Creating HyperView Reports

Introduction

Version 3.0 of PC-DMIS greatly enhanced the reporting capabilities of prior versions with the integration of HyperView™, advanced reporting tools from HyperCube Inc. and Dundas Software. Using this tool, you can now lay out custom reports (called “HyperView Reports”) from within PC-DMIS. The new reporting tools are tightly integrated with PC-DMIS BASIC, allowing advanced users to layout VB-like forms that support user interaction and Basic language event handlers (“scripted forms”).

The following topics have been organized into a series of tutorial steps to guide you through the creation and use of a HyperView Report.

The main topics in this chapter include:

- Creating a New HyperView Report
- Adding Objects to the HyperView Report
- Changing Object Properties with the Property Sheet
- Performing Events using BASIC Handler Functions
- Using Objects and Controls in HyperView Repots
- Working with the Grid
- Using the Layout Bar to Organize Your Report
- Using Layers to Manage Multiple Objects
- Creating Multiple Page Reports
- Generating Automatic Reports
- Using HyperView Reports in Part Programs

For additional information beyond the scope of this tutorial, download a complete Hyperview documentation set from Dundas Software: http://www.dundas.com/downloads/docs/files/HViewdocs.exe

Creating a New HyperView Report

All HyperView Reports are initially associated with a specific part program.

To create a HyperView Report select the **View | Hyperview Editor** menu option. A blank report window opens with several new toolbars.

HyperView Reports may be viewed in one of two modes:

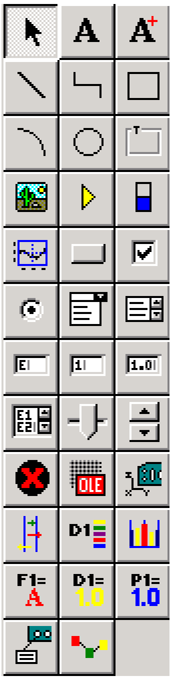
1) Edit Mode

2) Run Mode

Later topics will explain how to switch between Edit and Run Modes. For now, you only need to know that the two modes exist and that a new report defaults to Edit Mode when created.

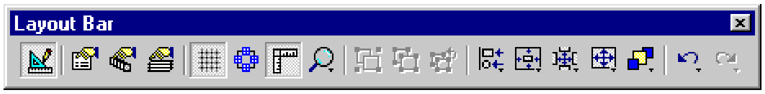
Edit Mode

Upon creation, a new HyperView Report window is opened by default in Edit Mode. An **Object** toolbar similar in appearance to those found in many dialog box editors will be present, as shown here:



Object toolbar

A number of toolbars for editing the objects are also present, such as this layout toolbar:



Layout toolbar

Reports are designed and their component objects placed while in Edit Mode. Objects, such as bitmaps, text, frames, and other form components can be moved and sized while in Edit Mode. Object properties, like fonts, colors, and background styles, can all be set while in Edit Mode.

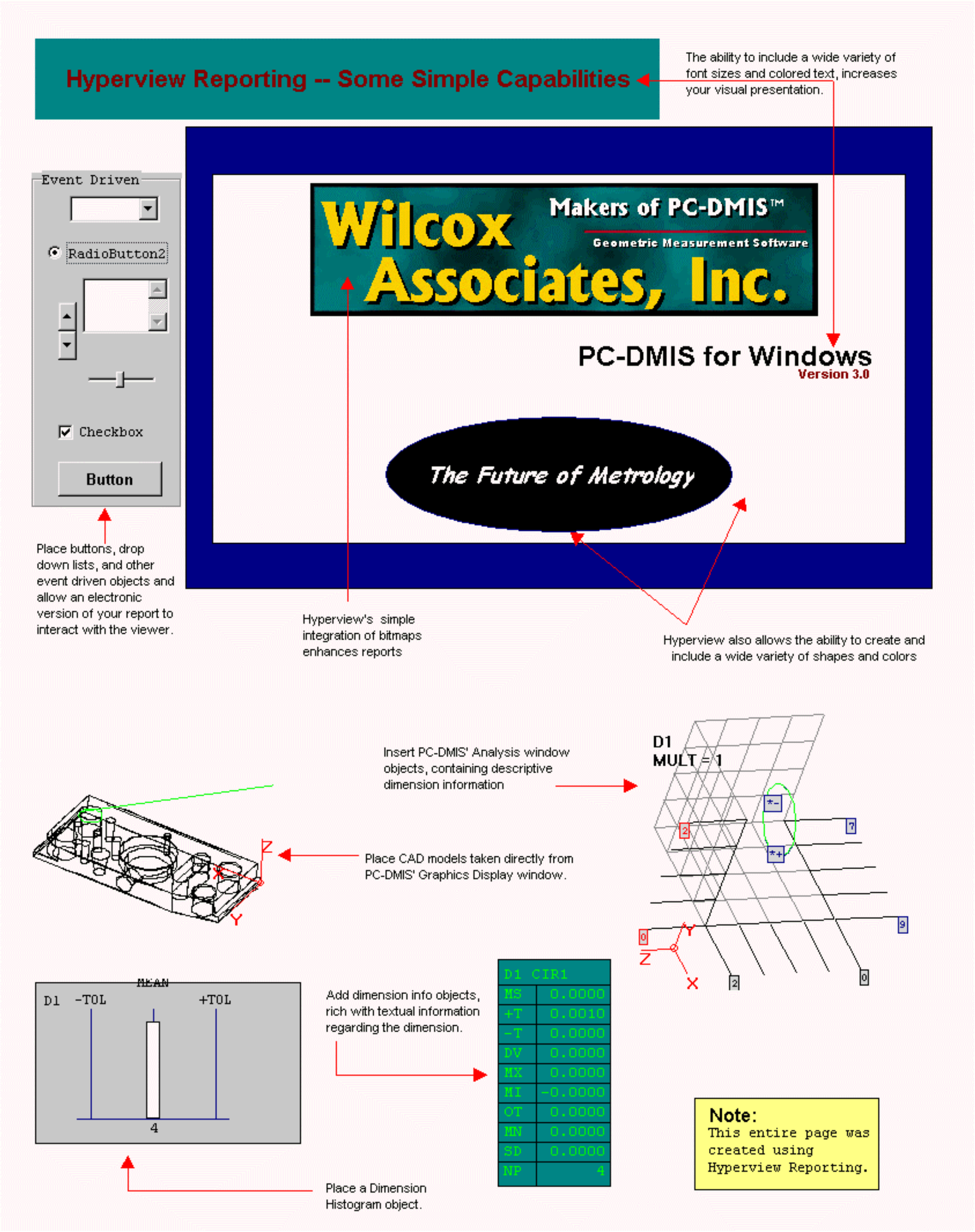
These objects give you many reporting options such as:

- Adding Bitmap Objects

- Adding Text Objects
- Adding Analysis Window Objects
- Adding Cad Objects
- Adding Dimension Info Objects
- Adding Dimension Histogram Objects

These and other options are described in greater detail later.

Consider this sample report.



Hyperview Reporting -- Some Simple Capabilities

The ability to include a wide variety of font sizes and colored text, increases your visual presentation.

Place buttons, drop down lists, and other event driven objects and allow an electronic version of your report to interact with the viewer.

Hyperview's simple integration of bitmaps enhances reports

Hyperview also allows the ability to create and include a wide variety of shapes and colors

Insert PC-DMIS' Analysis window objects, containing descriptive dimension information

Place CAD models taken directly from PC-DMIS' Graphics Display window.

Add dimension info objects, rich with textual information regarding the dimension.

D1	CIR1
MS	0.0000
+T	0.0010
-T	0.0000
DV	0.0000
MX	0.0000
MI	-0.0000
OT	0.0000
MN	0.0000
SD	0.0000
NP	4

Place a Dimension Histogram object.

Note:
This entire page was created using Hyperview Reporting.

Run Mode

Run Mode is used to view the finished HyperView Report, whereas Edit Mode is used to design and/or edit a new or existing report. When a HyperView Report is viewed in Run Mode, the **Object** toolbar and other layout and edit toolbars that were present in Edit Mode will be hidden.

Adding Objects to the HyperView Report

To add an object to a HyperView Report:

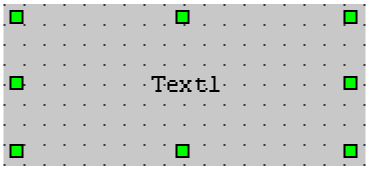
1. Click on the desired object in the **Object** toolbar.
2. Hold down the left mouse button and drag a rectangle at the desired location in the HyperView Report.
3. Release the mouse button. The object appears in the HyperView Report. The object is given a default name and other default attributes that you can change later. (See " Edit in Place" on page 393 and " Access the Property Sheet" on page 395 for additional information.)

Add an Object

Add a text object to the blank HyperView Report you’ve created:

A *Text object button*

1. Click on the **Text** icon on the **Object** toolbar.
2. Hold down the left mouse button and drag a rectangle on the HyperView Report at the position you wish the text to appear.
3. Release the mouse button. Upon creation, the object is selected, as shown by the small green squares at each corner of the text object.



Text Object

4. Click elsewhere in the HyperView Report—anywhere but within the boundaries of the text object—the text object is no longer selected.

Select an Object

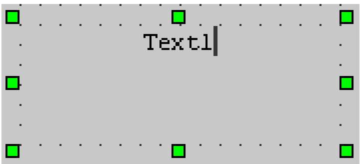
Click within the boundaries of the text object. You will see the small green squares re-appear. The object is now selected.

When an object is selected, you can use the toolbars to modify its properties. See if you can figure out how to modify the font face and size for the text object using the toolbars. You may also want to experiment with different border types, background and foreground colors, and patterns using the toolbars.

Edit in Place

The text object also allows in-place editing.

1. Make sure the object is selected.
2. Click once directly on the text in the text object. A cursor should appear next to the text, allowing you to directly edit the object.



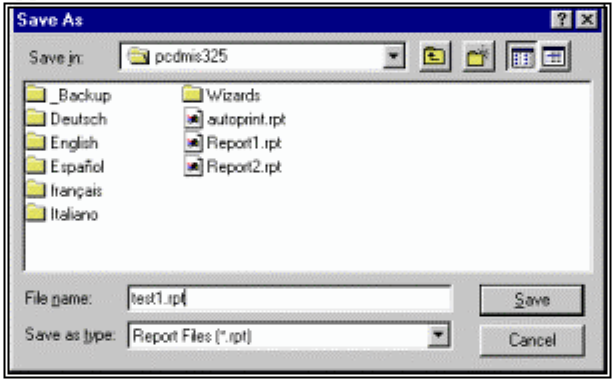
Modifying the text

3. Try adding a few different kinds of objects to the report and modifying some of their properties from the toolbar.

Save the Report

When finished, save the report. To do this:

1. Select **File | Save** from the menu bar. The **Save As** dialog box appears.



Save in: pcdmis325

File name: test1.rpt

Save as type: Report Files (*.rpt)

Save As dialog box

2. Navigate to the directory where you want to save your report.
3. Type a file name for the report in the **File name** box.
4. Click the **Save** button.

The first time you save a report, the **Save As** dialog will open, since you must give the report a filename and specify what folder you wish to save it in. By default, HyperView report files are given the filename extension ". rpt".

Saving your Report to an Older Version of PC-DMIS

PC-DMIS versions 3.6 and higher allow you to save a HyperView report so that it's compatible with older versions of PC-DMIS. To do this, select the appropriate version type from the **Save as type** list of the **Save As** dialog box before clicking the **Save** button. PC-DMIS will use save future reports as the selected version type until you select a different version.

Test the Report

With the report saved, you're now ready to test the report for the first time in Run Mode. The CTRL + E key combination switches the HyperView report between Edit and Run Modes. Alternately, you can use the menu item for Edit/Run Mode found under the **Report** menu.

1. Try the CTRL + E key combination now. HyperView moves into Run Mode. The HyperView toolbars should disappear, as well as the grid that formed the background of the HyperView Report.

2. Now switch back to Edit Mode using the same CTRL + E key combination.

In Run Mode, whatever objects you created in your report are present and active, but are no longer editable. Whenever PC-DMIS loads a HyperView report from a part program, the HyperView report defaults into Run Mode. You can Print or Print Preview your report from Run Mode.

Note: Print and Print Preview are *not* available when in Edit Mode.

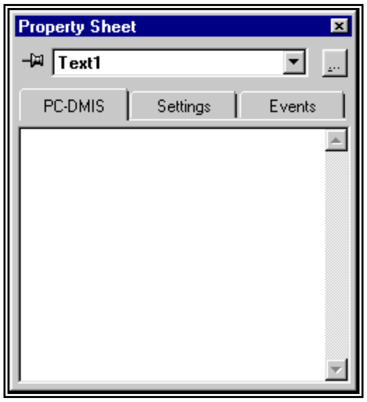
Changing Object Properties with the Property Sheet

Each object added to a HyperView report has a set of properties (or attributes) associated with it that control such things as the object's position, size, colors, font, text, and name. Each object also has associated with it a set of event handler functions that are triggered when certain events take place, such as the mouse being clicked or double clicked on that object. These properties and event handlers may be viewed and/or modified using a dialog called the **Property Sheet**.

Access the Property Sheet

The property sheet for an object may be viewed by highlighting the object (by single clicking within its boundaries) and then right-clicking. Alternately, double clicking most objects will bring up the property sheet. However, the double clicking fails with some objects (such as OLE objects), so it is recommended that you use the former (select and then right-click) method.

1. Select the text object you created in the prior section.
2. Click you right mouse button. The **Property Sheet** appears.



Property Sheet

3. Click the drop-down list box at the top of the **Property Sheet**.

Notice that this list box contains an entry for each of the objects in your HyperView report, as well as an entry entitled **The Frame/The View**. This entry will be discussed in more detail later. The currently selected entry should be "Text1" (or something similar), since you are viewing the properties for the first text object that you created. The name identifier "Text1" was automatically assigned to the object when you created it, but can be modified from the **Property Sheet**. For information on how to use the **Property Sheet** to modify objects, see the example after the note below.

Once you have the **Property Sheet** open, you can view the properties for any of the objects in your HyperView report by selecting a different object in the drop-down list box at the top of the **Property Sheet**.

The **Property Sheet** for most objects has three tabs:

1) PC-DMIS

2) Settings

3) Events

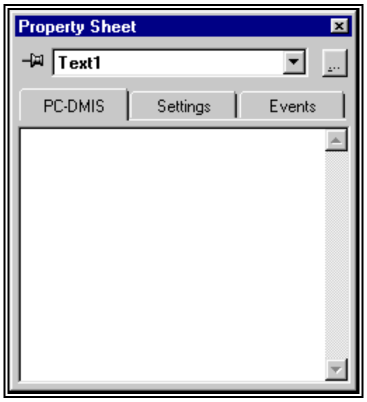
These are explained in detail below:

Note: ActiveX and OLE objects have an additional tab, which will be discussed in "ActiveX Object" on page 433 and "OLE Object" on page 434.

Example: To modify your "Text1" name identifier using the property sheet follow these steps:

1. Select the text object on the screen
2. Press the right mouse button. The **Property Sheet** opens.
3. Click the **Settings** tab.
4. The first item you see in the **Settings** tab is **(ObjectCode)** displayed in the left column. In the right hand column you will see "Text1", the name identifier.
5. Select "Text 1" and rename it. Once you close the **Property Sheet**, the name identifier "Text1" will have changed to the new name you selected.

PC-DMIS Tab



Property Sheet —PC-DMIS tab

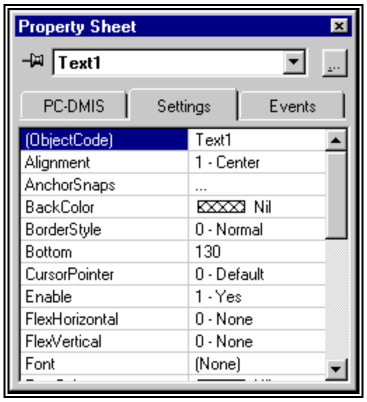
The **PC-DMIS** tab contains properties specific to PC-DMIS and is currently empty of many of the standard objects. However, many PC-DMIS objects contain settings in their PC-DMIS tab. These allow you to individually customize each PC-DMIS object.

See these topics for Information specific to each object PC-DMS extension object:

- "Cad Object" on page 435
- "Analysis Object" on page 436

- "Clamping Value" on page 195
- "Modifying Headers and Footers " on page 338
- "Feature Text Object" on page 439
- "Leader Line Object" on page 442

Settings Tab



(ObjectCode)	Text1
Alignment	1 - Center
AnchorSnaps	...
BackColor	▨ Nil
BorderStyle	0 - Normal
Bottom	130
CursorPointer	0 - Default
Enable	1 - Yes
FlexHorizontal	0 - None
FlexVertical	0 - None
Font	(None)

Property Sheet—Settings tab

The **Settings** tab contains the core set of common properties for the standard HyperView objects. As an example, see the following description of some of the more important text object properties. Many of these properties are common to all HyperView objects.

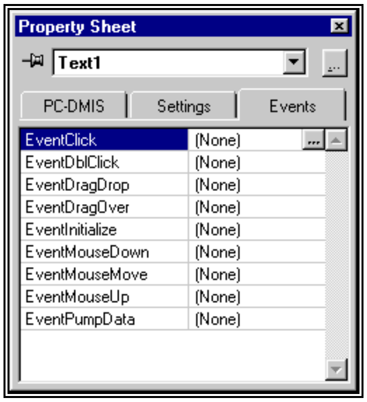
Property	Description
(Object Code)	Unique name that identifies an object in a HyperView report. Whenever you need to access a property or method of a HyperView object from basic, use the following syntax: **{Object Code}.{property or method name}** When you create a HyperView object, it is automatically assigned a unique object code. For example, the fifth text object you add to a HyperView report will probably have Object Code “Text5”. You can edit the object code for a HyperView object, but must make sure that it is unique. Any Basic script that makes use of the object code must also be updated if the object code is changed.
Alignment	Text alignment within the text object’s defining rectangle.
AnchorSnaps	Field which allows you to create, delete or modify snap points by using the **Enter Snap Points** dialog box. Snap points are found on the

	ellipse, rectangle, text box, or bitmap image objects. You can connect **Line** or **Polyline** objects to an object's snap points by dragging tip of the line onto the displayed blue snap point. Once connected, when you drag the object to a new location The resulting line remains connected to the snap point, rotating, elongating, and shrinking as necessary. This feature is helpful when you need to create a "leader-line" that constantly points to an object that you'll frequently be moving. To display snap points, from the **Layout** toolbar click the **View Snap Points** icon:
BackColor	Background color of the object. This color is used to fill the object’s defining rectangle.
BorderStyle	Style of border drawn around the object.
Bottom	Distance, in pixels, of the bottom of the object’s rectangle from the top of the HyperView report.
CursorPointer	Cursor displayed when the user drags the mouse over the object.
Enable	Enable or disable the object. A disabled object may still be visible, but will not respond to user input.
FlexHorizontal and **FlexVertical**	When the view is resized in Run mode you can elect to shift, expand, or keep objects proportional. Selecting **Shift** moves the object to the right or down. Selecting **Expand** enlarges the object right or down. Selecting **Proportional** centers the object.
Font	Font used for text rendered in the object.
ForeColor	Foreground color of the object. In the case of the text object, specifies the text color.

HatchStyle	Hatch style used in filling the background box of the object.
HelpContextID	This is the context ID of a specific help topic to call. This is often used in conjunction with the Help button type which invokes WinHelp( ).
HiliteColor	(This object not included with PC-DMIS)
Layer	This assigns the current object to the selected layer.
Left	Distance, in pixels, of the left side of the object's rectangle from the left side of the HyperView report.
Orientation	Orientation of the text inside the object. Horizontal text flows from left to right. Vertical Up flows from bottom to top. Vertical Down flows from top to bottom.
PenWidth	Width of the pen used to draw the object. In the case of the text object, the pen width only affects the border drawn around the text.
Right	Distance, in pixels, of the right side of the object's rectangle from the left side of the HyperView report.
ShadowStyle	Shadow effect used when rendering the object.
Tag	Place holder that stores user-defined data.
Text	Text appearing in the text object. You can also type an ampersand symbol (&) in front of any alphanumeric character in this field to create a shortcut key (shown as an underlined character) that when pressed with the ALT key in Run Mode, causes that control to immediately have the focus. If you plan on using a lot of shortcut keys, you can ensure they are unique by using the **Edit | Layout | Check Mnemonics** menu item in Edit

	Mode.
ToolTipText	Text displayed when you move the mouse over the object inside Run mode.
Top	Distance, in pixels, of the top of the object's rectangle from the top of the HyperView report.
Visible	Visibility state of the object. A value of **1** corresponds to visible, **0** to hidden.

Events Tab



EventClick	(None)
EventDblClick	(None)
EventDragDrop	(None)
EventDragOver	(None)
EventInitialize	(None)
EventMouseDown	(None)
EventMouseMove	(None)
EventMouseUp	(None)
EventPumpData	(None)

Property Sheet dialog box—Events tab

The **Events** tab contains a list of all of the event handler functions that an object supports. This list is covered in more detail in "Performing Events Using BASIC Handler Functions" on page 400.

Performing Events Using BASIC Handler Functions

This section assumes a reasonable proficiency in the BASIC programming language. If you need to learn BASIC, please consult a book on the subject. The **Events** tab on the **Property Sheet** contains a list of event handler functions that a HyperView object supports. An event handler function will be called whenever the event implied by the name of the function occurs for the given object. In order to add BASIC language code to any of the event handlers, double click on the function's entry in list of event handlers. A BASIC language editor will pop up, with the opening and closing statements of the subroutine already filled in. All you need to do is add BASIC language code to the handler.

Event Handlers

Before you add any event handler code, here is a description of a set of event handlers that are supported by many of the different HyperView object types:

EventClick

Called when the mouse is clicked on the object.

EventDblClick

Called when the mouse is double clicked on the object.

EventDragDrop

Called when another "draggable" object is dropped onto the HyperView object.

EventDragEnter

Called when a draggable object is initially dragged onto the HyperView object.

EventDragOver

Called when a draggable object is dragged over the HyperView object.

EventInitialize

Called when the HyperView first enters into Run Mode before any other handlers are called on the object.

EventMouseDown

Called when a mouse button is depressed while over the object. Parameters passed to the function indicate which button was pressed, if the SHIFT key was pressed, and the coordinates of the mouse at the moment the mouse button was depressed.

EventMouseMove

Called when the mouse is passed over the object. Parameters passed to the function indicate which button was pressed, if the SHIFT key was pressed, and the coordinates of the mouse when the move event was sent.

EventMouseUp

Called when the mouse button is released over the object. Parameters passed to the function indicate which button was pressed, if the SHIFT key was pressed, and the coordinates of the mouse at the moment the mouse button was released.

EventPumpData

This is perhaps the most widely used handler function. **EventPumpData** is called whenever BASIC variables that may affect the object change value. Your basic code can also force **EventPumpData** to be called on individual objects (or on all objects in a report) when you wish an object to update itself. Basically, any code that an object needs to reinitialize itself based on variable values or the state of other objects should occur in **EventPumpData**. This handler is particularly important when manually creating multiple page reports, as described in the section "Creating Multiple Page Reports" on page 451.

EventTerminate

Called when the HyperView is being closed or when toggling from Run Mode to Edit Mode.

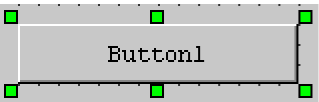
Note: This handler function is only available for **The Frame/The View**.

Add a Callback Function

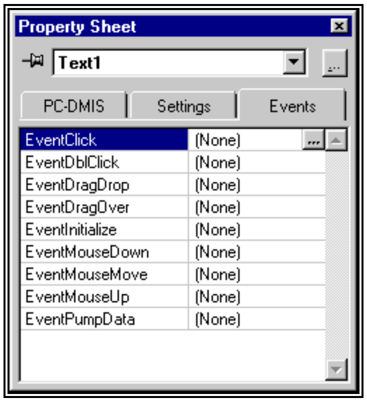
Now let's try adding a simple callback function.

***Button** icon*

First, drop a button into your HyperView report by clicking on the **Button** icon on the **Object** toolbar and then dragging out a rectangle for the button on the HyperView report.

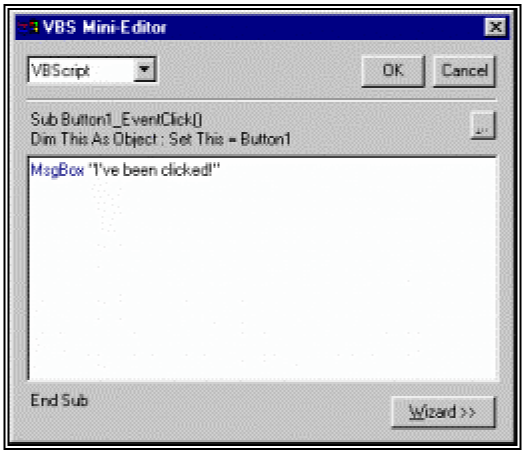


1. Make sure that the button is selected (highlighted).
2. right-click on the button to bring up the button's **Property Sheet**. dialog box.
3. Click on the **Events** tab on the **Property Sheet**. You should see a list of all of the available event handler functions for the button object.



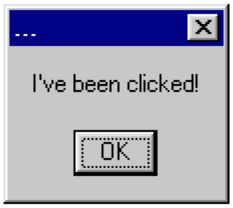
EventClick	(None)
EventDblClick	(None)
EventDragDrop	(None)
EventDragOver	(None)
EventInitialize	(None)
EventMouseDown	(None)
EventMouseMove	(None)
EventMouseUp	(None)
EventPumpData	(None)

4. Double click on the **EventClick** function in the list. A Visual BASIC Editor entitled **VBS Mini-Editor** opens.
5. Add the following code to the **EventClick** function: `MsgBox "I've been clicked."`



```
Sub Button1_EventClick()
Dim This As Object : Set This = Button1
MsgBox "I've been clicked!"
End Sub
```

6. Click the **OK** button in the upper right hand corner of the **VBS Mini-Editor**. Close the **Property Sheet**.
7. Save the report by selecting **File | Save**. You are now ready to test the report in Run Mode.
8. Switch to Run Mode (press CTRL + E).
9. Click on the button you created. A message box with the text “**I’ve been clicked.**” should appear.



I've been clicked!

This simple example illustrates the basics of scriptable forms and event handlers.

Modify Object Properties on Triggered Event

Now, suppose you want to modify the properties of another object (say the text in a text object) when the button has been clicked.

1. Switch back to Edit Mode (press CTRL + E).
2. Verify that the original text object that you created still exists. If it doesn't, create a new text object.
3. Open the text object’s property sheet (select and then right mouse click).
4. Click the **Settings** tab.
5. Make sure that the text item’s **Object Code** property reads “Text1”. The **Object Code** property of a HyperView object is the name that is used to refer to that object from BASIC code.
6. Click on the **Events** tab.
7. Double click on the **EventInitialize** handler function. The **VBS Mini-Editor** opens.
8. Add the following code to the **EventInitialize** function:

   ```
   Set This = Text1
   ```
9. Press the ENTER key to get to a new line. On the next line type this code:

   ```
   This.Text = “The button has not yet been
   clicked”
   ```
10. Click the **Ok** button to close the editor.
11. Once the object Text1 has been set up as described, open the property sheet for the button object added in the prior exercise.
12. Click the **Events** tab.
13. Double click on the **EventClick** function.
14. Replace the basic code you added in the previous exercise with the following two lines:

    Line 1: `MsgBox “I’ve been clicked, and am about`

`to modify Text1's text"`

Line 2: **`Text1.Text = "The button was clicked!!!"`**

15. Click the **OK** button to close the **VBS Mini-Editor**.
16. Close the **Property Sheet**.
17. Switch to Run Mode (press CTRL + E). The code you added to Text1's EventInitialize function will set Text1 to initially read "The button has not yet been clicked".
18. Now try clicking on the button. The "**I've been clicked, and am about to modify Text1's text**" message should appear. As soon as you close the message box, Text1's text should be modified to read "**The button was clicked!!!**".

This example illustrates how the properties of any object in a HyperView report may be accessed from Basic code via the **ObjectCode.property_name** syntax.

Access Object Methods

Similarly, many of the HyperView objects support various methods in addition to their properties. These methods may be accessed from BASIC code via the **ObjectCode.method_name** syntax. You can see what properties and methods are available for any given object by typing the object's Object Code in the **VBS Mini-Editor** followed by a period. As soon as the period is typed, a mini wizard will pop up showing the available methods and properties for that object.

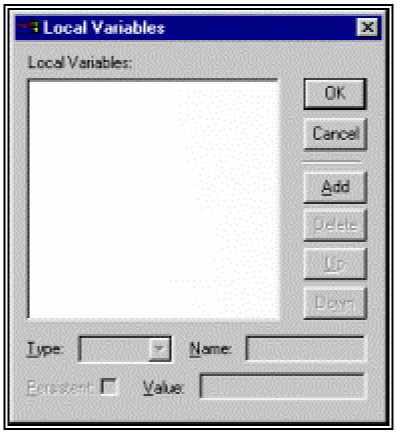
Variables declared within an event handler function (via the BASIC Dim **var_name** as **var_type** syntax) are in scope only during that event handler function. There will undoubtedly arise situations when you would like to declare variables, functions, or subroutines that are accessible from all BASIC code in a HyperView report --- variables with global scope for a particular report. HyperView provides a place to make these declarations.

Declare Global Variables Using the LocalVariables Property

If you wish to declare global variables of any of the standard data types (String, Integer, Double, etc.) and have them show up in the Basic Wizard, you can use the **LocalVariables** property of **The Frame/The View**.

To use the **LocalVariables** property:

1. Open the **Property Sheet** for **The Frame/The View**. Do this by selecting, then right-clicking on an object. The **Property Sheet** will appear.
2. Select the **Settings** tab.
3. Scroll down to the **LocalVariables** property.
4. Double clicking on the **LocalVariables** property will bring up the **Local Variables** dialog box



Local Variables dialog box

The **Local Variables** dialog box allows local variables of any of the basic types to be added, removed, named, and initial values to be set. Again, these variables are global to this HyperView and will be accessible from any of the BASIC code written for this HyperView. They will also appear in the **Local Variables** area of the **Script Wizard**.

Note: The name "LocalVariables" seems something of a misnomer, as variables declared as Local Variables have global scope to that particular HyperView. The syntax was possibly chosen to indicate that the variables declared here are "local" to this particular HyperView report, but nonetheless available anywhere in the HyperView report.

Declare/Define Variables, Functions, or Subroutines

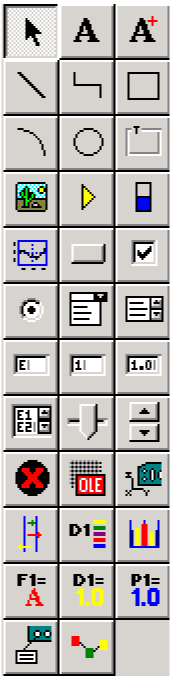
To declare and/or define variables, functions, or subroutines that are to be accessible from all Basic code in a report, use the **LocalDecls** property of **The Frame/The View** found on the **PC-DMIS** tab of the **Property Sheet**. To do this:

1. Open the **Property Sheet** for an object.
2. Select **The Frame/The View** from the drop-down list.
3. Click on the **PC-DMIS** tab.
4. Double click on the **LocalDecls** item on the property sheet. A **VBS Mini-Editor** appears.

You may add any BASIC code you wish to this window, including variable declarations and function or subroutine definitions. Any variables, functions, or subroutines declared here will be global, but will not appear in the **Script Wizard**.

Using Objects and Controls in HyperView Reports

This topic describes each object available to you from the **Object** toolbar. The **Object** toolbar appears in Edit Mode and allows you to insert a wide variety of objects into your report. You can insert text, pictures, multi-media, dialog box controls, and other items.



Object toolbar

PC-DMIS extends the base set of HyperView objects available to your reports with several PC-DMIS specific objects already included on your **Object** toolbar. These objects may be added to a report in the same way as normal HyperView objects, but often have new properties added to the **PC-DMIS** tab of the property sheet and pertain to very specific reporting functions. Where appropriate, the documentation identifies those objects that are unique to PC-DMIS with this text:

PC-DMIS EXTENTION OBJECT

Importing Objects:

You can import controls into the grid display by using the **Edit | Import Controls** menu option. This option allows you to select and import controls from other dialog boxes, windows, etc. into the grid display within Edit Mode.

To do this:

1. Select **Import Controls** menu option. Your mouse cursor changes.
2. Select the desired control using your mouse. PC-DMIS imports the object into the grid display.

You can only import controls from a Win32 application.

Select Object

The **Select** object allows you to select an object already inserted into your HyperView report. Simply click on the object in the report and PC-DMIS will surround the object with a border and small green squares. This lets you know the object has been selected. Once selected, you can move, resize, or edit the object.

Text Object

The **Text** object allows you to insert textual labels, descriptions, and markings into your HyperView report. This is helpful in making your report more meaningful. You can change an inserted text object's display by selecting the text object, right-clicking on it, and editing its properties.

Common properties used on a text object include:

Property	Description
Text	This is the text displayed in the object.
Orientation	This rotates the text object horizontally or vertically. Be aware that you can only display some fonts horizontally (such as the MS SANS SERIF font).
Alignment	This aligns the text within the object. Available alignments are: Left, Center, and Right.

TextVar Object

The **TextVar** object allows you to insert dynamic, numerical text into your HyperView report. This object differs from the **Text** object by deleting the TEXT property and adding these 2 properties:

Property	Description
ValueEq	Value equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). The value is formatted and displayed at run time. -$, $, CONST
Format	This formats the numerical value using 'C' language standards. However, you should use only the numeric portion. For example, the format %5.2lf becomes 5.2, %5d becomes 5. One exception to this is when using BOOLEAN variables, the $ string gets converted into NO or YES.

You can use the **TextVar** object to easily display page numbers in a multi-page HyperView report. Simply edit the **ValueEq** property so that it's set to display the set variable value of CurPage (for the current page number) or NumPage (for the total number of pages).

Example of Using BASIC Scripting to Dynamically Display Text

You can also use BASIC code in the **VBS Mini-Editor** to dynamically display a value inside a **TextVar** object. For example, suppose you have text for a heading that you want to display only on certain pages of a multi-page report. You could use code similar to that shown here:

1. Create ten page multi-page report by following the instructions given in the "Creating Multiple Page Reports" topic on page 451.
2. Double click on the **TextVar** object to access the **Property Sheet** dialog box.
3. Select the **PC-DMIS** tab.
4. Click on the `ValueEQ` property. The **VBS Mini-Editor** appears much of it unavailable for selection.
5. Select **VBScript** from the list. The text area of the editor becomes available.
6. Add some string variables containing the messages you want to display on different pages:

```
Dim strMessage1 As String

Dim strMessage2 As String

strMessage1 = "HEADING FOR PAGES 1-5"

strMessage2 = "HEADING FOR PAGES 6-10"
```

7. Add conditional code so that depending on which page is the current page the text displayed in the **TextVar** object changes.

```
If curpage <= 5 Then Retn = strMessage1 Else
   Retn = strMessage2
```

   This code changes the text to display the string given to the `strMessage1` variable when the current page is one through five and then to display the string given to the `strMessage2` variable when the current page is six through ten.

8. Click **OK** to close the **VBS Mini-Editor**.
9. Close the **Property Sheet** dialog box.
10. Press CTRL + E to place your report in run mode.
11. Press PAGEUP or PAGEDOWN to cycle through the multiple pages and watch the text change dynamically.

Line Object

The **Line** object allows you to quickly insert a standard line into your HyperView report. You can then right-click on the inserted line and modify its properties.

Common properties used on a line object are:

Property	Description
`PenWidth`	This sets the width of the line in

	pixels.
`PenStyle`	This changes the line style to `Solid`, `Dash`, `Dot`, `Dash-dot`, and `Dash-dot-dot`.

`Arrowhead`	This determines if the line should contain an arrowhead and the direction the arrowhead points. You can choose one of these formats:
	`----(none)`
	`<---`
	`--->`
	`<--->(both)`

`ArrowheadHeight`	This determines the height of the arrowhead in pixels.

Polyline Object

The **Polyline** object allows you to link lines together to construct an area. When you click and create the first line, a second line is automatically started at the end point of the first line. Polyline objects contain the same properties as a standard line object with these additions:

Property	Description
`PenWidth`	This sets the width of the object's border in pixels.
`HatchStyle`	This sets the type of pattern (or hash) to use to fill the resulting polygon. Available patterns are: ▤ Horizontal ▥ Vertical ▧ Diagonal ▨ Reverse Diagonal ▦ Cross ▩ Diagonal Cross
`ShadowStyle`	This sets the location of the 3D shadow on the resulting polygon. Available locations are: `Top/Left`, `Bottom/Right`, or `None`.

Border Object

The **Border** object inserts a rectangular border into the HyperView report. Along with the ability to resize the object, add a fill color, add a border color, and change other attributes, the following editable properties are included for border objects:

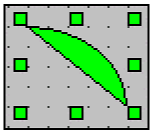
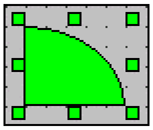
Property	Description
`PenWidth`	This sets the width of the object's border in pixels.
`HatchStyle`	This sets the type of pattern (or hash) to use to fill the resulting polygon. Available patterns are: ▤ Horizontal ▥ Vertical ▧ Diagonal ▨ Reverse Diagonal ▦ Cross ▩ Diagonal Cross
`ShadowStyle`	This sets the location of the 3D shadow on the resulting polygon. Available locations are: `Top/Left`, `Bottom/Right`, or `None`.
`HiliteColor`	This sets the color used for the `BorderStyle` property.
`BorderStyle`	This sets the 3D highlight style of the object's border. Options include `Normal`, `3D`, `Sunken` and `Raised`.

Arc Object

The **Arc** object inserts an elliptical arc into the HyperView report. A default arc doesn't have a fill color and has a beginning angle of 0 degrees and an ending angle of -90 degrees.

Along with the ability to resize the object, add a fill color, add a border color, and change other attributes, the following editable properties are included for arc objects:

Property	Description

`PenWidth`	This sets the border width in pixels.
`PenStyle`	This changes the arc's border style to `Solid`, `Dash`, `Dot`, `Dash-dot`, and `Dash-dot-dot`.
`Angle1`	This number sets the start angle for the arc. The default is 0 degrees.
`Angle2`	This number sets the end angle for the arc. The default is -90 degrees.
`IsWedge`	If set to `NO` (the default), PC-DMIS draws a filled arc without a wedge, like this:  If set to YES, PC-DMIS draws a filled arc with a wedge like this: 

Ellipse Object

The **Ellipse** object inserts an ellipse into the HyperView report. The default ellipse is a circle with the same distance from the centroid of the circle to all points along the circumference; however you can easily stretch the circle to form an ellipse.

Along with the ability to resize the object, add a fill color, add a border color, and change other attributes, the following editable properties are included for ellipse objects:

Property	Description
`Width`	This sets the width of the object's border in pixels.
`PenStyle`	This changes the line style to `Solid`, `Dash`, `Dot`, `Dash-dot`, and `Dash-dot-dot`.
`HatchStyle`	This sets the type of pattern (or hash) to use to fill the resulting polygon. Available patterns are: ▤ Horizontal ▥ Vertical ▨ Diagonal

	▨ Reverse Diagonal ⊞ Cross ▩ Diagonal Cross

Frame Object

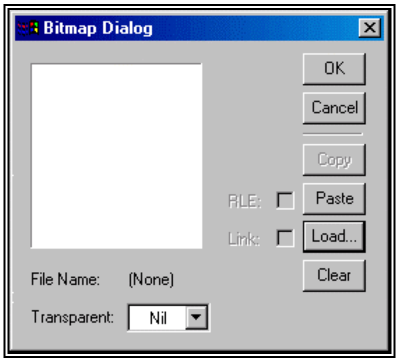
The **Frame** object inserts a frame into the HyperView report. You'll generally use this object to set off groups of similar options when designing your own dialog boxes inside your HyperView report.

Along with the ability to resize the object, add a fill color, add a border color, and change other attributes, the following editable properties are included for frame objects:

Property	Description
`BorderColor`	This sets the color of the border.
`HiliteColor`	This sets the color used for the `BorderStyle` property when you use `Raised` or `Sunken`.
`BorderStyle`	This sets the 3D highlight style of the object's border. Options include `Normal`, `3D`, `Sunken` and `Raised`.
`BorderWidth`	This sets the width of the object's border in pixels.
`Text`	Text typed here determines the name used for the frame's title.
`Alignment`	This aligns the text within the object. Available alignments are: `Left`, `Center`, and `Right`.

Bitmap Object

The **Bitmap Object** allows you to insert a pre-created bitmap graphic into your HyperView report. When you click this icon, a dialog box appears.



File Name: (None)

Transparent: Nil

Bitmap Dialog

- The **OK** button allows you to insert a bitmap loaded into the dialog box into the HyperView report.
- The **Cancel** button allows you to close the dialog box without inserting any bitmap.
- The **Copy** button allows you to copy a bitmap to the Clipboard that you have already loaded into this dialog box through the **Load** button.
- The **Paste** button allows you to paste a copied bitmap graphic from the Clipboard into the dialog box. Selecting the **RLE** check box compresses the graphic when it's saved with the report.
- The **Load** button allows you to load a bitmap into the dialog box. Selecting the **Link** check box also links the bitmap to the report by its directory path.
- The **Clear** button allows you to clear the loaded bitmap out of the dialog box.
- The **Transparent** list allows you to specify one color within the bitmap that will be made transparent.

Along with the ability to resize the object and change other attributes, the following editable properties are included for bitmap objects:

Property	Description
`Bitmap`	This contains the file name for the bitmap.
`Layout`	`Center, Left,` or `Right` is the alignment of the bitmap within the rectangle. `Stretch` expands or shrinks the graphic to fit the size of the box. `Size to Fit` automatically expands the rectangle size to fit the size of the graphic.

Pointer Object

The **Pointer** object inserts a dynamic, movable, colored pointer into the HyperView report. The color defaults to yellow.

Along with the ability to resize the object, change colors, and change other attributes, you can cause this object to move by using these properties in conjunction with Visual BASIC code:

Property	Description
`ValueEq`	Value equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). `-$, $, CONST`
`MinimumEq`	Minimum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). `-$, $, CONST`
MaximumEq	Maximum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). -$, $, CONST

Tip: If you change the direction of the pointer, the movement of the arrow will also change to the direction specified.

Example of Using BASIC Scripting to Dynamically Move the Pointer

You can also use BASIC code in the **VBS Mini-Editor** to dynamically move the pointer. Suppose you want to create a report that moves a pointer along an axis as you progress through a multi page report. You could use code similar to that shown in this example:

1. Create a ten page multi-page report by following the instructions given in the "Creating Multiple Page Reports" topic on page 451.
2. Insert a **Pointer** object into the report.
3. Select the pointer and drag its box to increase its vertical distance. Notice that when you elongate the box vertically, the size of the pointer doesn't change, just the total distance that can be traveled by the pointer.
4. Double click on the **Pointer** object to access the **Property Sheet** dialog box.
5. Select the **PC-DMIS** tab.
6. Click on the `ValueEQ` property. The **VBS Mini-Editor** appears much of it unavailable for selection.
7. Select **VBScript** from the list. The text area of the editor becomes available.
8. Add the following code to cause the pointer to move based on a user input with each new page on a multi-page report:

```
intValue =  ("Type a value. The pointer will
   move this percentage amount of the total
   distance.")
Retn = intValue
```

9. Click **OK** to close the **VBS Mini-Editor**.
10. Close the **Property Sheet** dialog box.
11. Press CTRL + E to place your report in run mode.
12. Press PAGEUP or PAGEDOWN to cycle through the multiple pages and watch the pointer move dynamically based on your input.

Gauge Object

The **Gauge** object inserts a gauge object (or progress bar) into the HyperView report.

Along with the ability to resize the object, change colors, and change other attributes, you can cause the object's percentile bar to move by using these properties in conjunction with Visual BASIC code:

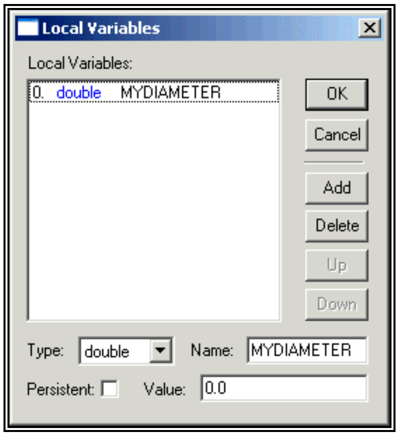
Property	Description
`ValueEq`	Value equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). `-$, $, CONST`
`MinimumEq`	Minimum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). `-$, $, CONST`
MaximumEq	Maximum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). -$, $, CONST

Tip: If you change the direction of the gauge, the progress of the gauge will also change to the direction specified.

Example of Using BASIC Scripting to Dynamically Change the Gauge's Value

You can also use BASIC code in the **VBS Mini-Editor** to dynamically change the value of the gauge at a given time. Suppose you want to create a report that moves the gauge from 0 to 100 percent based on how closely a circle's diameter approaches its minimum and maximum allowed tolerances. You can do this by following this procedure:

1. Create a new HyperView report and make sure it's in Edit mode. First, you will need to create a global variable named `MYDIAMETER` that will take the diameter value of the circle.
2. Right-click on the grid and make sure **The Frame / The View** is selected from the object list on the property sheet.
3. Click on the **Settings** tab on the property sheet and then select the value for the **LocalVariables** property. The **Local Variables** dialog box appears.
4. Click the **Add** button to add a new variable. Select **Double** from the **Type** list. In the **Name** box type MYDIAMETER. Leave the **Value** as 0.



Local Variables:

0. double MYDIAMETER

Type: double

Name: MYDIAMETER

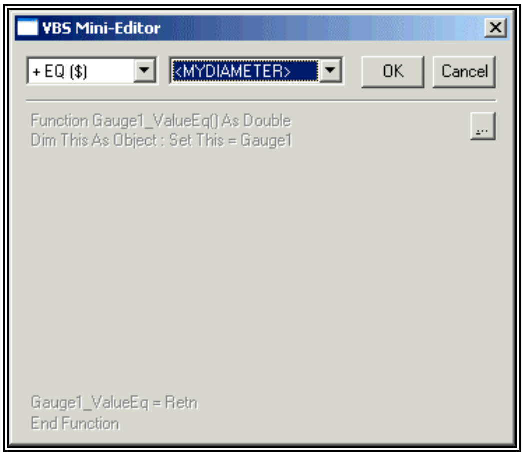
Persistent: □

Value: 0.0

Local Variables dialog box showing the MYDIAMETER global variable

5. Click **OK** to return to the **Property Sheet**. The MYDIAMETER variable is now added to the HyperView report.
6. Next, insert a **Gauge** object into the report.
7. Select the **Gauge** object and place and size it as desired. The vertical distance is the total distance the percentile bar can travel.
8. Double click on the **Gauge** object to access the **Property Sheet** dialog box.
9. Select the **PC-DMIS** tab.
10. Click on the `MaximumEq` property. The **VBS Mini-Editor** appears; much of it is unavailable for selection.
11. Select **CONST** from the list. This value allows you to define the gauge's 100% value.
12. Type the maximum allowed value for the circle's diameter. This example measures a one inch diameter circle with an allowed tolerance of .010. So the maximum allowed value would be **1.010**.
13. Click **OK** to return to the property sheet.
14. Click on the `MinimumEq` property. The **VBS Mini-Editor** re-appears. This value allows you to define the gauge's 0% value.
15. Select **CONST** from the list.

16. Type the minimum allowed value for the circle's diameter. Again, since this example measures a one inch diameter circle with an allowed tolerance of .010, the minimum allowed value would be **0.990**.
17. Click **OK** to return to the property sheet. Now you need to give the actual value of the gauge the diameter of the circle.
18. With the gauge object still selected from the property sheet, click the **PC-DMIS** tab and then select the **ValueEq** property. The **VBS Mini-Editor** appears; much of it is unavailable for selection.
19. In the list on the left, select **+ EQ ($)**.
20. In the list on the right, select the **<MYDIAMETER>**.



```
Function Gauge1_ValueEq() As Double
Dim This As Object : Set This = Gauge1


Gauge1_ValueEq = Retn
End Function
```

Using the VBS Mini-Editor to set the value of the gauge object

21. Click **OK** to close the **VBS Mini-Editor**.
22. Close the property sheet and save the report. This example gives the report the name of *gaugetest.rpt*.
23. You will now create a generic circle feature inside PC-DMIS.
24. Select PC-DMIS's Edit window and place it in Command mode.
25. Insert a generic feature by typing `GENERIC` and pressing TAB. Initially, a generic point feature appears in the Edit window with the feature's ID selected.
26. Press F9 to access the dialog box for this generic feature.
27. In the **Feature Type** area, select **Circle**.
28. In the **Data Type** area, select **Nominal Values**.
29. Type a name for the circle in the **Feature Name** box. This example uses CIR1 for the feature name.
30. Type the XYZ and IJK values for CIR1.
31. Select the **Diameter** option and give CIR1 a nominal diameter value of 1.
32. Change any other options as desired and then click **OK** when finished. The generic circle in your Edit window should look something like this:

```
CIR1      =GENERIC/CIRCLE,DEPENDENT,RECT,OUT,$
```

```
            NOM/XYZ,1,1,0.95,$
            MEAS/XYZ,1,1,1,$
            NOM/IJK,0,0,1,$
            MEAS/IJK,0,0,1,$
            DIAMETER/1,0
```

33. More your cursor *before* the CIR1 feature and insert an input comment that will take a measured diameter value (since this example is done in offline mode, the "measured" values are inserted manually). For example:

```
C1        =COMMENT/INPUT,Please type CIR1's
   measured diameter:
```

34. Now go to the command block for CIR1 and on the last line where it says DIAMETER/1,0 change the second parameter, the measured diameter of 0, to C1.INPUT. This assigns the value from the input comment to CIR1's measured diameter.

```
            DIAMETER/1,C1.INPUT
```

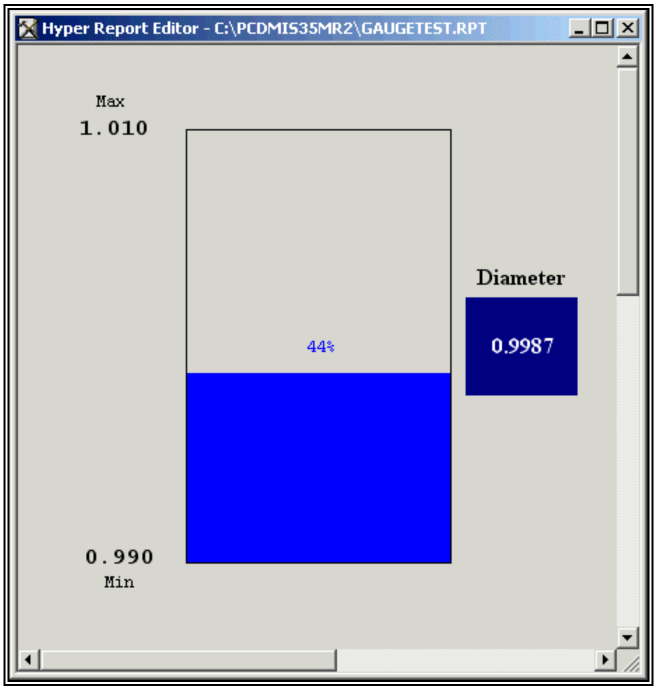
35. Now move the cursor *after* the CIR1 feature and embed the saved HyperView report by selecting the **Hyper-Report** menu option.

36. Now in the REPORT/FILENAME command where it says PARAM/=, place your cursor immediately before the equals sign and type `MYDIAMETER`. Then move the cursor immediately after the equals sign and type `CIR1.DIAMETER`. The code to embed your HyperView report should look something like this:

```
CS1       =REPORT/FILENAME=
   C:\PCDMIS35\GAUGETEST.RPT, AUTOPRINT=NO
            PARAM/MYDIAMETER=CIR1.DIAMETER
            PARAM/=
            ENDREPORT/
```

37. Mark the newly added commands and then execute the part program. The diameter for CIR1 gets passed as a parameter into the HyperView report and the gauge object's percentile dynamically changes depending on the circle's measured diameter.

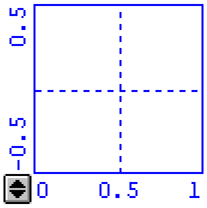
For example, if the measured diameter for CIR1 was .9987 you would get a report that looks something like this:



Sample HyperView Report Showing the Gauge Object Dynamically Linked to a Feature's Measured Diameter

Graph Object

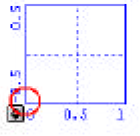
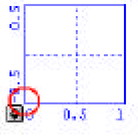
The **Graph** object inserts a graph into your HyperView report with the ability to dynamically update its data. The **Graph** object allows you to store and then display an array of data points. You can use X and Y values to specify the points of data on the graph.

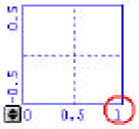
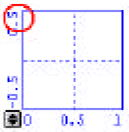


Default Graph object

Along with the ability to resize the object, change colors, and change other attributes, you can cause graph objects to update their data dynamically by using these properties in conjunction with Visual BASIC code:

Property	Description
`DataPointColor`	This changes the colors used on the X, Y data points and on any lines drawn between points.
`Width`	This changes the diameter (in pixels) of the data point.
`ConnectPoints`	Setting this to YES draws lines

	between the data points on the graph, visually connecting them together.
`Clockwise`	Setting this to `NO` reverses the Y axis, making the top negative and the bottom positive.
`Xdivisions`	This divides the X axis of the graph, essentially adding the specified number of columns.
`Ydivisions`	This divides the Y axis of the graph, essentially adding the specified number of rows.
`EnablePoint`	If set to `1`, your graph accepts the addition of new data points. If set to `0`, your graph rejects the addition of new data points.
`ClearAllPoint`	If set to `1`, all data points are cleared from the graph the next time your HyperView report enters Run mode. If set to `0`, all data points remain in the graph.
`XpointEq`	This is the X value for your data point on the graph. You can use a simple equation, or evaluate the data from a VBScript.
`YpointEq`	This is the Y value for your data point on the graph. You can use a simple equation, or evaluate the data from a VBScript.
`XminEq`	This is the X axis minimum equation. The value used determines the beginning value making up the X axis.  You can use a simple equation, or evaluate the data from a VBScript.
`YminEq`	This is the Y axis minimum equation. The value used determines the beginning value making up the Y axis.  You can use a simple equation, or evaluate the data from a VBScript.
`XmaxEq`	This is the X axis maximum equation. The value used determines the ending value making up the X axis.

	 You can use a simple equation, or evaluate the data from a VBScript.
`YmaxEq`	This is the Y axis maximum equation. The value used determines the ending value making up the Y axis.  You can use a simple equation, or evaluate the data from a VBScript.

Tip: You can re-scale the Y axis by clicking on the up and down arrows ⬍ near the graph's origin.

Example of Using BASIC Scripting to Dynamically Plot Points on the Graph

You can also use BASIC code in the **VBS Mini-Editor** to dynamically plot points on a graph object. Suppose you want to create a report that plots points based on user input. You could use code similar to that shown in this example:

Set Up the Graph

1. Create a ten multi-page report by following the instructions given in the "Creating Multiple Page Reports" topic on page 451.
2. Insert a **Graph** object into the report.
3. Double click on the **Graph** object to access the **Property Sheet** dialog box.
4. Select the **PC-DMIS** tab.
5. Click on the `XmaxEq` property. The **VBS Mini-Editor** appears.
6. Select **CONST** from the list and type **100**. This sets the upper end of the X range on the graph.
7. Click **OK** to return to the **Property Sheet**.
8. Click on the `YmaxEq` property.
9. Select **CONST** from the list and type **100**. This sets the upper end of the Y range on the graph.
10. Click **OK** to return to the **Property Sheet**.
11. Modify any other properties to customize the look and feel of the graph.
12. Close the **Property Sheet** dialog box.

Declare Global Variables

13. Declare the following global variables by following the instructions in the "Declare Global Variables Using the LocalVariables Property" on page 404.

    Xvalue

    Yvalue

Add a Button to Send User Input to the Global Variables

14. Add a **Button** object to the report.
15. Click on and change its label to read **Plot a Point**.
16. Access the **Property Sheet** dialog box for the newly added button object, and select the `EventClick` event. The **VBS Mini-Editor** appears.
17. Insert this code to get input from the user to plot the point:

    ```
    Xvalue = InputBox ("Type the X value:")

    Yvalue = InputBox ("Type the Y value:")
    ```

18. Click **OK** to accept you code and return to the **Property Sheet** dialog box.
19. Close the **Property Sheet** dialog box.

Edit the Graph to Use the Global Variables

20. Click on the `XpointEq` property. The **VBS Mini-Editor** appears.
21. Select **+EQ($)** from the list. Another list appears. Select **<Xvalue>** from the list.
22. Click **OK** to return to the **Property Sheet**.
23. Click on the `YpointEq` property. The **VBS Mini-Editor** appears.
24. Select **+EQ($)** from the list. Another list appears. Select **<Yvalue>** from the list.
25. Click **OK** to return to the **Property Sheet** dialog box.
26. Close the **Property Sheet** dialog box.

Test the Report

27. Press CTRL + E to place your report in run mode.
28. Press PAGEUP or PAGEDOWN to get the mini toolbar that allows you to cycle through multiple page reports to appear.
29. Click the **Plot a Point** button
30. Inserting the X and Y values.
31. Click **Refresh** from the mini toolbar and watch the point get dynamically plotted at the correct XY location.

Button Object

The **Button** object inserts a button control into the HyperView report. You can choose the action the report will take when the button is clicked by modifying the `ButtonType` property.

Along with the ability to resize the object, change colors, and change other attributes, you can further customize button objects by using these properties:

Property	Description
`Default`	This determines how buttons will respond when you press the ENTER key. `YES` - The button responds to a pressed ENTER key even if another button has the focus. `NO` - If ENTER is pressed only the button that has the focus will respond.
`ButtonShape`	This determines the look of the button. Options include: `0- Normal:` changes shape to a standard button. `1- Property tab Inactive:` changes shape to an unselected property sheet tab. `2- Property tab Active:` changes shape to a selected property sheet tab.
`Text`	This determines the text that appears on the button.
`Bitmap`	This places a bitmap of your choice on top of the button face. It uses the same dialog box and properties as described in the "Bitmap Object" on page 412 topic.

`Layout`	This determines the position of the bitmap or text on the button's face. The values include:
	`0- Center`
	`1- Left`
	`2- Right`
	`3- Top`
	`4- Bottom`

`ButtonType`	This determines the action a button will take when it's clicked. `0- Cancel: c`loses the report and does nothing. `1- EventClick: e`xecutes a C++ or VBScript action. `2- Goto: g`oes to the Multi Document Interface (MDI) associated file. `3- Help: i`nvokes the WinHelp() function using the

	HelpContextID property to open the proper topic. 4- OK: records changes and closes the report. 5- Record: records changes and keeps the report open.
GotoPath	This determines the MDI associated file the report will go to when the button is clicked. This is mostly used to implement property sheet-like controls.

Checkbutton Object

☑

The **Checkbutton** object inserts a checkbutton object, or check box, into the HyperView report. Check boxes, unlike option buttons, are never mutually exclusive. You can select as many check boxes as desired (provided they are enabled for selection).

Along with the ability to resize the object, change colors, and change other attributes, you can further customize check boxes by using these properties:

Property	Description
OwnerDrawn	This specifies how the report should draw the check box. If set to TRUE, the check box gets drawn from a bitmap specified in the Bitmap and Bitmap (Off) properties. If set to FALSE, the check box is drawn normally.
Bitmap	This allows you to specify a bitmap to use if OwnerDrawn is set to TRUE and if the check box is selected. Additionally, the specified bitmap must be the same shape as a standard check box.
BitmapOffState	This allows you to specify a bitmap to use if OwnerDrawn is set to TRUE and if the check box is *not* selected. Additionally, the specified bitmap must be the same shape as a standard check box.
TriState	Instead of just the TRUE (ON) and FALSE (OFF) states, this property lets your check box have three states: true, false, and partially true, thus allowing you determine whether *a portion* of a selected option is true.
AlignTextLeft	If set to YES, this moves the text to

	the left of the check box like this:
	`Text Here [  ]`
	If set to `NO`, this moves the text to the right of the check box like this:
	`[  ] Text Here`

`Text`	The text specified here gets displayed on the check box control.
`HelpHotButton`	Selecting YES shows a question mark help button next to the check box. ? Clicking this button in Run mode launches the PC-DMIS Online Help.
`ValueID`	This contains the `CurPage` and `NumPages` variable. You get and set this property using the control.

Radiobutton Object

◉

The **Radiobutton** object inserts option buttons into the HyperView report. Option buttons are mutually exclusive, you can only have one option button selected inside your report at a time. The `ListItems` property allows you to define a set of option buttons.

Along with the ability to resize the object, change colors, and change other attributes, you can further customize this object by using these properties:

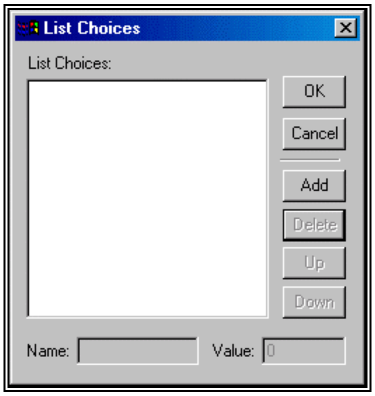
Property	Description
`Bitmap`	This allows you to specify a bitmap to use for a selected option button. The specified bitmap must be the same shape as a standard check box.
`BitmapOffState`	This allows you to specify a bitmap to use for a deselected option The specified bitmap must be the same shape as a standard option button.

`AlignTextLeft`	If set to YES, this moves the text to the left of the option button like this:
	`Text Here ( )`
	If set to `NO`, this moves the text to the right of the option button like this:
	`( ) Text Here`

`ListItems`	This defines a list of option buttons and its associated values (see the description on the **List Choices** dialog box below).
`ValueID`	This contains the `CurPage` and `NumPages` variable. You get and set

	this property using the control.

The `ListItems` property displays the **List Choices** dialog box.



List Choices dialog box

This dialog box allows you to add, rename, and delete option buttons and assign numerical values. Generally, when adding a new option the value for that option increments automatically, so only one button is selected at a time. However, you can change the numerical values to allow the selection of groups of options with one mouse click when in Run mode.

For example, suppose you have five option buttons that you've named Options A through E, and you change their assigned values to the following:

Option buttons	Value
Option A	0
Option B	0
Option C	1
Option D	2
Option E	2

When you enter Run mode and select **Option A**, both **Option A** and **Option B** will be selected. This is because they're assigned the same value. The same is also true if you select **Option E**, both **Option D** and **Option E** will be selected. Only **Option C** has a unique value, so when you select it, only it is selected.

ComboBox Object

The **ComboBox** object inserts a combo list into your HyperView report. Using this list you can display options to select when you click the list in Run mode.

Along with the ability to resize the object, change colors, and change other attributes, you can further customize this object by using these properties:

Property	Description
`Accelerator`	This allows you to define the letter to use in conjunction with the ALT key to create an accelerator for the control.

	The letter or symbol defined should be unique from those used for other accelerators (including the accelerators on the menu bar).
`UseColors`	This determines whether or not this control should use colors specified in `BackColor` and `TextColor` properties.
`ListItems`	This defines the list of items and the associated values (see the description of the **List Choices** dialog box described in the "Radiobutton Object" on page 425).
`NumDropped`	The number specified determines how many list items the report displays when you click on the list in Run mode.
`Sort`	Selecting YES sorts the list items in alphabetic order.
`ComboType`	This specifies the type of combo list. You can select either `Droplist` or `Dropdown`.
`HelpHotButton`	Selecting YES, shows a question mark help button next to the check box. ? Clicking this help button in Run mode launches the PC-DMIS Online Help.
`ValueID`	This contains the `CurPage` and `NumPages` variable. You get and set these variables using this control.

ListBox Object

The **ListBox** object inserts an open list inside the HyperView report.

Along with the ability to resize the object, change colors, and change other attributes, you can further customize this object by using these properties:

Property	Description
`Accelerator`	This allows you to define the letter to use in conjunction with the ALT key to create an accelerator for the control. The letter or symbol defined should be unique from those used for other accelerators (including the accelerators on the menu bar).
`UseColors`	This determines whether or not the control should use the colors specified in the `BackColor` and

	`TextColor` properties.
`Sort`	Selecting YES sorts the list items alphabetically.
`HorizontalScroll`	Selecting `Always` places a horizontal scroll bar on the list box even if the text for a list item isn't long enough to cause it to scroll. Selecting `None` removes the scroll bar.
`VerticalScroll`	Selecting `Always` places a vertical scroll bar on the list box even if there aren't enough list items to cause it to scroll. Selecting `None` removes the scroll bar.
`UseTabstops`	This determines whether or not the list box should use tab stops. If set to YES, pressing TAB in Run mode will eventually make this list box the active control.
`WantKeyInput`	This determines whether or not the list box should accept keyboard input.
`RemoveSelection`	Selecting YES removes the selection from the list box (`ListID`) when recorded.
`ListID`	This is the ID of the list. You can set or get this by using the control. It should be in this format: `item 1\r\nitem 2\r\nitem 3` and so on.
`ValueID`	This is the ID of the value. You can set or get this by using the control.

EditBox Object

The **EditBox** object inserts an editable box into the HyperView report. You can use this object with Visual BASIC scripting to collect information from those running the report.

Along with the ability to resize the object, change colors, and change other attributes, you can further customize this object by using these properties:

Property	Description
`Accelerator`	This allows you to define the letter to use in conjunction with the ALT key to create an accelerator for the control. The letter or symbol defined should be unique from those used for other accelerators (including the

	accelerators on the menu bar).
`Alignment`	This aligns the text within the object. Available alignments are: `Left`, `Center`, and `Right`.
`ReadOnly`	Setting this to YES makes the text read-only and therefore not editable when the report is in Run mode.
`BorderDrawn`	Setting this to YES draws a border around the control.
`CaseOrPassword`	This sets the style of the text in the box. You can select one of the following: `0 – None:` doesn't change the display of the text. `1 - Lower Case:` displays the case of the text in the box to all lower case. `2 - Upper Case:` displays the case of the text in the box to all upper case. `3 – Password:` masks each character of the text with an asterisk.
`UseColors`	If set to YES then colors specified in `BackColor` and `TextColor` are displayed.
`HelpHotButton`	Selecting YES, shows a question mark help button next to the check box. ? Clicking this help button in Run mode launches the PC-DMIS Online Help.
`ValueID`	This contains the `CurPage` and `NumPages` variables. You get and set these variables using this control.

EditLongBox Object

The **EditLongBox** object inserts an edit box in the HyperView report that only accepts Long numerical values. This has all the properties of the **EditBox** object (see "EditBox Object" on page 428) with the addition of `MinimumEq` and `MaximumEq` properties:

Property	Description
`MinimumEq`	Minimum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). `-$, $, CONST`

MaximumEq	Maximum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). `-$, $, CONST`

EditDbleBox Object

The **EditDblBox** object inserts an edit box in the HyperView report that accepts Hyper variables of double types. This has all the properties of the **EditBox** object (see "EditBox Object" on page 428) with the addition of `MinimumEq` and `MaximumEq` properties:

Property	Description
`MinimumEq`	Minimum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). `-$, $, CONST`
MaximumEq	Maximum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). `-$, $, CONST`

MultiEditBox Object

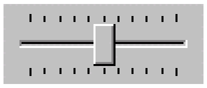
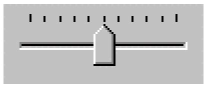
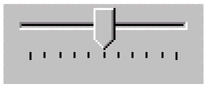
The **MultiEditBox** object inserts an edit box that supports more than one line of text. This has all the properties of the **EditBox** object (see "EditBox Object" on page 428), plus the following:

Property	Description
`HorizontalScroll`	Selecting `Always` places a horizontal scroll bar on the control even if the text isn't long enough to cause it to scroll. Selecting `None` removes the scroll bar.
`VerticalScroll`	Selecting `Always` places a vertical scroll bar on the control even if the text isn't long enough to cause it to scroll. Selecting `Automatic` places a vertical scroll bar only when the amount of text exceeds the vertical height of the box. Selecting `None` removes the scroll bar.

Slider Object

The **Slider** object inserts a slider into your HyperView report. With a slider in your report you can allow users to set a value from a range of continuous possible values.

Along with the ability to resize the object, and change other attributes, you can further customize this object by using these properties:

Property	Description
`Accelerator`	This allows you to define the letter to use in conjunction with the ALT key to create an accelerator for the control. The letter or symbol defined should be unique from those used for other accelerators (including the accelerators on the menu bar).
`Orientation`	This sets the orientation of the slider. Selecting `Horizontal` sets the slider to move horizontally, side to side. Selecting `Vertical` sets the slider to move vertically, up and down.
`TickStyle`	This allows you to set how ticks are displayed on a slider. `0 – Both:` sets the ticks to above and below the slider (if a horizontal slider) or to the right and left sides (if a vertical slider).  `1 - Top/Left:` sets the ticks above the slider (if a horizontal slider) or to the left (if a vertical slider).  `2 - Bottom/Right:` sets the ticks below the slider (if a horizontal slider) or to the right (if a vertical slider). 
`AutoTicks`	Setting this to YES displays ticks on the slider. A tick is a short vertical/horizontal line that is placed at intervals (set by the `TickFrequency` property) along

	the range of the slider.
TickFrequency	This value determines the frequency that ticks are displayed. You can set a value between 2 and 200 ticks equally spaced along the range of the slider.
HelpHotButton	Selecting YES, shows a question mark help button next to the slider. Clicking this help button in Run mode launches the PC-DMIS Online Help.
ValueID	This is the ID of the value. You can set or get this by using the control.
MinimumEq	Minimum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). -$, $, CONST
MaximumEq	Maximum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). -$, $, CONST
DisplayID	This determines whether or not the DisplayID gets updated when the slider gets adjusted.

Spinner Object

The **Spinner** object inserts a spinner control into your HyperView report. You can set this control to work with an edit box, allowing you to select a numerical value by cycling through a range of numbers. The range of numbers is set using the MinimumEQ and MaximumEQ properties.

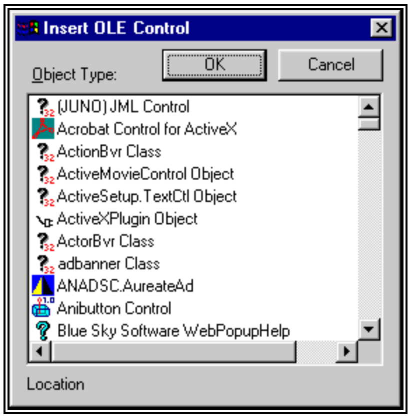
Along with the ability to resize the object vertically, and change other attributes, you can further customize this object by using these properties:

Property	Description
Accelerator	This allows you to define the letter to use in conjunction with the ALT key to create an accelerator for the control. The letter or symbol defined should be unique from those used for other accelerators (including the accelerators on the menu bar).
Orientation	This sets the orientation of the spinner control. Selecting Horizontal rotates the orientation of the spinner arrows to

	point horizontally. Selecting `Vertical` rotates the orientation of the spinner arrows to point vertically.
`WrapAround`	If incrementing or decrementing the list of values and you come to the minimum or maximum value, you can set this property to `YES`, and your report will wrap to the beginning or end of the list.
`EditBuddy`	Selecting `YES`, associates the spinner to the edit control immediately preceding the spinner in the tab order. You can view the tab order by selecting an object in Edit mode and pressing SHIFT + ENTER.
`DecimalBase`	This sets the base of the spinner to either decimal based (select `YES`) or hexadecimal based (select `NO`).
`IncrAccelerator`	This value determines the amount to increment or decrement the spinner each time you click an arrow. The default is `1`.
`ValueID`	This contains the `CurPage` and `NumPages` variable. You get and set these variables using this control.
`MinimumEq`	Minimum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). `-$, $, CONST`
`MaximumEq`	Maximum equation (This value can be a set constant number, a value evaluated from a simple equation, or a returned value from a VBScript). `-$, $, CONST`

ActiveX Object

The **ActiveX** Object inserts an ActiveX control into your HyperView report. Once inserted, a dialog box titled **Insert OLE Control** will open.



Insert OLE Control dialog box

This dialog allows you to select the type of control to be inserted from a list of known controls on your computer. The objects in the **OLE Control** list are unique to each computer, based on the programs installed, etc.

If you click the **Cancel** button, nothing is added and the dialog box closes.

Accessing an ActiveX Object's Methods and Properties from Basic

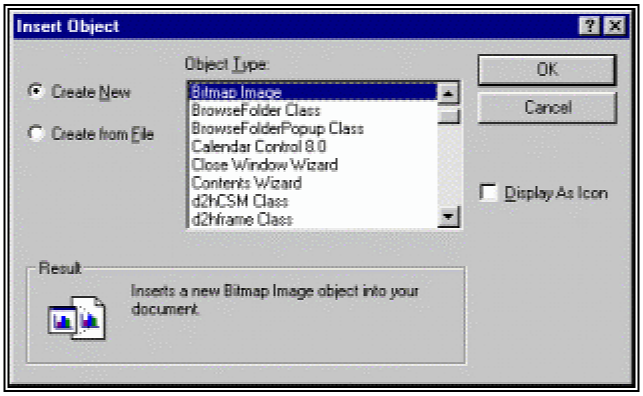
If you have added a control, for example, an ActiveX calendar control and called it MSCAL, you can access its events and properties through the property sheet just like any other object. In this case, since it is an ActiveX object, a new tab will be visible called **ActiveX**. This tab contains a user accessible list of variables and properties with their default values.

To set properties for the ActiveX control inside the **VBS Mini-Editor** for an event handler, type the name of the control, in this case, "`MSCAL.`" (with the period). A window appears allowing the variables and methods to be accessible. Select the "`X`" from the list. Another popup window will appear listing a set of internal editable variables for the ActiveX control.

OLE Object

The **OLE** object embeds or links an object from another application into your HyperView report. For example, using this object you can insert an editable Microsoft Excel file into your report.

Once you insert an **OLE** object, a dialog box titled **Insert Object** appears.



Insert Object

● Create New

○ Create from File

Object Type:

- Bitmap Image
- BrowseFolder Class
- BrowseFolderPopup Class
- Calendar Control 8.0
- Close Window Wizard
- Contents Wizard
- d2hCSM Class
- d2hframe Class

□ Display As Icon

Result

Inserts a new Bitmap Image object into your document.

Insert Object dialog box

This dialog box allows you to select the type of OLE object from a list of objects that are currently available on your computer. The OLE objects in the list are unique to your individual computer depending on what is installed, etc.

If you select the **Cancel** button nothing is added to the HyperView report and the dialog closes.

Accessing an OLE Object's Automation Methods and Properties from Basic

If you have added an OLE control, for example the Bitmap control, and called it `BITMAP`, you may wish to access the OLE object's automation methods and properties from BASIC. For example, you might wish to add program code to change the bitmap being displayed, populate an embedded Microsoft Excel spreadsheet object with data, or set the text in an embedded Microsoft Word document. In order to access the OLE automation methods and properties for the object named "`BITMAP`", use the following code:

```
AttachOLE("BITMAP_X")
```

The object variable BITMAP_X may now be used as follows:

```
BITMAP_X.{property} = {value}
```

or

```
BITMAP_X.{automation method call}.
```

Note: You must have documentation for the particular OLE object you are using to know what methods and properties are available.

Cad Object

PC-DMIS EXTENTION OBJECT

The CAD object is used to insert a graphical view into a report. When a CAD object is initially dropped into a report, it is created with a single pane and grabs the current active graphics and viewset from PC-DMIS's Graphics Display window for the part program.

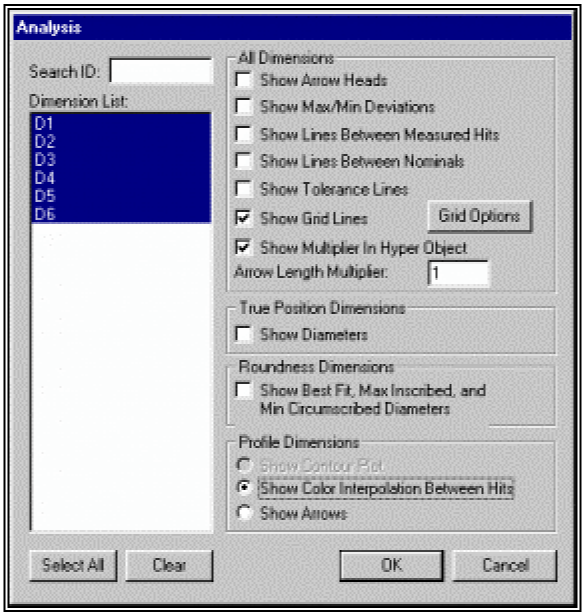
You can "activate" the CAD object by double clicking within its boundaries. When activated, the frame around the CAD object will change, and the translate, rotate 2D, rotate 3D, edit graphics, edit CAD, and viewset functions are all available from PC-DMIS'S toolbar. Once activated, clicking anywhere in the HyperView report (other than on the CAD object itself) will deactivate the CAD object. OLE objects behave in a similar manner, as described in the "ActiveX Object" topic on page 433 and the "OLE Object" topic on page 434. When deactivated, the CAD object is simply a bitmap image of the graphics, and isn't "live" as it is when activated. The CAD object may be activated in both Edit and Run Modes. If the viewset of the CAD object is altered while in Run Mode, it is automatically saved when the report is closed or switched back to Edit Mode. The CAD object doesn't add any additional properties to the **PC-DMIS** tab on the **Property Sheet**.

Analysis Object

PC-DMIS EXTENTION OBJECT

The analysis object is used to insert a PC-DMIS analysis window into a report. The analysis object behaves in a similar way as the CAD object in the sense that you can "activate" or "deactivate" the analysis object in the same way as you do the Cad Object on page 435. When an analysis object is initially dropped into a report, a dialog will appear allowing the user to select which dimensions to graphically analyze. Once created, this dialog is available by doing the following:

1. Open the Property Sheet for the analysis object.
2. Click on the **PC-DMIS** tab.
3. Double click on the "(Settings Dialog)" property entry. The **Dimension Analysis Options** dialog boxappears.



Analysis

Search ID:

Dimension List:

D1
D2
D3
D4
D5
D6

All Dimensions

☐ Show Arrow Heads
☐ Show Max/Min Deviations
☐ Show Lines Between Measured Hits
☐ Show Lines Between Nominals
☐ Show Tolerance Lines
☑ Show Grid Lines
☑ Show Multiplier In Hyper Object
Arrow Length Multiplier: 1

True Position Dimensions

☐ Show Diameters

Roundness Dimensions

☐ Show Best Fit, Max Inscribed, and Min Circumscribed Diameters

Profile Dimensions

○ Show Contour Plot
● Show Color Interpolation Between Hits
○ Show Arrows

Dimension Analysis Options dialog box

The **Analysis**dialog box allows you to determine the information that gets displayed in the Analysis window.

To access the available options:

1. Select the dimension from the list box on the left side of the dialog box. Available options for that particular dimension type will become selectable.
2. Select the appropriate options for the dimension.

The following options are available in this dialog box:

Show Arrow Heads

The **Show Arrow Heads** check box toggles the arrow heads on or off of the deviation lines.

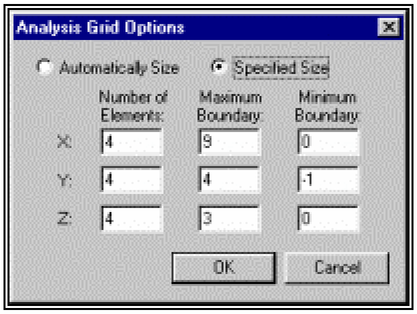
Show Max/Min Deviations

The **Show Max/Min Deviations** check box displays a text box with'*+' at the position for the maximum deviation of a dimension, and a text box with '*-' at the position for the minimum deviation of a dimension.

Show Grid Lines

The **Show Grid Lines** check box allows grid lines to be made available in the Analysis window. When this check box is selected the **Grid Options** button appears.

Grid Options



Analysis Grid Options

○ Automatically Size ● Specified Size

	Number of Elements:	Maximum Boundary:	Minimum Boundary:
X:	4	9	0
Y:	4	4	-1
Z:	4	3	0

Analysis Grid Options dialog box

The **Grid Options** button opens the **Analysis Grid Options** dialog box, which allows you to determine the size and number of divisions within the grid displayed in the Analysis window.

Automatically Size
The **Automatically Size** option automatically sizes the grid based on the number of objects specified in the X, Y, and Z fields.

Specified Size
The **Specified Size** option allows you to key in specific sizes for the maximum and minimum boundaries.

Number of Elements
The **Number of Elements** boxes allow you to specify the number of divisions that will be drawn on the analysis grid.

Maximum Boundary
The **Maximum Boundary** boxes allow you to specify the upper limits of the analysis grid coordinates.

Minimum Boundary
The **Minimum Boundary** boxes allow you

to specify the lower limits of the analysis grid coordinates.

Arrow Length Multiplier

The **Arrow Length Multiplier** box allows you to enter a scaling factor that magnifies the deviation arrows and tolerance zone for the graphical analysis mode. If a value of 2.0 is entered, PC-DMIS will scale the arrows two times the calculated deviation for each feature hit.

This option is used for viewing purposes only, and is not reflected in the text printout.

Show Diameters

The **Show Diameters** check box allows you to show the diameters of available True Position dimensions.

Show Contour Plot

The **Show Contour Plot** option only works for patch scans. The contour plot uses the patch scan points to create a mesh and then shades the mesh with the colors related to the profile deviations from each hit.

Show Color Interpolation Between Hits

The **Show Color Interpolation Between Hits** check box toggles inserted colors between hits.

Show Arrows

The **Show Arrows** option allows the arrows for deviation lines to be toggled on or off.

Show Lines Between Measured Hits

The **Show Lines Between Measured Hits** check box toggles on or off lines between the measured hits.

Show Lines Between Nominals

The **Show Lines Between Nominals** check box toggles lines drawn between the nominal values.

Show Tolerance Lines

The **Show Tolerance Lines** check box toggles the display of the acceptable tolerances for the dimension.

Histogram Object

PC-DMIS EXTENTION OBJECT

The histogram object is used to drop a histogram for a particular dimension directly into the HyperView report. When the histogram object is first dropped into the report, a dialog will appear allowing the user to select the dimension to be associated with the object. The histogram object displays the number of deviations within each of the different tolerance zones. To see how these tolerance zones are defined or to change the colors representing each tolerance zone, see “Editing Dimension Color” in the Editing the CAD Display chapter.

Note: Histogram objects are not available for dimensions that are not directly calculated using the individual hits of their features, such as location, true position, and keyin dimensions.

Dimension Color Key Object

PC-DMIS EXTENTION OBJECT

The dimension color key object is used to drop a color key directly into the HyperView report. This color key depicts the colors that are used to represent the dimensional tolerance zones for analysis and histogram objects. To see how these tolerance zones are defined or to change the colors representing each tolerance zone, see “Editing Dimension Color” in the Editing the CAD Display chapter.

Feature Text Object

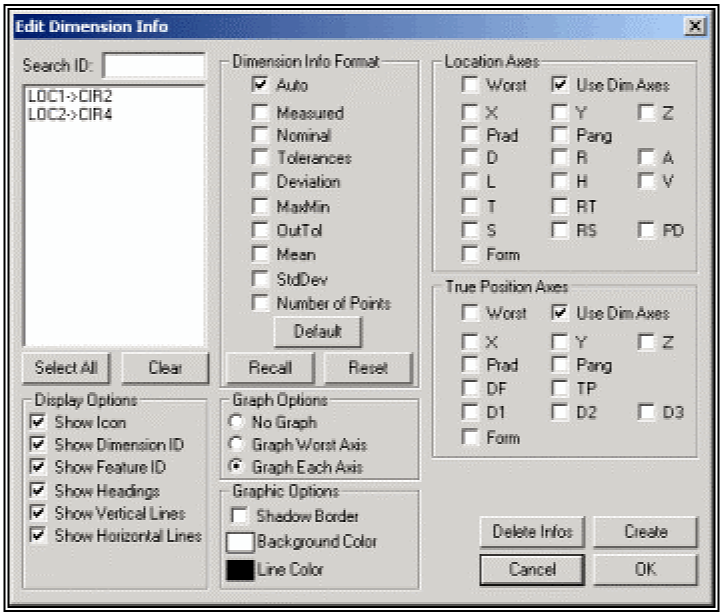
PC-DMIS EXTENTION OBJECT

The feature text object is used to drop the inspection report text for any feature or dimension in the part program directly into a HyperView report. When the feature text object is first dropped into the report, a dialog will appear allowing the user to select the feature or dimension to be associated with the object. The feature text object is only useful if you wish to have the text for a feature or dimension appear in the HyperView report exactly as it would appear in the inspection report. If a dimension has Textual Analysis set in the Edit window, the feature text object can display the point information related to the Textual Analysis of a dimension. The dimension info object described below is a useful alternative when displaying other textual information on a particular dimension.

Dimension Info Object

PC-DMIS EXTENTION OBJECT

The dimension info object is used to drop individual dimension information directly into a HyperView report. When the dimension info object is first dropped into the report, the **Dimension Info** dialog box will appear allowing the user to select the dimension to be associated with the object.


Edit Dimension Info

Search ID:

LOC1->CIR2
LOC2->CIR4

Dimension Info Format

☑ Auto
☐ Measured
☐ Nominal
☐ Tolerances
☐ Deviation
☐ MaxMin
☐ OutTol
☐ Mean
☐ StdDev
☐ Number of Points

Location Axes

☐ Worst ☑ Use Dim Axes
☐ X ☐ Y ☐ Z
☐ Prad ☐ Pang
☐ D ☐ R ☐ A
☐ L ☐ H ☐ V
☐ T ☐ RT
☐ S ☐ RS ☐ PD
☐ Form

True Position Axes

☐ Worst ☑ Use Dim Axes
☐ X ☐ Y ☐ Z
☐ Prad ☐ Pang
☐ DF ☐ TP
☐ D1 ☐ D2 ☐ D3
☐ Form

Display Options

☑ Show Icon
☑ Show Dimension ID
☑ Show Feature ID
☑ Show Headings
☑ Show Vertical Lines
☑ Show Horizontal Lines

Graph Options

○ No Graph
○ Graph Worst Axis
● Graph Each Axis

Graphic Options

☐ Shadow Border
□ Background Color
■ Line Color

Dimension Info dialog box

The dimension info object brings up the **Dimension Info** dialog box. This dialog allows you to display selected dimensional information in your HyperView report.

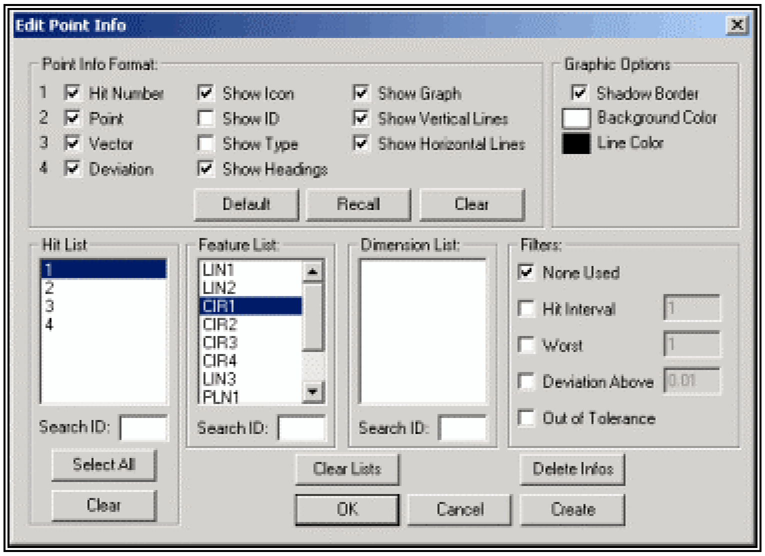
See “Inserting Dimension Info Boxes” in the "Editing the CAD Display" chapter for complete information on how to use this dialog box to display dimension information.

Point Info Object

P1= 1.0

PC-DMIS EXTENTION OBJECT

The edit point info object is used to place individual point information directly into the HyperView report. After selecting this object, and drawing a box on your grid, the **Edit Point Info** dialog box appears.



Point Info Format:

1 ☑ Hit Number ☑ Show Icon ☑ Show Graph
2 ☑ Point ☐ Show ID ☑ Show Vertical Lines
3 ☑ Vector ☐ Show Type ☑ Show Horizontal Lines
4 ☑ Deviation ☑ Show Headings

Graphic Options:

☑ Shadow Border
□ Background Color
■ Line Color

Hit List: 1, 2, 3, 4
Search ID:

Feature List: LIN1, LIN2, CIR1, CIR2, CIR3, CIR4, LIN3, PLN1
Search ID:

Dimension List:
Search ID:

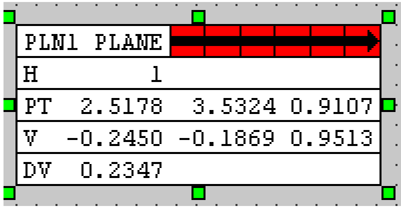
Filters:

☑ None Used
☐ Hit Interval 1
☐ Worst 1
☐ Deviation Above 0.01
☐ Out of Tolerance

Edit Point Info dialog box

Select the information you want in your report, then click **OK**. PC-DMIS creates point info commands for the selected features or dimensions and closes the dialog box.

You will notice that PC-DMIS displays a table on your grid that contains feature or dimension point information.



```
PLN1  PLANE
H          1
PT  2.5178  3.5324 0.9107
V  -0.2450 -0.1869 0.9513
DV  0.2347
```

An inserted Point Info object

This object contains XYZ coordinate information and IJK vector information for the second hit on a circle feature.

The **Show Leader Line** check box controls whether or not leader lines (lines from the feature to the displayed Point Info information) will automatically be generated.

Note: Analysis Objects or Cad Objects must exist in your HyperView report previous to selecting this check box.

See "Inserting Point Info Boxes" in the "Editing the CAD Display" chapter for information on the other options in this dialog box.

Properties in PC-DMIS tab for Point Info Objects:

The following table contains the PC-DMIS properties available for a Point Info object when you access its **Property Sheet** dialog box.

Property	Possible Values	Description of Associated Value
(Settings Dialog)	…	Clicking it opens the

		Edit Point Info dialog box
DimFeatID	Any string (default is something like F1, F2, F3 etc.).	This value determines the Feature's ID.
HitNumber	Any hit number for the given feature type.	This value determines what hit number is under examination.
OrderDeviation	1 through 4.	This value determines placement order of the deviation in the Point Info box.
OrderHitNum	1 through 4.	This value determines placement order of the hit number in the Point Info box.
OrderPoint	1 through 4.	This value determines placement order of the XYZ point location in the Point Info box.
OrderVector	1 through 4.	This value determines placement order of the IJK vector in the Point Info box.
ShowColumnLines	0 or 1.	A value of 1 shows column lines in the Point Info box.
ShowGraph	0 or 1.	A value of 1 shows the graph on the Point Info box.
ShowHeadings	0 or 1.	A value of 1 shows the row headers.
ShowID	0 or 1.	A value of 1 displays the feature's ID.
ShowPointInfo	0 or 1.	A value of 1 displays the hit's point info.
ShowType	0 or 1.	A value of 1 displays the feature type next to the graph.

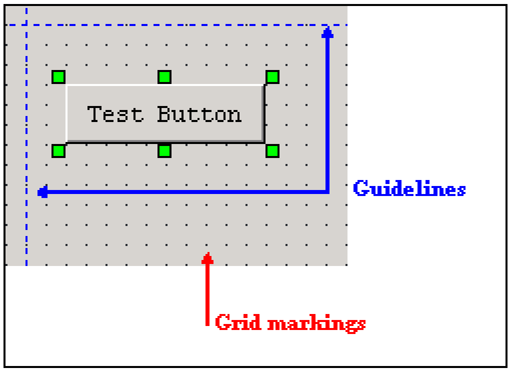
Leader Line Object

PC-DMIS EXTENTION OBJECT

The leader line object is used in conjunction with a feature text or dimension info object and a CAD window. It provides a leader line from either a feature text or dimension info object to a CAD window, with the leader line terminating on the feature in the CAD window associated with the feature text or dimension info object. When dropping a leader line object into a report, drag the mouse from the feature text or dimension info object to the CAD object (or vice versa). If done properly, the leader line will automatically snap between that feature text or dimension info object and the appropriate location on the CAD window. If not, a dialog will pop up allowing the user to select the feature text or dimension info object and the CAD object to be used. When a CAD, feature text, or dimension info object associated with a leader line object is deleted, the leader line object is automatically deleted as well.

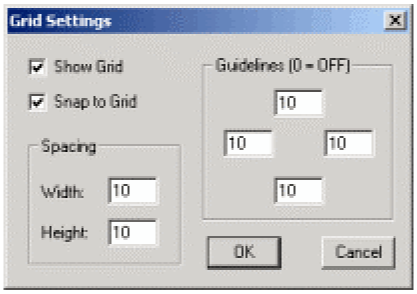
Working with the Grid

When working in the Hyperview Editor's Edit Mode PC-DMIS can display a grid in the background. You can use this grid as a tool for the accurate placing and/or sizing of objects and how to control a Hyperview report's margin guidelines.



Example showing the default grid's markings and guidelines

Select **Edit | Layout | Grid Settings**to access the **Grid Settings** dialog box.



Grid Settings dialog box

You can then use this dialog box to customize the grid. For a description of the controls in this dialog box, see the table below:

Dialog Box Control	Description
Show Grid check box	The **Show Grid** check box toggles the display of the grid.
Snap to Grid check box	The **Snap to Grid** check box determines whether or not controls placed on the grid get moved, or snapped, to the nearest grid location.
Spacing area	The **Spacing** area sets the amount of spacing between the grid markings. The smaller you set these numbers, the closer together the grid markings will be.
Guidelines area	The **Guidelines** area allows you to determine the location of the top,

	left, right, and bottom margin guidelines for the HyperView editor. The value in each of these boxes sets the distance in grid markings to where the guideline will be set. Setting the value to zero will remove the guideline completely. You cannot place objects beyond these guidelines.

Using Paste Special

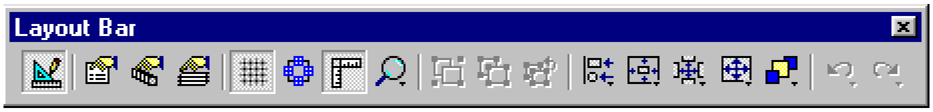
The **Edit | Paste Special** menu item differs from the standard **Paste** menu item since it allows you to maintain a link between the pasted item and the source from which it was copied so that if the source changes, the pasted item will also get updated.

For example, suppose you want to place a Microsoft Excel chart object in your Hyper Report and you want to keep the chart's data in synch with the data from the Excel file. To do this:

1. In the Microsoft Excel file, copy the chart object.
2. Access your Hyper Report.
3. Select **Edit | Paste Special**. The **Paste Special** dialog box appears.
4. Select the **Past Link** option.
5. Click **OK**. The chart object appears in your Hyper Report.

Now if you make changes to the data for your chart object in your Excel file, PC-DMIS will update the chart object in the Hyperview Editor.

Using the Layout Bar to Organize your Report



The **Layout Bar** gives you the tools you need to effectively arrange, organize, and display your objects inside the edit mode grid. The following topics discuss the functionality of each icon.

Edit

The **Edit** icon places the Hyper report into Run mode.

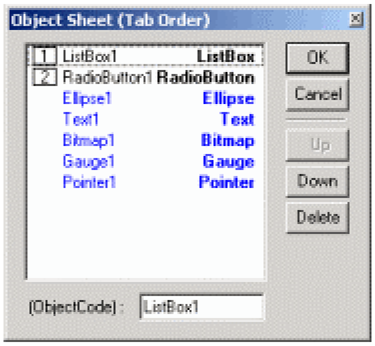
Property Sheet

The **Property Sheet** icon displays the properties for the selected object in the **Property Sheet** dialog box. The **Property Sheet** dialog box allows you to change the properties for a selected object. You can easily change the object whose properties you're looking at by choosing a different existing object from the list at the top of the **Property Sheet** dialog box.

For complete information on the available properties, see "Changing Object Properties with the Property Sheet" on page 395.

Object Sheet

The **Object Sheet** icon displays the **Object Sheet** dialog box.



Object Sheet (Tab Order)

1	ListBox1	ListBox
2	RadioButton1	RadioButton
	Ellipse1	Ellipse
	Text1	Text
	Bitmap1	Bitmap
	Gauge1	Gauge
	Pointer1	Pointer

(ObjectCode) : ListBox1

Object Sheet dialog box

You can use this dialog box to set the tab order for objects and controls typically used on dialog boxes by simply changing the order an object appears in this dialog box.

When you press TAB in run mode, PC-DMIS will jump to the next object defined in this dialog box, giving it the focus. Only some objects are affected by the TAB key. Pressing TAB only works on those objects that have a number to the left of them in the **Object Sheet** dialog box.

Layers Sheet

The **Layers Sheet** icon displays the **Layers Sheet** dialog box. This dialog box allows you to assign objects to different layers. You can hide or show layers. This is useful for multi-page reports that have a lot of objects. For detailed information on the **Layers Sheet** dialog box, see "Using Layers to Manage Multiple Objects" on page 449.

Toggle Grid

The **Toggle Grid** icon shows or hides the available grid. Used in Edit mode, the grid is useful for sizing objects precisely since it contains a dot every tenth of an inch.

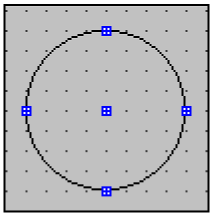
View Snap Points

The **View Snap Points** icon shows the snap points on each object. Snap points are useful in tying objects together with lines. For example, you can create a line between a text box description pointing to an object, such as a circle, by using a line to connect snap points on the text box description and the circle object. Then, like a leader line, when you move the circle to a new location, the attached line stretches and shrinks to accommodate the new position of the circle.

To use snap points as described in the above example,

1. Create a circle object.

2. Create a line object.
3. In edit mode, select the **View Snap Points** icon from the **Layout Bar**. You will notice that objects on the grid that have an area will display blue-colored points. These blue points are snap points.



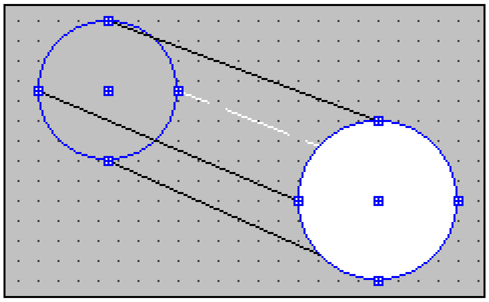
Sample circle showing five snap points

4. Select the line and drag one end of it until the end rests directly on the desired snap point of the circle. Try moving the circle object; the line is attached to the circle object's snap point.
5. Create a text object for a description for the circle and edit it with a brief description.
6. Select the other end of the line and drag it until it rests on one of the snap points of the text object.
7. Now select and drag either the text box description or the circle object. Notice that the line object gets longer or shorter and moves so that there's always a straight line between the used snap points of the two objects.

To detach objects attached to a snap point, simply drag the attached end of the object off of the snap point.

Snap Points vs. Leader Line objects

Most of the time you will probably use the **Leader Line** object instead of snap points. It has nearly all the same functionality as snap points and is easier to use (see the "Leader Line Object" topic on page 442). One advantage snap points have over the **Leader Line** object, is their ability to support more than one line between objects and their ability to fit the connecting line exactly to a snap point. For example, with snap points you can create a more complex shape directly in the edit mode, as shown here:



Sample Cylinder created with two four lines fitted to snap points on the two filled circles.

Supported Objects

Only Line and Polyline objects can attach to objects that have snap points. Below are the objects that have snap points:

- Text

- TextVar
- Border
- Ellipse
- Bitmap
- Dimension Color Key
- Histogram
- Feature Text
- Dimension Info
- Point Info

Viewing Attached Objects

On complex pages with a lot of objects, you may find it useful to use the **View | Route** menu option to see the chain of objects connected to the currently selected object. This menu option selects all objects connected in some way by snap points to the currently selected object.

Toggle Ruler

The **Toggle Ruler** icon allows you to turn on or off the ruler located on the top and left sides of the grid. The ruler shows the size of the grid in inches. It also displays your cursor's position along the grid's horizontal axis (top ruler) and vertical axis (side ruler).

Zoom In or Out

The **Zoom In or Out** icon allows you to zoom in or out of your Hyper report and then restore the original view. When you click this icon, the following icons become available for selection:

Zoom in 100% - This zooms in on your report by 100 percent per click. You can click this icon at most five times to get a 500 percent zoom.

Zoom out 100% - This zooms out of your report by 100 percent per click if you're already zoomed in. If you haven't zoomed in at all, you will only be able to zoom out by 100 percent one time.

Zoom to 1:1 – This restores the size of your report to the original 1 to 1 zoom size.

Group

The **Group** icon allows you to select multiple objects and group them together. Once grouped, the editor will show the grouped objects as one object.

To group multiple objects, either hold down the SHIFT button, and select all objects in the group, or box-select all objects, and then click the **Group** icon.

Ungroup

The **Ungroup** icon restores object in a group to their individual ungrouped states. To do this, select a grouped object and click **Ungroup**.

Regroup

The **Regroup** icon regroups the last ungrouped set of objects without your having to reselect the individual objects.

Align Edges

The **Align Edges** icon provides you with these additional icons that allow you to arrange a set of selected objects along their left, right, top, or bottom edges:

Aligns selected objects along their left edges

Aligns selected objects along their right edges

Aligns selected objects along their top edges

Aligns selected objects along their bottom edges

Align Center

The **Align Center** icon provides you with these additional icons that allow you to center selected objects either horizontally or vertically:

Centers selected objects along the horizontal axis

Centers selected objects along the vertical axis

Space

The **Space** icon provides you with these additional icons that allow you to equally space selected objects either horizontally or vertically:

Equally spaces selected objects along the horizontal axis

Equally spaces selected objects along the vertical axis

Make Same Size

The **Make Same size** icon provides you with these additional icons that allow you to equalize the width, the height, or both the width and height of all selected objects:

Equally sizes the widths of the selected objects

Equally sizes the heights of the selected objects

Equally sizes both the widths and the heights of the selected objects

To Front or Back

The **To Front or Back** icon provides you with additional icons that allow you to place a selected object (or objects) in front or in back of other objects. This is helpful when you have multiple objects on top of each other and you want certain objects to be visible. The available icons include the following:

- Moves the selected object(s) in front of all other objects so that it is shown on top
- Moves the selected object(s) behind all other objects so that is shown beneath

Undo the Last Action

The **Undo the Last Action** icon will undo the last action you took in the editor.

Redo the Previously Undone Action

The **Redo the Previously Undone Action** icon will redo the action that was undone when you clicked the **Undo the Last Action** icon.

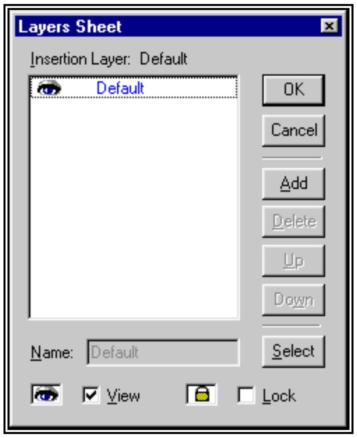
Using Layers to Manage Multiple Objects

In order to properly manage multiple objects (often the case when working with multiple page reports) you will want to assign your objects to different layers.

Layers Sheet Dialog Box Information

Layers Sheet icon

To access the **Layers Sheet** dialog box, click the **Layers Sheet** icon from the **Layout** toolbar located within the Hyperview interface.



Layers Sheet

Insertion Layer: Default

Default

Name: Default

☑ View ☐ Lock

Layers Sheet dialog box with the Default layer

Control	Description
Default layer	The **Layers Sheet** dialog box always has a layer called **Default**. Initially, all your objects are assigned to this

	layer.
Add button	Adds a new layer.
Delete button	Deletes the selected layer. Objects assigned to the layer are not deleted but are assigned to the **Default** layer.
Up / Down buttons	These buttons move a selected layer up or down in the dialog box's list of layers.
Select button	Selects those objects assigned to the specified layer.
Name box	Shows the name of the layer. You can change the name for all layers except the **Default** layer.
View check box	Determines whether or not a layer is visible in the report. If visible you'll see an eye icon appear next to the layer.
Lock check box	This turns off the ability to use the **Select** button for the specified layer.

To Add New Layers

1. Click the **Layers Sheet** icon from Hyperview's **Layout** toolbar. The **Layers Sheet** dialog box appears.
2. Click **Add**. A new layer named **Layer1** will appear in the dialog box. Subsequent layers, when added, will follow this naming pattern (**Layer2**, **Layer3** etc.).
3. Rename the new layer if needed by typing the text in the **Name** box.
4. Continue adding layers as desired.
5. Click **OK** when finished.

Your new layer is now added to the dialog box. Once new layers are created, you can add objects to them.

To Assign an Object to a Layer

1. Select the object.
2. Right-click to open the **Property Sheet** dialog box.
3. Select the **Settings** tab.
4. From the **Layers** setting, choose the layer you want associated with this object.
5. Close the property sheet.

6. Test to see if you've correctly associated the object with a layer by clicking the **Select** button from the **Layers Sheet** dialog box.

Creating Multiple Page Reports

There are two styles of hyperview reports.

1) Single Page
2) Multiple Page

Single page reports are the most common form of reports because they are easier to create. However, if you are a more advanced user, the ability to create a multiple page report lies at your fingertips. The creation of a multiple page report assumes you have a working knowledge of the BASIC programming language. You will need this understanding to cause multiple page reports to function properly.

With multiple page reports you are able to group specific features, dimensions or any other information from a part program onto various pages. This allows you to have much more organized reports and allow better access to a part's information.

Description of Handling Multiple Page Reports

A multiple page report is actually one report viewed in different ways at different times. You, as the programmer, can use the CurPage variable to determine what happens on each "page" of the multiple page report.

Each page of the multiple page report shares **The Frame/The View**. Whatever objects are put inside the Frame will be included on each page. Within an event handler, using the CurPage variable and BASIC code, you can cause changes to the current page, customizing each page to fit your needs. For example, you can make items hidden to the user or insensitive (unresponsive to user input). It is possible to make each page different using the BASIC language.

Description of the CurPage and NumPages local variables

The CurPage variable is updated anytime you change between pages. It contains a number that corresponds to the current active page. When writing BASIC code you can use this variable to determine which page is active. Once the active page is known, you can then write code with different actions for each page.

Example of creating a multiple page report

To create a multiple page report with three pages available:

1. Right-click on HyperView's main screen.
2. The **Property Sheet** for **The Frame/The View** opens.
3. Select the **Events** tab.
4. Double click EventInitialize. The **VBS Mini-Editor** will open.
5. Click the **Wizard >>** button. The **Script Wizard** dialog box opens.

6. In the **Variables** area of the dialog box, pull down the drop-down list on the left. You will see that **CurPage** and **NumPages** are available for selection.
7. Select **NumPages**.
8. Click the **Insert** button. This will bring up the **VBS Mini-Editor** with the following text:
   `"NumPages"`
9. Type "**=3**" after the NumPages command. So that the text reads:
   `“NumPages=3”`

This will cause you to have three pages available. Select CTRL + E and test the report. Press PAGE UP and PAGE DOWN to cycle through the pages.

Example of displaying an object on only one page

Unless you tell PC-DMIS otherwise, objects in your Hyperview report will be displayed on every page of your multi-page report.

If you want to display an object (such as a text box) on just the first page of your report, for example, do the following:

1. Access the **Hyperview Editor** option and then open your report.
2. Make sure you're in Edit Mode by pressing CTRL + E.
3. Click once on the text box to select it.
4. Right-click on it to bring up the **Property Sheet** dialog box.
5. Select the **Events** tab.
6. From the list of events, double click on **EventPumpData**. The **VBS Mini-Editor** for the text object appears.
7. Type the following code all on one line:
   ```
   If CurPage = 1 Then This.Visible = 1 Else
   This.Visible = 0
   ```

Note: This code will show the selected text box on only page one. On all other pages it will be invisible.

8. Click **OK**. You'll notice that this information is now included in the "Property Sheet" for this event.
9. Click on the title bar of the report window to make it the active window.
10. Press CTRL + E to test your report. PC-DMIS will ask if you want to save your changes. Click **Yes**.
11. Press PAGE UP and PAGE DOWN to cycle through your report pages. You'll notice that the text box only appears on your indicated page.

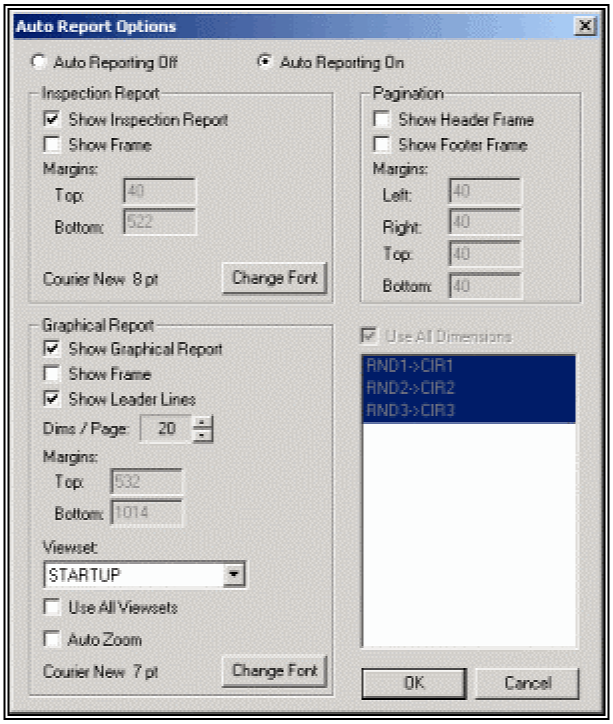
Using the CurPage variable you can determine which pages display your objects.

Generating Automatic Reports

PC-DMIS allows reports to be generated automatically. These reports can be presented in a textual fashion, graphically, or both. Even though these reports are automatically generated in a set format, PC-DMIS allows some customization by allowing you to add custom headers or footers by adjusting the report's margins.

To enable AutoReporting,

1. Access the Hyper-Report Editor by selecting **View | Hyperview Editor** from the menu bar.
2. Select **File | New** to create a new report.
3. Right mouse click on the background of the HyperView window to open a **Property Sheet**.
4. Make sure **TheFrame/TheView** is selected in the drop-down list.
5. Double click **Auto Reporting** to open the **Auto Report Options** dialog box.



○ Auto Reporting Off ● Auto Reporting On

Inspection Report

☑ Show Inspection Report

□ Show Frame

Margins:

Top: 40

Bottom: 522

Courier New 8 pt

Pagination

□ Show Header Frame

□ Show Footer Frame

Margins:

Left: 40

Right: 40

Top: 40

Bottom: 40

Graphical Report

☑ Show Graphical Report

□ Show Frame

☑ Show Leader Lines

Dims / Page: 20

Margins:

Top: 532

Bottom: 1014

Viewset: STARTUP

□ Use All Viewsets

□ Auto Zoom

Courier New 7 pt

☑ Use All Dimensions

RND1->CIR1
RND2->CIR2
RND3->CIR3

Auto Report Options dialog box

6. Select the **Auto Reporting On** option to enable AutoReporting. The **Auto Report Options** dialog box is divided into four areas:
   - Inspection Report
   - Graphical Report
   - Pagination
   - A Dimension Selection List

The Inspection Report

This report will mirror the standard inspection report generated by a part program. Any inspection report options that are set within PC-DMIS will be reflected in this report as well. For example, if the reporting options are set to display dimensions only, only dimensions will appear in the auto-generated inspection report.

Creating a Textual Report

To create a textual report for the current part program:

1. Select the **Show Inspection Report** check box.
2. Clear the **Show Graphical Report** check box.
3. Click the **OK** button.

Generating a Border

Selecting the **Show Frame** check box will generate a black border around the Inspection Report. This can be useful for visual clarity by keeping the Inspection Report distinct from the Graphical Report when both are selected.

Adjusting Font Attributes

Press the **Change Font** button and PC-DMIS displays the standard **Font** dialog box. From this dialog box you can modify the various font attributes such as font, font style, size, and color.

The Graphical Report

This section allows you to generate a graphical report. This option grabs the current active graphics from PC-DMIS's Graphics Display window. By default, a leader line will connect each measured feature to a DimensionInfo box.

Creating a Graphical Report

To create a graphical report:

1. Select the **Show Graphical Report** check box.
2. Deselect the **Show Inspection Report** check box.
3. Click the **OK** button. A graphical report will then be generated.

Generating a Border

Selecting the **Show Frame** check box will generate a black border around the Graphical Report. This can be useful for visual clarity by keeping the Inspection Report distinct from the Graphical Report when both are selected.

Showing Leader Lines

If the **Show Leader Lines** check box is selected, the leader lines connecting the measured features to the DimensionInfo boxes will be generated. Clearing this check box tells PC-DMIS not to generate the leader lines.

Specifying the Number of DimensionInfo Boxes Displayed

The **Dims/ Page** Spin box allows you to specify how many DimensionInfo boxes you want displayed on each page. If the number of DimensionInfo boxes on each page is causing the report to become cramped looking, reduce the number of DimensionInfo boxes per page.

Adjusting Margins

The **Margins** area shows the current margin sizes. These are for information purposes only. To adjust the margins of the Graphical Report, make sure you are in

Edit Mode then click and drag the cyan colored borders in the PC-DMIS Report window.

Using Viewsets

The **Viewset** list lets you select a saved view and include it in your hyper report. This is useful when you want to refer to important labels, dimension boxes, orientations, etc in the report.

If you select the **Show Graphical Report** check box and select a saved view from the **Viewset** list, PC-DMIS displays that one saved view in the report.

If you select the **Show Graphical Report** check box and select the **Use All Viewsets** check box, then PC-DMIS displays all saved views in your report. PC-DMIS will put each individual view on its own page.

For information on creating and recalling viewsets, see "Defining and Saving a Viewset" and "Recalling a Saved Viewset" in the "Inserting Report Commands" chapter.

Using Auto Zoom

Selecting the **Auto Zoom** check box causes PC-DMIS to magnify the part display in the Graphics Display window as far as possible while still displaying the points on the CAD model that match the reported hits.

Adjusting Font Attributes

Press the **Change Font** button and PC-DMIS displays the standard **Font** dialog box. From this dialog box you can modify the various font attributes such as font, font style, size, and color.

Using the Dimension List Box

When the Graphical Report is selected, the dimension selection list is activated. You can do the following:

- Select the **Use All Dimensions** check boxto generate a **DimensionInfo** boxfor all the dimensions in the Part Program.
- Deselect the **Use All Dimensions** check boxand select the dimensions you want in the report by toggling them on or off.
- When the **Use All Dimensions** check boxis *not* selected, you can double-click a dimension to bring up a **Dimension Info** dialog box Any modifications made in this dialog box will be saved with the HyperView report. See "Dimension Info Object" on page 439 for more information on the options in this dialog box.

The Dual Report

The Dual Report consists of both textual and graphical information. To create the dual report:

1. Select the **Show Graphical Report** check box.
2. Select the **Show Inspection Report** check box.
3. Click the **OK** button. A Dual Report with both text and graphics will be generated.

The textual portion of this report is similar to the Inspection Report. The graphical portion is similar to the Graphical Report except that the DimensionInfo boxes are not generated.

In addition to being able to adjust the page margins, a border is displayed between the textual and graphical portion. Click and drag this line to adjust the relative size of each portion.

Auto-Report Regeneration

Once an Auto-Report is created and saved, this report can be loaded for any Part Program. Each time a report is loaded for a program, it will be regenerated and will reflect the dimensions in the current Part Program.

Auto-Report Customization

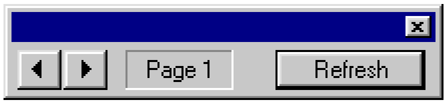
Auto-Reports can be customized by adjusting the report's margins to allow space for other HyperView objects.

Example: To add a page number at the bottom of every page of your Auto-Report, follow these steps:

1. Load a Part Program that has enough dimensions to generate a multi-page Auto-Report.
2. Create a new HyperView report by accessing the HyperView Reporting Editor. (the "Using the Road Map" appendix provides instruction on accessing all dialog boxes and menu options)
3. Right mouse click on the background of the HyperView window to get a **Property Sheet** dialog box. Make sure **TheFrame/TheView** is selected in the drop-down list. Double click **Auto Reporting** to open the **Auto Report Options** dialog box.
4. Turn Auto Reporting on by selecting the **Auto Reporting On** option.
5. Click the **OK** button to close the **Auto Report Options** dialog box.
6. Scroll to the bottom of the PC-DMIS Report window and drag the bottom margin up far enough to allow room for the page number text.
7. Select the **Text object** from the **Object** toolbar.
   A
8. Drag a rectangle in the report window to define the Text Object.
9. Right mouse-click the Text Object to bring up its **Property Sheet** and select the **Events** tab.
10. Double click **EventPumpData** to bring up the **VBS Mini-Editor**.
11. In the edit field, enter the following line of text: **This.Text="Page " & CurPage & " of " & NumPages**
12. Click the **OK** button.
13. Save the report.

14. Switch to Run mode by pressing CTRL + E.

Scroll to the bottom of your report. The page number should be displayed in the report window below the graphical portion. Use the **Page Selection** box to scroll through the pages of your report. The page number at the bottom of your report will change accordingly.



Page Selection box

Using this example as a guide, other custom objects can be added to an Auto-Report.

Using HyperView Reports in Part Programs

PC-DMIS allows HyperView reports to be executed inside of part programs. You can optionally execute a default HyperReport at the end of every part program or you can insert a custom HyperView report in the body of a part program.

Additionally, the HyperView report can interact with a part program by mapping global variables and property elements to PC-DMIS variables.

Setting a Default Report for a Part Program

A default HyperView report for a part program is a report that will automatically load and print upon completion of the Part Program execution. This default report can be set to load and print for every Part Program that is executed. To set the default report:

1. Select the **Edit | Preferences | Edit Window Layout** menu option. The **Edit Window Layout** dialog box appears.
2. Select the **Hyper-Report** tab on the **Report Display Options** dialog box that appears.

Edit Window Layout dialog box –Hyper-Report tab

If the **Auto Load Hyper-Report** check box is selected, a HyperView report will be executed upon completion of the Part Program.

When the **Auto Load Hyper-Report** check box is selected, specify a Hyper-Report file to load or use the Default Report. Specify a file by typing a file name in the **Hyper-Report File** edit box or click the **Browse** button to use a standard **Open Dialog** box to select a report file. To use the Default Report, check the **Use Default Report** check box.

Note: The Default Report is recorded in the PC-DMIS.INI file. Initially, there is no Default Report set in this file. To set the Default Report, make sure **Use Default Report** check box is cleared, select a Hyper-Report File, and then check the **Use Default Report** check box. This will set the Default Report in the .INI file to the file listed in the **Hyper-Report File** edit box.

Embedding HyperView Reports in a Part Program

To embed a HyperView report in a Part Program:

1. Make sure the Edit window is in Command Mode.
2. Insert the HyperView Report you just made by selecting the **Insert | Report Command | HyperView Report** menu option (the "Using the Road Map" appendix provides instruction on accessing all dialog boxes and menu options). The **Insert Hyper Report** dialog box will appear.
3. Select the HyperView Report you want to embed in the Part Program.
4. Click the **Open** button.

The HyperView Report command object will be inserted in the Edit window at the cursor position.

The HyperView Report Command Object

The HyperView Report command object is identified with the `REPORT/FILENAME` command in the Edit window. Following this command is the file name for the HyperView Report, the AUTOPRINT option, and then PARAM/ options.

AUTOPRINT

Switching the AUTOPRINT option to PRINT will cause the HyperView Report to be printed to the default printer after it has loaded and executed. The report will close immediately after the print job has been spooled and execution of the part program will resume.

Switching the AUTOPRINT option to PDF will cause the HyperView Report to be printed to a Portable Document Format (PDF) file.

About the Generated PDF Filename

- The name of the generated PDF file will have the same base name as the HyperView Report filename appended by a numerical index and extension.
- Also, the generated file will be located in the same directory as the Report file.

- If a file exists with the same name as the generated filename, the index will automatically increment itself until a unique file name is found.

Note: PC-DMIS overrides the AUTOPRINT option if you select the **Hyper Report Inline** check box found in the Edit window's **Print Options** dialog box. See "Printing from the Edit Window" in the "Using Basic File Options" chapter for information on this check box.

PARAM

The PARAM/ option allows you to map PC-DMIS expressions to HyperView variables and properties. The syntax of the PARAM/ option is:

```
PARAM/{HyperView variable name}={PC-DMIS expression}
```

Example: If you have a Global Variable in a HyperView Report named “NumBoltHoles”. The following PARAM/ option would map the PC-DMIS expression to NumBoltHoles:

```
PARAM/NumBoltHoles=360.0/angle
```

If the PC-DMIS expression being mapped to a HyperView Global Variable is an L-value, then changes to the HyperView variable will change the PC-DMIS variable.

Example: Using the following PARAM/ option, NumBoltHoles is set equal to numBolts. Changes within the HyperView report to the HyperView variable NumBoltHoles will be reflected in the PC-DMIS variable numBolts:

```
PARAM/NumBoltHoles=numBolts
```

Mapping PC-DMIS expressions to properties of HyperView Objects

PC-DMIS expressions can be mapped to HyperView property objects using the PARAM/ command in the Edit window. The syntax for mapping a PC-DMIS expression to a HyperView object property is:

```
PARAM/{object code}.{property name}={PC-DMIS expression}
```

Example: The following PARAM/ option changes the BorderStyle property of the object, Text1:

```
PARAM/Text1.BorderStyle=1
```

Bi-directional nature of expression mapping if expression is an l-value

If the PC-DMIS expression is an L-value, changes to the HyperView property element will be reflected in the PC-DMIS variable.

Example: The following PARAM/ option maps the PC-DMIS L-value myTitle to the HyperView frame object property Title. Any changes to Title within the HyperView report will be reflected in myTitle.

```
PARAM/TheFrame.Title=myTitle
```

Examples of HyperView report usage

Below are two examples that show how HyperView reports may be used. Example 1 demonstrates the use of putting a single page HyperView report in more than one place in a part program. Example 2 shows how HyperView reports are used to gather information from the user to modify PC-DMIS variables.

Example 1

This example demonstrates the use of a single page HyperView report used in multiple places in a Part Program.

1. Create a part program with at least two dimensions in it (D1 and D2).
2. Access the Hyper-Report Editor by selecting **View | Hyperview Editor** from the menu bar.
3. Select **File | New** to create a new report.

Dimension Info object

4. Select the **DimensionInfo** object from the **Object** toolbar.
5. Drag a rectangle in the report window to define the DimensionInfo Object.
6. Fill in the **Dimension Info** dialog box or leave the default values.
7. Save and close the report.
8. Insert the HyperView Report you just made by selecting the **Insert | Report Command | HyperView Report** menu option (the "Using the Road Map" appendix provides instruction on accessing all dialog boxes and menu options).
9. In the Edit Window, modify the `PARAM/=` line to read: `PARAM/DimensionInfo1.DimID=D1`

Note: You may have to change "DimensionInfo1" to match the name given to your DimensionInfo Object. To determine the name of the object, right-click on the object in the Report Window to view its Property Sheet and then select the Settings tab. The ObjectCode name is listed on the top line.

10. Now insert the HyperView Report again.
11. Modify the `PARAM/=` line to read: `PARAM/DimensionInfo1.DimID=D2`

When the part program is executed, the Dimension Info will be reported for both dimension Ids using the same HyperView report.

Example 2

The following example demonstrates how a HyperView Report can be used to gather information from the user to modify PC-DMIS variables.

1. Create a PC-DMIS long integer variable: `ASSIGN/NumBoltHoles=4`
2. Access the Hyper-Report Editor by selecting **View | Hyperview Editor** from the menu bar.
3. Select **File | New** to create a new report.

EditLongBox object

4. Select the **EditLongBox** object from the **Object** toolbar.
5. Drag a rectangle in the report window to define the **EditLongBox** object
6. Save and close the report.
7. Insert the HyperView Report you just made by selecting the **Insert | Report Command | HyperView Report** menu option (the "Using the Road Map" appendix provides instruction on accessing all dialog boxes and menu options).
8. In the Edit window, modify the `PARAM/=` line to read: `PARAM/EditLongBox1.Text=NumBoltHoles`

This will link the PC-DMIS variable NumBoltHoles to the Edit Box in the HyperView report. When the program is executed, the user can enter a new number in the Edit Box, which in turn will modify NumBoltHoles. This method can be used to gather information from the user to use in the program.