

# Docker Package Build Testing - Linux

## Overview

Currently, package build testing on Linux environments is kind of a hassle. You have to find a server or VM that has all of the prerequisite software installed, and go and do struggle through the many nuances of our installation process. Fortunately, Docker provides an easy and simple way to launch Linux Distributions quickly and effectively. By using docker, we can run package build installs for our supported Linux platforms on our local environments. The aim of this document is to show how Docker can be used to do package build testing, and provide an overview of the installation process.

This document assumes you have docker configured on your local instance. Since this configuration can vary greatly depending on the distro you are using, please see the docker website for instructions. **Windows users should see the [Docker Package Build Testing - Windows guide](#).**

## Step 1. Create the Base Image

The first step is to create a docker image. I created mine off of the official *centos:centos7.1.1503* image. The docker file bellow will create a new image with all of the required JDA software, as well as a few things needed for testing:

```
FROM centos:centos7.1.1503

RUN yum -y remove iputils

RUN yum -y update

RUN yum -y install sudo nano firefox ksh ant make gcc gcc-c++ glibc glibc.i386
glibc-common glibc-common.i386 glibc-devel glibc-devel.i386 glibc-headers libgcc
libgcc.i386 libstdc++.i386 libstdc++-devel libstdc++-devel.i386 compat-libstdc++-33
compat-libstdc++-33.i386 tar

RUN dbus-uuidgen > /etc/machine-id

# Replace 1000 with your user / group id
RUN export uid=1000 gid=1000 && \
    mkdir -p /home/developer && \
    echo "developer:x:${uid}:${gid}:Developer,,,:/home/developer:/bin/bash" >>
/etc/passwd && \
    echo "developer:x:${uid}:" >> /etc/group && \
    echo "developer ALL=(ALL) NOPASSWD: ALL" > /etc/sudoers.d/developer && \
    chmod 0440 /etc/sudoers.d/developer && \
    chown ${uid}:${gid} -R /home/developer

USER developer
ENV HOME /home/developer
```

After that run the following command to build a new image called pkg-no-java off of the Dockerfile

```
cd path/to/Dockerfile
docker build -t pkg-no-java .
```

We now have a base image with all of our required packages... Except for java. There currently is not a good way to install the official oracle JDK on a docker image without doing it manually.

Go to the Oracle website and download the latest Java JDK rpm. Then create the following directories somewhere in your workspace:

```
pkgShare
pkgShare/install
pkgShare/setup
```

Move the java rpm to the pkgShare/setup directory listed above. Next, let's start a container:

```
docker run -idt -v pkgShared:/shared --net=host --name pkgbld pkg-no-java
```

Make sure to include the entire path to your pkgShared is the complete. Now let's enter the container, and install java:

```
docker exec -it pkgbld bash
#You should now be in the container
cd /shared/setup

# Replace this with java_jdk.rpm with actual rpm
sudo rpm -ivh java_jdk.rpm

# Run these and select the non openJDK option for each
sudo /usr/sbin/alternatives --config java
sudo /usr/sbin/alternatives --config javac
```

We now have java installed. One final step, we will need the installer to think this is a redhat release, instead of a centos installation. In the future, I may base this off of a redhat image, making this step unnecessary.

```
sudo nano /etc/redhat-release
```

And replace the existing contents with the following:

```
Red Hat Enterprise Linux Server release 7.1 (Maipo)
```

This container should now be ready to perform as our base image, which can now be used to do package build testing. Let's exit the container and commit it by create a new image:

```
docker commit pkgbld pkgbld-base
```

Finally, let's clean up and remove the old container:

```
docker stop pkgbld && docker rm pkgbld
```

## Step 2. Download and Extract the Installation Media

The installer can be found on the beluga server. In general, once build the instalation media is placed in a versioned directory in the following location:

```
//beluga/d1/packager/stage
```

You will want to mount this directory somewhere on your local instance. From there, navigate to the desired major version, and build, and copy the file with roughly the following name:

```
sce_linux3.1-x86_64-rhel-7_VERSION.tar
```

The next step is to extract the tar file to the 'pkgShared/install' directory, and move on to the installation steps.

### Step 3. Install Reporting/WMS/REFS

We now need to go ahead and install of the the products the customers will be using. In our case, we will be installing reporting, WMS, and REFS. In order to do the install, let's startup a container off of our base image:

```
docker run -idt -v pkgShared:/shared -p 4922:4922 --net=host --name pkgbld pkgbld-base
```

### Install Reporting

Let's start a bash session in the container:

```
docker exec -it pkgbld bash
```

### Root Install

We now need to run the Root Install. This can be done by executing the following

```
cd /shared/install/unixserver/install  
sudo sh ./RootInstall
```

The installer will ask you to accept the License Agreement, Go ahead and do it.

The installer will verify the source installation directory. Generally speaking the default value is correct, but it should match bellow. If so click enter to use the defaults:

```
* The distribution source is the directory that the distribution  
  resides in. This could be a pathname to the mount point of the  
  CD-ROM or a pathname to the directory that the distribution was  
  copied to.  
  Distribution source [/shared/install/unixserver]:
```

Next we will configure the Environment/User Name/Group/Auto Startup/MOCA and RMI Port/Installation Location. Use the settings Bellow:

```
Environment name: rpt91
    UNIX login: rpt91
    group: rpt91
    Auto startup: N
Port number: 4722
RMI Port: 4723
    Directory: /opt/redprairie/rpt91
```

Next it will provide options for the start kill scripts. Press Enter through this to use the default values:

```
* The start script name is the name that will be used to create the
  symbolic link in the /sbin/rc2.d directory.
* The kill script name is the name that will be used to create the
  symbolic link in the /sbin/rc1.d directory.
* Both the start and kill script names must be chosen so that they are
  executed in the correct sequence.
Start script name [S95redprairie]:
Kill script name [K05redprairie]:
```

Next you will be prompted for the location of your perl install. The default is fine. Press Enter to acknowledge.

```
Directory: /usr/bin
```

Next you will be prompted for the JDK Directory. Use /usr/java/latest for the JDK, and the JRE directory will then default to the correct location.

```
* The Java 8 64-bit JDK directory contains the Java Development Kit.
* The Java 8 64-bit JRE directory contains the Java run time.
* The Java 8 64-bit JRE directory value is REQUIRED for most products.
Java 8 64-bit JDK directory: /usr/java/latest
Java 8 64-bit JRE directory [/usr/java/latest/jre]:
```

The next prompt is asking if this is a web server, default answer of No is correct:

```
By selecting Yes below, the install will configure Apache
for use by this install as a Web Server.
    Will this install be a Web Server? (Y/N) [N]:
```

The installer will now do a pre install check. All should be well and your screen should look as followed:

```
* If any errors are reported, check the preinstall logfiles in the
/tmp/redprairie directory.
* Any errors reported must be fixed before continuing with the install.
* Any warnings reported can be fixed, but don't have to be. For
example, you may want to use a login that already exists as the login
for a product.
Validating configuration for administration files 9.1.0.0... ok
Validating configuration for Environment 9.1.0.0... ok
Press [ENTER] to continue.
```

The root install will then continue, and install the users and les directory to the desired location. This should go okay:

```
* The install process can take some time.
* You can check the progress by looking at the install logfiles in the
/tmp/redprairie directory.
* If any errors are reported, check the install logfiles in the
/tmp/redprairie directory.
Installing administration files 9.1.0.0... ok
Installing Environment 9.1.0.0... ok
The install completed successfully.
```

This concludes the root install process for reporting.

## User Install

We need to do the next part as the user we just created. Switching the user also executes the profile settings:

```
sudo su - rpt91

# If done sucessfully, your terminal prompt should now appear as followed:

Setting up the LES rpt91 environment...
rpt91 :>
```

Navigate back to the install directory, and run the user install

```
cd /shared/install/unixserver/install
java -jar userinstall.jar
```

Again, accept the License Agreement. Ensure the Distribution source is correct:

```
* The distribution source is the directory that the distribution
resides in. This could be a pathname to the mount point of the
CD-ROM or a pathname to the directory that the distribution was
copied to.
Distribution source [/shared/install/unixserver]:

Are all of these values ok? (Y/N) Y
```

Next it will prompt for the packages you would like to install. Select the following

```
Please select one or more choices that apply.
Do you want to install Warehouse Management? (Y/N) [N]
Do you want to install Workforce Labor Management? (Y/N) [N]
Do you want to install Track & Trace? (Y/N) [N]
Do you want to install Inventory Visibility? (Y/N) [N]
Do you want to install Other Supporting Products? (Y/N) [N]Y
Are all of these values ok? (Y/N)
```

Next, you will be prompted to install the Application Server and the Web Server. Only install the Application Server

```
Please select one or more choices that apply.
Do you want to install Application Server? (Y/N) [N]Y
Do you want to install Web Server? (Y/N) [N]
Are all of these values ok? (Y/N)
```

Now it will go through the list of extra products to install. We need to install MOCA and Reporting

```
* At least one feature must be chosen for installation.
* If no products features are chosen, the installer will exit.
Do you want to install Data Extractor? (Y/N) [N]
Do you want to install ERP Interface? (Y/N) [N]
Do you want to install Event Management? (Y/N) [N]
Do you want to install Integrator? (Y/N) [N]
Do you want to install MOCA? (Y/N) [N]Y
Do you want to install Reporting? (Y/N) [N]Y
```

The user installer will once again validate the JDK and JRE to be used. It should default to whatever was provided during the root install step. This is correct.

```
* The Java 1.8 64-bit bit JDK directory contains the Java Development Kit.
* The Java 1.8 64-bit bit JRE directory contains the Java run time.
Java 1.8 64-bit bit JDK directory: /usr/java/latest
Java 1.8 64-bit bit JRE directory: /usr/java/latest/jre
Are all of these values ok? (Y/N)
```

The configuration is complete. The installer will now go through it's preinstall check. Validate that everything is okay:

```
* If any errors are reported, check the preinstall logfiles in the
/opt/redprairie/rpt91/les/log directory.
* Any errors reported must be fixed before continuing with the install.
* Any warnings reported can be fixed, but don't have to be. For
example, you may want to use a login that already exists as the login
for a product.
Validating configuration for MCS... ok
Validating configuration for MOCA... ok
Validating configuration for Reporting... ok
Validating configuration for Environment... ok
```

Then the installation process will continue. Ensure that there are no errors:

```
* The install process can take some time.
* You can check the progress by looking at the install logfiles in the
/opt/redprairie/rpt91/les/log directory.
* If any errors are reported, check the install logfiles in the
/opt/redprairie/rpt91/les/log directory.
Installing MOCA... ok
Installing MCS... ok
Installing RPUX... ok
Installing Reporting... ok
Configuring Environment... ok
```

The installer will attempt to install database utils. This does nothing, which is fine. The preinstall config will also run and tell you to press enter. This is cool too.

```
* Installing Database Utils...
Press [ENTER] to continue.
```

After the user install should complete

```
Installation complete.
Press [ENTER] to Exit.
```

We now need to exit to get back to the super user. Type exit:

*Note: It's important to exit out of the reporting users session so in the next step we can set the product configurations by re-logging in as the reporting user*

```
exit
```

## Registry Install

The final installation is the registry installation. We should now be the root user again. Let's switch back into reporting user:

```
sudo su - rpt91

# If done successfully, your terminal prompt should now appear as followed:

Setting up the LES rpt91 environment...
rpt91 :>
```

Now let's run the registry install

```
cd $LESDIR/dtbs/database
java -jar mocareg.jar
```

The installer will start. The first screen should be configured correctly with the ports we set earlier. Make sure that the host matches the hostname of your machine:

```
* General service configuration.

Service port: 4722

Service RMI Port: 4723

Service url: http://j1014822NIXLT:4722/service
```

Java configuration setting should be correct:

```
The Sun JDK is used for the two Linux platforms that we support - RedHat Enterprise Linux and SUSE Linux Enterprise Server. SUN has a separate 32-bit and 64-bit distribution for each, which install into two separate locations. The java.vm registry key should point to the 64-bit Java executable. The java.native-vmargs registry key is set to define a maximum heap size of 128 M and should be considered only a starting point. The JAVA_HOME environment variable in the .profile file should point to the 64-bit Java installation.

Java 1.8 (64-bit) path (vm): /usr/java/jdk1.8.0_73/jre/bin/java
Java 1.8 (64-bit) arguments (vmargs): -server -Xmx4096m
Java 1.8 (64-bit) native arguments (native-vmargs): -server -Xmx128m
```

The next page is for the reporting database configuration. In our case, we are not going to hook this into an actual database. Select 3 for none:

```
* The database connection information is required for most products
  * Supported database types:
  * 1. Oracle 12
  * 2. SQL Server 2014
  * 3. NONE
  Choose database to use (1-3) [3]: 3
```

We are prompted for database settings. Let's use the standard settings. After this screen you will be asked to confirm the settings. :



```
The user's tablespace definition file can be used to override the default tablespaces
in which
```

```
    schema will be created.
```

```
    Choose whether you want to customize this file.
```

1. Use Standard settings
2. Manually customize usrtbldef.h
3. Use dbuser's default tablespace for all tables

```
Enter option between [1-3] [3]: 1
```

Now we are asked to set the reporting URL. Since this is the reporting server, we are not using an external host. Put Nothing

```
Reporting url:
```

```
Are all of these values ok? (Y/N)
```

We do not have an event Management URL. Again put none.

```
Event Management url:
```

```
Are all of these values ok? (Y/N)
```

Next you will be asked for an RPWEB URL. We will not be hooking reporting up to RPWEB. Put none. Do the same for refs-dir:

```
Required if REFS server will be connecting to this instance
```

```
RPWEB url (if any):
```

```
refs-dir (if any):
```

The installer will now look for custom components, mbuild, and do a wsdeploy. Click enter through all of those screens. There may be warnings about components being missing during mbuild, this is okay. Validate the install by tunneling the mocaserver. If the server listens for connections, the installation was successful, and reporting is now installed.

Let's exit back to root:

```
exit
```

## Install WMS

### Root Install

We now need to run the Root Install. This can be done by executing the following. Make sure that we are against the sudo develop user.

```
cd /shared/install/unixserver/install
sudo sh ./RootInstall
```

The installer will ask you to accept the License Agreement, Go ahead and do it.

The installer will verify the source installation directory. Generally speaking the default value is correct, but it should match below. If so click enter to use the defaults:

```
* The distribution source is the directory that the distribution
resides in. This could be a pathname to the mount point of the
CD-ROM or a pathname to the directory that the distribution was
copied to.
Distribution source [/shared/install/unixserver]:
```

Next we will configure the Environment/User Name/Group/Auto Startup/MOCA and RMI Port/Installation Location. Use the settings Below:

```
Environment name: wms91
      UNIX login: wms91
      group: wms91
      Auto startup: N
Port number: 4422
RMI Port: 4423
      Directory: /opt/redprairie/wms91
```

Next it will provide options for the start kill scripts. Press Enter through this to use the default values:

```
* The start script name is the name that will be used to create the
symbolic link in the /sbin/rc2.d directory.
* The kill script name is the name that will be used to create the
symbolic link in the /sbin/rc1.d directory.
* Both the start and kill script names must be chosen so that they are
executed in the correct sequence.
Start script name [S95redprairie]:
Kill script name [K05redprairie]:
```

Next you will be prompted for the location of your perl install. The default is fine. Press Enter to acknowledge.

```
Directory: /usr/bin
```

Next you will be prompted for the JDK Directory. Use /usr/java/latest for the JDK, and the JRE directory will then default to the correct location.

```
* The Java 8 64-bit JDK directory contains the Java Development Kit.
* The Java 8 64-bit JRE directory contains the Java run time.
* The Java 8 64-bit JRE directory value is REQUIRED for most products.
Java 8 64-bit JDK directory: /usr/java/latest
Java 8 64-bit JRE directory [/usr/java/latest/jre]:
```

The next prompt is asking if this is a web server, default answer of No is correct:

```
By selecting Yes below, the install will configure Apache
for use by this install as a Web Server.
```

```
Will this install be a Web Server? (Y/N) [N]:
```

The installer will now do a pre install check. All should be well and your screen should look as followed:

```
* If any errors are reported, check the preinstall logfiles in the
  /tmp/redprairie directory.
* Any errors reported must be fixed before continuing with the install.
* Any warnings reported can be fixed, but don't have to be. For
  example, you may want to use a login that already exists as the login
  for a product.
Validating configuration for administration files 9.1.0.0... ok
Validating configuration for Environment 9.1.0.0... ok
Press [ENTER] to continue.
```

The root install will then continue, and install the users and les directory to the desired location. This should go okay:

```
* The install process can take some time.
* You can check the progress by looking at the install logfiles in the
  /tmp/redprairie directory.
* If any errors are reported, check the install logfiles in the
  /tmp/redprairie directory.
Installing administration files 9.1.0.0... ok
Installing Environment 9.1.0.0... ok
The install completed successfully.
```

This concludes the root install process for WMS.

## User Install

We need to do the next part as the user we just created. Switching the user also executes the profile settings:

```
sudo su - wms91

# If done sucessfully, your terminal prompt should now appear as followed:

Setting up the LES wms91 environment...
wms91 :>
```

Navigate back to the install directory, and run the user install

```
cd /shared/install/unixserver/install
java -jar userinstall.jar
```

Again, accept the License Agreement. Ensure the Distribution source is correct:

```
* The distribution source is the directory that the distribution
resides in. This could be a pathname to the mount point of the
CD-ROM or a pathname to the directory that the distribution was
copied to.
Distribution source [/shared/install/unixserver]:

Are all of these values ok? (Y/N) Y
```

Next it will prompt for the packages you would like to install. Select just WMD:

```
Do you want to install Warehouse Management? (Y/N) [N]Y
Do you want to install Workforce Labor Management? (Y/N) [N]
Do you want to install Track & Trace? (Y/N) [N]
Do you want to install Inventory Visibility? (Y/N) [N]
Do you want to install Other Supporting Products? (Y/N) [N]
Are all of these values ok? (Y/N)
```

Next, you will be prompted to install the Application Server and the Web Server. Only install the Application Server

```
Please select one or more choices that apply.
Do you want to install Application Server? (Y/N) [N]Y
Do you want to install Web Server? (Y/N) [N]
Are all of these values ok? (Y/N)
```

Now it will go through the list of extra products to install. We need to install just the WMS.

```
* At least one feature must be chosen for installation.
* If no products features are chosen, the installer will exit.
Do you want to install ERP Interface? (Y/N) [N]
Do you want to install Event Management? (Y/N) [N]
Do you want to install Reporting? (Y/N) [N]
Do you want to install Slotting? (Y/N) [N]
Do you want to install Warehouse Management? (Y/N) [N]Y
```

The user installer will once again validate the JDK and JRE to be used. It should default to whatever was provided during the root install step. This is correct.

```
* The Java 1.8 64-bit bit JDK directory contains the Java Development Kit.
* The Java 1.8 64-bit bit JRE directory contains the Java run time.
Java 1.8 64-bit bit JDK directory: /usr/java/latest
Java 1.8 64-bit bit JRE directory: /usr/java/latest/jre
Are all of these values ok? (Y/N)
```

The configuration is complete. The installer will now go through it's preinstall check. Validate that everything is okay:

```
* If any errors are reported, check the preinstall logfiles in the
/opt/redprairie/wms91/les/log directory.
* Any errors reported must be fixed before continuing with the install.
* Any warnings reported can be fixed, but don't have to be. For
example, you may want to use a login that already exists as the login
for a product.
Validating configuration for Warehouse Management... ok
Validating configuration for Integrator... ok
Validating configuration for MTF... ok
Validating configuration for SAL... ok
Validating configuration for MCS... ok
Validating configuration for MOCA... ok
Validating configuration for Reporting... ok
Validating configuration for Environment... ok
```

Then the installation process will continue. Ensure that there are no errors:

```
* The install process can take some time.
* You can check the progress by looking at the install logfiles in the
/opt/redprairie/wms91/les/log directory.
* If any errors are reported, check the install logfiles in the
/opt/redprairie/wms91/les/log directory.
Installing MOCA... ok
Installing MCS... ok
Installing RPUX... ok
Installing SAL... ok
Installing MTF... ok
Installing Integrator... ok
Installing Warehouse Management... ok
Installing Reporting... ok
Configuring Environment... ok
```

The installer will attempt to install database utils. This does nothing, which is fine. The preinstall config will also run and tell you to press enter. This is cool too.

```
* Installing Database Utils...
                                Press [ENTER] to continue.
Persisting Install.cfg...
                                Press [ENTER] to continue.
```

After the user install should complete

```
Installation complete.
                                Press [ENTER] to Exit.
```

We now need to exit to get back to the super user. Type exit:

*Note: It's important to exit out of the wms users session so in the next step we can set the product configurations by re-logging in as the wms user*

```
exit
```

## Registry Install

The final installation is the registry installation. We should now be the root user again. Let's switch back into the wms user:

```
sudo su - wms91

# If done successfully, your terminal prompt should now appear as followed:

Setting up the LES wms91 environment...
wms91 :>
```

Now let's run the registry install

```
cd $LESDIR/dtbs/database
java -jar mocareg.jar
```

The installer will start. The first screen should be configured correctly with the ports we set earlier. Make sure that the host matches the hostname of your machine:

```
* General service configuration.

Service port: 4422

Service RMI Port: 4423

Service url: http://j1014822NIXLT:4422/service
```

Java configuration setting should be correct:

```
The Sun JDK is used for the two Linux platforms that we support - RedHat Enterprise Linux and SUSE Linux Enterprise Server. SUN has a separate 32-bit and 64-bit distribution for each, which install into two separate locations. The java.vm registry key should point to the 64-bit Java executable. The java.native-vmargs registry key is set to define a maximum heap size of 128 M and should be considered only a starting point. The JAVA_HOME environment variable in the .profile file should point to the 64-bit Java installation.

Java 1.8 (64-bit) path (vm): /usr/java/jdk1.8.0_73/jre/bin/java
Java 1.8 (64-bit) arguments (vmargs): -server -Xmx4096m
Java 1.8 (64-bit) native arguments (native-vmargs): -server -Xmx128m
```

The next page is for the wms database configuration. In my case, I have an oracle database install. This is recommended as it is uncommon for a Unix server installation to have a SQL database tied in. Provide the standard JDBC connection string, and the database user.

```
* The database connection information is required for most products
* Supported database types:
* 1. Oracle 12
* 2. SQL Server 2014
* 3. NONE
Choose database to use (1-3) [3]: 1
```

We are prompted for database settings. Let's use the standard settings. After this screen you will be asked to confirm the settings. :

```
The user's tablespace definition file can be used to override the default tablespaces
in which
    schema will be created.
Choose whether you want to customize this file.
1. Use Standard settings
2. Manually customize usrtbldef.h
3. Use dbuser's default tablespace for all tables

Enter option between [1-3] [3]: 1
```

Now we are asked to set the reporting URL. We will use the Reporting URL from the previous section:

```
Reporting url: http://j1014822NIXLT:4722/service
Are all of these values ok? (Y/N)
```

We do not have an event Management URL. Again put none.

```
Event Management url:
Are all of these values ok? (Y/N)
```

Next you will be asked for an RPWEB URL. We will configure this in the next section, but we do know the URL and ports we are going to use. Replace my hostname with the host of your local instance.

```
Required if REFS server will be connecting to this instance
RPWEB url: http://j1014822NIXLT:4922
```

The installer will now look for custom components, mbuild, and do a wsdeploy. Click enter through all of those screens. There may be warnings about components being missing during mbuild, this is okay. Validate the install by running the mocaserver. If the server listens for connections, the installation was successful, and WMS is now installed.

Let's exit back to root:

```
exit
```

## RPWeb Install

## Root Install

Once again, we should be the sudo developer user. Let's run the rpweb root install:

```
cd /shared/install/rpweb
sudo sh ./RootInstall
```

Accept the license agreement, and confirm the distribution source directory is correct:

```
* The distribution source is the directory that the distribution
  resides in. This could be a pathname to the mount point of the
  CD-ROM or a pathname to the directory that the distribution was
  copied to.
  Distribution source [/shared/install/rpweb]:
```

Next we will configure the environment/user/group/auto start/port/working directory. Use the values below:

```
* The Environment is what ties together your Application Server
  instance and will contain customizations created just for your
  installation.
  Environment name: refs91
    UNIX login: refs91
      group: refs91
    Auto startup: N
  Port number: 4922
    Directory: /opt/redprairie/refs91
```

The start and kill scripts are next up. The defaults here are fine:

```
* Administration files are used to automate the startup and shutdown
  of each Application Server instance on the system.
  Start script name: S95rpweb
  Kill script name: K05rpweb
```

Next configure the java settings. They should be defined as they are below:

```
* The Java 8 64-bit JDK directory contains the Java Development Kit.
  * The Java 8 64-bit JRE directory contains the Java run time.
    Java 8 64-bit JDK directory: /usr/java/latest
    Java 8 64-bit JRE directory: /usr/java/latest/jre
```

It should now start the installation. Confirm all of the products were successfully installed:



```
* The install process can take some time.
* You can check the progress by looking at the install logfiles in the
  /tmp/rpwebinstall directory.
* If any errors are reported, check the install logfiles in the
  /tmp/rpwebinstall directory.
Installing admin...  ok
Installing env...   ok
Installing profile... ok
                        The install completed successfully.
```

## RPWeb Installer

Let's start the user install by logging into the refs user.

```
sudo su - refs91

# If successful you should see this

Setting up the RPWEB refs91 environment...

refs91 :>
```

Now lets run the rpweb installer:

```
cd /shared/install/rpweb
java -jar install_rpweb.jar
```

Confirm the source installation location:

```
Enter REFS installation directory [/opt/redprairie/refs91/rpweb]:
```

Next we will be asked which products we need to install. We need to select REFS, WM, and Reporting:

```
Do you want to install REFS(Y/N)? [N]: Y
Do you want to install Warehouse Management(Y/N)? [N]: Y
Do you want to install Event Management(Y/N)? [N]:
Do you want to install Reporting(Y/N)? [N]: Y
Are all of these values ok? (Y/N)
```

Now we will set all of the RPWeb URLs. The default values here should be correct:

```
Server Port [4922]:  
  Stash URL [http://j1014822NIXLT:4922/stash]:  
  Static URL [http://j1014822NIXLT:4922/web]:  
  Base Locale [en-us] [en-us]:
```

```
[E]dit    [B]ack    Next [ENTER]
```

Next we will configure the WMS settings for refs. The Default Connection name type and application are all correct. Set the address to the MOCA URL of the WMS instance.

```
Connection Name [WM]:  
  Connection Type [moca/rps/sem/mock/cognos] [moca]:  
  Application Name [ems/mcs/tm/wm/trc/parcel/cognos/reporting] [wm]:  
  Address [http://localhost:9300/service]: http://j1014822NIXLT:4422/service
```

```
[E]dit    [U]ndo    [A]dd    [B]ack    Next [ENTER]
```

Next we will configure the Reporting settings for REFS. The Default Connection name type and application are all correct. Somewhat surprisingly, we will use the URL for the WMS for the address of the reporting server.

```
Connection Name [REPORTING]:  
  Connection Type [moca/rps/sem/mock/cognos] [moca]:  
  Application Name [ems/mcs/tm/wm/trc/parcel/cognos/reporting] [reporting]:  
  Address [http://localhost:5500/service]: http://j1014822NIXLT:4422/service
```

```
[E]dit    [U]ndo    [A]dd    [B]ack    Next [ENTER]
```

Next we will configure the WMS settings for refs. The Default Connection name type and application are all correct. Once again, set the address to the MOCA URL of the WMS instance.

```
Connection Name [MCS]:  
  Connection Type [moca/rps/sem/mock/cognos] [moca]:  
  Application Name [ems/mcs/tm/wm/trc/parcel/cognos/reporting] [mcs]:  
  Address [http://localhost:5500/service]: http://j1014822NIXLT:4422/service
```

```
[E]dit    [U]ndo    [A]dd    [B]ack    Next [ENTER]
```

You should now see the authentication information screen. We will use Native authentication, and the site name will be WM:

```
Authentication Type [Native/ReverseProxy/FederatedSSO/LDAP] [Native]:  
Authentication Site Name (MCS/REPORTING/WM) [MCS]: WM
```

The database configuration is next. Again, this should probably be an oracle database as our customers will likely use this for their configuration. Change the JDBC URL and user to match your rpweb database configuration:

```
Database Type [oracle/sqlserver]: oracle  
Example jdbc connection url: jdbc:oracle:thin:@{hostname}:1521:orcl  
JDBC Url: jdbc:oracle:thin:@10.47.4.205:15221:orcl  
User: RPWEB  
Password: RPWEB  
DBO User: RPWEB  
DBO Password: RPWEB  
Running dbcheck:  
Turning Off AutoCommit...  
OK  
[E]dit      [B]ack      Next [ENTER]
```

This should complete the rpweb installation. You should be able to run the refs server successfully.

### Step 3 Alternate. Install Via Silent Installer

On versions post 2012.2, there is a silent installer available for the UNIX platforms. Configuring the silent install scripts can speed up the install process a lot. The downside is from time to time screens will be added/removed the installation process, causing the scripts to become out of date. Attached to this document is a set of [response files](#) which can be modified and used to automatically run the package install. Extract the directory to the pkgShared directory, so it can be shared with your container. Make sure the root level directory is as followed:

```
pkgShared/auto_rsp
```

Next, the following files will need to be modified with your personal Host and database URL and credentials:

```
pkgShared/auto_rsp/reporting/mocareg  
pkgShared/auto_rsp/wms/mocareg  
pkgShared/auto_rsp/refs/install_rpweb
```

We will once again start a container in the same fission mentioned above.

```
docker run -idt -v pkgShared:/shared -p 4922:4922 --net=host --name pkgbld pkgbld-base
```

Now, a script has been provided in the auto\_rsp directory for convince, All we have to do is execute that script against the container:

```
docker exec -it pkgbld sh /shared/auto_rsp/install_script.sh
```

### Step 4. Run the Servers and Test

We are now ready to start the Reporting/WMS/RPWeb Servers. If you are not still in the container enter one:

```
docker exec -it pkgbld bash
```

Inside of the container, switch to the reporting user, start moca, and exit back to the sudo user:

```
sudo su - rpt91  
rp start  
exit
```

Inside of the container, switch to the WMS user, start moca, and exit back to the sudo user:

```
sudo su - wms91  
rp start  
exit
```

Finally, we will start refs manually. Switch to the refs user, and start the server

```
sudo su - refs91  
cd bin  
./refs-server
```

Now open up a web browser in your local instance, and hit the container using the hostname of your machine and the configured port:

```
http://{HOST_NAME}:4922
```

You should now be able to do smoke testing against the container.