

Upgrading Your VIO Server

July 2018 | by [Jaqui Lynch](#)

The care and feeding of your VIO servers is an extremely important part of systems maintenance. Keeping your VIO servers current allows you to take advantage of new features and functions, plus it also ensures you have the latest security and system patches installed.

Planning for the upgrade is important. You need to ensure that any prerequisites (HMC and firmware levels) are met prior to upgrading the VIO servers. I recommend checking to make sure that any storage is zoned to both VIO servers so that LPARs can remain up (if you so choose) during the upgrade. I'd still do the upgrade in a maintenance window just in case.

Prior to doing any update you need to download the files and ensure you read all of the readme/description files. You'll find these at IBM's Fix Central site. You should also run your current levels through the IBM FLRT tool on the web to make sure the VIO server level you plan to install is compatible with your current firmware, etc. I normally upgrade my HMC, then my firmware and then my VIO servers. In this article we will just focus on the VIO server portion of the upgrade.

Figuring Out Which Level to go to

As of right now, the most recent level of the VIO server is 2.2.6.21. Depending on the level you are at currently this may be a multiple step update. The first step is to check your current level on the VIO server. Login as padmin and type in ioslevel. It will show something like 2.2.2.2, 2.2.3.52 or 2.2.6.0 etc.

Then run oem_setup_env and check the AIX equivalent. You should see something like: 6100-09-05-1524 if ioslevel shows something like 2.2.3.52. You can check the VIOS to NIM mapping table to find out how these should match. If they do not match then most likely you have missing filesets that should be corrected ahead of time.

Before You Start

Whenever you're planning upgrades you want to check that there are no current issues with the system as it is. I recommend running errpt to make sure there are no current issues as there is no point in trying to update a system that is broken or having problems.

The next step is to take a backup. For a VIO server this is a two-step process. You need to use viosbr to backup the virtual resources and then you take a mkysyb. Another option for the mkysyb is to have a second lun in the VIO server and use alt_disk_copy to clone the VIO prior to the updates. This gives you the fastest fallback and is the method that I tend to use now although I still take a mkysyb.

In order to upgrade to 2.2.6.21 you'll need to first upgrade to 2.2.6.0 or 2.2.6.10. There's a single step option for 2.2.1.1 through 2.2.5.x that involves putting the updates for 2.2.6.21 into the same directory as the 2.2.6.10 updates—you then run updateios against that directory. If you prefer, you can keep them separate and upgrade to 2.2.6.10 and then to 2.2.6.21. It's a case of personal preference. I prefer the double upgrade but either works.

Staging the Updates

This assumes you are at VIO 2.1.0 or higher. Ensure you read the readme for 2.2.6.10 and 2.2.6.21 prior to upgrading and check that your HMC and firmware for the server are at a supported level for the updates. If you're using shared storage pools (SSPs) there are specific steps you must take prior to and during the upgrade. You should go to fix central ahead of time and download 2.2.6.10 and 2.2.6.21. Make sure you download the service pack version, not the mini pack. The service pack is required to upgrade from 2.2.6.10 to 2.2.6.21. I normally copy these to the VIO servers into a local directory. You can use sftp to upload the updates to the VIO servers. I

normally create a /usr/local/soft filesystem to load them into as there is more than just the VIO 2.2.6.10 and 2.2.6.21 code to update and the files are very large.

For this example I prestaged the updates using smitty bffcreate from the iso image as follows:

```
/usr/local/soft/vios22610  
/usr/local/soft/vios22621
```

Additional updates should be done (once finished) to Java7, SSH and SSL. You may also have to add a couple of filesets (bos.alt ones) as well as remove some. We will cover those below.

Pre-upgrade Checks

There are some checks that should be done prior to the upgrade

```
ls -al /usr/ios/utils/openssl  
lrwxrwxrwx  1 root  system  16 Apr 12 2016  /usr/ios/utils/openssl -> /usr/bin/openssl
```

Make sure it points to SSL as per the above

Also check that there is enough space in the filesystems using df -g

```
lppchk -v  
lppchk -vm3  
instfix -I | grep ML
```

The three commands above are checking to see if there are any fileset issues.

As padmin run lsvopt. If any media is loaded then you will need to unload it using unloadopt -vtd vtopt?, where ? is the virtual optical device number shown in lsvopt.

Updating VIO Servers

I try to have a spare disk in the VIO server so I can take a quick copy which will let me recover very quickly. As an example if my boot disk is a 70GB hdisk0 and I have hdisk1 as a 70GB hdisk that's not being used then I will take a copy to hdisk1 as follows:

```
alt_disk_copy -B -d hdisk1
```

The -B says don't change the bootlist but I set it below anyway to be certain

```
bosboot -a -d hdisk0  
bootlist -m normal hdisk0
```

I always set the bootlist back to the currently booted copy that I will be updating so I don't have any unexpected results when I reboot. After you do your backups then you should uninstall any efixes that have been applied. As root run emgr -l. This will list any efixes. You can then use emgr -r -L fixname to remove those fixes.

```
Now as padmin  
updateios -commit  
The above commits all previous patches on the VIO  
updateios -accept -install -dev /usr/local/soft/vios22610
```

The above installs the patches for VIO 2.2.6.10

ioslevel should now show 2.2.6.10

As root run `oslevel -s` and see if it shows a level starting 6100-09-10. If it does not then there are some fileset issues. You should also run the following to check for missing filesets:

```
instfix -I | grep ML
```

You can now remove Java5 which is not used by VIO 2.2.6.10

```
updateios -remove Java5
```

On both VIOs you may need to stop the `pmperfrec` subsystem or it will cause upgrade issues.

```
lssrc -s pmperfrec
  stopsrc -s pmperfrec
  lssrc -s pmperfrec
```

Earlier levels of VIO (e.g., 2.2.2.2) run some old director code that is no longer needed – you can clean this up by doing the following as root: Stop director and cimserver:

```
/opt/ibm/director/agent/bin/stopagent_vios
cimserver -s
```

Comment out

```
cimserver and director in /etc/inittab
: director_agent:2:once:/opt/ibm/director/agent/bin/startagent_vios >/dev/null 2>&1
: climgrcim:23456789:once:/usr/ios/sbin/climgr cimserver start > /dev/null 2>&1
```

Then:

```
/opt/ibm/director/agent/runtime/nonstop/bin/installnonstop.sh -uninstallservice
```

Then go back to padmin

```
updateios -remove DirectorCommonAgent
```

The above takes a long time (20-30 minutes)

```
updateios -remove DirectorPlatformAgent
```

I've found that the following filesets often stay backlevel if the update is from a much older version

```
bos.alt_disk_install.boot_images
adde.v2.common.ddk
adde.v2.ethernet.ddk
adde.v2.rdma.ddk
```

I usually have the updates to these in a directory of their own so I can update them if needed. In this case, the directory is `/usr/local/soft/missing`. We'll use that once we get to 2.2.6.21.

As padmin we now apply the 2.2.6.21 service pack:

```
updateios -commit  
updateios -accept -install -dev /usr/local/soft/vios22621-update
```

NOTE: There were 211 filesets to go on, however when it got to 201 of 211 it hung while updating itm.vios_premium.agent. This is a known problem and the fix is to login on another session and:

```
ps -ef | grep stopsvc
```

Then kill the process for the stopsvc for itm At the end of the upgrade it may hang again so you perform the same steps.

ioslevel should now show 2.2.6.21

We now clean up those missing files:

```
updateios -commit  
updateios -accept -install -dev /usr/local/soft/missing
```

Then make sure they get to 2.2.6.21

```
updateios -commit  
updateios -accept -install -dev /usr/local/soft/vios22621-update
```

Now remove Java6 as it is no longer needed

```
updateios -remove Java6
```

At this point you should see:

```
ioslevel  
2.2.6.21
```

As root: oslevel -s now shows:

```
6100-09-11-1810
```

Now you should update Java7 to at 625, install SSH at 7.5 and openssl at the 1300 level. You should also apply the openssl_fix26 to openssl. After the updates you should see:

```
lslpp -l | grep ssh  
       7.5.102.1100  
lslpp -l | grep ssh  
       1.0.2.1300  
lslpp -l | grep ava  
       Java7 7.0.0.625
```

To put the fix26 patch on SSL

```
cd /usr/local/software/flrtfixes/openssl_fix26  
emgr -p -e 102ma_ifix.180410.epkg.Z  
      emgr -e 102ma_ifix.180410.epkg.Z
```

If there are adapter firmware updates I'd install those now prior to the reboot.

At this point I normally run bosboot, set the bootlist and reboot the VIO server.

Post-Install Procedures

Once the updates are applied you should check a few things. I use the following:

```
oslevel -s
```

Make sure you see the expected OS level for 2.2.6.21. If you see something other than 6100-09-11-1810 then you most likely have some missing filesets that need to be corrected.

```
instfix -i | grep ML
```

This will tell you if you have missing filesets and which level they apply to

```
lppchk -v  
lppchk -vm3
```

The above will tell you about issues with filesets

```
oslevel -sq
```

Lists known service packs – you use the highest level of these in the command below

```
oslevel -s -l 6100-09-11-1810
```

The above would list any filesets that were not at least at this level. You now know the filesets that need correcting.

```
errpt
```

Check for errors

At this point you are either done or you have missing fixes to resolve.

If you do have missing or incorrect filesets you can either remove them or correct them. For the VIO server you should be using updateios, not install or smitty. If you have missing filesets for the VIO server, put them in their own directory with nothing else and then run updateios -commit and then updateios -accept pointing to that directory. Don't ever update the VIO by pointing to a full AIX LPP directory.

Once everything is updated then you should run flrtvc to identify security patches that need to be updated. Once all patches are on you may need to reboot again. Prior to rebooting I always run bosboot to reinitialize the boot sector on each boot disk and the bootlist command to ensure the bootlist is correctly set, then it's time to reboot and test. Don't forget to back everything up after the updates are completed.

Summary

System maintenance is a time-consuming process that takes planning and the VIO server is no exception. I use the flrt tool and fix central ahead of time to ensure I know what levels I am updating from and to. I also read the readme files for every step to ensure things are happening in the right order. So many problems are fixed in firmware, HMC and operating system updates that there really needs to be a regular schedule for applying these.

The key to a successful update is to have a well-planned update strategy. I pre-document every step including the commands before I do any upgrade—that way I am not searching for commands or trying to remember the next step. I also pre-stage all the files I'll need to use the day before so that I'm not delayed waiting on fix central the day of the update. That way, if fix central is down on the day I can

still move forward. With the advent of fix central, flrt, flrtvc and other updates on the IBM tool site this process has become much simpler to perform.

IBM Systems Magazine is a trademark of International Business Machines Corporation. The editorial content of IBM Systems Magazine is placed on this website by MSP TechMedia under license from International Business Machines Corporation.

©2019 MSP Communications, Inc. All rights reserved.