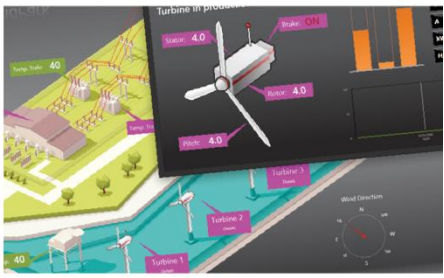
INDUSTRIAL AUTOMATION SOFTWARE
progea

Movicon.next™
Automation Platform.next generation



Movicon NExT

19.0 Redundancy

Ver.3.4.268

Table of Contents

1. Redundancy

1.1. Redundancy (Fault Tolerance)

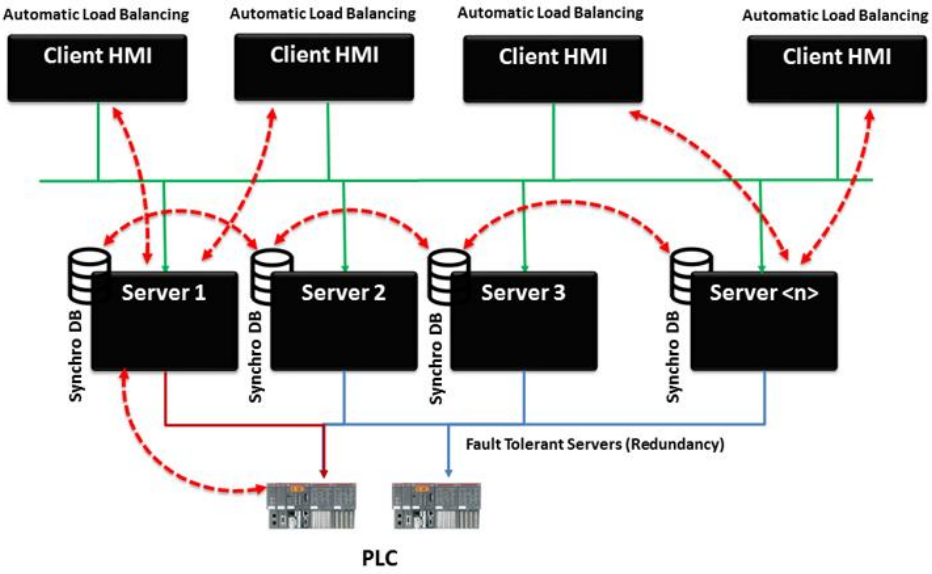
Those control systems which manage critical automation processes must be fault tolerant to ensure reliable and continuous performance and service in the event of system errors or downtimes. Continuous service means full operation of the supervision system and full integrity of recorded process data.

To ensure continuous service the supervision system must provide a 'redundancy' fault tolerant management composed of two or more server systems and two and more client systems that are capable of synchronization with each other automatically.

Platform.NExT integrates powerful and automatic functions for supporting redundancy management of mission critical server and client stations connected in net. This functionality synchronizes data transparently and automatically from one server to the next in a series of servers connected in net for both communications and historical data. This is done in order to have one Server active as the 'Primary' server with other servers active as 'Secondary' servers continuously at all times.

Analogously, the Client stations are permanently connected to one of the active Servers to ensure an automatic switch over to another server station to avoid interruptions and downtimes.

The redundancy functions are enabled in runtime only if the Redundancy and Networking options have been enabled on the license of each PC station participating in the system.



An example of a fault tolerant architecture for supervision stations. It can consist of several Servers, of which one will be active, and several Clients connected to servers.

Redundancy Level

Possessing a redundancy function in automation systems is an absolute must to ensure that in the event of malfunctioning of any kind in one component another identical one is ready on standby to take over automatically.
The "Hot Backup" functions automatically enters the secondary unit into operation without requiring any manual intervention by the operator.
The redundancy concept can be applied both to the hardware as well as the software to determine the minimum loss of data or functionality of the system when switching control over from the Primary unit to the backup Secondary unit.
The redundancy concept in automation systems can be applied to the following **components:**

Field	**Connection**	**Supervision**
PLC, Slave, I/O	Serial, Fieldbus, Ethernet network	Server Scada, HMI (Platform.NExT)

The Redundancy functions that have been integrated in Platform.NExT support the Fault Tolerance functionality in PC stations both on the server and client side so that the communication, visualization and control functions of the active stations can be transferred to the inactive stations completely in automatic. Historical data are also synchronized automatically as well.
The specific Movicon proprietary technology has been designed to synchronize data in an instant no matter even those of large amounts. This is due to synchronizing data acquired while operating in emergency mode by transmitting the data in binary format rather than data structures in database format.

1.2. Redundancy Functionality (Fault Tolerant)

The redundancy management with Server and client functionality has been completely integrated in Platform.NExT platform. It has been explicitly designed to guarantee the automatic intervention of one of the Secondary Servers (inactive) in the event of fault on the Primary Server (active) in such a way that it is completely transparent to the user and detectable after a timeout (modifiable).
In addition, the platform supports the option to define an array of servers to make absolutely sure that the plant system continues to operate if another fault should occur on the active server.

How the redundancy feature functions

The principle of the Platform.NExT Redundancy functionality is to have two or more Server modules connected to field devices and connected to each other in an ethernet network. Each one of these Servers must be defined with a project configuration consisting of a list of these Servers connected to each other along with an intervention timeout.
Each one of these servers must also be equipped with a runtime license enabled with the redundancy option.

Normal working conditions

During normal working conditions, the Primary Server is connected to the field devices and manages the data and data recording. The other Servers are operative, provide realtime data and record information in the exact same way in perfect synchronization with the Primary Server. They do not communicate with the field even though they are predisposed to do so.

Emergency working conditions

In the event of a fault on the Primary Server, the next Secondary Server in line (in a list of backup servers) steps in to communicate directly with the field devices and record data. It will also automatically notify the other Servers on the list that it has taken over as Primary Server. All the other Secondary Servers will adapt accordingly.
If the new Primary Server should go into fault, the next Server in line will assume the role of Primary Server as indicated above.

Restoring normal working conditions

When a Primary Server returns back into operation it automatically restores the local historical situation and acquires all active information. Once synchronization has taken place the restored primary server will start to communicate with the field devices resuming its primary function. As a consequence the Server, that was active when this happened, will return to its 'secondary' operative condition in accordance to the data provided by its Primary Server.

Client connections

The data display client stations, connect to the system servers, are permanently and automatically connected to one of the active servers according to the load balancing principle. Therefore, if a Client station is connected to a Server that goes out of service, the Client will automatically search for the next Server station to ensure continuous service in automatic mode.

Thanks to the automatic redundancy functionality based on the working principles described above, the Platform.NExT supervision system is capable of ensuring excellent

service continuity that is ideal for mission critical applications where operation must always be guaranteed in any situation.

Definitions and Concepts

- Active Server: this is the station that in normal working conditions manages the plant, communicates with it, collects data and controls it. If this server should stop working due to a fault in its system, one of the inactive Servers will enter into service.
- Inactive Servers: these are the stations (one or more than one) that in normal working conditions permit plant management in redundancy mode by means of sharing variable memory areas. These stations permit independent plant interaction and provide archives that are exactly identical to those in the active station. When a fault is detected in the Active unit,, one of these inactive stations enters into active mode and automatically manages the plant by starting up the driver communication, recording engine and data collection functions to take over control where the primary server left off.
- The Inactive Server Drivers are kept on stand-by and do not communicate directly. They operate by receiving and sending variable value notifications to the Active Server and other servers. This is done in complete automatic and transparent mode. As a consequence a command can be invoked towards the field from any one of the Servers indifferently, while a change page operation, for instance, is performed locally as each unit processes its own graphics.
- The Inactive Server historicals (Historian, Data Logger, Recipes, Historical Log) do not operate directly in order to ensure the absolute sameness of recorded data. The appropriate system's redundancy functions perform in such a way so that data which is collected and recorded by the Active Server are archived in the same identical and transparent way on the other Servers as well. The synchronization mechanism always guarantees integrity and time precision of data.
- The Alarm state of the Active Server is redundant in all the other Servers. Any Ack. and Reset command executed on any one of the Servers will be transmitted to the Active Server which will perform the action.
- The Event list and Events on Variables of scripts are not executed in Inactive Servers.

Supported Functions

The Fault Tolerance functions integrated in Platform.NExT provide redundant management of the following functional modules:

- Communication Driver Manager
- Historian and Data Logger Manager
- Historial Log Manager
- Alarm Management

Unsupported Functions

Recipes are currently not supported (at work).
Below is a list of the other unsupported functionalities whereby 'unsupported' means that the functionality is executed autonomously and individually by the relative server without automatic synchronization.

- Event Objects
- Alarm Dispatcher
- Schedulers

Logic and Variables

A system variable is available on the Server that manages information such as local logic when the Redundancy Server is 'active' or 'inactive'.

List of Servers

Even though the Platform.NExT redundancy function can support up to 64 servers, Progea can only **guarantee its functionality up to 4 servers** for the time being.

The names of the servers to be used in the projects must be inserted in the " names should be inserted by means of using the “Redundancy Server” settings. The order in which the servers are inserted on the list is very important as this defines their order of priority which will be used to identify the Active Server.
During normal working conditions all the servers on the list are started up and operative in the plant as configured.
The Active Server will be the first one on the List and therefore in charge of managing driver communications and logging data on the Data Base according to the normal running of each project. The other Servers on the list are available and operative to perform the same functionalities as the Active Server but independently and according to their own configuration.

Time Synchronization

It is important to consider the redundant Server system's time synchronization for greater congruity of historical data synchronization. Each PC is automatically synchronized with the Microsoft Windows main server clock according to the operating system's automatic functionality.
However it is also possible to synchronize the time of each Server with a Server Time that is different to that of windows for default. This can be done by changing the settings in the "Internet Time" controls of the Operating System's Settings.

2. Redundancy: Server

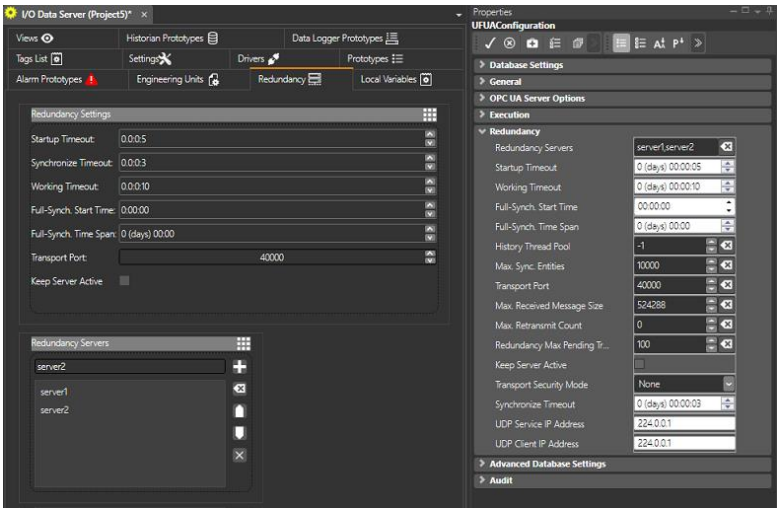
The Redundancy Settings can be configured by means of using the 'Redundancy' resource which is available in the "I/O Data Server" group settings. However, some of the Redundancy's advanced setting parameters are only available in the Properties Window in the specific 'Redundancy' group.

For further information please see the chapter headed: "Advanced Server Settings"

The Server's Redundancy functionality is based on the UDP Multicase protocol for which it is necessary to set the Gateway's IP address in the network form.
In cases in which the project is run on PCs without internet access, and therefore the use of the Gateway is irrelevant, you will need to set the Gateway's IP address in the network form with any IP address of the same subnetwork.

The Server's Redundancy functionality is only supported for PCs that belong to the same subnetwork (those with the IP addess of the same subnet mask).
Therefore, It is not allowed to use this functionality if the PCs are connected to each other by means of a router.

The parameters available are:



Redundancy Settings

Redundancy Servers

This list of Servers participating in the redundancy are inserted in this box. Therefore the Host Names of the machines in question are to be inserted here.
Caution: You must insert the machine's Host Name and not its IP address. This is due to the fact that each machine often has more than one IP address and because the Redundancy manager always refers to the machine's name.

The order in which the servers are inserted also determines the priority of which Server will act as the system's Active Server.
The first server on the top of the list of those started up and available will be the Active one which will have complete control of the system.

Attention! The hostname must be inserted in the Server list and NOT the IP address.
In cases where the project is run on a PC in a network where a DNS Server is not available, you will need to set "<P_Address> <PC_Name>" aliases in the Windows 'Host' file to resolve the names of each PC involved in Redundancy.

Startup TimeOut

Maximum time out used for searching for other servers in the network using the udp protocol.
When the redundancy starts up it searches the network for those endpoints which satisfy the redundancy requirements and whose host name is on the list configured in the project currently running.

Synchronize TimeOut

Maximum time used to wait for messages that the inactive server exchanges with the active server to maintain synchronization.
These messages travel through the tcp channel and include the request for the initial tag and alarm values as well as the list of historical logs to keep synchronized.

In addition the same time is used as a ping interval that the inactive server uses to see whether the active user is still active.

Redundancy Timeout

Timeout used for messages that the active server sends to other servers through the udp channel (for instance when a tag or alarm changes).
This time is also used at the redundancy startup as the maximum timeout between the start and notification that the redundancy has started. If nothing initializes within this time an error is displayed.

Full Synchronization Start Time

This is the startup time for the entire database. During the startup and shutdown phases of several drivers data misalignment may occur between the various servers. Full synchronization will realign the databases if this should occur. When the "Full Synchronization Time Span" is set with another value that is not zero, this property will be ignored and the "Full Synchronization Time Span" will be used only.

Full Synchronization Time Span:

This setting is used for defining how often each full database synchronization is to be performed.

History Synchronization Thread Pool

The number of Threads to be used for handling the Historical synchronization. The "-1" value means that the same number of threads will be created to match the same number of existing CPUs or CPU Cores. Historical synchronization performance will increase when the number of threads is increased but this will effect the performance of other functionalities such as communications and animations for instance.

Max. Sync. Entities:

This parameter involves the synchronization of the historicals and corresponds to the maximum number of records that are read by the active server for each synchronization job done by the other servers.

Transport Port

Port used for transporting the redundancy functions.

Max Received Message Size (bytes)

This is the maximum packet size that can be exchanged for synchronizing tags.

Max Retransmit Count

Number of retransmissions of udp packets sent by the active server to send tag and alarm updates to the other servers.
The udp protocol does not control whether the packet has arrived at its destination to eventually resend it if it hasn't. This parameter however allows the packet to be sent several times to reduce any eventual problems of recipients not receiving it; contrary to this by setting this parameter with a value other than zero will compromise the redundancies performances in sending information.

Max Pending Transmit Message

Maximum number of pending messages. When the number of messages to be sent to Inactive Servers exceeds this limit, those exceeding this limit will be discarded. This is used only for updating tags in realtime. It concerns the maximum number of changes manage for each tag. If the number of changes to be sent exceeds the one send in this parameter, the oldest changes are removed leaving those most recent.

Keep Active Latest Server Active

This option has effect **only when there are two servers on the list** and is used for keeping the server which has control active while the other server reenters into operation after a fault.

- When this option is enabled and there are only two Servers on the list, priority will not be given to the next active server on the list to take control. The control will remain with the server currently active when the other one returns into operation. For example, if a fault occurs and causes the first active server on the list to crash, the second server on the list will activate and take over full control. However when the first Server returns into operation, it will not take over full control until a fault occurs causing the second server to crash.
- When this option is disabled, or there are three or more servers on the list, this option will be ignored and priority will be given to the one according to their

position on the list. When the highest listed Server returns active, it will take over full control.

In cases when enabling the 'Keep Server Active' option, and only two Servers are listed, a switch over to the active server can be done using the "RedundancySwitchActiveServer" command.

Transport Security Mode

Security level to be used for when transporting data. The options are:

- None
- Transport
- Message
- TransportWithMessageCredential

UDP Service IP Address

The "Listening" Area for UDP packets. The 224.0.0.1 value indicated the group consisting of all the LAN network Hosts.

UDP Client IP Address

Defines the area for sending UDP packets. The 224.0.0.1 indicates the group consisting of all the LAN network Hosts.

3. Redundancy: Client

The Platform.NExT Client module, whether for local or remote server visualization, can always connect automatically to any one of the Servers started up in the redundancy system. This functionality is intrisic to each Platform.NExT Client and does not require any additional option to be activated on the license.

- Even though run as Client only, the project is defined with the Server List as previously described.

  At the startup the Client checks the availability of a Local Server connection using the "net.pipe" transport if enabled in the project. If a Local Server is not available, the Client will check for a Remote Server connection using the network transport configured in the project (eg. "net.tcp").

  The choice of Server to connect to is based on the Load Balancing which is revealed upon connecting to the server.
  Therefore when connecting to a Server, each Client checks for a server with an 'optimal' workload situation. The Client will first try to connect to the machine's local Server if present by using the "net.pipe" transport when enabled in the project. The moment in which the Server becomes unavailable locally, the Client will try to connect to a remote server using a network transport such as "net.tcp" that must be enabled in the project. The choice of Server is based on
  the amount of workload detected when connecting.
  When connecting to a Server, each Client must verify which server is the most appropriate to use at that moment before choosing the best one to connect to. This will ensure an effective distribution of workload in each server, above all in those situations where a lot of Clients are in operation, in order to obtain the best performances of the whole system at large.
  After the Client has connect to a Server, the connection will be kept active until that Server is no longer available. In this case when the connected Server
  becomes unavailable (e.g. go into Fault), the Client will automatically try to activate an connection in on of the other redundant Servers available in the network.

  The Automatic Connection and Load Balancing technology are both available in Platform.NExT to ensure continuous Client Station service that is automatic and transparent to the user. At the same time this technology optimizes data traffic settings contributing to system efficiency and its performances.

INDUSTRIAL AUTOMATION SOFTWARE

progea

Progea Srl
Via D'Annunzio, 295
I-41123 Modena
info@progea.com
Tel +39 059 451060

Progea International SA
via Sottobisio, 28
6828 Balerna (CH)
international@progea.com
Tel +41 91 96 76 610

Progea Deutschland GmbH
Marie-Curie Str., 12
D-78048 VS Villingen
info@progea.de
Tel +49 (0)7721 99838 0

Progea North America Corp.
2380 State Road 44, Suite C
Oshkosh, WI 54904
info@progea.us
Tel. +1 (888) 305-2999