WANdisco Subversion MultiSite Administration Guide

Revision History

REVISION	DATE
1.0	December 2008
2.0	February 2009
3.0	July 2009

Contents

Contents, cont'd

Contents, cont'd

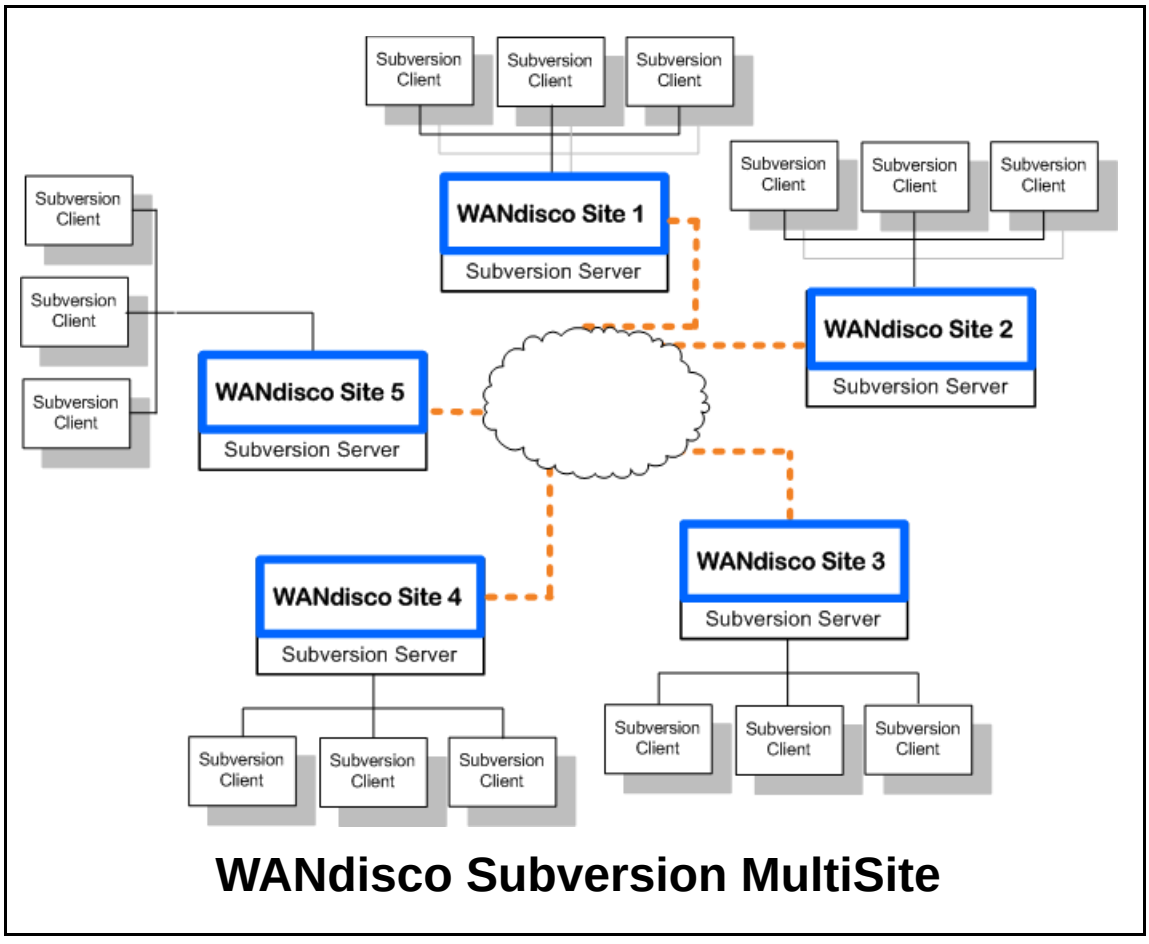
Contents, cont'd

Contents, cont'd

1 Introduction

Welcome to the WANdisco world of replication. Subversion is designed to run as a central server to which multiple Subversion clients connect. WANdisco's replication technology makes it possible to have multiple active replicas of a Subversion repository that are in synch. The Subversion replicas can be anywhere on a WAN - distributed throughout a company's campus or throughout the world. WANdisco users experience the performance of a local Subversion repository, with the semantics of a single shared Subversion repository. We call this "active replication with one-copy-equivalence."

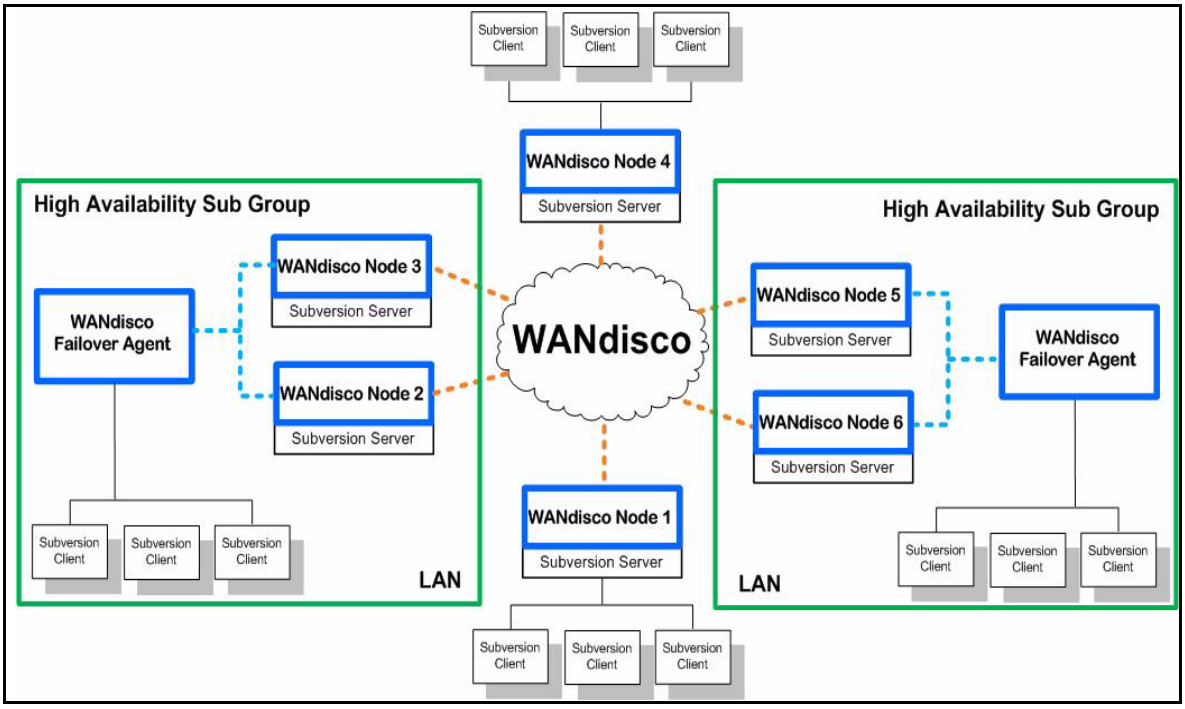
Replication implicitly ensures that each replica acts as a hot backup to every other replica. If a local server does experience a problem that takes it offline, local users experience a disruption in service, while the rest of the replication group continues unaffected. The following illustration shows a replication group with five Subversion severs.



WANdisco Subversion MultiSite

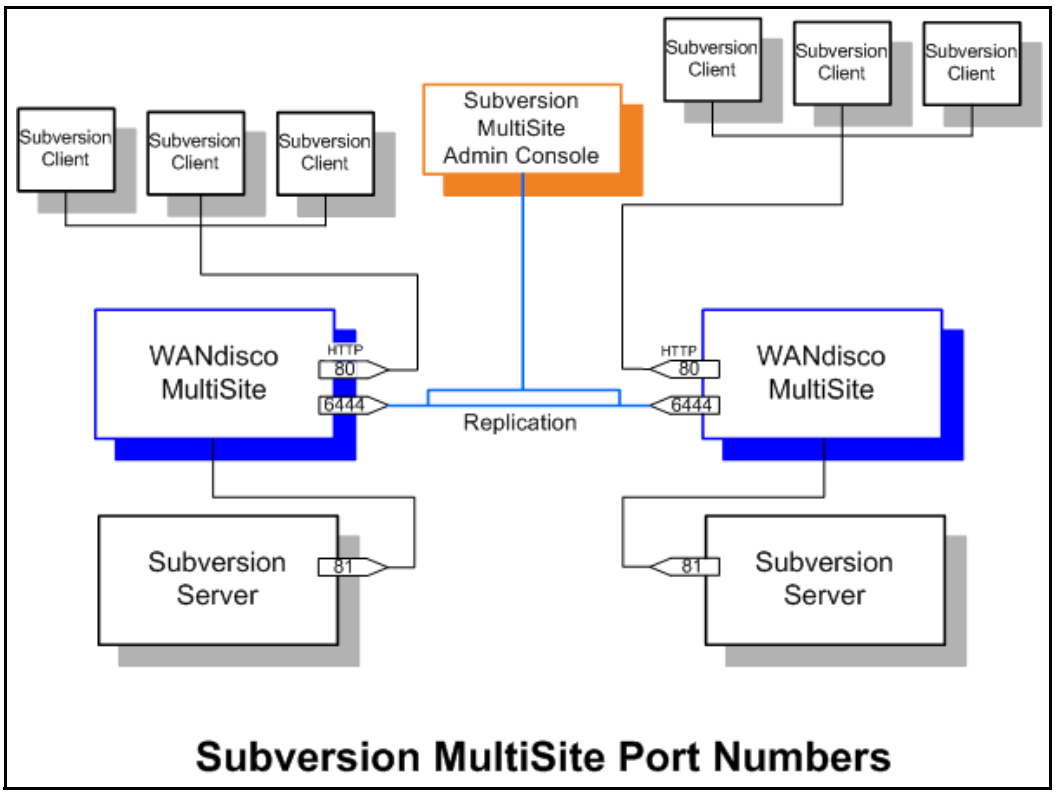
WANdisco offers a High Availability solution that ensures no disruption in service. Even if a local Subversion server failed, a local backup takes over so service is uninterrupted. High Availability sub groups reside on a LAN, which can either be implemented stand-alone for a local Subversion server, or as part of the WAN-based MultiSite.

The following illustration shows a MultiSite group of six nodes at four locations, with two High Availability sub groups of two nodes. Each High Availability sub group contains a Failover Agent, a stateless member of the replication group membership.



WANdisco has the Admin Console, a web-based user interface, to administer and monitor the replication group.

WANdisco functions as a proxy to the Subversion server. An instance of the proxy operates at each replica. All the communication paths involved in the operation of WANdisco are illustrated in the diagram below.



1.1 WANdisco MultiSite Concepts

All MultiSite sites are synchronized at all times: each Subversion repository is a functional replica of the others. WANdisco replication technology is the concept of one repository, multiplied. Because there are multiple synchronized repositories, each replicated node is effectively a current hot backup, which makes disaster recovery easy to plan and implement.

The Subversion usernames and passwords on all repository hosts must match. This is required because WANdisco creates a peer-to-peer replication system. Any replica of the Subversion repository is accessible by every valid Subversion user. WANdisco offers the user the option of having WANdisco manage the Subversion password file.

1.1.1 How Replication Works

The sites in the replication group are continuously coordinating the Subversion write transactions users are making. The group establishes transaction ordering through the agreement of a quorum of replicas. When you install the first node, that node by default is the distinguished node with Singleton quorum. When you create the replication group, or membership, that includes other nodes, you select the quorum type best suited to your configuration. For more information on quorum types for various configurations, see 2.2.2, Quorum Recommendations.

1.1.1.1 Singleton Quorum

Singleton Response quorum means that only one of the nodes in the membership decides on the transaction order. With Singleton Response quorum, the node that decides transaction ordering is called the distinguished node. The Singleton quorum offers the fastest response time for those users working at the distinguished node, because as soon as the distinguished node determines that a transaction can be processed in the correct order, WANdisco sends that transaction to Subversion. Any replicator except the distinguished node can go down, but the replication group continues. The replication group replays the missing transactions when that node rejoins the group.

However, the Singleton quorum also represents a single point of failure, since replication halts if the distinguished node fails.

You can schedule the distinguished node to rotate to different sites around the world, to “follow the sun.” Rotating the distinguished node allows other sites to take advantage of the quickest response time when they are most active. See 8.3, Rotating the Distinguished Node.

1.1.1.2 Majority Quorum

Majority Response is another quorum option, whereby you specify that a majority of the sites must agree on transaction order before any transaction is committed.

Having a majority quorum ensures that if one site goes down in a replication group, even the distinguished node, the other sites can continue uninterrupted, as long as a majority of the sites remain available. The replication group replays the missing transactions when that site rejoins the group.

In a majority quorum, the distinguished node’s role is that of a tie-breaker. For example, in a four node replication group, three sites make the quorum (three sites must agree about transaction ordering). If two nodes want one transaction first, and the other two want another transaction first, then the distinguished node gets a weighted vote. The group with the distinguished node determines the transaction ordering.

With an even number of nodes with majority quorum, you can schedule the distinguished node to rotate to different nodes around the world, to “follow the sun.” See 8.3, Rotating the Distinguished Node.

1.1.1.3 Unanimous Quorum

The last quorum option is unanimous response, which requires that all replicators must be reachable to accomplish transaction ordering.

1.1.2 Replication Example

Here is an overview of what occurs when a write transaction is received by any replicator in the replication group.

Step 1 The originating client sends the transaction to Subversion MultiSite, which passes it along through the replication group.

Step 2 Transaction data is successfully received by the quorum (the distinguished node for Singleton quorum, or a majority of sites for Majority quorum). The quorum assigns the transaction a Global Sequence Number (GSN).

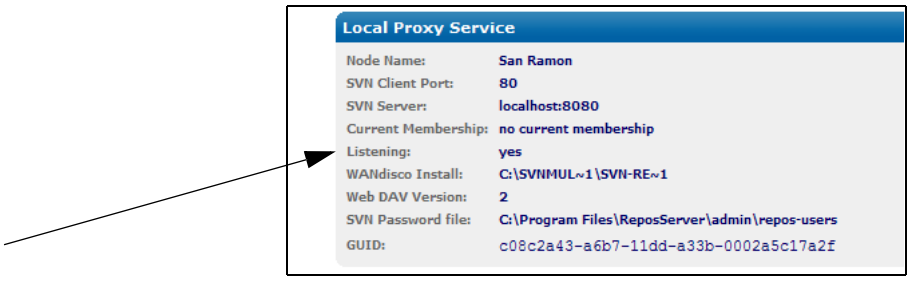
Step 3 After receiving the transaction, each Subversion MultiSite passes the transaction data to its local Subversion server.

Step 4 Each local Subversion server processes the transaction.

Step 5 Subversion MultiSite waits for Subversion to complete the transaction. Subversion MultiSite only marks the transaction complete when Subversion returns a completion status. If for some reason replication goes down during this process (the replicator crashes, is stopped by an admin or the server it's running on shuts down), Subversion MultiSite does not mark the transaction as complete, and it gets reprocessed upon replication restart.

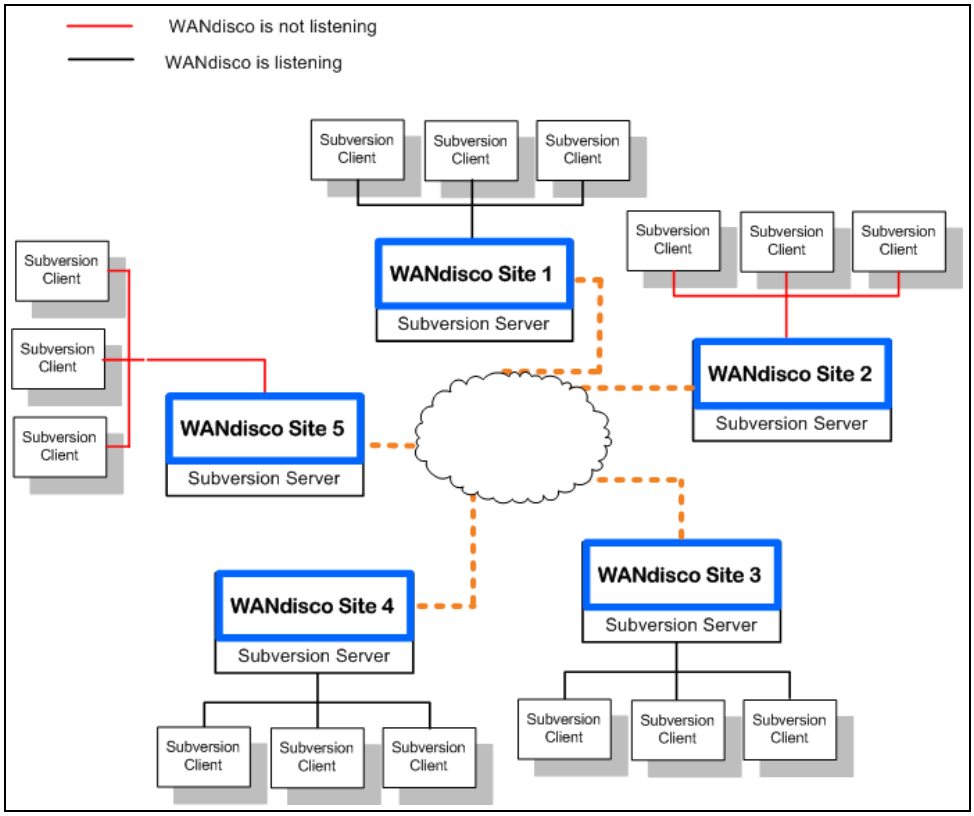
1.1.3 WANdisco is Listening

There is a field in the Admin Console that tells you whether WANdisco is accepting any incoming Subversion client requests. Replication still continues among the WANdisco sites, whether WANdisco is listening or not at one or more sites.



You can turn the listening on and off through the Admin Console (through the **Start Proxy** and **Stop Proxy** commands). The Admin Console is described in Chapter 4, Using the Admin Console. Issuing the **Stop Proxy** command on a site puts that Subversion server in read-only mode.

The following illustration shows Sites 2 and 5 are not listening. (An administrator executed the **Stop Proxy** command for those sites.) Replication continues, and Sites 2 and 5 are still receiving and processing replicated transactions originating from the other sites. However, Subversion users at Sites 2 and 5 cannot make any write transactions. Once an administrator issues the **Start Proxy** command for Sites 2 and 5, the local Subversion users can again issue Subversion commands.



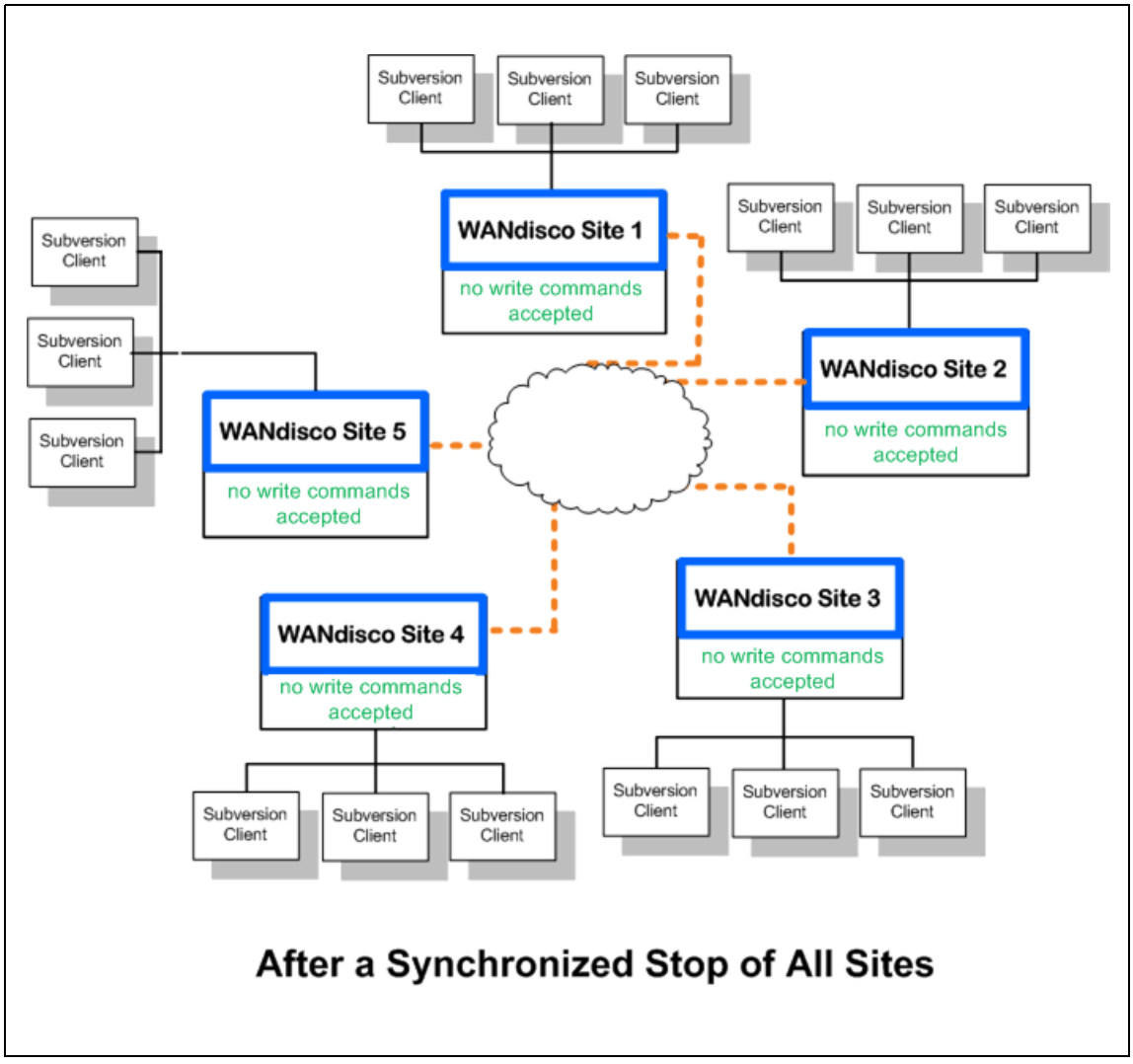
For High Availability sub groups, shutting down the Failover Agent stops WANdisco from accepting local client requests.

Please follow your company guidelines in regards to notifying Subversion users of maintenance.

1.1.4 Synchronized Stop of All Sites

When an administrator issues a **synchronized stop** command, the Subversion servers stop accepting write commands from clients. Pending transactions are processed, but no new write transactions are accepted. Subversion users continue to have read access to the repository, but cannot perform write operations, such as **commit** or **lock**.

When an administrator issues a **resume** command, the WANdisco proxies restart and begin accepting write transactions.



After a Synchronized Stop of All Sites

1.2 About the High Availability Option

High Availability offers WANdisco customers business continuity, because if one Subversion server fails, the Failover Agent transparently directs users to the next available replica in the High Availability sub group. Each WANdisco Subversion High Availability sub group consists of a Failover Agent and two or more Subversion servers, each with an associated WANdisco proxy, on a LAN.

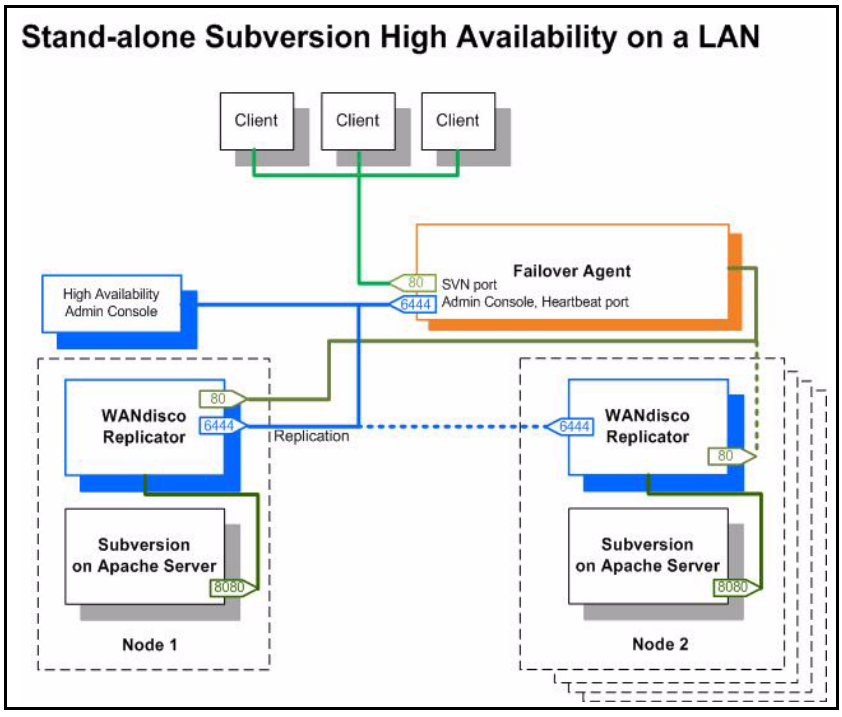
You can add one or more High Availability sub group to a MultiSite replication group, or you can have a stand-alone High Availability group at a single site. Each option is discussed further on in this section.

NOTE:

For a stand-alone High Availability deployment, WANdisco strongly discourages using High Availability with just two nodes in a replication group. Deploying a three node replication group automatically handles the failure of a node and its subsequent recovery.

With a stand-alone HA two-node replication group, some failure scenarios require manual intervention to complete recovery.

All communication paths involved with High Availability are shown below.



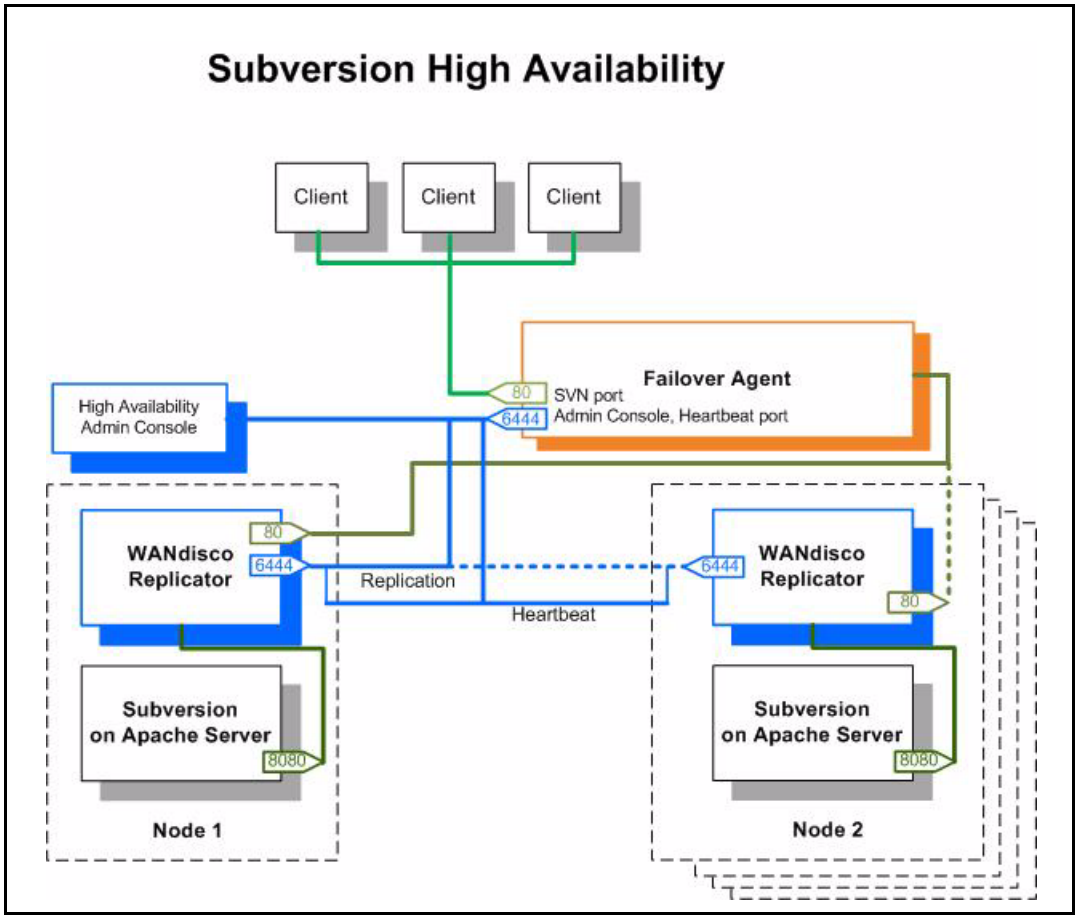
The Failover Agent is a stateless node, and is displayed in the Admin Console and configured with other nodes during installation. It stays outside actual replication communication, and does not participate in the quorum or any decisions regarding replication.

1.2.1 Failover and the Heartbeat

The Failover Agent continuously sends out a heartbeat, or an “are you alive?” message, to each node in its High Availability sub group on the WANdisco port using the DConENet protocol. By default the heartbeat messages are sent at intervals of one second. The High Availability nodes respond to this with an “I am alive” message.

The Failover Agent expects a response from each of its High Availability nodes. If the Failover Agent does not get a response within a configurable number of heartbeats, the Failover Agent marks that node as unresponsive. WANdisco administrators can change the heartbeat interval and missing heartbeat count in the Admin Console.

If the Failover Agent has marked node 1 (the current node) as unresponsive, actual failover to the next node occurs lazily, that is, only when a Subversion client request comes through. This reduces the number of false failover alarms, as a replicator may not respond within the configurable number of heartbeats during a restart, or a node may be restarted before the next client request, eliminating the need for failover.



1.2.2 Failover Sequence

The Failover Agent passes Subversion transaction data to only one node in the sub group. That node then replicates the transactions to the entire replication membership, whether the nodes are within its High Availability sub group or are distributed throughout a MultiSite membership.

1.2.2.1 The Current Node and the Designated Node

The node that communicates with the Failover Agent is called the “current” node. During normal operation, the current node is also the “designated” node, the node the Failover Agent expects to talk to. If the designated node should fail, and failover occurs, the current node would become node two, however node one is still the designated node. When node one comes back online, the Failover Agent resumes talking to node one (which again becomes the current node as well as the designated node).

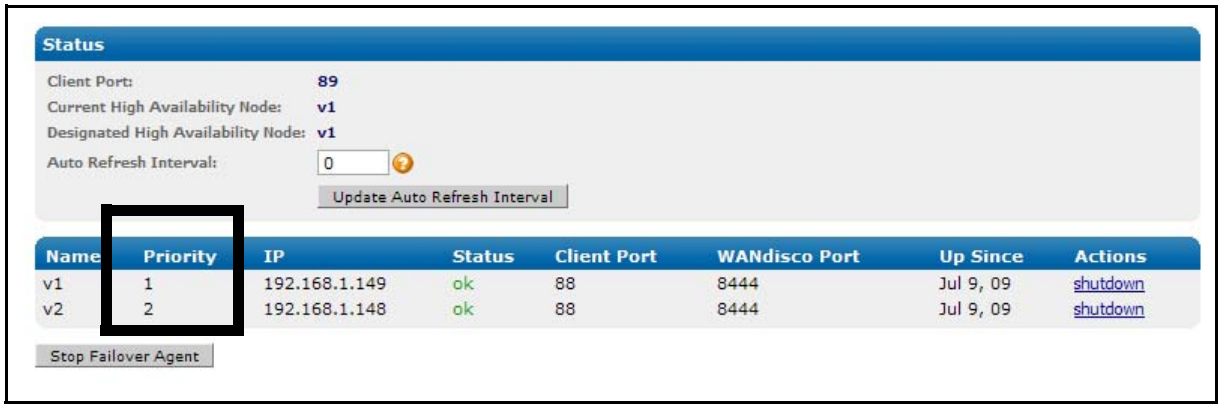
1.2.2.2 Priority Order

During installation, you determine the failover order for a sub group. For example, node 1 is first in the failover order (priority one), node 2 is the second in the failover order (priority two), and so on.

The first HA node, as the current node, receives all Subversion transactions through the Failover Agent. If that node fails, the Failover Agent begins communicating directly with node two (upon receiving a client request). If node two fails, then the Failover Agent begins communicating with node three, and so on. The order is referred to as the priority order, meaning the order the Failover Agent takes during a failover.

If node one comes back up, the Failover Agent returns to communicating with that node.

Set during installation, priority order is always visible on the Failover Agent Status tab.



You can also see the failover priority order by looking at the value for the `PriorityOrder` element in the prefs.xml file (shown in **bold** in the following example). In the example, you see that this node is the current node, the first node in the priority order.

```
...
<ServerProxy>
   <ListenerIP>192.168.1.184</ListenerIP>
   <ListenerPort>2401</ListenerPort>
   <PriorityOrder>1</PriorityOrder>
</ServerProxy>
...
```

1.2.3 About High Availability within the Replication Group

You can integrate High Availability sub groups within a MultiSite replication group. Each High Availability sub group exists on a LAN and has its own Failover Agent.

All sites are considered standard MultiSite nodes, communicating with each other for replication. The nodes that are in an HA sub group exist on the same LAN, and communicate with the Failover Agent, also on the LAN, as well as all the other nodes.

The Failover Agent sends local client transactions to one designated HA node. That designated HA node then communicates that transaction throughout the replication group. If the designated node goes down, the Failover Agent communicates with the next HA node in the sub group priority order. That HA node continues communicating with the larger replication group, so there is no disruption of service, either for local Subversion users or for the replication group.

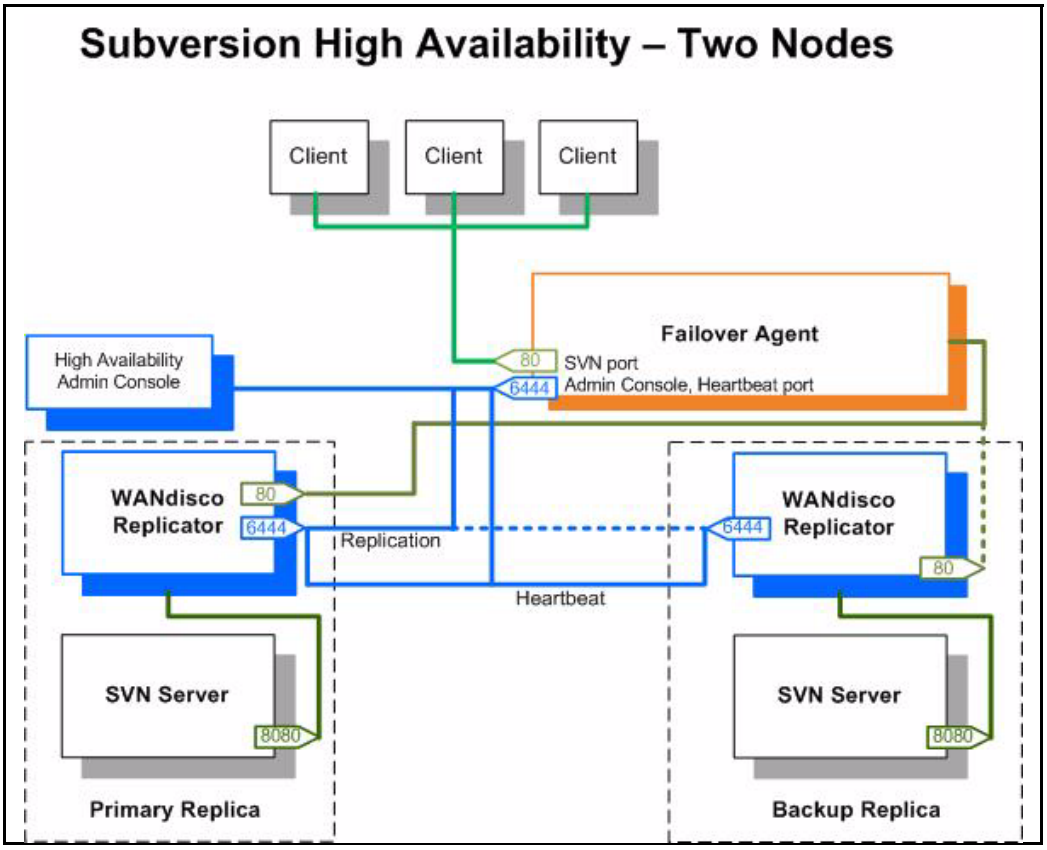
1.2.4 About the Stand-alone High Availability Group

You can have a stand-alone High Availability group on a LAN. If desired, you can add additional sites distributed either on the LAN or a WAN at a later time.

1.2.5 Stand-Alone Failover Group of Two Replicators

As noted previously, WANdisco strongly discourages using stand-alone High Availability with just two nodes in a sub group. To automatically handle the failure of any single replicator node, and its subsequent recovery, a minimum of three nodes are required in a sub group. As documented in 7, Procedures for Stand-Alone Two-Node HA Groups, in a two-node deployment, some failure scenarios require manual intervention by the administrator.

With a stand-alone two-node High Availability group, the second node defaults to be the distinguished node.



1.2.5.1 What Happens If the First (Designated) Node Fails

If the designated node fails, the Failover Agent engages the second node, setting the “failed to backup” flag. Users see no interruption. In order to have the designated node rejoin the HA group once it is back up, you use a wizard in the Admin Console.

1.2.5.2 What Happens If the Second Node Fails

If the second node fails, users see no interruption, since Subversions transactions are engaging the first node anyway. The Admin Console displays current status. Since the second node is the distinguished node (a requirement for the stand-alone two-node scenario), its failure forces the first node to go into unilateral mode, setting the unilateral flag. Returning the distinguished node to the membership once it is back up is described in 7.2, Recovering from Backup Node Failure.

1.3 Replication and Site or Network Failures

MultiSite supports either Singleton, Majority or Unanimous Response quorum. Behavior during failures depends on quorum type.

Note that these scenarios exclude the nodes in an HA sub group. Failures within an HA sub group was discussed in 1.2.2, Failover Sequence and 1.2.5, Stand-Alone Failover Group of Two Replicators.

1.3.1 Site Failures

For Singleton Response quorum: say you have a five node replication group, spread across three continents. One of the sites goes down. Replication continues at the remaining sites, as long as the quorum is reached, although users connecting to the downed site are read-only until that site can reach the quorum.

As soon as a node comes back up, the replication group catches up the node on its missing transactions, so that all nodes are again synchronized.

For Majority Response quorum: say you have a five node replication group. One or two sites could go down, and replication would continue at the other sites, as long as a majority of sites remain up. The one or two downed sites go into read-only mode. As soon as a node comes back up, the replication group catches up the node on its missing transactions, so that all nodes are again synchronized.

1.3.2 Network Failures

If a network link goes down for one node, and outside connectivity is completely lost, there are two possible scenarios, depending on your quorum:

- If you have Singleton quorum, and the distinguished node's network link goes down, the distinguished node alone can make progress. The Subversion users local to the distinguished node continue working uninterrupted, while users at other sites in the replication group can make only read operations (like up, co, log, etc.) working with stale data.
- If you have Majority quorum, and one site's network link is lost, then users at that site can execute only read operations (like up, co, log, etc.) working with stale data. Providing that the remaining sites can still meet quorum (having a majority of sites responding), the other sites continue working uninterrupted.

When connectivity is restored or the errored node is back online, the local node syncs up with the replication group automatically. First, the local node consults its local recovery journal (similar to a database redo log), and then, if necessary, attempts recovery from any of the quorum sites.

The recovery infrastructure and details of WANdisco fault-tolerance can be found at http://www.wandisco.com/pdf/dcone-whitepaper.pdf.

1.3.3 Failover Agent Failures

A watchdog monitors the Failover Agent, so if the Failover Agent crashes, the watchdog immediately restarts it. If the machine should crash, service to the HA sub group is unavailable until you reboot and restart the Failover Agent.

If you do not want to wait for the Failover Agent machine to be restored, you could 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/.

1.4 Establishing a Replication Baseline

When you deploy Subversion MultiSite, you must ensure that all the repositories start out in sync, meaning that all of them are identical. Once Subversion MultiSite is deployed, WANdisco's replication technology ensures they remain in sync.

You start with one Subversion repository, referred to as a replication baseline. To create this baseline, follow the procedure in 6.2, Establishing a Baseline for Replication.

1.5 Terms

You should familiarize yourself with these terms.

TERM	DEFINITION
replica	a Subversion instance that is an exact equivalent or copy of another Subversion instance. In WANdisco's MultiSite product, a replica is also called a node.
replicator	The intermediary that acts as an application proxy/gateway between Subversion clients and a given Subversion server. Each *Replica* has an associated *Replicator*. It coordinates with other peer replicators to ensure that all replicas of the SCM repositories stay in sync with each other.
replication group	a collection of replicators that work together to keep replicas Subversion repository in sync.

TERM	DEFINITION
one copy equivalence	all replicas are functionally equivalent copies of each other
GUID	Globally Unique Identifier. WANdisco Subversion MultiSite assigns each node a GUID on installation. The nodes identify each other by their GUIDs.
site	a server on which is installed a replicator and a replica. The sites comprise the replication group.
distinguished node	The distinguished node acts as a tie breaker for a majority quorum when there are an even number of nodes, making the final decision on replicator operations.
Quorum	A set of nodes that can reach agreement to determine transaction order. In the case of an even number of nodes, the distinguished node settles a conflict. Quorum is defined in the prefs.xml file. MultiSite by default has Singleton Response quorum.
prefs.xml	The preferences files contain information on the replication group. Each site contains all preference files for the entire replication group. The files are specific to each site. The preference files are located in `svn-replicator/config`.
SCM Repository	Software Configuration Management repository like Subversion
SCM Server	A network server that provides remote access to an SCM Repository
installDir	this is the installation directory for WANdisco MultiSite
DConE	WANdisco's Distributed Coordinated Engine, the software engine underlying replication
Install node	The node where you run the Subversion MultiSite install program. This is the first site in your replication group.
Unique to the High Availability Option	
Failover Agent	It is the intermediary that acts as an application proxy/gateway between the SVN client and the High Availability replicators. The Failover Agent keeps track of which High Availability replicas are available, and issues the SVN client's request to one of them.
current node	The High Availability node the Failover Agent is communicating with within the HA group.
designated node	The High Availability node with priority order 1 (as listed in the prefs.xml file). This is the High Availability node the Failover Agent expects to communicate with.
heartbeat	mechanism the Failover Agent uses to monitor availability of all HA nodes
heartbeat interval	the interval, in seconds, the Failover Agent waits to send an "are you alive?" message to all HA nodes

TERM	DEFINITION
heartbeat connection timeout	The time the Failover Agent waits before assuming the non-responding node is unavailable. If the current active primary is unavailable, this triggers failover, and the Failover Agent begins communicating with the next node in the priority order.

2 Recommended Deployment Practices

2.1 MultiSite Administrator Pre-requisites

This guide is intended for an Subversion administrator or a user who is reasonably comfortable with:

- Setting up a Subversion based repository
- Configuring inetd/xinetd service on Unix/Cygwin or Windows service
- Installing Perl
- Installing Java
- Installing Apache plus server if using HTTP protocol
- Unix or Windows system administration

If you don't meet the above pre-requisites, you may want to contact your Subversion administrator or request that WANdisco perform a professional install for you.

You can have one MultiSite administrator for your replication group; one person can administer all sites through the Admin Console. The username and password is identical for all sites, which makes it easier for an admin to work between the various site's Admin Consoles.

There should be at least one person at each site who is familiar with WANdisco, as sites distributed throughout the world could need attention if a problem arises.

2.2 Physical Environment

WANdisco strongly recommends that you follow these guidelines to ensure the successful installation and use of Subversion MultiSite:

- The contents of `svnroot` directories on all the replicas match. Make sure the initial contents are exactly the same, including the repository UUID.
- The Subversion user/passwords on all repository hosts should match.
- running servers for each site in the replication group, pre-configured with
  - the same operating system
  - the same version of Subversion server
  - If using the `HTTP` protocol, the same `Apache` major/minor version, and `mod_dav` and `mod_svn_dav` versions
  - matching file and directory level permissions on Subversion repositories
  - a command line compression utility
  - Java (see Appendix A - Installing Java and Perl)
  - Perl (see Appendix A - Installing Java and Perl)
  - browser with network access to all servers

- have a dedicated server for the Failover Agent of each High Availability sub group
- If WANdisco and Subversion are running on the same server, the WANdisco installer can package a copy of the repository when setting up sites other than the first site
- If WANdisco and Subversion are running on the same server at all sites, then WANdisco can manage the Subversion password file
- e-mail from WANdisco containing the tar file link and attached production licence key file

2.2.1 WANdisco and Subversion Password Files

WANdisco offers to control the Subversion password files. If WANdisco controls the password files, any user you set up in WANdisco is also set up in Subversion. (WANdisco and Subversion must be running on the same severs at all sites.)

The majority of WANdisco customers elect to have WANdisco handle the passwords. The Admin Console offers an easy way to manage users and passwords.

During installation, you identify the Subversion password file's location, and WANdisco incorporates it into the replication group. If you have a lot of Subversion users, you can bulk import your users into WANdisco. This is discussed in section 5.4, Importing Users.

All replicas should have user information for all the users. The best way to ensure this is to have WANdisco control the Subversion password files.

2.2.2 Quorum Recommendations

Armed with an understanding of quorum types as described in the previous chapter, you can select the best solution for your configuration.

2.2.2.1 For MultiSite with No High Availability Sub Groups

WANdisco recommends selecting either Singleton Response or Majority Response quorum, balancing performance versus availability.

2.2.2.2 For MultiSIte with High Availability Sub Groups

WANdisco recommends Majority Response quorum. If you select Singleton Response quorum, the distinguished node represents a single point of failure.

2.2.2.3 For A Stand-Alone High Availability Group at One Location

WANdisco recommends having a group of at least three nodes, which automatically handles the failure of any single replicator node and its subsequent recovery.

For customers with a two node group, WANdisco recommends either Singleton or Majority Response quorum, and the second node must be the distinguished node.

2.2.3 Firewalls and Virus Scanners

2.2.3.1 Firewalls

You must determine if your replication group sits inside a firewall or outside of one. If the replication group is inside a firewall, the replication group ports are untouched by the firewall and you need take no action.

However, if any part of your replication group sits outside a firewall, you must configure the firewall so that the port numbers you specify during installation are not blocked or filtered.

2.2.3.2 Virus Scanners

If you have a virus scanner running on your network, you must configure it to not filter traffic on the ports you specify during installation.

2.3 Deployment Checklist

You may be familiar with this checklist from an evaluation copy of Subversion MultiSite. It is included here as reference..

System Setup ❖ All sites must share the same operating system	
Supported Operating Systems	Fedora (32 or 64 bit): 6, 7, 8, 9 Red Hat Linux Enterprise Server (32 or 64 bit): 4 Sun Solaris (32 or 64 bit): 9, 10 Linux: Linux kernel 2.6 or higher CentOS-4 Windows Server, (32 or 64 bit) 2003 Please read this for more information on NPTL. **Note:** VMware has a tendency to become unresponsive due to memory paging issues even without WANdisco present. Extra tuning may be needed to ensure optimal performance.
Subversion Server Version	1.3 and above. If you are using Subversion 1.5.4, use version 1.3.0 of Apache Portable Runtime.

Subversion Client Version	compatible with local Subversion servers
triggers	pre-commit triggers have to be deterministic, meaning they have the same outcome at every site. post-commit triggers can be tested at only one site.
System Memory	Ensure RAM and swapping containers are at least three or four times the largest Subversion file you have. Recommended: 1 GB RAM; 2 GB swapping container
Disk Space	• Subversion: depends on the number of projects and issues • MultiSite Transaction Journal: Recommended - equivalent of seven days of changes
File Descriptor limit	Ensure hard and soft limits are set to 64000 or higher. Check with the `ulimit` or `limit` command.
Journaling File System	Replicator logs should be on a journaling file system, for example, ext3 on Linux or VXFS from Veritas. Notes: NTFS is not a journaling file system: ext4 is a journaling file system, however WANdisco does not support its use because of its deferred writes.
Maximum User Process Limit	At least three times the number of Subversion users.
Java	Install JDK 1.5.0. **Note:** There should not be any spaces or control characters in the path where Java is installed. For example, c:\Program Files\java does not work with WANdisco as a JAVA install directory. See Appendix A - Installing Java and Perl.
Perl	Install version 5.6.1 or later. See Appendix A - Installing Java and Perl
Network Setup	
Reserved Ports (i.e. 6444, another for synchronizing)	WANdisco needs a dedicated port for DConENet (replication protocol) as well as HTTP protocol (for the Admin Console). WANdisco also recommends having a port available in case you have to copy (rsync) the repository from one site to another. If your network has a firewall, notify the firewall of the port numbers.
Firewall or virus scanner	Notify the firewall and any virus scanners of the Subversion MultiSite port numbers.
VPN	Set up IPsec tunnel, and ensure WAN connectivity.
Persistent Connection Keep Alive	Ensure VPN doesn't reset persistent connections for WANdisco, or else ensure there are no RST bugs
Bandwidth	Understand the available bandwidth for testing, and set user expectations.
DNS Setup	Use IP addresses instead of DNS hostnames, due to performance and DNS server unavailability issues. If using DNS hostname is the only option, then ensure DNS availability.

Apache 2 Setup (for http:// access)	
Apache version	All sites have the same version, 2.2.3 or above.
Apache modules version	All sites have the same version of mod_dav and mod_svn_dav
Apache KeepAlive	Ensure proper keep-alive settings. See Using HTTP with Apache, Apache and SVNKit, and http://httpd.apache.org/docs/2.2/mod/core.html.
mod_deflate.c for SVN_DAV	WANdisco requires that is module is not used
Location URI	Ensure that all sites' apache conf files have the same location URI for Subversion repository access
Require valid user for write methods	Ensure that all WebDAV methods require authentication for SVN-DAV protocol
Use port 80 for WANdisco	Standard Port 80 avoids confusion, change default Apache port if using 80.
Apache server port	Use non-standard Apache server port to avoid conflict with replicator port. See Dedicated Apache - Changing Subversion Port On Unix Flavor and Sharing Apache, and Using HTTPS.
File Permissions in svnroot	See this article (http://www.reallylongword.org/articles/svn/)
WANdisco Setup	
Quorum	Default is singleton. Trade off availability with performance.
Rotating Quorum Schedule	Ensure the distinguished node is in the active time zone.
Agreement Threads	Tune based on number of concurrent Subversion writers
Reader/Writer Network IO Thread Pool	Tune based on Subversion client connection rate, file transfer rate
ConnectionKeepAlive time-out	Tune inactivity timeout for persistent DConENet/DFTP connections based on VPN/WAN router set up
Message Queue Max Thread Pool Size	Tune based on Subversion write concurrency
Maximum connections per IO thread	Tune if active Subversion user population is large (greater than 100)
Disk space for recovery journal	Provision large disk for `svn-replicator/systemdb`, at least number of commits within a two to four hour window
Batch Processing	If there are any batch processes that interact with WANdisco, turn the deferred-writes to false.

For High Availability Groups	
HeartBeatInterval	Tune based on WAN latency, ensure interval is >> max ping time between Failover agent machine and replicator
MissingHeartBeat Count	Default is 4, if you see too many false alarms (spurious failover events) increase
Admin Email Address	To generate email notifications from the failover agent.
Notify all users that they must flush their client cache.	

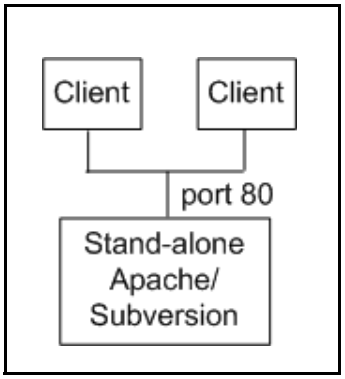
2.4 Creating a New Subversion Repository?

If you are creating a new Subversion repository, please follow the Subversion documentation at http://svnbook.red-bean.com.

2.5 Configuring Apache

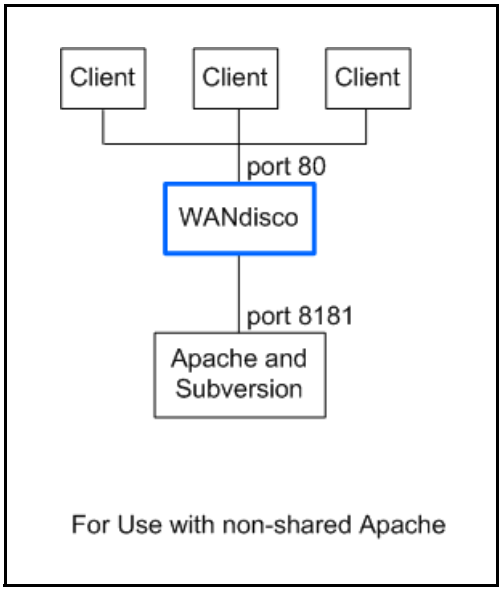
Carefully consider your environment when setting up Apache. You can either dedicate Apache to WANdisco, or share Apache with other locating or applications. WANdisco also supports using HTTPS.

Here is an example of a simple configuration without WANdisco.



2.5.1 Dedicated Apache - Changing Subversion Port On Unix Flavor

You can configure Apache to be dedicated to WANdisco, as shown in the following illustration.



Change the port number in the `httpd.conf` configuration file in the Apache server. Please see the `Listen` directive, discussed in the article at http://httpd.apache.org/docs/2.2/bind.html.

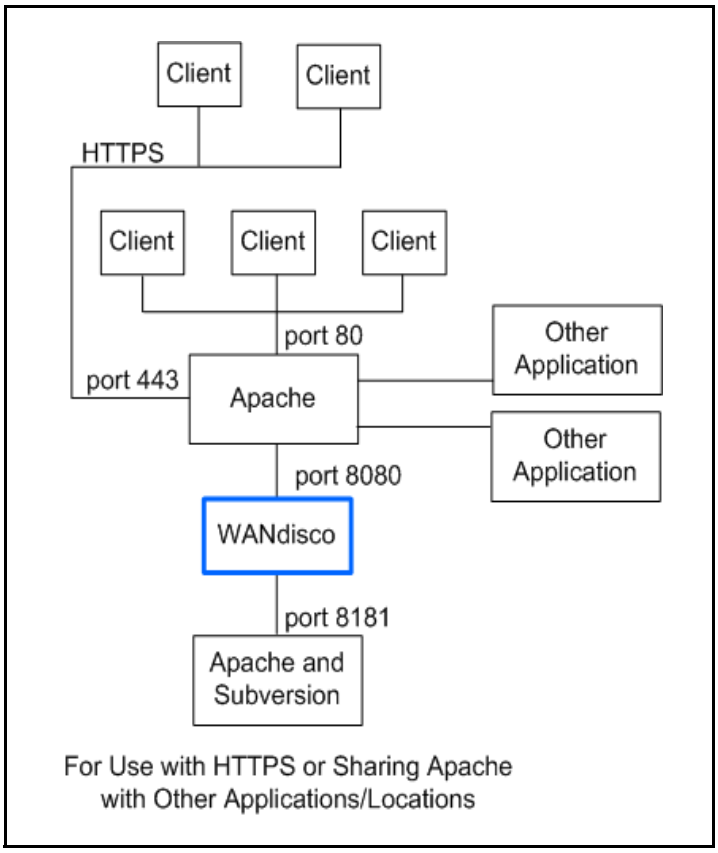
Here is a snippet of an `httpd.conf` file:

```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 8080
```

With this configuration, Apache server listens on port 8080 instead of default port 80.

2.5.2 Sharing Apache, and Using HTTPS

Alternately, you can configure Apache to use HTTPS, and you can share Apache with other locations or applications, as shown in the following illustration.



For Use with HTTPS or Sharing Apache with Other Applications/Locations

You can use this configuration if you enable a proxy.

These points are assumed:

- Apache server is running on port 80
- Apache web-dav module is running on port 8181
- WANdisco Subversion replicator is configured to listen on port 8080, and it forwards the requests to apache web-dav module on port 8181
- WANdisco port is listening on port 6444
- Apache SSL is running on port 443 to handle HTTPS requests (if this is set up, the stunnel package is not required)

- All the processes are running on the same machine
- Apache is compiled with `mod-proxy` and `mod-ssl` modules
- The Subversion URL is */svnrepos* (either as a parent-path or path)

When you perform the WANdisco installation, specify these ports for each replicator:

- WANdisco port 6444
- Proxy port 8080
- Apache web-dav port 8181 on localhost

In the `httpd.conf` file, specify the following parameters:

```
#
# Define apache port and pass anything that matches location /svnrepos to
# WANdisco SVN Replicator
#
NameVirtualHost *:80
<VirtualHost *:80>
ProxyPass /svnrepos http://127.0.0.1:8080/svnrepos
ProxyPass /!svn http://127.0.0.1:8080/svnrepos/!svn
ProxyPassReverse /svnrepos http://127.0.0.1:8080/svnrepos
ProxyPassReverse /!svn http://127.0.0.1:8080/svnrepos/!svn
RequestHeader edit Destination ^https: http: early
</VirtualHost>



Listen 8181
NameVirtualHost *:8181

<VirtualHost *:8181>
<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>
</VirtualHost>


# For the SSL option

Listen 443
<VirtualHost *:443>
ProxyPass /svnrepos http://127.0.0.1:8080/svnrepos
ProxyPass /!svn http://127.0.0.1:8080/svnrepos/!svn
```

```
ProxyPassReverse /svnrepos http://127.0.0.1:8080/svnrepos
ProxyPassReverse /!svn http://127.0.0.1:8080/svnrepos/!svn
RequestHeader edit Destination ^https: http: early
SSLEngine on
SSLCertificateFile /etc/httpd/conf/ssl.crt/server.crt
SSLCertificateKeyFile /etc/httpd/conf/ssl.key/server.key
SSLCACertificateFile /etc/httpd/conf/ssl.crt/ca-bundle.crt
</VirtualHost>
```

2.5.3 Using HTTP with Apache

This section applies for any Apache configuration.

In order to make a Subversion repository function in a distributed environment, Subversion Multi-Site requires exactly the same Apache/Subversion setup at all the sites. In addition, Apache's configuration needs to be modified to work in a peformant manner with Subversion (this is independent of Subversion MultiSite). Perform the following steps.

Step 1 Change Apache's connection keep-alive settings to allow long lived HTTP connections. Add this to the Apache configuration file `conf/httpd.conf` or included `conf/extra/httpd-defaults.conf`. For instance,

```
$ vi conf/httpd.conf
...
# Various default settings
Include conf/extra/httpd-default.conf
...
$ vi conf/extra/httpd-default.conf
...
#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300000
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection).
#
KeepAlive On
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
#
MaxKeepAliveRequests 0
#
# KeepAliveTimeout: Number of seconds to wait for the next request from
the
# same client on the same connection.
#
KeepAliveTimeout 500
...
```

Step 2 Ensure the SVN DAV settings in Apache's configuration files are exactly the same at all sites. The top level location URI prefix should be the same. We recommend copying the current `conf` file, and then changing the host:port settings. For instance, here is a `conf` file snippet with Apache virtual hosts (you do not have to use Apache virtual hosts, this is only an illustration):

```
# Site A
$ cat conf/extra/httpd-svn-dav.conf
...
NameVirtualHost site-a:8181
<VirtualHost site-a:8181>
<Location /dir0>
DAV svn
SVNPath /home/site-a/svnroot
AuthType Basic
AuthName wandisco
AuthUserFile /home/site-a/apache2/dist/conf/htpasswd
Require valid-user
</Location>
</VirtualHost>
...
# Site B
$ cat conf/extra/httpd-svn-dav.conf
...
NameVirtualHost site-b:9191
<VirtualHost site-b:9191>
<Location /dir0>
DAV svn
SVNPath /home/site-b/svnroot
AuthType Basic
AuthName wandisco
AuthUserFile /home/site-b/apache2/dist/conf/htpasswd
Require valid-user
</Location>
</VirtualHost>
...
```

Step 3 The Apache user names and passwords should match at all sites. Subversion MultiSite requires a valid username inside the HTTP authorization header to be passed for all DAV commands.

2.5.4 Apache and SVNKit

There are some known behaviors with connection pooling when using Apache and SVNKit. Adding a WANdisco product does not change the behavior of these issues, but you may need to revisit them after adding WANdisco.

WANdisco recommends using JavaHL with Eclipse IDE, which does not use connection pooling, and thereby eliminates any problems.

2.5.4.1 SVNKit and Connection Pooling

SVNKit uses connection pooling. For a given client, SVNKit opens two connections and keeps them open for later use. On a system with a heavy load and numerous clients, this can cause performance degradation. An open connection consumes an Apache worker thread, and with many clients and connection pooling, Apache may run out of worker threads. Apache provides various tuning parameters to optimize connection pooling but still release the unused connections. The tunable parameters are Timeout, KeepAliveTimeout, MaxKeepAliveRequests, and KeepAlive. Please refer to the Apache configuration documentation for further details at http://httpd.apache.org/docs/2.2/mod/core.html.

2.5.4.2 Tuning Values to Optimize Your Configuration

Apache has two timeout configurations: Timeout and KeepAliveTimeout. In general, the Timeout value should be higher than the value for KeepAliveTimeout.

With KeepAlive set to `true`, command line SVN clients are very diligent in closing connections. However, SVNKit keeps connections open. As Apache documentation states, if KeepAlive is set to `true`, the client should use an existing connection, and close it when done. Giving a somewhat conflicting command, SVNKit opens a connection and keeps it open. On issuance of a subsequent command, SVNKit may open a new connection, regardless of how many established connections are still open.

To force a connection closure upon command completion, set the KeepAliveTimeout to a smaller value than the value for Timeout.

Generally, a KeepAliveTimeout value of 15 seconds works for WANdisco products. If your application ends up dropping and then establishing the connection because of this low value, you may have to increase the KeepAliveTimeout value. (For example, you may notice a larger number of pending transactions on the Admin Console's Dashboard page.)

Under a normal load, a client follows a request within 15 seconds, but under a heavy load or no load, the number of seconds could vary widely, depending on your specific configuration.

The server sends the Timeout value to the client as a part of the response header, and the client uses this value to reset or resend the command if the server does not reply within that time. With a low value for Timeout (for example, 120 seconds), if the server for some reason does not complete the action and does not reply back, the client sends the command again on a different, newly established connection. If this happens with WANdisco, WANdisco ends up replicating an unnecessary transaction, and may not parse correctly with an invalid response.

WANdisco also requires some extra time for transaction coordination. In a singleton quorum, with a client connecting to the distinguished node, the transaction overhead is about 300 milliseconds.

To be on the safe side, WANdisco recommends you set the KeepAliveTimeout value much higher than 300 milliseconds. With KeepAlive set to `true`, and an appropriate value for KeepAliveTimeout, any Apache worker threads and lingering connections should be taken care of.

NOTE:

> WANdisco does not recommend setting KeepAlive to `false`. If you set KeepAlive to `false`, a client's transactions have an enormous overhead of establishing and destroying the connection. This overhead exists regardless of WANdisco.

2.5.5 Apache 2.2 with SVN-DAV on Windows

Step 1 Install Apache2.2.

Step 2 Install svn-win32-1.4.4 for Apache 2.2. Make sure it's Subversion for Apache 2.2.

Step 3 Copy `svn-win32-1.4.4/bin/intl3_svn.dll` to `apache/bin`.

Step 4 Copy `svn-win32-1.4.4/bin/libdb44.dll` to `apache/bin`.

Step 5 Copy `svn-win32-1.4.4/mod_authz_svn.so` to `apache/modules`.

Step 6 Copy `svn-win32-1.4.4/mod_dav_svn.so` to `apache/modules`.

Step 7 Uncomment these lines in `apache/conf/httpd.conf`:

```
LoadModule dav_module modules/mod_dav.so
LoadModule dav_fs_module modules/mod_dav_fs.so
```

Step 8 Add these lines to `apache/conf/httpd.conf`:

```
LoadModule dav_svn_module modules/mod_dav_svn.so
LoadModule authz_svn_module modules/mod_authz_svn.so

<Location /myDavLocation>
DAV svn
SVNPath C:\repo
SVNAutoversioning on
AuthType Basic
AuthName "SVN Repo"
AuthUserFile C:\repo\dav-auth
Require valid-user
</Location>
```

Step 9 Check that the users have been added to the `C:\repo\dav-auth` file. To add new users or change passwords, use `apache/bin/htpasswd.exe`.

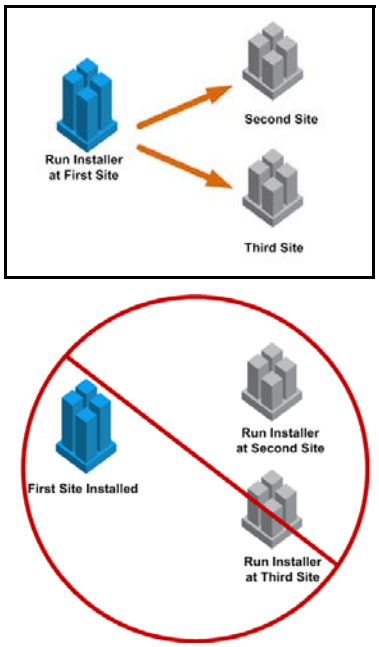
Step 10 Restart Apache.

Step 11 Point a web browser to

`http://server:port/<Your Dav Location>`.

3 Installation

You are going to run the installer at the first site, which creates packages for the other sites. Do not run the installer at any other site.



3.1 First Time Installation of First Site

Make sure that Subversion is running.

If you are going to set up a stand-alone High Availability group, the first site you set up here is the first site in the High Availability group.

Step 1 Save the svnrep.tar.gz file.

Step 2 For Windows, create a directory, and unzip the file in that directory. For other platforms, untar the file. The uncompressed file produces a directory, `svn-replicator`.

Step 3 Copy the license evaluation key file to the `svn-replicator/config` directory.

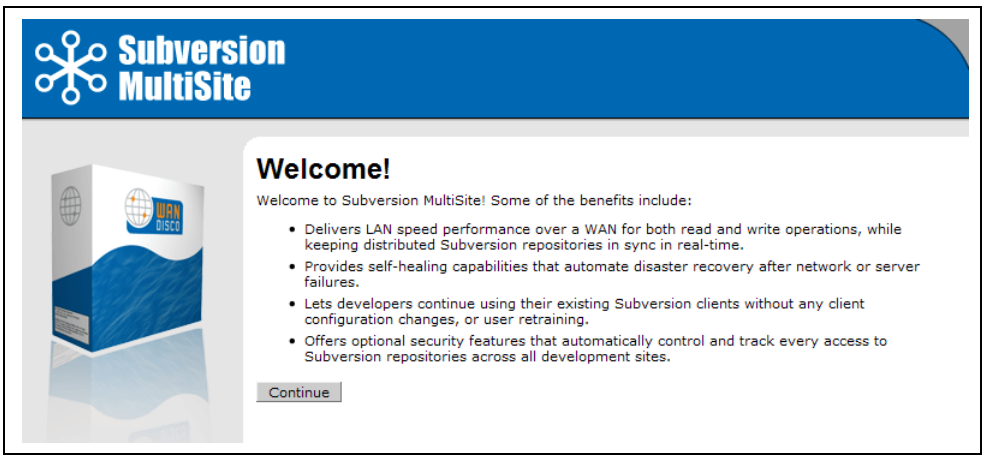
Step 4 At the command prompt (or editor), go to `svn-replicator/bin`.

Step 5 Type

```
perl setup
```

The last line of text returned contains a WANdisco URL.

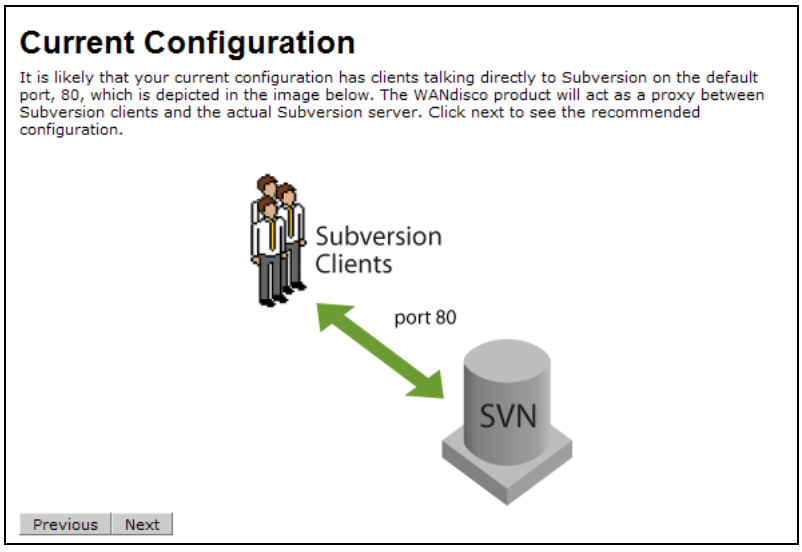
Step 6 Copy the URL returned in the last step and paste it into a browser. The MultiSite Welcome page appears.



Step 7 Click **Continue** at the Welcome text.

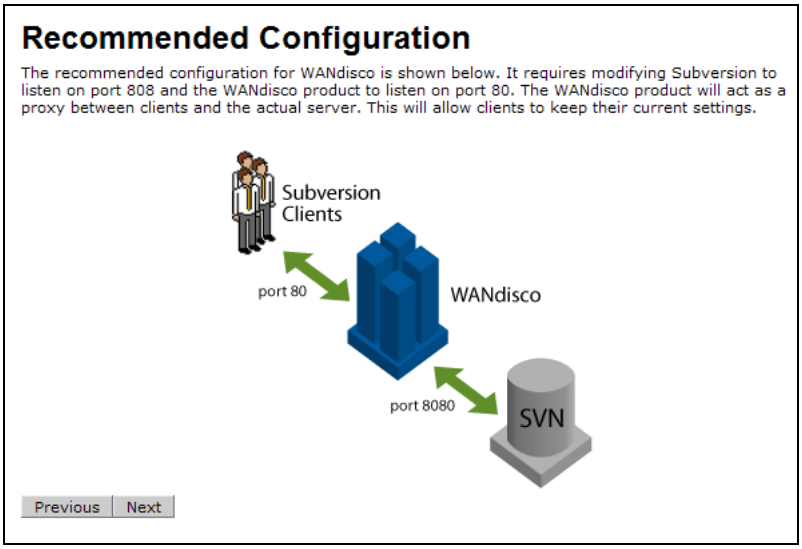
Step 8 Read the User Licence Agreement. You must agree to the terms to continue.

Step 9 The Current Configuration page appears. After reading it, click **Next**.

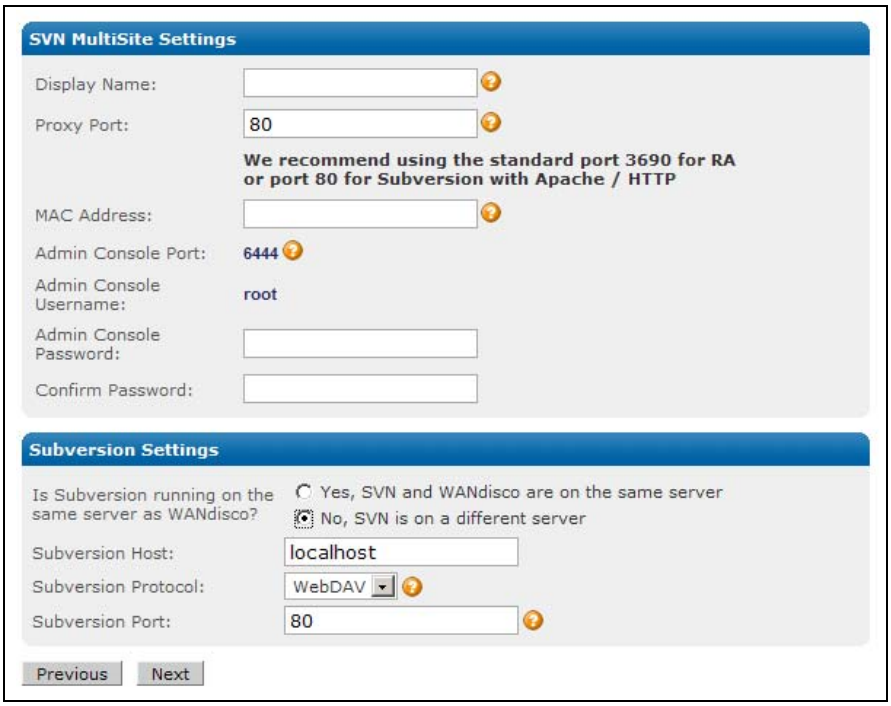


Step 10 Read about the recommended configuration. See 2.3, Deployment Checklist.

Step 11 Click **Next**. The Subversion MultiSite Settings page appears.



3.1.1 Specifying MultiSite Settings



Step 12 In the Display Name field, enter a name for this first site in the replication group.

Step 13 In the Proxy Port field, enter a port number. WANdisco recommends using 80 as the port number if you are using Apache/HTTP.

Step 14 Enter the MAC address. To find the MAC address, go to the machine's command prompt. For Unix, type

```
ifconfig
```

For Windows, type

```
ipconfig /all
```

The MAC address is the physical address.

For example, on a Windows machine, the output would look like:

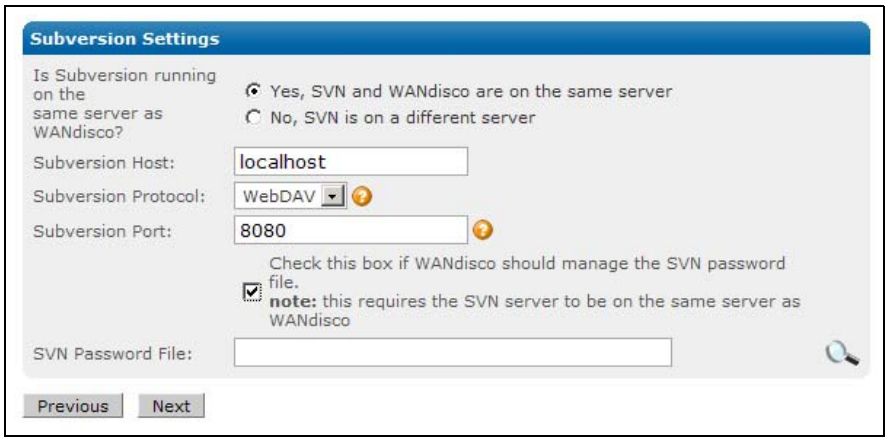
```
Physical Address. . . . . . . . . : 00-1A-A0-36-53-3C
Dhcp Enabled. . . . . . . . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IP Address. . . . . . . . . . . . : 192.168.1.124
Subnet Mask . . . . . . . . . . . : 255.224.255.0
Default Gateway . . . . . . . . . : 192.124.1.1
DHCP Server . . . . . . . . . . . : 192.122.1.1
```

```
DNS Servers . . . . . . . . . . . . : 192.128.1.50
                                      64.405.172.26
```

Step 15 WANdisco always sets the Admin Console port to 6444 and the Admin Console login name to `root`. All sites you set up initially have the same login name and password. You can later change a login name: see 6.13, Changing WANdisco's Root Login.

Type in a password for the login name, then confirm it.

3.1.2 Specifying Subversion Settings



Step 16 Check **yes** or **no** to tell WANdisco if Subversion is running on the same server as MultiSite (your current server).

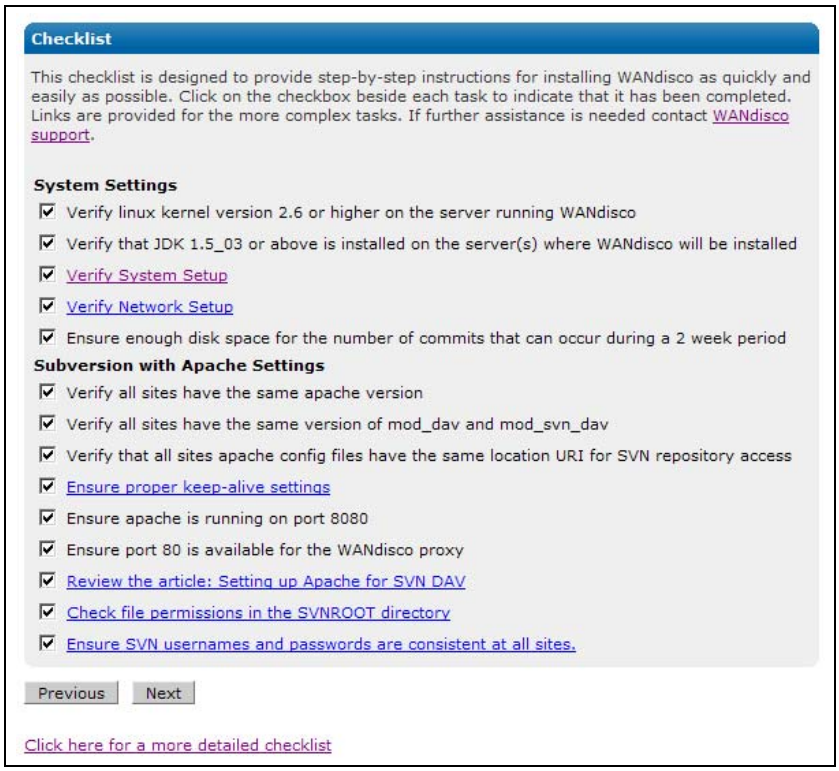
Step 17 Enter the Subversion host name.

Step 18 Enter the Subversion protocol.

Step 19 Enter the Subversion port number. This must be different than the one you specified in step 13.

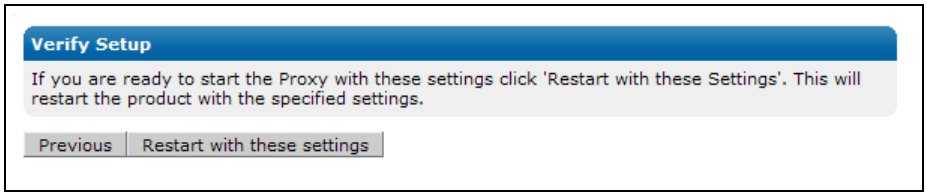
Step 20 If you have Subversion and WANdisco on the same server, and you want WANdisco to manage the Subversion password file, browse to the location of the Subversion password file.

Step 21 Click **Next**. The Checklist appears.

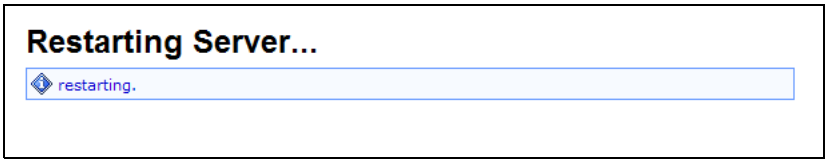


Step 22 Read over each item, and check each checkbox. You must be logged in to the WANdisco support site for the links to work.

Step 23 Click **Next**. The Verify Setup page appears.



Step 24 Click **Restart with these settings**. The installer configures and restarts your server.

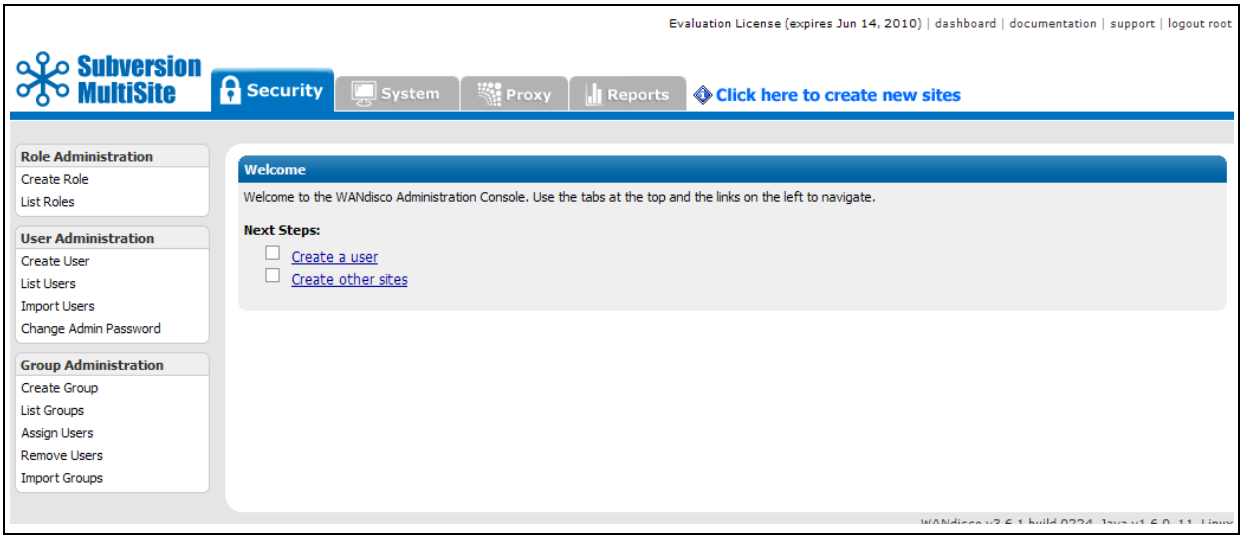


The installer creates a directory called wandisco that has the following directories inside.

DIRECTORY	CONTENTS
bin	Contains scripts like `svnreplicator`, `shutdown`
config	Contains the [replicator]/config/prefs.xml file used to configure MultiSite.
lib	Contains the `jar` files and DLLs that are required to run the product.
docs	Contains this *Administration Guide* in PDF format.
logs	Contains the pid file, log files and other temporary files. WANdisco Multi-Site's log file is named `SVNProxyServer-prefs.log`.
systemdb	Contains the system database with its transaction journal. **Warning**: Deleting or modifying files from systemdb will likely corrupt your installation.

3.1.3 Verifying the Installation

Step 25 Log in as root. You now have MultiSite's Admin Console running in your browser.



You must perform a few steps to ensure WANdisco is properly installed and communicating with Subversion.

Step 26 Create a user in WANdisco. If you previously chose not to have WANdisco update the password file, you must update it for the new user.

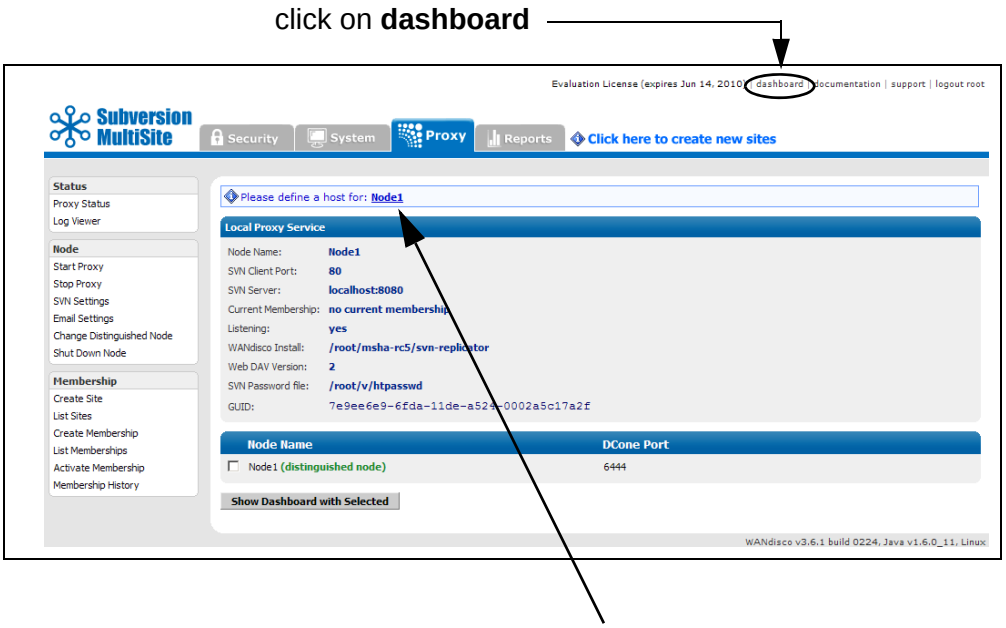
Step 27 From a client, log in to Subversion as that user.

NOTE:

Do not log in from the server where you installed WANdisco.

Step 28 Check out a file, modify it, and check it back in.

Step 29 Go to the WANdisco Admin Console's Dashboard to view the transaction.



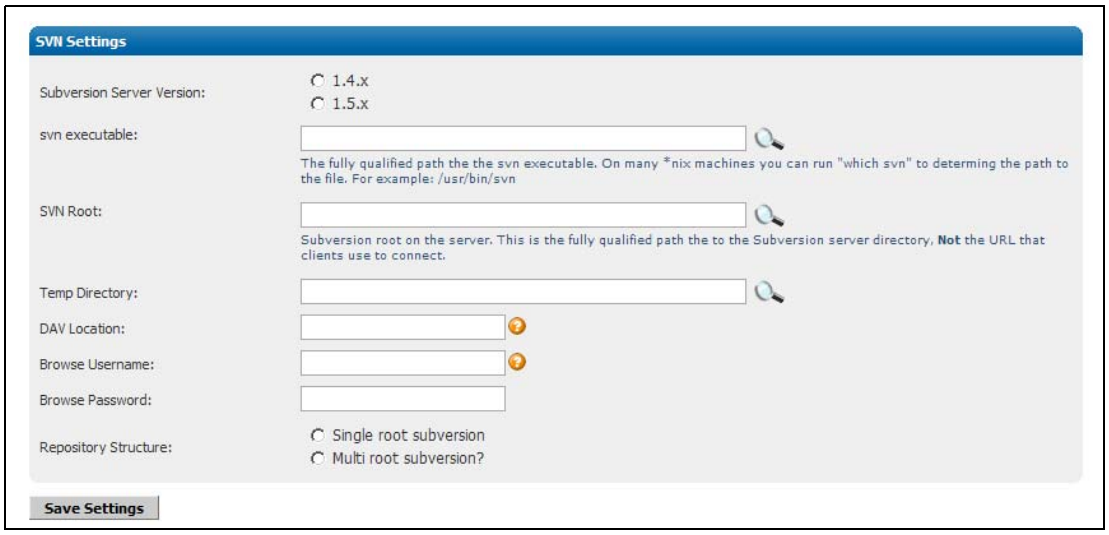
Step 30 Define the host for this node. Click on the node name link.

Step 31 Enter the IP address for this node.

3.1.4 Optional: Defining Subversion Repository to Package

Perform this step if you have elected to have WANdisco package the Subversion repository. You must define it now.

Step 32 If you have not already done so, define Subversion root by clicking on the **SVN Settings page** link and browsing to the SVN Repository Root. This setting gets replicated to the other nodes when you create the membership.



Step 33 Click **Save Settings**.

NOTE:

The installer packages whatever users are in the first site for distribution to the other sites.

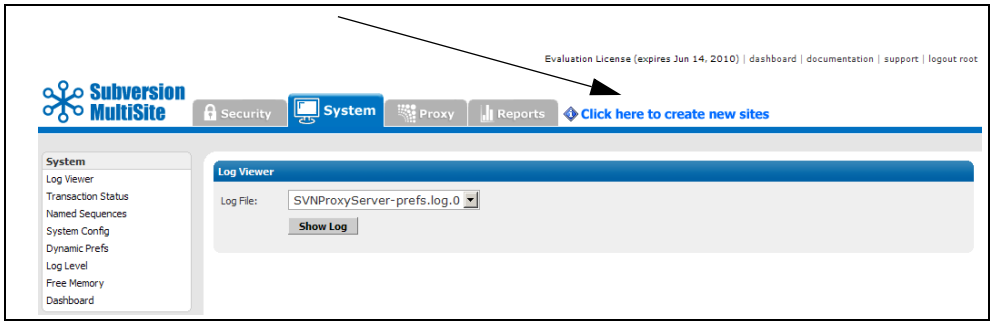
3.2 Specifying Subsequent Sites

Now that you have verified the first site's installation, you can continue to distribute MultiSite to other sites. You first have to create each site in your replication group, including any High Availability nodes. Each High Availability sub group needs its own Failover Agent.

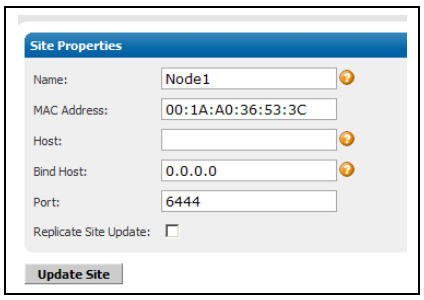
For clarification, here is the task flow for three possible customer scenarios.

Action	MultiSite	MultiSite and High Availability	High Availability
create other sites	✓	✓	✓
create Membership	✓	✓	✓
create HA Sub groups		✓	✓
prepare Membership	✓	✓	✓
activate Membership	✓	✓	✓
copy installer files to other nodes	✓	✓	✓
bring up other nodes	✓	✓	✓
bring up Failover Agent		✓	✓

The procedure steps you through all the scenarios. The screen shot illustrations are based on a sample installation group of six nodes, with two High Availability sub groups of two nodes each. (That means a total of eight nodes are defined for this group, including the two Failover Agents.)



Step 34 Click on the link **Click here to create new sites**. The Site Properties page appears.

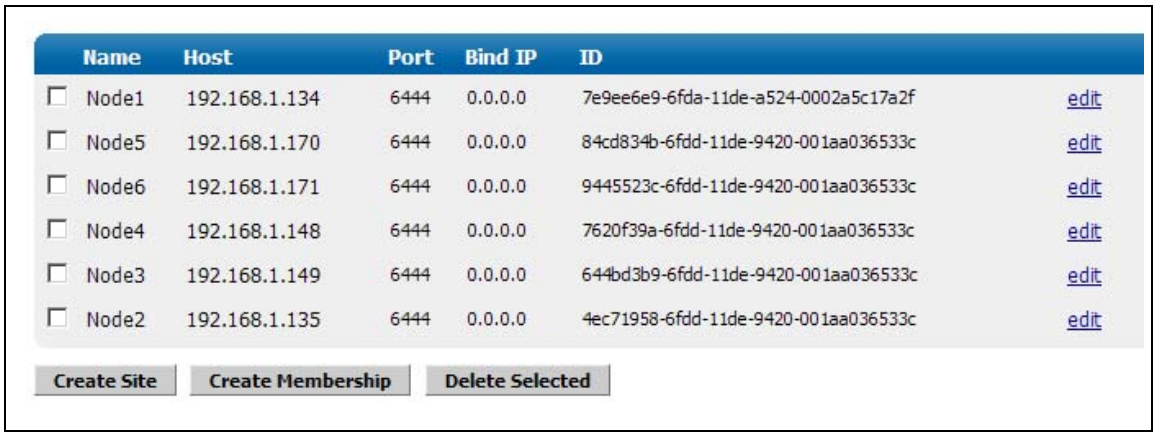


Step 35 Enter a name, the MAC address, host name, bind host and WANdisco port number for the site.

Step 36 Click **Create Site**. A page appears containing information for the first site and the site you just created.

Create all the sites in your replication group, including any that will be in an HA sub group. For HA: you'll create the Failover Agent in another step.

The following illustration shows six nodes.



Step 37 Create the membership. Click **Create Membership**. The Membership Properties page appears.

Step a Enter a name for the Membership.

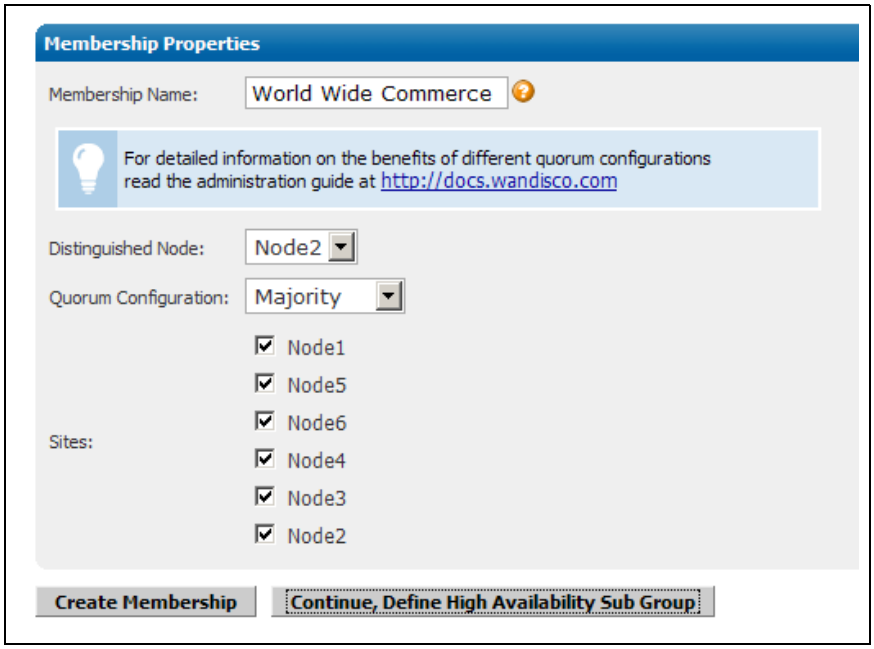
Step b Select a distinguished node for the replication group. For MultiSite installations (without HA), the first site is the distinguished node. For MultiSite HA, you can select any node. For two-node MultiSite HA groups with no other nodes, make the distinguished node the back up node. See 1.2.5, Stand-Alone Failover Group of Two Replicators.

Step c Select a quorum. For a discussion of quorum, see section 1.1.2, Replication Example. Note that evaluation licenses for MultiSite with no High Availability only allow singleton quorum.

Step d Check all the nodes that you want in this replication group.

Step e For non-HA deployments, click **Create Membership** and go on to step 39.

For HA deployments, click **Continue, Define High Availability Sub Group**.



Step 38 Define the HA Sub Group.

Step a Enter a name for the Failover Agent. The Failover Agent is now a stateless, non-voting node in the membership.

Step b Enter the Failover Agent's IP address.

Step c Replication Port is the port that WANdisco uses for its communications. (Protocols are replication, HTTP and heartbeat.) Make sure no other process shares this port.

Step d The Bind Host is the IP address/host name that WANdisco listens on for incoming connections. Unless there are multiple network cards on the Failover Agent, this is typically set to 0.0.0.0.

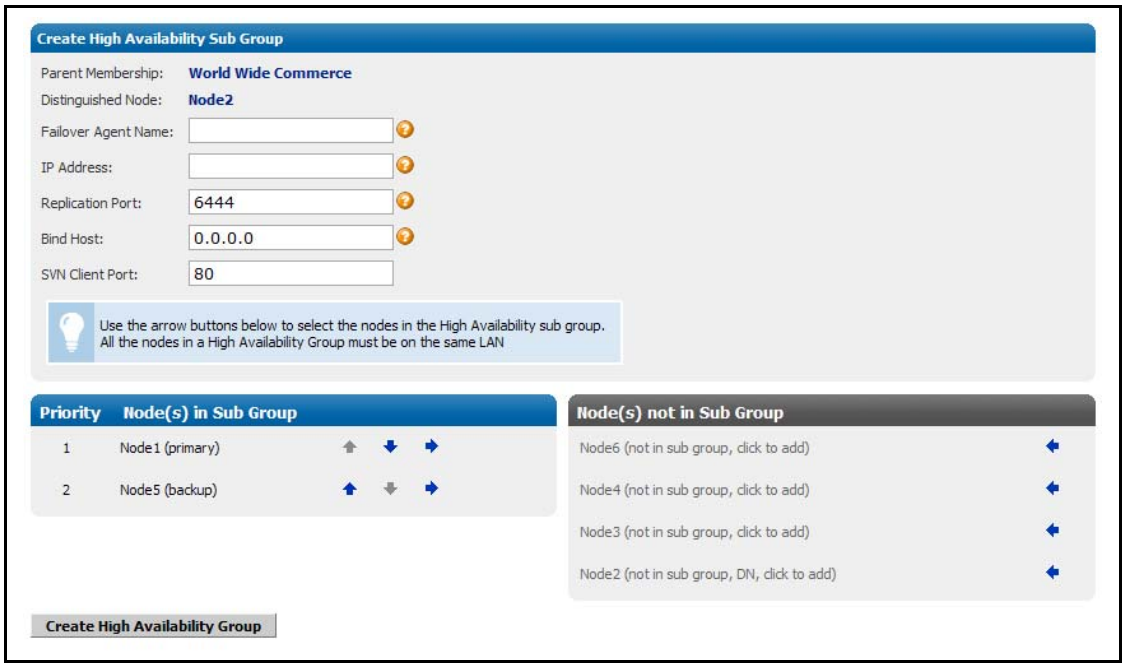
Step e The SVN Port is for Subversion client communication.

Step f Select the nodes for this sub group. Use the arrow keys.

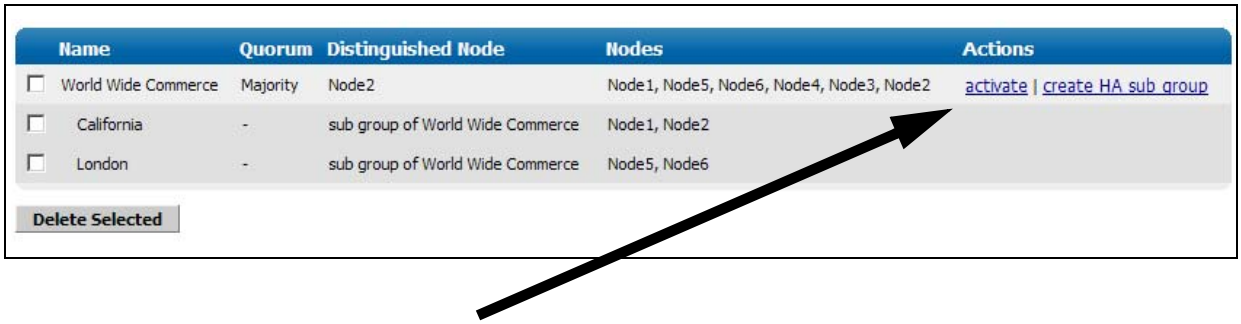
Step g Specify priority order. See 1.2.2.2, Priority Order. For two-node stand-alone MultiSite HA groups, the backup node must be the distinguished node.

Step h Click **Create High Availability Group**.

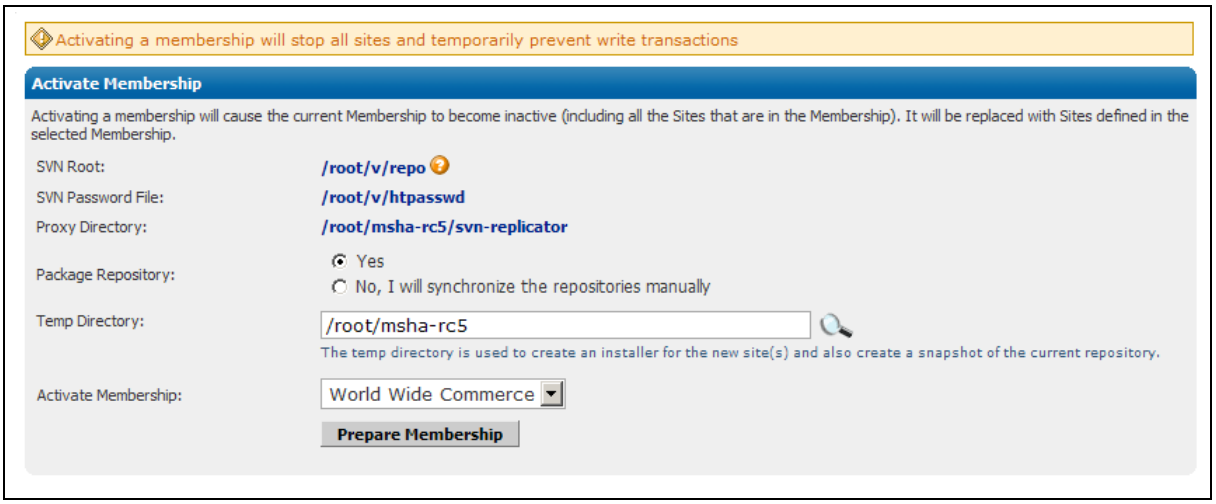
Step i Create all the HA sub groups you planned for.



Step 39 Activate the membership. Click **List Memberships**. Click **activate**.



The Activate Membership page appears.



3.2.1 Completing Membership Activation

Later in this section, a step lets you know when you can release the first node to Subversion users. Customers with an HA-only installation must finish the HA group and Failover Agent installation before releasing the nodes to users.

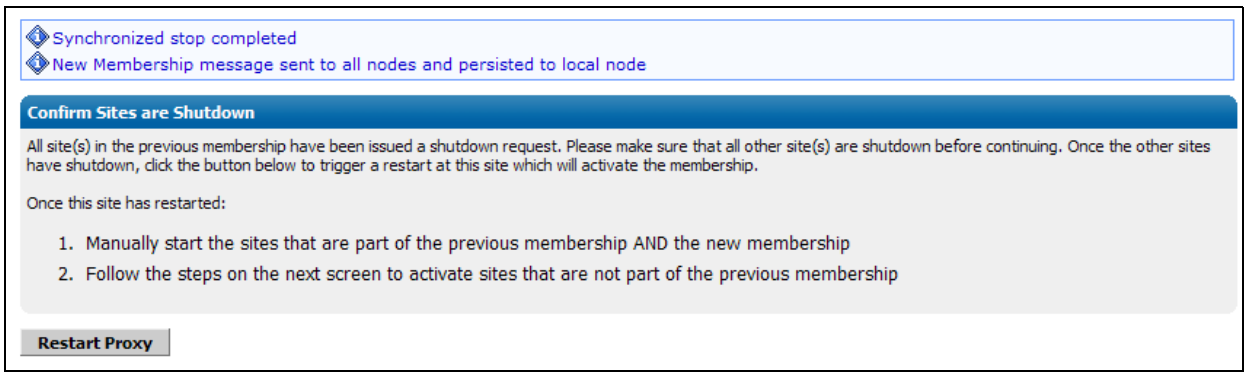
Replication captures any transactions made between the time the first node is released to users, and the time any subsequent sites come up, and replays them for the sites coming up. WANdisco recommends getting the other sites up as quickly as possible, certainly within one week of the first node's going live.

Step 40 Specify whether you want the installer to package the Subversion repository for the other sites.

Step 41 Specify a temporary directory where you want the installer to put the compressed installation files for the other sites.

Step 42 Select which membership group to activate from the pull-down menu.

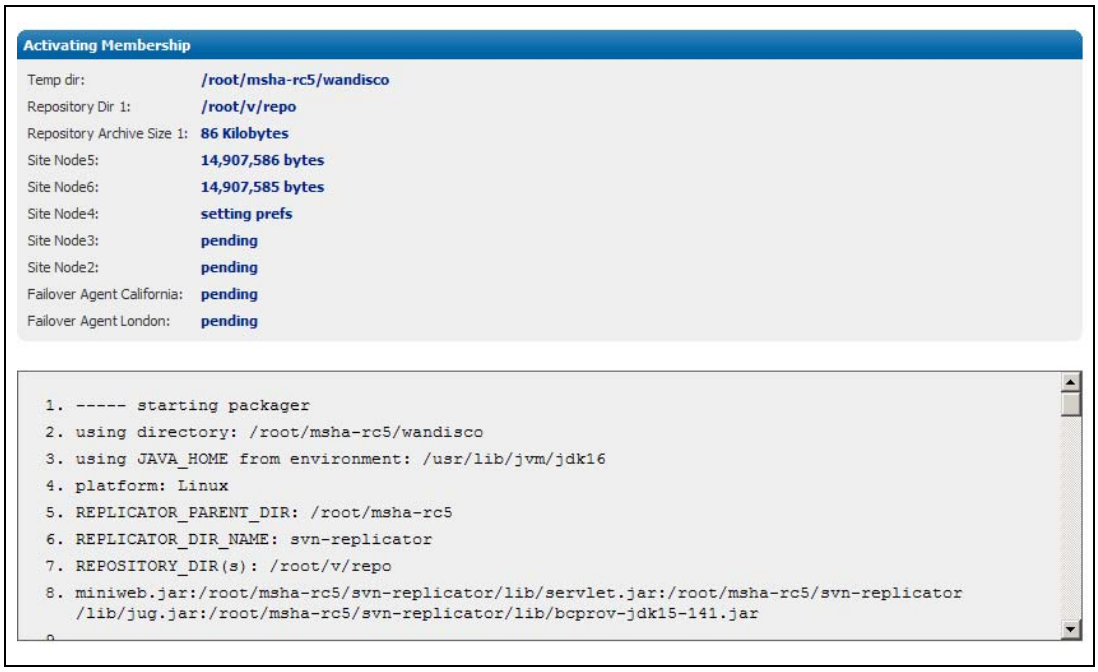
Step 43 Click **Prepare Membership**. The Confirm Sites are Shut Down page appears.



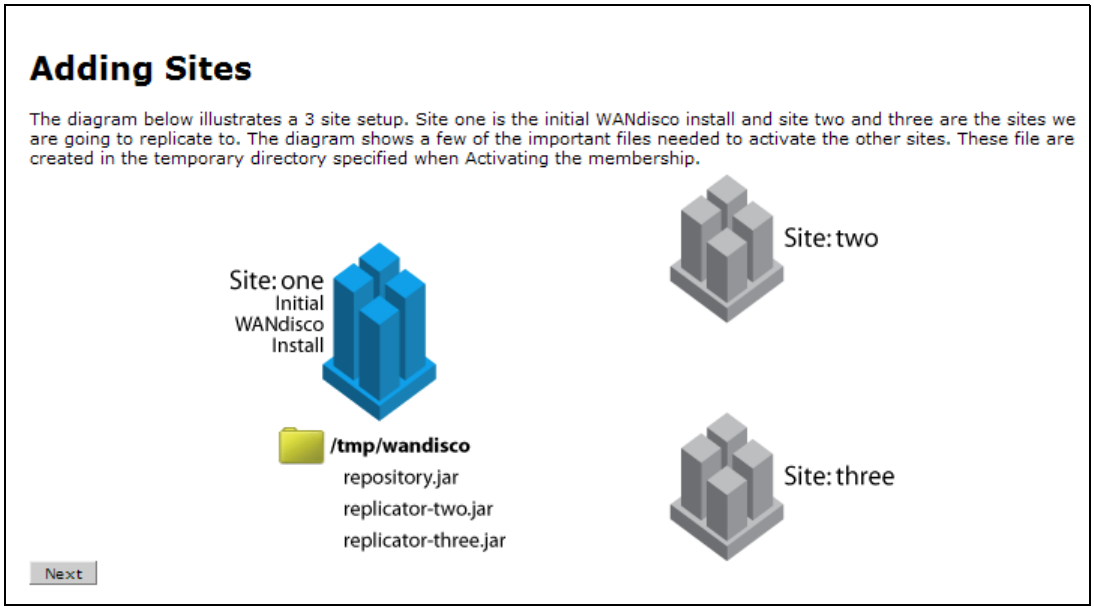
The first site is stopped, and the membership is activated.

3.2.1.1 Packaging Other Sites

Step 44 Click **Restart Proxy**. The Activating Membership page appears.

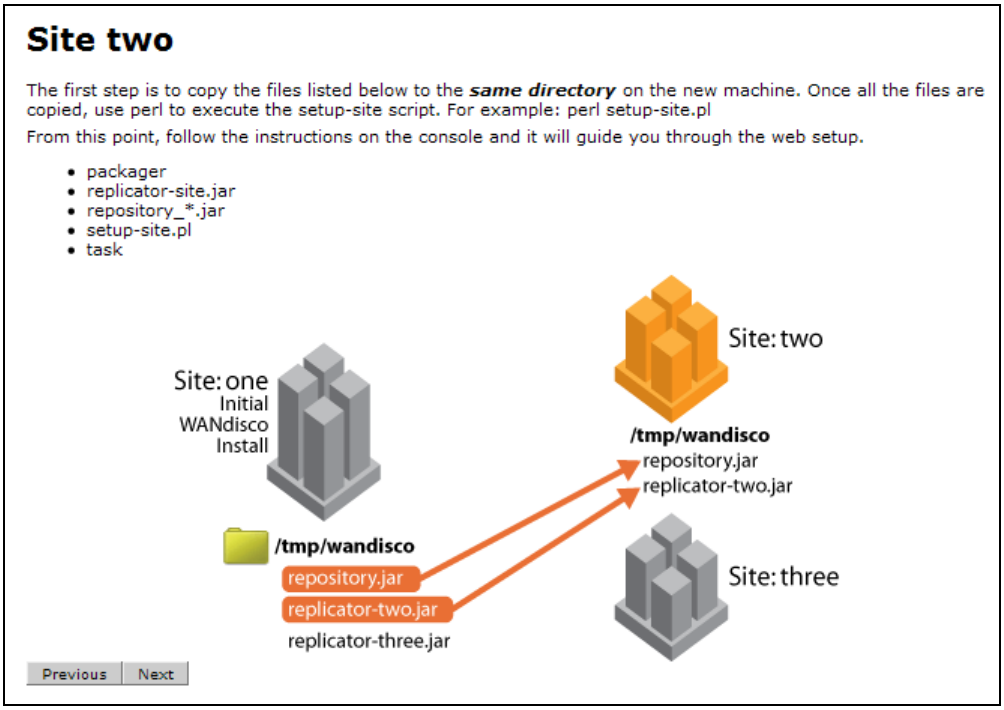


The Adding Sites page displays.



Step 45 You can now release the first site to Subversion users (unless this site is part of an HA sub group). As a member of the replication group, any modifications to site one are replicated to the other sites in the replication group when those sites come up.

Step 46 At the Adding Sites page, click **Next**.



Step 47 Make the same `WANdisco` directory (you specified in step 41) at each of the new sites.

Step 48 Copy five files for each site. The files are located at the first site, in the `WANdisco` directory. For each site, copy:

```
packager
replicator-<site>.jar
repository_*.jar
setup-site.pl
task
```

For HA sub groups, each Failover Agent has its own compressed file. Copy

```
failover-<Failover Agent Name>.zip
```

3.3 Activating Subsequent Sites

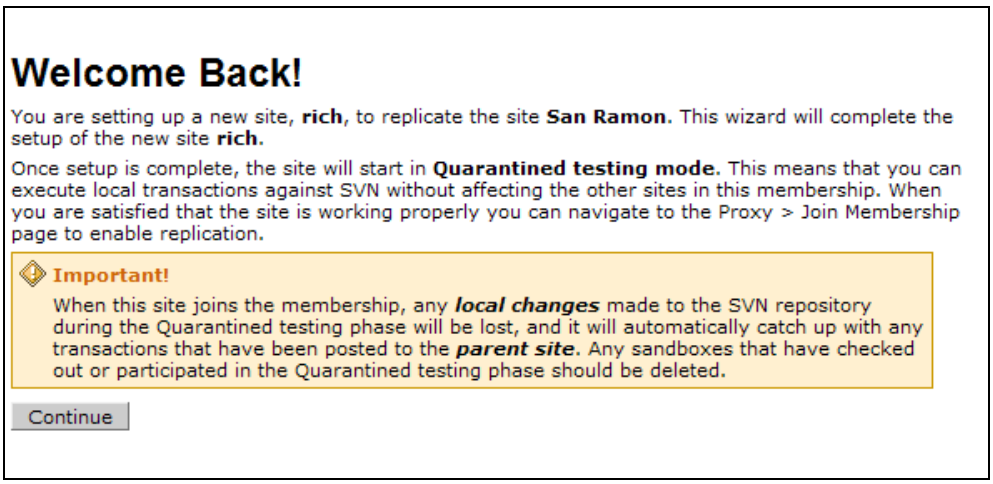
This section is for activating sites. For an HA group, activate all your sites, then activate the Failover Agent. See 3.4, Activating a Failover Agent.

Step 49 Start the installer at the new site. At the command prompt, type

```
perl setup-<site name>.pl
```

The output returns a URL in the last few lines.

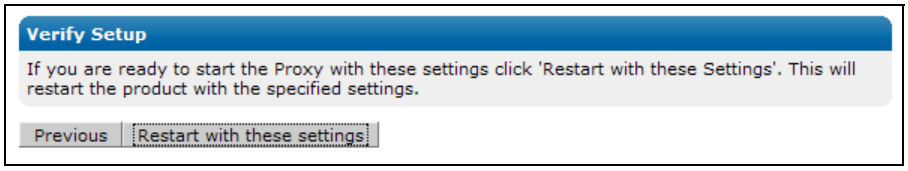
Step 50 Copy the URL into a browser. The Welcome Back! page appears.



Step 51 Click **Continue**. The Checklist page appears for this site.

Step 52 Check all the boxes in the Checklist and click **Next**. The Verify Setup page displays.

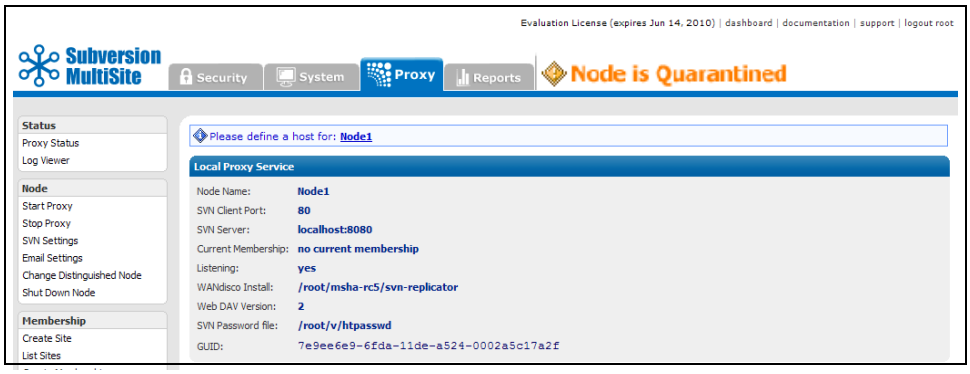
Step 53 If everything looks ok, click **Restart with these settings**.



When the server restarts, the Admin Console appears for the new site.

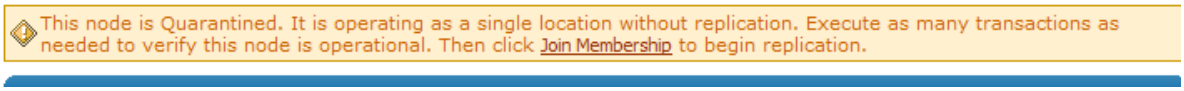
3.3.1 New Site Quarantined

MultiSite is up at the new site. Note that the site is quarantined until you verify the validity of the installation. After you verify the installation, you will replace the Subversion repository with either the packaged one or one you've copied.



3.3.1.1 Verifying the Installation

Step 54 Log in as root. The following message appears.



Step 55 Test this installation. Perform, at a minimum, a commit transaction and a create user command. (Create User is in the User tab.) Verify the user was created in Subversion as well.

Step 56 Re-synchronize the repositories. If you had WANdisco package the repository, this occurs in the next step.

If you copied the repository yourself to the new location, use a tool like `rsync`, keeping the first node's repository as the baseline.

3.3.1.2 Taking a Site Out of Quarantine

Step 57 Join this site to the replication group membership. Click **Activate Membership** in the Proxy tab. WANdisco joins this site to the replication group, catching it up on any transactions that occurred within the group.

Step 58 This site can be released to Subversion users while you bring up any other sites. Sites that belong to HA sub groups are not available to users until you install the Failover Agent. See 3.4, Activating a Failover Agent.

Step 59 Repeat steps 49 through 58 for each additional site.

Step 60 Set up Watchdog email notifications. Subversion MultiSite starts in watchdog mode by default, which restarts MultiSite if it goes down. You may want to set up the watchdog email now, so that an administrator receives an email from WANdisco if MultiSite goes down. Note that the watchdog email setting is per site. See 8.5, About Watchdog Mode.

You now have successfully installed Subversion MultiSite.

3.4 Activating a Failover Agent

If your replication group has a Failover Agent, follow these steps to activate it.

Step 1 If you have not done so already, move the compressed file (`failover-<sitename>.zip`) to the Failover Agent.

Step 2 Uncompress the file.

Step 3 For Unix, change the permissions for the files in the `/bin` directory. Type

```
chmod +x svn-failover/bin/*
```

Step 4 At `svn-failover/bin`, type

```
./failoveragent
```

For Windows, type

```
failoveragent
```

Step 5 The Failover Agent Admin Console is now available to view in a browser. You can also see it through any node's Admin Console.

3.4.1 Post Installation Configuration

There are a few issues you may want to address pretty quickly after installation, and are described here. Any action you take is through the Admin Console, described in 4, Using the Admin Console.

3.4.1.1 Configuring Node Start Up Commands

The WANdisco Failover Agent is capable of starting and stopping its sub group nodes from the Admin Console directly, provided you enter the startup commands in the Admin Console.

To configure start up commands, see the Start Command definition in Chapter 4, Using the Admin Console.

3.4.1.2 Email Alerts for Failover Events

The WANdisco Failover Agent can generate email alerts whenever it detects an event related to failover. Examples of such events are:

- unable to transition to unilateral mode
- when transitioned to unilateral mode
- when unilateral mode starts
- when failed to backup
- when the primary is not available when the failover agent starts

To set up email alerts, see the Admin email definition in Chapter 4, Using the Admin Console. Since HA groups are on a LAN, in all likelihood, the Failover Agent email settings would be the same as you configured for the HA nodes watchdog email alerts.

3.5 Installing Upgrades

NOTES:

This procedure involves taking Subversion offline. Please follow your company procedures about notifying Subversion users of down time.

3.5.1 Stop WANdisco

Step 1 After notifying Subversion users of the downtime, stop Subversion MultiSite. Perform a synchronized stop of all sites. See 6.7, Performing a Synchronized Stop.

Step 2 Shut down each site. Go to each site's Proxy tab, and click **Shut Down Node**.

3.5.2 Export WANdisco Data

Step 3 On the System tab, select **Export Settings**. These settings include user information. You only need to do this at one site, since the files at other sites are identical.

3.5.3 Preparing the Sites

At each site:

Step 4 Copy the entire `svn-replicator` directory to a backup location.

Step 5 On the original install node, zip the `config` directory.

Step 6 Delete the `systemdb` directory.

Step 7 In the `svn-replicator/config` directory, delete these directories:

```
membership
security
passwd
```

3.5.4 At the Install Site

Step 8 Save the `svnrep.tar.gz` file and extract it to a new directory, for example `new/svn-replicator`.

Step 9 Unzip or untar the file.

Step 10 Verify the md5 checksum. For Unix, type

```
md5sum <filename>
```

Step 11 Copy the file `new/svn-replicator/lib/svn-replicator.jar` to the `svn-replicator/lib` directory.

Step 12 Copy the entire directory `new/svn-replicator/lib/html` to `svn-replicator/lib/html`.

Step 13 Copy the licence key file to `svn-replicator/config` directory.

3.5.5 Running the Install Program

Step 14 Run this command on the original install site. The install program automatically populates all previous configuration information.

```
svn-replicator/bin/setup
```

The Setup page appears.

Step 15 Click **Next** to continue.

Step 16 Make changes as needed. The Admin Console appears.

3.5.6 Importing WANdisco Data

Step 17 Import WANdisco data. On the System tab, select **Import Settings**. Import the file you exported in step 3.

Step 18 Perform a synchronized stop.

Step 19 For additional sites, refer to section 3.2, Specifying Subsequent Sites.

You have successfully installed a Subversion MultiSite upgrade.

4 Using the Admin Console

The Admin Console is a simple interface that allows you to monitor and perform administrative tasks for Subversion MultiSite.

You can run the Admin Console from any site. The login is `root`. Passwords are the same for all sites, and are set during installation.

To access the Admin console, all you need is the IP address of any site and the WANdisco port number.

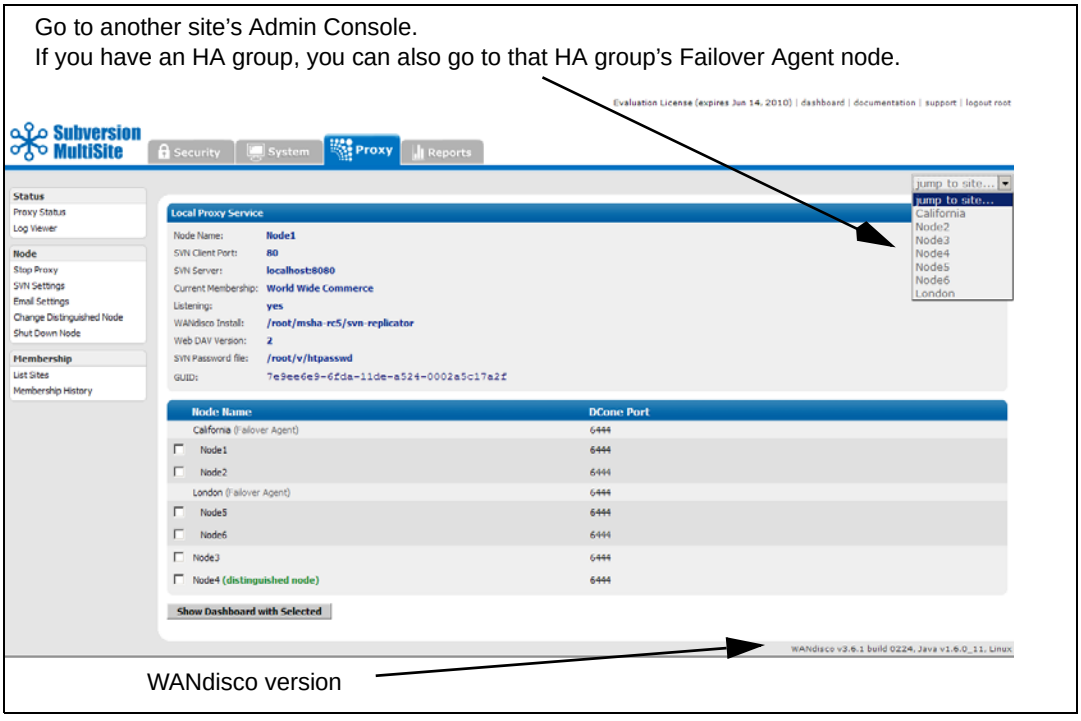
4.1 Starting the Admin Console

To start the Admin Console, in a browser's address bar, type

http://<IP address>:<replication port number>

The Admin Console's Home page appears.

MultiSite's Admin Console has four tabs, identified by their tabs: the Security, System, Proxy, and Reports tabs.



4.2 The Security Tab

Role Administration

For Access Control. Refer to the *Subversion Access Control Administration Guide* at http://docs.wandisco.com/svn/ac/3.6.1/adminguide.pdf.

User Administration

Create User	Create any Subversion user.
Username	Enter in the Subversion user's username.
Password	Enter the user's password.
Confirm Password	Confirm the password.
First Name	Enter the user's first name.
Last Name	Enter the user's last name.
Email	Enter the users email address.
List Users	This command displays all users.
Import Users	You can import an existing list of users. The import file must be a comma delimited text file, of the format `userid,last-name,firstname,email`.
Change Admin Password	You can change the WANdisco Admin password *for this site only* with this command. See 6.13.1, Changing the Password.

Group Administration

For Access Control. Refer to the *Subversion Access Control Administration Guide* at http://docs.wandisco.com/svn/ac/3.6.1/adminguide.pdf.

4.3 The System Tab

The System tab offers several commands and utilities for the MultiSite replication group.

4.3.1 Left Side Menu

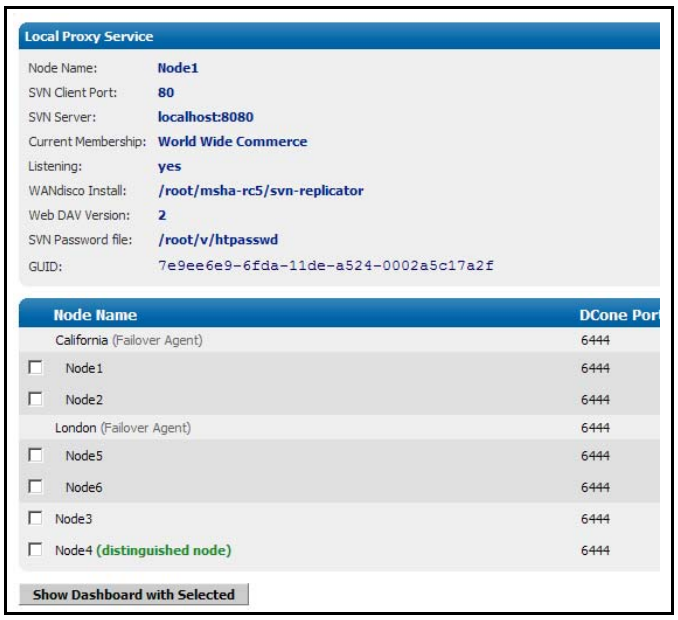
System

Log Viewer	
SVNProxyServer-prefs.log	The installation log file.
web proxy.log	A log from the installation.
Show Log	Click this to display a log in the Dashboard.
Transaction Status	You can search for a specific transaction.
Transaction Number	Enter in the transaction number.
Site Submitted From	Select the site where the transaction originated.
Show Status	Click this after entering the data in the above fields.
Named Sequences	Used internally.
System Config	For Access Control.
Dynamic Prefs	Used internally.

Log Level	WANdisco uses one log, and the default level is **info**. The levels vary from **severe**, where you get only the most severe warnings, to **finest**, which logs every action.
Free Memory	This command frees the memory (GC stands for garbage collection) for the current site. The command occurs when you click on this menu selection. The display shows information on the command that was just performed.
max mem used by JVM	the maximum memory that JVM can use on the current site
free memory before GC	the amount of free memory before you ran this command
free memory after GC	the amount of free memory after you ran this command
memory freed	the total amount of memory freed at the command's completion
Dashboard	offers another way to get to the Dashboard. The Dashboard is discussed in detail in section 4.6, The Dashboard.
Backup	
Import Settings	This command allows you to import WANdisco settings, including all users.
Export Settings	This command allows you to export WANdisco settings, including all users, for later importation into a WANdisco product.

4.4 The Proxy Tab

The Proxy tab by default lists the Proxy Status for this node.



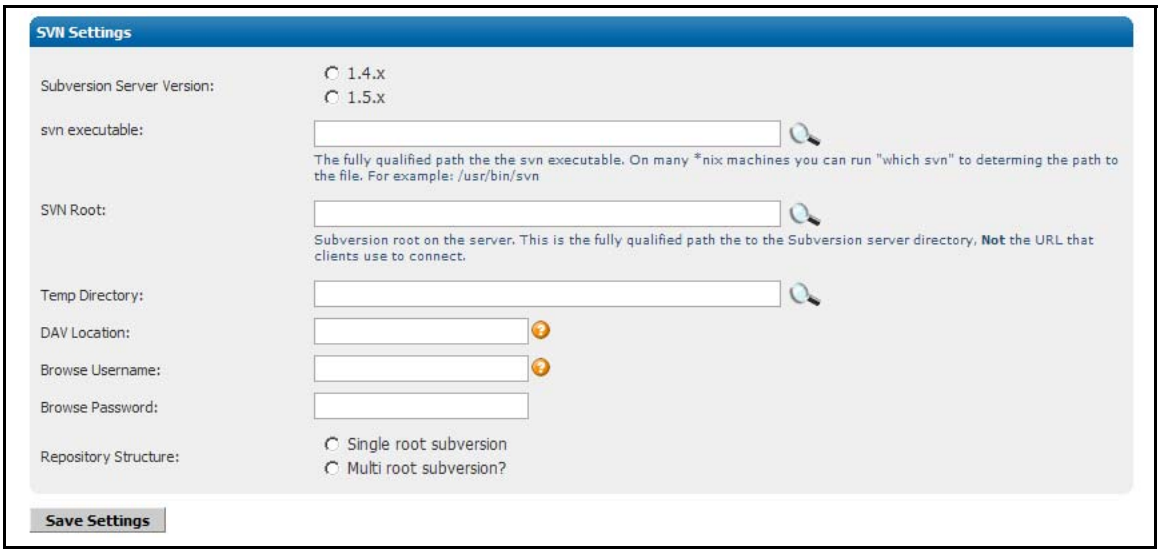
4.4.1 Left Side Menu

Status

Proxy Status	This command displays the site's status in the tab's main area.
Log Viewer	You can view the `SVNProxyServer-prefs.log` file in the Dashboard.

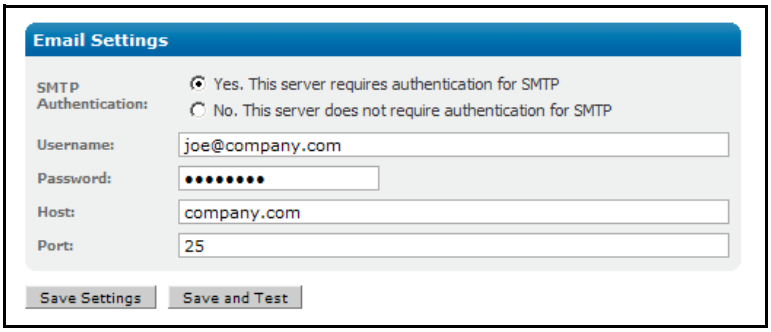
Proxy

Start Proxy	This starts WANdisco at this site.
Stop Proxy	
Stop this proxy only	This stops WANdisco at this site. See 6.1.1, Temporarily Disabling Subversion Access At Selected Sites. If this node is part of an HA sub group, this option is not available. See 4.5.1.1, Starting and Stopping HA Sub Group Nodes.
Synchronized stop of all proxies	A synchronized stop stops WANdisco at all sites, and replication is suspended. See 6.7, Performing a Synchronized Stop.



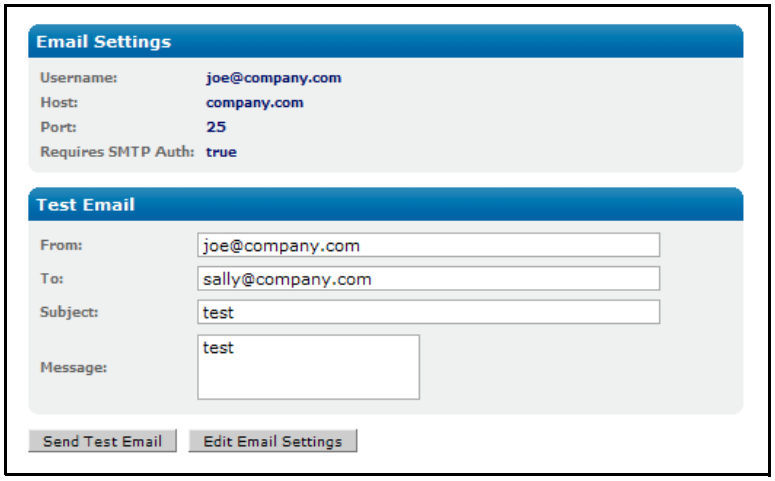
SVN Settings

Subversion Server Ver	You only need to specify this if you are using pre-replication hooks. See 6.14, Setting Up a Pre-Replication Hook.
svn executable	Only needed for Access Control. Refer to the *Subversion Access Control Administration Guide* at http://docs.wandisco.com/svn/ac/3.6.1/adminguide.pdf.
SVN Repository Root	Browse to the Subversion root on the server (where the directories are stored).
Temp Directory	This is the install directory on the node.
DAV Location	Only needed for Access Control. Refer to the *Subversion Access Control Administration Guide* at http://docs.wandisco.com/svn/ac/3.6.1/adminguide.pdf.
Browse Username	Only needed for Access Control. Refer to the *Subversion Access Control Administration Guide* at http://docs.wandisco.com/svn/ac/3.6.1/adminguide.pdf.
Browse Password	Only needed for Access Control. Refer to the *Subversion Access Control Administration Guide* at http://docs.wandisco.com/svn/ac/3.6.1/adminguide.pdf.
Repository Structure	You only need to specify this if you are using pre-replication hooks. See 6.14, Setting Up a Pre-Replication Hook.
Save Settings	Save any changes.



Email Settings — Set the email settings for the watchdog contact. You have to set the email in watchdog also. See 8.5, About Watchdog Mode.

- SMTP Authentication — Select whether you need SMTP Authorization.
- Username — For authentication, enter a valid username.
- Password — For authentication: enter a valid password.
- Host — Enter the email host.
- Port — Enter the email port number.
- **Save Settings** — Save any changes.
- **Save and Test** — Save any changes, and run an email test. Selecting this displays the test page.

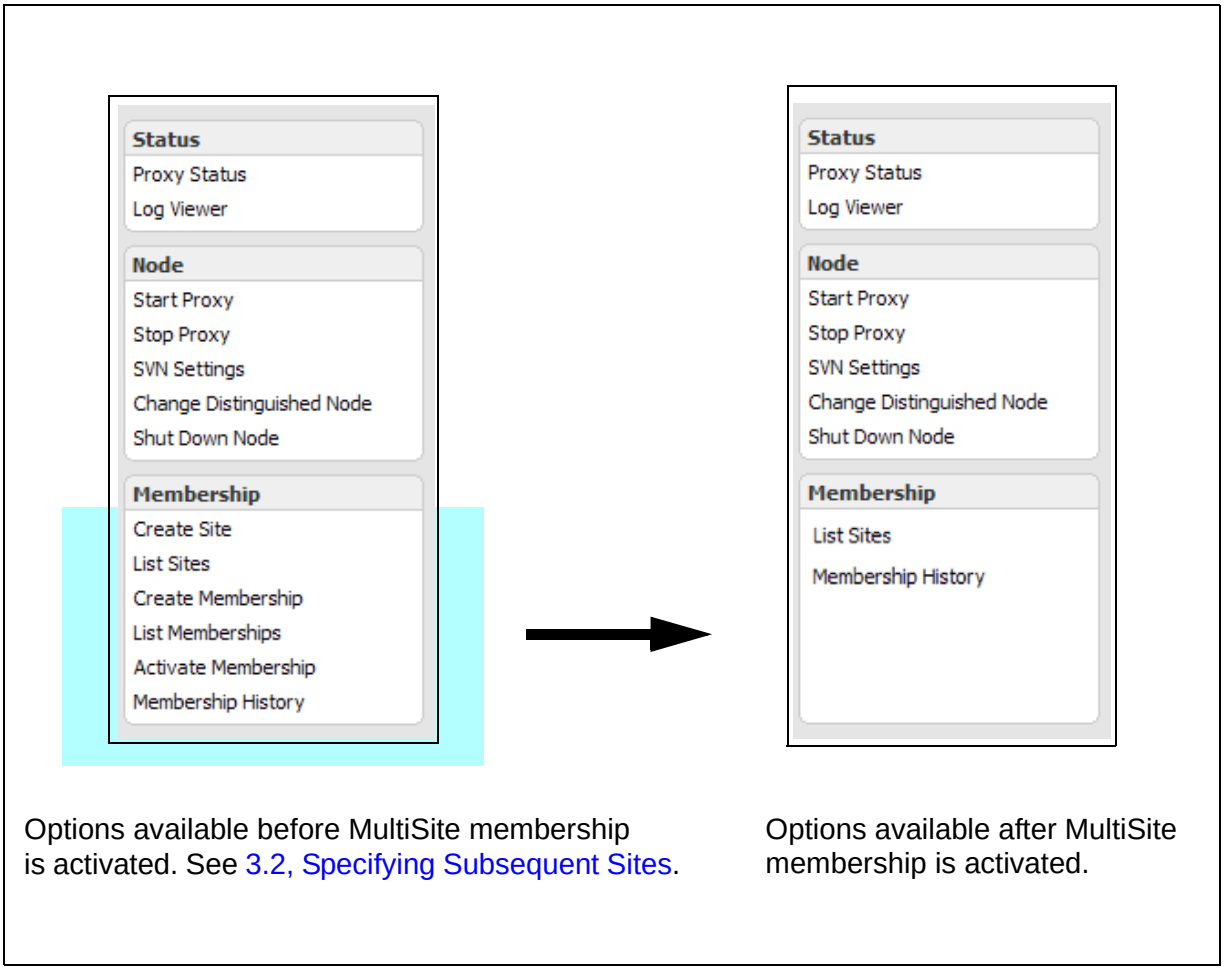


Test Email

- From — Enter the sender address.
- To — Enter in another address to send a test email.
- Subject — Name a subject.
- Message — Enter test text.
- **Send Test Email** — Click to send the test email.
- **Edit Email Settings** — Click to edit the email settings.

Change Distinguished Node	Change the distinguished node. Verify with Proxy Status. See section 1.1.2, Replication Example.
Current Distinguished Node	This names the current distinguished node.
New Distinguished Node	Select another site from the replication group.
Assign Selected Node	Click this after selecting another distinguished node.
Shut Down Node	This command shuts down WANdisco at this site.

There are certain commands available while you are establishing the membership to the replication group. Once you have activated the membership, most of the commands are no longer offered. For information on creating a membership, see 3.2, Specifying Subsequent Sites.

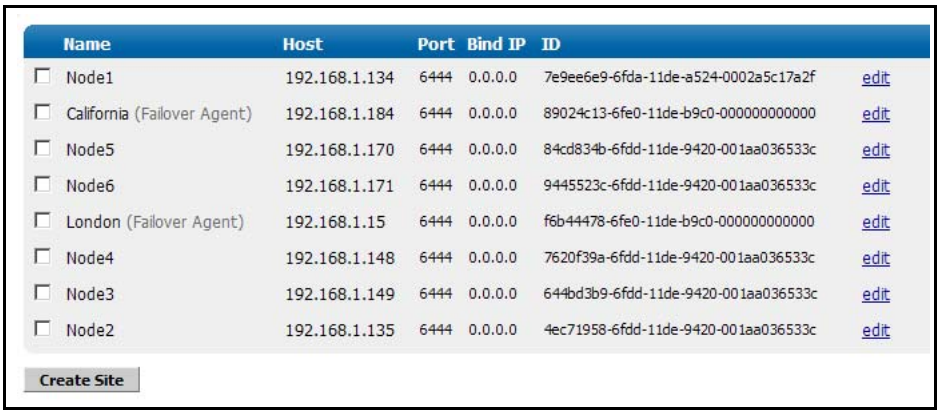


Options available before MultiSite membership is activated. See 3.2, Specifying Subsequent Sites.

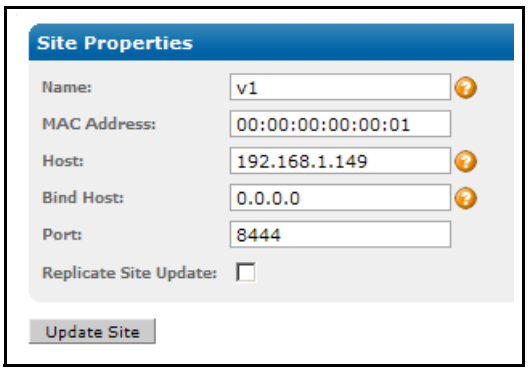
Options available after MultiSite membership is activated.

List Sites

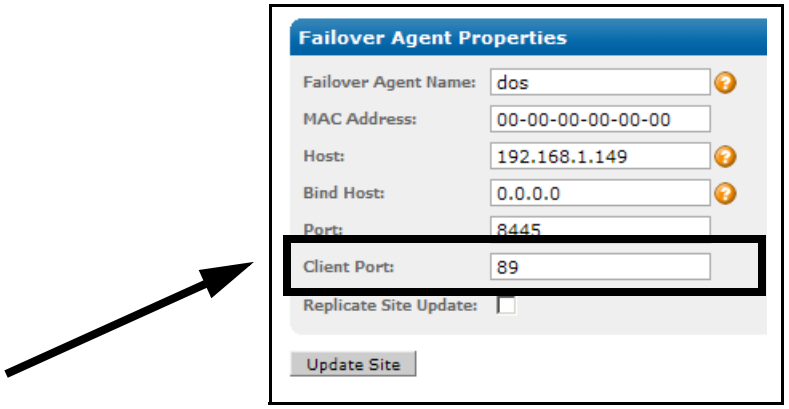
Shows all the sites in the replication group. Any Failover Agents are noted.



You can edit the name, MAC address, host, bind host and replication port for each site. Click **edit**. You must restart each site for the change to take effect.



For Failover Agents, you can also change the Subversion client port.



Membership History

gives a table of past membership activity

4.5 Failover Agent Tabs

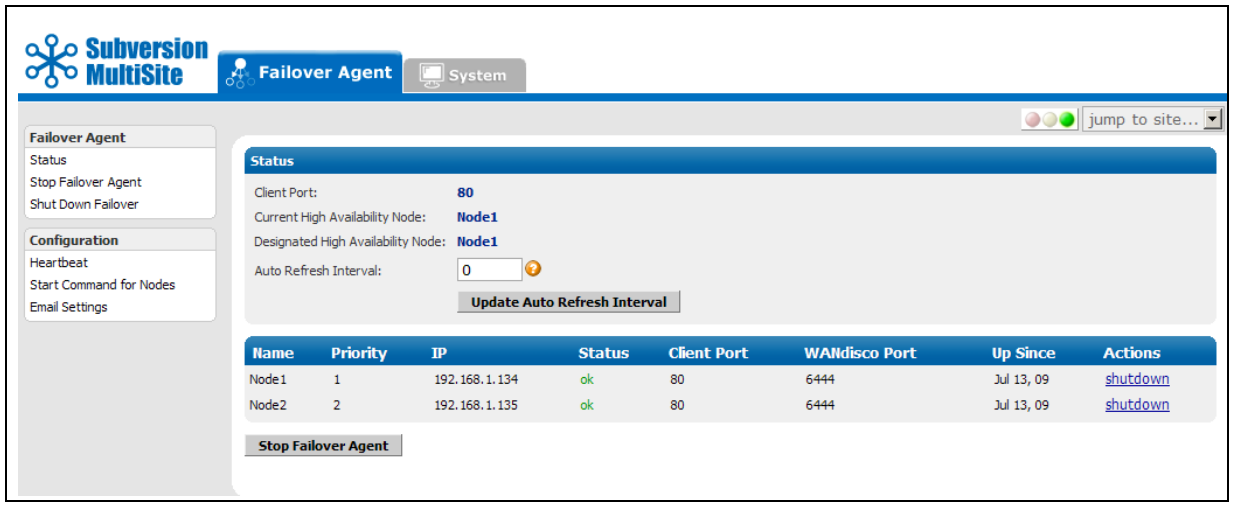
The Failover Agent's Admin Console has two tabs: The Failover Agent tab and the System tab. The Failover Agent tab displays only those nodes within its High Availability sub group.

4.5.1 Failover Agent Tab

The Failover Agent tab displays the status of other nodes in its HA sub group, gives information on the current active primary, and offers several commands in the left menu.

The status of the High Availability sub group displays in the upper right. One of three circles pulses, giving the current status: green signifies all nodes are up, yellow signifies at least one node is not responding, and red signifies that all nodes are not responding.

The Status section names the Current Active Primary (the node listening to the Subversion client through the Failover Agent) and the Designated Primary (priority level 1). These are the same node unless failover has occurred. The Auto Refresh Interval offers you the ability to automatically update the page. The default is 0.



4.5.1.1 Starting and Stopping HA Sub Group Nodes

This is where you start and stop the HA sub group nodes.

NOTE:

If you shut down the current node, you trigger a failover. Shutting down any other node in the HA sub group does not result in a failover.

A stand-alone High Availability sub group of three or more nodes has a majority response quorum. If you shut down a majority of nodes in your HA group, replication stops, and can only continue when a majority of the HA nodes are running.

Each node is listed by name, priority order, IP address, current status (**ok** and **not responding**), client port (in use only by the current active primary), replication port, date of last start up, and action (**shutdown** or **start**).

4.5.1.2 Left Side Menu

High Availability

Status	This command displays the HA sub group's status in the tab's main area.
Stop Failover Agent	This stops Subversion access to Subversion clients. You must confirm your action.
Shutdown Failover	This shuts down the Failover Agent. Restart the Failover Agent by typing `<product directory>/svn-failover/bin/failoveragent`.

Config

Heartbeat	For a complete discussion on the heartbeat, see 1.2.1, Failover and the Heartbeat.
Missing Heartbeat Count	The number of missing heartbeat responses the FA gets before marking a node as unavailable.
Interval	The time between queries from the FA to the nodes. You can specify a different number for each node.
Connection Timeout	The time to wait before assuming the query has failed. You can specify a different number for each node.
Save All	Save changes for all nodes.
Start Command	Enter in commands to start the sub group nodes.
SSH Command	Enter syntax for the SSH command.
Start Command	Enter syntax to start the node. For Windows, in the Start Command field, enter `perl -x -S <pathname>\svn-ha\bin\svnreplicator`
Save All	Save changes for all nodes.
Email Settings	The Failover Agent sends emails when: a) current active primary dies, triggering failover; b) priority 1 node comes back online; c) in a two node group, the second node dies, triggering backup exclusion protocol. See Email Settings on Proxy tab.

Primary

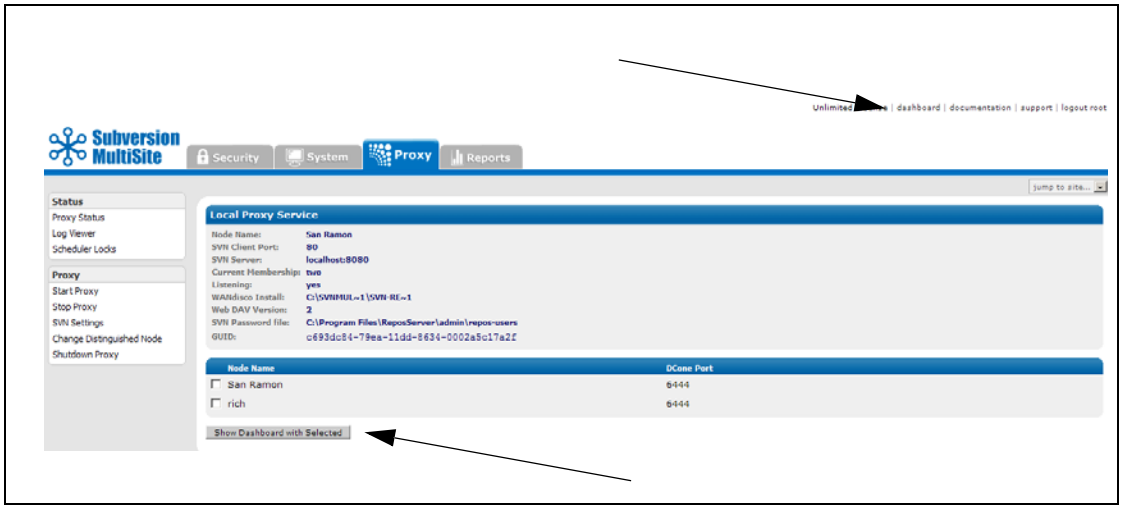
Start Primary	This command only appears if the current acting primary is off. Start the current acting primary with this feature if you programmed the Start command on this tab. Otherwise, use the command line `<prod directory>/svn-replicator/bin/svnreplicator`

4.5.2 The System Tab

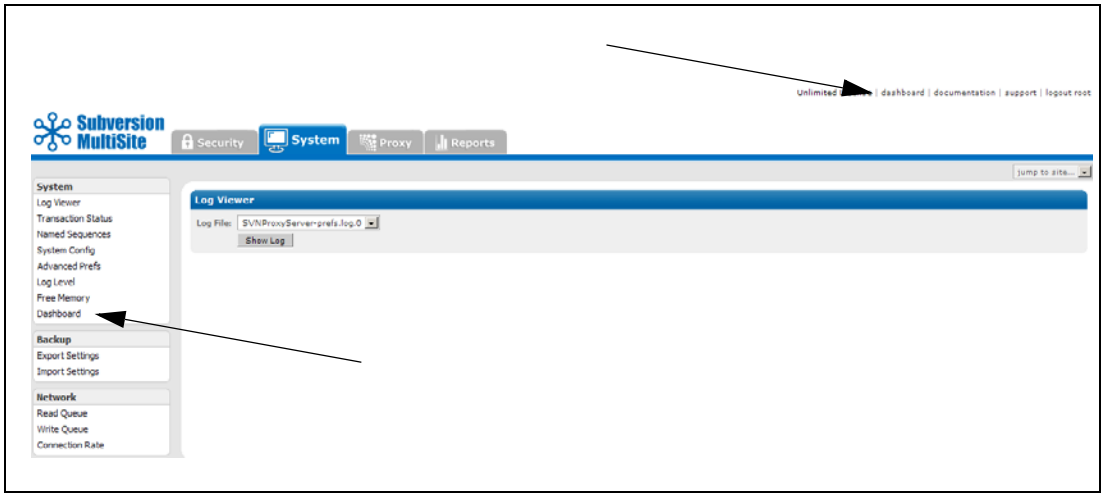
The commands on this tab are the same as on a node's System tab. See 4.3, The System Tab.

4.6 The Dashboard

There are two ways to get to the Dashboard from the Proxy tab.

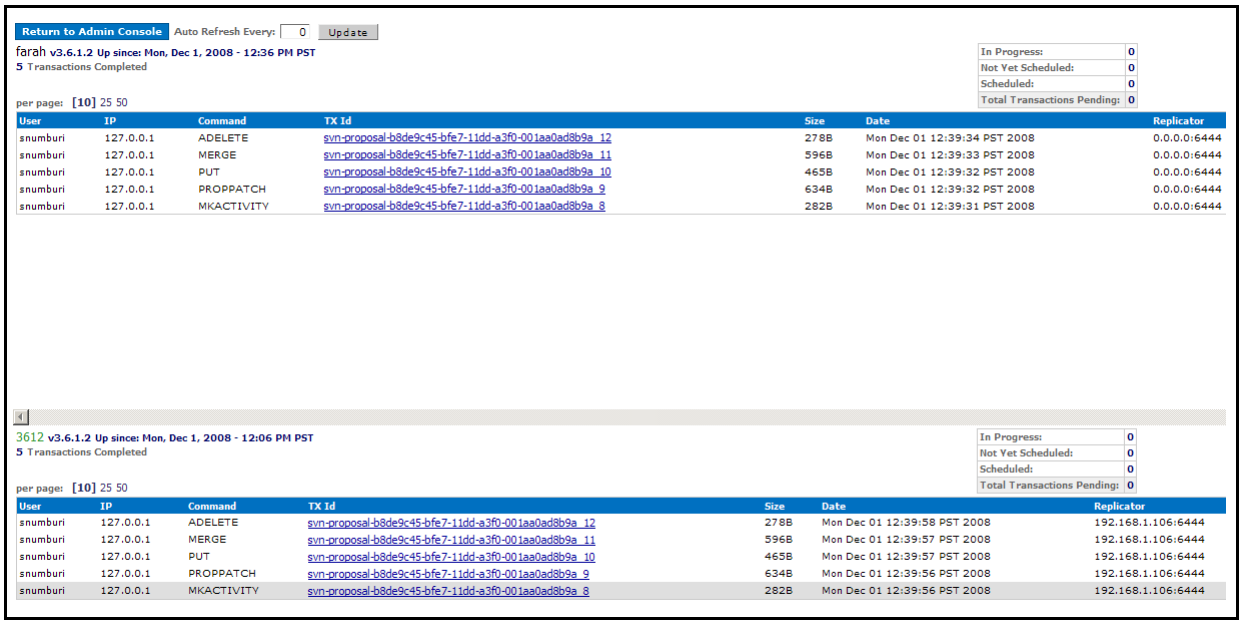


And there are two ways to get to the Dashboard from the System tab.



The Dashboard shows each site's transactions. As soon as MultiSite receives a Subversion transaction request, that transaction joins the replication group's queue. There is one queue for the replication group. Pending transactions are not displayed, but are counted in the **Total Transactions Pending** box in the upper right.

When MultiSite finishes processing a transaction, the transaction displays in the **TX Id** field. The **Replicator** field displays the site where the transaction originated. For example, 0.0.0.0:6444 means the local site originated the transaction, while another IP address identifies that site as the transaction's originator.



4.6.1 Pending Transactions

There are three statuses of replicated transactions before they are committed to Subversion.

- **Not Yet Scheduled** - these transactions are in the queue
- **Scheduled** - these transactions are approved by the quorum and are waiting to be executed
- **In Progress** - these transactions are in the process of being completed

The **Total Transactions Pending** lists the total number of transactions in all statuses.

4.6.2 Completed Transactions

Once a transaction is completed, the Dashboard lists it in the transaction list, and that transaction is no longer considered in the **In Progress** count. The completed transaction joins the count in the **Transaction Completed** display. This display keeps count of transactions since replication began.

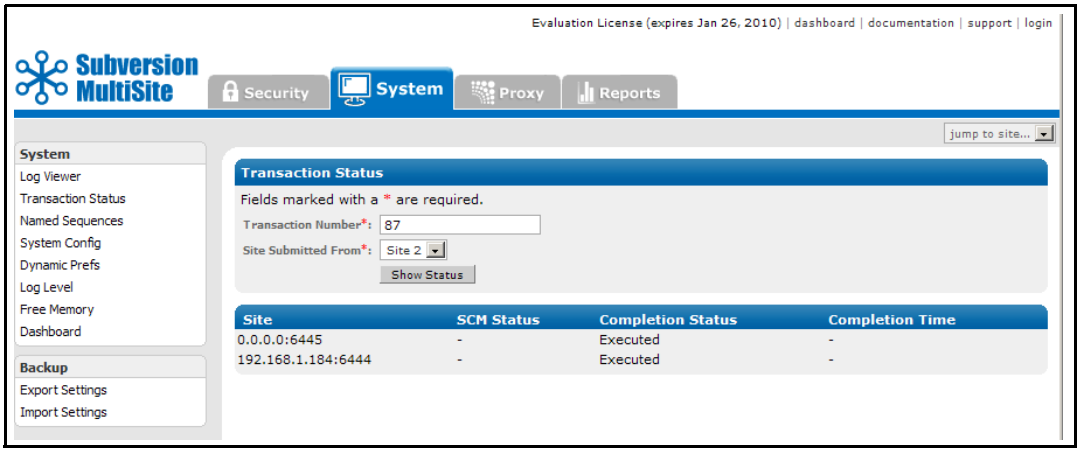
4.6.3 Refreshing the Dashboard Display

The Dashboard by default does not refresh. As MultiSite is completing many transactions, a display that refreshes often would be very confusing to read. While the order of completed transactions is the same at all nodes, the real time of when a transaction is posted may vary from node to node. To refresh the Dashboard, click **Update**. You can also set the Dashboard to refresh automatically by entering a number in seconds in the field.

4.6.4 Transaction Status

To see details on a transaction's status, click on the transaction in the **Tx Id** column.

That transaction appears in the Transaction Status page (on the System tab). You can see information about that transaction.



5 Managing Users

This chapter provides information on setting up users for MultiSite. You can create users, delete users, and search users by several criteria. You should be familiar with the Admin Console, described in Chapter 4, Using the Admin Console.

If you have an existing LDAP or NIS database, you can integrate it with WANdisco. WANdisco offers a free, unsupported plug-in. See the *LDAP Plug-in* document at http://www.wandisco.com/php/products_documentation.php. The document contains a download link for the plug-in.

5.1 Is WANdisco Controlling the Password File?

Upon WANdisco installation, if you selected to have WANdisco control the Subversion password file, any user you enter in WANdisco can use Subversion.

If WANdisco is not managing the Subversion password file, and a user is entered in WANdisco but has not been registered in Subversion, and he or she tries to access Subversion, they would see an `Access Denied` error message in their client.

You can check if WANdisco is managing the password file by looking at the **SVN Password file** value on the Proxy tab. If a path is listed, WANdisco is managing the password file. Otherwise, you see a `not managed by WANdisco` message.

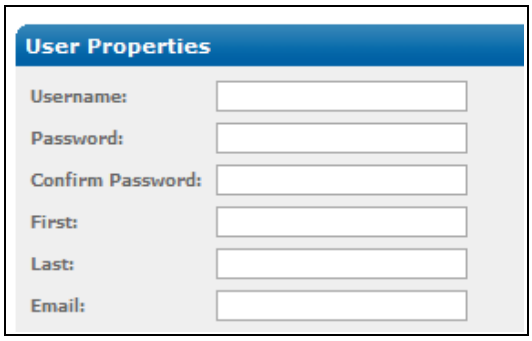
If you selected the wrong choice during installation, you can still have WANdisco manage the password file. In prefs.xml, change the value for the `svnautoupdate` element from `false` to `true`, and specify the absolute pathname to the password file.

Repeat this change at each site. See the procedure 6.6, Changing a prefs.xml File.

```
<SVNProxy>
    ...
    <svnautoupdate>true</svnautoupdate>
    <svnpasswdfile><C:\svn-repository\dav-auth</svnpasswdfile>
    ...
</SVNProxy>
```

5.2 Creating or Deleting Users

To add a new user, click on **Create User** in the Security tab. Specify a (Subversion) username.



Enter the password, and the user's names. The email address is optional.

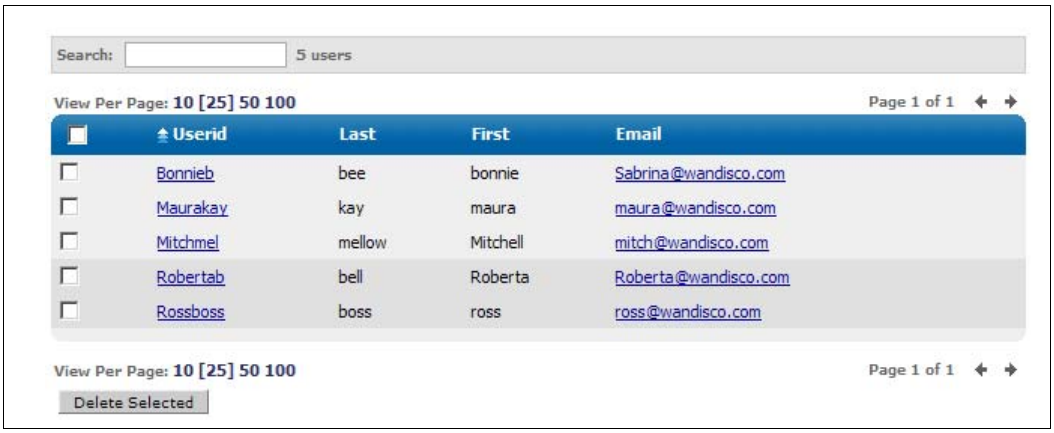
To remove users, click **List Users**. Select the users you want to delete with the checkbox on the left and click **Delete Selected**.

5.3 Listing and Searching for Users

To get a list of all the registered users, click on the **List Users** command. The User List page shows all users by default. The page size is set to show 25 users per page, but you can change that by selecting **View Per Page** on top of the user list. Arrows at the right corner allow you to scroll to the next or previous page.

Use the **Search** box to find users. Begin typing a user's first or last name, and an incremental search starts. Return to the full list by clearing the **Search** box.

All the columns in the user list are enabled for sorting. Clicking on the column header lets you sort in ascending or descending order. The sortable columns include: Userid, last name, first name, and email.



You can click on the user id hyper link to edit the user. You can also delete as many users as you like. Delete all users by checking the checkbox in the table header, and then click the **Delete Selected** button.

5.4 Importing Users

You can import an existing list of users with the **Import Users** command. The import file must be a comma delimited text file, of the format `userid,lastname,firstname,email`.

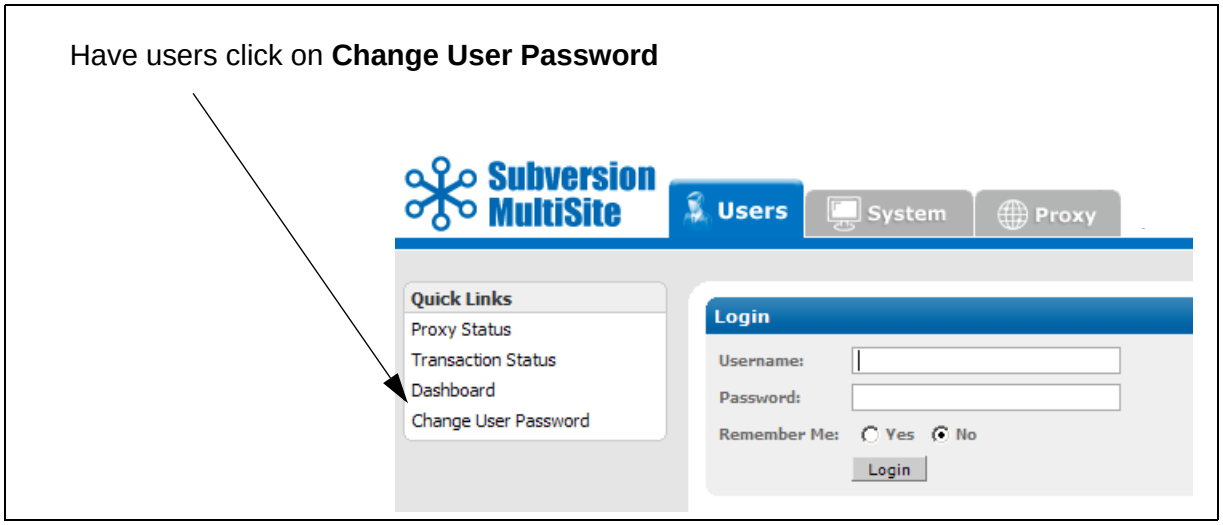
If WANdisco is controlling the Subversion password file, user passwords are changed to user email addresses upon importation. WANdisco recommends notifying users to change their Subversion password, as described in the next section.

5.4.1 Having Subversion Users Change Their Passwords

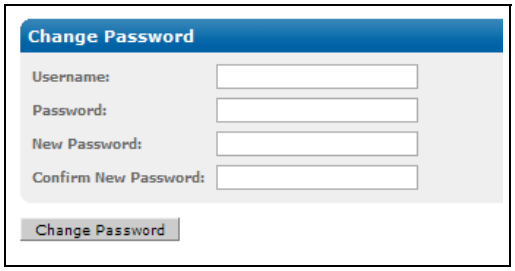
Use this only for users imported with the Import Users command. Importation changes users passwords to email addresses. Users can change Subversion passwords in WANdisco without logging in to WANdisco. Have the users go to

```
http://localhost:6444/
```

The Admin Console appears. Have the users click on **Change User Password**.



The Change Password box appears. Users can enter their Subversion username, and their password (which is now their email address). Have them enter a new password and confirm it, then click **Change Password**. The users have successfully changed their passwords.



6 Procedures

6.1 Temporarily Disabling User Access to Subversion

NOTE:

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

6.1.1 Temporarily Disabling Subversion Access At Selected Sites

You can stop transactions at one or more sites. For a discussion of stopping one, but not all, sites, see 1.1.3, WANdisco is Listening.

NOTE:

Do not use this procedure if you want to do a synchronized stop. For a synchronized stop, see 6.7, 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 site you want to stop.

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

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

6.1.2 Temporarily Disabling Subversion Access At All Sites

To stop all sites at once, you do a synchronized stop. See 6.7, Performing a Synchronized Stop.

6.2 Establishing a Baseline for Replication

Before starting WANdisco, you should ensure that all sites 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 sites, 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 sites, for example, if, prior to deploying WANdisco, you were using a master-slave replication solution such as svnup, choose the **Synchronize** procedure.

6.2.1 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 sites. If you conclude that it takes too long, you will want to ship the repository to the remote sites 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 site 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 sites 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).

6.2.2 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.

6.2.3 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 sites 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
```

6.2.4 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 sites before using WANdisco. Instead, you can follow the procedure below.

Using WANdisco before the baseline is available at all sites

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

Step 2 When choosing a quorum, ensure that the sites 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 site 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 sites 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 site 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 site. Instead, you should `init` the repository at one site, and copy the empty repository to other sites.

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 sites. The best way to ensure this is to copy the repository from one site to the others.

6.3 Finding the Last Committed Transaction

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

Go to any site's Dashboard. Type

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

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

6.4 Adding a Site to the Replication Group

You can add a site to an existing replication group. Notify WANdisco of the IP addresses of the new sites, and WANdisco sends you a new installation file and license key file.

Follow the instructions in 3.5, Installing Upgrades.

Note that you run the installer at the initial install site, which auto-populates previously filled out fields. You probably want to choose to not have WANdisco package the repository, and just copy it for the new sites.

As you bring each previously existing site out of quarantine, you can release it to users. There should be minimal downtime; the longest task is copying the repository to the new sites.

6.5 Removing a Site from the Replication Group

You can remove a site or sites from an existing replication group. Follow the instructions in 3.5, Installing Upgrades, but just eliminate the site from the group.

Note that you run the installer at the initial install site, which auto-populates previously filled out fields. You should choose to have WANdisco not package the repository, since the repositories are already in sync.

6.6 Changing a prefs.xml File

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

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

6.6.1 Changing One Site's prefs.xml File

Step 1 Shut down the site or sites 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 site or sites you stopped. At the command line, type

```
svnreplicator
```

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

6.6.2 Changing All Sites' prefs.xml File

Step 1 Perform a synchronized stop. See 6.7, Performing a Synchronized Stop. That procedure includes resuming WANdisco.

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

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

6.7 Performing a Synchronized Stop

NOTES:

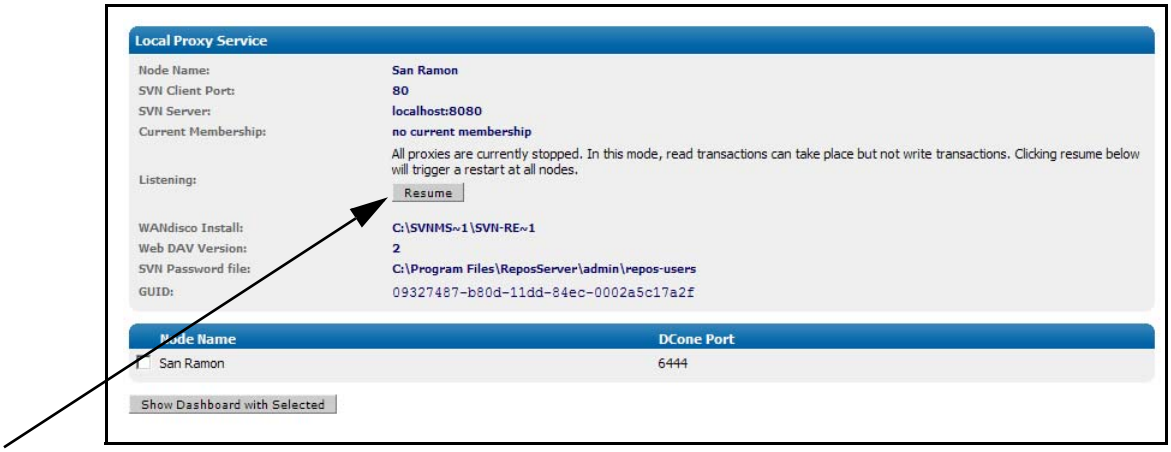
A synchronized stop completes only when all sites are available. If one or more sites are unavailable, the process suspends until they are available.

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 sites 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**.

6.8 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
```

6.9 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 site'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 sites, 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 6.7, Performing a Synchronized Stop.

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

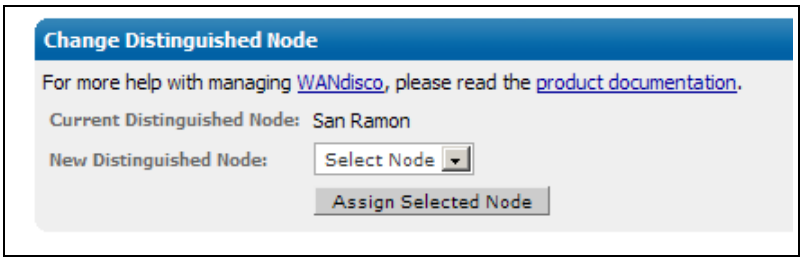
Step 6 Restart the group. Click **Resume** on the Proxy tab.

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.

6.10 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.

6.11 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 site within the replication group. Alternatively, you could use the time of day to fire the e-mail alerts from a specific site. 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 sites but commit at other sites. The `pre-commit` trigger behavior at each site 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.

6.12 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 site is also the distinguished node, the quorum check will always succeed irrespective of network connectivity to other sites.

6.13 Changing WANdisco's Root Login

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

You can change the login for any site. Each site can have its own login, however if you do change it, ensure that all site administrators throughout all sites know it. If four sites each had their own

logins, then each administrator would see the other sites in the Dashboard, and could not access the other sites' Admin Consoles without entering those sites' passwords.

NOTE:

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

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

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

See the procedure 6.6, Changing a prefs.xml File.

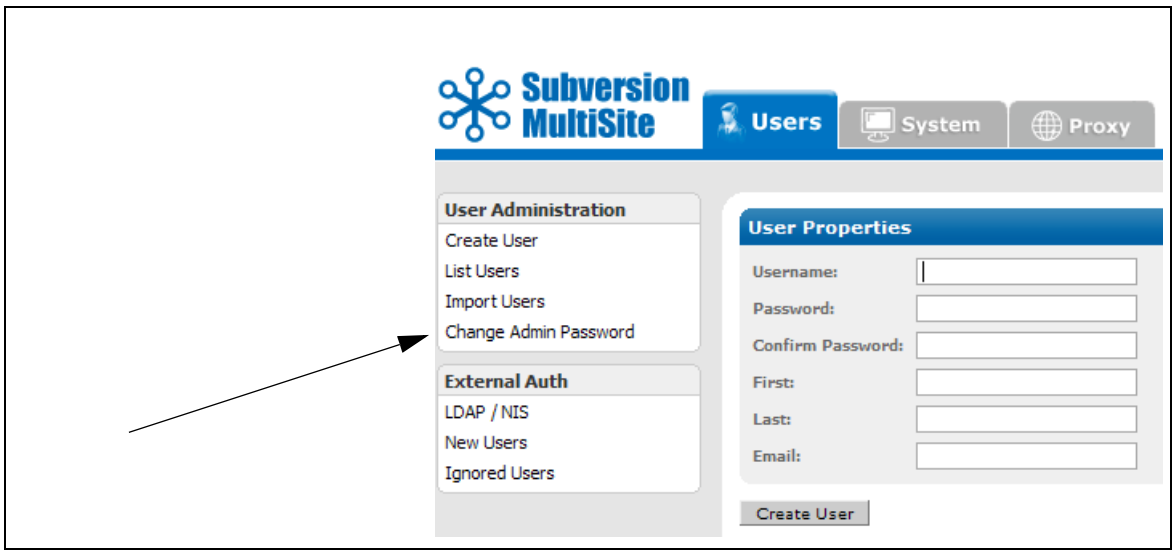
6.13.1 Changing the Password

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 site only. The password does not get replicated.

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



6.14 Setting Up a 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 Chapter 4, Using the Admin Console.

The pre-replication hook currently implemented is the equivalent of the SVN DAV pre-commit hook. 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.

6.14.1 Configuration

Modify your prefs.xml file at all sites to contain the following tags. See 6.6, 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>
```

6.14.2 Other Configuration

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

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

6.14.3 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>
```

7 Procedures for Stand-Alone Two-Node HA Groups

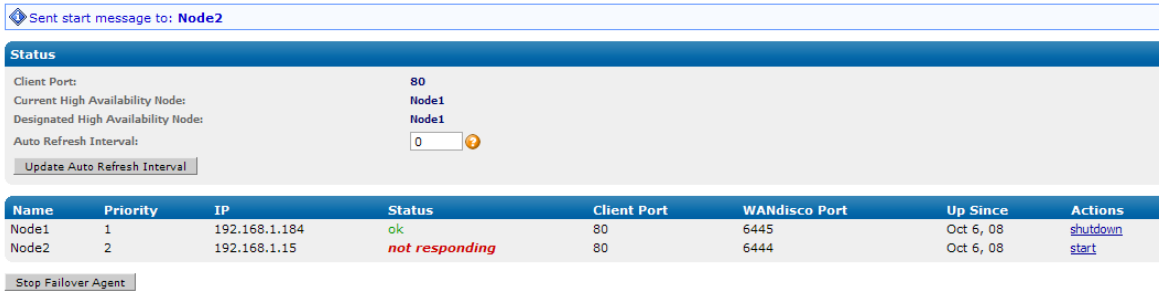
This section is for stand-alone HA sub groups that have just two nodes, and there are no other nodes in the replication group.

7.1 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.

7.2 Recovering from Backup Node Failure

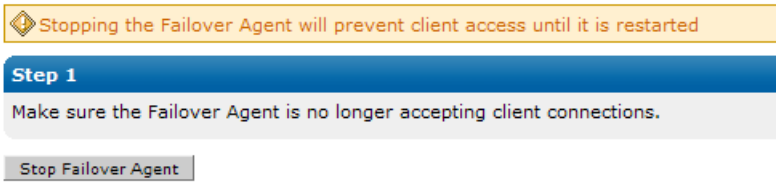
When the backup node fails, there are a few more steps to complete.



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**.



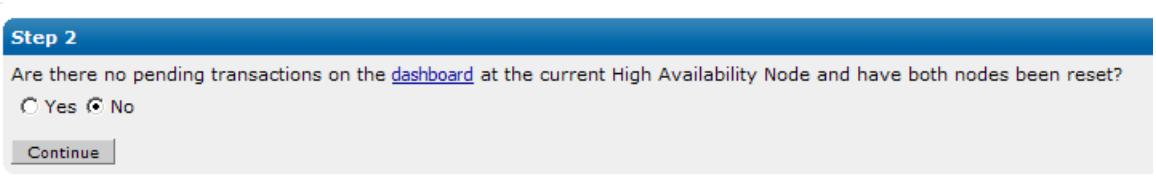
Step 2 Make sure there are no pending transactions on the dashboard, and reset both nodes. Reset the nodes by typing in the `/bin` directory

```
reset
```

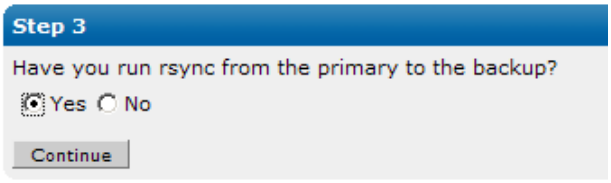
On Windows, type

```
perl reset
```

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



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

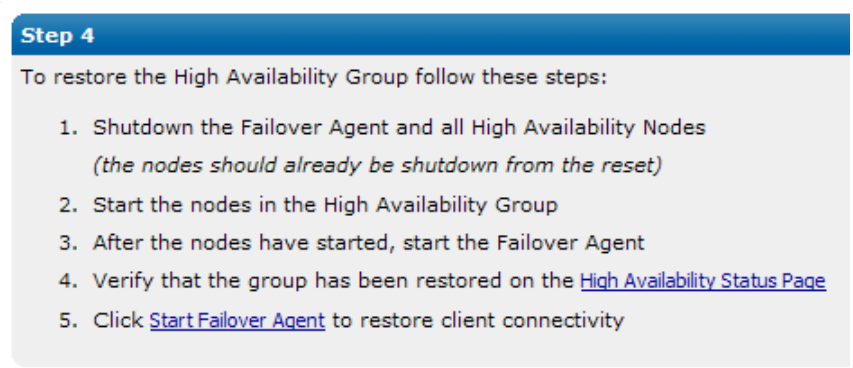


Step 4 Start the nodes. In the /bin directory, type

```
svnreplicator
```

Start the Failover Agent. Type

```
failoveragent
```



Your HA group is restored.

8 Replicator Management

8.1 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
}
```

8.2 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.

8.3 Rotating the Distinguished Node

For replication groups with a Singleton quorum (or Multiple quorum with a even number of sites), you can schedule the Distinguished Node to change, based on physical location around the world.

For example, say you have Subversion MultiSite in San Jose, CA, in Madrid, and in New Zealand. Since being the distinguished node offers the fastest response time, you would want each of those locations to take advantage of the time they could each be the distinguished node.

To do that, you would edit the prefs.xml file **at only one location**. Use the local time for that location, and figure out the times the distinguished node would change.

Leave the existing `<Quorum>` tag alone, and add two new tags, `<Quorum>` and `<Schedule>` under the `<AgreementManagerList>` tag. For example,

```
<AgreementManagerList>
  <AgreementManager name="52ec6735-ce20-11d9-8e57-00065be02953">
    <DisplayName>svn-am</DisplayName>
    <Quorum>
      <Schedule>
        <at name="09:00 AM">
          <!-- San Jose Site --> <DistinguishedNode>fb7723de-ce1e-11d9-
ae57-00065be02953</DistinguishedNode>
        </at>
        <at name="08:30 PM">
          <!-- Madrid Site --> <DistinguishedNode>659a768d-ce1f-11d9-
aeec-00065be02953</DistinguishedNode>
        </at>
        <at name="03:00 AM">
          <!-- New Zealand Site --> <DistinguishedNode>3fae40f3-ce20-
11d9-8e6a-00065be02953</DistinguishedNode>
        </at>
      </Schedule>
    </Quorum>
  </AgreementManager>
</AgreementManagerList>
```

The time syntax is `hh:mm:ss: AM|PM`. Seconds are optional.

NOTES:

If you do rotate your distinguished node, the other sites' Preferences files do not reflect the rotation. You must rely on the Admin Console (available at all sites) to display the current distinguished node (the site's name displays in green).

Make sure that the local MultiSite administrators know about the change.

See 6.6, Changing a prefs.xml File.

8.4 Changing the Quorum Type

Quorum is established when you define a membership. See step 39 in chapter 3, Installation. If you want to change the quorum type from what was initially defined, please call WANdisco support at 887-207-1439.

8.5 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. Set up the email account for each site with the Email Settings, described in Chapter 4, 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 8.1, Setting Replicator to Start Up on System Boot.

8.6 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.

9 Troubleshooting

9.1 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.

9.1.1 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, for Unix

```
svn-replicator/bin/talkback
```

For Windows, type

```
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*.

9.2 General Subversion MultiSite

9.2.1 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.

9.2.2 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 sites 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.

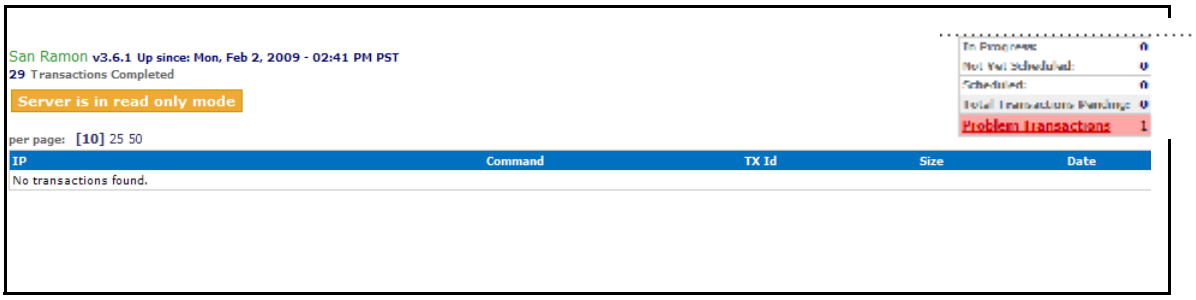
- On the replicator host, tune the TCP stack to have a smaller fuse on the TCP keep alive timer. For example, on Linux, you can specify a value like 1800 (seconds) in `/proc/sys/net/ipv4/tcp_keepalive_time` to reduce the interval from default 2 hours to 30 minutes.

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

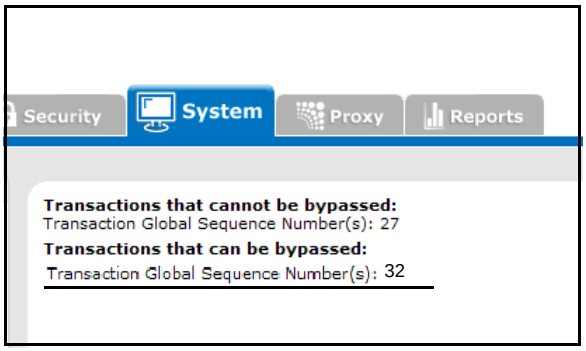
9.2.3 A Site Is In Read-Only Mode

If any sites are in read-only mode,

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



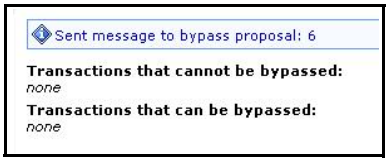
Step 2 Click on the Problem Transactions link. The System tab displays a message, noting the bypassable transactions and the non-bypassable transactions.



Step 3 If there are no bypassable transactions, and you do not know why, see 9.1, 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 site in read-only mode.

9.3 Error Messages

9.3.1 Missing License Key File

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

9.3.2 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 site without giving the replicator any chance of replicating. This can be easily resolved by following the reset procedure outlined in 9.4.1, 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.

9.4 Oops!

9.4.1 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: For Unix, type

```
$ svn-replicator/bin/reset
```

For Windows, type

```
svn-replicator\bin\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.

9.4.2 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.

10 Frequently Asked Questions

10.1 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.

10.2 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.

10.3 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.

10.4 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.

10.5 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 site 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.

10.6 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.

10.7 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 sites. 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 sites.

10.8 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 site/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.

10.9 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 10.8, 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>
```

10.10 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 site, and are not replicated.

10.11 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/.

Appendix A - Installing Java and Perl

You should have already installed Java and Perl at all the sites in your replication group for your trial evaluation. However, any new site you add to the replication group needs Java and Perl installed as well.

Installing Java

Step 1 Install JDK 1.5 and define the `JAVA_HOME` environment variable to point to the directory where the JDK is installed. You can download JDK 1.5 from the URL below.

```
http://java.sun.com/javase/downloads/index_jdk5.jsp
```

Step 2 Add `$JAVA_HOME/bin` to the path and ensure that no other java (JDK or JRE) is on the path.

```
$ which java
/usr/bin/java

$export JAVA_HOME="/usr"
```

or

```
$which java
/export/share/apps/jdk/1.5.0/bin/java

$export JAVA_HOME="/export/share/apps/jdk/1.5.0"
```

Step 3 Ensure the full JDK is installed, not just the JRE. This can be confirmed by running `java -server-version`. If it generates a **not found** error, repeat Steps 1 and 2.

If you find package management problems or conflicts with the JDK version you are downloading (for example, rpm download for Linux), you may want to use the self-extracting download file instead of the rpm (on Linux) package. The self-extracting download easily installs in any directory without any dependency checks.

Installing Perl

Step 1 On UNIX or Cygwin, install perl version 5.6 or greater and ensure that the perl executable is on the system path.

Step 2 On Windows, install ActivePerl version 5.8 or greater and ensure that the perl executable is on the system path. You can download the MSI installer for ActivePerl from the URL below.

`http://activestate.com/Products/Download/Download.plex?id=ActivePerl`.