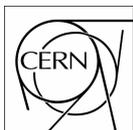




User Guide

Version 2.2

Document Version: 2.2.1
Authors: M. Gulmini



Revision History

Date	Version	Description	Author
March 31, 2005	1.0	Document creation	M. Gulmini
April 20, 2005	1.0	Document finalization	M. Gulmini
July 27, 2005	1.1	Update to RCMS2.1	M. Gulmini
August 2, 2005	1.1	Installation script and Oracle DB	A. Petrucci
August 30, 2005	2.1.3	Document update	M. Gulmini
February 15, 2006	2.2.1	Document update to RCMS 2.2	M. Gulmini
March 11, 2006	2.2.1	Document finalization	M. Gulmini

CI Record

Field	Description
CI Identifier	
Description	
Submission Date	
Submitted By	
Components	
Dependencies/Related	
External Identifier	
Point of Contact	Michele.Gulmini@cern.ch
Comments	
Physical Location	

Table of Contents

1	Release Overview	5
2	RCMS2.2 Installation	6
2.1	Installation prerequisites	6
2.1.1	Java installation	6
2.1.2	MySQL installation	7
2.1.3	Jakarta Tomcat installation	7
2.2	Checkout the code from CVS	8
2.3	Running the installation script	9
2.4	Installation reference guide	10
2.4.1	Preparing the databases needed for running RCMS 2 software	10
2.4.2	Configuring the Tomcat installation	12
2.4.3	Configuring RCMS	12
2.4.4	Deploying RCMS	14
3	Run the Hello Example	15
3.1	Compile and deploy the HelloFunctionManager example	15
3.2	Fill the Resource Service	16
3.3	It works	17
4	Appendix A: Configuring and using Eclipse 3.0	18
	Configuring Eclipse IDE	18
	Tomcat plugin configuration	18
	Lomboz Plugin configuration	19
	Properties page	23
	Download RCMS2 Eclipse projects from CVS	23
	Creating Projects for the State Machines	24

RCMS 2.2 User Guide

1 Release Overview

The RCMS2.2 software is composed of the following main packages:

- RCMS2 framework
 - Function Manager Framework
 - Resource Service
 - Generic Control Graphical User Interface (GUI)
- Function Manager Example implementations
- RCMS2 tools
 - DuckCAD
 - Log Collector

The Function Manager Framework provides a set of facilities to control DAQ applications.

The **Function Managers** (FM) are objects which implement finite state machines. The states and transitions are customizable to the needs of the controlled resources. A partition (group of controlled resources) can contain other partitions, through which a hierarchical organization of Function Managers is possible. The communication of states and commands between function managers and resources and amongst function managers is achieved through SOAP messages with a defined XML schema.

The **Resource Service** (RS2) stores control structure configurations.

The Resource Service Database for XDAQ applications (RSX) stores XDAQ configurations. Together, the RS2 and RSX, provide the means to store the configurations of a full DAQ system.

In this release the RS2 and RSX are implemented as a single database.

The generic **GUI** allows to control a given configuration according to the state machine defined in the Function Manager.

The **DuckCAD** is a graphical utility to generate DAQ and control configurations and store them in the RS2 and RSX.

The **Log Collector** is a service that collects logs from RCMS and XDAQ applications. Log messages can be visualized, written to database, or dispatched to clients through a JMS (Java Message Service) system.

This User Guide provides information on how to install and deploy the RCMS 2.2 framework. An **installation script** can be used to facilitate the job. The script also installs the LogCollector.

Documentation for RCMS2 tools, DuckCAD and Log Collector, can be found in separate documents.

2 *RCMS2.2 Installation*

This section explains how to install the RCMS 2 Framework software.

Provided some installation prerequisites (paragraph 2.1), an installation script can be used to perform most of the installation tasks (paragraph 2.3). The installation script allows also to install the Log Collector.

2.1 *Installation prerequisites*

The following external packages are needed in order to run RCMS 2 software:

- Java 1.4.2
- MySQL DBMS
- Jakarta Tomcat 5.0.x

If you have already installed these packages, then skip this section and go to section 2.2.

Support for Oracle 10g is also provided. Skip the MySQL section if you plan to use Oracle instead.

2.1.1 *Java installation*

The RCMS software requires the Java 2 Platform Standard Edition 1.4.2.

If you already have it on your system, then skip this paragraph.

You can download Java J2SE 1.4.2 from the Java Web Site (<http://java.sun.com>), and follow the installation procedure for your system.

For a Linux system you could for instance download an RPM file (j2sdk-1_4_2_04-linux-i586-rpm.bin) into a directory of your choice and install it.

```
> chmod +x j2sdk-1_4_2_04-linux-i586-rpm.bin
> ./j2sdk-1_4_2_04-linux-i586-rpm.bin
> rpm -i j2sdk-1_4_2_04-linux-i586.rpm
```

The new installation can be found in the directory /usr/java/j2sdk1.4.2_04.

In order to make sure that you always pick the correct executables of the java environment you should set your PATH to the directory where the java binaries can be found.

If you use the tcsh shell, you can add the following lines to your .tcshrc file:

```
> setenv JAVA_HOME /usr/java/j2sdk1.4.2_04
> setenv PATH ${JAVA_HOME}/bin:$PATH
```

1
2 For a bash shell:

3
4 > export JAVA_HOME=/usr/java/j2sdk1.4.2_04
5 > export PATH= /usr/java/j2sdk1.4.2_04/bin:\$PATH
6

7 If you have problems to install java, see the documentation in
8 <http://java.sun.com/j2se/1.4.2/docs/index.html>
9
10

11 2.1.2 MySQL installation

12
13 The RCMS 2 software has been tested with versions 3 and 4 of MySQL.
14 Development was done using a MySQL 4.1.10 running on Linux SLC3.
15 Version 3.23 of Linux SLC3 has also been used.
16 The following instructions explain how to install MySQL 4.1.10.
17

- 18 • Download the MySQL database server 4.1.10 from
19 <http://dev.mysql.com/downloads/index.html>, choose MySQL 4.1, select the version
20 you need. For example mysql-standard-4.1.10a-pc-linux-gnu-i686.tar.gz
- 21 • copy the file mysql-standard-4.1.10a-pc-linux-gnu-i686.tar.gz into your installation
22 directory. For example in usr/local/
- 23 • execute this group of commands:

```
24 shell> groupadd mysql  
25 shell> useradd -g mysql mysql  
26 shell> cd /usr/local  
27 shell> gunzip < /PATH/TO/MYSQL-VERSION-OS.tar.gz | tar xvf -  
28 shell> ln -s FULL-PATH-TO-MYSQL-VERSION-OS mysql  
29 shell> cd mysql  
30 shell> scripts/mysql_install_db --user=mysql  
31 shell> chown -R root .  
32 shell> chown -R mysql data  
33 shell> chgrp -R mysql .  
34 shell> bin/mysqld_safe --user=mysql &
```

35
36 For any problems, refer to the documentation in
37 <http://dev.mysql.com/doc/mysql/en/index.html>

38 See also the README and INSTALL-BINARY file in the directory FULL-PATH-TO-
39 MYSQL-VERSION-OS (for example: mysql-standard-4.1.10a-pc-linux-gnu-i686).
40
41

42 2.1.3 Jakarta Tomcat installation

43
44 The RCMS 2 software needs an Apache Jakarta tomcat 5.0.x installation.
45 The software has been developed and tested using version 5.0.28 running on a Linux SLC 3
46 machine.
47 Apache Jakarta tomcat is available on the RCMS 2 web site.
48
49
50



Jakarta tomcat 5.0.28 for Linux is available on the RCMS2 web site (<http://cmsdoc.cern.ch/TriDAS/RCMS/Downloads/data/rcms2/externalpackages/software/jakarta-tomcat-5.0.28.tar.gz>)

If you want install and try another Tomcat 5.0.x version follow the instructions below.

- Download the Tomcat version 5.0.x from the jakarta site. <http://jakarta.apache.org/site/downloads>, choose Tomcat, Tomcat 5, 5.0.x Binary for your operating system.
- Linux: jakarta-tomcat-5.0.28.tar.gz
 - Gunzip the file in your installation directory, usually in /usr/local using `tar -xzvf jakarta-tomcat-5.0.x.tar.gz`
 - This creates a directory containing the Tomcat application.
 - Set CATALINA_HOME:
`export CATALINA_HOME=/usr/local/jakarta-tomcat-5.0.28`
for bash or korn shell (bash, ksh)
`setenv CATALINA_HOME /usr/local/jakarta-tomcat-5.0.28`
for c shell (tcsh, csh)
- Windows: jakarta-tomcat-5.0.28.zip
 - Gunzip the file in your installation directory.
 - This creates a directory containing the Tomcat application.
 - Set the System Variable, CATALINA_HOME, in the Control Panel.
- To start and stop tomcat use `startup.sh` and `shutdown.sh` for linux, and `startup.bat` and `shutdown.bat` for windows in `$CATALINA_HOME/bin`.

2.2 Checkout the code from CVS

The RCMS source code is available on the CERN CMS CVS server. This paragraph explains how to checkout the software.

It is assumed that the user shell is **tcsh**, and that the installation directory is **CheckoutDir** (please replace **CheckoutDir** with the full path to your installation directory).

Set the environment variable **CVSROOT** to point to the CMS CVS server.

If you are at CERN:

```
> setenv CVSROOT :kserver:cmscvcs.cern.ch:/cvs_server/repositories/TriDAS
```

If you are outside CERN you can use the anonymous login:

```
> setenv CVSROOT :pserver:anonymous@cmscvcs.cern.ch:/cvs_server/repositories/TriDAS
> cvs login
```

using the password **98passwd**.

More information can be found at http://cmsdoc.cern.ch/cmsoo/projects/cvs_server.html.



1
2 Go to the installation directory.

3
4 > cd **CheckoutDir**

5
6 Now you can checkout release 2 of the RCMS source code.

7
8 The module name is `TriDAS/RunControl`.

9
10 Verify the last stable tag on the RCMS2 web site (<http://cmsdoc.cern.ch/TriDAS/RCMS>),
11 and checkout it.

12
13 > cvs co -P -r <cvs_tag> TriDAS/RunControl

14
15 At the moment of writing the tag is **RCMS_2_2_0_rc1**.

16
17 Assuming the RCMS installation directory is **RCMSInstallationDir**.

18
19 The **RCMSInstallationDir/framework** directory contains the RCMS 2 framework.

20
21 The **RCMSInstallationDir/functionmanagers/examples** directory contains Function
22 Managers developed as examples by the DAQ group.

23 You can take these Function Managers as examples for developing your own Function
24 Manager.

25
26 The **RCMSInstallationDir/tools** directory contains RCMS 2 utilities.

27 The DuckCAD utility allows for generating XDAQ and RunControl configurations, and the
28 Log Collector tool for collecting, storing and visualizing both RCMS and XDAQ log
29 messages.

30
31 If you prefer you can also checkout separately the three directories separately.

32
33 > cvs co -P -r <cvs_tag> TriDAS/RunControl/framework

34 > cvs co -P -r <cvs_tag> TriDAS/RunControl/functionmanagers

35 > cvs co -P -r <cvs_tag> TriDAS/RunControl/tools

36 37 38 *2.3 Running the installation script*

39
40 The RCMS 2.2 Release provides an installation script that performs the tasks explained in the
41 following paragraphs.

42
43 The script assumes that you are using a MySQL database.

44 For Oracle 10g please follow the steps in paragraph 2.4.

45
46 The script installs the RCMS framework, the RCMS Log Collector tool, it configures the
47 needed MySQL databases, and performs a few modifications, if needed, in the tomcat
48 configuration .

1
2
3 The steps necessary for deploying and running the Hello Example explained in paragraph 4
4 are also performed.

5
6 If you want to use the installation script you must set three environment variables:

- 7 1. RCMS_HOME
- 8 2. JAVA_HOME
- 9 3. CATALINA_HOME

10
11 For instance:

```
12  
13 > setenv RCMS_HOME RCMSInstallationDir  
14 > setenv JAVA_HOME /usr/java/j2sdk1.4.2_04  
15 > setenv CATALINA_HOME /users/rcms/jakarta-tomcat-5.0.28  
16
```

17 Go in the directory **RCMSInstallationDir/framework/installation/** and type :

```
18  
19 > perl ./installRCMS.pl  
20
```

21 and follow the instructions.

22
23 If you use that script keep the next paragraphs as a reference guide to be used in case of
24 problems.

25 The script also installs and deploys into your tomcat the Hello example (section 3)

26 27 28 *2.4 Installation reference guide*

29
30 If you use a MySQL database most of the tasks described here are automatically performed
31 by the installation script (paragraph 2.3).

32 For Oracle 10g please read the following instructions.

33 34 *2.4.1 Preparing the databases needed for running RCMS 2 software*

35
36 This RCMS 2 release needs an installation of a mySQL DBMS (see paragraph 2.1).

37 Oracle 10g DBMS is also supported.

38
39 Two databases are needed:

- 40 - Resource Service 2 and RSX database
- 41 - Realm database (tomcat users and roles)

42
43 In the directory

44 **RCMSInstallationDir/framework/installation/configuration/databases/mysql**
45 you can find two SQL scripts for the creation of the two databases.

46 A directory for Oracle exists as well:

47 **RCMSInstallationDir/framework/installation/configuration/databases/oracle**
48
49

1
2
3 The **Realm** script creates a database named “**realm**”. This database stores RCMS users,
4 passwords, and permissions. It is used by tomcat to check if a user can access the RCMS
5 services. By default the script creates a user **rcms** with password **rcms2**.
6 If you want to change these default values modify the mysqlRealm.sql or the oracleRealm.sql
7 according to your needs.

8
9 The script can be run from the mySQL prompt with the command:

```
10  
11 mysql> source mysqlRealm.sql;
```

12
13 Or from the sqlplus prompt with the command:

```
14  
15 SQL> @oracleRealm.sql;
```

16
17 The **RS2Config** script initializes the Resource Service 2 database tables.
18 The RS2 database will be used to store the RCMS configurations for your DAQ and RCMS
19 system.

20
21 From the mySQL prompt create a new database (for instance **rs2**) and run the script:

```
22  
23 mysql> create database rs2;  
24 mysql> use rs2;  
25 mysql> source mysqlRS2Config.sql;
```

26
27 From the sqlplus prompt:

```
28  
29 SQL> @oracleRealm.sql;
```

30
31 Now for MySql we have to grant the privileges to the new databases.

32 For the tomcat **realm** database:

```
33  
34 > GRANT ALL PRIVILEGES ON realm.* TO realmuser@% identified by  
35 realmpassword;  
36 > GRANT ALL PRIVILEGES ON realm.* TO realmuser@localhost identified by  
37 realmpassword;  
38 > GRANT ALL PRIVILEGES ON realm.* TO realmuser@localhost.localdomain  
39 identified by realmpassword;  
40 > GRANT ALL PRIVILEGES ON realm.* TO realmuser@HOSTNAME identified by  
41 realmpassword;
```

42
43 where **realmuser** and **realmpassword** are mySQL users.

44
45 For the **RS2** database:

```
46  
47 > GRANT ALL PRIVILEGES ON rs2.* TO rsdbuser@% identified by rsdbpassword;  
48 > GRANT ALL PRIVILEGES ON rs2.* TO rsdbuser@localhost identified by  
49 rsdbpassword;  
50 > GRANT ALL PRIVILEGES ON rs2.* TO rsdbuser@localhost.localdomain  
51 identified by rsdbpassword;  
52 > GRANT ALL PRIVILEGES ON rs2.* TO rsdbuser@HOSTNAME identified by  
53 rsdbpassword;
```

1
2 where `rsdbuser` and `rsdbpassword` are MySQL users.
3



A suggestion: probably you use the same mysql installation for **realm** and **rs2** database. So you can use the same mysql user and password for **realmuser** and **rsdbuser** (for instance **rcms** and **rcms2**).

4 5 6 7 2.4.2 *Configuring the Tomcat installation*

8
9 In this section we assume that your environment variable **CATALINA_HOME** is set to your
10 Jakarta tomcat installation.

```
11  
12 > setenv CATALINA_HOME <Jakarta-tomcat-5.0.28-installation>
```

13
14 RCMS 2 needs a tomcat installation configured to use a **Realm MySQL** database (see
15 paragraph 3.4) to store tomcat user names, passwords and permissions.

16
17 Make the mySQL driver visible to tomcat (the driver is available in the RCMS2 distribution):

```
18  
19 > cp RCMSInstallationDir/framework/rcms/WEB-INF/lib/mysql-connector-java-  
20 3.1.6-bin.jar ${CATALINA_HOME}/common/lib
```

21
22 Modify the tomcat **server.xml** configuration file located in the directory
23 `$(CATALINA_HOME)/conf`.

24 Set here the correct information about the realm database you created in paragraph 3.3.

25 The directory **RCMSInstallationDir**/framework/configuration/tomcat contains an example
26 configuration file (`server_example.xml`).

27
28 Tomcat by default runs on port 8080.

29 Modify the `server.xml` if you want to change this default port.

30 More information can be found on the Apache web site (<http://apache.org>).
31
32

33 2.4.3 *Configuring RCMS*

34
35 All the configuration files needed by the RCMS services are located in the directory
36 **RCMSInstallationDir**/TriDAS/RunControl/framework/installation/configuration.

37 Here is the list of the configuration files:

- 38 ○ `rcms_build.properties`
- 39 ○ `RCMS.properties`
- 40 ○ `log4jDefault.properties`
- 41 ○ `web.xml`

42
43 Go to the configuration directory:

```
44  
45 > cd RCMSInstallationDir/TriDAS/RunControl/framework/installation/configuration  
46  
47
```



1
2 The **rcms_build.properties** file is used by the ant build scripts to compile and deploy the
3 software.

4 RCMS base directory and Jakarta Tomcat full path have to be specified:

```
5  
6 rcms.home=RCMSInstallationDir/framework  
7 catalina.home=<your_tomcat_installation>
```

8
9 For instance:

```
10  
11 rcms.home=/users/rcms/TriDAS/RunControl/framework  
12 catalina.home=/users/rcms/jakarta-tomcat-5.0.28
```

13
14 The **RCMS.properties** file contains the configuration information that allows the RCMS
15 services to connect to the RS2 database:

```
16  
17 dbURL=<jdbc_url_for_db_connection>  
18 user=rsdbUser  
19 password=rsdbPassword
```

20
21 For instance:

```
22  
23 dbURL=jdbc:mysql://localhost:3306/rs2 ( for oracle  
24 jdbc:oracle:thin:@oracms.cern.ch:1521:cmsomds )  
25 user=rcms  
26 password=rcms2
```

27
28 The **log4jDefault.properties** file is used by the RCMS services to send log4j log messages.
29 Log messages can be sent to a file, to the console, directly to Chainsaw, or to the RCMS Log
30 Collector. See the log4j and the Log Collector documentation for more details.

31 By default log messages are stored in a file.

32 The perl installation script modifies log4jDefault.properties in order to use the RCMS Log
33 Collector.

34
35 The **web.xml** file is the descriptor for the RCMS web application. You might need to modify
36 the **resourceServiceEndpointAddress** and the **configurationsEndPointPrefix** entries
37 those are used by the GUI:

```
38  
39 <param_name>  
40 resourceServiceEndpointAddress  
41 </param_name>  
42 <param_value>  
43 http://<tomcat_host>:<tomcat_port>/rcms/services/RSRetrieveService  
44 </param_value>  
45 .....  
46 <param-name>  
47 configurationsEndPointPrefix  
48 </param-name>  
49 <param-value>  
50 http://<tomcat_host>:<tomcat_port>  
51 </param-value>
```

1
2 Once you have modified all the configuration files, you can run the ant build script located in
3 the same directory. Use classic-ant on Linux machines.

4
5 > classic-ant

6
7 You must re-run the ant script and redeploy the RCMS web application every time you
8 modify one of these files.
9
10

11 2.4.4 *Deploying RCMS*

12
13 Once you have configured the RCMS you need to deploy it.

14
15 Go to the directory **RCMSInstallationDir**/framework/installation/deployment

16
17 > cd **RCMSInstallationDir**/framework/installation/deployment

18
19 Run the deployment build file using the **release** target:

20
21 > classic-ant release

22
23 The RCMS framework is now ready to be used.

24 Just startup your tomcat installation:

25
26 > cd \${CATALINA_HOME}

27 > bin/startup.sh

28
29 Connect now with a browser to <http://<your host>:<your port>/rcms> .

30 For instance <http://pccms.cern.ch:8080/rcms> .

31
32 You should see the RCMS web page.

33
34 A login and password will be asked. Type your tomcat realm user and password (for instance
35 **rcms** as user, and **rcms2** as password).

36
37 Clicking on “Configuration Chooser” you should see an empty blank page.

38 The reason is that you do not have any configuration inserted in your Resource Service 2.

39
40 The next chapter will explain how to configure and run your first “Hello” Example.
41

3 *Run the Hello Example*

The installation script (paragraph 2.3) installs and deploys the example. Go to paragraph 3.3 in that case.

Once you have correctly installed the RCMS software you need to install and run your first Hello Example.

The directory **RCMSInstallationDir/functionmanagers/examples** contains a directory named **helloFM1**.

3.1 *Compile and deploy the HelloFunctionManager example*

The HelloFunctionManager example is a simple Function Manager implementation that does not control any resource, but simply changes its State upon the reception of commands from the GUI.

The document “**RCMS Function Managers – Getting Started**” explains how to write your own Function Manager implementation, and how to compile and deploy it.

For your first example, just follow these steps.

Go to the helloFM1 example directory:

```
> cd RCMSInstallationDir/functionmanagers/examples/helloFM1
```

Go to the installation directory:

```
> cd installation
```

Compile and deploy the State Machine:

```
> classic-ant
```

The build script should work as it is. If not, edit the build.xml and verify the path to the rcms_build.properties file is correctly set.

Go to your web browser.

The link http://<your_host>:<your_port>/functionmanagers shows a directory listing. It must contain the **HelloFunctionManager.jar** file.

3.2 Fill the Resource Service

In order to run our Hello example The Resource Service database has to be filled with the proper configuration.

There are two main tools to fill the database:

- The DUCK (Data Utility for Configuration Knowledge)
- The DUCK-CAD: a graphical user interface.

Both the tools have their own manual where you can find more detailed information.

In order to provide the configuration for our Hello example we use the DUCK tool. The tool reads configuration information from a file and stores them into the Resource Service database.

Go to the `duck` directory of the `HelloStateMachine` example:

```
> cd RCMSInstallationDir/statemachines/examples/HelloStateMachine/duck
```

The directory contains two duck files: `hello.duck` and `hellotop.duck`.

`hello.duck` fills the Resource Service with a very simple configuration, just a Function Manager, while `hellotop.duck` fills the RS with a configuration containing two Function Managers; a top one that controls a leaf one.

Let us use the simplest configuration; `hello.duck`.

The `hello.duck` file needs a few modifications related to your specific configuration.

`rcms` is the owner of the configuration you are going to store. The owner must be a tomcat user as defined in the realm database. Replace `rcms` with `your_realmuser` if you did not choose `rcms` as user name.

The URL <http://localhost:8080> has to be replaced with the actual host and port where you are running your tomcat.

Now you can run the DUCK tool:

```
> java -jar  
RCMSInstallationDir/tools/DuckCAD/DuckConfiguratorParser/duck.jar  
hello.duck rsdbuser rsdbpassword rs_jdbc_url
```

Where `rsdbuser` and `rsdbpassword` are the MySQL user and password for the Resource Service DB, and `rs_jdbc_url` is the JDBC url of your RS DB.

For instance:

```
> java -jar  
/users/rcms/TriDAS/RunControl/tools/DuckCAD/DuckConfiguratorParser/duck.jar  
hello.duck rcms rcms2 jdbc:mysql://localhost:3306/rs2
```

3.3 *It works*

The HelloFunctionManager was compiled and deployed.
The configuration was inserted in the Resource Service database.

Now you should be able to drive the Hello Function Manager using the web GUI.
(http://<your_host>:<your_port>/rcms).

After being logged in.

Clicking on “**Configuration Chooser**” you see a Directory named **Hello**.

Clicking on the Hello directory the configuration **helloSimple** appears.

Clicking on the helloSimple configuration a control page containing the chosen configuration is displayed.

Click now on the **Create** button.

The Function Manager is created and our HelloFunctionManager.jar is loaded.

The HelloFunctionManager allows three commands: **TURNON**, **TURNOFF** and **Reset**.
Clicking on them the Function Manager changes its state.

The **Destroy** button will kill the Function Manager.

The **Detach** button will just disconnect the GUI from the Function Manager, without killing it.

After a Detach the **Attach** button will allow you to reconnect the GUI to the Function Manager previously created and running on your tomcat.

The **Status Display** button will show a tree containing the Status of all the resources defined in your configuration.

The **Refresh** button will just Refresh the GUI.

The **Diagnostic Page** tab displays the configurations of the log4j logger and Resource Service DB.

The **Monitoring Tools** tab displays the running Function Managers.

The **Logging Collector** tab goes to the Logging Collector page.

1 4 *Appendix A: Configuring and using Eclipse 3.0*

2
3 The Eclipse Integrated Development Environment (<http://www.eclipse.org>) is a tool used to
4 develop the RCMS 2 software.

5 The RCMS can of course be deployed and used without the Eclipse platform.
6 Eclipse might anyway help the development, debugging and deployment and of User Custom
7 Function Manager code.

8
9 If you want to use Eclipse this appendix provides some information on how to configure and
10 use it with the RCMS software.
11

12 *Configuring Eclipse IDE*

13
14 First of all you need to install the Eclipse platform on your system.

15 At the purpose follow the instructions on the Eclipse web site.

16 To install Eclipse, follow the instructions on the Eclipse web site (<http://www.eclipse.org>).

17 Eclipse can work together with your tomcat installation. You will have full tomcat support
18 only if you install Eclipse on the same host where you run your tomcat.
19

20 Many plugins can be installed on the Eclipse platform.
21

22 *Tomcat plugin configuration*

23 <http://www.sysdeo.com/eclipse/tomcatPlugin.html>

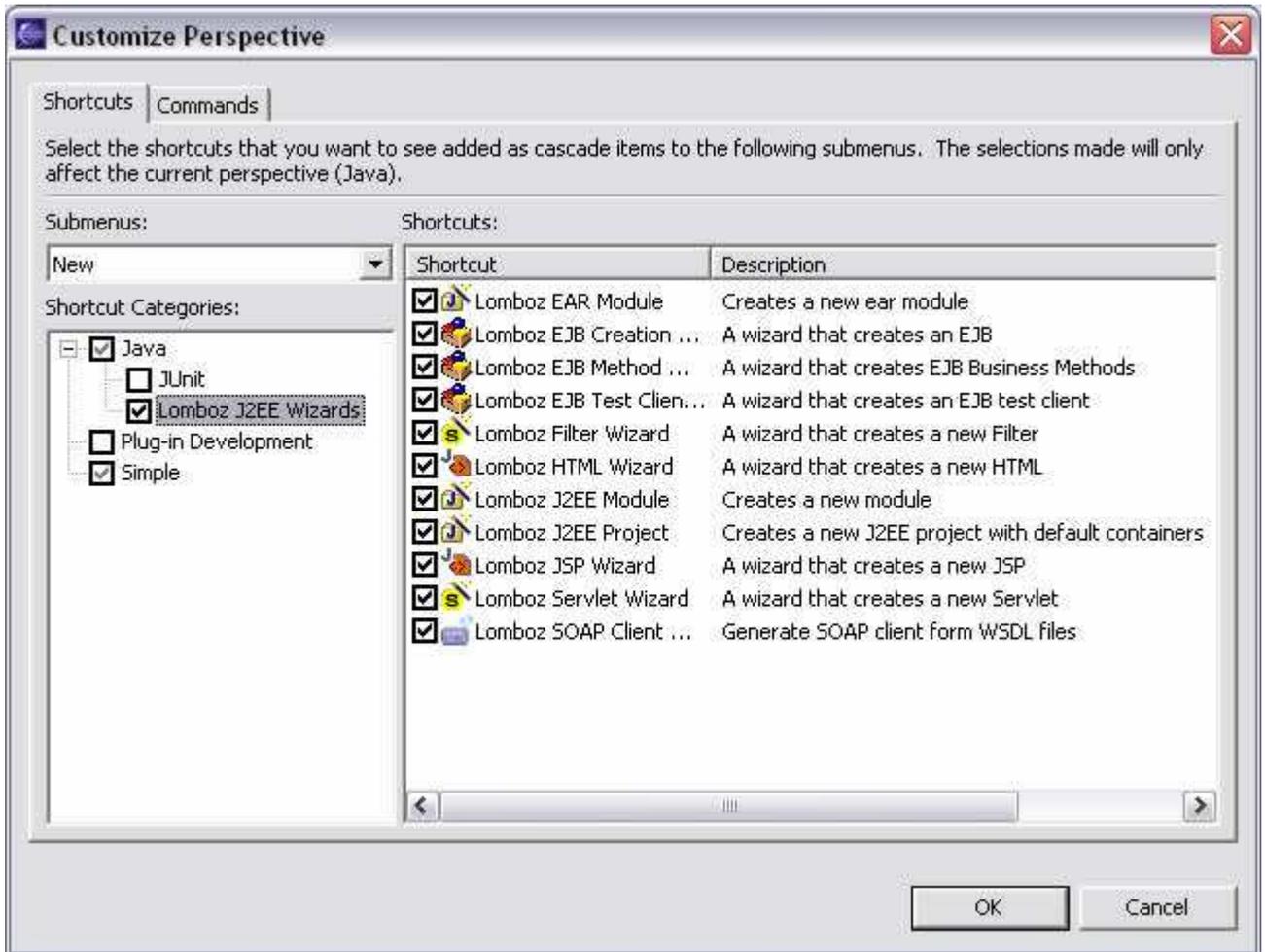
- 24 • Download and install Tomcat before using this plugin.
- 25 • Plugin activation for Eclipse 3.0 :
26 select menu 'Window->Customize Perspective...->Commands', and check 'Tomcat' in
27 'Available command groups'
- 28 • Set Tomcat home : Window -> Preferences, select Tomcat and set Tomcat home. Set
29 Context files in Context Declaration Mode.
- 30 • This plugin launches Tomcat using the default JRE checked in Eclipse preferences
31 window.
32 To set a JDK as default JRE for Eclipse open the preference window : Window ->
33 Preferences -> Java -> Installed JREs.
34 This JRE **must** be a JDK (This is a Tomcat prerequisite).
- 35 • The plugin sets itself Tomcat classpath and bootclasspath. Use Preferences -> Tomcat
36 ->JVM Settings, only if you need specific settings.

1 *Lomboz Plugin configuration*

2 <http://www.objectlearn.com/support/docs/installation/install.jsp>

3 The Lomboz plugin is needed only for RCMS internal development and not for Custom State
4 Machine development.

- 5 • Plugin Activation: Once installed, Lomboz plugin must be activated. Lomboz
6 currently adds actions, wizards and views to the Eclipse environments. To
7 activate these add-ons:



8
9 ■ Select menu 'Window>Customize Perspective...', in the Shortcuts tab choose 'Submenus:
10 New'

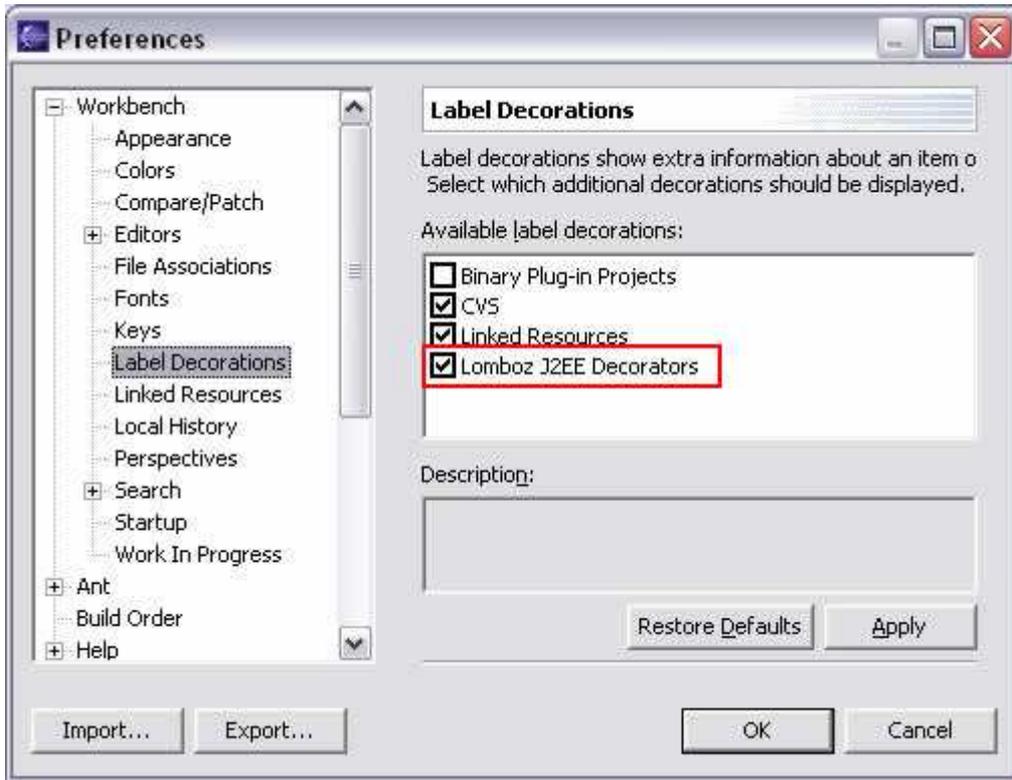
11 check all Lomboz related items. (Click OK button).

12 ■ Select menu Window>Customize Perspective...', in the Shortcuts tab choose 'Submenus:
13 Show View'

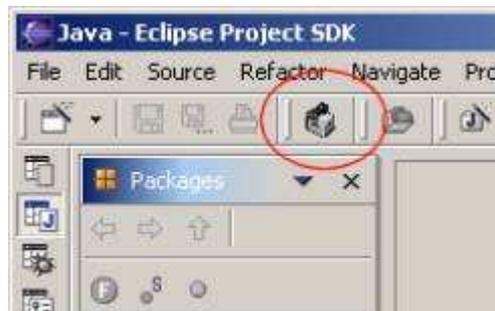
14 check 'Lomboz Views', J2EE View. (Click OK button)

15 ■ Select menu Window>Customize Perspective...', in the Commands tab in the available
16 commands tab, check Lomboz Actions. (Click OK button).

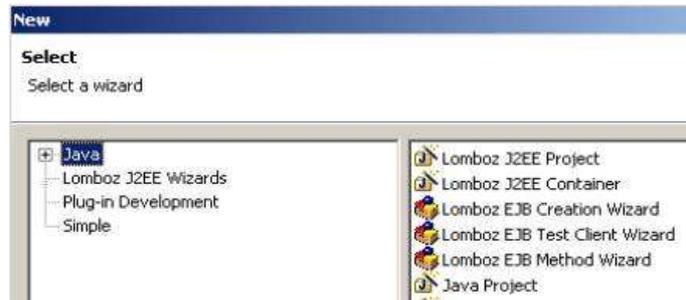
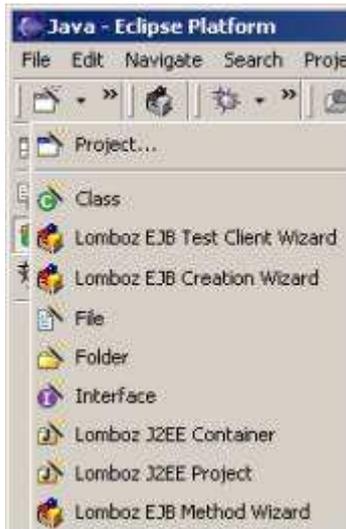
- 1 ■ Select menu Window>Preferences', in the Workbench->Label Decorations, check
- 2 Lomboz J2EE Decorators.



- 3
- 4 Once you have confirmed your selections, you will see the newly added Toolbar button:



- 5
- 6 and the new wizards will be available to you from the New project dialog and the quickly
- 7 accessible New Items menu.



1

2 You must complete the following minimum configurations before you can start using
3 Lombob as your J2EE development tool.

4

- Global Settings : Most application servers use the standard Java compiler (javac) to compile JSP files. Javac is found in the tools.jar distributed with standard Java JDKs (NOT JREs). You must select the tools.jar that will be used by the application servers here. It is normally found inside the “lib” folder under the JDK installation (for instance the path is /usr/java/j2sdk1.4.2_04/lib/tools.jar).

5

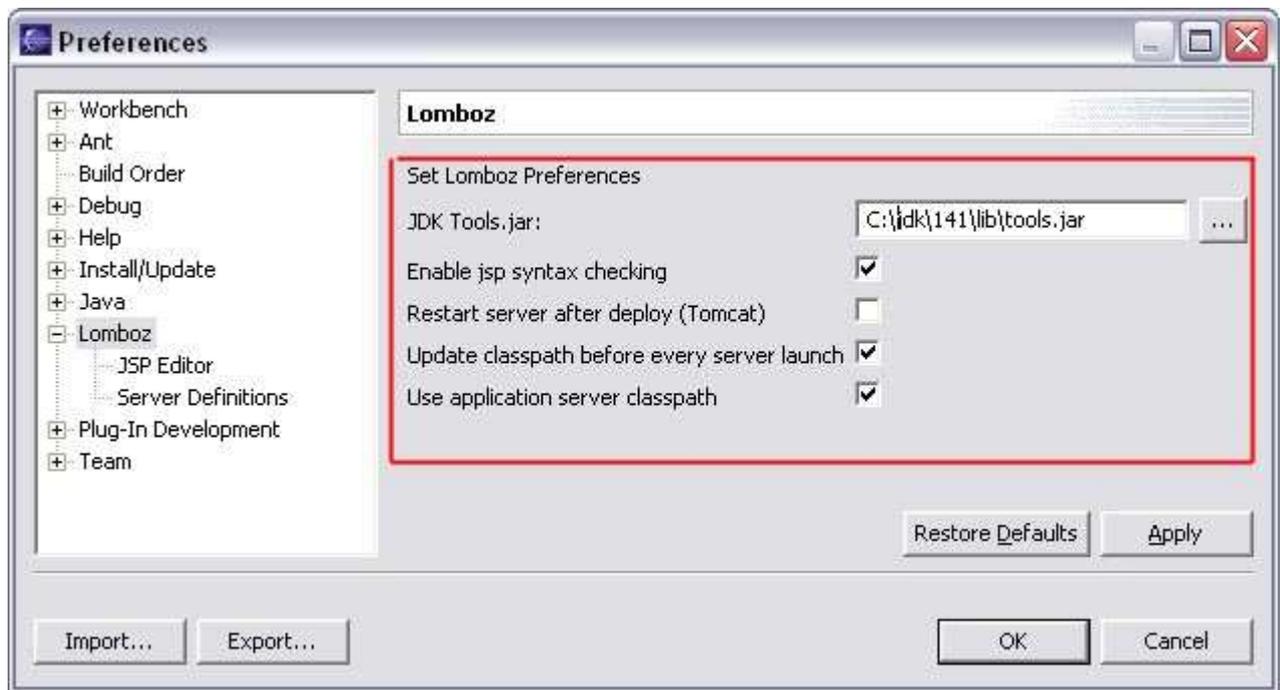
6

7

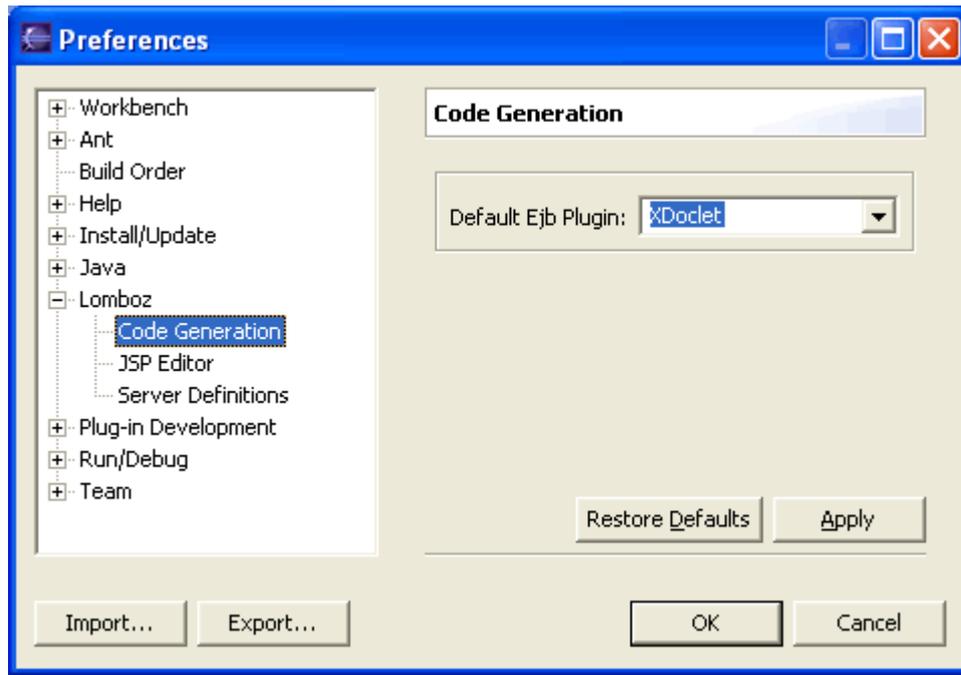
8

9

10

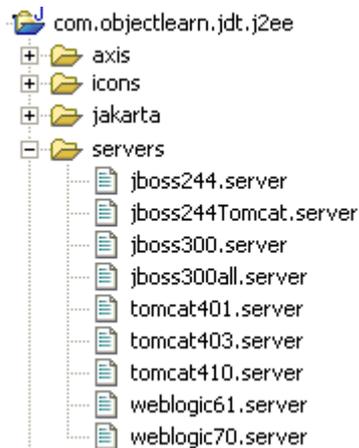


11



1
2
3
4
5
6
7
8
9
10

- Server Definitions: Lomboz can work with almost all Java application servers. You will have to set these parameters so that Lomboz can create scripts which point to the right places. The parameters you set here are used for setting default values for launching the servers and finding the location for deploying Web applications (.war files) and EJB jar files. These files are stored in a folder named "servers" inside the plugin. Lomboz scans the folder everytime it needs a definition. Adding a new server type is adding one of these files into this folder.



11
12
13
14
15
16
17

- In order to use (activate) any of these definitions you must ALWAYS visit the Lomboz preferences. This will load the files in the servers folder into eclipse.
- Next task is to check and set the properties and classpaths defined for these servers. Although there are predefined values for these properties, it is most likely that they are incorrect for your configuration. It is very important that you click APPLY after any change.

1 *Properties page*

2 To change the properties open the preferences dialog from Workbench -Preferences menu
3 and highlight the Lomboz server definitions item in the list. You must enter the following
4 parameters:

5 **Classpath variables:**

6 In order for Lomboz to find the proper application server classes without hard coding them
7 into Java projects we use classpath variables. These class variables are automatically added
8 into the Eclipse environment each time you apply these properties.

9 To verify that these variables are set, open the preferences dialog from Workbench -
10 Preferences menu and Expand the Java item first highlight the 'Classpath Variables' item
11 under the Java list. You SHOULD SEE the values such as 'WEBLOGIC700', 'JBOSS300'
12 and 'TOMCAT410' etc, which will point to the application server directories.

13 All server definitions have these variables. You can now choose a name for the classpath
14 variable you would like to use and set the path for this variable. These settings will be used to
15 add a classpath variable, which you can verify by looking at the Java classpath variables page
16 as described above. The jar files and libraries you will choose in the next few steps will be
17 relative to this classpath variable.

18 **Tomcat:**

Parameter	Description
Server Home Directory	The path to the application server installation directory

19

20 *Download RCMS2 Eclipse projects from CVS*

21 Open a New Eclipse Project.

22 Select "Checkout Projects from CVS".

23 Set the CERN CMS TriDAS configuration for cvs:

24

25 **Host:** cmscvs.cern.ch
26 **Repository:** /cvs_server/repositories/TriDAS
27 **User:** <your_username>
28 **Password:** <your_password>
29 **Connection Type:** extssh

30

31 Click on Next.

32

33 Select "Use an Existing Module".

34 Select the Folder "RunControl/framework"

35

36 Click on Next.

37

38 Checkout as a project in the workspace.

39

40 Click on Next.

41



-
- 1 Use default workspace location.
2
3 Click on Next.
4
5 In the Select Tag Window Click on “**Refresh from Repository**”.
6 Then choose the tag you want.
7
8 Click on Finish.
9
10 Now you have an Eclipse project called **framework**.
11
12 Change the Properties of the framework Project:
13 - Right click on the Project and select Properties.
14 - Select Tomcat.
15 - Check if the “is a Tomcat Project” is selected. If so remove it. Click on Apply and
16 then OK.
17 - Return into the properties of the framework project.
18 - Select Is a Tomcat Project
19 - Put **/rcms** as “Context Name”
20 - Put **/rcms** as “Web Application root”
21

22 *Creating Projects for the State Machines*

23 Download using CVS the State Machine examples as new Eclipse projects. The examples are
24 in the cvs folder RunControl/statemachine/examples.
25

26 Once you have checked out the State Machine you might need to change the Eclipse Project
27 properties. It depends on the RCMS **framework** Project in order to compile.
28

- 29 - Right click on the Project and select Properties.
 - 30 - Select Java Build Path.
 - 31 - Select Projects.
 - 32 - Click on the framework Project.
- 33
34
35

1

2 **References**

3

