

# **HMI - DATA LOGGING INTEGRATION ENHANCEMENTS**

**A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF**

**MASTER OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE**

**BY**

**SANDHYA BOLLINENI**



**DEPARTMENT OF COMPUTER & INFORMATION SCIENCES  
SCHOOL OF MATHEMATICS & COMPUTER/INFORMATION SCIENCES  
UNIVERSITY OF HYDERABAD  
HYDERABAD – 500 046, INDIA  
APRIL 2010**

## **CERTIFICATE**

This is to certify that the project work entitled “**HMI - Data Logging Integration Enhancements**” is being submitted by **Ms. Sandhya Bollineni** bearing Regd. No. **08MCMT12** in partial fulfillment of the requirements for the award of **Master of Technology in Computer Science** by **University of Hyderabad**, is a bona fide work carried out by her under our supervision and guidance.

The results embodied in this thesis have not been submitted to any other University or Institution for the award of any degree or diploma.

**Prof. Chakravarthy Bhagvati**  
Project Supervisor,  
Department of CIS,  
University of Hyderabad

**Mr. Anil Kumar NP**  
Tech Lead,  
GE – IIC,  
Hyderabad

**Prof. Arun Agarwal**  
Head of Department,  
Department of CIS,  
University of Hyderabad

**Prof. T. Amaranath**  
Dean,  
School of MCIS,  
University of Hyderabad

## **ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people and who made it possible and whose constant guidance and encouragement crowns all efforts with success.

I would like to express my deep sense of gratitude and whole-hearted thanks to my project guide **Dr. Chakravarthy Bhagvathi**, Professor, Dept. of CIS, University Of Hyderabad, for giving me the privilege of working under his esteemed guidance in carrying out this project work.

With great sense of pleasure and privilege, I express my deep sense of gratitude to my Guide **Mr. Anil Kumar NP** and Manager Sri **Nirmala Pelluri**, GE India Innovation Centre, Hyderabad for their valuable suggestions, sagacious guidance, scholarly advice and comprehensive critical remarks during the course of project.

I profoundly thank **Dr. Arun Agarwal**, Head, Dept. of CIS, University Of Hyderabad, for his help and encouragement.

I also extend my thanks to the entire **PME Development team** at GE India Innovation Centre, Hyderabad; and the **faculty of Dept. of CIS**, University Of Hyderabad, who have encouraged me throughout the course of Master's degree.

Lastly I thank my family, friends and all those helped me directly or indirectly for the completion of the project.

**Sandhya Bollineni**

## **Abstract**

Proficy Machine Edition (PME) HMI is the most powerful, easy-to-use supervisory monitoring and control software. It consolidates the collection of data from your facility sensors and devices like, PLCs (Programmable Logic Controllers), and then transforms the data into dynamic text, alarm and *graphic displays* and *Data Logging*. It gives users access to real-time information, helping them make appropriate decisions to improve quality, productivity and, ultimately, profitability.

This project is to integrate an enhanced trend tool (graph drawing tool) into Proficy Machine Edition's environment with additional trend enhancements. The tool enables reading data from different data sources with different scan rates supporting both real time and historic modes. After integrating the tool into PME, its features will be well exposed to the user and user can effectively analyze the trends with the provision of additional trend enhancements.

## List Of Figures:

Fig: 1.1.PME Environment

Fig: 1.2. Common PME tools

Fig: 2.1. Present trend tool

Fig: 2.2. Proposed Trend Tool

Fig: 4.1. Pugh Matrix

Fig: 4.2. Template of classes used in design

Fig: 4.3. Sequence of steps to trend

Fig: 5.1. Classes with fields shown

Fig: 6.1. Trend tool component in Navigator

Fig: 6.2. Right Click menus at component node

Fig: 6.3. Variable Selection/Reselection Window

Fig: 6.4. Group with desired variables

Fig: 6.5. Variable selection/Reselection window along with groups

Fig: 6.6. New trend tool in editor window of PME

Fig: 6.7. Common Plot Editor

Fig: 6.8. Plot with modified Trace Configuration, additional axis and multi column legend

Fig: 6.9. Plot window that is zoomed-in

Fig: 6.10. Plot Window with Add/Remove Data Cursor Dialog and Data Cursor added

## **List of tables:**

Table: 6.1. Test Cases

# Contents

<b>Abstract</b>	<b>iv</b>
<b>List Of Figures</b>	<b>v</b>
<b>List Of Tables</b>	<b>vi</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1. Problem Definition And Motivation	1
1.2. Scope of the project	1
1.3. Proficy Machine Edition	1
<b>2. Related Work.....</b>	<b>6</b>
2.1. Present Trend Tool	6
2.2. Proposed Trend Tool	7
<b>3. Requirements Analysis.....</b>	<b>9</b>
3.1. Integration With PME	9
3.2. Additional Trend Enhancements	9
3.3. Assumptions And Dependencies	12
3.4. Software and Hardware Requirements	12
<b>4. Detailed Design.....</b>	<b>13</b>
4.1. Design Approach for Integration	13
4.2. Selected approach	15
4.3. Design for additional Trend Enhancements	19
<b>5. Implementation.....</b>	<b>24</b>
5.1. Integrating tool into PME	24
5.2. Implementing Trend Enhancements	24
<b>6. Testing and Results.....</b>	<b>30</b>
6.1. Test Cases	30
6.2. Results	34
<b>7. Conclusion And Future Work.....</b>	<b>42</b>
<b>8. References.....</b>	<b>43</b>

# Chapter 1

## INTRODUCTION

### 1.1. Problem Definition and Motivation:

Current Project work is to tightly integrate an enhanced trend tool into Proficy Machine Edition HMI (Human Machine Interface) target and to provide some additional trend capabilities to the proposed trend tool.

Motivation for the integration of new trend tool into PME HMI target is to overcome some of the limitations of existing trend tool along with the provision of additional trend capabilities as per the user requirements.

### 1.2. Scope Of the project:

The project problem over here is to tightly integrate an enhanced trend tool into Proficy Machine Edition's environment and to provide some additional trend capabilities. But the scope of this project is to tightly integrate the trend tool into PME HMI target and to provide the trend capabilities by using the logged files obtained through *Data Logging* feature of PME HMI. This project deals much with configuration than the communication with Runtime Environment

### 1.3. Proficy Machine Edition:

Proficy Machine Edition (PME), a Human Machine Interface, is the most powerful, easy-to-use supervisory monitoring and control software. It consolidates the collection of data from your facility sensors and devices like PLCs (Programmable Logic Controllers), and then transforms the data into dynamic text, alarm and graphic displays. It gives users access to real-time information, helping them make appropriate decisions to improve quality, productivity and, ultimately, profitability.

Simply to say PME offers us a complete solution for the development of automation applications under one roof. Proficiency Machine Edition (PME) Projects are projects created with Machine Edition products (such as View or Logic Developer – PC) that are stored on the local computer or in a shared Projects folder. Generally, a Machine Edition project consists of one or more targets. A target is a machine/device where the configured projects run. Targets can also have one or more components. Using a project template automatically adds targets, components, and other items to our new project.

A project can be saved and backed up while working with the project and if necessary we can restore an existing or previously backed up Machine Edition Project. The IDE of PME is comprised of different types of toolbars, Navigator, Property Inspector, Help Window, Feedback zone, Tool chest, Tabbed Editor Window etc. and appears as follows.

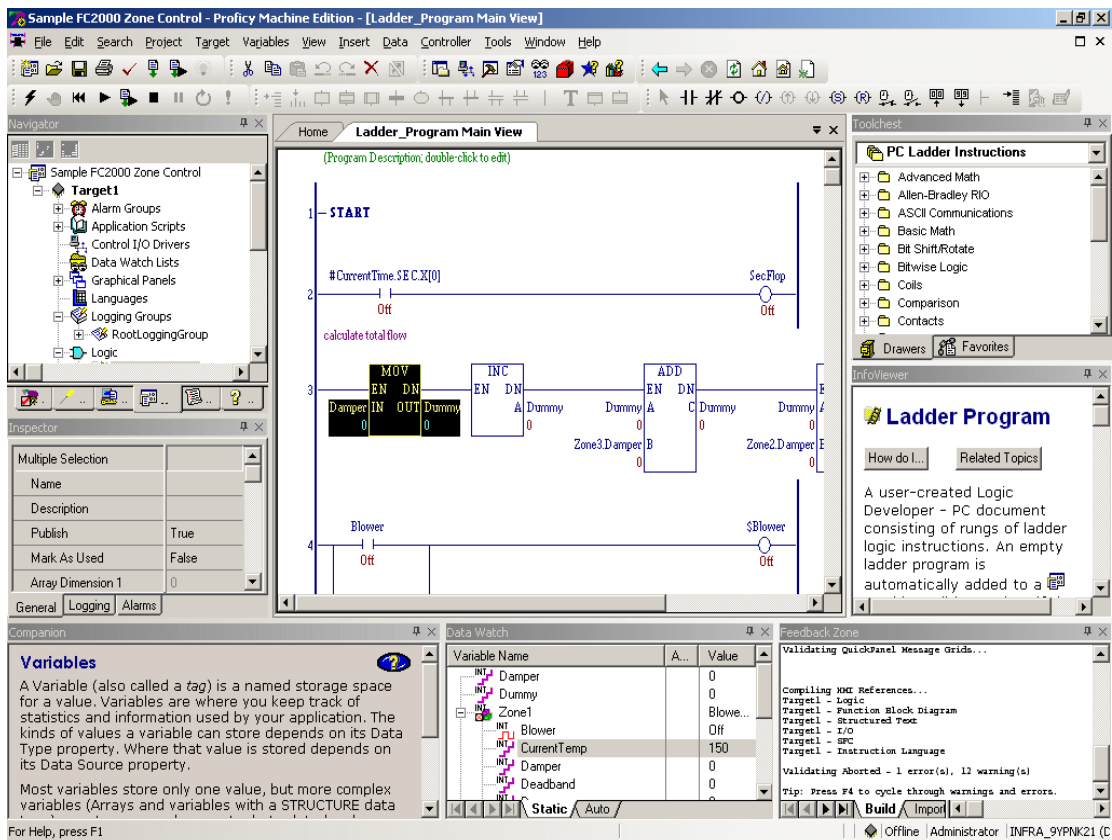


Fig: 1.1.PME Environment

## Common PME tools:

Following are the common Machine Edition Tools

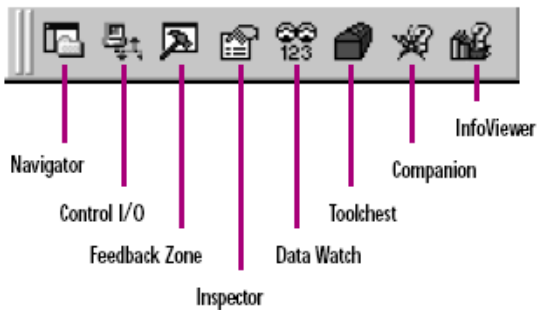


Fig: 1.2. Common PME tools

### *Navigator:*

Organizes and displays project information in a tree structure

### *Navigator Tabs:*

Conveniently layers information about our projects

### *Inspector:*

Displays properties of the selected object

### *Companion:*

A dynamic help window that provides information based on what we are doing

### *Data Watch:*

Displays the current status of our project variables

### *Feedback zone:*

Displays output information generated by Machine Edition components

### *Info Viewer:*

An embedded browser used to display reports and comprehensive help.

### *Tool chest:*

Contains preconfigured objects, which can be dragged into Machine Edition projects.

### *Tabbed Editor Window:*

It's one of the many windows, where we can create and edit our application. Tabs at the top let us switch between active tabbed editor windows.

In Machine Edition, user configures the project first and then validates the configured project. If any errors exist in project configuration, those will be displayed in feed back zone during validation process. After successful configuration, project will be downloaded to its relevant targets.

**View Runtime:**

View Runtime is the program that runs our project's configured HMI (Human Machine Interface) on a target computer. It is a separate application from Machine Edition; while we can start view runtime from within Machine Edition, we do not need to be running Machine Edition to use it.

**HMI – Data Logging Feature:**

As the scope of this project work falls under data logging feature of PME HMI target, it's necessary to revise what it does exactly.

Human Machine Interface Data Logging Feature stores a history of changes in application data. That means, whenever View Runtime is running, logging enabled variables' data will be stored in a log file as per the specified logging strategy. We can then view and analyze the logged data to detect the trends, events and inefficiencies if any.

The trend tool that is going to be integrated uses the log files generated by HMI – data logging mechanism.

## **Organization of the project:**

**Chapter 1** gives the project motivation and provides the background needed to understand the details of the project.

**Chapter 2** describes the features of present trend tool and its working and features of proposed trend tool and its functionality.

**Chapter 3** gives the overview of the requirements and detailed design approaches to implement the requirements.

**Chapter 4** gives the details of implementation of the requirements.

**Chapter 5** gives the test cases and results of the work undergone.

**Chapter 6** Concludes the work done and mentions the future work.

# Chapter 2

## RELATED WORK

To Under go the problem to be solved, the features of present trend tool and the proposed trend tool need to be revised and should understand the working of both the tools in the existing HMI target/project. This section gives us the description of present trend tool and proposed trend tool.

### 2.1. Present Trending Tool:

To view the trend of the logged application's data already there exists a trend tool. Current trend tool takes the application's log files as input and plots the logged data. The input file format for this trend tool is specific. It won't consider data from different types of data sources.

Figure below shows the snapshot of the proposed trend tool.

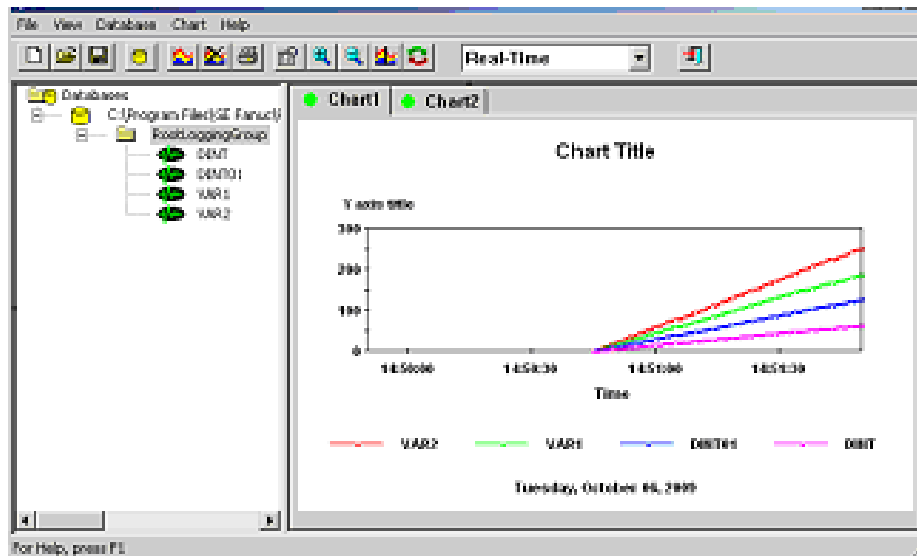


Fig: 2.1. Present trend tool

List of all the logged variables will be displayed left side and whenever user selects to plot the tag, selected tag will be plotted in chart window (Right side). There

exists a provision to plot in more than one chart window. User at any time can export the plotted data and can use in other applications. User can start observing the trend of application data of a project without launching the Machine Edition.

But, the existing trend tool has following limitations.

- Displays only limited number of tags for plot window
- Limited number of chart windows
- No provision to group the variables and plot the group
- Very old and don't have complete code control over code base
- Real time mode supported is very limited

## **2.2. Proposed trending Tool:**

To overcome the limitations in present trend tool and also to provide additional trend facilities, a new and trend tool is going to be integrated into Proficy Machine Edition environment. New trend tool is developed with latest .Net technologies and is comprised of several sub object interfaces like channels, labels, annotations, axes etc. (unlike the existing tool). It has support of different features like,

- Multiple annotations,
- Multiple labels,
- Multiple legends,
- Multiple plot windows,
- Multiple axes,
- Multiple data cursors
- Panning
- Binary and text file format support

Most of the properties of each component can be edited dynamically at run time. Following is a sample snapshot of the proposed trend tool showing legend with 9 channels, displaying corresponding channel color on plot, X-axis and Y-axis.

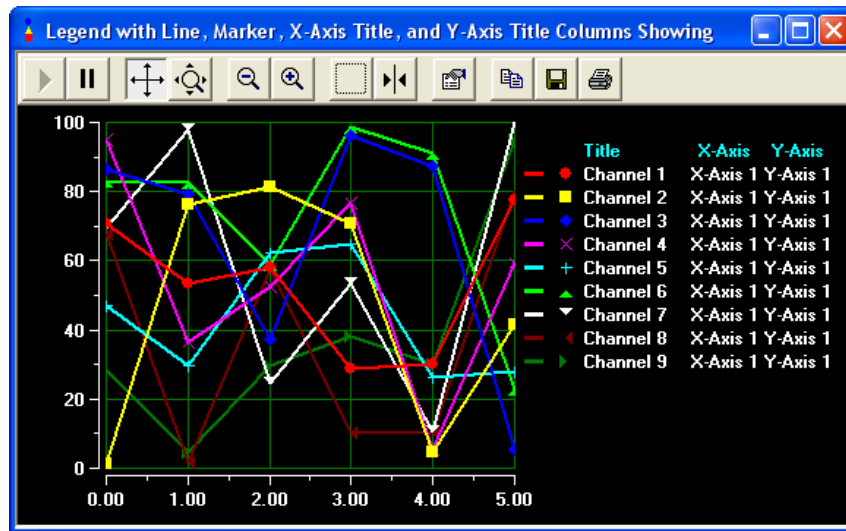


Fig: 2.2. Proposed Trend Tool

Proposed trend tool that is going to be integrated allows the user to flexibly change properties of any plot component. New trend tool is capable of trending the live data continuously until the application/process has consumed maximum amount of memory available to it.

# Chapter 3

## REQUIREMENTS ANALYSIS

To shape the system, an analysis of precise specification of requirements is necessary. Following are the requirements that need to be satisfied.

### 3.1. Integration With PME:

Currently when user is working with Machine Edition and if user wants to observe the trend of the configured variables' data of a target, user has to launch the trend tool from outside of Machine Edition environment. The requirement over here is to tightly integrate an enhanced trend tool into Machine Edition environment, so that the user can access the tool from the same environment without moving to another window.

### 3.2. Trend Capabilities:

Following are the important additional trend capabilities that need to be provided to the tool as per the requirement and usage.

- **Number of variables displayed:**

In automation industry, user needs to observe trend of thousands of variables. So new trend tool should be able to display the trend of many variables.

- **Selection/Reselection of variables:**

Some times user need not observe the trend of all the traces. So, User should have a facility to select the desired variables and observe the plot of only selected variables.

At any point of time, user should also be able to reselect the new variables or deselect the already selected variables.

- **Variables Grouping:**

User should have a provision to group the desired variables and can observe the plot of the grouped variables. At any time user should be able to add or remove the tags to the group.

So that, whenever user reconfigures the plot, user need not select the traces individually; instead can select the group at a time.

- **Saving/Recalling:**

Sometimes user needs to analyze already trended data. So user should have a provision to save/Recall the data/Configuration.

- **Scrolling the plot:**

While data is being captured, user should be able to view the variables scrolling in the plot window. Window can be of fixed time duration or may span from a selected starting time to the present, or may span from start of the data acquisition to the present.

- **Legend:**

A legend should be provided to allow the user to view and modify some of the properties of the variables. Legend provided should display the name of the variable trended, its scale values (like maximum value trended, minimum value trended, mean of the trended values etc.), color of the trended variable, variable's trend line style, the axis to which the variable is assigned etc.

- **Trace properties:**

Some times user needs to observe a particular trace among the hundreds of variables. So to differentiate the trace among all other user needs to configure some properties of that trace.

Hence, at any point of time user should be able to modify some of the trace configuration items like color of the trace, line style of the trace, line width of the

trace, enabling/disabling the data point markers on the trace, axis assignment of trace etc.

- **Axes Scaling:**

Tool should automatically adjust the scales of the axes (either X-Axis or Y-Axis) as per the values of the variables trended. But some times user may need to manually adjust the trace of the axes. So tool should also provide the facility to manually scale the axes.

- **Multiple Axes:**

Sample rates of all the variables need not be same. So some times some variables may have very low sample rate and some variables may have very high sample rates. To visualize the plot of all those variables in a plot window, there is a need to assign them to different axes.

So, user should able to add multiple axes to the plot and should be able to adjust the scales of those axes.

- **Play/Pause:**

When data is being captured and the captured data is being trended, user may need to observe the trend at particular instance. So user should be given facility to pause the live trend and then continue again when needed.

- **Panning And Zooming:**

Sometimes user needs to clearly observe the trace values at a particular instance. At that time user should be able to quickly and conveniently Pan and zoom the plot window.

- **Multiple Data Cursors:**

To measure data values of trended variables data cursor is used. User should be able to place more than one data cursor whenever needed.



# Chapter 4

## DETAILED DESIGN

Beginning once software requirements have been analyzed, software design is the last software engineering action within modeling activity and sets the stage for construction (code generation and testing). Following sections gives the detailed design approaches to tightly integrate the tool into PME and to provide additional trend capabilities to the tool.

### 4.1.Design Approach for Integration of tool:

To tightly integrate the new trend tool into the Proficy Machine Edition Environment there are several approaches to configure. But each approach has some advantages and disadvantages in configuration perspective.

Some of the possible ways to access the tool are,

- As a tool in tool chest browser
- As a tool in toolbar
- As a standalone tool which can be accessed from start menu
- As a utility which can be opened in editor of PME.
- As an utility which can be launched from PME but running independently as separate process
- As one of the components of Target in navigator

Each of the above approaches is explained below.

#### **From Tool Chest:**

To access the tool from tool chest, first it should be placed as a tool in tool chest. Then whenever needed it can be dragged from tool chest and can be dropped into Editor Window.

If tool is configured with this approach, user can easily access the tool by simply dragging and dropping. But, if the tool is placed in tool chest, it's possible to have multiple instances of the tool in same editor window and tool will not be target specific.

**From Toolbar:**

Currently there exist some dynamical tools in PME, which can be accessed from toolbar. So the new trend tool can be integrated as one of those dynamic tools available.

If this could be the case, it will be flexible to the user, but a difficult task to place the tool in toolbar. It could also be possible to have multiple instances of the tool. If the tool is placed in toolbar dependencies are high and is difficult to maintain.

**From Start Menu:**

It is also possible to launch the tool as the existing trend tool. But it will not be tight integration of tool with PME.

**As an utility:**

It could also be possible to configure the tool in such a way that the user can access the new trend tool as a utility. User can access it just by shifting to utilities tab in the project navigator.

If configured through this approach, tool could not be specific to a HMI target/project. Whatever the other type of the target/project, tool is accessible. But this approach makes the tool to be tightly integrated with PME.

**As an utility running independently:**

It could also be possible to configure the new tool as a utility, but the tool runs as a separate process, independently. Simply, tool can be launched from utilities tab in the scope of PME, but it can run independent of PME.

If configured with this approach, tool should be launched only from PME. After that it could run as a standalone tool. But development effort could be more if this approach is followed and also tool could not be specific to a HMI target/Project.

**As a component in target:**

It could be possible to configure the new trend tool in one of the branches of a target navigator. Whenever a new target is added to the project, it should have new trend tool as one of the branches of the target.

If configured through this approach, it is specific to a target and the tool is tightly integrated with PME.

## 4.2. Selected Approach

Observing the possible design approaches to integrate the tool into PME, each approach has its own advantages and disadvantages. Hence, to decide the design approach a six-sigma tool called “Pugh Matrix” (also called as Criteria Based matrix) is used to suggest a best fit for the integration

### 4.2.1. Pugh Matrix:

A Pugh Matrix is a method that helps determine which potential solutions are better (based on criteria) than others. It is a scoring matrix used for concept selection, in which options are assigned scores relative to criteria. The ‘best’ approach is selected based on the consolidated scores.

#### Steps:

- Create a Pugh Matrix
- Develop criteria to evaluate possible improvement solutions
- Assign rank (relative importance) for each criterion
- Identify alternative solutions
- Evaluate how well the possible solutions perform on each criterion and assign scores
- Compute the consolidated scores, sum product of scores for each solution and the criteria ranking
- Select the best approach with the highest consolidated score

Following is the Pugh Matrix Constructed with the available solutions for integration of tool into PME and the criteria to integrate the tool.

## Pugh Matrix

Key Criteria	Alternatives							
	Importance Rating	Existing Trend Tool (Bench Mark)	Tool Chest	Toolbar	As utility in Utilities list	from Project Node	From Start Menu	Utility- running as separate process
Low Dependency while configuration	5	s	-	-	-	-	s	s
Low Dependency with real time	5	s	-	-	s	s	s	s
Integration with real time	5	s	-	-	s	s	s	s
Synchronization with real time	5	s	-	-	s	s	s	s
Single Instance	5	s	-	-	s	s	s	s
Less Development Effort	4	s	-	-	-	+	-	-
Less operational effort ( User Friendly)	4	s	-	-	+	+	+	+
Less Memory Consumption	4	s	-	-	-	-	+	+
Project Specific (not generic)	4	s	-	+	-	+	-	-
Launching Dependency	3	s	-	-	-	-	s	-
Reusability of code	3	s	-	-	-	+	-	-
Future Enhancement / Maintainance	3	s	-	-	+	+	+	+
Multiple chart Support	3	s	-	-	+	+	+	+
Sum of Positives			0	1	3	6	4	4
Sum of Negatives			13	12	6	3	3	4
Sum of Sames			0	0	4	4	6	5
Weighted Sum of Positives			0	4	10	21	14	14
Weighted Sum of Negatives			53	49	23	12	11	14

Fig: 4.1. Pugh Matrix

The key criteria as per the requirements are listed in Pugh Matrix and were given rating as per the importance. Existing trend tool is considered as bench marking. The symbols '+', 'S' and '-' represents 'Better', 'Same' and Worse respectively. For example if the symbol '+' is used in a cell, that criteria is well satisfied in corresponding alternative compared with bench mark approach.

### Observation and Selection:

From the Pugh matrix constructed, it is observed that it is not at all advisable to access the tool from the tool chest as this approach has least consolidated scores. Comparing all the scores, weighted sum of positives is more for the project node approach i.e., accessing tool as a component of a target in a project and sum of negatives is less for start menu and project node approaches.

Considering the highest consolidated scores, **Pugh Matrix suggested to integrate the new tool in project node of the navigator in PME environment.**

Following is the over view of the few important classes which are evolved in design of project problem implementation.

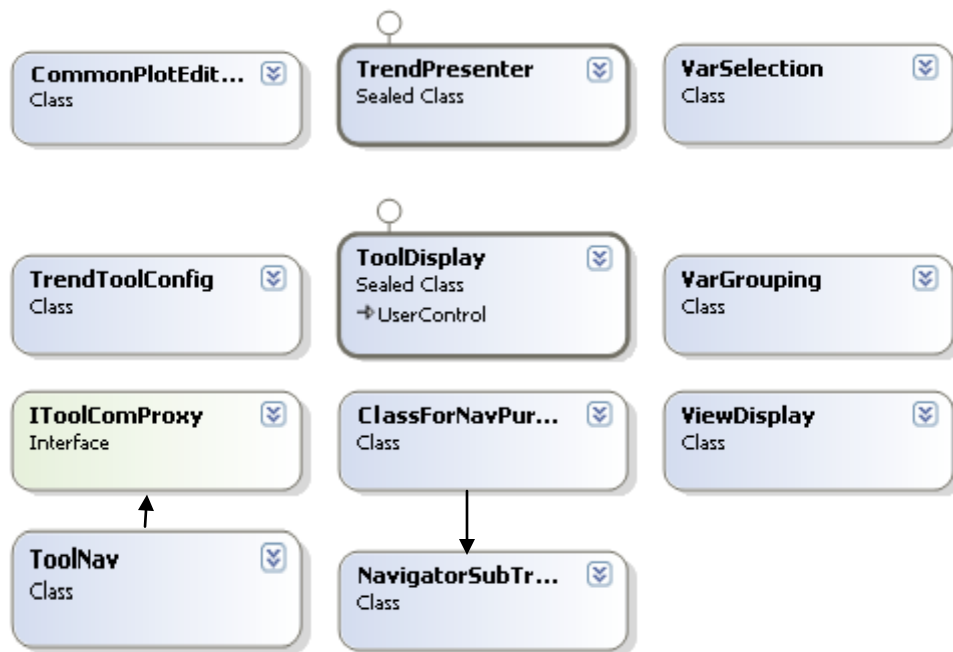


Fig: 4.2. Classes that are designed

Here is the overview of each class that is used in design.

***NavigatorSubTree:***

This class is used to add component nodes, sub-nodes to the navigator and to handle the events in navigator side.

***ClassForNavPurpose:***

This class is used to implement our tool component in navigator. This class extends from *NavigatorSubTree* class. A pointer to the interface *IToolComProxy* is used to take care of the navigator actions.

***ToolNav:***

This class implements the *IToolComProxy* interface's methods and configures the variables to be plotted using *VariableSelection* and *GroupingVar* classes.

***VariableSelection:***

This class is used to select/reselect the variables.

***GroupingVar:***

This class is used to group the variables.

**ToolPresenter:**

This class takes care of configuring and displaying using *TrendToolConfig* and *ViewDisplay* classes.

**TrendToolConfig:**

This class is used to configure the views, data sources and variables/groups to be plotted.

**ViewDisplay:**

This class is used to display the configured view's variables on plot window using *ToolDisplay* class

**ToolDisplay:**

Actual tool's classes and API's are used in this class.

**CommonPlotEditor:**

Properties of the tool's component can be modified using this class.

Consider a scenario where user wants to display a set of variables in the plot window. The sequence of classes involved can be as follows.

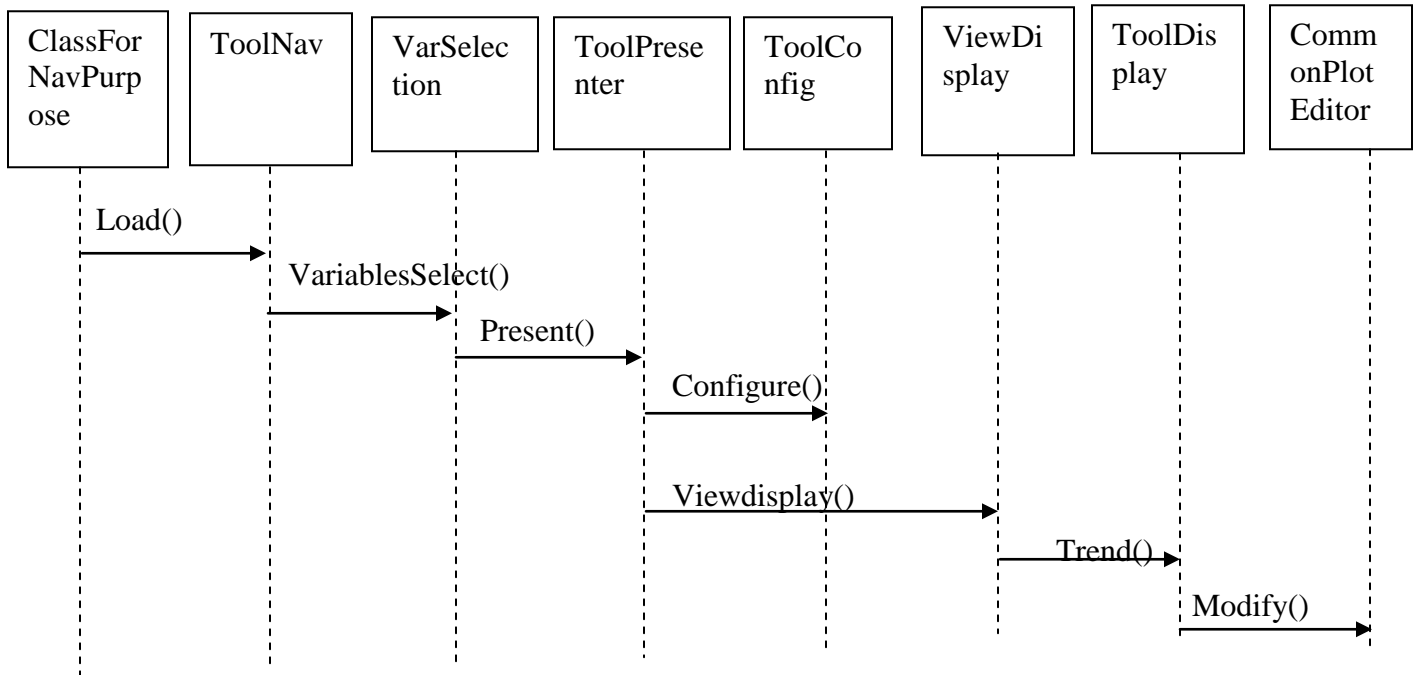


Fig: 4.3. Sequence of steps to trend

- User right clicks on the navigator node, where *ClassForNavPurpose* (extends the *NavigatorSubTree* class) is involved and calls to a method of *ItoolComproxy*.
- *ToolNav* class implements the *ItoolComProxy* interface methods.
- Then variables and Groups will be selected for plotting using *VarSelection* and *VarGrouping* Classes.
- Now the data sources, groups, views, which are in *TrendToolConfig* Class can be configured from *ToolPresenter* Class.
- After configuring, *ViewDisplay* class is used where in all the groups of variables and variables to be displayed are collected, which in turn calls *ToolDisplay*. In *ToolDisplay*, actual tool's classes and APIs are used. If user wants to modify any of the Properties *CommomPlotEditor* class can be used which is in *ToolDisplay* class

### **4.3. Design Approaches For Trend capabilities**

As part of the design to provide additional trend capabilities to the new trend tool based on the available functionality, additional tasks to be accomplished are design of the configuration for the tool's accessibility as a HMI target component and the Design of a common editor in which most of the plot components' properties can be modified.

At the component node in HMI target navigator, some configuration is necessary to access the tool. It can be configured in such a way that the component can be expanded to have different sessions. Each session can have a different data source and user can have different views of available variables in the data source. Component node and each of the sub nodes in component are provided with right click menus where some of the requirements can be implemented.

#### **Design of Common Editor:**

A common editor is designed, which can be used from *ToolDisplay* class, in such a way that most of the additional trend requirements can be satisfied.

The editor can have different tabs for each plot component and each tab has its own components, which can be edited by the user. Every tab can have an apply button, ok button and a cancel button. If the apply button is clicked, properties edited by user are set to corresponding component and the editor is still alive, so that the user can navigate to other tabs. If ok button is clicked, properties edited by user are set to the corresponding component and the editor will be closed. If cancel button is clicked, edited properties will not be set and the editor will be closed.

This common plot editor can be accessed through a context menu item in the plot window.

Design approach to meet each of the additional trend requirements is as follows.

- **Number of variables displayed:**

It is designed in such a way that all the selected variables (what ever the number may be) will be displayed in one plot window.

\* This functionality is being reused in our design

To display the variables in one plot window, *ViewDisplay* class can be used.

- **Selection/Reselection of variables:**

The selection/Reselection facility is provided to the user with right click options at the navigator (on the component). Whenever user wants to plot, user will right click at navigator and a window with available traces for trending will appear. User can check the desired variables and can observe the plot of checked/selected variables.

To reselect the variables user will use the same window where user can deselect already selected variables and can reselect some other variables.

Selection and reselection of variables can be done at *ToolNav* class, before configuring views, data sources.

\* This functionality is being reused in our design

- **Variables Grouping:**

User can group the variables by a right click menu item at the component in navigator. When ever user wants to group the variables, user will click on menu item and a window will be opened with list of available variables for trending, a

provision to create a group and select the desired variables to be placed into the group.

After grouping the variables, group name will also appear in the variable selection window. Users can select/deselect the group for trending. If user selects the group for trending, traces of all the variables in the group will be displayed in plot window.

Grouping can be done at *ToolNav* class.

\* This functionality is being reused in our design

- **Saving/Recalling:**

At any point of time user can be able to save the data/configuration by an option provided at the component. Whenever user chooses to export on component node, a file input dialog will be opened and user gives a file name. Then data/configuration can be saved to that file.

User can recall the same file from the component node itself. User needs to input the saved file and the data/configuration will be recalled.

Saving/Recalling the trended data/configuration can be done in *ClassForNavPurpose* through a method in *ItoolComProxy* interface.

- **Adjustment of trend in plot window:**

To view the variables scrolling in the plot window, Tracking of the axes should be enabled, so that as and when the data is captured, the scales of the axes will be adjusted. If tracking is enabled, trend tool will automatically takes care of tracking.

To set the time duration for the plot window, tracking style of the X-axis should be adjusted as per the requirement. This can be done using the *CommonPlotEditor* class.

- **Legend:**

A provision will be given to the user to select the legend to be viewed or not. If user desires to view the legend, then legend will display the list of all the variables that are trended. User can select the particular variable from the legend and can edit that variable's trace configuration using the common plot editor.

User can add/remove some items (or columns) to/from the legend using *CommonPlotEditor* Class

- **Trace properties:**

At any point of time, user can modify the trace properties using the common plot editor. User can select the trace to be modified either from plot window or from legend.

- **Axes Scaling:**

Automatic scaling facility of the new trend tool will be enabled by default, i.e., as when the data is captured, tool will automatically scales the axes. To manually scale the axes, common plot editor can be used.

- **Multiple Axes:**

By default a single X-Axis and a single Y-axis will be provided to the plot window. If user wants to add axes to the window, user can use the common plot editor, where Add and remove buttons exist. If user clicks on add button, a new axis will be added to the plot. User can go on adding any number of axes. User can select an axis and can click remove button, to remove the axis from the plot. User can select an axis and can modify its properties.

- **Panning and Zooming:**

Panning and zooming facility is provided to the user with a context menu on the plot window. If user selects the zoom mode, user can Pan the window. Through the context menu, zoom-in, zoom-out and zoom to fit features are provided to the user. User can use zoom-in any number of times and can use zoom-to-fit to get the regular plot of the traces.

\* This functionality is being reused in our design

- **Multiple Data Cursors:**

Multiple data cursors can be provided using a window, which lists out the available cursors, with Add, remove, rename and ok buttons. Whenever user clicks add button a new data cursor will be added to the list as well as to the plot. Remove button can be used to remove the selected cursor in the list. Rename button is used to rename the selected cursor. This window can be accessed through a context menu item in the plot window.

This can be done by using a dialog, which can be used in *ToolDisplay* class.

- **Trace configuration:**

User can select a trace and can configure it using common plot editor. In common plot editor, user can select a trace, modify its properties and can apply to the corresponding trace. Otherwise if Data markers are enabled for a trace, user can right click on that particular trace and can edit.

**Existing Trend Tool Migration:**

As new trend tool is going to be integrated into PME, it should also support the trending of existing tool's data files. Existing tool uses files from HMI – Data Logging feature, and uses some calculating factor. So, it is designed in such a way that new trend tool also plots the data from HMI – logging mechanism.

## **Chapter 5**

# **IMPLEMENTATION**

Implementation of the work is done based on the priorities that were assigned in the design phase. All the work is implemented using technologies listed in the software requirements section, like C# .net, VC++, and COM concepts.

### **4.4. Integration of proposed tool Through HMI Navigator:**

New trend tool has to be accessed by the user as a component in HMI target. Target components can be accessed from navigator. So the new tool should be included in the navigator. Following are the steps followed to integrate the new tool into PME.

- For the navigator purpose an ATL class is defined which implements all the operations that can be performed at the navigator side.
- While initializing the target components, check the target type and attempt to load the new class by passing its class id as one of the parameters to the method.
- Create the COM object using CoCreateInstance, specifying the Class id.
- Initialize the component's document to corresponding target by specifying Tab ID , Target ID etc.
- Finally, add this component node to navigator tree.

### **4.5. Implementation of additional trend capabilities:**

Some of the additional trend capabilities can be implemented in the class defined for the navigator purpose. This class gets the pointer of another class where the plot components are nested and implements the exposed methods. Hence, right click operations on navigator node affects the plot.

Here are the classes that are re-used in design with the fields of each class shown.

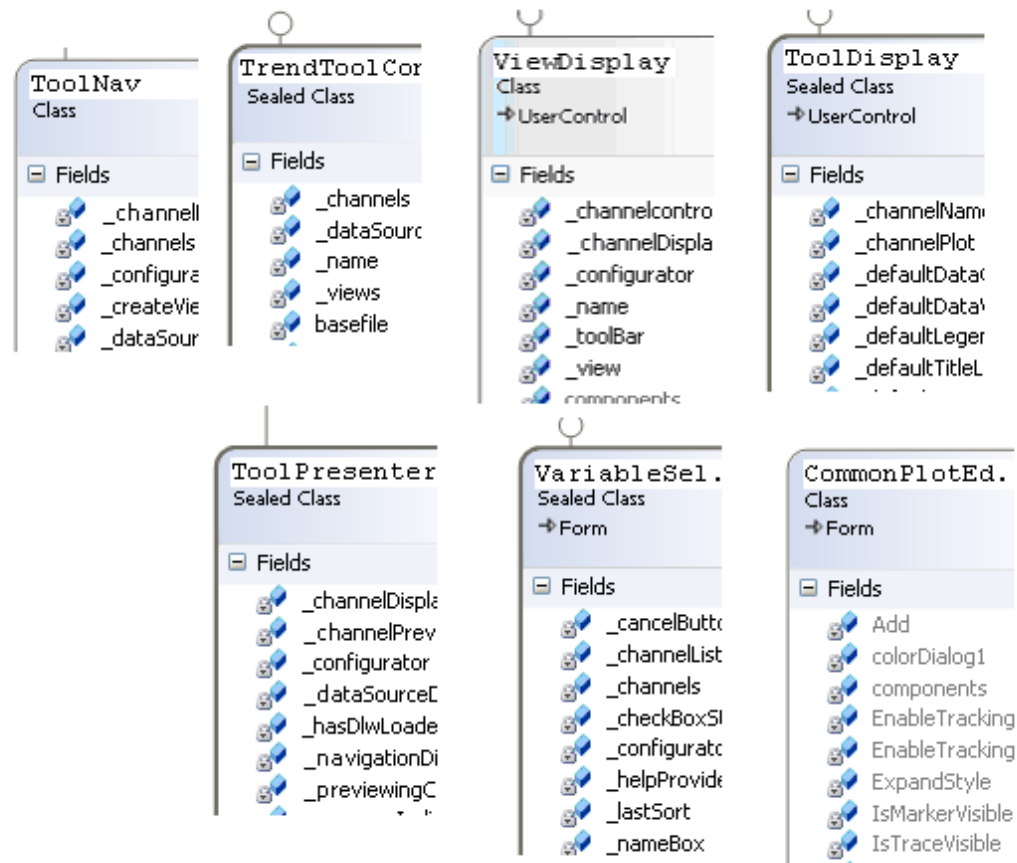


Fig: 5.1. Classes with fields shown

To implement additional trend enhancements, *CommonPlotEditor* need to be implemented.

```

CommonPlotEditor()
{
Controls.Add(X-Axis); // add X-axis Page
Controls.Add(Y-Axis); //add Y-axis Page
Controls.Add(Traces); //Add Traces Page
Controls.Add(Legend); //add legend page
//add controls in each tab page
}

```

Implementation steps of each of the additional trend capabilities are given below.

- **Number of variables displayed:**

To display a variable to the plot window, an instance of TrendChannelTrace class can be added to the collection of Channels of the plot component.

```
Plot.Channels.Add(TrendChannelTrace trace1);
```

This can be repeated for each variable that is going to be plotted. To display all tags in one page *ViewDisplay* class is used. In *ViewDisplay* class an object is created for *TraceDisplay* class and is used.

\*\* This functionality is being reused in the implementation.

- **Selection/Reselection of variables:**

Selection/Reselection of variables is decided to be done using a window, which obtains through a menu item from navigator node. Hence,

```
Foreach item // go through the list of all variables
```

```
{
```

```
If(variable.checked = true)
```

```
//add the variable to the plot channels list.
```

```
}
```

*VariableSelection* class is used to implement this feature.

\*\*This functionality is being reused in the implementation

- **Variables Grouping:**

To group the variables an interface, which gets or sets the list of data trace interfaces, can be used. All the desired variables to be grouped are added to a list.

Each item name in list is checked whether it matches with any of the available traces for plotting. If the name matches, it will be added to the list in the group interface. *GroupingVar* class is used to implement this.

\*\* This functionality is being reused in the implementation

- **Saving/Recalling:**

To save the data/configuration, the class containing all the information regarding traces, groups, views, data sources etc. is serialized to a file stream.

While recalling the data/configuration, same file stream will be de-serialized and the values will be extracted to the corresponding objects.

TrendToolConfig is mainly used to implement this feature.

- **Adjustment of trend in plot window:**

To view the variables scrolling in the plot window as and when the data is captured, tracking and scrolling features of axes will be enabled.

```
PlotXAxis axis;
```

```
Axis.tracking.enabled = true;
```

Tracking style of the window is adjusted using

```
Axis.tracking.style = //desired style;
```

This is done by modifying properties in *CommonPlotEditor*

- **Legend:**

User can select whether legend can be displayed in the plot window or not.

```
PlotLegend legend1;
```

```
If( legend1.visible = false)
```

```
//hides the legend in plot window.
```

User add or remove the columns to be displayed in legend.

```
legend1.columns.add(plotlegendcolumnitem item1);
```

```
legend1.columns.remove(plotlegendcolumnitem item1);
```

To add/remove the columns and to modify other properties of legend, *CommonPlotEditor* is used.

- **Trace properties:**

User can modify the trace properties using some APIs from the new trend tool.

To change the trace color of the selected trace,

```
TrendChannelTrace Trace1 ;
```

```
Trace1.Trace.color = //selected color from the dialog;
```

```
Trace1.Trace.style = //desired trace style available from tool's trace styles
```

*Trace1.Trace.Markers.enabled = true/false;*

Similarly other trace properties are also configured using *CommonPlotEditor*.

- **Axes Scrolling:**

By default tool enables the automatic scrolling facility for the axes. To manually scale the axes values,

*PlotXAxis Axis1;//or PlotYAxis Axis1;*

*Axis1.ScaleRange.Max = //desired maximum value*

*Axis1.ScaleRange.Min = //desired minimum value*

This feature can be implemented from *CommonPlotEditor* class.

- **Multiple Axes:**

To add/Remove the axis to the plot following APIs are used

*Plot.XAxes.Add(PlotXAxis axis1);*

*Plot.Yaxes.Add(PlotYAxis axis1);*

To remove the axes from the plot, following APIs are used

*Plot.Xaxes.Remove(plotXAxis axis1);*

Instead of axis name, index of the axis can also be provided as parameter to the plot component. *CommonPlotEditor* class is used to implement this feature.

- **Panning and Zooming:**

To Zoom-in and Zoom-out the plot following APIs are used.

*Plot.ToolBarAdapter.Dobuttonclickzoom-in();*

*Plot.ToolBarAdapter.Dobuttonclickzoom-out();*

Zoom-box is provided using the API

*Plot.ToolBarAdapter.Dobuttonclickzoom-box();*

\*\* This functionality is being reused in the implementation.

- **Multiple Data Cursors:**

A data cursor is added to the plot using

*Plot.DataCursors.add(PlotDataCursorChannels datacursor);*

Multiple data cursors can be added using a dialogbox. Whenever user clicks the add button this API will be used. If user clicks on remove button, following API is used.

*Plot.DataCursors.remove(selecteddatacursor);*

An *AddRemoveCursor* class is used to implement this feature. This class can be accessed from *ToolDisplay* class.

- **Trace configuration:**

Trace to be configured is selected using some event handlers.

**Existing Trend Tool Migration:**

Existing tool plots the data taking input from files obtained through HMI – data logging feature, with some calculating factor. New tool wont uses the calculating factor and trends the data by considering only the required fields from data logging files.

## Chapter 6

# TESTING AND RESULTS

This section describes how testing is performed on the tool integrated into PME HMI environment.

To perform testing some test cases are written based on the assumption that tester is familiar with the terminology used. To initiate the testing activity all preparatory activities like booting system, launching PME, creating and validating test data etc. need to be completed.

### 6.1. Test cases:

These are the test cases under gone as part of **Functionality Testing**.

Sl.No	Instruction	Expected Result	Pass/Fail	Comments
1	Tool Component node in navigator	In Navigator, tool should appear as a target component	Pass	
2	Right Click Options at navigator	→ All the right click options in the navigator side should work correctly → When ever user right clicks on the component or on the sub components, corresponding menu items should be displayed and should work as desired	Pass	
3	Number of Variables in the plot window	Minimum of 10000 variables need to be plotted		As per the work on evaluation copy of the tool it is able to display any number of variables. But it is not tested after integration.

4	Selection/reselection of variables to be trended	<p>→User should be able to select / reselect the desired variables</p> <p>→Should be able to view the plot of only those variables selected.</p> <p>→ Should be able to reselect/deselect variables to be plotted</p>	Pass	
5	Grouping the variables to be trended	<p>→User should be able to select the desired variables, group them and should be able to view the trend of all the variables in the group.</p>	Pass	
6	Saving/Recalling the trend Data/Configuration	<p>→At any point of time user should be able to save the trend data/configuration and should be able to recall the same</p>	Pass	
7	Common plot editor	<p>→Common Plot Editor should contain all the tabs related to plot component and all the controls on the editor should work as desired</p> <p>→All the properties should be loaded and applied correctly</p>	Pass	
8	Adjustment of trend in plot window	<p>→ User should be able to view the variables scrolling in plot window,</p> <p>→ User should be able to adjust the plot window as desired.</p>		Scrolling of variables is checked with the evaluation copy of the tool. This functionality is not tested after integration
9	Legend of trended variables	<p>→When ever user wants to show/hide the legend, user can check/uncheck the check box in legend tab of common plot editor. Checking/un checking should function correctly</p> <p>→By clicking on Add/Remove buttons</p>	Pass	

		columns should be added/removed from the legend		
10	Modifying trace properties	→When user sets the trace properties in the common plot editor, properties like trace color, trace style, trace line width, enabling/disabling of markers should work well	Pass	
11	Manual and Automatic scaling of axes	→Axes should be scaled automatically as per the trace values. →When user modifies the scales from common plot editor, those values should be reflected in plot	Pass	
12	Provision of multiple axes	→When user clicks on the add button in common plot editor, a new axis should be added to the plot →When user clicks on the remove button, axis should be removed from the plot. User should not be able to remove default axis	Pass	
13	Facility to play/pause	→Plot should be paused when zoomed or stopped tracking to the axes.		Checked with the evaluation copy of the tool. It is not tested after integration
14	Convenient Panning and Zooming	→User should be able to conveniently zoom and pan. Zoom-in, zoom-out, zoom-box, axes mode-zoom should work well	Pass	
15	Provision of Multiple data cursors	→Whenever user selects data cursor from menu, a dialog should open listing cursors in the plot. When user clicks add button, a new data cursor should be added to the plot. →When clicks on remove, it should be removed from the plot	Pass	

		→When clicks on rename, should be able to rename the cursor.		
16	Selection of trace for configuration	→Should be able to conveniently select the trace for configuring, either from legend or from plot window.	Pass	

**Table: 6.1. Test cases**

## 6.2. Results:

This section gives the results obtained throughout the work undergone.

The results shown below are after implementation and functionality testing.

Following snap shot shows the existence of tool as a component of the HMI target, after successful integration.

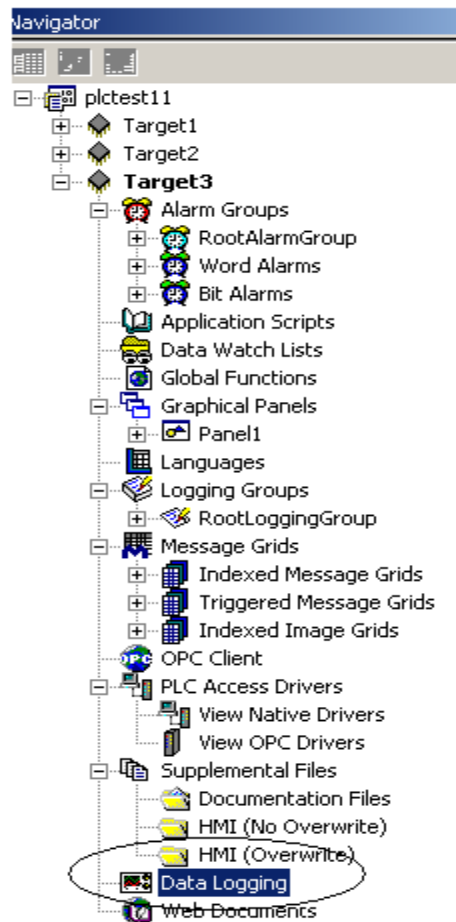


Fig: 6.1. Trend tool component in Navigator

Some of the requirements are implemented through navigator right click options. User can use the right click for different operations as shown in figure below.

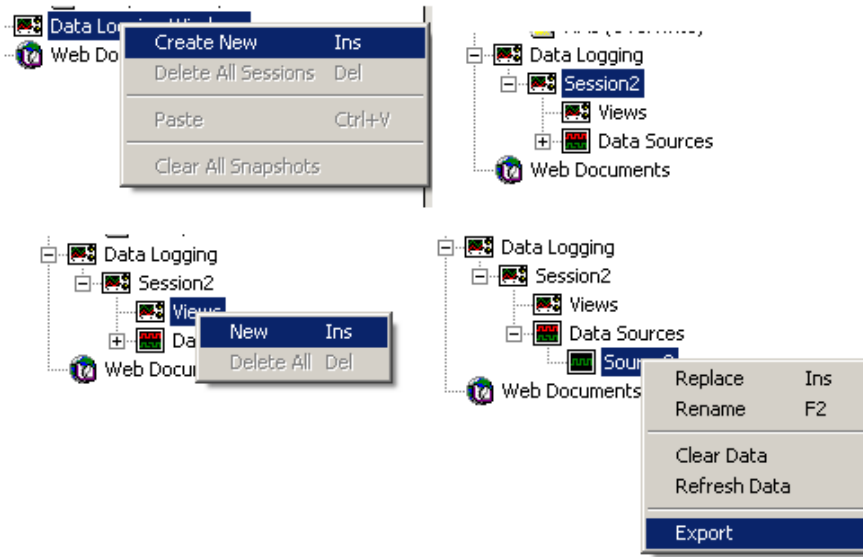


Fig: 6.2. Right Click menus at component node

Variables can be selected or reselected using the window shown below. All the variables available for trending are displayed in window. Only user-selected variables will be displayed in plot window. If user wants to deselect/reselect, user can reconfigure the selection using same window. To select/reselect and group the variables already configured dialogs in another application are used (Fig: 6.3., 6.4., 6.5.).

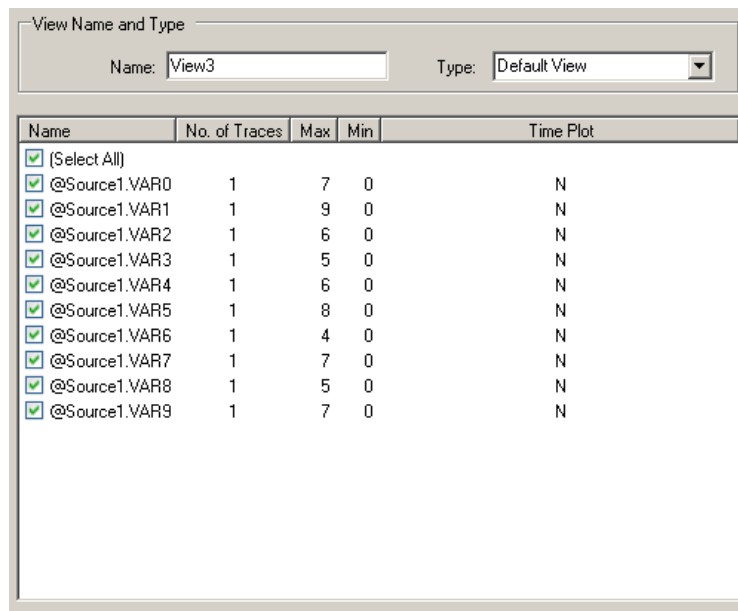


Fig: 6.3. Variable Selection/Reselection Window

User can group the variable by right clicking on session and choosing 'edit channels'. Then window below will be opened user can create a new group. In figure shown below 'Group1' is created.

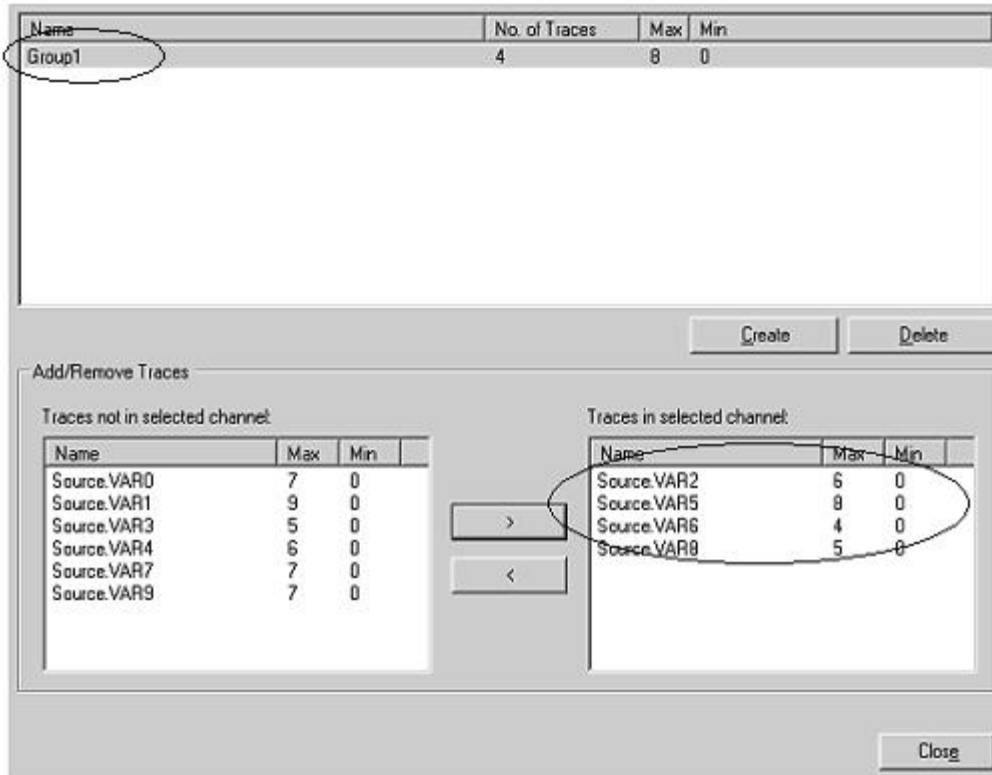


Fig: 6.4. Group with desired variables

If user wants to view the plot of the group, user has to reconfigure. When window is opened for reconfiguration, group created is also added to the list as shown in figure below.

Name	No. of Traces	Max	Min	Time Plot
<input checked="" type="checkbox"/> (Select All)				
<input checked="" type="checkbox"/> @Source1.VAR0	1	7	0	N
<input checked="" type="checkbox"/> @Source1.VAR1	1	9	0	N
<input checked="" type="checkbox"/> @Source1.VAR2	1	6	0	N
<input checked="" type="checkbox"/> @Source1.VAR3	1	5	0	N
<input checked="" type="checkbox"/> @Source1.VAR4	1	6	0	N
<input checked="" type="checkbox"/> @Source1.VAR5	1	8	0	N
<input checked="" type="checkbox"/> @Source1.VAR6	1	4	0	N
<input checked="" type="checkbox"/> @Source1.VAR7	1	7	0	N
<input checked="" type="checkbox"/> @Source1.VAR8	1	5	0	N
<input checked="" type="checkbox"/> @Source1.VAR9	1	7	0	N
<input checked="" type="checkbox"/> Group1	4	7	0	N

Fig: 6.5. Variable selection/Reselection window along with groups

Snapshot shown below shows the editor window with the tool integrated. All the selected variables are displayed in the plot window as shown.

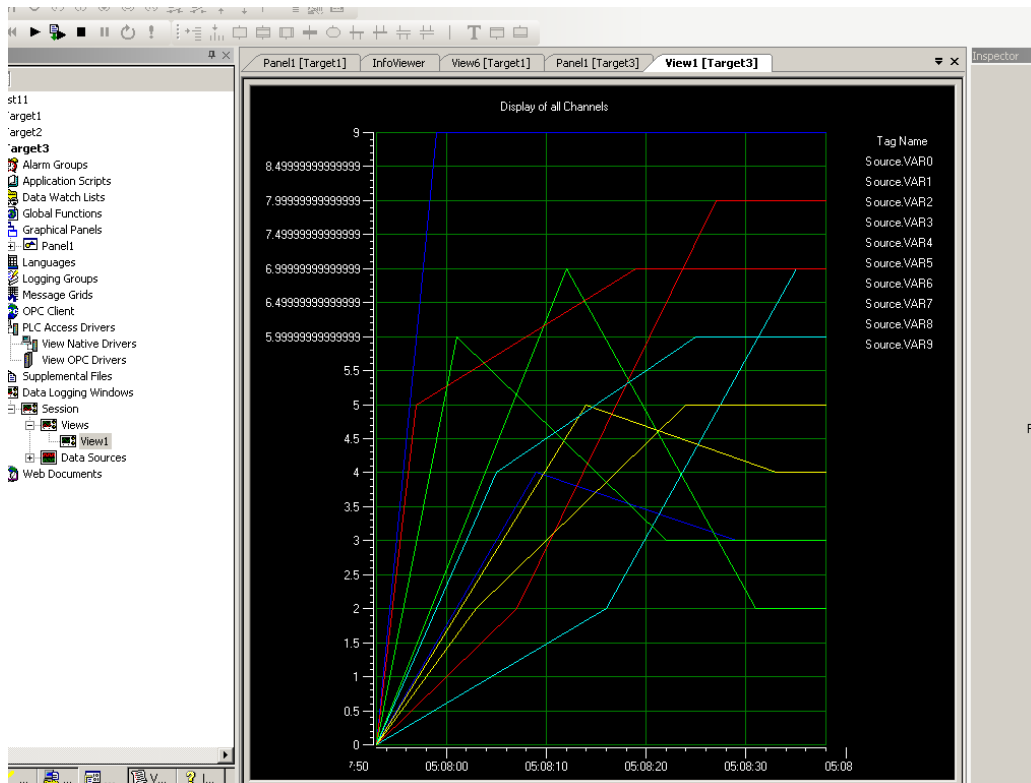


Fig: 6.6. New trend tool in editor window of PME

Most of the requirements are implemented from the context menu on the plot window. Figure below is the common plot editor, accessed from context menu, with different tabs for different plot components. User can navigate to the different tabs and can modify different plot components' properties without closing the editor.

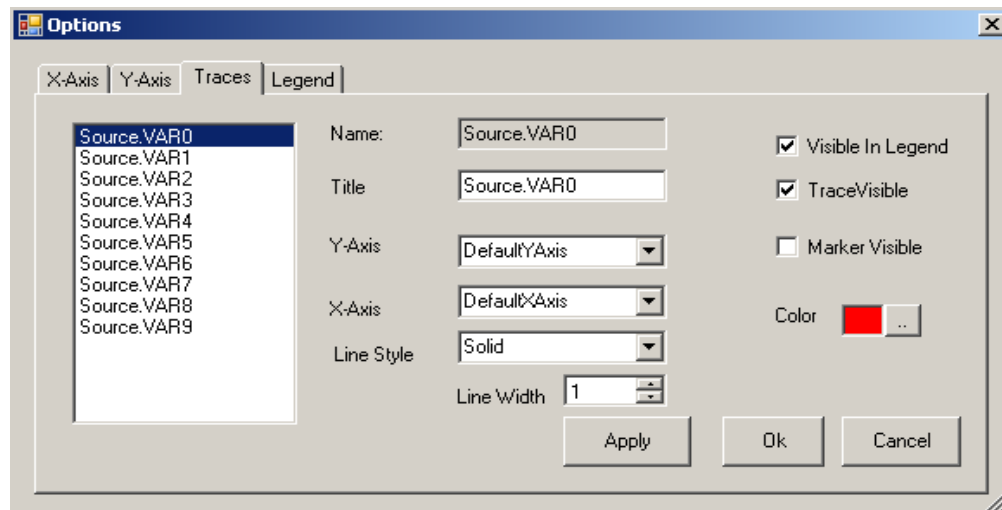


Fig: 6.7. Common Plot Editor

Following figures shows the plot window after changing a variable's trace line style to dash (instead of line) and trace color, trace line width (increased trace line width), additional y-axis, and multi column legend with three columns, zoomed-in, and window with added data cursors.

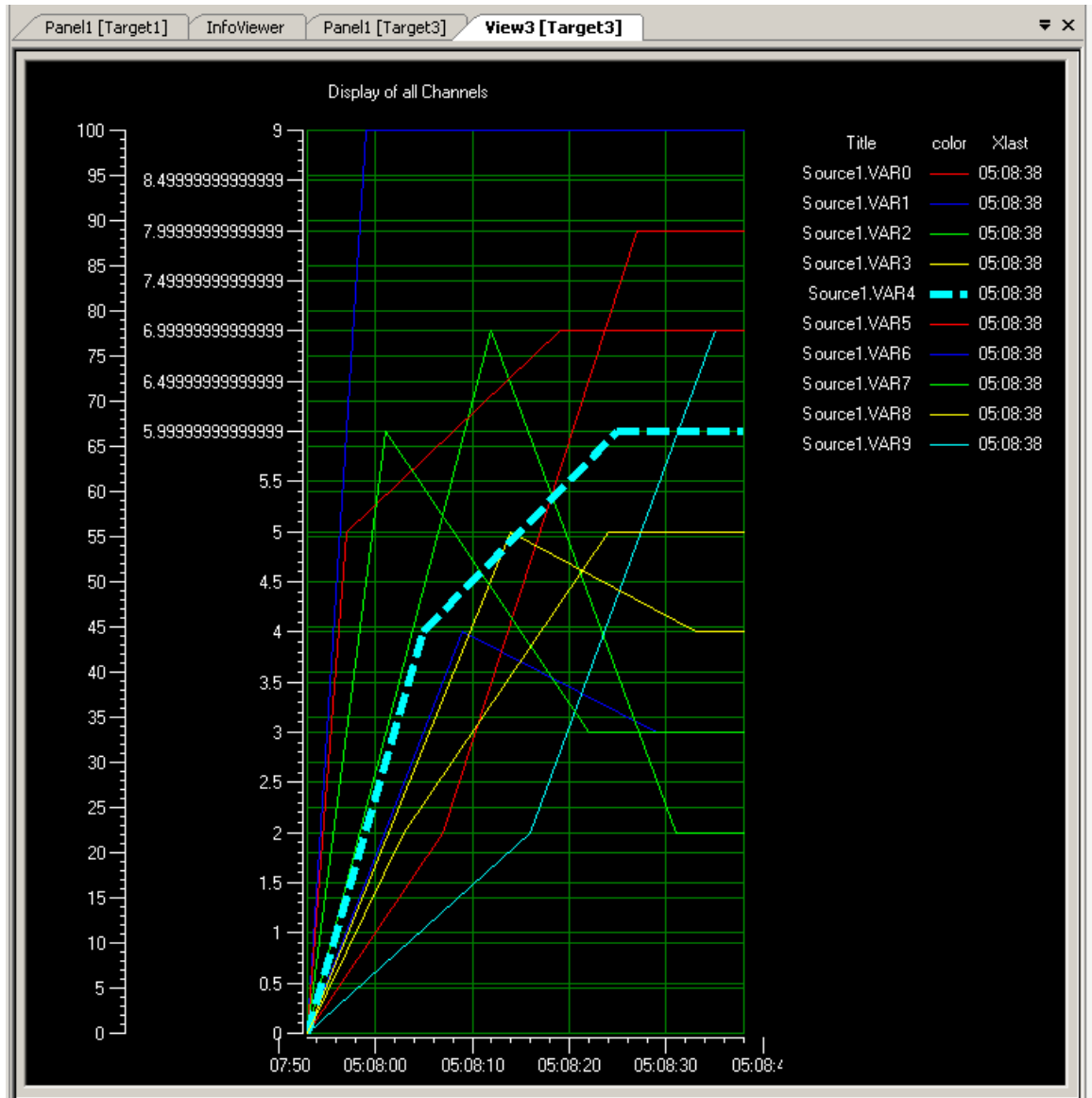


Fig: 6.8. Plot with modified Trace Configuration, additional axis and multi column legend

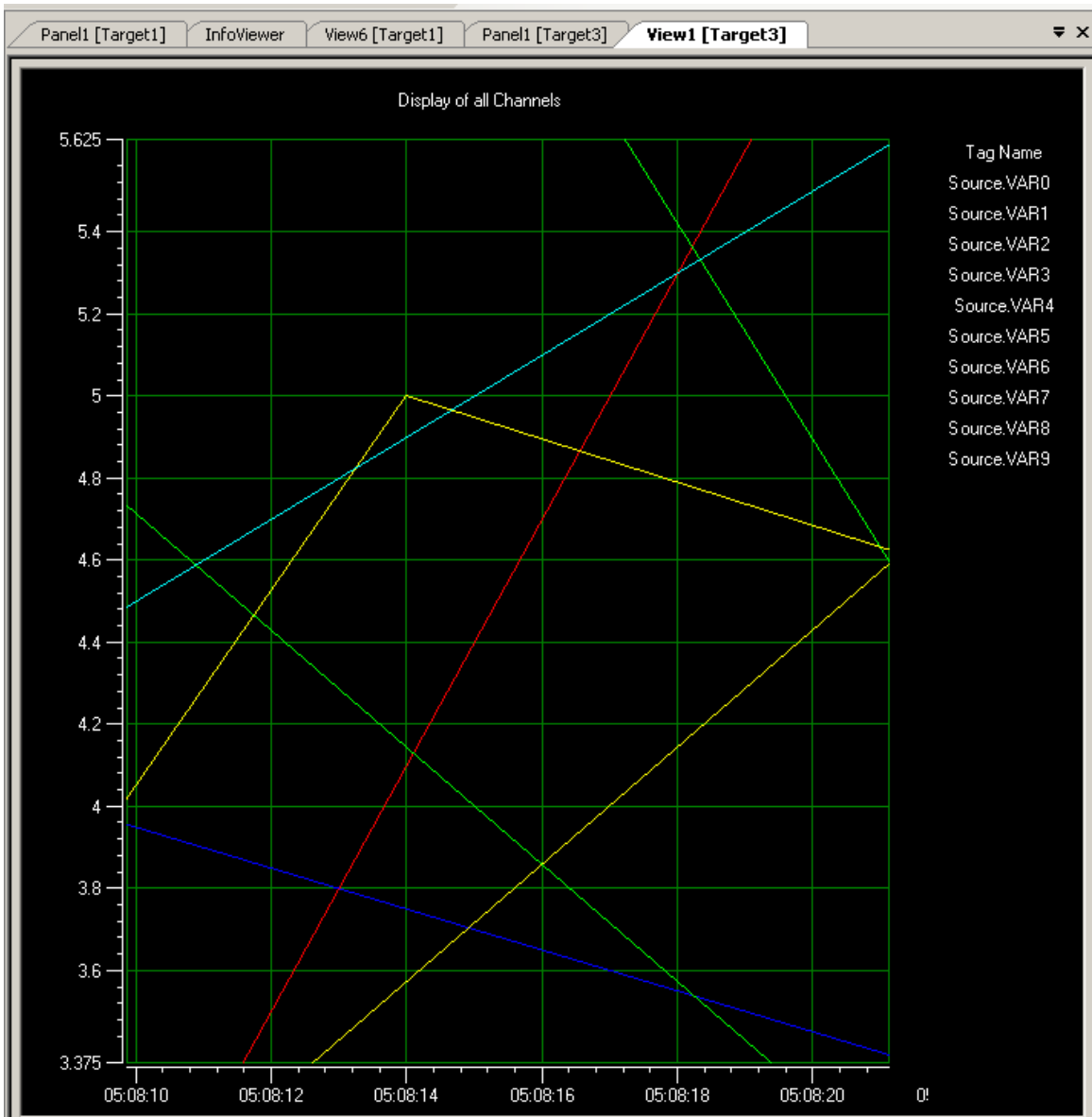


Fig: 6.9. Plot window that is zoomed-in

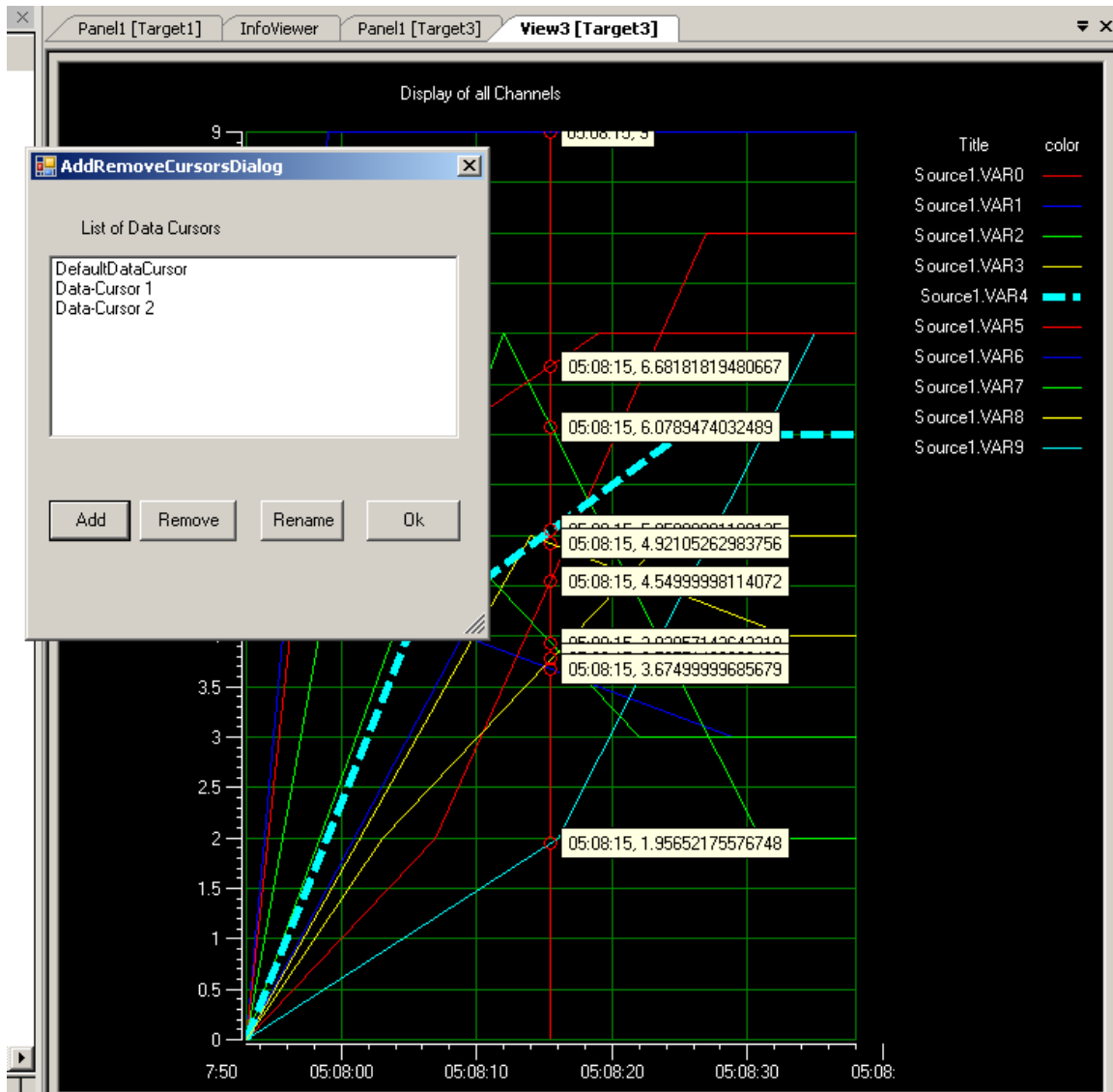


Fig: 6.10. Plot Window with Add/Remove Data Cursor Dialog and Data Cursor added

## **Chapter 7**

# **CONCLUSION AND FUTURE WORK**

### **7.1. Conclusion:**

A new enhanced trend tool is successfully integrated into the Proficy Machine Edition Environment. After integration of tool into PME, its features are well exposed to the user and user can now be able to effectively observe the trends to increase productivity and profitability.

### **7.2. Future Work and scope:**

New trend tool need to be integrated with the runtime environment and some other additional trend capabilities can be provided to the tool in PME environment, as the tool is given with many features (provides many classes and APIs). This work can be extended to the other targets instead of tool being target specific/project specific. As the tool is developed with latest technologies, durability is high. This work can also be extended/ported to work with other operating systems like Windows Vista & Windows 7.

In future tool can be connected to the Historian Data Server, where previous data is archived for days/years.

## Chapter 8

# REFERENCES

1. Proficy Machine Edition Help Manuals
2. Existing and Proposed trend tools Help manuals
3. MSDN library from Microsoft
4. C#. Net from CodeProject site and codeplex and C-sharpcorner sites.
5. Programming C# by jesse Liberty and Donald Xie.
6. Beginning C# by Jack Purdum