[Python3, Tkinter - display message after process completion](https://stackoverflow.com/questions/44637473/python3-tkinter-display-message-after-process-completion)

I'm using python 3 and tkinter. I have a simple GUI that triggers a back-end process (in a different class, instantiated in the GUI class). I want to display a message to the user once the back-end process is complete, i.e. use messagebox (or anything else, doesn't really matter) but only after the process is complete. How can I do that?

Thanks in advance

If the process blocks the main loop while it is executed, you can just create a method in your main class that runs the process and shows the messagebox:

import tkinter as tk

from tkinter import messagebox

import time

class BackendProcess:

def \_\_init\_\_(self):

self.finished = False

def run(self):

time.sleep(10)

self.finished = True

class GUI(tk.Tk):

def \_\_init\_\_(self):

tk.Tk.\_\_init\_\_(self)

self.process = BackendProcess()

tk.Button(self, text='Run', command=self.run).pack()

def run(self):

self.process.run()

messagebox.showinfo('Info', 'Process completed!')

if \_\_name\_\_ == '\_\_main\_\_':

gui = GUI()

gui.mainloop()

If the process does not block the main loop, then the above method does not work and the messagebox is displayed right after the start of the process. To avoid this, the after method can be used to periodically check the finished flag of the process:

import tkinter as tk

from tkinter import messagebox

import time

import threading

class BackendProcess:

def \_\_init\_\_(self):

self.finished = False

def task(self):

time.sleep(2)

print('finished')

self.finished = True

def run(self):

thread = threading.Thread(target=self.task, daemon=True)

thread.start()

class GUI(tk.Tk):

def \_\_init\_\_(self):

tk.Tk.\_\_init\_\_(self)

self.process = BackendProcess()

tk.Button(self, text='Run', command=self.run).pack()

def run(self):

self.process.run()

self.check\_process()

def check\_process(self):

""" Check every 1000 ms whether the process is finished """

if self.process.finished:

messagebox.showinfo('Info', 'Process completed')

else:

self.after(1000, self.check\_process)

if \_\_name\_\_ == '\_\_main\_\_':

gui = GUI()

gui.mainloop()

From the comments, I think the first method should work since the process is executed in the main thread. I have shown the second case anyway because completion notification are especially useful for time-consuming tasks that are often executed in a separated thread to avoid freezing the GUI.