



Advanced Process Automation Technologies

# HPI Quick Start Manual

Version 1.2

## Table of Contents

1 Overview .....	4
2 HPI introduction .....	4
3 HPI projects.....	4
3.1 Single station project .....	4
3.2 Multi station project .....	4
4 Design a project .....	5
4.1 Create a project.....	6
4.2 Project setting .....	8
4.3 Server configuration.....	8
4.4 Node monitoring .....	10
4.5 Graphic pages .....	11
4.5.1 Drawing a shape .....	12
4.5.2 Using symbol library .....	13
4.5.3 Using windows controls .....	14
4.5.4 Using Dynamics .....	15
4.5.5 Add local variables.....	16
4.6 Writing Scripts.....	18
4.6.1 Creating a graphic page script.....	18
4.6.2 Script module .....	19
4.6.3 Writing a Simple Script .....	21
5 Historical Data Viewer .....	21
6 Alarm configuration .....	25
6.1 Create Alarm .....	26
6.2 Configuring alarm coloring .....	27
7 Report configuration .....	27
7.1 Create a new report page .....	28
7.2 How to configure a report .....	28
7.3 How to insert a data source.....	28
7.4 Order Columns in a Data Source .....	29
8 Working with smart objects .....	30
8.1 About smart objects .....	30
8.2 Aspects of smart objects.....	30
8.3 Create a smart object template .....	31
8.4 Create data of a smart object .....	31
8.5 Defining alarms and events of a smart object .....	32
8.6 Smart object nodes archiving by DIS.....	32
8.7 Defining symbol of a smart object.....	33

8.7.1 Defining the graphical view of the smart object .....	33
8.7.2 Defining the script of the smart object .....	35
8.8 Adding a faceplate to a smart object .....	36
8.9 Create an instance of a smart object .....	37
8.10 Update smart instances.....	39
9 Project deployment.....	40
9.1 Adding/removing network nodes (stations) .....	41
9.2 Assigning Packages .....	42
10 HPI Security.....	44
10.1 Introduction to HPI Security .....	44
10.2 Security Manager .....	44
10.2.1 Opening Security Manager .....	44
10.2.2 Define an Operation Instance .....	44
10.2.3 View or change operation instance properties.....	46
10.2.4 Add New Role.....	48
10.2.5 View or change role properties .....	50
10.2.6 Assign users to the roles .....	53
11 Run a project .....	54

# 1 Overview

This manual makes you familiar with **HPI** environment and tools. There are many tools with a lot of parameters and configurations to create a powerful HPI project. The goal of this document is not to go through every configuration and setting, but to walk through a few fundamental concepts at least sufficient enough to get a project started.

For additional information about using **HPI** see the relevant chapter in full application help.

## 2 HPI introduction

**HPI** is a fully scalable, powerful, easy to configure, web-enabled HMI/SCADA (Human Machine Interface / Supervisory Control and Data Acquisition) software for industrial automation and it is fully portable and independent of the control system.

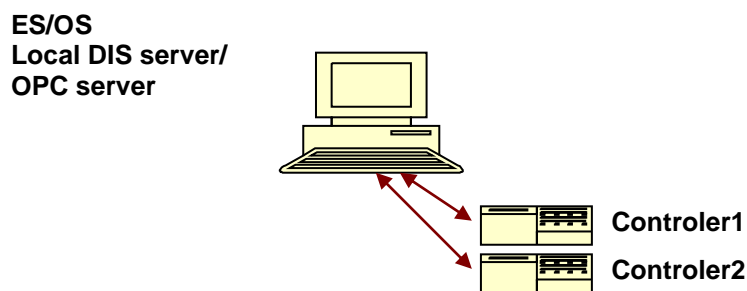
It is possible to use **HPI** for a small project as a single station project or use it for a very large project with several servers and clients as a multi station project.

The **HPI** software is available in two modes: **HPI****Design** and **HPI****Run**. The first one provides an interface for designing projects, graphic pages, reports, alarms and server configurations. **HPI**Run is used for running the designed project and monitoring the production line and generating reports.

## 3 HPI projects

### 3.1 Single station project

A single station project (or standalone project) is a project which has only one ES/OS with a local DIS server.



**Figure 1) Single station configuration**

Location of OPC servers in network doesn't affect project type. This means a standalone project can use OPC servers in other PC stations.

### 3.2 Multi station project

A multi station project is a project whose components can be installed in different PC stations.

Different components of a project are:

- **Engineer Work Station (ES):** This is a station for the engineers to design project using **HPIDesign** and **PCE** applications. Each project can have unlimited number of engineer work stations. In the case of having more than one **ES**, Project server component is necessary.
- **Operator Work Station (OS):** This station is used for running the designed project using **HPIRun**. Each project can have unlimited number of operator work stations.
- **Historical Data Warehouse (HDW):** This package is designed for providing features for backup, archive and deleting project run-time data. Each project can have one **Historical Data Warehouse**.
- **Project Server (PS):** This is a server which provides services for maintaining the design source of the project and delivers a multi-engineering environment. Each project can have one **Project server**.
- **DXS Information Server (DIS):** This server collects data from the available devices and delivers them for monitoring in **HPIRun**. Each project can have several **DISs** and each **DIS** can have one redundant.

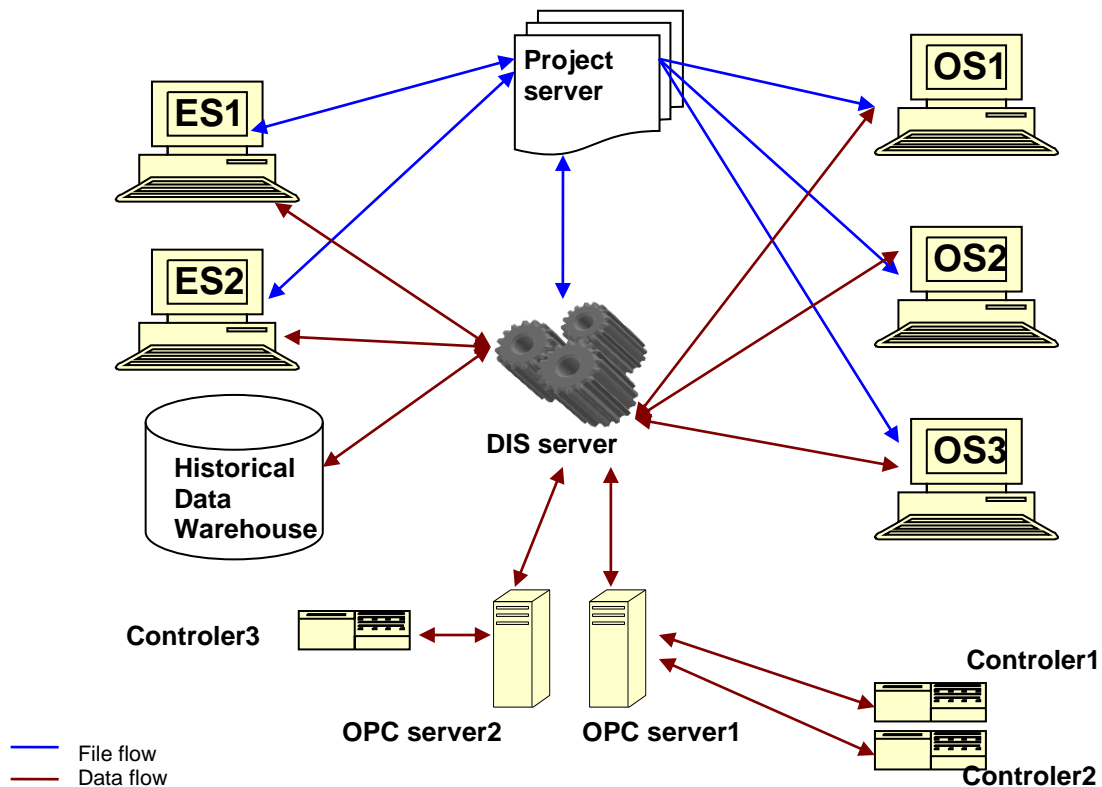


Figure 2) An example of a multi station project

## 4 Design a project

When you want to design a project for a **APAT Supervisory Control system**, you can find all necessary tools for design, configuration and deployment of the project in a unified configuration environment named **HPIDesign**.

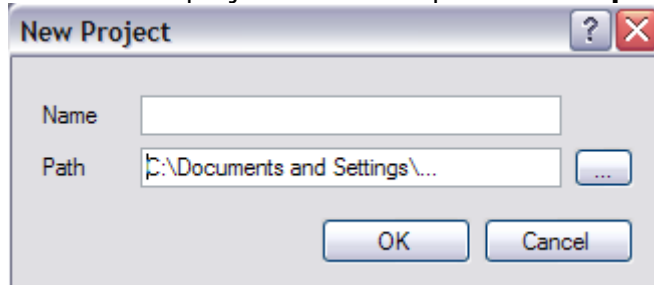
This chapter introduces different parts of an HPI project and how to create and configure these parts using **HPIDesign**.

## 4.1 Create a project

You can design and configure a project using **HPIDesign**.

To create a single station project:

1. Select **File→New→Project**, or click on  button from toolbar or press Ctrl+N keys.
2. Enter the project name and path in **New project** dialog.



**Figure 3) New project dialog**

This will create a folder with the project name in the selected path.



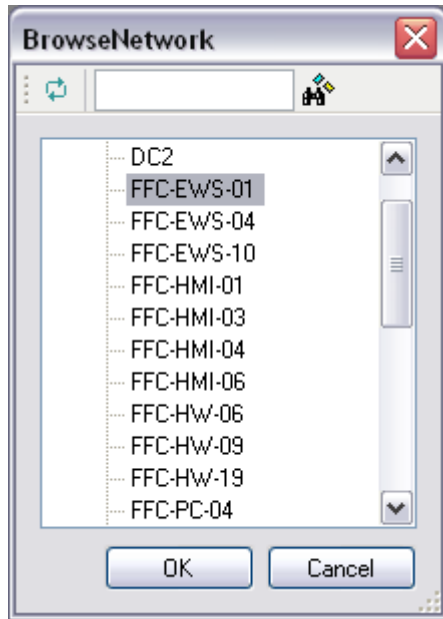
**Compact** version of **HPI** can only create this type of projects. Demo version of **HPI** is also a Single station version.

To create a multi station project,

1. Create a standalone project as described before.
2. Use **Deployment** window to define project components in the network. To know more about how to deploy a project on a network, see [1.6 Project Deployment](#).
3. Bind the project with **Project Server**, using **File→Project Server → Add to project server** command. This command makes a copy of project in the project server machine (which is defined in **Deployment** window). Now you have a multi-station project which is accessible from all other project components.

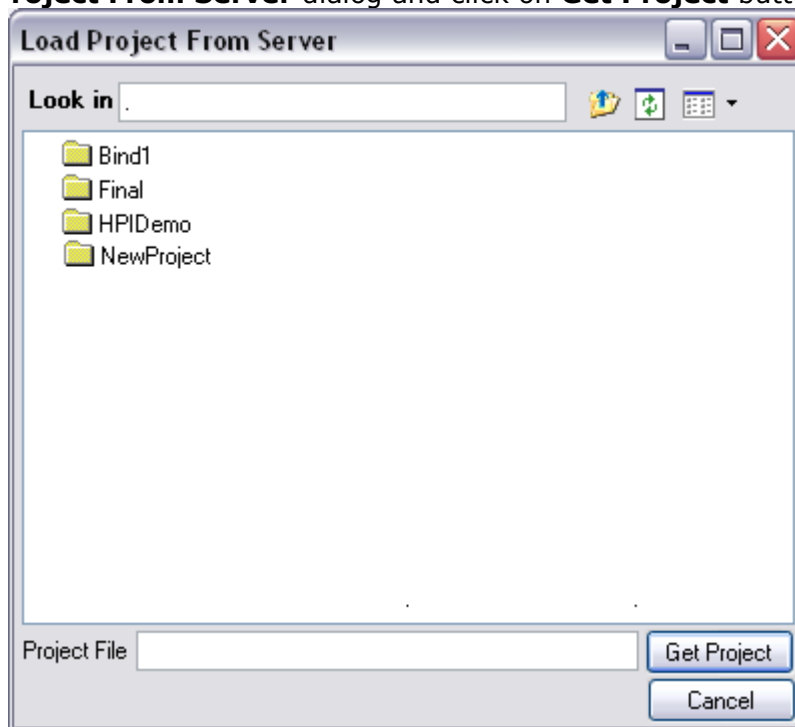
To open a multi station project in another engineering station for the first time:

1. Select **File →Project Server → Open From Project Server** command. Now select the project server machine from **Browse Network** dialog and press **OK** button.



**Figure 4) Browse Network dialog**

2. Browse for the project file (with the prj extension) you want to load in **Load Project From Server** dialog and click on **Get Project** button.



**Figure 5) Load project from server dialog**

3. Successful execution of this command makes a local copy of the project in your computer which is bound with the project in the project server. To read more about bounded projects and how to control them, see **File provider** chapter.

To get more information about multi station projects and how to create them, see the relevant chapter in application help.

## 4.2 Project setting

There are some settings which are related to each project specifically. These settings called Project Settings are categorized in the following groups:

**General:** It contains general information about the project and its history such as its physical location, creation date and version.

**UI security:** You can define keyboard shortcuts and security for **HPIRun** toolbar items.

**Run:** Here you can configure the options and behavior of the project in the run time such as the startup page, application help path, auto-save settings, etc.

**Alarm History:** In this tab you can define maximum length of text fields of alarms individually. If these sizes are not set, then maximum 10 characters of each field will be saved in the archive.

**AlarmStateColoring:** Here you can define font, colors and sounds of different kinds of alarms, based on their state and severity.

**Template:** Each project can have 4 graphic pages as its header, footer, left and right templates. These pages will be shown in appropriate places of each graphic page of the project in run mode.

**Alarm Categories:** In this tab you can define specific categories for the alarms and then add some attributes to them in addition to the default attributes of the categories.


Project setting is accessible from **Project** menu, **Setting** command. This command opens **ProjectSetting** dialog which has a tabular view for each group listed above.

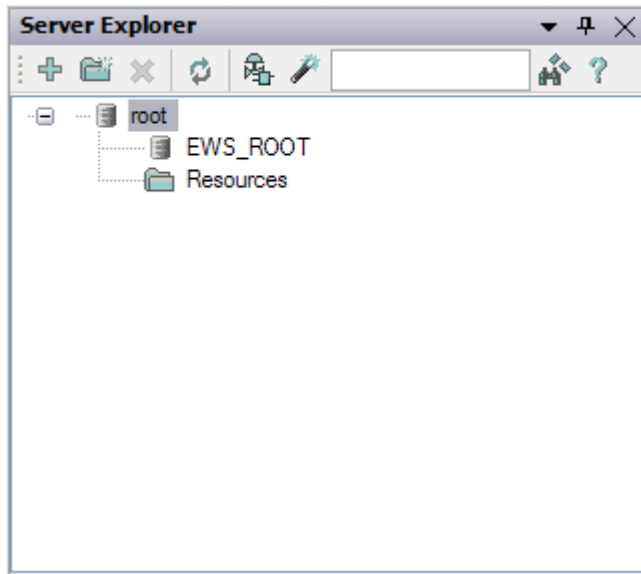
## 4.3 Server configuration

DXS Information Server (DIS) acquires information from production line and other sources and serves it for HMI clients.


The first step to DIS configuration is server structure definition, which means defining and naming the desired server nodes.

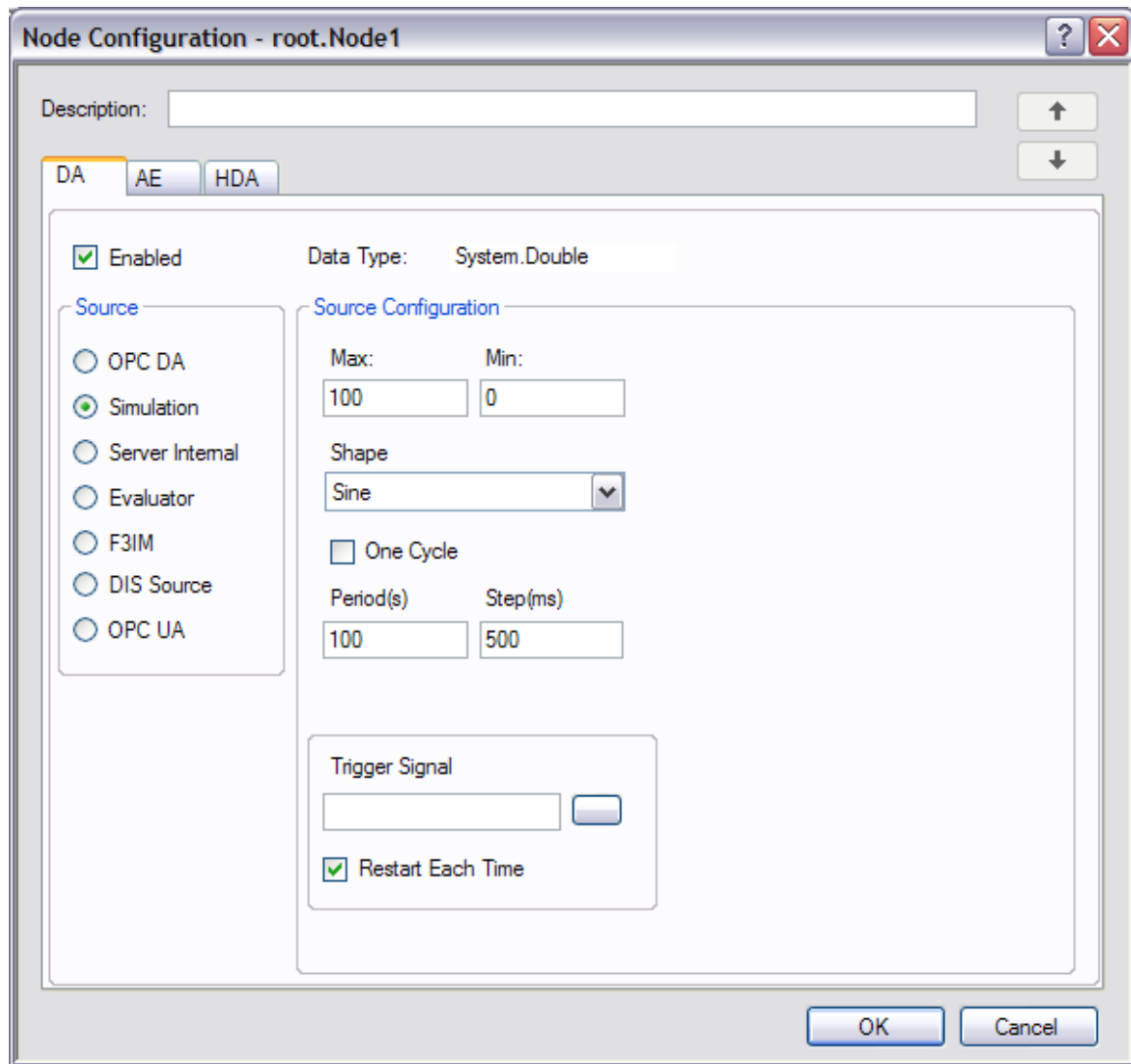
To create a node in DIS:

Open **Server Explorer** by selecting it from **View** menu or pressing  button in toolbar.



**Figure 6) Server Explorer**

1. Select the parent node.
2. Use  button to add a node under selected node.
3. Right-click on the node and select **Rename** to change the node name to a desired one (only English characters, numbers and “\_” are acceptable as the characters of the name).
4. Double-click on the node to open **Node Configuration** dialog and set the properties of the node.



**Figure 7) Node Configuration dialog**

**Save the configuration, using  button on Server Explorer toolbar.**

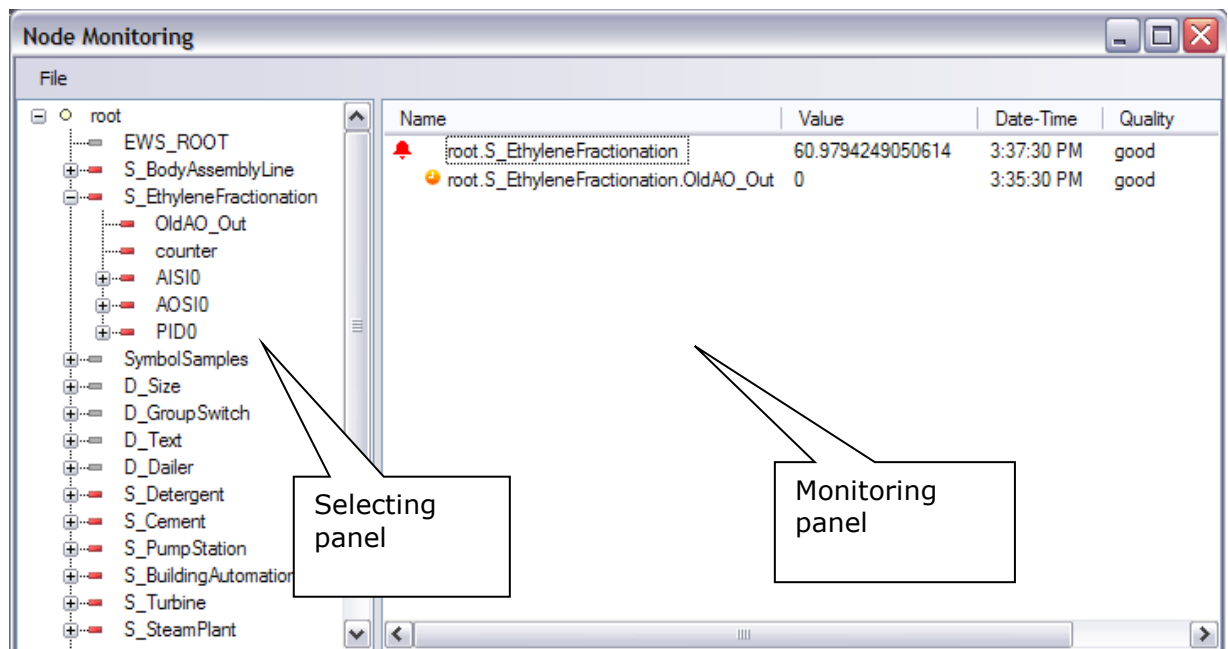
DIS has some tools and wizards to create all nodes connected to an OPC Server or a PPC all together, but in this project we don't have any device or OPC Server and we need to create simulated and internal nodes, Therefore we will create each node separately when we want to use it and configure it base on our needs.

## 4.4 Node monitoring

**Node Monitoring** helps project engineers for commissioning the project and assists maintenance staff for troubleshooting the project.

**Node Monitoring** provides us the ability to monitor the value and the quality of **DIS** nodes and to temporarily enable/disable them.

To open **Node Monitoring** tool, click on  button on **HPIRun** toolbar.




**Figure 8) Node monitoring tool**

**Node Monitoring** has two panels. In the Selecting Panel, there is a tree view of nodes; here you can select the node you want to monitor and double-click on it. Then, it will be added to the monitoring panel, where you can see its value and quality.

In monitoring panel, you will see the following information for each node:

- Node name.
- Current value of the node.
- Last read time/date of the node.
- Quality of the data.

You can change the column width by dragging the column dividers. When the mouse pointer is changed to , hold the left mouse button and change the column width as much as you want.

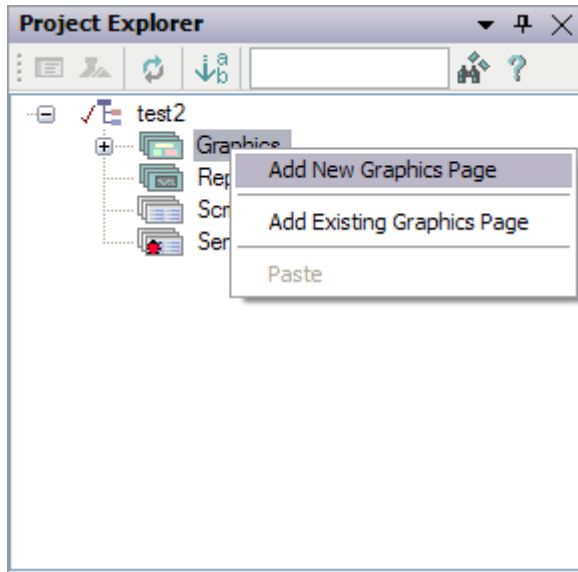
If you right-click on a node in monitoring panel, a popup menu appears in which you can enable or disable alarms, events and history of the nodes.

## 4.5 Graphic pages

In this section you will learn how to design graphic pages for a project.

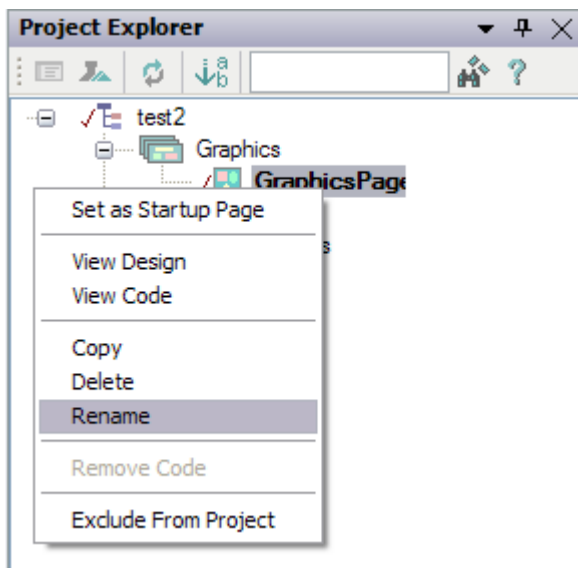
Open **Project Explorer** from **View** menu or pressing  button on toolbar.

1. Right-click on project name or **Graphics** category in **Project Explorer**.
2. Select Add **New Graphics Page** from the **popup** menu.



**Figure 9) Add new graphic page**

3. Right-click on the page which is added to the graphic pages.
4. Select **Rename** from the **popup** menu.
5. Type the name you want.

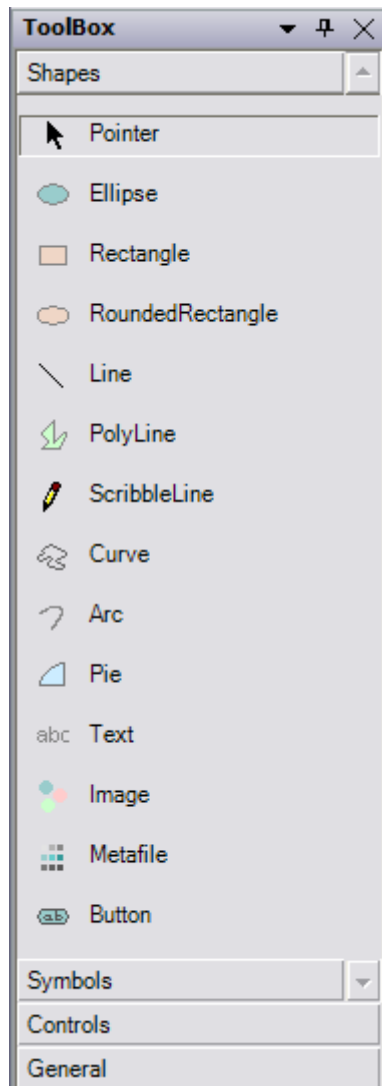


**Figure 10) Rename a page**

#### **4.5.1 Drawing a shape**

To draw a shape select **View Menu** → **Toolbox** → **Shapes tab**.

You can use different shapes to draw whatever you want. Shapes can be resized, rotated, flipped, colored and combined to make more complex shapes. Most of them include an adjustment handle that you can use to change the most prominent features of a shape.

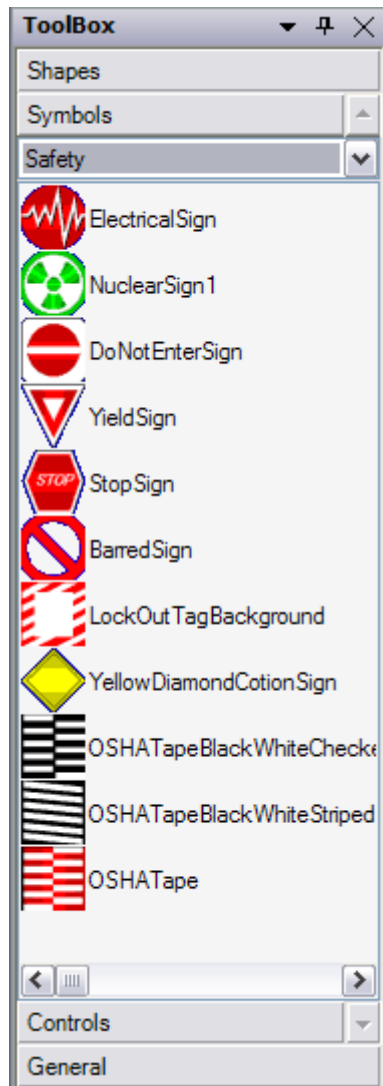


**Figure 11) Shapes tab of toolbox**

There are several properties assigned to each shape. You can change property via **property grid**.

#### **4.5.2 Using symbol library**

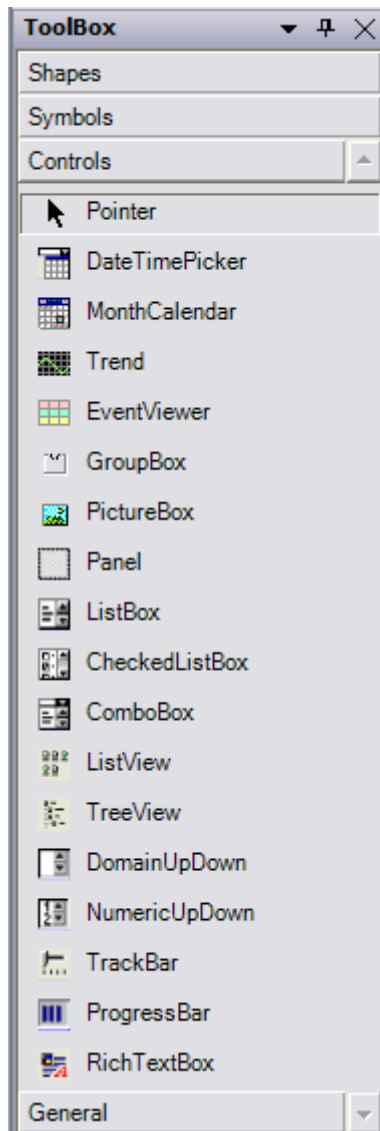
To use symbol library select **View Menu** → **Toolbox** → **Symbol Tab**. Then select your desired symbol and drag and drop it to the screen. Set its property via **property grid**. You can also create your own symbols.



**Figure 12) Symbols tab of toolbox**

### **4.5.3 Using windows controls**

To use windows controls select **View Menu** → **Toolbox** → **Controls Tab**. Click on your desired **Control** and it will be appeared in the page. Set its property via property grid.



**Figure 13) controls tab of toolbox**

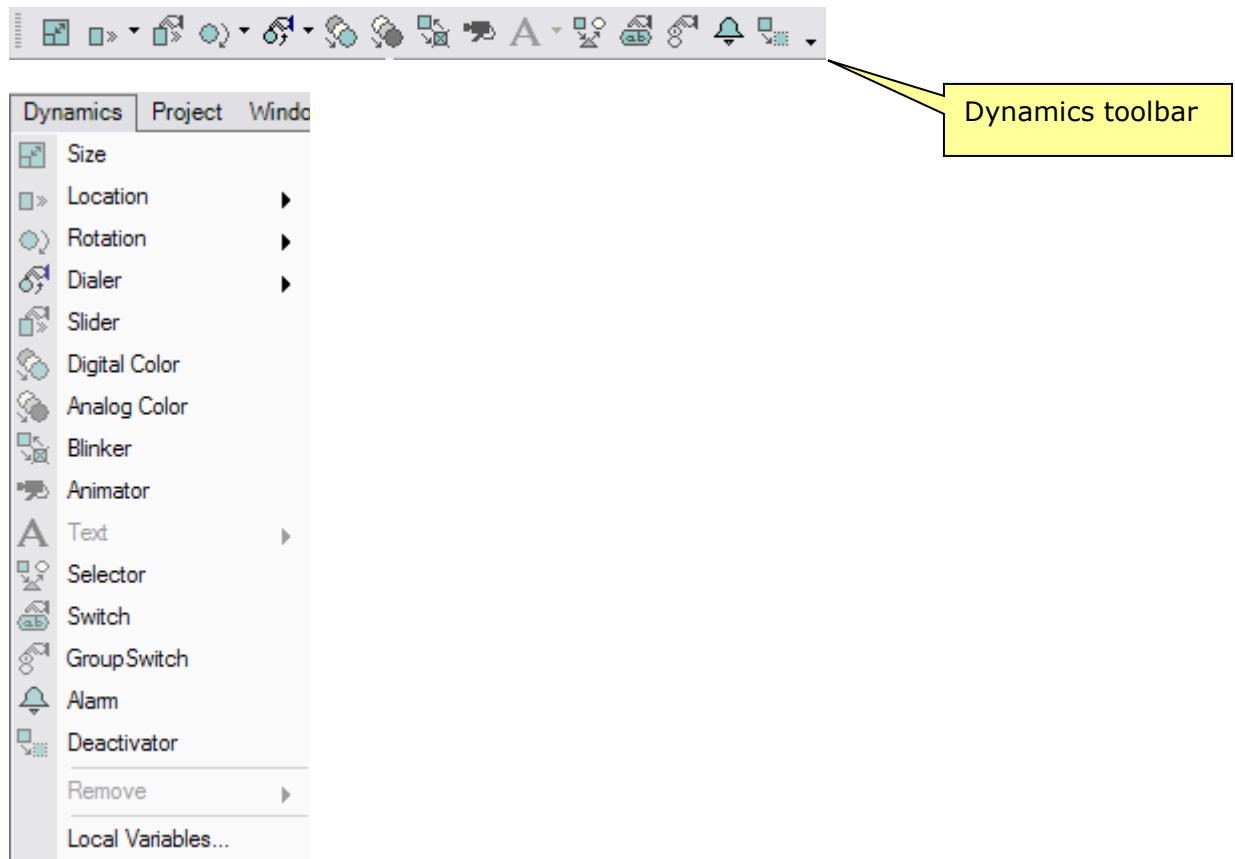
#### **4.5.4 Using Dynamics**

Dynamics transformers enable you to show objects animations and changes during the run time. Dynamic transformers apply specific transformation to objects in a graphic page, based on DAL server nodes value. For example you can create an object that changes color and size based on values from two different node connections. You can create these transformers through the **Dynamics** Menu or by clicking the appropriate button from the Dynamics Toolbar.

All dynamic transformers in **APATHPI** have a **source** property. You can assign a DAL server node or a local variable. So to use dynamics, at first step you need to create a DAL node or a local variable as dynamic source. Then to use dynamics:

1. Select the shape
2. Select your desired dynamic from **Dynamics menu** or press related button on the toolbar.
3. Set Dynamic property via **property Grid**

For more information please see "**About Dynamic Transformer**" chapter of APAT graphics.

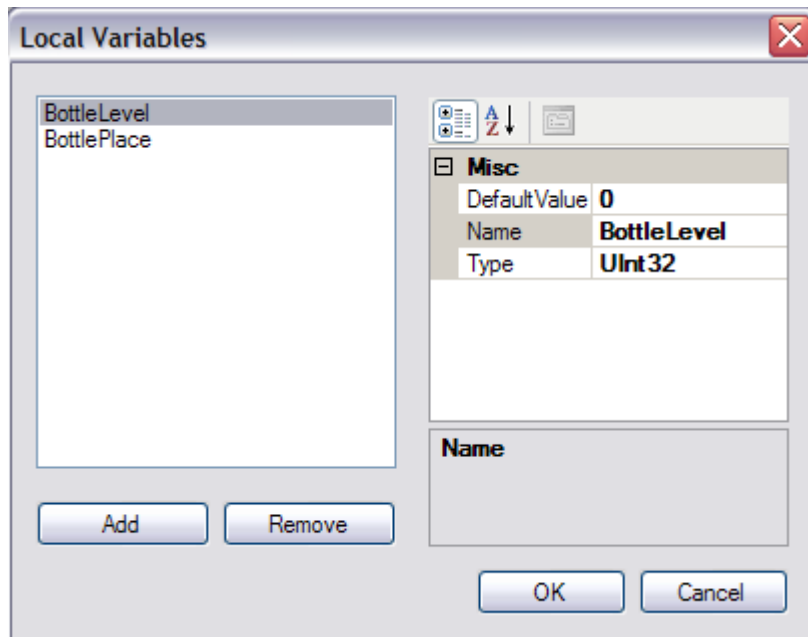


**Figure 14) Dynamics menu and toolbar**

#### **4.5.5 Add local variables**

In each graphic page, we can define local variables, which are accessible only in that page.

To add a local variable select **Dynamics Menu** → **Local Variables**. The **local Variables** dialog will be opened.



**Figure 15) Local variables dialog**

To add a new variable:

Click on **Add** button in **local variable** dialog box.

To remove a variable:

1. Click on variable in the **member** panel.
2. Click on **Remove** button.

## 4.6 Writing Scripts


Each graphic page can have a script page, which is run on **HPIRun** when the graphic page is open.

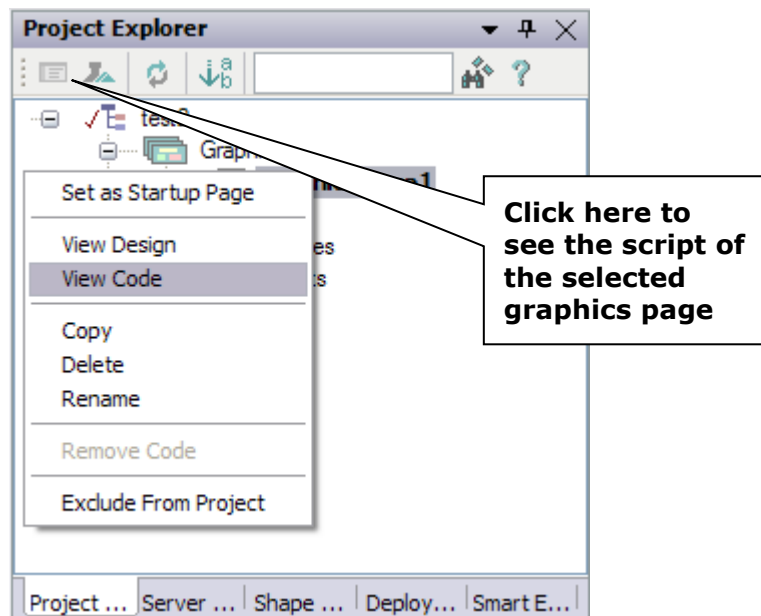
Graphic page scripts have some characteristics as follow:

- Graphic page script runs on its graphic page thread; but multi thread is possible with programming.
- Each graphic page script can be written in VB.Net or C#.Net.
- Variables, subroutines or functions, defined of project script level, can be used without considering its programming language.

### 4.6.1 Creating a graphic page script

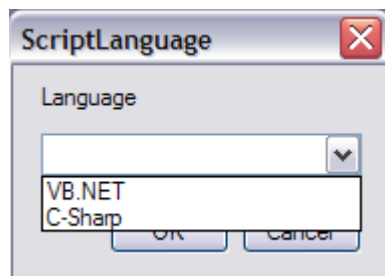
To create a script for a graphic page:

1. Select the page in **Project Explorer**.
2. Click on  button or right-click the page name and select **View Code** item from the popup menu.



**Figure 16) How to create a script page for a graphic page**

3. Select the programming language from **Script Language** dialog.



**Figure 17) Script Language dialog**

☞ You can't change the programming language of a script unless you remove and then recreate it.

## 4.6.2 Script module

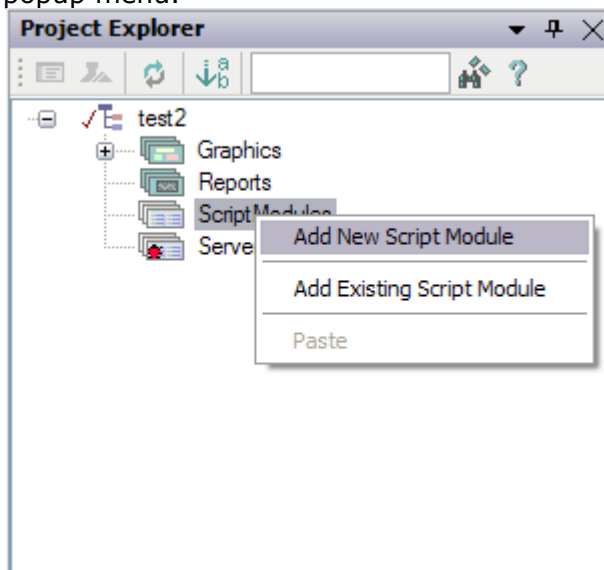
Script modules are containers of all functions, variables, classes... which are used all over the project.

Script modules have some characteristics as follow:

- Each script module can be written in VB.Net or C#.Net.
- There isn't any limitation in the number of variables, subroutines or functions in each module.
- They are easily called from other modules, independent of their programming languages.
- Access to static and non-static members, depends on the scope.
- Variables, subroutines or functions, defined in project script level, can be used without considering its programming language.

### 4.6.2.1 Create a script module

To create a new script module, right-click on the project name or **ScriptModules** branch on **Project Explorer** and select **Add New Script Module** item in the popup menu.

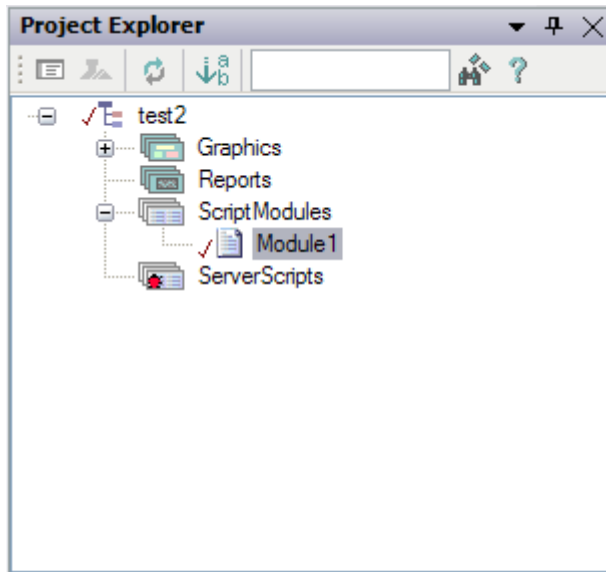


**Figure 18) Add a script module**

When you add a script module, **Script Language** dialog will be opened to ask for the scripting language of the created script page. You can select either VB.Net or C#.Net.

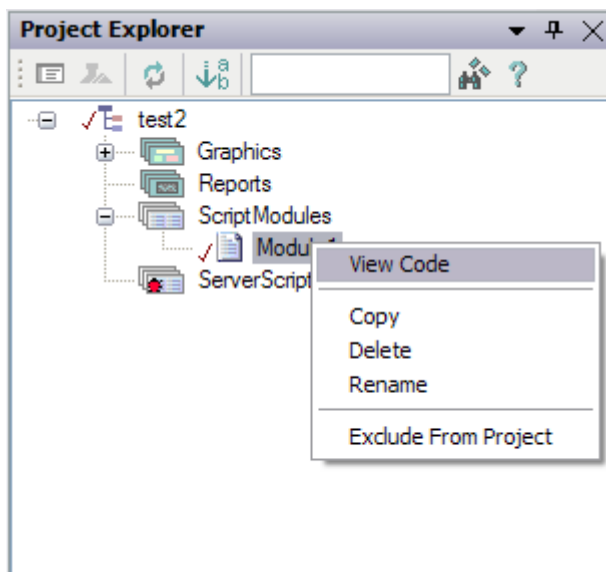
☞ You can't change the programming language of a script module unless you delete and then recreate it.

After selecting a language, a new server script page will be added to **ServerModules** branch on **Project Explorer** with the default name of **Module#**. To change this name, right-click on it and select **Rename** from popup menu.



**Figure 19) Creation of a script module**

To open a script module for editing, right click on its name in **Project Explorer** and select **View Code** from the popup menu.



**Figure 20) Server script popup menu**

To delete a script module, right-click on its name in **Project Explorer** and select **Delete** from the popup menu.

To exclude a server script from the project without deleting its file, right-click on its name in **Project Explorer**, and select **Exclude From Project** from the popup menu.

#### **4.6.2.2 Using members of a script module in another script page**

After you create, write and compile a script module, you can use its members in the other script pages. To do this:

- Add the module to the script page references.
  - Instantiate an object from the module class.
  - Use the module members using the instance.

```

Public Sub Main()
dim m as new Module1
m.Test
End Sub

```

**Figure 33) Using members of a script module**

### 4.6.3 Writing a Simple Script

Now you can begin to write your first script. Here we begin with the script of a graphic page.

Insert a **Text** shape in the graphic page and open the page script.

The script has a main body which begins with:

**Public Sub main()**

And ends with



**End Sub**

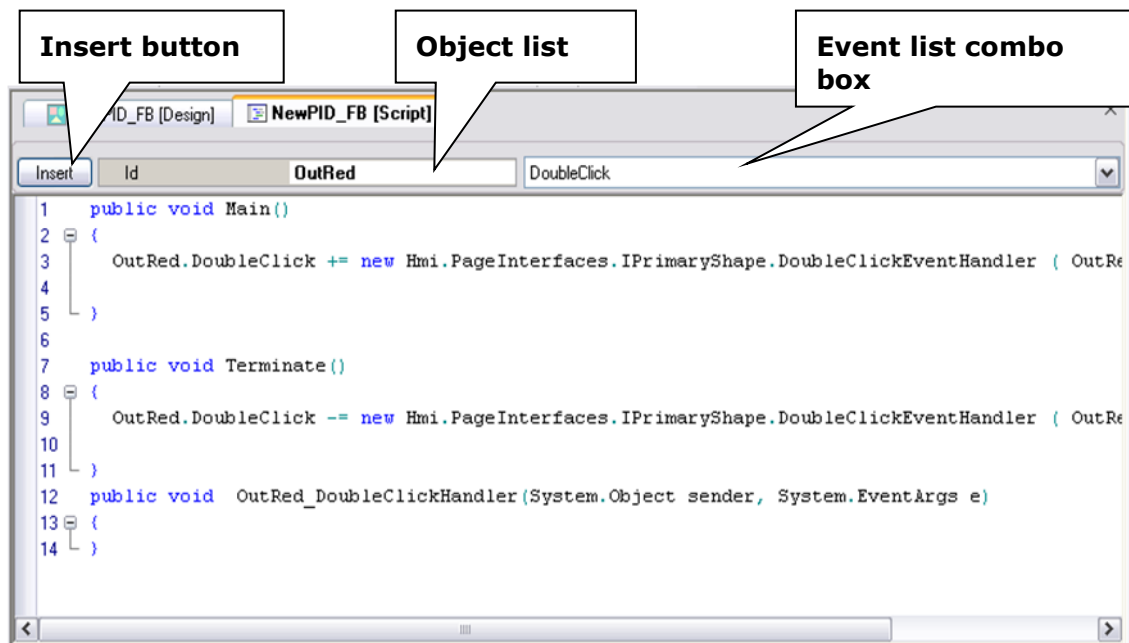
For the simplest script you can easily select the Text shape which you have inserted in your script page in the object list (up left combo box) and press **Insert** button.

The name of that text is inserted into the script.

Now type a "." after it and you will see the list of all text object members.

Select **Text** property, type an "=" sign and then type "Hello World" string.

Now you have finished the code. Compile it using  button and Run the project using  button. You can see that the text shows "Hello World".



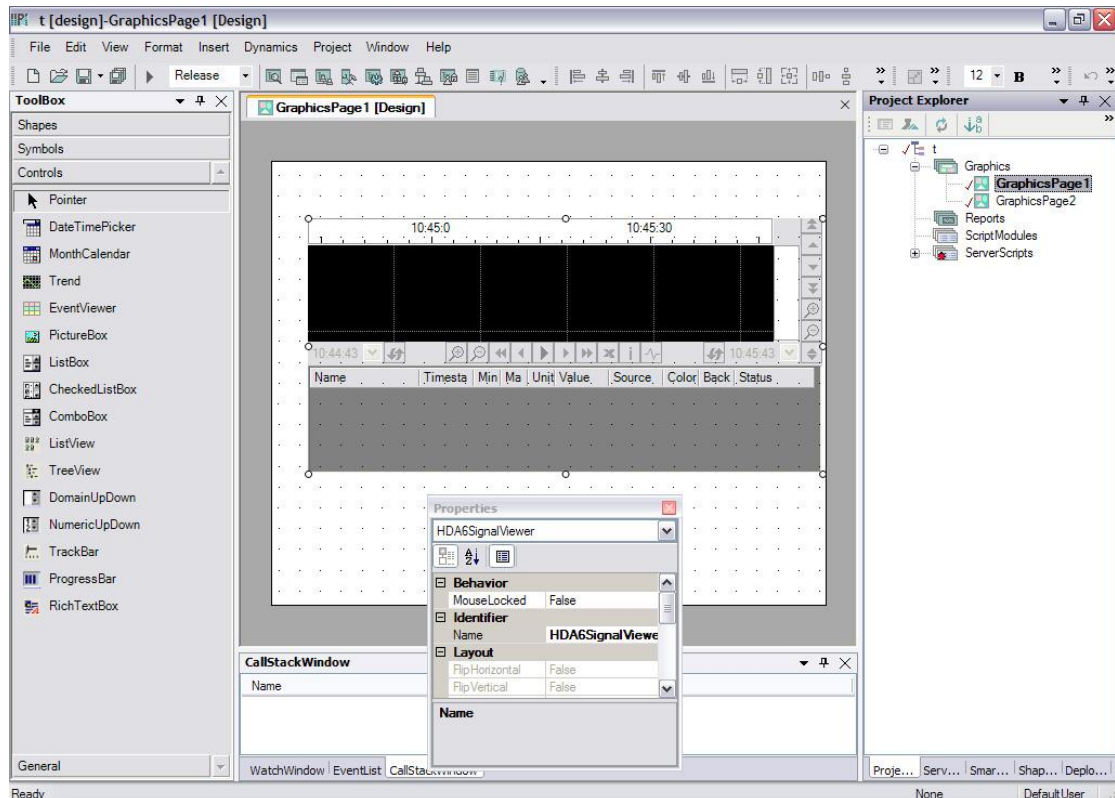
**Figure 21) Sample script and script page tools**

## 5 Historical Data Viewer

There is a trend control available in the **control** tab of toolbox for viewing the trend of historical data in a graphic page, which can be embedded in a graphic

page. This section defines how to use and configure this control for showing trend of a desired signal.

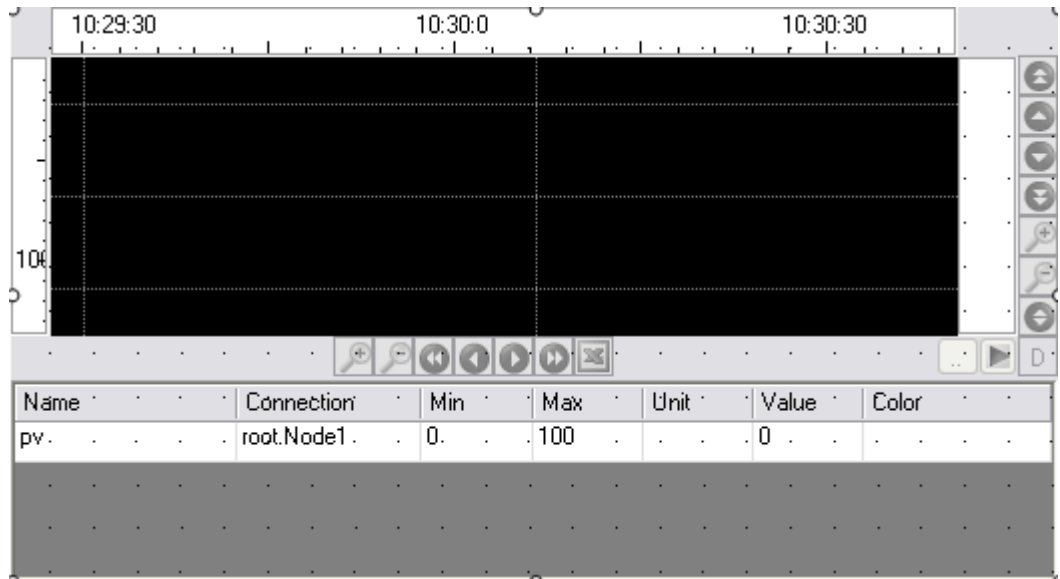
Each signal in a Trend control has a specific color, as defined in properties. To select the **Trend** Control in design mode choose it from **Controls** category of **Toolbox**. By clicking **Trend** control it will be pasted to the graphic page which is already open.



**Figure 22) Trend Control in a graphic page**

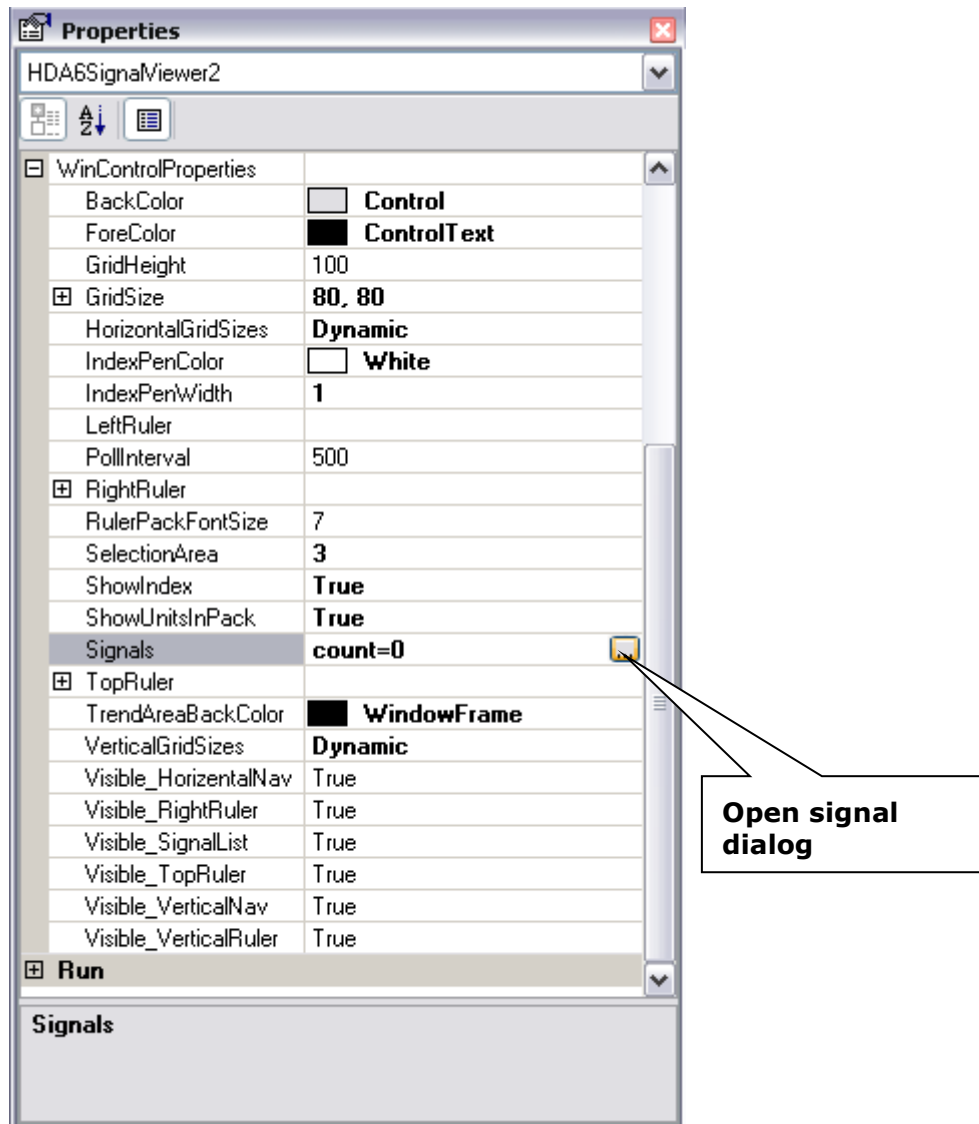
Trend control consists of 2 panels. Upper panel shows trend of the signal and in lower part, signals which we want to be viewed in trend control are defined. You can add signals to a **Trend** control or remove anyone through **Signals** property in the Property-Grid.

When you add a signal to trend control, a line will be added to the lower part which contains some information about the signal. A left Ruler also will be visible after adding each signal.



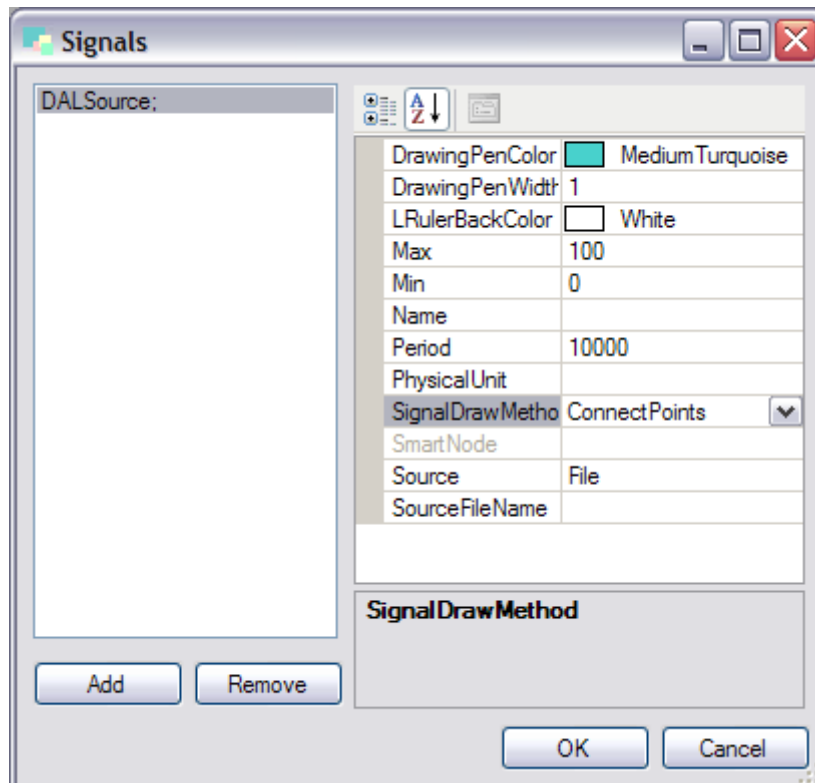
**Figure 23) Trend Control after adding a signal**

To add signals to a **Trend** control (or remove any) click the  button in front of **Signals** property in **WinControlProperties** section as the following figure:



**Figure 24) Open Edit Signals Dialog**

You can use **Add** and **Remove** buttons to add new signals and delete existing ones. Selecting each signal will result in showing properties of that signal in the property grid on the right. In signal list editor click **Add**, and click **HDA Signal** to view properties.

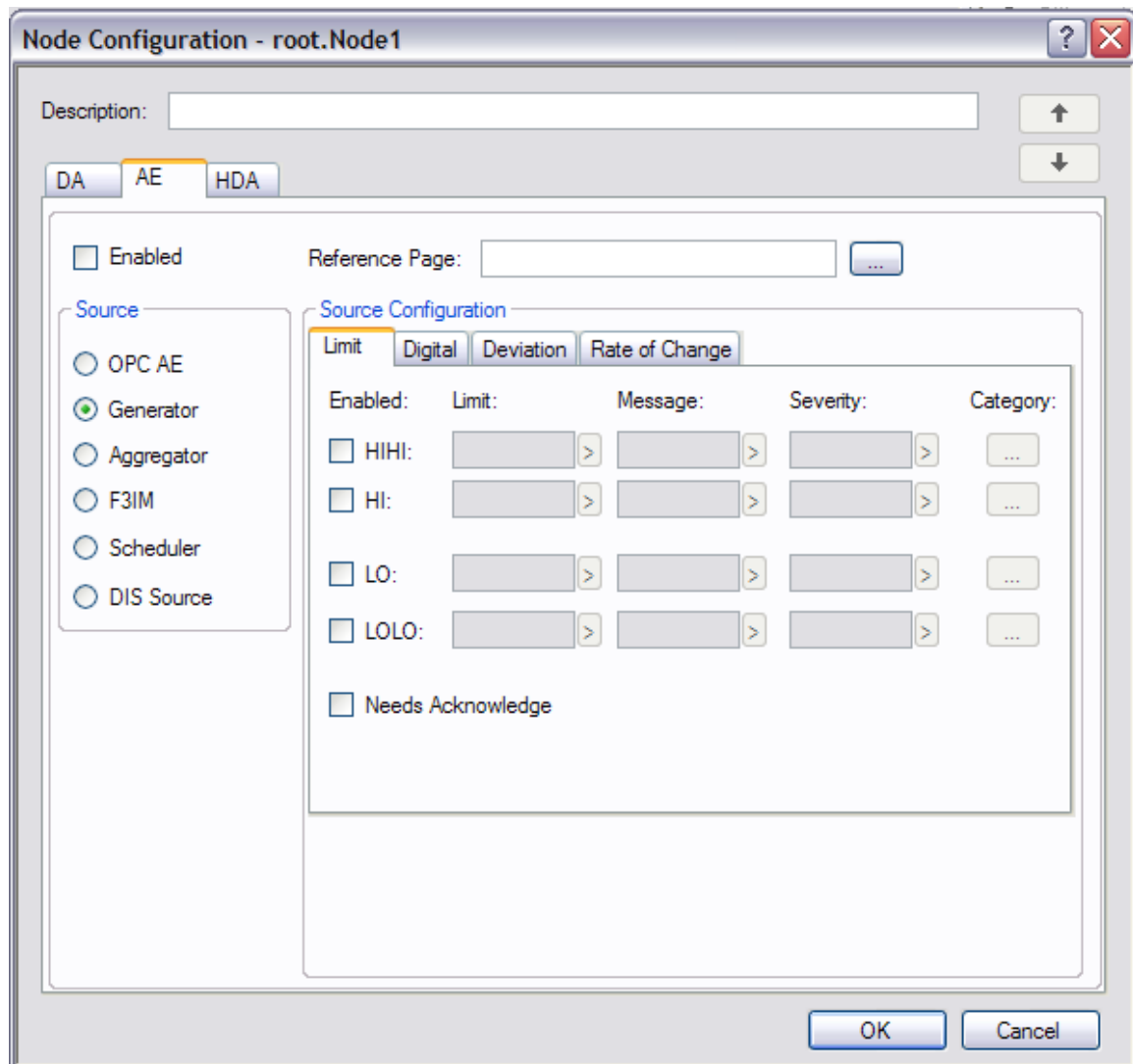


**Figure 25) Add a signal to Signal List Editor**

There is an easier way available for adding a connection to trend viewer; you can drag your desired node from the node monitoring window directly to the trend viewer in the run mode. The signal will be added to the trend viewer and you can change its color for better display by clicking on signal button on the trend viewer and setting the related property.

## 6 Alarm configuration

DXS Information Server (DIS) acquires information from production line and other sources and serves it for HMI clients. Part of this information is alarm information, so alarm definition is part of DIS configuration. Each node in **Server Explorer** has **AE** tab, here you can enable alarm system for the node.



**Figure 26) AE tab of Node configuration**

## 6.1 Create Alarm

To Create an Alarm:

1. Create a node as source of alarm.
2. Double-click on the node.
3. Set DA tab for providing data and AE tab for indicating alarm on specific situations.
4. Select the AE tab and then choose your desired data source from the left group box and then fill in the required fields to set your alarm successfully.

See the above figure.

Different sources of alarms are available for nodes:

- **OPC AE:** Alarm state generated by an OPC server.
- **Generator:** Alarm generated in **DIS** based on defined condition.
- **Aggregator:** This option lets the user to aggregate defined alarms.
- **System events:** Events are generated by the **HPI** system.

- **Scheduler:** Events are created based on defined schedule.
- **Dis Source:** Alarms and events are generated by another node in another **DIS**.

## 6.2 Configuring alarm coloring

To see alarms with different severity and states in different colors you must set some properties in project setting. There is a default setting available; but if you want to change this setting, open **project setting** from **Project** menu → **Settings**. Then open **Alarm State Coloring** tab and set color for each state.

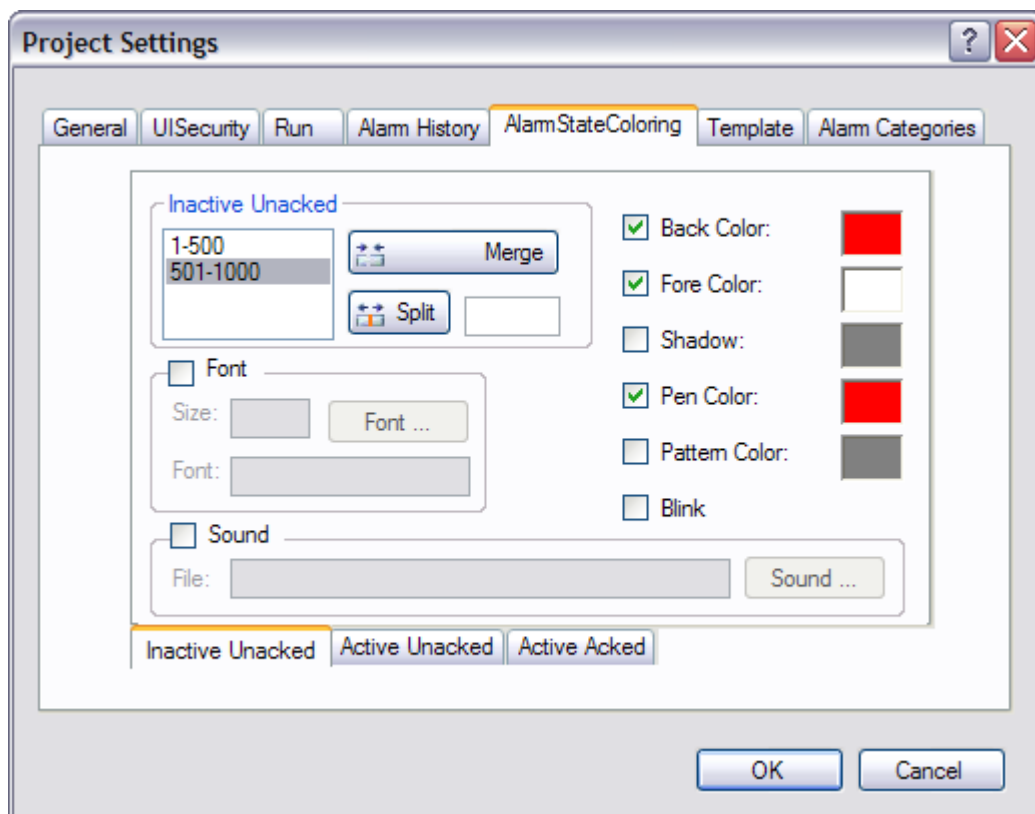


Figure 27) Alarm state coloring

## 7 Report configuration

**HPI** provides an easy to configure report designer, based on **Microsoft Excel**. **HPI** has different built-in reports which user can use them without doing any programming.

You can design a report in **MS Excel** in design mode and configure it to be generated on a defined schedule or on demand in run mode.

Built-in reports are those which get data from **DIS** and consist of:

- Real Time Data
- Historical data
- Multi Historical Data
- Alarm and Event Log

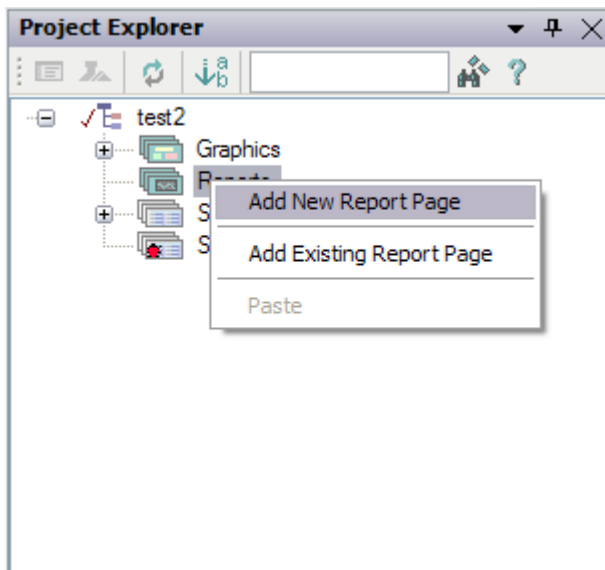
- Duration Event Log
- Aggregate Duration Event Log

There are other reports available on relational data bases which can show results of a query on a database.

There are also some built-in reports which get configuration data of **DIS**. We call these types of reports "Design mode reports".

## 7.1 Create a new report page

To create a new report page, you can select **Add New Report Page** from **File** menu or right-click on the project name in **Project Explorer** and select **Add New Report Page** from the popup menu. This will add a new report page under the **Report** category in **Project Explorer**.



**Figure 28) Adding a new report page**

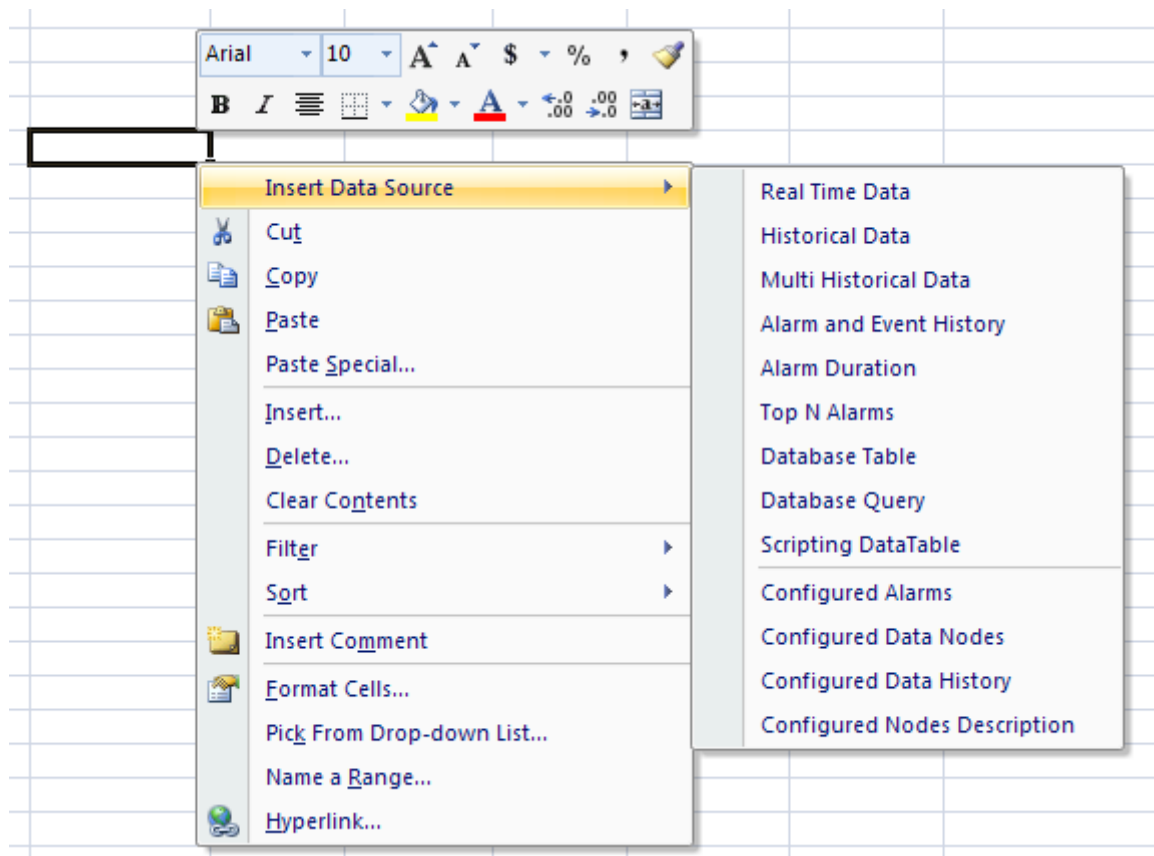
To rename a report page right-click on its name in **Project Explorer** and select **Rename** from the popup menu.

## 7.2 How to configure a report

To open a report page for configuration, right click on its name in **Project Explorer** and select **View Design** from the popup menu. This action opens the page in **Microsoft Excel 2007**. Now you can add deferent data sources and configure your report.

## 7.3 How to insert a data source

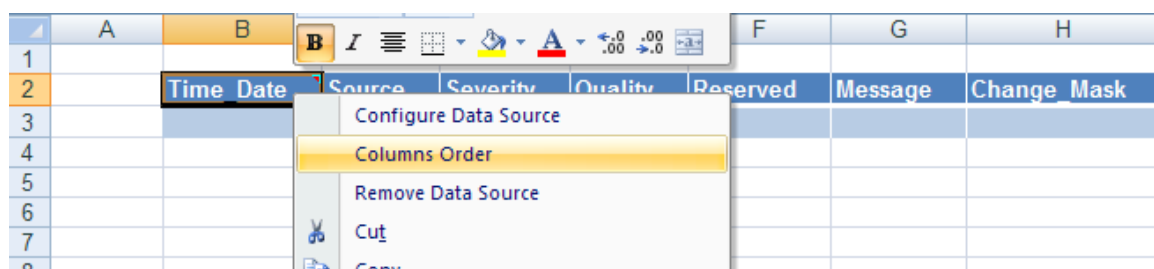
To set a cell's data source, right click on the appropriate cell. Then select the desired data sources.



**Figure 29) Right-click and select a Data Source**

## 7.4 Order Columns in a Data Source

After a data source binds to a cell, a header row will be inserted, based on a default setting of the selected data source. You may want to remove some columns or change their order. To do this, right click on the cell and select Columns Order from the popup menu.



**Figure 30) Select Columns Order**

A list of available columns (based on the data source) will be shown in **Select Columns** dialog. Uncheck unwanted columns. You can use **All/None** button to check/uncheck all columns.

To change a column order, select it on the list and use **Up/Down** button to move it upward/downward.

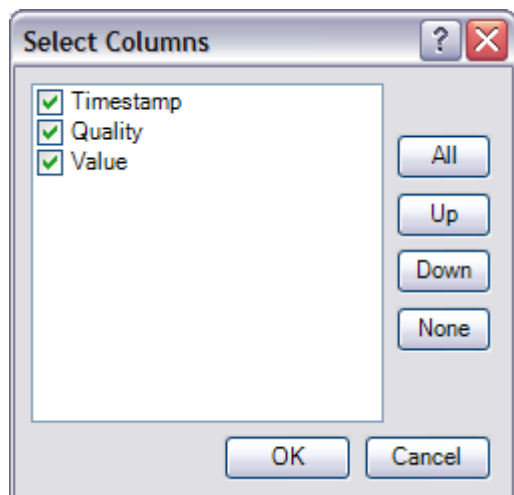


Figure 31) Select Columns Dialog

## 8 Working with smart objects

### 8.1 About smart objects

**Smart objects** are developed in order to create a template for an object type which contains all aspects of that object (including its symbol, data, alarms, faceplates, documents and control strategy) and to create unlimited object instances from one template. If this template changes, all instances will change as well; that's why we call them smart.

Smart objects are the key of integration of **HPI** with **DXS** supervisory system. Each Function block in **DXS** library is actually the control strategy aspect of a smart object in **HPI** library

**Smart objects** reduce development time of a project dramatically and ease the change management of the project.

HPI has a library of smart objects which can be extended by the user.

### 8.2 Aspects of smart objects

In automation world, various things can be considered as an object. It is the same in **HPI**. Equipment or control loops both can be examples of smart objects in **HPI**.

Each smart object in **HPI** has the following aspects:

- **Data:** Consist of information about the object, acquired from control system by **DIS**. It is also possible to define Server variables as data of a smart object.
- **Alarms:** Alarms of the smart object are defined on the smart object data. They can be generated by **DIS** server or can be received from the control system.
- **Graphical symbol:** Each smart object can have a graphical symbol, which represents the object in graphic pages.
- **Faceplates:** Each smart object can have unlimited faceplates which can be open in run mode by clicking on a defined part of the smart graphical symbol.
- **Documents:** Each smart object can have unlimited documents (data sheet, maintenance manual...) which can be shown in run mode. To open these

documents, it is necessary to have appropriate software installed on the workstation.


- **Control strategy:** Each smart object can have a Function Block as its control strategy. Main data of the smart object is also defined by this function block parameters.

☞ Control strategy is only available for those smart objects which are controlled by F300 devices.

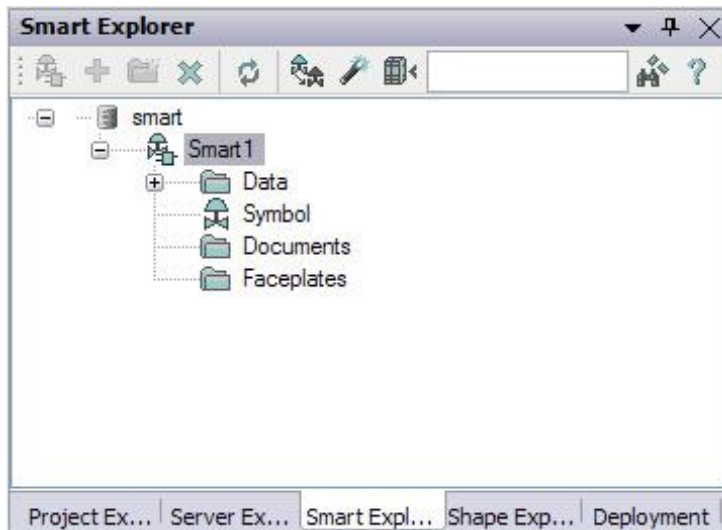
☞ Depending on the usage of a smart object, it may have one or more of these aspects.

### 8.3 Create a smart object template

In the **HPI** environment, **Smart Explorer** is used to create, edit and delete smart objects of a project. To activate and open **Smart Explorer**, select it from **View** menu. You can organize smart objects via smart explorer.

To create a smart object, press  button. A new smart object will be created. If you click on + sign next to it on the explorer, you will see its four folders:

- **Data:** In this folder you can define data, alarms and the function block of the object.
- **Symbol:** Here you can create the graphical symbol of the smart object.
- **Documents:** In this folder you can add different documents related to the smart object.
- **Faceplates:** In this folder you can define faceplates of the object.



**Figure 32) Add a smart object template**

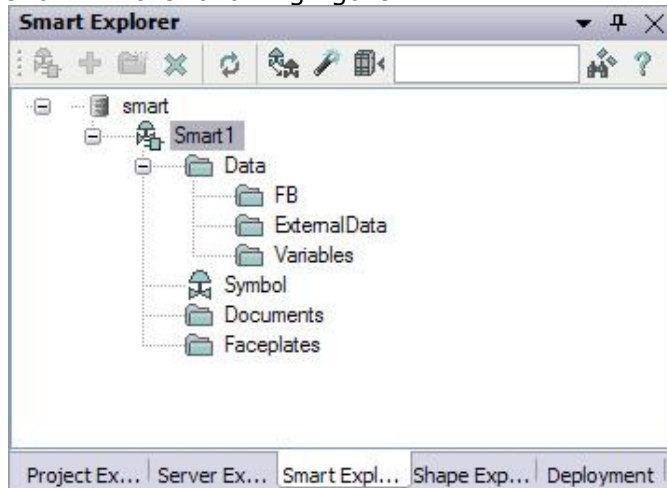
To rename the object, right click on its name and select **Rename** item from the popup menu or simply click on the name.

### 8.4 Create data of a smart object

Data of a smart object can be taken from 3 different sources:

- A function block in the correspondent PCE project.
- An OPC server.
- DIS.

For each smart object there are these categories available under its data folder as shown in the following figure.



**Figure 33) Data categories of a smart object**

The same as all the data of the project, smart objects data are defined as nodes in **Smart Explorer**. You can use **+** button to create a node under the selected node and use **X** button to delete the selected node in each category. So you have to create all nodes one by one and set their properties manually; but it is also possible to assign all data of a function block to the smart object or use **OPC wizard** to acquire them from an OPC server. In this way you don't need to set the properties manually, they will be read from their source automatically.

## 8.5 Defining alarms and events of a smart object

As mentioned before one of the aspects of smart objects is their alarms. Similar objects have similar alarms, so it is normal to define them in the smart object template. In this way, when a smart object instance is created, its alarms will be created automatically.

The same as all alarms, smart object alarms are served to HPI clients by **DIS**. Depending on where the alarm is generated, its configuration may be different.

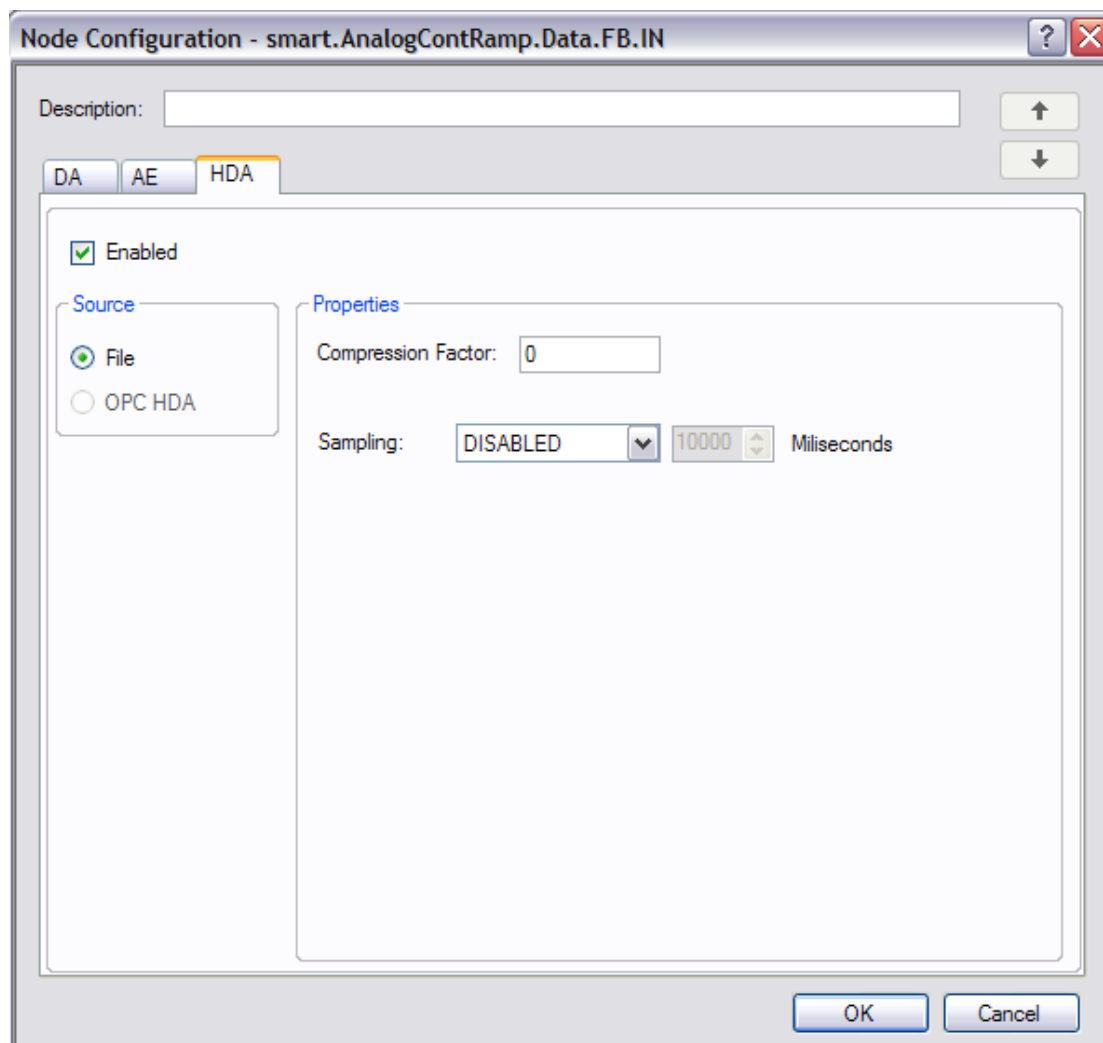
## 8.6 Smart object nodes archiving by DIS

If you want a node of smart object instances to be archived by **DIS**, you can enable **HDA** source of the node in the smart object template. In this way when you create an instance of the smart object the correspondent node will be **HDA** enabled.

To set **HAD** source of a smart object node:

1. Open **Node Configuration** of the node by double clicking on its name.
2. Open **HDA** tab of **Node Configuration**.
3. Check **Enable** box.
4. Select appropriate setting. To know more about these setting see [Configuring HDA source of a node](#) section in **DXS Information Server** chapter.

5. At the end, save your work using  button on **File** toolbar.



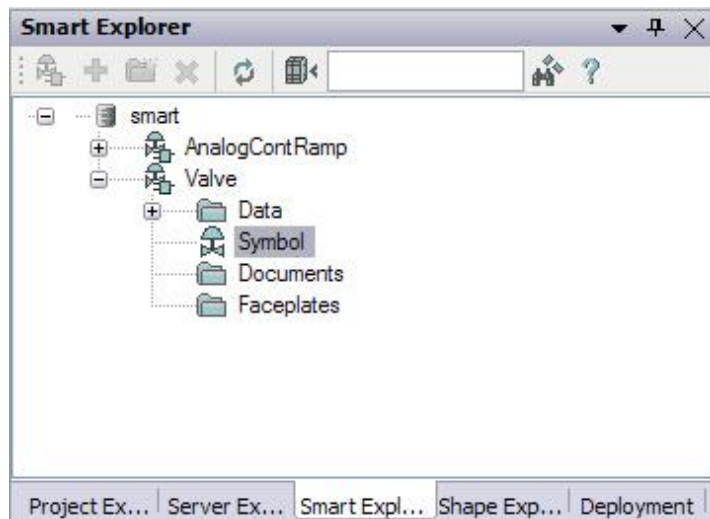
**Figure 34) Enabling DIS to archive history of node in an smart object template**

## 8.7 Defining symbol of a smart object

Each smart object can have a graphical symbol which represents the smart object instances in graphic pages. This symbol has all shapes, dynamics and scripts which are defined in its template and bound to the smart object data seamlessly.

### 8.7.1 Defining the graphical view of the smart object

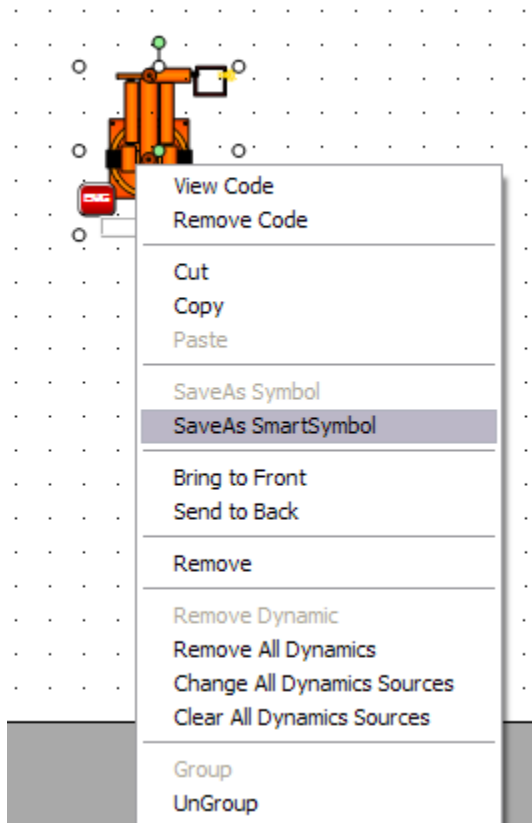
To define the graphical view of a smart object, double-click on its **Symbol** category on **Smart Explorer**. This action will open an empty graphic sheet with the name of the smart object.



**Figure 35) Opening a graphic page to design the symbol of the smart object**

Same as designing a graphic page, in this sheet all graphical tools and facilities are available for designing the symbol of smart object.

1. Design graphic for the smart symbol. (According to **HPI Graphics** chapter of the **HPI Manual**)
2. Group all parts of the symbol.
3. Right-click on the symbol and select **Save As Smart Symbol** item from the popup menu. After this action the group will be saved as the symbol of the smart object.



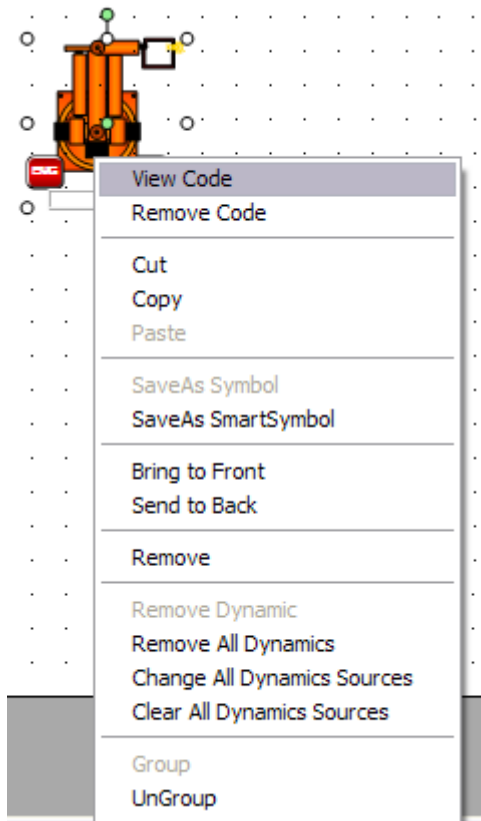
**Figure 36) Save a group as the symbol of a smart object**

### **8.7.2 Defining the script of the smart object**

Graphical symbol of a smart object template can have a script. When an instance of a smart object is created and its symbol is dragged into a graphic page, the script of the symbol is added to the graphic page script automatically and seamlessly.

To create script for the symbol of a smart object:

1. Open the symbol of the smart object by double-clicking on its **Symbol** in **Smart Explorer**.
2. Right-click on the group which is defined as the smart object symbol.
3. Define script for the smart as explained in **Writing Scripts**.





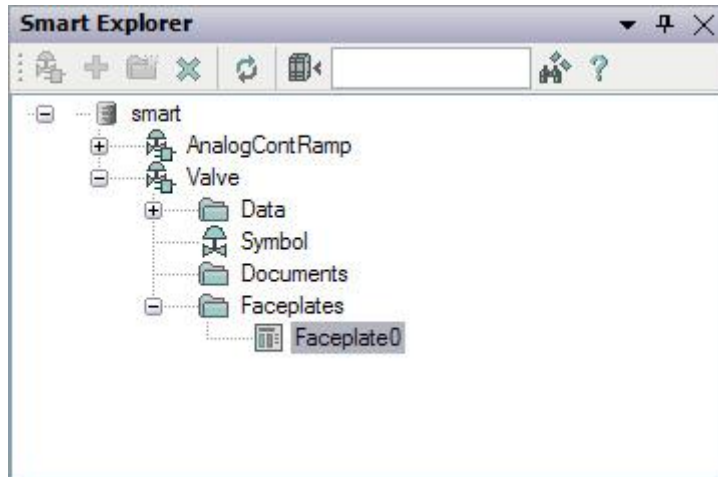
**Figure 37) Create/Edit Script of a smart object symbol**

## 8.8 Adding a faceplate to a smart object


Each smart object can have unlimited faceplates. Each faceplate of a smart object is a graphic page which has access to the nodes (as a result also data, alarms and history) of the same smart object.

To create a faceplate for a smart object:

1. Select **Faceplate** category of the smart object in **Smart Explorer**.
2. Press  button, a faceplate will be added to the smart object.
3. Right-click on the faceplate on **Smart Explorer** and select Rename from the popup menu or simply click on the faceplate name and change the name to a desired one.
4. Save the name using  button in **File** toolbar.




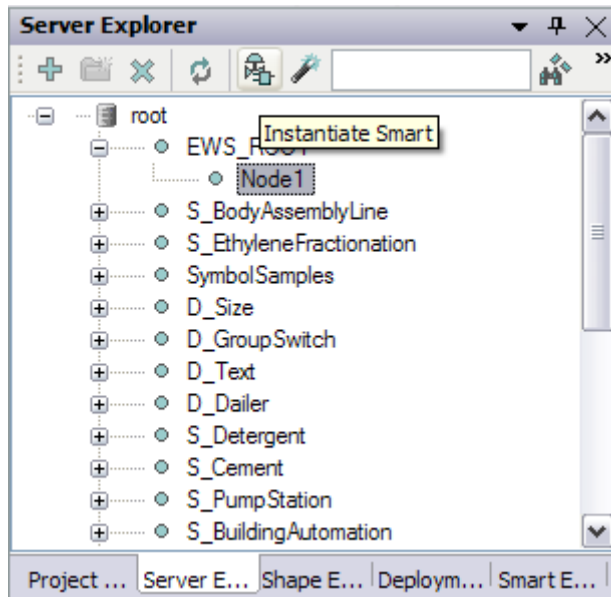
**Figure 38) Adding a faceplate to a smart object**

5. By double clicking on faceplate name in **Smart Explorer**, a black graphic page will be open.
6. Design the faceplate using all graphical tools and facilities as normal graphic pages.
7. Since the smart object faceplate can only link to its own data, only nodes of the same smart object are available in all dynamic transformers where you want to select a node.
8. Save the graphic page using  button on toolbar or select **Save** item in **File** menu.

## 8.9 Create an instance of a smart object

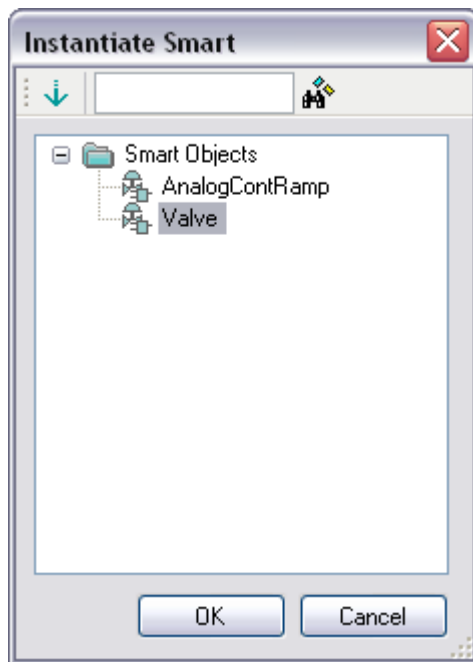
To create an instance of a smart object:

1. Open **Server Explorer**.
2. Select the node you want to create the smart object instance as its child.
3. Click  button to open **Instantiate Wizard**.



**Figure 39) Create an instance from a smart object**

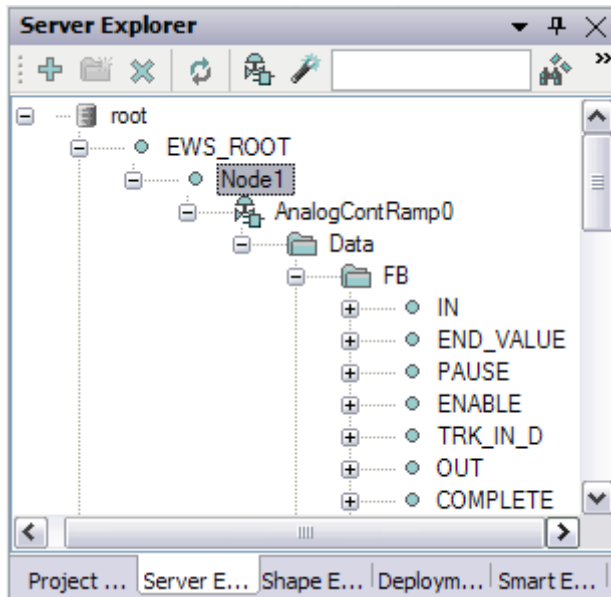
4. Select the desired smart object and press **OK** button.



**Figure 40) Instantiate Wizard**

5. An instance of the selected smart object will be created under the selected node. Right-click on its name and select **Rename** or simply click on its name and type the desired name for the instance.

As you can see in the following figure all nodes of the smart object has been created automatically.



**Figure 41) Smart object instance in Server Explorer**


To insert the symbol of a smart object instance on a graphic page:

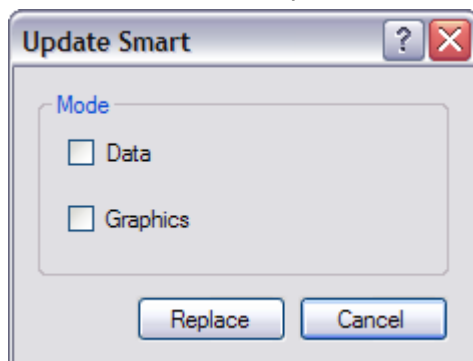
1. Open the desired graphic page.
2. Select the smart object instance in **Server Explorer**.
3. Drag the smart object and drop it into the graphic page.

## 8.10 Update smart instances

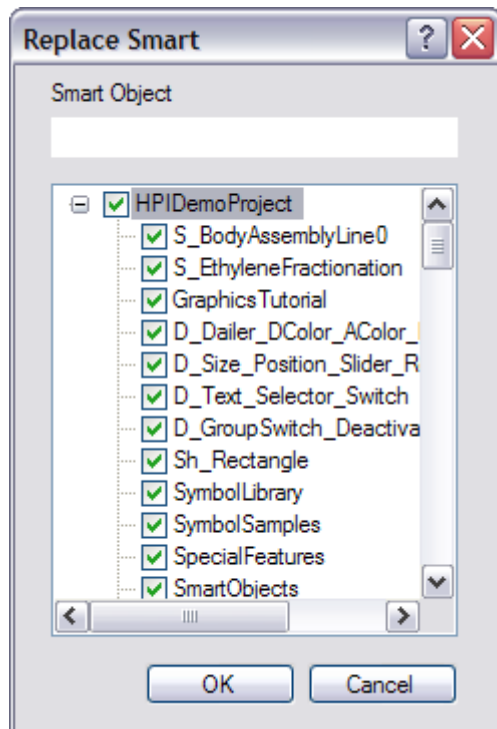
If you change a smart object template which you have already created instances of it in the project, you have to update these instances otherwise these instances will remain based on the old template.

To update smart instances based on the new template:

1. Select the smart object in **Smart Explorer**.
2. Click on  button on toolbar, this will open **Update Smart** dialog. If you have changed the symbol of the smart object, check the **Graphics** option, the **Replace Smart** dialog will open and you can select the desired pages to update. If there is any change in the data of the smart, check **Data** option.




**Figure 42) Update Smart dialog**




**Figure 43) Select the graphic pages**

3. Click **OK** button.

This action deletes all nodes and symbol of all instances of the selected smart object in **Server Explorer** and selected graphic pages and recreates them on their previous location. In relation to the symbol, the new symbol will be created at the previous size, position and rotation. In relation to the nodes, the new nodes will be created with the previous name and position on the tree of nodes and data address (Device, Function block and OPC group item).

 As mentioned before, this action recreates nodes and the symbol for all instances of the smart object, so any manual change on these nodes or symbols will be lost.


 If you want to apply changes on the script of a smart object in a project in run mode, you need to compile the code of each page that contains one or more instance of the smart too; then refresh the page in run mode.

## 9 Project Deployment

As discussed before there are two types of projects:

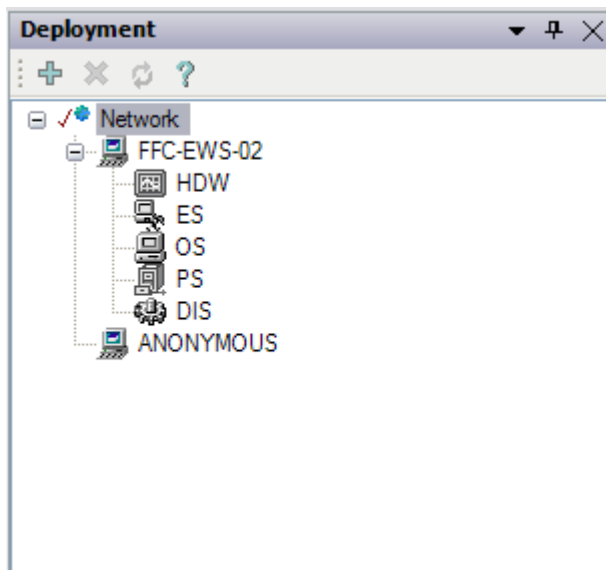
- Single Station
- Multi Station

In order to create a Multi Station project, you need to deploy the project on desired nodes on the network. **Deployment** explorer lets you configure each HPI package and deploy the project on them.




 Deployment settings are unique for each project.

## 9.1 Adding/removing network nodes (stations)

When you create a project, at the beginning it is a single station project so all the necessary packages are on the same PC station; so if you open **Deployment** window, only one PC is available there. In order to change this project to a multi station one, first you need to add the network nodes which you want to deploy the project on them.





**Figure 44) Deployment Explorer**

To add a new network station, select the network and click on  button. The **Add New Node** window will open which shows list of the available network nodes where you can choose a station to add. You also can use  button to search for a specific node. If the desired node is not available you can use  button to define a new node which must be later bound to the network. Finally press ok and the node will be added to the network.



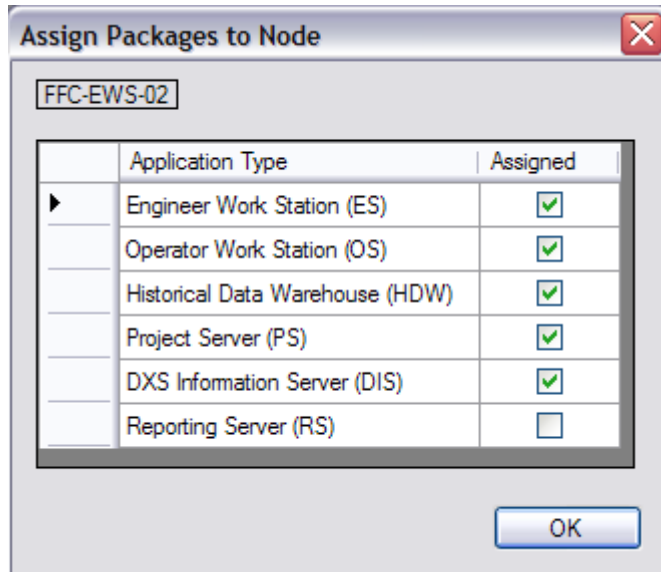
**Figure 45) Add New Node**

To remove a station, select it on **Deployment** explorer and press  button.

 If necessary, press  button for refreshing the network tree.

## 9.2 Assigning Packages

Each station in the network will play a specific role in relation to a multi station project. These roles are categorized into some packages which should be entitled to each machine. For assigning packages to each station, right-click on the station node and select **Assign Packages** item from the popup menu.



**Figure 46) Assign packages**

☞ If the project has already been bound to a project server, for assigning packages to the stations, first you have to right-click on the **Network** node and *Check-Out* the network, unless it has already been checked-out.

This action opens **Assign Packages to Node** window. You can select your desired packages for assigning to the station. These packages are:

- **Engineer Work Station (ES):** This is a station for the engineers to design project using **HPIDesign** and **PCE** applications.
- **Operator Work Station (OS):** This station is used for running the designed project using **HPIRun**.
- **Historical Data Warehouse (HDW):** This package is designed for providing features for backup, archive and deleting project run-time data.
- **Project Server (PS):** This is a server which provides services for maintaining the design source of the project and delivers a multi-engineering environment.
- **DXS Information Server (DIS):** This server collects data from the available devices and delivers them for monitoring in **HPIRun**.

Also there is a list of prerequisite applications for each package, displayed on the right side. If a machine does not have the listed applications installed, they should be installed on the machine first. Meanwhile the current installed applications are made distinct with a check-mark sign in front of their name.

☞ It is to be pointed out that it is possible to have common packages assigned to different machines, for example two stations can have OS or ES packages simultaneously. But only one station can be the *Project Server*, if this package is assigned to more than one station, the first station will be considered.

# 10 HPI Security

## 10.1 Introduction to HPI Security

HPI Security Manager provides users with the ability to define operations & roles based on their application needs. HPI Security supplies access control mechanism for all parts of the application.


There are some important concepts in **HPI Security Manager**:

- **Operation Category**: Operation category is a class of operations which are related together in some aspects. For example **Design** Category contains class of operations that are related to design application.
- **Operation**: Operation is a single action that can be done by the user. For example **Load** is an operation that can be done.
- **Operation Instance**: Operation instances are realization of the Operations. Many different instances can be made by means of Operation instances. For example "**LoadDesignMode**" Operation instance can be defined as an instance of Load Operation.
- **Role(s)**: Role specifies a category of users. For example "Designer" role is a category for users that have the rights to design in **HPIDesign**. Each user who is assigned to this role is allowed to perform all operations that a designer is permitted to do.

## 10.2 Security Manager

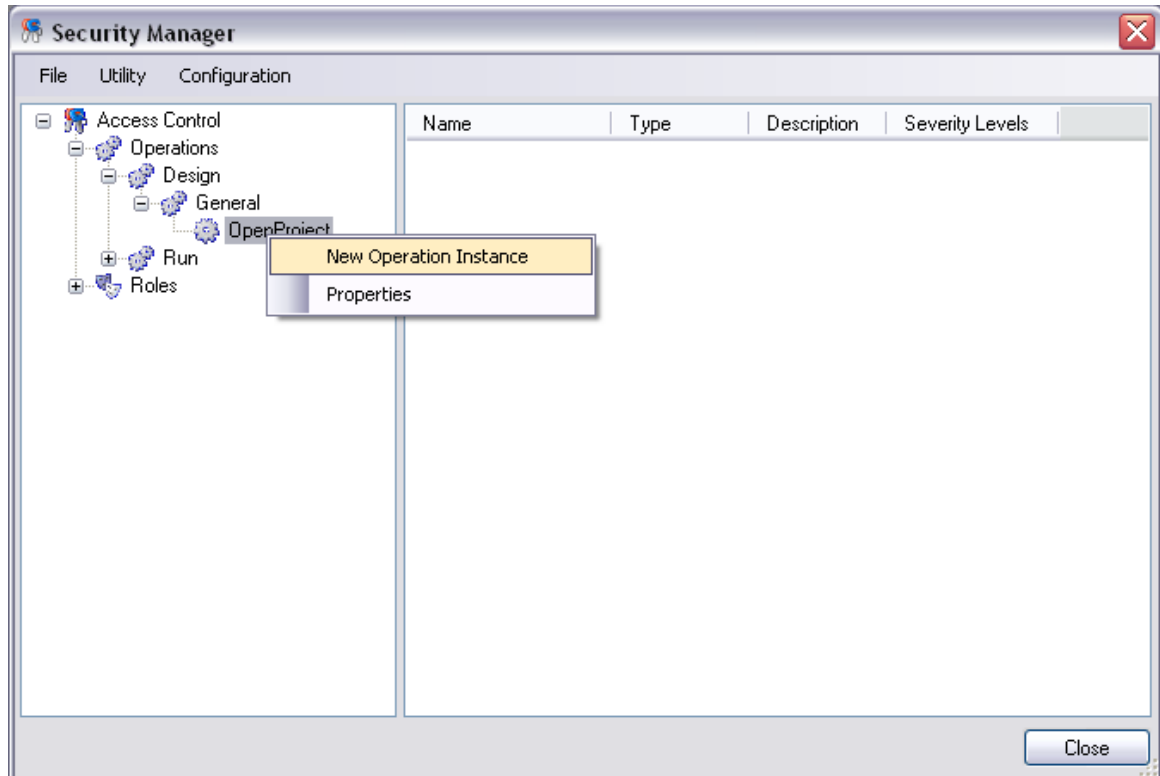
### 10.2.1 Opening Security Manager

You can open this tool in 2 ways:

- In **HPIDesign** menu, click **View** and select **Security Manager**.
- Click on Security manager  icon from toolbar

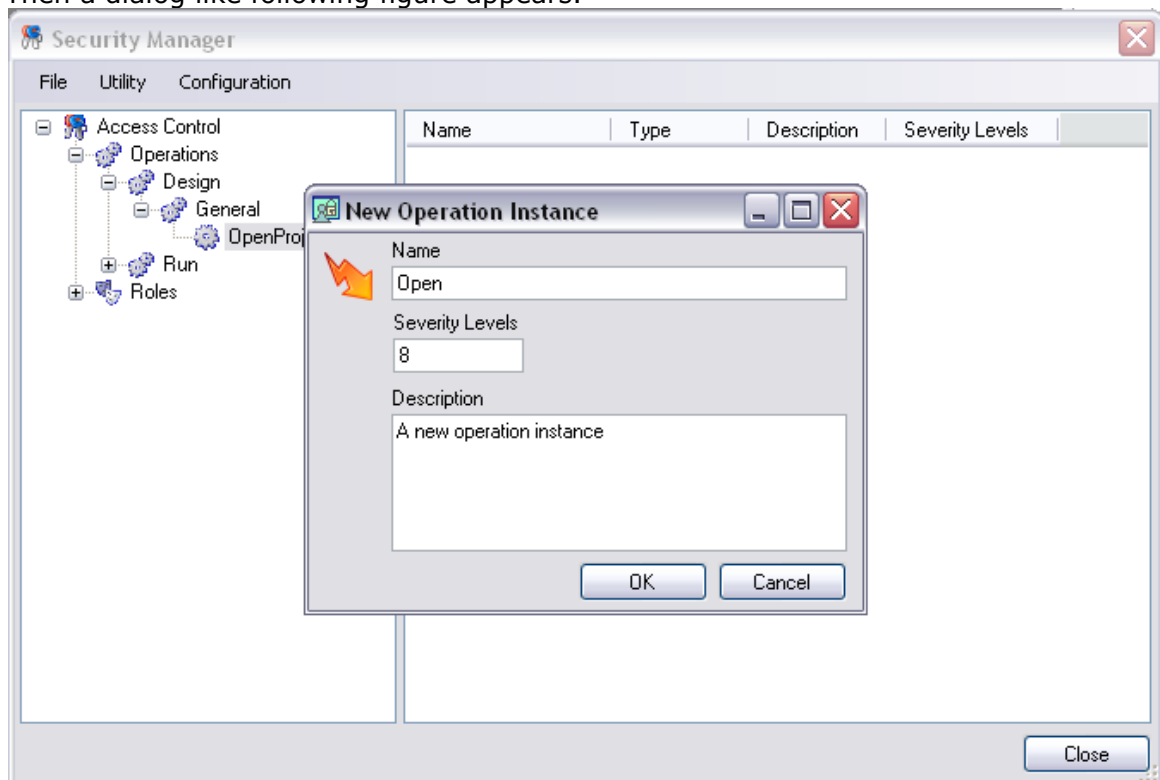
### 10.2.2 Define an Operation Instance

To define a new operation instance, use the **Security Manager** console. Right-click on the operation from which you want to create an instance, then select **New Operation Instance** from the popup menu.



**Figure 47) Add New Operation Instance**

Then a dialog like following figure appears.



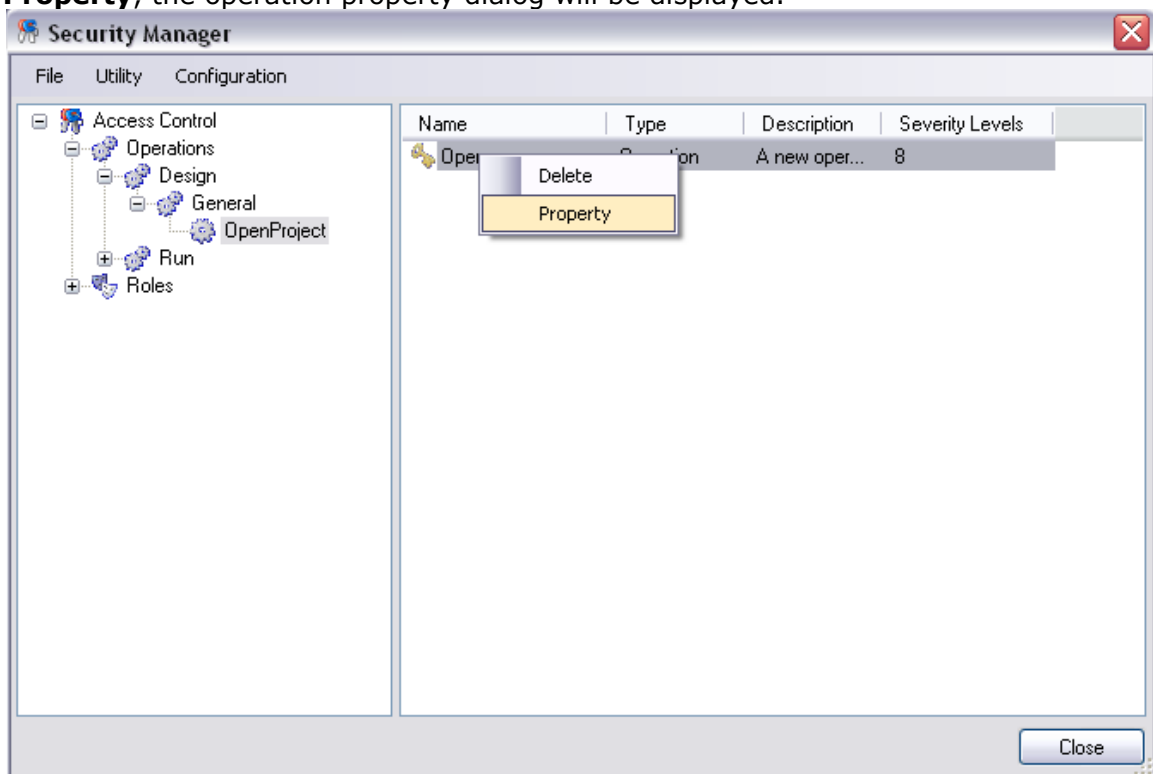
**Figure 48) New Operation Instance properties**

This dialog has three fields: **Name**, **Severity Level**, and **Description**.

- **Name:** Indicates the name of new operation instance.
- **Severity Levels:** Indicates the number of severity levels for this operation instance.
- **Description:** User can write any useful description about new operation in this field.

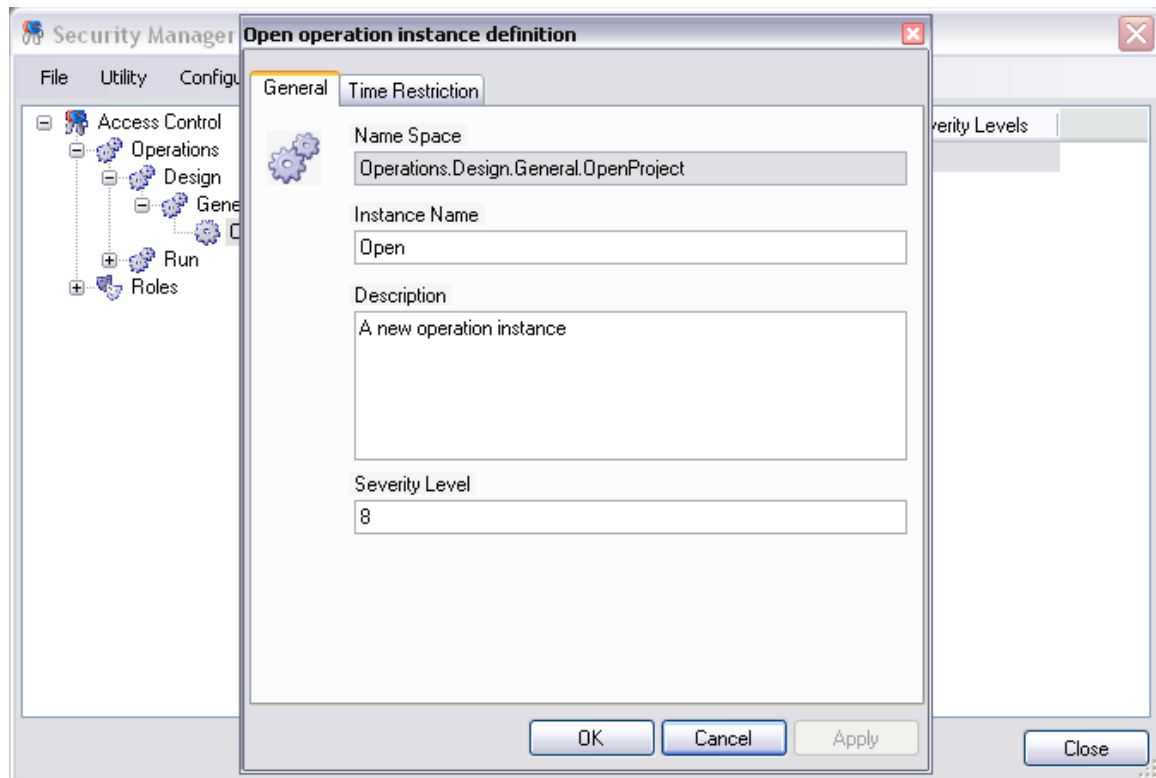
### 10.2.3 View or change operation instance properties

**Security Manager** provides a panel to set or update properties of operation instances. Double/Right-click on the desired operation instance and select **Property**, the operation property dialog will be displayed.



**Figure 49) Changing operation instance properties**

The dialog includes two tabs. **General** tab contains the hierarchy of this operation instance: Operation instance name, description and severity level.

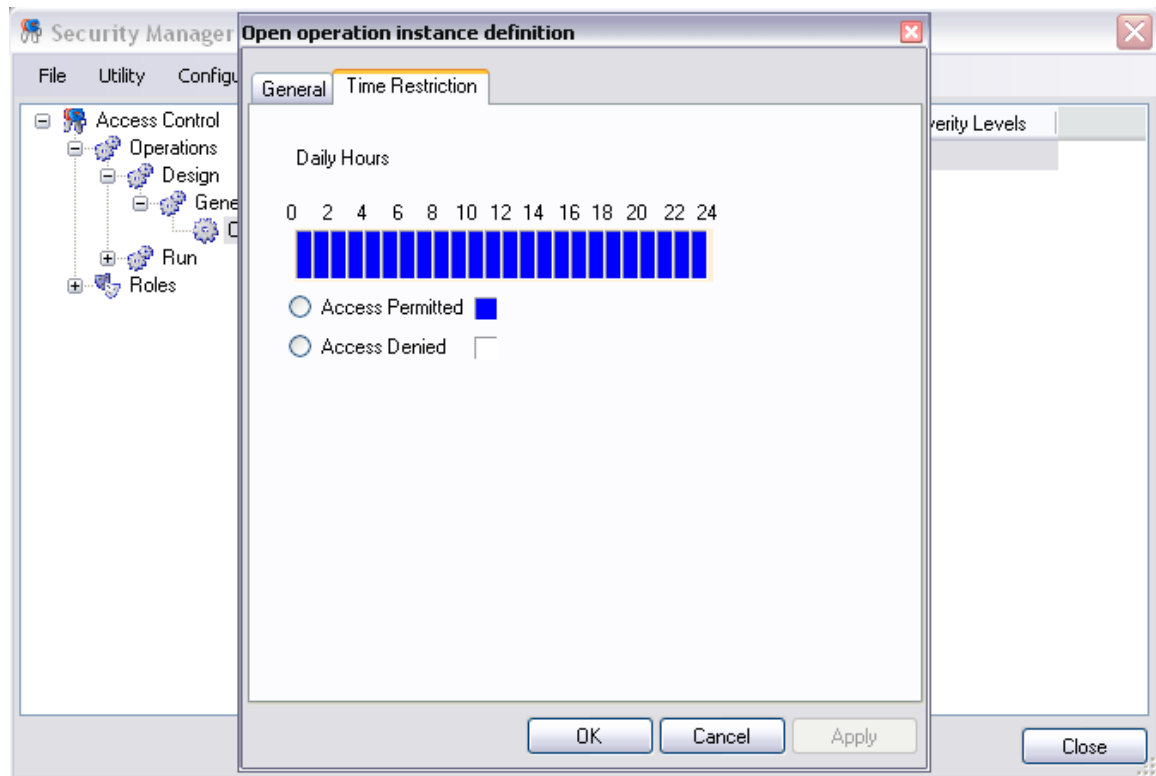


**Figure 50) Properties of an operation instance**

**Severity Level** is the maximum level of severity for this instance.

**Time Restriction** tab is used to set the time restriction for the operation instance. By default, all operation instances are accessible at all times, but they can be customized.

The blue slots indicate number of access permitted times and the white ones indicate the number of access denied times.

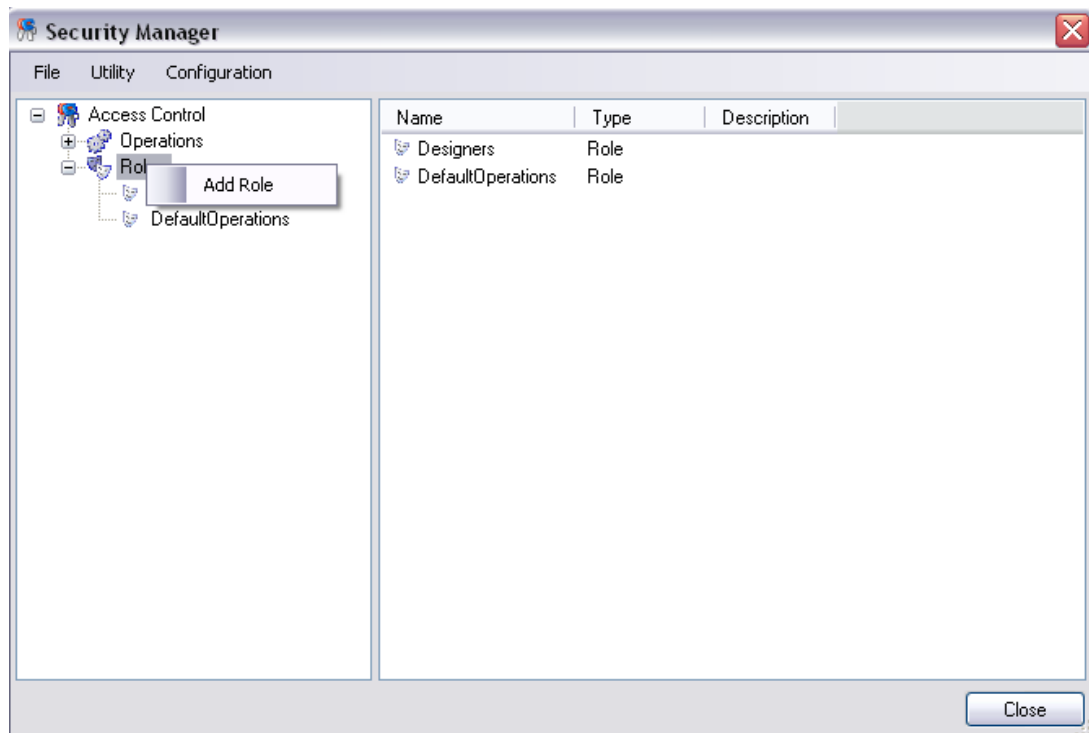


**Figure 51) Time restriction for operation instances**

The access time to operations can be changed simply by clicking on the related slot and selecting either **Access Permitted** or **Access Denied** option.

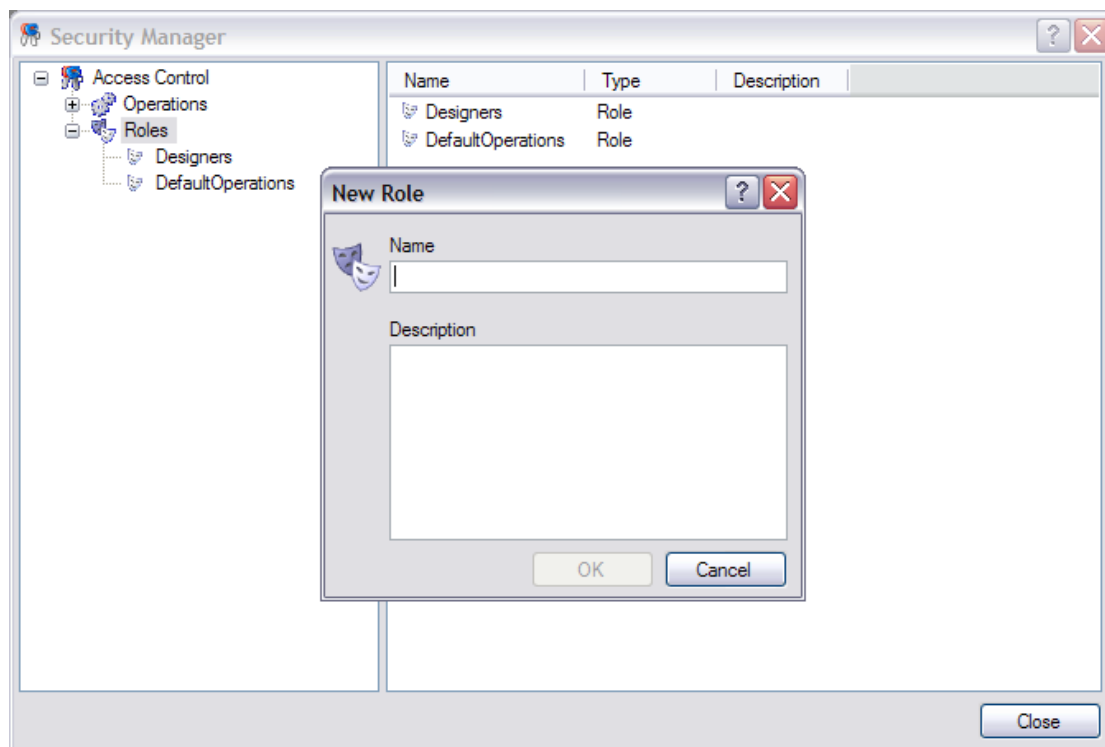
#### **10.2.4 Add New Role**

New role can be added by using **Security Manager**. Right-click on **Roles** and select **Add Role** from the popup menu.



**Figure 52) Add a role**

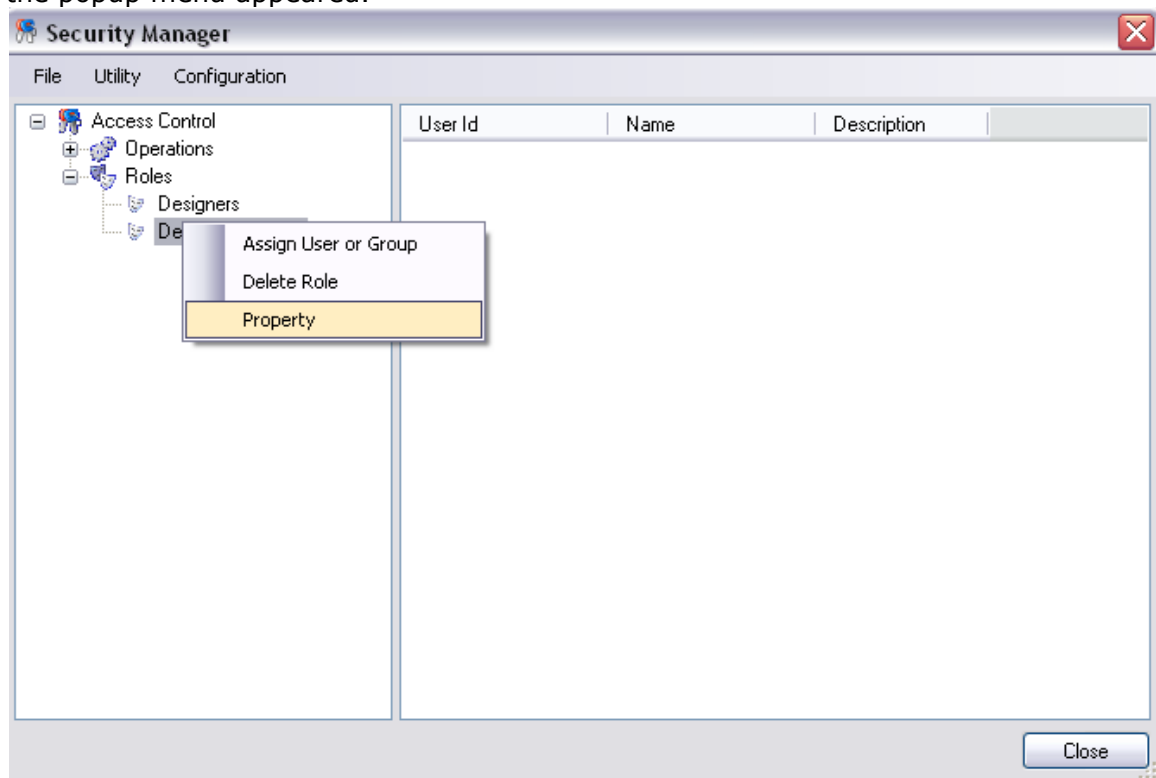
A dialog appears and receives new role name and description. It is sufficient to write a new name and click **OK button**.



**Figure 53) New Role dialog box**

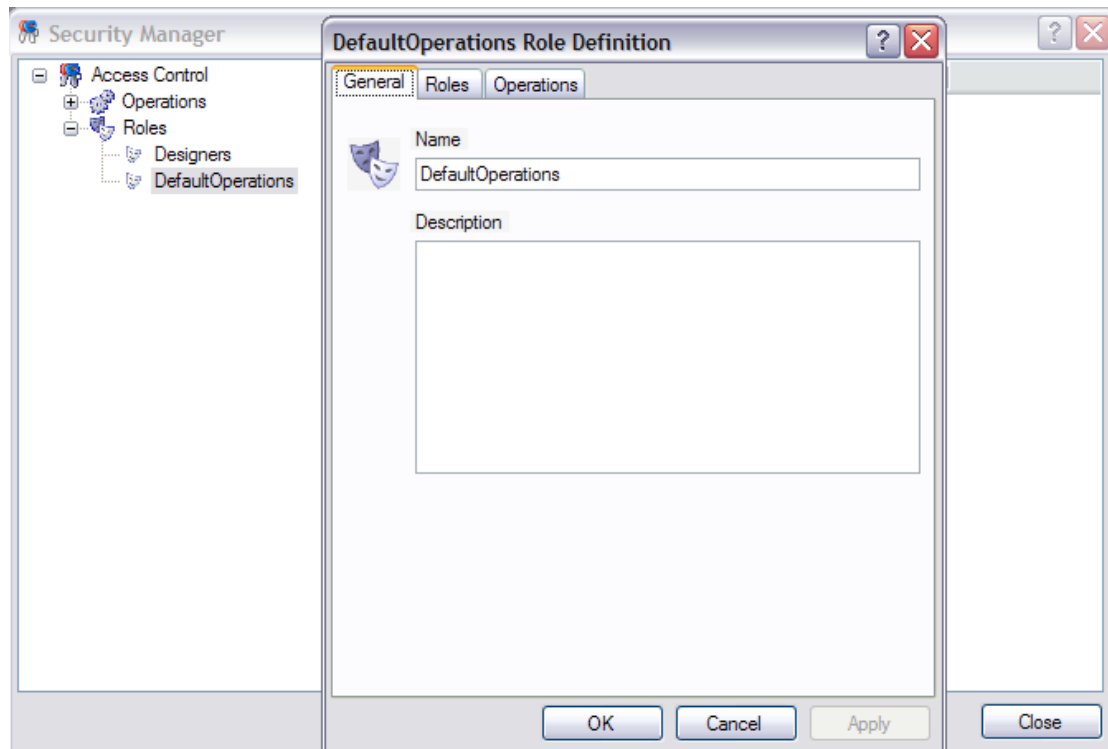
## 10.2.5 View or change role properties

To view or change role properties, right-click on the role and select **property** in the popup menu appeared.



**Figure 54) Changing properties of a role**

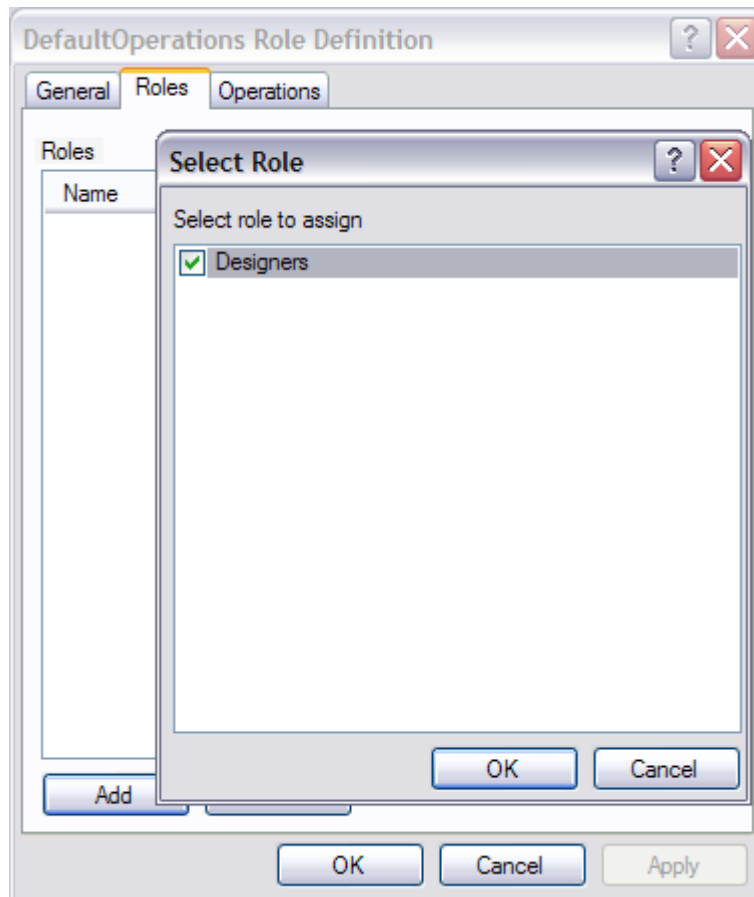
In **General** tab, the name and description of a role appear. It is possible to change its name and description and apply it by using **Apply** or **OK** button.



**Figure 55) General tab of a role properties dialog**

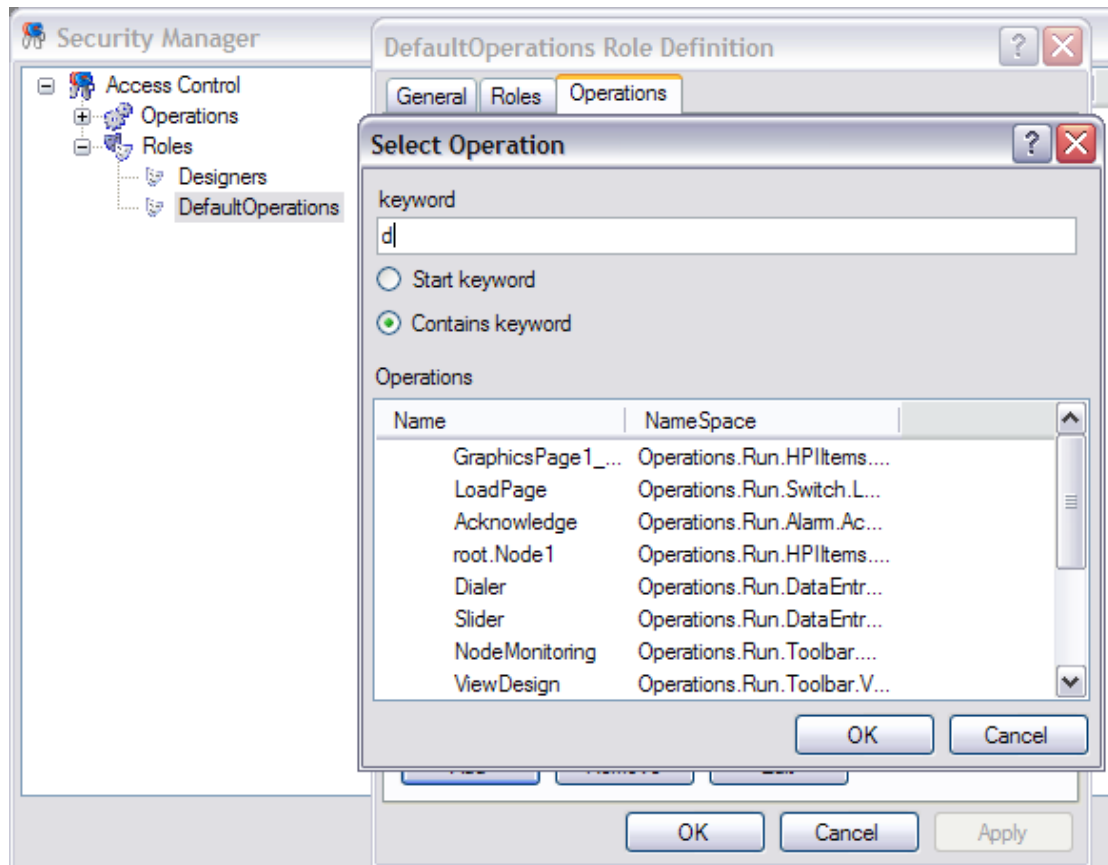
In **Roles** tab, it is possible to assign another role to the current role. It means that a role can contain other roles authorities. When a role is assigned to another role, the destination role is permitted to perform all operations that are assigned to the source role.

For example in the following figure, "Designers" role is selected to be assigned to "Default operations" role. After this, users with "Default operations" role are allowed to perform all operations that designers are permitted to do. In this way, simple roles can be constructed from more complicated roles. **Roles** tab provides the ability to add/remove a role to/from another role.



**Figure 56) Add a role to another role**

**Operation** tab can be used to manipulate the operation instances related to a role. A user who is assigned to a role is authorized to perform the operation instances that are assigned directly/indirectly to that role. **Operations** tab can be used to add/remove or edit operations assigned to a role. The following figure shows how to assign an operation instance to a role.



**Figure 57) Assigning operation instances to a role**

When an operation instance is assigned to a role, lowest severity of that operation will be assigned to that role.

Edit button in **Operations** tab is used to change the severity levels assigned to the role. This value indicates that users with this role can perform all actual operations with the same operation instance and severity equal or lower than this number.

You can find an example in [Security for dynamics](#) section.

### 10.2.6 Assign users to the roles

A user can be assigned to different roles and then he/she is authorized to perform operations with severity level equal or lower than the severity level assigned to those roles.


By selecting each role, assigned users to that role will be shown in the right panel. To assign a user to a role, right-click on the role and select **Assign User from the popup menu**.

Afterwards a window like the following figure will appear. This window provides the ability to select desired users or groups (among windows users/groups) and to assign them to a role.

You can search for users in the network or local users of the machine.

## 11 Run a project

To run a designed project, click on  button from toolbar or select **Project menu** → **Run**. This action executes the designed project in **HPIRun**.

 You can run the project number of times after designing to check your design. In this case you can use **Refresh** button on toolbar to reload a page in run mode.



171 East Liberty Street, Suite 276,  
Toronto, Ontario, M6K 3P6, Canada

© 2011 APAT Co. All Rights Reserved. All other brands or trademarks are property of their respective owners.

### ***Additional Resources:***

For more information about this DXS Supervisory Control Platform products please visit APAT website site at: [www.apatechnologies.com](http://www.apatechnologies.com)

