

**uniformance**

# Introduction to VisualPHD

# Lesson Objective

---

## Objective

- Recognize the basic steps required to access PHD data using the PHD ActiveX (OLE Automation) Server and the PHD ActiveX(OCX) Controls.

## Topics

- What is VisualPHD?
- VB and VBA Programming steps to use the PHD ActiveX Server.
- Honeywell Excel/VBA Workbook Example (included with Uniformance Desktop)
- PHD ActiveX Controls
  - Data Control
  - Graph Control
  - Text Control
  - Bar Control
  - Tag Tree Control and Tag Text Control
- Honeywell VB Examples (included with Uniformance Desktop)

## References:

Online documents:

*VisualPHD Help*

*VisualPHD Excel Examples Help*

CD documents:

*VisualPHD User Guide, PIM2201*

*VisualPHD Developer's Reference Guide, PIM3001*

# VisualPHD

VisualPHD consists of

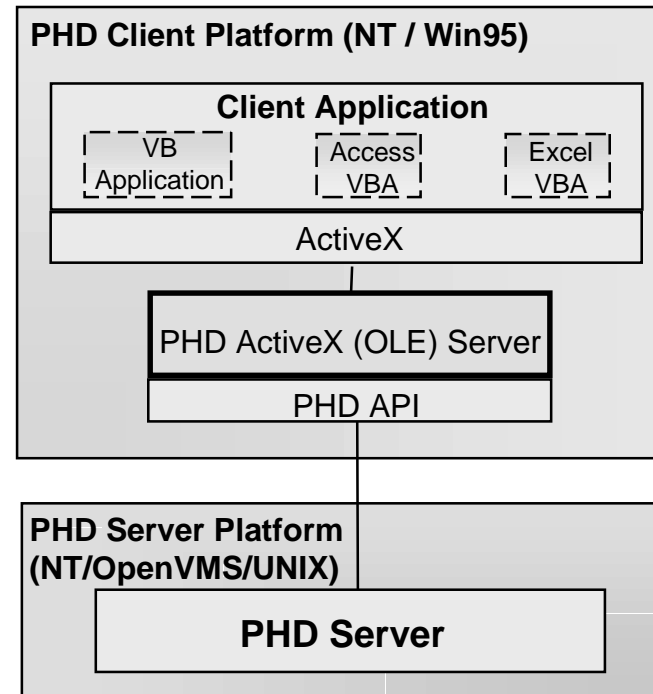
- a PHD ActiveX (OLE Automation) Server to retrieve data from PHD
- a collection of ActiveX(OCX) controls to visually access data from PHD
- Honeywell prebuilt VB and Excel VBA Examples

The basic concepts of accessing data through the ActiveX server are the same for VB and VBA.

The PHD ActiveX Server is an object that handles all requests for PHD data. The PHD ActiveX Server talks to applications and knows how to talk to PHD.

Two DLLs act as the PHD ActiveX Server (part of the Desktop):  
PHDBASE  
PHDATA

OLE - Object Linking and Embedding  
VBA - Visual Basic for Applications



# Using PHD ActiveX Server

---

The programming steps required to use the ActiveX (OLE Automation) Server for the core part of data access:

1. Initialize OLE Object
2. Setup OLE Automation Server
3. Add Tags to OLE Object
4. Setup Tag Attributes (properties)
5. Setup For Data Fetch
6. Fetch Data
7. Unpack Data

There are other ways, but for simplicity, we show these steps.

You can combine the blocks; however, there is some required order.

The programmer decides what to do here, depending on the application being used and what he/she plans to do with the data.

(Refer to *VisualPHD User Guide*, OLE Automation Server)

# 1. Initialize OLE Object

---


## ■ Setup Global variables

- Global PHDObject as Object
- Global values() as variant
- Global timestamps() as variant
- Global confidences() as variant

## ■ Create OLE Object

Set PHDObject = CreateObject (“VisualPHD.Data”)

Registered name of this  
OLE object (registered  
when installed)



## **2. Setup OLE Automation Server Properties**

---

### ■ Setup basic attributes of OLE Object

- PHDObject.Hostname = <string of node name>
- PHDObject.Login sUsername, sPassword
- PHDObject.SampleFrequency = ##
- PHDObject.SampleMethod = ssssssss

(where ssssssss = raw or interpolate raw)

If setting up a reduction request, then use interpolate raw as the sample method.

### 3. Add Tags to OLE Object

---

- PHDObject.Tags.Add <name1>,[name2],[hostname]
  - name1 = actual name of tag on PHD server
  - name2 = application alias name for tag (defaults to name1 if left out)
  - hostname = Each tag can have a different hostname; if not provided, then object hostname is used.

## 4. Setup Tag Attributes

---

### ■ If reduction request then

- PHDObject.SampleMethod = “InterpolateRaw”
- PHDObject.ReductionFrequency = nnn
- PHDObject.ReductionOffset = “before” or “around”, or “after”
- PHDObject.Tags(sTagname).ReductionType = ssssssss  
(where ssssssss= average, first, last, etc.)
- PHDObject.Tags(sTagname).NullValue = “ ”
- PHDObject.UseSampleFrequency = False
- PHDObject.MinimumConfidence = <0 - 100>



## 4. Setup Tag Attributes, *continued*

---

- If Raw or Snapshot request then
  - PHDObject.SampleMethod = “Raw” or “Snapshot”
  - PHDObject.Tags(sTagname).ReductionType = “ ”
  - PHDObject.Tags(sTagname).NullValue = “ ”
  - – PHDObject.MaximumRows = nnnn
  - PHDObject.MinimumConfidence = <0 - 100>

**Allows PHD and ActiveX server to do “best fit.”** For each interval, you will get up to 5 values: the first value, the maximum value, the minimum value within the interval, the last value, and the first bad value if it exists. If any of the values are the same value (such as, the first value is the maximum value), then only the unique values will be returned. This is a useful option to prevent using up the resource. For example, if a tag had changed every second and you asked for a day--you could get a day's worth of one second samples unless you specified maximum rows.

## 5. Setup for Data Fetch

---

- PHDObject.StartTime = <absolute time or relative time>  
ex: “06/22/97 12:30:00” or “now-1h”
- PHDObject.EndTime = <absolute time or relative time>  
ex: “06/22/97 14:30:00” or “now”

## 6. Fetch Data

---

### ■ PHDObject.Fetch

- This call returns all data for all tags per the retrieval attributes defined previously

The PHD ActiveX Server holds the data in memory - the application does not have it yet.

### ■ PHDObject.GetTimes dStartT, dEndT

(where dStartT and dEndT are dimensioned as a date datatype:

Dim dStartT as Date, dEndT as Date)

There are timer controls in Visual Basic to collect data at an interval. The timer control triggers an event, your code can check the timer; when timer fires, run code.

## 7. UnPack Data

---

- PHDObject.Tags(sTagname).UnpackResults  
values, timestamps, confidences
  - This call unpacks all of the data for the tag defined into a set of globally defined arrays.

## **Last Step**

---

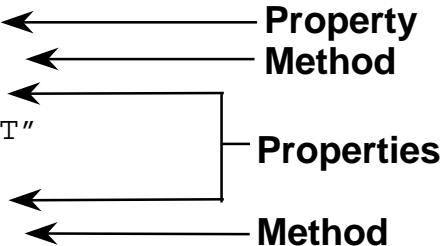
- Once data is retrieved and unpacked into user arrays, the application needs to perform whatever it needs to do on the returned data.

# Excel VBA Demonstration

---

- Instructor demonstration:

```
SUB PHDREAD()  
DIM PHDOBJECT AS OBJECT  
SET PHDOBJECT = CREATEOBJECT("VISUALPHD.DATA")  
  
PHDOBJECT.HOSTNAME = "ACPHD1" ← Property  
PHDOBJECT.TAGS.ADD "TIC21941.PV" ← Method  
PHDOBJECT.SAMPLEFREQUENCY = 60 ← Properties  
PHDOBJECT.SAMPLEMETHOD = "SNAPSHOT" ← Properties  
PHDOBJECT.STARTTIME = "NOW" ← Method  
PHDOBJECT.ENTDIME = "NOW" ← Method  
PHDOBJECT.FETCH  
  
WORKSHEETS(1).RANGE("A2").VALUE = PHDOBJECT.TAGS("TIC21941.PV").TIMESTAMP  
WORKSHEETS(1).RANGE("B2").VALUE = PHDOBJECT.TAGS("TIC21941.PV").VALUE  
WORKSHEETS(1).RANGE("C2").VALUE = PHDOBJECT.TAGS("TIC21941.PV").CONFIDENCE  
  
SET PHDOBJECT = NOTHING  
  
END SUB
```



# Excel VBA Demonstration, *continued*

---

- **Instructor demonstration:**

```
SUB PHDREAD()  
DIM PHDOBJECT AS OBJECT  
SET PHDOBJECT = CREATEOBJECT("VISUALPHD.DATA")  
  
PHDOBJECT.HOSTNAME = "ACPHD1" ← Property  
PHDOBJECT.TAGS.ADD "TIC21941.PV" ← Method  
PHDOBJECT.SAMPLEFREQUENCY = 60 ← Properties  
PHDOBJECT.SAMPLEMETHOD = "SNAPSHOT" ← Properties  
PHDOBJECT.STARTTIME = "NOW-1H" ← Properties  
PHDOBJECT.ENDTIME = "NOW" ← Method  
PHDOBJECT.FETCH ← Method  
  
FOR I = 1 TO 60  
  
WORKSHEETS(1).CELLS(I+1,2).VALUE = PHDOBJECT.TAGS("TIC21941.PV").TIMESTAMP  
WORKSHEETS(1).CELLS(I+1,3).VALUE = PHDOBJECT.TAGS("TIC21941.PV").VALUE  
WORKSHEETS(1).CELLS(I+1,4).VALUE = PHDOBJECT.TAGS("TIC21941.PV").CONFIDENCE  
  
PHDOBJECT.MOVENEXT  
  
NEXT I  
  
SET PHDOBJECT = NOTHING  
  
END SUB
```

# Excel VBA Examples

Honeywell includes Excel VBA Workbook example with the PHD product.

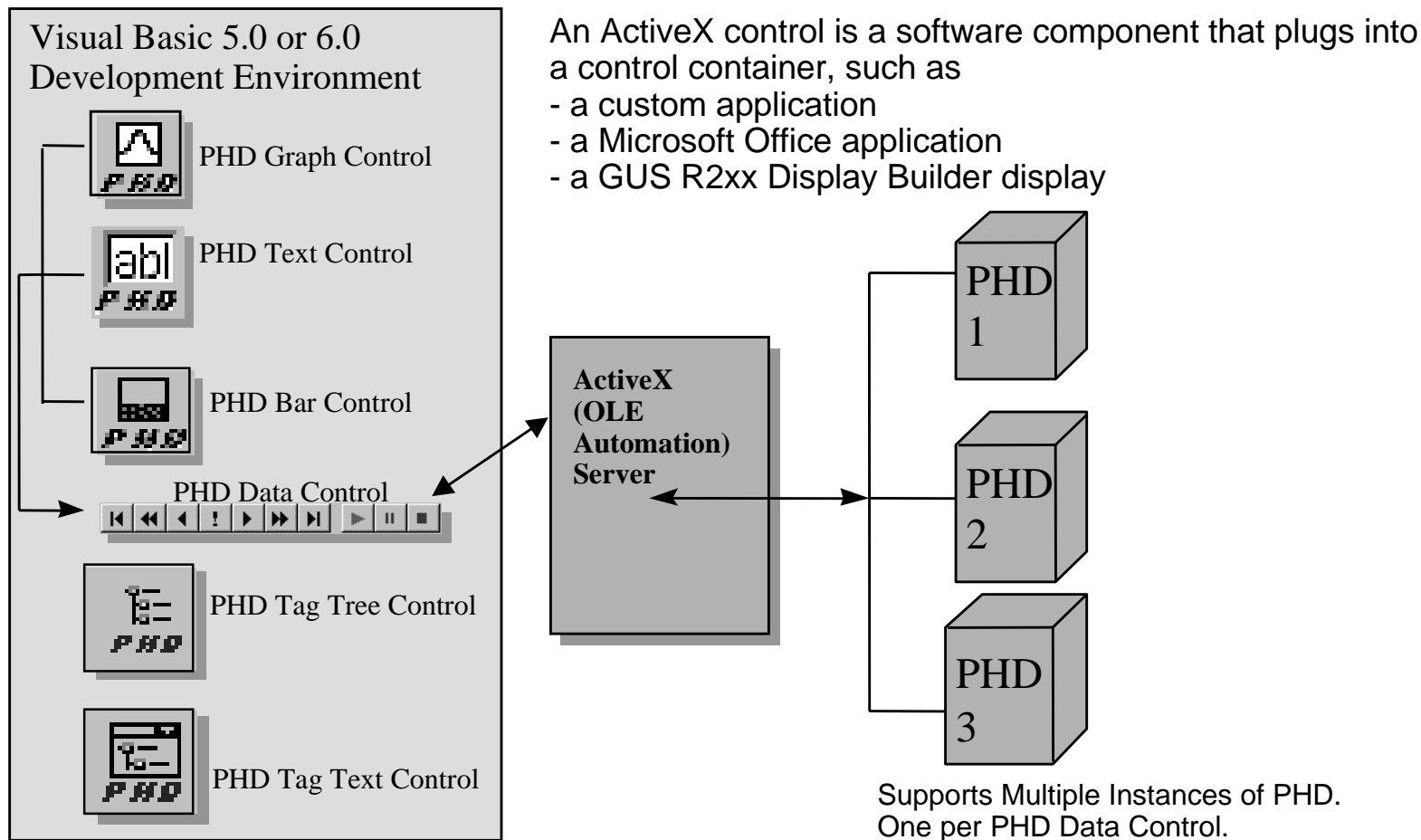
Start/Programs/TotalPlant Information/VisualPHD/Excel VBA Examples:

	A	B	C	D	E
1	<b>PHD Process History Data</b>				
2	StartTime	11/10/97 1:33	MinimumConfidence	95	
3	EndTime	11/10/97 13:33	NULL Value		Get Units
4	SampleFrequency	600	ReductionFrequency	3600	Get Data
5	SampleMethod	Average	ReductionOffset	Before	
6			FilterData	FALSE	
7	Tagname	tic21941.pv			
8	Units	DEGC			
9	Offset				
10	ReductionType				
11	10-Nov-97 01:33:48	45.47560883			
12	10-Nov-97 01:43:48	34.28372192			
13	10-Nov-97 01:53:48	30.0483017			
14	10-Nov-97 02:03:48	45.57706833			
15	10-Nov-97 02:13:48	37.0793457			
16	10-Nov-97 02:23:48	29.38186264			
17	10-Nov-97 02:33:48	45.55933762			
18	10-Nov-97 02:43:48	39.24417114			
19	10-Nov-97 02:53:48	26.48608208			
20	10-Nov-97 03:03:48	44.96413803			

Reference: *Visual PHD Excel Examples*



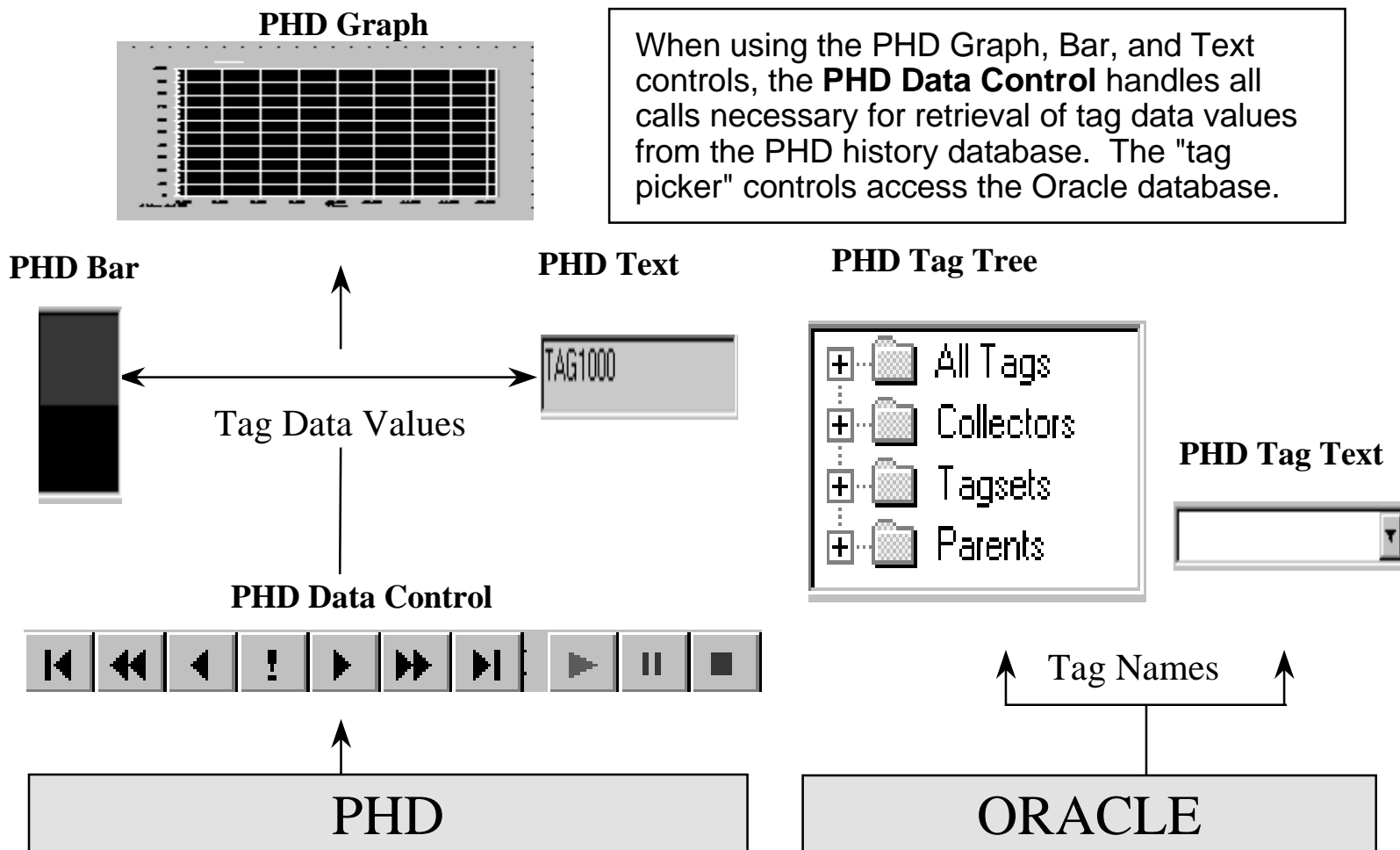
# Visual Basic ActiveX Controls for PHD



ActiveX Controls - Originally known as OLE Controls or OCXs.

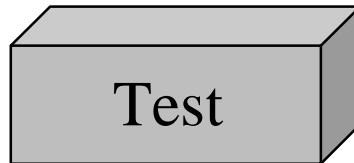
(Refer to *Visual PHD User Guide*)

# ActiveX Controls for PHD



# OLE Objects

---



**Properties:** Describe the Object

- Name: Test
- Top: 1000
- Left: 5000
- Width: 1500
- Caption: Test
- Color: Green
- ***Event Properties***

Available only at Design time  
Read Only at Run time  
Read/Write at Run time

**Methods:** Action the Object can perform

- Get: Get data
- MoveFirst: Move to first data record
- MoveLast: Move to last record
- MoveNext: Move to next record
- MovePrevious: Move to previous record
- Delete: Delete the current record
- Close: Close the object

May or may not contain a visual component.

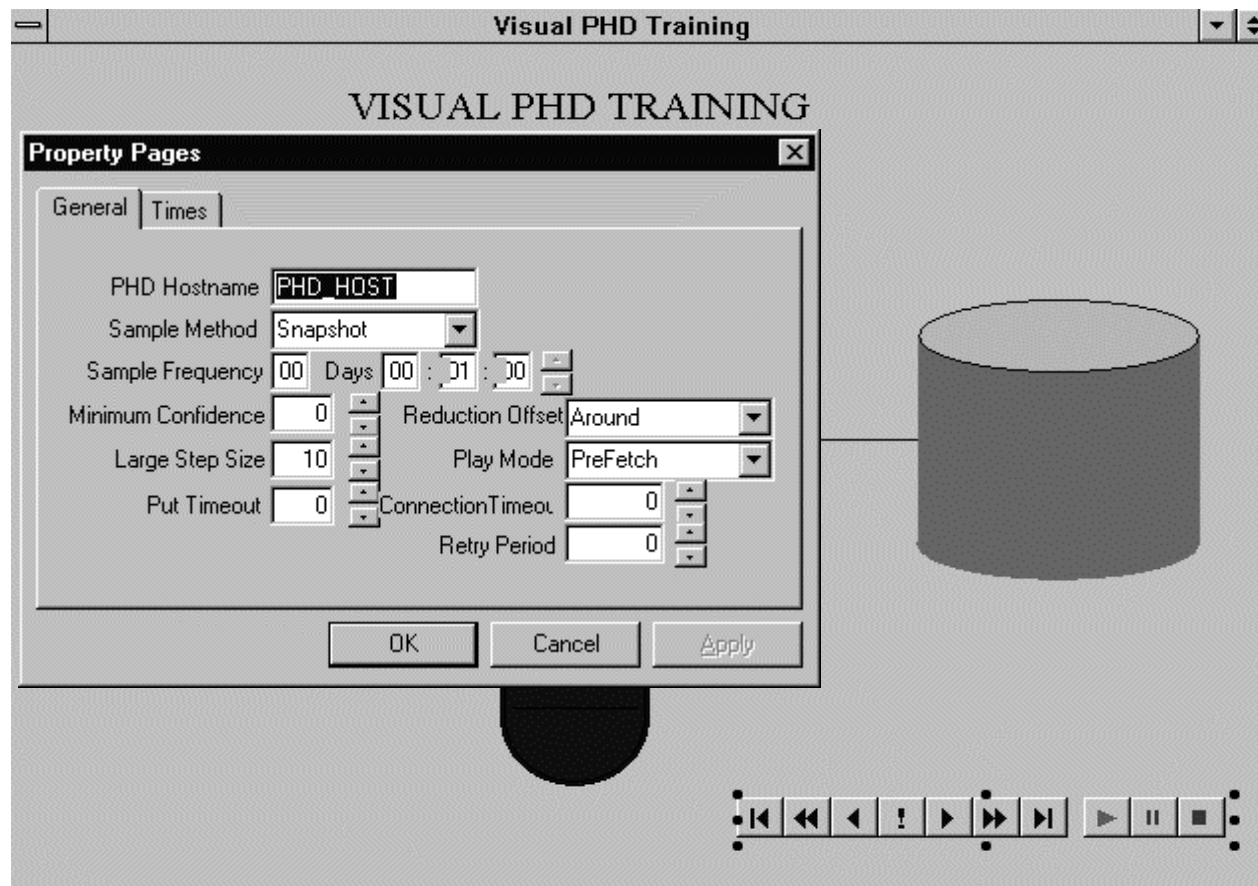
# PHD Data Control Object - Properties

The screenshot displays the 'Visual PHD Training' application. On the left, the 'Properties - PHDCtrl1' window is open, showing a list of properties for the 'PHDCtrl1' object. The 'HostName' property is highlighted and set to 'PHD\_HOST'. The main window, titled 'Visual PHD Training', shows a diagram with a vertical bar on the left and a cylinder on the right, connected by a horizontal line. Below the diagram is a control panel with various buttons for navigation and playback.

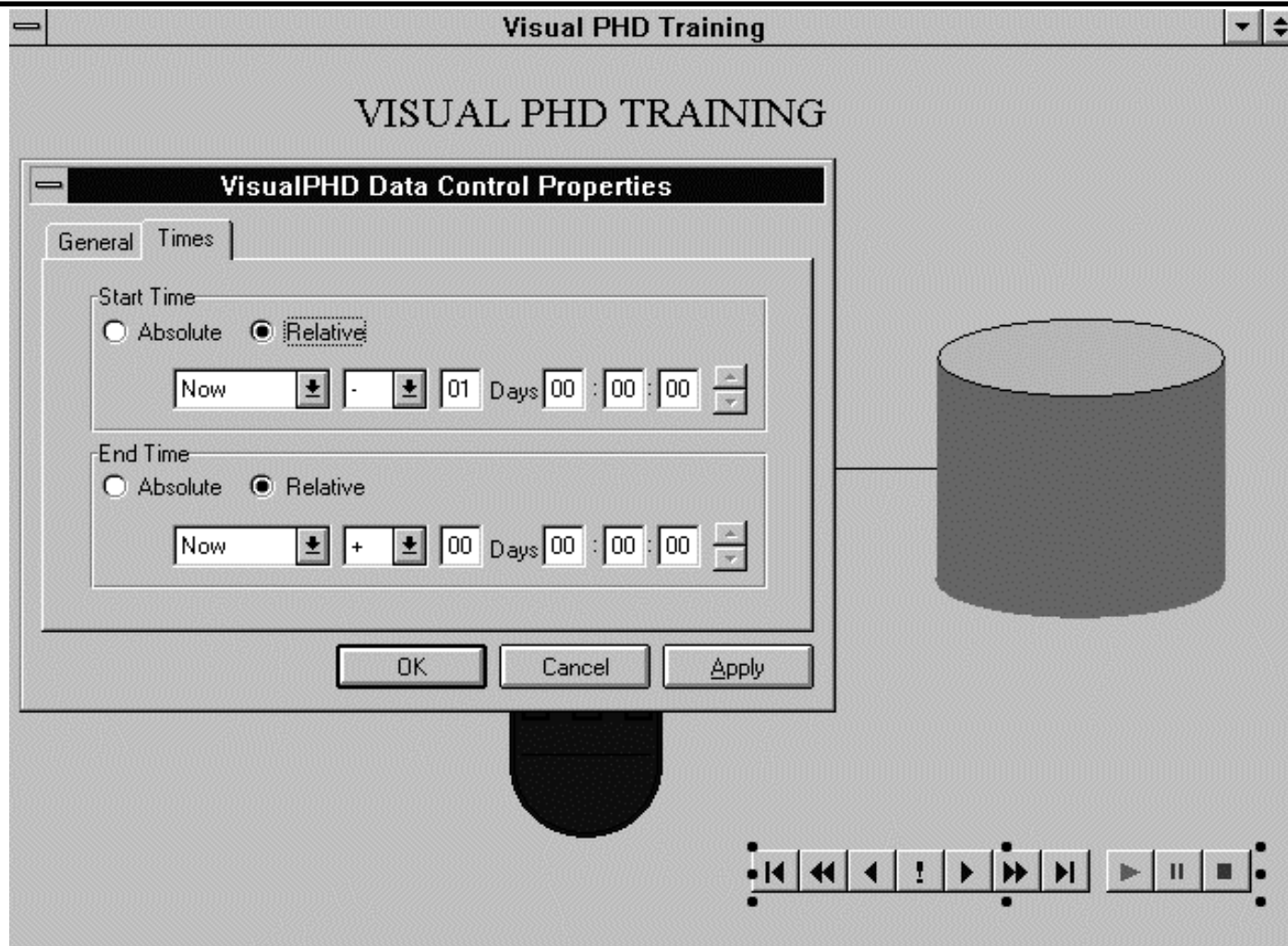
Properties - PHDCtrl1	
PHDCtrl1 PHDCtrl	
Alphabetic   Categorized	
(About)	
(Custom)	
(Name)	PHDCtrl1
BOF	False
ConnectionTimeou	0
DragIcon	(None)
DragMode	0 - vbManual
EndTime	NOW
EOF	False
FilterData	False
Height	495
HelpContextID	0
HostName	PHD_HOST
Index	
LargeStepSize	10
Left	360
MinimumConfidenc	0
PlayMode	PreFetch
PutTimeOut	0
ReductionFrequen	60
ReductionOffset	Around
RetryPeriod	0
RowCount	0

## PHD Data Control Object - Properties, *continued*

---



## PHD Data Control Object - Properties, *continued*



# PHD Bar Object - Properties

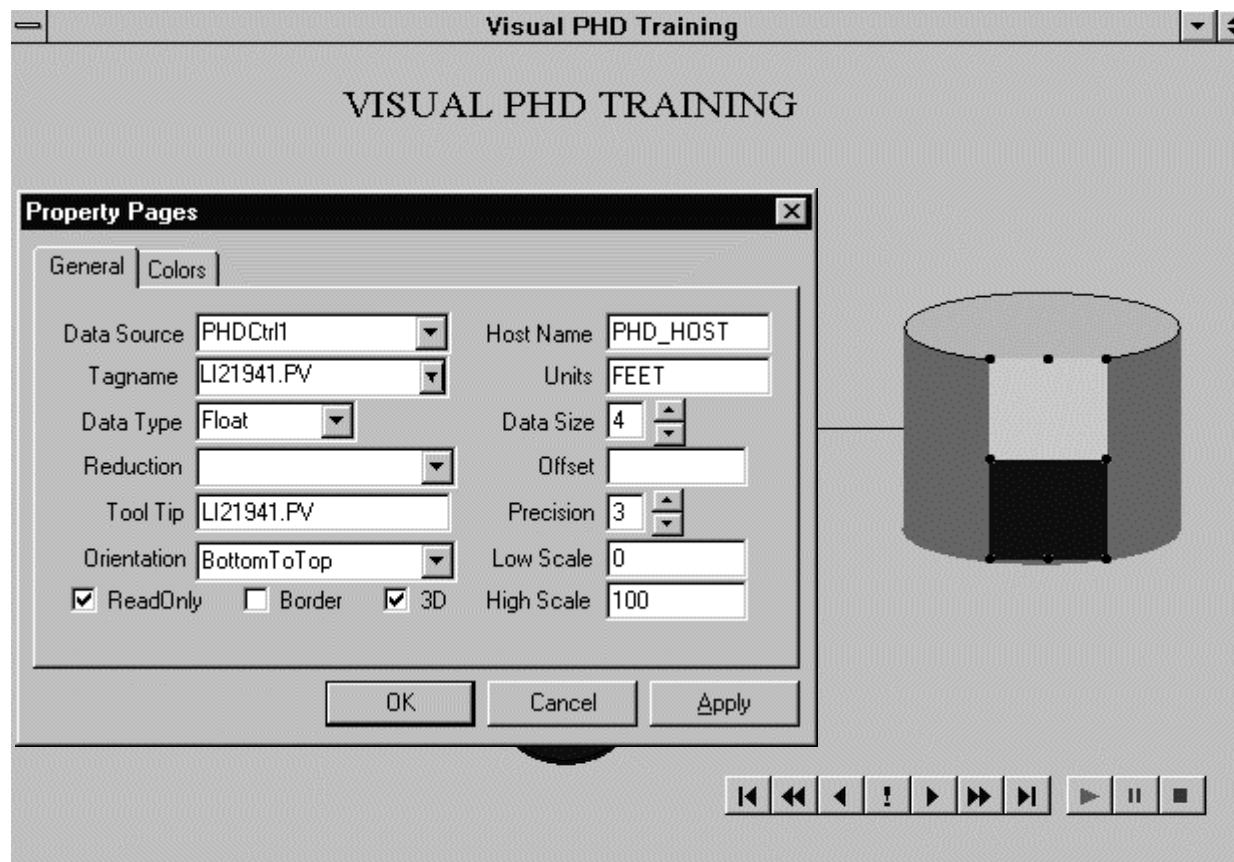
The screenshot displays the 'Visual PHD Training' application. On the left, the 'Properties - PHDBar1' window is open, showing a list of properties for the 'PHDBar1' object. The 'Categorized' tab is selected. The properties list includes:

Property	Value
(About)	
(Custom)	
(Name)	PHDBar1
Appearance	1
BackColor	&H8000000F&
BorderStyle	0 - None
Confidence	
DataSize	4
DataSource	PHDCtrl1
DataType	Float
DecimalPlaces	3
DownloadStatus	
DragIcon	(None)
DragMode	0 - vbManual
Enabled	True
ForeColor	&H80000012&
Height	1095
HelpContextID	0
HighScale	0
HostName	PHD_HOST
Index	
Left	480
LowScale	0

The main 'Visual PHD Training' window shows a 3D visualization of a PHD bar. The bar is a vertical cylinder with a black top and bottom. The main body is white with a black outline. A black rectangular area is visible on the right side of the bar. The text 'L PHD TRAINING' is displayed at the top. At the bottom right, there is a control panel with buttons for navigation and playback.

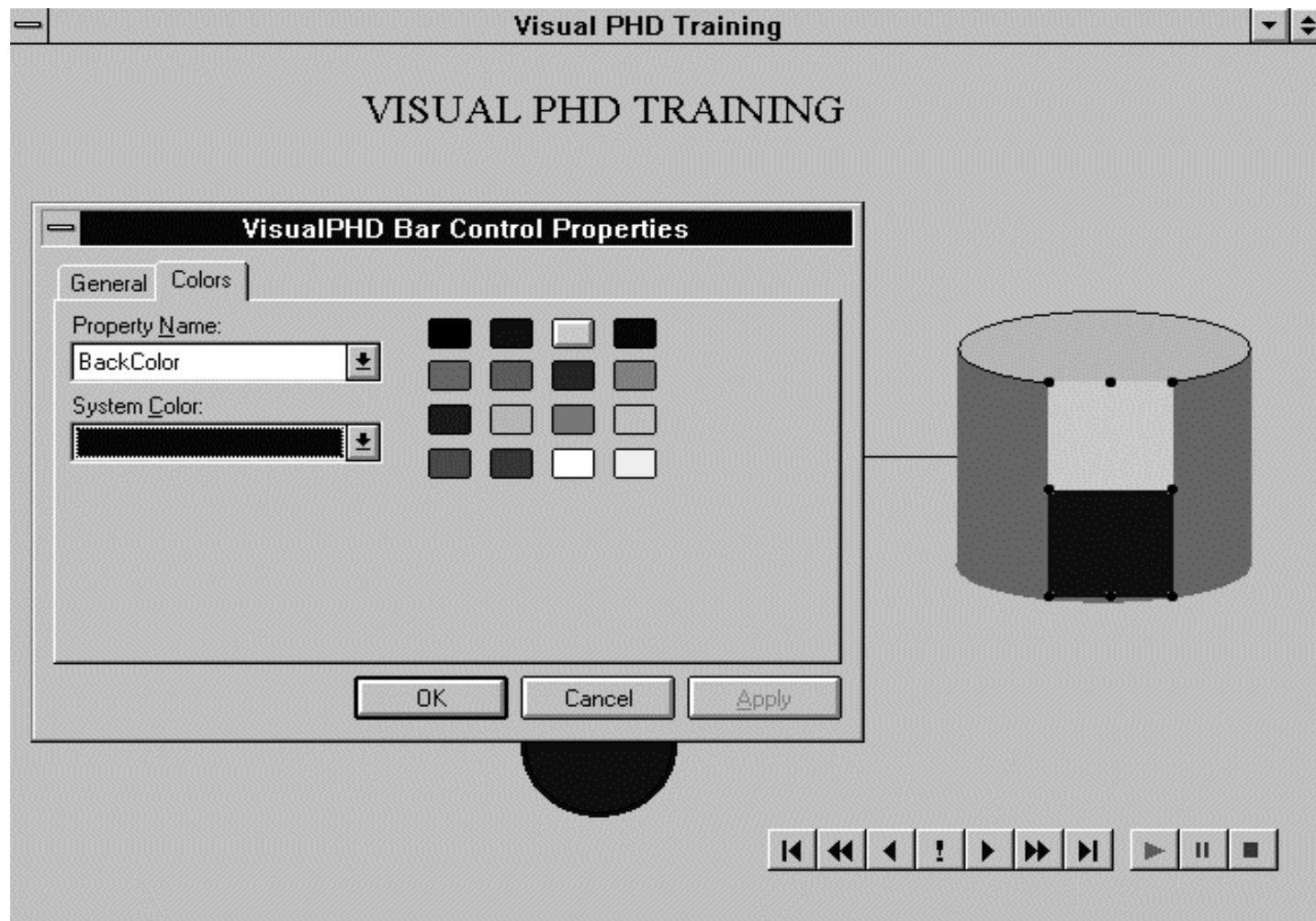
## PHD Bar Object - Properties, *continued*

---



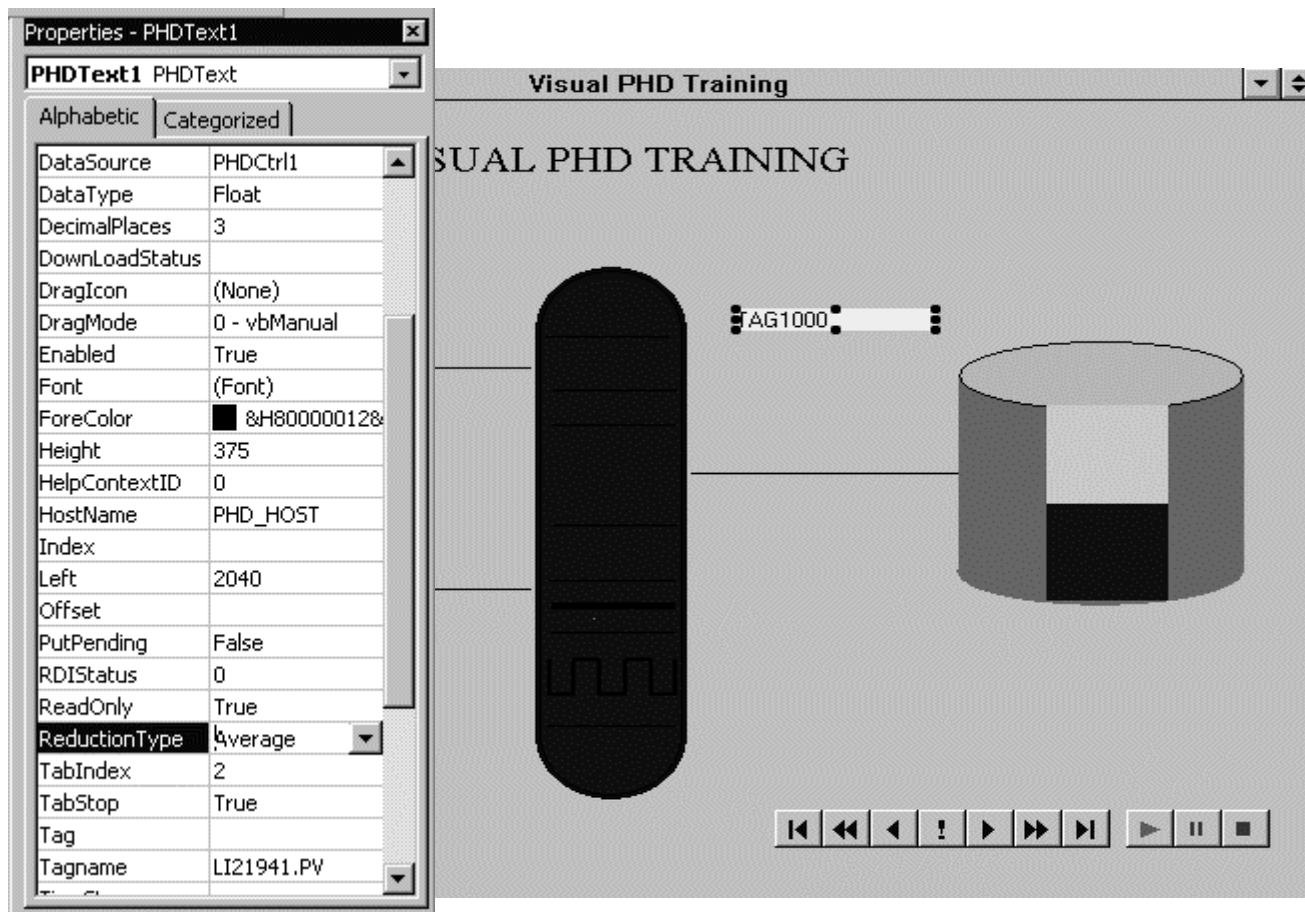


## PHD Bar Object - Properties, *continued*



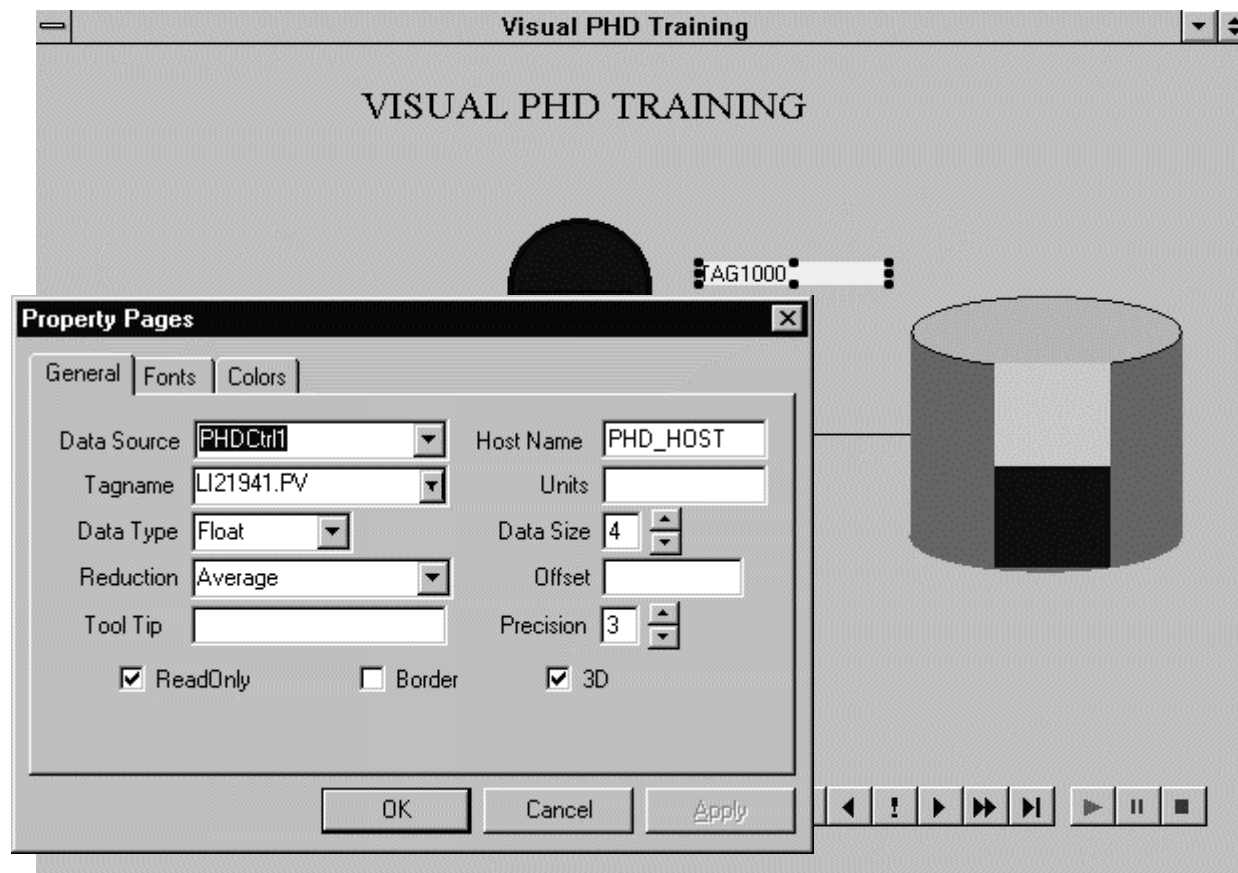
# PHD Text Object - Properties

---

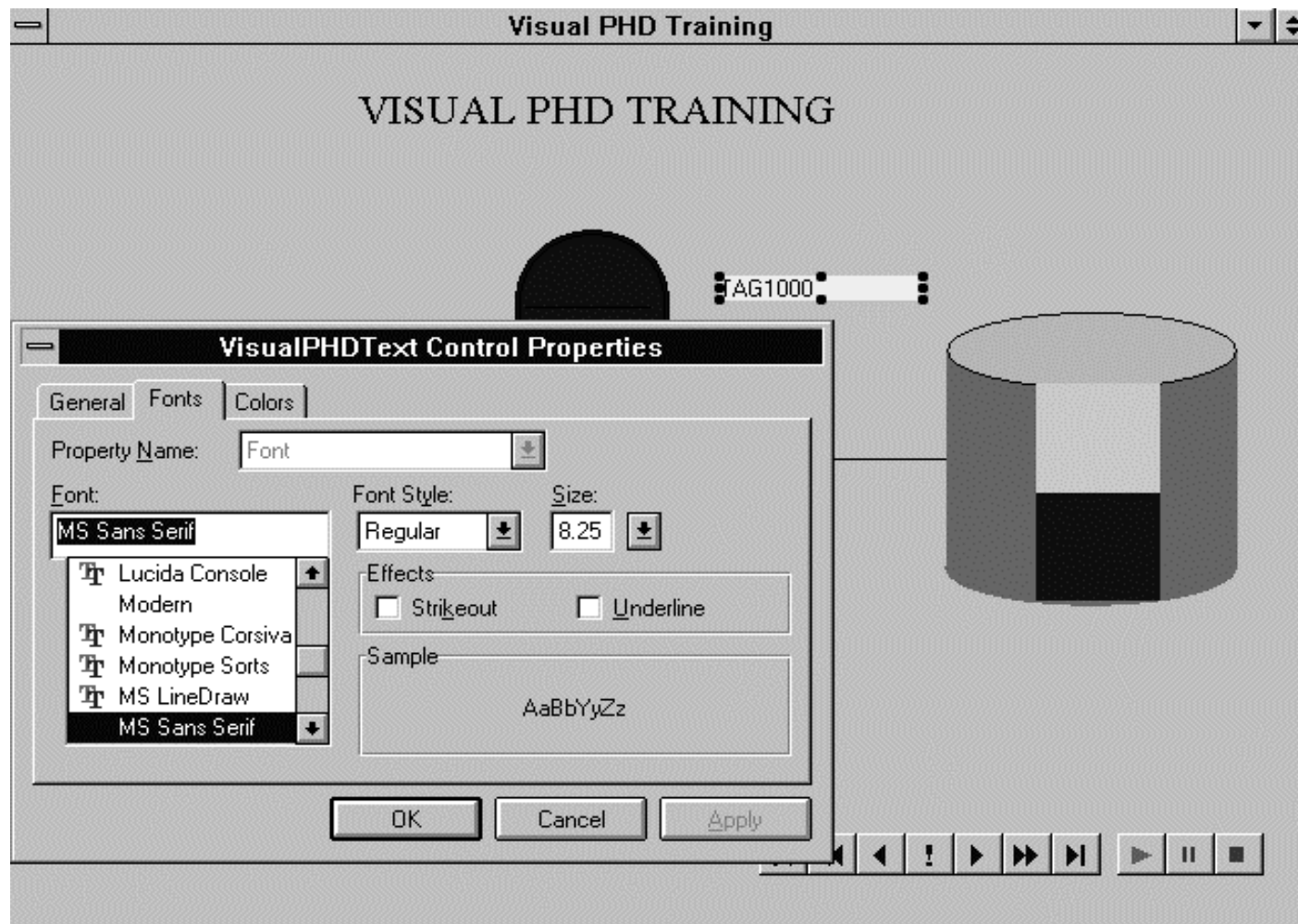


## PHD Text Object - Properties, *continued*

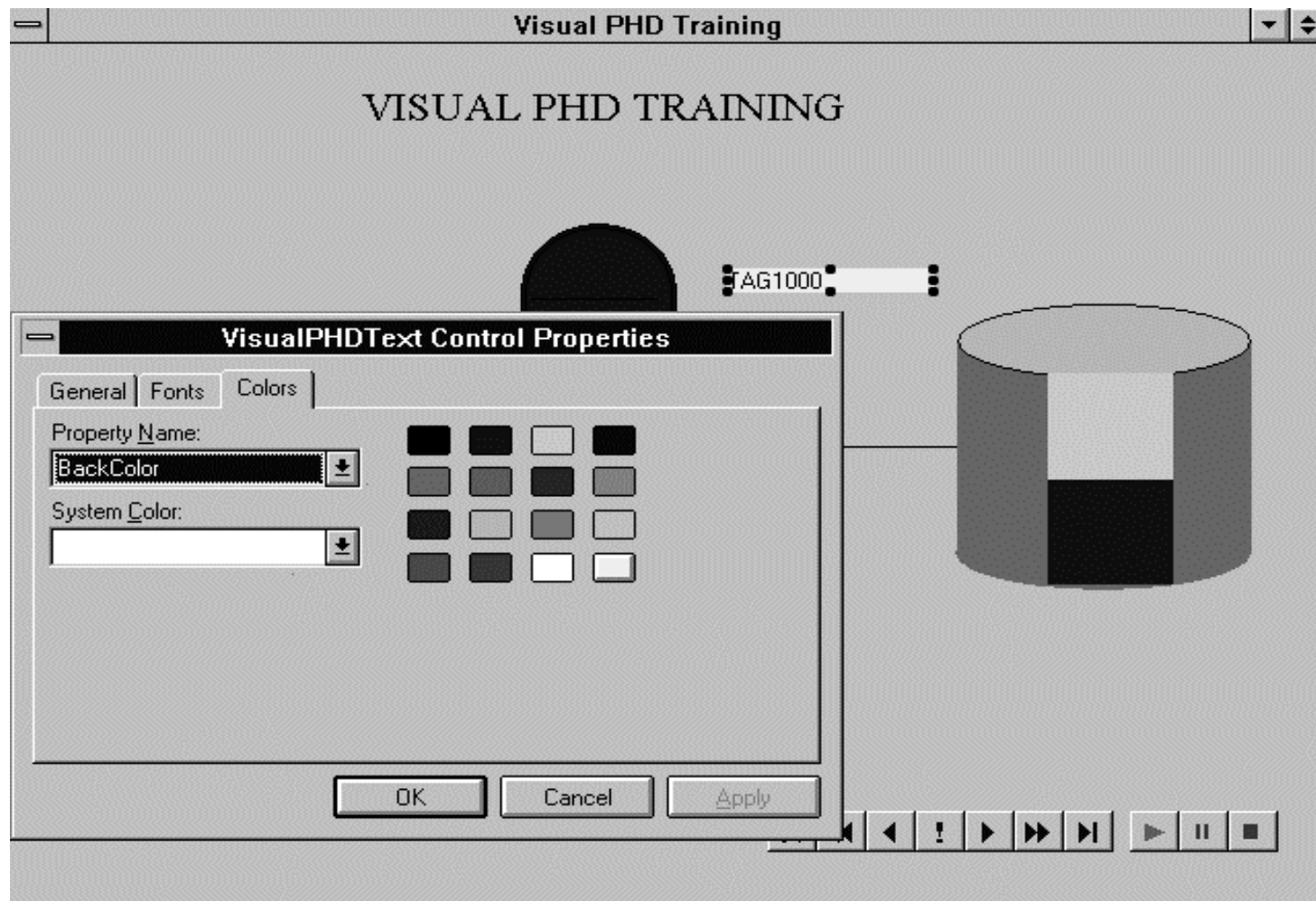
---



## PHD Text Object - Properties, *continued*

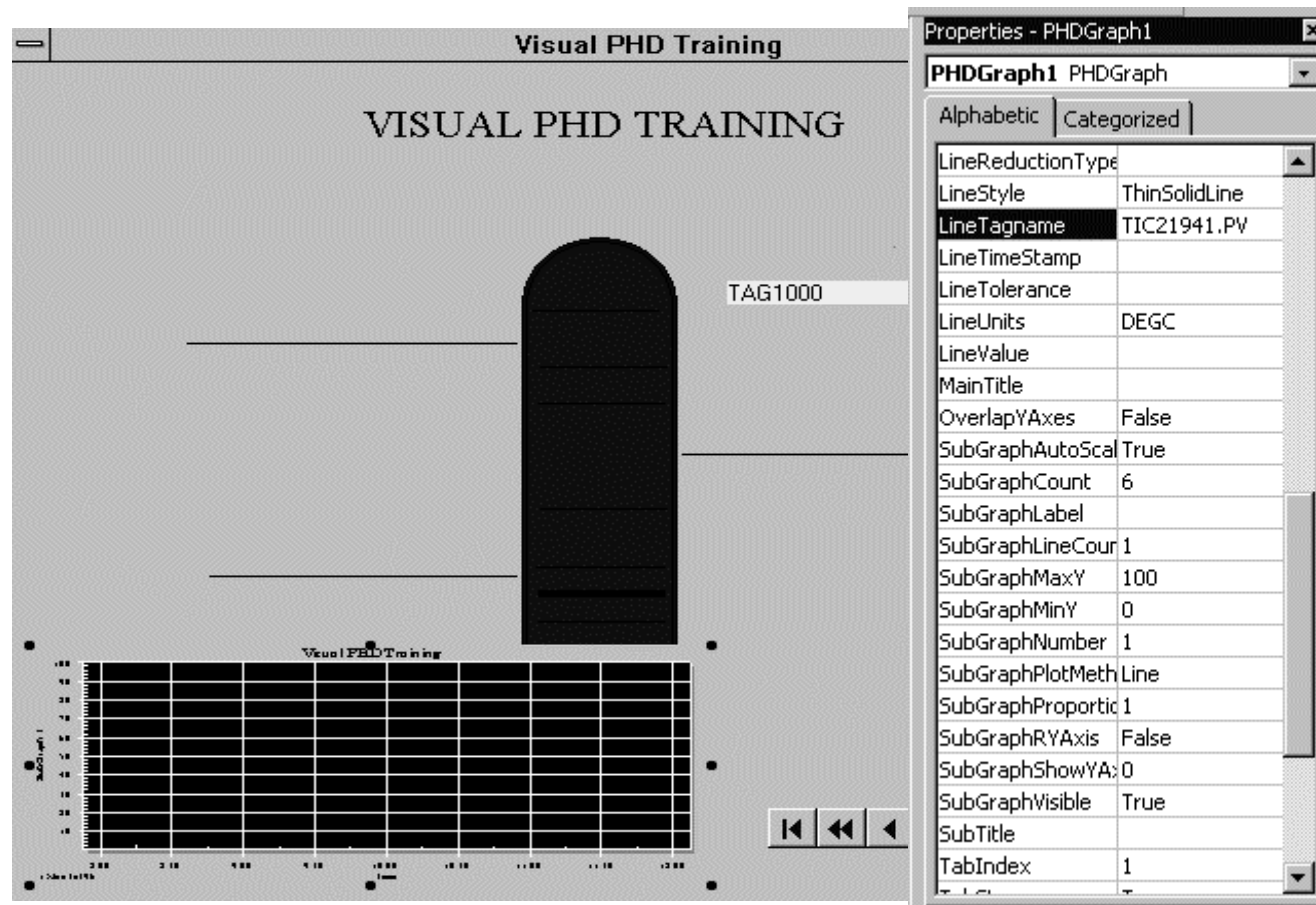


## PHD Text Object - Properties, *continued*



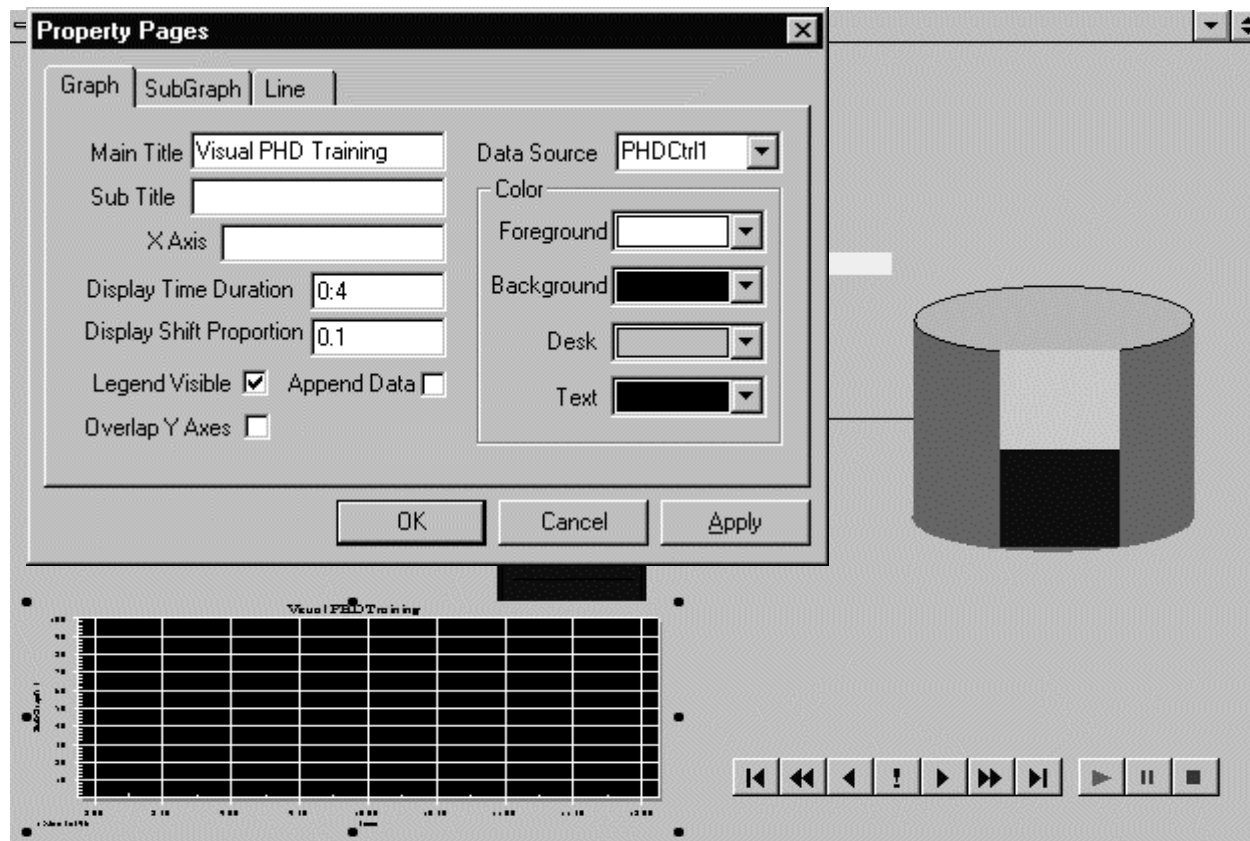
# PHD Graph Object - Properties

---



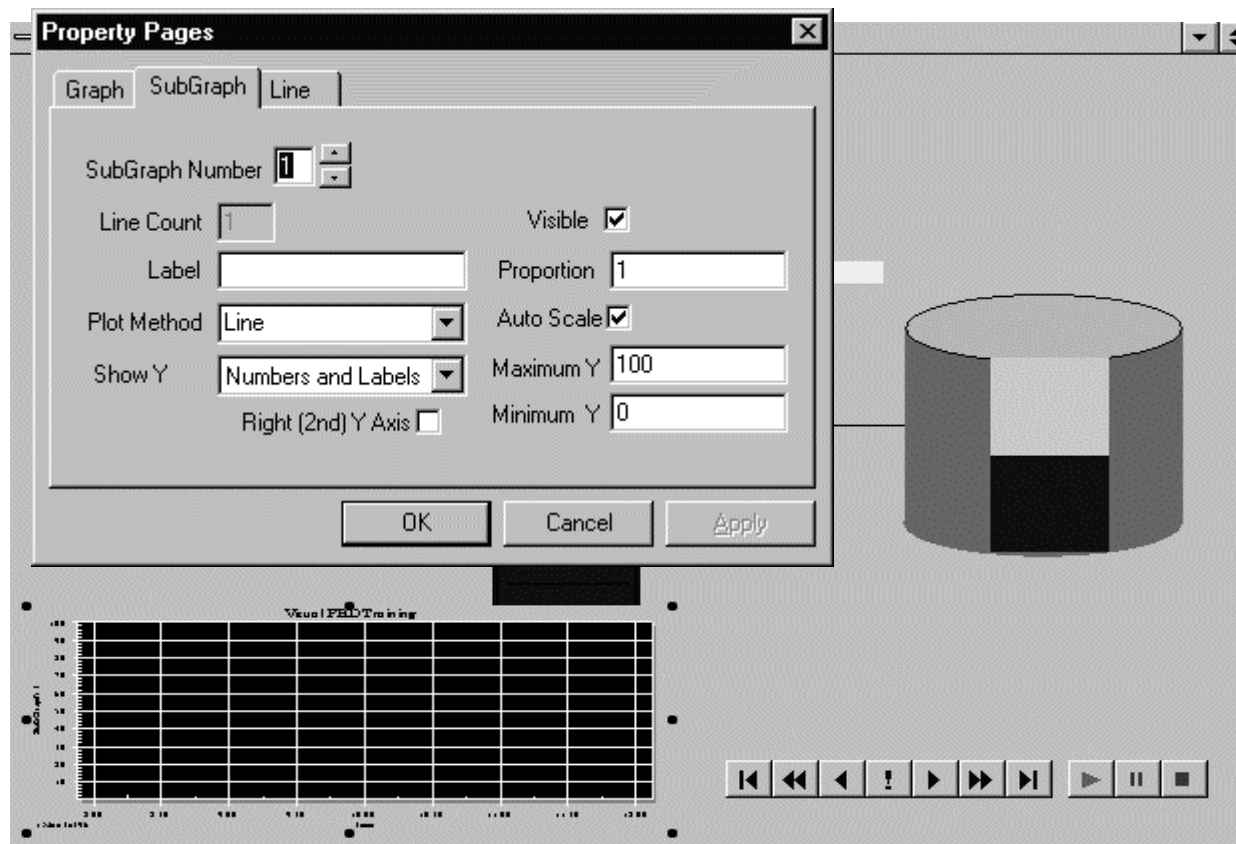
## PHD Graph Object - Properties, *continued*

---



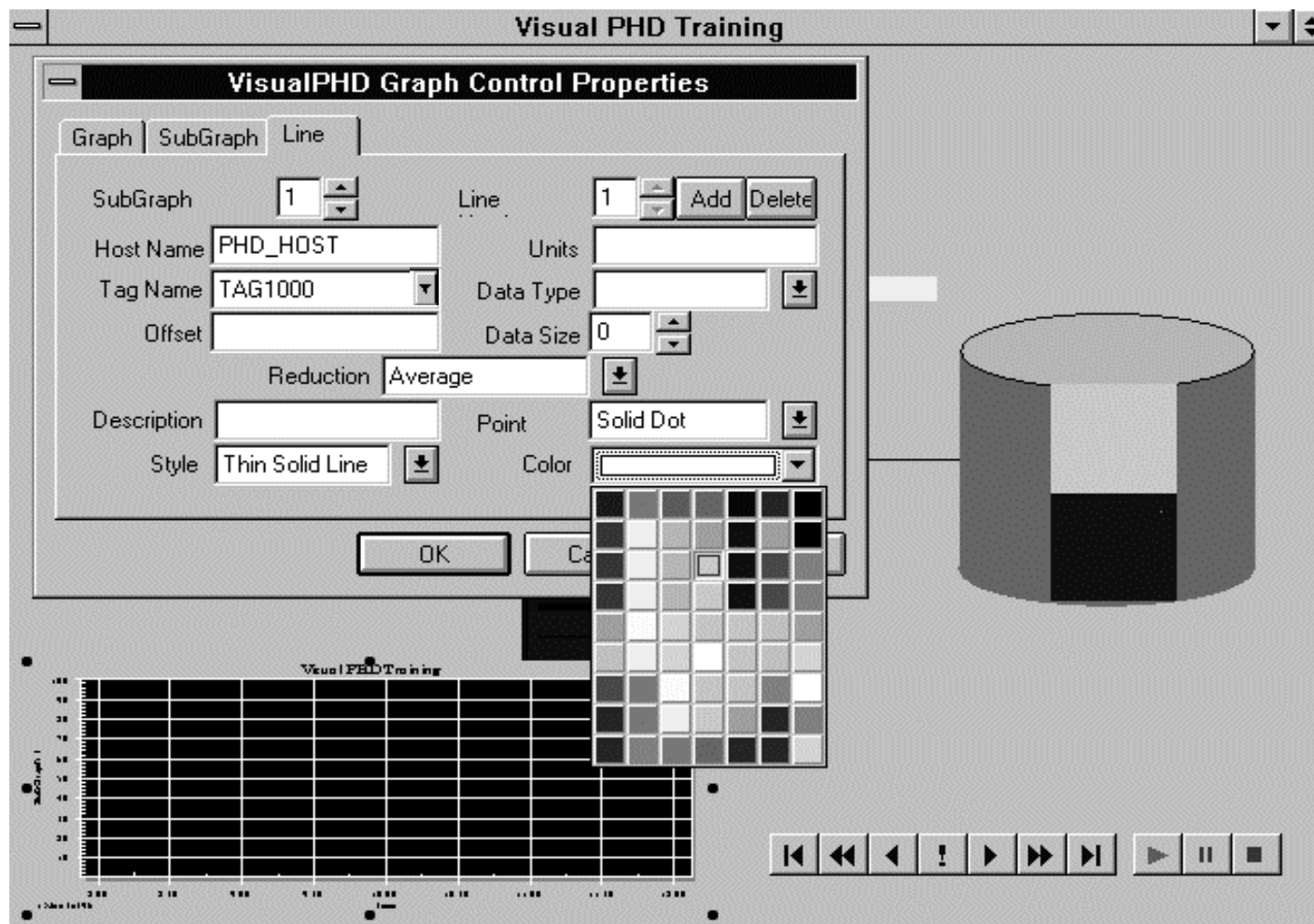
## PHD Graph Object - Properties, *continued*

---



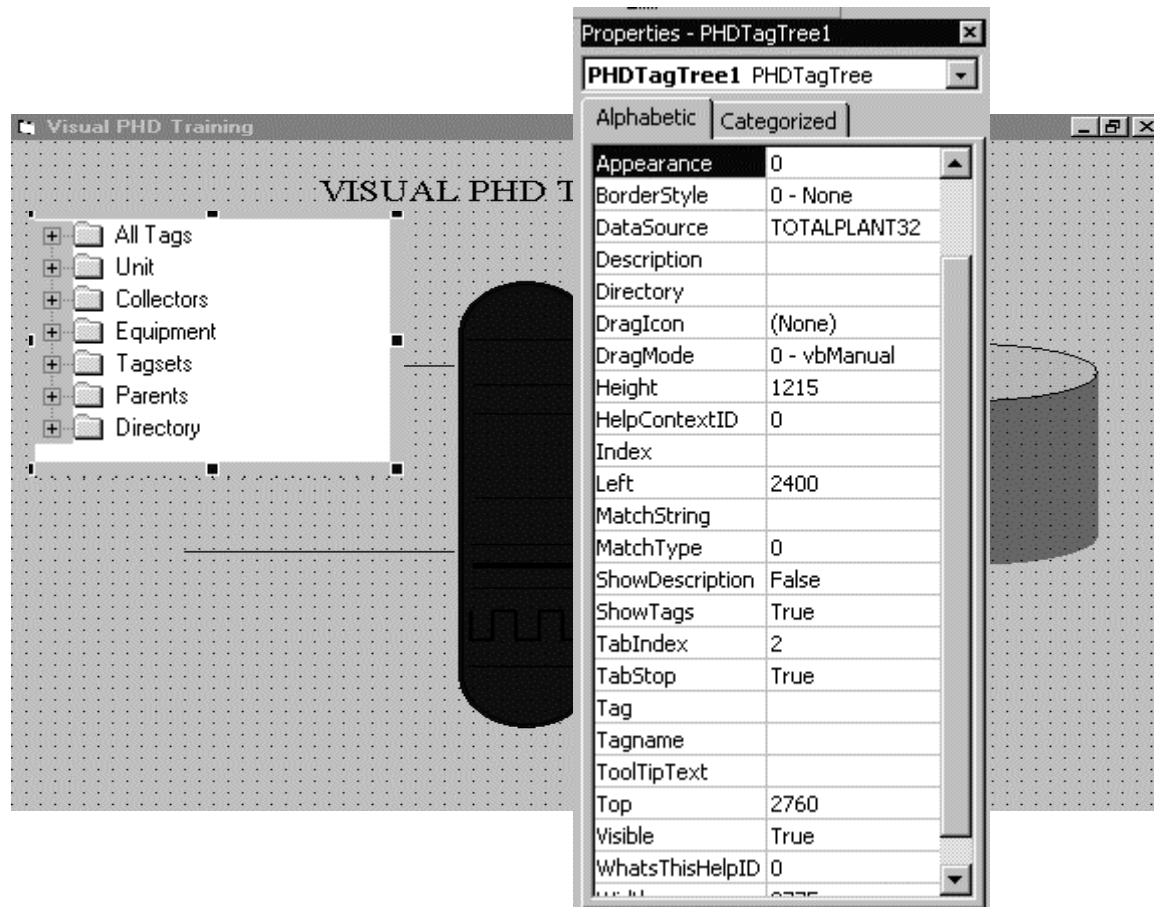


## PHD Graph Object - Properties, *continued*



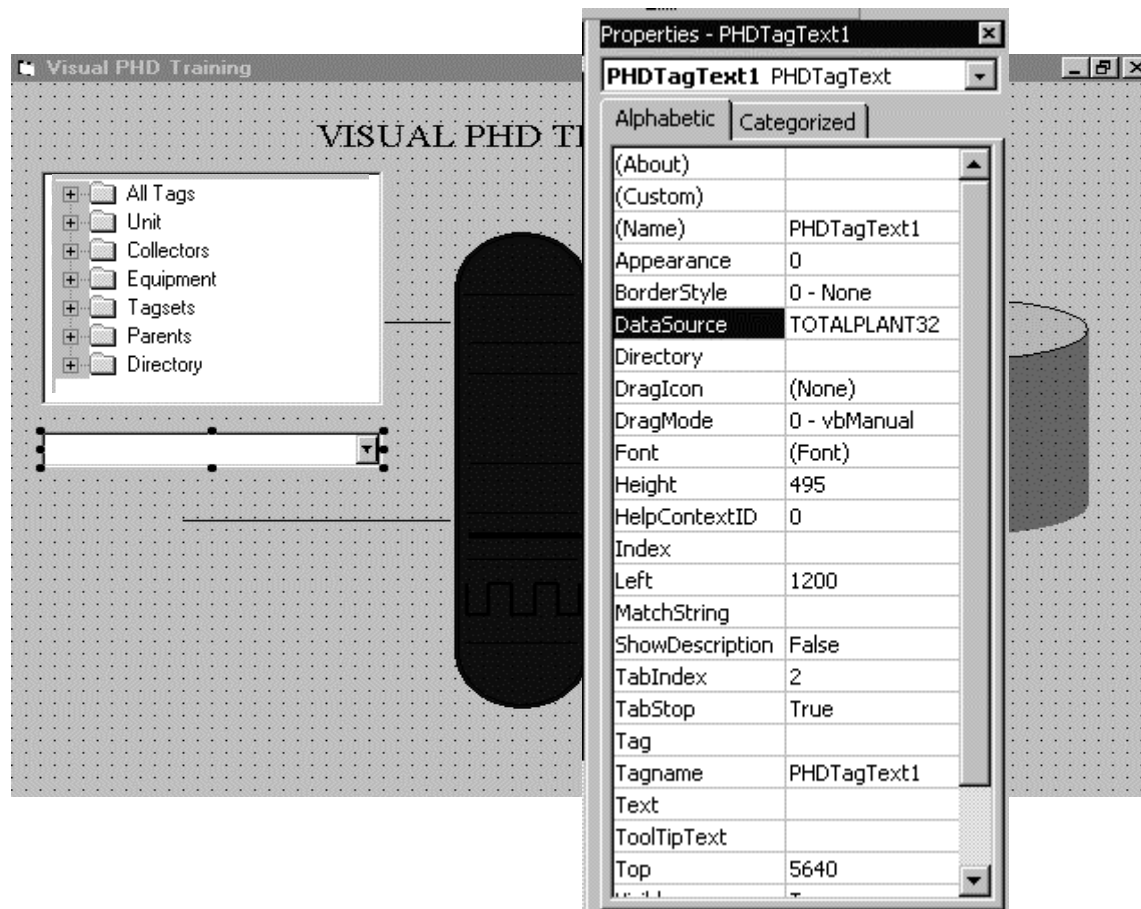
# PHD Tag Tree Object - Properties

---



## PHD Tag Object - Properties, *continued*

---



# Events

---

## Data



- AfterFetch
- BeforeFetch
- Clear
- DragDrop
- DragOver
- GotFocus
- LostFocus
- Move
- Play
- Stop

## Bar



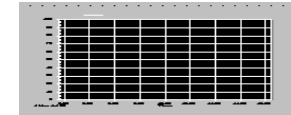
- AfterPut
- BeforePut
- Click
- DoubleClick
- GotFocus
- LostFocus
- DragDrop
- DragOver

## Text



- AfterPut
- BeforePut
- Click
- DoubleClick
- GotFocus
- LostFocus
- DragDrop
- DragOver

## Graph



- DragDrop
- DragOver
- GotFocus
- LostFocus

## Events, *continued*

---

### Tag Tree



- Click
- DoubleClick
- DragDrop
- DragOver
- GotFocus
- LostFocus
- Selected
- SelChanged
- Escape Key

### Tag Text

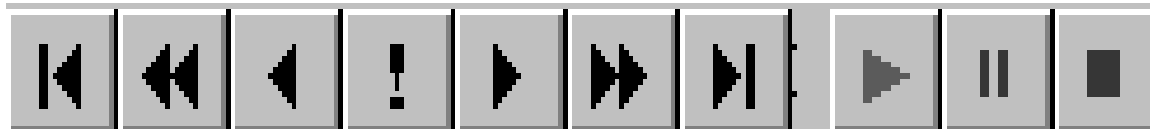


- Click
- DoubleClick
- DragDrop
- DragOver
- GotFocus
- LostFocus
- Selected

# Methods

---

## PHD Data Control



- PHDCtrl.**Fetch** Get specified PHD Data
- PHDCtrl.**MoveFirst,Last** Go to First, Last data record
- PHDCtrl.**MovePrevious,Next** Go to Previous, Next data record
- PHDCtrl.**Play, Pause, Stop** Play data, Pause or Stop in real-time mode
- PHDCtrl.**UnpackTimestamp** Retrieve all Timestamp information
- PHDCtrl1.**Login, Logout** Explicit login to , logout of PHD

# Methods, *continued*

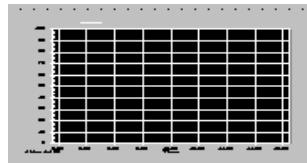
---

## PHD Text and Bar Controls



- PHDText.***UnpackResults*** Load retrieved data into an array
- PHDText.***Put*** Put values into PHD and the DCS
- PHDText.***Refresh*** Refresh the control

## PHD Graph Control



- PHDGraph.***Export*** Export data from the graph
- PHDGraph.***Maximize*** Maximize the graph to fill the screen
- PHDGraph.***Refresh*** Refresh the graph
- PHDGraph.***Print*** Print the graph

## Methods, *continued*

---

### Tag Text Control



- *PHDTagText.Collapse* Collapse the TagText Control
- *PHDTagText.HideTree* Hide the TagTree Control Dropdown for the control.
- *PHDTagText.OpenDatabase* ODBC Connect to a database
- *PHDTagText.ShowTree* Show the TagTree Control Dropdown for the control

### Tag Tree Control



- *PHDTagTree.Collapse* Collapse the TagText Control
- *PHDTagTree.OpenDatabase* ODBC Connect to a database

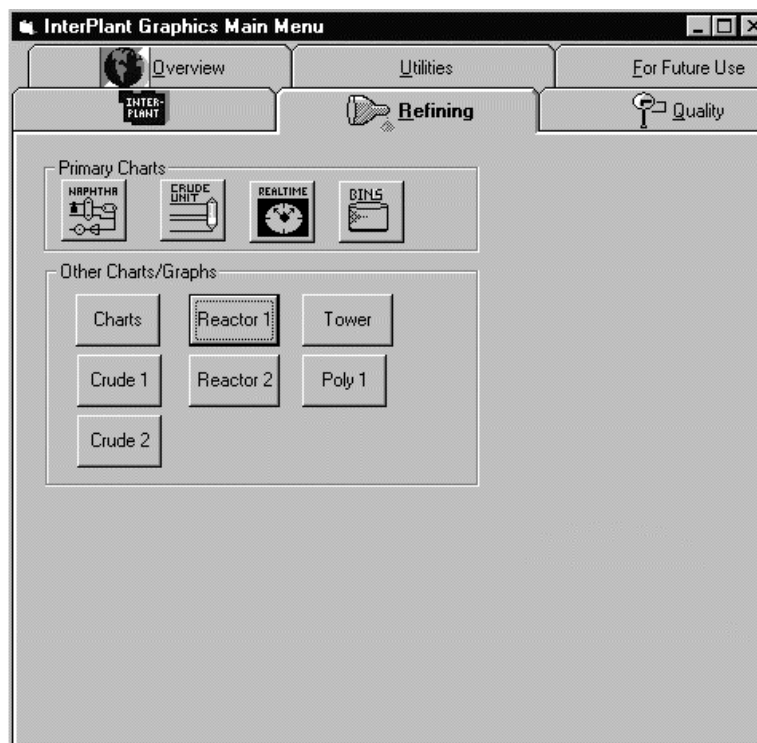


# VB Examples

---

Honeywell includes VB examples with the PHD product.

Start/Programs/TotalPlant Information/VisualPHD/Visual Basic Examples:



# System Requirements

---

## Server

No changes required.

## Client

Windows NT, Windows 95, or Windows 98  
486 or better

Currently VisualPHD supports only the Intel environment.

16 MB RAM minimum

Uniformance Desktop VisualPHD Installed

32 Bit ODBC Driver for ORACLE

## Development Environment

Any OLE compliant development tool

Visual Basic 5.0 or 6.0 (32 bit)

With PHD R110 and later, VisualPHD does  
not work with VB 4.0 or earlier.

Visual C++

Delphi

Powerbuilder

# Hands-on Exercise

---

## Excel VBA:

1. Retrieve PHD data using the Honeywell Excel examples.
2. View the VBA code for the examples.
3. Enter the same code used during the instructor demonstration into a new Excel macro and run the macro to retrieve PHD data.
4. Optional: Enter code to do a Put Download to the DCS.

```
Private Sub Worksheet_Activate()  
Dim A As Object  
Set A = CreateObject ("VISUALPHD.DATA")  
A.HOSTNAME = "PCn"  
A.TAGS.Add "Gnn.PUT"  
A.TAGS("Gnn.PUT").PUT Cells (1,1)  
End Sub
```

## VB:

1. View the Honeywell VB examples.
2. Optional - Create a VB graph control and a data control that accesses data for a PHD tag.

---

# Honeywell

---

*Helping You Manage Your World*

**[www.iac.honeywell.com](http://www.iac.honeywell.com)**