

# Magentix2 Guide



# INDEX

Introduction.....	3
Install Qpid C++ Broker .....	4
Download QPID .....	4
Pre-Requisites .....	5
libboost-iostreams 1.35-dev .....	5
e2fsprogs 1.41.4 .....	6
pkg-config 022-1 .....	6
uuid 1.2-1.41.4 .....	7
ruby 4.2 .....	7
ruby 1.8 (required for install ruby 4.2) .....	8
Configure QPID .....	8
Modify Qpid Source Code (patch).....	9
Install Qpid.....	10
Start Qpid Broker .....	11
Magentix2.....	12
Download Magentix 2 .....	12
Unzip Magentix 2 .....	12
Pre-requisites.....	13
QPID (required).....	13
JDK 6 (required).....	13
ECLIPSE (recommended IDE).....	14
Configure Eclipse IDE .....	14
Basic Examples .....	16
Description .....	16
Content.....	17
Run Magentix2 examples (windows).....	18
Thomas .....	20
Architecture.....	20
Service Facilitator.....	20
Organization Manager Service.....	21
Additional requisites.....	22
TOMCAT 6.0 (recommended server) .....	22
MySQL (recommended DDBB) .....	22
Deploy Thomas in TOMCAT.....	23
Install Thomas DDBB in MySQL.....	25
Run Thomas Example .....	30
Developing Agents for Magentix2.....	31
Sender Agent.....	31
Consumer Agent.....	31
Client Agent .....	32
PROVIDER AGENT.....	33
Launching agents:.....	34
Initialization tasks:.....	34
Connecting to Qpid broker .....	34
Instantiating agents.....	34
APENDIX I Qpid Source Code modifications .....	35



# Introduction

In multi-agent system (MAS) field one of the goals is to build systems capable of making decisions in an autonomous and flexible way. Moreover, these systems must cooperate with other systems inside a "society". The term "society" needs to meet requirements such as: distribution, constant evolution, flexibility to allow members to enter or exit in the society, appropriate management of the organizational structure that defines the society, multi-device agent execution including devices with limited resources, and so on. On the other hand is necessary to build systems that enable complete interoperability for messaging middleware and also guarantee a very high efficiency. Additionally is necessary to provide developers methods, tools and appropriated architectures which support all the requirements in order to develop this kind of systems.

The main goal of the Magentix2 project is to develop a new open multi-agent system platform that is suitable for the development of systems with all these characteristics. The features of the Magentix2 platform can be structured in two basic levels, as follows:

- Low-level architecture: networking protocol and semantics required for messaging middleware services in Magentix2 are defined using the open standard *Advanced Message Queuing Protocol* (AMQP)<sup>1</sup> and specifically using *Apache Qpid*<sup>2</sup> Open Source implementation of this protocol. Magentix2 is FIPA-ACL compliant and is able to communicate with any platform that understands FIPA-ACL.
- High-level architecture: This architecture has been called THOMAS (MeTHods, Techniques and Tools for Open Multi-Agent Systems). The proposal tries to raise a total integration of two promising technologies, that is, multi-agent systems and service-oriented computing. In THOMAS architecture agents can offer and invoke services in a transparent way to other agents or entities, as well as external entities can interact with agents through the use of the offered services. This architecture is the first step in order to obtain true deployed virtual organizations.

Magentix2 is distributed under the terms of the LGPL (Lesser General Public license) <http://www.gnu.org/copyleft/lesser.html>

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

---

<sup>1</sup> Networking protocol and semantics required for messaging middleware services are defined using the open standard *Advanced Message Queuing Protocol* (AMQP).  
<http://jira.amqp.org/confluence/display/AMQP/Advanced+Message+Queuing+Protocol>

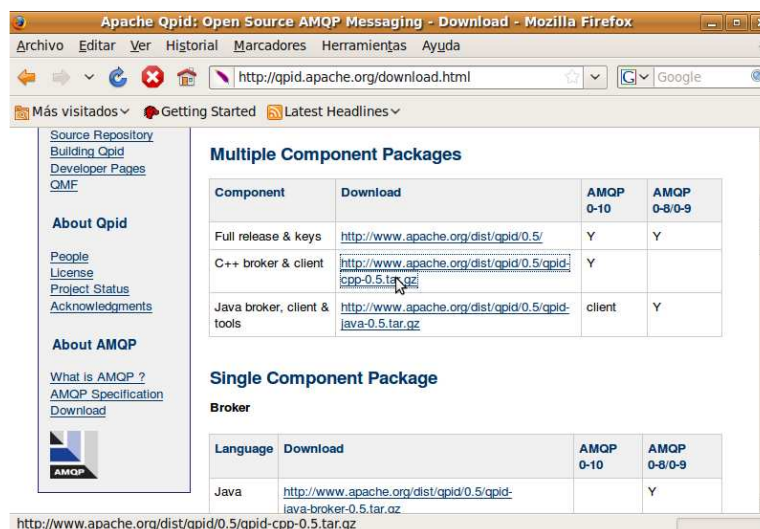
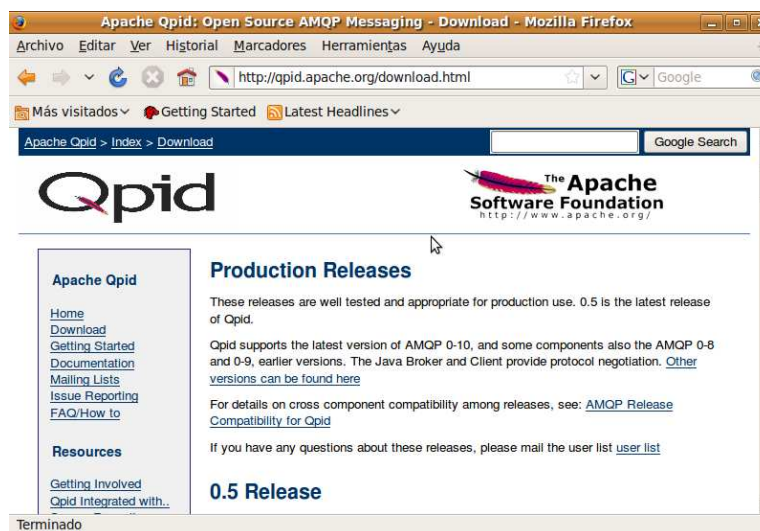
<sup>2</sup> *Apache Qpid* implements the latest AMQP specification and provides two AMQP messaging brokers implemented in C++ and Java and Client APIs in C++, Java, JMS, Ruby, Python, and C#.  
<http://qpid.apache.org/>

# Install Qpid C++ Broker

This section<sup>3</sup> describes how to install **Qpid/C++**<sup>4</sup> messaging broker from the source distribution <http://qpid.apache.org/> on Linux/Unix operating system, particularised to **Ubuntu 9.04 32-bit** version <http://www.ubuntu.com/>. In particular, this section explains how to install the some libraries (pre-requisites) required to install Qpid/C++ and also the modifications on the source code of Qpid/C++ required to compile and install the broker<sup>5</sup>.

## Download QPID

<http://qpid.apache.org/download.html>



<sup>3</sup> This section is based on the INSTALL file included on the Qpid distribution  
<sup>4</sup> Qpid has been built using the GNU C++ compiler: gcc <<http://gcc.gnu.org/>> (3.4.6).  
<sup>5</sup> A patch has been developed to facilitate these modifications.

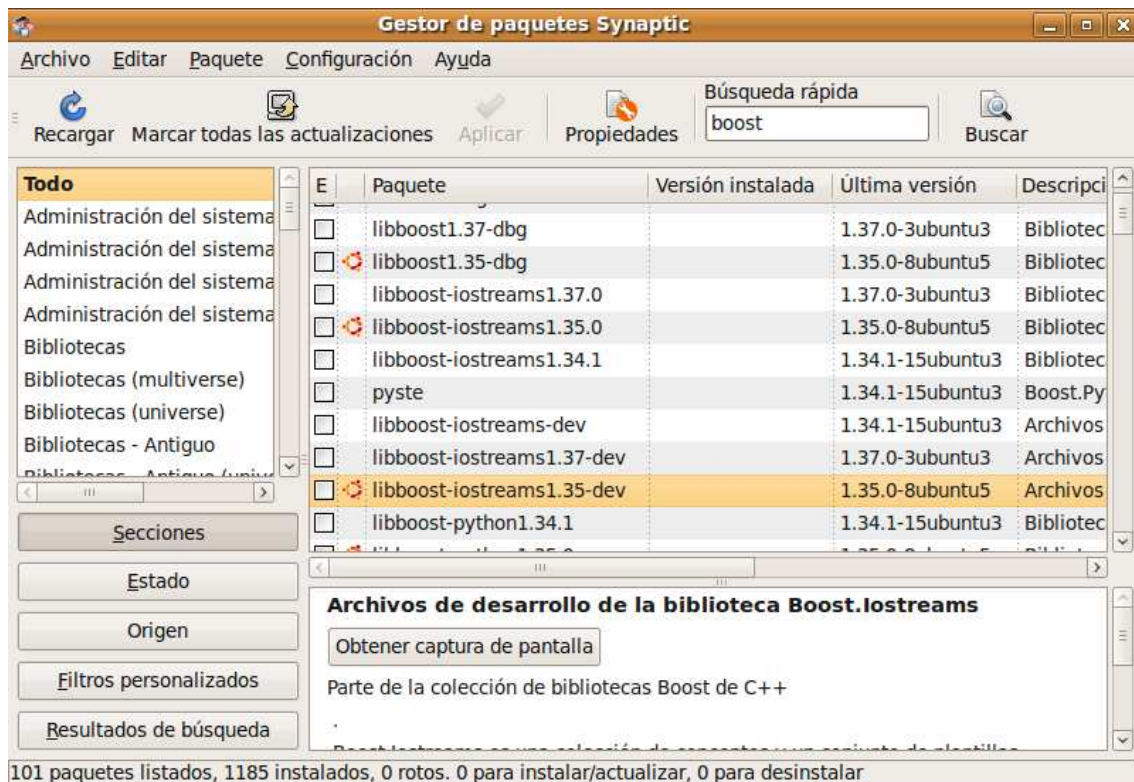
## Pre-Requisites

The following libraries must be installed to build a source distribution of Qpid:

- **libboost-iostreams 1.35-dev**: <<http://www.boost.org>> (1.35)<sup>6</sup>
- **e2fsprogs**: <<http://e2fsprogs.sourceforge.net/>> (1.39)
- **pkgconfig**: <<http://pkgconfig.freedesktop.org/wiki/>> (0.21)
- **uuid 1.2-1.41.4**
- **ruby 4.2**
- **ruby 1.8**

On *Ubuntu* operating system these packages can be installed using the distribution's package management tool *Synaptic*.

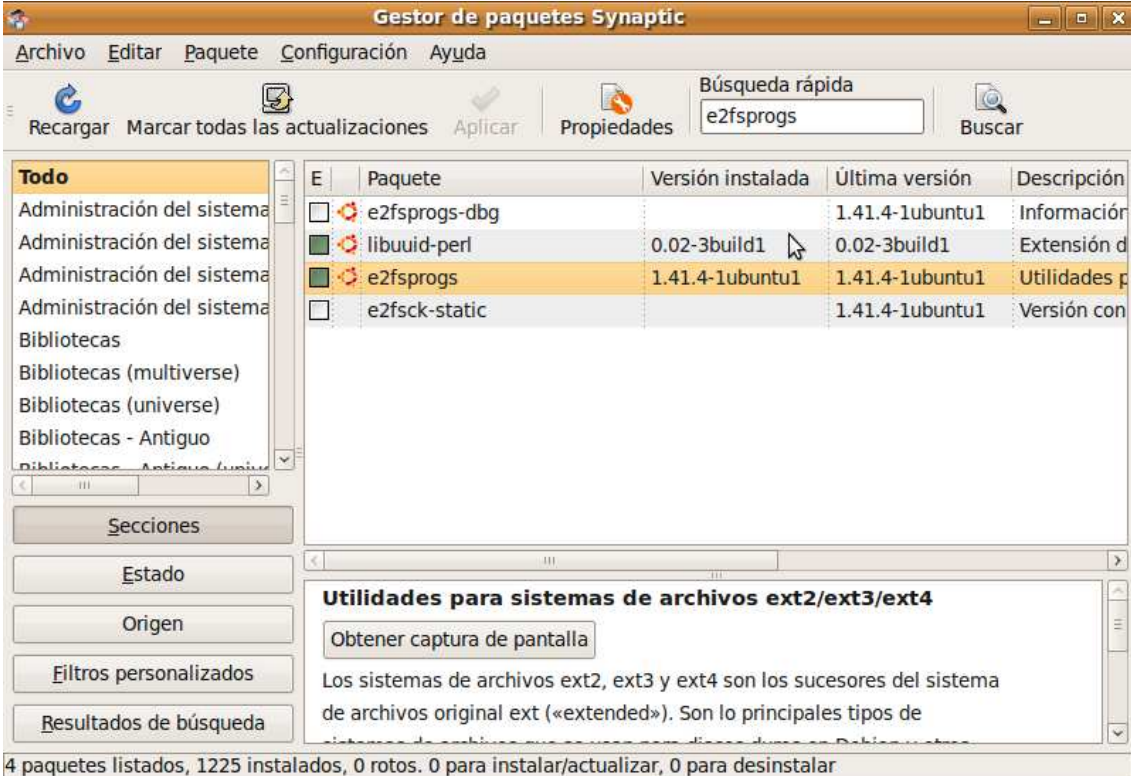
### libboost-iostreams 1.35-dev



<sup>6</sup> Earlier versions of boost e.g. 1.33 also work and there is a patch to get 1.32 working in the svn tree though that is only recommended as a last resort.



## e2fsprogs 1.41.4

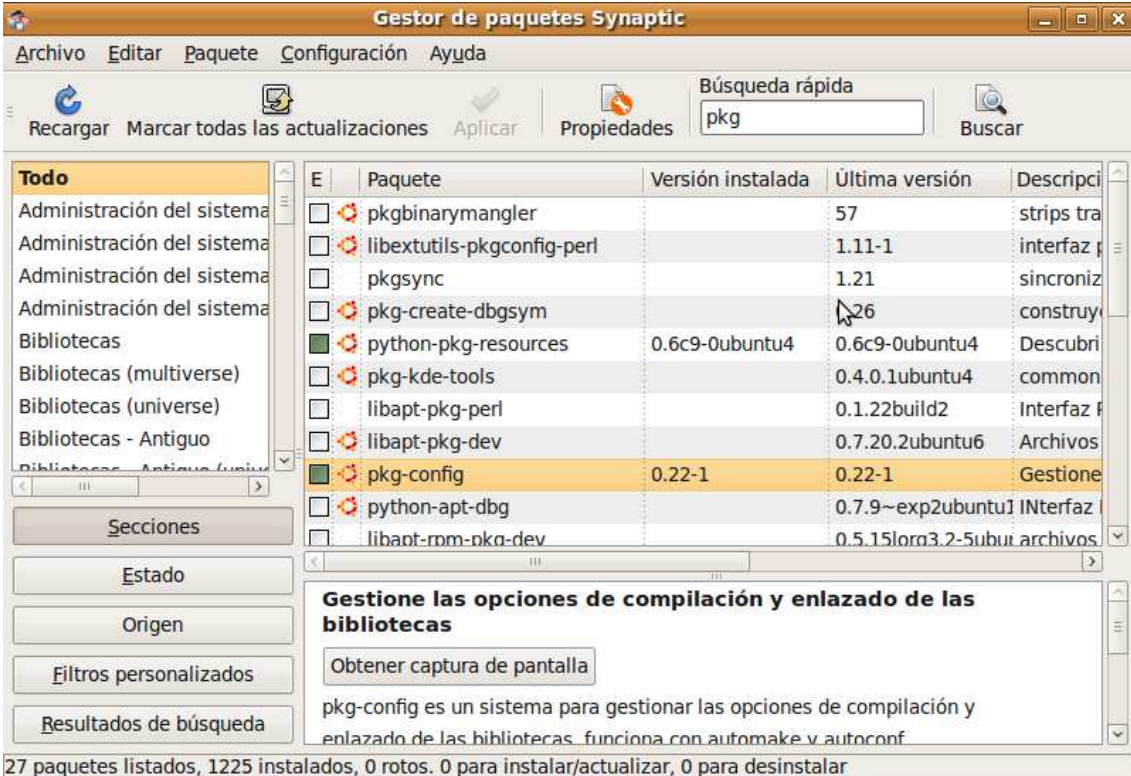


The screenshot shows the Synaptic Package Manager window. The search bar contains "e2fsprogs". The package list shows the following details for e2fsprogs:

E	Paquete	Versión instalada	Última versión	Descripción
<input type="checkbox"/>	e2fsprogs-dbg		1.41.4-1ubuntu1	Información
<input checked="" type="checkbox"/>	libuuid-perl	0.02-3build1	0.02-3build1	Extensión d
<input checked="" type="checkbox"/>	e2fsprogs	1.41.4-1ubuntu1	1.41.4-1ubuntu1	Utilidades p
<input type="checkbox"/>	e2fsck-static		1.41.4-1ubuntu1	Versión con

Below the package list, there is a description for "Utilidades para sistemas de archivos ext2/ext3/ext4". The status bar at the bottom indicates: "4 paquetes listados, 1225 instalados, 0 rotos. 0 para instalar/actualizar, 0 para desinstalar".

## pkg-config 0.22-1

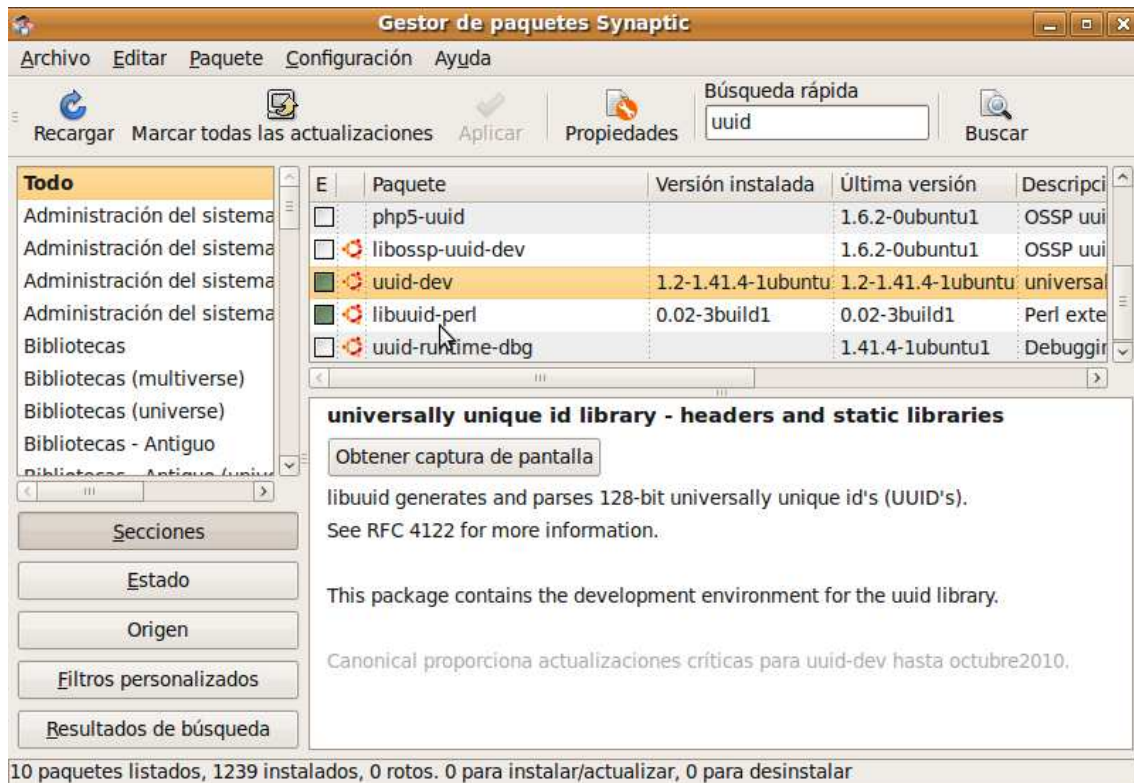


The screenshot shows the Synaptic Package Manager window. The search bar contains "pkg". The package list shows the following details for pkg-config:

E	Paquete	Versión instalada	Última versión	Descripción
<input type="checkbox"/>	pkgbinarymangler		57	strips tra
<input type="checkbox"/>	libextutils-pkgconfig-perl		1.11-1	interfaz p
<input type="checkbox"/>	pkgsync		1.21	sincroniz
<input type="checkbox"/>	pkg-create-dbgsym		26	construy
<input checked="" type="checkbox"/>	python-pkg-resources	0.6c9-0ubuntu4	0.6c9-0ubuntu4	Descubri
<input type="checkbox"/>	pkg-kde-tools		0.4.0.1ubuntu4	common
<input type="checkbox"/>	libapt-pkg-perl		0.1.22build2	Interfaz f
<input type="checkbox"/>	libapt-pkg-dev		0.7.20.2ubuntu6	Archivos
<input checked="" type="checkbox"/>	pkg-config	0.22-1	0.22-1	Gestione
<input type="checkbox"/>	python-apt-dbg		0.7.9~exp2ubuntu1	Interfaz l
<input type="checkbox"/>	libapt-rom-pkg-dev		0.5.15lora3.2-5ubu	archivos

Below the package list, there is a description for "Gestione las opciones de compilación y enlazado de las bibliotecas". The status bar at the bottom indicates: "27 paquetes listados, 1225 instalados, 0 rotos. 0 para instalar/actualizar, 0 para desinstalar".

## uuid 1.2-1.41.4



The screenshot shows the Synaptic Package Manager window. The search bar contains 'uuid'. The package list shows 'uuid-dev' selected. The details pane shows the package description for 'universally unique id library - headers and static libraries'.

E	Paquete	Versión instalada	Última versión	Descripción
<input type="checkbox"/>	php5-uuid		1.6.2-0ubuntu1	OSSP uui
<input type="checkbox"/>	libossp-uuid-dev		1.6.2-0ubuntu1	OSSP uui
<input checked="" type="checkbox"/>	uuid-dev	1.2-1.41.4-1ubuntu1	1.2-1.41.4-1ubuntu1	universal
<input checked="" type="checkbox"/>	libuuid-perl	0.02-3build1	0.02-3build1	Perl exte
<input type="checkbox"/>	uuid-runtime-dbg		1.41.4-1ubuntu1	Debuggir

**universally unique id library - headers and static libraries**

Obtener captura de pantalla

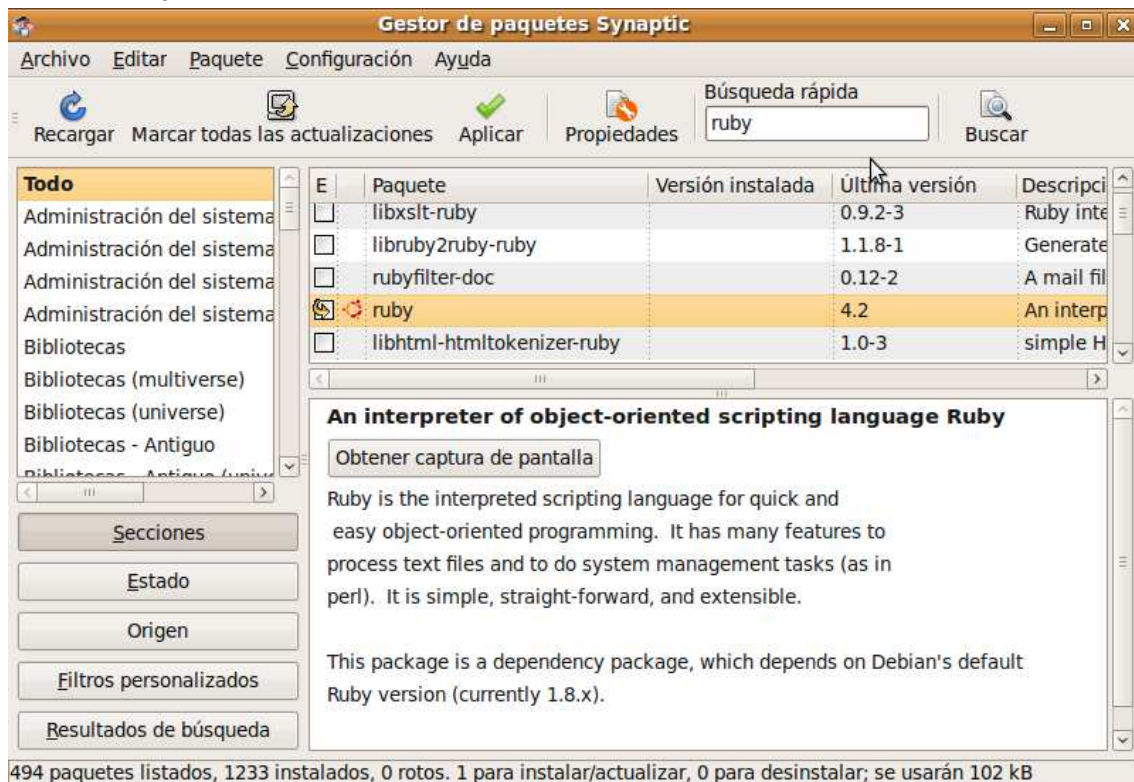
libuuid generates and parses 128-bit universally unique id's (UUID's).  
See RFC 4122 for more information.

This package contains the development environment for the uuid library.

Canonical proporciona actualizaciones críticas para uuid-dev hasta octubre2010.

10 paquetes listados, 1239 instalados, 0 rotos. 0 para instalar/actualizar, 0 para desinstalar

## ruby 4.2



The screenshot shows the Synaptic Package Manager window. The search bar contains 'ruby'. The package list shows 'ruby' selected. The details pane shows the package description for 'An interpreter of object-oriented scripting language Ruby'.

E	Paquete	Versión instalada	Última versión	Descripción
<input type="checkbox"/>	libxslt-ruby		0.9.2-3	Ruby inte
<input type="checkbox"/>	libruby2ruby-ruby		1.1.8-1	Generate
<input type="checkbox"/>	rubyfilter-doc		0.12-2	A mail fil
<input checked="" type="checkbox"/>	ruby		4.2	An interp
<input type="checkbox"/>	libhtml-htmltokenizer-ruby		1.0-3	simple H

**An interpreter of object-oriented scripting language Ruby**

Obtener captura de pantalla

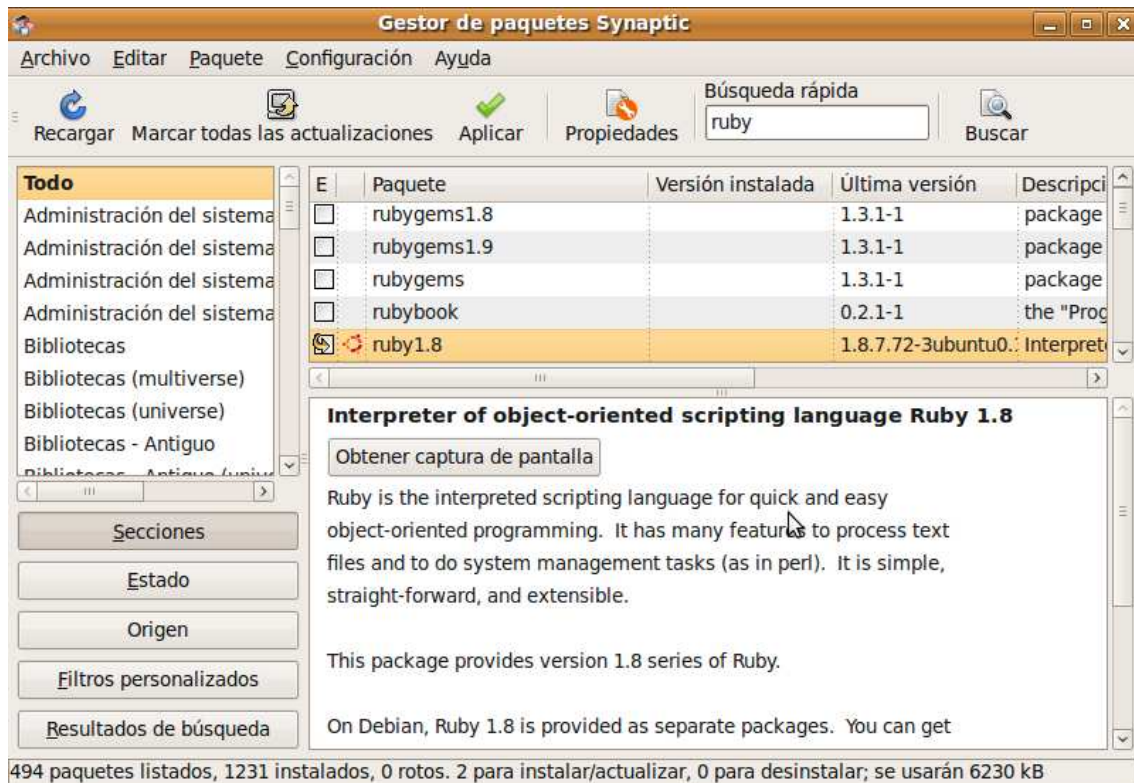
Ruby is the interpreted scripting language for quick and easy object-oriented programming. It has many features to process text files and to do system management tasks (as in perl). It is simple, straight-forward, and extensible.

This package is a dependency package, which depends on Debian's default Ruby version (currently 1.8.x).

494 paquetes listados, 1233 instalados, 0 rotos. 1 para instalar/actualizar, 0 para desinstalar; se usarán 102 kB



## ruby 1.8 (required for install ruby 4.2)



## Configure QPID

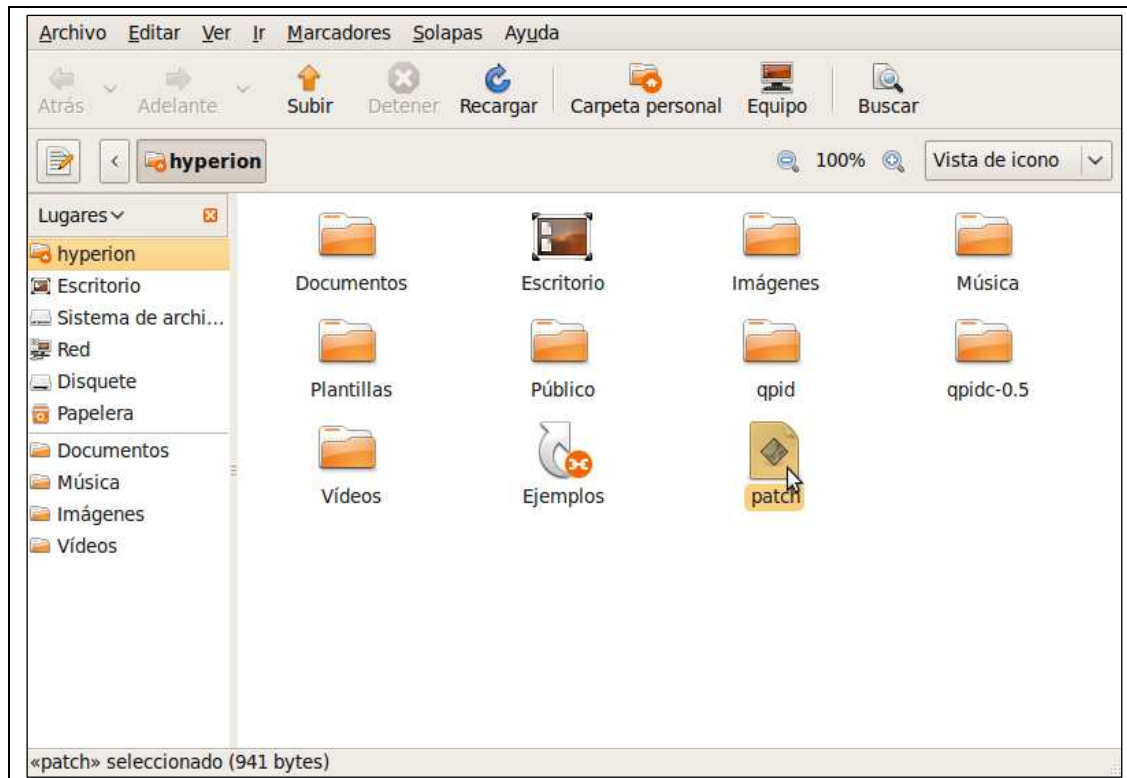
`./configure --prefix=/home/hyperion/qpid`

```
Archivo Editar Ver Terminal Ayuda
hyperion@hyperion:~/qpidc-0.5$ ./configure --prefix=/home/hyperion/qpid
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking for style of include used by make... GNU
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking dependency style of gcc... gcc3
checking for gcc option to accept ISO C99... -std=gnu99
checking for gcc -std=gnu99 option to accept ISO Standard C... (cached) -std=gnu99
checking whether gcc -std=gnu99 and cc understand -c and -o together... yes
checking for g++... no
```

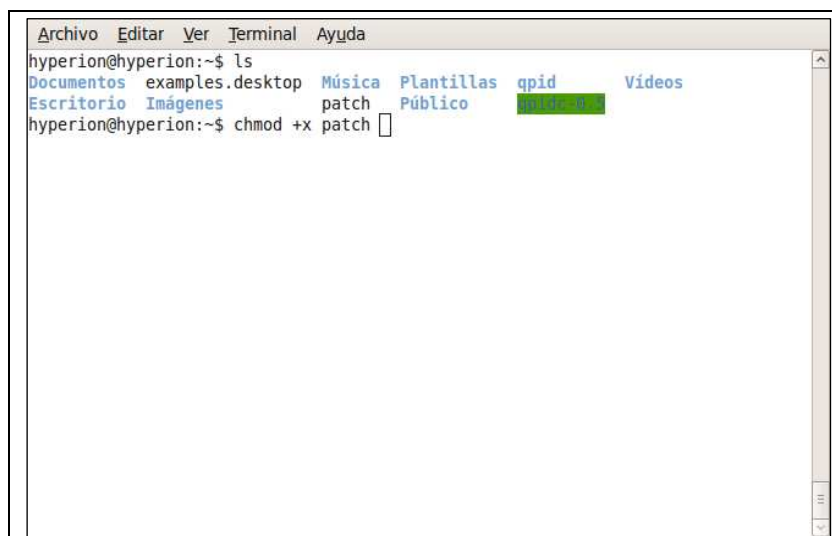


## Modify Qpid Source Code (patch)

In order to compile and install the Qpid C++ broker in Ubuntu 9.04 some modifications of the source code are required. A detailed explanation of these modifications is included in the *Appendix I* of this document. A patch has been developed in order to facilitate this process and is included in the Magentix2 distribution.

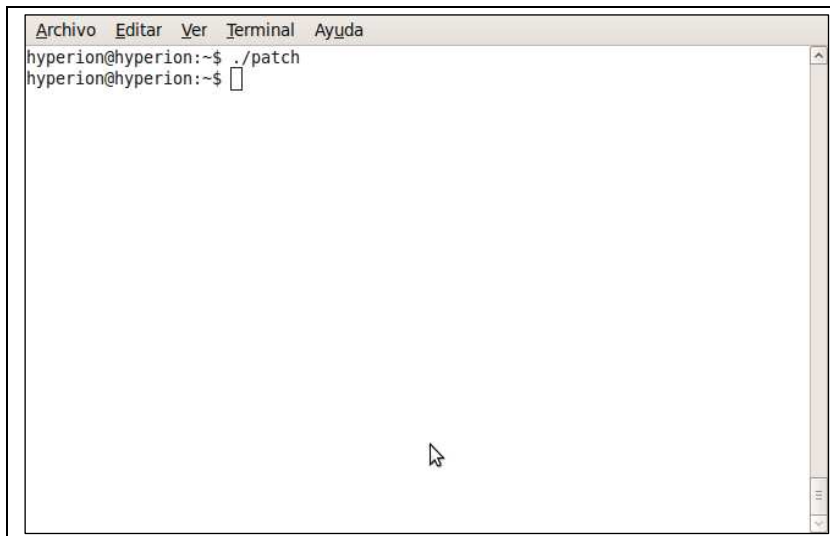


Changes the permission of the PATCH file to run it as a program:  
`chmod +x patch`



Run the patch file:

*`./patch`*



```
Archivo  Editor  Ver  Terminal  Ayuda
hyperion@hyperion:~$ ./patch
hyperion@hyperion:~$
```

A terminal window with a menu bar containing 'Archivo', 'Editor', 'Ver', 'Terminal', and 'Ayuda'. The command prompt shows 'hyperion@hyperion:~\$' followed by the command './patch' and a new prompt 'hyperion@hyperion:~\$' with a cursor.

## ***Install Qpid***

*Make install*

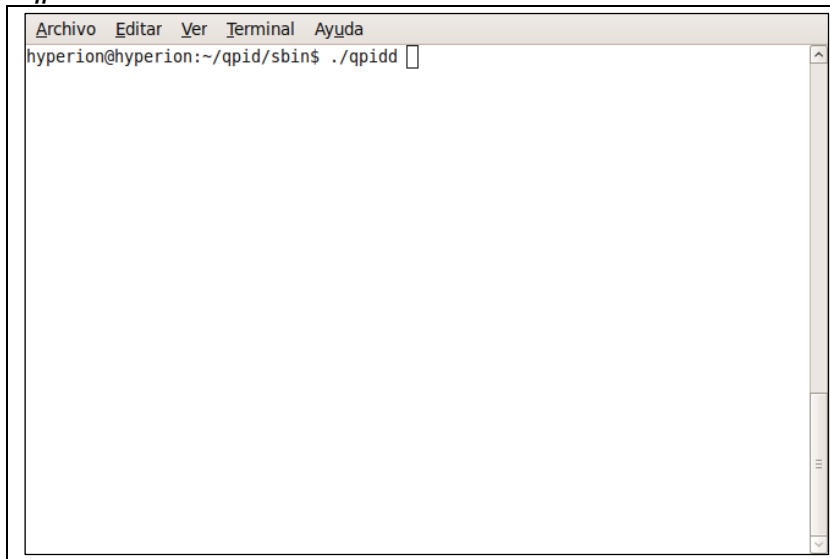


```
Archivo  Editor  Ver  Terminal  Ayuda
hyperion@hyperion:~/qpidc-0.5$ make install
```

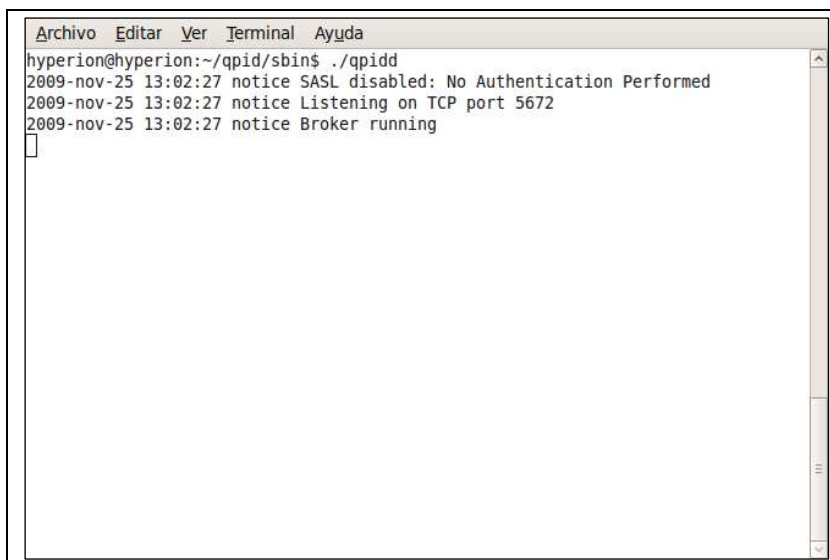
A terminal window with a menu bar containing 'Archivo', 'Editor', 'Ver', 'Terminal', and 'Ayuda'. The command prompt shows 'hyperion@hyperion:~/qpidc-0.5\$' followed by the command 'make install' and a cursor.

## Start Qpid Broker

`./qpidd`



```
Archivo  Editar  Ver  Terminal  Ayuda
hyperion@hyperion:~/qpid/sbin$ ./qpidd
```

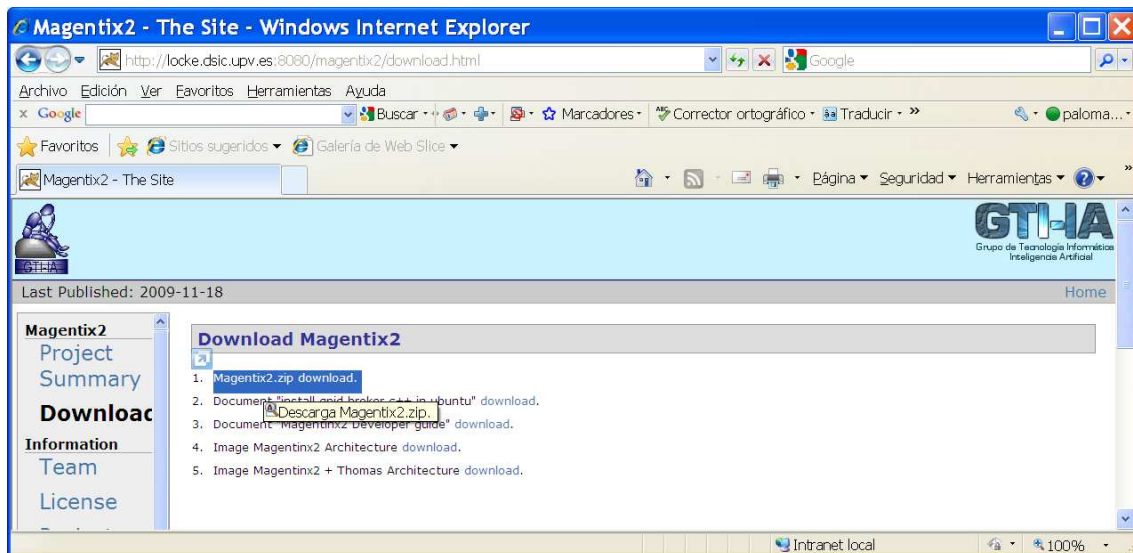


```
Archivo  Editar  Ver  Terminal  Ayuda
hyperion@hyperion:~/qpid/sbin$ ./qpidd
2009-nov-25 13:02:27 notice SASL disabled: No Authentication Performed
2009-nov-25 13:02:27 notice Listening on TCP port 5672
2009-nov-25 13:02:27 notice Broker running
```

# Magentix2

## Download Magentix 2

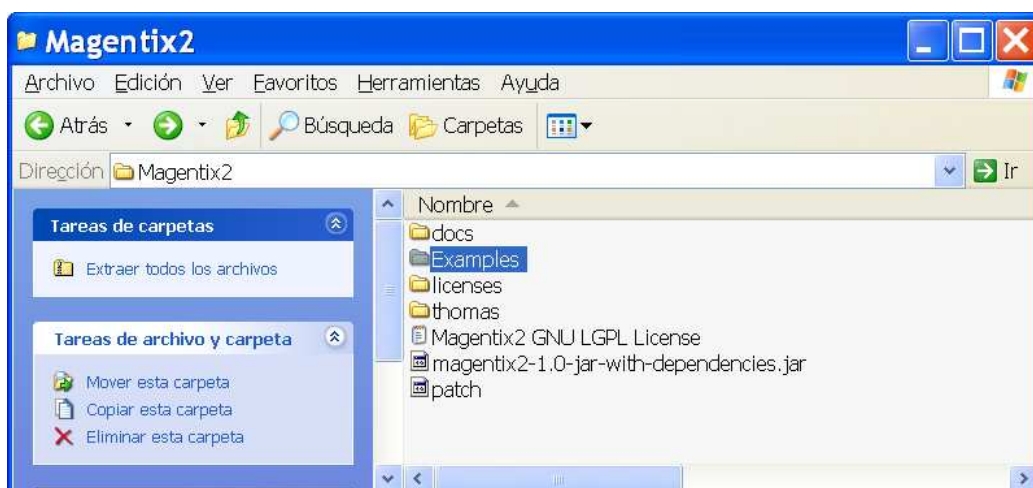
<http://locke.dsic.upv.es:8080/magentix2/>



## Unzip Magentix 2

Unzipped Magentix2 directory includes:

- **Magentix2-1.0-jar-with-dependencies.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 *login.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.
- **Patch file**: executable that corrects Qpid source code problems.





## Pre-requisites

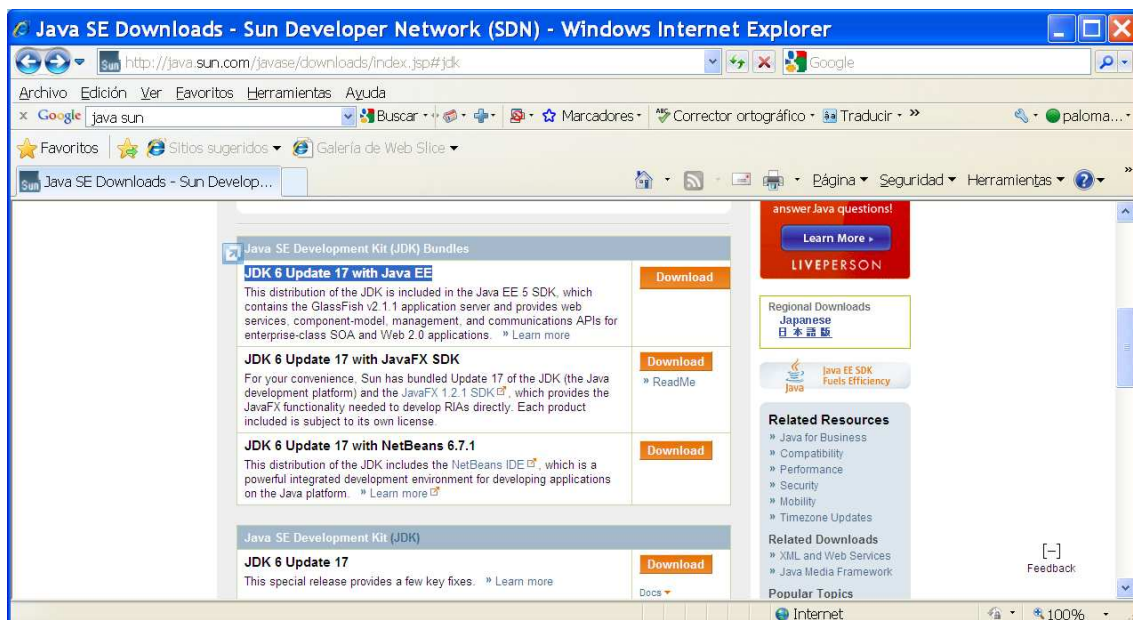
### QPID (required)

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**<sup>7</sup>. Other Broker configuration parameters as “**virtualhost**”<sup>8</sup>, “**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>
```

### JDK 6 (required)

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

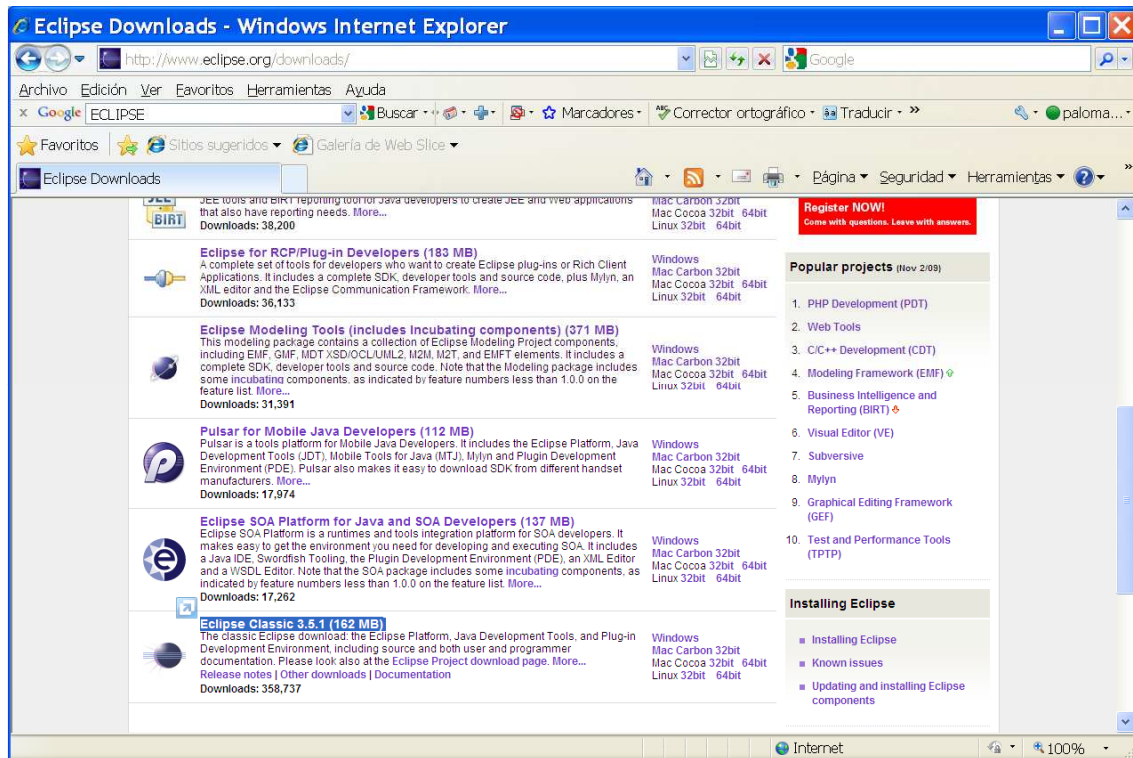


<sup>7</sup> The QPID C++ broker is pre-configured by default to listen messages on port 5672 but also it is possible to configure to listen on other ports (<http://qpid.apache.org/rasc.html>).

<sup>8</sup> 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.

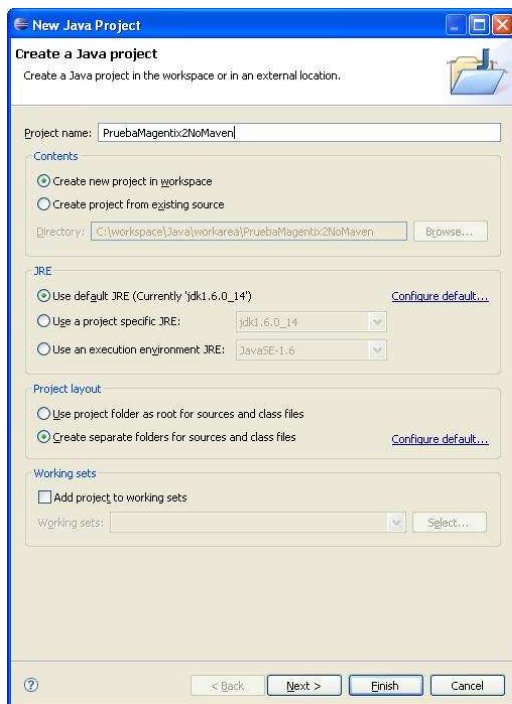
## ECLIPSE (recommended IDE)

<http://www.eclipse.org/>

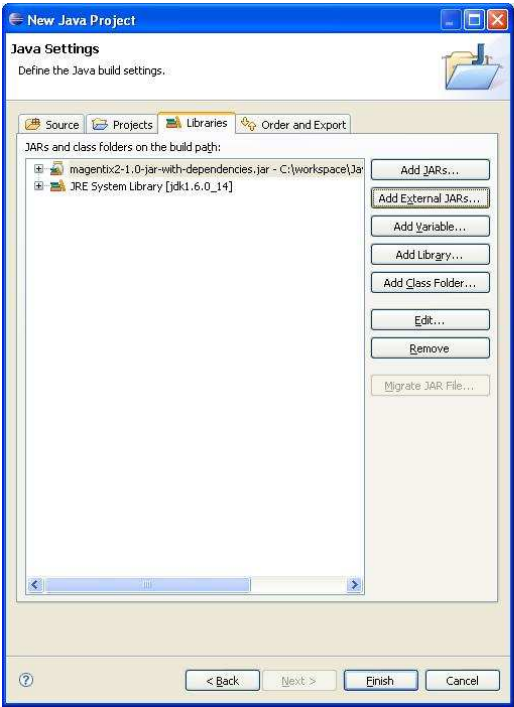


## Configure Eclipse IDE

Create a New Java project:



# Add Magentix2.jar dependence



## ***Basic Examples***

### **Description**

Subdirectory “*examples*” of Magentix2 distribution includes some basic examples of agents developed for this platform:

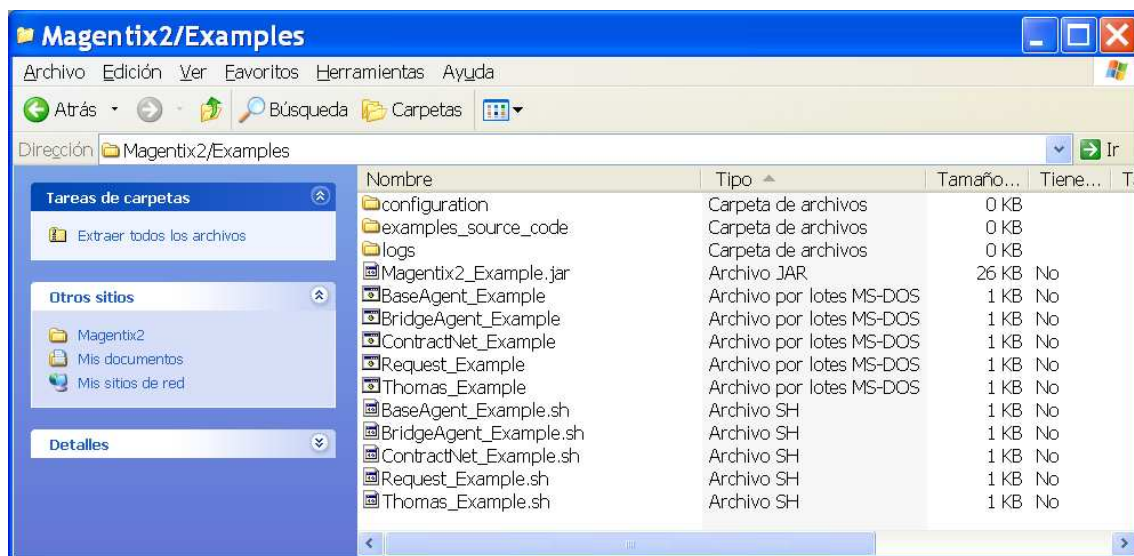
- *BaseAgent*: example of issuer/consumer messages. The sender agent builds and sends an ACLMessage to the consumer. When the ACLMessage arrives, the consumer displays the message on screen.
- *BridgeAgent*: example in which an internal agent send a message to an external agent of our platform.
- *Request*: example of an agent that implements the FIPA Request protocol. In this example, create two types of agents, one with the role responding to simulate a hospital attending emergency calls and the other a witness to an accident. When the witness see an accident sends a message of support to the hospital, hospital staff will tell whether the accident is within its area of action and where it can be the result of the operation.
- *ContractNet*: example of an agent that implements the FIPA ContractNet protocol. In this example, create two types of agents, one with the role responding to pretend to be a dealer and another with the role iniciador to pretend to be a buyer, the initiator send a purchase request to each dealer, the dealer will return your bid and wait for the initiator decides, the client will choose one and this will return the contract, the rest are sent a rejection
- *Thomas*: example of an agent that access to Thomas SF and OMS web services. In this example development of a new agent client and a provider agent, is modelled as a unit (travelagency) within which are provided search services tourist information and booking of hotels and flights. Two types of roles within the unit interact travelagency: the role client (customer) and the role of service provider (provider).



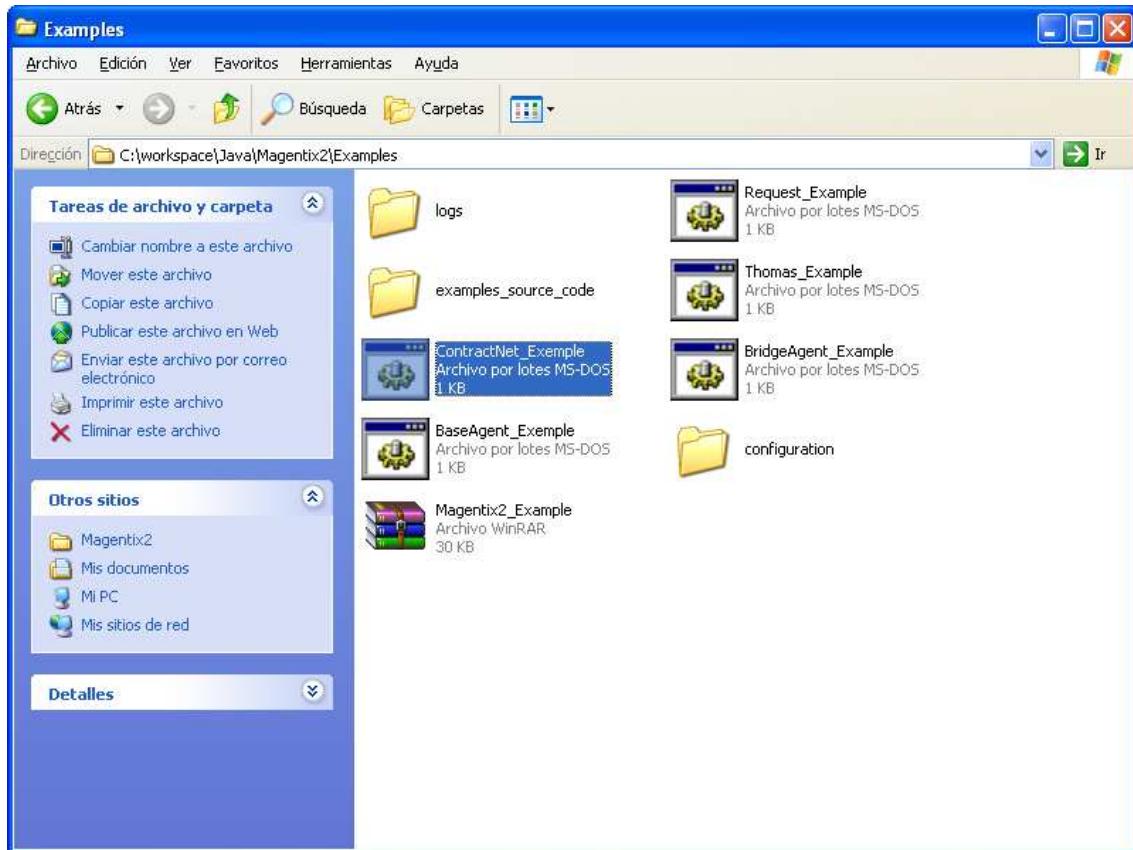
## Content

The examples subdirectory 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)<sup>9</sup>.
- “*\*.sh files*”: run examples directly (linux)
- “*/logs*” subdirectory: default log4j exit.
- “*/configuration*”: includes “*settings.xml*” and “*logs.xml*” configuration files of Magentix2 platform.



## Run Magentix2 examples (windows)



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-de
pendencies.jar;
C:\workspace\Java\Magentix2\Examples>set CLASSPATH=../magentix2-1.0-jar-with-de
pendencies.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

## **Architecture**

Recently several works have appeared that try to solve the problem of integrating the multi-agent system paradigm and the service-oriented computing paradigm. It is obvious, that there are many similarities among them. Both paradigms try to offer solutions for the development of complex and adaptive systems in distributed open environments. In this line, integrating these technologies is possible to model autonomous and heterogeneous computational entities in dynamic and open environments. Such entities may be reactive, proactive and with the ability to communicate in a flexible way with other entities. One of the existing proposals works in the line to create links, as a gateway, between the two directions. The proposed solution tries to communicate agents and web services in a transparent, but independent, way. This is the line of the Agent and Web Services Interoperability (AWSI) IEEE FIPA Working Group (<http://www.fipa.org/subgroups/AWSI-WG.html>). Although interesting, our proposal tries to go beyond, raising a total integration of both technologies. So agents can offer and invoke services in a transparent way to other agents or entities, as well as external entities can interact with our agents through the use of the offered services. THOMAS architecture consists basically of a set of modular services. THOMAS feeds initially of the FIPA architecture expanding its capabilities. The agents have access to the infrastructure offered by THOMAS through a range of services including on different modules or components. The main components of THOMAS are the following:

- Service Facilitator (SF), this component offers simple and complex services to the active agents and organizations. Basically, its functionality is like a yellow page service and a service descriptor in charge of providing a green page service.
- Organization Manager Service (OMS), it is mainly responsible of the management of the organizations and their entities. Thus, it allows the creation and the management of any organization.

### **Service Facilitator**

The SF is a mechanism and support by which the organization and agents can, at the same time, offer and discover services. The SF provides a place in which the autonomous entities can register service descriptions as directory entries.

The SF acts as a gateway to access the THOMAS platform. It manages this access transparently, by means of security techniques and access rights management. The SF can find services searching for a given service profile or searching by the goals that can be fulfilled executing the service. This is done using the matchmaking and service composition mechanisms that are provided by the SF.

The SF acts also as a yellow pages manager and in this way it can also find which entities provide a given service. A service offers some capacities, each of which to fulfil a given goal. The service may have some pre-conditions that have to be true before the service can be executed. It exchange one or more input and output messages. Before a successful service execution it has some effects on its environment. Moreover, there could be additional parameters,



which are independent of the service functionality (non-functional parameters), such as quality of service, deadlines, and security protocols among other. And finally, the service results can be enhanced using automatic service composition mechanisms (for example, partial matchmaking). To do this the SF maintains the description of the internal processes that are executed when the service is running.

A service represents an interaction of two entities, which are modelled as communications among independent processes. In our case, the Multi-agent Technology provides us with FIPA communication protocols which are well established mechanisms in order to standardize the interactions. In this way, every service has an associated protocol. In those cases in which the service requires the execution of a chain of protocols, the service is marked as "complex". Taking into account that we are working with semantic services, another important data is the ontology used in the service. In this way, when the service description is accessed, any entity will have all the needed information in order to interact with the service and how to make an application that can use the service. Such a description can also be used for pre-compiled services, in which the process model of the service will be, instead of the internal processes of the service, the sequence of the elementary services that will be executed.

### **Organization Manager Service**

This component is in charge of organizations life cycle management, including specification and administration of both their structural components (roles, units and norms) and their execution components (participant agents and the roles they play; active units in each moment). OMS offers all services needed for a suitable organization performance. These services are classified as:

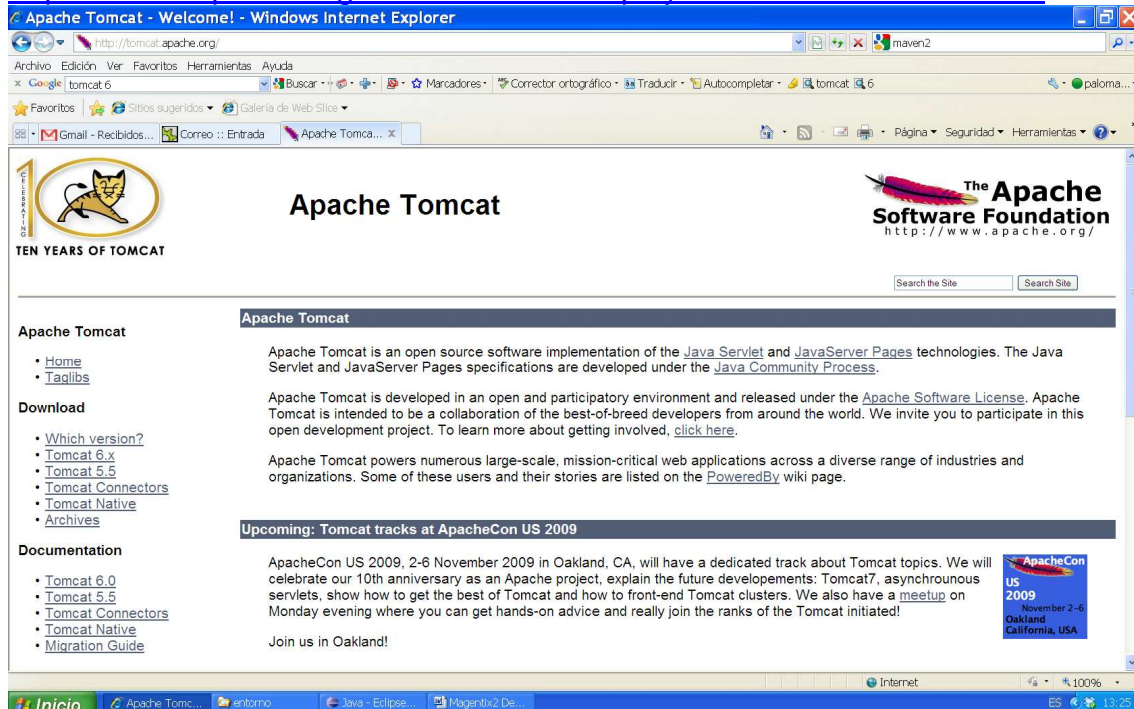
- Structural services, that modify the structural and normative organization specification;
- Dynamical services, that allow agents to entry or leave the organization dynamically, as well as role adoption.

By means of the publication of the structural services, OMS allows modifying, in execution time, some aspects related to the organization structure, functionality or normativity. For example, a specific agent of the organization could be allowed to add new norms, roles or units. This type of services should be restricted to internal roles of the system, which have enough permission for doing this kind of operations (i.e. supervisor role). Moreover, in some concrete applications those services might not be published in the SF, so then agents cannot dynamically modify the structural components. Dynamical services manage creation of new agents in the organization, entry or exit of unit members and role adoption. These services are always published in the SF.

## Additional requisites

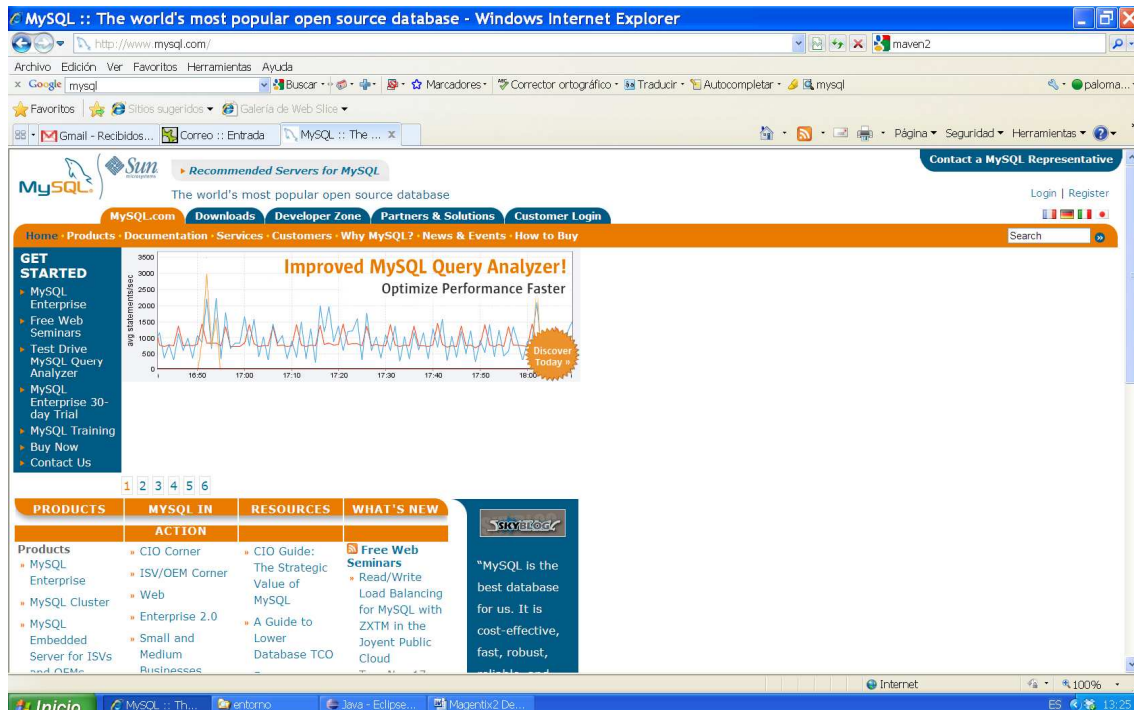
### TOMCAT 6.0 (recommended server)

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



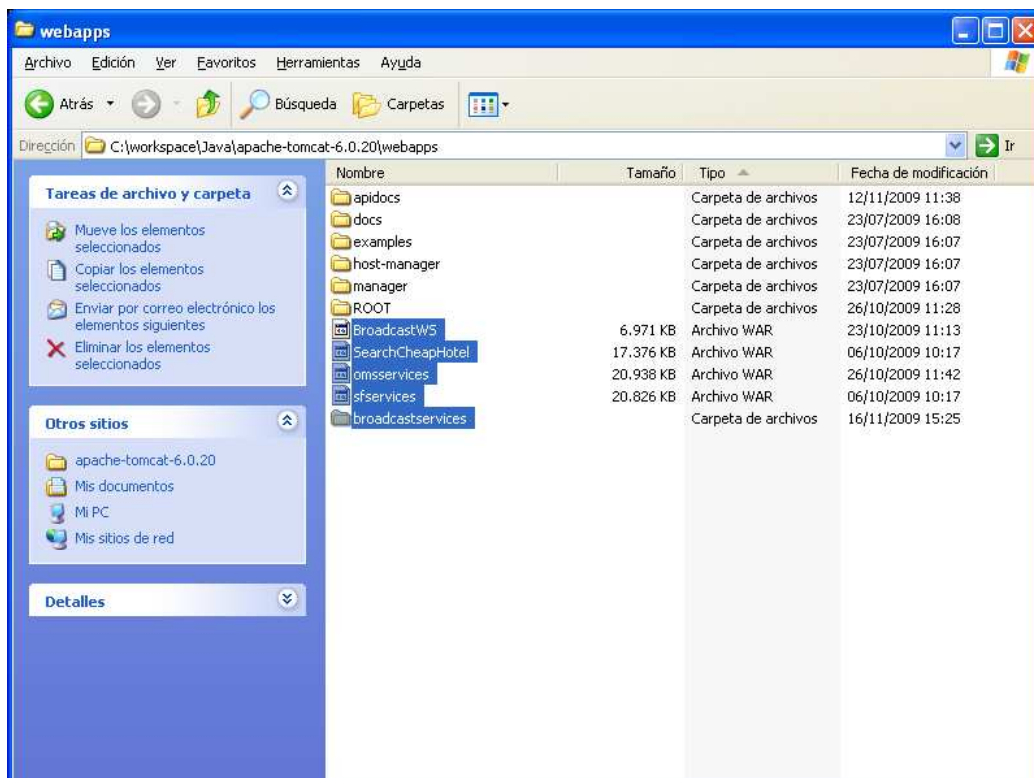
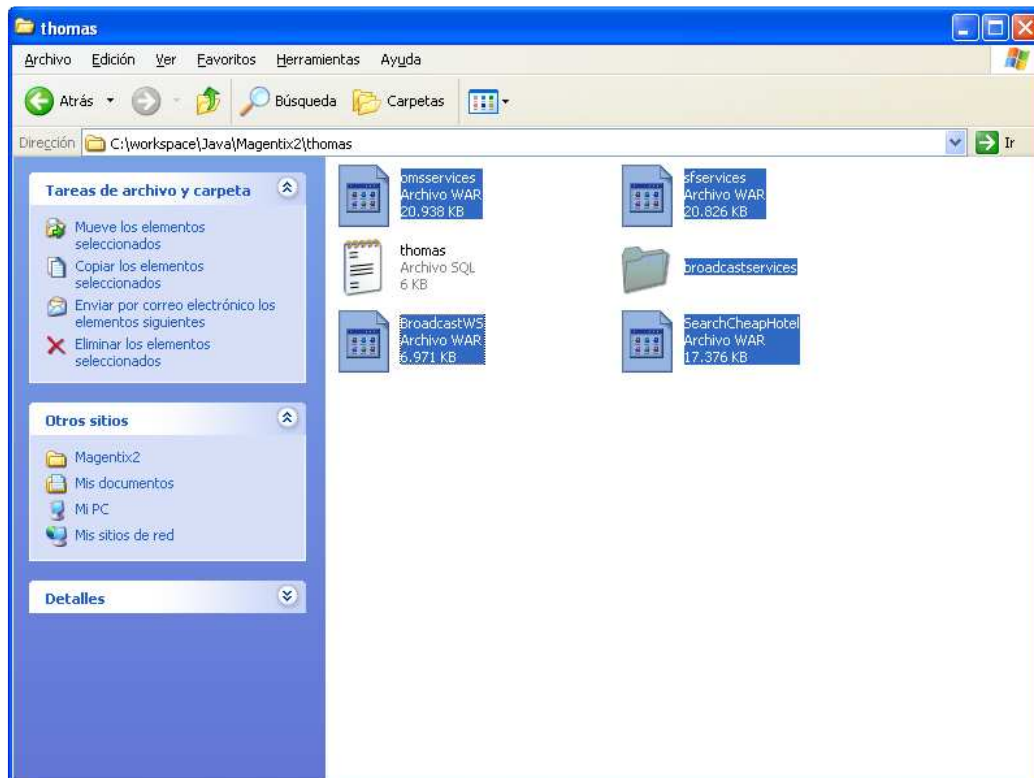
### MySQL (recommended DDBB)

<http://www.mysql.com/>

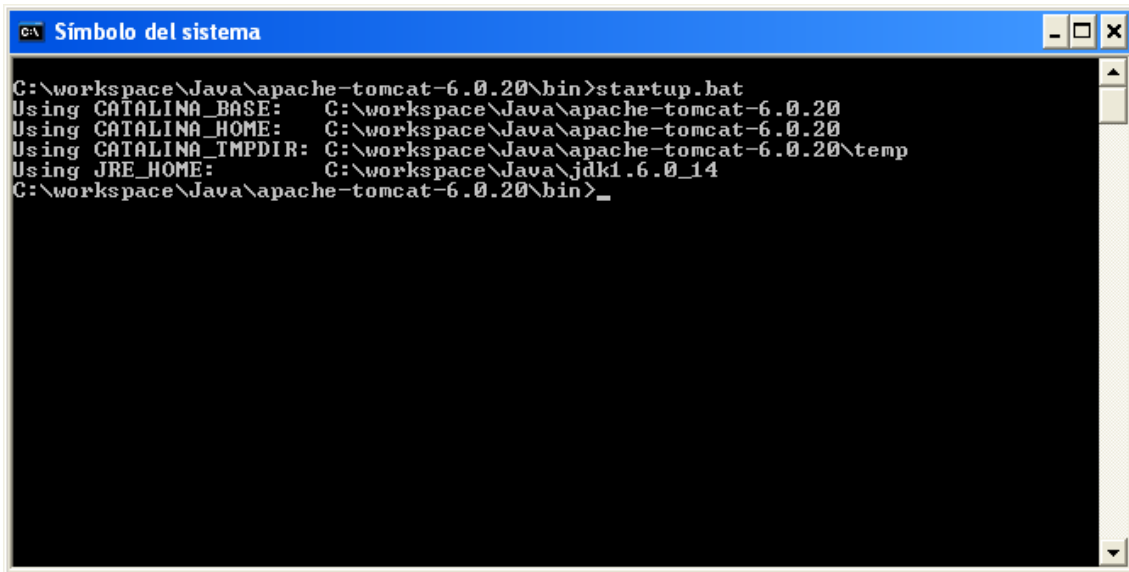


## Deploy Thomas in TOMCAT

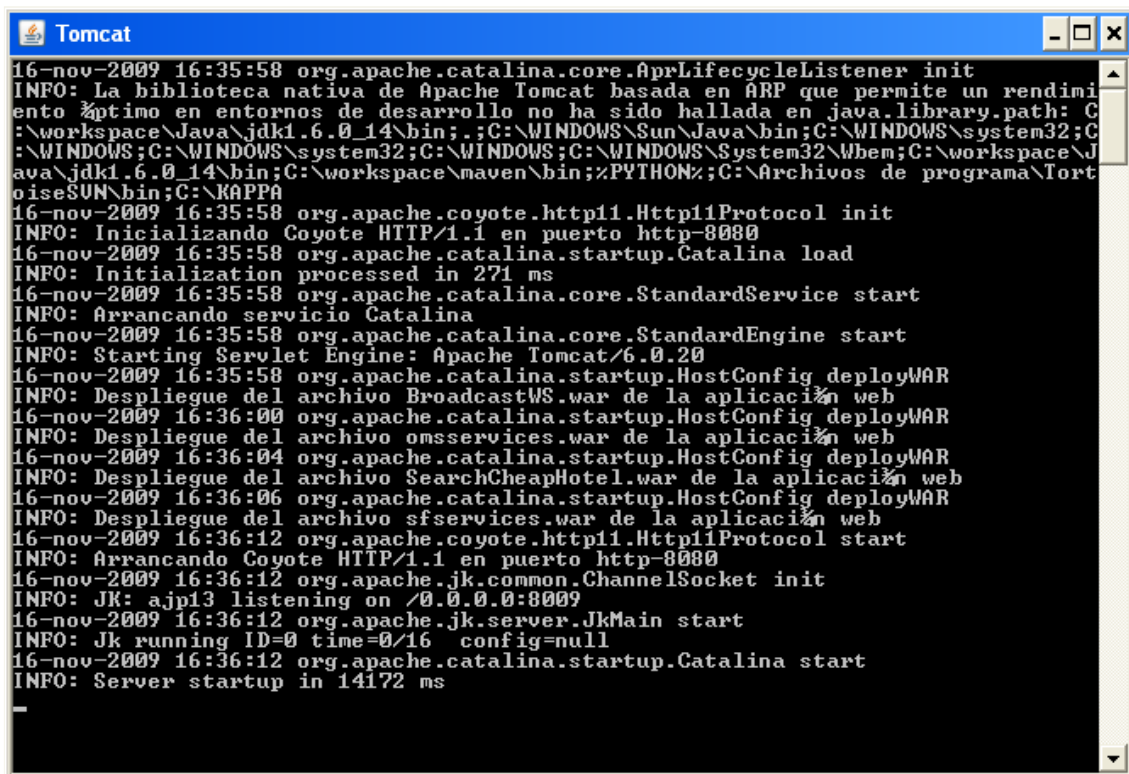
Copy packaged libraries of Thomas (*omsservices.war*, *sfservices.war*) and also the Thomas examples (*SearchCheapHotel.war*, *BroadcastWS.war* and the *broadcastservices* directory) from Magentix2 distribution to the subdirectory *webapps* of Tomcat:



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.



```
C:\workspace\Java\apache-tomcat-6.0.20\bin>startup.bat
Using CATALINA_BASE:   C:\workspace\Java\apache-tomcat-6.0.20
Using CATALINA_HOME:   C:\workspace\Java\apache-tomcat-6.0.20
Using CATALINA_TMPDIR: C:\workspace\Java\apache-tomcat-6.0.20\temp
Using JRE_HOME:        C:\workspace\Java\jdk1.6.0_14
C:\workspace\Java\apache-tomcat-6.0.20\bin>
```

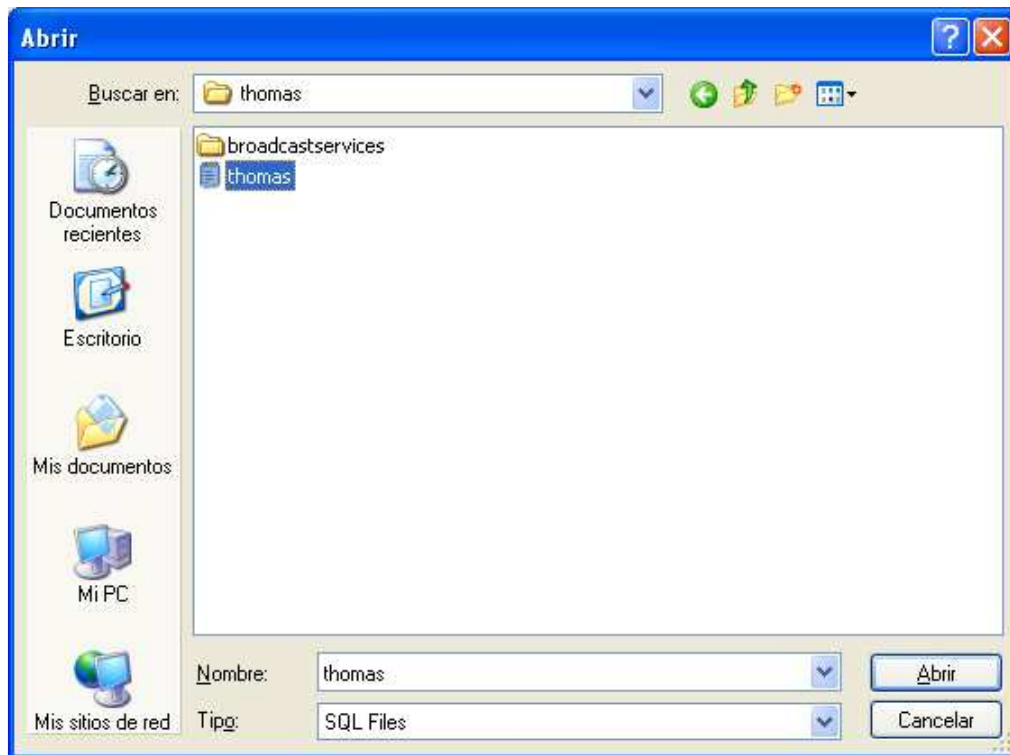


```
16-nov-2009 16:35:58 org.apache.catalina.core.AprLifecycleListener init
INFO: La biblioteca nativa de Apache Tomcat basada en APR que permite un rendimien
to ptimo en entornos de desarrollo no ha sido hallada en java.library.path: C
:\workspace\Java\jdk1.6.0_14\bin;.;C:\WINDOWS\Sun\Java\bin;C:\WINDOWS\system32;C
:\WINDOWS;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\workspace\J
ava\jdk1.6.0_14\bin;C:\workspace\maven\bin;%PYTHON%;C:\Archivos de programa\Tort
oise\SUN\bin;C:\KAPPA
16-nov-2009 16:35:58 org.apache.coyote.http11.Http11Protocol init
INFO: Inicializando Coyote HTTP/1.1 en puerto http-8080
16-nov-2009 16:35:58 org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 271 ms
16-nov-2009 16:35:58 org.apache.catalina.core.StandardService start
INFO: Arrancando servicio Catalina
16-nov-2009 16:35:58 org.apache.catalina.core.StandardEngine start
INFO: Starting Servlet Engine: Apache Tomcat/6.0.20
16-nov-2009 16:35:58 org.apache.catalina.startup.HostConfig deployWAR
INFO: Despliegue del archivo BroadcastWS.war de la aplicacin web
16-nov-2009 16:36:00 org.apache.catalina.startup.HostConfig deployWAR
INFO: Despliegue del archivo onsservices.war de la aplicacin web
16-nov-2009 16:36:04 org.apache.catalina.startup.HostConfig deployWAR
INFO: Despliegue del archivo SearchCheapHotel.war de la aplicacin web
16-nov-2009 16:36:06 org.apache.catalina.startup.HostConfig deployWAR
INFO: Despliegue del archivo sfservices.war de la aplicacin web
16-nov-2009 16:36:12 org.apache.coyote.http11.Http11Protocol start
INFO: Arrancando Coyote HTTP/1.1 en puerto http-8080
16-nov-2009 16:36:12 org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
16-nov-2009 16:36:12 org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/16 config=null
16-nov-2009 16:36:12 org.apache.catalina.startup.Catalina start
INFO: Server startup in 14172 ms
```

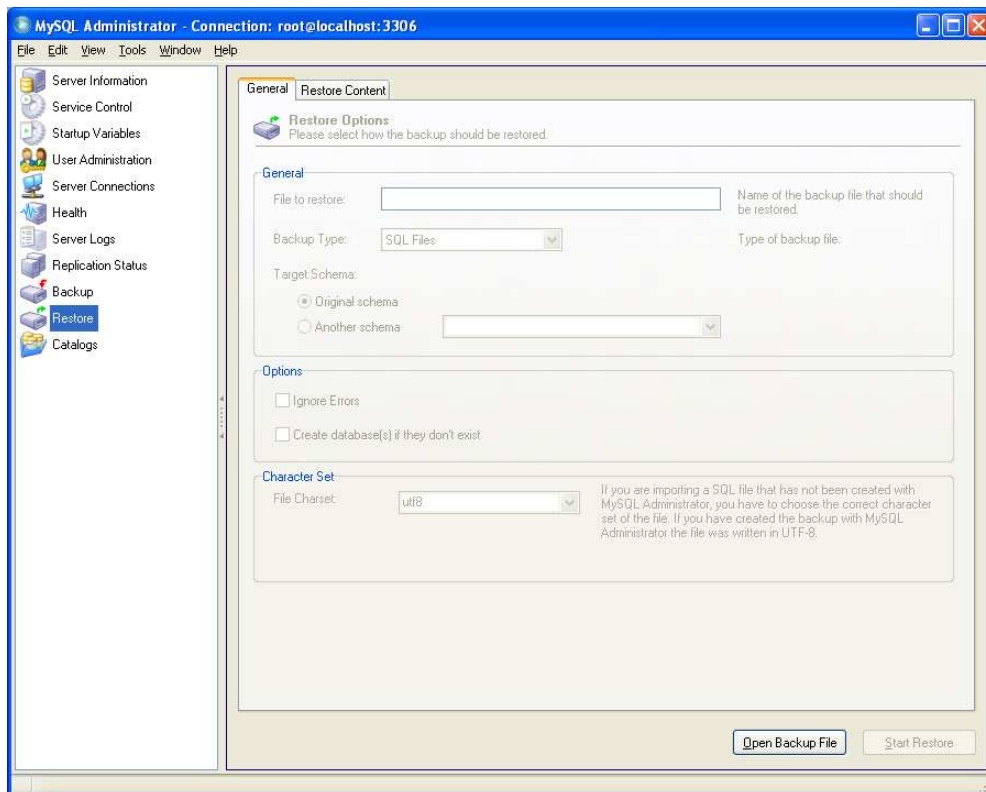


## Install Thomas DDBB in MySQL

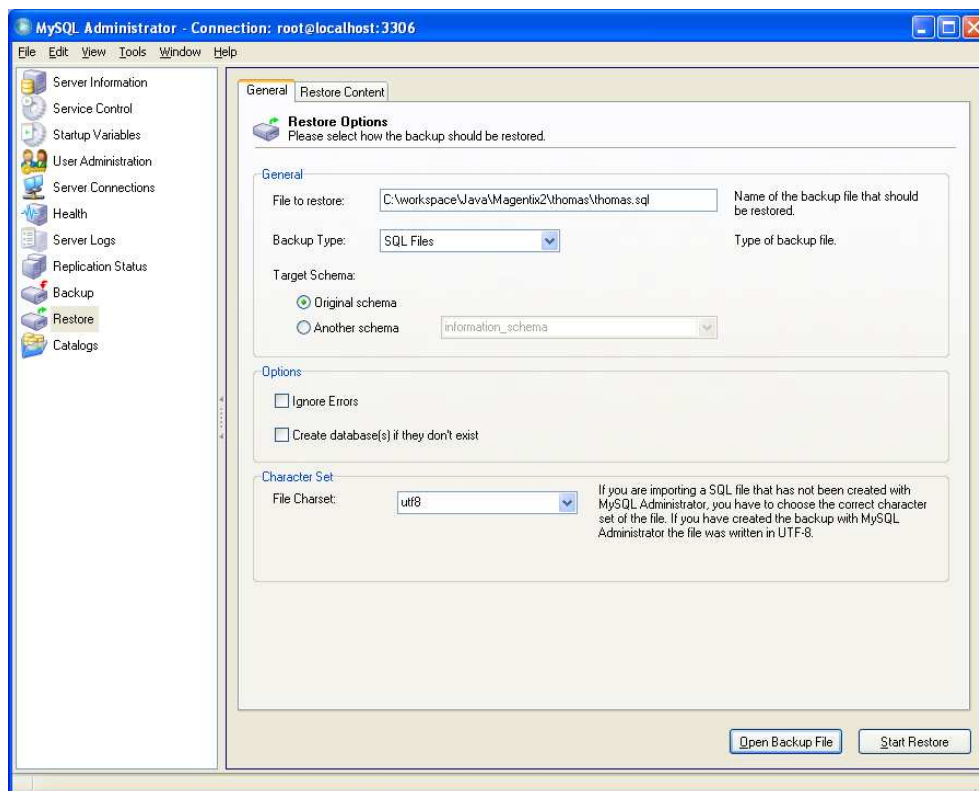
“Thomas.sql” file allow restore Thomas Data Base in MySQL:



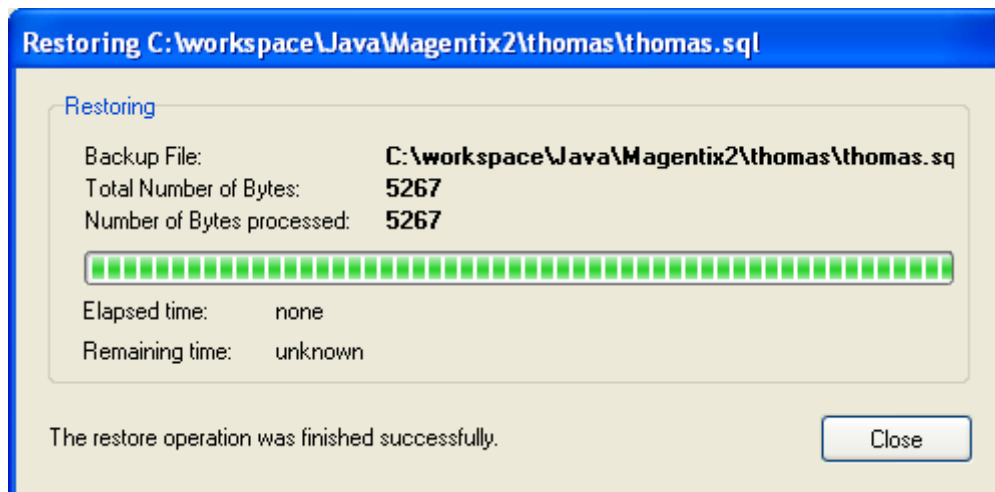
Open the MySQL Administrator and select *restore* option:



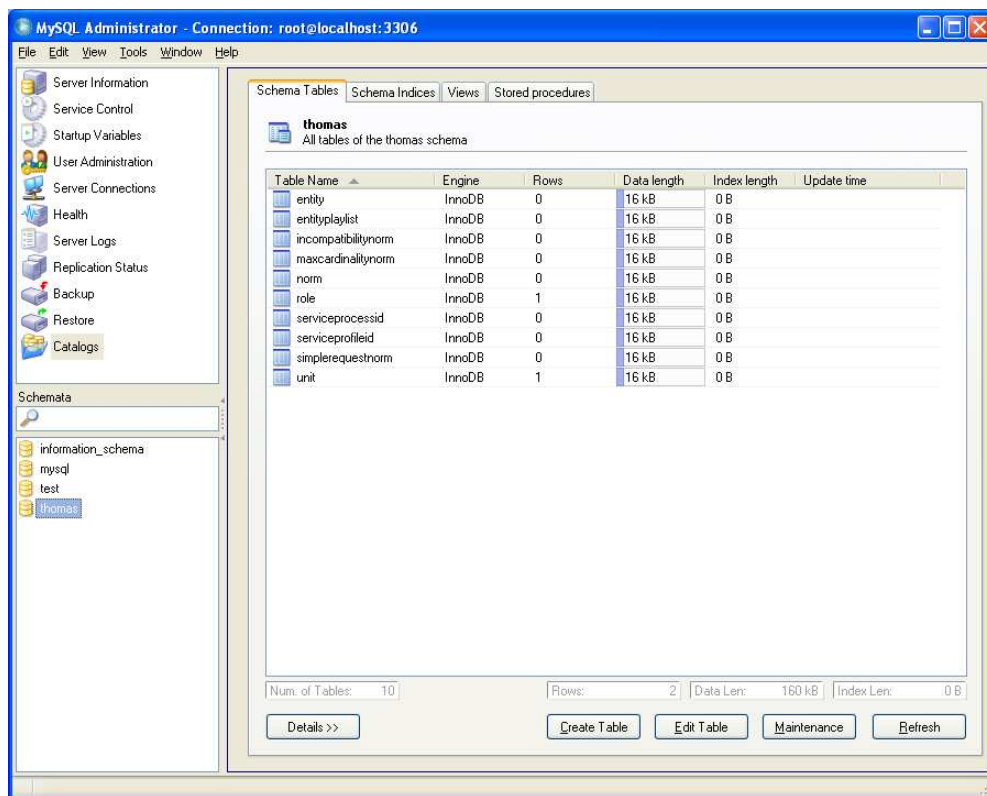
Select *Thomas.sql* file:



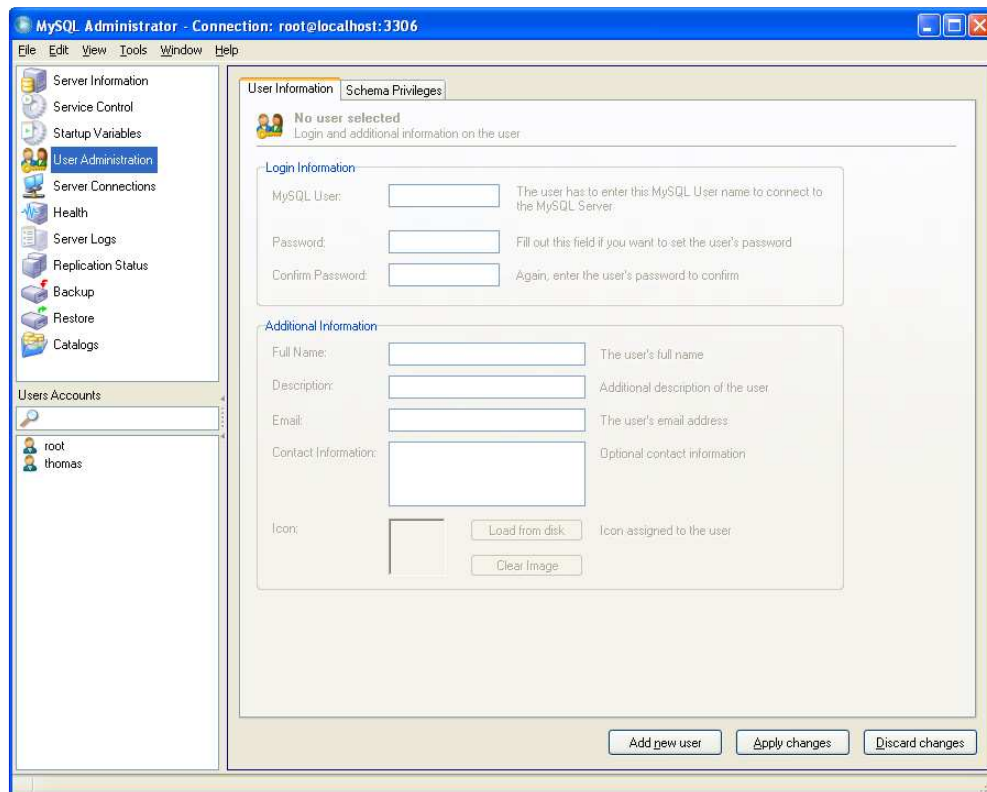
Init restore process:



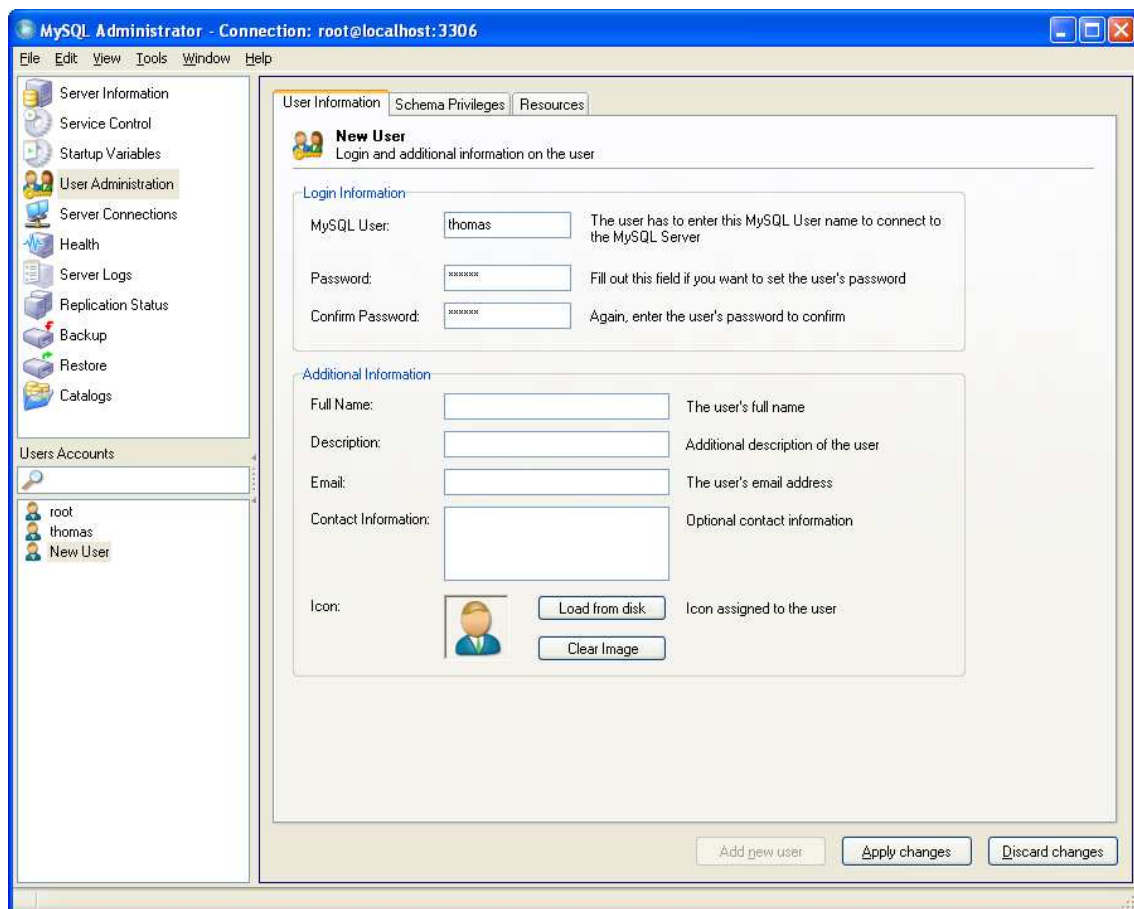
Now Thomas DDBB could be installed in MySQL



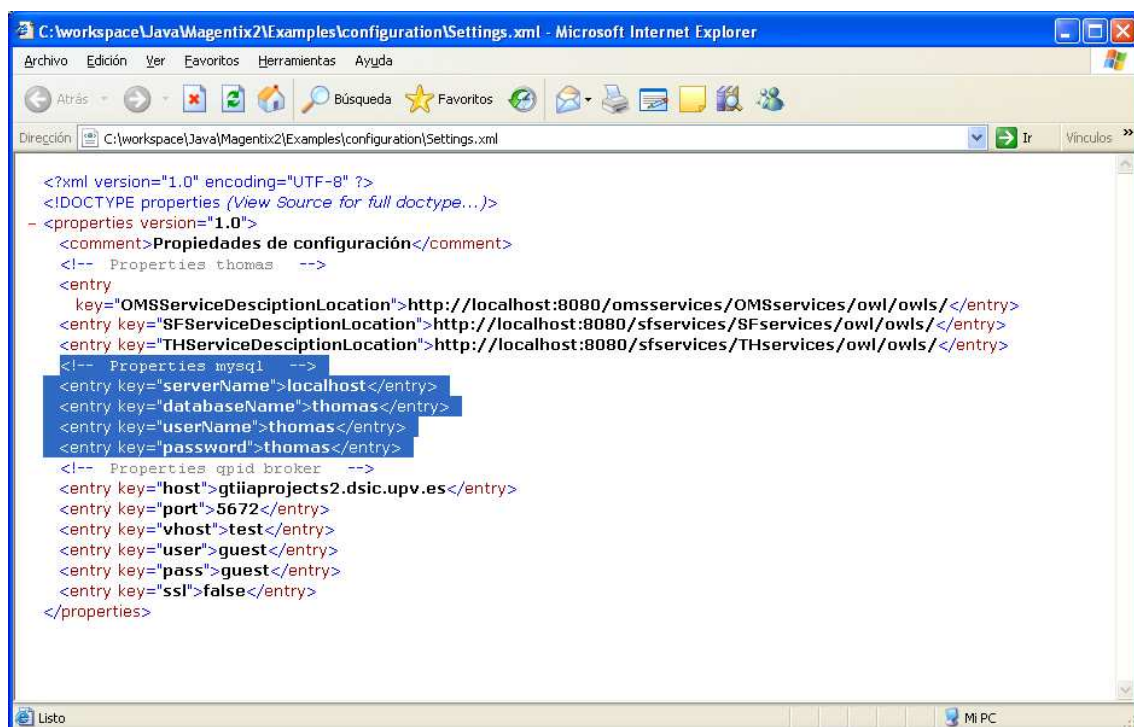
Add a new user to Thomas Schema:



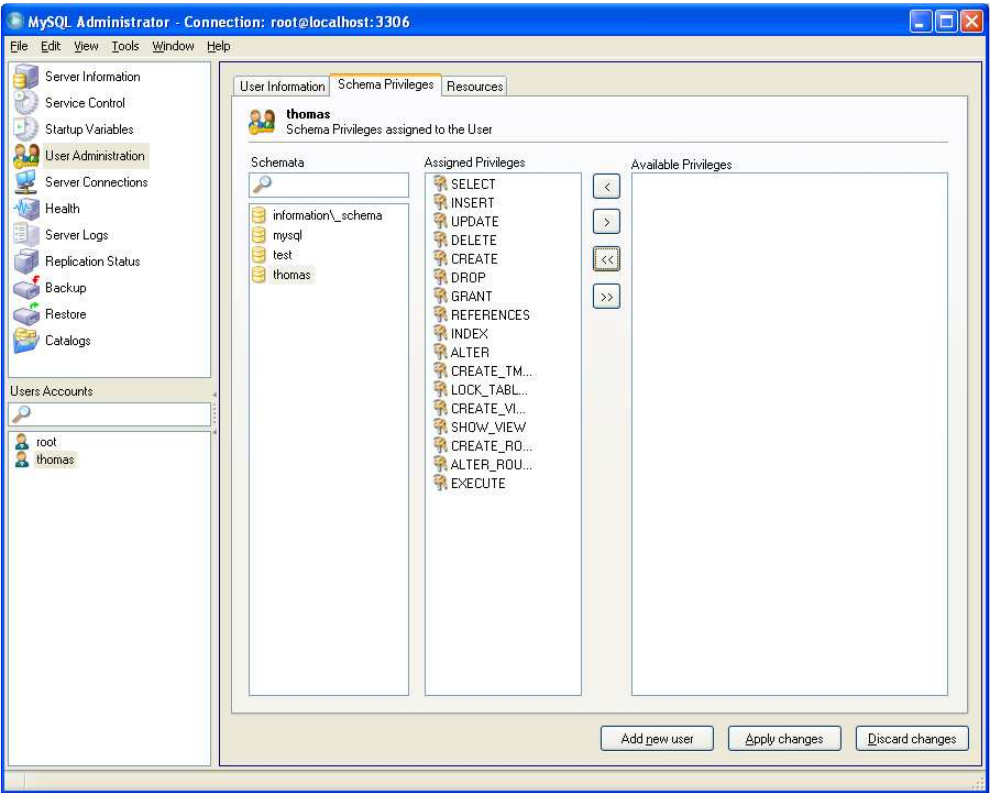
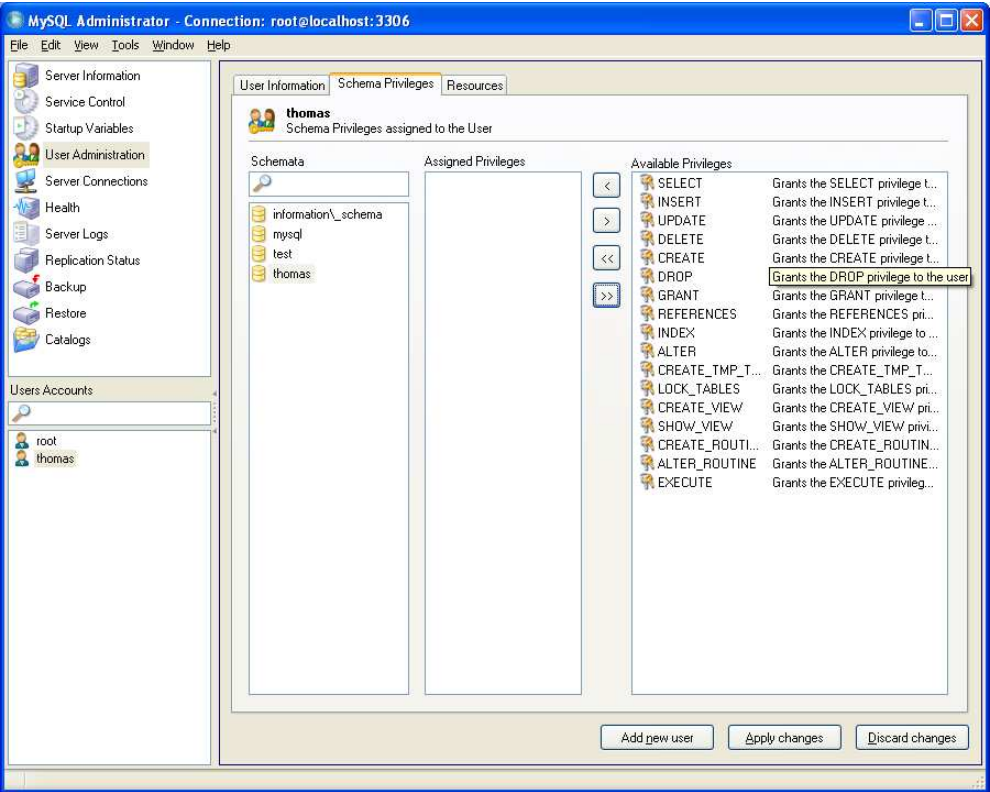
For example we add the user="thomas" with password= "thomas"



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

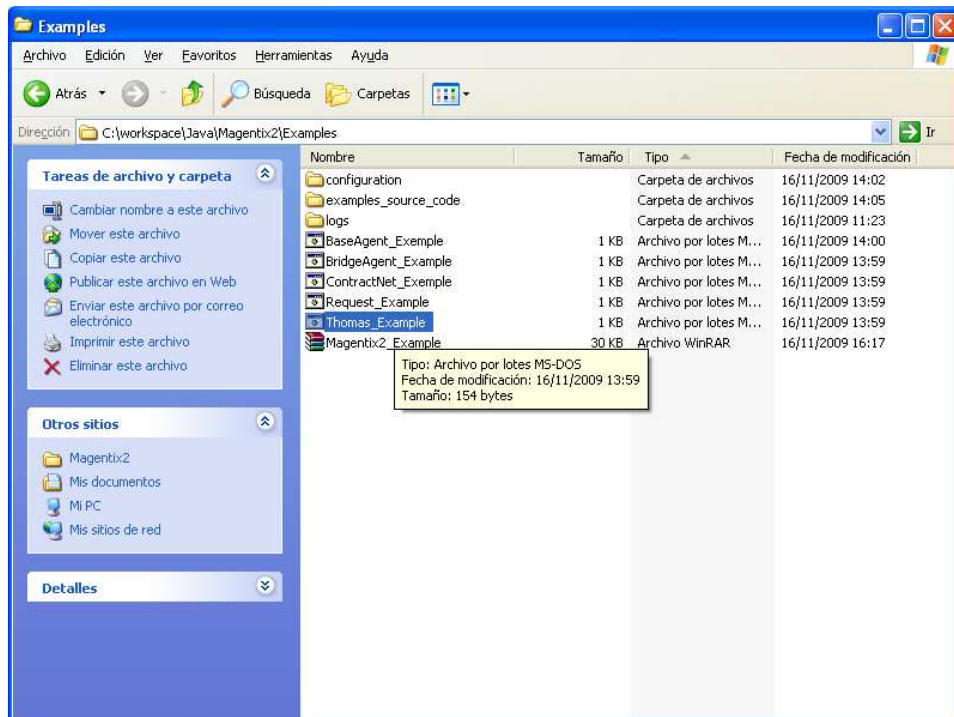


Finally available privileges for all tables of Thomas must be assigned to this user:





## Run Thomas Example



```
C:\WINDOWS\system32\cmd.exe

C:\workspace\Java\Magentix2\Examples>set CLASSPATH=;../magentix2-1.0-jar-with-de
pendencies.jar;;Magentix2_Example.jar;

C:\workspace\Java\Magentix2\Examples>java Thomas_Example.Run
[BroadcastAgent]Acquire Role result: Ok

[ClientAgent]Acquire Role member return: Ok

[BroadcastAgent]Register Unit result: Ok

[ClientAgent]Acquire Role customer return: Ok

[BroadcastAgent]Register Role result: Ok

[BroadcastAgent] conencted with rol customer: Ok
[BroadcastAgent]The operation getProfile return: 30

[BroadcastAgent]The operation getProcess return: 30@30-qpid://BroadcastAgent@loc
alhost:8080

[BroadcastAgent]Agents who has the broadcast service: BroadcastAgent

[BroadcastAgent]Register Profile return: 31

[BroadcastAgent]Register Porcess return: 31@31-qpid://BroadcastAgent@localhost:8
080

[ClientAgent]agents that offered SearchCheapHotel service: 1

[ClientAgent] agents who have the service SearchCheapHotel: BroadcastAgent

[ClientAgent]get Profile return: http://localhost:8080/sfservices/THservices/owl
/owls/SearchCheapHotelProfile.owl

[Provider]Doc OWL-S: http://localhost:8080/sfservices/THservices/owl/owls/Search
CheapHotelProcess.owl
AGREE
[Provider]Sending First message:es.upv.dsic.gti_ia.core.ACLMessage@10e4fd7
[Provider]Doc OWL-S: http://localhost:8080/sfservices/THservices/owl/owls/Search
CheapHotelProcess.owl
[Provider]Executing... [SecondParam, ThirdParam, FirstParam]
[Provider]Values obtained... :{http://localhost:8080/sfservices/THservices/owl/o
wls/SearchCheapHotelProcess.owl#SearchCheapHotelOutputHotel=Hotel Puerta de Vale
ncia, http://localhost:8080/sfservices/THservices/owl/owls/SearchCheapHotelProce
ss.owl#SearchCheapHotelOutputHotelCompany=NH}
[Provider]Creating inform message to send...
[Provider]Before set message content...
```

# Developing Agents for Magentix2

## *Sender Agent*

```
public class SenderAgent extends BaseAgent {

    public SenderAgent(AgentID aid) throws Exception {
        super(aid);
    }

    public void execute() {
        logger.info("Executing, I'm " + getName());
        AgentID receiver = new AgentID("consumer");

        /**
         * Building a ACLMessage
         */
        ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);
        msg.setReceiver(receiver);
        msg.setSender(this.getAid());
        msg.setLanguage("ACL");
        msg.setContent("Hello, I'm " + getName());
        /**
         * Sending a ACLMessage
         */
        send(msg);
    }
}
```

## *Consumer Agent*

```
public class ConsumerAgent extends BaseAgent {

    LinkedBlockingQueue<MessageTransfer> internalQueue;

    public ConsumerAgent(AgentID aid) throws Exception {
        super(aid);
    }

    public void execute() {
        logger.info("Executing, I'm " + getName());
        /**
         * This agent has no definite work. Wait infinitely the arrival of new
         * messages.
         */
        while (true) {

        }
    }

    public void onMessage(ACLMessage msg) {
        /**
         * When a message arrives, its shows on screen
         */
        logger.info("Mensaje received in " + this.getName()
            + " agent, by onMessage: " + msg.getContent());
    }
}
```

## ***Client Agent***

acquired the member role at the organization

```
result = OMSServices.acquireRole(this, "member", "virtual");
```

```
result = OMSServices.acquireRole(this, "customer", "travelagency");
```

waiting that the agentBroadcast registered service SearchCheapHotel

```
do{  
    results = SFservices.searchService(this, "SearchCheapHotel");  
}while(results.size()==0);
```

```
agents = SFservices.getProcess(this, results.get(0));
```

```
for (AgentID agent : agents)  
    System.out.println("[ClientAgent] agents who have the service  
SearchCheapHotel: "  
                        + agent.name+"\n");
```

```
String profile = SFservices.getProfile(this, results.get(0));
```

```
    ArrayList<String> arg = new ArrayList<String>();  
    arg.add("FirstParam");  
    arg.add("SecondParam");  
    arg.add("ThirdParam");
```

call the service SearchCheapHotel

```
SFservices.genericService(this, agents.get(0), profile, "http://localhost:8080/sfservices/THservice  
s/owl/owls/SearchCheapHotelProcess.owl", arg);
```

## PROVIDER AGENT

//We create the class that will make us the agent proxy oms, facilitates access to the methods of the OMS

```
OMSPProxy OMSServices = new OMSPProxy();
```

//We create the class that will make us the agent proxy sf, facilitates access to the methods of the SF

```
SFProxy SFservices = new SFProxy();
```

//We create a SFServiceDescription, one for service that we have

```
SFServiceDescription serviceOne = new
```

```
SFServiceDescription("http://localhost:8080/broadcastservices/owl/owls/", "http://localhost:8080/broadcastservices/owl/owls/");
```

acquired the member role at the organization

```
try
```

```
{
```

```
result = OMSServices.acquireRole(this, "member", "virtual");
```

```
result = OMSServices.acquireRole(this, "customer", "travelagency");
```

```
serviceOne.setServiceGoal("SearchCheapHotel");
```

```
SFservices.registerProfile(this, serviceOne);
```

```
System.out.println("[ProviderAgent]The operation register Profile return:  
"+ serviceOne.getID()+"\n");
```

```
SFservices.registerProcess(this, serviceOne);
```

//RoI responder

```
Responder responder = new Responder(this);
```

```
this.addTask(responder);
```

//when we do not have to create more roles we await the expiration of the other roles

```
Monitor m = new Monitor();
```

```
m.waiting();
```

```
}catch(Exception e){
```

```
logger.error(e.getMessage());
```

```
}
```

## Launching agents:

### ***Initialization tasks:***

Our platform have been developed with log4j, for this reason is necessary it will be initialized as follows:

- i. `DOMConfigurator.configure("configuration/loggin.xml");`
- ii. `Logger logger = Logger.getLogger(Run.class);`

### ***Connecting to Qpid broker***

Always before launch any agent, a connection to the Qpid broker must have been established. Thus any agent in our platform will use this communication. There are three different ways to establish a connection with Qpid broker:

- `Connect()`  
Take the input connection parameters from settings.xml file.
- `connect(qpidHost, qpidPort, qpidVhost, qpidUser, qpidPassword, qpidSSL)`  
Take into account all the parameters specefied as input
- `Connect(qpidHost)`

Take into account the qpidhost parameter and considering the rest as default parameters.

For instance:

- `AgentsConnection.connect();`
- `AgentsConnection.connect(localhost, 5672,"test","guest","guest",false);`
- `AgentsConnection.connect("localhost ");`

### ***Instantiating agents***

Once created agents, we can now instantiate it. Please note that the platform can not allow several agents with the same name.

- i. `SenderAgent agent1 = new SenderAgent(new AgentID("qpid://emisor@localhost:8080"));`
- ii. `ConsumerAgent agent2 = new ConsumerAgent(new AgentID("consumer"));`

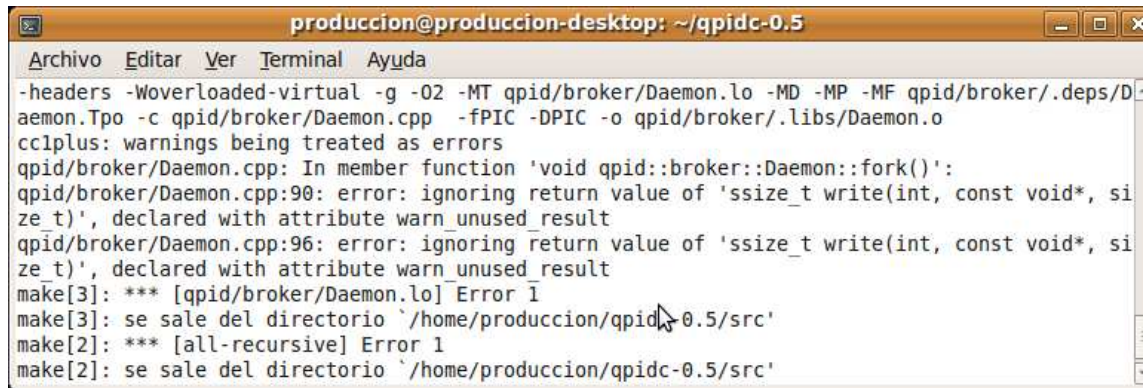
#### 1. Launching agents

- i. `agent2.start();`
- ii. `agent.start();`



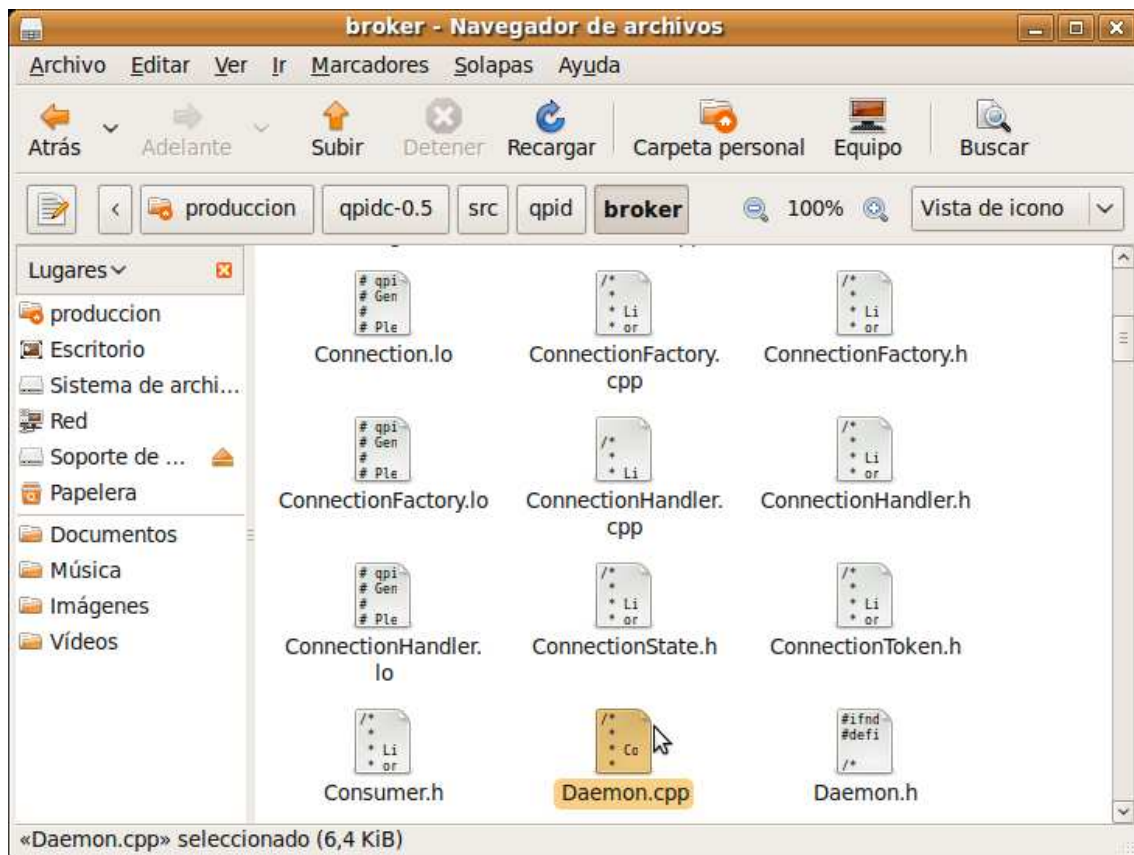
## APENDIX I Qpid Source Code modifications

**Problem:** ignoring the return value of 'ssize\_t' in the 'write' function causes a compilation error. This error was a warning in previous versions of the c++ compiler but must be corrected in order to compile and install the broker in Ubuntu 9.04



```
produccion@produccion-desktop: ~/qpidc-0.5
Archivo  Editar  Ver  Terminal  Ayuda
-headers -Woverloaded-virtual -g -O2 -MT qpid/broker/Daemon.lo -MD -MP -MF qpid/broker/.deps/Daemon.Tpo -c qpid/broker/Daemon.cpp -fPIC -DPIC -o qpid/broker/.libs/Daemon.o
cc1plus: warnings being treated as errors
qpid/broker/Daemon.cpp: In member function 'void qpid::broker::Daemon::fork()':
qpid/broker/Daemon.cpp:90: error: ignoring return value of 'ssize_t write(int, const void*, size_t)', declared with attribute warn_unused_result
qpid/broker/Daemon.cpp:96: error: ignoring return value of 'ssize_t write(int, const void*, size_t)', declared with attribute warn_unused_result
make[3]: *** [qpid/broker/Daemon.lo] Error 1
make[3]: se sale del directorio `/home/produccion/qpidc-0.5/src'
make[2]: *** [all-recursive] Error 1
make[2]: se sale del directorio `/home/produccion/qpidc-0.5/src'
```

Different `ssize_t` type variables “basura” have been added in the classes Daemon.cpp, LockFile.cpp and ManagementAgentImpl.cpp in order to compile Qpid broker in UBUNTU.



Daemon.cpp (~/qpidc-0.5/src/qpid/broker) - gedit

Archivo Editar Ver Buscar Herramientas Documentos Ayuda

Nuevo Abrir Guardar Imprimir... Deshacer Rehacer Cortar Copiar Pegar

INSTALL x Daemon.cpp x

```
1 if(setuid(0) < 0) throw ErrnoException("Cannot set session id",  
if(chdir(pidDir.c_str()) < 0) throw ErrnoException("Cannot  
change directory to "+pidDir);  
umask(027);  
  
// Child behavior  
child();  
}  
catch (const exception& e) {  
    ssize_t basura;  
    QPID_LOG(critical, "Daemon startup failed: " << e.what());  
    uint16_t port = 0;  
    basura = write(pipeFds[1], &port, sizeof(uint16_t));  
  
    std::string pipeFailureMessage = e.what();  
    write ( pipeFds[1],  
           pipeFailureMessage.c_str(),  
           strlen(pipeFailureMessage.c_str())  
        );  
}  
}  
else { // Parent
```

C++ Ancho de la tabulación: 8 Ln 91, Col 65 INS

\*Daemon.cpp (~/qpidc-0.5/src/qpid/broker) - gedit

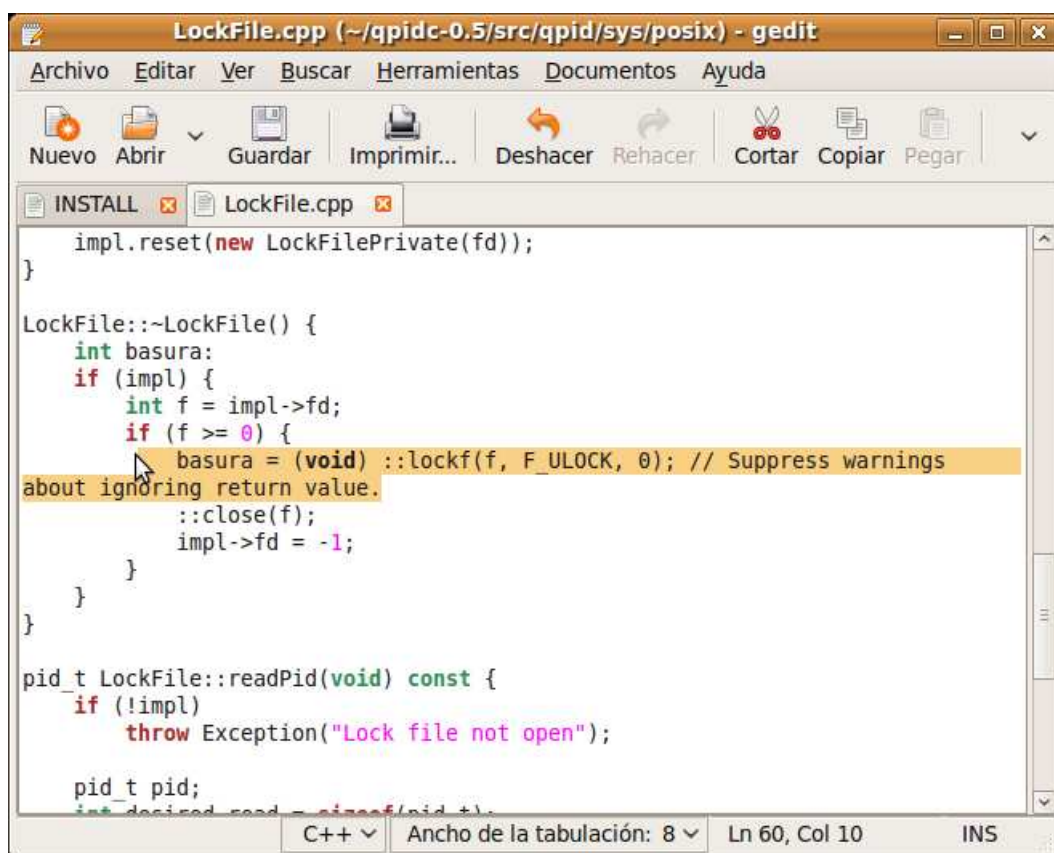
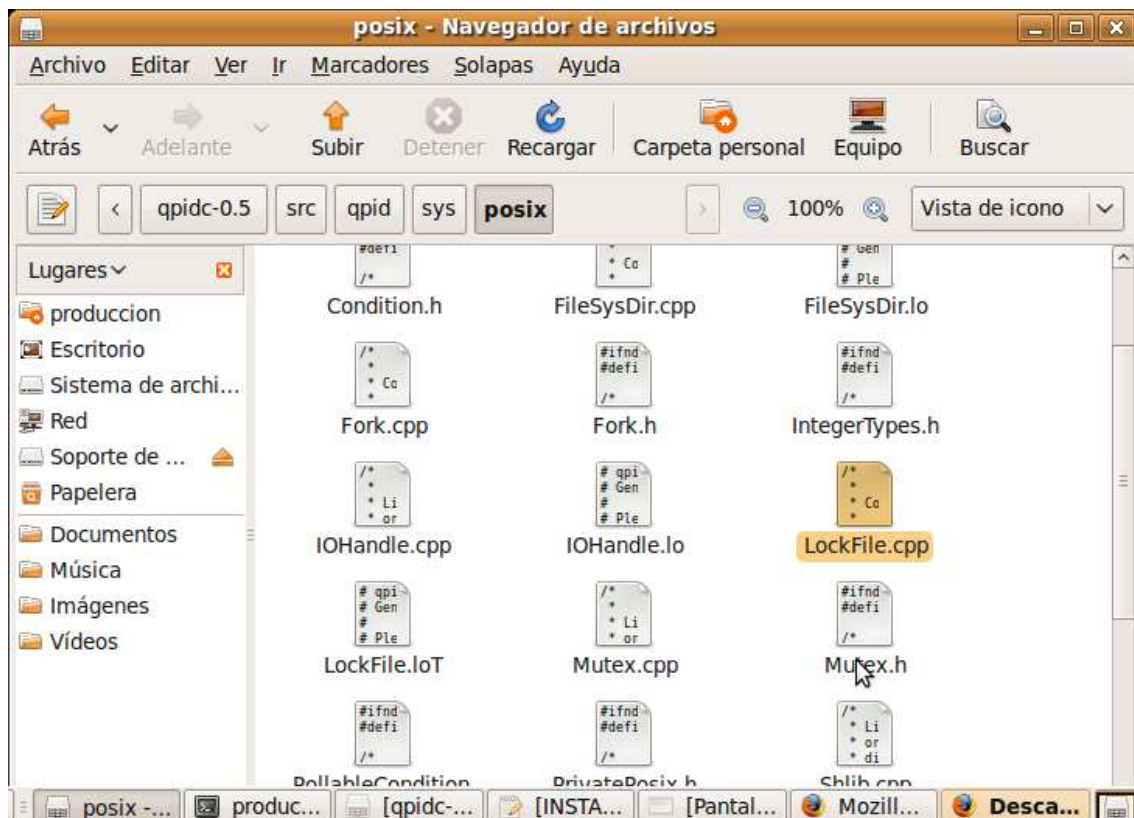
Archivo Editar Ver Buscar Herramientas Documentos Ayuda

Nuevo Abrir Guardar Imprimir... Deshacer Rehacer Cortar Copiar Pegar

INSTALL x \*Daemon.cpp x

```
1 if(setuid(0) < 0) throw ErrnoException("Cannot set session id",  
if(chdir(pidDir.c_str()) < 0) throw ErrnoException("Cannot  
change directory to "+pidDir);  
umask(027);  
  
// Child behavior  
child();  
}  
catch (const exception& e) {  
    ssize_t basura;  
    QPID_LOG(critical, "Daemon startup failed: " << e.what());  
    uint16_t port = 0;  
    basura = write(pipeFds[1], &port, sizeof(uint16_t));  
  
    std::string pipeFailureMessage = e.what();  
    basura = write ( pipeFds[1],  
                    pipeFailureMessage.c_str(),  
                    strlen(pipeFailureMessage.c_str())  
                );  
}  
}  
else { // Parent
```

C++ Ancho de la tabulación: 8 Ln 94, Col 40 INS





LockFile.cpp (~/qpdc-0.5/src/qpid/sys/posix) - gedit

Archivo Editar Ver Buscar Herramientas Documentos Ayuda

Nuevo Abrir Guardar Imprimir... Deshacer Rehacer Cortar Copiar Pegar

INSTALL LockFile.cpp

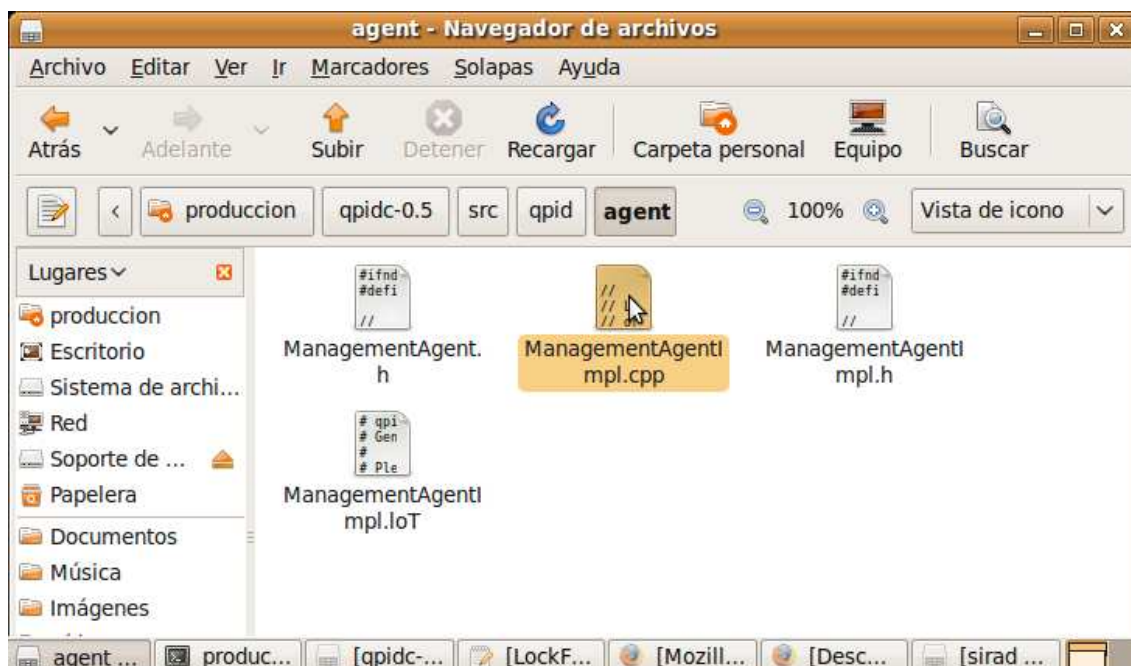
```
impl.reset(new LockFilePrivate(fd));
}

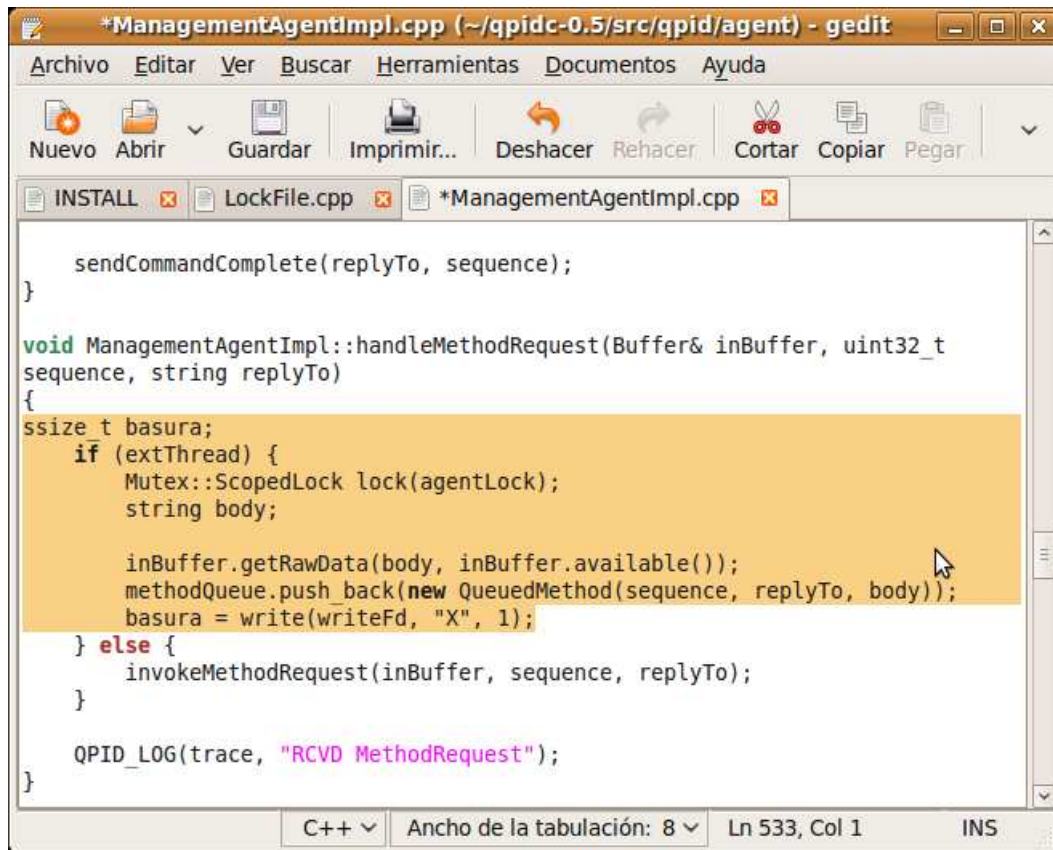
LockFile::~LockFile() {
    int basura;
    if (impl) {
        int f = impl->fd;
        if (f >= 0) {
            basura = ::lockf(f, F_ULOCK, 0); // Suppress warnings about
            ignoring return value.
            ::close(f);
            impl->fd = -1;
        }
    }
}

pid_t LockFile::readPid(void) const {
    if (!impl)
        throw Exception("Lock file not open");

    pid_t pid;
    int desired_read = sizeof(pid + 1);
```

C++ Ancho de la tabulación: 8 Ln 60, Col 23 INS





```
*ManagementAgentImpl.cpp (~/qpidd-0.5/src/qpidd/agent) - gedit
Archivo  Editar  Ver  Buscar  Herramientas  Documentos  Ayuda
Nuevo  Abrir  Guardar  Imprimir...  Deshacer  Rehacer  Cortar  Copiar  Pegar
INSTALL  LockFile.cpp  *ManagementAgentImpl.cpp

    sendCommandComplete(replyTo, sequence);
}

void ManagementAgentImpl::handleMethodRequest(Buffer& inBuffer, uint32_t
sequence, string replyTo)
{
    ssize_t basura;
    if (extThread) {
        Mutex::ScopedLock lock(agentLock);
        string body;

        inBuffer.getRawData(body, inBuffer.available());
        methodQueue.push_back(new QueuedMethod(sequence, replyTo, body));
        basura = write(writeFd, "X", 1);
    } else {
        invokeMethodRequest(inBuffer, sequence, replyTo);
    }

    QPID_LOG(trace, "RCVD MethodRequest");
}

C++  Ancho de la tabulación: 8  Ln 533, Col 1  INS
```

The modified version of QPID source code can be compiled for UBUNTU 9. 04