
Connect:Direct for UNIX in the Veritas Cluster Server Environment

Connect:Direct for UNIX was installed and configured in a Veritas Cluster Server (VCS) 5.0 test environment and successfully passed several failover scenarios. Secure+, with strong authentication, signatures and data encryption was also configured and tested. Control and failover of the Connect:Direct File Agent was verified.

This document is intended as a supplement to the Connect:Direct for UNIX and VCS product documentation and addresses the changes required during the typical Connect:Direct for UNIX installation and setup to accommodate the specific requirements of a VCS cluster environment. This document is not a tutorial on the installation and setup of Connect:Direct or VCS.

The following information is included in this document:

- ◆ Test environment details
- ◆ Connect:Direct for UNIX installation and configuration requirements
- ◆ Sample scripts
- ◆ Failover test scenarios

Please refer to <http://www.symantec.com/business/cluster-server> for descriptive information of the VCS environment.

Test Environment

The following hardware was used in the test environment:

- ◆ Two HP ProLiant DL380 G5 computers running Red Hat Enterprise Linux AS release 4 (Nahant Update 7) with 15.86G memory and four Intel(R) Xeon(R) CPU 5160 @ 3.00GHz CPUs
- ◆ A 1000baseT network connection via Broadcom Corporation NetXtreme II BCM5708 Gigabit Ethernet adapters.
- ◆ Heartbeat connection via second set of Broadcom Corporation NetXtreme II BCM5708 Gigabit Ethernet adapters.
- ◆ Shared drive via 2Gbps fibre channel SAN. SAN unit is a Nexsan ATABoy2x with 14x400G drives configured as a single RAID 5 column with a 500GB shared LUN exposed.

Connect:Direct for UNIX Installation and Configuration Requirements

Use the information and instructions in this section when installing and configuring Connect:Direct for UNIX to operate in a VCS High Availability environment. These procedures assume that you have set up a service group for Connect:Direct for UNIX that includes a virtual IP address (VIP) resource and a Connect:Direct for UNIX resource type defined as follows:

```

type ConnectDirect (
    static str ArgList[] = { CDInstallDir, CDNodeName, CDUser, VirtualIP }
    str CDInstallDir
    str CDNodeName
    str CDUser
    str VirtualIP
)

```

It is also assumed that there is a shared storage area available or mountable to all nodes in the cluster where Connect:Direct can be installed and for data files.

Installation

Perform the following steps before beginning the installation outlined in the *Connect:Direct for UNIX Getting Started Guide*:

1. Create a Connect:Direct user with the same name and uid on each cluster node.
2. Create a Connect:Direct subdirectory on the shared data filesystem.
3. Ensure that the Connect:Direct subdirectory is owned by the Connect:Direct user.

Configuration

Perform the following steps after installing Connect:Direct for UNIX in the subdirectory on the shared data filesystem:

1. Modify the `node.listen` record of the initialization parameters file (`initparm.cfg`) as follows:

```
:comm.info=0.0.0.0;1364:\
```

2. Modify the `api.parms` record of the NDMAPI configuration file (`ndmapi.cfg`) as follows:

```
:tcp.hostname=logical_host_ip_name:\
```

3. Modify the `local.node` record of the netmap file (`netmap.cfg`) as follows:

```
:tcp.api=logical_host_ip_name;1363:\
```

4. Set the outgoing address parameter. This parameter enables you to specify an IP address for the adjacent node to use for netmap checking. It prevents sessions from failing when you initiate from within a High Availability environment to a Connect:Direct node that has netmap checking turned on. See the *Connect:Direct for UNIX Administration Guide* for details.

The outgoing address parameter is defined in the local.node record of the netmap file as follows:

```
:outgoing.address=(logical_host_ip_name|VIP):\
```

5. Modify the loopback adjacent.node record of the netmap file (netmap.cfg) as follows:

```
:comm.info=logical_host_ip_name;1364:\
```

6. Modify the tcp.max.time.to.wait parameter of the netmap file (netmap.cfg) to a value other than zero, such as 20 seconds.
7. Modify the short term session retry interval (conn.retry.stwait) and long term session retry interval (conn.retry.ltwait) parameters to a value greater than the tcp.max.time.to.wait value set in step 6.
8. For asset protection, acquire a key that includes the CPU-ID of all possible recovery machines.
9. Create a user data filesystem that is shared by all cluster nodes. This area is used for all file transfers.
10. Place the following sample scripts in the \$VCS_HOME/bin/ConnectDirect directory. Update the scripts as required for your current environment. Copy the sample scripts to all nodes in the cluster.

Sample Scripts

The following scripts are the working examples from the Sterling Commerce test environment; they are provided as an example only. It is the user's responsibility to write, test, and verify scripts appropriate to their environment.

Online Script

```
#!/bin/ksh
# Name: online
#
# Purpose: Start Connect:Direct in Veritas HA cluster environment
# Arguments: None
# Returns: 0
#
# 1. Start Connect:Direct cdpmgr (server process)
# 2. Start Connect:Direct CLI
# 3. Release any NDM processes in HOLD state
# 4. Start Connect:Direct File Agent
# 5. exit 2 (to tell HA manager to wait two seconds before verifying online
#    status)

#
# Setup Connect:Direct variables
#
export CDInstallDir=$2
export CDNodename=$3
export CDUser=$4

# Start up Connect:Direct
echo "Starting up Connect Direct."
su $CDUser -c "${CDInstallDir}/ndm/bin/cdpmgr -i
${CDInstallDir}/ndm/cfg/${CDNodename}/initparm.cfg"

# Release any NDM processes in the HOLD state. This will release
# ALL HOLD processes, not just Hold in Error.
echo "Releasing any Connect Direct processes in the HOLD state."
NDMAPICFG=${CDInstallDir}/ndm/cfg/cliapi/ndmapi.cfg ;export NDMAPICFG
su $CDUser -c "${CDInstallDir}/ndm/bin/direct" << EOF > /dev/null 2>&1
change process pnum=* release;
q;
EOF

# Start Connect:Direct File Agent
echo "Starting Connect Direct File Agent in the background."
su - ${CDUser} -c "${CDInstallDir}/file_agent/cdfa &" > /dev/null 2>&1

exit 2
```

Offline Script

```

#!/bin/ksh
# Name: offline
#
# Purpose: Normal Stop Connect:Direct in a Veritas HA cluster environment
# Arguments: None
# Returns: 0

# Actions:
# 1. Issues Stop to Connect:Direct File Agent
# 2. Waits up to 20 seconds to allow Connect:Direct FA to stop
# 3. Issues Stop Connect:Direct command with force option
# 4. Waits up to 20 seconds to allow Connect:Direct to stop
# 5. Kills any remaining Connect:Direct processes. This step is to ensure HA
#    resources are freed.
#
#    CAUTION: we recommend using a Connect:Direct node name that is unique,
#             else the kill commands below may inadvertently kill non-
#             Connect:Direct processes.
#
# 6. exit 2 (to tell HA manager to wait 2 seconds before verifying offline
#    status)

export CDInstallDir=$2
export CDNodeName=$3
export CDUser=$4

# Stop Connect:Direct File Agent
ps -ef | grep ${CDUser} | grep cdfa | grep -v grep > /dev/null 2>&1
if [ $? -ne 0 ]
then
    echo "cdfa is not running, bypassing stop command."
else
    echo "Issuing stop command to CDFA."
    su - ${CDUser} -c "touch ${CDInstallDir}/file_agent/shut"
    echo "Waiting for CDFA to stop."
    LOOP=0
    ps -ef | grep ${CDUser} | grep cdfa | grep -v grep > /dev/null 2>&1
    while [ $LOOP -lt 10 -a $? -eq 0 ]
    do
        echo "cdfa still running, waiting 2 seconds to recheck."
        sleep 2
        ((LOOP+=1))
        ps -ef | grep ${CDUser} | grep cdfa | grep -v grep > /dev/null 2>&1
    done
    if [ $LOOP -lt 10 ]
    then
        echo "cdfa stop successfull."
    else
        echo "cdfa stop unsuccessful."
    fi
fi

# Stop Connect direct
ps -ef | grep cdpmgr | grep ${CDNodeName} | grep -v grep > /dev/null 2>&1
if [ $? -ne 0 ]

```

Offline Script (continued)

```

then
  echo "cdpmgr is not running, bypassing stop command."
else
  echo "Issuing stop force to cdpmgr in background."
  su - ${CDUser} -c "setenv NDMAPICFG ${CDInstallDir}/ndm/cfg/cliapi/ndmapi.cfg; cd
${CDInstallDir}/ndm/bin; ./direct " <<- EOF > /dev/null 2>&1 &
  stop force;
  EOF

  # Wait until the stop force to complete
  echo "Waiting for the stop force to complete."
  LOOP=0
  ps -ef | grep cdpmgr | grep ${CDNodeName} | grep -v grep > /dev/null 2>&1
  while [ $LOOP -lt 10 -a $? -eq 0 ]
  do
    echo "cdpmgr still running, waiting 2 seconds to recheck."
    sleep 2
    ((LOOP+=1))
    ps -ef | grep cdpmgr | grep ${CDNodeName} | grep -v grep > /dev/null 2>&1
  done
  if [ $LOOP -lt 10 ]
  then
    echo "cdpmgr stop successfull."
  else
    echo "cdpmgr stop unsuccessful."
  fi
fi

#
# Finally, remove all Connect Direct processes owned by the Connect Direct user
# id
#
echo "Cleaning up any remaining Connect Direct processes."
for I in `ps -ef | egrep "${CDNodeName}|cdfa" | grep -v grep | grep -v offline | awk
' { print $2 } ' `
do
  echo "killing pid $I"
  kill -9 $I
done

exit 2

```

Monitor Script

```

#!/bin/ksh
# Name: monitor
#
# Purpose: Monitors Connect:Direct in a Veritas HA cluster environment
# Arguments: None
# Returns: 110 if Connect:Direct is running, 100 if absent

# Actions:
# 1. Queries system presence of cdpmgr daemon, returns 100 if daemon is not present.
# 2. Runs simple pnode=snode process, returns 100 if process fails.
# 3. Queries system presence of cdfa process, returns 100 if process is not present,
#    110 if process is present.
#

export CDInstallDir=$2
export CDNNodeName=$3
export CDUser=$4
export VirtualIP=$5

echo "Checking for presence of cdpmgr daemon."
ps -ef | grep cdpmgr | grep ${CDNNodeName} > /dev/null 2>&1
MonitorRC=$?
if [ MonitorRC -ne 0 ]
then
    echo "cdpmgr daemon is not present."
    exit 100
fi
echo "cdpmgr daemon is present."

echo "Checking virtual IP accessibility."
ping -c 1 ${VirtualIP} > /dev/null 2>&1
if [ $? -eq 0 ]
then
    echo "Submitting pnode=snode process."
    NDMAPICFG=${CDInstallDir}/ndm/cfg/cliapi/ndmapi.cfg ;export NDMAPICFG
    su $CDUser -c "${CDInstallDir}/ndm/bin/direct" <<- EOF > /dev/null 2>&1
    submit maxdelay=20
    monitor process snode=${CDNNodeName}
    step1 exit
    pend;
    EOF
    MonitorRC=$?
    if [ MonitorRC -ne 0 ]
    then
        echo "pnode=snode process failed."
        exit 100
    fi
    echo "pnode=snode process succeeded."
else
    echo "Virtual IP not accessible, monitor failed."
    exit 100
fi

```

Monitor Script (continued)

```
echo "Checking for presence of cdfa process."
ps -ef | grep cdfa | grep ${CDUser} > /dev/null 2>&1
MonitorRC=$?
if [ MonitorRC -ne 0 ]
then
    echo "cdfa process is not present."
    exit 100
else
    echo "cdfa process is present."
    echo "ConnectDirect is running."
    exit 110
fi
```


Clean Script

```

#!/bin/ksh
# Name: clean
#
# Purpose: Force Stop Connect:Direct in a Veritas HA cluster environment
# Arguments: None
# Returns: 0

# Actions:
# 1. Issues Stop to Connect:Direct File Agent
# 2. Waits up to 10 seconds to allow Connect:Direct FA to stop
# 3. Issues Stop Connect:Direct command with force option
# 4. Waits up to 10 seconds to allow Connect:Direct to stop
# 5. Kills any remaining Connect:Direct processes. This step is to ensure HA
# resources are freed.
#
# CAUTION: we recommend using a Connect:Direct node name that is unique,
#           else the kill commands below may inadvertently kill non
#           Connect:Direct processes.
#
# 6. If any Connect:Direct processes remain, exit 1, else exit 0.

export CDInstallDir=$3
export CDNNodeName=$4
export CDUser=$5
export VirtualIP=$6

# Stop Connect:Direct File Agent
ps -ef | grep ${CDUser} | grep cdfa | grep -v grep > /dev/null 2>&1
if [ $? -ne 0 ]
then
    echo "cdfa is not running, bypassing stop command."
else
    echo "Issuing stop command to CDFA."
    touch ${CDInstallDir}/file_agent/shut
    #su - ${CDUser} -c "touch ${CDInstallDir}/file_agent/shut"
    echo "Waiting for CDFA to stop."
    LOOP=0
    ps -ef | grep ${CDUser} | grep cdfa | grep -v grep > /dev/null 2>&1
    while [ $LOOP -lt 5 -a $? -eq 0 ]
    do
        echo "cdfa still running, waiting 2 seconds to recheck."
        sleep 2
        ((LOOP+=1))
        ps -ef | grep ${CDUser} | grep cdfa | grep -v grep > /dev/null 2>&1
    done
    if [ $LOOP -lt 5 ]
    then
        echo "cdfa stop successfull."
    else
        echo "cdfa stop unsuccessful."
    fi
fi
# Stop Connect direct
ps -ef | grep cdpmgr | grep ${CDNodeName} | grep -v grep > /dev/null 2>&1
if [ $? -ne 0 ]

```

Clean Script (continued)

```

then
  echo "cdpmgr is not running, bypassing stop command."
else
  echo "Checking virtual IP accessibility."
  ping -c 1 ${VirtualIP} > /dev/null 2>&1
  if [ $? -eq 0 ]
  then
    echo "Issuing stop force to cdpmgr in background."
    su - ${CDUser} -c "setenv NDMAPICFG ${CDInstallDir}/ndm/cfg/cliapi/ndmapi.cfg;
cd ${CDInstallDir}/ndm/bin; ./direct " <<- EOF > /dev/null 2>&1 &
    stop force;
    EOF

    # Wait until the stop force to complete
    echo "Waiting for the stop force to complete."
    LOOP=0
    ps -ef | grep cdpmgr | grep ${CDNodeName} | grep -v grep > /dev/null 2>&1
    while [ $LOOP -lt 5 -a $? -eq 0 ]
    do
      echo "cdpmgr still running, waiting 2 seconds to recheck."
      sleep 2
      ((LOOP+=1))
      ps -ef | grep cdpmgr | grep ${CDNodeName} | grep -v grep > /dev/null 2>&1
    done
    if [ $LOOP -lt 5 ]
    then
      echo "cdpmgr stop successfull."
    else
      echo "cdpmgr stop unsuccessful."
    fi
  else
    echo "Virtual IP not accessible, bypassing normal stop attempt."
  fi
fi

#
# Finally, remove all Connect Direct processes owned by the Connect Direct user id
#
echo "Cleaning up any remaining Connect Direct processes."
for I in `ps -ef | egrep "${CDNodeName}|cdfa" | grep -v grep | grep -v clean | awk '
{ print $2 } ' `
do
  echo "killing pid $I"
  kill -9 $I
done

ps -ef | egrep "${CDNodeName}|cdfa" | grep -v grep | grep -v clean > /dev/null 2>&1
ALL_DOWN=$?
if [ $ALL_DOWN -ne 0 ]
then
  echo "All Connect Direct processes are down."
  exit 0
else
  echo "Not all Connect Direct processes are down."
  exit 1
fi

```

Failover Test Scenarios

The following failover test scenarios were initiated and failover was then triggered via the following mechanisms:

- ◆ Kill the cdpmgr process
- ◆ Kill the cdfa process
- ◆ Pull the network cable on the node running the Connect:Direct service

All of the following test scenarios passed.

- ◆ Connect:Direct and C:D File Agent running but idle.
- ◆ Connect:Direct as pnode sending a file.
- ◆ Connect:Direct as pnode receiving a file.
- ◆ Connect:Direct as snode receiving a file.
- ◆ Connect:Direct as snode sending a file.
- ◆ Connect:Direct as pnode running a task.
- ◆ Connect:Direct as snode running a task.
- ◆ Connect:Direct running stress test involving multiple pnode and snode sessions running processes with a variety of steps.

Connect:Direct Known Issue Involving Held Processes

The following is a known issue with Connect:Direct for UNIX version 4.0.00 in the Veritas Cluster Server Environment.

In order to restart all UNIX-initiated Connect:Direct Processes that were Held in Error, **all** held Processes are restarted when HACMP starts Connect:Direct for UNIX. This includes any Connect:Direct Processes intentionally Held by the Operator. In the event of a failure, **all** Connect:Direct Processes being held, for whatever reason, will be restarted when the ndm package restarts. This could be a security problem, so please note this functionality.

