

Making images portable within xCAT

Originated by: vallard@sumavi.com

Updated by: IBM
09/15/2010, PM 12:35:29

Table of Contents

1. Overview.....	1
2. Exporting an image.....	1
3. Importing an image.....	2
4. Modify an image (optional).....	3
5. Deploying nodes	3
6. Appendix.....	3
manifest.xml.....	4
Exported files.....	5

1. Overview

We want to create a system of making xCAT images more portable so that they can be shared and prevent people from reinventing the wheel. While every install is unique there are some things that can be shared among different sites to make images more portable. In addition, creating a method like this allows us to create snap shots of images we may find useful to revert to in different situations.

Image exporting and importing are supported for statefull (diskfull), stateless (diskless) and statelite clusters. In the following chapters we'll show you how to use *imgexport* and *imgimport* commands to export and import images. The man pages for the commands can be found at <http://xcat.sourceforge.net/man1/imgimport.1.html> and <http://xcat.sourceforge.net/man1/imgexport.1.html>

2. Exporting an image

1. The user has a working image and the image is defined in the *osimage* table and *linuximage* table.

example:

```
lsdef -t osimage myimage
Object name: myimage
  exlist=/install/custom/netboot/sles/computel.exlist
  imagetype=linux
  netdrivers=e1000
  osarch=ppc64
  osname=Linux
  osvers=sles11
  otherpkgdir=/install/post/otherpkgs/sles11/ppc64
  otherpkglist=/install/custom/netboot/sles/computel.otherpkgs.pkglist
  pkgdir=/install/sles11/ppc64
  pkglist=/install/custom/netboot/sles/computel.pkglist
  postinstall=/install/custom/netboot/sles/computel.postinstall
  profile=computel
```

```
provmethod=netboot
rootimgdir=/install/netboot/sles11/ppc64/compute1
synclists=/install/custom/netboot/sles/compute1.list
```

2. The user runs the *imgexport* command

example:

```
imgexport myimage -p node1 -e /install/postscripts/myscript1 -e
/install/postscripts/myscript2
```

(-p and -e are optional)

A bundle file called *myimage.tgz* will be created under the current directory. The bundle file contains the ramdisk, boot kernel, the root image and all the configuration files for generating the image for a stateless and statelite cluster. For statelite, it also contains the contents of the *litefile* table for the image. For statefull, it contains the kickstart/autoyast configuration file. (see appendix). The *-p* flag puts the names of the postscripts for node1 into the image bundle. The *-e* flags put additional files into the bundle. In this case two postscripts *myscript1* and *myscript2* are included.

This image can now be used on other systems.

3. Importing an image

1. User downloads a image bundle file from somewhere. (Sumavi.com will be hosting many of these)

2. User runs the *imgimport* command

example:

```
imgimport myimage.tgz -p group1
```

(-p is optional)

This command fills out the *osimage* and *linuximage* tables, and populates file directories with appropriate files from the image bundle file such as ramdisk, boot kernel, root image, configuration files for stateless and statelite. Any additional files that come with the bundle file will also be put into the appropriate directories. For statelite, the *litefile* table will be populated with the settings for the image. However, *litetree* and the *statelite* tables are not changed. If *-p* flag is specified, the postscript names that come with the image will be put the into the *postscripts* table for the given node or group.

Very often, the user wants to make a copy of an existing image on the same xCAT mn as a start point to make modifications. In this case, you can run *imgexport* first as described on chapter 2, then run *imgimport* with *-f* flag to change the profile name of the image. That way the image will be copied into a different directory on the same xCAT mn.

example:

```
imgimport myimage.tgz -p group1 -f compute2
```

4. Modify an image (optional)

Skip this section if you want to use the image as is.

1. The user can modify the image to fit his/her own need. The following can be modified:

- Modify `.pkglist` file to add or remove packages that are from the os distro.
- Modify `.otherpkgs.pkglist` to add or remove packages from other sources. Please refer to <http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2-updatenode.pdf> for details.
- For statefull, modify the `.tmpl` file to change the kickstart/autoyast configuration.
- Modify `.synclist` file to change the files that are going to be synchronized to the nodes.
- For statelite, modify the `litefile`, `litetree` and `statelite` tables
- Modify the `postscripts` table for the nodes to be deployed.
- Modify the `osimage` and/or `linuximage` tables for the location of the source rpms and the rootimage location.

2. For stateless and statelite, run `genimage`

```
genimage image_name
```

3. Run `packimage/liteimg`

For stateless run `packimage`:

```
packimage image_name
```

For statelite run `liteimg`:

```
liteimg image_name
```

5. Deploying nodes

1. The user runs

```
nodeset osimage=<image_name>
```

and the node is able to deploy.

6. Appendix

You can only export/import one image at a time. Each tarball will have the following simple structure:

```
manifest.xml
<files>
extra/ (optional)
```

manifest.xml

The manifest.xml will be analogous to an autoyast or windows unattend.xml file where it tells xCAT how to store the items. The following is an example for a stateless cluster.

manifest.xml:

```
<?xml version="1.0"?>
<xcatimage>
  <exlist>/install/custom/netboot/sles/computel.exlist</exlist>
  <extra>
    <dest>/install/postscripts</dest>
    <src>/install/postscripts/myscript1</src>
  </extra>
  <imagename>myimage</imagename>
  <imagetype>linux</imagetype>
  <kernel>/install/netboot/sles11/ppc64/computel/kernel</kernel>
  <netdrivers>e1000</netdrivers>
  <osarch>ppc64</osarch>
  <osname>Linux</osname>
  <osvers>sles11</osvers>
  <otherpkgdir>/install/post/otherpkgs/sles11/ppc64</otherpkgdir>
  <otherpkglist>/install/custom/netboot/sles/computel.otherpkgs.pkglist</otherpkglist>
  <pkgdir>/install/sles11/ppc64/pkgdir</pkgdir>
  <pkglist>/install/custom/netboot/sles/computel.pkglist</pkglist>
  <postbootscripts>my4, otherpkgs, my3, my4</postbootscripts>
  <postinstall>/install/custom/netboot/sles/computel.postinstall</postinstall>
  <postscripts>syslog, remoteshell, my1, configrmcnode, syncfiles, my1, my2</postscripts>
  <profile>computel</profile>
  <provmethod>netboot</provmethod>
  <ramdisk>/install/netboot/sles11/ppc64/computel/initrd-diskless.gz</ramdisk>
  <rootimg>/install/netboot/sles11/ppc64/computel/rootimg.gz</rootimg>
  <rootimgdir>/install/netboot/sles11/ppc64/computel</rootimgdir>
  <synclists>/install/custom/netboot/sles/computel.list</synclists>
</xcatimage>
```

In the above example, we have a directive of where the files came from and what needs to be processed.

For statelite, the following items are added:

```
<litefile>/install/netboot/sles11/ppc64/test/litefile.csv</litefile>
<rootimgtree>/install/netboot/sles11/ppc64/test/rootimg/rootimgtree.gz</rootimgtree>
```

Where <litefile> contains the contents of the *litefile* table for the image. The contents will be populated on the *litefile* table on the new xCAT mn when the image is imported. Please note that the two other important tables for statelite, which are *litetree* and *statelite*, are not exported because they contain the server info which may not be appropriate for the new xCAT mn. The user need to fill out these two tables by hand after the image is imported.

The <rootimgtree> contains all the files (compressed) from the rootimg directory. This file will be decompressed into

/install/netboot/<os>/<arch>/<profile>/rootimg directory when the image is imported.

Note that even though source destination information is included, all files that are standard will be copied to the appropriate place that xCAT thinks they should go.

Exported files

The following files will be exported, assuming x is the profile name:

For statefull:

```
x.pkglist
x.otherpkgs.pkglist
x.tmpl
x.synclist
```

For stateless:

```
kernel
initrd.gz
rootimg.gz
x.pkglist
x.otherpkgs.pkglist
x.synclist
x.postinstall
x.exlist
```

For statelite:

```
kernel
initrd.gz
root image tree
x.pkglist
x.synclist
x.otherpkgs.pkglist
x.postinstall
x.exlist
```

Note: Although the postscripts names can be exported by using the *-p* flag. The postscripts themselves are not included in the bundle file by default. The user has to use *-e* flag to get them included one by one if needed.