



Federal Health Architecture RHEX Direct Installation Notes

An Installation Aid for Direct Project Java based bare metal installation

Gregg Ganley

September 2012



Table of Contents

1	Introduction.....	4
2	Direct Installation Overview.....	4
3	Helpful Links	4
4	Component Overview	6
4.1.1	Assumptions.....	6
4.2	Component Installation	7
4.2.1	Linux OS setup	7
4.2.2	Obtain and Install the Assembly	7
4.2.3	Direct Configuration	7
4.2.4	Anchor cert.....	8
4.2.5	User Certificates.....	8
4.2.6	Direct Config UI	9
4.2.7	Apache James email server	10
4.2.8	DNS Responder	11
4.3	Adding a Direct User.....	12
4.4	Testing and Troubleshooting Direct.....	14
5	Appendix I – DNS Configuration	16
6	Appendix II – Direct Configuration and James database MySQL conversion.....	23



1 Introduction

The following document offers a set of Direct installation notes starting with download, installation, operation and database configuration of all aspects related to a java-based Direct Health Information Service Provider (HISP). The document records the steps that were taken as part of a pilot Direct HISP effort. Along the way, we participated in a Direct connect-athon which validated the overall workings of our installation. It is with hopeful intent that these notes can aid others in their installation of a Direct HISP.

2 Direct Installation Overview

The Direct project specifies a trusted patient health information based email system. To quote the Direct Project Overview:

“The Direct Project specifies a simple, secure, scalable, standards-based way for participants to send authenticated, encrypted health information directly to known, trusted recipients over the Internet.”

The Direct standard-based approach to exchanging email can be realized by following its specification and implementing, from the ground up, a HISP end point or effectively a dedicated email server, Domain Name Server (DNS) responder, and a certificate management component. While this approach is viable, the Direct project offers several “bare metal” installation, a Java and a .NET version. When downloaded onto a newly created Linux or Windows server, a set of detailed instructions is provided to guide one step by step.

This document describes installation notes for the Java bare metal install for the **rhex.us** domain with the ultimate goal of being able to exchange email as tjones@direct.rhex.us with other Direct HISPs.

3 Helpful Links

The Direct project has several important resource pages to aid in the installation of Direct.

This link is the top level for the implementation group:

<http://wiki.directproject.org/Reference+Implementation+Workgroup>



The java reference implementation is predominantly in use today, and its main Wiki page is here:

<http://wiki.directproject.org/Java+Reference+Implementation>

The java bare metal project – HISP is the specific component that is to be installed. It has a step-by-step guide, here:

<http://wiki.directproject.org/Bare+Metal+Project+-+Java+%28HISP+only%29>

The steps described on this page were followed explicitly and will be described in turn below. Also, along the way a few issues were found, so further instructions will be described as well.

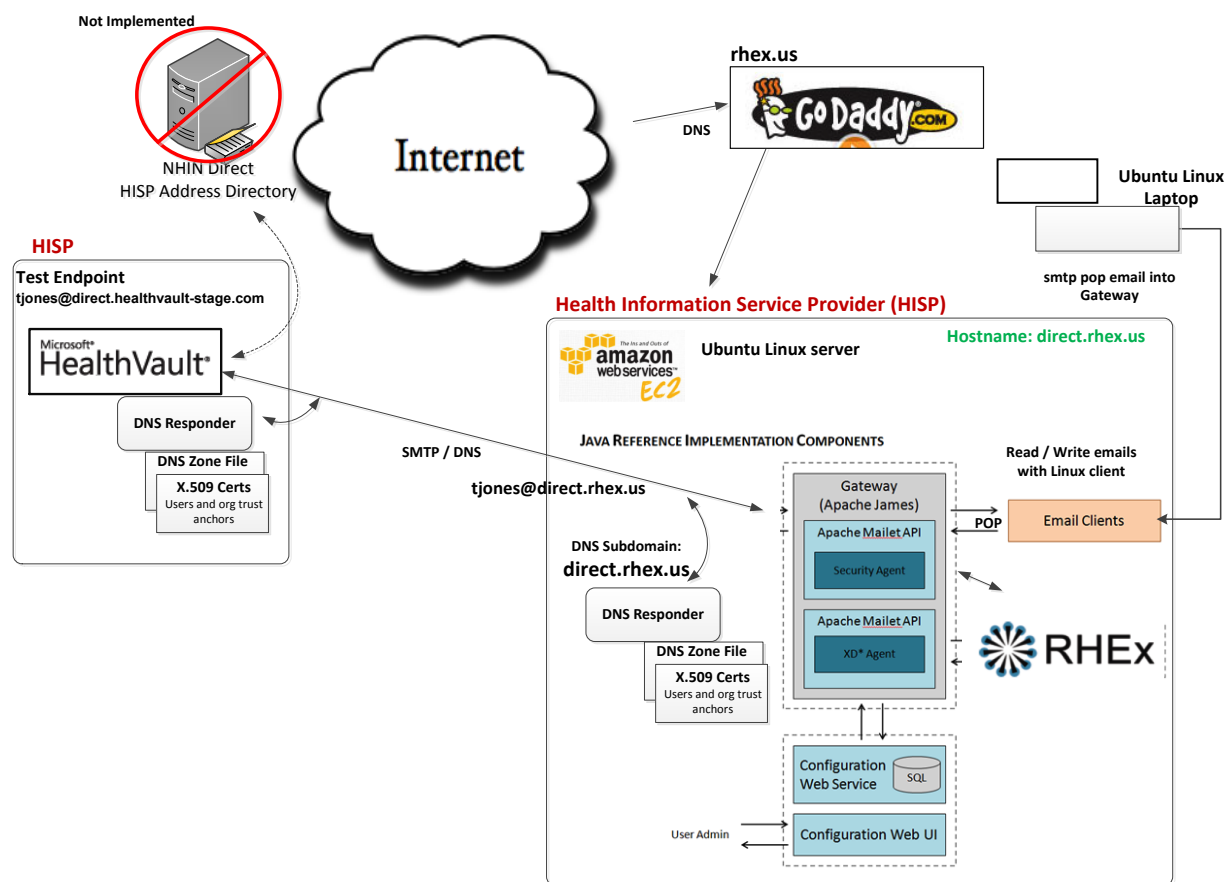
A critical and very helpful document was found at this link, which contained many more details of the Direct install:

http://api.nhindirect.org/java/site/assembly/stock/1.3.1/users-guide/depl-hisp-only.html#HISP_Only_Deployment_no_source



4 Component Overview

To begin, the diagram below describes the components needed and their relation to each other.



From a Direct perspective, the HISP contains three major components, a DNS responder, an Apache James email server, and the Direct gateway which runs as a web app on top of an Apache Tomcat JSP server.

4.1.1 Assumptions

In the Assumptions section of the guide, the installation assumes that the mail exchanger (MX) and related DNS records have been setup up prior to the installation. This is actually



a tricky and challenging area. Briefly, goDaddy.com hosts the rhex.us DNS domain. In turn, it contains DNS pointers to the Direct sub-domain direct.rhex.us which points to an Amazon EC2 virtual machine and together this machine is known as the **direct.rhex.us** HISP. For more details, I have listed individual screen shots to aid in the DNS setup. See the first Appendix of this document for a step by step description on how this was accomplished.

4.2 Component Installation

4.2.1 Linux OS setup

Java 6 SEJDK 1.6 along with JCE_1.4.2 is the specific version of Java that is needed. On Ubuntu Linux, due to a license dispute, the JDK 1.6 cannot be downloaded using the standard Ubuntu install OS methods. Instead, the site below has specific instructions on downloading and installing JDK1.6:

<https://help.ubuntu.com/community/Java>

The Java Cryptographic Extensions (JCE_1.4.2) installed without incident.

4.2.2 Obtain and Install the Assembly

I followed the download link here to secure a copy of 1.3.1:

<https://oss.sonatype.org/content/repositories/releases/org/nhind/direct-project-stock/1.3.1/direct-project-stock-1.3.1.tar.gz>

After unzipping, all installed well.

4.2.3 Direct Configuration

The Direct java code runs on top of Apache Tomcat as stated earlier. In order for the James email server and the DNS responder to function properly the Direct HISP configuration needs to be established. The configuration has three main areas, DNS, anchor and user certificates. See the appendix for the DNS configuration.



4.2.4 Anchor cert

The anchor certification establishes the public/private keys for the Direct HISP. The Direct Project reference implementation assembly ships a tool called certGen for generating root CAs and certificates for the purpose of pilots and interoperability testing.

Direct provides a simple Java UI tool to help generate the needed X.509 certificates. The tool is located here - `direct/tools/certGen.sh`

An example anchor cert:

```
CN                direct.rhex.us
Country           US
State             MA
Location          Bedford
Org               FHA
Email             < blank >
expr date:        365
key strength      1024
password:         < your password >
```

This generates two files, the public and private key certificates or better known as your HISP trust anchors.

<code>direct.rhex.usAnchorPublic.der</code>	Public key
<code>direct.rhex.usKey.der</code>	Private key

The public trust anchor file should be given to organizations that you have established a Direct trust relationship with and is required to be in place before Direct emails can be exchanged.

4.2.5 User Certificates

For each Direct email user, a user certificate must be generated and installed. Again, using the certGen.sh tool, reload your anchor cert into the tool. Next select generate leaf node. Below is an example user certificate:

```
CN                Tom Jones
Country           US
```




```
State          MA
Location       Bedford
Org            FHA
Email          tjones@direct.rhex.us
expr date:     365
key strength   1024
password       none
```

Once create is selected, three files are created:

```
tjones.der          Public certificate file
tjonesKey.der       Private certificate file
tjones.p12          A certificate which both priv/pub keys
```

The p21 file is critical and must be loaded into the Direct Config UI as described in the next steps.

4.2.6 Direct Config UI

The configuration UI is located at:

```
direct.rhex.us:8081/config-ui
```

The instructions do a fairly good job of describing the necessary steps, and in summary they are as follows:

1. Create a domain under manage domains
2. Add anchor cert
 - a. Click on sub tab under your domain (for example, direct.rhex.us)
 - b. Load the public X.509 cert (direct.rhex.usAnchorPublic.de)
3. Add user certs
 - a. Under manage certificates, add user certificates
 - b. Load the p12 file for each user (e.g. tjonesKey.p12)

Now that this is complete, it is recommended to change the password of the Direct Config admin account. The Linux steps are indicated below.

```
cp $DIRECT_HOME/apache-tomcat-7.0.23/webapps/config-ui/WEB-INF/config-
servlet.xml $DIRECT_HOME/apache-tomcat-7.0.23/webapps/config-ui/WEB-INF/config-
servlet.xml.orig
```



```
sed -i "s/adm1nD1r3ct/your_new_password/g" $DIRECT_HOME/apache-tomcat-7.0.23/webapps/config-ui/WEB-INF/config-servlet.xml
```

```
sh bin/shutdown.sh
sh bin/startup.sh
```

At this point, the Direct user configuration is complete, however the bare metal install uses an in-memory database (Derby), which means that each time the apache tomcat server is restarted the configuration above will be deleted. It is recommend that either Derby's configuration be changed to persist the DB to a file or an alternative SQL database be used. MySQL was used as the persistent store for this example, see the appendix II for details.

4.2.7 Apache James email server

The James email server is the "front-end" or internet facing component of Direct. It is responsible for the actual sending and receiving of email. It has several parameters that need configuration.

```
Set the email server domain name
cd direct/james-2.3.2
sh bin/setdomain.sh direct.rhex.us
```

Add email users

```
run james
sudo -E sh bin/run.sh > james.log 2>&1 &
```

```
telnet localhost 4555
root
root
adduser tjones < password >
```

Ensure the HISP server ports are open through the firewall:

```
ports 110 for pop
port 25 for smtp
```

Change the James root password for the email Config server:

```
Edit: james-2.3.2/apps/james/SAR-INF/config.xml
and look for management port password and change to < your root password >
```



Email client configuration settings

Email Server name: mail.direct.rhex.us

Default ports: pop 110, send port 25

Username: tjones

Password: < password >

For read and send

No SSL

Authentication type: Password

4.2.8 DNS Responder

The last component is the DNS responder. Use the following commands to start the service:

```
cd direct/DirectDNSServices/DirectDNSServer  
sudo ./bin/DirectDNSServer start
```



4.3 Adding a Direct User

```
####  
#### Direct User create steps  
  
# connect to HISP Config app  
direct.rhex.us:8081/config-ui  
  
# click on the certificates tab  
# confirm new to be created name does not exist  
  
## step outside of the config tool and create the user's cert  
## assumption - anchor has been created and is ready for use  
## certGen.sh provided by the Direct java baremetal team  
  
./certGen.sh  
  
# click on load Cert  
# load public anchor cert under "Certificate Authority File:"  
# browser for file  
    direct.rhex.usAnchorPublic.der  
# open  
  
# load private key anchor file under "private key file"  
# browser for file  
    direct.rhex.usKey.der  
# open  
  
# enter anchor password and press load  
# will get a dialog box that states "CA certificate and private key loaded successfully"  
# press ok  
  
## Now ready to make a user cert  
# press "Create Leaf Cert" button  
  
#  
CN                Tom Jones  
Country           US  
State             MA  
Location          Bedford
```



```
Org          FHA
Email        tjones@direct.rhex.us
expr date:   365
key strength 1024
password     < blank >
# do not check remaining boxes
```

```
# will get a dialog asking about blank password - press ok
# will get a SUCCESS dialog, press ok
# close all remaining certGen windows
```

```
## creates three files
```

```
tjones.der      Org certificate file
tjonesKey.der   Org certificate private key file
tjones.p12      Org certificate PKCS12 file (has both priv/pub keys)
```

```
# tjones.p12 is the file to load into the Direct Config tool
```

```
## Back in Direct Config tool web interface
```

```
# add user cert
# navigate to Certificates
# click "choose File", find the .p12 file and load
# set status to enabled
# click "add certificate"
## confirm the new user is in the list of user certificates
```

```
## Note, the "Addresses" tab under manage domains is not for users, but is the postmaster
email address(s) in case something should go wrong
```

```
##
```

```
## Add user to James email server
```

```
## ssh into direct.rhex.us box
```

```
# telnet to james email server
telnet localhost 4555
root
< password >
```

```
## add username password
```



```
adduser tjones < password >
listuser
quit
```

```
## setup email client
## ensure test computer is connected directly to the internet
```

```
###
## Send a test message !
```

4.4 Testing and Troubleshooting Direct

Direct Config log files

```
direct/tomcat/logs/catalina.out
```

This log file is critical and contains any errors that may occur when running the Direct Config server.

James email log files

```
direct/james/james.log
```

This log file is critical and contains output and errors regarding the actual transfer of mail as well as CERT resolution errors.

HISP Shutdown and restart procedures

Startup

```
cd direct/apache
bin/startup.sh

cd direct/DirectDNSServices/DirectDNSServer
sudo ./bin/DirectDNSServer start

cd direct/james
sudo -E sh bin/run.sh > james.log 2>&1 &
```

Shutdown



```
cd direct/apache  
bin/shutdown.sh
```

```
cd direct/DirectDNSServices/DirectDNSServer  
sudo ./bin/DirectDNSServer stop
```

```
##  
cd direct/james  
ps -eaf | grep james  
kill <pid>
```



5 Appendix I – DNS Configuration

For Direct, a DNS subdomain is recommended. For this example, rhex.us was chosen as the DNS domain, and direct.rhex.us is the sub-domain. Go-Daddy hosts the rhex.us domain name, which then has DNS records that point to the IP address of the Direct DNS responder machine, which is the Amazon EC2 instance described in the overview section. Each will be discussed in turn below.

Nameservers

Nameservers: (Last update 4/25/2012)
PDNS01.DOMAINCONTROL.COM
PDNS02.DOMAINCONTROL.COM

[Set Nameservers](#)
[Manage DS Records](#)

DNS Manager

DNS Manager: Available

A	@	68.178.232.100
A	ns1.direct	23.23.196.75
CNAME	calendar	login.secureserver.net
CNAME	email	email.secureserver.net
CNAME	fax	login.secureserver.net
MX	@	mailstore1.secureserver.net
MX	@	smtp.secureserver.net

[Launch](#)

TLD Specific

Nexus Agreements:
[Edit Agreement](#)

Two entries into the rhex.us zone file are needed. The first is a host entry to indicate where the direct.rhex.us subdomain nameserver is located, in this case ns1.direct can be found at the IP address 23.23.196.75. See the red circle below



Zone File Editor

RHEX.US

[change zone](#)



25% off your order of \$/5 or more*
Save BIG on websites, domain names a



Make life easier with zone file templates that lock in known working configurations. Go to the **Templates** tab to get started.

rhex.us (Last saved 4/25/2012 1:55:03 PM MST)

Saved

A (Host) ?

<input checked="" type="checkbox"/> Host	Points to	TTL
<input type="checkbox"/> @	68.178.232.100	1 Hour
<input type="checkbox"/> ns1.direct	23.23.196.75	1 Hour

Quick Add

CNAME (Alias) ?

Restore Defaults

<input checked="" type="checkbox"/> Host	Points to	TTL
<input type="checkbox"/> calendar	login.secureserver.net	1 Hour
<input type="checkbox"/> email	email.secureserver.net	1 Hour
<input type="checkbox"/> fax	login.secureserver.net	1 Hour
<input type="checkbox"/> files	login.secureserver.net	1 Hour
<input type="checkbox"/> ftp	@	1 Hour
<input type="checkbox"/> mail	pop.secureserver.net	1 Hour
<input type="checkbox"/> mobilemail	mobilemail-v01.prod.mesa1.secureserver.net	1 Hour
<input type="checkbox"/> pop	pop.secureserver.net	1 Hour
<input type="checkbox"/> smtp	smtp.secureserver.net	1 Hour
<input type="checkbox"/> www	@	1 Hour

Quick Add

MX (Mail Exchanger) ?

Restore Defaults

<input checked="" type="checkbox"/> Priority	Host	Points to	TTL
<input type="checkbox"/> 10	@	mailstore1.secureserver.net	1 Hour
<input type="checkbox"/> 0	@	smtp.secureserver.net	1 Hour

Quick Add

TXT (Text) ?

<input checked="" type="checkbox"/> Host	TXT Value	TTL
--	-----------	-----

Quick Add

Add SPF Record

The second entry is a name server (NS) which responds ties the host direct to ns1.direct.rhex.us. So, essentially, when a dns query is requested, godaddy will respond with the direct.rhex.us DNS machine. Together, these records are referred to as “DNS glue records”. Interestingly, some domain name companies, do not allow DNS glue records. For example, during setup, ZoneEdit was used and it did not allow configuration of DNS glue records.

17



[Quick Add](#) [Add SPF Record](#)

SRV (Service) [?](#)

✓ Service	Protocol	Name	Priority	Weight	Port	Target	TTL
-----------	----------	------	----------	--------	------	--------	-----

[Quick Add](#)

AAAA (IPv6 Host) [?](#)

✓ Host	Points to	TTL
--------	-----------	-----

[Quick Add](#)

NS (Nameserver) [?](#)

✓ Host	Points to	TTL
<input type="checkbox"/> @ (Informational)	pdns01.domaincontrol.com (Informational)	1 Hour (Informational)
<input type="checkbox"/> @ (Informational)	pdns02.domaincontrol.com (Informational)	1 Hour (Informational)
<input type="checkbox"/> direct	ns1.direct.rhex.us	1 Hour

[Quick Add](#)

SOA (Start of Authority) [?](#)

✓ Responsible Party Email	Refresh	Retry	Expire	Serial	TTL
<input type="checkbox"/> dns.jomax.net	28800	7200	604800	2012042501	1 Hour

Copyright © 1999-2012 GoDaddy.com, LLC. All rights reserved.

The Direct DNS responder for direct.rhex.us is next to configure which is found on the DNS Entries tab of the Direct configuration gateway (:8081/Config-ui). A total of 6 entries are necessary for proper configuration. The first are three host records, which resolve ns1, direct, and mail.direct.rhex.us respectively.



Manage Domains Agent Settings Certificates DNS Entries You are logged in. Log out

DNS Resolver Configuration

"A" Records

Name IP Address TTL

86400

Add Record

Host	Points To	TTL	Remove
ns1.direct.rhex.us.	/23.23.196.75	86400	<input type="checkbox"/>
direct.rhex.us.	/23.23.196.75	86400	<input type="checkbox"/>
mail.direct.rhex.us.	/23.23.196.75	86400	<input type="checkbox"/>

Remove Selected As

"AAAA" (IPv6) Records

Name IP Address TTL

86400

Add Record

"CNAME" (Alias) Records

Name Alias for TTL

86400

Add Record

The next is a MX record such that outside email servers can find and exchange emails with the Direct James email server.

"MX" (Mail Exchange) Records

Priority 0

Host

Points To

TTL 86400

Add Record

Priority	Host	Alias For	TTL	Remove
0	direct.rhex.us.	mail.direct.rhex.us.	86400	<input type="checkbox"/>

Remove Selected MXs

"SRV" (Service) Records

Service

Protocol

Domain

TTL 86400

Priority 0

Weight 0

Port 0

Target

Add Record



A start of authority (SOA) record is needed to establish that this DNS responder and name server is where the sub-domain direct.rhex.us origin is. This allows DNS clients to find the source of the direct.rhex.us sub-domain. The individual parameters are fairly generic and can be duplicated in other setups if desired.

Finally, an NS record is needed establish that direct.rhex.us has ns1.direct.rhex.us as its name server which happens to reside on the same machine.

"SOA" (Start of Authority) Records

Name

Host Master

Name Server

TTL

Expire

Minimum

Refresh

Serial

Retry

Name	Host Master	Name Server	TTL	Expire	Min.	Refresh	Serial	Retry	Remove
direct.rhex.us.	hostmaster.direct.rhex.us.	ns1.direct.rhex.us.	86400	12096	3600	17280	2012042502	900	<input type="checkbox"/>

"NS" Records

Name Target TTL

Name	Target	TTL	Remove
direct.rhex.us.	ns1.direct.rhex.us.	86400	<input type="checkbox"/>

Once all of the above entries are in place, SAVE the configuration!! Next start or restart the Direct DNS responder to ensure all values are in place. I found the following list of commands invaluable to ensure the DNS configuration is in place and working, because without it, the Direct gateway will not function properly.

The "dig" utility is an excellent tool to send DNS queries, coupled with the short response, one can quickly discern positive results.



The next commands test the GoDaddy configuration to ensure rhex.us is responding correctly. As you can see most answers redirect you to the Direct DNS responder on the 23.23 network which is the expected behavior.

```
dig -t A rhex.us +short
68.178.232.100
```

```
dig -t A direct.rhex.us +short
23.23.196.75
```

```
dig -t NS direct.rhex.us +short
ns1.direct.rhex.us
```

```
dig -t A ns1.direct.rhex.us
23.23.196.75
```

Utilizing the @<name server> dig options, we are able to directly query the direct.rhex.us DNS responder

```
dig -t A direct.rhex.us @ns1.direct.rhex.us +short
23.23.196.75
```

```
dig -t NS direct.rhex.us @ns1.direct.rhex.us +short
ns1.direct.rhex.us
```

```
dig -t SOA direct.rhex.us @ns1.direct.rhex.us +short
ns1.direct.rhex.us tjones@direct.rhex.us 2012042502 ...
```

Once all of the above tests work properly, your setup is now ready for querying for a Direct email user cert. If the test commands above do not work, one issue might be that most enterprises block DNS UDP traffic through their firewalls, so it is important to ensure that your test machine has a clear and direct network path to the internet.

The Direct DNS cert lookup test is the final and most important test to try. Utilizing the “cert” DNS type, the dig command sends a request to the direct.rhex.us DNS responder for the tjones.direct.rhex.us X.509 cert. The result below should be the correct response. If this does not work, it is likely an issue with the certificate that was loaded using the Direct configuration UI as described above. It is recommended to go back to this step and revisit.

```
dig -t cert tjones.direct.rhex.us @ns1.direct.rhex.us
```

```
;; ANSWER SECTION:
```



tjones.direct.rhex.us. 86307 IN CERT PKIX 34517 RSASHA1
MIIDJzCCApCgAwIBAgIIY/IhiEopa34wDQYJKoZIhvcNAQEFBQAwwVTEX
MBUGA1UEAwOZGlyZWNOLnJoZXgudXMxCzAJBgNVBAYTA1VMTQswCQYD
VQQIDAjNQTETQMA4GA1UEBwwHQmVkJm9yZDEOMAwGA1UECgwFTU1UUkUw
HhcNMTIwNDI2MTkyNTQwWhcNMTMwNDEwMTkyNTQwWjB6MSUwIwYJKoZI
hvcNAQkBFhZnZ2FubGV5QGRpcmVjdC5yaGV4LnVzMdUwEwYDVQDDAxH
cmVnZyBHYW5sZXkxCzAJBgNVBAYTA1VMTQswCQYDVQIDAjNQTETQMA4G
A1UEBwwHQmVkJm9yZDEOMAwGA1UECgwFTU1UUkUwgZ8wDQYJKoZIhvcN
AQEBBQADgY0AMIGJAoGBAIBsCSgXUfypdG9RSOkb6rIvBTKj6BNmsl+F
i4YlNtTFFVjsUbQjYhDuJLoq+wsZLzcFhot7PJEg8zROvo8y6e9Q3yeh
TiXkTXfntIIzz5JzdBaovJ+K9uXL6K1Byp725VON24dVVFAPGUseUCT0
mM1qLW84zyeAgxachSOkcYbVAgMBAAGjgdowgdcwgYQGA1UdIwR9MHuA
FOrOKlhsnfab0WjPPIoC8jfMde/yoVmkVzBVMRcwFQYDVQDDA5kaXJl
Y3QucmhleC51czELMAkGA1UEBhMCVVMxCzAJBgNVBAGMAk1BMRAwDgYD
VQQHDAZCZWrmB3JkMQ4wDAYDVQQKDAVNSVRSRYIIR3jRKsVeMGQwHQYD
VR0OBBYEFJDPMp7Qjd9FoQSH091ui2phV5nYMAwGA1UdEwEB/wQCMAAw
IQYDVR0RBBowGIEWZ2dhbmXleUBkaXJlY3QucmhleC51czANBgkqhkiG
9w0BAQUFAAOBgQDKtOgo2t5b1t3MvjW2e9KEJpp1/Ls04429BFg969a1
lm+U9501DahvN1x5Nnx1opoYknXFQxo5mxnG8LMN4YKLy08oVr0JLVDA
39z5o/sHfaAgFxOBZDj3+1s/3yQ1f7ChdMxckTs4HWk+fkoghq4skgUJ
I0OJM5DbHK4VcuZ76w==



6 Appendix II – Direct Configuration and James database MySQL conversion

Both components make use of the Derby in memory database, which is unable to retain its data across process restarts, unless it is configured to do so. As a matter of preference, we chose to move both databases to MySQL. The following steps and issues were followed:

MySQL install on server. Below is the list of Ubuntu Linux MySQL libraries I installed.

```
## prep work for DB transition to Mysql
sudo apt-get install mysql-client-5.1
518 sudo apt-get install libmysql++-dev
519 sudo apt-get install libmysql++3
520 sudo apt-get install mysql-server mysql-common
521 sudo apt-get install libmysql-java
522 sudo apt-get install mysql-server-5.1
523 sudo apt-get install mysql-client-core-5.1
524 sudo apt-get install libdbd-mysql-ruby1.9.1
525 sudo apt-get install libmysqlclient16
526 sudo apt-get install libmysql-ruby1.9.1
527 sudo apt-get install sqlline
```

Next, the MySQL.jar library file was installed in several locations:

```
./direct/james-2.3.2/lib/mysql-connector-java-5.1.20-bin.jar
./direct/apache-tomcat-7.0.23/lib/mysql-connector-java-5.1.20-bin.jar
```

Next, the configuration bean configurations for Direct and James were configured

```
direct/apache-tomcat-7.0.23/webapps/config-service/WEB-INF/beans.xml
```



```
<!-- MySQL JPA CONFIGURATION TEMPLATE -->
<bean id="entityManagerFactory"
      class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
  <property name="jpaVendorAdapter">
    <bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
      <property name="showSql" value="false" />
      <property name="generateDdl" value="true" />
      <property name="databasePlatform" value="org.hibernate.dialect.MySQL5Dialect" />
    </bean>
  </property>
  <property name="dataSource" ref="dataSource"/>
  <property name="persistenceUnitName" value="config-store" />
</bean>
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
  <property name="url" value="jdbc:mysql://localhost:3306/nhind"/>
  <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
  <property name="username" value="nhind" />
  <property name="password" value="<password>" />
</bean>
<!-- END MySQL JPA CONFIGURATION -->
```

Then the James Config file:

direct/james-2.3.3/apps/james/SAR-INF/config.xml

```
<data-source name="maildb" class="org.apache.james.util.dbcp.JdbcDataSource">
  <driver>com.mysql.jdbc.Driver</driver>
  <dburl>jdbc:mysql://localhost:3306/james_mail_userdb?autoReconnect=true</dburl>
  <user>james</user>
  <password><password></password>
  <max>20</max>
</data-source>
```

Next, mysql users were created:

```
create database nhind;
CREATE USER 'nhind'@'localhost' IDENTIFIED BY < password >;
GRANT ALL PRIVILEGES ON nhind.* TO 'nhind'@'localhost' WITH GRANT
OPTION;
```

Then found the config-service schema file to load the schema

```
mysql -unhind -p<password> < createMySQL.sql
```




```
~/projs/rhex/direct-src/java/config/config-store/src/main/resources ] $ ll
total 16
drwxr-xr-x  3      staff   102 Jun 21 10:26 META-INF
-rw-r--r--  1      staff  2006 Jun 21 10:26 createMySQL.sql
-rw-r--r--  1      staff  1980 Jun 21 10:26 createPostgres.sql
```

Doing a quick MySQL login and check showed that all of the Direct DB tables were loaded.

```
mysql> show tables;
+-----+
| Tables_in_nhind |
+-----+
| address          |
| anchor           |
| certificate       |
| dnsrecord        |
| domain           |
| setting          |
+-----+
6 rows in set (0.00 sec)
```

Finally, to test, I restarted the Tomcat server. Unfortunately, the restart did not properly start up the application. A database error was thrown indicating the date and time was incorrect for certain fields. Digging deeper, the `TIMESTAMP` data type in MySQL has a limited range, essentially from 1970 to present day. So, if a record had a 0 timestamp, the database would throw an exception. The following web site:

<http://stackoverflow.com/questions/409286/datetime-vs-timestamp> had an excellent explanation.

The `DATETIME` type is used when you need values that contain both date and time information. MySQL retrieves and displays `DATETIME` values in 'YYYY-MM-DD HH:MM:SS' format. The supported range is '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.

...

The `TIMESTAMP` data type has a range of '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC. It has varying properties, depending on the MySQL version and the SQL mode the server is running in.

So, basically, all `TIMESTAMP` data types were replaced with `DATETIME` and the Direct Config service was able to function properly storing its data in MySQL.