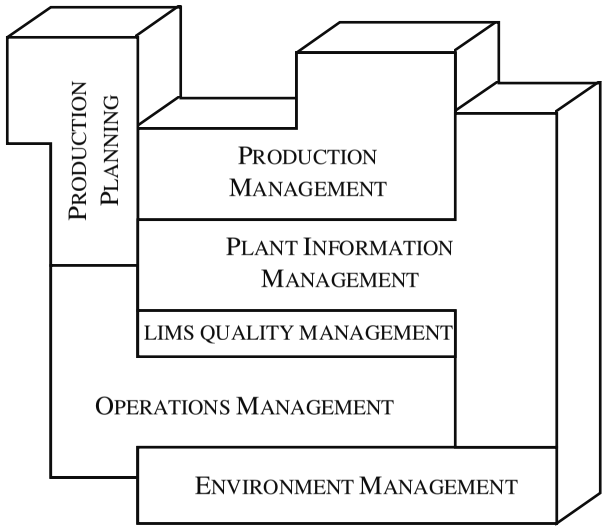
PHD Virtual (Calculation) Tags



Lesson Objective

Objective
- Configure Virtual Tag calculations.

Topics
- What Are Virtual Tags?
- Applications
- Statements
- Expressions
- Operators
- Intrinsic Functions
- Steps to Define Virtual Tags
- Virtual Tag Configuration Report
- Storing Virtual Tag Calculation Results
- PRC (Procedure) Shared Memory Section
- Hands-On Exercise

References:
PHD System Manual, Virtual Tag Definition
PHD User Guide, Configuring Virtual Tags

What are Virtual Tags?

- Virtual tags are tags whose data values are mathematical and/or logical procedures rather than values collected from the real-time system. Calculations can be based on collected process variables, manual input, or other calculations. Each calculation is simply specified using its source tag names.

  Example: TAG1*(TAG2+TAG3)

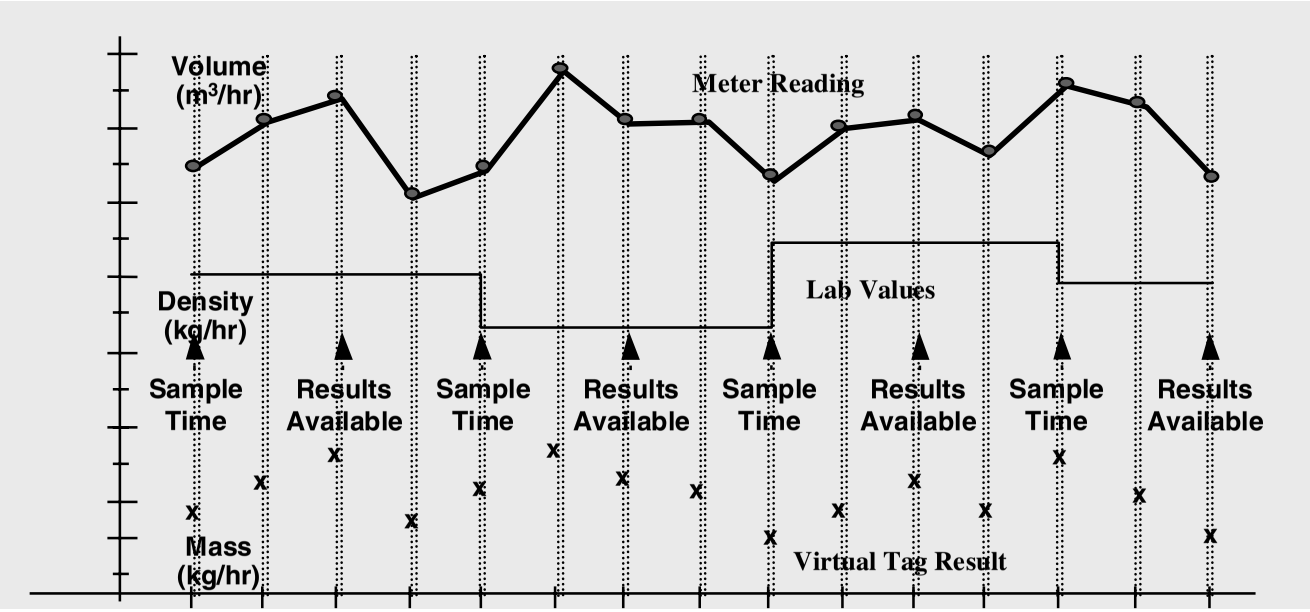
- Virtual Tags return a calculated result along with the calculated confidence for the result based on the reliability of the tags referenced by the calculation.

- All calculated tag values are "demand calculated" at the time a data request is made for the tag. The valid data time range for a virtual tag is bound only by the available range for other tags referenced in its procedure.

Virtual Tag Applications

- Virtual tags can be used for functions such as
  - filling in missing instrumentation,
  - performing continuous mass/volume conversions, and
  - tracking unit and furnace efficiencies or other complex calculations.
- The PHD database provides the capability to perform a calculation to handle situations where lab physical properties required for the calculation are not available for several hours after the sample time.
- The database also provides the ability to evaluate logical time based inferences of process data or other manual inputs from operator logs.
  For example: What was the change in value over the last 5 minutes, hour, or other interval, and did this change exceed a logic operating threshold?

Virtual Tag Applications, *continued*



- In this example, a virtual tag is used to calculate the mass flow based on a meter reading (volume) and a lab sample (density). If you used the traditional method of real-time calculation and storage of values, the answer would always be wrong, because lab values are not available at the same time as the metered values. Only by performing the calculation after the sample is taken can the true value be known.
- If you attempt a query before the lab value is entered, the result returns a zero confidence level.

Virtual Tag Functions

The following functions are supported within calculations:

- Use of "IF THEN ELSEIF ELSE ENDIF" statements for conditional execution

- User-specified reductions for selected time periods such as AVG, DELTA, MAX, MIN, REGSLOPE, STDEV

  Example: A rolling ten minute standard deviation around the current tag timestamp. TFN002{,,-300,300,STDEV}

- Nested function calls to reference standard calculations such as meter corrections, mass conversions, and vessel level to quantity conversions.

Statements

- **RETURN** *expression*
  - Returns the value of the expression as the value of the tag or function.
  - An expression is evaluated as true if its result is non-zero.

- **IF** *expression* **THEN**
  statements
  ELSEIF *expression* **THEN**
  statements
  ELSE
  statements
  ENDIF

Examples

- *Simple text calculation reference tags by name*

```
TAG VIRTTAG1
        RETURN TAG1+((TAG2{"BPH"}*24)+TAG3{"BPD"} )
```

- *Function definitions*:

```
FUNC1(TAG1)
        RETURN TAG1 * 2


FUNC2 (INSWSTS)
        IF INSWSTS = 0 THEN
                RETURN "OPEN"
        ELSEIF INSWSTS = 1 THEN
                RETURN "OPENING"
        ELSEIF INSWSTS = 2 THEN
                RETURN "CLOSING"
        ELSE
                RETURN "CLOSED"
        ENDIF
```

Expressions

An expression consists of a string of identifiers and operators with parenthesis indicating precedence of evaluation.

Identifiers may be:

- Numeric constants (ex: 45.3 or 105)
- Tag specifications - A tagname specification consists of a tagname or a tagname with tag options in the following format.

TAGNAME{ ["units"] [, time_delta_tag] [, start_offset] [, end_offset,"reduction"] }

Examples

A rolling one hour average for the hour prior to the current tag timestamp processed:

Tag1{,,-3600,0,"AVG"}

Tag1000 {"BPH",,-3600,0,"AVG"}

Tag2000 {"KPH",,-3600,0,"REGSLOPE"}

Refer to *PHD User Guide*, Virtual Tag Syntax Examples

Expressions, continued

UNITS Option. Desired engineering units for the tag values. (valid for 'F' data type tags)

TIME_DELTA_TAG Option. A tagname holding a PHD integer time as a value. If specifying a TIME_DELTA_TAG, the value for the tag is the delta between the value of the tag at the current tag timestamp processed, and the value of the tag at the time specified by the TIME_DELTA_TAG value. (valid for 'F' data type tags) Reduction processing may not be specified for tag requests with time delta tags.

START_OFFSET Number of seconds that PHD offsets values for the tag from the current tag timestamp processed. (applies to tags of any data type)

Example: TFN002{,,-60}

PHD uses the value for TFN002 at one minute prior to the current tag timestamp processed. This allows comparisons between values for a tag at various offset times.

END_OFFSET Number of seconds (from the current tag timestamp being processed) that specifies the end of a time range over which a data reduction is performed. (applies to 'F' or 'I' data type tags only)

REDUCTION A text string indicating the data reduction.

"AVG"	Average of time range.
"DELTA"	Value at END_OFFSET minus value at START_OFFSET.
"MAX"	Maximum value in time range.
"MIN"	Minimum value in time range.
"REGSLOPE"	Linear regression slope of values in time.
"STDEV"	Standard deviation over time range.

Operators

A + B	Addition (or string concatenation)
A - B	Subtraction
A * B	Multiplication
A / B	Division
A % B	Remainder of A/B (zero if B is zero)
A ^ B	Raise A to the power of B
A & B	1 if both A and B are non-zero, otherwise 0
A | B	1 if either A or B is non-zero, otherwise 0
A = B	1 if A equal to B, otherwise 0
A != B	1 if A not equal to B, otherwise 0
A > B	1 if A greater than B, otherwise 0
A >= B	1 if A greater than or equal to B, otherwise 0
A < B	1 if A less than B, otherwise 0
A <= B	1 if A less than or equal to B, otherwise 0
! A	Logical negate, 1 if A is zero, otherwise 0
- A	Unary negative
+ A	Unary positive

Virtual Tag Intrinsic Functions

Math functions

- $MIN, $MAX
- $ABS, $EXP, $LOG, $LOG10

Trigonometric functions

- $SIN, $COS, $TAN, $ASIN, $ACOS, $ATAN

String Functions

- $LEN, $PAD, $TRIM, $SUBSTR

Support functions

- $TIME, $CONF

Specialized functions

- $LOOKUP

Virtual Tag Intrinsic Functions, *continued*

Examples:

$MAX(TAG1{,,-300},TAG2{,,-300})

TAG1{,,300,0,”DELTA”}

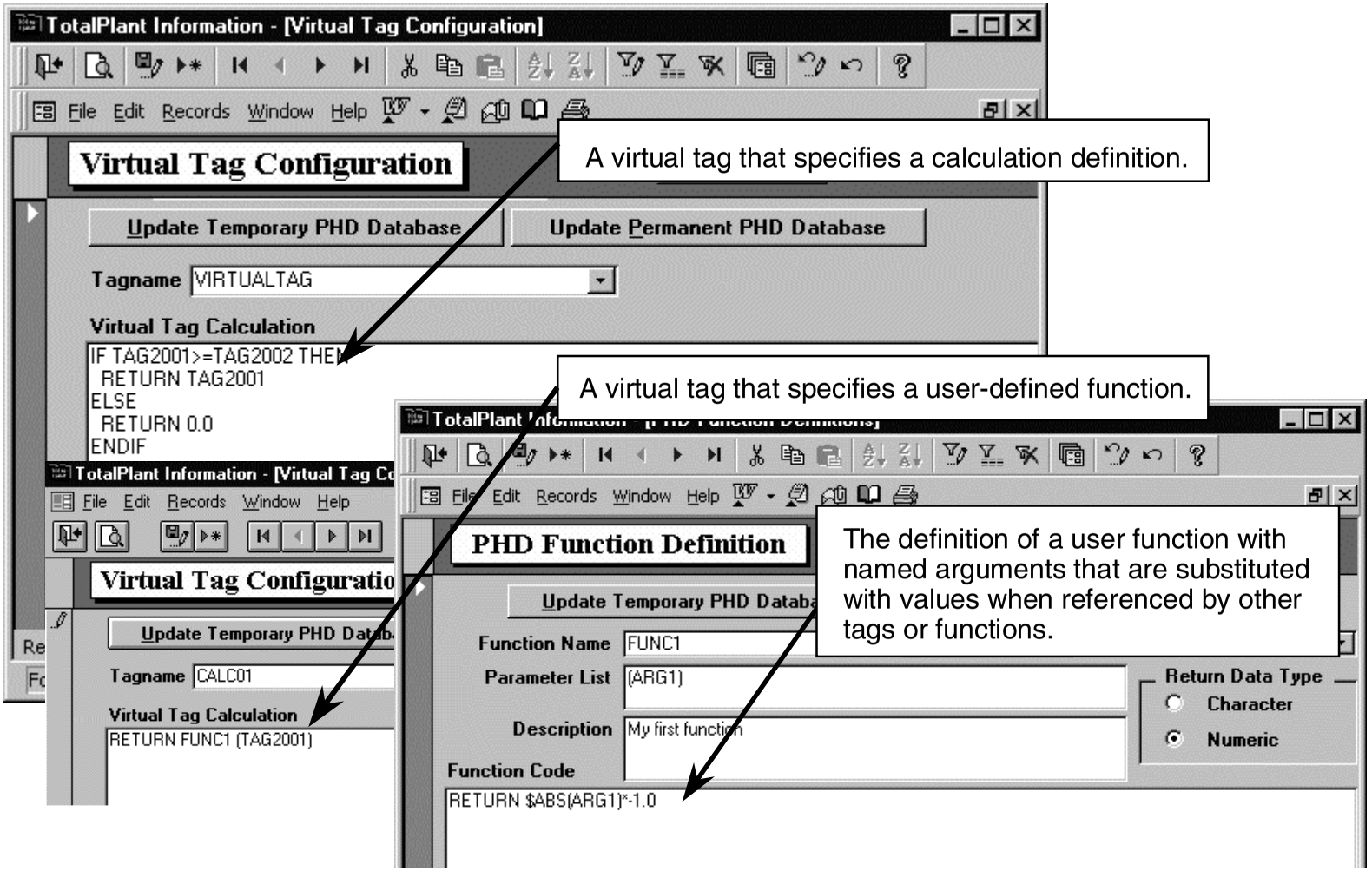
$TIME() - Returns integer time equal to now

$TIME("now-:30")

$TIME ("21-MAR-97 08:00")

Year 2000:
PHD uses an integer value for time - seconds from 01Jan70.
If the seconds converts to an earlier date, PHD assumes it is in the future.

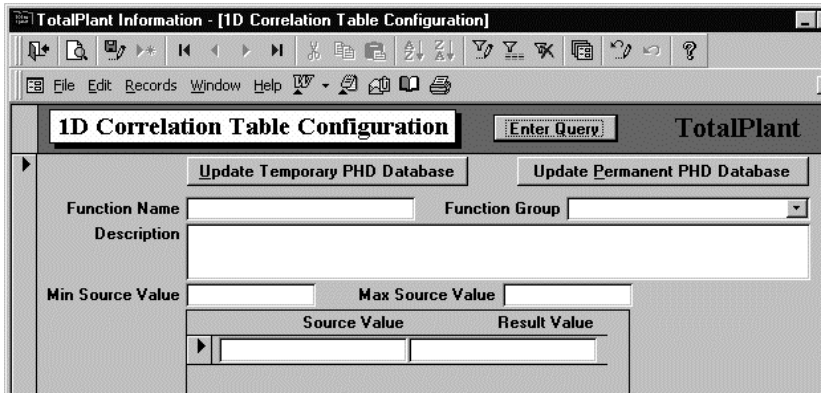
Forms for Virtual Tags and PHD Functions



1D Correlation Table

Defines functions based on 1 dimensional tables. The tables are treated as curves created by a set of linear segments with a pair of XY coordinates defining the start and end of each segment. In this case, the X coordinates are the Source Values and the Y coordinates are the Result Values.

An example of this function would be a tank strapping table, where the Source Value would be the current height of product in a tank, and the Result Value would be the volume of product at that height.



The functions are to be called as follows:

```
FunctionName( Constant or Tagname (this is the Source Value) )
```

Example: A function used in a virtual tag that returns the current volume of product in a tank:

```
Virtual Tag Tank1Volume
   Return Tank1ProductVolume ( Tank1Height_Tag)

Function Tank1ProductVolume ( Height )
   Determine volume (Result Value) for current height (Source Value)
   Return Volume
```

Steps to Define Virtual Tags

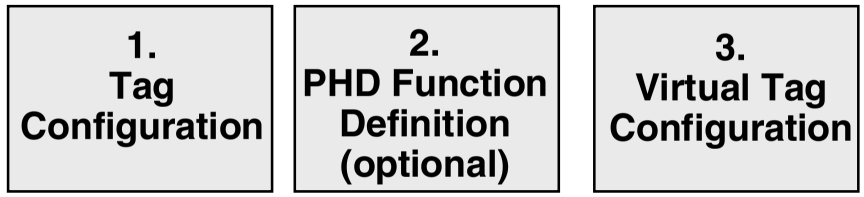
These steps are required to define a virtual tag:

1. Create a tag definition for the virtual tag, specifying Demand Calc as enabled.

2. Create a PHD function, if needed.

3. Create the virtual tag calculation.

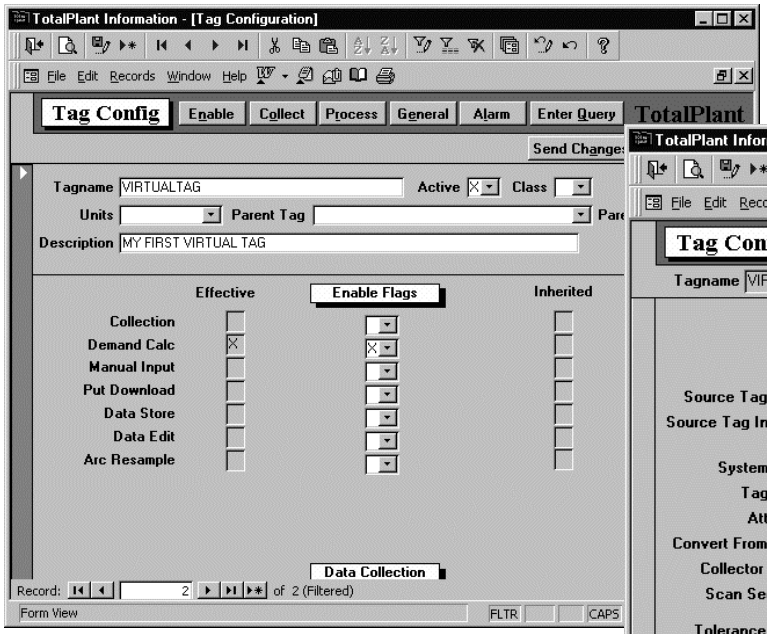
   Use the Update Temporary command to test the module definition.
   Use the Update Permanent command to compile and load the module definition into the PHD system.

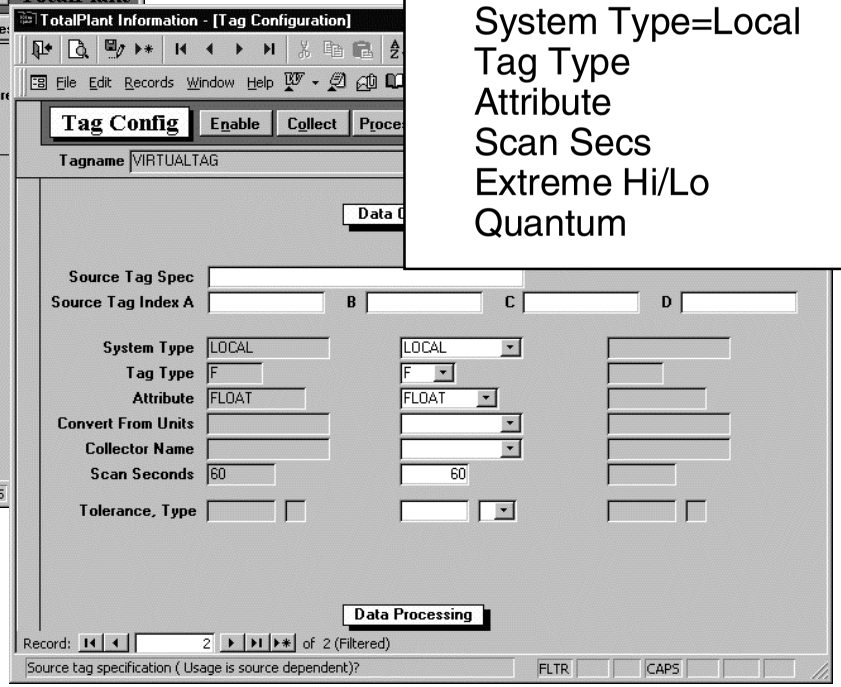


1. Tag Configuration

Create a tag definition for the virtual tag.

- Enable Demand Calc
- Scan seconds determines recalculation interval for data requests made over a time range.





Minimum Entries:
- Demand Calc=Enabled
- System Type=Local
- Tag Type
- Attribute
- Scan Secs
- Extreme Hi/Lo
- Quantum

- Tag data type must match calculation result data type.

2. User-Defined Function (optional)

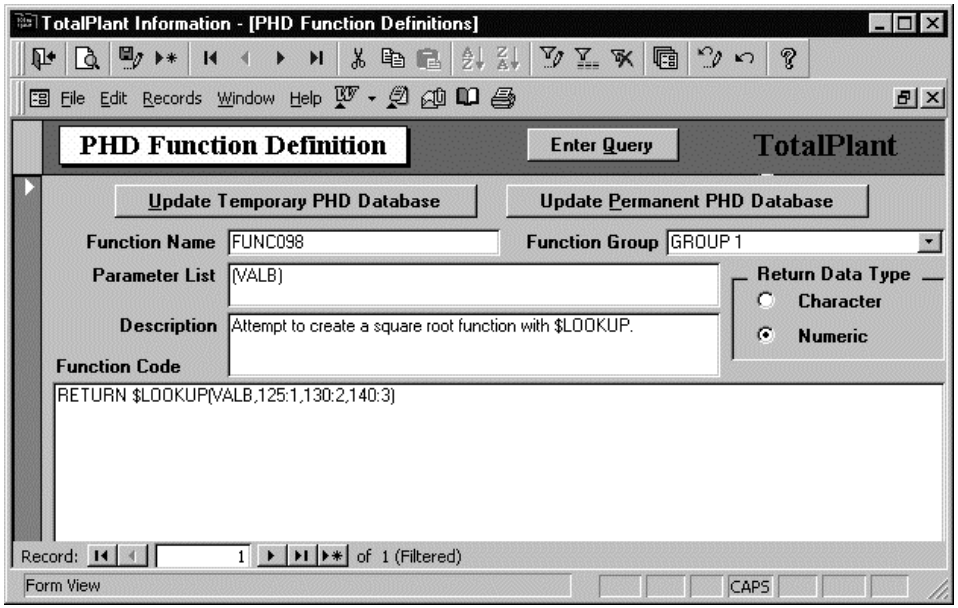
If necessary, create a PHD Function Definition or Correlation Table, then Update Temporary PHD Database.

Function Name
Required - Uniquely identifies function.

Parameter List
Optional - List of input parameters required by function.

Description
Required - Describes function for reference by other users.

Function Code
Required - Text to be executed by PHD.



Function Group
Optional - If assigned, configuration access to function can be protected through PHD Security form.

Return Data Type
Required - Defines type of data that function generates.

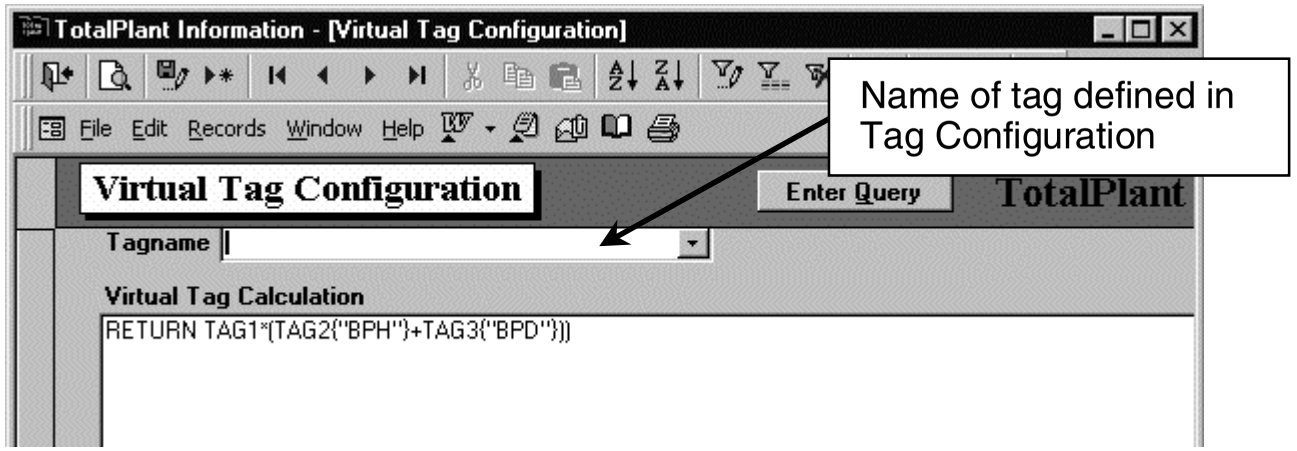
3. Virtual Tag Configuration

1. Define the Virtual Tag calculation.
   - Can include calls to other Virtual Tags or use user-defined Functions.

   Update Temporary PHD Database to test the module definition.
   - compiles tag definition
   - verifies syntax
   - loads module definition

2. Checkout Virtual Tag using the Excel VBA Example, PHDREAD, or PHDTEST.

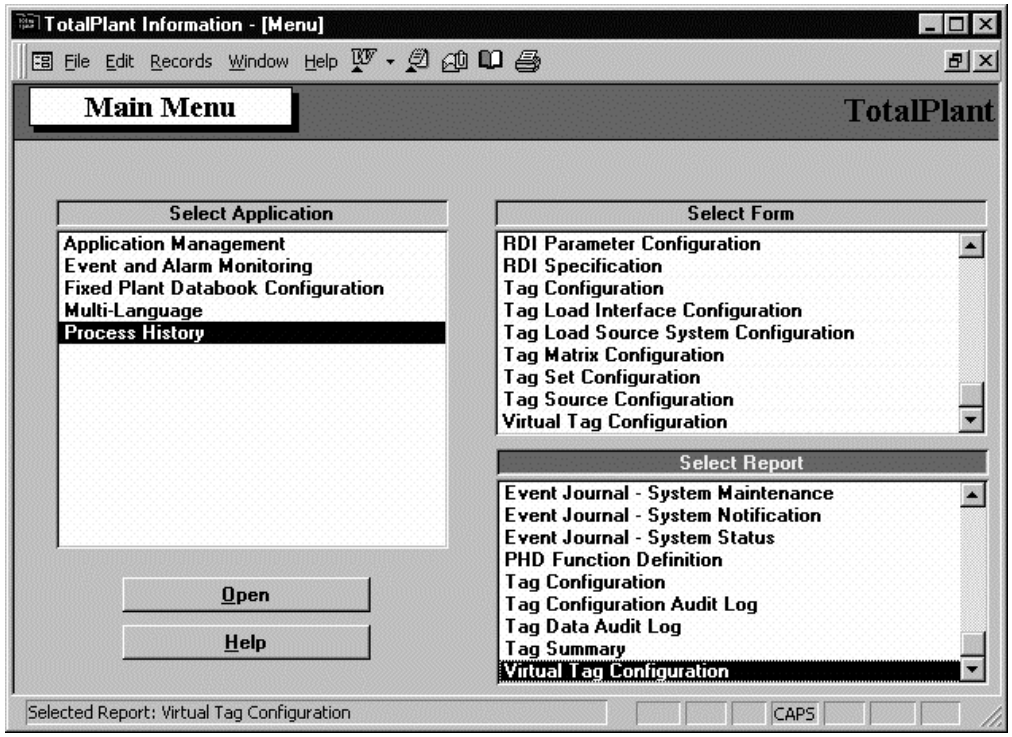
3. Update Permanent PHD Database.



Virtual Tag Configuration Report

Use the Virtual Tag Configuration Report to list the Virtual Tags currently loaded into PHD.

The report includes tagnames, descriptions, and calculation definitions.



Storing Virtual Tag Calculation Results

Virtual tag calculation results are not normally stored in PHD. They are calculated on demand in real-time as users request the information; however, certain types of calculations requested over long periods of time can place heavy demands on server hardware resources. A virtual tag RDI resolves the need for this type of demand.

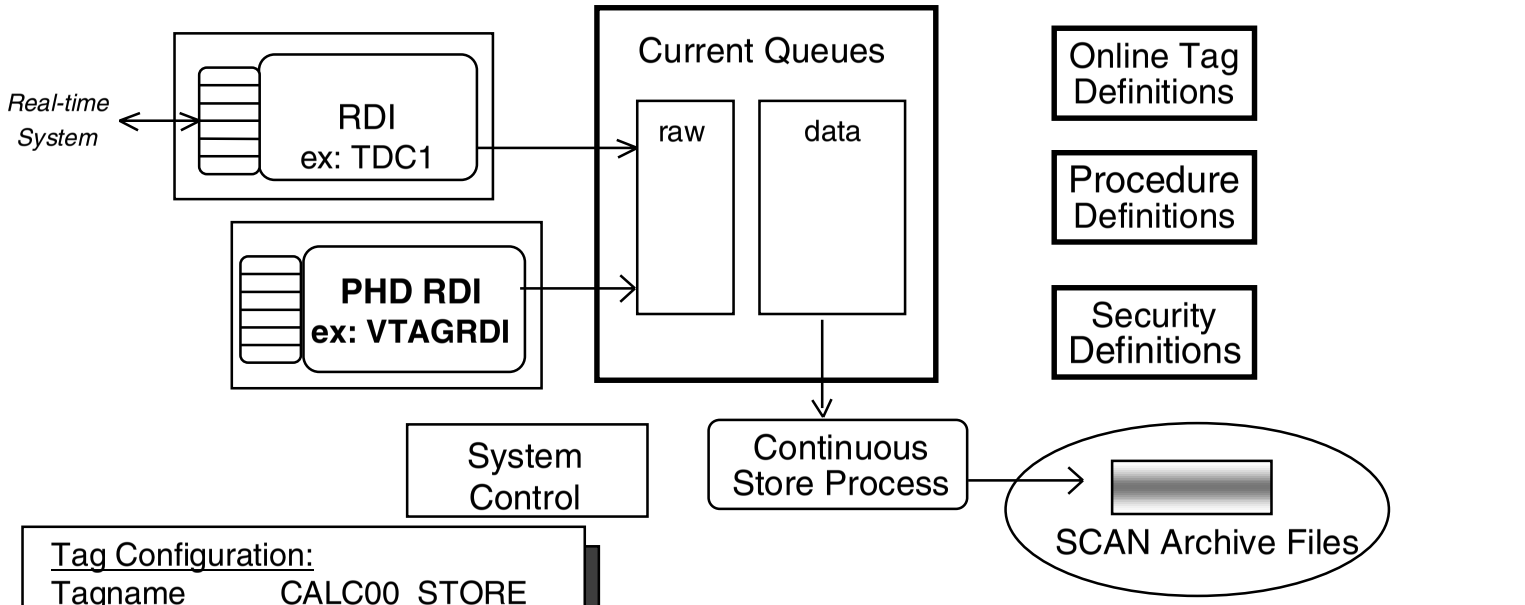
1. Create a PHD Virtual Tag RDI (Complete the RDI Specification form, then run the RDI_SERVICES Oracle client program.) The virtual tag RDI resides on a PHD server.

2. Configure a local tag that references the Virtual Tag as its source tag, and references the virtual tag RDI as its collector. Enable Collection and Data Store for the local tag.

3. The RDI periodically performs the calculation and stores the results.
   The stored results of the calculation are now available for normal use.

Refer to the *PHD User Guide*, Stored Virtual Tags and Remote PHD Interfaces

Storing Virtual Tag Calculation Results, *continued*



Tag Configuration:

Tagname	CALC00_STORE
Collection	X
Data Store	X
System Type	LOCAL
Tag Type	F (does not have to
Attribute	FLOAT apply to calc.)
Src_tag	CALC00
Collector	VTAGRDI
ScanSeconds	60
Extreme Hi/Lo	±1000000
Quantum	-1

Tag Configuration:

Tagname	CALC00
Demand Calc	X
System Type	LOCAL
Tag Type	
Attribute	
ScanSeconds	60
Extreme Hi/Lo	
Quantum	

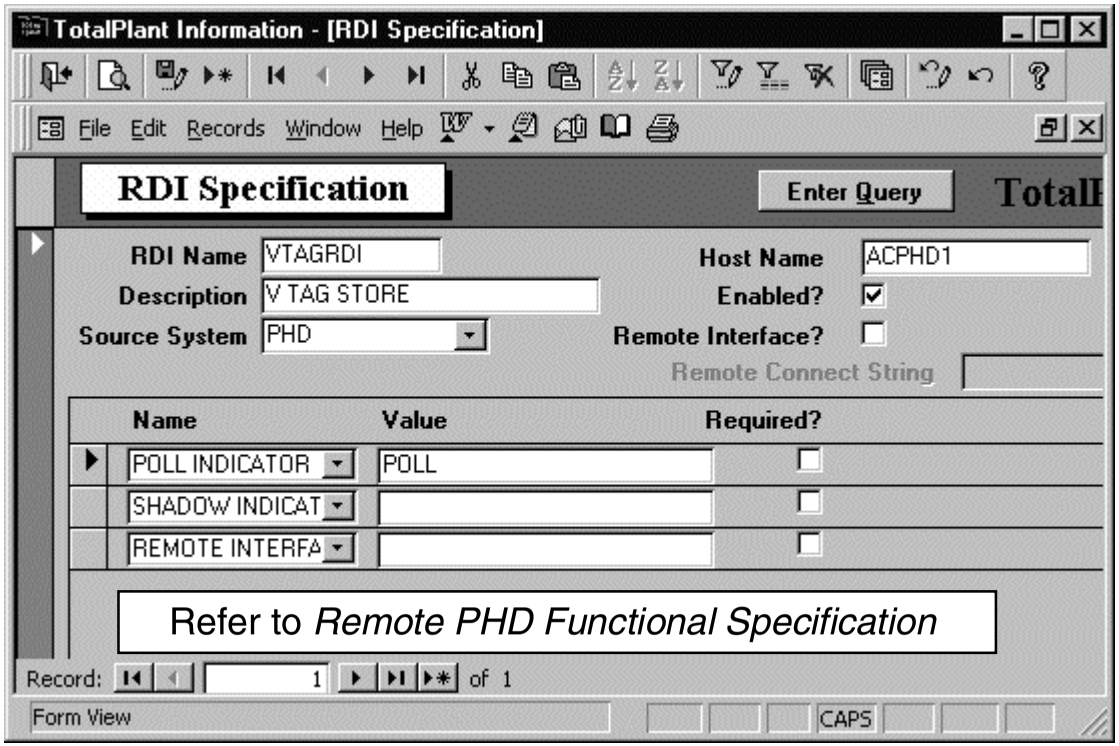
Virtual Tag Configuration:

Tagname CALC00

Calculation
Return tag1+tag2

Storing Virtual Tag Calculation Results, *continued*

1.



2. Run RDI_SERVICES (see *PHD System Manual*).

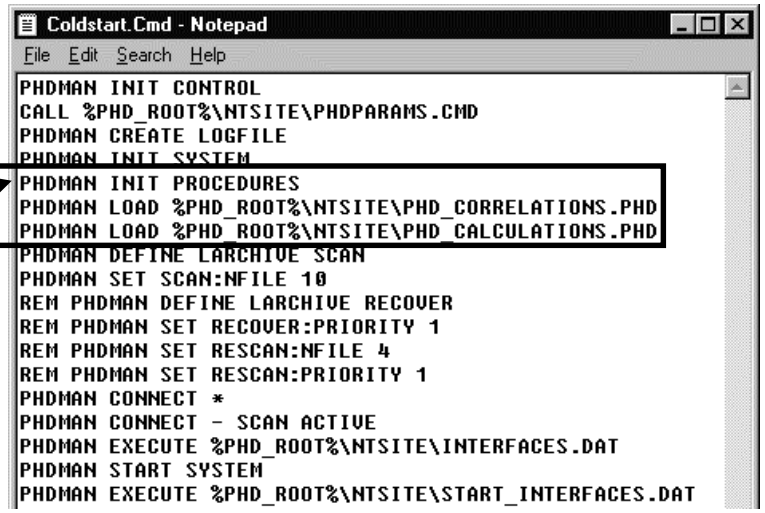
PRC Shared Memory Section

The INIT commands in the Coldstart procedure build the shared memory.

All virtual tag procedure definitions are held in the PRC (procedure) section of shared memory.

The PRC section is created and initialized by the PHDMAN INIT PROCEDURES command.

Following initialization, PHDMAN LOAD commands re-load any previously loaded procedure definitions (ntsite/phd_calculations.phd).



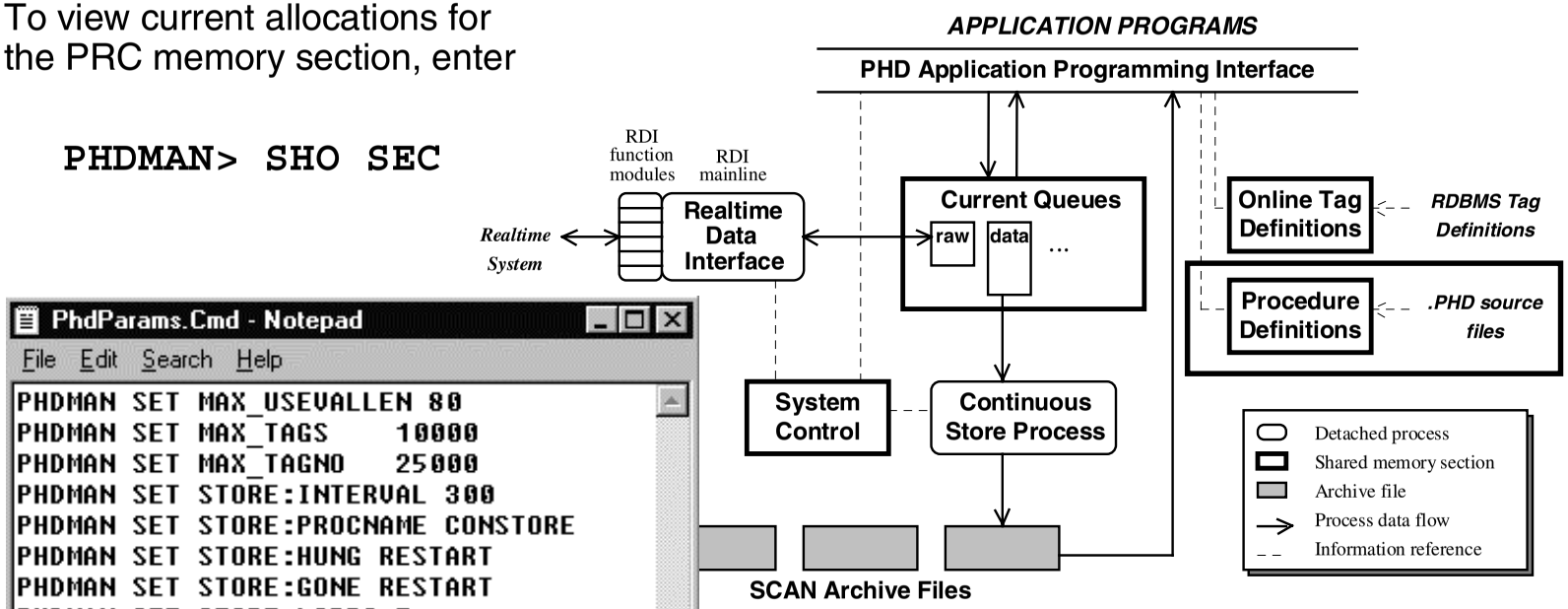
If you add to your calculations (such as more functions), PHD may tell you that it does not have enough memory:

- If the server does not have enough memory to load all calculations at load time, it will load what it can and give you an error in the event log.
- If you edit a tag while the server is running and run out of memory, you will get an error when you send it to PHD. You can increase PCT_PRCGROW, which is space to allow for growth of each calculation tag (ex. 10%).

PRC Shared Memory Section, continued

To view current allocations for the PRC memory section, enter

```
PHDMAN> SHO SEC
```



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 TOTALPLANT
PHDMAN SET DBMPASSWORD TOTALPLANT
```

See *PHD System Manual,* System Wide Parameters for the Named Parameters used to size the PRC section on initialization:

- MAX_PRCFUNC
- MAX_PRCMODULE
- MAX_PRCPOOLKB
- PCT_PRCGROW (ex. 10%)

Hands-on Exercise

Exercise 1

In this exercise, you will create a Virtual Tag that calculates a rolling hourly average, then use ModTag to check the calculation.

1. Create a class tag (named Gnn.VPARENT) that is suitable for virtual tags (to define DemandCalc, Local, range, and quantum).

2. Create a virtual tag (named Gnn.1VTAG) using Gnn.VPARENT as its parent.

3. Examine the PHD online definition of the virtual tag using PHDMAN> SHO TAG Gnn.1VTAG

4. Display and write down the PRC section statistics using PHDMAN> SHO SEC.

---

5. Create a calculation for your new tag:
   From the TPI Main Menu, select the Virtual Tag Configuration form.
   From the Tagname pulldown list, select your new virtual tag Gnn.1VTAG.
   Type-in a calculation for the rolling hourly average of your temperature tag
   `Gnn.TI.PV{,,-3600,0,"AVG"}`

6. Load the calculation definition into memory using the 'Update Temporary PHD Database' button.

7. When your calculation is error free, load the calculation into the .PHD file using the Update Permanent Database button.

Hands-on Exercise, continued

Exercise 1, *continued*

8. Use ModTag to read the value of the virtual tag and verify that it is functioning correctly.

9. Use PHDMAN to display the altered PRC section statistics (SHO SEC).

10. Use the Windows Explorer to view the contents of the .PHD file in the NTSITE directory of the PHD server.

Exercise 2

In this exercise, you will create a Virtual Tag calculation to use conditional logic.

1. Create another Virtual Tag (named Gnn.2VTAG) to return the square root of a process tag if its value is greater than 50; otherwise, return the value zero.

2. Load the new calculation definition into the PHD system.

3. Use ModTag to verify the calculation.

Hands-on Exercise, continued

Exercise 3
In this exercise, you will modify your calculation to use conditional logic.

1. Create a PHD Function Definition (named GnnFUNC) that passes in a value and returns a resultant value based on the conditional logic of exercise 2.

2. Modify the definition of the virtual tag in exercise 2 to use the new function.

3. Use the ModTag to verify the function.

4. Use the PHDREAD utility to verify the virtual tag.

Exercise 4
In this exercise, you will use another tag in a virtual tag calculation.

1. Create a virtual tag (named Gnn.3VTAG) to be the difference between the current value of a real tag, and its value 8 minutes ago.

2. Load the new calculation.

3. Use ModTag to verify the calculation performance.

Hands-on Exercise, continued

Exercise 5

In this exercise, you will store a virtual tag calculation. Your instructor will configure a virtual tag RDI and then will run RDI_SERVICES to create the RDI.

1. View the NTSITE/INTERFACES.DAT file on the PHDSERVER to verify that the virtual tag RDI exists and to verify its name.

2. Configure a local tag named Gnn.3VSTORE to store your virtual tag calculation Gnn.3VTAG.

Collection	X
Data Store	X
Source Tag	Gnn.3VTAG
System Type	LOCAL
Tag Type	F
Attribute	FLOAT
Collector Name	VTAG (we are assuming this RDI has been created)
ScanSeconds	30
Hi_Extreme	1000000
Lo_Extreme	-1000000
Quantum	-1

2. Verify that Gnn.3VSTORE is collecting and storing the calculation.

Hands-on Exercise, continued

Exercise 6
In this exercise, you will print a report of your virtual tags.

1. From the PHD main menu, select the Virtual Tag Configuration report.

2. Print the report for your virtual tags (Gnn*) on the classroom printer.

END OF EXERCISES

Honeywell

Helping You Manage Your World

www.iac.honeywell.com