# Scheduling R Scripts on the Epi Server

Overview of using the epi server to schedule scripts.

The server is basically a virtual computer that is always on. From there, we can run scripts, make R Markdown documents, etc.

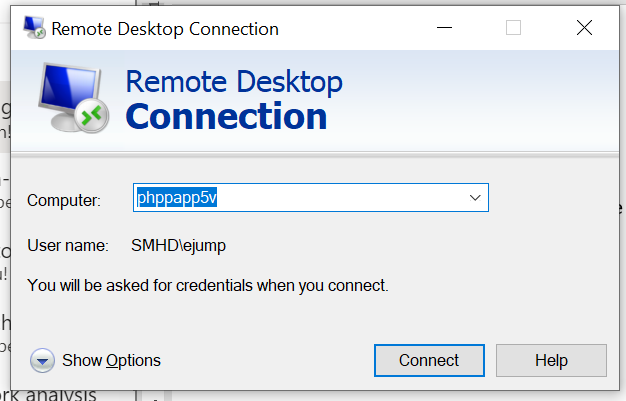
## Updating R scripts so they can be scheduled:

For the script to run when you’re not logged on, you need to do a couple things:

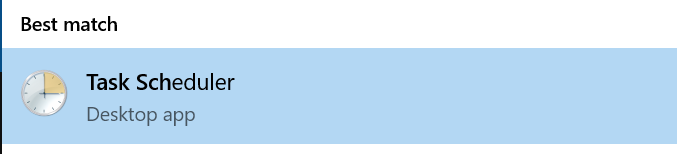
1. Make sure the script doesn’t reference any drives by letter. Instead of using: J:/path, you should use //SVPHFS/path (the path is not case sensitive).
2. Make sure all the packages you are using in the script are library-ed at the beginning of the script (just like you need to do with a markdown)
3. Make sure your script runs without errors. Even if the error isn’t important to the script (ex: you accidentally reference a dataset that doesn’t exist and doesn’t affect the code after it), you need to remove it. The scheduler will stop running the R script as soon as it encounters an error.

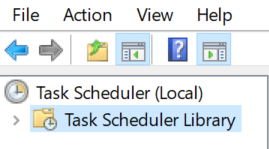
## Scheduling R scripts to run:

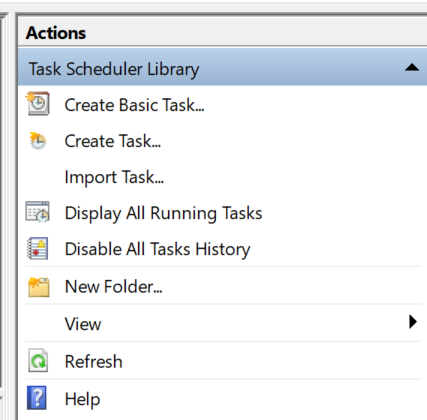
1. Connect to the epi server using the Remote Desktop application. The server name is: **phppapp5v**. Like any remote desktop, you need to be on be on the county network (either in office or on the VPN).



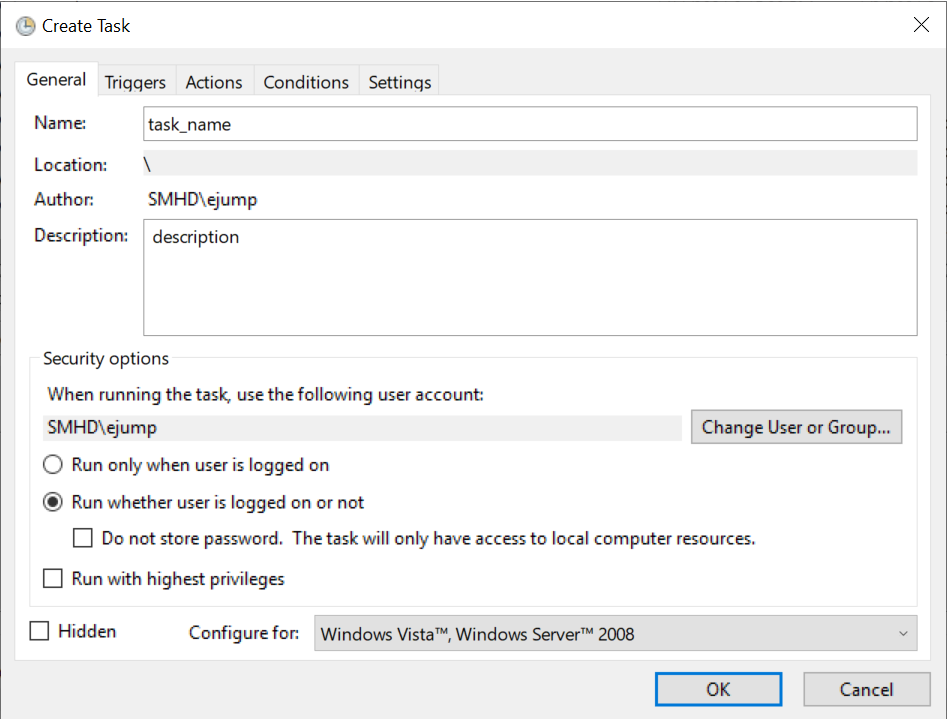
1. The application should launch you into a computer that looks like your desktop, except that it doesn’t have Teams or Outlook. Go to the search bar and look for the **Task Scheduler** application.

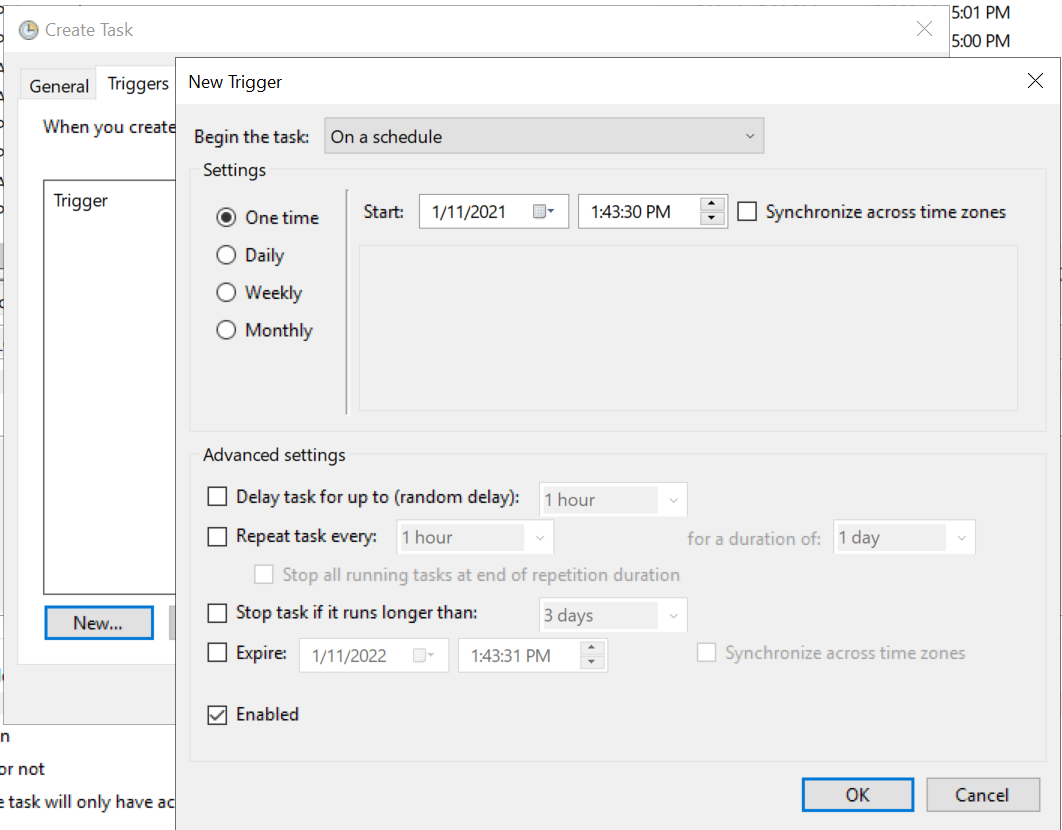


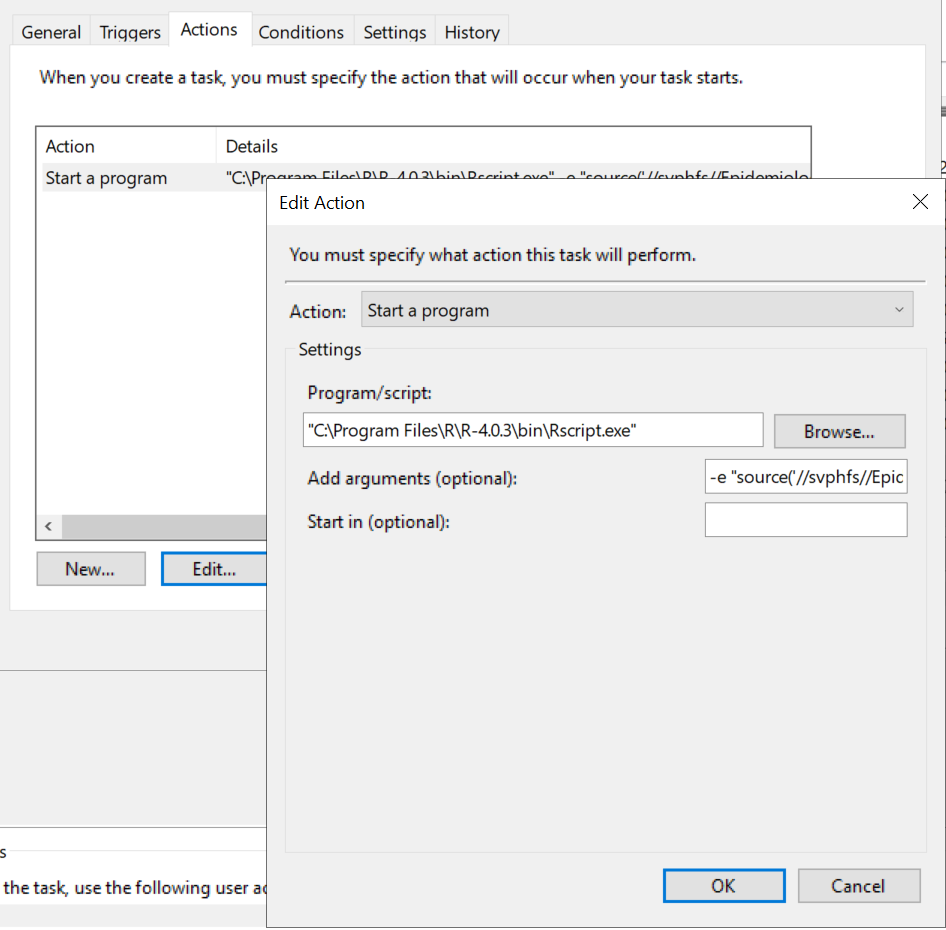
1. In the task scheduler, click on the **Task Scheduler Library**:
2. Then on the right-hand side, select the option to **Create Task…**:



1. In the **General** tab, you’ll need to put in a name and select that the task should **Run whether user is logged on or not**. Add a description so others know what the script does.



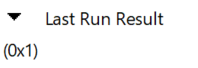
1. In the **Triggers** tab, select **New** and then choose when and how often the task should run:
2. In the **Actions** tab, select **New**. In the **Edit Action** tab, you’ll need to fill in the **Program/script** and **Add arguments (optional):** fields.
   1. **Program/script:** "C:\Program Files\R\R-X.X.X\bin\Rscript.exe"
      1. R-X.X.X: This should be the newest version of R that exists on the server.
   2. **Add arguments (optional): -**e "source('//svphfs//path//Code Repository//R Scripts//<PATH>.R')"
      1. The path needs to be EXACTLY where the script lives. Triple check that you don’t have extra spaces, characters, etc.



## Keep in mind:

1. When you change your password, you also need to re-save your tasks with your new password. Windows uses your authentication to run the tasks so it needs an updated password. You should change something in the task (ex: change the time from 6:09 to 6:10) so you will be prompted to re-enter your password
2. If you’re using ESSENCE APIs to schedule your scripts, make sure you change your password in the keyring or RNSSP package when you change it in Biosense.

## Troubleshooting steps/ideas:

If your task isn’t working, when you try to run it, you’ll get a Last Run Result of “(0x1)”:

When this happens, here are some steps you can take to troubleshoot.

1. Check that the scripts will run on the server
   * Copy all the code from the R script and run it in R on the server.
     + You can access the R application from Windows Explorer > Local Disk (C:): > Program Files > R > R-4.X.X (the most current one) > bin > R (not Rscript)
   * See if the script runs all the way through or if it errors out. If there is an error, review it, fix it, and then try it again.
   * Once the script is running in R on the server, try triggering the task again. If the task works, yay! If not, continue on.
2. Check the task file path
   * Once you know that the script runs on the server, you know the issue must be somewhere in the task. The issue is probably in the file path in the task
   * Open the task, go the Actions tab, edit the action and copy the file path from the **Add Arguments (optional):** field into a text editor of your choice (I usually do it in an email draft)
   * Review the text. The format should be:
     + e "source('path.R')"
     + the actual path should:
       - point to the folder using the actual path not letters (ex: [\\svphfs\epidemiology](file:///\\svphfs\epidemiology) vs. J:\)
       - point exactly to the file!!
       - not have extra spaces.
         * NOT OK: `svphfs //path`
         * OK: `svphfs//path
       - Make sure the file path has the .R or .Rmd at the end
   * Once you’re sure the path is correct, try running the script again. If the task doesn’t work, continue on:
3. If you’re sure the script works and are sure the task is correct, try scheduling a tiny task and running it. Ex: write a script save a string as a data frame only using base R functions.
   * Example: string <- c("hello", "world"), string\_df <- data.frame(string), write.csv(string\_df, “path”)
   * See if that works.
   * If it works, then the issue is still with your script, feel free to reach out to Beth to troubleshoot at this point.

## Maintenance

* When you change your password, go into the task scheduler and change your password for each task