

Title: Monitor Pro ++ : Active Alarm Arbitration When Communication Is Lost Between Master and Slave

Resolution Number: 1331

What is the Purpose of the Resolution

When communication is lost between the Master and Slave redundant servers in MONITOR PRO ++, a user may have to manually intervene to ensure the correct state is established when communication is restored between the servers. There are four cases where communications can be lost between the servers.

1. The Master server loses its' network connection (card or cable failure)
2. The Slave server loses its' network connection (card or cable failure)
3. The Master server goes down (hard drive, power failure, etc.)
4. The Slave server goes down (hard drive, power failure, etc.).

The first case is the one that requires user intervention as explained in the following scenario.

What Customer Activities does this Involve

SET UP
OPERATE

What Units are Affected

Monitor Pro ++

Case 1 - The Master server loses its network connection (card or cable failure)

Node A is Master: Node B is Slave: the clients are connected to Node A.

- The two servers are running correctly as Master and Slave. An alarm is generated and logged on the Master server. The alarm is copied to the Slave to synchronize the alarm list.
- Communication between the two servers is lost (this could be simulated by disconnecting the network cable from the Master server).
- Node B, the old Slave, now assumes the role of Master. Node A remains a Master since it thinks Node B is not working. The clients failed over to Node B.
- The user now acknowledges the alarm from one of the clients and the acknowledgment is recorded on Node B. Since Node B is not connected to Node A, it cannot copy the acknowledgment to Node A to synchronize the data.
- At this point, if connection is reestablished, Node A sends its' alarm list to Node B to synchronize the data. This is the correct action for Node A to take since it doesn't know why it lost communications with Node B. If Node B had failed (case 2 or 4 above), the alarm lists would be synchronized and all is well. However, in this case, it overwrites the user acknowledgement. It would also overwrite any new alarms detected and logged on Node B.

To prevent the above condition, the user must shutdown the Monitor Pro Server on Node A before communications is reestablished. When the Monitor Pro Server is restarted (after communications is reestablished), it will correctly establish itself as a slave and Node B will copy its' alarm list to Node A to synchronize the data.

Case 2 : Slave server loses its' network connection (card or cable failure)

No user intervention is required. The clients do not failover to Node B since they still have communication to Node A (the Master). Any alarms logged or acknowledged are recorded on Node A. When communication is reestablished, Node A copies its' alarm list to Node B to synchronize the data.

Case 3 : Master server goes down (hard drive, power failure, etc.)

No user intervention is required. Node B takes over as Master and the clients failover to it. When Node A is returned to service, it will correctly establish itself as the Slave and Node B will copy its' alarm list to Node A.

Case 4 : Slave server goes down (hard drive, power failure, etc.)

No user intervention is required in this case since the system will behave the same as in Case 2.
