About
WANdisco Subversion MultiSite

Revision History

REVISION	DATE
1.0	December 2008
2.0	February 2009
3.0	July 2009

Contents

1 About Subversion MultiSite 1

1.1 WANdisco MultiSite Concepts 3

1.1.1 How Replication Works 4

1.1.1.1 Singleton Quorum 4

1.1.1.2 Majority Quorum 4

1.1.1.3 Unanimous Quorum 5

1.1.2 Replication Example 5

1.1.3 WANdisco is Listening 5

1.1.4 Synchronized Stop of All Sites 7

2 About the High Availability Option 8

2.1 Failover and the Heartbeat 9

2.1.1 Actual Failover Occurs Upon Receiving a Client Request 9

2.2 Failover Sequence 10

2.2.1 The Current Node and the Designated Node 10

2.2.1.1 Priority Order 10

2.3 About High Availability within the Replication Group 11

2.4 About the Stand-alone High Availability Group 12

2.5 Stand-Alone Failover Group of Two Replicators 12

2.5.1 What Happens If the First (Designated) Node Fails 12

2.5.2 What Happens If the Second Node Fails 13

3 About the Access Control Option 14

3.1 Access Control Concepts and Terms 14

3.2 Users, Roles, and Groups 14

3.3 Access Control Lists 15

3.3.1 Reports 15

3.4 Sequence For Implementing Access Control 16

3.5 Perl Style Regular Expressions and Syntax 16

4 Replication and Site or Network Failures 17

4.1 Node Failures 17

4.2 Network Failures 17

4.3 Failover Agent Failures 18

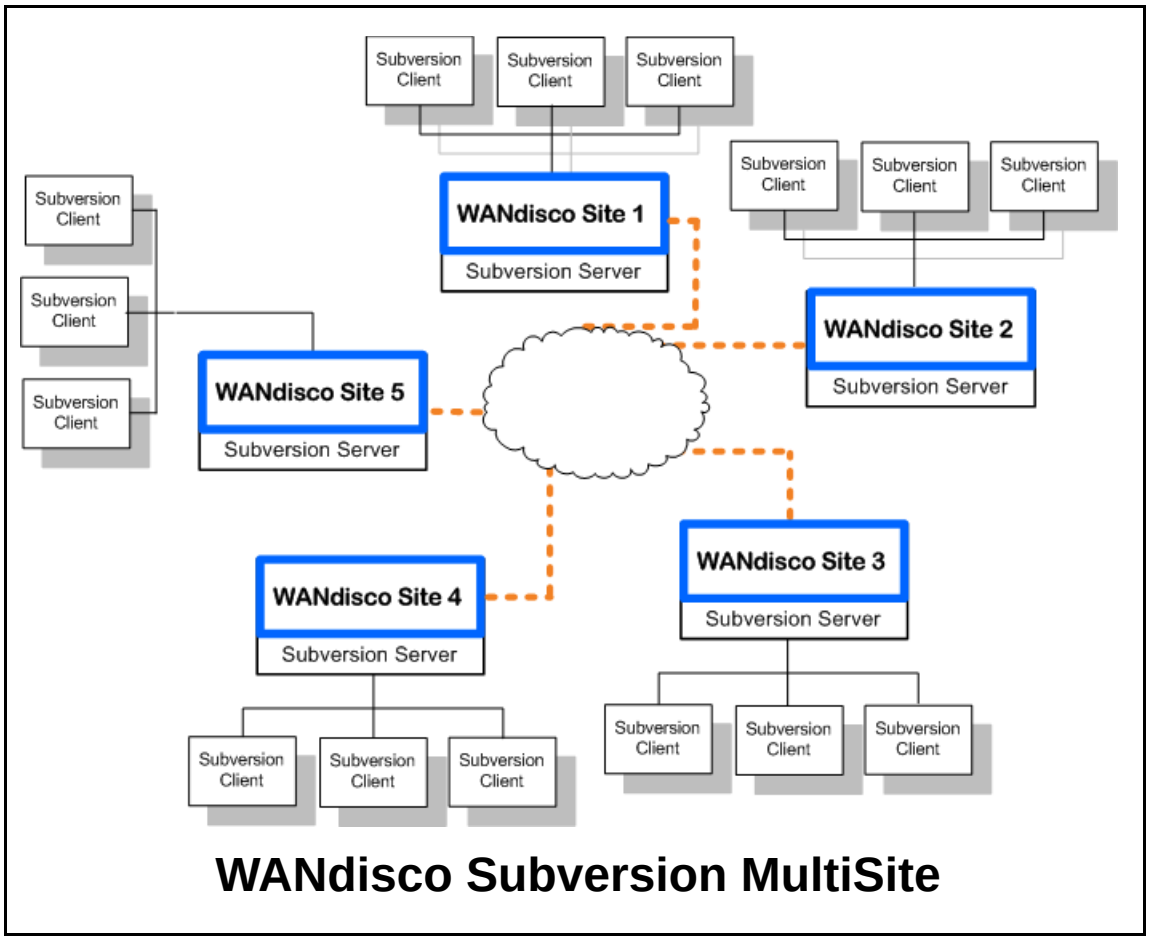
Contents, cont'd

5 Establishing a Replication Baseline 19
6 Terms 20

1 About Subversion MultiSite

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."

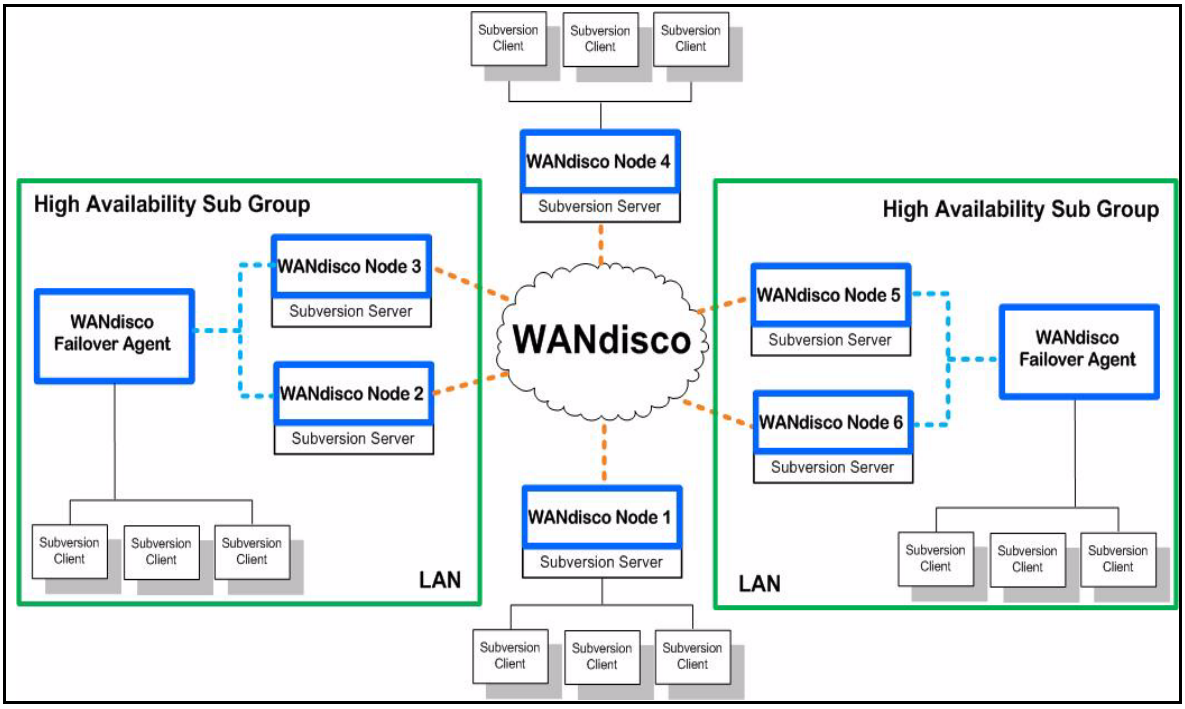
Replication implicitly ensures that each replica acts as a hot backup to every other replica. If a local server does experience a problem that takes it offline, local users experience a disruption in service, while the rest of the replication group continues unaffected. The following illustration shows 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 is served by a Failover Agent, a stateless proxy intermediating between SVN clients and members of that sub group.



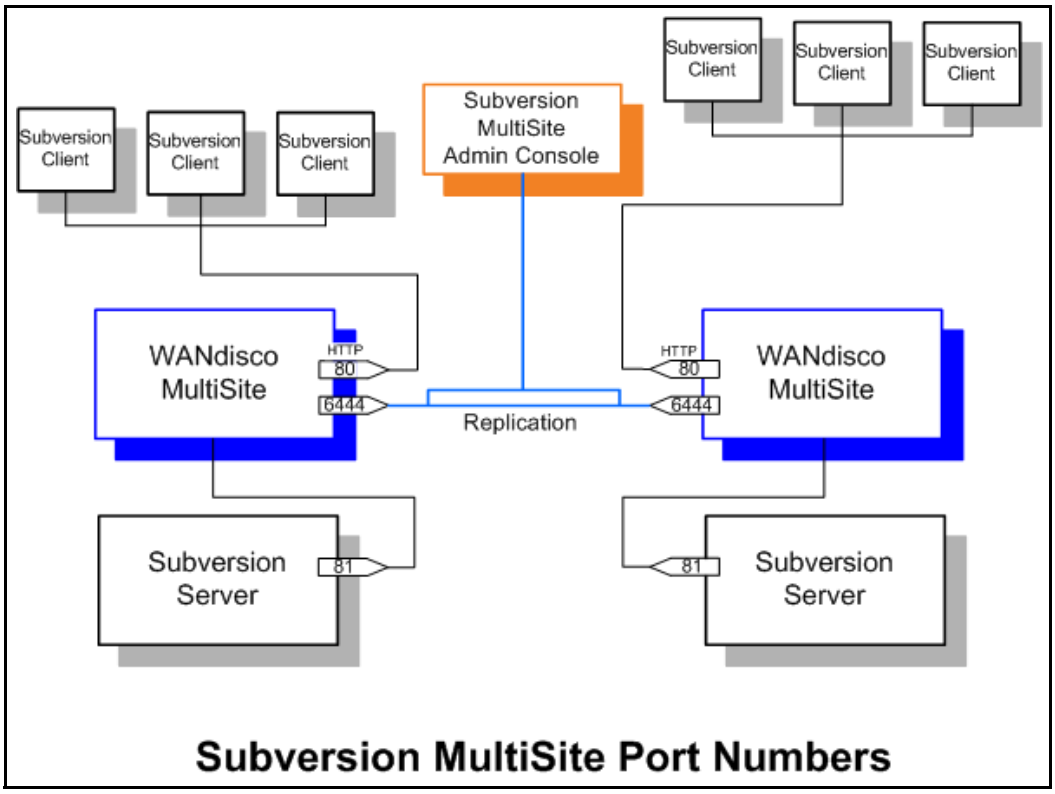
WANdisco also offers an Access Control solution that enables you to implement comprehensive security features, including:

- Setting up Access Control Lists (ACLs) across multiple SVNROOTs and modules
- Role-based ACLs
- Hierarchical Groups and Privileges
- Roles for separate actions:
  - List
  - Read
  - Write
  - Tag
  - Delete
  - Admin (all actions)
- Access control of files, directories, and modules
- Ability to limit access via network masks or IP address
- Full Perl-style regular expression support
- Two default reports: all users, and all groups, and the resources they can access

- Ability to configure more detailed reports with MySQL utility

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 nodes 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 nodes 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 that includes other nodes, you select the quorum type best suited to your configuration. For more information on quorum types for various configurations, see Quorum Recommendations.

1.1.1.1 Singleton Quorum

Singleton Response quorum means that only one of the nodes in the replication group 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 daily rotation of the distinguished node to different nodes around the world, to “follow the sun.” Rotating the distinguished node allows other nodes to take advantage of the quickest response time when they are most active. See Changing the Quorum Type.

1.1.1.2 Majority Quorum

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

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

In a majority quorum, the distinguished node’s role is that of a tie-breaker. For example, say you have four nodes, A, B, C, D. If A and B are talking but can’t reach C and D, and C and D are communicating but can’t reach A and B, both could agree on distinct transactions and the two pairs would diverge. So, only the pair that includes the DN (say it’s A) would achieve agreement. The pair C and D would not be able to agree.

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 Changing the Quorum Type.

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 the Subversion server host where it is received by the Subversion MultiSite replicator.

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

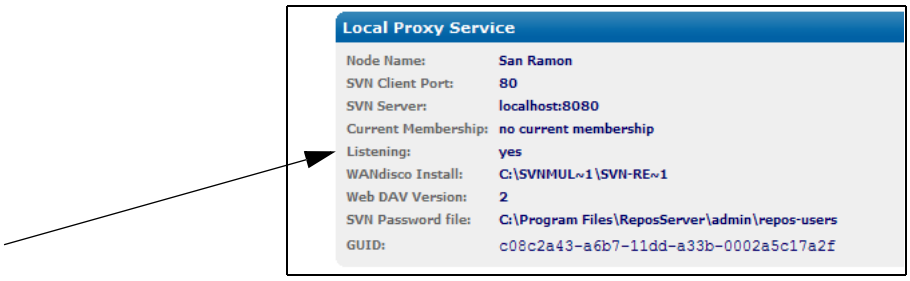
Step 3 After receiving the transaction, each Subversion MultiSite passes the transaction data to its local Subversion server.

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 comlete when Subversion returns an error status. If there is a failure (the replicator is stopped by an administrator, the host server crashes, or Apache hangs or crashes), Subversion MultiSite does not mark the transaction as completed. Upon restart of the replicator, that transaction will be reprocessed and then subsequent transactions will continue to be processed.

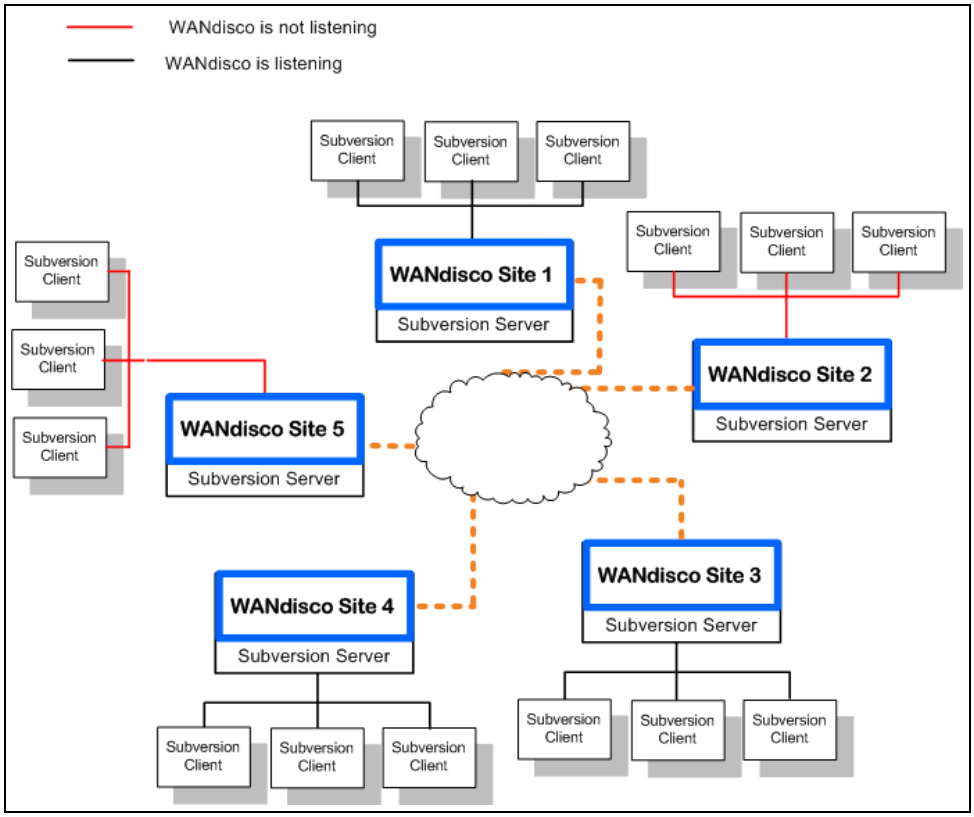
1.1.3 WANdisco is Listening

There is a field in the Admin Console that tells you whether WANdisco is accepting any incoming Subversion client requests. Replicated transactions originating from other nodes will still be processed in this state but local clients will not be able to connect to their Subversion server.



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 Using the Admin Console in Using MultiSite. Issuing the **Stop Proxy** command on a node 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 nodes.) Replication continues, and Sites 2 and 5 are still receiving and processing replicated transactions originating from the other nodes. 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.



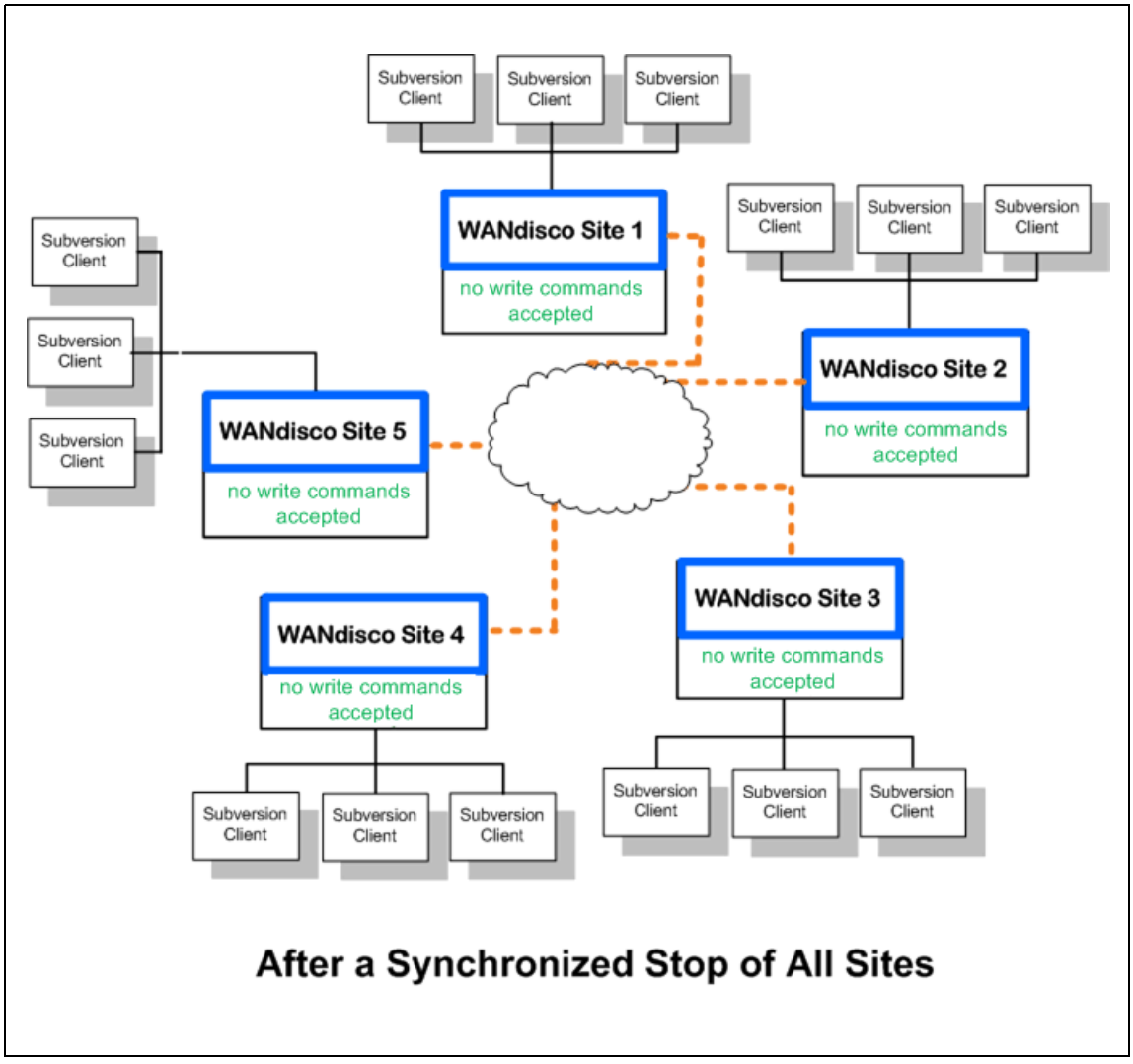
For High Availability sub groups, shutting down the Failover Agent stops WANdisco from accepting local client requests.

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

2 About the High Availability Option

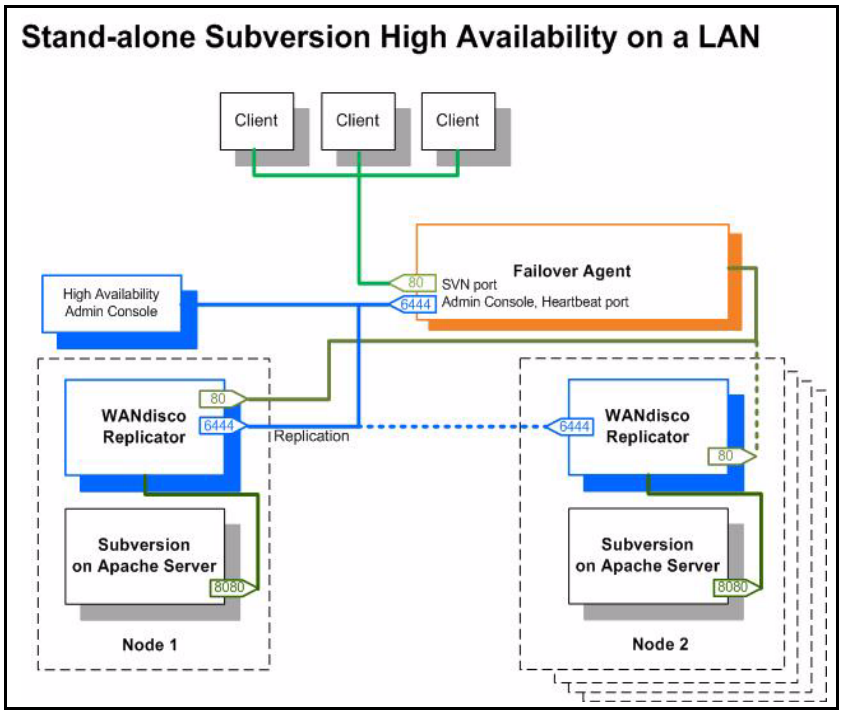
High Availability offers WANdisco customers business continuity, because if one Subversion server fails, the Failover Agent transparently directs users to the next available replica in the High Availability sub group. Each WANdisco Subversion High Availability sub group consists of a Failover Agent and 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.

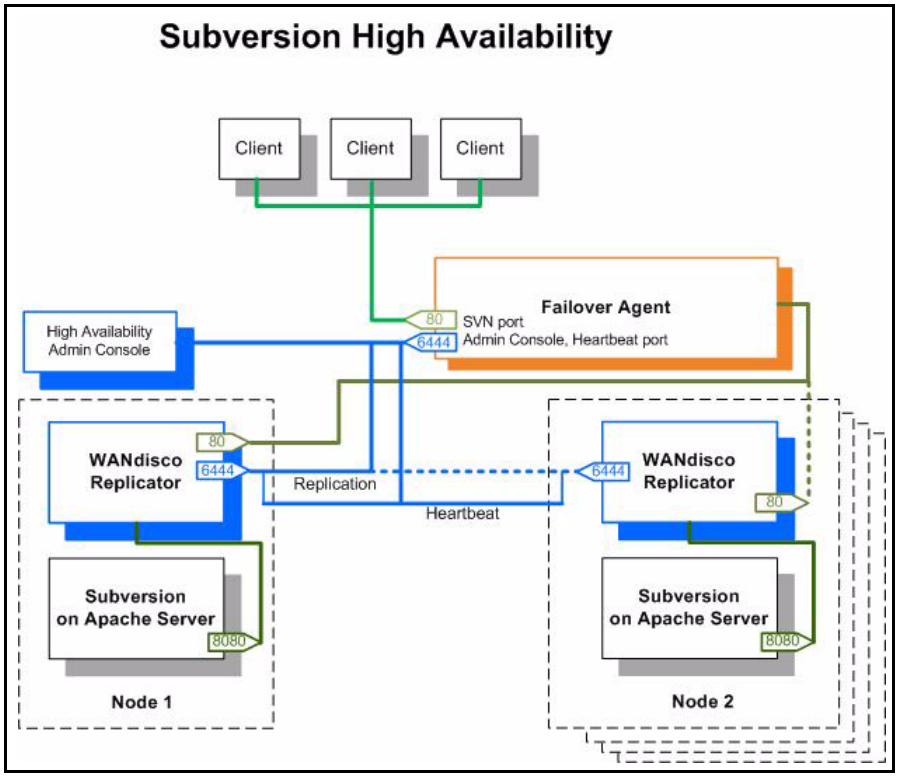
2.1 Failover and the Heartbeat

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.

2.1.1 Actual Failover Occurs Upon Receiving a Client Request

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.



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 group, whether the nodes are within its High Availability sub group or are distributed throughout a MultiSite group.

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).

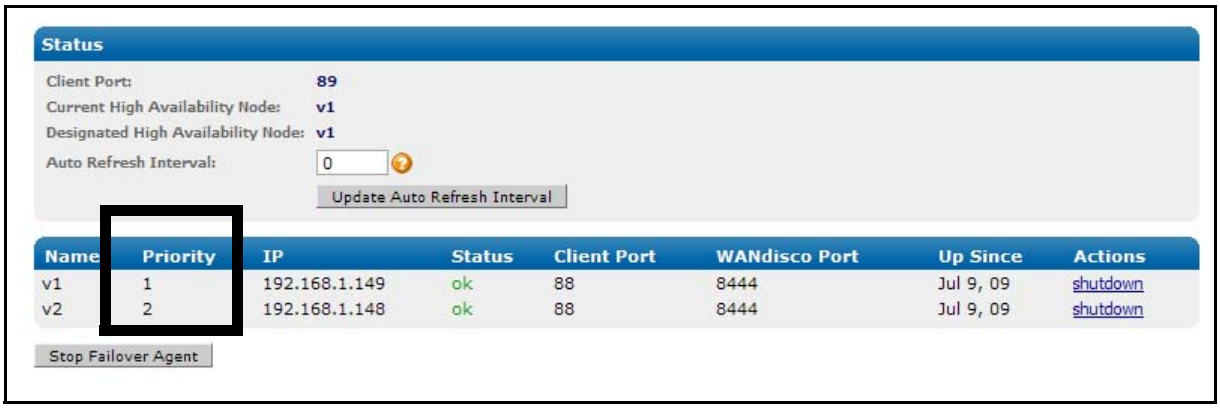
2.2.1.1 Priority Order

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>
...
```

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 nodes 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.

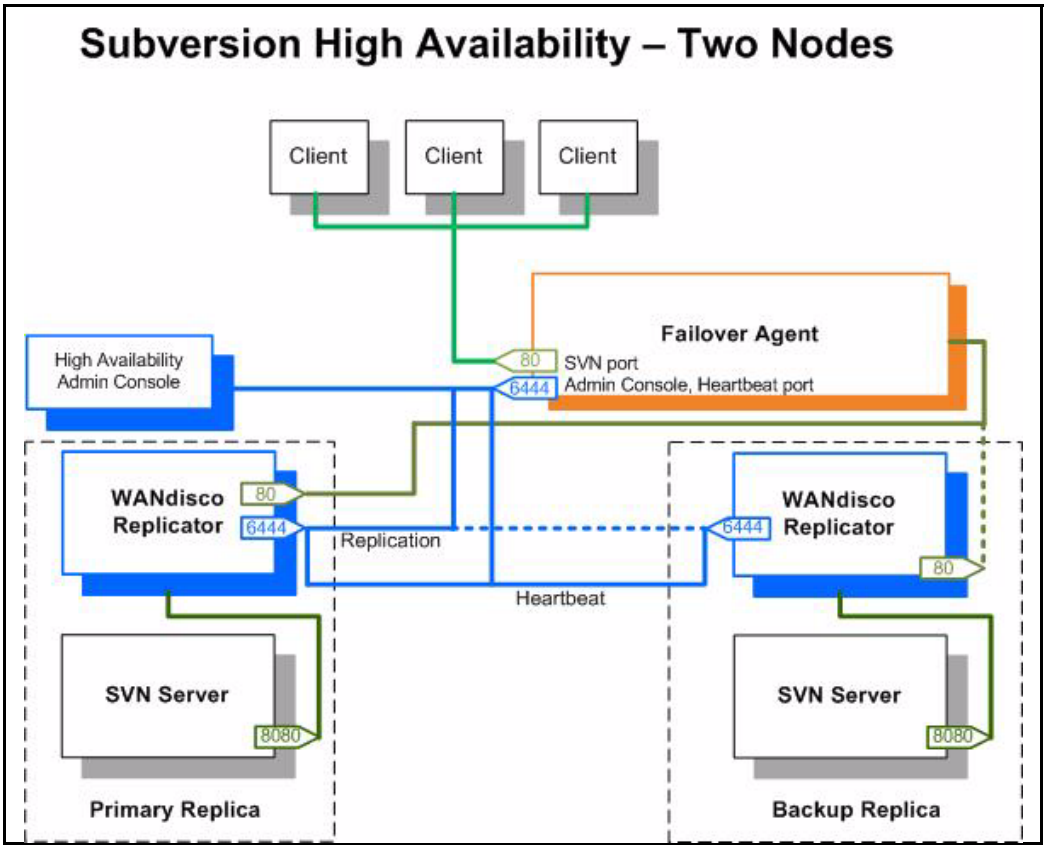
2.4 About the Stand-alone High Availability Group

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

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 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.



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.

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 replication group once it is back up is described in Recovering from Backup Node Failure.

3 About the Access Control Option

3.1 Access Control Concepts and Terms

Here are definitions for commonly used terms in WANdisco Access Control:

Login id	The actual Subversion account name that can be successfully authenticated by Subversion or SSH daemon (if using the ext SSH protocol). The Subversion login id is also the primary key for a user in the Access Control user database.
Principal	Principal can be any valid user or group. After authentication, Access Control maps a login id to a set of rules that include the actual user and all its associated groups and sub-groups.
Resource	Resource is a file, directory, module or the SVNROOT itself. Resource patterns can be specified as Perl-style regular expressions in the ACL. All directory paths should be specified in the slash-terminated form. For example, specify `/a/b/c/`, not `/a/b/c`.
IP Mask	A Perl-style regular expression specifying the Subversion client's IP address. It is used in the ACL to restrict access to a specific client network, subnet or a machine.
Privilege	Privileges are needed by a user to execute specific Subversion commands. WANdisco Access Control supports these privileges: ■List ■Read ■Write ■Copy ■Delete ■Admin

Admin serves as a privilege, a role, and a group. It is meant for a System Administrator who has no restrictions. Do not create ACLs for an Admin user.

3.2 Users, Roles, and Groups

You set up Subversion users by assigning them roles. Each role defines specific privileges. The fundamental permissions for a user are defined by their role. Then, you assign each user to a group, and give that group access to the desired files and directories (referred to as “resources”).

The group level defines a user’s access to the resources. With this structure, you can have multiple projects with complex resource definitions, allowing users to perform tasks associated with their roles’ permissions on those resources.

For example, you create two users: Mary, who has the developer role, and Bob, who has the QA role. Next, you create a group named *project1* and specify which Subversion directories the group has access to (you define the resources). So Mary is allowed to perform write operations and Bob is allowed to perform read and copy operations on those directories.

NOTE:

> Any user you add to Access Control must already exist in the authentication database used by the Subversion server. WANdisco offers an import tool to do some external SQL verifying of user lists, but it is up to the Access Control administrator to control the validity of the users.

3.3 Access Control Lists

For some customers, roles and groups provide enough control. However, you can further refine the granularity of control for the roles, groups and resources by creating Access Control Lists (ACLs), also called rules.

For example, in the group *project1*, if you don't want Mary to have access to one sub-directory in the resources you have defined, you must create an ACL to prohibit her write privilege to that sub-directory.

You can create as many ACLs as you need, however you should have a comprehensive plan to avoid creating conflicting ACLs.

3.3.1 Reports

Access Control offers two reports, Users and Groups. You may want to perform more in depth audits of the ACLs. This requires importing Access Control information into an SQL database.

WANdisco offers an import tool that automatically creates the table schema in that database. The import tool uses standard SQL syntax, and makes use of the system function `FROM_UNIXTIME`. Please ensure your database version supports it. MySQL and Microsoft SQLServer both support the `FROM_UNIXTIME` function.

3.4 Sequence For Implementing Access Control

The general sequence for implementing Access Control is listed below. It is a fairly simple process. Setting up any ACLs can be the most complicated.

Step 1 define roles

Step 2 create groups

Step 3 add users and assign roles

Step 4 set up ACLs

3.5 Perl Style Regular Expressions and Syntax

You can use Perl style regular expressions wherever patterns are allowed throughout the Admin Console. Principal (user/group) or IP patterns, for instance - `engineering.*` (note the dot) or `217.[0-9]+` are all valid patterns.

With the Perl regular expression syntax, if you need to use the '.' (dot) character literally, you need to escape it with a backslash, otherwise '.' (dot) matches any character. To learn more about regular expressions, read this tutorial.

For example, to allow any user with an IP address that begins with 192.168, enter the value: `192\.168.*`. Note that the period character must be prefixed by a backslash. With regular expressions, the period character is used to represent a character that can have any value.

4 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 Failover Sequence and Stand-Alone Failover Group of Two Replicators.

4.1 Node Failures

For Singleton Response quorum: say you have a five node replication group, spread across three continents. One of the nodes goes down. Replication continues at the remaining nodes, as long as the quorum is reached, although users connecting to the downed node are read-only until that node 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 nodes could go down, and replication would continue at the other nodes, as long as a majority of nodes remain up. The one or two downed nodes 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.

4.2 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 nodes 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 node can execute only read operations (like up, co, log, etc.) working with stale data. Providing that the remaining nodes can still meet quorum (having a majority of nodes responding), the other nodes 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 nodes.

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

4.3 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/.

5 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. See Strategies for Achieving Identical Repositories at All Nodes in Recommended Deployment Practices.

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

6 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.
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 nodes 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 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 node 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 providing 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 node in your replication group.

TERM	DEFINITION
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
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.