

Schtasks

Archived: 2026-04-05 17:02:40 UTC

Schedules commands and programs to run periodically or at a specific time. Adds and removes tasks from the schedule, starts and stops tasks on demand, and displays and changes scheduled tasks.

To view the command syntax, click the following command:

```
schtasks create
```

Creates a new scheduled task.

Syntax

```
schtasks /create /tn TaskName /tr TaskRun /sc schedule [/mo modifier] [/d day] [/m month[,month...]] [/i IdleTime] [/st StartTime] [/sd StartDate] [/ed  
/u [domain\]user /p password]] [/ru {[Domain\]User | "System"} [/rp Password]] /?
```

Parameters

/tn TaskName : Specifies a name for the task.

/tr TaskRun : Specifies the program or command that the task runs. Type the fully qualified path and file name of an executable file, script file, or batch file. If you omit the path, SchTasks.exe assumes that the file is in the Systemroot\System32 directory.

/sc schedule : Specifies the schedule type. Valid values are MINUTE, HOURLY, DAILY, WEEKLY, MONTHLY, ONCE, ONSTART, ONLOGON, ONIDLE.

Value	Description
MINUTE, HOURLY, DAILY, WEEKLY, MONTHLY	Specifies the time unit for the schedule.
ONCE	The task runs once at a specified date and time.
ONSTART	The task runs every time the system starts. You can specify a start date, or run the task the next time the system starts.
ONLOGON	The task runs whenever a user (any user) logs on. You can specify a date, or run the task the next time the user logs on.
ONIDLE	The task runs whenever the system is idle for a specified period of time. You can specify a date, or run the task the next time the system is idle.

/mo modifier : Specifies how often the task runs within its schedule type. This parameter is required for a MONTHLY schedule. This parameter is valid, but optional, for a MINUTE, HOURLY, DAILY, or WEEKLY schedule. The default value is 1.

Schedule type	Modifier	Description
MINUTE	1 - 1439	The task runs every <i>n</i> minutes.

Schedule type	Modifier	Description
HOURLY	1 - 23	The task runs every <i>n</i> hours.
DAILY	1 - 365	The task runs every <i>n</i> days.
WEEKLY	1 - 52	The task runs every <i>n</i> weeks.
MONTHLY	1 - 12	The task runs every <i>n</i> months.
LASTDAY		The task runs on the last day of the month.
FIRST, SECOND, THIRD, FOURTH, LAST		Use with the <i>/d day</i> parameter to run a task on a particular week and day. For example, on the third Wednesday of the month.

/d day : Specifies a day of the week or a day of a month. Valid only with a WEEKLY or MONTHLY schedule.

Schedule type	Day values
WEEKLY	Optional. Valid values are MON - SUN and * (every day). MON is the default.
MONTHLY	A value of MON - SUN is required when the FIRST, SECOND, THIRD, FOURTH, or LAST modifier (<i>/mo</i>) is used. A value of 1 - 31 is optional and is valid only with no modifier or a modifier of the 1 - 12 type. The default is day 1 (the first day of the month).

/m month [, month ...] : Specifies a month of the year. Valid values are JAN - DEC and * (every month). The **/m** parameter is valid only with a MONTHLY schedule. It is required when the LASTDAY modifier is used. Otherwise, it is optional and the default value is * (every month).

/i IdleTime : Specifies how many minutes the computer is idle before the task starts. Type a whole number from 1 to 999. This parameter is valid only with an ONIDLE schedule, and then it is required.

/st StartTime : Specifies the time of day that the task starts in HH:MM:SS 24-hour format. The default value is the current local time when the command completes. The **/st** parameter is valid with MINUTE, HOURLY, DAILY, WEEKLY, MONTHLY, and ONCE schedules. It is required with a ONCE schedule.

/sd StartDate : Specifies the date that the task starts in MM/DD/YYYY format. The default value is the current date. The **/sd** parameter is valid with all schedules, and is required for a ONCE schedule.

/ed EndDate : Specifies the last date that the task is scheduled to run. This parameter is optional. It is not valid in a ONCE, ONSTART, ONLOGON, or ONIDLE schedule. By default, schedules have no ending date.

/s Computer : Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [domain \] user : Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p password : Specifies the password of the user account specified in the **/u** parameter. This parameter is required when the **/u** parameter is used.

/ru { [Domain \] User | "System" } : Runs the tasks with the permission of the specified user account. By default, the task runs with the permissions of the user logged on to the computer running SchTasks.

Value	Description
[Domain\]User	Specifies a user account.
"System" or ""	Specifies the NT Authority\System account, which is used by the operating system.

/rp Password : Specifies the password of the user account that is specified in the **/ru** parameter. If you omit this parameter when specifying a user account, SchTasks.exe prompts you for the password and obscures the text you type. Tasks run with permissions of the NT Authority\System account do not require a password and SchTasks.exe does not prompt for one.

/? : Displays help at the command prompt.

- Using the **/s**, **/u**, and **/p** command-line options

The **/u** and **/p** command-line options are available only when you use **/s**. You must use **/p** with **/u** to provide the user's password.

- SchTasks.exe does not verify program file locations or user account passwords. If you do not enter the correct file location or the correct password for the user account, the task is created, but it does not run. Also, if the password for an account changes or expires, and you do not change the password saved in the task, then the task does not run.
- The NT Authority\System account does not have interactive logon rights. Users do not see and cannot interact with programs run with system permissions.
- Each task runs only one program. However, you can create a batch file that starts multiple tasks, and then schedule a task that runs the batch file.
- You can test a task as soon as you create it. Use the **run** operation to test the task and then check the SchedLgU.txt file (Systemroot\SchedLgU.txt) for errors.

Syntax and examples for each schedule type

To view the command syntax, click a command:

schtasks create minute

Syntax

```
schtasks /create /tn TaskName /tr TaskRun /sc minute [/mo {1 - 1439}] [/st StartTime] [/sd StartDate] [/ed EndDate] [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User | "System"} [/rp Password]]
```

Example

To schedule a task to run every 20 minutes

The following command schedules a security script, Sec.vbs, to run every 20 minutes. Because the command does not include a starting date or time, the task starts 20 minutes after the command completes, and runs every 20 minutes thereafter whenever the system is running. Notice that the security script source file is located on a remote computer, but that the task is scheduled and executes on the local computer.

```
schtasks /create /sc minute /mo 20 /tn "Security Script" /tr \\central\data\scripts\sec.vbs
```

In response, SchTasks.exe displays a message explaining that the task will run with the permissions of the current user and requests the current user's password. When you enter the password, SchTasks.exe obscures the text you type.

The task will be created under current logged-in user name.
Please enter the password

Then, SchTasks.exe displays a message indicating that the task is scheduled:

SUCCESS: The Scheduled Task "Security Script" has successfully been created.

A query shows the task that the command scheduled:

TaskName	Next Run Time	Status
Security Script	10:50:00 AM , 4/4/2001	

schtasks create hourly

Syntax

```
schtasks /create /tn TaskName /tr TaskRun /sc hourly [/mo {1 - 365}] [/st StartTime] [/sd StartDate] [/ed EndDate] [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User | "System"} [/rp Password]]
```

Example

To schedule a command that runs every hour at five minutes past the hour

The following command schedules the MyApp program to run hourly beginning at five minutes past midnight. Because the /mo parameter is omitted, the command uses the default value for the hourly schedule, which is every (1) hour. If this command is issued after 12:05 A.M., the program will not run until the next day.

```
schtasks /create /sc hourly /st 00:05:00 /tn "My App" /tr c:\apps\myapp.exe
```

To schedule a command that runs every five hours

The following command schedules the MyApp program to run every five hours beginning on the first day of March 2001. It uses the /mo parameter to specify the interval and the /sd parameter to specify the start date. Because the command does not specify a start time, the current time is used as the start time.

```
schtasks /create /sc hourly /mo 5 /sd 03/01/2001 /tn "My App" /tr c:\apps\myapp.exe
```

schtasks create daily

Syntax

```
schtasks /create /tn TaskName /tr TaskRun /sc daily [/mo {1 - 365}] [/st StartTime] [/sd StartDate] [/ed EndDate] [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User | "System"} [/rp Password]]
```

Example

To schedule a task that runs every day

The following example schedules the MyApp program to run once a day, every day, at 8:00 A.M. until December 31, 2001. Because it omits the /mo parameter, the default interval of 1 is used to run the command every day.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc daily /st 08:00:00 /ed 12/31/2001
```

To schedule a task that runs every other day

The following example schedules the MyApp program to run every other day at 1:00 P.M. (13:00) beginning on December 31, 2001. The command uses the /mo parameter to specify an interval of two (2) days.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc daily /mo 2 /st 13:00:00 /sd 12/31/2001
```

schtasks create weekly

Syntax

```
schtasks /create /tn TaskName /tr TaskRun /sc weekly [/d {MON - SUN | *}] [/mo {1 - 52}] [/st StartTime] [/sd StartDate] [/ed EndDate] [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User |
```

"System" [/rp Password]]

Example

To schedule a task that runs every six weeks

The following command schedules the MyApp program to run on a remote computer every six weeks. The command uses the /mo parameter to specify the interval. It also uses the /s parameter to specify the remote computer and the /ru parameter to schedule the task to run with the permissions of the user's Administrator account. Because the /rp parameter is omitted, SchTasks.exe prompts the user for the Administrator account password.

Also, because the command is run remotely, all paths in the command, including the path to MyApp.exe, refer to paths on the remote computer.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc weekly /mo 6 /s Server16 /ru Admin01
```

To schedule a task that runs every other week on Friday

The following command schedules a task to run every other Friday. It uses the /mo parameter to specify the two-week interval and the /d parameter to specify the day of the week. To schedule a task that runs every Friday, omit the /mo parameter or set it to 1.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc weekly /mo 2 /d FRI
```

```
schtasks create monthly
```

Syntax

General Monthly Schedule Syntax : `schtasks /create /tn TaskName /tr TaskRun /sc monthly [/mo {FIRST | SECOND | THIRD | FOURTH | LAST | LASTDAY} [/d {MON - SUN | 1 - 31}] [/m {JAN - DEC[,JAN - DEC...]} | *] [/st StartTime] [/sd StartDate] [/ed EndDate] [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User | "System"} [/rp Password]]`

Specific Week Syntax : `schtasks /create /tn TaskName /tr TaskRun /sc monthly /mo {FIRST | SECOND | THIRD | FOURTH | LAST} /d {MON - SUN} [/m {JAN - DEC[,JAN - DEC...]} | *] [/st StartTime] [/sd StartDate] [/ed EndDate] [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User | "System"} [/rp Password]]`

Lastday Syntax : `schtasks /create /tn TaskName /tr TaskRun /sc monthly /mo LASTDAY /m {JAN - DEC[,JAN - DEC...]} | *] [/st StartTime] [/sd StartDate] [/ed EndDate] [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User | "System"} [/rp Password]]`

Specific Date Syntax : `schtasks /create /tn TaskName /tr TaskRun /sc monthly /d {1 - 31} [/m {JAN - DEC[,JAN - DEC...]} | *] [/st StartTime] [/sd StartDate] [/ed EndDate] [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User | "System"} [/rp Password]]`

Examples

To schedule a task for the first day of every month

The following command schedules the MyApp program to run on the first day of every month. Because the default modifier is none (no modifier), the default day is day 1, and the default month is every month, the command does not need any additional parameters.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly
```

To schedule a task for the last day of every month

The following command schedules the MyApp program to run on the last day of every month. It uses the /mo parameter to specify the last day of the month and the /m parameter with the wildcard character (*) to indicate that the program runs on the last day of every month.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly /mo lastday /m *
```

To schedule a task that runs every three months

The following command schedules the MyApp program to run every three months. It uses the /mo parameter to specify the interval.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly /mo 3
```

To schedule a task for the second Sunday of every month

The following command schedules the MyApp program to run on the second Sunday of every month. It uses the **/mo** parameter to specify the second week of the month and the **/d** parameter to specify the day.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly /mo SECOND /d SUN
```

To schedule a task for the 15th days of May and June

The following command schedules the MyApp program to run on May 15 and June 15 at 3:00 PM (15:00). It uses the **/d** parameter to specify the date and the **/m** parameter to specify the months. It also uses the **/st** parameter to specify the start time.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly /d 15 /m MAY,JUN /st 15:00:00
```

```
schtasks create once
```

Syntax

```
schtasks /create /tn TaskName /tr TaskRun /sc once /st StartTime /sd StartDate [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User | "System"} [/rp Password]]
```

Example

To schedule a task that runs one time

The following command schedules the MyApp program to run at midnight on January 1, 2002. It uses the **/ru** parameter to run the task with the permissions of the user's Administrator account and the **/rp** parameter to provide the password for the Administrator account.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc once /st 00:00:00 /sd 01/01/2002 /ru Admin23 /rp p@sswordD1
```

```
schtasks create onstart
```

Syntax

```
schtasks /create /tn TaskName /tr TaskRun /sc onstart [/sd StartDate] [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User | "System"} [/rp Password]]
```

Example

To schedule a task that runs every time the system starts

The following command schedules the MyApp program to run every time the system starts, beginning on March 15, 2001:

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc onstart /sd 03/15/2001
```

```
schtasks create onlogon
```

Syntax

```
schtasks /create /tn TaskName /tr TaskRun /sc onlogon [/sd StartDate] [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User | "System"} [/rp Password]]
```

Example

To schedule a task that runs when a user logs on to a remote computer

The following command schedules a batch file to run every time a user (any user) logs on to the remote computer. It uses the **/s** parameter to specify the remote computer. Because the command is remote, all paths in the command, including the path to the batch file, refer to a path on the remote computer.

```
schtasks /create /tn "Start Web Site" /tr c:\myiis\webstart.bat /sc onlogon /s Server23
```

```
schtasks create onidle
```

Syntax

```
schtasks /create /tn TaskName /tr TaskRun /sc onidle /iIdleTime [/sd StartDate] [/s computer [/u  
[domain\user /p password]] [/ru {[Domain]User | "System"} [/rp Password]]
```

Example

To schedule a task that runs whenever the computer is idle

The following command schedules the MyApp program to run whenever the computer is idle. It uses the required /i parameter to specify that the computer must remain idle for ten minutes before the task starts.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc onidle /i 10
```

More examples

To view the example, click the example name:

To create a task that runs with System permissions

The following command schedules the MyApp program to run with permissions of the NT Authority\System account. In this example, the task is scheduled to run on the first day of every month, but you can use any schedule type for a task run with system permissions.

The command uses the /ru "System" parameter to specify the system security context. Because system tasks do not use a password, the /rp parameter is omitted.

```
schtasks /create /tn "My App" /tr c:\apps\myapp.exe /sc monthly /d 1 /ru "System"
```

In response, SchTasks.exe displays an informational message and a success message. It does not prompt for a password.

```
INFO: The task will be created under user name ("NT AUTHORITY\SYSTEM").  
SUCCESS: The Scheduled task "My App" has successfully been created.
```

To create a task that runs more than one program

Each task runs only one program. However, you can create a batch file that runs multiple programs and then schedule a task to run the batch file. The following procedure demonstrates this method:

1. Create a batch file that starts the programs you want to run.

In this example, you create a batch file that starts Event Viewer (Eventvwr.exe) and System Monitor (Perfmon.exe).

- Open a text editor, such as Notepad.
- Type the name and fully qualified path to the executable file for each program. In this case, the file includes the following statements.

```
C:\Windows\System32\Eventvwr.exe
```

```
C:\Windows\System32\Perfmon.exe
```

- Save the file as MyApps.bat.

2. Use SchTasks.exe to create a task that runs MyApps.bat.

The following command creates the Monitor task, which runs whenever anyone logs on. It uses the /tn parameter to name the task, and the /tr parameter to run MyApps.bat. It uses the /sc parameter to indicate the OnLogon schedule type and the /ru parameter to specify the Administrator account.

```
schtasks /create /tn Monitor /tr C:\MyApps.bat /sc onlogon /ru Reskit\Administrator
```

As a result of this command, whenever a user logs on to the computer, the task starts both Event Viewer and System Monitor.

schtasks change

Changes one or more of the following properties of a task.

- The program that the task runs (/tr).
- The user account under which the task runs (/ru).
- The password for the user account (/rp).

Syntax

```
schtasks /change /tn TaskName [/s computer [/u [domain\]user /p password]] [/tr TaskRun] [/ru [Domain\]User | "System"] [/rp Password]
```

Parameters

/tn TaskName : Identifies the task to be changed. Enter the task name.

/s Computer : Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [domain \] user : Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p password : Specifies the password of the user account specified in the **/u** parameter. This parameter is required when the **/u** parameter is used.

/tr TaskRun : Changes the program that the task runs. Enter the fully qualified path and file name of an executable file, script file, or batch file. If you omit the path, SchTasks.exe assumes that the file is in the *Systemroot\System32* directory. The specified program replaces the original program run by the task.

/ru [Domain \] User | "System" : Changes the user account for the task.

Value	Description
[Domain\]User	Specifies a user account.
"System" or ""	Specifies the NT Authority\System account, which is used by the operating system.

When you change the user account, you must also change the user password. If a command has an **/ru** parameter but not an **/rp** parameter, SchTasks.exe prompts for a new password and obscures the text you type.

Tasks run with with permissions of the NT Authority\System account do not require a password and SchTasks.exe does not prompt for one.

/rp Password : Changes the account password for the task. Enter the new password.

/? : Displays help at the command prompt.

- Using the **/s**, **/u**, and **/p** command-line options

The **/u** and **/p** command-line options are available only when you use **/s**. You must use **/p** with **/u** to provide the user's password.

- The **/tn** and **/s** parameters identify the task. The **/tr**, **/ru**, and **/rp** parameters specify properties of the task that you can change.
- A command that uses the **change** operation must change at least one task property.
- The NT Authority\System account does not have interactive logon rights. Users do not see and cannot interact with programs run with system permissions.

Examples

To change the program that a task runs

The following command changes the program that the Virus Check task runs from VirusCheck.exe to VirusCheck2.exe. This command uses the **/tn** parameter to identify the task and the **/tr** parameter to specify the new program for the task. (You cannot change the task name.)

```
schtasks /change /tn "Virus Check" /tr C:\VirusCheck2.exe
```

In response, SchTasks.exe displays the following success message:

```
SUCCESS: The parameter of the Scheduled Task "Virus Check" has been changed.
```

As a result of this command, the Virus Check task now runs VirusCheck2.exe.

To change the password for a remote task

The following command changes the password of the user account for the RemindMe task on the remote computer, Svr01. The command uses the **/tn** parameter to identify the task and the **/s** parameter to specify the remote computer. It uses the **/rp** parameter to specify the new password, p@ssWord3.

This procedure is required whenever the password for a user account expires or changes. If the password saved in a task is no longer valid, then the task does not run.

```
schtasks /change /tn RemindMe /s Svr01 /rp p@ssWord3
```

In response, SchTasks.exe displays the following success message:

```
SUCCESS: The parameter of the Scheduled Task "RemindMe" has been changed.
```

As a result of this command, the RemindMe task now runs under its original user account, but with a new password.

To change the program and user account for a task

The following command changes the program that a task runs and changes the user account under which the task runs. Essentially, it uses an old schedule for a new task. This command changes the Notepad task, which starts Notepad.exe every morning at 9:00 a.m., to start Internet Explorer instead.

The command uses the **/tn** parameter to identify the task. It uses the **/tr** parameter to change the program that the task runs and the **/ru** parameter to change the user account under which the task runs.

The **/rp** parameter, which provides the password for the user account, is omitted. You must provide a password for the account, but you can use the **/rp** parameter and type the password in clear text, or wait for SchTasks.exe to prompt you for a password, and then enter the password in obscured text.

```
schtasks /change /tn Notepad /tr "c:\program files\Internet Explorer\iexplore.exe" /ru DomainX\Admin01
```

In response, SchTasks.exe requests the password for the user account. It obscures the text you type, so the password is not visible.

```
Please enter the password for DomainX\Admin01: *****
```

Note that the **/tn** parameter identifies the task and that the **/tr** and **/ru** parameters change the properties of the task. You cannot use another parameter to identify the task and you cannot change the task name.

In response, SchTasks.exe displays the following success message:

```
SUCCESS: The parameter of the Scheduled Task "Notepad" has been changed.
```

As a result of this command, the RemindMe task now runs under its original user account, but with a new password.

To change a program to the System account

The following command changes the SecurityScript task so that it runs with permissions of the NT Authority\System account. It uses the **/ru ""** parameter to indicate the System account.

```
schtasks /change /tn SecurityScript /ru ""
```

In response, SchTasks.exe displays the following success message:

```
SUCCESS: The parameter of the Scheduled Task "SecurityScript" has been changed.
```

Because tasks run with System account permissions do not require a password, SchTasks.exe does not prompt for one.

```
schtasks run
```

Starts a scheduled task immediately. The **run** operation ignores the schedule, but uses the program file location, user account, and password saved in the task to run the task immediately.

Syntax

```
schtasks /run /tn TaskName [/s computer [/u [domain\]user /p password]] /?
```

Parameters

/tn TaskName : Identifies the task. This parameter is required.

/s Computer : Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [domain\] user : Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p password : Specifies the password of the user account specified in the **/u** parameter. This parameter is required when the **/u** parameter is used.

/? : Displays help at the command prompt.

- Using the **/s**, **/u**, and **/p** command-line options

The **/u** and **/p** command-line options are available only when you use **/s**. You must use **/p** with **/u** to provide the user's password.

- Use this operation to test your tasks. If a task does not run, check the Task Scheduler Service transaction log, *Systemroot\SchedLgU.txt*, for errors.
- Running a task does not affect the task schedule and does not change the next run time scheduled for the task.
- To run a task remotely, the task must be scheduled on the remote computer. When you run it, the task runs only on the remote computer. To verify that a task is running on a remote computer, use Task Manager or the Task Scheduler transaction log, *Systemroot\SchedLgU.txt*.

Examples

To run a task on the local computer

The following command starts the "Security Script" task.

```
schtasks /run /tn "Security Script"
```

In response, SchTasks.exe starts the script associated with the task and displays the following message:

```
SUCCESS: The Scheduled Task "Security Script" is running .....
```

To run a task on a remote computer

The following command starts the Update task on a remote computer, Svr01:

```
schtasks /run /tn Update /s Svr01
```

In this case, SchTasks.exe displays the following error message:

```
ERROR: Unable to run the Scheduled Task "Update".
```

To find the cause of the error, look in the Scheduled Tasks transaction log, *C:\Windows\SchedLgU.txt* on Svr01. In this case, the following entry appears in the log:

```
"Update.job" (update.exe) 3/26/2001 1:15:46 PM ** ERROR **  
The attempt to log on to the account associated with the task failed, therefore, the task did not run.  
The specific error is:  
0x8007052e: Logon failure: unknown user name or bad password.  
Verify that the task's Run-as name and password are valid and try again.
```

Apparently, the user name or password in the task is not valid on the system. The following **schtasks /change** command updates the user name and password for the Update task on Svr01:

```
schtasks /change /tn Update /s Svr01 /ru Administrator /rp PassW@rd3
```

After the **change** command completes, the **run** command is repeated. This time, the Update.exe program starts and SchTasks.exe displays the following message:

```
SUCCESS: The Scheduled Task "Update" is running.....
```

```
schtasks end
```

Stops a program started by a task.

Syntax

```
schtasks /end /tn TaskName [/s computer [/u [domain\]user /p password]] /?
```

Parameters

/tn TaskName : Identifies the task that started the program. This parameter is required.

/s Computer : Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [domain\] user : Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p password : Specifies the password of the user account specified in the **/u** parameter. This parameter is required when the **/u** parameter is used.

/? : Displays help.

- Using the **/s**, **/u**, and **/p** command-line options

The **/u** and **/p** command-line options are available only when you use **/s**. You must use **/p** with **/u** to provide the user's password.

- SchTasks.exe ends only the instances of a program started by a scheduled task. To stop other processes, use TaskKill, a tool included in Windows XP Professional. For more information about TaskKill, see [Taskkill](#).

Examples

To end a task on a local computer

The following command stops the instance of Notepad.exe that was started by the My Notepad task:

```
schtasks /end /tn "My Notepad"
```

In response, SchTasks.exe stops the instance of Notepad.exe that the task started, and it displays the following success message:

```
SUCCESS: The Scheduled Task "My Notepad" has been terminated successfully.
```

To end a task on a remote computer

The following command stops the instance of Internet Explorer that was started by the InternetOn task on the remote computer, Svr01:

```
schtasks /end /tn InternetOn /s Svr01
```

In response, SchTasks.exe stops the instance of Internet Explorer that the task started, and it displays the following success message:

```
SUCCESS: The Scheduled Task "InternetOn" has been terminated successfully.
```

```
schtasks delete
```

Deletes a scheduled task.

Syntax

schtasks /delete /tn {TaskName | *} [/f] [/s computer [/u [domain\]user /p password]] [/?]

Parameters

/tn { TaskName | *} : Identifies the task being deleted. This parameter is required.

Value	Description
<i>TaskName</i>	Deletes the named task.
*	Deletes all scheduled tasks on the computer.

/f : Suppresses the confirmation message. The task is deleted without warning.

/s Computer : Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [domain \] user : Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p password : Specifies the password of the user account specified in the /u parameter. This parameter is required when the /u parameter is used.

/? : Displays help at the command prompt.

- Using the /s, /u, and /p command-line options

The /u and /p command-line options are available only when you use /s. You must use /p with /u to provide the user's password.

- The **delete** operation deletes the task from the schedule. It does not delete the program that the task runs or interrupt a running program.
- The **delete *** command deletes all tasks scheduled for the computer, not just the tasks scheduled by the current user.

Examples

To delete a task from the schedule of a remote computer

The following command deletes the "Start Mail" task from the schedule of a remote computer. It uses the /s parameter to identify the remote computer.

```
schtasks /delete /tn "Start Mail" /s Svr16
```

In response, SchTasks.exe displays the following confirmation message. To delete the task, type **y**. To cancel the command, type **n**:

```
WARNING: Are you sure you want to remove the task "Start Mail" (Y/N)? y
SUCCESS: The Scheduled Task "Start Mail" was successfully deleted.
```

To delete all tasks scheduled for the local computer

The following command deletes all tasks from the schedule of the local computer, including tasks scheduled by other users. It uses the /tn * parameter to represent all tasks on the computer and the /f parameter to suppress the confirmation message.

```
schtasks /delete /tn * /f
```

In response, SchTasks.exe displays the following success messages indicating that the only task scheduled, SecureScript, is deleted.

```
SUCCESS: The Scheduled Task "SecureScript" was successfully deleted.
```

schtasks query

Displays all tasks scheduled to run on the computer, including those scheduled by other users.

Syntax

schtasks [/query] [/fo {TABLE | LIST | CSV}] [/nh] [/v] [/s computer [/u [domain\]user /p password]]

Parameters

/query : The operation name is optional. Typing **schtasks** without any parameters performs a query.

/fo { TABLE | LIST | CSV } : Specifies the output format. **TABLE** is the default.

/nh : Omits column headings from the table display. This parameter is valid with the **TABLE** and **CSV** output formats.

/v : Adds advanced properties of the tasks to the display.

Queries using **/v** should be formatted as **LIST** or **CSV**.

/s Computer : Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.

/u [domain \] user : Runs the command with the permissions of the specified user account. By default, the command runs with the permissions of the user logged on to the computer running SchTasks.

/p password : Specifies the password of the user account specified in the **/u** parameter. This parameter is required when the **/u** parameter is used.

/? : Displays help at the command prompt.

- Using the **/s**, **/u**, and **/p** command-line options

The **/u** and **/p** command-line options are available only when you use **/s**. You must use **/p** with **/u** to provide the user's password.

- The **query** operation lists all tasks scheduled for the computer, not just the tasks scheduled by the current user.

Examples

To display the scheduled tasks on the local computer

The following commands display all tasks scheduled for the local computer. These commands produce the same result and can be used interchangeably.

schtasks

schtasks /query

In response, SchTasks.exe displays the tasks in the default, simple table format, as shown in the following table:

TaskName	Next Run Time	Status
Microsoft Outlook	At logon time	
SecureScript	14:42:00 PM , 2/4/2001	

To display advanced properties scheduled tasks

The following command requests a detailed display of the tasks on the local computer. It uses the **/v** parameter to request a detailed (verbose) display and the **/fo LIST** parameter to format the display as a list for easy reading. You can use this command to verify that a task you created has the intended recurrence pattern.

schtasks /query /fo LIST /v

In response, SchTasks.exe displays a detailed property list for all tasks. The following display shows the task list for a task scheduled to run at 4:00 A.M. on the last Friday of every month:

HostName:	RESKIT01
TaskName:	SecureScript
Next Run Time:	4:00:00 AM , 3/30/2001

Status:	Not yet run
Last Run Time:	Never
Last Result:	0
Creator:	user01
Schedule:	At 4:00 AM on the last Fri of every month, starting 3/24/2001
Task To Run:	C:\WINDOWS\system32\notepad.exe
Start In:	notepad.exe
Comment:	N/A
Scheduled Task State:	Enabled
Scheduled Type:	Monthly
Modifier:	Last FRIDAY
Start Time:	4:00:00 AM
Start Date:	3/24/2001
End Date:	N/A
Days:	FRIDAY
Months:	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
Run As User:	RESKIT\user01
Delete Task If Not Rescheduled:	Enabled
Stop Task If Runs X Hours and X Mins:	72:0
Repeat: Until Time:	Disabled
Repeat: Duration:	Disabled
Repeat: Stop If Still Running:	Disabled
Idle: Start Time(For IDLE Scheduled Type):	Disabled
Idle: Only Start If Idle for X Minutes:	Disabled
Idle: If Not Idle Retry For X Minutes:	Disabled
Idle: Stop Task If Idle State End:	Disabled
Power Mgmt: No Start On Batteries:	Disabled
Power Mgmt: Stop On Battery Mode:	Disabled

To log tasks scheduled for a remote computer

The following command requests a list of tasks scheduled for a remote computer, and adds the tasks to a comma-separated log file on the local computer. You can use this command format to collect and track tasks that are scheduled for multiple computers.

The command uses the `/s` parameter to identify the remote computer, Reskit16, the `/fo` parameter to specify the format and the `/nh` parameter to suppress the column headings. The `>>` append symbol redirects the output to the task log, p0102.csv, on the local computer, Svr01. Because the command runs on the remote computer, the local computer path must be fully qualified.

```
schtasks /query /s Reskit16 /fo csv /nh >> \\svr01\data\tasklogs\p0102.csv
```

In response, SchTasks.exe adds the tasks scheduled for the Reskit16 computer to the p0102.csv file on the local computer, Svr01.

- SchTasks.exe performs operations similar to those in Scheduled Tasks in Control Panel. You can use either tool to create, delete, configure, or display scheduled tasks.
- Typing **schtasks** without any parameters performs a query.
- The user must be a member of the Administrators group on the computer that the command affects.
- To verify that a scheduled task ran or to find out why a scheduled task did not run, see the Task Scheduler service transaction log, *Systemroot\SchedLgU.txt*. This log records attempted runs initiated by all tools that use the service, including Scheduled Tasks and SchTasks.exe.
- On rare occasions, task files become corrupted. Corrupted tasks do not run. When you try to perform an operation on corrupted tasks, SchTasks.exe displays the following error message:

```
ERROR: The data is invalid.
```

You cannot recover corrupted tasks. To restore the task scheduling features of the system, use SchTasks.exe or Scheduled Tasks to delete the tasks from the system and reschedule them.

- SchTasks.exe replaces At.exe, a tool included in previous versions of Windows.

Formatting legend

Format	Meaning
<i>Italic</i>	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items
Between braces ({}); choices separated by pipe (). Example: {even odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

[Command-line reference A-Z](#)

[Taskkill](#)

Source: <https://technet.microsoft.com/en-us/library/bb490996.aspx>