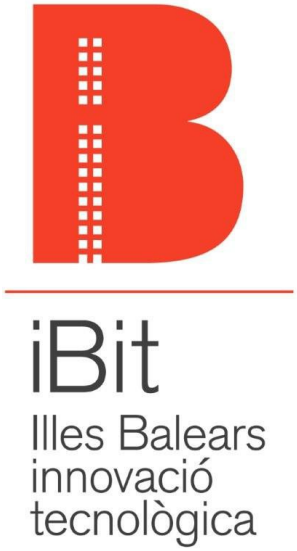


SISTRA Installation

Quick Installation Guide of SISTRA

Overview document.

Description.

Title:	Quick Installation Guide of SISTRA
State:	Draft / Approved
Version:	1.0
Author / s:	Antoni Nadal
Created:	23/11/09
Modified	21/12/11
File:	Quick_Installation_Guide_of_Sistra.odt

History of changes.

Comment:	Author / s:	Date:

Source documentary.

INDEX

1.- Introduction

This document explains how SISTRA can be installed on a JBoss server and database management system from a SISTRA binary or a SISTRA compilation code.

(A) COMPILED: instructions in Appendix I of this document should be followed.
(B) BINARY DOWNLOAD: corresponding binary files can be downloaded from SISTRA project in sourceforge. For example: next address can be accessed via Web: http://sourceforge.net/projects/sistra/files/sistra-1.1/sistra.-1.1.1/bin/. And file can be downloaded according to database we have (sistra-1.1.1_postgresql_jaas.zip sistra-1.1.1_oracle_jaas.zip for Oracle). Then, it's necessary to unzip this file in corresponding home directory.

Actions to implement a SISTRA system can be read below.

2.- Configure JBOSS

We will use an environment variable to point to our server JBoss for improving our scripts:

```
$ export JBOSS =/usr/local/jboss-3.2.8.SP1
```

2.1.- Libraries

1.- Copy commons-codec-1.3.jar file to /usr/local/jboss-3.2.8.SP1/server/default/lib directory.

```
$ Sudo cp ~ /sistra/lib/commons-codec-1.3.jar $JBOSS/server/default/lib
```

2.- Update file parsers:

```
$ Sudo cp ~ /sistra/doc/resources/xerces2.7.1/* $JBOSS/lib/endorsed/
```

3.- Update SAA library

```
$ Sudo rm $JBOSS/server/default/lib/jboss-saaj.jar
$ Sudo cp ~ /sistra/lib/saaj-api.jar $JBOSS/server/default/lib/
$ Sudo cp ~ /sistra/lib/saaj-impl.jar $JBOSS/server/default/lib/
```

4.- Download ojdbc14.jar file

Access to http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_10201.html and download ojdbc.jar file and copy it to ~ /sistra/lib and in /usr/local/jboss-3.2.8.SP1/server/default/lib/.

5. PostgreSQL JDBC Driver

If PostgreSQL is used as DBMS then download file from next address http://jdbc.postgresql.org/download/postgresql-8.4-703.jdbc3.jar and copy it into the directory of Jboss libraries: $ JBOSS/server/default/lib/.

2.2.- Configuration files

1.- Edit the $JBOSS/server/default/conf/jboss-service.xml file:

Modify line adding "deploysistra /":

CURRENT	MODIFIED
<Attribute name = "URL"> Deploy / </ Attribute>	<Attribute name = "URL"> deploy /, **deploysistra /** </ Attribute>

Then create new deploysistra directory

```
$ Sudo mkdir -p $JBOSS/server/default/deploysistra
```

2.- Indicate properties files locationdand create a new file called SISTRA-properties-service.xml in deploysistra (/usr/local/jboss-3.2.8.SP1/server/default/deploysistra) with following content:

```
<? Xml version = "1.0" encoding = "UTF-8"?>
<Server>
<MBean code = "org.jboss.varia.property.SystemPropertiesService"
name = "jboss: type = Service BootProperties name =">
<Attribute name = "Properties">
<! - In / config_sistra / SISTRA will Archives of configuration ->
ad.path.properties = / config-SISTRA /
</ Attribute>
</ MBean>
</ Server>
```

Run next command:

```
$sudo mkdir /config-sistra
$ Sudo mkdir /config-sistra/sistra
$ Sudo cp ~ /sistra/doc/resources/config/sistra/* /config-sistra/sistra
```

3.- Add DISPLAY on UNIX systems. Add following lines in $JBOSS/bin/run.sh:

```
DISPLAY =: 0.0
export DISPLAY
```

4.- Add following entry at the end of $JBOSS/server/default/deploy/jms/jbossmq-destinations-service.xml file (before </ server>)

```
<MBean code = "org.jboss.mq.server.jmx.Queue"
name = "jboss.mq.destination: service = Queue, name = AvisadorBTE">
<Depends optional-attribute-name = "DestinationManager">
jboss.mq:service=DestinationManager </ depends>
<! - 30 redelivery delay [s] ->
<Attribute name = "RedeliveryDelay"> 30000 </ attribute>
<! - Unlimited redelivery ->
```

```
<Attribute name = "RedeliveryLimit"> 3 </ attribute>
</ MBean>
```

5.- Configure Mail

(A) Copy service file:

```
$ Sudo cp ~ /sistra/doc/resources/databases/datasources/mobtratel-mailTest-
service.xml $JBOSS/server/default/deploysistra/
```

(B) Edit $ JBOSS/server/default/deploysistra/mobtratel-mailTest-service.xml file and configure it:

```
<! DOCTYPE server>
<Server>
<MBean code = "org.jboss.mail.MailService"
name = "jboss: service = MobtratelMailTest">
<Attribute name = "JNDIName"> java: /es.caib.mobtratel.mailTest </ attribute>
<Attribute name = "User">anadal@ibit.org</ Attribute>
<Attribute name = "password">password</ Attribute>
<Attribute name = "Configuration">
<Configuration>
<Property name = "mail.transport.protocol" value = "smtp" />
<Property name = "mail.smtp.host" value = "smtp.mailserver.es"/>
<Property name = "mail.from" value = "cuenta1@mailserver.es"/>
<Property name = "mail.debug" value = "false" />
<Property name = "mail.smtp.auth" value = "true" />
</ Configuration>
</ Attribute>
</ MBean>
</ Server>
```

6.- Uncomment entry in $JBOSS/server/default/conf/jboss-service.xml file

CURRENT	MODIFIED
<MBean code = "org.jboss.tm.XidFactory" name = "jboss: service = XidFactory"> <! - Attribute name = "Pad"> true </ attribute -> </ MBean>	<MBean code = "org.jboss.tm.XidFactory" name = "jboss: service = XidFactory"> <Attribute name = "Pad"> true </ attribute> </ MBean>

7.- Configure user access. Add next entry at the end of $JBOSS/server/default/conf/login-config.xml file before "</ policy>":

```
<Application-policy name = "seycon">
<Authentication>
<Login-module code = "es.caib.mock.loginModule.MockCertificateLoginModule"
flag = "sufficiant">
<Module-option name = "roleTothom"> everyone </ module-option>
</ Login-module>
<Login-module code = "es.caib.mock.loginModule.MockDatabaseLoginModule"
flag = "sufficiant">
<Module-option name = "unauthenticatedIdentity"> nobody </ module-option>
```

```
<Module-option name = "dsJndiName"> java: /es.caib.mock.loginModule.db </ module-option>
<Module-option name = "principalsQuery"> SELECT USU_PASS, USU_NOM, USU_NIF FROM
SC_WL_USUARI USU_CODI WHERE =? </ Module-option>
<Module-option name = "rolesQuery"> SELECT UGR_CODGRU, 'Roles' FROM SC_WL_USUGRU
WHERE UGR_CODUSU =? </ Module-option>
</ Login-module>
</ Authentication>
</ Application-policy>
```

8.- Modify Tomcat to enable single sign on between applications. Uncomment valve implemented by $JBOSS/server/default/deploy/jbossweb-tomcat50.sar/server.xml

```
<Valve className = "org.apache.catalina.authenticator.SingleSignOn" debug = "0"
/>
```

2.3.- Copy Binaries

1.- Place loginModuleMOCK.jar file in $JBOSS/server/default/lib directory. In addition, bouncy castle libraries for certificate handling (bcmail.jar, and bcprov.jar bctsp.jar) are also required in this directory. These libraries can be found in project directory: ~/sistra/pluginsMOCK/lib. They are necessary if test login module and test signature plug-in are used.

```
$ Sudo cp ~/sistra/pluginsMOCK/product/lib/loginModuleMOCK.jar $JBOSS/server/default/lib
$ Sudo cp ~/sistra/lib/bouncy/bcmail.jar $JBOSS/server/default/lib
$ Sudo cp ~/sistra/lib/bouncy/bcprov.jar $JBOSS/server/default/lib
$ Sudo cp ~ /sistra/lib/bouncy/bctsp.jar $JBOSS/server/default/lib
```

2.- Copy ears to deploysistra

```
$ Sudo cp ~/sistra/product/ear/1-sistra.ear
$JBOSS/server/default/deploysistra/

$ Sudo cp ~/sistra/pluginsMOCK/product/ear/2-module-pluginsMOCK.ear
$JBOSS/server/default/deploysistra/
```

2.4.- DataSources

Datasources defined at ~ /sistra/doc/resources/DB/datasources/ should be copied to /usr/local/jboss-3.2.8.SP1/server/default/deploysistra/:

- audit-ds.xml
- loginMOCK-ds.xml

- redose-ds.xml
- bantel-ds.xml
- mobtratel-ds.xml
- SISTRA-ds.xml
- form-ds.xml
- zonaper-ds.xml

```
$ Sudo cp ~/sistra/scripts/datasources/DB]/*.xml
$JBOSS/server/default/deploysistra /
```

where [DB] can be Oracle or PostgreSQL.

3.- Database management

3.- Create users and databases:

3.1.- Connect to database

```
$ Sudo bash
$ Su postgres
$ Psql -U postgres
```

3.2.- Create a user for each DB:

```
CREATE USER "audits" WITH Encrypted password 'audits' NOCREATEUSER;
CREATE USER "loginMock" WITH Encrypted password 'loginMock' NOCREATEUSER;
CREATE USER "redose" WITH Encrypted password 'redose' NOCREATEUSER;
CREATE USER "bantel" WITH Encrypted password 'bantel' NOCREATEUSER;
CREATE USER "mobtratel" WITH Encrypted password 'mobtratel' NOCREATEUSER;
CREATE USER "SISTRA" WITH Encrypted password 'SISTRA' NOCREATEUSER;
CREATE USER "form" WITH Encrypted password 'form' NOCREATEUSER;
CREATE USER "zonaper" WITH Encrypted password 'zonaper' NOCREATEUSER;
```

3.3.- Create DB

```
CREATE DATABASE "audits" WITH OWNER = audits;
CREATE DATABASE "loginMock" WITH OWNER = "loginMock";
CREATE DATABASE "redose" WITH OWNER = redose;
CREATE DATABASE "bantel" WITH OWNER = bantel;
CREATE DATABASE "mobtratel" WITH OWNER = mobtratel;
CREATE DATABASE "SISTRA" WITH OWNER = SISTRA;
CREATE DATABASE "form" WITH OWNER = form;
CREATE DATABASE "zonaper" WITH OWNER = zonaper;
```

3.4.- Out

```
\ Q
```

4.- Access permissions to the database by the user and other hosts

4.1.- /etc/postgresql/8.4/main/pg_hba.conf file must be edited and next lines added:

```
=================== ## ======================== audits
local audits       audits                  Password
host audits audits www.xxx.yyy.zzz / 24 Trust
=================== ## ======================== loginMock
local loginMock    loginMock               Password
host loginMock loginMock www.xxx.yyy.zzz / 24 Trust
=================== ## ======================== redose
```

```
local redose       redose                  Password
host redose redose www.xxx.yyy.zzz / 24 Trust
==================== ## ======================== bantel
local bantel       bantel                  Password
host bantel bantel www.xxx.yyy.zzz / 24 Trust
==================== ## ======================== mobtratel
local mobtratel    mobtratel               Password
host mobtratel mobtratel www.xxx.yyy.zzz / 24 Trust
==================== ## ======================== SISTRA
local SISTRA       SISTRA                  Password
host SISTRA SISTRA www.xxx.yyy.zzz / 24 Trust
==================== ## ======================== form
local form   form             Password
host form form www.xxx.yyy.zzz / 24 Trust
==================== ## ======================== zonaper
local zonaper      zonaper                 Password
host zonaper zonaper www.xxx.yyy.zzz / 24 Trust
```

Where www.xxx.yyy.zzz should be substituted by the three first numbers of our IP. For instance, if our IP is 192.168.121.132, then we will write this number instead of www.xxx.yyy.zzz. IP can be obtained typing ipconfig on Windows or ifconfig on Linux systems.

4.2.- Restart postgress to apply changes:

```
$ Sudo restart /etc/init.d/postgresql
```

4.3.- If you do not have access from other computers than localhost, follow recommendations found in http://www.cyberciti.biz/tips/postgres-allow-remote-access-tcp-connection.html

5.- Create and configure database (for each user and database)

5.1.- Connect to database server. If database is on the same server execute

```
$ Psql -h localhost -p 5432 -u [USER] -W -d [DB]
```

and if it is on another server then run:

```
Www.xxx.yyy.zzz $ psql -p 5432 -h -u [USER] -W -d [DB]
```

5.2.- Give permission to the user:

```
GRANT ALL PRIVILEGES ON DATABASE "[USER]" TO [DB];
GRANT ALL PRIVILEGES ON SCHEMA PUBLIC TO [USER];
```

5.3.- Import structure and table data for specific database

```
\ And ~/sistra/scripts/database/x.y/[Sgdb]/Create/sistra_[DB]_create_schema.sql
\ And ~/sistra/scripts/database/x.y/[Sgdb]/Create/sistra_[DB] _create_data.sql
```

Where [Sgdb] can be Oracle or PostgreSQL and [DB] is one of databases described above: audita, form ...

5.4.- Out:

```
\ Q
```

For example "redose" and a postgresql database of sistra 1.1.x, commands to run are next:

```
$ Psql -h localhost -p 5432 -U redose -W -d redose

GRANT ALL PRIVILEGES ON DATABASE "redose"TO redose;

GRANT ALL PRIVILEGES ON SCHEMA TO PUBLIC redose;

\ And
~/sistra/doc/resources/databases/cripts/DB/1.1/postgresql/create/sistra_redose_create_sc
hema.sql

\ And ~/sistra/scripts/database/1.1/postgresql/create/sistra_redose_create_data.sql

\ Q
```

Do the same for the remaining scripts.

6.- Give total permissions to admin user.

By default, users are in SC_WL_USUARI table of loginMock database. An administrator user is created with “admin” as username and “admin” as password (it cannot access to all existing SISTRA contexts by default).

~/sistra/doc/resources/bbddscripts/permisos_totals_per_admin.sql file contains SQL sentences to give admin user permissions to acess any sistra place. It's achieved executing previous file or sentences it includes.

7.- Create sequences. The number from sequence mane is the year, so it will be necessary to add new sequences after 2016 for next years (STR_SEQE17, STR_SEQE18 ...)

7.1.- In Sistra DB:

```
CREATE SEQUENCE INCREMENT STR_SEQE12 MinValue 1 1 9223372036854775807 MaxValue;
CREATE SEQUENCE INCREMENT STR_SEQE13 MinValue 1 1 9223372036854775807 MaxValue;
CREATE SEQUENCE INCREMENT STR_SEQE14 MinValue 1 1 9223372036854775807 MaxValue;
CREATE SEQUENCE INCREMENT STR_SEQE15 MinValue 1 1 9223372036854775807 MaxValue;
CREATE SEQUENCE INCREMENT STR_SEQE16 MinValue 1 1 9223372036854775807 MaxValue;

CREATE SEQUENCE INCREMENT STR_SEQP12 MinValue 1 1 9223372036854775807 MaxValue;
CREATE SEQUENCE INCREMENT STR_SEQP13 MinValue 1 1 9223372036854775807 MaxValue;
CREATE SEQUENCE INCREMENT STR_SEQP14 MinValue 1 1 9223372036854775807 MaxValue;
```

```
CREATE SEQUENCE INCREMENT STR_SEQP15 MinValue 1 1 9223372036854775807 MaxValue;
CREATE SEQUENCE INCREMENT STR_SEQP16 MinValue 1 1 9223372036854775807 MaxValue;
```

7.2.- In Bantel DB:

```
CREATE SEQUENCE INCREMENT BTE_SEQE12 MinValue 1 1 9223372036854775807 MaxValue;
CREATE SEQUENCE INCREMENT BTE_SEQE13 MinValue 1 1 9223372036854775807 MaxValue;
CREATE SEQUENCE INCREMENT BTE_SEQE14 MinValue 1 1 9223372036854775807 MaxValue;
CREATE SEQUENCE INCREMENT BTE_SEQE15 MinValue 1 1 9223372036854775807 MaxValue;
CREATE SEQUENCE INCREMENT BTE_SEQE16 MinValue 1 1 9223372036854775807 MaxValue;
```

8.- Modify postgresql.conf

Configure postgresql.conf file and modify "# max_prepared_transactions = 0" entry to "max_prepared_transactions = 20".

3.1.1.- Oracle PostgreSQL Functions

It's necessary to install a package called orafce containing most important Oracle SQL functions implemented by PosgreSQL because some parts of sistra are internally implemented using Oracle specific functions in SQL queries.

Aforementioned package together with installation notes can be downloaded from http://orafce.projects.postgresql.org/

Below, instructions for version 3.0.3 can be read (for newer versions read installation wiki section for this product):

1) Run following command:

```
$ Wget http://pgfoundry.org/frs/download.php/2908/orafce-3.0.3.tar.gz
$ Tar xvfz orafce-3.0.3.tar.gz
```

2) Before compiling it's necessary to install some development tools. Next commands must be executed to achive it:

```
$ Sudo apt-get install libpq-dev
$ Sudo apt-get install postgresql-server-dev-8.4
$ Sudo apt-get install bison
$ Sudo apt-get install flex
```

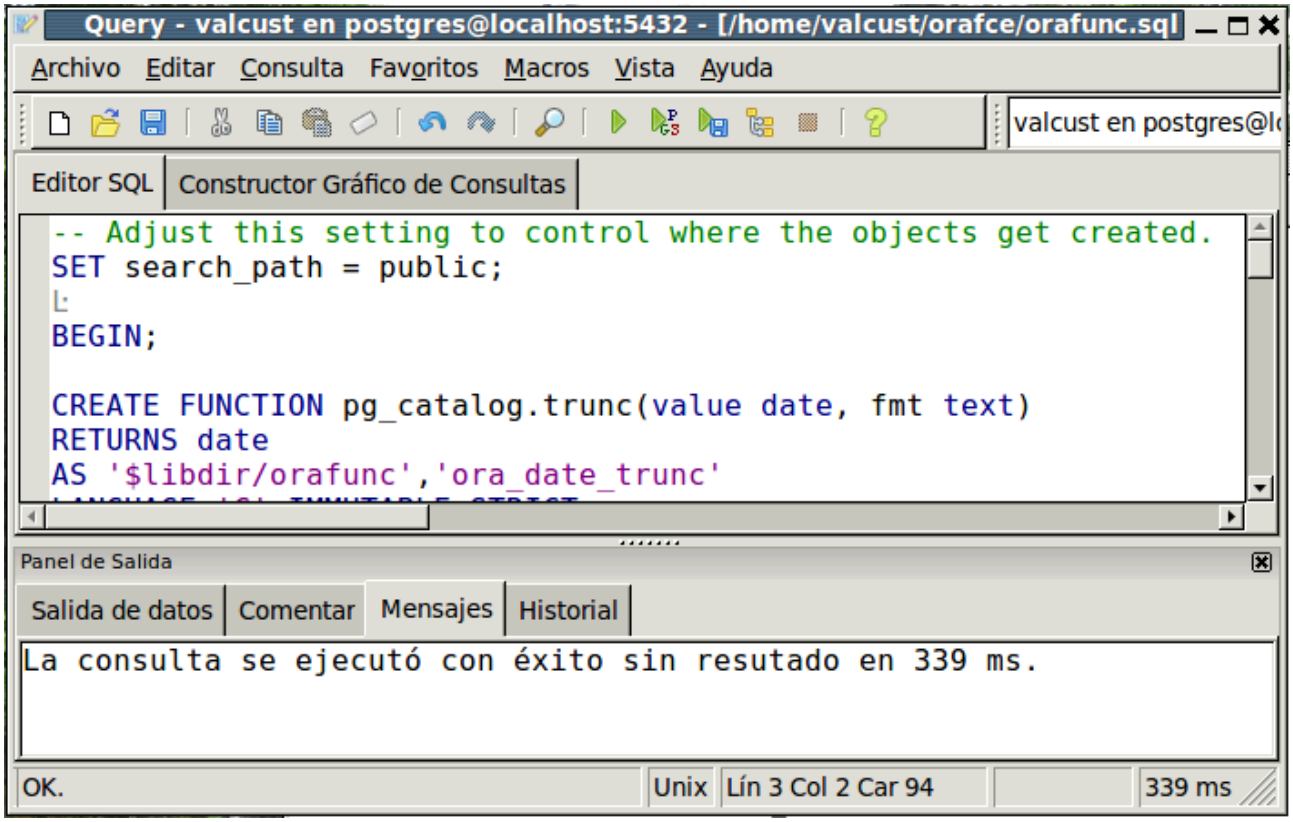
3) Go to orafce directory (cd orafce) and run following command

```
USE_PGXS make $ 1 =
$ Sudo make install USE_PGXS = 1
```

4) Restart postgresql:

```
$ Sudo restart /etc/init.d/postgresql-8.4
```

5) Open a connection to database using "pgAdmin III" with postgres user accessing to "valcust" database. Open SQL editor and load ~/orafce/orafunc.sql file. Press green button to execute sentences.



6) To verify that everything worked fine, go to "Archivo"-"Nueva Ventana" of pgAdmin III menu and type/execute next sentence:

```
select next_day (current_date, 'saturday');
```

4.- Set SISTRA

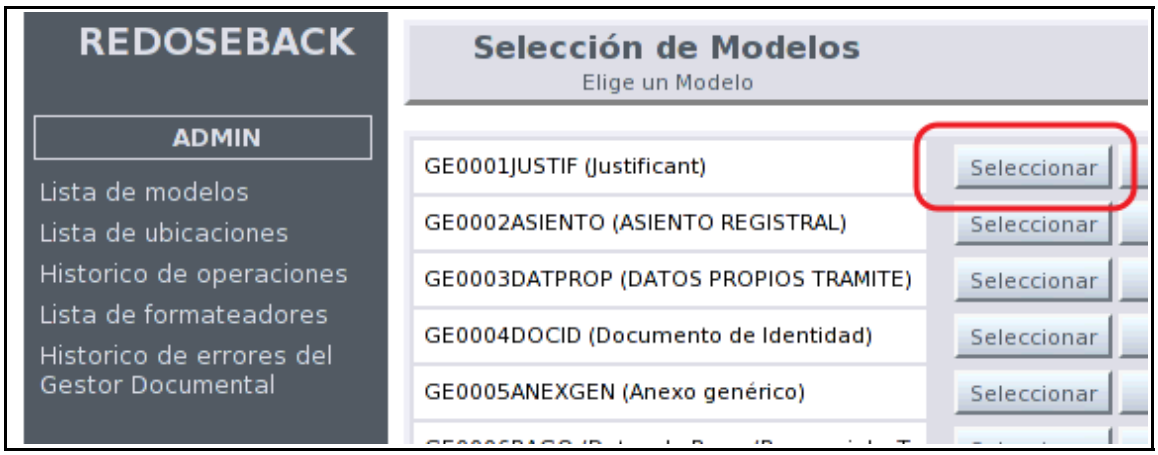
4.1.- Configuring IP's

Open /config-sistra/sistra/global.properties file and replace all "localhost" entries by Jboss addrees or public IP.

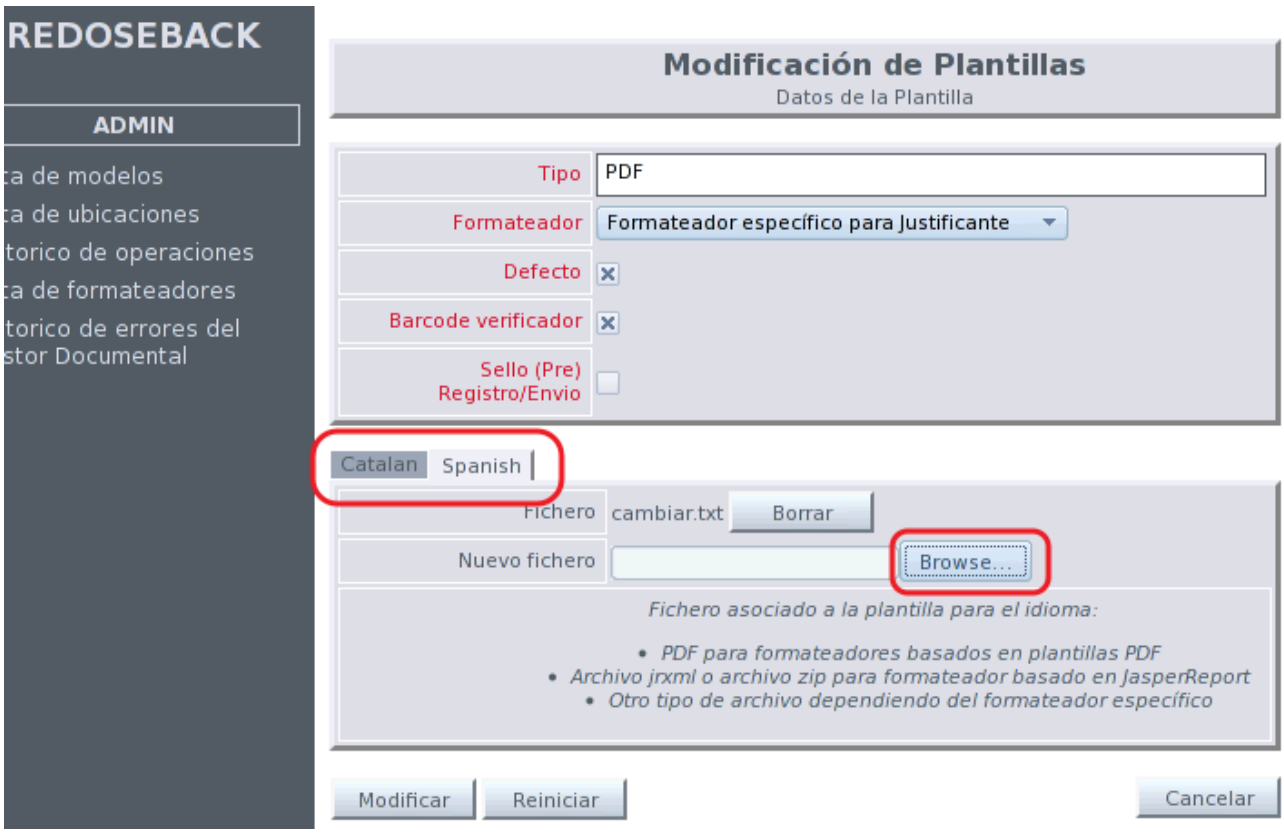
4.2.- Default Template Proof

A document, proof of process user just finished, is created when process instance is completed. Default basic configuration to create this document is found in a model called "GE0001JUSTIF (Proof)" from RedoseBack. By default, it comes preconfigured with a non valid properties file (cambiar.txt), so it must be initialized as follows:

(A) Access to RedoseBack (http://[SERVER]:8080/redoseback) and select GE0001JUSTIF model.



(B) Select version 1 and then select "Associated Templates." We see a screen like following:



(c) Two files (justificante-messages_es.properties and justificante-messages_ca.properties) are found in scripts\plantilles directory from sistra source code repository. It's necessary to include them throgh Browser button for Spanish and Catalan tab respectively.

4.3.- Additional documentation review

<< PENDING: Plug-ins, Connectors, Documentation Managers, ... >>

5.- External configurations

5.1.- Configure OpenOffice

5.1.1.- Install OpenOffice

Run next command to install OpenOffice on a Linux system:

```
# Apt-get install openoffice.org
```

5.1.2.- Start OpenOffice as a Service

We will create "openoffice" file in /etc/init.d directory with content described in "5.1.3.-Script to start OpenOffice as a Service". Next command must be executed in directory /etc/init.d/ if we want that system executes it each time it starts up:

```
$ Sudo chmod 777 /etc/init.d/openoffice
$ Chkconfig --add openoffice
```

5.1.3.- Script to start OpenOffice as a Service

```
#! / Bin / bash
### BEGIN INIT INFO
### END INIT INFO
# Openoffice.org headless server script
#
# Chkconfig: 2345 80 30
# Description: openoffice headless server script
# ProcessName: openoffice
#
# Author: Vic Vijayakumar
# Modified by Federico Ch. Tomasczik
#
OOo_HOME = / usr / bin
SOFFICE_PATH = $ OOo_HOME / soffice
PIDFILE = / var / run / openoffice-server.pid
September -e
case "$ 1" in
start)
if [-f $ PIDFILE]; Then
echo "OpenOffice headless server has already started."
sleep 5
exit
end
echo "Starting OpenOffice headless server"
$ SOFFICE_PATH -headless -nologo -nofirststartwizard -accept = "socket, host
= 127.0.0.1, port = 8100; URP" &> / dev / null 2> & 1
touch $ PIDFILE
;;
```

```
    stop)
    if [-f $ PIDFILE]; Then
    echo "Stopping OpenOffice headless server."
    killall -9 soffice && killall -9 soffice.bin
    rm -f $ PIDFILE
    exit
    end
    echo "Openoffice headless server is not running."
    exit
    ;;
    *)
    echo "Usage: $ 0 {start | stop}"
    exit 1
    ESAC
    exit 0
```

6.- APPENDIX I: Compile SISTRA from sourceforge subversion

This manual explains how to compile SISTRA application from sourceforge subversion repository. It requires java and ant. Compilation will be made in a sistra directory from user home directory (~/sistra/).

(1) Do check out executing next command from user's home:

```
$ Svn co https://sistra.svn.sourceforge.net/svnroot/sistra/sistra-core/trunk SISTRA
```

We can go down main branch (trunk) or any other branch (branch/INDRA-103-110)

(2) Download ojdbc14.jar file from http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_10201.html and copy it to ~ /sistra/lib and in /usr/local/jboss-3.2.8.SP1/server/default/lib/.

(3) Edit ~/sistra/config.properties file and select DBMS you want for hibernate: (Uncomment desired option). For Postgresql it would look like:

```
Hibernate Configuration #
# ---- PostgreSQL
hibernate.dialect = net.sf.hibernate.dialect.PostgreSQLDialect
hibernate.query.substitutions =

# ------ Oracle
# = Hibernate.dialect net.sf.hibernate.dialect.Oracle9Dialect
Hibernate.query.substitutions # 1 = true, false 0
```

(4) Go to ~/sistra/ and compile executing next command:

```
$ Ant
```

(5) Go to ~/sistra/pluginsMOCK and execute:

```
$ Ant
```