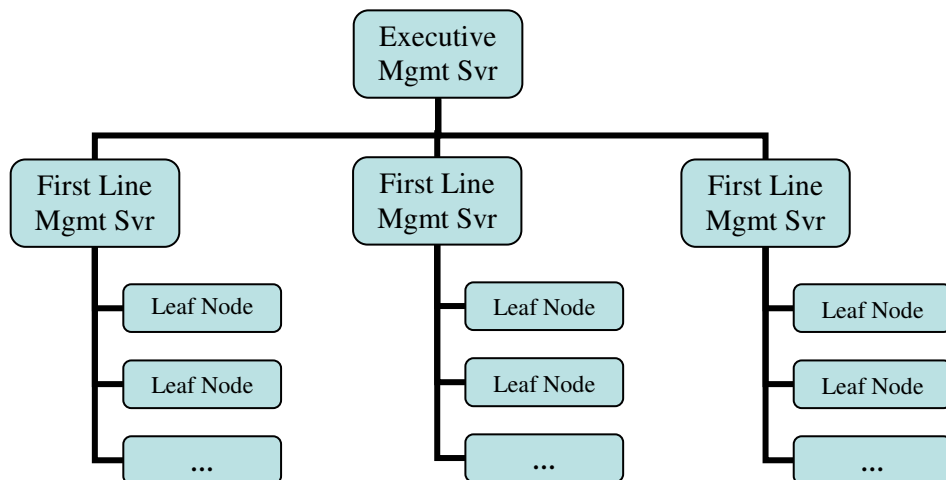


Using CSM on Linux in a Hierarchical Fashion

Author: Bruce Potter, Draft version 1.1, 10/4/04

Introduction & Terminology

There are several possible reasons why one might be motivated to use CSM in a hierarchical fashion. The obvious reason is to manage more nodes than allowed by the current CSM scaling limit. Another reason is to divide up the nodes into smaller sets that can be managed individually, sometimes by different administrators. This paper provides an overview of how to use CSM in a 3 level hierarchy, in which there are sets of nodes, with each set being managed by a CSM management server. A top level CSM management server manages the rest of the management servers.



First, a little terminology is necessary. We will call the management servers that are connected directly to the leaf nodes the first line management servers (FMS). The top management server, we will call the executive management server (EMS). Although this paper will focus on a 3 level hierarchy (EMS, FMSs, and nodes), most of what is described in this design could be generalized into more levels in the hierarchy. In this case, the management servers between the EMS and FMS would be called mid-level management servers (MMS).

One more overall comment about this paper: the approach outlined here does **not** provide a single point of administrative control for all the leaf nodes for some operations. Instead, its primary purpose is in setting up and maintaining the FMSs easily. Secondly, it also provides a way to do some of the administrative operations from the EMS to the leaf nodes.

The instructions in this paper work with CSM for Linux 1.3.1 or above. However, less steps are required if you use 1.3.2 or above. Version dependent steps are noted in the paper.

What about AIX? Some of the techniques described in this paper (e.g. those related to dsh, CFM, and monitoring) will also work with CSM on AIX. But most of the techniques related to software installation and maintenance will not work on AIX at this time.

A working knowledge of CSM is assumed in this paper. For more information about CSM, see the documentation at <http://publib.boulder.ibm.com/clresctr/windows/public/clusterbooks.html> . The dcp command that is referred to in this paper is currently part of the csm.ect RPM available from <http://www.alphaworks.ibm.com/tech/ect4linux> .

Installing the EMS

The EMS should be installed manually in the normal way, following the instructions in <http://publib.boulder.ibm.com/clresctr/docs/csm/linux/200311/am7il101/am7il10107.html#HDRINSTALLMAN> . Remember to accept the license agreement and install the license key, using:

```
csmconfig -L <license_key_file>
```

The resulting nodelock license will be propagated to the FMSs in a later step in this paper.

Installing and Configuring the FMSs

After the EMS is installed manually, the machines that are to be the FMSs should be defined as nodes and the OS and CSM client code installed on them in the normal manner. See <http://publib.boulder.ibm.com/clresctr/docs/csm/linux/200311/am7il101/am7il10108.html#HDRDEFINENODEL> and <http://publib.boulder.ibm.com/clresctr/docs/csm/linux/200311/am7il101/am7il10110.html#HDRINSTALLFULLL> for details.

After you have installed the FMSs, then you need to specify to CSM that they are each install servers for themselves and get any other necessary software on the FMSs. The procedure for doing this is described in the following sections. (Install servers are described in <http://publib.boulder.ibm.com/clresctr/docs/csm/linux/200311/am7il101/am7il10119.html#HDRMULTINST> .)

Identifying the FMSs as Install Servers

If you are using CSM 1.3.2 or above, run the following command on the EMS for each node (FMS):

```
chnode <fms-hostname> InstallServer=<fms-hostname>:/csminstall
```

This will cause the files for the distro that is on the FMSs to be copied to the FMSs (when updatenode is run).

If you are using CSM 1.3.1, setting the InstallServer directory to /csminstall is not allowed, so you must do the following:

```
dsh -a "ln -s /csminstall /csmserver"
chnode <fms-hostname> InstallServer=<fms-hostname>
```

(Note: When CSM 1.3.1 runs certain operations on a node (in this case, the FMSs), it mounts the EMS's /csminstall on top of /csminstall on the FMS. This temporarily hides the /csminstall that was created on the FMS when it was installed as a management server. This means that you can not run certain operations (e.g. updatenode) from the EMS to the FMS at the same time you are running operations from the FMS to its leaf nodes. This is fixed in CSM 1.3.2.)

FMSs That Manage Mixed Clusters

This section has not yet been verified...

In CSM 1.3.1 and 1.3.2, there is no way to tell CSM that a FMS install server needs to have the files necessary to be able to install leaf nodes with distro versions that are different from the FMS. If you need to do this for 1 or more of your FMSs, do the following with CFM:

The ISO images of the RedHat and/or SuSE CDs need to get on the FMSs from which you want to be able to install those distros on the leaf nodes. This can be accomplished using CFM (Configuration File Manager). Start by putting the ISO files in /cfmroot in some appropriate sub-directory (e.g. install-images). To copy an ISO image from the CD to a file, use the dd command, for example:

```
dd if=/dev/cd0 of=/cfmroot/install-images/rh80-1.iso
```

If you have some FMSs that will be installing different distros, you can tag the ISO files with node group extensions, so that they will only be copied to the correct machines. For example, create a node group call rh80-installers and add all the FMSs that will be installing RedHat 8.0 on some of its leaf nodes. Then name the file in the above example "rh80-1.iso._rh80-installers". (See <http://publib.boulder.ibm.com/clresctr/docs/csm/linux/200311/am7al101/am7al10104.html#HDR CFENGINE> for more information on CFM.) The directory /install-images on the FMSs should probably be created in its own file system to ensure there is enough space for the large ISO files. After everything is set up, run updatenode to get all the files pushed out to the FMSs.

It is most convenient to have all the ISO images loopback mounted on the FMSs so they are immediately available for use. This can be done by setting up pre and post scripts for each file. For example, to get rh80-1.iso mounted after it is copied, create a file called rh80-1.iso.post._rh80-installers in the same directory with the contents:

```
#!/bin/bash
# Mounts this iso file after it is copied
mkdir -p /install-images/mnt/rh80-1
mount -o loop,ro /install-images/rh80-1.iso /install-images/mnt/rh80-1
```

To unmount the iso file before updating it, create a file call rh80-1.iso.pre._rh80-installers with the contents:

```
#!/bin/bash
# Unmounts this iso file before it is re-copied
```

```
umount /install-images/mnt/rh80-1
```

Note: both the pre and post scripts need to be executable by root.

Replicating the Image Files and Installing the Management Server Code

You also need to put the nodelock license of the EMS in CFM so that it can be propagated to the FMSs:

```
ln -s /var/opt/csm/lic/nodelock /cfmroot/var/opt/csm/lic/nodelock
echo "refsrc-api -c IBM.DmsCtrl" > /cfmroot/var/opt/csm/lic/nodelock.post
```

(If you don't do this step, you will have to accept the license on every FMS.)

Now run "updatenode -a" to replicate all the CSM and distro files to the FMSs.

After all the software is set up on the FMSs, dsh can be used to run installms on the FMSs. Specifically, run:

```
dsh -a 'echo -e "\n" | /opt/csm/bin/installms -x'
```

The -x option tells installms not to copy the CSM packages and distro prerequisite packages from CDs to /csminstall. These packages are already there as a result of making it an install server and running updatenode. One other thing to keep in mind is that installms will also prompt or quit if you have any conflicting RPMs installed on your system. For instance, CSM requires tftp-hpa and pdksh. If you have any other variant of tftp or ksh installed, installms will prompt or quit. So make sure you don't have any conflicting RPMs before running installms. For more information on installms, see

<http://publib.boulder.ibm.com/clresctr/docs/csm/linux/200311/am7il101/am7il10107.html#HDRINSTALLMAN>.

Note: if you are running 1.3.1, you must at this point run: dsh -a "rm /csminstall/csm/status"

After the FMSs are set up as management servers, it is useful to get /opt/csm/bin in root's path for convenience in running CSM commands. This can be done by creating the following simple script with the file name /cfmroot/etc/profile.d/csm.sh on the EMS:

```
export PATH="/opt/csm/bin:$PATH"
export MANPATH="/opt/csm/man:$MANPATH"
```

The next time updatenode is run to the FMSs, this file will be placed in /etc/profile.d and will add CSM to the search path. If you also have csh users, create a second script on the EMS called /cfmroot/etc/profile.d/csm.csh:

```
setenv PATH /opt/csm/bin:${PATH}
setenv MANPATH /opt/csm/man:${MANPATH}
```

Note: you do not need to do these two steps if you are running CSM 1.3.2 or above, because installms will automatically add CSM to the PATH and MANPATH.

Automating the Set Up of the FMSs

The steps above to turn a node/install server into a FMS can be automated with a postreboot customization script from xCSM (<http://www.alphaworks.ibm.com/tech/ect4linux>) called **###_makeMS**. This script should be placed in /csminstall/csm/scripts/installpostreboot and will do the following things:

- remove any existing tftp server RPM
- run installms
- copy CSM license to the proper place
- refresh IBM.DmsCtrl" so it will pick up the new license
- have to make the nodelock directory

There are a few set up tasks that must be done on the EMS before running **###_makeMS**. These are documented in the header of the script.

Defining Nodes

Nodes can be defined on the FMSs in the standard ways. If there are a lot of similarities between the first line clusters, it might be most convenient to create a standard nodedef file on the EMS and then create a different version of it for each of the FMSs, and place them in /cfmroot on the EMS, tagged with the hostname of that FMS. (For more information on node definitions see <http://publib.boulder.ibm.com/clresctr/docs/csm/linux/200311/am7il101/am7il10108.html#HDRDEFINENODEL>)

For example, create a nodedef file for fms1 and put it in:

```
/cfmroot/etc/opt/csm/nodedef._fms1
```

Then you can push the file to the FMS and get the nodes defined using:

```
updatenode fms1  
dsh -n fms1 "definenode -mf /etc/opt/csm/nodedef"
```

The -m option tells definenode to modify any node definitions that already exist. This means, if you want to make a change to one or more nodes later on, you can update the copy of the nodedef file on the EMS and then rerun the 2 commands above.

Installing Nodes

Once the leaf nodes are defined on the FMSs, they can be installed using:

```
dsh -a "csmsetupks -xP"  
dsh -a "installnode -P"
```

For SuSE leaf nodes, use csmsetupyast instead of csmsetupks.

Reinstalling the FMSs

If you want to reinstall an FMS at some point, and you are not using CSM 1.3.2 or above, you must first remove the InstallServer attribute for this FMS. Otherwise, CSM (1.3.1) will try to install it from itself, which won't work very well. After installing the FMS, replace the InstallServer attribute value.

Mixing RedHat and SuSE

This section has not yet been verified...

In a hierarchical environment it is common to have a mixture of RedHat and SuSE clusters. This implies that the EMS needs to be able to install both RedHat and SuSE FMSs. This is not officially supported by CSM, but has been done with CSM 1.3.2 . Contact CSM development for more information.

Maintaining the FMSs

Updating Software

Since the FMSs are defined as install servers, new RPMs (for CSM, installnode, or SMS) can be pushed to all the FMSs via updatenode. Install and updatenode customization scripts will also get pushed to the FMSs when updatenode is run.

Configuration Files

Configuration files that you want pushed all the way to the leaf nodes can be placed in the appropriate sub-directory of /cfmroot/cfmroot on the EMS. (Note: this only works in CSM 1.3.1.1 and above.) Running “updatenode -a” on the EMS will put the files in /cfmroot on the FMSs. Then running the following command on the EMS will push the files out to the leaf nodes:

```
dsh -a “updatenode -a”
```

When you place files in /cfmroot/cfmroot to transfer files from the EMS to the leaf nodes, if you add 2 node group extensions to the file name, the right-most specified node group will be used during the transfer of the file from the EMS to the appropriate FMSs. After that, the right-most extension will be stripped off and the remaining extension will be used to transfer the file from the FMS to the appropriate leaf nodes. (Note: this only works in CSM 1.3.2 and above). By the way, there is also currently no way to create a pre or post script that will execute on the leaf nodes, without creating another post script that will rename it appropriately when it gets to the FMS.

Other Configuration Data

Other configuration data (e.g. node groups, conditions, responses, etc.) can be pushed to the FMSs using dcp and dsh and the commands’ ability to read and write its data to a file. For example, to push the node group definitions out to the FMSs, first dump the node groups on the EMS to a file using:

```
nodegrp -L >/tmp/nodegroups
```

Then copy that file to the FMSs and define the node groups using:

```
dcp -a /tmp/nodegroups /tmp/nodegroups  
dsh -a nodegrp -f /tmp/nodegroups
```

For more information about the nodegrp command, see

<http://publib.boulder.ibm.com/clresctr/docs/csm/linux/200309/am7cl100/am7cl10004.html#HDRNODEGRP> . For more information about the dsh command, see

<http://publib.boulder.ibm.com/clresctr/docs/csm/linux/200309/am7cl100/am7cl10004.html#HDRDSH> .

Other Operations

FMS data can be queried or modified from the EMS using dsh or, in some cases, distributed RMC commands. For example, you can list the OS version to be installed on each leaf node in a particular cluster using:

```
dsh -n fms1 "lsnode -a InstallDistributionName,InstallDistributionVersion"
```

As another example, all the nodes in a particular cluster can be powered on using:

```
dsh -n fms1 "rpower -a on"
```

Some commands allow you to list information on the FMSs without using dsh. For example, you can list all the conditions on the FMSs with:

```
lscondition -a
```

For more information on the lscondition command and other ERRM commands, see <http://publib.boulder.ibm.com/clresctr/docs/rsct/linux/200312/a2278933/am711mst02.html#ToC55>.

Most administrative operations (e.g. getadapters, installnode, updatenode, rpower, probemgr, etc.) can be run on the FMSs via dsh. For example, to update all the leaf nodes of a particular cluster:

```
dsh -n fms1 "updatenode -a"
```

Interactive cmds, like rconsole, will not work this way. They can only be run by telneting or ssh'ing into the specific FMS.

Monitoring

Monitoring is a key function that is useful to be able to do from a central point. The EMS can monitor aspects of the FMSs in the normal ways. The EMS can monitor aspects of the leaf nodes by defining the appropriate conditions on each FMS (via dsh) and then defining conditions on the EMS that monitor the IBM.Condition class on each FMS for the EventOccurred dynamic attribute. Responses can be defined on the EMS or FMSs that run as a result of events on the FMSs or events indirectly from the leaf nodes.

Distributing Conditions and Responses

If you want the same conditions and/or responses on the EMS and FMSs, you can copy them from the EMS to the FMSs. To do this, first dump all the resources into a file using lsrsrc:

```
lsrsrc -i IBM.Condition | egrep -v '^\.NodeNameList +=' > /tmp/conditions
```

(The NodeNameList attribute is filtered out by grep because these resources will be defined on another node.) Then copy that file to the FMSs and define the resources using:

```
dcp -a /tmp/conditions /tmp/conditions  
dsh -a mkrsrc -f /tmp/conditions IBM.Condition
```

If you want conditions/responses on the FMSs, that are not useful on the EMS, you can create them on the FMSs using dsh and mkcondition. For more information about lsrsr and mkrsr, see

<http://publib.boulder.ibm.com/clresctr/docs/rsct/linux/200312/a2278933/am7l1mst02.html#ToC16>.

Example: Monitoring the Status of All Leaf Nodes

To monitor the Status attribute of all the leaf nodes from the EMS, use the predefined condition called NodeReachability on the FMS. To cause that condition to be monitored on all the FMSs, simply create and activate a condition on the EMS that watches that condition on the FMSs:

```
mkcondition -r IBM.Condition -e "EventOccurred.Occurred==1" -m m -n '*' -s
"Name=='NodeReachability' " LeafNodeReachability
startcondresp LeafNodeReachability BroadcastEventsAnyTime
```

This creates a condition on the EMS called LeafNodeReachability, which monitors the NodeReachability condition on each FMS. The startcondresp command associates the BroadcastEventsAnyTime response (which does a wall on the EMS) with the LeafNodeReachability condition and initiates the monitoring of LeafNodeReachability. The monitoring of LeafNodeReachability in turn initiates the monitoring of NodeReachability on each FMS. Of course, you can associate any response on the EMS with LeafNodeReachability.

Example: of monitoring paging space...

This section has not yet been verified...

To monitor the paging space of all the leaf nodes from the EMS, use the predefined condition called AnyNodePagingPercentSpaceFree on the FMS. To cause that condition to be monitored on all the FMSs, simply create and activate a condition on the EMS that watches that condition on the FMSs:

```
mkcondition -r IBM.Condition -e "EventOccurred.Occurred==1" -m m -n '*'
"Name=='AnyNodePagingPercentSpaceFree'" LeafNodePagingSpace
startcondresp LeafNodePagingSpace BroadcastEventsAnyTime
```

What if you want the response to run locally on the FMS of the leaf node that has the paging space problem? For example, you may want this problem just to be logged in the AuditLog of the FMS. You can do this by creating the association on the FMS:

```
dsh -n fms1 "startcondresp AnyNodePagingPercentSpaceFree LogOnlyToAuditLogAnyTime"
```

For more information about monitoring RMC attributes, see

<http://publib.boulder.ibm.com/clresctr/docs/rsct/aix/linux/200312/a2278893/am703mst35.html#HDROVRWCOMP>.

Conclusion

Using the techniques outlined in this paper, multiple FMSs can be set up and maintained from a single EMS, and many operations to the leaf nodes can be performed from the EMS. Although

not all possible operations have been explained in this paper, the examples that are described in this paper should provide direction to the reader on how to apply these techniques to all CSM management tasks.