Magentix2
Developers Guide

INDEX

Introduction 3
Download Magentix 2 3
Unzip Magentix 2 3
Pre-requisites 4
  QPID (required) 4
  JDK 6 (required) 5
  ECLIPSE (recommended IDE) 5
  MAVEN2 (recommended management tool) 6
    Eclipse plugin for Maven2 6
    Configure Eclipse IDE (without Maven2) 9
    Configure Eclipse ID (with maven) 10
Magentix2 Examples 13
  Content 13
  Run Magentix2 examples 15
Thomas Examples 17
  Additional requisites 17
    TOMCAT 6.0 (recommended server) 17
    MySQL (recommended DDBB) 17
  Deploy Thomas in TOMCAT 18
  Install Thomas DDBB in MySQL 20
  Run Thomas Example 25

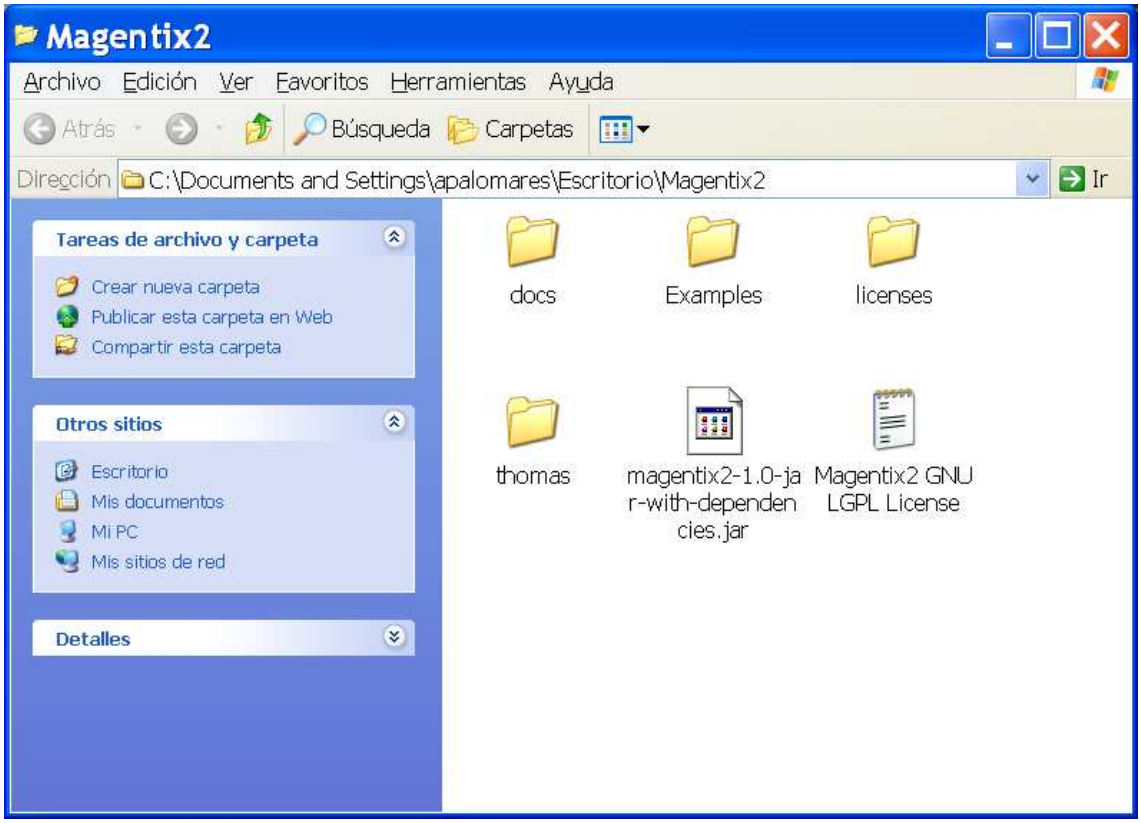
Introduction

This document briefly describes how to install, develop and run some examples of the Magentix2 platform.

Download Magentix 2

http://locke.dsic.upv.es:8080/magentix2/

Unzip Magentix 2



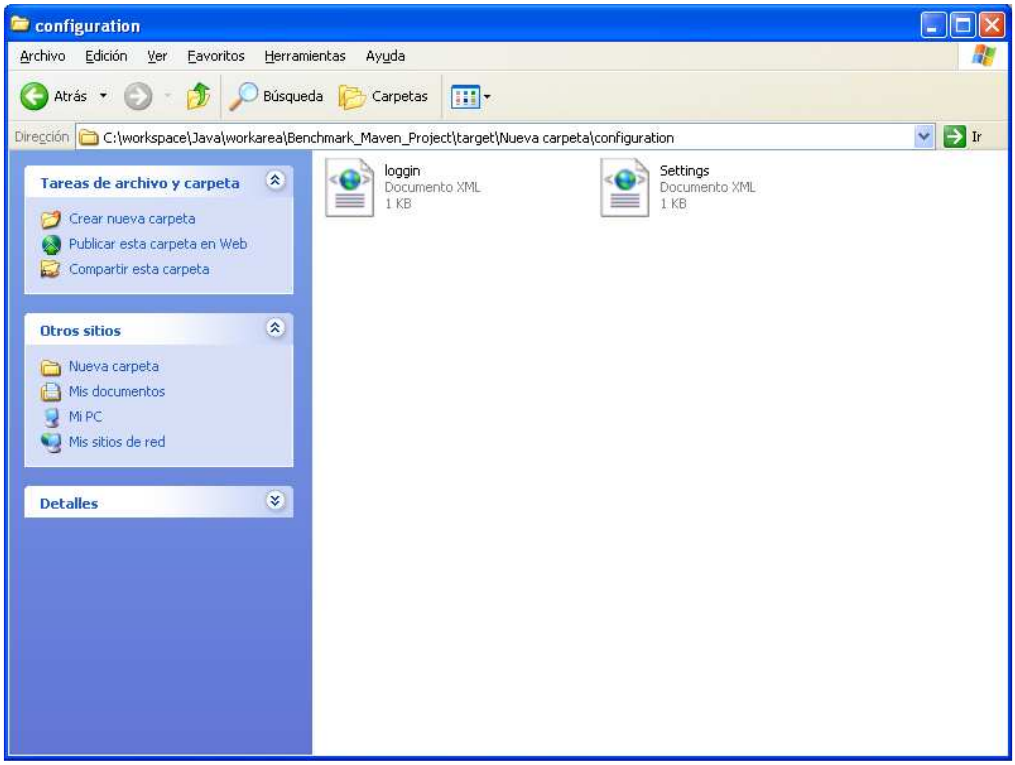
Unzipped Magentix2 directory includes:

- *Magentix2-1.0-with-dependencies.jar*: this jar includes all additional libraries required by Magentix2.
- *Magentix2 GNU PGGL License file*: license statement of Magentix2.
- *Configuration sub-directory*: includes the *settings.xml* and *loggin.xml* configuration files of Magentix2.
- *Examples sub-directory*: includes some examples of Magentix2 agent's implementation.
- *Licenses sub-directory*: includes license statement terms of all software libraries required by Magentix2.
- *Docs sub-directory*: includes guides of Magentix2.

Pre-requisites

QPID (required)

The document "**INSTALL QPID IN UBUNTU 9.04 32 BITS.pdf"** describes how to install Qpid/C++ broker from the source distribution http://qpid.apache.org/ on Linux/Unix operating system. Suppose that QPID c++ broker is running in a host called "broker.host.name" and listen messages on port 5672[1]. Other Broker configuration parameters as "virtualhost"[2], "user" "password" and "ssl" can be configured on the *settings.xml* file of Magentix2 (*configuration directory*):




```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
……….
        <entry key="host"> broker.host.name</entry>
        <entry key="port">port</entry>
        <entry key="vhost">virtualhost</entry>
        <entry key="user">username</entry>
        <entry key="password">password</entry>
        <entry key="ssl">false</entry>
……….
</properties>
```

[1] The QPID C++ broker is pre-configured by default to listen messages on port 5672 but also it is possible configure to listen on other ports (http://qpid.apache.org/rasc.html).

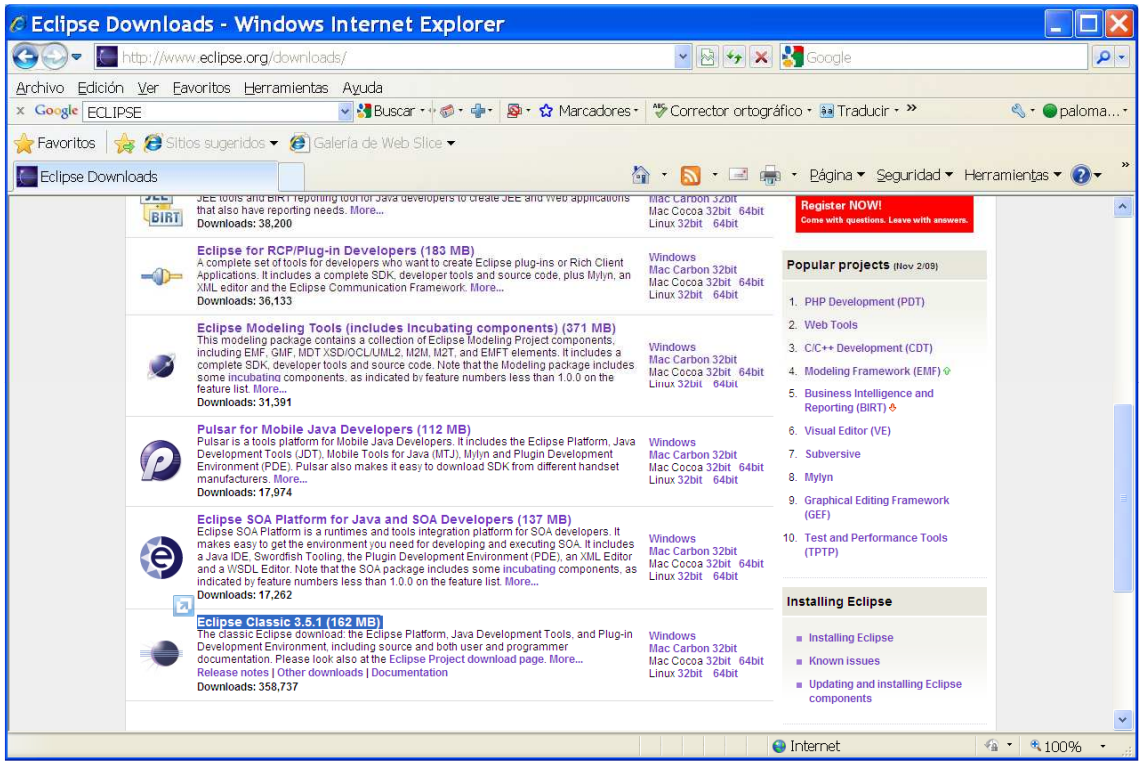
[2] The term Virtual Host refers to the practice of running more than one web site on a single machine. The fact that they are running on the same physical server is not apparent to the end user.

JDK 6 (required)

http://java.sun.com/javase/downloads/index.jsp#jdk

ECLIPSE (recommended IDE)

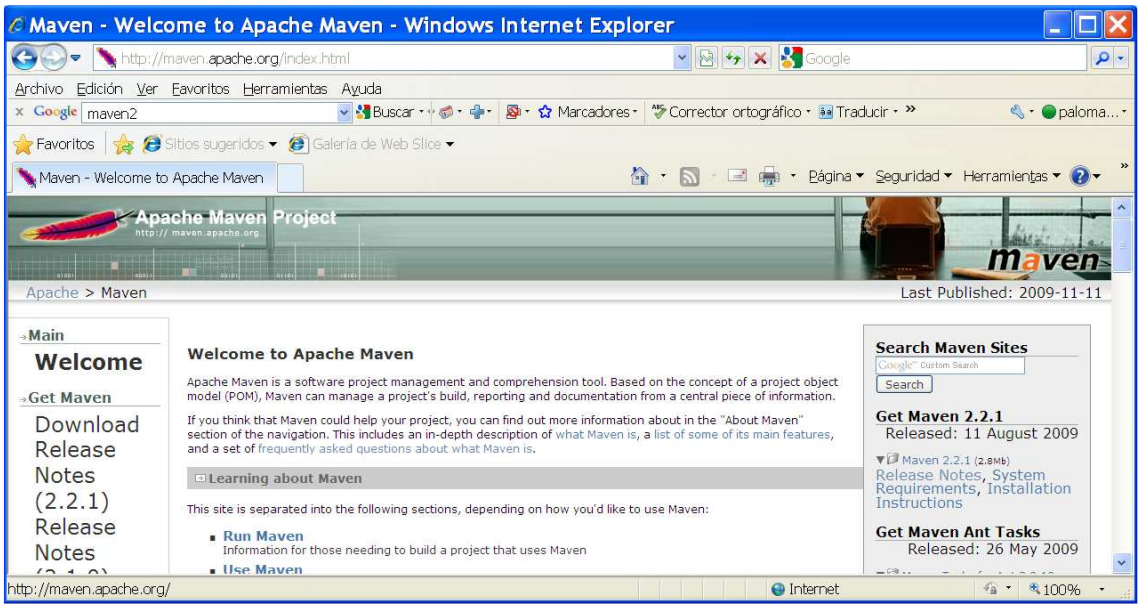
http://www.eclipse.org/



MAVEN2 (recommended management tool)

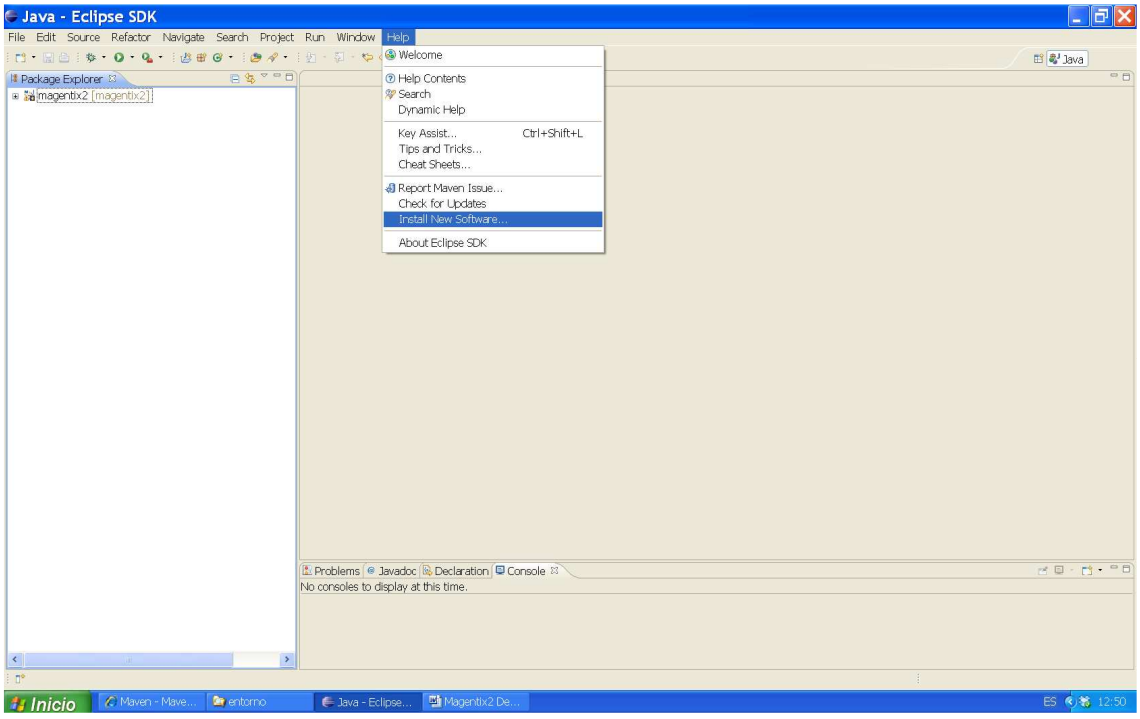
Apache Maven is a software management tool based on the concept of a project object model (POM). Maven can manage a project's build, reporting and documentation from a central piece of information.

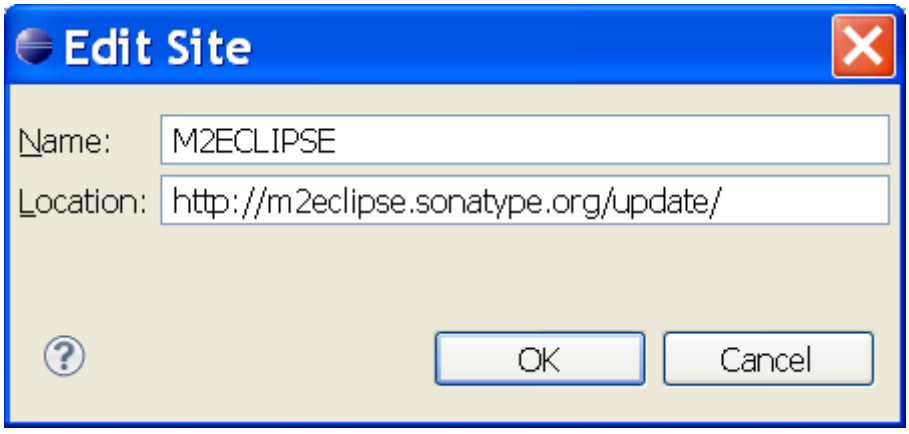
http://maven.apache.org/

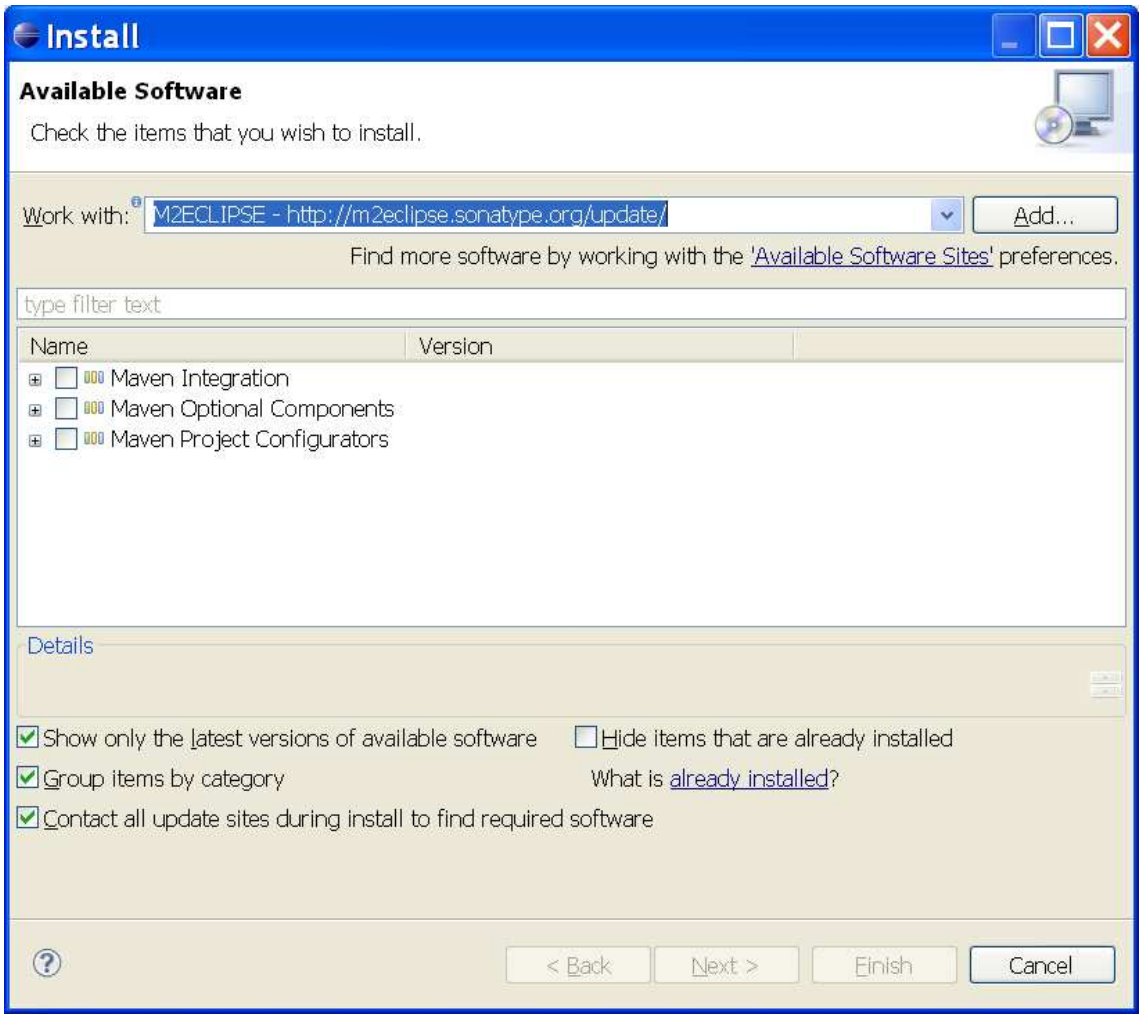


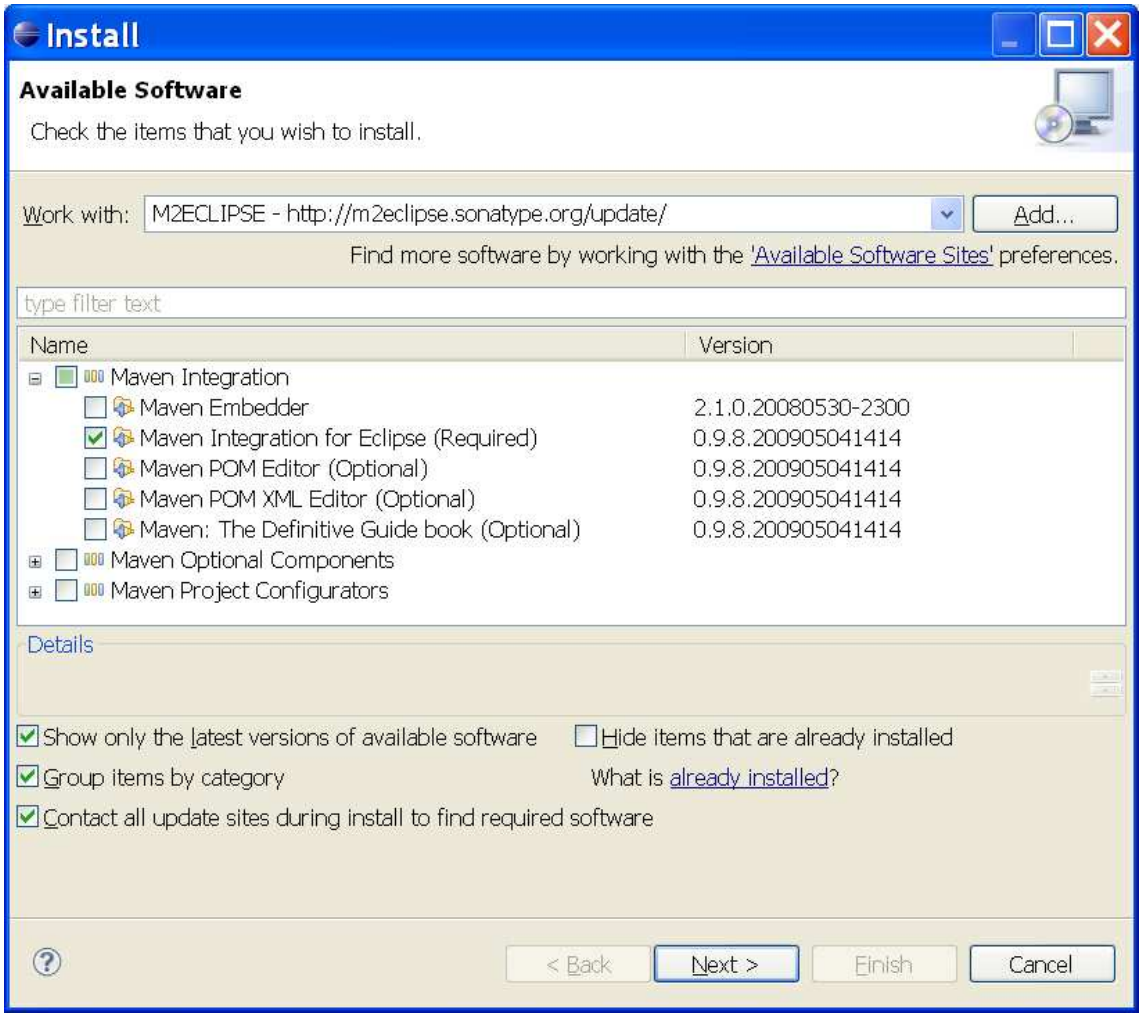
Eclipse plugin for Maven2

M2 Eclipse plugin allow using Maven with Eclipse IDE.

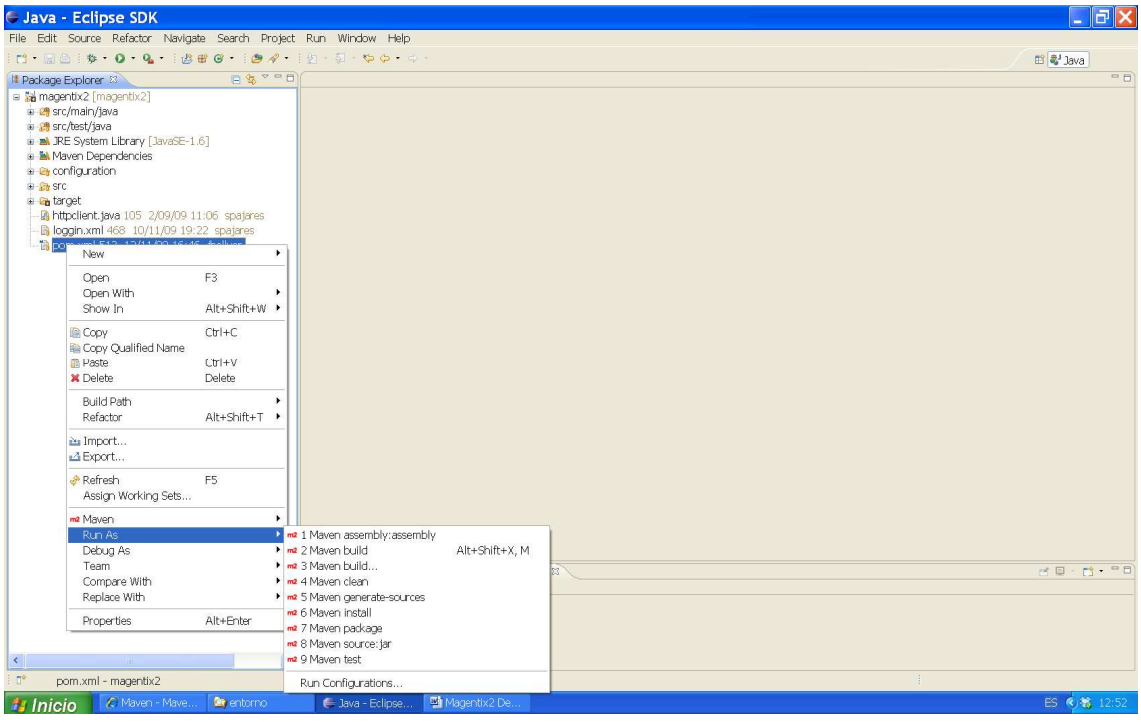



Edit Site
Name:
M2ECLIPSE
Location:
http://m2eclipse.sonatype.org/update/
OK
Cancel


Install
Available Software
Check the items that you wish to install.
Work with:
M2ECLIPSE - http://m2eclipse.sonatype.org/update/
Add...
Find more software by working with the 'Available Software Sites' preferences.
type filter text
Name
Version
Maven Integration
Maven Optional Components
Maven Project Configurators
Details
Show only the latest versions of available software
Hide items that are already installed
Group items by category
What is already installed?
Contact all update sites during install to find required software
< Back
Next >
Finish
Cancel

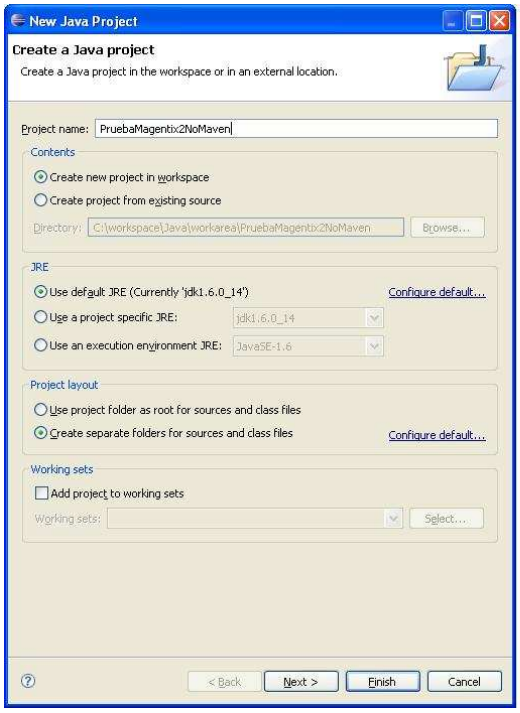


The Maven2 plugin has been installed in Eclipse:

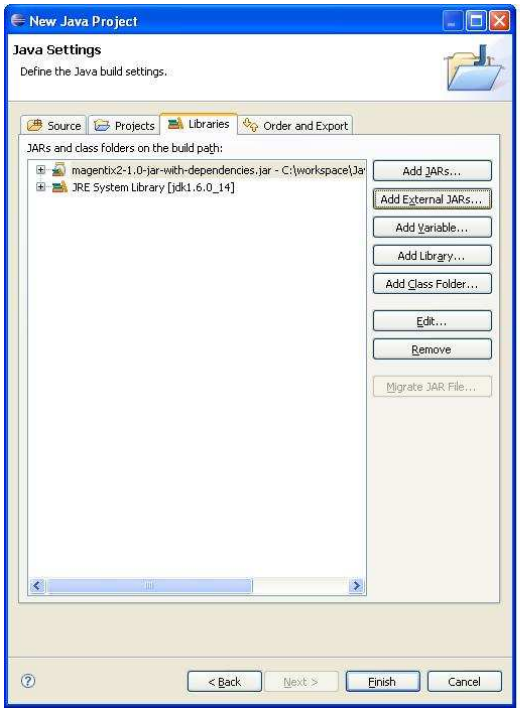


Configure Eclipse IDE (without Maven2)

Create a New Java project:



Add Magentix2.jar dependence

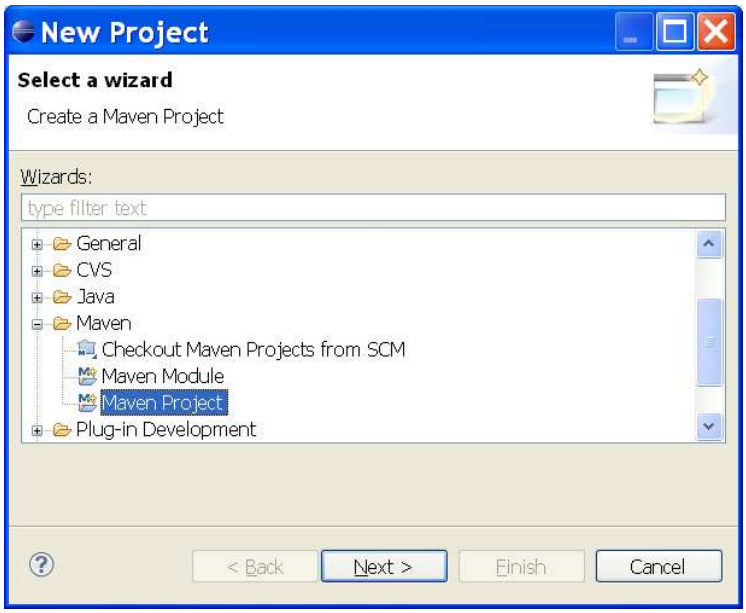


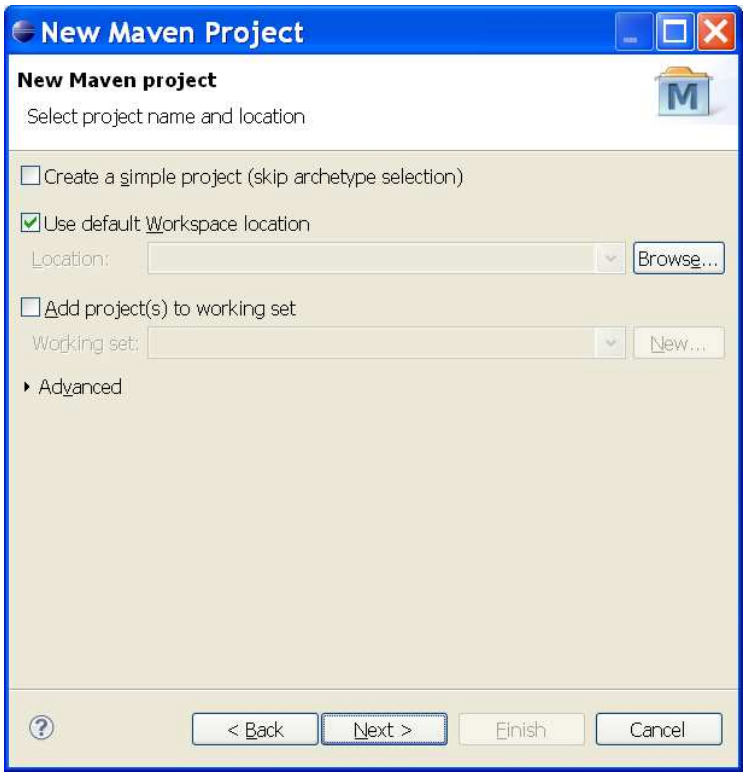
Configure Eclipse ID (with maven)

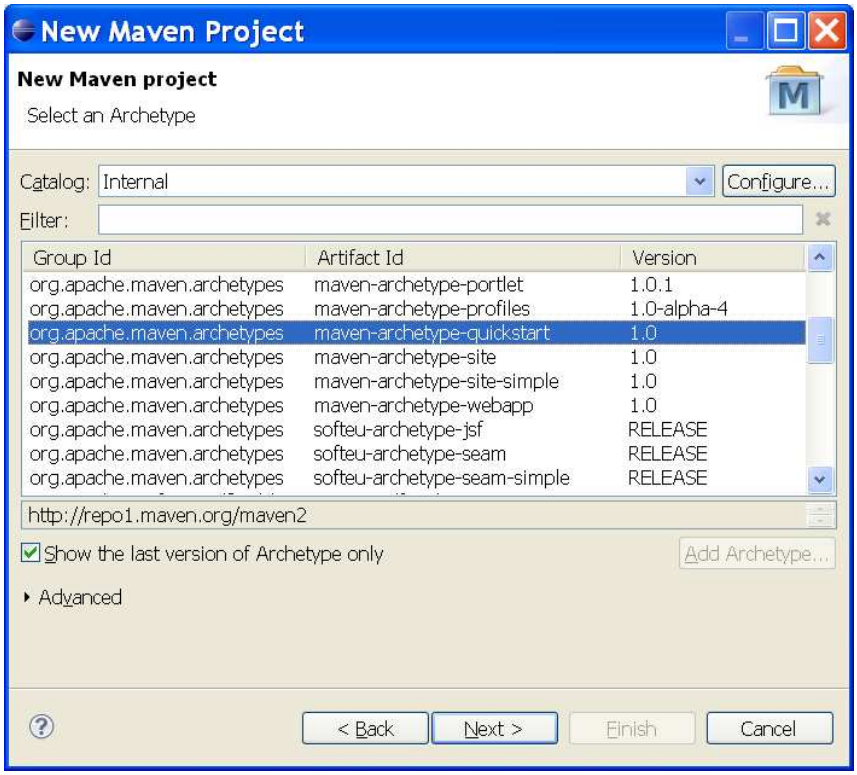
Install magentix2 in "*Maven Local Repo*":

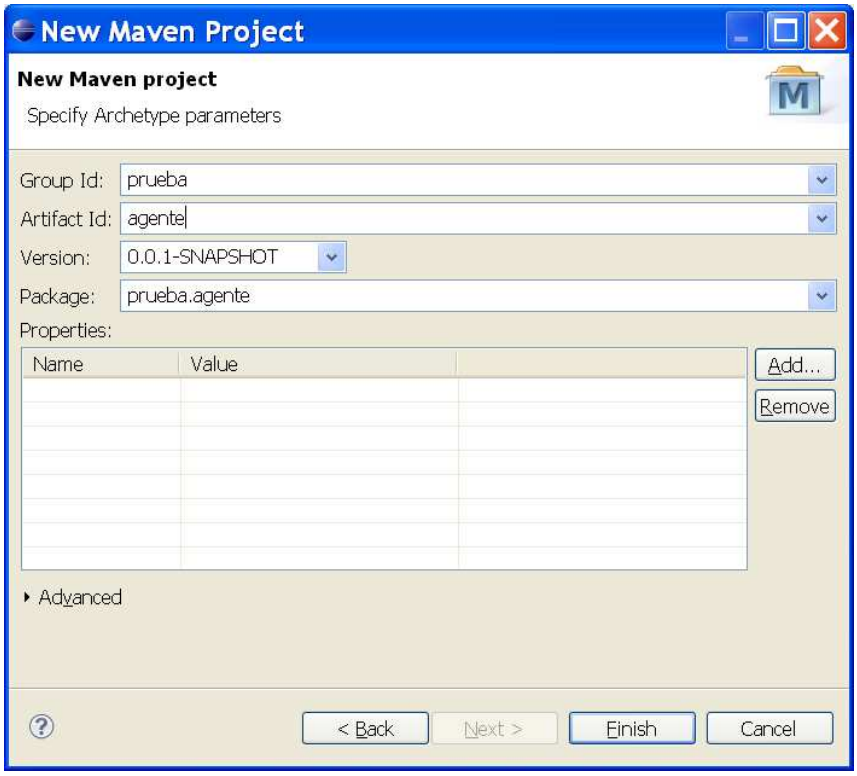
```
mvn install:install-file -Dfile=magentix2_1.0.jar
                         -DgroupId=es.upv.dsic.gti-ia
                         -DartifactId=magentix2
                         -Dversion=1.0
                         -Dpackaging=jar
```

Create a New Maven project:

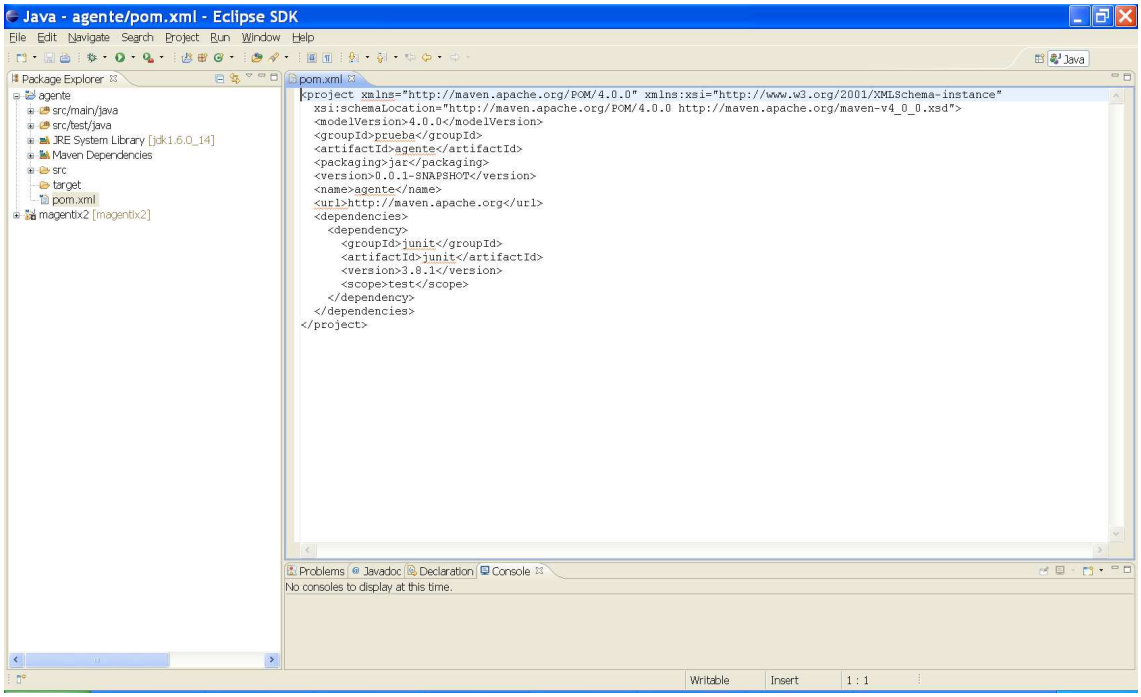





New Maven Project
New Maven project
Select an Archetype
Catalog: Internal
Configure...
Filter:
Group Id
Artifact Id
Version
org.apache.maven.archetypes maven-archetype-portlet 1.0.1
org.apache.maven.archetypes maven-archetype-profiles 1.0-alpha-4
org.apache.maven.archetypes maven-archetype-quickstart 1.0
org.apache.maven.archetypes maven-archetype-site 1.0
org.apache.maven.archetypes maven-archetype-site-simple 1.0
org.apache.maven.archetypes maven-archetype-webapp 1.0
org.apache.maven.archetypes softeu-archetype-jsf RELEASE
org.apache.maven.archetypes softeu-archetype-seam RELEASE
org.apache.maven.archetypes softeu-archetype-seam-simple RELEASE
http://repo1.maven.org/maven2
Show the last version of Archetype only
Add Archetype...
Advanced
< Back
Next >
Finish
Cancel

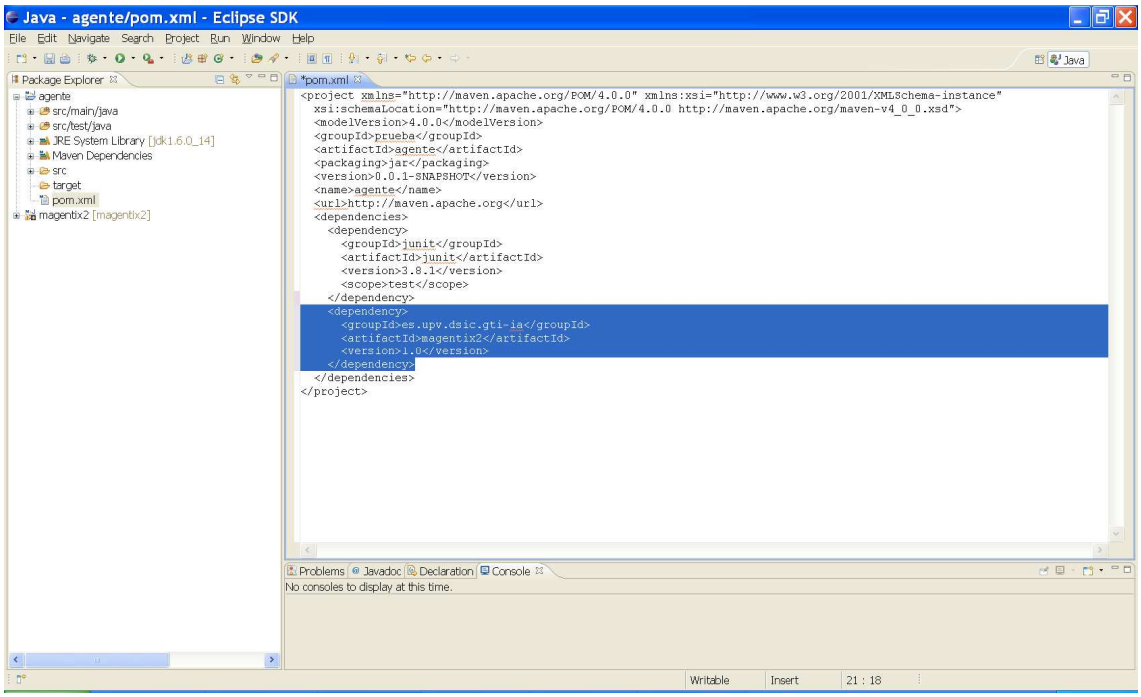

New Maven Project
New Maven project
Specify Archetype parameters
Group Id: prueba
Artifact Id: agente
Version: 0.0.1-SNAPSHOT
Package: prueba.agente
Properties:
Name
Value
Add...
Remove
Advanced
< Back
Next >
Finish
Cancel

A single maven project is created in Eclipse IDE:



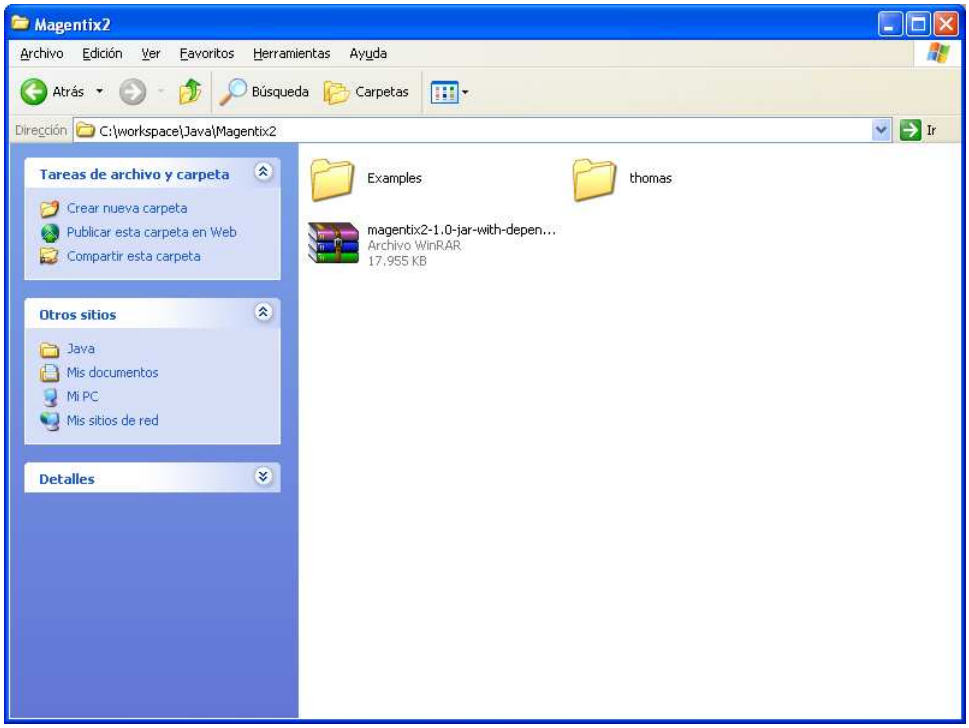
Add Magentix2 dependence in “POM.XML” file

```
<dependency>
        <groupId>es.upv.dsic.gti-ia</groupId>
        <artifactId>magentix2</artifactId>
        <version>1.0</version>
</dependency>
```



Magentix2 Examples

Subdirectory "*examples*" of Magentix2 distribution includes some examples of agents developed for this platform:



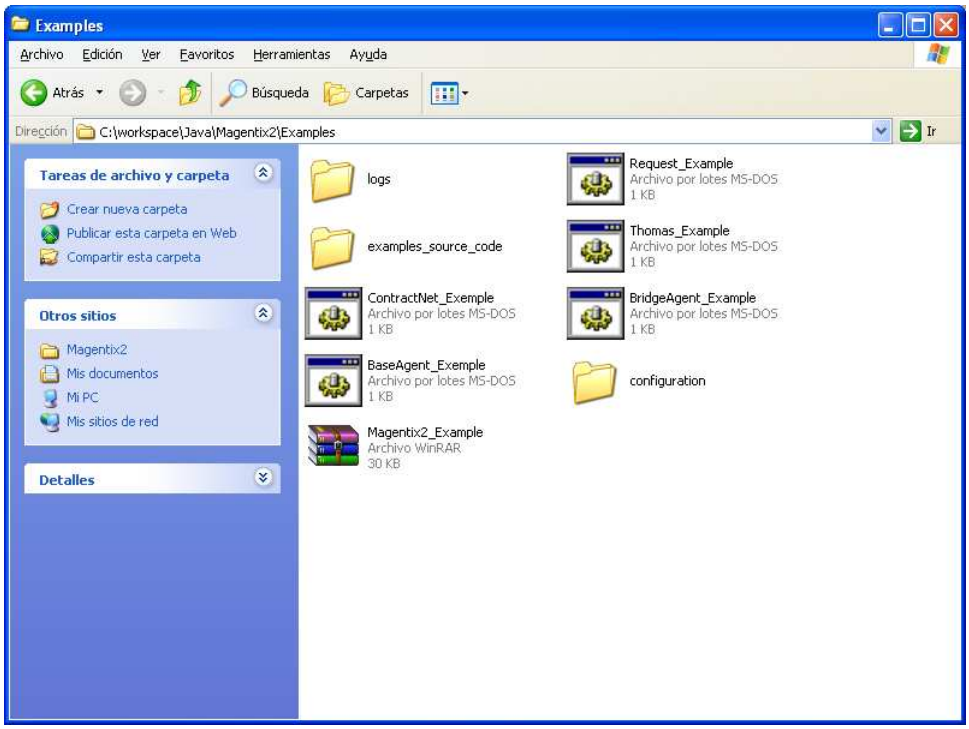
Content

Examples included in this distribution are:

- *BaseAgent*: example of issuer/consumer messages. The sender agent builds and sends an ACLMessage to the consumer. When the ACLMessage arrives, the consumer displays the message on screen.
- *BridgeAgent*: example in which an internal agent send a message to an external agent of our platform.
- *Request*: example of an agent that implements the FIPA Request protocol.
- *ContractNet*: example of an agent that implements the FIPA ContractNet protocol.
- *Thomas*: example of an agent that access to Thomas SF and OMS web services.

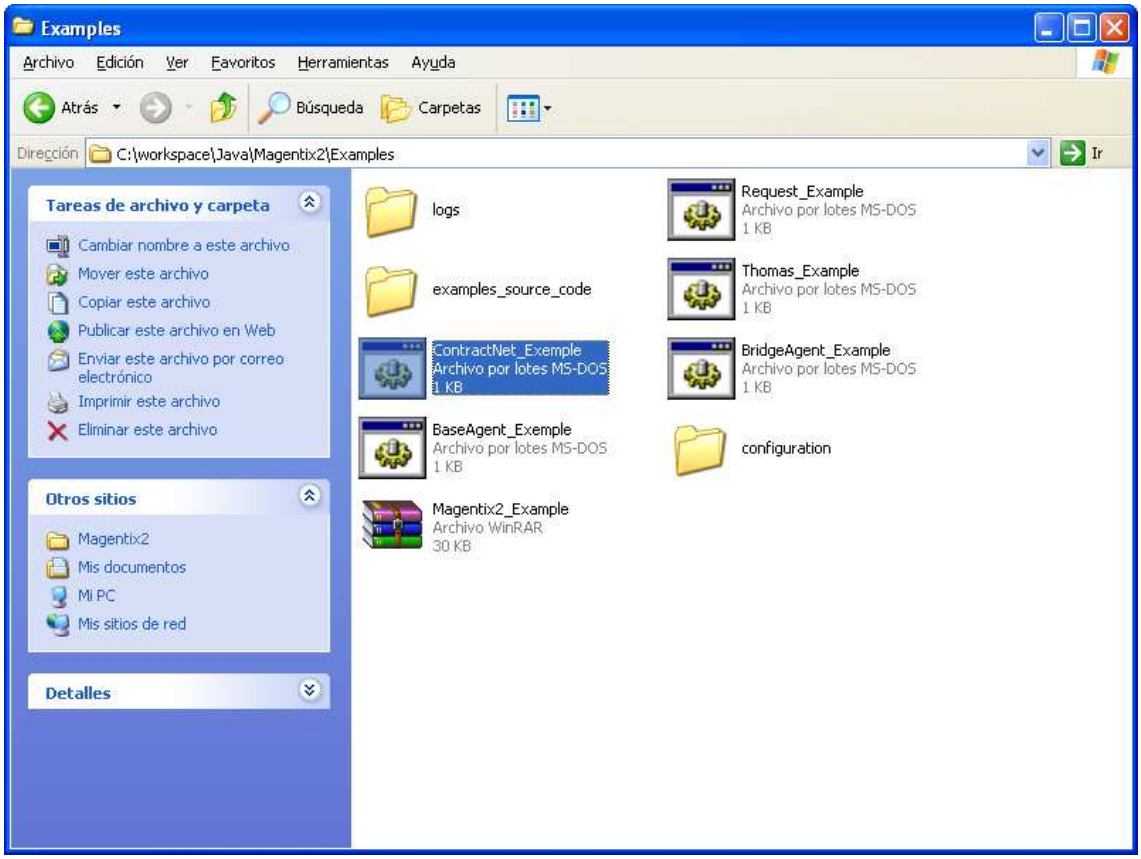
Subdirectory "*examples*" includes:

- "*Magentix2_Example.jar":* compiled code required to run examples.
- "*/examples_source_code*" subdirectory: the source code of examples.
- *"*.bat files*": run examples directly (windows)[3].
- *"/logs" subdirectory*: default log4j exit.
- *"/configuration"*: includes "*settings.xml*" and "*logs.xml*" configuration files of Magentix2 platform.



[3] For UNIX Systems the corresponding *"*.sh*" files are required

Run Magentix2 examples



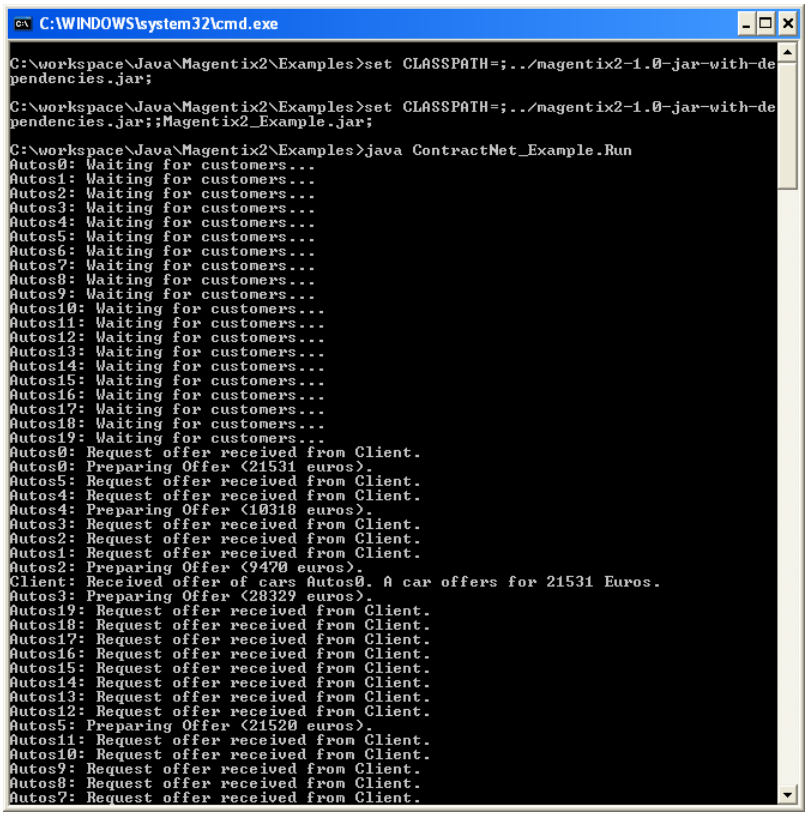
Required jars Classpath and run commad are included in *.bat files. For example:

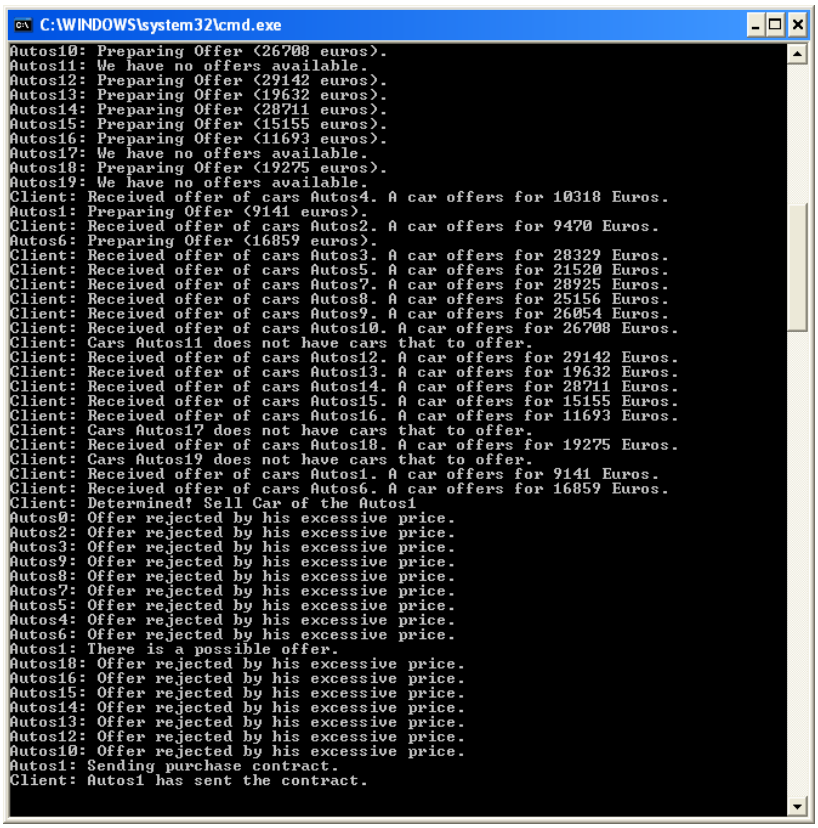
ContractNet_Example.bat

```
set CLASSPATH=%CLASSPATH%;../magentix2-1.0-jar-with-dependencies.jar;

set CLASSPATH=%CLASSPATH%;Magentix2_Example.jar;

java ContractNet_Example.Run
```


C:\WINDOWS\system32\cmd.exe
C:\workspace\Java\Magentix2\Examples>set CLASSPATH=;../magentix2-1.0-jar-with-dependencies.jar;
C:\workspace\Java\Magentix2\Examples>set CLASSPATH=;../magentix2-1.0-jar-with-dependencies.jar;;Magentix2_Example.jar;
C:\workspace\Java\Magentix2\Examples>java ContractNet_Example.Run
Autos0: Waiting for customers...
Autos1: Waiting for customers...
Autos2: Waiting for customers...
Autos3: Waiting for customers...
Autos4: Waiting for customers...
Autos5: Waiting for customers...
Autos6: Waiting for customers...
Autos7: Waiting for customers...
Autos8: Waiting for customers...
Autos9: Waiting for customers...
Autos10: Waiting for customers...
Autos11: Waiting for customers...
Autos12: Waiting for customers...
Autos13: Waiting for customers...
Autos14: Waiting for customers...
Autos15: Waiting for customers...
Autos16: Waiting for customers...
Autos17: Waiting for customers...
Autos18: Waiting for customers...
Autos19: Waiting for customers...
Autos0: Request offer received from Client.
Autos0: Preparing Offer (21531 euros).
Autos5: Request offer received from Client.
Autos4: Request offer received from Client.
Autos4: Preparing Offer (10318 euros).
Autos3: Request offer received from Client.
Autos2: Request offer received from Client.
Autos1: Request offer received from Client.
Autos2: Preparing Offer (9470 euros).
Client: Received offer of cars Autos0. A car offers for 21531 Euros.
Autos3: Preparing Offer (28329 euros).
Autos19: Request offer received from Client.
Autos18: Request offer received from Client.
Autos17: Request offer received from Client.
Autos16: Request offer received from Client.
Autos15: Request offer received from Client.
Autos14: Request offer received from Client.
Autos13: Request offer received from Client.
Autos12: Request offer received from Client.
Autos5: Preparing Offer (21520 euros).
Autos11: Request offer received from Client.
Autos10: Request offer received from Client.
Autos9: Request offer received from Client.
Autos8: Request offer received from Client.
Autos7: Request offer received from Client.

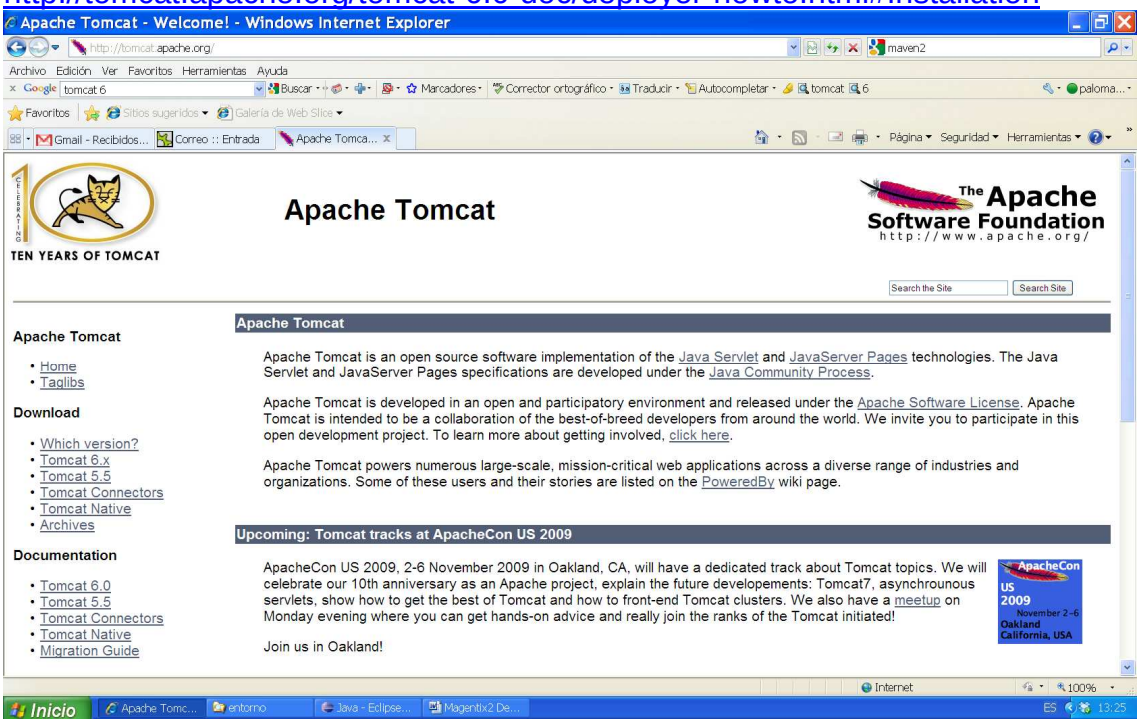

C:\WINDOWS\system32\cmd.exe
Autos10: Preparing Offer (26708 euros).
Autos11: We have no offers available.
Autos12: Preparing Offer (29142 euros).
Autos13: Preparing Offer (19632 euros).
Autos14: Preparing Offer (28711 euros).
Autos15: Preparing Offer (15155 euros).
Autos16: Preparing Offer (11693 euros).
Autos17: We have no offers available.
Autos18: Preparing Offer (19275 euros).
Autos19: We have no offers available.
Client: Received offer of cars Autos4. A car offers for 10318 Euros.
Autos1: Preparing Offer (9141 euros).
Client: Received offer of cars Autos2. A car offers for 9470 Euros.
Autos6: Preparing Offer (16859 euros).
Client: Received offer of cars Autos3. A car offers for 28329 Euros.
Client: Received offer of cars Autos5. A car offers for 21520 Euros.
Client: Received offer of cars Autos7. A car offers for 28925 Euros.
Client: Received offer of cars Autos8. A car offers for 25156 Euros.
Client: Received offer of cars Autos9. A car offers for 26054 Euros.
Client: Received offer of cars Autos10. A car offers for 26708 Euros.
Client: Cars Autos11 does not have cars that to offer.
Client: Received offer of cars Autos12. A car offers for 29142 Euros.
Client: Received offer of cars Autos13. A car offers for 19632 Euros.
Client: Received offer of cars Autos14. A car offers for 28711 Euros.
Client: Received offer of cars Autos15. A car offers for 15155 Euros.
Client: Received offer of cars Autos16. A car offers for 11693 Euros.
Client: Cars Autos17 does not have cars that to offer.
Client: Received offer of cars Autos18. A car offers for 19275 Euros.
Client: Cars Autos19 does not have cars that to offer.
Client: Received offer of cars Autos1. A car offers for 9141 Euros.
Client: Received offer of cars Autos6. A car offers for 16859 Euros.
Client: Determined! Sell Car of the Autos1
Autos0: Offer rejected by his excessive price.
Autos2: Offer rejected by his excessive price.
Autos3: Offer rejected by his excessive price.
Autos9: Offer rejected by his excessive price.
Autos8: Offer rejected by his excessive price.
Autos7: Offer rejected by his excessive price.
Autos5: Offer rejected by his excessive price.
Autos4: Offer rejected by his excessive price.
Autos6: Offer rejected by his excessive price.
Autos1: There is a possible offer.
Autos18: Offer rejected by his excessive price.
Autos16: Offer rejected by his excessive price.
Autos15: Offer rejected by his excessive price.
Autos14: Offer rejected by his excessive price.
Autos13: Offer rejected by his excessive price.
Autos12: Offer rejected by his excessive price.
Autos10: Offer rejected by his excessive price.
Autos1: Sending purchase contract.
Client: Autos1 has sent the contract.

Thomas Examples

Additional requisites

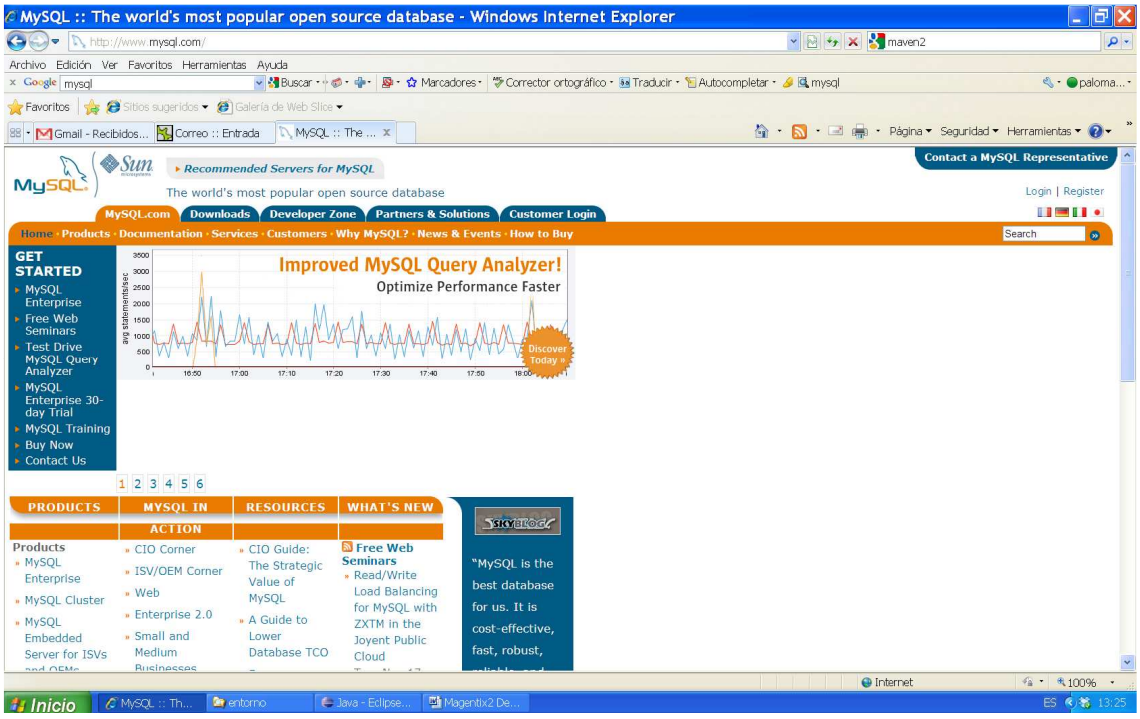
TOMCAT 6.0 (recommended server)

http://tomcat.apache.org/tomcat-6.0-doc/deployer-howto.html#Installation



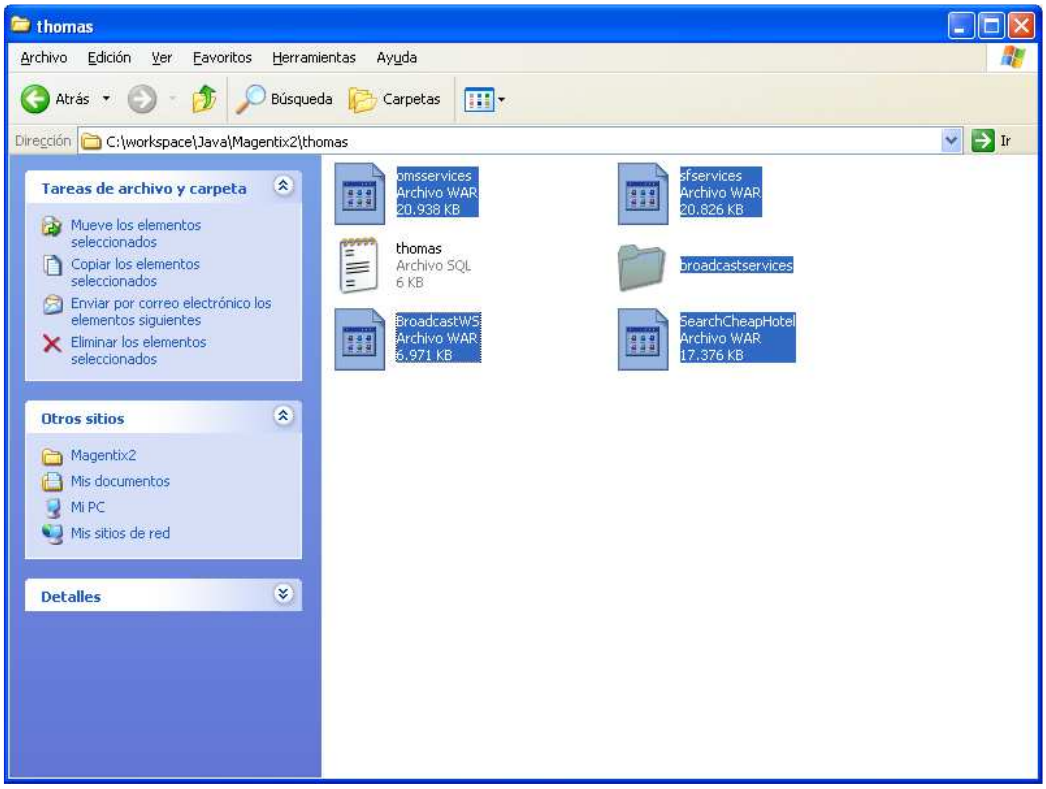
MySQL (recommended DDBB)

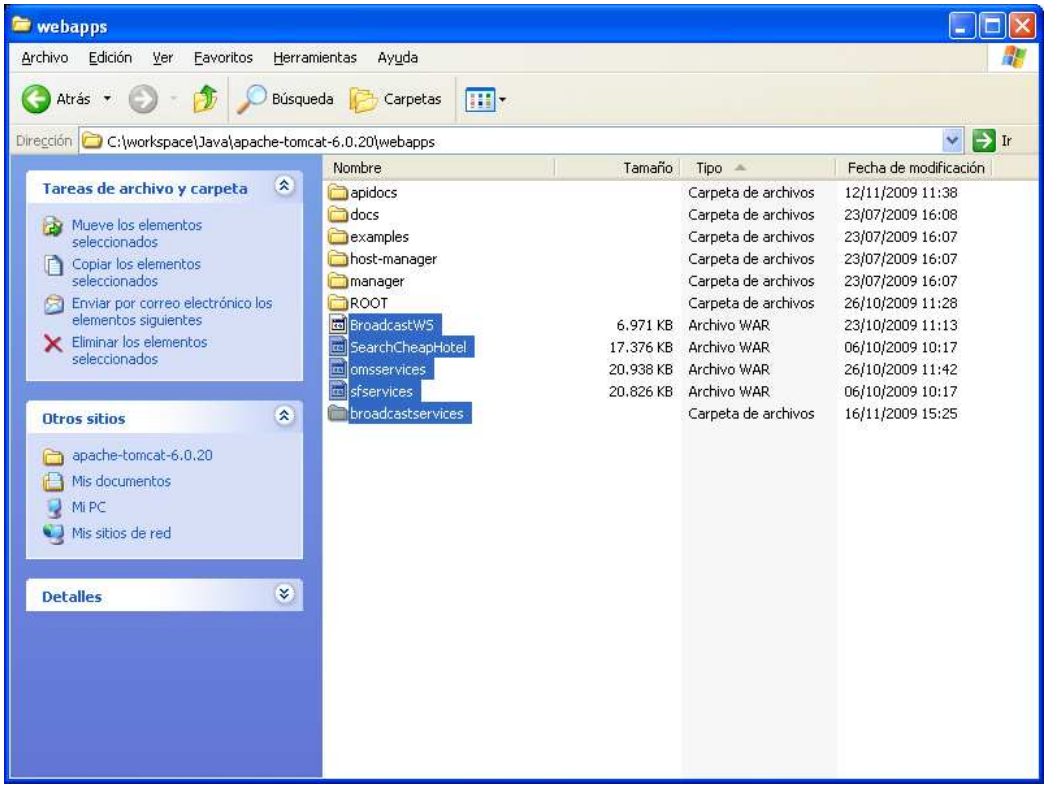
http://www.mysql.com/



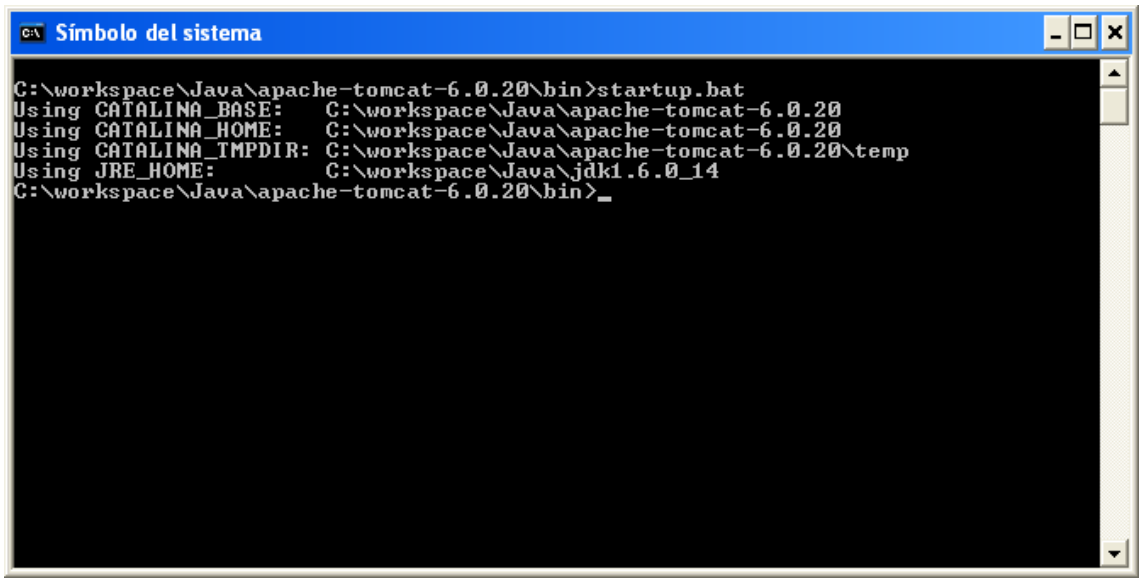
Deploy Thomas in TOMCAT

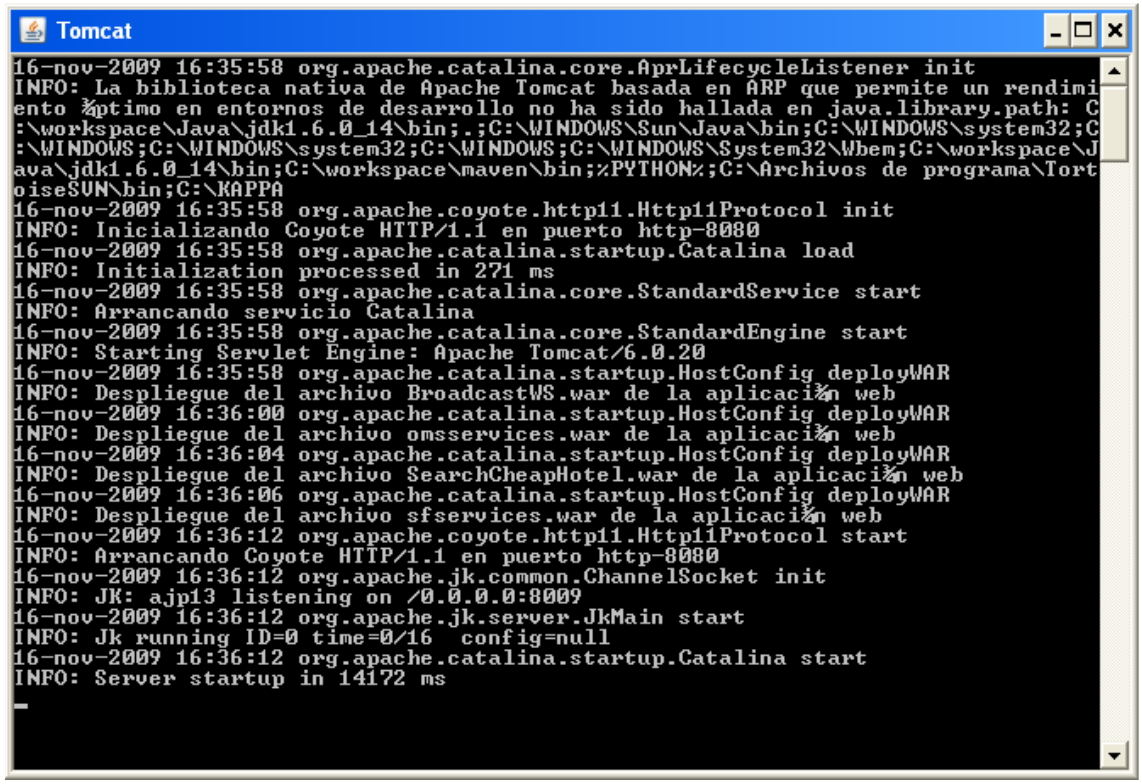
Copy packaged libraries of Thomas (*omsservices.war, sfservices.war*) and also the Thomas examples (*SearchCheapHotel.war, BroadcastWS.war and the broadcastservices directory*) from Magentix2 distribution to the subdirectory *webapss* of Tomcat:





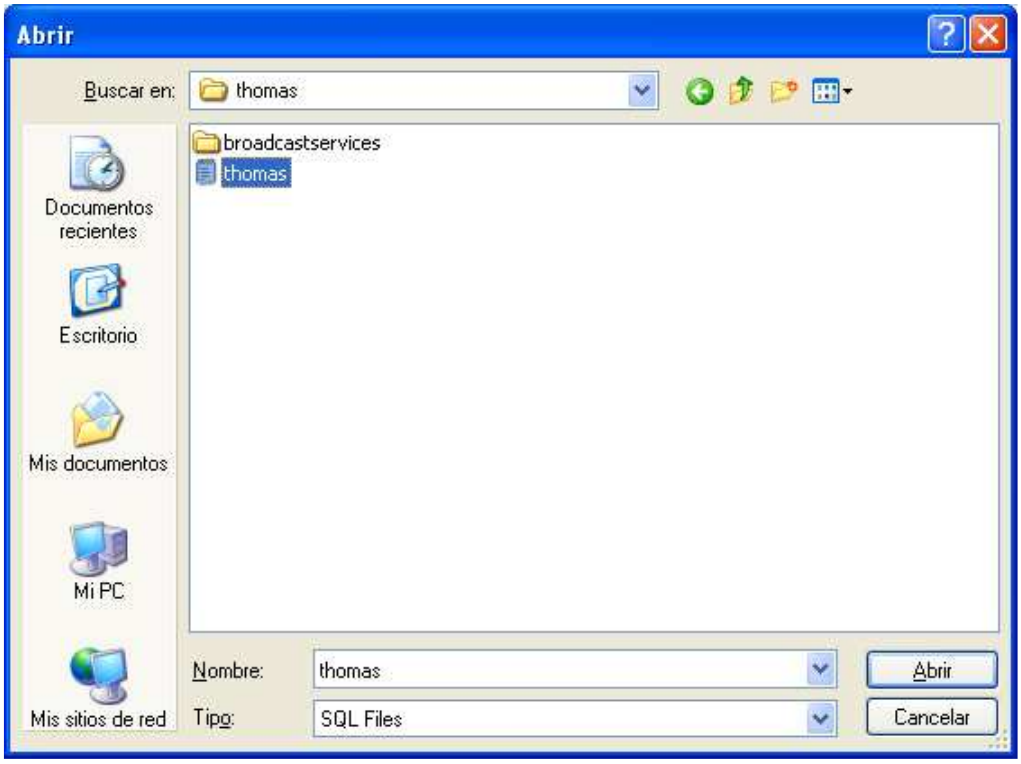
Start Tomcat by running the *startup.bat* file on the */bin* subdirectory of Tomcat. You must have the JDK 1.6 installed before you can start Tomcat.



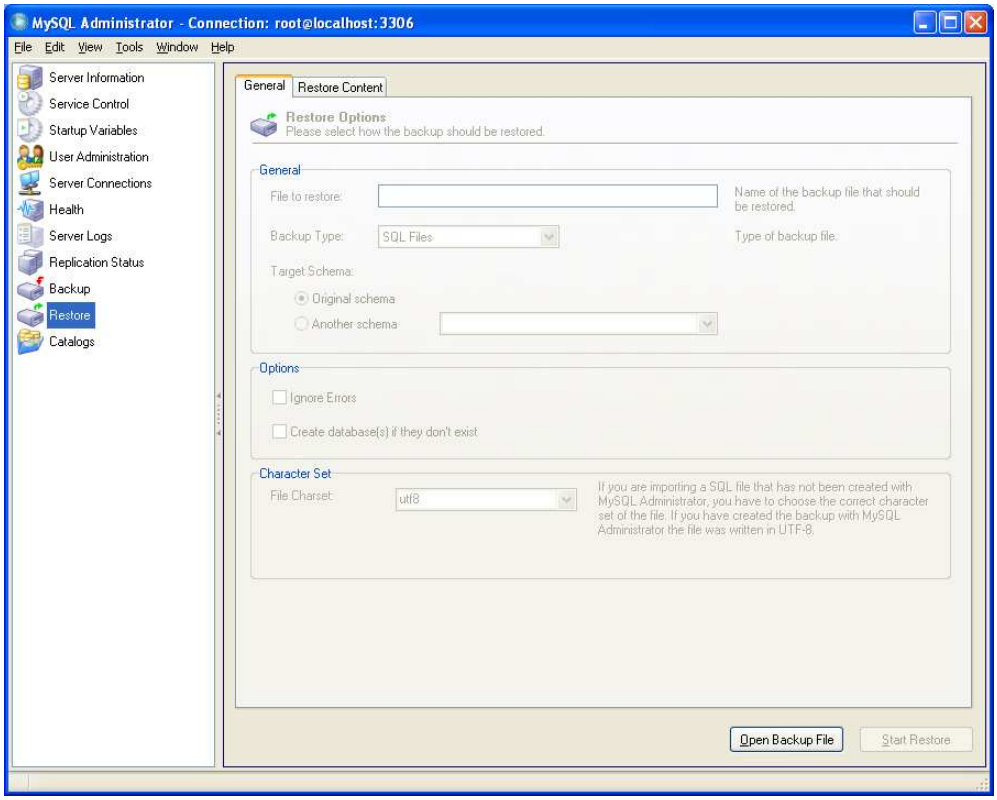


Install Thomas DDBB in MySQL

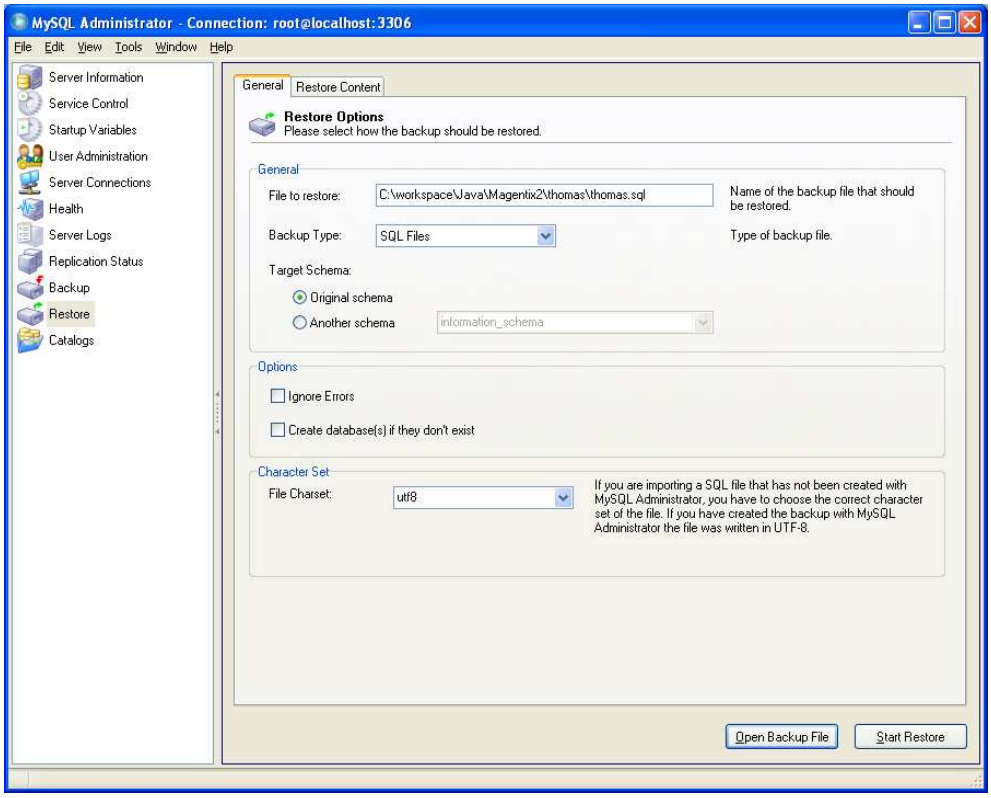
"*Thomas.sql*" file allow restore Thomas Data Base in MySQL:



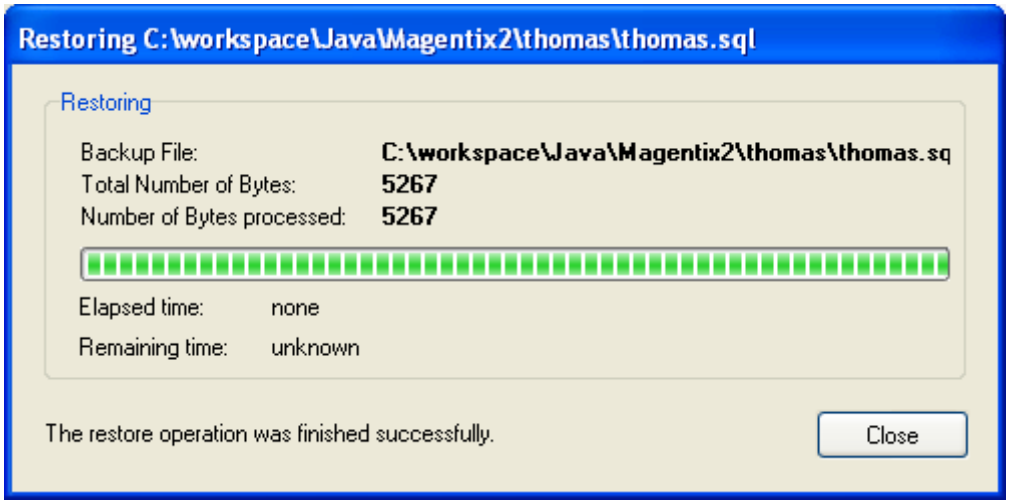
Open the MySQL Administrator and select *restore* option:



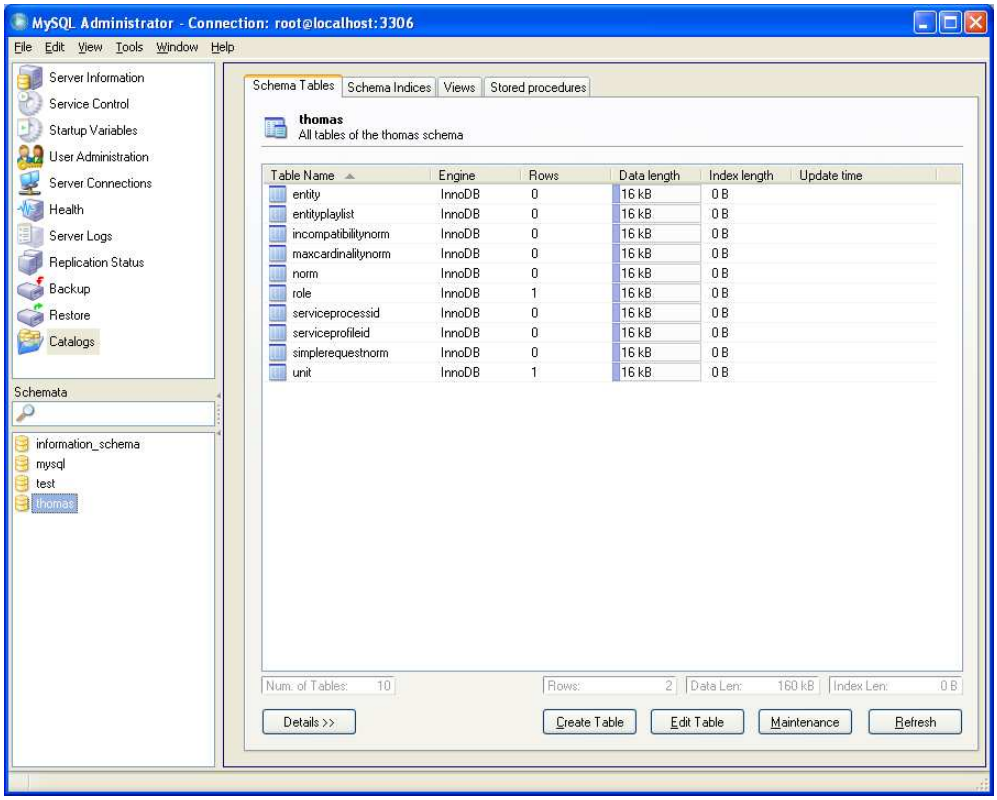
Select *Thomas.sql* file:



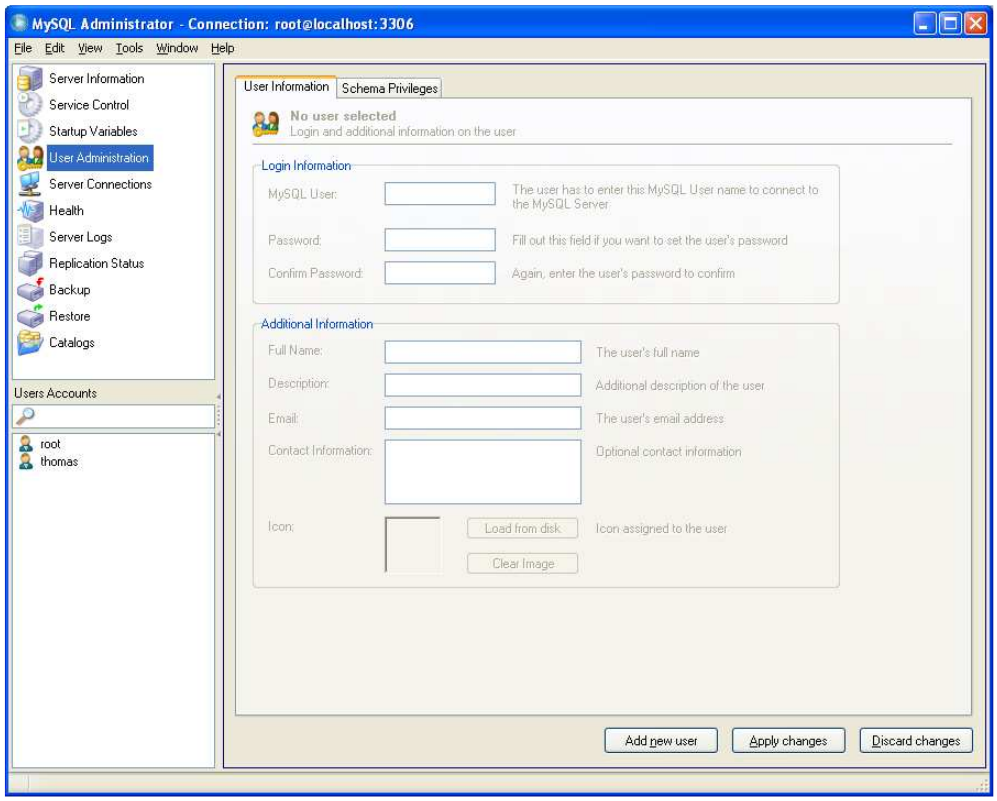
Init restore process:



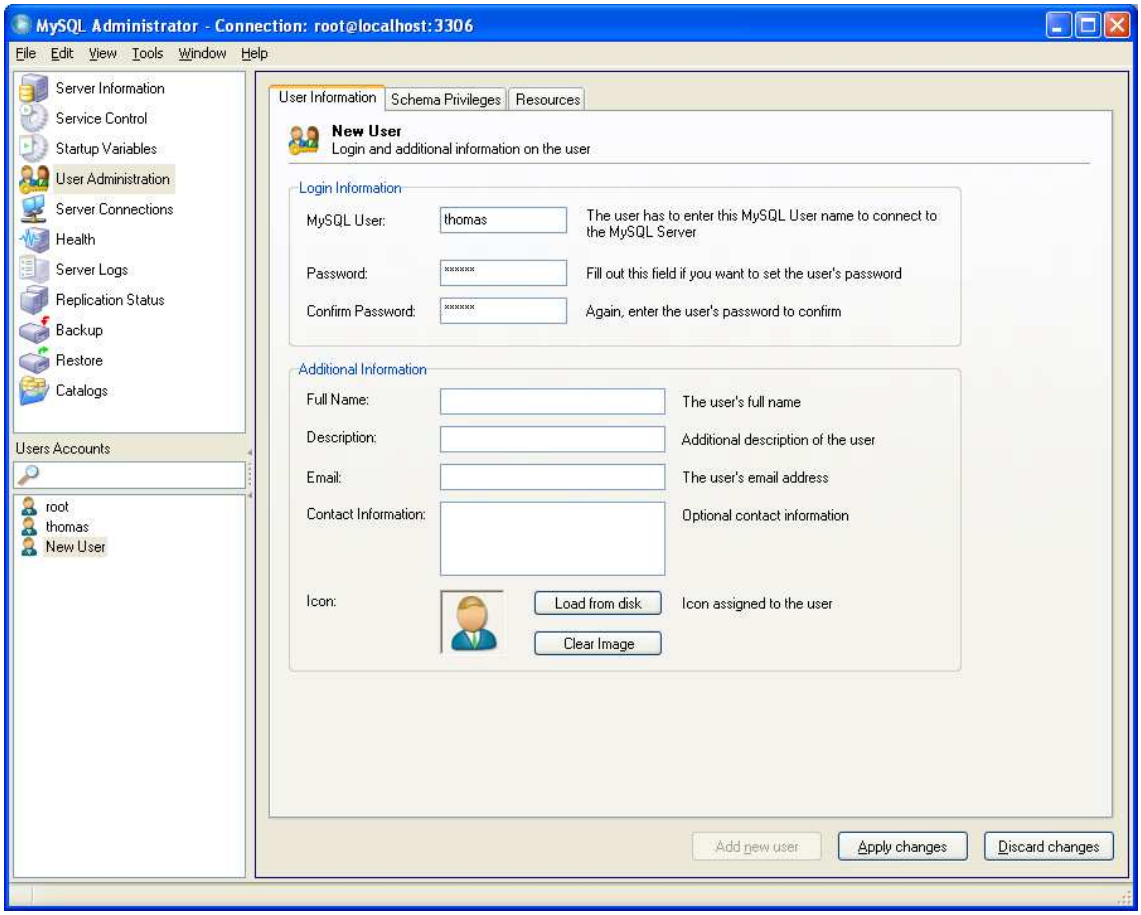
Now Thomas DDBB could be installed in MySQL



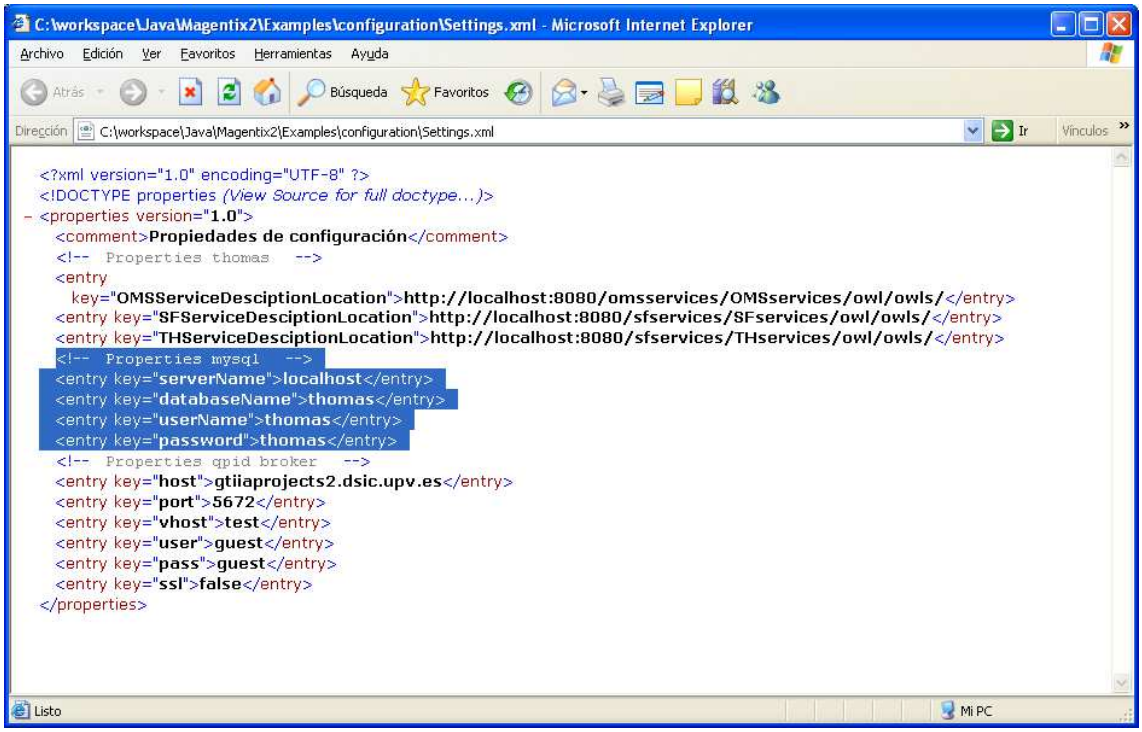
Add a new user to Thomas Schema:



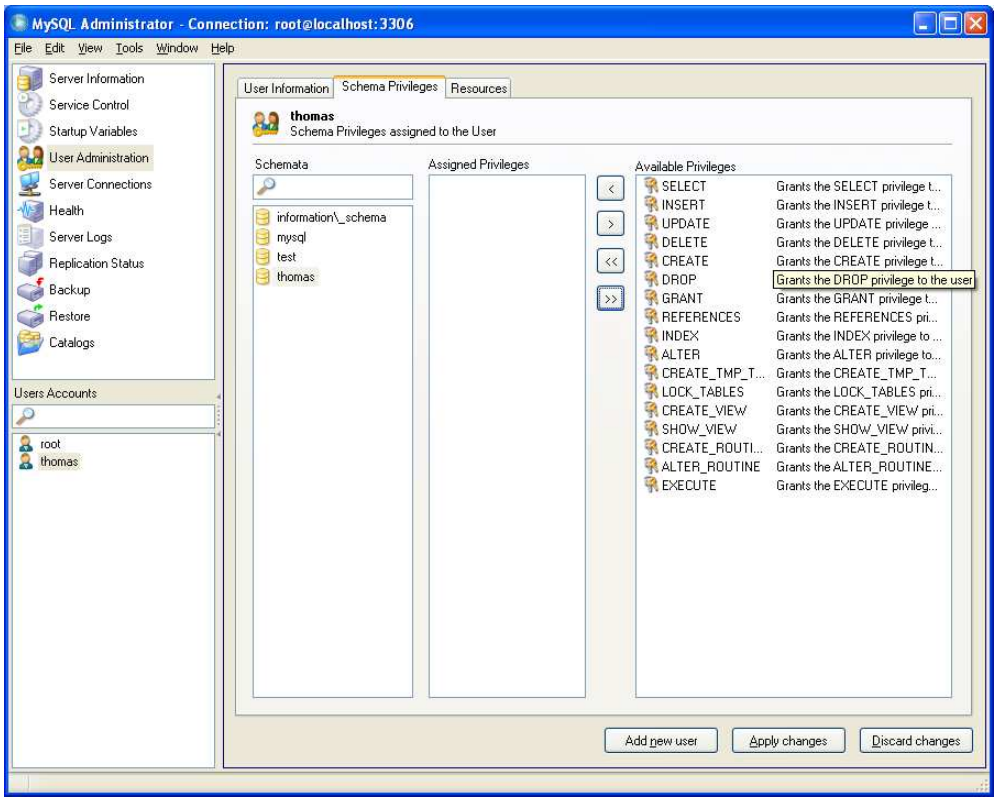
For example we add the user=”thomas” with password= “thomas”

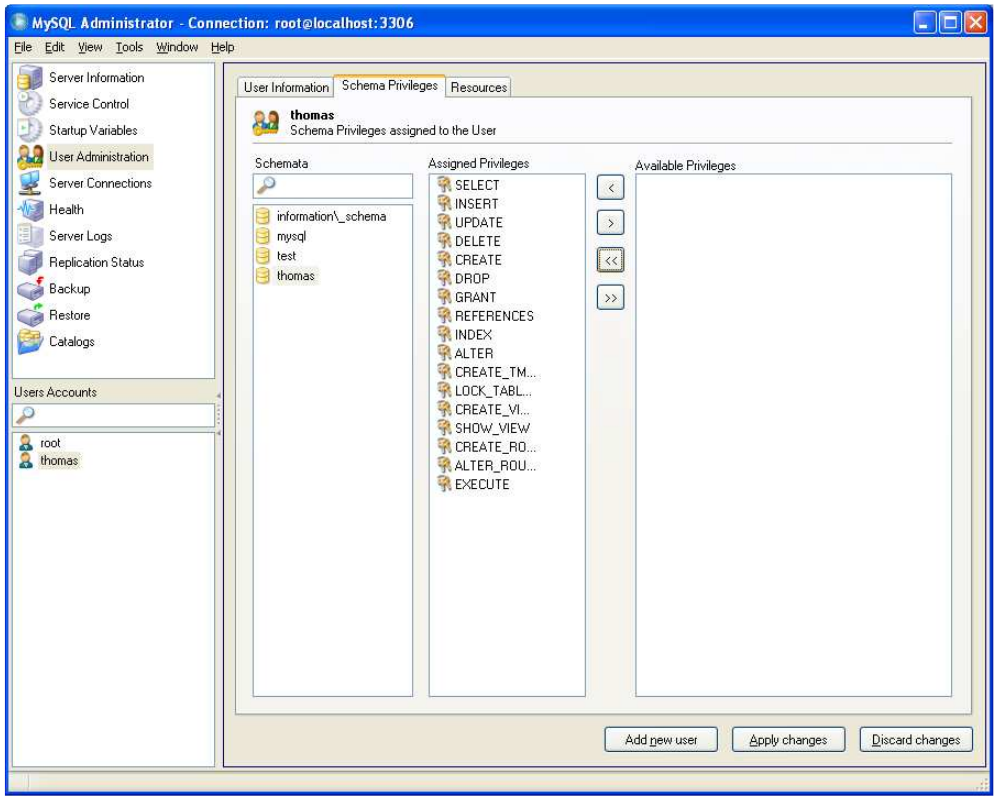
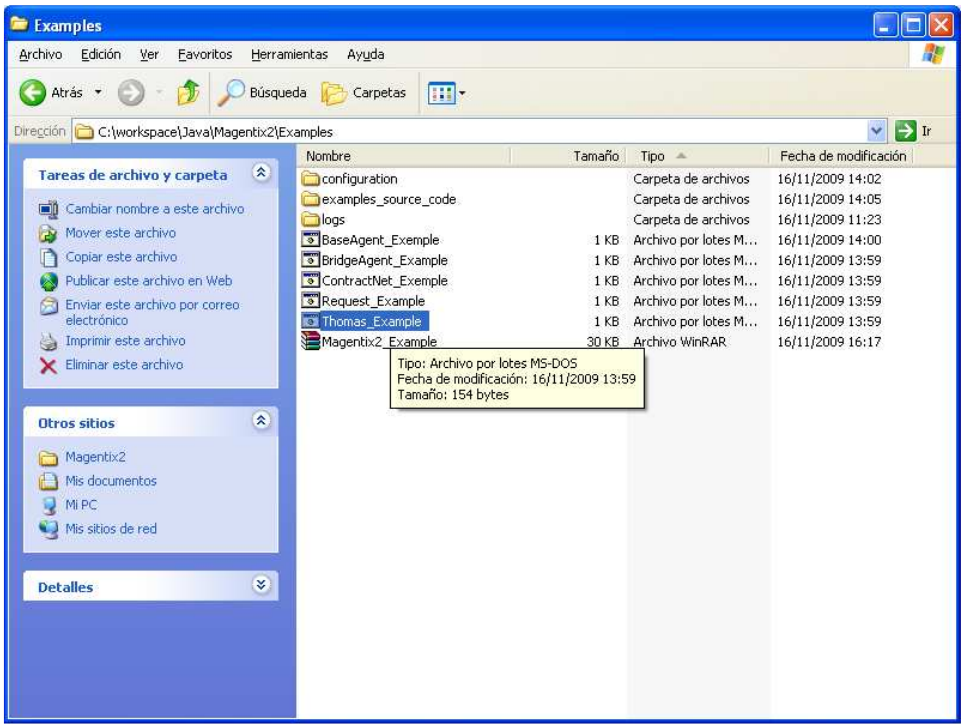
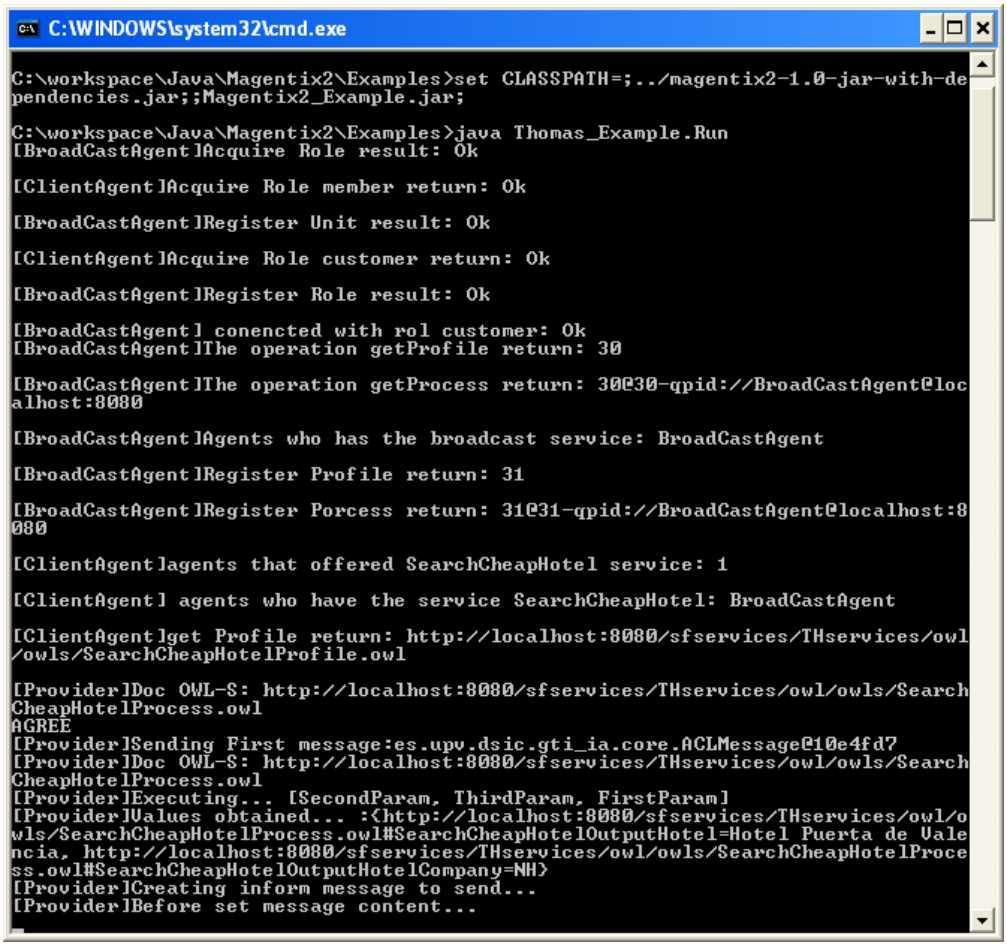


Also these values must be configured on the *settings.xml* file of Magentix2:



Finally available privileges for all tables of Thomas must be assigned to this user:





Run Thomas Example