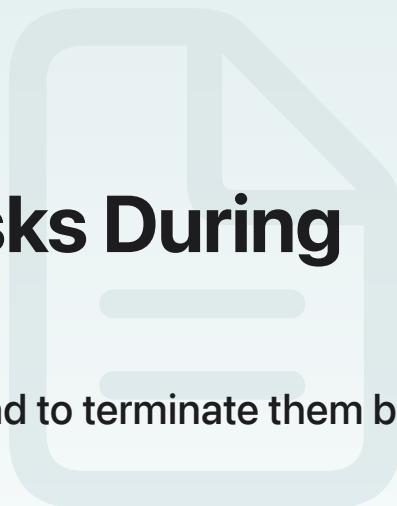


[Background Tasks](#) / Starting and Terminating Tasks During Development

Article

Starting and Terminating Tasks During Development

Use the debugger during development to start tasks and to terminate them before completion.



Overview

The delay between the time you schedule a background task and when the system launches your app to run the task can be many hours. While developing your app, you can use two private functions to start a task and to force early termination of the task according to your selected timeline. The debug functions work only on devices.

Important

Use private functions only during development. Including a reference to these functions in apps submitted to the App Store is cause for rejection.

Launch a Task

To launch a task:

1. Set a breakpoint in the code that executes after a successful call to `submit(_:_:)`.
2. Run your app on a device until the breakpoint pauses your app.
3. In the debugger, execute the line shown below, substituting the identifier of the desired task for `TASK_IDENTIFIER`.
4. Resume your app. The system calls the launch handler for the desired task.

```
e -l objc -- (void)=[[BGTaskScheduler sharedScheduler] _simulateLaunchForTaskWithIder
```

Force Early Termination of a Task

To force termination of a task:

1. Set a breakpoint in the desired task.
2. Launch the task using the debugger as described in the previous section.
3. Wait for your app to pause at the breakpoint.
4. In the debugger, execute the line shown below, substituting the identifier of the desired task for `TASK_IDENTIFIER`.
5. Resume your app. The system calls the expiration handler for the desired task.

```
e -l objc -- (void)=[[BGTaskScheduler sharedScheduler] _simulateExpirationForTaskWith
```