Industrial Automation and Control
Automation College

uniformance

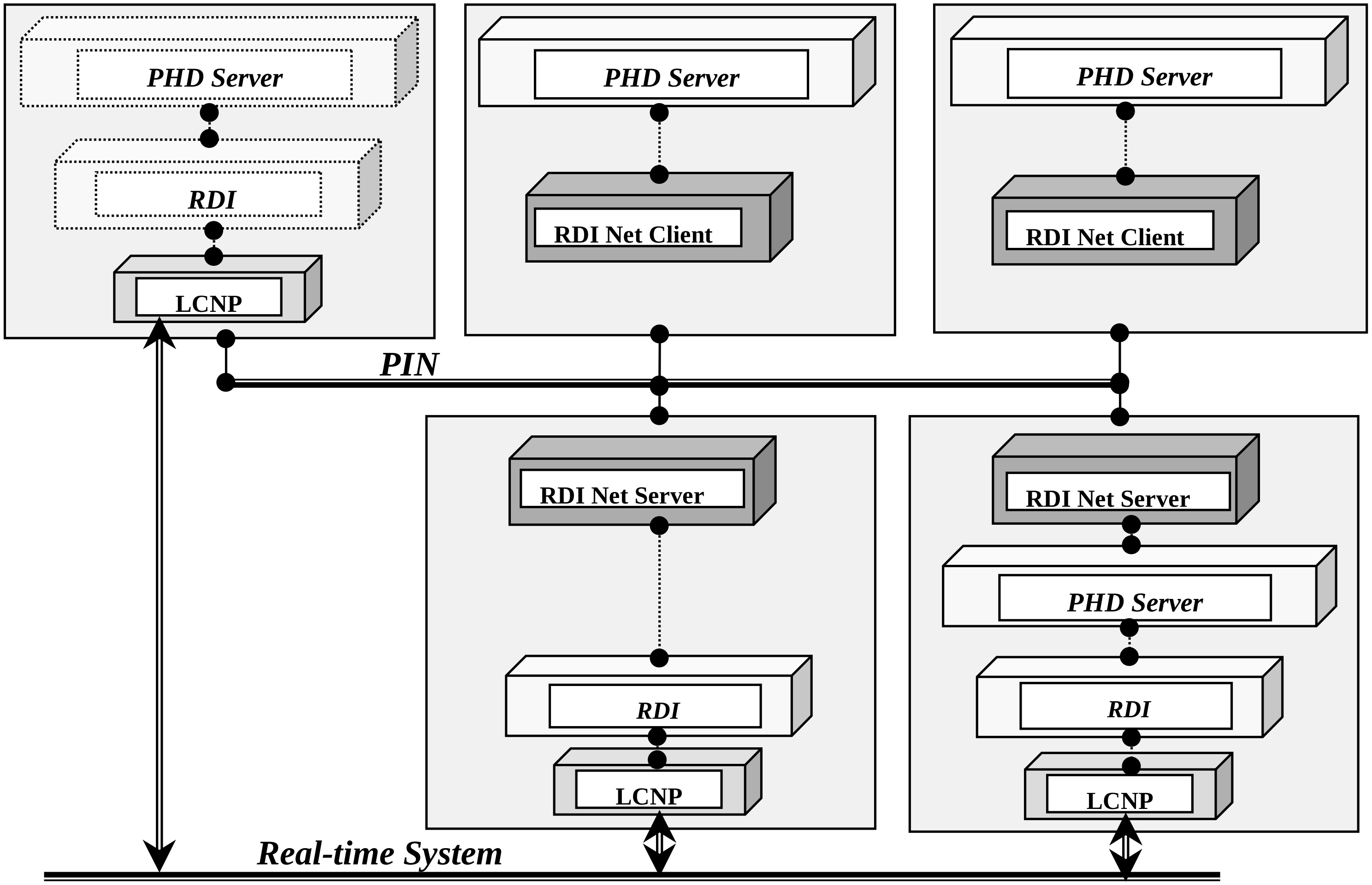
PHD RDI Management

Lesson Objective

Objective
- Manage, monitor, and support PHD real-time interfaces

Topics
- RDI Client/Server Components
- RDI Configuration/Startup
- RDI Performance/Offset RDIs
- RDI Parameter Configuration Form
- RDI Specification Form
- Remote PHD interfaces
- Remote Shadow Interface
- Demo/Hands-on Exercise

RDI Client/Server Components



RDI Configuration/Startup

After planning a site's RDI requirements, a qualified installer would do the following to configure and startup an RDI.

1. Enter data into the RDI Specification Form to define the interface.

RDI Name
Source System
Host Name (see form definition in *PHD User Guide*)
Enabled?
Remote?

2. At the DOS prompt, enter the following command to create interface control procedures (such as GO and KILL files):

RDI_SERVICES user/password@oracledbname
ex. RDI_SERVICES totalplant/totalplant@totalplant

Set parameters for the new RDI by responding to the prompts.

Use remote system RDI for timestamps? Y/N
Set offset for scan cycle intervals (seconds).
(You have to answer questions for any other existing RDIs, so you must have a record of their configuration in order to match the entries.)

3. Using Notepad, verify the interfaces.dat file.

Execute the interfaces.dat file, then execute the start_interfaces.dat file.

Refer to - *PHD System Manual,* RDI Services Utility
- *RDI Functional Specification*

RDI Performance/Multiple RDIs

The parameter per second rate is a function of the underlying hardware and software interface between the RDI and the real-time system.

The par/sec rate cannot be increased by having multiple RDIs, unless those multiple RDIs consist of more hardware connected to the real-time system (for example, another TPS node, running a remote LxS RDI).

Honeywell “gateway” performance:

- CM50S—Approximately 240 to 260 parameters per second.
- $L^{X}S$—350 parameters per second for most PV tags on LCNP2

256 is the maximum number of RDIs a PHD server can handle.
There is no maximum number of points per RDI.

Most PHD systems, except for the smallest plants, need several RDIs. There are various reasons for dividing tags between multiple RDIs.

The Honeywell TotalPlant Support Online web site has an article on "Minimizing the Impact of PHD Collection on the LCN/UCN":

- http://www.iac.honeywell.com/Pub/Mktg/support.stm
  - Unifomance>Technical Support>Knowledge Base>Article ID: 11912

RDI Performance/Multiple RDIs, *continued*

Purpose of multiple RDIs
To ensure tag scan frequency

With multiple offset RDIs, you can ensure that the system achieves the desired scan frequency for all tags.

Example:
If you have 3000 tags on a 30 second cycle and 5000 tags on a 60 second cycle, then 8000 tags would be collected on the 60 second cycle in 27 seconds (3000 + 5000/300).

Taking 27 seconds to collect the data puts it too close to the next 30 second cycle time, so you should create two RDIs.

Other considerations:
PHD may be competing with other applications for the resource. If an application (such as a blending application) runs once a minute, you may want to offset the RDI.

RDI Performance/Multiple RDIs,*continued*

Purpose of multiple RDIs, *continued*

To ensure tag scan frequency, *continued*

Create separate RDIs for fast and slow tags. “Slow” can be defined as tags >60 seconds, and “fast” as tags <= 60 seconds.

Identify key tags that should have a fast scan frequency (such as 5 seconds), if any. Try to assign all of the fast tags to the same RDI, so it is more efficient.

The ability to collect tags on high frequency is totally dependant on the overall throughput capability of the device serving data to the RDI; that is, the gateway and/or its communications ability.

When checking out “fast” tags, check the timestamps in the queue to see if you are getting a value at the desired frequency (such as 5 seconds). If you are getting a value only every six seconds for example, the gateway is not handling the load.

If you have a lot of tags doing Put Downloads, create a separate RDI for putting values.

RDI Performance/Multiple RDIs, *continued*

Purpose of multiple RDIs, *continued*

To balance the load on the DCS system

Create multiple offset RDIs to help balance the load on the real-time system network.

For example, if you are getting all the data from one NIM, you may overload the NIM with requests if you do not put the data into two RDIs with an offset.

Creating RDIs per AM unit, per NIM, per HPM helps the RDI performance for retrieving data from the LCN; however, at some point, increasing the number of RDIs no longer helps.

Another good way to divide data is by data type. Strings and enumerations take longer to collect than reals and ordinals and integers, so if you group RDIs by data type, you can help the RDI performance.

RDI Performance/Multiple RDIs, *continued*

Purpose of multiple RDIs, *continued*

For more efficient tag maintenance

The fewer tags you assign to an RDI, the faster it will initialize.
For a large number of tags, the time to initialize can be, for example, over three hours.

If you split tags between multiple RDIs, then when you need to initialize one RDI, tags in the other RDIs continue collecting and historizing.

You may decide to initialize an RDI after deleting and re-adding LCN tags. It is the easy way to get PHD to recognize the new internal LCN ID for the tags. The alternative is to rebuild all tags that changed in PHD, or to do a complete coldstart.

<u>To initialize an RDI</u>, shutdown the RDI, turn off the RDI data cache parameter, then restart the RDI.

```
PHDMAN SHU intname
PHDMAN SET intname:RDIDATACACHE 0
PHDMAN STA intname
```

Tag Management Scenarios

Example 1

Task: Due to processor and/or memory requirements, a Unit (involving hundreds of tags) was moved from one Honeywell Application Module to another.

Effect: All tags in the Unit were deleted and re-added to the LCN.

PHD Decision: *Which of the following PHD options should you use*?

A. Do an RDI by RDI restart of affected RDIs.

B. Do a complete cold start.

C. Use the following “change tagname” method:
Go into TPI, change the PHD tagnames for the Unit tags to dummy tagnames that do not exist on the LCN (so that data is not collected inadvertently), then send the changes to PHD. All of the tags will re-initialize and fail. Go back into TPI, change the tagnames back to the proper ones, then send changes again. The tags will re-initialize again, and start collecting normally.

Answer: The RDI by RDI method is the best choice.

Tag Management Scenarios, *continued*

Example 2

Task: The tags in a Honeywell hiway device were moved to an HPM.

Effect: Six tags were deleted and re-added to the LCN.

PHD Decision: *Which of the following PHD options is the best choice*?

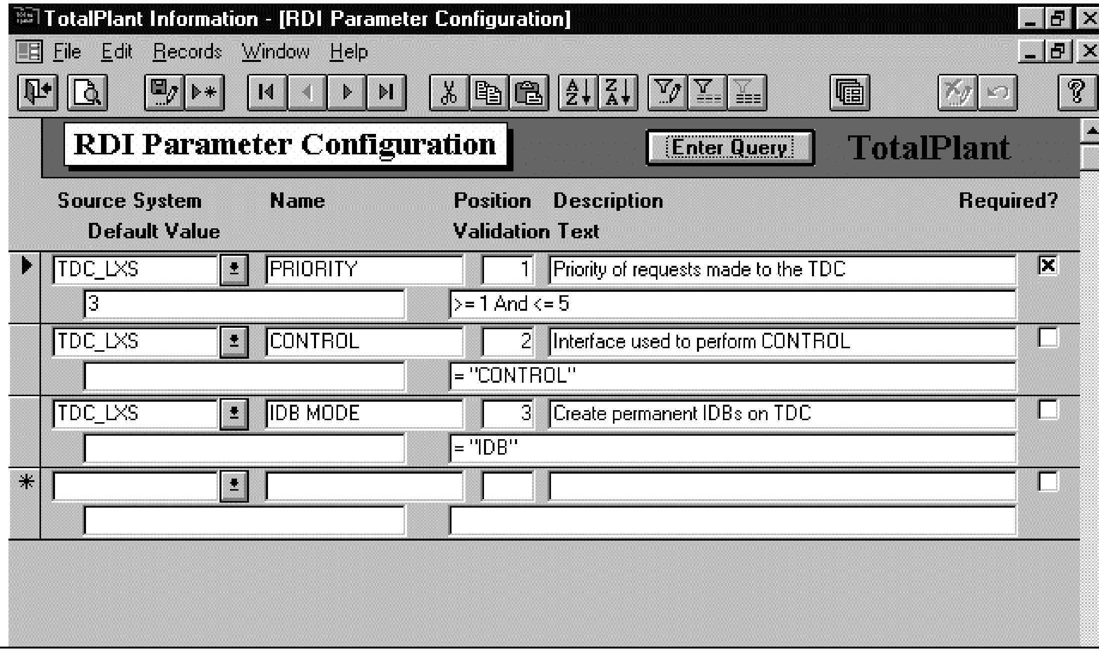
A. Do an RDI by RDI restart of affected RDIs.

B. Do a complete cold start.

C. Use the following "change tagname" method:
Go into TPI, change the PHD tagnames for the Unit tags to dummy tagnames that do not exist on the LCN (so that data is not collected inadvertently), then send the changes to PHD. All of the tags will re-initialize and fail. Go back into TPI, change the tagnames back to the proper ones, then send changes again. The tags will re-initialize again, and start collecting normally.

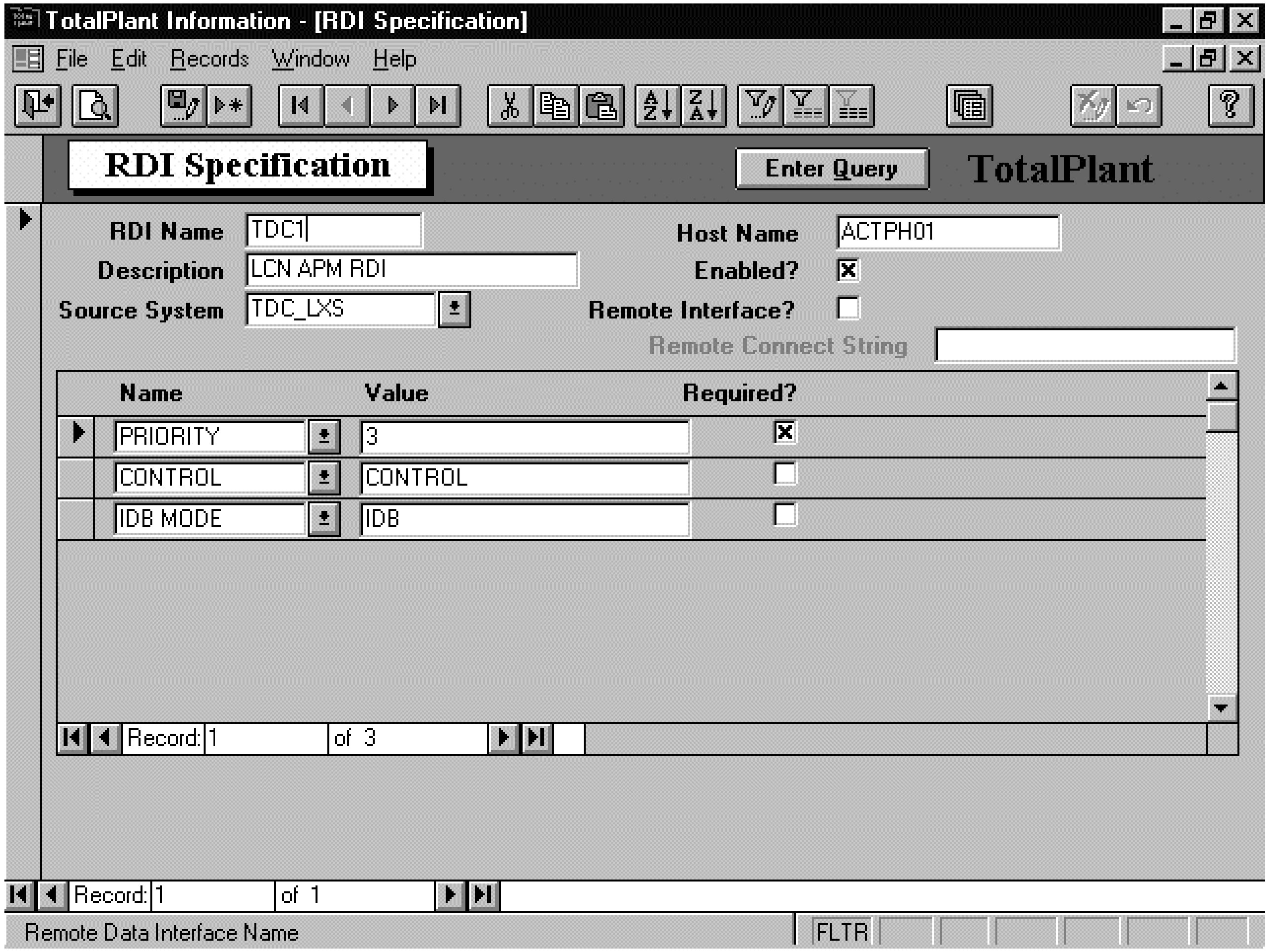
Answer: The "change tagname" method is feasible in this case.

RDI Parameter Configuration Form



References: *Remote PHD RDI Functional Specification*
TPN/A^{X}M (L^{X}S) RDI Functional Specification
CM50 RDI Functional Specification
PlantScape RDI Functional Specification
PHD User Guide, RDI Parameter Configuration, RDI Specification

RDI Specification Form



PHD Shadow System

A remote PHD interface is used to gather and store process information from secondary PHD Servers.

- **Shadow Interface**
  - The main PHD Server acts as a shadow to a remote PHD.
  - The RDI configured with the shadow option does not do data processing (such as compression).
  - The main PHD Server is a complete PHD system that shares the same tag definitions with the remote PHD Server.

- **Buffer Interface**
  - The remote PHD system acts as a buffer.
  - If communications between the main PHD system and the remote PHD system are interrupted for any reason, all process data is saved.
  - Once communications are restored, the main PHD server retrieves all history values saved during the missing time frame and bring itself up to date.
  - This is automatic and is limited only to the remote PHD system archive system capability. The archive system is usually configured to save at least one month of data.

Refer to *PHD User Guide,* Remote PHD Interfaces

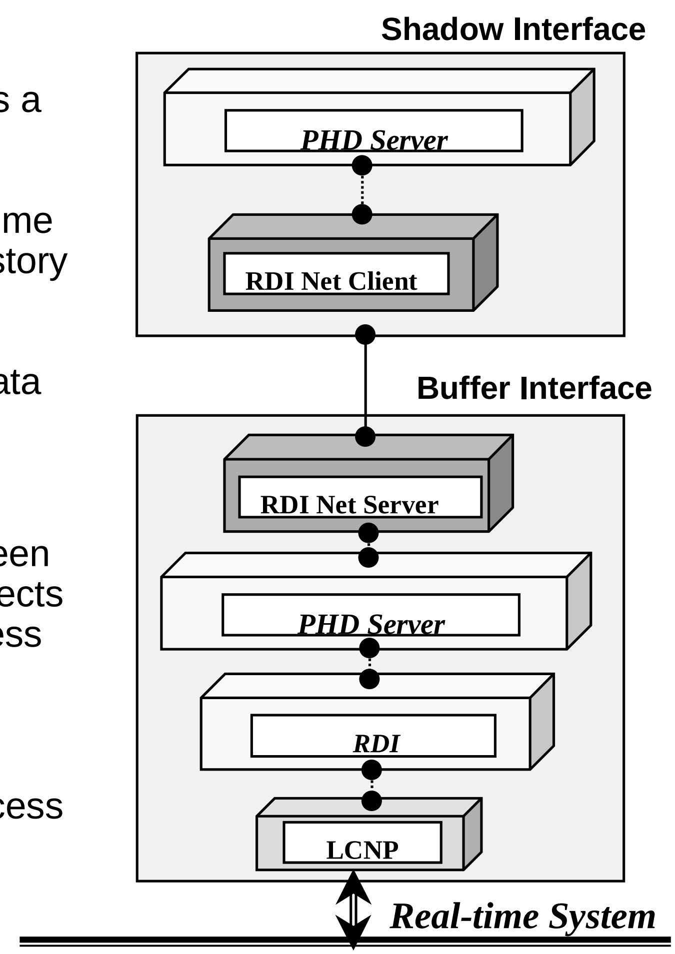
PHD Shadow System, *continued*

- The main PHD RDI accesses the remote PHD system through the use of the internal PHD API routines. The interface is typically implemented as a network client/server pair.

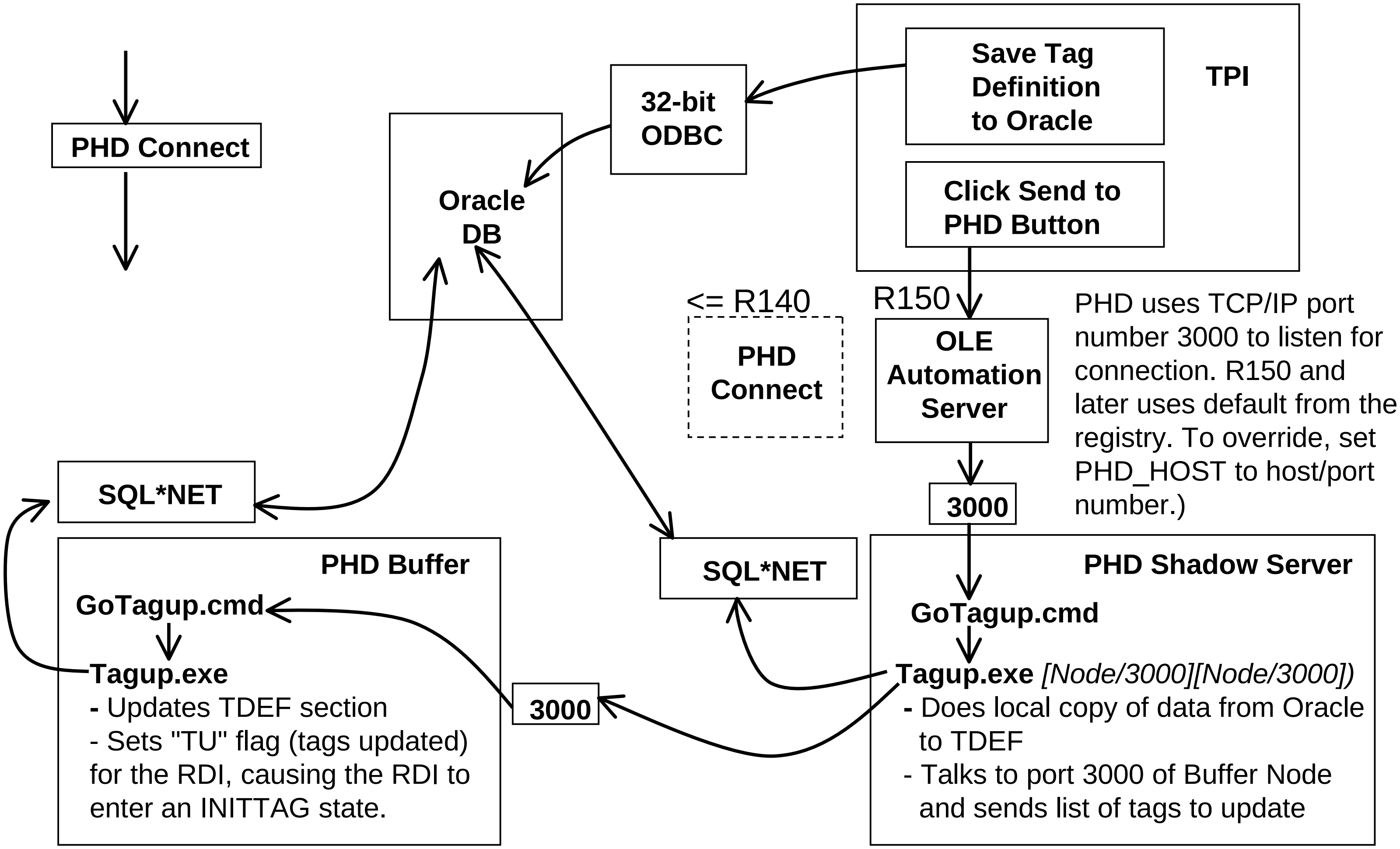
- The remote PHD collects process data from real-time systems and stores the data in its own process history database.

- The main PHD Server periodically retrieves this data and replicates it or shadows the results.

- Shadowing has two main advantages:
  - Distributes PHD processing requirements between multiple servers. (The remote PHD system collects real-time data while the main PHD serves process history data to end users.)
  - Provides a secure layer between the end user environment and the real-time data system environment. (The end user has access to process data without having to connect to a real-time system.)

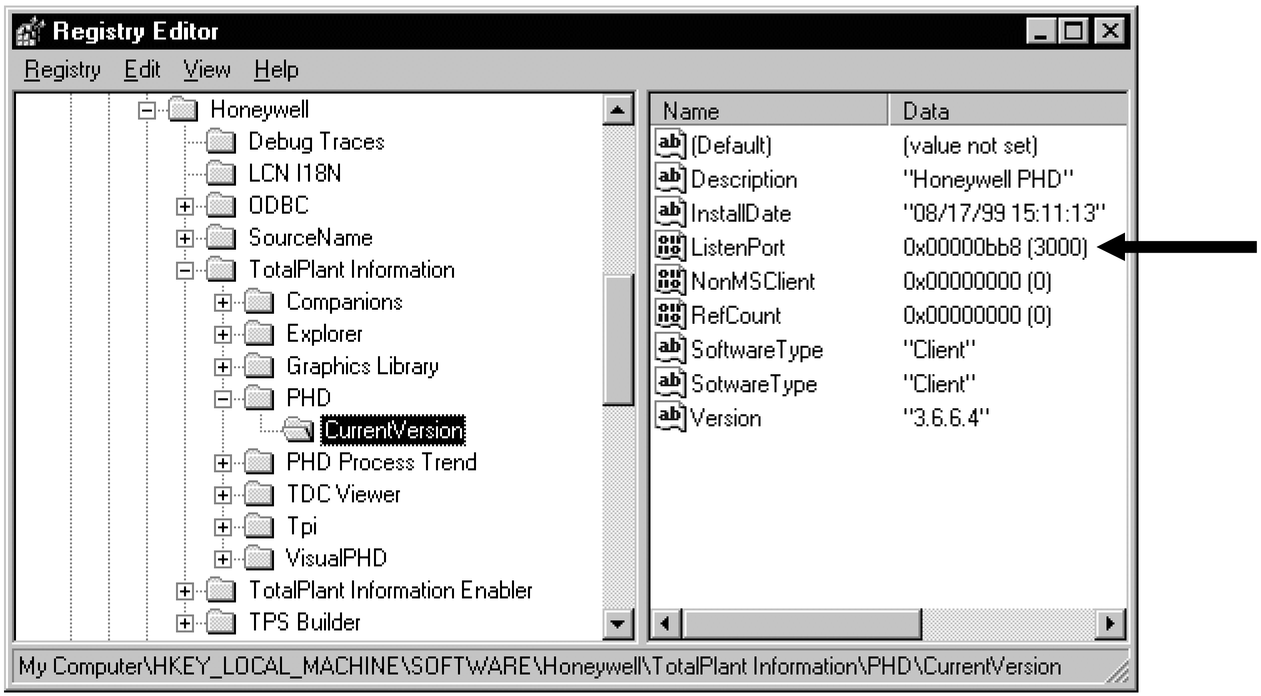


TPI “Send to PHD” Button or PHDMAN “Update Tag” Command

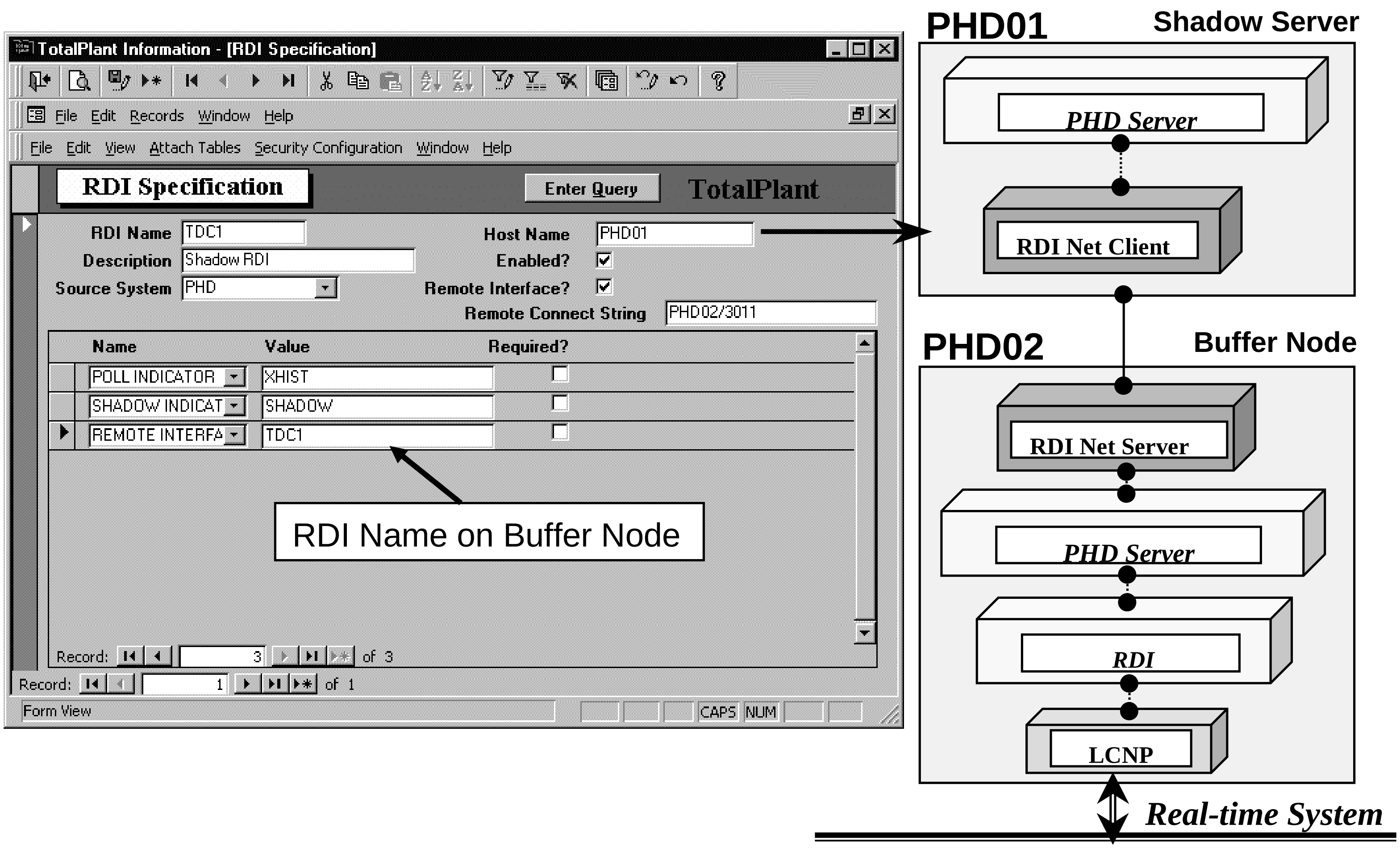


TPI “Send to PHD” Button or PHDMAN “Update Tag” Command, *continued*

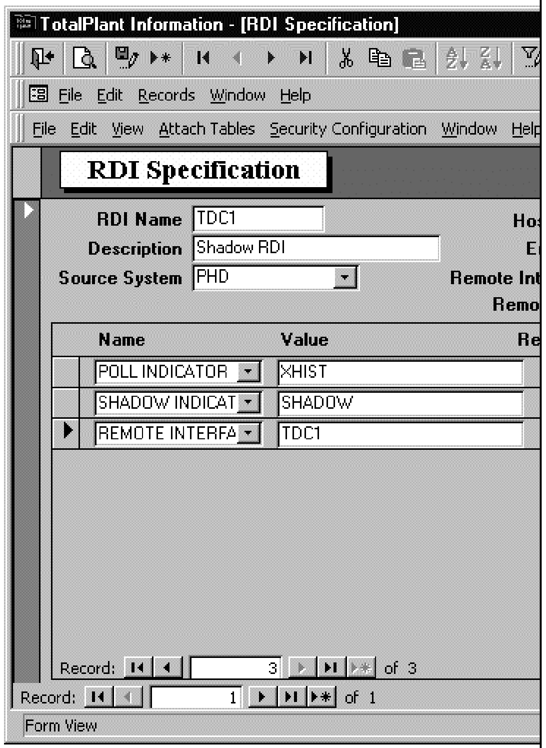
To view the port number that was entered during installation of PHD Client, use regedit:



RDI Specification for PHD Shadow Server



PHD RDI Parameters



There are three parameters for a PHD to PHD RDI.

1. **POLL INDICATOR**
   - **POLL**
     - This option scans the tag on the frequency specified in your Tag Configuration form.
   - **EXCEPTION**
     - This option reports only when tag values change.
   - **XHIST**
     - This option causes tags to be collected in history recovery mode.

2. **SHADOW INDICATOR**
   - Set to Shadow for shadow.
   - Leave blank for Peer-to-Peer.

3. **REMOTE INTERFACE**
   - Set to name of RDI being shadowed for Shadow.
   - Leave blank for Peer-to-Peer.

POLL Option Advantages/Disadvantages

- **POLL**
  - **Advantages**
    - Collects values at same frequency as Buffer
    - Tag timestamps are the same as the Buffer tag timestamps
      - PHD 130.2 and later provide option to use lower node's timestamp
      - **Tag Source Configuration**: Create PHD System Type

System Type	PHD
Tag Type	F
Attribute	Float
Data Type	F
Data Length	8

      - **RDI Parameter Configuration**

System Type	PHD
Name	Scantime
Position	4
Description	Use timestamp from lower PHD
Default Value	(leave blank)
Validation Text	"SCANTIME"

  - **Disadvantages**
    - Higher network bandwidth

EXCEPTION Option Advantages/Disadvantages

- **EXCEPTION**
  - **Advantages**
    - Collects values at same frequency as Buffer
    - Lower network bandwidth
  - **Disadvantages**
    - CPU usage on Buffer is the largest in this mode
    - Only collects from the Buffer the last exception value
      - To prevent this problem, make sure XSCANSECS is equal to half (1/2) the lowest scan rate for your tags
      - The CPU usage is proportional to the XSCANSECS - smaller value for XSCANSECS, the greater the CPU usage

If using this option, note the Interface Parameter ***xautointerval****. Its default value is 1. If an exception value is not received for a tag within the specified number of src_scansec intervals, the previous value is automatically reloaded. If an RDI does not complete its scan of tags, it writes the previously scanned value. A zero value disables generation. If it is determined that his would be a problem, the user should choose to use POLL.*

XHIST Option Advantages/Disadvantages

- **XHIST**
  - **Advantages**
    - Tag timestamps are the same as the Buffer tag timestamps
    - Lower network bandwidth
  - **Disadvantages**
    - Slow on collecting values
    - Collects in History Recovery mode
    - May not contain current values

RDI Configuration on PHD Shadow Server

- **In TPI, build RDI**
  - **Name** (ex: TDC1)
  - **Host Name (your node name)**
  - **Description (your choice)**
  - **Enabled** ✓
  - **Source System** **PHD**
  - **Remote Interface** ✓
  - **Remote Connect String** Buffer Name/Port on which PHD to PHD listens (ex: PHD02/3011)
  - **Other Fields**
    - **Poll Indicator** **XHIST**
    - **Shadow Indicator** **SHADOW**
    - **Remote Interface** (RDI Name on Buffer)

- **Run RDI Services (turn history recovery on)**
  - **Min Hist Rec** **= 1**
  - **Max Hist Rec** **= 10,080** (minutes in a week - Causes server to recover up to a week of history)
  - **XSCANSECS** **= 60** (frequency at which shadow retrieves new values from buffer)

RDI Configuration on PHD Buffer

- **In TPI, build RDI**
  - **Name** (ex: TDC1)
  - **Host Name (your node name)**
  - **Description (your choice)**
  - **Enabled** ✓
  - **Source System** **TDC_LXS**
  - **Remote Interface** ✓
  - **Remote Connect String** (ex: PHD_00/3010)
  - **Other Fields**
    - **Priority** **3**
    - **Control**
    - **IDB Mode** **IDB**
- **Run RDI Services**
    - **Use Realtime Clock** **Y**
    - **Offset** **5**

Modify GoTagup.cmd and GoUserup.cmd files

- **On PHD Shadow Server, modify files by adding “Node/Port Number” for updates**
  - **GoTagup.cmd**

    echo Initiating offline *tag definition* update
    setlocal
    cd /D %PHD_ROOT%\SITETEMP
    RUN -m -o%PHD_ROOT%\SITETEMP\TAGUP.OUT TAGUP “PHD_02/3000”
    endlocal

  - **GoUserup.cmd**

    echo Initiating offline *user definition* update
    setlocal
    cd /D %PHD_ROOT%\SITETEMP
    RUN -o%PHD_ROOT%\SITETEMP\USERUP.OUT USERUP “PHD_02/3000”
    endlocal

PHD Buffer Configuration

- PhdParam.Cmd

PhdParams.Cmd - Notepad

File Edit Search Help

```
PHDMAN SET MAX_USEVALLEN 80
PHDMAN SET MAX_TAGS     10000
PHDMAN SET MAX_TAGNO    25000
PHDMAN SET STORE:INTERVAL 300
PHDMAN SET STORE:PROCNAME CONSTORE
PHDMAN SET STORE:HUNG RESTART
PHDMAN SET STORE:GONE RESTART
PHDMAN SET STORE:LOPRI 7
PHDMAN SET STORE:HIPRI 9
PHDMAN SET MAX_ARCFILEKB 2500
PHDMAN SET MAX_PRCPOOLKB 200
PHDMAN SET DEF_QMINUTE 15
PHDMAN SET PCT_POOLGROW 10
PHDMAN SET MIN_PRCTAGCACHE 70

PHDMAN SET DBMID TOTALPLANT
PHDMAN SET DBMUSER PHD_READONLY
PHDMAN SET DBMPASSWORD PHD_READONLY
PHDMAN SET DBM_TAGFILTER 1
```

Adjust these numbers

1 indicates to filter only those tags assigned to a currently defined initialized interface.

Hands-on Exercise

In this exercise, you will create an RDI, then verify its operation.

1. Using the RDI Specification form, create an LXS RDI named GnnRDI on the training TPS node.

2. Run the RDI services utility to configure your RDI into the system.

3. Using PHDMAN, start your RDI, then show the status of your RDI:

   Execute the `interfaces.dat` file, Then the `start_interfaces.dat` file.

   `SHOW INT GnnRDI`

4. Build a tag on your RDI referencing the PV of DCS tag FIC21941.

5. Verify that the RDI collects values into the queue.

END OF EXERCISE

Honeywell

Helping You Manage Your World

www.iac.honeywell.com