IBM Rational ClearCase 8.0
VOB Schema 80 and Feature Level 7

Migration Strategies

Christopher Guild
Brian Cowan
Benjamin Raymond

November 17, 2011

Introduction

ClearCase version 8.0 introduced both a VOB Schema and Feature Level update to allow for new functionality within the release. To be able to benefit from these changes, existing ClearCase 7.x and prior VOBs will require both a database reformat and then an update to the new Feature Level. While a ClearCase version 8.0 VOB server can support both Schema 54 and Schema 80 VOBs simultaneously, careful planning must be done to minimize downtime and ensure a successful migration from Schema 54 to 80 as well as the Feature Level update.

New features in ClearCase and ClearCase MultiSite version 8.0

New VOB schema and feature level:

VOBs at Schema 54 can now coexist on the same machine with VOBs at Schema 80. For information about version 8.0 features that are enabled when the feature level is raised to 7, refer to **About Feature Levels and ClearCase**.

Feature Level 7:

Raising the feature level to 7 once the VOB Schema has been reformatted to 80 (or in the case of a replicated VOB, raising the replicated VOB family to feature level 7) introduces the following features:

- Evil twin detection and prevention
- Support for additional integrations with Rational Team Concert
- Predefined element types for Unicode type managers
- Improved UCM Performance
- More precise event time storage during checkin and checkout

Compatibility across releases

The following tables located in the Compatibility across releases section of the IBM Rational ClearCase Information Center are extremely important to note with regards to ClearCase Client and VOB/View Server compatibility:

Table 3. Client version compatibility for 7.1.2 and 8.0
Table 4. Client version compatibility for ClearTeam Explorer (CTE)

As indicated, backwards compatibility can be maintained when a VOB database has been raised from Schema 54 to 80. However, once the Feature Level of the VOB has been updated to 7, only ClearCase version 8.0 Clients can access the newly introduced features in the updated VOBs.

Therefore, careful planning should be considered with regards to:

- Length of time to reformat a VOB database and associated downtime
- Status and timing of the ClearCase Client migration to version 8.0
- Consideration for Client interoperability between version 8.0 and 7.1.2 and correctly timing the Feature Level update
- Consideration for interoperability between version 8.0 and 7.1.2 MultiSite environments

Preparation

Hardware Requirements

In the goal to reformatting VOB databases within the shortest time possible, the VOB servers should be at least within the minimum requirements specified within the **System Requirements for ClearCase 8.0**. Additional server hardware, aside from the current production VOB servers, may be required depending on the overall size of the VOB databases. If any VOB database is tens of gigabytes in size or greater, or the duration for the database reformat may take longer than a site-acceptable amount of time, a VOB could be offline within a production environment. The following additional considerations should be made:

1. Additional VOB reformat server(s) of the same hardware architecture as the existing VOB servers, with multiple CPUs and abundant free memory.

2. Where feasible, Solid state drives should be used in place of conventional hard drives to reduce I/O latency and decrease random access time to a fraction of a millisecond. With the advances of solid state drives, an Enterprise-class Redundant Array of Independent Disks controller could be leveraged to further increase the performance and reliability over the entire solid state drive subsystem. For example, an IBM System X server: **ServeRAID B5015 SSD Controller**.

3. If solid state drives are not available, multiple traditional rotating magnetic hard disk drives could be joined together by an enterprise-class hardware controlled Redundant Array of Independent Disk drive striping solution.

4. In order for a reformat to complete successfully, free disk space of approximately 2 ½ times the size of the VOB database being reformatted is required.

VOB Database Evaluation

The following script can be used to gather countdb and du related information from the VOBs hosted on the VOB server:

```
#!/usr/bin/csh

set HOST=`uname -n`
mkdir -p /var/tmp/ibm

foreach VOB (`/usr/atria/bin/cleartool lsvob -host $HOST -quick | grep -v "*" | awk '{ print $2 }' |
xargs`)
        echo "$VOB - Started: `date`"
        set VBS=`echo $VOB | awk -F"/" '{ print $NF }'`
        echo "$VOB -" >> /var/tmp/ibm/$VBS-countdb.out
        /usr/atria/etc/utils/countdb $VOB/db/vob_db >>& /var/tmp/ibm/$VBS-countdb.out
        du -k -H -s $VOB/db >> /var/tmp/ibm/du.log
        echo "$VOB - Ended: `date`"
end
```

```
foreach VOB (`/usr/atria/bin/cleartool lsvob -host $HOST -quick | grep "*" | awk '{ print $3 }' |
xargs`)
        echo "$VOB - Started: `date`"
        set VBS=`echo $VOB | awk -F"/" '{ print $NF }'`
        echo "$VOB -" >> /var/tmp/ibm/$VBS-countdb.out
        /usr/atria/etc/utils/countdb $VOB/db/vob_db >>& /var/tmp/ibm/$VBS-countdb.out
        du -k -H -s $VOB/db >> /var/tmp/ibm/du.log
        echo "$VOB - Ended: `date`"
end

echo "Top 10 VOB Databases (Kb):"
cat /var/tmp/ibm/du.log | sort +rn | head -10
```

The below Perl script will collect the top 10 databases by size:

```
#!/usr/bin/Perl -w

my $hostname = `hostname`;
chomp($hostname);
my $voblist = `cleartool lsvob -host $hostname`;
@voblist = split(/\n/,$voblist);
@vobsizes = ();
foreach $vob (@voblist)
{
        $dirsize = 0;
        # printf ($vob);
        if (substr($vob,0,1) eq "*")
        {
                # printf ("{mounted}\n");
                $vob=~/\*\s(\\\S*)\s*([\\\S\.]*)/;
                # print $1."\t".$2."\n";
                $vtag = $1; $gpath=$2;
        }
        else
        {
                # printf ("{not mounted}\n");
                $vob=~/\s*(\\\S*)\s*([\\\S\.]*)/;
                # print $1."\t".$2."\n";
                $vtag = $1; $gpath=$2;
        }
        opendir(DIRH,$gpath."\\db");
        @dirlist=readdir(DIRH);
        foreach $fileitem (@dirlist)
        {
                if (substr($fileitem,0,2) eq "vo")
                {
                        ($dontcare,$dontcare,$dontcare,$dontcare,
                         $dontcare,$dontcare,$dontcare,$size,
                         $dontcare,$dontcare,$dontcare,$dontcare,
                         $dontcare) = stat($gpath."\\db\\".$fileitem);
                        # print("$size\t$fileitem\n");
                        $dirsize += $size;
                }
        }
        # printf("$gpath\\db\t$dirsize\n");
        push(@vobsizes,{tag=>$vtag,gp=>$gpath,ds=>$dirsize});
}

#foreach $vobsize (@vobsizes)
#{
#       print $vobsize->{'gp'} . "\t" . $vobsize->{'ds'} . "\n";
#}

@sortedsizes = sort {$b->{'ds'} <=> $a->{'ds'}} @vobsizes;

print "Top 10 Databases by size:\n";
$printed = 0;
foreach $vobsize (@sortedsizes)
{
        print $vobsize->{'tag'} . "\t" . $vobsize->{'gp'} . "\t" .
            $vobsize->{'ds'} / 1024 . "\n";
        last if ($printed++ >= 10);
}
```

Once you have been able to determine which VOBs have the largest database based on the "Top 10 VOB Databases" output in Kilobytes, one can begin the process of verifying how long an individual reformat VOB will take by using the native db_dumper and db_loader utilities. Wrapping the database tools with timex or time on UNIX/Linux and timeit on Windows (shipped within the **Windows Server 2003 Resource Kit Tools**) will provide an accurate recording of the duration the dump and load steps took to complete.

WARNING: Before you begin, **backup the VOB**. Under no circumstances should these commands be run on a production or live VOB!

PREREQUISITES:

- The VOB should be backed up in the event a restore is required: the VOB database should be locked for the duration of the backup process.

- The same identities and restrictions that apply to reformatvob also apply to running the db_dumper and db_loader commands. For more information, refer to the **IBM Rational ClearCase Information Center** under the topic of **reformatvob** (cleartool man reformatvob).

- Be sure to know which schema version (53 or 54) you are working with at the following levels:

    - **VOB level:** You can determine which schema a VOB is using by reading the **vob_db_schema_version** file in the db directory of the VOB.

    - **Server level:** You can determine which schema a Server was installed with by reading the **vob_db_schema_version** file in the install directory as follows:

UNIX and Linux: /opt/rational/clearcase/etc
Windows: C:\Program Files\Rational\ClearCase\bin

Note: VOB schema 53 is no longer supported as of ClearCase 7.0. Refer to **Technote 1201579 Support Policy for VOB schema 53 and ClearCase 7.0** for further information. However, it is still possible to run db_dumper.53 on a ClearCase version 8.0 VOB server to bring a VOB's database up to Schema 80. The db_dumper.53 is still available within the "dumpers" directory as noted below.

INSTRUCTIONS:

db_dumper

1. Log in as *root* (UNIX and Linux) or *Administrator* (Windows)

2. Open a command prompt and **Lock** the VOB (cleartool lock vob:<vob-tag>)

3. Copy the VOB db directory to a temporary location. The location should be based on the planned migration or reformat server and chosen storage to be used. Note that the server should be based on a stand-alone ClearCase 8.0 installation which acts as its own VOB, View and Registry server. This is to prevent accidental reformatting of the production VOB and will act as a trial for the proposed migration plan.

   Since this is a **copy of the db** directory, the identity information (UNIX/Linux: *.identity* directory and Windows: *identity.sd* file) is required along with the db directory. Be sure to maintain the existing directory structure when copying to the new location:

   UNIX example:

   ```
   <storage path>/test.vbs/
                           .identity/
                           db/
   ```

   Windows Note: Use a copy utility that will preserve the permissions on the files like **ccopy** found in the *C:\Program Files\Rational\ClearCase\etc\utils directory*.

4. Change into the **db** directory of the VOB storage in the copy made in step #3.

5. Run the db_dumper utility that matches the VOB schema level.

   For example. If you are migrating from ClearCase 2003.06 or earlier installation and the server was installed as Schema 53 and you need to dump a VOB that is presently at Schema 53, you need to use the *db_dumper.53* (UNIX and Linux) or db_dumper.53.exe (Windows) utility.

There are separate dumpers for schema 53 and 54, each of which are installed within the following locations:

UNIX and Linux: */opt/rational/clearcase/etc/dumpers*

Windows: *C:\Program Files\Rational\ClearCase\bin\dumpers*

Notes: There is also a version of the db_dumper (without a 53 or 54 in the name) on the host as well. This version matches Schema 54 or 80 on a ClearCase version 8.0 server. Use only the version in the directory specified above when manually dumping the database of the VOB.

Example:

- UNIX/Linux:
  /opt/rational/clearcase/etc/dumpers/db_dumper.# <path to db directory>

  Example:

  ```
  %>pwd
  /export/vobs/test.vbs/db

  %>timex /opt/rational/clearcase/etc/dumpers/db_dumper.54 /export/vobs/test.vbs/db

  0 hyperlinks dumped.
  44 events dumped.
  1 locks dumped.
  Dumper done.
  ```

- Windows:
  timeit C:\Program Files\Rational\ClearCase\bin\dumpers\db_dumper.#.exe <path to db directory>

  Example:

  ```
  C:\ClearCase_Storage\VOBs\test.vbs\db>"C:\Program
  Files\Rational\ClearCase\bin\dumpers\db_dumper.54.exe"
  C:\ClearCase_Storage\VOBs\test.vbs\db

  0 hyperlinks dumped.
  44 events dumped.
  1 locks dumped.
  Dumper done.
  ```

The last line of the output on the command line will read **Dumper done** if it was successful.

Note: Attempts to use a relative path from the db directory will result in the following error:

```
C:\ClearCase_Storage\VOBs\test.vbs\db>db_dumper .
db_dumper: Error: Unable to open file "\replica_uuid": No such file or directory.
```

db_dumper: Error: Cannot open database in '.': db_dumper: Error: Error from libdb (1)

This command will generate 3 files from the existing database files and store them in the db directory:

pass1_dump_file
pass2_dump_file
string_dump_file.

db_loader

Note: These instructions assume the directory structure remains from running the above db_dumper, and that the current working directory is within the db directory.

WARNING: Two points that should be adhered to:

- The VOB must be dumped by the originating platform and architecture prior to any planned migration to a different architecture. For instance, regardless of whether x86 platforms have the same endianness between Windows, Linux and Solaris x86 operating systems, the originating operating system and platform **must** dump the database prior to the copying of the database for load.

- NEVER execute the db_loader against the same directory that contains the database which was just dumped (as was done in the previous section). This will cause db_loader to overwrite the unformatted database when creating the reformatted version. This will lead to problems if the db_loader fails for some reason. If this does occur, the database will now be within an inconsistent state and will have to be re-copied.

  A new directory should be created to load the database into:
  1. Rename the old db directory to "db.old"
  2. Create a new directory called "db"
  3. Load into the new "db" directory
  4. The "db.old" directory can be safely removed once the VOB is back online and has been tested.

Instructions to dump and load the database:

1. The db_loader requires the three files from a dump phase (pass1_dump_file, pass2_dump_file, string_dump_file) to be in the db directory.

2. Run the db_loader. This will re-generate the files for the db directory that will be logically identical to these files when the dumper was run. If successful, the last line of the db_loader output will be **Loader done**.

   - UNIX/Linux:
     opt/rational/clearcase/etc/db_loader <path to db directory>

Example:

```
%>pwd
/export/vobs/test.vbs/db

%>timex /opt/rational/clearcase/etc/db_loader /export/vobs/test.vbs/db
Dumped schema version is 54
44 events loaded.
68 pass 2 actions performed.
Loader done.
```

- Windows:

  ```
  timeit C:\Program Files\Rational\ClearCase\bin\db_loader <path to db directory>
  ```

  Example:

  ```
  C:\ClearCase_Storage\VOBs\test.vbs\db>db_loader
  "C:\ClearCase_Storage\VOBs\test.vbs\db"
  Dumped schema version is 54
  44 events loaded.
  68 pass 2 actions performed.
  Loader done.
  ```

The last line of the output on the command line will read **Loader done** if the reformat was successful.

Note: Attempts to use a relative path from the db directory will result in the following error:

```
C:\ClearCase_Storage\VOBs\test.vbs\db>db_loader .
db_loader: Error: Unable to open file "\replica_uuid": No such file or directory.
db_loader: Error: Cannot open database in '.': db_loader: Error: Error from libdb (1)
```

Be Proactive:

All VOB databases should run through a test dump and load prior to any planned Schema 80 update. Not only will this exercise verify that the databases are consistent, but in the event there are db_dumper or db_loader errors, a Problem Management Request (PMR) can be opened to address those issues well ahead of the planned migration. This will ensure a seamless update and will minimize user downtime.

Furthermore, by following the steps above, an accurate time to reformat can be established. This is important as the total time to reformat all VOBs may change the approach to update the environment and a structured approach to update rather than a one-time update may be the preferred method.

Improving time to reformat

Based on the above times for the complete reformat of the VOB databases, the resources of the dumping and loading systems have a direct impact on the overall time to complete the database operations. If the desire is to significantly reduce the time to reformat the VOB databases, the above times can be used as a reference point to gauge overall improvements as adjustments are made.

Furthermore, there are many benchmarking utilities obtained from the Operating System or Hardware vendor as well as the Web which can accurately provide statistics concerning the overall I/O and throughput of a server. Based on those numbers, the individual vendors can be engaged to provide greater assistance with the correct tuning of the operating system and provide further hardware guidance in the event additional resources are required to realize the full potential of the server environment.

In addition to what has been stated within the Hardware Requirements section:

1. Additional VOB reformat server(s) of the same hardware architecture as the existing VOB servers with multiple CPUs and abundant free memory will help spread the reformat load between systems.

2. Where feasible, Solid state drives should be used in place of conventional hard drives to reduce I/O latency and decrease random access time to a fraction of a millisecond. With the advances of solid state drives, an Enterprise-class Redundant Array of Independent Disks controller could be leveraged to further increase the performance and reliability over the entire solid state drive subsystem. For example, an IBM System X server: **ServeRAID B5015 SSD Controller**

3. If solid state drives are not available, multiple traditional rotating magnetic hard disk drives could be joined together by an enterprise-class hardware controlled Redundant Array of Independent Disk drive striping solution.

4. 2 ½ times the size of a VOB database is required in free disk to reformat a single VOB database. Therefore, a review of the available disk space must be accomplished prior to starting multiple reformats.

5. The reformat operation will leave the successfully reformatted database in the VOB storage directory as a directory named "db.mm.dd". This directory should be removed after a successful reformat if disk space is an issue.

Reduce the Database Size for MultiSite VOB replicas:

The default settings within the */var/adm/atria/config/vob_scrubber_params* file for scrubbing oplogs are as follows:

```
# Oplog Scrubbing (applies to MultiSite replicated VOBs only)
# Format of entry:
#    oplog -keep {<n> | forever}
# -keep
#   Keep operation log entries for at least <n> days or forever.

oplog -keep forever
```

By reviewing the /var/tmp/ibm/*-countdb.out files, the values of the OPLOG_ENTRY can be determined: If the values are in the tens of thousands or even millions, scrubbing the oplog entries by changing the retention to 90 days and then scrubbing will reduce the loaded database size.

```
#!/usr/bin/csh

foreach VOB (`grep -l OPLOG_ENTRY /var/tmp/ibm/*-countdb.log | xargs`)
        set OPLOG=`grep OPLOG_ENTRY $VOB | awk '{print $1,"\t",$3}'`
        echo "$VOB - $OPLOG"
end

echo "Top 10 VOBs (oplog count):"
cat /var/tmp/ibm/*-countdb.log | grep OPLOG_ENTRY | sort +2rn | head -10
```

The below Perl script will collect and print oplog counts for the 10 VOBs with the most oplog entries:

```
#!/usr/bin/Perl -w
use Config;

my $prodhome;
my $countdb;
my $dontcare;
if ($Config{osname} eq "MSWin32")
{
            use Win32::Registry;
            my $regkey;
            # find ClearCase install directory
            # Try 32-bit host regkey first
            $result = $::HKEY_LOCAL_MACHINE-
>Open("Software\\Atria\\ClearCase\\CurrentVersion",$regkey);
            $regkey->QueryValueEx("ProductHome", $dontcare, $prodhome);
            #
            # Cannot find ProductHome value where a 32-bit host or 32-bit Perl
            # expects to see it. Check 64-bit location.
            #
            if (!(defined $prodhome))
            {
                        $result = $::HKEY_LOCAL_MACHINE-
>Open("Software\\Wow6432Node\\Atria\\ClearCase\\CurrentVersion",$regkey);
                        $regkey->QueryValueEx("ProductHome", $dontcare, $prodhome);
                        if (!(defined $prodhome))
                        {
                                    print "ClearCase is not installed!\n";
                                    return -1;
                        }
            }
```

```
        # set $countdb to the full path to the executable
        $countdb="\"" . $prodhome."\\etc\\utils\\countdb\"";
}
else
{
        $prodhome = "/opt/rational/clearcase";
        $countdb = $prodhome."/etc/utils/countdb";
}
# print "$countdb\n";
my $hostname = `hostname`;
chomp($hostname);
my $voblist = `cleartool lsvob -host $hostname`;
@voblist = split(/\n/,$voblist);
@vobsizes = ();
foreach $vob (@voblist)
{
        $oplog_count = 0;
        # printf ($vob);
        if (substr($vob,0,1) eq "*")
        {
                # printf ("{mounted}\n");
                $vob=~/\*\s(\\\S*)\s*([\\\S\.]*)/;
                # print $1."\t".$2."\n";
                $vtag = $1; $gpath=$2;
        }
        else
        {
                # printf ("{not mounted}\n");
                $vob=~/\s*(\\\S*)\s*([\\\S\.]*)/;
                # print $1."\t".$2."\n";
                $vtag = $1; $gpath=$2;
        }
        if ($Config{osname} ne "MSWin32")
        {
                chdir($gpath."/db");
                $cmdout = `$countdb vob_db | grep OPLOG_ENTRY`
        }
        else
        {
                $batchfile = $ENV{"TEMP"}."\\countoplogs.cmd";
                # print $batchfile."\n";
                unlink($batchfile);
                open(BATFILE,">$batchfile");
                #
                # Windows "pushd" command used because it sets a
                # temporary drive letter. This allows us to handle
                # both local and remote (NAS) VOB storage.
                #
                print BATFILE "pushd $gpath\\db\n";
                print BATFILE $countdb." vob_db\n";
                print BATFILE "popd\n";
                close(BATFILE);
                $cmdout = `\"$batchfile\" 2>&1 | findstr "OPLOG_ENTRY"`;
        }
        # print $cmdout;
```

```
 if (length($cmdout) > 11)
 {
         $cmdout =~ /(\S*)\s*\:\s*(\S*)/;
         # print "$1\t$2\n";
         $oplog_count = $2;
 }
 else
 {
         $oplog_count = 0;
 }
 # printf("$vtag\t$gpath\\db\t$oplog_count\n");
 push(@vobsizes,{tag=>$vtag,gp=>$gpath,oc=>$oplog_count});
}

#foreach $vobsize (@vobsizes)
#{
#        print $vobsize->{'gp'} . "\t" . $vobsize->{'ds'} . "\n";
#}

@sortedsizes = sort {$b->{'oc'} <=> $a->{'oc'}} @vobsizes;

print "Top 10 Databases by Oplog Count:\n";
$printed = 0;
foreach $vobsize (@sortedsizes)
{
 print $vobsize->{'tag'} . "\t" . $vobsize->{'gp'} . "\t" . $vobsize->{'oc'} . "\n";
 last if ($printed++ >= 10);
}
```

IMPORTANT:

You should verify that all replicas being scrubbed have synchronized out to other family members prior to making the vob_scrubber_params change. If the oplog changes have not been replicated prior to being scrubbed, the scrubbed replica will loose the ability to send out the scrubbed oplog changes. If this occurs, a new replica will need to be created. Be conservative by changing the oplog retention to 90 days as a start. Be careful not to be overly aggressive with the retention value!

Methods to reformat VOB Databases

MultiSite Replicas - Minimizing Downtime for large databases

When working with very large databases, care must be taken to prevent a significant amount of production downtime while reformatting the database. The following steps will allow the preservation of all View relationships with a replica and will significantly minimize the amount of impact to production. These steps have been tested and must be followed as instructed within the documentation.

WARNING: Before you begin, **backup the production VOB**!

WARNING: Because a copy of the replica will be made to facilitate the dump and load of the database, a standalone VOB, View and Registry server MUST be created to host the replica copy. ALL MultiSite replication should be disabled and DNS name resolution should be limited for this standalone VOB server so that the risk of a syncreplica packet leaving the system is minimized. Furthermore, **NO oplogable client-side cleartool or multitool operations** should be performed on the replica copy as divergence will immediately occur with the production replica VOB!

IMPORTANT: You should read through the documentation and familiarize yourself with the steps. If any questions should arise, you should open a Problem Management Request (PMR) with ClearCase support to discuss your concerns. The following steps are highlighted within the following online documentation:

ClearCase 8.0:
Administering Rational ClearCase
Administering Rational ClearCase MultiSite

ClearCase 8.0 Upgrade and Migration:

Prior to initiating the steps below, the production VOB server must be upgraded to ClearCase 8.0 as this will ensure a seamless transition to host Schema 80 databases when the reformatted VOB replica is migrated back. If another VOB host will be chosen to host the reformatted replica, ClearCase 8.0 must be installed on that system as well.

1. Lock the production VOB to all users and take a complete copy and full backup of the VOB replica (filer snapshot, and such): Note that the VOB replica COPY will remain LOCKED throughout the entire duration of the reformat on the standalone server. Be aware that VOBs with remote pools will need to be consolidated prior to the backup and copy.

```
bash-3.00$ cleartool lock vob:/vob/test
Locked versioned object base "/vob/test".
```

For further guidance on VOB Backup strategies as well as consolidating remote pools, review:

Backing up critical Rational ClearCase data
If the VOB has remote pools
Consolidating remote pools

2. Once the ORIGINAL VOB replica copy and full backup has completed, the ORIGINAL VOB can be unlocked to the production environment, thus minimizing downtime:

```
bash-3.00$ cleartool unlock vob:/vob/test
Unlocked versioned object base "/vob/test".
```

3. Prepare the standalone VOB, View and Registry server by disabling the MultiSite replication tasks. The following Jobs **MUST** be disabled within the ClearCase scheduler:
   A. Daily MultiSite Export
   B. Daily MultiSite Shipping Poll
   C. Daily MultiSite Receive
   D. Daily CC MultiSite Family Health Generation
   E. Daily MultiSite Syncmgr DB Scrubber
   F. Daily MultiSite Sync Manager Export
   G. Daily MultiSite Sync Manager Receive

   Review **The Rational ClearCase scheduler** for more information concerning the ClearCase scheduler.

   WARNING: Many sites have custom MultiSite synchronization scheduled and cron jobs as well as shipping methods to distribute packets to and from MultiSite VOB replica hosts: ensure that **ALL** custom MultiSite replication tasks and cron jobs have been disabled!

   Reduce the risk of a packet being shipped:
   One way to further reduce the risk of a packet being shipped from the standalone VOB server is to limit the Domain Name System hostname resolution. By limiting the system hostname resolution, the likelihood of a packet being shipped from this standalone server will be mitigated. By modifying either the */etc/netsvc.conf* (AIX) or */etc/resolv.conf* and */etc/nsswitch.conf* (covers most other UNIX and Linux hosts) as well as Local Area Connection properties for "Internet Protocol (TCP/IP)" on Windows hosts, name resolution can be controlled. To complete the task, the */etc/hosts* (UNIX and Linux) as well as the *C:\<windows_installation>\system32\drivers\etc\hosts* should be checked for any predefined hostnames to resolve.

4. Verify the standalone VOB, View and Registry server is hosting the Registry Server. From the standalone server, issue the following:

```
bash-3.00$ uname -n
standalone

bash-3.00$ cleartool hostinfo -long
Client: standalone
  Product: ClearCase 8.0
  Operating system: SunOS 5.10 Generic_137137-09
  Hardware type: sun4u
  Registry host: standalone
  Registry region: unix-standalone
  License host: @licenseserver
```

If the server is not configured to be a Registry host, refer to **The registry server host** and **Defining a Rational ClearCase registry host** to correctly configure the standalone ClearCase Server to host the ClearCase Registry operations.

Configure the standalone server to be unique: Be sure to use a region name that is unique from all other regions currently in use globally.

5. Once the standalone VOB server has been configured, move the COPY VOB to the new standalone VOB, View, Registry Server. The move process is identical to the steps highlighted within **Moving VOBs** for preserving permissions when migrating the data between hosts. For example:

UNIX and Linux Servers:

To move a VOB to a different host:

```
cd /vobstore

tar –cf – test.vbs | rsh standalone 'cd /vobstore2 ; tar –xBpf –'
```

Note: The –B option to the tar command might not be needed on some platforms. Also, the rsh command might have a different name, such as remsh, on some platforms. See the reference pages for your operating system.

Windows Servers (moving within the same domain is recommended):

Copy the VOB storage directory, preserving all ownership information. You must use a copy utility that preserves the ownership information contained in the VOB storage directory ACLs. See **Preserving NTFS ACLs when copying a VOB or view storage directory**. Use the Rational ClearCase ccopy utility, located in *ccase-home-dir\etc\utils\ccopy.exe* for this purpose. You must run ccopy on the host to which the VOB is being moved.

```
C:\ClearCaseStorage\VOBs> net use E: \\standalone\vobstg
C:\ClearCaseStorage\VOBs> ccopy test.vbs E:\test.vbs
```

6. Register and tag the locked VOB replica COPY on the standalone VOB, View, Registry server.

   WARNING: The COPY VOB must remain locked for the entire duration on the standalone VOB server. Under no circumstances can **ANY** oplogable client-side operation be performed on the COPY VOB otherwise **divergence** will immediately occur and the process will have to be redone to copy the VOB back from the production VOB server. Oplogable operations would include check-out, check-in and any other database modify operation that would be prevented when a VOB is locked.

7. Reformat the VOB replica COPY to Schema 80 on the standalone VOB server by using the cleartool reformatvob command:

   ```
   cleartool reformatvob -dump:
   bash-3.00$ cleartool reformatvob -dump /vobstore2/test.vbs
   ```

   Reformatting can take multiple hours for large VOBs. During the reformat, the VOB will be unavailable for normal use. If you install a different version of ClearCase or install ClearCase with different options, you may need to reformat again.

   If you are in doubt about the significance of the reformat operation or about how to perform VOB backups, see the documentation for administering ClearCase.

   If the reformat processing is interrupted for any reason, resume the processing by re-running the reformatvob command.

   ```
   Reformat versioned object base "/vobstore2/test.vbs"?  [no] yes
   Dumping database...
   0 hyperlinks dumped.
   0 derived objects dumped.
   46 events dumped.
   1 locks dumped.
   Dumper done.
   Dumped versioned object base "/vobstore2/test.vbs".
   Checking for VOB tag registry entry...
   ```

```
VOB tag registry entry found for versioned object base "/vobstore2/test.vbs".
```

```
cleartool reformatvob -load:
bash-3.00$ cleartool reformatvob -load /vobstore2/test.vbs
```

Backup the VOB before using this command! If the backups are not done properly, you are putting your data at risk.

Reformatting can take multiple hours for large VOBs. During the reformat, the VOB will be unavailable for normal use. If you install a different version of ClearCase or install ClearCase with different options, you may need to reformat again.

If you are in doubt about the significance of the reformat operation or about how to perform VOB backups, see the documentation for administering ClearCase.

If the reformat processing is interrupted for any reason, resume the processing by re-running the reformatvob command.

```
Reformat versioned object base "/vobstore2/test.vbs"?  [no] yes
cleartool: Warning: Renamed old database directory to
"/vobstore2/test.vbs/db.11.01".
cleartool: Warning: Please remove this database backup when you are satisfied with
the reformat.
Loading database...
Dumped schema version is 80
46 events loaded.
68 pass 2 actions performed.
Loader done.
Loaded versioned object base "/vobstore2/test.vbs".
The VOB is now unlocked.
```

Checking for VOB tag registry entry...
VOB tag registry entry found for versioned object base "/vobstore2/test.vbs".

Immediately lock the VOB after the database load has completed:

The completion of the reformatvob load operation will unlock the VOB for access. Be sure to immediately lock the VOB once the database has loaded successfully to prevent any errant write operation.

8. When the COPY VOB reformatvob completes and the COPY VOB database has been relocked, lock the ORIGINAL production VOB to all with the exception of the VOB Admin:

```
cleartool lock -nusers vobadm vob:/vob/test
```

```
Locked versioned object base "/vob/test".

cleartool lslock vob:/vob/test
--11-01T00:49  vobadm     lock versioned object base "/vob/test" (locked)
  "Locked except for users: vobadm"
```

9. Replicate out the changes from the ORIGINAL production VOB to at least one replica within the production family: verify that the changes have been replayed successfully at the remote site and the ORIGINAL replica changes were successfully applied to the replica family member.

Replica Synchronization:

Refer to **Replica synchronization** and be sure to note the following sections:

Assumption of successful synchronization
Manual synchronization

The ORIGINAL production VOB **must successfully synchronize out** one last time to at least one replica within the VOB family before the process can continue.

10. **IMMEDIATELY** lock the ORIGINAL production VOB to all users: rmtag and unregister the ORIGINAL production VOB.

```
cleartool lock -replace vob:/vob/test
Locked versioned object base "/vob/test".

cleartool lslock vob:/vob/test-reformat
--11-01T00:51  vobadm     lock versioned object base "/vob/test" (locked)
  "Locked for all users."

cleartool rmtag -vob -all /vob/test

cleartool unregister -vob /vobstore1/test.vbs
```

Making a VOB inaccessible

11. As root or the System Administrator, find the vob_server process for the ORIGINAL production VOB and kill the Process ID:

```
ps -ef | grep vob_server | grep test.vbs
vobadm  4059   601   0 22:49:26 ?        0:00 vob_server /vobstore1/test.vbs
-S13:19,udp:20,udp

kill -9 4059
```

```
ps -ef | grep vob_server | grep test.vbs
<process terminated>
```

12. **MOVE** the ORIGINAL production VOB Storage aside temporarily to preserve a way to recover in the event of a problem when bringing online the reformatted VOB Replica COPY VOB:

    ```
    mv /vobstore1/test.vbs /vobstore1/test.vbs_ORIGINAL
    ```

13. Copy the **locked** reformatted COPY VOB back to either the original production VOB Server host or to another ClearCase VOB server that is presently running ClearCase 8.0. The restore directory should be within the same named original production VOB directory: */vobstore1/test.vbs*

    Note: The directory may change if copying the VOB to a different VOB server host.

14. Register and tag the reformatted VOB. Relock the VOB to verify the lock is applied to all users:

    ```
    cleartool register -vob -host production -hpath /vobstore1/test.vbs /vobstore1/test.vbs

    cleartool mktag -vob -tag /vob/test /vobstore1/test.vbs

    cleartool lock -replace vob:/vob/test
    Locked versioned object base "/vob/test".
    ```

    Special Considerations for VOB Registry Entries:

    Review the following links depending on whether Network Attached Storage (NAS) is in use within the environment:
    Replacing a VOB server host storage for a NAS device

    For overall ClearCase Registry Administration Questions review:
    Administering the Rational ClearCase registry

15. Run a multitool restorereplica process to bring all the updates from the time interval between steps 1 and 14.

    Why is restorereplica required?

    To preserve the changes in oplogs that transpired when the ORIGINAL production VOB was locked to when the COPY of the production VOB was returned with a reformatted database to Schema 80, MultiSite is being leveraged to recover those missing oplogs from remote replicas. This ensures that the COPY of the replica being restored will be recovered to the identical condition of the ORIGINAL production VOB that is being replaced. This will ensure that Views will not require replacement when the COPY replica has been recovered and brought back into production.

WARNING: Review the restore replica documentation **fully and completely** as the MultiSite steps being recommended are contained within the following sections:

Restoring and replacing VOB replicas
Restoring a replica from backup

If **ANY** of the recommended steps are not clear or you require further assistance, you should open a Problem Management Request (PMR) and have those addressed prior to implementing any of the steps below.

A. Run checkvob against the restored VOB backup to check for and resolve any container issues before running the restorereplica command

NOTE: After the restorereplica command has locked the VOB for restoration, you can not interrupt the restorereplica process to make checkvob updates.

```
cleartool mkview -tag vobadm /viewstore1/vobadm.vws

cleartool checkvob -view vobadm -log /tmp/checkvob -pool -data -protections
/vobstore1/test.vbs
```

Warning: Fix **ALL** errors reported by the cleartool checkvob command before continuing. Review **About checkvob** for details about how the command functions and what types of errors are checked and reported.

B. Run the multitool command to restore the replica:

```
multitool restorereplica –invob /vob/test
Are you sure that you want to begin restoration operations for this VOB? [no]
yes
Comments for "original":
Recover from Reformat of Database
.
VOB locked and marked for restoration.
VOB set to require a restoration update from replica "remote-replica".
```

This places a special lock on the VOB object, which is separate from the ClearCase lock set during the backup process. Between this point and when you complete the final syncreplica –import operation later in this process, the syncreplica –import command adjusts the ClearCase lock temporarily to permit application of the update, then restores the full lock. During this time, only syncreplica –import can modify the replica.

C. Verify that all update packets have been processed at their destination replicas.

D. (Applicable only if the replica you are restoring was used to create one or more new replicas between the time of the backup/copy and the time of the restore, and the other replicas in the family do not have information about the new replicas.) The new replicas are unknown to your restored replica and all other replicas in the family, and lsreplica does not list them. If this is the case:

a) At each new replica, set the estimated states of the siblings to their actual states. (Use **chepoch –actual** or **lsepoch**/ **chepoch**. See **Recovering from lost packets**.)
b) At each new replica, export update packets to all other replicas in the family except the restored replica
c) Import the packets exported in the preceding step

E. At the restored replica, generate update packets for all other replicas, and send the packets to the sibling replicas.

You can send the packets using your standard synchronization method. To recover the replica more quickly, create the packets with syncreplica –export –fship

Because your replica is in the special restoration state, each outgoing update packet includes a special request for a return acknowledgment. It also includes your replica's old epoch numbers, which are now its current epoch numbers, by virtue of the restoration backup in the first step of this procedure. Each destination replica uses these numbers to roll back its row for your replica.

F. Wait for each other replica in the family to send an update packet to the restored replica. As in the preceding step, you can accelerate the creation and delivery of the update packets.

Collectively, these update packets include all operations that occurred between the time of the backup and the last update that your replica sent out before its storage was lost, even operations that originated at your replica. (The packets also include more recent operations that originated at other replicas.) In addition, each incoming packet includes the requested return acknowledgment from the sending host.

G. Process the incoming update packets with **syncreplica –import**. When your replica has received return acknowledgments from all other replicas in the VOB family, **syncreplica –import** reports that restoration of the replica is complete:

VOB has completed restoration: ...

H. (Applicable only if you had to perform Step 4) At one of the replicas that did not have information about the new replicas before the restoration procedure, export update packets to all of the new replicas and import the packets at the new replicas. (Do not perform this export from the restored replica.)

16. Unlock the VOB once the restorereplica finishes

```
cleartool unlock vob:/vob/test
Unlocked versioned object base "/vob/test".
```

17. The VOB replica has now successfully been upgraded to Schema 80, and the production environment can be brought back online

Reformatting small to medium sized databases

If the goal is to reduce the time spent reformatting VOB databases and, thus, limiting the impact of production downtime, the VOB servers should be at least within the minimum requirements specified within the **System Requirements for ClearCase 8.0**. Additional server hardware, aside from the current production VOB servers, may be required depending on the overall size of the VOB databases. If any VOB database is tens of gigabytes in size or greater, or the duration for the database reformat may take longer than a site-acceptable amount of time a VOB could be offline within a production environment, additional hardware may be required. As well, Snapshot Views can be leveraged during the reformat downtime to continue production.

In addition to what has been stated within the Hardware Requirements section:

1. Additional VOB reformat server(s) of the same hardware architecture as the existing VOB servers with multiple CPUs and abundant free memory will help spread the reformat load between systems.

2. Where feasible, Solid state drives should be used in place of conventional hard drives to reduce I/O latency and decrease random access time to a fraction of a millisecond. With the advances of solid state drives, an Enterprise-class Redundant Array of Independent Disks controller could be leveraged to further increase the performance and reliability over the entire solid state drive subsystem. For example, an IBM System X server: **ServeRAID B5015 SSD Controller**.

3. If solid state drives are not available, multiple traditional rotating magnetic hard disk drives could be joined together by an enterprise-class hardware controlled Redundant Array of Independent Disk drive striping solution.

4. 2 ½ times the size of a VOB database is required in free disk to reformat a single VOB database. Therefore, a review of the available disk space must be accomplished prior to starting multiple reformats.

Reduce the Database Size for MultiSite VOB replicas:

The default settings within the */var/adm/atria/config/vob_scrubber_params* file for scrubbing oplogs are as follows:

```
# Oplog Scrubbing (applies to MultiSite replicated VOBs only)
# Format of entry:
#    oplog -keep {<n> | forever}
# -keep
#    Keep operation log entries for at least <n> days or forever.

oplog -keep forever
```

By reviewing the */var/tmp/ibm/*-countdb.out* files, the values of the OPLOG_ENTRY can be determined: If the values are in the tens of thousands or even millions, scrubbing the oplog entries by changing the retention to 90 days and then scrubbing will reduce the loaded database size.

```
#!/usr/bin/csh

foreach VOB (`grep -l OPLOG_ENTRY /var/tmp/ibm/*-countdb.log | xargs`)
        set OPLOG=`grep OPLOG_ENTRY $VOB | awk '{print $1,"\t",$3}'`
        echo "$VOB - $OPLOG"
end

echo "Top 10 VOBs (oplog count):"
cat /var/tmp/ibm/*-countdb.log | grep OPLOG_ENTRY | sort +2rn | head -10
```

IMPORTANT:

Verify that all replicas being scrubbed have synchronized out to other family members prior to making the vob_scrubber_params change. If the oplog changes have not been replicated prior to being scrubbed, the scrubbed replica will loose the ability to send out the scrubbed oplog changes. If this occurs, a new replica will need to be created. Be conservative by changing the oplog retention to 90 days as a start. Be careful not to be overly aggressive with the retention value!

Using MultiSite for non-replicated VOBs

If a VOB has not been replicated prior to the upgrade of ClearCase 8.0, and you wish to do so at this time, consider the following points:

1. The destination replica server is ClearCase 8.0.
2. The time to generate the MultiSite mkreplica packet does not conflict with the amount of acceptable time determined that production development work can be impacted. Review Timing of replica creation as well as Creating VOB Replicas for the number of issues surrounding making a replica.
3. Using Snapshot Views to continue production work while the MultiSite mkreplica packet is being generated.

Once the replica has been created, the steps on migrating with a MultiSite replica can now be followed. However, by following this specific path to reformat the database, the steps will double the time required to complete the migration.

If there are any concerns about implementing MultiSite with the upgrade to ClearCase 8.0, you should open a Problem Management Request (PMR) to discuss these concerns.

Notes and warnings

You should read and fully understand the steps required to reformat a VOB. If any steps within this documentation are not clear, you should open a Problem Management Request (PMR) to address these concerns.

References

The following were used within references hyperlinks as stated within the above documentation:

- **New features in ClearCase and ClearCase MultiSite version 8.0**
  http://www.ibm.com/support/docview.wss?uid=swg21566805

- **About Feature Levels and ClearCase**
  http://www.ibm.com/support/docview.wss?uid=swg21119269

- **Compatibility across releases**
  http://www.ibm.com/support/docview.wss?uid=swg21566805

- **System requirements for ClearCase 8.0**
  http://www.ibm.com/support/docview.wss?uid=swg27020796

- **ServeRAID B5015 SSD Controller**
  http://www.redbooks.ibm.com/abstracts/tips0763.html?Open

- **Windows Server 2003 Resource Kit Tools**
  http://www.microsoft.com/download/en/confirmation.aspx?id=17657

- **Technote 1201579 Support Policy for VOB schema 53 and ClearCase 7.0**
  http://www.ibm.com/support/docview.wss&uid=swg21201579

- **Backing up critical Rational ClearCase data**
  http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/c_backup.htm

- **Moving a VOB on Linux or UNIX**
  - **If the VOB has remote pools**
    http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/t_mvvob_ux_pools.htm

  - **Consolidating remote pools**
    http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/t_mvvob_ux_pools_consolidating.htm

- **Administering the Rational ClearCase registry**
  http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/c_rgy_admin.htm

- **The Rational ClearCase scheduler**
  http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/c_maint_sched.ht

m

- **The registry server host**
http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/c_rgy_host.htm

- **Defining a Rational ClearCase registry host**
http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/c_rgy_host.htm

- **Preserving NTFS ACLs when copying a VOB or view storage directory**
http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/c_tshoot_acls_preserving.htm

- **Replica synchronization**
    - **Assumption of successful synchronization**
    http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_ms_admin.doc/c_assuming_synch.htm
    - **Manual synchronization**
    http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_ms_admin.doc/c_manual_synch.htm

    Making a VOB inaccessible
    http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/c_vobadm_noaccess.htm

- **Replacing a VOB server host storage for a NAS device**
http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/t_vobadm_replace_nas.htm

- **Restoring and replacing VOB replicas**
http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_ms_admin.doc/c_restore_vob_replicas.htm

    - **Restoring a replica from backup**
    http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_ms_admin.doc/t_restore_replica_backup.htm

- **About checkvob**
http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?

topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/c_maint_checkvob.htm

- **Recovering from lost packets**
  http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_ms_admin.doc/c_recovering_from_lost_pckts.htm

- **Timing of replica creation**
  http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_ms_admin.doc/c_replica_creation_timing.htm

- **Creating VOB Replicas**
  http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_ms_admin.doc/c_creating_vob_replicas.htm

- **IBM Rational ClearCase Information Center**
  http://www.ibm.com/support/docview.wss?uid=swg27010158

- **Administering Rational ClearCase 8.0**
  http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_admin.doc/topics/c_admin_container.htm

- **Administering Rational ClearCase MultiSite 8.0**
  http://publib.boulder.ibm.com/infocenter/cchelp/v8r0m0/index.jsp?topic=/com.ibm.rational.clearcase.cc_ms_admin.doc/c_ccms_container.htm