FlashingSLOF

Flashing Slof

This article is about flashing SLOF on JS20/JS21. Especially the steps required to get SLOF running on a system which has another firmware running. This description, however, can also be used to update from an older SLOF release (all SLOF releases <=JS2XFW704) .

A high level view on the migration process includes the following steps:

- it is important to make sure that the machine is running from the **temporary** flash−side
- (optional) backup the original flash to net using `flashsave` (available in SLOF release >= JX−1.1.0−0)
- boot the to the firmware prompt
- boot the takeover image (usually over the network)
    - `boot network1` or `boot net1` on SLOF
    - our experience has shown that serial over lan (SOL) offers higher reliability if booting over the second network interface
- flash a SLOF image from the now running SLOF

Temporary flash−side

In case there is an error during the programming of the flash it is recommend that the machine is running from the **temporary** flash−side. The easiest way to change to the temporary flash−side is to pull the blade out of the bladecenter and put it back in. It is also possible to change the flash−side by restarting the service processor via the management modul.

Boot to the firmware prompt

Original JS20/JS21 firmware

To get to the firmware prompt it is necessary to press *8*. Usually it helps pressing it quite often. It is important to note that once the firmware loaded a bootloader (like yaboot), even after leaving it (in yaboot: *bye*), it is not possible to load a second boot loader (in our case the network based). Another way to get to the firmware prompt is to disable auto boot by changing the auto−boot? environment variable.

SLOF

To get to the firmware prompt it is necessary to press *s*. As soon as following printout from the firmware can be seen the *s* can be pressed to not automatically boot to the OS.

```
JS2XBlade Starting
Build Date = Jan 25 2007 14:58:36
 FW Version = JS2XFW704
 Check ROM  = ok
 Press "s" to enter Open Firmware.
```

Backup the original flash (available in SLOF release >= JX–1.1.0–0)

The *flashsave* command will save the complete 4MB Flash image to a TFTP Server. *flashsave* uses the default *net* device, be sure the devalias for this device is set.
The TFTP Server must be setup to allow write requests.

Type *flashsave <filename>* to save the Flash image to the default TFTP Server. The output will look something like this:

```
0 > flashsave /flash_backup
  Reading MAC address from device: 00:11:25:C9:01:AD
  Requesting IP address via DHCP: 10.20.2.101
  Saving to file "/flash_backup" via TFTP
  Transmitting data: 4096 KBytes
  TFTP: Saved /flash_backup (4096 KBytes)
 ok
```

Parameters can be passed to `flashsave`:

```
0 > flashsave
Usage: flashsave [bootp|dhcp,]filename[,siaddr][,ciaddr][,giaddr][,bootp-retries][,tftp-retries][
 ok
```

To save the image e.g. to a different TFTP Server use the command `flashsave <filename>,<tftp_server_IP>`

Restoring a saved image

A saved image can be restored using the following procedure:

Use the *load <device>:<filename>* command to load the saved image. Note that the path separator for *load* is a backslash ("\")! The output will look something like this:

```
0 > load net:\flash_backup
Trying to load:  from: /ht/pci@2/ethernet@4,1:\flash_backup ...
 Bootloader 1.5
  Reading MAC address from device: 00:11:25:C9:01:AD
  Requesting IP address via DHCP: 10.20.2.101
  Requesting file "/flash_backup" via TFTP
  Receiving data:  4096 KBytes
  TFTP: Received /flash_backup (4096 KBytes)
  Successfully loaded
 ok
```

Then use the *flash4* command to write the previously loaded 4MB image to flash.

WARNING: This will overwrite the complete 4MB Flash so make sure, that the previously loaded image is a valid 4MB flash image!
If it is not, this may render your machine unusable!

```
0 > flash4
       0 Erasing... Writing... Done.
   20000 Erasing... Writing... Done.
   40000 Erasing... Writing... Done.
   60000 Erasing... Writing... Done.
   80000 Erasing... Writing... Done.
   a0000 Erasing... Writing... Done.
   c0000 Erasing... Writing... Done.
   e0000 Erasing... Writing... Done.
  100000 Erasing... Writing... Done.
  120000 Erasing... Writing... Done.
  140000 Erasing... Writing... Done.
  160000 Erasing... Writing... Done.
  180000 Erasing... Writing... Done.
  1a0000 Erasing... Writing... Done.
  1c0000 Erasing... Writing... Done.
  1e0000 Erasing... Writing... Done.
  200000 Erasing... Writing... Done.
  220000 Erasing... Writing... Done.
  240000 Erasing... Writing... Done.
  260000 Erasing... Writing... Done.
  280000 Erasing... Writing... Done.
  2a0000 Erasing... Writing... Done.
  2c0000 Erasing... Writing... Done.
  2e0000 Erasing... Writing... Done.
  300000 Erasing... Writing... Done.
  320000 Erasing... Writing... Done.
  340000 Erasing... Writing... Done.
  360000 Erasing... Writing... Done.
  380000 Erasing... Writing... Done.
  3a0000 Erasing... Writing... Done.
  3c0000 Erasing... Writing... Done.
  3e0000 Erasing... Writing... Done. ok
```

And the *reboot* to the restored image.

Boot the takeover image

After having reached the firmware prompt the takeover image can now be booted over the network. It is necessary that the network infrastructure is in place and that the required services are set up correctly. The most common scenario is that there exists a DHCP/BOOTP server which also, in addition to IP address and network details, informs our system from where to boot which file.

Original JS20/JS21 firmware

Type *boot network1* (or *net* or *network*, depending over which network interface you want to boot). The output usually looks something like this:

```
0 > boot network1
BOOTP: chosen-network-type = ethernet,auto,none,auto
BOOTP: server   IP =        0.0.0.0
BOOTP: requested filename =
BOOTP: client   IP =        0.0.0.0
BOOTP: client   HW addr =   0 14 5e 9c 12 ab
BOOTP: gateway  IP =        0.0.0.0
BOOTP: device    /pci@8000000f8000000/pci@2/ethernet@4,1
BOOTP: loc-code  U788D.001.99DYT2C-P1-T8


BOOTP R = 1 BOOTP S = 2
FILE: /tftpboot/p/s10b01
FINAL Packet Count = 1896
FINAL File Size = 970500 bytes.
load-base=0x4000
real-base=0xc00000
```

SLOF

Type *boot net1* (or *net0* or *net*, depending over which network interface you want to boot). The output usually looks something like this:

```
0 > boot net1
Trying to load:  from: /ht/pci@2/ethernet@4,1 ...

 Bootloader 1.5
  Reading MAC address from device: 00:14:5E:9C:12:AB
  Requesting IP address via DHCP: 10.32.10.101
  Requesting file "/tftpboot/p/s10b01" via TFTP
  Receiving data: |
  TFTP: Received /tftpboot/p/s10b01 (947 KBytes)
  Successfully loaded
```

After the takeover image has been loaded successfully the firmware will now execute the takeover image. The output looks like this:

```
Failed to allocated memory for RTAS

 takeover on cpu1 E284CE8 0
 takeover on master cpu \
SLOF
***************************************************************************
     Copyright International Business Machines Corporation 2004 - 2007
     US Government  Users  Restricted  Rights  --  Use, duplication or
     disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
***************************************************************************
JS2XBlade Starting
Build Date = Jan 25 2007 14:58:36
 FW Version = JS2XFW704
 Check ROM  = ok
 Press "s" to enter Open Firmware.
```

...

```
Boot list successfully read from VPD but no useful information received
Please specify the boot device in the management module

Aborting boot :  No boot device!

0 >
```

At this point the firmware prompt of the takeover image has been reached. At this point SLOF is operational and can now be used to actually flash SLOF. It is also possible to boot the OS at this point.

Flashing SLOF (with netflash (deprecated))

Once on SLOF's firmware prompt flashing is pretty similar to booting the takeover image. If the network is still configured all it takes to load the firmware image over network and write it to the flash is the following command: *netflash −f /tftpboot/p/JX−1.0.0−0−js2x.bin* The *netflash* command should give output like this:

```
0 > netflash -f /tftpboot/p/JX-1.0.0-0-js2x.bin

 Flasher 1.4
  Reading MAC address from device: 00:14:5E:9C:21:C7
  Requesting IP address via DHCP: 10.32.10.104
  Requesting file "/tftpboot/p/JX-1.0.0-0-js2x.bin" via TFTP
  Receiving data: -
  TFTP: Received /tftpboot/p/JX-1.0.0-0-js2x.bin (913 KBytes)
  Now flashing:
Checking CRC : succeeded!
Erasing Flash: 0x000E0000
Writing Flash: | ######################################## |
Recheck CRC  : succeeded!
 ok
1 >
```

The *netflash* command tries to load the file specified over the network interface which is described by the device alias *net*. To change or view the *net* alias the command *devalias* can be used.

Flashing SLOF (with update−flash)

Once on SLOF's firmware prompt flashing is pretty similar to booting the takeover image. If the network is still configured all it takes to load the firmware image over network and write it to the flash is the following command: *update−flash −f net:\tftpboot\p\JX−1.1.0−0−js2x.bin* The *update−flash* command should give output like this:

```
0 > update-flash -f net:\tftpboot\p\JX-1.1.0-0-js2x.bin
Trying to load:  from: net:\tftpboot\p\JX-1.1.0-0-js2x.bin ...
 Bootloader 1.5
  Reading MAC address from device: 00:14:5E:9C:21:C7
  Requesting IP address via DHCP: 10.32.10.104
  Requesting file "/tftpboot/p/JX-1.1.0-0-js2x.bin" via TFTP
  Receiving data:  700 KBytes
  TFTP: Received /tftpboot/p/JX-1.1.0-0-js2x.bin (700 KBytes)

Checking magic string : succeeded!
Checking platform : succeeded!
Checking CRC : succeeded!
Erasing : 0x000A0000
Writing Flash: | #########################################|
Recheck CRC : succeeded!
 ok
0 >
```

The *update−flash* command tries to mimic the options and behaviour of the *update_flash* script available under linux. An overview of the possible options is printed if the command is started without any parameters.

```
0 > update-flash
update-flash tool to flash host FW
             -f <filename>      : Flash from file (e.g. net:\boot_rom.bin)
             -l                 : Flash from load-base
             -d                 : Flash from old load base (used by drone)
             -c                 : Flash from temp to perm
             -r                 : Flash from perm to temp
```

If everything went like it should the only thing left to do is to type *reset−all*, hit enter and SLOF should boot. If there should have been an error during the programming of the flash the machine should switch to the permanent flash−side from which the broken temporary flash−side can be recovered.