



Módulo 3 : Unity Pro
Sección 3 : Configuración PLC
Página 0/15

Contenido

Siguiente

Salida

Ayuda

Módulo 3.3

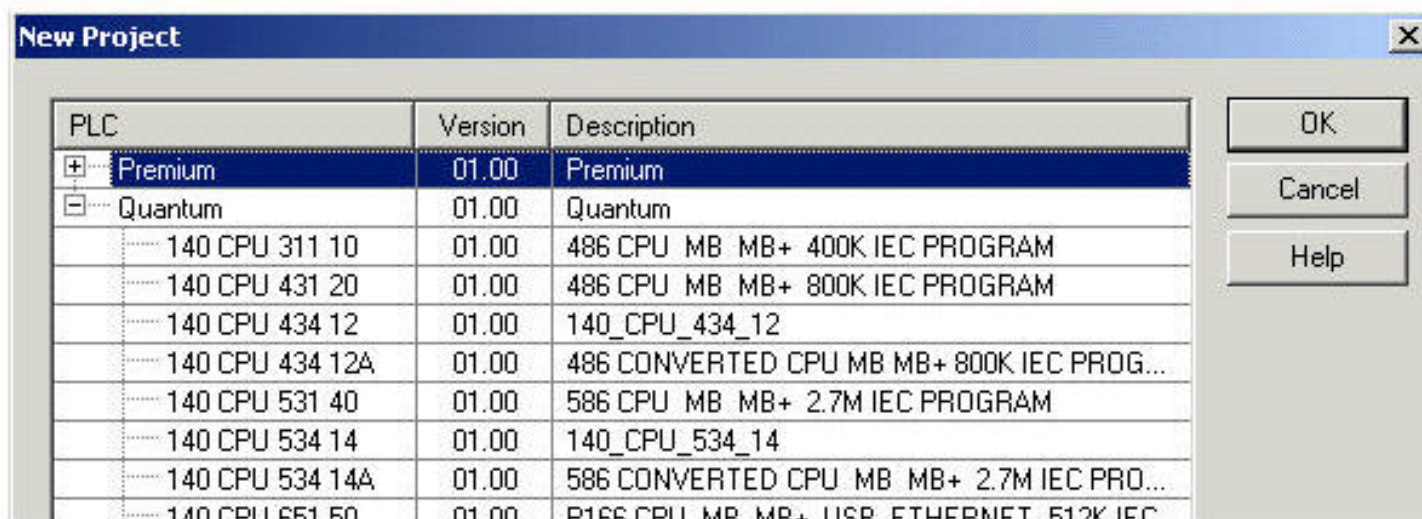
Configuración PLCs

Módulo 3 : Unity Pro

Sección 3 : Configuración PLC

Página 1/15

● Elección procesador



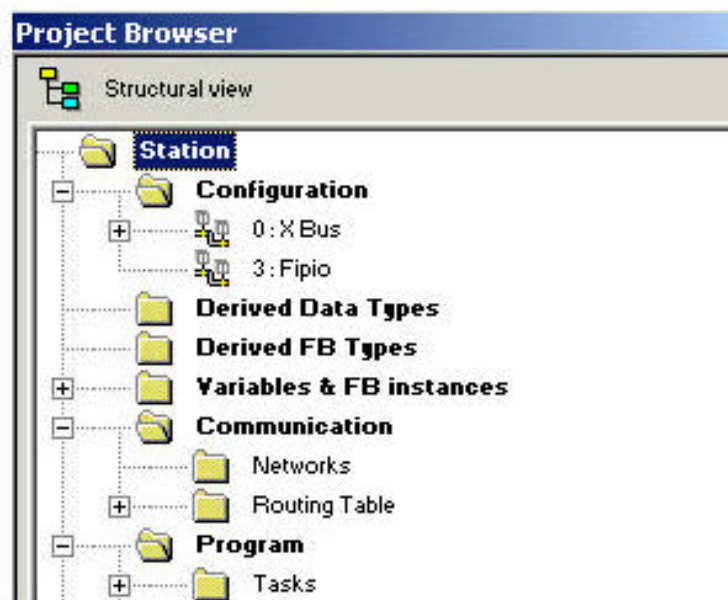
Primer paso para crear una aplicación

- Elegir la plataforma: Premium o Quantum (no intercambiable)
- Elegir el tipo de procesador

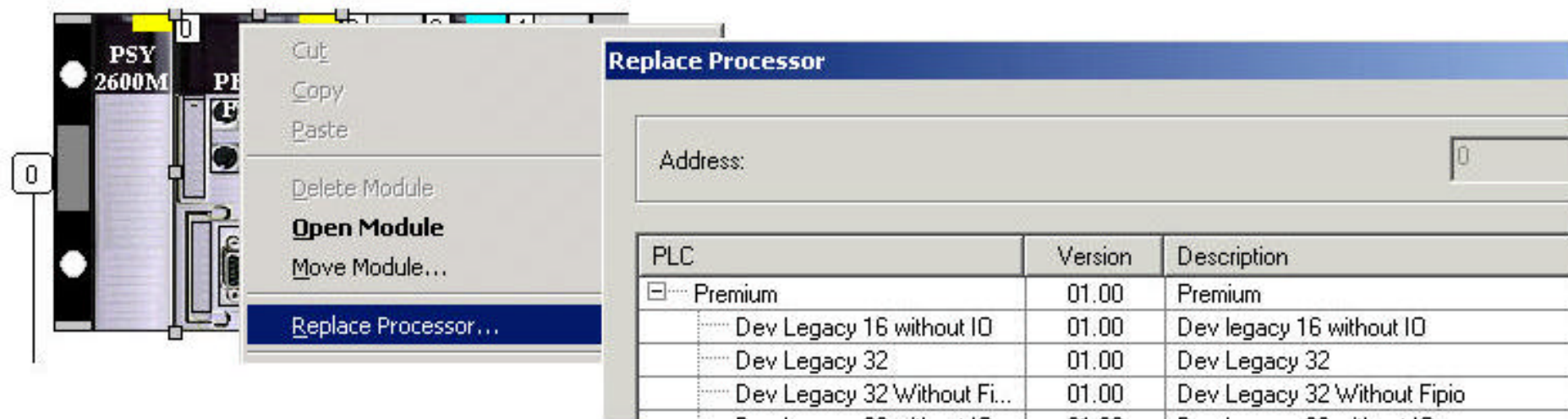
● Acceso al editor configuración

Accesible desde **Vista Estructural** para

- Configurar los racks del bus local
 - ▶ dirección 0 para los PLCs Premium
 - ▶ dirección 1 para los PLCs Quantum
- Configurar los racks remotos (X Bus o RIO)
- Configurar el bus de campo
- Enlazar hardware comunicación a red lógica (Ethernet, Modbus +, Fipway)



● Reemplazo procesador

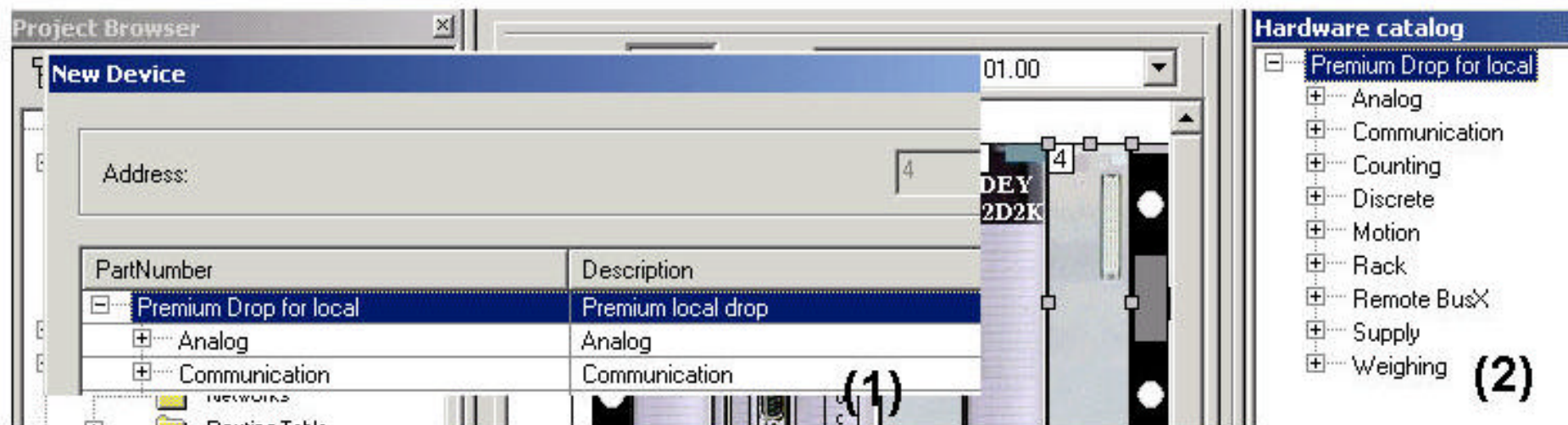


Para modificar el tipo de procesador

- Elija con botón derecho al procesador
- Solo se proponen miembros de la misma familia de procesadores (ej Premium)
- Reemplazo procesadores es posible solo en modo Fuera-de-Línea

Nota : Los procesadores Quantum pueden colocarse en cualquier parte del rack

● Configuración Premium

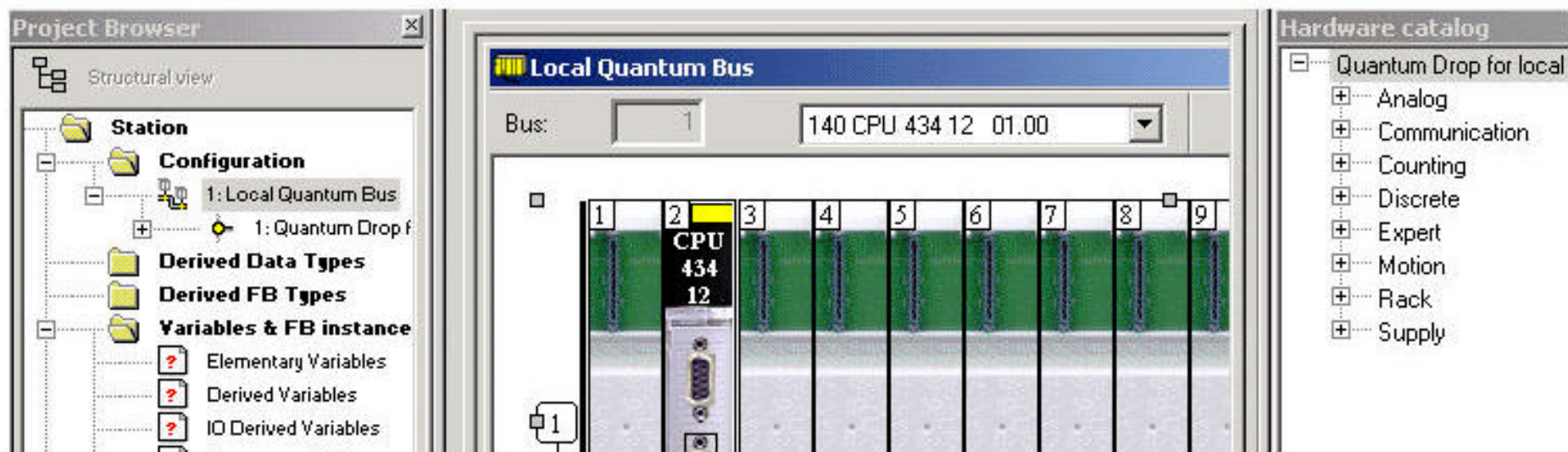


The screenshot displays the Unity Pro software interface. On the left, the 'Project Browser' shows a 'New Device' dialog box with an 'Address' field set to '4'. Below this is a table with columns 'PartNumber' and 'Description'. The table lists 'Premium Drop for local' (Premium local drop), 'Analog', and 'Communication'. A red circle with the number '1' is placed over the 'Communication' row. On the right, the 'Hardware catalog' panel shows a tree view with 'Premium Drop for local' expanded, listing 'Analog', 'Communication', 'Counting', 'Discrete', 'Motion', 'Rack', 'Remote BusX', 'Supply', and 'Weighing'. A red circle with the number '2' is placed over the 'Weighing' item.

PartNumber	Description
Premium Drop for local	Premium local drop
+ Analog	Analog
+ Communication	Communication

- Elegir el rack
- Definir el módulo fuente de poder (posición izquierda)
- Reemplazar al procesador (si es necesario)
- Definir los módulos: doble click en la posición vacía y añadir un dispositivo (1) o arrastrar uno desde el catálogo hardware (2)

● Configuración Quantum



- Elegir el rack
- Definir el módulo fuente de poder (se sugiere primera o última posición)
- Reemplazar o mover el procesador (si es necesario)
- Definir los módulos

● Configuración procesador Premium

Operating mode

Run/Stop input

Memory protect

Automatic start in Run

%MWi Reset on cold start

Size of global address fields

%M: 512 %MW: 2048 %KW: 256

%S: 128 %SW: 168

Memory cards

A: TSX MRP C 01M7

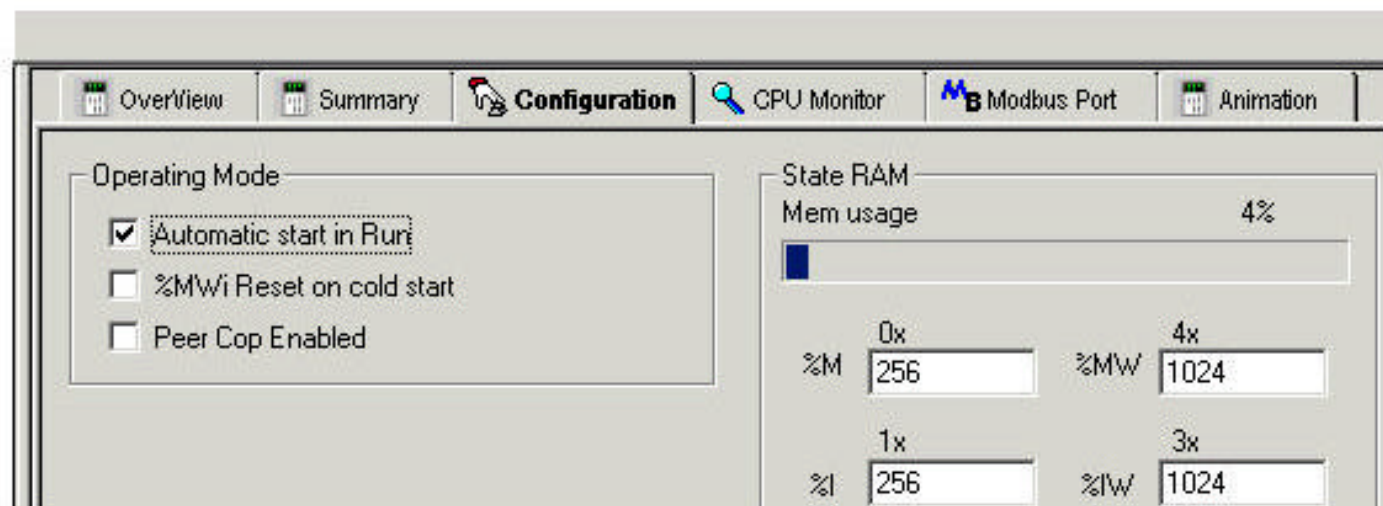
Usage: Data Storage Size: 1280 KBytes

Application size: 512 KBytes

Default values

- Seleccionar modo operación: Entrada Marcha/Paro, protección memoria, etc...
- Definir las tarjetas memoria
- Definir los objetos globales de la aplicación: número de bits y palabras

● Configuración procesador Quantum



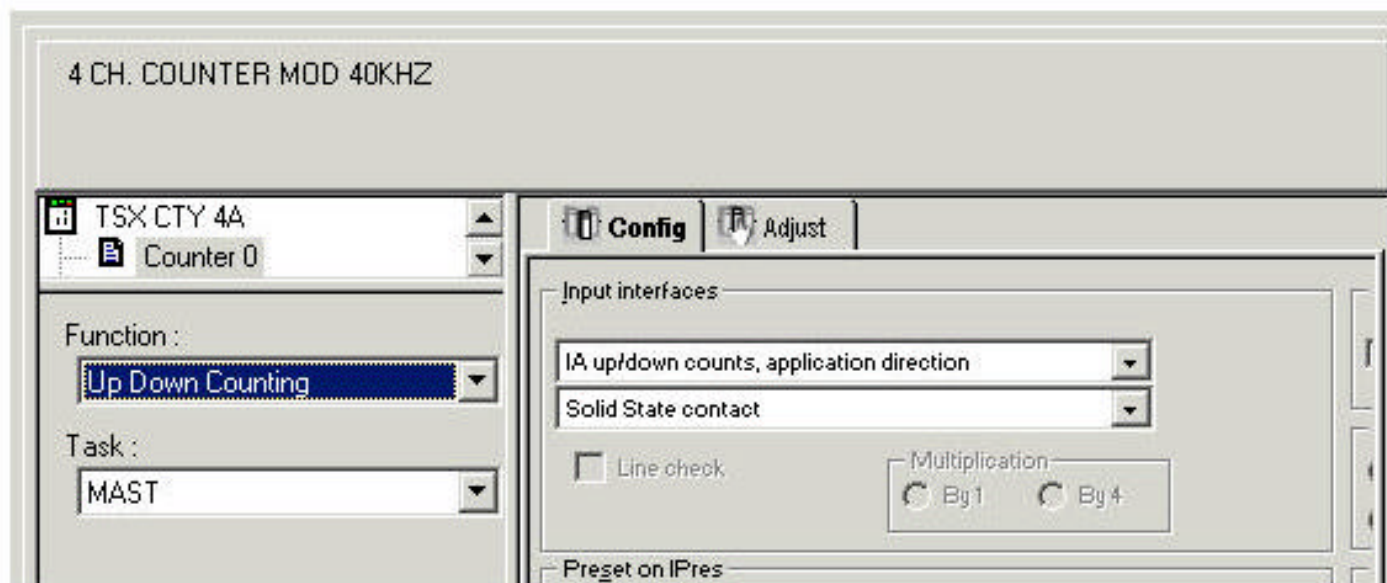
- Seleccionar modo operación: Arranque automático en Marcha, Peer Cop, etc...
- Definir los objetos globales de la aplicación: número de bits y palabras

Módulo 3 : Unity Pro

Sección 3 : Configuración PLC

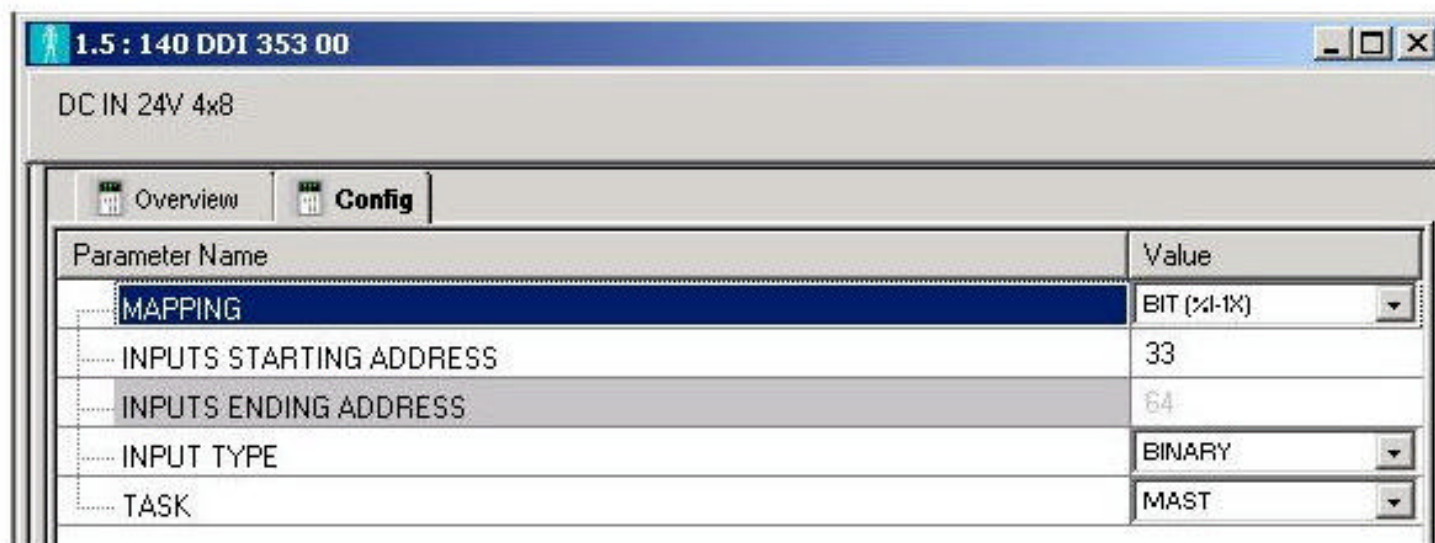
Página 8/15

● Configuración Módulo



- Vista general del módulo
- Configuración de los parámetros de cada canal (en modo local)
- Ajuste de canales en modo local (módulo experto)

● Direccionamiento módulo Quantum



Parameter Name	Value
MAPPING	BIT (%I-IX)
INPUTS STARTING ADDRESS	33
INPUTS ENDING ADDRESS	64
INPUT TYPE	BINARY
TASK	MAST

- Definir las direcciones usadas por el módulo Quantum: mapeo (bit, palabra) y primera dirección (ejemplo: 33)
- Definir los otros parámetros: tipo, tarea, etc...

● Direccionamiento topológico

● Direccionamiento físico de entrada / salida

%	I / Q	X / W / D	r	.	m	.	c	.	d	.	j
Símbolo	Tipo I = Entrada Q = Salida	Formato X = Booleano W = Palabra D = Doble Palabra	Rack		Módulo		Canal		Rango		Bit

● Direccionamiento topológico de datos localizables Quantum

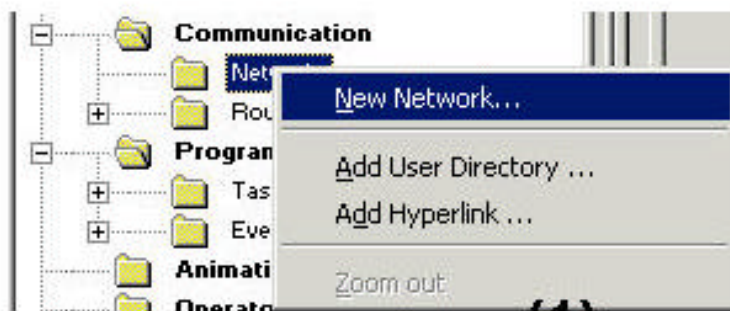
Dirección	Dirección Topológica	Uso Datos
0xxxxx	%Qr.m.c.d / %Mi	Bits Módulo Salida y bits Internos
1xxxxx	%lr.m.c.d / %li	Bits Módulo Entrada
3xxxxx	%lWr.m.c.d / %lWi	Palabras Módulo de Entrada o E/S
4xxxxx	%QWr.m.c.d / %MWi	Palabras Módulo Salida o Entrada/Salida y palabras internas

● Principio para configurar una red

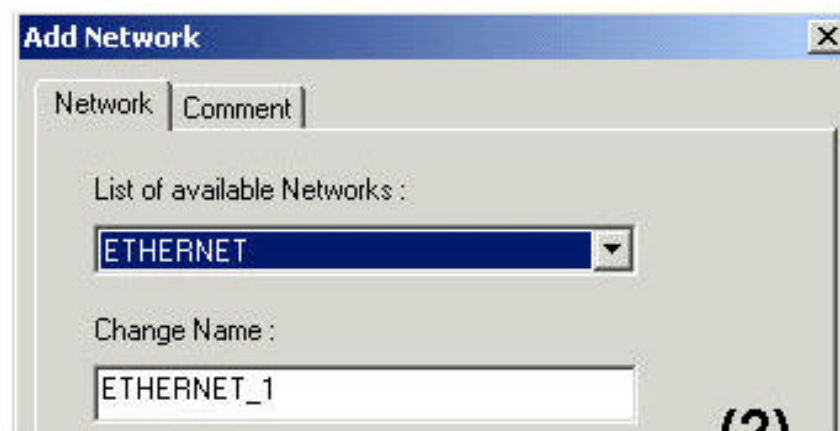
Para configurar una red (Ethernet, Modbus + o Fipway) :

- Crear la red lógica
- Configurar la red lógica
- Definir el módulo comunicación o tarjeta PCMCIA
- Asociar el módulo o tarjeta PCMCIA a la red lógica

● Crear una red lógica



(1)



(2)

- Añada una nueva red (click derecho sobre el Folder **Redes** del Explorador Aplicación) (1)
- Elegir el tipo de red a crear (Ethernet, Modbus +, Fipway) y definir su nombre (2)
- Si es necesario, adicionar un comentario

● Configurar la red lógica



(1)

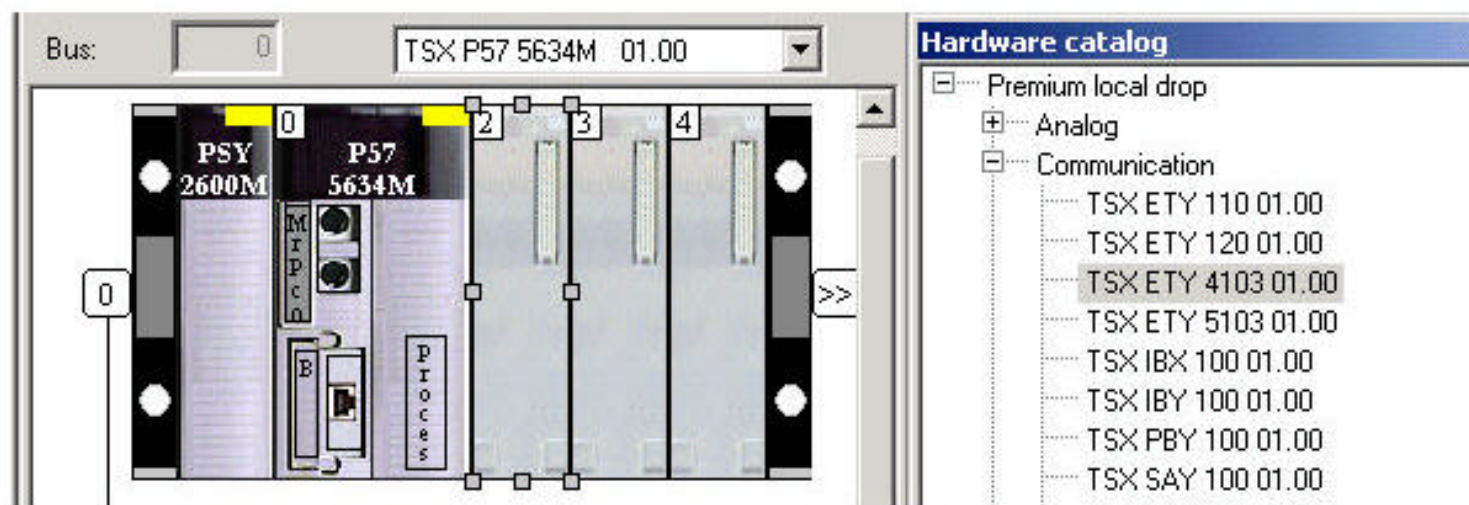
- Activar la red lógica a configurar (1) (2)
- Configurar la red lógica: Datos Globales, Exploración E/S, etc... (2)

Módulo 3 : Unity Pro

Sección 3 : Configuración PLC

Página 14/15

● Definir al módulo comunicación / PCMCIA



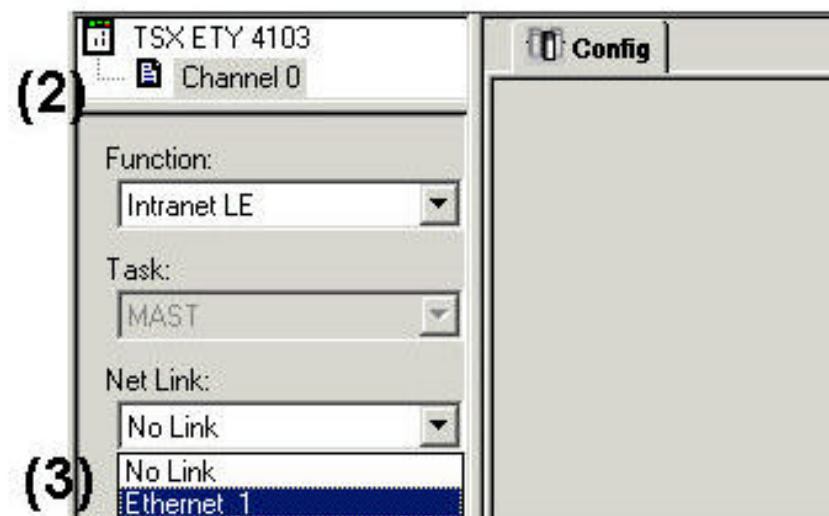
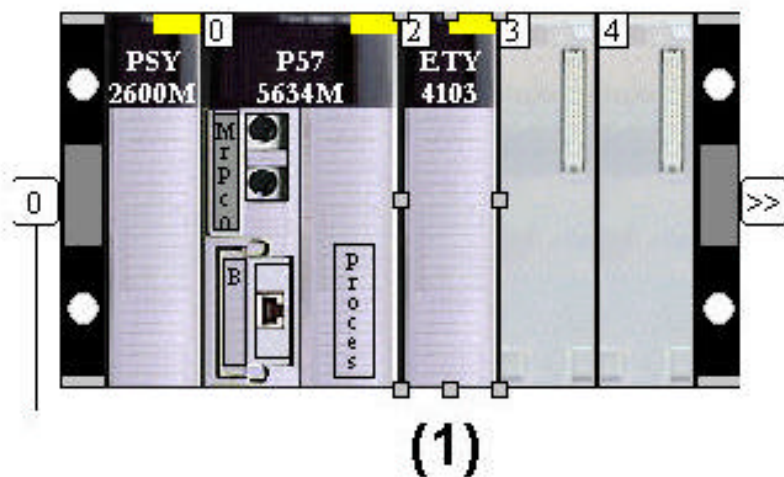
- Definir el módulo comunicación (arrastre & coloque desde el catálogo hardware)
- O definir la tarjeta PCMCIA (doble click sobre la posición de la tarjeta y añade el submódulo)

Módulo 3 : Unity Pro

Sección 3 : Configuración PLC

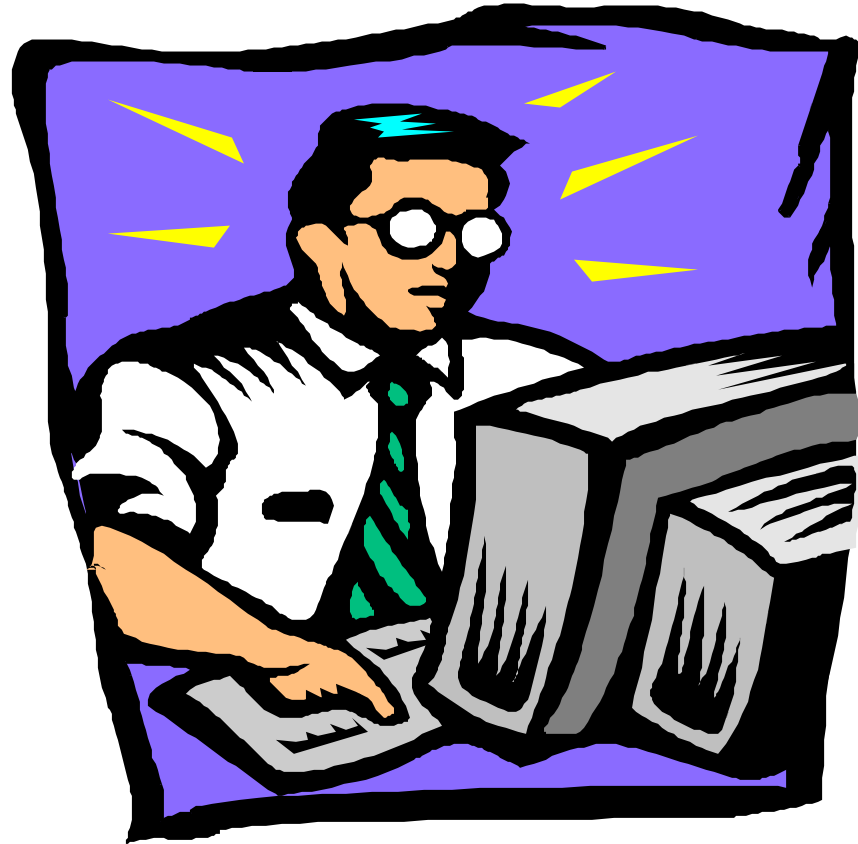
Página 15/15

● Asociar el módulo / tarjeta y la red lógica



- Abrir el módulo comunicación (1)
- Seleccionar el canal (2)
- Asociar el módulo a la red lógica (3)

Ejercicio 3





Módulo 3 : Unity Pro
Sección 4 : Variables
Página 0/22

Contenido

Siguiente

Salida



Ayuda

Módulo 3.4

Variables

● Variables

Una **Variable** es una entidad memoria del tipo BOOL, WORD, DWORD, etc., cuyo contenido puede modificarse por el programa durante la ejecución.

Una **Variable Localizable** es una variable que está mapeada a un Módulo E/S o se asoció a una referencia memoria. Por ejemplo, la variable `Water_pressure` se asocia con la Palabra Memoria `%MW102`. `Water_pressure` se dice que es una Variable Localizable.

Una **Variable No Localizable** es una variable que no está mapeada a E/S ni asociada a referencia memoria (no es posible determinar su posición en memoria). Una variable que no tiene asignada dirección se dice que es No Localizable.

● Variables (cont)

Una **Variable Pública** es una variable disponible en ciertos Bloques Función. Estas variables transfieren valores estadísticos (valores que no son influenciados por el proceso) al bloque función. Se usan para ajuste de parámetros del bloque función.

I/ODDT es la abreviación de **Input/Output Derived Data Type**. El término I/ODDT designa un tipo estructurado datos representando un módulo o un canal de un módulo PLC. Cada módulo experto aplicación posee sus propios I/ODDTs.

Constante es una variable tipo INT, DINT o REAL ubicada en el campo constante (%K), o variables usadas en direccionamiento directo (%KW, %KD, ó %KF). Su contenido no puede modificarse por el programa durante ejecución.

● Variables (cont)

Cada Variable debe declararse con el Editor Variables Datos antes de usarla:

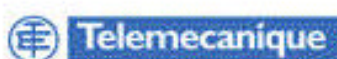
- Ya sea de modo independiente o con doble click en el área Variables & Instancias FB del árbol Vista Estructural...
- O durante la programación con un doble-click en la asignación de una entrada / salida.

Un tipo dato debe asignarse a cada Variable.

Unity Pro software proporciona tipos de datos elementales y derivados

● Tipos Datos Comunes y Rangos

- **BOOL/EBOOL**: Debe ser ON (1) ú OFF (0)
- **WORD**: Representa una “cadena 16 bits”, lo que significa que la longitud de datos es de 16 bits
- **INT**: Representa un valor entero. El rango de valores va desde -32768 hasta 32767 (15 bits más signo)
- **UINT**: Representa un valor entero sin signo . El rango de valores va desde 0 hasta 65535 (16 bits)
- **REAL**: Representa un valor puntoflotante. El rango de valores va desde $8.43e-37$ hasta $3.36e+38$



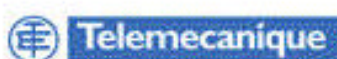
● Ingresando Valores Literales

- Valores Literales se usan para asignar valores a pines, o para asignar valores constantes a variables, y no fueron hechos para ser modificados por el programa.

Pueden ingresarse valores literales como

- Base 2 (binario) 2#1111111111111111
- Base 8 (octal) 8#177777
- Base 10 (decimal) 65535 (no se necesita anteponer 10#)
- base 16 (hex) 16#FFFF

Todos los valores anteriores son iguales, solo que fueron ingresados en un formato diferente



● Nombres Variables

- Long. máxima 32 caracteres, puede empezar con número

No Localizable:

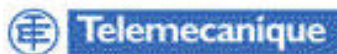
- Nombre sin una dirección de hardware.
- Implementada en el código, por lo que se procesa más rápido que una variable localizable.

Localizable:

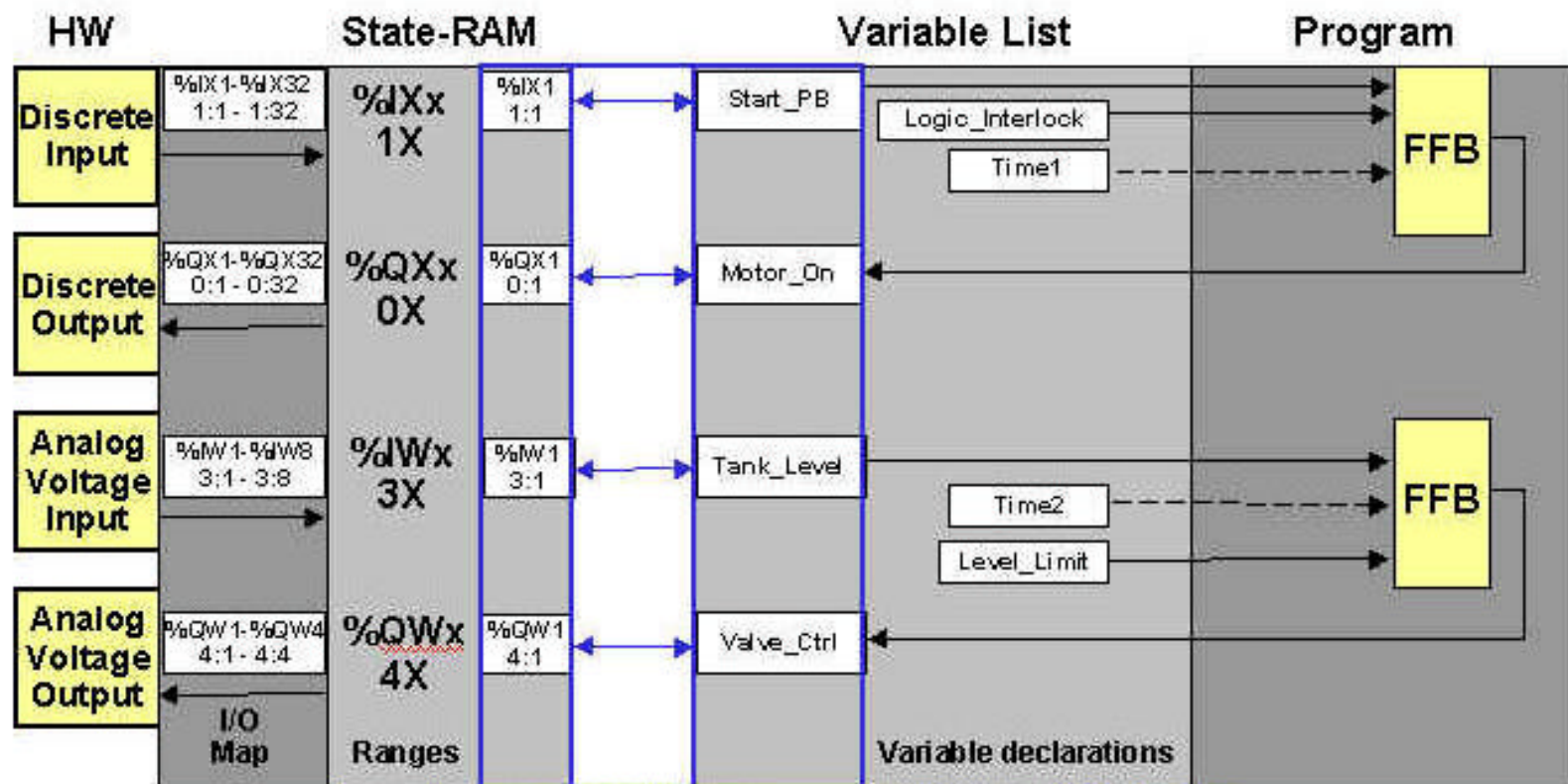
- Nombre con una dirección hardware. (State RAM.)
- Not implementada en el código, por lo que no se procesa tan rápido como una variable no localizable.

Constantes:

- Usadas para definir valor a un nombre, ejemplo, Hi_Temp = 103.7
- No pueden ser localizables.



Configuración, Direccionamiento y Asignación de Variables

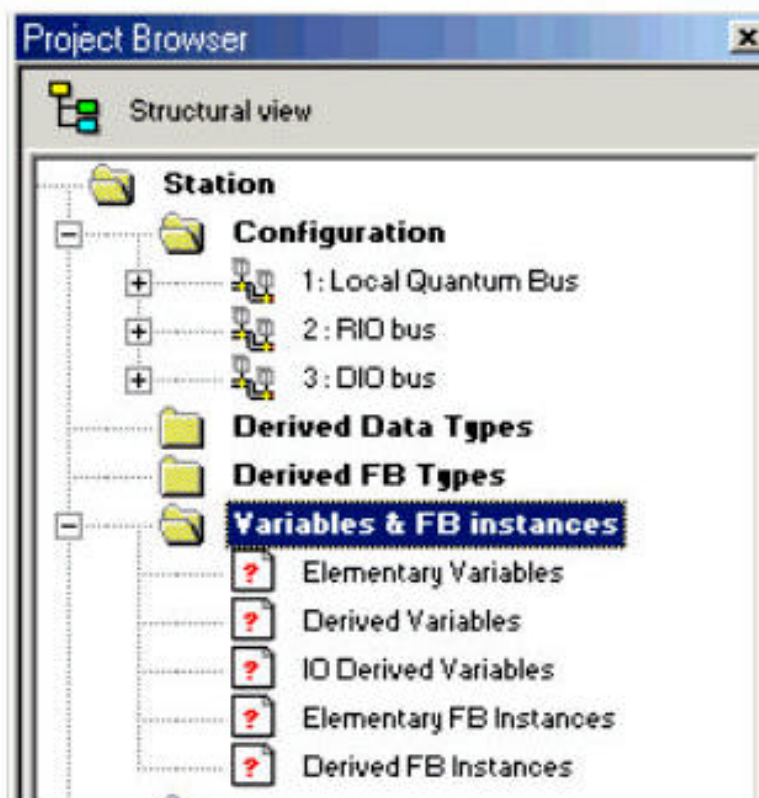


● Direccionamiento Directo (Quantum)

- Cada dirección directa tiene una referencia que indica su posición en la secuencia ya sea una dirección de entrada (solo lectura) o una dirección de salida (lectura / escritura).
 - ▶ Area 0x / %Mx = bit salida (discreta)
Notación **ANTIGUA**: 000001 es Bit Salida Discreta 1
Notación **IEC**: %M00001 es Bit Salida Discreta 1
 - ▶ Area 1x / %Ix = bit entrada (discreta)
Notación **ANTIGUA**: 100017 es Bit Entrada Discreta 17
Notación **IEC**: %I00017 es Bit Entrada Discreta 17
 - ▶ Area 3x / %IWx = registro entrada (palabra)
Notación **ANTIGUA**: 300300 es Registro Entrada Palabra 300
Notación **IEC**: %IW00300 es Registro Entrada Palabra 300
 - ▶ Area 4x / %MWx = registro salida (palabra)
Notación **ANTIGUA**: 400029 es Registro Salida Palabra 29
Notación **IEC**: %MW00029 es Registro Salida Palabra 29

● Directorio Variables e Instancias FB

**Ver
Comentarios**



● El Directorio Variables e Instancias FB:

Directorio Variables:

El directorio Variables e Instancias FB de la Vista Estructurada del proyecto permite acceder las variables (EDT, DDT, IODDT) y las instancias bloque función (EFB, DFB).

La ilustración muestra un ejemplo de un directorio de las Variables e Instancias FB.

Utilerías accesibles

El directorio Variables e Instancias FB permite acceder las utilerías siguientes, las cuales pueden accederse vía el menú:

Directorio

Variables e Instancias FB

Variables Elementales

Variables Derivadas

Variables Derivadas E/S

Instancias FB Elementales

Instancias FB Derivadas

Utilerías

Abrir: permite acceder el editor variables,

Exportar: permite acceder **la exportación de todas las variables** del proyecto,

Importar: permite acceder **la importación de todas las variables** del proyecto,

Abrir: permite acceder el editor variables,

Exportar: permite acceder **la exportación de todas las variables** de la familia seleccionada (EDT, DFB, etc.) .

● Editando Variables

**Ver
Comentarios**

Data Editor

Variables | ODT types | Function Blocks | DF3 types

Filter: Name * [] [x] EDT [x] DDT [x] IO3DT [3]

Dataname	Type	Address	Value	Comment
ack	Bool			Acknowledgement
alarm	Bool			General alarm
Analog_input_1	T_ANA_I...	%CH4.0.1		channel 1
auto_cmd	Bool			and in auto mode
auto_mode	Bool			anu mode
blocking	Bool	%IO.0.4		blocking
MotorFeedingLimitSwitch				
cmd	Bool			
conveyor_belt_ctl	T_DIS_IN...	%ch3.0.0.3		
CH_ERROR	Bool	%I3.0.0.ERR		
VALUE	EBool	%I3.0.0.0		
default	Bool			
Drilling	SFCSTEP...			
Drilling_approach	SFCSTEP...			
drilling_motor	motor			

Context Menu:

- Copy
- Paste
- Cut
- Delete
- Insert new
- Properties
- Go to type definition
- Search
- Columns
- Add Watch
- Expand Data
- Contract Data
- Expand all Data
- Add Hyperlink
- Customization
- Sort

Annotations:

- 1: Identifier
- 2: Identifier
- 3: Filter checkboxes
- Unlocated Variable: ack, alarm
- Located Variable: CH_ERROR
- Data Type: Type column

Callout: Double-click on cells to enter in edit mode

● Editando Variables:

Definición Tipos e declaración instancias en UNA HERRAMIENTA

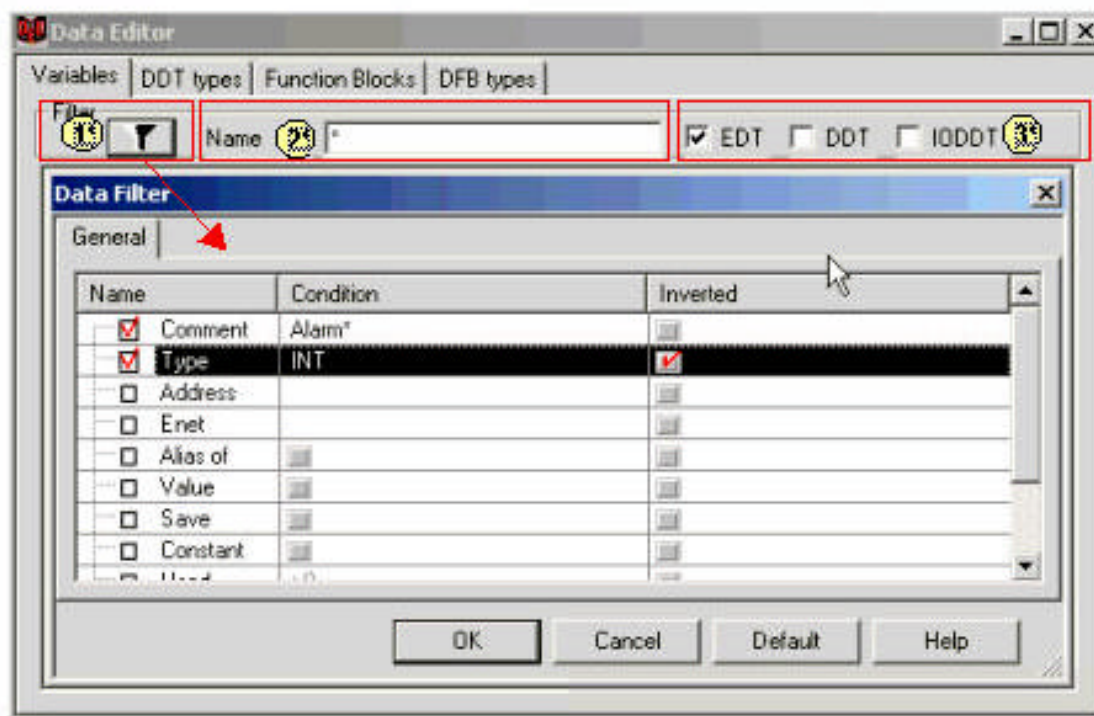
- Funcionalidad Copiar/Cortar/Empastar
- Búsqueda variable
- Hiperenlace sobre el comentario de cualquier variable
- Particularización Columnas
- Despliegue de todas las propiedades en una ventana específica
- Contraer/Expandir datos estructurados (1)
- Sortear por tipo, símbolo, dirección... (2)
- Filtrado (3)
- Selección tip desde una lista o desde la selección tipo variable
- facilidades Arrastre-Pegue entre editores y fuera de Unity Pro
- Característica UNDO en un nivel (última modificación)
- Importar/Exportar

Respecto a Variables:

- Promover el uso de símbolos
- Simbolizar todos los objetos simples o estructurados
- Asociación mapeo topológico y/o valor inicial
- simbolización automática variable E/S

Filtrado

**Ver
Comentarios**



● Filtrado

(1) Filtro permite despliegue versátil de una parte de la base datos variables .

El nombre de la columna a ser filtrada puede seleccionarse y debe darse una condición, tal condición puede invertirse como opción.

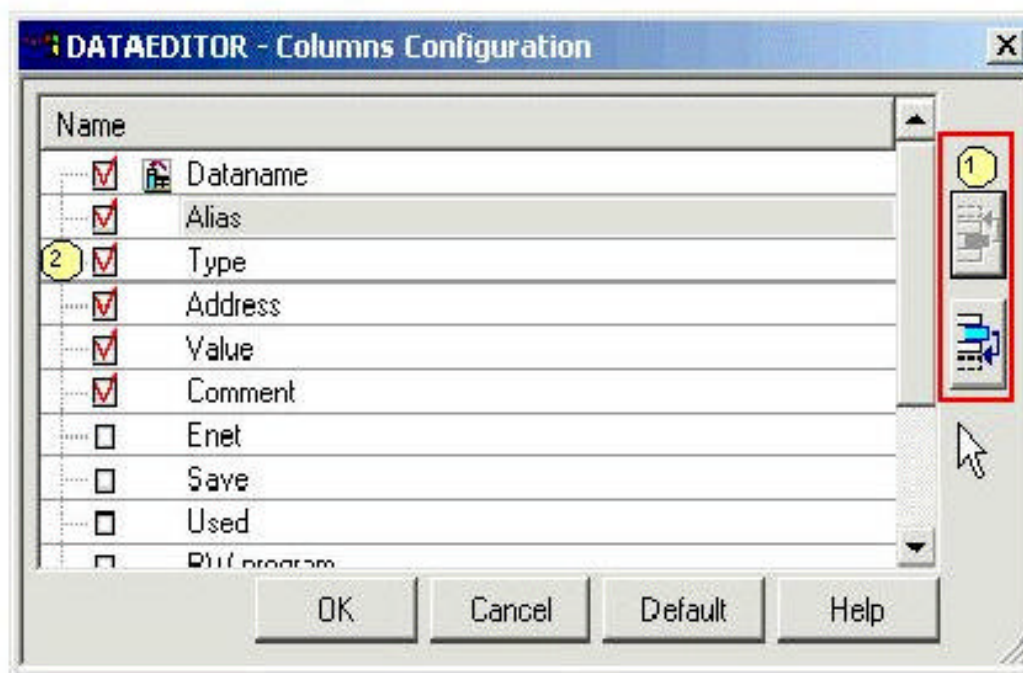
En el ejemplo superior, el filtro desplegará TODAS las variables incluyendo Alarma en el inicio del comentario y el tipo es diferente del INT.

(2) El nombre de la variable puede filtrarse directamente. Comodines , como ? (cualquier caracter) o *(todos caracteres) pueden reemplazar caracteres

(3) Filtrado directo en el tipo general de la variable también es posible eligiendo las opciones EDT, DDT o IODDT.

● Propiedades Datos

**Ver
Comentarios**

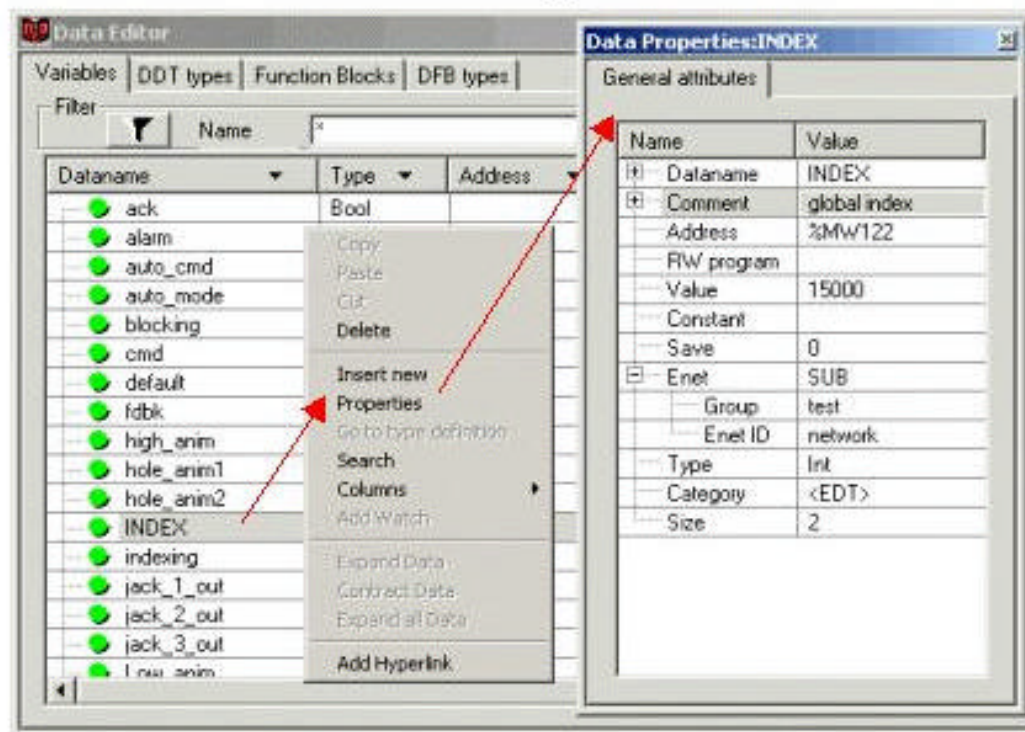


● Propiedades Datos

Columna configuración (particularización) despliega los parámetros variables tal y como el usuario lo desea.

- (1) Columnas pueden moverse con los botones superiores a la derecha
- (2) Una opción al frente del nombre de la columna, al elegirse, permite su despliegue

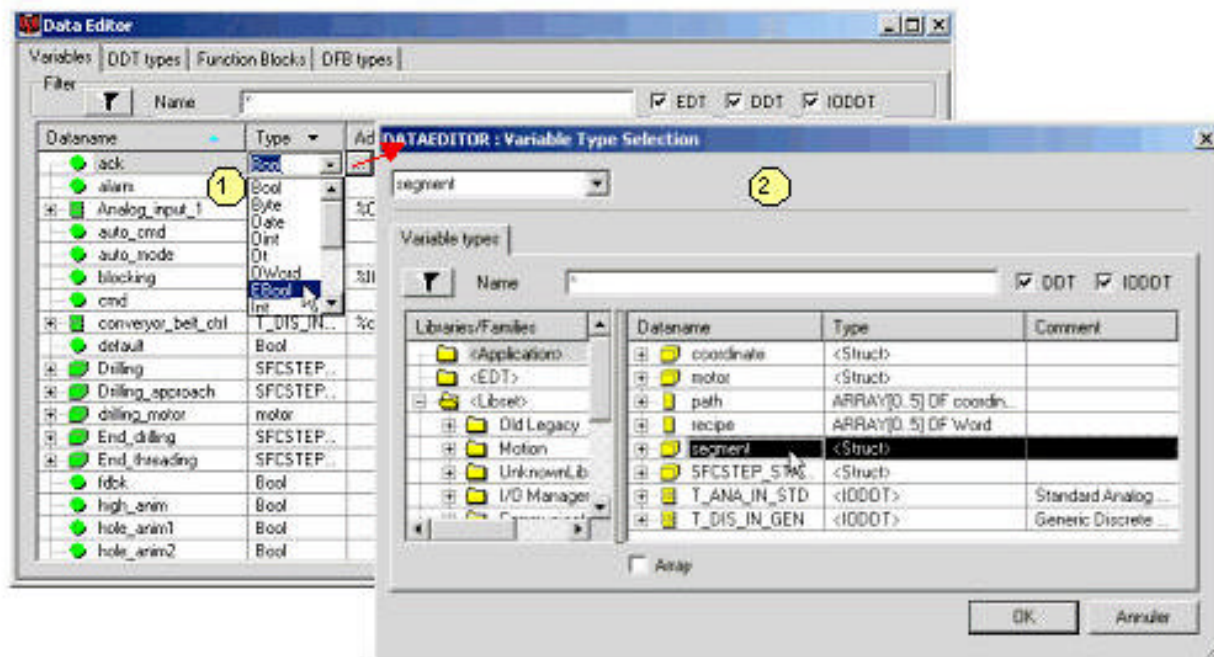
● Propiedades Datos (cont)



Name	Value
(R) Dataname	INDEX
(R) Comment	global index
Address	%MW122
RW program	
Value	15000
Constant	
Save	0
<input checked="" type="checkbox"/> Enet	SUB
Group	test
Enet ID	network
Type	Int
Category	<EDT>
Size	2

Aún si no se despliegan las columnas, todos los parámetros de una variable pueden mostrarse en la Ventana Propiedades Datos.

● Editar Tipo Datos



El tipo de datos puede seleccionarse a partir de una lista de tipos usados anteriormente (1) o directamente desde la selección Tipo Variable eligiendo el botón [...] (2).

● IODDT

**Ver
Comentarios**

Data Editor

Variables | DDT types | Function Blocks | DFB types

Filter: Name [] EDT DDT IODDT

Dataname	Alias	Type	Address	V
reset_switch		T_DIS_IN_GEN	%ch0.0.3.1	
paint_station_sensor		T_DIS_IN_GEN	%CH0.0.3.2	
initiate_transfer		T_DIS_IN_GEN	%ch0.0.3.0	
conveyer_belt_ctl		T_DIS_IN_GEN	%ch3.0.0.3	
temp_feeding_box		Int	%IW4.0.1.0	
Analog input 1		T_ANA_IN_STD	%CH4.0.1	
CH_ERROR		Bool	%I4.0.1.ERR	
temp_feeding_box (.VALUE)	temp_feeding_box	Int	%IW4.0.1.0	
EXCH_STS		Int	%MW4.0.1.0	
STS_IN_PROGR		Bool	%MW4.0.1.0.0	
CMD_IN_PROGR		Bool	%MW4.0.1.0.1	
ADJ_IN_PROGR		Bool	%MW4.0.1.0.2	
EXCH_RPT		Int	%MW4.0.1.1	
STS_ERR		Bool	%MW4.0.1.1.0	

● IODDT

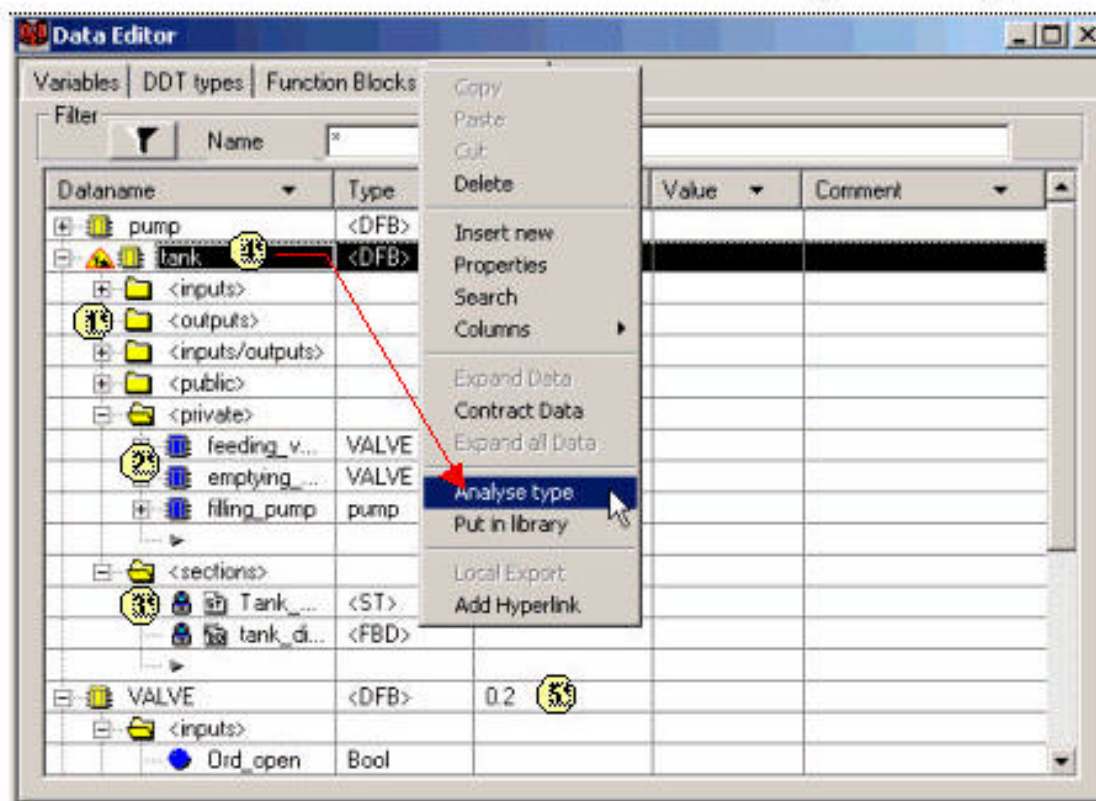
El Tipo específico IODDT permite mapear una estructura completa E/S (1) a partir del canal de un módulo (2) directamente con un solo nombre datos (3). (Es dependiente de la plataforma)

La estructura es mapeada automáticamente a los bits y palabras de entrada/salida (4).

Un Alias (5) puede usarse para darle un nombre usuario a la variable de una estructura particular.

IODDT permite la programación sin una dirección física, lo cual simplifica la creación de módulos estándar, el mapeo puede efectuarse después de la importación.

● Editando Bloques Función Derivados (DFB)



**Ver
Comentarios**

● Propiedades Datos (cont)

Tipos Bloques Función (DFB):

- (1) Creación de todos tipos variables de DFB: in, out, in / out, private, public
- (2) DFBs Anidados
- (3) Creación del programa DFB: secciones en LD, FBD, IL y ST. Son posibles múltiples secciones en un DFB
- (4) Analizar y generación código, un ícono pequeño informa que el DFB está bajo trabajo.
- (5) Gestión Versiones y Librería

● Editando Bloques Función (EFB, DFB)

Data Editor

Variables | DDT types | Function Blocks | DFB types

Filter: Name: EFB DFB

Dataname	n°	Type	Value	Comment
BlockingValve		VALVE		
diag_app		D_PRE		
FBI_285_2		TON		
FBI_285_3		TON		
FBI_285_4		CTUD		
FBI_285_5		TON		
FBI_395_4		TON		
FBI_395_6		SR		
indexing_valve		VALVE		
pump1		pump		

**Ver
 Comentarios**

● Editando Bloques Función (EFB, DFB):

Bloques Función (EFB, DFB):

- Symboliza y despliega instancias Bloques Función en aplicación cliente.
- La simbolización de Bloques Función puede llevarse a cabo durante la programación directamente en el editor de programa.



Tipos Datos Derivados

Data Editor

Variables | DDT types | Function Blocks | DFB types

Filter: Name *

Dataname	Type	Value	Comment
recipe	ARRAY[0..5] OF Word		
coordinate	<Struct>		
x	Real		
y	Real		
z	Real		
segment	<Struct>		
start	coordinate		
end	coordinate		
path	ARRAY[0..5] coordinate		
path[0]	coordinate		
x	Real		
y	Real		
z	Real		
path[1]	coordinate		
path[2]	coordinate		
path[3]	coordinate		
path[4]	coordinate		
path[5]	coordinate		

**Ver
Comentarios**

● Tipos Datos Derivados:

Un Tipo Dato Derivado es una definición de una estructura de cualquier tipo predefinido (EDT, DDT).

Pueden estar anidados (hasta ocho niveles), incluyendo arreglos (hasta de 6 dimensiones) de tipos datos elementales (1) o tipos de datos derivados (3).

Tipos Variables (DDT)

Data Editor

Variables | DDT types | Function Blocks | DFB types

Filter: Name * EDT DDT IODDT

Dataname	Type	Address	Value	Comment
welding_path	path	%Mw100		
welding_path{0}	coordinate	%Mw100		
x	Real	%Mw100	0.0	
y	Real	%Mw102	10.0	
z	Real	%Mw104	10.0	
welding_path{1}	coordinate	%Mw106		
x	Real	%Mw106	100.0	
y	Real	%Mw108	10.0	
z	Real	%Mw110	10.0	
welding_path{2}	coordinate	%Mw112		
welding_path{3}	coordinate	%Mw118		
welding_path{4}	coordinate	%Mw124		
welding_path{5}	coordinate	%Mw130		

**Ver
Comentarios**

● Tipos Variables (DDT):

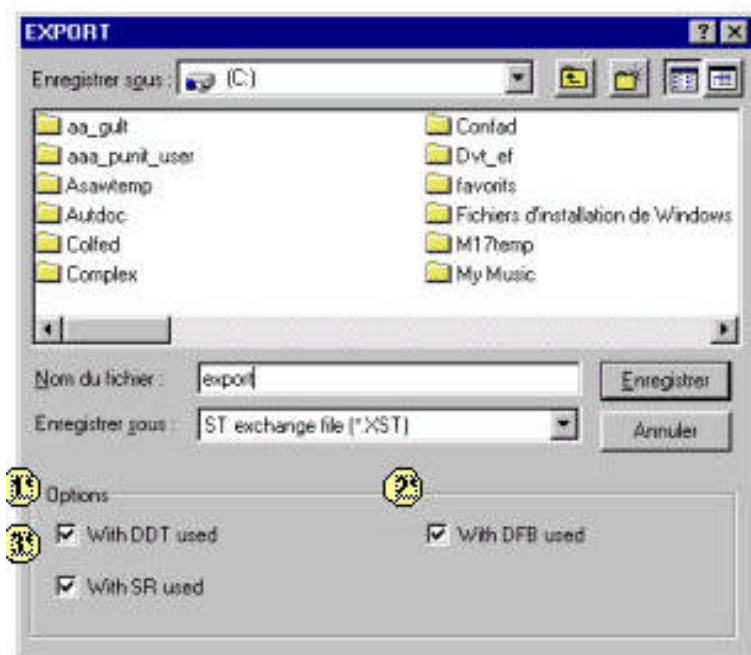
1 Definición Estructura o Arreglo

2 Estructuras Anidadas

DDTs pueden utilizarse como cualquier tipo de datos elementales para una instancia de una variable. Pueden mapearse a memoria física, si fuera necesario (4).

● Funciones Exportación

- Esta opción permite exportar toda una sección con todos sus componentes:
 - ▶ (1) Tipos Datos Derivados (DDT)
 - ▶ (2) Bloques Función Derivados (DFB)
 - ▶ (3) Subrutinas



● Funciones Exportación

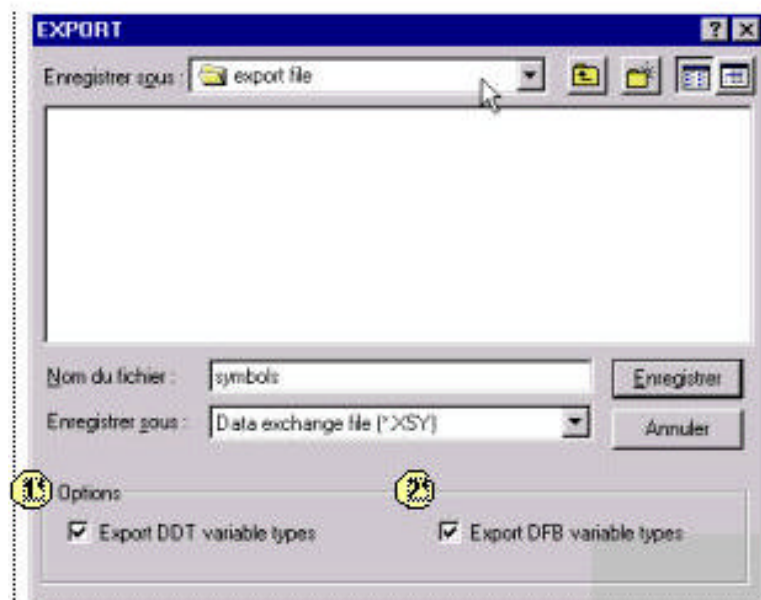
Unity Pro ofrece muchas posibilidad de exportar e importar archivos. El formato de archivos utilizado para las funciones importar/exportar es el estándar XML, el cual permite la creación o uso de de archivos de exportación de modo simple a partir de herramientas de terceros.

Los archivos exportación de los módulos comprenden:

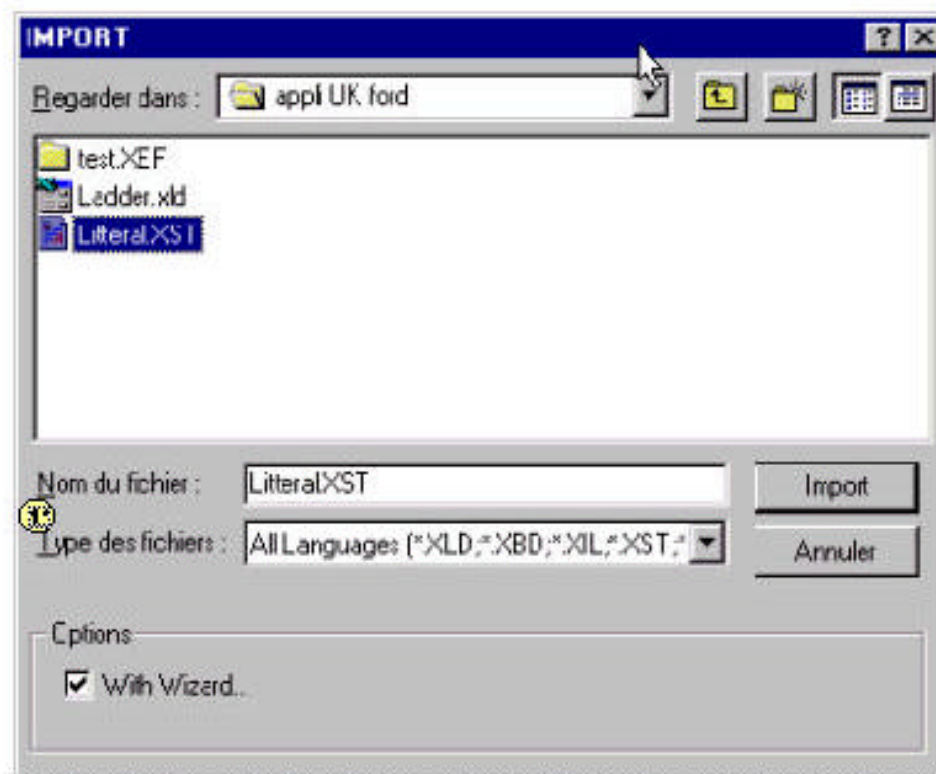
- Aplicación completa , incluyendo configuración.
- Secciones en todos los lenguajes (LD, ST, IL, FBD, SFC)
- Bloques Función Derivados (DFB)
- Tipos Datos Derivados (DDT)
- Variables
- Pantallas Operador
- Configuración en formato FNES

● Funciones Exportación (cont)

- Variables pueden exportarse con todas las instancias de tipos DDT o DFB incluyéndose su mismo tipo con el fin de asegurar un reuso simple en un nuevo programa.
 - ▶ (1) Tipos Datos Derivados (DDT)
 - ▶ (2) Bloques Función Derivados (DFB)



● Funciones Importación



**Ver
Comentarios**

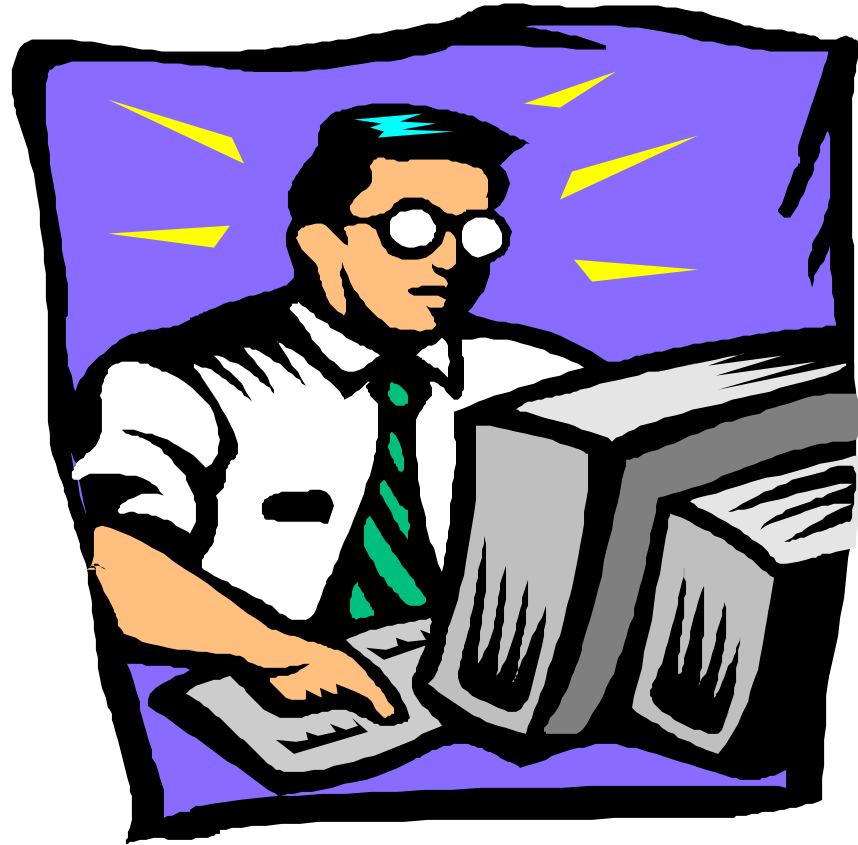
● **Funciones Importación**

La importación funciona para toda la aplicación o parte de ella.

Se proporcionan dos modos de importación con Unity Pro:

- Importación directa, importación del módulo tal y como fué exportado.
- Importación con el asistente (1), para modificar el nombramiento de las Variables, Secciones o Módulo Funcional, así como para cambiar el mapeo direcciones.

Ejercicio 4



Notas

Notas