

FieldServer Hot Standby Operation

1 Terminology

1.1 Active FieldServer

The FieldServer which is actively polling the field nodes

1.2 Standby FieldServer

A FieldServer which is running, but is not polling field nodes, nor responding to client polls. It will assume active status if the Active FieldServer fails to issue a heartbeat in the designated time frame.

1.3 Failover Timeout

The time interval between the Active FieldServer failing and the Standby FieldServer preparing to become the Active FieldServer.

1.4 Transfer Interval

The total time interval between the Active FieldServer failing and the Standby FieldServer actually resuming communications as the Active FieldServer

1.5 Primary FieldServer

The FieldServer designated to be the Active FieldServer on system startup

1.6 Secondary FieldServer

The FieldServer designated to be the Standby FieldServer on system startup

1.7 Commbit Data Array

Bit data array that shows all the online nodes, one bit per node address. Practical limit is 255 nodes, the offset corresponds to the node ID.

1.8 NodeStat Data Array (new)

Int data array that shows all the status of all nodes, one integer per node address. Practical limit is 255 nodes, the offset corresponds to the node ID. The value of the integer corresponds to the current node status.

1.9 Hot Standby Status Data Array (new)

A data array showing the status of all Hot Standby FieldServers in a pair. E.g. Which FieldServer is active, is it the primary or secondary, is the standby FieldServer active, why did the switchover occur, if it did, etc.

2 Possible FieldServer Configurations for Redundant, Hot Standby operation

2.1 Configuration 1: Single FIELDSEVER, single connection to field node.

This is not a hot standby or redundant configuration. It is included here for comprehensiveness.

2.2 Configuration 2: Single FIELDSEVER, dual connection to field node.

This mode of operation requires the field node to have two serial ports.

For any given node configured in the FIELDSEVER, there is a “Backup Connection”. If there is a single communication failure, the kernel will attempt the next poll out the backup connection. If this poll is successful, then the backup connection is used from this point onwards. If the poll fails, then the kernel will alternate between the primary and the backup connection until the number of allowed retries is used up. Once this happens, the node is marked offline.

2.3 Configuration 3: Hot Standby FIELDSEVER Pair, Ribbon Cable, single connection to field node.

This is the classic hot standby mode, where the ribbon cable shares the FieldServer serial ports with a single field node serial port. If the primary FIELDSEVER fails, the secondary FIELDSEVER takes over. If a single port on the primary fails, at present, the primary FIELDSEVER will continue to be the active FieldServer.

2.4 Configuration 4: Hot Standby FIELDSEVER Pair, Ribbon Cable, backup connection to field node.

Combination of Configuration 2.3 and Configuration 2.2

2.5 Configuration 5: Hot Standby FIELDSEVER Pair, No Ribbon Cable, single connection to field node from each FIELDSEVER.

Although this is labeled as single connection to field nodes, each FieldServer in the pair is connected to one of two ports on field node. When the Active FieldServer fails, the standby FieldServer will automatically take over. The question is, what happens if a single cable is cut? We want to switch to the standby FieldServer and attempt to recover communication. If however a node fails, communication recovery is not possible, and we want to prevent flip-flopping between the FieldServers. This can be prevented by looking at the “Health Data” of the Standby FieldServer. (See section 5.1.4)

2.6 Configuration 6: Hot Standby, No Ribbon Cable, backup connection to field node.

This is not a legal configuration. Included for comprehensiveness. (It would require the PLC to have four serial ports)

3 Heartbeat Connections

The Active FieldServer keeps the Standby FieldServer passive by sending a regular heartbeat. The heartbeat connection can be made via various methods.

3.1 Ribbon Cables

This is included for comprehensiveness, but is no longer used

3.2 Serial Connections

The heartbeat message is sent over dual serial ports.

3.3 Ethernet

The heartbeat message is sent over Ethernet links.

4 Theory of Operation

4.1 Operation of Active FieldServer

The active FieldServer sends heartbeat messages at very regular and short intervals. The messages may contain data array data which is used to update the data arrays of the standby FieldServer, so that if and when the standby becomes active, any clients polling for data will be sent current data values.

4.2 Operation of Standby FieldServer.

The Standby FieldServer listens to the heartbeat messages. If the heartbeat messages fail to arrive within the Failover Timeout, it will switch to Active mode and start to send heartbeat messages to keep the other FieldServer firmly in standby mode (assuming it is still operational, or has been replaced)

Consider configuration 5.

Mode 1: FieldServer fails.

This is relatively straightforward. The Standby FieldServer fails to receive the heartbeat messages, and after the configured Failover Timeout, takes over as the Active FieldServer. If the previously active FieldServer restarts, it will remain as the new Standby FieldServer until it no longer gets a heartbeat message.

Mode 2: Cable is cut, or serial port on FieldServer or field node fails.

This is more complex. It is quite possible that a cable is cut or disconnected, and we want to try and anticipate this condition.

5 Hot Standby FieldServer configuration issues

5.1 Configuring a FieldServer for Hot Standby

5.1.1 *Establishing Hot Standby Operation*

5.1.1.1 *Defining a “Hot_Standby” connection in configuration file.*

The FieldServer can determine that it is configured to act as part of a Hot Standby Pair when a “hot_standby” connection is defined in the configuration file. At startup a particular FieldServer would now need to determine if it should become the Active or the Standby FieldServer

At startup the FieldServer will wait for a heartbeat message from an Active FieldServer for a random time of between 2.0 and 2.8 seconds. If the Failover Timeout has expired, the FieldServer will become the active FieldServer and start sending heartbeat signals on the Hot Standby connection. Typically this will be used in Configuration example 3 above (Section 2.3).

5.1.1.2 *Explicit configuration.*

The FieldServer can be explicitly configured as either a Primary or Secondary FieldServer by way of a separate hot standby configuration file. This file will be stored on the FieldServer flash disk. If the FieldServer is configured as a Secondary FieldServer, then it will wait an additional one-second before starts up. This will give the Primary FieldServer (if one exists) time to become the Active FieldServer when the hot standby pair is switched on at the same time. If both FieldServers in the hot standby pair are configured the same way, then essentially the mode of operation reverts back to the automatic configuration explained above. Typically this will be used in the Configuration Example 5 above (Section 2.5).

5.1.2 *Heartbeat Message Operation*

Heartbeat operation can be configured to work over a Serial Connection or over an Ethernet connection (new) or a combination thereof (new). This is achieved by setting the Connection_Mode of the relevant Connection to Hot_Standby. Two connections must be designated for Hot Standby operation.

5.1.2.1 *Heartbeat Messages over a Serial Connection.*

Configuration of this is straightforward and requires two serial connection cables to run between the Hot Standby FieldServer pair.

5.1.2.2 *Heartbeat messages over Ethernet.*

Sending heartbeat messages over Ethernet require that FieldServers of a particular Hot Standby pair be explicitly configured with a “Pair Name”. This will be done in the hot_stby.ini file. When a FieldServer wants to send a heartbeat message over Ethernet, it would first send an Ethernet “Who-Is Standby” broadcast message containing the FieldServer Hot Standby “Pair Name”. Any FieldServer that receive this message and has a matching Hot Standby Pair Name will respond with an “I-Am Standby” message. Subsequent heartbeat messages will be sent directly to the MAC address of the response message.

Note that this mechanism does NOT USE IP ADDRESSING. This eliminates the need for annoying IP address configuration. It also means that the messages cannot be routed however, but this is OK since anybody who wants to put a router between the two FieldServers in a Hot Standby pair need to re-evaluate their career choices anyway.

5.1.3 *Transferring Data Arrays to the Standby FieldServer.*

An integral part of Hot Standby operation is to transfer “live” FieldServer data from the Active FieldServer to the Standby FieldServer. For this to work the Active and Standby FieldServers

must have identical data arrays configured. This transfer of data is achieved by using the heartbeat signal.

If the Connection_Mode is set to Hot_Standby_Data, then the heartbeat message will continuously transfer all the Data Array data to the standby FieldServer, one section of data at a time.

If the Connection_Mode is set to Hot_Standby_Data_On_Change then only the part of the Data Array that changes will be sent to the Standby FieldServer. Typical applications for this are when the Data Arrays are very large and the data changes very seldom, such as for a fire panel system. In particular this would make sense if the heartbeat message is send over Serial connections.

5.1.4 Monitoring Standby FieldServer Health from the Active FieldServer

Both FieldServers in a hot standby configuration pair can be configured with a Data Array containing the FieldServer status. This data will be send back to the Active FieldServer by including it into the heartbeat message acknowledge packet. Typically this would contain the last known status of all the nodes on the Standby FieldServer. This information could then be used by the Active FieldServer to decide if it should cause a FieldServer swap-around in case of a port/node failure.

5.1.5 Connecting and downloading to the Standby FieldServer.

Ruinet can be used to connect to the Active as well as the Standby FieldServer. For this to happen it will be required to know the IP_Address of the relevant FieldServer.

Ruiping (using the 'h') switch can be used to determine the FieldServer IP Address, Hot Standby Pair Name and Hot Standby status (i.e. active or standby)

5.1.6 Naming convention and operation of the hot_standby ini files

- If the FieldServer is configured as a Primary FieldServer, the file *hsb_p.ini* must be created and downloaded to the bridge.
- If the FieldServer is configured as the Secondary FieldServer, the file *hsb_s.ini* must be created and downloaded to the FieldServer.
- The syntax of the *hsb_x.ini* files is exactly the same as that of the standard FieldServer configuration file.
- The Hot Standby Pair Name must be set as the "Bridge" Parameter "HS_Pair_Name".
- Normally, the Hot Standby connections will also be configured in the ini files.

6 Legal Node States

This section describes all the existing and new legal states a node can be in.

6.1 Startup (new)

When the FieldServer starts up the initial state of a node is 'startup'. This remains until there has been one successful poll, at which time the node will go into normal state. If there have been 3 retries and the node does not respond, then the state will become "Offline".

6.2 Normal (existing)

A node that is being polled, and is responding is considered normal. Note that if the response is something like illegal memory address, this is NOT considered a reason to take a node offline. Some PLCs will respond to some addresses but not others, eg Modbus 81 02 "illegal memory address".

Checksums, timeouts, etc. are still valid reasons to take a node offline.

6.3 Failing (existing)

If there is no response to a poll, the state of the node will become Failing. In this mode, there will be a number of retries (as per configuration). If there is a backup connection defined, the kernel will attempt to connect on the backup connection. Each retry is attempted at the retry interval time. If the number of retries are used up, then the kernel will either attempt to switch over to the standby FieldServer (new) if there is a standby FieldServer to switch to, or will mark the node offline.

6.4 Offline (existing)

The kernel will attempt to recover the offline node at the recovery interval time. If there is a backup connection, then the kernel will alternate the connection for each attempt. If a node responds, then the state will go to 'probation'

6.5 Probation (existing)

Probation is a state after a node has reached offline, and is starting to recover. For all practical purposes the node is still considered offline. This state is to prevent a flaky connection from coming online, then offline, then online etc. This probation state will currently last 1 minute, but is planned to default to 1 second, so a user has to explicitly set this delay.

6.6 Trying Alternative Connection

If an alternative connection is available, then this is a possible state. However there is a possibility that this is a duplicate of the "Failing" state.

7 IP Addressing

7.1 Mode 1: Static Dual IP Addresses

In this mode, each FieldServer is provided with an IP Address via the normal configuration process. The IP Addresses remain static and the client software package has to detect that the Primary FieldServer has gone offline and that the Secondary FieldServer with another IP Address has taken over.

Client software such as FieldServer Technologies OPC driver will handle this occurrence seamlessly.

Packages such as Intellution's Fix, without FieldServer Technologies OPC driver, will handle the switchover automatically, but report a communications failure during this period.

Packages such as GE Cimplicity do not handle this process at this time.

7.2 Mode 2: Dynamic Single IP Address

This mode is for the client packages that do not handle Mode 1. The Secondary FieldServer will assume the IP Address and MAC address of the Primary on switchover.

Most client software will still report a communications failure during this interval..

7.3 Mode 3: Bumpless Transfer IP Address

This mode is the same as Mode 2, but the Standby FieldServer closely shadows the state of the Active FieldServer's socket by watching traffic on the network. If a switchover is required, the client software will not be able to detect that another FieldServer has taken over the communications.

8 Hot Standby Status Array

This table is obtained from ONLY the Primary FieldServer.

HSBY data array offset	Value	Description
0	1	Primary OK *
1	1	Secondary OK *
2	1	Primary is active *
3	1	Secondary is active *
4	1	Hot standby system OK
5	x	Number of times primary has become active *
6	x	Number of times secondary has become active *
7	1	Cable N1-N1 is OK *
8	1	Cable N2-N2 is OK *
9	1	Hub N1 is OK, 0 Hub is NOK, 2 indeterminate.
10	1	Hub N2 is OK, 0 Hub is NOK, 2 indeterminate.
11	1	Cable PFS:N1 is missing
12	1	Cable PFS:N2 is missing
13	1	Cable SFS:N1 is missing
14	1	Cable SFS:N2 is missing
15	1	Cable PFS:N1 is plugged into incorrect hub
16	1	Cable PFS:N2 is plugged into incorrect hub
17	1	Cable SFS:N1 is plugged into incorrect hub
18	1	Cable SFS:N2 is plugged into incorrect hub
19	1	Cable N1 is plugged into N2 of other bridge
20	1	Cable N2 is plugged into N1 of other bridge
21	1	Cable N1 is plugged into incorrect bridge partner
22	1	Cable N2 is plugged into incorrect bridge partner
23	1	Hot standby hubs not on independent networks (e.g. joined by some link – not allowed for hot standby operation.) ³
24	1	Somewhere a backup connection used

9 Node Status Array *

This is the data array which reflects the node state according to the Configuration Manual describing the possible states of each node in the FieldServer. Offset 1 represents the node state of Node ID 1.

Revision History

Date	Driver Version	Document Revision	Resp	Comment
2/22/01	1.0aA		EKH	Releasing
4/3/01	1.0aB		BDW	
4/4/03	1.0aC		EKH	
4/6/01	1.0aD		BDW	
4/21/01	1.0aE		EKH	
3/20/03	1.0aF		JD	