StateWORKS

Specifying a State Machine - Tutorial

Introduction

- The tutorial teaches you how to prepare the specification of a virtual finite state machine using StateWORKS Studio.
- We assume that you know about the Vfsm concept [1].

- When it starts StateWORKS Studio opens the last project or nothing if started the first time.
- A specification of a new state machine is independent of the project opened : you may leave it or close it.

- The specification uses RTDB (real time data base) objects, such as digital input (DI), digital output (DO), timer (TI), etc.
- The VFSM and UNIT object type are „empty“ and we may use them to specify specific variants of those types.

Introduction

- The tutorial uses the project ***Pumps*** to illustrate the design steps from the book [1] where you find detailed require-ments and the analysis of the control task.
- For the purpose of that tutorial a partial specifications are provided which may be loaded to accelerate the training (observe corresponding notes).

Terminology

- **Always (table)**
  A table used for specification of combinational systems or Input actions valid for all states
- **Entry action**
  An Output name describing an action performed by entering a state
- **Exit action (written also as eXit action)**
  An Output name describing an action performed by exiting a state
- **Id name**
  A name of an object
- **I/O Object Dictionary**
  A list of all defined objects
- **I/O Object Id**
  see: Id name
- **Init (flag)**
  A flag: if marked instructs the execution system (RTDB) to initialize the virtual input to that value
- **Init (state)**
  A default state which cannot be deleted but can be renamed
- **Input**
  see: Input Name
- **Input (tab)**
  see: Input Name Dictionary
- **Input action**
  An Output name describing action performed if an Input action condition is due

Terminology

- **Input action condition**
  A condition defined using Input names linked by AND and OR operators
- **Input action expression**
  Input action condition and Input action
- **Input action priority**
  The sequence of Input action expressions in the ST table; used for documentation purpose
- **Input Name**
  A name of a control condition (defined on an Input Value)
- **Input Name Dictionary**
  A list of all defined Input Names
- **Input Value**
  Object input value
- **MyCmd**
  A default Input Name of a type CMD which cannot be deleted but can be renamed
- **Next State priority**
  The sequence of state transitions in the ST table; determines the execution sequence
- **Operators: AND (&), OR (|)**
  Boolean operators
- **Output**
  see: Output Name
- **Output (tab)**
  see: Output Name Dictionary
- **Output Name**
  A name describing an action (defined on an Output Value)

Terminology

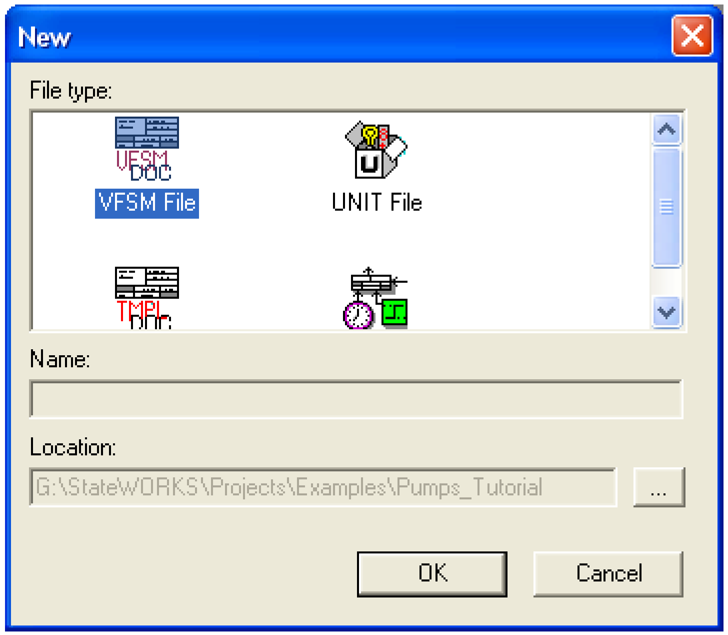
- **Output Name Dictionary**
  A list of all defined Output Names
- **Output Value**
  An Object output value
- **Prefix**
  A VFSM specific prefix used in h-files generated for each VFSM
- **ST diagram**
  A state transition diagram used for graphic presentation of a state machine behavior
- **ST table**
  A state transition table used for detailed specification of a state.
- **State**
  see: State Name (drawn as a circle on the ST diagram)
- **State Name**
  A state name
- **State Name Dictionary**
  A list of all defined State Names
- **Transition**
  A transition between two states (drawn as an arrow on the ST diagram)
- **Transition condition**
  A condition defined using Input names linked by AND and OR operators
- **Transition expression**
  Next state and Transition condition

Creating a virtual finite state machine (VFSM)

- A virtual finite state machine (VFSM) is an RTDB object type.
- Creating a new VFSM means a definition of a new specific VFSM type.
- The new VFSM object type gets a name, for instance ***Pressure*** and can be used in the project exactly as any other RTDB objects: once or in several instances.

Creating a virtual finite state machine (VFSM)

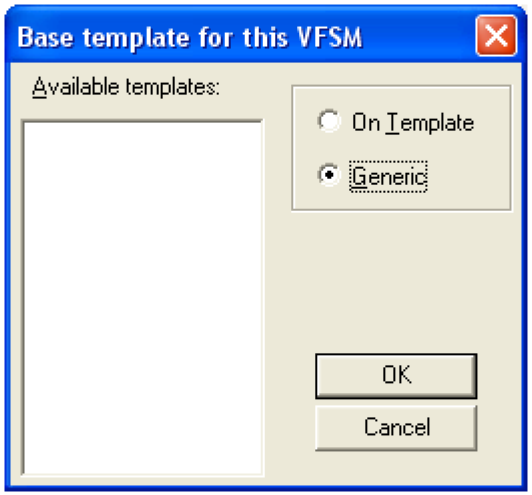
- Click on the **New** button on the toolbar or on the command **New** in the menu **File**.
- Select the **VFSM File** icon in the dialog windows.
- Leave this dialog window by clicking on the **OK** button.



Location: G:\StateWORKS\Projects\Examples\Pumps_Tutorial

Creating virtual finite state machine (VFSM)

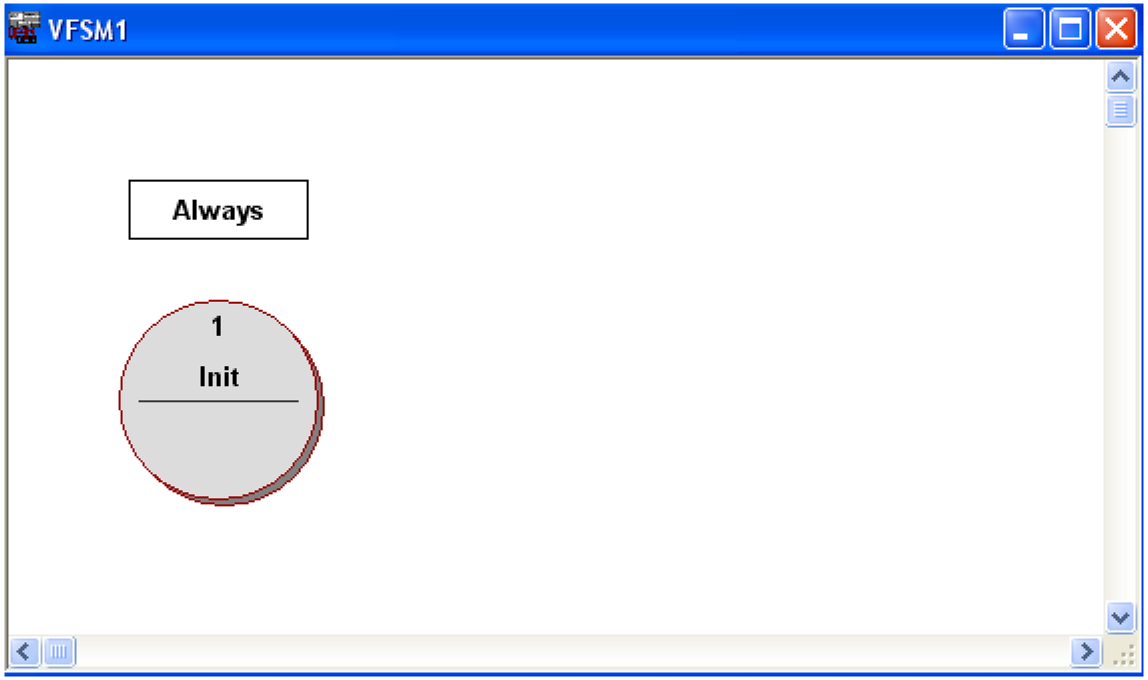
- Select the radio button **Generic**.
- Leave the dialog window by clicking on the **OK** button.



Generic [✓]

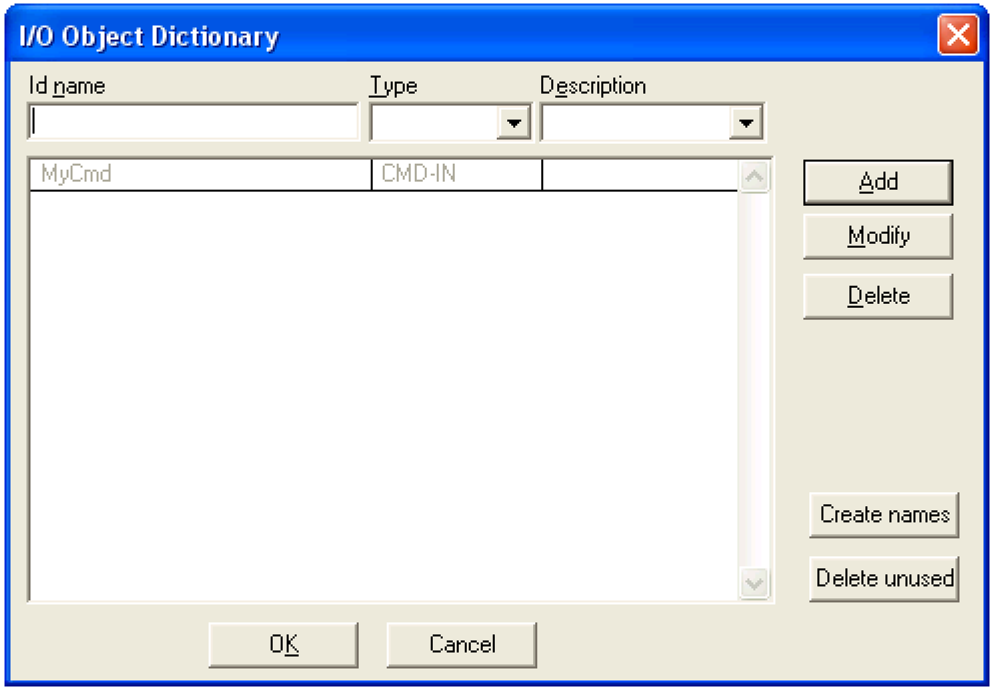
Creating virtual finite state machine (VFSM)

- A state transition (**ST) diagram** will be created, with a state **Init** and the table **Always**.
- The state machine is given a default name VFSM1; that name can be changed while saving the file.



Defining required objects

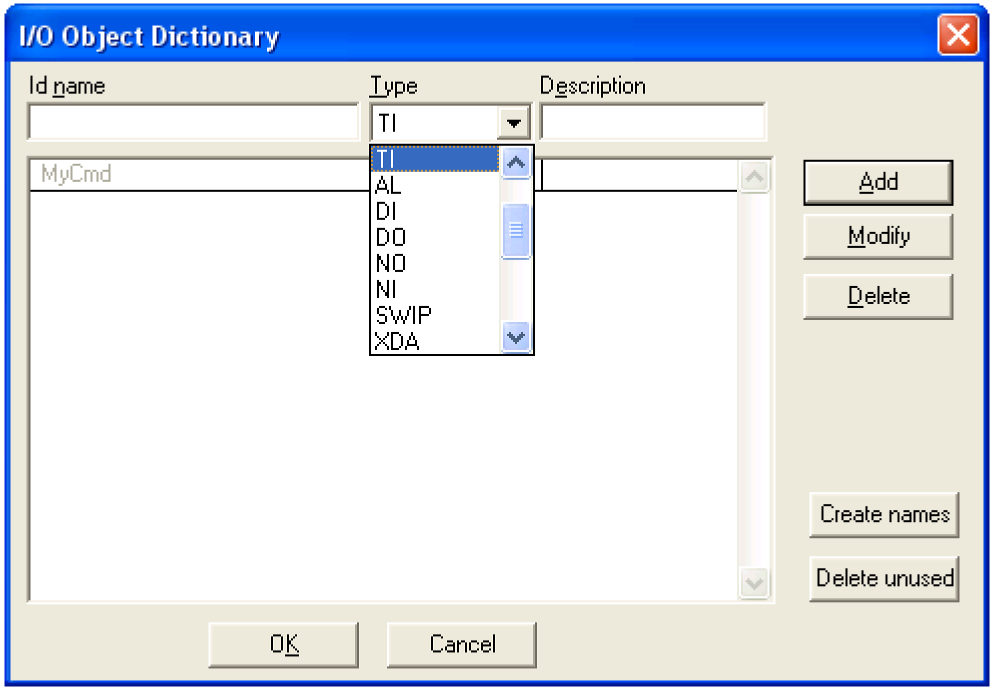
- Open the **I/O Object Dictionary** by clicking on the icon on the toolbar or on the command **I/O Object...** in the menu **Dictionary** or using the function key **F5**.
- The dialog window opens with a default object **MyCmd** of type **CMD-IN**. That object cannot be **Deleted**.



Id name	Type	Description
MyCmd	CMD-IN	

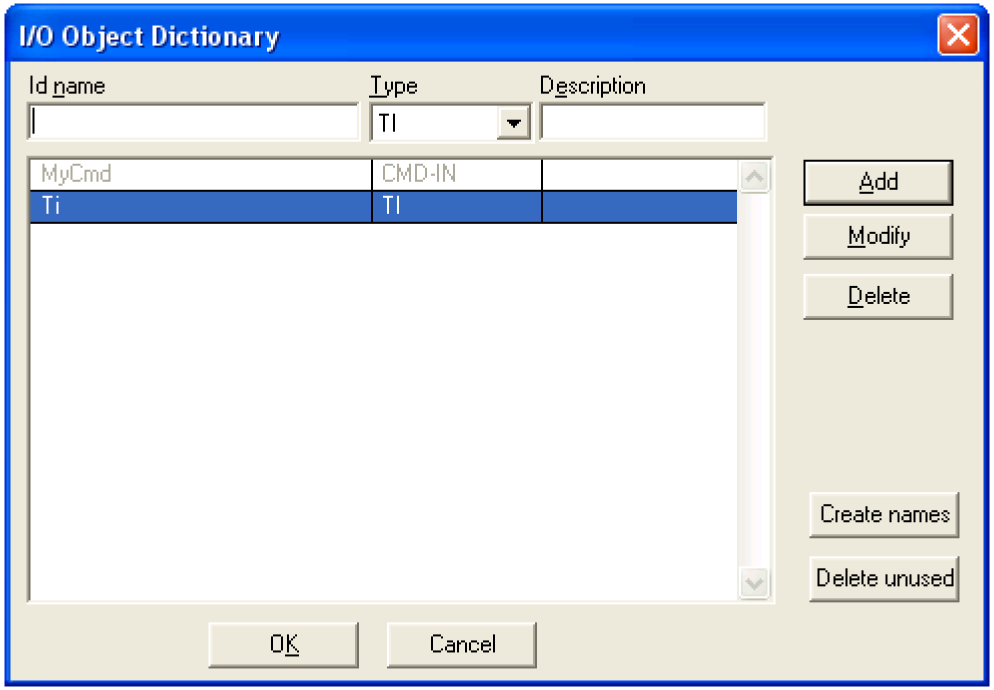
Defining required objects

- Select a **Type** of object, for instance ***TI*** (timer).



Defining required objects

- Define the object name in the **Id name** field, for instance ***Ti***.
- Add the name to the **I/O Object Dictionary** by clicking on the button **Add**.
- Note that any state machine has a default object ***MyCmd*** of a type CMD-IN.

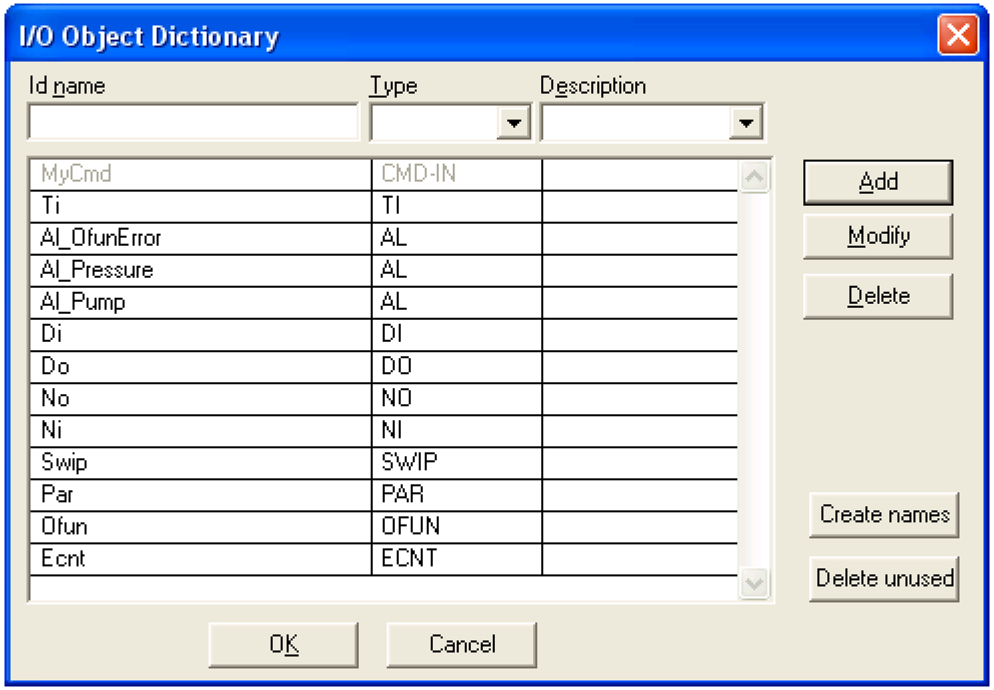

I/O Object Dictionary

Id name: (empty) · Type: TI · Description: (empty)

Id name	Type	Description
MyCmd	CMD-IN	
Ti	TI	

Defining required objects

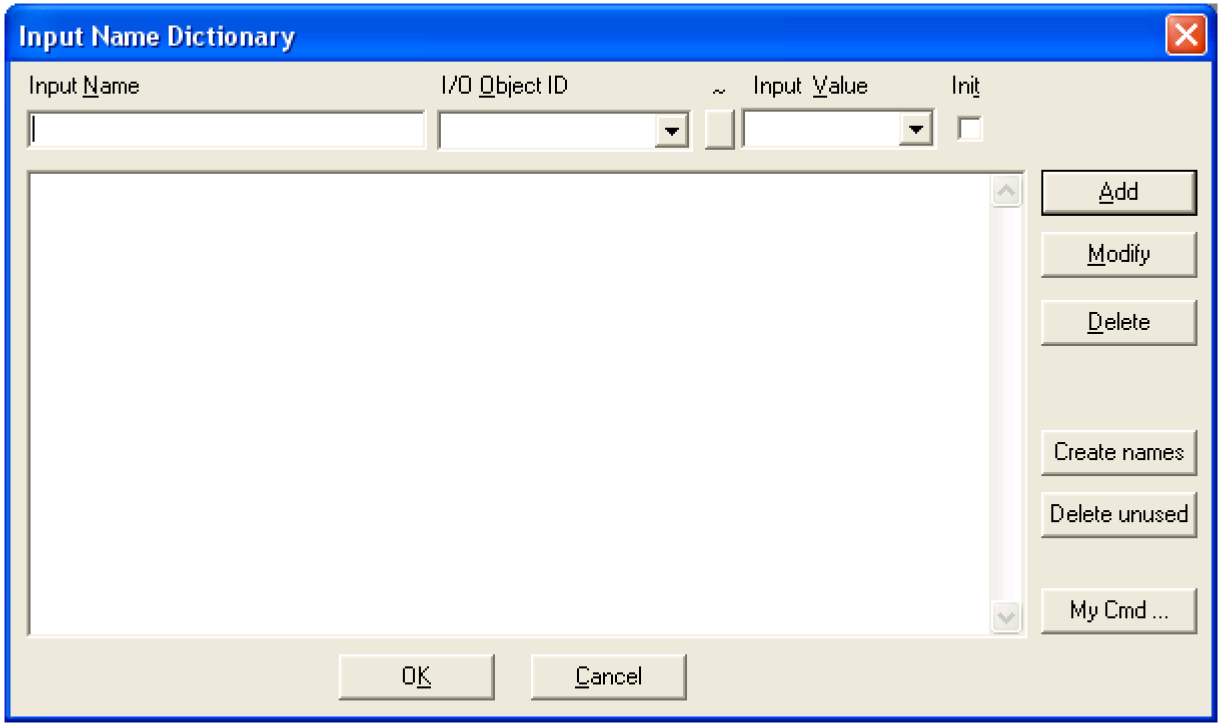
- If you add all required objects the **I/O Object Dictionary** may look for instance as below (open Pressure_IODictionary).
- The list may be changed at any time: the objects may be **Deleted**, **Added** and **Modified**.
- Leave the dialog window with **OK**.



MyCmd	CMD-IN	
Ti	TI	
Al_OfunError	AL	
Al_Pressure	AL	
Al_Pump	AL	
Di	DI	
Do	DO	
No	NO	
Ni	NI	
Swip	SWIP	
Par	PAR	
Ofun	OFUN	
Ecnt	ECNT	

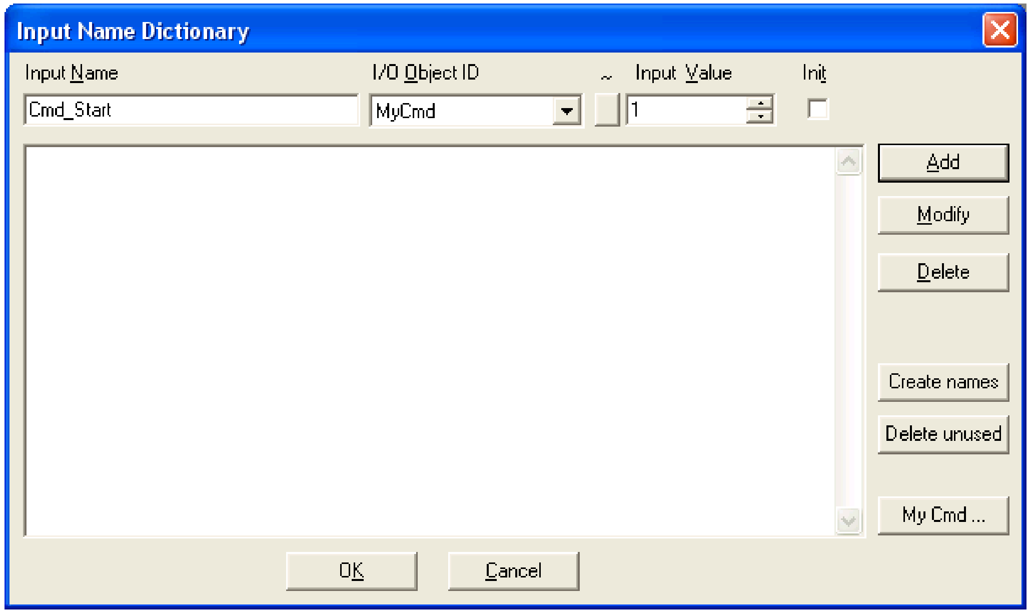
Defining Input Names

- Open the **Input Name Dictionary** by clicking on the icon on the toolbar or on the command **Input...** in the menu **Dictionary** or using the function key **F2**.



Defining Input Names

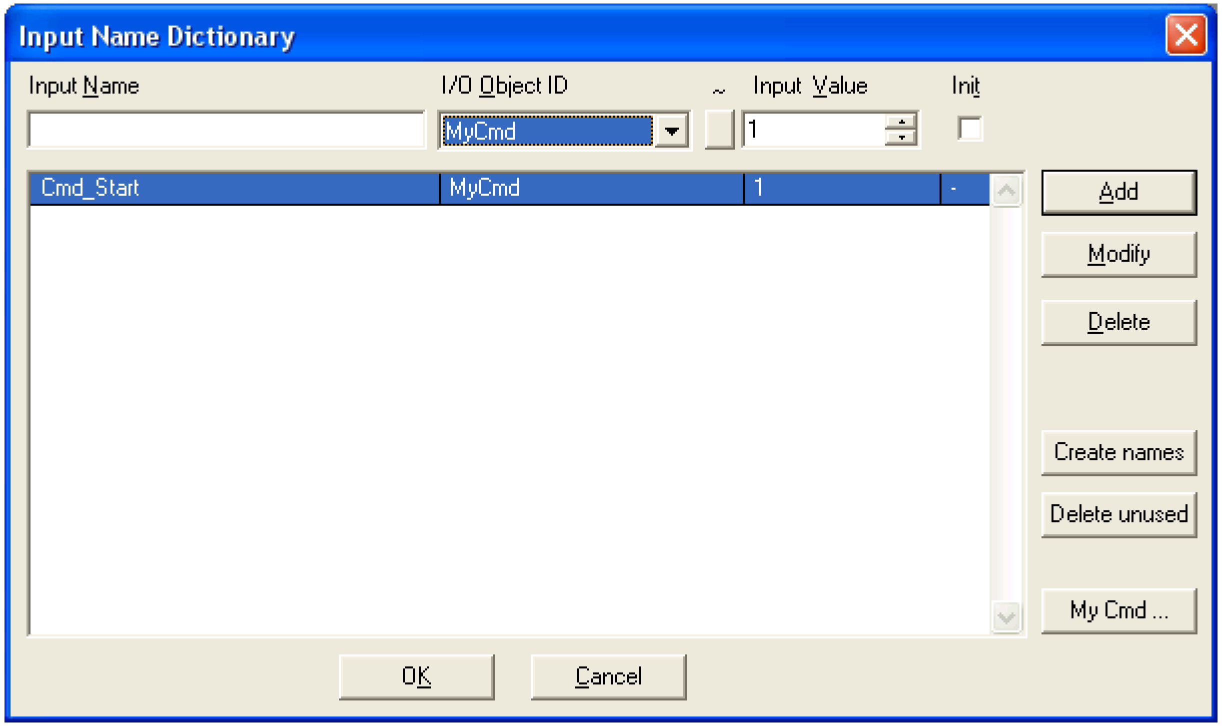
- Select the **I/O Object ID**, for instance ***MyCmd***.
- Select a command value in the **Input Value** field, for instance ***1***.
- Edit the **Input Name,** calling the value, for instance ***Cmd_Start***.
- Note that there is no automatic suggestion of a name for objects whose values are numbers.


Input Name Dictionary

Input Name	I/O Object ID	Input Value	Init
Cmd_Start	MyCmd	1	☐

Defining Input Names

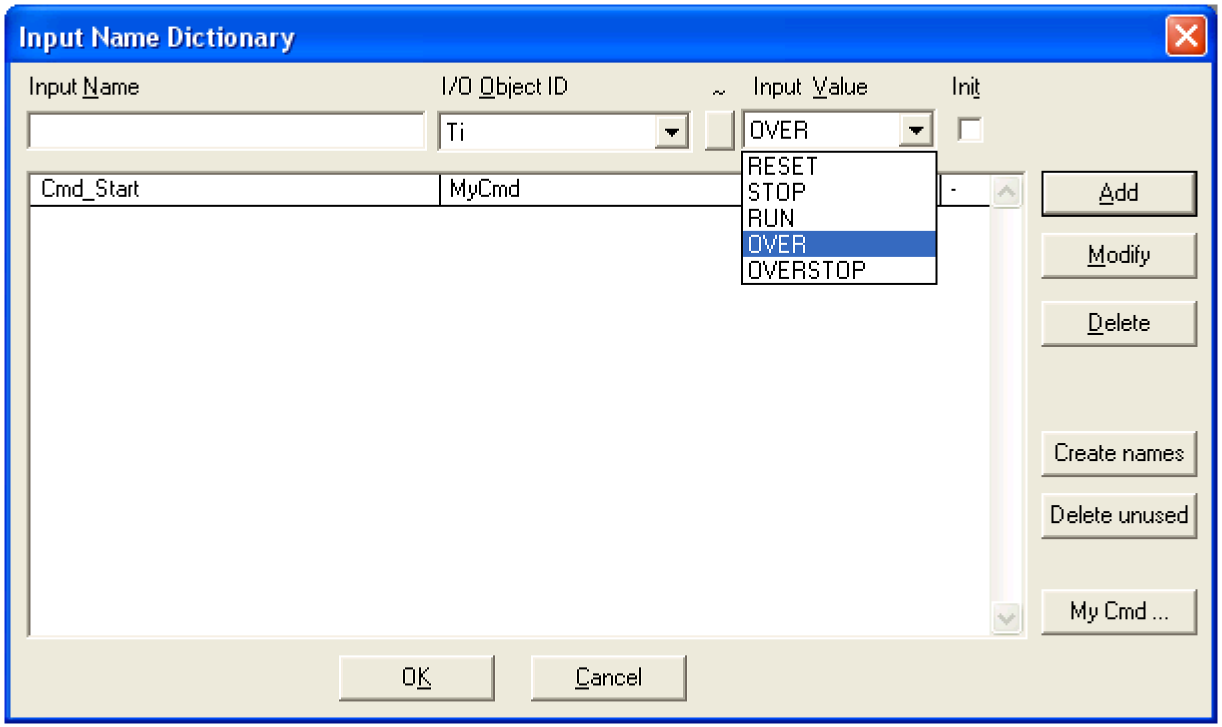
- Add the name ***Cmd_Start*** to the **Input Name Dictionary** by clicking on the button **Add**.


Input Name Dictionary

Input Name	I/O Object ID	Input Value	Init
	MyCmd	1	
Cmd_Start	MyCmd	1	-

Defining Input Names

- Select another **I/O Object ID**, for instance ***Ti***.
- Open the list of object **Input Values.**
- Select the required value, for instance ***OVER***.



Input Name Dictionary

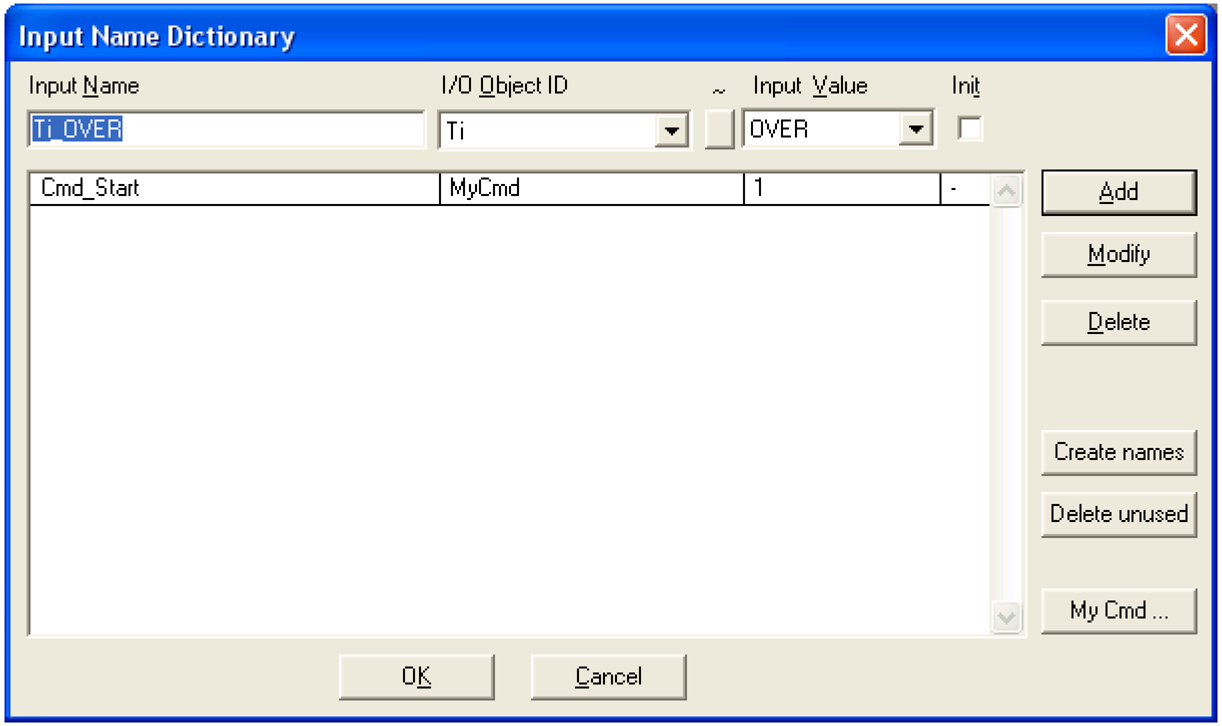
Input Name	I/O Object ID	~	Input Value	Init
	Ti		OVER	☐

Input Value list: RESET, STOP, RUN, OVER, OVERSTOP

Cmd_Start	MyCmd		-

Defining Input Names

- Define a name by clicking on the button **Add:** the name***Ti_OVER*** appears in the **Input Name** field.
- If you do not like the name edit it in the **Input Name** field.

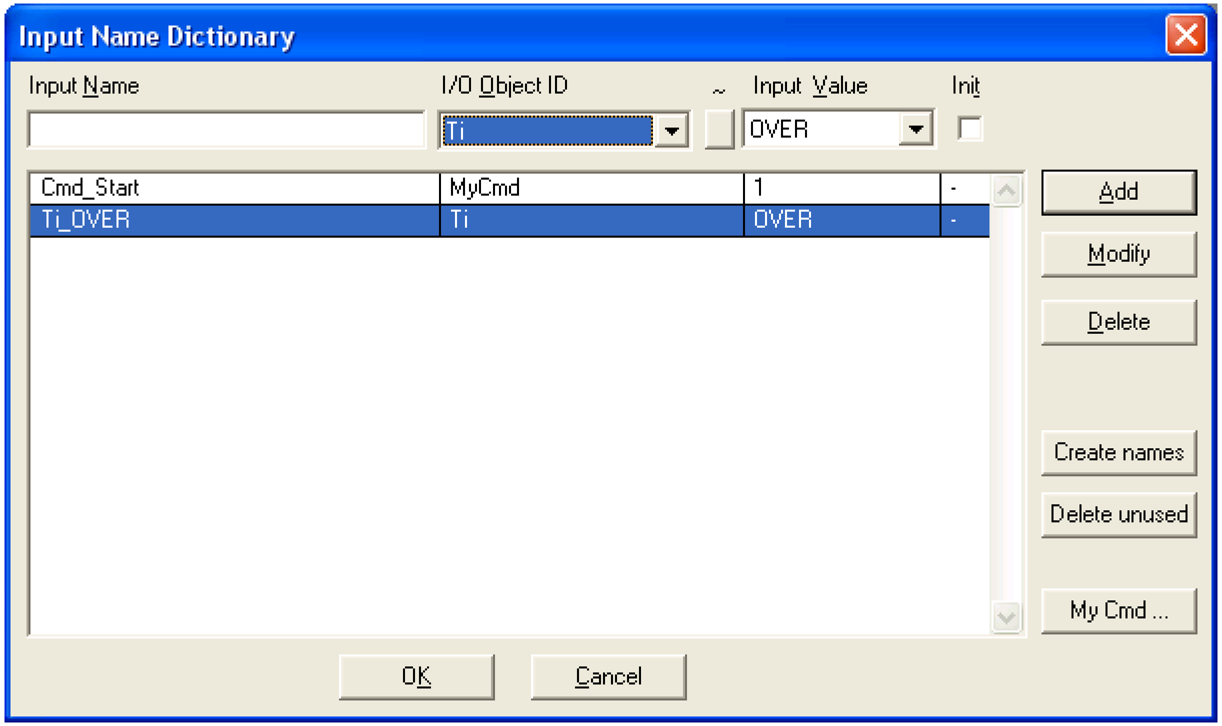

Input Name Dictionary

Input Name	I/O Object ID	~	Input Value	Init
Ti_OVER	Ti		OVER	☐

Cmd_Start	MyCmd	1	-

Defining Input Names

- Add the name ***Ti_OVER*** to the **Input Name Dictionary** by anew click on the **Add** button.


Input Name Dictionary

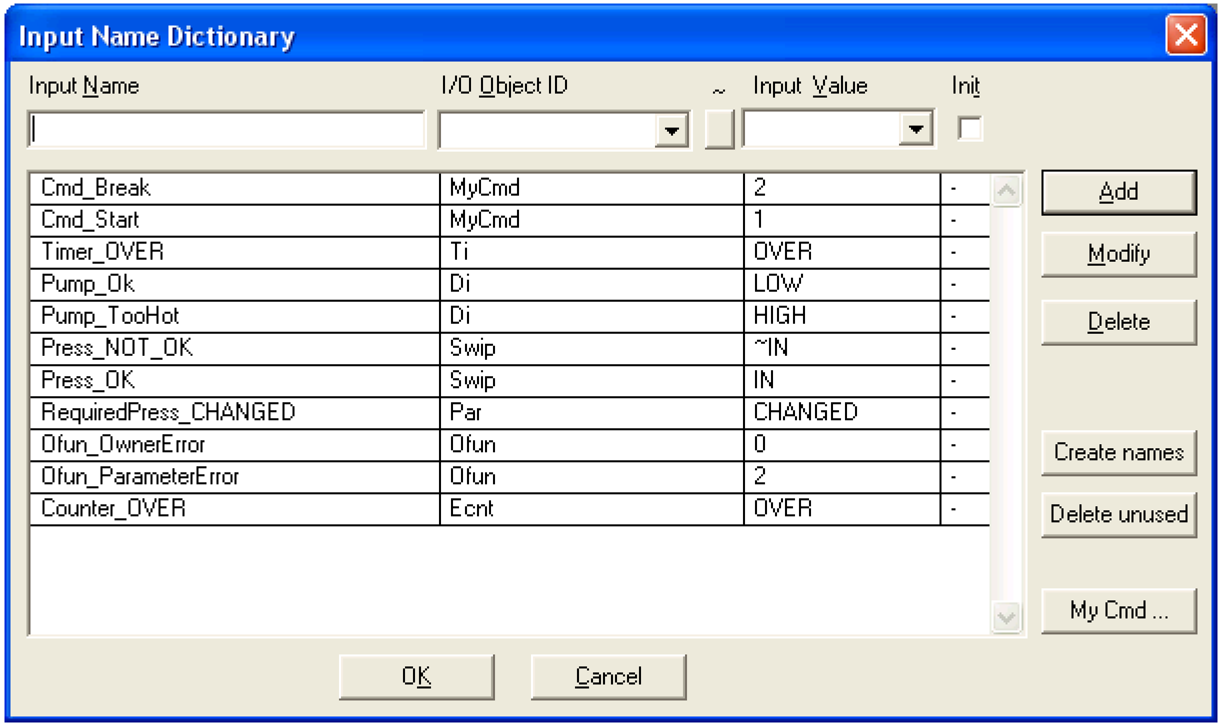
Input Name:
I/O Object ID: Ti
Input Value: OVER
Init: ☐

Cmd_Start	MyCmd	1	-
Ti_OVER	Ti	OVER	-

Defining Input Names

- If you define all required **Input Names** the **Input Name Dictionary** may look for instance as below

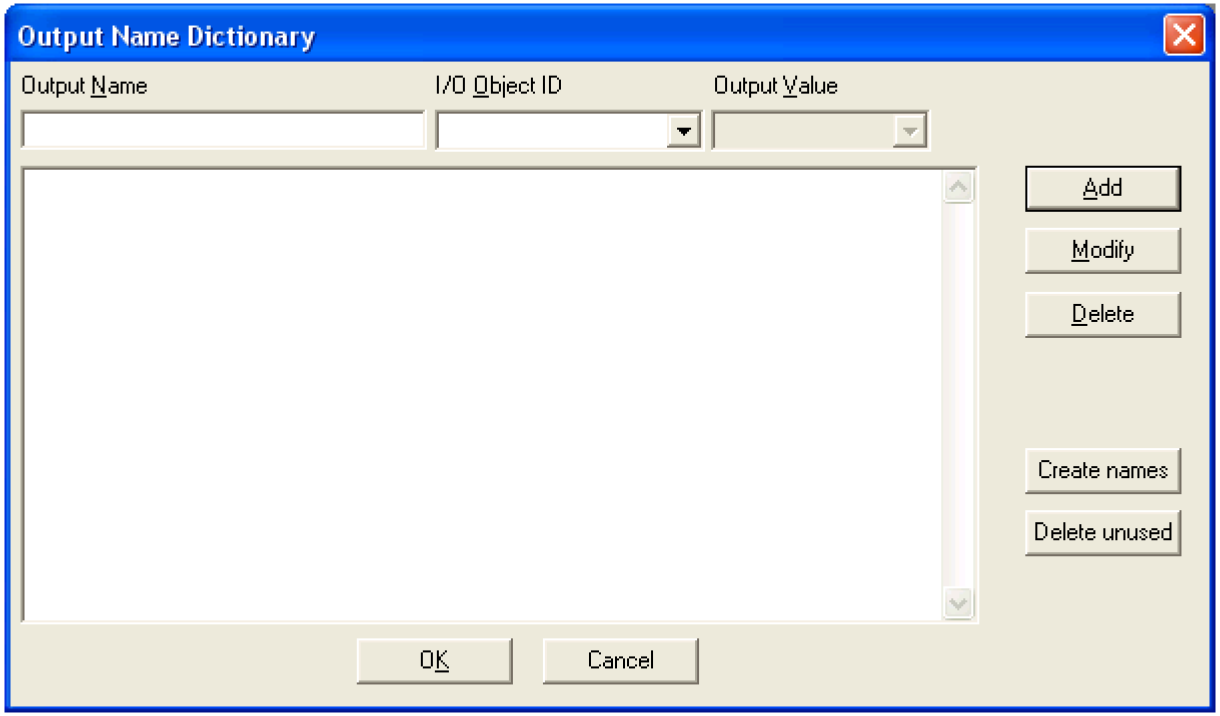
  Note: Open Pressure_InputNameDictionary from the Pumps_Tutorial folder.
- The Dictionary may be changed at any time: selected names may be **Deleted**, **Added** and **Modified**.
- Leave the dialog window with **OK**.



Input Name	I/O Object ID	Input Value	Init
Cmd_Break	MyCmd	2	-
Cmd_Start	MyCmd	1	-
Timer_OVER	Ti	OVER	-
Pump_Ok	Di	LOW	-
Pump_TooHot	Di	HIGH	-
Press_NOT_OK	Swip	~IN	-
Press_OK	Swip	IN	-
RequiredPress_CHANGED	Par	CHANGED	-
Ofun_OwnerError	Ofun	0	-
Ofun_ParameterError	Ofun	2	-
Counter_OVER	Ecnt	OVER	-

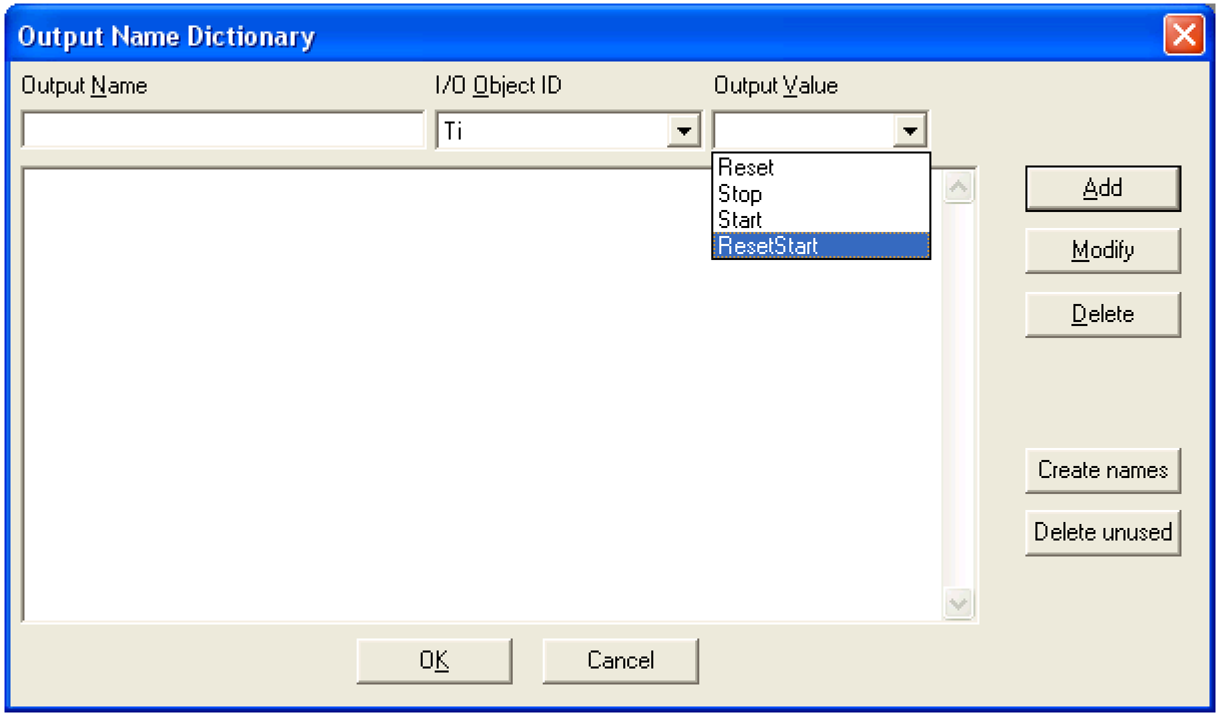
Defining Output Names

- Open the **Output Name Dictionary** by clicking on the icon on the toolbar or on the command **Output...** in the menu **Dictionary** or using the function key **F3**.



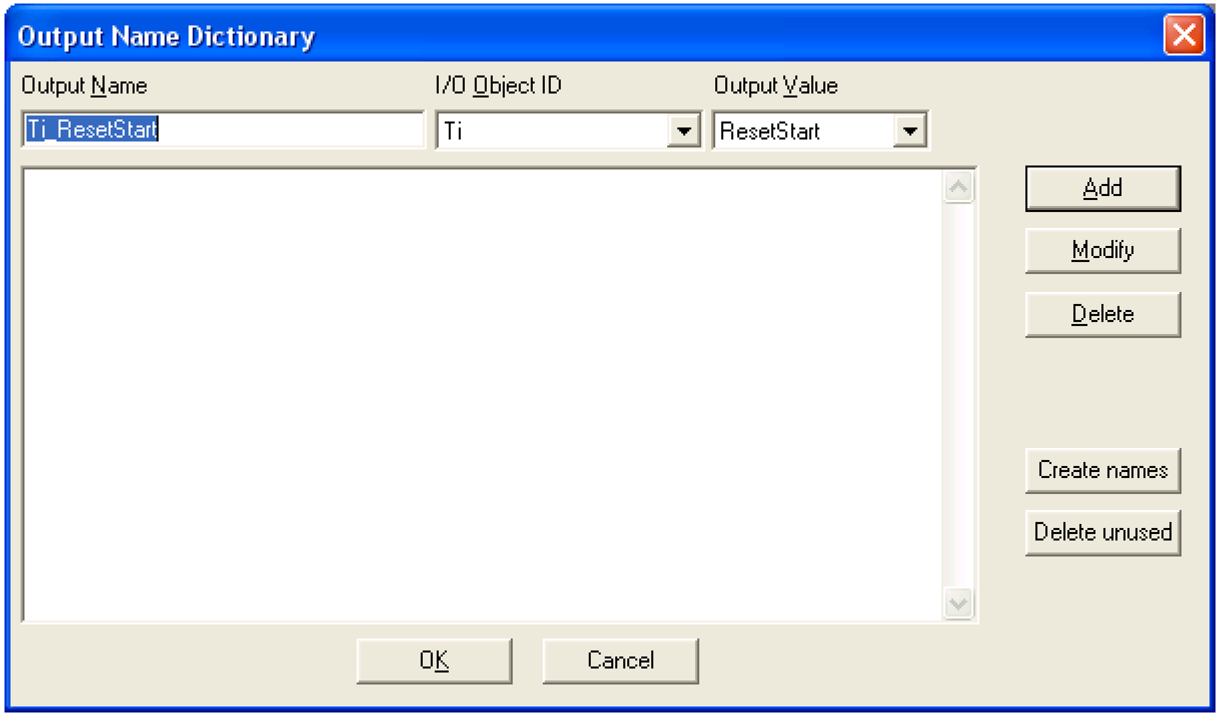
Defining Output Names

- Select an **I/O Object ID**, for instance ***Ti***.
- Open the list of object **Output Values.**
- Select required value, for instance ***ResetStart***.



Defining Output Names

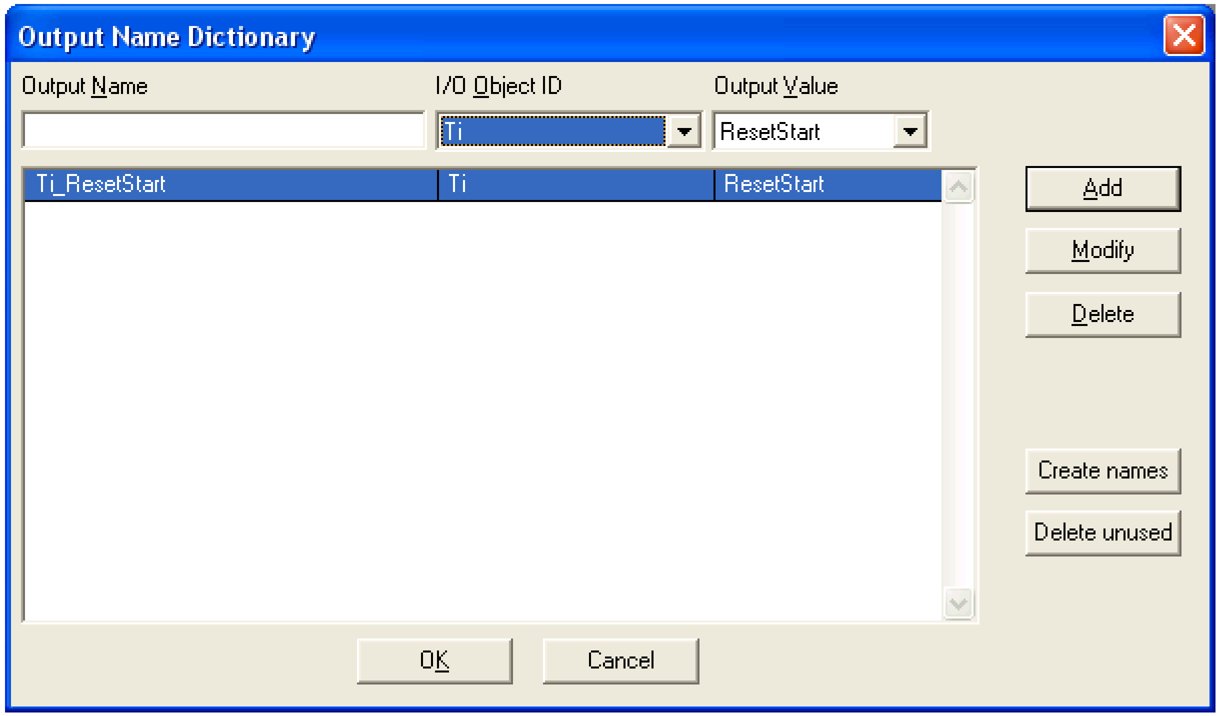
- Define a name by clicking on the button **Add:** the name***Ti_ResetStart*** appears in the **Output Name** field.
- If you do not like the name edit it in the **Output Name** field.



Output Name: Ti_ResetStart
I/O Object ID: Ti
Output Value: ResetStart

Defining Output Names

- Add the name ***Ti_ResetStart*** to the **Output Name Dictionary** by a new click on the button **Add**.


Output Name Dictionary

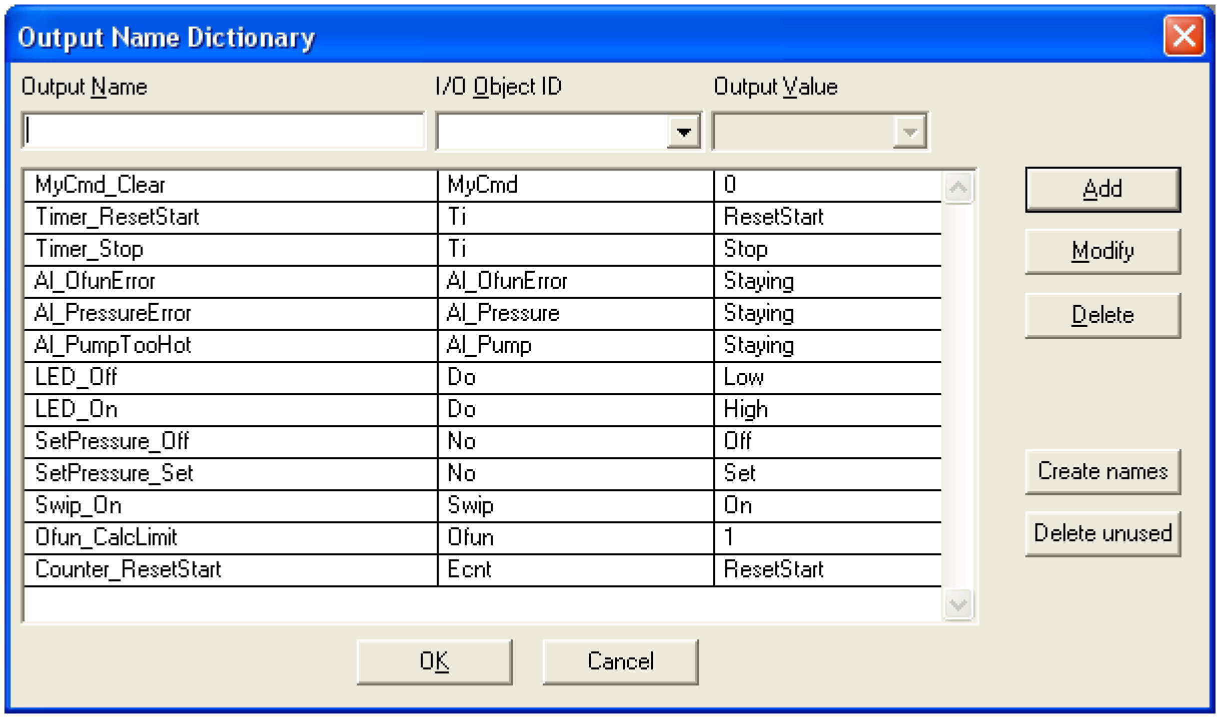
Output Name	I/O Object ID	Output Value
	Ti	ResetStart

Ti_ResetStart	Ti	ResetStart

Defining Output Names

- If you define all required **Output Names** the **Output Name Dictionary** may look for instance as below

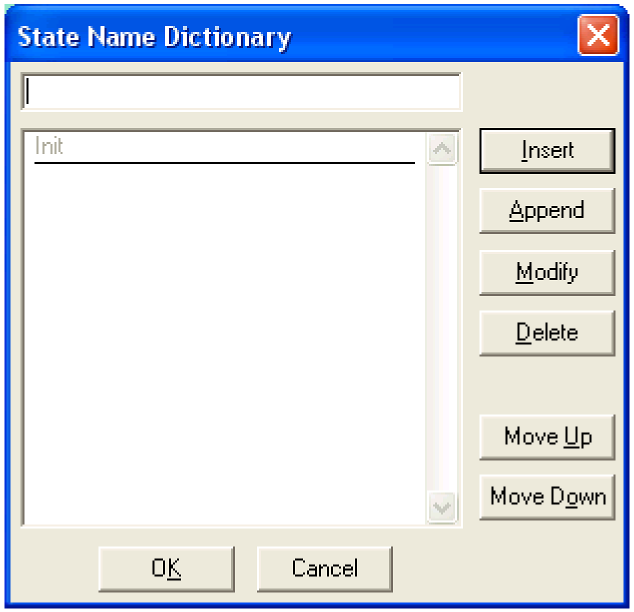
  Note: Open Pressure_OutputNameDictionary from the Pumps_Tutorial folder**.**
- The Dictionary may be changed at any time: selected names may be **Deleted**, **Added** and **Modified**.
- Leave the dialog window with **OK**.



Output Name	I/O Object ID	Output Value
MyCmd_Clear	MyCmd	0
Timer_ResetStart	Ti	ResetStart
Timer_Stop	Ti	Stop
Al_OfunError	Al_OfunError	Staying
Al_PressureError	Al_Pressure	Staying
Al_PumpTooHot	Al_Pump	Staying
LED_Off	Do	Low
LED_On	Do	High
SetPressure_Off	No	Off
SetPressure_Set	No	Set
Swip_On	Swip	On
Ofun_CalcLimit	Ofun	1
Counter_ResetStart	Ecnt	ResetStart

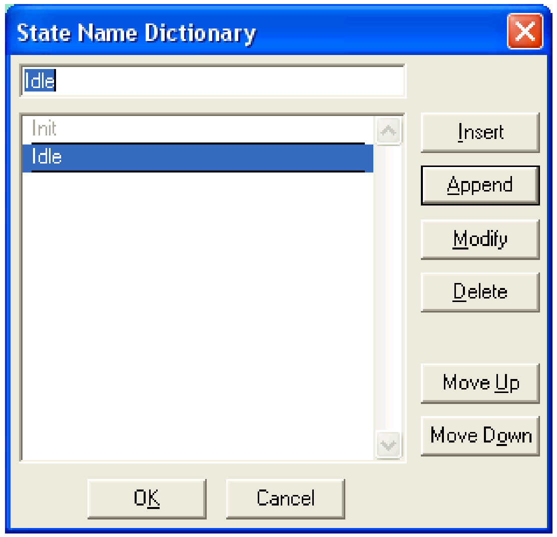
Defining State Names

- Open the **State Name Dictionary** by clicking on the ST diagram.
- The dialog window opens with a default state **Init**. That state cannot be **Deleted** but can be renamed: select it, edit and **Modify**.



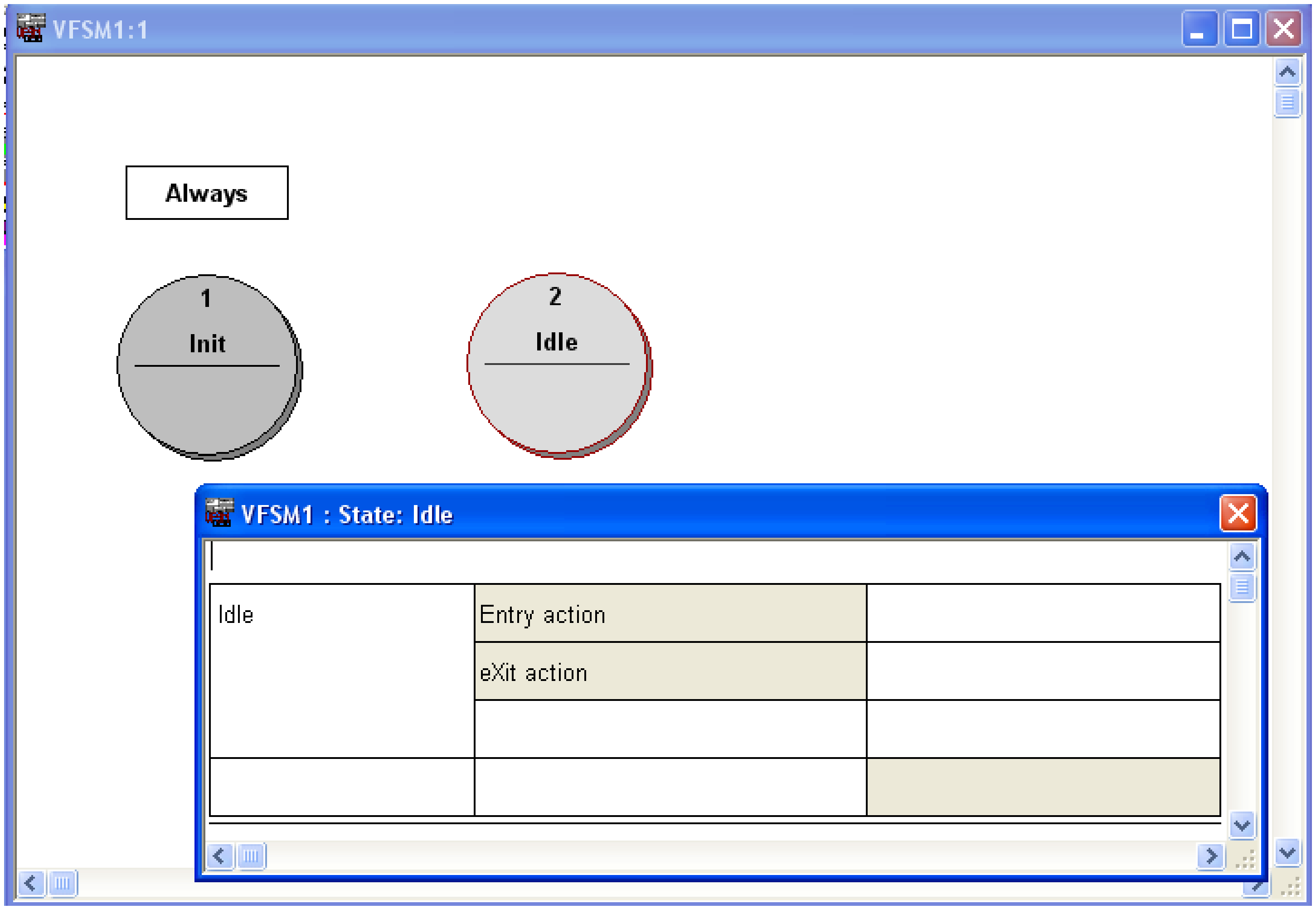
Defining State Names

- Edit a new state name for instance ***Idle***.
- Add the state name to the **State Name Dictionary** by clicking on the button **Append** or on **Insert**.
- The state names can be **Modified** and **Deleted**.
- The sequence of names in the list may be changed using Buttons **Move Up** and **Move Down**. The sequence has only cosmetic relevance, for instance for documentation purpose.
- Leave the dialog window by clicking on the button **Ok**.



Defining State Names

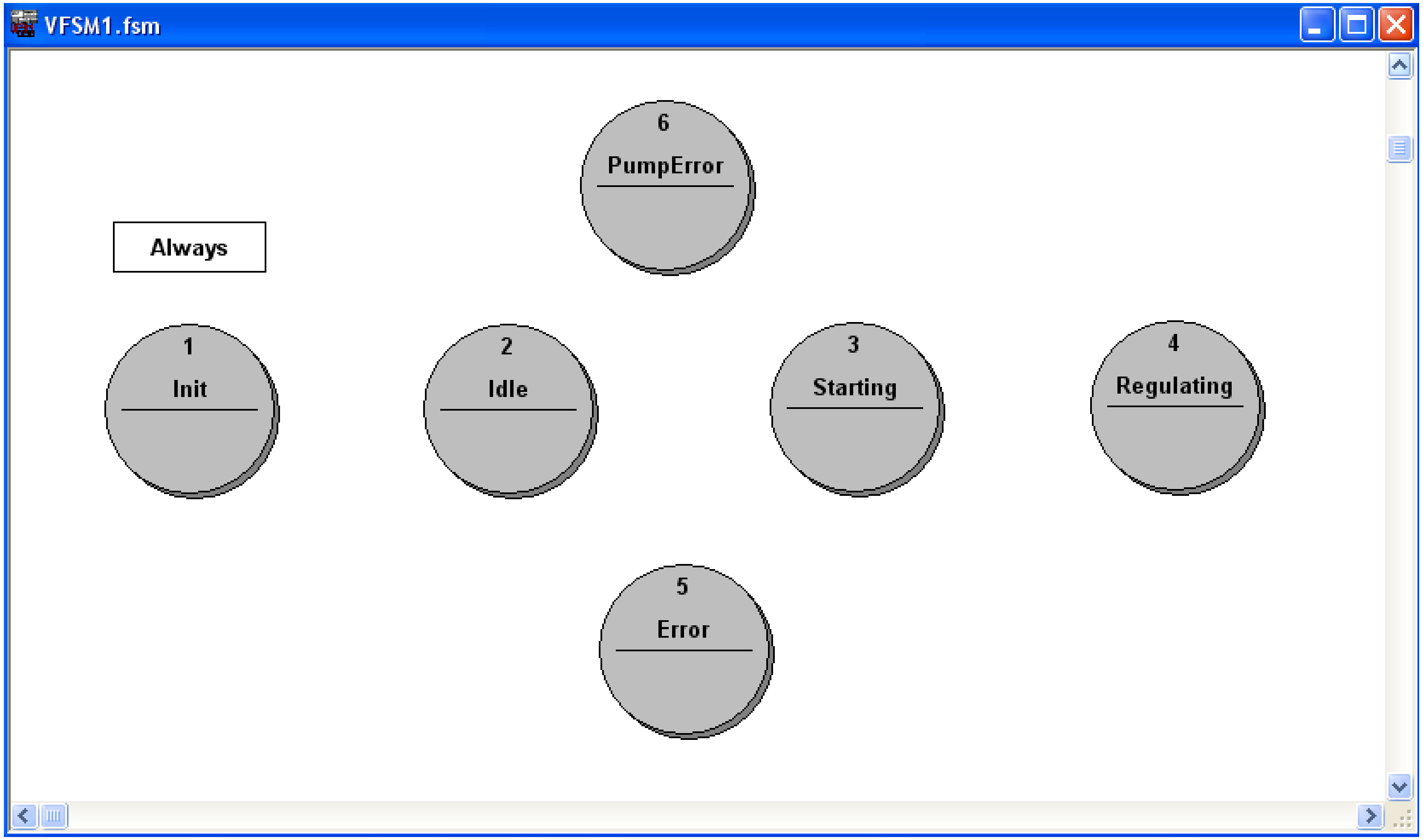
- The state ***Idle*** will appear on the **ST diagram** and the **ST table** of the state ***Idle*** opens.



Idle	Entry action	
	eXit action	

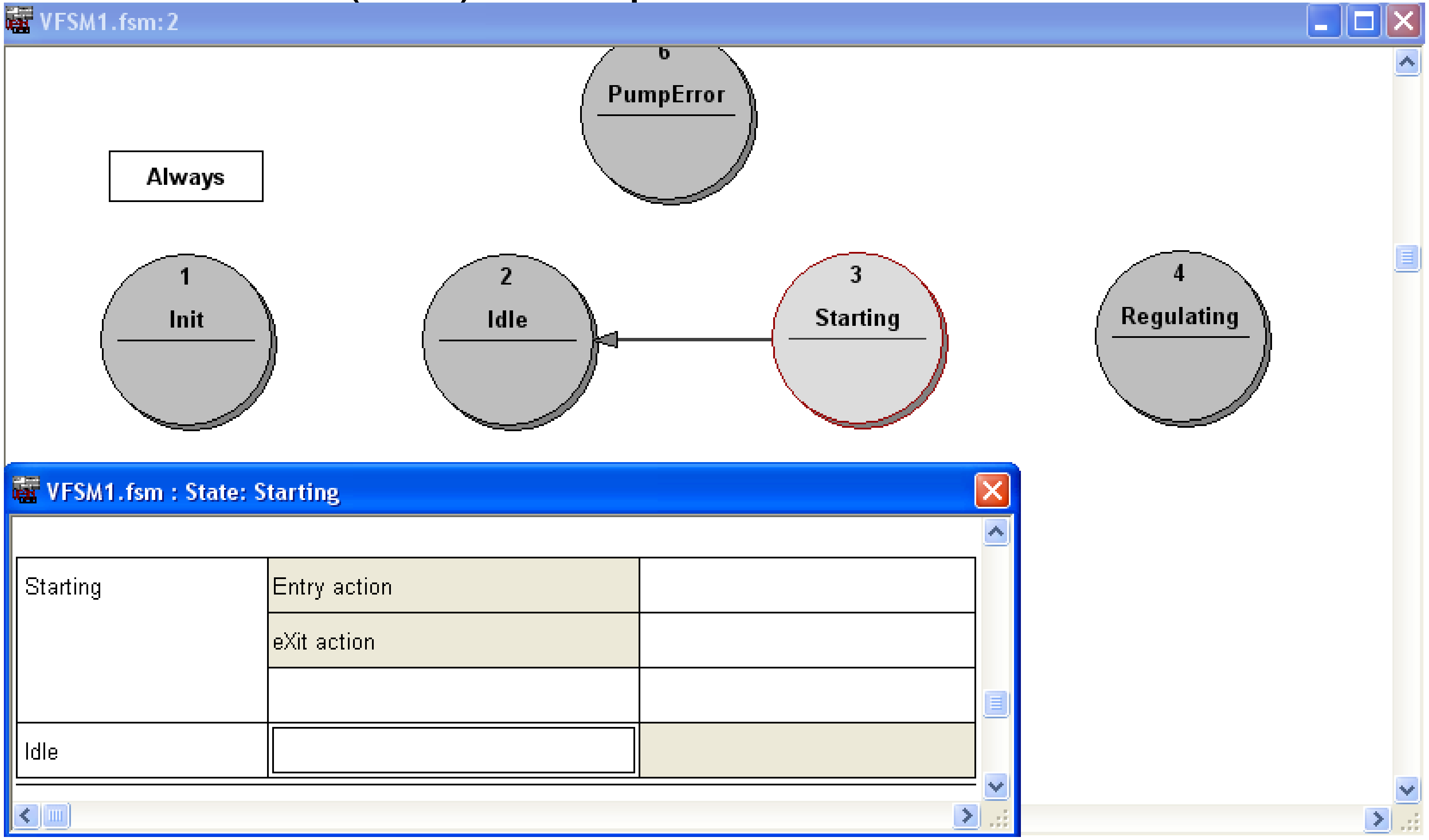
Defining State Names

- Repeating that procedure you may create a few states as shown below.



Specifying Transitions

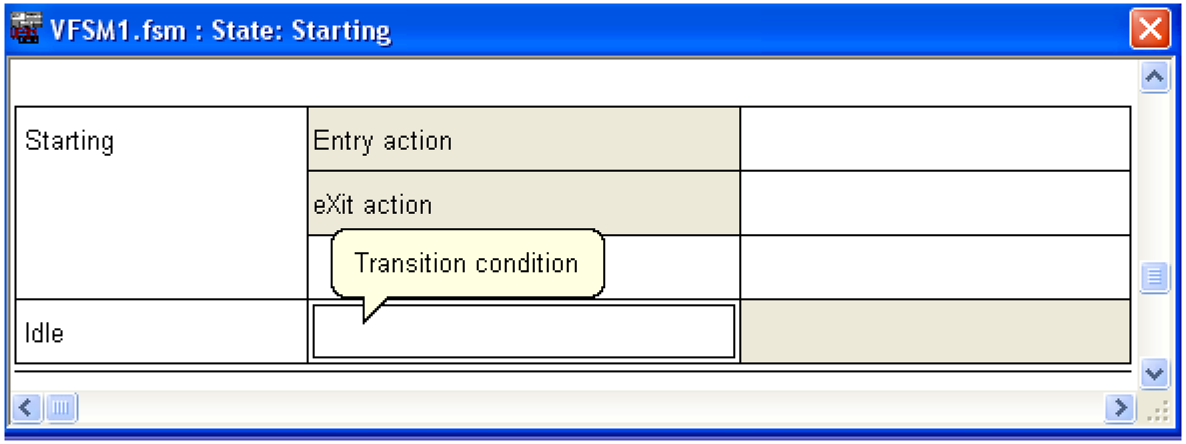
- Position the cursor over a state (for instance ***Starting***).
- Pushing the right mouse button, draw a transition arrow to another state (***Idle***) and open the **ST table**.



Starting	Entry action	
	eXit action	
Idle		

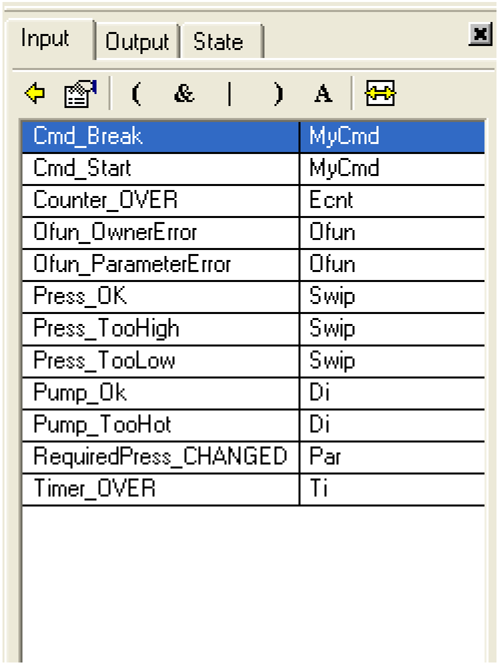
Specifying Transitions

- The content of **Input**, **Output**, and **State Dictionaries** is available as overlapped tabs; the position of the cursor in the **ST table** activates a relevant Dictionary.
- As the cursor is in the transition condition field the **Input Names Dictionaries** will be active.



VFSM1.fsm : State: Starting

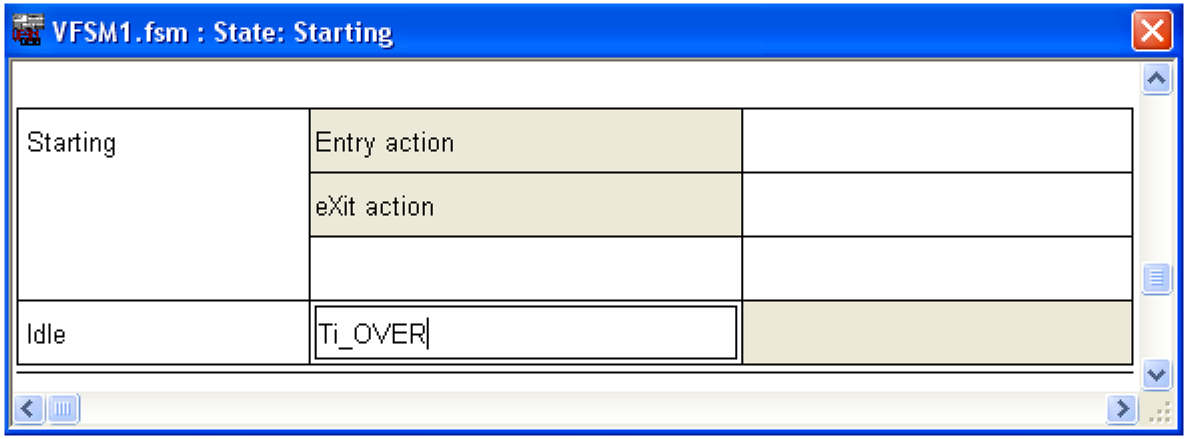
Starting	Entry action	
	eXit action	
	Transition condition	
Idle		



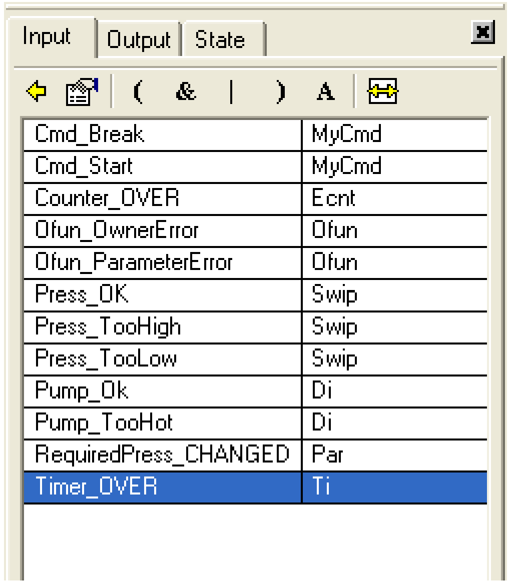
Cmd_Break	MyCmd
Cmd_Start	MyCmd
Counter_OVER	Ecnt
Ofun_OwnerError	Ofun
Ofun_ParameterError	Ofun
Press_OK	Swip
Press_TooHigh	Swip
Press_TooLow	Swip
Pump_Ok	Di
Pump_TooHot	Di
RequiredPress_CHANGED	Par
Timer_OVER	Ti

Specifying Transitions

- A click on the required name (***Ti_OVER***) in the **Input** tab copies the name into the field selected by the cursor (the transition field).



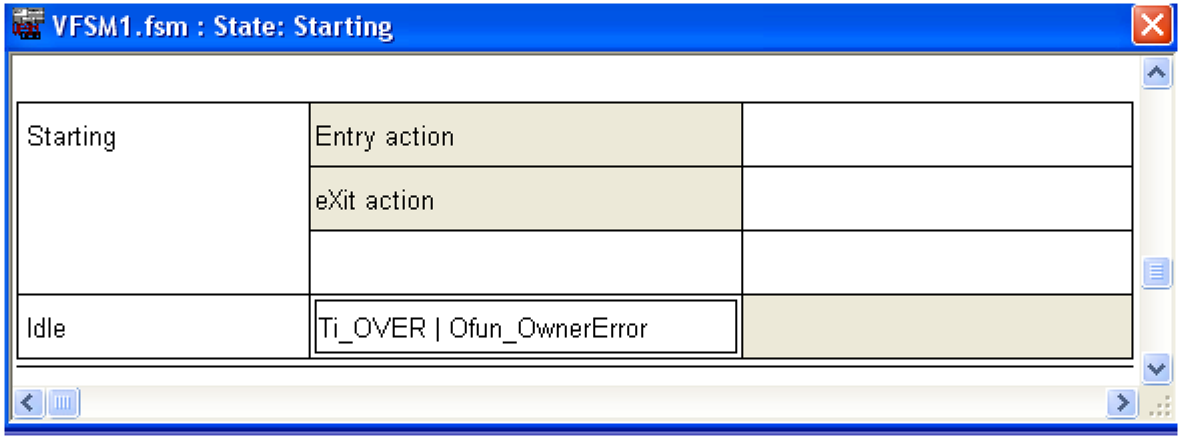
Starting	Entry action	
	eXit action	
Idle	Ti_OVER	



Cmd_Break	MyCmd
Cmd_Start	MyCmd
Counter_OVER	Ecnt
Ofun_OwnerError	Ofun
Ofun_ParameterError	Ofun
Press_OK	Swip
Press_TooHigh	Swip
Press_TooLow	Swip
Pump_Ok	Di
Pump_TooHot	Di
RequiredPress_CHANGED	Par
Timer_OVER	Ti

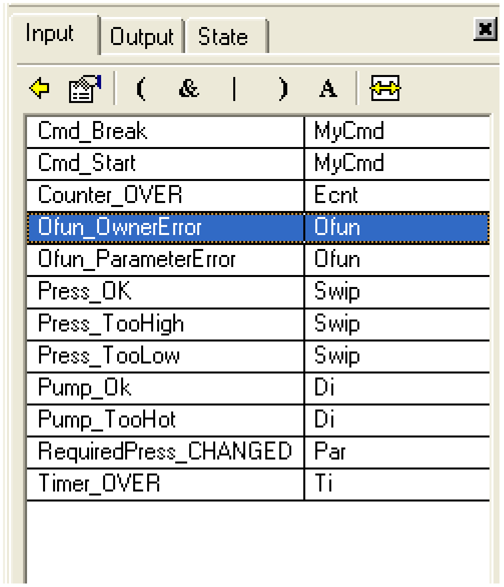
Specifying Transitions

- **AND** and **OR** operators as well as brackets available at the top of the **Input** tab may be copied into the transition field.
- Clicking first on the **OR** operator (|) and later on the name ***Ofun_OwnerError*** you may define a more complex trans-ition.



VFSM1.fsm : State: Starting

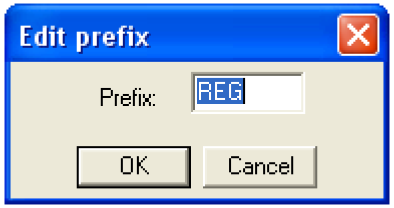
Starting	Entry action	
	eXit action	
Idle	Ti_OVER | Ofun_OwnerError	



Cmd_Break	MyCmd
Cmd_Start	MyCmd
Counter_OVER	Ecnt
Ofun_OwnerError	Ofun
Ofun_ParameterError	Ofun
Press_OK	Swip
Press_TooHigh	Swip
Press_TooLow	Swip
Pump_Ok	Di
Pump_TooHot	Di
RequiredPress_CHANGED	Par
Timer_OVER	Ti

Storing VFSM file

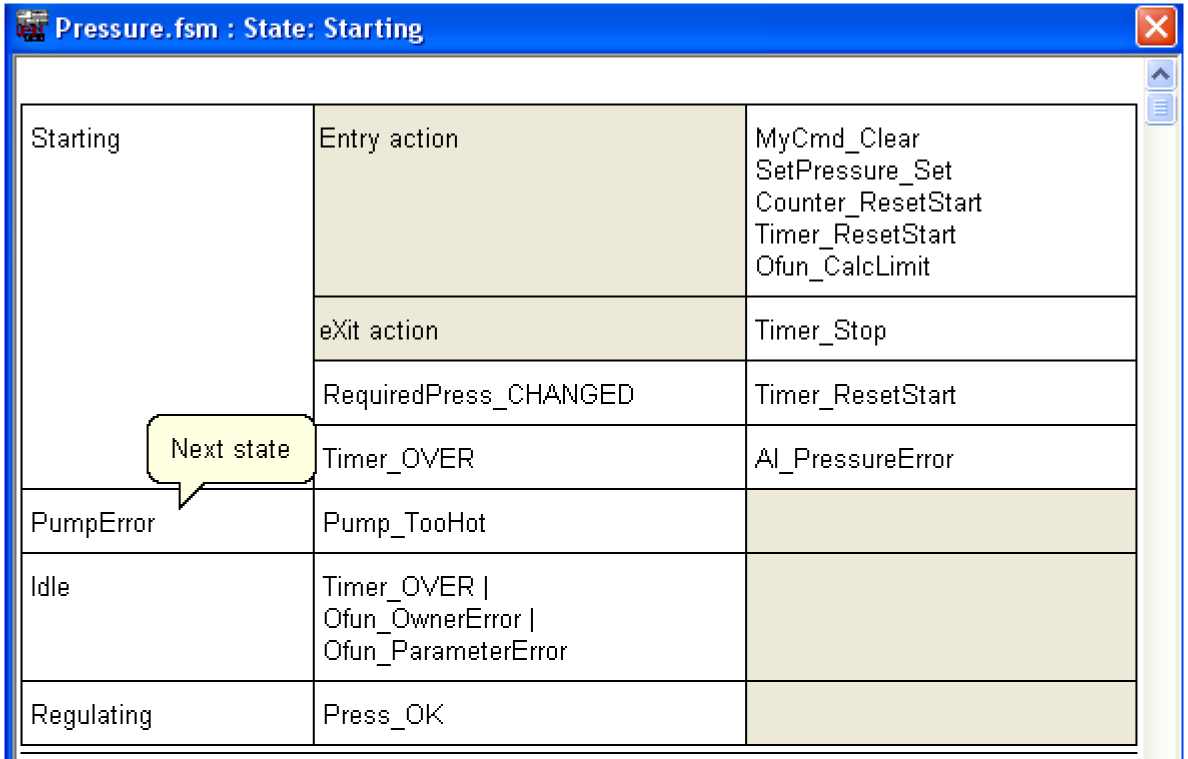
- You may store the VFSM file at any time by clicking on toolbar icon or on the **Save** command in the menu **File** or using the shortcut **CTRL/S**.
- During saving you define the file name and you are asked to define a VFSM **Prefix**. The 3-characters Prefix will be used in a h-file generated by performing the command **Project/Build**. You may accept the default Prefix which is built of the three first letters of the file name.
- Let's name the file ***Pressure*** and define the **Prefix** ***REG***. The **Prefix** may be changed at any time using the command **Options/Prefix...**


Edit prefix

Prefix: REG

Specifying Transitions

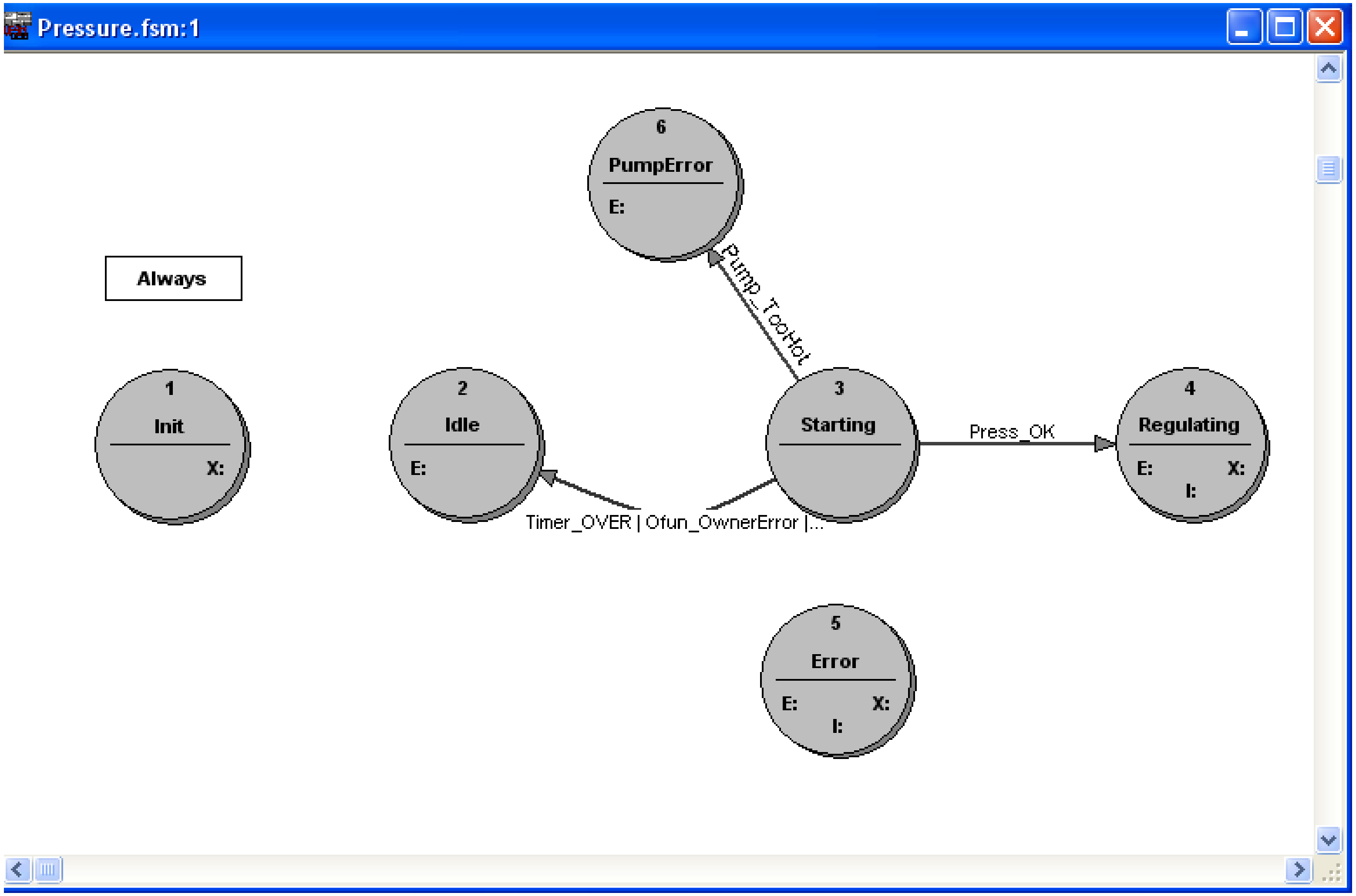
- Similarly you may specify transition conditions to other states receiving eventually the **ST table** of the state ***Starting*** shown below.
- The sequence of **Next states** in the **ST table** defines their execution priority. You may change the priority using toolbar arrows or commands **Move expression** .. in the menu **Edit**.



Starting	Entry action	MyCmd_Clear SetPressure_Set Counter_ResetStart Timer_ResetStart Ofun_CalcLimit
	eXit action	Timer_Stop
	RequiredPress_CHANGED	Timer_ResetStart
	Timer_OVER	AI_PressureError
PumpError	Pump_TooHot	
Idle	Timer_OVER | Ofun_OwnerError | Ofun_ParameterError	
Regulating	Press_OK	

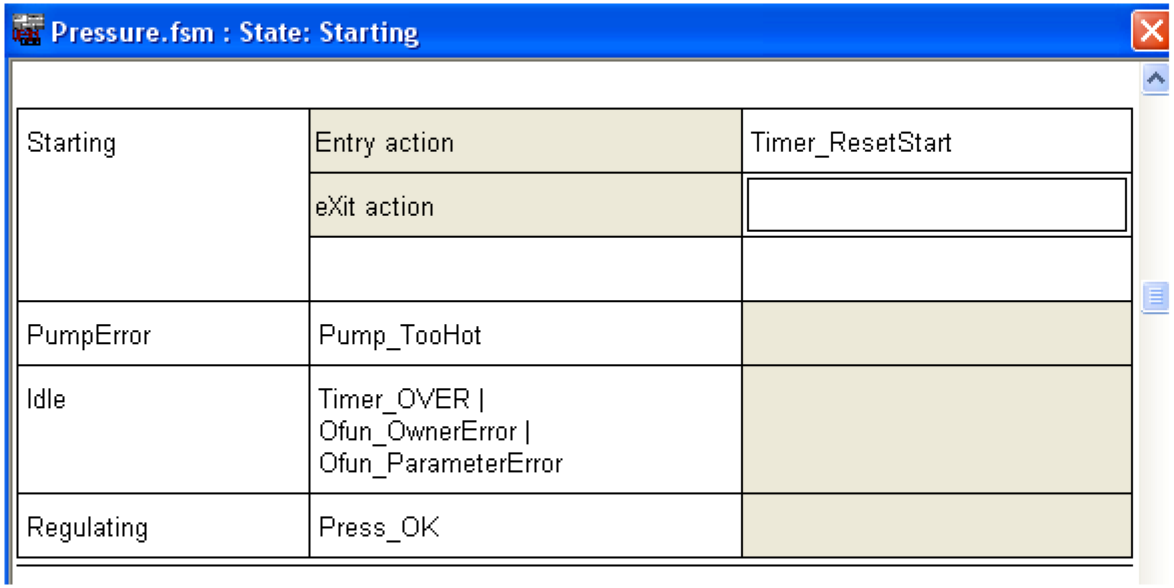
Specifying Transitions

- The ST diagram of the state ***Starting*** will look at that moment as shown below.

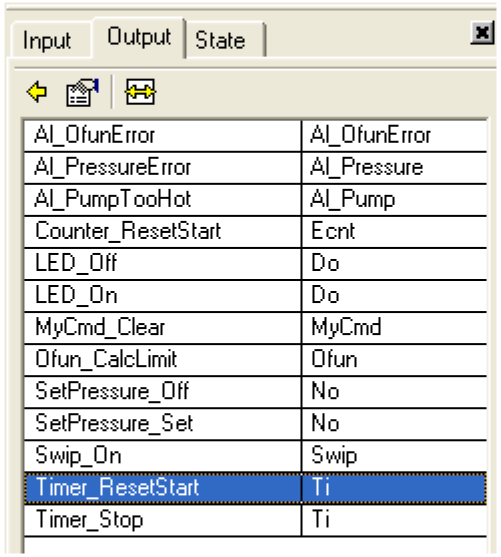


Specifying Entry and Exit Actions

- Position the cursor in the field **Entry action:** the tab **Output** will be active.
- Select a name in the tab **Output** (***Timer_ResetStart***) and by clicking on the name copy it to the Entry action field.


Pressure.fsm : State: Starting

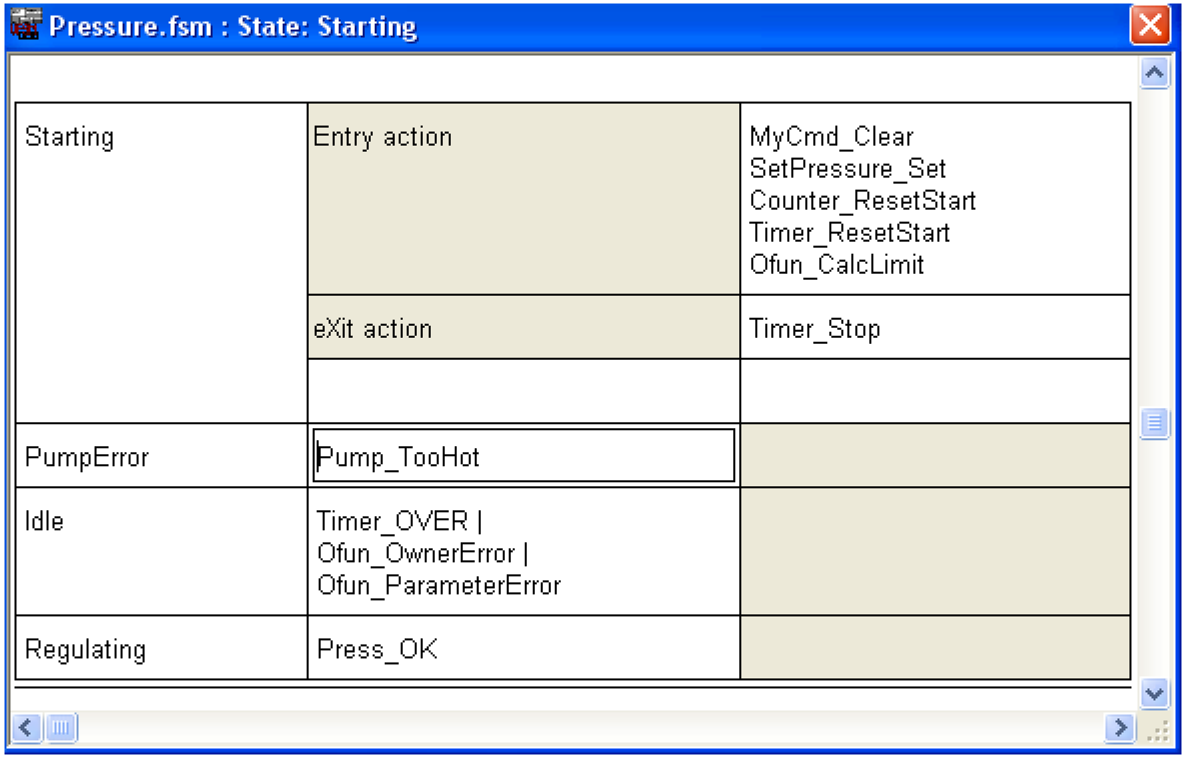
Starting	Entry action	Timer_ResetStart
	eXit action	
PumpError	Pump_TooHot	
Idle	Timer_OVER | Ofun_OwnerError | Ofun_ParameterError	
Regulating	Press_OK	



AI_OfunError	AI_OfunError
AI_PressureError	AI_Pressure
AI_PumpTooHot	AI_Pump
Counter_ResetStart	Ecnt
LED_Off	Do
LED_On	Do
MyCmd_Clear	MyCmd
Ofun_CalcLimit	Ofun
SetPressure_Off	No
SetPressure_Set	No
Swip_On	Swip
Timer_ResetStart	Ti
Timer_Stop	Ti

Specifying Actions

- Repeating that procedure for fields **Entry action** and **eXit action** you get the **ST table** shown below.

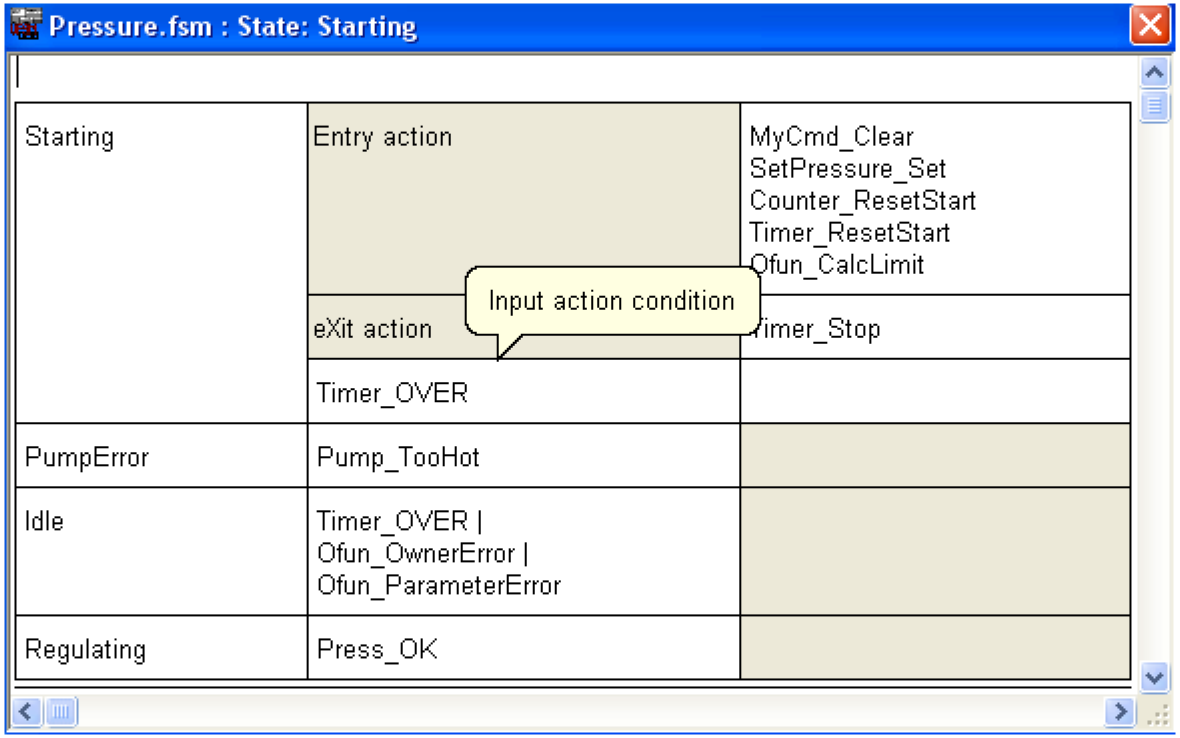


Pressure.fsm : State: Starting

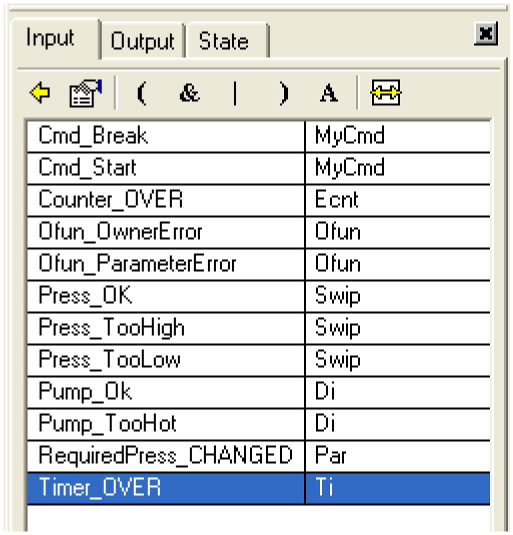
Starting	Entry action	MyCmd_Clear SetPressure_Set Counter_ResetStart Timer_ResetStart Ofun_CalcLimit
	eXit action	Timer_Stop
PumpError	Pump_TooHot	
Idle	Timer_OVER | Ofun_OwnerError | Ofun_ParameterError	
Regulating	Press_OK	

Specifying Actions

- Specification of Input actions requires:
  - Definition of input action condition: position the cursor in the field **Input action condition**.
  - Select the name (***Timer_OVER***) in the tab **Input** and clicking on it copy it to the field **Input action condition**.



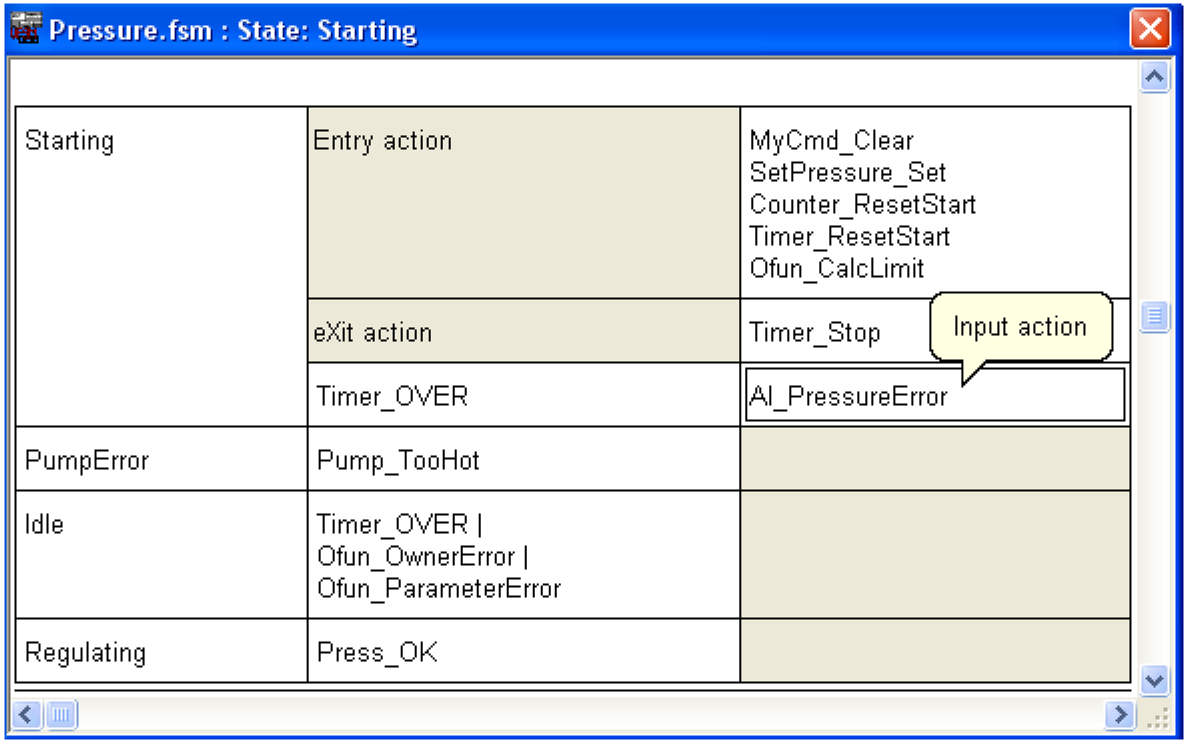
Starting	Entry action	MyCmd_Clear SetPressure_Set Counter_ResetStart Timer_ResetStart Ofun_CalcLimit
	eXit action	Timer_Stop
	Timer_OVER	
PumpError	Pump_TooHot	
Idle	Timer_OVER | Ofun_OwnerError | Ofun_ParameterError	
Regulating	Press_OK	



Cmd_Break	MyCmd
Cmd_Start	MyCmd
Counter_OVER	Ecnt
Ofun_OwnerError	Ofun
Ofun_ParameterError	Ofun
Press_OK	Swip
Press_TooHigh	Swip
Press_TooLow	Swip
Pump_Ok	Di
Pump_TooHot	Di
RequiredPress_CHANGED	Par
Timer_OVER	Ti

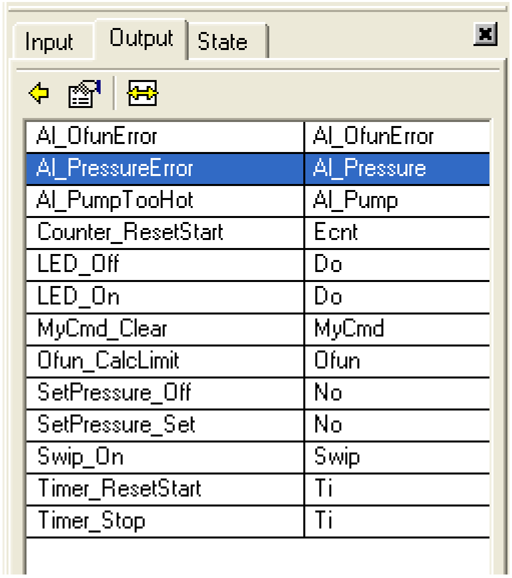
Specifying Actions

- and:
  - Definition of action: position the cursor in the field **Output action**.
  - Select the name (***AI_PressureError***) in the tab **Output** and by clicking on it copy it to the field **Output action**.



Pressure.fsm : State: Starting

Starting	Entry action	MyCmd_Clear SetPressure_Set Counter_ResetStart Timer_ResetStart Ofun_CalcLimit
	eXit action	Timer_Stop
	Timer_OVER	AI_PressureError
PumpError	Pump_TooHot	
Idle	Timer_OVER | Ofun_OwnerError | Ofun_ParameterError	
Regulating	Press_OK	



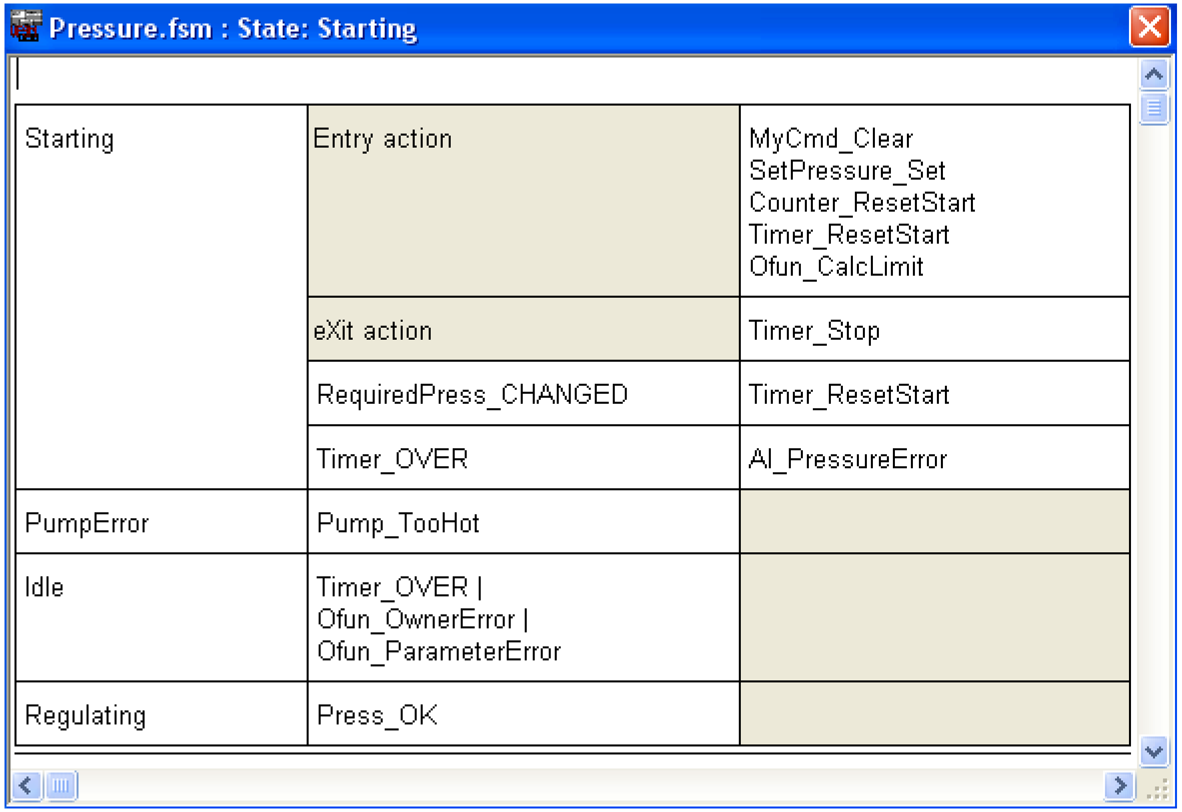
AI_OfunError	AI_OfunError
AI_PressureError	AI_Pressure
AI_PumpTooHot	AI_Pump
Counter_ResetStart	Ecnt
LED_Off	Do
LED_On	Do
MyCmd_Clear	MyCmd
Ofun_CalcLimit	Ofun
SetPressure_Off	No
SetPressure_Set	No
Swip_On	Swip
Timer_ResetStart	Ti
Timer_Stop	Ti

Specifying Actions

- Using toolbar tabs or command **Insert**, **Append**, **Delete expression** in menu **Edit** you may specify any number of **Input action expressions**.
- The sequence of **Input action expressions** may be changed using toolbar arrows or commands in menu **Edit**.
- The sequence of **Input action expressions** does not play any role and it does not define any execution priority. Changing of the sequence is provided for the user's convenience, to make his documentation easier to follow.

Specifying ST table

- Eventually the **ST table** may be complete as shown below.

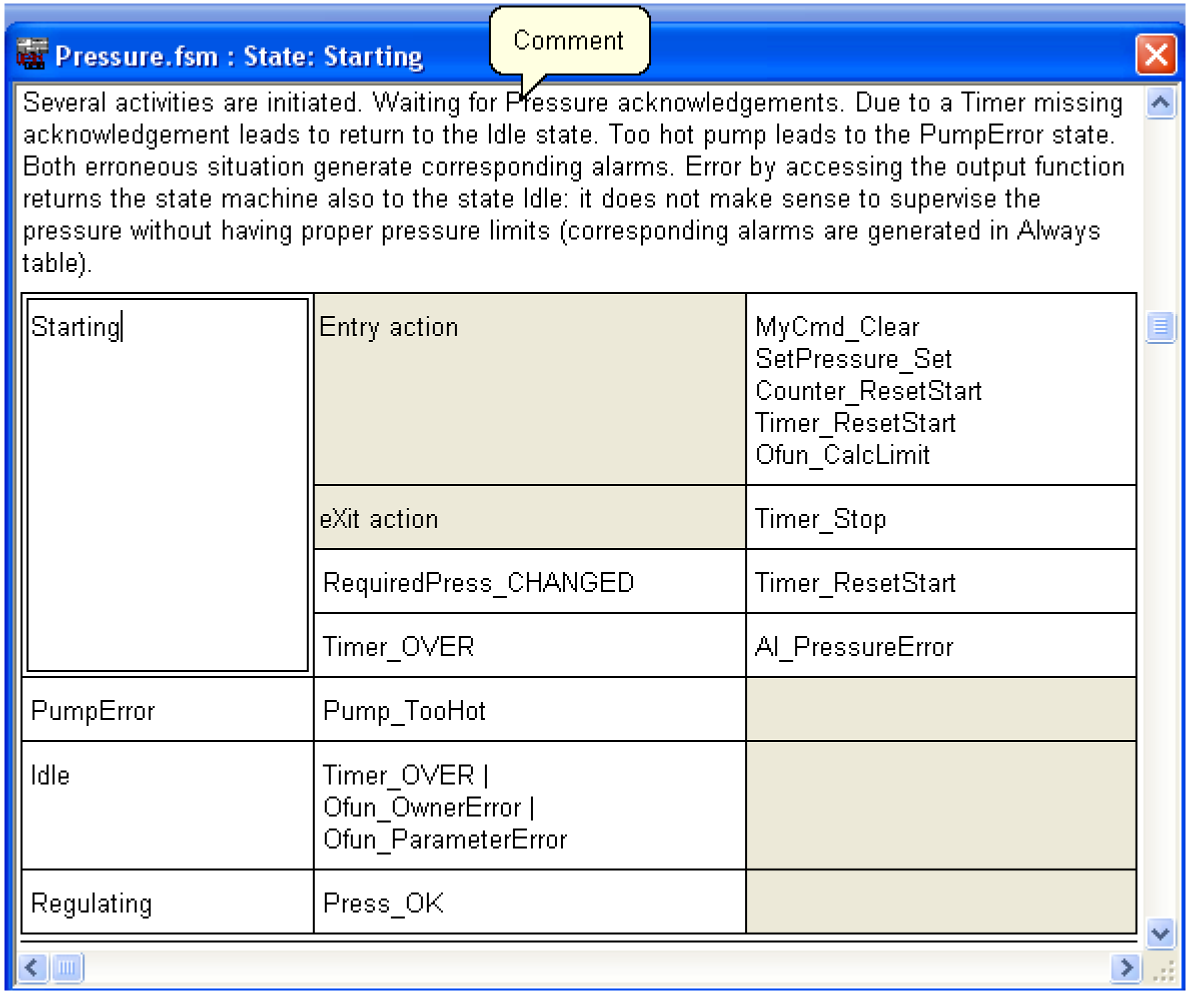


Pressure.fsm : State: Starting

Starting	Entry action	MyCmd_Clear SetPressure_Set Counter_ResetStart Timer_ResetStart Ofun_CalcLimit
	eXit action	Timer_Stop
	RequiredPress_CHANGED	Timer_ResetStart
	Timer_OVER	AI_PressureError
PumpError	Pump_TooHot	
Idle	Timer_OVER | Ofun_OwnerError | Ofun_ParameterError	
Regulating	Press_OK	

Specifying ST table

- To make it easier to understand you should add a description in the field **Comment**.

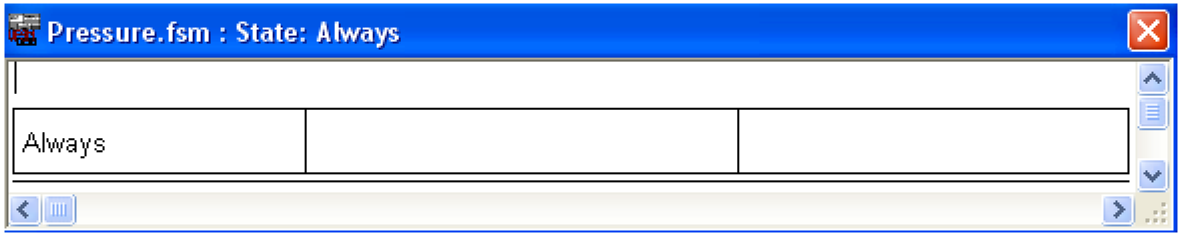


Several activities are initiated. Waiting for Pressure acknowledgements. Due to a Timer missing acknowledgement leads to return to the Idle state. Too hot pump leads to the PumpError state. Both erroneous situation generate corresponding alarms. Error by accessing the output function returns the state machine also to the state Idle: it does not make sense to supervise the pressure without having proper pressure limits (corresponding alarms are generated in Always table).

Starting	Entry action	MyCmd_Clear SetPressure_Set Counter_ResetStart Timer_ResetStart Ofun_CalcLimit
	eXit action	Timer_Stop
	RequiredPress_CHANGED	Timer_ResetStart
	Timer_OVER	Al_PressureError
PumpError	Pump_TooHot	
Idle	Timer_OVER | Ofun_OwnerError | Ofun_ParameterError	
Regulating	Press_OK	

Specifying the table Always

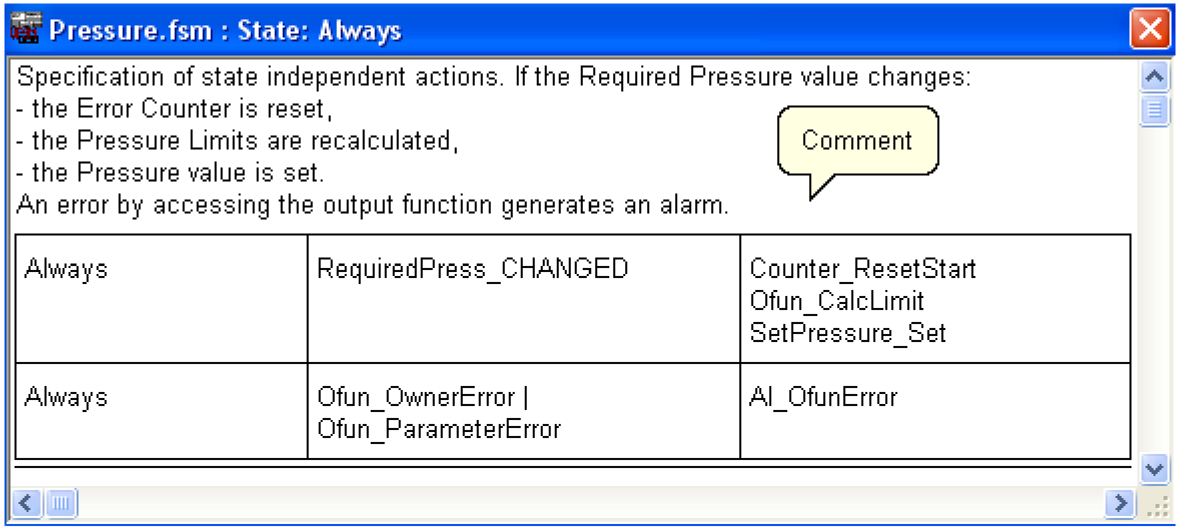
- You may specify Input actions valid for all states.
- Clicking on the table **Always** in the **ST diagram** open the table as below.
- You may define, edit and manipulate the content of fields: **Input action condition** and **Input action** in a very similar way as in the **ST table**.
- You may add a description in the **Comment** field.



Always		

Specifying the table Always

- Eventually you get the content of the table **Always** as shown below.



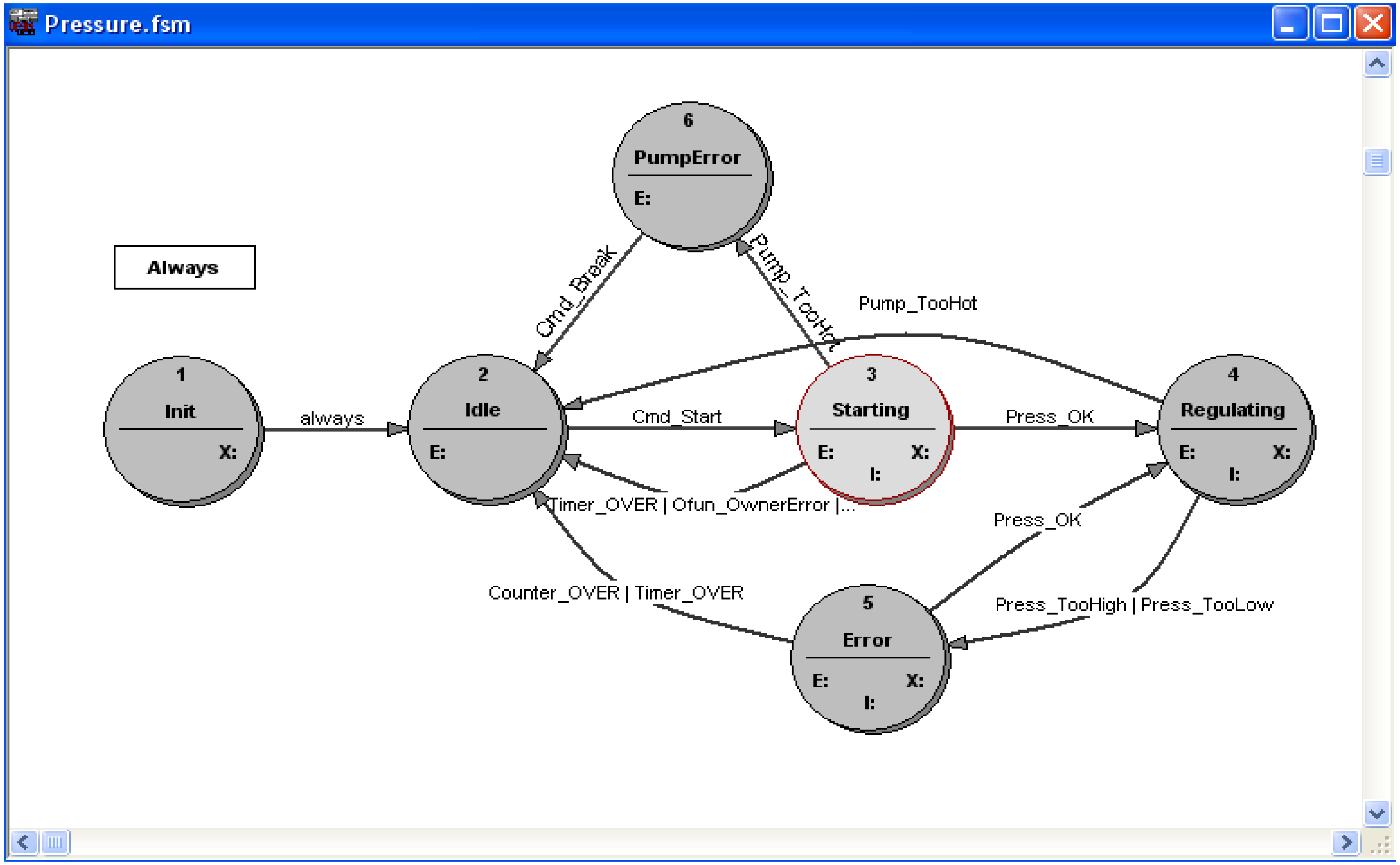
Specification of state independent actions. If the Required Pressure value changes:
- the Error Counter is reset,
- the Pressure Limits are recalculated,
- the Pressure value is set.
An error by accessing the output function generates an alarm.

Always	RequiredPress_CHANGED	Counter_ResetStart Ofun_CalcLimit SetPressure_Set
Always	Ofun_OwnerError | Ofun_ParameterError	Al_OfunError

Your VFSM specification is ready

- If you have specified **ST tables** for all states and the table **Always** the task is done: the virtual finite state machine is specified and its **ST diagram** is shown below

  Note: Open Pressure.fsm from the VFSM folder.



References

[1] Wagner F., al., *Modeling Software with Finite State Machines: A Practical Approach*. Taylor & Francis CRC Press, 2006.
[2] StateWORKS Studio Help.
[3] StateWORKS Development Tools: User’s Guide & Training Manual. SW Software 2005.
[4] www.stateworks.com - Technical Notes.