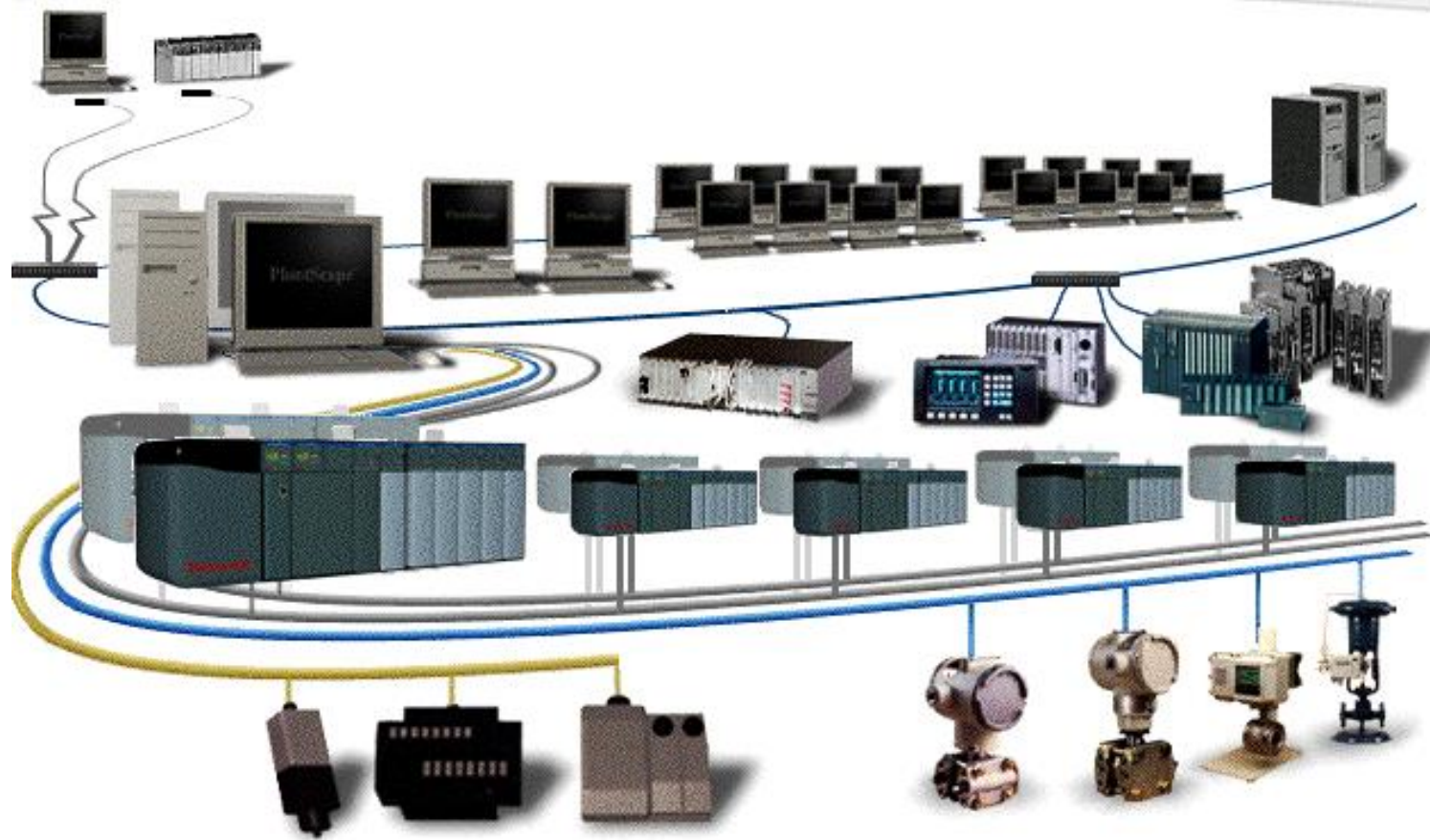
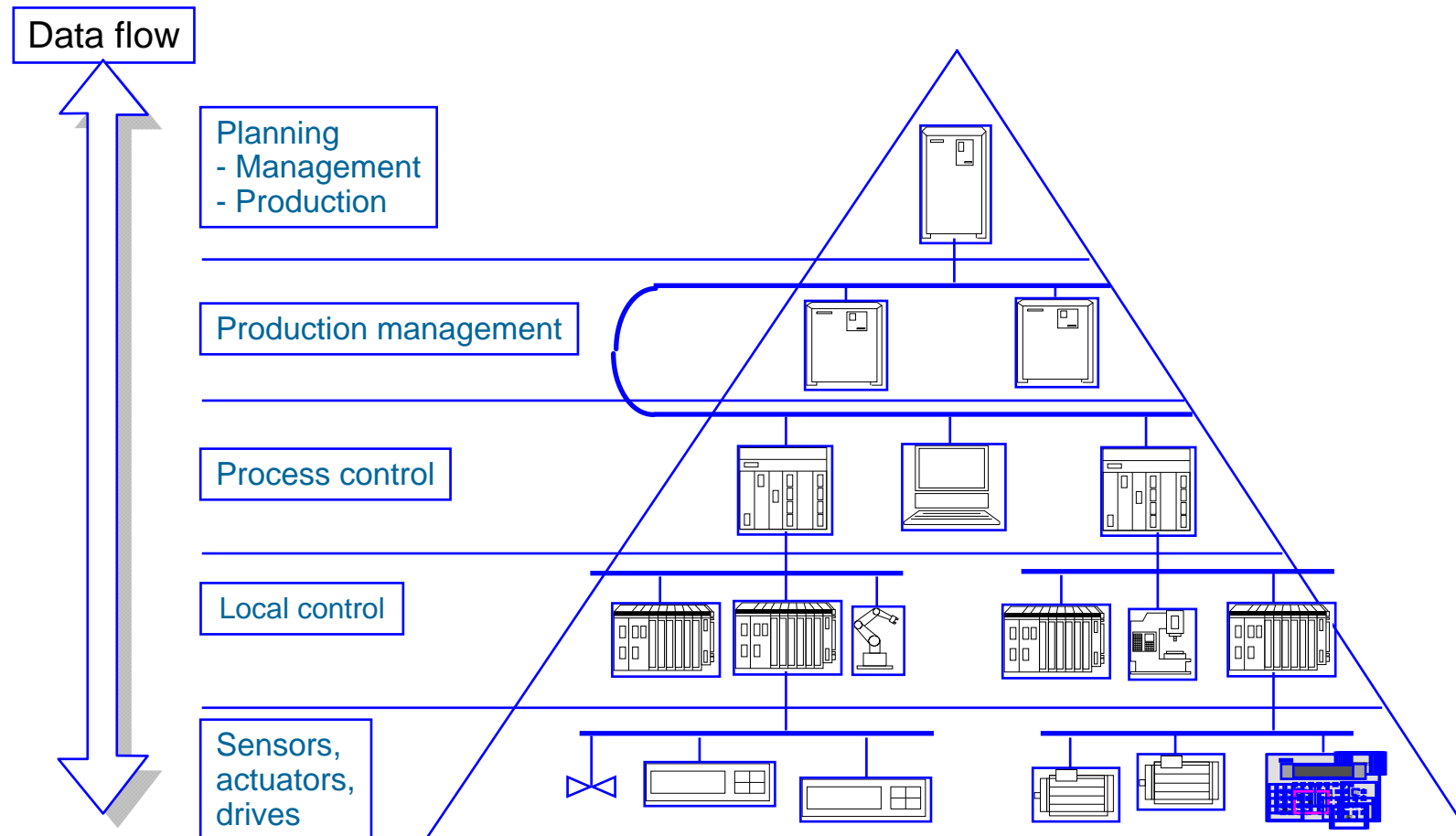


# LẬP TRÌNH WINCC CHO HỆ THỐNG SCADA

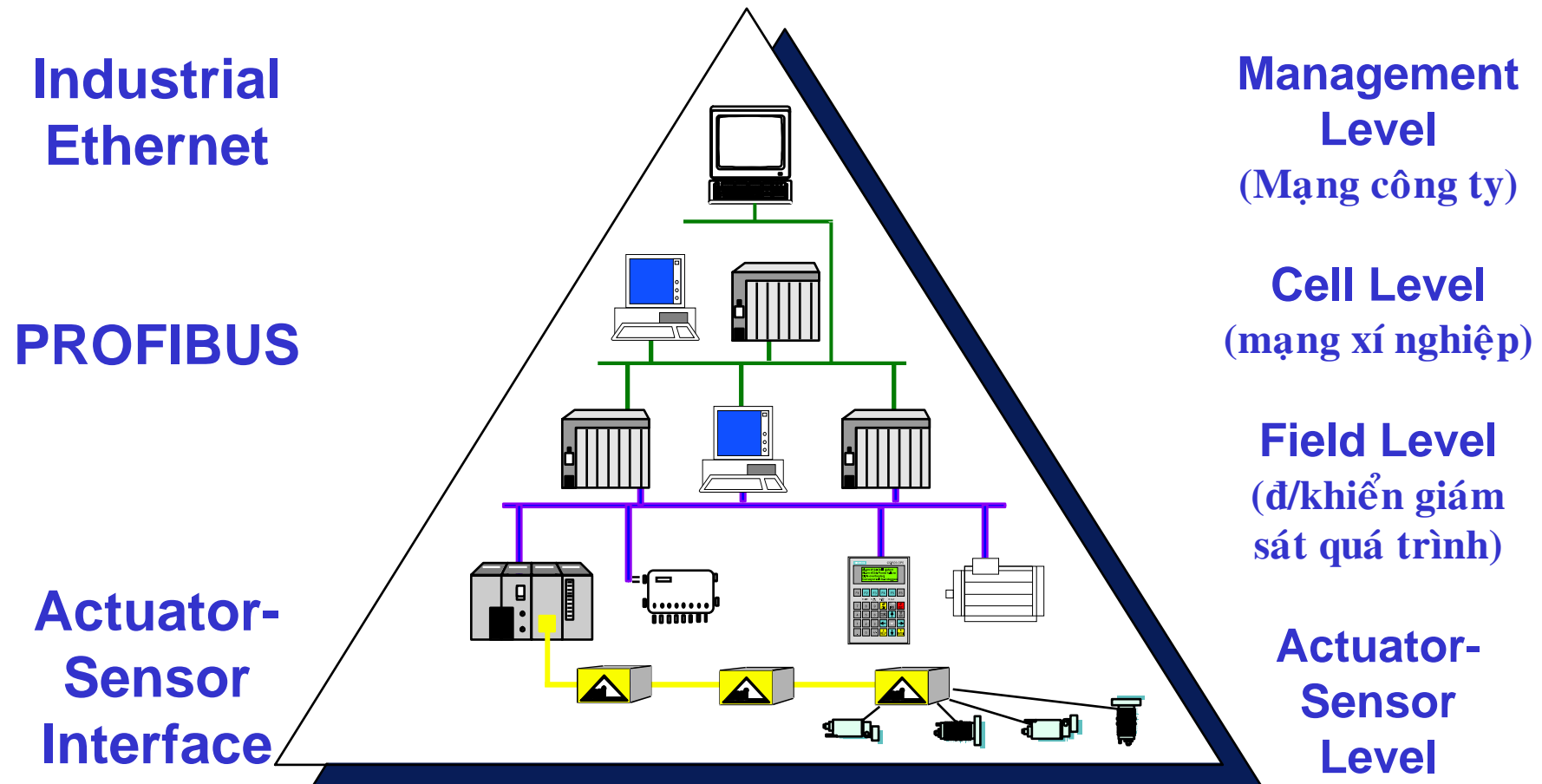


Hệ thống giám sát và thu thập dữ liệu SCADA điển hình  
Supervisory Control and Data Acquisition System

# Cấu trúc phân cấp của một hệ thống tự động



# Giải pháp truyền thông của Siemens: Simatic-net



## **Một số chức năng cơ bản của hệ thống SCADA**

- + Điều khiển (Control)**
- + Hiển thị (Display): Trend (Historical), Table...**
- + Cảnh báo (Alarm)**
- + Lưu trữ (Archive)**
- + In ấn (Report)**

## **Một số phần mềm cho hệ SCADA hiện có tại Việt Nam**

WinCC (Siemens), Fix-Intellution, Wonderware, Scitect,  
Plantscape (Honeywell)...

**Để trao đổi dữ liệu với PLC các phần mềm này đều sử dụng  
khái niệm Tag hay Point**

**WINCC**  
**WINDOWS CONTROL CENTER**

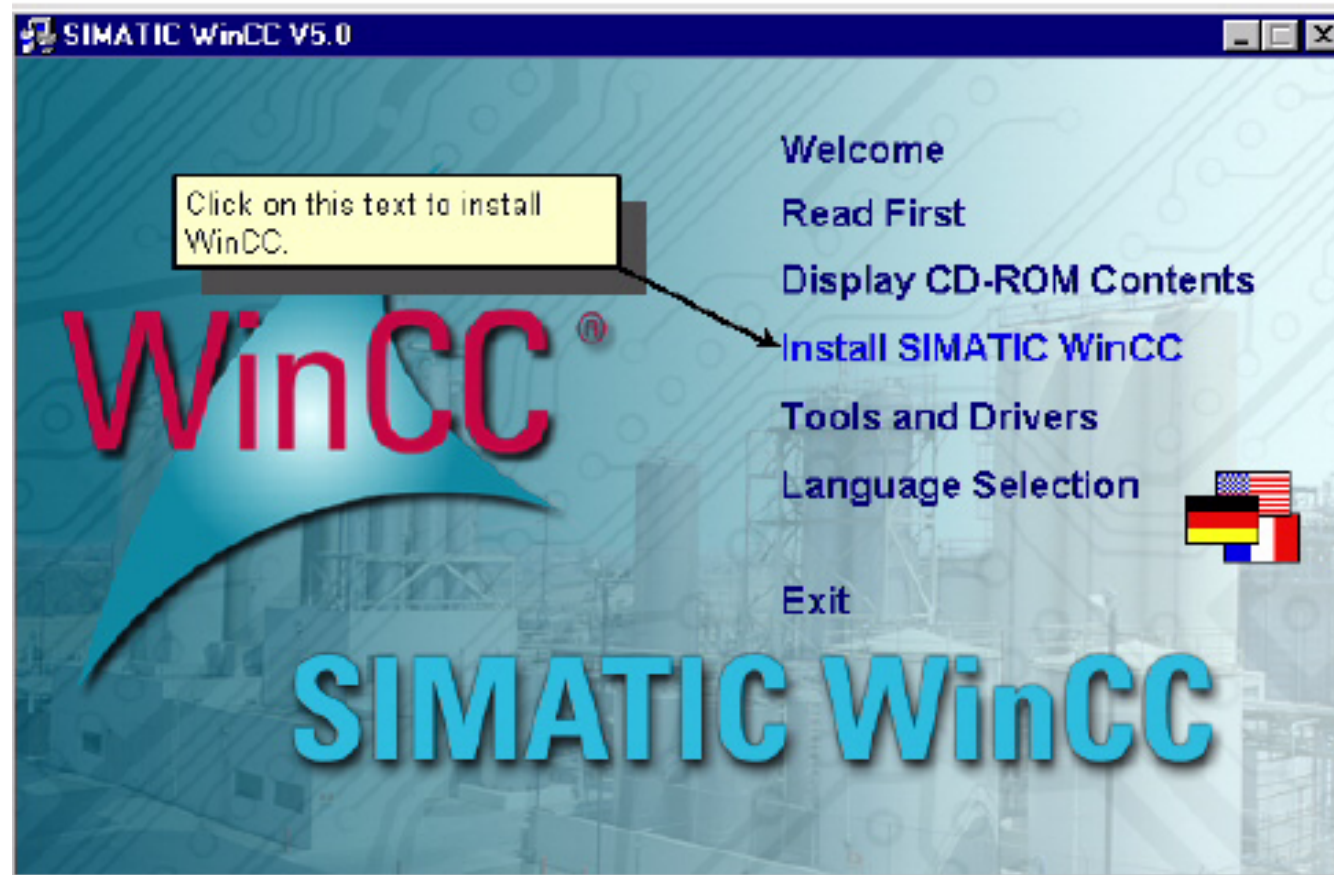
## Cài đặt WinCC

### Cấu hình máy tính:

	Minimum	Recommended
CPU	INTEL Pentium II 266 MHz	INTEL Pentium II 400 MHz
RAM	96 MB	128 MB
Graphic Controller	SVGA (4 MB)	XGA (8 MB)
Resolution	800 * 600	1024 * 768
Hard Disk	500 MB available space	> 500 MB available space
	CD-ROM Drive	CD-ROM Drive


**Hệ điều hành sử dụng cho WinCC5.0: Win2000 Service Pack 2, WinNT Service Pack 5**

## Đưa đĩa CD vào





**User Information** [X]



Type your name below. You must also type the name of the company you work for and the product serial number.

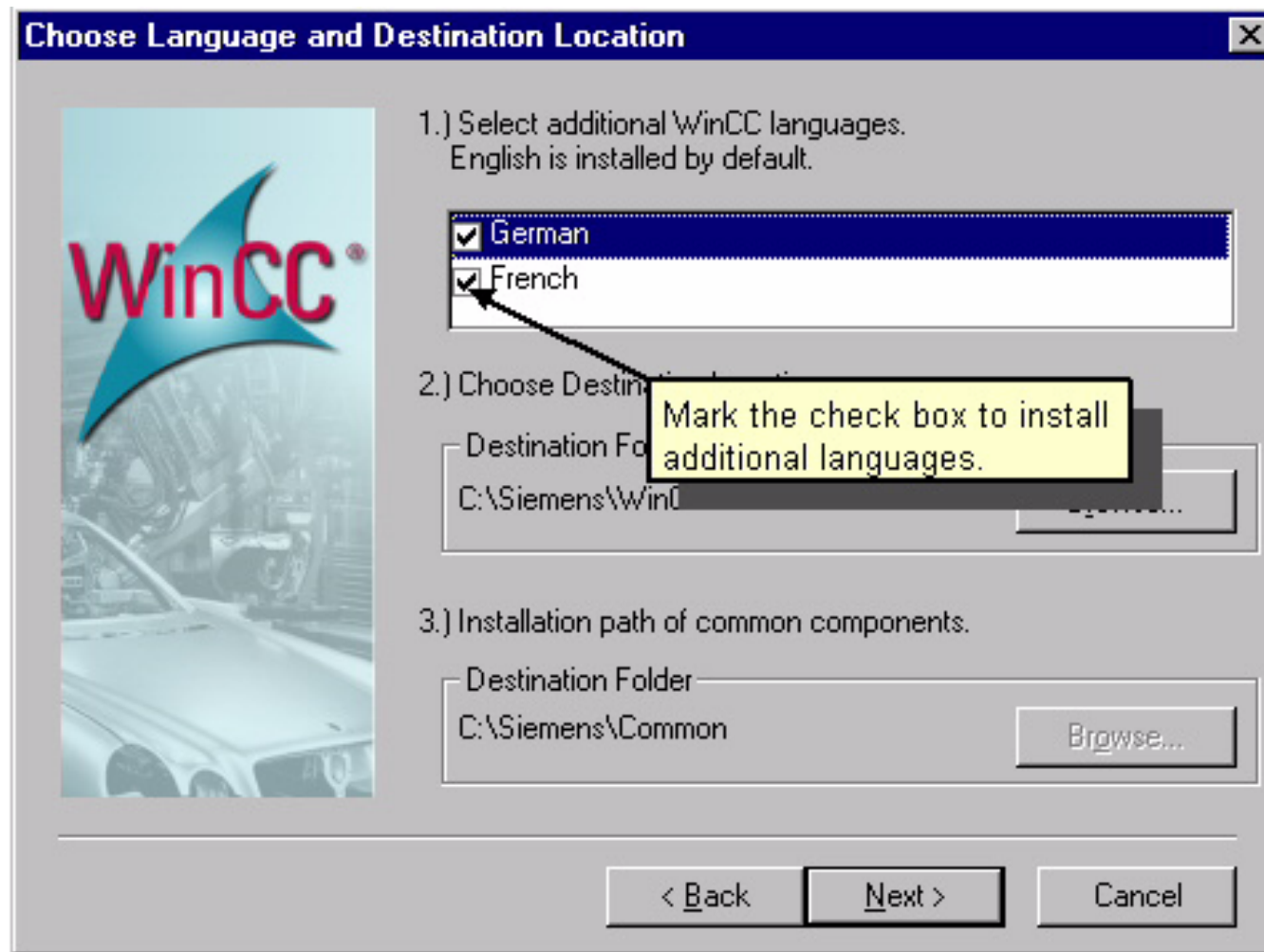
Name:

Company:

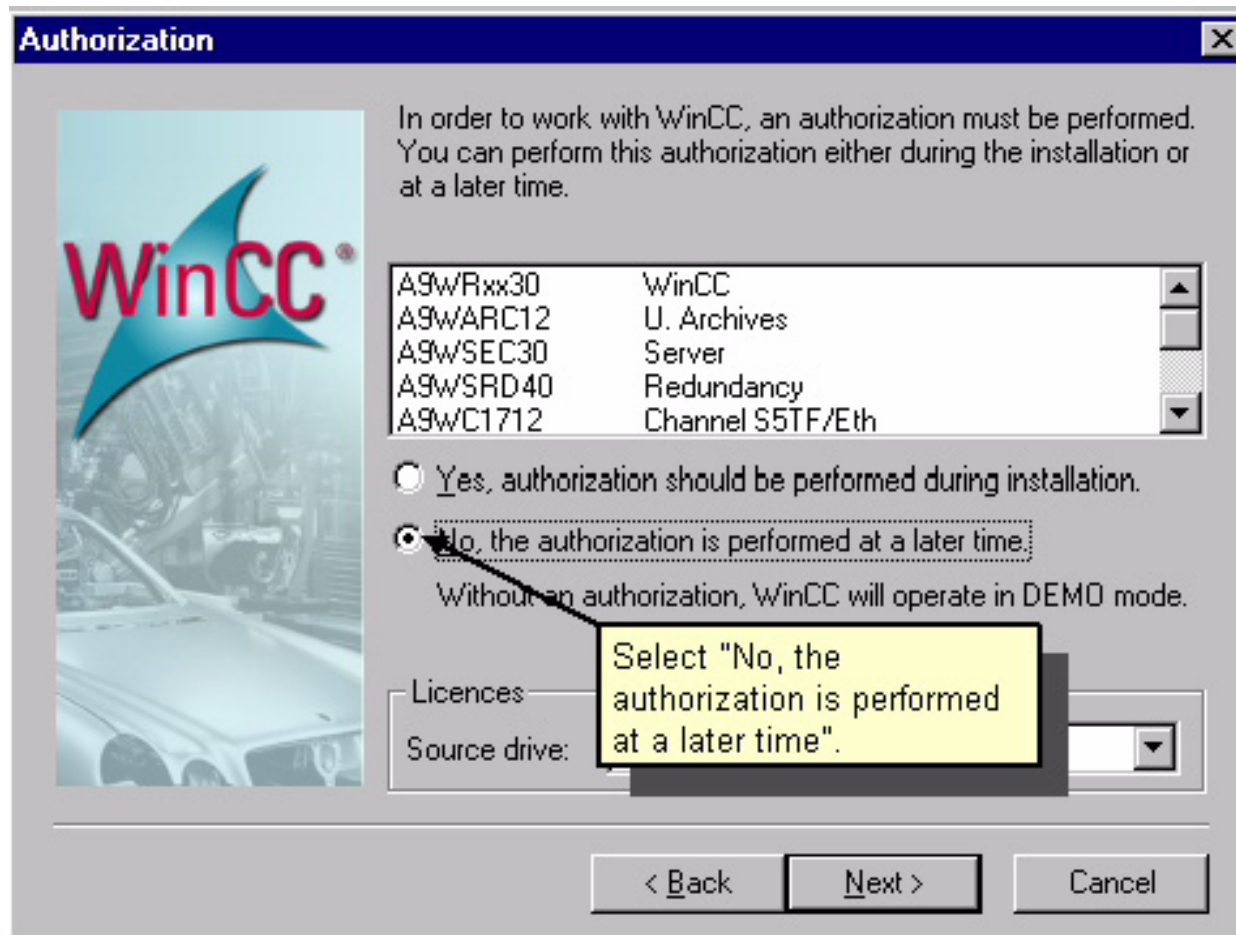
Serial:

Enter the word "Demo" in this field. Afterwards, click on "Next".

< Back   Next >   Cancel



Chọn Next và làm theo hướng dẫn trên màn hình



Làm theo chỉ dẫn trên màn hình cho đến khi hoàn tất

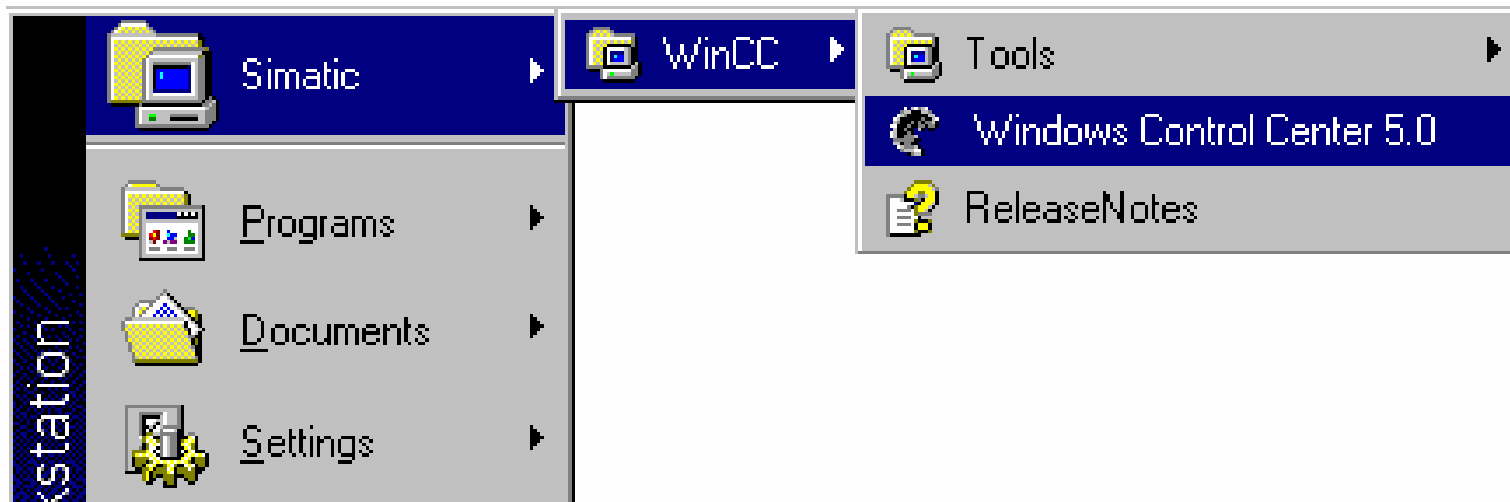
## **Các bước để tạo một Project trong WinCC**

- 1. Khởi động WinCC**
- 2. Tạo một Project mới**
- 3. Cài đặt Driver kết nối PLC**
- 4. Định nghĩa các Tag sử dụng**
- 5. Tạo và soạn thảo một giao diện người dùng**
- 6. Cài đặt thông số cho winCC Runtime**
- 7. Chạy chương trình (Activate)**
- 8. Có thể dùng chương trình WinCC Variable Simulator để mô phỏng hoạt động của các Tag**

## Trình tự tạo một Project

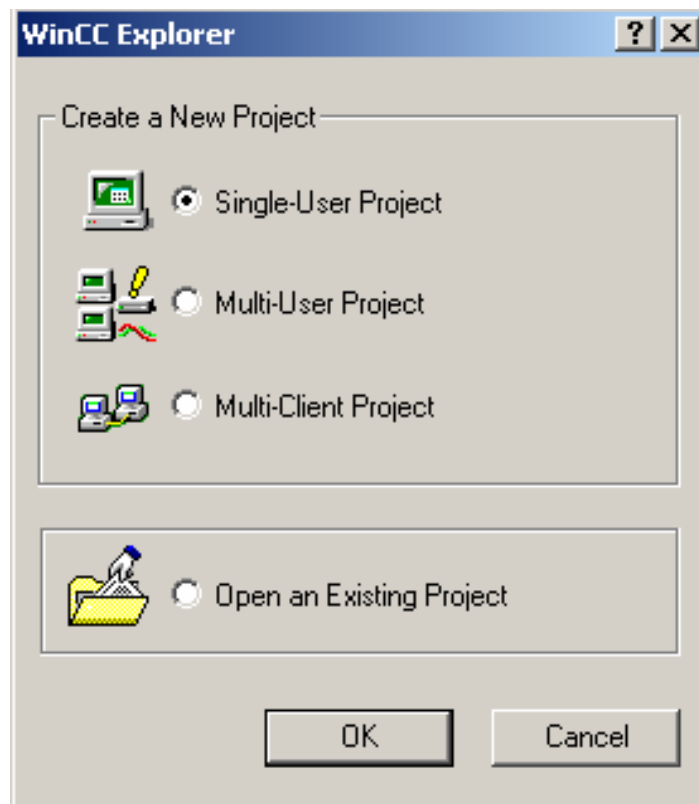
1. Khởi động WinCC:

Nhấn nút Start→Simatic→WinCC→Windows Control Center

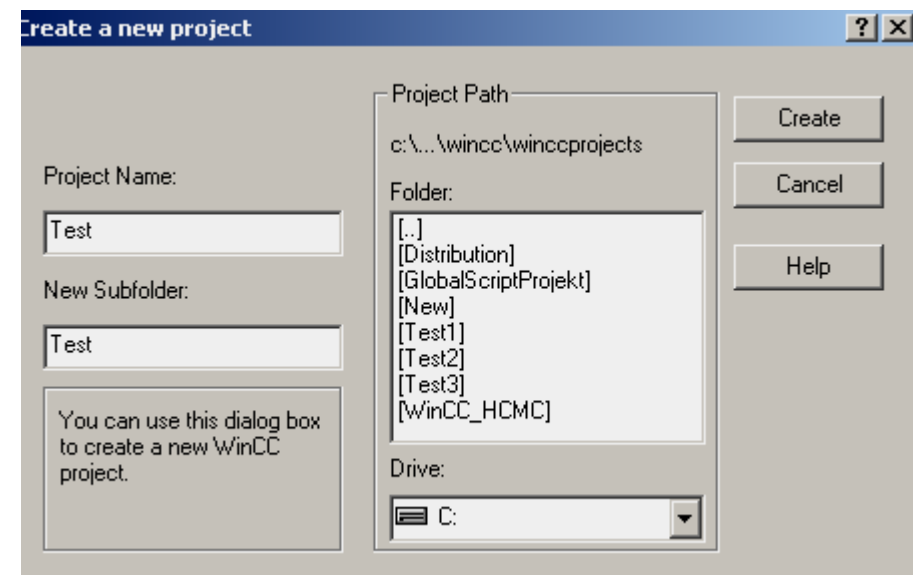


## 2. Tạo một Project

Chọn Single-User Project

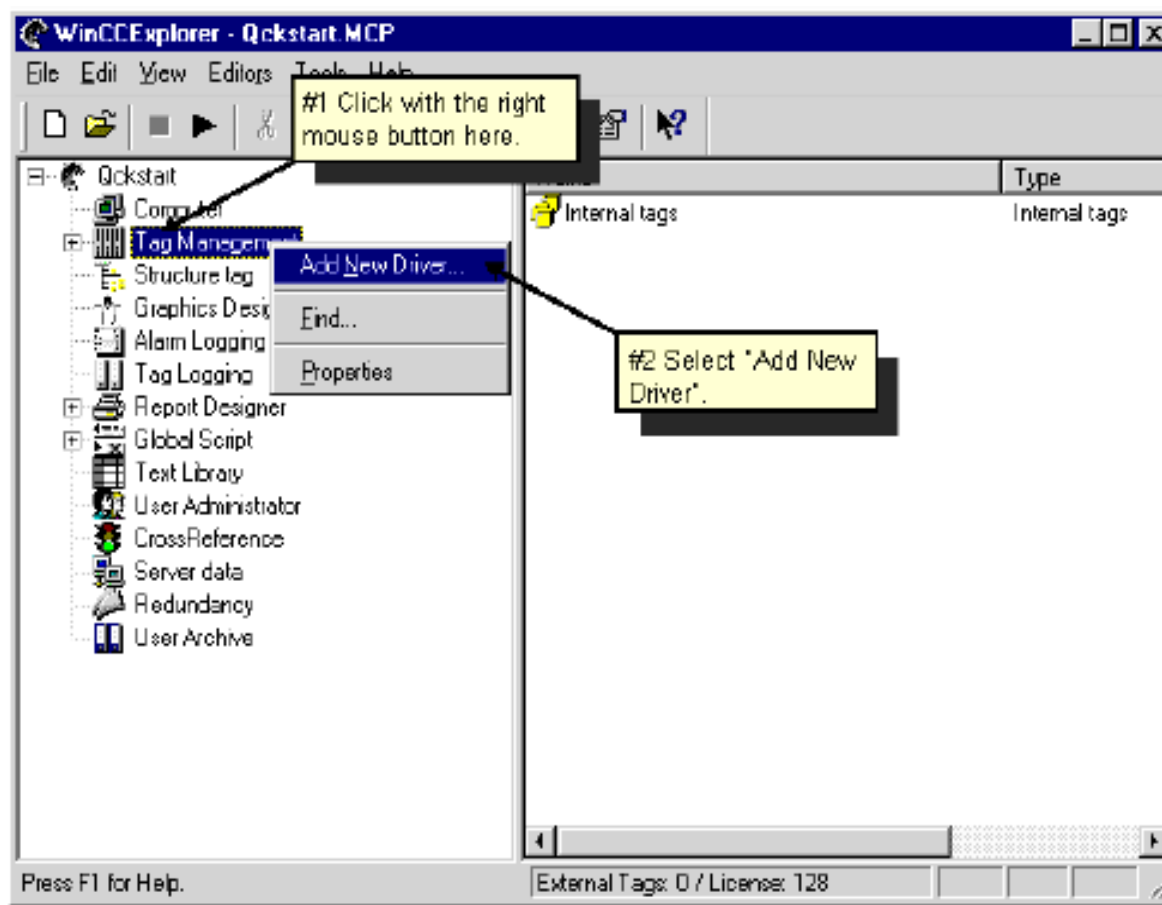


Gõ tên Project vào project name

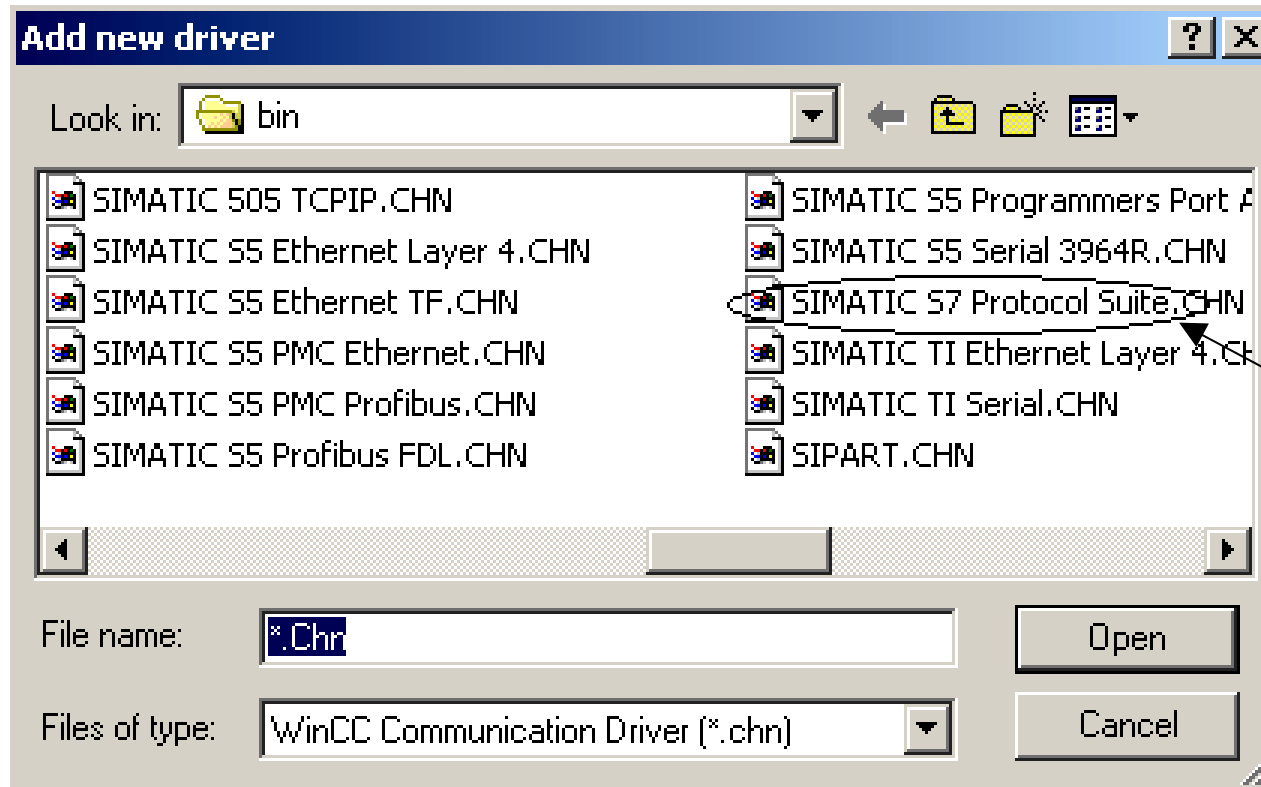


### 3. Cài đặt Driver kết nối PLC

Click chuột phải vào Tag Management → Add new Driver



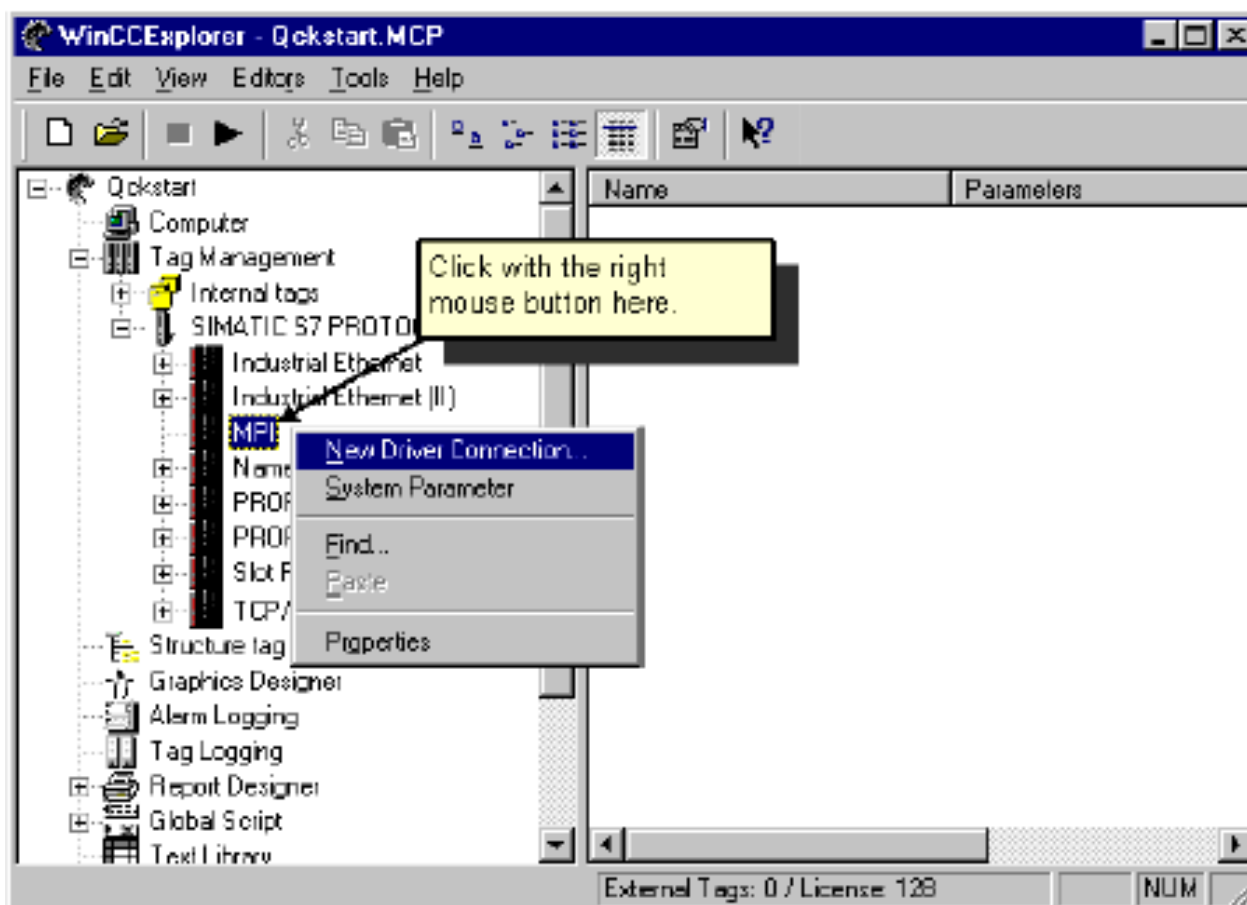
Chọn Driver để kết nối PLC từ hộp thoại sau:



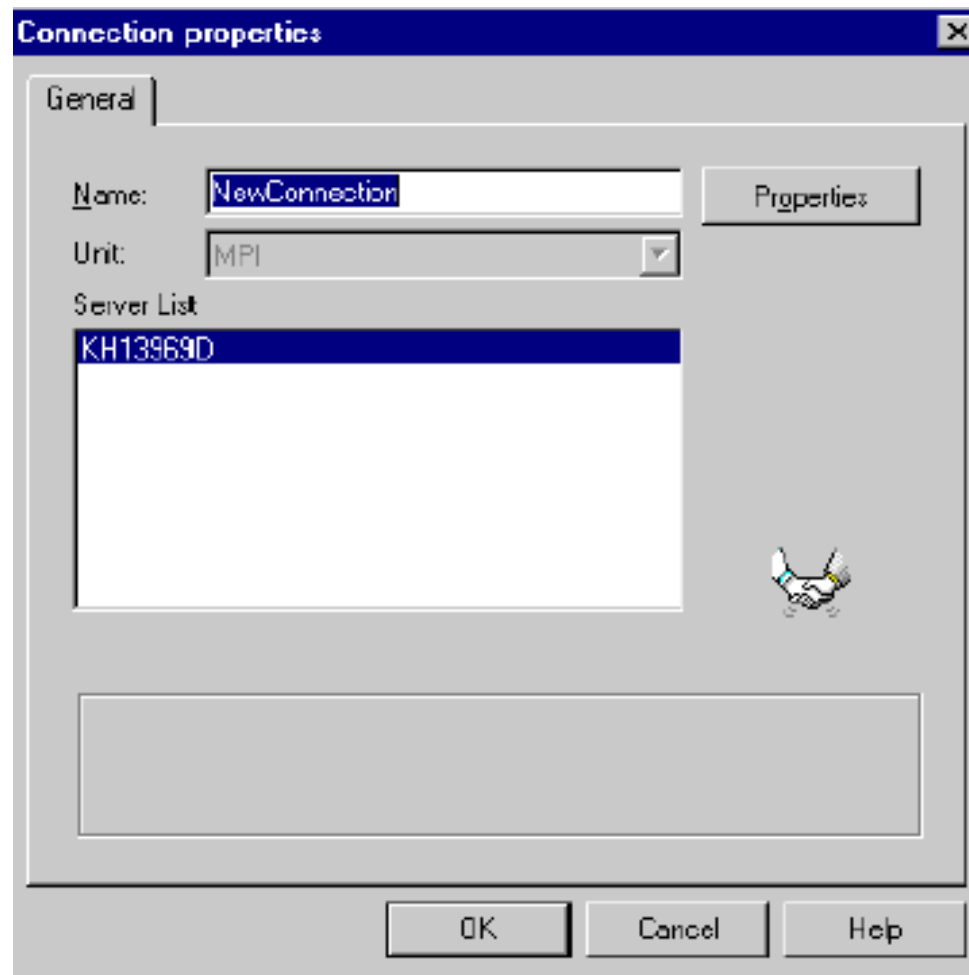
Kết nối với  
PLC S7-  
300 chọn  
Driver này



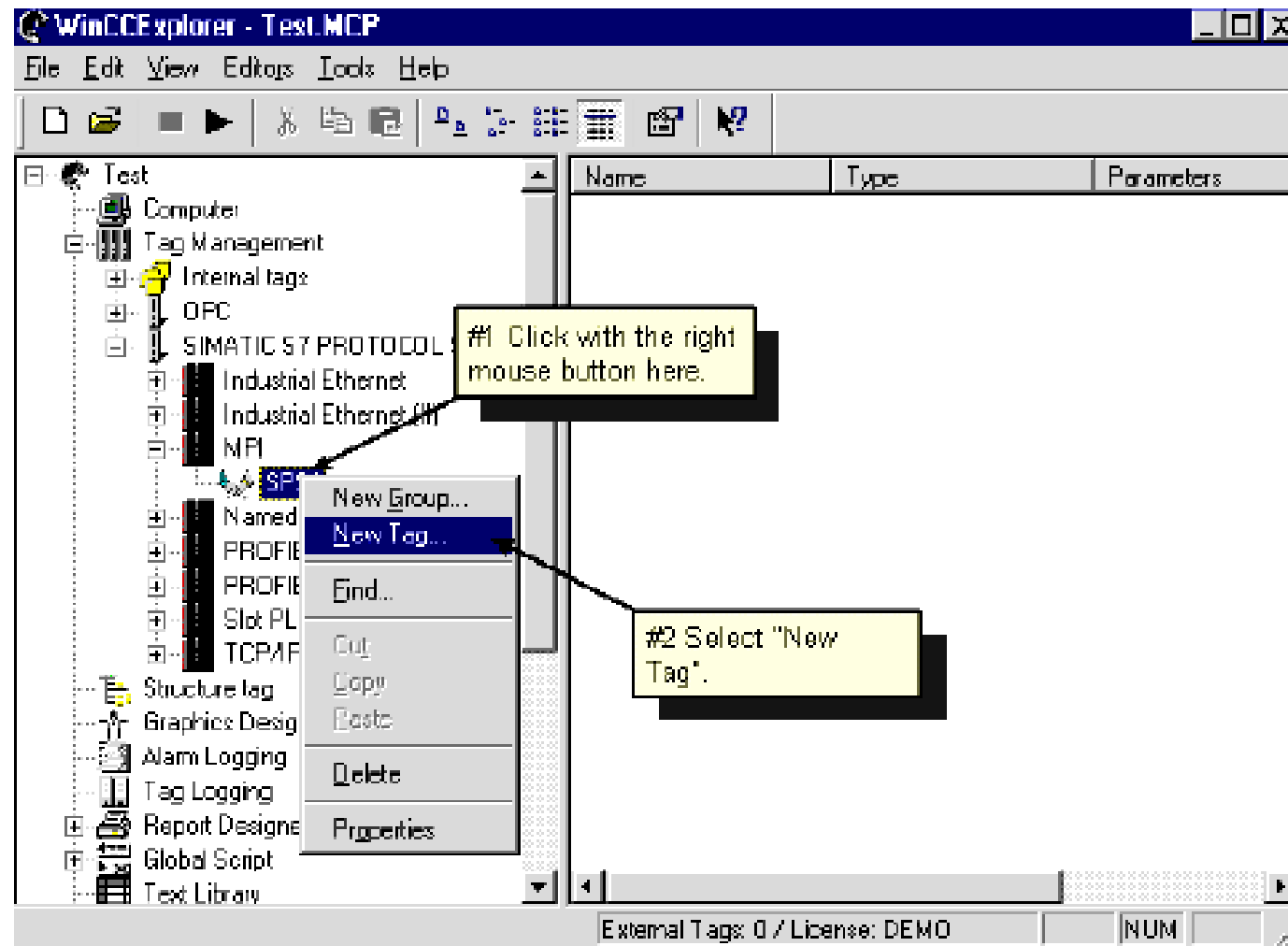
+ Cần phân biệt hai loại Internal Tag và External (Process) Tag  
**Tạo External Tag**



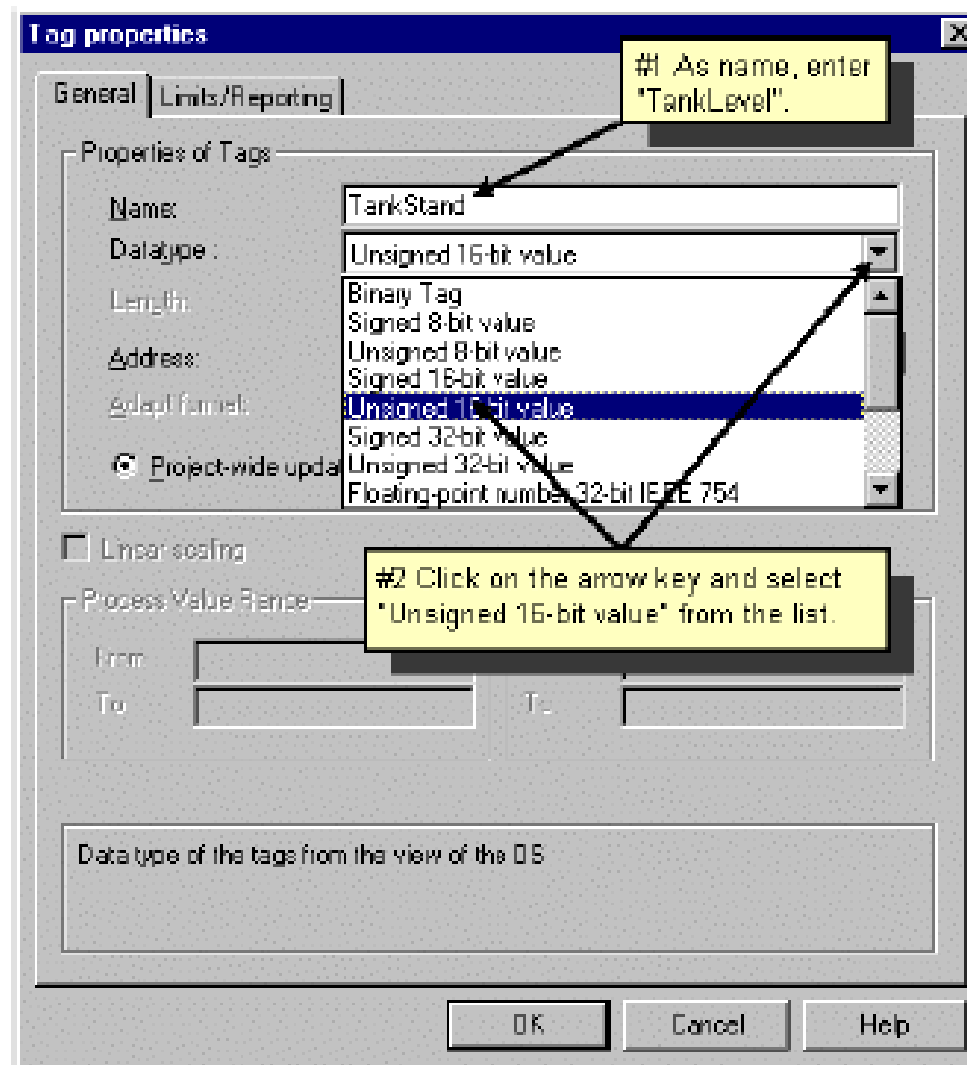
Gõ tên và nhấn OK từ hộp thoại New Connection Properties



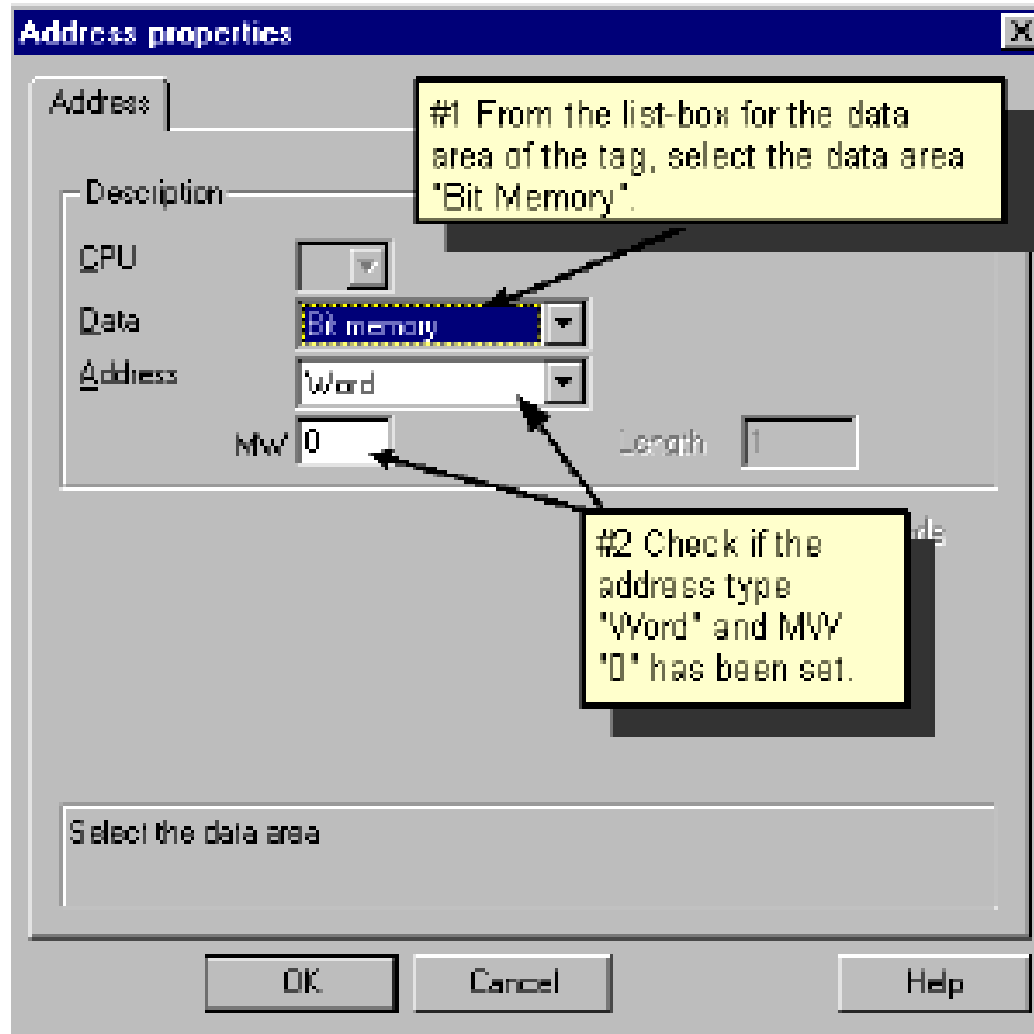
## Chọn New Tag từ Connection vừa tạo



## Gõ tên Tag và chọn kiểu dữ liệu của Tag

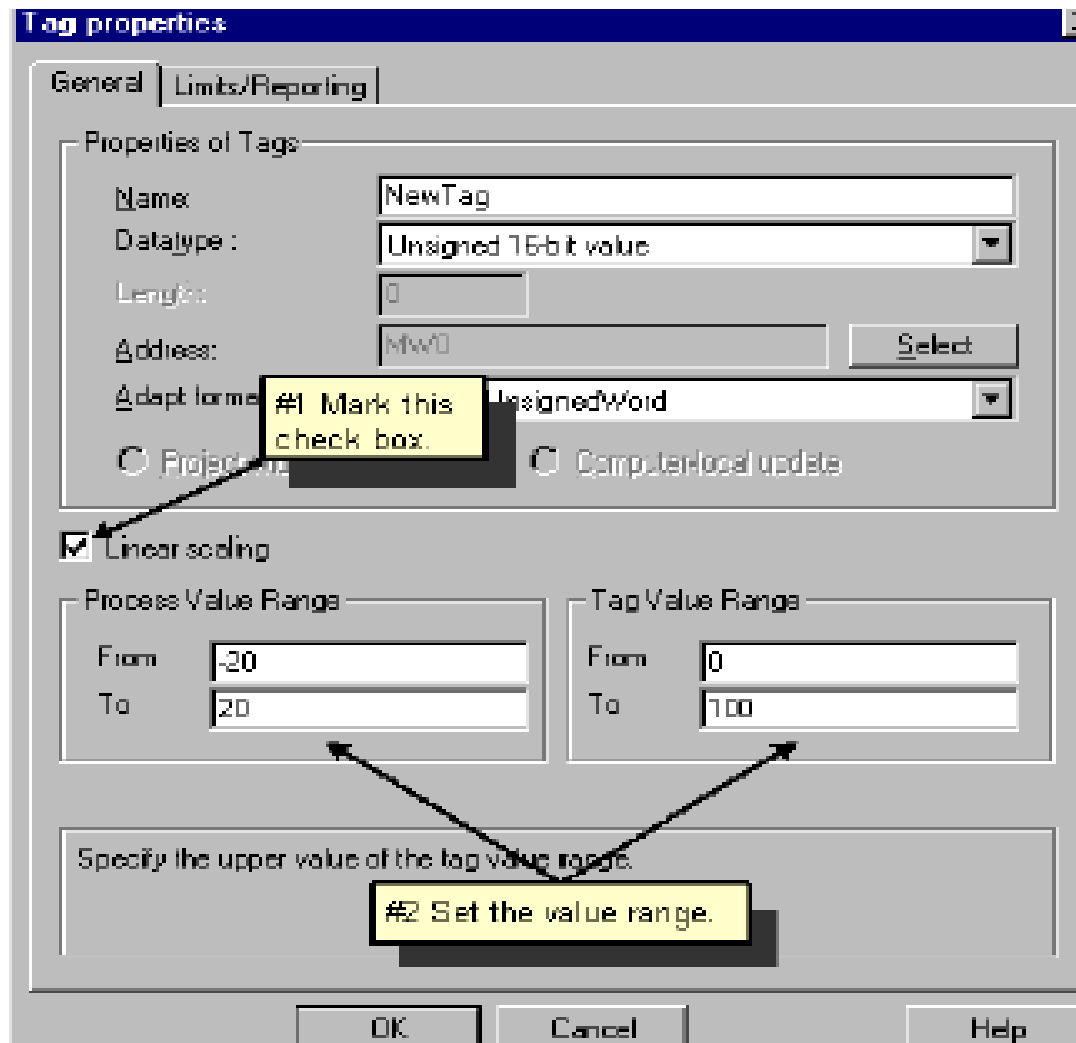


Click chuột vào nút Select để gán địa chỉ cho Tag



Chọn OK để lưu địa chỉ cho Tag

## Có thể chọn ô Limit Scaling để tạo Scale(Tỷ lệ) cho Tag Analog



**Tag properties**

General | Limits/Reporting

Properties of Tags

Name: NewTag

Datatype: Unsigned 16-bit value

Length: 0

Address: MW0 [Select]

Adapt format: #1 Mark this check box. UnsignedWord

Project local update  Computer/local update

Linear scaling

Process Value Range

From: -20

To: 20

Tag Value Range

From: 0

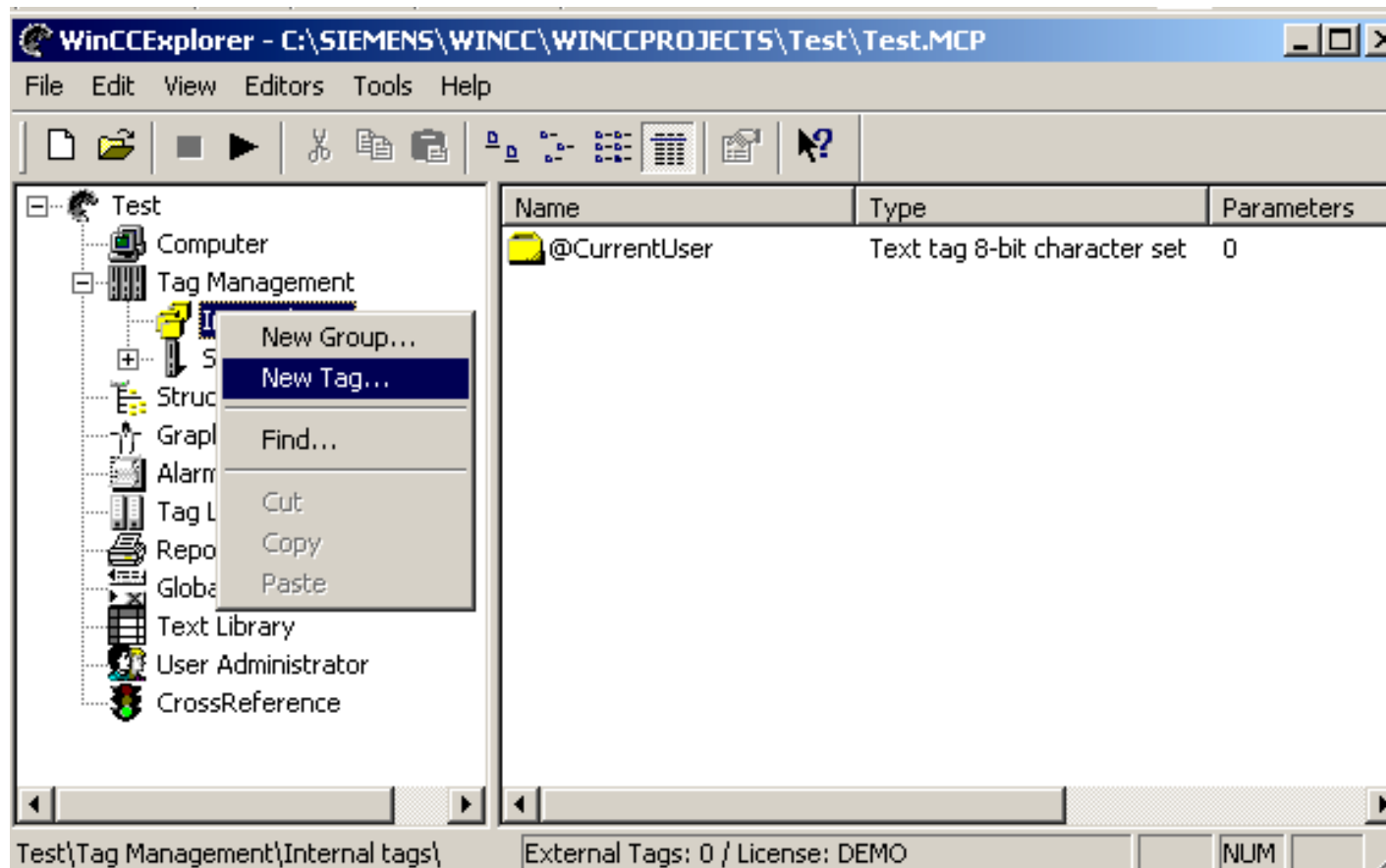
To: 100

Specify the upper value of the tag value range.

#2 Set the value range.

OK Cancel Help

Click chuột phải vào Internal Tag và chọn New Tag để tạo  
**Internal Tag**



## Gõ tên và chọn kiểu dữ liệu cho Internal Tag

Tag properties

General | Limits/Reporting

Properties of Tags

Name:

Datatype:

Length:

Address:

Adapt format:

Project-wide update  Computer-local update

Linear scaling

Process Value Range

Value1

Value2

Tag Value Range

Value1

Value2

Please note that when you use tags in the dynamic dialog, the name of the tag does not contain any national special characters and does not begin with a number.

Chọn OK để lưu Internal Tag



**Thực hành:** Tạo Internal tag có tên là tank level, kiểu dữ liệu là Unsigned 16 bits

Tag properties

General Limits/Reporting

Properties of Tags

Name: tanklevel

Datatype: Unsigned 16-bit value

Length: 2

Address: [ ] Select

Adapt format: [ ]

Project-wide update  Computer-local update

Linear scaling

Process Value Range

Value1 [ ]

Value2 [ ]

Tag Value Range

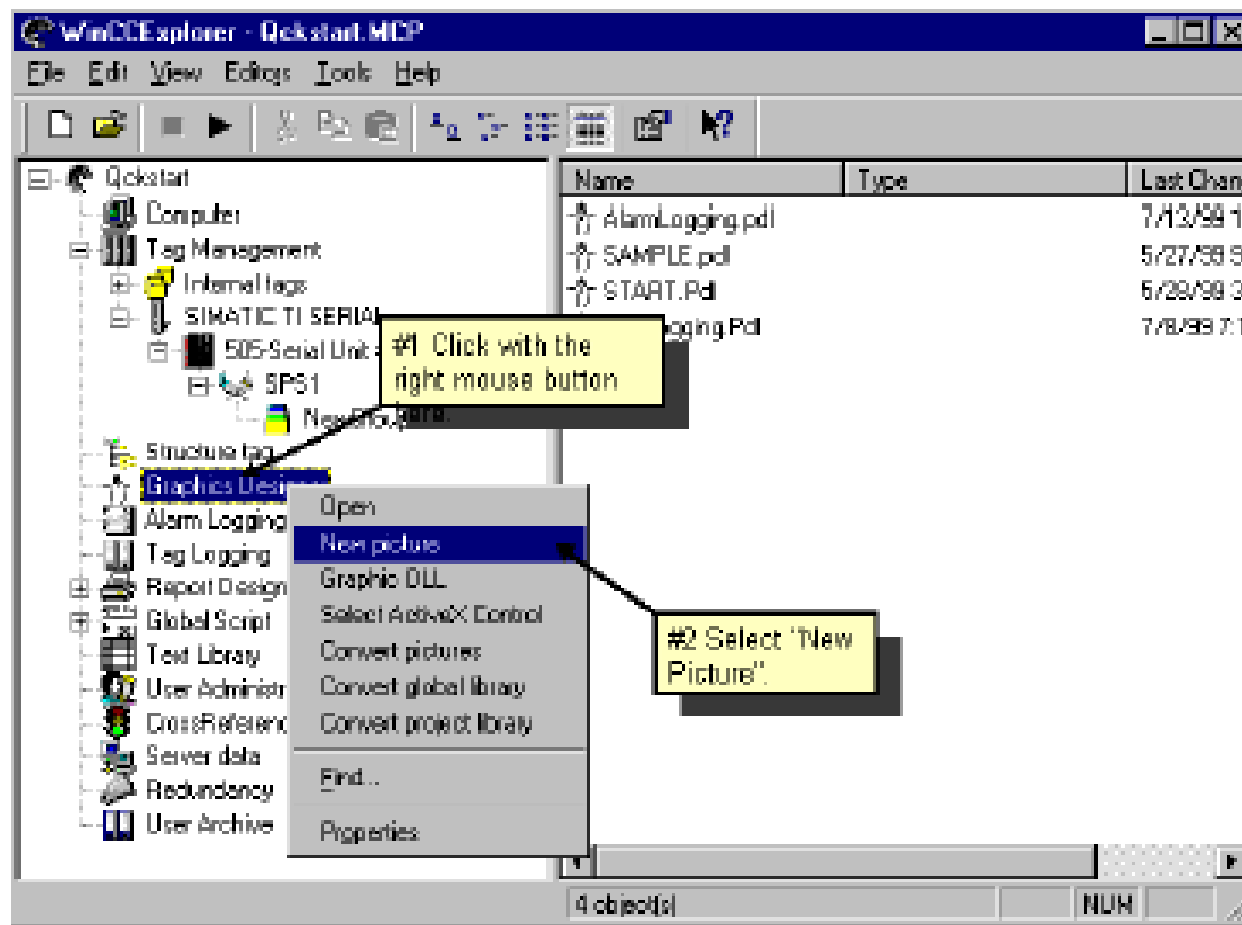
Value1 [ ]

Value2 [ ]

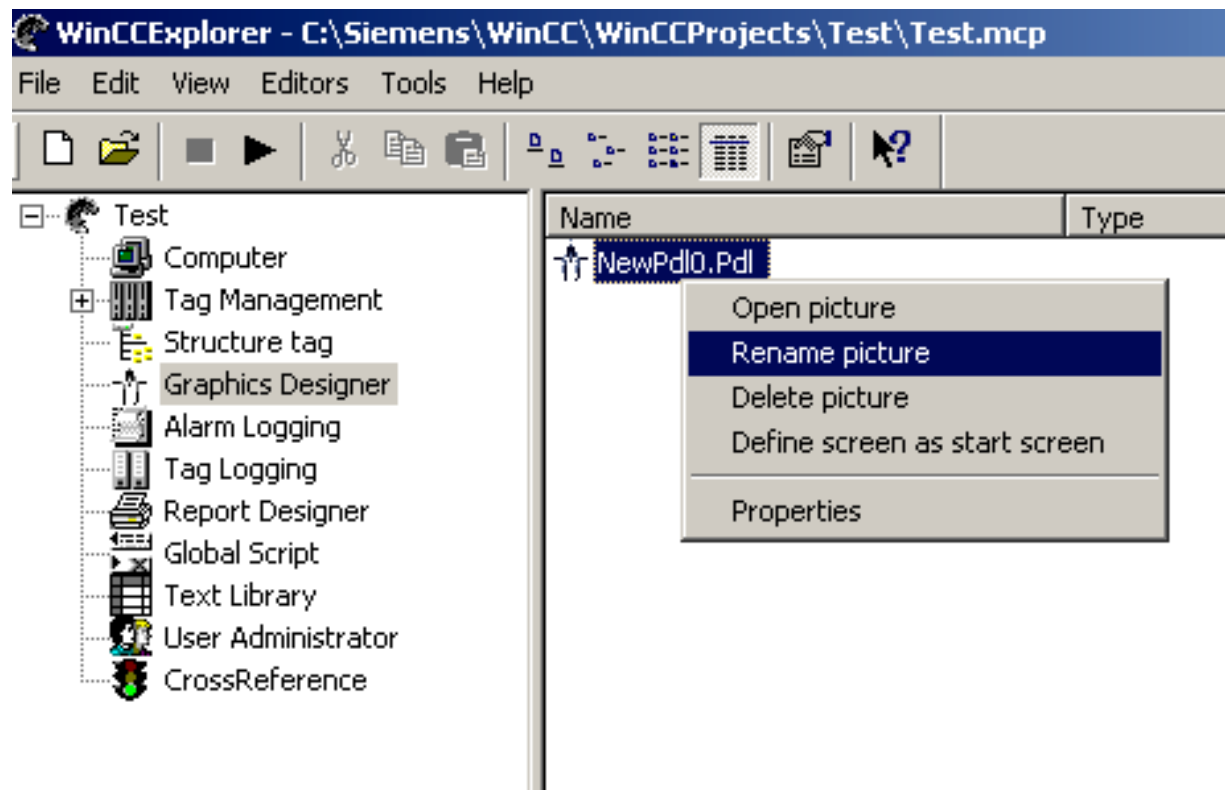
Data type of the tags from the view of the OS

## 4. Tạo giao diện

Click chuột phải vào Graphics Designer → Chọn New

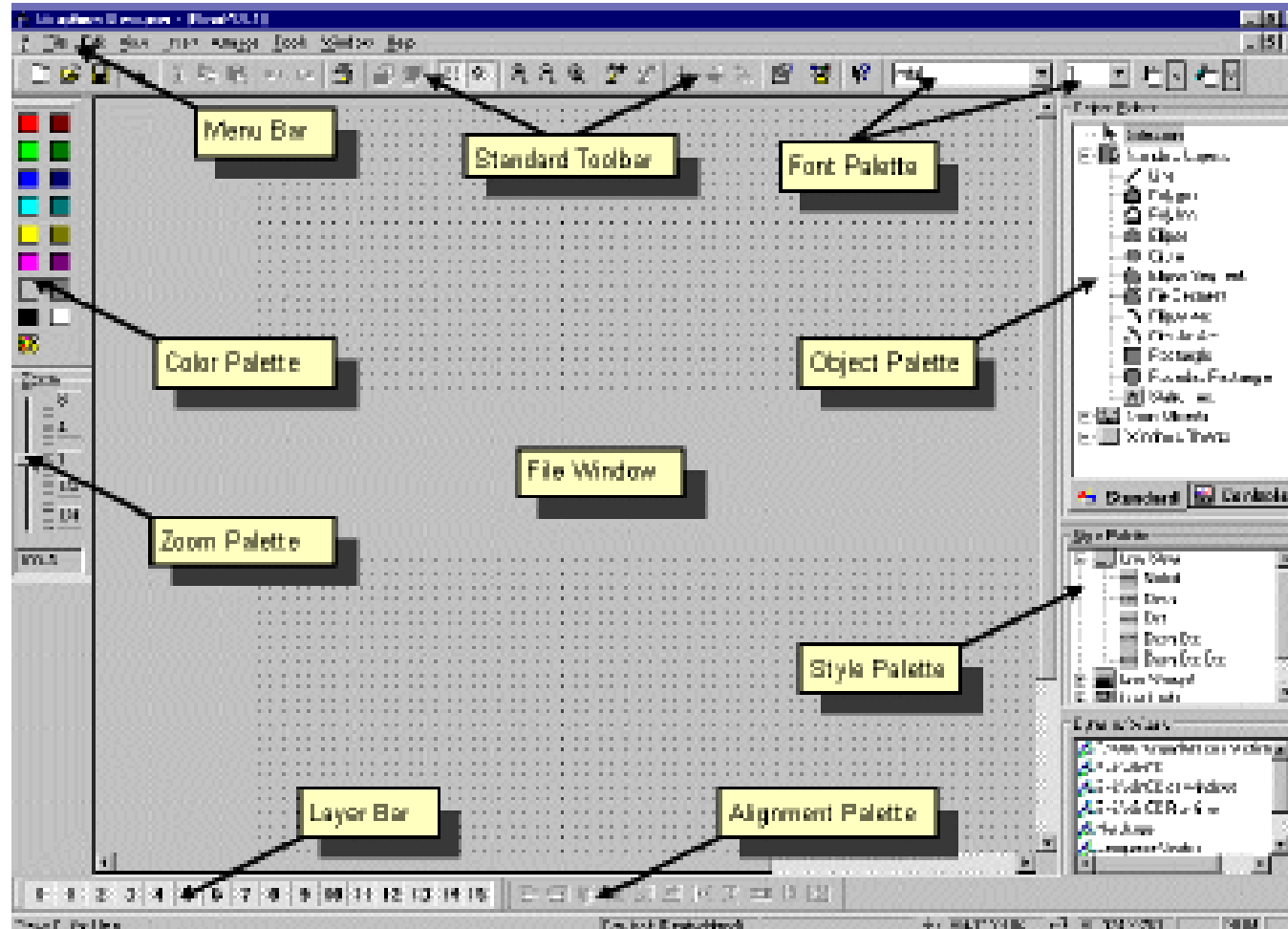


Có thể đổi tên Picture bằng cách nhấp chuột phải, chọn Rename

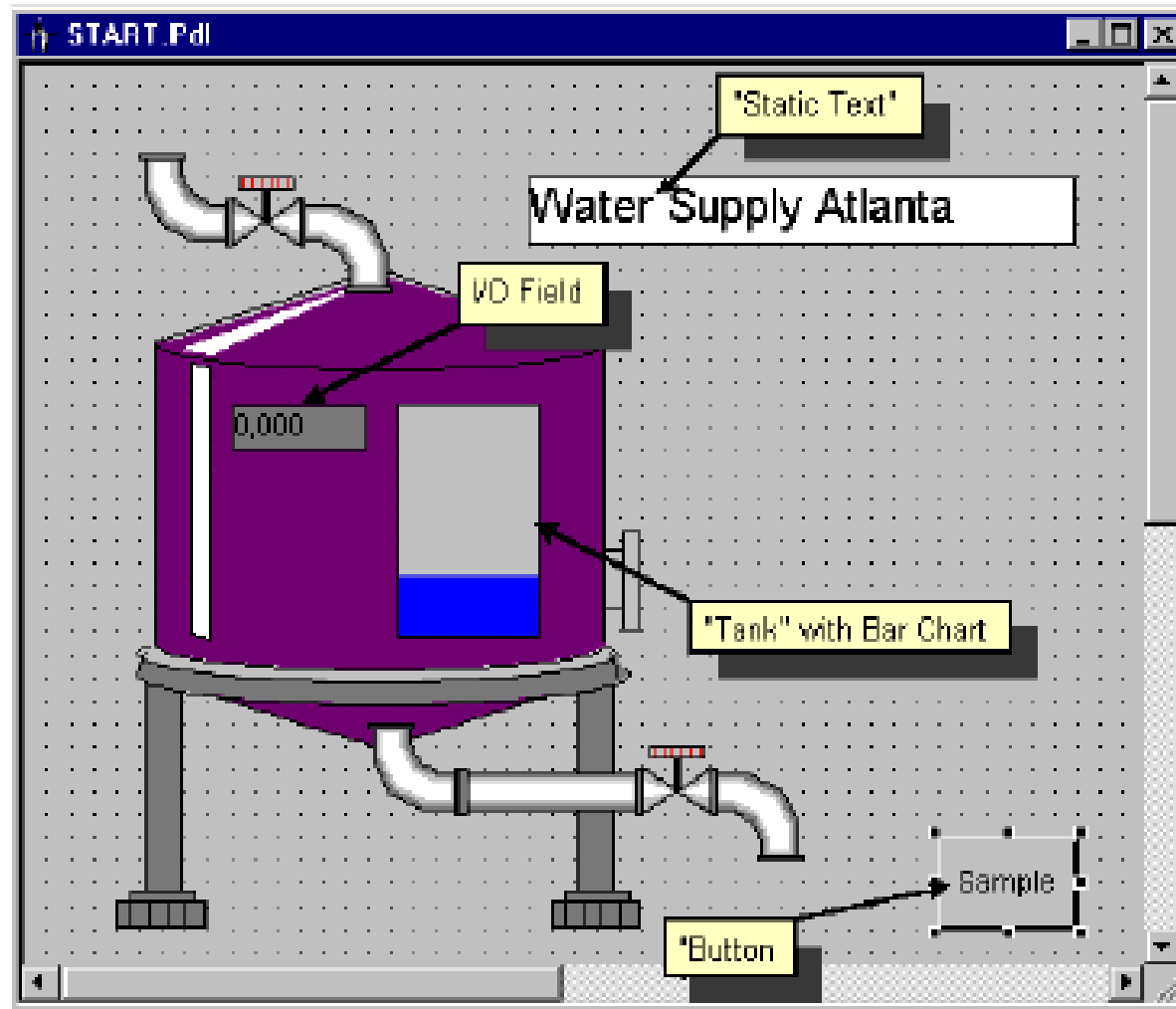


**Thực hành:** Tạo hai trang màn hình có tên là Start.pdl và Sample.pdl

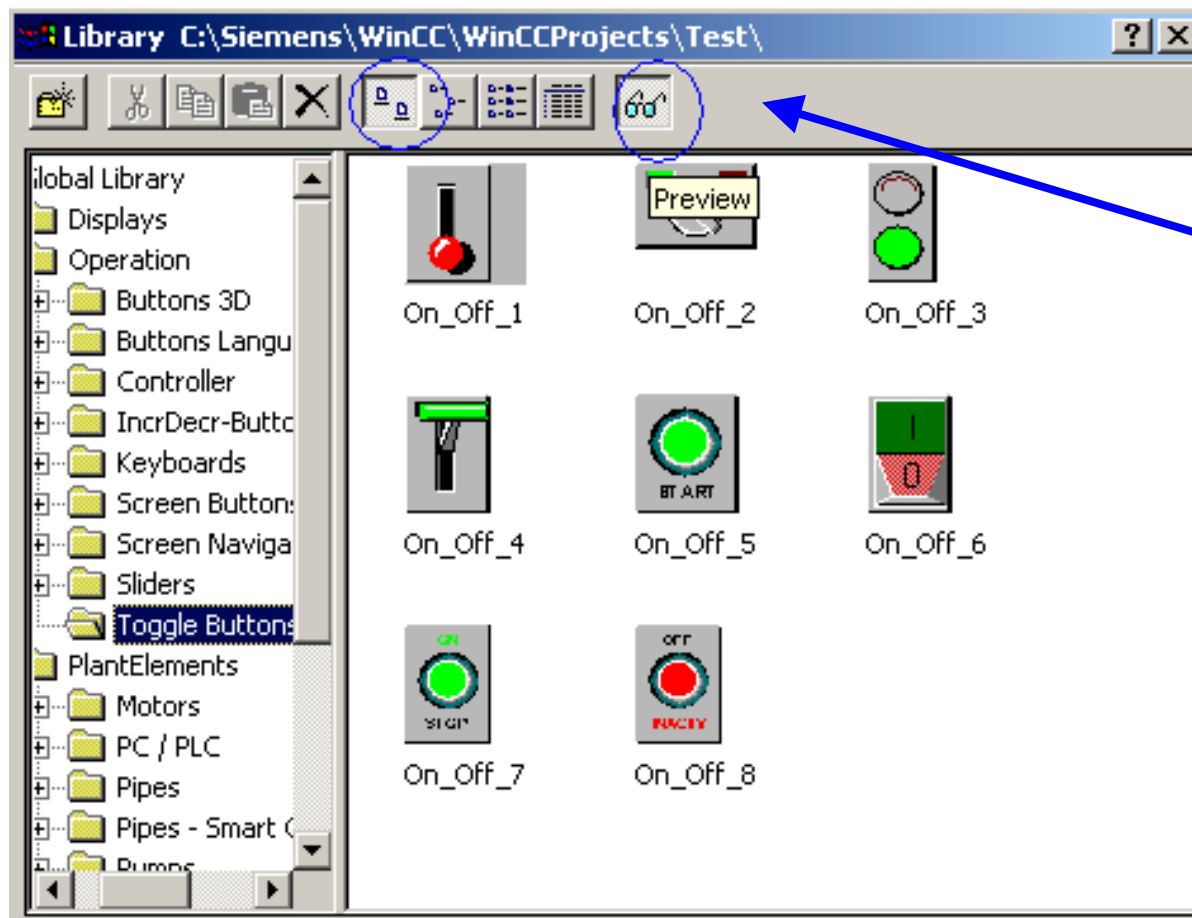
## Các công cụ trên Graphics Designer



Trên trang màn hình tên Start.pdl ,tạo giao diện sau:



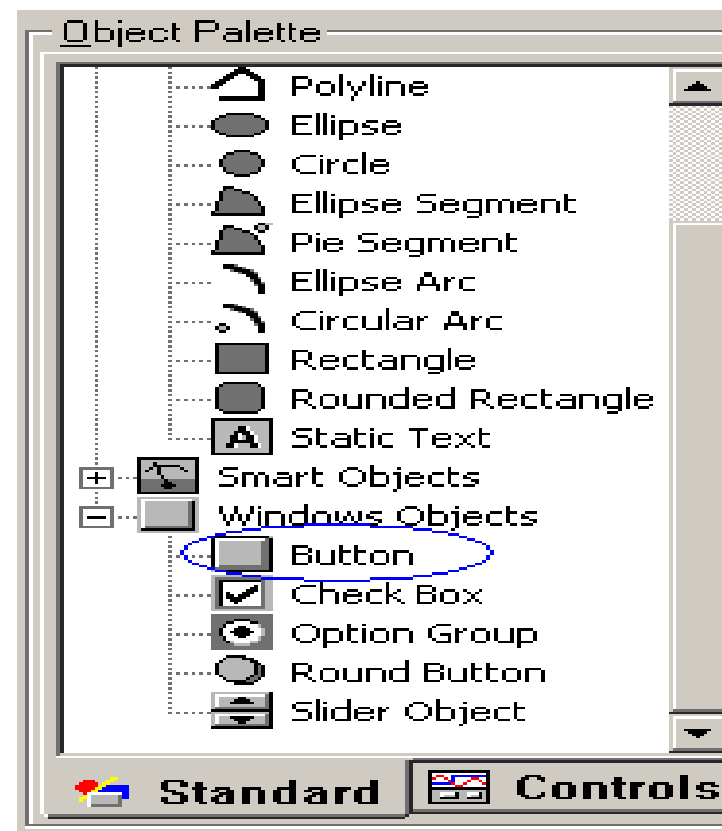
Để lấy các hình ảnh có sẵn vào menu View, chọn Library



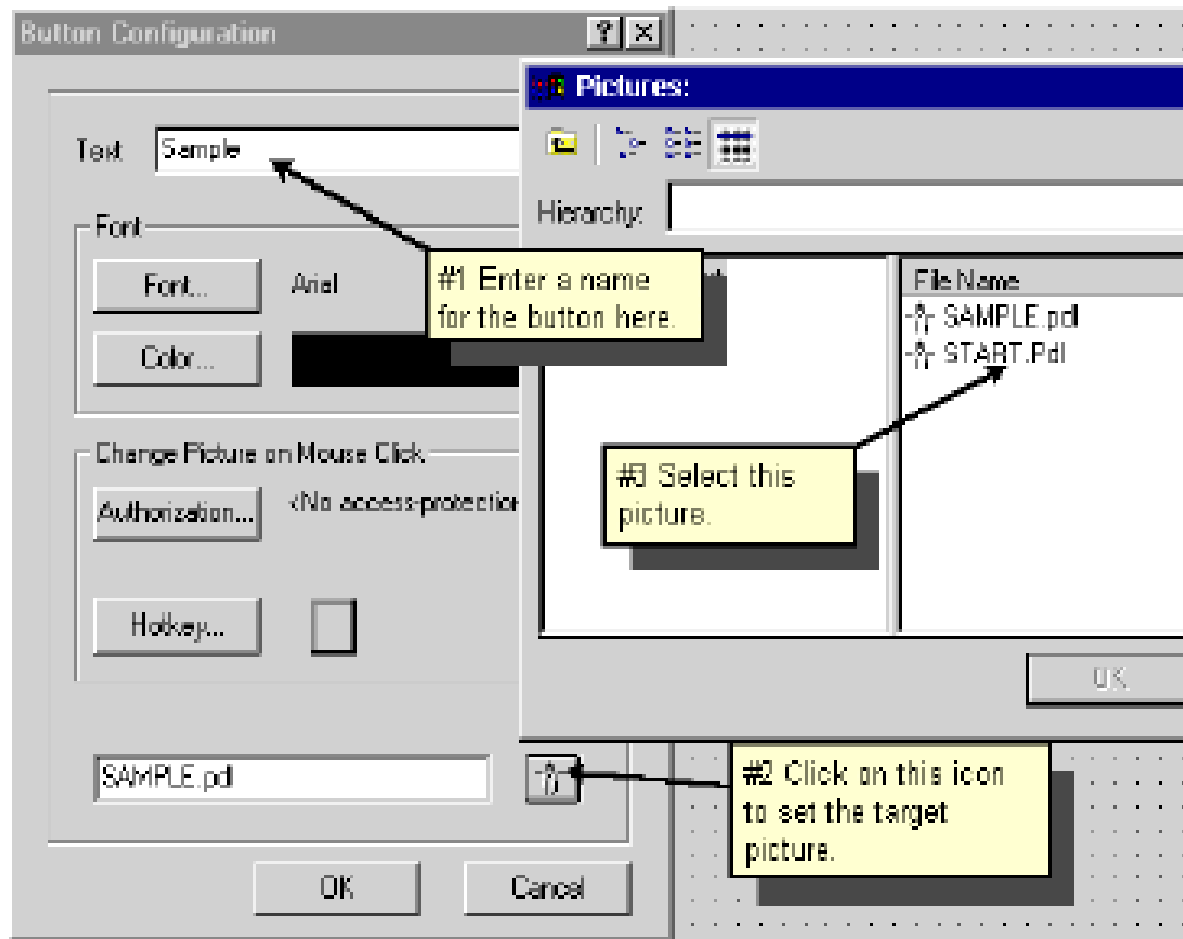
Chọn hai nút này để xem các hình mẫu

## Tạo nút nhấn

Chọn Button trong mục Windows Object bên Object Palette, kéo qua và đặt tại vị trí mong muốn trên màn hình soạn thảo



Gõ vào mục Text và chọn trang màn hình muốn chỉ tới

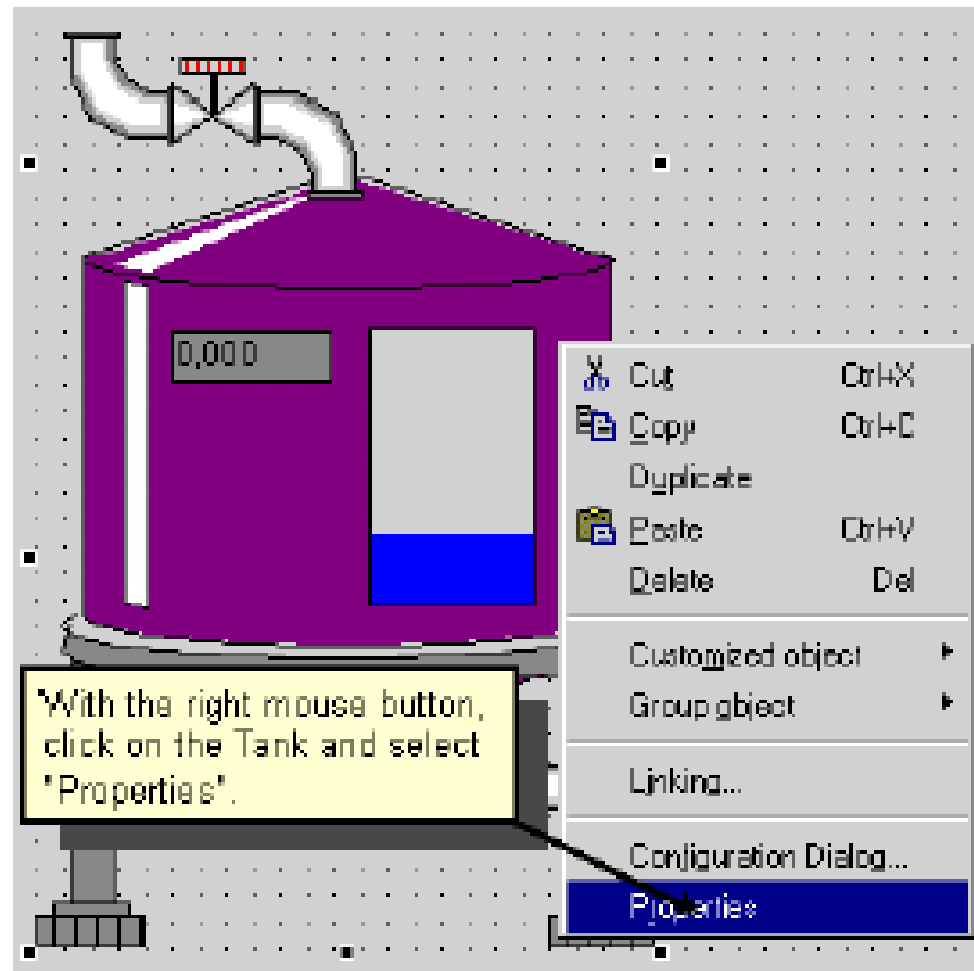


Thực hiện một nút nhấn tương tự như trên trang màn hình Sample.pdl



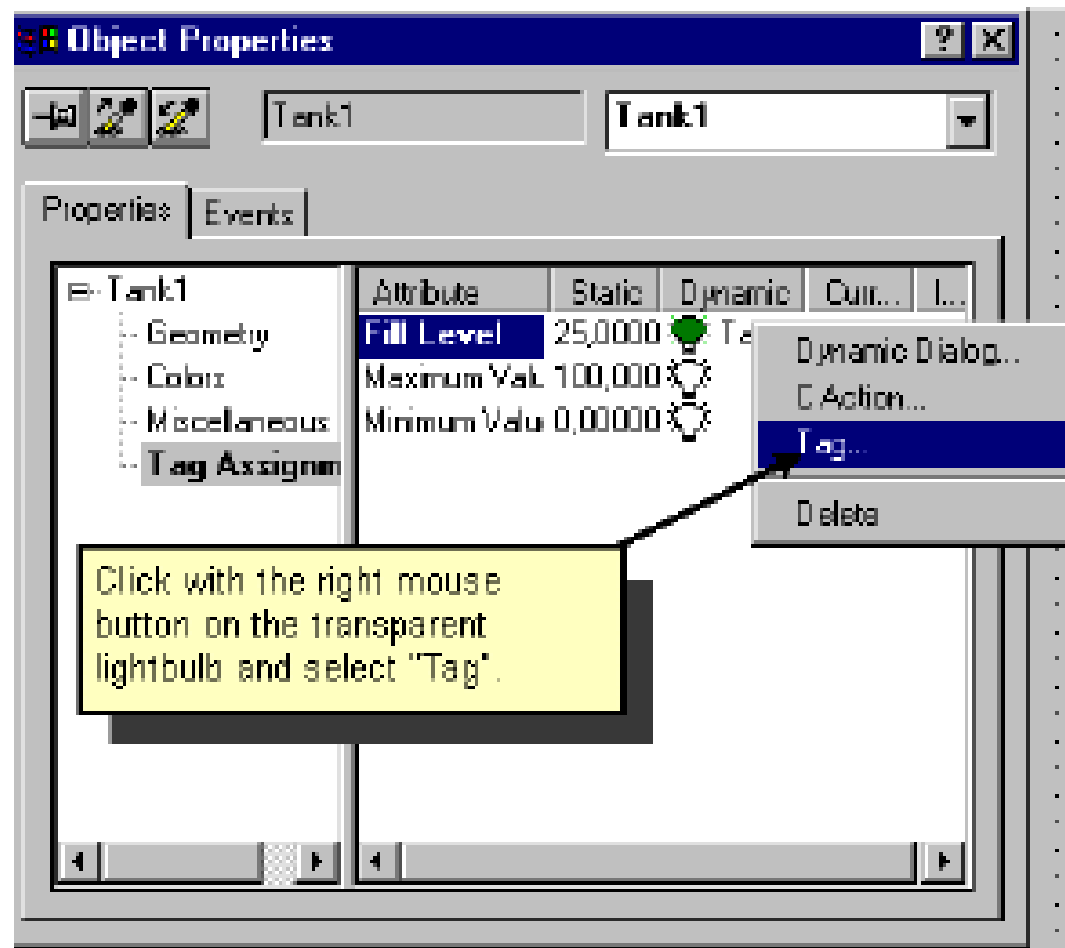
Tạo hình ảnh động cho đối tượng

Nhấp chuột phải vào đối tượng, chọn Properties

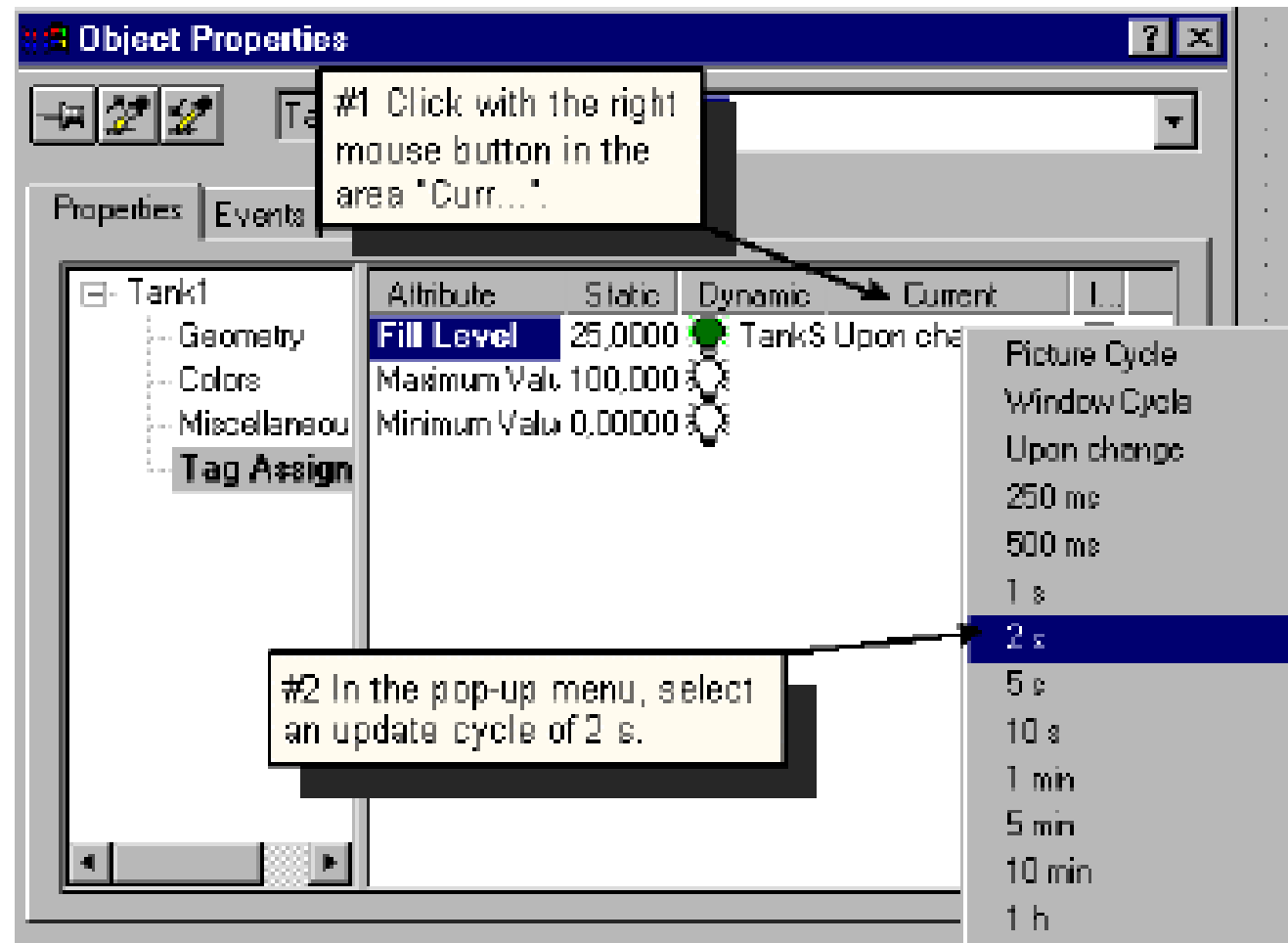


Chọn Tag Assignment trên tab Properties

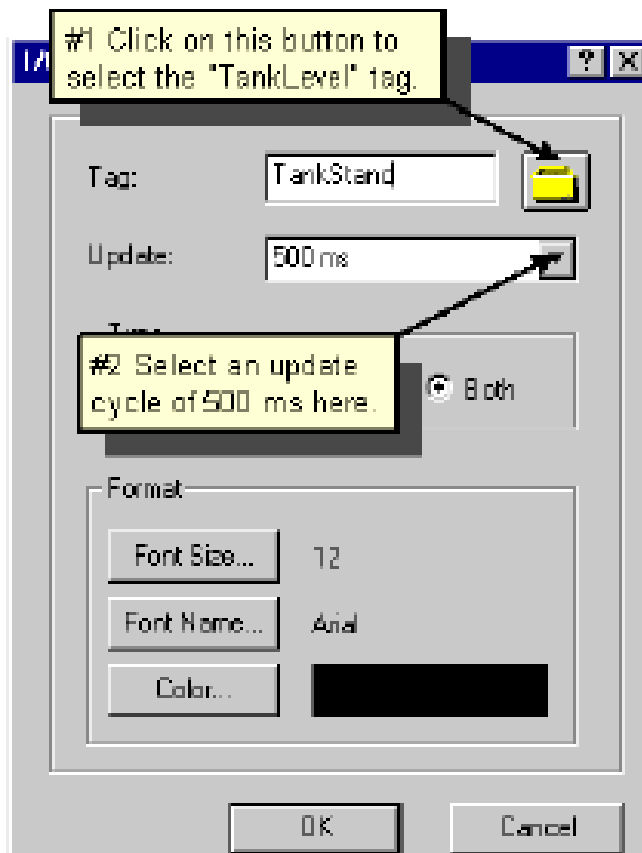
Nhấp chuột phải vào mục Dynamic → Chọn Tag muốn liên kết cho đối tượng



## Nhấp chuột phải vào mục Current chọn Upon Change



Tạo I/O Field để quan sát hay cài đặt dữ liệu cho đối tượng  
Kéo I/O Field có trong mục Smart Object bên của sổ Object  
Palette vào màn hình soạn thảo.

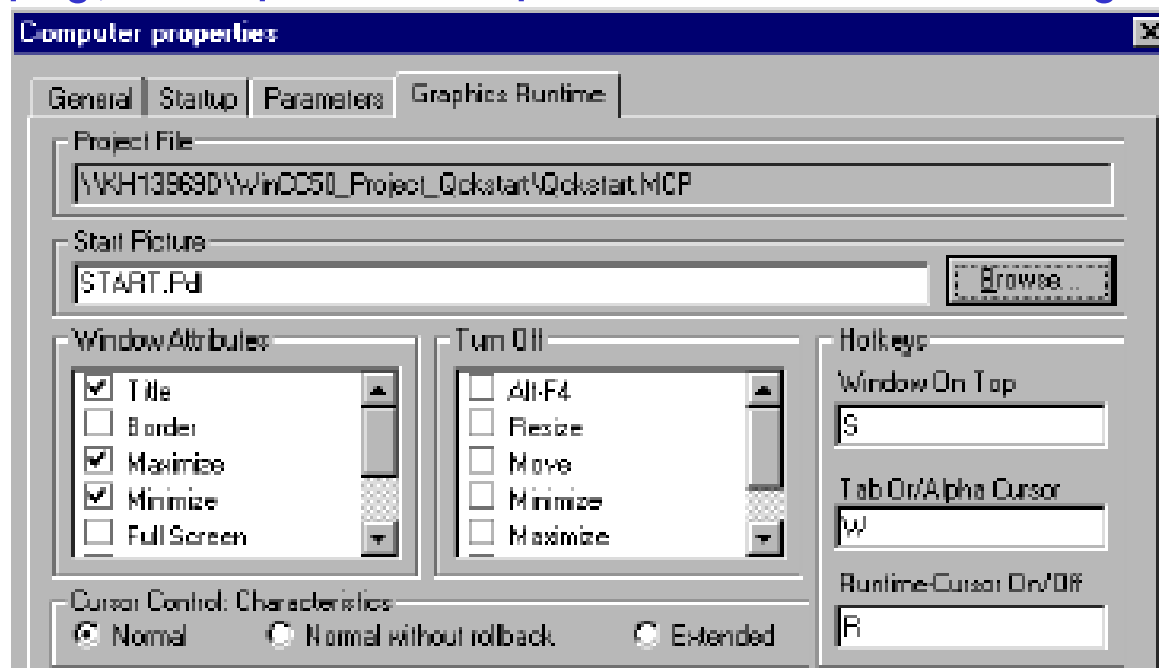


+ Chọn Tag và thời gian cập nhật từ của sổ I/O Field.

+ Chọn Save để lưu lại trang màn hình trước khi chạy Runtime

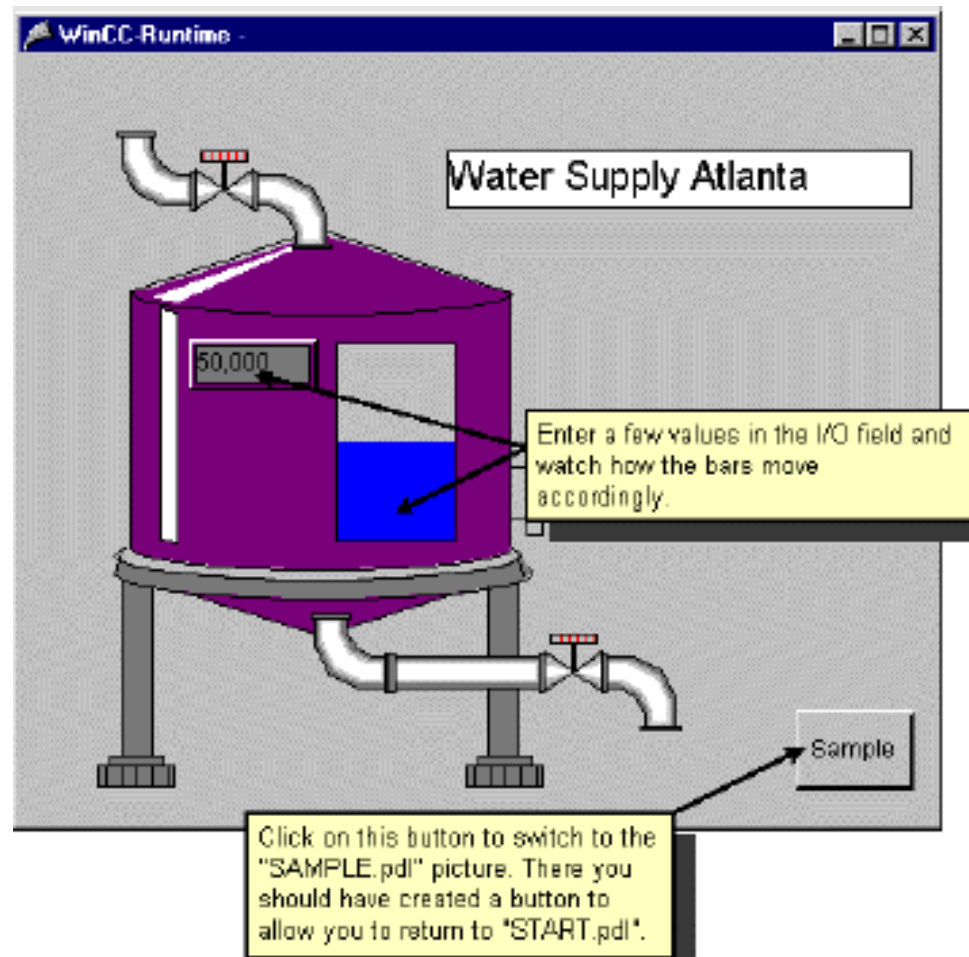
## 5. Cài đặt tham số khi chạy Runtime

Trên cửa sổ WinCC Explorer click chuột phải vào mục Computer, chọn Properties → Chọn Properties lần nữa. Trên tab Graphics Runtime chọn trang màn hình khởi động, và đặt các thuộc tính cho màn hình giao diện



Nhấp OK để thoát

Nhấn nút Activate  để chạy chương trình

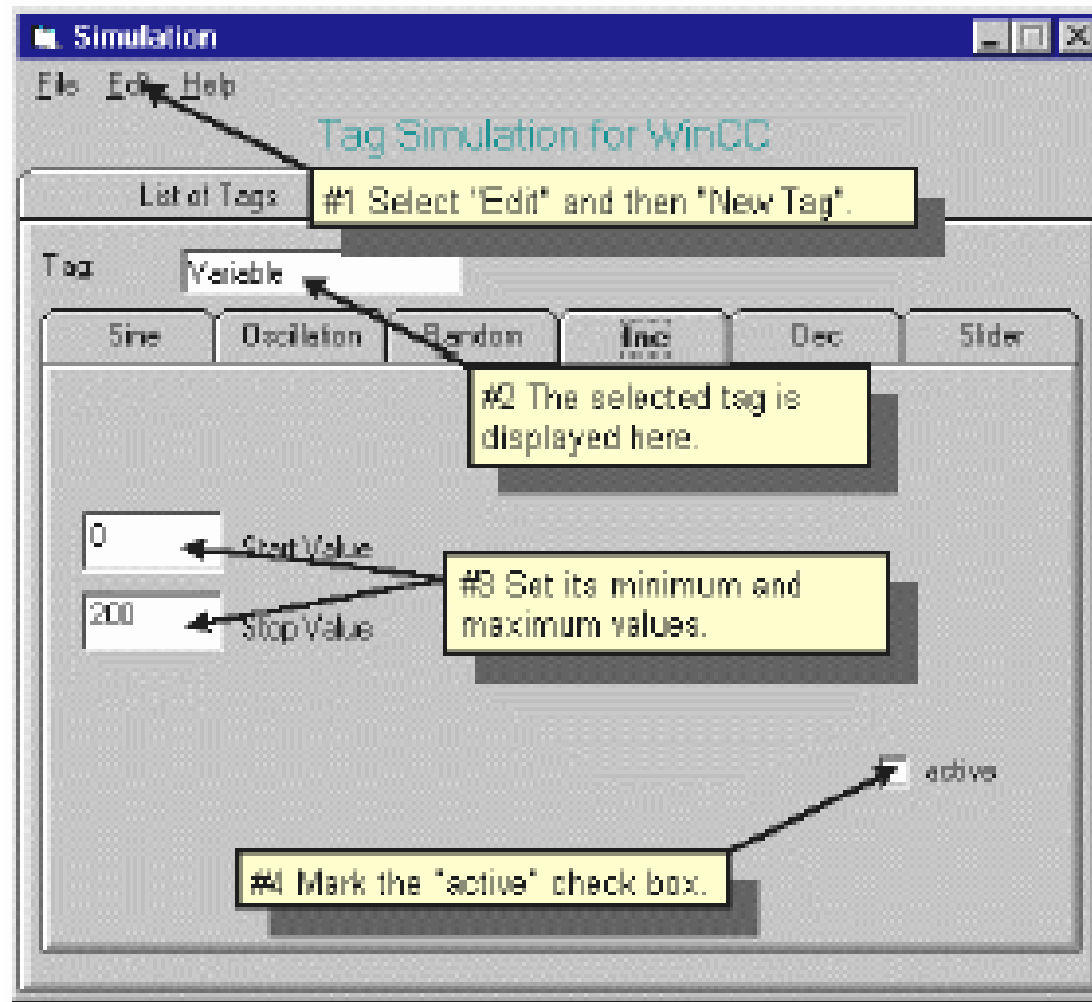


## Sử dụng chương trình mô phỏng các Tag trong WinCC

+ Từ màn hình windows, vào Start → Chọn Simatic → WinCC  
→ Tool → WinCC Variable Simulator.

+ Trên mục Properties, chọn New Tag từ menu Edit → Chọn Tag cần mô phỏng và cách thức thay đổi dữ liệu của Tag: Dạng Sin, Tăng dần, giảm dần, dao động.. → Chọn active.

+ Trở lại mục List of Tags chọn Start Simulation để bắt đầu mô phỏng.





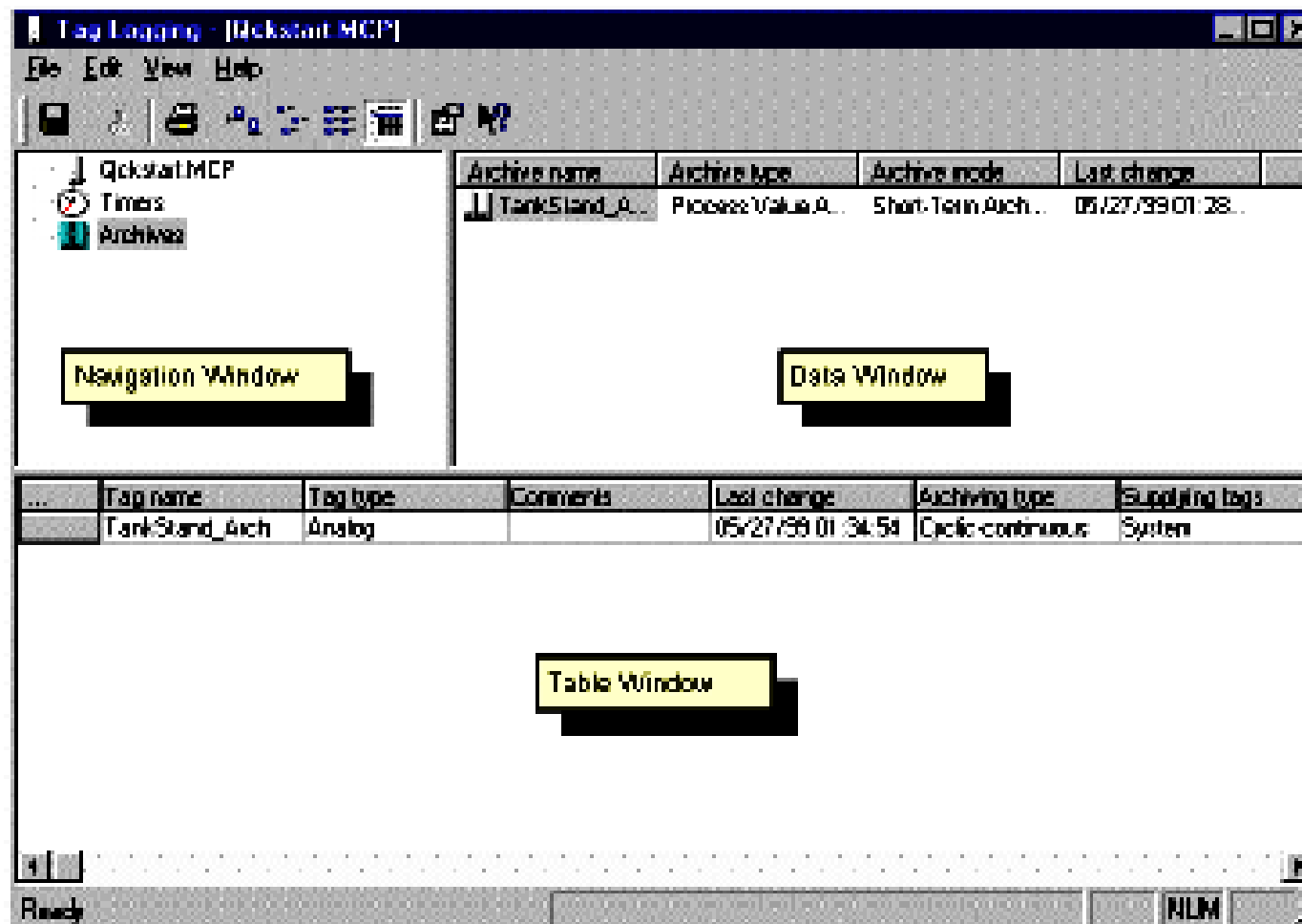
# Hiển thị thông số của quá trình sản xuất

## **Sử dụng Tag logging để hiển thị thông số của quá trình sản xuất qua các bước sau:**

1. Mở Tag Logging Editor
2. Đặt cấu hình cho Timer
3. Tạo dữ liệu lưu trữ với Archieve Wizard
4. Vẽ đồ thị hiển thị thông số của quá trình sản xuất
5. Tạo một bảng báo cáo
6. Cài đặt tham số Runtime
7. Chạy chương trình

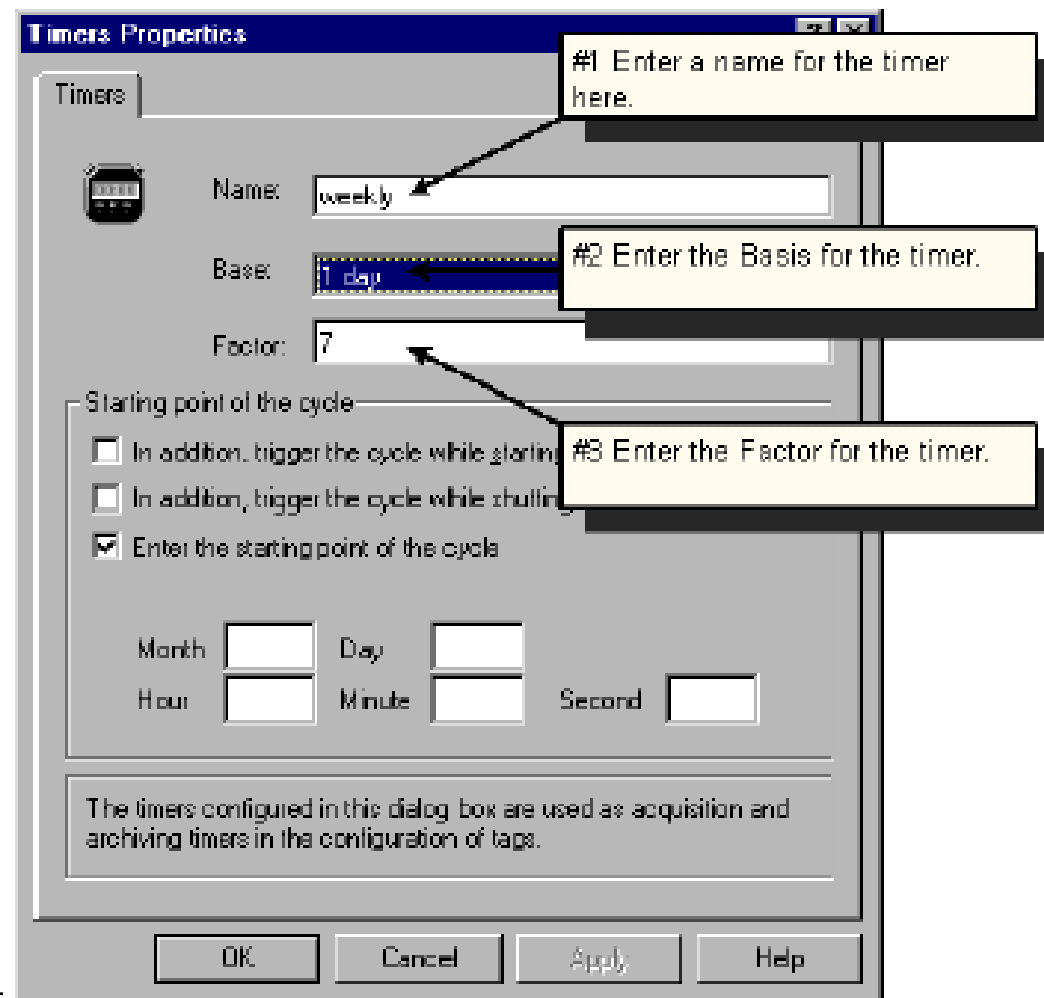
# 1. Mở Tag Logging Editor

Nhấn chuột phải vào Tag Logging trên cửa sổ WinCC Explorer → Open



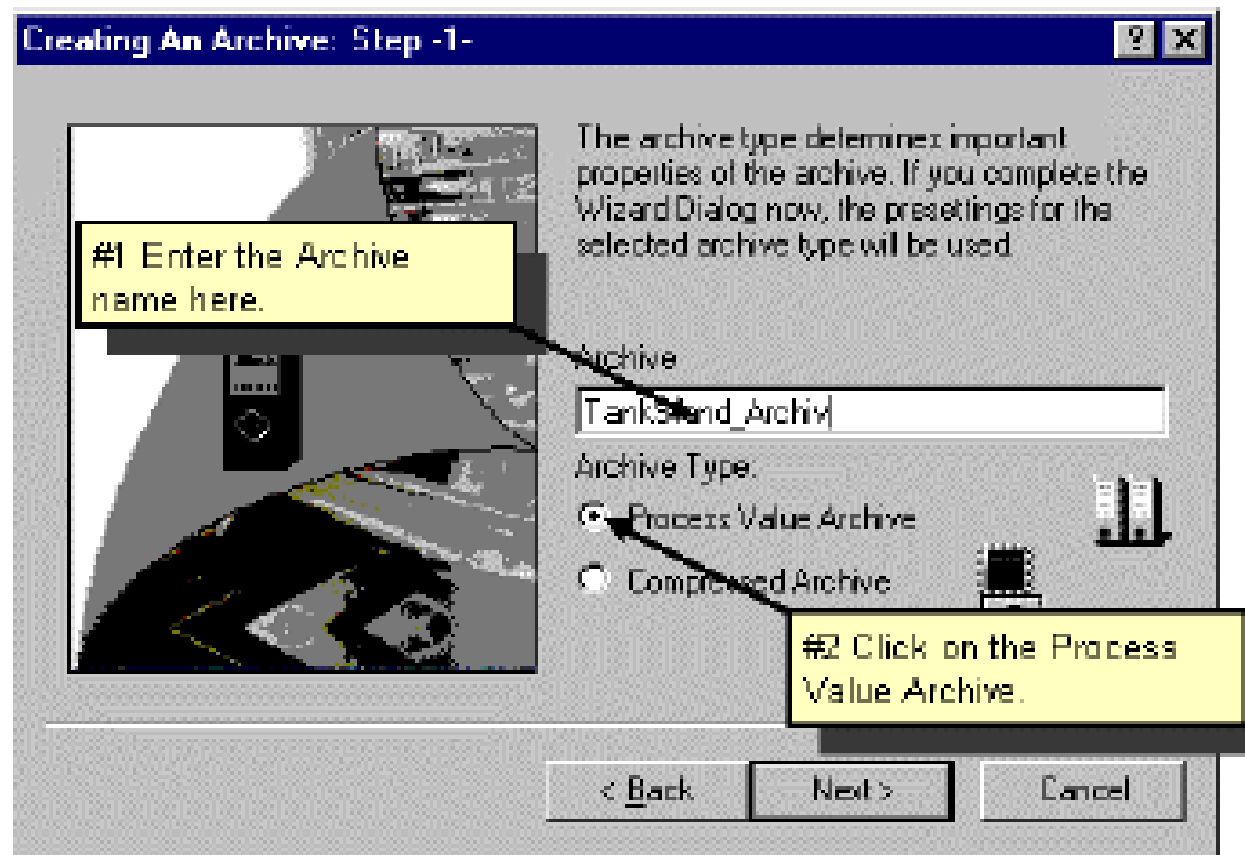
## 2. Đặt cấu hình cho Timer

Nhấn chuột phải vào timer, chọn New và cài đặt thời gian lấy dữ liệu

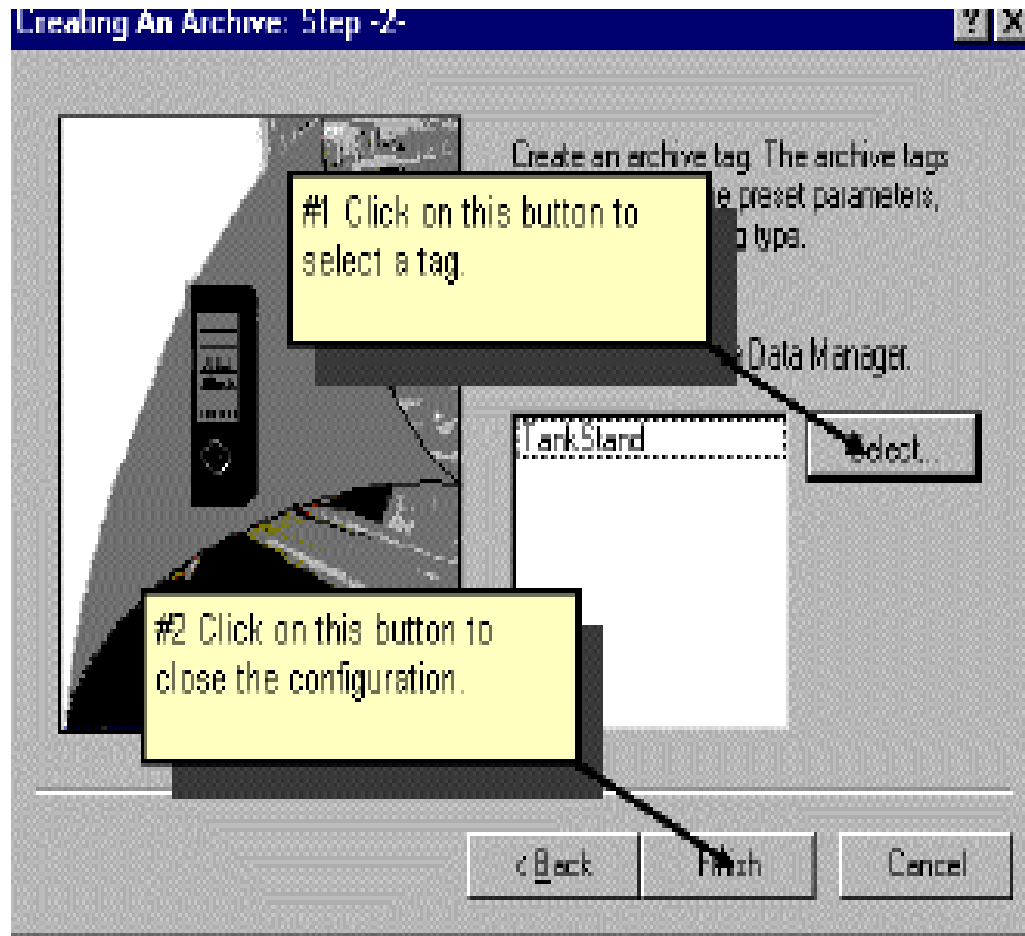


### 3. Tạo dữ liệu lưu trữ với Archive Wizard

Nhấp chuột phải vào Archieve , chọn Archieve Wizard, nhấn Next và thực hiện như các bước dưới đây:



## Nhấn Select để chọn Tag cần lưu trữ



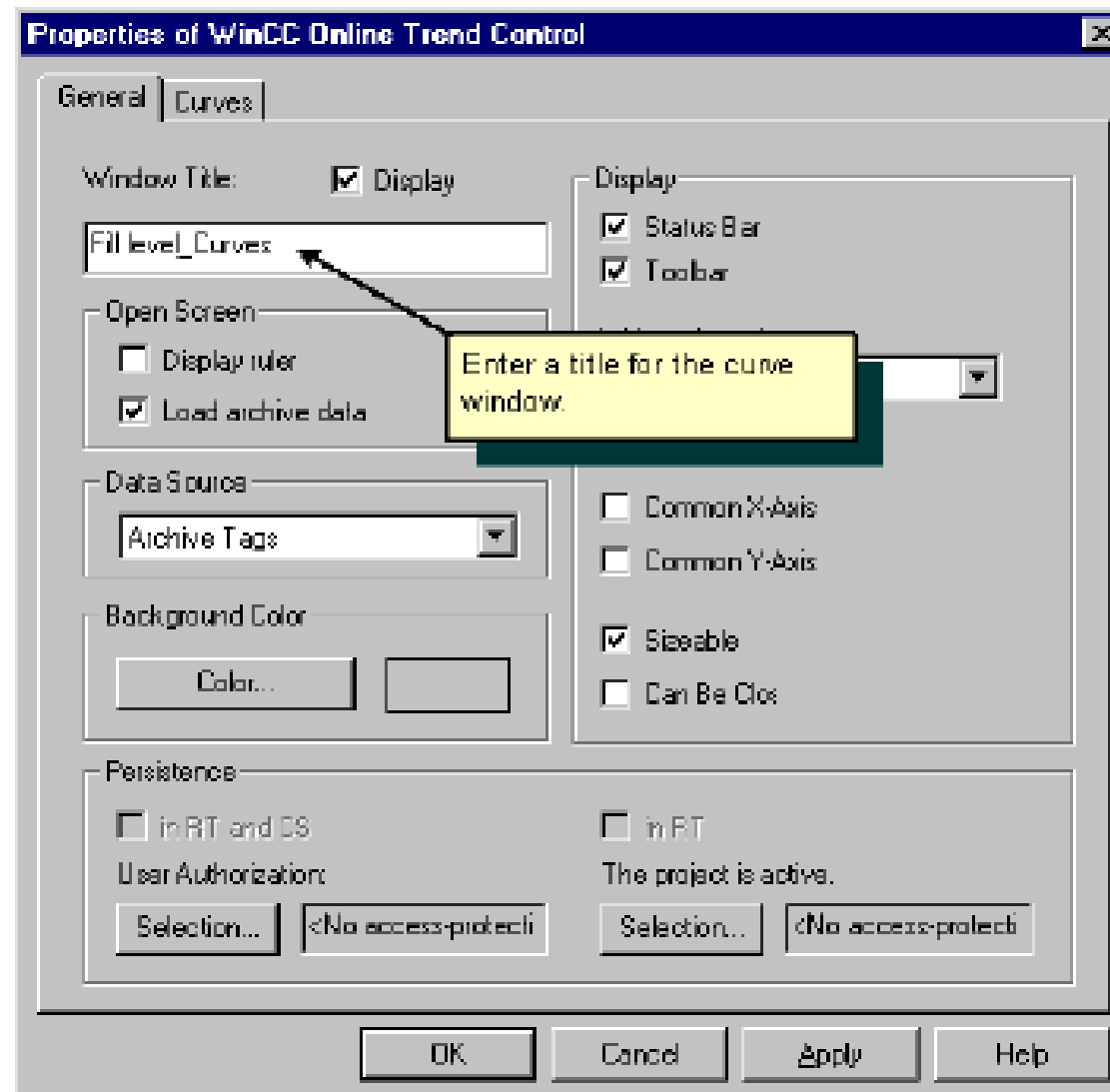
Nhấn nút Save trước khi đóng của sổ Archive

## 4. Vẽ đồ thị hiển thị thông số của quá trình sản xuất

+ Có thể tạo một trang màn hình mới.

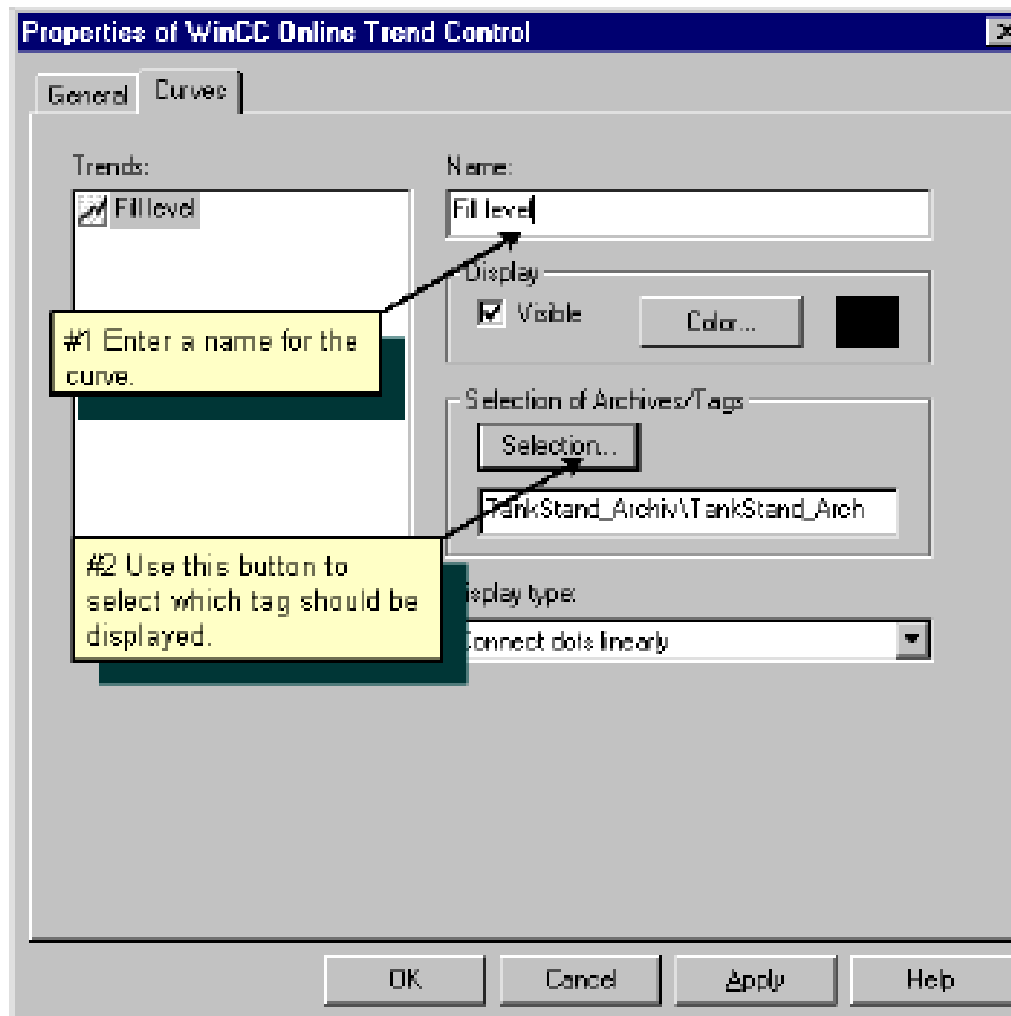
+ Trong cửa sổ Graphics Designer, chọn WinCC Online Trend Control trên Tab Control bên cửa sổ Object Palette và kéo vào màn hình soạn thảo.

+ Có thể gõ tên đồ thị và cài đặt các thuộc tính cho đồ thị từ cửa sổ sau:



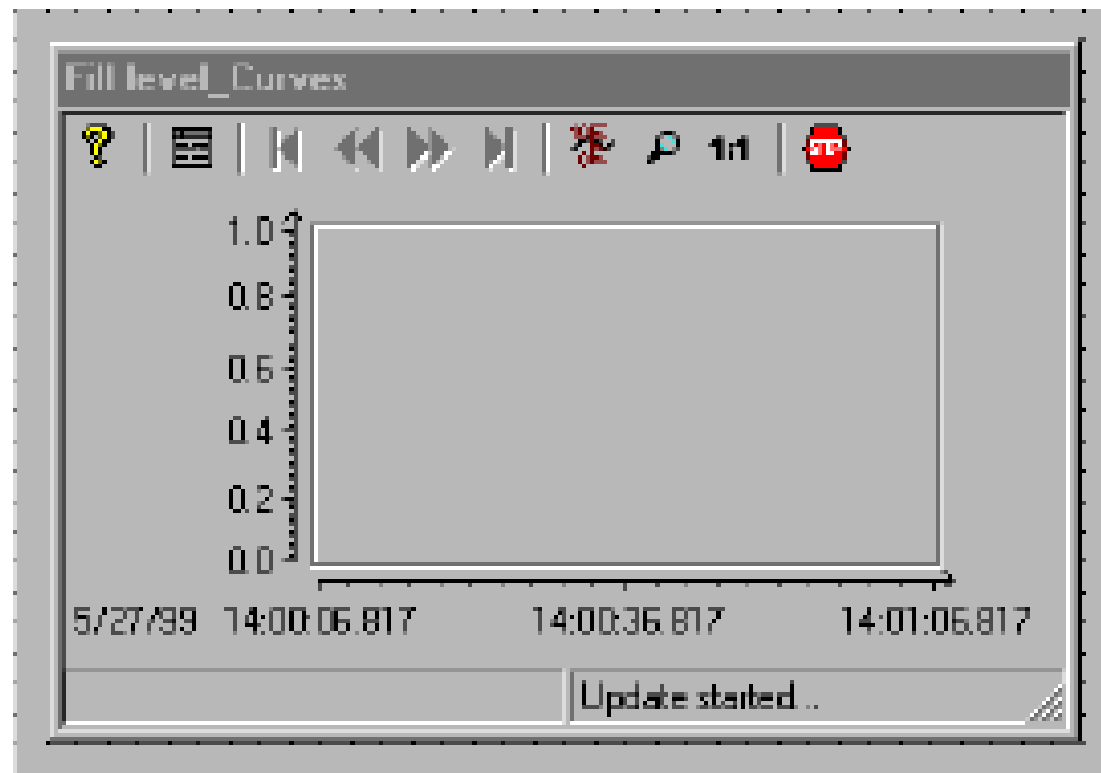


Chuyển sang Tab Curve để chọn tên Tag mà cần vẽ đồ thị như sau



Nhấn OK để lưu lại cài đặt này và thoát

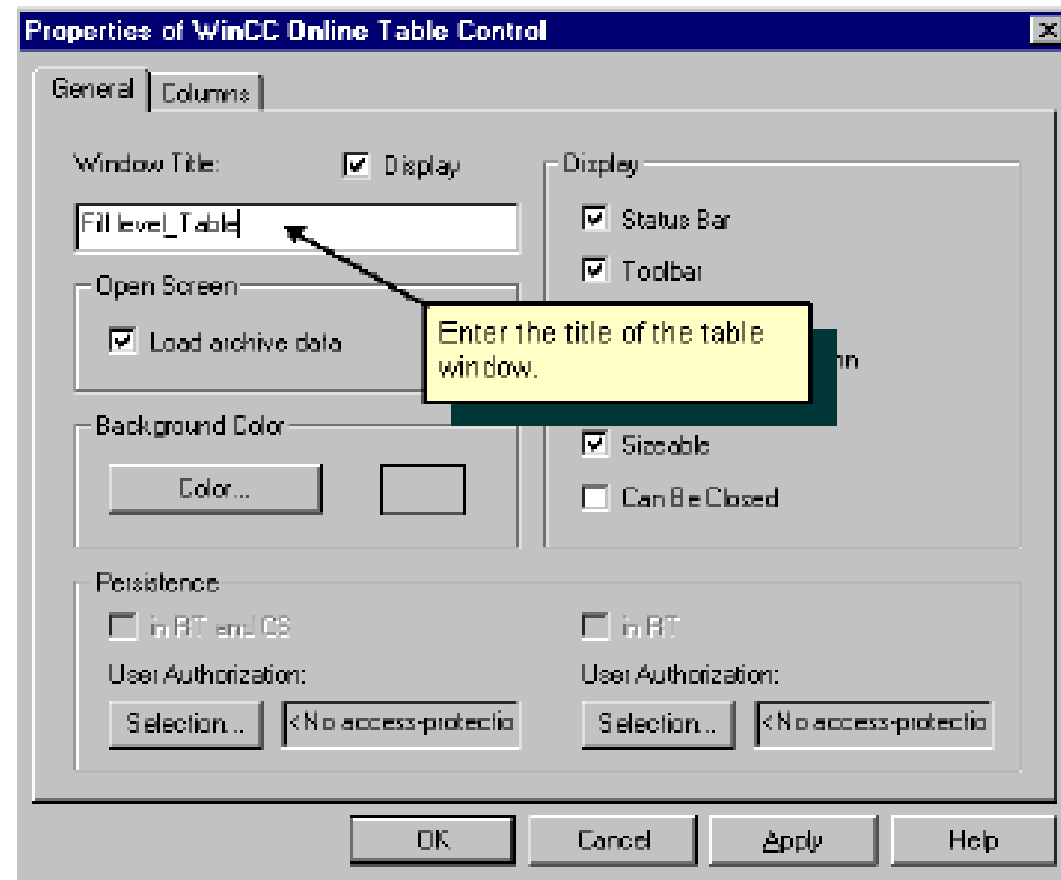
Nhấn CTrl + Double Click để xem thử hình ảnh của đồ thị lúc chạy Runtime, có thể sửa lại các thông số của đồ thị sao cho hợp lý.



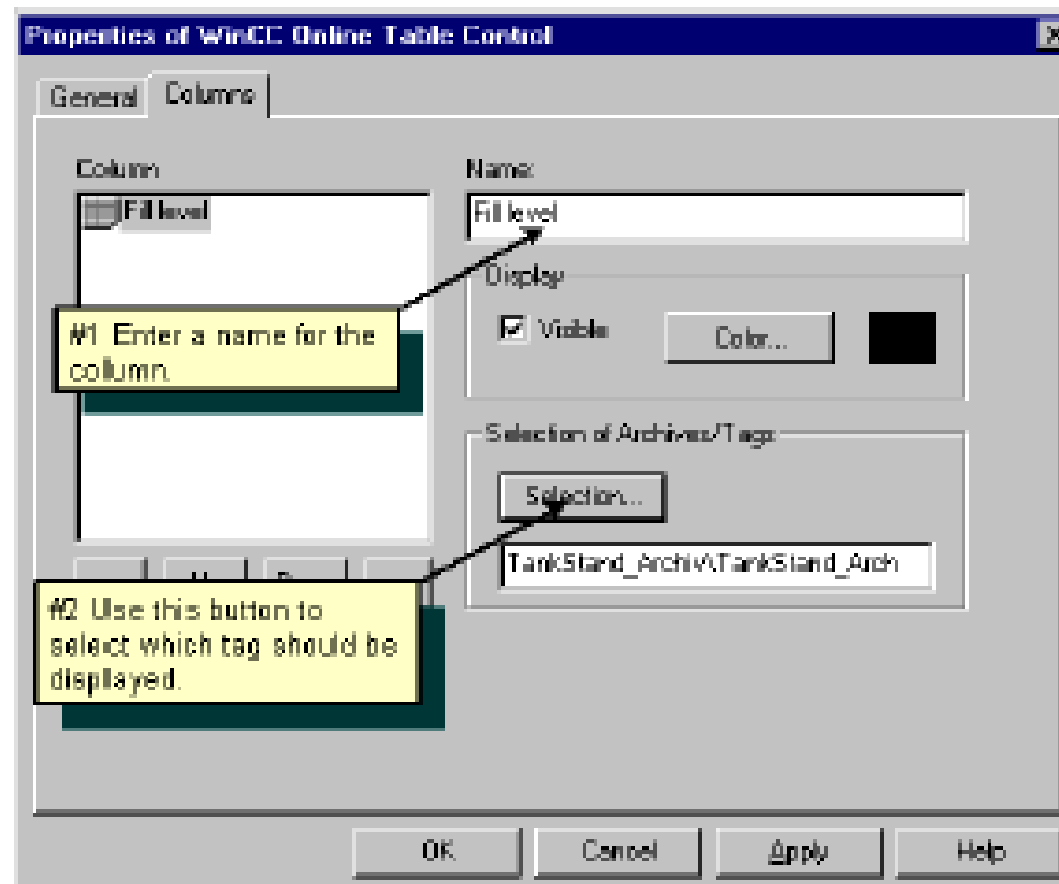
## 5. Tạo bảng báo cáo

+ Chọn WinCC Online Table Control có trong Tab “Control” bên cửa sổ Object Palette và đặt vào trang màn hình soạn thảo.

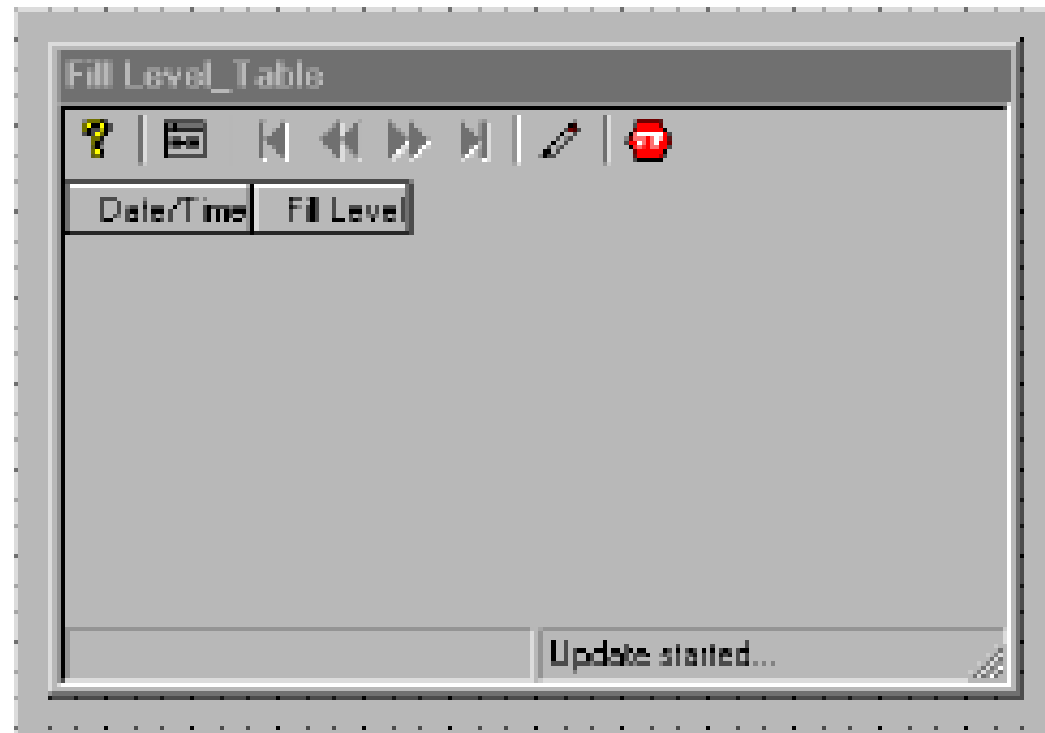
+ Điền tên và các thông số của bảng từ cửa sổ hiện ra.



Chuyển qua tab “Column” để gõ tên của giá trị hiển thị trên bảng và chọn Tag cần quan sát như hình sau:

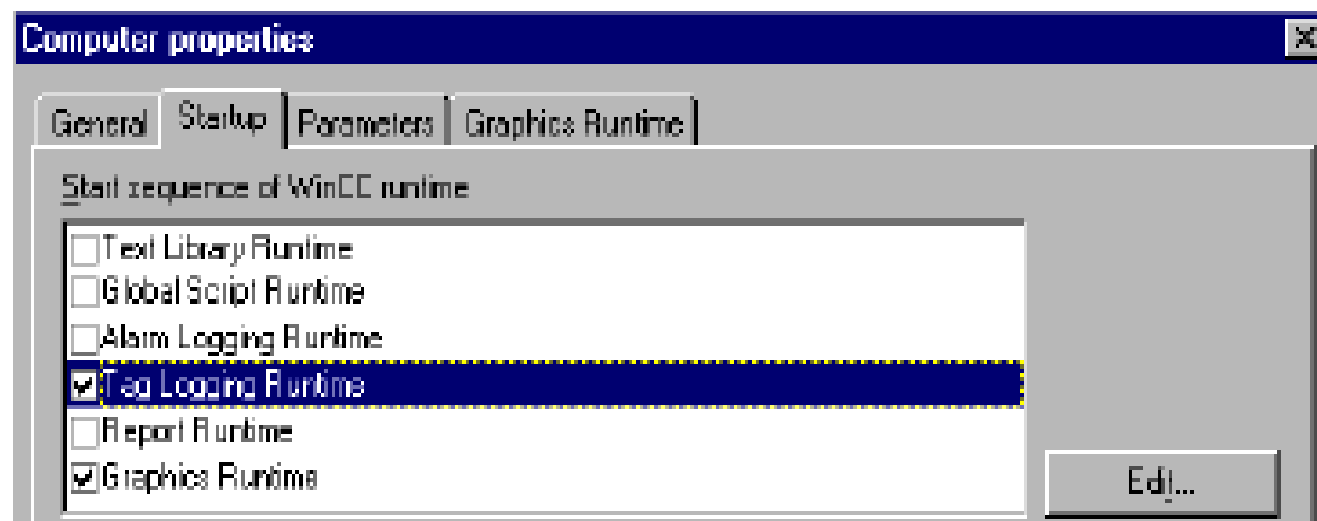


Có thể nhấn Ctrl+ Double Click để quan sát trước bảng thông số lúc chạy Runtime



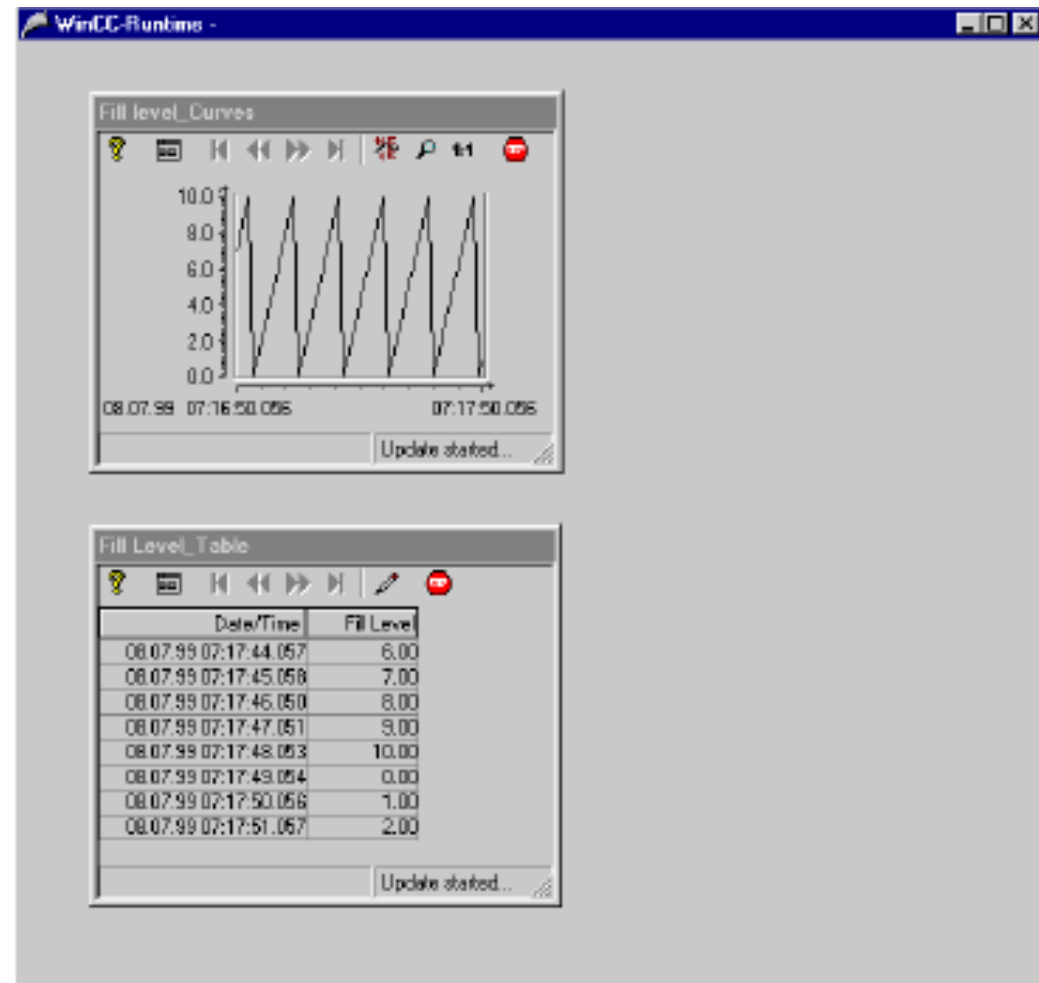
## 6. Cài đặt tham số khi chạy Runtime

- + Trên cửa sổ WinCC Explorer click chuột phải vào mục Computer, chọn Properties → Chọn Properties lần nữa.
- + Trên tab “Start up”, chọn tag Logging Runtime

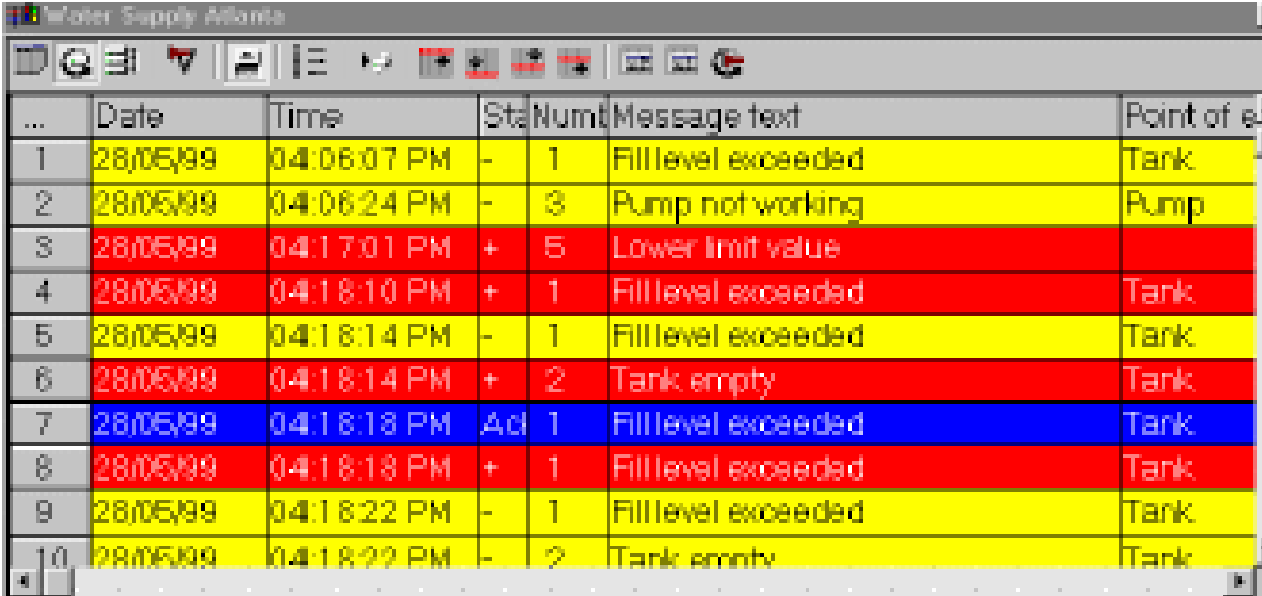


+ Nhấn nút Activate  để chạy chương trình.

+ Có thể kích hoạt chương trình mô phỏng để xem hình ảnh của đồ thị và bảng thông số khi chạy Runtime



# Thiết lập các cảnh báo và thông báo lỗi



...	Date	Time	Sta	Num	Message text	Point of e
1	28/05/99	04:06:07 PM	-	1	Filllevel exceeded	Tank
2	28/05/99	04:06:24 PM	-	3	Pump not working	Pump
3	28/05/99	04:17:01 PM	+	5	Lower limit value	
4	28/05/99	04:18:10 PM	+	1	Filllevel exceeded	Tank
5	28/05/99	04:18:14 PM	-	1	Filllevel exceeded	Tank
6	28/05/99	04:18:14 PM	+	2	Tank empty	Tank
7	28/05/99	04:18:18 PM	Act	1	Filllevel exceeded	Tank
8	28/05/99	04:18:18 PM	+	1	Filllevel exceeded	Tank
9	28/05/99	04:18:22 PM	-	1	Filllevel exceeded	Tank
10	28/05/99	04:18:22 PM	-	2	Tank empty	Tank



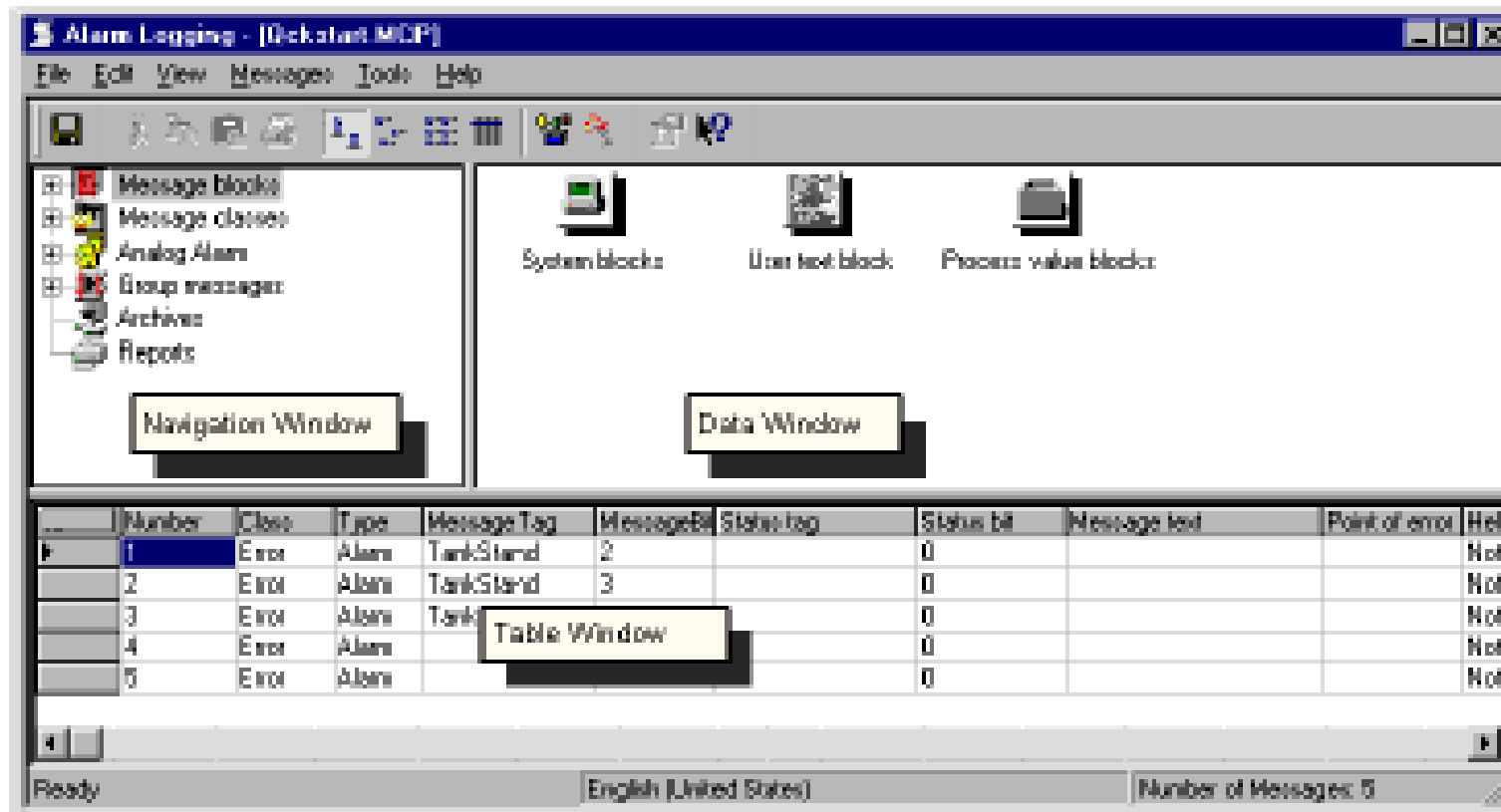
**Sử dụng Alarm Logging để thiết lập các cảnh báo và thông báo lỗi.**

**Các trình tự có thể thực hiện như sau (tuỳ theo yêu cầu) :**

- 1. Mở cửa sổ soạn thảo Alarm Logging Editor**
- 2. Khởi động System Wizard**
- 3. Cài đặt Message Text**
- 4. Dùng Message Class để đặt màu cho các message**
- 5. Cài đặt Limit value**
- 6. Tạo Message Window trên màn hình soạn thảo**
- 7. Thiết lập các tham số Runtime**
- 8. Chạy chương trình**

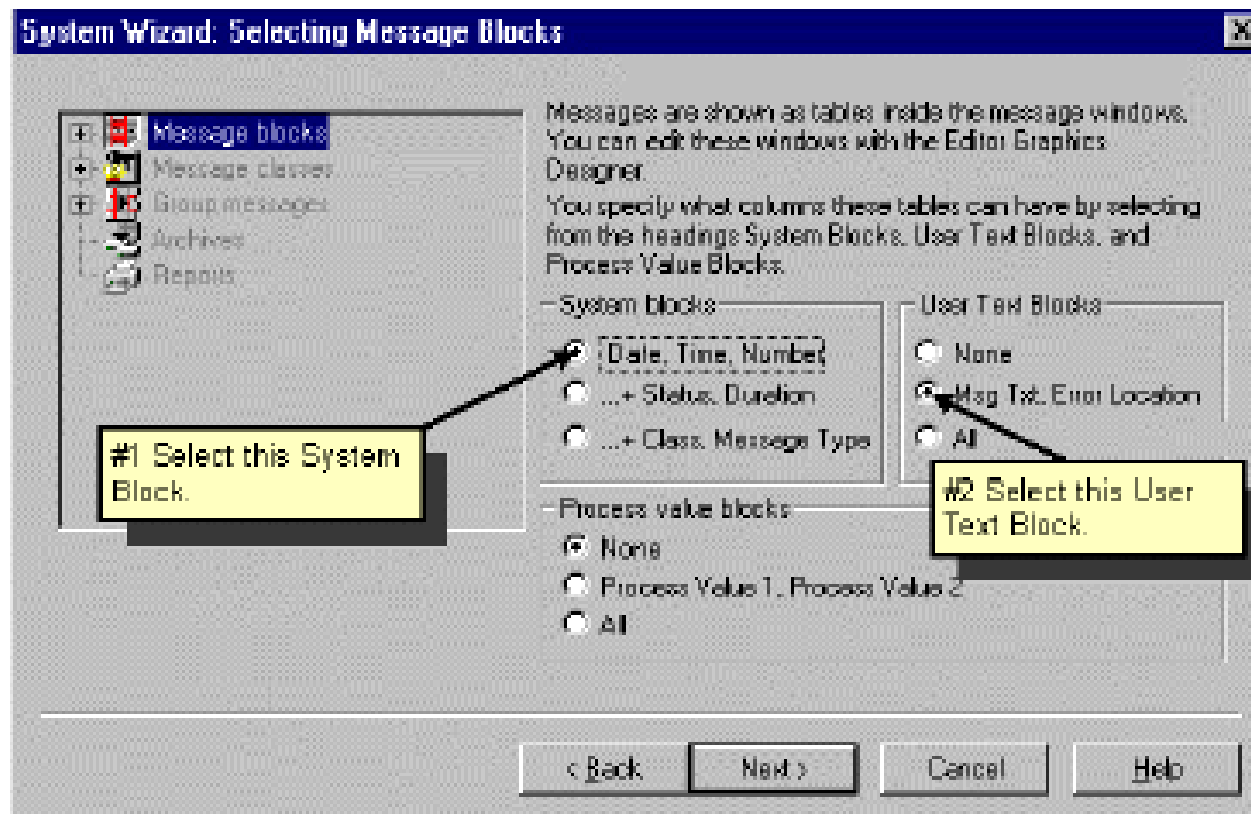
# 1. Mở cửa sổ Alarm Logging

Nhấn chuột phải vào Alarm Logging, chọn Open

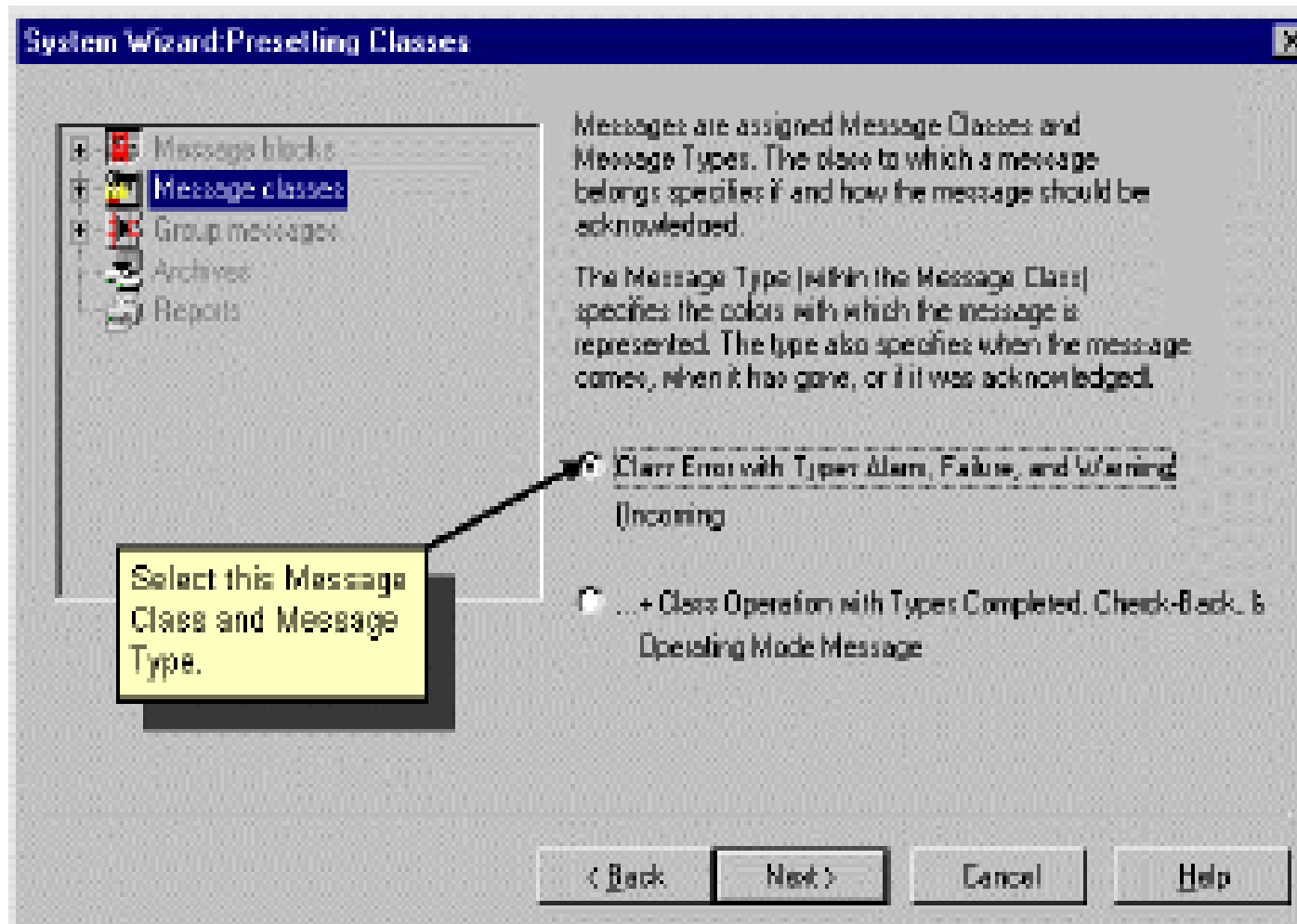


## 2. Khởi động System Wizard

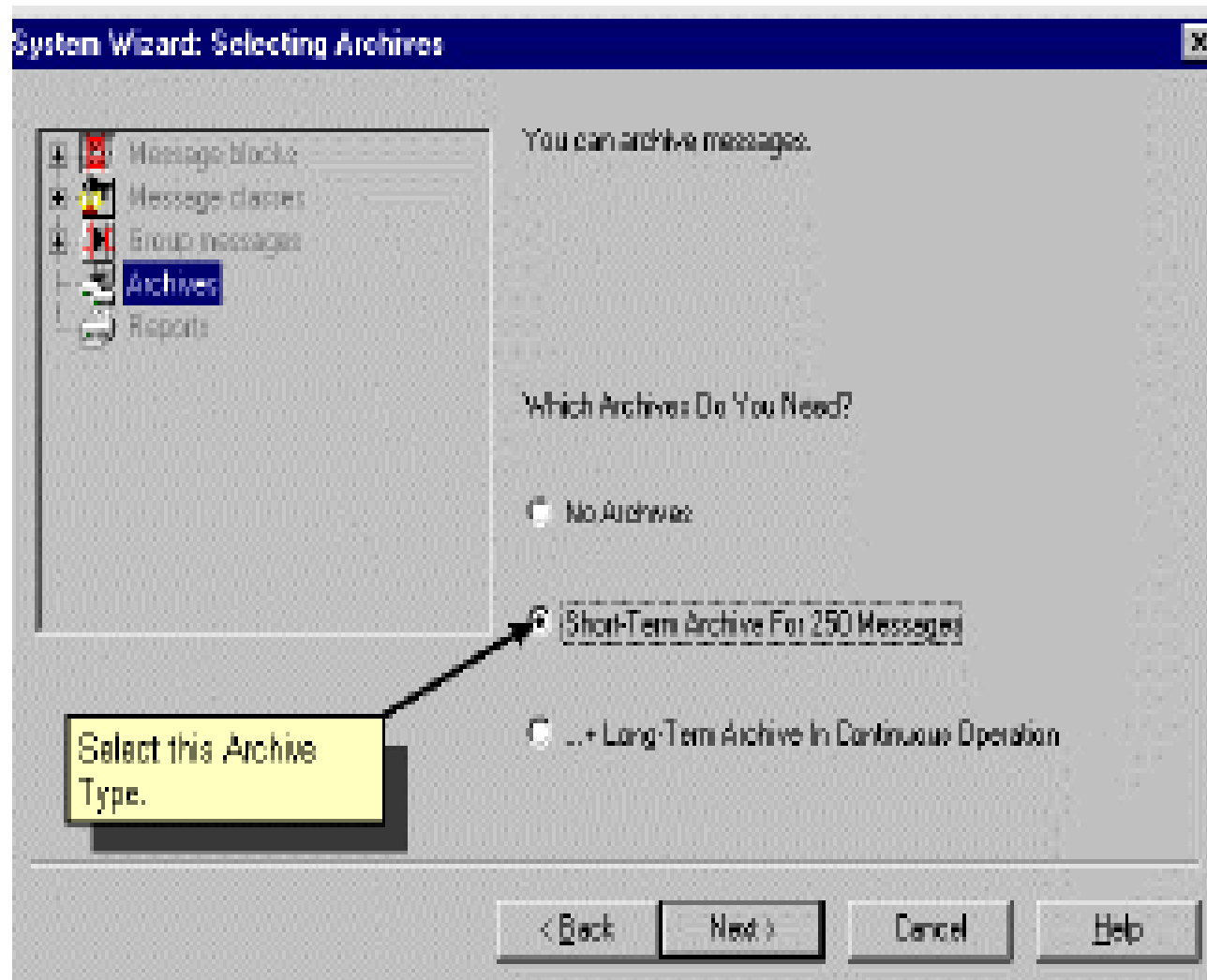
Vào File → Select Wizard → chọn System Wizard → Next.  
Chọn các thông số như những hình sau đây:



Nhấn Next



Nhấn Next

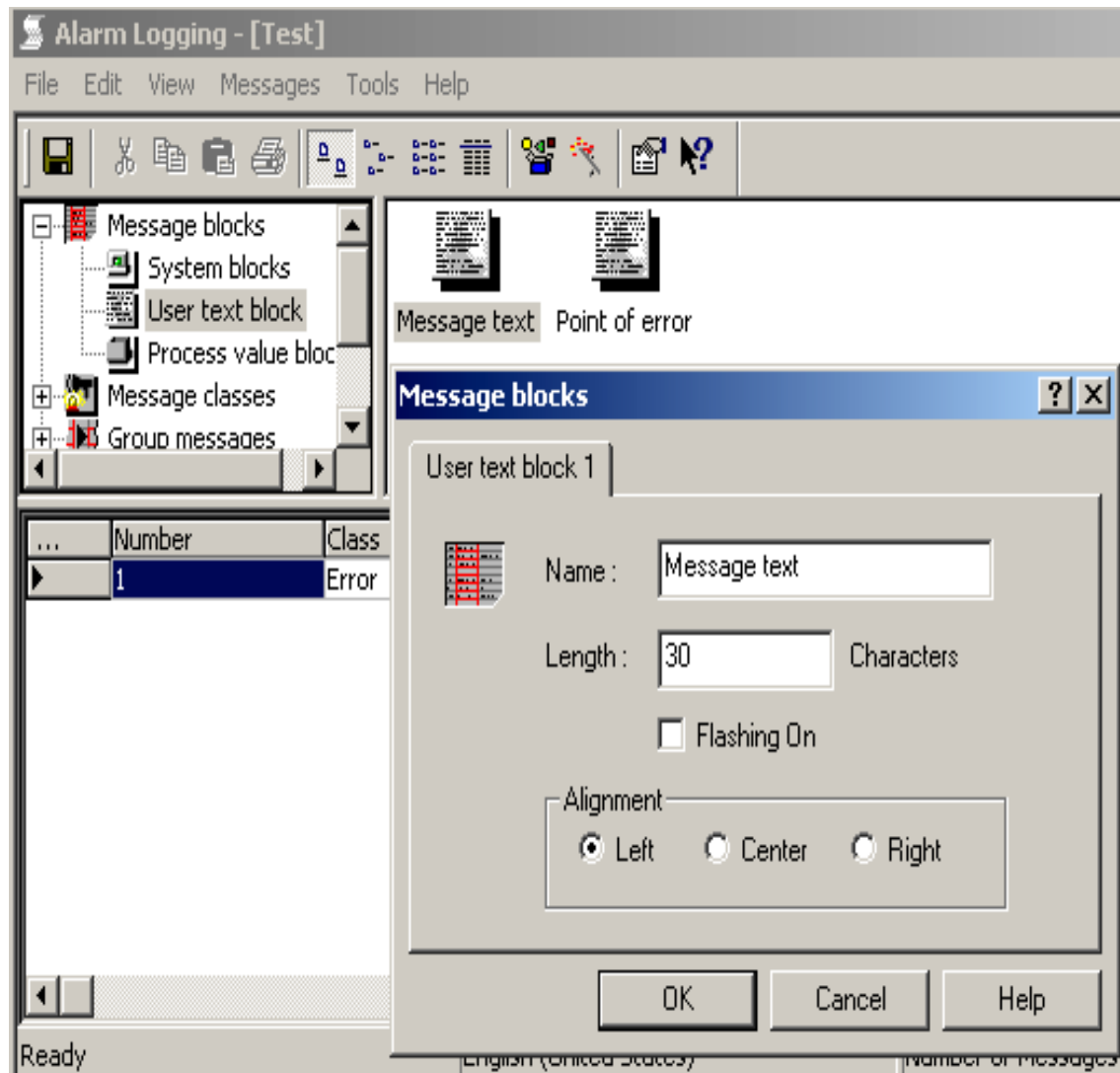


Nhấn Next.  
Trên bảng tóm tắt nhấn Apply

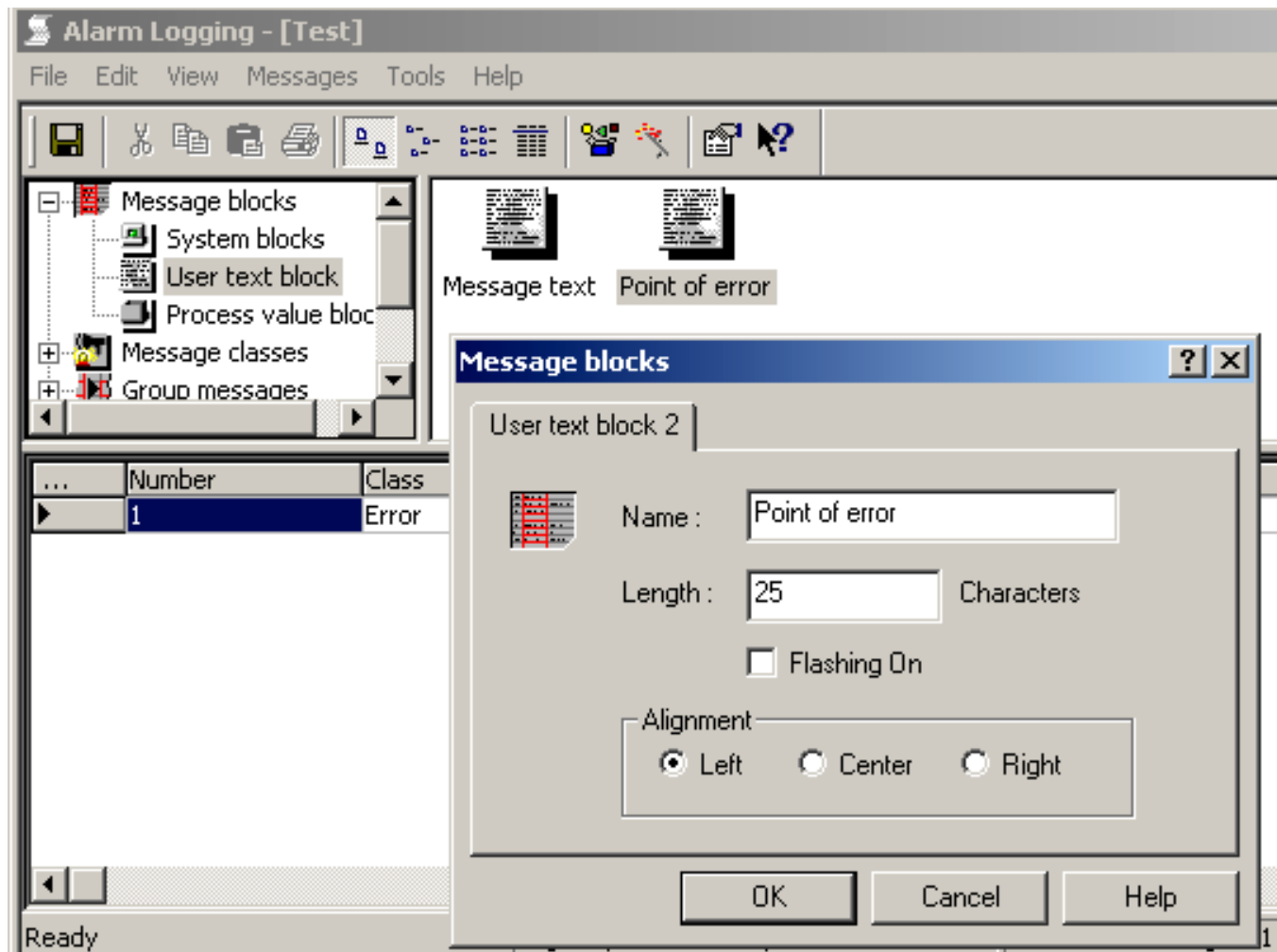
### 3. Cài đặt Message Text

+ Nhấn chuột phải vào dấu + ở mục Message Blocks, chọn User Text Blocks.

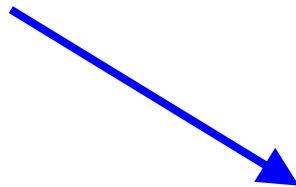
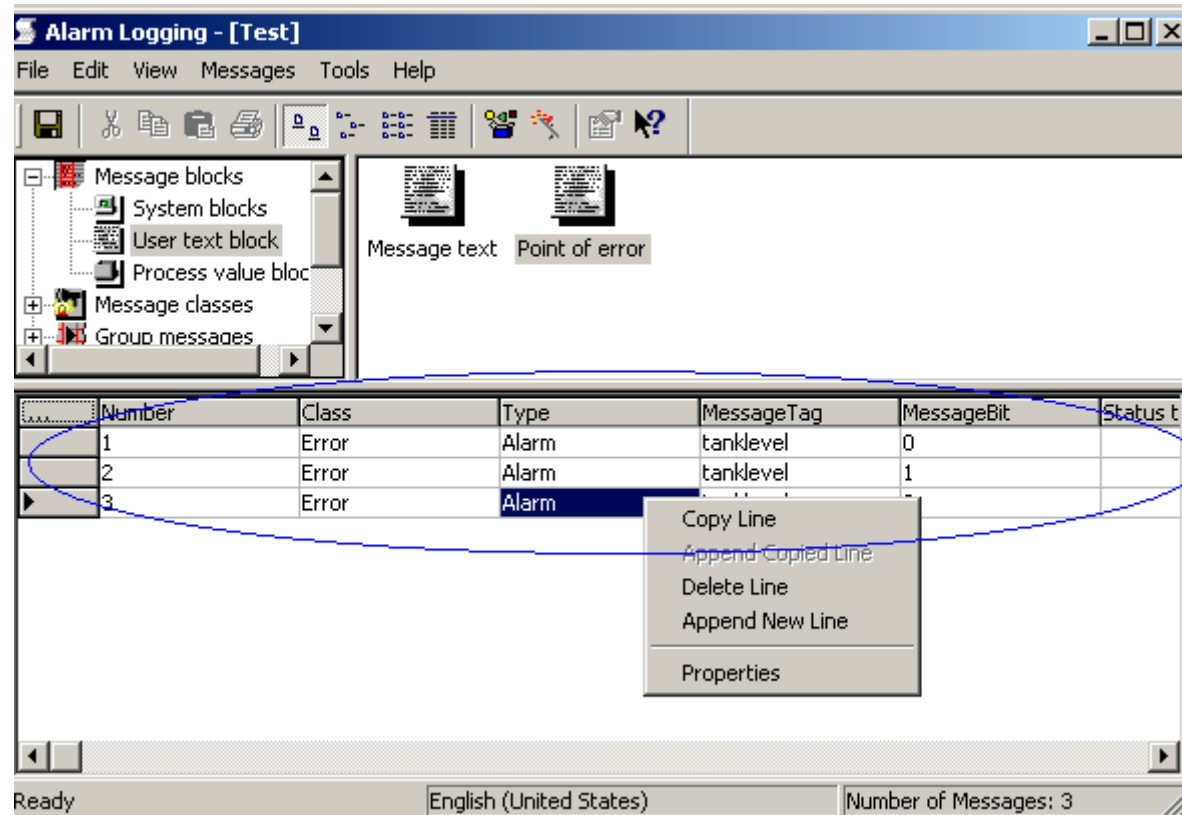
+ Tiếp theo nhấn chuột phải vào Message Text bên cửa sổ Data Window, chọn Properties và gõ vào chiều dài của Message Text



## Thực hiện tương tự với Point of Error



Cài đặt cho các Message từ các hàng này





Number	Class	Type	MessageTag	MessageBit	Status t
1	Error	Alarm	tanklevel	0	
2	Error	Alarm	tanklevel	1	
3	Error	Alarm			



## Double Click chuột vào :

- Các hàng chọn Append New Line để tạo một hàng thông báo mới.
- Message Tag để chọn Tag báo lỗi.
- Message Bit để chọn Bit lỗi trong có trong Message Tag.
- Message Text để điền câu thông báo lỗi
- Point of error để chỉ vị trí bị lỗi trong hệ thống sản xuất.



The screenshot shows a software window titled 'Reports' containing a table with the following columns: Number, Class, Type, Message Tag, Message Bit, Status tag, Status bit, Message text, Point of error, and Help. The table contains three rows of data. Callouts with arrows point to specific cells in the table, explaining the function of double-clicking:

- #1** Double-click here to select the message tags. (Points to the 'Message Tag' column of the first row)
- #2** Double-click here to set the message bits. (Points to the 'Message Bit' column of the first row)
- #3** Double-click here to enter the message text. (Points to the 'Message text' column of the first row)
- #4** Double-click here to enter the point of error. (Points to the 'Point of error' column of the first row)

At the bottom of the window, there is a status bar with the text 'Ready', 'English (United States)', and 'Number of Messages: 5'.

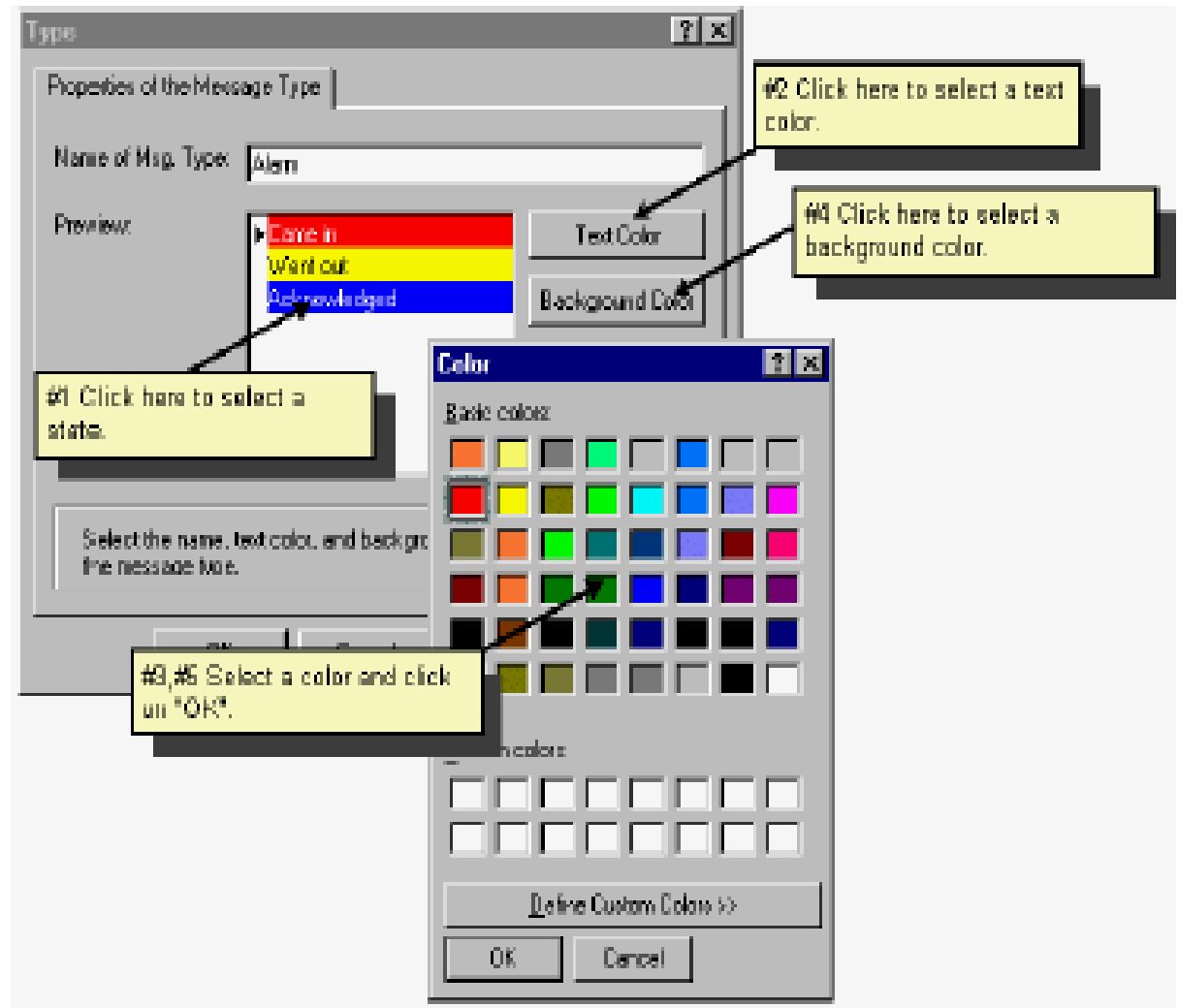
Number	Class	Type	Message Tag	Message Bit	Status tag	Status bit	Message text	Point of error	Help
1	Error	Alarm	TankStand	2		0			Not
2	Error	Alarm	TankStand	3		0			Not
			TankStand						Not
									Not

## 4. Cài đặt màu cho các Message

- Click chuột vào dấu + ở mục Message Class trong cửa sổ Alarm Logging.

- Click vào Error và Click phải chuột vào Alarm, chọn Properties.

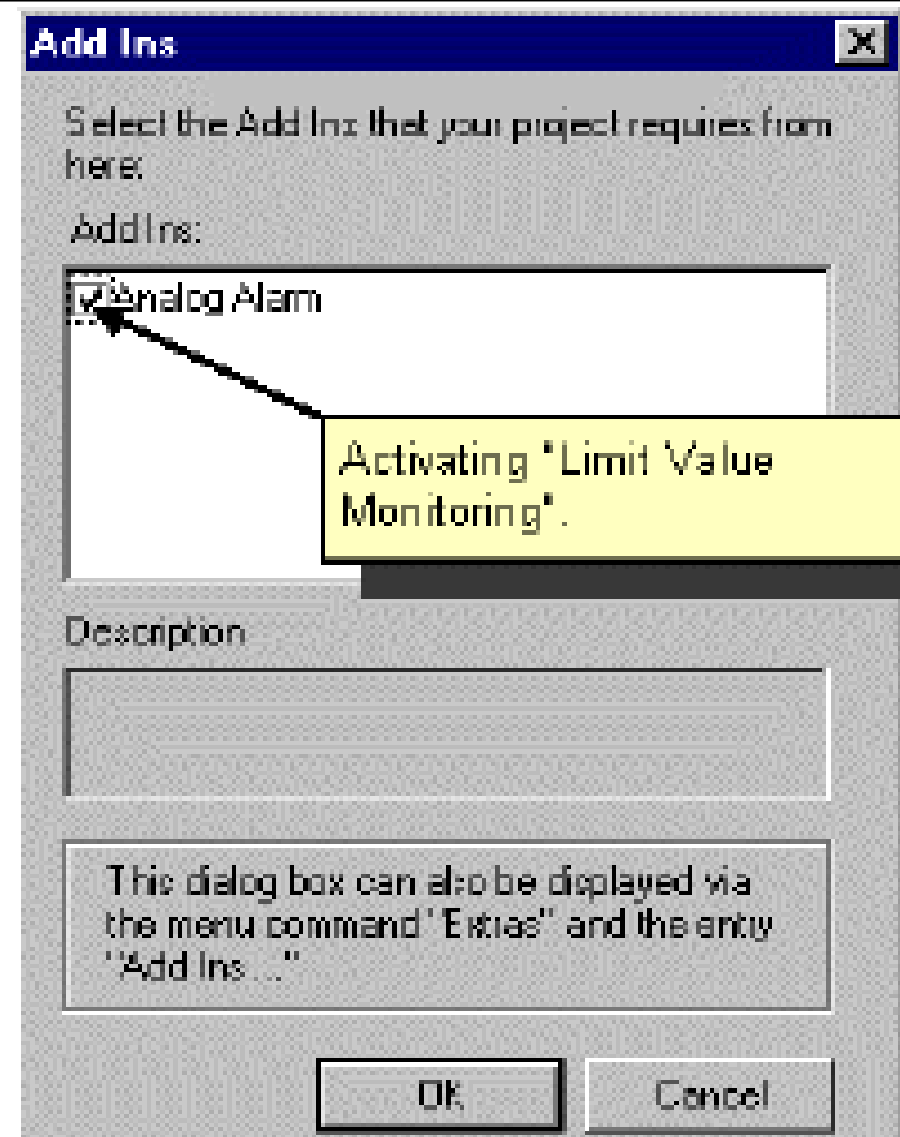
- Chọn màu Text và màu nền cho các thông báo lỗi: Came in, Went out, Acknowledge.



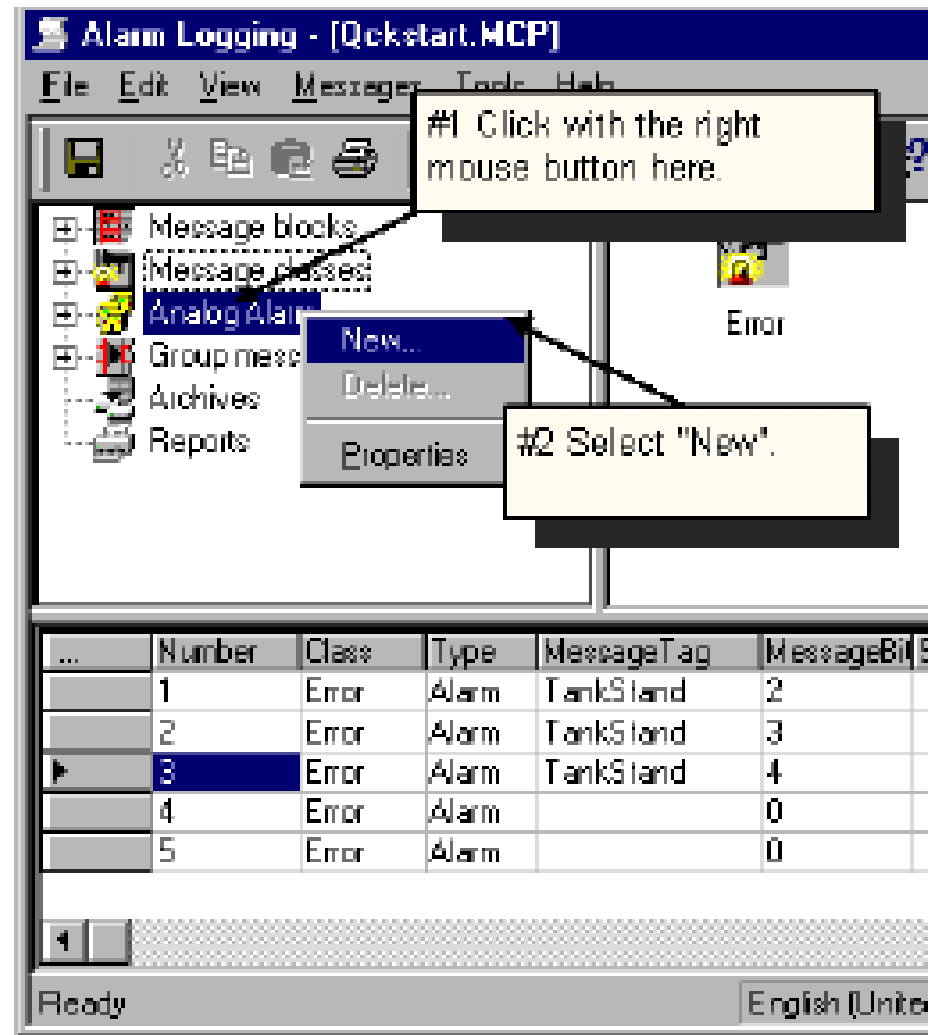
## 5. Cài đặt Limit Value

+ Trên cửa sổ Alarm Logging Editor chọn menu Tools → Add-Ins.

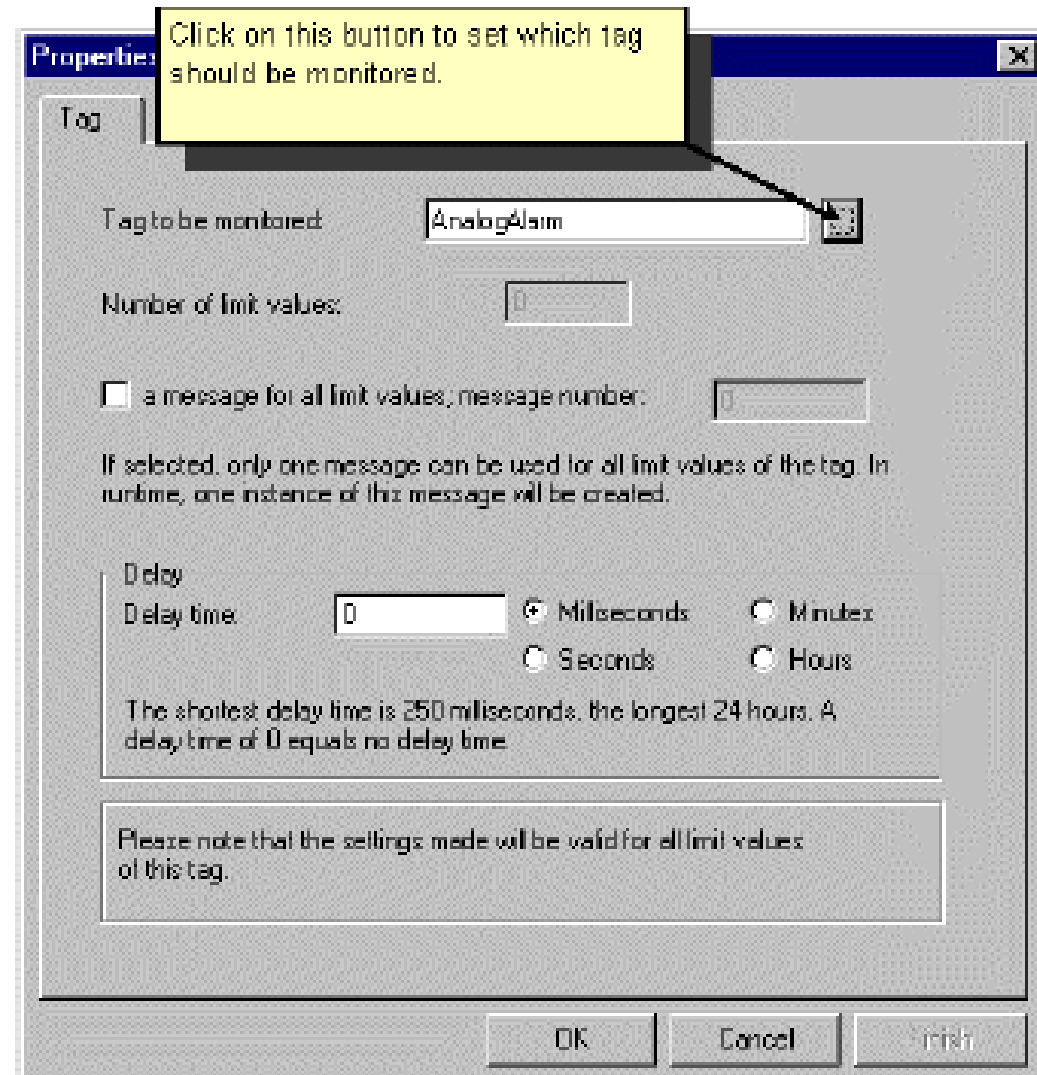
+ Trên cửa sổ Add-Ins chọn Analog Alarm



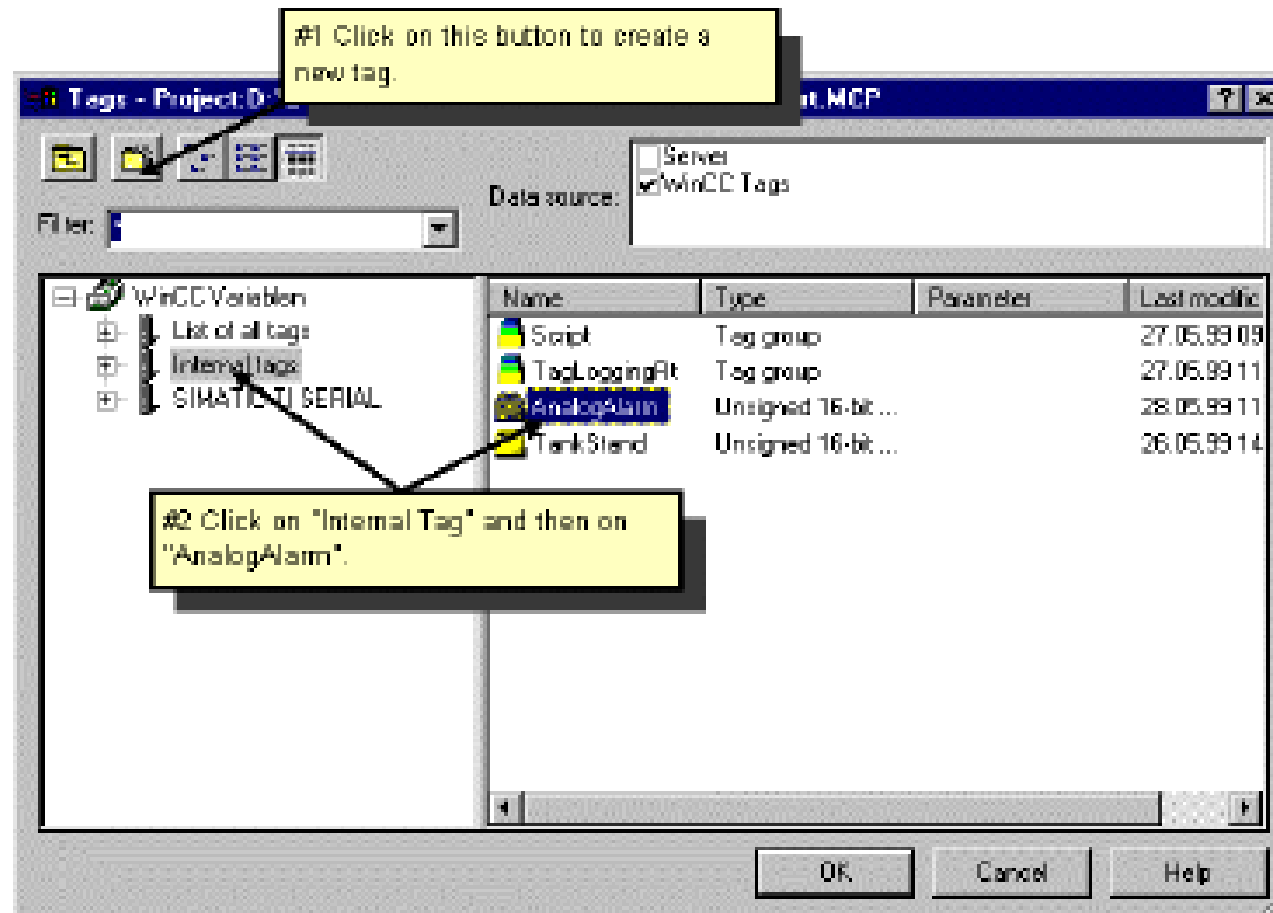
Nhấp phải chuột vào Analog alarm, chọn New



Trên cửa sổ  
Properties, chọn Tag  
muốn quan sát hai  
giá trị mức Lower và  
Upper Value



Có thể chọn Tag có sẵn hoặc tạo mới Tag từ cửa sổ hiện ra

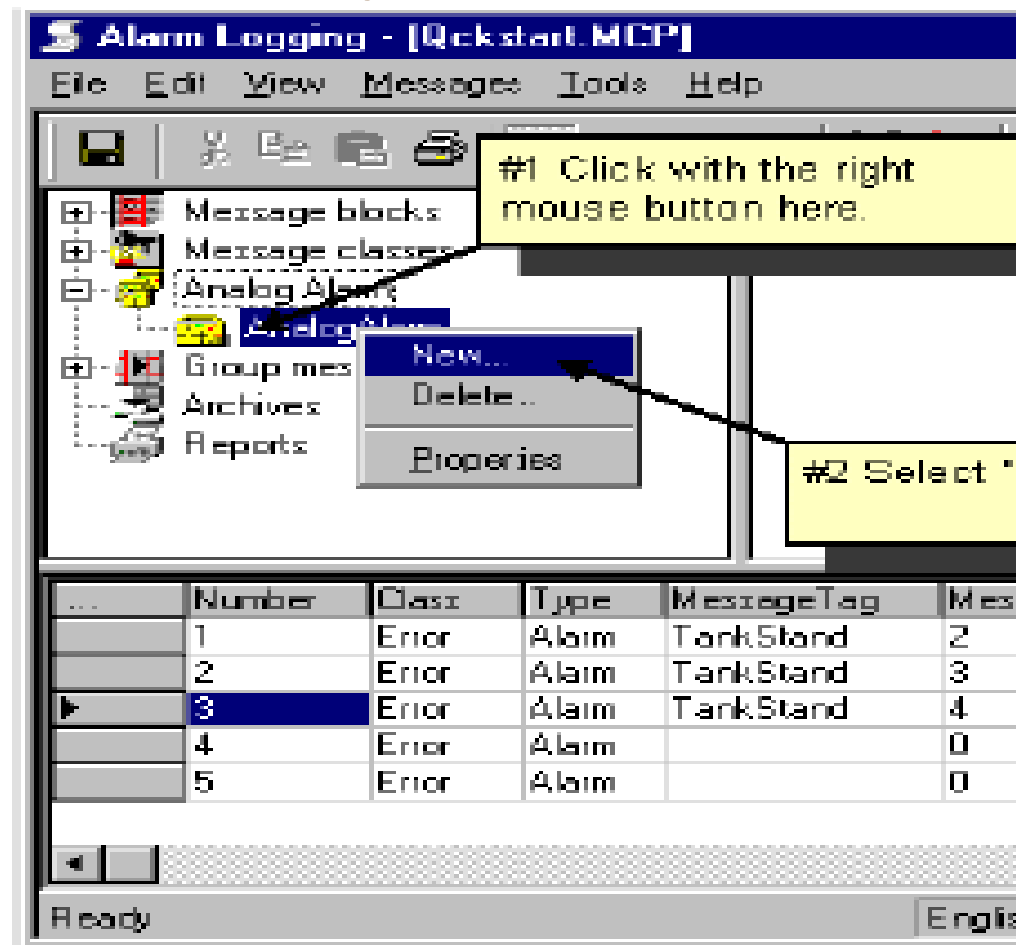


+ Ở đây có thể tạo một tag mới có tên là AnalogAlarm, kiểu Unsigned 16bits

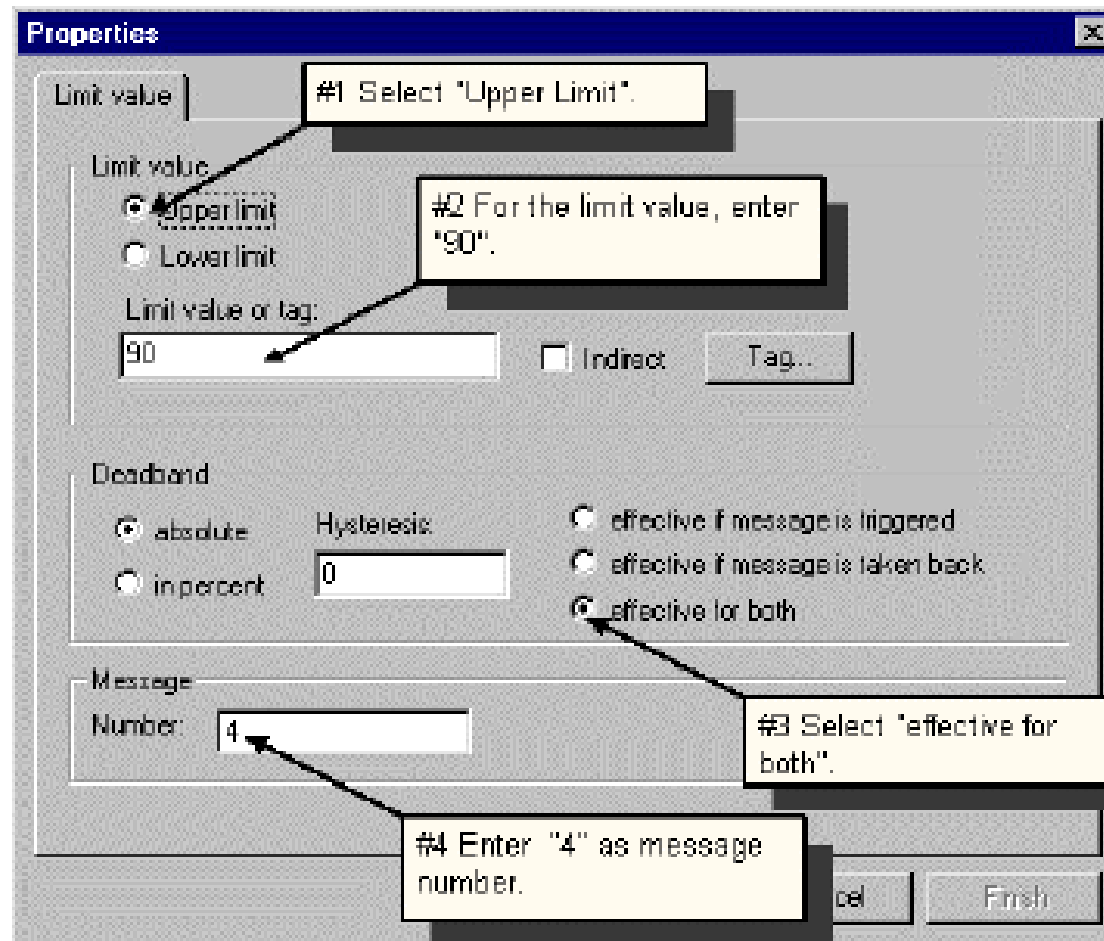
+ Nhấn OK để xác nhận cài đặt và thoát

## Cài đặt Limit Value

Nhấp phải chuột vào Tag mới vừa khai báo, chọn New



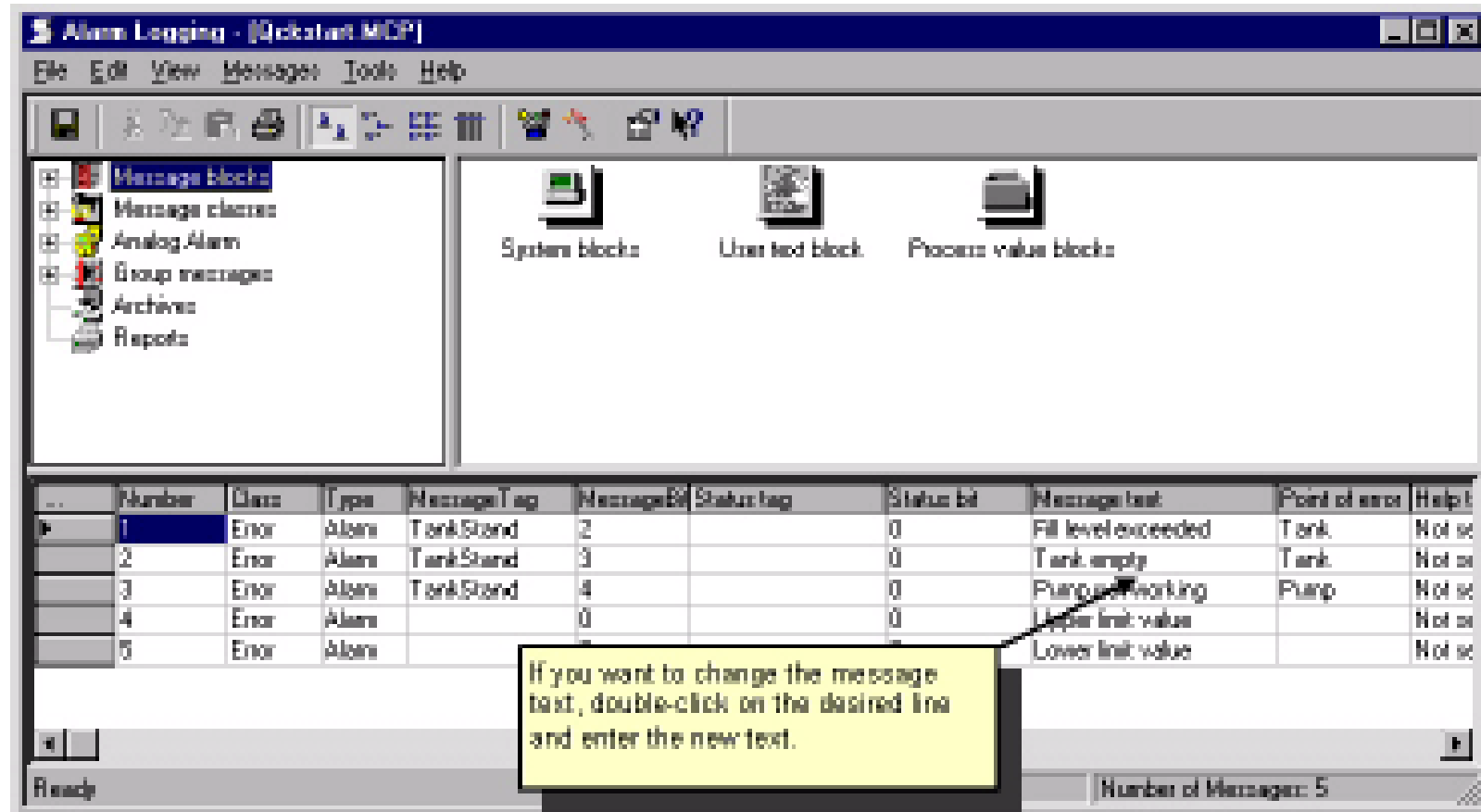
Trên cửa sổ Properties hiện ra, chọn các giá trị cho Upper và Lower Limit



Nhấn OK để  
xác nhận  
cài đặt và  
thoát



## Chọn Save để thoát khỏi Alarm Logging Editor



...	Number	Class	Type	MessageTag	MessageBit	Status tag	Status bit	Message text	Point of error	Help
	1	Error	Alarm	TankStand	2		0	Fill level exceeded	Tank	Not se
	2	Error	Alarm	TankStand	3		0	Tank empty	Tank	Not se
	3	Error	Alarm	TankStand	4		0	Pump not working	Pump	Not se
	4	Error	Alarm		0		0	Upper limit value		Not se
	5	Error	Alarm					Lower limit value		Not se

If you want to change the message text, double-click on the desired line and enter the new text.

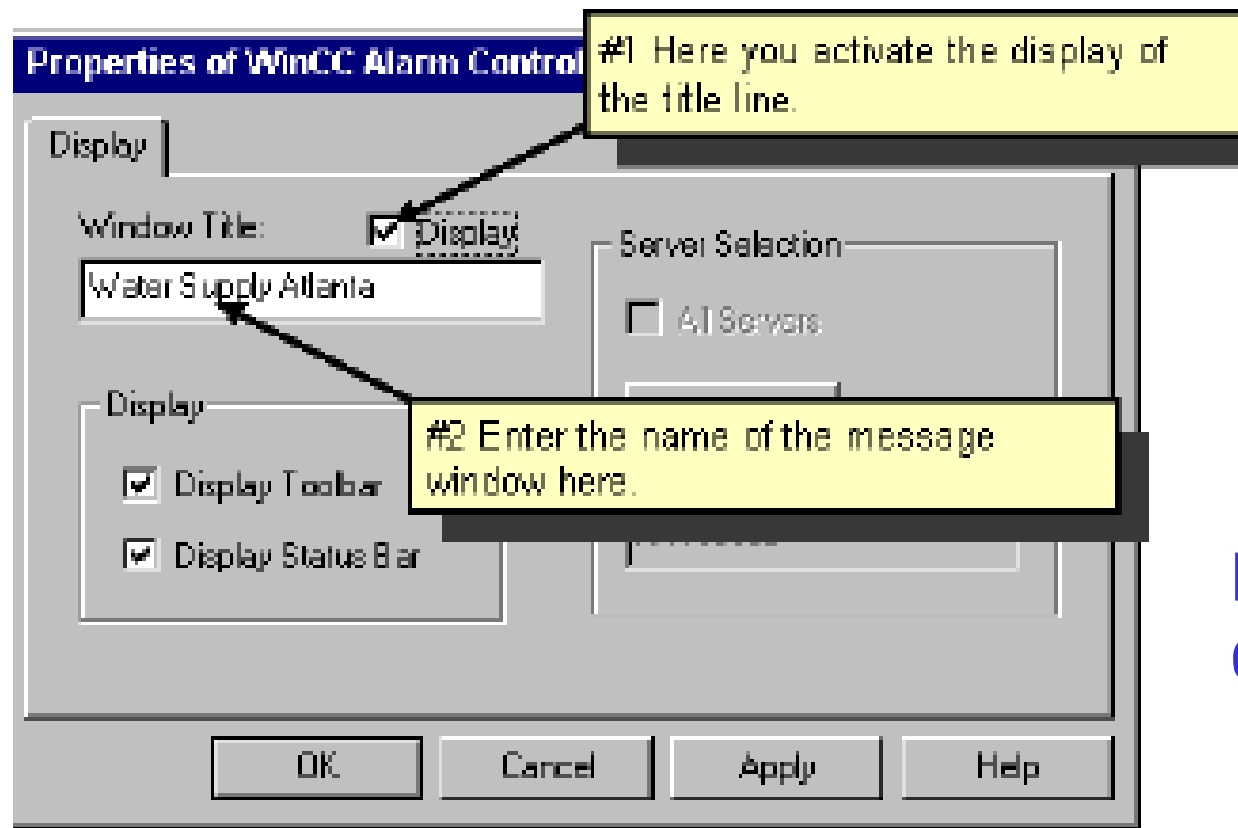
Ready

Number of Messages: 5

## 6. Tạo Message window trên màn hình soạn thảo

Chọn WinCC Alarm Control trên tab “ Control” bên của sổ Object Palette và đặt vào màn hình soạn thảo.

Gõ Tên cửa sổ và chọn vào ô Display

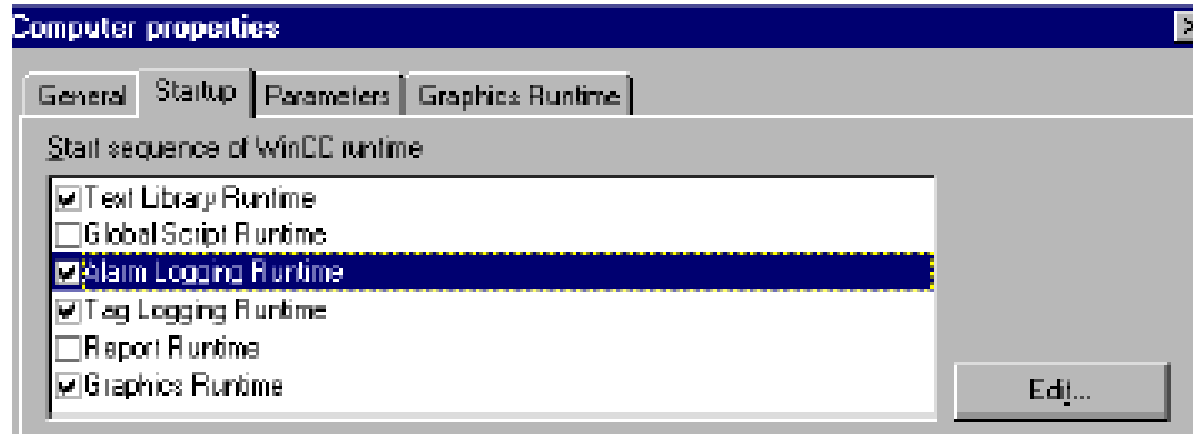


Nhấn nút OK để thoát

## 7. Thiết lập tham số Runtime


+ Trên cửa sổ WinCC Explorer click chuột phải vào mục Computer, chọn Properties → Chọn Properties lần nữa.

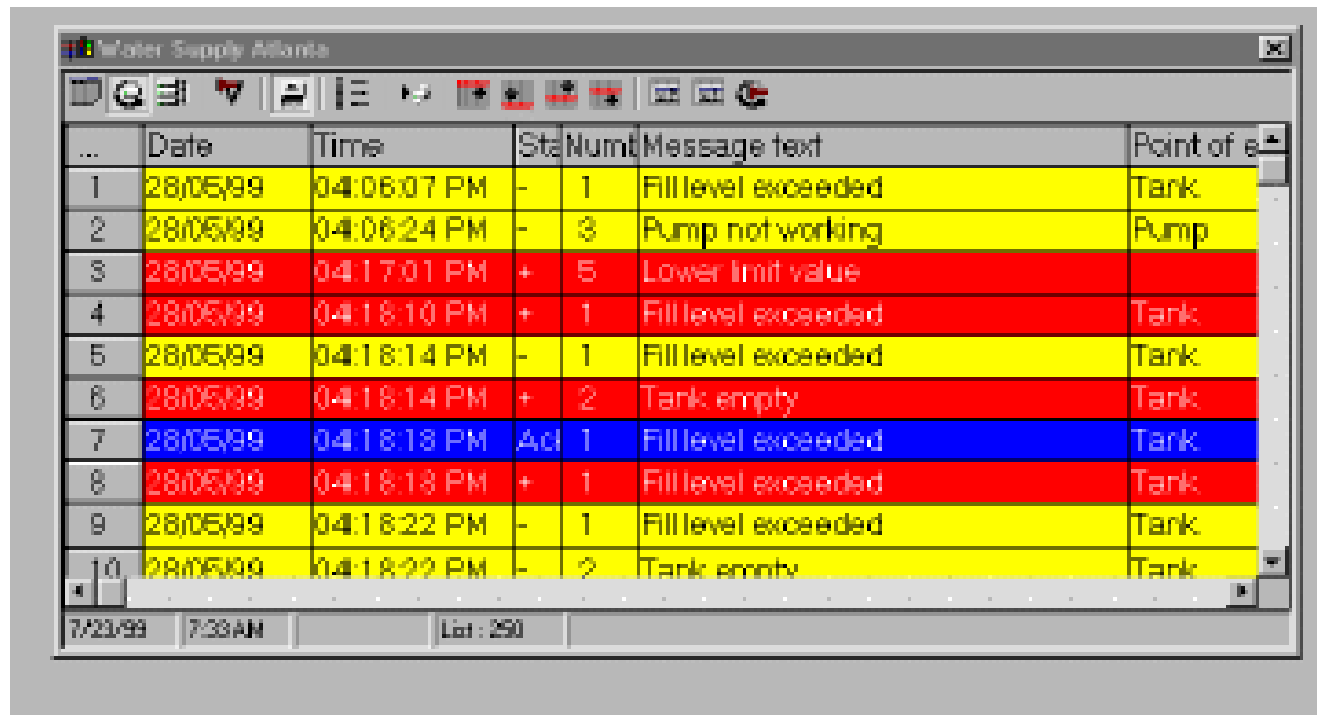
+ Trên tab “Start up”, chọn tag Alarm Logging Runtime và Text Library Runtime.



Nhấn OK để thoát

## 8. Chạy chương trình

Nhấn nút Activate  để chạy chương trình. Có thể kích hoạt chương trình mô phỏng để xem các thông báo lỗi khi chạy Runtime



The screenshot shows a software window titled "Water Supply Atlanta" with a toolbar and a table of log entries. The table has columns for Date, Time, Sta, Num, Message text, and Point of s. The entries are numbered 1 through 10 and show various status changes and messages such as "Fill level exceeded", "Pump not working", and "Tank empty".

...	Date	Time	Sta	Num	Message text	Point of s
1	28/05/99	04:06:07 PM	-	1	Fill level exceeded	Tank
2	28/05/99	04:06:24 PM	-	3	Pump not working	Pump
3	28/05/99	04:17:01 PM	+	5	Lower limit value	
4	28/05/99	04:18:10 PM	+	1	Fill level exceeded	Tank
5	28/05/99	04:18:14 PM	-	1	Fill level exceeded	Tank
6	28/05/99	04:18:14 PM	+	2	Tank empty	Tank
7	28/05/99	04:18:18 PM	Act	1	Fill level exceeded	Tank
8	28/05/99	04:18:18 PM	+	1	Fill level exceeded	Tank
9	28/05/99	04:18:22 PM	-	1	Fill level exceeded	Tank
10	28/05/99	04:18:22 PM	-	2	Tank empty	Tank

7/23/99 7:33 AM Lat: 290

# LẬP TRÌNH C CHO WINCC

# TẠO FUNCTION VÀ ACTION TRONG WINCC

**Function:** Sử dụng hàm khi cần tính toán nhiều lần trong chương trình. Lợi ích là:

- + Chỉ cần lập trình một lần. Khi cần thực thi thì chỉ việc gọi hàm và đưa vào các đối số thích hợp.
- + Chương trình ngắn và dễ hiểu.

**Action:** Khác với Function, action chỉ hoạt động khi có điều kiện kích (Function không tự hoạt động khi chạy Runtime).

- + Action không có đối số.
- + Có thể tạo bản quyền cho Action

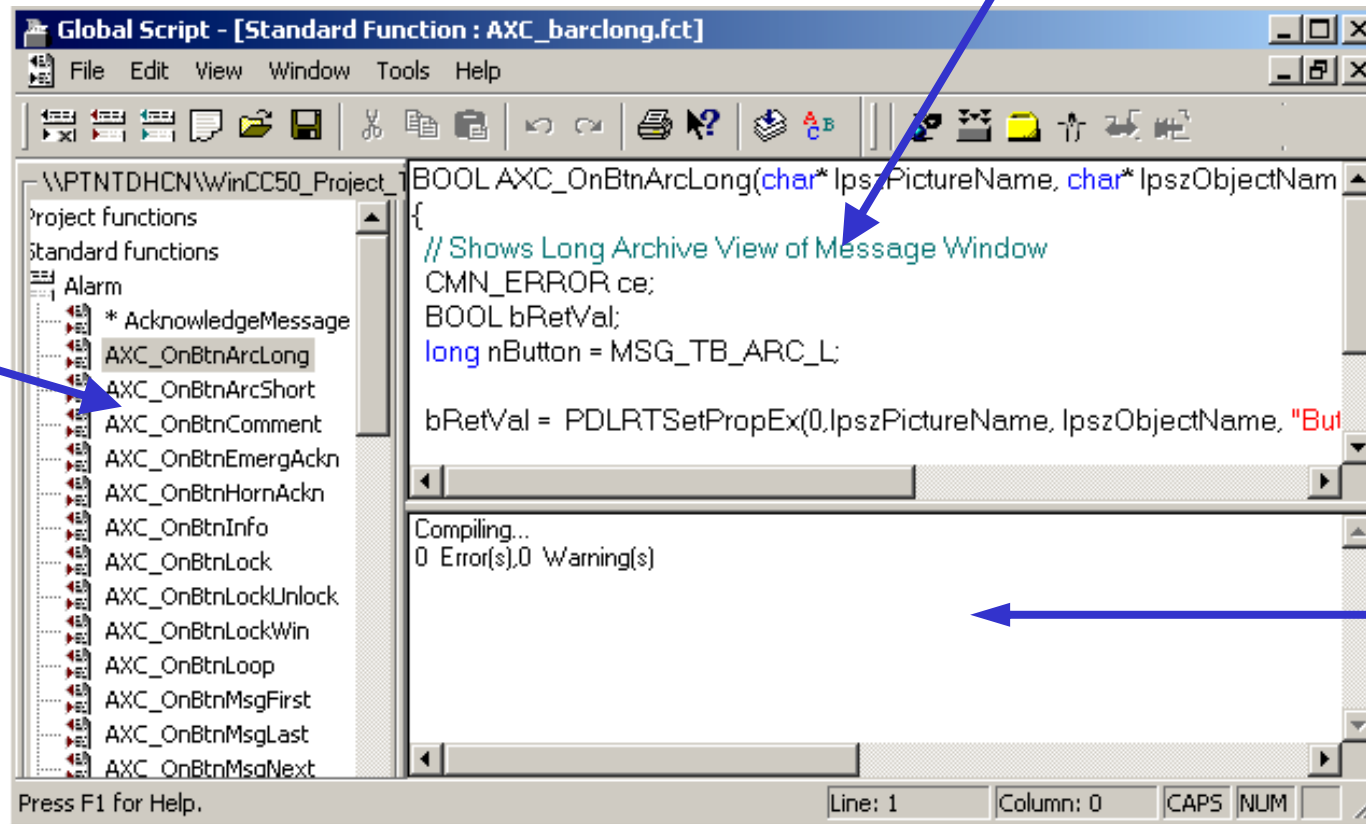
Để tạo và soạn thảo Function hay Action dùng Global Script của WinCC

# GLOBAL SCRIPT



Cửa sổ soạn thảo của Global Script: **Editing Window**

**Navigation Window**



**Output Window**

**Navigation Windows:** Cửa sổ để lựa chọn Function hay Action để soạn thảo.

**Editing Windows:** Cửa sổ soạn thảo chương trình.

**Output Windows:** Cửa sổ hiển thị kết quả biên dịch của hàm và Action hoặc Search Files

## Khả năng lập trình và ứng dụng:

**Action:** Được tạo mới hay sửa đổi trong Global Script. Chỉ dùng được trong Project mà nó được tạo ra.

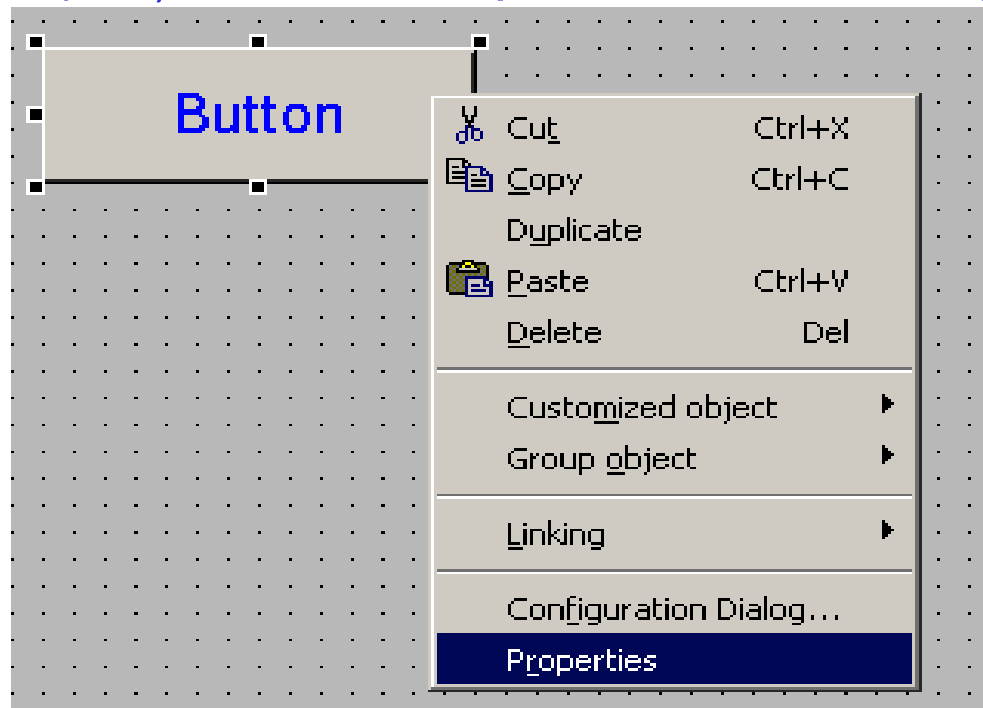
**Project Function:** Chỉ được dùng trong Project hiện tại. Có thể tạo mới hay sửa những hàm có sẵn.

**Standard Function:** Dùng được trong tất cả Project. Có thể tạo mới hay sửa những hàm có sẵn

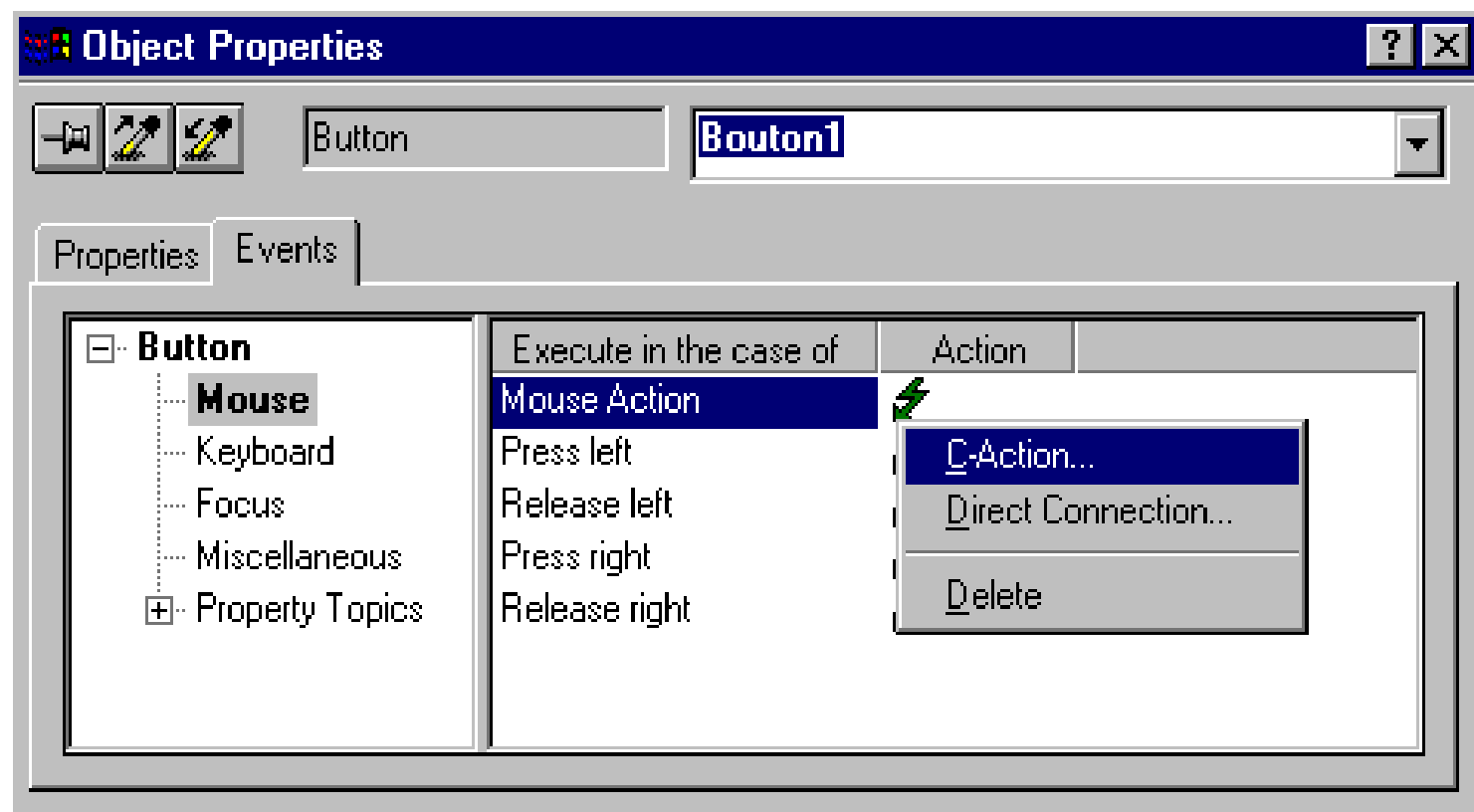
**Internal Function:** Dùng được trong tất cả Project. Không thể tạo mới hay sửa những hàm có sẵn.

## Lập trình tạo Action

Từ màn hình Graphics Designer, click phải vào đối tượng (object) cần thiết lập Action, chọn Properties.



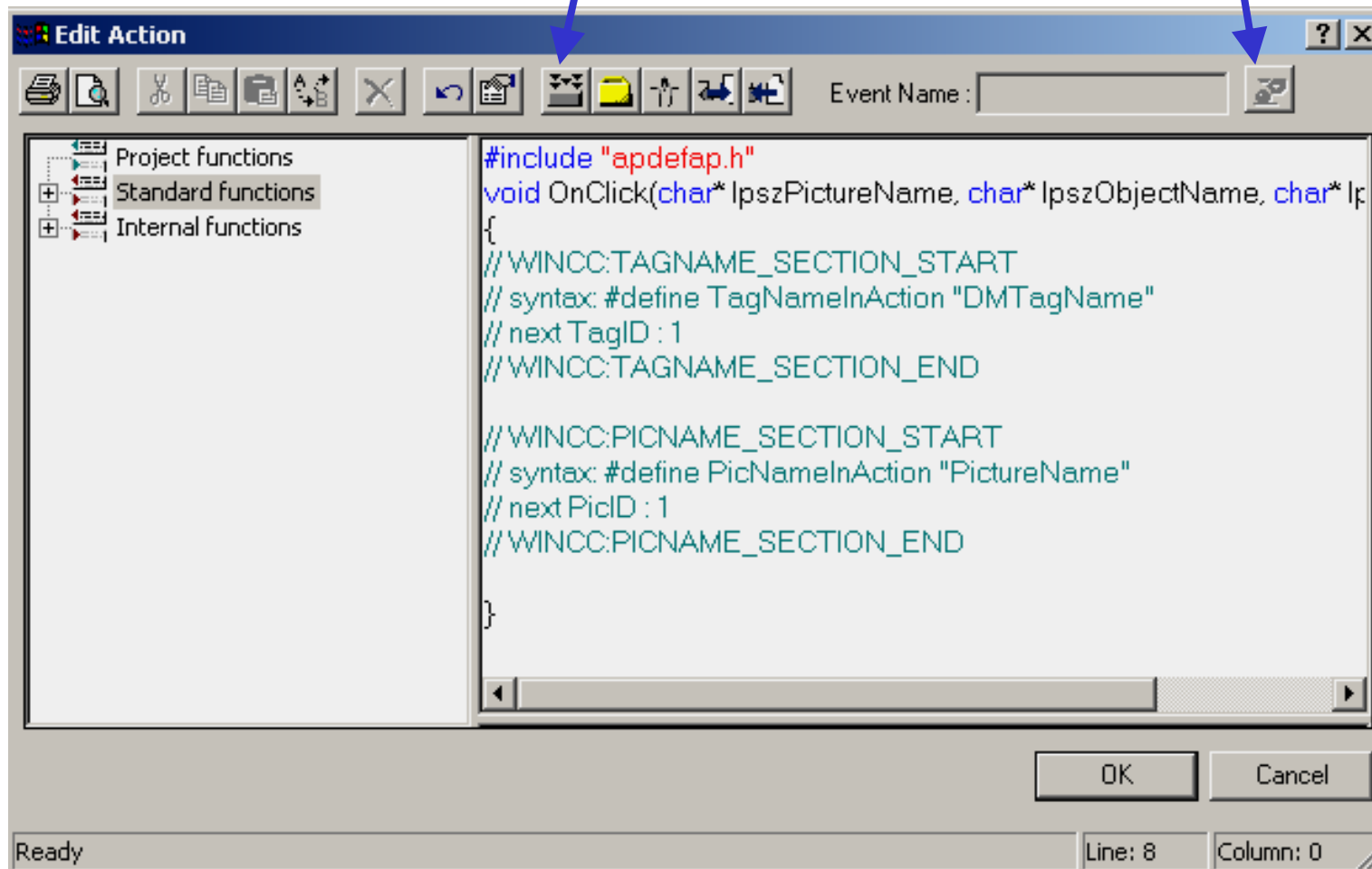
Click phải vào mục Dynamics hay Action của Properties hay Events và chọn C-Action từ menu hiện ra




Cửa sổ Global Script

Biên dịch

Tạo Trigger



+ Lập trình C-Action cho Properties của đối tượng thì cần phải có Trigger, còn Events thì không(vì bản thân event là trigger rồi).

+ Sau khi lập trình cho Action xong thì biên dịch bằng n   
Nếu có lỗi hiển thị trong Output Window thì Double click vào ngay lỗi để biết vị trí lỗi trong chương trình và sửa lỗi, nếu không có lỗi thì chọn OK để thoát.

## Cấu trúc chương trình của một C-Action cho một Property của đối tượng:

```
#include "apdefap.h"
long _nain(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    /*1*/ long lReturnValue;
    /*2*/ lReturnValue = GetTagSDword('S32i_course_test_1');
    /*3*/ return lReturnValue;
}
```

Câu lệnh # include"apdefap.h" là khai báo thư viện các Function và Action mà WinCC hỗ trợ sẵn.

Header(màu xám): Đây là dòng mã lệnh tự động phát sinh giống nhau cho các Properties và không được thay đổi. Bao gồm :

- + Picture Name (lpszPictureName)
- + Object Name (lpszObjectName)
- + Property Name (lpszPropertyName)



- (1) Khai báo biến
- (2) Tính toán các giá trị cho Property
- (3) Trả về giá trị cho Property.

**Sau khi lập trình xong Action cho một Property của đối tượng thì phải chọn Trigger cho nó**

**Cấu trúc chương trình của một C-Action cho một Event của đối tượng:**

```
#include 'apdefap.h'
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    /*1*/ long lValue;

    /*2*/ lValue = GetTagSDWord("S32i_course_test_1");

    SetLeft(lpszPictureName, lpszObjectName, lValue);
}
```

## **Bài tập thực hành:**

Viết chương trình tính tổng hai số integer và xuất kết quả tính được ra cửa sổ GSC Diagnostics khi nhấn một Button

**Header(màu xám):** Đây là dòng mã lệnh tự động phát sinh và không được thay đổi. Header có thể khác nhau cho các events khác nhau.

- (1) Khai báo biến
- (2) Lập trình xử lý sự kiện.

### **Lập trình tạo một Function**

+ Project Function, Standard Function và Internal Function đều có thể lập trình trong Global Script.

+ Click chuột phải vào loại Function muốn tạo, chọn New

## Cấu trúc của một Function đơn giản:

```
double MeanValue(double dValue1, double dValue2, double dValue3)
{
    double dMeanValue;
    dMeanValue = (dValue1+dValue2+dValue3)/3;
    return dMeanValue;
}
```

### Khai báo

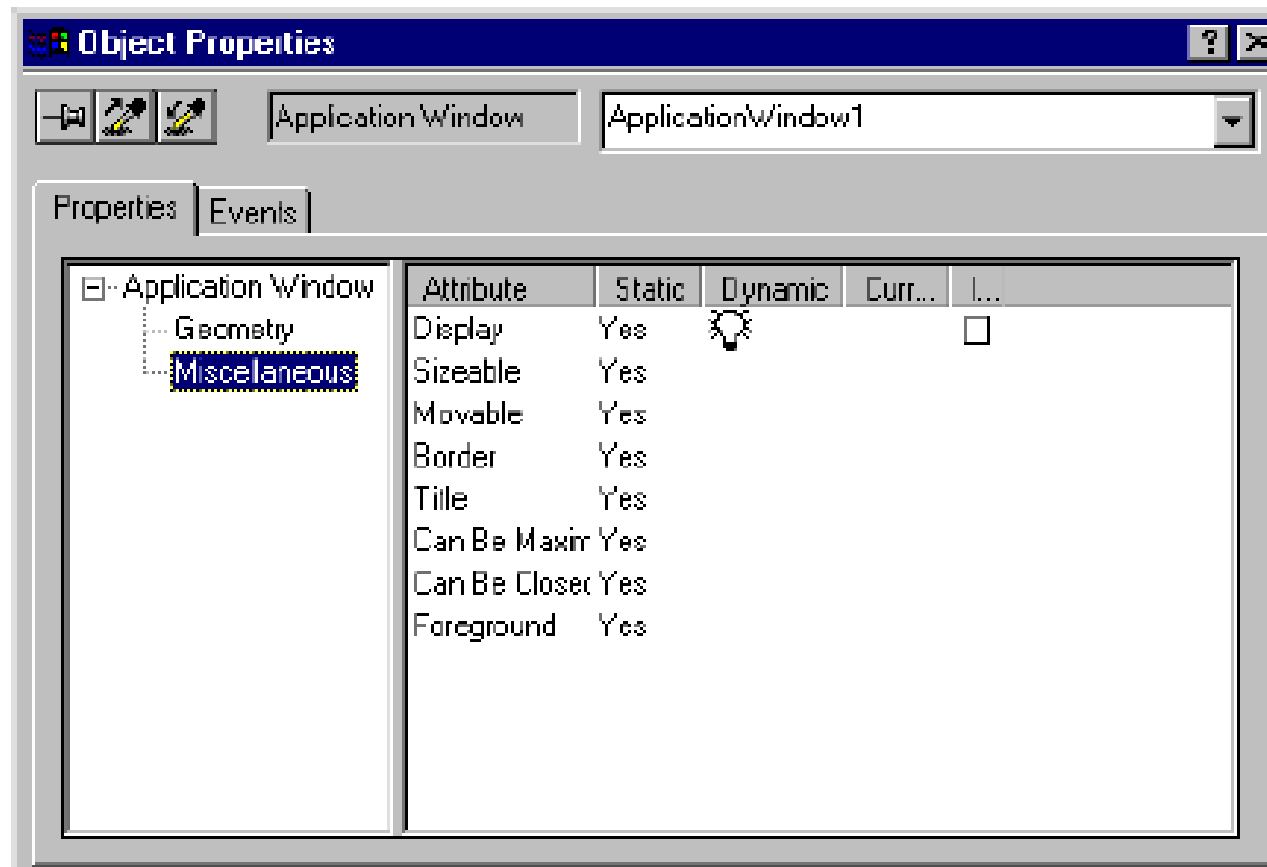
- + Kiểu dữ liệu trả về của hàm
- + Tên hàm
- + Đối số cùng với kiểu dữ liệu của nó
- + Thân hàm
- + Giá trị trả về của hàm.

+ Để xem kết quả tính toán của hàm và chẩn đoán lỗi ta dùng cửa sổ Global Script Diagnostics Window, cách thực hiện như sau:

- Mở một trang màn hình trong Graphics Designer.
- Chọn Application Window có trong Smart Object bên Object Palete và kéo vào trang màn hình đang thiết kế và thả chuột ra
- Chọn Global Sript từ Window Contents hiện ra → Chọn GSC Diagnostic → OK.



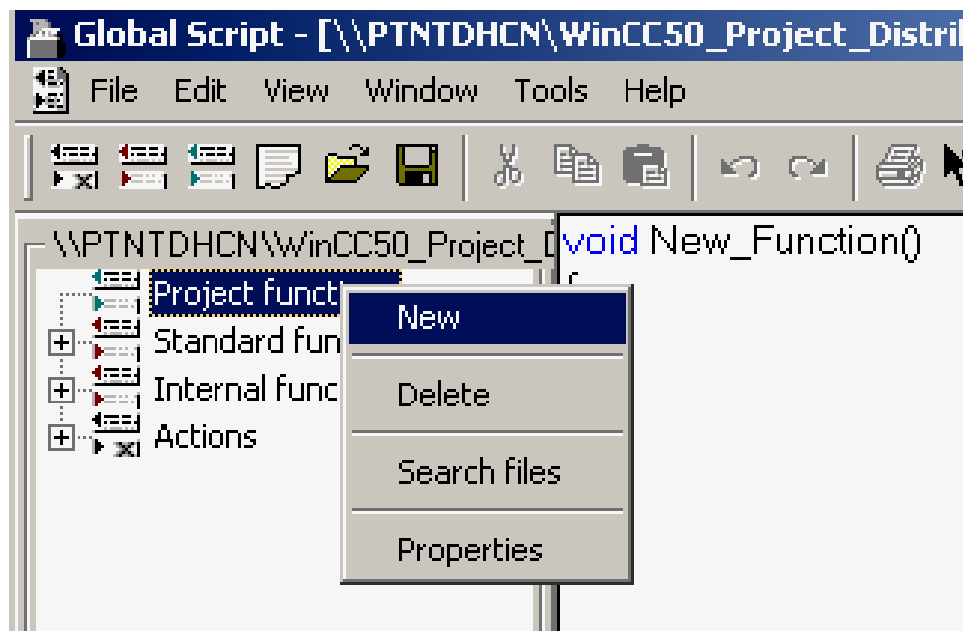
Click phải vào Application Window vừa tạo chọn Property → Chọn Yes cho tất cả các thuộc tính của Miscellaneous



- + Khi chạy Runtime kết quả của lệnh printf sẽ được hiện ra trên cửa sổ này.
- + Có thể dùng nút nhấn để dừng việc cập nhật giá trị để in ấn hoặc lưu trữ các kết quả.

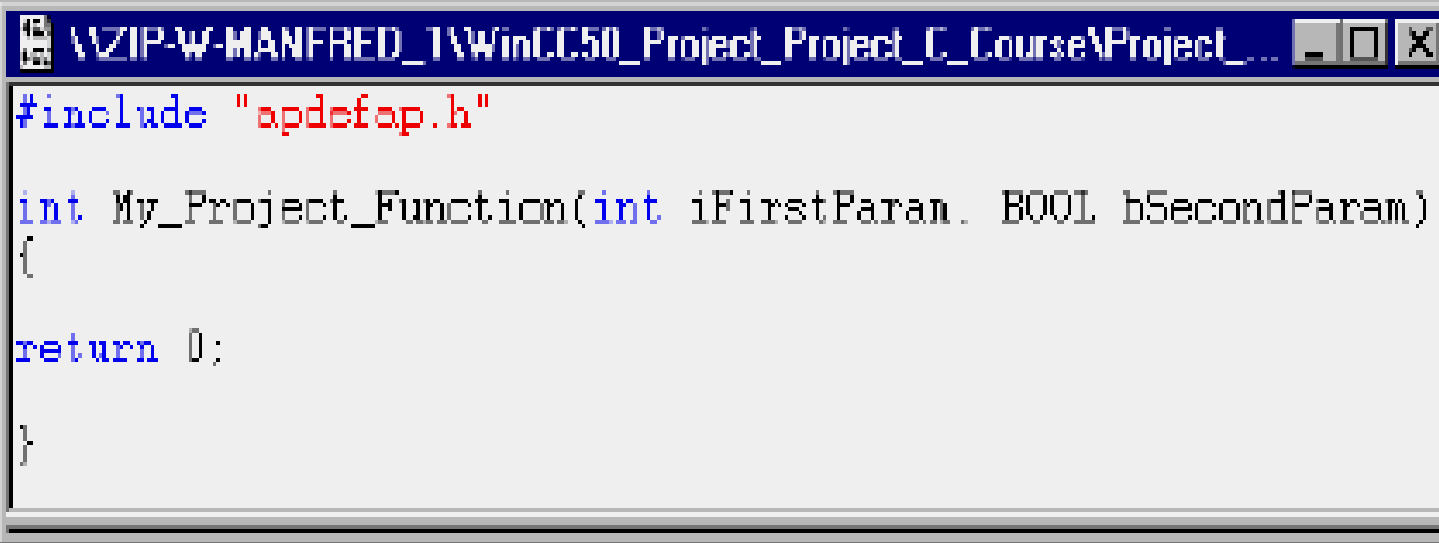
## Trình tự tạo một Project Function

1. Mở cửa sổ Global Script
2. Nhấp phải chuột vào Project Functions và chọn New





3. Viết chương trình cho hàm bên của sổ Editor, bao gồm:  
Kiểu dữ liệu trả về của hàm, tên hàm, khai báo các đối số, thân hàm, giá trị trả về cho hàm.



```
#include "apdefap.h"

int My_Project_Function(int iFirstParam, BOOL bSecondParam)
{
return 0;
}
```

***Lưu ý là sau mỗi dòng lệnh là dấu “;”***

4. Biên dịch cho hàm.

5. Có thể tạo Password để giữ bảo mật cho hàm bằng cách nhấn nút



Info / Trigger

Properties

Info

Created By:

Creation Date: 8 1 2004

Changed By:

Date of change: 8 1 2004

Password

Version: 1 0

Comment :

OK Cancel Apply Help

## Các thủ tục hay sử dụng khi lập trình

### Định nghĩa Tag:

Cú pháp: # define Tên Tag khi lập trình” Tên Tag đã khai báo trong Tag Manager” .

### Ví dụ:

Giả sử ta đã định nghĩa một Tag tên là Start trong Tag Manager có địa chỉ trên PLC đã xác định từ trước. Sử dụng câu lệnh:

```
# define Tag0”Start”
```

thì khi đó trong chương trình ta dùng biến Tag0 thay cho Tag “Start” đã khai báo trong Tag Management

## Khai báo hằng, biến

### Khai báo hằng:

Cú pháp: # define Tên hằng\_Giá trị của hằng

Ví dụ: #define Max 365;

### Khai báo biến:

Cú pháp: Kiểu dữ liệu của biến\_Tên biến

(Có thể gán giá trị ban đầu cho biến khi khai báo)

Ví dụ:           int a;  
                  int b=0;

## Các kiểu dữ liệu của biến và tầm sử dụng:

Data Type	Value Range
int	-2 147 483 648 to 2 147 483 647
unsigned int	0 to 4 294 967 295
short	-32 768 to 32 767
unsigned short	0 to 65 535
long	-2 147 483 648 to 2 147 483 647
unsigned long	0 to 4 294 967 295
char	-128 to 127 (all ASCII characters)
unsigned char	0 to 255 (all ASCII characters)
float	$-10^{38}$ to $0^{38}$
double	$-10^{308}$ to $0^{308}$

## Thủ tục xuất dữ liệu ra màn hình:

**printf ();**

Ví dụ:

```
tuoi=10;
```

```
printf("I am %d years old\r\n",tuoi);
```

%d,\r,\n: là các định dạng khi xuất dữ liệu ra màn hình

## Định dạng dữ liệu khi xuất ra màn hình

Parameter	Description
%d	Output of a decimal number (int or char)
%ld	Output of a variable of the type long as a decimal number
%c	Output of a character (char)
%x	Output of a number in hexadecimal format (with lower case a...f)
%X	Output of a number in hexadecimal format (with upper case A...F)
%o	Output of a number in octal format
%u	Output of a decimal number (only for unsigned types)
%f	Output of a float value in floating point notation, e.g. 3.43234
%e	Output of a float value in exponential notation, e.g. 23e+432
%E	Like %e, but with upper case E, e.g. 23E+432
%s	Output of a character string (char*)
%le	Output of a double value
%%	Output of a % character
\n	Output of a line change (carriage return)
\r	Output of a line feed
\t	Output of a tab
\\	Output of a \ character

## Một số hàm hay sử dụng trong chương trình

### 1. SetTagBit

Cú pháp: `BOOL SetTagBit(Tag Tag_Name, short int value);`

Nội dung: Định giá trị cho một Tag có kiểu dữ liệu là Binary

### 2. SetTagByte

Cú pháp: `BOOL SetTagByte(Tag Tag_Name, BYTE value);`

Nội dung: Định giá trị cho một Tag có kiểu dữ liệu là 8bit không dấu

**3. SetTagSByte**   Cú pháp: `BOOL SetTagSByte(Tag Tag_Name, signed char value);`

Nội dung: Định giá trị cho một Tag có kiểu dữ liệu là 8bit có dấu

Tương tự cho các hàm SetTagWord, SettagDWord...



## 4. GetTagBit

Cú pháp: **BOOL GetTagBit(Tag Tag\_Name);**

Nội dung: Lấy giá trị hiện tại của một Tag có kiểu Binary

## 5. GetTagByte

Cú pháp: **BYTE GetTagByte(Tag Tag\_Name);**

Nội dung: Lấy giá trị của một Tag có kiểu dữ liệu là 8bit không dấu

**6. GetTagSByte** Cú pháp: **signed char GetTagSByte(Tag Tag\_Name);**

Nội dung: Lấy giá trị hiện tại của một Tag có kiểu dữ liệu là 8bit có dấu

Tương tự cho các hàm GetTagWord, GetTagDWord...

## Các hàm điều khiển

### 1. Thoát khỏi Runtime:

Cú pháp: **BOOL DeactivateRTProject();**

Nội dung: Thoát khỏi chương trình WinCC đang chạy Runtime

### 2. Thoát khỏi WinCC:

Cú pháp: **BOOL ExitWinCC();**

Nội dung: Thoát khỏi chương trình WinCC, kể cả WinCC Explorer

## Các hàm xử lý tính toán

Operator	Description
+ (unary)	Positive sign (actually has no effect)
- (unary)	Negative sign
+ (binary)	Addition
- (binary)	Subtraction
*	Multiplication
/	Division
%	Modulo (returns the remainder of a division)
++	Increment
--	Decrement

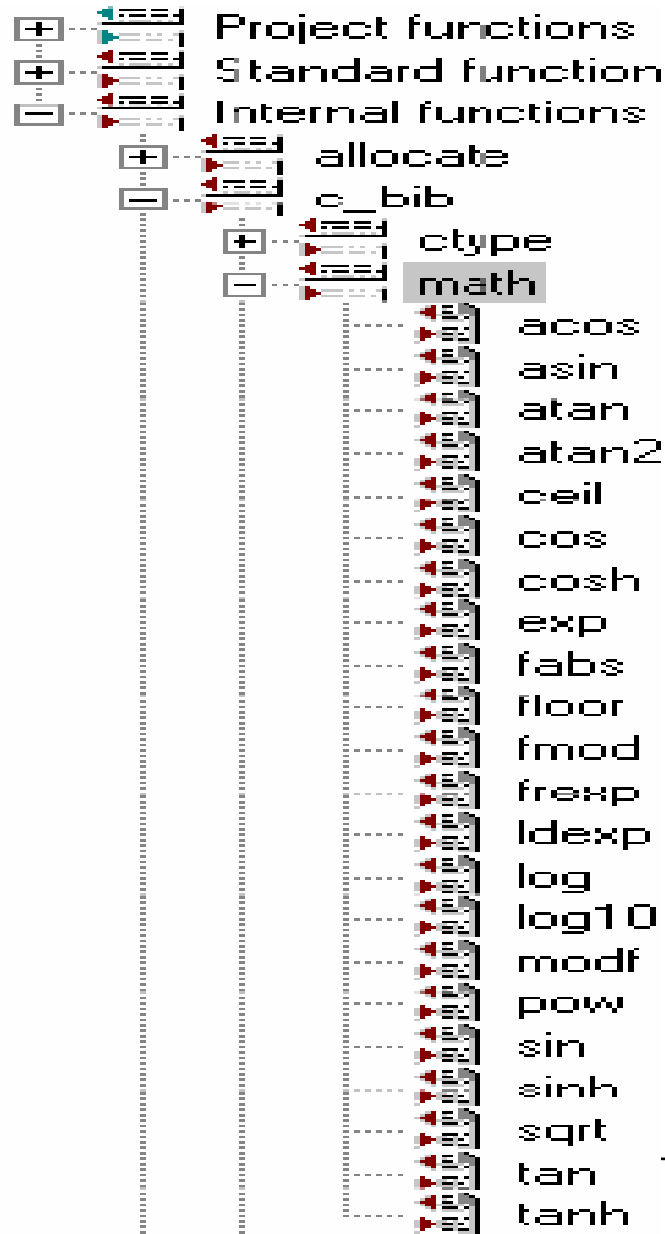
## Các hàm tính toán trên Bit

Operator	Description
&	Bit-by-Bit AND
	Bit-by-Bit OR
^	Bit-by-Bit exclusive OR
~	Bit-by-Bit inversion
<<	Move bits to the left
>>	Move bits to the right

## Các toán tử Logic

Operator	Description
>	Greater than
>=	Greater than or equal to
==	Equal to
!=	Not equal to
<=	Less than or equal to
<	Less than
&&	Logical AND
	Logical OR
!	Logical inversion

**Các  
hàm  
toán  
học**



## Pointer- Con trỏ

- + Là công cụ lập trình thiết yếu của ngôn ngữ C
- + Con trỏ là một biến chứa một địa chỉ, thường là địa chỉ của một biến khác

Khai báo biến kiểu con trỏ:

**Kiểu dữ liệu\* Tên biến;**

Ví dụ: int\* a;

Thì biến con trỏ a sẽ trỏ vào một địa chỉ có kiểu dữ liệu là số nguyên. Tuy nhiên với khai báo này thì nội dung con trỏ chưa được xác định.

+ Để sử dụng biến con trỏ thì phải gán địa chỉ của một biến cho nó.

Ví dụ: `int a;`  
`int* b;`  
`b= &a; // Biến con trỏ b sẽ có nội dung là địa chỉ của biến a kiểu int.`

+ Để truy xuất đến nội dung của biến mà con trỏ trỏ tới ta dùng toán tử “ \* “

Ví dụ: `int a,`  
`int b=5;`  
`int* c;`  
`c=&b;`  
`a=*b;`  
thì kết quả của biến a là 5;



## Bài tập:

Kết quả khi nhấn nút **Pointers** thực hiện đoạn chương trình sau là gì?

```
#include "opdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszProperty)
{

    int iValue1 = 126;
    int iValue2 = 23;

    //declare and initialize pointer
    int* piValue = NULL;

    printf("\r\nExample 1\r\n");
    printf("Address: %x\tValue: undefined\r\n", piValue);

    //point at iValue1
    piValue = &iValue1;
    printf("Address: %x\tValue: %d\r\n", piValue, *piValue);

    //point at iValue2
    piValue = &iValue2;
    printf("Address: %x\tValue: %d\r\n", piValue, *piValue);

}
```

## Vector- Dữ liệu kiểu mảng

Cú pháp:

**Kiểu dữ liệu \_Tên biến[ Số phần tử của biến];**

Ví dụ: int a[5];

thì biến a là một vector có 5 phần tử kiểu int:

a[1],a[2],a[3],a[4],a[5]

+ Muốn truy xuất từng phần tử của vector ta dùng toán tử [ ].

Ví dụ b=a[3];...

Bài tập:

Kết quả khi nhấn nút **Vectors** thực hiện đoạn chương trình sau là gì?

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszProperty)
{
    //declare and initialize int vector
    int iValue[5] = { 10, 20, 30, 40, 50 };

    int iIndex;

    printf("\r\nExample 2\r\n");

    //access vector elements
    for (iIndex = 0; iIndex<5; iIndex++)
    {
        printf("Index %d\t Value: %d\r\n", iIndex, iValue[iIndex]);
    }
}
```

## Dữ liệu kiểu String

String có thể được khai báo theo hai cách:

+ Là một vector các ký tự.

Ví dụ: `char st [10]="Hello";`

+ Hoặc là một con trỏ trỏ tới một ký tự.

Ví dụ: `char* st="Hello".`

Nếu định nghĩa theo kiểu pointer thì không cần thêm kích thước cho chuỗi.

## Nhóm lệnh tạo vòng lặp

### 1. Lệnh While

```
int i = 0;
while (i < 5)
{
    //do something
    ++i;
}
```

Bài tập:

Kết quả khi nhấn nút  thực hiện đoạn chương trình sau là gì?

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszProperty)
{
    //loop count
    int iCount = 0;

    printf("\r\nExample 1\r\n");

    //while loop
    while (iCount < 5)
    {
        //do something
        printf("Executed loop: iCount = %d\r\n",iCount);

        ++iCount;
    }

    printf("Exit loop: iCount = %d\r\n",iCount);
}
L
}
```

## 2. Lệnh do -while

```
int i = 0;

do
{
    //do something
    ++i;
}
while (i < 5);
```

## Bài tập:

Kết quả khi nhấn nút **Do-While** thực hiện đoạn chương trình sau là gì?

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    //loop count
    int iCount = 0;

    printf("\r\nExample 2\r\n");

    //do-while loop
    do
    {
        //do something
        printf("Executed loop: iCount = %d\r\n",iCount);

        ++iCount;
    }
    while (iCount < 5);

    printf("Exit loop: iCount = %d\r\n",iCount);
}
```



### 3. Lệnh for

```
int i = 0;
for (i=0, i<5, i++)
{
    //do something
}
```

## Bài tập:

Kết quả khi nhấn nút **for** thực hiện đoạn chương trình sau là gì?

```
#include "opdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszProperty)
{
    //loop count
    int iCount = 0;

    printf("\r\nExample 3\r\n");

    //for loop
    for (iCount=0; iCount<5; iCount++)
    {
        //do something
        printf("Executed loop: iCount = %d\r\n",iCount);
    }

    printf("Exit loop: iCount = %d\r\n",iCount);
}
```

## Nhóm lệnh điều khiển chương trình

### 1. Lệnh if-else

```
if (i < 5)
{
    //do something
}
else
{
    //do something else
}
```

### 2. Lệnh switch-case

```
switch (i)
{
    case 0: //do something
        break;
    case 1: //do something
        break;
    default: //do something default
        break;
}
```

## Bài tập:

Kết quả khi nhấn nút `switch-case` thực hiện đoạn chương trình sau là gì?

```
#include "apdetap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszProperty)
{
    BYTE byValue;

    //get value to check
    byValue = GetTagByte("U08i_course_loop_1");

    printf("\r\nExample 6\r\n");

    switch (byValue)
    {
        case 0: //do something
                printf("byValue = 0\r\n");
                break;
        case 1: //do something
                printf("byValue = 1\r\n");
                break;
        case 2:
        case 3:
        case 4: //do something
                printf("byValue = 2,3 or 4\r\n");
                break;
        default: //do something
                printf("byValue != 0,1,2,3 and 4\r\n");
                break;
    }
}
```

## Function và cách truyền đối số qua hàm

Ví dụ:

```
double MeanValue(double dValue1, double dValue2, double dValue3)
{
    double dMeanValue;
    dMeanValue = (dValue1+dValue2+dValue3)/3;
    return dMeanValue;
}
```

## Bài tập:

Kết quả khi nhấn nút **Value Parameter** thực hiện đoạn chương trình sau là gì?

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{

double dValue1 = 126.2;
double dValue2 = 23.9;
double dValue3 = 45.7;

double dMeanValue;

//calculate mean value
dMeanValue = MeanValue(dValue1, dValue2, dValue3);

//output into diagnostics window
printf("\r\nExample 1\r\n");

printf("The mean value of %.1f, %.1f and %.1f = %.1f\r\n",
       dValue1, dValue2, dValue3, dMeanValue);

}
```

# Sử dụng DDE Server - Dynamic Data Exchange - để liên kết dữ liệu của WinCC với Excel

## Hai cách để thiết lập kết nối DDE

### + Local DDE connection

Chạy WinCC trên máy cục bộ với Single-User System  
Hay trên máy Server với Multi-User System

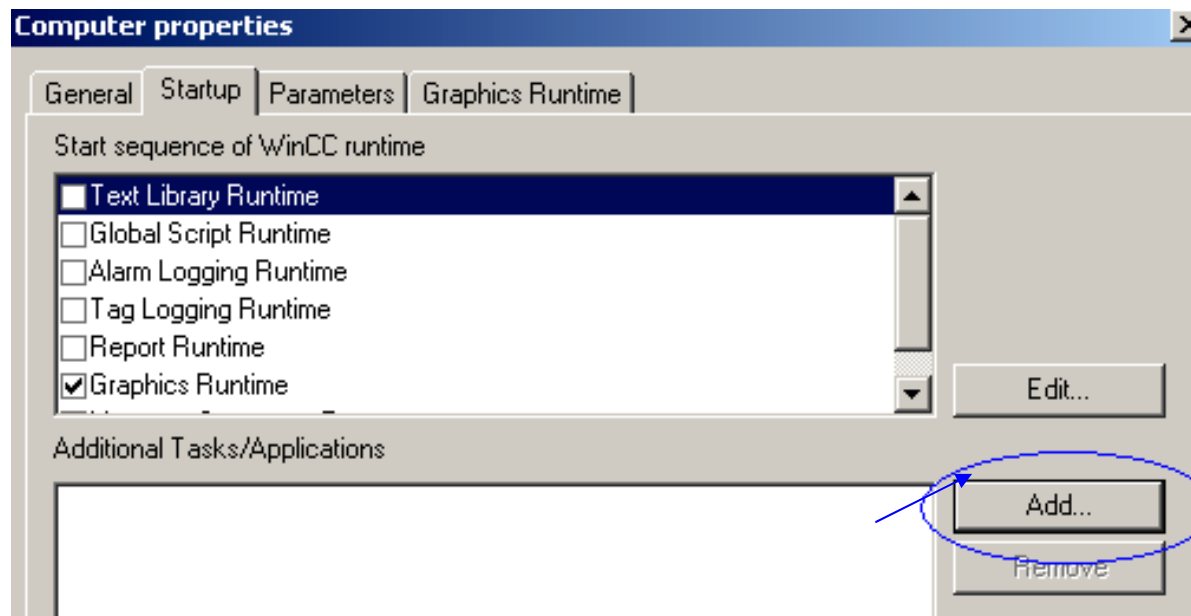
### + Kết nối DDE thông qua mạng

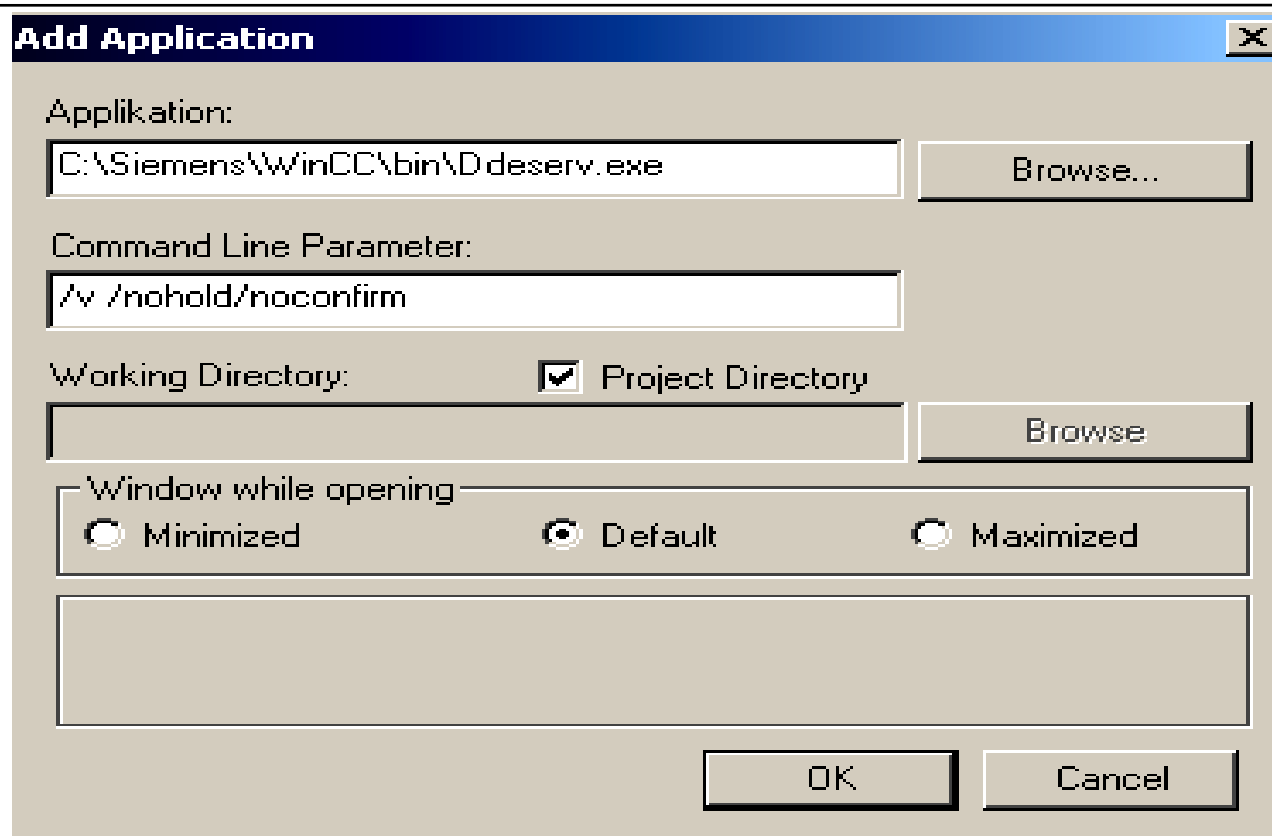
Hai cách thiết lập trên cần phải Add thêm Chương trình: DDEServer.exe với tham số “/v /nohold/noconfirm” vào Start-up list của Computer đang chạy WinCC



## Sử dụng Local DDE connection

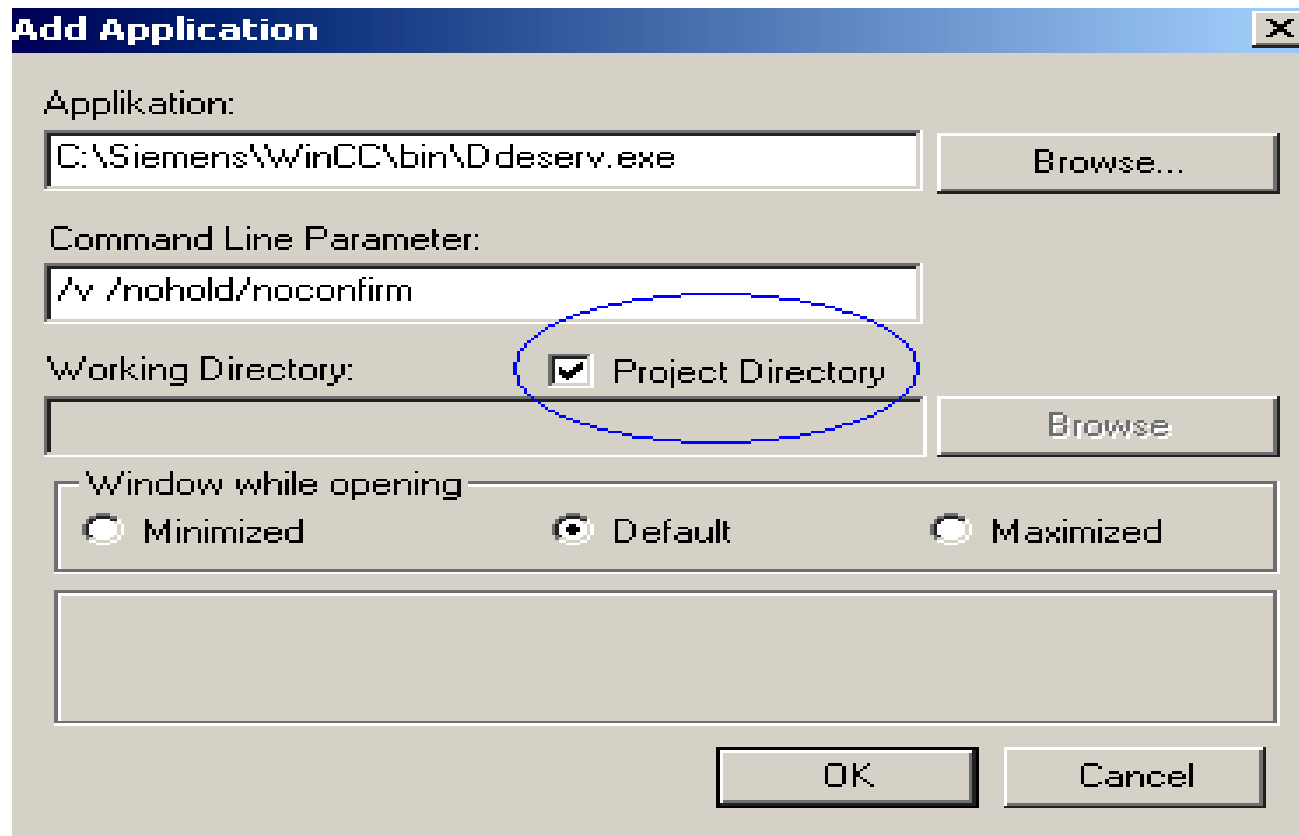
- + Chạy WinCC trên máy cục bộ với Single-User System hay trên máy Server với Multi-User System
- + Click chuột phải vào Computer chọn Properties → Chọn Properties lần nữa. Trên Tab Startup, nhấn nút Add





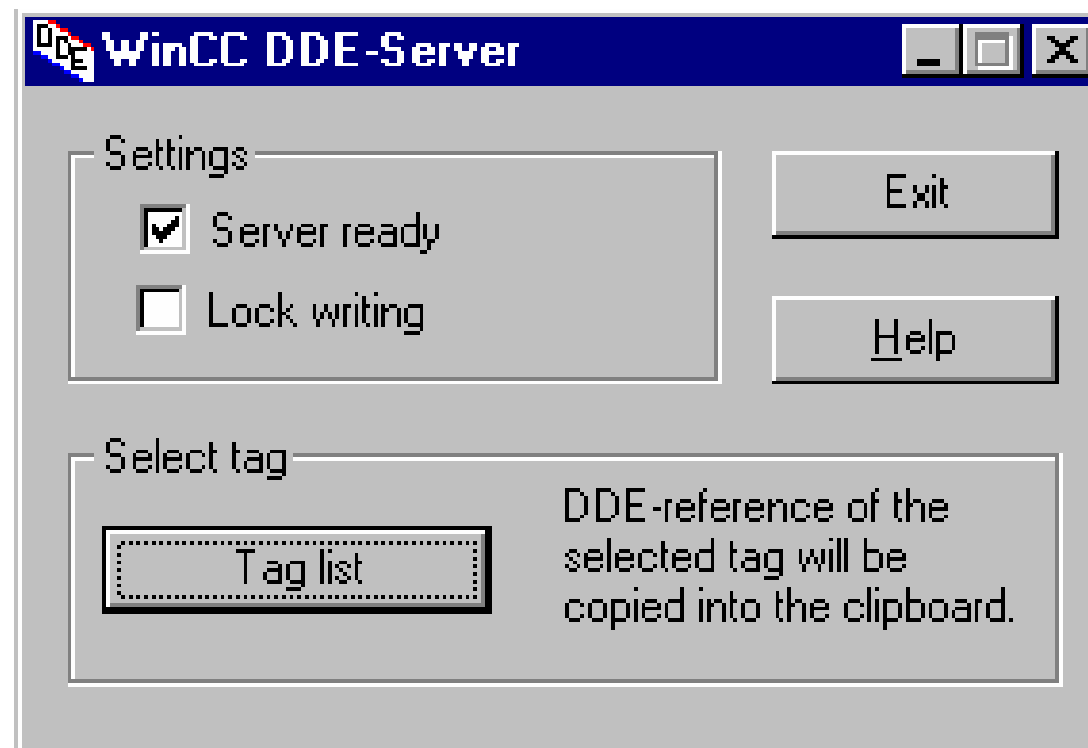
Nhấn Browse bên cạnh ô Application, chọn File DDEServer.exe (thường nằm trong đường dẫn "C:\SIEMENS\WINCC\BIN\DDESERV.EXE").  
Với tham số `"/v /nohold /noconfirm"`.

## Đánh dấu vào ô Project Directory ở mục Working Directory



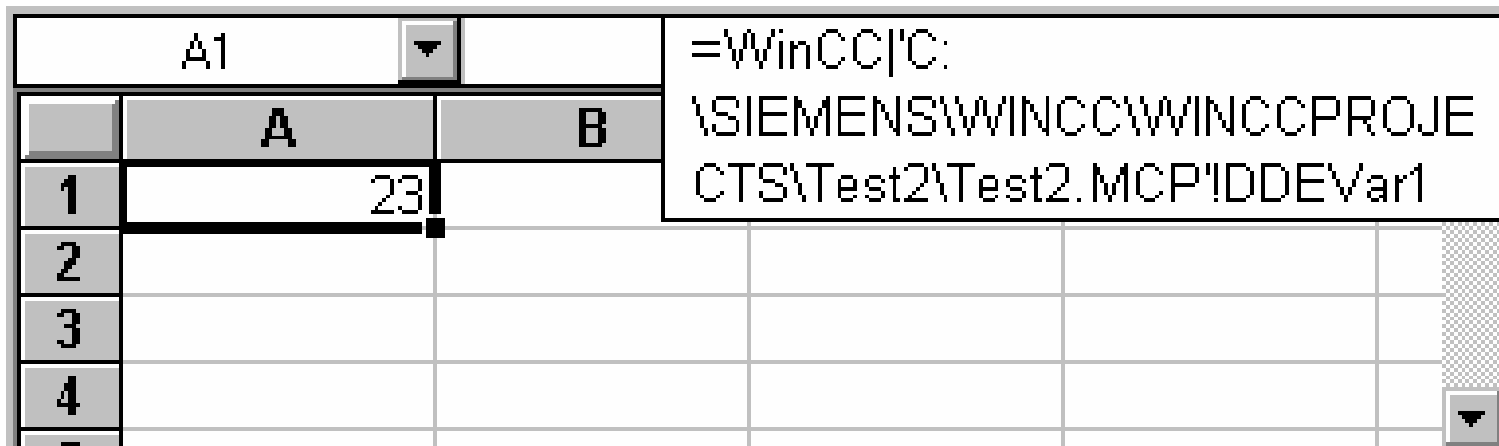
**Khởi động lại máy thì chương trình DDEServer mới thực thi được**

- + Sau khi khởi động lại máy, chuyển sang chạy Runtime cho WinCC Project.
- + Chọn Tag muốn lưu trữ dữ liệu từ cửa sổ WinCC DDE-Server.



## Thực hiện các bước sau để kết nối với excel

1. Click chuột vào Tag List để chọn Tag cần lưu trữ → Chọn OK để lưu vào Clipboard.
2. Mở chương trình Excel.
3. Chọn một cell muốn ghi giá trị của Tag vào.
4. Gõ dấu “=” vào Excel Formular Bar.
5. Nhấn Ctrl+V để Paste dữ liệu từ WinCC ra cell vừa lựa chọn.



## Bài tập thực hành cuối khoá

Thiết kế giao diện WinCC hoàn chỉnh để điều khiển và giám sát hệ thống FMS-500 cho từng trạm:

- + Distribution
- + Testing
- + Sorting
- + Handling