



Customers generally say the PI System is one of the most reliable components of their operations software infrastructure. They also say that the longer they have PI, the more problems they discover that can be solved using it. The more they use it, the more important it becomes to their operations strategy. Finally, the more important it becomes, the greater is their concern about the risk that something might happen to preclude access to the data or, worse, to prevent data from being collected for some period of time.

Consider the simplest of PI configurations, one node collecting data - through one of the hundreds of PI interfaces – connected to one node that is the PI server.

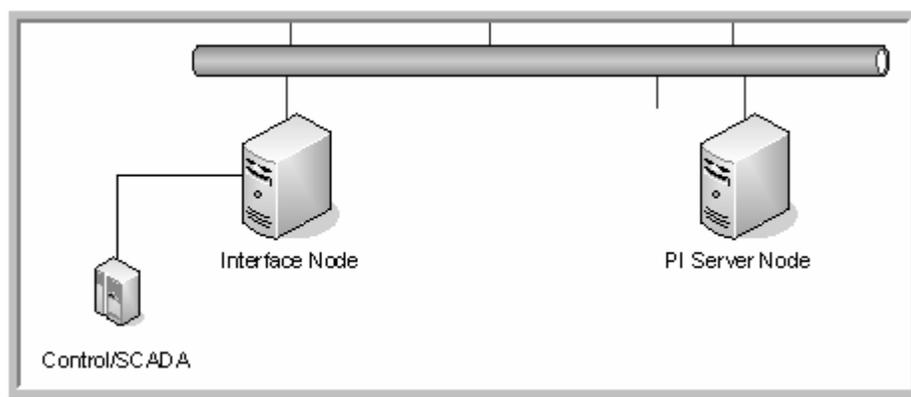


Figure 1 - Simple PI Hardware Configuration

Even in this simple configuration, there are a number of possible conditions that could trigger data loss or render data inaccessible. The most obvious of these is planned maintenance. Taking the Interface Node down for service blocks data from being gathered from its source and creates a gap in data stored that may not be acceptable. Taking the PI Server Node down for service does not prevent data from being gathered due to the design of the OSIsoft Interface software that buffers collected data while the PI Server is unavailable. Still, data is not available to the clients while the PI Server is down for maintenance. Because the circumstances of planned maintenance can be controlled, the impact can be minimized, but generally not eliminated entirely.

Unplanned downtime represents the major risk. Cables are exposed, network issues can crop up, system and even software bugs may arise that can bring down a system momentarily or until such time as the failure is detected and repaired ... which could extend to several hours. Generally, exposure is similar to that of planned maintenance, with a couple of exceptions. If the Interface Node has collected data into its buffers, but not yet transmitted those to the PI Server, any failure bringing down the Interface Node can cause permanent loss of data from operations. Any permanent failure of the PI Server storage will result in loss of any data that had not been previously backed up. There may be permutations of either or both of these failure modes that can also have serious impact to the system's ability to store and/or recover data for users.



Depending on how important these data have become to customers, they have taken certain steps to minimize potential impact using various tools provided by OSIsoft. The following is a sampling of those steps.

- Periodic backup of the PI archive is the most obvious; a capability that PI has had since the first release, although until version 3.4.370 the PI server had to be temporarily unavailable. Now, on-line backups can be done without requiring that any part of the Server be down. This is sufficient to assure that data is preserved and restored periodically, but does not provide alternative storage or access when the Primary Server is down.
- The PI to PI Interface has often been used by customers to save critical data to another PI Server with some; and remains a valuable tool for distributing data points among different archives. This is essentially an alternative backup strategy, for when the Primary Server is down, the Secondary Server cannot receive data.
- Some customers have turned to fault tolerant solutions, such as Stratus or Marathon, to reduce the risk of down-time on their valuable RtPM applications. They provide some degree of protection, although they are designed specifically to protect from hardware failure. For example, Stratus has a single point of failure, the Stratus operating system.
- Microsoft clustering technology has also been implemented by some customers to reduce risk of data loss, but this alone does not assure that data will always be accessible or that it cannot be lost. For example, when a clustered server goes down, any data it has that has not been flushed to disk will be lost. Moreover, the time between a server rolling over and another coming up could span several minutes; time when data is not accessible.

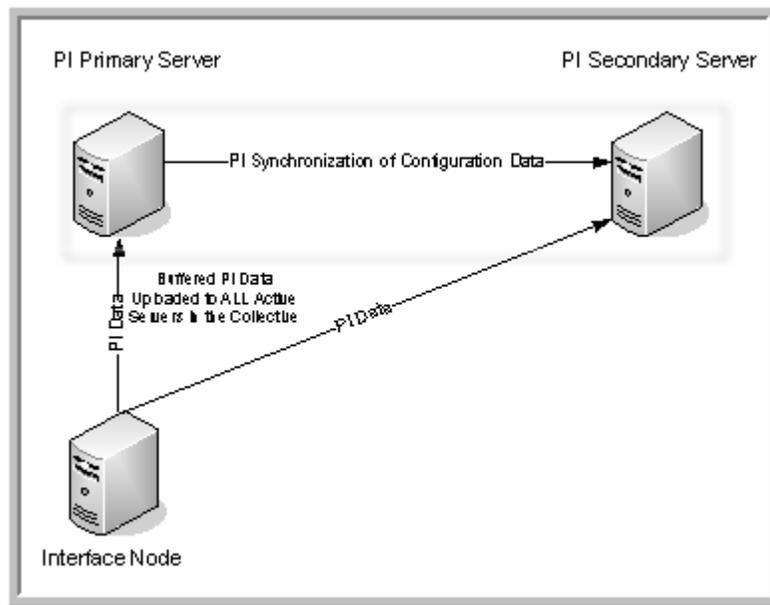
As customers have realized the strategic importance of PI, the configurations represented in its deployment tend to be more complex than is represented in Figure 1, potentially introducing numerous additional points of failure; increasing the risk that some or all of the RtPM data may not be available to key users for critical decision making, or that it may go uncollected altogether. What customers want, what you have asked us to provide is essentially non-stop computing that assures that PI data will always be gathered and will always be available for use by decision makers in your organization.

HA is embodied in two concepts that have been implemented across the OSIsoft server and client products as part of the recent PR 1 release (announced January 2007). The High Availability concepts, which will also appear in all future platform releases¹ are:

¹ OSIsoft announced to its customers at the 2006 Annual User Conference that, henceforth, it will be releasing all software in platform cycles, rather than individual product. This is to assure that features which touch multiple servers and clients in the product mix will be released together, rather than only when the relative products are released.



1. PI Server replication, which allows for redundant PI Servers, including a primary and one or more secondary servers, together referred to as a “collective”. The PI server point database, module database, user database, trust table and most of the configuration tables will be replicated across the collective.



- All interfaces will write time-series data directly to (all members of) the collective, buffering data temporarily for those unable to receive data for a period of time and assuring that time-series data stored in each archive is an exact duplicate of the others.
- Any SDK based PI client (e.g. ProcessBook, DataLink) will be able to automatically switch from the primary (or preferred) PI Server to any of the replicated servers in the event connection to the primary is unavailable; assuring that all clients will always have read-access to PI data.
- Any configuration changes can only be written to the Primary Server of the collective, where a configuration change log is maintained. Periodically the Secondary Servers review the change log and update their configuration data accordingly. If the Primary Server is down, configuration data cannot be changed and the user attempting to make changes will be given an error message. In the event that the Primary Server cannot be restored or the time to restore it exceeds acceptable time-frame, there is a simple manual procedure to promote a Secondary Server to the Primary Server role.



2. PI Server Interface failover and startup without connection to the PI Server:

- Changes to the PI Interface design to accommodate HA are included in the PR1 release. With this release, you can have a pair of PI Interface nodes connected to a PI Server or to the Collective. If the Primary Interface Node fails to deliver data to the PI Servers, it will fail over to the secondary PI Interface, running in hot standby mode.
- Additionally, PI Interfaces can now be started without a connection to a PI Server; accomplished by maintaining a local point cache as part of the UNIINT implementation.

These changes provide assurance that data from Interface Nodes will always be able to be stored into, and standard SDK based clients will always be able to retrieve data from, the PI archive. Subsequent releases will extend the model, making it more automatic and providing PI Replication benefits to some of the less often used features (e.g. VBA add-ins used to write data), etc.

The chart below is a high level view of the individual client tool's limitations with this HA release during such a failover scenario. In PR1, all clients will behave as normal in an HA environment while connected to the Primary server. After a failover event to a Secondary server some limitations may occur for several of the clients. Most limitations involve writing of data by the clients. For example, no client can write configuration data when the Primary Server is not available. Interface nodes will be writing data to all servers in a collective at all times and buffering data for those Servers in failure, but clients that write data should be in a read only mode after a failover event.

Questions and comments can be directed to: HA@OSIsoft.com.



| Client | PR 1 | | PR 2 | |
|-----------------------|-------------|----------|-------------|----------|
| | No Failover | Failover | No Failover | Failover |
| ACE | ✓ | ● | ✓ | ● |
| ActiveView | ✓ | ● | ✓ | ● |
| AF Explorer | ✓ | ▲ | ✓ | ● |
| AF Modeler | ✓ | ▲ | ✓ | ● |
| AlarmView | ✓ | ◆ | ✓ | ● |
| APS | ✓ | ● | ✓ | ● |
| BatchView | ✓ | ▲ | ✓ | ● |
| ControlMonitor | ✓ | ▲ | ✓ | ● |
| DataLink (w/ PutVal) | ✓ | ▲ | ✓ | ● |
| DataLink (wo/ PutVal) | ✓ | ● | ✓ | ● |
| DataSheet | ✓ | ◆ | ✓ | ● |
| DevNet | ✓ | ▲ | ✓ | ● |
| ICU | ✓ | ▲ | ✓ | ● |
| ManualLogger | ✓ | ◆ | ✓ | ● |
| MDB Builder | ✓ | ▲ | ✓ | ● |
| Performance Equations | ✓ | ▲ | ✓ | ● |
| ProcessBook | ✓ | ● | ✓ | ● |
| ProfileView | ✓ | ◆ | ✓ | ● |
| RtReports | ✓ | ▲ | ✓ | ● |
| RtWebParts | ✓ | ● | ✓ | ● |
| Sigmafine | ✓ | ▲ | ✓ | ● |
| SMT | ✓ | ▲ | ✓ | ● |
| TagConfigurator | ✓ | ▲ | ✓ | ● |
| VBA (Custom) | ✓ | ▲ | ✓ | ● |

| | | | | |
|-----------------|---|---|---|---|
| Interface Nodes | ✓ | ● | ✓ | ● |
|-----------------|---|---|---|---|

| Legend | |
|--------|--|
| ✓ | All functionality is preserved. |
| ● | All functionality preserved, except configuration data cannot be written when the Primary Server is in failover. |
| ▲ | Acceptable limitations: client cannot write archive or configuration data while the Primary Server is in failover. |
| ◆ | Undesirable limitations: read only while Primary Server is in failover; significant client functionality is curtailed. |