Grow your own Evergreen!

Michele M. Morgan
Technical Support Analyst
NOBLE, North Of Boston Library Exchange
Why Do You Need Your Own Evergreen?

Test Drive and Learn About the Evergreen System
Test Existing and New Functionality
Explore New Releases
Test and Confirm Launchpad Bugs
Test and Sign off on Launchpad Bugs
Fix Bugs!
Why Grow Your Own?

It’s YOUR System

You Can Get Under the Hood

You Can Get Adventurous and Try Things You Would Not on Other Systems You Use or Have Access To

You Can Break It Without Inconveniencing Anyone Else

If You Break It, You Can Just Start Over

Great Learning Opportunity!
What Skills Do You Need to Build Your System?

That Depends on Your Goals…

The NewDevs Page on the Evergreen Wiki is a Good Reference

New Developers Working Group (Terran McCanna++)

New Developers Working Group
The New Developers Working Group was formed at the 2019 Evergreen International Conference with the intent of gathering together people in the Evergreen community who have interest in learning more about coding for Evergreen, but very little experience. Our goal is to pool our current knowledge, share and review code, and learn together. At times we will invite experienced developers to assist us with reviewing our code or give us mini-tutorials on specific topics.

List-Serv
Sign up for the Evergreen New Developers group mailing list at: http://list.evergreen-lts.org/mailman/listinfo/eg-newdevs

Meetings
The New Developers Working Group meets the third Wednesday of each month at 3pm Eastern. Login information will be sent to the list-serv prior to each meeting.
Useful Skills to Have (Or Learn Along the Way)

Command Line Interface
Navigating Linux File Systems
SQL
IP Addresses and Basic Networking
Git for Testing, Signing Off, Fixing Launchpad Bugs
How Do We Build Our System?
We Could:

Find a Piece of Hardware
Install Ubuntu or Debian OS

Follow the Evergreen Installation Instructions from Evergreen Downloads:

https://evergreen-ils.org/documentation/install/README_3_7.html
# Installing the Evergreen server

## Table of Contents
1. Preamble: referenced user accounts
2. Preamble: developer instructions
3. Installing prerequisites
4. Extra steps for web staff client
   4.1. Install AngularJS files for web staff client
   4.2. Install Angular files for web staff client
   4.3. Install OPAC skin dependencies
5. Configuration and compilation instructions
6. Installation instructions
7. Change ownership of the Evergreen files
8. Run Idconfig
9. Additional Instructions for Developers
10. Configure the Apache Web server
11. Configure OpenSRF for the Evergreen application
   11.1. Configure action triggers for the Evergreen application
12. Creating the Evergreen database
   12.1. Setting up the PostgreSQL server
   12.2. Creating the Evergreen database and schema
   12.3. Loading sample data
   12.4. Creating the database on a remote server
13. Starting Evergreen
14. Testing connections to Evergreen
15. Getting help
16. License
Installing the Evergreen server

Table of Contents
1. Preamble: referenced user accounts
2. Preamble: developer instructions
3. Installing prerequisites
4. Extra steps for web staff client
   4.1. Install AngularJS files for web staff client
   4.2. Install Angular files for web staff client
   4.3. Install OPAC skin dependencies
5. Configuration and compilation instructions
6. Installation instructions
7. Change ownership of the Evergreen files
8. Run Ideconfig
9. Additional Instructions for Developers
10. Configure the Apache web server
11. Configure OpenSPR for the Evergreen application
    11.1. Configure action triggers for the Evergreen application
12. Creating the Evergreen database
    12.1. Setting up the PostgreSQL server
    12.2. Creating the Evergreen database and schema
    12.3. Loading sample data
    12.4. Creating the database on a remote server
13. Starting Evergreen
14. Testing connections to Evergreen
15. Getting help
16. License
We Can Instead:

Build a Virtual Ubuntu Server on Our Desktop or Laptop Workstations

Install Evergreen on Our VM Using Scripts That Do All the Work
Why a Virtual Server?

Can Run on Your Workstation
Low Overhead
Invoke When Needed
Save State
Can Have Multiples
Disposable - But in a Good Way
Note: You Will Need a Powerful Workstation to Build and Run Your Virtual Evergreen
Components

Platform for Virtual Machines - VirtualBox
Ubuntu Server Installation Download
Script to Install Evergreen - Ansible
VirtualBox

https://www.virtualbox.org/wiki/Downloads
Scripts for Building Evergreen: Ansible (Bill Erickson++)

https://github.com/berick/evergreen-ansible-installer
Quick How-To

1. Install Ubuntu 20.04

2. Clone and run the Ansible playbook using an Ubuntu login which has sudo (but not as root).

```bash
# Use the latest ansible version for Ubuntu
sudo apt-get install software-properties-common # sometimes necessary
sudo apt-add-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install git ansible

## Install with the sample locales defined in translations.yml
# Install with a different deployment user (named 'deploy') on a remote machine
# sudo ansible-playbook playbook.yml -e hosts=other.example.org -e deploy_user=deploy

1. In Chrome/FF navigate to https://<HOSTNAME>/eg/staff/ and click through the SSL warning to access the staff client.
```
Scripts for Building Evergreen: Docker (Blake Henderson++)

https://hub.docker.com/r/mobiusoffice/evergreen-ils


Will load a self contained ubuntu 16.04 server with postgres 9.5/9.6. Web based staff client and xul runner staff client included (pre EG3.2). Once running, you should see an ansible script kick off, which will end with PLAY RECAP *****
127.0.0.1 : ok=74 changed=53 unreachable=0 failed=0

Once you see that, the server is setup and ready for connections. Keep in mind if you press CTRL+C - the server will die. Please use CTRL+pq instead to leave it running.

The SSH login to the docker container is defaulted to: user/password The Evergreen global admin account is defaulted to: admin/demo123 The Git repo that is used to create the images is here: https://github.com/mcoia/eg-docker
Ubuntu - https://ubuntu.com/download/server
Downloaded Ubuntu Server Installer

For Use With VirtualBox:

ubuntu-20.04.2-live-server-amd64.iso
Let’s Grow an Evergreen!
Disclaimer: I Am Not A Sysadmin

Much Trial and Error Went Into Developing This Process

There May Well Be Better Ways to Accomplish Some of These Steps and I Am Always Willing to Learn Ways to Make Life Easier
Build Your Ubuntu VM Using VirtualBox
Build Your Ubuntu VM
Build Your Ubuntu VM

Name Your Server
Increase the Memory Size ~ 8 GB
Create a virtual hard disk now
Click Create
Build Your Ubuntu VM

Increase the Hard Disk Size ~ 40 GB

Click Create
Build Your Ubuntu VM
Build Your Ubuntu VM

Change from NAT to Bridged Adapter

Click OK
Build Your Ubuntu VM
Build Your Ubuntu VM

Select Your Ubuntu .iso File
Downloaded Earlier

Click Start
Build Your Ubuntu VM

Use the Arrows, Space, and ENTER Keys to Navigate Through the Installation

On Most Screens Just Continue to the Next Screen
Build Your Ubuntu VM

You Will Need to Confirm Destructive Action
Build Your Ubuntu VM

When Prompted, Provide:

Your Name
Your Server’s Name
Your Ubuntu Username
Your Ubuntu password

Navigate to Done and Press Enter
Build Your Ubuntu VM

Install the OpenSSH Server if You Intend to Connect to Your Server Using a Terminal Application or Text Editor
Build Your Ubuntu VM

Install Finished!

Reboot!
Build Your Ubuntu VM

Login With Your Username and Password
Build Your Ubuntu VM

Note Your Server’s IP Address

10.0.2.15
Congratulations! You Have an Ubuntu System!
Save the State of Your Server When Closing
Now Let’s Install Evergreen
A Note About Users

michele - $ - User Created During Ubuntu Install, For Logging Into Your VM and Running the Install Script

root - # - All Powerful User

opensrf - $ - User That “Owns” Evergreen
michele@ubuntu09:~$ sudo su - root
[sudo] password for michele:
root@ubuntu09:~# su - opensrf
opensrf@ubuntu09:~$ exit
root@ubuntu09:~# exit
michele@ubuntu09:~$
Ansible Install Instructions

As michele user:

$ sudo apt-get install software-properties-common
$ sudo apt-get update
$ sudo apt-get install git ansible
$ git clone --branch ubuntu-18.04 https://github.com/berick/evergreen-ansible-installer.git
$ cd evergreen-ansible-installer
$ sudo ansible-playbook playbook.yml
Installing Evergreen

Ansible Script in Action
Installing Evergreen

Finished!
Connecting to Evergreen

In Your Browser, Go To http://<vm ip address>/eg/staff/

Username:
admin

Password:
demo123
Connecting to Evergreen

http://<vm ip address>/eg/staff/
Connecting to Evergreen - OPAC

http://<vm ip address>
Cheatsheet - Fix the OPAC (Jeff Davis++)

As opensrf user

$ cd /openils/var/web/opac/deps

$ npm install
Stock Database

“Concerto” Database - Small Collection of Users, Records, Holdings and Transactions

Concerto Logins:

Congratulations! You Have an Evergreen System!
A Few Ways to Customize Your System
Phppgadmin - A GUI Interface to the Evergreen Database
Cheatsheet - Install Phppgadmin

# apt-get install phppgadmin

# vi /etc/apache2/conf-enabled/phppgadmin.conf

Edit the file to comment out Require local

  # Require local

Save the file and restart apache

# systemctl restart apache2.service

Browse To http://<ip address>/phppgadmin

Username, Password: evergreen
Cheatsheet - Setting Up Git

$ git config --global user.email "mmorgan@noblenet.org"

$ git config --global user.name "Michele Morgan"

See the NewDevs Page for Instructions for Requesting Permission for the Evergreen Git Repository
Cheatsheet - Adding the Working Git Repository

As the opensrf User

$ cd /home/opensrf/Evergreen

$ git remote add working git@git.evergreen-ils.org:working/Evergreen.git

Show Your Repositories

$ git remote -v
Tip: Cloning Your VM Can Save You Some Work

Once You Have Your Evergreen System Running, Don’t Touch It!

If You Plan on Tinkering, Make One or Several Clones

Clones Are Exact Duplicates of the System You Just Built

You Can Feel Better About Breaking Your System When You Can Easily Make Another Copy
Clone Your VM
Clone Your VM
Clone Your VM

Clone Virtual Machine

Clone type

Please choose the type of clone you wish to create.

If you choose Full clone, an exact copy (including all virtual hard disk files) of the original virtual machine will be created.

If you choose Linked clone, a new machine will be created, but the virtual hard disk files will be tied to the virtual hard disk files of original machine and you will not be able to move the new virtual machine to a different computer without moving the original as well.

If you create a Linked clone then a new snapshot will be created in the original virtual machine as part of the cloning process.

- Full clone
- Linked clone

Clone	Cancel
Clone Your VM
Clone Your VM

- Ubuntu05
- Ubuntu06
- Ubuntu07 (Clone of ubuntu)
- Ubuntu08
- Ubuntu09

**Audio**
- Host Driver: Windows DirectSound
- Controller: ICH AC97

**Network**
- Adapter 1: Intel PRO/1000 MT Desktop (Bridged Adapter, Intel(R) WI-FI 6 AX201 160MHz)

**USB**
- USB Controller: OHCI
- Device Filters: 0 (0 active)

**Shared folders**
- None

**Description**
- fresh build post 3.7 release
- 192.168.1.193
- Use this to clone other vms
More Cheatsheets
Cheatsheets - Legend

$ - Means You’re the opensrf User

# - Means You’re the root User
Cheatsheet - IP Addresses

Find Your Server’s IP:

$ ifconfig

Get a New IP

$ sudo dhclient -r

$ sudo dhclient
Cheatsheet - Restart Evergreen

$ osrf_control --localhost --stop-all
$ osrf_control --localhost --start-all

# systemctl restart apache2.service
# systemctl restart websocketd-osrf
Cheatsheet - Recompile and Copy Angular Files

$ cd /home/opensrf/Evergreen/Open-ILS/eg2

$ ng build --prod

My Favorite Cheat

$ <UP ARROW>
Bug Testing and Signoff Process

Start Your VM

$ cd /home/opensrf/Evergreen

Make Sure Git is Up To Date

$ git pull

$ git fetch working
Bug Testing and Signoff Process

Create and Switch To a New Git Branch

$ git checkout -b <branchname> origin/master

Cherry-pick the Commits for the Patch You Are Testing - Oldest to Newest

$ git cherry-pick -s <first 7 characters from commit>
Bug Testing and Signoff Process

Put Patched Files into Place on Your VM

Depending on the Patch, This Could Involve:

- Copy the Patched File to its Installed Location
- Compile and Copy Angular Files into Place
- Rebuild Evergreen
- Restart Evergreen

Login to Your System and Test the Patch
Bug Testing and Signoff Process

Login to Your System and Test the Patch

When Testing, it’s Important to Consider:

- Permissions
- Library Settings
- Global Flags
- Staff View
- Patron View
- Workflows
Bug Testing and Signoff Process

If All Looks Good, Push Your Signoff

```
$ git push working <branchname>:user/mmorgan/<remote branchname>
```

Update the Launchpad Bug
Questions?
Thank You!

Michele M. Morgan
Technical Support Analyst
NOBLE, North Of Boston Library Exchange