

# StateWORKS

## Specifying a System of State Machines - Tutorial

# Introduction

- ◆ The tutorial teaches you how to prepare the specification of a system of virtual finite state machines using StateWORKS Studio.
- ◆ We assume that you know about the specification of a virtual finite state machine (see [5])
- ◆ When it starts StateWORKS Studio opens the last project or nothing if started the first time.
- ◆ A system of virtual finite state machines can be specified in a StateWORKS project. Starting the new project you will be asked to close the present one.

# Introduction

- ◆ The tutorial uses the project ***Pumps*** from the book [1] to illustrate the design steps. In the book you find detailed requirements and analysis of the control task.
- ◆ For the purpose of this tutorial some 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

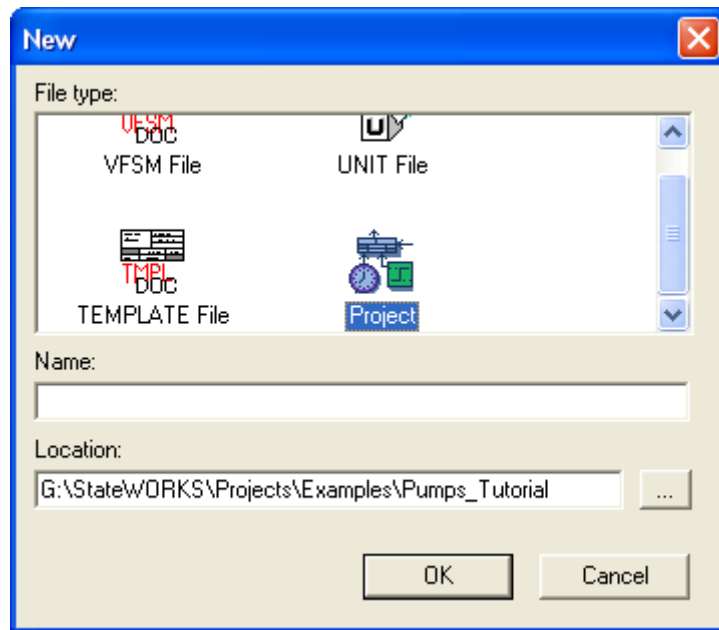
- ◆ **Input action condition**  
A condition defined using Input names linked by AND and OR operators
- ◆ **Input action expression**  
Input action condition and Input action
- ◆ **Input action priority**  
The sequence of Input action expressions in the ST table; used for documentation purpose
- ◆ **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

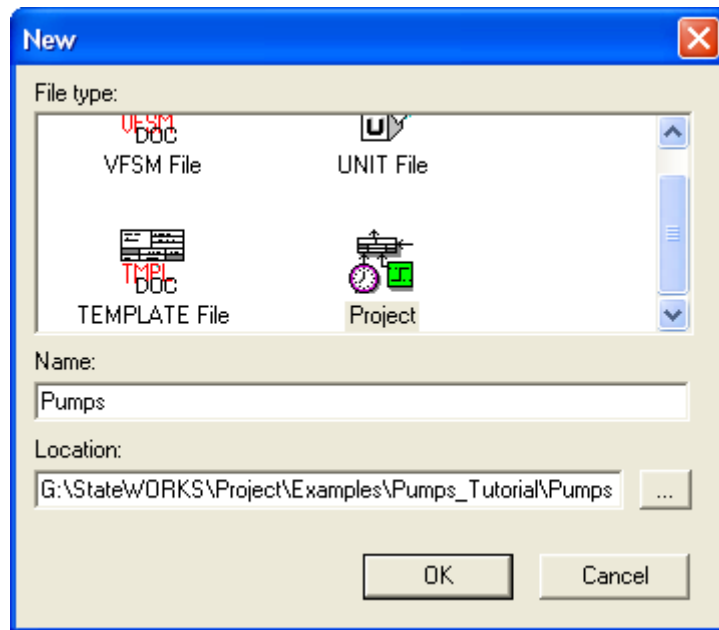
# Creating a project (a system of finite state machines)

- ◆ Click on the **New** button on the toolbar or on the command **New** in the menu **File**.
- ◆ Select the **Project** icon in the dialog windows.



# Creating a project (a system of finite state machines)

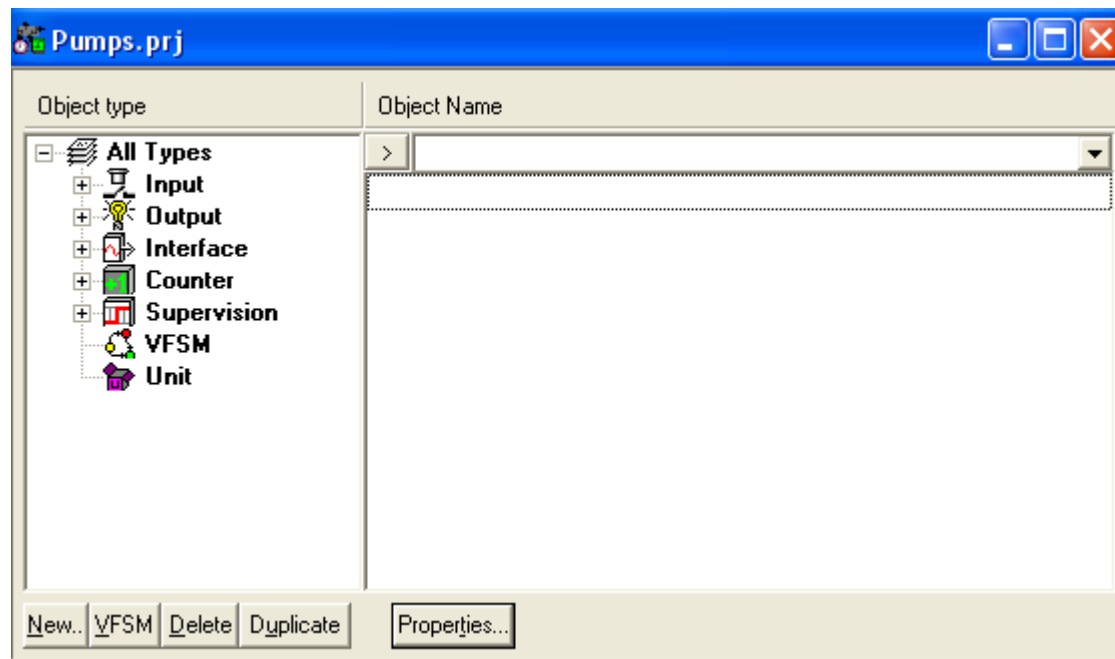
- ◆ Select a directory for the project by browsing / editing the edit box **Location**.
- ◆ Give the project a name editing the edit box **Name**. That would be also the actual directory of the project.
- ◆ Leave this dialog window by clicking on the button **OK**.





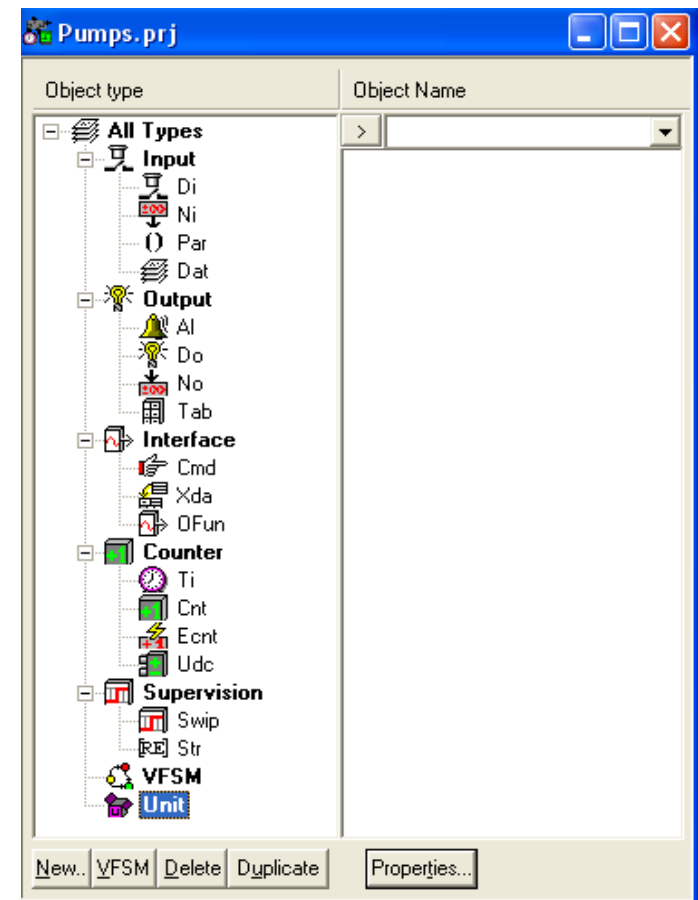
# Creating a project (a system of finite state machines)

- ◆ When leaving the dialog window you will be asked whether to close the present project; react appropriately.
- ◆ You will be confronted with an empty project window as shown below.



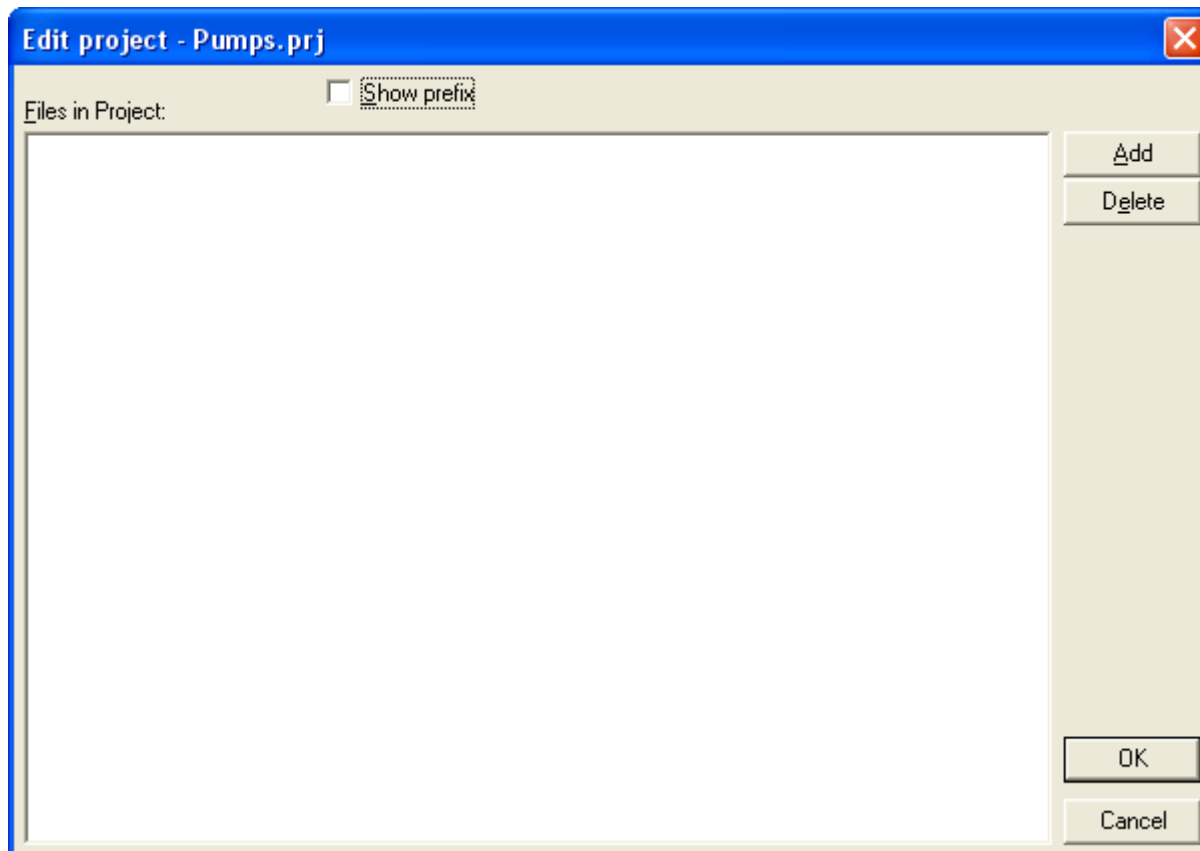
# Defining VFSM types used in the project

- ◆ By expanding the groups of objects (**Input**, etc.) display all RTDB object types available in the project.
- ◆ The **VFSM** and **Unit** groups are empty. Hence, you have to add types to the project needed for the system of state machines.
- ◆ For the purpose of this tutorial we will add only the **VFSM** types.



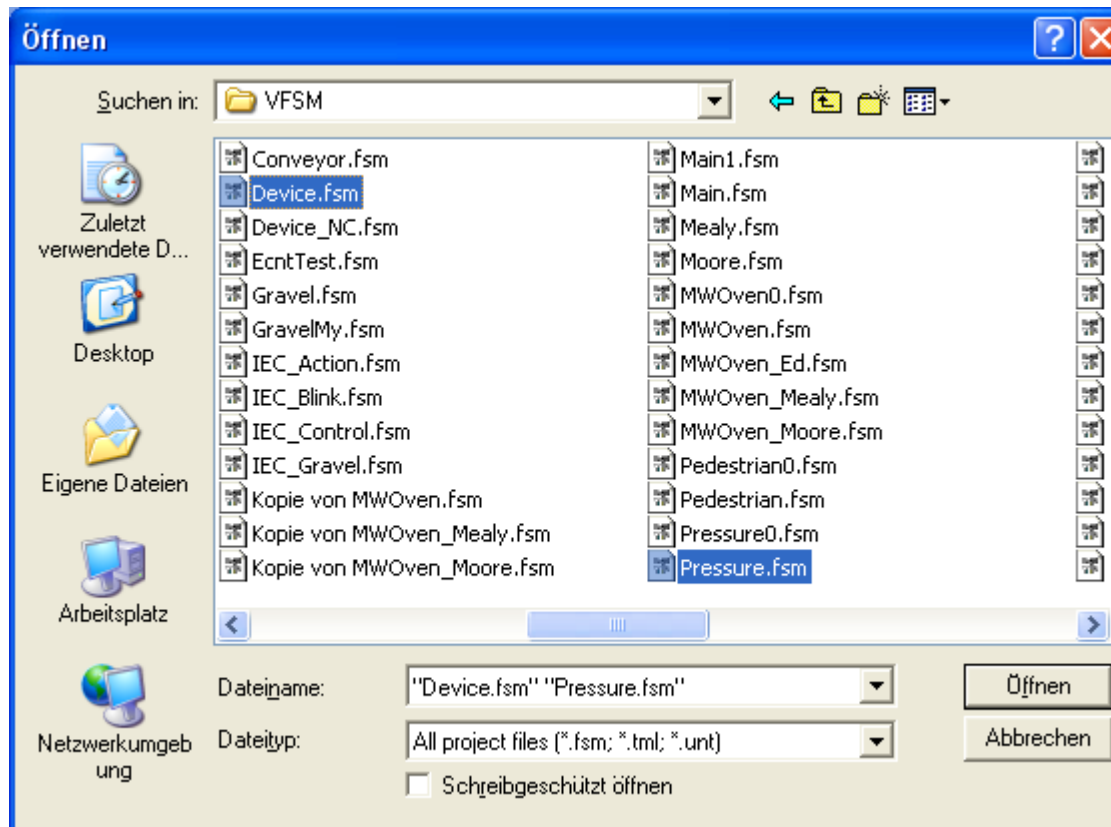
# Defining VFSM types used in the project

- ◆ Using command **Edit** in the menu **Project** open the dialog window which may be used to add VFSM and UNIT types to the project.



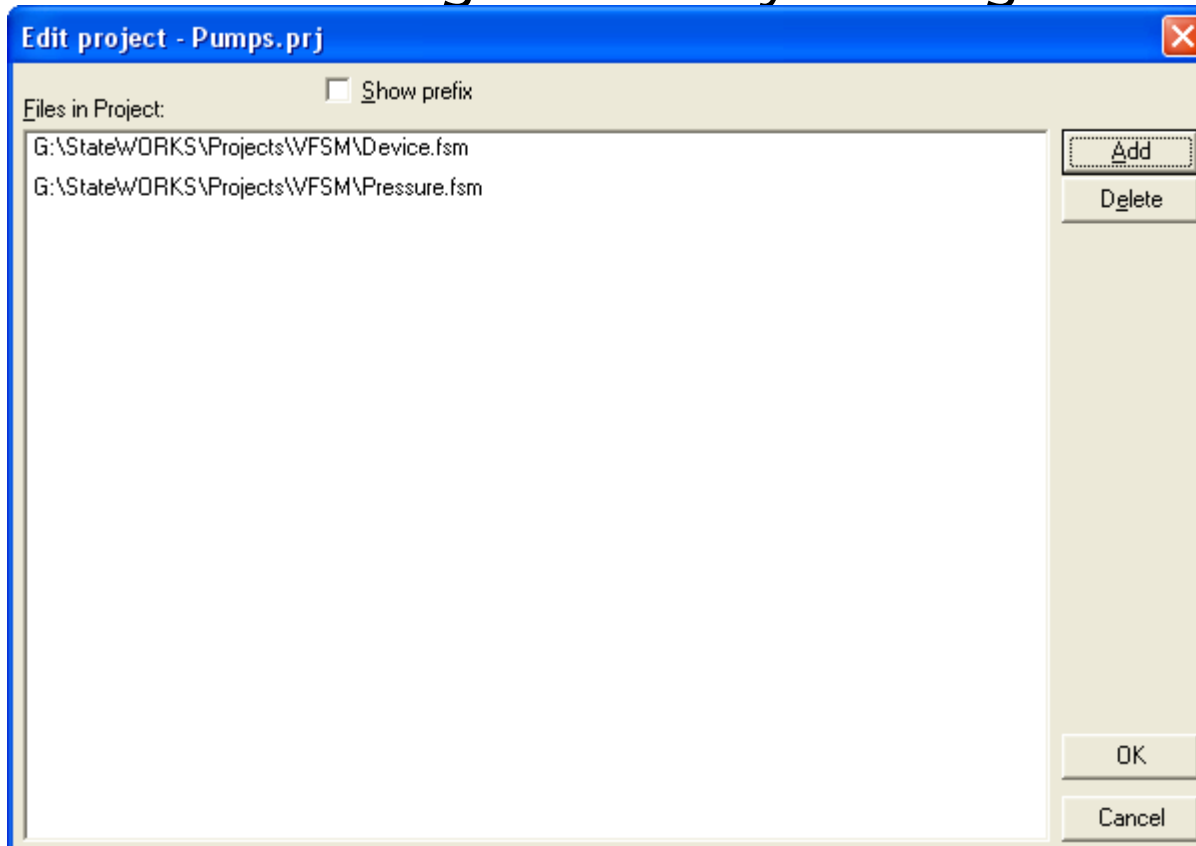
# Defining VFSM types used in the project

- Click on the button **Add** to open the File browser.
- Select fsm-files of state machines **Device** and **Pressure**.
- Close the browser clicking on the button **Open**.



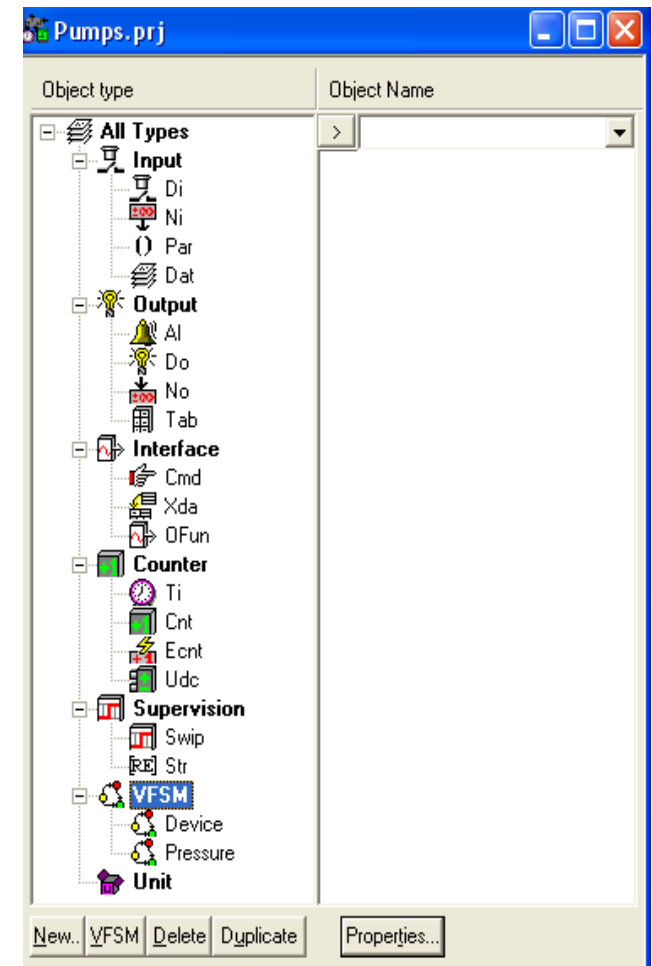
# Defining VFSM types used in the project

- ◆ The VFSM types **Device** and **Pressure** have been added to the project.
- ◆ You may **Add** or **Delete** VFSM types at any time while specifying the system.
- ◆ Leave the dialog window by clicking on the button **OK**.



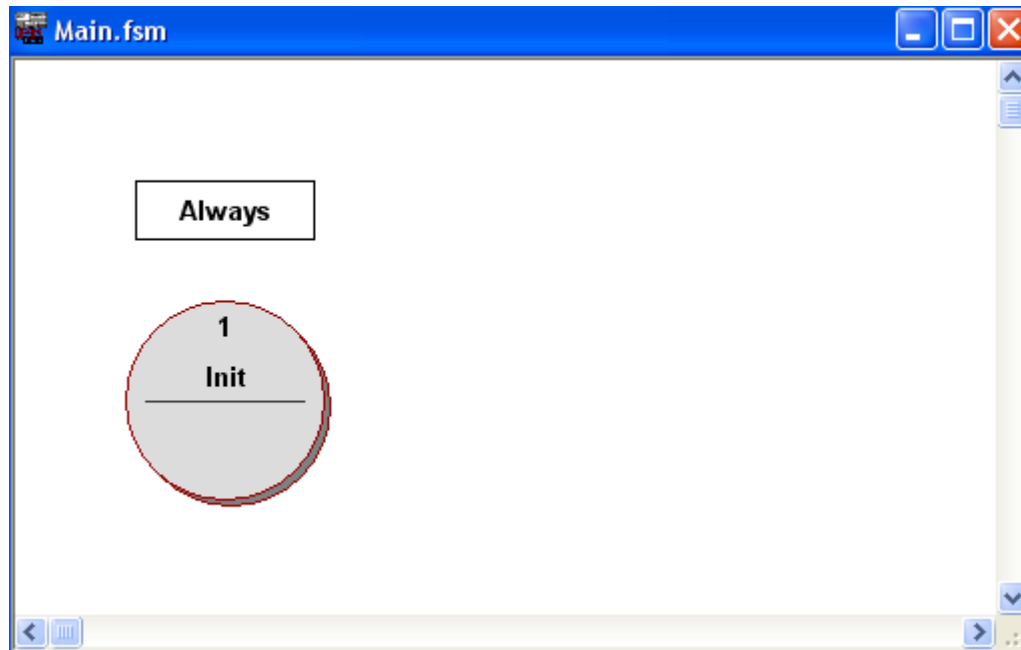
# Defining VFSM types used in the project

- ◆ The added VFSM types appear in the group **VFSM** of the RTDB objects.
- ◆ The VFSM types can be used in the system specification exactly as any other RTDB objects, especially you may define several instances of those types.
- ◆ A click on the VFSM name (for instance **Pressure**) opens the ST diagram of the state machine Pressure. Similarly, using the toolbar icon **Project VFSM files** you may display a list of VFSM types in the project and open the required ST diagram.



# Creating a Master state machine

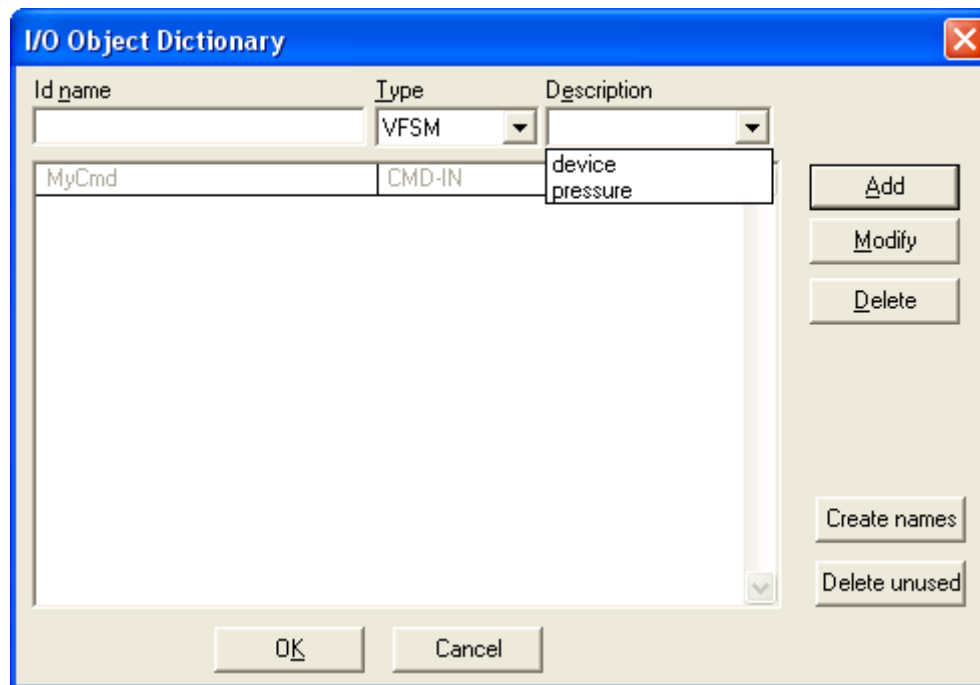
- ◆ To define a Master state machine create a new state machine (see the Tutorial [5]).
- ◆ Store the state machine giving it a name **Main** and the prefix **MAI**.





# Defining required objects

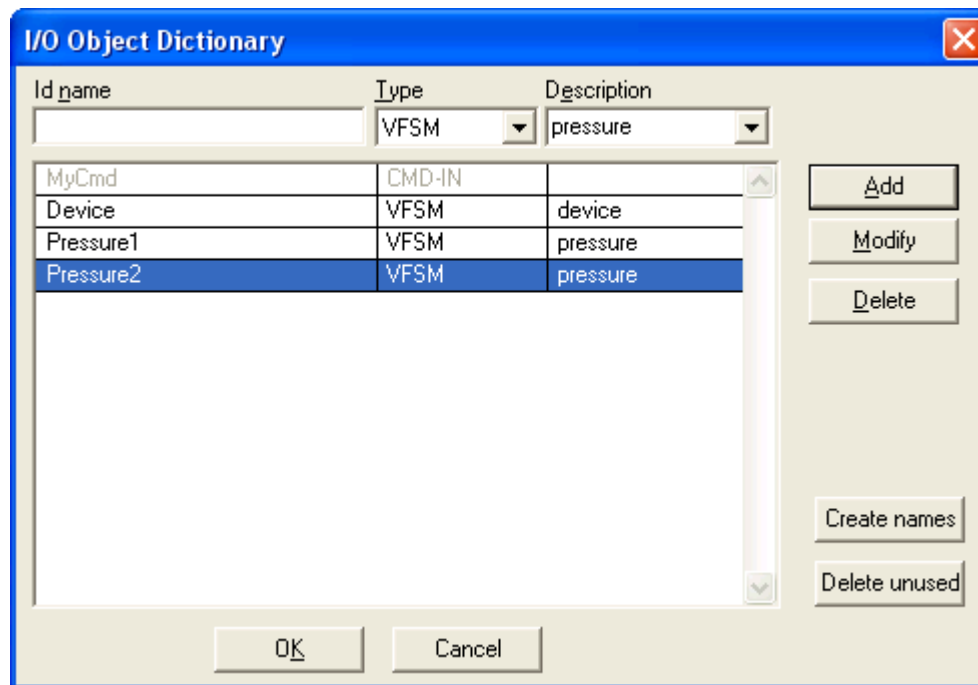
- ◆ Open the **I/O Object Dictionary**.
- ◆ Select in the field **Type** the type **VFSM**.
- ◆ Open the combo box **Description** which displays a list of VFSM types in the project: device and pressure.





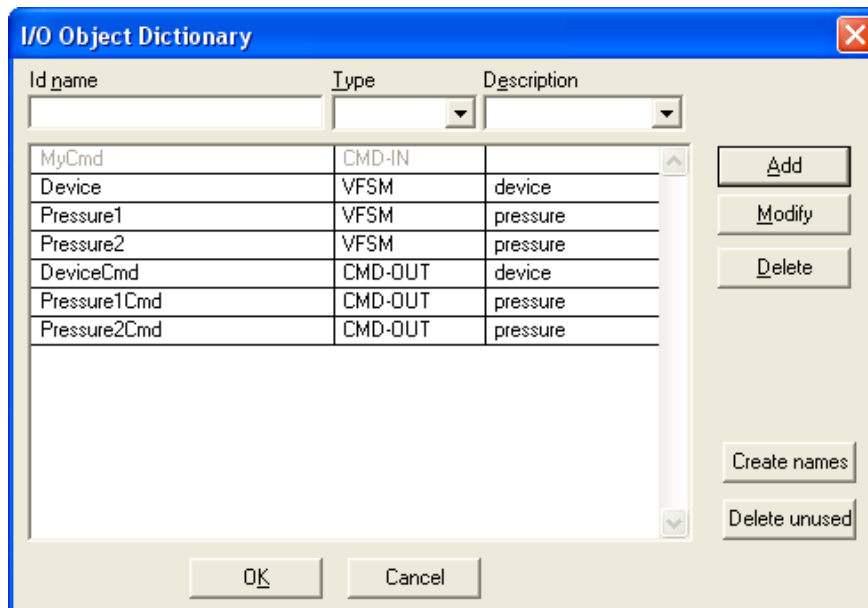
# Defining required objects

- ◆ Add the VFSM of type Device to the **I/O Dictionary**; give it a name **Device**.
- ◆ Add two VFSMs of type Pressure to the I/O Dictionary; give them names **Pressure1** and **Pressure2**.



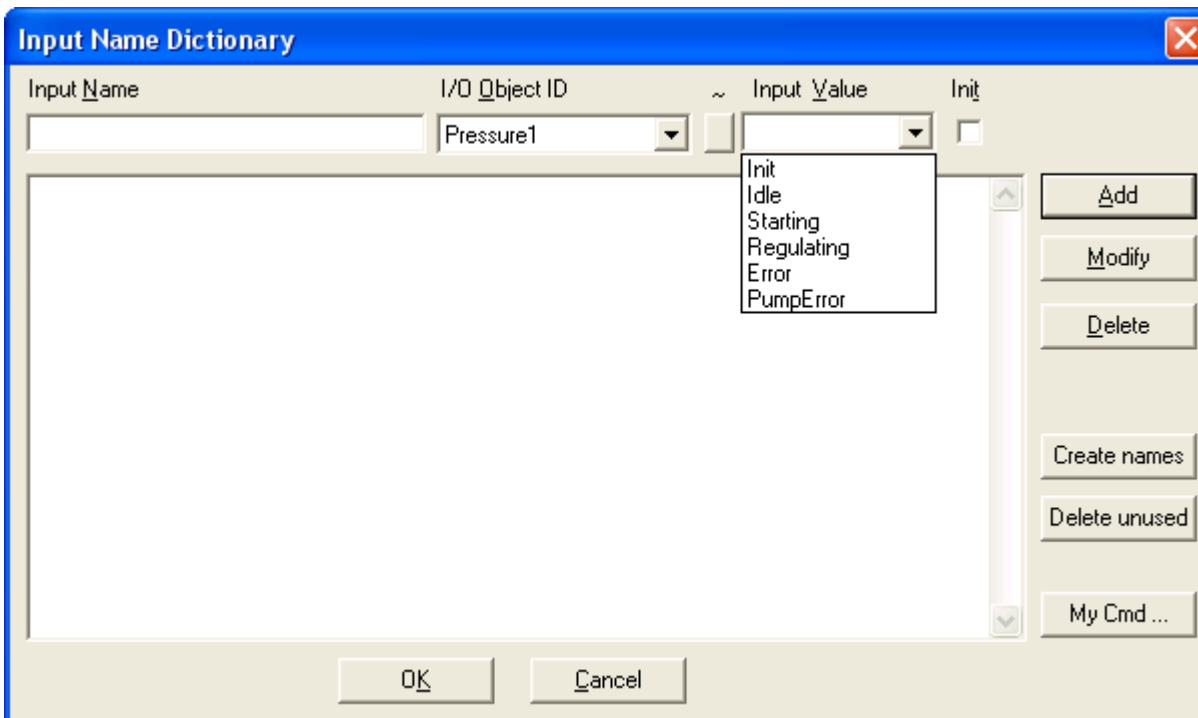
# Defining required objects

- ◆ Similarly, add the CMD-OUT of type Device to the I/O Dictionary; give it a name **DeviceCmd**.
- ◆ Similarly, add two CMD-OUT of type Pressure to the I/O Dictionary; give them names **Pressure1Cmd** and **Pressure2Cmd**.
- ◆ Note that the Master state machine has already been given the default object **MyCmd** of a type CMD-IN.



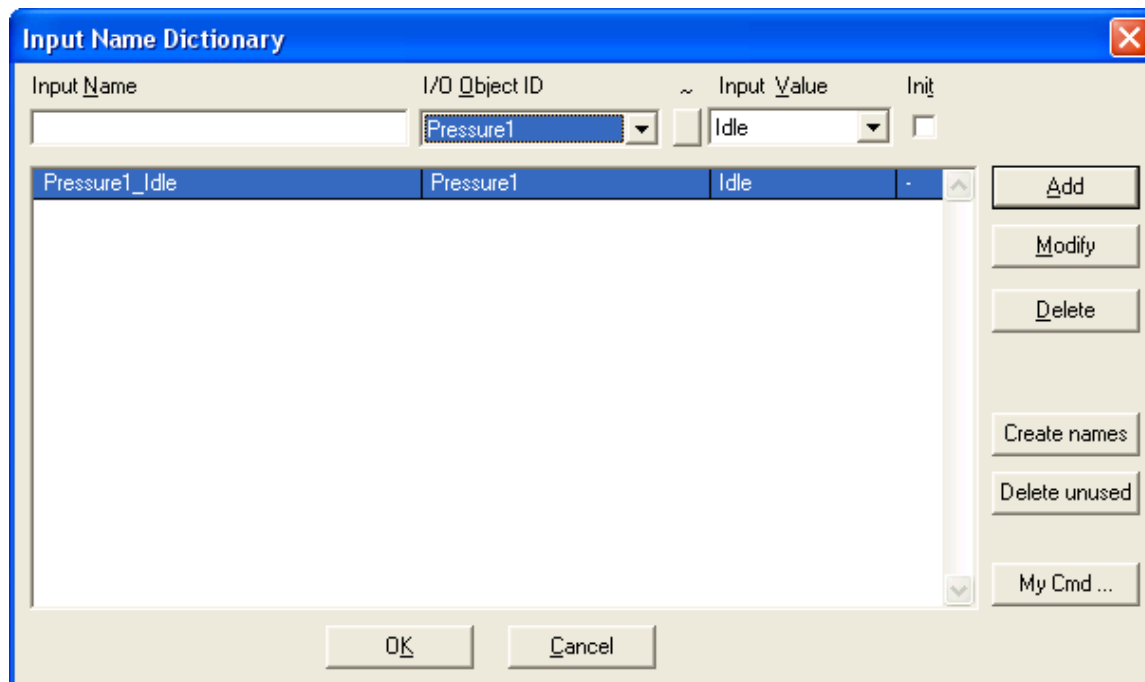
# Defining Master Input Names

- ◆ Open the **Input Name Dictionary**
- ◆ Select an **I/O Object ID** of type **Pressure1**.
- ◆ Open the list of object **Input Values**, which displays all the states of a state machine Pressure.



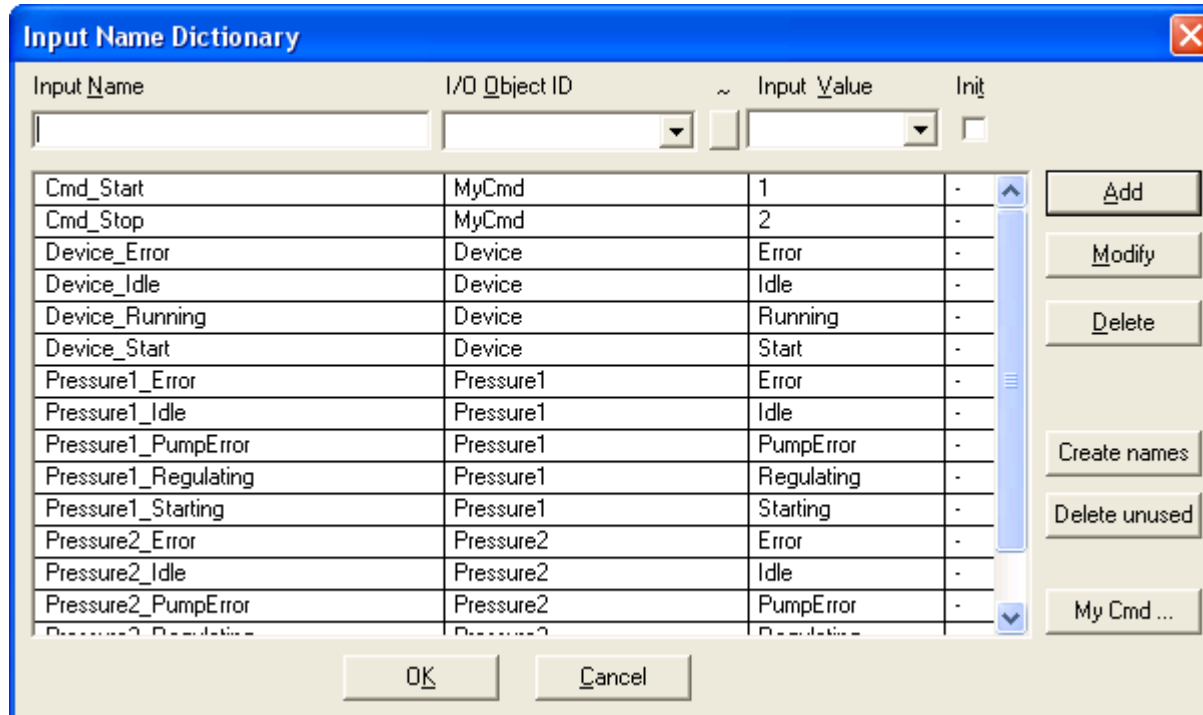
# Defining Master Input Names

- ◆ Select a value (effectively a state) in the field **Input Value**, for instance **Idle**.
- ◆ A click on the button **Add** creates a default Input Name; you may edit it or accept as it is.
- ◆ A next click on the button **Add** adds the name to the **Input Name Dictionary**.



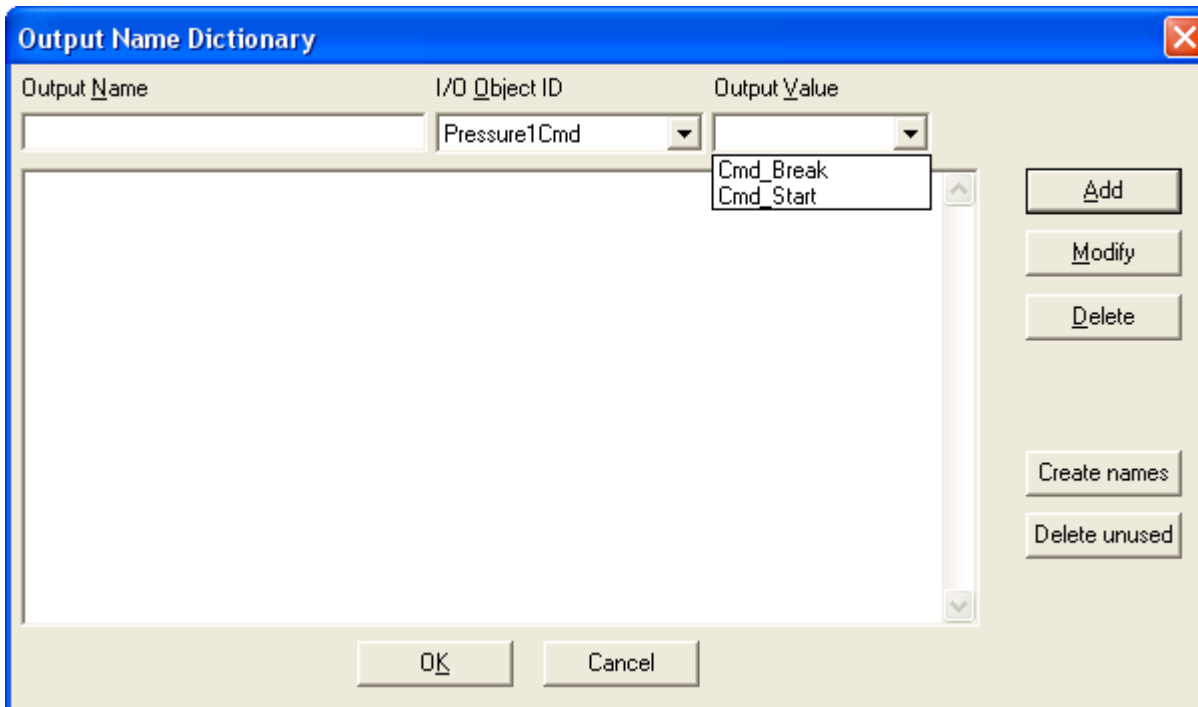
# Defining Master Input Names

- ◆ If you define all required **Input Names** the **Input Name Dictionary** may look for instance as below.  
Note: Open Main\_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**.



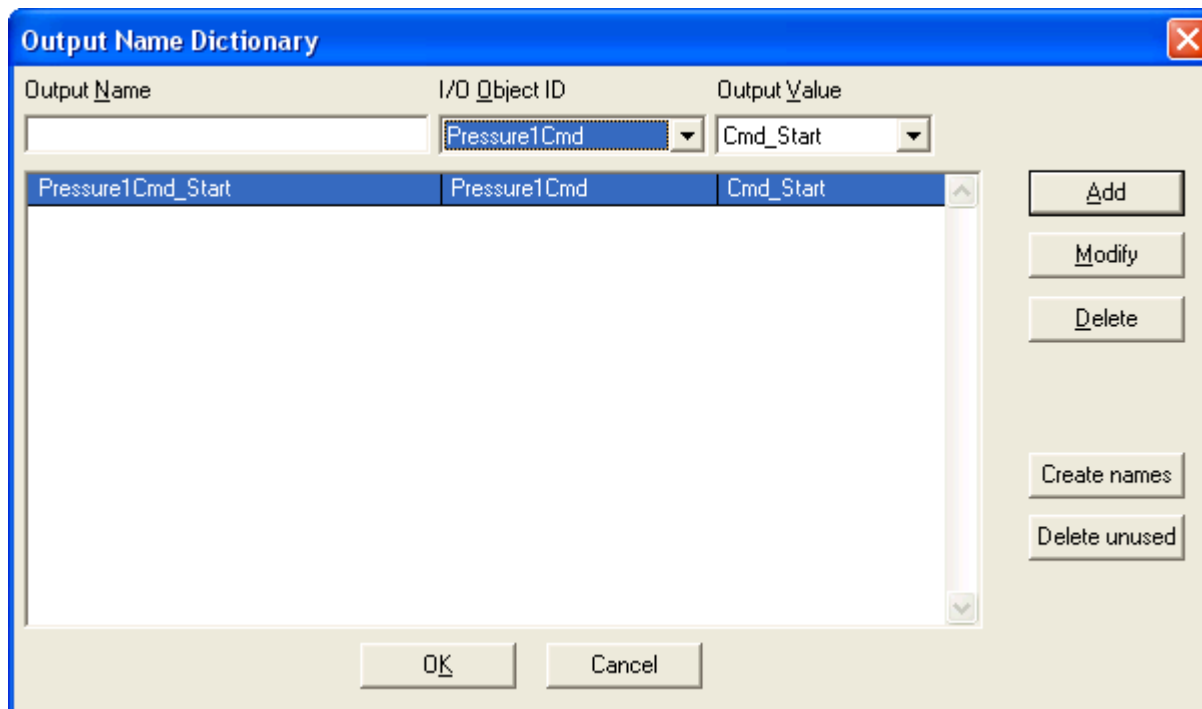
# Defining Master Output Names

- ◆ Open the **Output Name Dictionary**
- ◆ Select an **I/O Object ID** of type **Pressure1Cmd**.
- ◆ Open the list of object **Output Values** which displays the commands of a state machine Pressure.



# Defining Master Output Names

- ◆ Select a value (effectively a command) in the field **Output Value**, for instance **Cmd\_Start**.
- ◆ Clicking on the button **Add** suggests a default Output Name; you may edit it or accept as it is.
- ◆ A next click on the button **Add** adds the name to the **Output Name Dictionary**.





# Defining Master Output Names

- ◆ If you define all required **Output Names** the **Output Name Dictionary** may look for instance as below.  
Note: Open Main\_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**.

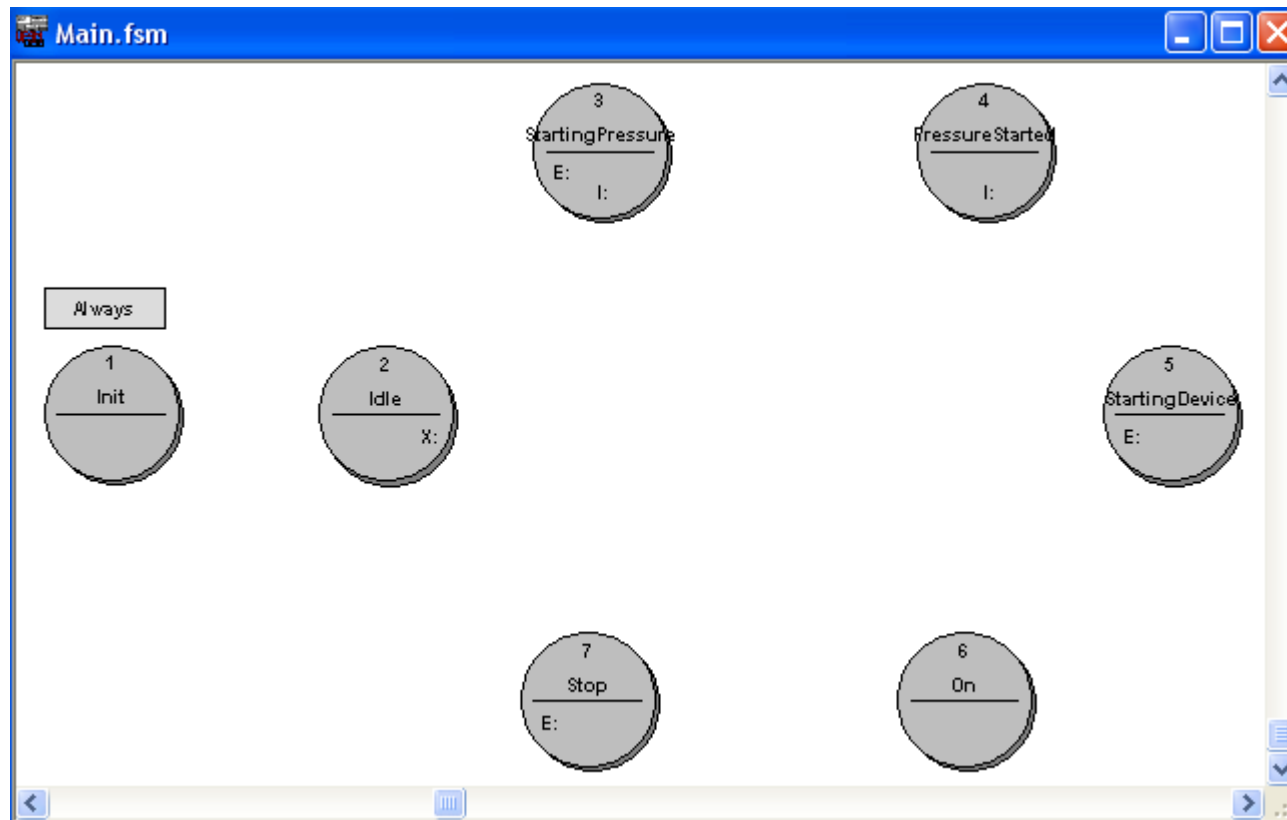
Output Name	I/O Object ID	Output Value
Cmd_Clear	MyCmd	0
DeviceCmd_Off	DeviceCmd	MyCmd_Off
DeviceCmd_On	DeviceCmd	MyCmd_On
Pressure1Cmd_Break	Pressure1Cmd	Cmd_Break
Pressure1Cmd_Start	Pressure1Cmd	Cmd_Start
Pressure2Cmd_Break	Pressure2Cmd	Cmd_Break
Pressure2Cmd_Start	Pressure2Cmd	Cmd_Start



# Defining Master State Names

- ◆ Create states which you need for a specification of the state machine Main.

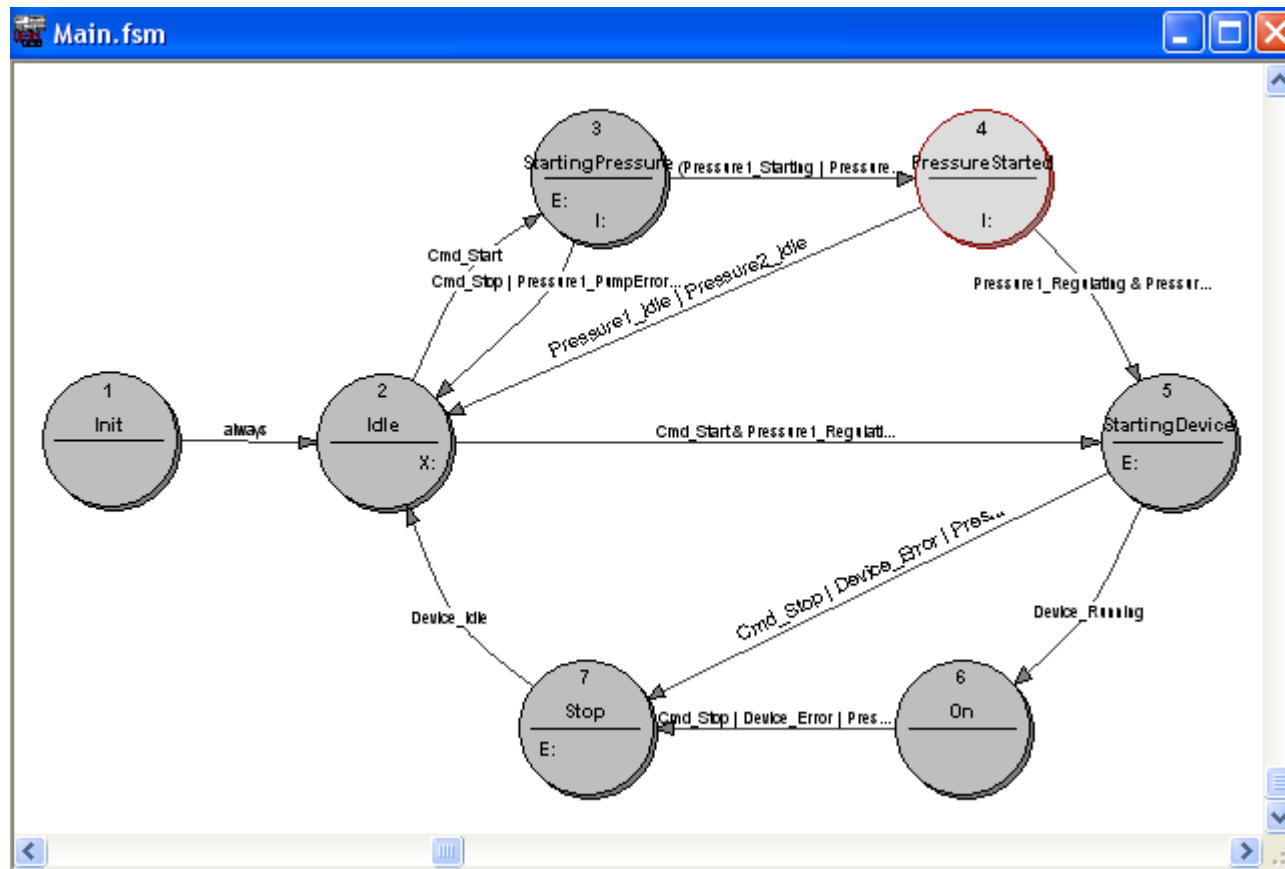
Note: You may open in this moment the Main\_StateNameDictionary.fsm from the Pumps\_Tutorial folder.



# Specifying Master state machine

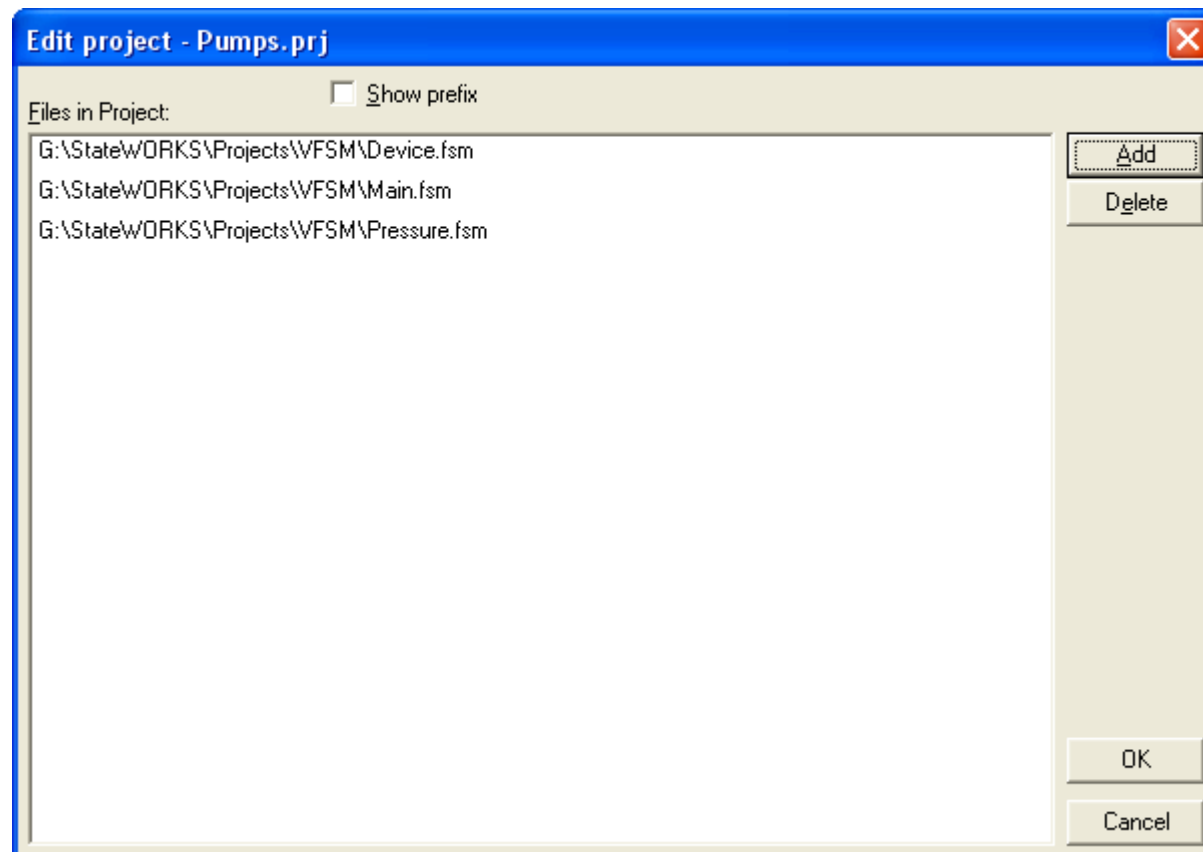
- ◆ The specification of a Master state machine is done exactly as any other state machine. The **ST diagram** of the state machine **Main** is shown below.

Note: Open Main.fsm from the VFMS folder.



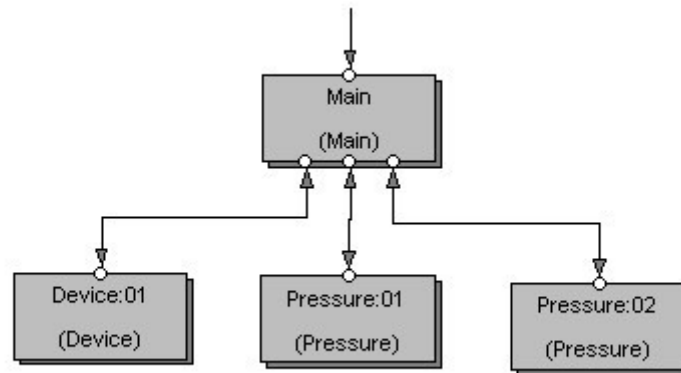
# Specifying Master state machine

- ◆ To complete the task add the **Main** state machine type to the project (**Project / Edit**).



# SMS diagram

- ◆ You have now created the system of state machines as shown in this **SMS diagram**.
- ◆ But to finish the process, and generate this diagram, you need now to do some further work to configure your Project.
- ◆ This process will be explained in the tutorial named "*Specifying RTDB (execution environment)*".



# 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](http://www.stateworks.com) - Technical Notes.
- [5] StateWORKS: Specifying a state machine - Tutorial.