

Getting Started with SERI



**© Copyright 2013 SailPoint Technologies, Inc., All Rights Reserved.**

SailPoint Technologies, Inc. makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SailPoint Technologies shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

**Restricted Rights Legend**. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of SailPoint Technologies. The information contained in this document is subject to change without notice.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

**Regulatory/Export Compliance**. The export and reexport of this software is controlled for export purposes by the U.S. Government. By accepting this software and/or documentation, licensee agrees to comply with all U.S. and foreign export laws and regulations as they relate to software and related documentation. Licensee will not export or reexport outside the United States software or documentation, whether directly or indirectly, to any Prohibited Party and will not cause, approve or otherwise intentionally facilitate others in so doing. A Prohibited Party includes: a party in a U.S. embargoed country or country the United States has named as a supporter of international terrorism; a party involved in proliferation; a party identified by the U.S. Government as a Denied Party; a party named on the U.S. Government's Entities List; a party prohibited from participation in export or reexport transactions by a U.S. Government General Order; a party listed by the U.S. Government's Office of Foreign Assets Control as ineligible to participate in transactions subject to U.S. jurisdiction; or any party that licensee knows or has reason to know has violated or plans to violate U.S. or foreign export laws or regulations. Licensee shall ensure that each of its software users complies with U.S. and foreign export laws and regulations as they relate to software and related documentation.

**Trademark Notices**. Copyright © 2013 SailPoint Technologies, Inc. All rights reserved. SailPoint, the SailPoint logo, SailPoint IdentityIQ, and SailPoint Identity Analyzer are trademarks of SailPoint Technologies, Inc. and may not be used without the prior express written permission of SailPoint Technologies, Inc. All other trademarks shown herein are owned by the respective companies or persons indicated.

Table of Contents

Introduction 4

SERI Basics 5

Getting SERI 5

Subversion Client 5

SERI Checkout from Subversion 6

Updating Your SERI Project 6

Building SERI 7

Obtaining IdentityIQ Bits 7

Configuring SERI Build Process 8

Copying the template properties file 8

Editing the build.properties file 8

Executing Build of SERI 9

Deploying SERI 10

Installation 10

Import 10

IdentityIQ Demo Setup Task 11

SERI Resource Images 12

SERI AD Resource Image 12

Appendix A: Active Directory Data Setup 13

Performing the Setup 13

Appendix B: Configuring VMware for Static IP 14

Assigning IP Addresses for NAT-Configured VMs 14

Assigning consistent IP addresses 14

Introduction

The SE Reference Implementation (SERI) is a standardized collection of IdentityIQ implementation and configuration artifacts. These artifacts are maintained in a centralized version control system at SailPoint (https://svn.sailpoint.com). The SERI “project” maintained in Subversion is based on the SailPoint Standard Services Build (SSB). The SSB provides configuration management and an automatic build process for creating a deployable IdentityIQ WAR file pre-packaged with SERI.

The first step in understanding how to use and contribute back to SERI is learning how to obtain it from Subversion, build it and deploy it to your IdentityIQ sandbox. This document provides details on how to do this.

SERI Basics

# Getting SERI

To ensure SERI can be maintained effectively and released in a controlled fashion, the SERI artifacts are kept in a centralized Subversion project hosted internally at SailPoint. The project is called “seri” and is located here:

**https://svn.sailpoint.com/svn/sales/seri**

## Subversion Client

You must have a suitable Subversion client on your pc/mac to connect to the SailPoint Subversion server and pull down the SERI Project.

**Mac Users**

Depending on what version of Mac OS X you are running, a Subversion client may already be available on your system. You can confirm you have a working client by typing the following from a Terminal shell:

$ svn --version

If you find you don’t have the client, you can use the following resource to get the Apple stuff:

<http://blog.grapii.com/2012/08/svn-missing-in-mac-os-x-10-8-mountain-lion/>

You can also consider these 3rd-party Subversion clients:

* <http://www.smartsvn.com/>
* <http://versionsapp.com/>

**PC Users**

There are multiple Subversion clients you can download for Windows OS. Here are some suggested ones:

* TortoiseSVN - <http://tortoisesvn.net/downloads.html>
* Slik SVN - <http://www.sliksvn.com/en/download>

Using a command-line svn client is not required. If you prefer a graphical interface, use it. The only requirement is that you know how to “checkout” and “update” with your svn client (and, if you want to contribute back to SERI, you know how to “commit”).

## SERI Checkout from Subversion

Armed with your Subversion client (aka, svn), you’re now ready to pull down the SERI project and create your own working copy (i.e., local to your machine).

Here are the steps:

1. Connect to the SailPoint VPN
2. Open a command-line/shell and navigate to the folder where you want to create your local copy of SERI (e.g., /Users/sean.koontz/work)
3. Execute the following command: svn co https://svn.sailpoint.com/svn/sales/seri

***Note****: There are other options you can use to control how the project is checked out. The above command creates a sub-directory called “seri” and copies the SERI project into it.*

Upon completion of the command, cd into the “seri” directory to examine the project structure and content

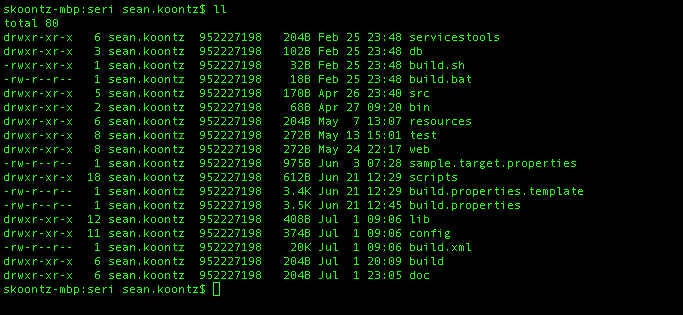


Figure 1: SERI project immediately after checkout

## Updating Your SERI Project

While the svn checkout command **creates** a working copy of the SERI project, the svn update command pulls down changes to an **existing** working copy. This is how you keep your local SERI project updated with the latest version in the Subversion repository.

For example, let’s say a new IdentityIQ workflow is implemented during a POC to meet a use case requirement. This workflow does some really cool things that (a) will make for a great demo scenario and (b) will be useful for other POCs with the same/similar requirement. The SE who implemented the workflow adds it to SERI by creating a new SERI Catalog entry and checking in the artifacts (svn commit).

You learn of this new SERI enhancement and want to use it immediately. To do so, do the following:

1. Open a command-line/shell and change directory (cd) to your SERI project folder
2. Execute the following command: svn update

The following screenshot shows an example of performing an svn update.

Macintosh HD:private:tmp:Terminal — bash — 177×47-4.png

Figure 2: Output from SVN Update command

# Building SERI

With a local copy of SERI on your machine, you’re now ready to configure and execute the build process. The build process is based on a modified version of the SailPoint Services Standard Build template. It uses the Java-based build tool, **Ant**, to automate compilation and packaging of SERI into a deployable IdentityIQ product WAR file. *[****Note****: If you need to download Ant, go here:* [*http://ant.apache.org/bindownload.cgi*](http://ant.apache.org/bindownload.cgi)*]*

The basic steps to building SERI are:

1. Obtain IdentityIQ product (either released or built locally)
2. Configure SERI build process for your environment
3. Execute the Ant build process

***Note***: *There is a Word document on the Standard Services Build (SSB) in the ‘doc’ directory of the SERI project. It provides more details about the SSB. It was created by the SailPoint Services team and is part of the SSB template that was used to create the SERI project.*

## Obtaining IdentityIQ Bits

The SERI build process supports two modes of operation for building against a given IdentityIQ version. These modes correspond to how you’ve obtained the IdentityIQ product bits.

1. ***Self*** – IdentityIQ bits are built locally. Useful for those who checkout IdentityIQ source from Subversion and build IdentityIQ locally on their machine.
2. ***Released*** – IdentityIQ bits are pulled from official release site. Useful for those who don’t build IdentityIQ locally, but instead, use builds generated by Engineering and available through Compass or Shipyard (shipyard.sailpoint.com/images/IdentityIQ).

For example, with Released mode, if you wanted to build SERI with IdentityIQ 6.1p1, you would pull down the 6.1 GA release and the 6.1p1 patch from Compass or Shipyard. You would then point the SERI build process at the location of these downloaded IdentityIQ bits to build SERI (see details in next section).

***Note****: SERI works with IdentityIQ 6.1 or greater. Support for 6.0 was not considered due to the timing of the 6.1 release and the SERI 1.0 GA release.*

## Configuring SERI Build Process

SERI uses a build.properties file to configure and control the build process. The SERI project you checked out has a template that you must copy and modify to configure building SERI in your local environment. This section explains how to do this.

### Copying the template properties file

1. Go to the top-level folder of SERI project
2. Copy the file build.properties.template to build.properties

### Editing the build.properties file

The edits required to the file are minimal and not all are necessary. The edits are categorized as follows:

1. Edits for building SERI [**REQUIRED**]
2. Edits for auto-deploying SERI to a Tomcat/MySQL install on same machine [**OPTIONAL**]

The following screenshot details the edits required for building SERI. Notice the comment character (#) for excluding a property that doesn’t apply to your environment. For example, this screenshot shows a ***Self*** mode configuration in that WARLocation is specified and GALocation is commented out.

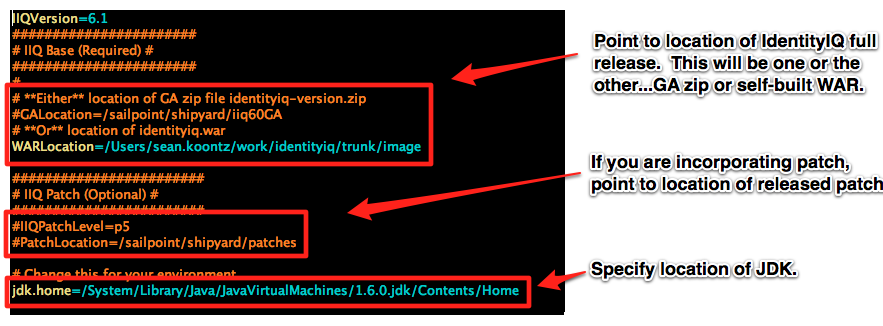


Figure 3: Edits required for building SERI

If you want the SERI build process to auto-deploy the built SERI package to your Tomcat/MySQL environment, you can make these additional edits.

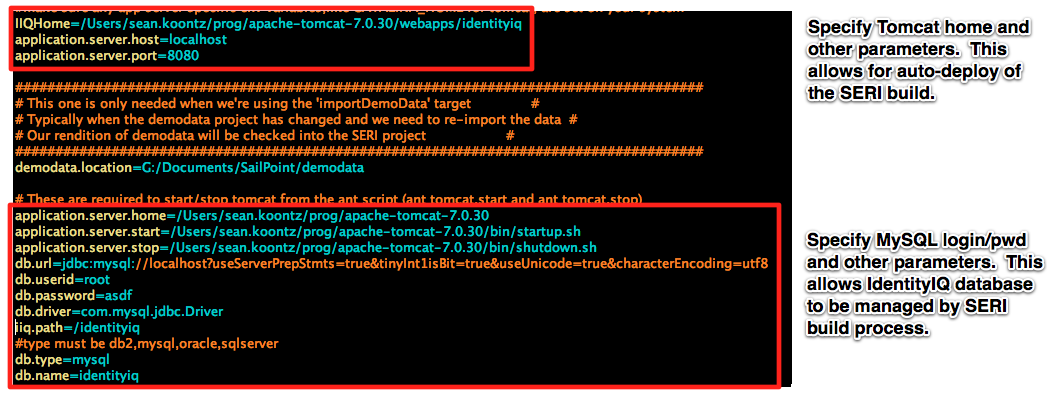


Figure 4: Edits required for auto-deploying SERI to Tomcat/MySQL

## Versioning of build.properties

Sometimes, adding a new feature to SERI requires the addition of more configuration options in build.properties. If this happens, the build.properties will be updated with a header section “since version X”, and the new options. The property build.properties.version will also be updated.

build.properties.version=5

###################

# Since version 5 #

###################

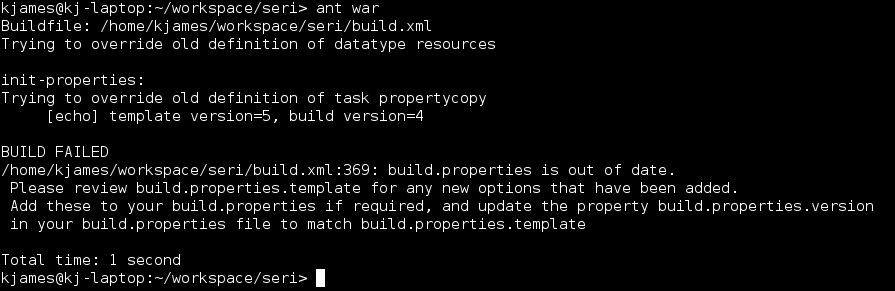
# This is the settings for your AWS domain

accessmgr.aws=true

accessmgr.aws.key=

accessmgr.aws.secretkey=

If this occurs, after you’ve done an svn update and attempt to build, you will get the following message:



At this point you should look at build.properties.template, and see what extra options have been added in the latest version. If you wish to use the new options, copy them to your build.properties and set appropriate values.

Whether you wish to use the new options or not, update the build.properties.version property in your build.properties file to match the version in build.properties.template

## Executing Build of SERI

With the SERI build process properly configured for your environment, you are ready to kick off a build. Using a command-line, navigate to the SERI project. You can type “ant” and hit return to see a list of the SERI build targets.

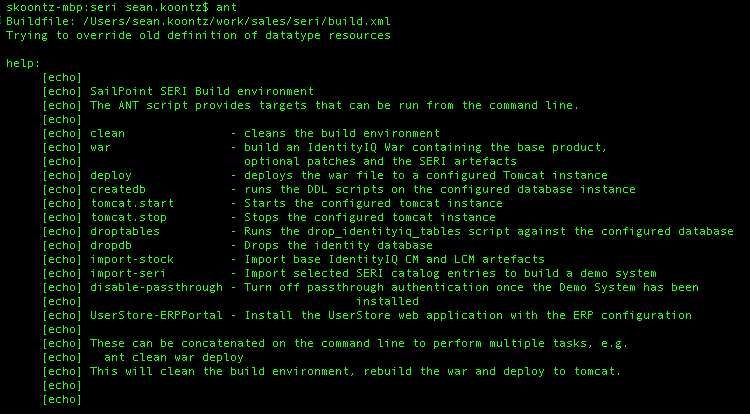


Figure 5: Built-in help for SERI build targets

Next, execute the “war” target to generate SERI. If something is misconfigured or missing for your environment, the build process will error off with a message. Upon a successful build, the ant process will output “BUILD SUCCESSFUL”. Change directory into $SERI\_HOME/build/deploy to find the identityiq.war file.

# Deploying SERI

The output of the SERI build process is an IdentityIQ WAR file bundled with SERI artifacts and setup configuration. This allows us to leverage a standard product install to get SERI deployed into the app server. SERI includes Ant targets for auto-deploying the WAR file to Tomcat/MySQL on the same machine. We will explain these targets, then walk through a manual deploy. This is the typical procedure for using SERI in POCs.

## Automated Deployment

The Ant build files contain a number of targets to automate the deployment of a new SERI instance from the command line.

createdb – Runs the SQL script to set up the IdentityIQ database

dropdb – Drops the IdentityIQ database

import-stock – Runs IIQ Console and imports the stock artifacts for IdentityIQ

import-seri – Runs IIQ Console and imports the SERI artifacts

tomcat.deploy – Deploy the IdentityIQ+SERI war to tomcat over the HTTP management interface

tomcat.undeploy – Undeploy the IdentityIQ+SERI war from tomcat over the HTTP management interface

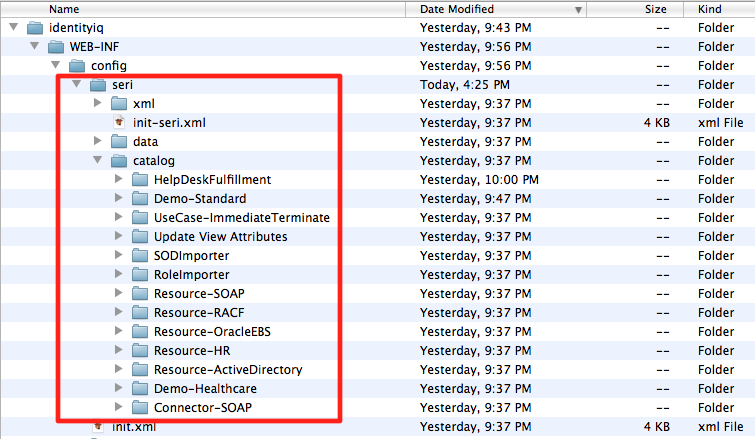
setup-demo – Runs IIQ Console and starts the ‘Setup Demo’ task to import all the SERI resource data, configure roles etc.

It is possible to add many targets to the command line, which will be excuted in the order specified. So, to wipe an existing SERI install, rebuild and deploy a new one, and start the data setup process, you would do:

$ ant tomcat.undeploy dropdb clean war createdb import-stock import-seri tomcat.deploy setup-demo

## Installation

Installing the SERI package is equivalent to installing IdentityIQ. The only difference is that when you’ve completed the install (e.g., unzipped the WAR file into /tomcat/webapps/identityiq), you will have some additional class files and configuration under the WEB-INF folder.



## Import

SERI is a collection of many configurations and use case artifacts. The majority of these are organized under $IIQ\_HOME/WEB-INF/config/seri/catalog. As SERI is used for both demos and POCs, which configurations are imported will depend on how SERI is being leveraged for a particular Sales Engineering engagement (e.g., default sales demo, custom demo, POC).

A default SERI import that supports a Standard Sales Demo (SSD) is a requirement of the SERI initiative. This default import will be continually updated as demonstration scenarios get built out and enhanced. To support the use of real applications and platforms, SERI provides one or more ***resource images***. The first release of SERI includes the AD Resource Image.

***NOTE:*** You need to download the AD Resource Image and configure it for static IP before attempting to use SERI for standard demo purposes. See Appendix B for configuring your VMware environment to use a static IP.

The following steps are used to import the default, SERI-as-a-Demo configuration (assuming a clean install with empty IIQ database):

1. Start the IIQ console
2. Execute the command, import init.xml
3. Execute the command, import seri/init-seri.xml
4. Start Tomcat

## IdentityIQ Demo Setup Task

To initialize a demo environment supported by SERI, there will be a pre-configured IdentityIQ task definition that is part of the init-seri.xml import. Executing this task will perform all of the setup necessary to initialize the demo. This includes aggregations, automatic cert generation and other automated demo setup. The SERI AD Resource Image will be required to be up and running before executing the setup task.

The goal is to automate as much as possible the onerous task of preparing for a demo, while also removing hard dependencies on having to maintain a demo image that needs continual updating and distribution to the SE team and others. The following steps outline how this streamlined demo setup process will work:

1. Deploy SERI-built IdentityIQ WAR file
2. Create IIQ database (e.g., mysql –u root –ppassword < create\_identityiq\_tables.mysql)
3. Run IIQ console and import init.xml and init-seri.xml
4. Start Tomcat
5. Login to IdentityIQ and run “Setup Demo” task
6. Begin demo

The time to perform the above tasks should not take more than 15 minutes.

SERI Resource Images

SERI exists independent of any particular VM. This allows SERI to support technical selling throughout the sales cycle by being a deployable package that can be used in demos, POCs and everything in between.

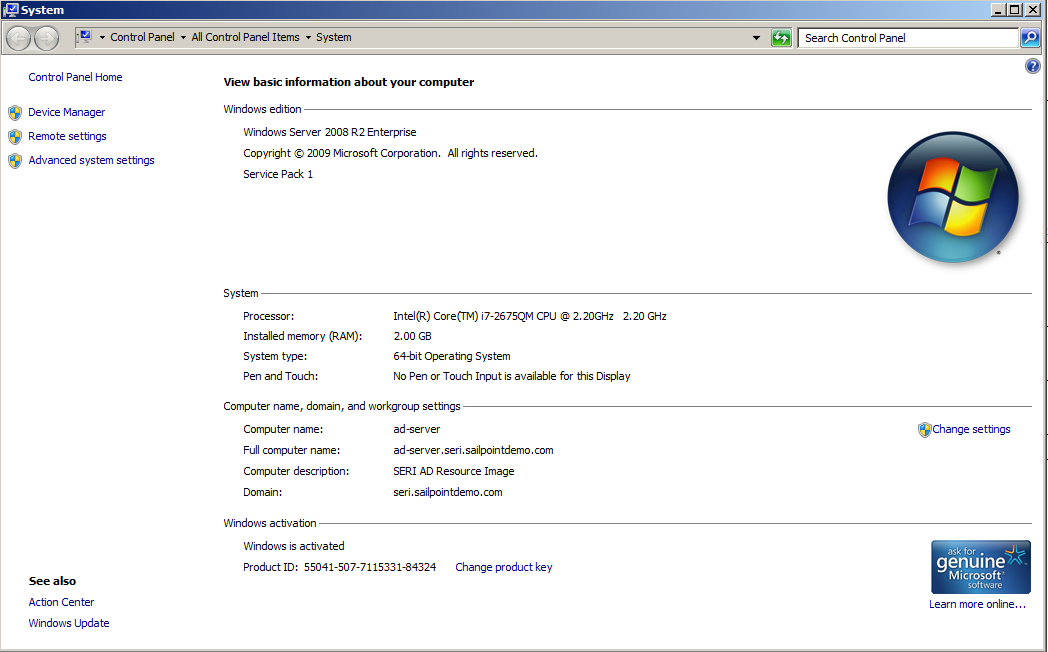
It is expected that some SEs will build SERI and deploy it locally on their laptop in the form of an IIQ “sandbox”. Others might choose to have a lightweight VM that serves as their sandbox (e.g., CentOS image). Whatever the case, it is still absolutely critical to have access to real targets and platforms for learning, demo/POC preparation and compelling demonstration of our comprehensive set of connectors.

To support this, SERI introduces the concept of “resource images”. These are VMs built with specific 3rd-party software installations that map to IdentityIQ Connectors (e.g., AD, Lotus Notes, SAP, OpenLDAP).

# SERI AD Resource Image

Coincident with the introduction of SERI, a new AD Resource Image is introduced. The primary goal of this VM is to provide a real Active Directory target as well as infrastructure previously found in the SSD image. The standard SERI demo requires this VM to be up and running.

The VM specs are as follows:



Appendix A: Active Directory Data Setup

To support demos/POCs and SE sandbox training, SERI includes artifacts and use cases for working with a real Active Directory environment. This appendix provides details on seeding the AD environment with the data the use cases are dependent upon.

***NOTE****: The pre-configured SERI AD Resource Image has this data setup already in place. If you need to create a new Windows image or wanted to reset your copy of the SERI AD Resource Image, use these steps to do so.*

# Performing the Setup

Copy the SERI AD Demo Data zip file to the target AD server. This zip file is located in the SERI project under resources/Active Directory Setup/createDemoData.zip. On the target AD server, perform the following steps:

1. Unzip the zip to C:\SailPoint\seri\data\AD\createDemoData
2. This will extract four files into the directory, three data files and a VBScript file
3. Execute the VBScript file, CreateThem.vbs

You can modify the VBScript to change import semantics and add/remove data from the data files to suit your needs.

Appendix B: Configuring VMware for Static IP

The AD Resource Image provided with SERI is configured for NAT (vmnet8) to allow connectivity to the Internet and other external hosts. Because we have IdentityIQ Application configuration that points at the image via IP address, it is desirable for these IP addresses to be static.

I found many references on the Internet for setting this up. The below info is from the following site: <http://nileshk.com/2009/06/24/vmware-fusion-nat-dhcp-and-port-forwarding.html>

# Assigning IP Addresses for NAT-Configured VMs

In VMWare Fusion, when using NAT, you often times want the IP addresses for the VMs to always be the same, so that you can connect to them from the host using the same IP address every time (and probably assign an entry in your /etc/hosts file). You may also want to connect to the host from the guests using a constant IP address. And finally, with VMs that use NAT, you may want to setup port forwarding so that machines outside of the host can connect to services on your VM (and this relies on the IP addresses remaining the same).

The configuration files for doing this reside in /Library/Preferences/Vmware Fusion/vmnet8 for **VMWare Fusion 4** and later (the NAT interface is called “vmnet8” hence why configuration for it is here). The files are specifically dhcpd.conf and nat.conf. These files may be read-only (even for root), so make sure to give root write permission before you edit them.

Note that changes to these files require a restart of the VMWare’s networking. For **VMWare Fusion 4** and later, simply restart the app.

## Assigning consistent IP addresses

As explained by [this blog entry](http://www.thirdbit.net/articles/2008/03/04/dhcp-on-vmware-fusion/) you can edit the dhcpd.conf file to configure VMWare Fusion’s DHCP server to assign specific IP addresses to a VM based on its MAC address. You should see something like this in your dhcpd.conf file:

subnet 192.168.87.0 netmask 255.255.255.0 {

range 192.168.87.128 192.168.87.254;

option broadcast-address 192.168.87.255;

option domain-name-servers 192.168.87.2;

option netbios-name-servers 192.168.87.2;

option domain-name “localdomain”;

option routers 192.168.87.2;

}

For our IP address assignments, we want to pick an IP address that is **outside** of the range that is set above. So in my case I picked 192.168.87.100 and 192.168.87.101 for two VMs that I wanted to configure.

We also need to get the MAC addresses for the VMs. We can either get it from the VM’s .vmx file under the property ethernet0.generatedAddress, or if the VM is currently running, we can run ifconfig if it is UNIX/Linux or ipconfig /all if it is Windows.

So if we were to configure a VM to have fixed IP addresses, we would add something like this to the dhcpd.conf file:

host seri-ad {

hardware thernet 00:0c:29:02:00:f1;

fixed-address 192.168.87.100;

}

Document Revision History

|  |  |  |
| --- | --- | --- |
| **Revision Date** | **Written/Edited By** | **Comments** |
| April 23, 2013 | Sean Koontz | Created… |
| May 20, 2013 | Sean Koontz | Major additions to support SERI 1.0 Alpha Release |
| June 28, 2013 | Sean Koontz | Updates for SERI 1.0 Beta Release |
| Nov 21, 2013 | Kev James | Added ant targets, build.properties.version. Converted all filenames and console I/O to ‘console’ quick style. |