



## **TREP-In-A-Box blueprint**

Document date : 2014-09-03

## Contents

1. Overview .....	3
1.1 Overall design.....	3
1.2 Virtualization technology used.....	4
1.3 Scripted creation of virtual hosts (zones) .....	5
1.4 More virtualization ? .....	5
2. Building TREP-In-A-Box.....	7
2.1 Requirements.....	7
2.2 Steps.....	7
3. Components that require Solaris 10 .....	9
4. Performance.....	10
5. Appendix A – Downloading Solaris 11 Install Image .....	11
6. Appendix B - Install Solaris VM in VirtualBox.....	12
7. Appendix C – Solaris 10 inside Solaris 11 .....	19
7.1 Download Solaris 10.....	19
7.2 Creating the Flash Archive.....	21
7.3 Creating the Solaris 10 zone on the target system .....	21
7.4 Configure the new Solaris 10 zone .....	22
8. Appendix C - Port forwarding in VirtualBox .....	25
8.1 Reaching the virtual hosts from outside via SSH .....	25
8.2 SSL and RSSL connection into ADS .....	26
8.3 DACS Web UI.....	26
8.4 Connections from inside to outside .....	27

# 1. Overview

---

This document is all about how to run a full Thomson Reuters market data system (TREP) on the minimum amount of hardware. It's about test/demo/UAT systems, not about production systems. However, we still want our test system to mimic our production system as much as possible. We'll present a way to run a complete TREP solution, complete with separated networks, separate hosts for each TREP component, etc ... on a laptop!

We will present a way to configure a TREP non-production system with the following attributes:

- Minimum amount of hardware (we'll present how you can run a full TREP system, bells and whistles, on a laptop).
- Separate backbone for inter-TREP traffic, in particular for multicast traffic. This backbone must be 'private' to TREP.
- Hosts that resemble those in production, i.e. you can move a host from production to test and vice versa. You can also without problems move hosts from physical to virtual and vice versa.
- It's a really an all-in-a-box solution but without sacrificing the requirement to make it look like a true production system.
- Low cost. In fact the only cost here is the TREP licenses.
- Scripted creation of virtual hosts and virtual network components.

To accomplish the above we make heavy use of virtualization and in particular we make use of virtualized networking<sup>1</sup>.

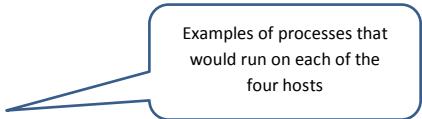
In terms of hardware all you really need is a single X86 box, with at least one NIC and at least 4 GB RAM. That's not a lot for a full TREP system!. In fact most of us have a box just like that right in front of us.

## 1.1 Overall design

We could of course put all TREP components into the same host. While such design will of course be efficient with regards to system resources it is very far from resembling a production system. Therefore we use the traditional design where each host only has only one primary role.

Traditionally, on a TREP production system, we would want to have a separate backbone for inter-TREP traffic. Most TREP systems are configured to use multicast for communication between the ADS and ADH components and this type of traffic we really want to keep private to the TREP so that there's no risk of "polluting" the rest of the network. The other advantage of keeping the backbone private to TREP is that we can configure and tune this part of the network solely for the purpose of TREP.

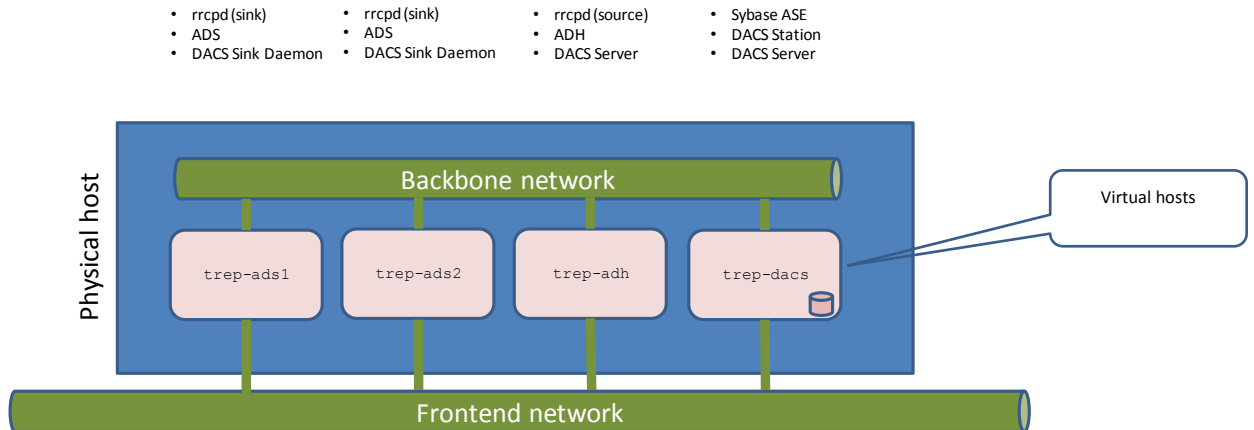
Here's an example of the layout we are working with in this paper:



Examples of processes that would run on each of the four hosts

---

<sup>1</sup> For our scenario we may also use the word SDN (Software Defined Networking), meaning network components (NIC, bridges, routers) that are defined in software as opposed to being physical devices.



The environment has four hosts: two ADS hosts, one ADH host and a DACS host. This is merely an example which you can expand upon. You can have more ADS or ADH hosts. For example if you want to be able to use the test system for testing out things like Source Hot Standby feature then you would want to put in at least one more ADH host. You can also expand the example with non-core TREP hosts, such as a DTS/ATS.

## 1.2 Virtualization technology used

This paper uses Solaris 11 virtualization technologies. Linux is another fine alternative for TREP installations but doesn't have the features we make use of here to keep the cost and hardware requirements at a minimum. You could probably achieve the same with Linux but you would have to bring in third party software<sup>2</sup>.

There are two Solaris specific features we make use of here:

- **Solaris Zones.** This is a virtualization technology based on "containerization" which is a very low-overhead technology and is the reason why we can spin up so many virtual hosts on so little hardware. Solaris Zones technology range from extreme low-overhead scenario where the zones share everything (same kernel, same IP stack) to the situation where they share very little (each zone runs its own kernel, each zone runs its own IP stack). In this paper we choose a configuration where the zones share the same kernel as their host but each zone uses its own IP stack. This means that we can tune the IP stack on a per-host basis just like we would do on a non-virtualized production system.

A zone is a virtual host. The application running inside the zone will not know that the host is virtual.

The words 'host' and 'zone' are used interchangeably in this document.

- **Solaris SDN.** The ability to fully virtualize network components such as NICs, routers and bridges, debuted with Solaris 11 in 2011. In this paper the Solaris SDN feature is the reason why we can create a system that so closely mimics a real-world physical hardware TREP production system.

<sup>2</sup> The Linux equivalent of Solaris Zones technology is LXC on top of which several projects are build, such a Docker. It is however fair to say that this technology – although looking very promising – is still some way from being production-ready. Similarly the SDN technology on Linux is also still in its infancy compared to the maturity of the equivalent technology on Solaris. Once Linux matures these technologies we could do the same low-cost environment on Linux as is presented in this paper.

## 1.3 Scripted creation of virtual hosts (zones)

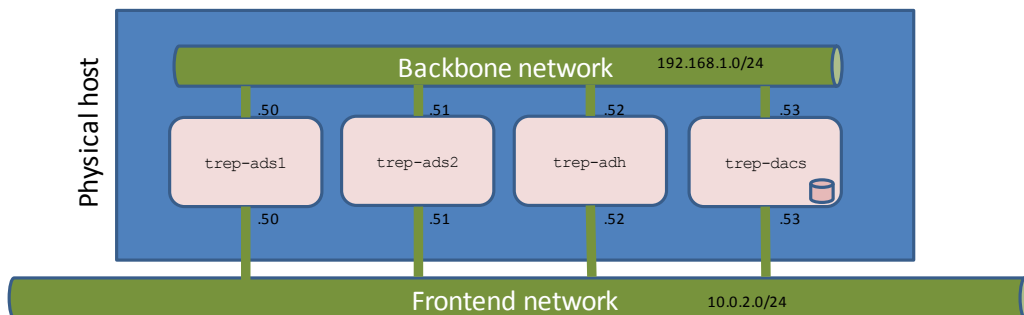
We provide a script to create the zones. Using a script provides consistency and means you can attend to something else while the zones are being created.

The script creates the virtual hosts (the *zones*). When it finishes the following is accomplished:

- Each created host has two network interfaces:
  - Frontend network interface. This IP is assigned the same name as the host, e.g. `trep-ads1`.
  - Backend network interface. This IP is assigned the same name as the host suffixed with '-bb', e.g. `trep-ads1-bb`.
- All created hosts have read/write access to a shared directory, `/reuters`. This directory exists in all hosts and is in fact the same physical directory. The idea is that all TREP software and configuration should be placed in this directory and below.
- All created hosts have a pre-configured user named `radmin`. The idea is that this is the user who will own all TREP processes. He's also the owner of the `/reuters` directory. In daily operation there should be no need to use the `root` account, ever!
- All created hosts know about all the other hosts and can access them by both their frontend name and their backend name. (the `hosts` file on all hosts is amended by the script)
- TREP specific ports are automatically added to `/etc/services` on all hosts.
- Network routes are configured on all hosts to make sure the backbone can do multicast traffic.

Thus we end up with a range of hosts that are perfectly primed for installing TREP into.

An example of an environment that can be created with the script is the following:



All the IP addresses and names can be configured in the script's configuration files.

**Choosing IP addresses for backbone:** This is a no-brainer. This network is private to the box and never escapes the virtual world. Therefore the recommendation is to use the classic IP range for small private networks, e.g. `192.168.1.0/24`.

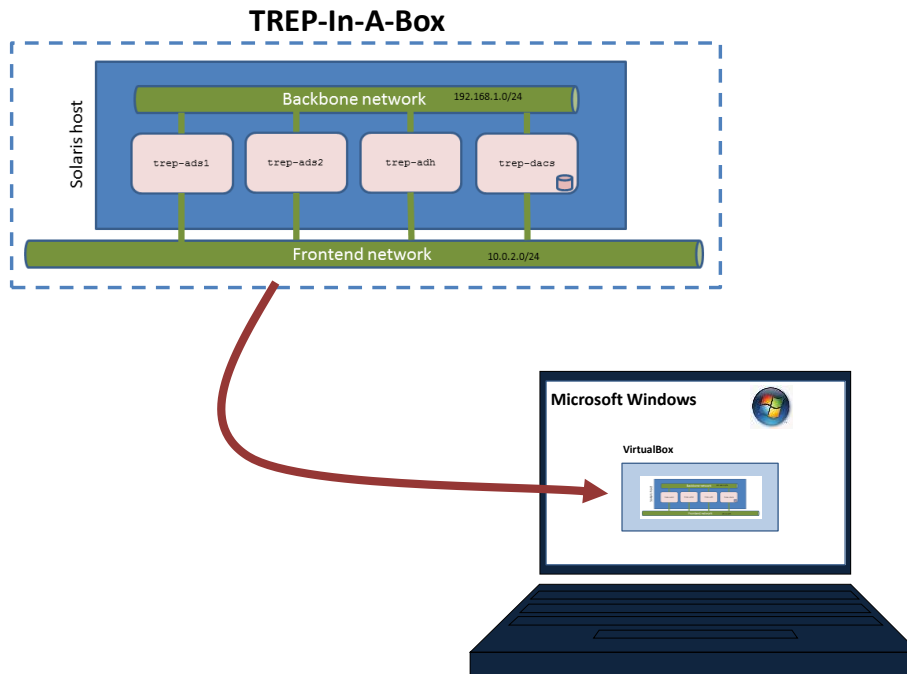
**Choosing IP addresses for frontend:** This choice requires careful consideration as these addresses must co-exist with the rest of your network. In the above example `10.0.2.0/24` is chosen.

## 1.4 More virtualization ?

What if you don't have a box to spare for a TREP test system and really don't want to involve network people, systems people, etc, in your endeavor?

No problem. We can take the design one step further and virtualize the blue box in the drawings. This will be virtualization-in-virtualization and is painless. For this we'll need some workstation virtualization software<sup>3</sup> such as VirtualBox<sup>4</sup> or VMware. For a test system VirtualBox is the most obvious choice. It is free and easy to use so that is what we'll use here.

Here's what it looks like. We simply take the environment inside the dotted line below and put it inside VirtualBox:



VirtualBox's default networking operating model is "NAT" which means that whatever runs inside VirtualBox cannot under any circumstances clash with the network on the outside. Previously – without VirtualBox - there was the problem that the frontend network IP addresses had to be chosen with due respect to the surrounding network. Now, with the VirtualBox "NAT" solution, we no longer have to think about that.

VirtualBox by default creates an "inside" network on 10.0.2.0/24 where the guest VMs resides. Furthermore VirtualBox provides a DHCP server (handing out addresses from 10.0.2.15 onwards), a router (on 10.0.2.2) and a DNS proxy (on 10.0.2.3). All we have to do is configure our zones with static IP addresses in that range with default route of 10.0.2.2 and then they can automatically talk to each other via the frontend network too. The backend network is unknown to VirtualBox as it only exists inside the Solaris world.

<sup>3</sup> By "workstation virtualization software" we mean software which can run on a workstation and can virtualize either a desktop OS or a server OS.

<sup>4</sup> <http://www.virtualbox.org>

## 2. Building TREP-In-A-Box

---

This is intended to be a step-by-step guide on how to build the solution outlined in section 1.4.

The end result will be a complete TREP-In-A-Box solution which is perfectly primed for installing TREP into. We'll not go through how to actually install the TREP components as that is out of scope for this document.

### 2.1 Requirements

#### Hardware

A decent workstation running Windows, Mac OS X or Linux<sup>5</sup> with a Quad Core Intel or AMD processor is required. It can probably run on 4 GB RAM but as always more is better.

#### Licenses

- *Oracle Solaris.* You can download and use Solaris for free under the terms of the [OTN License Agreement for Oracle Solaris](#). The basic rule is that you can use Solaris for free if it is "...only for the purpose of developing, testing, prototyping and demonstration". A TREP-In-A-Box solution would normally comply with this restriction. Make your own judgment.
- *Thomson Reuters TREP licenses.* In order to install TREP on anything you'll need license file from Thomson Reuters. Licenses are typically bound to hostnames and hostids. This hostid is not known until the environment has been booted for the first time so the TREP licenses cannot be ordered in advance. Your contract with Thomson Reuters will normally say how many test systems you can have. Contact Thomson Reuters for more information.

### 2.2 Steps

**Step 1:** Download Oracle Solaris 11. See *Appendix A – Downloading Solaris 11 Install Image* for more information.

**Step 2:** Download [VirtualBox](#) and install it on the designated workstation.

**Step 3:** Install Solaris as a guest OS inside VirtualBox. See *Appendix B - Install Solaris VM in VirtualBox*.

**Step 4:** Configure and install the zones.

Start by uploading the script ZIP file, `zonescripts.zip`, to the Solaris host that was created in step 3. Unpack the zip file.

Configure all relevant values in the three files in `reuters/zone-setup/config` folder. A lot of the values may not need customization.

Now execute the script from the directory where it lives:

```
cd reuters/zone-setup
chmod +x ./create_zones.sh
sudo ./create_zones.sh
```

Creating the zones may take 5-10 minutes per zone depending on the processor speed of your workstation.

---

<sup>5</sup> Section 1.4 only talks about Windows but in fact you can also use Mac OS X or Linux as VirtualBox runs equally well on all three of them.

**Step 5:** (Optional) Install additional Solaris10 zone for DACS. See *Appendix C – Solaris 10 inside Solaris 11* for more information.

**Step 6:** Install TREP software. Prior to this you'll need to acquire a license file from Thomson Reuters. Refer to Thomson Reuters documentation on how to install TREP components.

**Step 7:** Make the solution accessible to the outside world by configuring Port Forwarding rules in VirtualBox. See *Appendix C - Port forwarding in VirtualBox* for more information.



### 3. Components that require Solaris 10

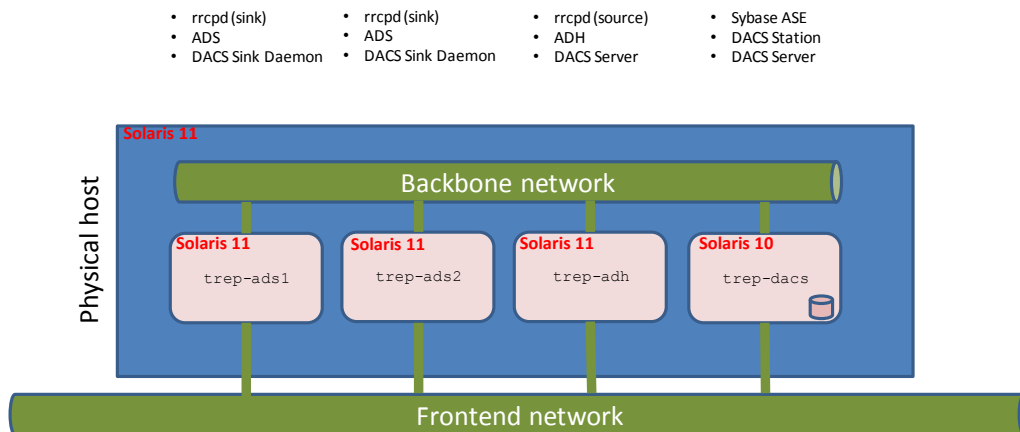
When this document was first prepared in beginning of 2013 DACS was not supported on Solaris 11. On a non-production system we can sometimes afford to be lenient with regards to the official certification policy of the vendor. In other words even if DACS wasn't officially verified by Thomson Reuters to run on Solaris 11 we could still choose to do so at our own peril. In particular on Solaris this is often a rather risk-free strategy as the binary backwards compatibility of Solaris is such a strong virtue of the operating system.

However the problem at the time wasn't that the Thomson Reuters DACS components wouldn't run under Solaris 11 rather it was a problem of the Sybase ASE Installer refusing to execute properly under Solaris 11. Solaris 11 is officially supported from Sybase ASE 15.7 ESD#2 onwards.

UPDATE: For DACS, Solaris 11 x86 is fully supported by Thomson Reuters as of DACS 6.6. You need to be using Sybase ASE 15.7 ESD#2 or later. Thus there's no longer a need for using the method described in this chapter solely for the purpose of DACS. If you have other components that simply won't run under Solaris 11 then you can use the method.

If you truly want a Solaris 10 host for ~~your DACS Station~~ something then this is possible within the framework presented in this document. You just have to use a so called 'solaris10' branded zone. This basically means running Solaris 10 inside Solaris 11. This is a more involved task and is not covered by the scripts presented in this document. It must therefore be done manually.

Your setup will then look something like this (using DACS Station as an example of something that requires Solaris 10):



By default all created zones would be Solaris 11. Now we simply let one of them use Solaris 10.

How to do this is detailed in *Appendix C – Solaris 10 inside Solaris 11*.

## 4. Performance

---

Most of the virtualization in this solution is done by Solaris, not by VirtualBox.

Solaris has incredibly low overhead for its Zones virtualization option. Therefore it is possible to start a rather large number of virtual hosts (zones) inside the Solaris instance without seriously affecting performance. Each zone (before any TREP software is installed) will only require an additional 60-70 MB of memory.

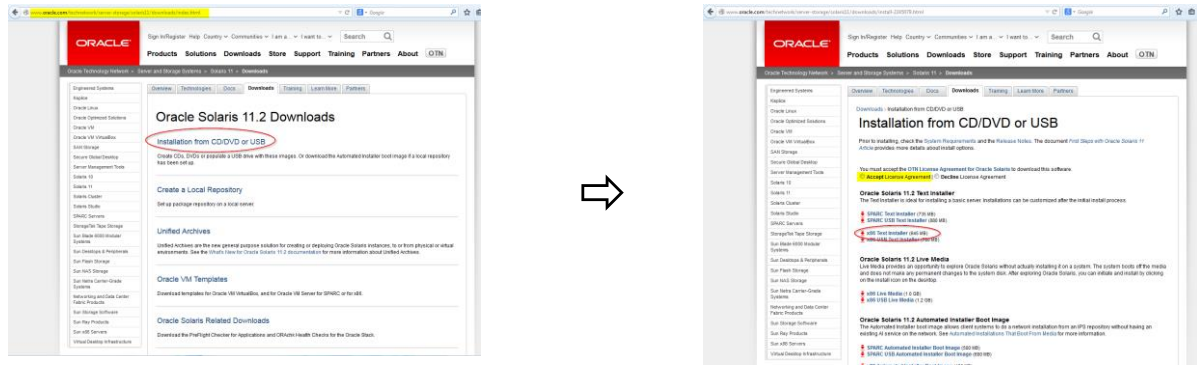
On a fairly standard quad core Intel laptop with 8 GB RAM and running Windows 7 you should be able to easily run 5-6 virtual hosts running ADS, ADH and DACS and at the same time be able use your laptop for other work. DACS Sybase will be the biggest memory hog.

In terms of network performance you'll find that the virtual backbone created in this solution actually performs surprisingly well. Since it is private to the OS instance Solaris is able to cut some corners, for example the network traffic never has to pass through the NIC layer. However the network performance of the backbone will of course be closely linked to available CPU power as everything happens in software.

## 5. Appendix A – Downloading Solaris 11 Install Image

Solaris has a plethora of options for the install 'image'. To make it simple we'll assume that you've downloaded the **Solaris 11 Text Installer** and will be using that for the installation.

<http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html>

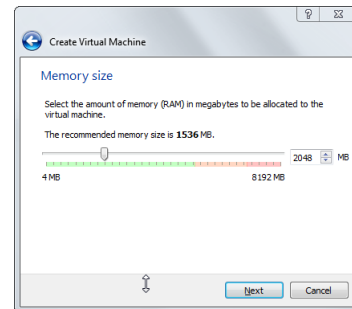
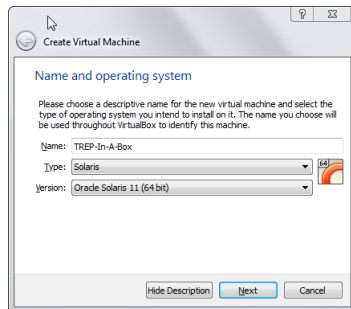


In order to download you'll need to have an Oracle account. It is free.

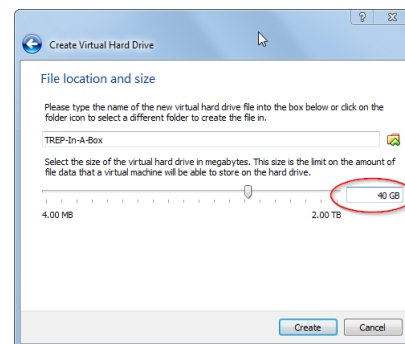
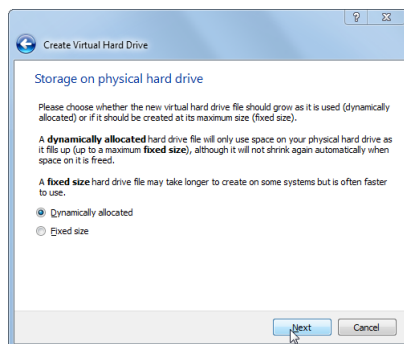
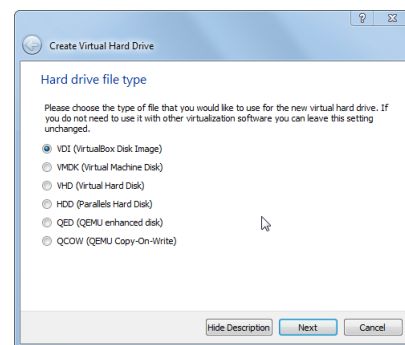
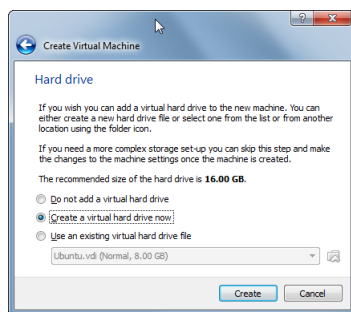
## 6. Appendix B - Install Solaris VM in VirtualBox

Start VirtualBox.

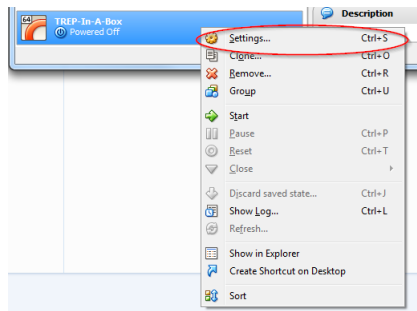
Select from the VirtualBox menu : **Machine** → **New...** and follow the steps below.



We allocate a maximum of 2 GB RAM to the guest OS. This is sufficient for our purpose here. If you have more to spare then you can increase the allocation. VirtualBox will only use it if its needed.

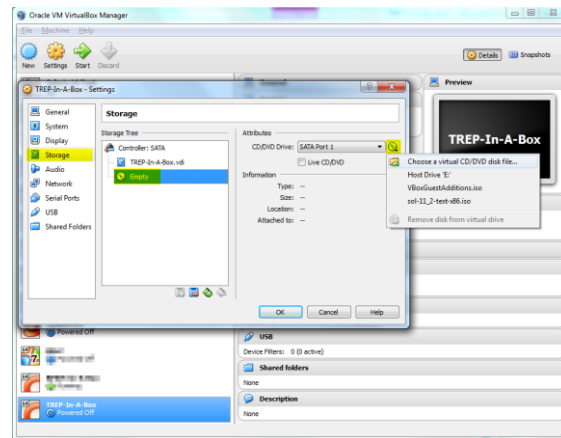


We allocate a virtual disk of 40 GB. This is more than sufficient for the OS and some TREP software. If you wish to run DACS as well then more is needed because of the size of the data in the Sybase Database.

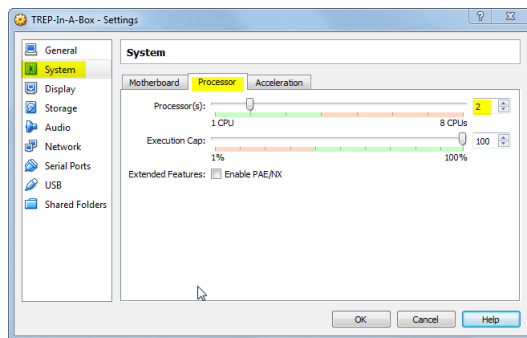


The next step is to "insert" an ISO image into the VM's CD-ROM drive so that when the machine boots it will do so off the Solaris Installer DVD.

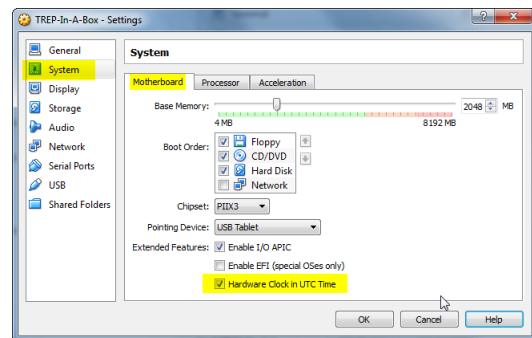
In the left pane of VirtualBox locate your new machine, TREP-In-A-Box, and right-click it. Choose **Settings**.



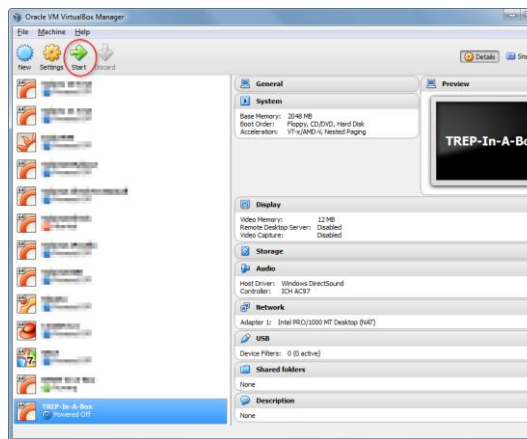
Choose **Storage** and highlight the CD-ROM image in the storage tree. Now click the CD-ROM icon on the right and select **Choose a virtual CD/DVD file**. Here you select your downloaded Solaris ISO image file, most likely it is named `sol-11_2-text-x86.iso` and probably resides in your browser's download folder.



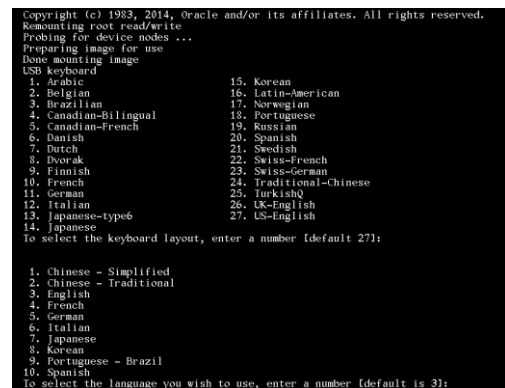
Now choose **System** in the left hand list. Configure the number of CPUs you want Solaris to see. In this example we let Solaris see 2 CPUs<sup>6</sup>.



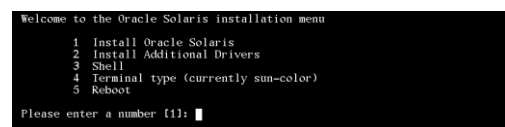
On the Motherboard tab make sure you tick "Hardware Clock in UTC time". Finally press "OK" button to end the configuration.



Now start the machine. The machine will boot off the virtual Install DVD.

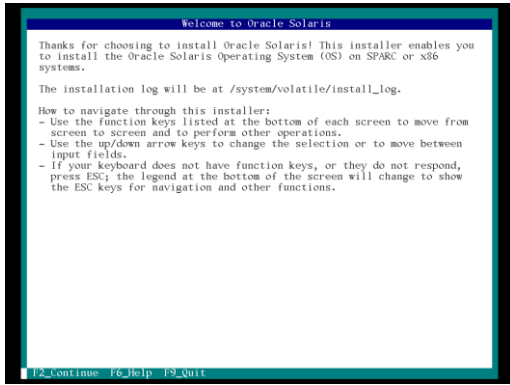


Choose keyboard and language for the installation process. (choice of language should be #3 (English) or you'll not be able to follow this guide)

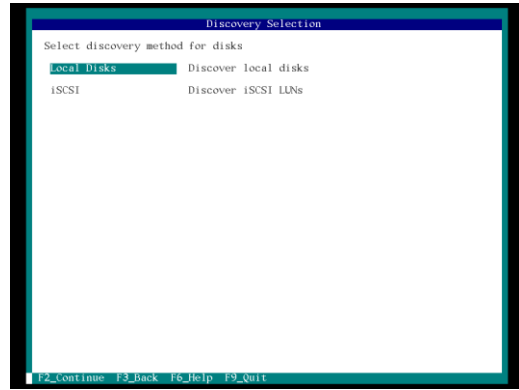


Choose option #1: "Install Oracle Solaris".

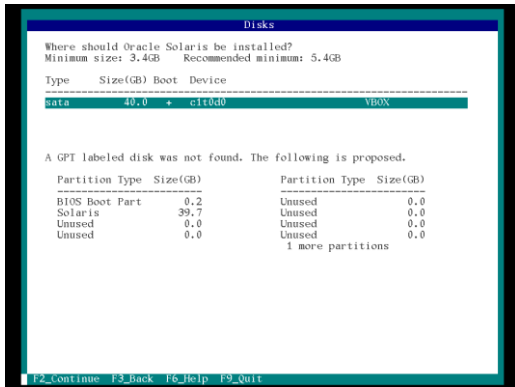
<sup>6</sup> The screenshot is from a laptop running Windows 7. The laptop has a Quad Core CPU with Hyperthreading. This means that Windows sees 8 cores in total. You should however never let guest OS (in this case Solaris) see more CPUs than you have physical cores in the mother host. For the hardware we're using in the screenshots it means the maximum CPUs we should configure for the guest OS to see is 4.



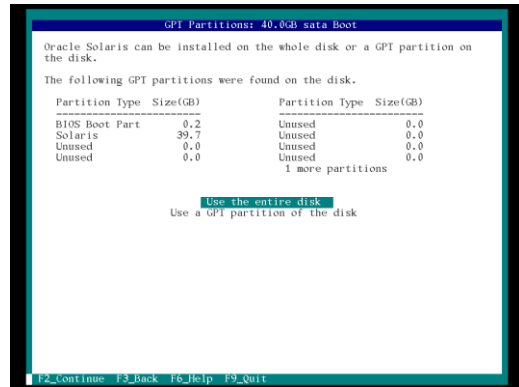
Press F2.



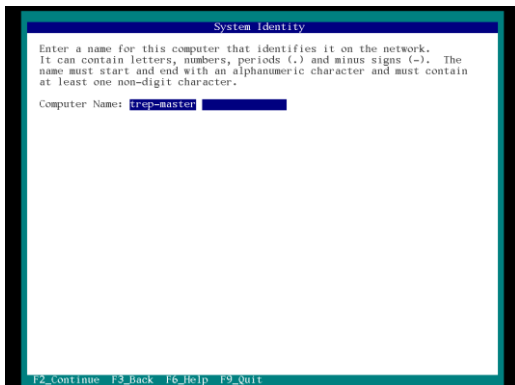
Press F2.



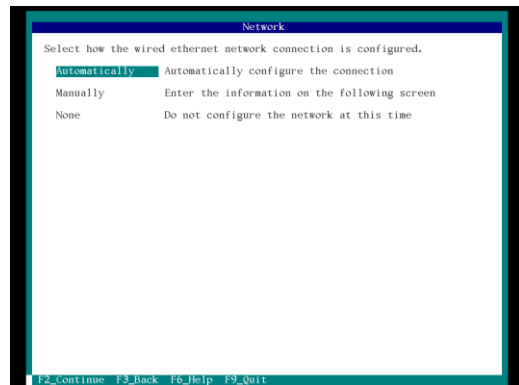
Press F2



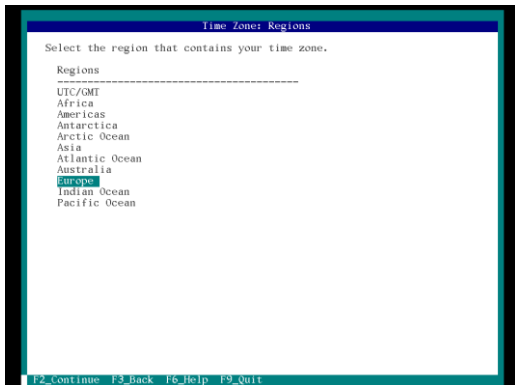
Press F2



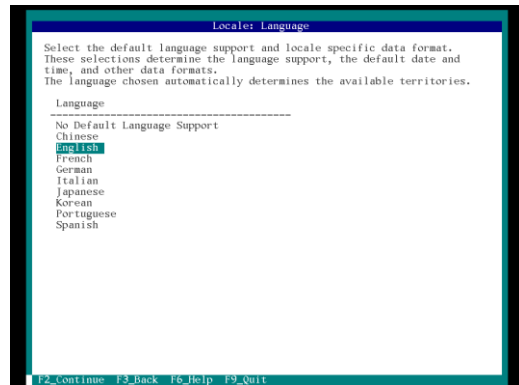
Give the machine a hostname of `trep-master` and press F2.



Leave the Network configuration at default. This means Solaris will use DHCP to configure the network address.

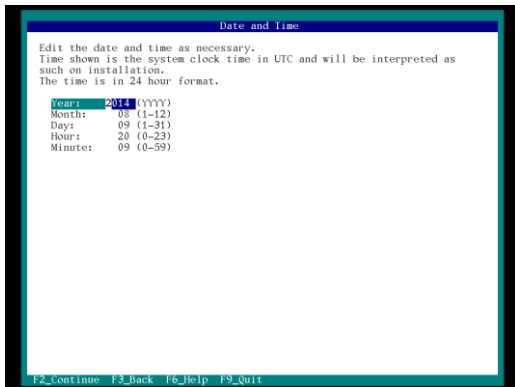


Choose time zone. (there are more screens than shown here)

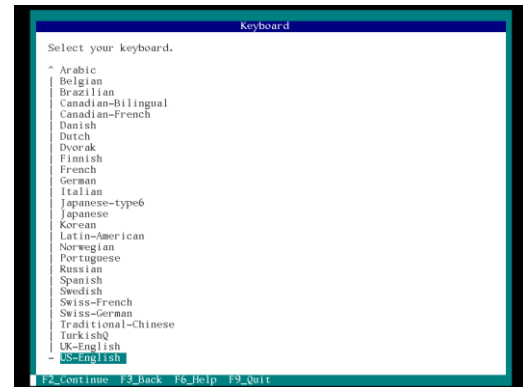


Select the locales you wish to install. You can always install more later. Except potentially for DACS Sybase Database, TREP as such will not care about which locales are installed on the OS. You can even choose "No Default Language"

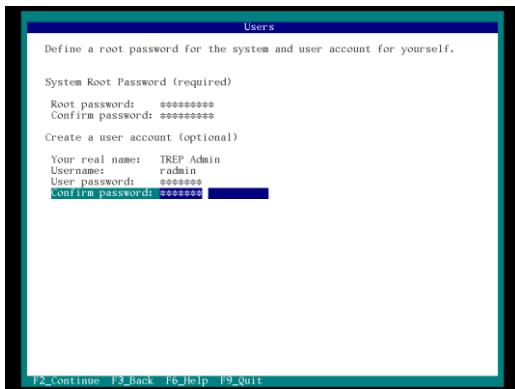
Support" (which means only the C locale will be installed) and TREP will work just fine. Make your selection and press F2. On the next page you'll be asked to choose language territory. Make that choice as well and end by pressing F2.



Press F2 (the time shown should be ok as it is, remember it is in UTC)



Choose your keyboard layout as you want it to be. This time it is for the installed OS, not for the installation process itself. Press F2 when done.

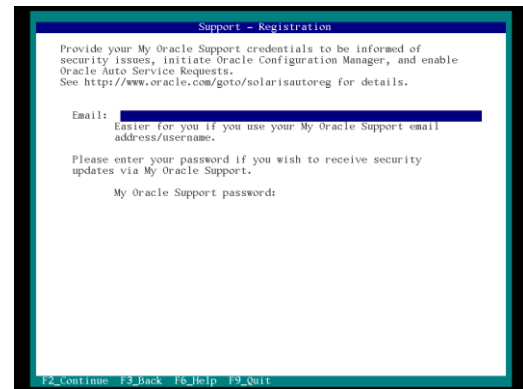


Create a password for root account (and make a note of it).

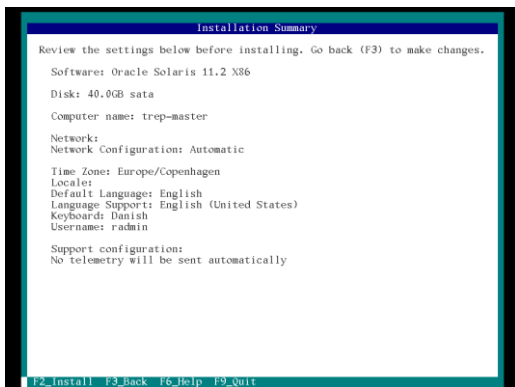
Also create additional user `radmin`. This user will also be administrative. He can do just about the same as 'root' except for security related tasks. For security related tasks he will be able to `su` to root account (provided he knows the password of the 'root' account).

The following passwords are suggested to use:

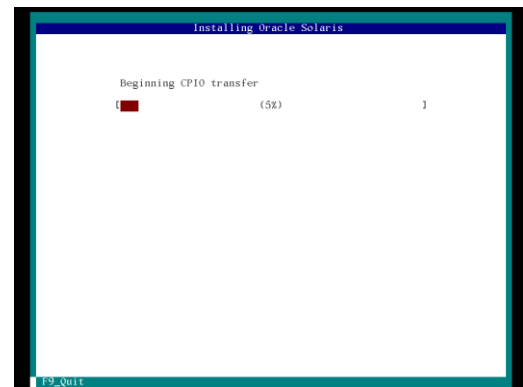
`root` : admin1234  
`radmin` : adm1234



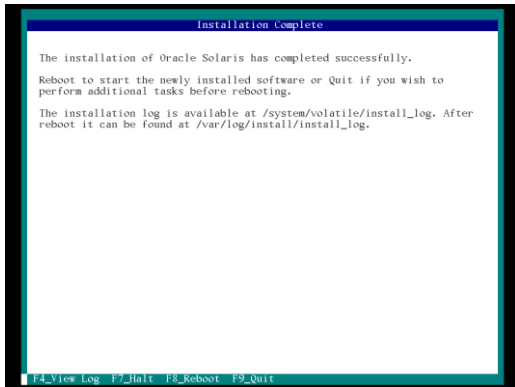
On the Support Registration input your e-mail address if you have an Oracle Support account. If not leave the e-mail field blank. Press F2. If you did not enter an e-mail address you'll be asked to press F2 again.



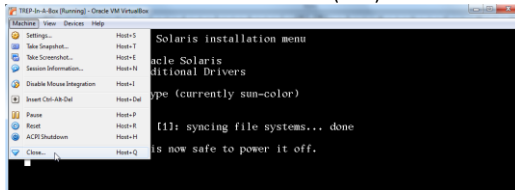
This is the final confirmation page before installation. Verify that everything is as you expect. You can always use F3 to go back and correct your choices. Press F2 to begin installation.



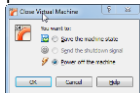
Installation will now begin and may take 10-20 minutes.



Solaris 11 is now installed. Choose F7 (Halt)

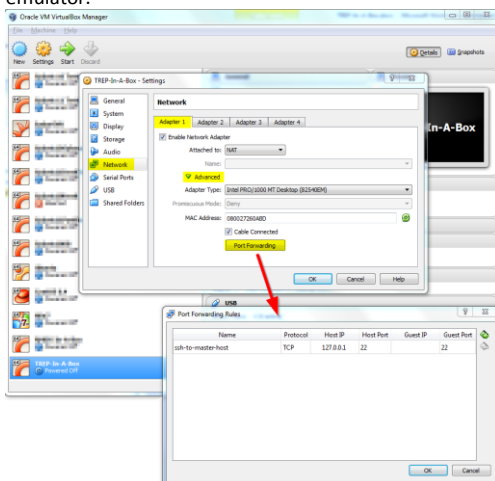


When the message on the screen says "now safe to power it off" choose menu **Machine** → **Close**.



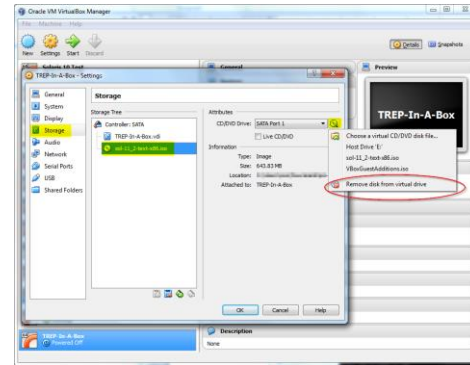
Press OK.

Next step is to set up a Port Forwarding rule so that we can access our newly created host from outside. If we did not do this we would have to work in VirtualBox's Guest Operating System Console Window which is very simple and doesn't have the features of a real terminal emulator.



Set up Port Forwarding as in the screenshot above. The VM will now be accessible by SSH'ing to localhost with your favorite terminal emulator solution.

Click OK a couple of times to exit the setup.



Now the machine is stopped. We now need to remove the virtual DVD disk we have in the DVD-drive. Otherwise the machine will boot off the Install DVD again and the install process will repeat itself.

Right click the machine's name in VirtualBox's left pane, choose **Settings** and remove the disk as in the screenshot above.

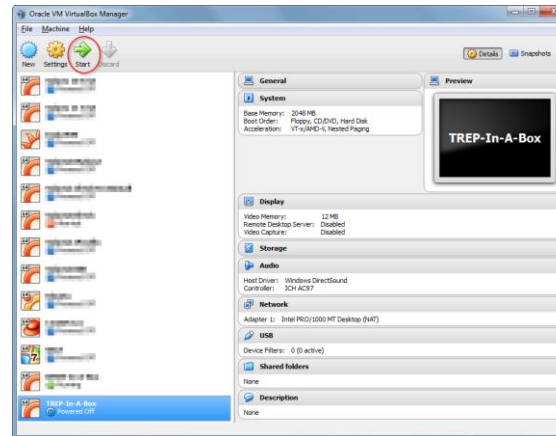


Open a Windows<sup>7</sup> command prompt and type

```
cd /D "C:\Program Files\Oracle\VirtualBox"
VBoxManage.exe modifyvm "TREP-In-A-Box" --natdnshostresolver1 on
```

This will make a DNS proxy available to the VirtualBox guest OS'es. This proxy will (by default) live on 10.0.2.3.

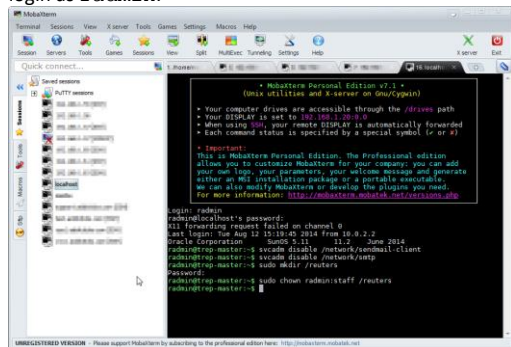
Unfortunately this is a setting that cannot be done from the VirtualBox GUI so needs to be done from the command line.



Now start the machine again. Since this is the first time the guest operating system is booted it will take some time to come up.

Give the boot 2 minutes to complete. From now on you can access the host via you favorite terminal emulator tool<sup>8</sup>.

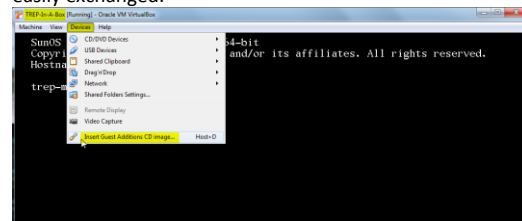
Simply SSH into localhost. This is Solaris 11 so you cannot (by default) login from outside as root. Instead login as radmin.



Notes about the radmin user and root:

There are two options for radmin user to elevate his privileges:

1. su to root account from radmin account. (you'll be prompted for password, this is the password of the root user, not the radmin user) or
2. Prefix command with 'sudo'. (you'll be prompted for password, this is the password of the radmin user, not the root user)



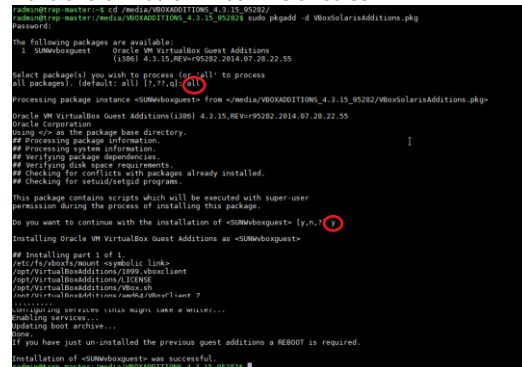
This will mount a virtual CD inside the VM. On Solaris 11 this becomes accessible under /media.

Installing these additions means installing an OS specific package. Here are the basic steps:

```
cd /media/VBOXxxxxxxx
sudo pkgadd -d VBoxSolarisAdditions.pkg
```



...and this is what it will look like on screen:



(the name of the directory under /media will vary according to the version of VirtualBox being used)

<sup>7</sup> Or equivalent on Mac OS X and Linux.

<sup>8</sup> Putty is a popular choice. In this example we use MobaXterm which in fact bundles putty as well as many other goodies.

### Post-installation tasks

The sendmail daemon on Solaris will make noise in the system log. Since we don't need sendmail we'll simply disable it permanently by issuing the following commands as `radmin` or `root`:

```
svcadm disable /network/sendmail-client  
svcadm disable /network/smtp
```

Create a directory to hold all TREP software. This directory will be shared among all zones we create.

```
sudo mkdir /reuters  
sudo chown radmin:staff /reuters
```

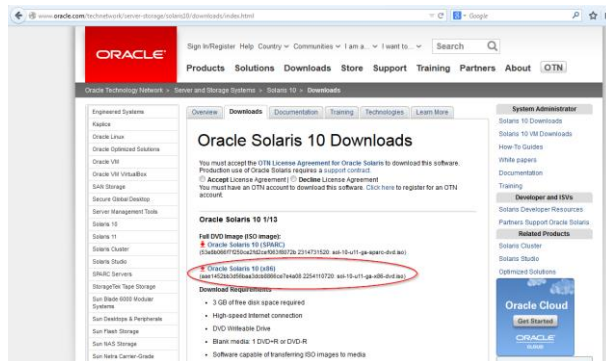
## 7. Appendix C – Solaris 10 inside Solaris 11

Before you venture into using Solaris 10 inside Solaris 11 for your TREP-In-A-Box solution you should consider if it is really necessary to use Solaris 10 as the install process is rather convoluted compared to only running Solaris 11. Remember that even if a TREP component isn't officially certified by Thomson Reuters to run under Solaris 11 chances are it will run just fine under Solaris 11 due to the [Oracle Solaris Binary Application Guarantee](#).<sup>9</sup>

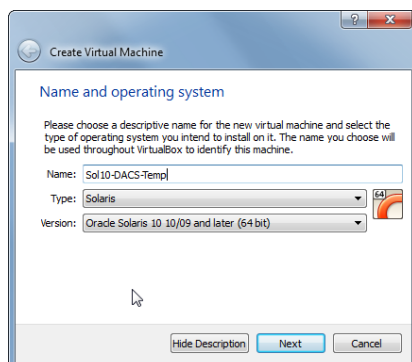
In order to install a Solaris 10 zone on a Solaris 11 host you'll need access to a Solaris 10 image. This is what will take most of the time. The `zoneadm install` command can only install a Solaris 10 zone from an image of an existing Solaris 10 host, not directly from the install media. Therefore we have to take an extra step and first create a clean Solaris 10 host. VirtualBox can be used for this purpose. We then take an image (called a Flash Archive) of this temporary host and then use that as input to the `zoneadm install` command.

### 7.1 Download Solaris 10

Download Oracle Solaris 10 Install DVD from Oracle (ISO file). ([link](#))



Use VirtualBox to create a fresh Solaris 10 (virtual) host. We will discard this host later. Create the new VM like this:



Before you start the new VM you must attach the downloaded ISO file to the DVD-drive of new VM. When you then start the VM it will boot off the DVD drive rather than booting off an empty hard disk.

<sup>9</sup> We're not advocating the use of non-certified combinations on production systems, however the target environment for this document is non-production systems.

```
SunOS Release 5.10 Version Generic_147148-26 64-bit
Copyright (c) 1983, 2013, Oracle and/or its affiliates. All rights reserved.
Configuring devices.

1. Oracle Solaris Interactive (default)
2. Custom JumpStart
3. Oracle Solaris Interactive Text (Desktop session)
4. Oracle Solaris Interactive Text (Console session)
   (Select option 3 or 4 to install a ZFS root file system)
5. Apply driver updates
6. Single user shell

Enter the number of your choice.
Automatically continuing in 12 seconds
```

When the Installer starts be sure to choose option 4. Only this option allows us to create a ZFS root file system.

```
Enabling remote services

Would you like to enable network services for use by remote clients?

Selecting "No" provides a more secure configuration in which Secure Shell is the only network service provided to remote clients. Selecting "Yes" enables a larger set of services as in previous Solaris releases. If in doubt, it is safe to select "No" as any services can be individually enabled after installation.

Note: This choice only affects initial installs. It doesn't affect upgrades.

Remote services enabled
-----
[ ] Yes
[X] No

F2_Continue F6_Help
```

Answer "No" to whether Remote Services should be enabled. If you answer yes Solaris will attempt to start lots of unnecessary network services, such as X Server, Web Server, etc.

```
Choose Filesystem Type

Select the filesystem to use for your Solaris installation

[ ] UFS
[X] ZFS

F2_Continue F3_Go Back F5_Exit F6_Help
```

Be sure to use ZFS, not UFS.

```
Select Software

Select the Solaris software to install on the system.

NOTE: After selecting a software group, you can add or remove software by customizing it. However, this requires understanding of software dependencies and how Solaris software is packaged.

[ ] Entire Distribution plus OEM support ..... 7257.00 MB
[X] Entire Distribution .. 7257.00 MB
[ ] Developer System Support ..... 7049.00 MB
[ ] End User System Support ..... 6012.00 MB
[ ] Core System Support ..... 2116.00 MB
[ ] Reduced Networking Core System Support ..... 2067.00 MB

F2_Continue F3_Go Back F4_Customize F5_Exit F6_Help
```

Be sure to install "Entire Distribution". This will install a lot of unnecessary stuff (like whole GUI environment) but none of the other choices is suitable for a server running DACS.

It doesn't matter what values you choose for things like hostname, network information, name service information, etc, during the installation of this temporary host. All such values will be replaced when we later install into the actual target location.

## 7.2 Creating the Flash Archive

Log into your new Solaris 10 host as `root`. Now use the Solaris `flarcreate` tool to create a full image of the newly installed host:

```
cd /
flarcreate -n dacstmp -S -c -L cpio sol10archive.compressed.flar
```

The created file will be approximately 2.3 GB size.

Next step is to transport this file to the target host.

Once the file is successfully transported to the target you no longer need the temporary Solaris 10 host from where you created the Flash Archive. This host can be deleted.

## 7.3 Creating the Solaris 10 zone on the target system

Now log into the target system. First login as `radmin` and then `su` to `root` account.

In the following we'll assume that the name of the new zone will be `trep-dacs`.

```
$ zonecfg -z trep-dacs
zonecfg:trep-dacs> create
create: Using system default template 'SYSdefault'
zonecfg:trep-dacs> set brand=solaris10
zonecfg:trep-dacs> set zonepath=/export/zones/trep-dacs
zonecfg:trep-dacs> set autoboot=true
zonecfg:trep-dacs> set ip-type=exclusive
zonecfg:trep-dacs> verify
zonecfg:trep-dacs> commit
zonecfg:trep-dacs> exit
```

Now install the Flash Archive into the newly configured zone:

```
zoneadm -z trep-dacs install -a /export/home/radmin/sol10archive.compressed.flar -u
```

The above command will take a couple of minutes. After this command completes successfully you can delete the flash archive file.

We also need to add the common folder (`/reuters`) to this zone so that it can be accessed from within the zone:

```
zonecfg -z trep-dacs "add fs; set dir=/reuters; set special=/reuters; set type=lofs; end; verify;
commit;"
```

Now boot the new zone:

```
zoneadm -z trep-dacs boot
```

Once the command above is completed give the zone another 1-2 minutes to fully boot.

## 7.4 Configure the new Solaris 10 zone

The new zone is now in state where it is "unconfigured". This means it doesn't have any network, keyboard, locale, name service, etc, configured.

The 'configure' application is actually now waiting for us to input this information at the console.

Enter the following command (while being `root` user):

```
zlogin -C trep-dacs
```

This will log the session into the console of zone `trep-dacs`. As the zone does not yet know anything about terminal and keyboard it means that inputting the information that is requested at the console can be a bit bewildering.

Try pressing Enter for starters. This should show the 'configure' application. You should see something like the following:

```
Select a Language
 0. English
 1. sv
Please make a choice (0 - 1), or press h or ? for help: 0
```

```
Select a Locale
 0. English (C - 7-bit ASCII)
 1. Bokmal, Norway (UTF-8)
 2. Denmark (ISO8859-1)
 3. Denmark (ISO8859-15 - Euro)
 4. Denmark (UTF-8)
 5. Finland (ISO8859-1)
 6. Finland (ISO8859-15 - Euro)
 7. Finland (UTF-8)
 8. Iceland (ISO8859-1)
 9. Iceland (UTF-8)
10. Norway (ISO8859-1 - Bokmal)
11. Norway (ISO8859-1 - Nynorsk)
12. Nynorsk, Norway (UTF-8)
13. Go Back to Previous Screen
Please make a choice (0 - 13), or press h or ? for help: 0
```

```
What type of terminal are you using?
 1) ANSI Standard CRT
 2) DEC VT52
 3) DEC VT100
 4) Heathkit 19
 5) Lear Siegler ADM31
 6) PC Console
 7) Sun Command Tool
 8) Sun Workstation
 9) Televideo 910
10) Televideo 925
11) Wyse Model 50
12) X Terminal Emulator (xterms)
13) CDE Terminal Emulator (dtterm)
14) Other
Type the number of your choice and press Return: 12
Creating new rsa public/private host key pair
Creating new dsa public/private host key pair
Configuring network interface addresses: net0
```

Some of the summary screens from a typical installation are:

```

Confirm Information for net0
> Confirm the following information. If it is correct, press F2;
to change any information, press F4.

Host name: trep-dacs
IP address: 10.0.2.110
System part of a subnet: Yes
Netmask: 255.255.255.0
Enable IPv6: No
Default Route: Specify one
Router IP Address: 10.0.2.2

Esc-2_Continue Esc-4_Change Esc-6_Help

```

```

Confirm Information
> Confirm the following information. If it is correct, press F2;
to change any information, press F4.

Name service: DNS
Domain name: trepinabox.org
Server address(es): 10.0.2.3

Esc-2_Continue Esc-4_Change Esc-6_Help

```

At the end of the configuration the screen will return to the console login prompt (you may have to press enter).

Enter characters `~.` (tilde followed by period) to end the console session. This will return you to the enclosing host (`trep-master`).

Now login to the host again. This time there's no longer any need for being on the console, so we simply do:

```
zlogin trep-dacs
```

Remember this is a very old Solaris so root's account is not exactly configured for usability. Start by doing

```
bash
```

and now disable sendmail:

```
svcadm disable /network/sendmail-client
svcadm disable /network/smtp
```

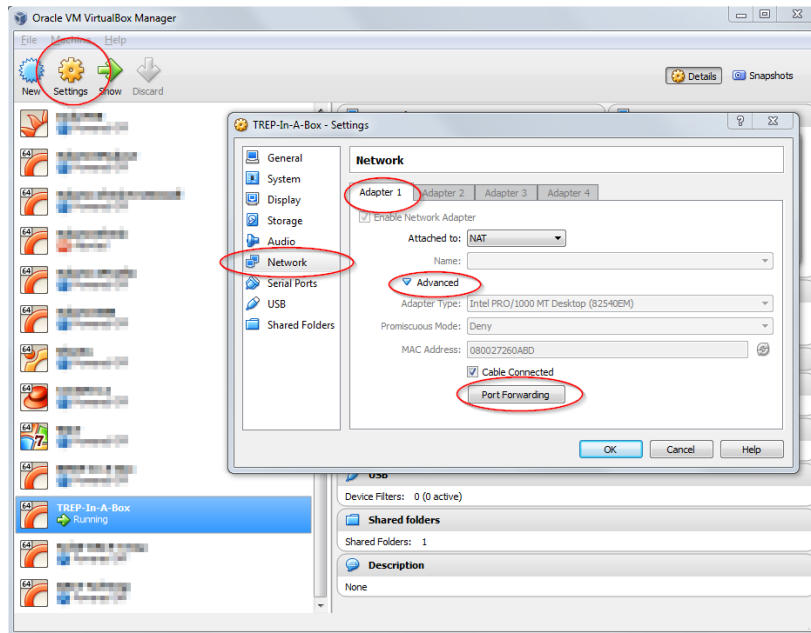
Remaining tasks to do

- Create the `radmin` user and assign him the `com.thomsonreuters.smf.manage.trep` authorization.
- Edit the file `/etc/hosts` so that the other TREP hosts are added.



## 8. Appendix C - Port forwarding in VirtualBox

Since the TREP-In-A-Box environment is secluded and behind a NAT solution you'll need to configure some Port Forwarding rules in VirtualBox to make it accessible from outside. Port Forwarding is configured in VirtualBox under the appropriate network adapter belonging to the VM machine you want to manage. Here's an example:



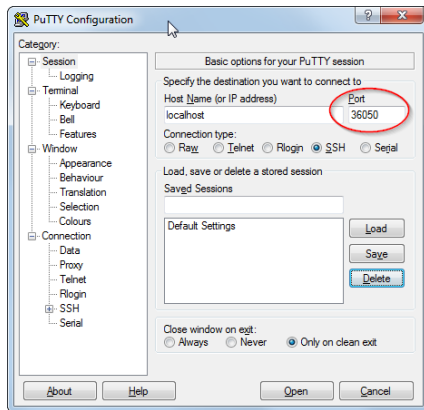
Supposedly Port Forwarding rule changes take place immediately even if the VM is running. We've found that to be the case only occasionally. If it doesn't work then try restarting the VM.

### 8.1 Reaching the virtual hosts from outside via SSH

Examples in yellow:

Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
ssh-to-master-host	TCP	127.0.0.1	36015	10.0.2.50	22
ssh-to-trep-ads1	TCP	127.0.0.1	36050	10.0.2.50	22
ssh-to-trep-ads2	TCP	127.0.0.1	36051	10.0.2.51	22
ssh-to-trep-adh	TCP	127.0.0.1	36052	10.0.2.52	22
trSSL-to-trep-ads1	TCP	127.0.0.1	36150	10.0.2.50	8101
trSSL-to-trep-ads2	TCP	127.0.0.1	36151	10.0.0.51	8101
trRSSL-to-trep-ads1	TCP	127.0.0.1	36150	10.0.2.50	14002
trRSSL-to-trep-ads2	TCP	127.0.0.1	36151	10.0.2.51	14002
DACS-GUI	TCP	127.0.0.1	39980	10.0.2.53	8080

When you connect using your favorite terminal emulator you would then have to instruct it to use e.g. port 36015 rather than the default port 22. For example if connecting to `trep-ads1` from Putty:



## 8.2 SSL and RSSL connection into ADS

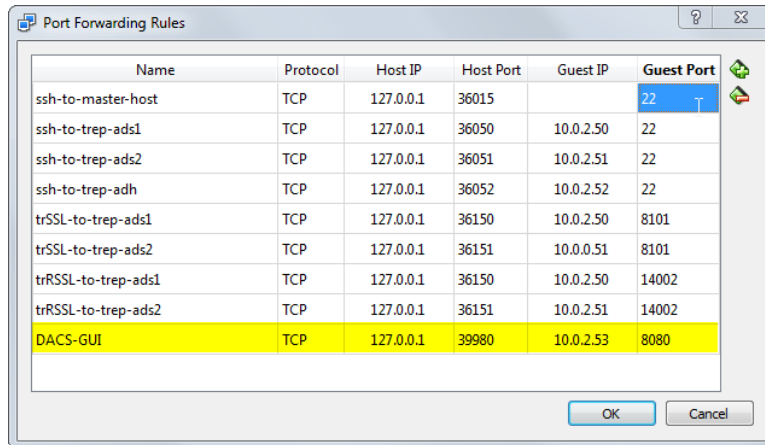
Examples in yellow:

Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
ssh-to-master-host	TCP	127.0.0.1	36015		22
ssh-to-trep-ads1	TCP	127.0.0.1	36050	10.0.2.50	22
ssh-to-trep-ads2	TCP	127.0.0.1	36051	10.0.2.51	22
ssh-to-trep-adh	TCP	127.0.0.1	36052	10.0.2.52	22
trSSL-to-trep-ads1	TCP	127.0.0.1	36150	10.0.2.50	8101
trSSL-to-trep-ads2	TCP	127.0.0.1	36151	10.0.0.51	8101
trRSSL-to-trep-ads1	TCP	127.0.0.1	36150	10.0.2.50	14002
trRSSL-to-trep-ads2	TCP	127.0.0.1	36151	10.0.2.51	14002
DACS-GUI	TCP	127.0.0.1	39980	10.0.2.53	8080

You would have to instruct your SSL/RSSL consumer applications to connect on a non-default port, e.g. for a RSSL consumer application on the workstation to reach `trep-ads1` it would connect to `localhost:36150`.

## 8.3 DACS Web UI

Example in yellow:



From your workstation you would then open the DACS Web UI on address: <http://localhost:39980/dacs/>

## 8.4 Connections from inside to outside

Inside-out connections do not require a NAT rule. An example of such a connection could be a route from an ADH to a source application living somewhere on the outside. Such a connection will work out-of-the-box. No further configuration is required.

In addition all the virtual hosts (the zones) have the same DNS information as the workstation host on where the VirtualBox instance runs. VirtualBox will proxy the DNS from the workstation into the guest OS.