



# jPOS<sup>TM</sup> Project Guide



# Contents

Abstract.....	5
Copyright.....	7
Status.....	9
<b>Part I: The jPOS Project.....</b>	<b>11</b>
<b>Chapter 1: Resources.....</b>	<b>13</b>
Mailing lists.....	14
Blogs and Feeds.....	14
Downloading.....	14
<b>Chapter 2: Build System.....</b>	<b>19</b>
jPOS Directory Structure.....	20
Module Directory Structure.....	20
Ant Targets.....	22
IDE support.....	23
Running jPOS.....	23
<b>Part II: jPOS.....</b>	<b>25</b>
<b>Chapter 3: Q2.....</b>	<b>27</b>
Running Q2 in Standalone mode.....	28
Running Q2 in Embedded mode.....	28
Q2 Command line arguments.....	28
QBean Descriptors.....	29
QBean Life cycle.....	30
QBean Loggeable.....	32
QBean Configurable.....	33
QBean Optional Attributes.....	33
QBeanSupport.....	33
<b>Chapter 4: jPOS Spaces.....</b>	<b>37</b>
Spaces Overview.....	38
Timeouts.....	40
SpaceFactory.....	40
Local Spaces.....	41

**Part III: jPOS-EE.....43**

**Part IV: HOWTO.....45**

    Creating a jPOS-EE project.....46

    Moving a project to production.....47

GNU AGPL License.....49

CREDITS.....61

Revision History.....63

# Abstract

---

jPOS is an ISO-8583 library/framework that can be used to implement financial interchanges, protocol converters, payment gateways, credit card verification clients and servers (merchant/issuer/acquirer roles), etc.

jPOS Extended Edition (jPOS-EE) goes one step beyond jPOS by providing a development environment with a continuously growing set of proven, ready-to-deploy components based on jPOS' best practices.

This document describes jPOS, jPOS-EE and related projects managed by jPOS.org.

Unless otherwise noted, jPOS.org's projects are distributed under the *GNU Affero General Public License*. See [GNU AGPL License](#) on page 49.



**Attention:**

**IMPORTANT NOTICE**

If, for any reason you don't intend to meet your open source obligations, **you will need a commercial license** - please [contact us](#).



# Copyright

---

jPOS Copyright notice



jPOS Project [<http://jpos.org>] Copyright (C) 2000-2010 Alejandro P. Revilla

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

See [GNU AGPL License](#) on page 49.

All rights reserved. No part of this book may be reproduced in any form or by any electronic or mechanical means, including information storage and retrieval systems, without permission in writing from Alejandro Revilla, except by a reviewer who may quote brief passages in a review.





# Status

---

## Status of this document

This document is work in progress. At some point, it will replace the *jPOS Programmer's Guide* as well as the *jPOS-EE guide*, but there's a long way to go. If you need to use jPOS now, you may consider purchasing the existing documentation, which in addition is part of the program to support jPOS.org projects.

👉 **Tip:** The latest version of this document can be downloaded at <http://jpos.org/doc/jPOS-Project-Guide.pdf>

👉 **Note:** This is version 1.6, \$Revision: 7210 \$, \$Date: 2010-12-09 10:04:06 -0200 (Thu, 09 Dec 2010) \$. See [Revision History](#) on page 63 for details.



---

# Part

# I

---

## The jPOS Project

---

**Topics:**

- [Resources](#)
- [Build System](#)

jPOS.org manages several projects, such as jPOS-EE, jPOS CMF, jCard, jPTS, etc. This part provides general information about all of them.



---

# Chapter 1

---

## Resources

---

### Topics:

- [Mailing lists](#)
- [Blogs and Feeds](#)
- [Downloading](#)

### Project related resources


👉 **Tip:** Up-to-date links to jPOS related resources, including a link to the latest version of this document can be found online at <http://jpos.org/resources>

## Mailing lists

At the beginning, we just had one mailing list, **jpos-dev** which has over 1600 members.

At some point we decided to split development related stuff from user related support questions and opened a new mailing list, **jpos-users**, hosted at Google Groups.

Although **jpos-dev** is still available, **jpos-users** has better and faster search capabilities, so we are discouraging the use of **jpos-dev** and moving to a single list: **jpos-users**.

 **Tip:** If you are searching the archives, you probably want to search both of them. If you are posting new comments or questions, we suggest you use the latter.

In addition to **jpos-dev** and **jpos-users**, we have **jpos-commits** where we post commit notifications.

**Table 1: Mailing list info**

List	Information
<b>jpos-users</b>	<a href="http://groups.google.com/group/jpos-users">http://groups.google.com/group/jpos-users</a>
<b>jpos-commits</b>	<a href="http://groups.google.com/group/jpos-commits">http://groups.google.com/group/jpos-commits</a>
<b>jpos-dev</b>	<a href="http://www.yahoogroups.com/group/jpos-dev">http://www.yahoogroups.com/group/jpos-dev</a>

## Blogs and Feeds

We regularly post project related news to the jPOS blog.


In addition, there's some activity in Twitter tagged as **#jPOS**

**Table 2: Blogs and Feeds**

Feed	URL	RSS
<b>Blog</b>	<a href="http://jpos.org/blog">jpos.org/blog</a>	<a href="http://feeds2.feedburner.com/jpos">feeds2.feedburner.com/jpos</a>
<b>Commits</b>	<a href="http://groups.google.com/group/jpos-commits">groups.google.com/group/jpos-commits</a>	<a href="http://feeds2.feedburner.com/jpos-commits">feeds2.feedburner.com/jpos-commits</a>
<b>Tweets</b>	<a href="http://search.twitter.com/search?q=jpos">search.twitter.com/search?q=jpos</a>	<a href="http://search.twitter.com/search.atom?q=jpos">search.twitter.com/search.atom?q=jpos</a>
<b>Commit Tweets</b>	<a href="http://twitter.com/jposcommits">twitter.com/jposcommits</a>	<a href="http://twitter.com/statuses/user_timeline/69335521.rss">twitter.com/statuses/user_timeline/69335521.rss</a>

## Downloading

From time to time, we post new jPOS releases at <http://jpos.org/download> but if you want to stay up to date with jPOS development and you want to get the latest features and bug fixes, we **strongly suggest** you to take the time to install a Git client (jPOS) and Subversion client (jPOS-EE) and get the latest and greatest version from <https://github.com/jpos/jPOS> and <http://code.google.com/p/jposee>.

 **Note:** We are moving from Subversion to Git, the jPOS project is already on Git <sup>1</sup> but jPOS-EE remains in Subversion <sup>2</sup> for a little while. At some point, we'll migrate jPOS-EE to Git too.

Stay tuned ( [Mailing lists](#) on page 14).

<sup>1</sup> Hosted at Github.com

<sup>2</sup> Hosted at Google Code

## jPOS via Git

Instructions to get the latest version of jPOS using Subversion

Use a git client to clone the jPOS project using the following steps:

1. `git clone git://github.com/jpos/jPOS.git`
2. Open a shell and change your working directory to the newly created 'jPOS' directory.
3. Call `mvn package` in order to build jPOS
4. `cd jpos/target/jpos-x-x-x`
5. `java -jar jpos-x-x-x.jar`

You should see output like this:

```
<log realm="Q2.system" at="Thu Dec 09 09:45:47 UYST 2010.728" lifespan="6ms">
  <info>
    Q2 started, deployDir=/Users/apr/git/jpos/jpos/target/jpos-1.8.0/deploy

    jPOS 1.8.0 ff063a4 (20101209_0945)

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

jPOS Community Edition, licensed under GNU AGPL v3.0.
This software is probably not suitable for commercial use.
Please see http://jpos.org/license for details.

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.9 (Darwin)

iQEcBAEBAgAGBQJMo1HDAAoJEOQyeO71nYtFv74H/3OgehDGEy1VXp2U3/GcAobg
HH2eZjPUz53r38ARPiU3pzm9LwDa3WZgJJaa/b9VrJwKvbPwe9+0kY3gScDE1skT
ladHt+KHHmGQArEutkzHlpZa73RbroFEIalqmN6MaDEHGoxZqDh0Sv2cpvOaVYGO
St8ZaddLBPC17bsjAPWo9sWbvL7FgPFOHhnPmbeux8SLtnfWxXWsgo5hLBanKmO1
lz+I/w/6DL6ZYZU6bAJUk+eyVVImJqw0x3IEE1I07Nh9MC6BA4iJ77ejobj8HI2r
q9u1RPEqH9NR796191NKVUkE206dV1Xo7xHmJS1QZy5v/GT66xBxyDVfTduPFXk=
=oP+v
-----END PGP SIGNATURE-----

  </info>
</log>
<log realm="Q2.system" at="Thu Dec 09 09:45:47 UYST 2010.735">
  <info>
    deploy:/opt/jpos/jpos/target/jpos-1.8.0/deploy/99_sysmon.xml
  </info>
</log>
...
...

```

## jPOS via standard download

Instructions to get the latest stable release via standard download

Use your favorite browser to visit:

1. <http://jpos.org/download>
2. After completing the form there, you will be provided with download instructions.
3. Once you complete your download, unzip/untar your `jpos-x.x.x.zip/tgz` file into a new directory, open a shell, `cd` there and,

4. call `ant clean run` in order to verify your setup

You should see output like this:

```

    ...
    ...
run:
 [java] <log realm="Q2.system" at="Tue Jul 21 09:41:51 UYT 2009.168">
 [java]   <info>
 [java]     Q2 started, deployDir=/opt/jpos6/build/deploy
 [java]   </info>
 [java] </log>
    ...
    ...

```

### jPOS-EE via subversion

Instructions to get the latest version of jPOS-EE using Subversion

Use your favorite subversion client to perform the following steps:

1. `svn checkout http://jposee.googlecode.com/svn/trunk/ jposee`
2. Open a shell and change your working directory to the newly created 'jposee' directory.
3. Call `ant clean run` in order to verify your setup

You should see output like this:

```

    ...
    ...
<log realm="Q2.system" at="Tue Sep 08 14:51:49 PDT 2009.168">
  <info>
    Q2 started, deployDir=/opt/jposee/build/deploy
  </info>
</log>
    ...
    ...

```

Since the default jPOS-EE configuration deploys the SystemMonitor QBean (see `modules/system/deploy/99_sysmon.xml`), you will also see an initial SystemMonitor log event.

```

    ...
    ...
<log realm="org.jpos.q2.qbean.SystemMonitor" at="Tue Sep 08 14:51:48 PDT
2009.230">
  <info>
    Starting SystemMonitor
  </info>
</log>
<log realm="org.jpos.q2.qbean.SystemMonitor" at="Tue Sep 08 14:51:48 PDT
2009.231">
  <info>
    <release>1.6.5 $Revision: 2771 $</release>
    <memory>
      freeMemory=4640568
      totalMemory=5177344
      inUseMemory=536776
    </memory>
  </info>
</log>

```



```
</memory>
<threads>
  delay=0 ms
  threads=4
  Thread[Reference Handler,10,system]
  Thread[Finalizer,8,system]
  Thread[Signal Dispatcher,9,system]
  Thread[Thread-0,5,main]
  Thread[DestroyJavaVM,5,main]
  Thread[Timer-0,5,main]
  Thread[SystemMonitor,5,main]
</threads>
--- name-registrar ---
  logger.Q2: org.jpos.util.Logger
</info>
</log>
...
...
```



---

# Chapter

# 2

---

## Build System

---

### Topics:

- [jPOS Directory Structure](#)
- [Module Directory Structure](#)
- [Ant Targets](#)
- [IDE support](#)
- [Running jPOS](#)

The jPOS Project build system uses [Apache's Ant](#)<sup>3</sup> to combine multiple modules into a single application.

---

<sup>3</sup> Version 1.7.1 or later

## jPOS Directory Structure

<code>jpos6</code>	The root directory
<code>-- COPYRIGHT</code>	Copyright notice
<code>-- CREDITS</code>	Due credit to jPOS contributors
<code>-- LICENSE</code>	License information
<code>-- README</code>	The README file
<code>-- bin</code>	Unix based shell scripts
<code>-- q2</code>	Starts Q2 (same as <code>java -jar jpos.jar</code> )
<code>-- bsh</code>	Starts a BeanShell <sup>4</sup> Interpreter with the jPOS classpath.
<code>-- build.xml</code>	Main Ant tasks configuration
<code>-- build.properties</code>	Additional Ant properties used by <code>build.xml</code>
<code>-- lib</code>	Support libraries required to build jPOS
<code>-- bsh-2.0b4.jar</code>	BeanShell
<code>-- jdbm-1.0.jar</code>	JDBM
<code>-- jdom-1.0.jar</code>	JDOM
...	Additional support libraries
<code>-- modules</code>	Modules directory
<code>-- include.list</code>	Optional list of modules to include at compile time
<code>-- exclude.list</code>	Optional list of modules to exclude at compile time
<code>-- jpos</code>	Main jPOS module
<code>-- q2</code>	The Q2 System
<code>-- q2mod_jpos</code>	Set of Q2 wrappers for jPOS components (jPOS adapters)
<code>-- security</code>	<code>org.jpos.security</code> related classes
<code>-- txnmgr</code>	jPOS Transaction Manager
!   <code>-- compat_1_5_2</code>	Compatibility classes from jPOS 1.5.2 <sup>5</sup>

## Module Directory Structure

jPOS modules can have zero or more directories using the following naming convention.

<code>src</code>	The source code for this module
<code>lib</code>	Supporting libraries (jars and dynamic libraries)
<code>deploy</code>	QBean descriptors
<code>cfg</code>	Configuration files
<code>webapps</code>	exploded WAR
<code>test</code>	Unit tests

<sup>4</sup> Lightweight Scripting for Java - <http://www.beanshell.org>

<sup>5</sup> To be deprecated

The jPOS ant-based build subsystem **merges** these directories off the available modules and **flattens** them inside the automatically created **build** directory.

If we have the files

```
module/mymoduleA/src/org/jpos/MyClassA.java
module/mymoduleA/lib/mydepl.jar
```

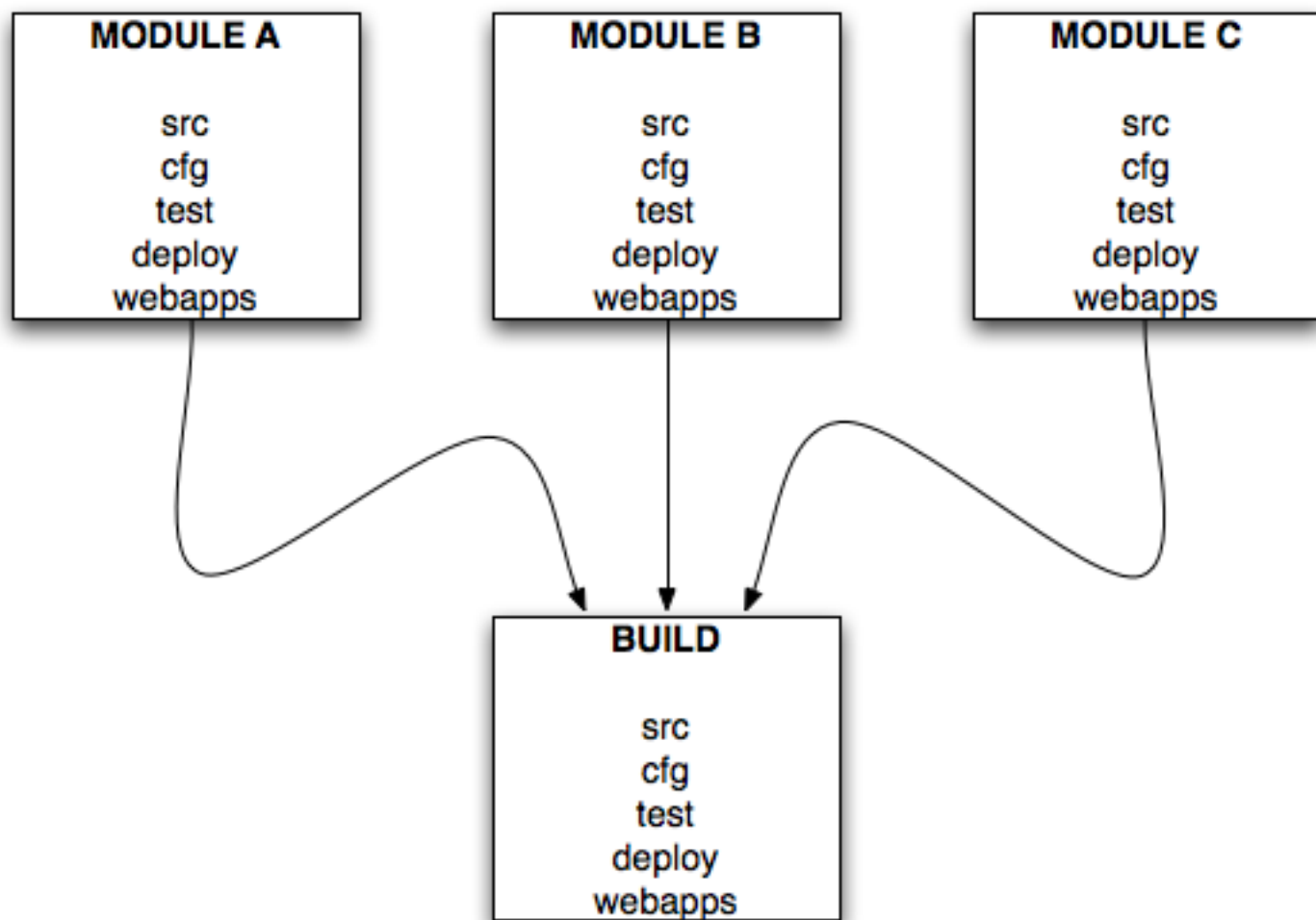
and

```
module/mymoduleB/src/org/jpos/MyClassB.java
module/mymoduleB/lib/mydep2.jar
```

and we call ant in order to build a jPOS (or jPOS-EE) application, as a first step ant will copy the files to the build/src directory and build/lib directory in order to end up with a directory structure like this:

```
build/src/org/jpos/MyClassA.java
build/src/org/jpos/MyClassB.java
build/lib/mydepl.jar
build/lib/mydep2.jar
```

The same goes for other directories such as cfg, deploy, webapps, etc.



**Figure 1: jPOS build merge process**

👉 **Note:** jPOS-EE uses a similar merge process. The main difference is that in jPOS-EE, jPOS core becomes just one pre-compiled module

The typical jPOS-EE application has a directory `modules/jpos` with just a `lib` directory containing a precompiled `jpos.jar` along with its supporting jars.

As of jPOS-EE r171, the `modules/jpos/lib` directory looks like this:

```

jposee/trunk/modules/jpos/lib$ ls -l
total 2912
-rw-r--r--  1 apr  apr   281694 Oct  9  2007 bsh-2.0b4.jar
-rw-r--r--  1 apr  apr    36174 Oct  9  2007 commons-cli.jar
-rw-r--r--  1 apr  wheel  31731 May 14  2008 cweb-extser-0.1-b2-dev.jar
-rw-r--r--  1 apr  apr   207499 Dec  5  2008 jdbm-1.0.jar
-rw-r--r--  1 apr  apr   153253 Oct  9  2007 jdom-1.0.jar
-rw-r--r--  1 apr  apr    61125 Oct  9  2007 jline-0.9.91.jar
-rw-r--r--  1 apr  apr   708947 Sep  9 18:04 jpos.jar
  
```

## Ant Targets

Calling `ant -projecthelp` provides information about the available targets:

```

Buildfile: build.xml

Main targets:

clean      clean up build and dist directories
compile    compile all
dist       Builds source distribution
jar        creates jpos.jar
javadoc    generate javadocs
run        invoke jPOS under Ant's JVM
singlejar  creates a single jpos.jar including all supporting jars
srcjar     creates jpos-src.jar for source browsing and debugging in
           IDEs like Eclipse
test       perform unit tests
zip        create zip distribution
Default target: compile

```

Installed modules can in turn have their own build.xml in order to perform module-specific tasks at build time.

## IDE support

jPOS is IDE-agnostic. If you want to use it in your favorite IDE we recommend that you create the `jpos.jar` using `ant` once and then add that jar as a library.

In addition, you can call `ant srcjar` to create a `jpos-src.jar` that you can drop into your IDE source classpath.

The same holds true for jPOS-EE, you can install `jpos-ee.jar` as well as `jpos-ee-src.jar` in your IDE.

## Running jPOS

Depending on how you get to use jPOS, you can consider it a Library, a Framework or an Application.

While using it as a library, running jPOS boils down to running your application, which in turns will use jPOS.

When you use it as an Application/Framework what you are actually running is **Q2**, jPOS' JMX Micro-Kernel.

Running Q2 is as simple as calling:


```
java -jar jpos.jar
```

Once started, Q2 will produce output like this:

```

<log realm="Q2.system" at="Sat Sep 19 19:51:39 UYT 2009.386">
  <info>
    Q2 started, deployDir=/opt/jpos/deploy
  </info>
</log>

```

 **Note:** You can run Q2 off the "build" directory (for testing purposes) but you want to move it to another destination for a production setup.

You may want to investigate the `bin` directory where you'll find a unix shell script -- that you can run in Windows inside Cygwin or you can use as a sample to create your own `.BAT` file -- called `bin/q2` that can be used to start Q2.

In addition, there's an Ant task called "run" that you can use to run Q2 under Ant's control. See `ant -projecthelp` for details.





---

# Part II

---

## jPOS

---

**Topics:**

- [Q2](#)
- [jPOS Spaces](#)

jPOS is a Java® platform-based, mission-critical, ISO-8583 based financial transaction library/framework that can be customized and extended in order to implement financial interchanges.



---

# Chapter

# 3

---

## Q2

---

### Topics:

- [Running Q2 in Standalone mode](#)
- [Running Q2 in Embedded mode](#)
- [Q2 Command line arguments](#)
- [QBean Descriptors](#)
- [QBean Life cycle](#)
- [QBean Loggeable](#)
- [QBean Configurable](#)
- [QBean Optional Attributes](#)
- [QBeanSupport](#)

**Q2** is jPOS' component assembler / JMX Micro-Kernel.

It provides an extremely simple life-cycle to its components (interchangeable called **QBeans** or **Q2 Services**). It allows to easily configure, start, and connect jPOS standard components as well as user defined ones.

Among Q2's salient features are its ability to **hot deploy** and manage its services via **JMX**, **dynamic [remote] classloading** and **push** as well as **pull** configuration.

**Q2** reads the 'deploy' directory looking for configuration files (called 'QBean descriptors') and instantiates 'services' (called 'QBeans'). Q2 provides a very simple life-cycle to its QBeans: `init`, `start`, `stop`, `destroy` and takes care of registering the newly created Object with the jPOS [NameRegistrar](#) as well as the JMX MBeanServer.

QBeans are MBeans (see JMX specs) that implement the Q2's lifecycle set of operations as defined in the [QBean](#) interface.

## Running Q2 in Standalone mode

We recommend using jPOS by launching Q2 in standalone mode. That's the way we use it in our production applications and the way most jPOS mission critical applications operate.


Running Q2 in standalone mode is as simple as calling:

1. `java -jar jpos.jar`, or if you are running jPOS-EE
2. `java -jar jpos-ee.jar`

There's a handy Unix shell script called `bin/q2` that you can use to start Q2, or `bin/start` and `bin/stop` in order to run it as a daemon service.

If you are running in a Windows environment, you can use jPOS-EE's `wrapper` module in order to run as an NT Service.

## Running Q2 in Embedded mode

 **Important:** While jPOS' Q2 can be embedded in other Java applications, we strongly recommend to run it in *standalone mode*. Q2 handles its own MBeanServer and creates several Threads in order to run its services and can have conflicts with some controlled/sandboxed environments.

In order to start Q2 from your application, you just need to:

1. `import org.jpos.q2.Q2;`
2. `Q2 q2 = new Q2 ("/path/to/your/deploy/directory);`
3. `q2.start();`

As a result, Q2 starts running and monitoring the `deploy` directory.

Q2 can be stopped using the command `q2.stop()`.

## Q2 Command line arguments

`bin/q2 --help` provides the following output:

```
usage: Q2
  -c,--command <arg>      Command to execute
  -C,--config <arg>       Configuration bundle
  -d,--deploydir <arg>    Deployment directory
  -e,--encrypt <arg>      Encrypt configuration bundle
  -h,--help                Usage information
  -i,--cli                 Command Line Interface
  -r,--recursive           Deploy subdirectories recursively
  -v,--version             Q2's version
```

Q2 uses `commons-cli` which supports POSIX-like as well as GNU long options.

**Table 3: Command Line Arguments**

Argument	Description
-c, --command	A CLI command to execute. 'CLI' is Q2's Command Line Interface
-C, --config	Deploy a single file (called a Configuration Bundle) containing a set of QBean descriptors
-d, --deploydir	Path to the deploy directory (defaults to 'deploy' in the current working directory)
-e, --encrypt	It is possible to encrypt an MBean deployment descriptor in order to protect it from an occasional lurker
-h, --help	dumps help screen
-i, --cli	launches interactive command line interface (press 'tab' to see available commands)
-r, --recursive	recursively scans for QBean descriptors in the deploy children directories
-v, --version	shows current jPOS version and revision

## QBean Descriptors

This section describes the anatomy of a QBean descriptor.

Q2 regularly scans the 'deploy' directory, and eventually its subdirectories, looking for QBean descriptors.


Those are simple and flexible XML files with the following requirements:

- The outer element has to either have an unique name, or it needs an unique 'name' attribute.
- Unless the outer element is a well know name defined in the `QFactory.properties` ResourceBundle, a `class` attribute has to be provided in order to specify the QBean implementation classname.

```
<qbean name="mybean" class="org.jpos.test.MyBean" />
```

In the previous example, an instance of `org.jpos.test.MyBean` would be created and registered with the `MBeanServer` as well as the `NameRegistrar` under the name `mybean`.

If we omit the 'name' attribute, then the root element name (in this case `qbean`) would be used instead.

 **Attention:** Names, either picked from the root element or the 'name' attribute have to be unique across a Q2 instance.

In addition to the `class` and `name` attributes, **Q2** supports two additional attributes:

- `logger`
- `realm`

i.e.:

```
<qbean name="mybean" class="org.jpos.test.MyBean" logger="Q2" realm="mybean" />
```

The ResourceBundle `QFactory.properties` provides some handy shortcuts for some well know stock QBeans, namely:

```
shutdown=org.jpos.q2.qbean.Shutdown
script=org.jpos.q2.qbean.BSH
jython=org.jpos.q2.qbean.Jython
```

```
spacelet=org.jpos.q2.qbean.SpaceLet
sysmon=org.jpos.q2.qbean.SystemMonitor
txnmgr=org.jpos.transaction.TransactionManager
transaction-manager=org.jpos.transaction.TransactionManager
qmux=org.jpos.q2.iso.QMUX
channel-adaptor=org.jpos.q2.iso.ChannelAdaptor
```

This allows you to use a configuration like this:

```
<script name="myscript">
  ...
  ...
</script>
```

Without having to specify:

```
<script name="myscript" class="org.jpos.q2.qbean.BSH">
  ...
  ...
</script>
```



**Caution:** Although you can omit the 'name' attribute, it is good practice to use it, otherwise, you'd have a name conflict on your second descriptor using the same shortcut.



**Tip:** shutdown is a useful shortcut, a simple descriptor with a single element called shutdown will initiate a Q2 clean shutdown, stopping all deployed services. Look at `bin/stop` for an example.

## QBean Life cycle

Q2 provides a very simple life cycle to the **QBeans**, it basically calls four methods:

- `init`
- `start`
- `stop`
- `destroy`

Writing a QBean is as easy as implementing the [org.jpos.q2.QBean](#) interface which in addition to the aforementioned life cycle operations has a couple of additional

```
int getState(); String
getStateAsString();
```

methods required for housekeeping and JMX-based monitoring.

### A Minimal QBean implementation



**Attention:** Actually a minimal QBean implementation that **DOESN'T WORK**, please keep reading, there's a minor final tweak.

```
package org.jpos.q2.qbean;

import org.jpos.q2.QBean;

public class Test implements QBean {
    int state;
    public void init () throws Exception {
        state = QBean.STARTING;
        System.out.println ("init");
    }
}
```

```

    }
    public void start () throws Exception {
        state = QBean.STARTED;
        System.out.println ("start");
    }

    public void stop () throws Exception {
        state = QBean.STOPPING;
        System.out.println ("stop");
    }


    public void destroy () throws Exception {
        state = QBean.STOPPED;
        System.out.println ("destroy");
    }
    public int getState () {
        return state;
    }
    public String getStateAsString () {
        return state >= 0 ? stateString[state] : "Unknown";
    }
}

```

In order to deploy our Test QBean, we need to place an XML descriptor, which is just a simple XML file in the deploy directory (i.e.: 10\_test.xml)

```
<qbean name="test" class="org.jpos.q2.qbean.Test" />
```

When Q2 detects the new XML descriptor, it attempts to instantiate the service, and this is what you'll see (several errors).

 **Note:** We believe that showing you these errors is good so you understand how JMX gets into play.


First, we'll see that Q2 is trying to deploy your new QBean:

```

<log realm="Q2.system" at="Sun Nov 15 19:17:06 UYST 2009.73">
  <info>
    deploy:/opt/jpos/build/deploy/10_test.xml
  </info>
</log>

```

Then a noisy error will basically tell us that:

 **Attention:** Class org.jpos.q2.qbean.Test is **not a JMX compliant MBean**

The full message looks like this:

```

<log realm="Q2.system" at="Sun Nov 15 19:17:06 UYST 2009.77">
  <warn>
    deploy
    <exception name="MBean class org.jpos.q2.qbean.Test does not implement
DynamicMBean,
    neither follows the Standard MBean conventions
    ...
    ...
    ...
  </exception>

```

```
</warn>
</log>
```

Then we'll see something useful, when **Q2** detects a problem with a descriptor, it renames it either to a file name ending in `.BAD`, or a file name ending in `.DUP` if the problem is related to a duplicate QBean name.

```
<log realm="Q2.system" at="Sun Nov 15 19:17:06 UYST 2009.83">
  <warn>
    Tidying /opt/jpos/deploy/10_test.xml out of the way, by adding .BAD
    It will be called: /opt/jpos/deploy/10_test.xml.BAD see log above
    for detail of problem.
  </warn>
</log>
```

### Fixing the problem with the minimal QBean implementation

**QBeans** are also **JMX MBeans**, so in addition to implementing `org.jpos.q2.QBean`, our `org.jpos.q2.qbean.Test` class has to implement an interface called `org.jpos.q2.qbean.TestMBean` which in turn has to extend `org.jpos.q2.QBean`.

So our `org.jpos.q2.qbean.TestMBean` will look like this:

```
package org.jpos.q2.qbean;

import org.jpos.q2.QBean;

public interface TestMBean extends QBean { }
```

and our `org.jpos.q2.qbean.Test` class, instead of implementing `QBean`, it will implement our `TestMBean`. The definition will look like this:


```
package org.jpos.q2.qbean;
import org.jpos.q2.QBean;
public class Test implements TestMBean {
    ...
    ...
}
```

The reason for the previous requirement is that **Q2** instantiates QBean implementations using the **JMX MBeanServer**, and it calls the life cycle operations `init`, `start`, `stop` and `destroy` **via JMX**.

## QBean Loggeable

if a QBean implements the *Loggeable* interface, and its QBean descriptor has a `logger` attribute, **Q2** will take care of calling its `setLogger` method, i.e:

```
<qbean name="my-qbean"
      class="org.jpos.q2.qbean.Test"
      logger="Q2" realm="Test" />
```

 **Tip:** **Q2** is the default logger name, usually defined in the `00_logger.xml` QBean descriptor.



## QBean Configurable


if a QBean implements the *Configurable* interface, and its QBean descriptor has `property` entries, Q2 will create a Configuration object and call its `setConfiguration` method, i.e:

```
<qbean name="my-qbean"
  class="org.jpos.q2.qbean.Test"
  logger="Q2" realm="Test">
  <property name="my-prop-1" value="1" />
  <property name="my-prop-2" value="3" />
  <property file="my-prop-file.cfg" />
</qbean>
```

## QBean Optional Attributes

Before calling its 'init' method, Q2 attempts to push several well known attributes via JMX:

**Table 4: QBean Optional Attributes**

Attribute	Description
logger	If the QBean descriptor has a 'logger' attribute, and the the QBean implementation has a void <code>setLogger (Logger)</code> method, then Q2 will set a logger.
realm	Same goes for 'realm', if the QBean descriptor has a 'realm' attribute and the QBean implementation has a void <code>setRealm (String)</code> method, Q2 will call it and set the appropriate realm.
server	Ideally, and in order to honor IoC <sup>6</sup> , QBean implementations doesn't need a reference to their server, but just in case we need it, if the QBean implementation has a void <code>setServer (Q2 server)</code> method, Q2 will push a reference to itself.
persist	If a QBean implementation has a void <code>setPersist (Element e)</code> method, Q2 will push the QBean descriptor represented by an <code>org.jdom.Element</code> Object.  <b>Note:</b> The name 'persist' is not a very good choice, it was named after the fact that this Element object basically represent the persistent representation of a QBean, including its attributes and properties. We definitely need a better name for this.

## QBeanSupport

In *QBean Life cycle* on page 30 example, we have implemented a QBean from scratch, but we could have extended `org.jpos.q2.QBeanSupport` instead.

QBeanSupport implements a QBean and expose its lifecycle operations as `protected` methods:

- `initService`
- `startService`
- `stopService`
- `destroyService`

<sup>6</sup> Inversion of Control

When extending `QBeanSupport`, our `QTest` would look like this:

```
package org.jpos.q2.qbean;

import org.jpos.q2.QBeanSupport;

public class Test extends QBeanSupport {
    protected void initService () throws Exception {
        System.out.println ("init-service");
    }
    protected void startService () throws Exception {
        System.out.println ("start-service");
    }

    protected void stopService () throws Exception {
        System.out.println ("stop-service");
    }

    protected void destroyService () throws Exception {
        System.out.println ("destroy-service");
    }
}
```

In this case, we've overridden all lifecycle methods, but we could have omitted any of those as `QBeanSupport` provides default implementations.

Using `QBeanSupport` as a helper base class gives us several benefits as it implements the [Loggeable](#), [Configurable](#), obviously the [QBean](#), [QPersist](#) and has a general purpose [QBeanSupportMBean](#).

In addition, the `initService`, `startService`, `stopService` and `destroyService` set of protected methods take care of catching and logging any possible exception that your implementation may throw and keeps track of the QBean's "state" (STARTING / STARTED / STOPPING / STOPPED / DESTROYED / FAILED) for you.

By extending `QBeanSupport` you don't have to implement all the QBean lifecycle callbacks, just the ones you need.

If for example, we want our QBean to be `Runnable`, we just need to implement the `Runnable` interface and start it in a new `Thread` at `startService` time, i.e.:

```
package org.jpos.q2.qbean;

import org.jpos.q2.QBeanSupport;

public class Test extends QBeanSupport implements Runnable {
    protected void startService () throws Exception {
        new Thread(this).start();
    }
    public void run() {
        for (int tickCount=0; running (); tickCount++) {
            getLog().info ("tick " + tickCount);
            ISOUtil.sleep (1000L);
        }
    }
}
```

The `run` method is very simple, just two things worth noting:

- We can use `getLog().info(...)` here because `QBeanSupport` creates a [Log](#) object for us.
- the `running()` method is also provided by `QBeanSupport` and returns true if the service is either in the `STARTING` or `STARTED` state.

QBeans support two kind of configuration: the so-called IoC where the container "pushes" the configuration onto the QBean and a regular "pull" method where you can query the Configuration object. (technically the second method is


still a "push" scheme as Q2 will "push" a Configuration Object onto the component at "init" time and the component can either hold a reference to the Configuration object or hold references to a subset of its properties).

If you configure your QBean using properties, the configuration would look like this:

```
<qtest class="org.jpos.test.QTest" logger="Q2">
  <property name="mypropA" value="abc" />
  <property name="mypropB" value="123" />
</qtest>
```

In order to access the properties, the code would look like this:

```
String a = cfg.get ("mypropA");
String b = cfg.get ("mypropB");
```

 **Note:** `cfg` is an `org.jpos.core.Configuration` Object either provided by `QBeanSupport` or by a QBean implementing the `Configurable` interface.

On the other hand, if you want to use attributes, you have to provide setters and getters and you need to expose those setters and getters as part of your MBean interface. The attributes based XML descriptor would look like this:

```
<qtest class="org.jpos.test.QTest" logger="Q2">
  <attr name="myLongAttribute" type="java.lang.Long">5000</attr>
</qtest>
```



---

# Chapter 4

---

## jPOS Spaces

---

### Topics:

- [Spaces Overview](#)
- [Timeouts](#)
- [SpaceFactory](#)
- [Local Spaces](#)

jPOS's Space is a general-purpose coordination component inspired after (see [The Linda Coordination Language](#)).

While jPOS's Space **is not** a Linda implementation, we highly recommend getting familiar with it in order to better understand our Space implementation.

## Spaces Overview

You can think of a jPOS' Space as some sort of **Map** where its entries are linked lists of objects and its operations are thread safe.

The Space has three basic operations:

- **void out (Object key, Object value)**

Put an object into the space. If an object under the given key already exists, the object is queued at the end of the linked list.

- **Object rd (Object key)**

Reads an object from the space. Block until one is present under the given key.

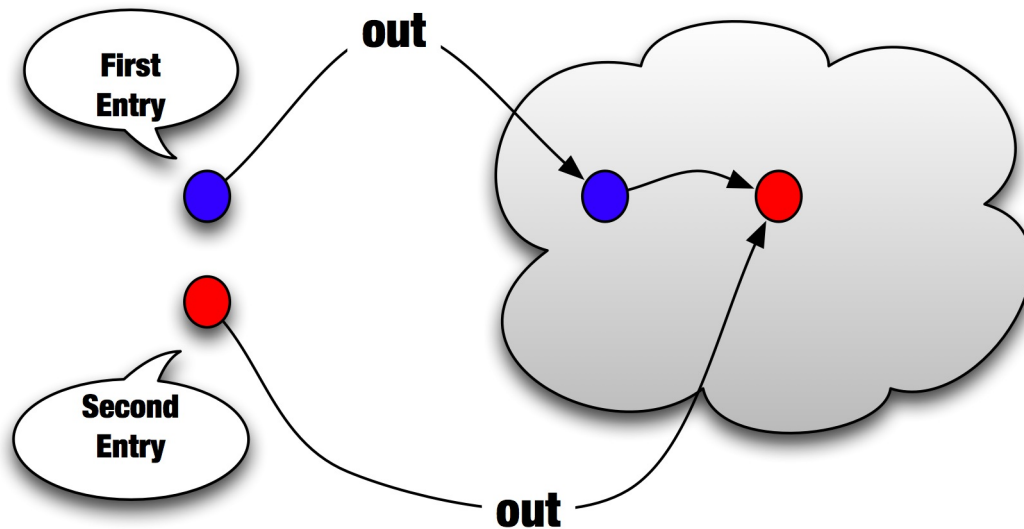
- **Object in (Object key)**

Take the object off the queue. Block until the object under the given key is present.

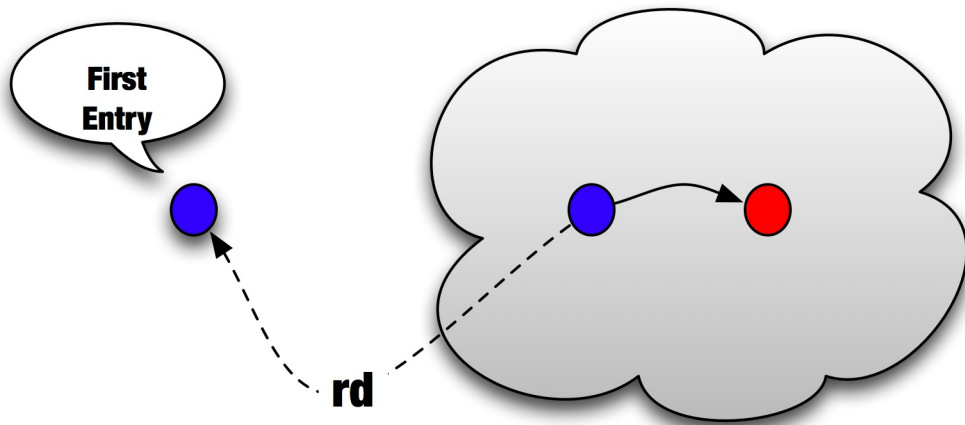
👉 **Tip:** We picked those cryptic operation names after the Linda Coordination Language, but could have used easier to remember names such as:

- **write** (instead of 'out')
- **read** (instead of 'rd')
- **take** (instead of 'in')

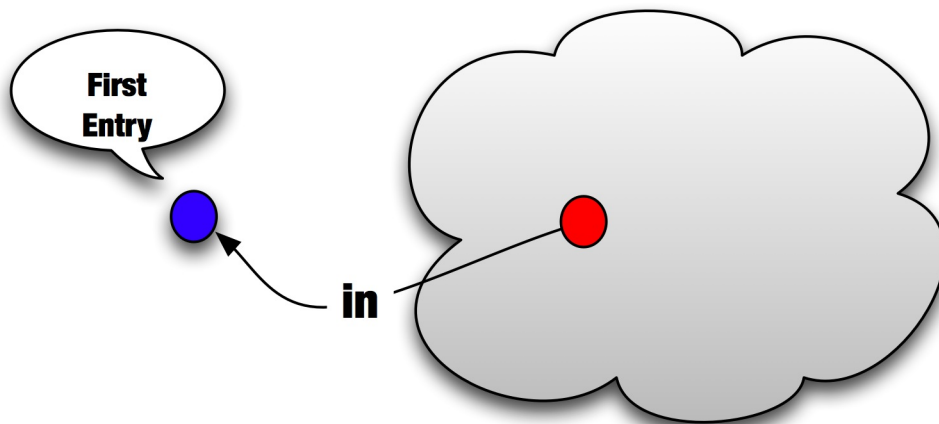
After two consecutive 'out' operations using the same 'key' value, the Space would look like this (first entry is printed as a blue circle while the second one is red):



Then an 'rd' operation would return the first entry (the blue one), without removing it from the space. The space remains with two entries for that particular key.



The 'in' operation on the other hand, takes the first entry (the blue one) off the Space, leaving the red one.



At this point, a new 'rd' operation will return the second entry (the red one) and an 'in' operation would return the red one as well, leaving the space empty (further 'rd' or 'in' operations on that particular key will block).

The following code demonstrates the previous operations:

```
import org.jpos.space.Space;
import org.jpos.space.SpaceFactory;

Space sp = SpaceFactory.getSpace(); // returns a default Space implementation
sp.out ("TEST", "The blue entry");
sp.out ("TEST", "The red entry");

sp.rd ("TEST"); // returns the 'blue' entry
sp.in ("TEST"); // returns the 'blue' entry, and takes it from the Space
sp.rd ("TEST"); // now the 'red' entry is in the queue's head
sp.in ("TEST"); // returns the 'red' entry and takes it from the Space

sp.rd ("TEST"); // or sp.in ("TEST") would block until some other thread
                // outs an entry to the "TEST" queue
```

## Timeouts

The Space operations `out`, `rd` and `in` accept an additional parameter with a timeout specified in milliseconds.

- **void out (Object key, Object value, long timeout)**

The entry will live in the space for as much as `timeout` milliseconds.

- **Object rd (Object key, long timeout)**

Reads an object from the space. Block for up to `timeout` milliseconds. If no entry with the given key is present after the timeout expires, this method returns null.

- **Object in (Object key, long timeout)**

Take the object off the queue. Block for up to `timeout` milliseconds. If no entry with the given key is present after the timeout expires, this method returns null.


A variation of the `rd` and `in` with timeout operations are `rdp` and `inp` (think 'p' for 'probe'). These methods return immediately (either the object or null).

- **Object rdp (Object key)**

Reads object with the given key if exists, otherwise return null.

- **Object inp (Object key)**

Take object with the given key from the Space, if exists, otherwise return null.

 **Tip:** For additional details, see the [Space package API documentation](#).

## SpaceFactory

The jPOS project includes an increasing number of Space implementations. Although you can use their regular constructors in order to create new instances, the recommended way to create and access existing spaces is by means of the **SpaceFactory**.

The recommended way to instantiate or get a reference to an already instantiated Space instance is by means of the [SpaceFactory](#).


The easiest way to get a reference to the space is by using

```
Space sp = SpaceFactory.getSpace();
```

that gives you a reference to the default space (TransientSpace named 'default').

This is the same as calling:

```
Space sp = SpaceFactory.getSpace("tspace:default");
```

 **Tip:** If you intend to access the default space for a given JVM, we recommend that you call `SpaceFactory.getSpace()` instead of `SpaceFactory.getSpace("tspace:default")` because from time to time, we change which is the default implementation.

We currently have the following space implementations:



**Table 5: Space Implementations**

Prefix	Classname and description
<b>tspace</b>	<code>org.jpos.space.TSpace</code> is an in-memory, fast, space implementation
<b>jdbm</b>	<code>org.jpos.space.JDBMSpace</code> is a persistent, JDBM based implementation that has been (and still is) in production at many jPOS sites
<b>je</b>	<code>org.jpos.space.JESpace</code> is a new persistent space based on Berkeley DB Java Edition. It's faster than jdbm space and can support larger spaces. We are fading away the use of 'jdbm' spaces in favor of this new 'je' ones.

In order to access a JESpace, one would use code like this:

```
Space sp = SpaceFactory.getSpace("je:myjespace");
```

this would create a directory called `myjespace` required by Berkeley DB Java Edition. If you want to place that data directory somewhere else, you could use a third argument within the space name descriptor (sometimes referred as space 'uri'), i.e:

```
Space sp = SpaceFactory.getSpace("je:myjespace:/tmp");
```

In this case, the directory `myjespace` will be created in the `/tmp` directory.

### Additional space implementations

We are always experimenting with new space implementations. We usually place them in the **jPOS-EE** project under the optional (directory `opt`) set of modules.


We currently have a **ReplicatedSpace** based on JGroups, an experimental **VoldemortSpace** based on Linked-In's Voldemort project and ideas for new implementations.

Those very special spaces are instantiated by means of a standard Q2 service, but they are registered in the NameRegistrar, so you can still access them using the SpaceFactory, as long as your code can guarantee that the special space startup service is initialized before you access it.

## Local Spaces

In addition to the `Space` interface, some space implementations also implements the `LocalSpace` interface.

The *LocalSpace* provides an event based interface to the spaces. We encourage not to use the `LocalSpace` interface whenever possible, because by using it one can't replace space implementations in the future. That said,

 **Tip:** QMUX is an example of a class that uses `LocalSpace` in order to avoid having to use additional threads.



---

**Part**

**III**

---

**jPOS-EE**

---

jPOS Extended Edition goes one step beyond jPOS by providing a development environment with a continuously growing set of proven, ready-to-deploy modules based on jPOS' best practices.



---

# Part IV

---

## HOWTO

---

### Topics:

- [Creating a jPOS-EE project](#)
- [Moving a project to production](#)

This part contains an assorted collection of jPOS and jPOS-EE related HOWTOs. We intend to 'graduate' them to full topics at some point.

## Creating a jPOS-EE project

---

Instructions to create a new jPOS-EE project

Open a terminal, create your new project's home directory, add it to your svn repository and perform an initial commit, then:

1. `svn export https://jposee.googlecode.com/svn/trunk/build.xml`
2. `svn export https://jposee.googlecode.com/svn/trunk/build.properties`
3. `svn export https://jposee.googlecode.com/svn/trunk/bin`
4. `touch devel.properties`
5. `mkdir modules`
6. `svn add build.xml build.properties devel.properties modules bin`
7. Call `svn ps svn:externals .` and add the following mappings:

- `modules/jpos http://jposee.googlecode.com/svn/trunk/modules/jpos`
- `modules/commons http://jposee.googlecode.com/svn/trunk/opt/commons`

Add optional modules from the jPOS-EE public repository as per the project requirements, either via `svn:externals` or `svn export`.

If using an IDE, you may consider adding `opt/jpos_src`. We recommend at least to export `modules/system` from `http://jposee.googlecode.com/svn/trunk/modules/system`

8. `svn commit`
9. `svn update`
10. `ant run` (stop with Ctrl-C)
11. Try an `svn status` and add `svn:ignore` in the root and 'modules' directory (you probably want to ignore 'build, dist' in the root directory, then 'include.list' and 'ignore.list' in the 'modules' directory).

The following 'create-jposee-project' shell script can be used to easily create a new jPOS-EE project. It assumes you are located inside a subversion-controlled directory.

```
#!/bin/bash
if [ "$1"x == "x" ]; then
    echo "Wrong number of parameters. Usage: $0 projectname";
    exit 1
fi

mkdir $1 || exit 1
echo "Creating directory $1"
svn add $1
svn commit -m 'Created $1 jPOS-EE project' $1
cd $1
svn export https://jposee.googlecode.com/svn/trunk/build.xml
svn export https://jposee.googlecode.com/svn/trunk/build.properties
svn export https://jposee.googlecode.com/svn/trunk/bin
touch devel.properties
mkdir modules
svn add build.xml build.properties devel.properties modules bin

svn ps svn:externals . --file - <<!
modules/jpos http://jposee.googlecode.com/svn/trunk/modules/jpos
modules/commons http://jposee.googlecode.com/svn/trunk/opt/commons
!
```

```
svn ps svn:ignore . --file - <<!
modules/jpos
modules/commons
build
dist
!
svn commit -m 'Initial export on project $1' .
svn update
svn status
```

## Moving a project to production

---

As described in [Creating a jPOS-EE project](#) on page 46 some modules are taken off jPOS-EE's trunk public repository using `svn:externals`.

Before moving to production, it is extremely important to **lock** using the `-r` switch in the `svn:externals` configuration to the version used during quality assurance.





# GNU AGPL License

---

## GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

The precise terms and conditions for copying, distribution and

modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU Affero General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to

produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

## 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

#### 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as

long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied

by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

#### 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance with third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties



receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

#### 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

#### 13. Remote Network Interaction; Use with the GNU General Public License.

Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the work with which it is combined will remain governed by version 3 of the GNU General Public License.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU Affero General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU Affero General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU Affero General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU Affero General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If your software can interact with users remotely through a computer network, you should also make sure that it provides a way for users to get its source. For example, if your program is a web application, its interface could display a "Source" link that leads users to an archive of the code. There are many ways you could offer source, and different solutions will be better for different programs; see section 13 for the specific requirements.

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU AGPL, see <http://www.gnu.org/licenses/>.



# CREDITS

---

Short reference description.

## Developers

Alejandro Pablo Revilla <apr@cs.com.uy>, Daniel Fernando Larrosa <dfic@cs.com.uy>, Victor A. Salaman <salaman@teknos.com>, Eoin Flood <eoin.flood@orbiscom.com>, Arun Kumar U <bksys@vsnl.com>, Bharavi Gade <bharavi@mchek.com>, Rajal Shah <rshah@enstage.com>, Hani S. Kirolos <h.samuel@egyptianbanks.net>, Jonathan O'Connor <Jonathan.O'Connor@xcom.de>, Cameron Young <cn\_young@atcanada.ca>, Alireza Taherkordi <taherkordi@dpi2.dpi.net.ir>, Alcarraz, Andrés <AAAlcarraz@cabal.com.uy>, Alwyn Schoeman <alwyn.schoeman@gmail.com>, Carlos Quiroz <carlos.quiroz@welho.com>, Kris Leite <yahoo@imcsoftware.com>, Luar Roji <luar@cs.com.uy>, Dave Bergert <dbergert@users.sourceforge.net>, Jeff Gordy <jgordy@users.sourceforge.net>, Thomas L. Kjeldsen <tlk@mayi.dk>, Adrian Marques <amarques@cabal.com.uy>, Anthony Schexnauldre <anthony@duocash.com>, Mark Salter <marksalter@dsl.pipex.com>, Andy Orrock <orrock@oldsallas.com>, Ansar Ansari <ansaransari@users.sourceforge.net>, Tushar Vaish <tusharvaish@users.sourceforge.net>, Manoj Krishna <manojkrishnak@users.sourceforge.net>, Vishnu Pillai <vishnu\_pillai@users.sourceforge.net>, Robert Demski <Robert.Demski@wp.pl>, John Overland <john@johnoverland.com>

## Code contributors

Mike Trank <mike@netcomsa.com>, Jonathan Easterling <Jonathan\_Easterling@s2systems.com>, Matías Salvador <mhs@sisa.com.ar>, Julien Moebis <julien.moebis@paybox.net>, Zhiyu Tang <tangzhiyu@resoft.css.com.cn>, Henry Chan <q8e192@yahoo.com>, Jose Eduardo Leon <jeduardo@cimex.com.cu>, Christopher Harris <Christopher.Harris@retail-logic.com>, Bernardinus <bbernard@cbn.net.id>, Niclas Hedhman <niclas@hedhman.org>, Stefano Arosio <stefano.ariosio@mobilmat.it>, Robert Demski <Robert.Demski@wp.pl>

## Support, Bugfixes

George McKinney <george@tantalus.com>, Vincent Green <Vincent.Greene@amo.com>, Leonard Thomas <leonard@rhinosystemsinc.com>, Bassam Al Arab <b\_arab@yahoo.com>, Murtuza Chhil <chhil@hotmail.com>, Mladen Mrkic <mmrkic@arius.co.yu>, Hitesh Sharma <h\_sha\_2000@yahoo.com>, Doron <doronf@xor-t.com>, Chris Lemper <clemper@ponicard.com>, Andrew R. Rothwell <arothwel@moneyswitch.net>, Matthew Milliss <mmilliss@moneyswitch.net>, Morgan Hankins <morganhankins@yahoo.com>, Ozzy Espailat <ozzy@zowi.com>, Alexei Gladkov <alex@abgcard.ru>, Alexander Fedorenko <afedoren@yahoo.com>, Nathan Brice McAfee <mcafeen@wam.umd.edu>, Bülent Erdemir <buler@kocbank.com.tr>, Nigel Smith <nsmith@moneyswitch.net>, Paul W. Cowan <pcowan@usatech.com>, Norbert Kroemeke <N.Kroemeke@actosoft.de>, Robbie Robinson <robbierobinson@m2-corp.com>, Sebastián Zeballos <sebaze@gmail.com>, Alan Honczar <alan@appi.com.br>, V.Goldin <vgoldin@gmail.com>, Daniel Walsh <walsh94022@gmail.com>, Alexander Yakovlev <alexander.v.yakovlev@gmail.com>, Christopher Barham <cbarham@pobox.com>, John Overland <john@johnoverland.com>, Alexander Sahler <alexander.sahler@brodos.de>, Leonardo Zabaleta <lzabaleta@uruware.com>, Björn Östmyren <bjorn.ostmyren@netgiro.com>

## Special thank you

RXTX: <http://www.rxtx.org/> Trent Jarvi <trentjarvi@yahoo.com>, JPublish: <http://jpublish.org> Florin Patrascu



# Revision History

---

Version	Date	Description
1.5	2010-11-06	Added reference to new Git repository to <i>jPOS via Git</i> on page 15.
1.4	2010-07-07	Added <i>jPOS Spaces</i> on page 37.
1.3	2009-11-15	Added <i>QBean Life cycle</i> on page 30, <i>QBean Loggeable</i> on page 32, <i>QBean Configurable</i> on page 33, <i>QBean Optional Attributes</i> on page 33 and <i>../q2/qbean-support</i> .
1.2	2009-11-04	Added jPOS part, placed Q2 chapter there
1.1	2009-09-28	Added Q2 chapter
1.0	2009-09-19	Initial revision, general project information





# Index

## A

Abstract 5  
 AGPL 49  
 Ant 20  
 Ant Targets 22

## B

Blogs 14  
 Build system 20  
 Build System 19

## C

Component Assembler 27  
 Configurable 33  
 Contributors 61  
 Credits 61

## D

Daemon 28  
 Directory Structure 20  
 Directory Structure (module) 20  
 Distribution 15  
 Download 14, 15, 16

## E

Embedded mode 28

## F

Feeds 14  
 Forums 14

## G

Git 15  
 Github 15  
 GNU 49  
 Google Code 16

## H

HOWTO 45

## J

jCard 11  
 JDBMSpace 37  
 JESpace 37

JMX Micro-Kernel 27  
 jPOS 11, 25  
 jPOS CMF 11  
 jPOS-EE 11, 43, 46  
 jPTS 11

## L

License 49  
 Life cycle 30, 33  
 Loggeable 32

## M

Mailing lists 14  
 MBeanServer 27

## N

New project 46  
 NT Service 28

## Q

Q2 27, 28  
 QBean 27  
 QBean Configurable 33  
 QBean Descriptors 29  
 QBean Life cycle 30  
 QBean Loggeable 32  
 QBean Optional Attributes 33  
 QBeanSupport 33

## R

Resources 13  
 RSS 14

## S

Service, NT 28  
 Source Control 14  
 Space 37  
 SpaceFactory 40  
 Space timeouts 40  
 Standalone mode 28  
 Subversion 14, 16

## T

Tarball 15  
 Targets 22  
 TSpace 37

