WANdisco Subversion MultiSite Procedures and Troubleshooting

Revision History

REVISION	DATE
1.0	December 2008
2.0	February 2009
3.0	July 2009
4.0	September 2009

Contents

Contents, cont'd

Contents, cont'd

1 Procedures

1.1 Temporarily Disabling User Access to Subversion

NOTE:

This step makes Subversion read-only. Please follow your company guidelines on notifying Subversion users of maintenance.

1.1.1 Temporarily Disabling Subversion Access At Selected Nodes

You can stop transactions at one or more nodes. For a discussion of stopping one, but not all, nodes, see WANdisco is Listening in *About Subversion MultiSite*.

NOTE:

Do not use this procedure if you want to do a synchronized stop. For a synchronized stop, see Performing a Synchronized Stop.

Step 1 After notifying your users of the downtime, navigate to the Dashboard.

Step 2 On the Proxy tab, click **Stop Proxy**.

Step 3 Select **Stop this proxy only**.

Step 4 Repeat steps 2 and 3 for each node you want to stop.

Step 5 To start a node, click **Start Proxy**. The replication group catches up the re-started node on any transactions that occurred at the nodes that remained up.

Step 6 Remember to re-start each node that you stopped.

1.1.2 Temporarily Disabling Subversion Access At All Nodes

To stop all nodes at once, you do a synchronized stop. See Performing a Synchronized Stop.

1.2 Establishing a Baseline for Replication

Before starting WANdisco, you should ensure that all nodes start with an identical copy of the repository (the svnroots) - identical in all respects, except as noted below.

Depending on the size of your repository and available bandwidth to the remote nodes, you can decide whether to copy or sync the repository over the network or ship a copy of the repository on a physical medium (for example, a CD, DVD or hard disk). Select the method that works best for your situation.

If you already have an older copy of the repository at the remote nodes, for example, if, prior to deploying WANdisco, you were using a master-slave replication solution such as svnup, choose the **Synchronize** procedure.

1.2.1 Copying the Subversion Database

Otherwise, start by estimating how long it may take you to copy the repository over the network by determining the size of your repository and the bandwidth available to the remote nodes. If you conclude that it takes too long, you will want to ship the repository to the remote nodes on a physical medium.

Step 1 Determine the size of the repository. From a Unix command prompt, `cd` to your repository and type

```
du -s
```

This reports the size of your repository in kilobytes.

Step 2 Determine the network bandwidth. Copy a reasonable-sized file (say 100 megabytes) to the remote node using any means available (example, `scp` or `ftp`). Time the copy.

Step 3 Estimate how long the copy will take. Using the information gathered above, you can estimate how long it can take you to copy the repository to the remote nodes over the network. For example, if copying a 100 megabyte file over the network took 10 minutes, copying your 5 gigabyte repository may take about 500 minutes (8 hours and 20 minutes).

1.2.2 Synchronizing With an Older Remote Copy

You can use `rsync` to sync up an older remote copy with your master copy. For example, from the machine with the master copy of myRepository, type

```
rsync -rvlHt /path/to/myRepository remoteHost:/path/to
```

Note that the final element, `myRepository`, is not specified in the remotehost's path. For further information, consult the `rsync` man pages.

1.2.3 Copying Over the Network

Use this procedure if:

- You do not have an older remote copy; i.e., you are copying the entire repository over.
- Your repository is small enough.
- You have enough network bandwidth to copy the repository to the remote nodes in reasonable time.

Step 1 Ensure that the repository is not in use. If necessary, shut down the SCM server. For example, type

```
/etc/init.d/xinetd stop
```

Step 2 Package the master copy of your repository.

Step 3 Copy the package to the remote host.

Step 4 Log in to your remote host and unpackage the repository. For example, on the server with the master copy, type

```
cd /path/to
tar pzcf myRepository.tgz myRepository
ssh remoteHost mkdir -p /path/to
scp myRepository.tgz remoteHost:/path/to/
log into remoteHost
cd /path/to
tar pzxf myRepository.tgz
```

1.2.4 Shipping on a Physical Medium

If copying over the network may take too long, you can ship the repository to the remote destination on a physical medium, such as a CD, DVD or hard disk. Note that you do not have to wait for the baseline to be available at all nodes before using WANdisco. Instead, you can follow the procedure below.

Using WANdisco before the baseline is available at all nodes

Step 1 Deploy WANdisco as usual, but do not start the WANdisco server at the nodes where the baseline is not yet available.

Step 2 When choosing a quorum, ensure that the nodes where WANdisco can be started are sufficient to form a quorum. The simplest way to do this is to

choose the Singleton Quorum policy, and choose the node that has the master copy of the repository as the distinguished node.

Safe Differences

The only things that can safely differ in the baselines across your nodes are post-commit triggers. For example, if you generate email notifications from a post-commit trigger, it is a good idea to do that at only one node to avoid generating duplicate email notifications.

Common Pitfalls

It is important that the repositories are identical in all respects except as noted above. A common mistake when the desired baseline is an empty repository is to `init` a new empty repository at each node. Instead, you should `init` the repository at one node, and copy the empty repository to other nodes.

For Subversion, WANdisco will not work correctly unless this is done right. A Subversion repository incorporates a GUID (Globally Unique Identifier) that is generated at the time of the init. WANdisco requires this GUID to be identical at all nodes. The best way to ensure this is to copy the repository from one node to the others.

1.3 Finding the Last Committed Transaction

Even though committed transactions are always in the same order for each node, the timing of the commits usually varies from node to node. So unless there are no Subversion users logged in, you probably are going to have variations per node for committed transactions.

Go to any node's Dashboard. Type

```
http://<IP address>:<WANdisco port number>/dashboard2
```

You see all the nodes on the Dashboard to compare the listed transactions.

1.4 Adding a Repository to a Replication Group

Use this procedure to add a repository to an existing replication group.

Step 1 Ensure that all nodes in the replication group have an exact copy of the repository to add. (The repository has already been copied to all nodes.)

Step 2 Add the repository to the Apache configuration files, being consistent with the existing method. You should be using either SVNParentPath or SVNPath. For more information, see Strategies for Having Two or More Repositories.

Step 3 Specify the new repository for WANdisco. Go to the Proxy page and select **SVN Settings**. The Repository page appears.

a Browse to the repository root.

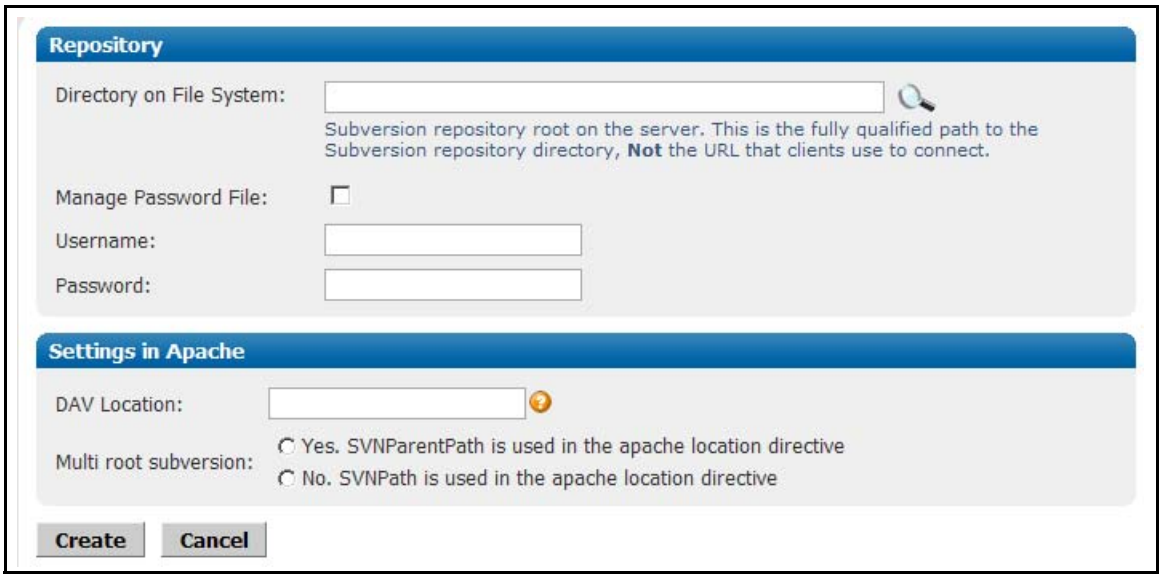
b If you want WANdisco to manage the password file, check the Manage Password File checkbox.

c If you checked the previous checkbox, browse to the password file's location.

d Enter a username and password for read access for the entire repository.

e Enter the repository's DAV location.

f Answer **Yes** or **No** if you are using multi root Subversion (really, WANdisco wants to know whether you are using SVNParentPath or SVNPath).



Step 4 Click **Create**.

Step 5 Perform a consistency check. See Checking Repository Consistency.

Step 6 Verify clients can see the new repository. Log in as a client and check out a file from the new repository.

Step 7 Notify users they can begin using the new repository.

1.5 Removing a Repository from the Replication Group

Step 1 Perform a synchronized stop of the replication group.

NOTES:

A synchronized stop completes only when all nodes are available. If one or more nodes are unavailable, the process suspends until they are available.

This step makes Subversion read-only. Please follow your company guidelines on notifying Subversion users of maintenance.

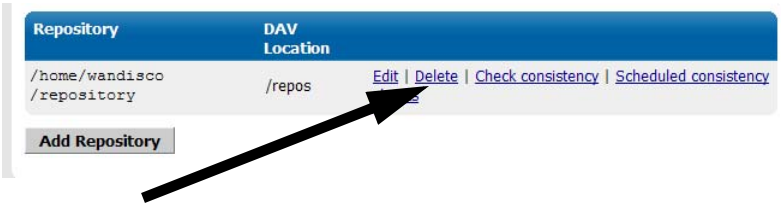
---

a Click on **Stop Proxy**. Go to the Proxy tab on the Admin Console. **Stop Proxy** is listed on the left.

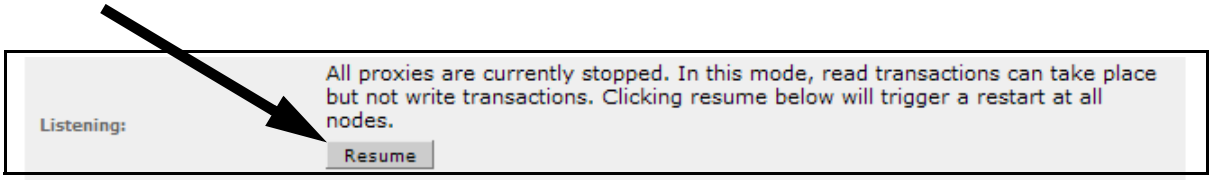
b Check **Synchronized stop of all proxies**. All nodes stop. Pending transactions get completed, but WANdisco accepts no further client transactions. The Subversion servers go into read-only mode.

Step 2 At all nodes in the replication group, delete the entry for the repository in the Apache configuration files, being consistent in repository specification. If you are using SVN ParentPath, use that method. If you are using the Location directive, specify that.

Step 3 Update WANdisco to reflect the change. Go to the Proxy page, and select **SVN Settings**. In the Repository section, click **Delete**. The change is replicated throughout the group.



Step 4 Restart the replication group. Go to the Proxy page and select the **Proxy Status** command. Click **Resume**.



Step 5 Verify clients can no longer see the repository. Log in as a client and verify that the repository is unavailable.

Step 6 Notify users they can no longer use the old repository.

Step 7 The repository is still in its same physical location.

1.6 Moving a Repository from One Replication Group to Another

Step 1 Remove the repository from the first replication group. See the procedure Removing a Repository from the Replication Group.

Step 2 Add the repository to the second replication group. See the procedure Adding a Repository to a Replication Group.

1.7 Adding a Node to the Replication Group

To add a node to an existing replication group, you have to create a new replication group that contains the new node and then activate the new replication group. See Creating a New Replication Group.

Determine a plan for copying the repository to the new node. See Strategies for Achieving Identical Repositories at All Nodes.

Step 1 Go to the **Nodes** command on the Proxy tab.

Step 2 Click **Create Node**.

Step 3 Fill in the fields on the Node Properties form, which are described in step 46 in the Installation section.

Step 4 Create a new replication group containing the new node. See Creating a New Replication Group.

The new replication group is activated with all nodes.

1.8 Removing a Node from the Replication Group

Only one replication group is active at a time.

You cannot delete a node from an active replication group. To delete a node that is part of a replication group, you must create a new replication group that excludes the node to be deleted, activate that new replication group, and then delete the node. See Creating a New Replication Group.

1.9 Creating a New Replication Group

Use this procedure when you want to create a new replication group. If the new replication group contains new nodes, you must contact WANdisco support (in the US at 877-207-1439, internationally at +1-925- 218-2300) to receive a new license file.

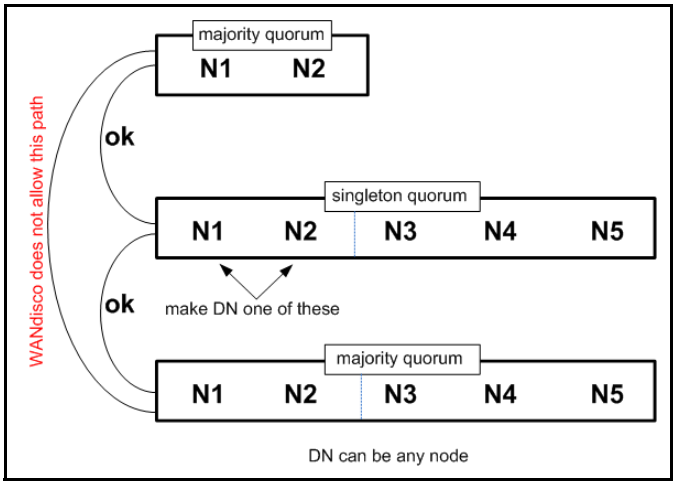
1.9.1 Considerations

- Only one replication group is active at a time.
- If possible, create a replication group from the node that is to be the distinguished node in the new replication group.
- If you have a current replication group, then you must first create another replication group, and then activate it. You cannot delete the current replication group.

1.9.2 Considerations for Majority Quorum Replication Groups

If you have majority quorum as the current replication group, and want another replication group with majority quorum that has more nodes than the current group, (say you are adding three nodes to an existing two node group, so that your new replication group would be five nodes), you implement your new replication group using a prescribed process. This is only for new groups whose majority would outweigh the current majority. If you add one or two nodes to a two node group, the original two nodes can still form a majority, so the standard procedure works fine.

By using this majority quorum-to-larger majority quorum method of creating replication groups, WANdisco ensures quorum integrity and protects against unnecessary down time. The new nodes can take a while to come online, depending on WAN latency and manual repository copying. WANdisco avoids having the original replication group be read-only while waiting for the new nodes to come up and form a majority quorum.



Follow these steps to successfully create and activate a new replication group (generic instructions for creating a replication group follow in the next section):

- Create a new replication group with all the nodes, but with singleton quorum and the distinguished node is the node you are working from.
- Activate the new replication group.
- Create another replication group that contains all five nodes, but has majority quorum. You can select the desired distinguished node.
- Activate the new replication group.

All nodes are up and running.

1.9.3 Procedure

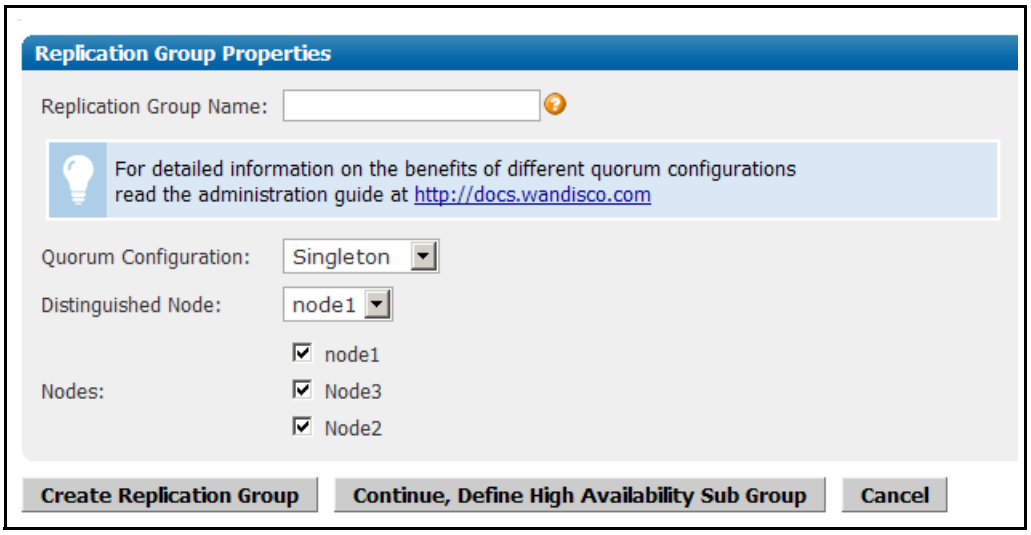
Step 1 Create any new nodes, if necessary. If a node has been previously defined, it is listed with the **Nodes** command on the Proxy tab. To create a node, click **Create Node**.

Enter all required information for the site.

Step 2 If you obtained a new license key, place it in this node's current `svn-replicator/config` directory.

Step 3 Create the Replication Group. On the Proxy tab, click **Replication Groups**. All previously defined replication groups are listed, including the current one.

Step 4 Click **Create Replication Group**.



Step 5 Name the replication group, and select the appropriate quorum. For information on quorum selection, see Replication Example in *About Subversion MultiSite*.

Step 6 Select the nodes that comprise the replication group.

Step 7 If you are defining a High Availability sub-group, click **Continue, Define High Availability Sub Group**. For information on defining High Availability sub groups, see step 52 in the Installation section of the Installation Guide.

If you are not defining a High Availability sub-group, click **Create Replication Group**.

Step 8 Activate the replication group. Click **Activate Replication Group**.

Step 9 Select the desired activation options.

Step 10 Verify that the Activate Information is accurate.

Step 11 Specify whether WANdisco is packaging the repository, or you are copying it manually.

Step 12 If you want to change the packaging directory from the default, do so in the Packaging Dir field.

Step 13 Verify that the displayed replication group (in the Activate Replication Group field) is the one you want to activate.

Step 14 Click **Continue**. The status displays, and then the replication group displays. Note that only the first node is activated at this time.

Step 15 Activate the other nodes by clicking **Activate Node(s)**.

All your nodes should be activated.

1.10 Deleting a Replication Group

Use this procedure when you want to delete a replication group. If the replication group you want to delete is the current one, you must first activate another replication group (if it does not already exist, you must create one), and then delete the desired replication group. Note that you can create a replication group with just one node.

NOTE:

Only one replication group is active at a time.

To delete a replication group that is not active, go to **Replication Groups** on the Proxy tab. Check the box next to the desired replication group, and click **Delete Selected**. The replication group is deleted.

To delete a replication group that is active, create another replication group, activate it, and then delete the desired replication group. See the previous procedure Creating a New Replication Group.

1.11 Emergency Reconfiguration of Quorum

Use this procedure as a last resort, for example, if one or more nodes are permanently unavailable and the surviving nodes do not constitute a quorum. Typically, this would mean the distinguished node in a singleton quorum becomes unreachable. This procedure basically creates a new replication group that excludes the unreachable node.

Do not use this procedure for a High Availability sub group.

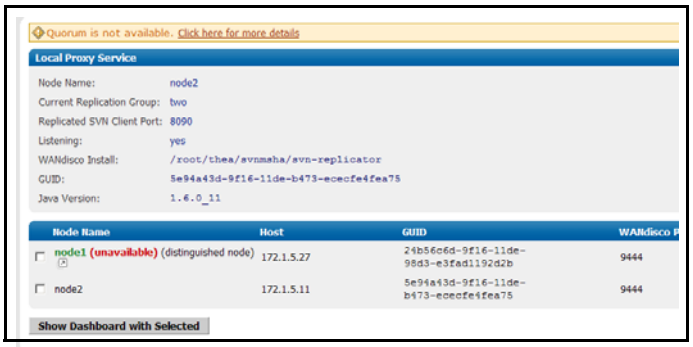
WARNING:

WANdisco does not recommend using this procedure, because of the potential for data loss. If the distinguished node is unreachable by other nodes, it could still be processing transactions on its own (with singleton quorum, the distinguished node can operate alone). Therefore, if the other nodes form a new quorum that excludes the distinguished node, any transactions made at the distinguished node since it became unreachable to the other nodes would be lost.

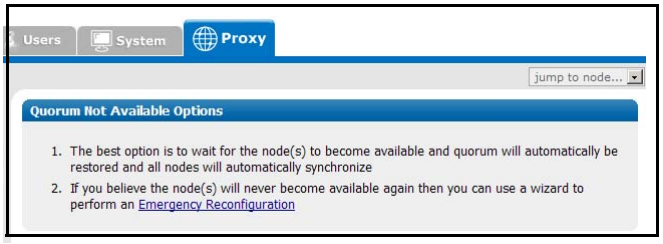
Step 1 Ensure users at the sites slated for elimination are not using the repository, since the changes they are making will be lost as a result of the emergency reconfiguration.

Step 2 Ensure that the sites slated for elimination are shut down, and will never be restarted.

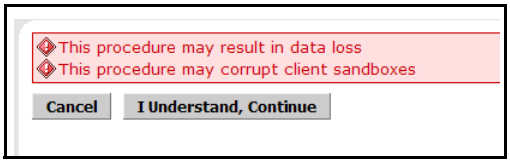
Step 3 Read the error message about not reaching quorum. Click the link **click here for more details**.



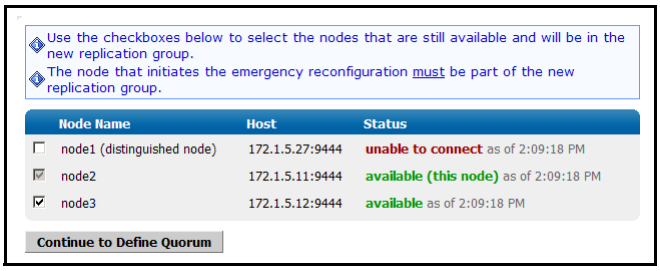
Step 4 Read the informational message. If you decide to continue, click **Emergency Configuration**.



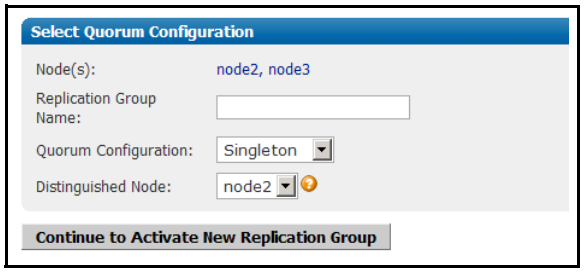
Step 5 Read another warning message about the possible data loss that can occur. You can wait to diagnose the other node, or click **I Understand, continue**.



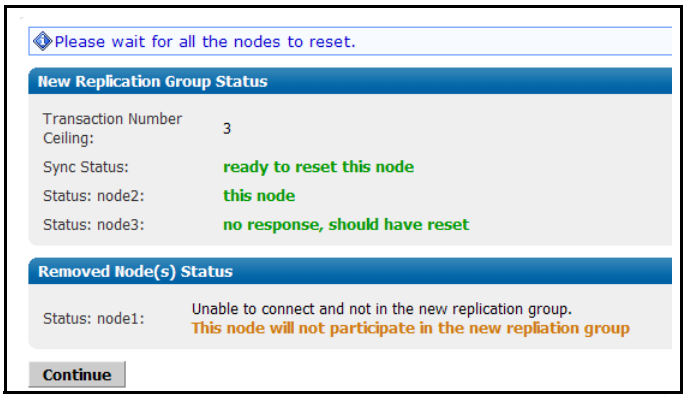
Step 6 Select the nodes for the new replication group.



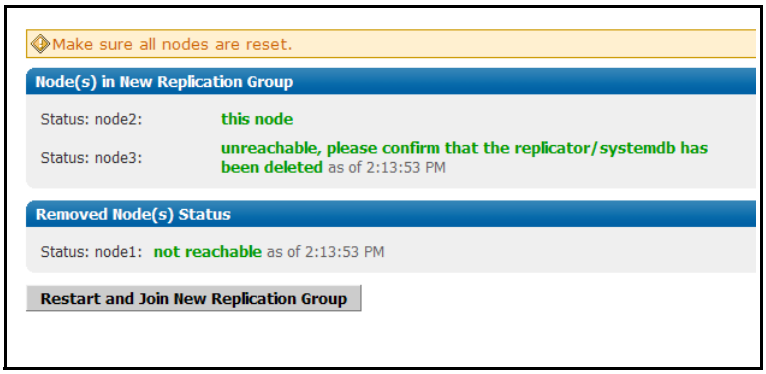
Step 7 Name the new replication group, and click **Continue to Activate New Replication Group**.



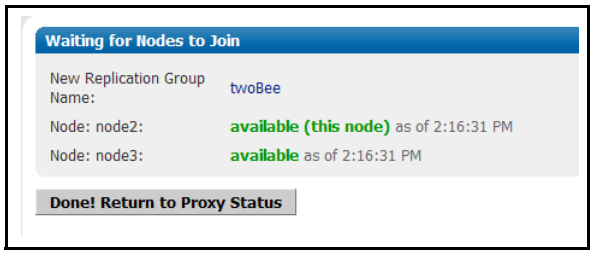
Step 8 When all the nodes are reset, click **Continue**.



Step 9 Verify that the systemdb has been deleted at the other nodes (other than the one you're on now), and click **Restart and Join New Replication Group**.



Step 10 After a few moments, you see that all nodes in the new replication group are available. These repositories are in synch.



1.12 Changing a prefs.xml File

The prefs.xml files for nodes are located in `svn-replicator/config`. Each file contains all preference information for the nodes in the group.

If you make changes that affect more than one node, you must change each node's specific file. But if your change affects just one node, you can change just that node's prefs.xml file.

1.12.1 Changing One Node's prefs.xml File

Step 1 Shut down the node or nodes where you changed the prefs.xml file. Go to the Proxy tab and click **Shut Down Node**.

Step 2 Make the desired changes to one or more prefs.xml file.

Step 3 Restart the node or nodes you stopped. At the command line, type

```
perl svnreplicator
```

The changes you made in the prefs.xml file are now in effect.

1.12.2 Changing All Nodes' prefs.xml File

Step 1 Perform a synchronized stop. See Performing a Synchronized Stop. That procedure includes resuming WANdisco.

Step 2 Make the desired changes to each node's prefs.xml file.

Step 3 Restart WANdisco at all servers. Click **Resume**.

1.13 Performing a Synchronized Stop

NOTES:

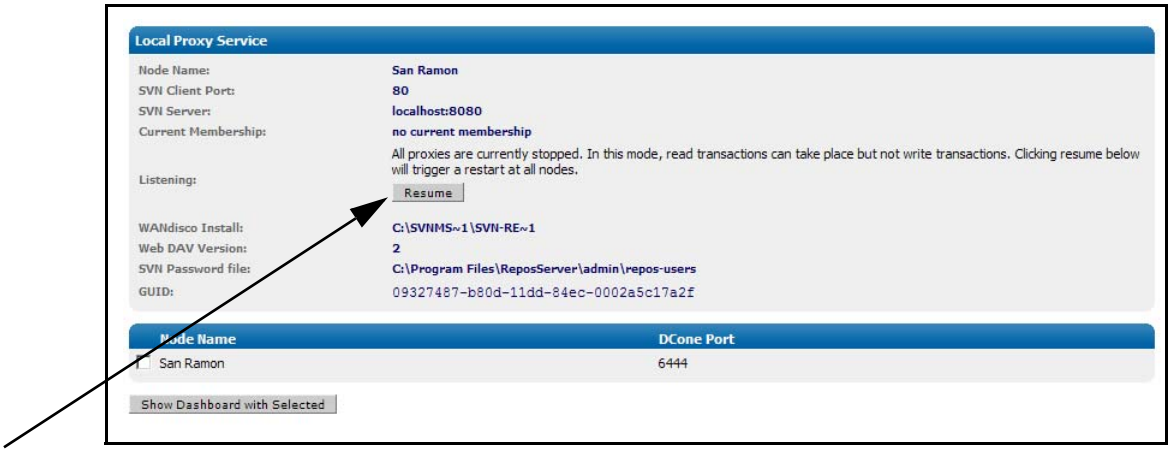
A synchronized stop completes only when all nodes are available. If one or more nodes are unavailable, the process suspends until they are available.

This procedure makes Subversion read-only. Please follow your company guidelines on notifying Subversion users of maintenance.

Step 1 Click on **Stop Proxy**. Go to the Proxy tab on the Admin Console. **Stop Proxy** is listed on the left.

Step 2 Check the **Synchronized stop of all proxies** radio button. All nodes stop listening. Pending transactions get completed, but WANdisco accepts no further client transactions. The Subversion servers go into read-only mode.

Step 3 Restart WANdisco at all the servers. Refresh the Proxy tab and click **Resume**.



The page refreshes, and you can see the Listening field display is now **yes**.

1.14 Verifying That the Replicator is Working

There are two ways you can check. You can make a minor change in Subversion on one client, wait a minute, and go to another client to ensure the change is reflected.

Another way to check if Subversion MultiSite is replicating, is to verify there are commit transactions posted to the log file `svn-replicator/logs/SVNProxyServer-prefs.log`.

```
INFO:  [listen-1] Listening on port : 0.0.0.0/0.0.0.0:6445
1219077847375 org.nirala.communication.transport.DConENet.AsyncConnector
makeConnection
INFO:  [main] Connection request to Node Id = c66b6db9-6a50-11dd-8675-
001aa036534c, host = 192.168.1.15, port = 6666, timed out in 500ms
1219077847875 org.nirala.communication.transport.DConENet.AsyncConnector
makeConnection
INFO:  [main] Connection request to DFTPEndpoint - Node Id =
192.168.1.156666, host = 192.168.1.15, port = 6666, timed out in 500ms
1219077848578 org.nirala.admin.DiskMon start
INFO:  [main] Diskmon is monitoring C:\Thursday\svn-ha\systemdb every
15min
1219077849000 org.nirala.communication.transport.DConENet.ListenReactor
setupListener
INFO:  [listen-1] Listening on port : 0.0.0.0/0.0.0.0:2403
1219077849000 org.nirala.communication.transport.svnproxy.ProxyServer
onStartedProxyListen
INFO:  [main] SVN Proxy listener is now turned ON at port :2403
1219077853765 org.nirala.communication.transport.DConENet.Listen-
Stage$TCPStopListening onStop
INFO:  [listen-1] Host: 0.0.0.0, Port: 2403 Stopped Listening.
1219077853765 org.nirala.communication.transport.svnproxy.ProxyServer
onStopProxyListener
INFO:  [p-queue-1] SVN Proxy listener is now turned OFF at port :2403
1219077872328 org.nirala.communication.transport.DConENet.ListenReactor
setupListener
INFO:  [listen-1] Listening on port : 0.0.0.0/0.0.0.0:2403
1219077872328 org.nirala.communication.transport.svnproxy.ProxyServer
onStartedProxyListen
INFO:  [mqueue-1] SVN Proxy listener is now turned ON at port :2403
```

1.15 Installing a .jar File Patch

Follow these steps to install a `svn-replicator.jar` patch to an existing Subversion MultiSite installation. You are going to copy the same jar file to each node's `lib` directory.

Step 1 Download the `svn-replicator.jar` file.

Step 2 Verify the md5 checksum. Type

```
md5sum <filename>
```

Step 3 At all the nodes, move the existing jar file to a back up directory. (All jar files in the `/lib` directories are in the WANdisco `CLASS` path.)

Step 4 Perform a synchronized stop. See Performing a Synchronized Stop.

Step 5 Copy the new jar file to each node's `lib` directory.

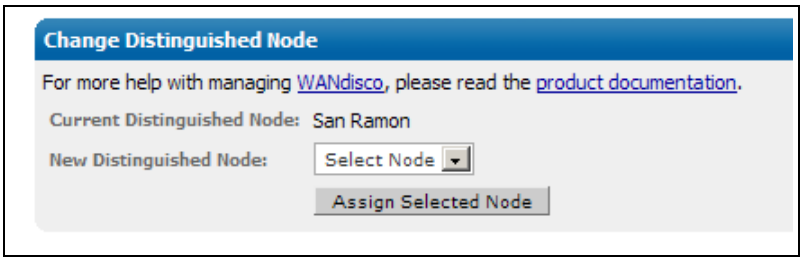
Step 6 Restart the group. Click **Resume** on the Proxy tab.

Step 7 Confirm the upgrade by checking the dashboard for the newer version, and check the log file under `svn-replicator/logs` for the start header with the new version.

1.16 Changing the Distinguished Node

Step 1 To change the Distinguished Node, go to the Admin Console.

Step 2 Click on **Change Distinguished Node** command on the left side menu.



Step 3 Select the desired node from the drop down.

Step 4 Click **Assign Selected Node**. The transaction may take a few moments to take effect. Refresh the page to see the change.

If there is a distinguished node rotation schedule set up for this replication group, WANdisco changes the distinguished node according to the schedule, so any change using this **Change Distinguished Node** command is a temporary change.

1.17 Using Subversion Triggers for Sending E-mails

Many administrators like to set up Subversion backend triggers that fire whenever a Subversion user commits a set of file changes. With a single/master Subversion server setup, e-mails can be initiated once when the `post-commit` trigger fires.

However, with the addition of WANdisco replicator, unless some safeguards are put in place, all your Subversion replicas may fire the `post-commit` trigger. This could potentially cause multiple e-mail notifications. Most likely, developers do not want several e-mails for the same transaction.

The easiest way to remedy this is to designate any one node as the “e-mail hub.” Just enable the `post-commit` trigger to fire from a single node within the replication group. Alternatively, you could use the time of day to fire the e-mail alerts from a specific node. For example, you could modify the `post-commit` trigger to send e-mails from India during 9:00 a.m. to 5:00 p.m. IST, and from the US during 9:00 a.m. to 5:00 p.m. PST.

It is allowable to have asymmetry in the e-mail triggers, but make sure not to disable the `pre-commit` trigger on any node. That may cause a Subversion commit transaction to abort at some nodes but commit at other nodes. The `pre-commit` trigger behavior at each node should be deterministic and should not cause the replicas to go out of sync.

When sending e-mail, it is important to set up the e-mail configuration to avoid long blockages or delays. Many times, an administrator uses the default SMTP settings on the Subversion host. These settings by default try to use the organization domain specific e-mail server to send e-mails (by looking up the MX records corresponding to the organization's domain).

The organization-wide SMTP server may be located on a remote WAN, or it may have throttling policies for e-mails originating from the same IP address to cut down on spam. This can cause it to block or reject e-mails, which may in turn cause scripts (like the `post-commit` script) to hang or terminate. To avoid such problems with e-mail triggers, WANdisco recommends that you set up a local e-mail hub or a local SMTP agent/server. The local SMTP server should preferably be on the same host as the Subversion server. It should be set up to forward/relay e-mails to the organization-wide SMTP server. This ensures the e-mail triggers are a lot faster and just need to enqueue the e-mails to the local SMTP server.

1.18 Toggling the Quorum Check

The replicator by default verifies if a network quorum is reachable when a write command is submitted. If the quorum is un-reachable, by default the write command is aborted and the following message appears on the Subversion client console:

```
Check the Network connectivity, failed to reach a
minimum quorum of nodes. Aborting the svn write operation.
```

To turn off the quorum check, set the parameter, `AlwaysVerifyQuorum` to `false` in the `svn-replicator/config/prefs.xml` file. For instance,

```
<SVNProxy>
    <AlwaysVerifyQuorum>false</AlwaysVerifyQuorum>
    ....
</SVNProxy>
```

If the check is turned off and quorum is un-reachable, the write transaction will be applied to the WANdisco Subversion replicator's transaction journal and stay in a pending state till network connectivity and quorum is restored. Note: With singleton quorum, if the current node is also the distinguished node, the quorum check will always succeed irrespective of network connectivity to other nodes.

1.19 Changing WANdisco's admin Login

By default, the login for the Admin Console is `admin`, and the password is user-defined during installation. That way, all nodes initially have the same login and password.

You can change the login for any node. Each node can have its own login, however if you do change it, ensure that all node administrators throughout all nodes know it. If four nodes each had their own logins, then each administrator would see the other nodes in the Dashboard, and could not access the other nodes' Admin Consoles without entering those nodes' passwords.

NOTE:

You can have different logins and passwords for different nodes, however WANdisco recommends you keep administration simple, and have the same login and password for all nodes.

To change the login at a particular node, enter the following in `prefs.xml`.

```
<Security>
   <Admin>
      <user>newlogin</user>
   </Admin>
</Security>
```

See the procedure Changing a prefs.xml File.

1.20 Changing the WANdisco Password

You can change the WANdisco password at any time by using the Change Admin Password command on the Users tab.

NOTE:

This changes the password for this node only. The password does not get replicated.

Ensure that all node administrators throughout the replication group know the new password.

1.21 Resetting or Recovering the Admin Console Password

The password is user-defined during installation. If you change the password from the Admin Console, that change affects only the current site. The password change is not replicated.

Once a site's password is changed, the new password's md5-hash is securely stored under the /config/passwd directory. This directory does not exists initially, and gets created when the password is changed the very first time.

Since WANdisco never stores the actual password, if you loose the changed password, there is no way to recover it. However, you can reset the password using two techniques:

1.21.1 Restore a Password From a Known Site

Step 1 Compress the `/config/passwd` directory at a node where you know the password.

Step 2 Copy and uncompress the file from the previous step into current node's (lost password) `/config` directory. You are over-writing the existing `config/passwd` directory.

1.21.2 Set Password To `wandisco`

Step 1 Delete the `/config/passwd` directory at the current site.

Step 2 Restart WANdisco at the current site.

The password is now `wandisco`.

1.22 Setting Up Hooks

See the following table for how to integrate Subversion hooks with WANdisco.

Hook	How to Integrate with WANdisco
start-commit	Standard Subversion implementation. Must be present at all nodes, and must either execute identically at all nodes or fail identically at all nodes.
pre-commit	This becomes the pre-replication hook. See the following sub-section.
post-commit	Standard Subversion implementation. Must be present at only one node.
pre-revprop-change	Standard Subversion implementation. Must be present at all nodes, and must either execute identically at all nodes or fail identically at all nodes.
post-revprop-change	Standard Subversion implementation. Must be present at only one node.

1.22.1 Pre-Replication Hook

To use pre-replication hooks, make sure you tell WANdisco what version of Subversion you are using. Go to the Proxy tab, select SVN Settings, and specify the Subversion Server Version. See Subversion Server Version in *Using the Admin Console*.

The SVN DAV pre-commit hook becomes the pre-replication hook. Subversion does not execute it: WANdisco invokes this hook before forming a proposal.

Per the SVN DAV specification, if the hook succeeds, nothing is communicated back to the client. The handling of the command proceeds normally.

If the hook fails, per the SVN DAV specification, `stderr` is packaged as an XML response to the client. In response, the client typically deletes the activity; i.e., cleans up the temporary files, etc., on the server side.

1.22.1.1 Configuration

Modify your prefs.xml file at all nodes to contain the following tags. See Changing a prefs.xml File. Below is a sample configuration of a pre-commit pre-replication hook.

NOTE:

> Make sure the hook is not installed in the repository's hooks directory, since you don't want the SVN server to find it.

```
<Hooks>
  <enabled>true</enabled>
  <list>
    <hook name="pre-commit">
      <command>C:/cygwin/home/user/bin/pre-commit.bat</command>
```

```
        <captureExitCode>true</captureExitCode>
      </hook>
    </list>
  </Hooks>
```

1.22.1.2 Other Configuration

Please provide the Subversion server version under the SVNProxy tag. See example below:

```
  <SVNProxy>
    .
    .
    .
    <svnServerVersion>1.5.1</svnServerVersion>
  </SVNProxy>
```

1.22.1.3 Repository-Specific Hooks

If you want different hooks to act on different repositories, you can do so. Create a script (for example, pre-replication.sh or .bat) that contains a case statement that calls each repository-specific hook by passing the repository name as a parameter. Change the prefs.xml to point to that script.

```
<Hooks>
  <enabled>true</enabled>
  <list>
    <hook name="pre-commit">
      <command>C:/cygwin/home/user/bin/pre-replication.bat</command>
      <captureExitCode>true</captureExitCode>
    </hook>
  </list>
</Hooks>
```

1.23 Selective Replication

By default, WANdisco replicates all `SVNROOTS` associated with a Subversion repository. However, you can specify a set of `SVNROOTS` that you do not want to replicate. Use Excludes for `SVNROOTS` in the `prefs.xml` file to identify the repositories you do not wish to have replicated. For example,

```
<SVNProxy>
   ....
      <ExcludeRepositories>/exclude,/dir0</ExcludeRepositories>
   ....
</SVNProxy>
```

If WANdisco cannot find the entry, it is included in the replication. There are no wild (regular) expressions. For Apache and SVNPath, the syntax must match what is listed in the Location

directive. For Apache and SVNParentPath, the syntax must match what is listed in the Location directive and Path to the repository.

All the included roots go through the same agreement manager. For example, say you have three repositories.

```
/repos/rep1
/repos/rep2
/repos/rep3
```

For SVNPath, the Location directive is

```
<Location/rep1>
     SVNPath   /repos/rep1
</Location/rep1>
<Location/rep2>
     SVNPath   /repos/rep2
</Location/rep2>
<Location/rep3>
     SVNPath   /repos/rep3
</Location/rep3>
```

For SVNParentPath, the Location directive is

```
<Location/repository>
     SVNParentPath   /repos
</Location/repository>
```

To exclude the first repository, rep1, for SVNPath, you would exclude `rep1`: for SVNParentPath, you would exclude `/repository/rep1`.

NOTE:

You must name a repository after the `/` character. Subversion does not support a location named with just the `/` character.

1.24 Updating Apache or Subversion in WANdisco Deployment

This procedure requires that your WANdisco deployment, including Subversion servers, are off-line during the time it takes to upgrade at each node. Please plan this procedure accordingly.

Step 1 Ensure WANdisco supports the version you want to upgrade to. Go to Supported Operating Systems and Versions (kayako link).

Step 2 Test the new version of Apache or Subversion in a test environment.

Step 3 Once you are ready to deploy the new version in your production environment, perform a synchronized stop. See Performing a Synchronized Stop.

Step 4 Update each of the nodes.

Step 5 Resume WANdisco.

Step 6 Perform some tests at each node.

The update is complete.

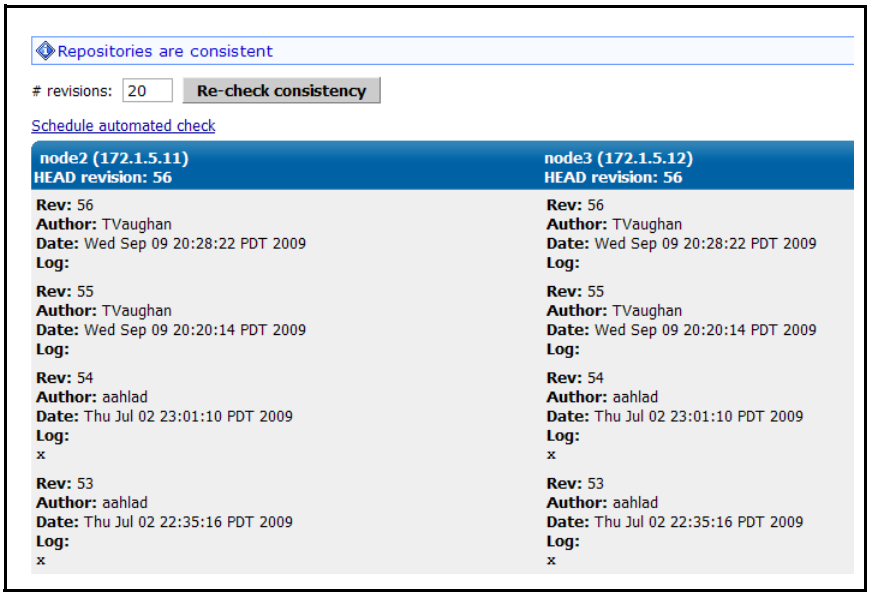
1.25 Checking Repository Consistency

WANdisco provies a command that checks for consistency in a replication group's repository. You can perform this once or set it up to run hourly or daily. WANdisco recommends setting it up to run daily. The results are sent to the administrator via email.

Step 1 Go to the **SVN Settings** command on the Proxy tab.

Step 2 Click Check consistency.

Step 3 Read the consistency report. An partial example follows.



2 Procedures for Stand-Alone Two-Node HA Groups

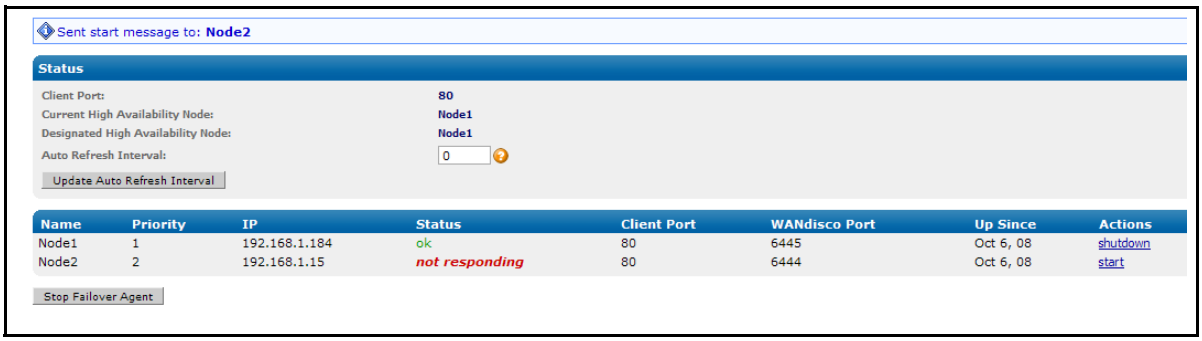
This section is for stand-alone HA sub groups that have just two nodes, and there are no other nodes in the replication group.

2.1 Recovering from Primary Node Failure

When the primary node fails, the Failover Agent sends transactions to the second node. This sets a failover flag. Use the onscreen wizard to clear the flag. WANdisco ensures the failed node gets caught up on missing transactions when it rejoins the replication group.

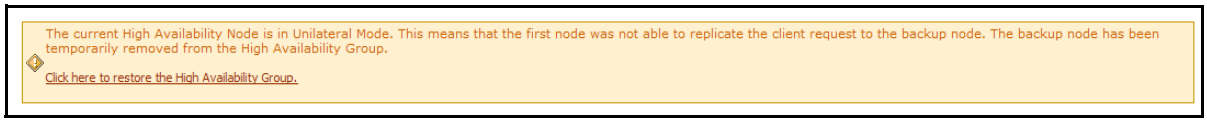
2.2 Recovering from Backup Node Failure

When the backup node fails, there are a few more steps to complete.



Upon an SVN commit, you see this error message about going into Unilateral mode.

Click **Click here to restore the High Availability Group**.

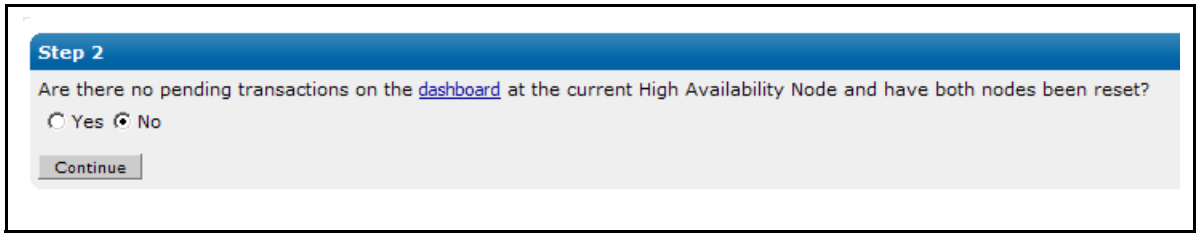


Step 1 Click **Stop Failover Agent**.

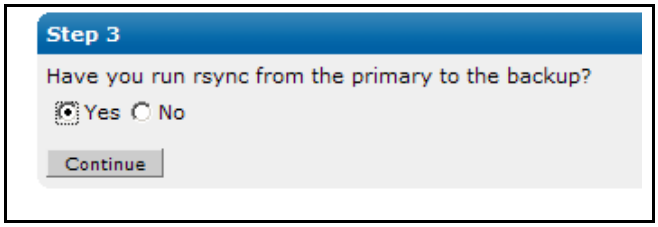
Step 2 Make sure there are no pending transactions on the dashboard, and reset both nodes. Reset the nodes by typing in the `/bin` directory

```
perl reset
```

When you have done this, click **Yes**.



Step 3 Make sure the repositories are in sync. Click **Continue**.



Step 4 Start the nodes. In the /bin directory, type

```
perl svnreplicator
```

Start the Failover Agent. Type

```
perl failoveragent
```

Your HA group is restored.

3 Replicator Management

3.1 Setting Replicator to Start Up on System Boot

To start up on boot, edit the `init.d` scripts. For instance, here is an `/etc/init.d/svnreplicator` script for Gentoo Linux:

```
#!/sbin/runscript
#
# Gentoo Linux dist compatible rc script for
# starting/stoppping svnreplicator
#
# Copyright WANdisco
#

REP_HOME="/home/admin0/svn-replicator"
REP_OPTS="-wdog -email admins@example.com"
export JAVA_HOME="/export/share/apps/jdks/1.5.0"
USER="admin0"

pidfile="my.pid"

depend() {
need net
}


checkconfig() {
if [ ! -f $REP_HOME/bin/svnreplicator ]; then
eerror "No ${REP_HOME}/bin/svnreplicator present"
return 1
fi
prog=$"svnreplicator"
}

start() {
checkconfig || return 1
ebegin "Starting $prog:"
ulimit -S -c 0 >/dev/null 2>&1
ulimit -n 65000 >/dev/null 2>&1
RETVAL=0
start-stop-daemon --start --quiet -u ${USER} --chuid ${USER} --exec
${REP_HOME}/bin/svnreplicator -- ${REP_OPTS}
RETVAL=$?

if [ "$RETVAL" -gt 0 ]; then
eend $RETVAL "Failed to bring up svnreplicator"
return $RETVAL
fi
```

```
eend $RETVAL
}


stop() {
checkconfig || return 1
ebegin "Shutting down $prog:"
su ${USER} -c \"${REP_HOME}/bin/shutdown\" >/dev/null 2>&1
start-stop-daemon --stop --quiet -u ${USER} --pidfile ${REP_HOME}/logs/
${pidfile}
RETVAL=$?
if [ "$RETVAL" -gt 0 ]; then
eend $RETVAL "Failed to shutdown svnreplicator"
return $RETVAL
fi
eend $RETVAL
}
```

3.2 Setting the Replicator Up as a Windows Service

To set the replicator to run as a Windows service, perform the following command at the command prompt:

```
sc create SVN-Replicator binpath= C:\perl\bin\perl.exe –x –S C:\svn-rep-
licator\bin\svnreplicator start= auto
```

Substitute the path for Perl in your environment, and give a different path to the Subversion replicator perl script, depending on where it was installed. You may want to also set `type= share`. The MicroSoft knowledge base article (http://support.microsoft.com/kb/251192) indicates that that is the default, but the `sc.exe` help for create indicates that `type= own` is the default. Note that there is a space between the equals sign, `=`, and the parameter's value.

The Windows Services Control Panel indicates that the service has not started, because our Perl script is currently not exiting because the watchdog is running to restart the replicator. This is actually fine, because the Perl script really takes over.

3.3 Changing the Quorum Type

To change the quorum type, you have to create a new replication group, and specify the new quorum then. See Creating a New Replication Group.

3.4 About Watchdog Mode

By default, WANdisco starts in watchdog mode. Whenever the replicator goes down, the watchdog mode restarts it. In watchdog mode, the replication process automatically disassociates from the terminal and becomes a daemon process, so you should not try running it in the background (with &).

NOTE:

Watchdog mode is not supported in Windows, but it is in Windows Cygwin.

You can turn off watchdog by typing `-nowdog`.

If WANdisco is unable to start up, for example if it terminates several times in quick secession, watchdog starts WANdisco in read-only mode.

```
$ ./bin/svnreplicator -h
Usage: svnreplicator [-v] [-verbose] [-nowdog] [-pause time]
[-email email-address]
```

-v	Print the svnreplicator version
-verbose	Verbose, console messages go to STDOUT/STDERR instead of logs/console.txt
-nowdog	Turn off watchdog mode. WANdisco will not restart automatically if it terminates. Use this option for testing.
-pause	Time in seconds that the watchdog pauses for, before restarting service. Defaults to 0 seconds.
-email	Specify an email address to send an alert to, whenever the Watchdog restarts or shuts down WANdisco. WANdisco generates an email per local replicator activity. If the email settings were not set up during installation, see Email Settings, described in *Using the Admin Console*.

Use the `-email` option to generate email alerts whenever WANdisco restarts. For instance:

```
$ svn-replicator/bin/svnreplicator -pause 5 -email “admin@blueand-
gold.com, scmuser@blueandgold.com”
```

In order to have WANdisco Subversion Replicator automatically started on system reboots, see Setting Replicator to Start Up on System Boot.

3.5 Temporary Files

During the normal course of running WANdisco Subversion Replicator, temporary files are generated. These have the prefix `svn-proposal-{GUID}_{seqnum}`. By default, they are written to `svn-replicator/systemdb/` directory. This can be over-ridden using the `prefs.xml` file as following:

```
<DirPrefixMap>
  <fp->/home/svn/replicator/tmp/dir</fp->
</DirPrefixMap>
```

WANdisco periodically garbage-collects these files at a configurable interval. For more details see the Distributed Agreement Engine Administration Guide.

WARNING:

> Do not manually remove these files.

4 Troubleshooting

4.1 How Do I Get WANdisco Support?

Before opening a ticket or submitting a new issue, always search the Knowledge base on http://www.support.wandisco.com.

If you want to open a ticket, you can do so at that URL.

4.1.1 How Do I Run the Talkback Script?

When you do contact WANdisco with a problem, the first thing WANdisco support asks for is the talkback file. Run the file by typing at `svn-replicator/bin`

```
perl talkback
```

Type in the pathname to SVNROOT when prompted. The output looks like this:

```
Please open a ticket by visiting http://support.wandisco.com and upload
the /talkback-<machine name>.zip, with a description of the issue.

Note: do not email the talkback files, only attach them via the web
ticket user interface.
```

The zip file is located at the root directory. Do not email the .zip file, just attach it to an issue at *http://www.support.wandisco.com*.

4.2 General Subversion MultiSite

4.2.1 Connection Request Timeout Messages

Sometimes in the WANdisco logs, you see connection request timeout information messages logged. These are informational messages and should be ignored unless it is guaranteed that the connection can be established in xxx milli-seconds and happens often.

In normal operation of WANdisco, two connections are established between each of the replicated machines, WANdisco connection and a DFTP connection. These two connections were established when MultiSite started and are used when required. A keep-alive signal is sent on the WANdisco port periodically. There is no traffic on DFTP until a file transfer.

Some lesser routers in the path of the two end points will close an established connection if there is no traffic on the connection without notifying the end points. When end points sent data on this stale connection, they hang forever. To deal with these lesser routers, MultiSite does not keep the DFTP open in its connection pool forever. MultiSite establishes a DFTP connection from receiver to sender when a file transfer was required. This solved the problem dealing with lesser routers.

Some companies have a corporate policy that network connections can only be established in one direction. To deal with this scenario, the replicated machines establish a DFTP connection to other nodes periodically and tear them down if there is no traffic within a known interval. Once a connection is established, any side is free to use the connection regardless of which side initiated the connection. A connection in use is never torn down until it is available as a free connection. This is the current implementation.

It takes between 300 to 400 milli-seconds to establish a network connection even on a slow Wide Area Network (WAN). By default, MultiSite waits for 500 milli-seconds before giving up that a connection cannot be established to a peer machine and prints this informational message. What if the establishments of connection always take 501 milli-seconds. In this case, a connection is never established. To solve this problem, the timeout value is adjusted in 10% increments of the last timeout, starting at 500 milli-seconds, to a maximum of 10 seconds for each timed out connection. Upon establishment of a successful connection, this timeout value is used for subsequent connection establishment unless an adjustment is required for failed attempts.

4.2.2 VPN, NAT, Firewall Timeouts

This section is useful if you are experiencing issues with slow commits on the non-distinguished node or if you have port-forwarding in your environment.

In a multisite configuration, most nodes are connected through a WAN. Often times VPN and NAT devices are used to do IP translation and port forwarding. These devices need to maintain state in order to do the port forwarding on-the-fly. This state can grow if not cleaned out. Many devices simply reset the internal state after an inactivity timeout. For example, some Cisco NAT routers reset state after 7200 seconds or 2 hours.

The WANdisco replicator uses persistent TCP connections between the replicators. If these TCP connection are going through a NAT or port forwarding device, it is important to tune the VPN and/or the TCP stack at the replicator host machine. Many NAT devices have buggy implementation that resets the internal state without resetting the TCP connections.

In such a situation, the replicators may see a connection as established but no communication actually happens. The symptoms include a slow commit that is blocking WAN communication. You can run `netstat -a | grep <DConEPort>` to see if the TCP send queues are backing up. That, in conjunction with slow commits that appear to be hanging, or frozen, typically indicates the NAT is not gracefully resetting TCP connections.

You can further confirm this by using `tcpdump` or `ethereal` to check for excessive retransmissions on the DConENet connections. You could also look at your VPN/NAT device log to see if it reset any DConENet connections that appear to be in an `ESTABLISHED` state via the `netstat -a` command.

These are a few ways of addressing the issue :

- Specify a connection keepalive timeout in the prefs.xml file as:

  ```
  <DConENet> <ConnectionKeepAliveTime>1800000</ConnectionKeepAliveTime>
  ```

  This causes inactive connections to be closed and refreshed periodically (after 1800K millis or 30 minutes).

- Increase the keep-alive timeout on the NAT/Port forwarding device. If possible, have the DConENet connections never expire. Some devices let you set port specific QoS.

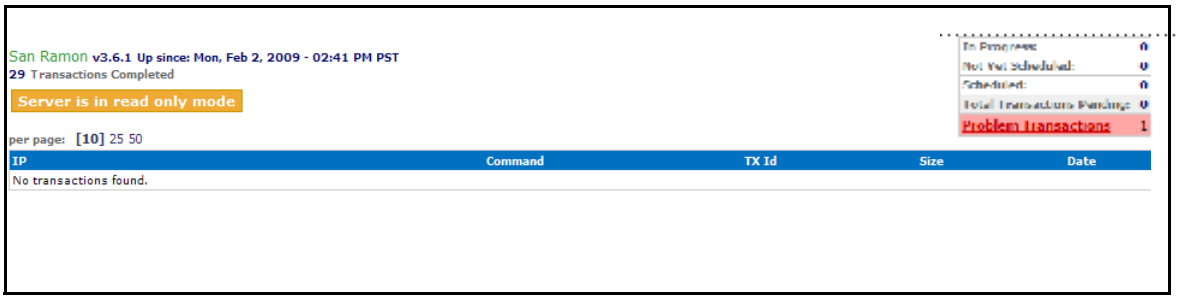
- On the replicator host, tune the TCP stack to have a smaller fuse on the TCP keep alive timer. For example, on Linux, you can specify a value like 1800 (seconds) in `/proc/sys/net/ipv4/tcp_keepalive_time` to reduce the interval from default 2 hours to 30 minutes.

- You can restart the Subversion replicator to kick out seemingly `established` connections but broken by NAT internal resets.

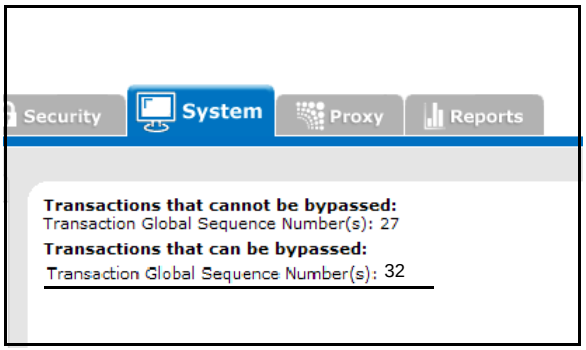
4.2.3 A node Is In Read-Only Mode

If any nodes are in read-only mode,

Step 1 Go to the Dashboard. You see the Problem Transactions listed for each node.



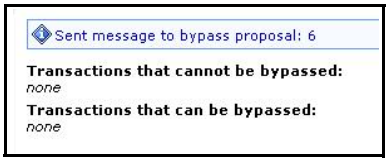
Step 2 Click on the Problem Transactions link. The System tab displays a message, noting the bypassable transactions and the non-bypassable transactions.



Step 3 If there are no bypassable transactions, and you do not know why, see How Do I Get WANdisco Support?

If there are bypassable transactions, there may be one or several. Click on each transaction.

Step 4 Read the message that returns, similar to:



Step 5 Verify that the Dashboard does not report any node in read-only mode.

4.3 Error Messages

4.3.1 Missing License Key File

Subversion MultiSite depends on a license key file being present in the `svn-replicator/config` directory for each node. Please get a valid license from WANdisco and copy the file to the config directory. WANdisco does not start without the license file.

4.3.2 I'm Getting a SEVERE Exception

I'm getting a SEVERE exception, and replicator is aborting the Subversion transaction and shutting down.

If you get a message in the logs/SVNProxy*.log file similar to

```
svn: Commit failed (details follow):
svn: File not found: transaction '10-d', path '/development/Hello.txt'
```

it means the replicator has detected an out of sync condition. Remember the replicator continuously monitors your repository for any out of sync issues. If it detects this has occurred, it triggers an automatic shutdown to prevent further corruption.

This could happen if some one accidently committed directly to Subversion, bypassing the replicator, and ramped up the version in one node without giving the replicator any chance of replicating. This can be easily resolved by following the reset procedure outlined in I Directly Committed to Subversion, How Do I Rsync?.

Follow all precaution to avoid bypassing the replicator:

Step 1 Ensure only svnreplicator host/IP address is allowed to connect to the Subversion server.

Step 2 Protect direct logins in Subversion replicator or Subversion server box from end user.

4.3.3 Compressed Stream Invalid

If you get an error message similar to

```
nchang:tools$ svn commit
> svn: Commit failed (details follow):
> svn: At least one property change failed; repository is unchanged
> svn: PROPPATCH of
```

```
> '/repo/!svn/wbl/2d0e7cef-62f8-40a0-bcbd-a3a03a8c59ae/10809':
> Compressed stream invalid (http://100.143.136.110)
```

Verify that you are not using the Apache mod_deflate option, which is not supported.

4.4 Oops!

4.4.1 I Directly Committed to Subversion, How Do I Rsync?

If you bypassed the replicator, you can reset the replicator state with these steps:

Step 1 Shut down all replicators.

Step 2 Reset each replicator: at `svn-replicator/bin`, type

```
perl reset
```

Step 3 If this happened on a production repository, you **must re-sync** all the repositories to the same state/data.

If this happened during an initial setup/evaluation stage, delete the old project in Subversion and create a new one.

Step 4 Restart all the replicators.

NOTE:

It is very important that you take all precautions to avoid directly checking in or committing to the backend Subversion repository.

4.4.2 I Pressed Ctrl-C During a Subversion Command!

If you were executing a read command (a command that does not modify the Subversion repository), you do not have to do anything.

If you were executing a write command, update your sandbox after the replicator has applied the command to the repository.

In addition, if you were adding files to the repository (either svn import or svn add, followed by svn commit), wait until you update your sandbox before you continue to use it.

5 Frequently Asked Questions

5.1 Why Are So Many Java Processes Running?

On older versions of Linux, every thread is listed as a process by the `ps` command. This does not affect the operation of MultiSite. WANdisco does not support the older versions of Linux.

5.2 Can I Store Logs or Content on NFS?

NFS (Network File System) allows files and directories to be accessed remotely over a network using NFS clients. NFS clients are typically built into the operation system kernel these days. However, some operations, like renaming a file, are not guaranteed to be atomic over NFS. Here is a snippet from the rename function's `man` page on Linux, for example:

```
BUGS
On NFS filesystems, you can not assume that if the operation failed the
file was not renamed. If the server does the rename operation and then
crashes, the retransmitted RPC which will be processed when the server is
up again causes a failure. The application is expected to deal with this.
See link(2) for a similar problem.
```

Code management systems such as Subversion make heavy use of the `rename` operation to modify the underlying databases. Independent of WANdisco, it is a risky practice to store Subversion database content on NFS. The code management community at large recommends not using NFS for storing repositories.

WANdisco MultiSite is bundled with a built-in transactional journal and an object database. These are by default stored in the `svn-replicator/systemdb` and `svn-replicator/config` directories. These directories should not be mounted on an NFS drive. The replicator itself may be installed on an NFS drive but the `systemdb` and `config` directories should be on direct storage (non-NFS options like RAID, SCSI, SAN, etc). Replicator's transactional integrity can be compromised if writes to an NFS server are lost due to a potential NFS client cache crash after the NFS server has indicated IO completion.

5.3 Why is Installer Configuring IP Addresses as 0.0.0.0?

The address 0.0.0.0 is a special IP address, treated as a wild-card IP address. In other words, on a machine with multiple NICs (Network Interface Cards), it binds to all interfaces. The advantages of using wild-card IP address include:

- It avoids binding to a fixed IP address. If the host's IP address changes, (for example, the subnet changes, or the machine is moved to a different location) you don't have to change the wild-card IP in the prefs.xml file to the new IP address.

- There is wider bandwidth to TCP clients. Now TCP clients can connect to any NIC, because MultiSite is listening on multiple NICs.

The disadvantage to using the wild-card IP is that it gives coarser access control at the IP address level, as all address are being listened to at the specified port.

You can always switch from the wildcard IP address to a fixed, static IP address or a DNS host-name, though for the most part, WANdisco recommends you stick with wild-card addressing.

5.4 Should I Worry About Time Changes or Time Zones?

Time changes have no effect on the operation of MultiSite. Times zones also have no effect: all machines use the standard UTC.

5.5 Does WANdisco Support Dynamic DNS?

Yes, WANdisco supports dynamic DNS, but strongly discourages its use.

If a hostname is specified during the setup process, WANdisco requires that it should be able to connect to a valid DNS and resolve the hostname to valid IP address upon startup. If the host-name cannot be resolved to an IP (either by not being able to connect to DNS, or no entry is found at the given hostname), WANdisco dies gracefully. This has never been a problem during production and with static IPs.

However, if dynamic DNS support is required, please modify the `prefs.xml` file at each node and set `UseDynamicDNS` to `true` in `DConENet` element.

```
<Preferences>
  ...
  <DConENet>
     ...
     <UseDynamicDNS>true</UseDynamicDNS>
  </DConENet>
```

In addition, the following Java security properties should be set to different Time-to-live (TTL).

```
networkaddress.cache.ttl
networkaddress.cache.negative.ttl
```

Please read InetAddress Caching for more details.

5.6 Can I Use SSH Tunnel to Navigate a Firewall?

You can use SSH tunnels to test connectivity to a replicator's port through a firewall.

NOTE:

> SSH tunnels are not recommended for a production environment.

SSH tunnels are temporarily created using a secure shell. If the shell hangs up for any reason, the tunnel goes away. You don't want the connectivity to a replicator's WAN port to be dependent on a transient shell. We recommend using permanent IPsec tunnels (VPN/NAT devices can help) for navigating firewalls.

5.7 WANdisco Authentication

Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. Authorization is the process of giving someone permission to do or have something.

The Apache user-names and passwords should match at all nodes. The WANdisco Subversion replicator's license manager requires a valid user-name inside the HTTP authorization header to be passed for all DAV commands, except `OPTIONS` and `PROPFIND`. In other words, anonymous access to Apache is not allowed to enforce license requirements, unless you have an unlimited or an evaluation license. With an unlimited or evaluation license, you are not required to register the user. This typically means ensuring a `Require` valid-user line is specified in the Apache SVN DAV configuration files in the `/etc/httpd/conf/httpd.conf` and `/etc/httpd/conf/conf.d/*` directories. When using Basic Authentication, it is the end user or administrator's responsibility to keep Apache authentication databases in sync across all nodes.

5.8 Encryption Around WANdisco Protocol

Details about any ECCN classifications you may have applied for and been granted from US Government for export (due to encryption capabilities in client for DAV over SSL).

The WANdisco Subversion MultiSite distribution does not actually perform any encryption or decryption of the DAV traffic. We rely on Apache to decrypt the SSL traffic and then use a proxy-pass definition within the Apache configuration to redirect the un-encrypted request to the WANdisco replicator.

Communication between the WANdisco replicators running at each node/replica does not get encrypted directly by the WANdisco replicator either. Instead, many customers may use something like a persistent VPN connection to communicate the replicator to replicator traffic over an encrypted connection, but our code actually is doing no encryption.

Lastly, the WANdisco replicator simply sits as a proxy on the SVN server itself (the host running Apache + mod_dav) so there is no client component that we provide that would be sending any traffic to the SVN server.

5.9 How Do I Restrict Direct Access to My Repository?

If you would like to prevent users from directly accessing your Subversion repository, use the Subversion `Location` directive as suggested below. You allow only specific IP addresses to access the repository.

This assumes that WANdisco and Apache server are running on the same machine.

From the example shown in Encryption Around WANdisco Protocol:

```
<Location /svnrepos>
AllowOverride None
Order allow,deny
Allow from 127.0.0.1
DAV svn
SVNParentPath /tmp/dav
AuthType Basic
AuthName wandisco
AuthUserFile /etc/httpd/conf/htpasswd
Require valid-user
</Location>
```

5.10 About WANdisco Log Files

WANdisco uses Java logging. See http://java.sun.com/j2se/1.4.2/docs/guide/util/logging/overview.html for a discussion on Java logging. Make any changes to the `svn-replicator/config/log.properties` file.

WANdisco places the log files in `svn-replicator/logs`. The current file is always `SVNProxyServer-prefs.log.0`, and the files are rotated out and eventually garbage collected. For rotation schedule, see the `svn-replicator/config/log.properties` file.

Here is a brief explanation of Java logging. The newest log is always `log.0`. When that log reaches a specified size (500 KB by default), that log gets renamed to `log.1` and a new `log.0` is started. The old `log.1` becomes `log.2`, `log.2` becomes `log.3` and so on. The second newest log is always `log.1`. WANdisco's Log Viewer displays a drop-down list showing the other logs. As the log file names increment higher, they represent going further back in time.

If you want to change the defaults on the file size before log rotation and how many logs to keep, change these parameters in `svn-replicator/config/log.properties`:

```
java.util.logging.FileHandler.limit = 500000
java.util.logging.FileHandler.count = 500
```

The limit is in bytes and the count is the maximum number of logs to keep. Any changes to `log.properties` are unique to each node, and are not replicated.

5.11 How Do I Deal with Failover Agent Failure?

If the failover agent fails, the watchdog script immediately restarts it. If the machine crashes, service is unavailable until the machine is rebooted and the failover agent is restarted.

You could also run the failover agent on a hardware cluster. The Veritas Cluster Server is an example of a commercial solution. See http://www.symantec.com/business/products/overview.jsp?pcid=pcat_business_cont&pvid=20_1.

Linux-HA is an example of an open-source solution. See http://www.linux-ha.org/.