

# Installing LaCOLLA 1.0.X

version 1	march 26, 2007	-	Enric Jaen
version 2	april 24, 2007	modified section 2.2	Enric Jaen

## Table of Contents

1. Introduction.....	1
2. LaCOLLA installation files.....	1
2.1 build.xml.....	1
2.2 GAPAs, UAs and RAs initialization files. ....	2
2.3 Initial members.....	3
2.4 Database files.....	4
2.5 Logging configuration file.....	4
2.6 Simulation files.....	4
3. LaCOLLA startup.....	4

## 1. Introduction

This document explains how to install LaCOLLA 1.0.X and the configuration files.

**Differences with LaCOLLA Beta1.0:** With respect the previous version of LaCOLLA, this version is now executed with ANT so the there isn't need to write different scripts for Linux and Windows, and it uses the Log4j package to print log messages. The rest of configuration remain the same.

## 2. LaCOLLA installation files

The current binary distribution of LaCOLLA is zipped in the file: [lacolla-dist-1.0.3.zip](#)

Next is given a description of the contained files, grouped by common functionality.

### 2.1 build.xml

This file contains the ANT project, with the targets to execute LaCOLLA. The targets are the following:

- *init* : only used for compiling LaCOLLA.
- *build*: only used for compiling LaCOLLA.
- **lacolla**: it starts the server. By default, it takes as parameters three configuration files:

gapas, ras, uas.

- **clean**: it removes the logs and database files generated during the execution.

The project has one configurable property: `lacolla.jar`, which must point to the latest released JAR file. The actual value is:

```
<property name="lacolla.jar" location="lib/lacolla-1.0.4.jar"/>
```

## 2.2 GAPAs, UAs and RAs initialization files.

LaCOLLA is preconfigured with 2 GAPAs, 2 UAs and 2 RAs. The main configuration files are named `gapas`, `ras`, `uas`, and their point to more specific configuration files. Their content is the following:

<b>gapas</b>	<b>ras</b>	<b>uas</b>
2	2	2
GAPA1.txt	RA1.txt	UA1.txt
GAPA2.txt	RA2.txt	UA2.txt
.	.	.

Figure 1 illustrates the initial configuration for the GAPAs, RAS and UAS. Notice that the components follow a hierarchical tree.

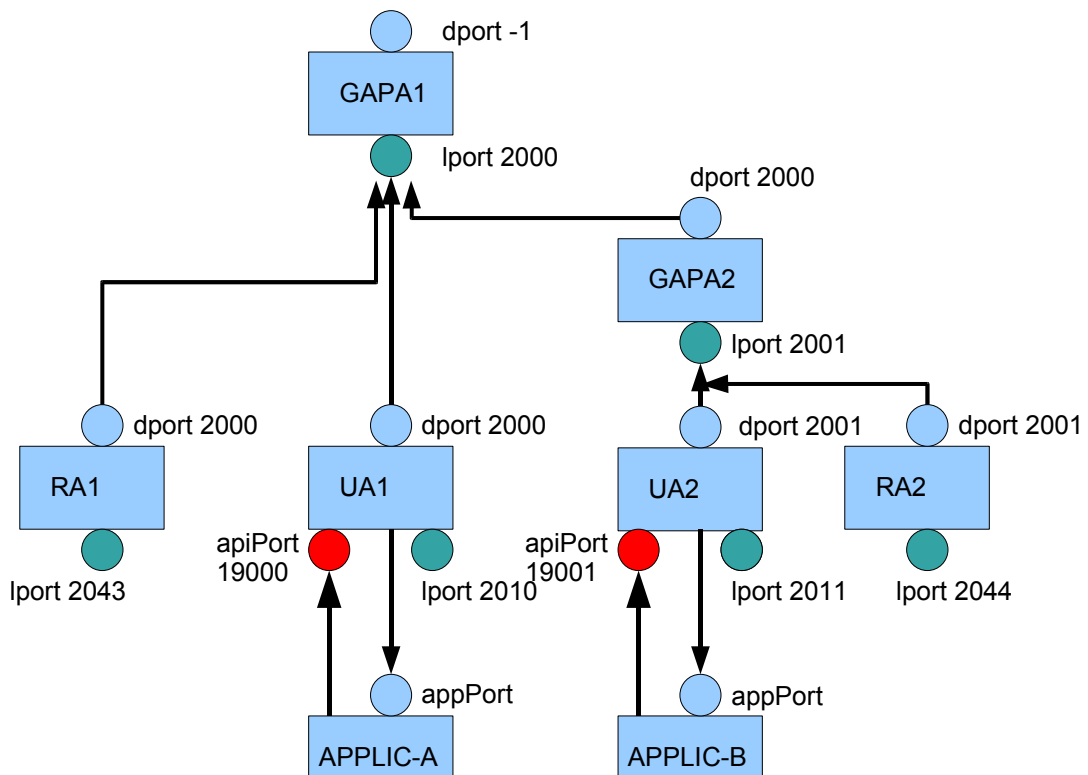


Figure 1. Initial configuration.

The contents of these files is the following:

<b>GAPA1.txt</b>	<b>RA1.txt</b>	<b>UA1.txt</b>
kindOfAgent=0 dHost=none dPort=-1 lPort=2000 groups=hola username=admin password=admin	kindOfAgent=1 dHost=127.0.0.1 dPort=2000 lPort=2043 groups=hola username=admin password=admin	kindOfAgent=2 dHost=127.0.0.1 dPort=2000 lPort=2010 apiPort=19000 groups=hola username=admin password=admin
<b>GAPA2.txt</b>	<b>RA2.txt</b>	<b>UA2.txt</b>
kindOfAgent=0 dHost=127.0.0.1 dPort=2000 lPort=2001 groups=hola username=joan password=flipa	kindOfAgent=1 dHost=127.0.0.1 dPort=2001 lPort=2044 groups=hola username=xavo password=lipa	kindOfAgent=2 dHost=127.0.0.1 dPort=2001 lPort=2011 apiPort=19001 groups=hola username=joan password=flipa

Notice that there is one initial group, `hola`. The user credential fields (username and password) refer to the initial user that has access to the component. A component sends messages to another component located at the pair (dHost, dPort). dHost (destination host) indicates the host where this component is connected. dPort (destination port) indicates the port of the destination host where this component is connected. lPort (local port) indicates the listening port of this component where other components can use to connect to this component. You should notice that the lPort and the dPort ports of two connected components must match. The UAs has an additional listening port (apiPort) where the applications are connected.

## 2.3 Initial members

Members must be registered in the GAPA database. The initial members are configured in the `members.ini` file. Their contents are the following:

<b>members.ini</b>
member1.ini member2.ini .

<b>member1.ini</b>	<b>member2.ini</b>
<pre>#member properties #Wed Mar 21 12:27:20 CET 2007 groups=hola, memberId=member\#0\# Username=admin Password=admin roles=ROOT, memberInfo=adminInfo.ini</pre>	<pre>#member properties #Wed Mar 21 12:27:20 CET 2007 groups=hola, memberId=member\#30425cb8beeb10048043ef2be51aaa1e\# Username=joan Password=flipa roles=ROOT, memberInfo=memberInfo2.ini</pre>

Accordingly the files, there are 2 initial users: `admin` and `joan`. Each user has a group, a `memberId`, username, password and a `memberInfo` field that may contain additional information. Notice that this additional information is optional, and it won't be processed by LaCOLLA. It can be obtained calling the `getMemberInfo()` operation. New users can be created at runtime using the `addMember()` operation.

## 2.4 Database files

LaCOLLA has embedded an SQL engine (HSQLDB). There is an SQL database file (.script) for each RA, GAPA and UA. These file names look like:

```
RA#9adb7868bf0310048ad3f91fef7a8bc9#.script
GAPA#744ddec0bee510048f828f2f3d9085b9#.script
UA#bd327eb0bf3110048958f52c8803bee2#.script
```

During execution, the HSQLDB package also creates the corresponding .log and .properties files.

The database tables can be accessed using any SQL tool (for example, the Quantum eclipse plugin). For example, the configuration parameters for the GAPA1 database are the following:

```
user = "sa"
password = ""
dbname = "GAPA#744ddec0bee510048f828f2f3d9085b9#"
jdbc:hsqldb:/home/enric/workspace/lacolla1.0.0/dist/lacolla/GAPA#744ddec0bee510048f828f2f3d9085b9#
```

## 2.5 Logging configuration file

The user can configure which output is displayed into the console during the execution. The logging is configured with the `resources/config/log4j.xml` file. The file is preconfigured with six appenders: FILE1, FILE2, FILE3, CONSOLE, CONSOLE2, CONSOLE3. Each appender defines a format for displaying the logging entries. By default the CONSOLE3 is used, for example:

```
<logger name="LaColla.core.components.UA"> <level value="debug"/> <appender-ref
ref="CONSOLE3" /> </logger>
```

## 2.6 Simulation files

The simulation configuration files (SA1.txt, simulationParameters.ini, simulationParameters.xml) are not explained in this document.

## 3. LaCOLLA startup

These are the logging messages that appear after starting LaCOLLA:

### **ant lacolla**

Buildfile: build.xml

lacolla:

```
[java] Starting Tyrex Version 1.0.3
[java] Original code is Copyright (c) 1999-2001, Intalio, Inc. All Rights Reserved. Contributions
by MetaBoss team are Copyright (c) 2003-2005, Softaris Pty. Ltd. All Rights Reserved.
```

```
[java] [LaColla.core.components.UA]
[java] UA:    APIPORT=19000
[java] [LaColla.Api.ApiHandler]
[java] constructor:    sense rmisecuritymanager
[java] [LaColla.Api.ApiHandler]
[java] constructor:    Registry created 11099
[java] [LaColla.Api.ApiImpl]
[java] ApiImpl:    inside ApiImpl
[java] [LaColla.Api.ApiHandler]
[java] bind:    new LC API instance
[java] [LaColla.Api.ApiHandler]
[java] bind:    LC API registered as /Api19000
[java] [LaColla.core.components.UA]
[java] UA:    Api Binded!
[java] [LaColla.core.components.UA]
[java] UA:    APIPORT=19001
[java] [LaColla.Api.ApiHandler]
[java] constructor:    sense rmisecuritymanager
[java] [LaColla.Api.ApiHandler]
[java] constructor:    Registry already created 11099
[java] [LaColla.Api.ApiImpl]
[java] ApiImpl:    inside ApiImpl
[java] [LaColla.Api.ApiHandler]
[java] bind:    new LC API instance
[java] [LaColla.Api.ApiHandler]
[java] bind:    LC API registered as /Api19001
[java] [LaColla.core.components.UA]
[java] UA:    Api Binded!
```

Now LaCOLLA is waiting for incoming requests from the applications.