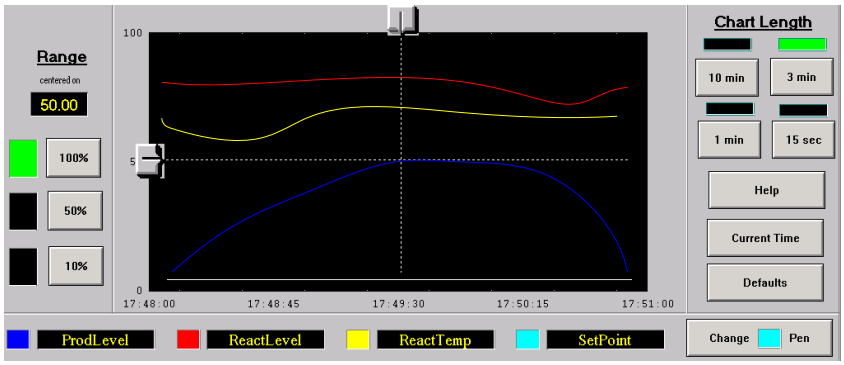
CHAPTER 10

Trending Tag Data

In This Chapter

About Trending Tag Data

You can create trends that graphically show data collected from an InTouch application. WindowMaker includes a set of utilities and wizards that enable you to create historical and real-time trends. The following figure shows an example of an InTouch Average/Scatter trend.



You can also use a set of trend controls. Using these controls, you can select the data shown in a trend and how data appears in the trend.

You can configure real-time and historical trends. Both trend types include configuration options to set a trend's data collection interval and visual appearance.

Types of InTouch Trends

A historical trend shows log data collected from the past and stored in InTouch data repositories.

Using a distributed history system, you can retrieve historical data from any InTouch historical log file located on an accessible network node. The distributed history system extends the retrieval capabilities of historical trends to include remote log databases.

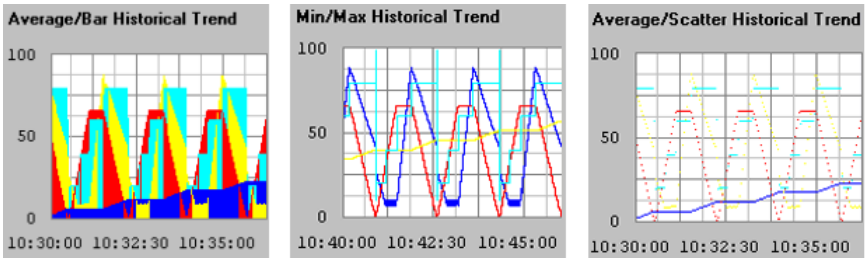
Real-time trends update continuously to show data as it occurs over relatively short periods. You can use WindowMaker's Real-time Trend tool to create a trend object in a window. If the optional 16-Pen Trend tool is installed, you can also create real-time trends that show data from up to 16 tags.

Understanding Historical Trends

Historical trends show a contiguous segment of data from the past. Unlike real-time trends, historical trends are updated only by a script or an operator action.

A historical trend shows a graphical representation of data from a maximum of eight tags. You assign the data that appears in a historical trend by assigning a tag to a trend pen.

The figure below shows the three types of InTouch historical trends.



- The Average/Bar historical trend shows the average value of a data point during the time intervals in bar form.
- The Min/Max historical trend shows the changes in the percentage of engineering units scale as a vertical line over the time span. The emphasis is on time flow and rate-of-change, rather than amount of change.
- The Average/Scatter historical trend shows the average value of the data points over each trend time interval.

You can create graphical sliders called scooters to access the details of trend data based on the scooter's current position within a trend. For example, when the operator positions the scooter over an area on the trend that has visible data, the time and values at that location for all database values being trended are shown.

You can also create buttons or sliders to zoom in and out between the scooters or to data, such as the maximum to minimum value. Average and standard deviation can be shown for the complete chart or for the area between the scooters.

Historical trends can also be scrolled by any amount of time. You can create custom scales and link them to the .**MinEU** and .**MaxEU** dotfields to create a trend that shows the full range of data set by its engineering unit.

Understanding Real-Time Trends

A real-time trend shows data from an InTouch application that is currently running. Real-time trends are continuously updated. Real-time trends plot current data values associated with up to four local tags or expressions.

You can:

- Create a real-time trend
- Select the tags for a trend
- Specify the time span and update interval of a trend
- Configure the display options of a trend

Showing Saved Tag Values in a Historical Trend

You can create historical trends by using any of the following WindowMaker utilities:

- Historical Trend tool

- Historical Trend Wizard
- 16-Pen Trend Wizard (Optional)

In addition, you can incorporate a ActiveFactory trend to show InTouch historical trend data saved to a IndustrialSQL ServerArchestrA Bulk Import Utility database.

Using Historical Trend Objects

You can create and configure a trend with the WindowMaker Historical Trend tool. You can

- Create a historical trend
- Select the tags for a trend
- Specify the time span and update interval of a trend
- Configure the display options of a trend

Creating a Historical Trend

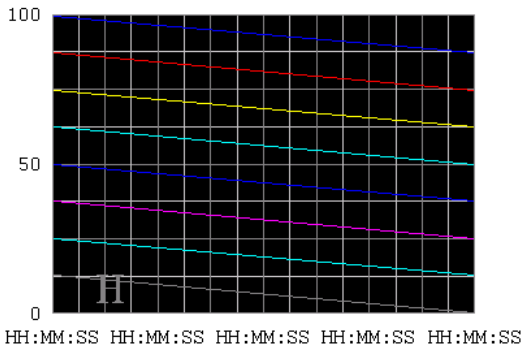
You can use the Historical Trend tool to create a trend object in a window. The first time you create a historical trend object, the InTouch default configuration settings are used.

After you configure a historical trend, WindowMaker uses the most recent configuration values as the initial settings for a new trend object.

You can draw the trend chart to any size within the borders of the window.

To create a historical trend

1. Open the window in WindowMaker in which you want to place a historical trend.
2. Click the **Historical Trend** button from the Drawing Toolbar.
3. Move the mouse over the window area where you want to place the historical trend. Drag the mouse diagonally to create a rectangle the desired size of the trend. The Historical Trend object appears in the window.



4. If needed, adjust the height and width of the trend with its object handles.

Configuring Which Tags to Show From a Historical Trend

A historical trend pen creates a graphical representation of logged data from a specified period. You assign trend pens to the tags that collect historical data.

A historical trend supports up to eight pens.

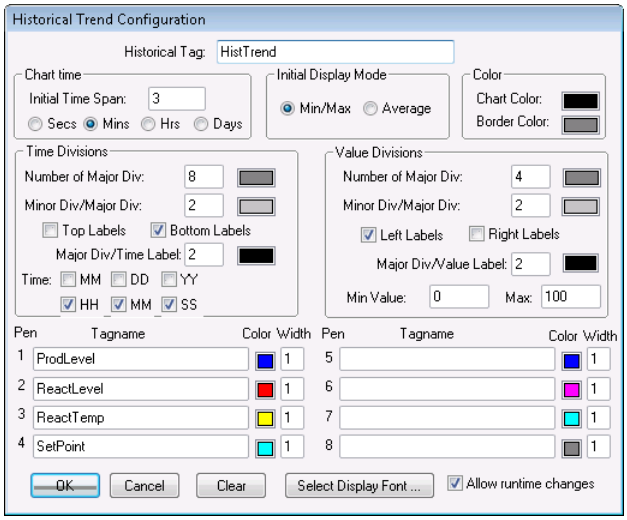
Note: WindowViewer must be closed. Otherwise, the Pen boxes cannot be selected.

You can select tags from remote history providers, if the providers are configured. For information about setting up a remote history provider, see Distributing Applications in the InTouch® HMI Application Management and Extension Guide.

Note: You can also configure a IndustrialSQL Server history provider to visualize historical trend data. For more versatility and other charting options, use the ActiveFactory trend tools to create trends with InTouch historical data saved to a IndustrialSQL Server database.

To configure which tags to display from a historical trend

1. Double-click the trend object in the window. The **Historical Trend Configuration** dialog box appears.



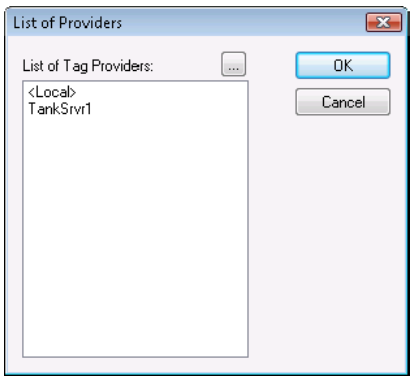
2. In the **Historical Tag box**, type the tag that you want to use for the trend.

   The tag must be defined as a Hist Trend type. You must assign a different Hist Trend tag to each historical trend in an InTouch application.

   If the tag you enter is not currently defined in the Tagname Dictionary, a dialog box appears and asks if you want to create a tag. If you select **OK** to define a tag, the **Tagname Dictionary** dialog box appears.

3. In the **Tagname** area, specify the name of an existing local or remote tag in one or more **Pen** boxes.
4. To assign an existing local or remote tag directly, click in a **Pen** box and type the name of the tag.
5. To browse to the tag to assign:

   a. Double-click in a **Pen** box. The **List of Providers** dialog box appears.



   b. Select the tag provider you want to use for the pen.

   c. Click **OK** to show a dialog box listing the tags for the selected provider.

d. Double-click on a tag from the list to select it.

6. Double-click the color box next to each pen assigned a tag to show the color palette. Click the color for the pen.
7. In the **Width** box, type the line width in pixels for each pen shown in the trend.
8. Repeat steps 3 to 7 for each tag that you want to assign to a historical trend pen.
9. If needed, select the **Allow runtime changes** check box to allow an operator to configure a historical trend while the application is running.

   For more information about updating a historical trend during run time, see *Changing the Trend Configuration at Run Time* on page 145.

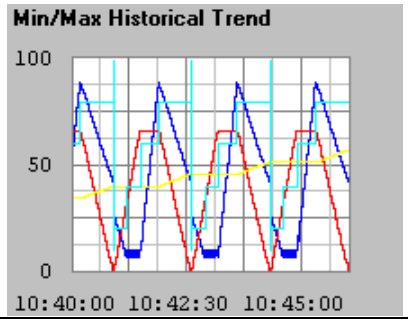
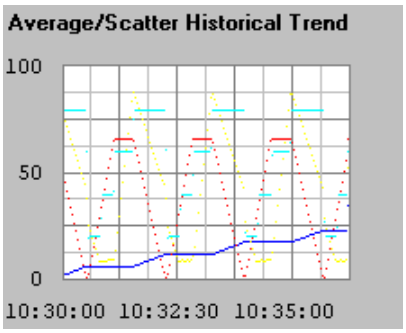
Configuring the Time Span of a Historical Trend

You can configure the time span of a historical trend.

To configure the time span of a historical trend

1. If needed, double-click on the trend object to show the **Historical Trend Configuration** dialog box.
2. In the **Chart Time** area, type the length of time in **Initial Time Span** that you want to appear on the horizontal x-axis of the trend.
3. Select the time unit of measure: Seconds (Secs), Minutes (Mins), Hours (Hrs), or Days (Days).

   For example, it you enter 8 in **Initial Time Span** and then select **Hrs**, the time span shown on the trend is 8 hours.
4. In the **Initial Display Mode** area, select the type of historical trend that appears when WindowViewer initially shows the window containing the trend.

Initial Display Mode	Description
Min/Max	The chart shows changes in the percentage of engineering units scale as a vertical line over the time span. 
Average	Each pixel within a trend time segment shows the average value of a tag over the period of time within the segment. 

5. Go to *Configuring Historical Trend Display Options* on page 144 to configure the visual appearance of a historical trend.

Configuring Historical Trend Display Options

You can configure the visual appearance of a historical trend.

To configure historical trend display options

1. If needed, double-click the trend object. The **Historical Trend Configuration** dialog box appears.
2. Set the color options. Do the following:
   - In the **Color** area, click the **Chart Color** box to open the color palette.
   - Click a color in the palette as the background for the trend.

     White is the default background color. Any other background color increases the time needed to print a trend.
   - Select **Border Color** to open the color palette.
   - Click a color in the palette as the border color of the trend.
3. Set the time divisions options. Do the following:
   - In the **Time Divisions** area, type the number of major trend time divisions in **Number of Major Div.**

     The major time divisions appear on the horizontal time axis of the trend. The maximum time between major time divisions is 65536 seconds, or 18 hours, 12 minutes, 16 seconds.
   - Select the color for the major division lines.
   - From **Minor Div/Major Div**, type the number of minor time divisions within each major time division.

     The number of minor time divisions should be evenly divisible within a major division. For example, if the major division is set to 60 seconds, entering a value of 2 in **Minor Div/Major Div** sets the minor time division to 30 seconds.
   - Select the color for the minor division lines.
   - Select **Top Labels** or **Bottom Labels** to specify the placement of time labels on the trend.
   - If you are using time labels, type the number of major time division lines per time label in the **Major Div/Time Label** box.
   - Select the color of the time division labels.
   - Select the time units shown as the label of the major time division.

Months (MM)	Hours (HH)
Days (DD)	Minutes (MM)
Years (YY)	Seconds (SS)

4. In the **Value Divisions** area, configure the appearance of the vertical axis of the trend.

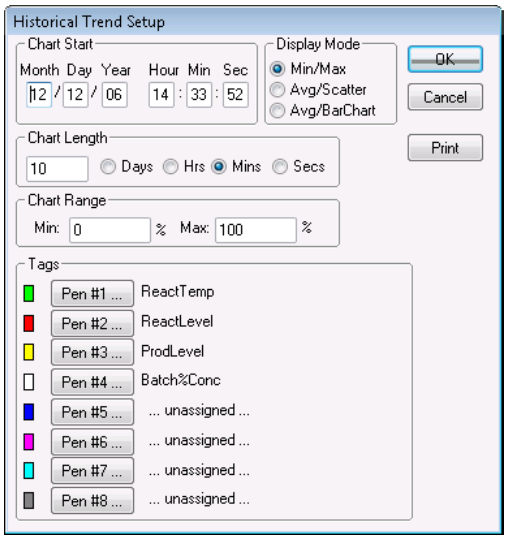
   Value Divisions options are configured the same way as the **Time Divisions** options. The vertical axis specifies the range of data values that appear in the trend based upon engineering units for all tags.
5. Click **OK** to save your configuration changes and close the **Historical Trend Configuration** dialog box.

Changing the Trend Configuration at Run Time

If you select the **Allow runtime changes** option when you configure your historical trend, operators can configure a historical trend while it is running. Operators configure the trend from a dialog box that appears after selecting the trend from a displayed window.

To configure a historical trend during run time

1. Click on the historical trend while it is running. The **Historical Trend Setup** dialog box appears.



2. In the **Chart Start** area, type the starting date and time of the historical trend data collection interval.
3. In the **Display Mode area, select the type of historical trend chart.**

   The trend display mode affects performance. The primary factor that determines trend performance is the length of the lines shown in a trend. The longer the lines, the longer it takes to generate the trend.

   Line width also affects performance. Wide lines take significantly longer to draw. Min/Max or Average/Scatter trends can be created more quickly than an Average/Bar Chart.
4. In the **Chart Length** area, type the duration to be displayed on the trend and then select the unit of measure.

   For example, if you enter 2 and select **Hrs**, the trend duration is 2 hours.
5. In the **Chart Range** area, type the percentage of engineering unit range shown on the vertical axis of the historical trend.

   The scale of the trend is a segment of the trended tag's engineering unit range defined by a percentage range. The values range from 0 to 100. For example, if you want to trend the variance of the selected tags from 40 to 60 percent of their engineering unit range, type 40 and 60 in the **Min** and **Max** range boxes, respectively.
6. In the **Tags** area, click a pen number to assign a tag.

   The **Select Tags** dialog box appears with a list of tags that can be assigned to the historical trend pen.
7. Use the following statement in a QuickScript or button to allow the operator to update the chart:

```
Hist_TrendTag.UpdateTrend = 1;
```

8. Use any of the following functions in a QuickScript or on a button:

   HTUpdateToCurrentTime(Hist_Tag);

HTScrollLeft(Hist_Tag,Percent);

HTScrollRight(Hist_Tag,Percent);

HTZoomIn(Hist_Tag,LockString);

HTZoomOut(Hist_Tag,LockString);

HTSetPenName(Hist_Tag,PenNum,Tagname);

For more information about using scripts containing trend functions, see *Controlling a Historical Trend Wizard Using Scripts* on page 160.

9. Change any of the following trend tag dotfields:

.ChartStart

.ChartLength

.MaxRange

.MinRange

.Pen1-.Pen8

For more information about using dotfields with historical trends, see *Controlling a Historical Trend Using Dotfields* on page 146.

Controlling a Historical Trend Using Dotfields

You can use dotfields to manage historical trends during run time.

.DisplayMode Dotfield

The .**DisplayMode** dotfield specifies the trend format used to display a tag's values.

Category

Historical

Usage

```
tag_name.DisplayMode
```

Parameter

tag_name
Any Hist Trend tag.

Data Type

Analog (read/write).

Valid Values

1 = Shows the min/max value that occurred in each sample period (default).

2 = Shows the average value of each sample period in a scatter historical trend.

3 = Shows the average of each sample period in a bar chart historical trend.

Example

This statement specifies that the values in the historical trend represented by the "HistTrend_Tag" are formatted as a bar chart historical trend.

```
HistTrend_Tag.DisplayMode=3;
```

See Also

.ChartLength, .ChartStart

.MinRange Dotfield

The .**MinRange** dotfield specifies the minimum percentage of the tag's engineering unit range to show for each tag in a Historical trend.

Category

Historical.

Usage

```
tag_name.MinRange
```

Parameter

tag_name
Any Hist Trend tag.

Remarks

A historical trend can show several types of tags at the same time. Specifying the minimum and maximum boundaries of the value range in engineering units is difficult because different types of tags can have different engineering ranges. Therefore, the minimum and maximum range values are expressed as a percentage of the engineering range of each tag. This way, regardless of the tag's true engineering range, the historical trend shows the indicated percentage of that tag's particular engineering range.

Data Type

Real (read/write).

Valid Values

The limits for the .MaxRange and .MinRange dotfields are from 0 to 100. .MinRange is always less than .MaxRange. If you assign a value less than 0 or greater than 100 to either of these dotfields, the value is clamped at 0 or 100. If .MinRange is greater than or equal to .MaxRange, the trend does not show any data.

Example

This example dotfield statement sets the minimum percentage range of the historical trend to 25 percent of the possible engineering unit range of a Hist Trend tag.

```
HistTrend.MinRange=25
```

See Also

.ChartStart, .ChartLength, .DisplayMode, .EngUnits, .MinEU, .MaxEU, .MaxRange, .MinRaw, .MaxRaw, .RawValue

.MaxRange Dotfield

The .**MaxRange** dotfield specifies the maximum percentage of the tag's engineering unit range to show for each tag in a Historical trend.

Category

Historical.

Usage

```
tag_name.MaxRange
```

Parameter

tag_name
Any Hist Trend tag.

Remarks

A historical trend can show many types of tags simultaneously. Specifying the minimum and maximum boundaries of the range in engineering units can be difficult because tags can have different engineering ranges. Therefore, the minimum and maximum range values are expressed as a percentage of the engineering range of each tag. This way, regardless of the tag's true engineering range, the historical trend shows the indicated percentage of that tag's engineering range.

Data Type

Real (read/write).

Valid Values

The limits for .MaxRange and .MinRange dotfields are from 0 to 100. .MinRange is always less than .MaxRange. If you assign a value less than 0 or greater than 100 to either of these dotfields, the value is clamped at 0 or 100. If .MinRange is greater than or equal to .MaxRange, the trend does not show any data.

Example

This example dotfield statement sets the maximum percentage range of the historical trend to 75 percent of the possible engineering unit range of a Hist Trend tag.

```
HistTrend.MaxRange=75
```

See Also

.ChartStart, .ChartLength, .DisplayMode, .EngUnits, .MinEU, .MaxEU, .MinRange, .MinRaw, .MaxRaw, .RawValue

.UpdateCount Dotfield

The .**UpdateCount** dotfield increments a count each time a historical trend is updated. The .**UpdateCount** dotfield can be used as a trigger for further functions.

Category

Historical.

Usage

```
HistTrendTag.UpdateCount
```

Parameter

HistTrendTag
HistTrend tag assigned the name of the trend.

Data Type

Integer (read-only).

Valid Values

Any positive integer.

Example

This example uses the **HTGetValueAtScooter()** function to retrieve the value of Pen1 at the right scooter's current position. A change to any function argument causes the function to be re-evaluated. When the update completes and the value of .**UpdateCount** is incremented, this statement is re-evaluated.

```
MyRealTag=HTGetValueAtScooter( MyHistTrendTag,MyHistTrendTag.UpdateCount, 2,
MyHistTrendTag.ScooterPosRight, 1, "PenValue");
```

See Also

.UpdateInProgress, .UpdateTrend

.UpdateInProgress Dotfield

The .**UpdateInProgress** dotfield indicates the current status of a historical trend update operation. The value of the dotfield is set to 1 if a historical retrieval is in progress; otherwise the dotfield is set to 0.

Category

Historical.

Usage

```
HistTrendTag.UpdateInProgress
```

Parameter

HistTrendTag
HistTrend tag assigned the name of the trend.

Remarks

Whenever new data is requested from the historical trend, this dotfield's value is set to 1. After the process completes, .**UpdateInProgress** is reset to 0. .**UpdateInProgress** can be used in functions related to historical trends.

If the operator scrolls the trend to a period outside the currently shown period, it can take some time to retrieve the historical data. The .**UpdateInProgress dotfield** provides a way to alert the operator that the requested data is being retrieved. Without feedback, the operator may not be aware that the trend is being updated.

Data Type

Discrete (read-only).

Value Values

0 = No update in progress

1 = Update in progress

Example

The .**UpdateInProgress** dotfield is typically used as the expression in a visibility link on a text object near or on the scroll buttons of a Historical Trend. You can use the .UpdateInProgress dotfield to show "Busy" on the window when the data is being retrieved with the following message value display animation link:

```
DText(HistTrend1.UpdateInProgress,"Busy","Ready")
```

See Also

.UpdateCount, .UpdateTrend

.UpdateTrend Dotfield

The **.UpdateTrend** dotfield triggers an update to a historical trend. Using the .**UpdateTrend** dotfield in a button action script, the operator can manually update the trend during run time.

Category

Historical.

Usage

```
HistTrendTag.UpdateTrend
```

Parameter

HistTrendTag
HistTrend tag assigned the name of the trend.

Remarks

Historical trends do not automatically update. A change must be made to either the .**ChartStart** or the .**ChartLength** dotfields to update the chart and show the current values for the specified tags. By using this dotfield in a button action script, the operator can update the chart during run time. You can also use this dotfield in a QuickScript if other dotfields associated with the historical trend are going to be changed.

You should only set the .UpdateTrend dotfield to a value of 1.

Data Type

Discrete (write only).

Valid Values

1

Example

This example triggers the historical trend associated with the **MyHistTrendTag** tag to update with the current values of all parameters.

```
MyHistTrendTag.UpdateTrend=1;
```

.ChartLength Dotfield

The .**ChartLength** dotfield specifies the length of time shown in a Historical trend.

Category

Historical.

Usage

```
HistTrendTag.ChartLength
```

Parameter

HistTrendTag
HistTrend tag assigned the name of the trend.

Remarks

The value assigned to .ChartLength specifies the length of the chart in seconds. The length is defined as the amount of time currently shown on the Historical Trend Chart. More specifically, the calculation retrieved as the Chart Length from a Historical Trend Chart is:

```
ChartLength=(Date/Time Stamp on Right-Hand Side of Chart) - (Date/Time Stamp on
Left-Hand Side of Chart);
```

Because Date/Time Stamps are expressed in seconds from midnight on January 1, 1970, the calculation results in seconds of time displayed between the left and right sides of the chart.

Whenever adding or subtracting from .**ChartLength**, time is expressed in seconds. Therefore, to subtract two hours from the current .**ChartLength,** convert hours to seconds before performing the calculation. For example:

```
(2 hours) * (60 minutes/hour) * (60 seconds/minute) = 7200 seconds.
```

Data Type

Integer (read/write).

Valid Values

Any positive integer.

Examples

This example forces the length of the historical trend to one hour.

```
HtTag.ChartLength=3600 {60 minutes * 60 seconds/minute};
```

This example scrolls the trend left by 50 percent.

```
HtTag.ChartStart=HtTag.ChartStart - HtTag.ChartLength / 2;
```

This example scrolls the chart left by 10 percent.

```
HtTag.ChartStart=HtTag.ChartStart - (.10 * HtTag.ChartLength);
```

See Also

.ChartStart

.ChartStart Dotfield

The .**ChartStart** dotfield can be used to set or verify the value of the starting (left side) date/time stamp of a historical trend.

Category

Historical.

Usage

```
HistTrendTag.ChartStart
```

Parameter

HistTrendTag
HistTrend tag assigned the name of the trend.

Remarks

This read/write dotfield is used to set or verify the value of the starting date/time stamp of a historical trend. The .**ChartStart** dotfield is expressed as the number of elapsed seconds after midnight January 1, 1970. The starting point is defined as the first date/time stamp on a historical trend.

Data Type

Integer (read/write).

Valid Values

Any positive integer.

Example

The following statement scrolls the chart to the right by one minute.

```
HtTagname.ChartStart=HtTagname.ChartStart + 60;
```

See Also

.ChartLength

.Pen1-8 Dotfields

The .**Pen1-8** dotfields assign a logged tag to a historical trend pen.

Category

Historical

Usage

```
HistTrendTag{.Pen1 | .Pen2 | .Pen3 | .Pen4 | .Pen5 | .Pen6 | .Pen7 | .Pen8};
```

Parameter

HistTrendTag
HistTrend tag assigned the name of the trend.

Remarks

You assign tags to trend pens using the .Pen1-8 dotfields with the following format:

HistTrend.Pen*X* = Tag_Name.TagID

Where *X* is an integer 1 to 8.

It is recommended that you use the HTSetPenName() and HTGetPenName() functions if possible.

Note: Only local tags can be assigned to a .PenX dotfield. The provider.tag notation cannot be used. The provider.tag can be used only with the HTSetPenName() function.

A good reference to use when learning how these dotfields work is a historical trend wizard placed on the screen and broken apart.

Data Type

TagID (read/write).

Valid Values

This dotfield data type is type **TagID**. This means only the handle of a tag can be assigned to the **.Pen1-8** dotfields. You cannot directly assign the name of a tag to the **.Pen1-8** dotfields. You must associate the associated .**TagID** dotfield of a tag to a .**Pen1-8** dotfield using the following syntax:

```
HistTrendTag.Pen1=LoggedTag.TagID;
```

In general, a TagID type tag can be equated only to another TagID tag. It cannot be used with any other tag type unless the .**TagID** dotfield extension is added to the other tag.

Although the .**Pen1-8** dotfields are considered read/write, their values cannot be directly shown on the screen.

Examples

The following example assigns a new tag to the .**Pen5** dotfield of the historical trend associated with the Hist Trend tag. You must append the .**TagID** dotfield to the name of the logged tag in order to assign it to .**Pen5** dotfield.

```
HistTrendTag.Pen5=PumpPress.TagID;
```

Working from the previous example, you can show the name of the tag assigned to HistTrendTag.Pen5. Creating a legend that shows the tag assigned to each trend pen is useful information for an operator.

You cannot show the tag assigned to HistTrendTag.Pen5 in a Message Display link. The actual value of the .Pen5 dotfield is an integer that represents a memory location within WindowViewer, which is not useful for display purposes. You need to create a new TagID type tag called **Pen05**. Place the following statement underneath the statement from the previous example:

```
Pen05=HistTrendTag.Pen5;
```

In the first example, the PumpPress tag is assigned to pen 5 of HistTrendTag. In this example, Pen05 is assigned the value of Pen5 of the HistTrendTag, which is the TagID of the PumpPress tag.

The .Pen1-8 dotfields are pointers to the tags that are associated with pens selected to appear in a trend. The .Pen1-8 dotfields are of a special data type, namely .TagID. After you make the assignment, you can use the .Name dotfield of the TagID tag to show the name of the tag.

.TagID Dotfield

The .**TagID** dotfield can be used in conjunction with the .**Pen1** - .**Pen8** dotfields to assign a tag to a historical trend pen.

Category

Historical tag.

Usage

```
tag_name.TagID
```

Parameter

tag_name
Any discrete, integer, real, indirect discrete, or indirect analog tag.

Remarks

The .**TagID** dotfield provides the handle of a tag and is used mainly to assign tags to pens in a historical trend.

Data Type

TagID (read-only).

Example

This example uses the .**TagID** dotfield to assign the **PumpRPM** tag to pen 6 of the historical trend.

```
HistTrendTag.Pen6=PumpRPM.TagID;
```

See Also

.Pen1-.Pen8

.ScooterLockLeft Dotfield

The .**ScooterLockLeft** dotfield specifies whether an operator can move the right scooter further left than the left scooter's current position on the historical trend.

Category

Historical.

Usage

```
HistTrendTag.ScooterLockLeft
```

Parameter

HistTrendTag
HistTrend tag assigned the name of the trend.

Remarks

In general, you should prevent an operator from moving the right scooter further left than the left scooter's current position. When the left scooter is locked, it forces the right scooter position to be equal to the left scooter position whenever the right scooter overtakes the left scooter.

Data Type

Discrete (read/write).

Valid Values

0 = False. Right scooter can move further left than the left scooter’s current position on the historical trend

1 = True. Right scooter cannot move further left than the left scooter's current position on the historical trend.

Example

The following example prevents the right scooter from moving further left than the left scooter's current position on the historical trend.

```
HistTrendTag.ScooterLockLeft=1;
```

See Also

.ScooterPosRight, .ScooterPosLeft, .ScooterLockRight

.ScooterLockRight Dotfield

The .**ScooterLockRight** dotfield specifies whether an operator can move the left scooter further right than the right scooter's current position on the historical trend.

Category

Historical.

Usage

```
HistTrendTag.ScooterLockRight
```

Parameter

HistTrendTag
HistTrend tag assigned the name of the trend.

Remarks

In general, you should prevent the operator from moving the left scooter further right than the right scooter's current position. When the right scooter is locked, if forces the left scooter position to be equal to the right scooter position whenever the left scooter overtakes the right scooter.

Data Type

Discrete (read/write).

Valid Values

0 = False. Left scooter can move further right than the right scooter's current position on the historical trend.

1 = True. Left scooter cannot move further right than the right scooter's current position on the historical trend.

Example

The following example prevents the left scooter from moving further right than the right scooter's current position on the historical trend.

```
HistTrendTag.ScooterLockRight=1;
```

See Also

.ScooterPosRight, .ScooterPosLeft, .ScooterLockLeft

.ScooterPosLeft Dotfield

The **.ScooterPosLeft** dotfield dynamically controls the position of the left scooter on a historical trend.

Category

Historical

Usage

```
HistTrendTag.ScooterPosLeft
```

Parameter

HistTrendTag
HistTrend tag assigned the name of the trend.

Remarks

This read/write dotfield dynamically controls the position of the left scooter. You can use this dotfield in a QuickScript function to retrieve the current position of the left scooter, or you can assign a value to this dotfield to adjust the position of the left scooter to another location on the trend.

This dotfield is most often used in conjunction with the set of **HTGetValue()** functions. These functions must specify which historical trend is being queried, as well as the current position of the trend's scooters.

Data Type

Real (read/write).

Valid Values

0.0 to 1.0; where 0.0 is the extreme left of the historical trend chart, and 1.0 is the extreme right of the historical trend chart.

Examples

The following example repositions the left scooter. The left scooter moves to a location 34 percent of the chart's total length from the left side of the historical trend chart currently associated with the MyHistTrendTag tag.

```
MyHistTrendTag.ScooterPosLeft=.34;
```

In the following statement, the QuickScript **HTGetValueAtScooter()** function retrieves the value of pen 1 at the left scooter's current position. A change to any value within a function's argument list causes the function to be re-evaluated. Each time the position of the left scooter changes, this statement is re-evaluated.

```
MyRealTag=HTGetValueAtScooter (MyHistTrendTag,MyHistTrendTag.UpdateCount, 1,
MyHistTrendTag.ScooterPosLeft, 1, "PenValue");
```

See Also

.ScooterPosRight, .ScooterLockLeft, .ScooterLockRight

.ScooterPosRight Dotfield

The read/write .**ScooterPosRight** dotfield dynamically controls the position of the right scooter.

Category

Historical.

Usage

```
HistTrendTag.ScooterPosRight
```

Parameter

HistTrendTag
HistTrend tag assigned the name of the trend.

Remarks

This read/write dotfield dynamically controls the position of the right scooter. You can use this dotfield in a QuickScript function to retrieve the current position of the right scooter. You can also assign a value to this dotfield to move the right scooter to another location on the trend.

This dotfield is most often used in conjunction with the **HTGetValue()** functions. These functions must specify which historical trend is being queried, as well as the current position of the trend's scooters.

Data Type

Real (read/write)

Valid Values

0.0 to 1.0; where 0.0 is the extreme left of the historical trend chart, and 1.0 is at the extreme right the historical trend chart.

Examples

The following statement specifies a new location for the right scooter. The right scooter moves to a location 34 percent of the chart's total length from the left side of the Historical Trend chart associated with the MyHistTrendTag tag.

```
MyHistTrendTag.ScooterPosRight=.34;
```

The following statement uses the **HTGetValueAtScooter()** QuickScript function to retrieve the value of pen 1 at the right scooter's new position. A change to any variable within a function's parameter list causes the function to be re-evaluated. Each time the position of the right scooter changes, this statement is re-evaluated.

```
MyRealTag=HTGetValueAtScooter(MyHistTrendTag, MyHistTrendTag.UpdateCount, 2,
MyHistTrendTag.ScooterPosRight, 1, "PenValue");
```

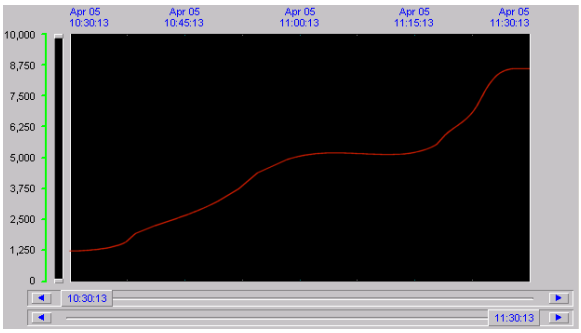
See Also

.ScooterPosLeft,. ScooterLockLeft, .ScooterLockRight

Using the Historical Trend Wizard

The Historical Trend Wizard automatically creates a historical trend. Other than manually assigning tags to historical trend pens, the wizard automatically configures the historical trend using standard values.

The figure below shows the standard trend created with the Historical Trend Wizard. The trend includes sliders called scooters to show data at a specific location on the trend plot or zoom in on a selected range of trend data.



To add zoom and movement functions or pen controls to a historical trend, use the trend Zoom/Pan Panel and Trend Pen Legend wizards.

You can create and configure a historical trend. You can:

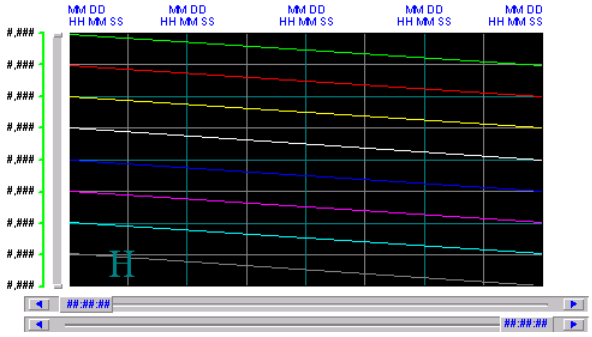
- Create a historical trend with wizards
- Select the tags for a trend
- Configure the historical trend time span
- Controlling a trend with QuickScripts

Creating a Trend With the Historical Trend Wizard

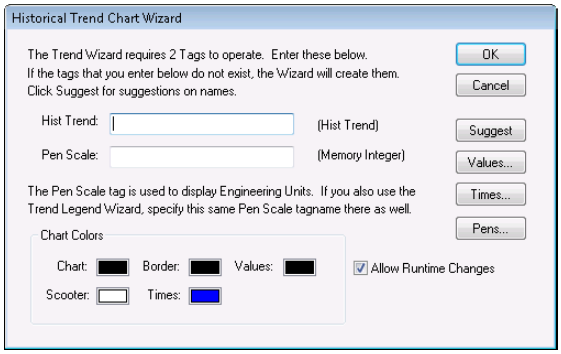
You can create a standard historical trend using the automation features of the Historical Trend Wizard. Other sections describe how to manually configure a historical trend by using wizard options.

To create a historical trend with wizards

1. Open a window from WindowMaker to place a historical trend.
2. On the WindowMaker menu bar, click the **Wizard** icon. The **Wizard Selection** dialog box appears.
3. Select **Trends** from the list of wizards. The right pane of the **Wizard Selection** dialog box shows a set of trend wizard icons.
4. Select the **Hist Trend with Scooters** wizard and click **OK**. The **Wizard Selection** dialog box closes and your window reappears.
5. Move the cursor to the window location where you want to place the upper left corner of the historical trend. Click to place the trend in the window.



6. Double-click the trend. The **Historical Trend Chart Wizard** dialog box appears.



7. Click **Suggest**. The Historical Trend Chart Wizard automatically assigns default configuration values to the trend.

   The only remaining configuration task is to assign tags to the trend pens.

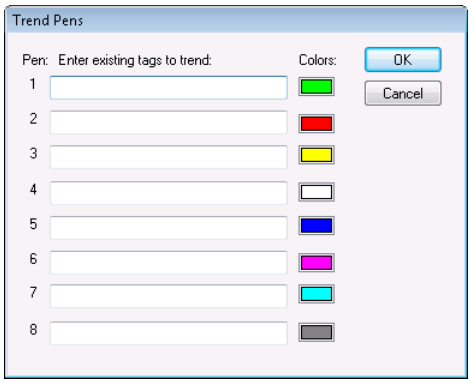
Configuring Which Tags to Display on the Trend Graph

Assigning tags to trend pens in the Historical Trend Chart Wizard is similar to the procedure for the Historical and Real-Time tools.

To assign tags from the Historical Trend Chart Wizard

1. If needed, double-click the historical trend. The **Historical Trend Chart Wizard** dialog box appears.

2. Click **Pens**. The **Trend Pens** dialog box appears.



3. Enter the name of an existing local tag in the **Pen** box. You can enter a maximum of 49 characters.

Note: WindowViewer must be closed. Otherwise, the Pen boxes cannot be selected.

If you double-click within the **Pen** box, the **Select Tag** dialog box appears with a list of tags assigned the **Log Data** option for the application. You can assign a tag to the pen by selecting it from the **Select Tag** dialog box.

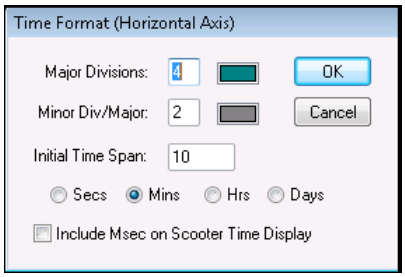
4. Click the color box next to each pen and select another color if you want to change the default pen color. Otherwise, skip this step and accept the default color.
5. Click OK to close the **Trend Pens** dialog box.
6. Click **OK** to close the **Historical Trend Chart Wizard** dialog box.

Configuring the Historical Trend Time Span

The **Historical Trend Chart Wizard** dialog box includes an option to manually configure the time span shown from a trend created with the Historical Trend Wizard. You can manually configure a trend's time instead of accepting the default configuration of the Historical Trend Wizard.

To configure the time span of a historical trend

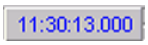
1. Double-click the historical trend. The **Historical Trend Chart Wizard** dialog box appears.
2. Click **Times**. The **Time Format** dialog box appears.



3. Configure the time format. Do the following:
   a. In the **Major Divisions box**, type the number of major time divisions shown on the horizontal time axis of the trend.
   b. In the **Minor Div/Major** box, type the number of minor time divisions within each major division.
   c. In the **Initial Time Span** box, type the length of the time period shown on the horizontal axis of the trend. Trends created with the Historical Trend Wizard can be updated while the application is running in WindowViewer. Operators can change the length of the trend time period. But a historical trend always starts with the time period set from the **Time Format** dialog box.

d. Select the unit of measure for the trend time period in seconds, minutes, hours, and days.

e. Optionally, include milliseconds in the scooter time display. The following example shows a scooter slider with milliseconds appended to the current time.



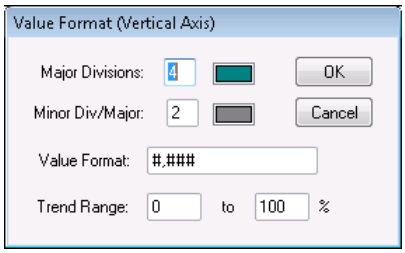
4. Click **OK** to close the **Time Format** dialog box.
5. Click **OK** to close the **Historical Trend Chart Wizard** dialog box.

Configuring Display Options

The **Historical Trend Chart Wizard** dialog box includes an option to configure the vertical units of a historical trend. You can manually configure the major and minor value divisions shown on the vertical axis of a trend.

To configure display options with the Historical Trend Chart Wizard

1. If necessary, double-click on the historical trend. The **Historical Trend Chart Wizard** dialog box appears.
2. Click **Values.** The **Value Format** dialog box appears with options to configure the vertical value axis of the trend.



3. Configure the value format. Do the following:

a. In the **Major Divisions** box, type the number of major value divisions shown on the trend's vertical axis. Click the color box to access the color palette, and then click the color that you assign to the major value axis division lines.

b. In the **Minor Div/Major** box, type the number of minor divisions that you want to be visible within each major value axis division. Click the color box to access the color palette, and then click the color that you assign to the minor value axis division lines.

c. In the **Value Format** box, type the format of numbers that appear in the trend's vertical value axis. The default number format is #,###.

d. In the **Trend Range** boxes, type the lower and upper percentage boundaries of a tag's engineering units that appear in the trend.

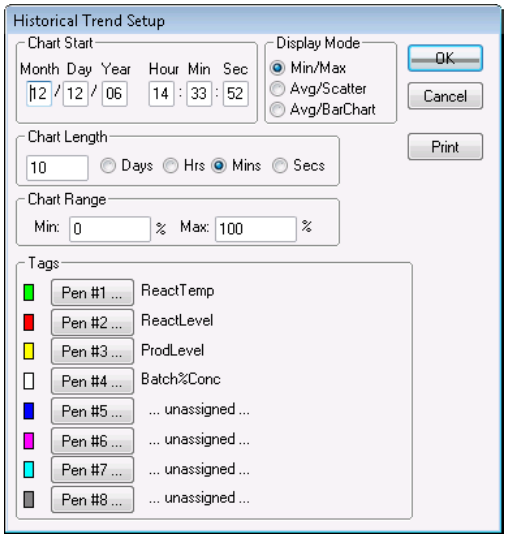
4. Click **OK** to close the **Value Format** dialog box.
5. Click **OK** to close the **Historical Trend Chart Wizard** dialog box.

Changing the Configuration at Run Time

If you select the **Allow runtime changes** option when you configure your historical trend, operators can change some aspects of a historical trend during run time.

To configure a historical trend during run time

1. Click the trend in WindowViewer. The **Historical Trend Setup** dialog box appears



2. In the **Chart Start** area, type the starting date and time of the chart.
3. In the **Display Mode** area, select the type of historical trend.
4. In the **Chart Length** area, type the length of time to show on the trend, and then select the time increment for the length.
5. In the **Chart Range** area, type the percentage of engineering unit scale shown as the vertical range of the trend.
6. In the **Tags** area, click each **Pen#** to assign a tag to the trend pen. The **Select Tag** dialog box appears and shows those tags for which logging is enabled.
7. Double-click on the name of a tag to assign it to the trend pen.
8. Click **OK** to save your run-time changes to the trend.

Controlling a Historical Trend Wizard Using Scripts

You can use QuickScript functions with trend objects or animation link expressions to control a historical trend during run time. For example, you can use QuickScripts to update a trend to the current time, reassign tags to trend pens, connect pens to the chart, replot the grid, and remove or replot the scooters.

Updating the Trend to the Current Time

You can create a script to update a historical trend to show recent tag data.

HTUpdateToCurrentTime() Function

The HT**UpdateToCurrentTime()** function retrieves and shows the data with an end time equal to the current time. The start time is equal to end time minus the width of the chart.

Category

Historical

Syntax

```
HTUpdateToCurrentTime(Hist_Tag);
```

Argument

Hist_Tag
HistTrend tag assigned the name of the historical trend.

Example

The following statement retrieves and shows data for the **Trend1** historical tag at the current time:

```
HTUpdateToCurrentTime("Trend1");
```

If the current time is 3:04 PM and the width of the trend is 60 seconds, the new end time is 3:04 PM. The new trend start time is 3:03 PM.

Changing the Trend Configuration

You can use these script functions to change the tags assigned to the pens of a historical trend:

- *HTSelectTag() Function* on page 161
- *HTSetPenName() Function* on page 161

HTSelectTag() Function

The **HTSelectTag()** function opens the **Select Tag** dialog box for the operator to assign a different tag to a trend pen.

Note: The **Select Tag** dialog box only lists the tags that are defined for historical logging with the **Log Data** option selected in the Tagname Dictionary.

Category

Historical

Syntax

```
HTSelectTag();
```

Remarks

The HTSelectTag() function only shows tags in which the **Log Data** option has been selected from the Tagname Dictionary. However, it is possible to use the Tag Browser's filter to display a smaller set of tags. For example all tags that begin with "A". The function returns the selected tag and can be used as function parameter to assign a tag to a pen.

Example

The following QuickScript causes the **Select Tag** dialog box to appear in WindowViewer. The user can then select a tag from the list. This tag is assigned to pen 1 by the Historical Object named HistTrend.

```
HTSetPenName("HistTrend",1,HTSelectTag());
```

See Also

HTSetPenName()

HTSetPenName() Function

The **HTSetPenName()** function assigns a different tag to a trend's pen.

Category

Historical

Syntax

```
HTSetPenName(Hist_Tag,PenNum,Tagname);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

PenNum
Integer tag or value representing the pen number (1-8) of the trend.

Tagname
Name of the new tag to assign to the pen.

Remarks

This QuickScript function is the only method to add tags from a distributed history provider during run time.

You can enter a maximum of 49 characters for a reference in a pen name.

You may see the following error message when you attempt to unassign a trend pen:

```
VIEW /UpdateData: Invalid DBS.TAGNAME handle - 0
```

This error occurs if you're trying to unassign a pen that was previously assigned to a remote tag in the form *histprovider.tag_name*. To resolve this error, create a local tag with the **Log Data** option selected. Then, use the following script to unassign the pen:

```
HTSetPenName( "HistTrend", 1, "localtag" );
{assigns the pen to a locally logged tag---localtag}
HistTrend.Pen1=None;
{unassigns the pen}
```

Where None is a TagID type tag.

Examples

The following statement assigns the OutletPressure tag to pen 3 of Trend1.

```
HTSetPenName("Trend1",3,"OutletPressure");
```

The following statement assigns the **HistPrv1.Tag1** tag to TrendPen4 of Trend1.

```
HTSetPenName("Trend1",TrendPen4,"HistPrv1.Tag1");
```

See Also

HTSelectTag()

Retrieving Information About the Trendand Historical Data

You can create scripts that retrieve information from a historical trend while it is running. Use the following functions:

- *HTGetPenName() Function* on page 162
- *HTGetTimeAtScooter() Function* on page 163
- *HTGetTimeStringAtScooter() Function* on page 164
- *HTGetValue() Function* on page 164
- *HTGetValueAtScooter() Function* on page 165
- *HTGetValueAtZone() Function* on page 166
- *HTScrollLeft() Function* on page 167
- *HTScrollRight() Function* on page 167
- *HTZoomIn() Function* on page 168
- *HTZoomOut() Function*

HTGetPenName() Function

The **HTGetPenName()** function returns the name of the tag currently assigned to the specified pen number of the historical trend.

Category

Historical

Syntax

```
MessageResult=HTGetPenName(Hist_Tag,UpdateCount, PenNum);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

UpdateCount
Integer representing the trend's **.UpdateCount** dotfield. The argument value acts as a data change trigger to re-evaluate the function

PenNum
Integer tag or value representing the pen number (1-8) of the trend.

Example

The following statement retrieves the name of the tag assigned to Pen 2 of the Trend1 trend and places the name in the **TrendPen** message tag:

```
TrendPen=HTGetPenName("Trend1", Trend1.UpdateCount,2);
```

HTGetTimeAtScooter() Function

The **HTGetTimeAtScooter()** returns the time in seconds after 00:00:00 hours GMT, January 1, 1970 for the sample at the scooter location specified by the *ScootNum* and *ScootLoc* arguments.

Category

Historical

Syntax

```
IntegerResult=HTGetTimeAtScooter(Hist_Tag, UpdateCount,ScootNum,ScootLoc);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

UpdateCount
Integer representing the trend's **.UpdateCount** dotfield.

ScootNum
Integer representing the left or right scooter:
1=Left Scooter
2=Right Scooter

ScootLoc
Real number representing the value at the .ScooterPosRight or .ScooterPosLeft positions on the trend.

Remarks

Any changes to the values assigned to the UpdateCount, ScootNum, and ScootLoc arguments cause the expression to be evaluated. This ensures the expression is evaluated after new data retrievals or after a scooter is moved.

Example

The following statement retrieves the time in seconds for the value at the current left scooter location of the Trend1 trend:

```
HTGetTimeAtScooter("Trend1",Trend1.UpdateCount,1, Trend1.ScooterPosLeft);
```

HTGetTimeStringAtScooter() Function

The **HTGetTimeStringAtScooter()** function returns the string containing the time/date for the sample at the specified scooter location.

Category

Historical

Syntax

```
MessageResult=HTGetTimeStringAtScooter(Hist_Tag, UpdateCount, ScootNum,
ScootLoc, Format_Text);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

UpdateCount
Integer representing the trend's **.UpdateCount** dotfield.

ScootNum
Integer representing the left or right scooter:
1=Left Scooter
2=Right Scooter

ScootLoc
Real number representing the value at the .ScooterPosRight or .ScooterPosLeft positions on the trend.

Format_Text
String specifying the time/date format to use. The following Format_Text strings are acceptable: "Date", "Time", "DateTime", "DOWShort" (Wed, for example), and "DOWLong" (Wednesday, for example).

Remarks

Any changes to the values assigned to the UpdateCount, ScootNum, and ScootLoc arguments cause the expression to be evaluated. This ensures the expression is evaluated after new data retrievals or after a scooter is moved. The format of the string determines the contents of the return value.

Example

The following statement retrieves the date and time for the value at the current scooter location for the right scooter of the Trend1 trend. The value is stored in the NewRightTimeString message tag and is in "Time" format:

```
NewRightTimeString=HTGetTimeStringAtScooter ("Trend1",Trend1.UpdateCount,2,
Trend1.ScooterPosRight,"Time");
```

HTGetValue() Function

The **HTGetValue()** function returns a value of the requested type for the trend's specified pen.

Category

Historical

Syntax

```
RealResult=HTGetValue(Hist_Tag,UpdateCount, PenNum,ValType_Text);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

UpdateCount
Integer representing the trend's **.UpdateCount** dotfield.

PenNum
Integer tag or value representing the pen number (1-8) of the trend.

ValType_Text
String indicating the type of value to return:

PenAverageValue = Average for the entire trend.

PenMaxValue = Maximum pen value for the entire trend.

PenMinValue = Minimum pen value for the entire trend.

PenMaxEU = Maximum engineering units value for the entire trend.

PenMinEU = Minimum engineering units value for the entire trend.

PenStdDev = Standard deviation for the entire trend.

Remarks

The function returns the requested value as a real value.

Example

The following statement obtains the standard deviation for the pen 2 data retrieved from the PumpPress trend. The value is stored in the LeftHemisphereSD memory real tag:

```
LeftHemisphereSD=HTGetValue("PumpPress",
PumpPress.UpdateCount,2,"PenStdDev");
```

HTGetValueAtScooter() Function

The HTGetValueAtScooter() function returns a value of the requested type for the sample at the specified scooter position, trend, and pen number. The UpdateCount argument causes the expression to be evaluated after function processing is finished.

Category

Historical

Syntax

```
RealResult=HTGetValueAtScooter(Hist_Tag,
UpdateCount,ScootNum,ScootLoc,PenNum, ValType_Text);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

UpdateCount
Integer representing the trend's **.UpdateCount** dotfield.

ScootNum
Integer representing the left or right scooter:
1 = Left Scooter
2 = Right Scooter

ScootLoc
Real number representing the trend's .ScooterPosRight or .ScooterPosLeft dotfields.

PenNum
Integer tag or value representing the pen number (1-8).

ValType_Text
String indicating the type of value to return:

PenValue = Value at scooter position.
PenValid = 0 if value is invalid, 1 if valid.

When the ValType_Text argument is used with the HTGetValueAtScooter() function, use one of the valid types listed.

Example

The following function returns a 1 if the value is an actual sample or a 0 if the value is invalid for pen 3 of the Trend1 trend for the right scooter's current position:

```
HTGetValueAtScooter("Trend1",Trend1.UpdateCount, 2,Trend1.ScooterPosRight,3,
"PenValid");
```

HTGetValueAtZone() Function

The HTGetValueAtZone() function returns a value of the requested type for the data located between the right and left scooter positions for a trend's specified pen.

Category

Historical

Syntax

```
RealResult=HTGetValueAtZone(Hist_Tag,UpdateCount,
Scoot1Loc,Scoot2Loc,PenNum,ValType_Text);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

UpdateCount
Integer representing the trend's **.UpdateCount** dotfield. It is used only as a trigger to evaluate the function.

Scoot1Loc
Real representing the trend's .ScooterPosLeft dotfield. It is used only as a trigger to evaluate the function.

Scoot2Loc
Real representing the trend's .ScooterPosRight dotfield. It is used only as a trigger to evaluate the function.

PenNum
Integer tag or value representing the pen number (1-8) of the trend.

ValType_Text
String indicating the type of value to return.
PenAverageValue = Average for zone between the scooters.
PenMaxValue = Maximum value for the zone between the scooters.
PenMinValue = Minimum value for the zone between the scooters.
PenMaxEU = Maximum engineering unit value for the zone between scooters.
PenMinEU = Minimum engineering unit value for the zone between the scooters.
PenStdDev = Standard Deviation for the zone between the scooters.

Remarks

A real value is returned representing the calculated value of the given type. Specifying constant values for the Scoot1Loc and Scoot2Loc arguments has no effect and are only used to trigger the evaluation of the function. The function uses the trend tag's .ScooterPosLeft and .ScooterPosRight dotfield values directly, regardless of the values you specify for the Scoot1Loc and Scoot2Loc arguments.

Example

The following statement calculates the average value for data between the right and left scooters of the Trend1 trend for pen 1. The value is stored in the AvgValue memory real tag:

```
AvgValue=HTGetValueAtZone("Trend1",
Trend1.UpdateCount,Trend1.ScooterPosLeft,
Trend1.ScooterPosRight,1,"PenAverageValue");
```

Panning and Zooming the Trend

You can create QuickScripts containing functions that select specific data from a historical trend during run time.

HTScrollLeft() Function

The **HTScrollLeft()** function sets the start time of the trend to an earlier time than the current start time by a percentage of the trend's total time span. The effect is to scroll the chart to the left to an earlier time by a specified percentage of the trend's total time span.

Category

Historical

Syntax

```
HTScrollLeft(Hist_Tag,Percent);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

Percent
Real number representing the percentage of the chart's time span to scroll (0.0 to 100.0) left.

Example

The following statement scrolls the time/date left by 10 percent of the PumpPress trend's total width:

```
HTScrollLeft("PumpPress",10.0);
```

If the current display starts at 12:00:00 PM and the display width is 60 seconds, then the new trend starts at 11:59:54 AM after the function is processed.

HTScrollRight() Function

The **HTScrollRight()** function sets the start time of the trend to a time later than the current start time by a percentage of the trend's width. The effect is to scroll the date/time of chart to the right by a specified percentage of the trend's width.

Category

Historical

Syntax

```
HTScrollRight(Hist_Tag,Percent);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

Percent
Real number representing the percentage of the chart to scroll (0.0 to 100.0) right.

Example

The following statement scrolls the PumpPress trend to the right by 20 percent.

```
HTScrollRight("PumpPress",20.0);
```

If the current display starts at 12:00:00 PM and the display width is 60 seconds, then the new trend starts at 12:00:12 PM after the function is processed.

HTZoomIn() Function

The HTZoomIn() function calculates a new chart width and start time. If the trend's scooters are at the left and right sides of the trend, then the new chart width equals the old chart width divided by two. The new start time is calculated based on the value of the *LockString* argument.

If the scooters are not at the left and right sides of the trend, the HTZoomIn() function zooms the trend to the zone defined by the scooters and ignores the *LockString* argument.

Category

Historical

Syntax

```
HTZoomIn(Hist_Tag,LockString);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

LockString
String representing the type of zoom:

StartTime	Keep the start time equal to before zoom
Center	Keep center time equal to before zoom
EndTime	Keep end time equal to before zoom

Remarks

If the scooter positions are not at the left and right sides of the trend, the new chart width is the time between .ScooterPosLeft and .ScooterPosRight positions. In this case, the value of LockString is not used. The minimum chart width is one second. The scooter positions are set to .ScooterPosLeft=0.0 and .ScooterPosRight=1.0 after the zoom.

Example

The following statement zooms the display by a factor of two and maintains the same start time for the Trend1 trend. Trend1.ScooterPosRight is equal to 1.0 and Trend1.ScooterPosLeft is equal to 0.0. If the start time before zooming was 1:25:00 PM and the chart width was 30 seconds, the new start time remains at 1:25:00. The new chart width is 15 seconds.

```
HTZoomIn("Trend1","StartTime");
```

HTZoomOut() Function

The **HTZoomOut()** function calculates a new chart width and start time. The new chart width is the old chart width multiplied by two. The new start time is calculated based on the value of the *LockString* argument .

Category

Historical

Syntax

```
HTZoomOut(Hist_Tag,LockString);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

LockString
String representing the type of zoom:
StartTime = Keep start time equal to before zoom
Center = Keep center time equal to before zoom
EndTime = Keep end time equal to before zoom

Remarks

The current scooter positions have no effect on HTZoomOut(). After the function zoom finishes, the new scooter positions are set to .ScooterPosLeft=0.0 and .ScooterPosRight=1.0.

Example

The following statement zooms out the trend time by a factor of two and maintains the same center time for the Volume trend. If the start time before zooming was 2:15:00 PM and the chart width was 30 seconds, the start time after zooming is now 2:14:45. The chart width is 60 seconds and the center of the trend remains at 2:15:15.

```
HTZoomOut("Volume","Center");
```

Printing the Trend

You can print a historical trend currently visible from a WindowViewer screen using the PrintHT() function in a script.

PrintHT() Function

The **PrintHT()** function prints the historical trend currently visible on the screen. Usually, the **PrintHT()** function is associated with a screen button included on the historical trend window. Operators click the button to print the visible historical trend with its current values.

Category.

Historical

Syntax

```
PrintHT(Trend_Tag);
```

Argument

Trend_Tag
HistTrend tag.

Example

This example prints the PumpPress historical trend currently visible on the screen.

```
PrintHT(PumpPress);
```

Troubleshooting the Trend

You can create QuickScripts to verify if data was successfully retrieved that appears in a historical trend. Use the HTGetLastError() function to troubleshoot the trend.

HTGetLastError() Function

The **HTGetLastError()** function can be used in a script to determine if an error occurred during the last data retrieval for a specified historical trend pen.

Category

Historical

Syntax

```
[Result=]HTGetLastError(Hist_Tag,UpdateCount, PenNum);
```

Arguments

Hist_Tag
HistTrend tag assigned the name of the trend.

UpdateCount
Integer representing the trend's **.UpdateCount** dotfield.

PenNum
Integer tag or value representing the pen number (1-8) of the trend.

Result
Integer assigned to a tag that represents the status of the last script function call for the specified pen.

0 = No error

1 = General server error

2 = Old request

3 = File error

4 = Server not loaded

5 = Trend/Pen passed in function does not exist

6 = Trend tag does not exist in database

7 = Pen number passed to function is invalid (not in range of 1 to 8).

8 = No tag or a non-logged tag assigned to the pen number

Examples

The following statement retrieves the status of the last data retrieval for pen 3 of the Trend1 trend and assigns the result to the ResultCode integer tag.

```
[ResultCode=]HTGetLastError("Trend1", Trend1.UpdateCount,3);
```

In an animation Analog Value Display QuickScript the following statement would be used:

```
HTGetLastError("Trend1",Trend1.UpdateCount,3);
```

Displaying Real-Time Values in a Trend

You can create real-time trends by two methods. The Real-Time Trend object provides a standard set of controls to select the data, set a time range, and specify the physical appearance of the graph. InTouch also includes the 16-Pen Trend Wizard, which is an optional control to create real-time and historical trends. For more information about creating real-time trends with the 16-Pen Trend Wizard, see *Creating a 16-Pen Trend* in the InTouch® HMI Supplementary Components Guide.

Using Real-Time Trend Objects

You can create a real-time trend to show current values in your application.

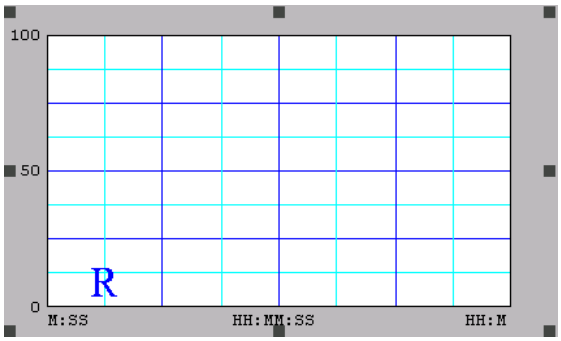
Creating a Real-Time Trend

You can use the Real-time Trend tool to create a trend object in a window. The first time you paste a real-time trend object, WindowMaker uses default settings. After configuring a real-time trend, WindowMaker uses the last configuration values as the initial settings for any new real-time trend object.

You can draw the trend chart any size within the borders of the window.

To create a real-time trend

1. Select the **Real-time Trend** tool from the Drawing Toolbar.
2. Move the mouse over the window area where you want to place the real-time trend. Drag the mouse diagonally to create a rectangle the desired size of the trend. The Real-time Trend object appears in the window.



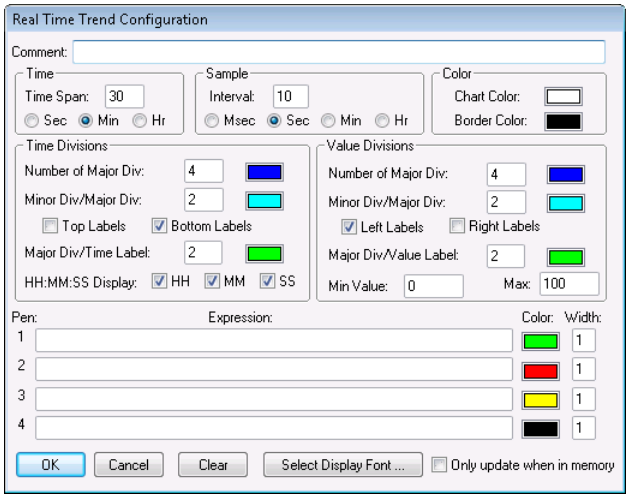
3. If needed, adjust the height and width of the trend with its object handles.

Configuring Which Tags to Display on a Real-time Trend

A real-time trend pen creates a graphical representation of current data from any local tag or an expression that contains one or more local tags. You configure the pens that show tag data in a real-time trend.

To configure real-time trend tags

1. Double-click the trend object in the window. The **Real Time Trend Configuration** dialog box appears.



2. In the **Expression** area, type the name of a local tag or expression that contains one or more local tags.

If you double-click in the **Pen** box, the **Select Tag** dialog box appears showing a list of tags defined for the application. You can assign a tag to the pen by selecting it from the **Select Tag** dialog box.

3. Click the color box next to each pen assigned a tag to show a color palette.
4. Click the color that you want to assign to the pen.
5. In the **Width** box, type the line width in pixels for each pen shown in the trend.

   Selecting a line width greater than 1 increases the time required to update or print a trend.
6. Select the **Only update when in memory** check box if you want the trend to update only when shown in the active window.

   If you do not select this option, the trend updates continuously even if the window is closed. Continuous trend updating slows system performance.
7. Keep the **Real Time Trend Configuration** dialog box open and go to the next procedure described in *Configuring the Real-Time Trend Time Span and Update Rate* on page 172.

Configuring the Real-Time Trend Time Span and Update Rate

You can configure the time span and update rate of a real-time trend.

To set a real-time trend time span and update rate

1. If needed, double-click the trend object. The **Real Time Trend Configuration** dialog box appears.
2. In the **Time** area, type the length of time in the **Time Span** box that you want to appear on the horizontal x-axis of the trend.
3. Select the unit of measure for trend time.
   - Seconds (Sec)
   - Minutes (Min)
   - Hours (Hr)

   For example, if you enter 30 in the **Time Span** box and then select **Min**, the time span shown on the chart is 30 minutes.
4. In the **Sample** area, type a number in the **Interval** box that the trend expression is evaluated and the chart updates.
5. Select the interval unit of measure.
   - Milliseconds (Msec)
   - Seconds (Sec)
   - Minutes (Min)
   - Hours (Hr)

   For example, if you enter 10 in **Interval** and then select **Sec,** the real-time trend is updated every 10 seconds.
6. Keep the **Real Time Trend Configuration** dialog box open and go to the next procedure described in *Configuring Real-time Trend Display Options* on page 172.

Configuring Real-time Trend Display Options

You can configure the visual appearance of a real-time trend.

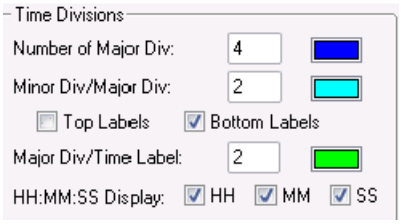
To configure real-time trend display options

1. Double-click on the trend object. The **Real Time Trend Configuration** dialog box appears.

2. In the **Color** area, configure the color. Do any of the following:
   - Click the **Chart Color** box to open the color palette. Select the background color for the trend.

     White is the default background color. Any other background color significantly increases the time needed to print a trend.
   - Click on the **Border Color** box to open the color palette. Select the border color of the trend.
3. In the **Time Divisions** area, configure the time divisions. Do the following:
   - In the **Number of Major Div box,** type the number of major trend time divisions The major time divisions appear on the horizontal time axis of the trend.

     The number of major time divisions must be an even multiple of the **Major Div/Time Label** value. For example, a division number of 20 is an even multiple of the **Major Div/Time Label** value of 4.

   - Select the color for the major division lines.
   - In the **Minor Div/Major Div** box, type the number of minor time divisions visible within each major time division.

     The number of minor time divisions should be evenly divisible within the major division period. For example, if the major division period is set to 60 seconds, entering a value of 2 in **Minor Div/Major Div** creates two minor time division periods of 30 seconds.
   - Select the color for the minor division lines.
   - Select either the **Top Labels** or **Bottom Labels** check box to specify the placement of time labels on the trend.

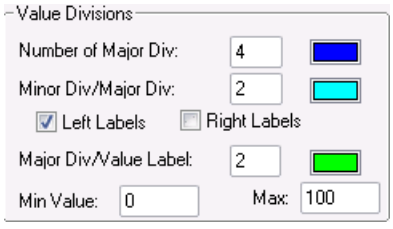
     You can select both options to place time labels at the top and bottom of the trend. Leaving both options blank removes time labels from the horizontal axis of the trend.
   - If you are using time labels, type the number of major time division lines per time label in **Major Div/Time Label**. The number of major divisions must be an even multiple of the **Major Div/Time Label** value.

     Select the color of the time division labels.
   - Select the time units shown as part of the major time division label.

     Hours (HH)

     Minutes (MM)

     Seconds (SS)
4. In the **Value Divisions** area, configure the appearance of the vertical axis of the trend.

Value Divisions options are configured like **Time Divisions** options. The major and minor divisions on the y-axis show the magnitude of data values rather than time. The vertical axis specifies the range of data values that appear in the trend based upon engineering units for all tags.

To show decimal points for the minor and major value divisions, type real numbers for the **Min Value** and **Max** options. For example, 0.00 to 100.00.

5. Click **Select Display Font**. The **Font** dialog box appears with options to set the font, style, and size of text that appears in the trend.
6. Click **OK**.

Printing a Trend at Run Time

Several factors determine how fast a trend can be printed. The primary factor is the size of the trend on the printed page. The display mode of the trend also affects printing performance. Min/Max or Average/Scatter trends can be printed more quickly than Average/Bar Chart trends. Also, the longer and wider the lines on the trend are, the longer it takes to print.

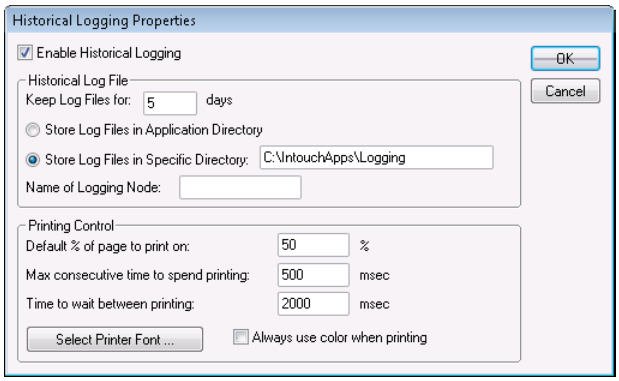
Another factor that affects printing performance is the background color of the trend. In most cases, a white background prints more quickly than a colored background.

Configuring Trend Printing Options

You can configure options that determine how a trend is printed.

To configure historical trend printing

1. On the **Special** menu, click **Configure** and then **Historical Logging**. The **Historical Logging Properties** dialog box appears.



2. In the **Printing Control** area, specify the percentage of the page to print the trend in the **Default % of page to print on** box.

   If you enter 50, the trend is printed on half of the page vertically and horizontally. A trend printed at 50 percent takes much less time to print than a full-page trend.

   As a printing alternative, you can use the **PrintWindow()** QuickScript function.
3. In the **Max consecutive time to spend printing** box, type the process time slice in milliseconds.

   A time slice represents the period allocated to the computer processor to run the print module process in the foreground and print the trend. A longer time slice enables the trend to be printed more quickly at the expense of other processes running on the computer.
4. In the **Time to wait between printing** box, type the time in milliseconds the print module waits between processor time slices.

   A shorter waiting period between processor time slices enables the trend to be printed more quickly.

5. Click **Select Printer Font.** The **Font** dialog box appears. Select the characteristics of the text appearing in a trend.
6. Click **OK** to save your printing configuration and close the **Historical Logging Properties** dialog box.

Displaying Historical Tag Values from Other InTouch Nodes or IndustrialSQL Server

If you want to use data stored remotely to create historical trends, the remote provider must be registered in the InTouch history provider list. This list specifies the name and network location of each history provider. These names are referred to whenever historical trend pens point to tags at the remote history provider.

You can define a remote history provider and assign historical trend pens to tag data stored at the remote location. You can:

- Configure remote history providers.
- Configure pens to display data from a remote history provider.
- Assign pens to tag data stored at a remote history provider using the Tag Browser.
- Assign a pen to a remote tag using a QuickScript.

For more information about using data from a remote history provider, see *Distributing Applications* in the InTouch® HMI Application Management and Extension Guide.

Using the InTouch HMI with the IndustrialSQL Server

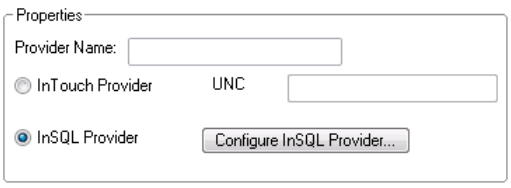
The IndustrialSQL Server is a real-time, relational database designed specifically for industrial applications. You can optionally configure the Historical Logger to store InTouch historical data to a IndustrialSQL Server database.

Note: For more information about logging InTouch historical data to a database, see the IndustrialSQL Server documentation. For more information about setting up the InTouch HMI with a remote history provider, see *Distributing Applications* in the InTouch® HMI Application Management and Extension Guide.

If you use the IndustrialSQL Server to store historical data, you must use the Distributed Name Manager from WindowMaker to specify a connection to the database.

To configure a connection to a **IndustrialSQL Server** database

1. Open WindowMaker.
2. On the Tools view, expand the **Configure list**.
3. **Select Distributed Name Manager**. T**he Distributed Name Manager** dialog box appears.
4. On the **Distributed History** tab, type **InSQL** as the new provider in the **Provider Name** box.
5. Select **InSQL Provider**.



6. Click **Configure InSQL Provider**. The **InSQL History Provider Properties** dialog box appears.

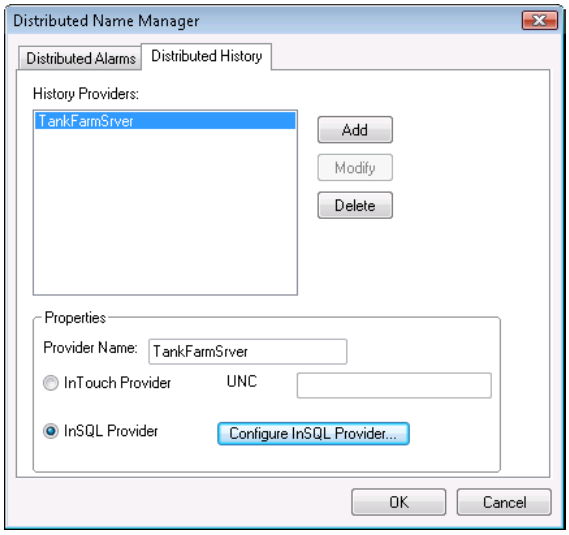
a. In the **Data Source** box, type the node name of the server where the IndustrialSQL Server is installed.

b. Enter a name for a IndustrialSQL Server database user account.

c. Enter the password for the user account in both the **Password** and **Re-enter password** boxes.

d. Click **Test** to verify the connection to the IndustrialSQL Server database. A message appears indicating whether the connection to the database is successful or not.

7. Click **OK** to close the dialog box.
8. Click **OK** to close the **InSQL History Providers** dialog box. The IndustrialSQL Server node appears in the **History Providers** list.



9. Click **OK** to close the **Distributed Name Manager** dialog box.

Configuring Pens to Display Remote Trend Data

Historical trends can display tag data from both local and remote history providers. You can assign trend pens to display data from a remote history provider.

To display a tag from a remote history provider

1. Double-click on the historical trend to show the **Historical Trend Configuration** dialog box.
2. In each pen's **Tagname** box, type the reference to a remote history provider. The format of the reference to a remote history provider is:

   history_provider_name.tag_name

   Example:

   ```
   TankFarm1.Pump1RPM
   ```

   Each pen of a historical trend can refer to a different remote history provider.
3. Click **OK** to save your configuration changes.

Note: The .TagID dotfield cannot be used in remote history provider tag references.

Using the Tag Browser to Assign Pens to Remote History Providers

The following procedure explains how to use the Tag Browser to assign a trend pen to tag data from a Remote History provider. Using the Tag Browser to select tags eliminates the need to manually enter each tag name and reduces the likelihood of errors.

The remote node name you specify in the Access Name does not have to be the actual name of the node where the tag resides. But, you must create an Access Name to define the remote history provider as a tag source. For more information about creating an Access Name, see *Setting Up Access Names* on page 59.

To define a remote history provider as a tag source

1. Create an Access Name that specifies the node name where the history provider is located.
2. Double-click the historical trend to open the **Historical Trend Configuration** dialog box.
3. Double-click a pen's **Tagname** input box to show the **Select Tag** dialog box.
4. Click **Define Tag Sources** to define the remote history provider as a tag source.
5. Click the **Tag Source** arrow and select the new remote history provider tag source in the list, or click the **Tree View** button and select the tag source in the tree view pane. The **Select Tag** dialog box refreshes and shows the tags from the selected remote history provider.
6. Select the tag that you want to assign to the historical pen and click **OK**. The **Historical Trend Configuration** dialog box reappears with the selected tag listed in the pen's **Tagname** box as: *AccessName*:*Item*.
7. Replace the AccessName: portion with the history provider name you defined in the **Distributed Name Manager**.

   For example, HistPrv1.*tag_name*

   This process may seem cumbersome, but after you have defined the history provider as a tag source in the Tag Browser, each time you double-click another tag input box, you simply double-click the tag name in the **Tag Browser**, and then replace the AccessName: portion with the history provider name.

In WindowViewer, if run-time changes are allowed for the historical trend, when the user clicks a pen button to change the tagname, the Tag Browser appears but only the local application's tags are accessible.

Using a QuickScript to Assign a Pen to a Remote History Provider

While an InTouch application is running, you can configure a trend pen to show tag data from a remote history provider. Create a QuickScript that specifies the remote history provider tag reference in the HTSetPenName() function. For example:

```
HTSetPenName("HistTrendTag", 1, "HistPrv1.Boiler1");
```

In this example, the number 1 specifies the number of the pen in the historical trend that plots the remote **Boiler1** tag values from the HistPrv1 remote history provider.

The run-time **Historical Trend Setup** dialog box and **Pen** dotfields are not supported for remote history providers.