just decompress it and place it in this same folder.

The JcmsDB backup involves three parameters:

1. Backup planning (derby.backup.schedule): every day at 03:00h by default;
2. Number of backups to preserve (derby.backup.max): 10 backups by default;

7.9.3 JCMS Webapp Backup Procedure

Before carrying out a backup, it is recommended to stop JCMS. If this is not possible, it is important to check that the content of the copy of the folder <WEBAPP_DIR>/WEB-INF/data/ is identical to the original.

In an environment replicated by JSync it is recommended to save each replica.

7.9.4 JCMS Webapp Restoration Procedure

The restoration procedure is the inverse of the backup procedure. However, there are two possible approaches.

In the case of a simple restoration, stop the JCMS application before restoring by overwriting the files.

In the case of a complete restoration, stop the application server, delete the current version of the folder containing the JCMS application then replace it with the backup.
Although simple restoration is sufficient in most cases (since all the old files are restored), any new files are not replaced, which can sometimes cause undesirable effects. A complete restoration is the only guaranteed way of re-establishing a coherent, exhaustive image.

In both cases, cache effects can occur. It may be useful to clear the JSP compilations folder:

- **Apache Tomcat:** &#62;TOMCAT_DIR;&#62;/work/Standalone/localhost/jcms/
- **IBM WebSphere:** &lt;WAS_DIR&gt;/temp/&lt;machine&gt;/&lt;server&gt;/jcms
- **Oracle WebLogic:** &lt;WLS_DIR&gt;/user_projects/domains/{domain-name}/servers/AdminServer/tmp/_WL_user/{application-name}
8 Update and Deployment

The execution of the deployment script is explained in detail on Jalios Community (http://community.jalios.com/howto/deploy)

8.1 General Procedure

- Developers work on their development webapp (developing a new application, migrating to a new version, etc.).
- A maintainer recovers a version and generates a new complete war.
- A maintainer deploys this war on the production system in a command line using the script deploy.sh.

The script deploys the application modifications (static files, jsp, plugin, classes, library, etc.) and structural modifications (workflow, types). All the production environment data are conserved (upload, properties, base).

During the deployment you must decide how to treat the store.xml file. There are three possibilities:

1. The store.xml of the new war is ignored (this is the default choice): the production store.xml is conserved as-is.
2. The store.xml of the war and of the production are merged: the StoreMerge application is invoked during the deployment and the two stores are merged.
3. The store.xml of the war replaces that of the production webapp.

8.2 Developing and Preparing the war File

1. Recover a complete copy of the production system.
   - This must imperatively contain the plugins.
   - The presence of the upload, archives, log folders or other data is not important for the later deployment; but recover them if you need them for your developments.
2. Write and validate your developments.
3. Package the new version of the webapp in a war, respecting the following constraints:
   - All plugins must imperatively be present.
   - The presence of the store.xml file is necessary only if modifications need to be reincorporated in production.
   - It is pointless to add the upload and archives folders in this new war: they will be ignored.
   - It is pointless to put the following folders or files (although leaving them in the war will have no effect on the deployment):
     - Derby base (WEB-INF/derby)
     - Lucene index (WEB-INF/lucene)
     - Statistics, logs, monitoring, backup, etc. (WEB-INF/stats/, WEB-INF/logs/, WEB-INF/jsynclog/, WEB-INF/backups/, WEB-INF/monitoring.xml)
8.3 Deploying the war on the Server

8.3.1 Initial Installation

This configuration must be carried out only once at the start of the site's life:

- Displace the three following folders outside the webapp:
  - archives/
  - upload/
  - WEB-INF/data/

- Create **absolute** symbolic links (for example, /opt/jcms/data/) to reference each of these folders in the webapp.

All these constraints are verified by the deployment script (but it will make no modification if they are not respected).

With Tomcat, on **linux environment only**, the `allowLinking` attribute must be configured in the webapp context:

```xml
<Context ... allowLinking="true" ... />
```

or in the default context (conf/context.xml)

**Important**: For security reason, do not set this attribute on Windows environment.

8.3.2 Deploying a New Version

**Basic usage:**

This usage is suitable in the majority of cases:

1. Place the new war on the server (ftp, ssh,...),
2. Stop the Java EE server,
3. Invoke the deployment script

   ```bash
   ./deploy.sh new.war webapp-folder/
   ```

4. Restart the Java EE server.

**Advanced usage:**

- **Store management:**
  - use the `-s "merge"` option to merge the production store and the war store (a backup is made before modification),
  - use the `-s "overwrite"` option to replace the production store by the war store (a backup is made before modification).

- Option `-b` is used to save the webapp in another folder: at the end of the process, the original webapp is saved in the same folder in which it was found, simply suffixed with the deployment date. To avoid this backup interfering with the operation of the Java EE server (for example with Tomcat in automatic deployment mode in the webapp folder), it is possible to specify another folder into which the original webapp will be moved with the option `-b`.

- **Dry-run mode** (-d option): in this mode no modification is made; only the prerequisites are checked and the operations that would have been carried out are shown.
• Verbose mode (-v option): use this mode to obtain detailed logs on the execution of the deploy.sh script (this is useful for support, debugging or understanding the script's operation).

8.4 Remarks

• rollback: in the event of deployment failure (disk full, unsuccessful store merge, or other error) a rollback is performed and the webapp returns to its initial state.
• save: the existing version is saved if the store.xml merged or replaced.
• structure: the production types and workflows are saved before being replaced by those of the war.
• upload: the upload folder is not recovered from the war, so new documents cannot be deployed using this mechanism.
• properties: the WEB-INF/data folder is NOT recovered from the new war (except store.xml, types and workflow), so the properties files that would be modified in the war (custom.prop) are not propagated. The deployment script can be modified or completed with another script to meet different needs.
9 Equipment and Configuration

9.1 Configuring the Server

9.1.1 Server and Operating System

JCMS 9.0 must be installed on a server with one of the following configurations:

- INTEL-compatible server running under:
  - or Linux RedHat AS5 or equivalent
- Sun Oracle server with Solaris 9 or later
- AIX 6.1 Server

The server must have at least 2 GB of RAM and 10 GB of disk space. To handle particularly large volumes of data more memory may be necessary. Systems handling more than 100,000 objects will necessarily require more RAM and possibly a 64-bit server and operating system.

9.1.2 Java EE Application Server

JCMS 9.0 is a Java EE5 compliant web application. It is independent of the operating system but requires a Java EE5 compatible application server, in particular for the servlet 2.4 API.

JCMS 9.0 requires a JVM 1.6:

- Oracle Java SE 6Update 30 (or later) for Apache Tomcat
- Oracle or IBM JVM 1.6 certified for JBoss
- Oracle or IBM JVM 1.6 certified for WebSphere.
- Oracle JVM 1.6 certified for Oracle WebLogic

The use of traditional Chinese requires a Java 1.7 environment.

The table below summarizes the operating systems and application servers certified for JCMS 9.0:

<table>
<thead>
<tr>
<th>Application server</th>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Windows</td>
</tr>
<tr>
<td>Apache Tomcat 7.0</td>
<td>●</td>
</tr>
<tr>
<td>IBM WebSphere 8.5.5</td>
<td>●</td>
</tr>
<tr>
<td>Oracle WebLogic 12c</td>
<td>●</td>
</tr>
<tr>
<td>JBoss EAP 5.2</td>
<td>●</td>
</tr>
</tbody>
</table>

To guarantee the security of your platform and be sure you always have latest corrective updates, we recommend that you systematically use the latest maintenance version of the application server (conserving the same major version indicated).
9.1.3 **RDBMS**

JCMS 9.0 is delivered with an embedded RDBMS (Derby). JCMS is not certified to run in production with Derby. For production environments it is recommended to replace Derby by an external RDBMS. JCMS 9.0 is certified for several external database systems:

- PostgreSQL 8.4, 9.0 & 9.3
- MySQL 5.1 & 5.5 (InnoDB storage engine)
- Oracle 10g & 11g
- Microsoft SQL Server 2008 & 2012
- IBM DB2 9.7

9.1.4 **Web Server**

JCMS 9.0 is compatible with the following web servers:

- Apache HTTP 2.2 and mod_jk as Tomcat connector,
- IBM HTTP Server (IHS) in WebSphere environments.

9.1.5 **SMTP Host**

JCMS 9.0 requires a SMTP host in order to send e-mails (notifications, alerts, bulletins, message to a friend, etc.).

9.1.6 **POP3/IMAP Server (optional)**

If incoming e-mail is activated, JCMS 9.0 requires a messaging account on a POP3 or IMAP server in order to manage incoming e-mail.

9.1.7 **LDAP Directory (optional)**

JCMS 9.0 can manage authentications autonomously or it can be interfaced with an LDAP/LDAPS server. JCMS 9.0 is certified for the following directories:

- Microsoft Active Directory
- Novell eDirectory
- OpenLDAP
- Oracle Directory Server

9.1.8 **Recommended Configurations**

Jalios recommends the following platform for operational simplicity and high performance:

- Intel Xeon server (3 GHz) or Intel Core 2 Duo equivalent, with 4 GB RAM, 320 GB disk
- Linux 64-bit
- JDK 7
- Tomcat 7.0
- Apache 2.2 + mod_jk 1.2
- PostgreSQL 9.3

To guarantee the security of your platform, we recommend that you install the latest maintenance versions of the above software.
9.2  Configuring the Workstations

9.2.1  Contributor and Administrator Workstations

Contributors and administrators must have a machine with at least 512 MB of RAM and a screen with 1280 x 800 minimum resolution. They interact with JCMS 9.0 using one of the following browsers:

- Microsoft Internet Explorer 8 or later, without any compatibility mode
- Mozilla Firefox 18 or later
- Google Chrome 24 or later
- Safari 6 or later

Adobe Flash Player 9 (or later version) is required notably to display workflow graphics.

9.2.2  User Workstations

Sites built using JCMS 9.0 can be consulted using the following browsers, provided the site's "house style" is compatible:

- Microsoft Internet Explorer 8 or later, without any compatibility mode
- Mozilla Firefox 18 or later
- Google Chrome 24 or later
- Safari 6 or later
- Opera

To be able to read video streams, a browser supporting HTML 5 or the Adobe Flash Player (v9 minimum) is required. Some JCMS plugins may also require the Flash reader or a specific browser version.

9.2.3  Developer Workstations

Jalios recommends the IDE Eclipse Indigo (or later version) for development work with JCMS 9.0. Developers must have a machine with at least 2 GB of RAM and a screen with 1280 x 800 minimum resolution. They should use one of the browser approved for administrators (see above).