Bare-Metal Restore in Windows Hyper-V

The following article describes the steps needed to create a customized CentOS LiveCD, with support for Microsoft Hyper-V virtual machines, for the purpose of performing Bare-Metal Restores in CDP. The article is meant to be a proof of concept. The methods outlined can be used to create LiveCDs with other Linux distros.

Creating a LiveCD is not as daunting a task as many people think. The customized LiveCD is based on a standard CentOS 5.6 LiveCD.

```
[root@livecd ~]# cat /etc/redhat-release
CentOS release 5.6 (Final)
```

To create the customized LiveCD, follow the instructions below.

1. Obtain the standard LiveCD and boot from it.

2. After booting, you have to provide some writable disk space (at least 10 gigabytes). In this document it is assumed that the writable space is a hard disk partition and this partition is mounted in /usr/src directory. If you do not have a hard disk, you can use USB flash drive or network share.

3. After you have configured writable space, install development tools (GNU C compiler, make, etc.) and kernel sources. The kernel sources should be exactly the same version as the kernel on LiveCD (for CentOS 5.6 it is version 2.6.18-238.el5) . So do not use yum for installing kernel sources and other tools, because yum will try to upgrade your kernel to a newer version. You should download all rpms directly from one of CentOS mirrors:

```
[root@livecd ~]# rpm -i http://mirror.rackspace.com/CentOS/5.6/os/i386/CentOS/kernel-headers-2.6.18-238.el5.i386.rpm
[root@livecd ~]# rpm -i http://mirror.rackspace.com/CentOS/5.6/os/i386/CentOS/kernel-devel-2.6.18-238.el5.i686.rpm
[root@livecd ~]# rpm -i http://mirror.rackspace.com/CentOS/5.6/os/i386/CentOS/glibc-headers-2.5-58.i386.rpm
[root@livecd ~]# rpm -i http://mirror.rackspace.com/CentOS/5.6/os/i386/CentOS/glibc-devel-2.5-58.i386.rpm
[root@livecd ~]# rpm -i http://mirror.rackspace.com/CentOS/5.6/os/i386/CentOS/libgomp-4.4.4-13.el5.i386.rpm
[root@livecd ~]# rpm -i http://mirror.rackspace.com/CentOS/5.6/os/i386/CentOS/gcc-4.1.2-50.el5.i386.rpm
[root@livecd ~]# rpm -i http://mirror.rackspace.com/CentOS/5.6/os/i386/CentOS/make-3.81-3.el5.i386.rpm
```

4. Next, you have to download Linux Integration Components for Windows Server 2008 Hyper-V R2 from Microsoft's website to a Windows 2008 R2 machine. At the time of writing this the URL was as follows:

http://www.microsoft.com/downloads/en/details.aspx?FamilyID=c299d675-bb9f-41cf-b5eb-74d05

5. Launch the executable file on Windows 2008 machine. This executable is a self-extracting archive. You will have to upload extracted LinuxIC v2.iso file to Linux. Rename it so that there

are no spaces in the name - for example, to LinuxICv2.iso - and place in /usr/src directory. Mount this iso file using loopback device:

```
[root@livecd ~]# mkdir /mnt/ICv2
[root@livecd ~]# mount -o loop /usr/src/LinuxICv2.iso  /mnt/ICv2
```

6. The iso file is mounted read-only. But to be able to compile drivers for Hyper-V virtual hardware, the directory with the drivers' source code should be writable. Copy the content of the iso file to the writable space using tar instead of cp, so all file permissions are preserved:

```
[root@livecd ~]# cd /mnt/ICv2
[root@livecd ICv2]# mkdir /usr/src/ICv2work
[root@livecd ICv2]# tar -c -f - * | tar -C /usr/src/ICv2work -x -f -
```

7. Enter /usr/src/ICv2work directory and run the "setup.pl drivers" command to build drivers:

```
[root@livecd ICv2]# cd /usr/src/ICv2work/
[root@livecd ICv2work]# ./setup.pl drivers
Updating Linux integration components (vmbus, enlightened ide, enlightened scsi and network drivers) for Hyper-V...
Building vmbus driver...done.
Building blkvsc driver...done.
Building storvsc driver...done.
Building netvsc driver...
........
```

8. Errors in creating initrd image are safe to ignore. Make sure that the drivers have been built successfully. If they have, the directory /lib/modules/2.6.18-238.el5/kernel/drivers/vmbus should exist and contain five files:

```
[root@livecd ICv2work]# ls -al /lib/modules/2.6.18-238.el5/kernel/drivers/vmbus
total 1216
-rwxr-xr-x 1 root root 297963 May 24 14:34 blkvsc.ko
-rwxr-xr-x 1 root root    365 May 24 14:34 Module.symvers
-rwxr-xr-x 1 root root 320448 May 24 14:34 netvsc.ko
-rwxr-xr-x 1 root root 296318 May 24 14:34 storvsc.ko
-rwxr-xr-x 1 root root 279208 May 24 14:34 vmbus.ko
```

9. The next step is to embed the drivers into LiveCD image. It is a good idea to embed R1Soft repository configuration too, so after booting from this customized LiveCD you can always install the latest version of CDP Server or Agent using yum.

Copy the content of the CentOS LiveCD to the writable space. The content of LiveCD is in /mnt/live directory:

```
[root@livecd ~]# mkdir /usr/src/LiveCDwork
[root@livecd ~]# cd /mnt/live
[root@livecd live]# cp -R * /usr/src/LiveCDwork
[root@livecd live]# cd /usr/src/LiveCDwork
```

10. Inside the LiveCDwork you should see LiveOS directory, and inside LiveOS there is the squashfs image file - squashfs.img:

```
[root@livecd LiveCDwork]# cd LiveOS
[root@livecd LiveOS]# ls -al squashfs.img
-r-xr-xr-x 1 root root 719618048 May 24 14:13 squashfs.img
```

11. You should unpack this squashfs image using unsquashfs command (it will create squashfs-root directory):

```
[root@livecd LiveOS]#  unsquashfs squashfs.img
created 1 files
created 2 directories
created 0 symlinks
created 0 devices
created 0 fifos
[root@livecd LiveOS]# ls
livecd-iso-to-disk  osmin.img  squashfs.img  squashfs-root  TRANS.TBL
[root@livecd LiveOS]# cd squashfs-root
[root@livecd squashfs-root]# ls
LiveOS
[root@livecd squashfs-root]# cd LiveOS
[root@livecd LiveOS]# ls
ext3fs.img
```

12. Inside the squashfs-root directory there is another LiveOS directory, and inside this LiveOS directory there is the ext3fs.img file, that has to be modified. This file can be mounted read-write using the loopback device. Mount it and examine its content:

```
[root@livecd LiveOS]# mkdir /mnt/ext3
[root@livecd LiveOS]# mount -o loop ext3fs.img /mnt/ext3
[root@livecd LiveOS]# ls /mnt/ext3
bin  boot  dev  etc  home  lib  lost+found  media  misc  mnt  opt  proc  root  sbin  selinux  srv  sys  tmp  usr  var
```

13. Now you should copy /lib/modules/2.6.18-238.el5/kernel/drivers/vmbus directory to /mnt/ext3/lib/modules/2.6.18-238.el5/kernel/drivers and then run depmod to rebuild module dependencies:

```
[root@livecd LiveOS]# cp -R /lib/modules/2.6.18-238.el5/kernel/drivers/vmbus /mnt/ext3/lib/modules/2.6.18-238.el5/kernel/drivers
[root@livecd LiveOS]# depmod -b /mnt/ext3
```

14. Next, create the R1Soft repository configuration for yum. Open the file /mnt/ext3/etc/yum.repos.d/r1soft.repo in the text editor and enter the following text:

```
[r1soft]
name=R1Soft Repository Server
baseurl=http://repo.r1soft.com/yum/stable/$basearch/
enabled=1
gpgcheck=0
```

15. Save the file, exit from the text editor, exit /mnt/ext3 directory, and unmount the ext3 image:

```
[root@livecd LiveOS]# cd
[root@livecd ~]# umount /mnt/ext3
```

16. Delete the squashfs.img file and recreate it:

```
[root@livecd ~]# cd /usr/src/LiveCDwork/LiveOS
[root@livecd LiveOS]# rm -f squashfs.img
[root@livecd LiveOS]# mksquashfs squashfs-root squashfs.img
Creating little endian 3.0 filesystem on squashfs.img, block size 65536.
Little endian filesystem, data block size 65536, compressed data, compressed metadata, compressed fragments
Filesystem size 703333.81 Kbytes (686.85 Mbytes)
        16.77% of uncompressed filesystem size (4194560.32 Kbytes)
Inode table size 79376 bytes (77.52 Kbytes)
        30.26% of uncompressed inode table size (262306 bytes)
Directory table size 46 bytes (0.04 Kbytes)
        100.00% of uncompressed directory table size (46 bytes)
Number of duplicate files found 0
Number of inodes 3
Number of files 1
Number of fragments 0
Number of symbolic links  0
Number of device nodes 0
Number of fifo nodes 0
Number of socket nodes 0
Number of directories 2
Number of uids 1
        root (0)
Number of gids 0
```

17. Change directory to /usr/src/LiveCDwork, remove squashfs-root directory, and create the ISO file using mkisofs command:

```
[root@livecd ~]# cd /usr/src/LiveCDwork
[root@livecd LiveCDwork]# rm -rf LiveOS/squashfs-root
[root@livecd LiveCDwork]# mkisofs -o /usr/src/CentOS-5.6-i386-LiveCD-plus.iso -b isolinux/isolinux.bin
-c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -r -R -J -V "CentOS-5.6-i386-LiveCD"
-A "CentOS-5.6-i386-LiveCD" -T.
INFO:   UTF-8 character encoding detected by locale settings.
        Assuming UTF-8 encoded filenames on source filesystem,
        use -input-charset to override.
Size of boot image is 4 sectors -> No emulation
  1.41% done, estimate finish Tue May 24 14:54:20 2011
  2.82% done, estimate finish Tue May 24 14:53:45 2011
**********skipped**********
 97.16% done, estimate finish Tue May 24 14:53:27 2011
 98.57% done, estimate finish Tue May 24 14:53:27 2011
Total translation table size: 4937
Total rockridge attributes bytes: 1749
Total directory bytes: 4096
Path table size(bytes): 40
Max brk space used 1a000
355092 extents written (693 MB)
```

18. Transfer the ISO file you have just created (/usr/src/CentOS-5.6-i386-LiveCD-plus.iso) to your Hyper-V server and boot the VM you want to restore from this ISO. After logging in as root, load the netvsc module and check if seth0 intrface has appeared:

```
[root@livecd ~]#  modprobe netvsc
[root@livecd ~]#  ifconfig seth0
```

19. If you see the interface, you can assign IP address to it manually, or run dhclient to obtain IP address from DHCP server. When the network is configured, you can install CDP Agent or CDP Server Standard or Advanced Edition using yum. If you are using Agent, do not forget to turn off the firewall which blocks the Agent port and to restart Agent in recovery mode:

```
[root@livecd ~]# /etc/init.d/iptables stop
[root@livecd ~]# touch /usr/sbin/r1soft/.recovery-mode
[root@livecd ~]# /etc/init.d/cdp-agent restart
```

20. Now you can go to the Web Interface of CDP Server and start the Bare-Metal Restore process. See Launching Bare-Metal Restore.