xCAT Integration with IBM Supported and Licensed Tools

1 IBM Bootable Media Creator and Advanced Settings Utility - Introduction

The IBM Toolscenter for System X and Bladecenter, see http://publib.boulder.ibm.com/infocenter/toolsctr/v1r0/index.jsp, has introduced a number of utilities that are interesting for the automated setup and deployment of systems. One of the things that we'd like to do is to allow xCAT to use IBM's *Bootable Media Creator* (**BoMC**) tool set to flash the firmware on all the subsystems of a machine. The tool downloads all of the required updates from IBM's website and then creates a Linux netboot image that will run the firmware updates in a consistent and controlled fashion. We'd also like to use IBM's *Advanced Settings Utility* (**ASU**) to set the CMOS values that control the server operating environment in a consistent fashion. xCAT allows us to take these tools and the images they create and make them work in the xCAT netboot framework.

Since the Bootable Media Creator and Advanced Settings Utility are licensed products they can not be distributed as part of xCAT and must be acquired separately.

The **BoMC** has a dependency on network connectivity to www.ibm.com. If your cluster does not have good external network connectivity, you can create the environment on a machine that does, we'll refer to it as a "surrogate" and still use it on your cluster in a "static" configuration.

1.1 The Big Picture on How This Works

There are three distict operational phases:

1. Collect the tools (**BoMC** and **ASU**) and create *or refresh* a repository of firmware updates. You use the **mktoolscenter** script to do this. This phase *must* occur on a machine that has network connectivity. In the simple case this is the management node of the cluster. In the case of a "surrogate" it isn't – it's just a machine with a similar architecture, OS and xCAT installed (but not necessarily configured) to give you the scripts and libraries required. If you have a firewall you may need to specify a proxy host, port, userid and password – options are provided to allow that to occur if required.

2. Distribution of the repository and customization of the server(s) that will serve the updates from the repository. In the simple case this is optional – you already have the repository in place and you may or may not want to set up CMOS definitions for **ASU** invocation. When you use a "surrogate" you need to move the repository prepared on the "surrogate" to a target management node, untar it and then optionally customize the CMOS definitions as you would in the simple case. When you are using service nodes, if you wish to distribute the firmware update function – you will need to distribute the repository in a similar fashion, treating the "master" as a "surrogate" and the service nodes as target nodes. Most

commonly, cluster administrators with service nodes use rsync or syncfiles to effect the distribution. In the case of a service node CMOS definitions are propagated thru the shared database.

3. Execution of the firmware updates and (optionally) CMOS updates on the target nodes. Once the tools and repository have been prepared and are in place we can update nodes.

Note that a repository is “static” and that unless you go thru step 1 above you will not get updates to the tools or firmware. This works well for folks that want to have consistent results. Conversely if you are interested in picking up the latest, greatest firmware and tools you will need to go back and execute all of the steps above on a regular basis.

1.2 xCAT Conventions

xCAT allows for OS and architecture specific versions of code and data using a directory path convention – that convention can be used to differentiate between versions of the **BoMC** tool set and later CMOS values that we will set up using **ASU**. Since firmware updates are rarely architecture or operating system version specific but device drivers and other tools that can be delivered using **BoMC** can be - this is a useful framework. It is also potentially useful to have a different set of firmware that would apply to a different version of the operating system – but we don't have to follow it slavishly and we won't as you'll see.

/<installroot>/netboot/<osver>/<arch>/<profile> is the path that we will use where:

<installroot> is by default **/install** but can be set to something different in the site table (**site.installdir**).

<osver> is the target operating system. In our case this will be either **toolscenter** or **bomc** because our OS will be a toolscenter *anyos* image.

<arch> is the target architecture. In our case either **x86** or **x86_64.**

<profile> is the name of an appropriate profile in the nodetype table. In our case it does not need to be in the nodetype table it just needs to be a unique name.

To invoke the tools we will use the **nodeset** command with the argument **netboot=<osver>-<arch>-<profile>**. An example would be:

```
Nodeset node1 netboot=toolscenter-x86_64-bomc
```

The default path as we set up the environment will be **/<installroot> /netboot/toolscenter /<arch> /bomc**.

1.3 CMOS Settings

ASU has a batch command facility that we will use. You put whatever commands you wish to execute into a file and assuming that they apply to the machine that you running on **ASU** will execute them. The *firmware* table provides the mapping of node or node group to **ASU** batch file.

There are many different ways that you can associate CMOS setttings with nodes. You can choose to do it by machine type, by group or by specific node. Different machine types may have unique CMOS mappings and at times different firmware versions may introduce new functions that will have new mappings. A set of proposed CMOS settings are supplied for a number of popular machine types. If you have an **ASU** batch file prepared you can use that as well. If you don't have one and you have a machine whose configuration you like – you can create one using the **ASU** "show" function and an editor.

1.4 Download BoMC and ASU

Using your browser, navigate from the IBM home page to the System X support page and select a system you are trying to update, e.g. **x3550-m3**. Click on **download** and you should find a link to *Bootable Media Creator*. Click on the link and select a version of the toolset appropriate for the OS that you are running on the *management or service node that will be constructing the netboot images*. Options currently are RHEL 3, RHEL 4, RHEL 5, SLES 9, SLES 10, SLES11 and Windows. **BoMC** is a framework that can be used to download the updates for just about any System X server – you only need one copy to do that and you don't have to worry about the operating environment of the servers that are going to be flashed. In the examples below we'll use the version for RHEL 5 x86_64 – but your version may vary, the parameters that we use are version independent. The file we downloaded was:

```
ibm_utl_bomc_2.00_rhel5_x86-64.bin
```

We saved it to **/root/Desktop**. Similarly on the web page where you found the link to *Bootable Media Creator* you should see a link to *Advanced Settings Utility*. Click on the link and select a version of **ASU** appropriate for the OS that you are running on the *management or service node that will be constructing the netboot images*. For Linux systems do not select the RPM package – use the .tgz version. The file that we downloaded was:

```
ibm_utl_asu_asut68g_linux_x86-64.tgz
```

We saved it to **/root/Desktop** as well**.**

1.5 Are We on a Surrogate or the Management Node?

In the introduction we discussed the ability to create the environment on a surrogate and then copy or carry it (sneakernet anyone?) to the "real" system. Because the surrogate will be executing code designed for the "real" system – it will need to be a system of the same architecture and OS version. In addition in the case of a surrogate we will have to know some of the specifics of the "real" environment to be able to set it up. When running in the "real" environment we have the ability to interrogate databases to get the information required. When running in "surrogate" mode - we don't and we'll need to ask for the following:

1. The installation directory. In the site table this is site.installdir.
2. A "best guess" at the machine types in the cluster. IBM organizes firmware

updates by machine type which is a more specific definition than the product family or system name. In the vpd table this is the first four characters of vpd.mtm. You'll be given the option to add machine types so this can start with a single machine type if need be.

If you are running in “surrogate” mode you will need to invoke **mktoolcenter** with the **-s** flag.

1.6 Mktoolcenter Options

Available options include:

-l logfile – what this does is output the questions and answers to a log file that can be later used as input to a subsequent invocation of mktoolcenter. If this is not specified it will default to “mktoolcenter.log” in the current directory.

-ph <hostname> – Is used to set the proxy host name or IP address if you have a firewall and a proxy is required. You must also specify the proxy port if this option is used.

-pp <port> – Is used to set the proxy port if you have a firewall and a proxy is required. You must also specify the proxy host if this option is used.

-puser <userid> – Is used to set the userid if required by your proxy.

-ppw <password> – Is used to set the password if required by your proxy.

-s – Is used to indicate that we are running as a “surrogate” and that configuration data will be supplied by the user and not as a result of table inquires.

<input_file> - this is an input file containing the questions and answers from a previous session, i.e. a logfile. If an input file is not specified – an interactive session is assumed.

So one might invoke **mktoolcenter** thus:

mktoolcenter -l /tmp/foo2 /tmp/foo

Where /tmp/foo2 is the log file and /tmp/foo is the input file – a log from a previous session.

1.7 Run mktoolcenter to Gather Tools and Prepare the Repository

Now run (/opt/xcat/share/xcat/tools/)mktoolcenter and answer the questions. Default answers to the questions are in square brackets at the end of the question. If the default is correct – you can just hit <enter> to accept it.

After the **BoMC** image is created you are given the opportunity to set up the CMOS definitions to be used by **ASU** when the image is run. If you choose to decline to configure CMOS settings – that's fine, **ASU** will not be invoked when the Toolscenter image is run. Basically what **mktoolscenter** tries to do is to populate the firmware table for known nodes with suggested defaults by machine type if it knows the machine type. You will also be asked if you are using Serial Over LAN (SOL) and if this is an HPC cluster as these generally have unique settings. You can always go back and edit the firmware table by hand and you will need to do that if you want to specify CMOS settings by a method other than machine type. Note that this part of the setup cannot be

done on a “surrogate” machine – it must be done on the target because it is manipulating the tables. If you are running in “surrogate” mode – just decline to configure CMOS settings and then when you get to the target machine edit the firmware table by hand. Note that the CMOS settings are “bound” to a node when the nodeset command for that node is invoked so you can change the settings in the file or which file is used up until that time. The “suggested” **ASU** batch files can be found in:
/<installroot>/netboot/<osver>/<arch>/repo/cmos_settings/[sol | nosol]/[hpc | default]

Our dialog is shown.

```
Your target OS version and architecture are: rhels5.5 x86_64
It appears that you have these machine types in your inventory: 7870,7871
Would you like to add others? [no]
Where is the ibm_utl_bomc*.bin file? [/root/Desktop]
Where is the ibm_utl_asu_asut*.tgz file? [/root/Desktop]
Do you want to change the target path? [/install/netboot/toolscenter/x86_64/bomc]
Extracting...
Executing...


IBM ToolsCenter Bootable Media Creator Version 2.00.27
(C) Copyright IBM Corp. 2009. All Rights Reserved.
This Program is licensed under the terms of the agreement available by invoking this utility with the --license option.
                By extracting, copying, accessing, or using the Program, you agree to the terms of this agreement. If
                you do not agree to the terms, please uninstall the software and return it to IBM or the reseller from
                whom you acquired the software for a refund, if any.



Acquiring list of updates for Machine-Type=7870 ...  Done
(1 of 10) Acquiring brcm_fw_nic_2.1.3c_linux_32-64...
 Already downloaded.
(2 of 10) Acquiring ibm_fw_hdd_sas-1.06_linux_32-64...
 Already downloaded.
(3 of 10) Acquiring ibm_fw_dsa_3.01_dsyt60k_linux_32_64...
 Already downloaded.
(4 of 10) Acquiring qlgc_fw_fc_f5.02.01-b2.08-e2.18_linux-bc_32-64...
 Already downloaded.
(5 of 10) Acquiring ibm_utl_uxsp_p9sp06a-1.40_sles10_32-64...
 Already downloaded.
(6 of 10) Acquiring ibm_fw_uefi_p9e146a_linux_32-64...
 Already downloaded.
(7 of 10) Acquiring ibm_fw_imm_yuoo57h_linux_32-64...
 Already downloaded.
(8 of 10) Acquiring qlgc_fw_cna_qmi8142-1.01.57-bc_linux_32-64...
 Already downloaded.
(9 of 10) Acquiring ibm_fw_mptsas_hs22-2.70_linux_32-64...
 Already downloaded.
(10 of 10) Acquiring ibm_fw_sraidmr_10ie-11.0.1-0024_linux_32-64...
 Already downloaded.

Acquiring list of updates for Machine-Type=7871 ...  Done
(1 of 9) Acquiring brcm_fw_nic_2.1.3c_linux_32-64...
 Already downloaded.
(2 of 9) Acquiring ibm_fw_hdd_sas-1.06_linux_32-64...
 Already downloaded.
(3 of 9) Acquiring ibm_fw_dsa_3.01_dsyt60k_linux_32_64...
 Already downloaded.
```

```
(4 of 9) Acquiring qlgc_fw_fc_f5.02.01-b2.08-e2.18_linux-bc_32-64...
 Already downloaded.
(5 of 9) Acquiring ibm_utl_uxsp_p9sp11a-2.00_sles10_32-64...
 Already downloaded.
(6 of 9) Acquiring ibm_fw_uefi_p9e146a_linux_32-64...
 Already downloaded.
(7 of 9) Acquiring ibm_fw_imm_yuoo57h_linux_32-64...
 Already downloaded.
(8 of 9) Acquiring ibm_fw_mptsas_hs22v-2.70_linux_32-64...
 Already downloaded.
(9 of 9) Acquiring qlgc_fw_cna_qmi8142-1.01.57-bc_linux_32-64...
 Already downloaded.
Acquiring ibm_utl_uxspi_4.00_sles10_x86-64...
 Already downloaded.
Acquiring list of updates for tools ...  Done
(1 of 1) Acquiring ibm_utl_sep_1.01_sles10_x86-64...
 Already downloaded.
Acquiring ibm_utl_boot_tools-120_anyos_x86-64-mid...
 Already downloaded.
Do you want to configure CMOS Settings? [yes]
Are you using SOL? [yes]
Is this an HPC cluster? [yes]
Extracting...
Executing...


IBM ToolsCenter Bootable Media Creator Version 2.00.27
(C) Copyright IBM Corp. 2009. All Rights Reserved.
This Program is licensed under the terms of the agreement available by invoking this utility with the --license option.
                    By extracting, copying, accessing, or using the Program, you agree to the terms of this agreement. If
                    you do not agree to the terms, please uninstall the software and return it to IBM or the reseller from
                    whom you acquired the software for a refund, if any.


Warning! You haven't specified any machine type and it will be "all".
has no updates...
No applicable updates found in the working directory you specified
Are you sure you want to continue?(y or n)y Creating bootable media...
 Copying tools for bootable pxe creation...
 Extracting files to working directory...
 Copying files to pxe directory...
 Removing temporary files...
 Zipping files...
The bootable media was created successfully!
```

If this is the first time running this command, it may take a while to download all of the files, depending on how many different types of machines you have. This will set up our repository for all our firmware updates. Subsequent invocations of the command will take less time because we may already have some of the required components.

1.8 If you are using a Surrogate

You will need to move the repository that was created in the previous step to the target machine and then edit the firmware table if you wanted to use **ASU** to customize the CMOS settings.

1.9 Use nodeset to Invoke the Configuration

Now we're ready. We tell xCAT to flash the node:

```
nodeset node1 netboot=toolscenter-x86_64-bomc
rpower node1 boot
```

These netboot parameters are appropriate because we selected the defaults during the setup – yours may be different. Once xCAT completes the required steps it will move the node to the next step in the chain.

1.10 Examine the Logs

Node specific log files are saved in subdirectories of the repository. For example the log files associated with node "b9" will be found in : **/<installroot>/netboot/<osver>/<arch>/repo/b9**. The most significant ones will be:

1. bomc.log – stdout from **bomc**
2. asu.log – stdout from **asu**
3. bomc.error – stderr from **bomc –** generally a 0 length file
4. asu.error – stderr from **asu** – generally a 0 length file
5. UXSPI_tc<**ip_address**>**_**<**machine_type**>**_**<**date**>.log – this is detailed output from **bomc** on actions it took

The most important log file is "bomc.log" as it will show you summary output.

1.11 Common Problems

A NIC firmware update will cause the NFS mount to fail. If you see a log file that complains about:

```
No payload present for update <update name>
```

See /var/log/IBM_Support/UXSPI_tc-**<ip_address>_<machine_type>_<date>**.log for details.

You can ignore this and you should attempt the update a second time, i.e. nodeset and rpower as shown above.

A system may be flagged as having an error – but there is none. If you see a log file that complains about:

```
One or more updates did not succeed. Please verify all required
hardware is present and that all pre-requisites are met.
```

Then you should scan the log to see if the only "errors" are that the update is not applicable to this system configuration.

```
        Status:         ***Install did not succeed***:
                 Update not applicable to this system configuration
```

You can ignore this. Future versions of **bomc** will better differentiate between not applicable and error conditions.

1.12 Verify the Configuration

There may be some cases where either due to a hardware problem or a operational issue a firmware or CMOS update fails or is not applied immediately. Some things are expected, for example – a reboot may be required to change the hyper-threading state of a processor or in the case of Blades you may need to force the reacquisition of firmware VPD by the management module to be able to see firmware updates. Others are not expected but in any case it is useful to verify the changes that have been made.

To ensure that changes made take effect the end of the update process includes a reboot of the system. So that part is taken care of. If we have Blades we still need to telnet into the management module and do an:

```
info -reload fw
```

To allow us to see firmware updates on the Blades. If you prefer to use the management module web browser interface you can use that as well to force the firmware VPD to be reloaded.

1.13 Building an ASU Batch File

Here is a sample ASU batch file for a uEFI system, e.g. HS22, x3550-M2, etc:

```
loaddefault uEFI
loaddefault BootOrder
set uEFI.ProcessorHyperThreading Disable
set uEFI.RemoteConsoleRedirection Enable
set uEFI.Com2TextEmul VT100
set uEFI.Com2ActiveAfterBoot Enable
set uEFI.Com2FlowControl Hardware
set BootOrder.BootOrder "Legacy Only=PXE Network=Hard Disk 0"
```

The normal form of a line is <command> <object> <data>. *loaddefault* is a “macro” that sets multiple objects to predefined (default) data values. You use the *set* command to set individual objects. <data> values that are not single words must be quoted. A fragment of the output of the *show* command looks something like this:

```
IBM Advanced Settings Utility version 3.50.68G
Licensed Materials - Property of IBM
(C) Copyright IBM Corp. 2007-2009 All Rights Reserved
Discovered IMM at IP address 169.254.95.118
Connected to IMM at IP address 169.254.95.118
uEFI.TurboModeEnable=Enable
uEFI.OperatingMode=Custom Mode
uEFI.ProcessorEistEnable=Enable
uEFI.ProcessorCcxEnable=Enable
uEFI.PackageCState=ACPI C3
uEFI.ProcessorC1eEnable=Disable
uEFI.ProcessorHyperThreading=Disable
uEFI.ProcessorVmxEnable=Enable
uEFI.HardwarePrefetcher=Enable
uEFI.AdjacentCacheLinePrefetch=Enable
uEFI.DCUStreamerPrefetcher=Enable
uEFI.IPStreamerPrefetcher=Enable
uEFI.QPISpeed=Max Performance
uEFI.DdrSpeed=Max Performance
```

So if you remove the first few lines, insert a “set” on each line, change “=” to “ “ and then quote the strings that are not single words – you have a usable batch file. Not a great one – because a number of the values would have been covered by a loaddefaults – but something that can be used to clone settings.

2 Bugs

At this time these are the known bugs:

None.

3 Wish List

Here are some things that can be improved:

1. When invocation of ASU is indicated – use values from the nodehm table (if they are specified) to set up the serial console. This would eliminate a source of human error and use a single authoritative source for the data values.
2. Sucess/Failure reporting – scan logs or something like that.
3. Record keeping – keep a history of when updates are applied. An optional audit record?
4. Automatic CMOS verification.