Vijeo Look

User manual

Merlin Gerin

Modicon

Square D

Telemecanique

Schneider Electric

Installation

The Installation Process

Vijeo Look is distributed on a CD-ROM containing several software components.

- OFS Configuration Tool.
- OPC Factory Server.
- Java Runtime Environment (JRE).
- Microsoft Database Engine (MSDE).
- VBA.
- Vijeo Look components.
- UnitelWay communication drivers.
- Laplink

The installation should take about 20-30 minutes to complete, depending on the performance of your PC. The installation requires up to 520 Mb of disk space depending on the options selected.

Information required

The installation process uses a standard installation wizard that will prompt you to enter information and make decisions about software placement and settings. Before you start the installation you should have the following information available.

- The languages in which the HMI is to run: any pair selected from English, French, German or Spanish.
- The registration details for the software license: User name and Company name.
- Destination folders. You can accept defaults or select your own.
- The installation type: *Typical, Compact* or *Custom* (see below).

Getting ready for the installation

The installation process is automatic, but you will need to prepare your PC before you start it.

- If you have a previous version of Vijeo Look installed, its components should first be uninstalled.
- Check that your PC meets the minimum specification required for installing and running the software.

The Installation Process

The installation process is started automatically when you insert the CD-ROM into the CD drive on your PC. If your CD does not have auto-start enabled then use the Windows Explorer to select and run Setup.exe.

Before the main installation starts you will be asked to select the language used for text the installation process. Once you have selected the installation language a Welcome screen appears to guide you through the process.

The steps of the installation sequence are marked by dialog boxes. The first few just gather information from you. Each dialog box that the installation wizard displays includes three buttons. Use these to enter settings, to review the available options, and to make changes to your entries up to the point at which you initiate the copying of files. (See Installing Vijeo Look and Related Components, below.)

- *Back* – (from the second box onwards) revisit the previous dialog box.
- *Next* – (up to the last-but-one box) proceed to the next dialog box.
- *Cancel* – exit from the installation wizard. (See the note about incomplete installation below.)

The following process illustrates a typical installation. Each step and its corresponding dialog box is indicated by a coloured heading.

The License agreement

You are asked to accept the terms of the License Agreement. If you do not accept the terms the installation process will be aborted.

Customer Information

Used to enter your User name and Company name.

Language selection

Used to choose the two languages in which Vijeo Look, its menus and dialog boxes will be available, and to select one as the default language.

Some dialog boxes are provided by the host operating system and will appear in the language in which it was installed.

Set-up type and location

Used to select the destination folder and the type of installation from the following options.

- *Typical* - Installs the program files, demonstration projects, libraries, documentation and help files.
- *Compact* - Installs the program and help files only.
- *Custom* - Allows the user to select some of the optional components from the following.
  MSDE - Microsoft Database Engine.
  Program files - The programs and configuration files that together form product.
  Demonstration project - An example project.
  Help files - Help files in compiled hypertext (.chm) format.
  Documentation files - Printable documentation in portable document .pdf format.
  Shared library files - Libraries of pre-defined symbols and bitmaps.
  Laplink - An application program that may be used to transfer the program and project files between computers.

If you choose a Custom Installation, you are then asked to choose the components to be installed. The components are listed in the left pane. If you select the name of a component, a short description of it is displayed in the right hand pane. Tick the required components in the left hand pane, or click again to un-tick them.

Program Folder

Used to select the folder in which the program shortcuts will be created.

Review and start copying files

A scrolling dialog box is displayed to allow you to review the settings and details you have entered, before the wizard starts copying the installation files. Select the Back button to make any changes, or the Next button when you are ready to proceed.

File installation

The installation will now proceed with installing the main components.

- OFS Configuration Tool - Displays 'Installing OFS Configuration Tool...' (This takes several minutes.)
- OPC Factory Server - Displays 'Installing OPC Factory Server...'.
- Java Runtime Environment (JRE) – Displays 'Installing Java Runtime Environment...'.
- Microsoft Database Engine (MSDE) – Displays 'Installing MSDE...', then a time-bar in a box called Setup Status to represent the copying process.

- VBA - Displays 'Installing VBA'.
- Program Files - Displays 'Installing Program Files...', including Configuration Explorer, with time-bar for Setup Status.

When the main installation is complete, the wizard installs one further component.

- UnitelWay Communication Driver - Displays a dialog box called Choose Destination Location, then 'Installing UnitelWay Communication Driver...'.

Finishing the Installation

When all of the components have been installed, the Installation wizard will invite you to restart your PC. When you do so, the Control Panel is opened in Windows. You may close it and check for the icons by which you may start Vijeo Look (see below).

The installation process is complex, installing components from a number of sources. If, for any reason, the installation is not successful (ceases part-way through, or if you choose to continue with the installation when a problem occurs) you must un-install the software, re-start the PC and start the installation process again.

You can select the destination folder for most installation options except MSDE which must be installed in the system folders. MSDE requires approximately 50 Mb of disk space.

Starting Vijeo Look

What Happens When I Start Vijeo Look? | The Microsoft SQL Server

Vijeo Look can be started in three ways.

- from the icons created in the Start menu or on the Windows Desktop.
- by double clicking on a project (.vlp) file.
- from a command line.

What Happens When I Start Vijeo Look?

Vijeo Look is not one, but a collection of interacting programs. When you start Vijeo Look all the necessary programs are started automatically. What the User is most aware of is the HMI, which normally occupies the entire screen. In addition there are a number of other programs, known as servers, that provide the HMI with its data. These appear as icons in the status area of the Windows task bar.

Icon	Program	What does it do
DS	Real-Time Data Server (RTDS)	Provides an interface between the OPC Factory Server, the HMI and the Historical Data Server.
H▶	Historical Data Server (HDS)	Records real-time data in a database and makes it available to the HMI.
⁂	OPC Factory Server (OFS)	Manages the communication to PLC's and provides real-time data to the Real Time Data Server.

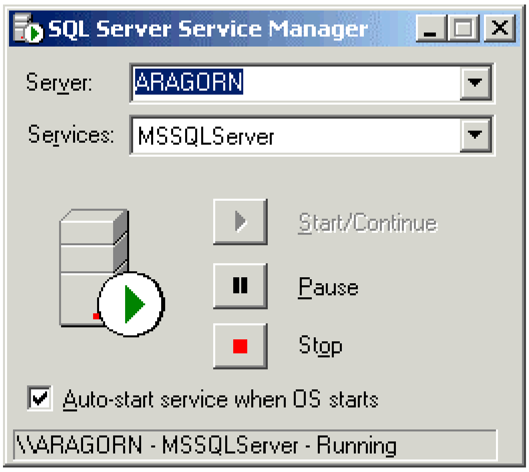
If you right click on any of the server icons on the task bar a pop-up menu will be displayed with two options.

- *About* - Provides basic information about the server such as its revision number.
- *Exit* - Allows you to stop the server manually.

! **Do not stop a server while Vijeo Look is running.** The exit command should only be used to stop a server in the event that Vijeo Look has exited abnormally.

The Microsoft SQL Server

The Microsoft SQL Server is provided with Vijeo Look and is one of the options you can choose when selecting a database. By default, the SQL server starts automatically when you start Windows and appears as an icon in the status area of the task bar. The SQL server may be controlled from a pop-up menu displayed by right clicking on its icon.


SQL Server Service Manager

Server: ARAGORN

Services: MSSQLServer

☑ Auto-start service when OS starts

\\ARAGORN - MSSQLServer - Running

- *Start/Continue*, *Pause* and *Stop* - Select the buttons to control the server process.

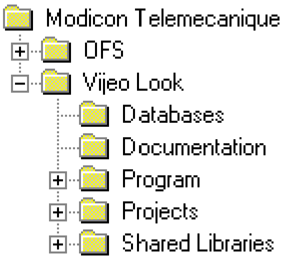
- *Auto-start service when OS starts* - If you are not using the SQL server then you can un-tick this box. The SQL server will not run next time Windows is started.

! **If you are using the SQL server do not stop it whilst Vijeo Look is running.**

The Vijeo Look Folder Structure

Project Folder Structure | Library Folder Structure

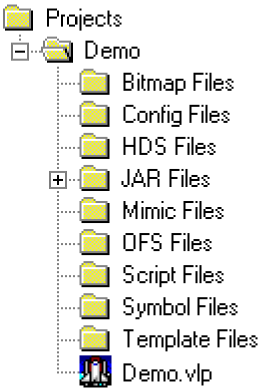
All Vijeo Look installations follow the same basic directory structure as shown below.



```
Modicon Telemecanique
  OFS
  Vijeo Look
    Databases
    Documentation
    Program
    Projects
    Shared Libraries
```

- *Modicon Telemecanique* – The default root folder for the software. An alternative folder may be selected during installation.
- *OFS* - The folder containing the OFS software. See the documentation supplied with the OFS for more information.
- *Databases* - Database files for storing historical data generated by Vijeo Look.
- *Documentation* - Complete Vijeo Look documentation in .pdf format.
- *Program* – Program executable files, libraries and other resources.
- *Projects* – The root folder for all the projects.
- *Shared Libraries* – Libraries available to all projects containing reusable objects, scripts bitmaps and mimics.

The Project Folder Structure



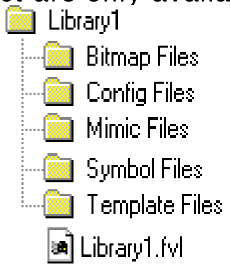
```
Projects
  Demo
    Bitmap Files
    Config Files
    HDS Files
    JAR Files
    Mimic Files
    OFS Files
    Script Files
    Symbol Files
    Template Files
    Demo.vlp
```

- *Demo* – The root folder for the project Demo.
- *Bitmap Files* – The folder containing any local image files used by the project. Images in this folder are not available to other projects.
- *Config Files* – Project configuration files.
- *HDS Files* – Temporary files generated by the Historical Data Server.
- *Jars* – Java Run-time files.
- *Libraries* – Libraries available to this projects only containing reusable objects, scripts, bitmaps and mimics. (This folder will only exist if you have created additional Local Libraries as part of the project configuration.)
- *Mimic Files* – Mimic definition files.
- *OFS Files* – OPC Factory Server related files.
- *Script Files* – Script files containing Functions, Subroutines and Variables not linked to a particular object.
- *Symbol Files* – Reusable graphic objects.
- *Template files* – Mimic template files.

- *Demo.vlp* – The project definition file. Double clicking on a .vlp file will automatically start Vijeo Look. (Similar to double clicking on a .doc file to start Word)

The Library Folder Structure

The shared libraries and project libraries follow the same structure. Resources available in the shared libraries in the root folder are available to all projects. Resources in libraries in a project are only available to that project.



Library1
Bitmap Files
Config Files
Mimic Files
Symbol Files
Template Files
Library1.fvl

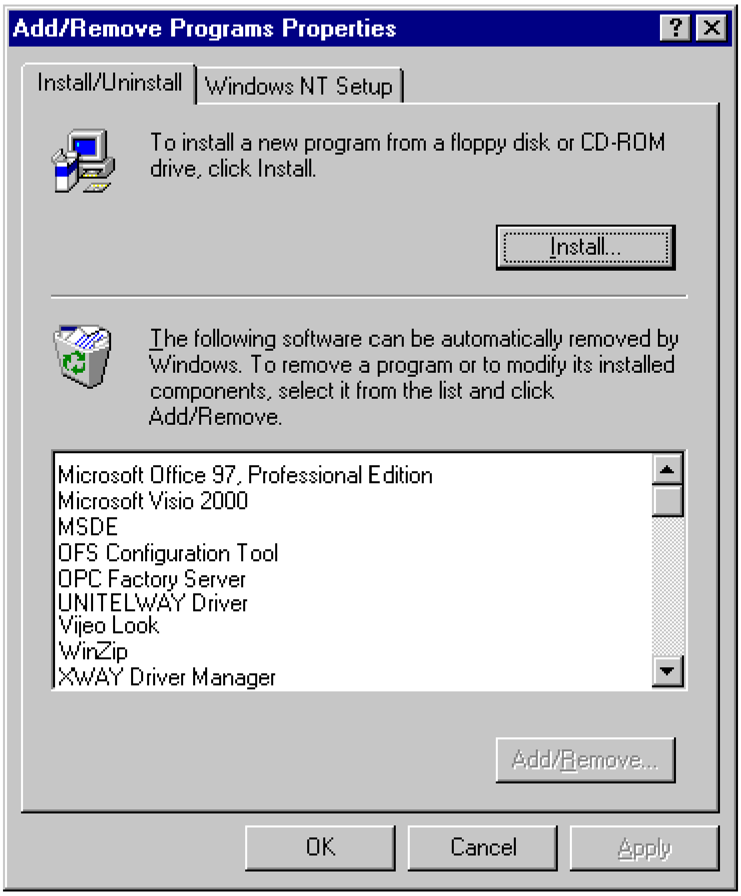
- *Library 1* – The root folder for the library Lib 1.
- *Bitmap files* – Image files.
- *Config files* – Configuration files.
- *Mimic files* – Mimic definition files.
- *Symbol files* – Reusable graphic objects.
- *Template files* – Mimic template files.
- *Library 1.fvl* – The library definition file.

Removing Vijeo Look

Before Removing Vijeo Look | How to Remove a Program in General | Cleaning Up After Removing Vijeo Look

! Before removing a licensed version of Vijeo Look you must first remove or transfer the License or it will be lost.

To remove Vijeo Look you must use the Add/Remove Programs command in the Windows' Control Panel. You cannot delete the Vijeo Look files manually from the hard disk.



Add/Remove Programs Properties

To install a new program from a floppy disk or CD-ROM drive, click Install.

The following software can be automatically removed by Windows. To remove a program or to modify its installed components, select it from the list and click Add/Remove.

Microsoft Office 97, Professional Edition
Microsoft Visio 2000
MSDE
OFS Configuration Tool
OPC Factory Server
UNITELWAY Driver
Vijeo Look
WinZip
XWAY Driver Manager

Vijeo Look contains several components each of which have a corresponding entry in the Add/Remove Programs dialog. Typically you will find:

- MSDE
- OFS Configuration Tool
- OPC factory Server
- Vijeo Look
- UNITELWAY Driver
- XWAY Driver

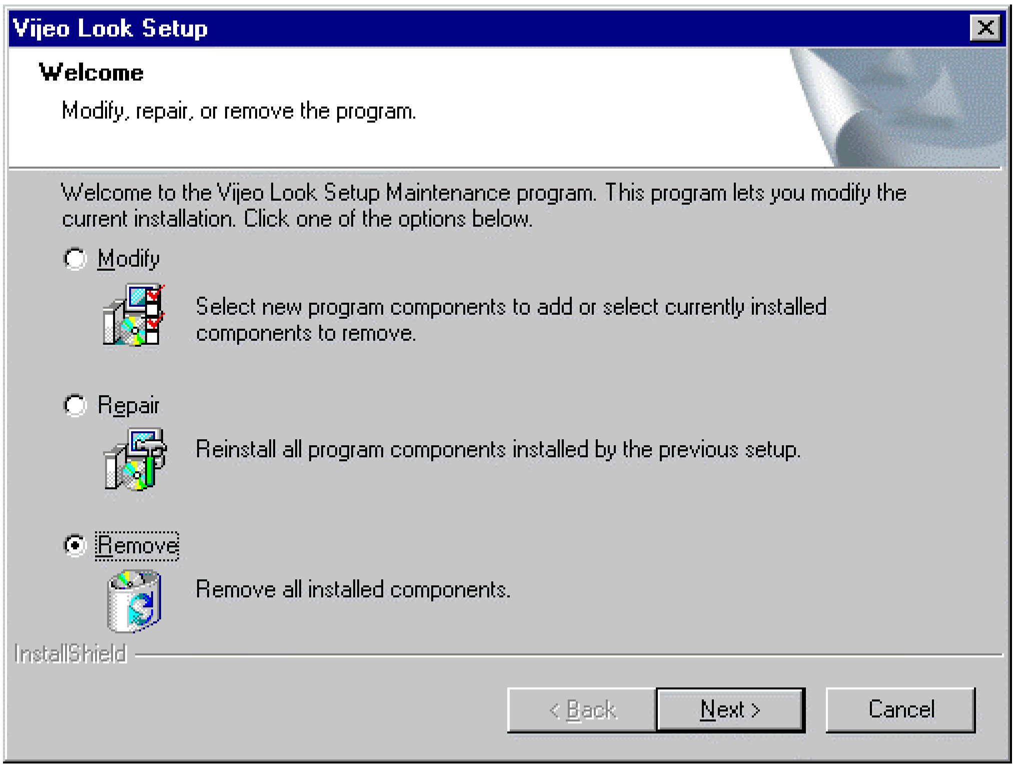
To remove Vijeo Look completely, you must remove each of these components in turn.

Before Removing Vijeo Look

1. Ensure all components of Vijeo Look have been stopped. The best way of doing this is to stop Vijeo Look and then shut down and reboot the PC. Make sure that Windows is not configured to restart Vijeo Look automatically when the PC is restarted.
2. Archive any projects that have been created. Removing Vijeo Look does not remove project files but this is a sensible precaution to take.
3. Ensure the Vijeo Look License (if any) has been removed or transferred to another installation.

How to Remove a Program in General

4. Open the Windows Control Panel.
5. Select the Add/Remove Programs command.
6. Select the program you want to remove from the list of installed software and click the *Add/Remove* button.
7. Depending on the program you have selected, either the program will be immediately removed, or a further dialog box will be displayed with a number of options.



Vijeo Look Setup

Welcome

Modify, repair, or remove the program.

Welcome to the Vijeo Look Setup Maintenance program. This program lets you modify the current installation. Click one of the options below.

○ Modify

Select new program components to add or select currently installed components to remove.

○ Repair

Reinstall all program components installed by the previous setup.

● Remove

Remove all installed components.

8. If you are presented with a further dialog box such as the one above, select the *Remove* option and select *Next*. The program will then be removed. Depending on what you are removing you may, or may not, get a progress box displayed.
9. Make a note of any messages you get whilst removing the program. You may need to delete some components manually when the automatic process has finished.
10. When you have finished removing programs close the Add/Remove Programs dialog and close the Control Panel.

Cleaning Up After Removing Vijeo Look

Removing Vijeo Look using the Add/Remove Programs dialog will remove most of the Vijeo Look files and, most importantly, its registration with the operating system. To remove all traces you will have to use the Windows File Manager to delete any folders or files that the automatic part of the process was unable to remove.

License Manager Overview

The License Manager protects Vijeo Look against unLicensed use by locking the installation to a specific hard disk on the target machine. It also determines the functionality available to the user.

- The installation type (Standard or Demonstration).
- The operational modes (BuildTime and RunTime or RunTime only).
- The maximum I/O (Input / Output) count.
- The data servers allowed (Any OPC server or the OFS server only).

A License document is provided with all versions of Vijeo Look except the demonstration version. The License document specifies what options have been purchased and includes unique serial and reference numbers. The following License types are available.

- BuildTime and RunTime - 128 I/O - OFS server only
- BuildTime and RunTime - 512 I/O - OFS server only
- BuildTime and RunTime - 1024 I/O - all OPC servers
- RunTime only - 128 I/O - OFS server only
- RunTime only - 512 I/O - OFS server only
- RunTime only- 1024 I/O - all OPC servers

The License manager may be started from a shortcut in the Vijeo Look program group or, for an unLicensed installation, it is started automatically whenever Vijeo Look is started. For a BuildTime – RunTime version, the License Manager is also available from the Help About menu. (Select **?.About.License**.)

The first time the License Manager is started the user will be prompted to enter the serial and reference numbers. A check is then made to ensure these are valid and a temporary installation code is generated allowing Vijeo Look to run for 21 days. During this period the user must request a permanent authorisation code which will allow Vijeo Look to run indefinitely.

If you install the demonstration version no License is required. Vijeo Look will run for 30 days after it is first started. You can only install the demonstration version once on any one PC.

Please note that you must have the NT administration rights when you use the License manager.

The License Procedure

How to Activate Vijeo Look | How to Create an Authorisation Code Request File | How to Enter the Authorisation Code

The following explanation refers to the procedure to activate and License Vijeo Look installation. All licensing takes place within the License Manager. The License Manager is started automatically each time you attempt to start an unLicensed installation of Vijeo Look or it may be started from the shortcut in the Vijeo Look program group.

Before you start make sure that you have the License document supplied with the release kit available.

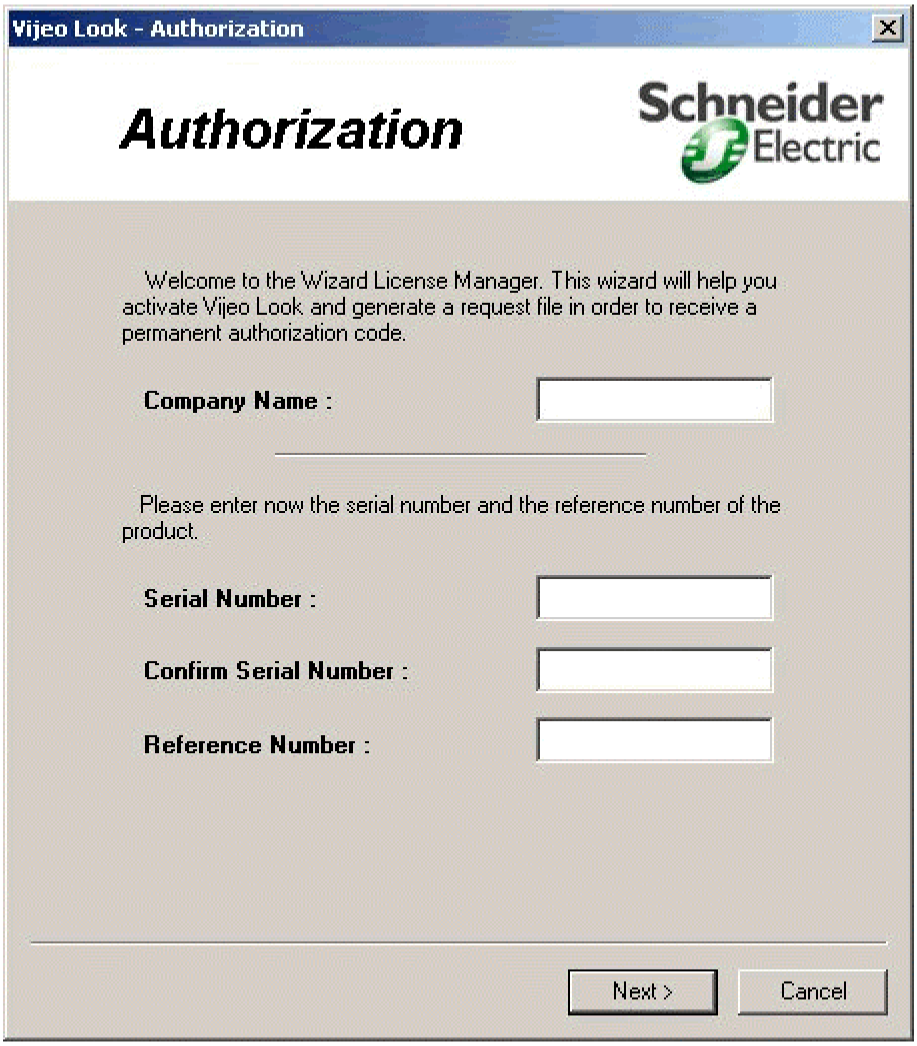
How to Activate Vijeo Look

11. Start Vijeo Look and enter the serial and reference numbers found on the License document.

12. Click the *Next* button. The License Manager will check that the numbers are valid - this may take a few seconds. If the check is successful you will be asked if you want to generate a License request file or generate the file later.

13. If you choose to generate the file later Vijeo Look will start using the temporary installation code.

The temporary installation code will allow Vijeo Look to work for 21 days after which it will no longer run. During this period each time you start Vijeo Look, the License Manager will display a warning that the product is unLicensed and prompt you to generate an authorisation code request file.



Vijeo Look - Authorization

Authorization

Welcome to the Wizard License Manager. This wizard will help you activate Vijeo Look and generate a request file in order to receive a permanent authorization code.

Company Name :

Please enter now the serial number and the reference number of the product.

Serial Number :

Confirm Serial Number :

Reference Number :

How to Create an Authorisation Code Request File

Before entering this stage you must have already entered the serial and reference numbers in the License Manager.

14. Start the License Manager and click the *Request File* button.

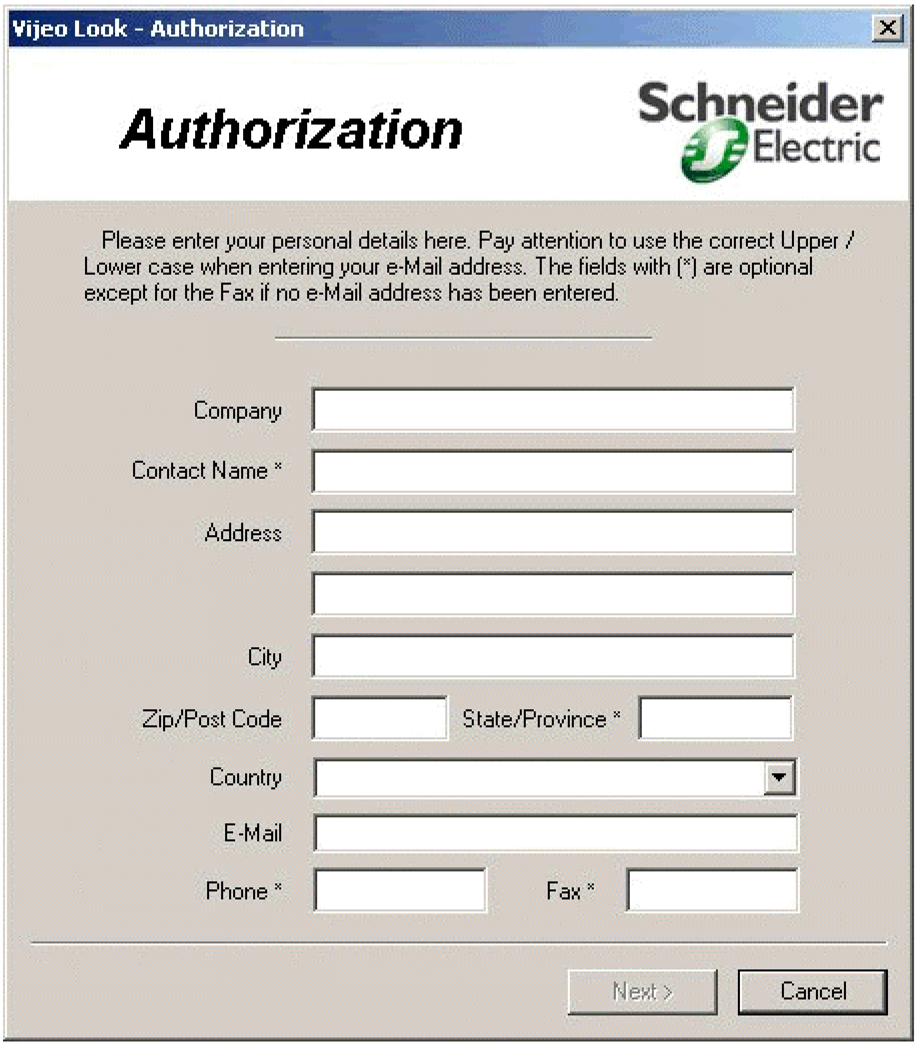
15. You will be asked to enter the following information.

- Company name
- Contact name (optional)
- Address
- City
- Zip or post code
- State or province
- Country
- Either your Fax number or e-mail address
- Phone number (optional)

16. When you have entered all the information click the *Next* button. If you have entered everything correctly an Authorisation Code Request file is generated. The name and location of the file will be displayed.

17. Close the License Manager.

18. Fax or e-mail the request file to the License authority to obtain the permanent authorisation code. (The fax number and e-mail address may be found on the License document.)



Vijeo Look - Authorization

Authorization

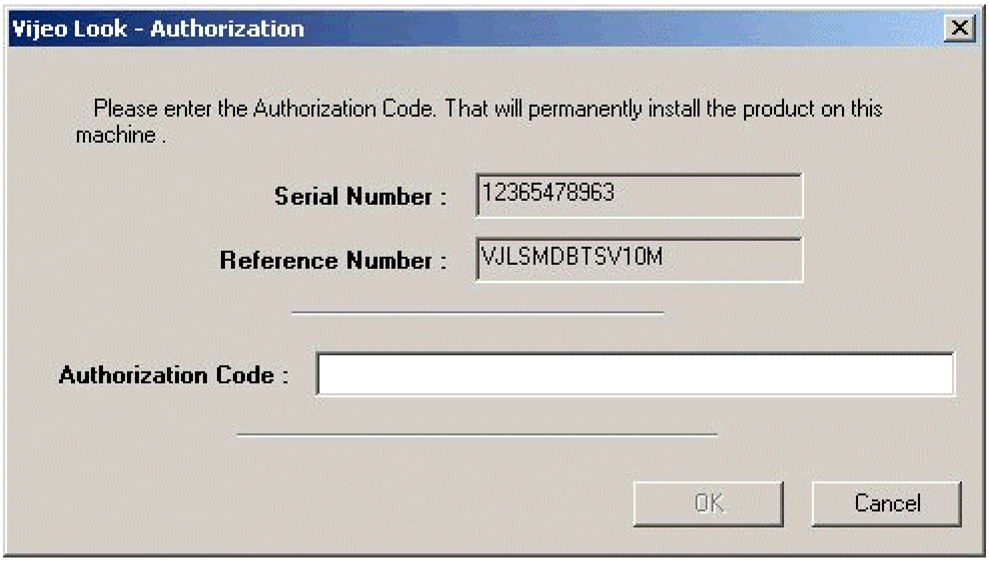
Please enter your personal details here. Pay attention to use the correct Upper / Lower case when entering your e-Mail address. The fields with (*) are optional except for the Fax if no e-Mail address has been entered.

How to Enter the Authorisation Code

Once you have obtained the authorisation code you can enter it using the License manager and the Vijeo Look options will be upgraded.

19. Start the License Manager and click the *Authorisation Code* button.
20. Enter the authorisation code and click the *OK* button. The License Manager will check that the information is correct and, if so, start Vijeo Look. Subsequently, when starting Vijeo Look, the License Manager is no longer displayed.

! The authorisation code will only work for the specific installation of Vijeo Look (Serial Number, Reference Number and hard disk) for which it has been requested.



Vijeo Look - Authorization

Please enter the Authorization Code. That will permanently install the product on this machine .

Serial Number : 12365478963

Reference Number : VJLSMDBTSV10M

Authorization Code :

Upgrading a License

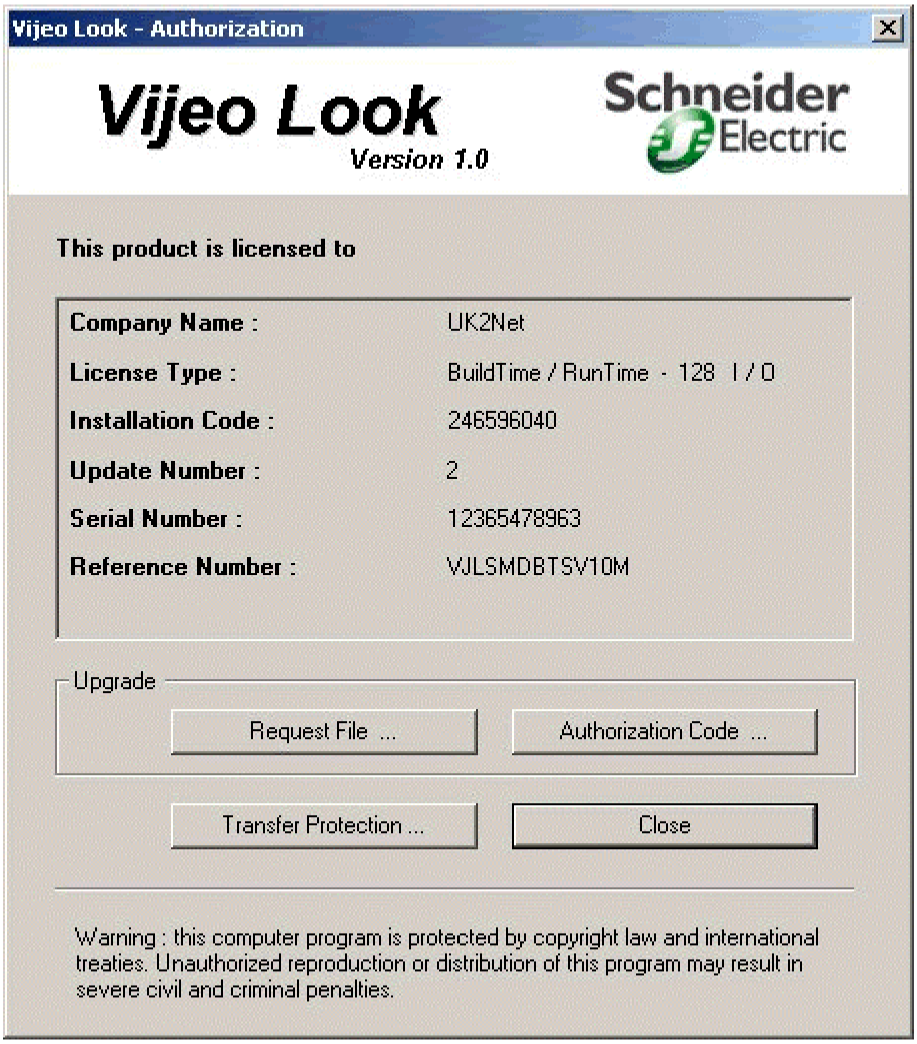
How to Create the Upgrade Request File | How to Enter the New Authorisation Code

In order to upgrade Vijeo Look options, (from 128 to 512 I/O for example) you must first purchase an upgrade License from your supplier. The upgrade will contain a new License document with new serial and reference numbers. Once you have the upgrade License you must generate a License request to obtain a new authorisation code in a similar way to the normal License procedure.

! You cannot upgrade from a RunTime to a BuildTime / RunTime version.

How to Create the Upgrade Request File

21. Shut down Vijeo Look.
22. Start the License Manager from the shortcut in the Vijeo Look program group and click the *Request File* button.
23. Enter the new serial and reference numbers from the License document and click the *Next* button. The License Manager will check that the information is correct and, if so, display the Request File box.
24. Enter the information requested and click the *Next* button. If you have entered everything correctly a License Request file is generated. The name and location of the file will be displayed which must then be faxed or e-mailed to the License authority to obtain the new authorisation code.
25. The License Manager may then be closed and Vijeo Look restarted. At this time Vijeo Look will still have the options specified in the original License.



This product is licensed to

Company Name :	UK2Net
License Type :	BuildTime / RunTime - 128 I / O
Installation Code :	246596040
Update Number :	2
Serial Number :	12365478963
Reference Number :	VJLSMDBTSV10M

Warning : this computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program may result in severe civil and criminal penalties.

How to Enter the New Authorisation Code

Once you have obtained the new authorisation code you can enter it using the License Manager and the Vijeo Look options will be upgraded.

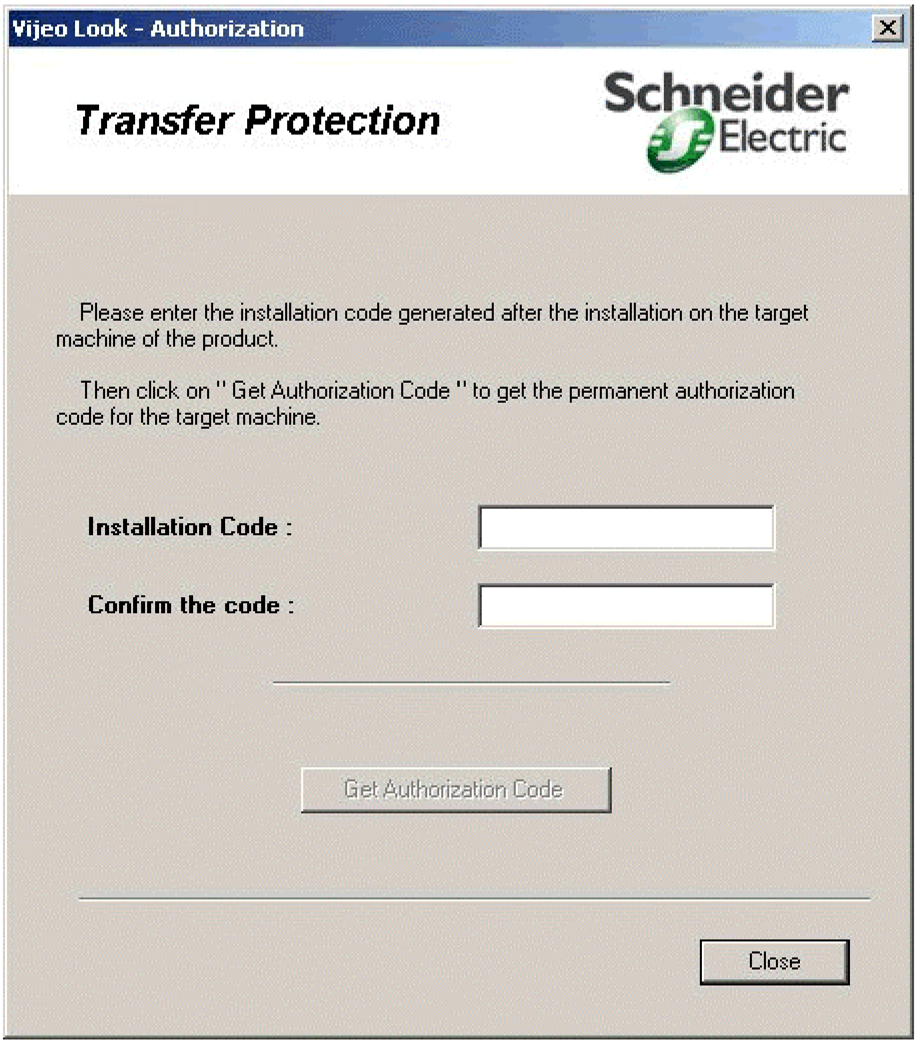
26. Shut down Vijeo Look.
27. Start the License Manager from the shortcut in the Vijeo Look program group and click the *Authorisation Code* button.
28. Enter the new authorisation code and click the *OK* button. The License Manager will check that the information is correct and, if so, upgrade Vijeo Look with the new options. The new License type will be displayed in the License Manager.
29. The License Manager may then be closed and Vijeo Look restarted.

Transferring a License

Transferring a License means removing the License from a fully working version of Vijeo Look on one PC and transferring it to another.

In the following explanation the PC from which Vijeo Look License will be removed is referred to as PC A and the PC to which the License will be transferred is PC B.

30. Install and activate Vijeo Look on PC B.
31. Make a note of the temporary installation code.
32. Start the License Manager on PC A and click the *Transfer Protection* button.
33. Enter and the installation code generated on PC B.
34. Click the *Get Authorisation Code* button. You will be warned that if you continue then the License on PC A will no longer be valid.
35. A new authorisation code for PC B will be generated and saved in the file NEWCODE.TXT found in Vijeo Look's Program folder.
36. Start the License Manager on PC B and click the *Authorisation Code* button. Enter the new authorisation code.
37. If Vijeo Look has also been upgraded at any time you will also be prompted to enter the serial and reference numbers of the upgrade. This information can be found in the file TRANSFER.TXT in the Program folder on PC A.

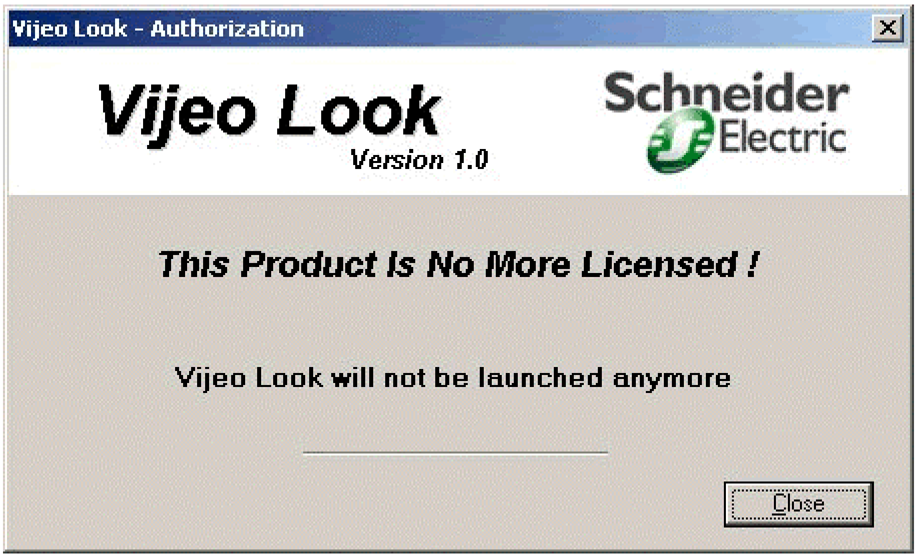


Please enter the installation code generated after the installation on the target machine of the product.

Then click on " Get Authorization Code " to get the permanent authorization code for the target machine.

What Happens if the License Expires

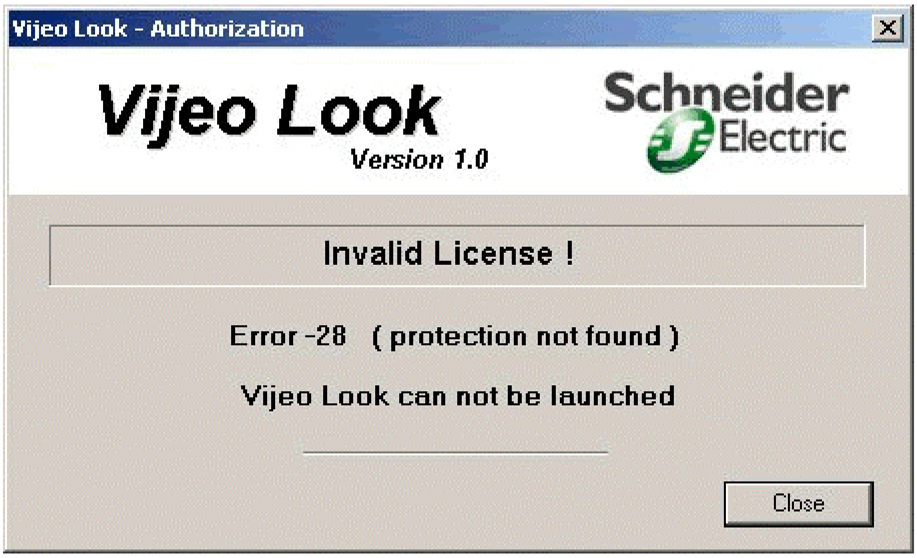
If you have not entered the authorisation code before the end of the 21 day period allowed by the temporary installation code the following dialog box will appear warning the User that Vijeo Look can no longer be started on the PC.



This Product Is No More Licensed !

Vijeo Look will not be launched anymore

Vijeo look will continue to run but once it has been shut down it will not be possible to restart it. Any attempt to restart it will display the following warning.



Invalid License !

Error -28   (protection not found)

Vijeo Look can not be launched

After this time you are still able to enter an authorisation code, but you must open the License Manager directly using the shortcut provided in the Vijeo Look program group.

The Demonstration License

Vijeo Look will run for up to 30 days in demonstration mode. The demonstration version has both RunTime and BuildTime capability but is limited to 25 I/O points and the OFS server only.

! You can only install the demonstration version once in a given PC.

Configuration Explorer Overview

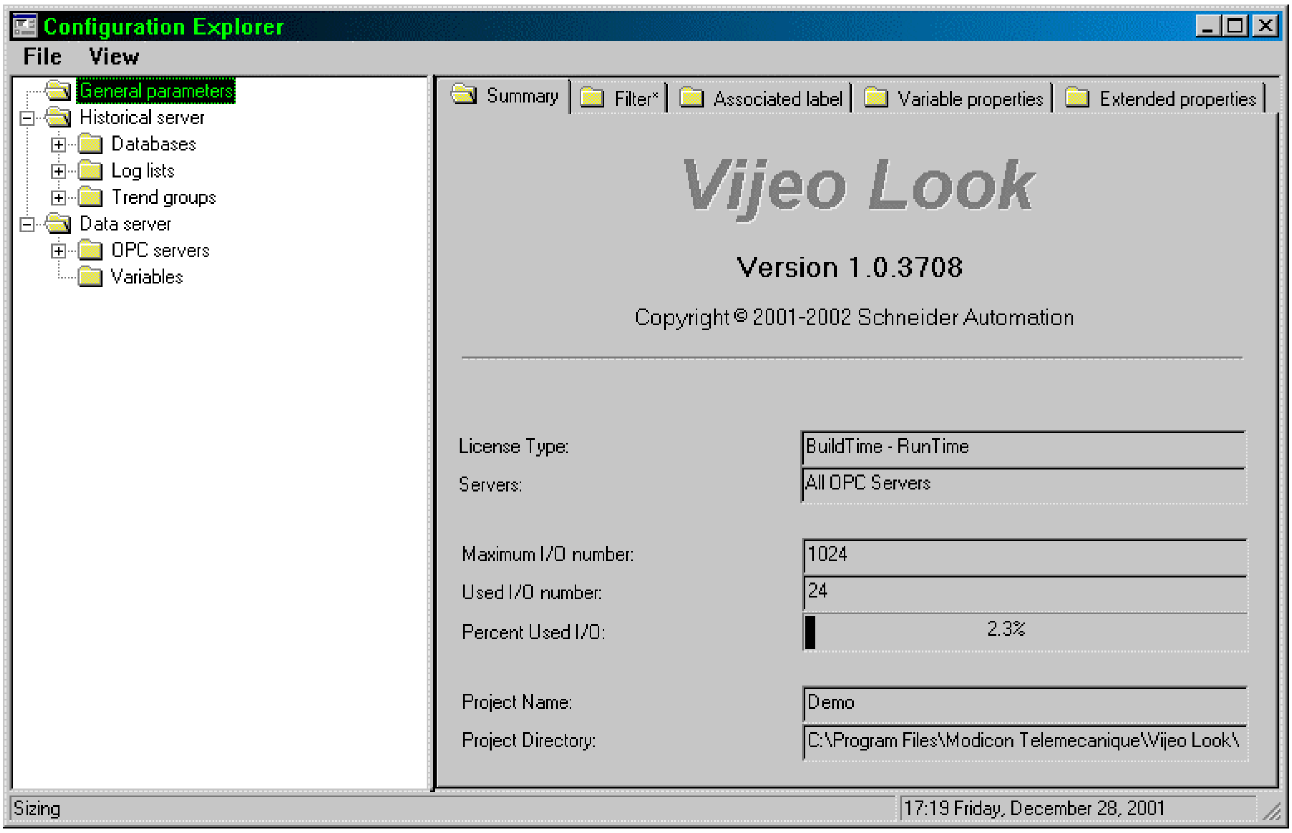
The Tree Control | The Menu

The Configuration Explorer is used to configure the (Real Time) Data Server and Historic (Data) Server. The Configuration Explorer has a menu bar and two panes.

The menu bar contains commands that effect the general behaviour of the Configuration Explorer.

The left pane contains a tree control from which the various configuration aspects may be selected.

The right pane contains one or more configuration tabs, depending on the configuration aspect that has been selected, from where configuration properties are selected.



Version 1.0.3708

Copyright © 2001-2002 Schneider Automation

License Type: BuildTime - RunTime

Servers: All OPC Servers

Maximum I/O number: 1024

Used I/O number: 24

Percent Used I/O: 2.3%

Project Name: Demo

Project Directory: C:\Program Files\Modicon Telemecanique\Vijeo Look\

Sizing

17:19 Friday, December 28, 2001

The Tree Control

The tree control contains three main folders under which all configuration categories are contained.

General parameters

- Configuration of Filters, Associated Labels and Extended properties. Enabling and disabling OPC and vendor specific variable properties.

Historical server (Historical Data Server, HDS)

- Database - Configuration of one or more database(s) for recording Trend and Log data including the definition of the tables and columns they contain.
- Log lists - Configuration of the lists that define what and where events are to be logged.

- Trend groups - Configuration of the groups that define the rate at which variables become eligible to be trend recorded.

Data server (Real Time Data Server, RTDS)

- OPC Servers - Selection and configuration of OPC servers.
- Variables - Selection and configuration of OPC variables that are to be available to the HMI. Creation and configuration of internal variables.

The Menu

The Configuration Explorer menu provides the following commands.

File

- Save - Save any changes made.
- Exit - Close the Configuration Explorer.

View

- Status bar - Enable or disable the status bar at the bottom of the Configuration Explorer dialog box.
- Advanced settings - Enable or disable advanced settings. If not enabled, some of the configuration tabs are not visible.
- Arrange variables - Sorts the list of variables displayed under each OPC server according to the following options.
  Alphabetically.
  By variable type. For example, Bit, Register, Text.
- Variables by filter - Select the variables that are displayed according to their properties.
  None - Deselect all of the filter tick boxes.
  All - Select all of the filter text boxes.
  Trend - If ticked display all variable that are trend recorded.
  Alarm - If ticked, display all bit variables that are alarms.
  Log - If ticked display all variables that have been enabled for logging.
  HMI access - If ticked, display all variables that have been selected as being available to the HMI.
  Any - If ticked, display all variables that match any of the above criteria (Trend, Alarm, Log, HMI access).
- Associated labels - Selection of the language set for Associated Label configurations.
- Historical language - Selection of the language set used when Associated Labels are recorded by the Historical Data Server.

Dynamic and Static Properties

Most of the properties that can be changed within the Configuration Explorer are dynamic; when they are changed the effect is seen immediately. However, there are some properties that affect the internal operation, for which Vijeo Look must be shut down and restarted before any change takes effect.

Configuration Area	Configuration Tab	Property
General Parameters	Filter	All
Database	Database	All
Trend Tables	Log Table	All
Log Tables	Log Table	All
Trend Groups	Trend Group	Set as default
OPC Servers	Properties	All
OPC Server	Diagnostic	All

These properties are highlighted by marking them with an asterisk in the configuration dialog.

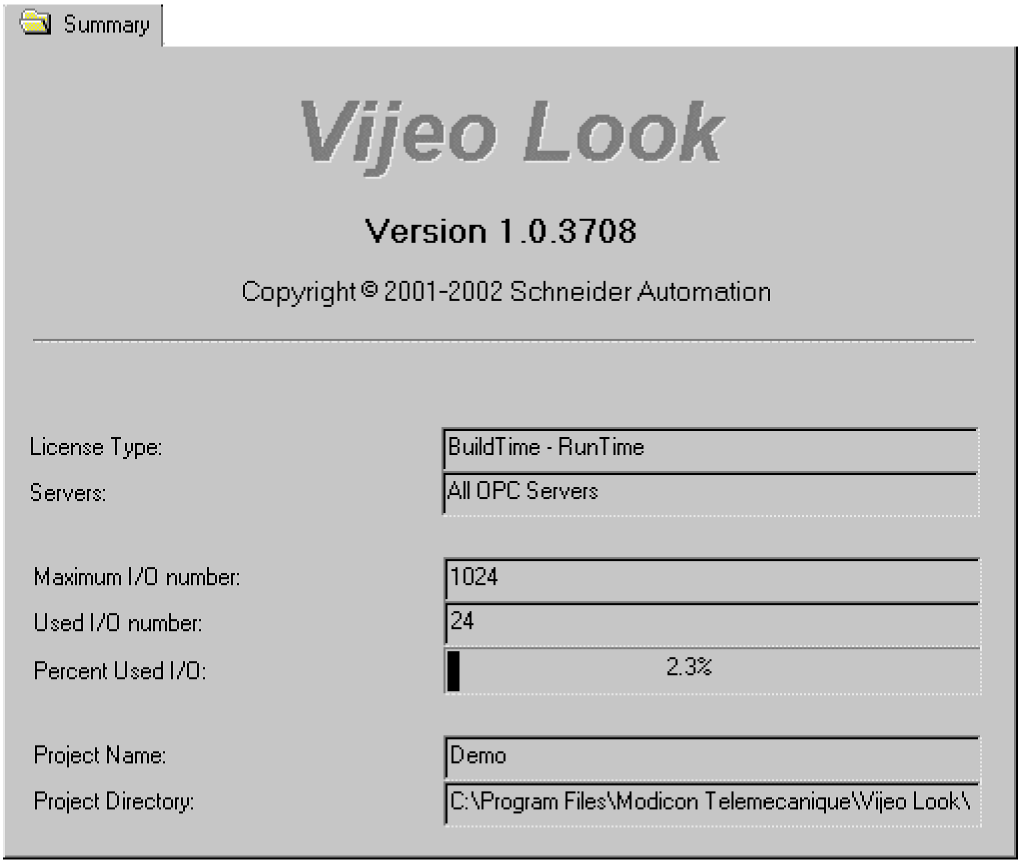
General Properties Folder Overview

The General Properties folder is used to configure options that are inherited by other parts of the Configuration Explorer. Some of the tabs only appear if the Advanced Settings option is selected from the Configuration Explorer menu bar. These are indicated with an $^{(A)}$ in the topic title.

From the General Properties folder you configure:

- Filters
- Associated Labels
- Variable Properties
- Extended Properties

The Summary tab of the General Properties folder displays the software version and information relating to the installed license.



Vijeo Look

Version 1.0.3708

Copyright © 2001-2002 Schneider Automation

License Type: BuildTime - RunTime

Servers: All OPC Servers

Maximum I/O number: 1024

Used I/O number: 24

Percent Used I/O: 2.3%

Project Name: Demo

Project Directory: C:\Program Files\Modicon Telemecanique\Vijeo Look\

Filter Tab

The Filters are used to select variables according to their properties.

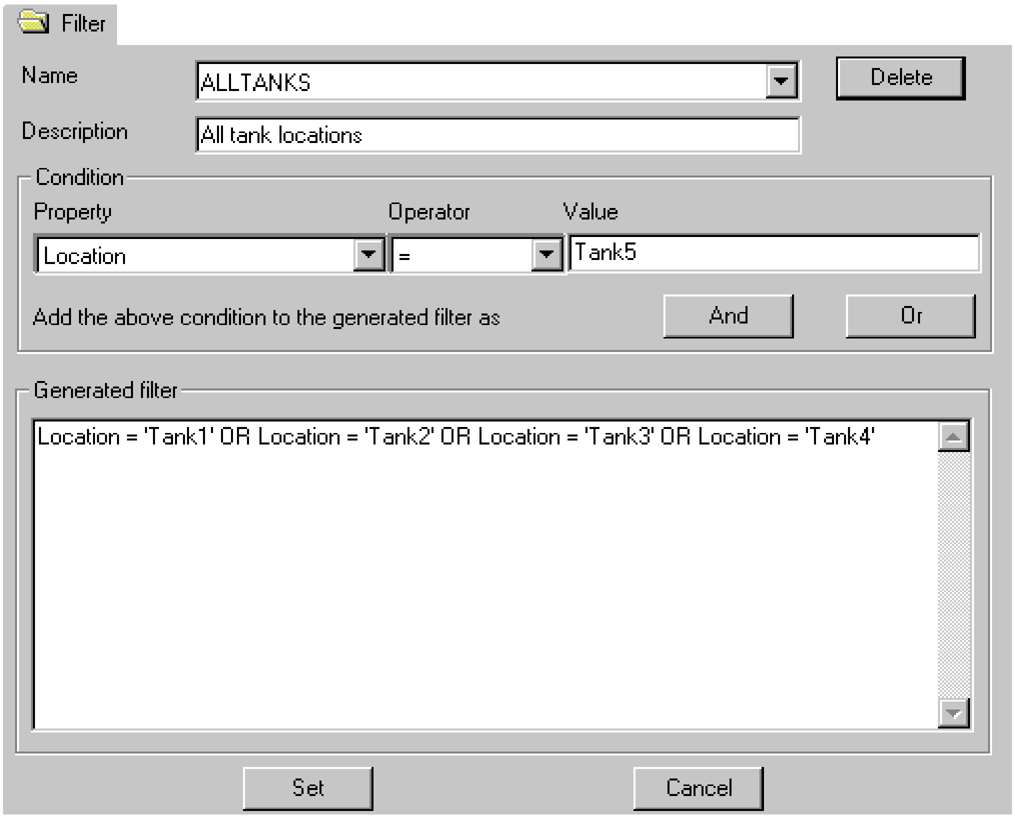
A Filter is able to select variables using the value of the standard variable properties NAME, Description and Units, plus any extended properties that have been configured. Each property may be tested against a fixed value using one of a number of standard operators. The Property/Operator/Value tests may be logically combined using AND or OR to form the complete filter.

Where are Filters used?

- In the Log List configuration to select which variables are recorded in each Database table.
- In the Log Viewer to select the variables from which historic data is displayed.
- In the Alarm Viewer to select the alarms that are displayed.

How to create a filter

38. Select the Filter tab in the General parameters folder.
39. Select *Creation* in the name field and type in a suitable name for the Filter. You can enter an optional description in the Description field.
40. Select the Property and Operator and enter the Value that together form the first test condition.
41. Select the AND button to add the test condition to the filter.
42. Select further Property/Operator/Value test conditions if required and combine them using either the AND or OR buttons.
43. Click the *Set* button to confirm configuration of the filter.



Name: ALLTANKS

Description: All tank locations

Property: Location
Operator: =
Value: Tank5

Location = 'Tank1' OR Location = 'Tank2' OR Location = 'Tank3' OR Location = 'Tank4'

Associated Label Tab$^{(A)}$

An Associated label provides a way of substituting textual information for a change in value of a variable. The text provided by a variable's associated label has the following uses.

- In the Label animation in the HMI. For example it could be used to display Running when a bit is 1 and Stopped when it is 0.
- To replace the standard event strings displayed in the Alarm Viewer.
- To populate some of a variable's OPC properties.

Each Associated Label is actually a suite of four sets of labels, one for each of the languages supported by Vijeo Look. You select the language set that you are going to configure from the Configuration Explorer menu bar command **View.Associated labels**.

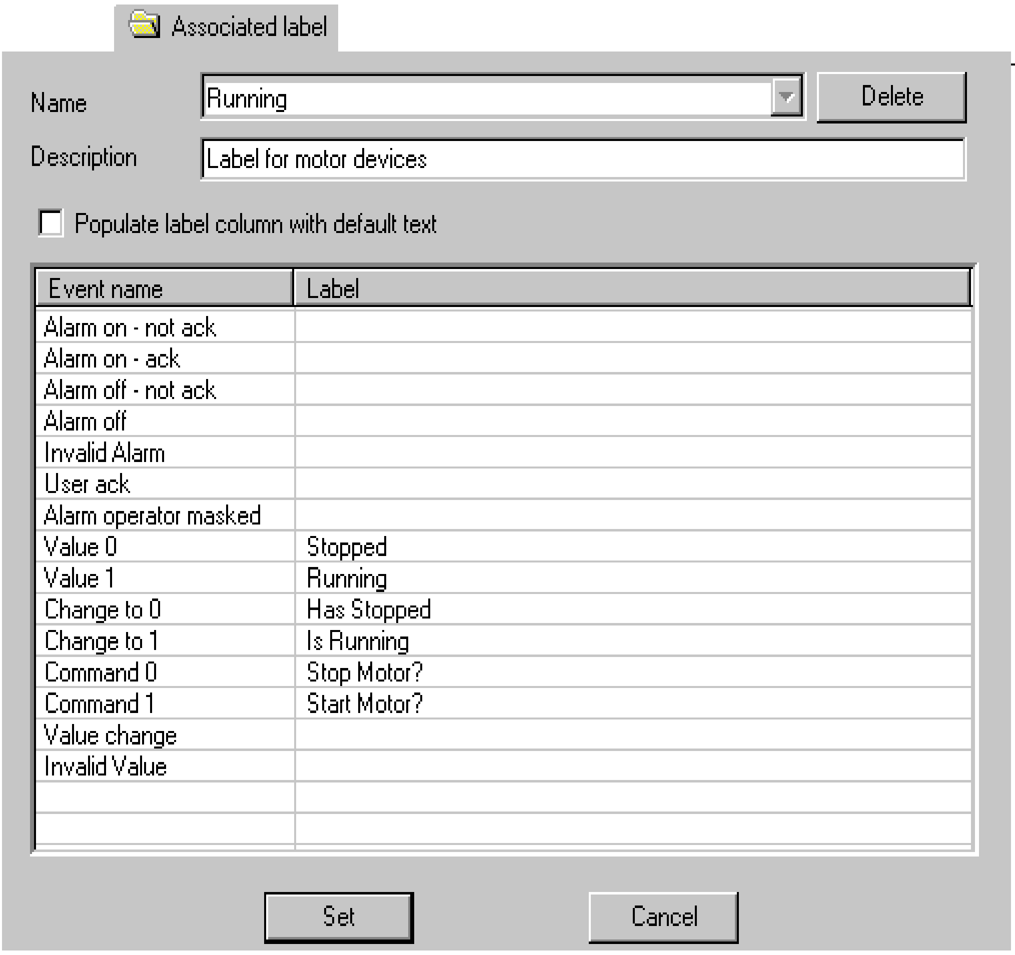
How to create an Associated Label

44. Select the Associated Label tab in the General parameters folder.
45. Select *Creation* in the name field and type in a suitable name for the label. You can enter an optional description in the Description field.
46. Select the language set to be entered using the menu command **View.Associated labels**.
47. Enter text in each of the applicable label fields. It is likely that only some of the fields will be used. For example, if the label is to be attached to bit variable, the alarm fields are meaningless.
48. Click *Set* to confirm the configuration.
49. If the label is to be used in more than one language repeat items 3 to 6 for the other language(s).

The information an Associated Label provides depends on the type of variable it is attached to. See the table below for the use of each field.

Event name	OPC name	Type	Use
Alarm on - not ack	n/a	Alarm	Displayed in the AssociatedLabel field of the Alarm Display and recorded in EVTTITLE field if the variable is logged.
Alarm on - ack	n/a	Alarm	As above
Alarm off - not ack	n/a	Alarm	As above
Alarm off - ack	n/a	Alarm	As above
Invalid alarm	n/a	Alarm	As above
User ack	n/a	Alarm	As above
Alarm masked	n/a	Alarm	As above
Value 0	CloseLabel	Bit	Displayed in the Associated Label animation.
Value 1	OpenLabel	Bit	Displayed in the Associated Label animation.
Change to 0	n/a	Bit	Recorded in the EVTTITLE field if the variable is logged.
Change to 1	n/a	Bit	Recorded in the EVTTITLE field if the variable is logged.
Command 0	ForceOffLabel	Bit	Displayed in the confirmation dialog box when the User is forcing a bit.
Command 1	ForceOnLabel	Bit	Displayed in the confirmation dialog box when the User is forcing a bit.
Value change	n/a	Register	Recorded in the EVTTITLE field if the variable is logged.
Invalid value	InvalidValue	All	Recorded in the EVTTITLE field if the variable is logged.

When using the Associated Label, default entries will be used for any field that has not been completed.



Name: Running

Description: Label for motor devices

□ Populate label column with default text

Event name	Label
Alarm on - not ack	
Alarm on - ack	
Alarm off - not ack	
Alarm off	
Invalid Alarm	
User ack	
Alarm operator masked	
Value 0	Stopped
Value 1	Running
Change to 0	Has Stopped
Change to 1	Is Running
Command 0	Stop Motor?
Command 1	Start Motor?
Value change	
Invalid Value	

Variable Properties Tab(A)

This tab allows you to select which of the variable properties (in addition to the value, timestamp and quality) are available to the HMI and the Historical Data Server. The default setting of all properties available need not be changed for the majority of projects.

The advantage of reducing the number of available properties is that it saves system resources (memory, CPU loading etc). This should only be necessary under exceptional circumstances.

Changing the variable properties that are available will also effect what is available in the following other configuration tabs.

- The Filter Tab in the General parameters folder (Description and Units only).
- The Log Table tab in the Databases folder.
- The Properties tab in folder for the each configured OPC server.
- The Properties tab of each variable folder (Depending on variable type).

The properties that are available are as follows.

Name in Variable Properties Tab	OPC Name	Description	Key
Description	Description	A textual description of the variable. Example: TANK 1 LOW LEVEL	**M**
Units	Units	The engineering units of a register variable. Example: LITRES	**O**
High/Low limits	HiValueLimit LoValueLimit	The minimum and maximum displayable value for a register variable.	**O**
Open/Close labels	OpenLabel CloseLabel	Supplied by a variable's Associated Label. Equivalent to the Value 1 and Value 0 fields.	**O**
Security level	SecurityLevel	The security level for the variable. A User must have the corresponding security rights to be able to change the value of the variable at run time. Range 0 to 63.	**V**
Display format	Format	The string specifying the display format for a register variable. This is used if the auto format option is selected in the register display.	**V**
Force On/Off labels	ForceOnLabel ForceOffLabel	Supplied by a variable's Associated Label. Equivalent to the Command 1 and Command 0 fields.	**V**
Invalid label	InvalidLabel	Supplied by a variable's Associated Label. Equivalent to the Invalid Value field.	**V**
Alarm level	AlarmLevel	The level of the alarm. Supplied by the configuration of the variable in the RTDS.	**V**
Alarm state labels	AlarmStateLabel	A text array providing information on the status of the alarm to the Alarm Viewer.	**V**

The OPC Name is the name by which the property is known within the HMI.

The Key indicates the type of property.

M Mandatory properties specified by the OPC standard. These must provided by all OPC servers.

O Optional properties specified by the OPC standard. These may be optionally provided by the OPC server. Check the documentation supplied with your server for details of which properties it supports.

V Vendor specific properties. These are properties that are specific to Vijeo Look.

If you de-select a property any values that you entered for it in a variable's configuration will be lost.

Extended Properties Tab$^{(A)}$

This tab allows you to create customised properties that are available to all variables. For example you could create a property that represents the physical location of the variable's source. You can create up to 16 Extended Properties. Each property can be either a string (Text), a number (Integer, Long, Float or Double) or a Boolean.

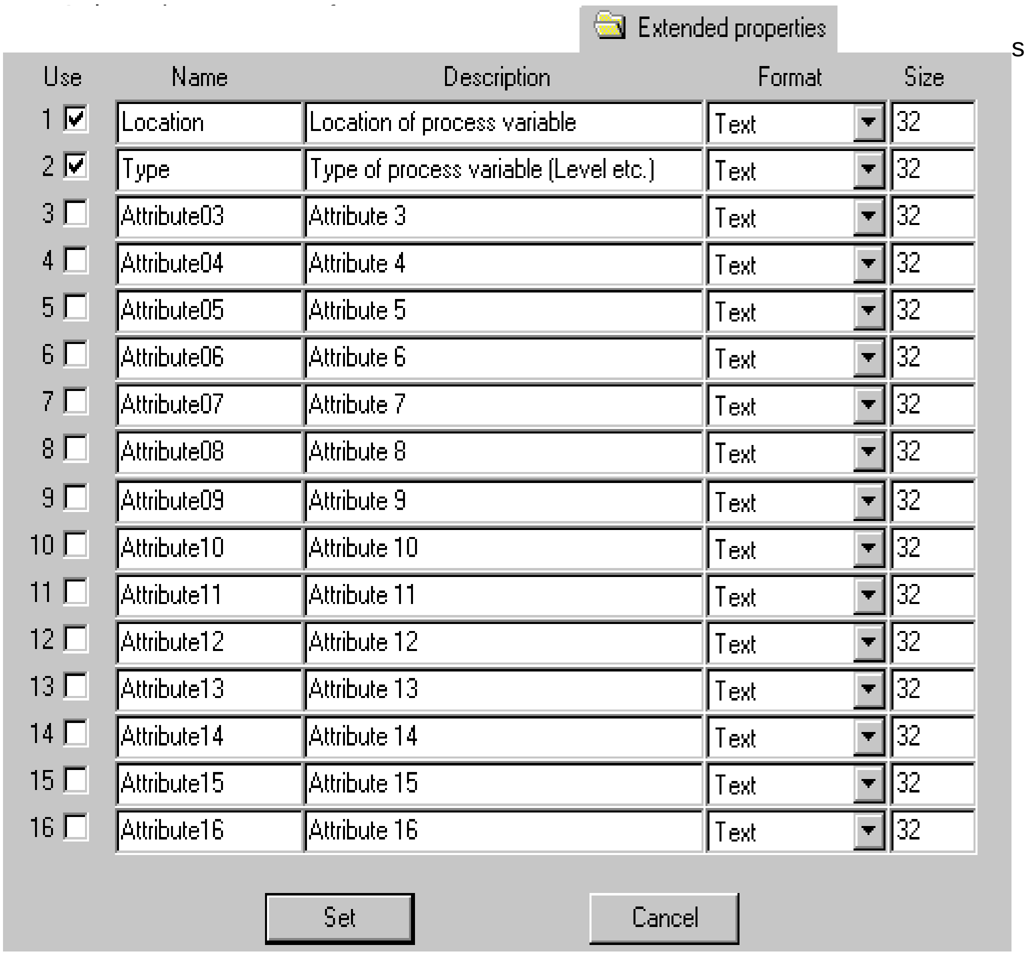
Where are Extended Properties Used?

Configured Extended Properties appear in the following configuration tabs.

- In the Extended properties tab for each variable. Used to give the property a value specific to that variable.
- In the Trend and Log table tabs. The value of an extended property can be recorded by the Historical Data Server.
- In the Filters tab. The value of an extended property may be part of the filter configuration.

How to create an Extended property

50. Select the Extended Properties tab in the General parameters folder.
51. Tick the *Enable* field adjacent to the property to enable it.
52. Enter a name by which the property will be known. The name appears as a reference in the other tabs supporting Extended Properties. You can also enter an optional description.
53
54 s.
55



Use	Name	Description	Format	Size
1 ☑	Location	Location of process variable	Text	32
2 ☑	Type	Type of process variable (Level etc.)	Text	32
3 □	Attribute03	Attribute 3	Text	32
4 □	Attribute04	Attribute 4	Text	32
5 □	Attribute05	Attribute 5	Text	32
6 □	Attribute06	Attribute 6	Text	32
7 □	Attribute07	Attribute 7	Text	32
8 □	Attribute08	Attribute 8	Text	32
9 □	Attribute09	Attribute 9	Text	32
10 □	Attribute10	Attribute 10	Text	32
11 □	Attribute11	Attribute 11	Text	32
12 □	Attribute12	Attribute 12	Text	32
13 □	Attribute13	Attribute 13	Text	32
14 □	Attribute14	Attribute 14	Text	32
15 □	Attribute15	Attribute 15	Text	32
16 □	Attribute16	Attribute 16	Text	32

! After you have made any changes in the Extended properties Tab Vijeo Look must be shut down and restarted. This is essential to allow Vijeo Look to create the corresponding OPC properties.

Real Time Data Server Overview

Configuring Data Update Rates | The I/O Information Tab | How the I/O is Counted

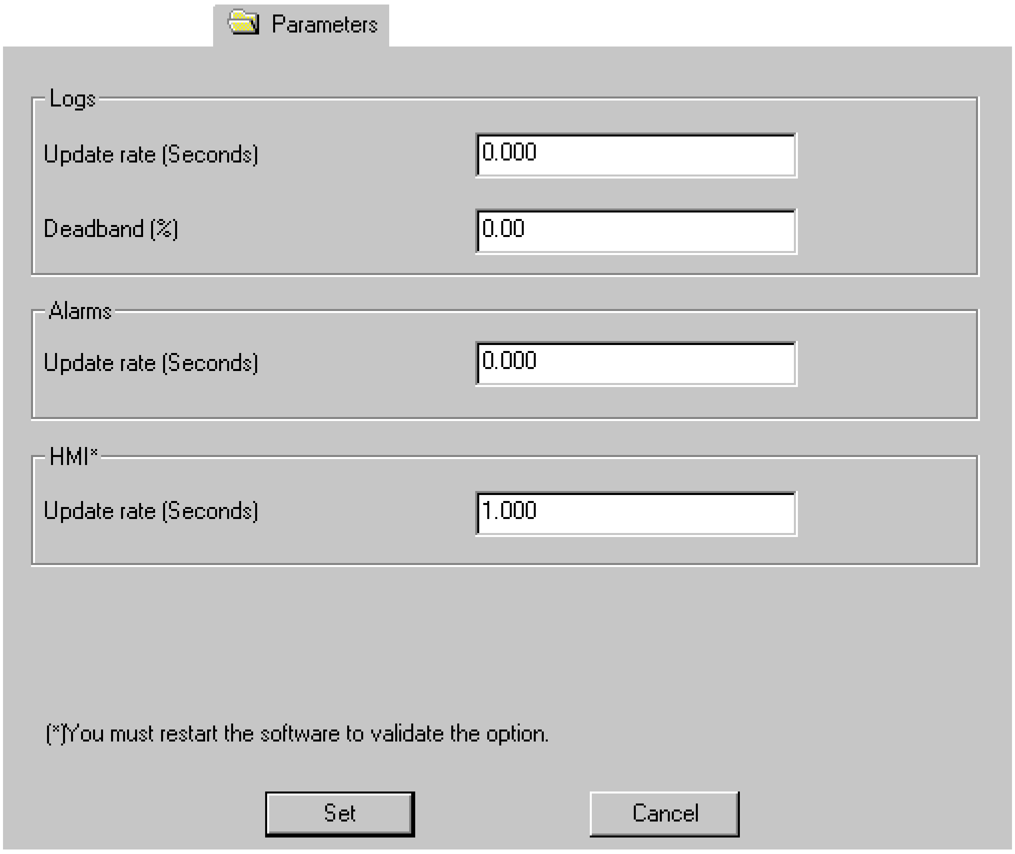
The Real Time Data Server (RTDS) provides an interface between the OPC server(s), the HMI, Historical Data Server (HDS) and the Real Time Alarm Server (RTAS).

The HMI, HDS and RTAS require information that is not supplied as standard by most OPC servers; for example, the HiValueLimit and LoValueLimit for register variables. The RTDS provides the tools by which this information is added. The RTDS can be thought of as a tool for enhancing the information provided by OPC servers.

The RTDS also allows the creation of internal variables. These are variables that are available to the HMI, HDS and RTAS but have their value calculated in some way rather than coming from an OPC server.

Configuring Data Update Rates

The Parameters tab is displayed by clicking on the Data Server folder in the left pane of the Configuration Explorer. It allows you to configure the rate at which the RTDS will request real time data from the OPC servers. You can individually configure the update rate for Logs (data supplied to and recorded by the HDS), Alarms (supplied to the RTAS) and the real-time data supplied to the HMI. In addition you can configure a deadband which is used when logging register variables.



Logs

Update rate (Seconds) 0.000

Deadband (%) 0.00

Alarms

Update rate (Seconds) 0.000

HMI*

Update rate (Seconds) 1.000

(*)You must restart the software to validate the option.

The update rate for Trend data is configured by the Trend Groups.

What is the fastest rate at which data may be updated?

The fastest rate at which data may be updated is the greater of three values; the Update rate entered in the Parameters tab, the minimum update rate supported by the RTDS and the minimum update rate supported by the OPC Server. The minimum rate supported by the RTDS is 0.1 seconds. The minimum rate supported by the OPC server will depend on the OPC server is being used. In the case of the OFS it is configurable and set to a default of 1 second.

- Example 1. Log update rate 0.5 seconds. OFS server update rate 1 second. The actual Log update rate would be 1 second.
- Example 2. Alarm update rate 0 seconds. OFS Server update rate 0.1 seconds. The actual Alarm update rate would be 0.1 seconds.

In practice, the actual update rate achieved will also be influenced by other factors including the physical connection between the PLC and OPC server and the protocol being used.

You must close all mimics before any changes in the HMI (Data access) rate are taken into account.

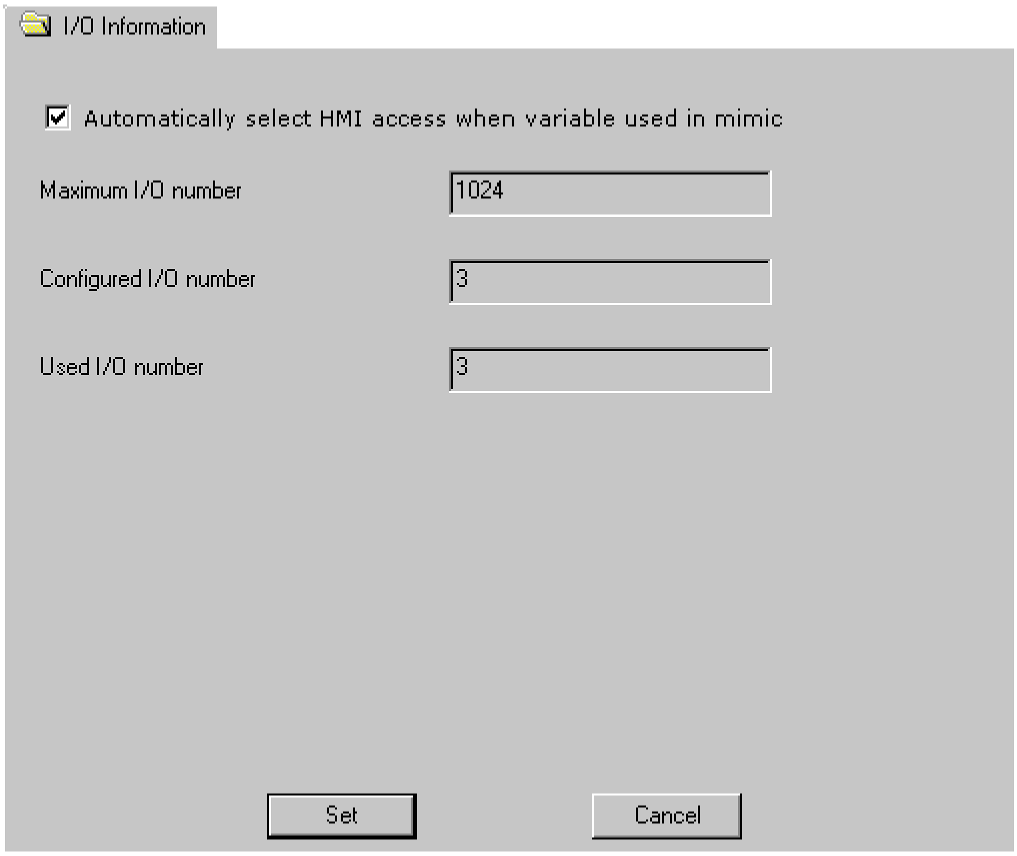
How does the deadband work?

The deadband is only taken into account when logging (not Trend recording) register variables. The deadband is entered as a percentage of the register range in the OPC server. Before any value change is notified to the RTDS the value of a register must change by more than the deadband.

The deadband is sent to and managed by the OPC server. Not all 3rd party OPC servers support this feature.

The I/O Information Tab

The I/O Information tab is displayed by clicking on the Data Server folder in the left pane of the Configuration Explorer. It displays information on the available and used I/O.


I/O Information

☑ Automatically select HMI access when variable used in mimic

Maximum I/O number: 1024

Configured I/O number: 3

Used I/O number: 3

If the property *Automatically select HMI access when variable used in mimic* is selected, the first time you use a variable in a mimic, Vijeo Look will automatically set the *HMI Access* property in the variable's configuration and it will be included in the I/O count. Otherwise you will have to manually enable *HMI Access* for each variable before you use it in a mimic.

Automatic setting of the *HMI Access* property only works when displaying the real time value of a variable in a mimic. It does not work if you are displaying another property of the variable (its description for example).

How the I/O is Counted

The number of I/O points (variables) that may be used in a project is limited by the installed license to 25, 128, 512 or 1024. The properties that decide if a variable contributes towards the I/O count depends on the license type.

For the 25, 128 and 512 versions

- HMI Access
- Log
- Trend

For the 1024 version

- HMI Access

A variable is only counted once. If you have the Log, Trend and HMI Access options all selected it only adds one to the I/O count.

Selecting and Configuring OPC Servers

Configuring an OPC Server | The Properties Tab | The Diagnostic Buffer Tab

The OPC servers you are able to configure depends on the license type that has been installed.

- Versions with 128 or 512 I/O support only the OFS server. The set-up is pre-configured.
- Versions with 1024 I/O support the OFS server plus the possibility to add further servers from a 3rd party manufacturer.

Configuring an OPC Server

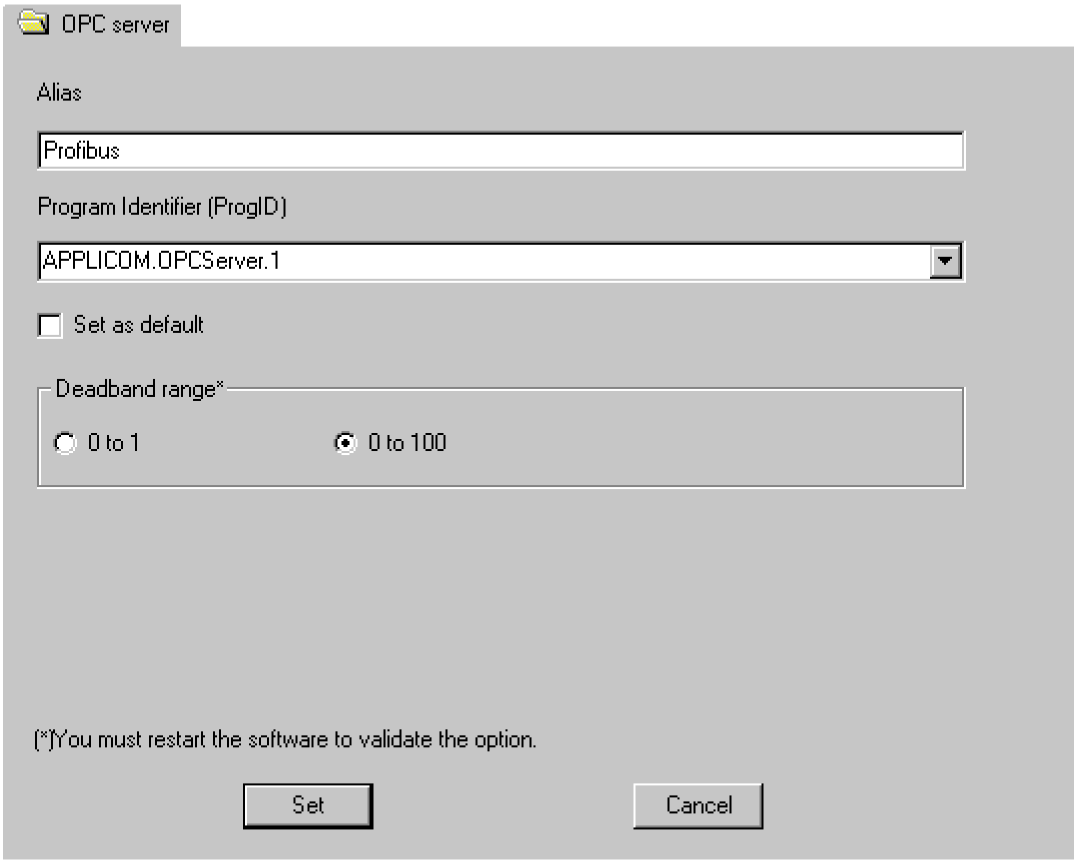
To configure an OPC server you must supply the Configuration Explorer with the following information.

- Alias - the name by which the OPC server will be referred.
- Program Identifier - the unique identifier by which the OPC server is known within the operating system. Clicking the down arrow to the right of the field will display the list of installed servers.

In the case of the OFS server this information is pre-configured.

How to add a 3rd party OPC server

56. In the left pane of the Configuration Explorer, expand the tree to display the (Real Time) *Data Server* folder.
57. Right click on the folder and, from the pop-up menu, select the *New* command.
58. In the right pane enter an alias name for the new OPC server.
59. Select the OPC server program ID. A list of servers available on the selected computer can be displayed using the down arrow button adjacent to the *Program ID* field.
60. Select the deadband range. Some OPC servers expect the deadband to be in the range 0 to 1 whilst others expect it to be 0 to 100. Check the manufacturer's documentation to see which one you need.
61. Click *Set* to confirm the configuration.



OPC server

Alias

Profibus

Program Identifier (ProgID)

APPLICOM.OPCServer.1

☐ Set as default

Deadband range*

○ 0 to 1 ● 0 to 100

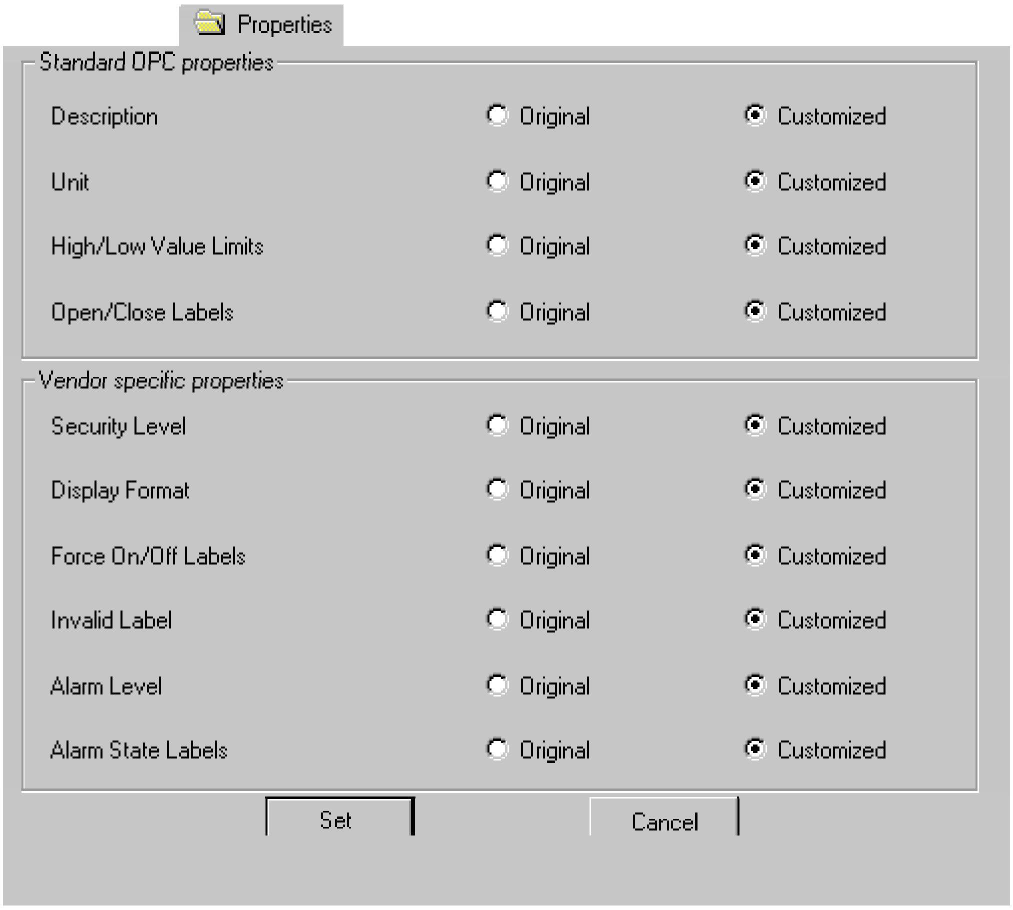
(*)You must restart the software to validate the option.

The use of remote OPC servers is not supported.

The Properties Tab

The Properties tab allows you to select the source for each of the standard OPC and vendor specific properties. There is a separate Properties tab for each OPC server that is configured.

- *Original* - The value for the selected property is supplied directly by the OPC server.
- *Customised* - The value for the selected property is supplied by the RTDS. For some properties the value is entered in the configuration of the variable, for others it is supplied by the variable's Associated Label. The OPC server will provide a default value if it supports the property.



Properties

Standard OPC properties

Description	○ Original	● Customized
Unit	○ Original	● Customized
High/Low Value Limits	○ Original	● Customized
Open/Close Labels	○ Original	● Customized

Vendor specific properties

Security Level	○ Original	● Customized
Display Format	○ Original	● Customized
Force On/Off Labels	○ Original	● Customized
Invalid Label	○ Original	● Customized
Alarm Level	○ Original	● Customized
Alarm State Labels	○ Original	● Customized

The Diagnostic Buffer Tab

See the topic Using the Diagnostic Buffer.

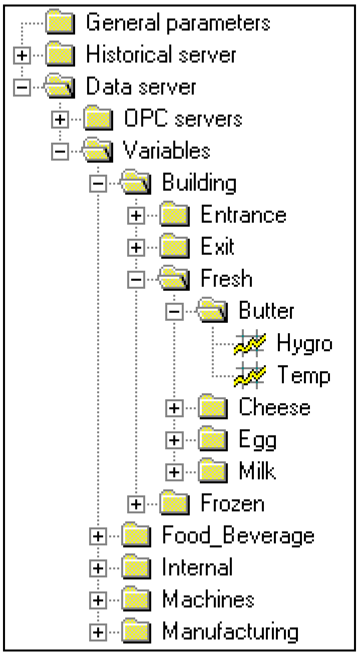
Browsing and Creating Variables

Creating Folders in Which to Organise Variables | Creating an Internal Variable | Creating an External Variable

The variables available to the RTDS are displayed in the tree control in the left pane of the Configuration Explorer. The list of available variables appears under the Variables folder organised according to the variable name. Selecting a variable name will display its properties in the right pane.

The variables that appear in the Variables folder are generated in one of three ways.

- External variables supplied by the OPC server browse facility. The variable names and folders in which they appear reflect the names of the variables in the OPC server.
- External variables that have been manually configured. The variable names and folders in which they appear are configured within the RTDS. The name of the variable to which they are linked in the OPC server is manually entered.
- Internal variables. The variable names and folders in which they appear are configured within the RTDS. These variables have no connection to an OPC server.



- General parameters
- Historical server
- Data server
  - OPC servers
  - Variables
    - Building
      - Entrance
      - Exit
      - Fresh
        - Butter
          - Hygro
          - Temp
        - Cheese
        - Egg
        - Milk
      - Frozen
    - Food_Beverage
    - Internal
    - Machines
    - Manufacturing

In the tree display the variable type is indicated by an icon adjacent to the variable name.

01 A bit variable. Used to represent the Boolean type in the OPC server.

A register variable. Used to represent any analogue variable type in the OPC server.

A text variable. Used to represent the String type in the OPC server.

Sorting variables within the display tree

The display of variables may be sorted alphanumerically or by type by clicking on the sub-folder to be sorted and selecting the *View.Arrange* command from the Configuration Explorer menu.

Creating Folders in Which to Organise Variables

Variables that are manually created within the RTDS may be organised in sub-folders. In the HMI, the sub-folders appear as part of the variable name and are known as the branch. Organising variables in sub-folders has two major advantages.

- It is much easier to search for a variable in the RTDS and the HMI.

- The project is able to take full advantage of the object technology in the HMI.

How to create a Folder in which to organise variables

62. In the left pane of the Configuration Explorer right click on the *Variables* folder and, from the pop-up menu, select the *New sub-branch* command.
63. In the right pane enter a name for the new folder and select *Set* to confirm.
64. Further sub-folders may be created as required.

With variables supplied by the browse facility of an OPC server, the folders are created automatically according to the names of the variables.

Creating an Internal Variable

Internal variables may be created directly under the Variables folder or under one or more sub-folders.

Creating an internal variable

65. Right click the folder under which the variable will appear. From the pop-up menu, select either *bit*, *register* or *text* as applicable.
66. In right pane enter a name for the variable and select *Internal* type.
67. Click *Set* to confirm.

Variables names must be unique. You cannot create an internal variable with the same name as one supplied by an OPC server.

Creating an External Variable

If you are using an OPC server that does not support the browse facility you will have to manually configure any of its variables that are to be used in the project. External variables may be created directly under the Variables folder or under one or more sub-folders.

How to create an external variable

68. Right click the folder under which the variable will appear. From the pop-up menu, select either *bit*, *register* or *text* as applicable.
69. In right pane enter a name for the variable and select *External* type.
70. Select the OPC server that is to be the source of the variable using the *Server Name* drop down list box.
71. In the text field adjacent to the *External* option button, type in the name of the variable by which it is known in the OPC server.
72. Click *Set* to confirm.

The RTDS is unable to validate the name you enter. The first indication you will get is when you use the variable in a mimic and its value is invalid.

Configuring Variables

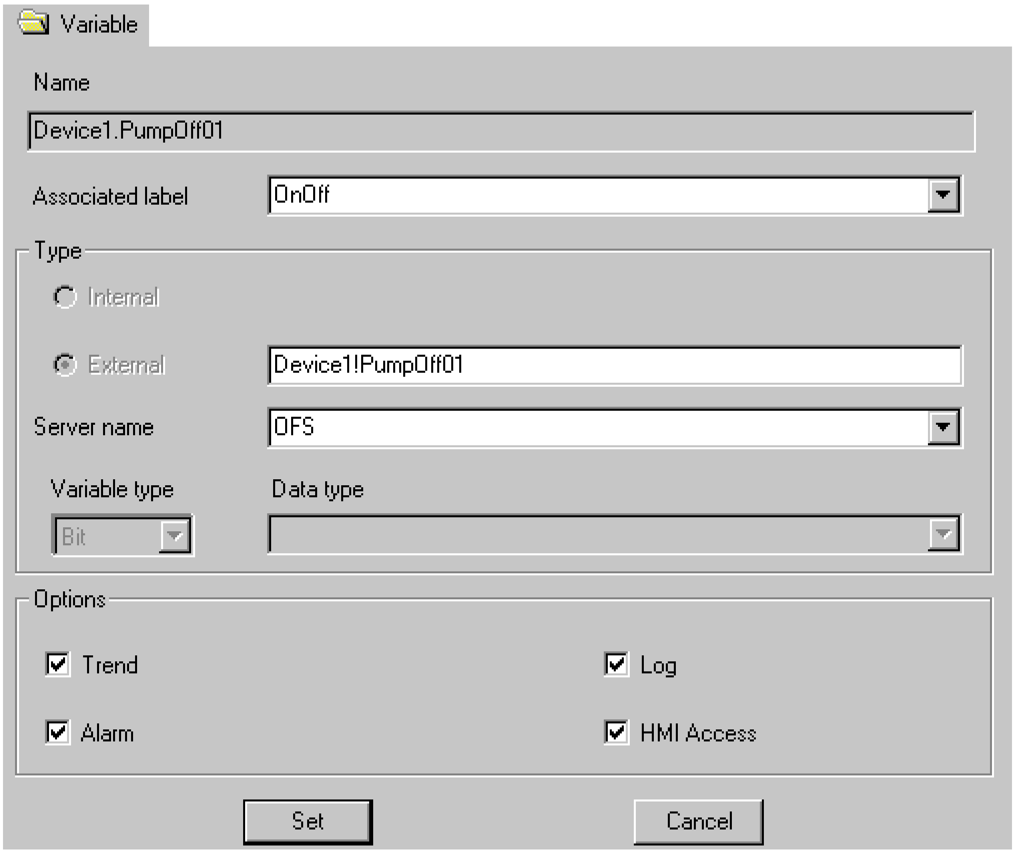
The Variable Tab | The Properties Tab | The Extended Properties Tab | The Trend Tab | The Alarm Tab

A variable's properties are displayed and configured by selecting its name in the left pane of the Configuration Explorer. The properties are organised in a number of tabs according to the part of the variables' behaviour they effect.

The Variable Tab

The Variable tab is used to set up the basic properties for each variable.

- *Type* - The type of variable, either Internal (value managed within the RTDS) or External (value managed by an OPC server).
- *Associated label* - Allows the selection of a previously configured Associated Label. Associated Labels provide a way of substituting textual information for a change in value of the variable. For further information see the topic on Associated Labels.
- *Trend* - Allows the variable to be trend recorded. Trend Recording records the value, quality and timestamp of the variable in the database. The history of the variable may then be played back through a Trend Display or imported into an external database application such as Access. Both bit and register variables may be trend recorded. If you have enabled the *Advanced* option in the Configuration Explorer menu an additional tab will be displayed to allow customisation of the trend recording for that variable.
- *Log* - Allows the variable to be logged. Logging records the value, quality and timestamp of the variable in the database each time it changes. Note that if you have configured any log filters these will also effect what is logged.
- *HMI Access* - Enables access to the variable by the HMI and programming language.
- *Alarm* - Enables alarm behaviour for bit variables. If you have enabled the *Advanced* option in the Configuration Explorer menu an additional tab will be displayed to allow customisation of the alarm behaviour for that variable.



Name: Device1.PumpOff01

Associated label: OnOff

○ Internal

● External: Device1!PumpOff01

Server name: OFS

Variable type: Bit

Data type:

☑ Trend ☑ Log

☑ Alarm ☑ HMI Access

The Properties Tab$^{(A)}$

The *Properties* tab contains a list of configurable properties that are available to the HMI and HDS in addition to the variable's value, timestamp and quality. The properties may be displayed or used by certain animations, and also recorded in the database. Some properties are those that are supported as standard by OPC (for example, Description) whilst others are vendor specific.

The properties that are available depend on:

- The variable type (Register, Bit or Text).
- What has been enabled in the *Variable Properties* tab of the General folder.

The standard OPC properties may inherit a value from the OPC server (if supported). The value of each property may be changed if the corresponding option in the *Properties* tab of the OPC server has been set to customise.

Property	**Type**	**Available for**
Description	Standard	All variables
Units	Standard	Register only
LoValueLimit	Standard	Register only
HiValueLimit	Standard	Register only
SecurityLevel	Specific	All variables
Format	Specific	Register only

How to change the value of a property

73. Open the Configuration Explorer.

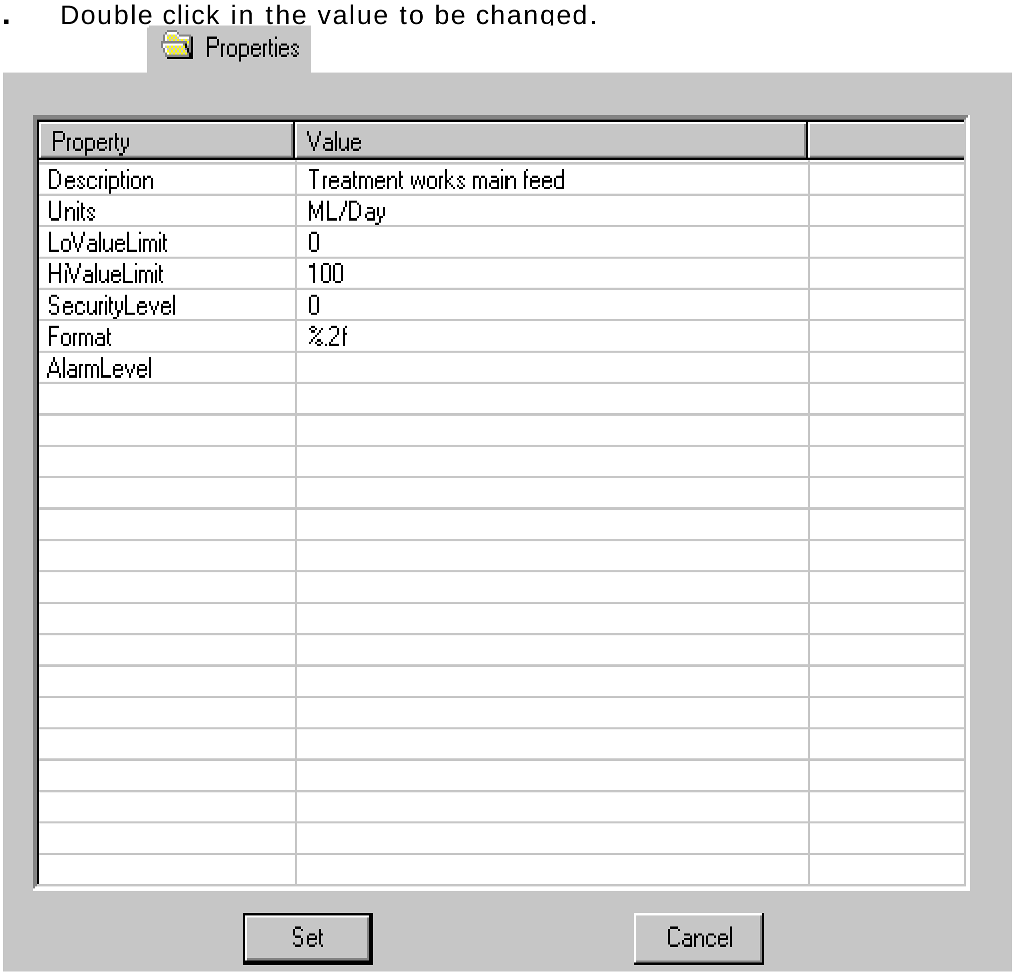
74. Expand the tree in the left pane to display the list of variables. Select the variable to be modified.

75. In the right pane select the *Properties* tab.

76. Double click in the value to be changed.

77

78

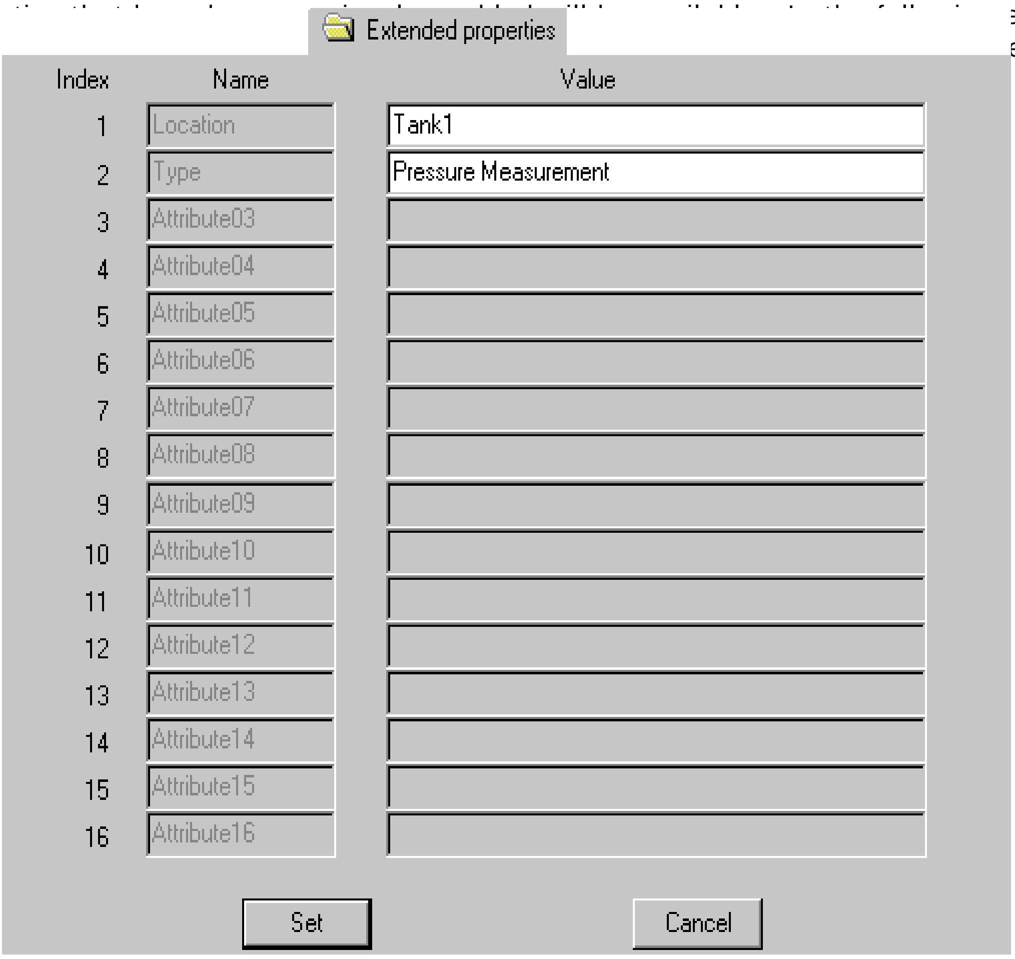


Property	Value	
Description	Treatment works main feed	
Units	ML/Day	
LoValueLimit	0	
HiValueLimit	100	
SecurityLevel	0	
Format	%.2f	
AlarmLevel		

If you change the properties of an external (OPC) variable, the new values for the properties are saved within the project - it does not affect the configuration of the OPC server.

The Extended Properties Tab(A)

The *Extended Properties* tab allows you to enter a value for any of the extended properties that were enabled in the *Extended Properties* tab of the General Configuration folder. Only prope example two prope ed (text, Boole



Index	Name	Value
1	Location	Tank1
2	Type	Pressure Measurement
3	Attribute03	
4	Attribute04	
5	Attribute05	
6	Attribute06	
7	Attribute07	
8	Attribute08	
9	Attribute09	
10	Attribute10	
11	Attribute11	
12	Attribute12	
13	Attribute13	
14	Attribute14	
15	Attribute15	
16	Attribute16	

If you add extended properties to an external (OPC) variable, the values for the properties are saved within the project - it does not affect the configuration of the OPC server.

The Trend Tab(A)

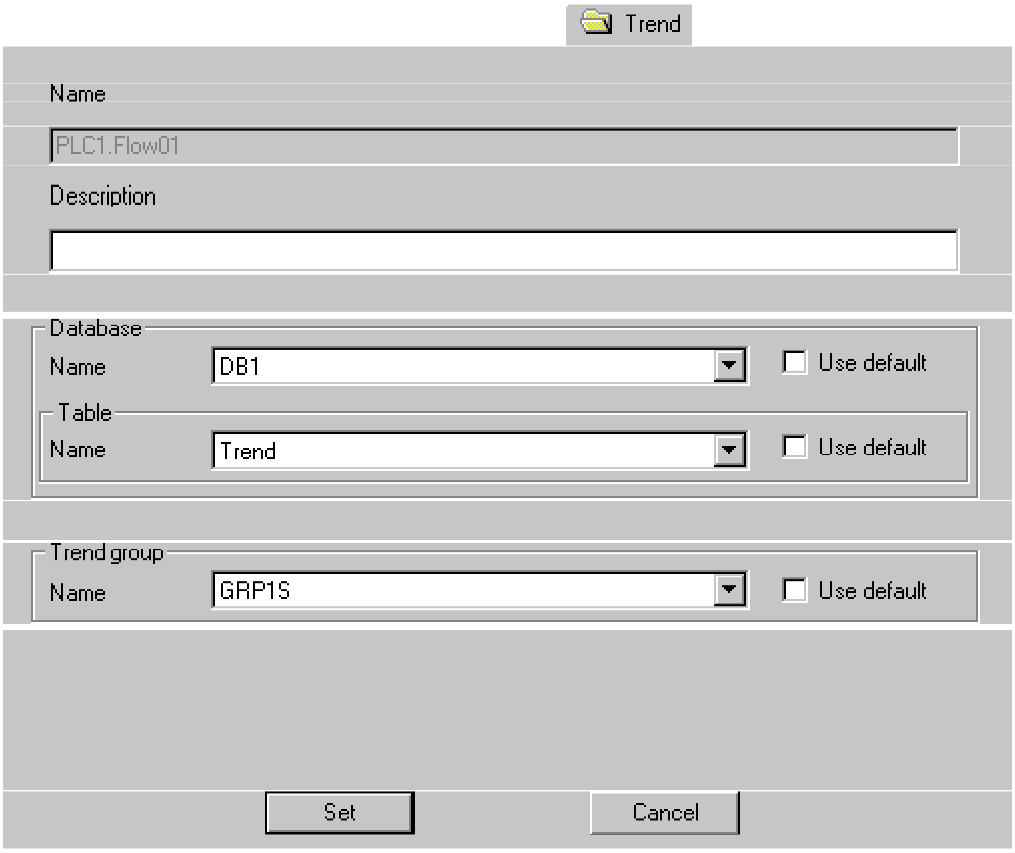
The *Trend* tab allows you to customise the trend recording for a variable.

Unless you specify otherwise the variable will be recorded in the database table that was configured as default. Using the options in the *Trend* tab you can select the database and Table in which the variable is recorded. You can also select the Trend Group which determines the period at which the variable is recorded.

What is the Trend group?

The Trend Group defines the period at which the variable becomes eligible for recording. If it has not changed during this period it will not be recorded. For example a register is changing approximately once every 5 seconds. If it was allocated to a Trend Group with a sampling period of 1 second it would be recorded every 5 seconds. If it was allocated to a Trend Group with a sampling period of 10 seconds it would be recorded every 10 seconds.

See the topic on Configuring a Trend Group for further information.



Trend

Name

PLC1.Flow01

Description

Database

Name DB1 □ Use default

Table

Name Trend □ Use default

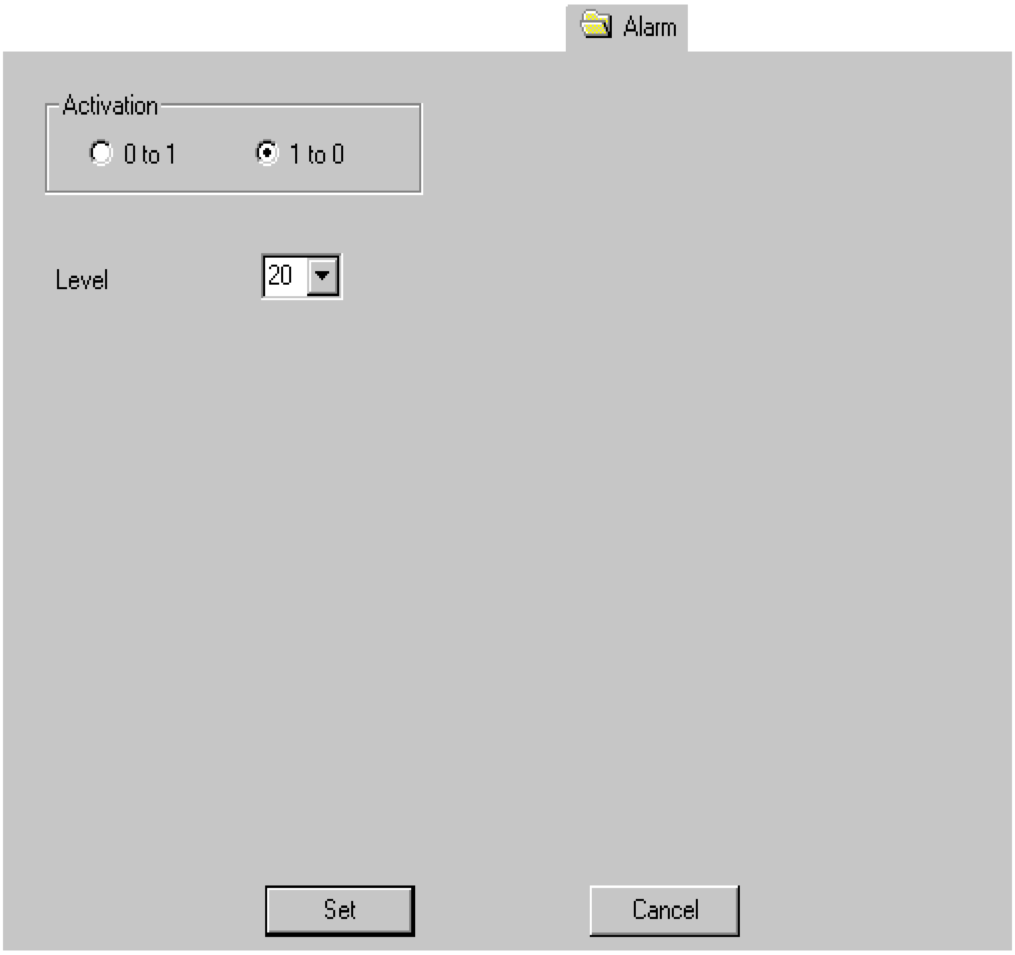
Trend group

Name GRP1S □ Use default

The Alarm Tab(A)

The *Alarm* tab allows you to customise the alarm behaviour for a bit variable.

- *Activation 0 to 1, 1 to 0* - Selects which transition is to be used as the active state (On and Not Acknowledged).
- *Level* - The level of the Alarm in the range 0 to 29. An Alarm's level defines its colour and behaviour in the Alarm Viewer, and (according to the User's profile) which Users are



Activation
○ 0 to 1 ● 1 to 0

Level 20

Variable Properties Reference

Displaying Variable Properties in a Mimic

The following tables list the standard variable properties that are available in addition to the real-time value.

Properties supported by all variables

Name	Type	Source	Use
Description	Text	OPC Mandatory	A T L
SecurityLevel	Register	Vendor Specific	A T L
InvalidLabel	Text	Associated Label	A

Properties specific to Bit variable

Name	Type	Source	Use
CloseLabel	Text	OPC Standard or Associated Label	A T L
ForceOfflabel	Text	Associated Label	A T L
ForceOnLabel	Text	Associated Label	A T L
OpenLabel	Text	OPC Standard or Associated Label	A T L

Properties specific to Alarm variables

Name	Type	Source	Use
AlarmLevel	Register	RTDS	A T L
AlarmState	Register	RTAS	*
AlarmStatelabels	Text	RTAS	*
CloseLabel	Text	OPC Standard or Associated Label	A T L
ForceOfflabel	Text	Associated Label	A T L
ForceOnLabel	Text	Associated Label	A T L
OpenLabel	Text	OPC Standard or Associated Label	A T L

Properties specific to Register variables

Name	Type	Source	Use
Format	Text	Format	A T L
HiValueLimit	Register	OPC Standard	A T L
InvalidLabel	Text	Associated Label	A T L
LoValueLimit	Register	OPC Standard	A T L
Units	Text	OPC Standard	A T L

Properties specific to Text variables

There are no properties specific to Text variables.

Key for the Use column

L - Log Table

T - Trend Table

A - Animation

* - Internal use only

Displaying Variable Properties in a Mimic

You can display the value of any variable properties, including any Extended properties, in a mimic using the appropriate animation. For example, to display the Description, you would have to use an animation that displayed text.

Extended properties only may also have their values changed from a mimic using the appropriate animation. For example, you could change a numeric Extended property using the Send-register animation.

To use a variable property, rather that its real-time value, you append the name of the property to the variable name. For example, to display the Description of the variable Building.Fresh.Butter.Hygro you would use Building.Fresh.Butter.Hygro.Description.

System Variables

A number of pre-configured variables (known as system variables) are provided that contain information about the state of Vijeo Look's OPC servers and the system date and time. These may be logged, alarmed and used in Trends and animations the same as any other variable.

Name	Type	Read / Write	Source	Use
System. DateTime	Text	R	System date and time.	
System. HDS.ErrorCount	Register	R/W (reset)	Increments by 1 each time there is an error or warning from the HDS.	If logged allows the User to see a history of any errors with the HDS. May be reset by the User.
System. HDS.Overflow	Bit	R	Set to one when the HDS stops storing information because it reaches the MaxBufferedEvents limit. This limit is fixed in HDS.INI file. It specifies the maximum number of messages that may be buffered in memory.	If this variable is set to 1 it indicates that the HDS is overloaded and something must be done to reduce the amount of data being recorded.
System. HDS.Started	Bit	R	Set to 1 when the HDS is running. Set to 0 when it is stopped.	This bit provides a way to detect that the HDS is stopped. Note that the HDS stops automatically if no more hard disk space is available to it.
System. RTDS.*OFS*.State	Bit	R	There is one bit for each configured OPC server. It is set to 1 if the connection is OK else 0.	*OFS* is the Alias of the OPC server.

Historical Data Server Overview

The Historical Data Server (HDS) collects and records real time data from the Real Time Data Server (RTDS). The recorded data is made available to the Trend and Log displays of the HMI via a connection based on OPC.

Historic data is recorded in one or more database files using standard database drivers. This version of the software has been qualified for use with the following database standards.

- Access 2000 (Using Jet 4.0)
- MSDE (Microsoft Database Engine)

Data is recorded in configurable tables of which there are two types.

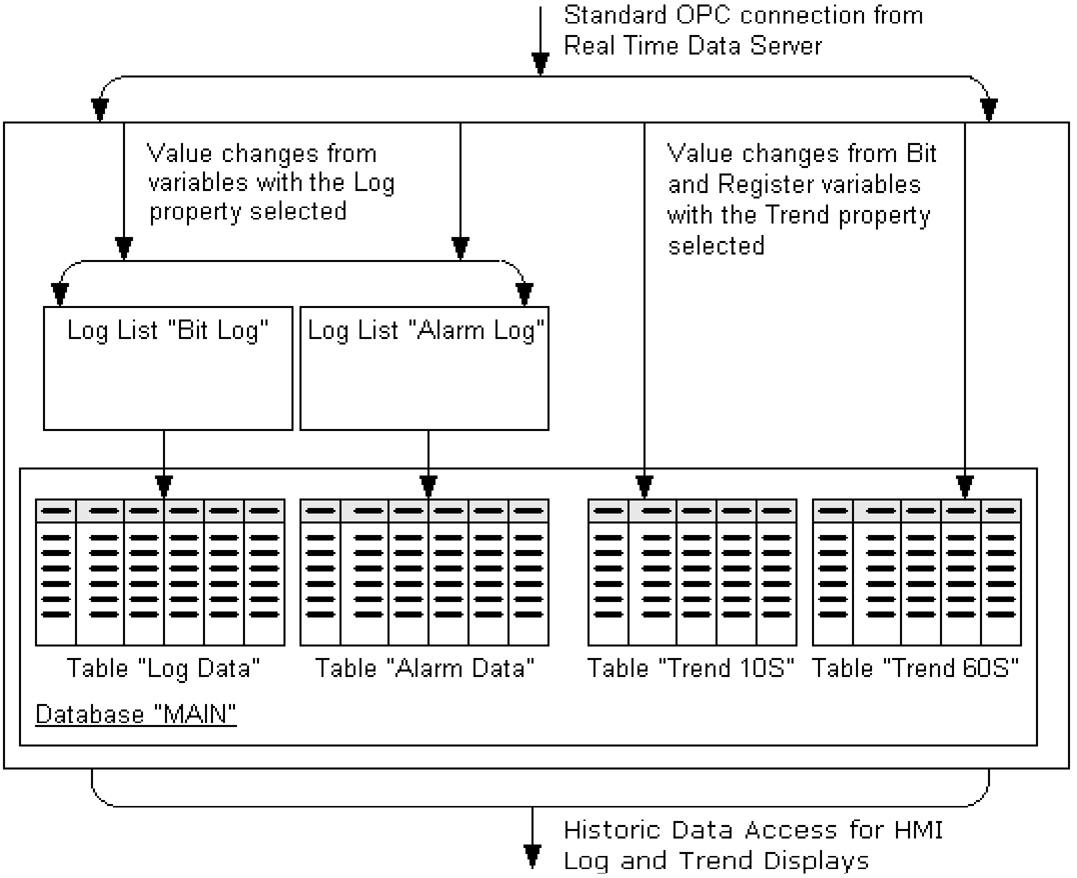
Trend Tables

Trend Tables are used for the periodic recording of register and bit variables. You can configure one or more Trend Table in each Database. Recorded data may be played back through a Trend Display. When a variable is selected for trend recording you select the database, the table within that database and the period at which the variable becomes eligible to be recorded.

Log Tables

Log Tables are used for the event recording of bit, alarm, register and text changes of state. You can configure one or more Log Tables in each Database. The configuration of a Log Table selects which properties are recorded from each eligible variable. The variables that are eligible are selected by the configuration of a given Log List. Recorded data may be played back through a Log Display.

An example of Log and Trend Table configuration is shown below.


Standard OPC connection from Real Time Data Server
Value changes from variables with the Log property selected
Value changes from Bit and Register variables with the Trend property selected
Log List "Bit Log"
Log List "Alarm Log"
Table "Log Data"
Table "Alarm Data"
Table "Trend 10S"
Table "Trend 60S"
Database "MAIN"
Historic Data Access for HMI Log and Trend Displays

For performance reasons, all data recorded in a database is first stored in a memory buffer. The buffer is written to the database every ten seconds, or when it contains fifty events, whichever is the sooner. For this reason, you may see a delay between an event occurring and it being visible in a Trend or Log Viewer.

Creating a Link to a Database File

An Example of Using the Customize Option | Deleting a Link to a Database File

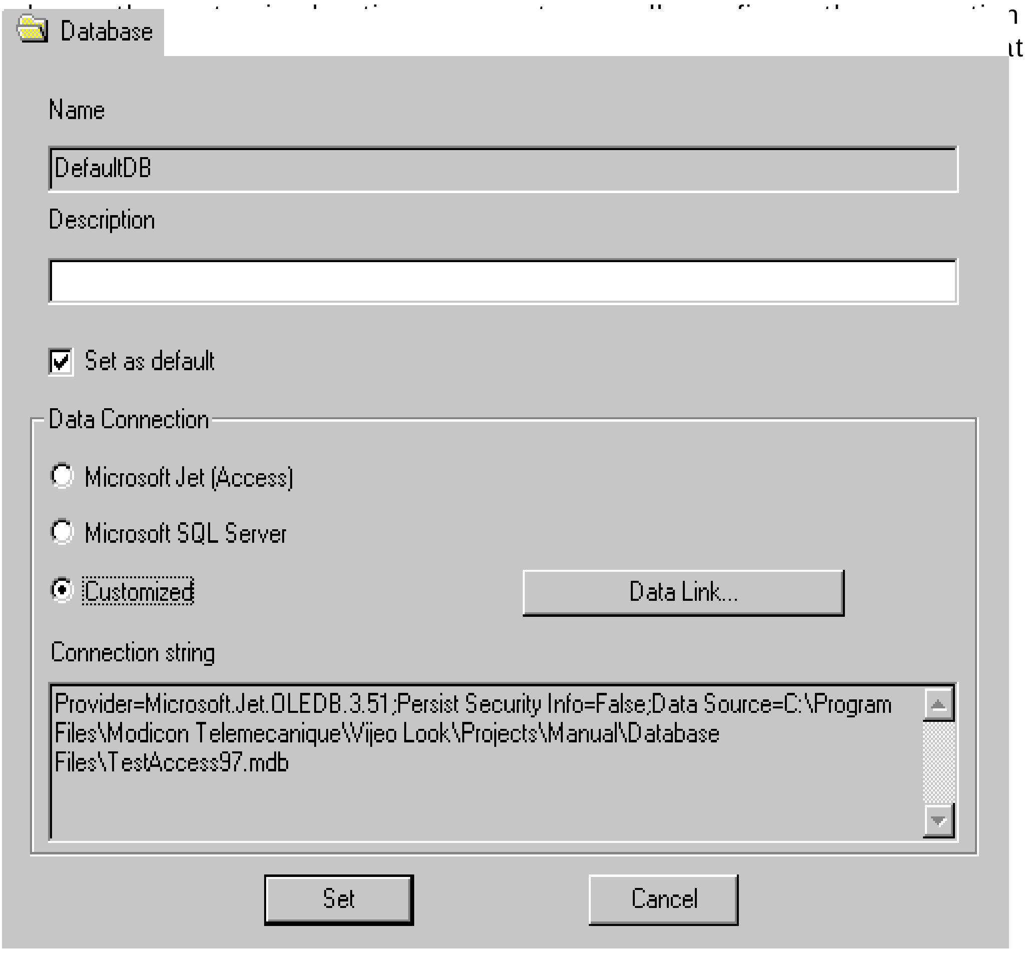
The configuration of the database link tells the software in which database file it will record historic data and how it will connect to it. You can choose from the following database and connection types.

- Microsoft Jet (Access).
- Microsoft SQL Server.
- Customized.

If you choose either *Microsoft Jet* (the default setting) or *Microsoft SQL Server*, no configuration is required. The database file is automatically created and the connection made to it.

The standard Microsoft Jet connection uses Jet 4.0, compatible with Access 2000. If you want to use a different version of Jet, for example 3.51 (compatible with Access 97), you must use the *Customize* option.

If you … n using the
Data … ated by 3rd
party



Name: DefaultDB

Description:

☑ Set as default

Data Connection

○ Microsoft Jet (Access)

○ Microsoft SQL Server

● Customized

Connection string: Provider=Microsoft.Jet.OLEDB.3.51;Persist Security Info=False;Data Source=C:\Program Files\Modicon Telemecanique\Vijeo Look\Projects\Manual\Database Files\TestAccess97.mdb

If you use the Microsoft SQL Server the option *Auto-start service when OS starts* in the SQL Service Manager dialog must be selected. This dialog may be opened from the Windows Start menu using the command **Programs.MSDE.Service Manager**.

Where are the database files located?

When using either the Microsoft Jet or Microsoft SQL Server option, the database files are created in a folder called Databases, located under the Vijeo Look root folder. (C:\Program Files\Modicon Telemecanique\Vijeo Look\Databases if you accepted the standard location when installing Vijeo Look.)

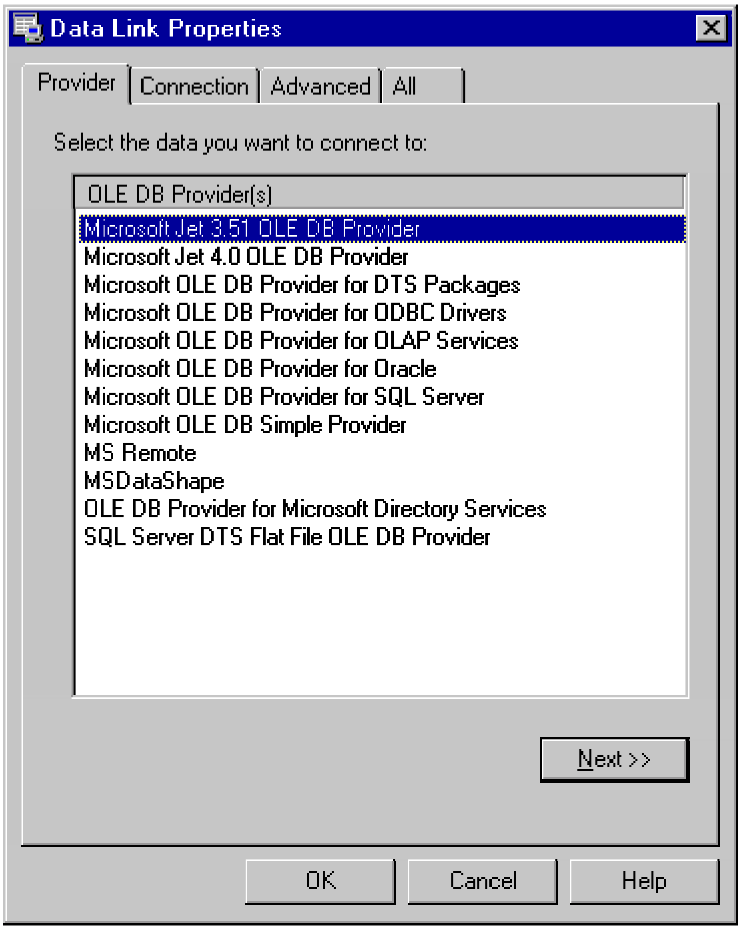
The name of the database file follows the form *ProjectName_DatabaseName* where *ProjectName* is the name of the project and *DataBaseName* is the name by which it is known in Vijeo Look. The file type depends on the type of database selected and will be .mdb for Microsoft Jet and .mdf for Microsoft SQL Server.

Be careful not to create two projects with identical names and identical database names as they will use the same database file.

An Example of Using the Customize Option

In this example the Customize option will be used to create a link to an Access 97 database using Jet 3.51. Please note that if you want to try this example, the Jet 3.51 driver is not installed as part of Vijeo Look, it must obtained from Microsoft (normally supplied as part of Access 97).

79. Open the Configuration Explorer.
80. Expand the tree in the left pane and right click on the Databases folder. From the pop-up menu select the *New* command.
81. Enter the name by which the database will be known. You can enter an optional comment about the database in the description field.
82. Click the *Data Link* button to open the Data Link Properties box. Select the *Provider* tab.
83. Select Microsoft Jet 3.51 OLE DB provider.
84. Select the *Next* button. You are taken to the Connection tab. Select or enter the name of the database file to use. The file must already exist, it cannot be created automatically.
85. Test the connection using the *Test Connection* button. If the connection fails then check your settings.
86. Click the *OK* button to return to the Database tab.
87. Select the *Set as default* option if this database will be used as the default.
88. Click the *Set* button to confirm the configuration.



Select the data you want to connect to:

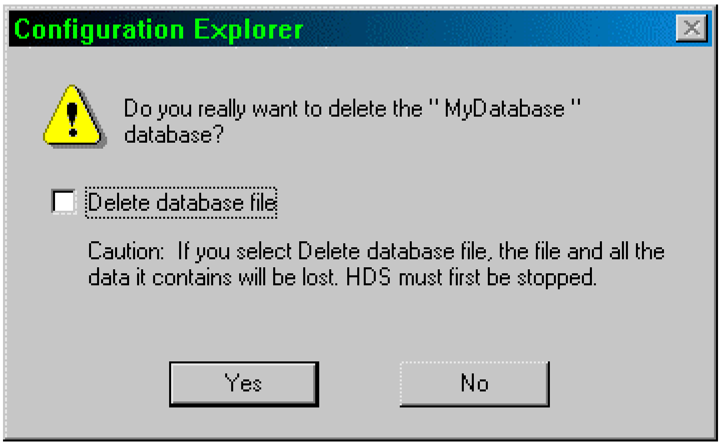
OLE DB Provider(s)

Microsoft Jet 3.51 OLE DB Provider
Microsoft Jet 4.0 OLE DB Provider
Microsoft OLE DB Provider for DTS Packages
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for OLAP Services
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MS Remote
MSDataShape
OLE DB Provider for Microsoft Directory Services
SQL Server DTS Flat File OLE DB Provider

Once a link to a database has been created it cannot be modified. It must be deleted, along with the database, and a new link and database created.

Deleting a Link to a Database File

You can delete a link to a database file by right clicking on its name in the left pane of the Configuration Explorer and selecting the *Delete* command. A dialog will be displayed giving you the option to also delete the database file itself.



Do you really want to delete the " MyDatabase " database?

□ Delete database file

Caution: If you select Delete database file, the file and all the data it contains will be lost. HDS must first be stopped.

Clicking *Yes* will complete the operation, clicking *No* will close the dialog without any action.

Selecting *Destroy permanently* will delete the database file and all the data it contains. For this option to work the HDS must not be running.

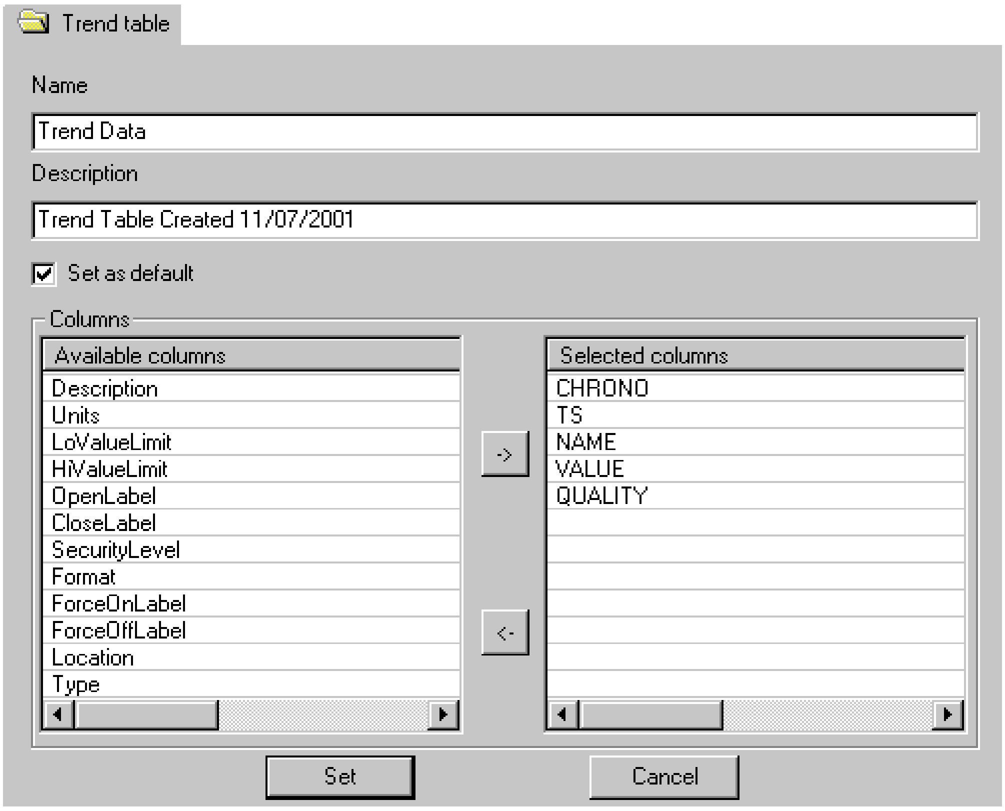
Configuring a Trend or Log Table

What can you record?

You can configure one or more Trend and Log Tables for each configured database. Each table has a number of columns in which data is recorded. Some columns are mandatory and are pre-selected, whilst others are optional. To differentiate them, the mandatory columns have names that are capitalised.

How to create a Table

89. Open the Configuration Explorer.
90. Expand the tree in the left pane and select a Database folder.
91. Expand the Database folder and right click on either the Trend Table or Log Table folder.
92. Enter the name by which the Trend Table will be known. You can enter an optional comment about the Trend Table in the description field.
93. Select the columns corresponding to the data you want to record. If you do not intend to use the recorded data in an external program then you only need the mandatory columns.
94. Select the Set as default option if this Trend Table will be used as the default.
95. Click the Set button to confirm the configuration.



Trend table

Name

Trend Data

Description

Trend Table Created 11/07/2001

☑ Set as default

Columns

Available columns

Description
Units
LoValueLimit
HiValueLimit
OpenLabel
CloseLabel
SecurityLevel
Format
ForceOnLabel
ForceOffLabel
Location
Type

Selected columns

CHRONO
TS
NAME
VALUE
QUALITY

You must restart the Historical Data Server before any table modifications are taken into account.

Modifying a table

Once a table has been created it you can add columns, but not remove them. The only way to remove columns is to delete the table and re-create it with the new configuration. Any existing data will be lost.

What can you Record?

The mandatory columns record data that is essential for the playback of data through the HMI. Other columns can be added if you require the information to be available to other programs such as Access. The optional columns that are available depends on what has been selected and configured in the Variable Properties and Extended Properties tabs of the General Properties folder.

The following table describes the mandatory properties.

Column	Format	Description
CHRONO	Numeric	The timestamp of the record in numeric format.
TS	Date/Time	The timestamp of the record in Date/Time format.
NAME	Text	The name of the variable.
VALUE	Numeric	The value of the variable if it is a Register or Bit.
VALUET*	Text	The value of the variable if it is Text.
QUALITY	Numeric	The quality (from the OPC server) of the variable.
LOGLIST*	Text	The name of the Log List which was the source of the data.
ASSOCLABEL*	Text	The name of the associated label.
EVTNUMBER*	Numeric	A number representing the type of event.
EVTTITLE*	Text	The event description from the associated label.

* Log Table only.

The following table lists the optional properties. A description of the properties may be found in the Variable Properties Tab topic.

Column	Format
Description	Text
Units	Text
LoValueLimit	Numeric
HiValueLimit	Numeric
OpenLabel	Text
CloseLabel	Text
SecurityLevel	Numeric
Format	Text
ForceOnLabel	Text
ForceOffLabel	Text
AlarmLevel	Numeric

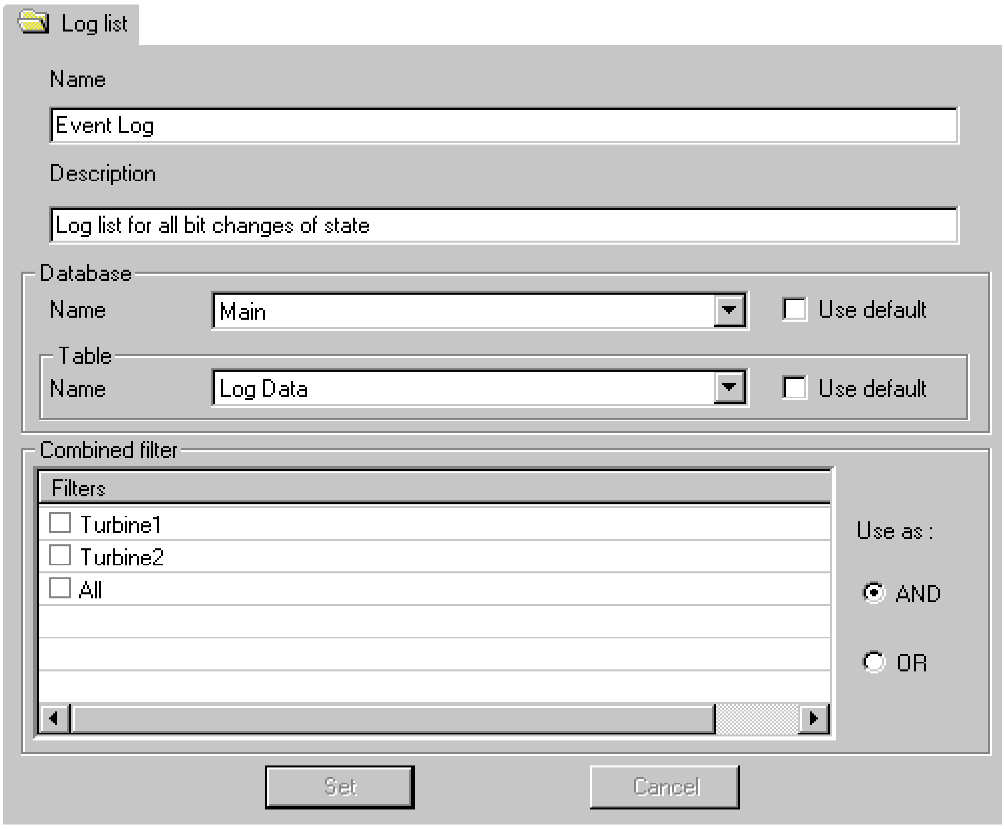
Configuring a Log List

The Log Lists are used to filter Log data before it is recorded in a Log Table. Each Log List is attached to a single Database Table. A Database Table may have one or more Log Lists attached to it.

In a Log List you can filter data by a variable's properties (for example its name) and by the change of state (for example changing from 1 to 0).

How to create a Log List

96. Open the Configuration Explorer.
97. Expand the tree in the left pane and select the Log Lists folder.
98. Right click on the folder and from the pop-up menu select the New command.
99. Enter a name by which the Log List will be known. You can enter an optional comment in the description field.
100. Select the Database and Table to which the Log List will be attached. Alternatively you can use the default Database and Table by selecting the Use default property.
101. Click the *Set* button to confirm the configuration



Log list

Name: Event Log

Description: Log list for all bit changes of state

Database
Name: Main ☐ Use default

Table
Name: Log Data ☐ Use default

Combined filter

Filters
☐ Turbine1
☐ Turbine2
☐ All

Use as : ● AND ○ OR

Using a filter to select the data that a Log List passes to the Database

The Filters select data according to a variable's properties and extended properties. The Filters themselves are created and configured from the General Properties folder of the Configuration Explorer. In the Log List you select the Filters you want using the tick box adjacent to the filter name in the Combined Filter list. When you select more than one filter they are combined using either an AND or OR criteria as selected.

Using change of state to select the data that a Log List passes to the Database

You can also select data according to its change of state. For example, for a given Log List, you can choose to select only bit transitions from 0 to 1. State changes are selected from the State change tab. The following are available.

- Alarm off. The change of an alarm to the off state.
- Alarm off, Not Ack. The change of an alarm to the off, not acknowledged state.
- Alarm on, Ack. The change of an alarm to the on, acknowledged state.
- Alarm on, Not Ack. The change of an alarm to the on, not acknowledged state.
- Invalid alarm. The change of an alarm to invalid (No value, normally indicating a communication failure)
- Alarm operator masked. The change of an alarm when it has been masked (inhibited) by the User.
- Invalid value. The change of a bit, register or text variable to invalid. (No value, normally indicating a communication failure.)
- Change to 0. The change of a bit from any state to 0.
- Change to 1. The change of a bit from any state to 1.
- Value change. The change in value of either a register or text variable.
- Command to 1. The forcing of a bit using a control zone animation from any state to 1.
- Command to 0. The forcing of a bit using a control zone animation from any state to 0.

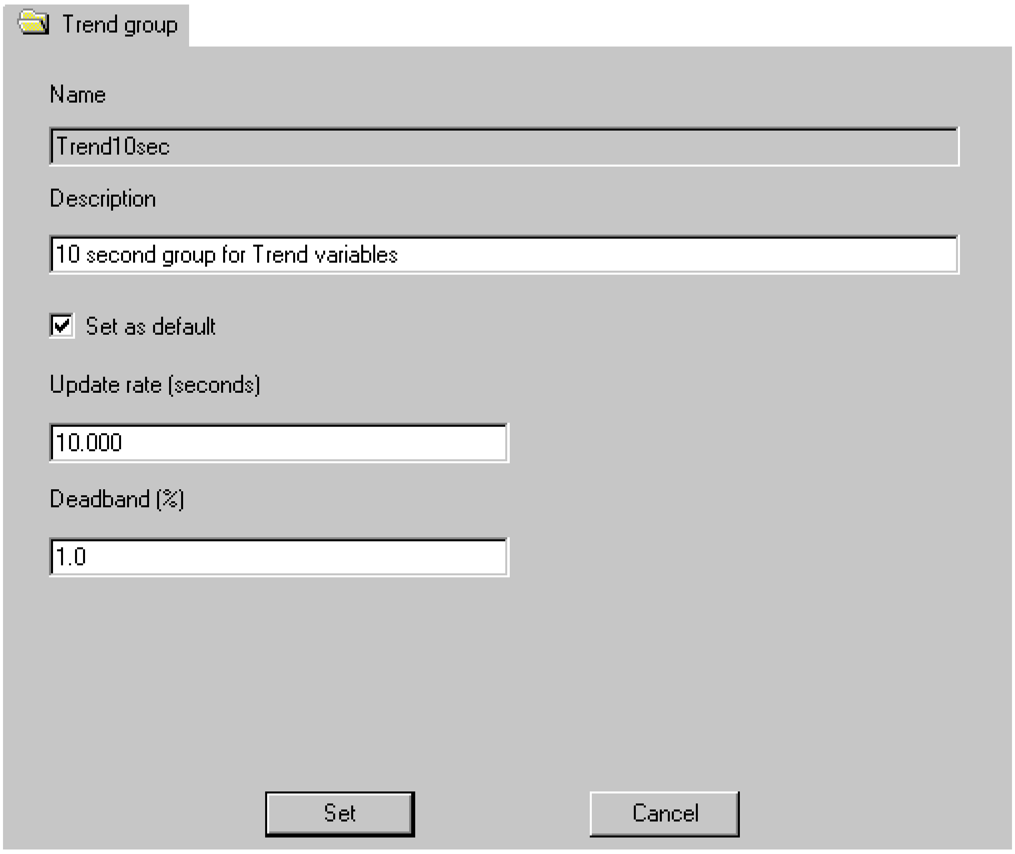
Configuring a Trend Group

A Trend Group is used to determine when a Bit or Register variable that has been assigned to it, is recorded. Two criteria or used.

- A period. A variable only becomes eligible to be recorded if its value has changed in the period. The period is entered in seconds. The minimum value is 0.2. It will only be recorded once in that period.
- A deadband. A variable's value must change by more than the deadband before it is recognised as changing. The deadband is entered as a percentage of the variable range. If the deadband is zero then it is not used. The deadband is only applicable to Register variables.

How to configure a Trend group

1. Open the Configuration Explorer.
2. Expand the tree in the left pane and select the Trend Group folder.
3. Right click on the folder and from the pop-up menu select the New command.
4. Enter a name by which the Trend group will be known. You can enter an optional comment in the description field.
5. Enter the *Update rate* for the group in seconds.
6. Enter the *Deadband* for the group as a percentage.
7. Select the *Set as default* property if the Trend group is to be used as the default.
8. Click the *Set* button to confirm the configuration.



Name: Trend10sec

Description: 10 second group for Trend variables

☑ Set as default

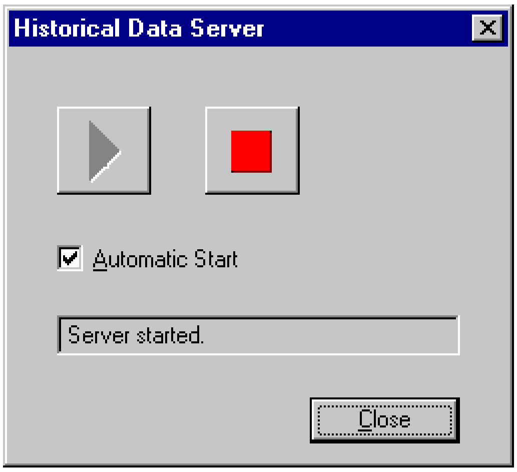
Update rate (seconds): 10.000

Deadband (%): 1.0

Whilst the Deadband is optional it is recommended that it is always used. When measuring analogue values digitally there is always a small variation in the measured value, even if the process value is steady, due to the way in which analogue to digital converters work.

Manually Stopping the HDS

You can manually stop and start the HDS from a dialog displayed using the menu command **Tools.Application.Historic Data Server**.



Server started.

The button with the square icon stops the HDS, the button with the triangle icon starts it. While the HDS is stopped the start icon is green; while it is running the stop icon is red.

You can also choose to not automatically start the HDS when Vijeo Look start by de-selecting the *Automatic Start* property.

Stopping the HDS is not instantaneous. It may take from several seconds to a number of minutes from clicking the stop button before it stops.

Using the Diagnostic Buffer

How to Configure a Diagnostic Buffer | Using an Alarm Viewer to Display Messages from the Diagnostic Buffer | How to Configure a Filter for the Diagnostic Buffer

The Diagnostic Buffer is only available when using the OFS server. The Diagnostic tab provides a way of linking the PLC Diagnostic Buffer with an Alarm Viewer. This feature is only available with PLCs supporting diagnostic DFBs, currently the Premium TSX57, PCX57 and PMX57. Furthermore the PLC must have a minimum software revision - see the PL7 documentation for more information. The Diagnostic Buffer provides the following information for those elements it is monitoring.

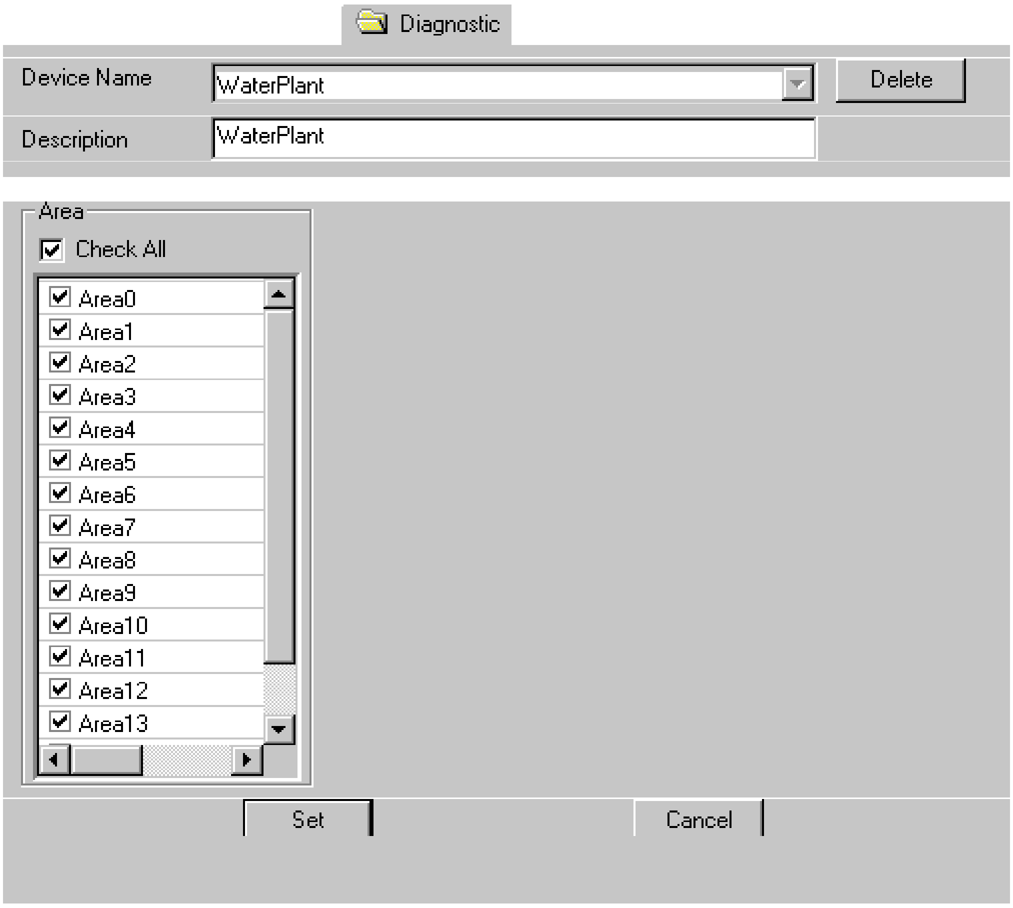
- The type of error
- The start date and time
- The end data and time
- The area (a number between 0 and 15) in the case when several modules are declared in the same PLC.
- Alarm comments.

Diagnostic Buffer messages from the PLC are retrieved by the RTDS and sent to the Alarm Viewer via the Alarm Server. Alarm acknowledgements from the User are routed back to the PLC.

How to Configure a Diagnostic Buffer

1. Open the Configuration Explorer and select the OFS Server folder. (**Data Server.OPC Servers.OFS**)

2. Select the Diagnostic tab. Select the *Device Name* from the list box. This corresponds to the Alias name of the PLC as entered in the OFS configuration tool.
3. Enter a Description. The text here is used in Filter configuration to select which Diagnostic Buffer messages are presented to a particular Alarm Viewer and it is also displayed in the Description column of the Alarm Viewer.
4. Select which of the 16 areas are to be taken into account.
5. Confirm the configuration using the *Set* button.



Device Name: WaterPlant

Description: WaterPlant

Area

☑ Check All

☑ Area0
☑ Area1
☑ Area2
☑ Area3
☑ Area4
☑ Area5
☑ Area6
☑ Area7
☑ Area8
☑ Area9
☑ Area10
☑ Area11
☑ Area12
☑ Area13

Using an Alarm Viewer to Display Messages from the Diagnostic Buffer

The Alarm Viewer is used to display and acknowledge messages from a Diagnostic Buffer. See the Alarm Viewer topic for detailed configuration information. No special configuration is required but the following configuration options are worthy of special note.

The Format Tab

The Format tab contains 4 columns that are specific to Diagnostic Buffer Messages.

- *AreaNumber* - The area associated with the message.
- *DiagMessage* - The alarm message.
- *ProgAddress* - The PLC program address.
- *StatusFile* - The information file reference.

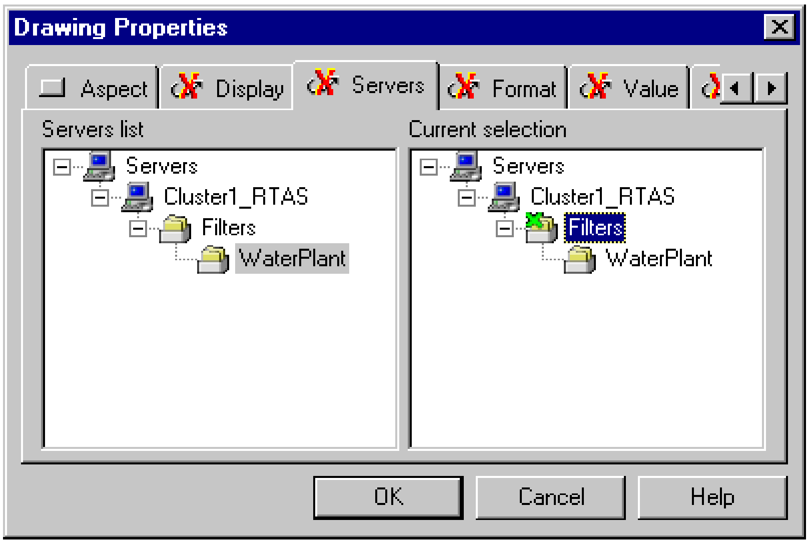
In addition the *Description* column will display the device and instance name of the DFB.

The Servers tab

If, in a particular Alarm Viewer, you want to display messages from the Diagnostic Buffer only, you must create and select a Filter using the condition Description = "the description field of the Diagnostic Buffer".

The following explanation assumes an Alarm Viewer has already been pasted into a mimic and the mimic is in design mode.

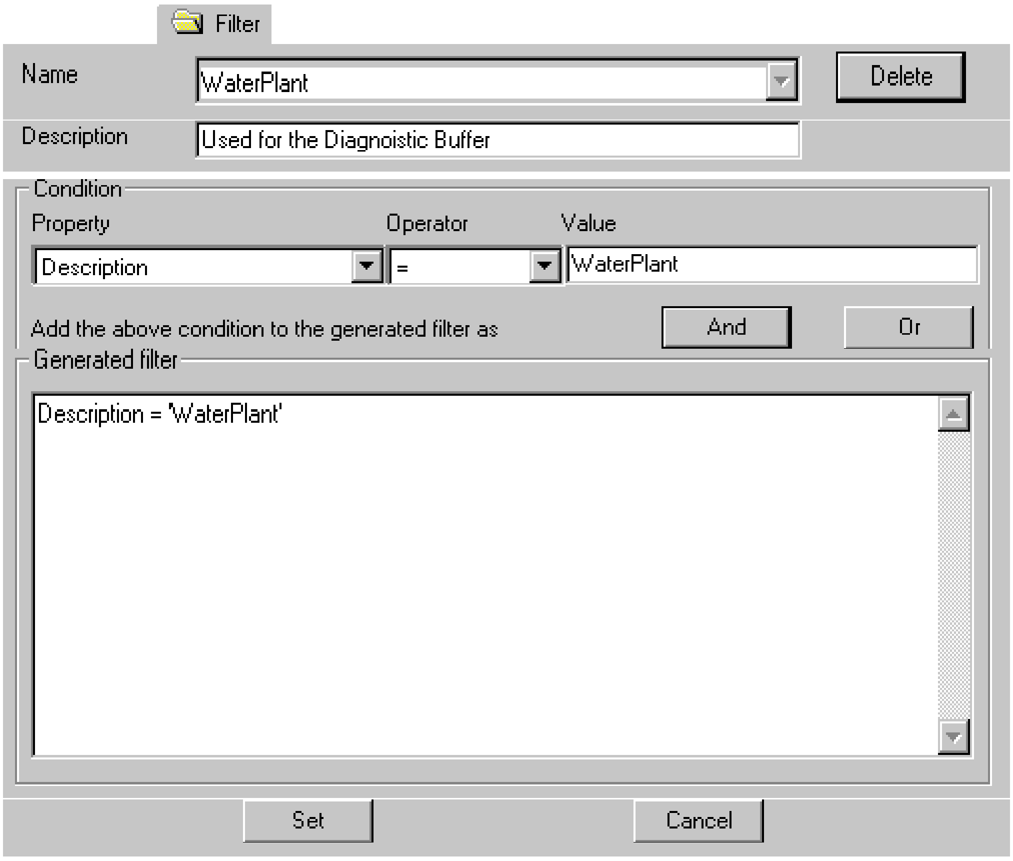
6. Double click on the Alarm Viewer to display its properties. Select the *Servers* tab.
7. Expand the tree control in the left pane to display the Filters folder.
8. Drag a previously configured filter to the right pane and drop it under the *Filters* folder
9. Click the OK button to confirm the configuration.



How to Configure a Filter for the Diagnostic Buffer

1. Open the Configuration Explorer and select the General Parameters folder.

2. Select the Filters tab. Enter a reference name for the Filter in the *Name* field. This can be anything you like but it is suggested that you use the same name as in the Diagnostic tab.
3. Enter a description of the use of the Filter in the *Description* field. This serves no other purpose other than as a reminder of what the Filter is used for.
4. In the *Property* list box select Description.
5. In the *Operator* list box select =.
6. In the *Value* field enter the description as used in the Diagnostic tab, and click the *And* button
7. Confirm the configuration using the *Set* button.


Name: WaterPlant

Description: Used for the Diagnoistic Buffer

Condition

Property: Description

Operator: =

Value: WaterPlant

Add the above condition to the generated filter as

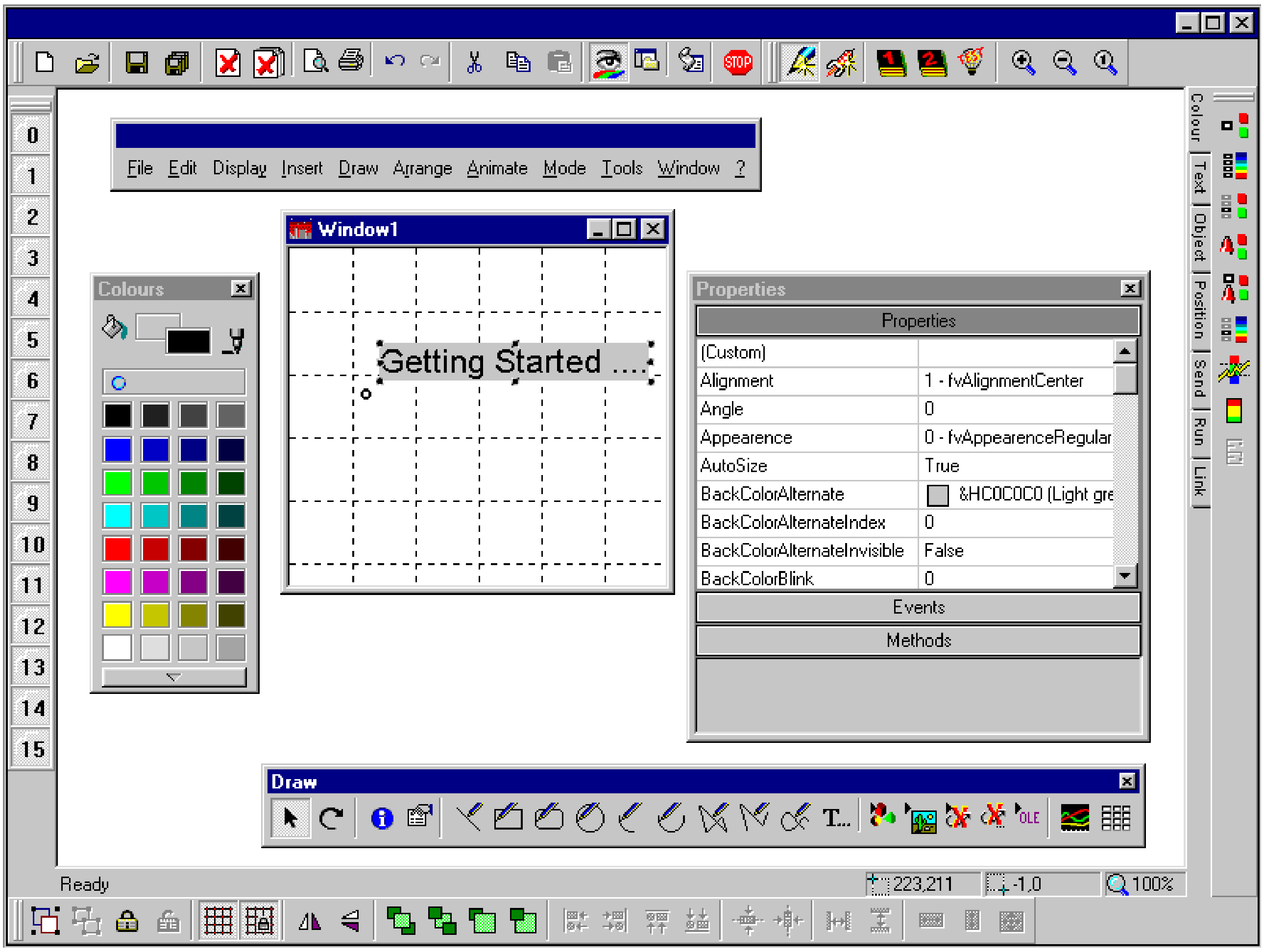
Generated filter

Description = 'WaterPlant'

The Workspace

Changing the Workspace Properties

The Workspace is the name given to the area that FrontVue occupies on the screen of your PC. The workspace contains all the tools necessary to develop HMI applications and is the container for the windows that form the User Interface at runtime. Within the workspace you will find the following features:



Properties	
(Custom)	
Alignment	1 - fvAlignmentCenter
Angle	0
Appearence	0 - fvAppearenceRegular
AutoSize	True
BackColorAlternate	□ &HC0C0C0 (Light gre
BackColorAlternateIndex	0
BackColorAlternateInvisible	False
BackColorBlink	0

The Menu Bar

The Menu Bar provides access, using either the keyboard or the mouse, to all the development tools. The menu bar can be dragged with the mouse and docked to any outside edge of the workspace, or it can be left floating anywhere on the screen. The menu bar may be opened and closed from a pop-up menu displayed by clicking with the right mouse button anywhere within the workspace background. A tick indicates that the menu bar is visible.

Toolbars

The toolbars give you quick access, using the mouse, to the most commonly used tools for window development and run-time operation. As with the menu bar the toolbars may be floating or docked. You can select which toolbars are open from a pop-up menu displayed by right-clicking anywhere on the menu bar. A tick indicates that a toolbar is opened.

The Colour Palette

The colour palette is used to define and select colours for drawing and animation. It may be opened and closed using the Display.Colour Palette command.

The Properties List

The properties list displays a Visual Basic style list of properties, methods and events for windows, native drawing elements and activeX controls. It may be opened and closed using the Display.Properties List command.

Scroll Bars

The scroll bars are used to pan around a mimic window when it is larger than the Workspace.

Status Line

The status bar is located at the bottom of the workspace, above any toolbars you may have docked there. It provides information on the current state of the workspace including the current cursor position and zoom level of the active window.

Changing the Workspace

The workspace appearance and behaviour is changed using the Display.Workspace Properties command. The following properties may be changed.

- The size of the workspace and its location on the screen.
- The workspace colour.
- The appearance of the title bar and it's contents.
- The tools that are available from the title bar.

MS IntelliMouse Support

The HMI has built in support for the Microsoft Intellimouse. In addition to the normal two buttons the IntelliMouse has a centrally mounted wheel that can be used to zoom, scroll and pan around the HMI Workspace and windows.

Using the Wheel to Zoom

To zoom in:

1. Select a window by clicking anywhere in it.
2. Click and hold the CTRL key, and then roll the wheel forward (toward your monitor).

To zoom out:

3. Select a window by clicking anywhere in it.
4. Click and hold the CTRL key, and then roll the wheel back (toward you).

Using the Wheel to Scroll

You can use the wheel to scroll vertically whenever scrollbars are displayed.

Scrolling a mimic within a window:

1. Select a window by clicking anywhere in it.
2. Roll the wheel to scroll the mimic up or down within the window.

Scrolling a window within the workspace:

1. Click in the workspace (outside of any windows).
2. Roll the wheel to scroll the mimic up or down within the window.

Using the Wheel to Pan

You can use the pointer to pan (scroll vertically and horizontally) in either the workspace or a window when scrollbars are displayed.

Panning a mimic within a window:

1. Click the wheel button within the window and observe the position of the origin mark that appears.
2. Move the pointer in the direction that you want to pan. The further you move the pointer away from the origin mark, the faster the mimic pans.
3. Click the wheel button again to stop panning.

Panning the workspace:

1. Click the wheel button anywhere within the workspace and observe the position of the origin mark that appears.
2. Move the pointer in the direction that you want to pan. The further you move the pointer away from the origin mark, the faster the workspace pans.
3. Click the wheel button again to stop panning.

Panning a zoomed mimic when the workspace also has scrollbars:

1. Hold down the Ctrl key and click the wheel button within the window and observe the position of the origin mark that appears.
2. Move the pointer in the direction that you want to pan. The further you move the pointer away from the origin mark, the faster the mimic pans.
3. Click the wheel button again to stop panning.

How OPC Variables are Named

The OPC standard allows for a flexible approach to variable naming with either a flat or hierarchical naming scheme. The variable names are determined entirely by the OPC server. The HMI requests a list of names from the server, which is then displayed in the Variable Browser.

Most OPC servers use a hierarchical naming scheme of one form or another. Some servers allow variables to be named in a way that represents the outside world, whilst others use names that represent locations of a data table in a PLC or other piece of equipment. Each variable name has one or more elements (also known as branches and leaves) separated by decimal points. The following are all valid variable names.

```
Process.Engine1.Compressor1.B1
Process.Engine1.Compressor1.A1
CARD1.DATABASE.B1
Node1.DataTable3_256
Simulation.Square Waves.Int4
B1
```

Within the HMI variable names are preceded by the name of the cluster followed by a colon. The following are all valid variable names within the HMI.

```
Cluster1:Process.Engine1.Compressor1.B1
Cluster1:Process.Engine1.Compressor1.A1
Applicom:CARD1.DATABASE.B1
Data:Node1.DataTable3_256
Simu:Simulation.Square Waves.Int4
C1:B1
```

Variable Value, Quality and Timestamp (VTQ)

For each variable (OPC item) used in an animation the HMI expects the server to provide the item Value, Timestamp and Quality (known as VTQ). Normally only the item value is displayed but, by the use of two special suffixes, the quality or timestamp may be displayed.

To display the timestamp the suffix #_T is added to the item name.

To display the quality the suffix #_Q is added to the item name.

The following example shows the use of this with variable names within the HMI.

```
Cluster1:Process.Engine1.Compressor1.B1#_T
Cluster1:Process.Engine1.Compressor1.A1#_Q
```

Other Variable Properties

In addition to the Value, Quality and Timestamp, certain animation types expect a linked variable to support additional properties. For example, a register variable, when linked to a bargraph animation, would require a minimum and maximum value. Most of these additional properties are optional with a default or user configured value being used if they are not available.

If an OPC variable supports additional properties that are accessible by the HMI they will be displayed in the Variable Browser and may also be used in an animation in their own right. For example you could display the minimum and maximum values of a register.

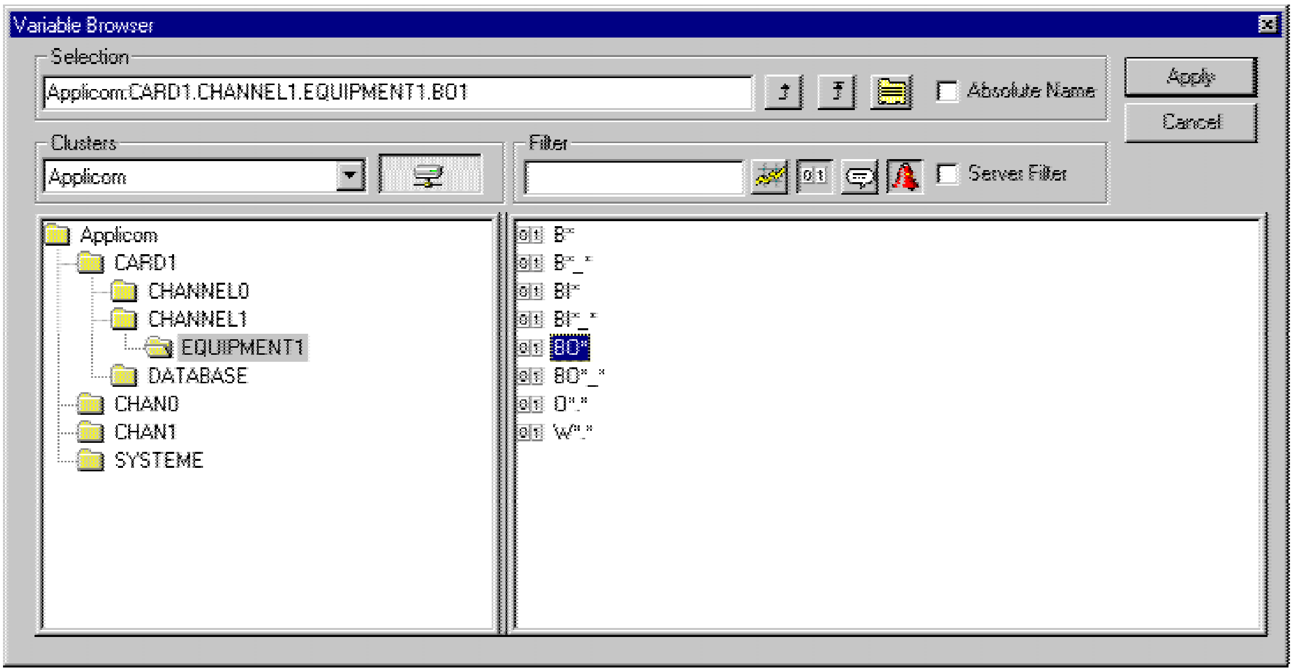
The following shows an example of naming of additional properties.

```
Cluster1:Process.Engine1.Compressor1.M1
Cluster1:Process.Engine1.Compressor1.M1.Min
Cluster1:Process.Engine1.Compressor1.M1.Max
Cluster1:Process.Engine1.Compressor1.M1.UnitText
```

Using the OPC Variable Browser

Filtering the Variable List | Using the Variable Browser to Select a Variable for an Animation or Expression | Displaying More Information About a Variable | Using the Variable Browser when the OPC Server Supports Additional Properties

The Variable Browser displays a list of all variables available in a selected OPC server. The display is similar to the Windows NT Explorer with a tree control in one pane and a detail display in the other. Instead of representing folders and files the icons represent the hierarchical structure of variables within of the server. The information displayed may be filtered by data type and a lexical filter field.



Selection: Applicom:CARD1.CHANNEL1.EQUIPMENT1.B01

Clusters: Applicom

Filtering the Variable List

The variable list that appears in the right hand pane may be filtered by type (using the buttons corresponding to bit, register, text or alarm) and by using a filter string.

If the *Server Filter* property is selected, the filter criteria is sent to the OPC server. In this case the list of variables is filtered before it is presented to the HMI greatly improving the speed of browsing. Not all OPC servers support this feature and so you must check with the manufacturer's literature before using it.

The filter string allows the use of wildcard characters, character lists, or character ranges, in any combination, to match variable names. The following table shows the characters allowed in the filter and what they match.

Characters in filter	Matches in variable name
?	Any single character.
*	Zero or more characters.
#	Any single digit (0-9).
[charlist]	Any single character in charlist.
[!charlist]	Any single character not in charlist.

A group of one or more characters (charlist) enclosed in brackets ([ ]) can be used to match any single character in string and can include almost any character code, including digits.

By using a hyphen (-) to separate the upper and lower bounds of the range, charlist can specify a range of characters. For example, [A-Z] results in a match if the corresponding character position in string contains any uppercase letters in the range A-Z.

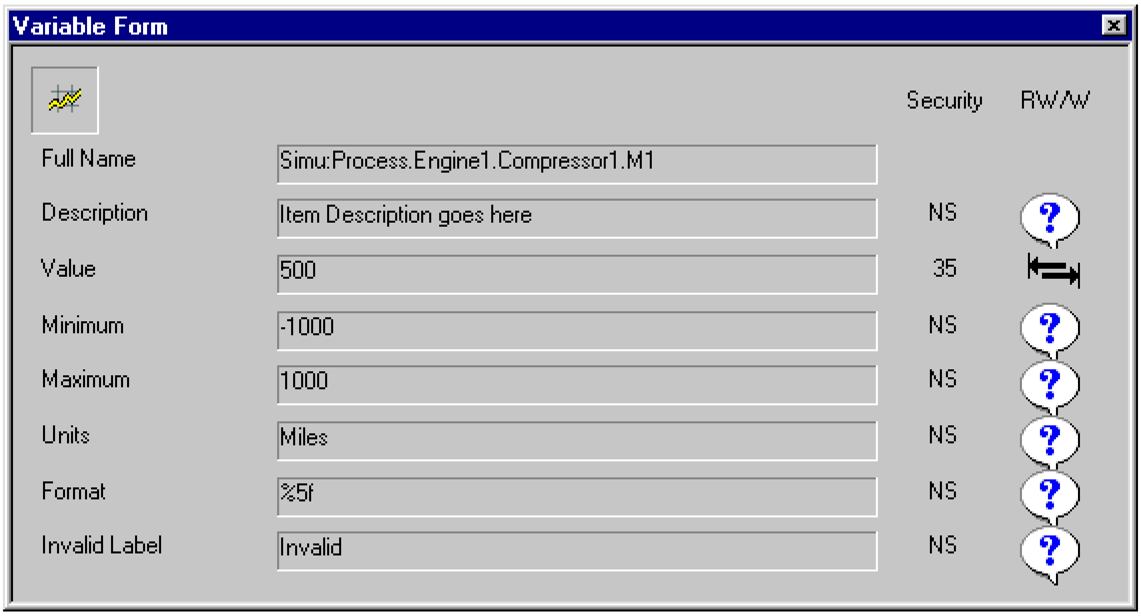
Using the Variable Browser to Select a Variable for an Animation or Expression

1. Open the Variable Browser. (Click on the button in an animation tab or the Browser button in the Expression Manager.)
2. Select a cluster using the cluster list box.
3. Request a list of variables from the server by clicking the button. This may take a few seconds depending on the type of server.
4. Expand the tree display of the server's variables and select the variable you require by clicking on it. The variable name appears in the selection field.
5. Check that the variable name is correct. Some OPC browsers provide only partial names and you may need to complete the name manually by typing in the selection field.
6. Click the Apply button.
7. Close the Variable Browser.

You must select a variable of the correct type. If you select a variable of the wrong type (for example a text variable for an animation that expects a bit) a warning message will be generated and the link between the animation and OPC server will not work.

Displaying More Information About a Variable

You can display more information about a variable by clicking on the button once you have selected a variable name. What is then displayed will depend on the structure of the OPC server.

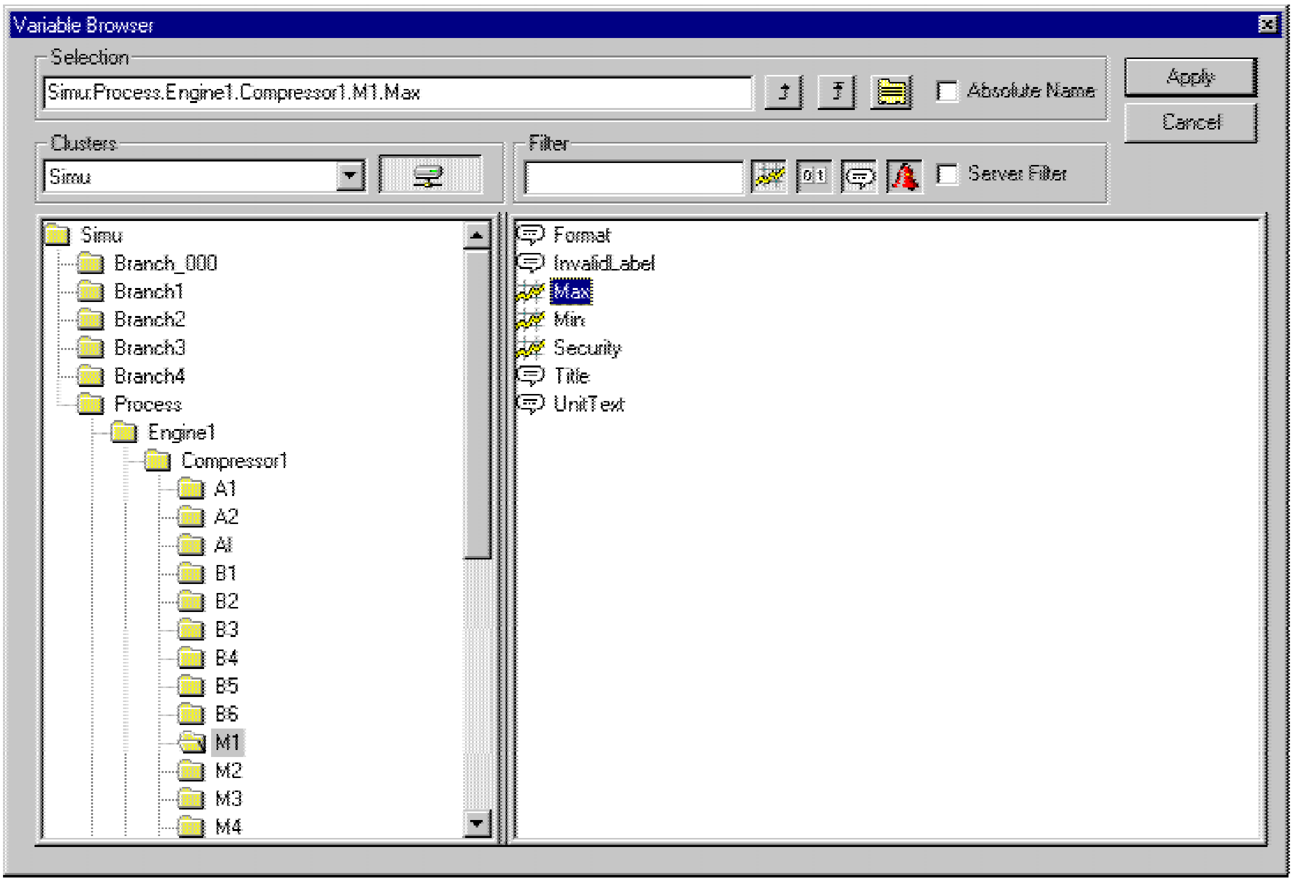


Variable Form

		Security	RW/W
Full Name	Simu:Process.Engine1.Compressor1.M1		
Description	Item Description goes here	NS	?
Value	500	35	⇆
Minimum	-1000	NS	?
Maximum	1000	NS	?
Units	Miles	NS	?
Format	%5f	NS	?
Invalid Label	Invalid	NS	?

Using the Variable Browser when the OPC Server Supports Additional Properties

If an OPC Server supports additional properties for the variables you will be able to see both the variable name and its additional properties in the variable browser. The OPC variable, for example Simu:Process.Engine1.Compressor1.M1 appears as both a branch (left hand pane only) and a complete variable (left hand plus right hand pane). You can select either the variable or the variable plus one of its properties.



Variable Browser

Selection

Simu.Process.Engine1.Compressor1.M1.Max

☐ Absolute Name

Clusters

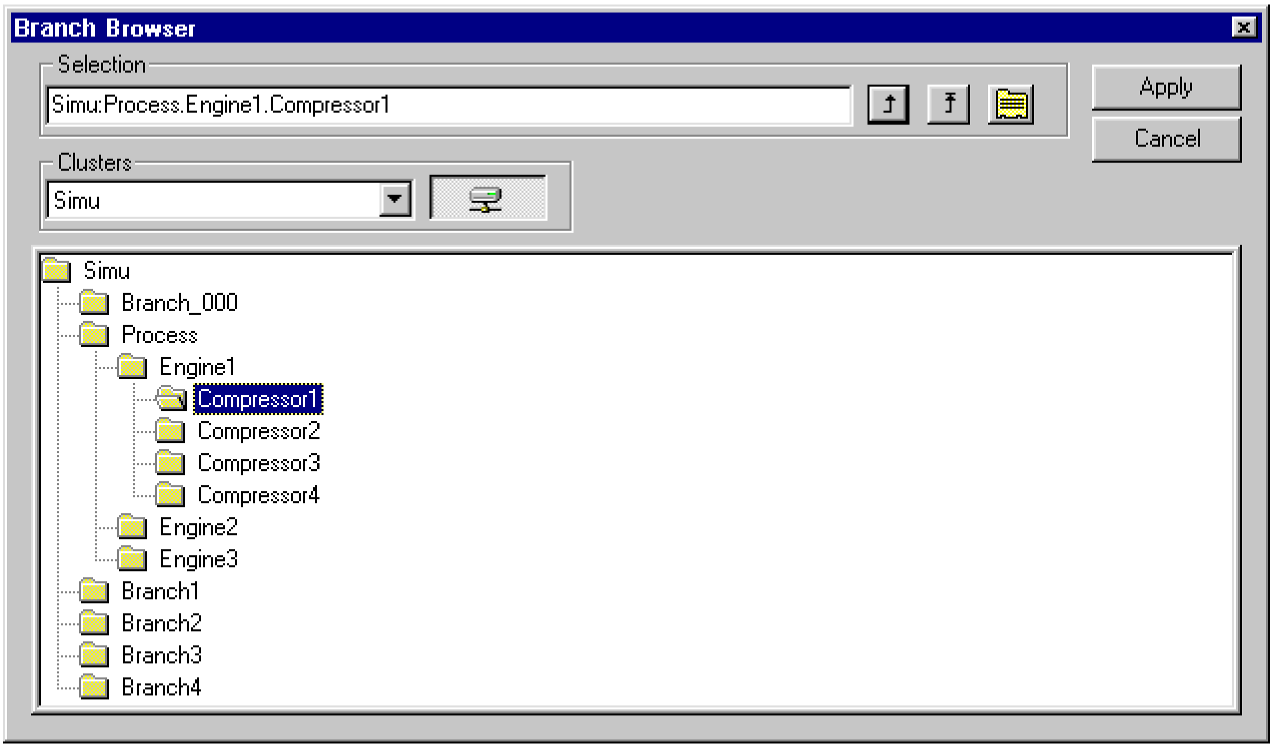
Simu

Filter

☐ Server Filter

Using the Branch Browser

The Branch Browser is used in a similar way to the Variable Browser except that it is used to select a branch rather than a complete variable name. The Branch Browser may be displayed whenever it is necessary to select a branch in any of the configuration dialog boxes.



Selection: Simu:Process.Engine1.Compressor1

Clusters: Simu

- Simu
  - Branch_000
  - Process
    - Engine1
      - Compressor1
      - Compressor2
      - Compressor3
      - Compressor4
    - Engine2
    - Engine3
  - Branch1
  - Branch2
  - Branch3
  - Branch4

What is OPC Quality?

The Standard Quality Flags

An OPC server provides a minimum of three properties for each of its variables, known as Value, Quality and Timestamp.

- The Value is the real-time value of the variable, for example the level of a tank.
- The timestamp is the time at which the value last changed.
- The Quality provides information about the reliability and status of the value.

The Quality is provided as a 16 bit word. The higher 8 bits are vendor specific and the lower 8 bits always follow a standard format.

The extent to which the Quality property is supported depends on the server you are using. For further information on the server your are using see the vendors documentation.

The Standard Quality Flags

The lowest 8 bits of the Quality flags are defined in the form of three bit fields; Quality, Sub-status and Limit status arranged in the form of QQSSSSLL as follows:

The quality (QQ) field

The quality field provides the most basic information about the quality of the variable.

QQ	Bit Value	Defines	Description
0	00SSSSLL	Bad	Value is not useful for reasons indicated by the Sub-status.
1	01SSSSLL	Uncertain	The quality of the value is uncertain for reasons indicated by the Sub-status.
2	10SSSSLL	N/A	Not used.
3	11SSSSLL	Good	The Quality of the value is Good.

If the OPC server does not provide Quality information then this value is always set to 3.

The Sub-status (SSSS) field

The Sub-status field provides information specific to the basic Quality. For example if the Quality was uncertain the status may indicate that the value is outside of its configured range. If the server does not provide Sub-status information then this field is always set to 0.

The Limit field

The Limit field provides an indication of the value of the variable with respect to its limits and is valid regardless of the Quality and Sub-status. If the server does not provide Limit information then this value is always set to 0.

What are the Quality Sequences Used For?

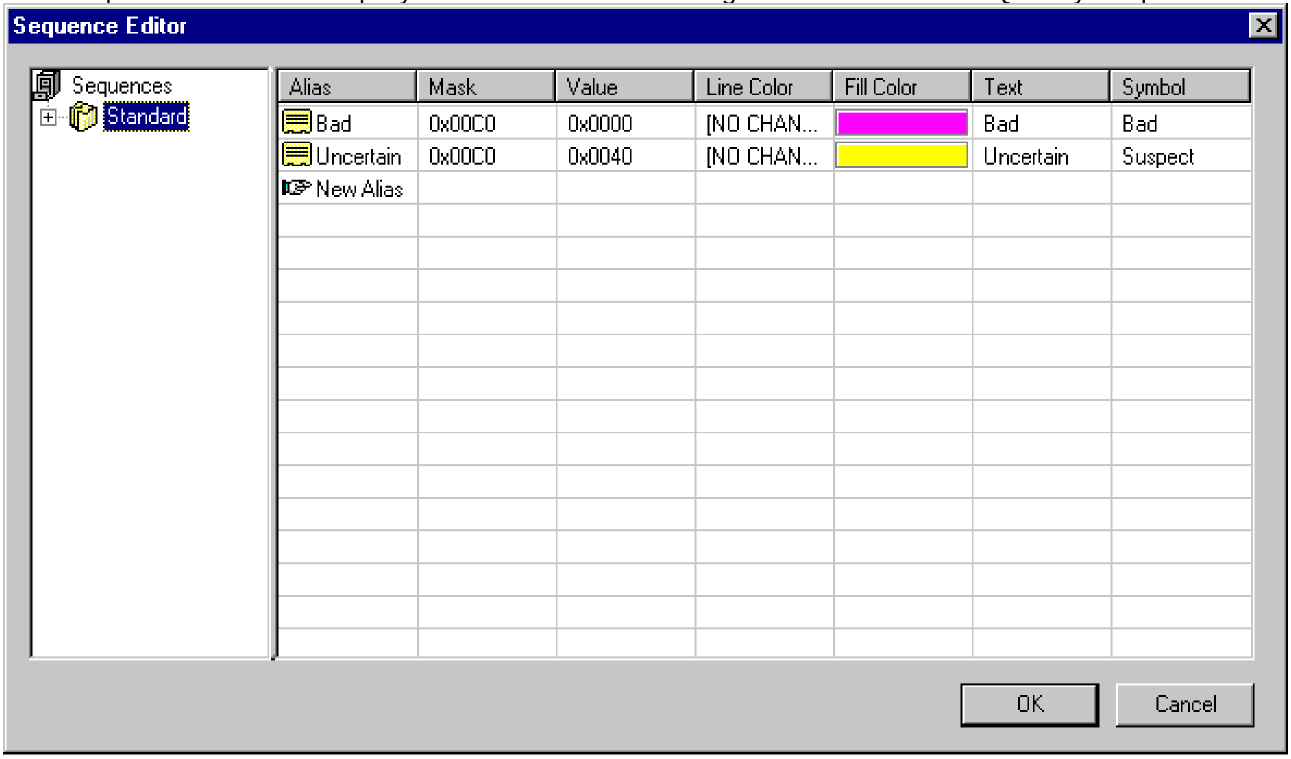
The Quality Sequences are used to customise the visual indication provided by an animation at run-time when an OPC variable has a Quality other than good. The indication may take the form of a colour change, the display of a text string or the selection of a symbol, depending on the animation type.

You can create any number of Quality Sequences which may then be assigned to each animation as required.

The Sequence Editor

How a Quality Sequence Works | Configuring Quality Sequences

The Sequence Editor is displayed from the menu using Tools.Preferences.Quality Sequences.



Alias	Mask	Value	Line Color	Fill Color	Text	Symbol
Bad	0x00C0	0x0000	[NO CHAN...	🟪	Bad	Bad
Uncertain	0x00C0	0x0040	[NO CHAN...	🟨	Uncertain	Suspect
New Alias						

How a Quality Sequence Works

A previously configured sequence is assigned to the animation in question. For example the sequence Standard is assigned to a Colour On Bit animation. At run-time, when the quality of any variable in the animation changes each row in the sequence is checked in turn. The quality value is ANDed with the Mask and the result checked against the Value. If there is a match the properties defined in the condition are substituted for those of the animation, otherwise the next row is checked and so on. If there are no matches then the properties configured in the animation are used.

The properties that are used from the sequence will depend on the animation type. If it is a colour change animation (for example colour on bit) then the Line Colour and Fill Colour will be used. If it is a text animation (for example register display) then the Text property will be used. If it is a symbol animation (for example symbol on bit) then the Symbol property will be used.

The order in which the conditions are checked is important as they are not mutually exclusive.

Configuring Quality Sequences

Open the Sequence Editor from the menu using Tools.Preferences.Quality Sequences.

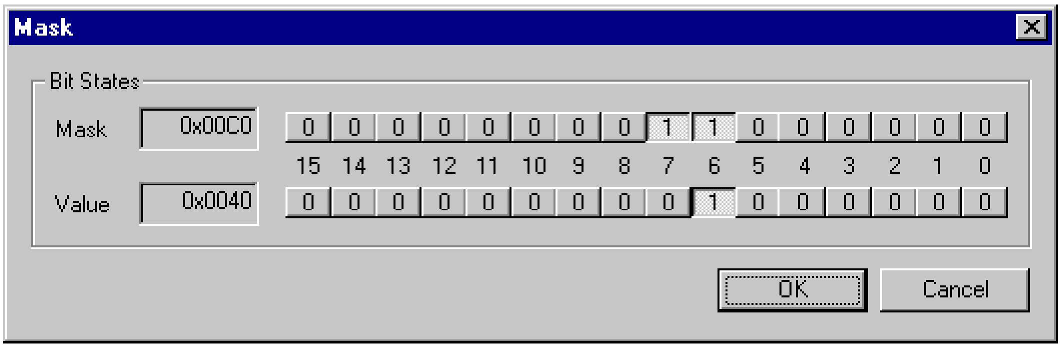
How to create a new sequence

1. Right click on the Sequences icon in the left pane and from the pop-up menu select New Sequence.
2. Enter a name for the sequence.

How to add a condition to a sequence

3. Select the sequence to which the condition is to be added.

4. In the right pane right click the New Alias icon. From the pop-up menu select New.
5. Enter a name for the condition.
6. Click in the Mask field and select the small button adjacent to the field. Select the mask and value bits. Remember that the mask is ANDed with the quality and if equal to the value the condition is valid.



Mask	0x00C0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	0x0040	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

7. Select the colour for the Line and Fill fields. If a colour change is not required, right click and from the pop-up menu select No change.
8. Enter the string in the Text field. If a text change is not required, right click and from the pop-up menu select No change.
9. Select the symbol in the Symbol field. If a symbol change is not required, right click and from the pop-up menu select No change.

To create a new sequence based on an existing one

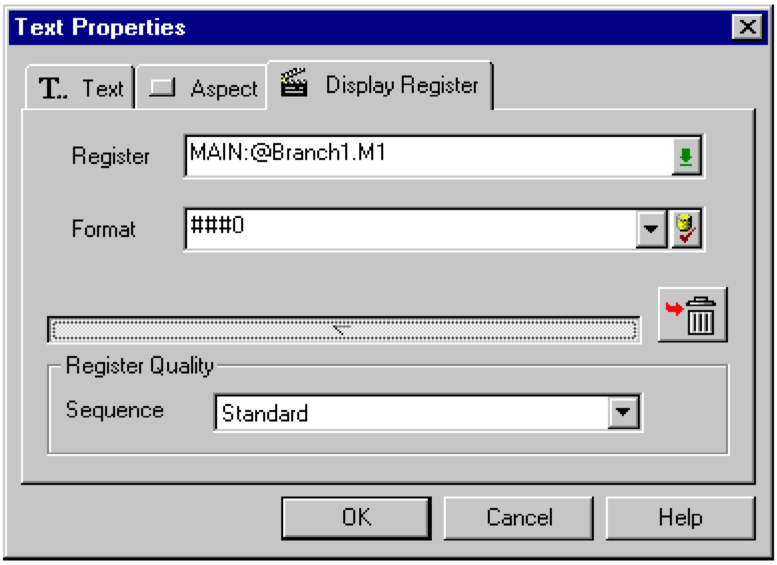
1. Select an existing sequence.
2. Right click on its name and from the pop-up menu select the copy command.
3. Click anywhere in left hand pane and from the pop-up menu select the paste command.

How to save the configuration

1. To save a new sequence or changes to an existing sequence click the OK button. Clicking the Cancel button will discard any changes.

How to Attach a Quality Sequence to an Animation

1. Display the properties box for the drawing element and select the tab for the animation.
2. Click the advanced properties button at the bottom of the box to display the Sequence selection drop down list box.



Text Properties

Register MAIN:@Branch1.M1

Format ###0

Register Quality

Sequence Standard

3. Select the required Sequence and confirm the modification using the OK button

Understanding the Difference Between Mimics and Windows

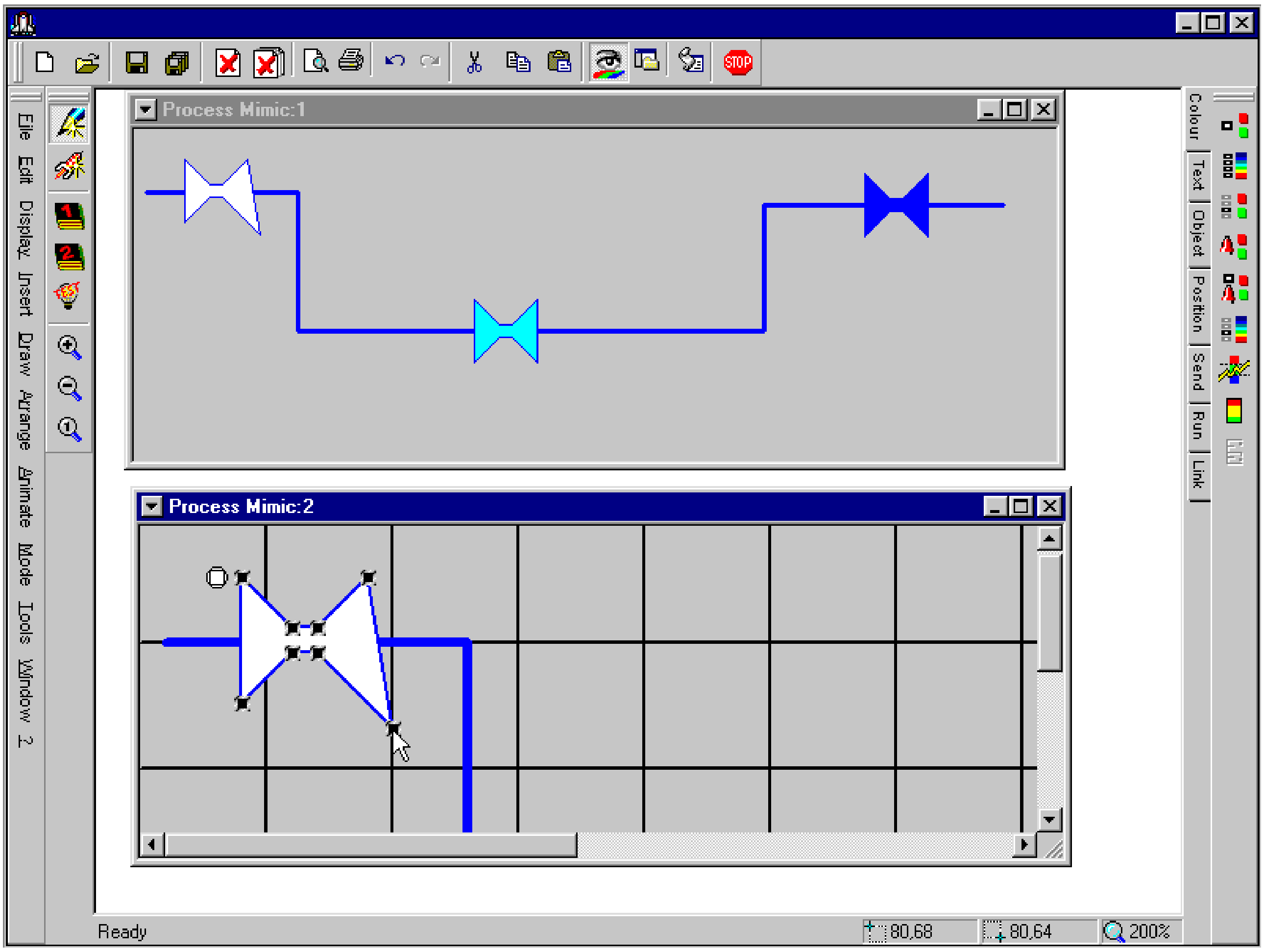
Window and Mimic Objects in the Scripting Language

The term's window and mimic are used many times throughout this manual and are generally interchangeable. However there is a subtle difference between a window and a mimic that is important to understand, particularly if you are manipulating windows and mimics programmatically.

A window is the container for a mimic. The window defines such things as the location on the screen, the style (borders, title etc.) and the access rights. There is always at least one window per opened mimic.

A mimic is the contents of a window. You draw in the mimic to produce menus, overviews, process diagrams etc. You can have more than one instance of the same mimic open at once, each in its own window. For example you could have one copy open in run mode and 100% zoom and another copy in design mode at 200% zoom.

For those readers familiar with the classic Microsoft Document/View architecture, Mimic is equivalent to Document and Window equivalent to View.



Window and Mimic Objects in the Scripting Language

When using the scripting language you will find the following objects concerned with mimics and windows.

The Mimics Collection Object

The Mimics collection object represents the collection of the currently opened mimics for the active Project.

The Mimic Object

A Mimic object represents an opened mimic. Mimic's properties or methods are related to the targeted mimic and to all windows containing this mimic (including the active one).

The Windows Collection Object

The Windows collection object represents the collection of the currently opened windows for the active Project, or the collection of currently opened windows for a specific Mimic

The Window Object

A Window object represents an opened window on a specified Mimic.

Mimic Basics

Operational Modes | The Mode Tool Bar | Designing a Mimic Hierarchy

The User Interface for your project is developed as a number of mimics. Mimics are easily and quickly developed to form Menus, Overviews, Process Diagrams, Trend Displays and so on. Mimics may be linked together, using a special type of animation, to form a hierarchical display.

Each mimic is contained within a window, the properties of which are individually selected including the size and position on the screen, the ability to move or re-size the window and the access level.

Operational Modes

Mimics may be displayed in a number of modes selected from Mode toolbar.

- Run. In run mode the animated properties of drawing elements in the mimic are updated in real time. It is not possible to change the properties of any drawing in the mimic or its support window.
- Test. Test is a sub-mode of run and is only operational when run mode is selected. In test mode the variables used in any animations are disconnected from their source and may have their real time value manually forced.
- Design. In design mode the properties of drawing elements do not update in real time. Drawing elements may be added, modified and deleted and the support window modified. This is the mode of operation used whilst developing a mimic.

The HMI supports the Microsoft Document View architecture that allows you to have several instances of the same mimic open concurrently. Each instance is contained within its own support window. For example you could have one instance opened in design mode and another in run mode.

The Mode Tool Bar

The Mode tool bar is used to switch a window and the mimic it contains between design and run modes and to zoom in and out.

Switch the selected view of the mimic to design mode.

Switch the selected view of the mimic to run mode.

Switch the selected view of the mimic to test mode.

Switch the selected view of the mimic to reference mode 1.

Switch the selected view of the mimic to reference mode 2.

Clicking on the zoom in tool will change the cursor to a magnifying glass with a plus sign in the centre. If you then click on an opened mimic the view of it will zoom in. (Its physical size on the screen will get larger). You may zoom in to a maximum of 6400 % of the original size. To deselect the zoom in mode, click on the tool again.

Clicking on the zoom out tool will change the cursor to a magnifying glass with a minus sign in the centre. If you then click on an opened mimic the view of it will zoom out. (Its physical size on the screen will get smaller) The maximum zoom out is 2% - that is the mimic is displayed on the screen one fiftieth of its normal size. To deselect the zoom out mode, click on the tool again.

Clicking on the zoom area tool will change the cursor to a magnifying glass. You can then click and drag a band to zoom into a specific area of a mimic. The zoomed mimic may open in the same or a different window depending on the configuration selected in Tools.Options.Magnifying Glass Mode. To deselect the zoom area mode, click on the tool again.

Close any windows opened as a result of using the zoom area tool. The original copy of the window remains open.

Return the selected mimic to normal size (1:1)

Designing a Mimic Hierarchy

The mimic hierarchy determines the way in which a User navigates around the project. As a User will normally not have access to the development tools which open mimics consideration must be given as to how mimics are linked together. A typical hierarchy might contain :-

- Main menu
- Sub-menus
- Plant wide overview
- Process diagrams
- Management information
- Diagnostic displays
- Trend displays

The mimic hierarchy may be designed so that the mimic’s available at any time are dependent on the access rights of the current User.

Creating and Managing Mimics

The Mimic Management Box | Creating a New Mimic | The Mimic Pop-Up Menu | Saving a Mimic | Closing a Mimic | Using Mimics With a Branch | Iconising and Restoring | Opening More Than One View of the Same Mimic

The tools used to create, save and manage mimics are found on both the Standard tool bar and on the File menu.

Create a new mimic.

Open an existing mimic.

Save the selected mimic.

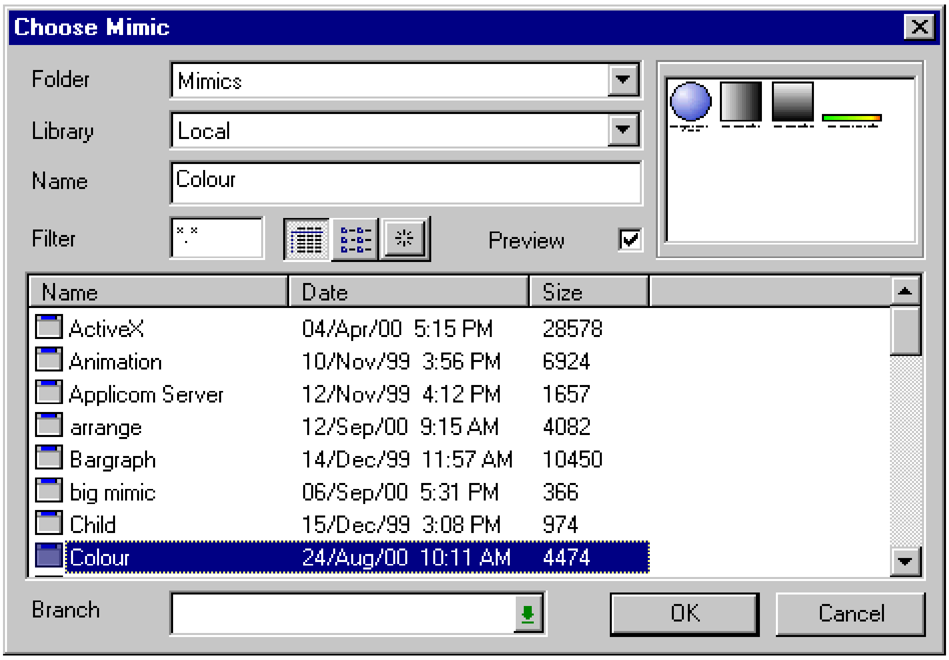
Save all open mimics.

Close the selected mimic window.

Close all mimic windows.

The Mimic Management Box

The Mimic Management dialog box is displayed when you save a mimic for the first time, save a mimic with a new name and when opening a mimic using the Open tool or the File.Open command.



Choose Mimic

Folder: Mimics

Library: Local

Name: Colour

Filter: *.*

Preview ☑

Name	Date	Size
ActiveX	04/Apr/00 5:15 PM	28578
Animation	10/Nov/99 3:56 PM	6924
Applicom Server	12/Nov/99 4:12 PM	1657
arrange	12/Sep/00 9:15 AM	4082
Bargraph	14/Dec/99 11:57 AM	10450
big mimic	06/Sep/00 5:31 PM	366
Child	15/Dec/99 3:08 PM	974
Colour	24/Aug/00 10:11 AM	4474

It provides a list of mimics, the option of a preview display of the selected mimic and tab options for selecting the mimic library and branch. The list may be switched between a simple or detailed format using the display buttons.

Tab Options and Command Buttons

- *Folder* - The name of the folder in which the mimic file is to be saved or opened from. This can either be Mimics if you are saving or opening a normal mimic or Mimic templates if you are creating or loading a mimic template. See the topic on mimic templates for further information.

- *Library* - The name of the library in which the mimic file is to be saved or opened from. By default there is one library known as Local. Other libraries may be available depending on the configuration of the project.
- *Name* - The name of the mimic to be opened or saved. You may type the name directly into this field or select it by clicking in the list. The name must comply with MS Windows' file naming conventions.
- *Filter* – A lexical filter that may be applied to reduce the number of mimic names displayed in the list.
- *Branch* - The name of a database branch to be used with the mimic. See the topic on opening a mimic with a branch for further information.

Creating a New Mimic

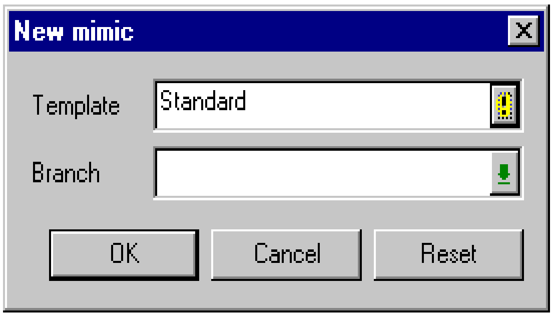
You can either create a new mimic with default properties, or a new mimic based on a mimic template.

To create a new mimic

1. Select the New tool on the Standard toolbar. A new mimic is created with default properties.

To create a new mimic using a mimic template

2. Select the File.New command. A dialog box is displayed from where you can select a



New mimic

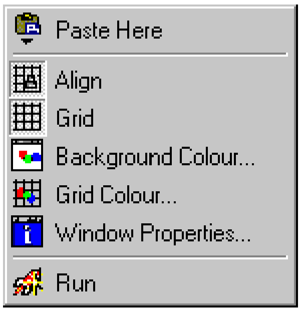
Template: Standard

Branch:

3. Select the name of the template and then select the OK button. You can optionally enter the name of a database branch for use with the mimic.

The Mimic Pop-Up Menu

The mimic pop-up menu gives quick access to some of the more frequently changed mimic properties and to the Window Properties box. It also allows you to paste the contents of the clipboard into the mimic at the current cursor position. The menu may be displayed by clicking anywhere in a mimic background with the right mouse button whilst the mimic is in design mode.



Paste Here	Paste the contents of the clipboard.
Align	Toggle alignment of drawings to the grid.
Grid	Toggle display of the grid.
Background Colour...	Change the background colour of the mimic.
Grid Colour...	Change the colour of the grid.
Window Properties...	Display the Window Properties box.
Run	Select run mode

Saving a Mimic

To save a mimic click the Save tool on the standard toolbar or select the File.Save command.

If you are saving a mimic for the first time then the mimic management box will be displayed and you must enter a mimic name. This name will be used in the future when opening the mimic and is also the name of the file in which the mimic definition will be saved.

Closing a Mimic

To close a mimic click the Close tool on the standard toolbar or select the File.Close command from the FILE menu.

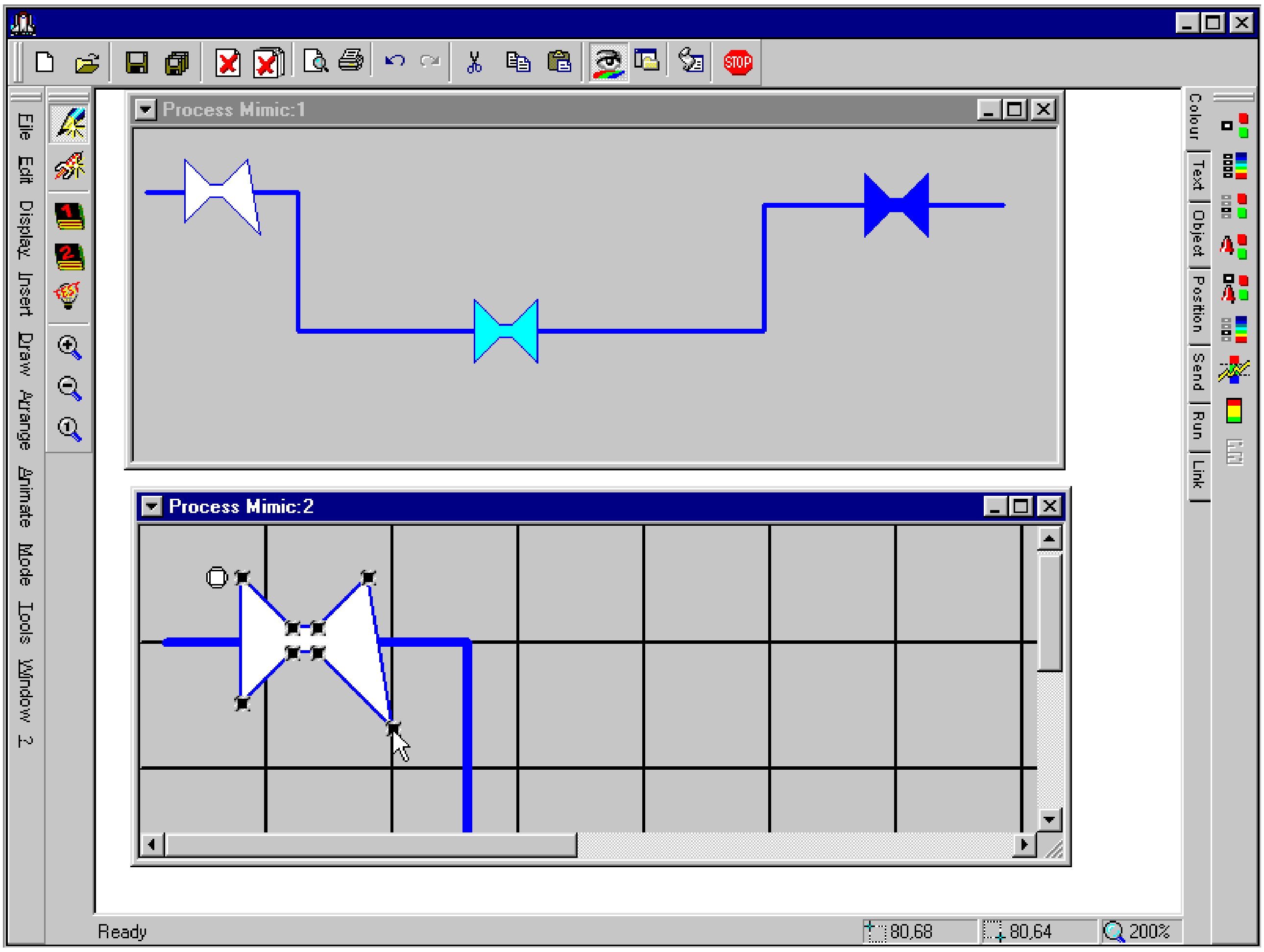
If changes have been made to the mimic since it was opened and they have not been saved you will be prompted to save the changes.

If the mimic is not displayed at 100% (normal) magnification, any changes to mimic size and position are not saved.

Opening More Than One View of the Same Mimic

It is possible to have more than one view of the same mimic open concurrently. Each view is opened in its own window and may be independently moved, zoomed and switched between design and run modes. For example you could have:

- One view in run mode and the other in design mode.
- One view zoomed in to change the fine detail and another view at full size to see the effect of those changes.



To Open Another View of a Mimic

1. Select the mimic by clicking anywhere in it.
2. Select the Window.New Window command
   Or

3. Right click on the window title bar and select the New window command from the displayed pop-up menu.

Iconising and Restoring a Mimic

Mimics that have been iconised are displayed in a column next to the right hand border of the Workspace. Iconised mimics in run mode continue to be updated in real time.

To iconise a mimic you double click in the mimic's title bar or select the minimise tool if enabled.

To restore a mimic, double click anywhere in the iconised view.

Using Mimics With a Branch

When a mimic is opened or created it may be given a database branch. Subsequently any reference to variables is relative to that branch. Used in conjunction with the structured OPC server this technique can be used to reduce the number of mimics required and the subsequent development time.

An example of using a mimic with a database branch

Consider an application where there are four identical gas turbines to be supervised. Each turbine has more than 100 variables associated with it. In the OPC server each variable name starts with the turbine number.

```
Data:turbine1.speed     Data:turbine1.pressure  etc.
Data:turbine2.speed     Data:turbine2.pressure  etc.
Data:turbine3.speed     Data:turbine3.pressure  etc.
Data:turbine4.speed     Data:turbine4.pressure  etc.
```

A mimic named Turbine is created initially with the branch `Data:turbine1`. Subsequently all variable references are relative to `Data:turbine1`, so speed actually refers to `Data:turbine1.speed`. A further mimic named Menu is created with four command buttons each one with a link to the Turbine mimic but using four different branches each corresponding to one of the turbines.

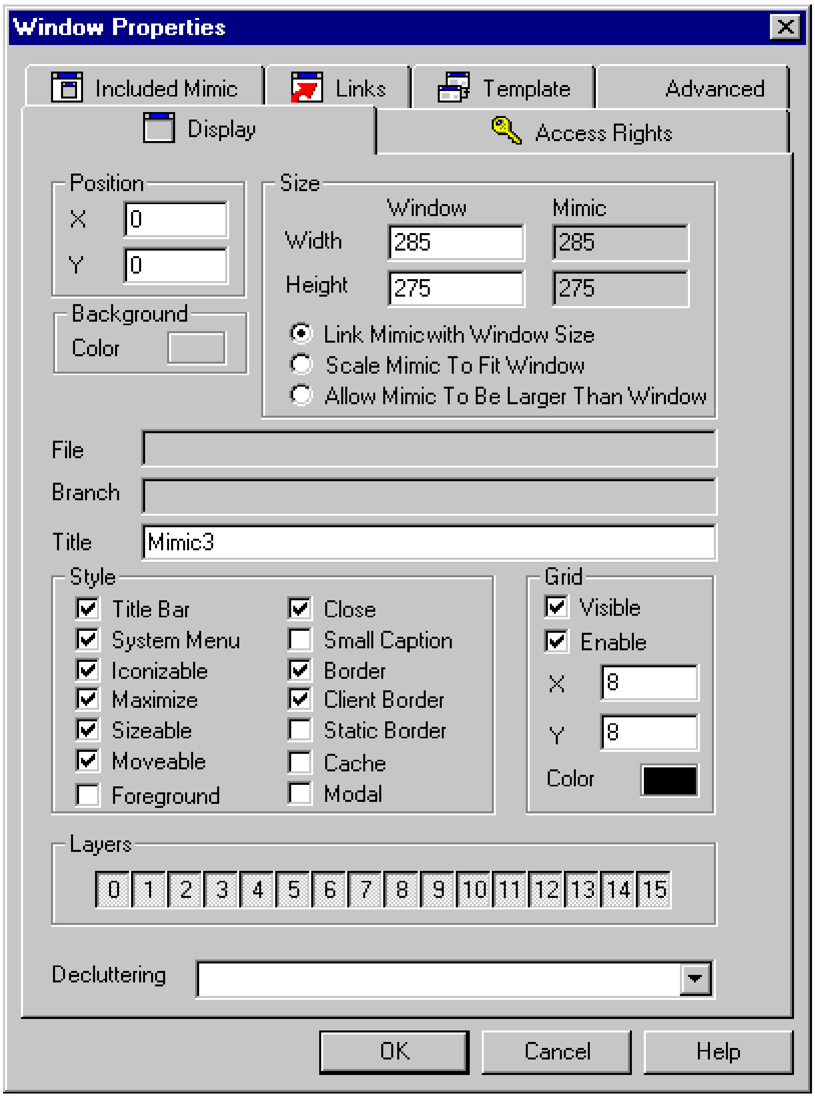
At run-time the Turbine mimic may be opened four times, each time displaying a different group of variables from the server.

With this example instead of creating four mimics it was only necessary to create one. This has the dual advantage of reducing development time and making maintenance much easier.

The Window Properties Box

Defining the Window Style | Selecting the Window and Mimic Size | Defining the Drawing Grid | Displaying Text in the Title Bar | Drawing Layers and De-clutter Templates | The Access Rights Tab | The Included Mimic Tab | The Links Tab

The Window Properties box is displayed from the mimic pop-up menu and defines the behaviour and operation of a mimic's support window.



Position
X 0
Y 0

Background
Color

Size

	Window	Mimic
Width	285	285
Height	275	275

● Link Mimic with Window Size
○ Scale Mimic To Fit Window
○ Allow Mimic To Be Larger Than Window

File
Branch
Title Mimic3

Style

☑ Title Bar	☑ Close
☑ System Menu	□ Small Caption
☑ Iconizable	☑ Border
☑ Maximize	☑ Client Border
☑ Sizeable	□ Static Border
☑ Moveable	□ Cache
□ Foreground	□ Modal

Grid
☑ Visible
☑ Enable
X 8
Y 8
Color ■

Layers
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Decluttering

Defining the Style

The style is the name given to the group of properties that effect the general behaviour of the window. Note that some of the properties interact with one another, for example if you enable the system menu the title bar is automatically displayed.

- *System menu* - Display the system menu icon on the title bar.
- *Iconizable* - Display the iconise button on the title bar.
- *Maximise* - Display the maximise button on the title bar.
- *Sizeable* - Allow the window to be sized by clicking and dragging the border.
- *Moveable* - Allow the window to be moved by clicking and dragging the title bar. Note that the title bar is not automatically enabled.
- *Foreground* – The window is always displayed in the foreground - that is any other windows opened at the same time will be forced behind it.

- *Modal* – The window exhibits modal behaviour. That is the window retains focus until it is closed.
- *Close* - Display the close button on the title bar.
- *Small caption* - Display a narrow title bar and use a small font for the window title.
- *Border* - Display a standard style border around the window.
- *Client border* - Display a client style border around the window.
- *Static border* - Display a static style border around the window.
- *Scale to fit size* - See the topic Using the Scale to Fit Size Property.
- *Cache* - The window is saved in cache memory after the first time it has been opened. Subsequently, when the window is opened it will appear on the screen quicker than if was being opened from the hard disk. Note that when a window is in cache it uses system resources (memory and CPU time) and so this option should only be used when essential. The cache is restricted to a finite number of windows as defined in the project start up conditions. If you try and cache more than this the window that was opened first, that is the oldest in the cache, will be removed from the cache.

Selecting the Window and Mimic Size

Normally the window and mimic it contains will be the same size. However there are some circumstances where it is desirable to have the mimic a different size to the window. The following properties set the window and mimic size and the relationship between them.

- *Link mimic and window size* - The default selection. The window and mimic it contains are the same size. The size may be changed either by entering the width and height in the corresponding fields or by dragging the window border.
- *Scale mimic to fit window* - The window and mimic can be different sizes. The mimic is scaled to fit it to the window. The mimic size is entered in the mimic width and height fields. The window size can be entered in the window width and height fields, or by dragging the window border. When this option is selected the zoom tools are not available.
- *Allow mimic to be larger than window* - The mimic can be larger than the window. The mimic size is entered in the mimic width and height fields. The window size can be entered in the window width and height fields, or by dragging the window border. If the window is smaller than the mimic scroll bars are automatically displayed on the window borders to allow navigation of the mimic.

Displaying Text in the Window Title Bar

If you enable the Title bar option you can enter a title or description to be displayed in the title bar. As well as fixed text you may also use the following substitution characters to display certain information about the system.

#h	The current hour as two digits.
#m	The current minute as two digits.
#s	The current second as two digits.
#B	The database branch that the window has been opened with.
#D	The current date as two digits.
#M	The current month as two digits.
#W	The window name.
#Y	The current year as two digits.
###Y	The current year as four digits.

If the title bar is enabled and no text is specifically entered the window name will be displayed.

Defining the Drawing Grid

The drawing grid is a tool to help you precisely position to your drawing elements. It is an imaginary matrix of co-ordinates within each window.

When you create drawing element, for example a rectangle, its size and position will automatically adjust so that each of its corners is located on the nearest grid point. When you move a drawing element, as you release the mouse button, the top left hand corner will automatically move to the nearest grid point.

The grid is displayed as a series of dashed lines in both the horizontal and vertical planes. The displayed grid is a multiple of the grid size and is only displayed as a guide. You should decide on a grid size before you start drawing in a window as changing it part way through development may cause problems when aligning drawing elements created with the previous grid size.

The following properties effect the operation of the grid.

- *Visible* - Displays or hides the lines representing the grid. The grid operation is unaffected.
- *Enable* – If selected than all subsequently produced drawing elements are aligned with the grid.
- *X & Y* - The grid size in pixels. It is recommended that the grid is made no smaller than 4 pixels. The default grid size of 8 pixels is adequate for most applications.
- *Colour* - The colour of the grid lines at design time.

Drawing Layers and De-clutter Templates

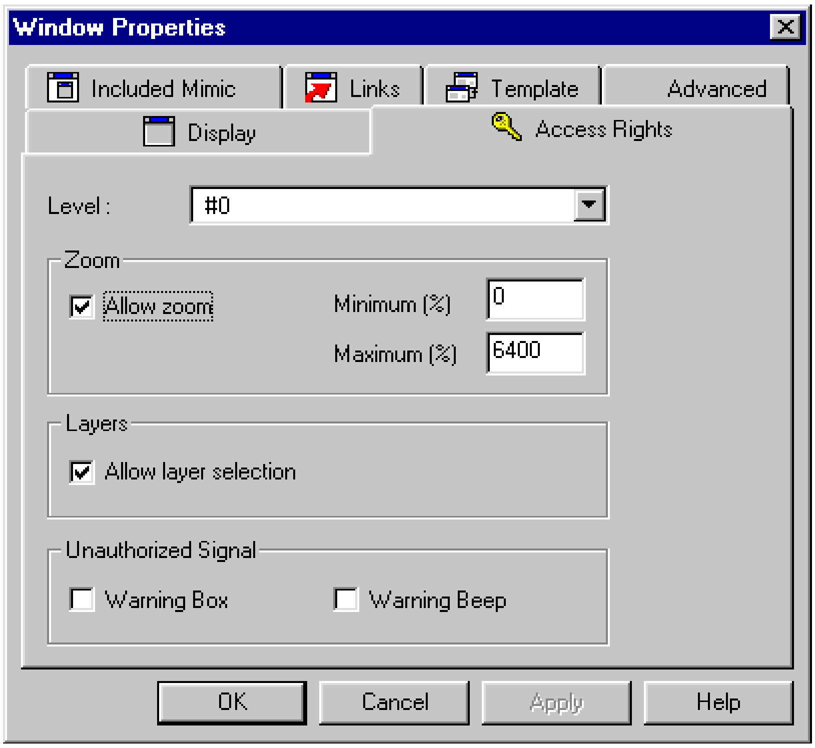
The HMI supports a CAD like layer system for drawing. When you create any drawing element it is allocated to one or more layers. In order for the drawing to be visible, one or more of the corresponding layers must be enabled.

In the window configuration you can select the enabled layers in two ways.

- By using the layer buttons.
- By specifying a de-clutter template. De-clutter templates enable layers according to zoom level. If you select a de-clutter template then the layer select buttons are no longer operational. See the topic on Creating and Using De-clutter Templates for more information.

The Access Rights Tab

The access rights tab contains properties that determine how the User can change the appearance of a mimic at run-time.



Level : #0

Zoom

☑ Allow zoom

Minimum (%) 0

Maximum (%) 6400

Layers

☑ Allow layer selection

Unauthorized Signal

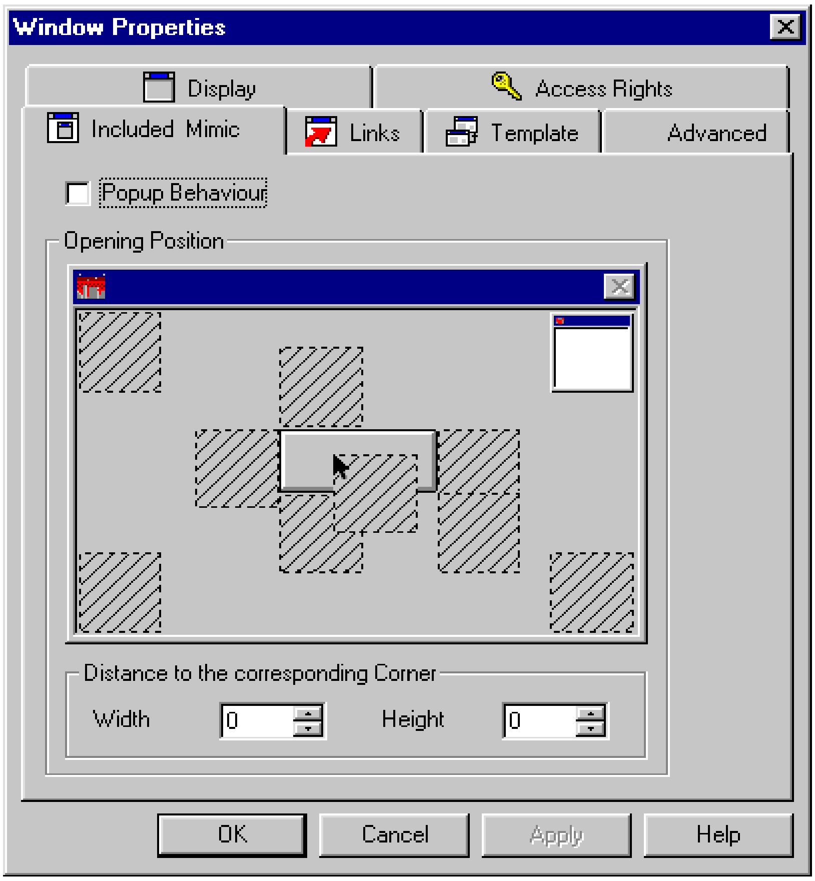
□ Warning Box □ Warning Beep

Tab options and command buttons

- *Level* - The access level that the User must have in order to open the mimic.
- *Allow zoom* – If selected the User is allowed to zoom the mimic between the specified zoom levels.
- *Allow layer selection* – If selected the User is allowed to change the layers that are displayed at run-time.
- *Warning box* – If selected a warning box is displayed if the User attempts to change the zoom or layers display when it is not permitted.
- *Warning Beep* – If selected a warning sound is produced if the User attempts to change the zoom or layers display when it is not permitted.

The Included Mimic Tab

This tab determines how the mimic behaves when it is opened as a child of another mimic (using the Link – Open animation) or when opened as a context mimic.



□ Popup Behaviour

Opening Position

Distance to the corresponding Corner

Width 0

Height 0

Configuring the opening position

The opening position determines where a mimic is opened relative to the parent mimic. You select the position by clicking on one of the hatched areas in the representation of the mimic.

You can choose to open the mimic relative to one of the corners of the parent mimic, or relative to the support drawing element for the link-open animation. You can also open the mimic relative to the cursor position.

If you choose to open the mimic relative to one of the corners then you can also specify the distance, in terms of X and Y co-ordinates, from that corner.

If you do not select any opening position then the mimic will be opened at the position defined by its own X and Y co-ordinates (as in the display tab).

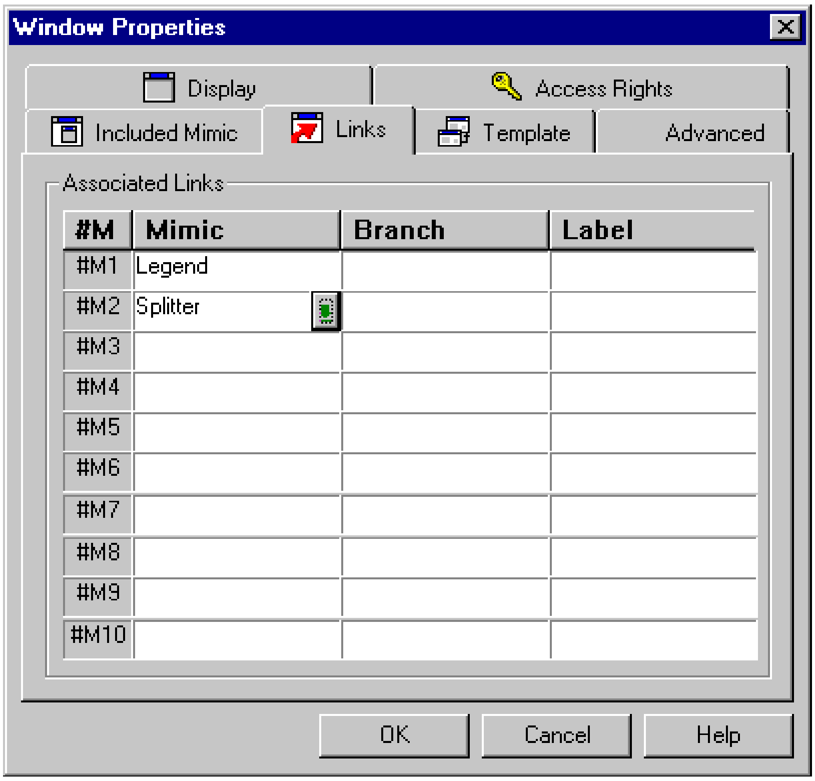
Enabling popup behaviour

When enabling popup behaviour the operation of the mimic at run time is modified as follows.

- A pop-up mimic is always opened as a child of another mimic.
- A pop-up mimic automatically closes as soon as focus is returned to the parent mimic, normally when the parent mimic is next clicked into.
- A box with dashed lines is displayed around the control zone from which the mimic is opened.

The Links Tab

The options available from this tab determine which mimics are opened from Link - open animations when using the substitution characters #M1 to #M10. For further information see the section on opening and closing windows in the Control Zone topic.



#M	Mimic	Branch	Label
#M1	Legend		
#M2	Splitter		
#M3			
#M4			
#M5			
#M6			
#M7			
#M8			
#M9			
#M10			

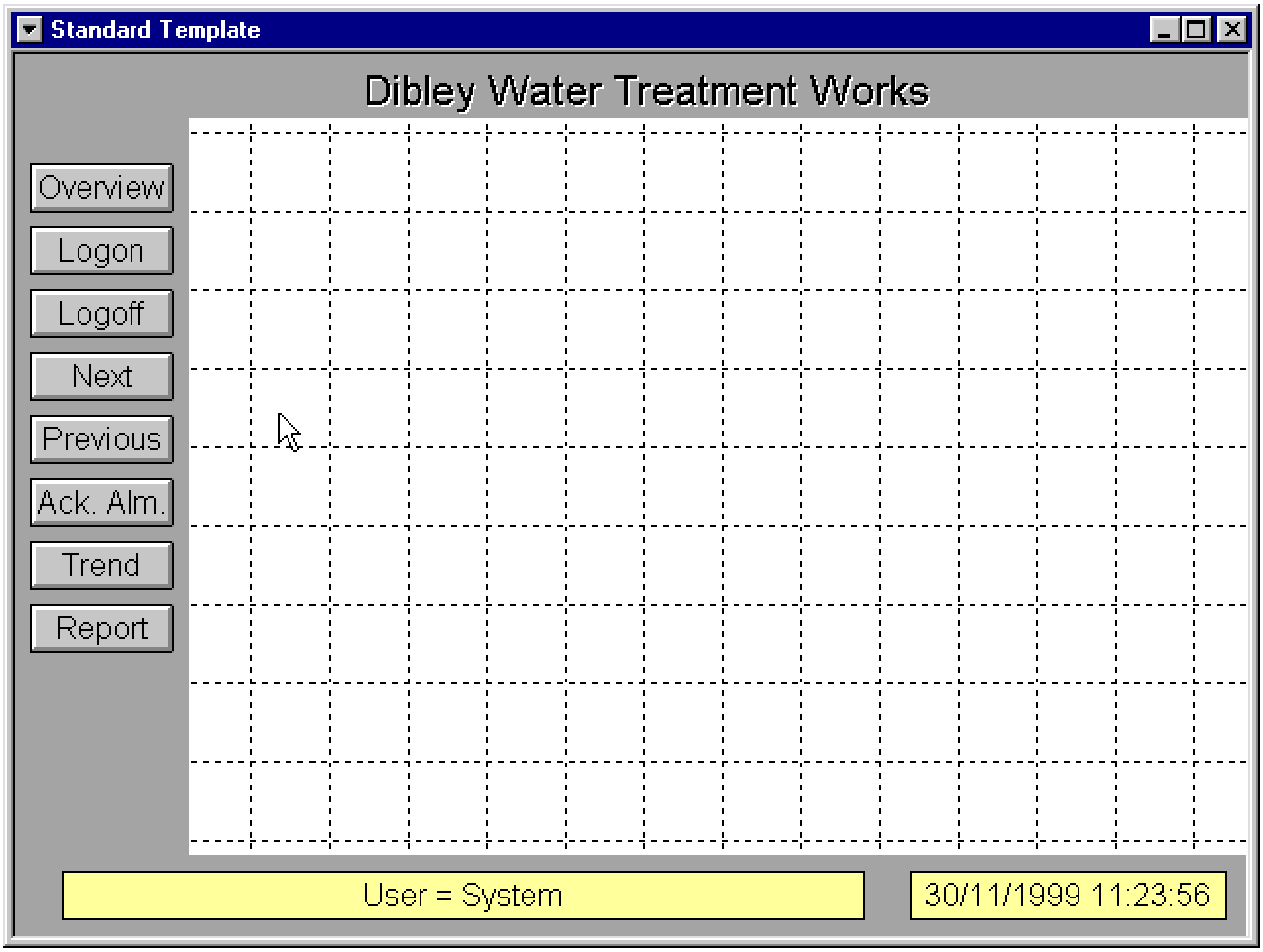
Creating and Using Mimic Templates

Creating Mimic Templates | The Template Tab

When you create a new mimic there is the option to base it on a mimic template. The new mimic will inherit certain properties from the template and any drawing in the mimic template will also appear in the new mimic.

The reference to the mimic template is dynamic. That is if you change any properties or drawing in the template these changes are automatically reflected in any mimic based on it. You can select which of the properties your new mimic inherits from the template from the template tab in the window properties box.

Used carefully mimic templates can considerably reduce the configuration time for a project and also have the advantage of giving mimics in the project a standard look and feel.



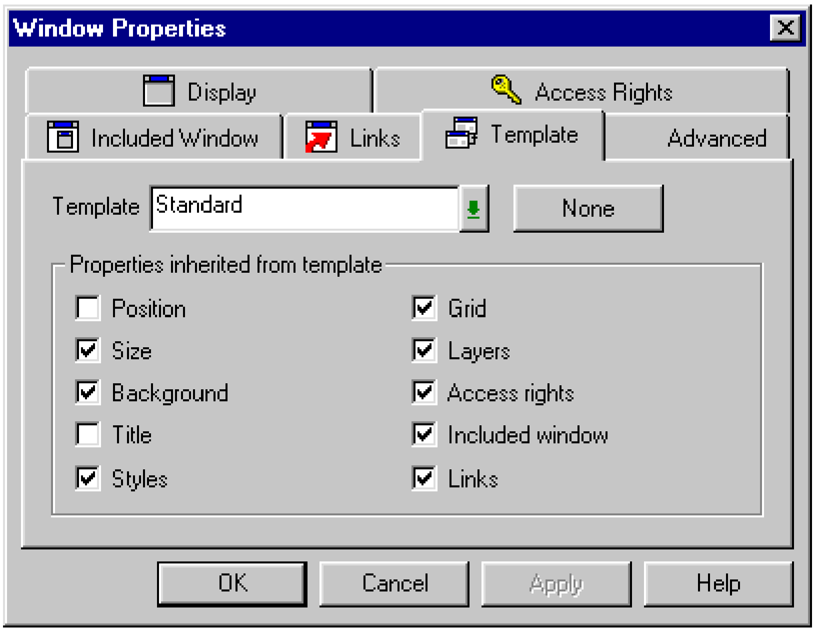
Creating Mimic Templates

A mimic template is created in the same way as any other window except that it is saved in the Mimic templates folder. Drawing elements and animations in the mimic template will be fully functional in any mimic based on it, but they can only be edited and changed in the template.

You cannot nest templates. That is a template cannot be based on another template

The Template Tab

In the Window properties box for any mimic based on a template you can select which of the properties of the template the mimic will inherit. You can also change template on which the mimic is based or even remove it altogether.



Template: Standard

Properties inherited from template

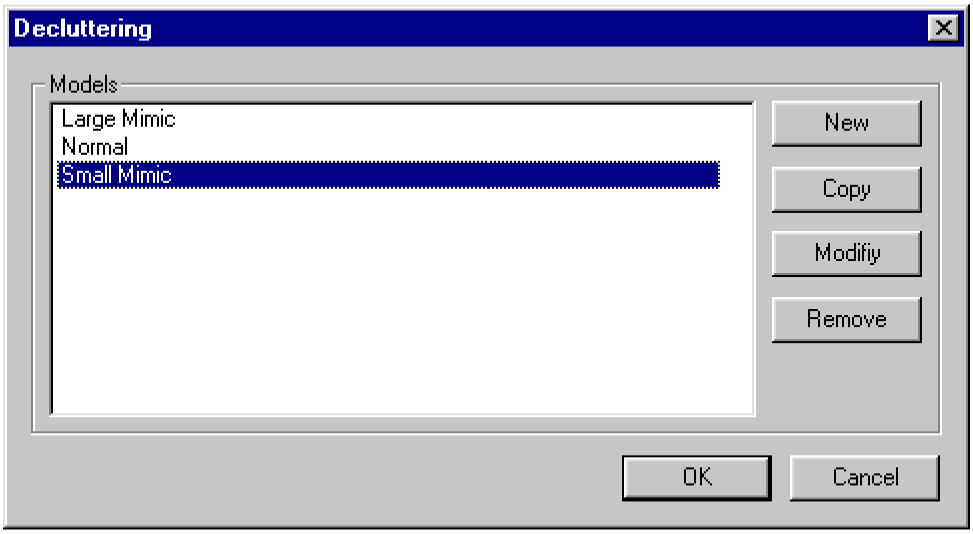
□ Position	☑ Grid
☑ Size	☑ Layers
☑ Background	☑ Access rights
□ Title	☑ Included window
☑ Styles	☑ Links

Creating and Using De-clutter Templates

When drawing, you can control the visibility of a drawing element by giving it zoom bounds. If the mimic in which it is contained is zoomed outside of these bounds the drawing element becomes invisible. This technique (known as de-clutter) is used to control the level of detail displayed in a mimic as it is zoomed. The disadvantage of controlling visibility in this way is that you must allocate the zoom bound to every drawing element it is to effect. A de-clutter template simplifies this by allocating zoom bounds to a layer. You then attach the drawing element to that layer.

How to Create a De-clutter Template

1. Select the Tools.Preferences.Decluttering command. The de-cluttering management box is displayed.

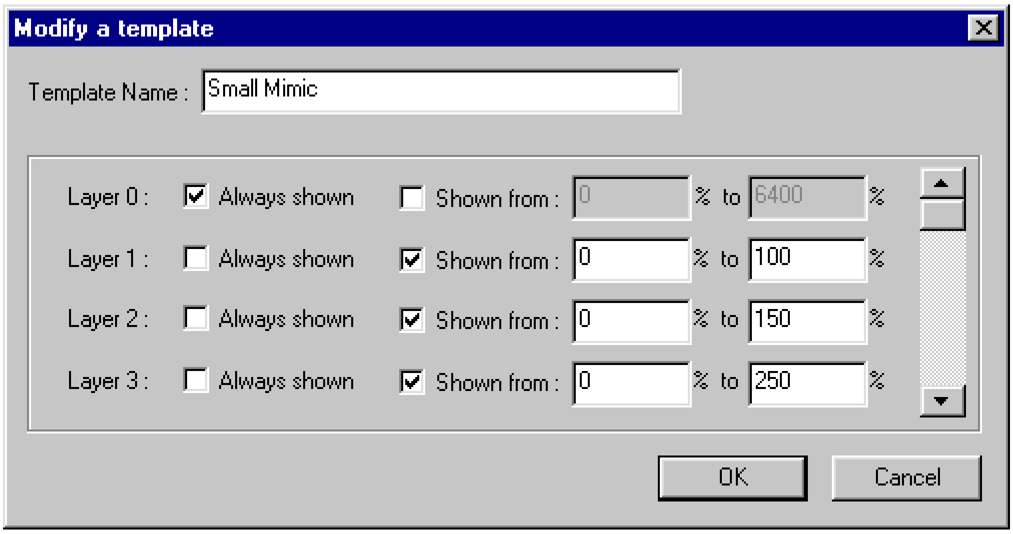


Decluttering

Models

Large Mimic
Normal
Small Mimic

2. Select the New command. A new de-clutter template is displayed.



Modify a template

Template Name : Small Mimic

Layer 0 : ☑ Always shown ☐ Shown from : 0 % to 6400 %
Layer 1 : ☐ Always shown ☑ Shown from : 0 % to 100 %
Layer 2 : ☐ Always shown ☑ Shown from : 0 % to 150 %
Layer 3 : ☐ Always shown ☑ Shown from : 0 % to 250 %

3. Enter a name for the template. This will be used when referencing it in a window properties box.
4. Enter zoom bounds for those layers that are to be dependent of zoom level.
5. Select OK to confirm the template and OK again to close the de-clutter management box.

What is Drawing?

Using Geometric Shapes | Using Images | Using OLE Objects

Drawing is the process by which you transform your blank mimics into Menus, Overviews, Process Mimics etc. You can draw in mimic using three different mechanisms.

- Using the Workspace tools you can create mathematically defined drawing elements (geometric shapes) including rectangles, ellipses, lines, polygons and text. Each element has a wide range of properties which allow you to quickly and easily change the appearance. For example a rectangle can appear as a button.
- By importing images in standard Windows' BMP, JPEG or WMF formats you can display scanned images, standard Windows' icons etc.
- Using OLE (Object Linking & Embedding) you can include information created in other applications, such as charts, graphics, and spreadsheet data.

You display the process you are supervising by attaching properties of the drawing elements to the real time value of variable in the database. For example a polygon representing a valve can change colour depending on the valve position. Attaching properties of drawing elements to the real time value of variables is known as animation. Animation is described in a later chapter.

Using Geometric Shapes

The drawing elements produced by the workspace tools are defined as a set of co-ordinates plus properties describing colour and style. They are, in effect, geometric shapes. The properties of geometric shapes are much easier to change and to attach to the value of variables in the database. The for and against points of the use of geometric shapes may be summarised as follows.

For

- Easy to draw and change.
- Very wide range of possibilities for dynamically changing properties.
- Mimic definition files containing only geometric shapes are very compact.
- Geometric shapes use memory efficiently when they are displayed.

Against

- Pictures created with just geometric shapes generally are less attractive than those using images.

Using Images

An image is a map of the area it occupies on the screen with the colour of each pixel (picture element) described by a number of digital bits. The number of bits used to define each pixel will depend on how many colours the picture is using, 8 bits per pixel can define 256 colours, 16 bits per pixel 64,000 etc. Once an image is created there is very little that can be done to change it dynamically. You cannot, for example, attach an area of an image to a database variable so that it changes colour according to the value of the variable. The points for and against the use of images may be summarised as follows.

For

- There are many drawing and paint packages able to produce stunning pictures in image format.
- images may be created from "scanned in" pictures of real world objects.
- The number of colours which may be displayed in a image is limited only by the capability of the host PC.

Against

- Images, particularly those with lots of colours, need very large files to store them and use considerable amounts of memory (RAM) when they are displayed.
- Limited possibilities for dynamically changing images according to the value of database variables.

Using OLE Objects

When linking or embedding information from another application, the information is displayed in the mimic in the same form as it would be in the original application. For example you could display a Word document. When you double click on the object in edit mode the application in which the object was originally created is automatically opened and the object is displayed ready for editing. When you return to the HMI the mimic reflects any changes you've made to the object.

For

- You can display information in virtually any form as long as there is a Windows' application, supporting OLE, capable of editing it.

Against

- You must use the original application to change the object.
- Very limited possibilities for dynamically changing OLE objects according to the value of database variables.

The Draw Toolbar

The Draw toolbar gives you quick access to all the commonly used tools to produce drawing elements. (The Arrange toolbar may be used after to align and manipulate drawing elements once they are produced.)

Select, move or size a drawing.

Rotate a drawing element.

Display the properties box for the selected drawing element.

Display the properties list for the selected drawing element.

Draw a line.

Draw a rectangle.

Draw a rounded rectangle.

Draw an ellipse or circle.

Draw a 1/4 ellipse.

Draw a 1/2 ellipse.

Draw an arc.

Draw a pie.

Draw a polygon.

Draw a polyline.

Draw a bézier curve.

Draw a bézier shape.

Insert text.

Insert a symbol from the library.

Insert an image from the library.

Insert an ActiveX control from the favourites list.

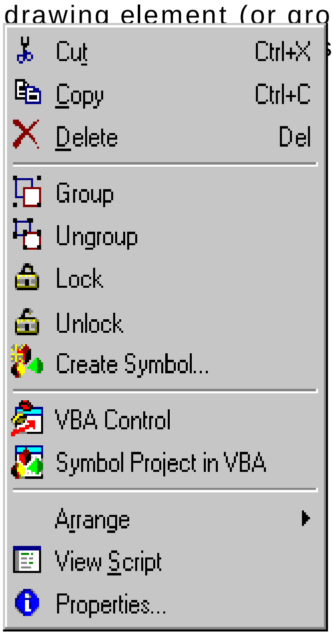
Insert an ActiveX control from the qualified list.

Insert an OLE link.

Insert a real time trend display.

The Drawing Pop-Up Menu

The drawing pop-up menu gives quick access to the most frequently used commands and to a drawing element properties box. The menu is displayed by clicking anywhere in a selected drawing element (or group of elements) with the right mouse button. The mimic in which the s must be in edit mode.



Cut the drawing element and place it on the clipboard.

Copy the drawing element to the clipboard.

Delete the drawing element.

Create a drawing group from the selected elements.

Restore a drawing group to its individual elements.

Lock the drawing element.

Unlock the drawing element.

Create a symbol using the selected drawing elements.

Enable the drawing element so that it may be used as a VBA control.

Enable use of VBA script inside symbol.

Arrange the selected drawing elements.

Display the scripting language editor.

Display the drawing properties box.

The Drawing Grid

The grid is a tool to help you precisely position your drawing elements. It is an imaginary matrix of co-ordinates within each mimic. When you create drawing element, for example a rectangle, its size and position will automatically adjust so that each of its corners is located on the nearest grid point. When you move a drawing element, as you release the mouse button, the top left hand corner will automatically move to the nearest grid point.

The behaviour of the grid may be modified as follows.

- The colour of the grid may be changed from the mimic pop-up menu or the mimic properties box.
- Display of the grid may be toggled on and off using the mimic pop-up menu or the mimic properties box. Even if the grid is not displayed drawing elements are still aligned to it.
- The resolution of the grid may be changed from the mimic properties box.
- The grid may be turned off completely using the mimic pop-up menu or the mimic properties box. Drawing elements subsequently produced or modified will not be aligned to the grid

Changing the grid resolution

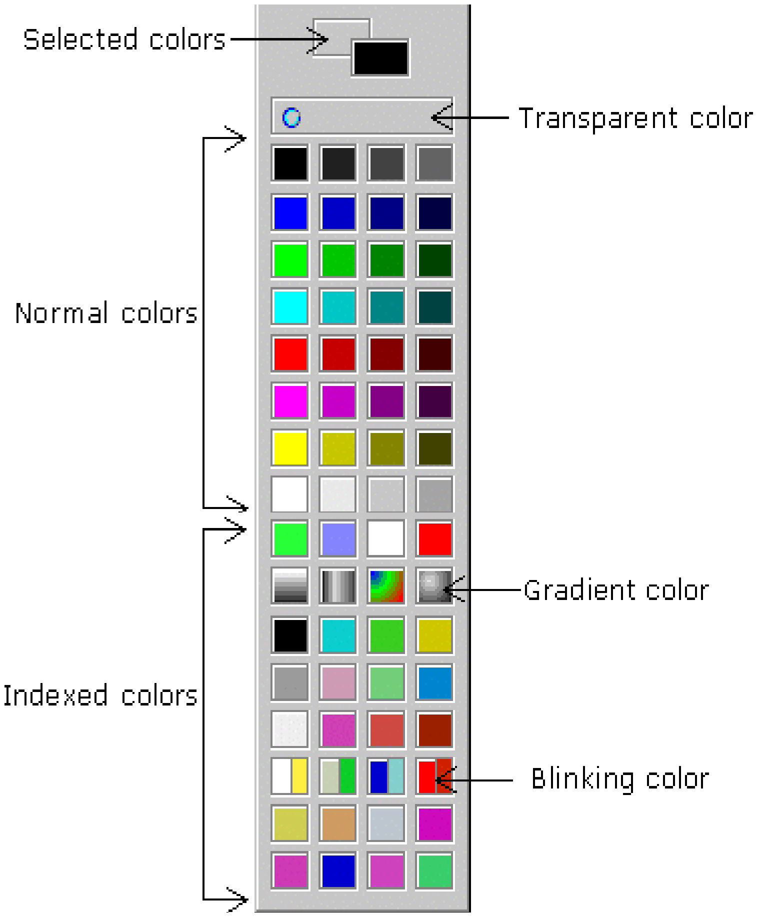
The standard grid resolution of 8 pixels is suitable for nearly all applications. However if you have a special requirement it is possible to change the grid resolution from the mimic properties box.

1. Display the Mimic properties box (using the Mimic pop-up menu).
2. Enter new values for the grid resolution in pixels. The minimum recommended grid size is 4 pixels.
3. Select OK to confirm the change.

The Colour Palette

The Colour Picker | Selecting a Solid Colour | Selecting a Blinking Colour | Selecting a Gradient Colour | Gradient Colour Performance Considerations | Using the Colour Palette to Colour a Drawing

The colour palette is displayed each time you select a colour property from any drawing element properties box and may also be displayed using the Display.Color Palette command.



It is essential to understand the use of each area of the Colour Palette before starting any drawing.

- Selected colours. The colours currently in use.
- Transparent colour. Click here to choose one of the selected colours as transparent. When you use a transparent colour for a drawing, the background will be seen through it.
- Normal colours. When you colour a drawing using one of the colours from this area the colour is saved in the drawing. If you change the colour in the palette the colour in the drawing is not affected.
- Indexed colours. When you colour a drawing using one of the colours from this area the colour is inserted as a reference. If you change the colour in the palette the colour of all drawings using it will also change.

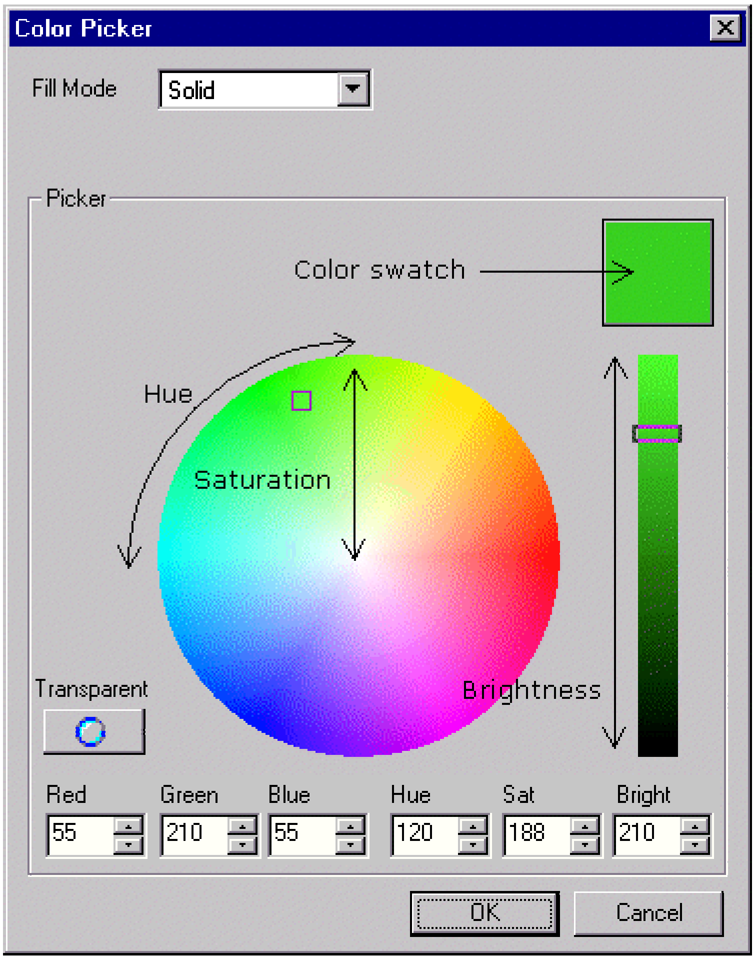
Colours in the palette are changed using the Colour Picker. To display the Colour Picker you double click on the colour in the palette that you want to change. There are three colour styles available.

- Solid. A single colour.
- Blinking. A blinking colour contains two colour selections. The affected part of the drawing element will alternate between the two colours at a pre-defined rate.

- Gradient. A gradient colour contains between two and eight colours. The affected part of a drawing element will be shaded starting with one colour gradually changing to the other colours across the drawing.

The Colour Picker

To display the Colour Picker you double click on the colour in the palette that you want to change. The Colour Picker is also displayed when double clicking on a colour box in any drawing properties box.



Color Picker

Fill Mode: Solid

Red: 55
Green: 210
Blue: 55
Hue: 120
Sat: 188
Bright: 210

A colour is defined by the three components of hue, saturation, and brightness.

- Hue describes the colour's shade or tint. It is measured on a circular spectrum running from red to green to blue and returning to red. Hue is selected on the colour circle by the angular position of the colour pick box with respect to the centre of the circle.
- Saturation describes the hue's purity. A colour with a saturation of 255 is bright and vivid, and a colour with a saturation of 0 is a shade of grey. Saturation is selected by the distance of the colour pick box from the centre of the colour circle.
- Brightness describes the colour's brightness. A colour with brightness of 0 is black.

Selecting a Solid Colour

1. Select the fill mode as solid.
2. Select the hue and saturation using the colour pick box.
3. Select the luminance using the pointer control.
4. Click OK to apply the colour.

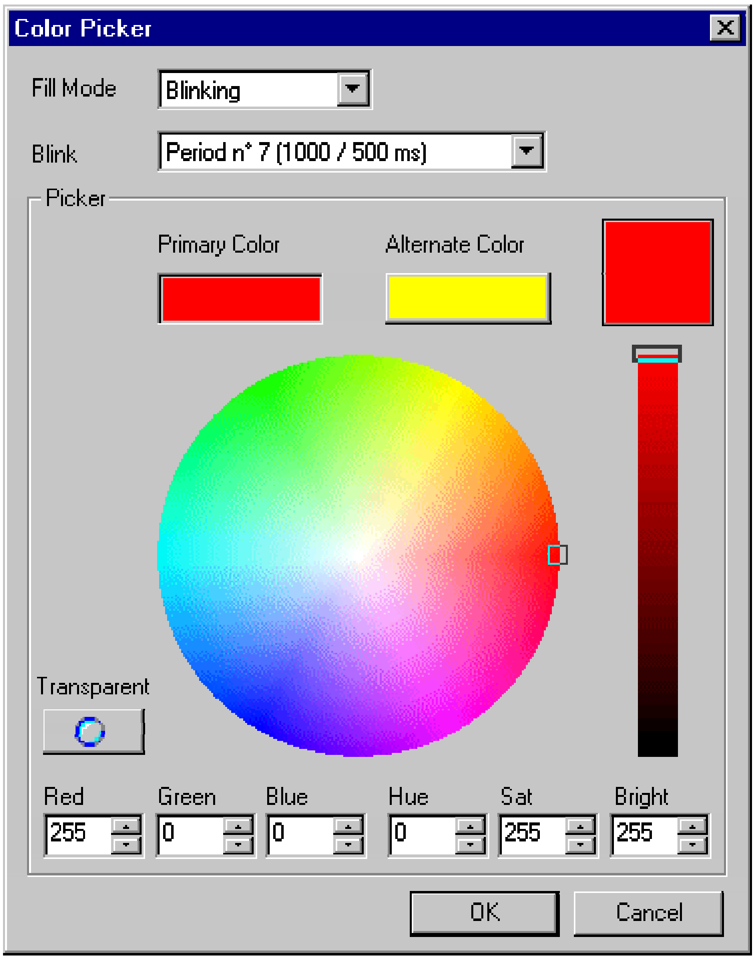
How to define a colour as transparent

When a transparent colour is applied to a drawing element the effected property becomes invisible and the background or any drawing elements under it may be seen.

1. Tick the Transparent colour tick box.
2. Click OK to apply the colour.

Selecting a Blinking Colour

Some drawing elements support blinking colours. A blinking colour contains two colour selections. At run time the affected part of the drawing element will alternate between them at a pre-defined rate.



Fill Mode: Blinking

Blink: Period n° 7 (1000 / 500 ms)

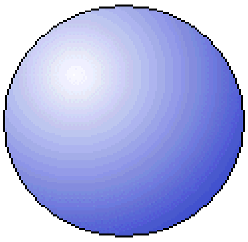
Red: 255
Green: 0
Blue: 0
Hue: 0
Sat: 255
Bright: 255

1. Select the fill mode as blinking.
2. Click the primary colour and use the colour picker to select it.
3. Click the alternative colour and use the colour picker to select it.
4. Select which of the blinking periods to use (There are 8 configurable blink rates).
5. Click OK to apply the colour.

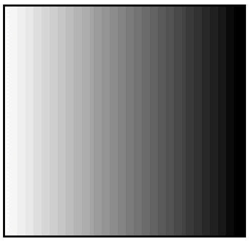
The blinking rates are selected from the menu using the Tools.Preferences.Blinking command.

Selecting a Gradient Colour

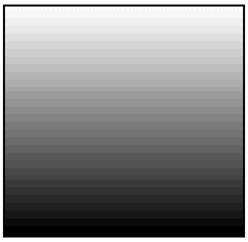
A gradient colour contains two or more colours. The affected part of a drawing element will be shaded starting with one colour gradually changing to the other colour(s) across the drawing. The type of gradient may be selected from either spot or linear. The following shows some example of the use of gradient colours.



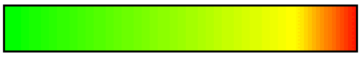
Spot with a
centre of 30, 30



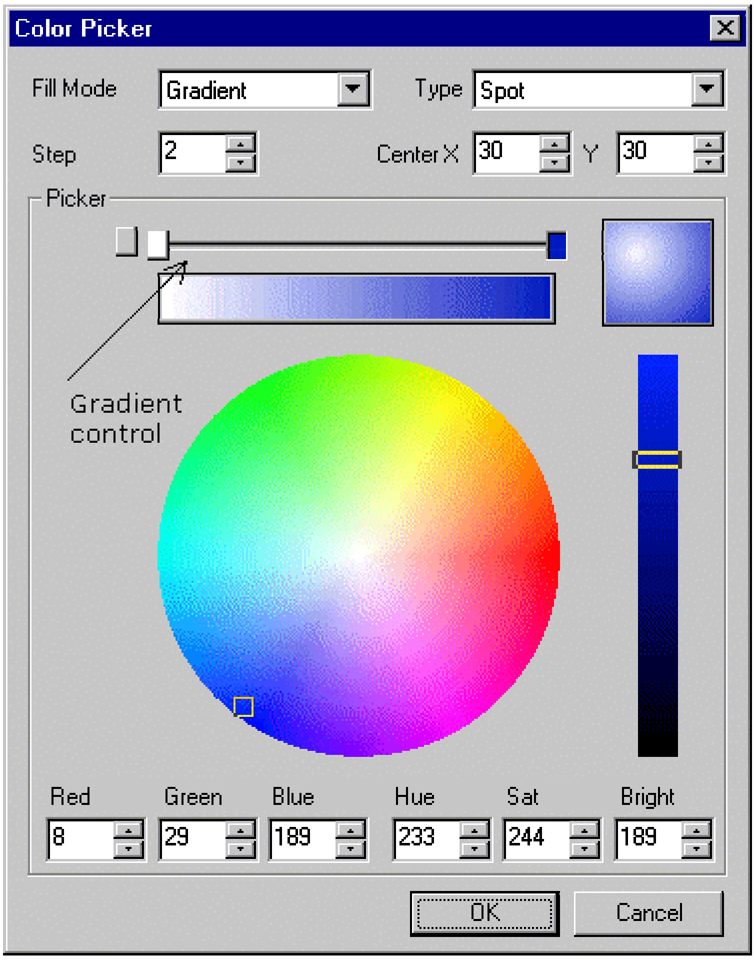
Linear with angle 0



Linear with angle 90



Linear with a third color



Color Picker

Fill Mode: Gradient
Type: Spot
Step: 2
Center X: 30 Y: 30

Red: 8 Green: 29 Blue: 189 Hue: 233 Sat: 244 Bright: 189

Gradient
control

How to select a colour with a spot gradient

1. Select the fill mode as gradient.
2. Select the type as spot.
3. Click the one of the small colour boxes on the gradient control above the colour circle and select its colour using the colour picker in the normal way.
4. Repeat for the other colour box.
5. Select the position at which the colour shading starts by dragging the colour boxes along the gradient control.
6. Select the centre for the colour spot by entering X and Y co-ordinates. These are entered as 0 to 100% with 0, 0 representing the top left of the drawing when it is filled.

7. Click OK to apply the colour.

How to select a colour with a liner gradient

1. Select the fill mode as gradient.
2. Select the type as linear.
3. Click the one of the small colour boxes on the gradient control above the colour circle and select its colour using the colour picker in the normal way.
4. Repeat for the other colour box.
5. Select the position at which the colour shading starts by dragging the colour boxes along the gradient control.
6. Select the angle at for the colour gradient.
7. Click OK to apply the colour.

How to add additional colours when using a gradient

By default a gradient uses two colours. However, if necessary, it is possible to add additional colours up to a maximum of eight.

1. Select the fill mode as gradient.
2. Click and drag the new colour box icon to the left of the gradient control onto the gradient control.
3. Select the colour for the new colour box in the normal way.
4. Select the position at which the colour shading starts by dragging the colour box along the gradient control.

How to remove additional colours

1. Click the colour box to be removed and drag it over the new colour box icon.

Gradient Colour Performance Considerations

The use of gradient colours is very attractive but, when using them, it is important to note that the speed at which they rendered will depend on the choice of graphic adapter and the colour configuration.

- Make the Step parameter us large as possible. The drawing performance depends directly on the size of the Step. For example a step of 4 will render twice as fast as a step of 2.
- When using a linear gradient, the angles 0, 90, 180 or 270 are most efficient.
- The radial gradient is more time consuming than linear particularly if the X and Y co-ordinates aren't close to the object centre (50%, 50%)
- The number of colours has no impact on the drawing performance.

Using the Colour Palette to Colour a Drawing

The Colour Palette may be used to colour a drawing directly or by changing the colours using the drawing's properties box.

To change a drawing's colour directly

1. Open the Colour Palette.
2. Select a drawing by clicking on it with the left mouse button. The palette will display the drawing's current colours in the colour selection area.
3. To change the colour of the fill (or background of text) click on the new colour with the left mouse button.
4. To change the colour of the border (or text) click on the new colour with the right mouse button.

If you select more than one drawing, the colour changes will be applied to all the selected drawings.

To change a colour in a drawing properties box

1. Display the drawing's properties box.

2. Click in the colour to change - the colour palette will be displayed.
3. Click on the colour in the palette with the left mouse button
4. If you want the colour to blink select the second colour with the right mouse button

To copy the colours from an existing drawing

1. Open the Colour Palette.
2. Click on the drawing element from where the colours are to be copied. Subsequent drawings will be initially created using these colours.

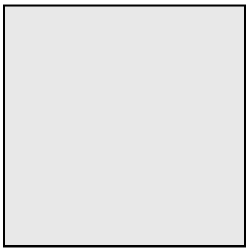
Drawing a Shape

Changing the Fill Pattern of a Shape | Changing the Appearance of a Shape | Drawing a ½ or ¼ Ellipse | Changing the Corner Radius of a Rounded Rectangle | Changing an Arc or Pie Shape

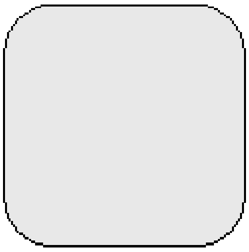
The draw tools, rectangle, rounded rectangle, ellipse, 1/2 ellipse, 1/4 ellipse, arc and pie all produce variations of the Shape drawing element. The appearance in a mimic is determined by the Shape's properties. The fundamental procedure to draw them all is the same.

1. Click on the relevant tool on the Draw toolbar.
2. Position the cursor on the screen where the drawing is to start.
3. Drag the cursor until the drawing is of the required size.
4. Release the mouse button.

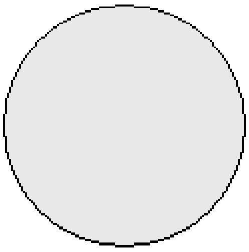
Once you have drawn the shape its properties may be displayed and by double clicking anywhere within its boundary.



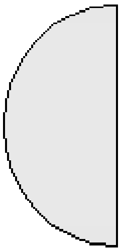
Rectangle



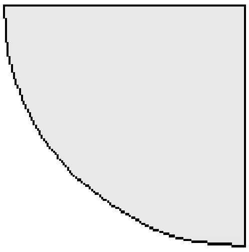
Rounded Rectangle



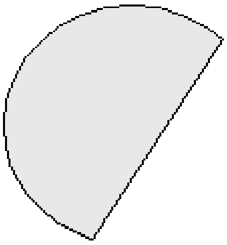
Ellipse



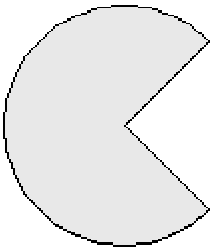
1/2 Ellipse



1/4 Ellipse



Arc

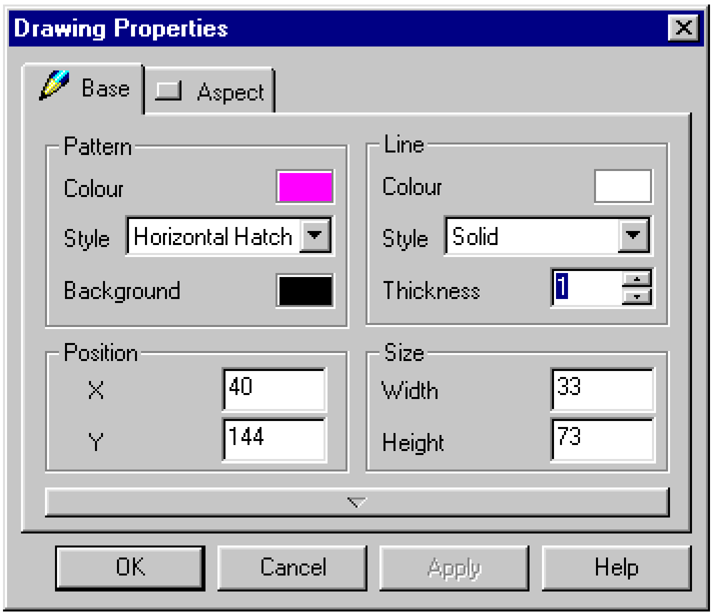


Pie

If the Ctrl button is held down whilst a Shape is drawn, the height and width will be the same.

Changing the Fill Pattern of a Shape

The fill style of a shape may be changed from the Base tab in the drawing properties box.



Drawing Properties

Base | Aspect

Pattern
Colour
Style: Horizontal Hatch
Background

Line
Colour
Style: Solid
Thickness: 1

Position
X: 40
Y: 144

Size
Width: 33
Height: 73

The following styles are available.

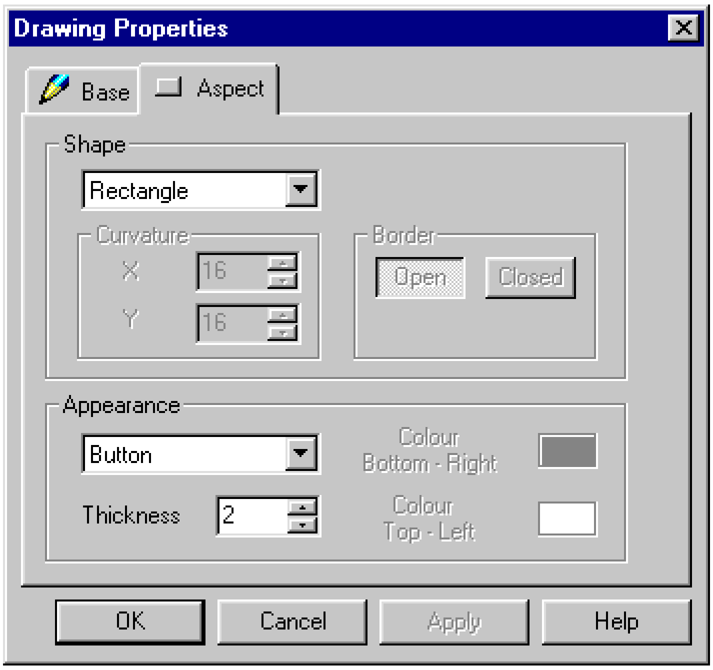
- ■ Solid
- □ Transparent
- ▨ Diagonal + 45
- ▩ Diagonal 45 & -45
- ▧ Diagonal –45
- ▥ Vertical
- ▤ Horizontal
- ▦ Horizontal & Vertical

With styles other than Solid or Transparent the pattern colour is selected using the Colour property and the fill colour using the Background property. With the Solid style the fill colour is selected using the Colour property.

It is also possible to change the border (line) style of a rectangle or ellipse. For information on changing the line style see the Drawing Lines topic.

Changing the Appearance of a Shape

The appearance of a shape may be changed using the Aspect tab in the drawing properties box.



Drawing Properties

Base | Aspect

Shape: Rectangle

Curvature: X 16, Y 16

Border: Open, Closed

Appearance: Button

Thickness: 2

Colour Bottom - Right

Colour Top - Left

The following options are available.

Regular. The shape is displayed as a normal rectangle. You can change the fill style and colour.

Button. The shape takes the appearance of a button. The fill colour and style is fixed. The border size can be changed using the Thickness property.

Coloured button. The shape takes the appearance of a coloured button. You can change the fill style and colour and the border size colour.

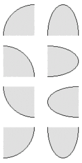
Relief. The shape takes a sunken appearance. You can change the fill style and colour and the border size and colour.

Inverse Relief. The shape takes a raised appearance. You can change the fill style and colour and the border size and colour.

Shadow. The shape is displayed as a normal rectangle with a coloured shadow. You can change the fill style and colour and the shadow width and colour.

Drawing a ½ or ¼ Ellipse

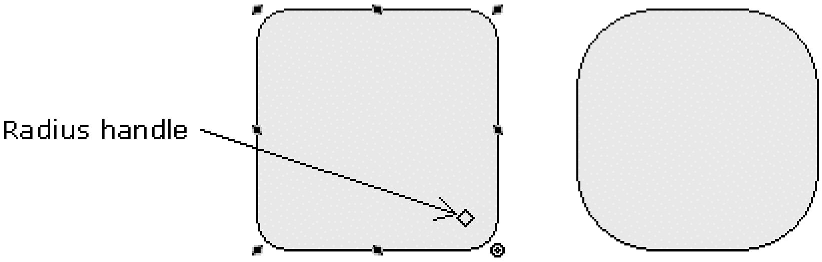
The method to draw a 1/2 ellipse or 1/4 ellipse is the same as for a drawing a full ellipse except that the point at which the drawing is started will determine the orientation of the drawing element.

Start at	**To draw**
Bottom left	
Bottom right	
Top left	
Top left	

The fill styles for a ¼ or ½ ellipse are the same as available for a rectangle.

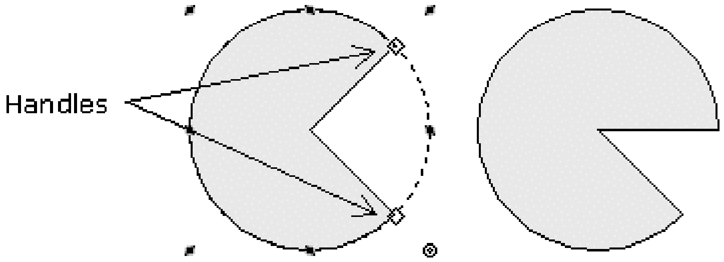
Changing the Corner Radius of a Rounded Rectangle

You can change the corner radius of a rounded rectangle either from the properties box or by clicking and dragging the radius handle that is displayed when the drawing is selected.



Changing an Arc or Pie Shape

You can change the segment of an Arc and the angle of the sector on the Pie either from the properties box or by clicking and dragging the handles displayed when the drawing is selected.



How to Draw a Line

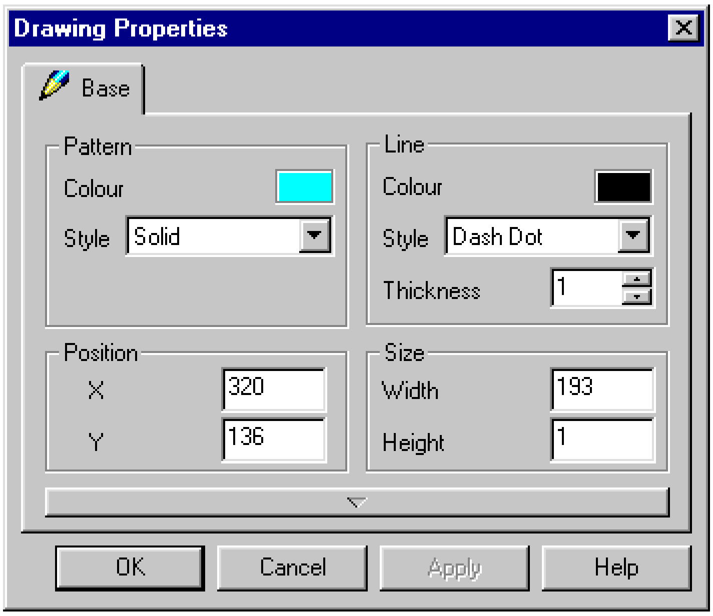
Selecting the Line Style

1. Select the Line tool from the Draw toolbar.
2. Position the cursor on the screen where the drawing is to start.
3. Drag the cursor until the line is of the required length.
4. Release the mouse button.

If the Ctrl button is held down whilst a line is drawn it will always be horizontal, vertical or at 45 degrees depending on which is nearest to the co-ordinates.

Selecting the Line Style

The line style, colour and width may be changed from the drawing properties box.



Pattern
Colour 🟦
Style Solid

Line
Colour ■
Style Dash Dot
Thickness 1

Position
X 320
Y 136

Size
Width 193
Height 1

The line style may be selected from the following options.

────────────	Solid
— — — — — — — —	Dashed
--------------------------------	Dotted
– - – - – - – - – - – - – - – -	Dash Dot
–--–--–--–--–--–--–--–	Dash Dot Dot

The line styles other than solid are only available for line widths of 1. The line styles may also be applied to polylines and the border of solid drawing elements such as rectangles and polygons.

The Pattern properties do affect the appearance of the line.

How to Draw a Polyline, Polygon, Bézier Curve or Bézier Shape

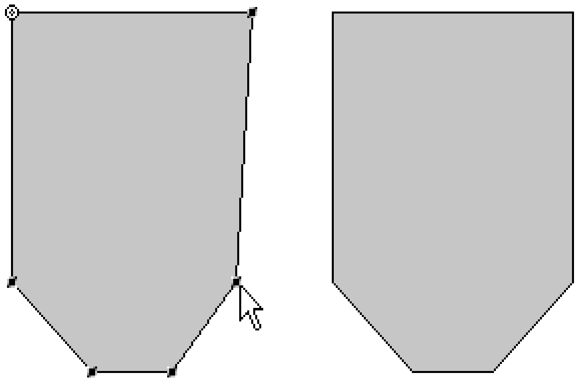
Re-shaping a Polyline or Polygon | Re-shaping a Bézier Curve | Adding or Deleting a Node
Drawing an Ellipse or Other Shape as a Polygon | Transforming a Bézier Curve into a Polygon

1. Select the Polyline, Polygon, Bézier curve or Bézier shape button from the Draw toolbar.
2. Position the cursor on the screen where the drawing is to start and click and release the mouse button.
3. Move the cursor to the next node and click and release the mouse button.
4. Repeat step 4 until the last but one node is drawn.
5. Move the cursor to the last corner and click the right button to complete the drawing.

The fill pattern of a polygon or Bézier shape may be changed in the same way as for a rectangle or ellipse. See the topic How to Draw a Rectangle for further information.

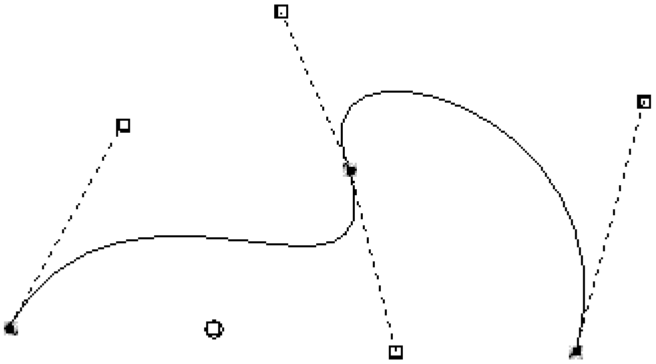
Re-shaping a Polyline or Polygon

When a polygon or polyline is selected a handle is displayed at each node. These handles may be used in the same way as for a normal drawing element except that instead of re-sizing the drawi



Re-shaping a Bézier Curve

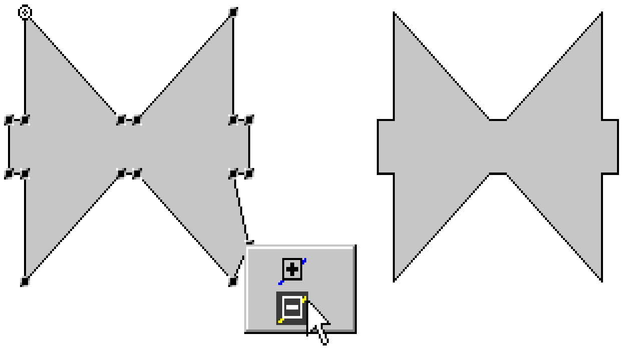
The nodes, which define the shape of a Bézier curve, may be moved in the same way as for a polyline or polygon. In addition each node has a tangent and associated handle which may be used to reshape the curve. The tangents are anchored at one end to each node and are parallel to the curve where they join it. By moving the tangent handles you adjust the angle at which the curve intersects with each node and its radius.



Adding or Deleting a Node

It is sometimes necessary to add or delete a node from a polyline, polygon, Bézier curve or Bézier shape. For example you may have made an error when drawing the shape.

To add or delete a node, first select the drawing element and then right click on either the node to be deleted, or a node adjacent to where the new node is to be added. A pop up menu will be displayed. Selecting the minus command will result in the node being deleted. Selecting the plus command will add a new node half way between the selected node and the next node in the shape.

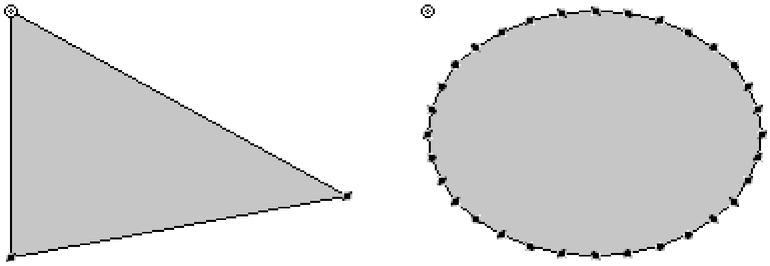


Drawing an Ellipse or Other Shape as a Polygon

You can draw any of the solid shapes, such as an ellipse, as a Polygon. This can be useful when you want to use the path animation.

To draw a shape as a Polygon:

1. Draw a polygon the height and width that you want the final shape to be. The number of nodes is not important.
2. Display the Polygon's properties box and select the Aspect tab.
3. In the shape drop down list box select the new appearance for the Polygon. You can select any of the normal solid shape types. (Rectangle, Rounded rectangle, Ellipse, 1/2 Ellipse, 1/4 Ellipse, Arc or Pie).
4. Click OK to confirm the transformation.

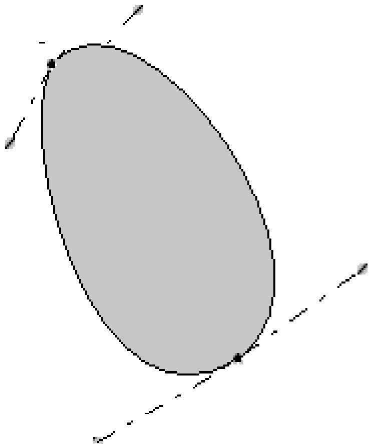


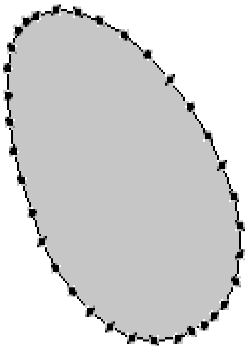
Transforming a Bézier Curve into a Polygon

You can convert a Bézier curve into a Polygon. This can be useful when you want to use the path animation.

To transform Bézier curve into a Polygon:

1. Draw a Bézier curve and adjust it to the final shape.
2. Display the Bézier curve's properties box and select the Aspect tab.
3. In the shape drop down list box select Polygon.
4. Click OK to confirm the transformation.





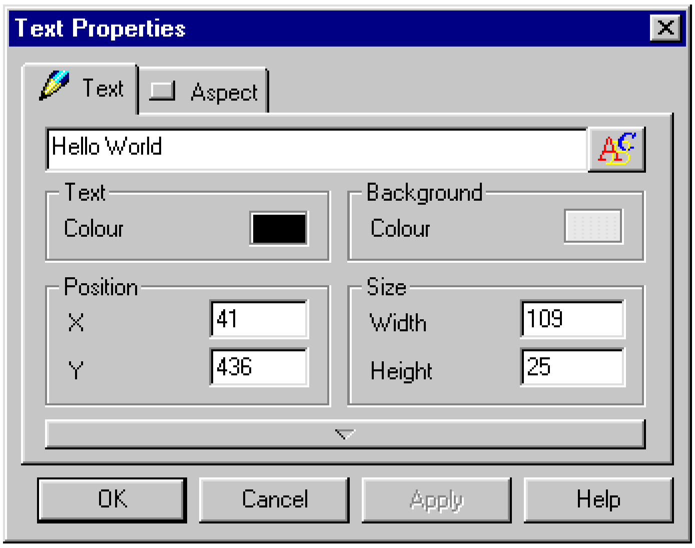
Inserting Text

Changing the Font | Changing the Appearance of the Support Rectangle | Using the Do Not Auto Size Property | Resizing Text by Dragging

Text is created in much the same way as any other drawing element except, as part of the process, the text string to be displayed is entered. When text is inserted its dimensions are supplied by a supporting rectangle. This rectangle may or may not be visible depending on the appearance chosen for the text.

1. Select the Text command from the Draw toolbar.
2. Position the cursor on the screen where the text is to start.
3. Type in the text to display.
4. Press the Enter key on the keyboard to confirm the text.

The text will initially appear with a default font and the colours currently selected in the palette. The text may be edited and other properties changes from the text properties box.



Hello World

Text Colour: ■

Background Colour: □

Position X: 41

Position Y: 436

Size Width: 109

Size Height: 25

Changing the Font

The font in which the text is displayed may be changed by clicking on the font tool in the text properties box. You can display text in any font installed on the host PC.

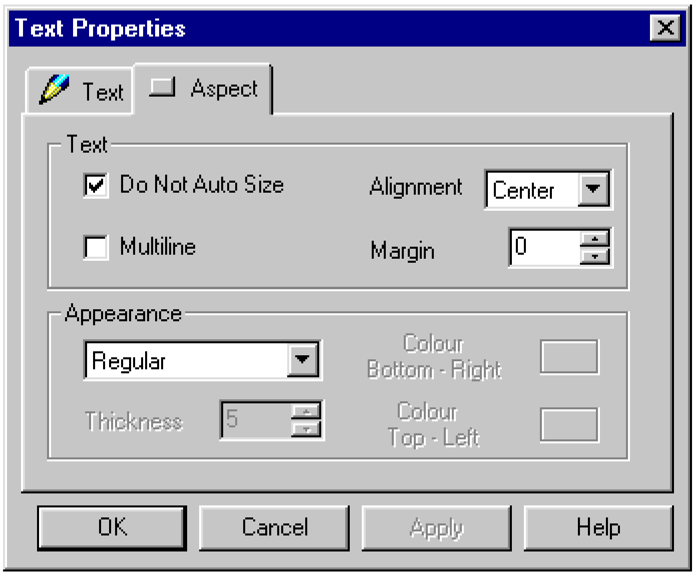
! Be careful when selecting a font that it will be available on the PC on which the application will finally run.

As long as you do not have the property Do Not Auto Size selected the support rectangle will automatically adjust to suit the size of the text.

The colour of the text may be selected from any of those in the palette.

Changing the Appearance of the Support Rectangle

The support rectangle is the rectangular area surrounding a text string. Normally this is transparent but it can be coloured and given one of a number of different styles. The style is selected using the appearance properties in the Aspect tab of the text properties box.



Text

☑ Do Not Auto Size

Alignment: Center

☐ Multiline

Margin: 0

Appearance

Regular

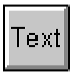
Thickness: 5

Colour Bottom - Right

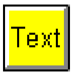
Colour Top - Left

The following options are available.

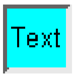
Regular. The default appearance for text. The background fill can be transparent or coloured.



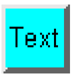
Button. The text takes the appearance of a button. The fill colour is fixed. The border size can be changed using the Thickness property.



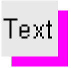
Coloured button. The text takes the appearance of a coloured button. You can change the fill colour and the border size.



Relief. The text takes a sunken appearance. You can change the fill colour and the border size and colour.



Inverse Relief. The text takes a raised appearance. You can change the fill colour and the border size and colour.



Shadow. The text support rectangle is displayed with a coloured shadow. You can change the fill colour and the shadow width and colour.

Using the Do Not Auto Size Property

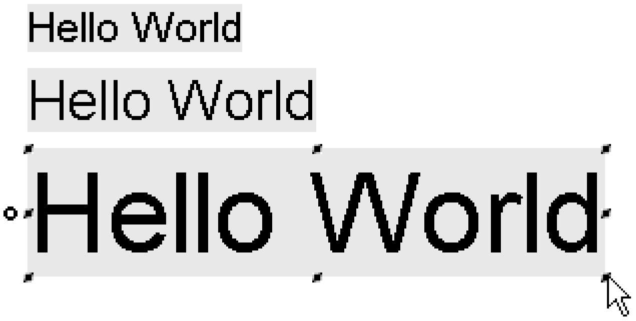
Selecting the Do Not Auto Size property has the following effects.

- If you change the size of the font the support rectangle does not change (and vice versa)
- The alignment of the text can be selected from centre (default) left or right.
- A margin, in pixels, can be applied to the side to which the text is aligned.
- The multiline property can be selected. Text that is too long to be displayed on one line will wrap onto further lines. The text is automatically broken at the most convenient space.

Resizing Text by Dragging

Resizing text is another way of changing the font size. If the Do Not Auto Scale property is not selected the point size of the font is changed automatically to fit the support rectangle.

1. Select the text
2. Click and drag one of the corner handles to its new position. The width and height of the support rectangle and the point size of the font will change simultaneously.



Inserting Images

How to Insert an Image | Using the Transparency Property | Using Animated GIF and AVI Files

The HMI supports the use of images in BMP, JPG, WMF. GIF and animated GIF formats. It also supports multimedia files in AVI format. The use of images can considerably enhance the appearance of your application, there are however two points which you must always take into consideration.

- Images, particularly those with large numbers of colours, use considerable amounts of memory.
- Images frequently use different colour palettes. If you display two or more images at the same time with different palettes, and the graphic adapter you are using does not have sufficient colours to display them, then you will get bizarre effects with the displayed colours changing.

Before you can insert an image you must move it to the correct directory using the Explorer or another file management tool. The image must reside in either:

- The Bitmap directory of your project.
- The Bitmap directory of any libraries you have created

An image is inserted as a reference. That is, the image file name is inserted in the mimic definition file. This means that, even when it has been inserted into a mimic, Paint or any other drawing package may still edit an image. Any changes made will be visible the next time the mimic is opened.

How to Insert an Image

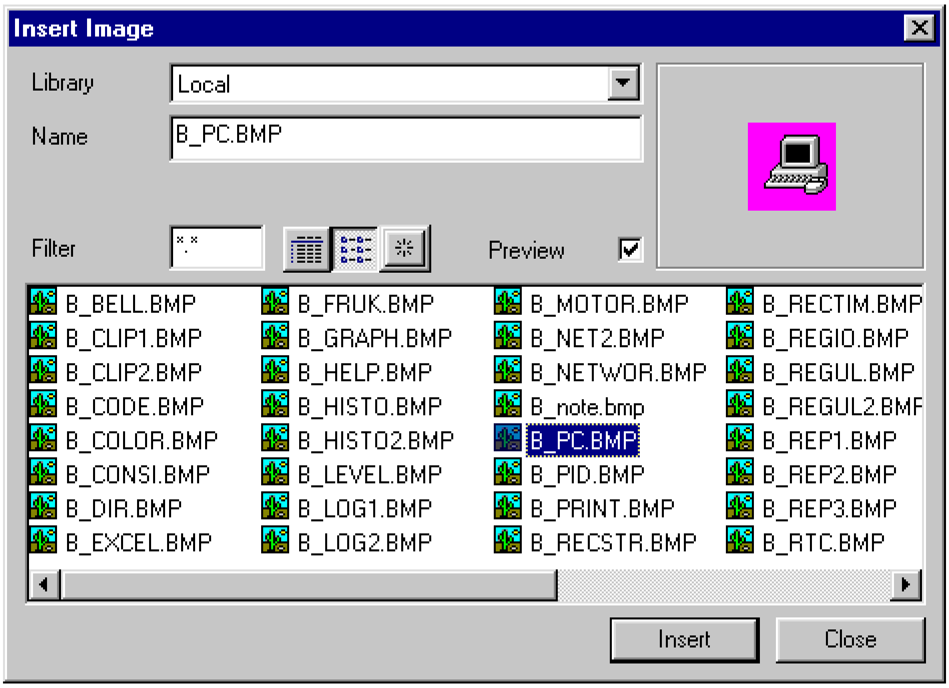
1. Click on the Image button in the Draw toolbar. A box will be displayed containing a list of available images and an area in which they may be previewed.
2. Select the library in which the image is located.
3. Select the name of the required image from the list. If necessary a lexical filter may be entered to reduce the number of images displayed in the list.
4. Click the Insert button to insert the image. The image will be placed at the top left hand corner of the mimic. Drag the image to the desired location.

Or

1. Click on the name of the image in the list and drag it to the required location in the mimic.

Or

1. Click on the thumbnail of the selected image and drag it to the required location in the mimic.



Insert Image

Library: Local

Name: B_PC.BMP

Filter: *.*

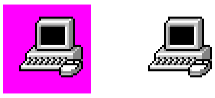
Preview ☑

B_BELL.BMP
B_CLIP1.BMP
B_CLIP2.BMP
B_CODE.BMP
B_COLOR.BMP
B_CONSI.BMP
B_DIR.BMP
B_EXCEL.BMP
B_FRUK.BMP
B_GRAPH.BMP
B_HELP.BMP
B_HISTO.BMP
B_HISTO2.BMP
B_LEVEL.BMP
B_LOG1.BMP
B_LOG2.BMP
B_MOTOR.BMP
B_NET2.BMP
B_NETWOR.BMP
B_note.bmp
B_PC.BMP
B_PID.BMP
B_PRINT.BMP
B_RECSTR.BMP
B_RECTIM.BMP
B_REGIO.BMP
B_REGUL.BMP
B_REGUL2.BMF
B_REP1.BMP
B_REP2.BMP
B_REP3.BMP
B_RTC.BMP

Using the Transparency Property

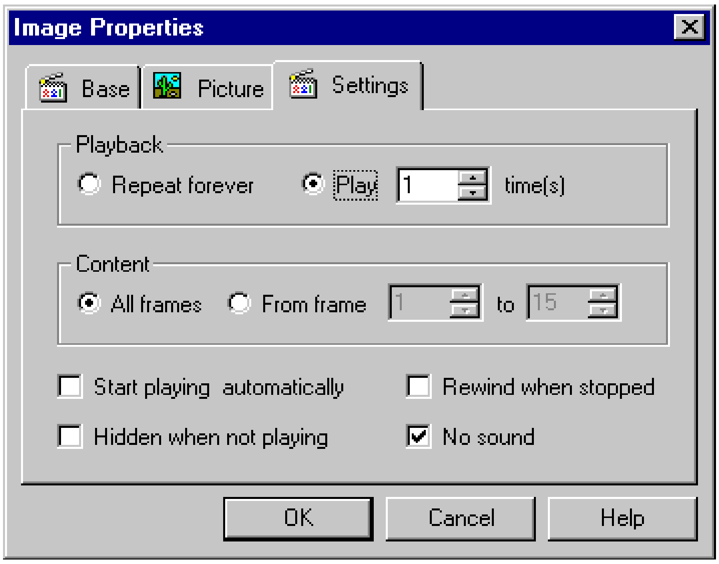
It is possible to define one of the colours that appears in each image as transparent. That is, you can see the background of the mimic or any drawing elements under it through any area of that colour.

1. Display the Image properties box by double clicking on it.
2. Tick the Transparent colour tick box. The cursor will change shape to a small circle.
3. Click on the area of the image that contains the colour to be transparent.
4. Click the OK button to confirm the change.



Using Animated GIF and AVI Files

Animated GIF or AVI files produce an animated image by playing a number of frames at a predetermined interval. The number of frames and interval is fixed when the image is designed. Images of this type have an extra tab in the properties box to control the behaviour at run-time.



Playback

○ Repeat forever ● Play 1 time(s)

Content

● All frames ○ From frame 1 to 15

□ Start playing automatically □ Rewind when stopped

□ Hidden when not playing ☑ No sound

Playing the file automatically

If you select the *Start playing automatically* property the file will play as soon as the mimic is switched to run time.

Controlling play programmatically

You can control play using the scripting language and the PlayCondition property. For example using the MouseEnter and MouseLeave events you can start and stop play when the pointer moves over the image.

Controlling the number of times the file is played.

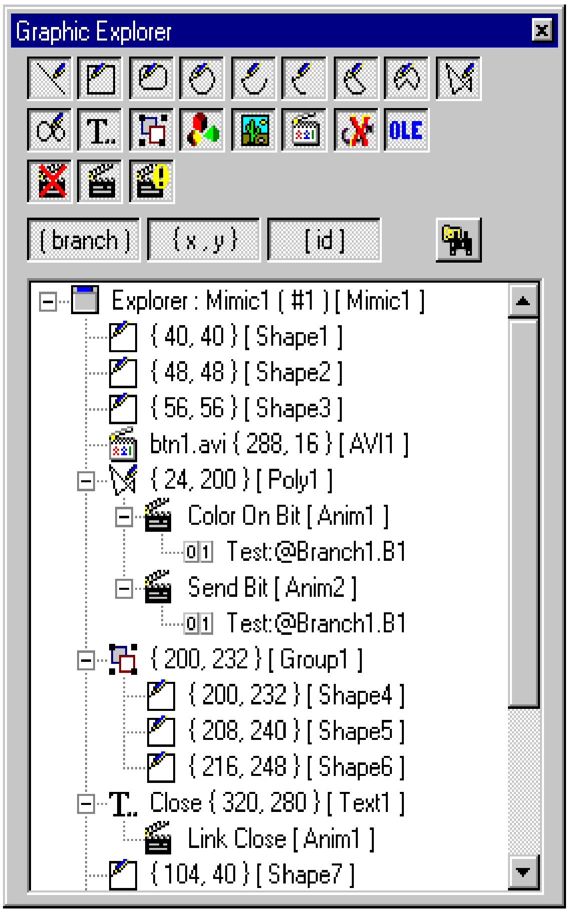
You can either chose to play the file continuously or a fixed number of times using the *Play continuously* and *Play* properties.

What happens when play stops?

If you select the *Rewind when stopped* property the first frame will be displayed when play is stopped, otherwise the frame displayed at the time play stops remains displayed. If the *Hidden when not playing* property is selected the image will not be visible when it is not playing.

The Graphic Explorer

The G :hical view of the elements forming a mimic. It is d r command.



```
Explorer : Mimic1 ( #1 ) [ Mimic1 ]
    { 40, 40 } [ Shape1 ]
    { 48, 48 } [ Shape2 ]
    { 56, 56 } [ Shape3 ]
    btn1.avi { 288, 16 } [ AVI1 ]
    { 24, 200 } [ Poly1 ]
        Color On Bit [ Anim1 ]
            Test:@Branch1.B1
        Send Bit [ Anim2 ]
            Test:@Branch1.B1
    { 200, 232 } [ Group1 ]
        { 200, 232 } [ Shape4 ]
        { 208, 240 } [ Shape5 ]
        { 216, 248 } [ Shape6 ]
    Close { 320, 280 } [ Text1 ]
        Link Close [ Anim1 ]
    { 104, 40 } [ Shape7 ]
```

The graphical explorer allows you to see:

- A list of all mimics currently open.
- For each mimic a list of drawing elements and the co-ordinates of their anchor point.
- For each drawing element, any animation that has been applied.
- For each animation, the name of any database variables and branch used.

In addition you may display and change the properties for any drawing element by double clicking on its entry in the Explorer.

The tools at the top of the explorer allow you to filter what drawing elements are displayed.

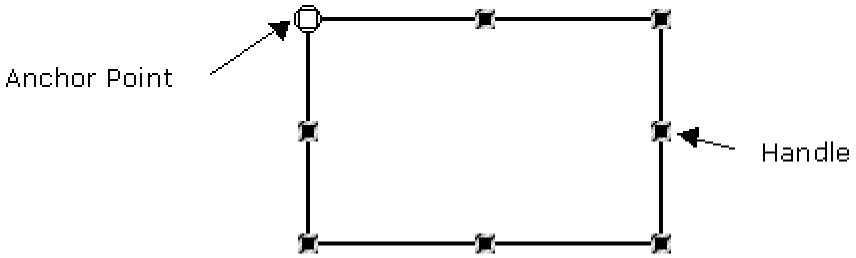
Selecting, Moving and Copying Drawing Elements

How to Select a Single Drawing Element | How to Select More Than One Drawing Element | Drawing Groups | Moving and Resizing | Copying and Pasting

A drawing element must be selected before you do anything to it. A selected drawing element is surrounded by a number of small blocks known as handles. An anchor point, about which certain drawing elements may be rotated, is also displayed.

How to Select a Single Drawing Element

1. Choose Select/Resize from the Draw toolbar.
2. Point to the drawing element and click.

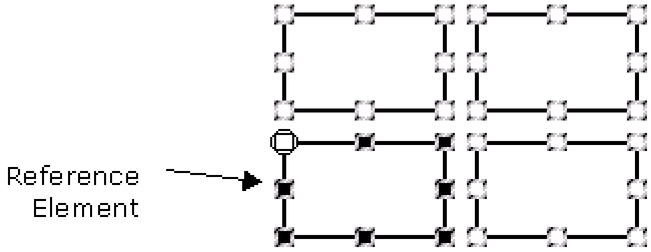


How to Select More Than One Drawing Element

1. Choose Select/Resize from the Draw toolbar.
2. Hold down the Shift button on the keyboard, point to each drawing element in turn and click.

Or

1. Choose Select/Resize from the Draw toolbar.
2. Point to one corner of an imaginary rectangle containing all the drawing elements to select. Drag the cursor to the diametrically opposite corner and release the mouse button. All drawing elements totally or partially within the rectangular area will be selected.



When selecting several drawing elements each one is surrounded by handles with a white centre except the last to be selected which has normal handles. This element is known as the reference and in certain operations is used to provide new properties for the other elements.

- You can use the commands on the Arrange toolbar to align and arrange them.
- You can drag them to a new location. Their relative positions remain the same.
- You can use the cut, copy and delete commands.
- You can create a drawing group.

! If you click anywhere outside of one of the selected drawing elements all the drawing elements are de-selected.

Drawing Groups

A drawing group is a collection of one or more drawing elements that behave as if they were one. For example when moving them within the mimic their relative positions remain the same although as a group they may be moved anywhere within the mimic. All the tools which operate on single drawing elements may also be used on a drawing group.

You cannot arrange or align individual drawing elements in a group.

To create a group

1. Select one or more drawing elements.
2. From the Drawing pop-up menu select the Group command.

Or

1. Select the Group tool from the Arrange toolbar.

To ungroup

1. Select a drawing group.
2. From the Drawing pop-up menu select the Ungroup command.

Or

1. Select the Ungroup tool from the Arrange toolbar.

Moving and Resizing

To move a drawing element

1. Select the drawing element.
2. Click anywhere within the drawing element and drag it to its new location.

You can drag the drawing element within the same, or to another mimic.

To nudge a drawing element

1. Select the drawing element.
2. Use the arrow keys on the keyboard to move the drawing. Each press moves the drawing one pixel.

Holding down the shift key and then using the arrow keys moves the drawing by one grid element.

To resize a drawing element

1. Select the drawing element.
2. Click and drag one of the drawing element handles to its new position. Clicking on one of the handles on the side of a drawing element will change its width. Clicking on one of the handles on the top or bottom of a drawing element change its height. Clicking on one of the corner handles will allow the width and height to be changed simultaneously.

To resize a polygon, polyline or bézier curve, group it, resize the group and ungroup it.

Copying and Pasting

To place a copy on the clipboard

1. Choose Select/Resize from the Draw toolbar.
2. Select one or more drawing element.
3. Click the Copy button on the Standard toolbar.

Or

1. Press Control C on the keyboard.

To cut and place a copy on the clipboard

1. Choose Select/Resize from the Draw toolbar.
2. Select one or more drawing element.
3. Click the Cut button on the Standard toolbar.

Or

1. Press Control X on the keyboard.

To duplicate

1. Choose Select/Resize from the Draw toolbar.
2. Press and hold the Control key on the keyboard.
3. Click on the drawing element and holding down the mouse button drag the cursor.
4. Release the mouse button. A copy of the drawing element will be inserted at the cursor location.

To paste the contents of the clipboard in a mimic

1. Click in the mimic to select it.
2. Click the Paste button on the Standard toolbar.

Or

1. Press Control V on the keyboard.

The contents of the clipboard will be pasted in the selected mimic. The location will be the same as that of the original drawing element.

To paste the contents of the clipboard at a specific location

1. Click in the mimic to select it.
2. Display the Mimic properties box.
3. Select the Paste here command.

The contents of the clipboard will be pasted at the current cursor position.

If the clipboard contains anything other than the HMI's native geometric drawing elements, the contents will be inserted as an OLE object.

Arranging Drawing Elements

Drawing elements are resized, moved, aligned, locked and grouped using the Arrange toolbar.

Click to resize the selected elements so that:

The width matches that of the reference.

The height matches that of the reference.

The width and height matches that of the reference.

Click to move the selected drawing elements:

Horizontally so that the left edge aligns with that of the reference.

Horizontally so that the right edge aligns with that of the reference.

Vertically so that the upper edge aligns with that of the reference.

Vertically so that the lower edge aligns with that of the reference.

Vertically so that their centre aligns with that of the reference.

Horizontally so that their centre aligns with that of the reference.

Horizontally so that they are evenly spaced.

Vertically so that they are evenly spaced.

Click to arrange the selected element so that:

It is in front of all overlapping elements.

It is behind all overlapping elements

It moves in front of the drawing element immediately in front of it.

It moves behind the element immediately under it.

Click to:

Lock a drawing element. Once locked you cannot change the drawing element's properties.

Unlock a drawing element.

Toggle (turn on and off) display of the drawing grid.

Toggle (turn on and off) the alignment of drawing elements to the grid.

Un-group the selected drawing group.

Group the selected drawing elements.

Click to flip a drawing element:

About its vertical axis. (1/2 and 1/4 ellipses, bitmaps and drawing groups)

About its horizontal axis. (1/2 and 1/4 ellipses, bitmaps and drawing groups)

To flip a single polyline, polygon or Bézier curve, group it, flip the group and then ungroup it.

Click to create a new polygon using Boolean operators by:

Adding the sum of the areas two drawing elements.

Taking the overlapping area of two drawing elements.

Subtracting the overlapping area of one drawing element from the other.

Click to:

Reverse the order of points in a polygon, polyline, Bézier curve or Bézier solid. Useful when using the Boolean graphic tools and the Path animation.

Using the Boolean Graphic Tools

Rules and Restrictions | Union of Two Drawing Elements | Intersection of Two Drawing Elements | Extraction of Two Drawing Elements

Using the Boolean Graphic Tools you can create a new polygon drawing element using the logical combination of two other drawing elements. The three tools available are Union, Intersection and Exclusion and are found on the Arrange toolbar.

Rules and Restrictions

The new polygon is calculated using a mathematical model and certain rules and restrictions apply.

Supported Drawing Elements

- Rectangle.
- Rounded rectangle.
- Ellipse or circle.
- 1/4 ellipse.
- 1/2 ellipse.
- Arc.
- Pie.
- Polygon.
- Bézier shape.

Drawing Element Intersections

The two drawing elements to be used in the combination must have two intersection points. If there are more than two intersections points the combination can fail or produce unexpected results.

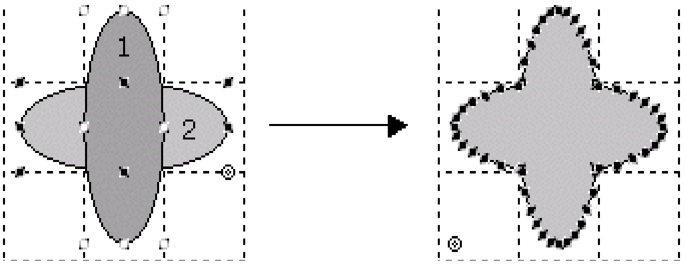
Drawing element orientation

This rule applies to the polygon and Bézier shape only.

The mathematical model requires the selected drawing to have a particular drawing orientation. That is the direction of the first point to the last point must be in an anti-clockwise direction. If you need to change the drawing orientation you can use the Reverse tool on the Arrange toolbar.

Union of Two Drawing Elements

The Union tool creates a polygon that is the sum of the areas two drawing elements.



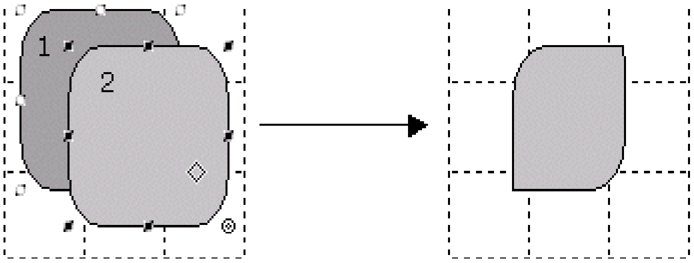
How to use the union tool

1. Select the first drawing element by clicking with the left mouse button.
2. Select the second drawing element holding the shift button so that both drawing elements are selected.

3. Select the Union tool from the arrange toolbar. A new polygon is created taking the colour from the second shape.
4. Click and drag the polygon to its location. The original drawing elements remain unchanged.

Intersection of Two Drawing Elements

The Intersection tool creates a polygon that is equivalent to the overlapping area of two drawing elements.

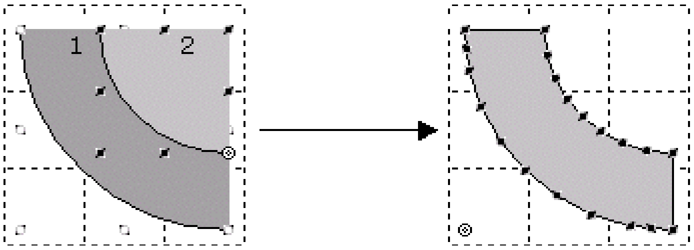


How to use the intersection tool

1. Select the first drawing element by clicking with the left mouse button.
2. Select the second drawing element holding the shift button so that both drawing elements are selected.
3. Select the Intersection tool from the arrange toolbar. A new polygon is created taking the colour from the second shape.
4. Click and drag the polygon to its location. The original drawing elements remain unchanged.

Extraction of Two Drawing Elements

The Extraction tool creates a polygon that is equivalent subtracting the overlapping area of one drawing element from the other.



How to use the extraction tool

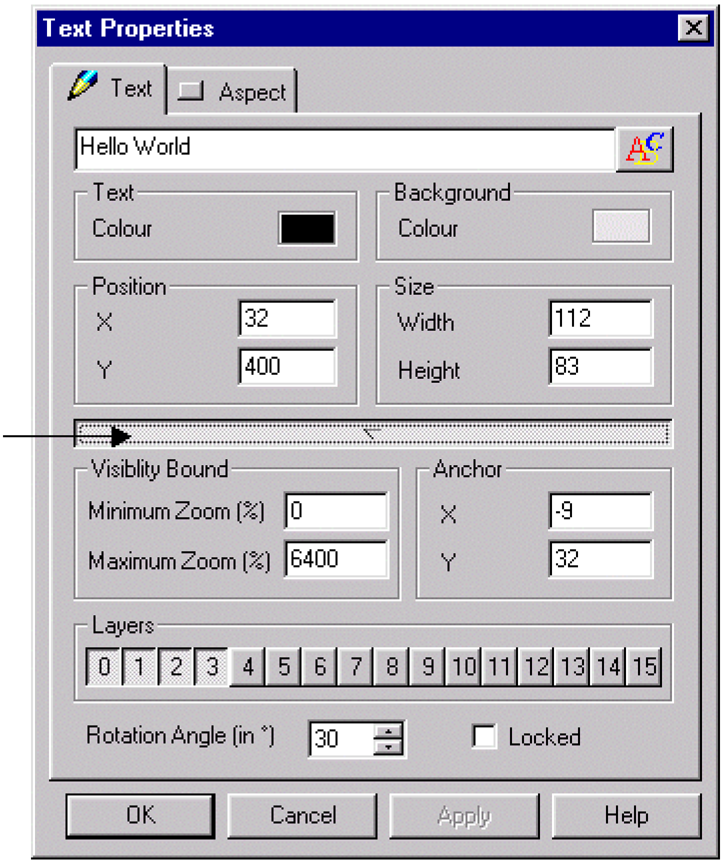
1. Select the first drawing element by clicking with the left mouse button.
2. Select the second drawing element holding the shift button so that both drawing elements are selected.
3. Select the Extraction tool from the arrange toolbar. A new polygon is created taking the colour from the second shape.
4. Click and drag the polygon to its location. The original drawing elements remain unchanged.

Advanced Properties

Attaching a Drawing Element to Specific Display Layers | Binding the Display of a Drawing Element to a Zoom Threshold | Locking and Unlocking | Rotating

Most of the drawing elements support additional advanced properties which are not normally displayed in the properties box. To display the advanced properties you can either:

- Select the Display.Advanced Mode command. The advanced properties are displayed every time you display a drawing element's properties.
- Click the down arrow button at the bottom of a property box. The box will change to show the advanced properties for that specific drawing element.



Text Properties

Hello World

Text Colour ■ Background Colour □

Position X 32 Y 400

Size Width 112 Height 83

Visiblity Bound Minimum Zoom (%) 0 Maximum Zoom (%) 6400

Anchor X -9 Y 32

Layers 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Rotation Angle (in °) 30

□ Locked

Click to toggle display of advanced properties

Attaching a Drawing Element to Specific Display Layers

Each mimic contains 16 display layers. At run-time you can select which of the layers are displayed either manually, using the Layer toolbar, or automatically according to defined criteria (the current User, process conditions etc.) The layers are completely independent of one another and you can choose to display one or more layers at any one time.

Each drawing element may be allocated to one or more layer. Any drawing elements not allocated to the selected layers are not visible in the mimic.

By default a mimic displays all layers. Likewise, when a drawing element is created it is allocated to all layers. Therefore if you do not intend to use layers there is nothing to do.

To allocate a drawing element to specific layers

1. Display the drawing element properties box.
2. Click on the down arrow button to display the advanced properties.
3. Select/de-select layers using layer buttons.(from 0 to 15).
4. Click OK to confirm the change.

Binding the Display of a Drawing Element to a Zoom Threshold

The visibility of drawing elements may be bound to a zoom range. That is, if the mimic is zoomed to a level outside of that range the drawing will not be visible.

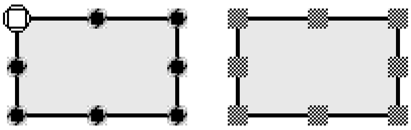
The main use of this technique is to de-clutter large or complex mimics. For example, if you have a mimic that is 1600 x 1200 pixels in size you might normally move around it using the scroll bars. You can also zoom out so that the entire mimic is visible. In this case you may now have so much activity on the screen that the most important information is no longer obvious. By binding some of the drawing elements so that they can only be seen between 0 and 100 (normal) view you can automatically reduce the amount of information visible so that the most important information is readily seen.

To lock the visibility of a drawing element to a zoom threshold

1. Display the drawing element properties box.
2. Click on the down arrow button to display the advanced properties.
3. Enter the minimum and maximum zoom level (range 0 to 6400%).
4. Click OK to confirm the change.

Locking and Unlocking

A drawing element that has been locked cannot be moved or changed in any way whilst in Edit mode. When a locked drawing element is selected its handles have a different appearance to a normal drawing element. In the picture below the drawing element on the right is locked whilst the one on the left is unlocked.



To lock a drawing element

1. Display the drawing element properties box.
2. Click on the down arrow button to display the advanced properties.
3. Tick the Locked tick box
4. Click OK to confirm the change.

Or

1. Select the drawing element.
2. Click the Lock button on the toolbar.

Or

1. Select the Lock command on the Drawing pop-up menu.

Rotating

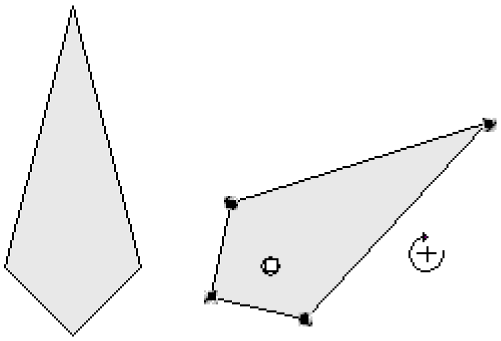
Rotation is the process of rotating a drawing element about its anchor point. You may rotate polygons, polylines and Bézier curves. You can also rotate text but only if the Do not auto size property is not selected and the Regular appearance is chosen.

To rotate a drawing element

1. Display the drawing element properties box.
2. Click on the down arrow button to display the advanced properties.
3. Enter the angle in the Rotation Angle field. A positive number is clockwise, a negative number anticlockwise.
4. Click OK to confirm the change.

Or

1. Select the Rotate tool from the Arrange toolbar. The cursor will change shape to a circle with an arrow head.



2. Click within the drawing element and drag the cursor. The angle of rotation is directly proportional to how far you move the cursor.
3. Release the mouse button when you have rotated the drawing to its new position.

The drawing element rotates about its anchor point. You can relocate the anchor point using the following method.

To move the anchor point

1. Choose Select/Resize from either the Draw toolbar.
2. Select the drawing element.
3. Move the cursor over the anchor point. The cursor will change shape to an anchor. Click and drag the anchor point to its new location.

What is an ActiveX Control?

Manually Registering an ActiveX Control

ActiveX controls are software components based on a common architecture that allows the interaction of software under Microsoft Windows. In order to use an ActiveX control you must incorporate it in another program that has been designed to be an ActiveX container. There are many programs that are ActiveX containers including the HMI. Communication between an ActiveX control and the program containing it is made using methods, events and properties.

A method is an action that an ActiveX can perform. It is used by the container program to control the behaviour of the ActiveX.

An event is a function used by ActiveX control to indicate a change to the application. For example an event can be generated when the mouse is clicked.

A property is information offered by the ActiveX to its container, for example it's size and position on the screen. Properties may be read only or read and write.

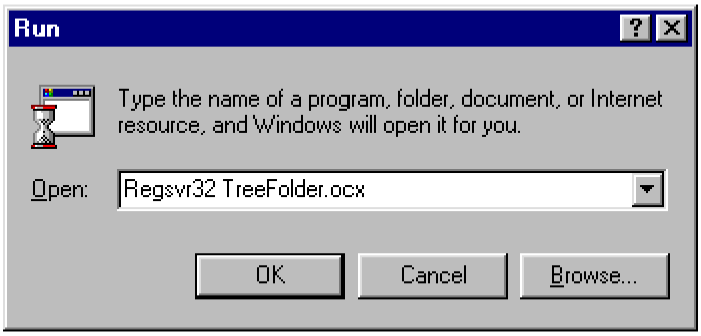
There is a vast range of ActiveX controls available from simple check boxes to complex graphical displays such as the Trend display provided by the HMI.

Manually Registering an ActiveX Control

Before you can use an ActiveX control it must be registered with the operating system. ActiveX controls supplied as part of an application are normally automatically registered when the application is installed. However if you want to install an ActiveX that has been supplied as a component use the following procedure.

1. Copy the ActiveX program file (*.ocx) to the Windows' System directory.
2. From the Start button select the Run command.
3. Type Regsvr32 followed by the name of the ActiveX control.
4. Select the OK command button.

A message will then be displayed indicating if the registration has been successful or not.



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: Regsvr32 TreeFolder.ocx

Authorising ActiveX Controls

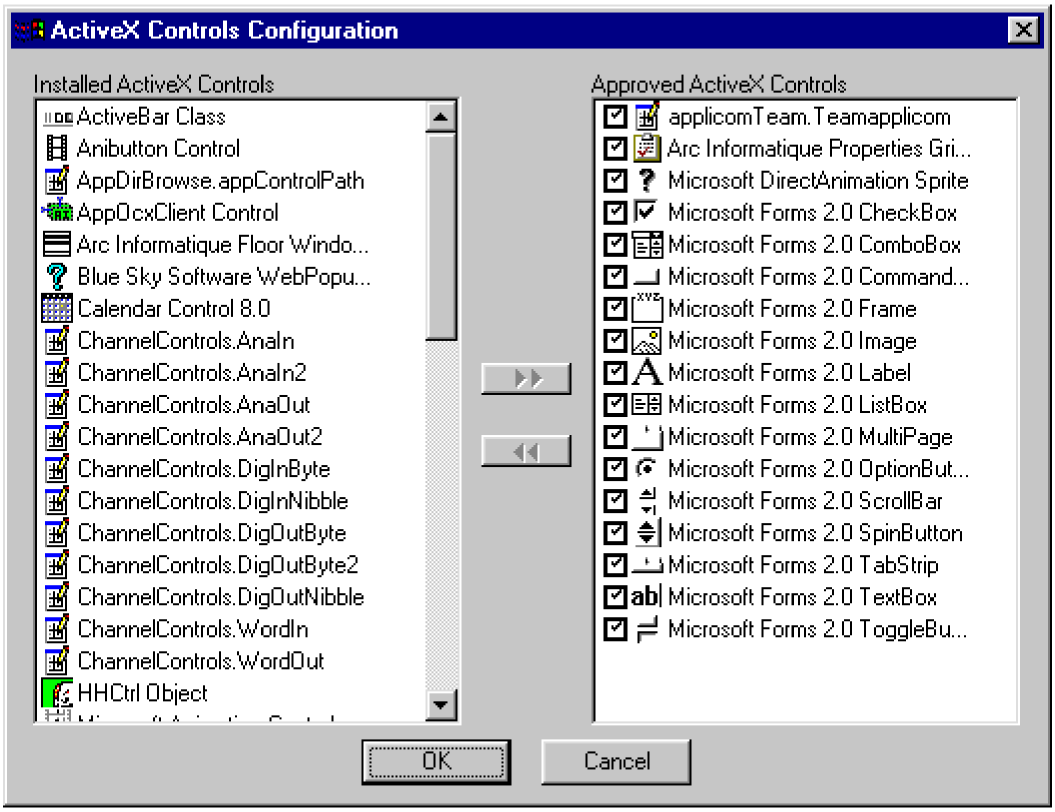
How to authorise an ActiveX control | How to add an ActiveX control to the Favourites menu

The ActiveX controls available on any one PC will depend on the application programs that have been installed. (Many programs automatically register ActiveX controls as part of the installation process without the user being aware of it) The quality and conformance of the ActiveX components to the Microsoft standard can vary considerably. For this reason before an ActiveX component can be used in the HMI it must first be authorised. The authorise process checks that the ActiveX control is compliant with the HMI and makes it available for use.

Be careful when selecting ActiveX controls for use with a project. The ActiveX control must be available on the target PC. Many ActiveX controls are supplied with other applications and cannot be freely distributed.

How to Authorise an ActiveX Control

1. Select the Tools.Preferences.ActiveX controls command from the menu. The Authorise ActiveX box is displayed.
2. Select the names of one or more ActiveX controls from the left pane. Click the >> command button to copy the names to the right hand pane.
3. Click the OK button.


ActiveX Controls Configuration
Installed ActiveX Controls
Approved ActiveX Controls
OK
Cancel

Installed ActiveX Controls:

ActiveBar Class
Anibutton Control
AppDirBrowse.appControlPath
AppOcxClient Control
Arc Informatique Floor Windo...
Blue Sky Software WebPopu...
Calendar Control 8.0
ChannelControls.AnaIn
ChannelControls.AnaIn2
ChannelControls.AnaOut
ChannelControls.AnaOut2
ChannelControls.DigInByte
ChannelControls.DigInNibble
ChannelControls.DigOutByte
ChannelControls.DigOutByte2
ChannelControls.DigOutNibble
ChannelControls.WordIn
ChannelControls.WordOut
HHCtrl Object

Approved ActiveX Controls:

☑ applicomTeam.Teamapplicom
☑ Arc Informatique Properties Gri...
☑ Microsoft DirectAnimation Sprite
☑ Microsoft Forms 2.0 CheckBox
☑ Microsoft Forms 2.0 ComboBox
☑ Microsoft Forms 2.0 Command...
☑ Microsoft Forms 2.0 Frame
☑ Microsoft Forms 2.0 Image
☑ Microsoft Forms 2.0 Label
☑ Microsoft Forms 2.0 ListBox
☑ Microsoft Forms 2.0 MultiPage
☑ Microsoft Forms 2.0 OptionBut...
☑ Microsoft Forms 2.0 ScrollBar
☑ Microsoft Forms 2.0 SpinButton
☑ Microsoft Forms 2.0 TabStrip
☑ Microsoft Forms 2.0 TextBox
☑ Microsoft Forms 2.0 ToggleBu...

How to add an ActiveX Control to the Favourites Menu

1. Select the Tools.Preferences.ActiveX controls command from the menu. The Authorise ActiveX box is displayed.
2. In the right hand pane select the tick box adjacent to each of the ActiveX controls that you want to appear in the Favourites menu.

Using an ActiveX Control in a Mimic

How to paste an ActiveX control into a mimic | Displaying and Changing an ActiveX Control's Properties

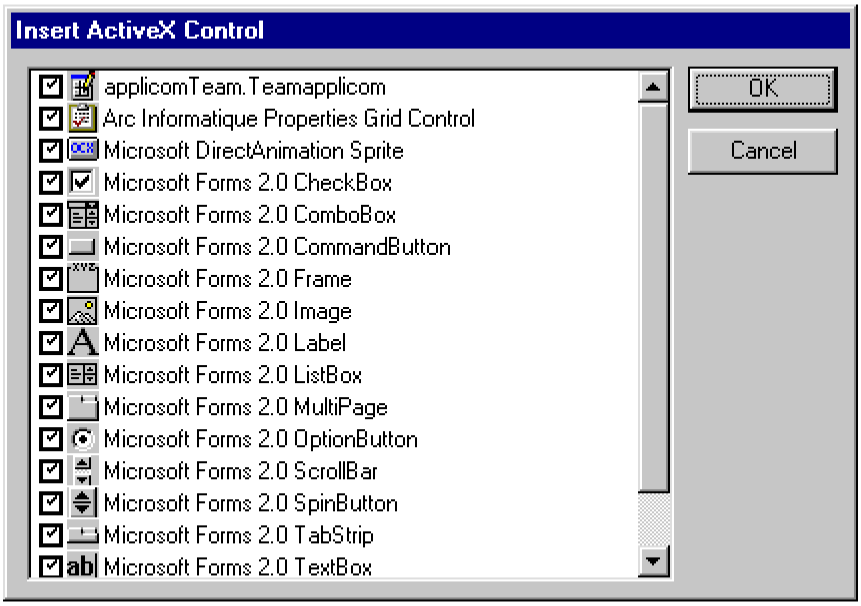
You can paste any of the authorised ActiveX controls into an HMI mimic. When you paste a control into a mimic it is automatically given a name by which it is then referenced. You can view and change the name using the properties list (see below).

The control becomes active as soon as the mimic is switched to run mode although, in general, for an ActiveX control to do something useful you will also have to write some software using the scripting language. For example if you insert a command button you will need to tell the HMI what to do when the button is clicked.

Once a control has been pasted into a mimic you can move it or size it the same as you would with a native drawing element.

How to Paste an ActiveX Control into a Mimic

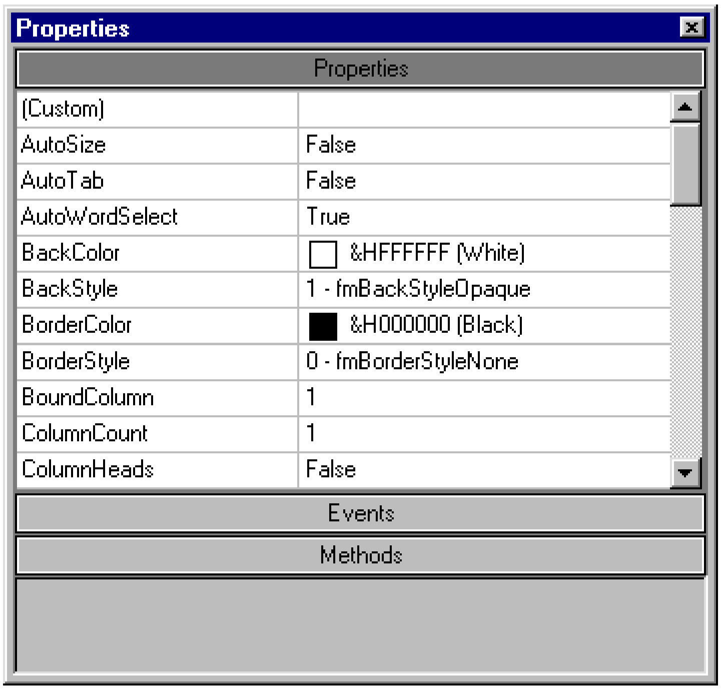
1. Open the mimic and switch it to draw mode.
2. From the menu select the Insert.ActiveX control command. A box is displayed listing the authorised ActiveX controls.
3. Select the control you require and confirm by clicking the OK button.



☑ applicomTeam.Teamapplicom
☑ Arc Informatique Properties Grid Control
☑ Microsoft DirectAnimation Sprite
☑ Microsoft Forms 2.0 CheckBox
☑ Microsoft Forms 2.0 ComboBox
☑ Microsoft Forms 2.0 CommandButton
☑ Microsoft Forms 2.0 Frame
☑ Microsoft Forms 2.0 Image
☑ Microsoft Forms 2.0 Label
☑ Microsoft Forms 2.0 ListBox
☑ Microsoft Forms 2.0 MultiPage
☑ Microsoft Forms 2.0 OptionButton
☑ Microsoft Forms 2.0 ScrollBar
☑ Microsoft Forms 2.0 SpinButton
☑ Microsoft Forms 2.0 TabStrip
☑ Microsoft Forms 2.0 TextBox

Displaying and Changing an ActiveX Control's Properties

When an ActiveX is pasted into a mimic it will initially appear with default settings specified by the developer. These may be totally different to those you require. You can display and modify the properties using the properties list box displayed from the Display.Properties list command on the menu. Note that this is different to the properties box that is displayed when you double click on a native drawing element.



(Custom)	
AutoSize	False
AutoTab	False
AutoWordSelect	True
BackColor	□ &HFFFFFF (White)
BackStyle	1 - fmBackStyleOpaque
BorderColor	■ &H000000 (Black)
BorderStyle	0 - fmBorderStyleNone
BoundColumn	1
ColumnCount	1
ColumnHeads	False

Some properties may be changed by typing in a new value. Some properties may be selected from a list that is displayed when you click in the field. Other properties may be read only – this is indicated by the value appearing in grey. Once you have entered a new value for a property the change becomes active immediately.

The Properties List also displays the Events and Methods available for an object. Selecting a method will display its syntax in the area at the bottom of the Properties List. Selecting an Event will open the script editor where you can type or edit the event script.

What is a Java Bean?

A Java Bean is a portable, platform-independent component written in the Java programming language. It enables developers to write reusable components that will run on any platform supporting Java technology.

In order to use a Java Bean you must incorporate it in another program that has been designed to be a Bean container. There are many programs that are Bean containers including the HMI. Communication between a Bean and the program containing it is made using methods, events and properties.

A method is an action that a Bean can perform. It is used by the container program to control the behaviour of the Bean.

An event is a function used by a Bean to indicate a change to the application. For example an event can be generated when the mouse is clicked.

A property is information offered by the Bean to its container, for example its size and position on the screen. Properties may be read-only or read and write.

Using a Java Bean in a Mimic

How to Paste a Bean into a Mimic | Displaying and Changing a Beans Properties | How to Link a Bean to a RTDS Variable

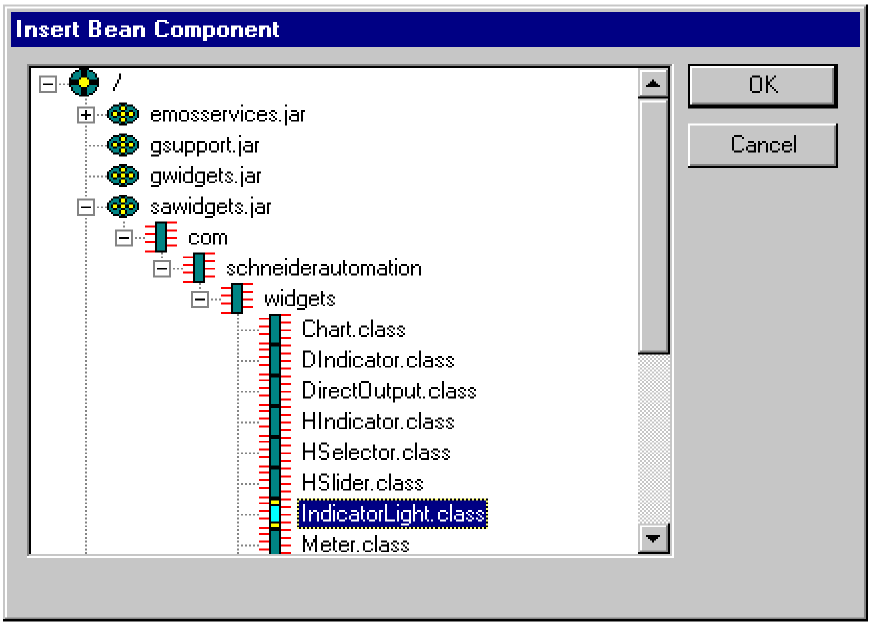
The source (.jar) files for any Java Bean you wish to use in your project must be located in the JAR project folder. When you paste a Bean into a mimic it is automatically given a name by which it is then referenced. You can view and change the name using the properties list (see below).

The Bean becomes active as soon as the mimic is switched to run mode. However, for a Bean to do something useful, you will have to do some further configuration such as linking one of the Bean's properties to the value of a variable from the RTDS.

Once a Bean has been pasted into a mimic you can move it or size it in the same as you would with a native drawing element.

How to Paste a Bean into a Mimic

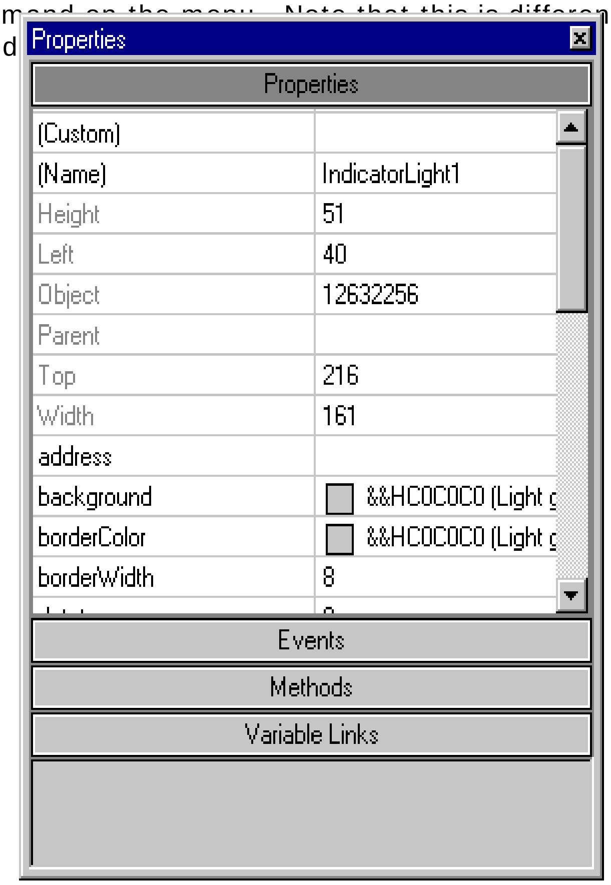
1. Open the mimic and switch it to draw mode.
2. From the menu select the **Insert.Bean** control command. A box is displayed listing the available Beans.
3. Select the Bean you require and confirm by clicking the *OK* button.



Before a Java Bean can appear in a mimic the Java Virtual Engine must be running. After a project is started, the first time you paste a Bean into a mimic, or open a mimic that contains a Bean, the Java Virtual Engine will automatically start. This may take anything between a few seconds and a few minutes depending on the PC you are using. During this time the pointer icon will change to the hourglass symbol indicating that the software is busy. The Java Virtual Engine then runs until the project is closed.

Displaying and Changing a Bean's Properties

When a Bean is pasted into a mimic it will initially appear with default settings specified by the developer. These may be totally different to those you require. You can display and modify the properties using the properties list box displayed from the **Display.Properties** list command on the menu. Note that this is different to the properties box that is displayed when you d



(Custom)	
(Name)	IndicatorLight1
Height	51
Left	40
Object	12632256
Parent	
Top	216
Width	161
address	
background	□ &&HC0C0C0 (Light g
borderColor	□ &&HC0C0C0 (Light g
borderWidth	8

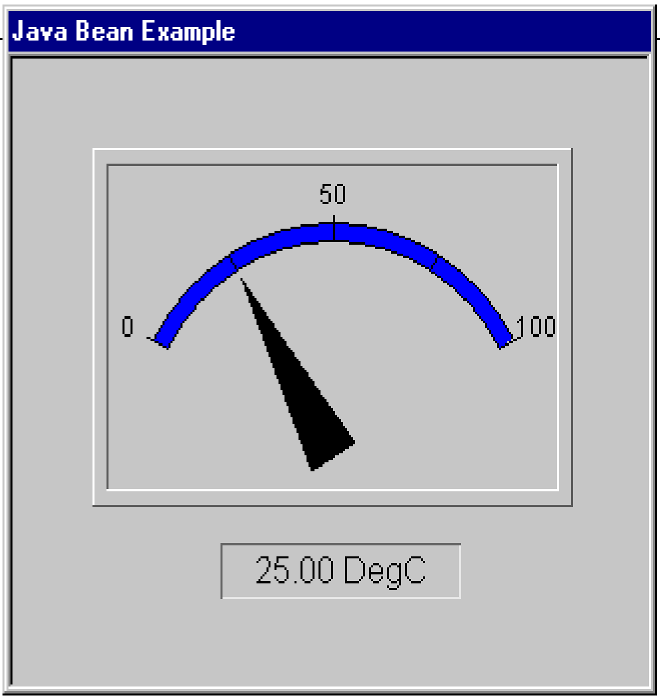
Some properties may be changed by typing in a new value. Some properties may be selected from a list that is displayed when you click in the field. Other properties may be read only – this is indicated by the value appearing in grey. Once you have entered a new value for a property the change becomes active immediately.

The Properties List also displays the Variable Links, Events and Methods available for an object. Selecting a method will display its syntax in the area at the bottom of the Properties List.

How to Link a Bean to a RTDS Variable

The following example uses the Meter bean from Schneider Automation. It assumes that the Bean has been pasted into a mimic and the mimic is in design mode.

1. Select the Meter Bean and display the properties list.
2. Select *Variable Links*. A list of Bean properties that are available to be linked to variables are displayed.
3. Select the *plcValue* property. For the Meter Bean this property is displayed as the angular displacement of the meter pointer.
4. Using the small button that appears adjacent to the property field open the Variable Browser.
5. Select a suitable variable and click the *Apply* button to confirm the selection and close the browser. The variable name should now appear in the field adjacent to plcValue.
6. Save the mimic and switch to run mode. The position of the meter pointer reflects the value of the variable.


Java Bean Example
50
0
100
25.00 DegC

Animation Basics

The Animation Toolbar | How to Apply an Animation | Applying Multiple Animations | Removing an Animation

Animation is the process by which a drawing element is given the capability of interacting with the operator either visually, using the keyboard and mouse, or both. There are three distinct types of animation, although the principle of applying them all is the same.

- Property animations connect a visual property of a drawing element, for example its colour, to the database. At run-time the appearance of the drawing element changes in real time according to the database and the animation you have applied to it. Most of the time you will be using animation to display the real-time value of variable but in addition it may also be used to display the result of an expression, an attribute such as a variable's Domain, or context information such as the current database branch.
- Control Zone animations create an area in a window from which an operator may interact with the application using the keyboard and mouse. Control zones allow the operator to change the value of variables, open and close windows, run programs, send recipes etc.
- Macro animations which create the more complex displays such as Trend Charts.

The Animation Toolbar

The Animation toolbar gives you quick access to the tools used for animation. The toolbar contains 7 tabs, one for each class of animation.

The Colour tab

Colour using one bit.

Colour using the binary combination of up to 4 bits.

Colour using a bit masked from a register.

Colour using a value calculated from weighted bits.

Colour using a register level.

Colour shape fill using the value of a register (Bargraph).

Display a scale using a minimum and maximum value.

The Text tab

Display a string

Display a string in a colour according to the value of a bit.

Display a label.

Display a label in a colour according to the value of a bit.

Display the value of a register numerically.

Display the value of a register using two bits to change the colour.

The Position tab

Display the value of a register using movement in one axis.

Display the value of two registers using movement in two axes.

Display the value of a register using rotation.

Display the value of a register using size.

The Object Tab

Select objects using a single bit.

Select objects using the binary combination of up to 4 bits.

Select objects using a bit masked from a register.

Select objects using the value of a register.

The Link tab

Open a window.

Close a window.

Open a text note.

The Send tab

Change the value of a single bit.

Change the value of a single bit. A colour on bit animation is also applied.

Change the value of two bits. One bit is the inverse of the other.

Change a register value.

Change a text (string) value.

Select a region.

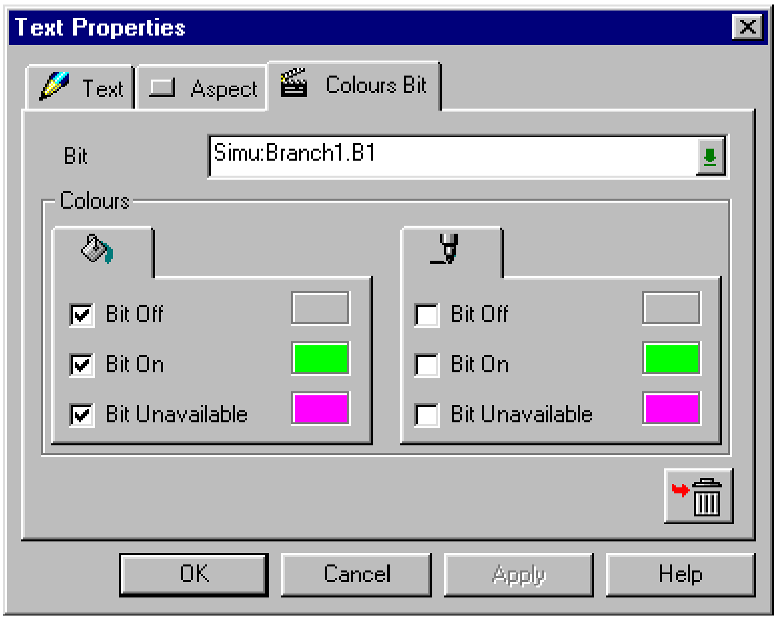
The Run tab

Run an application.

How to Apply an Animation

The basic principle of applying all animations is the same. The following example is for applying a Colour on Bit animation to a text string.

1. Open the window to be animated and select design mode.
2. Select the drawing element you wish to animate. A Colour on Bit animation may be applied to all drawing elements except bitmaps and pre-animated objects.
3. Select the Colour on Bit command from the toolbar or Animation sub-menu.
4. The Drawing properties Box will appear with a new tab called Colours - Bit.



Bit Simu:Branch1.B1

Colours

☑ Bit Off
☑ Bit On 🟩
☑ Bit Unavailable 🟪

☐ Bit Off
☐ Bit On 🟩
☐ Bit Unavailable 🟪

5. Enter the name of the bit either by typing it in directly or by using the Variable Browser. (The Variable Browser may be displayed by clicking the button adjacent to the Bit field). See the following section on using variables in an animation for more information.
6. Tick the boxes corresponding to the state of the bit and the property it is to effect. The tab with the paint can symbol is for the background of the text, the tab with the pen for the text itself.
7. Select the colours corresponding to state of each selected property. Blinking and transparent colours may be used. For further information on selecting colours see the section on drawing.
8. Select OK to confirm the animation.

Applying Multiple Animations

Sometimes it is necessary to apply more than one animation to a drawing element. There are no restrictions placed on this, but before you do this you should take into account the following points.

- Before applying more than one animation double check that one of the standard animations will not meet your requirements. Many of them support combination animation using two or more database variables.
- There is no way to prioritise animation. For example if you apply two colour on bit animations to the same property, the most recent bit change will provide the colour.
- Animations are only refreshed when the value of the attached variable changes.
- When applying multiple animations each animation will have its own tab in the drawing properties box.

Removing an Animation

To remove an animation without deleting the support drawing element:

1. Display the properties box for the drawing element.
2. Select the tab for the animation to be deleted.
3. Click on the dustbin (trashcan) symbol.
4. Click OK to confirm the change.

Using OPC Server Variables in an Animation

Using Relative Referencing | Forcing Direct Referencing | Errors When Selecting Variables

When using an OPC variable in animation you enter its name (as it is known in the server) preceded by the name of the cluster and a colon. If you know the name you can enter it directly but it is normally easier to use the Variable Browser to select the name from the list provided by the server. The following are all examples of valid variable names.

```
Cluster1:Process.Engine1.Compressor1.B1
Cluster1:Process.Engine1.Compressor1.A1
Applicom:CARD1.DATABASE.B1
Data:Node1.DataTable3_256
Simu:Simulation.Square Waves.Int4
C1:B1
```

For further information see the Variable Browser topic.

Using Relative Referencing

If the animation is in a window that has been opened with a branch (for example `Cluster1:Process.Engine1`) you can use relative referencing. With relative referencing, each variable reference uses only part of the variable name. For example you might use the name `Compressor1.B1`. The other part of the variable name is supplied by the branch of the window. The variable names are automatically calculated at run-time by joining the branch with the relative name in the animation.

When using the Variable Browser to select a variable in a window with a branch it automatically selects the correct part of the variable name for you. For example if the window was opened with the branch `Cluster1:Process.Engine2` and you selected the variable `Cluster1:Process.Engine2.Compressor1.B1` it would insert `Compressor1.B1`.

Forcing Direct Referencing

Sometimes in a window that has been opened with a branch you need to refer to a variable with a different branch or no branch at all. To achieve this you use direct referencing where the variable name is preceded by the @ character. For example `Applicom:@CARD1.DATABASE.B1`. When using direct referencing the same variable is always referred to no matter what branch the window is opened with. If you use the Variable Browser to select the variable the @ character is inserted automatically when you select a variable with a different branch to that of the window.

Errors When Selecting Variables

The following error symbols may appear adjacent to the support drawing element when selecting a variable for an animation.

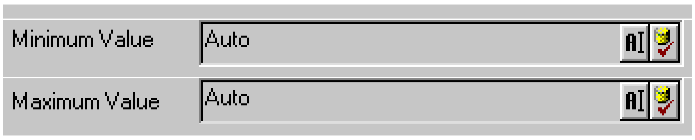
⚠ The HMI has been unable to start the referenced OPC server.

⚠ The variable does not exist in the referenced OPC server

⌛ The HMI is waiting for the real time value of the variable. This symbol sometimes appears briefly when a window is first opened if the server has not been previously started.

Configuring Register Limits

Many of the animations that display the value of a register require a minimum and maximum value for the register to be entered. These are used by the HMI when filling a shape, positioning a drawing on the screen etc.



Minimum Value: Auto

Maximum Value: Auto

By default the values are set to Auto. This means that the HMI will attempt to get them from the variable definition in the server. If the range of the variable in the server changes all animations linked to it automatically follow without the need to change the animation.

If the server doesn't support the maximum and minimum properties or you want to use a different value there are two other options accessed by clicking the buttons adjacent to the bound fields.

▼ Clicking this button allows you to enter a fixed value for the minimum and maximum bounds.

✓ Clicking this button allows the bounds to be defined by the real-time value of another register in the server. The variable browser is displayed to allow selection of the variable name.

Using Custom Output Formats

How to Enter a Format | General Formatting | Numeric Formatting | Time and Date Formatting | Using Conditional Formats

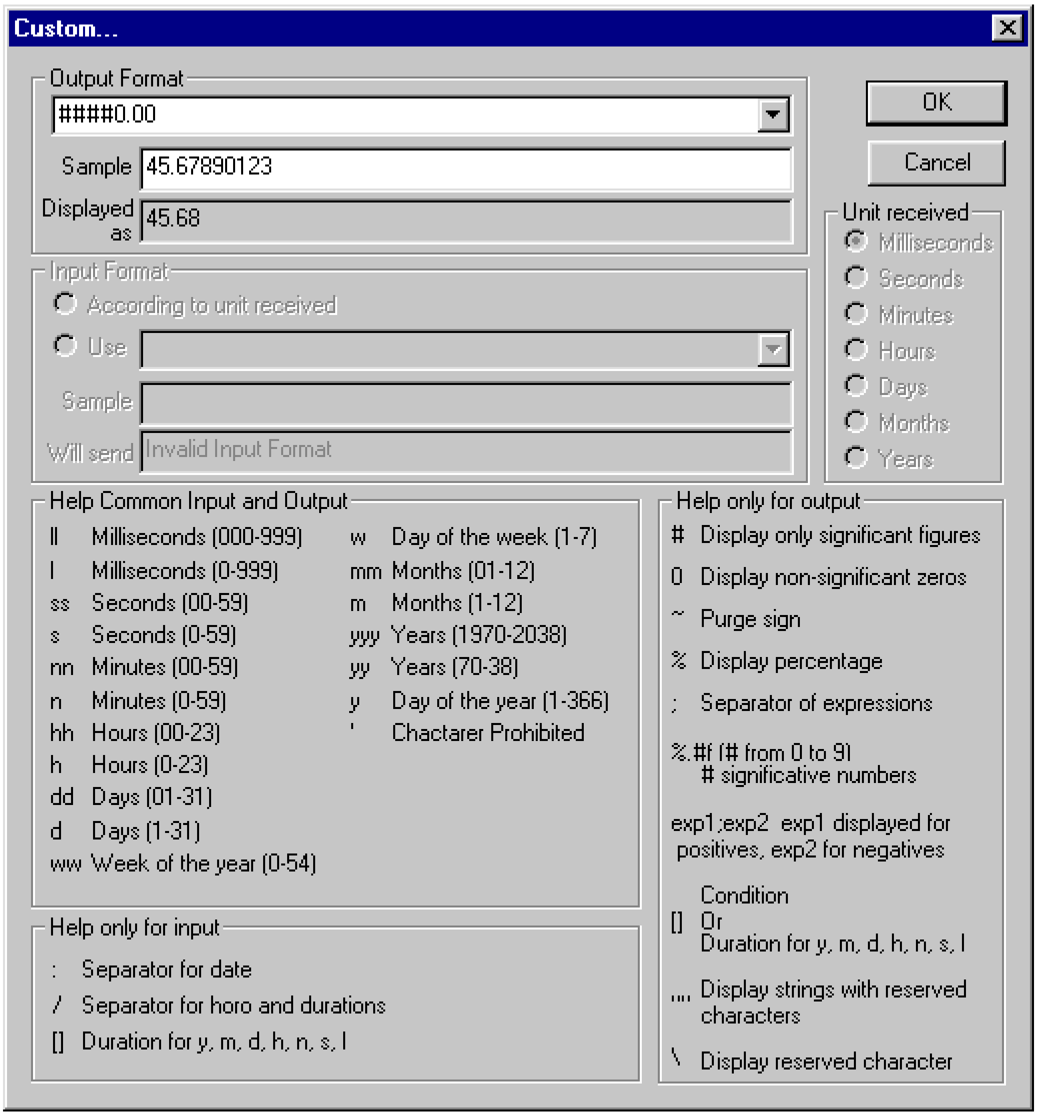
Custom output formats are used animations that display the value of a register.

- Register Display and Register Display with Colour on Bits.
- Send Register Value

They are also used when formatting the numbers displayed on the axis of the Legend Display.

To enter a custom display format you select the down arrow button adjacent to the format field in the animation tab and select the custom option. The custom format box is then displayed into which you can enter the format characters.

It is also possible to use custom formats when inputting a value.



Custom...

Output Format: #####0.00

Sample: 45.67890123

Displayed as: 45.68

Input Format

○ According to unit received

○ Use

Sample

Will send: Invalid Input Format

Unit received

● Milliseconds
○ Seconds
○ Minutes
○ Hours
○ Days
○ Months
○ Years

Help Common Input and Output

ll	Milliseconds (000-999)	w	Day of the week (1-7)
l	Milliseconds (0-999)	mm	Months (01-12)
ss	Seconds (00-59)	m	Months (1-12)
s	Seconds (0-59)	yyy	Years (1970-2038)
nn	Minutes (00-59)	yy	Years (70-38)
n	Minutes (0-59)	y	Day of the year (1-366)
hh	Hours (00-23)	'	Chactarer Prohibited
h	Hours (0-23)		
dd	Days (01-31)		
d	Days (1-31)		
ww	Week of the year (0-54)		

Help only for input

:	Separator for date
/	Separator for horo and durations
[]	Duration for y, m, d, h, n, s, l

Help only for output

#	Display only significant figures
0	Display non-significant zeros
~	Purge sign
%	Display percentage
;	Separator of expressions

%.#f (# from 0 to 9)
# significative numbers

exp1;exp2 exp1 displayed for positives, exp2 for negatives

[] Condition
Or
Duration for y, m, d, h, n, s, l

'''' Display strings with reserved characters

\ Display reserved character

How to Enter a Custom Output Format

1. Select the down arrow button adjacent to the format field
2. Select the custom option. The Custom Format box is displayed.
3. Enter the format in the Output Format field. You can either type in the format characters or select one of the example using the button adjacent to the field.
4. If you are displaying a time and date select the received input units.
5. Click the OK button to confirm the format.

General Formatting Characters

"" Display the string inside the double quotation marks.

\ Display the following character literally. Do not interpret it as a formatting character.

; Separator character. The characters on the left of the semicolon are used to format the value if it is positive and those to the right are used if it is negative.

Numeric Formatting Characters

0 Digit placeholder. Display a digit or a zero. If the expression has a digit in the position where the 0 appears in the format string, display it; otherwise, display a zero in that position. If the number has fewer digits than there are zeros (on either side of the decimal) in the format expression, display leading or trailing zeros.

# Digit placeholder. Display a digit or nothing. If the expression has a digit in the position where the # appears in the format string, display it; otherwise, display nothing in that position.

. Decimal placeholder. The decimal placeholder determines how many digits are displayed to the left and right of the decimal separator.

% Percentage placeholder. The expression is multiplied by 100. The percent character % is inserted in the position where it appears in the format string.

~ Displays the absolute value (unsigned).

, Thousand separator. The thousand separator separates thousands from hundreds within a number that has four or more places to the left of the decimal separator.

Time and Date Formatting Characters

When displaying a register in time or date format you specify the display format and also the raw input format. For example you could display a value as years, months and days and the raw input in hours. The character specifying the input format is placed to the right of the other formatting characters separated by an @. If a raw input is not specified than it is assumed to be in milliseconds. The time display is calculated using the number of raw input units (milliseconds, seconds etc.) since January 1st 1970.

You can also display time information as a duration by enclosing the format characters with square brackets. For example, the raw input could be in minutes and you can display the number of days it represents.

ll Milliseconds with non-significant zeros (000 to 999).

l Milliseconds without non-significant zeros (0 to 999).

ss Seconds with non-significant zeros (00 to 59).

s Seconds without non-significant zeros (0 to 59).

nn Minutes with non-significant zeros (00 to 59).

n Minutes without non-significant zeros (0 to 59).

hh Hours with non-significant zeros (00 to 23).

h Hours without non-significant zeros (0 to 23).

dd Days with non-significant zeros (01 to 31).

d Days without non-significant zeros (1 to 31).

ww Week of the year (1 to 54).

w Day of the week (1 representing Sunday to 7 representing Saturday).

mm Month with non-significant zeros (01 to 12).

m Month without non-significant zeros (1 to 12).

yyy Year as four digits (1970 to 2038).

yy Year as two digits (70 to 38).

y The day of the year (1 to 366).

: Separator for use between time elements.

/ Separator for use between date elements.

@l Use milliseconds as the raw input.

@s Use seconds as the raw input.

@n Use minutes as the raw input.

@h Use hours as the raw input.

@d Use days as the raw input.

@m Use months as the raw input.

@y Use years as the raw input.

Examples of time and date formatting

With a value of 934999200 seconds the formatted output will be:

19:00:00 18/08/1999 for a format of h:nn:ss dd/mm/yyy@s

Day 230 of Year 99 for a format of "Day" y of \year yy@s

Using Conditional Formats

You can configure the format so that it is conditional on the value of the raw variable. In this case the syntax of the format is as follows.

[Expression1]Format1;[Expression2]Format2;[Expression3]Foramt3]....;[ExressionN]FormatN

You can also enter two format strings separated by a semicolon but without conditional expressions. In this case the format on the left will be used for positive numbers and the one on the right for negative numbers.

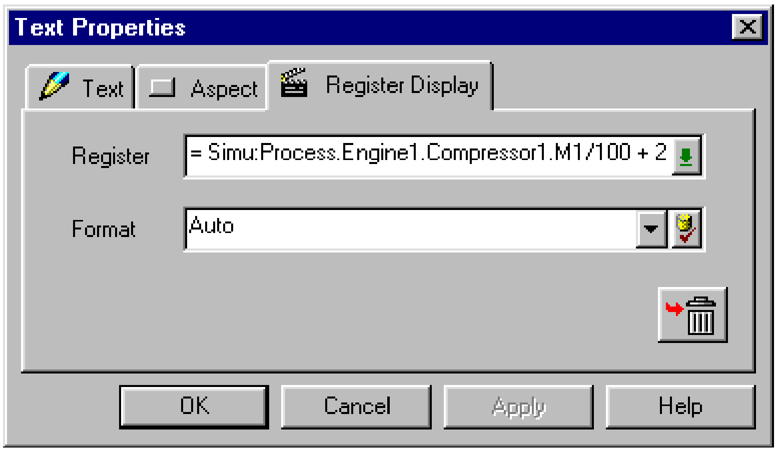
Using Expressions in an Animation

Instead of using a variable name in animation it is possible to use an expression containing operators, functions and variables.

The expression may be entered directly into the field normally used for the variable name, or it may be entered using the Expression Manager in which case a mnemonic referring to the expression model is entered in the variable field. An expression is differentiated from a variable name by immediately (no space) preceding it with an equal sign.

Expressions cannot be used in a animation where an operator enters a value or in a Trend Display.

For further information on the syntax of expressions see the Expression Manager topic.



Text Properties

Register: = Simu:Process.Engine1.Compressor1.M1/100 + 2

Format: Auto

Using Substitution Strings in an Animation

Substitution strings are used to provide context sensitive information to an animation. Unlike using a variable name, the information passed to the animation does not change according to a variable value, instead it reflects a parameter of the current environment. The following table lists the available substitutions and the animations in which they may be used.

String	Substitutes	Use in animation
*	The name of the current window or branch depending on the context.	Link - open and Link - close. Send - program.
#W	The name of the current window.	Link - open and Link - close.
#WB	The current window branch.	Link - send program.
#B1 to #B6.	Elements of the current window branch.	Format string in Text on bit.
#S1 to #S6	Elements of the object branch.	
#P	Name of previous window	Link - open and Link - close.
#M1 to #M10	Window names in window chaining choices.	

You can configure the format so that it is conditional on the value of the raw variable. In this case the syntax of the format is as follows.

[Expression1]Format1;[Expression2]Format2;[Expression3]Foramt3]....;[ExressionN]FormatN

You can also enter two format strings separated by a semicolon but without conditional expressions. In this case the format on the left will be used for positive numbers and the one on the right for negative numbers.

Changing Colour

What are the Visual Effects? | The Colour Properties Tab | How Many Colours? | List Of Colour Animations | Understanding the Thresholds Used in Colour Register Value

Changing the colour of a drawing element is one of the simplest and most commonly used ways of presenting process information to the User. You can use the status of bits and the value of register variables as the source of a colour change.

All colour animations allow you to independently change both the border and pattern (background and characters for text) of a drawing element giving a very wide range of colour combinations. You may use the blinking and transparent options in your choice of colours. All the animations include the capability to select a special colour to be used whenever any of the variables are invalid. (For example when communication has failed).

What are the Visual Effects?

Each type of colour animation allows you to independently change the colour of both the pattern and border of the drawing to which it is applied.

For the Pattern Property

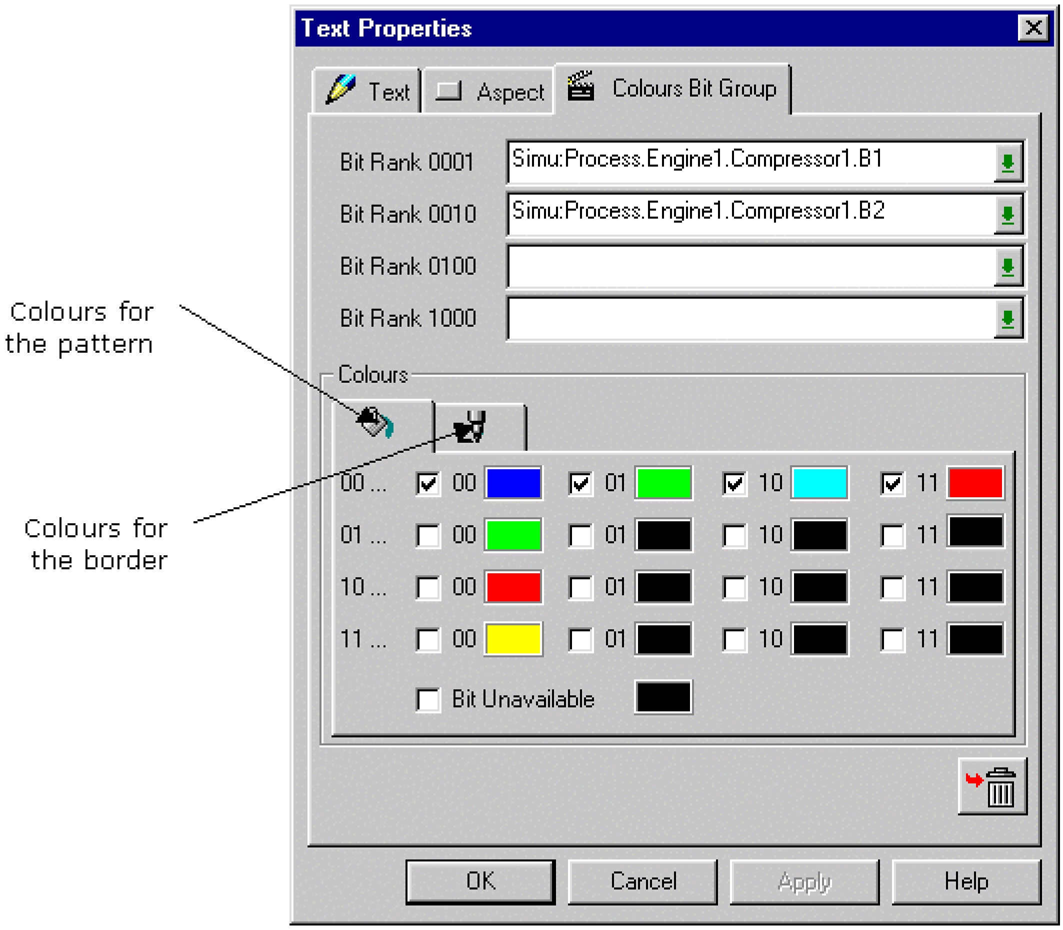
- For a drawing with a solid or coloured button style the whole of the pattern area changes.
- For a drawing with a hatched pattern, only the colour of the hatching changes.
- For text the colour of the background changes.
- There is no effect on lines or polylines.

For the Border property

- For a geometry (rectangle, ellipse etc.), the border changes colour.
- For a line or polyline the whole line changes colour.
- For text, the characters change colour.

The Colour Properties Tab

Each colour change animation is represented by a tab in the Properties box of the drawing to which it is applied. Although the appearance of the tab will vary depending on the type of colour animation chosen, all will contain two common symbols representing the pattern and border properties.



Colours for the pattern

Colours for the border

Text Properties

Bit Rank 0001 Simu:Process.Engine1.Compressor1.B1

Bit Rank 0010 Simu:Process.Engine1.Compressor1.B2

Bit Rank 0100

Bit Rank 1000

Colours

00 ...	☑ 00 🟦	☑ 01 🟩	☑ 10 ■	☑ 11 🟥
01 ...	☐ 00 🟩	☐ 01 ⬛	☐ 10 ⬛	☐ 11 ⬛
10 ...	☐ 00 🟥	☐ 01 ⬛	☐ 10 ⬛	☐ 11 ⬛
11 ...	☐ 00 🟨	☐ 01 ⬛	☐ 10 ⬛	☐ 11 ⬛

☐ Bit Unavailable ⬛

Using the Colour Palette to Select the Colours

The dynamic colours used in an animation are selected using the colour palette in the same way as when you colour a static drawing element. You may use single, blinking and transparent colours.

Specifying a Colour for Invalid Status

On all the colour animations you have the option to select a special colour that will be displayed when any of the variables used becomes invalid. A variable may become invalid for a variety of reasons, the most common one being the loss of communication to external equipment such as a PLC or RTU. **It is recommended that you make use of this option**. If you do not, when a variable becomes invalid the animation will display the colour representing the last valid value of the variable. As a consequence, the User may be unaware that he may be viewing out of date information.

How Many Colours Can I Select?

The number of colours that you can select depends on the animation chosen. You do not have to use all of the colours. For example, the animation Colour - Bit Group allows you to select up to 16 colours using 4 bits, but you could use it to select just 4 colours using 2 bits. However, whichever animation you choose, you must select at least 2 of the available colours as the transition to or from an unselected state is not recognised.

List Of Colour Animations and Behaviour

Animation	**Behaviour**
Bit	Selection of 2 colours according to the status of a single bit.
Bit group	Selection of 16 colours according to the binary combination of up to 4 bits.
Alarm	Selection of 5 colours according to the 5 states of an alarm. (Off, On - Nack, On - Ack, Off - Nack, Inhibited). If you select the Static option the colours used are those you specify in the animation. If you select the Dynamic option the colours are taken from the preferences for the appropriate alarm level. Changing the preferences will change the colours displayed in the animation.
Bit & Alarm	Selection of 8 colours according to the binary combination of the 2 states of a bit and 4 states of an alarm.
Register bit	Selection of 2 colours using a bit masked from a register with a binary rank. For example if you use the 3rd bit, when the binary representation of the register is xxxxxxxxxxxx1xx the colour selected for 1 will be displayed. When the binary representation of the register is xxxxxxxxxxxx0xx the colour selected for 0 will be displayed.(x = either 1 or 0)
Weighted bits	Similar to above except that the value is calculated by adding the values of 8 weighted bits. Bit 1 = 1, Bit 2 = 2, Bit 3 = 5, Bit 4 = 8 etc. For example 000010001 = 17. This animation is generally used with binary encoded transducers
Register value	Selection of up to 8 colours according to value of a single register. See following page for an explanation of threshold operation.

Understanding the Thresholds Used in Colour Register Value

In the animation Colour - Register Value you select a register and enter up to 8 threshold values and corresponding colours. The threshold values must be in ascending order. Each threshold determines the point at which a colour change takes place. This is best explained by example.

Threshold 1 = 10	Colour = Blue
Threshold 2 = 15	Colour = Green
Threshold 3 = 20	Colour = Yellow
Threshold 4 = 30	Colour = Red
Reg. out of range	Colour = White

A register value of less than 10 will display blue.
A register value of 10 or greater but less than 15 will display green.
A register value of 15 or greater but less than 20 will display yellow.
A register value of 20 or greater but less than 30 will display red.
A register value of 30 or above will display white.

If the Reg. out of range option is not selected then a register value of 20 or greater will display red.

Displaying Messages and Labels

The Message Animation | The Label Animation

Messages and labels provide real-time textual information to the User.

Both the Message and Label animations use a text field as the support drawing element. The text field provides the font, colour and the style (inverse, button etc.) properties.

The Message Animation

The Message animation displays the contents of text variable. The most common use of this is to display the time, date and user name. Other uses include operator prompts, warning messages and status information.

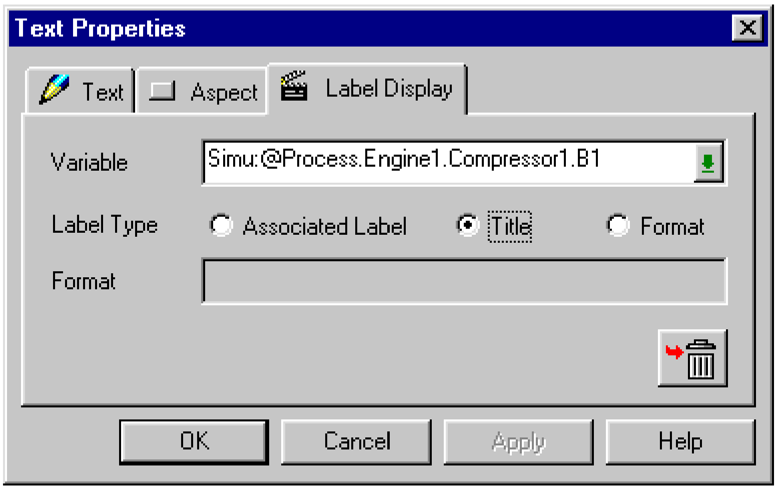
When using the Message animation it is normal to select the Do not auto size property of the support text field. If unselected, the length of the field will dynamically change at run-time according to the number of characters displayed. This can give an odd effect if you are using a style, such as inverse or button, that displays a background colour.

If you need to display very long messages you can select the Multiline property in the support text field. The message may then be displayed on a number of lines, automatically broken at the most convenient white space.

The Label Animation

The Label animation displays text associated with a bit variable. You can display:

- The variable’s associated label.
- A variable’s item description.
- Context sensitive information such as the name of the window in which the animation resides, the branch of an object etc.



Variable: Simu:@Process.Engine1.Compressor1.B1

Label Type: ○ Associated Label ● Title ○ Format

Format:

Displaying Associated Labels

When using a bit an associated label is a pair of strings, one string is displayed when the bit is off, the other when the bit on. If the bit has not been attached to an associated label then the default strings of `Animation 0` and `Animation 1` will be displayed.

Displaying A Variable's Title

If you choose the option to display the variable's title and the variable does not have a title then the variable name will be displayed instead. A variable's title is bilingual and so if you change the language for the project you must ensure that you have a title in the appropriate language.

Displaying Context Sensitive Information.

To display context sensitive information, you select the Format option button and then enter a substitution string in the field provided. The substitution string determines what is displayed in the label. The following options are available.

String	Substitutes
#W	The name of the current window.
#WB	The current window branch.
#B1 to #B6.	Elements of the current window branch.
#S1 to #S6	Elements of the object branch.

You can combine substitution strings and use other characters which are displayed literally.

Compatibility Note

In the animation toolbox and menu you will find, in addition to the standard Message and Label animations, Message – Bit and Label - Bit. These create a Colour on Bit animation at the same time as the Message or Label (See the section on changing colour for details). They are provided for compatibility with earlier versions of the software where it was only possible to attach one animation to each drawing element.

Using these animations is exactly the same as applying a Message (or Label) plus a Colour - Bit to the same drawing element.

Displaying a Register as a Numeric Value

There are two animations which display a register as a numeric value known as Register and Register & Bits. The animation Register & Bits is provided for compatibility with earlier versions of the software. Both animations use a text field as the support. The text field provides the font, colour and the style (inverse, button etc.) properties.

Selecting the display format

The display format is selected using the format field in the Display Register tab.

▼ Selects either the auto or custom formats.

✓ Allows the format to be supplied by a variable in the server.

Compatibility Note - Register & Bits

The animation Value - Register & Bits, which allows you to display a register and change its colour using a combination of up to 4 bits (2 for the text colour & 2 for the background colour), is provided for compatibility with earlier versions of the software. The functionality it provides is similar, but not identical, to using the Text - Register plus Colour – Four Bits animations on the same drawing.

Displaying a Register as a Bargraph

Using a bargraph, the value of a register is represented by filling one of the solid drawing elements. You can apply a bargraph to a rectangle, rounded rectangle, ellipse and polygon. The fill is calculated using the value of the variable with respect to its range and the height (or width) of the drawing element. In the case of a polygon the fill does not take into account the area occupied.

Configuring the Style

The colour of the bar and the background are specified by the configuration of the animation. However the style of the bargraph is inherited from the support drawing element. For example the bargraph can appear in relief, or as a coloured button.

Configuring the display bounds

The minimum and maximum bounds are the values representing 0% and 100% fill of the bargraph. By default the bounds are set to Auto. This means that the HMI will attempt to get the bounds for the variable from the OPC server. If the range of the variable in the OPC server changes all animations linked to it using the Auto bound will automatically follow without the need to change the animation.

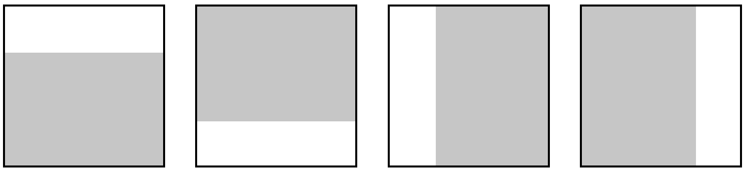
If the server doesn't support variable bounds or you want to use a different value there are two other options accessed by clicking the buttons adjacent to the bound fields.

▼ Clicking this button allows you to enter a fixed value for the minimum and maximum bounds.

✓ Clicking this button allows the bounds to be defined by the value of another register in the server. The variable browser is displayed to allow selection of the variable name.

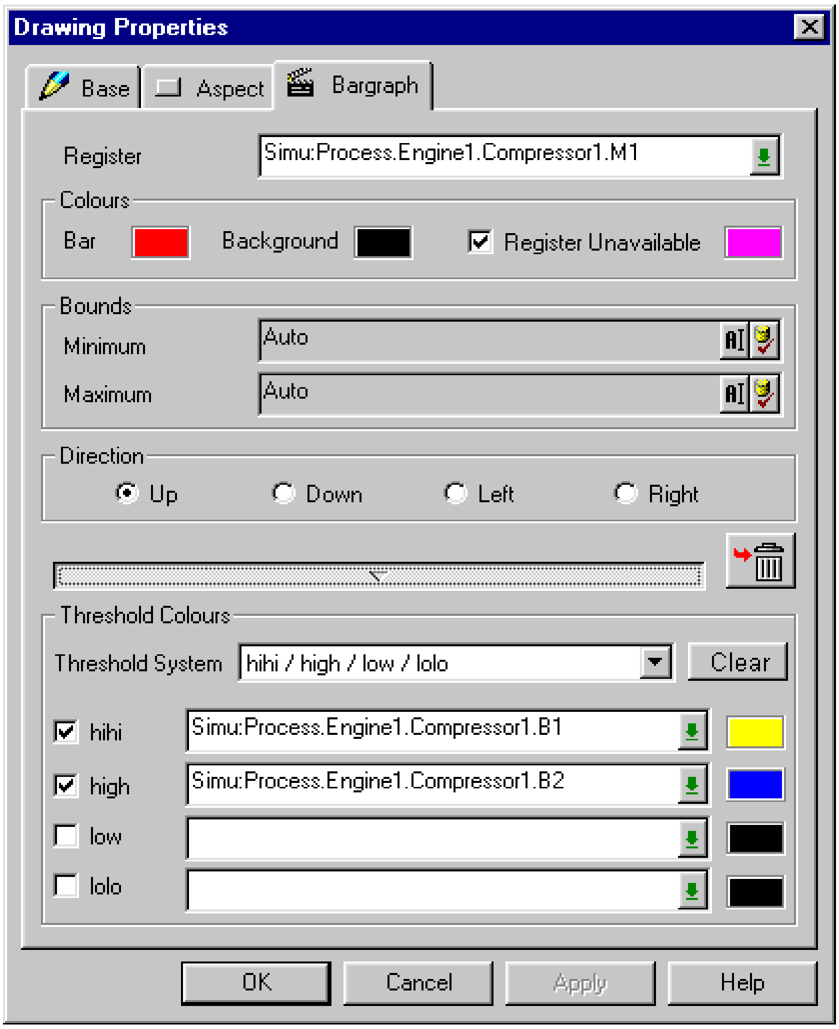
Configuring the fill direction

The animation properties allow you to enter the range of the variable for the fill, as a minimum and maximum, and the fill direction. You can fill from the bottom, top, left or right.



Changing the Bargraph Colour

You can change the colour of the bargraph bar at run-time using a threshold system controlled by up to 4 bits. To display the threshold properties you click the down arrow button at the bottom of the bargraph tab.



Drawing Properties

Base | Aspect | Bargraph

Register: Simu:Process.Engine1.Compressor1.M1

Colours
Bar 🟥 Background ⬛ ☑ Register Unavailable 🟪

Bounds
Minimum: Auto
Maximum: Auto

Direction
● Up ○ Down ○ Left ○ Right

Threshold Colours
Threshold System: hihi / high / low / lolo — Clear

☑ hihi: Simu:Process.Engine1.Compressor1.B1 🟨
☑ high: Simu:Process.Engine1.Compressor1.B2 🟦
☐ low: ⬛
☐ lolo: ⬛

OK | Cancel | Apply | Help

The way in which the bits effect the colour of the bargraph depends on which of the three threshold systems is selected. In the explanation below the threshold name corresponding to the bit is used.

hihi, high, low, lolo

```
If hihi is 1 then
    Use the colour for hihi
    Else if high is 1 then
        Use the colour for high
        Else if lolo is 1 then
            Use the colour for lolo
            Else if low is 1 then
                Use the colour for low
                Else the colour is not changed.
```

ppphigh, pphigh, hihi, high

```
If ppphigh is 1 then
    Use the colour for ppphigh
    Else if pphigh is 1 then
        Use the colour for pphigh
        Else if hihi is 1 then
            Use the colour for hihi
            Else if high is 1 then
                Use the colour for high
                Else the colour is not changed.
```

low, lolo, pplow, ppplow

```
If ppphigh is 1 then
    Use the colour for ppphigh
    Else if pphigh is 1 then
        Use the colour for pphigh
        Else if hihi is 1 then
            Use the colour for hihi
            Else if high is 1 then
                Use the colour for high
                Else the colour is not changed.
```

Exchanging Objects

List Of Object Animations | Exchanging Bitmaps

Simple changes in process status may be indicated by change of colour. For example, a pump may be displayed in green when it is running, red when it is stopped and blue when faulty. In some cases however, changing colour is not adequate.

Consider the case where a float switch is to be displayed. The requirements are that it should appear horizontal when its corresponding bit in the database is true, and at an angle of 45^o when the bit is false. This can easily be achieved by creating two objects, each representing one of the conditions, and exchanging them depending on the state of the database bit. In practice exchanging objects can create many visual effects.

Anchor Points

When exchanging objects the support drawing element is always another object, normally one of those used in the animation. The location of all objects used in the animation is determined by aligning their anchor points with that of the support object.

Using Objects With a Branch

If the objects being displayed contain animations which use relative variable referencing then a branch must be supplied. The branch may be entered directly, by using the branch selector or by using substitution characters. For further information see the chapter on An Introduction to Animation. For further information on objects see the topic on Creating and Using Objects.

Specifying an Object for Invalid Status

On all the object animations you have the option to select a special object that will be displayed when any of the variables used becomes invalid. A variable may become invalid for a variety of reasons, the most common one being the loss of communication to external equipment such as a PLC or RTU.

It is highly recommended that you make use of this option. If you do not, when a variable becomes invalid the animation will display the object representing the last valid value of the variable. As a consequence, the User will be unaware that he may be viewing out of date information.

List Of Object Animations and Behaviour

Animation	Behaviour
Bit	Selection of 2 objects according to the status of a bit. A third object may be selected to represent the invalid state.
Bit group	Selection of up to 16 object according to the binary combination of up to 4 bits. An additional object may be specified that will be displayed if any of the bits are invalid.
Alarm	Selection of 4 objects according to the 4 states of an alarm. (Off, On – Nack, On - Ack, Off - Nack). A further 4 objects may be selected to represent the various invalid conditions that an alarm may take.
Register bit	Selection of 2 objects by a bit masked from a register using a binary rank. For example if you use the 3rd bit, when the binary representation of the register is xxxxxxxxxxxxx1xx the object selected for 1 will be displayed. When the binary representation of the register is xxxxxxxxxxxxx0xx the object selected for 0 will be displayed.(x = either 1 or 0). A third object may be selected to represent the invalid state.
Register value	Selection of 8 objects according to up to 8 thresholds applied to a register. An additional object may be selected to represent the invalid state. See the section on changing colour for an explanation of threshold operation.

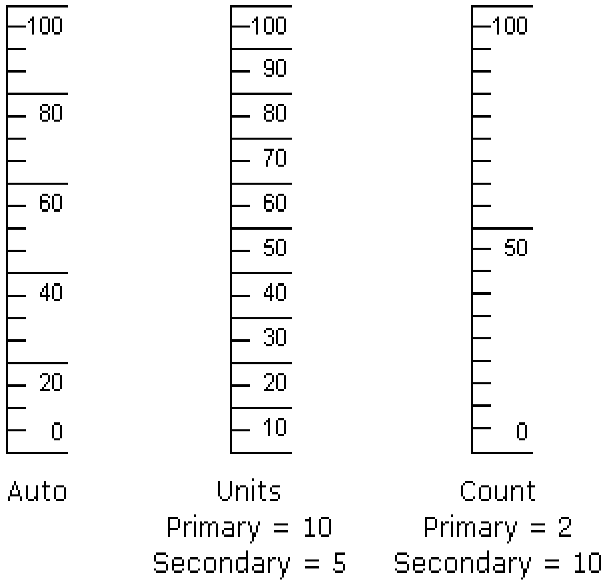
Exchanging Bitmaps

Sometimes it is necessary to exchange bitmaps. To do this objects are created, each one containing one of the bitmaps. The bitmaps may now be exchanged by exchanging the objects.

The Legend Animation

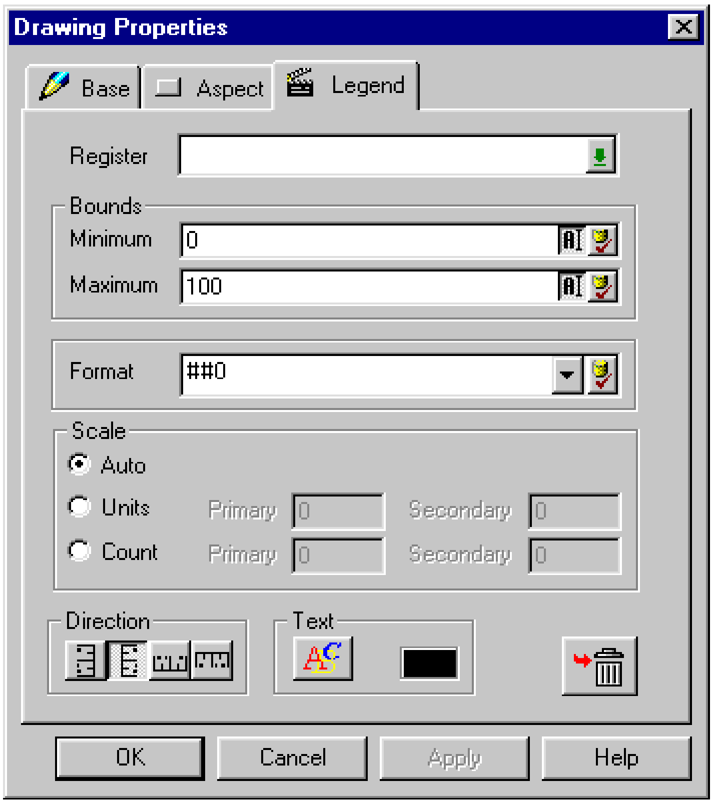
Configuring the Scale Appearance | Selecting the Scale Range

The Legend animation automatically creates a numeric scale that may be used in conjunction with bargraph or position animation to provide the User with an indication of the value. The Legend animation uses a rectangle as the support drawing element. The rectangle provides size o l the colour and width of the lines.



The advantage of using the Legend over a fixed scale drawn manually is that the Legend can be configured to get the maximum and minimum value for the scale from a variable. If the range of the variable changes the Legend follows automatically.

Configuring the Scale Appearance



Register

Bounds

Minimum: 0

Maximum: 100

Format: ##0

Scale

● Auto

○ Units Primary 0 Secondary 0

○ Count Primary 0 Secondary 0

Direction

Text

The number of primary and secondary divisions that appear on the scale may be selected using one of three methods.

- *Auto* – The HMI decides on the number of divisions using a built in algorithm.
- *Units* – The number of primary divisions is calculated using the range of the scale divided by the number entered in the primary field. For example, using a minimum of 0, a maximum of 100 and a primary unit of 25 there would be 4 primary divisions. The number of secondary divisions per primary division is calculated using the number entered in the primary field divided by the number in the secondary field. For example, using a primary unit of 25 and a secondary unit of 5 there will be 5 secondary divisions.
- *Count* – The number entered in the primary field is the number of primary divisions. The number entered in the secondary field is the number of secondary divisions per primary division.

The direction of the scale is selected using the direction command buttons. If one of the horizontal options is selected the text is rotated so that it reads bottom to top.

The format of the scale number is selected using the standard output formatting characters.

Selecting the Scale Range

By default the maximum and minimum values that define the scale range are set to Auto. This means that the HMI will attempt to get them from the register definition in the server.

If the server doesn’t support the maximum and minimum properties or you want to use a different value there are two other options accessed by clicking the buttons adjacent to the maximum and minimum fields.

▼ Clicking this button allows you to enter a fixed value for the minimum and maximum bounds. If you select a fixed value then it is not necessary to select a register name.

✓ Clicking this button allows the minimum and/or maximum to be defined by the real-time value of other register(s) in the server. The variable browser is displayed to allow selection of the variable name.

What Are the Position Animations?

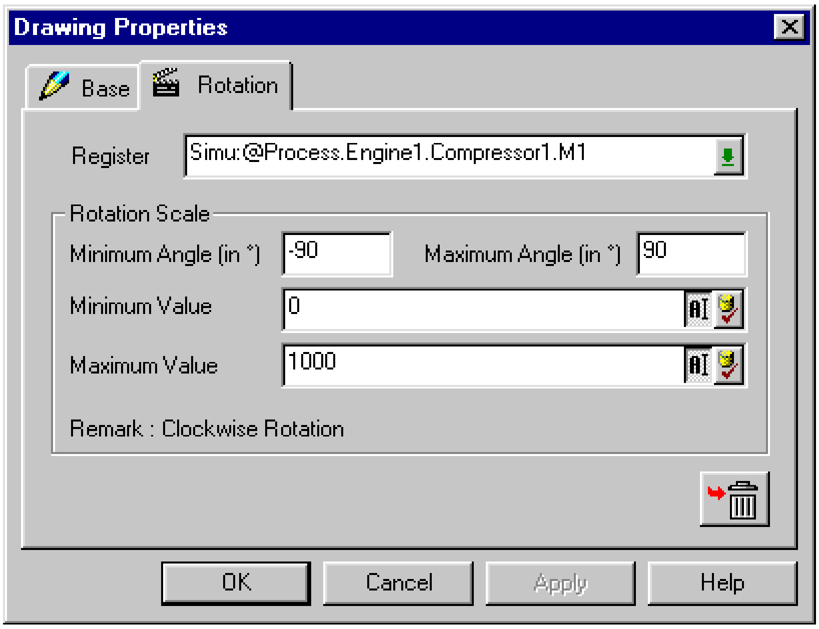
The position animations allow you to move, rotate and change the size of drawing elements at run-time according to the value of one or more register variables.

In addition the positioning animations allow the User to set the value of one or two registers by using the cursor to drag a drawing element.

Rotation

Specifying the Rotation Scale

When using the rotation animation the value of a register is represented by angular movement. A typical example of its use is to move the needle in a simulation of an analogue meter. You can apply rotation to all drawing elements except text and images.



Register: Simu:@Process.Engine1.Compressor1.M1

Rotation Scale

Minimum Angle (in °): -90

Maximum Angle (in °): 90

Minimum Value: 0

Maximum Value: 1000

Remark : Clockwise Rotation

Specifying the Rotation Scale

The angle through which the drawing rotates is defined by specifying a minimum and maximum value for the register together with their corresponding angles. The rotation angle starts at zero degrees for no rotation and is clockwise for increasing values of angle. You may specify positive and negative values for both the value and angle. Rotation takes place about the drawing's anchor point. The following are two examples of different rotation scales and the effect on the drawing element.

Minimum value = 0 Minimum angle = 0^0
Maximum value = 100 Maximum angle = 90^0

With a value of 0 the drawing will remain in its original position.
With a value of 50 the drawing will rotate clockwise 45^0.
With a value of 100 the drawing will rotate clockwise 90^0.

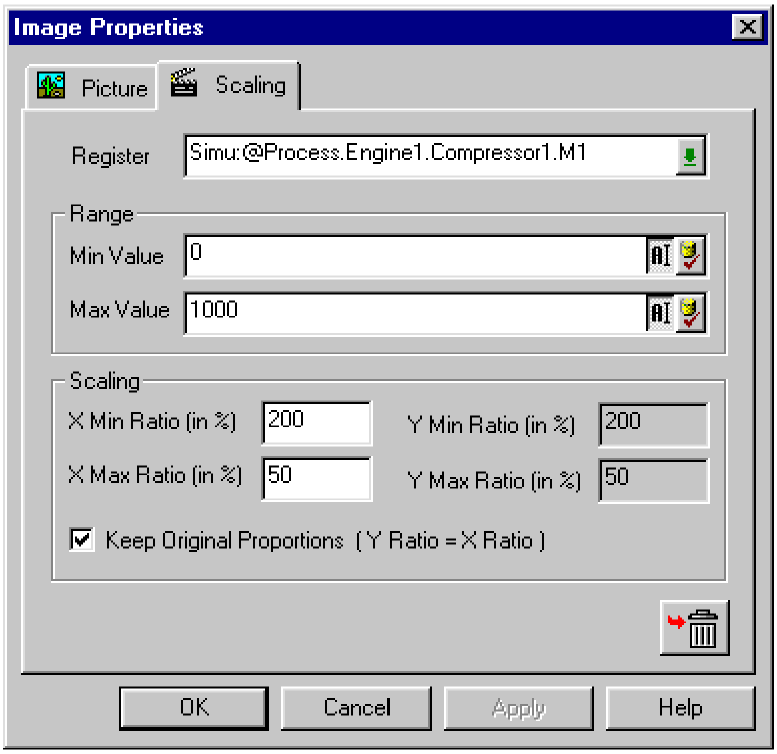
Minimum value = 0 Minimum angle = 0^0
Maximum value = 100 Maximum angle = -90^0

With a value of 0 the drawing will remain in its original position.
With a value of 50 the drawing will rotate anti-clockwise 45^0.
With a value of 100 the drawing will rotate anti-clockwise 90^0.

Register values less than the minimum are treated the same as a minimum value. In the first example above a value of -10 would result in a rotation of 0^0. Register values greater than the maximum are treated the same as a maximum value. In the first example above a value of 200 would result in a rotation of 90^0. If the register is invalid then the rotation angle is the same as for the minimum value.

Scaling

When using the scaling animation the value of a register is represented by changing the size of a drawing. You can scale a drawing in both the X and Y axes. Scaling may be applied to all drawing elements, although there are some limitations when using it with text.



Register: Simu:@Process.Engine1.Compressor1.M1

Min Value: 0

Max Value: 1000

X Min Ratio (in %): 200

Y Min Ratio (in %): 200

X Max Ratio (in %): 50

Y Max Ratio (in %): 50

☑ Keep Original Proportions (Y Ratio = X Ratio)

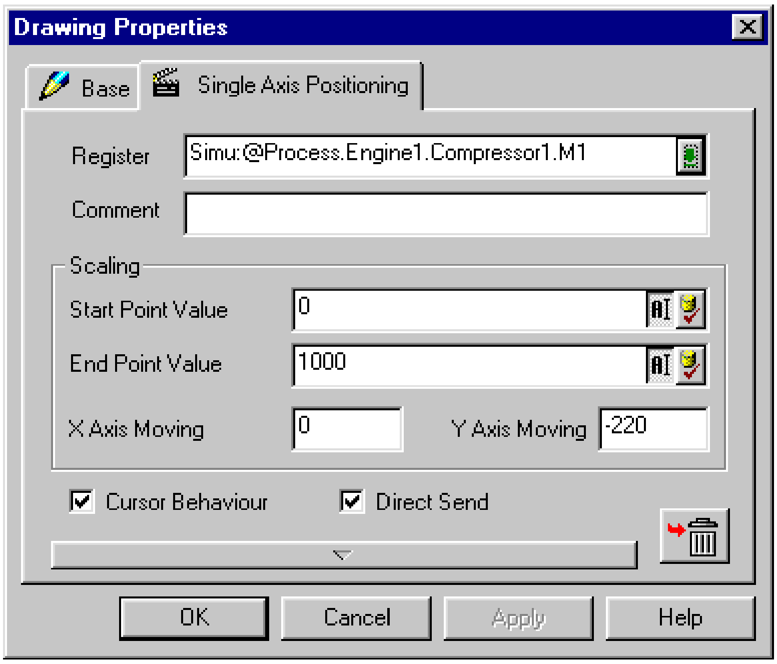
The scaling for the drawing is defined by specifying minimum and maximum values for the register together with the corresponding scale ratios. You can specify different scaling for the X and Y dimensions if required. At run-time the drawing is scaled about the anchor point.

When scaling text the *Do Not Auto Size* property must be selected.

Single Axis Positioning

Using the Cursor Behaviour Property

Single axis positioning is used to move a drawing according to the value of a register. Single axis positioning may be applied to all drawing elements.



Register: Simu:@Process.Engine1.Compressor1.M1

Comment:

Scaling

Start Point Value: 0

End Point Value: 1000

X Axis Moving: 0

Y Axis Moving: -220

☑ Cursor Behaviour ☑ Direct Send

The drawing movement is along an (imaginary) straight line between two points on the screen. The co-ordinates for the start of the line are supplied by the position of the support drawing element. The co-ordinates for the end of the line are the position of the support drawing element plus the values of the X and Y axis movement properties.

The X and Y axis movement properties may be positive or negative. Positive X axis values cause movement from the left to the right of the screen. Positive Y axis values cause movement from the top to the bottom of the screen.

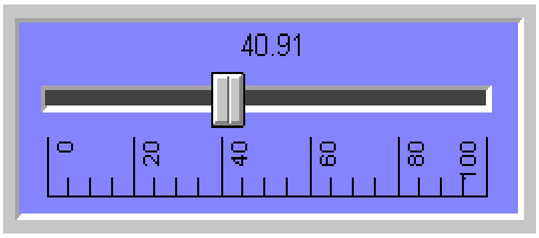
The start and end point values can be any value within the range of the selected register. The start value can be less than or greater than the end value.

It is possible to create movement in both axes.

Using the Cursor Behaviour Property

If the *Cursor Behaviour* property is selected the User is able to click and drag the drawing element to set the value of the register. This is used to create what is commonly known as a slider control.

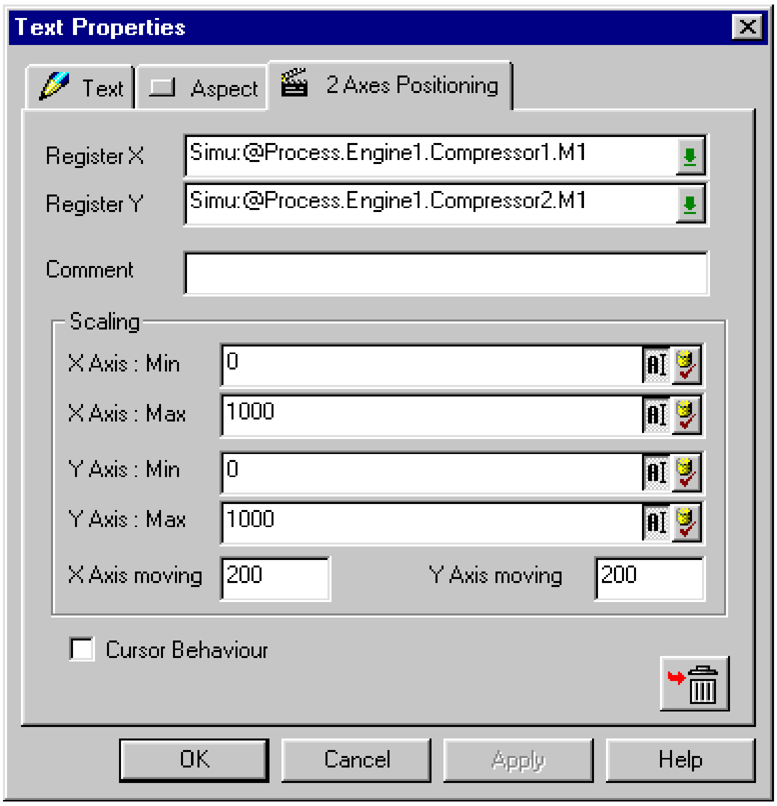
If the *Direct Send* property is selected the value of the register is changed as the drawing is dragged. If the Direct Send property is not selected the value is only set when the mouse button is released.



2 Axes Positioning

Using the Cursor Behaviour Property

2 axes positioning is used to move a drawing according to the value of 2 registers. 2 Axes positioning may be applied to all drawing elements.



Register X: Simu:@Process.Engine1.Compressor1.M1
Register Y: Simu:@Process.Engine1.Compressor2.M1
Comment:

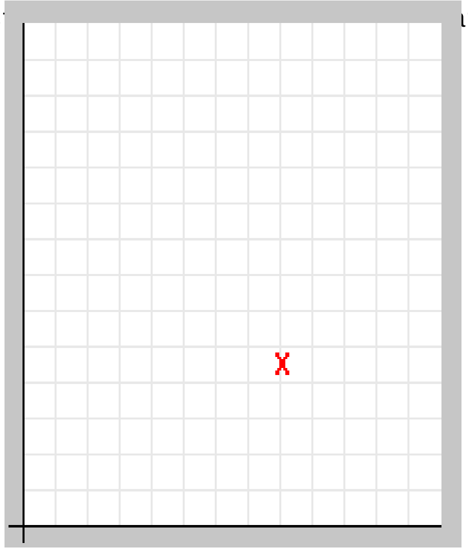
Scaling
X Axis : Min: 0
X Axis : Max: 1000
Y Axis : Min: 0
Y Axis : Max: 1000
X Axis moving: 200
Y Axis moving: 200

☐ Cursor Behaviour

The drawing movement is within an imaginary rectangle. One corner of the rectangle is defined by the position of the support drawing element. The opposite corner is defined by the position of the support drawing element plus the X and Y axis movement properties.

The X and Y axis movement properties may be positive or negative. Positive X axis values cause movement from the left to the right of the screen. Positive Y axis values cause movement from the top to the bottom of the screen.

The start and end point values can be any value within the range of the selected registers. The s ter than the end value.



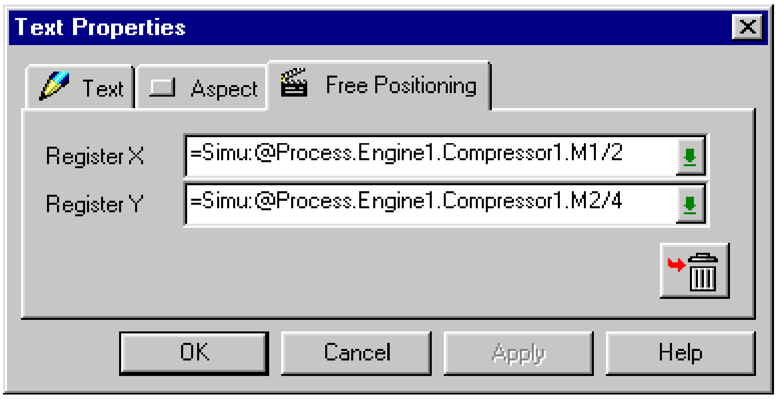
Using the Cursor Behaviour Property

If the *Cursor Behaviour* property is selected the User is able to click and drag the drawing element to set the value of the registers.

If the *Direct Send* property is selected the value of the registers is changed as the drawing is dragged. If the Direct Send property is not selected the value is only set when the mouse button is released.

Free Positioning

Free positioning is used to move a drawing according to the value of 2 registers. Free positioning may be applied to all drawing elements.



Register X: =Simu:@Process.Engine1.Compressor1.M1/2

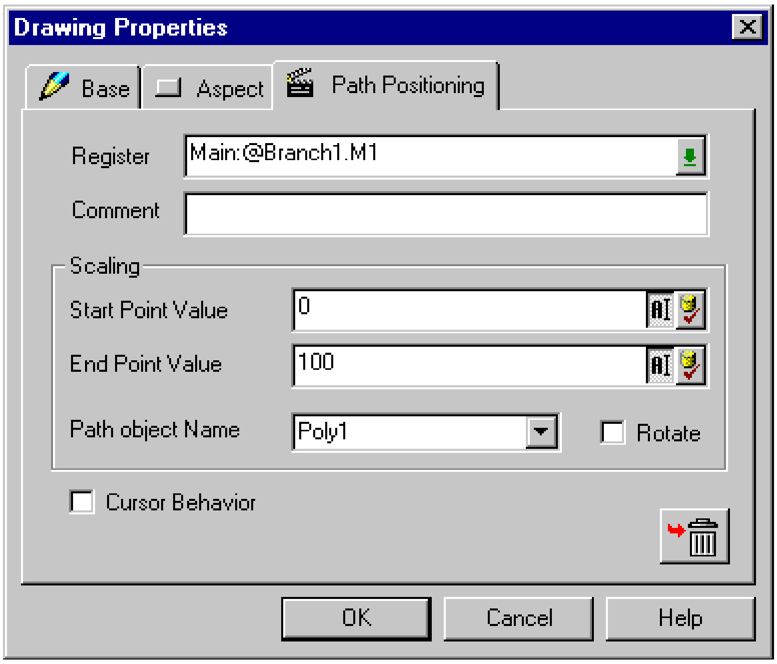
Register Y: =Simu:@Process.Engine1.Compressor1.M2/4

The position of the anchor point of the drawing element on the screen in pixels is taken from the value of the X and Y registers. (The position of the anchor point relative to the drawing element remains unchanged.) The initial position of the drawing element is not taken into account.

Path

How the Animated Drawing Moves | Using the Cursor Behaviour Property | Drawing Restrictions

The path animation allows you to move a drawing element along a path defined by a polyline, polygon, Bézier curve or Bézier shape. The path animation may be applied to any drawing element or symbol.



Drawing Properties

Register: Main:@Branch1.M1

Comment:

Scaling

Start Point Value: 0

End Point Value: 100

Path object Name: Poly1

□ Rotate

□ Cursor Behavior

The animation is applied to the drawing that is to move, not to the drawing element providing the path.

The animated drawing and the drawing providing the path must both belong to the same depth from a grouped point of view. That is, if the drawings are grouped, they must both be in the same group.

How the Animated Drawing Moves

The range of register values over which the drawing moves is divided equally by the number of segments in the drawing providing the support path. For example, if the range is 0 to 100 and there are 5 segments in the path drawing, the animated drawing will move along the first segment from 0 to 19, the second segment from 20 to 39 etc. The length of the segments is not taken into account and so if the segments are of different lengths the drawing will appear to move faster and slower as it moves along the different segments.

At run-time, when the register is at the start point value, the anchor point of the animated drawing element is located at the start position (X and Y in the drawing properties box) of the path drawing. The location of the animated drawing element in design mode has no relevance.

If the *Curved* property is enabled the animated drawing will rotate according to the angle of the segment it is following.

Using the Cursor Behaviour Property

If the *Cursor Behaviour* property is selected the User is able to click and drag the drawing element to set the value of the register.

If the *Direct Send* property is selected the value of the register is changed as the drawing is dragged. If the Direct Send property is not selected the value is only set when the mouse button is released.

Drawing Restrictions

The animated drawing and the drawing providing the path must both belong to the same depth from a grouped point of view.

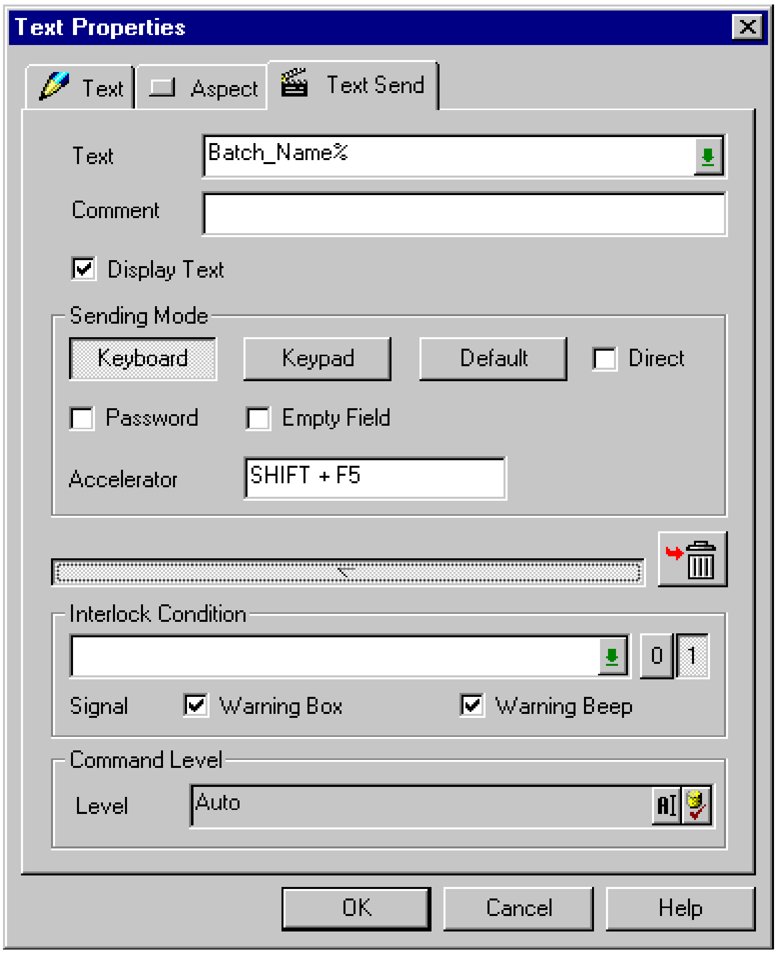
What is a Control Zone?

Using an Accelerator Key with a Control Zone | Using the Comment Field to Create a ToolTip | Attaching a Control Zone to a Command Level | Using an Interlock Bit with a Control Zone

A control zone is an area on the screen from where a User may interact with the HMI. Control zones allow the following actions.

- To send a value to a bit, register or text variable.
- To start another application (Excel, Word etc.)
- To open and close windows.
- To create a hyperlink.
- To display a text file.

A control zone is created in the same way as any other animation by selecting a drawing element and applying an animation to it. In the drawing properties box an additional tab will appear in which the properties for the animation can be selected. At run-time, to draw attention to the User, the cursor changes shape whilst it is over a control zone and the drawing is highlighted with a button style border.



Text: Batch_Name%

Comment:

☑ Display Text

Sending Mode

☐ Direct

☐ Password ☐ Empty Field

Accelerator: SHIFT + F5

Interlock Condition

0 1

Signal ☑ Warning Box ☑ Warning Beep

Command Level

Level: Auto

Whilst the actions produced by the various control zone animations are different, they have a number of common properties.

- *Accelerator key* - You can assign a keystroke that has the same effect as clicking on the control zone.
- *Tooltip* - You can create a prompt that is automatically displayed when the pointer is over the control zone.

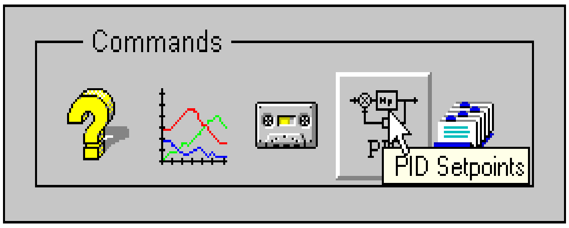
- *Command level* - You can lock a control zone to a command level restricting its use to certain Users.
- *Interlock condition* - You can specify a bit that may be used as an interlock.

Using an Accelerator Key with a Control Zone

An Accelerator key is a key or combination of keys that produces the same operation as clicking on a control zone. To allocate an accelerator to a control zone, click in the accelerator field and then press the combination of keys that are to act as the accelerator. The key strokes you have pressed will appear as text in the accelerator field. For example, Shift F2.

Using the Comment Field to Create a ToolTip

All of the animations used to create control zones have a text field called *Comment* in their properties tab. A string, preceded by the @ character, entered into the comment field will be displayed as a ToolTip when the cursor is positioned over the control zone.



Attaching a Control Zone to a Command Level

A control zone may also be attached to a command level. At run-time the User must have rights to that command level to operate the control zone.

For animations that do not change the value of a variable a fixed command level must be entered.

For animations that change the value of a variable the command right field is set to auto by default. This means that the HMI will attempt to get the value from the variable definition in the server. If the server support this property or you want to use a different value there are two other options accessed by clicking the buttons adjacent to the fields.

▼ Clicking this button allows you to enter a fixed value for command level.

Clicking this button command level to be defined by the value of another register in the server. The variable browser is displayed to allow selection of the variable name.

This property is available in the advanced section of the properties tab.

Using an Interlock Bit with a Control Zone

All control zones have the option of an interlock using a bit variable. The control zone is active only when the bit is in the selected state. A bit variable is used because of the flexibility it gives you. The value of the bit could come from a piece of equipment such as a PLC, from another computer, or derived internally using an expression or program. When using an interlock you have the option of a warning beep and/or a message box when a User attempts to use the control zone and it is locked.

This property is available in the advanced section of the properties tab.

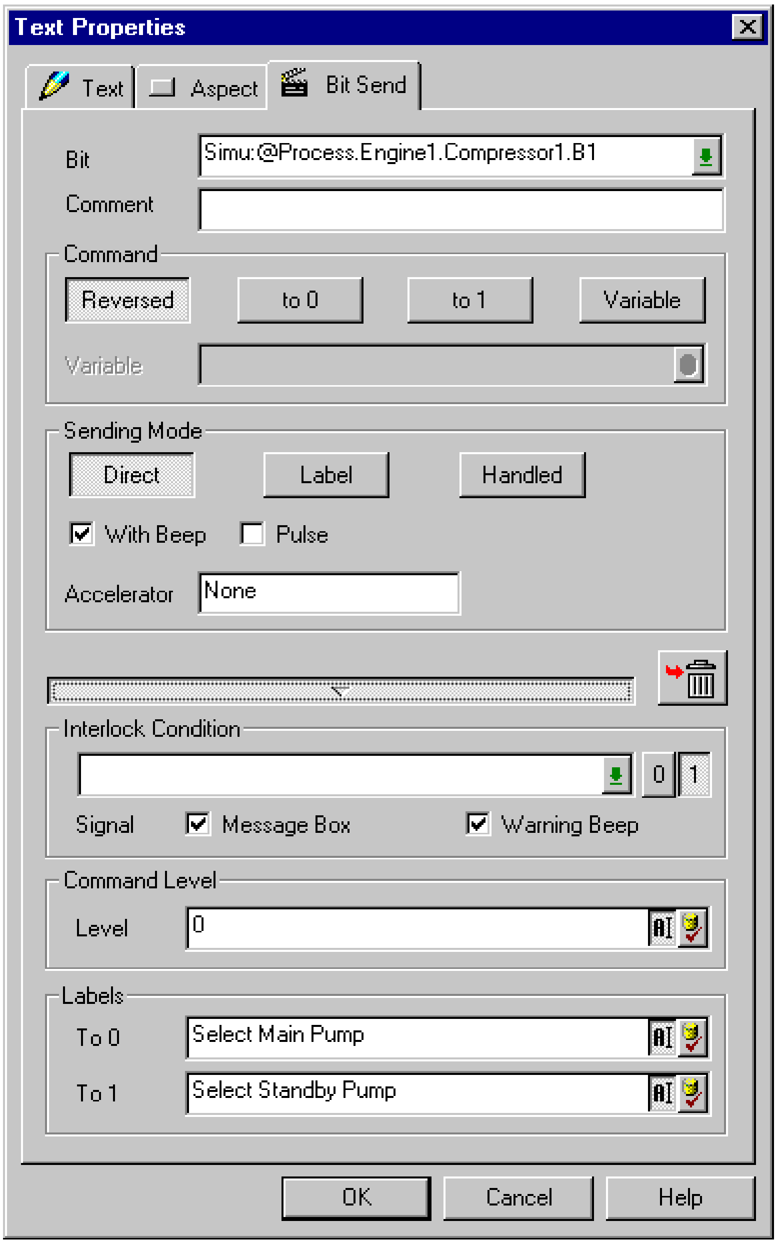
Sending Bits

The Send Bit Animation | The Double Bit Animation

There are two animation's which may be used to change the state of a bit, one operates on a single bit whilst the other operates simultaneously on two bits.

The Send Bit Animation

The Send Bit animation allows the User to force the state of a single bit.



Bit: Simu:@Process.Engine1.Compressor1.B1

Comment:

Command: Reversed

Variable:

Sending Mode: Direct

☑ With Beep ☐ Pulse

Accelerator: None

Signal ☑ Message Box ☑ Warning Beep

Level: 0

To 0: Select Main Pump

To 1: Select Standby Pump

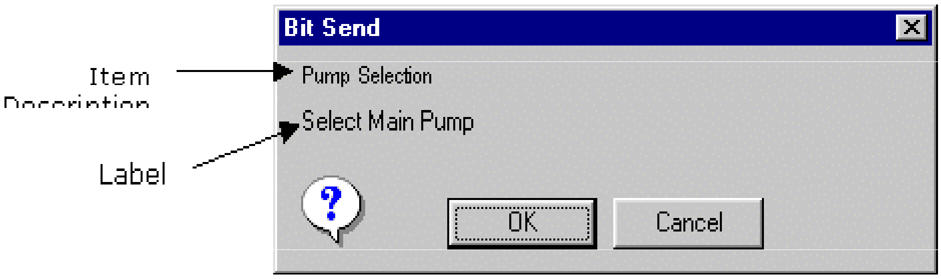
The behaviour of the control zone is determined by the following properties.

- *Reverse* - The control zone sends the bit to the opposite of its current state. That is, if the bit was 0 it becomes 1 or if it was 1 it becomes 0.
- *To 1* - The control zone always forces the bit to 1. If it is already 1 then there is no change.
- *To 0* - The control zone always forces the bit to 0. If it is already 0 then there is no change.

- *Variable* - At run time when the User clicks the control zone the state of the bit to be forced is copied from the state of another bit.
- *Direct* - The control zone sends the bit without confirmation.
- *Label* - See explanation below.
- *Handled* – The control zone behaves as a push to run button. At run time the bit is sent to the requested state whilst the User clicks and holds down the mouse button. When the button is released the bit is sent to the opposite state.
- *Pulse* - Sends the bit to the required state (0, 1 or reverse) for a period defined in seconds. After the period has elapsed the bit is returned to the original state.
- *Beep* - Sound the PC's speaker when the command is made.

Confirming a Command With a Label

If you choose the *Label* send mode, at run time when the User clicks the control zone a box will appear in which the command may be confirmed or cancelled. The box displays the bit item description plus the appropriate string as entered in the Labels section of the Send Bit tab. In this way the User is provided with a native language prompt for the action.



Pump Selection

Select Main Pump

By default the label selection is set to Auto. This means that the HMI will attempt to get them from the variable definition in the server.

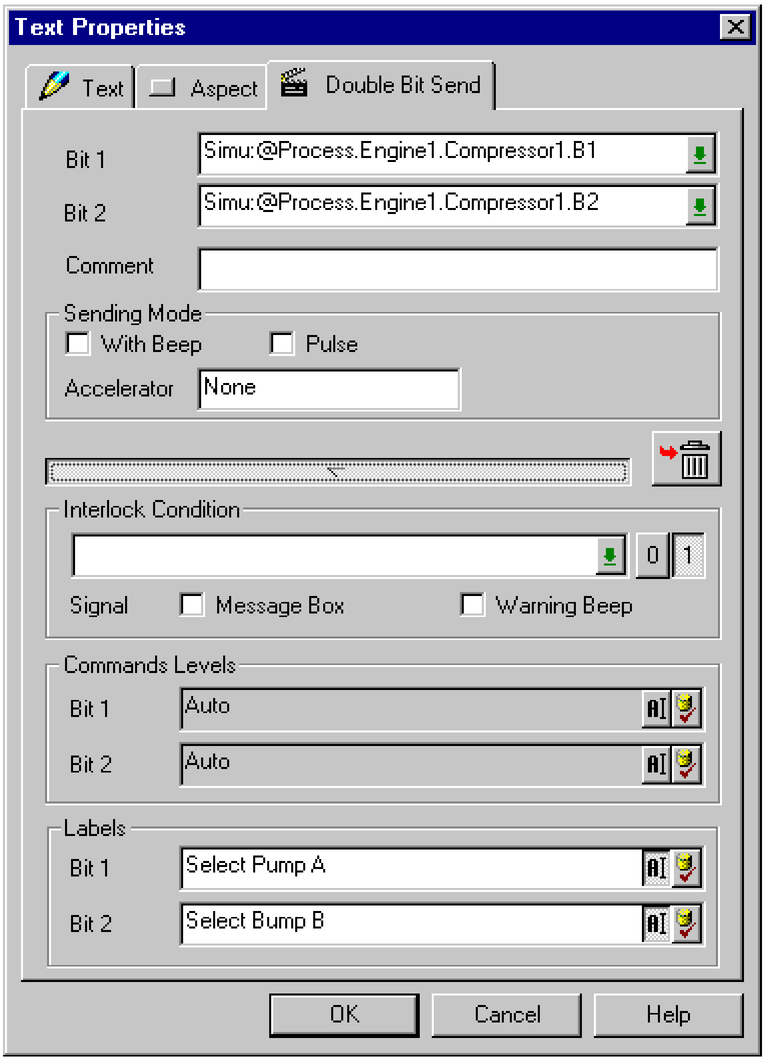
If the server doesn’t support labels or you want to use something different there are two other options accessed by clicking the buttons adjacent to the bound fields.

Clicking this button allows you to enter your own labels for the to 0 and to 1 conditions.

Clicking this button allows the labels to be defined by the real-time value of a text variable in the server. The variable browser is displayed to allow selection of the variable name.

The Double Bit Animation

The Double bit animation allows the User to change the state of 2 bits at the same time. This is useful when a device requires separate signals to start and stop it. The command is always confirmed with a dialog box.



Text Properties

Bit 1: Simu:@Process.Engine1.Compressor1.B1

Bit 2: Simu:@Process.Engine1.Compressor1.B2

Comment:

Sending Mode

☐ With Beep ☐ Pulse

Accelerator: None

Interlock Condition

Signal ☐ Message Box ☐ Warning Beep

Commands Levels

Bit 1: Auto

Bit 2: Auto

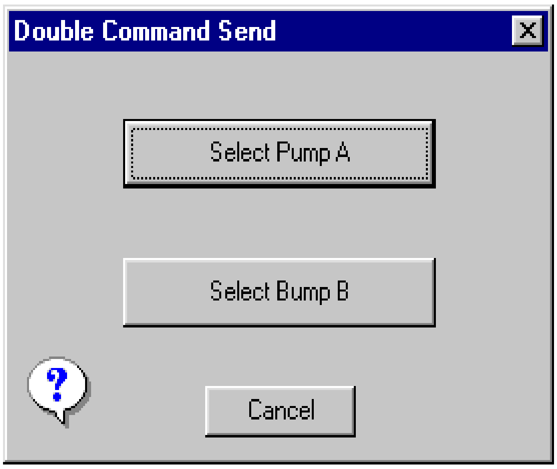
Labels

Bit 1: Select Pump A

Bit 2: Select Bump B

Operation

When the User clicks the control zone at run-time a dialog box appears containing 3 command butto the bit 1, the second the label for bit 2.



Double Command Send

The control operates in two ways depending on if the pulse option has been selected.

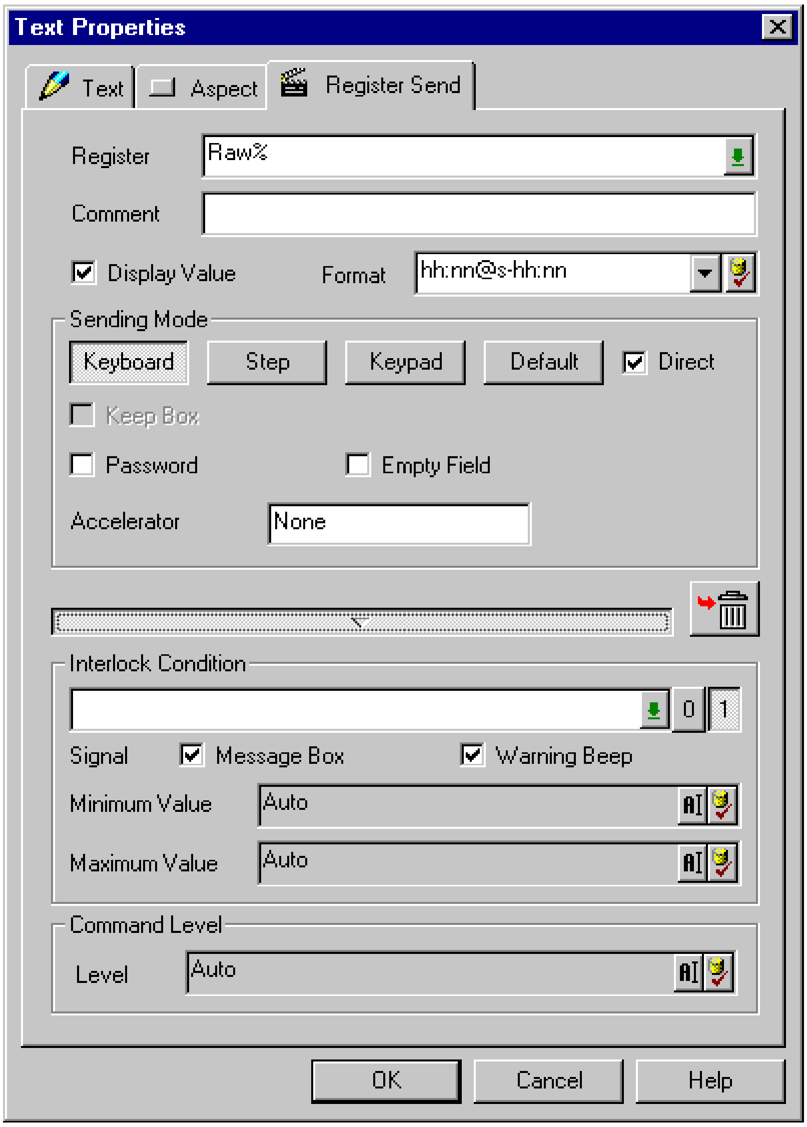
- Without the pulse option. Clicking on the first button will set the first bit to 1 and the second bit to 0. Clicking on the second button will set the second bit to 1 and the first bit to 0.
- With the pulse option. Clicking on the first button will set the first bit to 1 for the defined period after which it will return to 0. The second bit is not affected. Clicking on the second button will set the second bit to 1 for the defined period after which it will return to 0. The first bit is not affected.

The cancel button closes the dialog box without any action.

Entering Register Values

Selecting the Entry Method | Selecting the Output Format | Selecting the Input Format | Selecting the Control Range

The User may change register values from a control zone created using the Control Register animation. Register values may also be changed using a position control. Position controls do not behave in the same way as a normal control zone and are explained in a separate topic. The control value animation allows a User to enter a precise value for a register, either from the keyboard, from a displayed keypad or using positive and negative steps.



Register: Raw%

Comment:

☑ Display Value   Format: hh:nn@s-hh:nn

Sending Mode

☑ Direct

☐ Keep Box

☐ Password   ☐ Empty Field

Accelerator: None

Interlock Condition

0 1

Signal   ☑ Message Box   ☑ Warning Beep

Minimum Value: Auto

Maximum Value: Auto

Command Level

Level: Auto

Selecting the Entry Method

The Control Register animation provides a number of options that effect the way in which the User enters the value at run-time.

- *Keyboard* - The new value is entered by typing it directly into a field in a dialog box.
- *Step* - Similar to keyboard with the addition of plus and minus command buttons with which the User may increase and decrease the value by a defined increment. The dialog box will be automatically closed after each increment, unless the *Keep box* property is selected.
- *Default* - The User is prompted to confirm a (configurable) default value for the register.

- *Keypad* - Similar to keyboard except that a representation of a numeric keypad is displayed as part of the dialog box which is used in conjunction with the mouse to enter a value without using the physical keyboard.
- *Direct* - The value is entered directly into the text field of the support drawing element using the keyboard. This option does not display a dialog box.
- *Password* - If selected, each digit of the entered value is displayed as an asterisk. This allows the User to enter a value without anyone else knowing what it is.
- *Empty field* - If selected, the dialog box does not display the current value of the register in the field used to enter the value.

Selecting the Output Format

If you have chosen to display the register value you may select the output (display) format using the following command buttons adjacent to the format field.

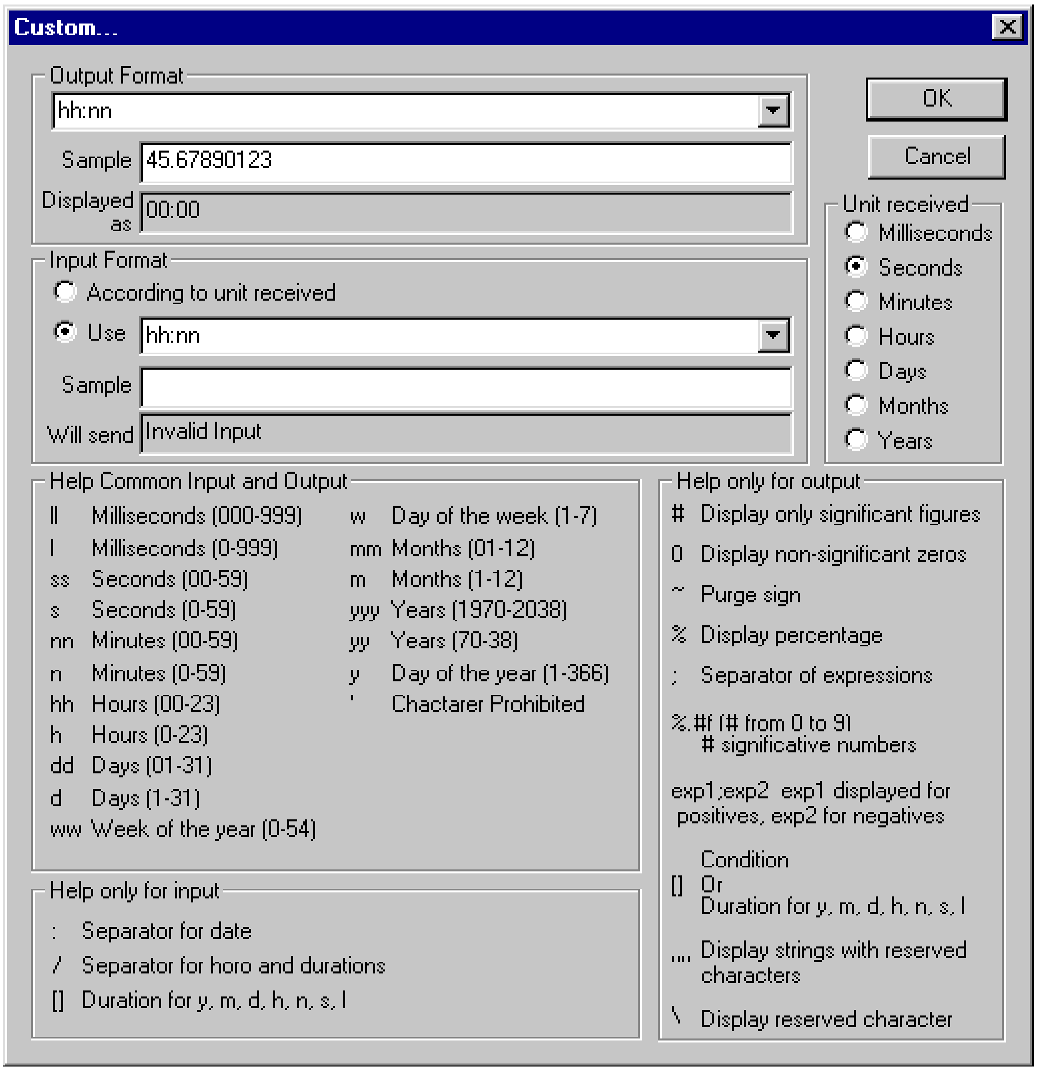
▼ Selects either the auto or custom formats.

Allows the format to be supplied by a variable in the server.

See the Using Custom Output Formats topic for further information.

Selecting the Input Format

If you have selected a time and date style output format then you can also select a format for entering the value. The conversion to the raw value in the register is handled automatically by the software.



Custom...

Output Format

hh:nn ▼

Sample: 45.67890123

Displayed as: 00:00

Input Format

○ According to unit received

● Use: hh:nn ▼

Sample:

Will send: Invalid Input

Unit received

- ○ Milliseconds
- ● Seconds
- ○ Minutes
- ○ Hours
- ○ Days
- ○ Months
- ○ Years

Help Common Input and Output

ll	Milliseconds (000-999)	w	Day of the week (1-7)
l	Milliseconds (0-999)	mm	Months (01-12)
ss	Seconds (00-59)	m	Months (1-12)
s	Seconds (0-59)	yyy	Years (1970-2038)
nn	Minutes (00-59)	yy	Years (70-38)
n	Minutes (0-59)	y	Day of the year (1-366)
hh	Hours (00-23)	'	Chactarer Prohibited
h	Hours (0-23)		
dd	Days (01-31)		
d	Days (1-31)		
ww	Week of the year (0-54)		

Help only for input

:	Separator for date
/	Separator for horo and durations
[]	Duration for y, m, d, h, n, s, l

Help only for output

#	Display only significant figures
0	Display non-significant zeros
~	Purge sign
%	Display percentage
;	Separator of expressions
%.#f (# from 0 to 9)	# significative numbers
exp1;exp2	exp1 displayed for positives, exp2 for negatives
[]	Condition Or Duration for y, m, d, h, n, s, l
""	Display strings with reserved characters
\	Display reserved character

! You can only use custom input formats when you have a time and date style output format selected.

Selecting the Control Range

The *Minimum Value* and *Maximum Value* properties define the input range that is accepted by the animation. By default these properties are set to auto in which case the HMI will attempt to get the values from the OPC server. If either the maximum or minimum values are changed in the OPC server all animations linked to it will automatically follow without the need to change the animation.

If the server doesn't support maximum and minimum values or you want to use a different value there are two other options accessed by clicking the buttons adjacent to the fields.

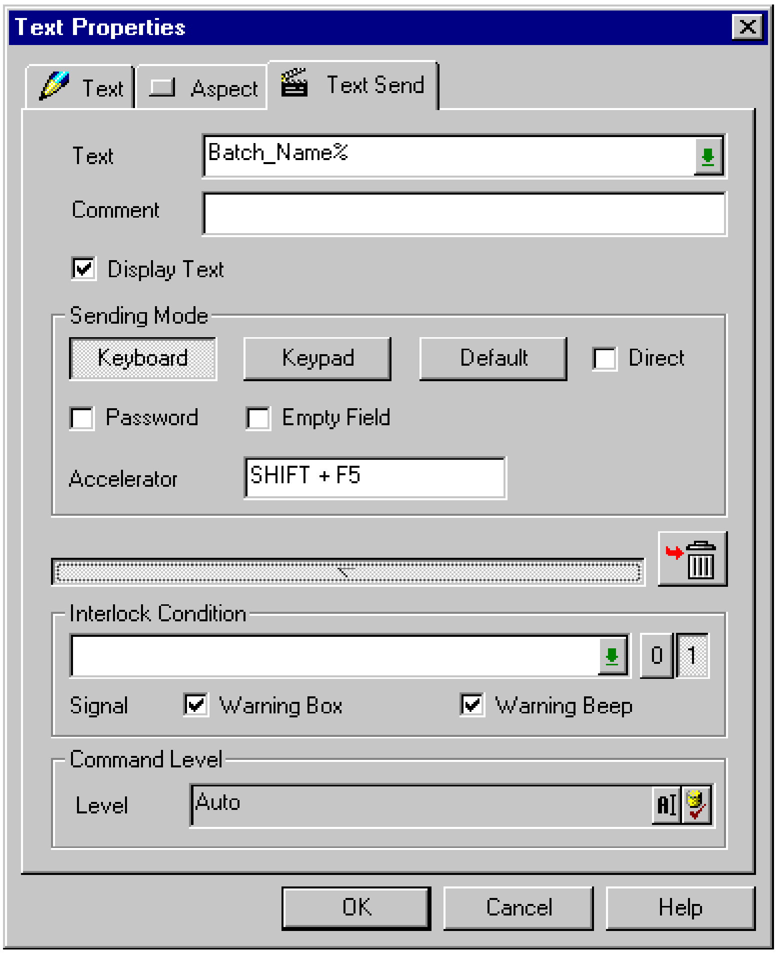
▼ Clicking this button allows you to enter fixed values for the minimum and maximum.

Clicking this button allows the maximum and minimum to be defined by the value of other registers in the server. The variable browser is displayed to allow selection of the variable names.

Entering a String

Selecting the Text Entry Method | Displaying the Text

The User enters strings from a control zone created with the Send Text animation. This animation may be applied to any of the solid drawing elements, but it is normally used with Text in which case the string may be displayed. Strings may be entered either from the keyboard, or by using the mouse and a displayed keyboard.



Text: Batch_Name%

Comment:

☑ Display Text

□ Direct

□ Password

□ Empty Field

Accelerator: SHIFT + F5

0 1

Signal ☑ Warning Box ☑ Warning Beep

Level: Auto

Selecting the Text Entry Method

At run-time, when an User clicks on a Send Text control zone, a dialog box is displayed containing the description of the text variable (or its name if it does not have a description) and the current string. The appearance of the dialog box and the way in which the string is entered is determined by the send mode properties.

- *Keyboard* - The new string is entered by typing it directly into a field in the dialog box.
- *Displayed keyboard* - Similar to *keyboard* except that a representation of a keyboard is displayed as part of the dialog box. This is used in conjunction with the mouse to enter a string without using the physical keyboard.
- *Direct* - The value is entered directly into the text field of the support drawing element using the keyboard. This option does not display a dialog box.
- *Password* - If selected, each character of the entered string is displayed as an asterisk. This allows the User to enter a string without anyone else knowing what it is.
- *Empty field* - If selected, the dialog box does not display the current string of the text variable in the field used to enter the string.

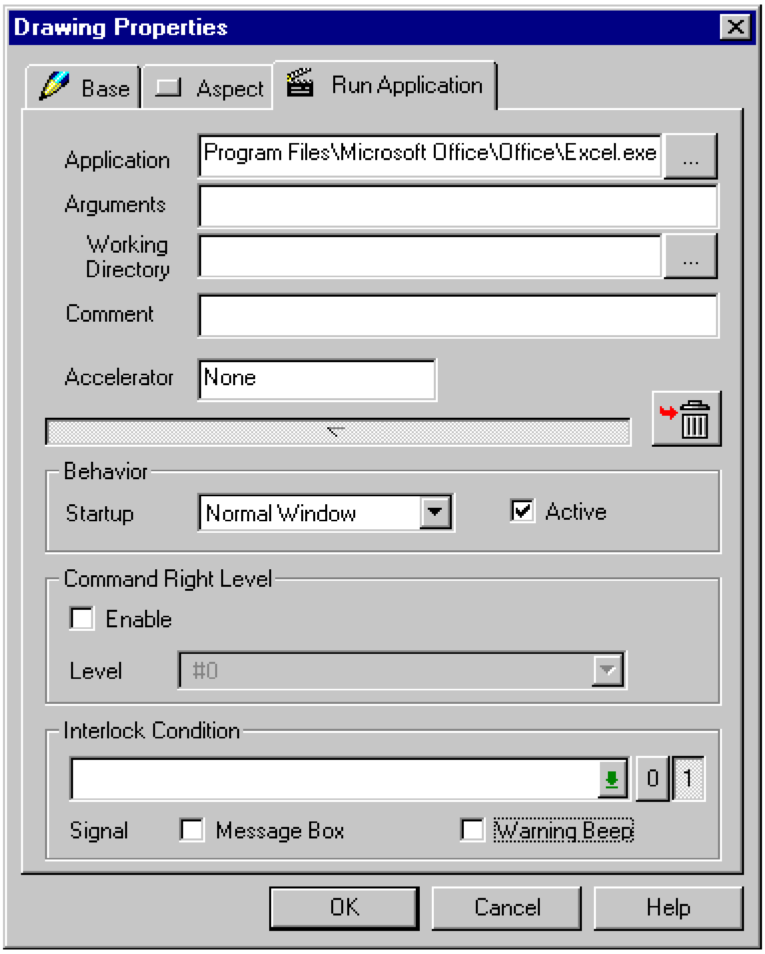
Displaying the Text

If you use a text field as the support drawing element you can display the string by selecting the *Display string* tick box.

Starting an Application

Selecting the Application to Run | Specifying the Start-up Behaviour

The Application animation allows you to start another Windows' application, such as Notepad, directly from a control zone.



Application: Program Files\Microsoft Office\Office\Excel.exe

Arguments:

Working Directory:

Comment:

Accelerator: None

Startup: Normal Window

☑ Active

☐ Enable

Level: #0

☐ Message Box

☐ Warning Beep

Selecting the Application to Run

- *Application* - The full path and name of the executable program required to run the application. The ellipsis button opens a file browser.
- *Arguments* - Optional command line arguments to be passed to the application. For example the name of a file to open.
- *Working directory* - The path of the directory that contains the original files or some related files. Sometimes an application needs to use files from a different location to that of its executable program.

Specifying the Start-up Behaviour

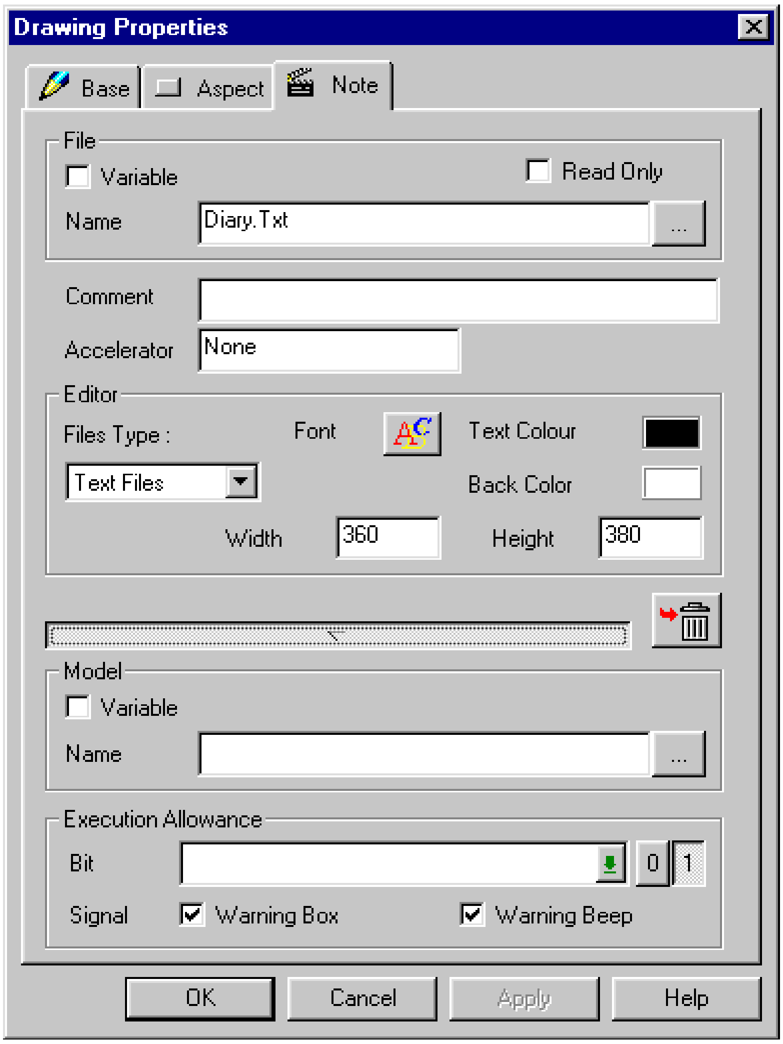
The start-up behaviour property determines the appearance of the application when it is started. It may be selected from the following options.

- Normal window - The application starts in a window, the size of which is normal for the application.
- Minimised - The application starts minimised. Depending on the access rights of the User and the configuration of the operating system the icon may not be visible or accessible.
- Full screen window - The application starts in a window occupying the whole screen.

Displaying a Text Note

Selecting the File to Open | Using a File Mode

The Note animation allows you to display, modify and save the contents of a text file. The text file is displayed in a window supporting the standard Windows' edit window keystrokes for text selection.



File
□ Variable
□ Read Only
Name: Diary.Txt
Comment:
Accelerator: None

Editor
Files Type : Text Files
Font
Text Colour ■
Back Color □
Width: 360
Height: 380

Model
□ Variable
Name:

Execution Allowance
Bit: 0 1
Signal ☑ Warning Box ☑ Warning Beep

Selecting the File to Open

In the Note properties tab you can either enter the name of the file to open or the name of a text variable that contains the name of the text file. If you omit the path for the file it is assumed that it is in the root directory of the active project.

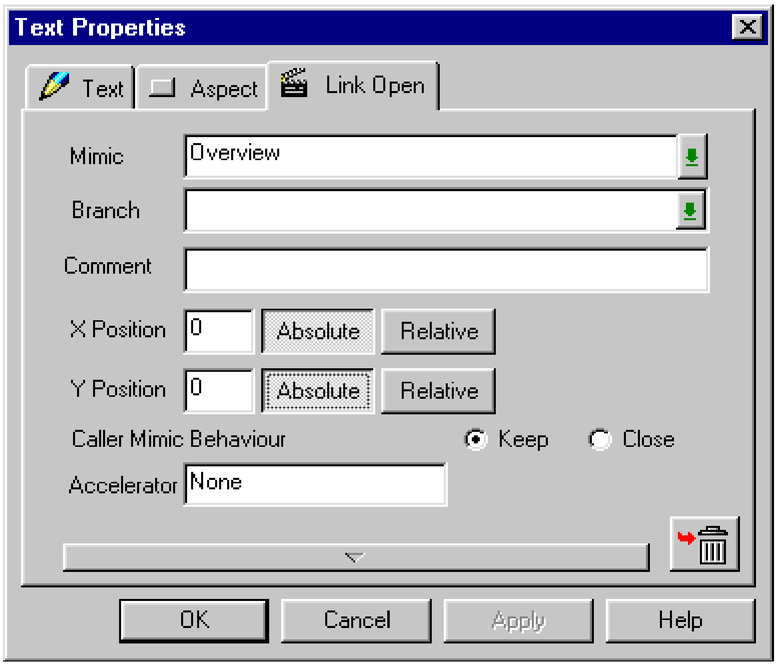
Using a File Model

If you want to create a new file that is based on the contents of an existing file you can specify the name of a model file. At run time, when the animation is selected, the model is opened and its contents placed in the file specified by the file name. You can then modify and save this file. The model file is not changed.

The Link Open Animation

Using Relative or Absolute Opening Positions | Opening a Mimic With a Branch | Opening a Child Mimic | Using Substitution Characters

The Link Open animation may be applied to any drawing element although normally it will be applied to a text string with the text used as an indication to the Operator.



Mimic: Overview

Branch:

Comment:

X Position: 0 Absolute Relative

Y Position: 0 Absolute Relative

Caller Mimic Behaviour ● Keep ○ Close

Accelerator: None

The name of the mimic to be opened is selected using the *Mimic* drop down list box.

Using Relative or Absolute Opening Positions

You specify the position at which the mimic is opened using the *X Position* and *Y Position* properties.

If you select the *Relative* property the position properties, plus the location that the mimic was originally saved in, determine the position in which the mimic is opened. For example if the mimic was saved at position 50, 70 and a relative position of 30, 10 is supplied, the mimic will be opened at 80, 80.

If you select the *Absolute* property the mimic is always opened at the location specified in the animation - the position in which the mimic was originally saved is ignored.

Opening a Mimic with a Branch

When a Link Open animation is configured an optional branch may be supplied which is then used with the mimic when it is opened. Subsequently any reference to variables within that mimic is relative to that branch. Used in conjunction with the structured server this technique can be used to reduce the number of mimics required and the subsequent development time.

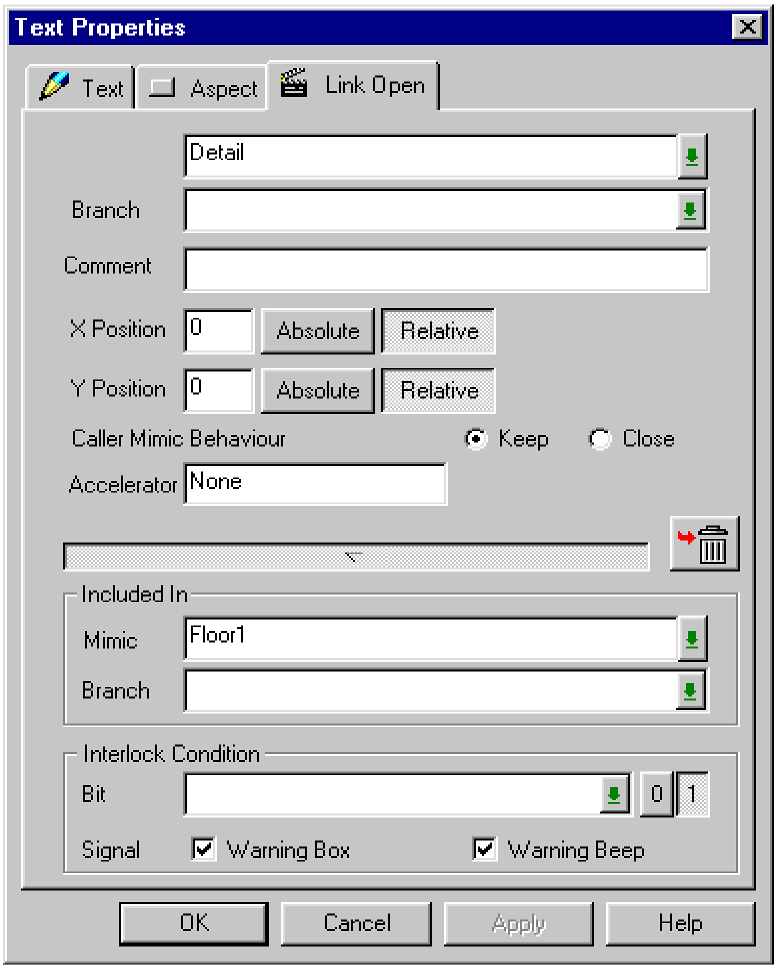
Several copies of a mimic may be opened at the same time as long as they have different branches.

Opening a Child Mimic

You can open a mimic as a child of another, already open, mimic. Child mimics behave in a slightly different way.

- Child mimics stay on top of the parent mimic until it is closed. You cannot move a parent mimic on top of a child mimic.
- When you close the parent mimic any child mimics also close.
- A child mimic may only be moved within the bounds of the parent mimic.

To open a mimic as a child you must specify the name of the parent mimic using the *Included in - Mimic* property. Note that if the parent mimic has been opened with a branch then this also must also be specified.



Detail

Branch

Comment

X Position 0

Y Position 0

Caller Mimic Behaviour ● Keep ○ Close

Accelerator None

Included In

Mimic Floor1

Branch

Interlock Condition

Bit 0 1

Signal ☑ Warning Box ☑ Warning Beep

If you use the substitution character * as the name of the “included in” mimic the mimic will always be opened as a child of the mimic in which the animation resides. Similarly if you use * for the branch, the child mimic will inherit the branch of the mimic in which the animation resides. This is very useful when including the animation in an object that may be inserted in any mimic.

Using Substitution Characters

Using substitution characters you can make a mimic open animation context sensitive.

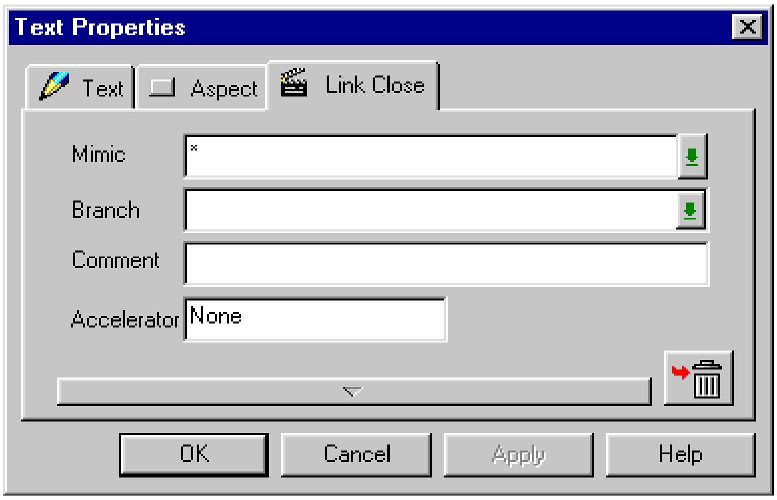
String	Opens
#P	The previous mimic that was opened.
#M1 to #M10	The mimics configured in host mimics Link tab.

If you base the Link-open animation on a text drawing element, and you are using one of the substitution strings #M1 to #M10, at run-time the label specified with the mimic name will be displayed in the text drawing element.

This technique may be used to create a standard menu object containing up to 10 Link-open animations. The text displayed in the support drawing elements, and the mimics opened will depend on the link properties of the mimic in which the object is placed.

The Link Close Animation

The Link Open animation may be applied to any drawing element although normally it will be applied to a text string with the text used as an indication to the Operator.



Mimic: *

Branch:

Comment:

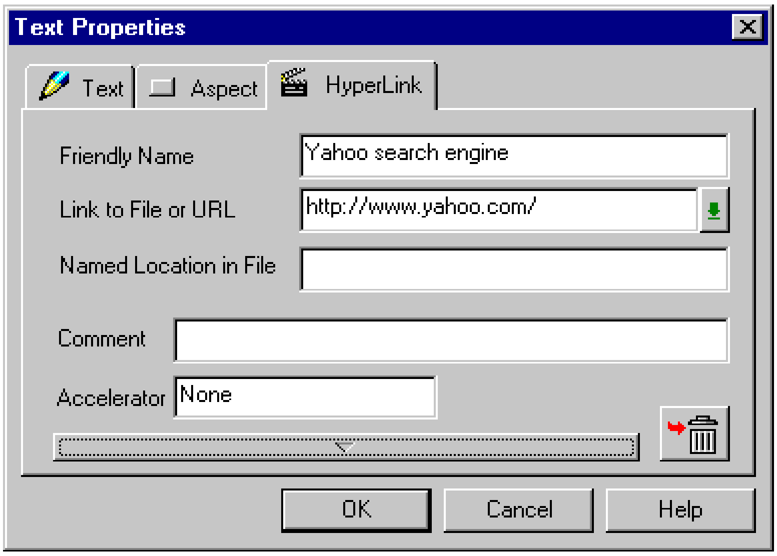
Accelerator: None

You specify the name of the mimic to be closed in the *Mimic* field. If the mimic has a branch then this also must be specified.

If you use the substitution character * as the mimic name, the animation will close the mimic in which it resides. This is very useful when including the animation in an object that may be inserted in any mimic.

The Hyperlink Animation

The Hyperlink animation only operates at run-time when the mimic containing it is opened in an ActiveX document container such as the Microsoft Explorer. It can be used to provide a link to a file (such as another mimic or a Word document) or a URL (Universal Resource Locator).



Friendly Name: Yahoo search engine

Link to File or URL: http://www.yahoo.com/

Named Location in File:

Comment:

Accelerator: None

You can also choose to go to a specific location within the file, for example to a bookmark in a Word document, by entering the reference in the *Named Location* field.

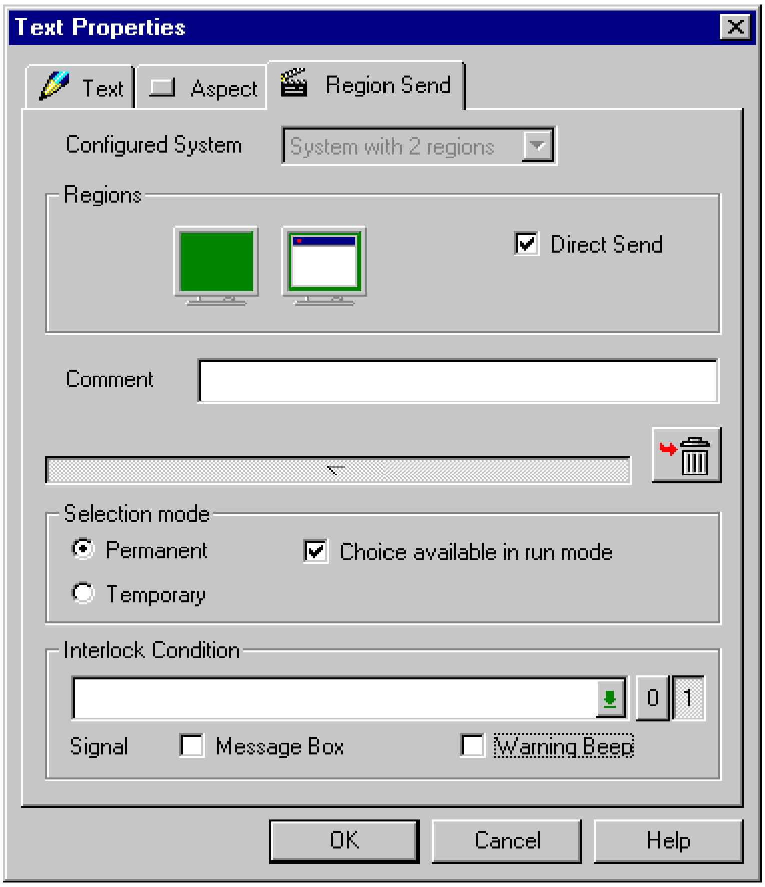
If used, the *Friendly Name* will be substituted for the File or URL name in the history of the Microsoft Explorer.

The Region Animation

The region animation is used to select where a mimic is opened on projects with more than one region. Regions are normally used with graphic adapters that support more than one screen in which case the number of regions is selected to match the number of screens. For example on a system with two regions, if you select the right region and then open a mimic at position 0,0 it will open with the top left hand corner at the top left hand corner of the right screen.

The number of regions is selected from the workspace properties box. The region in which a mimic is opened is selected using the Region animation.

The behaviour of mimics when using regions is affected by some of the properties in the options box displayed from the Configure.Options command.



Configured System: System with 2 regions

☑ Direct Send

● Permanent

○ Temporary

☑ Choice available in run mode

Signal ☐ Message Box ☐ Warning Beep

The region animation may be configured so that the User is allowed to select the region in which a mimic is opened, or so that the region is fixed. The default region in which the next mimic is opened is selected by clicking on the corresponding icon at the top of the dialog box. The following properties are available.

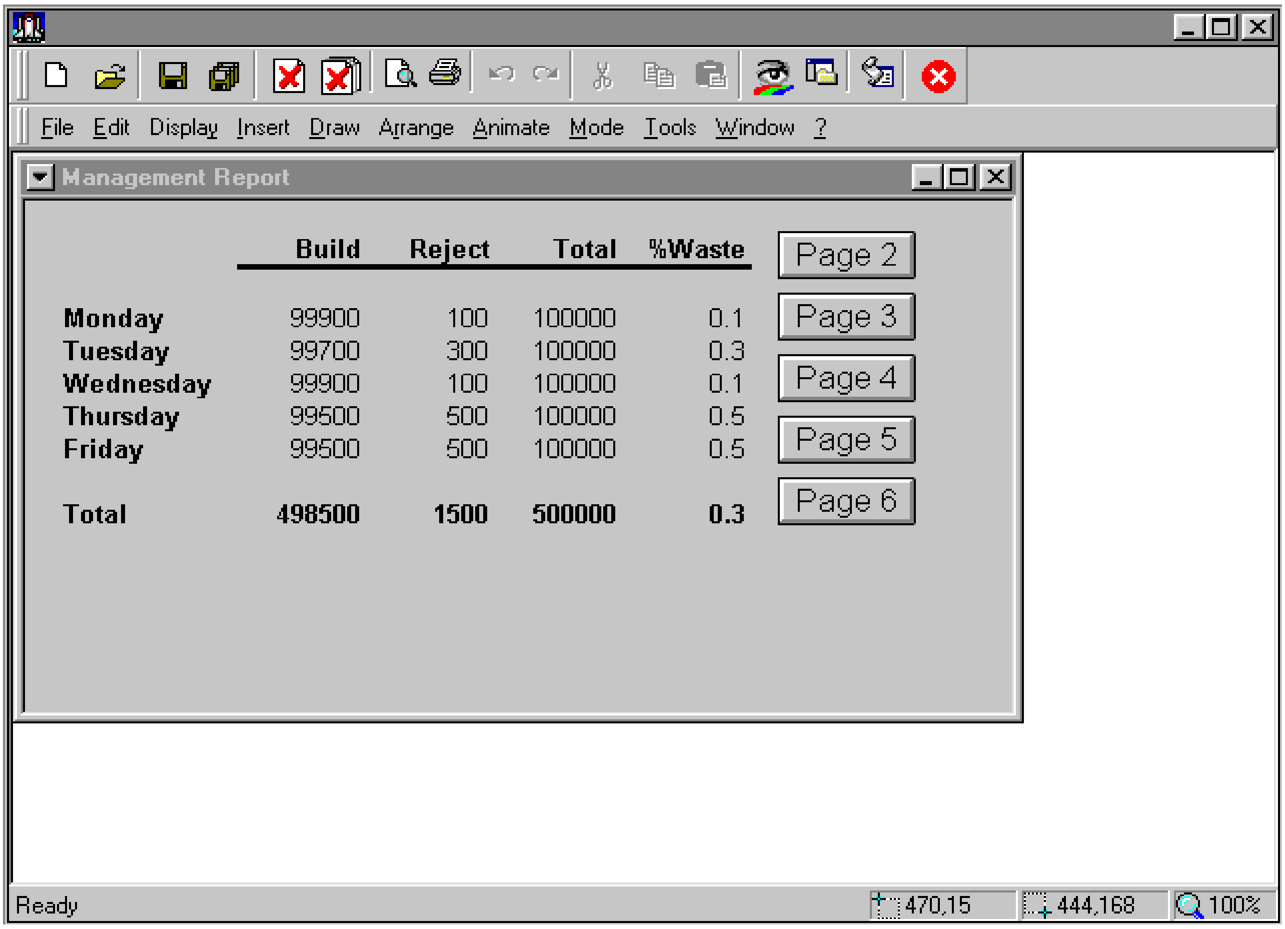
- *Direct send* - The region is selected without allowing User selection.
- *Permanent* - The selection is permanent until the next region animation or instruction is used.
- *Temporary* - The next mimic open uses the selected region after which the region reverts to the previous one. For example on a two region system the current region is 1 and then region 2 is selected using the region animation. The next mimic to open would be in region 2, the following one would be in region 1.
- *Choice available in run mode* - At run time the User is allowed to select the temporary or permanent properties. The direct send property must not be selected.

What is OLE

Using object linking and embedding you can include information created in other applications, such as a spreadsheet or word processor, in the Supervisor's windows. The included information is known as an object.

The main difference between linking and embedding is where the data is stored.

- Embedded objects become part of the window source file.
- Linked information, remains stored in the original source file with the window containing only the location of the file whilst still displaying a graphic representation of the linked data.



	Build	Reject	Total	%Waste
Monday	99900	100	100000	0.1
Tuesday	99700	300	100000	0.3
Wednesday	99900	100	100000	0.1
Thursday	99500	500	100000	0.5
Friday	99500	500	100000	0.5
Total	498500	1500	500000	0.3

Embedding an Object

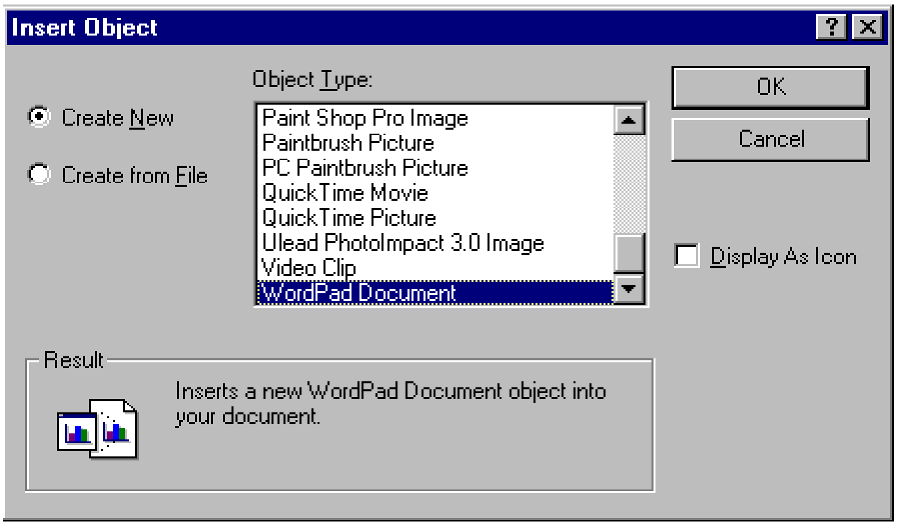
Embedding a New Object | Embedding an Object Using an Existing File | Embedding an Object Using Copy & Paste | Changing an Embedded Object

Embedding means inserting information such as an Excel chart or a Word document into a window. Once embedded, the information, or object, becomes part of the window. When you double-click an embedded object in edit mode, you open the application in which the object was created; the object is displayed and ready for editing. When you return to the Supervisor, the window reflects any changes you've made to the object. You can embed an object using an existing file, or you can create and embed a new object.

To edit an embedded object created in another application, you must have the other application installed on your computer.

Embedding a New Object

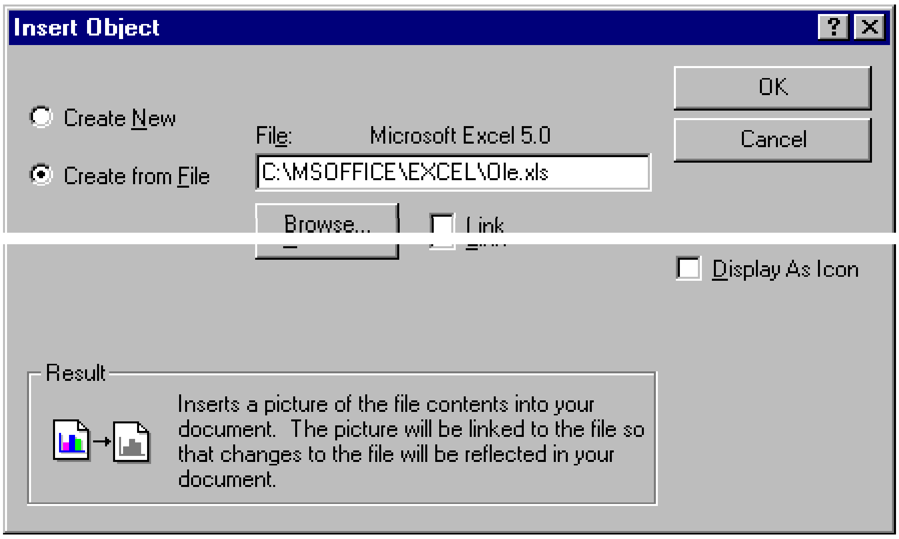
1. Open or select the window in which the object is to be embedded.
2. From the EDIT sub-menu select the Insert Object command.
3. In the Object Type box, select the type that describes the application in which you want to create the object.
4. Click the OK button.
5. A new object of the selected type will be created embedded in your window and the associated application opened to allow you to edit the new object. The application's toolbars will appear within the Workspace.
6. Move and size the object as required.
7. Enter the information you want to display in the new object. When you have finished click anywhere outside of the object - the associated application will close and the Supervisor's window will be updated.



Inserts a new WordPad Document object into your document.

Embedding an Object Using an Existing File

1. Open or select the window in which the object is to be embedded.
2. From the EDIT sub-menu select the Insert Object command.
3. Select the Create from file option button.
4. In the File box type or select the name of the object that you want to embed. You can use the Browse button to select a different drive or directory.
5. Click the OK button.



File: C:\MSOFFICE\EXCEL\Ole.xls

Inserts a picture of the file contents into your document. The picture will be linked to the file so that changes to the file will be reflected in your document.

Embedding an Object Using Copy & Paste

1. Open the application (Word, Excel etc.) with which the object is associated.
2. Open the application file.
3. Copy all or part of the application file to the clipboard. (Usually done using the Copy command on the EDIT menu or Ctrl C.)
4. Close the application.
5. Open or select the Supervisor window in which the object is to be embedded.
6. Paste the contents of the clipboard into the window by using the Paste command from the EDIT sub-menu or Ctrl V on the keyboard.

Changing an Embedded Object

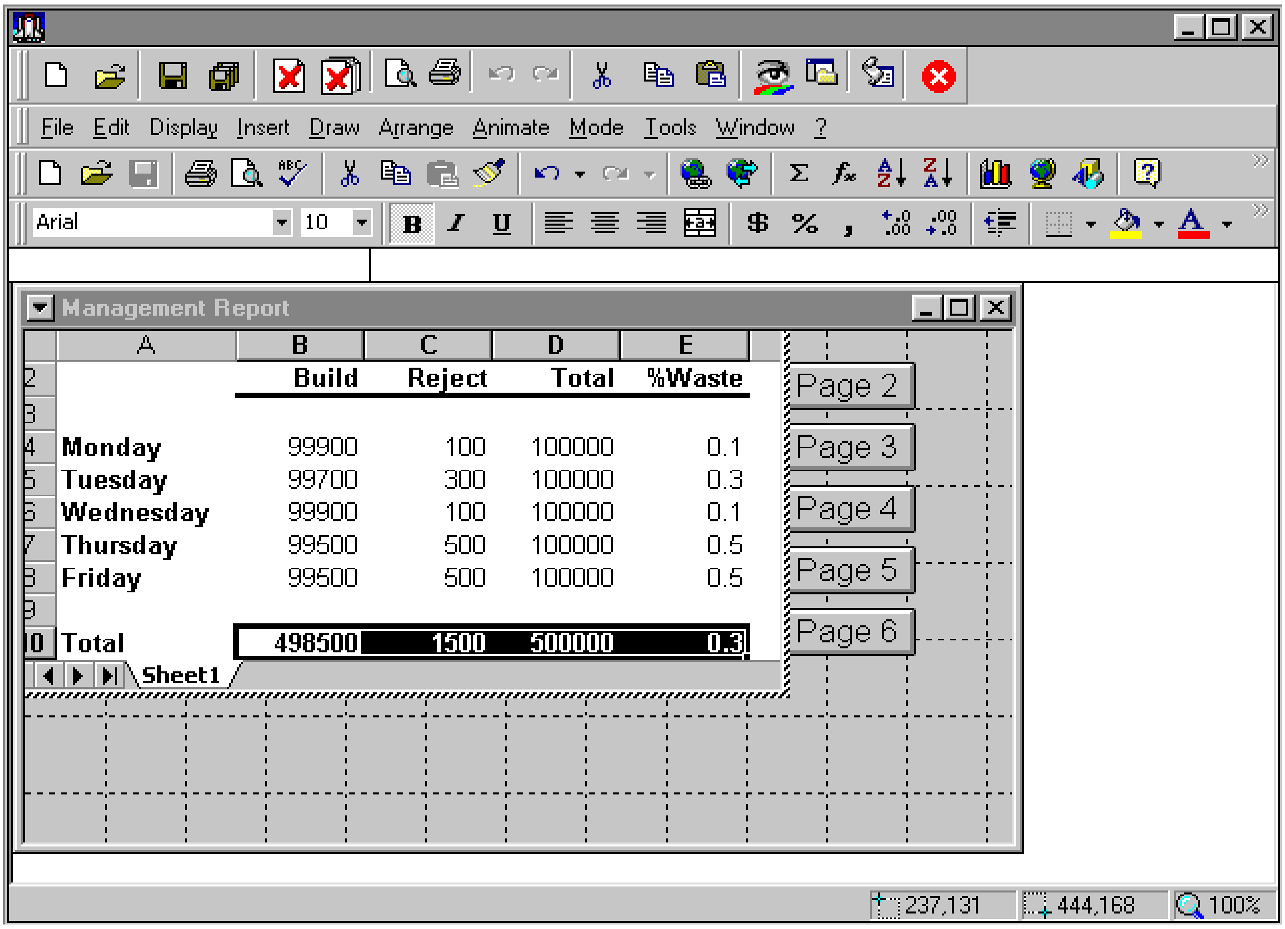
There are two ways to change an embedded object.

- You can edit it directly in the window in which it is embedded.
- You can open it in the original application.

An embedded object is surrounded by a solid border when a window is in edit mode.

To edit an embedded object

1. Open or select the window in which the object is embedded and select edit mode.
2. Double click on the embedded object. The object will be surrounded with a hatched border and the toolbars from the objects original application will be displayed within the Workspace.
3. Make your changes to the object.
4. When finished simply click anywhere outside of the object or its toolbars. The toolbars will be closed and the object returned to its static state.



	A	B	C	D	E
2		Build	Reject	Total	%Waste
3					
4	Monday	99900	100	100000	0.1
5	Tuesday	99700	300	100000	0.3
6	Wednesday	99900	100	100000	0.1
7	Thursday	99500	500	100000	0.5
8	Friday	99500	500	100000	0.5
9					
10	Total	498500	1500	500000	0.3

! When editing an object in this way many of the original application's tools may not be operable.

To open an embedded object

1. Display the Drawing pop-up menu by clicking on the object using the right mouse button.
2. The last entry in the menu is context sensitive dependent on the type of object you have selected. For example for a spreadsheet it will be Spreadsheet Object. Attached to this last entry will be a further sub-menu with the choice of Edit or Open. Select Open.
3. The object will now be opened within the original application where you can then make any required changes.
4. To return to the Supervisor when you have finished making any changes select Quit or Exit from the application's File menu.

Linking an Object

Changing a Linked Object

Linking means inserting the location of a source of information such as an Excel chart or a Word document into a window. Linked information, remains stored in the original source file although the window still displays a graphic representation of the data as with embedding.

As an example we can consider the situation where the management of a factory produces a report in an Excel spreadsheet of the production targets for the following week. The report is updated each day with revised targets. In one of the Supervisor's windows, a link to the spreadsheet object is made. The graphical view of the spreadsheet in the Supervisor's window is automatically updated each time the management revise the spreadsheet.

How to link an object

1. Open or select the window in which the object is to be embedded.
2. From the EDIT sub-menu select the Insert Object command.
3. Select the *Create from file* option button.
4. In the *File* box type or select the name of the object that you want to embed. You can use the browse button to select a different drive or directory.
5. Tick the *Link* tick box.
6. Click the OK button.

Changing a Linked Object

You can only change a linked object from within the original application. To do this you can either :

- Double click on the object whilst the window is in edit mode.

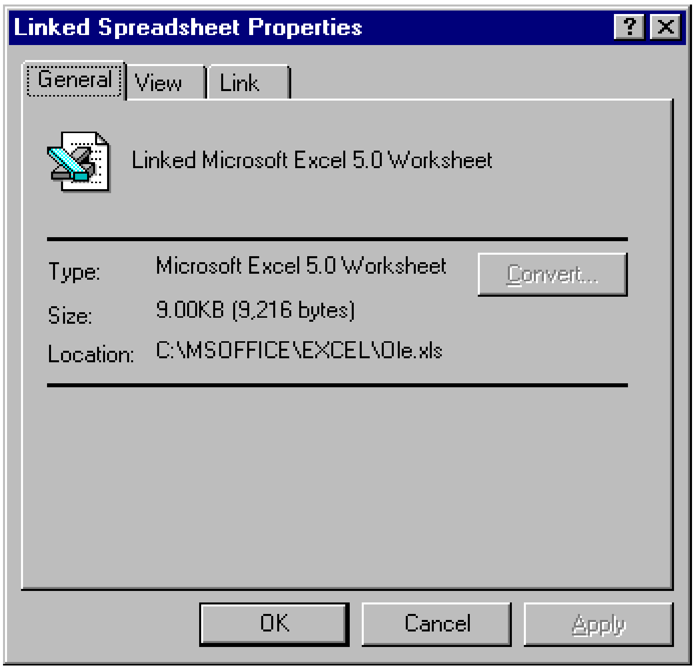
Or

- Open the original application and edit the original file.

A linked object is surrounded by a dashed border when a window is in edit mode.

The OLE Object Properties Box

The OLE object properties box is displayed by clicking on the object with the right mouse button and then selecting the Properties command from the displayed pop-menu.



Linked Microsoft Excel 5.0 Worksheet

Type: Microsoft Excel 5.0 Worksheet

Size: 9.00KB (9,216 bytes)

Location: C:\MSOFFICE\EXCEL\Ole.xls

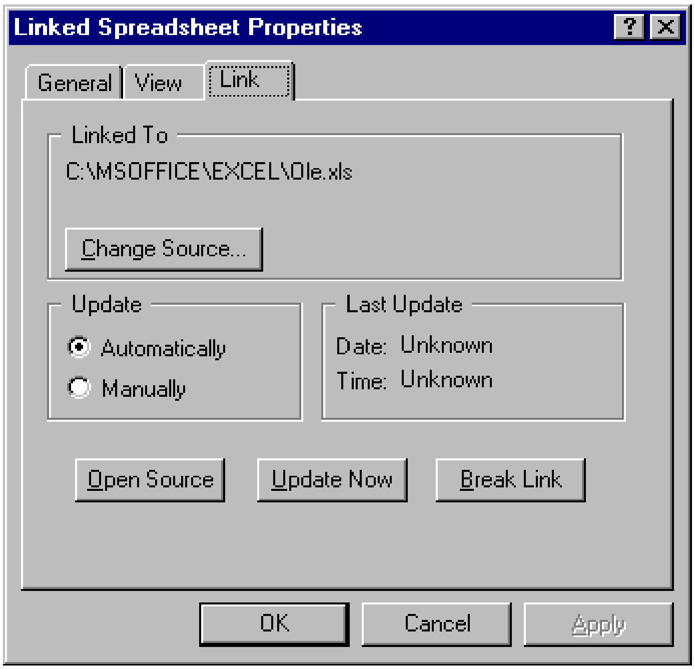
The appearance of the box may vary depending on the type of object. All objects will have the General and View tabs. Linked objects will have the additional Link tab.

The General Tab displays the type of object, its size and location. For a linked object the location will be that of the original file. For an embedded object the location will be the file of the host window.

The View Tab allows you to select the display of the object as editable information or as an icon. Displaying the object as an icon has little use in the Supervisor as you cannot double click on it to reveal the information at run-time.

The Link Tab

The Link tab is only displayed for linked objects.



Linked To
C:\MSOFFICE\EXCEL\Ole.xls

Update
● Automatically
○ Manually

Last Update
Date: Unknown
Time: Unknown

Command Buttons

- *Change source* - Displays a standard Windows' file selection box to allow you to choose a new source for the object.
- *Open Source* - Opens the source file for editing using the original application.
- *Update Now* - Updates the display of the object in the Supervisor's window with any changes that have been made.
- *Break Link* - Breaks the link between the display in the Supervisor's window and the source file. The display of the object within the window retains its appearance but becomes a non-editable Windows' Metafile.

What is a Symbol?

A Graphical Symbol is a collection of drawing elements and associated animations that have been grouped together and stored in a library. From the library, a symbol may be pasted into any mimic as many times as it is needed. A symbol may represent anything from a simple close button for a mimic, to a very complex piece of equipment with hundreds of animations. No special tools are needed to create Symbols; they are created in the same way as any other drawing using the Workspace tools. Using symbols in your project has a number of benefits.

- Repetitious drawing is avoided considerably reducing development time.
- The Developer may create his own reusable symbol library specific to industry sector in which he operates.
- When a symbol is pasted into a mimic, it is done as a reference. If a modification is to be made to the operation of a symbol, the changes need only be made to the symbol in the library, all instances of it will automatically be modified.

Direct, Relative and Deferred Referencing

When a symbol is created you can choose to use direct, relative or deferred referencing to the variables in the symbol's animations. In fact it is possible to mix all three types if necessary.

With direct referencing any variable names used in an animation are entered in their entirety. For example you could use the name `Simu:@Process.Engine1.Compressor1.Bit1`. Note that the name is prefixed with an @ symbol. This tells the HMI that this is a directly referenced name. Whenever the symbol is re-used the animation will always refer to the same variable irrespective of any branch.

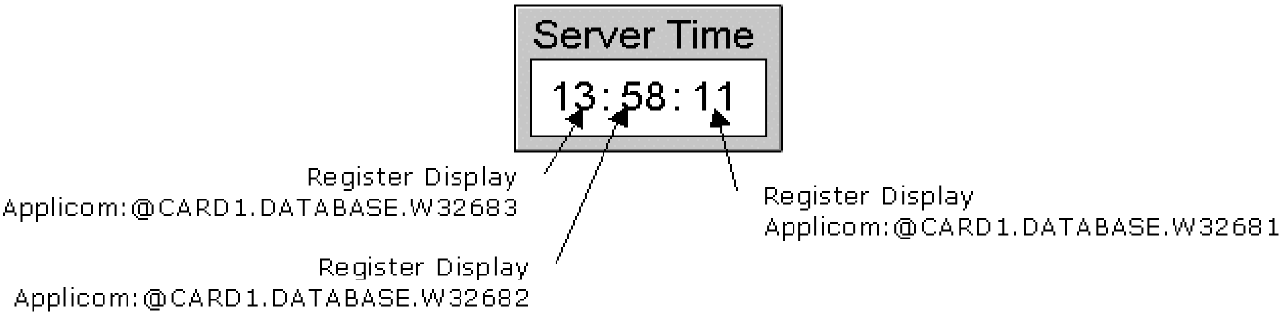
With relative referencing, each animation refers to just the last part of the variable name. For example you might use the name `M1`. The rest of the variable name is supplied as a branch when the symbol is pasted into a mimic. The variable names are automatically calculated at run-time by concatenating the branch with the relative name used in the animation. You can only make use of relative referencing if the server you are using has a structured naming scheme.

With deferred referencing, temporary variable names are used in each animation. Temporary variables only exist while an animation using them is displayed and do not appear in the server. Temporary variable names always end with a percent % character, for example `TEMPMV1%`. When a symbol containing temporary variables is pasted into a mimic a dialog box is displayed allowing the temporary variables to be replaced with actual variables in the server. This does not effect the symbol in the library, just the use of it in the mimic. The same symbol can be used any number of times, each time with a different set of variables.

Creating a Direct Referencing Symbol

The following example describes the creation of a simple symbol. Whilst this is a specific example the same principle will be used when creating any symbol. The symbol is to use direct referencing to server variables, that is, when it is re-used it will not use a branch.

The symbol will use the Register Display animation to display three registers representing the time in hours, minutes and seconds.


Server Time
13:58:11
Register Display
Applicom:@CARD1.DATABASE.W32683
Register Display
Applicom:@CARD1.DATABASE.W32681
Register Display
Applicom:@CARD1.DATABASE.W32682

1. Create a new mimic to use as a work area in which to create the symbol. The mimic may be deleted when you have finished.
2. In design mode use the drawing elements Text and Rectangle to draw the required shape. Draw the three Text elements that will be used to display the dynamic information.
3. Apply the three animations. When referencing the variables ensure the full name is used preceded by the @ character.
4. Select all the drawing elements and use the Group command to create a drawing group. The drawing elements will be surrounded by 8 large black squares when they have been grouped.
5. Select the Edit.Create Symbol command. A dialog box will be displayed prompting for the symbol name. If you have multiple libraries then a list of the available libraries is also displayed.
6. Choose the library and enter the symbol name and select OK. You have now created a symbol.

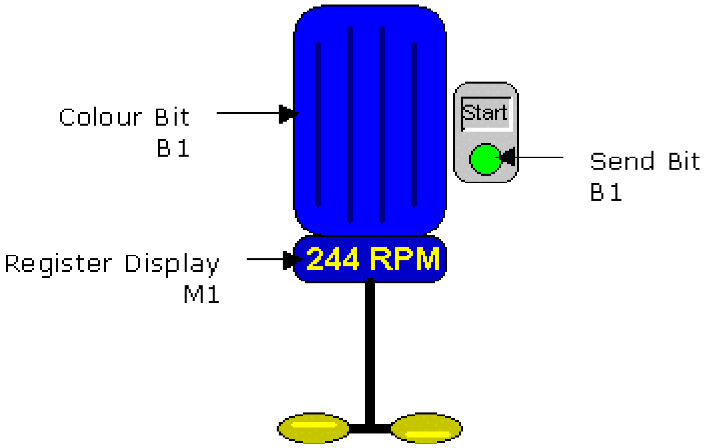
The symbol will now appear in the chosen library from where it may be pasted into any mimic.

Creating a Relative Referencing Symbol

More About Relative Referencing

The following example describes the creation of a symbol using relative variable referencing.

The symbol is to represent a motor attached to a paddle stirrer. The symbol is to display the motor status and speed. A control zone will allow a User to start and stop the motor. The completed symbol might look something like this.



1. Create or open a mimic to use as a work area in which the symbol will be created. The mimic must be opened with a valid branch, for example `Simu:Process.Engine1.Compressor1`, as the HMI checks the existence of a variable when using it in an animation.
2. In design mode draw the static shape.
3. Apply the animations. When entering the variables only enter the relative part of the name. The variable browser will do this automatically, or you can enter the name manually.
4. Select all the drawing elements and use the Group command to create a drawing group. Grouped drawing elements are surrounded by 8 large black squares when selected.
5. Select the Edit.Create Symbol command. A dialog box will be displayed prompting for the symbol name. If you have multiple libraries then a list of the available libraries is also displayed.
6. Choose the library and enter the symbol name and select OK.

More About Relative Referencing

In the previous example only the last part or the leaves of the variable names were used in the symbol. However there is no restriction to how much or how little of a variable name you use as long as when the variable reference is concatenated with the branch it produces a valid variable name. For example, in the above symbol we could have used `Compressor1.B1`, `Compressor1.M1` etc. and when the symbol was pasted into a mimic used the branch `Simu:Processs.Engine1`.

In some cases, a symbol might appear in a mimic that has been opened with a branch. In this case the mimic branch, symbol branch and variable reference in the animation are all concatenated to provide the final variable name.

Pasting a Symbol Into a Mimic

Displaying a Symbol's Properties | Using Symbols with Deferred Referencing | Re-sizing a Symbol

Before pasting a symbol into a mimic it must be in design mode. There are then a number of methods by which a symbol may be pasted into a mimic.

- Using the Insert.Symbol command.
- By dragging and dropping the symbol from another mimic.
- By using the standard Mimic's copy (Ctrl C) and paste (Ctrl V) commands to copy the symbol to either the same mimic or a different mimic.
- By using the Copy tool from the Standard toolbar to copy the symbol to either the same mimic or a different mimic.
- By dragging and dropping it from the library browser.

How to Paste a symbol Using the Insert.Symbol Command

1. Select the Insert.Symbol command. A box containing a list of symbols, referenced by name, will be displayed. A lexical filter may be applied to reduce the number of symbols displayed in the list.
2. If using multiple libraries select the relevant library.
3. Select the symbol name from the scrollable list. A visual preview of the symbol is displayed adjacent to the list.
4. If the symbol uses relative variables then enter a valid branch.
5. Paste the symbol into the mimic by selecting the OK button.

How to Drag and Drop a symbol From Another Mimic

1. Using the mouse click on the symbol in the source mimic with the left mouse button. Hold down the mouse button and drag the symbol to its new mimic - release the mouse button when it is in the correct position.
2. When the mouse button is released a dialog box will be displayed showing the names of any variables and allowing a branch to be entered. Enter a branch (if required) and select the OK button. The symbol has now been moved to the new mimic.

How to Copy a symbol From Another Mimic

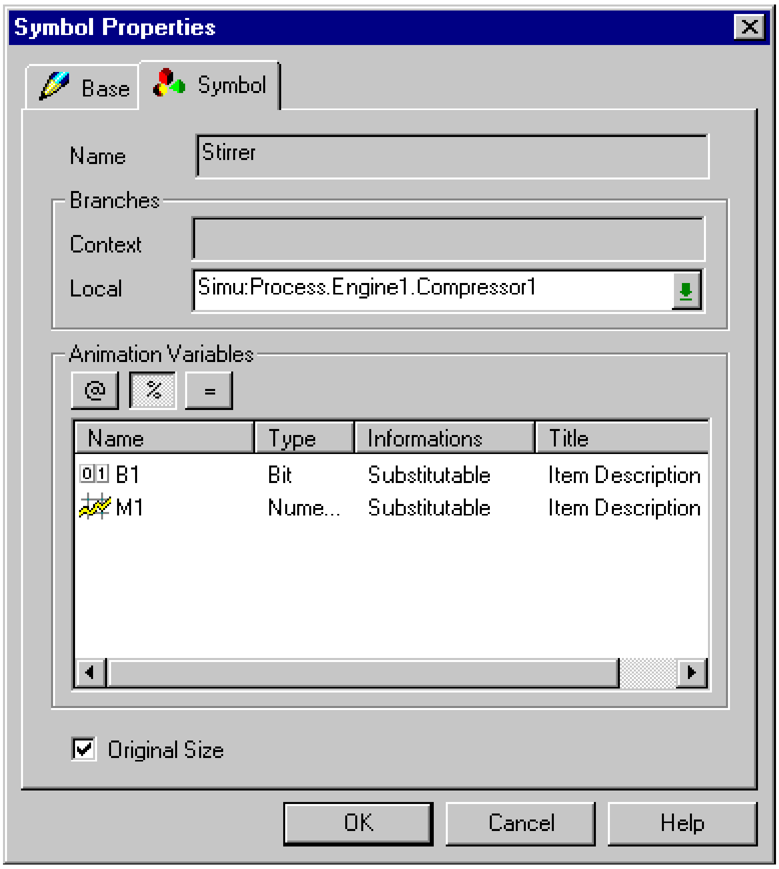
1. Using the mouse click on the symbol in the source mimic with the left mouse button. Hold down the mouse button and the Ctrl key on the keyboard. Drag the copy of the symbol to its new mimic - release the mouse button when it is in the correct position.
2. When the mouse button is released a dialog box will be displayed showing the names of any variables and allowing a branch to be entered. Enter a branch (if required) and select the OK button. The symbol has now been moved to the new mimic.

How to Paste a symbol Using Ctrl C and Ctrl V

1. Using the mouse select the symbol in the source mimic by clicking with the left mouse button. Use Ctrl C to copy the symbol.
2. Move the mouse pointer to destination mimic and click with the left button to select the mimic. Use Ctrl V to paste the symbol into the mimic.

Displaying a symbol's Properties

A symbol's properties may be displayed when in design mode by double clicking on it.



Name: Stirrer

Context:

Local: Simu:Process.Engine1.Compressor1

Name	Type	Informations	Title
B1	Bit	Substitutable	Item Description
M1	Nume...	Substitutable	Item Description

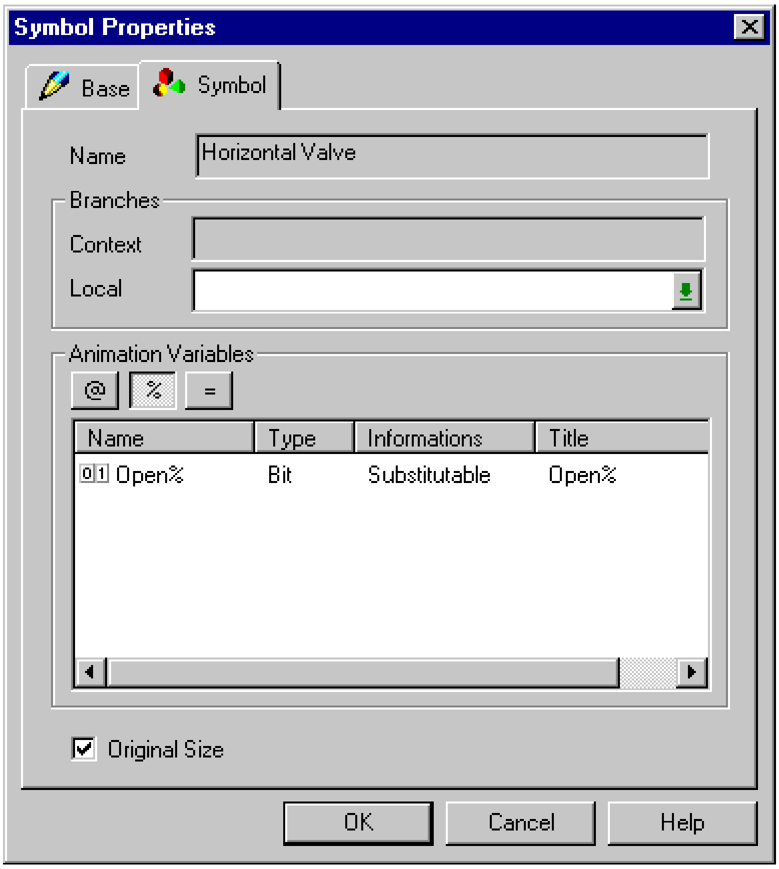
☑ Original Size

The properties box contains a list of the variables used in the symbol and its branch.

- *Original size* – If selected the symbol is displayed at 100% of its original size.

Using Symbols with Deferred Referencing

If the animations contained in a symbol use temporary variables, before the symbol will display anything meaningful the temporary variables must be substituted by variable from the server. This is done using the symbol properties dialog box that is displayed by double clicking on the symbol when in design mode.



Name: Horizontal Valve

Context:

Local:

Name	Type	Informations	Title
Open%	Bit	Substitutable	Open%

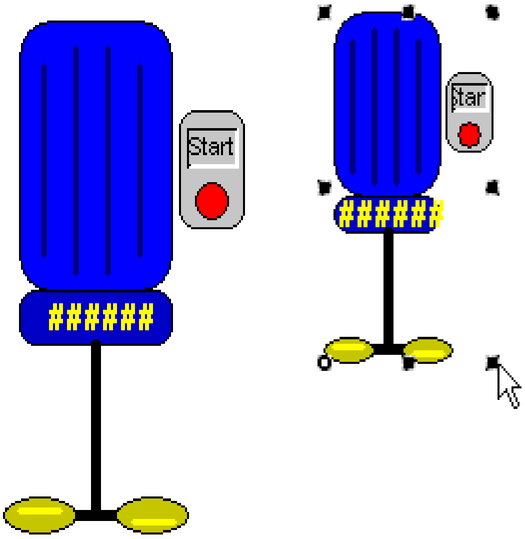
☑ Original Size

Any temporary variable names are indicated by displaying *Substitutable* in the information field. To change a temporary name to a real one double click on its entry in the variable list. The variable selector is then displayed from which a variable name may then be selected. In the example above the temporary variable `Open%` has been substituted by `Simu:@Process.Engine1.Compressor1.B1`.

Changing a reference to a temporary variable name only affects the copy of the symbol on which the action is performed. In this way the same symbol can be used to display an unlimited number of different variables.

Re-sizing a symbol

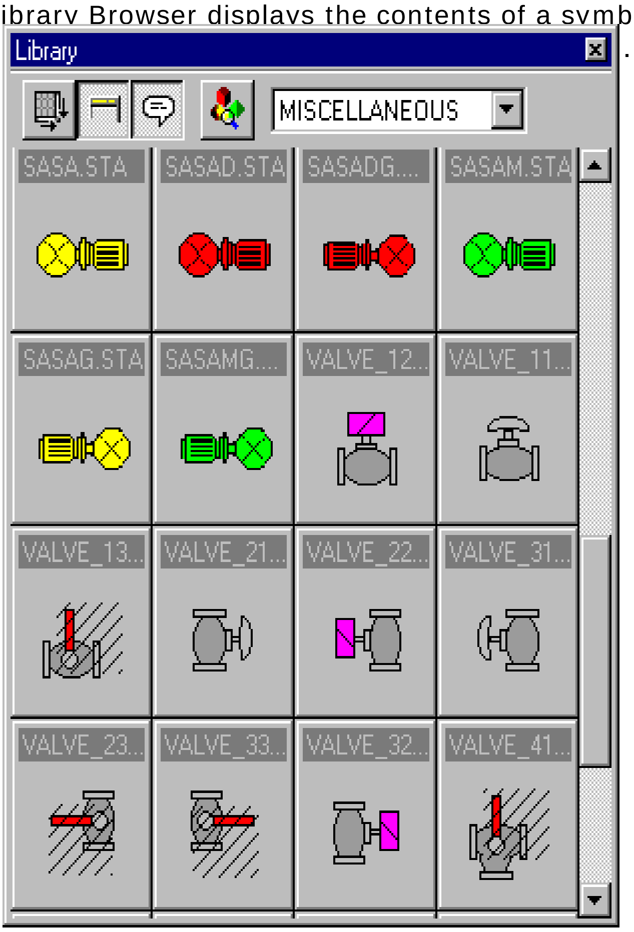
A symbol may be re-sized in the same way as a single drawing element by selecting it and clickir of the handles surrounding it.



The symbol may be re-sized in one or both dimensions depending on where the mouse is positioned. Holding down the Ctrl key whilst re-sizing will ensure that the X and Y dimensions retain the original ratios.

Using the Library Browser

The Library Browser displays the contents of a symbol library as thumbnails sorted into alpha . The library to be displayed is selected using



The Browser may be used to select and paste symbols into a mimic, and also to display a symbol's properties.

To select and paste a symbol into a mimic

1. Click on the symbol thumbnail in the library. Hold down the mouse button and drag the cursor to the location in the mimic.
2. Release the mouse button. A copy of the symbol will appear in the mimic.

To display a symbol's properties

1. Double click on the symbol thumbnail in the library. The symbol properties dialog box will be displayed.

What does it mean when a thumbnail is hatched?

When a thumbnail is hatched it means that the symbol contains one or more animations. A thumbnail that is not hatched does not contain any animations.

What does it mean when a thumbnail contains the ☒ symbol?

When a thumbnail contains this symbol there is problem with the symbol. In most cases this will be due to the symbol referring to another symbol or bitmap that does not exist. This can happen if you have copied symbols from one project to another without taking all the libraries.

Configuring Libraries

Creating a New Library | Enabling a Library

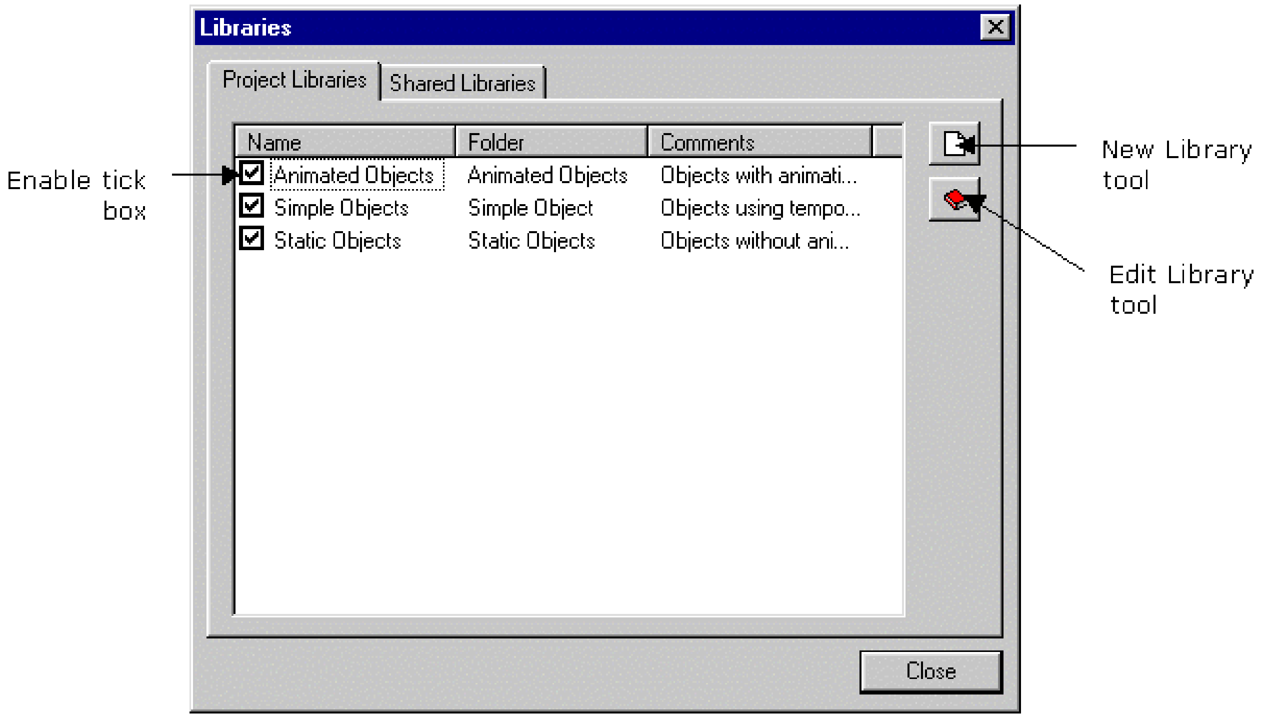
A library is the collective name for the folders used to store the window, window template, script, object and image files of which a project is composed. By default there is only one library, known as the Local Library, located in the project's root folder. In addition to the Local Library it is possible to create two other types of library.

- Project libraries. These are located in the Libraries folder under the project's root folder and are only available to that project.
- Shared libraries. These are located in the Shared Libraries folder under the HMI's root folder. The content of these libraries is available to all projects.

Using libraries has two main advantages.

- You can organise the components of a project in a logical way. For example you could create separate libraries of animated and non-animated objects.
- You can create libraries of windows, window templates, scripts, images and objects that can be reused in several projects.

There is no limit to the number of libraries that you can create, or the way in which they are used. Libraries are created and managed using the Tools.Project.Libraries **command**.

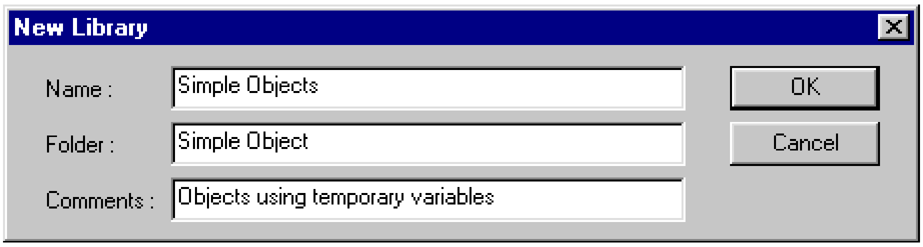


Name	Folder	Comments
☑ Animated Objects	Animated Objects	Objects with animati...
☑ Simple Objects	Simple Object	Objects using tempo...
☑ Static Objects	Static Objects	Objects without ani...

Creating a New Library

The following explanation applies to both project and shared libraries.

1. Open the library management tool using the Tools.Project.Libraries **command**.
2. Select either the Project Libraries or Shared Libraries tab as appropriate.
3. Click the New Library tool.



New Library

Name : Simple Objects

Folder : Simple Object

Comments : Objects using temporary variables

4. Enter the name for the library. This is the name by which it will be referenced in the HMI.
5. Enter the name of the folder. The folder name doesn't have to be the same as the library name but you may find it easier to manage if it is.
6. Enter a description of the library in the comment field.
7. Confirm the details by clicking the OK button.
8. Close the library management box.

Existing shared libraries are automatically registered when the Library Management box is opened. They do not have to be manually configured but they must be enabled if they are to be used in a project.

You cannot delete libraries online. To delete a library you must stop the HMI and manually delete its folders.

Enabling a Library

Before a library can be used in a project it must be enabled. Libraries that are not enabled do not appear in the list of libraries displayed when inserting a window, image or object. Libraries that you create online are automatically enabled when you create them. Existing shared libraries must be enabled before they can be used.

1. Open the library management box using the Tools.Project.Libraries **command**.
2. To enable a library tick the box immediately adjacent to it's name.
3. Close the library management box.

Library Converter

Using the Library Converter

The Library Converter is a tool for converting the mimics, mimic templates and symbols, stored in a library, between the binary and ASCII formats in which they can be stored. It does so in bulk in that it acts on the entire contents of one or more libraries that the User selects for conversion.

About the Libraries

A library is the collective name for the folders used to store the window, window template, script, symbol and image files of which a project is composed. By default there is only one library, known as the Local Library, located in the project's root folder. In addition to the Local Library it is possible to create two other types of library.

- Project libraries. These are located in the Libraries folder under the project's root folder and are only available to that project.
- Shared libraries. These are located in the Shared Libraries folder under the HMI's root folder. The content of these libraries is available to all projects.

The components of a library may be stored in either of two standard formats.

- Binary. The HMI's native format. In general the files are more compact and quicker to load than those using ASCII format.
- ASCII. Readable with any text editor. In general the files are much larger and slower to load than those using binary format.

For more information on Libraries see the topic Configuring Libraries.

In the Options for a project, you can specify the default formats for saving mimics and symbols, and also allow the developer to select the save format. (See the Options topic for how this is done.)

Using the Library Converter

The Library Converter is opened from the menu bar using the command Display.Library Converter.

The left-hand panel of the dialog box lists all libraries that are available to the current project. This includes the project's own libraries plus any available shared libraries. User options appear to the right, and the lower pane displays progress messages.

Before using the Library Converter all mimics must be closed.

How to convert one or more Library

1. Select the library for conversion by clicking on its name in the library list. To select more than one library hold down the Ctrl button and click on the name each library in turn.
2. Select which of the library components (Mimics, Symbols and Mimic Templates) are to be converted.
3. Select the target format from either Binary or ASCII.
4. Start the conversion process by clicking the *Convert* button.

The selected components are converted and saved under their original names. Whilst the conversion is in progress messages are generated in the Status pane at the bottom of the dialog box.

What happens if the Cancel button is used?

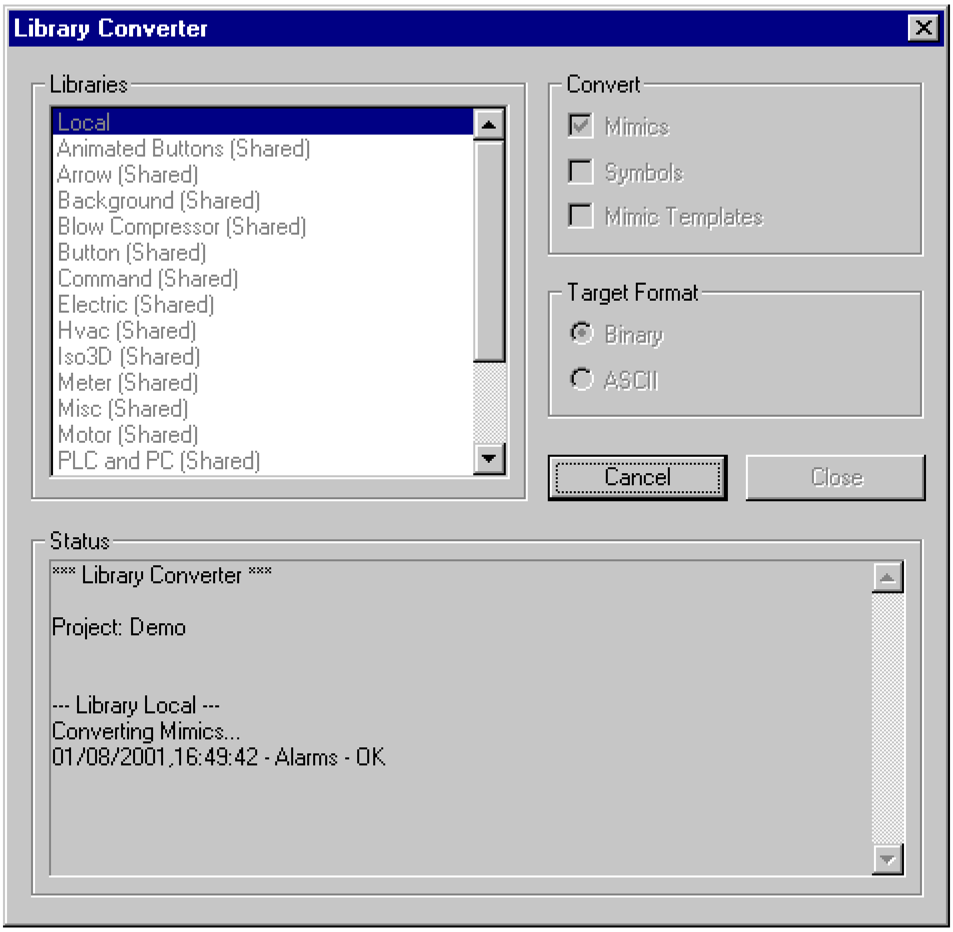
If the *Cancel* button is clicked at any time the conversion process stops. In this case you may be left with a library that is partially converted. As the converted components are saved under their original names the only way to know what has been converted is to examine the messages in the Status pane.

What happens if a bad file is encountered?

If a file is encountered that does not comply to either the recognised ASCII or binary formats it will be ignored. A message reporting the fact will appear in the Status area.

What happens to files containing ActiveX or Bean controls?

If you convert a file containing either an ActiveX control or Bean control to ASCII then, in addition to the original file, a file with the same name but with a .binary extension is created. For example if you convert a mimic named MyMimic to ASCII and it would produce a file named MyMimic plus another named MyMimic.binary.



```
*** Library Converter ***

Project: Demo


--- Library Local ---
Converting Mimics...
01/08/2001,16:49:42 - Alarms - OK
```

Security System Overview

The security system is used to manage users and their access rights. The features available from the HMI and the project at any time will depend on the access rights of the current user.

The HMI security system is based on a user and profile system.

- A profile contains a list of access rights selected from a number of categories.
- A user is given a name and password and allocated one or more profiles.

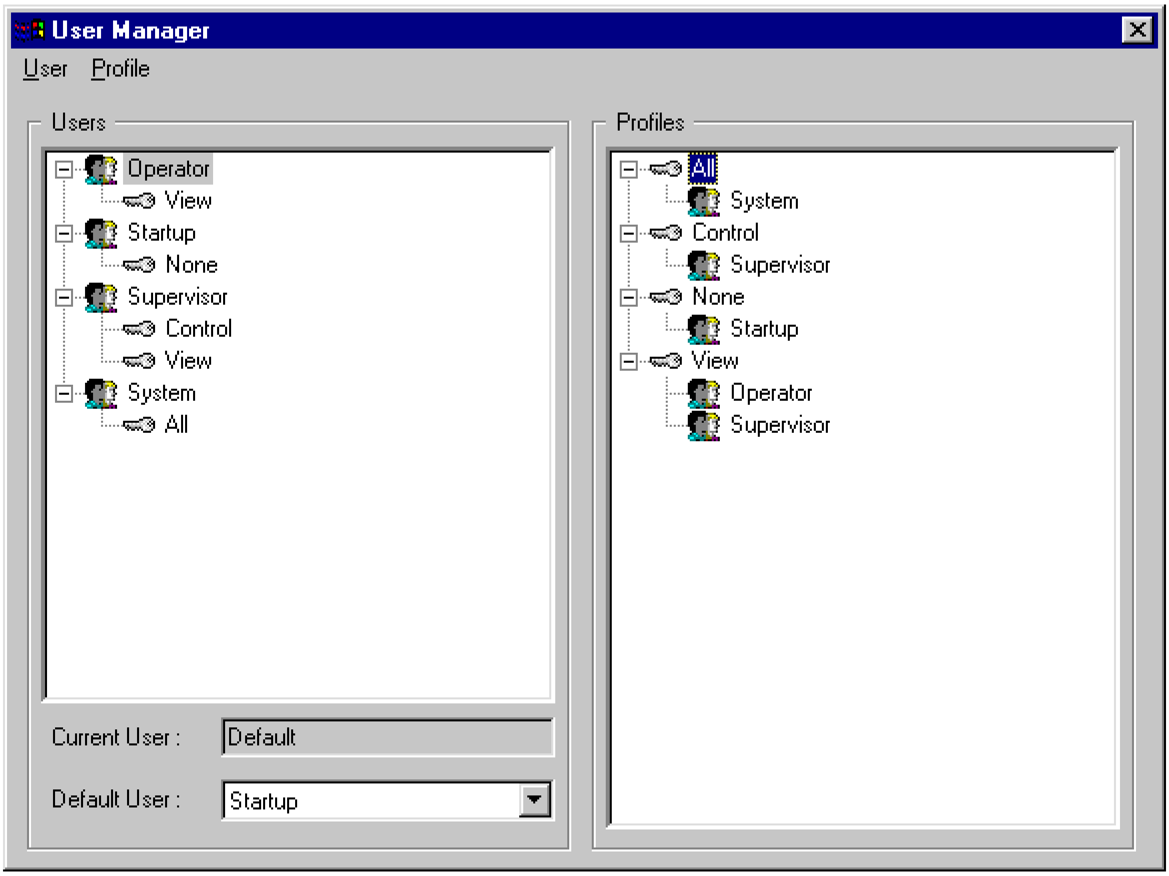
A user logs on to the system by entering his/her name and password in a special dialog box. Only one user may be logged on at any time.

As part of the security configuration you can create a default user which supplies the access rights when no other user is logged on.

The User Manager

How to Add and Configure a Profile | How to Add and Configure a User

The User Manager is used to add and configure users and profiles. It is displayed using the Tools.Security.Configure command.



- Operator
  - View
- Startup
  - None
- Supervisor
  - Control
  - View
- System
  - All

- All
  - System
- Control
  - Supervisor
- None
  - Startup
- View
  - Operator
  - Supervisor

Current User : Default

Default User : Startup

The User Manager contains two main areas.

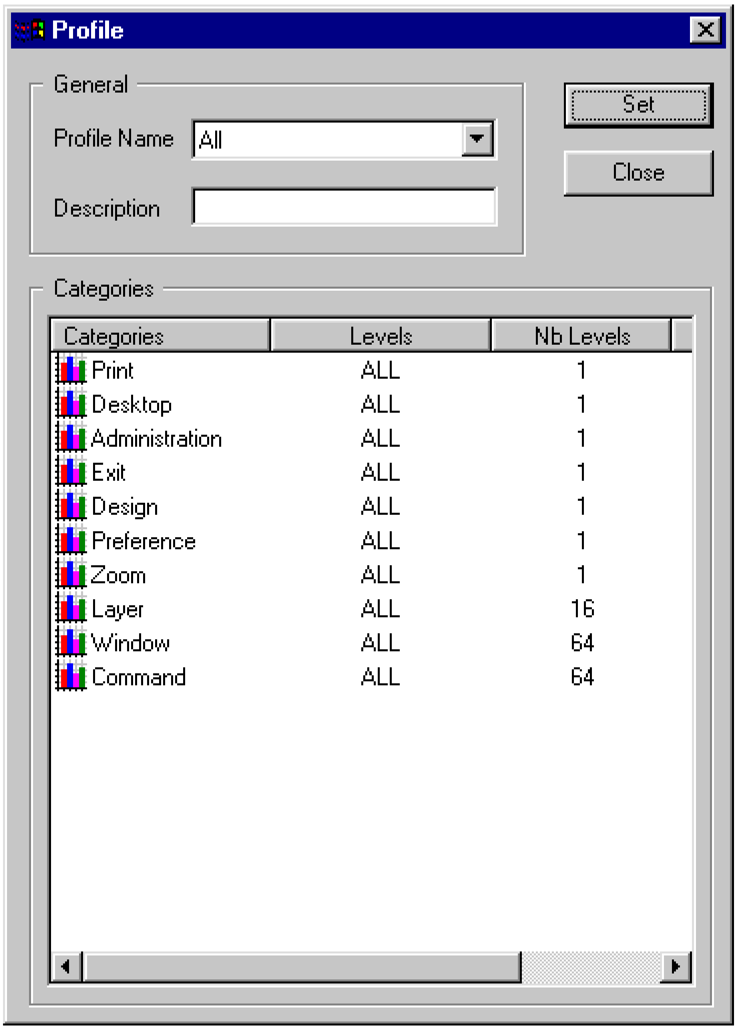
- The left pane, which contains a tree control showing the users and allocated profiles. Right clicking in this area will display a pop-up menu that may be used to add, copy, modify and delete users.
- The right pane, which contains a tree control showing the profiles and the users to which they have been allocated. Right clicking in this area will display a pop-up menu that may be used to add, copy, modify and delete profiles.

At the bottom is a list box from where you can select the default user.

How to Add and Configure a Profile

1. Open the User Manager using the Tools.Security.Configure command.

2. Right click in the right pane of the User Manager and from the pop-up menu select the New profile command.
3. Enter a name for the new profile. This creates a new un-configured profile.
4. Right click on the new profile and from the pop-up menu select the Edit profile command. A box is displayed containing a list of user right categories. If you are configuring a new profile then none of the categories will have any rights selected.

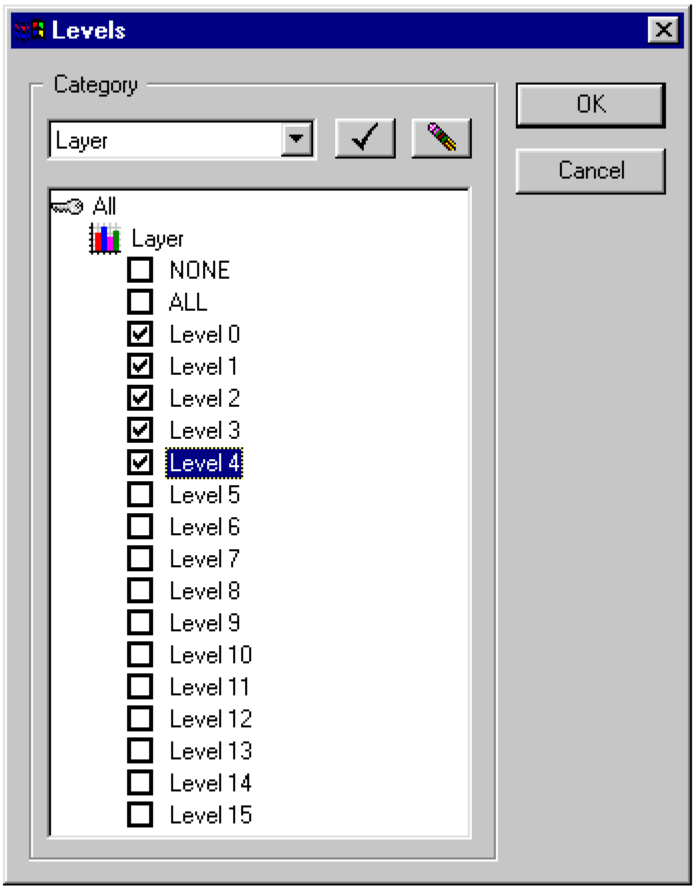


Profile Name: All

Description:

Categories	Levels	Nb Levels
Print	ALL	1
Desktop	ALL	1
Administration	ALL	1
Exit	ALL	1
Design	ALL	1
Preference	ALL	1
Zoom	ALL	1
Layer	ALL	16
Window	ALL	64
Command	ALL	64

5. To change a category double click on its name in the list. The individual rights may then be enabled and disabled by selecting the corresponding tick box. You can select other categories using category the list box. All and None options are provided to quickly select all or none of the rights.



Category: Layer

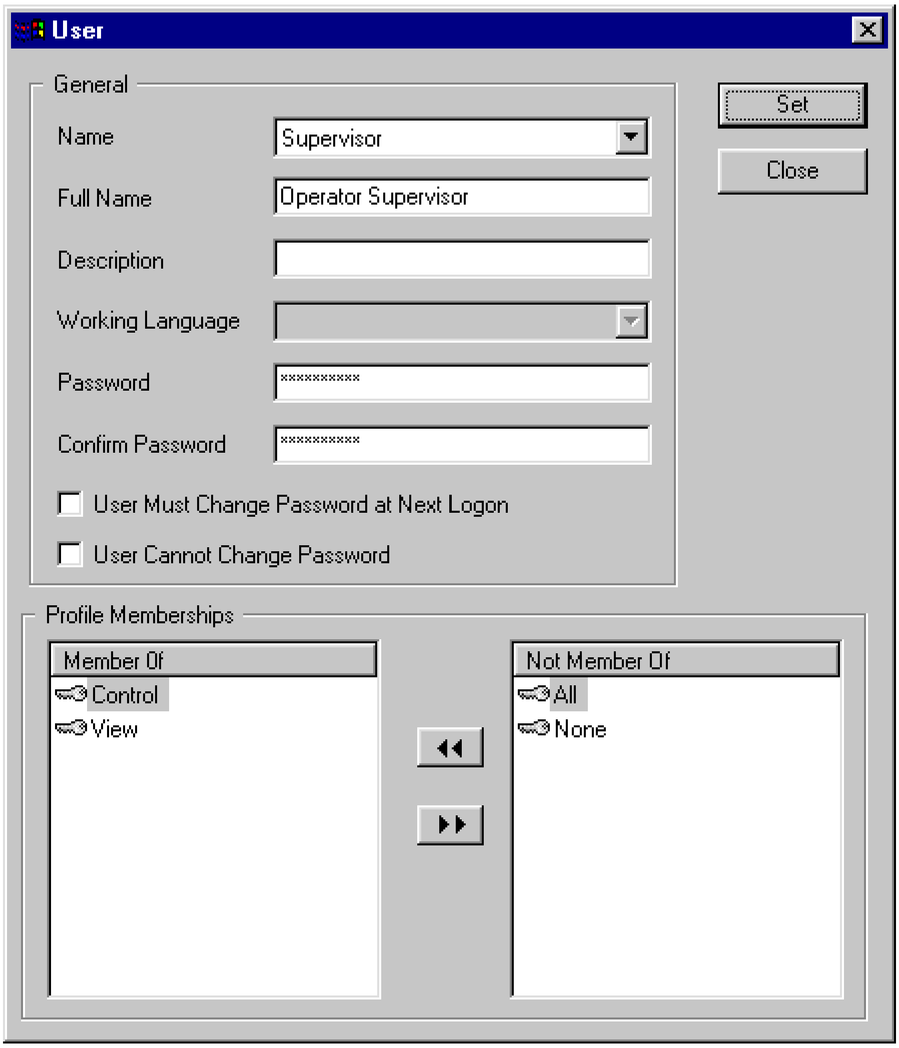
All
- Layer
  - □ NONE
  - □ ALL
  - ☑ Level 0
  - ☑ Level 1
  - ☑ Level 2
  - ☑ Level 3
  - ☑ Level 4
  - □ Level 5
  - □ Level 6
  - □ Level 7
  - □ Level 8
  - □ Level 9
  - □ Level 10
  - □ Level 11
  - □ Level 12
  - □ Level 13
  - □ Level 14
  - □ Level 15

6. Confirm the changes by selecting the OK button in the category box and the close button in the profile box.

How to Add and Configure a User

1. Open the User Manager using the Tools.Security.Configure command.

2. Right click in the left pane of the User Manager and from the pop-up menu select the New user command.
3. Enter a name for the new user. This creates a new un-configured user. User names are not case sensitive.
4. Right click on the new user and from the pop-up menu select the Edit user command. A box is displayed containing the user properties.



Name: Supervisor

Full Name: Operator Supervisor

Description:

Working Language:

Password: **********

Confirm Password: **********

☐ User Must Change Password at Next Logon

☐ User Cannot Change Password

Member Of	Not Member Of
Control	All
View	None

5. Enter a password for the user in the Password field and confirm the password by entering it again in the Confirm Password field. Click the Set button to set the password. Passwords can contain any alphanumeric character plus the underscore. Passwords are case sensitive.
6. Allocate a profile by selecting it in the list of Not Member Of and selecting the << button. You can add more than one profile. The rights are combined using a logical OR.
7. Close the User Management box using the Close button when finished.

You can also add profiles to users by dragging and dropping them in the main User Management configuration box..

Other options when configuring a User

You can select two other options when configuring a user.

Selecting the option Must change password at next log on, will force a user to change the password the first log on after this option has been selected.

Selecting the option User cannot change password will disable the option for a user to change his/her own password.

Security Categories

Adding Security Categories

Category	Options	Description
Administration	NONE or ALL	When set to ALL, the user has access to the following menu commands. - Tools.Security.Configure... - Tools.Servers... - Tools.Project.Open... - Tools.Project.Options... When set to ALL, the "Administration" security category allows users to perform all operations they want to do (and this, even if other security categories are set to NONE).
Preferences	NONE or ALL	When set to ALL, the user has access to the following menu commands. - Tools.Preferences.Colors... - Tools.Preferences.Blinking... - Tools.Preferences.ActiveX Controls... - Tools.Preferences.Decluttering...
Design	NONE or ALL	When set to ALL, the user can modify the current project. That is, add / modify / delete windows, graphical elements, animations and scripts.
Desktop*	Only two: NONE or ALL	When set to ALL, the user has access to the underlying operating system keys (example: Ctrl-Alt-Del).
Print*	Only two: NONE or ALL	When set to ALL, the user has access to the print tools.
Exit*	Only two: NONE or ALL	When set to ALL, the user may shut down the HMI.
Zoom	Only two: NONE or ALL	When set to ALL, the user may zoom windows.
Layer	0 to 15	At runtime, a drawing layer is displayed only if the user has the corresponding layer level.
Window	0 to 63	At runtime, a window may be opened only if the user has the corresponding window access level.
Command	0 to 63	At runtime, the value of a variable may be forced (from a control zone animation) only if the user has the corresponding command level.

* Not implemented at time of writing.

Adding Security Categories

The ability for the user to add additional security categories is provided to allow security checking for 3rd party embedded ActiveX controls. Additional security categories are added using a special text file which can be located in either the program directory of the HMI or the Config Files directory of the project. The text file can be given any name as long as it has a .SEC extension.

The format of the files is similar to the .INI files used by Windows itself. A section named [Categories] defines new security categories with the associated number of levels. Each new category then has a section of its own which is used to provide the name of the category to the user in one or more languages. This information is in the form of :

Language ID=Label

Where the Language ID is that used by Windows. The following is an example of this file.

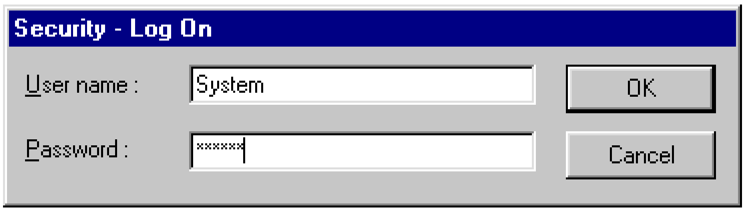
```
[Categories]
AlarmViewer=1
AlarmAck=32

[Category\AlarmViewer]
0009-English=AlarmViewer

[Category\AlarmAck]
0000-Neutral=Alarm Acknowledge
0009-English=Alarm Acknowledge
040C-French Standard=Acquittement d'alarmes
0C0C-French Canadian=Acquittement d'alarmes
```

For further information on using this file please contact technical support.

Logging On and Logging Off

Default User Rights

To log on you enter a user's name and password in the Log On dialog box. As standard this is displayed using the Tools.Security.Log On command. It may also be available from one of the keyboard function keys depending on the project configuration (see the topic on keyboard Layout). When you log on the current user (if any) is automatically logged off, it is not necessary for the current user to log off as a separate operation.



User name : System

Password : ******

You can, if necessary, log off manually using the Tools.Security.Log Off command.

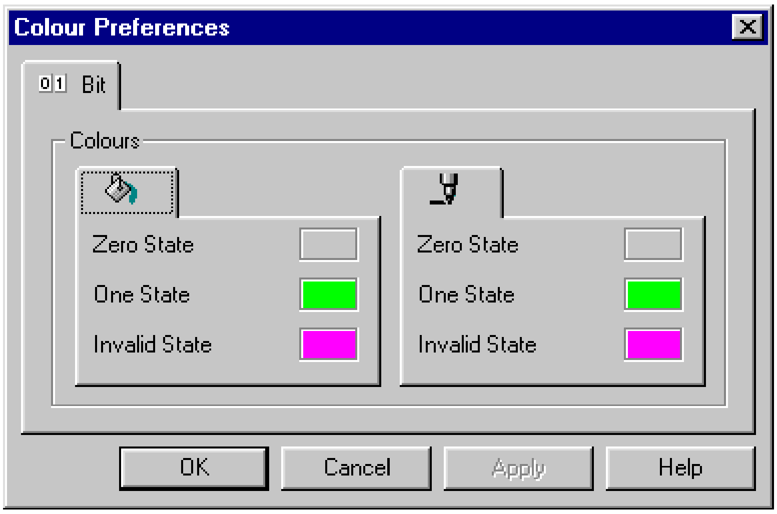
Default User Rights

When the current user logs off the access rights revert to those specified by the default user (as configured by the User Manager).

If there is no specified default user and there is one or more users configured then, at start-up or when a user logs off, the Log On box is displayed and all rights are disabled until another user logs on.

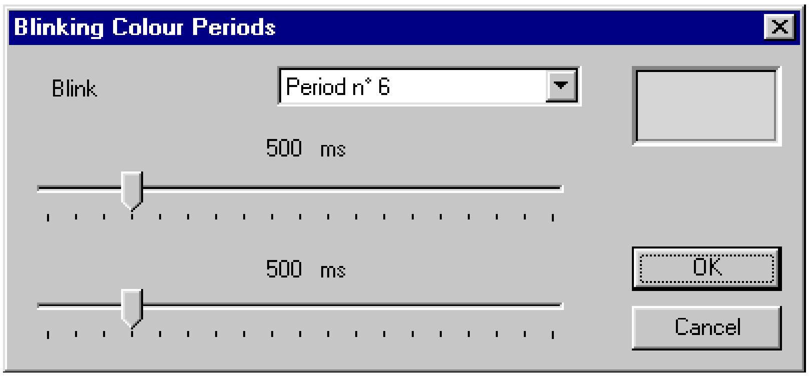
Selecting Colour Defaults

The default colours that are used when the colour bit animation is used may be changed using the Tools.Preferences.Colours command.



Selecting Blinking Colour Periods

In many of the drawing elements and animations you can select a blinking colour for one or more of the properties. A blinking colour alternates between two selected colours at pre-defined rate. You can change the period for which each colour is displayed using the Tools.Preferences.Blinking Rates command.



There are configurable 16 blink rates each of which has two configurable periods, one for each of the colours. To change either period click and drag the appropriate slider control. If you are using blinking colours to attract the operators attention it is recommended that you use a ratio of approximately 2 to 1 for the periods. Each period may be configured within the range of 150 to 2000 ms.

The Keyboard Manager

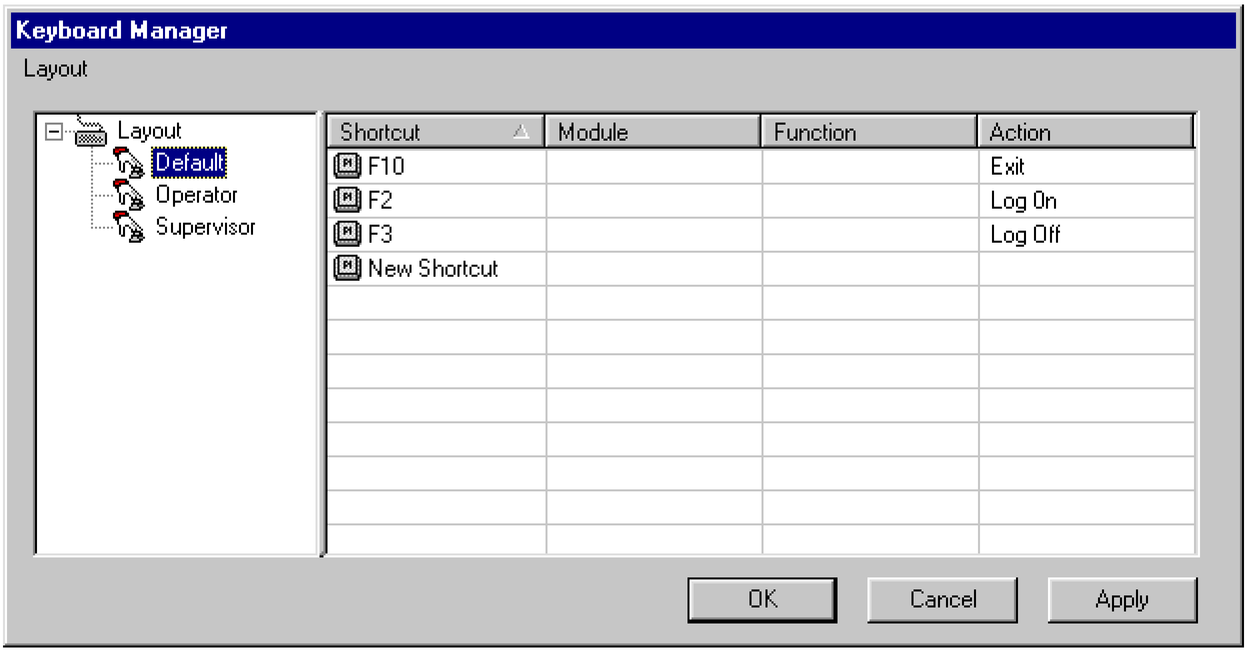
How to Create a New Keyboard Layout | How to Add a Shortcut for a Standard Action | How to Add a Shortcut to run a Function

The Keyboard Manager is used to create user-defined shortcut keys. A shortcut is the association of a keystroke with a standard HMI action or a function written using the scripting language. Each set of shortcut keys is allocated to a named keyboard layout. You can configure several layouts although only one may be active at any one time. At run-time a layout is activated using the scripting language – in this way you can associate a layout to a project, a window or a particular user.

A keystroke is a key or key combination from the keyboard. You can use the function keys (F1 to F12, Esc, Tab etc.) and any alpha-numeric key combined with Ctrl, Alt and Shift.

Once the keyboard layout is activated the shortcut that you allocate to a key or key combination replaces any standard functionality it may have had.

The Keyboard Manager is displayed from the menu using the Tools.Keyboard Manager command.



- Layout
  - Default
  - Operator
  - Supervisor

Shortcut	Module	Function	Action
F10			Exit
F2			Log On
F3			Log Off
New Shortcut			

The Keyboard manager contains two areas.

- The left hand pane with a tree control from where layouts may be created, copied and deleted.
- The right hand pane where individual layouts are configured.

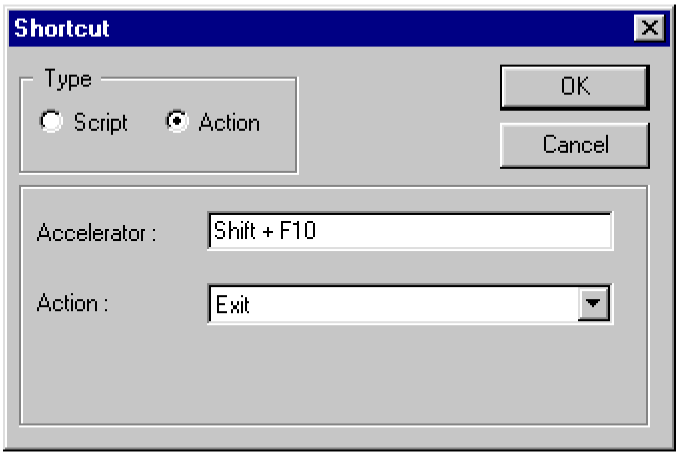
How to Create a New Keyboard Layout

1. Display the Keyboard Manager using the Tools.Keyboard Manager command.
2. Right click in the left hand pane and select New Layout from the pop-up menu.
3. Enter the name for the layout and click the Apply button.

How to Add a Shortcut for a Standard Action

1. Select the layout by clicking on it in the left hand pane.

2. Double click on the New Shortcut entry in the right hand pane.
3. Select the Action option button.



Shortcut

Type

○ Script ● Action

Accelerator : Shift + F10

Action : Exit

4. Click in the Accelerator field and press the key combination for this shortcut.
5. Select the action from the Action list box.
6. Confirm the configuration by clicking the OK button.

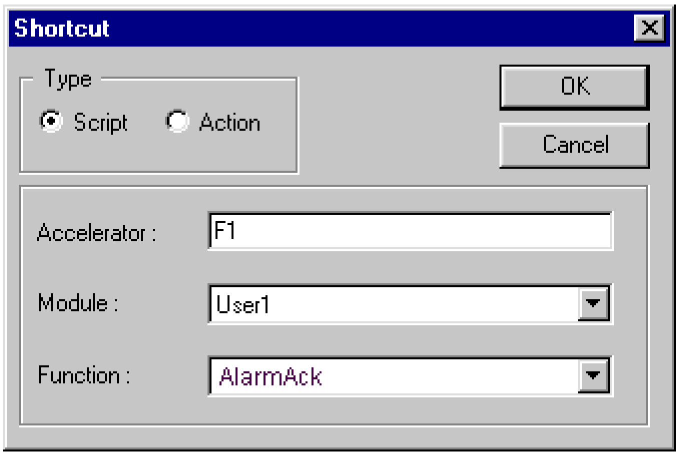
The following standard actions are available.

Action	Operation
Exit	Shutdown the HMI.
LogOn	Displays the “Log On” window.
LogOff	Logs off the current operator.
Previous Window	Activates the previous window.
Next Window	Activates the next window
Previous Zone	Activates the previous zone within the current window.
Next Zone	Activates the next zone within the current window.
First Zone	Activates the first zone declared in the current window.
Last Zone	Activates the last zone declared in the current window.

How to Add a Shortcut to run a Function

1. Select the layout by clicking on it in the left hand pane.

2. Double click on the New Shortcut entry in the right hand pane.
3. Select the Script option button.



Type: ● Script ○ Action

Accelerator : F1

Module : User1

Function : AlarmAck

4. Click in the Accelerator field and press the key combination for this shortcut.
5. Select the module in which the function is to be found using the Module list box.
6. Select the function using the Function list box.
7. Confirm the configuration by clicking the OK button.

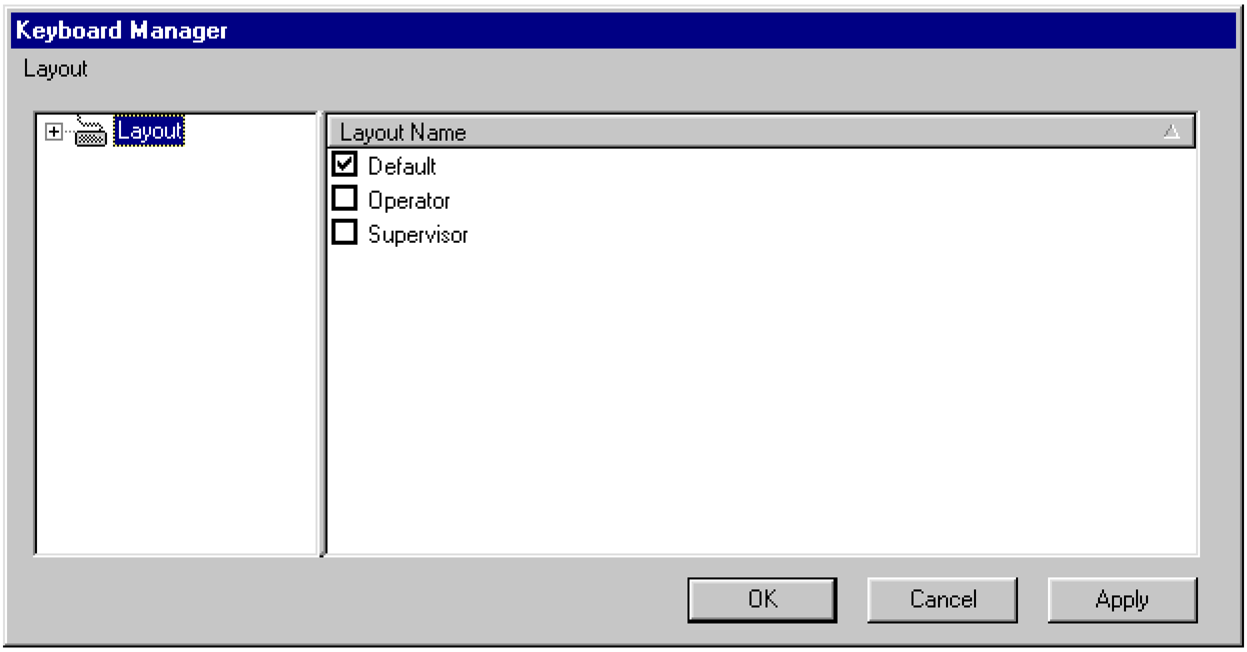
You can run either a function or subroutine. It is not possible to pass parameters.

Activating a Keyboard Layout

Activating a keyboard layout using a script

A keyboard layout is activated either by selecting the layout from the Keyboard Manager or by using a script.

To activate a layout from the keyboard manager you collapse the layout tree in the left hand pane and then click on the layout to be activated in the right hand pane. This activation is only temporary and will be forgotten when the HMI is restarted.



Activating a Layout Using a Script

To activate or de-activate a layout using a script you use the ActiveKeyboardLayout property of the ThisProject object.

```
Private Sub OperatorButton_Click()
  ThisProject.ActiveKeyboardLayout = "Default" ' Activate operator layout
End Sub

Private Sub NoneButton_Click()
  ThisProject.ActiveKeyboardLayout = "" ' Deactivate  layout
End Sub
```

! A runtime error will be generated if the name passed to this property can't be found in the layout list defined for the project

What are Expressions?

An expression allows the calculation of a value by using one or more variables, functions, operators and constants. An expression is automatically evaluated each time any of the variables within it change.

Expressions may be defined and used in two ways.

- They can be entered directly in any animation in the field normally used for a variable name. Expressions used in this way are only evaluated whilst the window containing the animation is open. An expression is differentiated from a variable name by preceding it with an equal (=) sign.
- They can be entered in the Expression Editor where they are known as an expression model. Each model is identified by a name that is then be used instead of a variable name. This is useful where the same expression is used a number of times. The animation uses the result of the expression (it does not take a copy of the expression and evaluate it) and hence if the expression is changed this is automatically reflected in any animation in which it is used.

Expression Syntax

Functions | Operators | Expression Examples

An expression contains the name of at least one or more database variables, operators, functions and numeric constants.

The following rules must be observed when entering an expression.

- Expressions are limited to 20 symbols. (A symbol is a database variable, a constant, an operator or a function).
- A space, a bracket or an equal sign must always precede a function.
- When an expression is entered directly in an animation it must be preceded by an equals sign. This tells the HMI to interpret the field as an expression and not as a reference to a variable.
- When using an expression model in an animation it must be immediately preceded with an equal sign (no space).

Functions

Function	Operation	Syntax
SIN	Sine (degrees)	SIN(val)
COS	Cosine (degrees)	COS(val)
TAN	Tangent (degrees)	TAN(val)
ASIN	Arc sine	ASIN(val)
ACOS	Arc cosine	ACOS(val)
ATAN	Arc tangent	ATAN(val)
LOG	Natural log	LOG(val)
EXP	Exponential	EXP(val)
ABS	Absolute value	ABS(val)
FLOOR	Round down	FLOOR(val)
CEIL	Round up	CEIL(val)

Operators

Operator	Operation	Syntax
+	Addition	Var1 + Var2
-	Subtraction	Var1 - Var2
*	Multiplication	Var1 * Var2
/	Division	Var1 / Var2
==	Equality	Var1 == Var2
!=	Inequality	Var1 != Var2
>	Greater than	Var1 > Var2
>=	Greater or equal	Var1 >= Var2
<	Less than	Var1 < Var2
<=	Less or equal	Var1 <= Var2
! or NOT	Logical NOT	! Var1 or NOT Var1
|| or OR	Logical OR	Var1 || Var2 or Var1 OR Var2
&& or AND	Logical AND	Var1 && Var2 or Var1 AND Var2
~ or BNOT	Binary NOT	~ Var1 or BNOT Var1
| or BOR	Binary OR	Var1 | Var2 or Var1 BOR Var2
& or BAND	Binary AND	Var1 & Var2 or Var1 BAND Var2
POW	Power	Var1 POW Var2
MOD	Modulus	Var1 MOD Var2
>> N or RSHIFT N	Right shift (N bits)	Var1 >> N or Var1 RSHIFT N
<< N or LSHIFT N	Left shift (N bits)	Var1 >> N or Var1 LSHIFT N

Expression Examples

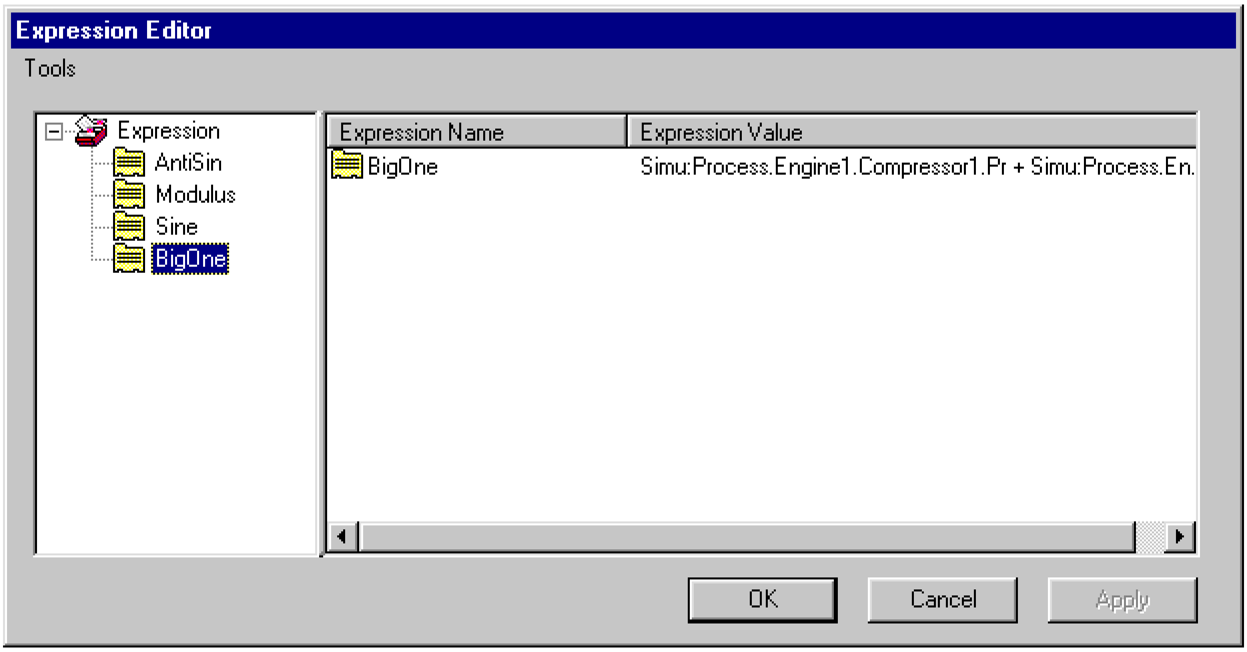
```
=CL:Motor1.Stop & CL:Motor1.Start | CL:Motor1.Fault
=(CL:Inlet.Flow1 + CL:Inlet.Flow2)/100    Note the use of brackets
=CL:Tank1.Temperature & 65520             Masks the first 4 bits of the value
=EXP(LOG(CL:Tank.Pressure)/2)             Takes the square root
```

The Expression Editor

How to Create an Expression Mode

When the same expression is required in several places, it can be useful to centralise its definition using the Expression Editor to create an expression model. Once you have created an expression model it is used in an animation by inserting its name instead of that of a variable. The syntax of expression models is the same as that used when you enter an expression directly in an animation.

The Expression Editor is displayed from the Tools.Expression Editor command on the menu.



Expression Name	Expression Value
BigOne	Simu:Process.Engine1.Compressor1.Pr + Simu:Process.En.

The Expression Editor contains two main areas.

- A tree control area that is used to browse the expression models defined in the project. A right click in this area displays a pop-up menu that allows you to create, copy, paste and delete expression models.
- A configuration area that shows the definition of the selected element in the tree control area. If the root icon is selected all expression models and their related definitions are displayed. Double clicking on an expression model displays a box in which it may be viewed and changed.

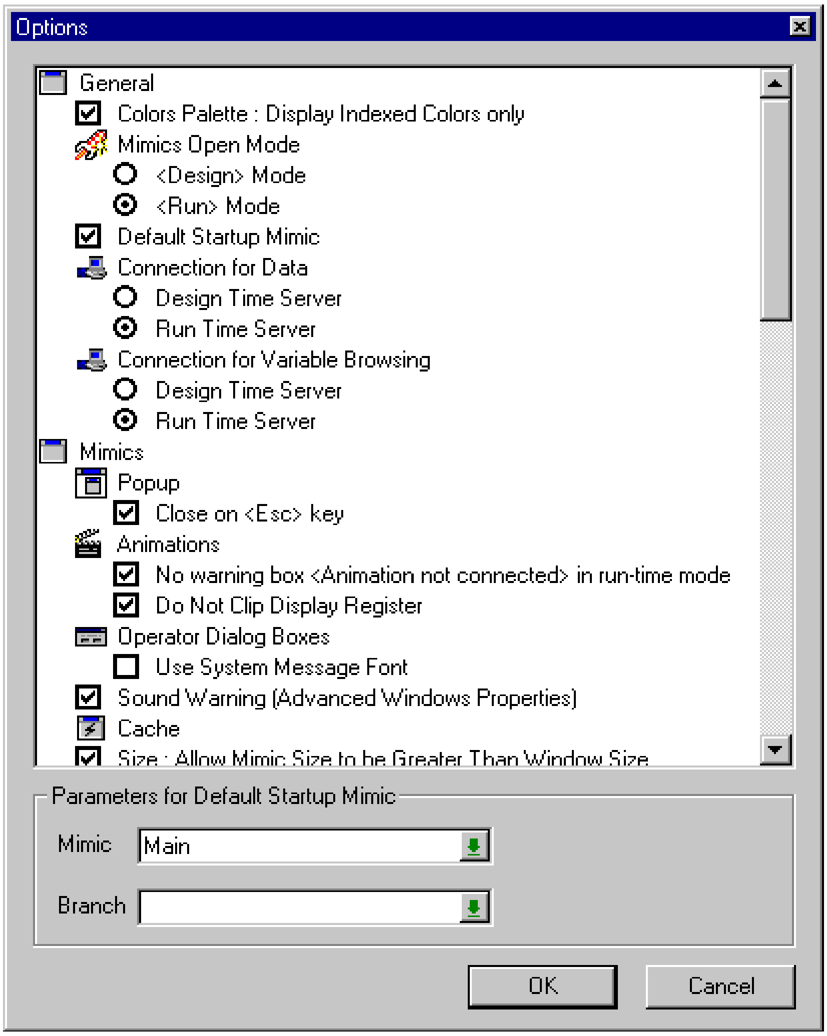
How to Create an Expression Model

1. Display the Expression Editor.
2. Right click in the tree control area. Select the new command from the pop-up menu.
3. Enter the name by which the expression model will be known.
4. Double click on the model name in the right hand pane. A box is displayed into which you enter the expression. The expression may be continued on several lines. The browser button may be used to select variable names.

The Options Tree

General | Mimics | File Format | Regions | Variable naming

The Tools.Options command displays a number of properties that affect the general operation of the HMI, many of which are not changed in most applications.



Options

- General
  - ☑ Colors Palette : Display Indexed Colors only
  - Mimics Open Mode
    - ○ <Design> Mode
    - ● <Run> Mode
  - ☑ Default Startup Mimic
  - Connection for Data
    - ○ Design Time Server
    - ● Run Time Server
  - Connection for Variable Browsing
    - ○ Design Time Server
    - ● Run Time Server
- Mimics
  - Popup
    - ☑ Close on <Esc> key
  - Animations
    - ☑ No warning box <Animation not connected> in run-time mode
    - ☑ Do Not Clip Display Register
  - Operator Dialog Boxes
    - ☐ Use System Message Font
  - ☑ Sound Warning (Advanced Windows Properties)
  - Cache
  - ☑ Size : Allow Mimic Size to be Greater Than Window Size

Parameters for Default Startup Mimic

Mimic: Main

Branch:

General

Colour Palette : Display Indexed Colours only - Disables display on the first 32 (non-indexed) colours in the palette.

- **Mimic Open Mode**

  <Design> Mode - Mimics are opened in design mode.
  <Run> Mode - Mimics are open in run-time mode. This is the default selection.

Default Startup Mimic - Enables a mimic to be opened automatically when the HMI starts. The name of the mimic and an optional branch are entered in the parameters area.

Connection for Data
Design Time Server - Selects the Design Time server for the collection of real time information from an OPC server.
Run-time Server - Selects the Run-time server for the collection of real time information from an OPC server. This is the default selection.

Connection for Variable Browsing

Design Time Server - Selects the Design Time server for variables browsing in an OPC server.

Run-time Server - Selects the Run-time server for variable browsing in an OPC server. This is the default selection.

Mimics

- **Popup**

  Close on <Esc> key – Enables popup mimics to be closed using the Esc key on the keyboard

- **Animation**

  No warning box <Animation not connected> in run-time mode – Disables the display of a run-time warning for incorrectly configured animations.

- **Operator dialog boxes**

  Use System Message font – Enables use of the Windows' system font for text in User dialog boxes.

Sound Warning (Advanced Windows Properties) – Enables the generation of a sound using the PC's loudspeaker when the Beep option in a window's properties is selected. The frequency and duration of the sound may be entered in the parameters area.

Cache - Enable mimic caching. Mimics that have been previously opened are temporarily stored in memory when they are closed. The next time they are opened they are retrieved from memory rather than hard disk making the initial display much faster. The maximum number of mimics that may be cached at any time is entered in the parameters area. Note that caching too many mimics may have a negative effect on overall system performance because of the resources required.

Size : Allow Mimic Size to be Greater Than Window Size - Allow mimics to be larger than their containing window. If the mimic is larger than its window, scroll bars are automatically displayed to aid navigation.

Magnifying Glass Tool : Open Zoomed Mimic in

Same Window - The zoomed mimic is displayed in the original window.

Current Region - The zoomed mimic is displayed in a new window opened in the same region.

Previous Region - The zoomed mimic is displayed in a new window opened in the previous region.

Next Region - The zoomed mimic is displayed in a new window opened in the previous region.

Region 1 - The zoomed mimic is displayed in a new window opened in region 1.

Region 2 - The zoomed mimic is displayed in a new window opened in region 2.

Region 3 - The zoomed mimic is displayed in a new window opened in region 3.

Region 4 - The zoomed mimic is displayed in a new window opened in region 4.

Screen resolution

Adapt Mimic Size to Screen Resolution - See the topic on The Window Properties Box.

File Format

Ascii Format reading with check – Checks the integrity of the file format for windows and symbols each time they are opened.

Mimics

User can select the format – Allow the user to select the format in which a mimic is saved.

- **Default Format**

  Binary Format – Use the binary format as default when saving a mimic.

  Ascii Format – Use the ASCII format as the default when saving a mimic.

Original Format – Use the current format as default when saving a mimic.

- **Symbols**

  User can select the format – Allow the user to select the format in which a symbol is saved.

- **Default Format**

  Binary Format – Use the binary format as default when saving a symbol.
  Ascii Format – Use the ASCII format as the default when saving a symbol.
  Original Format – Use the current format as default when saving a symbol.

Regions

Advanced Mode – Enable advanced multiple region operation with the following options.
Multi open capability of the same mimic – Allow a copy of a mimic to be opened concurrently in each region.

Previous mimic (#P)
Global to the project – The previous mimic substitution is global to the entire project, independent of any regions.
Local to each region – The previous mimic substitution is local to the region.

- **Link Open <Close Caller> Mode**

  Close the Caller – The calling mimic is closed (if enabled) when invoking a Link Open Mimic animation at run time.
  Close all Region Mimics – All open mimics in the calling mimic region are closed when invoking a Link Open Mimic animation at run time.

- **Open in other Region that the caller one**

  Close the Caller – The calling mimicis closed (if enabled) when using a Link Open Mimic animation.
  Close all target Region Mimics - All open mimics in the target region (the one in which the mimic is being opened) are closed when invoking a Link Open Mimic animation at run time.

Variable naming

Enable @ as Absolute Character – Enable use of the @ character in variable names to force a direct reference when using a window or object with a branch.

The VBA Environment

Accessing the VBA Environment | Getting Help on VBA

The HMI supports two alternative programming environments, Cypress Enable and Microsoft VBA. This document describes the Microsoft VBA environment.

Accessing the VBA Environment

The VBA development environment may be accessed in a number of different ways.

- From the keyboard using Alt F11.
- Using the shortcut from the pop-up menu displayed by right clicking on any drawing element.
- Using the shortcut from the Events table in the Properties List.

The shortcuts can display either the VBA or Enable environments. To direct the shortcuts to VBA you must select either the *Show VBA* or *Show both VBA and Enable, shortcuts to VBA* option in the options tree. (Menu.Tools.Options)

Getting Help on VBA

The standard (not specific to the HMI) VBA help is displayed using function key F1 when in the VBA environment. The help is very comprehensive and split into a number of topics.

- Visual Basic User Interface Help

  Look here for Help on interface elements of the Visual Basic Editor, such as commands, dialog boxes, windows, and toolbars.

- Visual Basic Conceptual Topics

  The Conceptual Help topics include information to help you understand Visual Basic programming.

- Visual Basic How-To Topics

  Look in the *How To* section of Help to find useful common procedures, for example, how to use the Object Browser or how to set Visual Basic Environment options.

- Visual Basic Language Reference

  The Language Reference is the place to find Help on Visual Basic the language: all its methods, properties, statements, functions, operators, and objects.

- Visual Basic Add-In Model

  If you want to customise the Visual Basic editor, see this language reference for Help on the object model that allows you to extend the environment.

- Microsoft Forms Reference

  Look here for Help on Userforms and controls, and how to program with them using Visual Basic.

VBA help topics specific to the HMI, for example the Native HMI Objects, is found in the help displayed using function Key F1 when using in the HMI's development environment.

How the VBA Project Explorer Interacts with the HMI

In the VBA Project Explorer there are Projects, Project Items and Controls, there is no other possible depth in the VBA Project Explorer. Because of this limitation, a single HMI application is seen in the VBA Project Explorer as several discreet projects.

The Main Project

The Main project always exists and provides access to two project items ThisProject and ThisSystem. The Main project is automatically added as a reference to all other projects (seen as Reference to WorkSpace in the project explorer) so you can access its functions and variables as if they were global.

The MimicProject Project

Each opened mimic appears in a separate MimicProject project. In a MimicProject there is one ThisMimic project item representing the Mimic object. Each MimicProject has its own design and run modes. When switching modes, the mimic and associated MimicProject follow one another. When in design mode events are not executed.

The SymbolProject Project

Each Symbol displayed in an opened mimic will appear in a separate SymbolProject project. Only one SymbolProject is displayed for each Symbol type even if there are several instances of it on a mimic. In a SymbolProject there is one ThisSymbol project item representing the Symbol object.

Global Modules

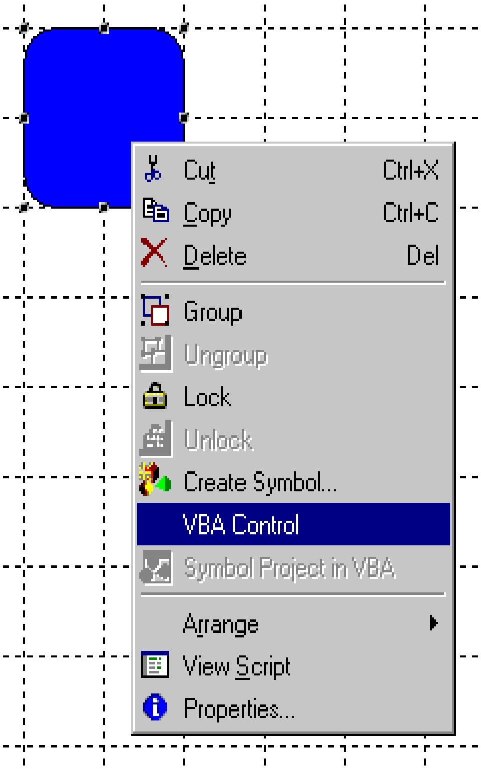
You can insert Modules and Class Modules in any VBA project. Modules in the Main project will be seen by all other projects. In the case of a name conflict use "Main." as a prefix.

ThisProject is an fvProject typed item. It is equivalent to the type Project in Enable, but has been renamed to avoid a conflict with the VBA Project object.

Using Native Drawing Elements in a Program

Dynamically Coding Events on a Drawing Element Using the Dim WithEvents Statement

Before any native drawing element (Shape, Text, Bitmap etc.) can be used as a control in a VBA p *ntrol* property in the pop-up menu displayed by right clickir



Once the drawing element has been enabled as a VBA control it becomes visible in the VBA environment and the EnableEvents property of drawing element is set to True.

- It can be selected using the left combobox in the code window.
- Once selected in the left combobox you can select any of its events in the right combobox and directly code the event.
- It is visible in the VBA property box.
- The object is directly known by its name (shape1 for example) in the VBA Project.

You can select/deselect the VBA Control property at any time. The associated code is not lost if you deselect a control, it just becomes a function of the Project.

It is possible for a drawing element to be a VBA control without the EnableEvents property being True. In this case you will be able to change its properties programmatically, but not be able to fire any events.

Each drawing element that is enabled as a control requires additional resources from the host PC. Only enable drawing elements as controls if you need to develop code for them.

Dynamically Coding Events on a Drawing Element Using the Dim WithEvents Statement

The Dim WithEvents statement allows you to code events on a drawing element without first selecting the *VBA Control* property.

This is best illustrated with an example. Within a mimic, Shape1 exists but has not been selected as a control.

```
'------ The following line must be placed in General Declarations
Dim WithEvents objShape as Graphic
```

```
'------ Assign the object when the mimic opens
Private Sub Mimic_Open ()
  Set objShape = Graphics("Shape1")
  ObjShape.EnableEvents = True
End Sub

Private Sub objShape_Click
  MsgBox "Click"
End Sub
```

You must enable events for the object using the EnableEvents method.

The references to objects that are created are temporary. When the mimic is closed, or its mode changed (from Run to Development), the reference is lost and the event will no longer work. For the event to work again the object must be reassigned to the variable and the events enabled.

Developing Code for use in Symbols

Using Drawing Elements Within a Symbol as VBA Controls | What Happens to Symbol Events? | Using BeforeEvent and Dim WithEvents to Dynamically Code Events

If you develop VBA code as part of a symbol it will be generic and executed for each instance of the symbol. To allow the development of code on a symbol you must select the *Symbol Project in VBA* property from the pop-up menu displayed when right clicking on the symbol. A new SymbolProject will appear in VBA and any code you develop will automatically be executed for any instance of the symbol. The symbol is automatically saved each time you edit any code.

Using Drawing Elements Within a Symbol as VBA Controls

To use a drawing element within a symbol as a VBA control you must select its VBA Control property before it is grouped and saved within a symbol (unless you use Dim WithEvents - see below). It will then be visible as a control within the symbol and appear in the left combobox of the code window for that symbol. Any code attached to the drawing element before it becomes part of symbol becomes a function of the mimic project, it does not become part of the symbol.

Example of how to create a Symbol containing a VBA control

1. Draw a rectangle "Shape1" and select the *VBA Control* property using the right click pop-up
2. Draw a rectangle "Shape2"
3. Group "Shape1" and "Shape2".
4. Right click on the group and select *Create Symbol*.
5. Give the Symbol a name and save it.
6. Right click on the newly created Symbol and select *Symbol Project in VBA*.
7. Display the VBA environment and the Symbol will appear in its own SymbolProject project.

If you modify a Symbol by ungrouping it you will loose any code attached to drawing elements within the symbol. To avoid this duplicate it first so that there is always at least one instance of the Symbol in the mimic. In doing this the VBA project automatically reattaches the code to the modified Symbol.

What Happens to Symbol Events?

By default, the *EnableEvents* property of the Symbol itself is set to false and the Symbol is unable to respond to events. However, any drawing elements within the symbol that are VBA controls are able to respond to events (assuming their *EnableEvents* property is True).

If you set the *EnableEvent* property of the Symbol to true then the Symbol itself will respond to events but any VBA control within it will not. You will still be able to change the properties of any VBA control within the Symbol programmatically.

Using BeforeEvent and Dim WithEvents to Dynamically Code Events

The Symbol object has two special events, BeforeEvent and AfterEvent. These are fired even if the EnableEvents property of the Symbol is False.

- The BeforeEvent event occurs before each event fired from any drawing element in the Symbol with the EnableEvents property set to True.
- The AfterEvent event occurs after each event fired from any drawing element in the Symbol with the EnableEvents property set to True.

Using BeforeEvent in conjunction with the Dim WithEvents statement allows you to code events on a drawing element in a Symbol without first selecting the *VBA Control* property.

This is best illustrated with an example. Within a Symbol, Shape1 exists and has the EnableEvents property set to True but has not been selected as a control.

```
'------ The following line must be placed in General Declarations
Dim WithEvents objGraphic as Graphic

'------ Assign the object when using the BeforeEvent Event
Private Function Symbol_BeforeEvent () As Boolean
  Set objGraphic = Graphics("Shape1")
End Function

'------ Code to execute when Shape1 is clicked
Private Sub objShape_Click
  MsgBox "Click"
End Sub
```

Using OPC Variables in a Program

Subscribing and Unsubscribing a Variable | Shortcuts for Variable Access | Running Code When a Variable's Value Changes

The Variables (also TheseVariables) collection provides access to all currently subscribed variables. A variable is automatically subscribed when used in an animation in an opened mimic, or can be manually subscribed using the Variables.Add method. The collection includes both OPC and temporary variables (those with names ending with a % sign)

Subscribing and Unsubscribing a Variable

How to subscribe a variable

To subscribe a variable you use the Variables.Add method. For example:

```
Variables.Add "Main:Branch1.B1", fvVariableTypeBit
Variables.Add "CommentDisplay%", fvVariableTypeText
```

The optional second parameter describes the variable type. For more information see the Variables Collection topic

If the variable is already subscribed by an animation it may still be subscribed using this method. When the mimic containing the animation is closed the variable will remain subscribed.

You must use the full variable name including the name of the cluster if it is an OPC variable.

How to unsubscribe a variable

To unsubscribe a variable you use the Variables.Remove method. For example:

```
Variables.remove "Main:Branch1.B1"
```

Shortcuts for Variable Access

The Item method is the default for the Variables collection, and the Value property is the default for the Variable object. Therefore:

```
Variables("CommentDisplay") = "Hello World"
```

Is the equivalent of:

```
Variables.Item("CommentDisplay").Value = "Hello World"
```

The [...] syntax provides an additional shortcut to directly access a variable.

```
[Main.Branch1.B1]
```

Is the equivalent of:

```
Variables("Main.Branch1.B1")
```

It is recommended that you use this shortcut to make your code more readable. For example you could write:

```
Private Sub Shape1_Click()
  If [Main.Branch1.B1] = True Then
    [Main.Branch1.B1] = False
  Else
    [Main.Branch1.B1] = True
  End If
End Sub
```

Running Code When a Variable's Value Changes

You can execute code when a variable's value changes using the Dim WithEvents statement and the ValueChange event.

This is best illustrated with a simple example.

```
'------ This code must be placed in General Declarations
Dim WithEvents objOPCVar As Variable

'------ Execute to set up events
Private Sub fvProject_StartupComplete()
'------ Subscribe to variables
  Variables.Add "Main:Branch1.B1", fvVariableTypeBit
' ------ Assigns an object reference to the variable and enable the events
  Set objOPCVar = [Main:Branch1.B1]
  objOPCVar.EnableEvents = True
End Sub

'------ Execute when value of Main:Branch1.B1 changes
Private Sub objOPCVar_ValueChange()
  MsgBox objOPCVar.Name & " = " & objOPCVar.Value
End Sub
```

You must enable events for the object using the EnableEvent method.

If the code is contained within a mimic any references to objects are temporary. When the mimic is closed, or its mode changed (from Run to Development), the reference is lost and the event will no longer work. For the event to work again the object must be reassigned to the variable and the events enabled.

Hints and Tips

The following hints & tips (not presented in any particular order) have been accumulated during the development of this manual. They do not concern programming techniques but more the way in which you use the environment. The novice programmer or HMI user may find them useful to read.

Stopping a VBA Program

To stop a VBA program that cannot be stopped in any other way (if it is in an infinite loop for example) you can press Ctrl + Break.

Native HMI Objects

The HMI Object Hierarchy

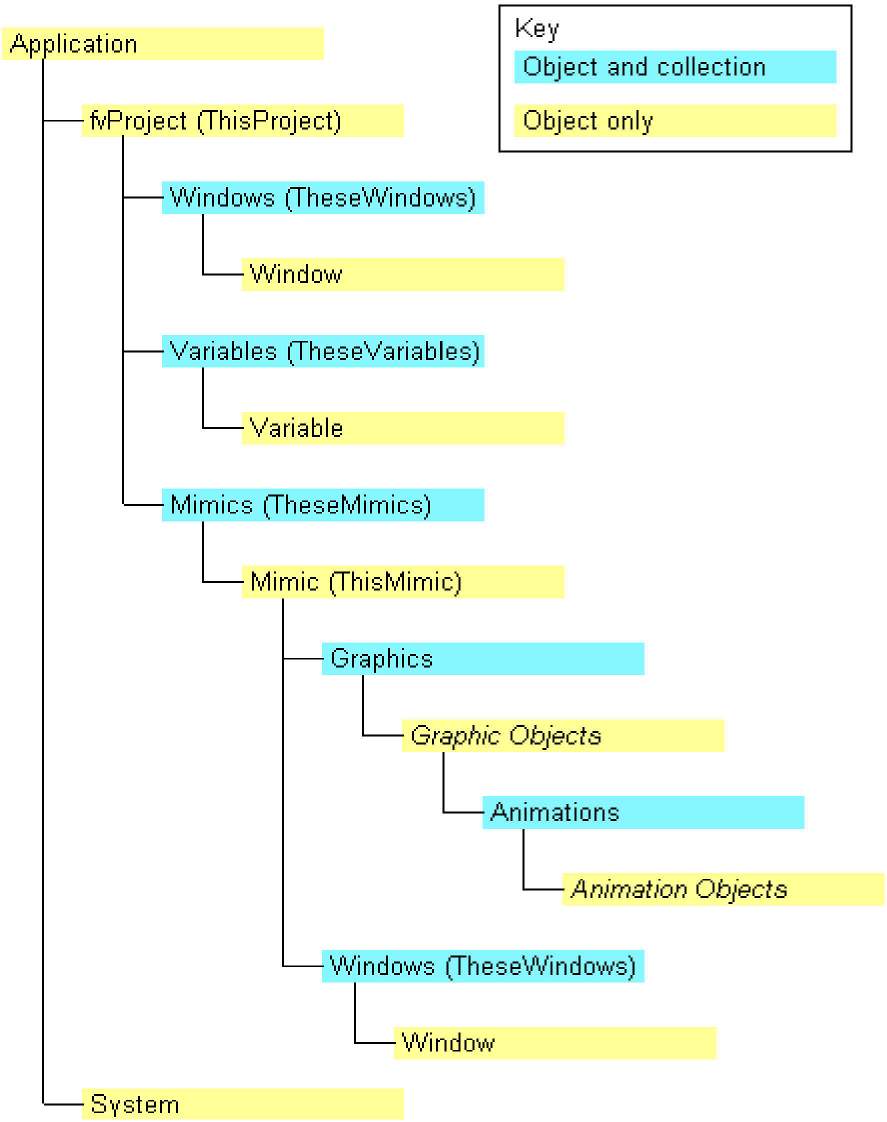
The HMI's object's properties, methods and events allow the User to develop any customised behaviour needed for the supervision and control of the process. Some object instances are always available as part of the HMI core functionality (for example ThisProject) whereas others will be created as part of the application (for example an opened mimic).

Class	Represents	Instance in the HMI
ActiveX	A 3rd party ActiveX object. Its properties, methods, functions and events are exposed once it is inserted into a mimic. For information on the behaviour of the object see the documentation provided by the vendor.	
Application	The HMI application.	Application
AVI	An AVI object.	
Bitmap	A bitmap object. This class supports BMP and JPG image formats.	
fvProject	The currently opened project.	ThisProject
GIF	A GIF object. This class supports GIF image formats.	
Graphics	A collection representing all drawing objects within a mimic. The following object classes are included. ActiveX AVI Bitmap GIF Group Metafile OLE Poly Shape Symbol Text	Graphics
Group	A group object. A number of drawing elements that have been grouped together. (Not a symbol inserted from the libraries)	
Metafile	A metafile object. This class supports WMF and EMF image formats.	
Mimic	An opened mimic.	ThisMimic
Mimics	A collection representing all opened Mimic objects.	TheseMimics, Mimics

Class	Represents	Instance in the HMI
OLE	A 3rd party OLE object. Its properties, methods, functions and events are exposed once it is inserted into a mimic. For information on the behaviour of the object see the documentation provided by the vendor	
Poly	A polygon/polyline object. This class includes the Line, Polyline, Polygon and Bézier curve drawing elements.	
Shape	A shape object. This class includes the Rectangle, Rounded Rectangle, Ellipse, Arc and Half-arc drawing elements.	
Symbol	A symbol object. A native HMI symbol from the symbol library. (previously known as an object).	
System	A set of properties that describes the system on which the HMI is running.	ThisSystem, System
TemplateGraphics	A collection representing all drawing objects within a mimic template. Identical to the Graphics collection except that it points to the Graphics of a mimic template. If the mimic has no template then TemplateGraphics is redirected to Graphics.	TemplateGraphics
Text	A text object. This class includes the text drawing element.	
Variable	A permanently subscribed variable. It provides properties, methods and events for variables of all types.	
Variables	A collection representing all permanently subscribed variables.	Variables, TheseVariables
Window	A view of an opened mimic. A mimic can have several views opened simultaneously. For example, one in design mode and the other in run mode.	
Windows	A collection representing all opened Window objects.	Windows

The HMI Object Hierarchy

To manipulate native HMI objects the User must know the relationships between them. The highest level object is Application with all other objects subordinate to it. Using the Application object's properties and methods you are able to access all objects subordinate to it. The object hierarchy is shown below.



Graphic Objects is a generic term referring to all drawing object classes (Shape, Text etc.).

Animation Objects is a generic term referring to all animation object classes. Specific animation classes are not available in this version

There is also a collection known TemplateGraphics that is identical to Graphics except that refers to the Graphics in a mimic template.

Application Object

Properties | Methods | Examples

The Application object represents the HMI itself and is the topmost object in the HMI native object hierarchy. From the Application object, you can directly access other native HMI objects by using the Application object's properties and methods, or you can indirectly access objects through other objects obtained by these properties and methods.

Application Object Properties

Property	As	R/W	Description
ActiveMimic	Mimic	R	Returns a Mimic object that represents the active mimic (the one with focus).
ActiveProject	Project	R	Returns a Project object that represents the currently opened project.
ActiveWindow	Window	R	Returns a Window object that represents the active mimic (the one with focus).
Application	Application	R	Returns the Application object itself.
Build	String	R	Returns the build number of the HMI.
Mimics	Mimics	R	Returns the collection of all opened mimics.
Name*	String	R	Always returns "Application"
Path	String	R	Returns the path of the application.
PathSeparator	String	R	Returns the character separator used in the folder names by the system on which the application is running ("\" under Windows platform).
ProductName	String	R	Returns the product name.
System	System	R	Returns the System object that describes the system on which the application is running.
Variables	Variables	R	Returns the collection of variables currently subscribed.
VBAVersion	String	R	Returns the version number of VBA.
VBE	Object	R	Returns the Visual Basic Environment root object that contains all other objects and collections represented in Visual Basic for Applications.
Version	String	R	Returns the version number of the application.
Visible	Boolean	R/W	Returns and sets the visibility property. When visibility is True the HMI is visible . When it is False it is hidden. WHEN USING THIS PROPERTY TO HIDE THE HMI YOU MUST ALSO ARRANGE A MECHANISM TO MAKE IT VISIBLE AGAIN.

Application Object Methods

Method	Description	
PrintOut	Sub PrintOut([Copies As Integer], [Orientation As fvOrientation], [Printer As String]) Prints everything in the workspace excluding menu and toolbars.	
	Copies	Optional argument. Default: 1 A number that specifies the number of copies to be printed.
	Orientation	Optional argument. Default: fvPortrait Orientation can take one of the following fvOrientation values: fvLandscape, fvPortrait
	Printer	Optional argument A string that indicates the printer to be used. If not specified then the system default printer is used.

Application Object Examples

```
Private Sub TextApplicationEx1_Click()
'-------- Returns information about the application
  TextInfo1.Text = Application.Name & " Version " & Application.Version
  TextInfo2.Text = "Product name  " & Application.ProductName
End Sub

Private Sub TextApplicationEx2_Click()
'-------- Opens a mimic and returns the mimic object
  Dim objMimic As Object
  Set objMimic = Application.Mimics.Open("TEST")
  MsgBox objMimic.Name & " mimic is opened "
End Sub

Private Sub TextApplicationEx3_Click()
  Application.PrintOut 1, fvLandscape
End Sub
```

The fvProject Object

Properties | Methods | Events | Examples

The fvProject object represents the currently opened project. This object is available through the ActiveProject property of the Application object. ThisProject is a pre-defined instance of the object.

fvProject Object Properties

Property	As	R / W	Description
ActiveKeyboardLayout	String	R/W	Returns and sets the name of the active keyboard layout.
ActiveMimic	Mimic	R	Returns the Mimic object that represents the active mimic of the project.
Application	Application	R	Returns the Application object from which the Project object is stemmed.
Mimics	Mimics	R	Returns the collection of all opened mimics.
Parent	Application	R	Returns the parent object of the Project object.
Path	String	R	Returns the path of the project.
ProjectName	String	R	Returns the name of the project.
UserFullName	String	R	Returns the full name of the user currently logged-on.
UserName	String	R	Returns the name of the user currently logged-on.
UserDescription	String	R	Returns the description of the user currently logged-on.
Variables	Variables	R	Returns the collection of all permanently subscribed variables.
Windows	Windows	R	Returns the collection of all opened windows.

fvProject Object Methods

Method	Description	
LogoffUser	Sub LogoffUser ([ShowDialog As Boolean]) Logs off the user currently logged-on.	
	ShowDialog	Optional argument. Default: True A Boolean that indicates if the logoff dialog must (True) or must not (False) be displayed.
LogonUser	Sub Logon User([ShowDialog As Boolean], [Name As Variant], [Password As Variant]) Logs on a new user.	
	ShowDialog	Optional argument. Default: True A Boolean that indicates if the Logon dialog must (True) or must not (False) be displayed.
	Name	Optional argument. Default: “” A String that specifies the name of the user that must be logged-on.
	Password	Optional argument. Default: “” A String that specifies the password of the user that must be logged-on.
Quit	Sub Quit() Closes all the opened mimics and quits the application.	

fvProject Object Events

Event	Description	
BeforeQuit	Sub fvProject_BeforeQuit(Cancel As Boolean) Occurs when project is about to be quit.	
	Cancel	Boolean value that can be set to True if the project must not be quit.
StartupComplete	Sub fvProject_StartupComplete() Occurs when project loading is complete.	
UserChanged	Sub fvProject_UserChanged() Occurs when the user currently logged-on has changed or if user information database has changed.	

fvProject Object Examples

```
Private Sub fvProject_StartupComplete()
'-------- Subscribe local variables
  Variables.Add "User%", fvVariableTypeText
  Variables.Add "Time%", fvVariableTypeText
  Variables.Add "Date%", fvVariableTypeText
'-------- Get Default user name for display in local variable
  [User%] = ThisProject.UserName
'-------- Open first mimic
  ThisProject.Mimics.Open "Menu"
 End Sub

Private Sub fvProject_UserChanged()
  Dim strUserName As String
  strUserName = ThisProject.UserName
'-------- Set keyboard layout - assumes layout name is the same as user name
  ThisProject.ActiveKeyboardLayout = strUserName
  [User%] = strUserName
 End Sub
```

The System Object

Properties | Methods | Examples

The System object represents the system on which the HMI is running. This object is available through the System property of the Application object. ThisSystem is a pre-defined instance of the object.

System Object Properties

Property	As	R/W	Description
Application	Application	R	Returns the Application object from which the System object is stemmed.
ComputerName	String	R	Returns the name of the computer on which the HMI is running.
HorizontalResolution	Long	R	Returns the horizontal resolution (in pixels) of the screen.
MemoryTotal	Long	R	Returns the total size, in kilobytes, of physical memory.
NumberOfProcessors	Long	R	Returns the number of processors in the system.
OperatingSystem	String	R	Returns the name of the operating system currently used.
Parent	Application	R	Returns the parent object of the System object.
Version	String	R	Returns the version number of the currently used operating system.
VerticalResolution	Long	R	Returns the vertical resolution (in pixels) of the screen.

System Object Methods

Method	Description	
PrintOut	Sub PrintOut([Copies As Integer], [Orientation As fvOrientation], [Printer As String]) Prints the entire screen.	
	Copies	Optional argument. Default: 1 A number that specifies the number of copies to be printed.
	Orientation	Optional argument. Default: fvPortrait Orientation can take one of the following fvOrientation values: fvLandscape, fvPortrait
	Printer	Optional argument A string that indicates the printer to be used. If not specified then the system default printer is used.

System Object Examples

```
Private Sub TextSystemEx1_Click()
'-------- Get the operating system name and screen resolution
'-------- Note the use of underscore as a continuation character
  TextInfo1.Text = "Operating System " & ThisSystem.OperatingSystem
  TextInfo2.Text = " Screen Resolution " & ThisSystem.HorizontalResolution _
               & " x " & ThisSystem.VerticalResolution
End Sub

Private Sub TextSystemEx2_Click()
'-------- Print the entire screen
  ThisSystem.PrintOut 1, fvLandscape
End Sub
```

The Mimics Collection Object

Properties | Methods | Examples

The Mimics collection object represents the collection of the currently opened mimics for the active Project.

Mimics Object Properties

Property	As	R/W	Description
Application	Application	R	Returns the Application object from which Mimics object is stemmed.
Count	Long	R	Returns the number of Mimic object in the collection.
Parent	Object	R	Returns the parent object of the Mimics object (For Mimics, the Parent object is always an fvProject object).

Mimics Object Methods

<table>
<tr><th>Method</th><th colspan="2">Description</th></tr>
<tr><td rowspan="2">Close</td><td colspan="2">Sub Close([SaveChanges As Variant])
Closes all the mimics handled in the collection.</td></tr>
<tr><td>SaveChanges</td><td>Optional argument.
Indicates if changes must be saved or not. SaveChanges can take one of the following fvSaveAction values: fvDoNotSaveChanges, fvSaveChanges or fvPromptToSaveChanges.
Default: fvPromptToSaveChanges</td></tr>
<tr><td rowspan="5">IsOpen</td><td colspan="2">Function Open(Name As String, [Branch As Variant], [ReferenceSet As Variant], [Region As Variant]) As Boolean
Tests to see if a Mimic is open according to the passed arguments. Returns true if it is open, else false.</td></tr>
<tr><td>Name</td><td>Name of the mimic.</td></tr>
<tr><td>Branch</td><td>Optional argument. Default: “”
String that specifies the name of the branch that must be used to open the Mimic.</td></tr>
<tr><td>ReferenceSet</td><td>Optional argument. Default: fvRefSetRealtime
The fvRefSet that must be passed to open the Mimic. ReferenceSet can take one of the following fvRefSet values: fvRefSetRealtime, fvRefSet1, fvRefSet2 or fvRefSetTest.</td></tr>
<tr><td>Region</td><td>Optional argument. Default: 1
Long that specifies the region that must be used to open the Mimic.</td></tr>
</table>

Method	Description	
Item*	Function Item(Index As Variant, [Branch As Variant], [ReferenceSet As Variant], [Region As Variant]) As Mimic Returns the Mimic object of the collection that matches the passed arguments.	
	Index	A Long with a range from 1 to Mimics.Count, which represents the position of the mimic object in the collection, that specifies the item that must be returned. In this case, optional arguments, if they are specified, are discarded. Or A String, which represents the name of the mimic object, that specifies the item that must be returned. In this case, optional arguments, if they are specified, are taken into account.
	Branch	Optional argument. Default: “” A String, which represents the name of the branch of the mimic object, that specifies the item that must be returned.
	ReferenceSet	Optional argument. Default: fvRefSetRealtime A fvRefSet, which represents the reference set of the mimic object, that specifies the item that must be returned. A ReferenceSet can take one of the following fvRefSet values: fvRefSetRealtime, fvRefSet1, fvRefSet2, fvRefSetTest.
	Region	Optional argument. Default: 1 A Long, which represents the region of the mimic object, that specifies the item that must be returned.

Method	Description	
Open	Function Open(FileName As String [,Branch As String] [,ReferenceSet As Variant] [,Region As Integer] [,ParentName As String] [,ParentBranch As String] [,ParentReferenceSet As Integer] [,ParentRegion As Integer] [,LeftPos As Integer] [,TopPos As Integer] [,AbsolutePos As Boolean]) As Mimic Opens and returns a Mimic object.	
	FileName	A string containing the name of the mimic.
	Branch	Optional argument. Default: “” String that specifies the name of the branch that must be used to open the Mimic.
	ReferenceSet	Optional argument. Default: fvRefSetRealtime The fvRefSet that must be passed to open the Mimic. ReferenceSet can take one of the following fvRefSet values: fvRefSetRealtime, fvRefSet1, fvRefSet2 or fvRefSetTest.
	Region	Optional argument. Default: 1 An Integer that specifies the region that must be used to open the Mimic.
	ParentBranch	Optional argument. Default "" A string specifying the name of the parent mimic when opening a mimic as a child.
	ParentReferenceSet	Optional argument. Default: fvRefSetRealtime The reference set of parent mimic when opening a mimic as a child. ParentReferenceSet can take one of the following fvRefSet values: fvRefSetRealtime, fvRefSet1, fvRefSet2 or fvRefSetTest.
	ParentRegion	Optional argument. Default: 1 An integer that specifies the region of the parent mimic when opening a mimic as a child.
	LeftPos	Optional argument. Default 0. A long indicating the opening position of a child window relative to the parent.
	TopPos	Optional argument. Default 0. A long indicating the opening position of a child window relative to the parent.
	AbsoloutePos	Optional Argument. Default True. If True then LeftPos and TopPos are absolute, else they are relative to the original saved position of the child mimic.

Save	Sub Save([NoPrompt As Variant]) Saves all the mimics handled in the collection.	
	NoPrompt	Optional argument. Indicates if the user must be prompted (False) or not (True). NoPrompt can take one of the following Boolean values: True or False. Default: True

Mimics Object Examples

```
Private Sub TextMimicsEx1_Click()
'-------- Open the Menu mimic and return its object
  Dim objMimic As Object
  Set objMimic = Mimics.Open("Menu")
End Sub

Private Sub TextMimicsEx2_Click()
'--------Loop through all opened mimics
  Dim objMimic As Object
  For Each objMimic In Mimics
     MsgBox objMimic.FullName
  Next
End Sub
```

The Mimic Object

Properties | Methods | Events | Examples

A Mimic object represents an opened mimic. Mimic's properties or methods are related to the targeted mimic and to all windows (views) opened on this mimic (including the active one). For example, BackColor property reflects the background colour of the mimic. A change to this property will change the background colour of all windows (views) opened on this mimic.

Mimic Object Properties

Property	As	R / W	Description
AlwaysActive	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the mimic's windows must (True) or must not (False) be always active (regarding other non always active windows).
Application	Application	R	Returns the Application object from which Mimic object is stemmed.
BackColor	OLE_Color	R/W	Returns or sets the RGB code of the background colour of the mimic.
BackColorIndex	Integer	R/W	Returns or sets the index of the indexed colour cell used as background colour of the mimic. (this index can vary from 1 to 32 if the chosen colour cell is indexed, if the chosen colour cell is fixed, this index is set to 0).
Branch	String	R	Returns the branch name used to open the mimic.
Cache	Boolean	R/W	Returns or sets the Boolean attribute that indicates that the mimic must be cached.
Caption	String	R/W	Returns or sets the text displayed in the title bar of the windows that display the mimic.
DeclutteringModel	String	R/W	Returns or sets the name of the decluttering model that must be used by the mimic.
EnableEvents	Boolean	R/W	Returns or sets the Boolean that indicates if events must (True) or must not (False) be fired for a mimic. When this property is disabled, mimic associated scripts are not performed.
EnableLayerSelection	Boolean	R/W	Returns or sets the Boolean that indicates if user can (in run mode) use the "Layers" tool mimic in order to modify the layer selection currently applied to the mimic's windows. Note: this boolean doesn't disable layer selection from script.

Property	**As**	**R / W**	**Description**
EnableZoom	Boolean	R/W	Returns or sets the Boolean that indicates if user can (in run mode) use the "Zoom" tools in order to modify the zoom currently applied to the mimic's windows. Note: this boolean doesn't disable zooming from script.
FullName	String	R	Returns the full name (the file's one) of the mimic.
Graphics	Graphics	R/W	Returns the collection of all graphic objects placed on the mimic.
GridColor	OLE_Color	R/W	Returns or sets the RGB code of the grid colour of the mimic.
GridColorIndex	Integer	R/W	Returns or sets the index of the indexed colour cell used as grid colour of the mimic (this index can vary from 1 to 32 if the chosen colour cell is indexed, if the chosen colour cell is fixed, this index is set to 0).
IncMimicHeightFromCorner	Long	R/W	Returns or sets the vertical distance between the two nearest corners of a newly opened included mimic and the animation that has caused its opening.
IncMimicOpeningPosition	fvOpenPosition	R/W	Returns or sets the attribute that indicates the position of a newly opened included mimic according to the animation that has caused its opening. IncMimicOpeningPosition can take one of the following fvOpenPosition values: FvOpenPositionNone fvOpenPositionTopLeftCorner fvOpenPositionBottomLeftCorner fvOpenPositionTopRightCorner fvOpenPositionBottomRightCorner fvOpenPositionOnObject fvOpenPositionBottomObject fvOpenPositionLeftObject fvOpenPositionTopObject fvOpenPositionRightObject fvOpenPositionBottomRightObject
IncMimicPopUp	Boolean	R/W	Returns or sets the Boolean attribute that indicates if newly opened included mimic for the mimic are exclusive (popup behavior = True) or not (False).
IncMimicWidthFromCorner	Long	R/W	Returns or sets the horizontal distance between the two nearest corners of a newly opened included mimic and the animation that has caused its opening.
Index	Integer	R	Returns the index of the Mimic object in the Mimics collection.

Property	As	R/W	Description
MaximumZoom	Long	R/W	Returns or sets the maximum limit of the zoom factor that can be applied (in run mode) to the mimic's windows using the "Zoom" tools. Without limits, zoom factor may vary from 0 to 6400. A zoom factor of 100 means original size. Note: this limit is not taken into account when zooming is performed by script.
MinimumZoom	Long	R/W	Returns or sets the minimum limit of the zoom factor that can be applied (in run mode) to the mimic's windows using the "Zoom" tools. Without limits, zoom factor may vary from 0 to 6400. A zoom factor of 100 means original size. Note: this limit is not taken into account when zooming is performed by script.
Modal	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the mimic's windows must (True) or must not (False) be modal.
Name	String	R/W	Returns the name used in code to identify the Mimic. Note: The default name for new object is the kind of object plus a unique integer.
Parent	Object	R	Returns the parent object of the Mimic object.
Path	String	R	Returns the path name (the file's one) of the mimic.
ReferenceSet	fvRefSet	R/W	Returns or sets the reference set used by the mimic's windows. ReferenceSet can take one of the following fvRefSet values: fvRefSetRealtime, fvRefSet1, fvRefSet2 or fvRefSetTest.
Region	Long		Returns or sets the region number where are displayed the mimic's windows.
Modified	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the mimic was modified since its opening. True indicates it has been modified.
ShowCaption	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the windows that display the mimic must (True) or must not (False) display its caption in their title bar.
SecurityLevel	Long	R/W	Returns or sets the security level needed to open the mimic.
Selection	Object	R	Returns the currently selected object (*Graphic* object) on the mimic.

Property	As	R/W	Description
Tag	Variant	R/W	Returns or sets a general purpose variant within the mimic object. The value is not saved.
TemplateGraphics	Graphics	R/W	Returns the collection of all graphic objects placed on the mimic template.
TemplateName	String	R/W	Returns or sets the name of the template mimic that must be used by the mimic.
UseTemplateAccessRights	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the template mimic's access rights properties must (True) or must not (False) be used by the mimic's windows. Access Rights refer to the following properties of the template mimic: EnableLayerSelection EnableZoom MaximumZoom MinimumZoom SecurityLevel WarningBeep WarningBox
UseTemplateBackground	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the template mimic's background colour properties must (True) or must not (False) be used by the mimic. Background colour properties refer to the following properties of the template mimic: BackColor BackColorIndex.
UseTemplateCaption	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the template mimic's Caption property must (True) or must not (False) be used by the mimic's windows.
UseTemplateGrid	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the template mimic's grid properties must (True) or must not (False) be used by the mimic's windows. Grid properties refer to the following properties of the template mimic: GridColor, GridColorIndex, GridDistanceHorizontal, GridDistanceVertical, ShowGrid, SnapToGrid.

Property	As	R/W	Description
UseTemplateIncludedMimic	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the template mimic's included mimic properties (True) or must not (False) be used by the mimic. Included mimic properties refer to the following properties of the template mimic: IncMimicHeightFromCorner, IncMimicOpeningPosition , IncMimicPopUp, IncMimicWidthFromCorner.
UseTemplateLayers	Boolean (Editable)	R/W	Returns or sets the Boolean attribute that indicates if the template mimic's Layers property must (True) or must not (False) be used by the mimic's windows.
UseTemplatePosition	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the template mimic's position properties must (True) or must not (False) be used by the mimic's windows. Position properties refer to the following properties of the template mimic: Left Top
UseTemplateSize	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the template mimic's size properties must (True) or must not (False) be used by the mimic's windows. Size properties refer to the following properties of the template mimic: Height Width

Property	As	R/W	Description
UseTemplateStyles	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the template mimic's styles properties must (True) or must not (False) be used by the mimic's windows. Styles properties refer to the following properties of the template mimic: AlwaysActive, Border Cache ClientBorder CloseButton ControlBox EnableMove EnableResize MaxButon MinButton Modal ShowCaption StaticBorder StretchContentsToFitSize TitleBar ToolMimic
Windows	Windows	R	Returns the collection of all opened windows that refer to the mimic.
WarningBeep	Boolean	R/W	Returns or sets the Boolean attribute that indicates if a warning beep must (True) or must not (False) occur when user attempts to perform an unauthorised action.
WarningBox	Boolean	R/W	Returns or sets the Boolean attribute that indicates if a warning message box must (True) or must not (False) be displayed when user attempts to perform an unauthorised action.

Mimic Object Methods

Method	Description
Activate	Sub Activate () Activate (sets focus) the mimic. If the mimic is opened in more than one window it will activate the first window.

Method	Description	
Close	Sub Close([SaveChanges As fvSaveAction]) Closes all windows of the mimic object.	
	SaveChanges	Optional argument. Default: fvPromptToSaveChanges SaveChanges can take one of the following fvSaveAction values: fvDoNotSaveChanges, fvPromptToSaveChanges, fvSaveChanges
NewWindow	Function NewWindow () As Window Opens and returns a new window for the mimic.	
Save	Sub Save () Saves the mimic object.	
SaveAs	Sub SaveAs(FileName As String) Saves the mimic object under a different file name. The original mimic is closed and the new mimic opened.	
	FileName	The name of the file in which the mimic object will be saved.
SaveCopy As	Sub SaveCopyAs(FileName As String) Saves the mimic object under a different file name. The original mimic is remains open.	
	FileName	The name of the file in which the mimic object will be saved.
ScrollTo	Sub ScrollTo(Horizontal As Long, Vertical As Long, Window As Integer) Scrolls the mimic within the window. The horizontal and vertical co-ordinates define the reference point of the mimic that must be placed, if possible, at the centre of the window after the scroll operation.	
	Horizontal	A long that defines the horizontal co-ordinate of the reference point for the scroll operation.
	Vertical	A long that defines the vertical co-ordinate of the reference point for the scroll operation.
	Window	A integer representing the window in which the operation is to take place.

Mimic Object Events

Event	Description	
BeforeClose	Sub Mimic_BeforeClose(Cancel As Boolean) Occurs when the mimic is about to be closed.	
	Cancel	Boolean value that can be set to True if the mimic must not be closed.
	Cancel	Boolean value that can be set to True if the mimic must not be printed.
Open	Sub Mimic_Open() Occurs when the mimic object is opened.	

Mimic Object Examples

```
Private Sub Mimic_Open()
'-------- On event mimic_open initialise output strings
  Text1.Text = ""
  Text2.Text = ""
  Text3.Text = ""
  Text4.Text = ""
  Text5.Text = ""
  Text6.Text = ""
End Sub

Private Sub TextMimicEx1_Click()
'-------- Set the focus to the Menu mimic (mimic must be open)
  Dim objMimic As Object
  Set objMimic = Mimics.Item("Menu")
  objMimic.Activate
End Sub
```

The Windows Collection Object

Properties | Methods

The Windows collection object represents the collection of the currently opened windows for the active Project, or the collection of currently opened windows for a specific Mimic

Windows Object Properties

Property	As	R/W	Description
Application	Application	R	Returns the Application object from which Windows object is stemmed.
Count	Long	R	Returns the number of Window objects in the collection.
Parent	Object	R	Returns the parent object of the Windows object.

Windows Object Methods

<table>
<tr><th>Method</th><th colspan="2">Description</th></tr>
<tr><td rowspan="2">Item*</td><td colspan="2">Function Item(Index As Variant) As Window
Returns the Window object of the collection that matches the passed arguments.</td></tr>
<tr><td>Index</td><td>A Long with a range from 1 to Windows.Count, which represents the position of the window object in the collection, that specifies the item that must be returned.</td></tr>
<tr><td>Refresh</td><td colspan="2">Sub Refresh()
Refreshes all opened windows (if the collection is the one handled by the Application object).
Refreshes all opened windows for a particular mimic (if the collection is the one handled by a Mimic object).</td></tr>
</table>

The Window Object

Properties | Methods | Examples

A Window object represents an opened window on a specified Mimic.

Window Object Properties

Property	As	R/W	Description
Application	Application	R	Returns the Application object from which Window object is stemmed.
Border	Boolean	R/W	Returns or sets the Boolean attribute that indicates if a border style must (True) or must not (False) be applied to the window.
ClientBorder	Boolean	R/W	Returns or sets the Boolean attribute that indicates if a client border style must (True) or must not (False) be applied to the window.
CloseButton	Boolean	R/W	Returns or sets the Boolean attribute that indicates if a close button must (True) or must not (False) be accessible on the window.
ControlBox	Boolean	R/W	Returns or sets the Boolean attribute that indicates if a Control-menu box must (True) or must not (False) be accessible on the window. Control-menu box refers to all buttons and menus accessible via the Title Bar of a window.
EnableMove	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the window must (True) or must not (False) be moveable
EnableResize	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the window must (True) or must not (False) be resizable.
GridDistanceHorizontal	Long	R/W	Returns or sets the distance that separates horizontally two points of the active window's grid.
GridDistanceVertical	Long	R/W	Returns or sets the distance that separates vertically two points of the active window's grid.
Height	Long	R/W	Returns or sets the height of the window.
Layers	Long	R/W	Returns or sets the layers currently displayed on the window (Sixteen different layers are available (0..15), the layers property of a window is stored as a long. If layer n and m are selected for this window, layers property is set to $2^n + 2^m$).

Property	As	R/W	Description
Left	Long	R/W	Returns or sets the distance between the internal left edge of the window and the left edge of its container.
MaxButon	Boolean	R/W	Returns or sets the Boolean attribute that indicates if a maximise button must (True) or must not (False) be accessible on the window.
MinButton	Boolean	R/W	Returns or sets the Boolean attribute that indicates if a minimise button must (True) or must not (False) be accessible on the window.
Mode	fvWindowMode	R/W	Returns or sets the mode of the window. Mode can take one of the following fvWindowMode values: fvWindowModeRun, or fvWindowModeDesign.
Parent	Windows	R	Returns the parent object of the Window object.
ShowGrid	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the window must (True) or must not (False) display the grid.
SnapToGrid	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the window must (True) or must not (False) activate the grid.
StaticBorder	Boolean	R/W	Returns or sets the Boolean attribute that indicates if a static border style must (True) or must not (False) be applied to the window.
StretchContentsToFitSize	Boolean	R/W	Returns or sets the Boolean attribute that indicates if the window's content must (True) or must not (False) be stretched when this window is resized.
TitleBar	Boolean	R/W	Returns or sets the Boolean attribute that indicates if a title bar must (True) or must not (False) be shown by the window.
ToolWindow	Boolean	R/W	Returns or sets the Boolean attribute that indicates if a tool mimic style must (True) or must not (False) be applied to the window.
Top	Long	R/W	Returns or sets the distance between the internal top edge of the window and the left edge of its container.
Width	Long	R/W	Returns or sets the width of the window.

Property	As	R / W	Description
ZoomRatio	Long	R	Returns the zoom ratio currently applied to the window.

Window Object Methods

Method	Description	
Activate	Sub Activate () Activate (sets focus) the window object.	
Close	Sub Close([SaveChanges As fvSaveAction]) Closes the windows object.	
	SaveChanges	Optional argument. Default: fvPromptToSaveChanges SaveChanges can take one of the following fvSaveAction values: fvDoNotSaveChanges, fvPromptToSaveChanges, fvSaveChanges
PrintOut	Sub PrintOut([Copies As Integer], [Orientation As fvOrientation], [Printer As String]) Prints the window object only.	
	Copies	Optional argument. Default: 1 A number that specifies the number of copies to be printed.
	Orientation	Optional argument. Default: fvPortrait Orientation can take one of the following fvOrientation values: fvLandscape, fvPortrait
	Printer	Optional argument A string that indicates the printer to be used. If not specified then the system default printer is used.

Method	Description	
ScrollTo	Sub ScrollTo(Horizontal As Long, Vertical As Long) Scrolls the mimic within the window. The horizontal and vertical co-ordinates define the reference point of the mimic that must be placed, if possible, at the centre of the window after the scroll operation.	
	Horizontal	A long that defines the horizontal co-ordinate of the reference point for the scroll operation.
	Vertical	A long that defines the vertical co-ordinate of the reference point for the scroll operation.
Zoom	Sub Zoom(ZoomValue As Long, [Horizontal As Long], [Vertical As Long]) Zooms the content of the window around a reference point. The horizontal and vertical co-ordinates define the reference point of the window's mimic that must be placed, if possible and after the zoom operation, at the centre of the window.	
	ZoomValue	A long that indicates the zoom factor (as a percentage of the original window size) to be applied to the window's content.
	Horizontal	Optional argument. Default: window.width / 2 A long that defines the horizontal co-ordinate of the reference point for the zoom operation.
	Vertical	Optional argument. Default: window.height / 2 A long that defines the vertical co-ordinate of the reference point for the zoom operation.

Window Object Examples

```
Private Sub ZoomOut_Click()
'-------- Doubles the zoom on a window of the Mimic FVOjects5
  Dim ObjWindow As Object
  Set ObjWindow = FVObjects5.Windows.Item(1)
  ObjWindow.Zoom ObjWindow.ZoomRatio*2, 0, 0
  Text1.Text = "Current zoom is " & ObjWindow.ZoomRatio
End Sub


Private Sub TextWindowEx1_Click()
'-------- Prints the active window
  Dim ThisWindow As Object
  Set ThisWindow = Application.ActiveWindow
  ThisWindow.PrintOut
End Sub
```

The Graphics Collection Object

Properties | Methods

The collection of *Graphic* objects.

Graphic is a generic term that refers to either all the native HMI drawing elements . The *Graphic* object type as such does not exist, the Graphics collection contains Object objects.

Graphics Object Properties

Property	As	R / W	Description
Application	Application	R	Returns the Application object from which Graphics object is stemmed.
Count	Long	R	Returns the number of *Graphic* object in the collection.
Parent	Object	R	Returns the parent object of the Graphics object - either a Group, Mimic or Symbol object depending on the context.

Graphics Object Methods

<table>
<tr><th>Method</th><th colspan="2">Description</th></tr>
<tr><td rowspan="2">Item</td><td colspan="2">Function Item(Index As Variant) As Object
Returns the indicated item (a <i>Graphic</i> object) from the Graphics collection.</td></tr>
<tr><td>Index</td><td>A Long with a range from 1 to Graphics.Count, which represents the position of the graphic object in the collection, that specifies the item that must be returned.
OR
A String, which represents the name of the graphic object, that specifies the item that must be returned.</td></tr>
</table>

Graphic Objects in General

Properties | Methods | Events | Examples

Graphic is a generic term referring to all native HMI drawing elements Poly, Shape, Text, Symbol and Group plus the Bitmap and Metafile objects. The *Graphic* object type as such does not exist, the Graphics collection contains Object objects.

The properties that define the appearance of all objects that belong to the Graphics collection conform to certain rules. All properties that are related to colours follow the standard colour property naming scheme.

Pattern style rules

A pattern style is defined using the following conventions:

- A background colour: the colour used to fill the object.
- A pattern style: the pattern style used to fill the object (Solid, 45° Diagonal Hatching, 135° Diagonal Hatching, Diagonal Cross Hatching, Horizontal Hatching, Vertical Hatching, Cross Hatching, Transparent).
- A hatching colour: the colour used to draw hatching.

Line style rules

A line style is defined using the following conventions:

- A foreground colour: the colour used to draw the lines of the object.
- A line style: the line style used to draw the lines of the object (Solid, Dashed, Dotted, Dash Dot, Dash Dot Dot, Invisible).
- A line thickness: the line thickness used to draw the lines of the object.

Appearance rules

Appearance is defined using the following conventions:

- An appearance: the appearance used to draw the object (Regular, Shadow, Button, Coloured Button, Relief, Inverse Relief).
- A bottom-right and a top-left colour: the colours used to draw the bottom-right and the top-left edges of the object when its appearance is not set to Regular.
- An appearance thickness: the thickness of the edges of the object when its appearance is not set to Regular.

Text colour rules

Text colours are defined using following conventions:

- A foreground colour: the colour used to draw the text contained in the text object.
- A background colour: the colour used to fill the area defined by the including rectangle of the text object.

Standard colour property naming

Each colours used in an object can be fixed, indexed, transparent or blinking. Thus multiple properties are needed to define a single colour. The names of these properties are constructed in a uniform manner using different but consistent suffixes. These suffixes are:

- Primary: Suffixed property returns or sets the OLE_COLOR code used for primary colour (information returned or set is valid only if Primary Transparent suffixed property is set to False).
- Alternate: Suffixed property returns or sets the OLE_COLOR code used for alternate colour (information returned or set is valid only if Alternate Transparent suffixed property is set to False).
- Primary Index: Suffixed property returns or sets the index of the indexed colour cell used as primary colour. Indexed colour cells refer to the ones that can be found in the

advanced part of the palette (Primary Index suffixed property can vary from 1 to 32 if the chosen colour cell is indexed, if the chosen colour cell is fixed, Primary Index suffixed property is set to 0).

- Alternate Index: Suffixed property returns or sets the index of the indexed colour cell used as alternate colour. Indexed colour cells refer to the ones that can be found in the advanced part of the palette (Alternate Index suffixed property can vary from 1 to 32 if the chosen colour cell is indexed, if the chosen colour cell is fixed, Alternate Index suffixed property is set to 0).
- Primary Transparent: Suffixed property returns or sets the transparent property that indicates that the primary colour is transparent (True) or not (False).
- Alternate Transparent: Suffixed property returns or sets the transparent property that indicates that the alternate colour is transparent (True) or not (False).
- Blink: Suffixed property returns or sets the index of the blinking frequency used for this colour (information is valid only if Primary suffixed property differs from Alternate suffixed property).

Common *Graphic* Object Properties

The following properties are common to all the *Graphic* object types.

Property	As	R/W	Description
AnchorX	Integer	R/W	Returns or sets the horizontal co-ordinate of the anchor point of the object. Anchor point co-ordinates are relative to the upper left point of the object.
AnchorY	Integer	R/W	Returns or sets the vertical co-ordinate of the anchor point of the object. Anchor point co-ordinates are relative to the upper left of the object.
Animations	Animations	R	Returns the animations collection representing the animations applied to the *Graphic* object
Application	Application	R	Returns the Application object from which the *Graphic* object is stemmed.
BackColorAlternate	OLE_Color	R/W	Returns or sets the alternate background colour of the shape. BackColorAlternate follows the Pattern Style rules.
BackColorAlternateIndex	Integer	R/W	Returns or sets the alternate background colour of the shape in the case of an indexed colour. BackColorAlternateIndex follows the Pattern Style rules.
BackColorAlternateTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the alternate background colour of the shape is transparent or not. BackColorAlternateTransparent follows the Pattern Style rules.
BackColorBlink	Integer	R/W	Returns or sets the blinking frequency index of the background colour of the shape. BackColorBlink follows the Pattern Style rules.

Property	As	R / W	Description
BackColorPrimary	OLE_Color	R/W	Returns or sets the primary background colour of the shape. BackColorPrimary follows the Pattern Style rules.
BackColorPrimaryIndex	Integer	R/W	Returns or sets the primary background colour of the shape in the case of an indexed colour. BackColorPrimaryIndex follows the Pattern Style rules.
BackColorPrimaryTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the primary background colour of the shape is transparent or not. BackColorPrimaryTransparent follows the Pattern Style rules.
Container	Object	R	Returns the object in which the graphic is contained. Either the mimic, or in the case of a graphic within a symbol, the symbol.
EnableEvents	Boolean	R/W	Returns or sets the Boolean that indicates if events must (True) or must not (False) be fired for the object..
ForeColorAlternate	OLE_Color	R/W	Returns or sets the alternate foreground colour of the shape. ForeColorAlternate follows the Line Style rules.
ForeColorAlternateIndex	Integer	R/W	Returns or sets the alternate foreground colour of the shape in the case of an indexed colour. ForeColorAlternateIndex follows the Line Style rules.
ForeColorAlternateTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the alternate foreground colour of the shape is transparent or not. ForeColorAlternateTransparent follows the Line Style rules.
ForeColorBlink	Integer	R/W	Returns or sets the blinking frequency index of the foreground colour of the shape. ForeColorBlink follows the Line Style rules.
ForeColorPrimary	OLE_Color	R/W	Returns or sets the primary foreground colour of the shape. ForeColorPrimary follows the Line Style rules.
ForeColorPrimaryIndex	Integer	R/W	Returns or sets the primary foreground colour of the shape in the case of an indexed colour. ForeColorPrimaryIndex follows the Line Style rules.
ForeColorPrimaryTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the primary foreground colour of the shape is transparent or not. ForeColorPrimaryTransparent follows the Line Style rules.

Property	As	R / W	Description
HatchingColorAlternate	OLE_Color	R/W	Returns or sets the alternate hatching colour of the shape. HatchingColorAlternate follows the Pattern Style rules.
HatchingColorAlternateIndex	Integer	R/W	Returns or sets the alternate hatching colour of the shape in the case of an indexed colour. HatchingColorAlternateIndex follows the Pattern Style rules.
HatchingColorAlternateTranparent	Boolean	R/W	Returns or sets the transparent property that indicates that the alternate hatching colour of the shape is transparent or not. HatchingColorAlternateTranparent follows the Pattern Style rules.
HatchingColorBlink	Integer	R/W	Returns or sets the blinking frequency index of the hatching colour of the shape. HatchingColorBlink follows the Pattern Style rules.
HatchingColorPrimary	OLE_Color	R/W	Returns or sets the primary hatching colour of the shape. HatchingColorPrimary follows the Pattern Style rules.
HatchingColorPrimaryIndex	Integer	R/W	Returns or sets the primary hatching colour of the shape in the case of an indexed colour. HatchingColorPrimaryIndex follows the Pattern Style rules.
HatchingColorPrimaryTranparent	Boolean	R/W	Returns or sets the transparent property that indicates that the primary hatching colour of the shape is transparent or not. HatchingColorPrimaryTranparent follows the Pattern Style rules.
Height	Integer	R/W	Returns or sets the height of the object.
Layers	Long	R/W	Returns or sets the layers on which the object is drawn (Sixteen different layers are available (0..15), the layers property of a object is stored as a long. If layer n and m are selected for this object, its layers property is set to $2^n + 2^m$). The object will be displayed only if one of the layers currently shown by its container matches one of the layers associated to it.
Left	Integer	R/W	Returns or sets the distance between the left edge of the object and the left edge of its container.

Property	As	R / W	Description
LineStyle	fvLineStyle	R/W	Returns or sets the line style of the shape (solid, dashed…). LineStyle follows the Line Style rules. LineStyle can take one of the following fvLineStyle values: FvLineStyleSolid FvLineStyleDashed fvLineStyleDotted fvLineStyleDashDot fvLineStyleDashDotDot fvLineStyleInvisible
LineThickness	Integer	R/W	Returns or sets the line width of the shape. LineThickness follows the Line Style rules.
Locked	Boolean	R/W	Returns or sets the "locked" property of the object.
MaximumZoom	Integer	R/W	Returns or sets the maximum limit of the zoom range that can be applied to the object. The object will be displayed only if the zoom factor applied to its container is inside the zoom range defined by its Maximum and Minimum Zoom properties (zoom factor for a Mimic object can vary from 0 to 6400).
MinimumZoom	Integer	R/W	Returns or sets the minimum limit of the zoom range that can be applied to the object. The object will be displayed only if the zoom factor applied to its container is inside the zoom range defined by its Maximum and Minimum Zoom properties (zoom factor for a Mimic object can vary from 0 to 6400).
Name	String	R/W	Returns or sets the name used in a script to identify the object.
Parent	Object	R	Returns the object in which the graphic is contained. Either the mimic, or in the case of a graphic within a symbol, the symbol.
PatternStyle	FvPatternStyle	R/W	Returns or sets the pattern style of the shape (solid, transparent, 45° diagonal hatching…). PatternStyle follows the Pattern Style rules. PatternStyle can take one of the following fvPatternStyle values: FvPatternStyleSolid fvPatternStyleDiag45 fvPatternStyleDiag135 fvPatternStyleDiagCross fvPatternStyleVertical fvPatternStyleHorizontal fvPatternStyleCross fvPatternStyleInvisible

Property	As	R/W	Description
Tag	Variant	R/W	Returns or sets a general purpose variant within the graphic object. The value is not saved.
Top	Integer	R/W	Returns or sets the distance between the top edge of the object and the top edge of its container
Width	Integer	R/W	Returns or sets the width of the object.

Common *Graphic* Object Methods

The following methods are common to all the *Graphic* object types.

Method	Description	
Move	Sub Move(left As Integer, [right As Integer], [width As Integer], [height As Integer]) Moves the text object.	
	left	An Integer value indicating the horizontal co-ordinate (x-axis) for the left edge of the text object in terms of the co-ordinate system used by its container (Independent of the zoom factor).
	right	Optional argument. Default: *the_object*.Right An Integer value indicating the vertical co-ordinate (x-axis) for the top edge of the text object in terms of the co-ordinate system used by its container (Independent of the zoom factor).
	width	Optional argument. Default: *the_object*.Width An Integer value indicating the new width of the object.
	height	Optional argument. Default: *the_object*.Height An Integer value indicating the new height of the object.
Size	Sub Size (width As Integer, [height As Integer]) Resizes the object.	
	width	An Integer value indicating the new width of the object.
	height	Optional argument. Default: *the_object*.Height An Integer value indicating the new height of the object.

Common *Graphic* Object Events

The following events are common to all the *Graphic* object types. Events are fired only if the EnableEvents property of the underlying object has been set to True. This happens automatically once the first event script has been associated with the object.

Event	Description	
Click	Sub object_Click() Occurs when the user presses and then releases a mouse button over the object.	
	object	Receiver object.
DblClick	Sub object_DblClick() Occurs when the user presses and releases a mouse button and then presses and releases it again over the object.	
	object	Receiver object.
MouseDown	Sub object_MouseDown(button As Integer, shift As Integer, x As Variant, y As Variant) Occurs when the user presses a mouse button over the object.	
	object	Receiver object.
	button	An integer that identifies the button that was pressed to cause the event. The *button* argument is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating the button that caused the event.
	shift	An integer that corresponds to the state of the SHIFT, CTRL, and ALT keys when the button specified in the *button* argument is pressed. A bit is set if the key is down. The *shift* argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The *shift* argument indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed. For example, if both CTRL and ALT were pressed, the value of *shift* would be 6.
	x	A number that specifies the current location of the mouse pointer. The *x* value is always expressed in terms of the co-ordinate system used by its container independently of the current zoom factor applied to this one.
	y	A number that specifies the current location of the mouse pointer. The y value is always expressed in terms of the co-ordinate system used by its container independently of the current zoom factor applied to this one.

Event	Description	
MouseEnter	Sub object_MouseEnter(button As Integer, shift As Integer, x As Variant, y As Variant) Occurs when the mouse pointer enters the area on the screen occupied by the object.	
	object	Receiver object.
	button	An integer that corresponds to the state of the mouse buttons when the event was caused. The *button* argument is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating the button that was pressed when events was caused.
	shift	An integer that corresponds to the state of the SHIFT, CTRL, and ALT keys when the event was caused. A bit is set if the key is down. The *shift* argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The *shift* argument indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed. For example, if both CTRL and ALT were pressed, the value of *shift* would be 6.
	x	A number that specifies the current location of the mouse pointer. The *x* value is always expressed in terms of the co-ordinate system used by its container independently of the current zoom factor applied to this one.
	y	A number that specifies the current location of the mouse pointer. The y value is always expressed in terms of the co-ordinate system used by its container independently of the current zoom factor applied to this one.

<table>
<tr><th><b>Event</b></th><th colspan="2"><b>Description</b></th></tr>
<tr><td rowspan="6">MouseLeave</td><td colspan="2">Sub object_MouseLeave(button As Integer, shift As Integer, x As Variant, y As Variant)
Occurs when the user takes the mouse pointer out of an object.</td></tr>
<tr><td>object</td><td>Receiver object.</td></tr>
<tr><td>button</td><td>An integer that corresponds to the state of the mouse buttons when the event was caused. The <i>button</i> argument is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating the button that was pressed when events was caused.</td></tr>
<tr><td>shift</td><td>An integer that corresponds to the state of the SHIFT, CTRL, and ALT keys when the event was caused. A bit is set if the key is down. The <i>shift</i> argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The <i>shift</i> argument indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed. For example, if both CTRL and ALT were pressed, the value of <i>shift</i> would be 6.</td></tr>
<tr><td>x</td><td>A number that specifies the current location of the mouse pointer. The <i>x</i> value is always expressed in terms of the co-ordinate system used by its container independently of the current zoom factor applied to this one.</td></tr>
<tr><td>y</td><td>A number that specifies the current location of the mouse pointer. The y value is always expressed in terms of the co-ordinate system used by its container independently of the current zoom factor applied to this one.</td></tr>
</table>

Event	Description	
MouseUp	Sub object_MouseUp(button As Integer, shift As Integer, x As Variant, y As Variant) Occurs when the user releases a mouse button over the poly object.	
	object	Receiver object.
	button	An integer that identifies the button that was released to cause the event. The *button* argument is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating the button that caused the event.
	shift	An integer that corresponds to the state of the SHIFT, CTRL, and ALT keys when the button specified in the *button* argument is released. A bit is set if the key is down. The *shift* argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The *shift* argument indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed. For example, if both CTRL and ALT were pressed, the value of *shift* would be 6.
	x	A number that specifies the current location of the mouse pointer. The *x* value is always expressed in terms of the co-ordinate system used by its container independently of the current zoom factor applied to this one.
	y	A number that specifies the current location of the mouse pointer. The y value is always expressed in terms of the co-ordinate system used by its container independently of the current zoom factor applied to this one.

Graphic Objects Examples

```
Private Sub ButtonYSmaller_Click()
'-------- Decreases height of the Boundary shape
  If (Boundary.Height > 100) Then
    Boundary.Height = Boundary.Height - 10
  End If
End Sub
```

```
Private Sub TextGraphicEx5_Click()
'-------- Gets the name of the container object
  Dim ThisContainer As Object
  Set ThisContainer = TextGraphicEx5.Container
  MsgBox "Container = " & ThisContainer.Name
End Sub
```

AVI Object

Properties | Methods

An object that represents an AVI image. The properties, methods and events are those as documented in the topic Graphic Objects in General plus the following.

AVI Object Properties

Property	As	R/W	Description
FileName	String	R	Returns the file name associated with the AVI object.
PlayCondition	Boolean	R/W	Returns or sets the PlayCondition property. A True condition initiates play. A False condition halts play.
PlayStatus	Boolean	R/W	Returns the PlayStatus property. A True condition indicates the AVI is playing. A false condition indicates the AVI is no playing.
Transparent	Boolean	R/W	Returns or sets the transparent property that indicates that a colour must be shown as transparent when the AVI object is drawn.
TransparentColor	OLE_Color	R/W	Returns or sets the colour (OLE_COLOR code) that must be shown as transparent when the AVI object is drawn (information returned or set is valid only if the Transparent property is set to True).
TransparentColorIndex	Integer	R/W	Returns or sets the colour (indexed color) that must be shown as transparent when the AVI object is drawn (information returned or set is valid only if the Transparent property is set to True).

AVI Object Methods

Method	Description
SetOriginalSize	Sub SetOriginalSize () Resets the object to its original size..

Also see the topic on common *graphic* object properties.

Bitmap Object

Properties | Methods

An object that represents a BMP, a JPG or a JPEG image. The properties, methods and events are those as documented in the topic Graphic Objects in General plus the following.

Bitmap Object Properties

Property	As	R/W	Description
FileName	String	R	Returns the file name associated with the bitmap object.
Transparent	Boolean	R/W	Returns or sets the transparent property that indicates that a colour must be shown as transparent when the bitmap object is drawn.
TransparentColor	OLE_Color	R/W	Returns or sets the colour (OLE_COLOR code) that must be shown as transparent when the bitmap object is drawn (information returned or set is valid only if the Transparent property is set to True).
TransparentColorIndex	Integer	R/W	Returns or sets the colour (indexed color) that must be shown as transparent when the bitmap object is drawn (information returned or set is valid only if the Transparent property is set to True).

Bitmap Object Methods

Method	Description
SetOriginalSize	Sub SetOriginalSize () Resets the object to its original size..

Also see the topic on common *graphic* object properties.

GIF Object

Properties | Methods

An object that represents a GIF image. The properties, methods and events are those as documented in the topic Graphic Objects in General plus the following.

GIF Object Properties

Property	As	R/W	Description
FileName	String	R	Returns the file name associated with the GIF object.
PlayCondition	Boolean	R/W	Returns or sets the PlayCondition property. A True condition initiates play. A False condition halts play.
PlayStatus	Boolean	R/W	Returns the PlayStatus property. A True condition indicates the GIF is playing. A false condition indicates the GIF is not playing.
Transparent	Boolean	R/W	Returns or sets the transparent property that indicates that a colour must be shown as transparent when the GIF object is drawn.
TransparentColor	OLE_Color	R/W	Returns or sets the colour (OLE_COLOR code) that must be shown as transparent when the GIF object is drawn (information returned or set is valid only if the Transparent property is set to True).
TransparentColorIndex	Integer	R/W	Returns or sets the colour (indexed color) that must be shown as transparent when the GIF object is drawn (information returned or set is valid only if the Transparent property is set to True).

GIF Object Methods

Method	Description
SetOriginalSize	Sub SetOriginalSize () Resets the object to its original size..

The Group Object

Group Object Properties

An object that represents a group of drawing elements. The properties, methods and events are those as documented in the topic Graphic Objects in General plus the following.

Group Object Properties

Property	As	R / W	Description
Graphics	Graphic	R	Returns the collection of all graphic objects in the group.

The Metafile Object

Properties | Methods

An object that represents a WMF or EMF image. The properties, methods and events are those as documented in the topic Graphic Objects in General plus the following.

Metafile Object Properties

Property	As	R/W	Description
FileName	String	R	Returns the file name associated with the bitmap object.

Metafile Object Methods

Method	Description
SetOriginalSize	Sub SetOriginalSize () Resets the object to its original size..

The Poly Object

Properties

An object that represents a Line, a Polyline, a Polygon or Bézier curve graphic object. The properties, methods and events are those as documented in the topic Graphic Objects in General plus the following.

Poly Object Properties

Property	As	R/W	Description
BottomRightColorAlternate	OLE_Color	R/W	Returns or sets the bottom right colour of the shape.
BottomRightColorAlternateIndex	Integer	R/W	Returns or sets the bottom right colour of the shape in the case of an indexed colour.
BottomRightColorAlternateTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the bottom right colour of the shape is transparent or not..
BottomRightColorBlink	Integer	R/W	Returns or sets the blinking frequency index of the bottom right colour of the shape..
BottomRightColorPrimary	Integer	R/W	Returns or sets the primary bottom right colour of the shape.
BottomRightColorPrimaryIndex	Integer	R/W	Returns or sets the primary bottom right colour of the shape in the case of an indexed colour.
BottomRightColorPrimaryTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the primary bottom right colour of the shape is transparent or not.
NbPoints	Integer	R/W	Returns the number of points that composes the poly.
TopLeftColorAlternate	OLE_Color	R/W	Returns or sets the top Left colour of the shape.
TopLeftColorAlternateIndex	Integer	R/W	Returns or sets the top Left colour of the shape in the case of an indexed colour.
TopLeftColorAlternateTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the top Left colour of the shape is transparent or not..
TopLeftColorBlink	Integer	R/W	Returns or sets the blinking frequency index of the top Left colour of the shape..
TopLeftColorPrimary	Integer	R/W	Returns or sets the primary top Left colour of the shape.
TopLeftColorPrimaryIndex	Integer	R/W	Returns or sets the primary top Left colour of the shape in the case of an indexed colour.

Property	As	R/W	Description
TopLeftColorPrimaryTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the primary top Left colour of the shape is transparent or not.

The Shape Object

Properties

An object that represents a Rectangle, a Rounded Rectangle, an Ellipse, an Arc or a Half-Arc graphic object. The properties, methods and events are those as documented in the topic Graphic Objects in General plus the following.

Shape Object Properties

Property	As	R/W	Description
Appearance	fvAppearance	R/W	Returns or sets the appearance of the shape (regular, shadow, button...). Appearance follows the Appearance rules. Appearance can take one of the following fvAppearance values: FvAppearanceRegular FvAppearanceShadow FvAppearanceButton FvAppearanceColoredButton FvAppearanceRelief fvAppearanceInverseRelief
AppearanceThickness	Integer	R/W	Returns or sets the edge thickness of the shape having the Appearance property not set to Regular. AppearanceThickness follows the Appearance rules.
BottomRightColorAlternate	OLE_Color	R/W	Returns or sets the alternate bottom-right colour of the shape. BottomRightColorAlternate follows the Appearance rules.
BottomRightColorAlternateIndex	Integer	R/W	Returns or sets the alternate bottom-right colour of the shape in the case of an indexed colour. BottomRightColorAlternateIndex follows the Appearance rules.
BottomRightColorAlternateTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the alternate bottom-right colour of the shape is transparent or not. BottomRightColorAlternateTransparent follows the Appearance rules.
BottomRightColorBlink	Integer	R/W	Returns or sets the blinking frequency index of the bottom-right colour of the shape. BottomRightColorBlink follows the Appearance rules.
BottomRightColorPrimary	OLE_Color	R/W	Returns or sets the primary bottom-right colour of the shape. BottomRightColorPrimary follows the Appearance rules.

Property	As	R/W	Description
BottomRightColorPrimaryIndex	Integer	R/W	Returns or sets the primary bottom-right colour of the shape in the case of an indexed colour. BottomRightColorPrimaryIndex follows the Appearance rules.
BottomRightColorPrimaryTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the primary bottom-right colour of the shape is transparent or not. BottomRightColorPrimaryTransparent follows the Appearance rules.
CurvatureX	Integer	R/W	Returns or sets the horizontal curvature of the rounded shape (expressed in pixels).
CurvatureY	Integer	R/W	Returns or sets the horizontal curvature of the rounded shape (expressed in pixels).
OpenedBorder	Boolean	R/W	Returns or sets the boolean that indicates if the drawing of the shape having the 1/2 or 1/4 Ellipse shape style must be opened or closed.
ShapeStyle	FvShapeStyle	R/W	Returns or sets the shape style of the shape (rectangle, rounded rectangle..). ShapeStyle can take one of the following fvShapeStyle values: FvShapeRectangle FvShapeRoundedRectangle FvShapeEllipse FvShapeHalfEllipse fvShapeQuartEllipse
TopLeftColorAlternate	OLE_Color	R/W	Returns or sets the alternate top-left colour of the shape. TopLeftColorAlternate follows the Appearance rules.
TopLeftColorAlternateIndex	Integer	R/W	Returns or sets the alternate top-left colour of the shape in the case of an indexed colour. TopLeftColorAlternateIndex follows the Appearance rules.
TopLeftColorAlternateTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the alternate top-left colour of the shape is transparent or not. TopLeftColorAlternateTransparent follows the Appearance rules.
TopLeftColorBlink	Integer	R/W	Returns or sets the blinking frequency index of the top-left colour of the shape. TopLeftColorBlink follows the Appearance rules.
TopLeftColorPrimary	OLE_Color	R/W	Returns or sets the primary top-left colour of the shape. TopLeftColorPrimary follows the Appearance rules.

Property	As	R/W	Description
TopLeftColorPrimaryIndex	Integer	R/W	Returns or sets the primary top-left colour of the shape in the case of an indexed colour. TopLeftColorPrimaryIndex follows the Appearance rules.
TopLeftColorPrimaryTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the primary top-left colour of the shape is transparent or not. TopLeftColorPrimaryTransparent follows the Appearance rules.

The Symbol Object

Properties | Events

An object that represents a Symbol composed of a number of other graphic objects. Unlike the group object, Symbol instances are stemmed from a library. The properties, methods and events are those as documented in the topic Graphic Objects in General plus the following.

Symbol Object Properties

Property	As	R/W	Description
Angle	Integer	R/W	Returns or sets the "angle" property of the symbol object. This angle, expressed in degree, is used to draw the symbol instance according to the original position and size of its model stored in the symbol library.
BranchContext	String	R	Returns the branch context (mimic branch) of the symbol.
FileName	String	R	Returns the name of the file containing the symbol definition.
KeepOriginal Size	Boolean	R	Returns True if the Original Size tick box is selected, else False
Graphics	Graphics	R	Returns the collection of all graphic objects in the symbol.
LocalBranch	String	R/W	Returns or sets the local branch of the symbol object.

Symbol Object Events

Event	Description
BeforeEvent	Function Symbol_BeforeEvent () As Boolean Occurs before each of the standard events fired from any of the sub-objects within a symbol. If you return True then the next event will not be fired.
AfterEvent	Sub Symbol_AfterEvent() Occurs after each of the standard events fired from any of the sub-objects within a symbol. AfterEvent is always fired event if the other event has not fired because it was disabled by a previous BeforeEvent returning True.

The Text Object

Properties

An object that represents a text object. The properties, methods and events are those as documented in the topic Graphic Objects in General plus the following.

Text Object Properties

Property	As	R/W	Description
Alignment	FvAlignment	R/W	Returns or sets the alignment of the text contained in the text object (right, center or left). Alignment can take one of the following fvAlignment values: FvAlignmentLeft FvAlignmentCenter fvAlignmentRight
Appearance	FvAppearance	R/W	Returns or sets the appearance of the text (regular, shadow, button…). Appearance follows the Appearance rules. Appearance can take one of the following fvAppearance values: FvAppearanceRegular FvAppearanceShadow FvAppearanceButton FvAppearanceColoredButton fvAppearanceRelief fvAppearanceInverseRelief
AppearanceThickness	Integer	R/W	Returns or sets the edge thickness of the text having the Appearance property not set to Regular. AppearanceThickness follows the Appearance rules.
AutoSize	Boolean	R/W	Returns or sets the "AutoSize" property of the text object. This property indicates if the text must (True) or must not (False) be stretched in order to fit in the including rectangle of the text object. Note: The reverse proposal also works - if the including rectangle of the text object must (True) or must not (False) be stretched in order to exactly include the contained text of the text object.
BottomRightColorAlternate	OLE_Color	R/W	Returns or sets the alternate bottom-right colour of the text. BottomRightColorAlternate follows the Appearance rules.

Property	As	R/W	Description
BottomRightColorAlternateIndex	Integer	R/W	Returns or sets the alternate bottom-right colour of the text in the case of an indexed colour. BottomRightColorAlternateIndex follows the Appearance rules.
BottomRightColorAlternateTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the alternate bottom-right colour of the text is transparent or not. BottomRightColorAlternateTransparent follows the Appearance rules.
BottomRightColorBlink	Integer	R/W	Returns or sets the blinking frequency index of the bottom-right colour of the text. BottomRightColorBlink follows the Appearance rules.
BottomRightColorPrimary	OLE_Color	R/W	Returns or sets the primary bottom-right colour of the text. BottomRightColorPrimary follows the Appearance rules.
BottomRightColorPrimaryIndex	Integer	R/W	Returns or sets the primary bottom-right colour of the text in the case of an indexed colour. BottomRightColorPrimaryIndex follows the Appearance rules.
BottomRightColorPrimaryTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the primary bottom-right colour of the text is transparent or not. BottomRightColorPrimaryTransparent follows the Appearance rules.
Font	Font	R/W	Returns or sets the font of the text object
Margin	Integer	R/W	Returns or sets the margin of the text contained in a text object (expressed in pixel).
Multiline	Boolean	R/W	Returns or sets the "Multiline" property of the text object. This property indicates if embedded "\n" sequence in the text object must (True) or must not (False) be considered as a line break.
Text	String	R/W	Returns or sets the text contained in the text object
Top	Integer	R/W	Returns or sets the distance between the top edge of the text and the top edge of its container
TopLeftColorAlternate	OLE_Color	R/W	Returns or sets the alternate top-left colour of the text. TopLeftColorAlternate follows the Appearance rules.

Property	As	R/W	Description
TopLeftColorAlternateIndex	Integer	R/W	Returns or sets the alternate top-left colour of the text in the case of an indexed colour. TopLeftColorAlternateIndex follows the Appearance rules.
TopLeftColorAlternateTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the alternate top-left colour of the text is transparent or not. TopLeftColorAlternateTransparent follows the Appearance rules.
TopLeftColorBlink	Integer	R/W	Returns or sets the blinking frequency index of the top-left colour of the text. TopLeftColorBlink follows the Appearance rules.
TopLeftColorPrimary	OLE_Color	R/W	Returns or sets the primary top-left colour of the text. TopLeftColorPrimary follows the Appearance rules.
TopLeftColorPrimaryIndex	Integer	R/W	Returns or sets the primary top-left colour of the text in the case of an indexed colour. TopLeftColorPrimaryIndex follows the Appearance rules.
TopLeftColorPrimaryTransparent	Boolean	R/W	Returns or sets the transparent property that indicates that the primary top-left colour of the text is transparent or not. TopLeftColorPrimaryTransparent follows the Appearance rules.

The Variables Collection

Properties | Methods | Examples

The collection of all currently subscribed *Variable* objects including all those used in animations in any opened Mimic.

Variables Object Properties

Property	As	R/W	Description
Application	Application	R	Returns the Application object from which Variables object is stemmed.
Count	Long	R	Returns the number of *Variable* objects in the collection.
Parent	Object	R	Returns the parent object of the Variables object (For Variables, the Parent object is always the Project object).

Variables Object Methods

Method	Description	
Add	Function Add(Name As String [,[Type As] fvVariableType]) As Variable Dynamically adds a variable to the Variables collection. If the variable is already subscribed by an animation, it may still be subscribed using this method. In this case it will not be un-subscribed when the animation is closed.	
	Name	A string containing the name of the variable to add.
	Type	Optional argument. Default: fvVariableTypeAny A constant indicating the variable type selected from the following fvVariableType values: FvVariableTypeAlarm fvVariableTypeAny fvVariableTypeBit fvVariableTypeRegister fvVariableTypeText
Item*	Function Item(Index As Variant) As Variable Returns the indicated item (a *Variable* object) in the Variables collection.	
	Index	A Long with a range from 1 to Variables.Count, which represents the position of the variable in the collection, that specifies the item that must be returned. OR A String, which represents the name of the variable, that specifies the item that must be returned.

Method	Description	
Remove	Sub Remove(Name As String) Un-subscribes a variables subscribed using the Add method.	
	Name	A string containing the name of the variable to remove.

Variables Object Examples

```
Dim WithEvents objOPCVar As Variable

Private Sub Text1_Click()
  Variables.Add "Main:Branch1.B1", fvVariableTypeBit
  Set objOPCVar = [Main:Branch1.B1]
  objOPCVar.EnableEvents = True
End Sub

Private Sub objOPCVar_ValueChange()
  MsgBox objOPCVar.Value
End Sub

Private Sub Text11_Click()
  Dim objVar As Object
    For Each objVar In TheseVariables
    MsgBox objVar.Name
    MsgBox objVar.Value
  Next objVar
End Sub
```

The Variable Object

Properties | Events | Examples

The Variable object represents all variable types.

Variable Object Properties

Property	As	R/W	Description
Application	Application	R	Returns the Application object from which Variable object is stemmed from.
CloseLabel		R	Returns the close label of the variable if it is of type fvVariableTypeBit
Description	String	R	Returns the description of the variable.
EnableEvents	Boolean	R/W	Returns or sets the Boolean that indicates if events must (True) or must not (False) be fired for the variable. When this property is disabled, the variable associated scripts are not performed.
ForceOffLabel	String	R	Returns the force off label of the variable if it is of type fvVariableTypeBit.
ForceOnLabel	String	R	Returns the force on label of the variable if it is of type fvVariableTypeBit.
MinimumValue	Double	R	Returns the minimum value of the variable if it is of type fvVariableTypeRegister.
MaximumValue	Double	R	Returns the maximum value of the variable if it is of type fvVariableTypeRegister
Name	String	R	Returns the name of the variable.
OpenLabel	String	R	Returns the open label of the variable if it is of type fvVariableTypeBit
Parent	Variables	R	Returns the parent object of the Variable object (For Variable, the Parent object is always the Variables object).
PreviousStatus	fvVariableStatus	R	Returns the previous status of the variable. See Status for further information.
PreviousValue	Variant	R	Returns the previous Value of the variable.
Quality	Integer	R	Returns the quality of a variable.
SecurityLevel	Long	R	Returns the security level of the bit variable.

Property	As	R / W	Description
Status	fvVariableStatus	R	Returns the status of the variable. Status can take one of the following fvVariableStatus values: FvVariableStatusWaiting FvVariableStatusGood FvVariableStatusBad FvVariableStatusConfigError FvVariableStatusNotConnected Status is automatically set to fvVariableStatusWaiting until the variable subscription is complete. Status is automatically set to fvVariableStatusNotConnected if the server cannot be reached (or has failed).
TimeStamp	Date	R	The time at which the variable was last updated.
Type	fvVariableType	R	Returns the type of the variable as it was returned by the server. Type can take one of the following fvVariableType values: fvVariableTypeAny fvVariableTypeBit fvVariableTypeRegister fvVariableTypeText. fvVariableTypeAny means that the returned type is unknown.
Units	String	E	Returns the Units of a variable if it is of type fvVariableTypeRegister
UserData	Variant	R/W	Returns or sets an expression that stores any extra data needed for your program. The User can use this property to identify objects.
Value*	Variant	R/W	Returns or sets the Value of the variable. If the type returned by the server for this variable is not compatible with the value passed for this property then the value will be discarded.

Variable Object Events

Event	Description	
ValueChange	Sub object_ValueChange() Occurs when the value of the variable has changed.	
	object	Receiver object.

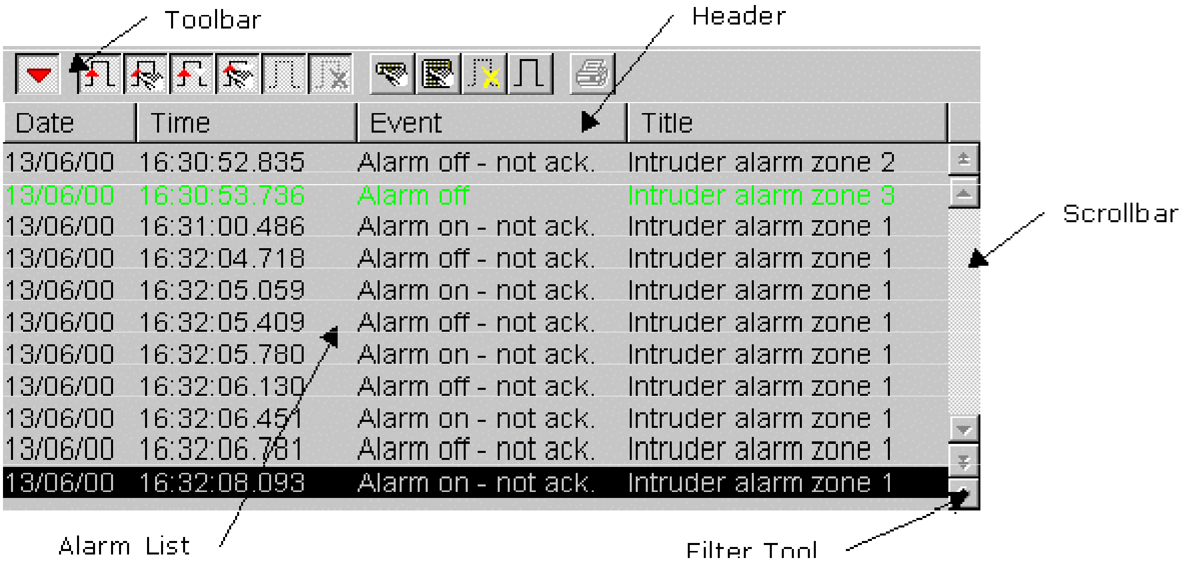
Variable Object Examples

```
Private Sub objOPCVar_ValueChange()
  '-------- Records the value of the variable
  Const BufferSize = 50
  Const LogFile = "HistoricLog.txt"
  Static Buffer(BufferSize) As String
  Static BPointer As Integer
'-------- Record in a string array
'-------- Note the use of the _ continuation character
  Buffer(BPointer) = Format(Date, " dd/mm/yyyy") & "," & _
  Format(Time, "hh:nn:ss") & "," & objOPCVar.Value
'-------- If string array full then flush to file and start again
  If (BPointer < BufferSize) Then
    BPointer = BPointer + 1
  Else
'    Print Date & " " & Time & " Sending buffer To file"
    Open LogFile For Append As #1
    For BPointer = 1 To BufferSize
      Print #1, Buffer(BPointer)
    Next BPointer
    Close #1
    BPointer = 1
  End If


Private Sub TextGetVarInfo_Click()
'-------- Gets and display information about a server variable
'-------- NOTE the variable must be subscribed
  Dim WorkVariable As Object
  Dim VarName As String
  VarName = TextInput.Text    ' Name of variable to test
  Set WorkVariable = TheseVariables.Item(VarName)
'-------- Get properties common to all variables
  Text102.Text = "Description: " & WorkVariable.Description
  Select Case WorkVariable.Status
    Case fvVariableStatusWaiting
      Text103.Text = "Status: Waiting"
    Case fvVariableStatusGood
      Text103.Text = "Status: Good"
    Case fvVariableStatusBad
      Text103.Text = "Status: Bad"
    Case fvVariableStatusConfigError
      Text103.Text = "Status: Configuration error"
    Case fvVariableStatusNotConnected
      Text103.Text = "Status: Not Connected"
    Case Else
      Text103.Text = "Status: Unknown"
  End Select
'-------- Get properties specific to variable types
  Select Case WorkVariable.Type
    Case fvVariableTypeBit
      Text104.Text = "Bit"
      If (WorkVariable.Value = 0) Then
        Text105.Text = "Value: " & WorkVariable.CloseLabel
      Else
        Text105.Text = "Value: " & WorkVariable.OpenLabel
      End If
    Case fvVariableTypeRegister
      Text104.Text = "Type: Register"
      Text105.Text = "Value: " & WorkVariable.Value & " " & WorkVariable.Units
    Case fvVariableTypeText
      Text104.Text = "Text"
      Text105.Text = "Value: " & WorkVariable.Value
    Case Else
      Text104.Text = "Unknown variable type: " & WorkVariable.Type
  End Select
End Sub
```

Introduction

Summary of Features

The Alarm Viewer is an ActiveX control that displays a list of the current state of alarms. Any number of Alarm Viewers may be embedded in a mimic, the only limitation being the amount of space available in the mimic.



Date	Time	Event	Title
13/06/00	16:30:52.835	Alarm off - not ack.	Intruder alarm zone 2
13/06/00	16:30:53.736	Alarm off	Intruder alarm zone 3
13/06/00	16:31:00.486	Alarm on - not ack.	Intruder alarm zone 1
13/06/00	16:32:04.718	Alarm off - not ack.	Intruder alarm zone 1
13/06/00	16:32:05.059	Alarm on - not ack.	Intruder alarm zone 1
13/06/00	16:32:05.409	Alarm off - not ack.	Intruder alarm zone 1
13/06/00	16:32:05.780	Alarm on - not ack.	Intruder alarm zone 1
13/06/00	16:32:06.130	Alarm off - not ack.	Intruder alarm zone 1
13/06/00	16:32:06.451	Alarm on - not ack.	Intruder alarm zone 1
13/06/00	16:32:06.781	Alarm off - not ack.	Intruder alarm zone 1
13/06/00	16:32:08.093	Alarm on - not ack.	Intruder alarm zone 1

Summary of Features

- Alarm Viewers may be embedded in any mimic.
- Unlimited number of Alarm Viewers in each project.
- Size of alarm list configurable from one line to the maximum allowed by the screen resolution.
- The alarms to be viewed may optionally be selected by one or more filters, by level and by status (for example, all alarms that are on and not acknowledged.)
- The format of the alarm text is fully configurable.
- A toolbar to select which alarm transitions are displayed, to acknowledge alarms, to inhibit alarms and to change the mode of operation.
- Optional scrollbars.

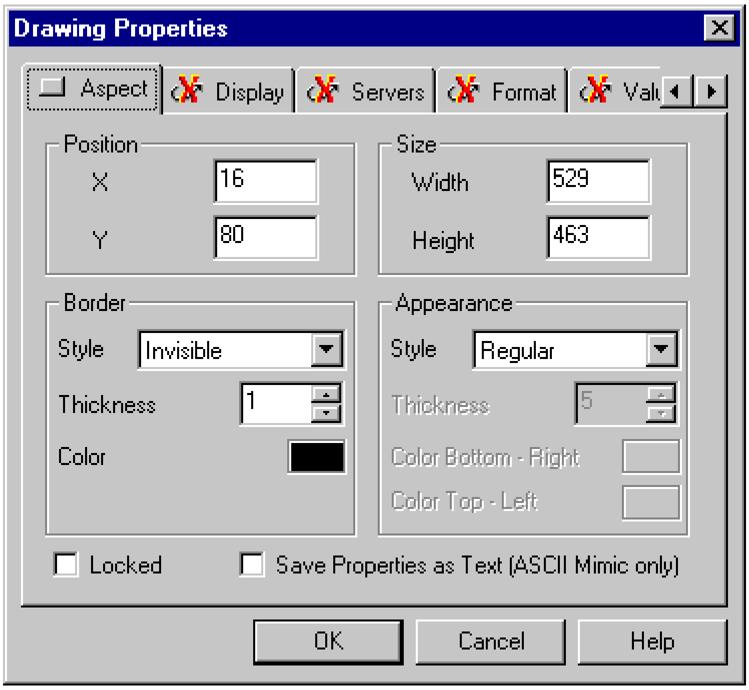
Creating an Alarm Viewer

Configuring the General Appearance | Configuring the Alarm List | Selecting Which Alarms Appear in the Viewer | Configuring the Start-up Behaviour | Selecting What the User can Change at Run-time

The Alarm Viewer is extremely flexible in its appearance and operation, with many configurable properties.

How to create an Alarm Viewer

1. Open the mimic in which the Alarm Viewer will be inserted and select Edit mode.
2. Select the Alarm Viewer Control tool from either the toolbar or the Insert menu.
3. An Alarm Viewer will be pasted into the mimic.
4. Drag the handles surrounding the Alarm Viewer to size and position it.
5. Double click on the Alarm Viewer to display its properties.
6. Use the arrow buttons ◄► in the top right hand corner of the properties box to select the configuration tabs, including those that are not currently visible.



Drawing Properties

Aspect | Display | Servers | Format | Valu

Position
X: 16
Y: 80

Size
Width: 529
Height: 463

Border
Style: Invisible
Thickness: 1
Color: ■

Appearance
Style: Regular
Thickness: 5
Color Bottom - Right
Color Top - Left

☐ Locked   ☐ Save Properties as Text (ASCII Mimic only)

OK | Cancel | Help

What are each of the tabs for?

- *Aspect* - Together with the *Display* tab defines the general appearance of an Alarm Viewer.
- *Display* - Together with the *Aspect* tab defines the general appearance of an Alarm Viewer.
- *Servers* - Selects the Alarm Server and filters to which the Alarm Viewer is connected. With Vijeo Look there is only one server which is automatically selected.
- *Format* - Configures the format of the alarm line.
- *Values* - Selects which of the alarm states appear in the alarm list.
- *Colours* - Selects the color in which each alarm level and state appears.
- *Rights* - Selects the operating features to which the User has access at run-time.
- *Parameters* - Miscellaneous operating parameters.
- *Levels* - Selects which alarm levels will be displayed.

Configuring the General Appearance

You configure the general appearance of an Alarm Viewer using the *Aspect* and *Display* tabs.

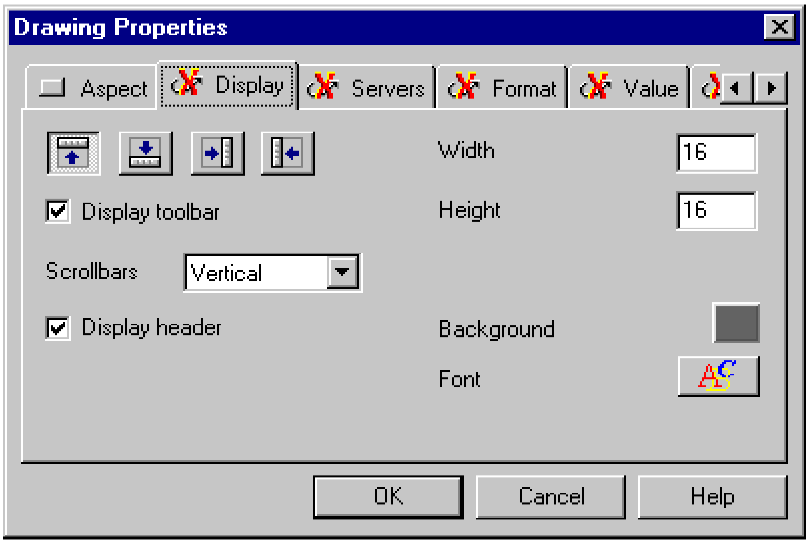
How to Configure the Position and Borders

In the *Aspect* tab, directly enter X-Y co-ordinates for the top left corner of the Alarm Viewer, and specify its Width and Height in pixels. For a simple line around the Viewer's edges, set the Style, Thickness and Colour in *Border*. You can choose more sophisticated surrounds in a similar manner in *Appearance*.

How to Configure the Alarm Viewer display

The Alarm Viewer can include a toolbar, scrollbars and a header. These may be individually positioned and enabled or disabled using the settings in the *Display* tab.

- *Display toolbar* - If you tick the checkbox, the toolbar will be shown at run-time on whichever side of the Alarm Viewer you then select .
- *Scrollbars.* – You can choose whether vertical and/or horizontal scrollbars are required and specify their width in pixels.
- *Display header* - You can show or suppress the header, and if shown, specify its width in pixels.
- *Background* - Click the coloured button to choose a background colour from the Color Palette.
- *Font* - The font used in the alarm list may be selected from any font available to the host operating system. Select the button to open the font dialog box and choose a font for display of text.



☑ Display toolbar

Scrollbars: Vertical

☑ Display header

Width: 16

Height: 16

Background

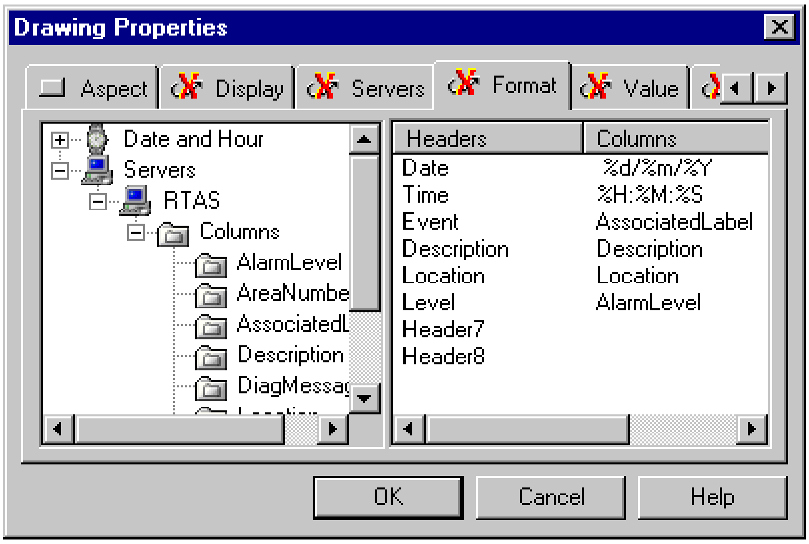
Font

If you disable the Header you will not be able to display a description for each of the columns.

Configuring the Alarm Line Format

The active part of the Alarm Viewer is known as the alarm list. This contains a number of lines of text each displaying the status of one alarm. The number of lines that are visible depends on the size of the Alarm Viewer and the font chosen. The information that is displayed in the list is arranged in up to eight configurable columns. The Alarm List is configured using the *Format* tab.

The *Format* tab is divided into two panes. The left pane contains a list of items that can be displayed. The right pane is used to enter the title for each column and select what is displayed in them. You configure what is displayed in each column by clicking on an item in the left pane and dragging it to a column in the right pane.



Headers	Columns
Date	%d/%m/%Y
Time	%H:%M:%S
Event	AssociatedLabel
Description	Description
Location	Location
Level	AlarmLevel
Header7	
Header8	

What can you display?

- The date and time. The structure of the date and time is made up of individual format elements. A full list of what you can display is contained in the Date and Time folder.
- The alarm level.
- The Associated Label. A string representing the current status of the alarm. For example: On and Not Acknowledged.
- The variable description. A string supplied either by the OPC server or entered in the variable configuration
- The variable name.
- Information from the Diagnostic Buffer, AreaNumber, DiagMessage, ProgAddress and StatusFile.
- Any Extended Properties that have been configured in the Configuration Explorer.

How to configure a column name

The column name is known as the header.

1. Click on the text representing the column name under the list of headers. For example Header7.
2. Enter the new name for the column.

How to configure a column

1. In the left pane select the element you want to be displayed
2. Click and drag it to the right pane and drop it on the name of the column where it is to appear.

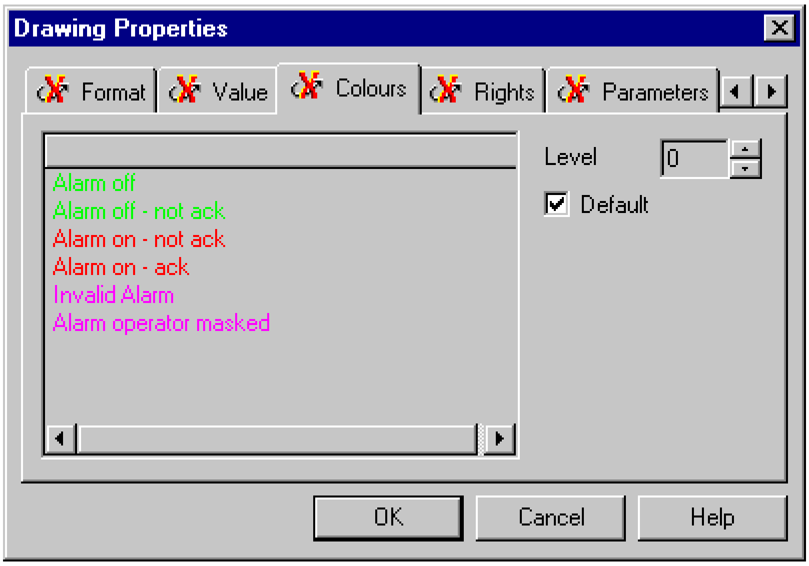
Selecting the Alarm Colours

The colours in which alarms are displayed are normally defined by the colour preferences selected from the HMI's main menu (Tools.Preference.Color). However these can be replaced using the *Colours* tab.

How to use a colour for an alarm other than that defined by the colour preferences

1. Select the *Colours* tab.

2. Select the alarm level.
3. Un-tick the *Default* box.
4. Double-click on the alarm state to be changed. The Color palette opens.
5. Click on the required colour in the palette. It is applied to the selected alarm state.



Alarm off
Alarm off - not ack
Alarm on - not ack
Alarm on - ack
Invalid Alarm
Alarm operator masked

Level 0

☑ Default

Selecting Which Alarms Appear in the Viewer

You can select the alarms that appear in the viewer according to:

- Alarm level.
- Alarm state (for example display only alarms On and Not Ack).
- One or more Filters.

Or any combination of the above.

Selecting alarms according to level

You may choose which alarms are eligible for display, by level, through the *Levels* tab.

- *From level* - Minimum alarm level to display.
- *To level* - Maximum alarm level to display.

Selecting alarms according to state

You may choose which alarms are eligible for display, by alarm state, using the *Event* tab.

- *Alarm on, not ack* - Display alarms that are on but have not yet been acknowledged by the User.
- *Alarm on, ack* - Display alarms that are on and have been acknowledged by the User.
- *Alarm off, not ack* - Display alarms that have gone on and off before being acknowledged by the User.
- *Alarm off* - Display alarms that are off.
- *Alarm invalid* - Display alarms that are invalid for any reason.
- *Alarm User Masked* - Display alarms that are unavailable because they have been masked by a User (from the Alarm Viewer).

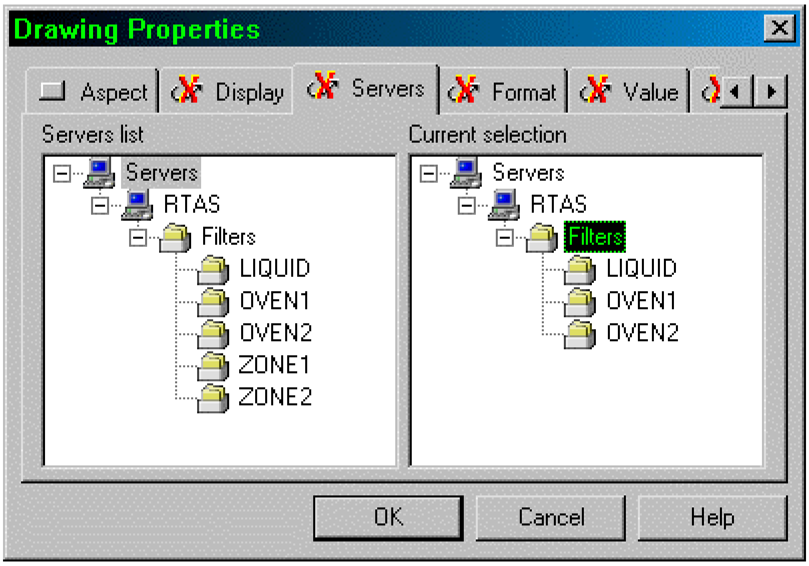
An alarm may be configured to be on when it is either in the one or zero state. See the topic RTDS for further information.

Selecting alarms using a filter

You can use one or more filters to select the alarms that will appear in the Alarm Viewer. Any filters used must have been previously configured using the Filters tab in the General

Properties folder of the Configuration Explorer. Filters may also be added or removed by the User at run-time depending on the selected run-time rights.

1. Select the Servers tab - In the left pane any available filters will appear under the folder *Server.Alarm.Filters*.
2. To apply a particular filter, click and drag it to the right hand pane and drop it under the Filter folder.
3. Repeat step 2 if you want to add more than one filter.



Configuring the Start up Behaviour

The Start-up behaviour is configured using the *Parameters* tab of the Properties dialog box.

- *Real-time Buffer* - The maximum number of lines that may be recorded in the display buffer when the Alarm Viewer is in on-line (Stream) mode.
- *On line* - Start the Alarm Viewer in on-line (Stream) mode.
- *List* - Start the Alarm Viewer in list mode.

Selecting What the User can Change at Run-time

What the User can change at run-time is configured using the *Rights* tab of the Properties dialog box.

- *Acknowledgement* - If selected the User may acknowledge alarms.
- *Mask/unmask* - If selected the User may mask and unmask alarms.
- *Filtering* - If selected the User may change the filters.
- *Multi-selection* - If selected the User may select more than one alarm at a time in the alarm list.
- *Size of Columns* - If selected, the User can adjust the width of columns at run-time by clicking and dragging with the mouse on the edge of the header.
- *Toolbar Position* - If selected the User can change the toolbar position at run-time.
- *Toolbar Modification* - If selected, the User can change the toolbar configuration at run-time.

If you want to change the column width or modify the toolbar, but do not want the User to do so, enable the properties and make the changes at run-time, then disable the properties again. The changes to the display will be saved.

The Alarm Viewer at Run-time

Using the Scroll Bar | Using the Toolbar | Customising the Toolbar | Changing the Filters | Operation in List Mode | Operation in Stream Mode

An Alarm Viewer shows, in a scrollable list, the current (real-time) status of all alarms that are eligible for display. Filters are used to determine which alarms are eligible. If no filters are applied, all alarms are eligible. Otherwise an Alarm Viewer shows only those alarms that meet the criteria of the filters assigned to it. A filter can operate on the Extended Properties, alarm level and alarm status. A filter is initially configured during the design stage and may also be changed at run-time if the configuration permits the User to make such changes.

Alarms are special bit variables that have the alarm property enabled. An alarm may be generated from either the 1 or 0 state of the bit and is given a level of 0 to 29. A bit with alarm attributes behaves differently to a standard bit in that, instead of having three states (true, false and invalid), it has five.

- Off.
- On and not acknowledged.
- On and acknowledged.
- Off and not acknowledged.
- Invalid (A general term for when an alarm is not operating normally for a variety of reasons).

The colours in which alarms appear are defined by the HMI's colour preferences, or at the design stage, by the *Colours* tab of the Alarm Viewer's properties box. Only the foreground (font) colour is used, the background colour being provided by the Alarm Viewer.

Using the Scroll Bar

The scroll bar is used to select which data in the display buffer are displayed on the list.

Tool	Action
⇞	Scroll the display back in time by one page.
▲	Scroll the display back in time by one line.
□	Scroll the display by clicking and dragging with the mouse.
▼	Scroll the display forward in time by one line.
⇟	Scroll the display forward in time by one page.
◆	Display the Alarm Viewer filter box.

Using the Toolbar

The toolbar allows the User to filter the alarm list, acknowledge and inhibit alarms. It varies so as to display the buttons that are relevant to any line or lines that you select in the alarm list.

Click	To
▼ ⇊	Switch between List and Stream modes. When the button is 'down' the display is in List mode.
	Display alarms that are On and Not Acknowledged.
	Display alarms that are On and Acknowledged.
	Display alarms that are Off and Not Acknowledged.
	Display alarms that are Off.
	Display alarms with an invalid status.
	Display alarms that have been masked by the operator.
	Acknowledge selected alarm(s).

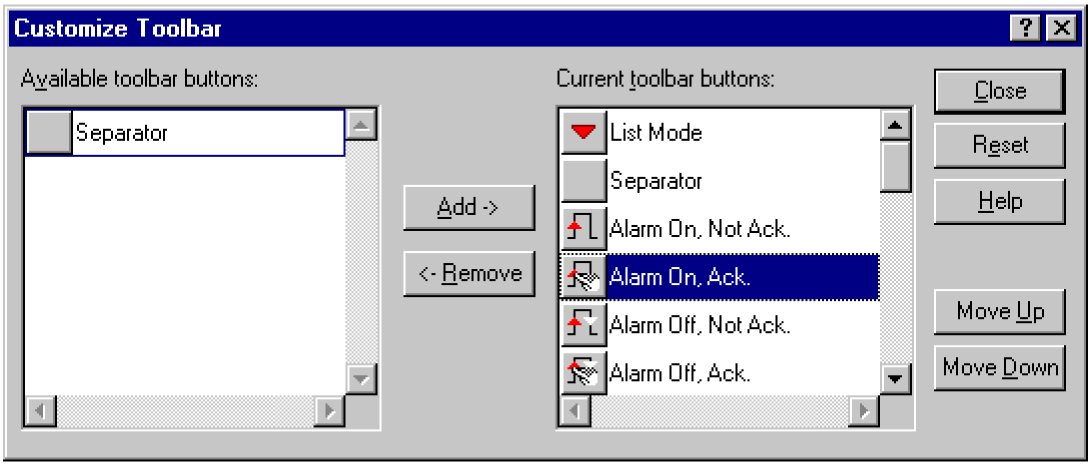
Acknowledge all displayed alarms.

Mask (inhibit) selected alarms.

Un-mask (enable) selected alarms.

Customising the Toolbar

If the Toolbar Modification property was selected when the Alarm Viewer was configured, at run time the User can select the position of the toolbar (top, bottom, left or right) and the tools that appear on it. (This is the only way in which the toolbar can be changed, so if you want to modify the tools that appear, but not let a User do so at run time, you must first enable toolbar modification in Edit mode, make the changes in Run mode, then in Edit mode disable toolbar modification.) To display the Toolbar Modification box, click on the toolbar with the right mouse button and select the Customize Toolbar tool.



Customize Toolbar

Available toolbar buttons:

- Separator

Current toolbar buttons:

- List Mode
- Separator
- Alarm On, Not Ack.
- Alarm On, Ack.
- Alarm Off, Not Ack.
- Alarm Off, Ack.

How to Add a Tool to the Toolbar

1. Select the position in which the tool will be added by clicking on the tool that will immediately follow it in the Current toolbar buttons list.
2. Select the tool to be added in the Available toolbar buttons list.
3. Select the Add button.

With the exception of the Separator, as tools are added they are removed from the Available toolbar button list. The Separator can be inserted repeatedly.

How to Remove a Tool from the Toolbar

1. Select the tool to be removed in the Current toolbar buttons list.
2. Select the Remove button.

How to Move a Tool's Position Within the Toolbar

1. Select the tool to be moved in the Current toolbar buttons list.
2. Select the Move up or Move down button once or more times as required.

Changing the Filters

The *Alarms Filters and Events* dialog box is displayed from the small command button below the vertical scroll bar.

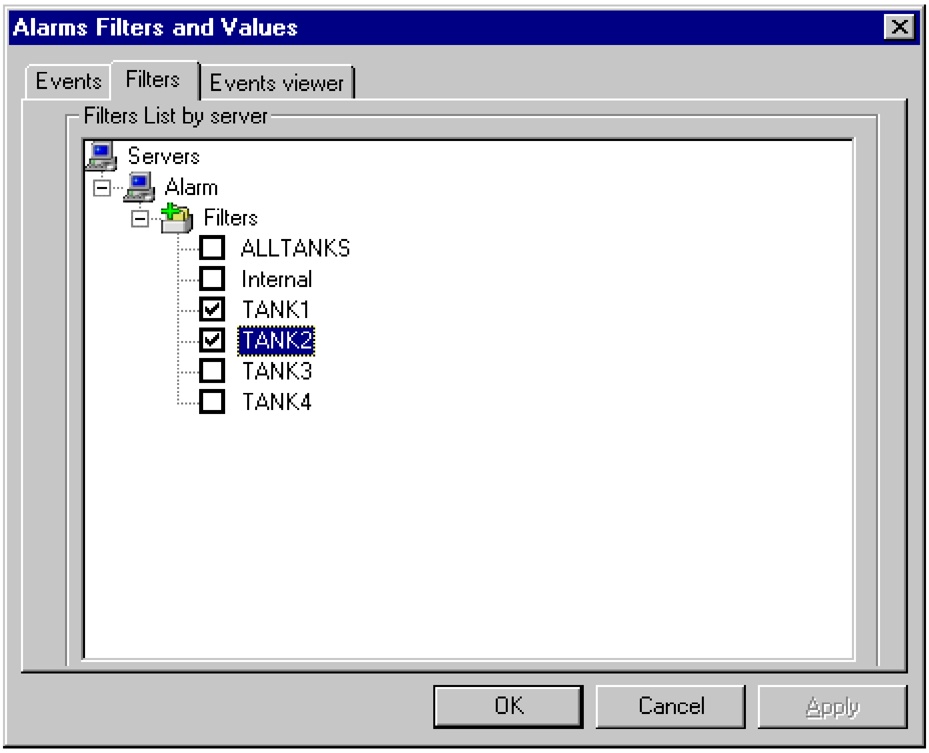
How to Select Alarms for Viewing

1. Select the Filters tab and open the Filters folder. A list of available Filters will be displayed.
2. Tick the checkbox of each filter that is to be applied. (To de-select a filter, click on its checkbox to remove the tick.)
3. You may set the cumulative effect of multiple filters as either exclusive (AND) or inclusive (OR). Right-click on *Filters* then choose either And condition or Or condition. The AND

condition includes alarms in the Viewer only if they satisfy all filters assigned to it. The OR condition includes alarms in the Viewer if they satisfy any filter assigned to it.

4. When the required filters are selected, select OK to close the dialog box.

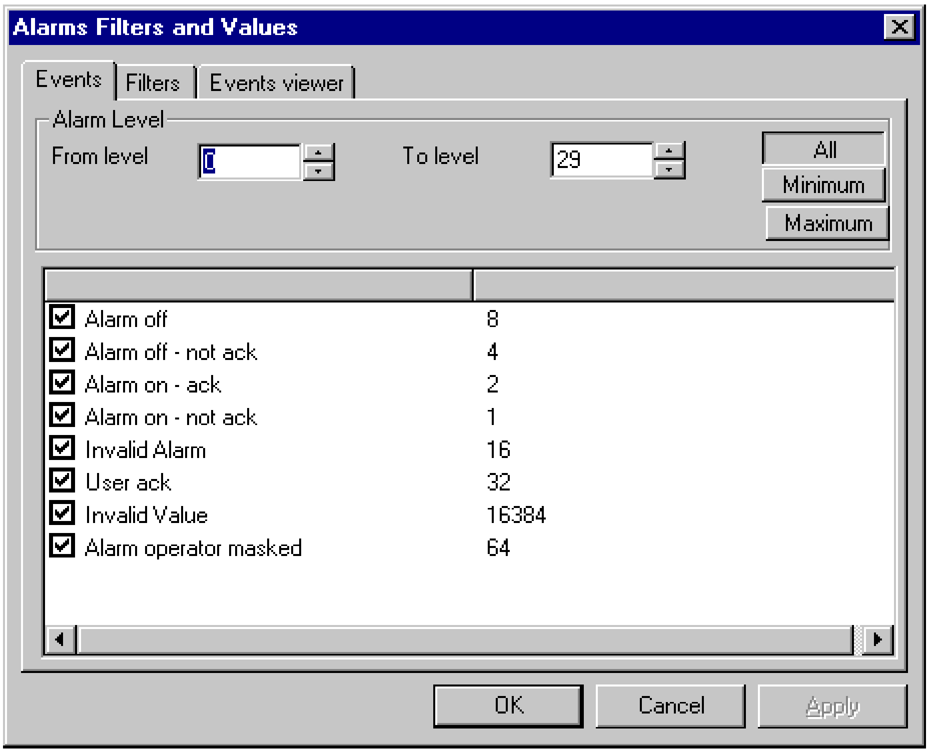
Changing the choice of filters at run-time is only a temporary operation; when the window containing the Alarm Viewer is closed the changes are lost unless the windows is saved either from VBA or configuration menu.



How to Select Alarms by Alarm State

You may choose which alarms are eligible for display, by alarm state, through the *Values* tab.

- *Alarm on, not ack* - Display alarms that are on but have not yet been acknowledged by the User.
- *Alarm on, ack* - Display alarms that are on and have been acknowledged by the User.
- *Alarm off, not ack* - Display alarms that have been on and gone off before being acknowledged by the User.
- *Alarm off* - Display alarms that are off.
- *Alarm invalid* - Display alarms that are invalid for any reason.
- *Alarm User Masked* - Display alarms that have been masked by a User (from the Alarm Viewer).



Alarm Level

From level 0

To level 29

☑ Alarm off	8
☑ Alarm off - not ack	4
☑ Alarm on - ack	2
☑ Alarm on - not ack	1
☑ Invalid Alarm	16
☑ User ack	32
☑ Invalid Value	16384
☑ Alarm operator masked	64

Operation in List Mode

You can switch an Alarm Viewer between two modes of use: List mode and Stream mode. List mode ▼ is the normal mode of operation of an Alarm Viewer. The mode tool button is in the 'down' position.

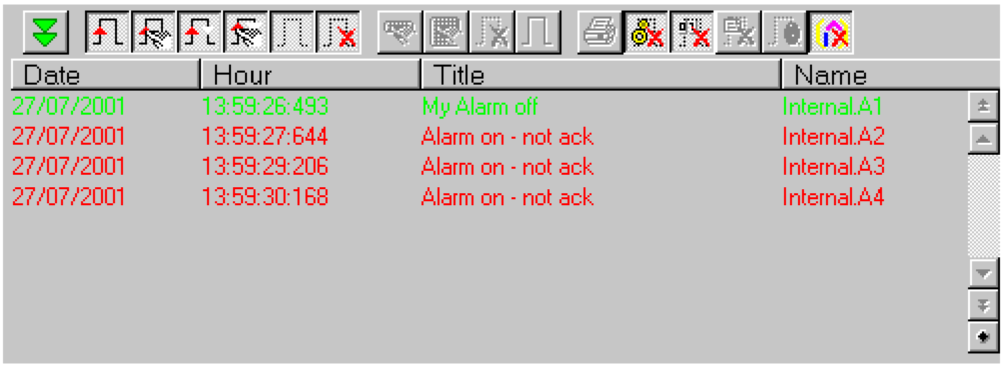
When the display is opened the eligible alarms are sorted into chronological order with the oldest at the top. If there are more alarms than can be viewed in the visible area then the scroll bar can be used for navigating through the list.

How to Select Alarms

Alarms are selected by clicking on the relevant line in the display list, and de-selected by clicking on a line again. Several alarms may be selected at a time by holding down the Shift button and then clicking on another line. A selected alarm is indicated by displaying it in reverse video. Selected alarms may be acknowledged, masked and un-masked using the tools on the toolbar.

Operation in Stream Mode

Stream mode operates when the mode button on the toolbar is shown in the 'up' position. Stream mode provides the User with a real-time list of alarm transitions in the order in which they occur. The alarm transitions are stored in the display buffer, which may be navigated using the scroll bar. The action of selecting an alarm automatically switches the Alarm Viewer to List mode.



Date	Hour	Title	Name
27/07/2001	13:59:26:493	My Alarm off	Internal.A1
27/07/2001	13:59:27:644	Alarm on - not ack	Internal.A2
27/07/2001	13:59:29:206	Alarm on - not ack	Internal.A3
27/07/2001	13:59:30:168	Alarm on - not ack	Internal.A4

Log Viewer Introduction

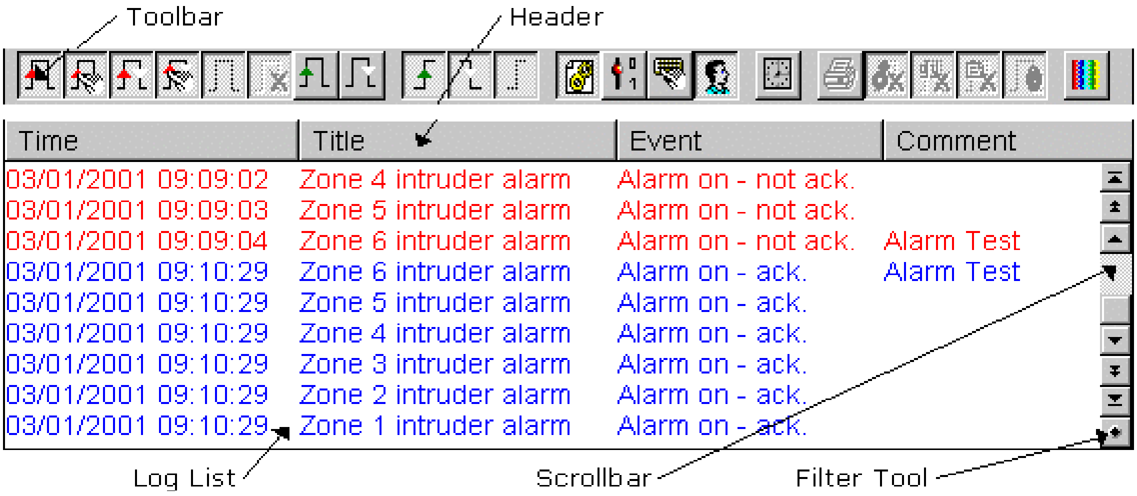
Summary of Features

The Log Viewer is an ActiveX control that displays event data that has been recorded in a database by the HDS. Any number of Log Viewers may be embedded in a mimic, the only limitation being the amount of space available in the mimic.

A Log Viewer is attached to one or more Log Lists. If you attach more than one Log List, at run-time you can select the one that is active. At run-time you can also select one or more Filters to display a subset of the events. Using a combination of the Log Lists and Filters you can easily create historic alarm lists, plant event logs, operator action lists etc.

For more information on how Log Lists are configured see the topic Historic Data Server.

For more information on how Filters are configured, see the topic General Properties.



Time	Title	Event	Comment
03/01/2001 09:09:02	Zone 4 intruder alarm	Alarm on - not ack.	
03/01/2001 09:09:03	Zone 5 intruder alarm	Alarm on - not ack.	
03/01/2001 09:09:04	Zone 6 intruder alarm	Alarm on - not ack.	Alarm Test
03/01/2001 09:10:29	Zone 6 intruder alarm	Alarm on - ack.	Alarm Test
03/01/2001 09:10:29	Zone 5 intruder alarm	Alarm on - ack.	
03/01/2001 09:10:29	Zone 4 intruder alarm	Alarm on - ack.	
03/01/2001 09:10:29	Zone 3 intruder alarm	Alarm on - ack.	
03/01/2001 09:10:29	Zone 2 intruder alarm	Alarm on - ack.	
03/01/2001 09:10:29	Zone 1 intruder alarm	Alarm on - ack.	

Summary of Features

- Log Viewers may be embedded in any mimic.
- Unlimited number of Log Viewers in each project.
- Size of Log Viewer configurable from one line to maximum allowed by screen resolution.
- Each Log Viewer may be attached to one or more Log Lists
- Displayed events may be selected using one or more Filters.
- Displayed events include:

Alarm changes of state.
Bit changes of state.

- Optional toolbar and scrollbars.
- The format of entries in the Viewer list is fully configurable.

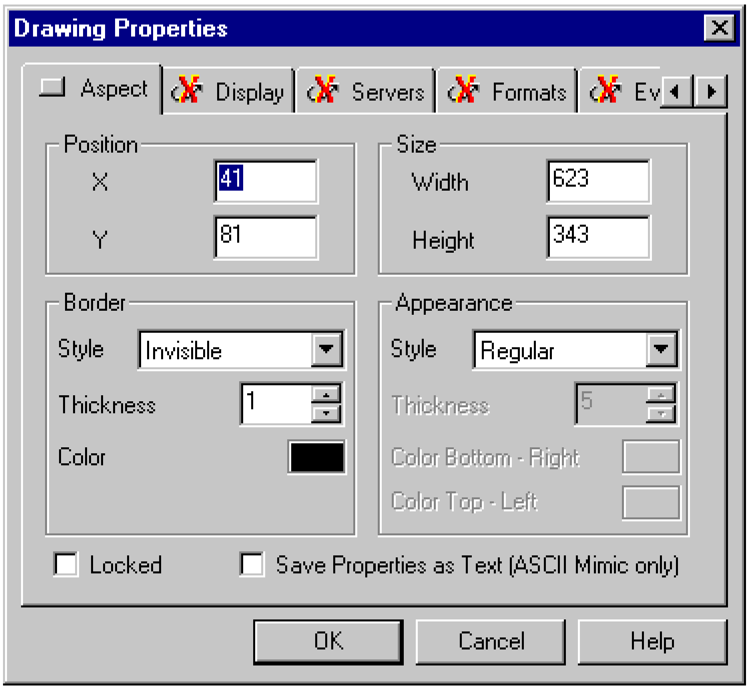
Creating a Log Viewer

General Appearance and Operation | Configuring the Log List | How to Configure the Colours | Selecting What Appears in the Viewer | Selecting What the User can Change at Run-time

The Log Viewer is extremely flexible in its appearance and operation, with many configurable properties.

How to create a Log Viewer

1. Open the mimic in which the Log Viewer will be inserted and select Edit mode.
2. Select the Log Viewer tool from either the toolbar or the Insert menu.
3. A Log Viewer will be pasted into the mimic.
4. Drag the handles surrounding the Log Viewer to size and position it.
5. Double click on the Log Viewer to display its properties.
6. Use the arrow buttons in the top right hand corner of the properties box to select the configuration tabs, including those that are not currently visible.



Position
X 41
Y 81

Size
Width 623
Height 343

Border
Style Invisible
Thickness 1
Color ■

Appearance
Style Regular
Thickness 5
Color Bottom - Right
Color Top - Left

□ Locked
□ Save Properties as Text (ASCII Mimic only)

What are each of the tabs for?

- *Aspect* - Together with the *Display* tab defines the general appearance of a Log Viewer.
- *Display* - Together with the *Aspect* tab defines the general appearance of a Log Viewer.
- *Servers* - Selects the HDS and filters to which the Log Viewer is connected. With Vijeo Look there is only one HDS which is automatically selected.
- *Formats* - Configures the format of the text on the log list.
- *Events* - Selects which of the events appear in the log list.
- *Colours* - Selects the color in which each event appears.
- *Rights* - Selects the operating features to which the User has access at run-time.
- *Parameters* - Miscellaneous operating parameters.

Configuring the General Appearance and Operation

You configure the general appearance and operation of a Log Viewer using the *Aspect* and *Display* tabs.

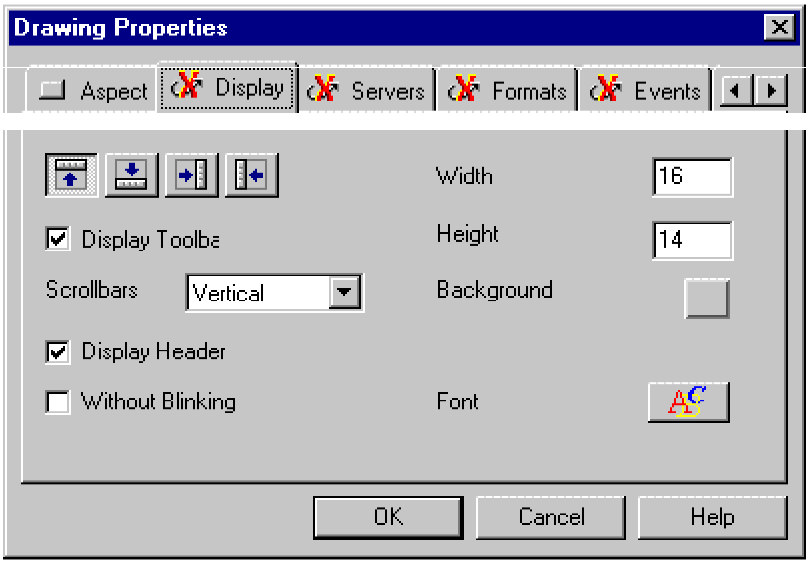
How to Configure the Position and Borders

The position and general appearance of the Log Viewer may changed using the properties found in the *Aspect* tab. You can directly enter the X-Y co-ordinates of the top left corner and the width and height in pixels. For a simple line around the Viewer's edges, set the Style, Thickness and Colour in *Border*. You can choose other surrounds (for example relief) in a similar manner using *Appearance*.

How to Configure the Log Viewer display

The Log Viewer can include a toolbar, scrollbars and a header. These may be individually positioned and enabled or disabled using the settings in the *Display* tab.

- *Display toolbar* - If you tick the checkbox, the toolbar will be shown at run-time along whichever side of the Log Viewer you then select .
- *Scrollbars.* – You can choose whether vertical and/or horizontal scrollbars are required, and specify their width in pixels.
- *Display header* - You can show or suppress the header, and if shown, specify its height in pixels.
- *Background* - Click on the button to choose a background colour from the Color Palette.
- *Without Blinking* - Tick the check box to suppress blinking of text. The text is displayed in its primary color only.
- *Font* - The font used in the log list may be selected from any font available to the host operating system. Select the button to open the *Font* dialog box and choose a font for display of text.
- *Display Buffer* – The maximum number of lines that may be recorded in the display buffer. (Parameters tab)



Drawing Properties

☑ Display Toolbe

Scrollbars: Vertical

☑ Display Header

☐ Without Blinking

Width: 16

Height: 14

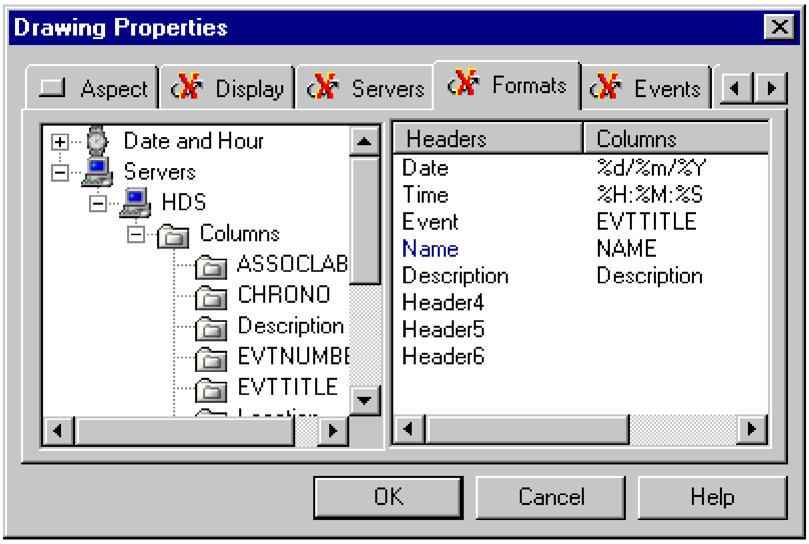
Background

Font

Configuring the Log Line Format

The active part of the Log Viewer is known as the log list. This contains a number of lines of text each displaying the status of an event. The number of lines that are visible depends on the size of the Log Viewer and the font chosen. The information that is displayed in the list is arranged in up to eight configurable columns. The Log List is configured using the *Format* tab.

The *Format* tab is divided into two panes. The left pane contains a list of items that can be displayed. The right pane is used to enter the title for each column and select what is displayed in them. You configure what is displayed in each column by clicking on an item in the left pane and dragging it to a column in the right pane.



Headers	Columns
Date	%d/%m/%Y
Time	%H:%M:%S
Event	EVTTITLE
Name	NAME
Description	Description
Header4	
Header5	
Header6	

What can you display?

- The date and time. The structure of the date and time is made up of individual format elements. A full list of what you can display is contained in the Date and Time folder.
- Any columns that have been configured in the selected Log List. See the topic Selecting What Appears in the Viewer.

How to configure a column name

The column name is known as the header.

1. Click on the text representing the column name under the list of headers. For example Header7.
2. Enter the new name for the column.

How to configure a column

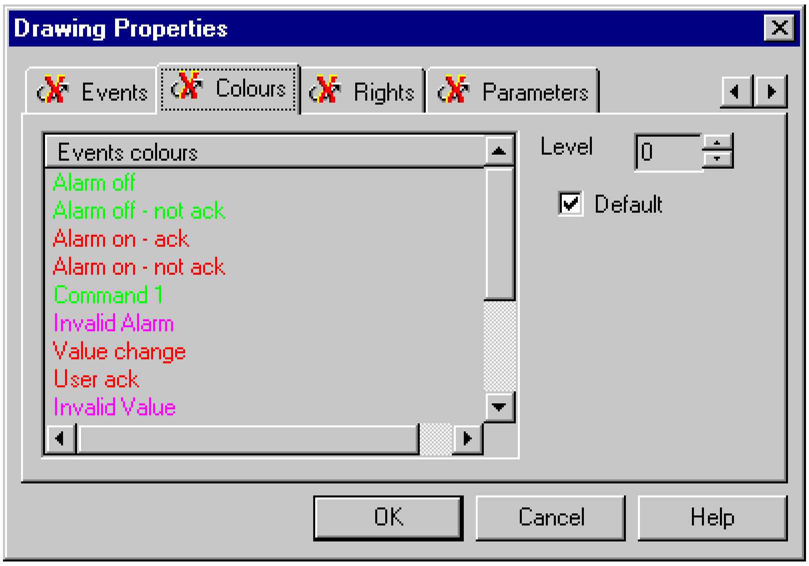
1. In the left pane select the column you want to be displayed
2. Click and drag it to the right pane and drop it on the name of the header under which it will be displayed.

How to Configure the Log Viewer Colours

The colours in which events are displayed are normally defined by the colour preferences selected from the HMI's main menu (Tools.Preference.Color). However these can be replaced using the *Colors* tab.

How to use a colour for an Event other than that defined by the colour preferences

1. Select the *Colors* tab.
2. Un-tick the *Default* box.
3. Double-click on the event to be changed. The Color palette opens.
4. Click on the required colour in the palette. It is applied to the selected event.



Events colours
Alarm off
Alarm off - not ack
Alarm on - ack
Alarm on - not ack
Command 1
Invalid Alarm
Value change
User ack
Invalid Value

Level 0

☑ Default

Selecting What Appears in the Viewer

You can select the events that appear in the viewer according to:

- The Log List.
- One or more filters.
- The event type.
- The alarm level

or a combination of the above.

How to select events using a Log List

The Log List is the primary factor in deciding what is displayed in Log Viewer. If you attach more than one Log List then, at run-time, the User can select which Log List is active. To change the selected Log List or add another Log List do the following.

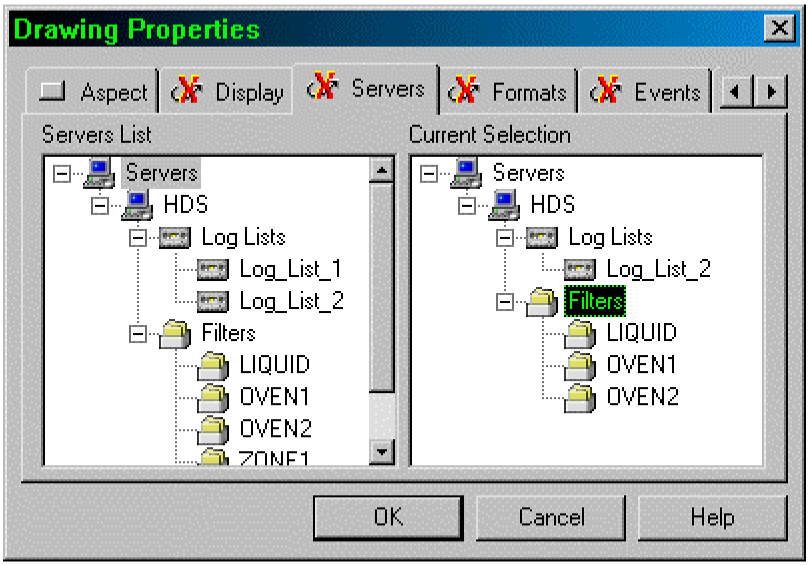
1. Select the Servers tab - In the left pane, available filters will appear under the folder Servers.HDS.Log Lists.
2. To select a Log List, click and drag it from the left to the right pane and drop it under the Log Lists folder.
3. Repeat step 2 if you want to add more than one Log List.

To remove a Log List from the Log Viewer, right click on its name in right pane and select the delete command. Note that this does not delete the Log List, it just removes its reference from the viewer.

How to select events using a Filter

You can attach one or more Filters to the Log Viewer. At run-time the Filters may be activated and de-activated by the User to filter the events that are displayed.

1. Select the Servers tab - In the left pane any available filters will appear under the folder Servers.HDS.Filters.
2. To apply a particular filter, click and drag it to the right hand pane and drop it under the Filters folder.
3. Repeat step 2 if you want to add more than one filter.



How to select events according to type

You can also use the Events tab to select which event types are displayed. You can also select alarm events by alarm level. Selections made in the Event tab are applied to the entire Log Viewer. Remember that event types are also selected in the Log Lists.

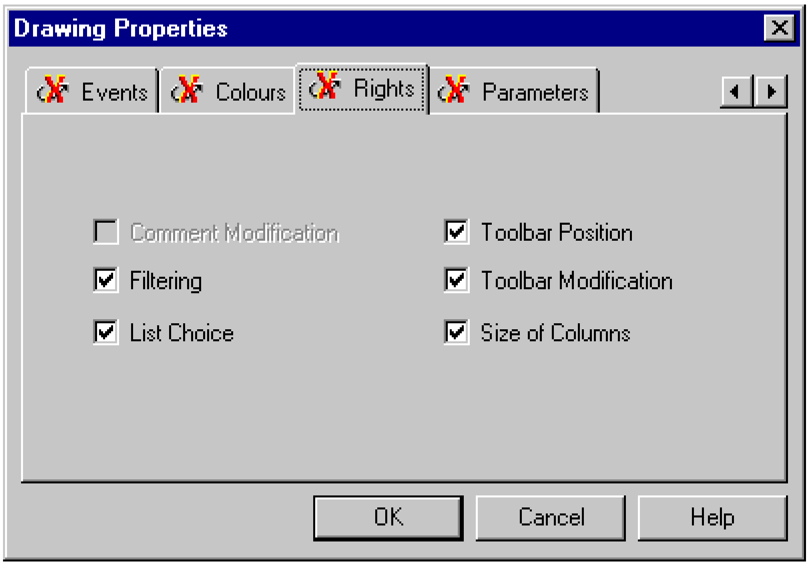
! If there is nothing in common between what is selected in the Log List and what is selected in the Events tab the Log Viewer will display nothing.

Selecting What the User can Change at Run-time

What the User can change at run-time is configured using the *Rights* tab of the Properties dialog box.

- *Filtering* - If selected the User may select a different Filter combination.
- *List choice* - If selected the User may change the Log Lists to which the Log Viewer is attached.
- *Toolbar Position* - If selected the User can change the toolbar position at run-time.
- *Toolbar Modification* - If selected, the User can change the toolbar configuration at run-time.
- *Size of Columns* - If selected, the User can change the width of columns at run-time by clicking and dragging with the mouse on the header.

If you want to change the column width or modify the toolbar, but do not want the User to do so, enable the options in design mode, switch to run mode and make the changes, then return to design mode and disable the options. Save the mimic containing the Log Viewer.



Drawing Properties

□ Comment Modification

☑ Filtering

☑ List Choice

☑ Toolbar Position

☑ Toolbar Modification

☑ Size of Columns

The Log Viewer at Run-time

How to Use the Scroll Bar | How to Use the Toolbar | Customising the Toolbar | How to Change the Display Filter

A Log Viewer displays data that has been recorded through a particular filter. The filter is attached to the Log Viewer when the Log Viewer is configured.

The types of data that may be displayed include the following :

- Alarms
  Transition to On and not acknowledged.
  Transition to On and acknowledged.
  Transition to Off and not acknowledged.
  Transition to Invalid.
  Transition to Off.
- Bits
  Change to zero.
  Change to one.
  Change to invalid.
- Registers
  Change in value.
  Change to invalid.
- Text
  Change in value.
  Change to invalid.

By using different filters, Log Viewers may be configured to create historic alarms lists, plant event logs and operator action logs. The Extended Properties may be used for instance to configure different Log Viewers for different plant areas.

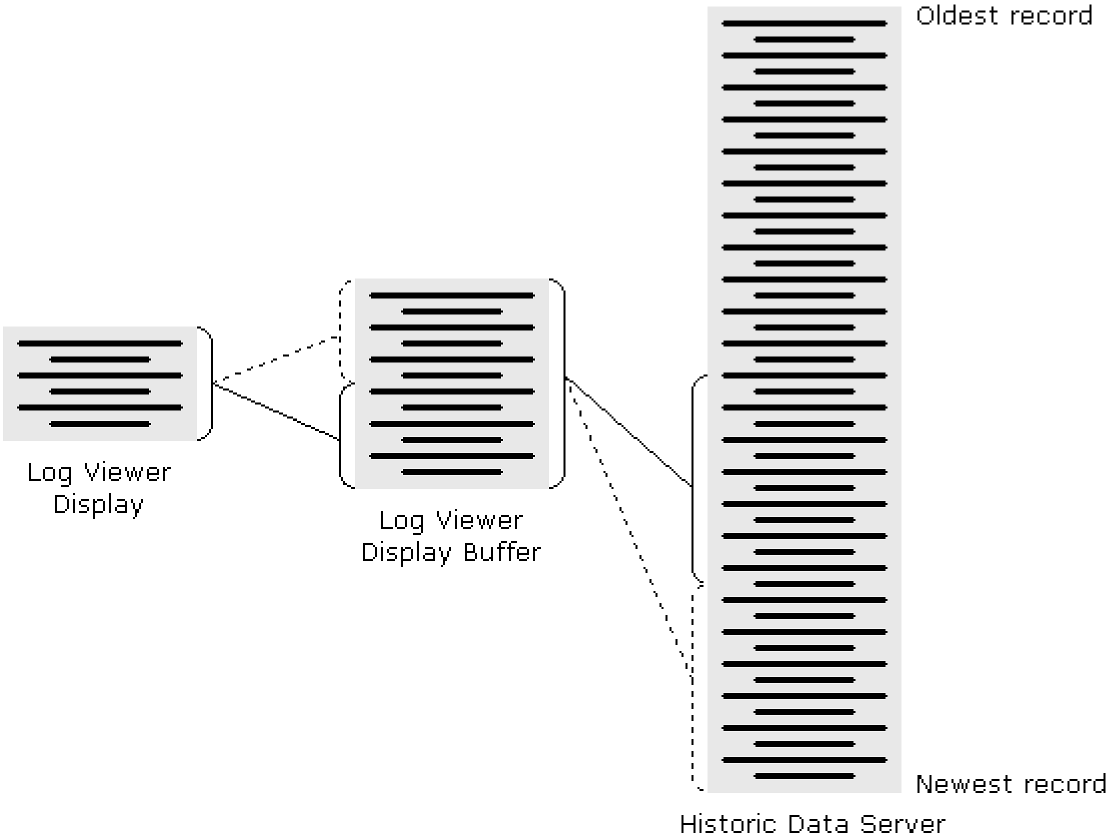
When a Log Viewer is opened, logged events are extracted from the Database through the filter and saved temporarily in the display buffer. Events are extracted for the current time backwards until the display buffer is full or there are no more events. Events are displayed starting with the most recent at the bottom of the list.

How to Use the Scroll Bar

The scroll bar is used scroll to through the data in the buffer and request new data from the HDS.

Tool	Action
⤒	Retrieve a buffer of data from the HDS starting at the oldest recorded data. The display is scrolled to show the most recent record in the buffer.
⇞	Retrieve a buffer of older data from the HDS. The newest record is one record older than the oldest record in the previous buffer. The display is scrolled to show the most recent record in the buffer.
▲	Scroll the display back in time through the buffer one line at a time. If the oldest data in the buffer is reached another buffer of data is requested, one record older than the previous record.
□	Scroll the display backwards and forwards through the display buffer.
▼	Scroll the display forward in time through the buffer one line at a time. If the newest data in the buffer is reached a new buffer of data is requested, one record newer than the previous record.
⇟	Retrieve a buffer of newer data from the HDS. The oldest record is one record newer than the newest record in the previous buffer. The display is scrolled to show the most recent record in the buffer.
⤓	Retrieves a new buffer of data from the HDS starting at the newest recorded data. The display is scrolled to show the most recent record in the buffer.
◆	Display the Log Viewer filter box.

Visually the effect of using the ⤒ button to retrieve an older buffer of data can be shown as follows. The dotted line represents the relation between the display, buffer and HDS data before the button was used.



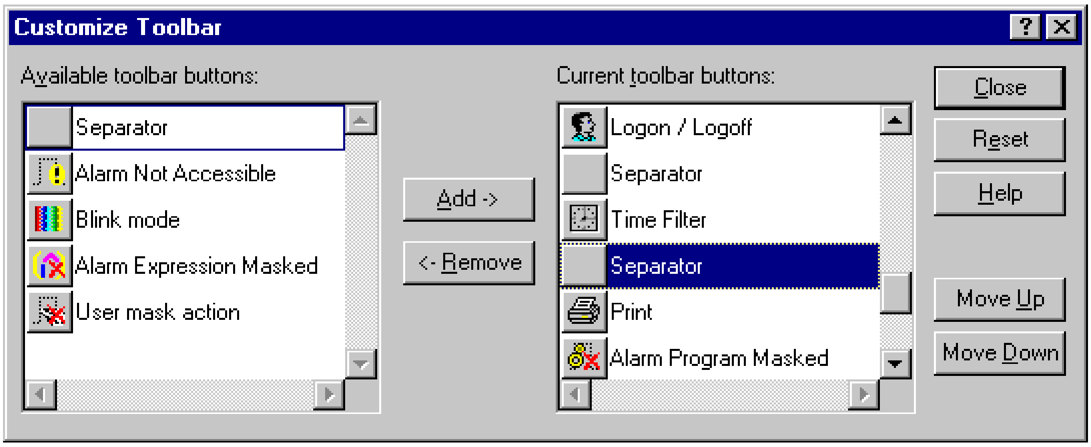
How to Use the Toolbar

The toolbar allows the User to select the type of data to be displayed and to scroll the display to a particular historic time.

Click	To
↑⎍	Display alarm transitions to Alarm on, not ack
↑⎍✓	Display alarm transitions to Alarm on, ack
↑⎍↓	Display alarm transitions to Alarm off, not ack.
↑⎍↓✓	Display alarm transitions to Alarm off.
⎍	Display alarm transitions to invalid.
↑	Display bit transitions to 1.
↓	Display bit transitions to 0.
⋮	Display bit transitions to invalid.
✗	Display alarms masked by the operator.
⎍↓	Display bit command to 0.
↑⎍	Display bit command to 1.
⋮▯	Display text or register value change.
◷	Enter a historic time to which the display is to be scrolled.
▥	Any blinking colours used in the display of alarms are inhibited and displayed as the primary colour.

Customising the Toolbar

If the Toolbar Modification property was selected when the log display was configured, at run time the User can select the position of the toolbar (top, bottom, left or right) and the tools that appear on it. (This is the only way in which the toolbar can be changed, so if you want to modify the tools that appear, but not let a User do so at run time, you must first enable toolbar modification, make the changes in Run mode, then in Edit mode disable toolbar modification.) To display the toolbar modification box, click on the toolbar with the right mouse button and select the Customize Toolbar tool.



How to Add a tool to the toolbar

1. Select the position at which the tool will be added by clicking on the tool that will immediately follow it in the Current toolbar buttons list.
2. Select the tool to be added in the Available toolbar buttons list.
3. Select the Add button.

With the exception of the Separator, as tools are added they are removed from the Available toolbar button list.

How to Remove a tool from the toolbar

1. Select the tool to be removed in the Current toolbar buttons list.
2. Select the Remove button.

How to Move a tool's position within the toolbar

1. Select the tool to be moved in the Current toolbar buttons list.
2. Select the Move up or Move down button once or more times as required.

How to Change the Display Filter and Log List

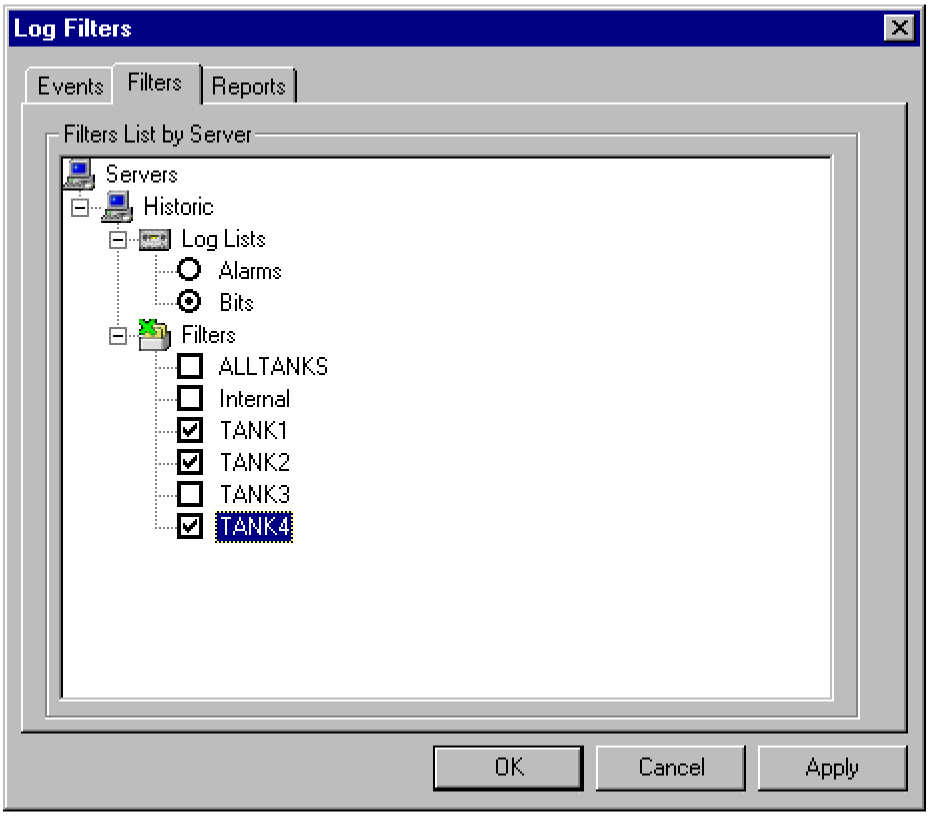
At run-time you can select the events that are displayed using the *Log Filters* dialog box opened using the small command button below the vertical scroll bar. From the *Log Filters* dialog box you can:

- Select an alternative Log List.
- Activate and de-activate Filters.
- Select events by event type.

How to select a Log List

1. Select the Filters tab.
2. Expand the Log Lists folder by clicking the plus sign adjacent to it.
3. Select the Log List by clicking on its name.
4. Click the *Apply* or *OK* button to apply the change.

Only one Log List may be activated at a time.



Filters List by Server

- Servers
  - Historic
    - Log Lists
      - ○ Alarms
      - ● Bits
    - Filters
      - □ ALLTANKS
      - □ Internal
      - ☑ TANK1
      - ☑ TANK2
      - □ TANK3
      - ☑ TANK4

How to activate Filters

1. Select the Filters tab.
2. Expand the Filters folder by clicking the plus sign adjacent to it.
3. Tick the checkbox of each filter that is to be applied. (To de-select a filter, click on its checkbox to remove the tick.)
4. Select the cumulative effect of multiple filters as either exclusive (AND) or inclusive (OR). Right-click on *Filters* then choose either And condition or Or condition. The AND condition includes events in the Viewer only if they satisfy all filters assigned to it. The OR condition includes events in the Viewer if they satisfy any filter assigned to it.
5. Click the *Apply* or *OK* button to apply the change.

How to select events by event type

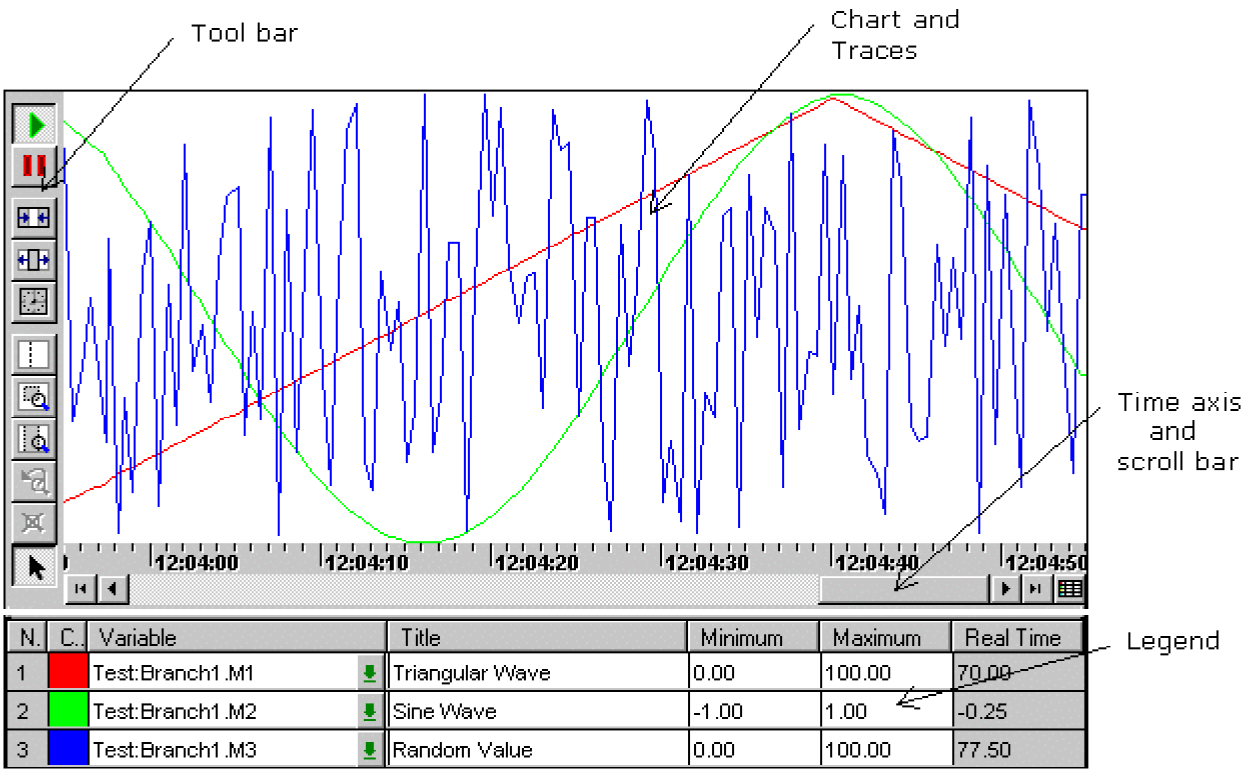
1. Select the Events tab.
2. Tick the checkbox of each event type to be displayed. (To de-select an event type, click on its checkbox to remove the tick.)
3. When the required event types are selected, select OK to close the dialog box.

Changing the choice of Log List, Filter and event types at run-time is only a temporary operation; when the window containing the Alarm Viewer is closed the changes are lost unless the windows is saved either from VBA or configuration menu.

Trend Viewer Introduction

Trend Features

The Trend Viewer is an ActiveX control that produces a display of up to eight Register or Bit variables plotted against time. Any number of Trend Viewers may be embedded in a mimic, the only limitation being the amount of space available in the mimic.



N.	C.	Variable	Title	Minimum	Maximum	Real Time
1	🟥	Test:Branch1.M1	Triangular Wave	0.00	100.00	70.00
2	🟩	Test:Branch1.M2	Sine Wave	-1.00	1.00	-0.25
3	🟦	Test:Branch1.M3	Random Value	0.00	100.00	77.50

Trend Features

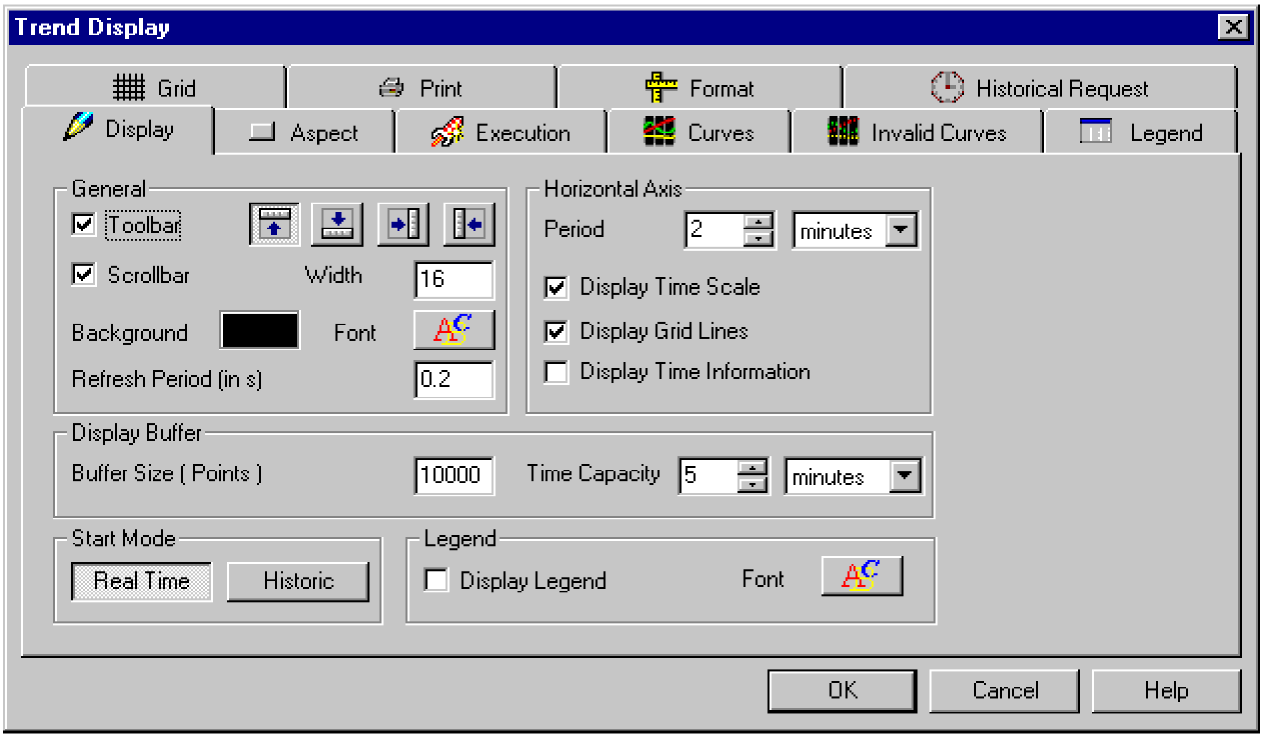
- Up to eight Register and/or Bit variables may be displayed on each Trend.
- Historical and real-time modes.
- Y axis (range) independently configurable for each trace.
- X axis (time) configurable from 1 second to 32767 days.
- Toolbar to change run time configuration.
- Zoom on both X and Y axes.
- Scroll bars for the X and Y axes.
- Configurable Legend area displaying detail of each trace.
- Highly configurable appearance.
- Trend Viewers may be embedded in any mimic
- Unlimited number of Trend Viewers in each project.

How to Draw a Trend Viewer

Choosing the Main Display Features | Configuring the Display Buffer | Configuring the Traces | Configuring Appearance of Trend Viewer | Configuring the Traces | Selecting the Run time Tools | Configuring the Legend Area | Configuring the Grid | Configuring the Printing of a Trend Viewer | Configuring Display Formats | Historical Request

The basic method of creating a Trend Viewer is the same as for any other native drawing element.

1. Open the mimic in which the Trend Viewer will be inserted and select Design mode.
2. Select the Trend Viewer tool from either the toolbar or the Insert menu.
3. A Trend Viewer will be created and its Properties dialog box automatically opened ready for the configuration process.



General
☑ Toolbar
☑ Scrollbar Width 16
Background ■ Font
Refresh Period (in s) 0.2

Horizontal Axis
Period 2 minutes
☑ Display Time Scale
☑ Display Grid Lines
□ Display Time Information

Display Buffer
Buffer Size (Points) 10000 Time Capacity 5 minutes

Start Mode
Real Time Historic

Legend
□ Display Legend Font

Once you have configured the Trend Viewer and closed its Properties box it can be moved, sized and copied in the same way as for any other drawing element.

Choosing the Main Display Features

The following properties, found in the Display tab, are used to select which of the optional features are displayed at run time.

- *Toolbar* - Display the trend toolbar. The toolbar may be positioned at the top, bottom, left or right of the Trend Viewer. The tools that are displayed on the toolbar can be selected in run mode.
- *Scroll bar* - Display the X axis scroll bar and tools. The height of the scroll bar, in pixels, can also be selected.
- *Background* - The colour of the chart area on which the traces are plotted.
- *Display Time Scale* - Display a time axis scale at the bottom of the chart area.
- *Display Grid Lines* - Display vertical grid lines in the chart area.
- *Display Time Information* - Display the time at the start and end of the chart, the period of the chart and the period of the time grid.
- *Period* - The period for the time axis entered as a number of seconds, minutes, hours or days.

- *Refresh period* – A background refresh period, in seconds and tenths of seconds, at which the chart area is refreshed when the values do not change. (Normally the display is refreshed when any of the values change.)
- *Real time* - Starts the Trend Viewer in real-time mode.
- *Historic* - Starts the Trend Viewer in Historical mode.
- *Display Legend* - Enable display of the legend area and select the font in which text in this area appears.

Configuring the Display Buffer

Each Trend Viewer contains a configurable display buffer where values for each trace are stored.

In a real-time Trend Viewer the display buffer temporarily records values for the traces, thus giving the User a limited ability to scroll back in time. The size of the slider on the scroll bar changes to reflect the amount of data stored in the buffer. For example, if the buffer contains 10 minutes of data and the chart is 2 minutes, the slider will be approximately one fifth of the length of the scroll bar.

In a Historical Trend Viewer the buffer is used to cache information from the Database so that it does not need to be accessed so frequently. The size of the slider within the scroll bar is proportional to the chart and buffer periods.

When the mimic containing the Trend Viewer is closed, the buffer is emptied unless the mimic is cached.

The following properties are used to configure the display buffer.

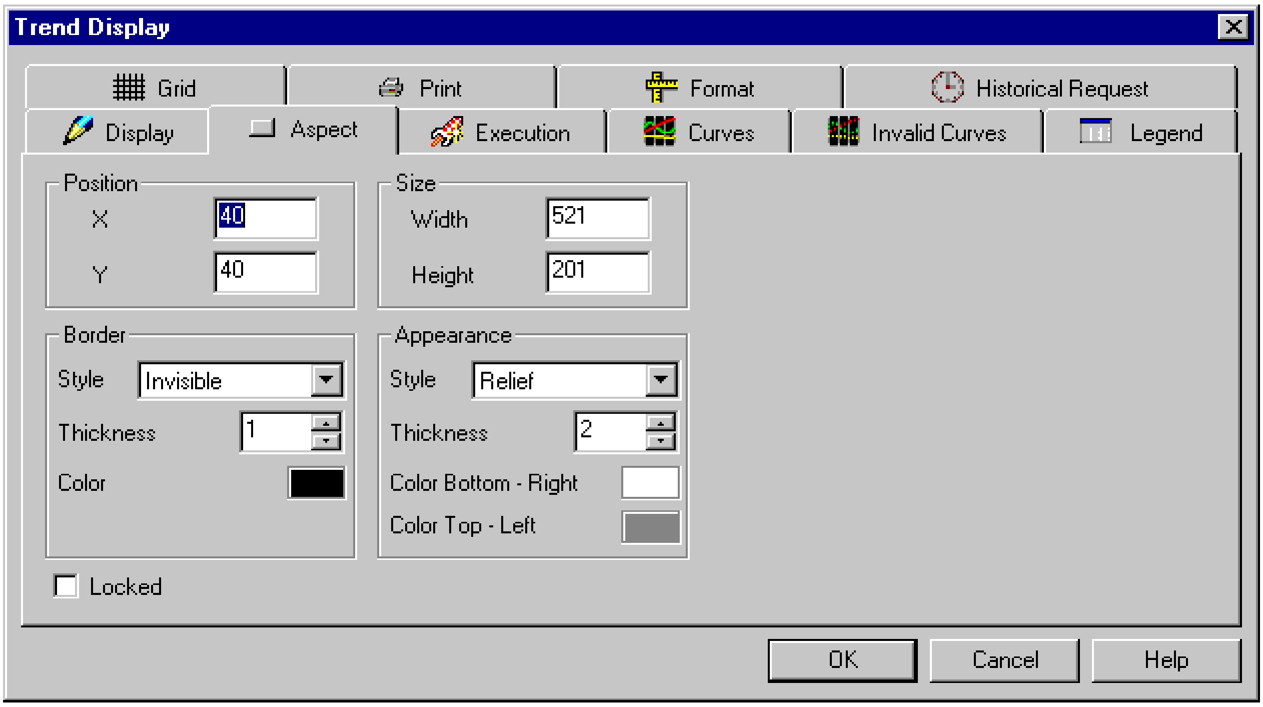
- *Time capacity* - The maximum period for which the display buffer will retain values for the traces.
- *Buffer Size (Points)* - The maximum number of values that may be recorded in the display buffer. Normally this will be left at the default value of 10,000. However if the application has any special requirements (for example, reducing the amount of memory required) this property may be changed. Values are discarded when either the time capacity is reached or the display buffer is full

Configuring the Appearance of the Trend Viewer

You can configure the appearance of a Trend Viewer by clicking on the Aspect tab of its properties dialog box. The available settings are as follows.

- *Position* - To set the distance in pixels between the Trend Viewer and the left hand edge of the workspace, enter a number in the box 'X'. To set the distance in pixels between the Trend Viewer and the upper edge of the workspace, enter a number in the box 'Y'.
- *Size* - To change the size of the Trend Viewer, enter the required number of pixels in Width or Height.
- *Border* - You may configure the border of a Trend Viewer. To select a Style (such as Solid), click on the down arrow. Enter a *Thickness* directly or by using the up and down arrows adjust the current thickness. To configure the colour of a border, click on the *Color* box ■ to open the Color Palette and select the required colour.
- *Appearance* – This enables a Trend Viewer to be given a surround other than a simple border. Click on the down arrow to select an Appearance (such as *Shadow*). Enter a Thickness directly or by using the up and down arrows adjust the current thickness. Depending on the Style you select, you can choose the colours for the sides of the Trend Viewer (Bottom and Right separately from Top and Left).
- *Locked* - Click the check box to lock the position of the Trend Viewer in the mimic, and click again to unlock it.

A Border is applied directly in design mode as it will appear in run mode. An Appearance is only visible in run mode.



Position
X 40
Y 40

Size
Width 521
Height 201

Border
Style Invisible
Thickness 1
Color ■

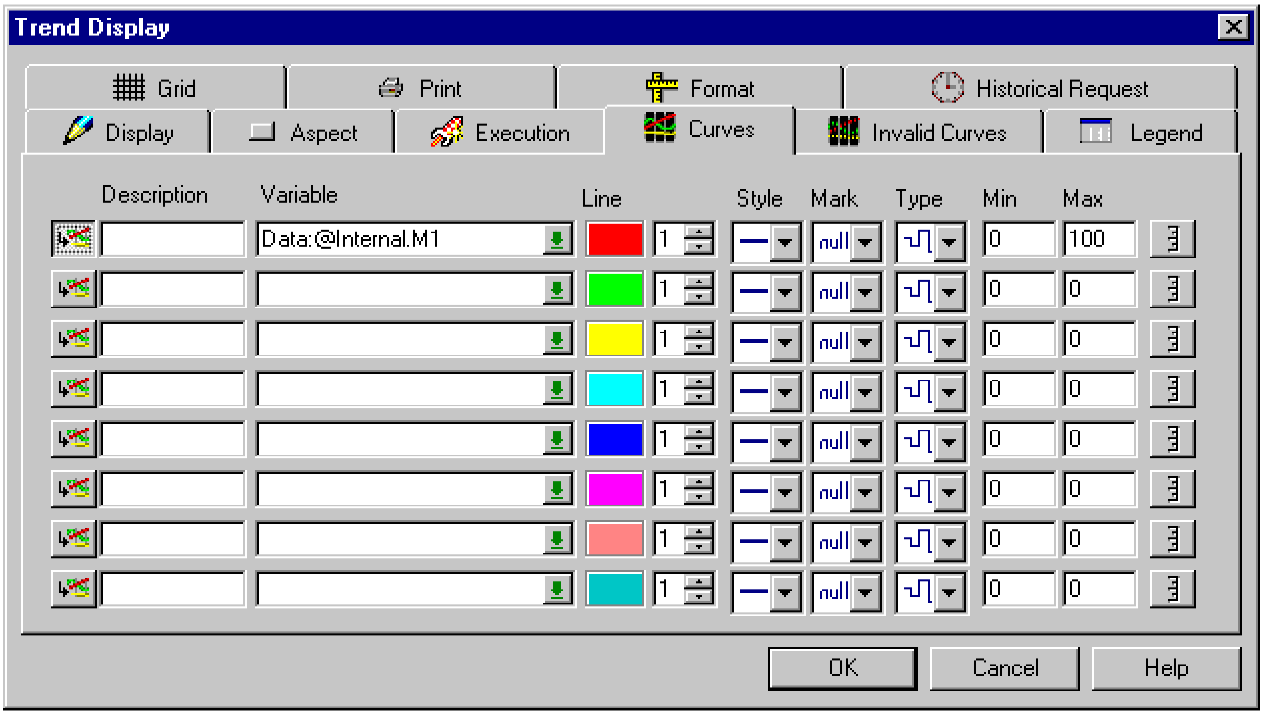
Appearance
Style Relief
Thickness 2
Color Bottom - Right □
Color Top - Left ■

☐ Locked

Configuring the Traces

The Curves and Invalid Curves tabs allow you to select which variables will be displayed and the appearance of the corresponding traces.

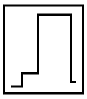
Configuring the Appearance of Traces for Variables



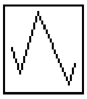
Description	Variable	Line	Style	Mark	Type	Min	Max
	Data:@Internal.M1	🟥 1	—	null		0	100
		🟩 1	—	null		0	0
		🟨 1	—	null		0	0
		🟦 1	—	null		0	0
		🟦 1	—	null		0	0
		🟪 1	—	null		0	0
		🟥 1	—	null		0	0
		🟦 1	—	null		0	0

- *List Trace in Legend* – Click a button in the first column to list a trace in the Legend at run time. If that trace is already listed, click the button if you wish to remove it.

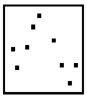
- *Description* – The description for each trace that appears in the Trace Detail Box at run time.
- *Variable* - The name of the variable to be displayed. Clicking the down arrow button will display the OPC Variable Browser from where the variable may be selected.
- *Line* - The colour and width of the trace.
- *Style* - The style of the line used for the trace.
- *Mark* - An optional marker that is plotted each time the variable attached to a trace changes. Click the down arrow to select a marker (such as ▲).
- *Type* - Selects the way in which the trace will be displayed.



Each point representing a change in value is connected to the previous one with a vertical and horizontal line.



Each point representing a change in value is connected to the previous one with a single line.



Each point representing a change in value is plotted with no interconnecting lines.

- *Min & Max* - The numeric range of the variable within the trace. If *Min* and *Max* are both left at zero then the range is taken from the variable's properties at run time.
- *Show Vertical Scale* – Click on a button in the final column to cause a vertical scale to be shown on the left side of the chart area. The scale will have the same colour as the trace to which it refers. If that scale is already shown, click the button if you wish to remove it.

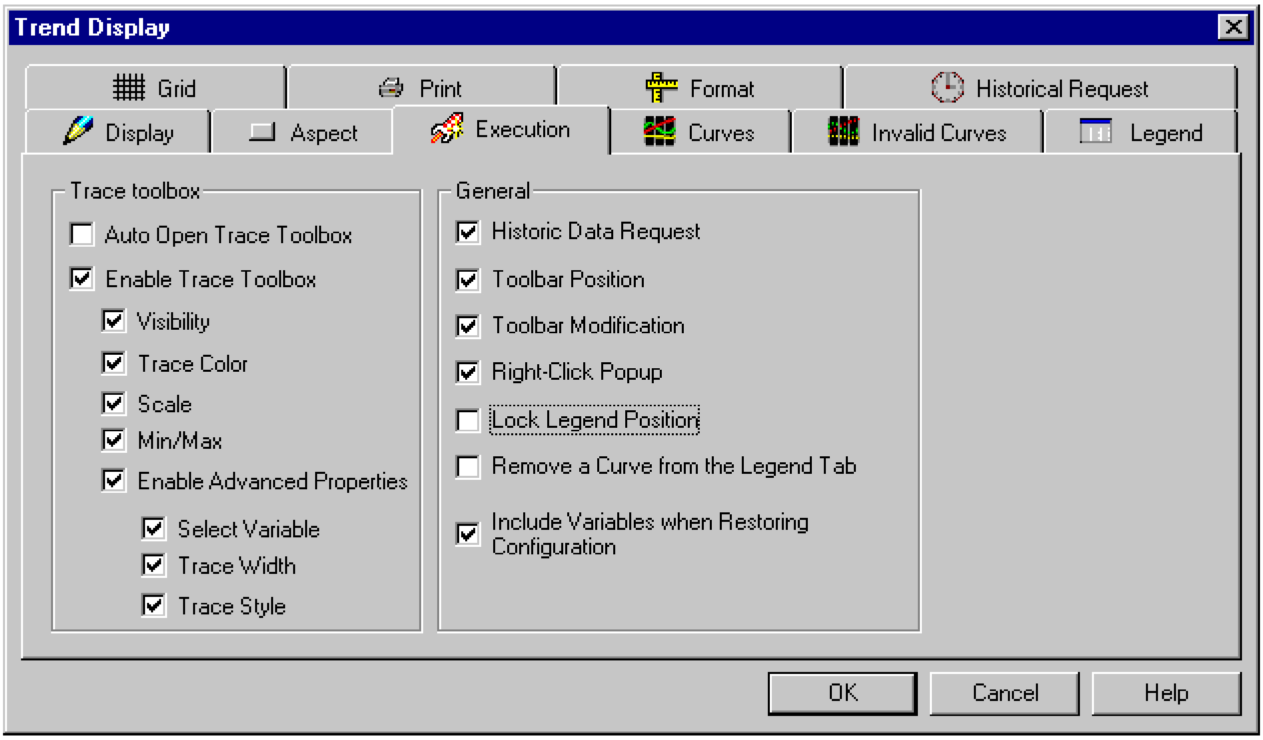
The technique of joining successive points with a single line ⋀ is often used for drawing a trace because of its simple layout. However remember that the values are only valid at each apex, and anywhere else on the line is interpolated. Hence the recommended option is to join each point with a vertical and horizontal line ⎍.

Configuring the Appearance of the Trace for an Invalid Variable

The Invalid Curves tab allows you to specify the appearance of a trace when its corresponding variable is invalid. Selecting a different appearance for a trace when the variable is invalid allows the User to detect easily when there is a problem with the variable. You can select both a different colour and style (for example dashed) for the invalid trace.

Selecting the Tools

The tools available to the User at run time are selected from the Execution tab.



Trace toolbox

- □ Auto Open Trace Toolbox
- ☑ Enable Trace Toolbox
  - ☑ Visibility
  - ☑ Trace Color
  - ☑ Scale
  - ☑ Min/Max
  - ☑ Enable Advanced Properties
    - ☑ Select Variable
    - ☑ Trace Width
    - ☑ Trace Style

General

- ☑ Historic Data Request
- ☑ Toolbar Position
- ☑ Toolbar Modification
- ☑ Right-Click Popup
- □ Lock Legend Position
- □ Remove a Curve from the Legend Tab
- ☑ Include Variables when Restoring Configuration

- *Historic Data Request* - Enables the Historical button on the tool bar at run time.
- *Toolbar Position* – Allows the User to change the location of the toolbar.
- *Toolbar Modification* – Allows the User to select the tools that appear on the toolbar.
- *Right-Click Popup* – Enables display at run time of the pop-up menu, that is used to select the cursor mode.
- *Lock Legend Position* – Prevents the User from dragging the border between Legend and chart area up and down at run-time.
- *Remove a Curve from the Legend Tab* – Enables the User to delete a variable from the trend at run time by right clicking on its entry on the Legend.
- *Include Variables When Restoring Configuration* - In run mode, when selecting the Restore tool, the variables displayed on the trend revert to those selected at design time.

! Changes made to a Trend Viewer at run time are not saved when you revert to Design mode or close a mimic. When the mimic is restarted, each Trend Viewer applies its default settings.

The Trace Toolbox

The Trace Toolbox is included only for backwards compatibility. The functionality it provides has now been superseded by that provided by the Legend.

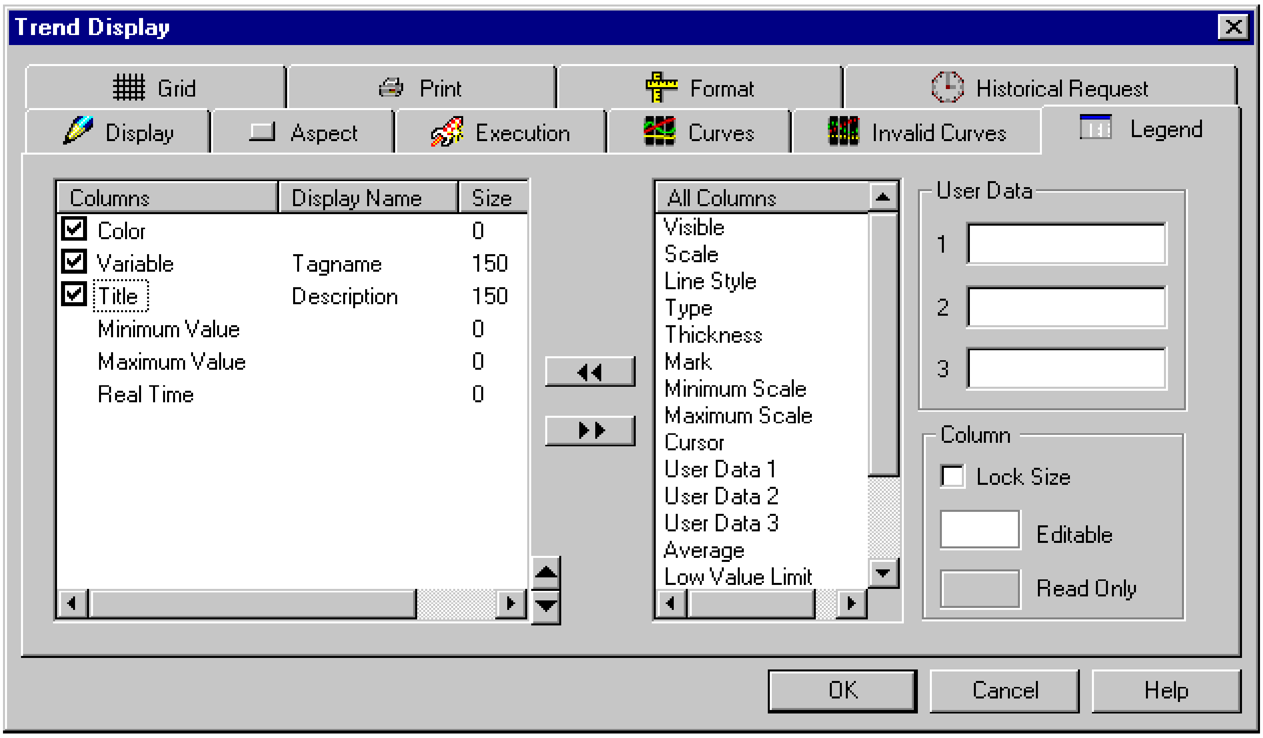
The Trace Toolbox provides additional information about each of the traces in the Trend and allows the User to modify the traces. The Trace Toolbox and each of its properties may be enabled or disabled as required. Note that any changes made are temporary and will be lost when the Trend Viewer is closed.

- *Auto Open Trace Toolbox* - If selected, the Trace Toolbox is automatically opened when a trend is displayed (when the mimic containing it is opened).
- *Enable Trace Toolbox* - Enables the User to open the Trace Toolbox.

- *Visibility* – Enables individual traces to be temporarily hidden. This can be useful when one trace is obscuring another.
- *Trace Color* – Allows the colour of each trace to be changed.
- *Scale* – Allows the scale for each trace to be displayed or hidden.
- *Min/Max* – Allows the range for each trace to be changed.
- *Enable Advanced Properties* – Allows access to the toolbox Advanced Properties. Each of the features below may be individually enabled and disabled.
- *Select variable* – Allows the variable for each trace to be changed.
- *Trace Width* – Allows the width of each trace to be changed.
- *Trace Style* – Allows the plotting style for each trace to be changed.

Configuring the Legend Area

In the Display tab, tick the *Display Legend* check box to cause a descriptive Legend of traces to appear below the chart area of the Trend Viewer. The Legend area will only appear when you switch to Run mode. The Legend tab is used to configure the Legend area. The Legend displays a line of properties for each trace. Some properties are always read only, whereas others can be selected to be read only or editable. You can select different colours to differentiate between read only and editable fields.



Columns	Display Name	Size
☑ Color		0
☑ Variable	Tagname	150
☑ Title	Description	150
Minimum Value		0
Maximum Value		0
Real Time		0

All Columns: Visible, Scale, Line Style, Type, Thickness, Mark, Minimum Scale, Maximum Scale, Cursor, User Data 1, User Data 2, User Data 3, Average, Low Value Limit

User Data
1
2
3

Column
□ Lock Size
Editable
Read Only

What information can be displayed on the Legend?

- *Visible* - A command button from which the display of the trace can be toggled.
- *Scale* - A command button from which the display of the trace scale can be toggled.
- *Color* - The colour of a trace. Editable.
- *Line Style* - The style in which the trace line is drawn. Editable.
- *Type* - The way in which the trace is drawn (angular, square wave or dotted). Editable.
- *Thickness* - The thickness of the trace. Editable.
- *Mark* - The optional marker shown at each change of the variable. Editable.
- *Title* - The description assigned to a trace. Editable.
- *Minimum Value* - Displays the minimum value for the variable within chart area. Historical mode only. Read only.

- *Maximum Value* - Displays the maximum value for the variable within the chart area. Historic mode only. Read only.
- *Average* – The average of the values for a variable within in the chart area. Historic mode only. Read only.
- *Variable* - The name of the variable allocated to a trace. Editable.
- *Real Time* - The current value of the variable allocated to a trace. Read only.
- *Cursor* - The value of the trace at the cursor position. Read only.
- *User Data 1-3* - User configurable information fields. Editable.
- *Minimum Scale* - Used with *Maximum Scale* to specify the range of values for a trace. Editable.
- *Maximum Scale* - Used with *Minimum Scale* to specify the range of values for a trace. Editable.
- *High Value Limit* – The highest possible value of the variable from its OPC property HiValueLimit. Real-time mode only. Read only.
- *Low Value Limit* - The lowest possible value of the variable from its OPC property LoValueLimit. Real-time mode only. Read only.
- *Variable Description* – The description of the variable from its OPC property Description. Read only.
- *Units* - The engineering units for the variable from its OPC property Units. Read only.

Selecting which properties are displayed

To select a property for display, click on its line in the *All Columns* list and click the double left arrow command button.

To deselect a property for display, click on its line in the *Columns* list and click the double right arrow command button.

Alternatively you can select or deselect a property by double clicking on its name.

Setting Colours to Designate Properties as Editable or Read Only

Click on the box ▬ marked Editable to open the Colour Palette and select a colour to distinguish editable properties. Click on the box marked Read Only to open the Colour Palette and select a colour to distinguish Read Only properties.

The colour-coding of traces of variables is unrelated to that of properties. Conventional colours for properties are light grey for Read Only and clear for Editable.

Configuring a column

- *Read Only* - To make the property displayed in the column read only, de-select the tick box.
- *Display Name* - To change the description of the column, click its line in the *Columns* list and then on the *Display Name* field. You can then type in your own description.
- *Size* - To change the width of a column, click on its line in the *Columns* list and then click the *Size* field. You can then enter a width for the column in pixels. The width of the columns can also be changed in run mode by the User, unless the *Lock Size* tick box is selected.
- *Lock size* - If selected the User is unable to change the column widths at run-time.

If you double click on the name of a column, the entry for that column is removed from the *Columns* list.

Configuring the Grid

The Grid tab enables you to configure the appearance of a grid that is displayed in the chart area at Run Time.

Configuring the Lines of the Grid

- *Automatic* - When the *Automatic* check box is ticked, the chart area is divided by vertical lines at time intervals that depend on the scale of the display. The other options are greyed out. Uncheck the Automatic check box to alter the settings.
- *Color* - To configure Vertical Lines for a main Division, click on the *Color* box ▬ to display the Color Palette and select a colour for the line. Click on the down arrow to select a Style. Use the Subdivision settings to select the colour and style of subdividing lines, if required.
- Configure Horizontal Lines in a similar manner.

Configuring the Time Scale

- Uncheck the Automatic check box to alter the settings.
- Enter an interval for main Divisions, and click the down arrow to select Days, Hours, Minutes or Seconds.
- Configure the interval for Subdivisions in a similar manner.

Closely-spaced grid lines can obscure the chart area when you zoom out. To prevent that, tick the *Time Scale . Automatic* check box.

Configuring the Printing of a Trend Viewer

The *Print* tab gives access to the settings that control printing of a Trend Viewer at Run Time.

- To set a background colour, click on the *Back Color* box.
- Enter the number of Copies to be printed.
- Choose the Orientation as either *Landscape* or *Portrait*.

Configuring Display Formats

The *Format* tab lets you modify the settings that control display formats for times and values at Run Time.

- Configure the time display in Cursor Mode in a similar manner.
- For display of values, click the downward arrow to select a setting for Maximum Number of Decimal Digits.
- When values with many digits are to be displayed, you can choose a threshold beyond which the values are shown in exponential format. Tick the check box Use Exponent when Value is Greater Than and click the down arrow to set an upper level. Tick the check box for Less Than and click the down arrow to set a lower level.
- The format in which time is displayed on the Time Scale is configured using special format characters with the following default setting: #D/#M/#Y #h:#m:#s. To change the format, adjust the components of the string as follows.

Character	Description	Format (Range)
#Y	The year	2 digits
#M	The month	2 digits (01 - 12)
#D	The date	2 digits (01 - 31)
#h	The hour	2 digits (01 - 23)
#m	The minute	2 digits (00 - 59)
#s	The second	2 digits (00 - 59)

Historic Request

The Historic Request tab enables you to specify when a historical request is to be processed.

- *At the Opening of the Mimic in Real Time* - To process a historical request as soon as the mimic is opened, tick the check box. Otherwise processing has to be initiated by the User.
- *When Switching from Historic Mode to Real Time* - To process any historical request when the Trend Viewer reverts to Real Time mode, tick the check box.
- If both boxes are left unchecked, Historical requests are processed when the User initiates them.

The icon indicates that Historic data is not available.

The Trend Viewer at Run Time

Real-time Mode | Pause Mode | Historical Mode | Using the Scroll Bar | Using the Toolbar | Customising the Toolbar | Using the Cursor

The Trend Viewer plots the value of up to eight Database variables (bit or register) against time. The horizontal axis represents time, and the vertical axis, value.

In a real-time Trend Viewer the display buffer temporarily records values for the traces, thus giving the User a limited ability to scroll back in time. The size of the slider on the scroll bar changes to reflect the amount of data stored in the buffer. For example, if the buffer contains 10 minutes of data and the chart is 2 minutes, the slider will be approximately one fifth of the length of the scroll bar.

In a Historical Trend Viewer the buffer is used to cache information from the Database so that it does not need to be accessed so frequently. The size of the slider within the scroll bar is proportional to the chart and buffer periods.

The Trend Viewer operates in three modes.

- Real-time. The traces are plotted on the chart in real-time as the value of the variables they represent change. A Trend Viewer may be configured so that it starts up in real-time or it may be selected using the ▶ tool at run time.
- Historical. The traces are plotted on the chart from the values of variables recorded in the Database. A Trend Viewer may be configured so that it starts up in Historical mode or it may be selected using the ◷ tool at run time.
- Pause mode. The traces are plotted on the chart from the values of variables recorded in the display buffer. The Trend Viewer may be switched from real-time to pause mode by using the ❚❚ tool, or by using the scroll bar.

To increase the height of the chart area and conceal inactive variables in the Legend, drag the lower edge of the chart area downwards. (The tick box, *Lock Legend Position,* in the Execution tab must be un-ticked to allow this action.) To adjust the widths of columns, drag the lines between them sideways. (The tick box, *Lock size*, in the Legend tab must be un-ticked to allow this action.)

To remove a variable's trace from the chart area and its details from the Legend, right-click on its Legend row and confirm the action *Delete Variable*. (The tick box, *Remove a Curve from the Legend*, on the Execution tab must be un-ticked to allow this action.)

Real-time Mode

In real-time mode, new values for each trace are plotted starting on the right hand side of the chart. Older data scroll across the chart, eventually disappearing off the left hand side.

The frequency at which the chart is refreshed depends on the configuration and how often the data are changing.

- The chart is always updated when any of the variables changes value.
- The chart is also refreshed at a background rate determined by the configuration of the Trend Viewer. For example if the background refresh was set to 5, the chart would refresh every 5 seconds even if none of the variables has changed.

A Trend Viewer records temporary Historical data in its display buffer. The buffer continues to record data all the time the trend is displayed, while discarding the oldest values as it becomes full. As the buffer fills the size of the slider on the scroll bar changes to reflect the amount of data stored as a ratio of the chart period.

Pause Mode

The Trend Viewer automatically switches to pause mode whenever you click and drag the slider button on the scroll bar, or select the pause tool.

In pause mode the chart area no longer updates in real-time, instead it displays the temporary Historical data recorded in the display buffer. However, the buffer of the Trend Viewer is still updated in real-time with the oldest values being discarded from the display. The effect is that the data being displayed on the chart is gradually moving through the buffer. This can be seen by the fact that the slider button on the scroll bar gradually moves towards the left. Hence, if you leave the Trend Viewer in Pause mode for long enough, the data for the time period displayed will be discarded.

When in pause mode you can scroll backwards and forwards through the data using the scroll bar. You can also zoom in and out in both the X and Y axis using either the toolbar or the cursor.

Historical Mode

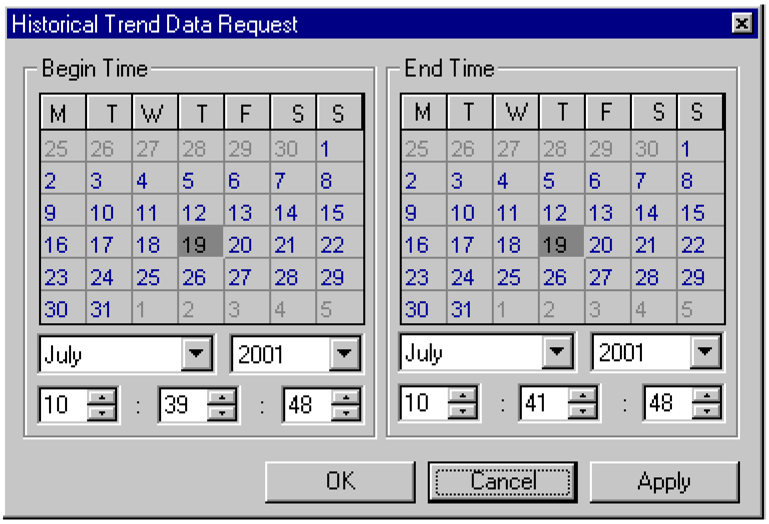
In Historical mode, the chart area displays Historical data extracted from the Database. When you open a Historical Trend Viewer at run time it will initially display data from the current time backwards for the period defined by its configuration.

The display buffer is used to cache the data from the Database to reduce the frequency at which they need to be accessed.

In Historical mode, you can scroll backwards and forwards through the data using the scroll bar and zoom in and out in both the X and Y axis using either the toolbar or the cursor.

Changing the Display Period

You can change the display period in Historical mode by selecting the same tool used to switch to Historical mode. The box that is then displayed allows you to select the start and end time (and hence the period) for the data displayed on the chart.



Historical Trend Data Request

Begin Time

M	T	W	T	F	S	S
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

July 2001

10 : 39 : 48

End Time

M	T	W	T	F	S	S
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

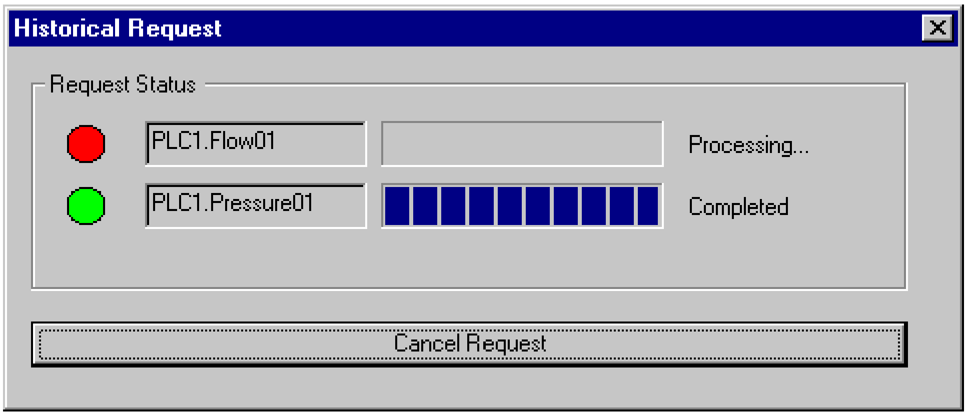
July 2001

10 : 41 : 48

Once you have selected the new start and end times you apply them to the chart by clicking either the *OK* or *Apply* buttons. If you select the *OK* button the new times are applied and the box is closed; selecting the *Apply* button will apply the new times but leave the box open so that you may make further adjustments to the time period.

Making a Historical Request

When you click *OK* to initiate a Historical Request, the time taken to process the request depends on the amount of data to be analysed from the Database. When the analysis lasts longer than an instant, you will see a progress report that shows the status of processing for each of the variables selected.



Historical Request

Request Status

🔴 PLC1.Flow01 Processing...

🟢 PLC1.Pressure01 Completed

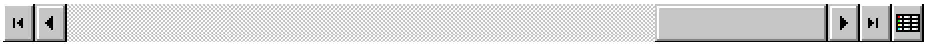
Cancel Request

- *Cancel Request* – To cancel the request, click on the long button.

The coloured circles denote the status of processing of each variable (not the colour codes of traces of the variables) as follows: red 🔴 – in progress, green 🟢 – completed.

Using the Scroll Bar

The scroll bar is used to select data in the display buffer for display on the chart.



You can get an approximate idea of the ratio of the number of values that may displayed in the chart to those held in the buffer by the size of the scroll button in relation to the area in which it may be moved.

The way in which the scroll bar tools behave changes slightly depending on which mode the Trend Viewer is in.

Tool	Real-time	Pause	Historical
⏮	Select Pause mode and scroll to oldest data in display buffer.	Scroll to oldest data in display buffer.	If not displaying the oldest buffered data, scroll through buffer to display it. If already displaying the oldest buffered data, load the buffer with older data from the Database.
◀	Select Pause mode and scroll back through data in display buffer.	Scroll back through data in display buffer.	Scroll back through data in display buffer.
▶	No Action.	Scroll forward through data in display buffer.	Scroll forward through data in display buffer.
⏭	No action.	Select real-time mode.	If not displaying the newest data in buffer, scroll through buffer to display it. If already displaying newest buffered data load the buffer with newer data from the Database.
▦	Display the Trace Toolbox.	Display the Trace Toolbox.	Display the Trace Toolbox.

□	Select Pause mode and scroll through data in display buffer.	Scroll through data in display buffer.	Scroll through data in display buffer.

Using the Toolbar

The toolbar contains the tools used to select the mode of display and for zooming in and out of the chart time axis.

Select Real Time mode.

Select Pause mode.

Explode time axis. The time axis is doubled.

Implode time axis. The time axis is halved.

Restore the time axis to its original setting.

Select Historical mode. The Historical Request dialog box is displayed.

Select Cursor mode to display the values of all traces at a particular time.

Zoom in area.

Zoom in time axis only.

Return to previous zoom level.

Cancel all zoom.

Restore the trend to its design time configuration.

Change the period represented by the chart. The Change Period dialog box is displayed.

Print the Trend Viewer on the default printer.

How to zoom the time axis

1. Select the zoom time axis tool. The mouse pointer will appear as a magnifying glass while it is over the chart area.
2. Click on the chart at the point representing the new start time for the chart. Hold down the mouse button and drag the cursor to the point representing the new end time. Release the mouse button.

How to zoom both axes

1. Select the zoom area tool. The mouse pointer will appear as a magnifying glass while it is over the chart area.
2. Click on the chart at the point representing the new start time and maximum value for the chart. Hold down the mouse button and drag the cursor to the point representing the new end time and minimum value. Release the mouse button.

Customising the Toolbar

If you have enabled the property *Toolbar Modification*, the User is able to change the toolbar at run time. You can position the toolbar on the left (default), right, top or bottom sides of the trend and select the buttons to be displayed.

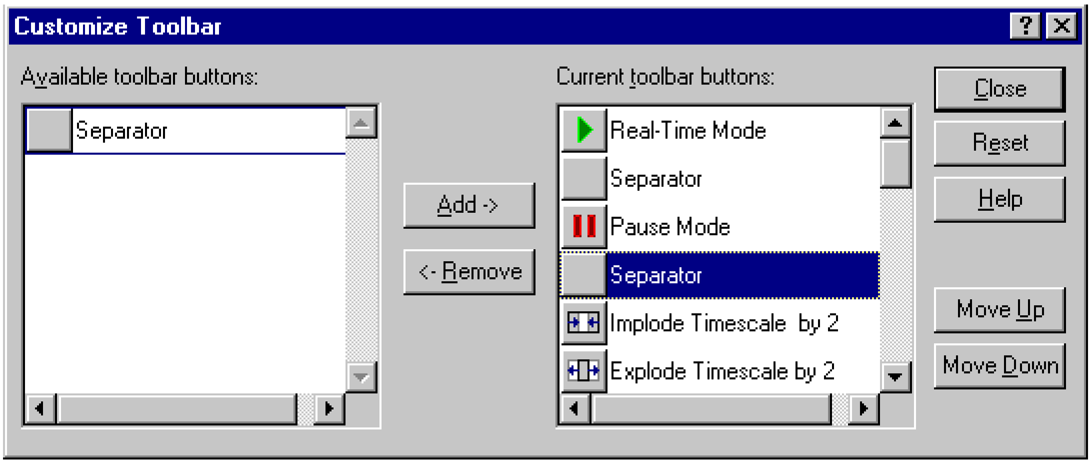
How to position the toolbar

1. Click on the toolbar with the right mouse button.
2. From the pop-up menu select the new position.

How to select which buttons are displayed

1. Click on the toolbar with the right mouse button.
2. From the pop-up menu select the *Toolbar Modification* command .
3. Use the list of toolbar buttons and the *Add* and *Remove* commands to select the buttons to be displayed.
4. Use the *Move Up* and *Move Down* buttons to change to position of a toolbar button within the list.
5. Select the *Close* command when finished.
6. Save the mimic in which the trend is placed.

You can customise the toolbar and then de-select the *Toolbar Modification* property if you want to change the toolbar but not make this option available to the User.



Available toolbar buttons:
- Separator

Current toolbar buttons:
- Real-Time Mode
- Separator
- Pause Mode
- Separator
- Implode Timescale by 2
- Explode Timescale by 2

Using the Cursor

When in Pause mode or Historical mode you may use the cursor to interact with the chart area. Using the cursor you can :

- Display the value of each of the traces at a particular point in time.
- Display crossed lines to define a point in the chart area.
- Zoom the trend simultaneously in both the time (horizontal) axis and value (vertical) axis.

You select the operating mode for the cursor from either the toolbar or a pop-up menu displayed by right clicking anywhere in the chart area

Select Cursor mode at a particular point on the time scale.

Revert to ordinary mode (neither Cursor nor Crosslines).

Select Crosslines mode.

How to display the value of the traces

1. Select the Cursor mode tool. The mouse pointer will appear as a cross while it is over the chart area.
2. Click anywhere in the chart area. The values for the traces at that point will be displayed in the Legend area and/or in the Scale area if enabled.

An Introduction to Archive and Restore

The Archive/Restore Wizard | What is and What is Not Archived?

The Archive/Restore Wizard is provided to guide you through the process of archiving a project or that of restoring a previously archived project.

The archive process stores a copy of a project in a .zip file. (A .zip file is equivalent to a folder for data files but it is compressed to occupy less space than that of the individual files it contains.) The destination media you choose must accommodate the files of your project in compressed form. Conversely, the process of restoration takes an archived project created by the Archive/Restore tool and de-compresses its contents into a storage location that you choose and registers the OFS configuration.

The archiving process leaves the original files in place, and the restore process leaves the archive files intact. Hence the Archive/Restore tool is fail-safe in both directions. However, you will need a scheme for managing copies of your projects on a regular basis. Here are some general points of good practice.

- Locations: Choose secure locations for archiving your project across a network, on removable / read-only media or off-site. In that way, your project data will be more likely to survive loss of your working copies, hard disk, computer or premises.
- Backup copies: To maintain backup copies of a project under development or live revision control, create a set of folders to be used in rotation. You can then avoid overwriting any backup copy until it is no longer required.
- Rehearsal: When you have archived a project, test your backup by restoring to an alternative location.
- Restoration: Before you restore a project that has apparently been damaged, archive the damaged copy. It will then be available for technical investigation or for salvaging of changes made since the archive was created.
- Long-term reference: To preserve a superseded project or version of a project for future reference, archive it to a permanent location, then delete it from the working folder.

What is and What is Not Archived?

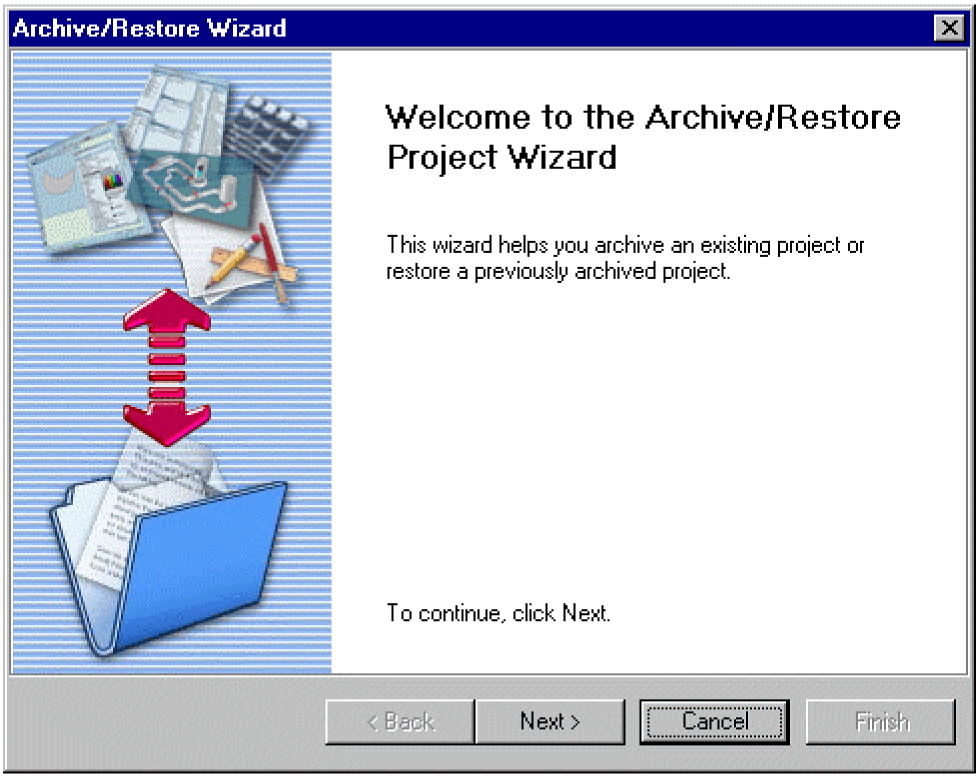
The archive process archives all sub-folders and files in the selected project folder including the Vijeo Look project file (.vlp).

The archive process does not archive any files outside of the selected project folder. Therefore database files, ActiveX controls and Global Libraries are not archived.

The Archive/Restore Wizard

The Wizard prepares your instructions for creating or restoring an archive in such a way that you can specify and revise your options up to the point of running your request.

- To go back to the previous step at any point, select the *Back* button.
- To cancel your request, select *Cancel*.
- To proceed to the next step, select *Next*.



Archive/Restore Wizard

Welcome to the Archive/Restore Project Wizard

This wizard helps you archive an existing project or restore a previously archived project.

To continue, click Next.

! Be careful not restore a project – thus overwriting it - instead of archiving it, since the processes are quite similar.

Creating an Archive

You can start the Archive/Restore Wizard either directly from Windows or from within the HMI:

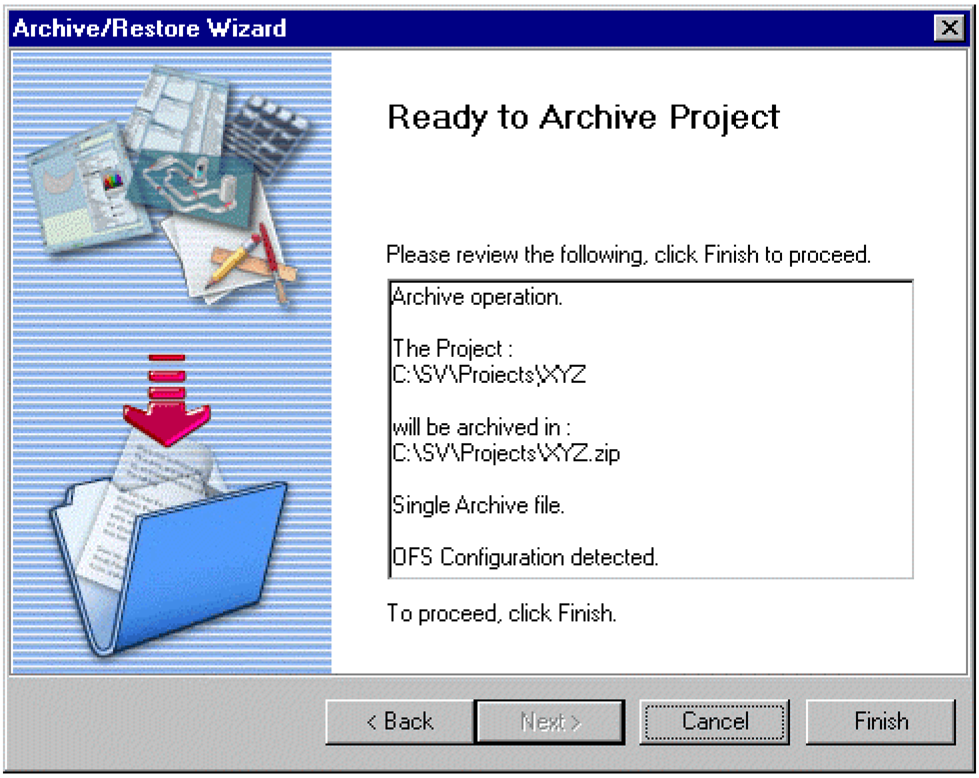
- From Windows Explorer, start Archive – Restore Project.
- from the HMI menu select **Tools.Project.Archive/Restore**.

When the Archive/Restore Wizard starts, select the Next button and follow the instructions.

How to Run the Archiving Process

The Wizard will guide you through the following steps.

1. Select Operation - Select the option *Archive an existing Project*.
2. Select Origin - Adjust or replace the path to the folder of the project you wish to archive, or select it by using the browsing button.
3. Select Destination - Pick the location to which you wish to archive your project. A drop-down box allows you to create an archive arranged in one of three ways:
   In a single file
   Split across files
   Split across floppy disks.
4. Browse - You may select an existing folder or create a new one. If you specify an existing .zip file, a message warns you that you are about to overwrite it.
5. Select Next to review your archiving options.
6. Finish - To run the archiving process, select *Finish*. The tool will copy your project to the Archive you specified and confirm successful completion.



Archive/Restore Wizard

Ready to Archive Project

Please review the following, click Finish to proceed.

Archive operation.

The Project :
C:\SV\Projects\XYZ

will be archived in :
C:\SV\Projects\XYZ.zip

Single Archive file.

OFS Configuration detected.

To proceed, click Finish.

Selecting a folder that is not a Vijeo Look project will cause erroneous operation of the Archive/Restore wizard and possible loss of data.

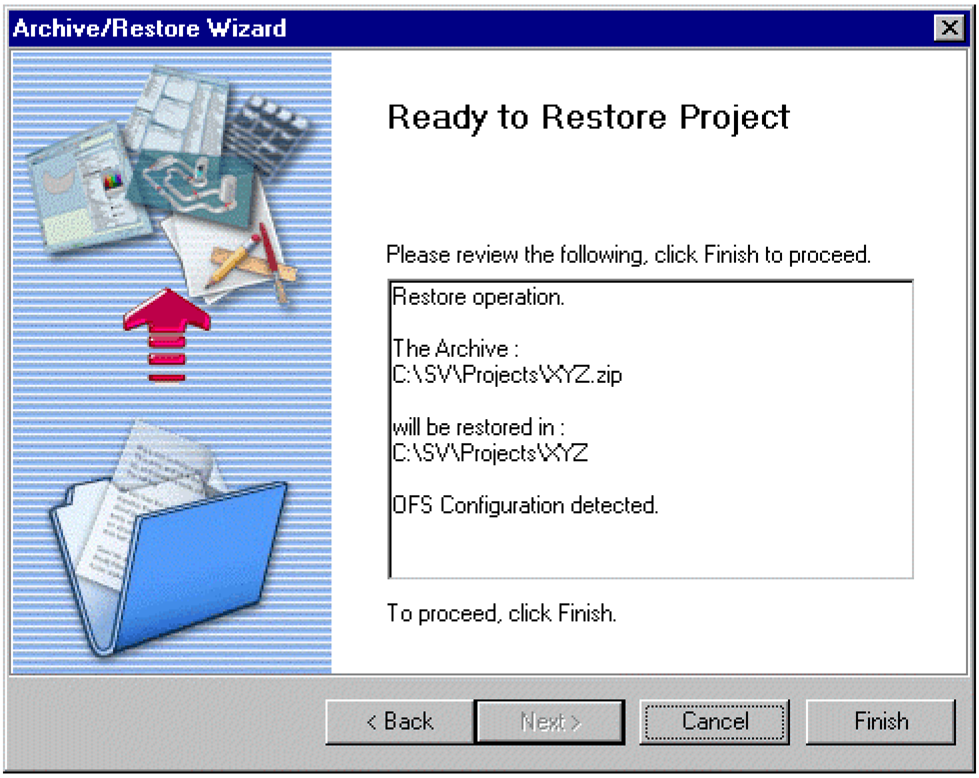
Restoring a Project

To restore a project from an Archive, start the Archive/Restore Wizard (Tools.Project.Archive/Restore...). Select the Next button to start the Wizard and follow its instructions.

How to Run the Restore Process

To restore a project, the Wizard works in a converse manner to that of archiving. It allows you to restore the Project to the folder from which it came, or to another location.

1. Select Operation - Select the option *Restore an archived Project*.
2. Where is the Archive you want to Restore? - Enter the path to the Archive, or use the Browse button to locate it.
3. Where would you like the Project to be Restored? - Enter the path to the Archive, or use the Browse button to locate it.
4. Finish – Review the options you have chosen, then run the archiving process by selecting *Finish*. The Restore tool will copy your project to the folder you specified and confirm successful completion.



Ready to Restore Project

Please review the following, click Finish to proceed.

Restore operation.

The Archive :
C:\SV\Projects\XYZ.zip

will be restored in :
C:\SV\Projects\XYZ

OFS Configuration detected.

To proceed, click Finish.

❗ When you restore a project, take care that you do not unintentionally overwrite a more recent version of the project.

After the restore process you must stop and start the OFS server to allow any changes in configuration to be taken into account.