Firmware Update Tool

What is this tool?

The firmware_update_tool is written in python script and designed to support automated firmware (bootloader/bios/bmc/fpga/cpld) of nodes in a cluster.

Where is the location of the tool?

Please request Support to provide the tool

Can this tool be used for all Platforms?

Firmware Update Tool v3.16b is **ONLY** supported for ***C4k, C5k, C6k & C8k*** series hardware.

Requirements to run this tool?

1. Only 'cohesity' user can execute the tool.

```
sudo su cohesity
```

2. File ownership of the firmware_update_tool.py should be 'cohesity'.

```
sudo chown cohesity {your working directory}/firmware_update_tool.py
```

The permissions should look like this:

```
[cohesity@rtp-clu-lts-005056ba314a-node-3 fwupdate_package_v3.16b]$ ls -al
total 60
drwxr-x--x. 3 cohesity cohesity  4096 Jan 24 00:42 .
drwxr-x---. 6 cohesity cohesity  4096 Feb 14 15:32 ..
drwxr-x--x. 7 cohesity cohesity  4096 Jan 23 21:21 firmware
-rwxr-x--x. 1 cohesity cohesity 42836 Jan 24 00:42 firmware_update_tool.py
-rw-r-----. 1 cohesity cohesity  2984 Jan 24 00:42 readme.txt
```

Can this tool work on a cluster with a non-default iris_cli password?

Yes, the firmware_update_tool.py script will work with non-default iris_cli credentials.

Commands for non-default iris_cli credentials:

```
./firmware_update_tool.py list_node -u <username> -p <password>
./firmware_update_tool.py update_node all  -u <username> -p <password>
./firmware_update_tool.py update_node xx.xx.xx.xx  -u <username> -p <password>
./firmware_update_tool.py update_node xx.xx.xx.xx,yy.yy.yy.yy -u <username> -p
<password>
```

Is this tool available now?

Yes, It will be provided by Support

Does this tool work with all the software versions?

Yes, It will work on 6.1.1 and later

In the 6.8 version, we have an underlying upgrade tool issue, so please apply the nexus patch first before firmware upgrade. If on 6.8+ the tool must be applied by Support

Execution Note:

- This is a python based plain script, so installation is not required.
- This script can be executed on a Cohesity cluster environment, but firmware update only can be performed on C4K,C5K,C6K & C8K nodes.
- You shouldn't run this script with sudo permission.

Sample of how to run the tool:

For now, this tool has two options:

1. list _node
2. update_node

```
[cohesity@jatin-ve-cluster-lts-005056811fb2-node-1 fwupdate_package]$ ./firmware_update_tool.py

Usage:
     firmware_update_tool.py [OPTION1] [OPTION2]
OPTION1 (required):
     list_node : list status of all nodes in cluster
     update_node : update firmwares of target node

OPTION2 (optional):
     -u : iris_cli username
     -p : iris_cli password
     * default credential will be used if not given

EXAMPLE:
./firmware_update_tool.py list_node
./firmware_update_tool.py list_node -u abc -p def
./firmware_update_tool.py update_node all
./firmware_update_tool.py update_node all -u abc -p def
```

1. list_node:

It will display all the node information such as ***Chassis-SN#, Slot-#, Hardware Model, Node IP Address, Location (Local Node/Remote Node)*.**

It will also display ***boot_loader, BMC, FPGA(*), BIOS(*) and BP_CPLD*** in (Installed Current Version) / (Available Latest Version) format.

Note: *Local Node* - SSH logged in Node from the cluster.

Remote Node - Other nodes from the cluster.

```
./firmware_update_tool.py list_node
./firmware_update_tool.py list_node -u <username> -p <password>
```

```
[cohesity@dev-cluster1-gsd7184986-node-2 hardware]$ ./firmware_update_tool.py list_node

------------------------------------------------------------
| Nodes information in the cluster                         |
| This tool only supports C4K nodes                        |
| (*) System will power cycle in the middle of update      |
| (installed current version) / (available latest version) |
------------------------------------------------------------

  chassis-sn   slot-#  model  version  ip_address    location     boot_loader          BMC            FPGA(*)           BIOS(*)          BP_CPLD

  GSD7184986     2     C4500   master    10.1.1.59   local node   1.06/1.06.img     1.28/1.28.img   1_10/1_14_0000.img  2_14/V02_14.img  1_08/1_08_0000.img
  GSD7184986     3     C4500             10.1.1.61   remote node  1.06/1.06.img     1.28/1.28.img   1_14/1_14_0000.img  2_14/V02_14.img  1_08/1_08_0000.img
  GSD7184986     4     C4500             10.1.1.63   remote node  1.02/1.06.img     1.28/1.28.img   1_14/1_14_0000.img  0_46/V02_14.img  1_08/1_08_0000.img
QTFCOU917001F    1     C6030            10.1.2.151   remote node  Not supported/   Not supported/   Not supported/    Not supported/   Not supported/
```

2. update_node:

update_node has the following options:

- **all :** This option will update firmware of all the C4xxx nodes in the cluster.
  ./firmware_update_tool.py update_node all
  ./firmware_update_tool.py update_node all -u <username> -p <password>

- **$(node_ip_address) :** This option will update firmware on target node (Specified Node IP Address)

  ```
  ./firmware_udpate_tool.py update_node xx.xx.xx.xx
  ./firmware_udpate_tool.py update_node xx.xx.xx.xx -u <username> -p
  <password>
  ```

  (Update a target node which ip address is xx.xx.xx.xx)

  Note: If you want to update firmware on multiple nodes then use the below command:

  ```
  ./firmware_update_tool.py update_node xx.xx.xx.xx,yy.yy.yy.yy
  ./firmware_update_tool.py update_node xx.xx.xx.xx,yy.yy.yy.yy -u
  <username> -p <password>
  ```

  (Rolling update target node which has ip address xx.xx.xx.xx and yy.yy.yy.yy)

  Run this tool in the background using "nuhup", so it can keep running regardless ssh terminal expires.

  NOTE: Run using nohup. (screen is not anymore available in the version 6.6.0d+ or 6.5.1fx due to security concern) Upgrade status can be monitored in the logfile or nohup.out.

  Make sure add "&" at the end of command line

  ```
  # run the command below from the fw pacakge directory
  ```

```
$ nohup ./firmware_update_tool.py update_node all -u '{user}' -p
'{password}' &
```

Refer: readme.txt

3. How to check completion of firmware update process?

- **Remote node :** remote node update status can be simply monitored via console output message. Same message will be also logged in the log file

```
(example)
01/08/2020 12:22:08 [INFO] : [10.1.1.239] starting update ['bmc', 'fpga']
01/08/2020 12:22:08 [INFO] : [10.1.1.239] stopping node
01/08/2020 12:25:03 [INFO] : [10.1.1.239] starting firmware update process,
estimated time to completion 30+ mins
01/08/2020 12:25:05 [INFO] : pid = 60816 on 10.1.1.239
01/08/2020 12:43:35 [INFO] : [10.1.1.239] restarting node
01/08/2020 12:51:21 [INFO] : [10.1.1.239] fw update completed
```

- **Local node :** If firmware update doesn't require power cycling a local node, console output message will be the same as remote node. However, if firmware updates require power cycling of local nodes, console output won't be available anymore. Once the local node boots up, please check the logfile to check completion of the firmware update process.

The output after the update is complete or fails will be stored as:
/home/cohesity/fwupdate_package_v3.16b/nohup.out

```
[cohesity@cohesityprd-cc543210046-node-4 fwupdate_package_v3.16]$ ls -l
total 120
drwxr-x--x. 7 cohesity cohesity  4096 Nov 22 22:56 firmware
-rwxr-x--x. 1 cohesity cohesity 42605 Nov 22 22:59 firmware_update_tool.py
-rwxrwxrwx. 1 root     root      3527 Feb  8 12:13 fwupdatetool_20230208_11:08:29.log
-rwxrwxrwx. 1 root     root      4026 Feb  8 14:29 fwupdatetool_20230208_12:16:26.log
-rwxrwxrwx. 1 root     root      4025 Feb  8 16:44 fwupdatetool_20230208_14:31:32.log
-rwxrwxrwx. 1 root     root      6180 Feb  8 16:40 ipmitool_log_10.10.28.91.log
-rwxrwxrwx. 1 root     root      7437 Feb  8 15:36 ipmitool_log_10.10.28.92.log
-rwxrwxrwx. 1 root     root      7437 Feb  8 14:24 ipmitool_log_10.10.28.93.log
-rwxrwxrwx. 1 root     root      7437 Feb  8 13:17 ipmitool_log_10.10.28.95.log
-rwxrwxrwx. 1 root     root      7437 Feb  8 12:09 ipmitool_log_10.10.28.96.log
-rw-------. 1 cohesity cohesity 13106 Feb  8 16:44 nohup.out
-rw-r-----. 1 cohesity cohesity  2982 Nov 22 22:59 readme.txt
```

```
(example)
01/08/2020 12:51:44 [INFO] : [10.1.1.237] starting update ['bios', 'bootloader']
01/08/2020 12:51:44 [INFO] : [10.1.1.237] stopping node
01/08/2020 12:54:36 [INFO] : [10.1.1.237] starting firmware update process, estimated
time to completion 30+ mins
01/08/2020 12:54:36 [INFO] : [10.1.1.237] this process will power cycle local node
multiple times. please check logfile for update status after system boots up
01/08/2020 12:54:38 [INFO] : pid = 203620 on 10.1.1.237
01/08/2020 13:07:48 [INFO] : PLEASE WAIT.. LOCAL NODE UPDATE STILL IN PROGRESS

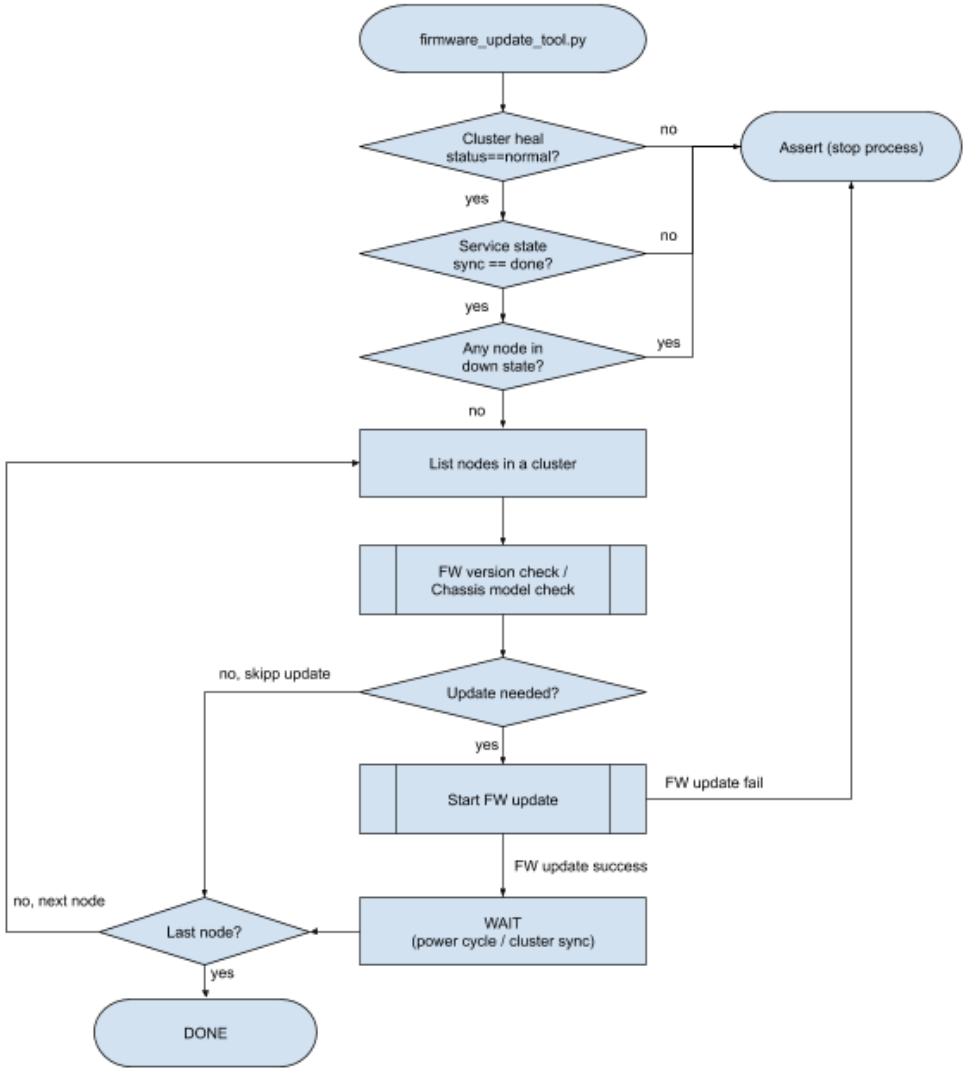
01/08/2020 13:19:39 [INFO] : PLEASE WAIT.. LOCAL NODE UPDATE STILL IN PROGRESS

01/08/2020 13:20:14 [INFO] : PLEASE WAIT.. LOCAL NODE IS RESTARTING

01/08/2020 13:28:04 [INFO] : FIRMWARE UPDATE IS COMPLETED
```

If the firmware update is running via RT access, local nodes can't be updated using the script as connection will be broken and script will be aborted after node stops services. So, if you are running firmware updates via RT, please run the current local node's update from another node at the end of your process.

Workflow:



==========

ReadMe File:

Firmware update tool V3.14:

```
# V_3.14
# Firmware update tool package:
```

```
# This tool is designed to support automated
firmware(bootloader/bios/bmc/fpga/cpld) of nodes in a cluster.
# This package includes script and the latest firmwares for C4K, C5K,
C6K, and C8K.
# All C4K/C5K/C6K/C8K node in a cluster will be updated with the
firmware in this package.

# EXECUTION :
# This is python based plain script, so installation is not required.
# This script can be executed on Cohesity cluster environment,
# but firmware update only can be proceeded on C4K, C5K, C6K, and
C8K.
# You shouldn't run this script with sudo permission.

# MAIN FUNCTION :
# list_node : list status of all nodes in cluster
# update_node : update firmwares of target node

# EXAMPLE :
# ./firmware_update_tool.py list_node  (this will display available
information of all nodes)
# ./firmware_update_tool.py update_node all (update all nodes)
# ./firmware_udpate_tool.py update_node xxx.xx.x.x (update a target
node which ip address is xxx.xx.x.x)
# ./firmware_update_tool.py update_node xxx.xx.x.x,yy.yyy.yy.yyy
(update xxx.xx.x.x and yy.yyy.yy.yyy)
# If iris_cli username and password are changed, please add -u
'[USERNAME]' -p '[PASSWORD]' option

# CONTACT :
# If you have any problems or questions, please contact nexus-hq

# Firmware version
# C4K: bios(3.14)/bmc(1.54)/BL(1.06)/fpga(1.32)/nvram(7.0)/cpld(1.08)
# C5K:
bios(1.16)/bmc(1.14)/BL(1.00)/fpga(0.30)/nvram(30.0)/cpld(0.14)
# C6K: bios(3B20.01)/bmc(4.64)
# C8K: bios(3.15)/bmc(3.24)/bpcpld(0204)/cpld(fafa)

# release history
# v 3.14  : C6K BIOS(3B20.01) and BMC(4.64)
# v 3.13  : C5K BIOS version change (v1.14 -> v1.16)
#           Handling multiple FW repository issue under /advantech
```

```
#           Fix script abort problem when product model isn't
supported
#           Fix C6K remote node upgrade hang issue
# v 3.12  : Add C5K support
#
bios(1.14)/bmc(1.14)/BL(1.00)/FPGA(0.30)/BPCPLD(0.14)/NVRAM(30)
#           C6K BMC version up to 4.63
# v 3.11  : fix 'remote_ipmi_log' assignment error
# v 3.1   : Add 6.6.0c+ and 6.5.1f+ support
# v 3.0   : Add Quanta C6K support
#           Advantech phase 11 / Inventec R3.15 / Quanta 3.19
# v 2.5.4 : fix bugs (ENG-225751)
#           Advantech phase 10 / Inventec R3.15
# v 2.5.2 : support new iris_cli security policy changes in
credential.
# v 2.5.1 : disable Quanta support temporarily.
#           minor bug fix for writing ipmitool logs.
# v 2.5   : Add Quanta support
# v 2.4   : Advantech pahse 10(V3.02) support
# v 2.3   : Inventec Denali R3.14
# v 2.2   : Advantech phase 8(V2.28) support
# v 2.1   : Add precheck existing fw update process in other nodes
# v 2.0   : Add Inventec support R3.13
# v 1.02  : Advantech new firmware release - bios(2.24), fpga(1.20),
nvram(6.0)
# v 1.01  : Error handling enhancement, adding retry
# v 1.0   : initial release
```