COMPU SYSTEMS

SCADA Training Manual

CompuSystems Training Center

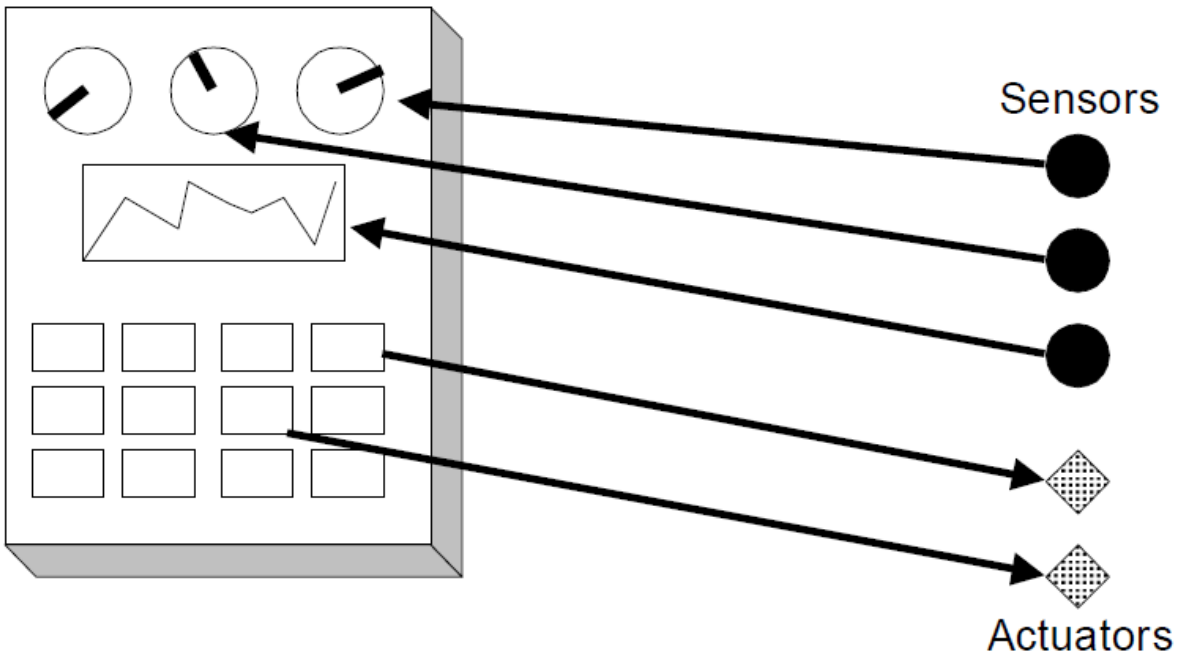
Table of Contents

Chapter One “Introduction to SCADA Systems”

1.1. Introduction & Brief History of SCADA

This chapter is designed to provide a thorough understanding of the fundamental concepts and the practical issues of SCADA systems. Particular emphasis has been placed on the practical aspects of SCADA systems with a view to the future. Formulae and details that can be found in specialized manufacturer manuals have been purposely omitted in favor of concepts and definitions. This chapter provides an introduction to the fundamental principles and terminology used in the field of SCADA.

It is a summary of the main subjects to be covered throughout the manual. SCADA (supervisory control and data acquisition) has been around as long as there have been control systems. The first 'SCADA' systems utilized data acquisition by means of panels of meters, lights and strip chart recorders. The operator manually operating various control knobs exercised supervisory control. These devices were and still are used to do supervisory control and data acquisition on plants, factories and power generating facilities. The following figure shows a sensor to panel system.



Sensors to Panel old days SCADA System

The sensor to panel type of SCADA system has the following advantages:

- It is simple, no CPUs, RAM, ROM or software programming needed
- The sensors are connected directly to the meters, switches and lights on the panel
- It could be (in most circumstances) easy and cheap to add a simple device like a switch or indicator

The disadvantages of a direct panel to sensor system are:

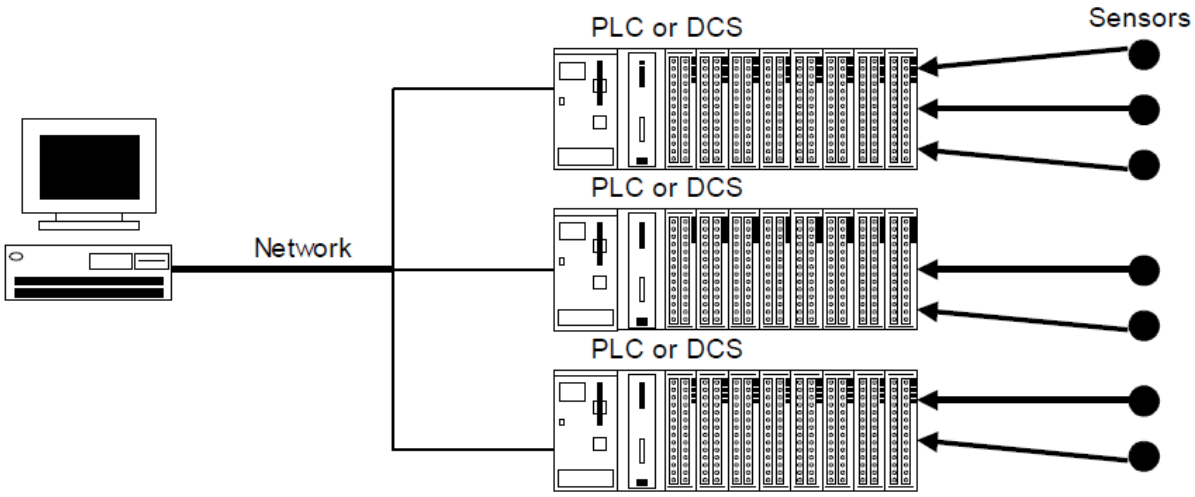
- The amount of wire becomes unmanageable after the installation of hundreds of sensors
- The quantity and type of data are minimal and rudimentary.
- Installation of additional sensors becomes progressively harder as the system grows
- Re-configuration of the system becomes extremely difficult
- Storage of data is minimal and difficult to manage
- No off site monitoring of data or alarms
- Someone has to watch the dials and meters 24 hours a day

1.2. Fundamental principles of modern SCADA systems

In modern manufacturing and industrial processes, mining industries, public and private utilities, leisure and security industries telemetry is often needed to connect equipment and systems separated by large distances. This can range from a few meters to thousands of kilometers. Telemetry is used to send commands, programs and receives monitoring information from these remote locations. SCADA refers to the combination of telemetry and data acquisition. SCADA encompasses the collecting of the information, transferring it back to the central site, carrying out any necessary analysis and control and then displaying that information on a number of operator screens or displays.

The required control actions are then conveyed back to the process. In the early days of data acquisition, relay logic was used to control production and plant systems. With the advent of the CPU and other electronic devices, manufacturers incorporated digital electronics into relay logic equipment. The PLC or programmable logic controller is still one of the most widely used control systems in industry. As need to monitor and control more devices in the plant grew, the

PLCs were distributed and the systems became more intelligent and smaller in size. PLCs and DCS (distributed control systems) are used as shown below.



The advantages of the PLC / DCS SCADA system are:

- The computer can record and store a very large amount of data
- The data can be displayed in any way the user requires
- Thousands of sensors over a wide area can be connected to the system
- The operator can incorporate real data simulations into the system
- Many types of data can be collected from the RTUs
- The data can be viewed from anywhere, not just on site

The disadvantages are:

- The system is more complicated than the sensor to panel type
- Different operating skills are required, such as system analysts and programmer
- With thousands of sensors there is still a lot of wire to deal with
- The operator can see only as far as the PLC

1.3. Components of Modern Automation Systems

Although Modern automation systems vary in their philosophies and architectures, we can define four separate basic components that form the automation systems. These components are:

a) Field devices

b) Automatic Control System
c) SCADA software
d) Communication Networks

1.3.1. Field Devices

Field devices are the inputs and outputs to the control system; they are its eyes and hands. They can be divided into two sections, sensors and actuators, sensors do the eyes part of the job, they provide instantaneous information about the status of the field or process whether these pieces of. Typical examples for sensors are flow meters, pressure sensors and temperature sensors (these ones provide an analog signal proportional to the physical value of the measured variable and may be called “Transmitters”), other examples of sensors are limit switches, photoelectric sensors and thermostats (these ones provide discrete signals that represents predefined process states). On the other hand, actuators do the hands part of the job, they receives control orders from the control system and intend to effect the process according to these orders. Typical examples of actuators are control valves, contactors and VFDs (Variable frequency drives).

1.3.2. Automatic Control System

Typically a PLC (Programmable logic controller) or a DCS (distributed control system), these devices provides an extremely reliable structure to perform huge control functions in a relatively very short response time.

PLCs and DCSs are made to be connected directly to field devices (sensors and actuators), and thanks to their ability to store control programs and algorithms, these devices can run plants and utilities in a standalone manner with minimum dependency on humans.

1.3.3. SCADA Software

SCADA stands for supervisory control and data acquisition, and as it is clear from its name, its function is to acquire filed data and to issue supervisory control commands to a PLC or a DCS. SCADA software provides an interface between plant operators and field devices through

graphical screens that represent the actual field, these screens is continuously updated with the latest field equipment status. Key features of SCADA software are:

- User interface
- Graphics displays
- Alarms
- Trends
- RTU (and PLC) interface
- Scalability
- Security
- Access to data
- Database
- Networking
- Client/server distributed processing

SCADA software can be divided into two types, proprietary or open. Companies develop proprietary software to communicate to their hardware. These systems are sold as 'turnkey' solutions. The main problem with this system is the overwhelming reliance on the supplier of the system. Open software systems have gained popularity because of the interoperability they bring to the system. Interoperability is the ability to mix different manufacturers' equipment on the same system.

Citect and WonderWare Intouch are just two of the open software packages available in the market for SCADA systems. Some packages are now including asset management integrated within the SCADA system.

1.3.4. Communication Networks

Here comes the telemetry part to the picture, communication networks are the means by which data is transferred from RTUs (PLC or DCS) at remote site to the SCADA software stations. The communication network may be as short as 10 meters or as long as thousands of kilometers. Many communication technologies have been implemented successfully in modern automation systems such as landlines (copper wires and fiber optics), microwave, radio and

even satellite. These technologies vary in their architectures, speeds, and most important, prices.

1.4. SCADA System Boundaries

When speaking about SCADA systems, it's very important to define what is exactly meant by the word "SCADA system", because people mean a lot of things with this word. To conclude what they say, we can say it is a matter of boundaries. Each definition include different components of the automation system, for example, some people mean with "SCADA system" all automation components and equipments starting from field devices and PLCs through the communication network and ending with the SCADA software. Other people exclude the field devices, other people mean only with that word the SCADA software.

1.5. Comparison of the terms SCADA, DCS, PLC and smart instrument

1.5.1. SCADA System

A SCADA (or supervisory control and data acquisition) system means a system consisting of a number of remote terminal units (or RTUs) collecting field data connected back to a master station via a communications system. The master station displays the acquired data and also allows the operator to perform remote control tasks. The accurate and timely data (normally real-time) allows for optimization of the operation of the plant and process. A further benefit is more efficient, reliable and most importantly, safer operations. This all results in a lower cost of operation compared to earlier non-automated systems. There is a fair degree of confusion between the definition of SCADA systems and process control system. SCADA has the connotation of remote or distant operation. The inevitable question is how far 'remote' is, typically this means over a distance such that the distance between the controlling location and the controlled location is such that direct-wire control is impractical (i.e. a communication link is a critical component of the system). A successful SCADA installation depends on utilizing proven and reliable technology, with adequate and comprehensive training of all personnel in the operation of the system.

There is a history of unsuccessful SCADA systems - contributing factors to these systems includes inadequate integration of the various components of the system, unnecessary complexity in the system, unreliable hardware and unproven software. Today hardware reliability is less of a problem, but the increasing software complexity is producing new challenges. It should be noted in passing that many operators judge a SCADA system not only by the smooth performance of the RTUs, communication links and the master station (all falling under the umbrella of SCADA system) but also the field devices (both transducers and control devices). The field devices however fall outside the scope of SCADA in this manual and will not be discussed further. A diagram of a typical SCADA system is given opposite.

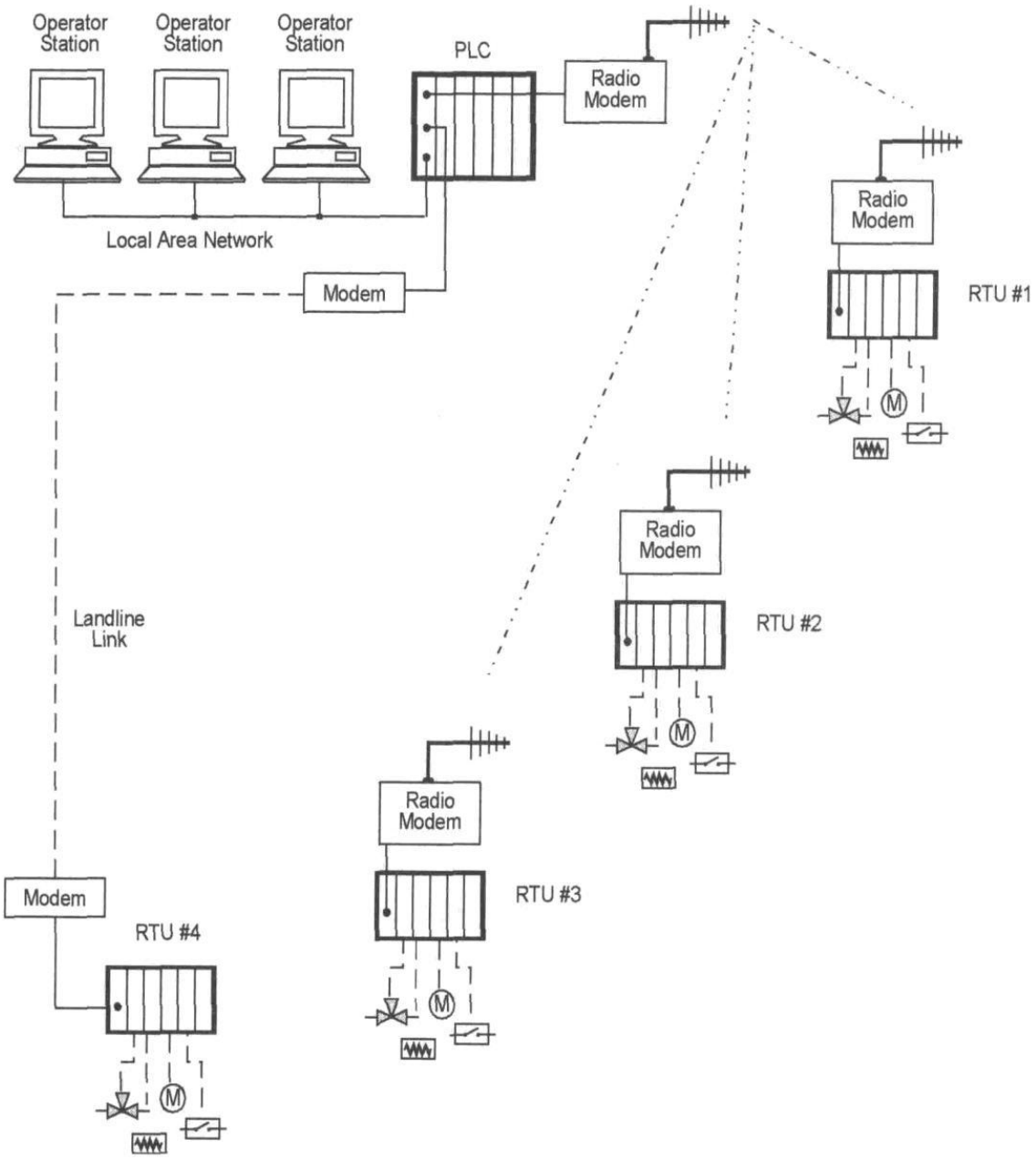


Diagram of a typical SCADA system

On a more complex SCADA system there are essentially five levels or hierarchies:

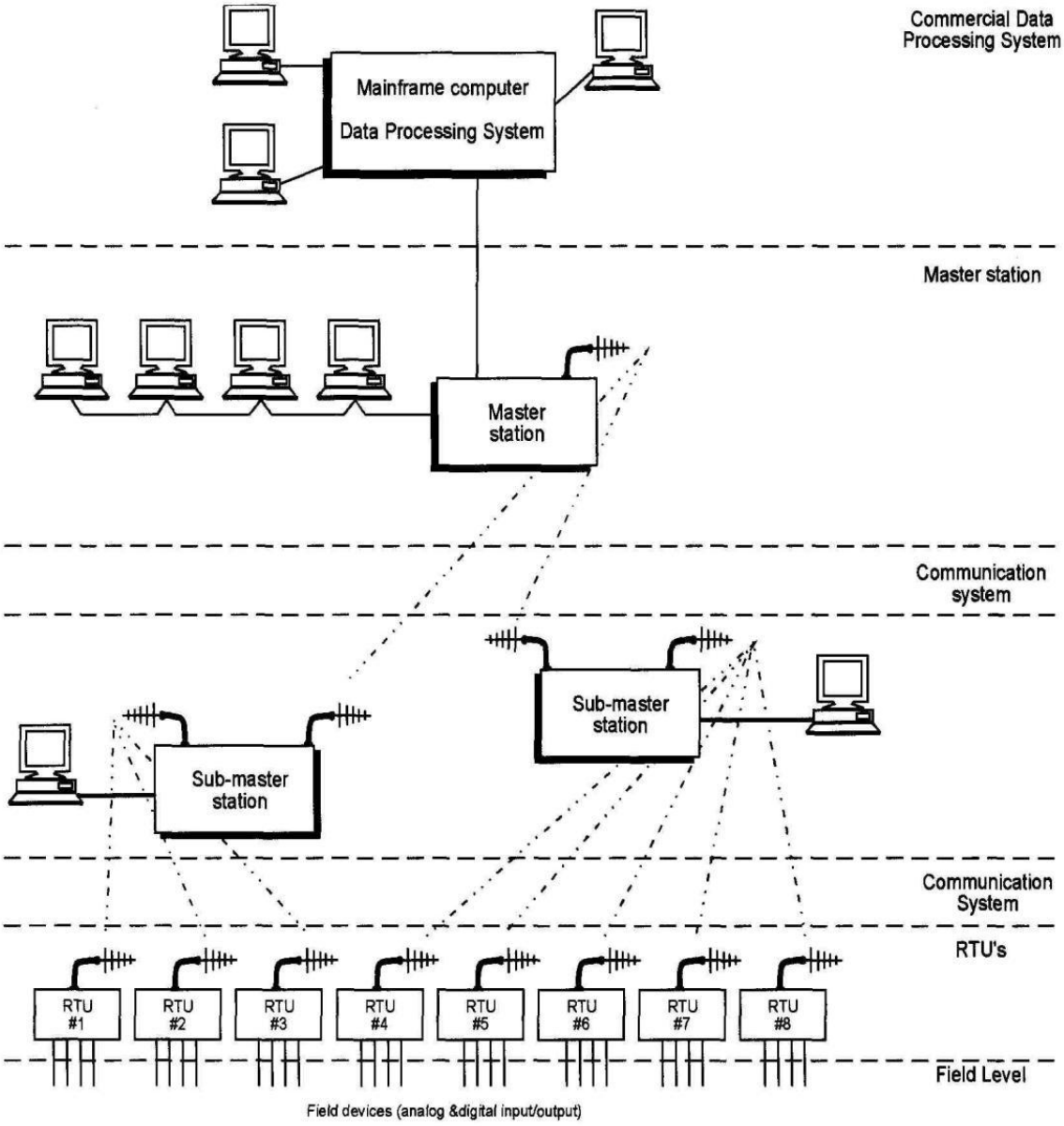
- Field level instrumentation and control devices
- Marshalling terminals and RTUs
- Communications system
- The master station(s)

- The commercial data processing department computer system

The RTU provides an interface to the field analog and digital signals situated at each remote site. The communications system provides the pathway for communications between the master station and the remote sites. This communication system can be radio, telephone line, microwave and possibly even satellite. Specific protocols and error detection philosophies are used for efficient and optimum transfer of data.

The master station (and submasters) gather data from the various RTUs and generally provide an operator interface for display of information and control of the remote sites. In large telemetry systems, submaster sites gather information from remote sites and act as a relay back to the control master station. SCADA technology has existed since the early sixties and there are now two other competing approaches possible - distributed control system (DCS) and programmable logic controller (PLC).

In addition there has been a growing trend to use smart instruments as a key component in all these systems. Of course, in the real world, the designer will mix and match the four approaches to produce an effective system matching his/her application.

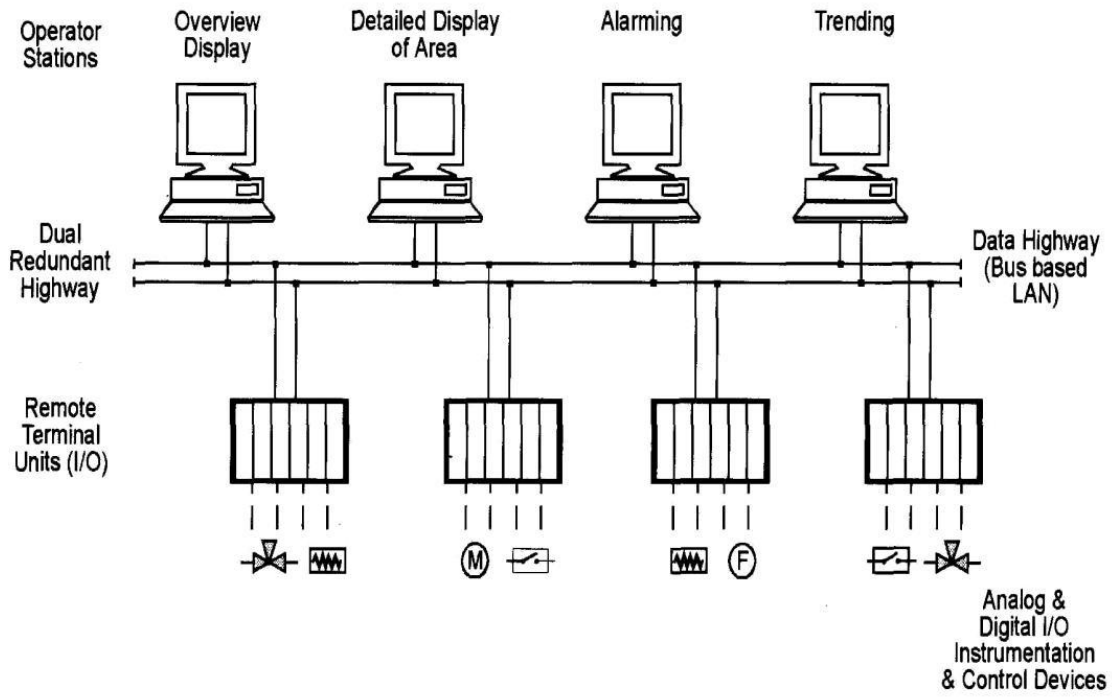


SCADA system

1.5.2. Distributed control system (DCS)

In a DCS, the data acquisition and control functions are performed by a number of distributed microprocessor-based units situated near to the devices being controlled or the instrument from which data is being gathered. DCS systems have evolved into systems providing very

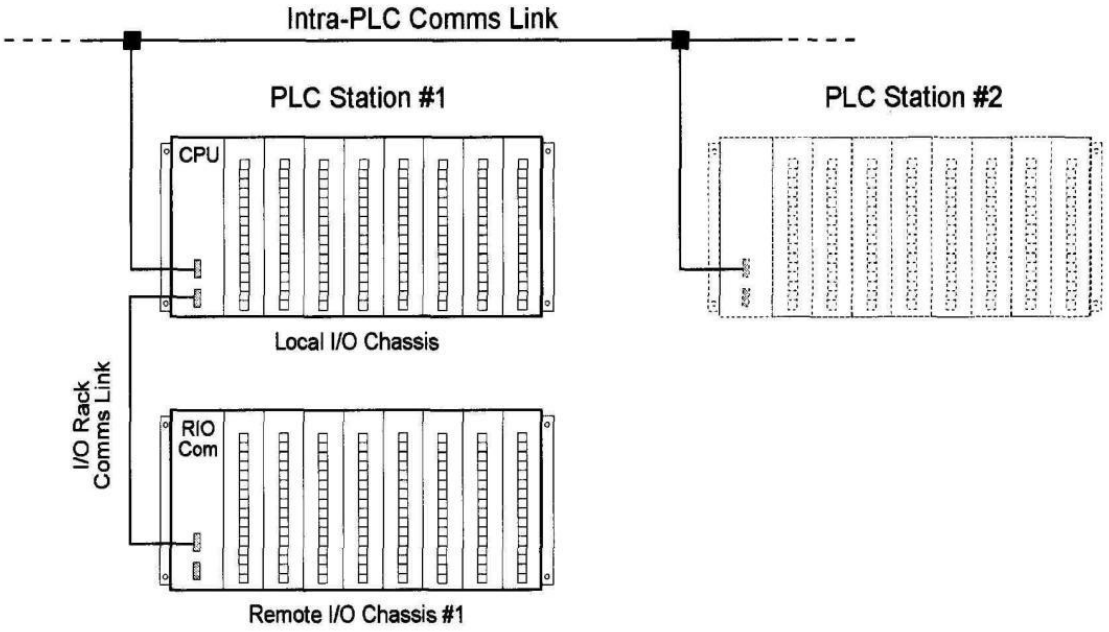
sophisticated analog (e.g. loop) control capability. A closely integrated set of operator interfaces (or man machine interfaces) is provided to allow for easy system configurations and operator control. The data highway is normally capable of fairly high speeds (typically 1 Mbps up to 10 Mbps).



Distributed Control System

1.5.3. Programmable logic controller (PLC)

Since the late 1970s, PLCs have replaced hardwired relays with a combination of ladder-logic software and solid state electronic input and output modules. They are often used in the implementation of a SCADA RTU as they offer a standard hardware solution, which is very economically priced.



Programmable Logic Controller (PLC) system

1.6. Remote terminal units

An RTU (sometimes referred to as a remote telemetry unit) as the title implies, is a standalone data acquisition and control unit, generally microprocessor based, which monitors and controls equipment at some remote location from the central station. Its primary task is to control and acquire data from process equipment at the remote location and to transfer this data back to a central station. It generally also has the facility for having its configuration and control programs dynamically downloaded from some central station. There is also a facility to be configured locally by some RTU programming unit.

Although traditionally the RTU communicates back to some central station, it is also possible to communicate on a peer-to-peer basis with other RTUs. The RTU can also act as a relay station (sometimes referred to as a store and forward station) to another RTU, which may not be accessible from the central station. Small sized RTUs generally have less than 10 to 20 analog and digital signals, medium sized RTUs have 100 digital and 30 to 40 analog inputs. RTUs, having a capacity greater than this can be classified as large.

1.7. PLCs used as RTUs

A PLC or programmable logic controller is a computer based solid state device that controls industrial equipment and processes. It was initially designed to perform the logic functions executed by relays, drum switches and mechanical timer/counters. Analog control is now a standard part of the PLC operation as well. The advantage of a PLC over the RTU offerings from various manufacturers is that it can be used in a general-purpose role and can easily be set up for a variety of different functions. The actual construction of a PLC can vary widely and does not necessarily differ much from generalizing on the discussion of the standard RTU. PLCs are popular for the following reasons:

- Economic solution: PLCs are a more economic solution than a hardwired relay solution manufactured RTU.
- Versatility and flexibility: PLCs can easily have their logic or hardware modified to cope with modified requirements for control
- Ease of design and installation: PLCs have made the design and installation of SCADA systems easier because of the emphasis on software
- More reliable: When correctly installed, PLCs are a far more reliable solution than a traditional hardwired relay solution or short run manufactured RTUs.
- Sophisticated control: PLCs allow for far more sophisticated control (mainly due to the software capability) than RTUs.
- Physically compact: PLCs take up far less space than alternative solutions.
- Easier troubleshooting and diagnostics: Software and clear cut reporting of problems allows easy and swift diagnosis of hardware/firmware/software problems on the system as well as identifying problems with the process and automation system.
- PLC software: The ladder-logic approach to programming is popular because of its perceived similarity to standard electrical circuits. Two vertical lines supplying the power are drawn at each of the sides of the diagram with the lines of logic drawn in horizontal lines.

1.8. System implementation

When first planning and designing a SCADA system, consideration should be given to integrating new SCADA systems into existing communication networks in order to avoid the substantial cost of setting up new infrastructure and communications facilities. This may be carried out through existing LANs, private telephone systems or existing radio systems used for mobile vehicle communications. Careful engineering must be carried out to ensure that overlaying of the SCADA system on to an existing communication network does not degrade or interfere with the existing facilities.

If a new system is to be implemented, consideration must be given to the quality of the system to be installed. No company has an endless budget. Weighing up economic considerations against performance and integrity requirements is vital in ensuring a satisfactorily working system at the end of the project. The availability of the communications links and the reliability of the equipment are important considerations when planning performance expectations of systems.

Chapter Two “Signal Journey in SCADA Systems”

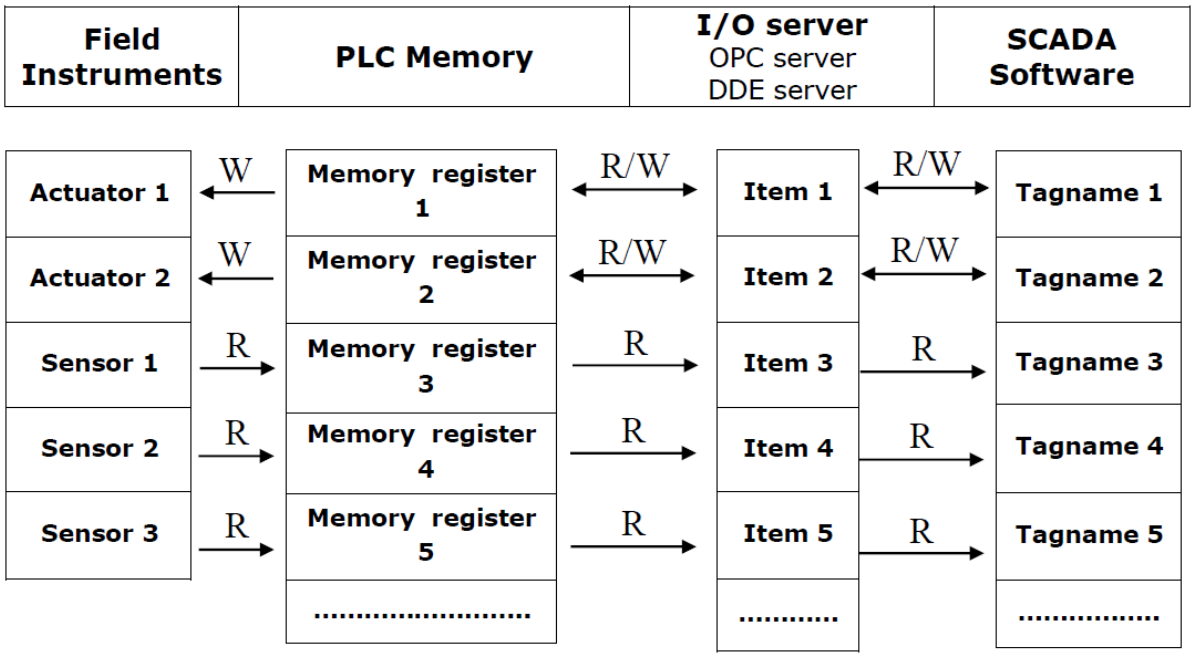
2.1. Introduction

This chapter provides the basic understanding of how the data is being moved through SCADA systems. We mean by the term “data” two things:

- The first one is the data being acquired using the sensors and transmitters from the field (Data acquisition).
- The second one is the data being instructed or ordered from the user through SCADA software (Supervisory control).

You should already know how the data being moved from measuring instruments to the PLC, and that is the first station of the signal journey. Following is a detailed description of the rest of the “signals journey” until it reaches its final station which is the SCADA software. It is clearly known that the PLC “holds” in its memory registers the data should be acquired by the SCADA software, so in this case, the SCADA software READs data from the PLC. And it’s also clearly known that the SCADA software should be able to WRITE data to other PLC memory registers.

As shown in the figure below, the data is transferred between the PLC and the SCADA software; you will notice that there is an intermediate step between the PLC and the SCADA software that is called “I/O server”, and hence, the SCADA software will work as a “client” for that “server”. The “I/O” server is software that “serves” the communication between the PLC and the SCADA software. It will be discussed in details later.



Signal journey

2.2. Typical Signal Journey

Let's take an example of a signal journey from the instruments to the SCADA software. Consider SCADA software that should display an oil tank level. Let's track the journey of this signal (value) from the beginning (i.e. transmitter) to the end (i.e. SCADA software)

The first step of the signal journey: (Field Instrument) The PLC is continuously acquiring the tank's oil level through its analog channel and holds it in a pre-specified place. This place in PLCs is called a "memory register".

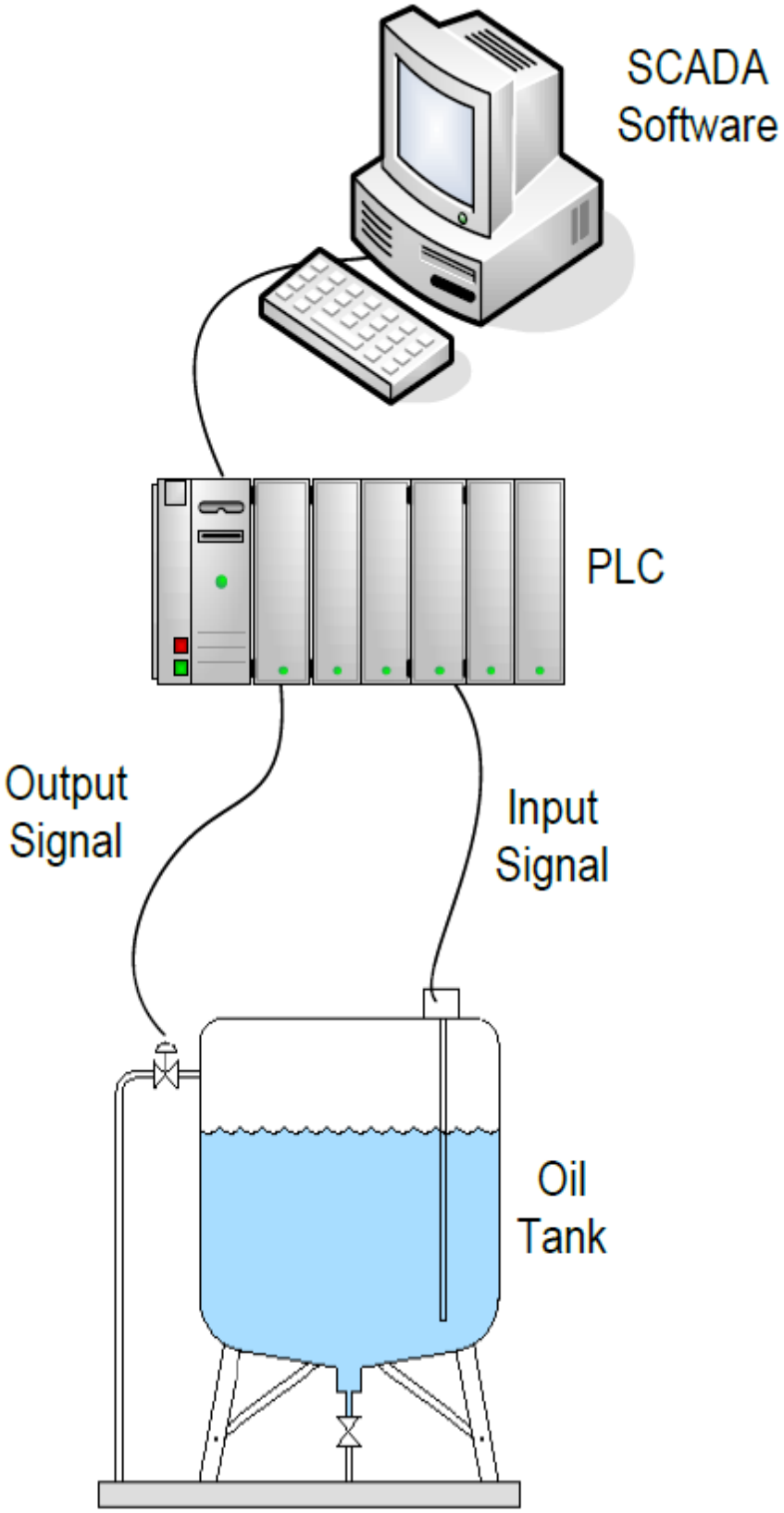
Second step of the signal journey: (I/O server) The I/O server will access this known "memory registers" and READs the value of the oil level and put it in a pre-specified place. This place in "I/O server" is called an "Item". For "items" in I/O servers, when do they are updated? Answer: the "item" is NOT continuously updated like "memory registers": in PLCs, but instead, it is only updated when the client asks for it (Remember the "client" is the SCADA software).

The third step of the signal journey: (SCADA software) The SCADA software is connected to the "I/O server" and typically, will continuously access this Known "item" to READ the value of the

oil level in the tank and put it in a pre-specified place. This place in SCADA software is called a “Tag”. This “Tag” in turn is always updated with the oil level value.

The fourth (final) step of the signal journey: (manipulating data and animation) Now, The SCADA software “knows” the value of the oil level in the tank, So It can be represented in graphics, stored in data base, trended, monitored for alarms and so on.

Reverse direction of a signal journey: The same thing is applied for parameters entered by the user to the SCADA software (The reverse direction), for example a set point or an order of starting the production, The SCADA software holds the data entered by the user in a pre-specified “Tag” and WRITEs it to a pre-specified “Item” in the I/O server and then the I/O server Writes the this value or order to a pre-specified “Memory register”, So the PLC now “Knows” the value of the required set point and will use it (i.e. output this value to a controller)


SCADA
Software
PLC
Output
Signal
Input
Signal
Oil
Tank

Oil Level Signal Journey

Let’s now take a quick introduction to both ‘Items” in I/O servers and “Tags” in SCADA software.

2.3. TAGs in SCADA software

Although we should first talk about “Items in I/O servers” as it matches the order of journey steps that are described in the last section, we will talk first about “Tags” for educational purposes only. We should know now that a “Tag” is the name used to identify the place that holds data in SCADA software. This data – in general- can be one of these three types:

- (integers or real)
- Discrete (i.e. True or False), It’s also called “Digital”.
- Text (message), it’s also called “string”

These three “things” that can be held by the “Tags” can either be an input from the user to the software (e.g. through mouse or keyboard), or an output from the software to the user (e.g. through the monitor). This is not the end; these three “things” that can be held by the “Tags” can either be an input form the PLC to the software (So the Software now READs from the PLC), or it can be an output from the software to the PLC (So the software now WRITEs to the PLC).

Note: Don’t forget that when we say the SCADA software READs/WRITEs form/to the PLC, that we mean that the SCADA software READs/WRITEs form/to the “I/O server” that communicates with the PLC; we just want to keep it simple.

Now, you may ask, Can I say that the data entered from the user to the software should be written to the PLC, so it’s only one tag used to represent this specific data? The answer: It depends! The user may enter a value or status to the software and the software in turn writes it to the PLC (e.g. open a valve, change set point) so it is the same tag, and we call it “External Tag” or “I/O Tag”. In other cases, the user may enter a value or status to the software and the software will not write it to the PLC directly, instead of that the software will use it for some data processing or calculation. In this case, we call this tag an “Internal Tag” or “Memory Tag”.

Now, you may ask another question: Can I say that the data displayed on the software is read directly from the PLC, so it’s only one tag used to represent this specific data? The answer (again): It depends! And this answer should be clear by now.

2.4. Items in I/O servers

The “I/O” server is software that “serves” the communication between the PLC and the SCADA software. It’s based on standard technologies. Before we get through the details of these technologies, we will describe the word “item” that is shown in figure. As it had been stated earlier, the word “item” is the name used to identify the place that holds data in I/O servers. This data –generally in I/O servers- can be one of these two types:

- Values (i.e. numbers)
- Status (i.e. True or False)

It’s very similar to “Tag” types in SCADA software except for the third type. And note that there is NO internal items that can be configured; all the “Items” are I/O type because that is the function of I/O servers which is to READ/WRITE from/to the PLC. Well, actually there are a number of built-in tags that most of the "I/O servers" provide, they are called "diagnostic tags", and they are used to provide feedback to client applications regarding the operation of the channel communications in the server. But the user still can't configure internal tags.

2.5. I/O Servers

We had discussed an overall description of signals journey from an instrument to a SCADA system. In this chapter we will describe the second step in deep details which is “I/O server”. I/O server is the name used in industry to point to commercial software packages that are used to handle the communication between hardware (e.g. RTU, PLC … etc) and the computer (e.g. SCADA software). These servers are based on a number of standard technologies, DDE (Dynamic Data Exchange) is the older one, and currently OPC (OLE for Process Control) is the commonly used standard technologies.

Chapter Three “Citect Configuration Environment”

3.1. Introduction

The Vijeo Citect Configuration Environment consists of five separate programs. The programs are the Citect Explorer, the Citect Project Editor, the Citect Graphics Builder and the Cicode Editor.

3.2. Citect Explorer

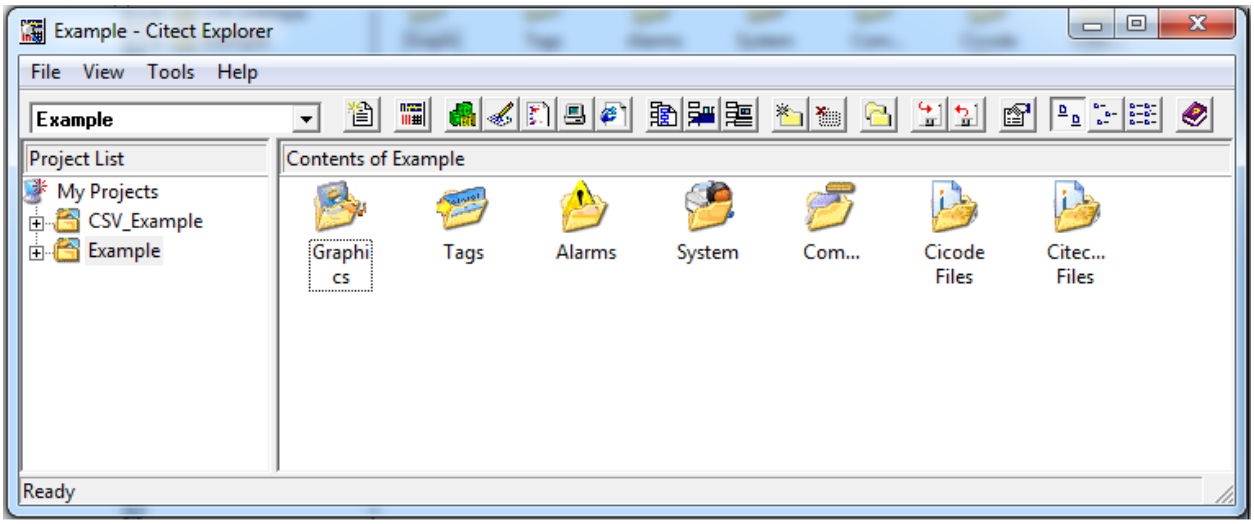
3.2.1. Overview

Citect Explorer allows you to create and manage your Vijeo Citect projects. It is also the controlling configuration application, from which you can run the Citect Project Editor, Citect Graphics Builder and Cicode Editor.

3.2.2. Starting the Citect Explorer

To start the Citect Explorer: Select the Citect Explorer item from the Windows start menu: “Start» All Programs» Schneider Electric» Vijeo Citect 7.10 » Vijeo Citect Explorer”

3.2.3. The Citect Explorer Screen



When you start Citect Explorer, the Citect Project Editor and Citect Graphics Builder automatically start and are minimized. When you close Citect Explorer, the other Vijeo Citect applications are shut down.

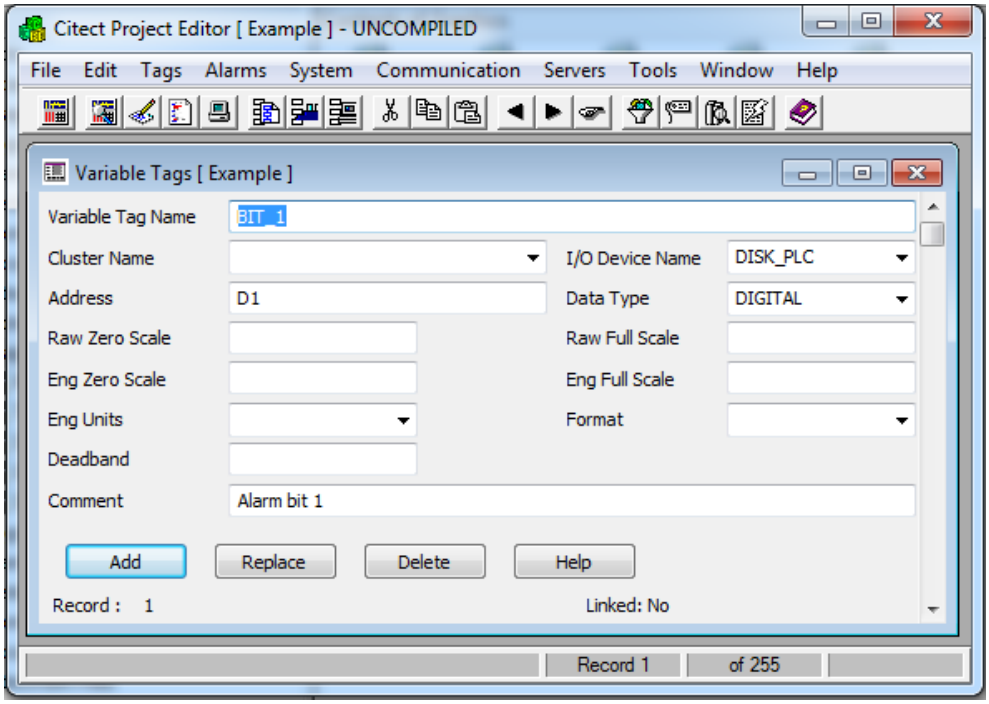
3.3. Citect Project Editor

3.3.1. Overview

The Citect Project Editor is used to create and manage the Vijeo Citect databases. These databases contain the configuration information for your Vijeo Citect project, which is not related to graphics pages. You can view all Vijeo Citect project database records in the Citect Project Editor.

3.3.2. Opening the Citect Project Editor

The Citect Project Editor is started when Citect Explorer is started.



- To open the Citect Project Editor:
  - Click the Project Editor button on the toolbar

    or
  - Choose “Tools » Project Editor from the menu”

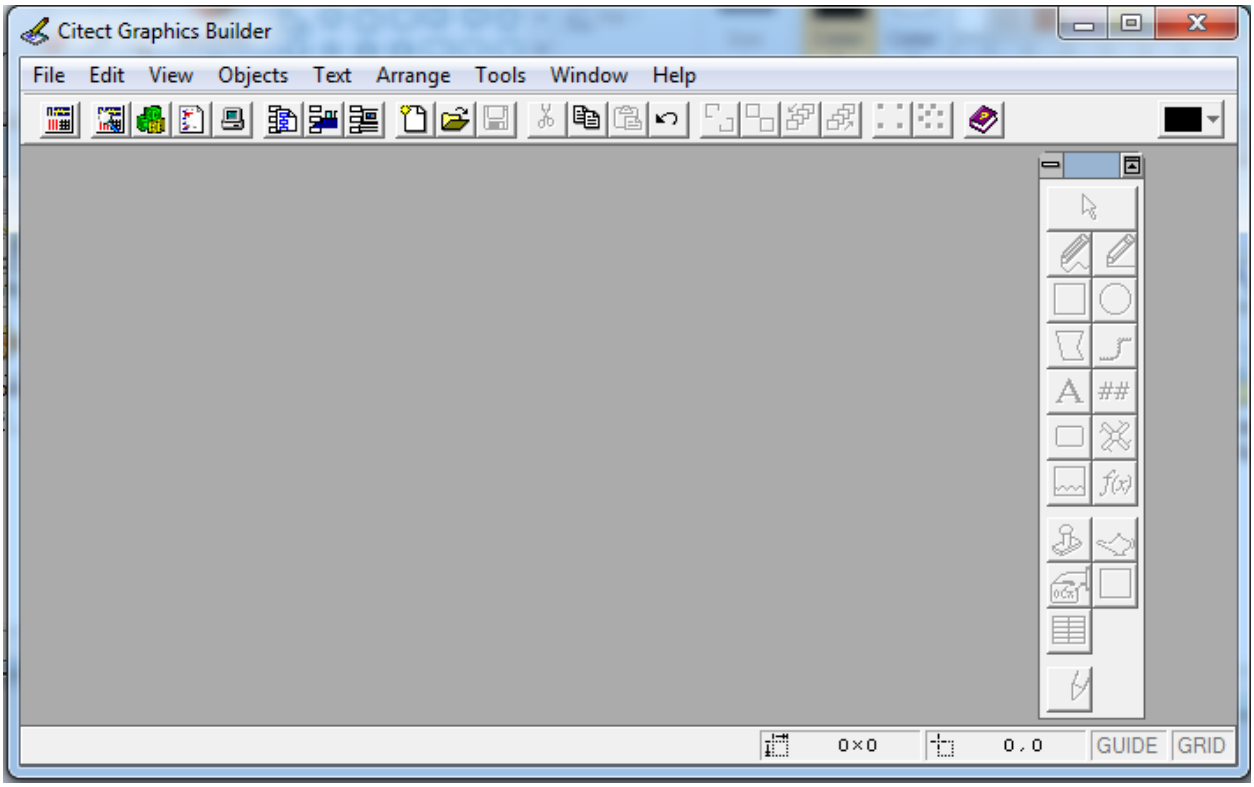
3.4. Citect Graphics Builder

3.4.1. Overview

The Citect Graphics Builder is used to create and edit graphics pages, including the objects that comprise the graphics pages.

3.4.2. Opening the Citect Graphics Builder

The Citect Graphics Builder is started when Citect Explorer is started.



- To open the Citect Graphics Builder:
  - Click the Graphics Builder button on the toolbar

    or
  - Choose “Tools » Graphics Builder from the menu”

3.5. Exploring the Environment

- **Exercise 3-1**

Step	Action
1	**Start the Citect Explorer.** a) From the Windows Start menu, select All Programs» Vijeo Citect » Vijeo Citect 7.10» Vijeo Citect Explorer.
2	**Select the example project and look at its content** a) In the Citect Explorer, click on the CSV_Example project icon in the Project List to select it as the active project.

	b) Click on the "+" symbol next to the icon to expand the project folder list and navigate through the components of the project.
3	**Click on the icons to switch to the applications**  **that are a part of Vijeo Citect**
4	**In the Citect Explorer, select the tools menu and switch to the other Vijeo Citect applications from there.**
5	**Run the CSV_Example project.** a) In the Citect Explorer, select the CSV _Example project. b) Click on the Run Project button  on the tool bar to run the project. c) Navigate through the project by clicking on buttons to change pages. d) When finished, click on the Close button  to shutdown the project when finished.

Chapter Four “Managing Projects”

4.1. Introduction

The Citect Explorer is the application from which you organise your projects and launch the configuration and run-time applications. Fundamental tasks such as creating, deleting, backing up and restoring projects are all performed in the Citect Explorer.

4.2. Creating a New Project

4.2.1. Overview

The first thing you need to do when configuring Vijeo Citect is to create a new project in which to store your project information. Each project has its own directory under the directory “C:\ProgramData\Schneider Electric\Vijeo Citect 7.10\User\”.

The project directory is created when the project is created and it is given the same name as the project.

Note: Vijeo Citect is able to use long file names for directories. The project name is restricted to 64 characters, and may contain any characters other than the semi-colon (;) or single quote (').

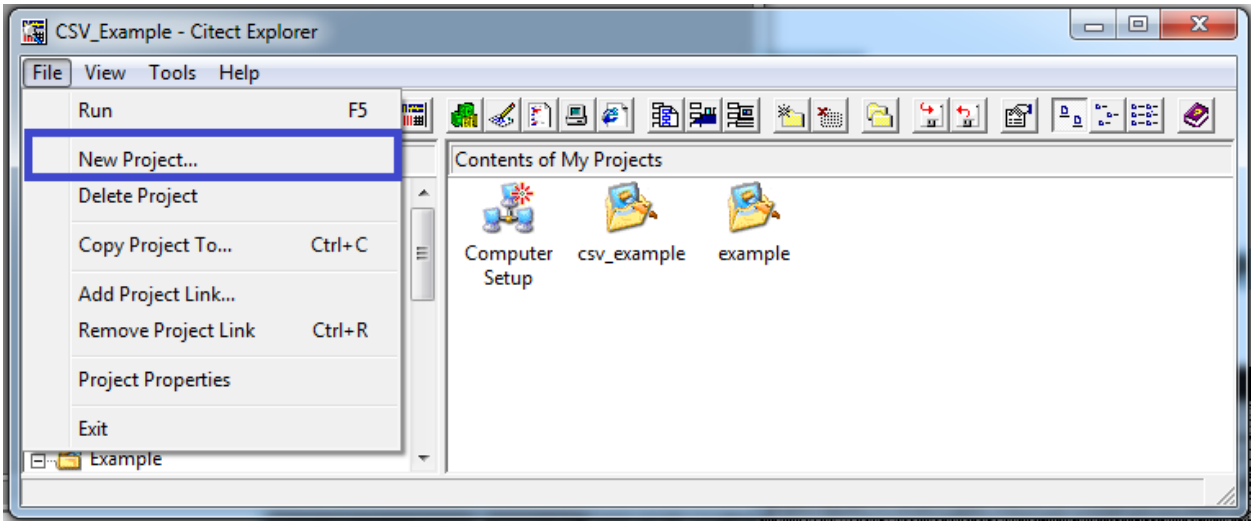
4.2.2. How to To create a new project:

- In the Citect Explorer, click on the New button 

or

- select the menu “File » New Project ...”

or

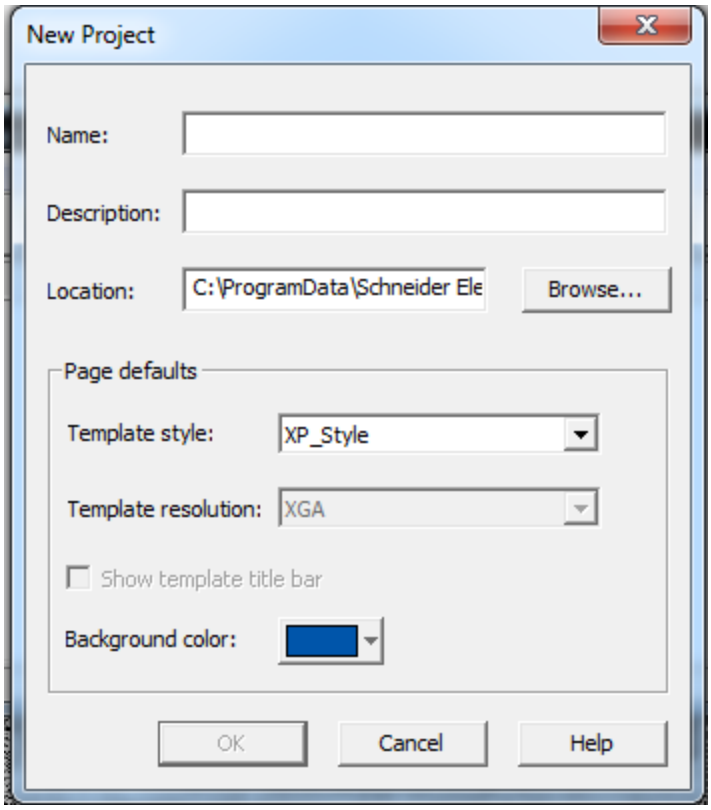


➢ Right-click on any icon in the Project List and select New Project..

4.2.3. XP Style Projects

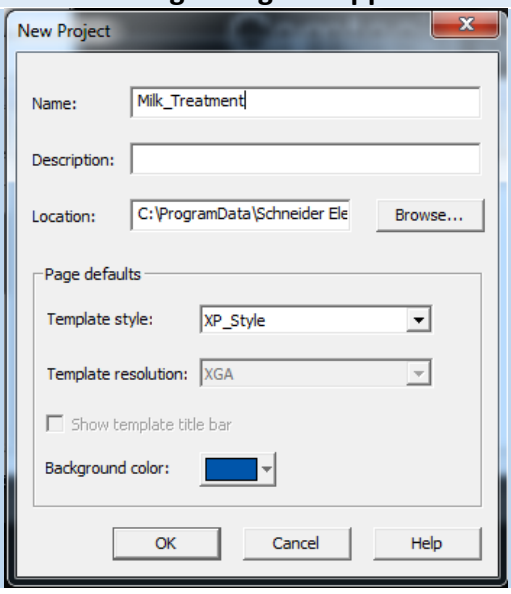
The CSV_Include Project is a pre configured project that is installed with Vijeo Citect Version 7.1 or later. The CSV _Include Project provides a set of templates that can be used to create new graphics pages with the look of the Windows XP style.

The project includes predefined trend and alarm display pages, an engineering tools page, file pages and a selection of popup windows. All feature common navigation and alarm menus for consistent functionality and appearance across an entire project, as well as providing a high degree of customization. When a new project is created XP Style is the default template.



- **Exercise 4-1**

Create a new project using the CSV _Include style

Step	Action
1	**Open the Citect Explorer, click on the New button** 
2	**The following dialog will appear**:
	

i. Complete the dialog as shown above.

Hints & Tips:

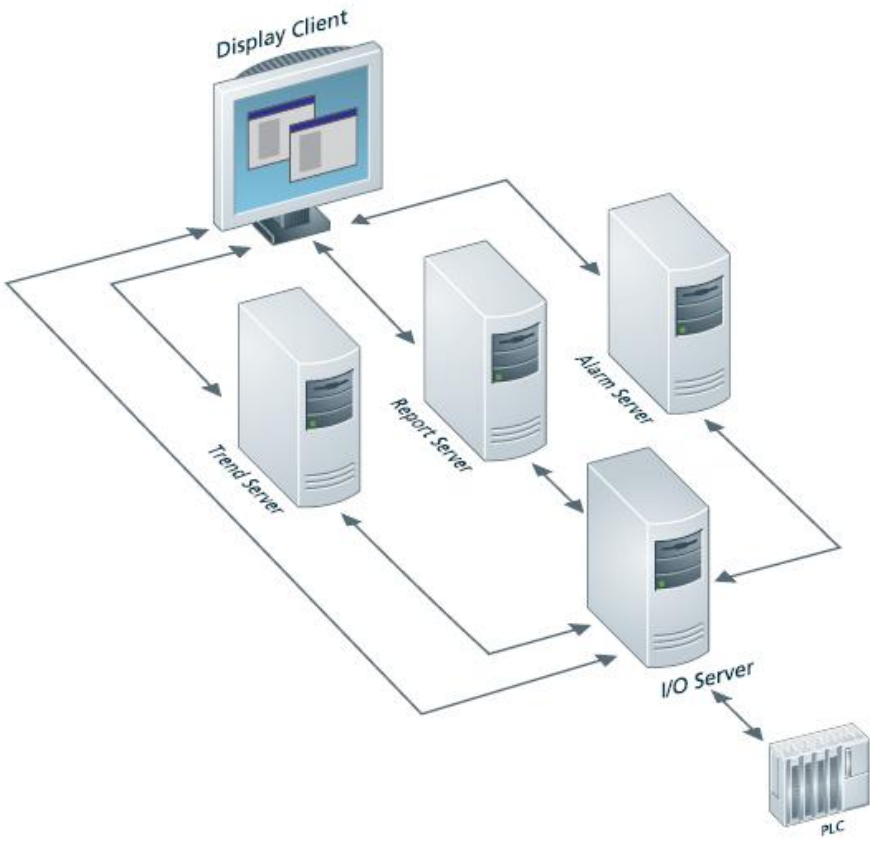
To go to the next field, use the mouse cursor or press the TAB key. To go back one field press SHIFT-TAB

Click OK. Vijeo Citect will now create the configuration databases that will be the building blocks of your project. Note the new entry called Milk_Treatment in the Project List – this entry will contain all of the configuration information for your project.

4.3. Clusters and Servers

4.3.1. Overview

Clustering allows you to group independent sets of VijeoCitect's server components within a single project, allowing multiple systems to be monitored and controlled simultaneously.



The most appropriate configuration will depend on the requirements for the solution to be deployed and the environment in which it is being deployed.

Every Vijeo Citect project needs 1 of each of these components:

- I/O Server
- Report Server
- Alarm Server
- Trend Server
- Display Client

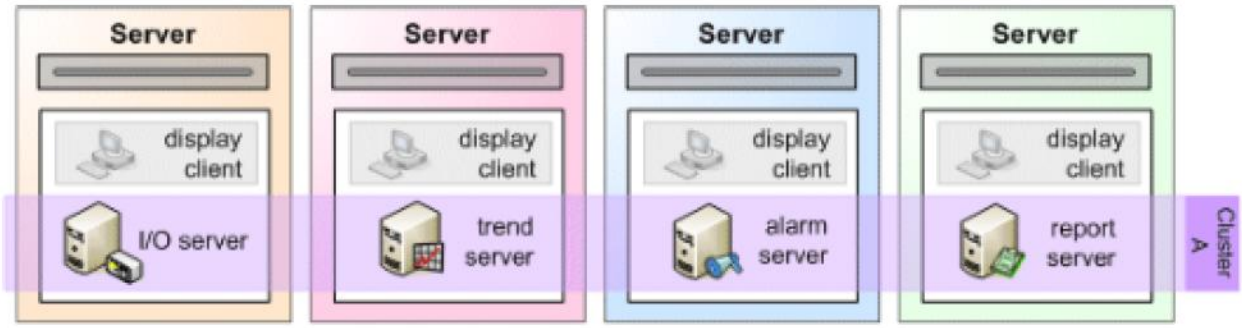
4.3.2. Standalone

These components may be spread across several computers but the simplest Vijeo Citect system has all of these components on one computer. This is called a Standalone system.

This course will be using a Stand-Alone system. Therefore one Cluster and one each of Report, Alarm and Trend Server needs to be defined within that cluster. The 1/0 Server will be defined in the next chapter.

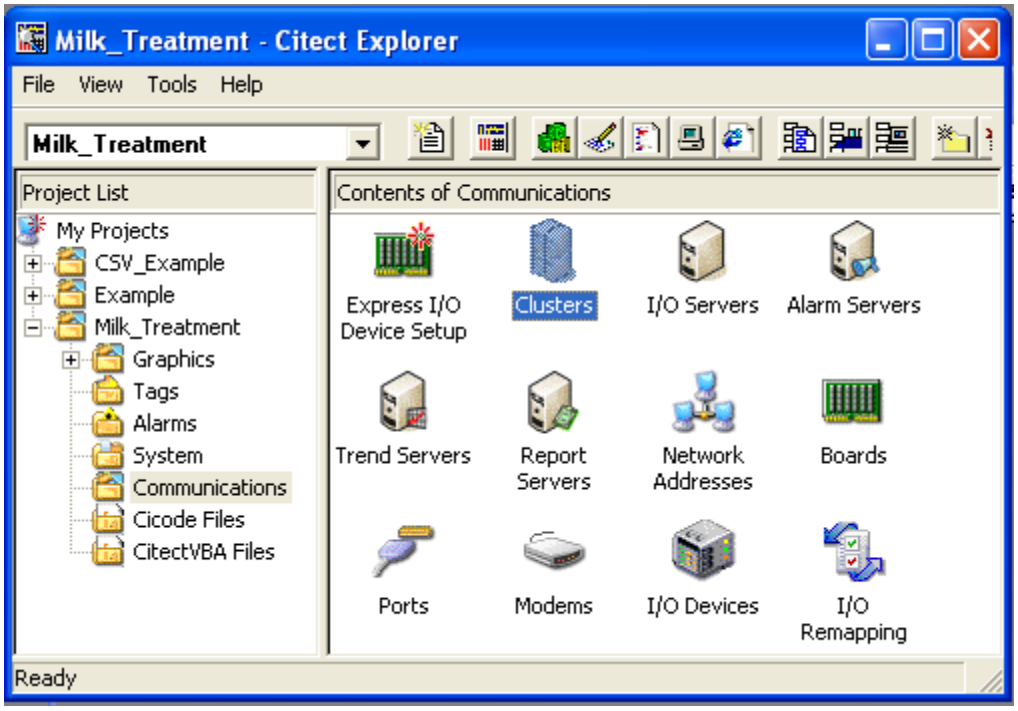
4.3.3. Clusters

Clustering allows the user to group different sets of the runtime components within a single project, allowing multiple independent systems to be monitored and controlled. The simplest system is a single cluster.



4.3.3.1. How To create a new Cluster:

In the Citect Explorer, expand the Milk_Treatment project branch. Select the Communications folder then double click the Clusters icon.

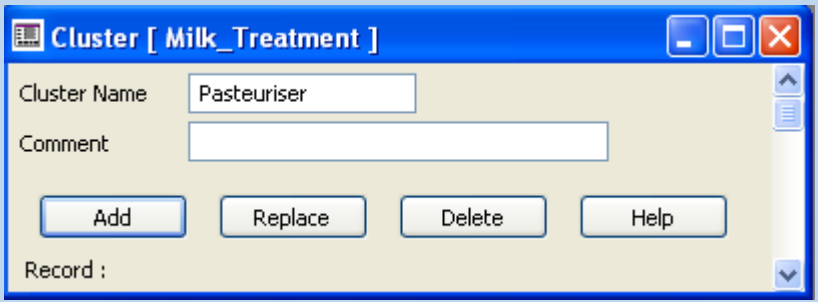


or

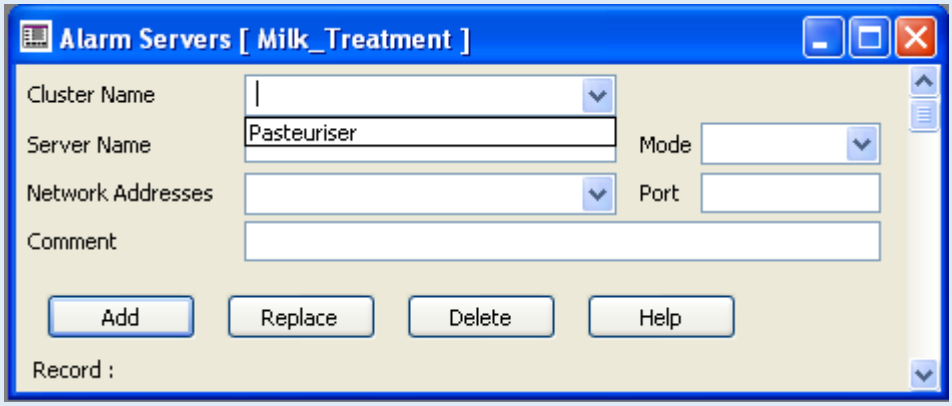
Open the Citect Project Editor and select the menu **Servers» Clusters**.

- **Exercise 4-2**

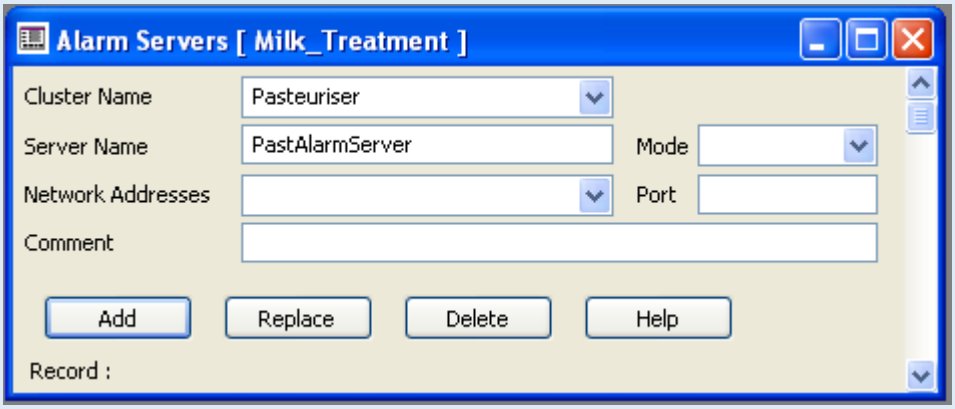
Add a Cluster and define the Report, Alann and Trend Servers.

Step	Action
1	**Define a Cluster for the Milk_Treatment project.** i. Open the Citect Project Editor. ii. Select Servers» Clusters from the menu. iii. Name the Cluster by typing **Pasteuriser** in the Cluster Name field. Click **Add**. 
2	**Define the Alarm Server.** i. Select Servers» Alarm Servers from the menu.

ii. In the Cluster Name field select Pasteuriser from the drop down list.



iii. In the Server Name field type **PastAlarmServer**. Click **Add.**

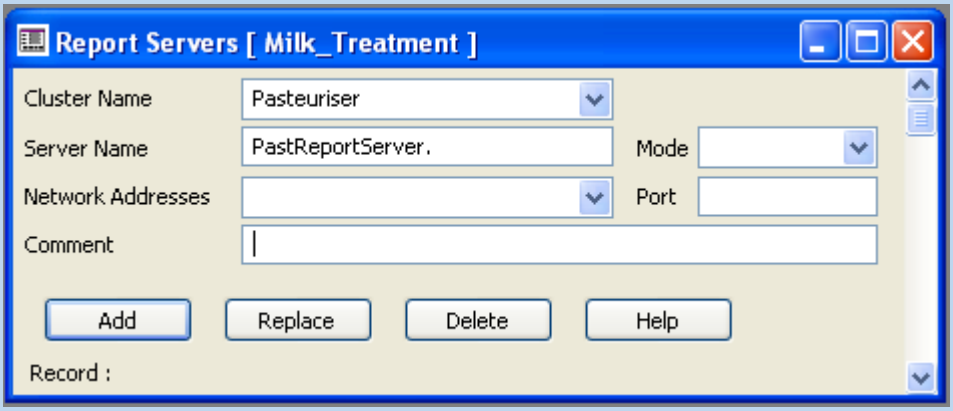


3 **Define the Report Server.**

i. Select **Servers» Report Servers** from the menu.

ii. In the Cluster Name field select **Pasteuriser** from the drop down list.

iii. In the Server Name field type **PastReportServer**. Click Add.



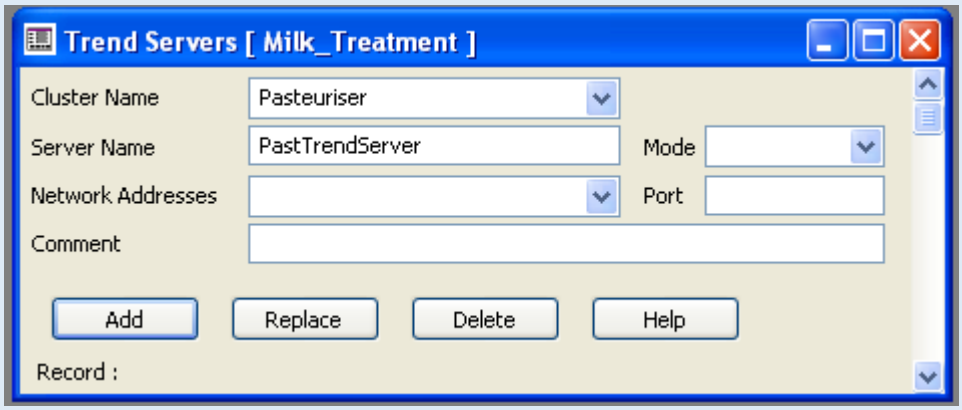
4 **Define the Trend Server.**

i. Select **Servers» Trend Servers** from the menu.

ii. In the Cluster Name field select **Pasteuriser** from the drop down list.

iii. In the Server Name field type **PastTrendServer**. Click Add.

iv. The dialogs should look like this.



4.4. Backing Up, Restoring and Deleting a Project

4.4.1. Overview

Vijeo Citect projects can be backed up to compressed files that take up much less space than the original project directory. Regular backups should be made whilst a project is being developed, in case files are accidentally deleted or become corrupted. A backup may be saved to a floppy drive, your local drive or to a network share. It is also very important that a history of backup files is kept, so that it is always possible to revert to a previous version of the project - especially when changing an already working system.

4.4.2. How to Backup your Project

➢ To backup a project:

In the Citect Explorer, click on the Backup button 

or

select the menu **Tools » Backup ...**

4.4.3. How to Restoring a Project

In the Citect Explorer, click on the Restore button 

or

select the menu **Tools » Bestore ...**

Note: By default, backup files are given the extension .CTZ if no extension is specified. These files are in a standard zip format. This means that any zip extraction tool may be used to open the file.

4.4.4. How to Deleting a Project

When a project is deleted in the Citect Explorer, all of the associated files and the project directory are permanently removed.

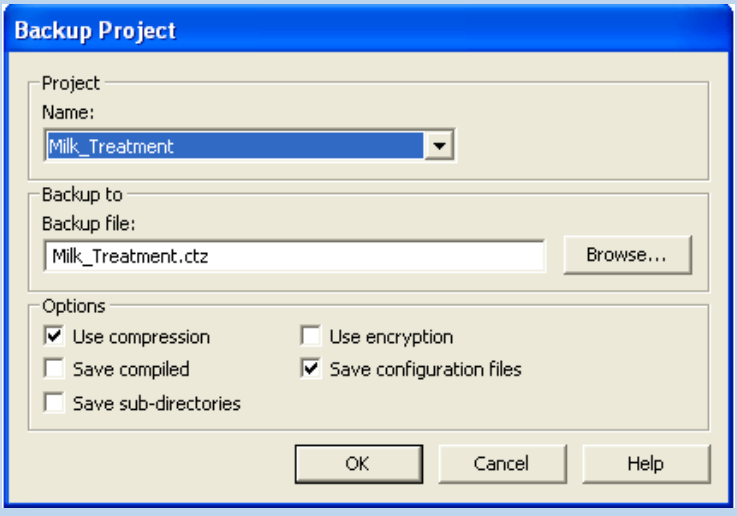
➢ To delete a project:

In the Citect Explorer, click on the project in the Project List that you want to delete, then either select the menu **file» Delete Project**

or

right-click on the project icon and select Delete Project from the context menu.

- **Exercise 4-3**

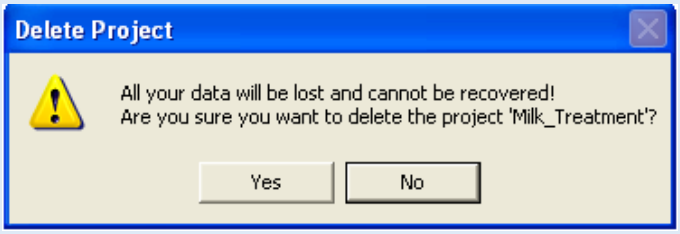
Step	Action
1	**Backup the Milk_Treatment Project** I. Open the Citect Explorer and select the **Milk_Treatment** project icon in the Project List. II. Click on the **Backup** button  . The following dialog will appear. 

III. Add a path for the file location by either typing it in front of the filename or by using the Browse button to select a location. Click OK to continue.

2 **Delete the Milk_Treatment project.**

I. In the Citect Explorer, click on the Milk_Treatment project icon in the Project List to select it.

II. Select the menu **file» Delete Project**. The following warning dialog will appear:

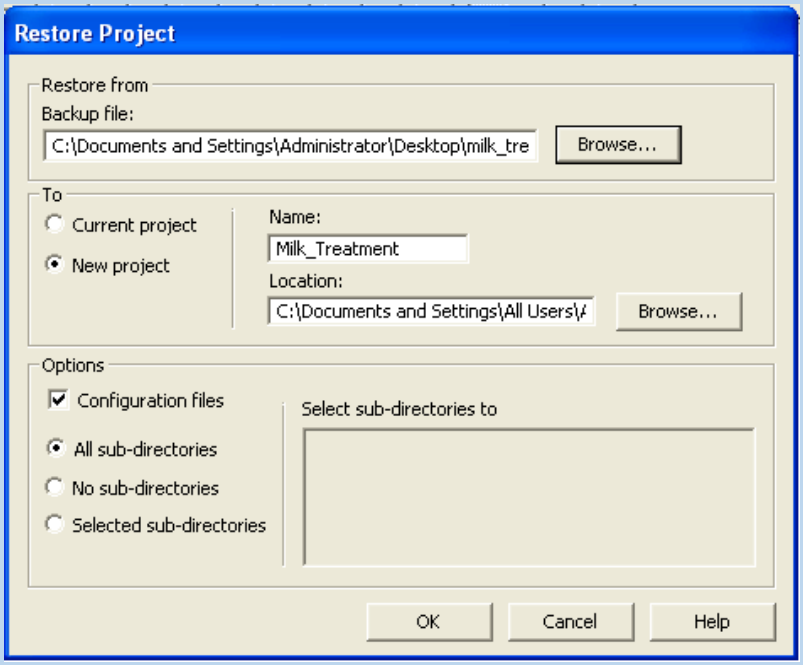


Click **Yes** to confirm deleting all project files.

3 **Restore the Milk_Treatment project.**

I. In the Citect Explorer, click on the Restore button 

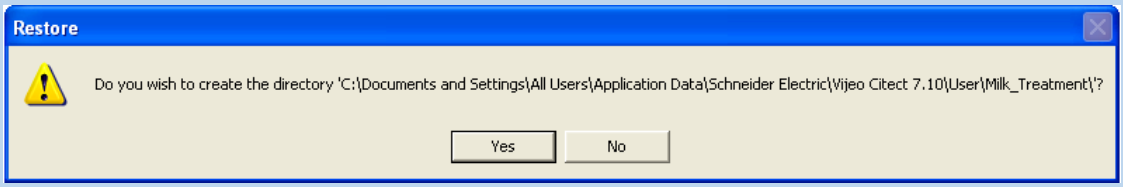
II. The following dialog will appear, use **Browse ...** to select the Backup file if it is not already selected and then select **New project**:



Caution! If you restore to the Current project, then the currently selected project in the Citect Explorer will be overwritten. You can restore any project to a New project and give it a new name or the same name as before as long as the name is unique.

III. Click OK to restore the Milk_Treatment project.

IV. This message prompt will appear. Click yes.

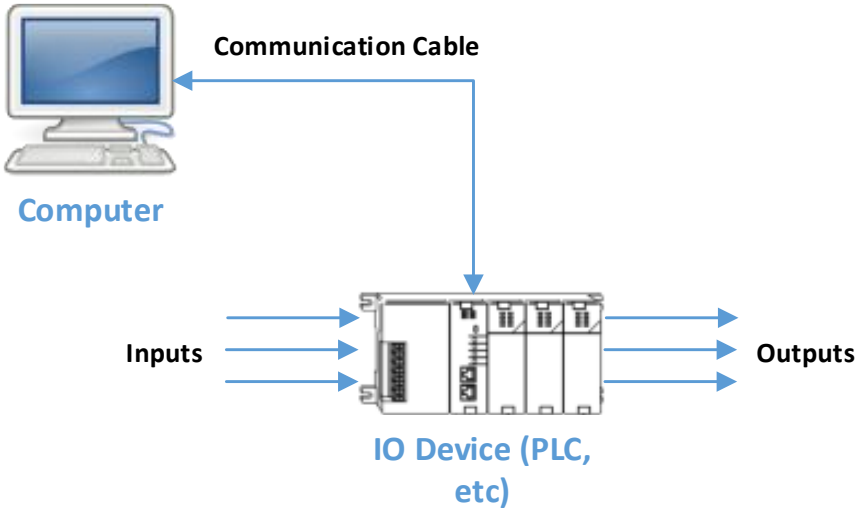


V. Click OK when the Restore Complete message prompt is displayed.

Chapter Five “Setting Up Communications”

5.1. Introduction

Vijeo Citect can communicate with many types of control or monitoring I/O Devices - including PLCs (Programmable Logic Controllers), loop controllers, bar code readers, scientific analysers, remote terminal units (RTUs), and distributed control systems (DCS). Vijeo Citect communicates directly with the I/O Devices in your plant or factory, allowing data transfer to or from the device and supervisory control of the system.



5.2. Overview

The Express Communications Wizard enables you to quickly set up communication with a new or existing VO Device.

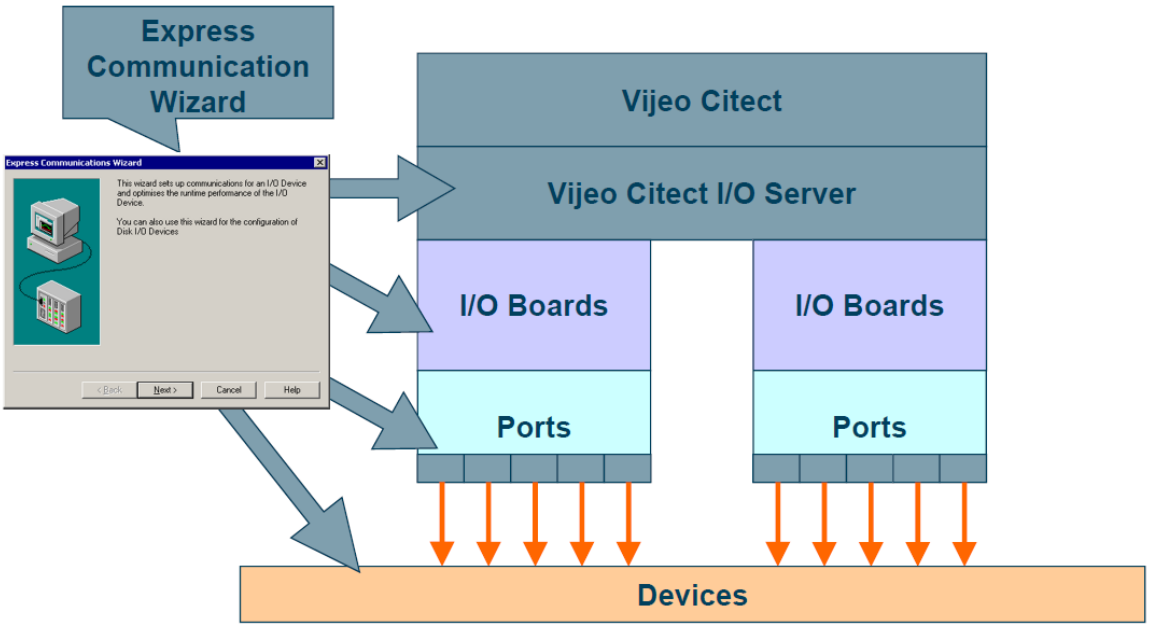
Each Vijeo Citect project will need to have at least one VO Server defined for it to communicate to I/O Devices. Each VO Device will need to be set up in Vijeo Citect with an appropriate communications protocol and settings and any interface boards and communications ports in the computer will also have to be defined.

Stepping through the Express Communications Wizard defines the:

- I/O Server names
- interface Boards in the computer
- communications Ports on the boards

- I/O Devices attached to the ports

Note: Each Vijeo Citect computer can only operate as one Vijeo Citect I/O Server. If a common project is run across a network of Vijeo Citect computers, with more than one computer connected to an I/O Device, then more than one VO Server may be defined in the project.



5.2.1. How to run the Express Communications Wizard:

In the Citect Explorer, select the project then double-click on the Communications folder and double-click on Express I/O Device Setup.

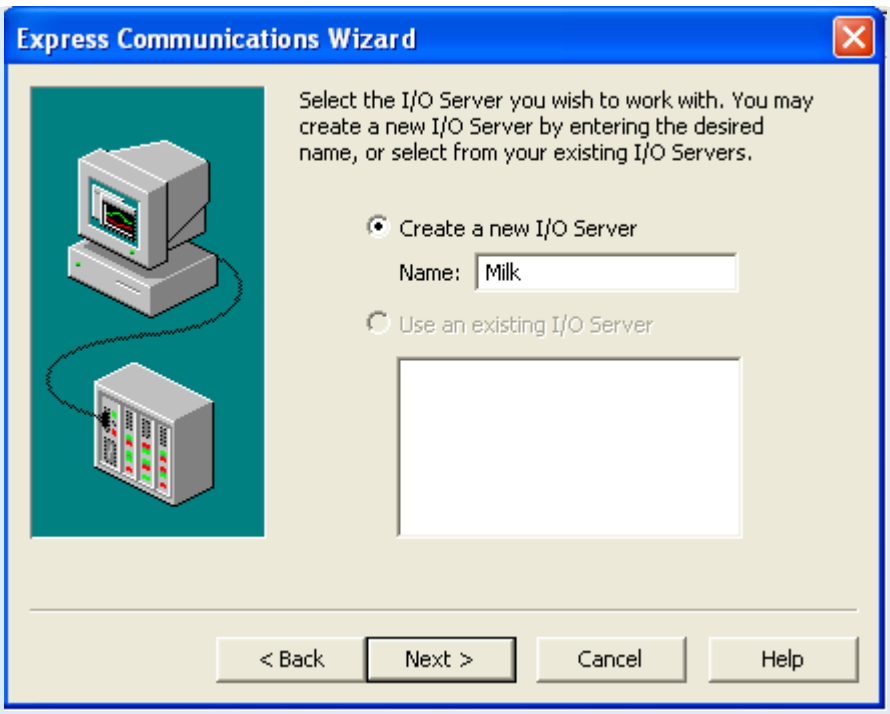
or

Go to the Citect Project Editor and select the menu **Communication» Express Wizard**

- **Exercise 5-1**

Step	Action
1	**Open the Citect Explorer and select the Milk_Treatment project.**
2	**Go to the Citect Project Editor by clicking the Citect Project**

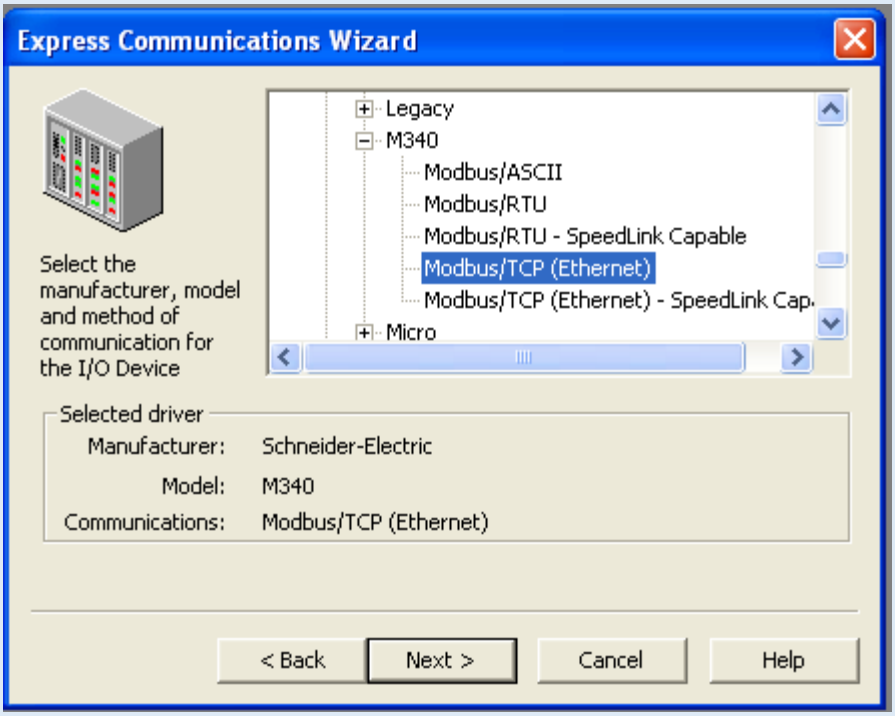
	Editor button on the toolbar.
3	**Open the Express Communications Wizard.** Select the menu **Communication I Express Wizard**. The following dialog will appear. Click **Next>** to continue. 
4	**Use the Express Communications Wizard to set up your I/O Device.** I. This dialog allows you to create and name your new 1/0 Server. Give the new I/O Server the name Milk. Click Next > to continue.



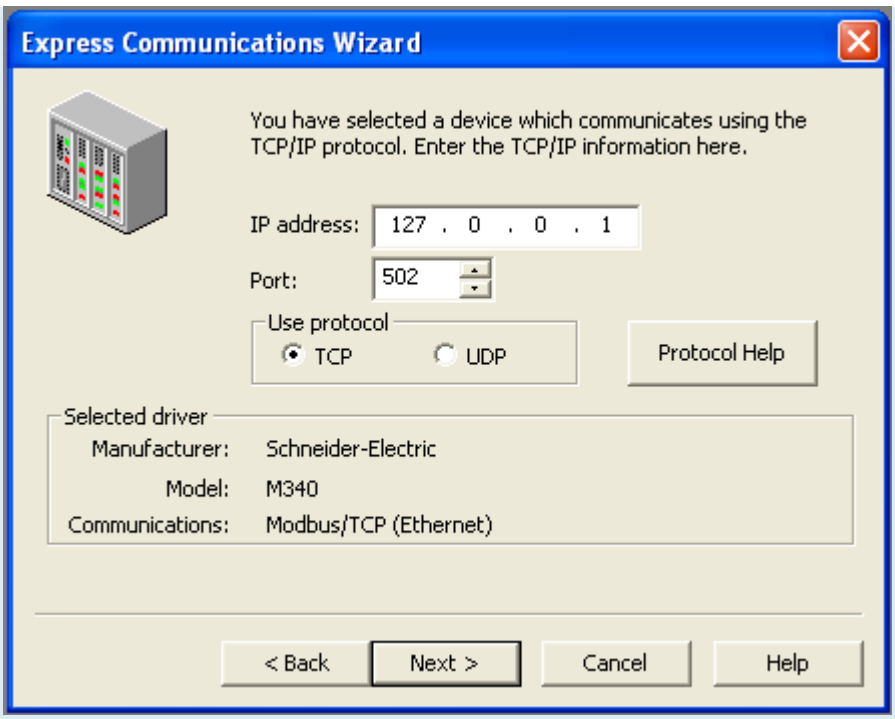
II. This dialog allows you to create a new I/O Device or edit an existing I/O Device. Select **Create a new I/O Device** and name the device MilkDev1 then click **Next >** to continue.

III. This dialog allows you to select the I/O Device type. Select External I/O Device.

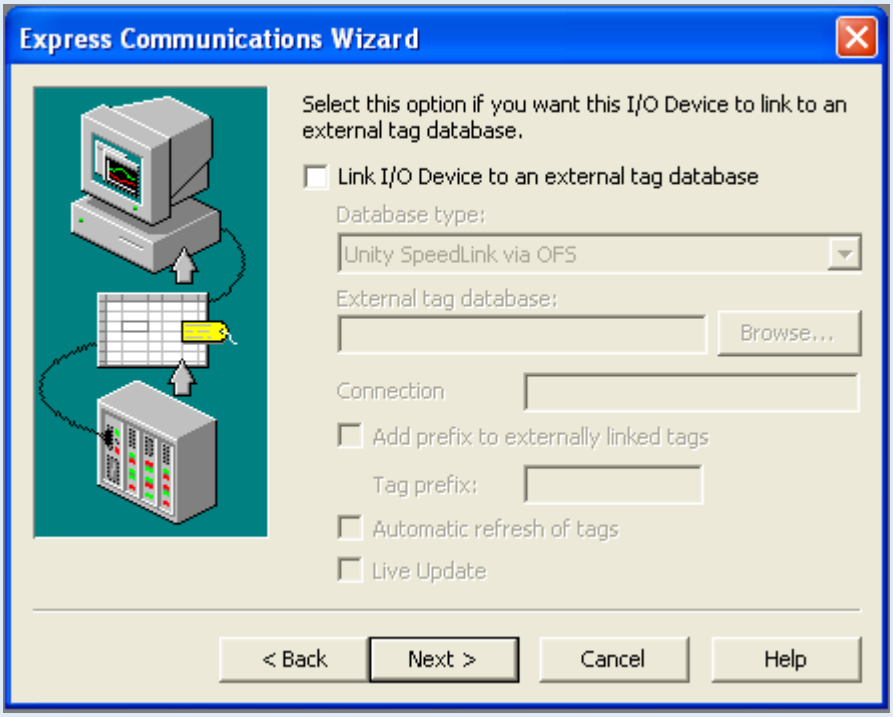
IV. Select the Manufacturer, Model and Communications Method from the list. Select Modbus/TCP (Ethernet) protocol. Click **Next>** to continue.



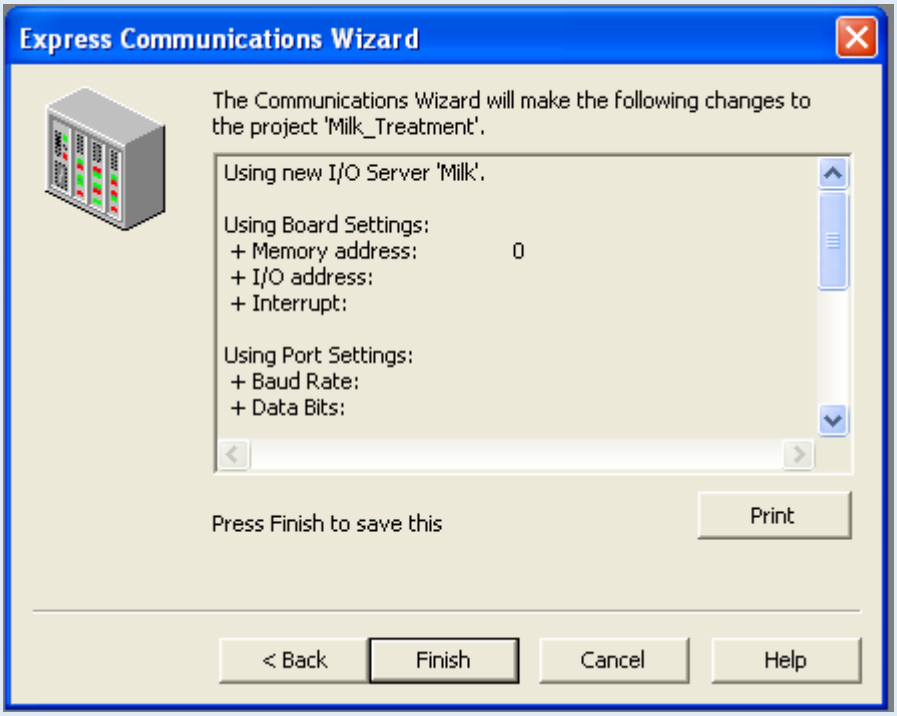
V. If you are connected to an Ethernet I/O Device you will see a dialog similar to this. You will need to enter the correct IP address for this device.



VI. The next dialog box allows you to link to an external database containing variable tag information. The link can be to a static file or it may take advantage of Microsoft's COM and DCOM technologies. Click **Next >** to continue.



VII. The final dialog displays a summary of the communications setup. If you have chosen to configure a real I/O Device then the summary will include your communications board and port settings.



5.3. Communications Dialogs

5.3.1. I/O Server, Boards, Ports and I/O Devices

After running the Express Communications Wizard, a set of communications dialogs are created as part of your project.

These can be opened from the Citect Explorer, in the Communications folder or in the Citect Project Editor, by selecting the menu **Communication**.

- **Exercise 5-2**

Open and review the communications forms in the Milk_Treatment project.

Step	Action

1	**Open the Citect Project Editor and choose Communication >> I/O Server from the menu.**
2	**Choose each of the items Boards, forts and I/O Devices from the Communication menu to display the associated forms.**

5.4. Variable Tags

5.4.1. Overview

Variable Tags define the data that is transferred between the I/O Device and the Vijeo Citect I/O Server. Each Variable Tag is defined with a unique name, a data type, an address and an associated I/O Device.

5.4.2. How to define a variable tag:

Open the Citect Explorer, select a project and open the Tags folder then double click on Variable Tags.

or

open the Citect Project Editor, select the menu rags » Variable Tags

or

open the Citect Project Editor, click on the Variable Tags icon  on the toolbar.

5.5. Forms

5.5.1. Overview

All forms in Vijeo Citect have the same standard interface. The buttons are:

Button	Function
Add	Add the currently displayed information to the database as a new record.

Replace	Replace the current record with the information currently displayed.
Delete	Delete the current record.
Help	Open the Citect Help Topics for the parameters on the current form.

The scroll bar can be used to scroll from one record to the next. Records are in the database in order of entry.

Hints & Tips:

To search for a particular record, select the menu gdit » find ... when the form is open and search the current form. The form will then filter only the matching records - use the scroll bar if more than one record match is found.

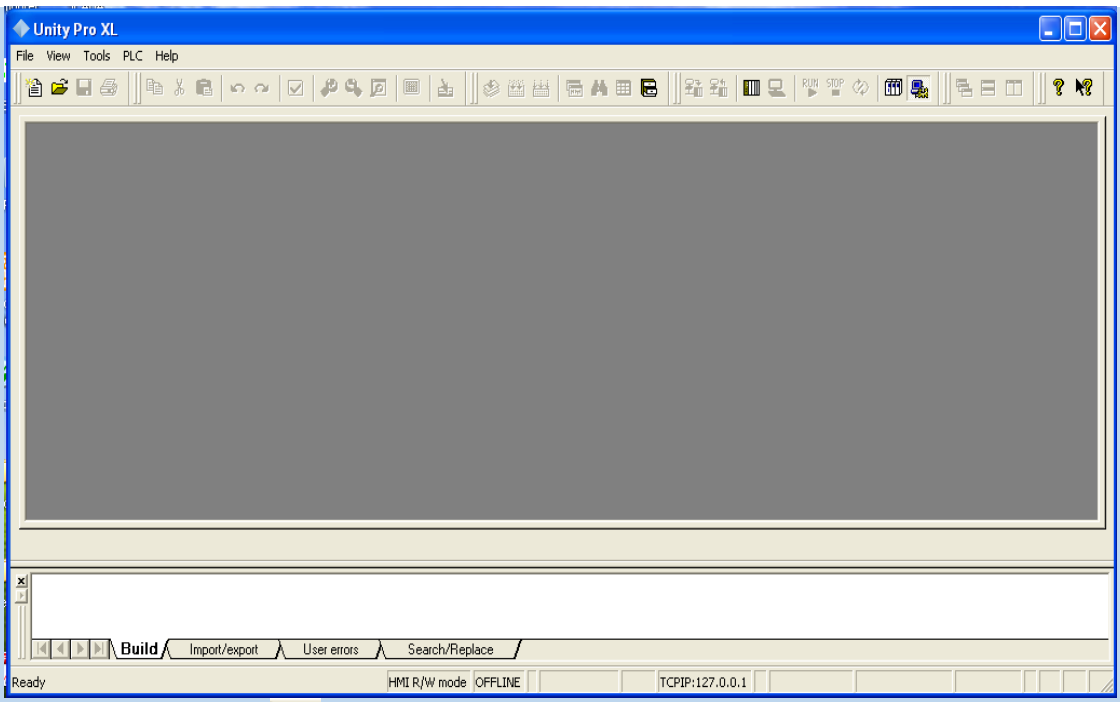
5.6. Testing Communications

5.6.1. Overview

It is very important to test communications early in the development of a new project. Without reliable communications, your Vijeo Citect project will not operate effectively. Testing communications also ensures that the correct addressing format is being used for the Variable Tags defined in the project.

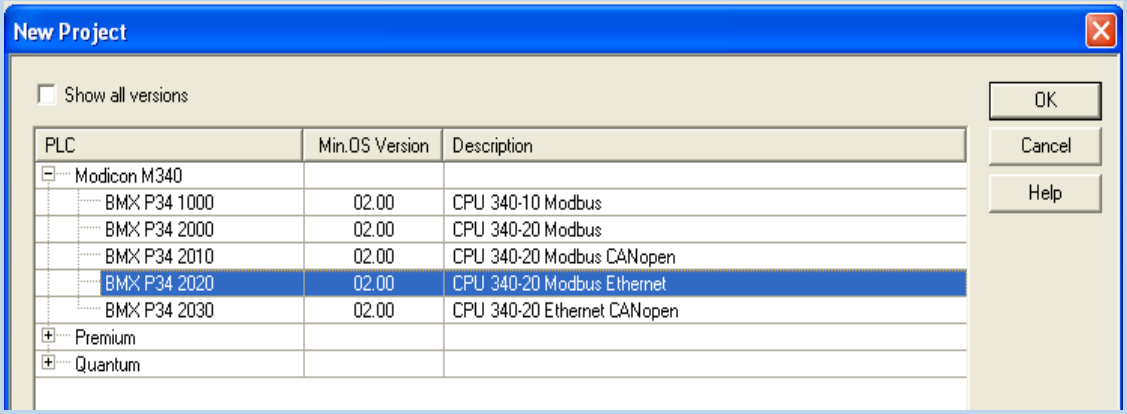
- **Exercise 5-3**

Step	Action
1	**Define a variable tag called Test in Unity Pro.** i. Open unity pro program.

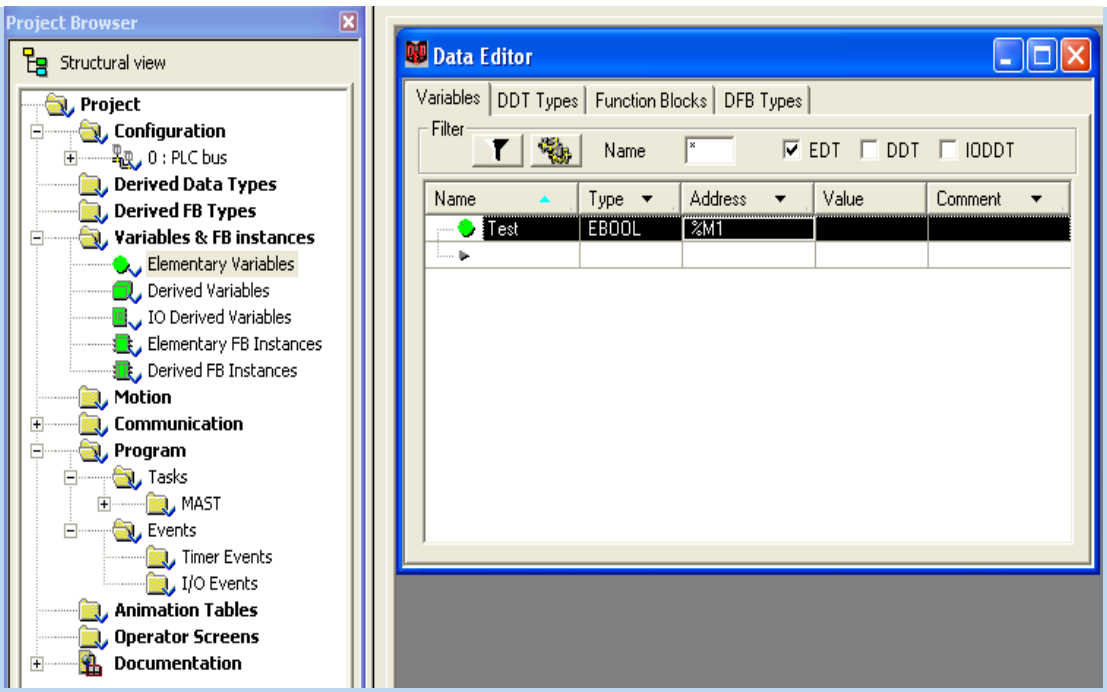


ii. Click on icon  to create new project

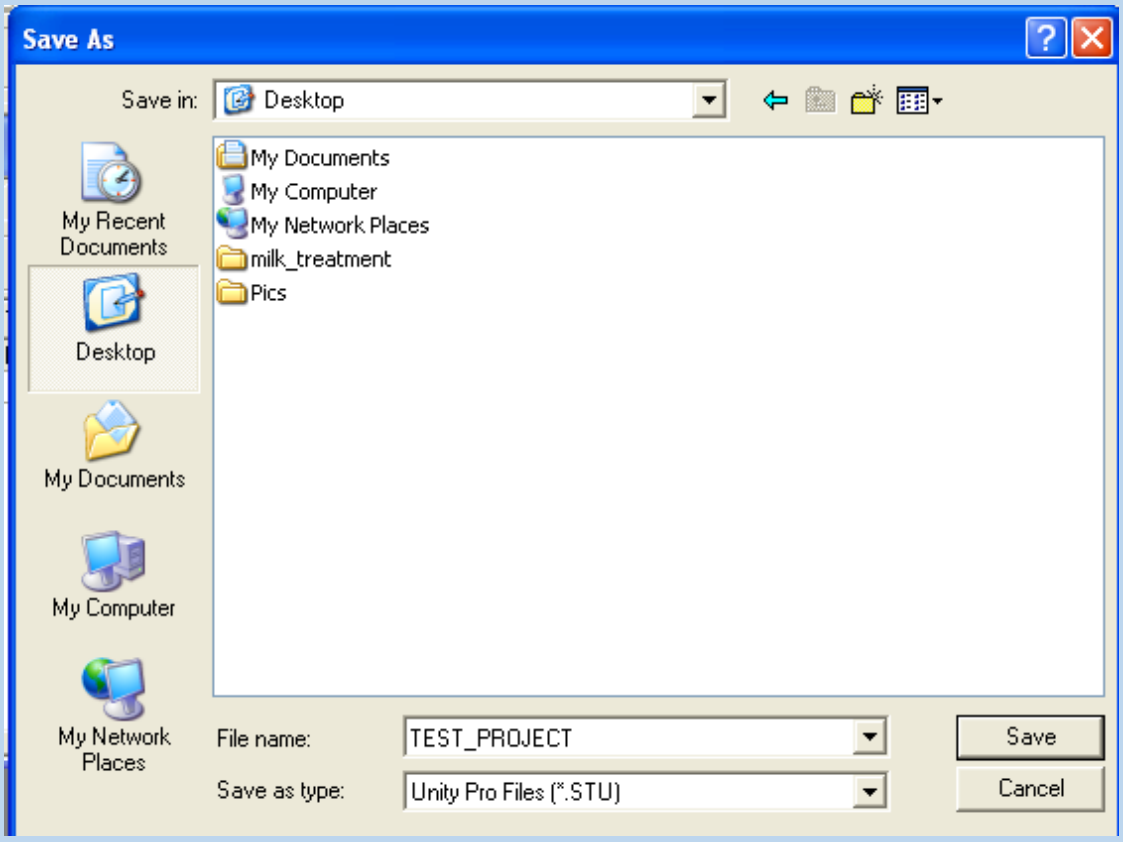
iii. Choose **Modicon M340 >> BMX P34 2020 >> CPU 340–20 MODBUS Ethernet** PLC type



iv. In the project browser go to **Project >> Variable and FB instances >> Elementary Variables.** Add variable **“Test”** with address **“%M1”.**



v. Save the Project with file name **"Test_Project".**

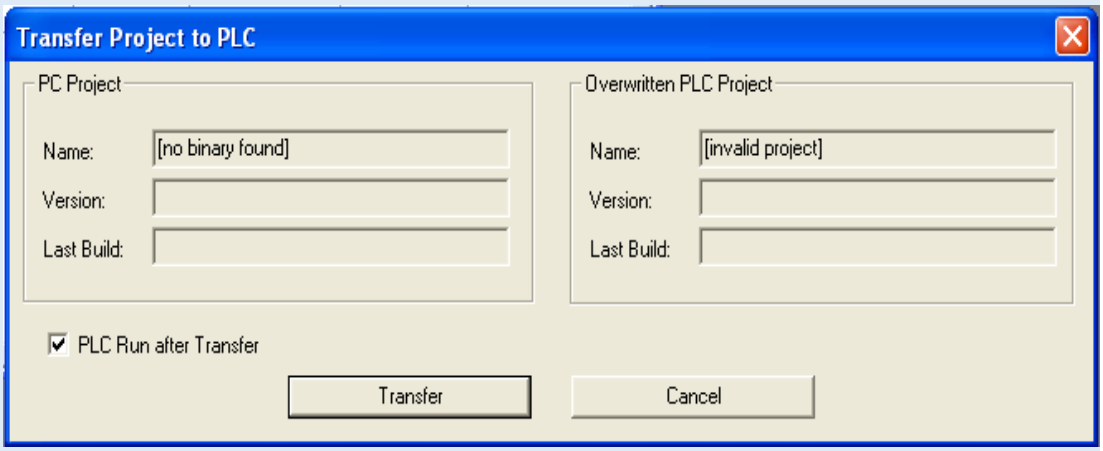


3	**Download Project to Virtual PLC**

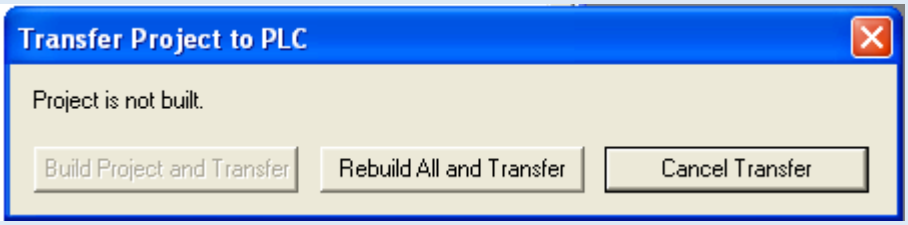
i. Click on icon  to connect to Virtual PLC.

ii. Click on icon  to download project to virtual PLC.

iii. Click **"Transfer"** after choosing **"PLC Run after Transfer".**

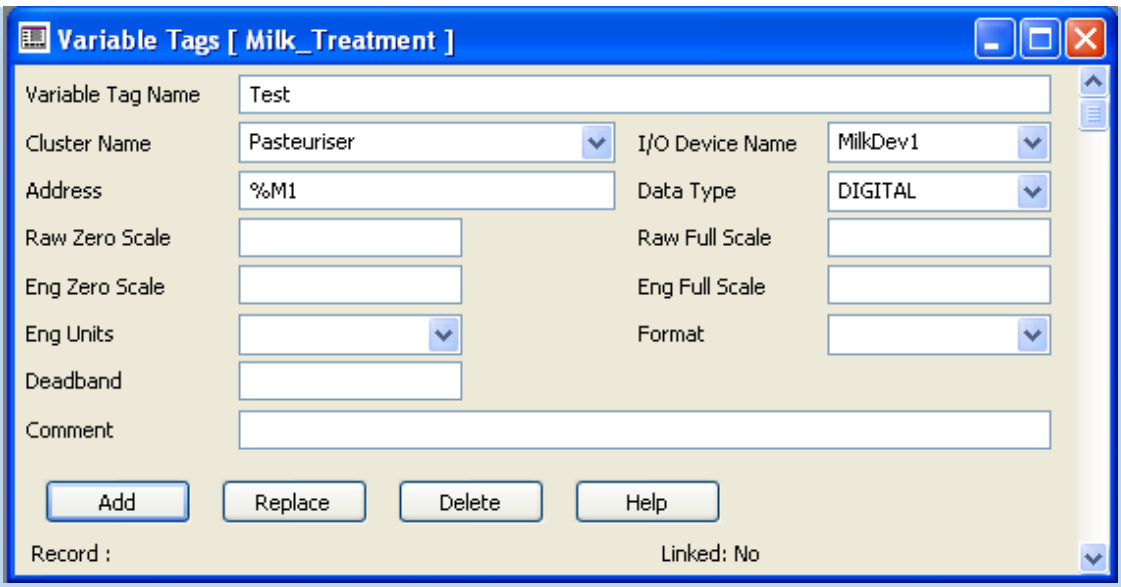


iv. Click "Rebuild All and Transfer".



2 **Define a variable tag called Test in Vejio Citict.**

v. Return to the Citect Project Editor. Choose **Tags >> Variable Tags** from the menu. The Variable Tags form will be displayed:



vi. Fill in the fields of the form as follows:

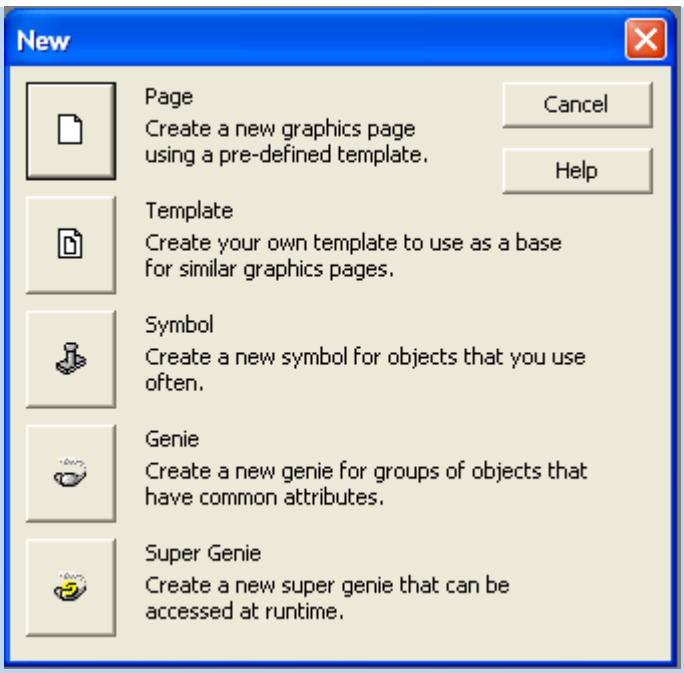
Variable Tag Name	Test
Cluster Name	Pasteuriser
Data Type	DIGITAL
110 Device Name	MilkDev1
Address	%M1

vii. Click Add to save the record.

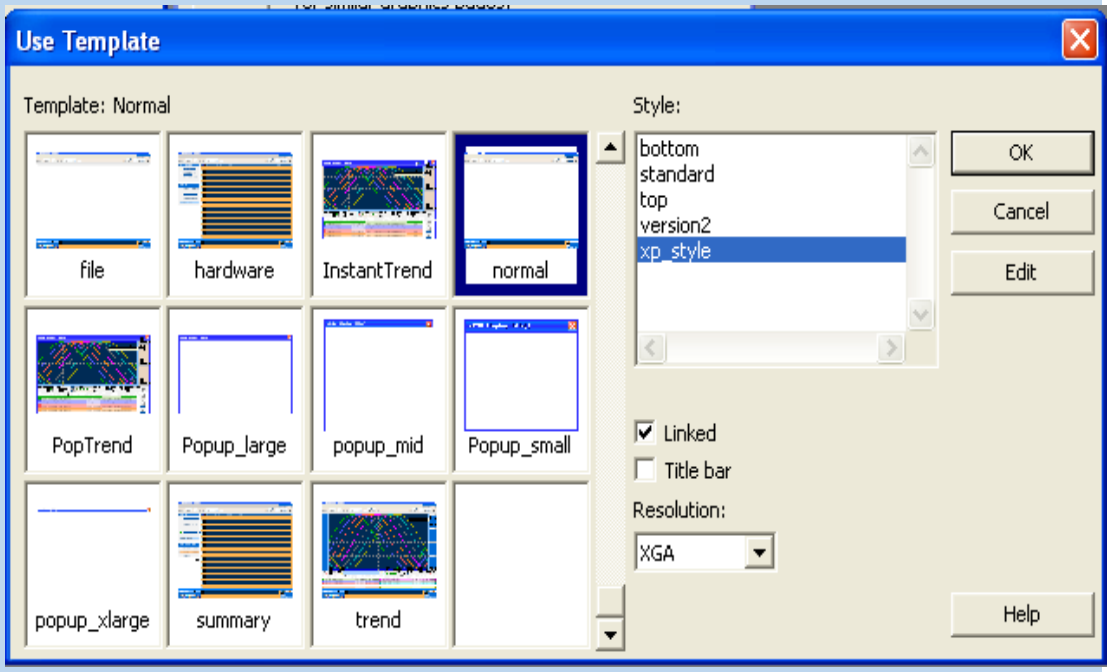
3 **Create a new graphics page to display the value of the Test tag.**

i. Open the Citect Graphics Builder, click on the new button  .

4 **Click the Page button when the following dialog is displayed:**



i. The following dialog will be displayed:



ii. Choose the xp_style with XGA resolution and no Title bar. Then select the Normal template. Click OK to create a new page based on the Normal template.

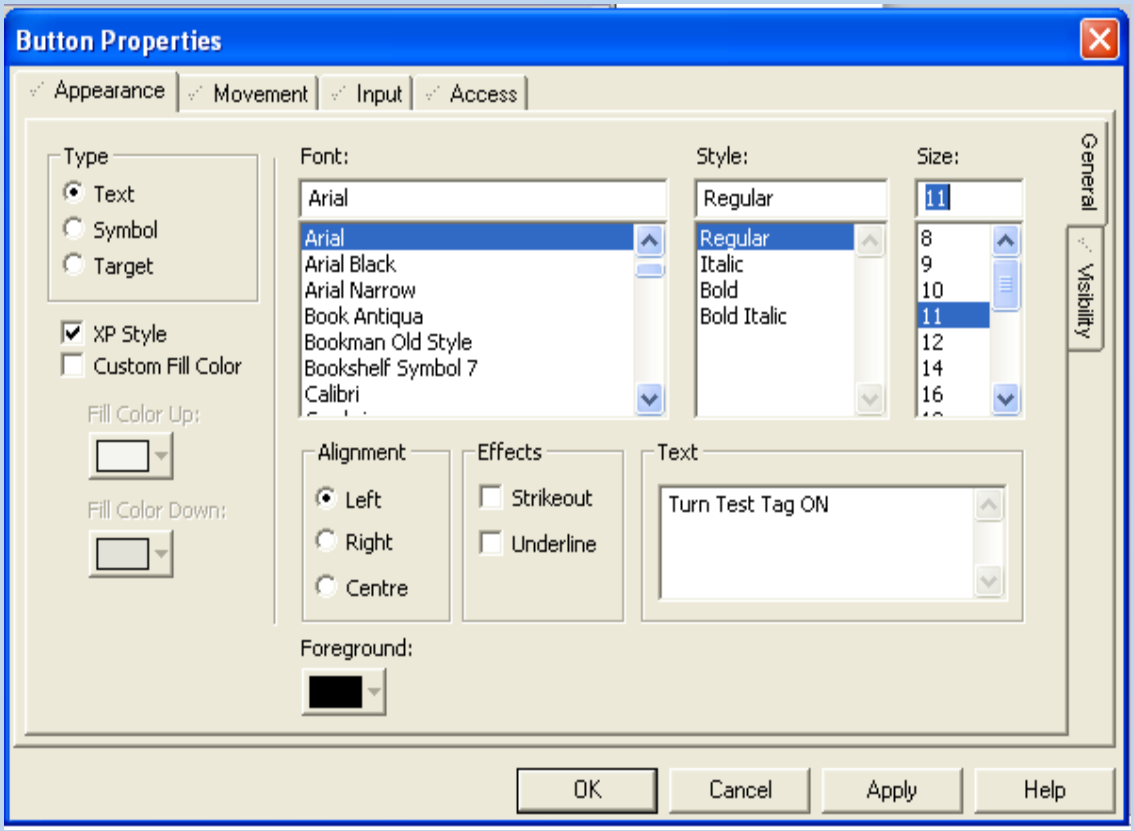
5	**Draw a button on the page.**

i. Select the Button tool  from the toolbox.

ii. Move the mouse to where you want the button to start and press (and hold) the left mouse button. Drag the mouse to where you want the button to finish and release the mouse button.

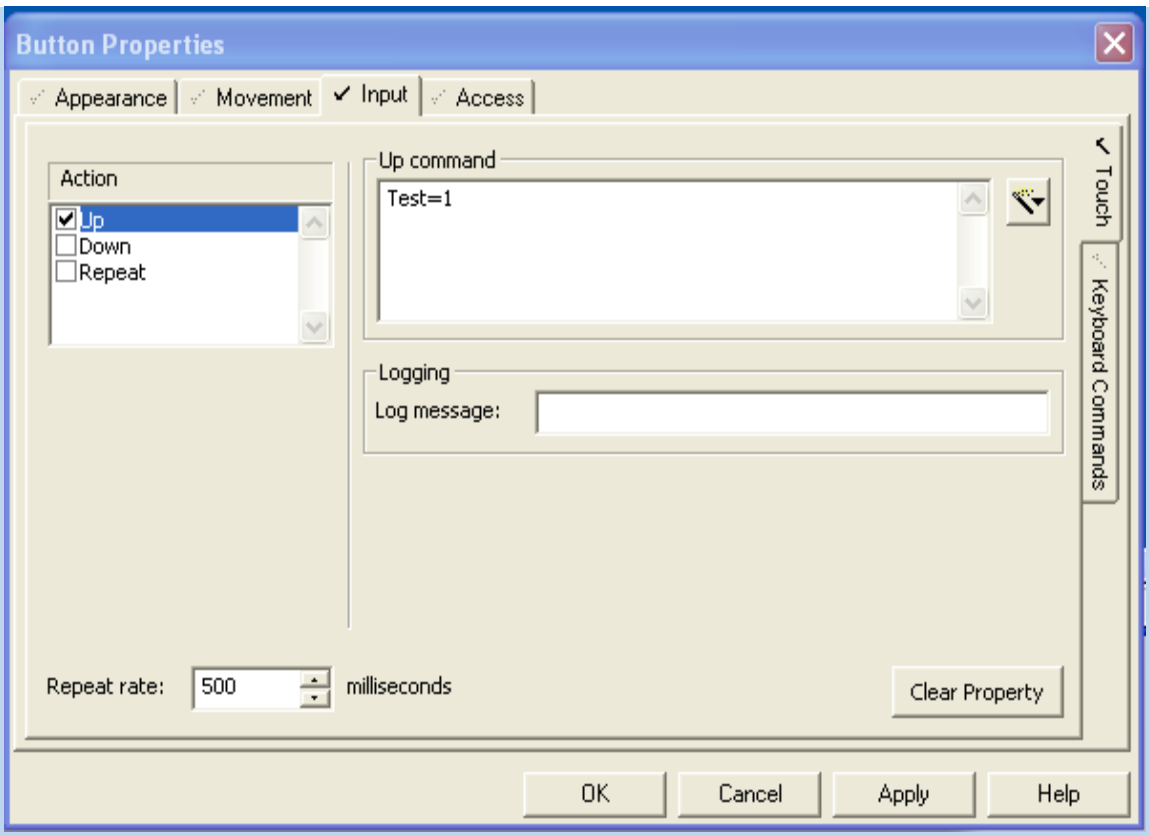
6

i. The following dialog will be displayed:



Complete the dialog as shown (but do not click OK).

ii. Click on the Input tab (at the top of the dialog) and the following dialog will be displayed. In the Up command field type Test = 1. Click OK.



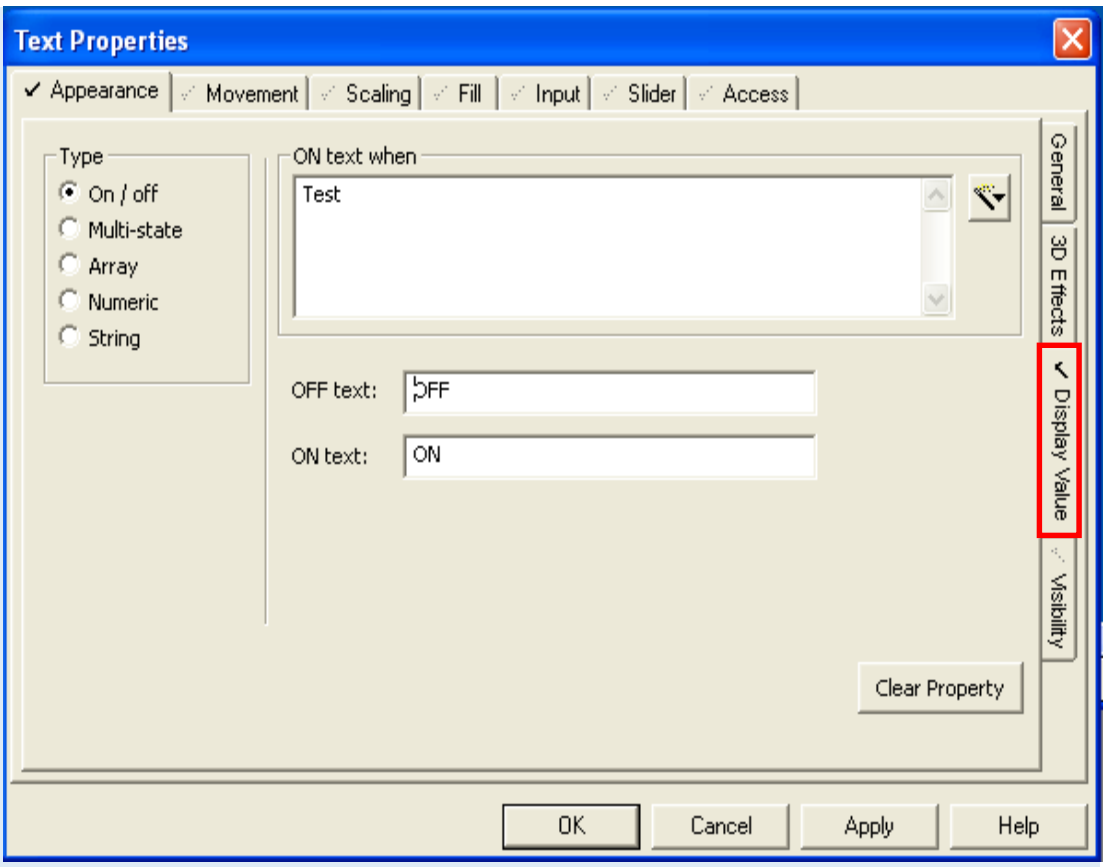
iii. Draw another button except in this case, type Turn Test Tag Off in the Text field and Test = 0 in the Up command field.

7 **Create a text object to display the value of the Test tag.**

i. Select the Text tool  from the toolbox and move the cursor to a point near the buttons you drew - by doing this we will place some text on the page to display the value of the Test tag.

ii. Press the space bar once and then click the left mouse button. A dialog will be displayed.

iii. Click on the Appearance tab along the top of the dialog and then click on the Display Value tab at the right of the dialog. The following dialog will be displayed:

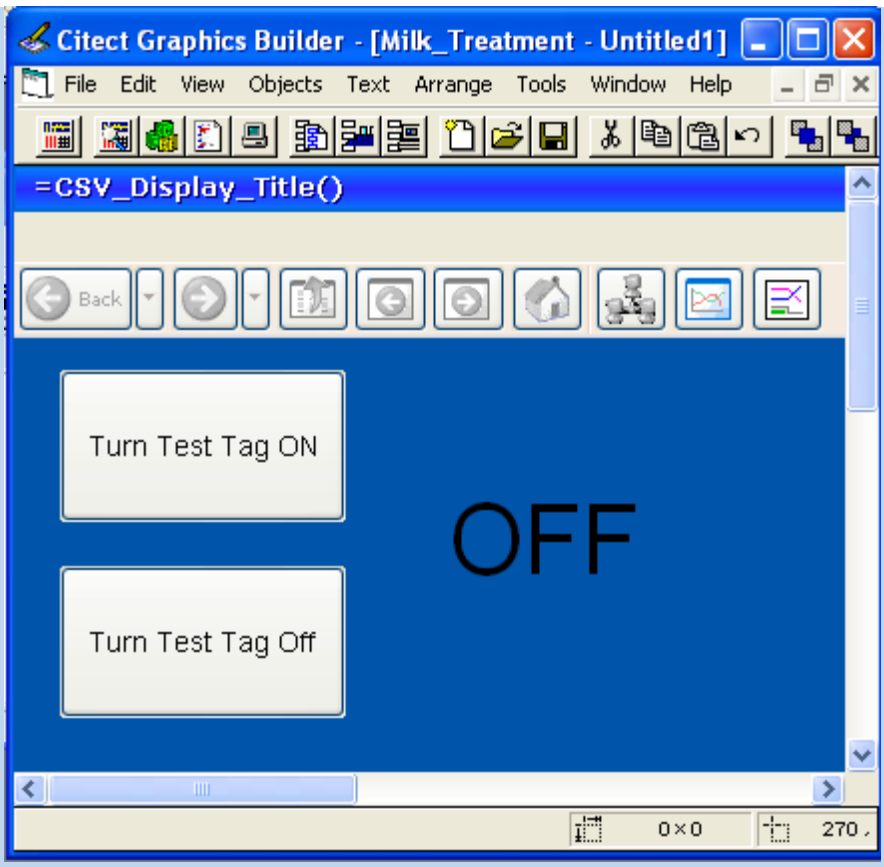


Note: Don't forget to look for the vertical tabs in these dialog boxes

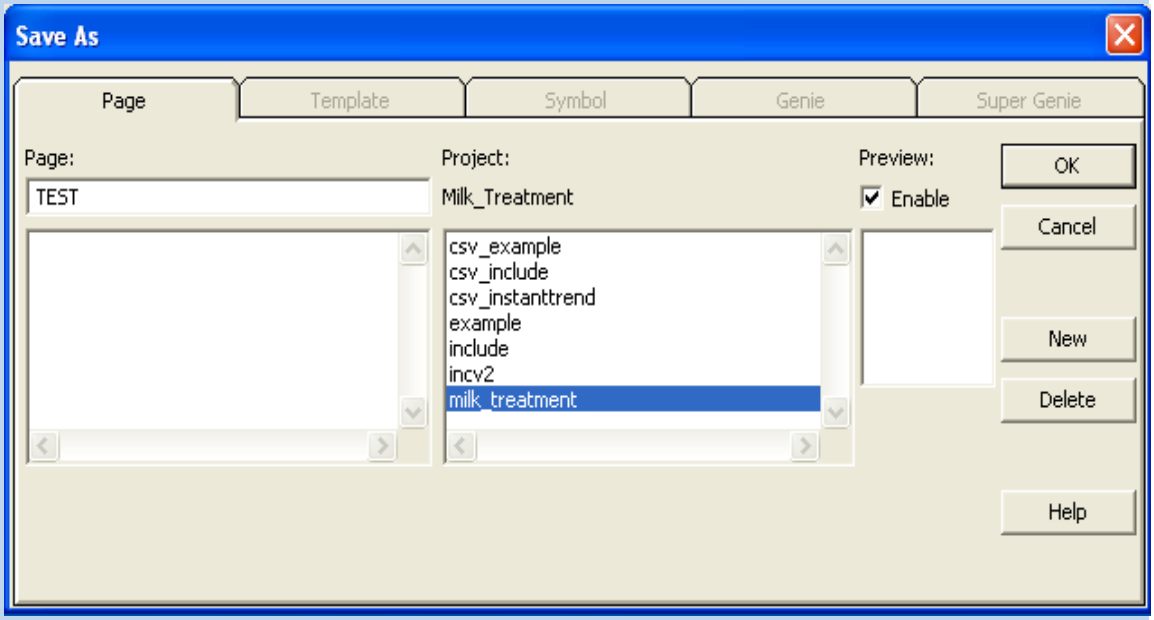
iv. In the ON text when field type Test (as a default the Type should already be set as On I off), in the OFF text field type OFF and in the ON text field type ON. Click OK.

8 **Save the page.**

i. Your page should now look similar to the following:



ii. Click on the **Save** button on the tool bar to save the page. The following dialog will appear:



iii. Select **Milk_Treatment** from the Project list and then type **Test** in the **Page** field.

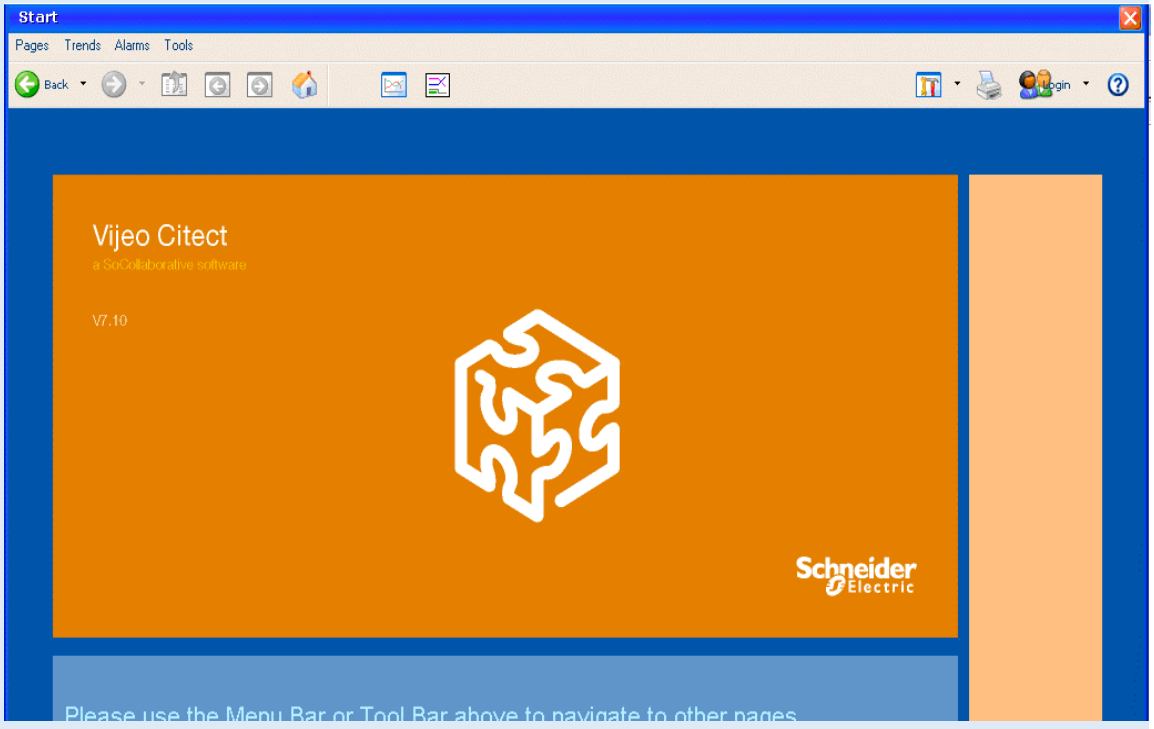
iv. Click **OK**.

Run the project and check that you have communications.

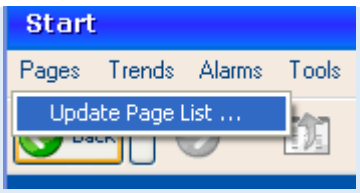
i. Click on the Run Project button  .

Note: When you compile the project, Vijeo Citect checks for errors. If the compiler reports any errors, select Goto to display the place where an error exists. When you have corrected all errors, save your pages and click the Run Project button again.

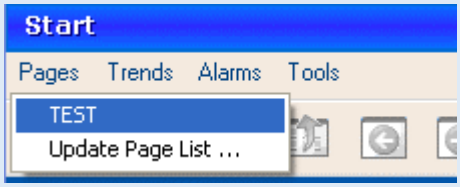
When the project starts up the page CSV Start is displayed.



ii. Select the menu **Pages» Update Page List...**



iii. The new page Test will be added to the menu. Go to the **Test** page.



iv. Check that you have communications and can change the value of the **Test** tag. If the I/O Server and I/O Device are not communicating, then **#COM** will appear where the text was drawn.

v. To shutdown the project, click on the close title bar of the project window.

5.7. Structured Tag Names

5.7.1. Overview

Vijeo Citect puts few restrictions on the names of variable tags, but you will benefit from using a tag naming convention. By using a tag naming convention, your project will be easier and faster to design, configure, and commission, and will require less time for future maintenance.

The following naming convention is recommended for a Vijeo Citect system to obtain maximum benefit when using features such as Genies and Super Genies.

Each tag name can contain up to 32 characters. To establish a convention, you must divide the characters in the tag name into sections that describe characteristics of the tag, for example, the area where the tag is located, the type of variable, and any specific attributes. Four basic sections are suggested for a Vijeo Citect naming convention:

Area_Type _Occurrence_Attribute

5.7.2. Area

The Area section identifies a plant area, number, or name. If you use a prefix that identifies tags within a particular area, you can easily duplicate all Vijeo Citect functions within the area. For example, if you have three pasteurisers with the same controls on each pasteuriser, you can configure the tags for pasteuriser number one, and copy the tags to pasteurisers two and three. You then only need to change the area section in the tag names to the area of the second and third pasteuriser. The remainder of the tags remains unchanged, for example:

Part	Tag Name
Pasteuriser 1	P1_TIC_101_PV
Pasteuriser 2	P2_TIC_101_PV
Pasteuriser 3	P3_TIC_101_PV

If you do not need this facility, you can omit the Area section of the Tag Name to reduce the number of characters in the tag.

5.7.3. Type

The Type section identifies the Type of parameter, process equipment, or control hardware. The ISA standard naming system is recommended.

Variable Tag	Meaning
P1_TIC_101_PV	Temperature indicating controller
P1_ FIC_ 101_ PV	Flow Indicating controller
P1_ PUMP_101_PV	Pump
P1_VALVE_101_PV	Valve

5.7.4. Occurrence

The Occurrence section identifies the equipment number.

Variable Tag	Meaning
P1_TIC_101_PV	Temperature indicating controller 101
P1_TIC_102_ PV	Temperature indicating controller 102
P1_PUMP_101_PV	Pump 101
P1_PUMP_102_PV	Pump 102

5.7.5. Attribute

The Attribute section identifies the attribute or particular parameter that is associated with the equipment.

Variable Tag	Meaning
P1_TIC_101_PV	Process Variable
P1_TIC_101_SP	Setpoint
P1_TIC_101_OP	Output
P1_TIC_101_P	Gain or proportional band
P1_TIC_101_I	Integral
P1_TIC_101_CMD	Command signal to start pump
P1_TIC_101_M	Auto/Manual mode
P1_TIC_101_V	Value (running/stopped)

Now that we have set up and tested the communications between the I/O Server and I/O Device, we can now define the variable tags required for the Milk_Treatment Plant. The variable tags for the Milk_Treatment project will be defined in the same way the Test tag was defined earlier. Variable tags can be configured rapidly, because much of the information for

each tag is the same. If two variable tags are similar, display the information for one tag, change the relevant fields and then click on Add. To correct errors, display the tag and make changes, then click on Replace.

- **Exercise 4-4**

Add the variable tags listed in the following table to your project.

Variable Tag Name	Type	Cluster Name	I/O Device	Address	Raw Zero	Raw Full	Eng Zero	Eng Full	Eng Units	Format
LIC_Silo_PV	INT	Pasteuriser	MilkDev1	%MW0	0	32767	0	1000	Lt	#### EU
LIC_Balance_PV	INT	Pasteuriser	MilkDev1	%MW1	0	32767	0	100	Lt	### EU
SIC_Cent_PV	INT	Pasteuriser	MilkDev1	%MW2	0	32767	0	1000	PRM	#### EU
PIC_Homog_PV	INT	Pasteuriser	MilkDev1	%MW3	0	32767	0	100	Kpa	### EU
TIC_Cool_PV	INT	Pasteuriser	MilkDev1	%MW4	0	32767	-10	120	deg	### EU
TIC_P1_PV	INT	Pasteuriser	MilkDev1	%MW5	0	32767	-10	120	deg	### EU
TIC_P2_PV	INT	Pasteuriser	MilkDev1	%MW6	0	32767	-10	120	deg	### EU
TIC_P3_PV	INT	Pasteuriser	MilkDev1	%MW7	0	32767	-10	120	deg	### EU
TIC_P4_PV	INT	Pasteuriser	MilkDev1	%MW8	0	32767	-10	120	deg	### EU
TIC_HW_PV	INT	Pasteuriser	MilkDev1	%MW9	0	32767	-10	120	deg	### EU
TIC_Hold_PV	INT	Pasteuriser	MilkDev1	%MW10	0	32767	-10	120	deg	### EU
Agitator_Silo_V	DIGITAL	Pasteuriser	MilkDev1	%M2						
Agitator_Alfast_V	DIGITAL	Pasteuriser	MilkDev1	%M3						
Centrifuge_Clar_V	DIGITAL	Pasteuriser	MilkDev1	%M4						
Pump_Feed_CMD	DIGITAL	Pasteuriser	MilkDev1	%M5						
Valve_Cool_CMD	DIGITAL	Pasteuriser	MilkDev1	%M6						
Valve_HW_CMD	DIGITAL	Pasteuriser	MilkDev1	%M7						
Valve_Flow_CMD	DIGITAL	Pasteuriser	MilkDev1	%M8						

Chapter Six “Graphics”

6.1. Introduction

The graphics pages are one of the principle components of a Citect system. They are the interface to plant operators, and can be designed to display data as well as to accept operator input. Graphics pages comprise of a page template, the objects drawn on the page, as well as properties specific to the page.

6.2. Creating Pages

6.2.1. Overview

When starting a new project, new pages created can be designed to follow the "look and feel" that is required. The simplest page is based upon the Blank template, which is simply a blank window. The developer can then add objects and functionality to the window and develop new templates for their project pages.

6.2.2. Standard Templates

For a developer with a short development schedule or someone who is new to Vijeo Citect, there are a number of pre-designed templates that enable the user to quickly create pages. The standard templates in the Include project are designed to reliably perform all of the basic functions required in a project.

6.2.3. Creating a New Page

6.2.3.1. How to To create a new graphics page:

Double click on the Create a new page button in the Citect Explorer.

or

click on the **New**  button in the Citect Graphics Builder,

Or

Select the menu **File » New ...** in the Citect Graphics Builder

- **Exercise 6-1**

Create a new graphics page based on the xp_style Normal template.

Step	Action
1	**Create a new page in your Milk_Treatment project - use the following template settings:** **Style** — **xp_style** **Linked** — √ **Resolution** — XGA **Template** — Normal
2	**Save your page, naming it Pasteuriser.**

6.2.4. Drawing Basic Objects

6.2.4.1. Overview

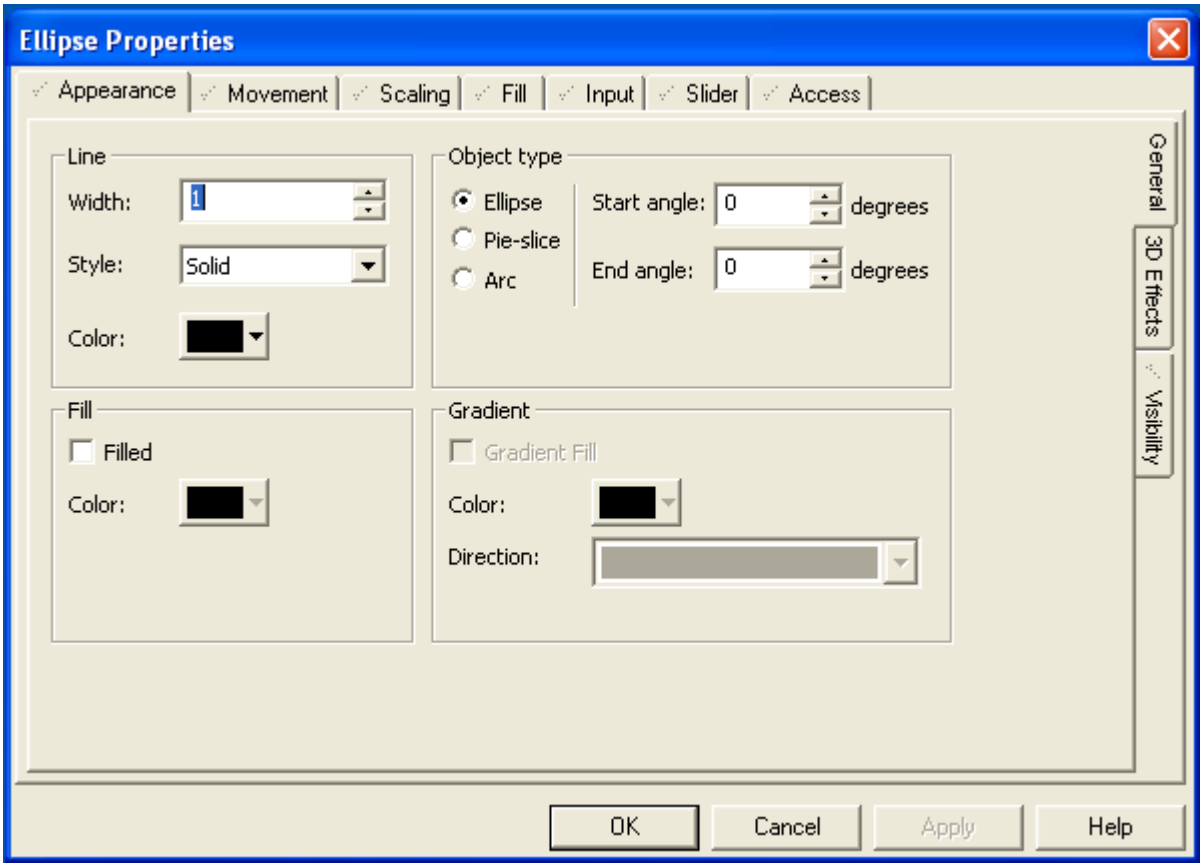
Objects can be drawn on a page by selecting the object from the toolbox, or by selecting the object from the Objects menu.

The steps for drawing each of the objects varies slightly – look in the Drawing category of Citect Help Index for specific information on how to draw each object.

The objects can be manipulated by using the items in the Edit, View, Text and Arrange menus. As in other drawing packages, they can be rotated, enlarged, grouped and aligned.

6.2.4.2. Object Properties

In the previous exercise, you will have come across property dialog boxes. Shown below is the property dialog box for an ellipse object. This dialog box contains all of the properties for that object:



Every object you draw has a set of properties that determine how it looks and behaves. These object properties include such things as the static appearance of an object in the Citect Graphics Builder, as well as runtime properties such as how the object moves and scales or how it responds to a mouse click.

- **Hints & Tips:**

The horizontal tabs are the major sections for the object properties. Each of the horizontal tabs will have different sub-sections in the vertical tabs. When a tab has had information inserted a tick √ appears on the tab. When you are following the screen shots make sure you have both the correct horizontal and vertical tabs selected.

Details about the properties can be found in the Citect Help - Object Properties.

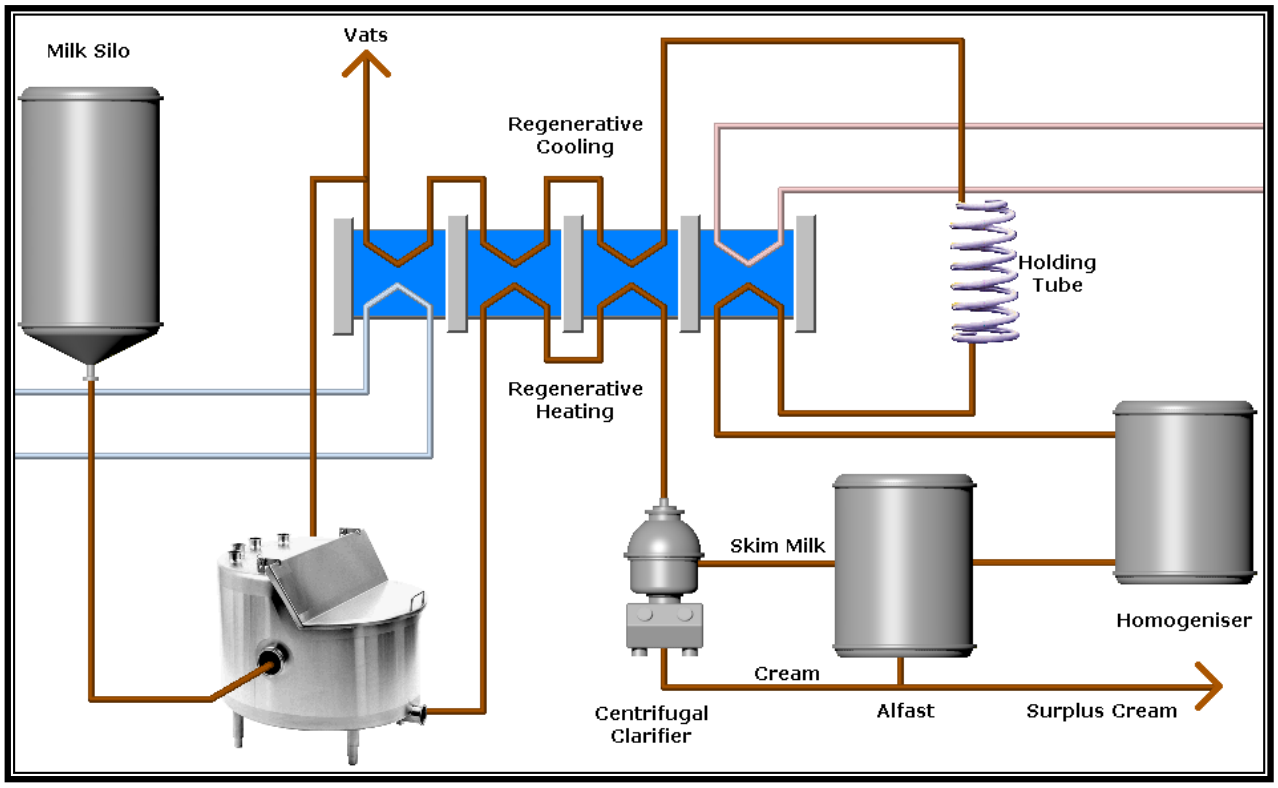
6.2.4.3. Vijeo Citect Libraries

Each Vijeo Citect project may contain one or more library files, each of which can hold a number of library objects such as page templates, symbols and genies (we will look at genies in a later chapter). When you create new symbols, you can create new library as part of your project and store them there. When you backup and restore a project only the libraries that are part of the project are transferred.

6.2.5. Drawing the Milk_Treatment Graphics Page

6.2.5.1. Drawing Basic Objects

The first step in building the graphics page is to draw the basic objects on the page. The objects to be drawn and their approximate positions on the page are shown in the following diagram:

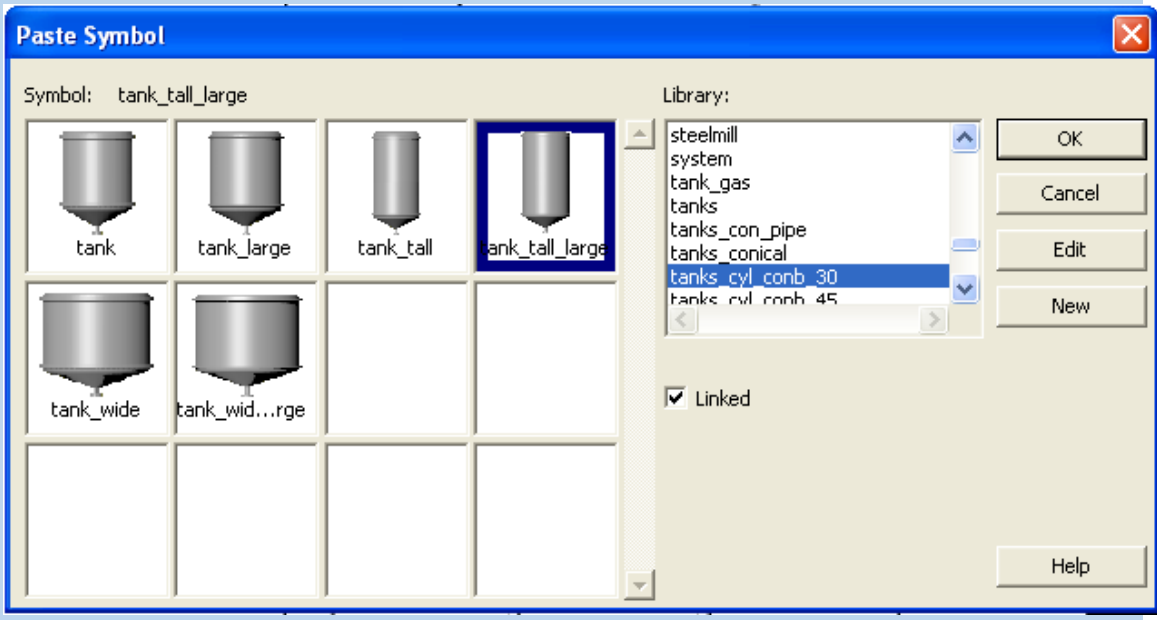


- **Exercise 6-2**

Draw the graphics page as shown on the previous page.

Step	Action
1	**Add the static symbols to your new page.**

Paste the **Milk Silo** symbol onto your page using the **Paste Symbol** tool 



Next paste the following symbols onto your page - leave all of the symbols Linked to the library.

Object	Library	Symbol
Balance Tank	tank_gas	GazTank_large
Holding Tube	Heat_exchange	Large_left
Centrifugal Clarifier	Centrifuge	Centrifuge_large
Alfast	Tanks Cylindrical	Tank_large
Homogeniser		Tank_large

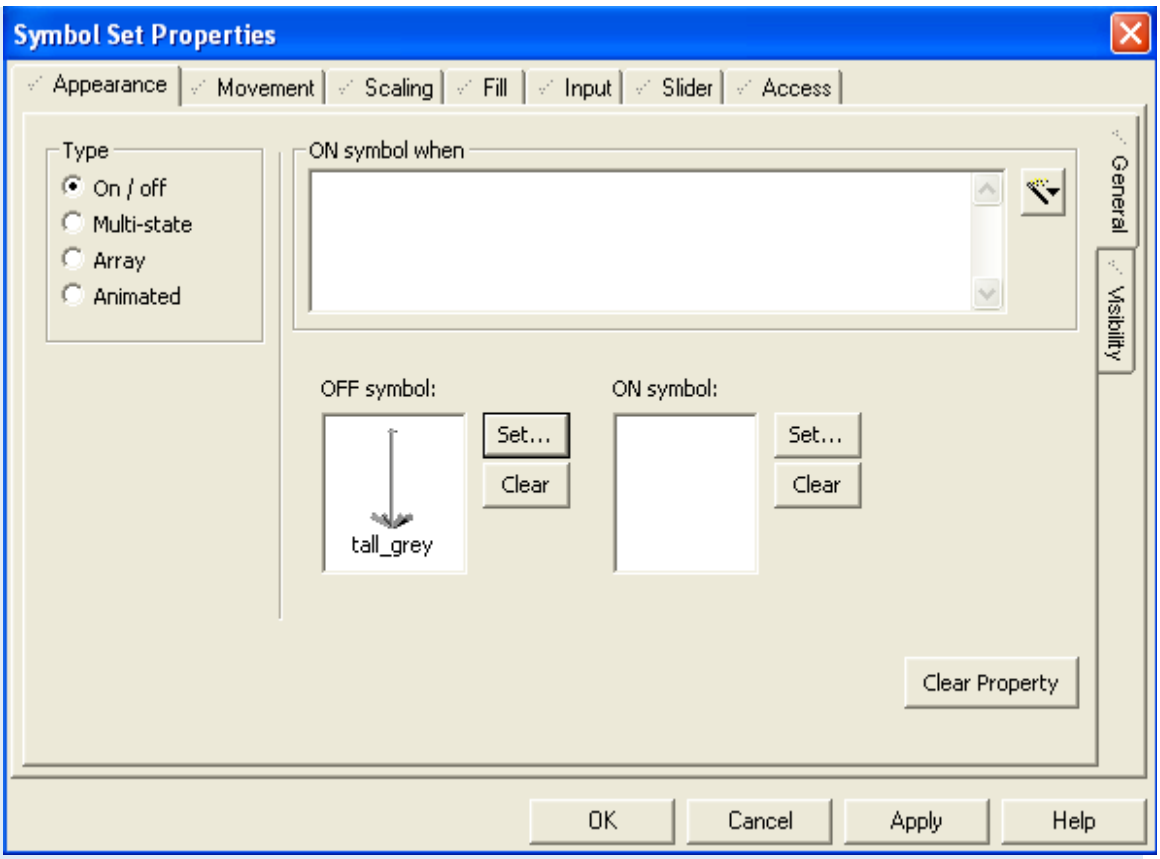
Hints & Tips:

Save your page regularly! Do not wait until the whole page is completed. You may find the items in the **Arrange** menu helpful. If you need to undo an action, select **edit »**

	undo from the menu. To display the grid, go to the menu View » Grid setup ... and select Display Grid.
2	**Add the Symbol Sets to the page. These will be animated later in this chapter.** Using Symbol Set tool  (NOT the Paste Symbol tool) to paste the Milk Silo Agitator and the Alfast Agitator onto your page. Choose On / off in the Appearance (General) properties. Clear the ON symbol and set the OFF symbols according to the following table:

Object	Library	Symbol
Milk Silo Agitator	Agitator _30	Tall_grey
Alfast Agitator	Agitator_15	Agitator_grey

The **Milk Silo Agitator** properties should look like this:



Hints & Tips:

If a dialog opens immediately after placing an object just click OK if you do not want to change its default properties. Turn on Snap to Grid to assist you when you draw the pasteuriser, pipes and arrows.

If you select an action in the Edit, View, Text or Arrange menus and a dialog box opens, click on Help for more information on that action.

6.2.6. Rectangles, Squares, Ellipses and Circles

6.2.6.1. How to Use the Rectangle Tool

You can use the Rectangle tool to draw rectangles and squares. These objects can be moved, resized, reshaped, brought to the front and so on, and their properties edited just like other types of object.

- **To draw a rectangle:**

Click the Rectangle tool 

Move the cursor to where you want the rectangle to start.

Click and drag the mouse to the opposite comer of the rectangle and release the mouse button. If you hold the Shift key before you start drawing the rectangle, it is drawn from its center outwards.

- **To draw a square:**

Click the Rectangle tool 

Press (and hold) the Ctrl key.

Move the cursor to where you want the square to start and click (and hold) the mouse button.

Drag the cursor to the opposite comer of the square and release the mouse button. If you hold the Shift key (and the Ctrl key) before you start drawing the square, it is drawn from its center outwards.

6.2.6.2. How to Use the Ellipse Tool

The Ellipse tool is used to draw ellipses, circles, arcs, and pie-slices. Like the

Rectangle tool the Ellipse tool can be moved, resized, reshaped, brought to the front and so on, and their propeliies edited just like other types of object.

- **To draw an ellipse:**

Click the Ellipse tool 

Move the cursor to a comer of the bounding rectangle (marquee) and click (and hold) the mouse button.

Drag the cursor to the opposite comer of the bounding rectangle and release the mouse button. If you hold the Shift key before you stmi drawing the ellipse, it is drawn from its center outwards.

- **To draw a circle:**

Click the Ellipse tool 

Press (and hold) the Ctrl key.

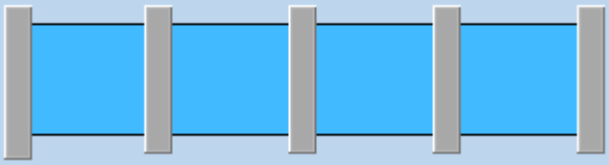
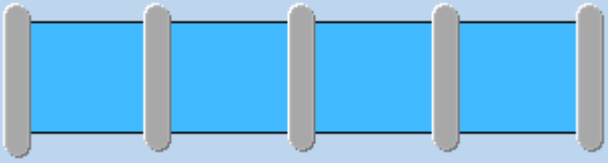
Move the cursor to a comer of the bounding rectangle (marquee) and click (and hold) the mouse button.

Drag the cursor to the opposite comer of the bounding rectangle and release the mouse button. If you hold the Shift key and the Ctrl key before you start drawing the circle, it is drawn from its center outwards.

- **Exercise 6-3**

Use squares and rectangles to create the Pasteuriser.

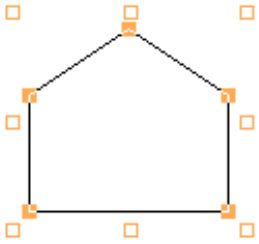
Step	Action
1	**Add rectangles to your page.** i. Use the Rectangle tool to draw the Pasteuriser. This object can be as simple as four rectangles.  ii. Experiment with the 3D effects and add more rectangles to create the look of four separate chambers.

	 iii. Change the Corner Radius property of the rectangles to add rounded corners. 
2	**Save the page.**

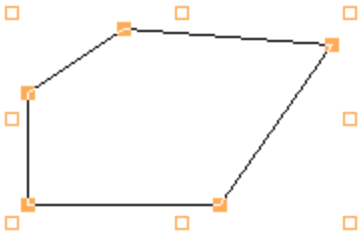
6.2.7. Pipes, Polylines and Polygons

6.2.7.1. Node Editing

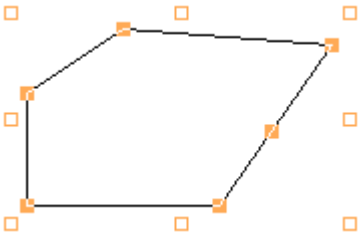
Pipe, Polyline, or Polygon objects can be edited to change their shape. Each of these objects consists of a continuous series of lines drawn between structural anchor points called nodes. Nodes are visible when an object is selected. Each node appears as a small square located at specific anchor points along the object. There is always a node located at the start and end of a polyline or pipe, and at every change of direction in an object's shape.



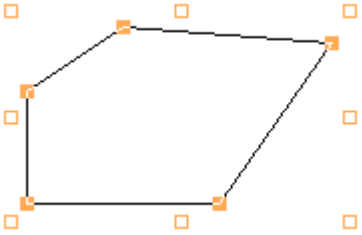
Pipe, Polyline, and Polygon objects can have their shapes changed in many ways. Their nodes can be selected individually or by group and moved to a different position, thus changing the shape of the object.



The Pipe, Polyline, and Polygon objects also support node adding and deleting.



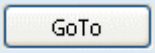
Position the cursor over the LINE and press INSERT



Position the cursor over the NODE and press DELETE

- **Exercise 6-4**

Add the rest of the static objects to your page.

Step	Action
1	**Add pipes and lines to your page.** i. Draw the pipes using the pipe tool  double click to terminate. ii. Draw the arrows using the line tool  . iii. Draw the text next to each object with the text tool  iv. Save your page.
2	**Compile and run the project and test your modifications.** i. Click on the Run Project button  in the Citect Project Editor. ii. If the compiler reports any errors, select  to display the dialog where an error exists and fix the problem.

	iii. When you have corrected all errors, click on Run Project again iv. Go to the Pasteuriser page and check that it displays correctly.
3	**Shutdown the project.** **i.** Click in the close box to shutdown.

6.3. Runtime Properties

6.3.1. Overview

When you drew the objects on your Pasteuriser page, they were drawn so that at runtime they would appear the same as they were drawn. This is fine if the objects that were drawn are intended only as a static graphical representation of your plant. However, at runtime you want to receive infoffi1ation about your plant from the graphical display - this can be done by setting the properties of the objects and by drawing some additional objects.

The following diagram shows the next stage in drawing the Pasteuriser page:

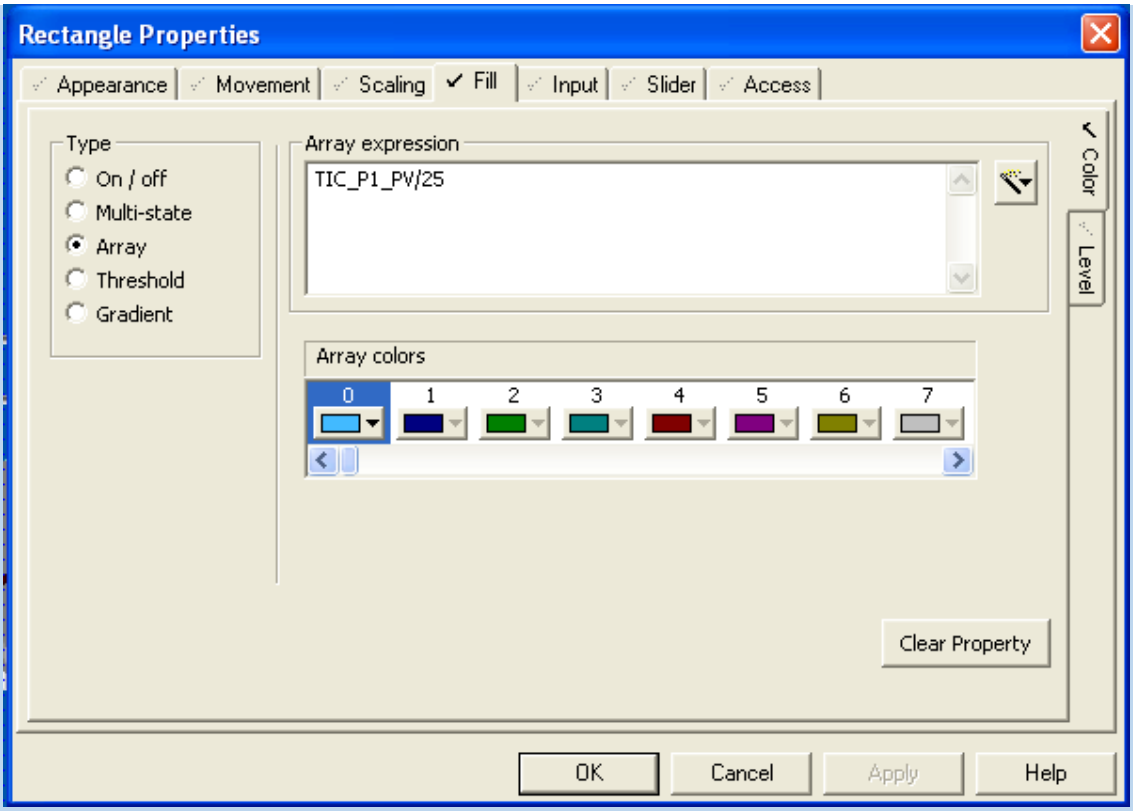
6.3.2. Overview

Graphics page objects have dynamic properties such as colour, size and position, which can change at runtime, to reflect a change in condition. For example, the colour of the Pasteuriser would be an ideal way of conveying its temperature to the operator.

- **Exercise 6-5**

Change the runtime properties of the Pasteuriser chambers to indicate their temperatures.

Step	Action
1	**Add an Array Expression to the runtime properties of the Pasteuriser objects.** i. Double click the first rectangle to open up the **Properties** dialog. ii. On the **Appearance (General)** properties, tick the **Filled** box. iii. Go to the **Fill (Colour)** tab - select Type: Array and then click on the Help button and read the information under the headings [Type] Array, Array expression and Array colours. iv. In the Array expression field type: TIC PI PV/25 --- **Hints & Tips:** You can use the Expression Wizard  to add tag names or function names, this helps avoid typing errors. --- v. Choose the Array colours for values 0, 1, 2, 3, 4 and 5 by clicking on the colour swatch.



Hints & Tips:

The engineering range of the pasteuriser tags TIC_P1_PV to TIC_P4_PV is from -10 to 120 degrees, so TIC_P1_PV 125 evaluated as an integer, will give a value from 0 to 5.

vi. Repeat for the other three Pasteuriser chambers.

vii. Click OK.

2 **Save your page.**

6.4. Displaying a Bar Graph.

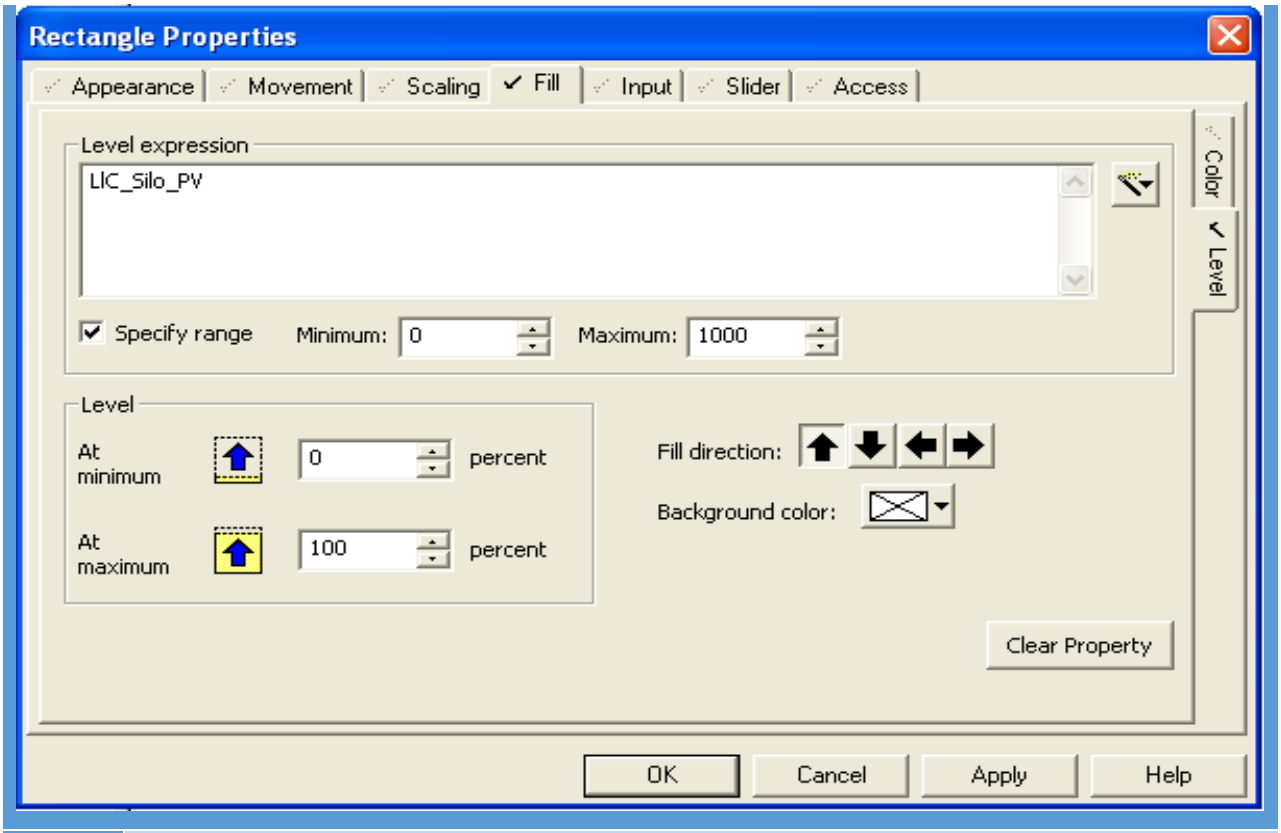
6.4.1. Overview

One of the dynamic properties is Fill. If fill is used in a rectangle object, then a bar graph can be constructed to display a changing value. For example, you can draw a bar graph and use the Fill (Level) properties to indicate the level of the milk silo.

- **Exercise 6-6**

Use runtime properties to create an indicator to display the silo levels.

Step	Action
1	**Add a bar graph that displays the Milk Silo Level.** i. Select the Rectangle tool  and draw a filled bar on the Milk Silo - tick Filled and Border in the Appearance (General) properties. ii. Go to the Fill (Level) properties of the rectangle and insert the tag “LIC_Silo_PV” into the Level expression. Then click OK. 



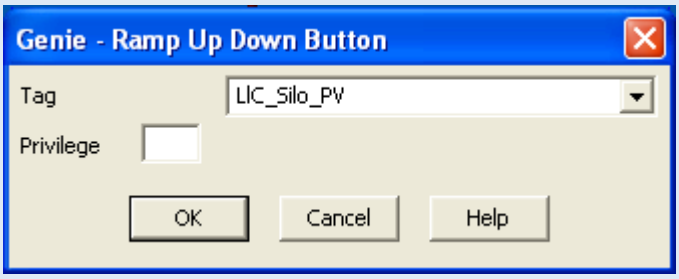
iii. Add a bar graph to the Balance Tank to display the value of the tag LIC_Balance_PV.

2 **Paste a pre-built genie to change the Milk Silo level.**

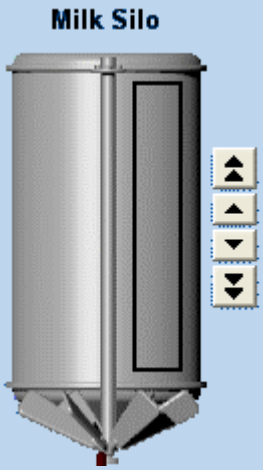
i. To change the value of the milk silo level. Click on the Paste Genie tool  .

ii. From the Controls library select the **Ramp_UpDown_btn2** Genie.

iii. In the Dialog box, select the LIC_Silo_PV variable tag.



iv. Place the Genie alongside the bar graph as shown below:

	
	v. Place another genie alongside the Balance Tank. Select the tag LIC_Balance_PV. vi. Save your page.

6.5. Displaying Text at Runtime

6.5.1. Overview

You can display different text messages on a graphics page, depending on the state of a digital tag or a condition. For example, you can display the word RUNNING next to a motor when it is on, and the word STOPPED in the same location when the motor is off.

- **Exercise 6-7**

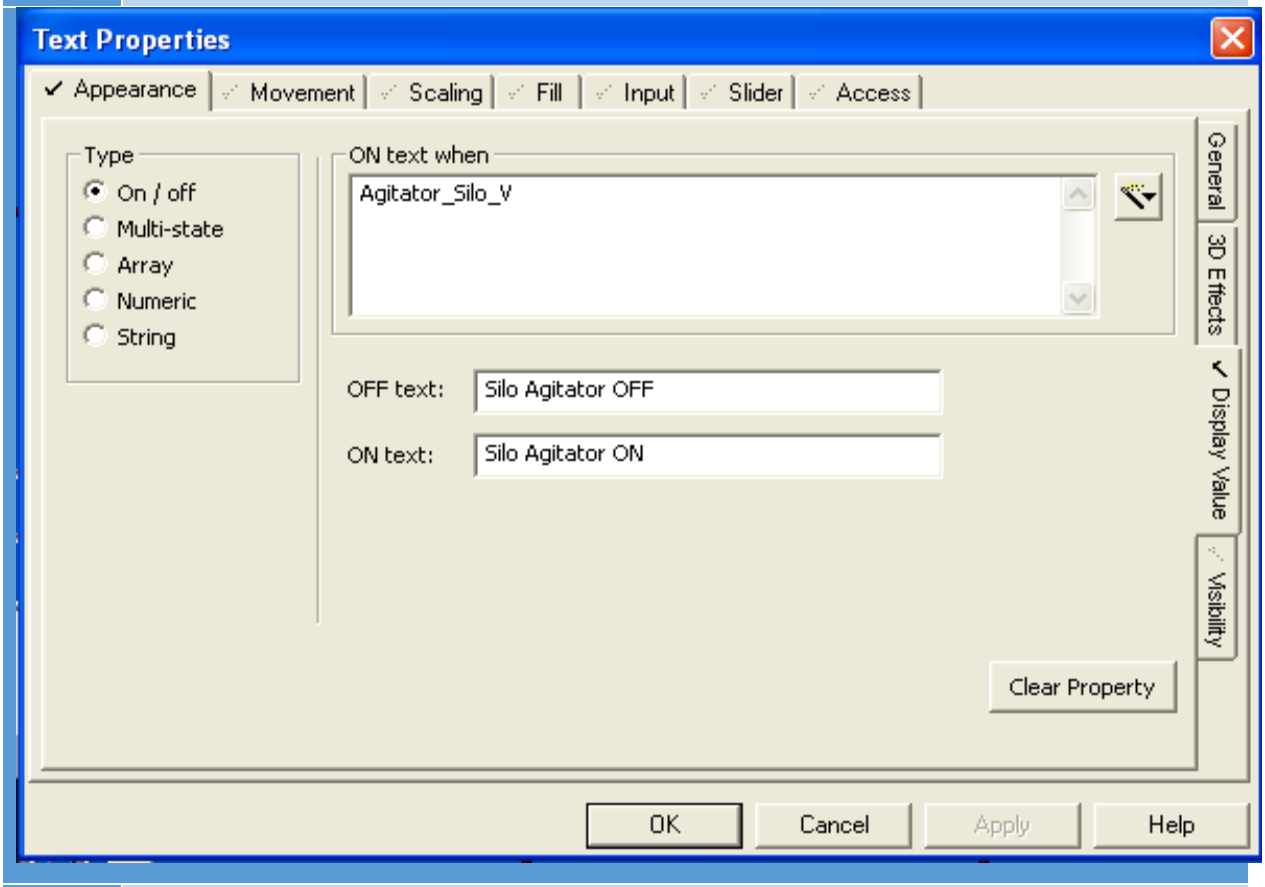
Display the status of the tag Agitator_Silo_V near the Milk Silo.

Step	Action
1	**Add a text object to the Pasteuriser page.** i. Click on the Text tool  in the Citect Graphics Builder. ii. Type the words Silo Agitator OFF. iii. Move the mouse pointer to where you want the text to be displayed and click the left mouse button to place the text. iv. Go to the Appearance (General) properties and set the desired display font, colour, alignment and effects.

v. Go to the Appearance (Display Value) properties and select Type: On I off.

vi. In the ON text when box enter the variable tag Agitator_Silo_V.

vii. Set the ON Text to Silo Agitator ON.



viii. Click OK.

2 **Repeat for the following conditions and text - place the text near the relevant object on your page:**

Tag		Text
Agitator_Alfast_V	OFF	Alfast Agitator Off
	ON	Alfast Agitator On
Centrifuge_Clar_V	OFF	Centrifugal Clarifier Off
	ON	Centrifugal Clarifier On

3 **Save the page when finished.**

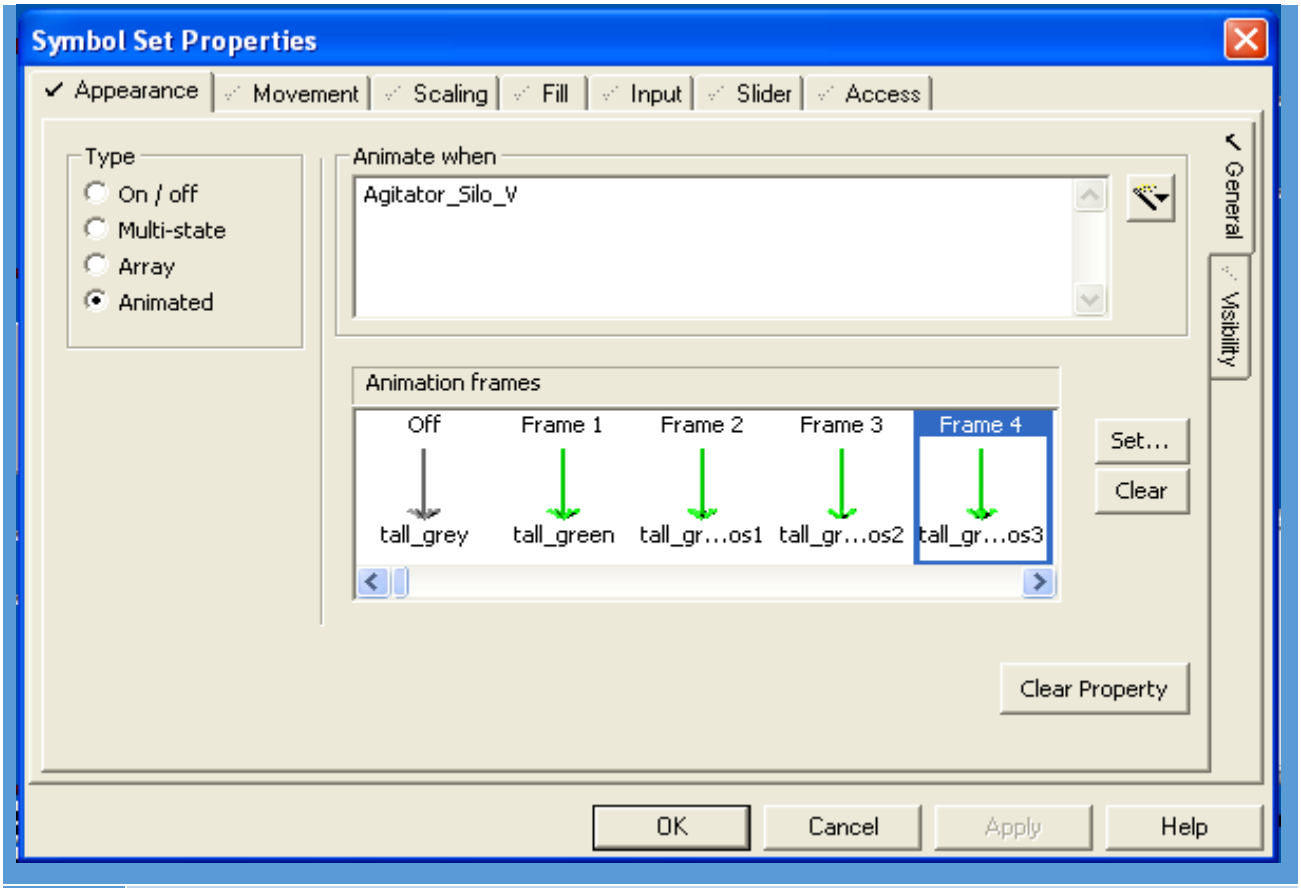
6.6. Using Symbol Sets

6.6.1. Overview

Symbol Sets can be used to display different symbols on your page depending on different variable tag values and expressions, as well as to simulate movement. You have already used the Symbol Set tool  to paste the Silo Agitator and Alfast Agitator onto your page - we will now go back to these items and add properties that will make these objects appear they are rotating when they are switched on.

- **Exercise 6-8**

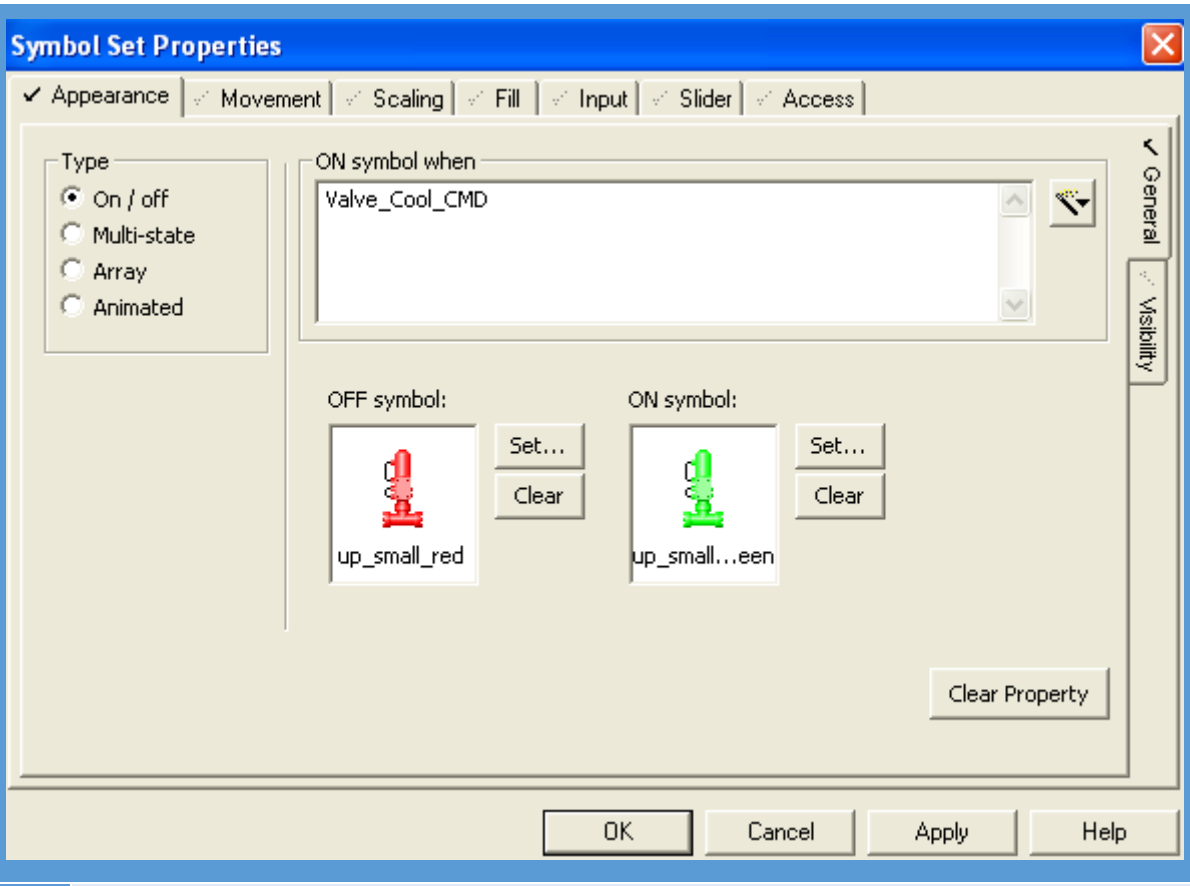
Step	Action
1	**Modify the Silo Agitator so that is animates at run time.** i. Double click on the agitator symbol to display the properties dialog. ii. Go to the **Appearance (General)** properties and select **Type: Animated**. Insert the tag “Agitator_Silo_V” into the Animate when box. **Hints & Tips:** You can use the Expression Wizard  to insert the tag name in to the Animate when expression. iii. In the Animation frames box, you will see that the symbol you previously selected is shown as the Off frame. By clicking on the Set ... button you can fill Frame 1 through to Frame 4 with the symbols you want displayed while the expression in Animate when is true. **Note** You will find that in the Agitator_30 library there are 4 tall_green symbols in order from left to right - each is drawn to show increasing rotation.



	iv. Click on OK.
2	**Modify the Alfast Agitator so that it animates when the tag "Agitator_Alfast_V" is true.**
3	**Add valves onto the pipes** i. Use the Symbol Set tool to add the following 3 valves. ii. Select the On/off type of Appearance (General) properties as follows:

Object	On symbol when		Symbols
Coolant Valve	Valve_Cool_CMD	OFF	valve_solenoid.up_small_red
		ON	valve_solenoid.up small_green
Hot Water Valve	Valve_HW_CMD	OFF	valve_solenoid.up_small_red
		ON	valve_solenoid.up_small_ green

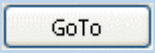
Flow Diversion Valve	Valve_Flow_CMD	OFF	valve_solenoid. right_small_red
		ON	valve_solenoid. right_small_ green



4	**Add a pump to the pipe between the Balance Tank and the Pasteuriser.** i. Use the Symbol Set tool to paste the pump. Select the On/off type of Appearance (General) properties as follows:

Object	On symbol when		Symbols
Feed_Pump	Pump_Feed_ CMD	OFF	pumps_base_small.right_red
		ON	pumps_base_small.right_green

5	**Save your page.**
6	**Compile and run the project and test your modifications.**

	i. Click on the Run Project button  in the Citect Project Editor. ii. If the compiler reports any errors select  to display the dialog where an error exists and fixes the problem. iii. When you have corrected all errors click on **Run Project** again. iv. Test the pages and objects in your project.
7	**Shutdown the project.**

6.7. ActiveX (Optional)

6.7.1. Overview

Vijeo Citect allows you to incorporate ActiveX objects into the graphics pages of your Vijeo Citect project, allowing you to make use of tools and components that have been developed independently of Vijeo Citect. For example, you could incorporate a batch application in a Vijeo Citect graphics page, and use it to communicate directly with your I/O Devices to control and monitor recipe levels, etc.

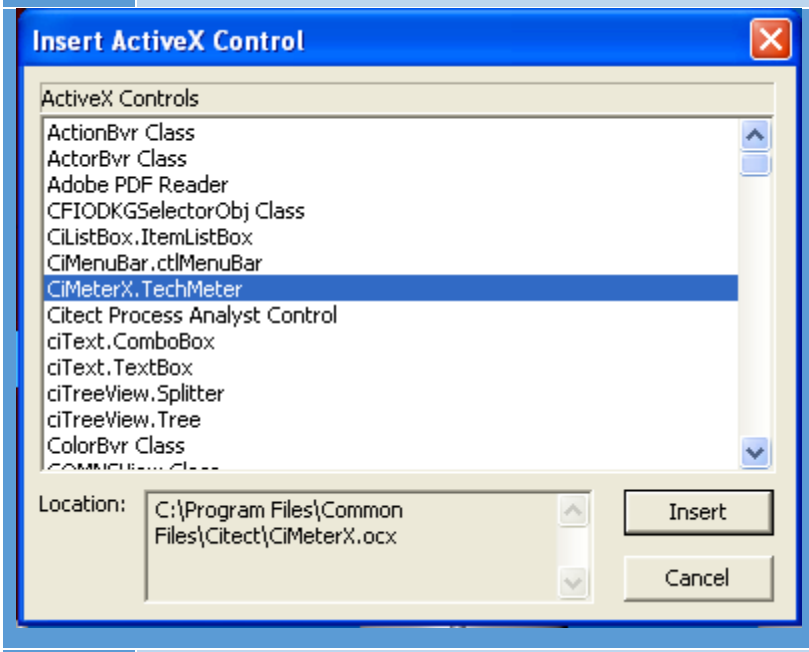
It is important to remember that the behavior of an ActiveX object within Vijeo Citect will be largely determined by the object itself. An object's functionality, reliability and its suitability to Vijeo Citect will depend on the way the object has been developed by its creator.

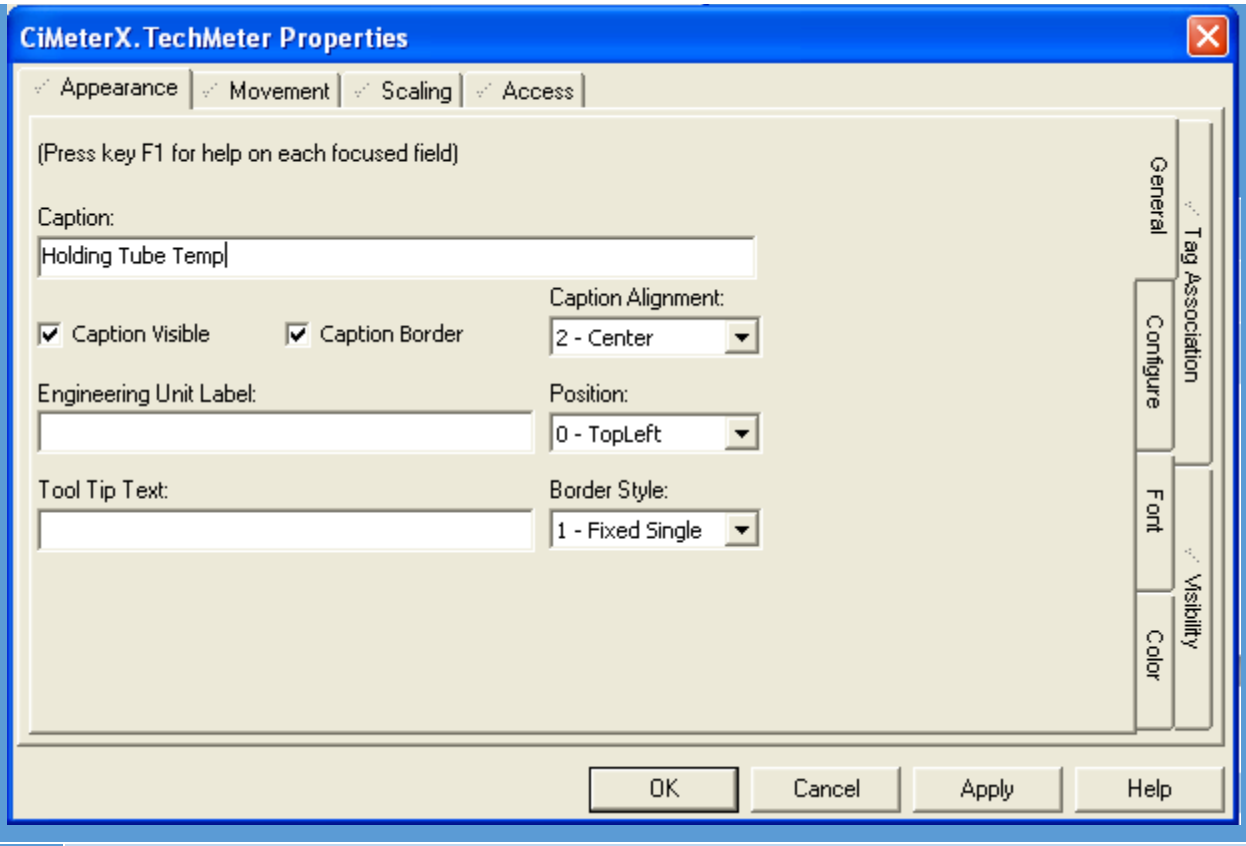
6.7.2. Inserting ActiveX Objects via Citect Graphics Builder

The easiest way to insert an ActiveX object into your Vijeo Citect project is via the Citect Graphics Builder. Using the ActiveX tool from the Citect Graphics Builder toolbox, you can select and insert ActiveX objects in your graphics pages in much the same way conventional objects are inserted. Just like other objects, they can also be moved, reshaped, and copied.

ActiveX objects have predefined properties that relate to their features and functionality. You can view these by simply double clicking on an object once it has been positioned.

- **Exercise 4-9**

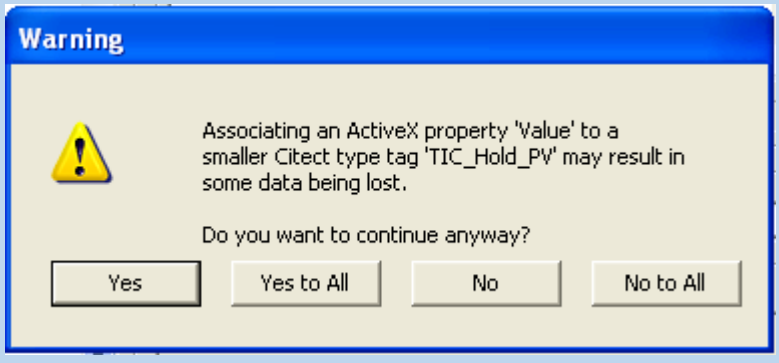
Step	Action
1	**Use the Active X Object CiMeterX to display the Holding Tube Temperature.** i. On the Pasteuriser page choose the Active X button  from the Objects menu. ii. Choose **CiMeterX.Techmeter** from the dialog box displayed. Click **insert**.
	
	iii. Change the Caption to Holding Tube Temp.



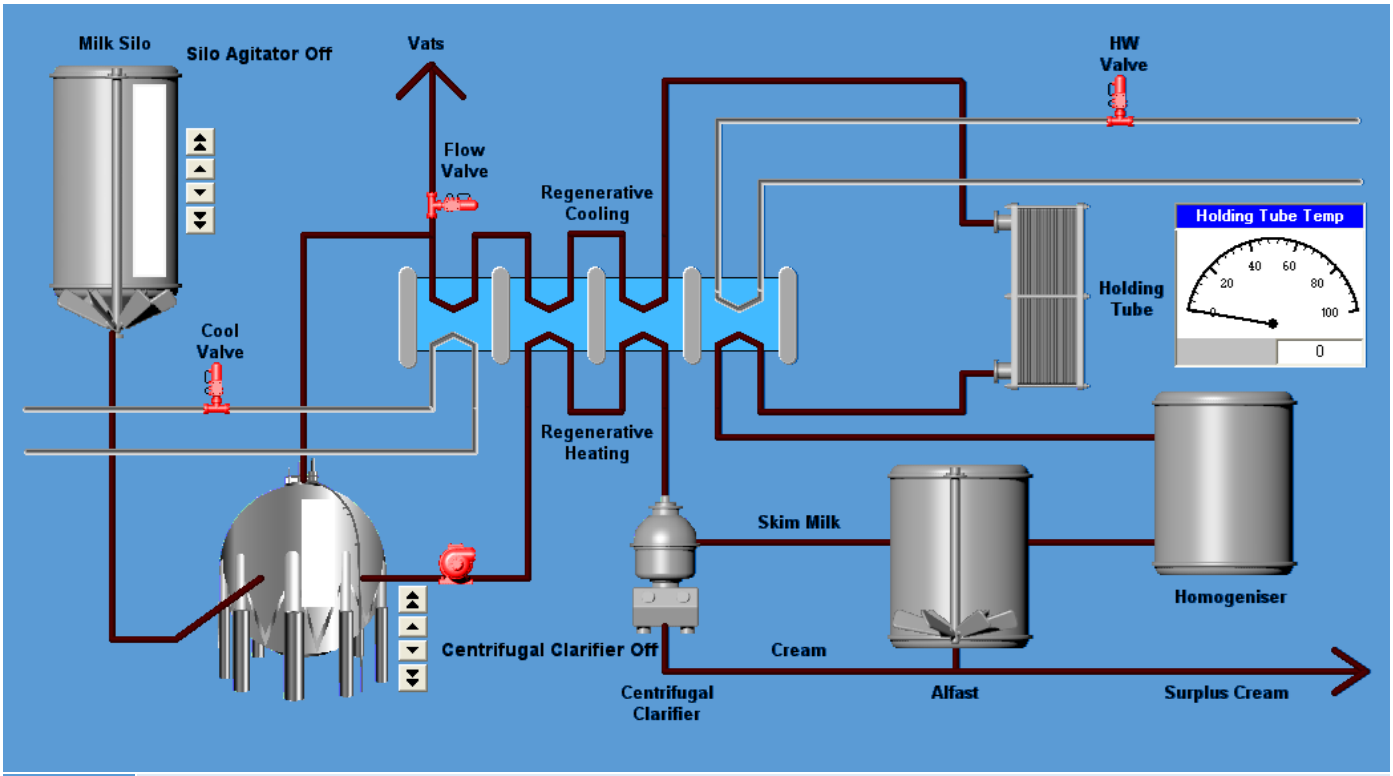
iv. Go to the Appearance (Tag Association) tab and choose the Value property from the left pane. Insert the tag "TIC_Hold_PV" into the right pane. Click OK.

v. The following message will appear because the **Property Type** is **REAL**. You can display compatible property types by clicking on the **List Property Type** button in the Tag Association tab.

vi. The TIC_Hold_PV is an INTEGER which is why the message is displayed. Click **Yes** to continue.



2 **Your page should now look something like this. Save the page.**



3	**Compile and run the project and test your modifications.** i. Click on the Run Project button in the Citect Project Editor. ii. If the compiler reports any errors select GoTo to display the dialog where an error exists and fixes the problem. iii. When you have corrected all errors click on **Run Project** again.
4	**Shutdown the project.**

Chapter Seven “Commands and Controls”

7.1. Introduction

Commands and controls allow operators to interact with the Vijeo Citect runtime system. There are three types of commands and controls:

1) ***Slider controls*** that an operator can use to change the values of analog variables.
2) ***Touch commands*** that an operator can issue by clicking on an object with the mouse.

7.2. Slider Controls

7.2.1. Overview

Slider controls allow an operator to change the value of an analog variable by dragging an object on the graphics screen. The position of a slider will also automatically update when the value it controls changes.

Sliders can operate in the following ways:

- Horizontally
- Vertically
- Rotationally

By filling in the Slider properties, most objects can be configured as a slider.

- **Exercise 7-1**

Add a slider control on the Pasteuriser page to adjust the Milk Silo level.

Step	Action
1	**Remove the genie you pasted in the last chapter to change the Milk Silo level.**
2	**Paste a new symbol on to your Pasteuriser page to be the slider.**

i. Select the Paste Symbol tool  and choose a pointer from the thumbs library that will suit a vertical slider. For example, knob_vert_red  would be a suitable choice.

Note: Keep the pointer Linked to the library, so then you can choose a different pointer later without having to re-type the object properties.

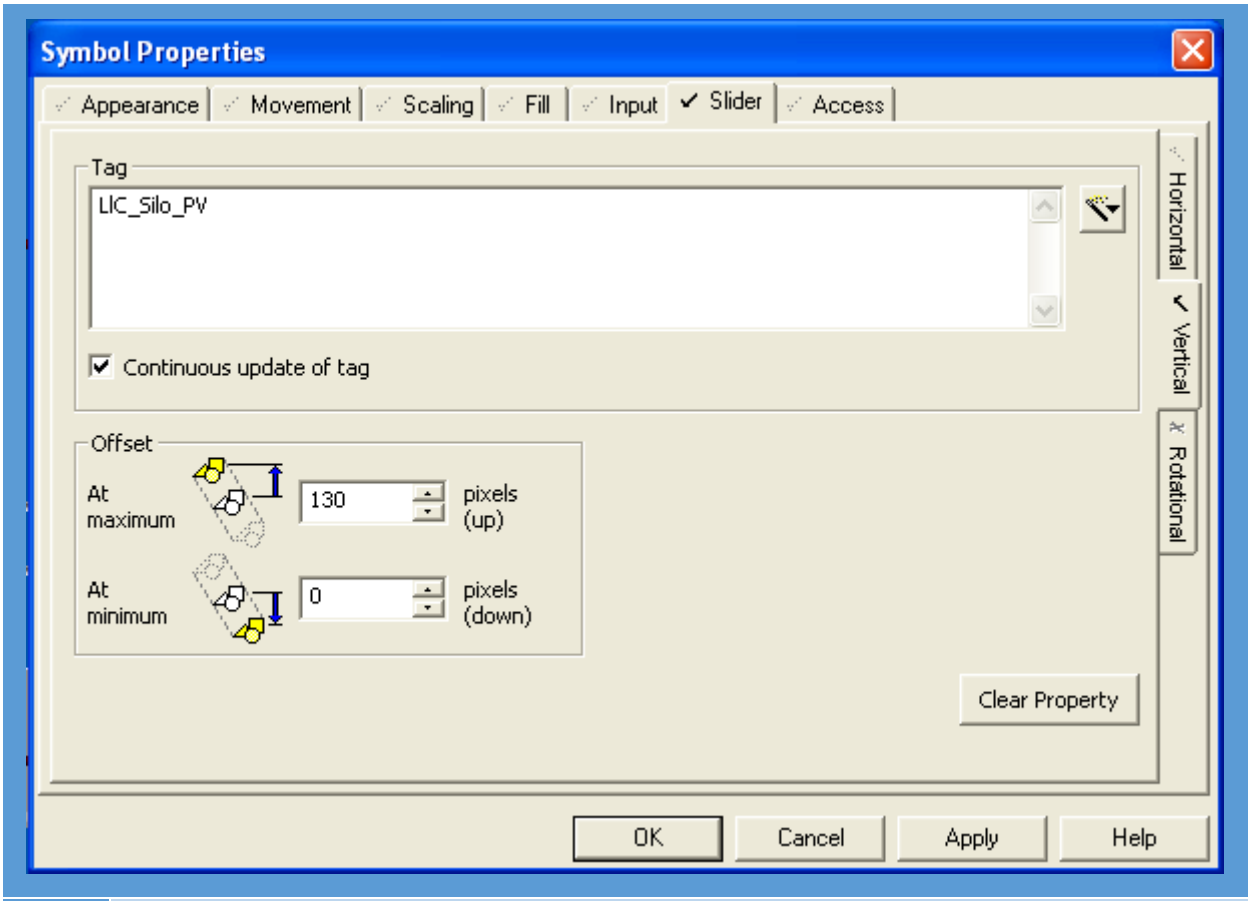
Hints & Tips:

To align the pointer with the base of the fill rectangle, use the Pixel Viewer, function key F10.

3

Add properties to the symbol to turn it into a slider

i. Go to the Slider (Vertical) properties and fill in the dialog as shown:

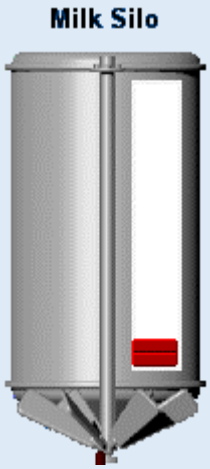


Hints & Tips:

Ideally you want the slider to line up with the top of the Bar Graph at its maximum value. Click on the rectangle object and check the size of the object in the bottom right of the Citect Graphics Builder 654 , 390 to get the offset. Put the y-coordinate as the Offset: At maximum value.

ii. Go to the Access (General) properties of the symbol and click on Help. Read the information on [Identification] Tool Tip. Then add the following tool tip: "Change the MILK SILO level with this slider"

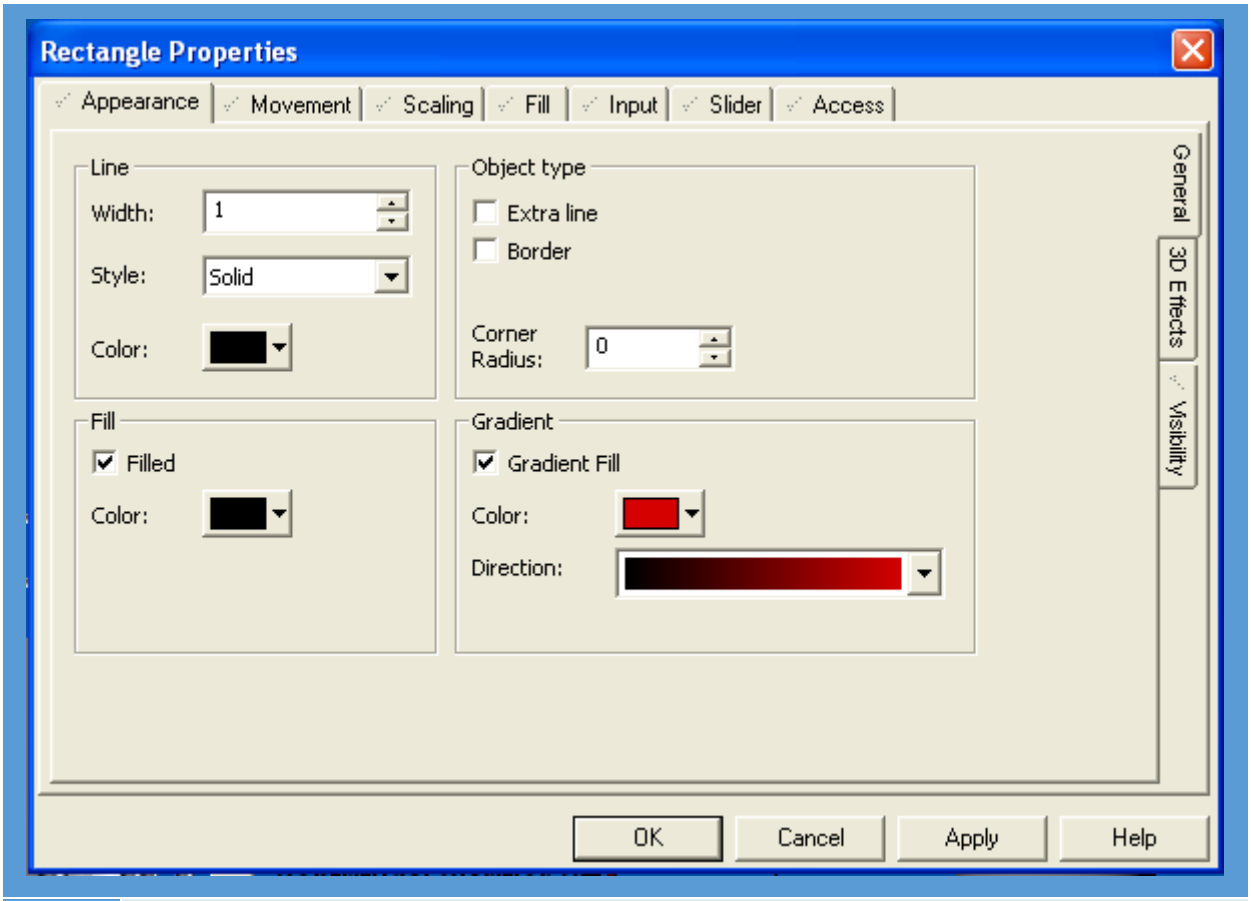
iii. Move the pointer to the bottom of the bar graph next the Milk Silo.



iv. Place A pointer on the Balance Tank bar graph.

Add another bar graph and slider to the Active X control displaying the Holding Tube Temperature.

i. Add Gradient Fill to the bar graph to indicate the temperature change from cold to hot.



5	**Save your page.**
6	**Compile and run the project and test your modifications.**
7	**Shutdown the project.**

7.3. Touch Commands

7.3.1. Overview

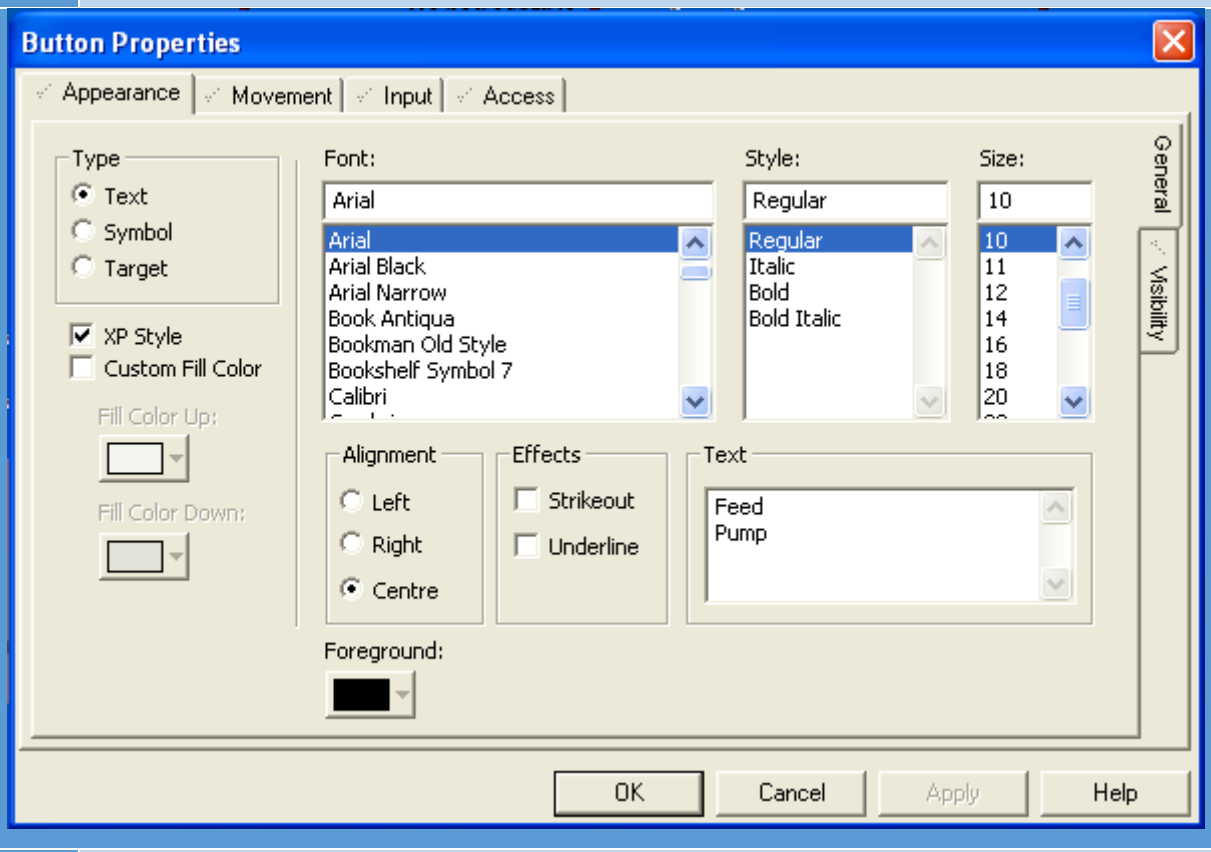
The operator can execute a command (or series of commands) by clicking on an object with the mouse.

You can define several commands for an object - one command to execute when the mouse button is pressed down, another for when it is released, and another to operate continuously while an operator holds the button down.

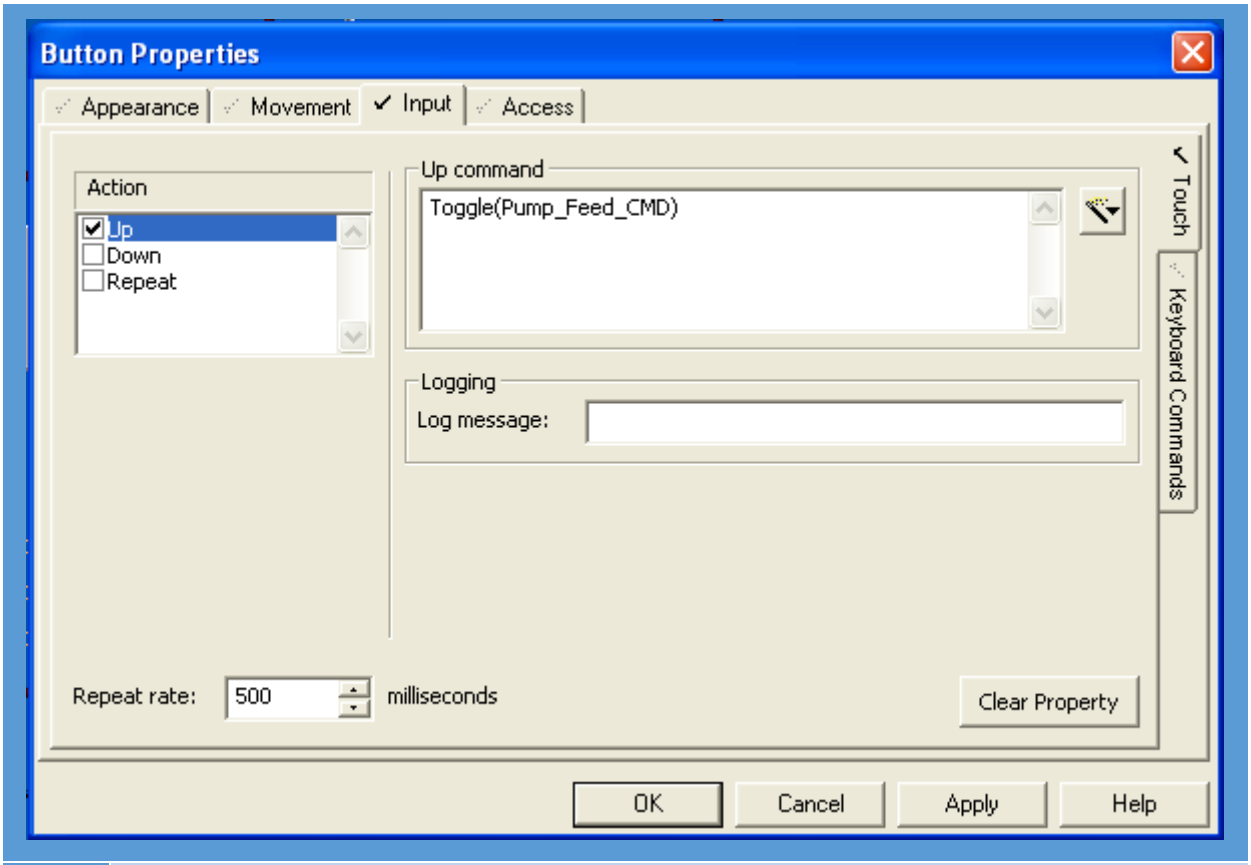
To define a touch command you need to fill in the Input (Touch) properties for an object.

- **Exercise 7-2**

Step	Action
1	**Draw a button on your Pasteuriser page.** i. Select the Button tool and draw a button near the Feed Pump. ii. In the Appearance (General) properties type the Text you want displayed on the button face, and choose one of the Vijeo Citect fonts.



	iii. Go to the Input (Touch) properties for the button and fill in the dialog as shown:



	Hints & Tips: You can use the Expression Wizard  to paste the function Toggle and then the tag name in to the Up command box.
2	**Save your page.**
3	**Compile and run the project and test your modifications.**
4	**Shutdown the project.**

Chapter Eight “Genies”

8.1. Introduction

Usually each graphical object on a graphics page is configured individually. With a genie, you can combine several related objects into a group and store the group in a genie library (similar to a symbol library). The genie can then be used as if it was a single object (pasted, moved, resized, etc.) and the elements of the genie are then configured collectively.

All types of graphic objects and their configuration data can be stored with the genie. For example, you can define a genie for a start/stop controller (with a start button, a stop button and an indication lamp), and use the same genie for all equipment (pumps, conveyors, etc.) that use that type of controller. When you use the genie you only need to specify the information that is unique to that particular pump or conveyor (i.e. the variable tag).

8.2. Pasting a Genie from the Libraries

8.2.1. Overview

Genies are usually defined without specific variable tags so that they can be reused, both in the same project and in different projects. When you paste a genie onto a page, a dialog box will request the entry of one or more variable tags and comments or other data, to control the display and actions of the genie object at runtime.

8.2.2. How to to paste a genie on to a graphics page:

In the Citect Graphics Builder, Click on the Paste Genie tool 

or

Select the menu edit » Paste Genie...

- **Exercise 8-1**

Paste a genie from a library in the Include project onto a test page called Utility.

Step	Action

1 **Create a new Graphics page in your project called Utility.**

i. Use the following template settings:

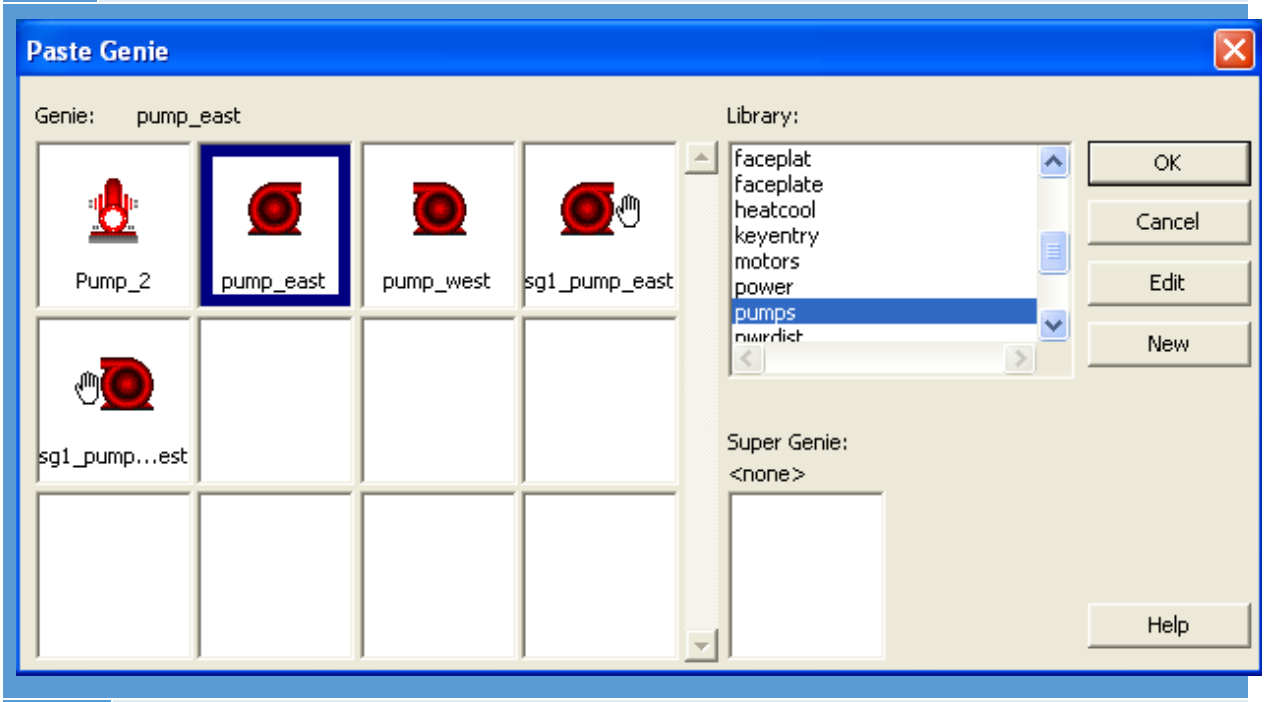
Style	**xp_style**
Linked	√
Resolution	XGA
Template	Normal

ii. Save your page, naming it Utility.

2 **Paste a genie onto the Utility page to display the Feed Pump status.**

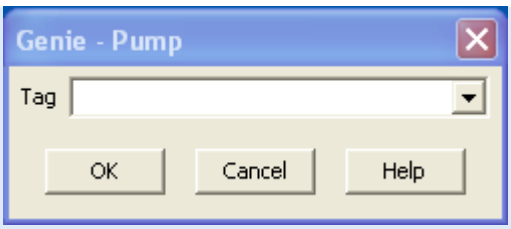
i. Click on the Paste Genie tool  .

ii. From the pumps library select the **pump_east** genie.



iii. Click on OK.

iv. In the dialog box, select the Pump_Feed_CMD variable tag.

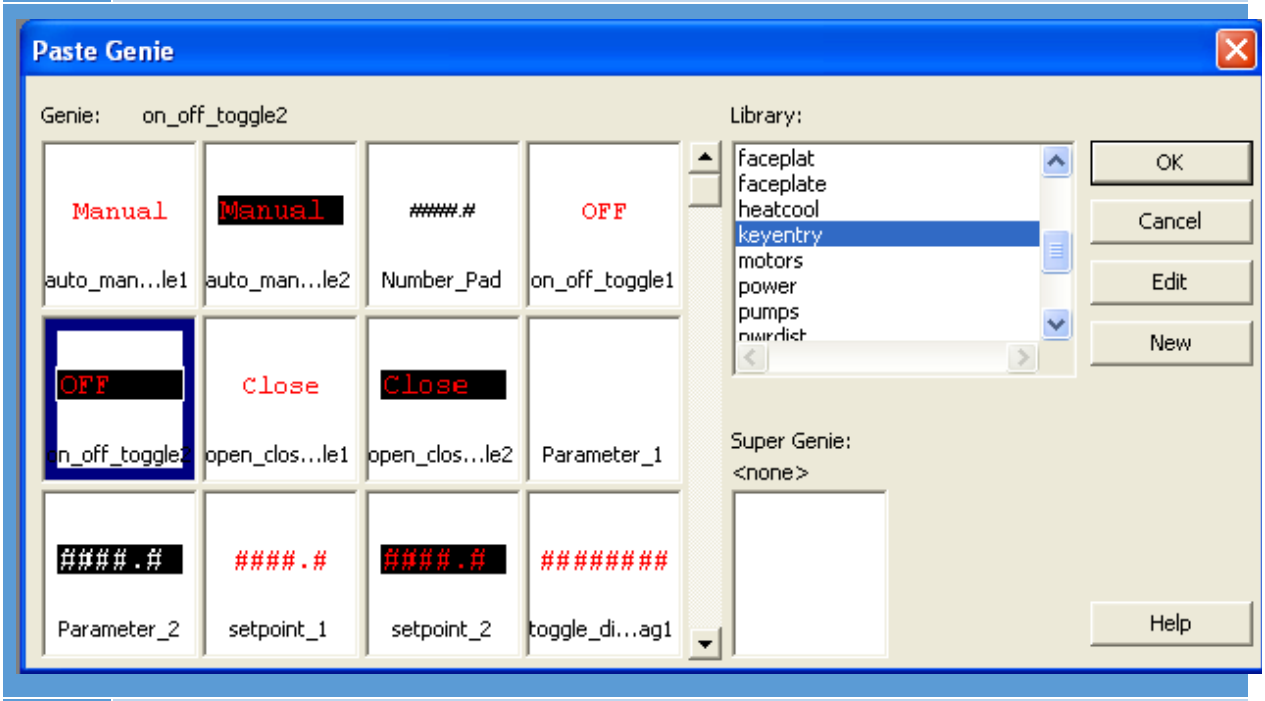


v. Click on OK.

3 **Paste a genie to toggle the Pump_Feed_CMD variable tag and place it under the pump genie on the Utility page.**

i. Click on the Paste Genie tool  .

ii. From the keyentry library select the on_off_toggle2 genie.



iii. Click on OK.

iv. Save the page.

4 **Compile and run the project and test your modifications.**

5 **Shutdown the project.**

8.3. Creating a New Genie

8.3.1. Overview

In the previous exercise, we pasted two genies to accomplish what could easily have been done with one genie. It is easy to create a single genie by pasting existing genies or objects on to a new genie and then saving this in a library as part of a project.

8.3.2. How to create a new genie:

In the Citect Graphics Builder, click on the New  button on the toolbar

or

Select the menu file» New ...

8.3.3. Modifying a Genie

You can at any time, modify a genie that you have created. If the genie has been pasted onto pages in the project, then ensure that all pages in the project are saved and closed before modifying the genie.

After making changes, save the Genie and select the menu Tools» Update Pages to refresh all pasted copies of the genie in the project.

8.3.4. Genie Syntax

Text or variable tags anywhere in a genie can be replaced with a substitution name using the syntax **%Name%**. When the genie is pasted onto a graphics page, the user will be prompted for a variable tag name or text to substitute for **%Name%** in that particular instance of the genie. The name supplied can be edited at any time by double clicking on the genie. For more information open the Cilect Help Topics and look under the heading Defining Substitutions or Genies.

Example -

The expression,

M_TagA = My_TagB + My_TagB * 5/100;

could be replaced with,

%tag1 % = %tag2% + %tag2% * 5/100:

and when the genie is pasted onto a page, the user will be prompted to supply the appropriate tag name for each different substitution name defined in the genie as shown:

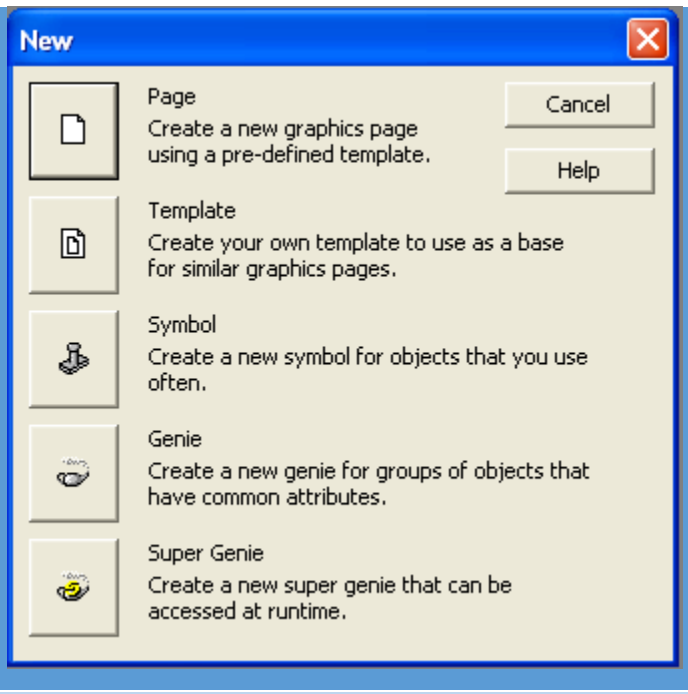
Note:

In the above example, %tag2% is used twice, however the user is only prompted once for the substitution. Every time a copy of the genie is pasted onto a graphics page, all instances of %tag2% within that copy will be replaced with the name supplied.

- **Exercise 8-2**

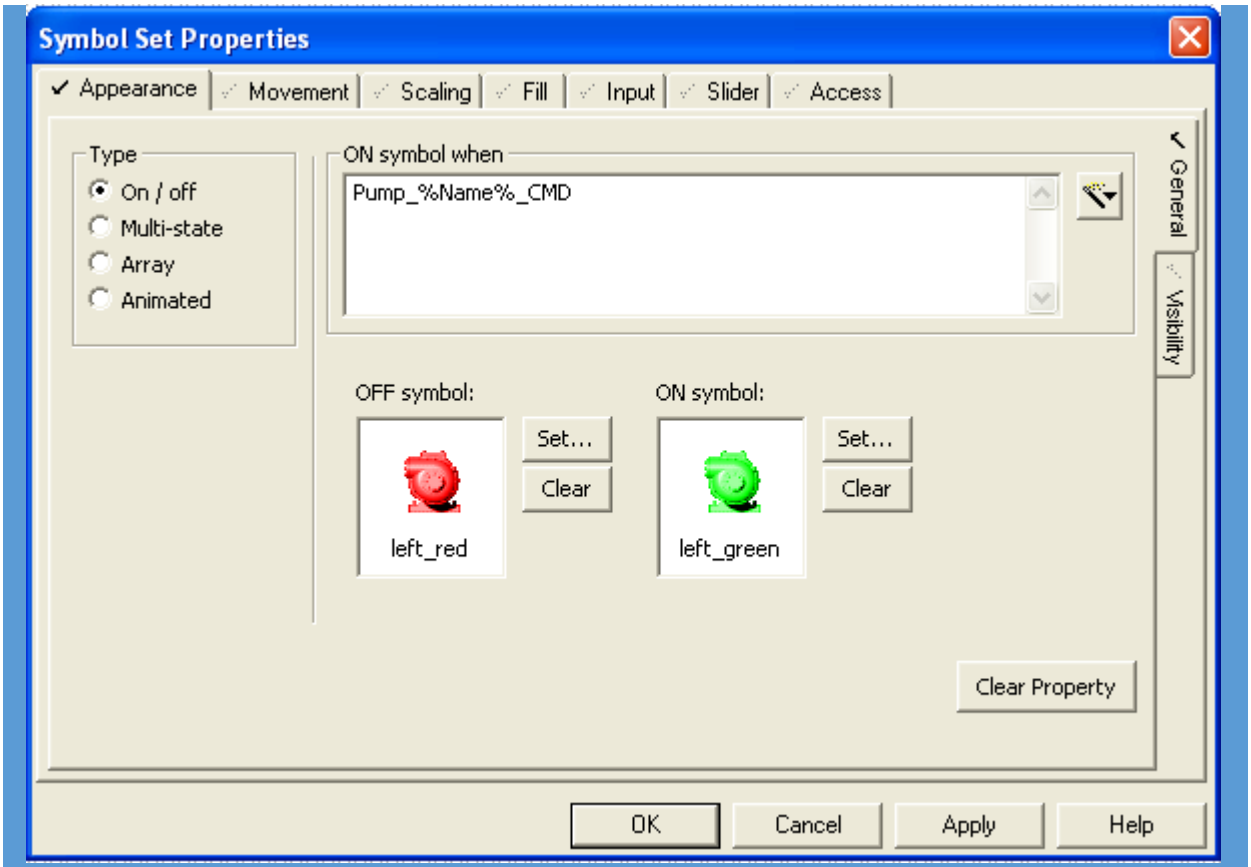
Create two new genies for your Pasteuriser page. The first is to control the Feed Pump and the second will be used in a number of places, for the keyboard input of values into variable tags.

Step	Action
1	**Create a new genie with an animated pump symbol.** i. Open the Citect Graphics Builder, click on the New  button on the toolbar. The following dialog will be displayed:



ii. Click the Genie button. A blank page will be displayed, with a **Genie Locator Point** (this is reference point when the genie is pasted onto a graphics page).

iii. Paste a pump to display the status near the Locator point, using the Symbol Set tool.



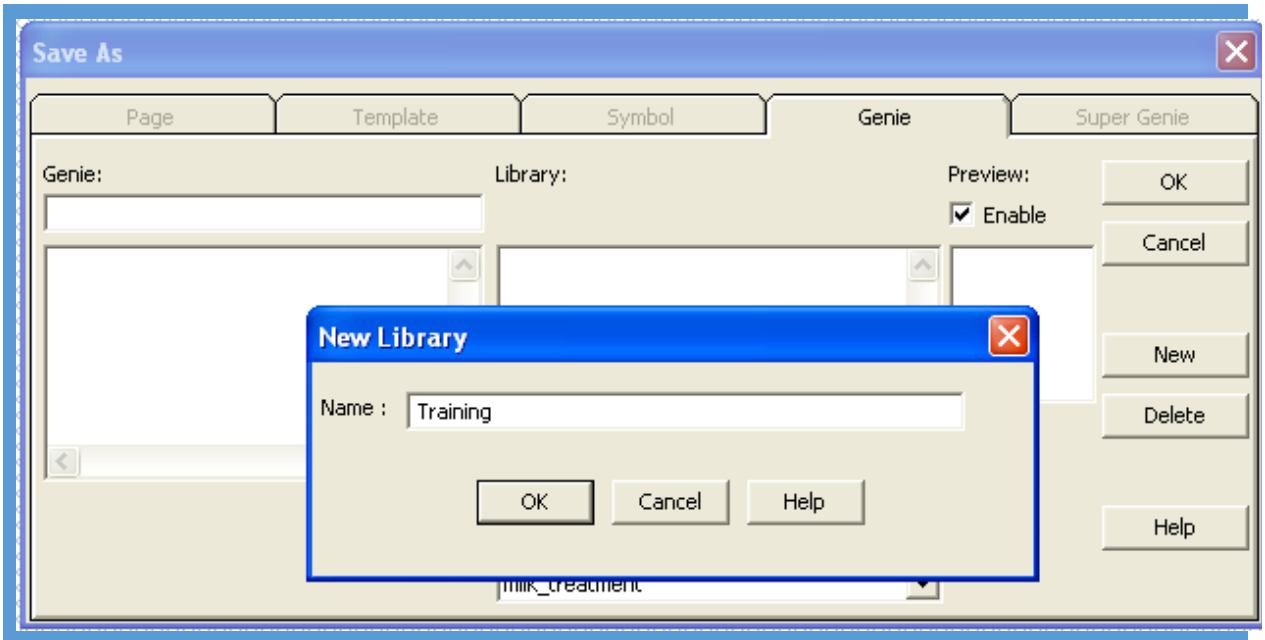
Select the On / off type of Appearance (General) properties, and select the following symbols:

OFF	pumps_base_small.left_red
ON	pumps_base_small.left_green

In the On symbol when field type Pump_%Name%_CMD.

iv. Click on OK.

v. Click on Save and fill in the dialog:



Click on New to create a new library in which to save your genie. Type a library name (training) and click on OK. Type a name (PumpControl) for your genie and click on OK.

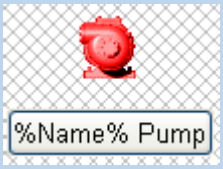
2 **Create a button to control the pump.**

i. Select the Button tool and draw a button below the pump. Set the object properties as follows:

Appearance (General) Text	%Name% Pump
Input (Touch) Up Command	Toggle (Pump_%Name%_CMD)

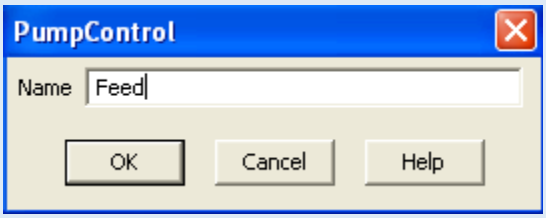
3 **Resave the genie with the button.**

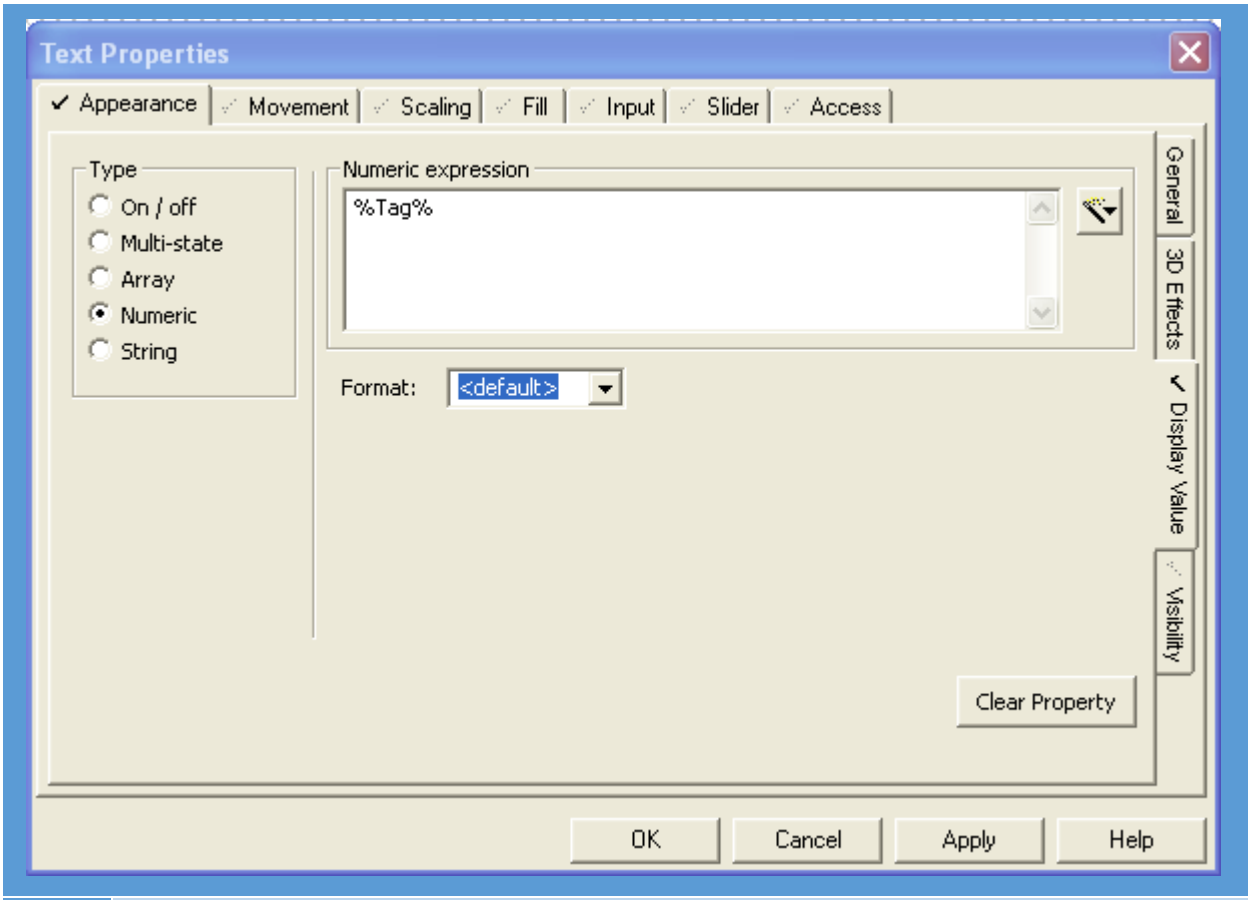
i. Position the button and symbol so that they look similar to this:



ii. Save your genie again.

iii. Select file» Close from the menus to close your genie.

4	**Use the Paste Genie tool  to paste the PumpControl genie onto your Pasteuriser page.** i. Specify Feed as the Name.  ii. Replace the Pump symbol set and toggle button with the genie.
5	**Create a genie to display the value of a tag and accept keyboard input to change the value of the tag.** i. Create a new blank genie as before. ii. Click on the Numeric tool ## and click near the Genie Locator Point to place a numeric object. iii. Insert %Tag% into the Numeric expression field, as shown:



	iv. Click on OK. v. Click on Save and save the new genie with the name ChangeValue to the training library. vi. Select file » Close from the menu.
6	**Repeat the previous step for each of the following variable tags and place each genie next to the associated plant item on the page:** LIC_Balance_PV TIC P1 PV TIC_P2_PV TIC_P3_PV TIC_P4_PV
7	**Compile and run the project and test your modifications.**

8	**Shutdown the project.**

Chapter Nine “Popup Pages and Genies”

9.1. Introduction

Popup pages and genies are dynamic pages to that you can use to pass information when the page displays in the runtime system. You can use them for pop-up type controllers for a process or a single piece of plant floor equipment. The same page can then be re-used with different sets of tags.

For example you could configure a graphics page with several pumps, and use a genie pop-up controller for each of the pumps on the page. In most cases where you use a popup page, you use it in association with a genie.

9.2. Creating Popup Pages

9.2.1. Creating a Popup Page

First create and save a new blank page. This page can be attached to a genie or called directly from a page. In either case, a super genie function will need to be used to open the popup page at runtime.

9.2.2. Genie Functions

There are a number of functions that can be used to call and modify a genie. These functions can be used from pages or genies to open and modify a genie page at runtime and many of them can contain a list of tag names to be passed to the super genie.

- **Example -**

The function,

AssPopUp("sgPage", "Tag1 ", "Tag2", "Tag3")

Calls the super genie page sgPage and associates three variable tags with the page. The order of the tags here will be the order in which they are referenced in the super genie page using super genie syntax.

9.2.3. Genie Syntax

Variable tag names can be substituted at runtime by usmg a senes of substitution names with the syntax:

?type number?

where number is the position of the tag name in the list provided in the super genie function that was called to open the super genie page. It is not strictly necessary to use type in the supergenie syntax. However, if you do wish to use type it is the data type in the variable tag (e.g. string, int, real or digital).

- **Example-**

In the previous example of AssPopUp(), if the tags were a string, integer and digital respectively, then they could be referenced anywhere in the supergenie as:

?string 1? Strings must have the type defined

?int 2? Or ?2?

?digitaI 3? or ?3?

9.2.4. Structured Tag Names

If a super genie function such as AssPopUp() is used in a genie, then part or all of the tag names passed can be specified as genie substitutions. The advantage is that if variable tags in the project all follow a standard naming convention, then only part of the name needs to be passed to the genie.

- **Example-**

The following function associates the two tags Pump1_ Valve1 and Pump1_ Valve2 with the super genie page sgPage:

AssPopUp("sgPage","Pump1_ Valve1 ","Pump1_ Valve2");

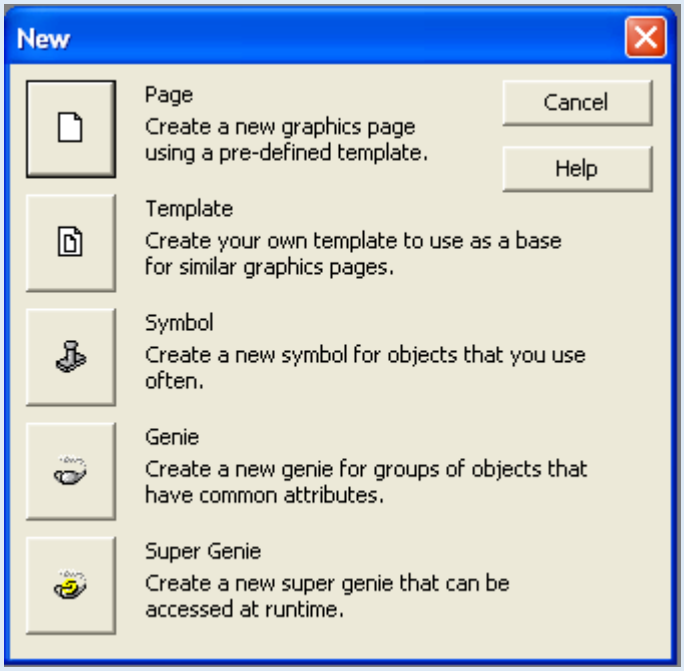
This function could be replaced with,

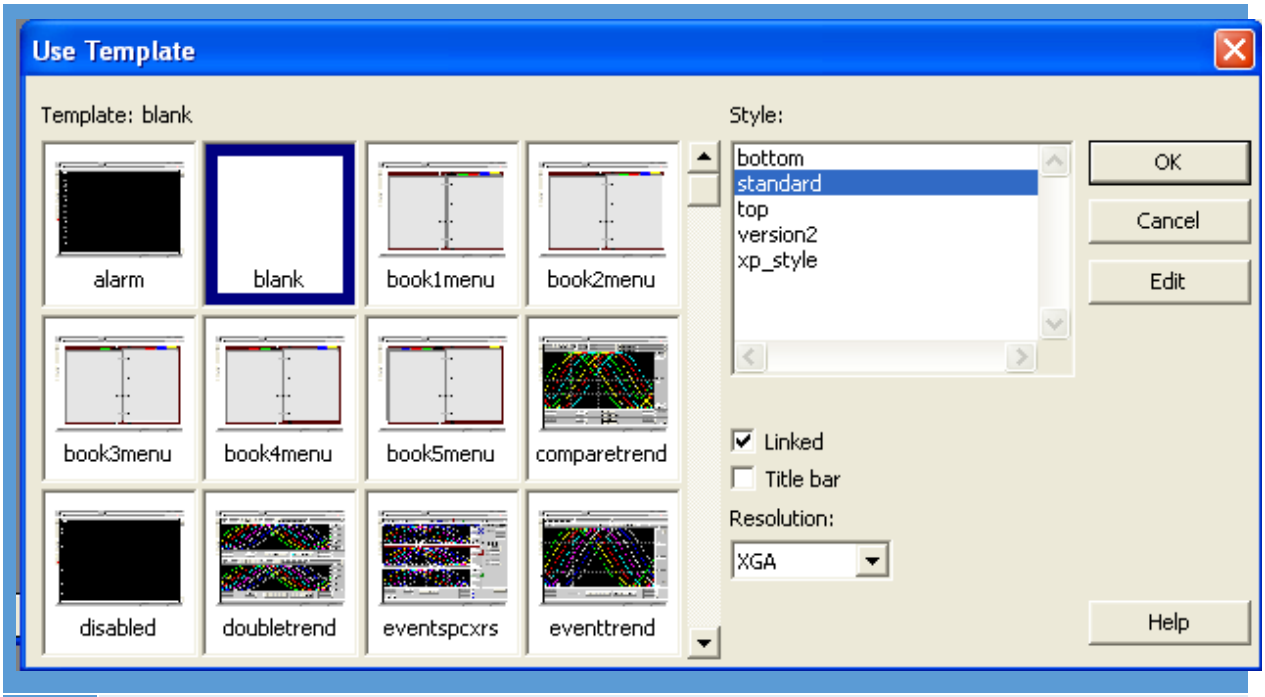
AssPopUp("sgPage" ,"%Pump%_ Valve1" ,"%Pump%_ Valve2");

and when the genie is pasted onto a graphics page, Vijeo Citect will only prompt for the pump name, which will be Pump1. This genie will now work with all pumps that have a variable tag naming scheme which follows that of Pump1.

- **Exercise 9-1**

We will now create a popup page to open and close the valves on the Pasteuriser page.

Step	Action
1	**Create a new popup page as shown called !Valve which can be used to open and close any valve.** i. Click on the New  button in the Citect Graphics Builder.
2	**Click the Page button when the following dialog is displayed:**  i. Choose the blank page template when this form is displayed.



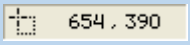
ii. Draw the three buttons near the top left hand corner of the page with the following properties:

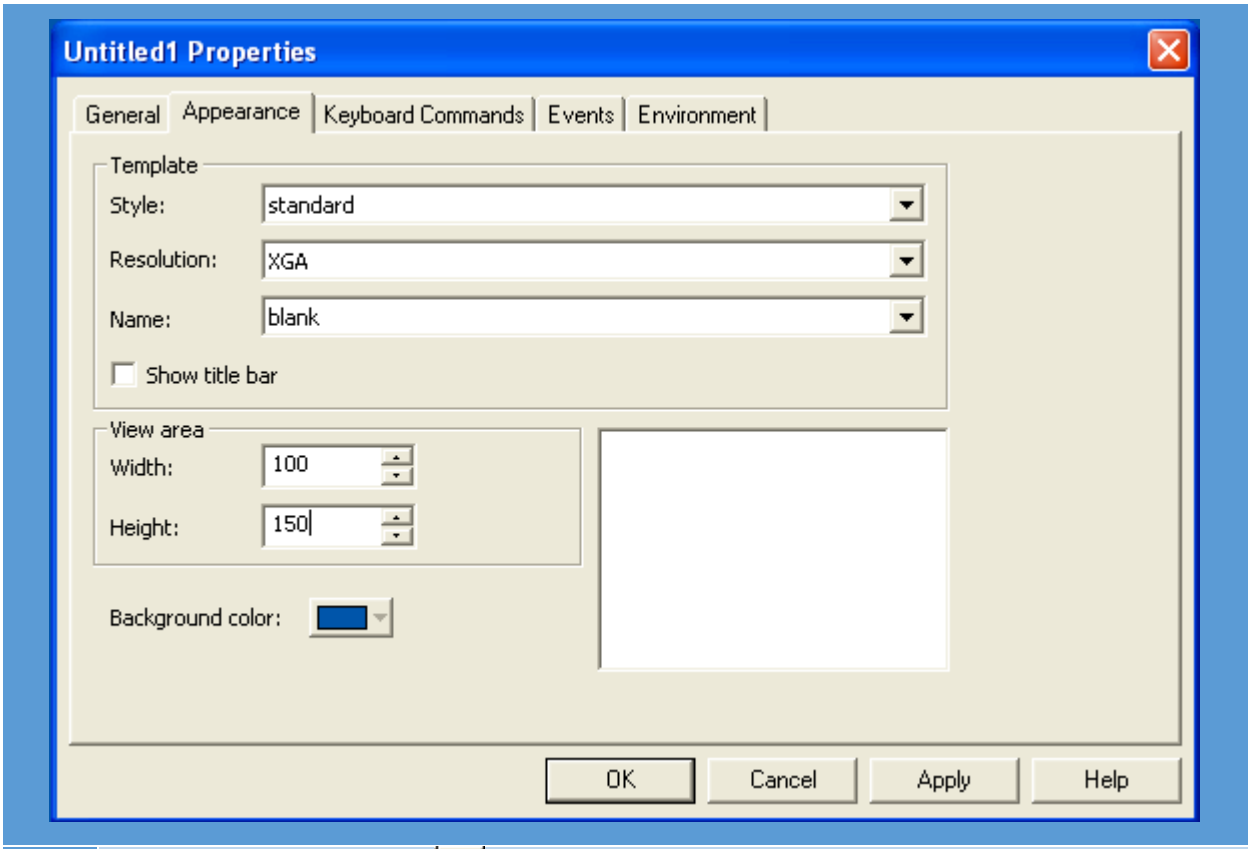
Appearance (General) Text/Symbol	Input (Touch) UP Command
OPEN	?1?=1
CLOSE	?1?=0
Close window	WinFree ()

Note:

The above popup has just one substitution (enclosed within question marks ?number?) for a digital tag, which is used in more than one place in the super genie.

iii. Draw the two light symbols using the Symbol Set tool  . Set the properties as follows:

	Object	ON symbol when	Symbols	
	OPEN light	?1?=1	OFF	lights_square_medium.grey
			ON	lights_square_medium.green
	CLOSE light	?1?=0	OFF	lights_square_medium.grey
			ON	lights_square_medium.green
3	**Continue with the configuration of the popup.** i. Place the pointer at the bottom right hand corner of the buttons. This is to indicate the size of the page you will need for your popup page. ii. The status bar at the bottom of the screen will display the pointer position from the top left hand corner of the page.  iii. Go to File» Properties» Appearance tab and enter the size of the rectangle into the Width and Height. This will define the size of the popup page. If you wish you may also change the Background colour of the page here.			



iv. Click on the Save button and save the page with the name !Valve.

Note:

Placing an exclamation mark (!) at the start of page name means that the page cannot be chosen from the Select Page dialog or the Page menu at runtime.

9.3. Calling Popup Pages from Symbols

- **Exercise 9-2**

Use the popup page you just created and attach it to an existing symbol.

Step	Action

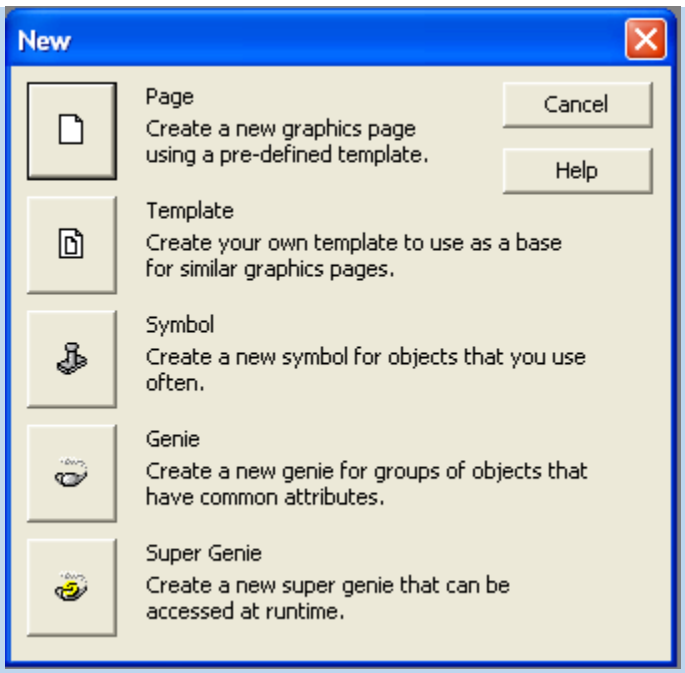
1	**Call the page !Valve from an existing symbol.** i. Go to the Pasteuriser page and open the **Coolant Valve** symbol set. ii. Add the following properties to the symbol set. **Input (Touch) Up Command** : AssWin("!Valve",145,330,1+8+512,"VALVE_Cool_CMD") **Note:** The tag name in this function must be inselied in quotes “ " If it is not between quotes then it is the value of the tag that is passed to the function rather than the tag name.
2	**Save the page.**
3	**Compile and run the project and test your modifications.**

9.4. Calling Popup Pages from Genies

- **Exercise 7-3**

Create a genie to call the popup page.

Step	Action
1	**Create a genie called ValveControl with a button to call the page !Valve.** i. Click on the New  button in the Citect Graphics Builder. ii. Click the Genie button when the following dialog is displayed:



A blank page will be displayed, with a Genie Locator Point:

iii. Draw a button near the Locator point with the following properties:

Text	**%Valve%^nValve**
Input (Touch) Up Command	AssWin("!Valve",%OriginX%,%OriginY%,1+8+512,"VALVE_%Valve%_CMD")

iv. Click on Save and save the genie with the name ValveControl in the training library of your Milk_Treatment project.

2 **Paste the ValveControl genie onto your Pasteuriser page next to the valves.**

i. Fill in the genie fields as follows:

Object	Origin X	Origin Y	Valve
Coolant Valve	**145**	**330**	**Cool**
Hot Water Valve	**660**	**160**	**HW**
Flow Diversion Valve	**360**	**160**	**Flow**

	ii. You will need to change the Origin X and Origin Y coordinates for your own page.
3	**Compile and run the project and test your modifications.**
4	**Shutdown the project.**

Chapter Ten “Alarms”

10.1. Introduction

Protection of valuable plant equipment is a central feature of your VijeoCitect system. The Vijeo Citect alarm facility constantly monitors equipment and alerts operators of any equipment fault or alam1 condition.

Vijeo Citect supports two types of alarms:

- Hardware Alarms

Vijeo Citect continually runs diagnostic routines to check all peripheral equipment, such as l/O Devices. All faults are reported automatically to the operator. This facility is fully integrated within Vijeo Citect – no configuration is necessary.

- Configured Alarms

Unlike hardware alarms, you must configure the alarms that report fault conditions in your plant (for example, when a tank level is too high or when a motor overheats).

10.2. Configuring Alarms

10.2.1. Overview

Alarms are entered into a database using forms, in a similar fashion to variable tags. Each type of alarm has different triggers and parameters and so has a separate configuration form with different options.

10.2.2. Digital Alarms

These depend upon the change of state of one or two digital tags. If two tags are specified, then both must change for the alam1 to be triggered.

10.2.3. Time Stamped Alarms

Time stamped alam1s are similar to digital alarms - the alarm is triggered by a state change in a digital tag. However, time stamped alarms have a time source, which supplies the exact time that alarm was triggered. The timer is typically a time stamp read from an I/O Device.

10.2.4. Analog Alarms

Analog alarms are triggered when an analog variable changes beyond one or more specified limits. Each alarm may be configured as any combination of the following types:

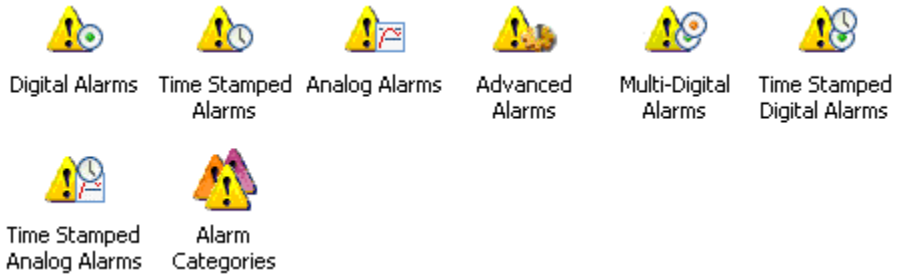
1) High and High High alarm
2) Low and Low Low alarm
3) Deviation alarm
4) Rate of Change alarm

10.2.5. Hardware Alarms

Hardware alarms are separate from the user alarm system. They display when problems are detected in the operation of Vijeo Citect or when a connection to an I/O Device fails. All hardware alarms are pre-configured in every Vijeo Citect system.

10.2.6. How to add a new alarm:

Open the Citect Explorer, select a project and open the Alarms folder, then double click on the icon for the type of alarm you wish to add:

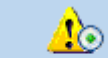


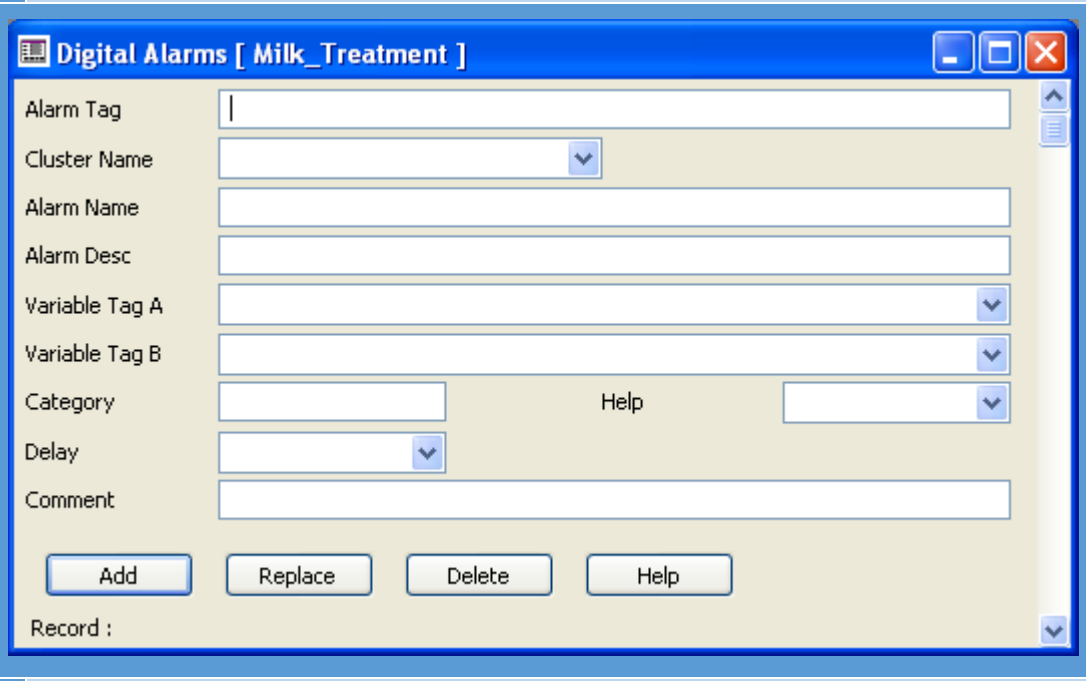
Or

Open the Citect Project Editor, select the Alarms menu and then choose the appropriate menu for type of alarm to be added.

- **Exercise 10-1**

Configure the alarms that will be used in your Milk_Treatment project.

Step	Action
1	**Configure one digital alarm for when the Silo Agitator is stopped and another digital alarm that is triggered if the Centrifugal Clarifier is on but the Feed Pump is off.** i. Open the Citect Explorer, select the Milk_Treatment project and open the Alarms folder, then double click on the Digital Alarms icon or open the Citect Project Editor, select the Alarms» Qigital Alarms menu item.  Digital Alarms ii. The following form will display:



iii. Add the following two digital alarms to the database:

Alarm Tag	Silo	Clarifier
Alarm Name	Silo Agitator OFF	Process Violation
Alarm Desc	Silo Agitator Stopped	Clarifier RUNNiNG & Feed Pump OFF

Var Tag A	NOT Agitator Silo V	Centrifuge_Clar_V
Var Tag B		NOT Pump_Feed_CMD

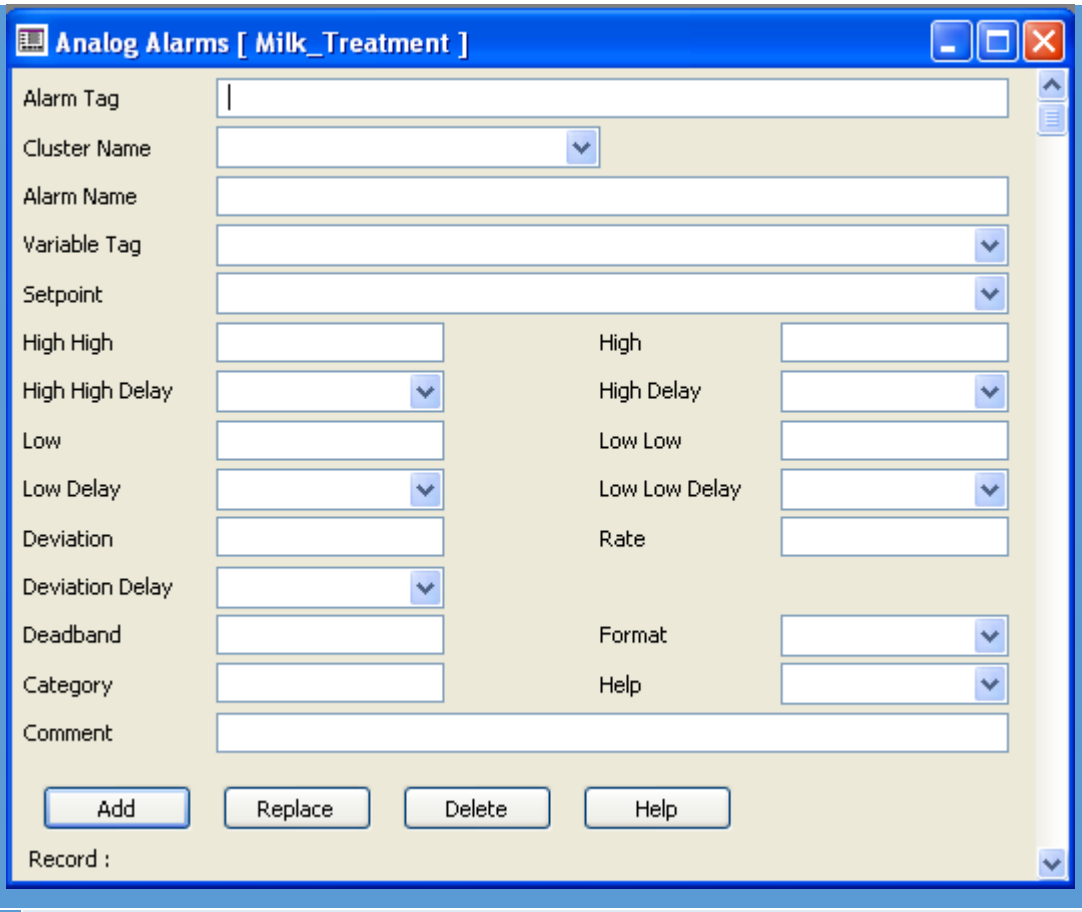
Hints & Tips:

Digital alarms are triggered by logical ON I OFF conditions. The logical operator NOT inverts the logic of any expression it precedes.

i. Open the Citect Explorer, select the Milk_Treatment project and open the Alarms folder, then double click on the Analog Alarms icon or open the Citect Project Editor, select the Alarms >> Analog Alarms menu item.

Analog Alarms

ii. The following form will display:



iii. Add the following analog alarm to the database:

Alarm Tag	**HTA**
Alarm Name	Holding Tube Alarm
Variable Tag	TIC_Hold_PV
Setpoint	73
High High	85
High	80
Low	65
Low Low	60
Deviation	3
Deadband	2
Format	###

Chapter Eleven “Trends”

11.1. Introduction

A visual representation of past and cun-ent activity improves understanding of the plant's performance. With a trend graph, you can display the values of a variable (or process) in graphical format. As these values change over time, the graph moves across the page so that the latest values are always displayed. You can also scroll back through historical data to display past values of the variable (or process). Historical data collection continues even when the display is not active. You can switch between pages without affecting trend graphs.

You can trend any single variable or Cicode expression. You can display any number of trends on the screen simultaneously, and display the trends of up to eight variables in any trend window. Vijeo Citect provides three types of trends: Periodic, Event, and Periodic Event.

11.2. Trend Tags

11.2.1. Overview

Trends in Vijeo Citect are added by creating trend tags. Each trend tag will have one or more separate files in which trend data is stored. Vijeo Citect will continuously store trend data, regardless of whether it is displayed on a graphics page.

The types of trends available are:

Periodic - samples are taken on a time basis.

Event - one sample is taken each time a trigger action switches on.

Periodic Event - samples are taken on a time basis only while a trigger value is on.

To trend a variable tag, create a trend tag for it and define one or more history files in which to log the value of the trend tag. You then display the trend by creating a trend page and assigning a pen to the trend tag to display it in the trend window.

11.2.2. How to To define a Trend Tag:

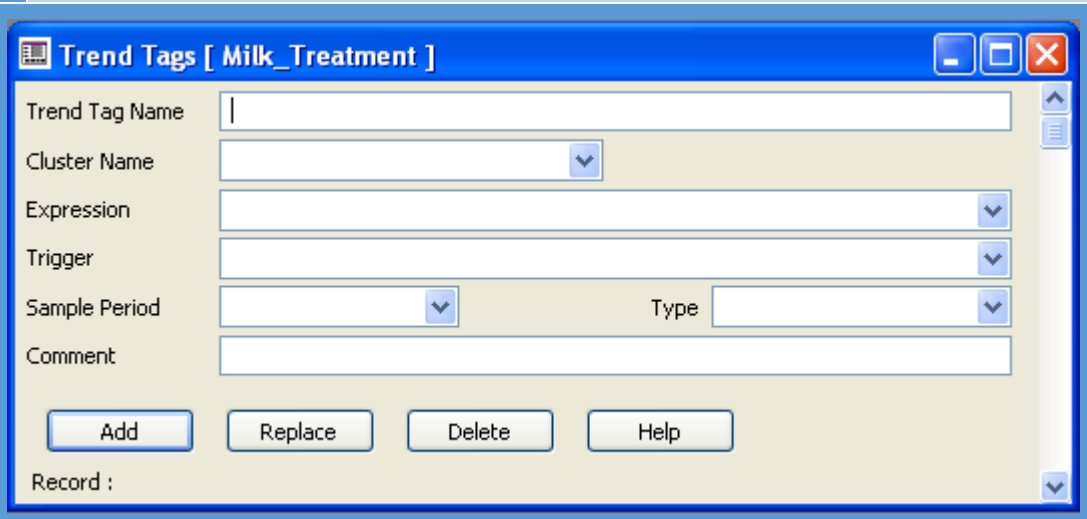
In the Citect Explorer, select a project, then open the Tags folder and double click on the Trend Tags icon:

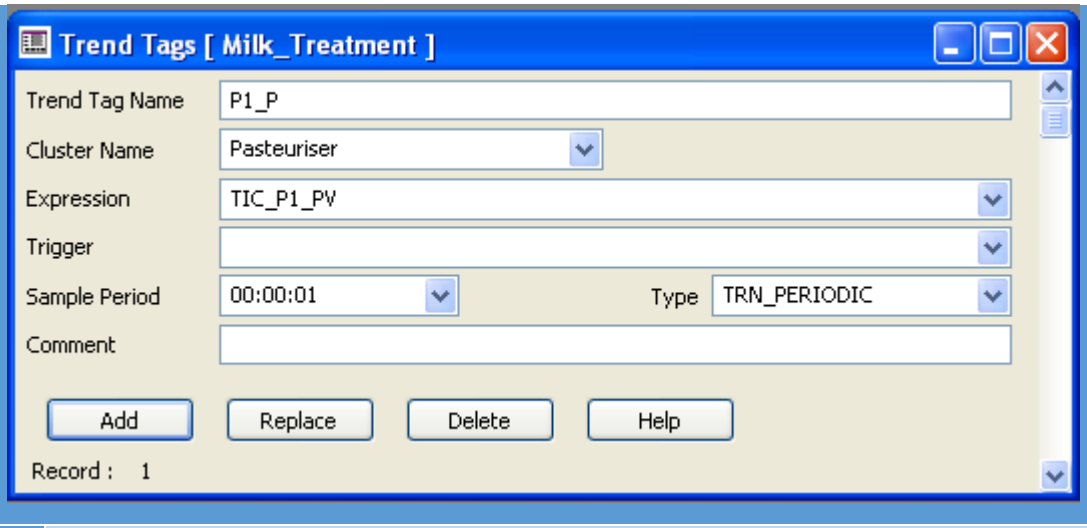
or

In Citect Project Editor, select the menu Tags» Trend Tags.

- **Exercise 11-1**

Trend some of the tags in the Milk_Treatment project.

Step	Action
1	**Create five periodic trends.** i. Open the Citect Project Editor and choose lags» Trend Tags from the menu. ii. The following form will appear.
	
	iii. Fill in the form as shown below to create a periodic trend tag that will trend the value of the tag TIC_P1_PV.



iv. Add a new record for each of the trend tags defined in the table below.

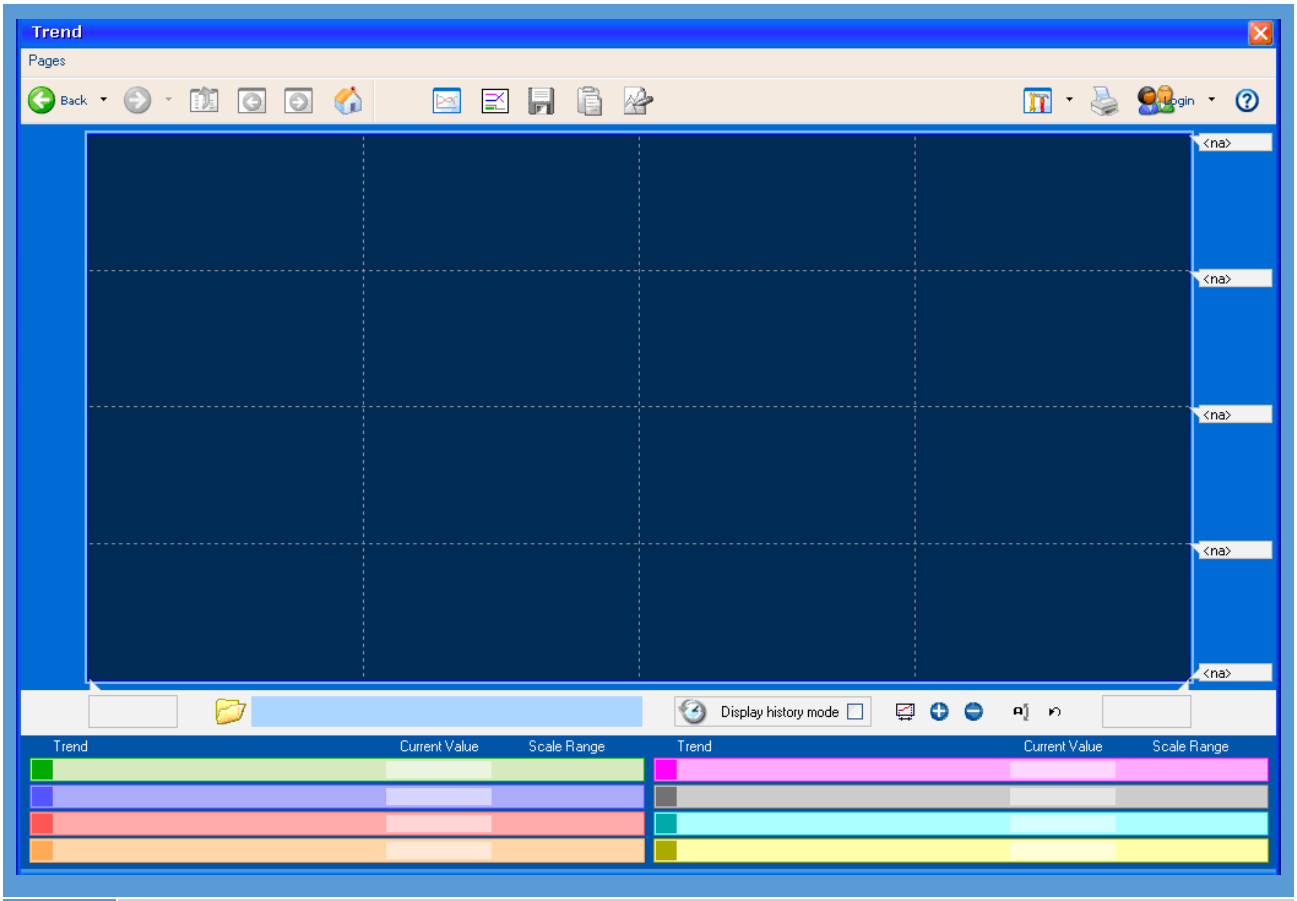
Trend Tag Name	Expression
P2_P	TIC_P2_PV
P3_P	TIC_P3_PV
P4_P	TIC_P4_PV
HT_P	TIC_HOLD_PV

11.3. Displaying Trends

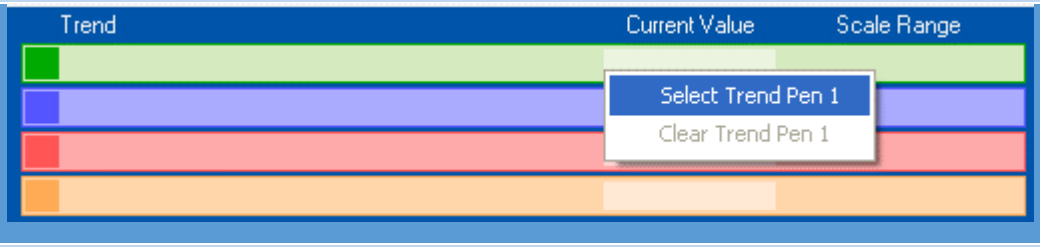
- **Exercise 11-2**

Trend some of the tags in the Milk_Treatment project.

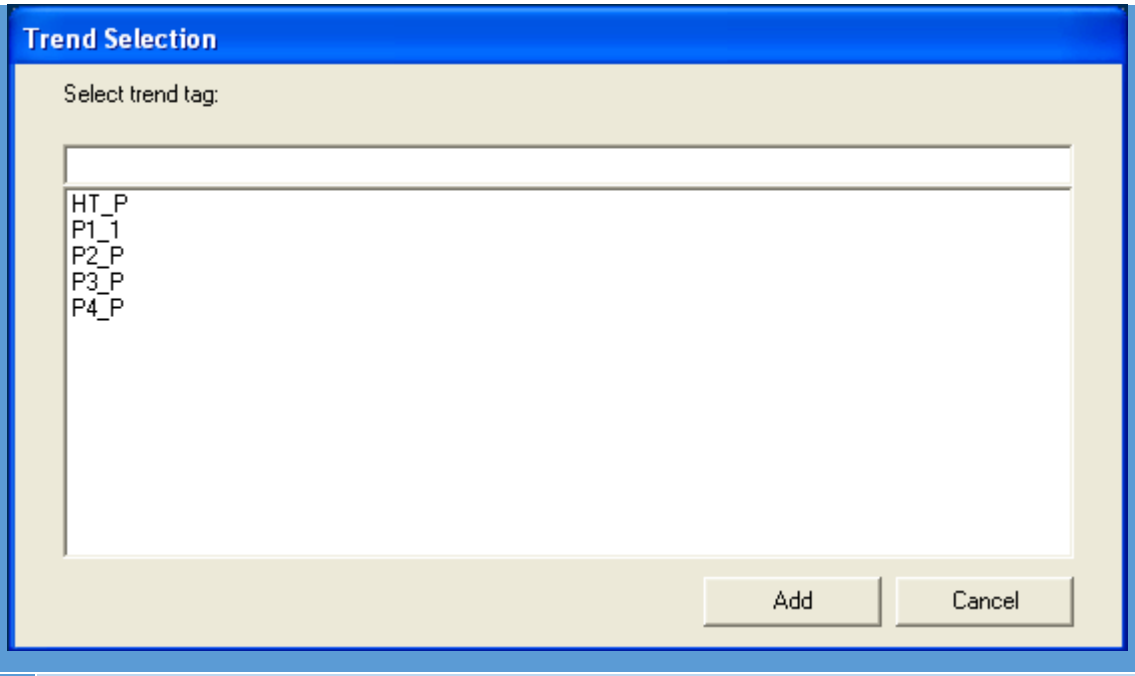
Step	Action
1	**View the trends you have just defined by using the preconfigured page CSV Trend.** i. Compile and run the project. ii. Open the Single Trend page by pressing trends button  . iii. When the Single Trend page opens it will look like this.



iv. Right click the first Trend Field and choose Select Trend Pen 1.



v. When the Trend Selection dialog opens select the trend tag HT_P and click the Add button.



	vi. Add the rest of the trend tags to the trend pens.
2	**Shutdown the Project.**

Disclaimer:

The information contained in this book,
are subject to change, without notice.

If you have any questions, comments, or concerns, please Contact Info

Phone 0128 887 7229

Email compu.systems@hotmail.com

Facebook www.facebook.com/pages/CompuSystems