Creating a Project

When a project is created in Logix5000, there are certain features which have to be considered.

Controller Type. The correct controller type must be configured. There are many to choose from.

Software Revision. The software selected must be the same revision as the firmware in the controller.

Controller Slot Number. This is extremely important. If you try to download to a controller that is in a different slot to the project, it will not download.

Project Name. A project name must be assigned. This can be any name relating to the plant or project.

Tasks, Programs and Routines

A Logix5000 project has an organizational structure which is different to a conventional PLC. It consists of the following

Task. These are mechanisms where programs and routines can be scheduled. If done efficiently, available bandwidth can be utilized effectively which will result in faster operation.

Program. Programs can be created in a logical fashion to suit the plant requirements. They can be given logical names and the order of execution can be decided. All the routines are created in programs.

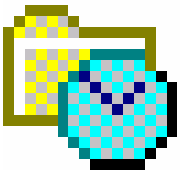
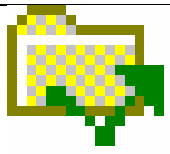
Routine. This is where the actual code is written. This decides the way the plant will operate. Four programming languages can be used. **Ladder Logic(LD), Function Block Diagrams(FBD), Structured Text(ST) and Sequential Function Charts(SFC).** These can be used for various purposes which are suited for a specific language. They can also be used according to personal preference.

Tasks. There are three types of task as shown below.

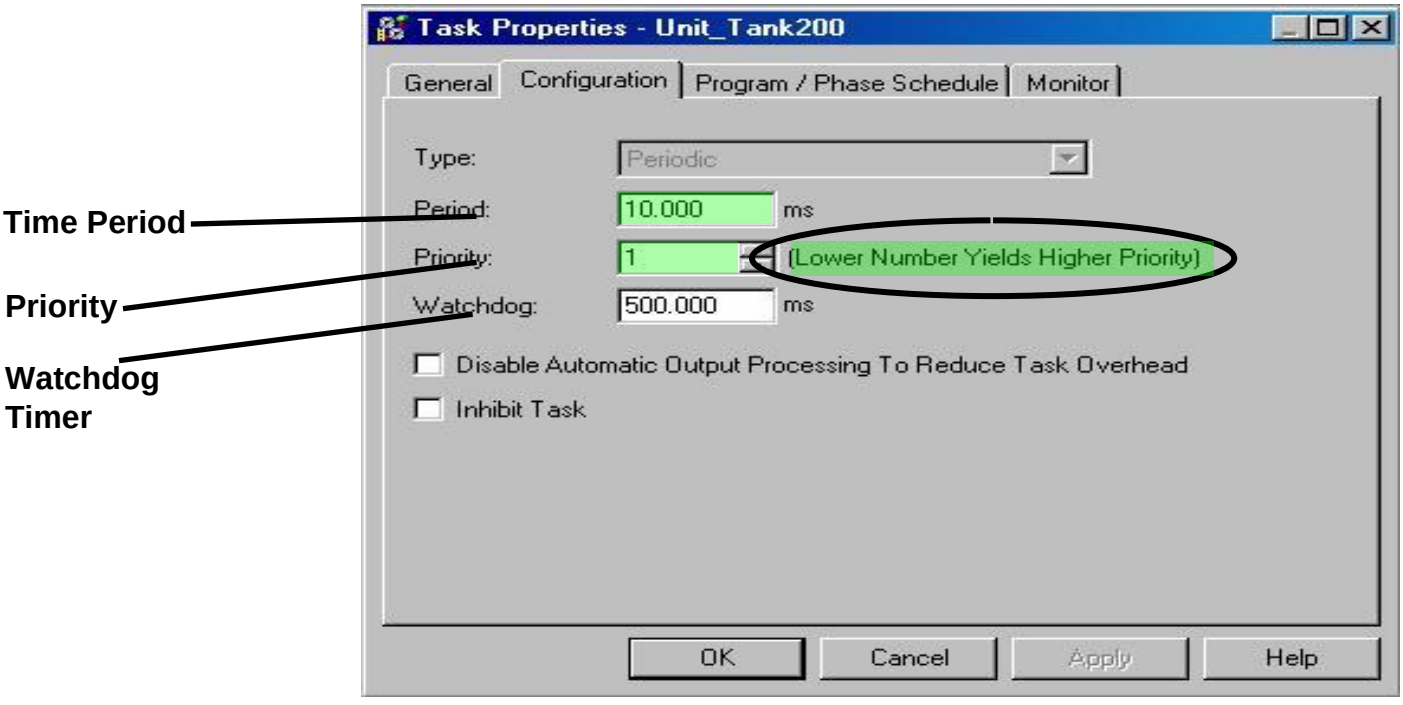
Continuous. When a project is created, there is one task created as default. This is a continuous task. One continuous task is permitted in a project. It is the lowest priority. Up to 32 tasks are permitted in a project.

Periodic. If a continuous task is retained, up to 31 periodic tasks can them be created. All will have different priorities and time intervals for execution. Up to 15 priorities can be set where 1 is the highest. It will always interrupt the continuous task and any other task of a lower priority.

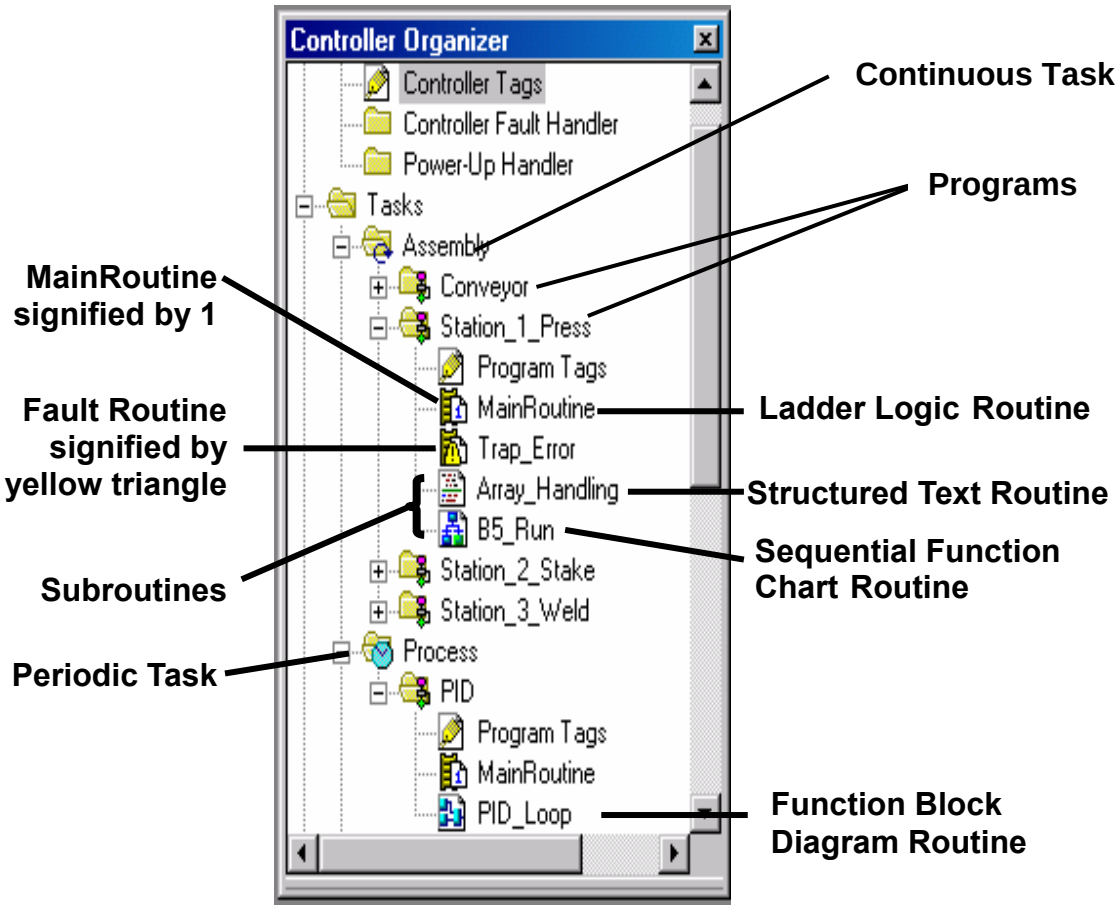
Event Task. An event task can be executed when a certain event takes place. This can be the change of state **(COS)** of a module or the execution of an event instruction. The following illustration briefly describes the functionality and the symbols for identification.

Task Type	Function	Symbol
Continuous	**This task will start and continue operation until interrupted by a higher priority task**	
Periodic	**This will execute after a certain time has elapsed. This time can be configured.**	
Event	**This will execute when a specific event has taken place such as a module update or the execution of an event instruction**	

Shown below is the configuration window for creating a periodic and event task. This is where the time interval for execution can be set and the priority. The Watchdog Timer is the maximum time allowed for the full execution of the task. If this is exceeded, the controller will fault.



Programs are the next level of scheduling. Within a logix5000 project, up to 100 programs can be configured in each task. The order of execution can be scheduled. See the illustration below, showing how the tasks, programs and routines are configured.



The illustration below shows how a project will look after initial creation. You have a **MainTask, MainProgram and MainRoutine.** Their names can be changed and assignments can be changed.

Default Structure and Names

There has to be a **Main Routine** in every program. A new project has a main routine assigned as default. Names can be changed and routines reassigned if required. A main routine is recognized by the 1 at the routine icon. It will execute as soon as the controller goes in to run mode.

Fault Routine. This is a routine that can be written to prevent the controller going to fault mode if an instruction execution for instance takes place. There can be 1 fault routine in each program. It is recognized with a yellow triangle next to the routine icon.

Sub Routine. These can be created within a program to divide and organize the project. They have to be called by a **JSR** instruction.

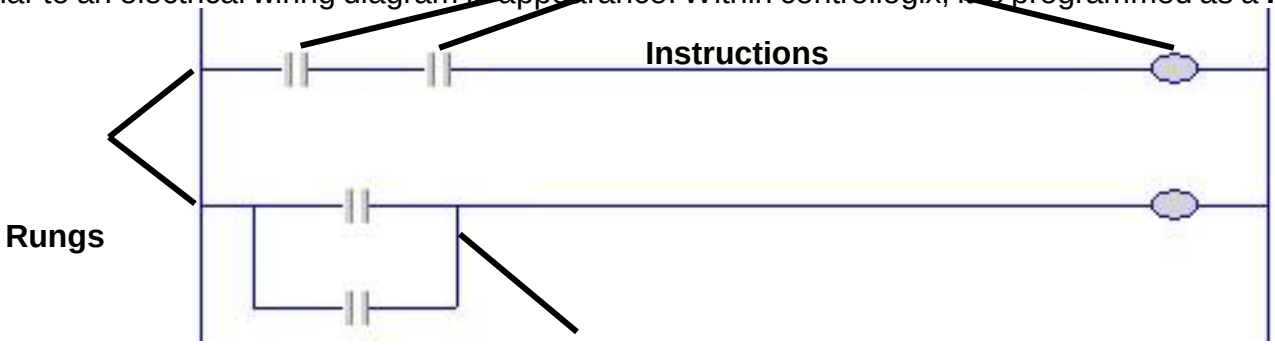
The number of routines that can be configured in one program is around 65000. The only limitation would be memory.

All the routine types can be created using any of the programming languages. All the programming languages are interactive, for example, a ladder subroutine can be executed, using an FBD or SFC to call the routine.

The main routine and fault routine can be created using any of the programming languages.

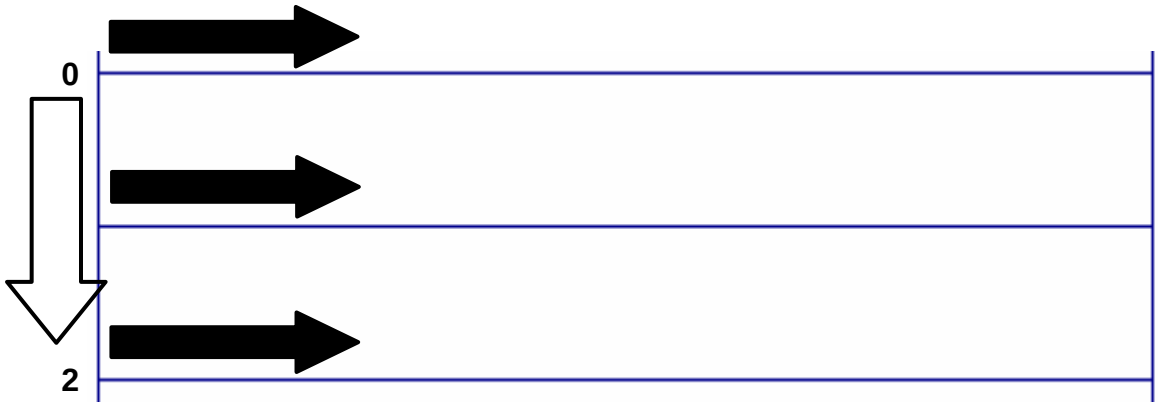
Ladder Logic Programming

The ladder logic programming language is probably the most commonly used in industry. It is very similar to an electrical wiring diagram in appearance. Within controllogix, it is programmed as a **routine.**

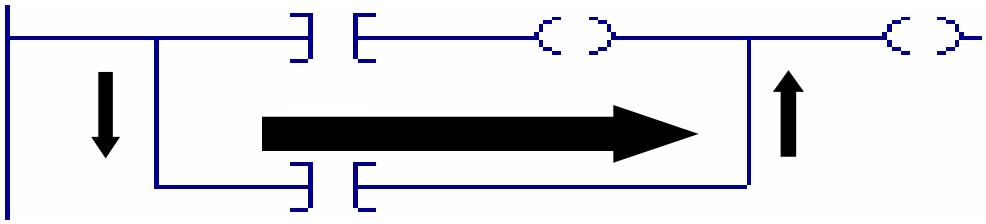


Branch

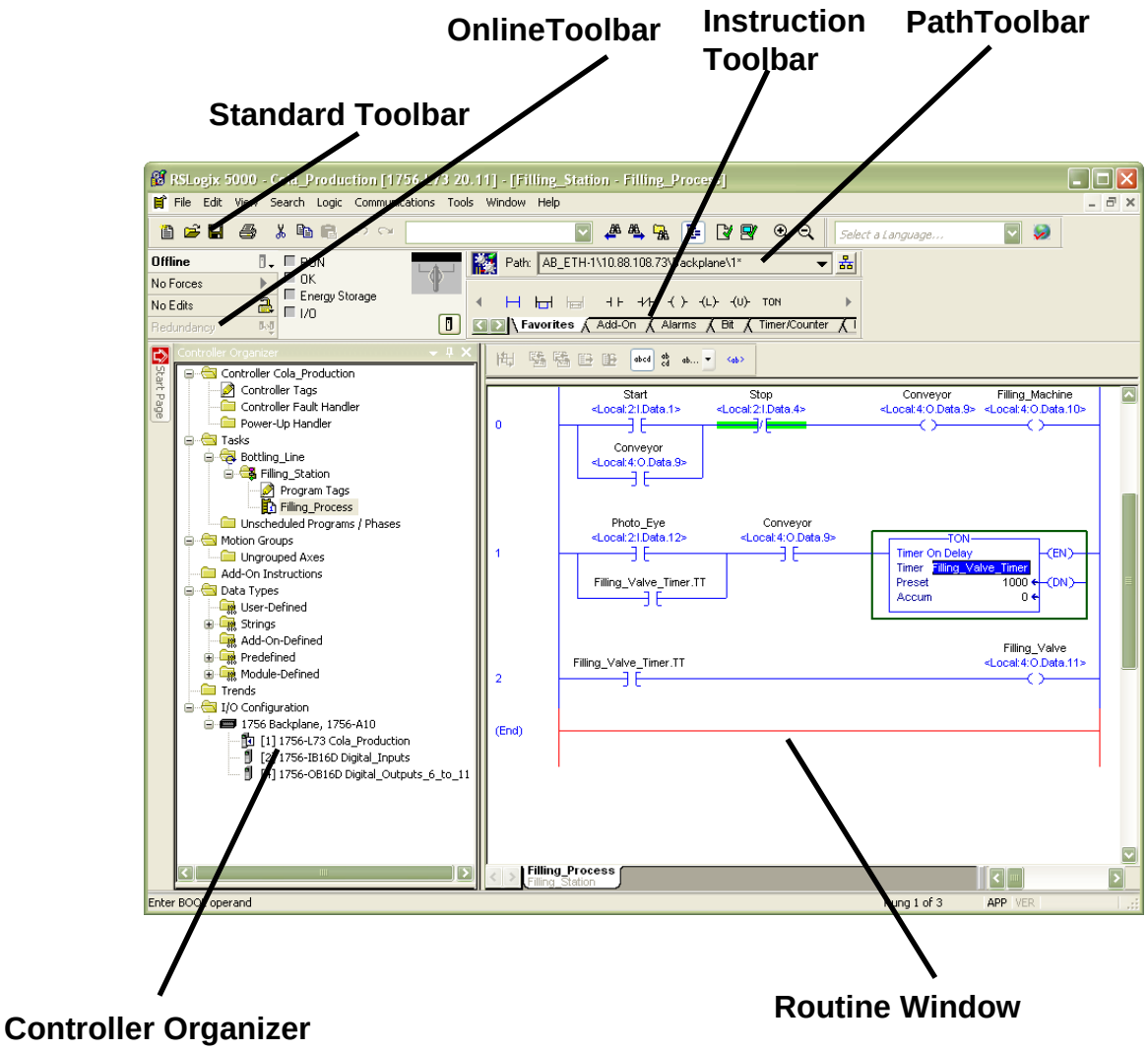
The instructions on the left are **Input Instructions** and the ones on the right are **Output Instructions.**



The **Program Flow** of a routine is always left to right, top to bottom. The last rung is an End rung and acts as the return There is a very unique feature with controllogix where you can interlace inputs and outputs in series on the same rung as can be seen below. The last instruction must **always** be an output instruction.



The rungs shown above, are an example of a controller in **Run Mode and Online** where the power rails are green. When the instructions are green, it shows that this rung is **True**. All instructions on a rung must be true for continuity to exist The first rung shows full continuity. The second rung shows one condition true, but all conditions are not true so full continuity does not exist.



The illustration above shows the **Project Window** with the various toolbars and organizers.

Controller Organizer. This is where all the tools reside for creating and maintaining a project.

Routine Window. All routines will be programmed here in any of the four available languages.

Online Toolbar. This shows the controller status when Online and any errors that may be present.

Standard Toolbar. This contains all the standard features such as cut, copy paste etc.

Instruction Toolbar. All usable instructions for a particular programming language are found here. They are categorized with named tabs for ease of access.

Path Toolbar. This shows the name of the driver used for communication. A new project will have “None” in here. Only when the correct driver is configured will it be possible to go online to the controller