

Redundancy in Adroit

Very few SCADA HMI products offer genuine, true redundancy whereby operational integrity of the engineered solution can survive hardware failure in one of the system components. And those that do, require complex script programming and system engineering to achieve the requirement of a highly-available, resilient solution.

One of Adroit's major features that translate directly into a key benefit for end-users and system integrators alike, is the ability to implement redundancy by simple point-and-click configuration at the time a solution is being engineered.

Redundancy in Adroit is available at three distinct, and yet independent levels: PLC, I/O Server, and Graphics Server. By *independent*, we mean redundancy can be implemented at all, or any sub-set of the three levels, as shown in Figure 1 below

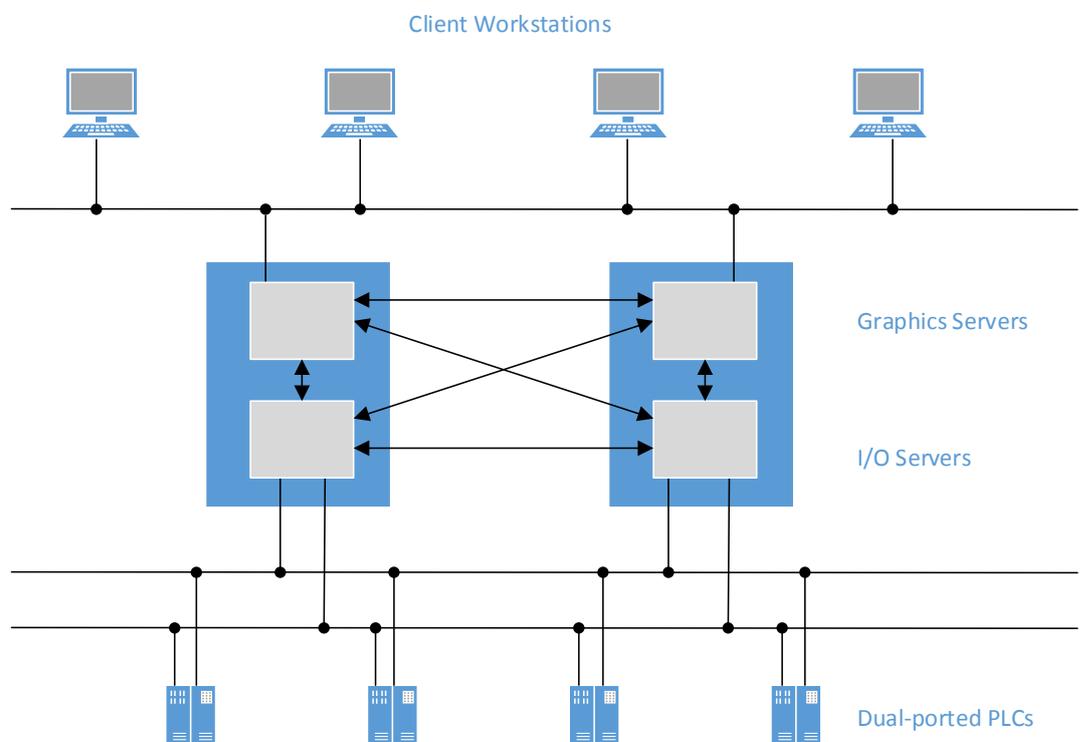


Figure 1: Three level redundancy in Adroit

PLC Redundancy

To implement PLC-level redundancy, either dual-ported PLCs are required, or alternatively completely redundant PLCs can be used. Figure 1 shows single, non-redundant PLCs, each with a primary and secondary communication channel.



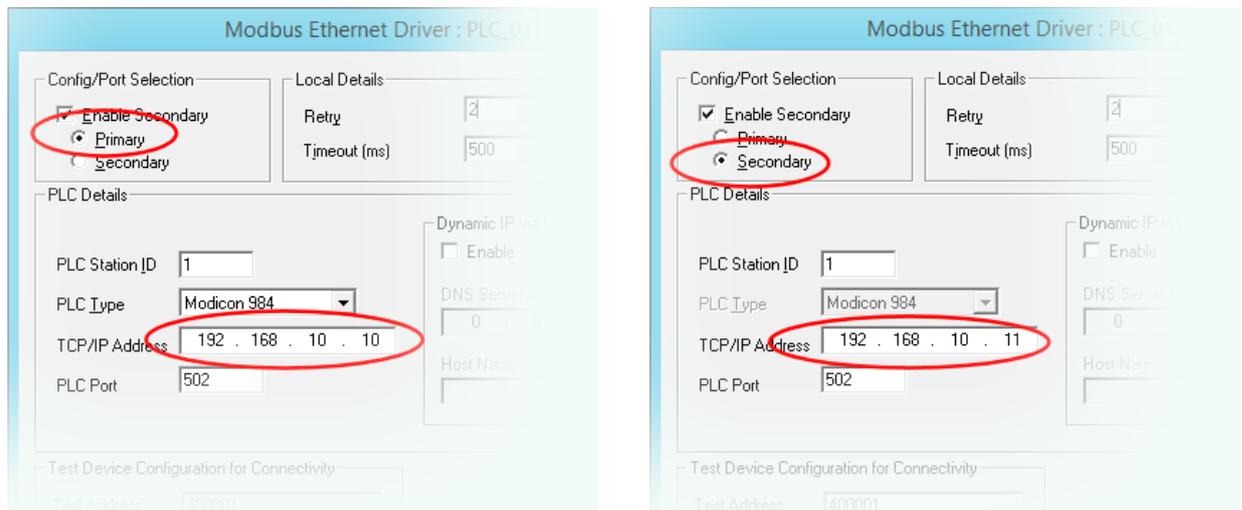
ADROIT

TECHNOLOGIES

Tel: +44 1270 627 072

Adroit Technologies Ltd
PO Box 19 Nantwich
Cheshire England CW5 6FF
www.adroit-europe.com

To configure PLC level redundancy, all you need do is check the *Enable Secondary* option in the PLC driver configuration as shown below. The screenshots show a PLC(s) having two different IP addresses. It is also possible with some PLC models to use the same IP address but connect to different PLC ports.



I/O Server clustering

With reference to Figure 1, the central blue rectangles represent different physical (or virtual) machines forming a dual-redundant pair. Inside each machine are two server processes – an I/O server (Agent Server) and a Graphics server (VIP Server), represented by the light grey rectangles. Although not represented in Figure 1, it is possible to run the I/O server and Graphics server processes in different machines.

The I/O servers are responsible for performing the usual range of SCADA functions: scanning the PLCs, maintaining a real-time and historical database of process values, checking for and managing alarm conditions in process values, etc. An I/O Server is also responsible for making real-time and historical values available to the Graphics servers for onward transmission to Client Workstations or HMIs, as well as receiving process value updates originating in the HMIs, e.g. controls and set-point changes, etc.

In a dual-redundant configuration, an I/O server needs to communicate in peer-to-peer fashion with its partner I/O server on the other machine. This is represented by the horizontal arrow in Figure 1, linking the I/O servers.

A number of options are possible when configuring this peer-to-peer I/O Server communication. These are shown on the configuration form below:

Server Type

Select *Cluster Server* to enable the dual-redundant I/O server functionality

Cluster Name

Make sure the cluster name for the two I/O servers is the same

Enable client runtime transaction shadowing

Check this option if you want control and set-point value changes shadowed across to the partner server

Shadowed scanning

This means PLC values acquired by the duty master server are shadowed across to the partner standby server

Configuration: Agent_Server_Configuration

Agent Server Type: Cluster Server [Options]

Cluster Name: Cluster1

Enable client runtime transaction shadowing

Shadow scanned data from master to standby nodes

Parallel scanning

Enable writes when in standby mode

Enable reads when in standby mode

Manual master mode assignment

Automatic master mode assignment

Make this server the master if no other cluster server is detected after 15 seconds from startup

Server Project Name: VM [Options]

Agent Server Name: VS2

WGP File: C:\ProgramData\Adroit Technologies\Adroit\Configuratic...

Parallel scanning

This means PLCs are scanned in parallel by both master and standby server. You also have the option of deciding whether to write out control and set-point values when in standby mode

Manual master mode assignment

At start up, I/O Servers initially take on the standby role, and if the *Manual master mode assignment* option is selected, then the server will stay in standby until something explicitly changes it to *master*

Automatic master mode assignment

This is the usual option, and means that after a configurable post-startup delay, a standby server will automatically promote itself to master if a partner server cannot be found that is already in master mode

Project Name

This is a way of allowing I/O Servers running on the same LAN to be independent, i.e. by having different project names. In other words to ensure that a pair of servers participate in a dual-redundant cluster, you need to make sure that their project name settings are the same

Server Name

Every I/O server running on a given LAN must have a unique agent server name. If not you will encounter an error starting up your server if it finds another server with the same name

The implication so far, in all the discussions of I/O server redundancy, is that servers exist on the same LAN. But it is also possible for I/O Servers to run on different geographically separate network domains. In this case a separate set of configuration options exist.

These are shown below, and are accessed by clicking the *Options* button next to the *Server Project Name* setting

The screenshot displays a configuration window with the following elements:

- Agent Server Type:** A dropdown menu set to "Cluster Server" with an "Options" button to its right.
- Server Project Name:** A text input field containing "VM" with an "Options" button to its right.
- Additional connections (using visible broadcasts):** A checkbox that is currently unchecked. Below it is a table titled "Projects / Servers" with a "+" button to add entries and an "X" button to remove them. The table is currently empty.
- Explicit connections (broadcasts not visible):** A checkbox that is currently unchecked. Below it is a table titled "Agent Servers" with a "+" button to add entries and an "X" button to remove them. The table has two columns: "Machinename" and "Agent Server Name:". The table is currently empty.
- Server timeout (seconds):** A numeric input field set to "15" with up and down arrow buttons.
- Enable remote servers to automatically detect this server:** A checked checkbox.

Working up from the bottom of the screenshot above...

[Enable remote servers to automatically detect this server](#)

This is the default setting (checked) and assumes servers exist on the same LAN and can see each other by exchanging UDP broadcast messages

[Explicit connections \(broadcasts not visible\)](#)

Essentially, this option allows you to specify your cluster partner by explicitly defining its physical details like machine name and server name. This is because UDP broadcast messages cannot travel between different network domains

[Additional Connections \(using visible broadcasts\)](#)

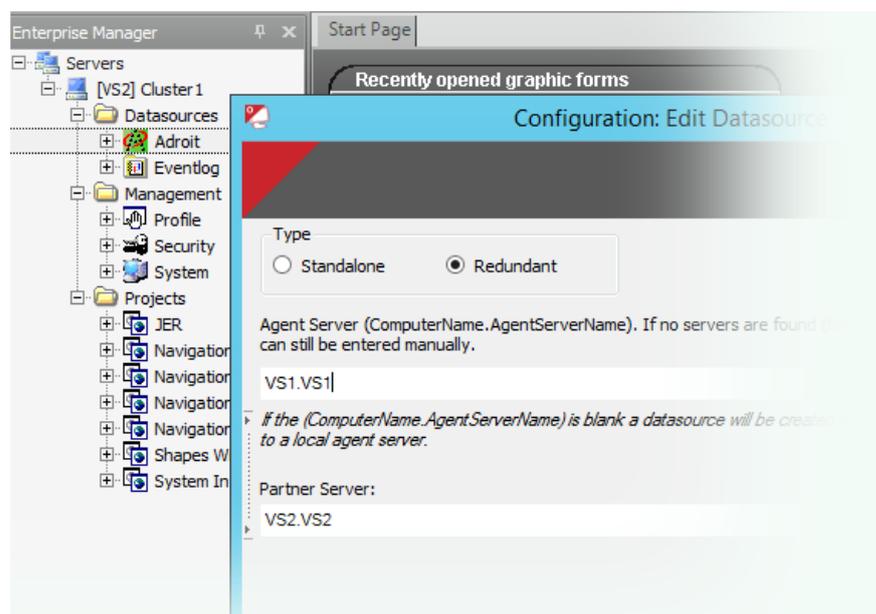
It is also possible to connect to servers on the same LAN, but running with different project names. The reason for this option has do with Adroit's distributed tag database capability, whereby all servers with the same project name interconnect and share their tag databases. If there are a large number of such servers, then the total number of network interconnections can become very large. To prevent this, we stop servers from automatically connecting, by having differing project names. By then explicitly listing the projects/servers to connect to, we can control exactly which servers connect to each other

Graphics Server clustering

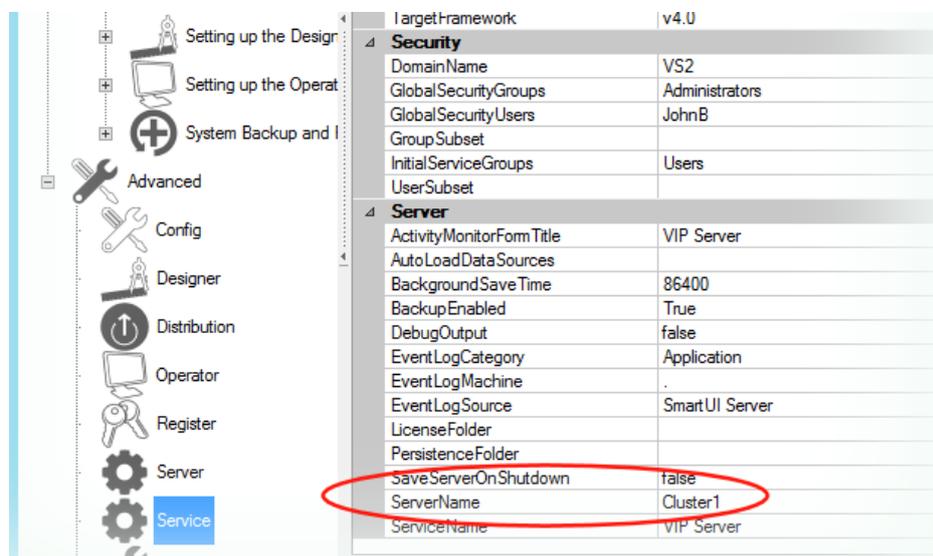
Clustering at the Graphics server level provides two benefits:

- Redundancy - which, like I/O Server clustering, provides for high-availability and resilience
- Load-balancing - which improves performance by allowing multiple Graphics servers to share their workloads

The first configuration aspect to be considered is the I/O server connections, assuming dual redundant I/O servers have been configured. As the screenshot below shows, this is done via the SmartUI Designer's Enterprise Manager, where the Adroit datasource is specified to be of type *Redundant*, and the machine and agent server names are explicitly configured. This results in the Adroit datasource icon showing up bright green, and is represented in Figure 1 by the diagonal arrows

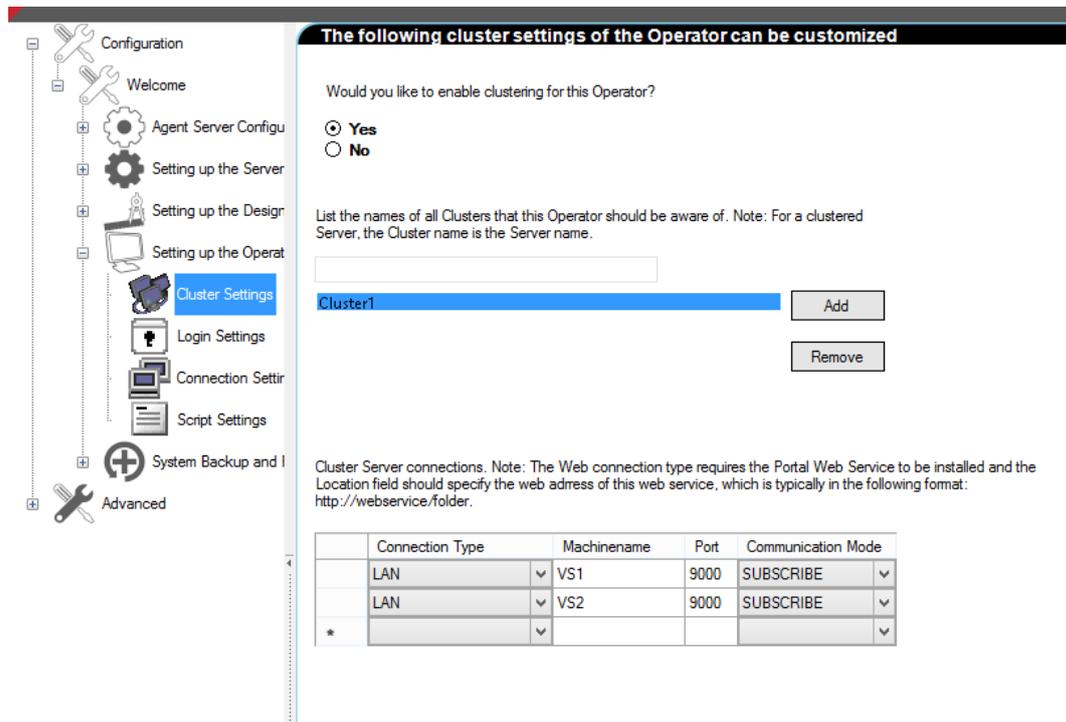


For the purpose of clustering the Graphics servers themselves to allow redundancy and load-balancing, the only thing you need do is change the *ServerName* setting in the Server/Service Advanced section in Config Editor to a common name, something like *Cluster1*, as shown below



The final consideration under Graphics server clustering has to do with letting the Client HMI *Operator* applications know that they are to connect in to a cluster of Graphics servers. This is done under Operator Cluster settings as shown below.

Firstly you need to check the option specifying clustering and then choose the cluster name that the Graphics servers are running under, e.g. Cluster1. Next you need to provide a list of machine names running the Graphic server processes, of which there will be two in a dual-redundant configuration



In Conclusion

Carrying out all of the aforementioned dual-redundant configuration options will provide resilience, high-availability, and optimum performance from a pair of physical or virtual server machines.

By virtue of load-balancing, assuming both servers are operational, the graphics processing load applied to the servers by the client HMI connections will be evenly distributed across the two machines.

If the duty master server should fail for whatever reason, the active and fully in-sync standby machine will detect this and promote itself to master mode. Client HMI workstations running the SmartUI Operator application will see the change in master machine and bumplessly reconnect to the newly promoted master.

Finally with PLC redundancy configured at the driver level, if any of the PLC connections should fail the driver will automatically switch to the secondary channel.

All changes like this in the twin system state are fully monitored, and can be alarmed and logged, etc., just as process values would be alarmed and logged. This results in a manageable, highly available, resilient solution, ideally suited to industries demanding this kind of integrity. Solutions of this type have been successfully deployed on several mission critical applications in the defence, marine, power and nuclear sectors.