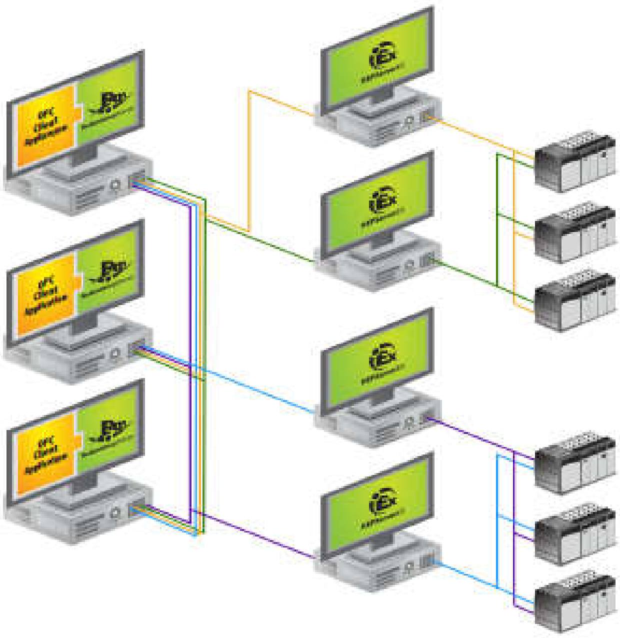
RedundancyMaster
SKU: KWP-RDNMS0-PRD

Product Overview

RedundancyMaster increases the reliability and availability of your OPC data by allowing multiple OPC servers to be configured into redundant pairs. Each redundant pair seamlessly appears as a single OPC server to any OPC client application. RedundancyMaster can be added to an existing client/server application without any reconfiguration of the application, keeping your processes going without any downtime.



Industrial Strength Reliability

OPC Data Access (OPC DA) technology has been proven to be reliable in virtually every industrial environment requiring consistent data access to devices and systems. However, there are other factors that can jeopardize the integrity of a system, including software, hardware, and even human error. By using OPC redundancy technology, you can help make these systems more reliable and efficient.

Increased ROI & Reduced Downtime

RedundancyMaster resides on your OPC client machine and facilitates connections to a primary and secondary OPC server on the system's networks by "hooking" into the OPC calls made between the client and the server. If for any reason the OPC client loses its communications link with the primary OPC server or a user-specified condition is met (such as an item is not receiving updates, a specific item value is met, or the quality of an item is set to bad), RedundancyMaster will drop the primary OPC server and promote the secondary OPC server on your network—reducing system downtime and saving you money.

Ease of Use

RedundancyMaster is a drop-in application that does not require you to make any changes to your OPC client or server applications. Its intuitive configuration takes only minutes and will allow you to easily establish a redundant OPC system. Simply browse and select your primary and secondary OPC servers, and then your system is up and running. RedundancyMaster includes features such as email notification, object and link monitoring, and diagnostics logging. In the situation where you need multiple redundant OPC server pairs that utilize the same OPC server vendor, we have added the capability to alias the ProgID (Program ID) of the OPC server. (Aliasing may require minor OPC client modifications.)

Product Features

Explore the features that will change how you think about OPC redundancy. The innovations in RedundancyMaster can work together seamlessly with your current OPC application to give you a more reliable, cost-effective solution.

Primary/Secondary Machine Names

Browse for the primary machine that specifies the preferred connection that should be made to an OPC server and the secondary machine which specifies the fallback connection that should be made to an OPC server when communications to the primary machine are unavailable. Every time a new client connection is made to the underlying server, the application will first attempt to make a connection to the server running on the primary machine. In the event that the connection to the primary fails or communications to the primary is lost, a connection to the secondary server will be attempted and, if available, established. Depending on the connection mode, you can configure the application to automatically establish communications with the primary machine when it becomes available.

Connection Mode

The connection mode defines how and when the redundancy application should connect to the underlying primary and secondary servers. The mode in which you operate affects the amount of time it takes to fail over from one OPC server to the other. Some modes allow you to automatically promote communications to the primary when it is available. The following summarizes connection modes:

Cold (active machine only): In this mode, the application will only connect to one underlying server at a time. On startup, a connection to the primary server will be made and all client related requests will be forwarded onto the primary. In the event that the connection to the primary fails, or communications to the primary is lost, a connection to the secondary will be made. If the redundancy application is unable to obtain a connection to the secondary, it will continue to ping-pong between the two servers until it makes a successful connection.

The "cold" connection mode minimizes the amount of system resources that are allocated since there will only be one connection to one server at any given time. It also reduces network traffic since there is no need to poll the inactive machine in addition to the active machine, as in other modes. The drawback to this setting is the amount of time it takes to fail-over to the inactive server. When communication loss is detected with the active server, the application needs to establish the connection to the inactive server, subscribe to all items on behalf of the client, and initiate the appropriate callback mechanisms.

Warm (both machines, subscribe to items on active machine): In this mode, the application will attempt to maintain a connection to both the primary and secondary servers at all times. Only items in the primary server will be active and polled. In the event that the connection to the primary fails, or communications to the primary is lost, the identical items in the primary server will be set to active in secondary server. Periodically, both servers will be pinged to determine if the connection is still valid.

The "warm" connection increases the amount of system resources that are allocated, since there will be two server connections made on behalf of the client. There is also a minimal increase in network traffic due to periodically pinging two servers instead of one, as in "cold" mode operation. The benefits are that fail-over time is minimized over "cold" mode operation, since the redundancy application will only have to initialize data callbacks to the inactive server to begin receiving data. If you need to minimize the loss of data in your application, and at the same time want to minimize network traffic, you should use this connection mode.

Hot (both machines, subscribe to items on both machines): In this mode, the application will attempt to maintain a connection to both the primary and secondary servers at all times. On startup, the application will initialize data callbacks for both primary and secondary servers so that both servers will send data change notifications. The data received from the primary server will be forwarded on to the client. In the event that the connection to the primary fails, or communication to the primary is lost, data received for the secondary will immediately be forwarded onto the client. In either case, writes will only be forwarded to the active server. Periodically, both servers will be pinged to determine if the connections are still valid. If at anytime the redundancy application loses communications to either server, it will periodically attempt to reconnect to the failed server. This setting increases the amount of system resources that are allocated, since there will be two server connections made on behalf of the client. There is also an increase in network traffic due to receiving data change notifications from both underlying servers, as well as periodically pinging both servers to determine if they are still available. The benefit of this setting is that fail-over time occurs immediately after detecting the loss of the active server. If loss of data is very crucial to your application, you should use this connection mode.

OPC Server Aliasing

This feature allows you to configure multiple pairs of OPC servers with the same ProgID. This feature permits you to use one OPC server vendor if you have multiple OPC server nodes on your network. This will allow OPC clients to connect to a specific redundant pair by referring to the aliased ProgID of that redundant pair.

Always Connect to Primary Machine Upon Availability

This setting enables RedundancyMaster to automatically promote communications back to the primary machine when the OPC server becomes available.

Query Server Status Interval

This interval (specified in milliseconds) determines how often RedundancyMaster will ping the underlying servers to determine if there has been a loss of communications. By querying at a faster rate, you can minimize fail-over time since failure detection occurs more frequently.

Query Server Status Timeout

This interval (specified in milliseconds) determines how long the redundancy application will wait for a ping response from the underlying servers before considering there to be a loss of communications.

Monitoring Settings

This feature allows you to configure certain conditions which will initiate a fail-over to the inactive server. These conditions allow you to monitor server items for specific states to determine the health of the underlying servers/devices, above and beyond the automatic fail-over that will occur due to the loss of communications.

Diagnostics Settings

Events can be preserved to disk when the application is shutdown. The next time the application is started, the events will be displayed and any new events will be concatenated to the end of the view.

Since diagnostics utilizes memory and storage resources, you may want to limit the number of diagnostics that are preserved at any given time. RedundancyMaster allows you to set the maximum number of events to capture. Once the maximum number of events has been reached, the oldest events will be discarded as necessary.

Notifications Settings

This feature allows you to configure one or more recipients to receive email notifications for one or more diagnostics events. The events that are available to send as email notifications are the same events visible to the local Diagnostics Settings event view.

Use Case Scenarios

Mitigating Object-Based and Link-Based Failures

There are a many variables that could impact the quality and reliability of your data or cause an OPC system to lose connection to an OPC server. The most common include:

- The PC running the OPC server is shut down
- User errors cause the OPC server to exit
- The network connection to the OPC server is lost or unreliable
- The network setting is changed causing link failure
- The OPC server itself fails for any reason, known or otherwise
- The login account is changed on the OPC server's PC

In most of the cases above, the OPC DA server fails to provide data due to an actual failure underlying the OPC server or the connection to that server. These types of failures are known as "object-based" failures. Object-based failures occur when the actual link between your OPC client application and the target OPC server breaks down. In these examples, software is at fault. However, physical hardware breakdowns within an application can dramatically affect reliability as well. Some of these physical factors include:

- Physical connection failure (the cable is pulled)
- Hardware failure (router failure)
- Electrical Interference (high current discharge)
- Delays due to signal propagation (radio links)
- Environmental factors (lightning)
- Random accidents

In these situations, the virtual connection between the OPC server and the client may be perfectly intact but the physical link to the underlying device or system may be broken. These types of failures are known as "link-based" failures. Link-based failures occur when the connection to the target device or system has been lost. In most cases, the OPC server is still completely operational, but simply cannot supply the data to the rest of the system.

RedundancyMaster can be configured to monitor these conditions and prevent unnecessary downtime in your system, saving you time and money.

Two OPC Servers Paired with RedundancyMaster

If multiple OPC DA client applications are accessing a single OPC server, the potential exists for both an object-based failure or a link-based failure. If for any reason the single OPC server fails to operate, an object-based failure can result. Furthermore, since this single PC is responsible for data collection from the underlying devices, a single point of failure exists for the device connections as well.

To increase the reliability of your OPC system, you need to remove these single points of failure by redesigning your OPC system to use more than one OPC server. To facilitate the redundant operation of the OPC servers, each OPC client is paired with RedundancyMaster.

Using the configurable options within RedundancyMaster, the use of either the Primary or Secondary OPC server can be controlled directly. Based on the modes selected, RedundancyMaster will keep both servers active or (if configured to do so) start the secondary server only when the primary server fails.

Local Machine Redundancy

In this scenario, the OPC client, RedundancyMaster, and the secondary OPC server all reside on the local machine, and the primary OPC server resides on a remote machine. For this system, be sure to make the most reliable server your secondary OPC server. This scenario reduces the need for another machine to run the secondary OPC server.

Multiple OPC Server Pair Redundancy

RedundancyMaster can be configured to have multiple OPC server pairs. In this scenario, there are two pairs of OPC servers that are gathering data from two separate device networks. If the multiple OPC server pairs are all of the same ProgID, then you will need to use the Aliasing feature. If the two pairs have different OPC servers with different ProgIDs, then you will not need to use the Aliasing feature.

Application Support

- OPC Data Access (OPC DA) Versions 1.0 and 2.05a

Operating Systems

- Windows 8
- Windows 7 Professional/Enterprise/Ultimate
- Windows Server 2012
- Windows Server 2008 and 2008 R2
- Windows Vista Business/Enterprise/Ultimate
- Windows Server 2003 SP2
- Windows XP Professional SP3 or higher

System Requirements

- 2.0 GHz Processor
- 1 GB installed RAM
- 180 MB available disk space
- Ethernet Card
- Super VGA (800x600) or Higher Resolution Video