

As with some things in ServiceNow, the Workflow Coordinator that is used by the Parallel Flow Launcher is not well documented. Below you can see a list of all methods which can be used with the Workflow Coordinator.

- `add(inputs, workflow)` – initializes a new workflow with inputs to be launched. Input parameters: `inputs` - input parameters for workflow, `workflow [optional]` - `sys_id` or name of workflow.
- `save(nameOrId, depot)` - serializes the data for this object into the running workflow. Input parameters: `nameOrId` - `sys_id` or name of workflow.
- `load(nameOrId, depot)` - deserializes an instance of this object. Input parameters: `nameOrId` - `sys_id` or name of workflow.
- `start(executing)` - starts executing group of flows (number specified by `poolsize`). It waits until all flows will be executed. Input parameter: `executing` - executing activity.
- `cancel(reason)` - cancels all subflows. Input parameter: `reason` - type: string, information about reason of cancelling.
- `getRunningCount()` - returns number of currently running flows.
- `isRunning(index)` - checks if subflow with specified index is still running.
- `isFinished(index)` - checks if subflow with specified index has been executed.
- `isCancelled()` - returns true if the workflow coordinator was cancelled.
- `hasNext()` - returns true if some subflows are awaiting to be executed.
- `getFlow(index)` - returns flow if exists.
- `setMax(max)` - specifies the maximum number of parallel workflows this activity is expected to launch.
- `setPoolsize(poolsize)` - it specifies how many subflows will run at one time.
- `setWorkflow(workflow)` - set workflow for WorkflowCoordinator (default setting for all workflows).
- `findFlow(contextId)` - returns index of flow if exists (based on `sys_id` of workflow context [`contextId`]).
- `getNumFlows()` - returns the total number of flows to be launched.
- `onFinish(ev)` - sets the result status of a subflow that finished. Input parameter: `ev` - event which stores the information: context, status, isComplete, result.
- `getError()` - returns the error message if any error occurred.

To see an example of using the Workflow Coordinator, please read [this article](#)