Coding Exercise

Many systems require recurring processing of tasks. Your job is to build a task runner that supports the following requirements:

1. The task runner should run a configured task when it is ready to be run.
    a. A task can be scheduled to run only after other tasks have been run first. Since these are recurring tasks, the successful run of the dependencies should be found within the current day.
    b. Tasks will have a configured maximum duration. If the task does not run during that period, then it should be considered to have failed.
2. The task runner should report on the successful completion or the failure of any task. If the task failed, the task runner should report on any exception information that is available from the task.
3. A task that does not complete is considered a failed task. A task may not complete because it takes too long, or it may simply hang and become unresponsive.
4. While tasks will have built-in defaults, the task runner should pass configured overrides to a task when it is run. These overrides can be a set of keys and values.
5. It should be possible to run more than one instance of the task runner at one time, and they should not conflict.

An example scenario:

ID	Task Name	Maximum duration	Dependencies
1	Rent Charges	20 minutes	
2	Rent Payments	20 minutes	1
3	Late Fees	60 minutes	2
4	Bill customers for EFT payments	30 minutes	2

The solution should have the following characteristics:

1. The code should be primarily written in C# and it should run on a 64-bit Windows machine with .NET 4.0.
2. The code should be run through the Windows task scheduler.
3. If you need to simplify the problem, be prepared to discuss how to address the remaining portion of the exercise. For instance, you may choose to create a stub implementation of the code that would keep track of what tasks have run and whether they were successful. You might also have a stub for the code that would communicate the success or failure of a task.

The exercise will be evaluated based on the following:

1. Does the task runner satisfy all of the requirements listed?
2. How does one know that the task runner will work correctly?

3. Is the solution clear?
4. Are different responsibilities in the solution abstracted reasonably?
5. Does the solution produce the right list of tasks to run under different scenarios?