



Supervision and control XML-based  
from Windows 7 to Windows CE

# **Technical Specifications**

---

Version 11.3 - Ed. Feb. 2012

Cod. DOCS 11 LC Build 1101

# Table Of Contents

<b>1. TECHNICAL SPECIFICATIONS .....</b>	<b>5</b>
1.1. GENERAL CONCEPTS.....	5
1.1.1. Developing & Running Applications.....	5
1.1.2. Installing & managing applications.....	6
<b>2. ARCHITECTURE.....</b>	<b>7</b>
2.1.1. System Architecture.....	7
2.1.2. Project Architecture.....	8
2.1.3. Data Recording Architecture.....	9
<b>3. PROJECT RESOURCES .....</b>	<b>13</b>
<b>4. SHORTCUTS (KEYBOARD COMMANDS).....</b>	<b>17</b>
4.1.1. Restrictions.....	17
Speech Recognition.....	18
4.1.2. Restrictions.....	18
<b>5. TAG DATABASE .....</b>	<b>19</b>
Data Types.....	20
Local Variables.....	20
System Variables.....	20
Variables In Use.....	21
Pointing to Variables.....	21
5.1.1. Restrictions.....	21
<b>6. COMMUNICATION DRIVERS.....</b>	<b>23</b>
6.1.1. Restrictions.....	24
<b>7. HMI SCREENS .....</b>	<b>25</b>
7.1.1. Opening Screens.....	26
7.1.2. Restrictions.....	27
Interface Graphics.....	27
Control and Object Graphics.....	29
7.1.3. Restrictions.....	30
Command Lists assignable to objects.....	30
7.1.4. Restrictions.....	31
Symbol Library.....	31
7.1.5. Graphics capability restrictions.....	33
<b>8. ALARM MANAGEMENT .....</b>	<b>35</b>
8.1.1. Restrictions.....	37
<b>9. EVENTS HISTORICAL LOG.....</b>	<b>39</b>
9.1.1. Restrictions.....	40
<b>10. ALARM DISPATCHER.....</b>	<b>41</b>
10.1.1. Restrictions.....	42
<b>11. ALARM STATISTIC .....</b>	<b>43</b>
11.1.1. Restrictions.....	44

<b>12. DATA LOGGERS .....</b>	<b>45</b>
12.1.1. Restrictions.....	46
<b>13. RECIPES .....</b>	<b>47</b>
13.1.1. Restrictions.....	48
<b>14. TREND AND DATA ANALYSIS .....</b>	<b>49</b>
14.1. TREND AND DATA ANALYSIS .....	49
14.1.1. Restrictions.....	51
<b>15. IP CAMERA VIEWER .....</b>	<b>53</b>
15.1.1. Restrictions.....	53
<b>16. TRACING AND AUDIT TRAILS.....</b>	<b>55</b>
16.1. TRACING & AUDIT TRAILS.....	55
16.1.1. Restrictions.....	55
<b>17. REPORTS.....</b>	<b>57</b>
Report Designer .....	57
17.1.1. Restrictions.....	57
Report Engine Crystal Report™ .....	58
17.1.2. Restrictions.....	58
Textual Reports.....	58
17.1.3. Restrictions.....	58
<b>18. EVENTS .....</b>	<b>59</b>
18.1.1. Restrictions.....	60
<b>19. SCHEDULERS.....</b>	<b>61</b>
19.1.1. Restrictions.....	62
<b>20. SCALING OBJECT LIST .....</b>	<b>63</b>
20.1.1. Restrictions.....	63
<b>21. VBA BASIC SCRIPT.....</b>	<b>65</b>
21.1.1. Restrictions.....	66
<b>22. IL LOGICA .....</b>	<b>67</b>
22.1.1. Restrictions.....	68
<b>23. STRINGS .....</b>	<b>69</b>
23.1. STRING AND LANGUAGE CHANGE .....	69
23.1.1. Restrictions.....	70
<b>24. SECURITY (USERS AND PASSWORDS) .....</b>	<b>71</b>
24.1. SECURITY (USERS AND PASSWORDS).....	71
Protection Criteria.....	72
24.1.1. Restrictions.....	73
<b>25. CHILD PROJECTS .....</b>	<b>75</b>
25.1.1. Restrictions.....	76
<b>26. NETWORKING .....</b>	<b>77</b>
Network Server .....	77
Network Client.....	78
26.1.1. Restrictions.....	78

**27. REDUNDANCY ..... 79**  
    *Advanced Redundancy Functions* ..... 80  
    27.1.1. *Restrictions* ..... 80

**28. OPC CLIENT & SERVER..... 81**  
    *OPC Client* ..... 81  
    *OPC Server*..... 82  
    28.1.1. *Restrictions* ..... 82

**29. WEB CLIENT ..... 83**  
    29.1.1. *Restrictions* ..... 84

**30. MOVICON FOR WINDOWS CE..... 85**  
    *Programming Constraints*..... 86

**31. DEBUGGER..... 89**

**32. CROSS REFERENCE ..... 93**

**33. INTEROPERABILITY..... 95**

**34. SYSTEM TOOLS ..... 97**

# 1. Technical Specifications

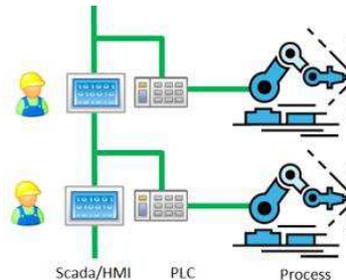
---

## 1.1. General Concepts

---

**This documentation provides a general overview on the fundamental Movicon 11 concepts. The principal components of the system's architecture and how to use them are all described within. More in-depth information can be found in the Programming Manual.**

Movicon is a SCADA/HMI platform designed to make the task of collecting real-time data from any plant or automation system, represent it graphically with man-machine interaction (Human Machine Interface), with complete process event, recording and data analyzing management, simple and pleasurable.



The Movicon supervision software has been designed to guarantee maximum deployment flexibility, consenting the developing and running of supervision projects in systems based on PC (Windows 32/64 bit) architecture and embedded or mobile (WinCE) systems, interfacing with any control device or instrumentation (PLC, Remote I/O, Control Systems, etc.).

### 1.1.1. Developing & Running Applications

---

Creating graphical interfacing and supervision projects is very easy and intuitive using the Movicon Editor with object oriented programming techniques, and easy-to-follow resource and object function property settings.

Each project resource is saved in persistent mode on XML file (encryptable) to guarantee openness to the most advanced programming techniques.

Projects created with Movicon in Editor mode, can be run by Movicon in runtime mode and as Windows service to allow system startup and execution of all the specific functions defined in design mode.

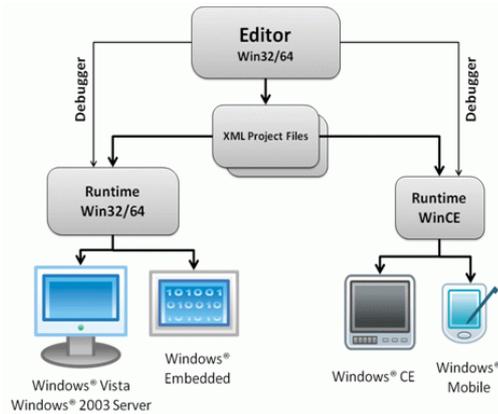


If not specified differently, Movicon is run with the **Movicon.exe** program, which provides both Editor and Runtime modes for Win32/64 systems. The **MoviconRunTime.exe** program, from the same installation folder, can be used for Win32/64 and consents the one execution of Movicon created projects in runtime. The **MoviconService.exe**, also found in the same installation folder, allows projects to be run as Windows service.

**MovCE.exe** is the runtime engine for Windows CE and consents to the one execution of project created in embedded devices based on Windows CE.



The project runtime module is available for both Windows 32/64 bit and WinCE systems.  
 The Debugging functions are available from the Windows 32/64 bit Editor for all the runtime systems.

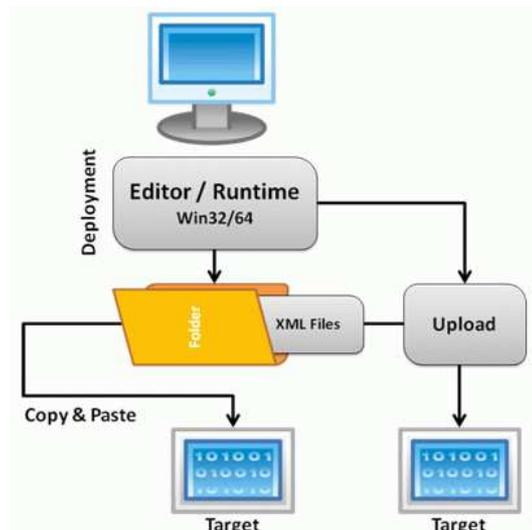


### 1.1.2. Installing & managing applications

When a Movicon project is created, you can specify the project file destination folder which will contain all the XML files (or encrypted) constituting the created project's resources. Movicon also uses subfolders to contain project data.

After having created your specific Movicon application, you can then manage project files in the following ways:

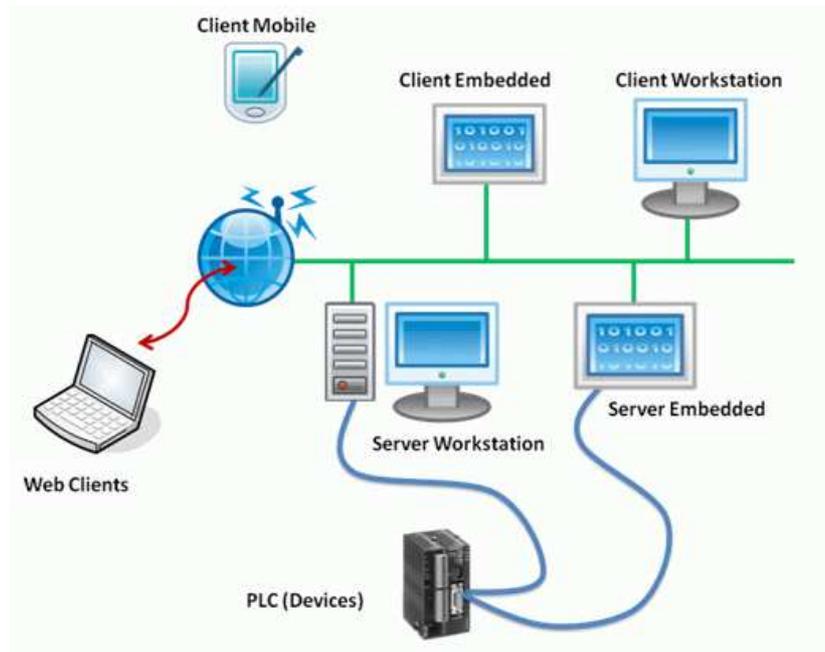
- By starting up in runtime mode from design mode, the created project files residing in the specified project folder will be put into execution.
- By installing the project on another PC by using the project's 'upload' command towards a target installed with Movicon beforehand. The upload command is available from a menu which shows when selecting the project from within the project explorer window. The command window includes commands for starting up or aborting the project on target after its transferal. Remember that target must be installed with Movicon before in order for this to work.
- By copying the project folder to desired target using the Windows techniques, and then using the local startup project techniques.



## 2. Architecture

The Movicon platform architecture is built with a software structure capable of generating XML format based project files, using a simple and intuitive approach involving object-oriented programming.

The generated projects (XML files) can be run directly from the design mode environment (in this case all the Debug functionalities are automatically active) or they can be executed from runtime executables for both Win32/64 and WinCE.



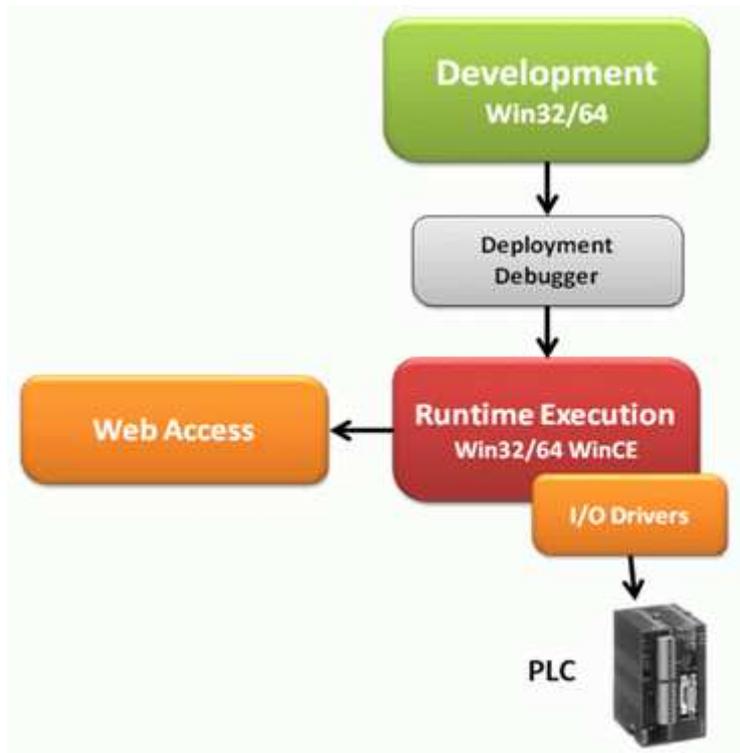
All Movicon projects can be run in the following modalities:

<b>Win32/64</b>	<b>Server</b>	<ul style="list-style-type: none"> <li>• Tags connectable to external devices</li> <li>• Networking Server/Client</li> <li>• All functionalities run locally</li> </ul>
<b>Win32/64</b>	<b>Client</b>	<ul style="list-style-type: none"> <li>• Networking Server/Client</li> <li>• All functionalities run locally</li> </ul>
<b>WinCE/Mobile</b>	<b>Server</b>	<ul style="list-style-type: none"> <li>• Tags connectable to external devices</li> <li>• Networking Server/Client</li> <li>• All functionalities run locally</li> </ul>
<b>WinCE/Mobile</b>	<b>Client</b>	<ul style="list-style-type: none"> <li>• Networking Server/Client</li> <li>• All functionalities run locally</li> </ul>
<b>Win32/64 WinCE Linux, ecc.</b>	<b>Web Client</b>	<ul style="list-style-type: none"> <li>• Using browser (on Client) to manage project running on server (Server Win32/64 or Server WinCE).</li> </ul>

### 2.1.1. System Architecture

The Movicon programming and developing environment (IDE) has been purposely designed to make project creating and managing simple.

The programming environment is totally integrated and does not require you to install or use external components for managing all the Movicon product's functionalities.  
The Movicon system philosophy is to provide an all-in-one environment for all the different functionalities and features, to reduce application development times and maintenance.



The development environment (Design Mode) produces projects based on open architecture (XML). The project interpreter is called Runtime which executes the projects created in the development environment.  
The XML project structure (project files can be encrypted) is an open structure to allow expert users the use of Macro or script Functions for auto-generating projects or parts of them. Project can be put into Runtime directly from the development environment (Design Mode) or from the runtime module.



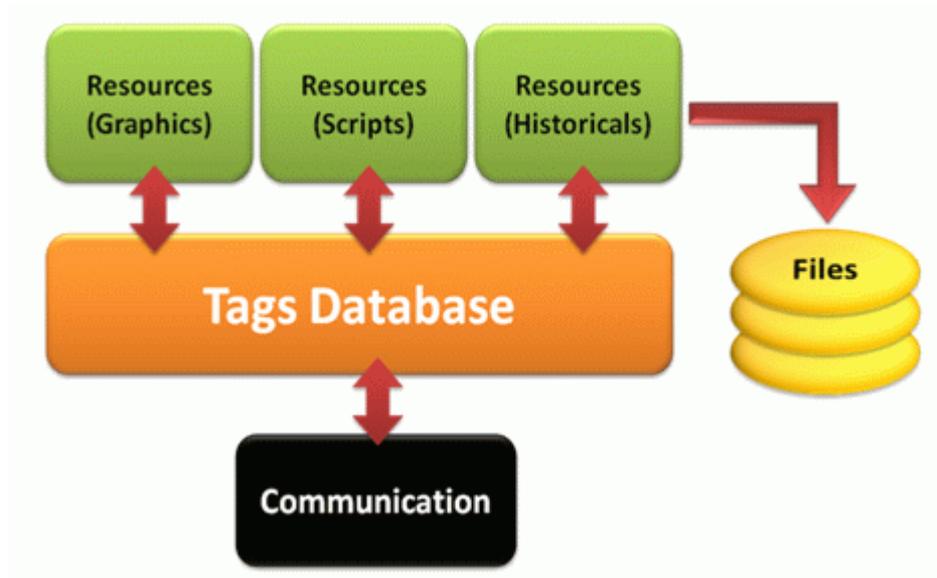
During runtime execution all the project functionalities will be activated, including communication with devices and historical data recording.

### **2.1.2. Project Architecture**

---

The Movicon platform is based on architecture capable of generating supervision projects according to an architecture based on the Real Time Database, the central project resource and main system information source.

The Real Time Database (Tag Database) manages real-time communication with the field devices, and notifies all project resources (or receives from them) on event value changes.



This block diagram outlines the Movicon platform architecture and structure.

The development environment is used for creating projects using the standard Windows™ editing techniques to insert the desired resources needed in the project, where each resource contains its relative objects.

All objects and resources have access to the Tag Database, being the gathering point of all the project's "real-time" information.

Project structures are displayed clearly within the "Project Explorer" window in a tree structure listing all the project resources.

Each resource is configurable through its respective properties, and can contain specific objects, which are inserted and configured through their own respective properties as well.

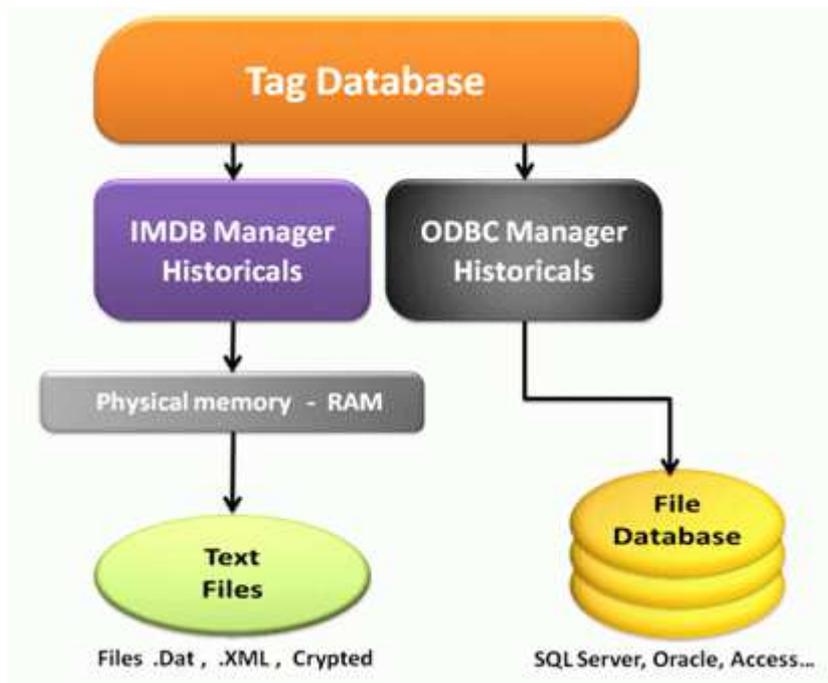
```

Project ▶
  Resource ▶
    Objects ▶
      Properties
  
```

### 2.1.3. Data Recording Architecture

The Movicon platform records data on files using an architecture which provides three types of recording engines to do this job:

- **Events Historical Log:** records all project and system events (i.e. Alarms, Messages, etc.).
- **Data Loggers/Recipes:** records process data archives.
- **Tracing/Audit:** records events and information linked to relative variable status changes.



Data recording on file can be managed according to these two different write on file modalities which can be set as pleased:

- **ODBC Manager**
- **IMDB (In Memory DB)**



In the end the basic use of historical logging does not change. In both cases Movicon will record data on files whose contents will, nevertheless, be displayed automatically in the Log window. However, the standard ODBC offers more advantages in using a data base in Relational DB. It has more features in terms of being able to perform analysis and statistics due to being able to use the SQL language.



Each resource provided with data recording facilities can be free configured according to the predefined modality chosen and simultaneously with different modalities. For instance, in one project one Data Logger can record by means of IMDB and another by means of ODBC.

- The ODBC Manager samples and buffers data to download into Relational Database file tables in predefined DB formats (i.e. SQL Server, Oracle, MySQL, etc.). This modality is used for default in Win32/64 systems.
- The IMDB manager samples data, stored in appropriated RAM areas for faster data accessing and saving on file in persistent mode (.DAT, XML or encrypted) with prefixed time intervals. This modality is used for default in WinCE systems.

All the data log tables created by Movicon have the four same columns:

- Time Column (TimeCol): shows date and time in GMT recording took place
- Local Time Column (LocalCol): shows the local date and time recording took place
- MSec Column (MSecCol): shows recording time in milliseconds
- User Column (UserCol): if recorded event was triggered by a user (by using a button etc.), the name of the user logged in the project at that moment is recorded in this field.

## ODBC Manager

The Open Data Base Connectivity (ODBC) is a Microsoft standard that allows applications to organize data in database files in the specified format of any Relational Database enabled with this standard.

The Movicon ODBC Manager consents to automatic historical file logging in the preferred database format set by the user.



In order for Movicon to record historical data on file in ODBC standard format, the Windows OS being used must be setup with the appropriate ODBC drivers suitable for the chosen database product. Movicon, during the normal installation phase, will install the files needed for the ODBC management with the right drivers needed for MsAccess™.

While creating a new project, using the wizard ("Database settings (ODBC)"), the user will be asked to choose a database format for default. The available options are:

- **MSAccess:** MS Access (uses the "Microsoft Access Driver (\*.mdb)" driver. This is the predefined default choice)
- **MySQL:** MySQL 5.1 (uses the driver installed with this version of MySQL)
- **OracleEx10:** Oracle Express (uses the client-less driver installed with this version of Oracle)
- **SQLServer2000:** MS SQL Server 2000 (uses the "SQL Server" driver)
- **SQLServer2005:** MS SQL Server 2005 (uses the "SQL Native Client" driver)
- **SQLServer2008:** MS SQL Server 2008 (uses the "SQL Native Client 10.0" driver)
- **SQLServerExp2000:** MS SQL Express 2005 (uses the "SQL Native Client" driver)
- **SQLServerExp2008:** MS SQL Express 2008 (uses the driver "SQL Native Client 10.0" driver)

Once the project has been created using the wizard, the plugin chosen for default, (through the "Database settings (ODBC)" window) will be indicated in the project's "Default ODBC Connection Plugin" property which can be changed afterwards if need be.

## IMDB (In Memory DataBase)

The IMDB (In Memory DB) for managing historical logs allows you to save data in XML or encrypted text format.

The Movicon historical log management can be performed using the IMDB manager which allows you to save process data in XML format. This feature can be used as an interesting alternative to the ODBC Manager, in order to get historical data stored in XML format text structured archive. In this case, however, even though Relational Database formats cannot be used, you will gain in being able to record data in embedded systems lacking Relational Database support, and having the option to encrypt data in binary propriety format.

Another advantage of not needing the ADOCE component and SQL Server when using IMDB in devices based on Windows CE can be seen in a noticeable improvement in performances. This modality is proposed especially for default by Movicon when creating projects for Windows CE.



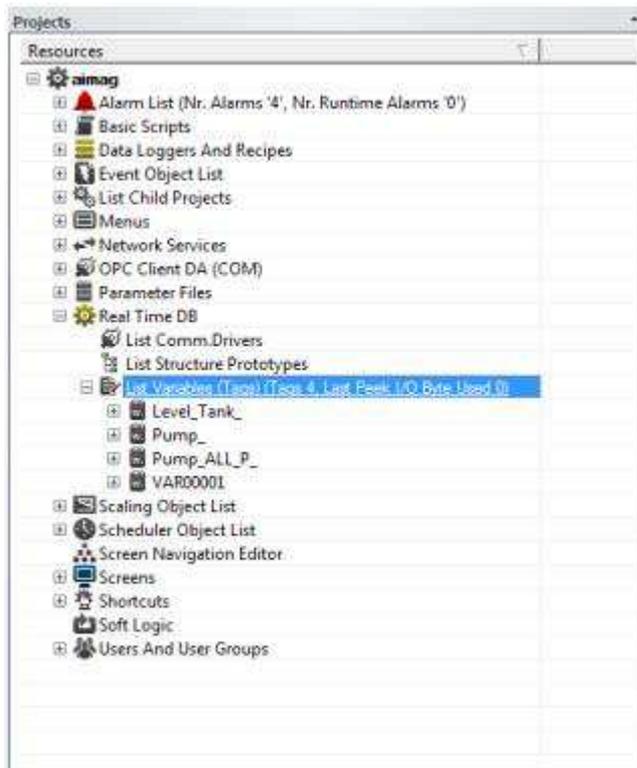
When using the IMDB manager for recording historical data you must take into account the fact that it does have some restrictions which relational databases do not have. For instance, the IMDB cannot be used for recording large amounts of data, above all in systems using Windows CE, when keeping data tables completely stored in RAM.



# 3. Project Resources

The Movicon project resources are containers of objects or function modules provided by each resource.

These resources are listed in the main project structure, and each resource is technically built with an XML file, saved in persistent mode in the project folder.



Each resource can be accessed and configured in Project edit mode through the **Project Explorer** window (as seen above). Each resource, according to function type, can be configured through its **Properties Window** and can contain specific elements and objects which are also configurable through their relating properties.

## Resource Table:

<b>Shortcuts (Accelerators)</b>	This resource allows you to define commands to activate in association with PC keyboard keys or Speech Recognition.
<b>Basic Script</b>	This resource allows you to define VBA™ script code to be executed on command or event.
<b>Data Loggers/Recipes</b>	This resource allows you to define objects to manage data recordings according to these two modalities: <ol style="list-style-type: none"> <li>1. Recipes (data archives associable to archive management and selection user interfaces)</li> <li>2. Data Logger (process data recording engine for creating databases for Trends, Reports or Tables).</li> </ol> <p>NB: The Data Loggers feature is a license option</p>
<b>Parameter File</b>	This resource allows you to defined files (simple Unicode text files) which can be used for screen parameterization

	(variable Alias).
<b>Alarms</b>	This resource allows you to define alarm or message lists to be managed in the project
<b>Event Objects</b>	This resource allows you to define objects which activate commands on events associable to Variables or other commands (i.e. buttons).
<b>Scaling object lists</b>	This resource allows you to define value conversions including non linear, from "raw" values to scaled values. NB: linear value conversions can also be defined directly in the Variable properties.
<b>Scheduler object lists</b>	This resource allows you to define objects that activate commands according to prescheduled dates and times. Commands can be scheduled to activate at fixed times in the scheduler properties or can be set in runtime through the 'Hour Selector' object window.
<b>Child projects</b>	This resource is used for defining the project's external Movicon projects in design mode, whose contents are linked and accessed from the project (Parent project).
<b>Menus</b>	This resource is used for defining configuring "menu" objects for used for activating command listed in pop menus or system menus (custom menu bars).
<b>Screen Navigation Editor</b>	This tool is object oriented to permit you to edit navigation project screen connectors visually. The navigation buttons are created automatically in runtime.
<b>OPC Client</b>	This resource permits you to define, in 'static' mode, connections between project variables (Tags) and OPC Server (OPC DA specifications) being the realtime data source origin. NB: These connections can also be defined directly in the variable properties, according to the dynamic connection modality used.
<b>Real Time DB</b>	This is the project's central resource in which you can define: <ul style="list-style-type: none"> <li>• List Variables (Tags). This list can also be created by importing data from the PLC according to the I/O driver inserted in the project.</li> <li>• List Structure prototypes: data prototypes can be defined on this list to structure definition.</li> <li>• List Comm. Drivers: Communication Drivers to be used for managing connections to field devices are inserted and defined in on this list.</li> </ul>
<b>Network Services</b>	This resource allows you to set or modify any Movicon Server or Client network communication configurations. You can also use this resource to configure the <b>Redundancy</b> feature (optional).
<b>Screens</b>	This resource is used for inserting and configuring Screen windows used for managing user interfaces and the project's supervision graphics.
<b>SoftLogic</b>	This optional resource permits you to access Logicon, the IEC61131 programming environment. This product requires a separate installation and a separate user license.
<b>Users and User Groups</b>	This resource is used for defining project Users and Password protection and security activation management.

**Project Functions Table:**

<b>String Table</b>	This resource permits you to define and centralize all the text strings that can be used in the project. This table is divided into columns, each one corresponding to a language. Externally, each text column corresponds to a XML text file.
---------------------	---

	The string table supports copy&paste directly in Ms Excel or in any text editor such as Notepad.
<b>Cross Reference</b>	This resource allows you to create documentation on how all the variables are used within the project by producing a Cross Reference table indicating where the variables are used and what has been associated to them.
<b>Upload/download File</b>	This tool is used for uploading or downloading project files to and from target devices (i.e. embedded or remote devices). In addition, it allows you to use remote command (i.e. remote project launch/abort, debug, etc).
<b>License Managment</b>	Tool is used for displaying and analyzing project license requirements and/or installed license typology.
<b>Win32/64 Services Management</b>	Automatic activation or deactivation tool of project in "Windows Service" mode.
<b>Check Missing Strings</b>	Very handy tool for finding strings that have not been defined in the String Table and see how they are used within the selected resource. When missing string is found this tool will automatically create and ID string and add it to the String Table.
<b>General IL Logic</b>	The project's general IL Logic resource. This resource permits you to insert code using the combinatory logic language (AWL or IL Instruction List) which will automatically be run in project background at startup.
<b>Symbol Library</b>	This library contains prebuilt symbols that can be inserted into screens. Programmers can add to this library with their own custom made symbols.

**Local Functions Table:**

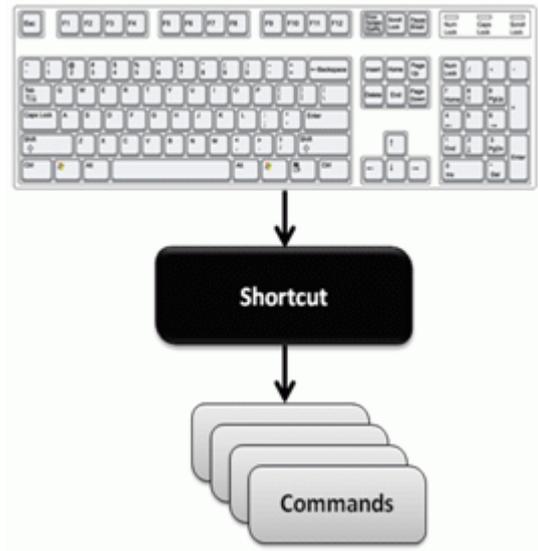
<b>Local IL Logic</b>	Local Logic resource relating to the <b>Screen</b> resource or the <b>Object</b> of the screen selected in the project. Code in the Combinatory logic language (AWL or IL Instruction List) can therefore be inserted and run only when the selected resource is active in runtime. Associated code will become part of the object it was inserted in.
<b>Local Script Logic</b>	Local Script Logic Resource, relating to the <b>Screen</b> or Screen's <b>Object</b> resource or the <b>Alarm Threshold</b> selected in the project. By using this resource you can insert code with the VBA™ language to be run in function with the event selected for the object. Associated code will become part of the object it was inserted in.
<b>XML Code Explorer</b>	Window used for exploring the selected project's object's or Resource's XML structure. This window can be used for displaying and/or modifying the object using a text editor directly in the object's XML structure.
<b>Refactoring Explorer</b>	Window used for the Movicon Refactoring feature. By using this window you can display and/or activate the Refactoring commands used for checking and correcting assigned name errors (Tag or resource) in the selected resource.
<b>System Output</b>	System Log Window, divided into sub-window for output data aggregations. This window can be used for displaying Log files automatically produced by Movicon relating to commands, errors, operations performed or occurred in project. These text files are recorded in the appropriated project folder. This is a very important debug tool which reports all errors that took place in the project during runtime.



# 4. Shortcuts (Keyboard Commands)

---

The Shortcut resource allows you to define command activation in association with keys from the pc keyboard or speech recognition.



Example:

<b>Shortcut1</b>	▶	<b>F1 Key</b>	▶	<b>Command 1</b>
		<b>ESC Key</b>	▶	<b>Command 2</b>
		<b>CTRL+F2 Key</b>	▶	<b>Command 3</b>

## Main Features

---

- Each Shortcut contains one or more keys (combo keys) that can execute a command or a list of commands.
- Shortcuts activate in relation to the name of the Screen Window. This relationship is established by the same <name> used by the Shortcut resource and the Screen. For instance, when activating the <Main> screen, the system will search for an existing <Main> shortcut. If found, the shortcut will activate. If one does not exist, any previous shortcut used will remain active. If the screen set as the "Startup Screen" has a shortcut, this shortcut will become the default shortcut for all screens that do not have one. In this case, the startup screen's shortcut will be activated when the screen being loaded doesn't have one.
- Shortcuts can also be associated to screens opened as Modal Windows (pop-up).
- Shortcuts manage command executions (even of different types) both upon **pressing down** the key or **releasing** it.
- Each shortcut can be associated **Speech Recognition** activation. The system therefore permits command activation based on recognizing a vocal command, with confirmation in text if appropriate (Only for Win32/64).

### 4.1.1. Restrictions

---

The Shortcut resource limits for inserting shortcuts in projects are defined in the below table:

<b>System</b>	<b>Restrictions</b>
Win32/64	Max 4096 shortcuts per project
WinCE	Max 128 shortcuts per project

## Speech Recognition

---

Movicon provides users with a very handy feature that recognizes spoken texts associated to the Shortcuts resource for activating commands vocally.

In order to use the speech recognition function for activating commands you need to first check whether your PC has been fitted with a microphone for giving vocal commands. Your PC will also need an audio card to allow the system to enounce request and confirm messages.

### Main Features

---

- Vocal commands can be associated and configured in the Shortcut Resource.
- Each command has been predisposed with text for eventual vocal "Confirm requests" or command text repetition for "Don't understand text".
- Vocal commands can be set with set points.

### 4.1.2. Restrictions

---

Systems which support the Speech Recognition feature are:

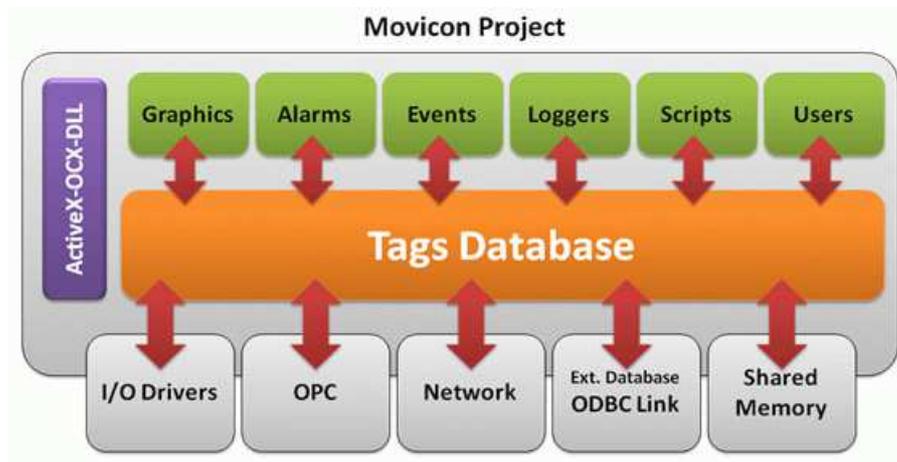
<b>System</b>	<b>Restrictions</b>
Win32/64	Only the "Microsoft English Recognizer v5.1" speech recognition in English is available at the moment and cannot be used with the Speech Recognition feature for other languages.  The Speech Recognition is also only available for Windows Vista/7 for the following languages: English (U.S.), English (U.K.), German (Germany), French (France), Spanish (Spain), Japanese, Chinese (Traditional), and Chinese (Simplified).
Win32/64	Max. 128 Vocal Commands can be inserted in one project.
Win32/64	Max. 64 characters for each single Vocal Command
WinCE	Not available
Web Client	Not available

# 5. Tag Database

---

The project variables, also called Tags, are defined in the List Variables (Tag) contained in the Real Time DB resource forming part of the project structure. The logic variables form and provide information (deriving from digital or analog values) which, combined together, consent to managing the Movicon program. The Movicon real-time database is based on extremely efficient and optimized technology. The Movicon platform kernel is based on an innovative system technology that, though based on XML, is totally managed in event-driven mode in runtime with very efficient thread pooling management.

The Movicon variables can be exchanged with the field using different modes, by means of Communication Drivers, OPC, Networking, ODBC, shared memory or using ActiveX or customized libraries:



The Movicon Tags have numerous properties and functionalities, making each tag a real operating centre capable of satisfying a vast range of system project engineering needs:

## Main Features

---

- Possibility to customize variable name and description.
- Possibility to define variable data type: bit, byte with and without sign, DWord with and without sign, Float, Double, String, Byte Arrays.
- Possibility to structured variables using structure prototypes defined by programmer
- Possibility to render variables retentive (value remains when turned off).
- Possibility to activate variable Statistic Data for obtaining variable value minimums, maximums and averages, plus total time On.
- Variable quality management through which variables can be controlled to see whether updated correctly by drivers or not.
- Possibility to associate variables with initial value, formats and default engineering units.
- Possibility to import variables from the PLC database when this feature is made available by communication driver
- Possibility to scale variable values (automatic data conversion).
- Possibility to directly associate variables with field addresses to connect to (driver, opc or networking).
- Possibility to activate variable Tracing or Audit Trail, a Movicon feature that consents historical logging of variable behaviour, by recording on Database or text files of each variable change and the reason causing it.
- Possibility to rename variables without having to then replace the old variable names in various parts of the project in which they are used.
- Possibility to create local screen variables. Local variables are initialized only when the screen is loaded in memory and they can only be used within the context of that screen

- Possibility to connect directly (read-write) to external relational database tables via ODBC Manager.
- TimeStamp Management
- Cross Reference for documenting variable uses in project.

### Using Variables

---



Movicon variables can be used by any the project's resources or objects, in animation properties, Basic Script code and IL logic. They can also be used for generating alarms or alert messages, or triggering events upon changing their values.

Each single child project variable bit can be accessed from the parent project and viceversa.

## Data Types

---

All variables (Tags) can be defined in the properties as one of the following data types:

- Bit
- Byte with sign / Byte without sign (8 bit)
- Word with sign / Word without sign (16 bit)
- Doubleword with sign / Doubleword without sign (32 bit)
- Float (32 bit single precision)
- Double (long) (64 bit double precision)
- String (each character one byte + termination character 0)
- Byte array (fixed length with starting address and byte number)
- Structure: A list of Structure types available will be listed if in the 'Structure Prototypes' group.

## Local Variables

---

Movicon screens can contain local variables. The local variables are initialized only when the screen is loaded in memory and can be used only inside the context of that screen. They cannot be retentive and cannot be used for communications or in scripts. In addition, they do not have all the properties which the global variables have.

## System Variables

---

Movicon has a range of prefixed variables to enable programmers to interact with the project and with the system, facilitate information availability and managed commands by exploiting logic. The System variables are grouped into a variable **structure** type. To use the System Variables you first need to insert the variable structure by using the 'Add a System Variable' command in the "Project Explorer" window or by right mouse clicking on the 'Real Time DB' resource. By executing this command the structure prototype and the relative variable called "\_SysVar\_" will be inserted. Each member of the structure variable has a precise significance as described in the table found in the user's manual.

## Variables In Use

The Variables InUse Management property allows you to optimize performances. A sophisticated architecture consents variables effectively in-use to be acknowledged by the system and exchanged with the field and devices if needed.

In this way the system's overall performances can be increased to the project's user's best advantage. The InUse Variable management is enabled for default but can be disabled by means of using the RealTime DB's InUse Management property.

## Pointing to Variables

Movicon also consents variables to be pointed to in the following ways:

Pointing to bit	Any one of a variable's bits can be pointed to by specifying the bit number after the variable's name separated with a dot: <tag_name>.<bit nr > i.e. WORD0001.15
Pointing to structure members	Any one of a structure variable's members can be pointed to by specifying the member's name after the variable's name separated with a colon: <tag_name>:<member_name> i.e. STRUCT0001:LOW

### 5.1.1. Restrictions

The restrictions to take into consideration when defining variables are:

Restrictions	
Maximum number of variables	Max. 128,000 declared variables (Tags) in one single project for Win32/64 and 4096 variables for WinCE. This includes basic variables or Structure Members or Array elements. Projects with higher quantities for Win32/64 can be managed in function with hardware resources available or throughout the project resources to the user's discretion and full responsibility. (Large numbers of Tags can be distributed in several projects declared as "Child Projects", in a distributed PC network).
Variable Names	<ul style="list-style-type: none"> <li>• Variable names cannot contain special chars other than "_\\"</li> <li>• Variable names cannot begin with a special char or a number</li> <li>• Variable names cannot be more than 64 chars long</li> </ul>
Structure Variable Size	Structure variables cannot exceed sizes of 32767 bytes.
Maximum number of Retentive Variables	A max. of 16,000 retentive variables per project. You need to take into consideration that Movicon saves the states of each retentive variable on XML file. Each Retentive Variable value change is written on this file for every Tag. Excessive use and indiscriminate use may compromise system performances and memory space (hard disk or CF).
Maximum number of variables connectable in network	A max. of 32,000 variables (Tags) can be connected to other Movicon projects through a network for Win32/64 and 512 variables for WinCE
Maximum number of variables with statistics	A max. of 2048 Variables can be used with statistics enabled per project. You will need to take into consideration that in order to calculate data statistics, Movicon saves each statistics enabled variable states in the appropriate permanent memory areas. An excessive use and discrimination may compromise system

## TECHNICAL SPECIFICATIONS

---

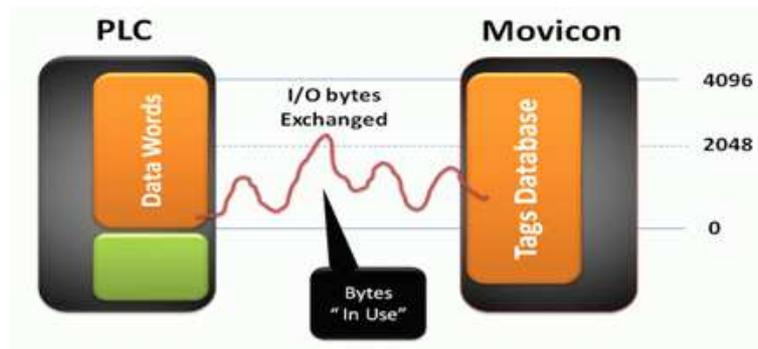
	performances and memory space (hard disk or CF).
Number of Variables connected via ODBC	Win32/64: Max. of 2048 variables (Tags) connected to external DB tables (read-write) using the ODBC I/O Link properties.  WinCE: Max. Of 64 variables (Tags) connected to external DB tables (read-write) using the ODBC I/O Link properties.
Number of Variable in Tracing-Audit	Win32/64: Max. of 2048 variables (Tags) with Tracing function enabled (Audit Trail).  WinCE: Max. of 64 variables (Tags) with with Tracing function enabled (Audit Trail).

## 6. Communication Drivers

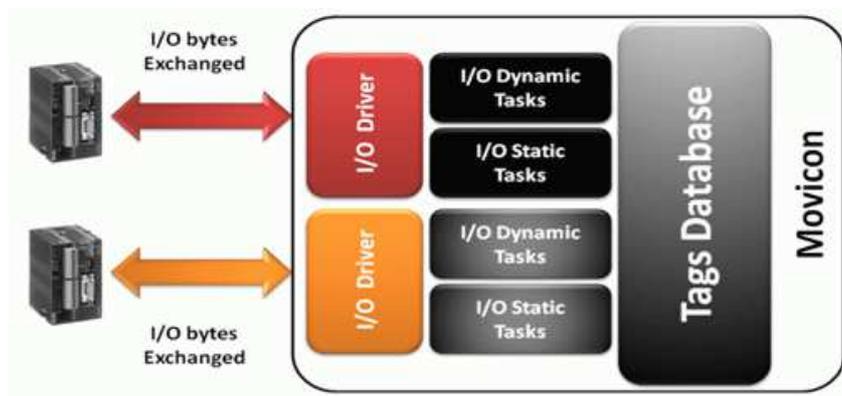
---

The Movicon Communication Drivers are built from dynamic libraries linked to the platform but work independently (.DLL files) that, according to "exception-based" logic, notify information received from the memory of devices connected to the Movicon memory areas and viceversa, according to predefined settings.

Associations between field variables and Movicon variables are established through the Driver's settings. The system, by means of using Ethernet, serial ports, fieldbuses or specific software libraries, will read and write variables "from" and "towards" the plant system, according to the modalities set.



Each project can be inserted and configured with different communication drivers at the same type (according to license type).



The communication drivers manage connections between the device areas and the Movicon variables using links normally defined in the Tag's properties (Dynamic Task), but communication tasks (Static Task) can also be defined to permit associations between device memory data "groups" and Tag groups.

### Automatic Tag Importing

---

The Movicon communication drivers communicate with PLCs and the most commonly used protocols, and has a function that automatically imports variables from the PLC program or PLC's data export file.

Thanks to this function, which can be activated in the drivers' properties, Movicon is capable of pointing to PLC data files, consenting project planners to choose whether to import all or only parts of file, to then automatically generate variable lists in the Realtime Database resource creating

variables with name, type and fixed address in the most simple, automatic and reliable way possible.

**Main Features**

---

- Links to PLC address which can be directly managed in the Tag properties or indirectly using communication tasks.
- Automatically efficient optimized communications. Automatic data aggregations in Dynamic tasks and data exchanged only with variables effectively in use promoting improved and optimized performances.
- Communications, connection and polling management using Static Tasks can all be customized as required.
- Offset addressing can be done by using Static tasks
- Automatic importing from database/program of PLC or device
- Runtime configurability through VBA script interface
- Conditioned executions of each single Tag-PLC address connection
- Bridging functions to consent external transparent access to devices via modem (i.e. teleservice)
- TAPI functions to allow automatic calls to remote serial devices via modem
- RAS functions to allow automatic calls to remote ethernet devices via modem
- Advanced debugging functions and Tracing Log
- Immediate and direct communications and Cable Testing

**Performances**

---

Communication performances depend on the quantity of data exchanged simultaneously and the protocol type used.  
 For example, this table shows results of performance tests and can be used as a 'rule of the thumb':

System	Detail
PC	Windows 32/64 bit, Pentium 4, Movicon 11
PLC	SIMATIC S7-314
I/O Driver	Siemens S7-TCP-IP
Data exchanged	10,000 Word simultaneously
Performances	1.15 sec. Refresh time

**6.1.1. Restrictions**

---

The restrictions to take into account, when defining drivers, are shown in the below table. Note: homogeneous data aggregation and appropriate configuration will improve.

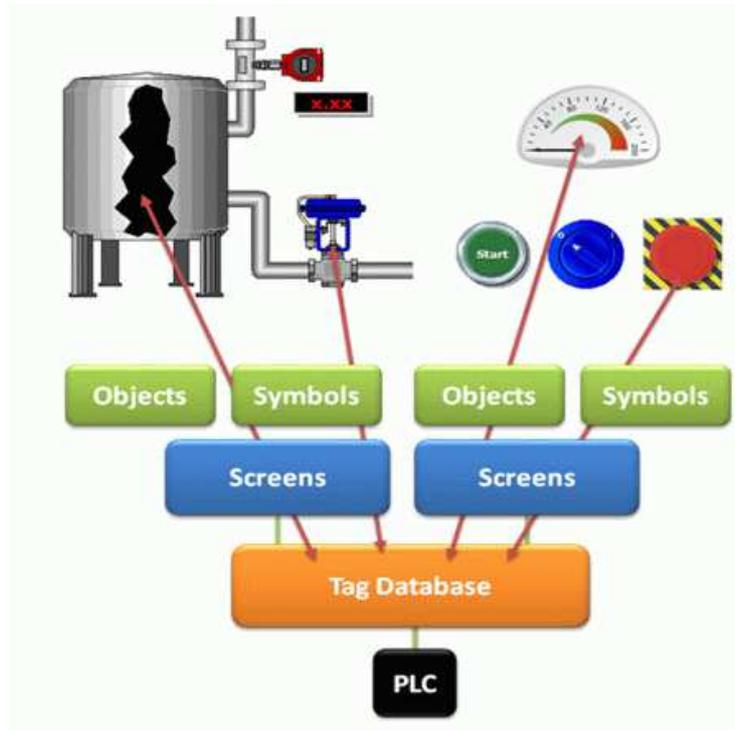
System	Restrictions
Win32/64	Max 64 drivers simultaneously
WinCE	Max 4 drivers simultaneously (Not more than 2 is recommended)
Number of Stations	Max 128 Stations for each single driver used.
Data Structure	Data structure support depends on the protocol being used.
Parent-Child:	The same driver cannot be used in parent-child project relationships: driver will have to be duplicated and renamed.

Variables from child projects can be linked to parent project communication drivers only by using a static task (not by using dynamic link).

# 7. HMI Screens

---

HMI Screens The Screen resources are one of the fundamental elements of Movicon projects. The Movicon Screens are in fact windows containing graphical interfaces (GUI) built from drawings, symbols and graphical objects. The screen window is a means through which the process is supervised (or part of it) and controlled using graphical commands for process animation associated to field variables.



The screen windows are operator interfaces and can be activated for operator use in “Maximized Screen” and “pop-up” mode being windows that display on top of other windows. Movicon also has an interesting object called the “Embedded Screen” through which screens can be displayed within other screens.

## Main Features

---

- Window Size, background color and image can be set in window’s properties.
  - Vectorial graphics and raster container, proprietary and external.
  - Each screen’s content can be adapted to screen resolution automatically (or to screen size).
  - Scroll bars and MDI Tabs can be displayed in runtime.
  - Option to keep screens in memory when not displayed.
  - Antialiasing function for vectorial graphics.
  - Possibility to manage 32 layers for displaying the various objects contained in screen both when in runtime and design mode.
  - Possibility to set a screen size and quality when displayed by Web client.
  - API Basic Script Interface for complete customization using VBA language
- Screens are inserted in the project’s general properties where the “Startup Screen” will be the one which will open up automatically when starting project up in Runtime mode.

### 7.1.1. Opening Screens

---

Commands for opening screens can be set with the following modes:

Activation from:	Details
<b>Button</b>	Each graphical command object can be set to open any screen in any mode deemed appropriate.
<b>Shortcuts</b>	Each shortcut command can be set to open any screen in any mode deemed appropriate.
<b>Menu</b>	Each Menu command can be set to open any screen in any mode deemed appropriate.
<b>Events</b>	Each Event object command can be set to open any screen in any mode deemed appropriate.
<b>Schedulers</b>	Each Scheduler command can be set to open any screen in any mode deemed appropriate.
<b>Logic</b>	Logic consent commands to open screens to activate whenever needed.
<b>Screen Navigator</b>	By using the 'Screen Navigator' resource you can define screen navigations graphically by creating connectors between the various screens in the sequence desired. This screen connector sequence will then automatically be converted into a navigation bar displayed at the bottom of the various screens involved in Runtime.

### Open Screen modes

---

The Movicon Screens, once created and configured, can be opened in runtime with different modes based on the command type being used. This permits the screen to be displayed in different modes according to their specific requirements. A screen can be opened as a simple video screen page or as a Modal window, such as a Dialog Window opening up within the screen currently being displayed. The different opening screen modes are:

Mode	Type	Details
<b>Open Normal</b>	<b>screen change</b>	The "Open Normal" command opens Screen as a Video Page (closing the previously opened screen).
<b>Open Modal</b>	<b>pop-up</b>	The "Open Modal" command activates the screen to open on top of screen already opened.
<b>Open Frame</b>	<b>multi-monitor</b>	The 'Open Frame' command opens a Screen in another window different from the Movicon main screen.
<b>Open in other process</b>	<b>safe mode</b>	This command allows the opening and displaying of a Screen executing another instance (therefore a process in a different memory area).

### Custom Dialog Windows

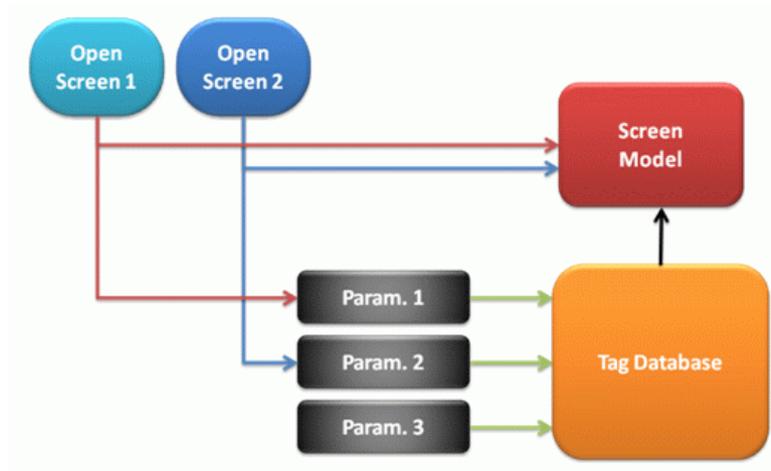
---

Movicon consents to customizing certain dialog windows such as the Numeric and Alphanumeric Pads, Enter Password and Password Expiry windows. This can be done by using the various Movicon screens and symbols to create more suitable ones to replace the Movicon standard windows which show when needed.

### Screen Parameterization

---

Parameterization techniques applied to screens permits one screen to be created (the model), which can be called many times by associating it parameters (the desired Tags replacing Aliases).



The parameter files are simple text files (UNICODE format) within which the associations between "Alias Parameters" and "Effective-Variables (Tags)" are specified. These files are created by the programmer directly in Movicon, or externally.

### 7.1.2. Restrictions

The restrictions to take into account when defining screens are:

System	Restrictions
Win32/64	Max. of 8192 screens per project. Max. of 64 opened at the same time. Max. of 8192 vectorial elements per screen. Max. of 256 objects per screen.
WinCE	Max of 2048 screens per project. Max. Of 8 opened at the same time. Max. of 1024 vectorial elements per screen. Max. Of 64 objects per screen.
Web Client	The Java Applet technology does not consent the use of modal or pop-up windows.

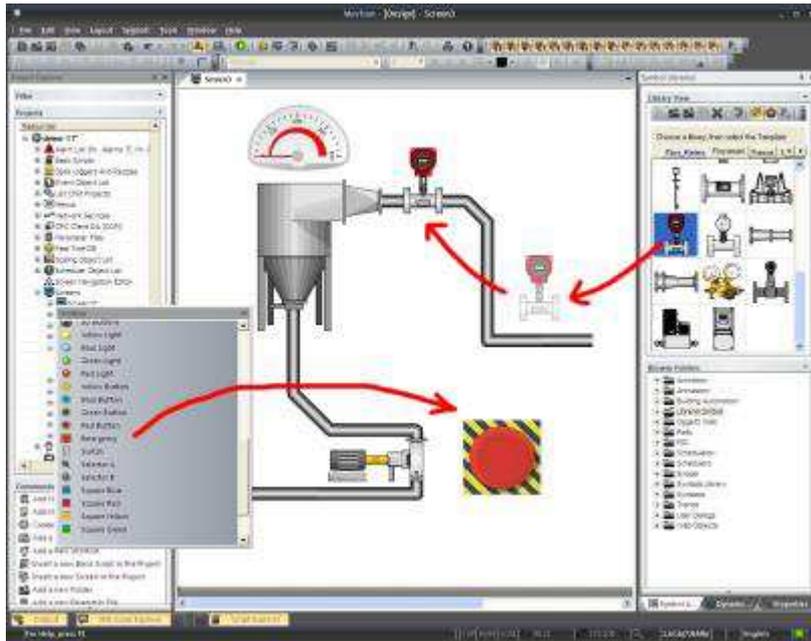
## Interface Graphics

Movicon provides two different methods for creating graphical interfaces where the user can choose to use one or the other or both together and independently: the inbuilt vectorial graphics editor or by inserting external designs (bmp, jpg, gif, wmf, emf, etc.).

The Movicon Objects and Controls have been created in vectorial format and are available in the "ToolBox".

The rich Symbol library is based on the internal vectorial graphics editor and be added to, modified or customized. Other symbols can be imported from external vector files (WMF, EMF) and added to enrich the Movicon library even further.

Each graphic design or symbol can be made "dynamic" by associating Database Tags to the design object's Animation Properties.



### Main Graphical Interface Features

- More than 16 dynamic animation properties associable to design objects and symbols
- Symbol movement and resizing
- Symbol Tab Order settings
- Overlapping symbol orders can be modified
- Selection and modification of multi-selected symbols
- Selection and modification of symbol drawing component properties without ungrouping symbol
- Alignment and grouping selected symbol functions
- Selected symbols can be all sized in the same way
- Selected symbols can be distributed into rows and columns
- Variables can be searched for and replaced in symbols
- Symbols can be exported/imported in "emf" or "wmf" format
- Possibility to embed images in symbols
- Supports Transparency function, both in static or dynamic mode
- Supports AntiAliasing for vectorial drawings (defined in screen)
- Symbols can be locked into position on screen
- Drawings and symbols can be selected with: background colors, background type, gradual fillings, edge colors, font types, etc.
- Variables can be dragged from the Project Explorer window and dropped into symbols
- Symbols can be created with various basic objects (rectangles, buttons, displays, etc)
- Composed symbols can be inserted into library for creating custom libraries in addition to those already available
- Animations can be associated to symbols and activated based on a variable values
- Basic Script code can be associated to symbols
- IL Logic can be associated to screens
- Support to "Power Templates" technology for symbols and those with script code
- Support to the "Public Symbol" concept (heredity)
- Support to automatic display graphics of associated symbol variable qualities
- Commands can be defined by clicking on symbol
- Focus, hilites, tabs, visualizing where project is being processed or active user can all be customized
- Dragging Properties in symbols

## Control and Object Graphics

The Movicon Objects and Controls that can be inserted on screen are those available from the Movicon ToolBox. These components can take on different ways of working, from being just simple geometric shapes or advanced controls for executing commands or displaying data.

The Movicon Objects and Controls are created in propriety vector format, and custom symbols can also be created and associated with animation properties. This format can be exported and imported from Metafile (WMF, EMF) formats.



The prebuilt Movicon objects and controls are available from the "ToolBox" divided into different categories:

<b>Basic Shapes</b>	contains a series of simple geometrical shapes which play a fundamental part in creating screen drawings. These shapes do not have specific properties for Command executions, but support all the animation properties common to all Movicon vector components.
<b>Buttons-Lights</b>	contains various "Button" control types with the task of executing Commands defined on the "Command List" or from simple variable (Tag) settings.
<b>Switches-Selectors</b>	contains an assortment of Switch and Selector control types with the task to execute commands defined on the "Command List" or from simple variable (Tag) settings
<b>Sliders-Gauges-Meters-Displays</b>	contains an assortment of "Meter" control types with the task to display and/or set variable values graphically
<b>Trend-Charts-Data Analysis</b>	contains a series of "Trend" type controls and Chart objects. These controls have the job of displaying dynamical or historical variable (Tag) behaviour with graphical trend curves.
<b>Advanced Shapes</b>	contains a range of different controls with different functions. Among these you will find Displays, Alarm Window, Log Window, Data Logger window, Tab Groups, Grids etc.

The Movicon graphical objects have a lot of different properties through which they can be configured and customized as needed. Some of these properties are common to all objects, while others are specific to object type. For instance, the Animation properties

are common to all objects and are used for managing symbol animations based on variable values during runtime.

**The Main Dynamic Animation Features**

---

<b>Visible-Blink</b>	makes symbol visible or invisible according to the assigned variable value
<b>Trasparency</b>	is used for setting symbol's transparency level based on the assigned variable value
<b>Composed Movement</b>	generates symbol's movement on a trajectory defined by the programmer based on the assigned variable values
<b>Move Horizontal</b>	generates symbol's movement along an X axis based on the assigned variable value
<b>Move Vertical</b>	generates symbol's movement along the Y axis based on the assigned variable value
<b>Start X point</b>	moves symbol's start X point along the horizontal axis based on the assigned variable value
<b>Start Y point</b>	moves symbol's start Y point along the vertical axis based on the assigned variable value
<b>End X point</b>	moves symbol's end X point along the horizontal axis based on the assigned variable value
<b>End Y point</b>	moves symbol's end Y point along the vertical axis based on the assigned variable value
<b>Scaling</b>	resizes symbol based on the assigned variable value
<b>Rotation</b>	rotates symbol around its barycenter based on the assigned variable value. The barycenter can also be modified according to the geometric barycenter
<b>Text – Display Value</b>	displays the assigned variable value in read only
<b>Background Color</b>	modifies background color based on the assigned variable value
<b>Dynamic Text – Edge Color</b>	displays different texts and different colors, if any, based on the assigned variable value
<b>Gradual Filling</b>	fills symbol's color gradually in proportion to assigned variable's contents
<b>Images Animation</b>	displays different images based on the assigned variable value

**Using Expressions instead of Tags**

---

Expressions with VBA syntax can be used instead of Tags associated to Dynamic object properties. The resulting expression value will be the value used for animation. This can be done by using the Expression Editor.

**7.1.3. Restrictions**

---



We advise the project planner to evaluate the impact of having too many animation graphics used frequently throughout project (i.e. blinking, coloring, movements, etc) will have of system performances and the effect it will have on active windows and those loaded in memory. Windows CE restrictions are described in the Movicon CE Manual.

**Command Lists assignable to objects**

---

- **Variables:** consents to setting values and actions in variables (Tags).
- **Screens:** consents to setting different action types for opening screens.

- **Scripts:** consents to executing actions or launching scripts in the projects.
- **Users:** consents to performing actions in password users.
- **Reports-Recipes:** consents to executing different action and command types relating to the project's recipes or reports.
- **Menu:** consents activation of one project menu resource.
- **System:** consents to executing actions in the operating system or Windows command lines.
- **Language:** permits language activation for project text strings.
- **Help:** permits external contextual or custom help activation.
- **Alarms:** permits execution of different command types relating to the Alarm management.
- **Event:** permits project event object activation and execution.

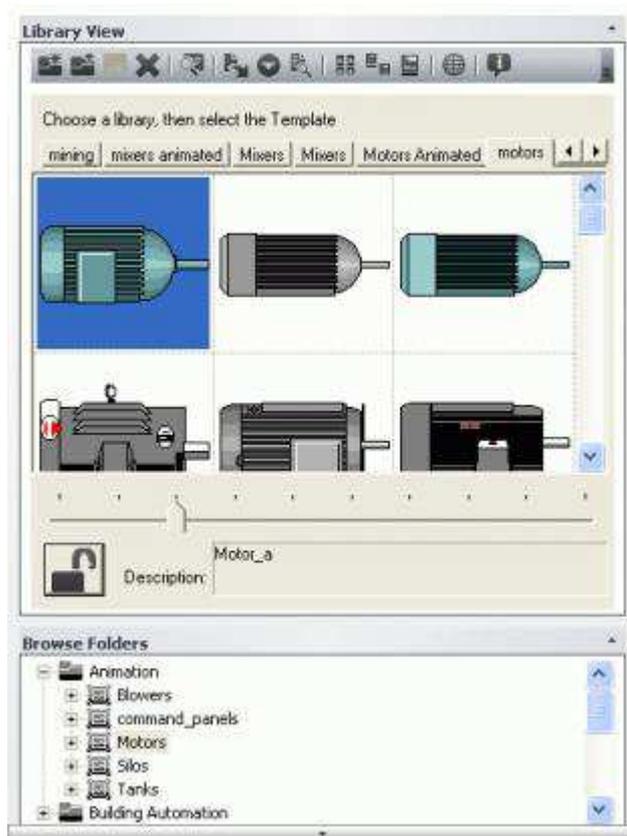
#### 7.1.4. Restrictions

The restrictions to take into consideration when defining commands that can be associated to objects:

System	Restrictions
Win32/64	Max. of 16 Commands can be inserted in one command list for each single object.
WinCE	Max. of 8 Commands can be inserted in one command list for each single object.

## Symbol Library

The basic Movicon Objects and Controls can be grouped together to form various composed symbols. This great feature allows users more complex graphical objects customized to suit every need. The Movicon symbols can be edited and saved as pleased in the "**Symbol Library**". In this library you will find a wide range of various pre-built symbols by Progea ready for you to insert onto project screens.



### **Import-Export**

---

All the Movicon graphical symbols can be imported or exported using the Windows standard WMF or EMF vectorial formats.

### **Public Symbols (Heredity)**

---

Each symbol can be defined as a Public Symbol by assigning a name in the relevant property. Any modifications made to one Public Symbol will then be propagated to all the others in its category.

### **Power Templates©**

---

The Movicon Power Templates technology allows you to create and add other powerful graphical symbols to the library. All the library's symbols maintain all their dynamic command and any Basic Script VBA properties. When inserting Power Templates in project screens, Movicon will create and add all the necessary Variables, Alarms and Data Loggers. Using the Macro for Dropping Code will allow you to automate or customize Template insertions.

### **Centralized Libraries**

---

Movicon consents the paths in which symbols libraries are found to be specified, allowing the editor to access local or centralized libraries for withdrawing symbols or Power Templates for project use.

### **Main Features**

---

- New symbols can be created or existing ones modified, customizing the Movicon Symbol Library.
- New categories can be added to or deleted from the Symbol Library.

- Symbol categories can be protected by passwords.
- Movicon permits the symbol "Dynamic Property Inspector" window to be used for verifying symbol components, code and associated variables. This window can be activated directly from the symbols while in library or while on screen.
- Data Loggers can be associated to symbols
- Alarms can be associated to symbols
- Favourite Categories" of symbols can be organized directly with ToolBox.

The Symbol Library categories are saved in files with the same category name and "msxz" extension. These files can be copied from one PC to another so that created symbols can be re-used in other Movicon installations.

### 7.1.5. Graphics capability restrictions

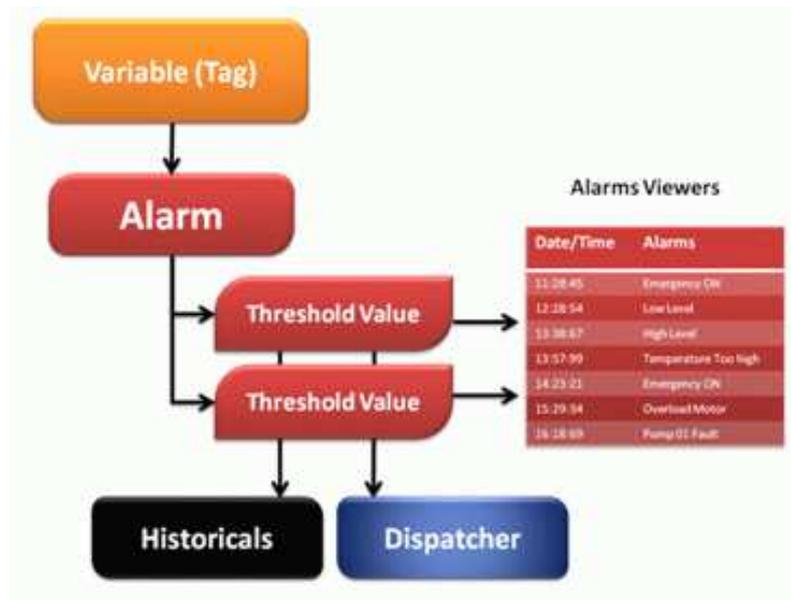
Below is described the restrictions to keep in mind when inserting graphic object on screen:

Restrictions	
Gradual Filling	The "Pipe" object does not support Gradual Filling. In addition, the Gradual Filling with WinCE is only supported by Rectangle objects (polygons do not support this feature either).
Expressions	Objects that use VBA expressions cannot use Local Variables within expressions.
Transparency	The Transparency property is not available if the object rotates (Rotation). The object transparency property is not supported in WinCE.
Transparent Colors	External PNG drawings which use transparent backgrounds are supported in WinCE, but you will need to check whether the WinCE operating system needs the AlphaBlend function (refer to builder) to avoid deteriorating performances.
Dundas Gauges objects	The Dundas Gauges objects are not supported by Windows CE.
Gradient Color	The Ellipse and Circle objects do not support Gradient Colors in WinCE.
Rotation	Rotation objects are not available for WinCE.
3D Effects	The 3D effect for objects is not available for WinCE.
WMF, EMF	WinCE does not support the WMF or EMF vectorial formats.

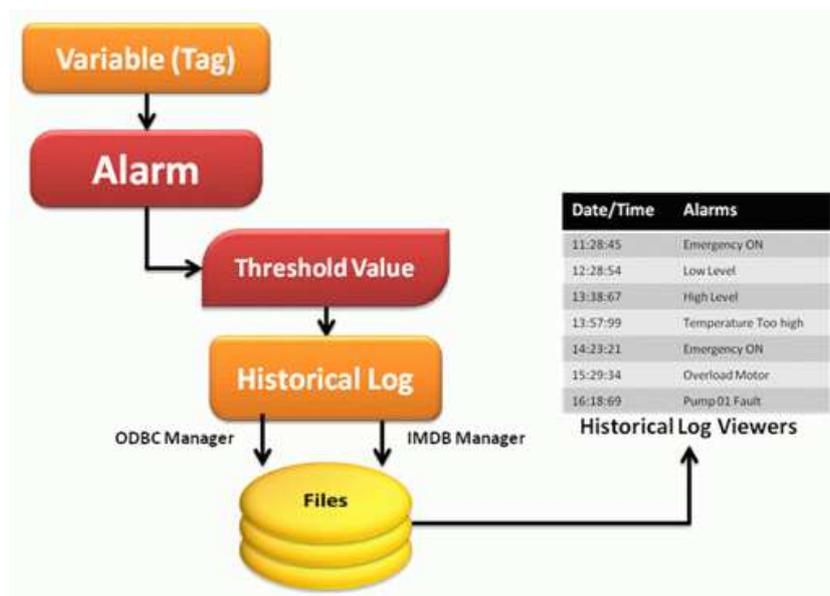


# 8. Alarm Management

The Movicon Alarm resource consents diagnostic and messaging management within the project. The diagnostics is one of the fundamental elements on which any supervision application project is based. Basically, the system's task is to generate alarms in relation to variable intervention thresholds or logic executions.



The Movicon Alarm management is completely configurable to adapt to any application requirement. The alarm property default configuration conforms to the ISA-S-18 specifications. The project alarms are objects which can be set with diverse thresholds. Alarm activation can be evoked by a fixed threshold value or by another variable.



The Acknowledge (Ack) or reset operations in each alarm can be configured, just as their styles and functioning modalities can be completely configured.

## Main Features

---

- Alarms can be defined with one or more intervention thresholds based on the variable value associated
- Alarms can be grouped into start Areas
- Alarms can be enabled/disabled during runtime based on variable values
- Option to enable/disable message sending to Dispatcher during runtime based on variable values
- Hysteresis values can be inserted in alarm activations
- Thresholds can be set exclusively for alarms with one or more threshold interventions
- Option to manage alarms only when associated variable value quality is good
- Alarm thresholds can be associated with help text strings which can also be ID strings (for managing project language changes)
- Fixed activation values can be assigned to alarms and dynamic values can be managed through variables
- Different alarm activation conditions can be selected, "major-equal", "minor-equal", "equal", "different" etc.
- Possibility to activate alarm based on its associated variable's "Total Time ON" (i.e. for managing preventive maintenance)
- Alarms can be associated with priorities
- Alarms can be set with activation delays
- State-command variables can be associated to alarms
- Command lists can be associated for executing on alarm event (ON, OFF, ACK and RESET)
- Alarms can be defined to be acknowledged, reset or just simple messages
- Define whether alarm is to be historically logged and printed
- Alarms displayed in the Alarm Window can have their colors changed as pleased
- Users can force comment entries when acknowledging alarms. All comments are recorded in the historical log
- Audio files can be executed upon alarm activation
- Speech texts can be executed upon alarm activation
- Alarm texts can be sent, such as SMS, E-Mail, FAX and Voice Messages using the Alarm Dispatcher tool. Notifications can be executed on one or more alarm events (ON, OFF, ACK and RESET) and sent to one or more selected recipients
- Basic Script code can be associated to alarm thresholds
- Alarms can be used as Templates, where one alarm can be defined and then associated different variables for creating a different alarm for each variable
- API Basic Script Interface for total customizing using VBA language.

Each alarm defined in the project will create a State File in XML format in the "ALARM" folder in Runtime. This file will store all the alarm's information when project is closed, such as its status (ON, OFF, etc.), total time ON, Unique ID and Transaction, etc.

## Alarm Visualization and History

---

All alarm events that are still active in the system, (alarms in the ON or OFF status still not acknowledged and reset) are displayed in the purposely designed "Alarm Window" object. When selecting an alarm in the "Alarm Window" and clicking on the "Get History", all history will be recovered concerning all the times it was activated and the events each activation generated ("ON", "ACK", "OFF" and "RESET"). A "+" symbol will appear on the left of the alarm's text that when clicked will display the alarm's chronology in a tree structure.

The alarm history, being all the alarm events recorded by the system, is displayed in the "Historical Log Window" grouped according to transaction type. Recorded "ON", "ACK", "OFF" and "RESET" events of an alarm event are displayed in subgroups under the alarm. A "+" symbol is displayed at the side of the alarm text that when clicked upon will open up a tree structure listing all the events recorded for that alarm.

### 8.1.1. Restrictions

---

The restrictions to keep in mind when inserting alarms are:

System	Restrictions
Win32/64/CE	Max. of 99 Alarm priorities
Win32/64	Max. Of 32,000 alarm objects per project, of which a max. of 2048 can have script internally.
Win32/64	Max. of 256 characters per alarm text string.
Win32/64	Max. of 128 thresholds for each single alarm.
WinCE	Max. of 4096 alarm objects per project, of which a max. Of 16 can have script internally.
WinCE	Max. of 256 characters for alarm text strings.
WinCE	Max. of 16 thresholds for each single alarm.
WinCE	Max. of 16 Viewer Clients connected to one Alarm server.

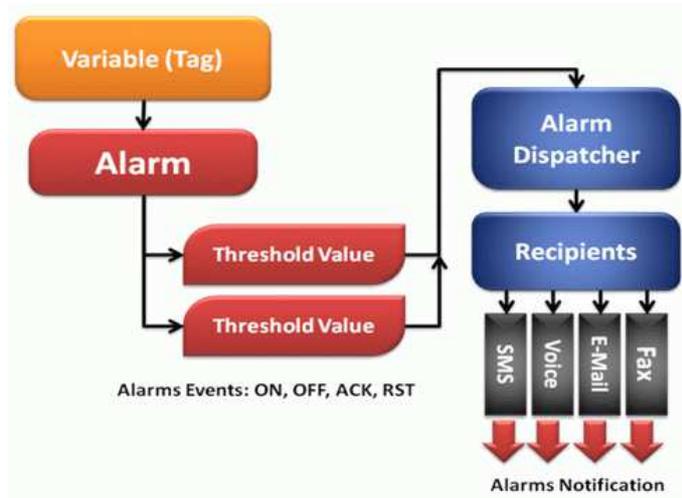


# 9. Events Historical Log

---

Every supervision system must be fully capable of recording all significant events (Alarms, Messages), which occur during project runs, in file archives. Event recording is therefore a fundamental feature to permit historical analysis of incidents that happened while system was running. The feature used for recording and storing historical events in Movicon is called the **Historical Log**.

The Historical Log is a project automatically managed to record system alarms or events in file archives in database format (ODBC Manager) or text format (IMDB Manager).



The Historical Log's task is to chronologically record all project events (Alarms, System Messages, Driver Messages) that happened during project runtime.

The events that are recorded in the Historical Log archives are:

- System operations (operations performed in system)
- System diagnostics (autodiagnosis, Driver diagnosis)
- All the project Alarms that programmer considers as being significant
- Project Alarms and Messages in 4 possible states (ON, OFF, ACK and RST)
- User Log On/Offs
- Status changes of variables associated to the Tracing function
- Contents of system variables associated to project strings on events established by programmer
- Basic script logic messages by means of using the "Debug.Print" function
- Basic script logic messages by means of using the Trace functions
- Messages deriving from other applications setup by the programmer

Movicon permits the use of two different types of historical log archives for managing data file on disk: ODBC Standard and IMDB. If you opt to use the ODBC standard, Movicon will use relational database tables and files as the file archive. Normally, these files are MsSQL Server type or MsAccess, but ODBC formats of any other Relational DB can be used (i.e. Oracle or other).

If you opt to use IMDB, Movicon will record historical log data in text mode that can be structure in XML format or encrypted.

## Main Features

---

- Choice of Database, ODBC or IMDB
- Possibility to disable Historical Log from recording
- Possibility to set historical data storage duration (in "days")

- Custom DSN can be created
- Database table names and columns can be customized
- Data order, filters and queries can be executed from Log Window
- Log can be printed (totally or partially)
- Log messages can be sent to printer directly
- Variable values can be recorded together with message text

**Data Tables**

---

The Log file tables are:

- Alarms: contains project alarm messages
- Drivers: contains messages deriving from project communication drivers
- SysMsgs: contains project system messages

Each one of these tables has the following structure:

Time Column (TimeCol)	Shows date and time in GMT when recording took place.
Local Time Column (LocalCol)	Shows the local date and time recording took place.
Msec Column (MSecCol)	Shows recording time in milliseconds
User Column (UserCol)	If recorded event was triggered by a user (by using a button etc.), the name of the user logged in the project at that moment is recorded in this field.
<b>Event Column (EventCol)</b>	Shows the event type recorded (i.e. Alarm ON, Alarm OFF, System, etc)
<b>Event Number Column (EvNumCol)</b>	Reports recorded event's ID number
<b>Event Description Column (EvDescCol)</b>	shows the event description
<b>Description Column (DescCol)</b>	Gives information on event type
Comment Column (CommCol)	Shows comments made on event
<b>Duration Column (DurCol)</b>	Shows how long event lasted
<b>Unique ID Column (UniID)</b>	Shows the alarm's Unique ID (value valid for the "Alarms" table only)
<b>Transaction ID Column (TraID)</b>	Shows the alarm's Transaction ID (value valid for the "Alarms" table only)

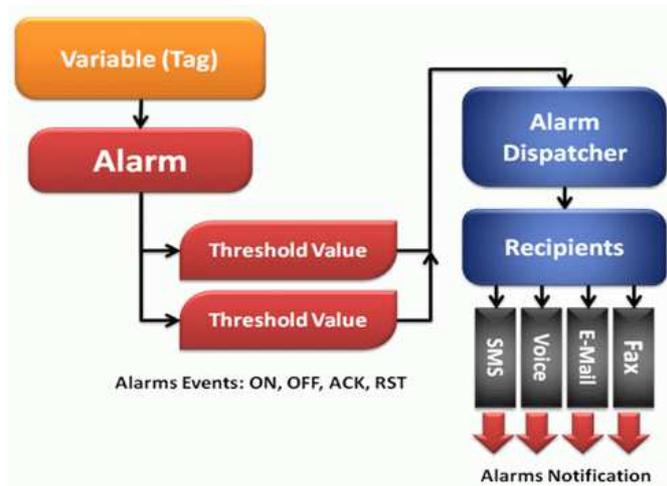
The Movicon "Historical Log Window" object can be used for visualizing Historical Log data and can be inserted in any screen.

**9.1.1. Restrictions**

<b>System</b>	<b>Restrictions</b>
Win32/64/CE	Max of 256 characters for text strings.

# 10. Alarm Dispatcher

Alarm DispatcherThe Alarm Dispatcher is a Movicon feature used for notifying (sending) messages and alarms using various means of communication (PlugIn) such as SMS, E-Mail, Fax or Voice Messages (Text-to-Speech).



The communication technologies available are:

Notifier	Win32/64	WinCE
<b>SMS messages using SMPP protocol</b>	•	
<b>SMS messages using GSM</b>	•	•
<b>Fax</b>	•	
<b>Voice Messages (Vocal Synthesis)</b>	•	
<b>E-mails using SMTP protocol</b>	•	•
<b>E-mail using MAPI</b>	•	

## Main Features

- SMS, Fax, Email, Voice notifier options
- Notifications given on for each of the 4 different alarm states (ON, OFF, ACK, RST)
- Notification activation or deactivation can be based on variable condition
- Notification via E-mail with document attachment support
- Notification via Voice with integrated Vocal Synthesis (TTS in SAPI 5.1)
- Voice calls with timeout, passing on to next user and call received confirmation
- Notification to individual recipients or groups of recipients
- Notifications can be sent to phone numbers in variable
- Incoming or outgoing vocal messages or new alarms can be customized
- Message queue management customization
- API Basic Script interface for total customization using VBA language

## Text to Speech

This vocal synthesis notification function has been fully tested and is guaranteed to work in:

- Windows Windows 32/64 bit OS

- Microsoft SAPI 5.1 Libraries
- PCI internal Modem Wisecom 56000 bps PCI - W92PCIS model

### **10.1.1. Restrictions**

---

The restriction to take into consideration when configuring the Dispatcher are:

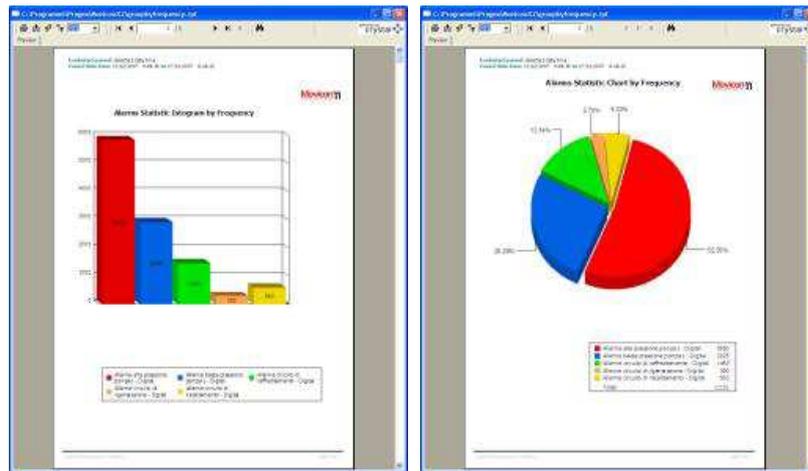
<b>System</b>	<b>Restrictions</b>
Win32/64	Max. of 32,000 notifications manageable per project
Win32/64	Max. of 32 users can be subscribed for each notification
Win32/64	Max. attached file size for emails is 512 Kbytes
Win32/64	Max. number of 10 attached files
WinCE	Max. of 4096 notifications can be managed per project
WinCE	Max. of 16 users can be subscribed for each notification
WinCE	Max. attached file size for emails is 256 Kbytes
WinCE	Max. number of 10 attached files

# 11. Alarm Statistic

---

Alarm StatisticsMovicon has an inbuilt tool for managing system downtime and alarm occurrence calculations and analysis. The tool interacts analytically with the Events Historical Log files to produce Reports with Tables and Charts classifying alarms with their total number of interventions and durations.

This tool has been completely integrated into the system and does not need additional installations or configurations. This tool is a dongle option.



## Main Features

---

- Integrated and native access to historical log data files, independently from DB format in use.
- Statistics Reports, integrated and predisposed on native command.
- Analysis in Table format listing alarms classified by duration (total time ON) or by intervention frequency.
- Analysis in charts and histograms or pie charts listing alarms classified by duration (total time ON) or by intervention frequency.
- Reports can be filtered by time range (total time or by time range set by user)
- Additional Queries and filters can be set
- Print preview and print
- Open and customizable report solutions (based on Crystal Report)

## Customizable Analysis

---

Movicon is predisposed for creating customized analysis on statistical alarm information. A file is created for each alarm which continuously saves all information relating to that alarm. By using the appropriate basic script functions you can get the various types of alarm statistics such as:

- the alarm's total time on
- the date and time of the alarm's last transaction
- the comment associated to the alarm
- total number of times alarm turned ON
- total number of times that alarm was acknowledged
- total number of times that alarm was reset
- alarm's Unique ID
- alarm's Transaction ID

### **11.1.1. Restrictions**

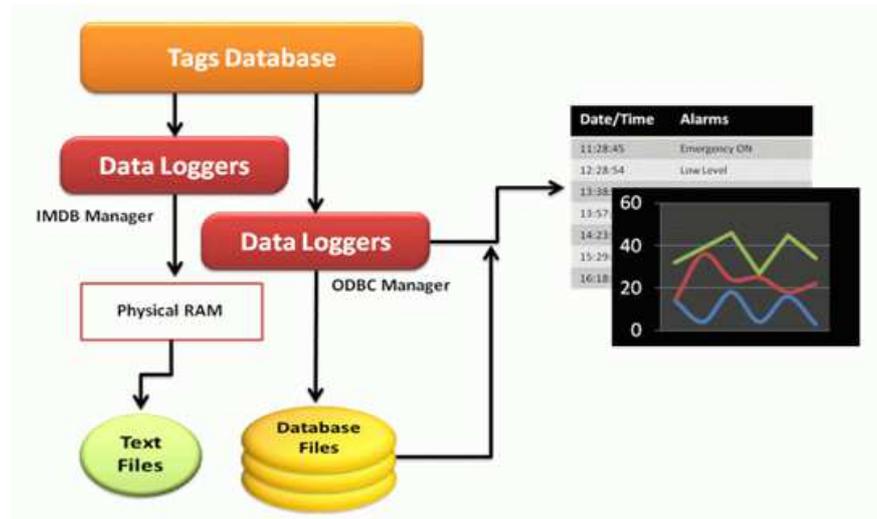
---

The restrictions to take into consideration when configuring are:

<b>Sistema</b>	<b>Limitazione</b>
WinCE	Does not support Alarm Reports
Web Client	Does not support Alarm Reports

# 12. Data loggers

The Data Logger is a powerful tool for recording and simplifying the managing of data in databases, both ODBC and XML text. The Data Logger's task is to sample associated variables (Columns) and record the data on data files according to the recording modes predefined in the data logger object's properties. The archive management and file input modes can be performed using ODBC in relational databases or IMDB in XML text format.



In the Database created by Movicon you will also find some other columns in addition to the data columns defined by the user (being the variables to be recorded). These columns are inserted by Movicon for reporting the following data:

Time Column (TimeCol)	Shows date and time in GMT recording took place.
Local Time Column (LocalCol)	Shows the local date and time recording took place.
Msec Column (MSecCol)	Shows recording time in milliseconds.
User Column (UserCol)	If recorded event was triggered by a user (by pressing a button etc.), the name of the user logged in the project at that moment is recorded in this field.
<b>Reason Column (ReasonCol)</b>	Reports whether recorded 'on time', 'on command' or 'on change'.

## Main Features

- Easy configurable data logger object recording, each data logger configured is independent from any others where data is logged on completely transparent userfriendly Database files.
- Data can be recorded 'on time' according to sample time settings
- Data can be recorded 'on command' using the appropriate Recording variable
- Data can be recorded 'on change' using the relevant deadband settings
- Option to enable recording on time based on variable status or a predefined time range in which to execute recordings
- Table can be reset (cancelling all records) based on variable status
- Report files can be associated to DataLoggers for displaying data
- Report files can be printed based on variable status
- Choice of databases to use: ODBC or IMDB

- Database age can be set to determine data storage in data log before recycling
- Custom DSN can be created
- Database tables and columns can be customized
- DataLoggers can be used as Templates and associated with variables to record
- Extra columns can be added for each variable to report variable's: "Quality", "TimeStamp" and "Number of Updates"
- Variable values can be recorded as "instantaneous", "minimum", "maximum" or "Average" values
- API Basic Script interface permitting complete customization with VBA language

### 12.1.1. Restrictions

---

The restrictions to take into consideration when configuring are:

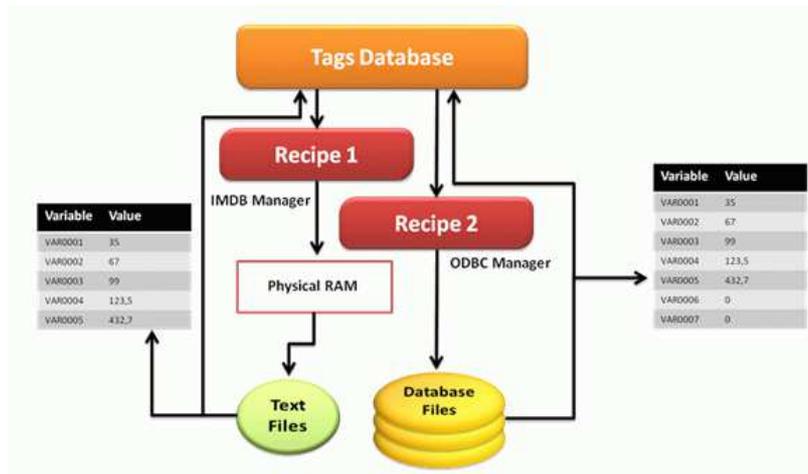
Feature	Restrictions
No. of Data Loggers per project	Win32/64: Max of 512 Data Loggers per project WinCE 5.0: Max of 4 Data Loggers per project WinCE 6.0: Max of 16 Data Loggers per project
No. of Columns per Data Logger	Win32/64: Max of 1024 Columns per DataLogger (Max. of 127 with IMDB) WinCE: Max of 128 Columns per Data Logger (Max. of 90 with IMDB)
Archive life cycle	Win32/64: Max of 9999 days for DataLoggers (may encounter further restrictions depending on pre-chosen data formats) WinCE: Max of 365 days for DataLoggers (may encounter further restrictions depending on pre-chosen data formats)  The maximum life duration will depend on the chosen database's maximum engine capacity (i.e. SQL Server CE and SQL Express Max. 2GBytes).
Sampling Frequencies	Win32/64: Max. of 100 milliseconds (may encounter further restrictions depending on pre-chosen data formats) WinCE: Max. of 1 second (may encounter further restrictions depending on pre-chosen data formats)

# 13. Recipes

Often production plants demand that parameters and set-points of product types be stored on file so that they can be easily modified when needed to activate production runs of a different product type. The original product type can be retrieved from the archive and reactivated in the system according to production needs and is commonly known "as production recipe management".

Movicon provides users with a "Recipe" resource which is a powerful tool for manipulating databases in order to use them for managing recipes through the predisposed functions in the Recipe or relating Object properties, or for customizing them using the Basic Script functions.

The archives and file input modalities can be performed using ODBC in relational databases or IMDB in XML text format.



## How Recipe Objects work

The Recipe objects are available in all Movicon versions and permit users to manage recipes using data file archives and graphical object interfaces.

The Recipe object constitutes the data management engine and is available from the DataLogger/Recipe resource. Variable (Tags) associations can be defined in the Recipe object's properties to form the recipe's set points, along with the variable which defines recipe name (Index).

Other variables can be defined for activating operation commands such as "Save" "Activate", "Delete". The data file archive structure can be built with normal database tables with set point value columns (variables) and a Index column to identify the recipe Index. Every time a new recipe is created, a new record is created by adding a new row to the Table.

The below table illustrates an example of how data is organized in a recipe log file. Each table column is associated to a process variable and the Index column indicates the recipe index record. In this case our example recipe has only been set with three variables represented by the three columns plus the obligatory Index column totaling a number of four columns always shown for default, the praxis for all Movicon Historical Logs.

Index	VAR0001	VAR0002	VAR0003
Recipe 1	234	234	12
Recipe 2	123	450	34
Recipe 3	321	230	44
Recipe 4	085	450	21
Recipe ...	061	450	13

The user can interact through the graphical objects connected to the Recipe object to select and activate the desired recipes presented in the data log files using the "Activate" command. While selecting recipes, filed data is automatically loaded into the appropriate memory buffer until receiving the Activation command to be effectively transferred to the Variables (Tags) in the RealTime database associated to the recipe in question.

## **The Recipe Interface Graphics**

---

The Recipe engine described above must be connected to a graphical interface, through which the user should be able to select and activate recipes, or insert new ones or delete existing ones. Movicon provides users with a graphical object designed for this purpose called the 'Recipe Manager' which is found in the ToolBox under the "Advanced Shapes" category. This object can be inserted on any screen and configured through its properties to be associated as the Recipe Management engine.

In addition, the Recipe Interface can be managed using several graphical objects which can be inserted onscreen as pleased to suit the desired recipe data management accordingly. A Wizard can be used to create screens and graphical objects associated to recipe data and operating commands.

In this case, corresponding temporary variables must be created for each process variable and Recipe Index variable. The temporary variables are variables of the same type which the recipe uses for holding data loaded from the database. When the "Activate" command is evoked, this data is transferred from the temporary variables to the process variables. This will also allow users to modify database recipes without interfering with the recipe currently used in the production run.

## **Main Features**

---

- Object oriented Recipe management, involving archives with Relational DB or text files managed automatically and transparently.
- Customizable Recipe Manger objects. Graphical interfaces can be managed independently and flexibly using objects on screen and wizards.
- Recipe data Import-Export in text files
- Execute data query extractions
- Move to first record of data selected from a RecordSet
- Move to last record of data selected from a RecordSet
- Move to previous record in respect the one currently selected from a RecordSet
- Move to next record in respect to the one currently selected from a RecordSet
- Possibility to check query execution statuses
- Recipe activation
- Save/Modify recipes
- Cancel recipes
- Updata RecordSet of data selected
- Execute recipe data filters (WHERE clause)
- Execute recipe data order (ORDER BY clause)
- API Basic Script Interface for complete customization using VBA language

### **13.1.1. Restrictions**

---

The restrictions to take into account when configuring are:

<b>Feature</b>	<b>Restrictions</b>
No. of Recipes	Win32/64 : Max of 512 Recipes per project WinCE 5.0: Max of 4 Recipes per project WinCE 6.0: Max of 16 Recipes per project
Nr. Colonne	Win32/64: Max of 1024 Columns per Recipe (Max. of 127 with IMDB) WinCE: Max of 128 Columns per Recipe (Max. of 90 with IMDB)

# 14. Trend and Data Analysis

## 14.1. Trend and Data Analysis

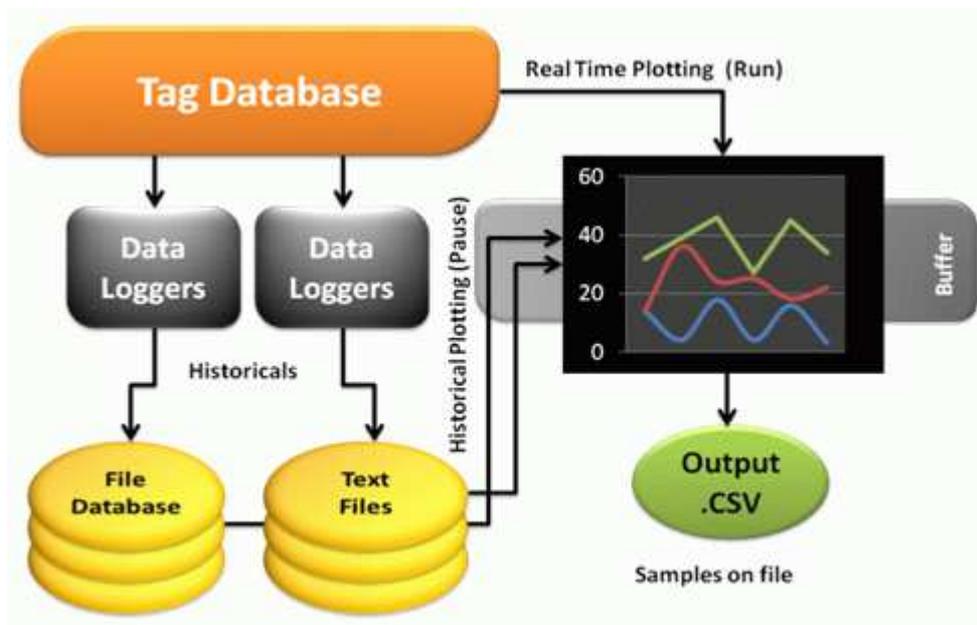
The Trends allow variables and analysis of recorded and filed data to be displayed graphically in trend curve lines. The Trend object is a tool that offers different function modes. It not only permits autonomous data visualization and storage, it also permits connections to database objected inserted in the Data Logger, to consent representation of stored data both according to prefixed time ranges or on event.

The Data Analysis permits analysis to be performed exclusively on historically logged data by providing functional features specific for this purpose making it an all round powerful graphical data scanner.

The Trends work two ways, "runtime data visualization and logged data visualization" (run-pause), while the Data Analysis only performs analysis on logged data but provides much more advanced analysis functions to ensure that this job is done thoroughly to suit all needs.

### Trend Features

The Trends can be set to work according to time or as XY Trends. Time based Trends display times on their X axis and traced measured values on their Y axis. XY Trends display two curve behaviors as X chart and Y chart. In this case the X axis is no longer associated to a time but to the first pen in the list. The Y axis is associated to the second pen on the list in a way that the X point and Y point coordinates are determined by the associated variables, and the Trend generates a trace plotting the previous XY points to the current ones for each sample.



### Main Trend Characteristics

- Displays curves based on real-time variables (Run mode)
- Displays curves based on data recorded by Data Loggers (Stop mode)
- Format, style, scale, legend etc. can all be customized
- Up to Millisecond data sampling
- Each pen can be displayed with Average, Minimum, Maximum value curves
- Scales can automatically adapt to object size or be logarithmic



**14.1.1. Restrictions**

---

The restrictions to take into account when inserting Trends and Data Analysis are:

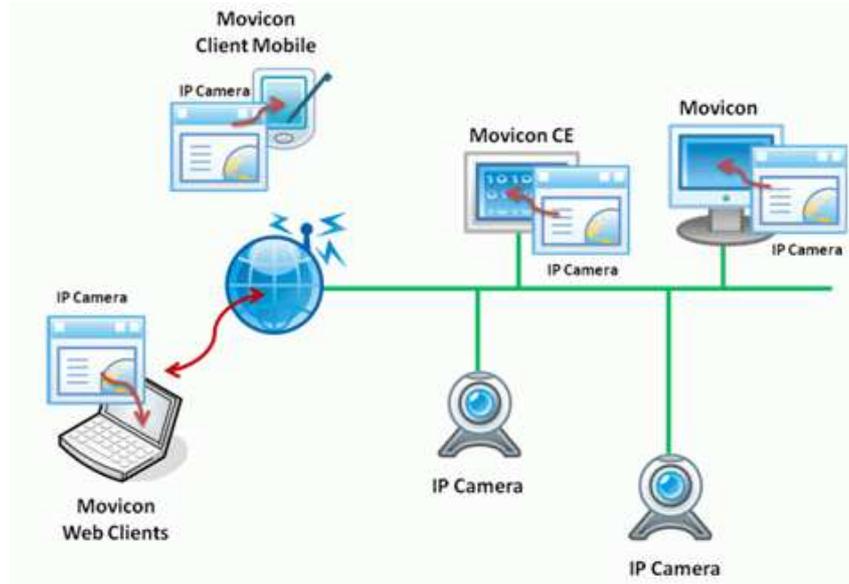
<b>Feature</b>	<b>Restrictions</b>
Maximum number of pens	Win32/64 : Max of 64 Pens for each Trend WinCE: Max of 16 Pens for each Trend
Maximum number of "points" (samples) can be displayed	Win32/64: Max. of 10,000 points for each Trend WinCE: Max of 500 points for each Trend
Maximum sampling frequency rate	Max. of 10 milliseconds WinCE : Max of 1 second



# 15. IP Camera Viewer

---

The Movicon "IP Camera Viewer" is an object through which images taken by a IP camera can be viewed. This object allows you view a sequence of images in realtime received from a IP camera. This type of camera supports current image downloading through http. Downloading is done using jpg images. Images downloaded from the IP Camera Viewer to Win32/64 platforms are stored in memory (therefore without using disk), while temporary files are used when downloading to WinCE platforms (the Temp folder is usually loaded in memory).



## Functional Characteristics

---

- Supports a wide range of popular standard IP Cameras
- Image visualization in screens with custom refresh
- Image visualization with parameterized IP Camera addresses
- Image visualization supported also in WinCE and Web Client

### 15.1.1. Restrictions

---

The restrictions to take into consideration when inserting IP Camera Viewers are:

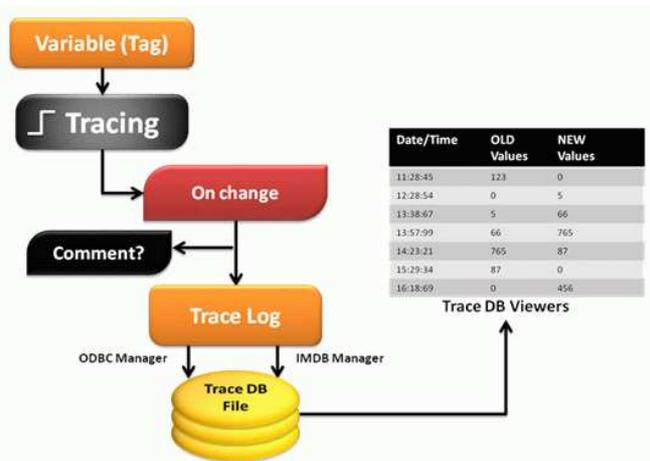
Feature	Restrictions
Max. refresh frequency	Win32/64: Max. 0.5 sec. WinCE: Max. 1 sec.



# 16. Tracing and Audit Trails

## 16.1. Tracing & Audit Trails

Variable Tracing or Audit Trail is a Movicon feature that consents all variable variations to be historically logged by recording each variable value change and reason causing it in Database or text files. The main scope of this feature is to keep track and document all the values obtained by variables during process runs and the causes that made them change in order to uphold proper maintenance (Tracing) and process certification (Audit).



Data relating to each status change occurred for each single variable Tag is files in the appropriated files (one Table for each Tag) and is made available to the user through display window objects which can be inserted on screen.

### Main Features

- This function can be directly inserted in Tag properties
- Properties are defined for setting time ranges to store on file
- Automatic association of information relating to date, operator name, previous value, current value, causing action, comments and strings, variable qualities, etc.
- Option to accept changes only after the obligatory entry by user commenting on action causing change (Audit Trails feature).
- Option to enable 24 hour a day tracing or only in certain time ranges.
- This function is available for all variable types.

### 16.1.1. Restrictions

The restrictions to take into consideration when setting variable Tracing are:

Funzionalità	Restrictions
Max. number of traceable variables	Win32/64: Max. of 2048 variables (Tags) with Tracing (Audit Trail) function enabled WinCE: Max. of 64 variables (Tags) with Tracing (Audit Trail) function enabled



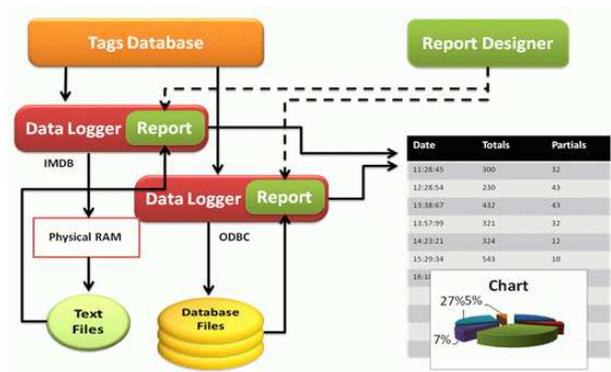
# 17. Reports

---

The Movicon Reports can be managed using the Report Designer with Crystal Report™ or Textual Reports.

The Reports are tools for displaying or printing filed process data in a determined time range. Thanks to Movicon, the Report management is extremely easy to realize, ensuring the necessary openness to get Report prints to exactly match the needs of end users.

Not only do they have access to historical log data recorded by the Data Loggers, they also must guarantee visualization and calculated amounts, totals, averages, variations according to selected time range, product batch type, operator, shift, etc.



Movicon has integrated the "Report Designer" totally free to allow users create autonomously reports relating to the Movicon databases. In addition, Movicon still remains compatible with the "Crystal Report™" report engines which, however, is not integrated in Movicon and must be purchased separately for creating the reports desired. Textual Reports are also compatible with the WinCE environment where they can be created with great ease and simplicity.

## Report Designer

---

The Report Designer is an extremely powerful tool that has been integrated in Movicon to give users the possibility to create reports autonomously in relation to the Movicon databases without having to purchase supplementary software packages. The Report Designer libraries are installed with Movicon totally free and do not need to be enabled on license for using.

### Main Features

---

- Report Editor integrated in development system
- Direct and integrated Report link to Database files using the Data Logger properties
- NET-based technology
- Powerful integrated calculation, table and graphics functions
- Print preview and direct printing management
- Reports can be automatically published on Web using the automatically created tables based on ASP.NET technology
- Supports Query and scripting

### 17.1.1. Restrictions

---

Report Designer management not available for Windows CE

## Report Engine Crystal Report™

---

Crystal Report™ is an easy and powerful tool to use for creating basic graphics designated to show data lists, tables, reports and any other thing else needed for presenting database data tables realized with the Movicon recording tools.

Report pages created with Crystal Report™ will be given the ".rpt" extension and are linked to the data tables generated by Movicon. This tool needs to be purchased separately.

Movicon supports the Crystal Report™ 10.0 version.

### Main Features

---

- External Crystal Report™ Editor independent from Movicon, based on different platforms independently from Movicon system and/or Licenses
- Direct and integrated Report link to Database files using the Data Logger properties
- Report Engine integrated in system, does not need Crystal Report license in Runtime.
- Powerful integrated calculation, table and graphics functions
- Print preview and direct printing management
- All Web publishing functions work according to the Crystal Report technology
- Supports Query and scripting

### 17.1.2. Restrictions

---

Crystal Report™ Report Engine not available for Windows CE

## Textual Reports

---

Movicon also gives users the possibility to use another management to create simple textual reports. The functional logic of these reports is based on a "template" layout file with which is used for defining the report file's structure. This layout supports fields with a special syntax, used for defining which Data Logger or Historical Log variables or fields are to be displayed and in which format to use. The textual report can also be of "mixed" type, being that it can display certain DataLogger or Historical Log field variables and values in the one same report. In addition, there are other fields predisposed for printing the date, time and page number. The layout file must be open in text mode (i.e. Rich Text Format or Text).

Textual Reports are also supported in runtime for Windows CE for displaying, printing and saving simple reports.

### Main Characteristics

---

- Simple and independent Report management, no need to use the technology often used for Report managements
- Report editing in RTF or HTML text files
- Data formatting for managing strings and Tag values
- Print management commands
- Management also available for Windows CE

### 17.1.3. Restrictions

---

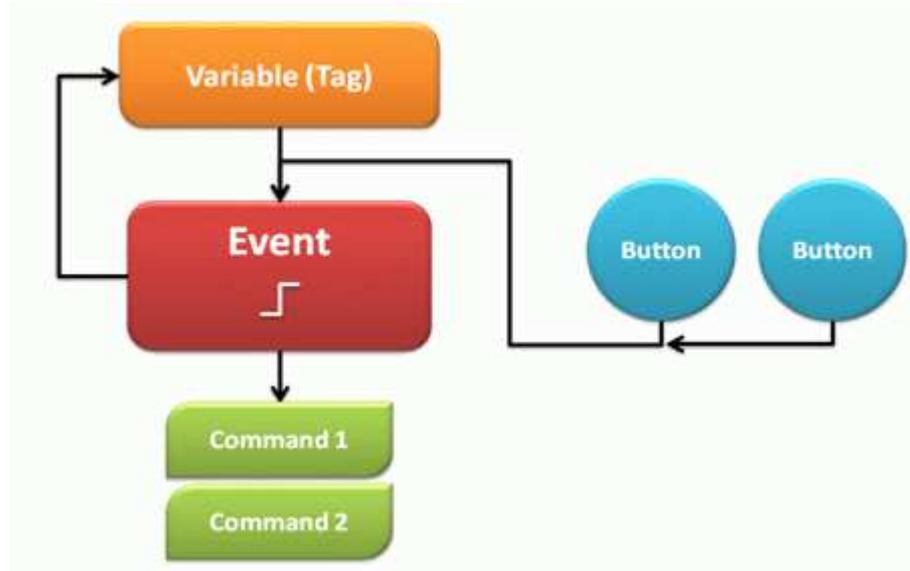
Graphical formats not supported (images, graphics, etc.)

# 18. Events

---

The Event objects are tools that consent to executing commands in function with Movicon variable changes or in association with other command objects.

The Event commands are very handy in projects for keeping variable values under control so that the necessary commands can be executed if any of these variables (event) changes. This saves the programmer a great deal of time in having to write often endless script code for controlling variables, which is now no longer needed.



The commands that can be associated to Event objects are executed when:

- the reference variable value changes
- the reference variable value becomes equal to the one defined by the programmer
- the reference variable value becomes more than the one defined by the programmer
- the reference value becomes less than the one defined by the programmer
- Any associated graphical command object, menu or shortcut is activated



These commands are executed only on the rising edge of a preset condition.

This can result very useful in the Movicon "Command List" where the event object execution can be selected. This functionality consents the project event resource to be used as a container of commands to be used later in the project. By keeping the command list in one unique place it will be possible to modify the command list of just the one event object, to automatically modify the command lists of all objects referring to that Event object.

Naturally if the event object has been associated an activation condition, this will remain valid as such.

## Main Features

---

- Command activation on logic conditions, managed with objects.
- Command list centralization
- Event object logic enabling
- Management also available for Windows CE

### 18.1.1. Restrictions

---

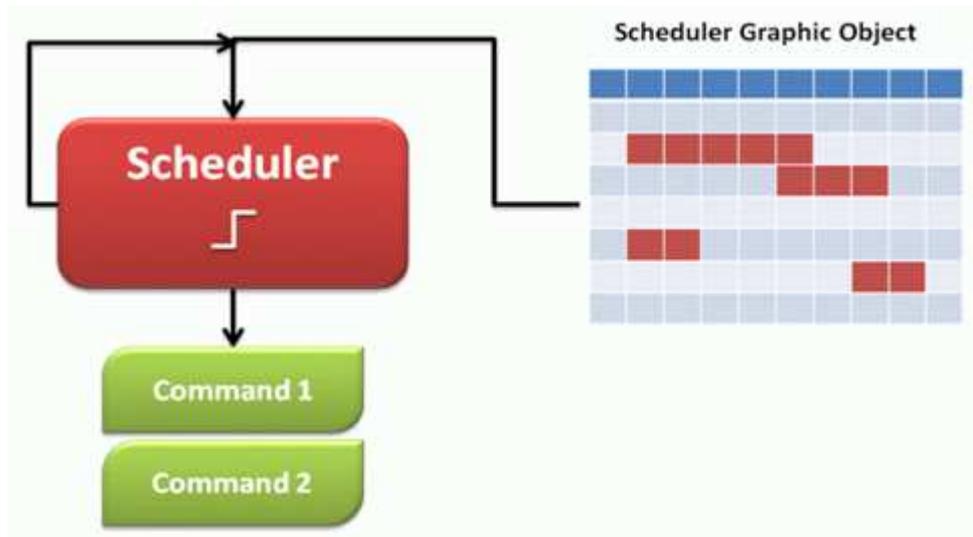
Feature	Restrictions
No. of Event Objects	Win32/64: Max of 512 Event Objects per project. WinCE: Max of 64 Event Objects per project

# 19. Schedulers

---

The Scheduler Command resource is the tool used for scheduling commands to activate at certain time periods.

The scheduler is a great thing to have and a big time saver when needing to prefix commands activations with different times or those that need activating repeatedly at certain times throughout project run or set freely through the graphic interface during runtime when needed.



The following time periods can be selected for executing commands:

- Every minute
- Every hour
- Every day
- Every Sunday
- Every Monday
- Every Tuesday
- Every Wednesday
- Every Thursday
- Every Friday
- Every Saturday
- Every Monday
- At a fixed date to be set by programmer
- Hour Plan: this option allows command to be activated throughout one week according to planned times which can also be managed and modified in runtime using the "Hour Selector" object.

## Scheduler Window/Hour Selector Object

---

The Movicon "Hour Selector" object is configurable and used for displaying and modifying timetables of Schedulers associated to this object. This object is used for editing weekly scheduler plans during project runtime.

It can also be used for adding new runtime schedulers (not defined when in design mode) through which only set commands can be activated in variables.

## Main Features

---

- Command activations on prefixed time ranges defined in design mode or settable in runtime using the graphical object.
- Schedulers can be activated/deactivated in runtime according to variable values.
- Option to enable the 'Holidays' management. In this case a second plan can be defined to execute commands on festive days/public holidays. This can be done only if the "Hour Plan" has been selected
- Option to use the Holidays plan for scheduling commands to activate on Sundays. This management can only be used if the "Hour Plan" mode has been selected
- "ON" command list definitions to be executed upon list condition activation
- "OFF" command list definitions to be executed upon list condition activation. This management can only be used with "Hour Plan" mode enabled.
- Objects can be centralized on Server and set from Clients

### **19.1.1. Restrictions**

---

<b>Feature</b>	<b>Restrictions</b>
No. of schedulers per project	Win32/64: Max of 512 Event objects per project WinCE: Max of 64 Event objects per project

# 20. Scaling Object List

---

The Scaling object list resource is used when needing to convert data from numeric values to engineering units automatically.

More than often variables collected in the project from the plant system are digitally converted raw values of measured physical quantities. In order to get these values accepted, they must be converted or "scaled" into engineering units in ratio to their physical measure. This involves a calculation process which transforms digital value minimums and maximums to raw value minimums and maximums, mostly done in linear. You may find that you will need to scale two variables within the project, or perform NON linear scaling procedures. The Scaling resource has been designed for just this by allowing you to specify the two variables, the one to be scaled and the one already scaled, and any NON linear conversions needed.

## Main Features

---

The variable scaling object can be configured with the following properties:

- Minimum and maximum limit definitions for input variable (variable to be scaled)
- Minimum and maximum limit definitions for the output variable (scaled variable)
- Deadband value definitions
- Non linear scaling possible

### 20.1.1. Restrictions

---

The limited number of Scaling resource objects that can be inserted and executed in a project are:

Sistema	Limitazione
Win32/64	Max of 512 Scaling objects.
WinCE	Max of 32 Scaling objects.



# 21. VBA Basic Script

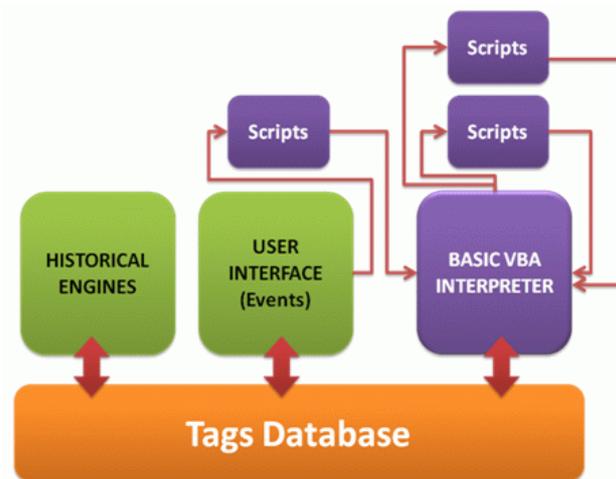
---

Basic Script functions can be made available in different circumstances and modes within a Movicon project. The Basic Script codes are used for programming logic functions using the VBA scripting language (Visual Basic for Application).



You can resort to using Script code for customizing any functionality requiring a specific way of programming that otherwise cannot be done configuring the other resources, symbols, objects and properties provided.

The basic code can be used in many different parts of the project: basic script resource, directly in drawing object properties, as code associated to object events (alarm, drawing or symbol) on screen.



Movicon comes integrated with a software component called WinWrap Basic which allows you to interact internally with routine applications in Basic language. In addition to the script functions provided by this component there are also all those belonging to Movicon which are needed for managing project resources and objects.

## Main Features

---

- Integrated scripting language totally compatible with VBA™ (Visual Basic for Application)
- Supports Multithreading
- Scripts can be embedded in graphical objects with response to events. Embedded script in objects from the library remain independent from the project
- Same VBA syntax and language for Win32/64 and WinCE
- Instruction sets can be extended with customized functions and methods
- Support to third party software (OCX, ActiveX, DLL)
- API and Windows™ WMI functions can be declared and called
- Supports VBA™ and VB.NET syntax
- Supports the ADO database management technology (Active Database Objects)
- Supports Intellisense during data entering
- Supports Unicode strings
- Scripts access directly to any variable type (Tags) from the project's RTDB Tag Database
- Scripts have a huge API set with all the functions needed for interfacing with all the project resources, in addition to the using the Windows API.
- Supports the Dialog Box and Message Box managements
- Supports scripting for managing Macros and Wizards in design mode as well
- Use of Debugger management in both design and runtime modes

## Basic Script Resource

---

The Basic Script resources are code portions which can be executed on command, event or automatically with specific project settings.

There must be a Sub Main procedure in the Basic Scripts that gets executed when the Basic Script is called. If this Sub Main is not defined, the code will not get executed.

Basic Script resources can be executed within the Main Movicon Thread or in a separate thread. However, only one Basic Script at a time can be executed in the main Movicon thread whereas scripts set in separate thread can be executed at the same time.

The main Basic Script properties in addition to those listed above are:

- Each script supports separate thread execution
- Each script supports controlled execution in redundancy systems
- Each script can be "instantiated" many times simultaneously
- Scripts can be defined "Public" and parameterized
- Each script provides the project control and debug with a Status Tag
- Possible to define where to define "Debug.print" output address
- Possible to execute a script in a new project instance (safe mode)

## Script Code in objects

---

Script can be associated to events made available from Movicon symbols, screens and alarms. However, in this case sub mains do not exist, and code is managed exclusively based on the events (SymbolLoading, Click, etc) available in the code and that have been inserted by the programmer. Movicon calls these events only when the symbol (or screen) is active, therefore when screen is loaded into memory. The script functions that can be used in object script code are the same one that can be used in Basic Script resources as well. The only thing that changes is the way the code is called. In addition, scripts associated to objects are always executed within the main Movicon thread and the events run in sequence with each other.

Events as well as properties and different methods can be made available for use according to the object type in which code is being edited.

### 21.1.1. Restrictions

---

Restrictions to inserting and executing Basic Script resources in projects are:

Sytem	Restriction
Win32/64	Max 4096 Basic Script, of which a max. of 128 are executable in runtime at the same time.
WinCE	Max of 64 Basic Script, of which a max. of 4 are executable in runtime at the same time.
Web Client	Not available (executed on server).
WinCE	The system's API are reduced in Windows CE. Please refer to the VBA Programming Manual
WinCE	The VB.NET syntax is not supported in Windows CE
WinCE	The ActiveX libraries are not always available in Windows CE. Please check ActiveX product makers.



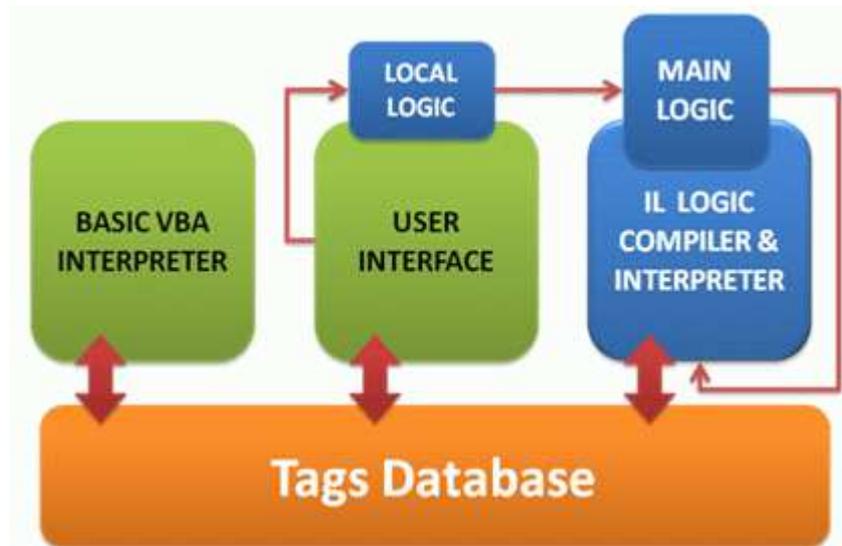
Movicon controls and corrects any syntax programming errors. Nevertheless, logic program syntax may be syntactically structured correctly but programmed or optimized incorrectly. Therefore, the full responsibility and reliability lies on the programmer's skill in programming.

## 22. IL Logica

---

The Movicon IL Logic permits the use of PLC type logic tasks in projects for managing the control of variables or I/O.

The **IL Logic (Instructions List)** is the key feature of the Movicon system. This resource can have all the effects of a PLC (Programmable Logic Controller) which runs its program in background independently of the system's logic, graphic or historical engines.



The IL Logic offers a number of obvious advantages. Basically it allows you to perform any type of non-standard executions using internal PLC type programs.

The IL Logic can be associated to projects, where it is known as **General Logic**, or it can be associated to symbols (drawings and controls) and to screens where it is defined as **Local Logic**. The fundamental difference between the General Logic and Local Logic are:

- At project startup the General Logic is run and continues running cyclically until the project stops. The Local Logic, however, is run only when the object containing it is active, i.e. when loading a screen in RAM
- The General Logic run has priority over the Local Logic

The Movicon Logic Editor provides you with a powerful set of logic, mathematic or compare instruction for combining variable with each other in order to activate logic commands which form the base of the project.

The Movicon programming logic language is in instruction lists (known also as AWL or IL Instruction List) which is very similar to the Simatic© STEP.

This type of language requires a vertical list of logic or compare instructions which, combined with each other, determine a binary result referred to as the "**Combined Logic Result**" or "**CLR**".

There are many instructions, grouped by functionality that can be used in the IL Logic:

- Bit Logic Instructions: allow binary logic variables, available in the system, to be combined in sequence with each other to create a CLR for activating following commands.
- Word Logic Instructions: allow logic variable combinations in byte, word or doubleword, which are available in the system, in order to allow data up to 32 bits to be loaded, transferred or manipulated.
- Compare Instructions: perform ACCU1 and ACCU2 content comparisons.

- Arithmetic Instructions: process and modify the ACCU1 and ACCU2 accumulator register contents.
- Mathematic Instructions: process and modify the ACCU1 and ACCU2 accumulator register contents.

**Main Features**

- Integrated native PLC type Logic Language in instruction lists (IL or AWL), similar to the Simatic Step7 language
- Combinatory Logic (instruction executions based on the CLR status (Combined Logic Result) executed in background independently from supervisor functions.
- Use of two 32 bit Accumulators for executing logic calculations
- Possible to use constant values in integer, hexadecimal integer and floating point formats.
- Comments can be inserted in at any point of the code preceded by the "//" slash characters
- Different bracket levels containing logic instructions can be used
- Possible to manage Jump to instructions to Jump to Labels that have been defined in program.
- Possible to process logic using floating point or integer values
- Possible to interact with system using "System Variables".

**22.1.1. Restrictions**

Here are some restrictions to take into account when creating IL Logic:

Funzionalità	Restriction
NON supported Variables	<ul style="list-style-type: none"> <li>• <b>Double type Variables (double 64 bit precision):</b> these variables cannot be used because ACCU1 and ACCU2 accumulators are 32 bit registers.</li> <li>• <b>String and Byte Array Variables:</b> these variables cannot be used because not supported in IL Logic.</li> <li>• <b>Variables with names beginning with "K":</b> these variables cannot be used because Movicon would expect to load a constant. If "Kf1" variables do exist in the Real Time DB, Movicon would not know whether to load the constant or the variable. Therefore variables starting the letter "K" are not supported in IL Logic.</li> </ul>
Number of logic lines	Win32/64: Max of 12,000 logic syntax lines WinCE: Max of 2048 logic syntax lines



Movicon IL Logic is executed in one unique multithread task in the same applied Movicon process. The scheduled logic times (indicated in the system Debugger) are not deterministic.



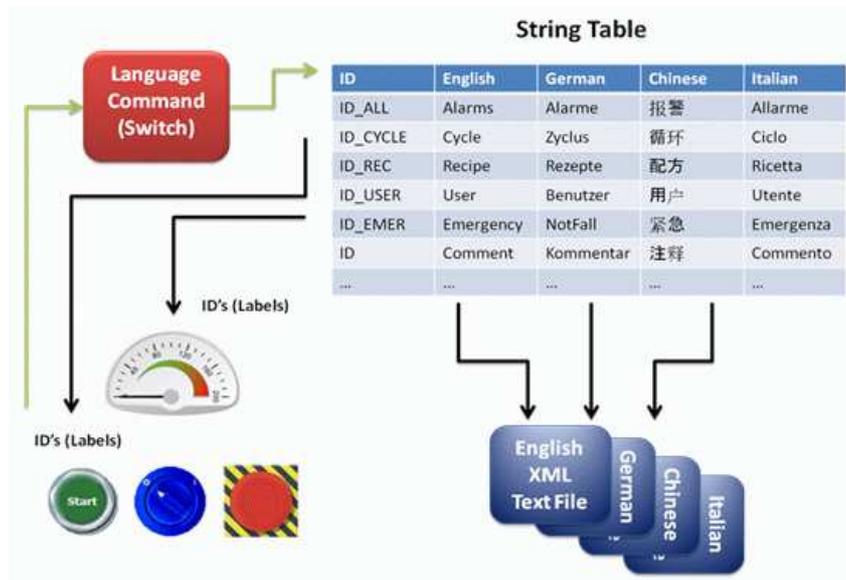
Movicon controls and corrects syntax errors. Nevertheless, logic program syntax may be syntactically structured correctly but programmed or optimized incorrectly. Therefore, the full responsibility and reliability lies on the programmer's skill in programming.

# 23. Strings

## 23.1. String and Language Change

To guarantee the complete support of the multilingual function, Movicon provides users with a purpose-built function used during runtime for changing languages of texts defined in the project. In order for this to happen, customized text has to be inserted in the "String Table" resource which must be indicated with a predefined "Key" ID code.

The String Table is the container of all texts to be managed dynamically in the Movicon project.



The Movicon strings consent projects to easily make use of any dynamic text and group text in different languages under one unique identifier by using the multilanguage functions.

The text strings are needed for handling alarms and messages, for displaying dynamic texts in the screens, and for handling customized popup type windows. When there is the possibility of representing a text dynamically, this text must almost always be contained in the Strings Resource. All the texts typed directly into the drawings or the Movicon components (and not inserted in the Strings Table) are presented in static mode in the resource representing them and therefore cannot be subjected to language change.

The Strings Table is a container divided into columns, each one representing a different language plus an ID column containing id string referring to the text written in the different language columns.

Once the strings have been added to the table, an ID string can be inserted corresponding to the text desired in the resource or object where it should appear. Movicon will automatically substitute the ID string by displaying the text contained in the column corresponding to the language active at that moment (whether in runtime or design mode).

### Main Features

- Texts can be managed in any string type and in any language, thanks to the UNICODE and UNICODE UTF-16 support.
- All project strings reside in external text files (Each language column has a XML file)
- Supports copy-paste of any string to/from Ms Excel™

- Change Language in runtime is immediate using commands associable to any object. Change Language is also immediate in design mode.
- Supports\n character for going to the beginning of texts.
- Automatic insert missing strings command (check missing strings): this command finds all strings used in the different project resources and those missing, being their ID, from the String Table. This command is available for all project resources and when activated automatically opens the string editor and adds the missing strings. Only the String ID is added therefore leaving the programmer to insert the texts in the different set languages used in the project.
- String Table Filter: all the strings on the table can be filtered within the string editor according to the text content in the "ID String" column.
- String Selection Window: project object or resource properties where texts can be inserted have a (...) button to the right which can be clicked on to open up a modal window, exactly identical to String Table, and select a string from those listed. This window can also be used for modifying, deleting existing strings, adding new ones or inserting new columns or deleting them.
- Formatting Strings through String ID: the String ID management also allows a number of ID strings to be linked into one text field using the @ character.
- Change Language in System texts: there are 'special' String IDs which can be used for managing change language when having to customize texts in some of the Movicon dialog windows (entering passwords, Filter window, Pad, etc).



Not only is the String Table indispensable for managing multilanguages, it is equally handy as a container for centralizing all the texts used in the project as well.

### **23.1.1. Restrictions**

---

<b>System</b>	<b>Restrictions</b>
Win32/64	Max 32,000 Strings for each language Max 128 Languages (Columns) Max 1024 characters per string
WinCE	Max 8192 Strings for each language Max 32 Languages (Columns) Max 256 characters per string

# 24. Security (Users and Passwords)

---

## 24.1. Security (Users and Passwords)

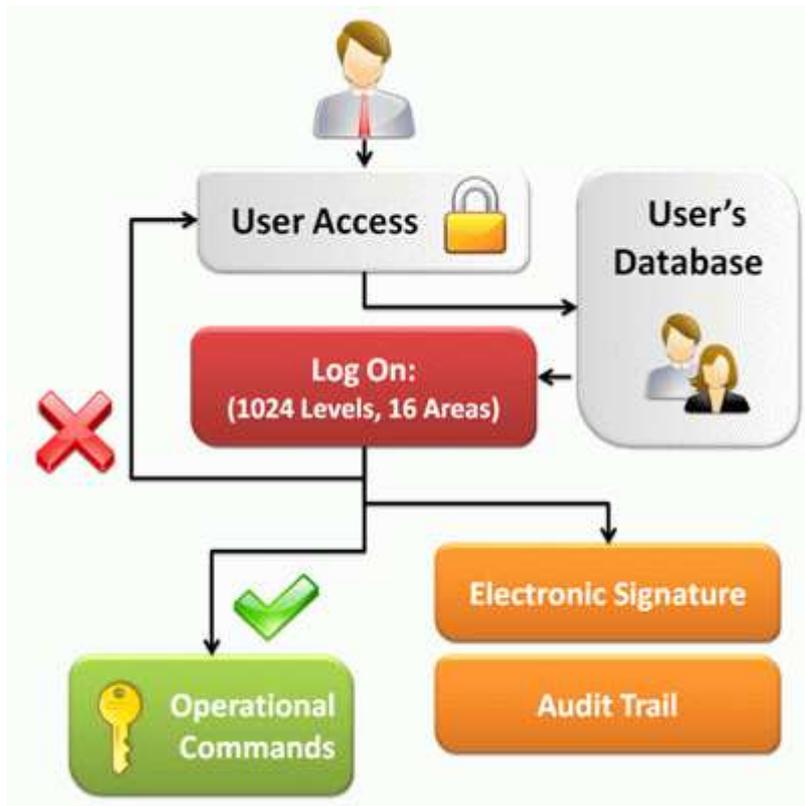
---

The User and Password management is used for protecting the project's various functions by consenting access in complete security with user Log On and audit trail recording all activities operations performed throughout project.

All the configurations relating to the project security management are defined in the **Users and User Groups resource**.

The Users and Passwords management is a supervision project to allow the possibility to manage supervisor access security, by declaring identification of users enabled to work in the system, by entrusting them with an exclusive password and access level.

The access privileges (hierarchical levels and Areas) can be issued normally or by group.



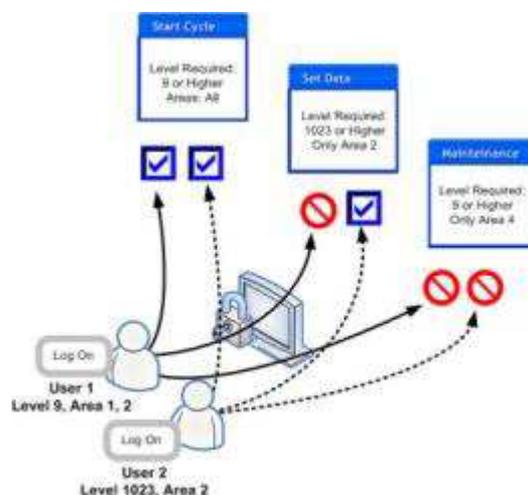
Access security is managed by enabling the password management using the Users and User Groups Resource general properties. Therefore, any command object can require user authentication (Log On) when defined with a hierarchy protection level in its Access Level properties. The will protect command activation by subjecting them to verify Log On access privileges of user trying to activate command before allowing them to do so.

## Protection Criteria

In the Movicon users management, each user is assigned a User Password Level (Hierarchy level) to define their access privileges, and an Access Level (Area) to define the area they can access to along with the read and write rights they are allowed to use in the different project controls in that area. The Access Level properties of each object define:

- **User Level 0 (0)** = no password level required (default)
- **User Level 1-1022 (1-1022)** = hierarchy password levels from 1 to 1022. These password levels are assigned to normal project users.
- **Administrator User Level (1023)** = password level for Administrator or System users. System operations can also be performed with this password level such as operate project closure, window controls, etc.
- **Developer User Level (1024)** = password level for Developer users. This password level can be used for accessing the project in Editor mode

There are 16 distinct levels, from 'Access Level 1' to 'Access Level 16'. In this case each Access Level, which can be activated by using a Check-Box, is independent of the others. For example, by enabling 'Access Level 5' on a Button control means that only the Users that have 'Access Level 5' checked on their 'Access Level' Masks can access this Button.



Project users must be inserted in groups. Some properties are made available both for User Group levels and single user levels. This eases propagating the same properties to the all users of the same group to then also differentiate these properties for each group user member or single user later on.

### User Profiles, Editor and Runtime

Groups and users belonging to these groups are inserted in the Users and User Groups resource. These users are inserted by the programmer and assigned properties and access privileges, including data relating to alarm notification addresses. These users will become part of the project and therefore inserted in the appropriated project configuration files (encryptable).

These users may be added to by Administrator Users in runtime. System Users inserted in Runtime will be saved in external files, independent from the project but will still have the same configurations possibilities as those defined by the programmer.

### FDA CFR21 Part 11

Movicon has been explicitly designed to support the FDA regulations defined by the CFR21 Part 11 specifications.

All the features provisioned by these regulations, which consent users to create FDA CFR21 Part 11 ready projects, have already been implemented in the Users and User Groups properties. There is also a Manual solely dedicated to helping Movicon users create FDA ready projects

## **Main Features**

---

- User Password Management with 1024 hierarchy levels and 16 access areas
- Manage especially designed for supporting the "FDA 21 CFR Part 11" normative
- User Group management, and individual Users can inherit Group properties
- Each User manages their own personnel and unique Name, Password and Electronic Signature
- Auto Log Off can be enabled with resettable times, both for groups and individual users
- Groups and individual users can be associated with one "LogOn Script", one "Logoff Script", one "Logon Command List" and one "LogOff Command List" that will be executed at User LogOn or LogOff
- Passwords can be set with expiry dates (expiring password)
- Passwords can be changed after first User LogOn
- Users can be associated with different phone numbers or email addresses for receiving SMS, FAX, emails etc. from the Alarm Dispatcher
- Project can be protected from opening in development mode by unauthorized users. Only users with the 1024 level (highest level) can open the project in development mode.
- Runtime Users Management in project. The Movicon Users management also allows users to be added, modified or deleted during project Runtime mode.
- Operating system domain users can be used in addition to defined project users.
- Languages can be associated for default to groups or individual users. In this case when the user logs, the language associated them will automatically activate.
- Pins (numeric password) can be associated as an alternative password for users to access mobile systems.
- Passwords can be fixed with minimum and maximum lengths
- Support to Operating System Domain user acknowledgement
- System management to protect system against unauthorized access, mishandling and intrusion attempts
- Any attempts made to mishandle or intrude system will be recorded in the Historical Log
- Child project users can be included in project

### **24.1.1. Restrictions**

---

<b>System</b>	<b>Restrictions</b>
Win32/64	Max 8192 Users, and Max. 512 Groups
WinCE	Max 1024 Users, and Max. 64 Groups

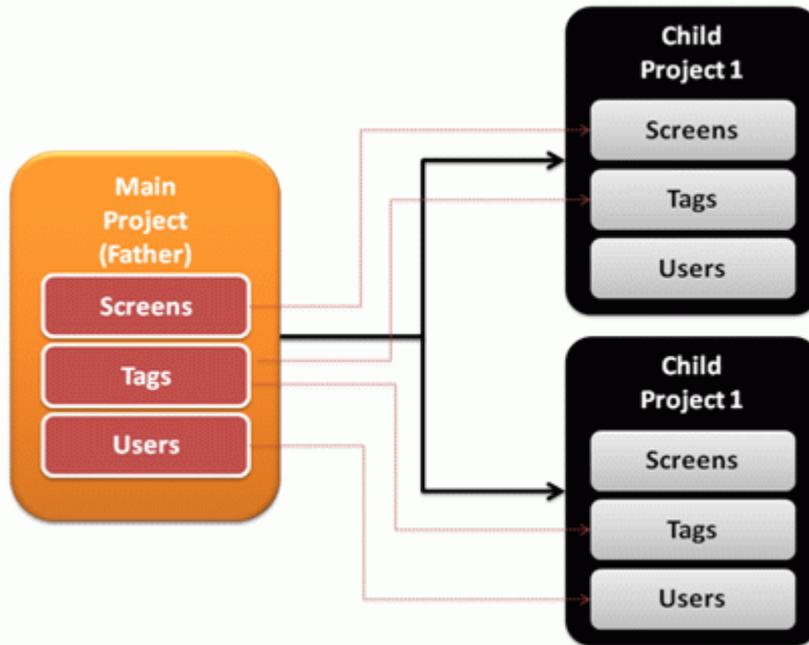


# 25. Child Projects

---

Movicon allows you to structure projects by decentralizing resources in other projects (child) with dynamic relationships, giving you the possibility to distribute your projects.

Movicon has a powerful and innovative feature, which allows you to face new challenges in planning supervision systems. **"Child Projects"** are normal Movicon projects which, even when planned to function independently, are linked to parent projects creating a "Parent-Child" relationship where the Parent project is provided with all the resources of the Child project as if they were its own.



## Distributed Projects

Projects structured with Parent-Child relationships provide many advantages to companies working in teams. In respect to the more conventional technologies, where many people work using and sharing tasks in the same project, Movicon offers the possibility to distribute work in different projects independently where the Team Leader can have, in their own Parent Project, all the resources of the Child Projects of its collaborators, who can also completely work independently.

## Distributed Run

The Father-Child Project relationship is very useful for modular systems or machines where, for instance, the plant is divided into zones which can also be independent from one to another. In a situation like this you can create more projects, one for each zone, and then integrate them into one Father project from which you can access the pages and the variables of the Child projects.

## Client-Server with Child Projects

Parent-Child Projects may also be very useful in situations when needing a Server station and a certain number <n> of the "same" Client projects. In this case, by using the "Parent-Child" relationship you can create child projects which can be easily be distributed in any PC network that won't need modifying if the Server has to be modified. In order to do this, just create an 'empty' Client project and setup a "Child" project within to correspond to the Server project.

The child project can reside in the Client physically (by making an exact copy of the Server project beforehand) or it can be directly linked to the Server project via net. This second configuration will allow you to use the same identical Client project in any PC connected in net. Any client can connect to the Server PC to run project files. Furthermore, this technique automatically provides you with software centralization: any modifications executed on the Server will automatically be available to all clients.

**Main Features**

---

- Structured and modular project planning thanks to dynamic links to "child" projects.
- "Child" projects are independent, linked to parent locally or in net.
- Client-Server management with "centralized" and independent Client projects.
- The "Parent" project accesses and displays the main child project resources (Tags, Screens, Users, strings).
- Resources can co-exist with the same name in "child" projects
- "Child" projects can be put into runtime mode from the "parent" project

**25.1.1. Restrictions**

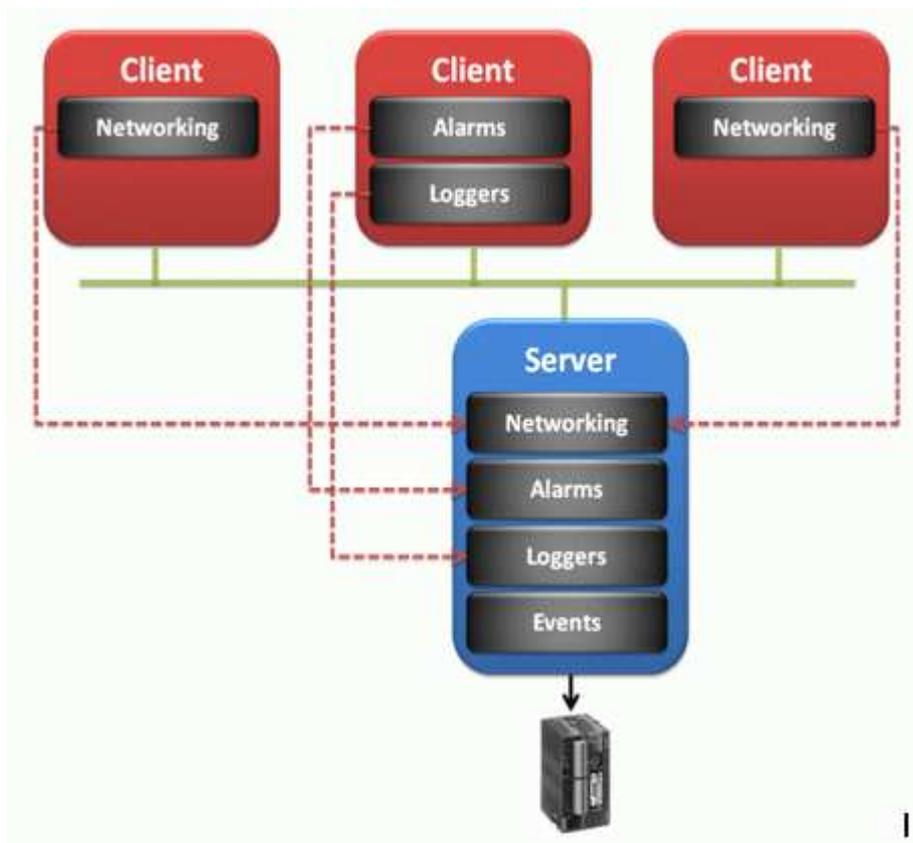
---

<b>System</b>	<b>Restrictions</b>
Win32/64	Max 32 "child" projects per each "parent" project
WinCE	Max 8 "child" projects per each "parent" project Note that WinCE does not consent disk sharing in network, therefore "child" projects cannot be distributed in net using WinCE.

## 26. Networking

---

The Networking exploits the potentialities of protocols capable of going on Ethernet networks for interlinking Movicon workstations in Client/Server mode. It is more than often that the distributed architecture of process control requires plants to include multiple workstations, control or display stations which are used through one or two PCs linked up together in Ethernet network. Today's Ethernet network is the most used means of communication due to its hi-performances/low cost ratio. This means, however, that more tools are needed for interlinking more stations capable of sharing process data. Movicon nevertheless is able to satisfy these Networking connectivity requirements by exploiting the TCP/IP, UDP, and HTTP protocols which are also used on the Ethernet network as being the most widely used and fastest.



Movicon permits one, a group or all the Real Time DB variables from the client project to be connected towards the same number of Real Time DB variables of another Movicon Server project application in the same network. Furthermore, all the client project window type objects, such as the "Alarm window", "Log window", "DataLogger window", "Recipe Manager", "TraceDB window", "Scheduler window", "Trend", "Data Analysis" and "Chart" can be connected to the Server project so that the Client can display data currently showing on the Server.

### Network Server

---

Each Movicon project enables the Network Server functions for default. However, these functions can be disabled through using the project's Network Services resource properties. The Network

Server function allows each Movicon station to supply information requested by any connected Clients (or any future clients) to the Server.

## **Network Client**

---

The Movicon project functionalities can connect to a network server. One of the main functionalities involves Tag connectivity. Each variable Tag has Networking properties that allow each Tag to be defined as connectable as network Client to the respective Server Stations. If not specified otherwise, the name of the Tag on Client side will be the same one to connect to on the Server side. Many other display objects, as mentioned above, can be managed as Clients and can connect to a Server station by specifying the Server in each objects' properties.

### **Main Features**

---

- Integrated Networking, simple and powerful with "event-driven" management. TCP-IP is the Protocol used for default.
- Direct connectivity in Tag networking.
- Server always enabled for default in projects. Does not require any license modifications.
- Client-Server connections through LAN, WAN and Wireless networks.
- Each project can be both Client and Server indifferently and simultaneously.
- Client-Server connection management also through using RAS remote connections (Remote Access Service). The RAS connection is an Operating System function that consents Server-Clients connections between two stations via Modem. Once connections have been established, the TCP/IP protocol can be used for exchanging data between Server and Client.
- Remote Project Debug: the remote debug function consents connection to another Movicon instance being run on a remote computer/device (Win32/64 or WinCE). This makes it possible to debug projects in remote and above all to debug projects being run on WinCE platforms.
- Windows Terminal Server: project can be created in a Terminal Server environment.

### **26.1.1. Restrictions**

---

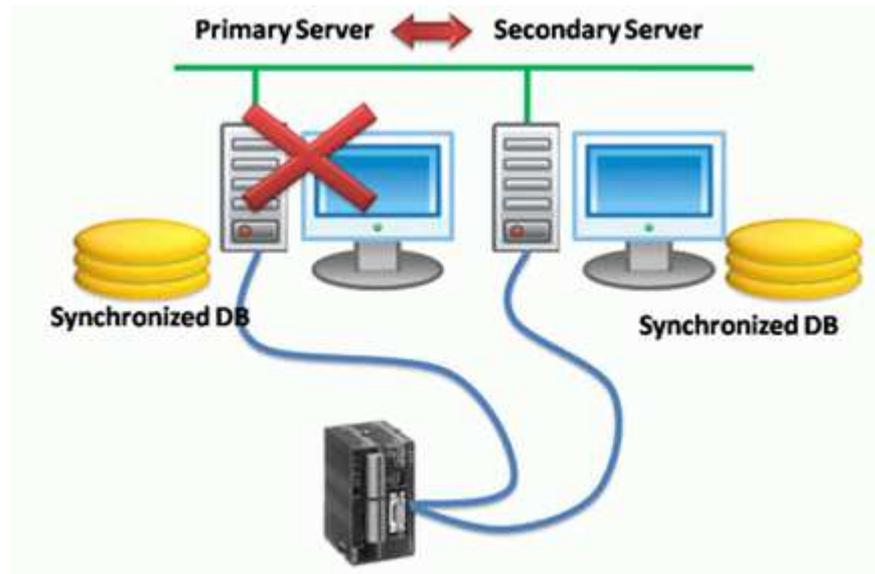
<b>System</b>	<b>Restrictions</b>
Win32/64	Max 32000 Network Connections in Tags
Win32/64	Max 32 Network Connections per Display (Alarms, Log, Trace, etc.)
WinCE	Max 512 Network Connections in Tags
WinCE	Max 8 Network Connection per Display Window (Alarms, Log, Trace, etc.)

The Networking Log function ("Enable Network Log" property) must remain enabled for the whole duration time needed for analyzing any eventual problems.

# 27. Redundancy

---

Movicon has powerful automatic functions to support the Hot Backup of critical workstations redundantly connected in network. Slave intervention, re-entry, data synchronization are managed in complete automatic mode. Some critical processes under supervision and control require emergency station intervention, known as Secondary Server, when the main PC unit, known as Primary Server, crashes. This ensures that the process's supervision and control service will continue to function when the Master supervision station unexpectedly becomes unavailable or crashes.



The redundancy management, whether with Master functionality or Slave functionality, is completely integrated in the Movicon software and ensures secondary system intervention in complete automatic mode after primary system timeout which can be defined in the system's configuration. The redundancy functions are imbedded and native in the following critical system functionalities in Movicon:

- Communication Drivers management
- Data Logger management
- Historical Log management
- Alarm management

Any one of these functionalities is kept on Stand-by in the secondary station, and go into action when the Primary station crashes to maintain system control. The Movicon redundancy control system requires two PC stations connected in network based on TCP/IP, with Primary and Secondary Server functions:

- **Primary Server:** is the workstation that manages the normal running of the system, by communication with it, collecting data and providing control. If this station crashes, the secondary station will take over immediately by entering into action and taking complete control.
- **Secondary Server:** is the workstation which manages the normal running of the system in redundancy mode through shared variable memory areas. This station allows independent system interaction and provides the exact same primary station functions. If the Primary server crashes, the secondary server will automatically take over system management by starting up the communication driver and recording engine functions, collect data and continue on with process control.

## Advanced Redundancy Functions

---

Movicon provides the user with addition and special built-in functions to use in the critical system redundancy management:

- **Synchronized Time Frequency:** The Movicon Redundancy has an automatic tool integrated in the synchronization control management of the two PC Servers: the automatic synchronization of the system's clock on the Secondary Server to tally with the Primary Server's.
- **System Diagnostics:** Movicon issues alert messages when Secondary Server enters into action, Primary Server re-enters and ongoing synchronized data etc. This information is displayed on both systems and recorded in the Historical Log automatically.
- **Status Variable:** The redundancy management also provides the programmer the option to use a status variable to inform logic on the Primary and Secondary Servers' operating statuses.
- **Switch on Driver Error:** this function permits the Secondary Server to activate its own Communication Drivers when problems occur with those on the Primary Server. This driver switch over can be done without the Secondary Server needing to take complete system control.
- **Cache File:** cache file sizes are completely customizable and are used by Movicon to save data to be synchronized between the two Servers.

### 27.1.1. Restrictions

---

The Redundancy management used in critical plant situations has some restrictions regarding project functionality that the programmer should take into account. These mainly involve the Movicon non-critical functions that do not get managed in redundancy conditions.

Function	Description
System Variables	The Movicon System Variables are not redundant.
Alarm Basic Scripts	The Alarm Basic Script interface is not managed in redundancy.
Trend Output	Trend output files do not get managed in redundancy, due to the fact that Trend connections to the Data Logger are managed for this.

Furthermore, you must keep in mind that while the IL Logic associated to the project is executed only on the active Server, the screen or designs' Local IL Logic is always executed on both the Primary and Secondary Servers independently from which one is active. To avoid any problems you will need to manage the Local IL Logic shrewdly by using, for example, the screens' local variables.

Also be reminded that the Event and Alarm managements are disabled in the Secondary Server when in 'Stand-by' mode. For instance, this will mean that Alarm Areas used for symbol animations ('Variable Is Alarm Group' property from the symbol's Dynamic 'Background Color' and 'Dynamic Text and Edge Color' property groups) will not have any effect in the Secondary Server when in 'stand-by'.

# 28. OPC Client & Server

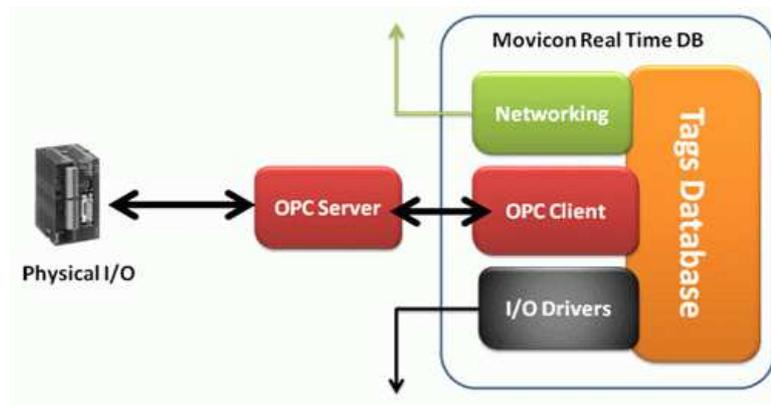
---

OPC is the most commonly used standard due to its user-friendly communication modalities based on different bus technologies.

OPC stands for "OLE for Process Control", where OLE refers to the Windows' abbreviation of: "Object Linking and Embedding", a powerful automation interface for the Windows applications.

OPC simply rounds off the programming concept which implements a unified interface of different bus technologies on one part and programs for vision and automation on the other. Thanks to the standard technologies, today's communication between automation devices means more independence for the individual producer, especially when their apparatus supports the OPC standard. Thanks to the unified interface, hardware producers can always guarantee OPC interface availability by equipping their hardware product with OPC Server software products. This will ensure the user of being able to interface their purchased product with any applied software supporting the OPC standard.

Movicon is an application based both on the OPC Client and OPC Server standards. As Client it can interface with all the OPC Servers which follow the 1.0A and 2.0 standards; as Server it allows all or part of the project variables to be shared with other applications and notifies them of events such as: Alarms, messages, system information and variable changes within the project.



## OPC Client

---

In the OPC communication Movicon has provided a Client interface which can be entirely configured and which supports any OPC Server created according to the OPC 1.0A and 2.0 specifications. This system integrated resource is identified with the "OPC Client DA (COM)" name and can be accessed through the Movicon "Project Explorer" window.

By using this resource you can setup communications in OPC with one or more Servers. Movicon displays a list of Servers available on the local computer, but you can also use OPC Servers installed on other remote computers connected in net or on the web.

Two OPC Server interfaces are currently supported: "DataAccess" and "Alarms&Events", identified respectively as "DA" and "AE".

Once the Movicon variable is connected to the OPC Server's item, you will be able to customize its configuration by setting various properties:

- Create OPC groups in which to insert defined items
- Group refresh times can be set and deactivated when not in use
- Items can be set as read, write or read-write
- Item reads can be forced at start-up

## OPC Server

---

In addition to the OPC Client, Movicon has also integrated the OPC Server functions, making events available and notifiable to any OPC Client applications connected. The Movicon OPC Server supports Clients created with the "DataAccess" and "Alarms&Events" OPC standards.

This technology permits the sharing of project variables with other applications with OPC Client DAs and at the same time it can notify Client AEs of events to do with Alarms, Messages, system information and variable changes in the applied project's real-time Database.

### Main Features

---

- Supports OPC DA and OPC XML technology
- Supports OPC Client and OPC Server technology
- Direct connectivity in Tag networking
- Integrated OPC Server Item Browsing

### 28.1.1. Restrictions

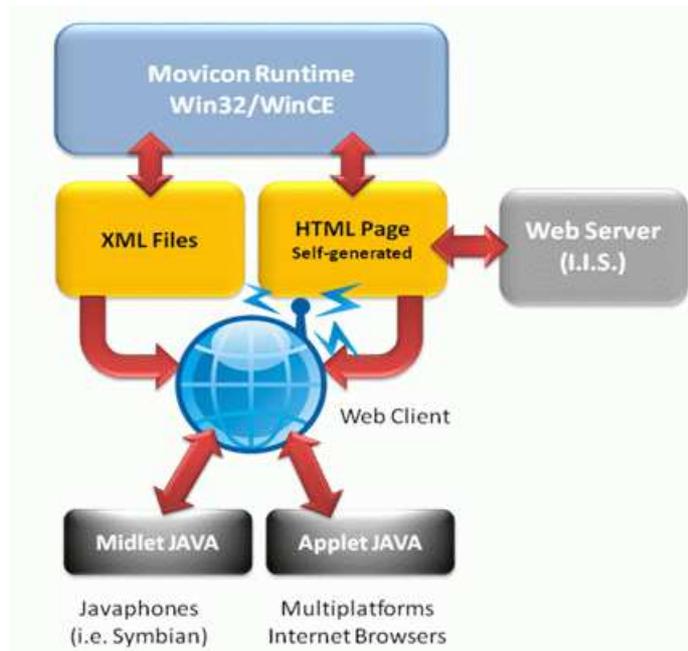
---

System	Restrictions
Win32/64	Max 8192 Connections via OPC Client/Server in Tags
WinCE	Max 512 Connections via OPC Client in Tags  The OPC Server and Client must reside on the same device, they cannot be connected via network.

# 29. Web Client

---

Being able to provide factory information in real time where and whenever needed is an essential factor which every production company is well aware of. Thanks to the Movicon® and Movicon Web Client© technology you can visualize and control any Movicon supervision station even by using an ordinary browser on the client, drastically cutting down on management, maintenance and licensing costs while increasing your company's productivity at the same time.



Movicon Web Client© is based on the Java Thin Client architecture and exploits the web Services communication for visualizing and/or controlling your automation system from any client station in LAN or WAN.

The advantage of using the Progea designed Web Client solution is that none of the complicated and tedious installation or configuration procedures are needed on the Server and clients are guaranteed with multiuserness, bidirectionality and security all at the same time.

When the Server is equipped with a Web Server (i.e. Windows Windows 32/64 bit includes IIS - Internet Information Server), client access is totally transparent with automatic downloading of the applet upon first browser connection. When the server is a Windows CE station, clients can access by configuring applet on Clients appropriately.



System Web Server stations can therefore be both Movicon Scada-based for Win32/64 versions and Movicon HMI-based stations for WinCE versions.

Movicon Web Client offers the possibility to access system data by using the Internet/Intranet architectures.

## Main Features

---

The advantages you get from using the Movicon Web Client are:

- Extreme simplicity and inexpensive to use.
- No need for any installations or configurations on both server and client.
- Supports Multiplatforms on Web client side thanks to the Java technology.
- System server both in Windows Windows 32/64 bit and WinCE.

- Supports graphics auto-adaptation to video screen resolutions or window sizes on Web client side.
- Optimized communications supporting also geographic networks with scarce band rates.
- Event-driven communications.
- User Log On Authentication on Web Client side.
- How Web Client is to be managed can be defined on server: display only, display and command, command with passwords management.
- Objects to be shown on Web Client side can be chosen and selected on server.
- Maximum security thanks to the JAVA technology and Web Services (no ActiveX on clients).
- Software centralization. One only applied project, residing on server, in a Web Client architecture.
- Communication based on SOAP technology and Web Services, with performances and "firewall friendly" security.
- Bidirectionality, performances with automatic data refresh, historical log analysis, data settings, command access, password, multi-userness.
- Clients also for mobile JavaPhones telephony.
- No license needed on Client side, no installation, no maintenance and no ActiveX needed.

### **29.1.1. Restrictions**

---

When planning applications for Web Client devices you should fully consider the unavoidable restrictions users have **on Client side**. Even though the Web Client technology offers many advantages it also has some restrictions in respect to the similar Runtime Client technology, where the Client processes its own project autonomously in respect to the Server. In a Web Client architecture, however, the Client does not process anything locally, as each Web Client instance is processed on the Server.



You will also need to consider whether the Web Client is to be used for PC platforms (J2SE) or mobile telephony (J2ME).

Furthermore, the project planner developing the Server project for Web Client access, must consider the following guide lines:

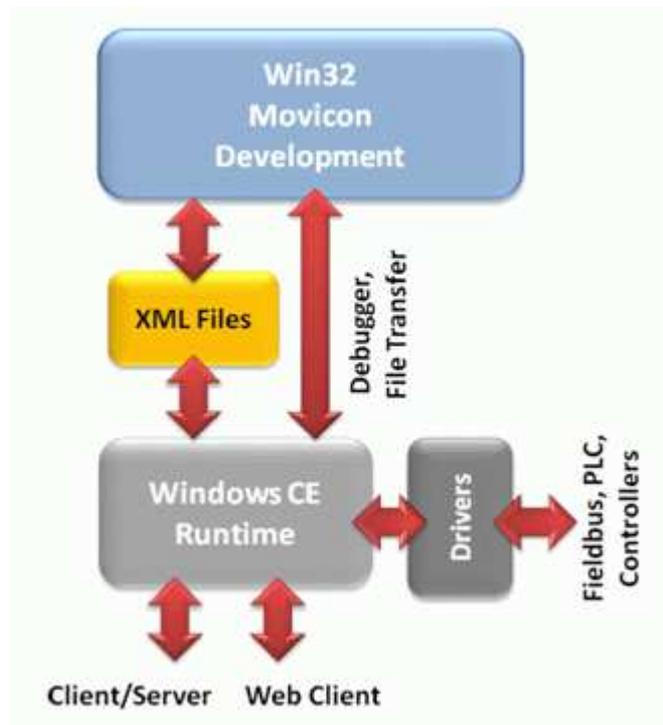
- The Server processes the information for its Clients, always guaranteeing priority to local executions over remote management.
- The hardware resource for the Server must always be proportioned to the amount of resources to be put into use for both the application being run and by the number of Clients which can connect at the same time.
- The quantity and complexity of the screen graphic information which can be accessed via Web Client must be in proportion to Client's capacity.
- You need to consider that the Client can get access to screen pages not necessarily displayed on the Server.
- Security on the Client side depends on the project configurations defined on Server side.
- Each Client user has independent and autonomous access to the Server, whether from other Clients and Server.
- In cases where the Server is based on Window CE, you should always consider the resources provided on that Server usually being quite scarce. We suggest you keep this in mind when creating screen pages accessible via the Web.
- Access to the Server via Movicon Web Client technology is also possible in stations without Web Server.

<b>Sistema</b>	<b>Limitazione</b>
Win32/64	Max 64 Web Clients connected to one server <b>at the same time.</b>
WinCE	Max 2 Web Clients connected <b>at the same time.</b>

# 30. Movicon for Windows CE

---

When engineering applications for WinCE devices make sure you are aware of those functions not supported by the CE device. Although the non supported functions are "masked" when selecting WinCE as the target OS, you should however check the hardware device's calculation capacity and memory use limits and assess whether this is compatible with your project needs.



## Main Features

---

The advantaged of using Movicon with Windows CE are:

- Great flexibility of use by being able to have Movicon runtime engine for Windows CE/Mobile as well
- Same editor availability: one unique programming environment for projects whether for Win32/64 or WinCE/Mobile
- Hardware independency: Movicon CE is available for platforms with x86, ARM, RISC, MIPS, processors etc.
- Real Time Database management with numerous communication drivers, networking and OPC Client
- Graphical interface management with vectorial editor supporting raster images (BMP, GIF, JPG, PNG, etc.). Supports 90% of the powerful Movicon graphics functionalities.
- Power Templates and Symbol Library
- Complete Alarm management, with the same Movicon functionalities.
- Alarm Log management both in relational DB (SQL CE) and IMDB
- Alarm Dispatcher management (SMS and Emails only)
- Data Loggers management with data recorded on event, time or change. Supports relational database (SQL CE) or IMDB
- Trend historical and Data Analysis
- Textual Reports

- Event Objects
- Scheduler Objects
- IP Camera Viewer Objects
- Password management with 1024 user levels and 16 access areas
- Security management, CFR21 and Audit Trail
- Basic Script VBA engine
- IL Logic
- Debugger on device from development ambient
- Web Client

## Programming Constraints

---

When creating projects for Windows CE you will need to take into account the operating system's limitations as well as those mentioned below:

Constraints for devices with less than 128 MBytes of memory or those using Windows CE 5.0.

Memory use in devices with Windows CE is usually very limited. We strongly advise you to read the following tips when using devices with Windows CE 5.0 or with Windows CE 6.0 which, however, has less than 128 MBytes of memory:

- **Avoid using more than 2 objects with script code per screen and not more than 2 basic script resources in separate threads.** It would be best to not use or limit use of basic script code in designs. The WinWrap engine does in fact require 2.5 MBytes of memory + 300 KBytes for each object or resource containing basic script code + 700 KBytes if the basic script resource is checked in a separate thread. This memory cannot be allocated in extra process memory and therefore the process memory may exceed the 32 Mb limit set by the Windows CE 5.0. operating system. If you cannot avoid using script code, you should try to use it in one point only, mainly being the screen containing the objects.
- **Limit basic script code in alarm thresholds to not more than 10 alarms.**
- **Avoid using basic script code for managing communications.**
- **Do not use the IMDB in cases needing to record large amounts of data exceeding total of 5000 records the overall total amount allowed for the historicals: historical log, data logger, recipes, trace db.**
- We suggest you set the "Shared Table" option for each IMDB database as to not occupy process memory (limited to 32 MB in Windows CE 5.0).
- **Use images in bitmap format with max. resolution of 256 colors.** Movicon CE uses a cache for bitmap images used in drawings which once loaded will remain in memory even when screen is not loaded. Only those images set as screen background will get unloaded from memory when not used.
- **Disable the "Network Server" option in the Networking settings if the project has nothing to do as Server.**

Constraints for devices using Windows CE 5.0 or Windows CE 6.0.

In general, you need to take into account the following constraints for all devices using Windows CE:

- **You should limit the use of animations that need updating graphically every second and do not use more than 5 objects per screen (quick blink, movements, resizing, animations with bitmaps, etc.).**
- **Make more use of objects from the Movicon toolbox (Object window) and less use of Templates from the Symbol Library which are mostly complex and contain basic script.** The reactivity of Template commands are, in fact, not as fast as those in standard objects.
- **Try to centralize the execution of IL Logic within your projects instead of having code portions spread out all over the place in different screens and symbols.**
- **Try to avoid using Synapse logic.**
- **Avoid using the Windows CE panel as Server for more than 2 Web clients. Furthermore, the Web client page sizes should not exceed the panel's resolution.**
- **Avoid using the Windows CE panel as Network Server for more than 5 Network Clients.**

- **Avoid recording data at a frequency inferior to one record a second.**
- **Always check that there is sufficient remaining compact flash space for recording the total amount of historical data programmed in the project.**
- **Use the heap manager only when necessary, if the "\_SysVar\_:MemoryAvailVirtual" system variable nears or goes below the value of 5.242.880 bytes.** When going below this value, Movicon will enter into 'low memory' status condition. Please refer to relating documentation.

Further details of the Movicon constraints with Windows CE are described in the Movicon CE Reference Manual.

Graphics Restrictions  
Alarm Restrictions  
Data Logger/Recipe/Report Restrictions  
WinWrap Basic Functions Restrictions  
IL Logic Restrictions  
Debugger  
Redundancy Restrictions  
Driver Restrictions  
OPC Restrictions  
Web Server Restrictions  
Web Client Restrictions

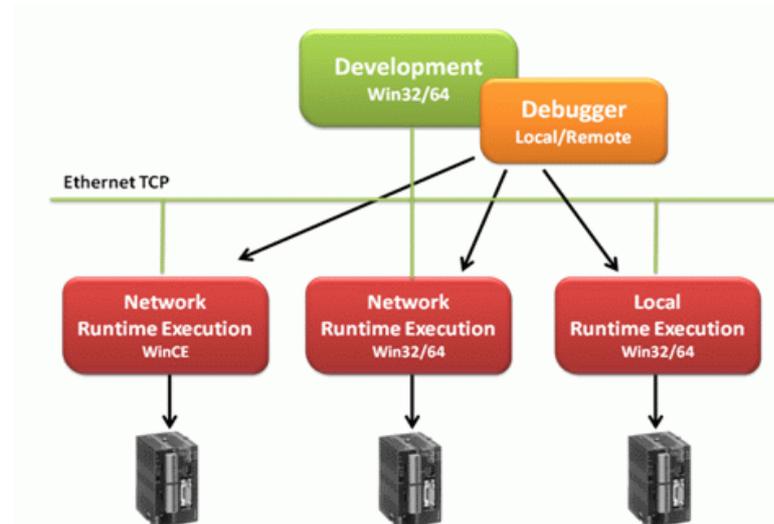


# 31. Debugger

---

Movicon provides users full use of its project debug and diagnostic tools. These tools include:

- **Output Window:** available in design and runtime mode
- **Watch Window:** available only at project startup in runtime from the development environment (Design mode)
- **Script Debug Window:** available in design mode for the Basic Script Resources. Available in runtime mode only at project startup from the development environment (Design mode) for Basic Script resources and code in objects.
- **Refactoring Window:** available in Design mode
- **Attach to Process Function:** only starting up project in runtime from the development environment (Design Mode)



## Output Window

---

The Movicon **"Output"** window is used for displaying system and error messages which are generated by Movicon during the development or Runtime phase. These messages are also saved on log files during the Runtime phase as usual, but by using this window you can get a more immediate view of the situation which is especially handy in the project debug phase. At the bottom of the window you can select different TABs for viewing messages from different groups.

## Watch Window

---

The Movicon **"Watch"** window is used for displaying the status of the project, logic debugging and forcing variables during the project Runtime phase. This window is only available when the project's Runtime is started up from the development session.

The **"Watch"** window is composed of groups, each one contains specific information:

- **Watch:** This group is composed of different TABs. The 'Project' TAB contains information on the project's general characteristics, such as the number of variables in use, the number of active screens, the number of connected Clients, etc. The other TABs, called 'Watch 1', 'Watch 2'. etc., allow one or more variables to be selected for monitoring or any modifications.
- **Project Statistics:** This group documents the use of the resources retrieved by the system by representing them on a table correlated with pie charts.

- **Locals:** This group allows information on the on screen components to be displayed. When clicking on an object on screen, the window will change its contents by displaying information based on that component.
- **IL General Logic:** This group allows the Debug of the IL Logic resource associated to the project to be run.
- **IL Local Logic:** This group allows the Debug of the IL Log associated to the component or screen selected with the mouse to be run.
- **Script:** this group lists the Basic Script resources loaded in memory with some useful information such as status (running or stop) and total execution time of last run or the run in progress.
- **Alarm Script Debugger:** This window is opened by using the "Alarm Script Debugger..." command from the "RunTime Bar". This window lists all the alarms managed in the project. When selecting one of these alarms, the debug window of the Basic Script code inserted in the alarm's threshold will open.

## **Script Debug Window**

---

The script debug window is in fact the "Script Explorer" window used in debug mode. Basic Script resources can also be debugged in project design mode. However, in both the basic script resources and script associated to objects, screen and alarms can be debugged in runtime mode. The script debug window is nevertheless available in runtime only if the project has been started up from the development environment (Design Mode).

## **Refactoring Explorer Window**

---

The "Refactoring Explorer" window, once displayed, allows you to view any errors that may have arisen for the resource you selected. The errors refer to variables inserted in the properties of the resources and objects which have not been defined in the Real Time DB, or can be due to objects placement out of the screen area (see "Outside Screen Objects"). This control is active for all the project's resources which can be associated variables. By selecting a resource, i.e. a Screen, the Refactoring Explorer window will be populated with the errors found for that screen or for the objects it contains.

Variables used in basic script code are not retrieved by the Refactoring Explorer window. Only "State Variables" set in the basic script execution properties are retrieved by the Refactoring Explorer Window.

## **Attach to Process Functionality**

---

The Attach to Process or remote project debug, allows you to connect to another Movicon instance being run on a remote computer/device (Win32/64 or WinCE). This feature is used for debugging a project in remote and above to debug projects run on WinCE platforms. This functionality exploits a major part of the features provided by the Movicon Networking resource.

When activating the "Attach to Process" command, a dialog window opens for inserting the Server to be connected to. This Server can be indicated by its set IP address settings or name, or selected from the tree listing the available network connections to which the computer is linked.

The remote debug features offered are:

- The Watch window allows the data relating to the Server project to be displayed and to monitor the variables on the Server
- The 'Scripts' Watch window allows you to verify which Scripts have been loaded in memory and the total execution time of those in run mode. The debug window is opened by double clicking on a script
- Debugs can be run locally and in the project's IL logic
- Possibility to restart remote processes (not available for the service). When disconnecting from the Server a window appears to allow user to restart project on Server if need be
- Analyses the project statistics with a purposely designed debug window (not supported on WinCE)

Other things to consider:

- The remote debug messages have the lowest priority and do not influence the normal running of the system where network traffic is concerned
- The project to be connected to must be enabled with the "Networking Server" option (as well as the "Debugger" option) in the Networking Services settings

- The Networking default transport is used for connecting in remote debug, this is the one selected in the "Client Network Settings" of the project to be connected to. This same transport must also be enabled in the Server's "Networking Service" otherwise any attempt to connect will fail
- The screen's file path and name can be entered in the command line instead of those of the project and are exploited by Movicon to execute the remote project debugging



# 32. Cross Reference

---

The Movicon Cross Reference list can be applied to both the project Variables and Screens and consents you to find out where a Variable is being used or from where the screen was opened.

## Variable Cross Referencing

---

The Cross Reference for variables allows you to single out the resources or objects which use specific variables. To compile or update the variable Cross Reference you can use the **"Compile Cross Reference"** command found in the Project Explorer's "Commands" pane or in the menu which opens by right clicking on the Real Time DB resource. Once the Cross Reference has been compiled a window will open within the workspace similar to the "List Variables Window", in extended form, but containing only those variables used with each one showing a link to the object or objects in which it is being used. You can also access this window by double clicking on the "List variables (Tags)" resource while keeping the SHIFT key pressed down. Another way in finding out where a variable is being used is by clicking the ""Variables used in ...." tree item available for each variable listed in the "List Variables (Tags)" resource.

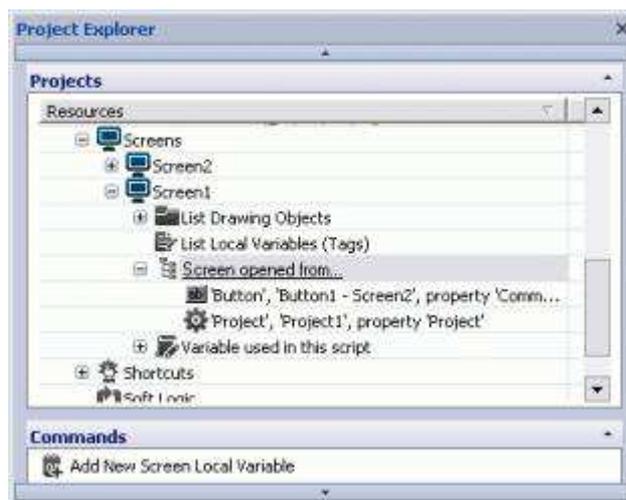
The variable Cross Reference has effect in all the project resources, even for those variables used in Communication Drivers, therefore in both in Stations and in Tasks.

In addition to this you can also compile Cross References for variables used in Basic Script codes on condition that they are used directly without using the "GetVariableValue()" and "SetVariableValue()" functions.

## Screen Cross Referencing

---

The Cross Reference for screens allows you to single out the resources or objects containing commands to open Screens. Unlike for Variables, a window does not display showing a Cross Reference for Screens. Instead, you can view the cross references of each Screen by expanding the "Screen Opened from..." tree item available for each Screen in the Project Explorer window. When this item is expanded (click on "+" node), and the Cross Reference has not yet been generated, you will be requested to compile and confirm the screen's cross reference which will then appear listed under the "Screen Open from..." item. You will then be able find out which resource or object contains a command to open the screen.





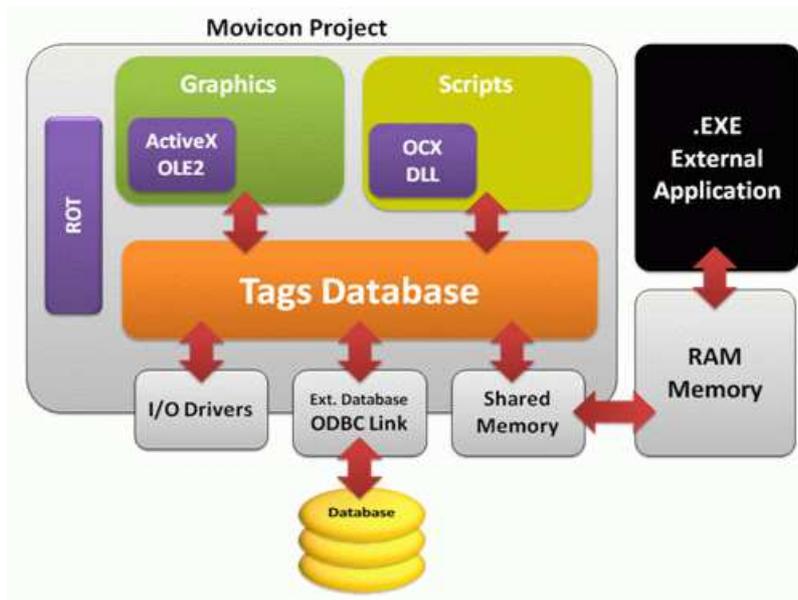
# 33. Interoperability

---

Movicon guarantees maximum openness and interoperability with third party software solutions or software components and external applications.

These technologies are:

- ActiveX, OCX, OLE Objects, DLL
- Windows ROT
- Shared DataBase
- Shared Memory



## ActiveX, OCX, OLE Objects, DLL

---

Movicon fully supports the Active X, OCX and OLE2 standards to allow you to integrate external applications into your project, whether as objects or as Basic Script support for event, method and property managements of third party software components.

In addition, you can use operating system libraries in scripts (API and WMI) or use appropriately created custom libraries.

## Windows ROT

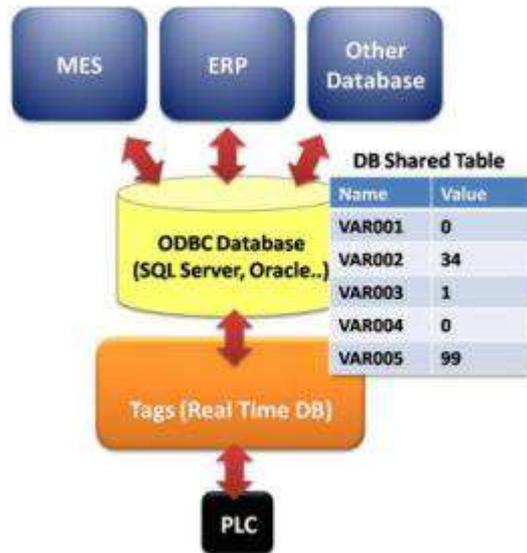
---

When this option is enabled the "PmeDocCmdTarget" interface will be recorded in the Windows' ROT (Running Object Table) each time a new project is opened for the first time. This interface is available to any script engine which supports OLE Automation (such as Visual Basic, by means of using the GetObject function). This allows you to keep the currently used "PmeDocCmdTarget" object so that you don't have to create a new one. By accessing the methods and properties of this interface you will also be able to access the project's variables in order to read or write them.

## Shared DataBases

---

This feature consents database variable value and statistical data sharing. This is mainly used for making variable data available to other applications capable of reading/writing Database files.



Movicon will create a table in the selected database format, in which a record will be reserved for each variable enabled with ODBC sharing. Each table column will report the current value or the statistical value of the corresponding variable.

### Shared Memory

---

Movicon provides a specific Communication Drivers through which one or more project RealTimeDB variables can be mapped in RAM. This shared memory can be accessed by other applications for reading and writing the same variables in the shared memory areas.

# 34. System Tools

---

Movicon uses some especially designed Progea tools to perform certain operations or functions. Those currently available are:

- **Upload/Download Service:** the "TCPUUploadServer.exe" executable file is installed with Movicon setup and also allows users to exploit services on desktop platforms for managing project upload/download to/from remote client stations..
- **Alarm Dispatcher:** Movicon uses this program, "Dispatcher.exe", for sending SMS, Voice, Fax, etc. for alarm event alerts.
- **CSV Tag Importer-Exporter:** This program, "CSVTagIE.exe", allows RealTimeDB variable to be imported/exported in CSV. files.
- **CSV String Importer-Exporter:** This "StringImpExp.exe" program consents strings to be imported/exported on csv. Files from the String Table.
- **PrnWnd.exe:** This "PrnWnd.exe" program is installed in Movicon installation folder and consents screens or the Window desktop to be printed. The raster print is a graphical print which generates a uniform bitmap of the page and sends it to the printer.
- **Tag-Alarm Wizard:** This "AlarmGen.exe" program, allows lists of alarms to be created quickly by using a wizard.
- **CleanLog:** This "CleanLog.exe" program is used by Movicon for deleting old networking log files exceeding a preset life cycle. This date is calculated by Movicon according to the project's "Log and Security" property settings from the Networking Services resource.
- **ReportViewer:** This "ReportViewer.exe" program is used by Movicon for displaying Report files created with Crystal Report and associated to DataLoggers, Recipes, Historicals etc.
- **ReportViewerNET:** This "ReportViewerNET.exe" program is used by Movicon for displaying Report files created with Report Designer and associated to DataLoggers, Recipes, Historicals etc.
- **Run Time Users:** This "EditUsr.exe" program, is used by Movicon to create and modify the RunTime users.
- **Language Manager:** This "LangManagerPlus.exe" program is used for changing languages in the Movicon development environment.
- **Development Code Info.:** This "RegDevCode.exe" program is used for displaying the Movicon development (or registration) code.



Movicon™ is a trademark of Progea, related to the HMI/SCADA platform entirely developed and produced by Progea. © 2012 All Rights reserved.

No part of this document or of the program may be reproduced or transmitted in any form without the express written permission of Progea.

Information in this document is subject to change without notice and is not binding in any way for the company producing it.



Via S.Anna, 88/E  
41100 Modena - Italy  
Tel. +39 059 451060  
Fax +39 059 451061  
Email: info@progea.com  
Http://www.progea.com



Via XX Settembre, 30  
Tecnocity Alto Milanese  
20025 Legnano (MI) Italy  
Tel. +39 0331 486653  
Fax +39 0331 455179  
Email: willems@progea.com



Progea Deutschland GmbH  
Marie-Curie-Str. 12  
D-78048 VS-Villingen  
Tel: +49 (0) 7721 / 99 25 992  
Fax: +49 (0) 7721 / 99 25 993  
info@progea.de



Progea International Ltd  
via Penate 16  
6850 Mendrisio - Switzerland  
tel +41 (91) 9676610  
fax +41 (91) 9676611  
international@progea.com



Progea USA LLC  
2800 East Enterprise Avenue  
Appleton, WI 54914  
Tel. +1 (888) 305 2999  
Fax. +1 (920) 257 4213  
info@progea.us