

Forum:[Hack Forums](#)

Thread: [Have status in a datagridview](#)

Takehito Morokuzu - June 7, 2019, 2:49 pm

Hi,
I've tried searching for this issue but I cannot form my question into words lol

Basically, I have a datagridview.
the first column will be a name, for example test1.
the second column will be a status, in this case running, or not running.
I am going to execute a program from my program, it will add a name to the datagridview and while it is running the status will say running. when the program closes I want the status column to change to not running.
I also need multiple of these, for example 5 programs running at once.

Any help will be appreciated :)

Vyxn - June 9, 2019, 12:22 am

Run your target process, check the process if running. That's all.

Hawk HF - June 9, 2019, 3:51 pm

It's in C#, but you should be able to make sense of it. I'm sure there's a few unhandled edge cases, but I'll leave those for you to figure out. There's plenty of room for expansion as well.
Good luck on your project. ?

Launch code:

```
Code private void LaunchProcess(string displayName, string path, params string[] args)
```

```
{  
    // create new process and allowed event raising  
    var process = new Process()  
    {  
        EnableRaisingEvents = true  
    };  
  
    // set file path  
    process.StartInfo.FileName = path;  
  
    // create process data object to store display name and process reference  
    var processData = new ProcessData(displayName, process);  
  
    // attempt to start the process  
    var started = process.Start();  
  
    // add process to DataGridView with the proper started status  
    dataGridView1.Rows.Add(processData, started ? "Running" : "Stopped");  
  
    // when the process is closed, find its ProcessData object in the DataGridView and update its  
    status  
    process.Exited += (s, e) =>
```

```

{
    foreach (DataGridViewRow row in dataGridView1.Rows)
    {
        if (row.Cells[0].Value == processData)
        {
            row.Cells[1].Value = "Stopped";
            break;
        }
    }
};
}

```

Simple wrapper class to store display name and process:

Code public class ProcessData

```

{
    public string DisplayName { get; }

    public Process Process { get; }

    public ProcessData(string displayName, Process process)
    {
        DisplayName = displayName;
        Process = process;
    }

    public override string ToString()
    {
        return DisplayName;
    }
}

```

Usage:

Code LaunchProcess("Notepad", "notepad.exe");