Microsoft | **Learn**　Discover ⌄　Product documentation ⌄　Development languages ⌄　Topics ⌄

Sign in

Windows Server　Get started　Failover clustering　Management　Identity and access　Networking　Troubleshooting　Related products ⌄

🔍 Filter by title

⬇ **Download PDF**

Learn / Windows Server /

schtasks create

Article • 10/10/2023 • 5 contributors •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10, ✅ Azure Stack HCI, versions 23H2 and 22H2

👍 Feedback

In this article

Syntax

To schedule a task to run every \<n> minutes

To schedule a task to run every \<n> hours

To schedule a task to run every \<n> days

Show 16 more

Schedules a task.

Syntax

📋 Copy

```
schtasks /create /sc <scheduletype> /tn <taskname> /tr <taskrun> [/s <computer>
```

Parameters

⛶ Expand table

Parameter	Description
/sc `<scheduletype>`	Specifies the schedule type. The valid values include: • **MINUTE** - Specifies the number of minutes before the task should run. • **HOURLY** - Specifies the number of hours before the task should run. • **DAILY** - Specifies the number of days before the task should run. • **WEEKLY** Specifies the number of weeks before the task should run. • **MONTHLY** - Specifies the number of months before the task should run. • **ONCE** - Specifies that that task runs once at a specified date and time. • **ONSTART** - Specifies that 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** - Specifies that 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** - Specifies that 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. • **ONEVENT** - Specifies that the task runs based on an event that matches information from the system event log including the EventID.
/tn `<taskname>`	Specifies a name for the task. Each task on the system must have a unique name and must conform to the rules for file names, not exceeding 238 characters. Use quotation marks to enclose names that include spaces. To store your scheduled task in a different folder, run **/tn** `<folder name\task name>`.
/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. The path name

	must not exceed 262 characters. If you don't add the path, **schtasks** assumes that the file is in the `<systemroot>\System32` directory.
/s `<computer>`	Specifies the name or IP address of a remote computer (with or without backslashes). The default is the local computer.
/u `[<domain>]`	Runs this command with the permissions of the specified user account. The default is the permissions of the current user of the local computer. The **/u** and **/p** parameters are valid only when you use **/s**. The permissions of the specified account are used to schedule the task and to run the task. To run the task with the permissions of a different user, use the **/ru** parameter. The user account must be a member of the Administrators group on the remote computer. Also, the local computer must be in the same domain as the remote computer, or must be in a domain that is trusted by the remote computer domain.
/p `<password>`	Specifies the password of the user account specified in the **/u** parameter. If you use the **/u** parameter without the **/p** parameter or the password argument, schtasks will prompt you for a password. The **/u** and **/p** parameters are valid only when you use **/s**.
/ru `{[<domain>]` `<user> | system}`	Runs the task with permissions of the specified user account. By default, the task runs with the permissions of the current user of the local computer, or with the permission of the user specified by the **/u** parameter, if one is included. The **/ru** parameter is valid when scheduling tasks on local or remote computers. The valid options include: • **Domain** - Specifies an alternate user account. • **System** - Specifies the local System account, a highly privileged account used by the operating system and system services.
/rp `<password>`	Specifies the password for the existing user account, or the user account specified by the **/ru** parameter. If you don't use this parameter when specifying a user account, SchTasks.exe will prompt you for the password next time you sign in. Don't use the **/rp** parameter for tasks that run with System account credentials (**/ru System**). The System account doesn't have a password and SchTasks.exe doesn't prompt for one.

/mo `<modifiers>`	Specifies how often the task runs within its schedule type. The valid options include:
	• **MINUTE** - Specifies that the task runs every `<n>` minutes. You can use any value between 1 - 1439 minutes. By default, this is 1 minute.
	• **HOURLY** - Specifies that the task runs every `<n>` hours. You can use any value between 1 - 23 hours. By default, this is 1 hour.
	• **DAILY** - Specifies that the task runs every `<n>` days. You can use any value between 1 - 365 days. By default, this is 1 day.
	• **WEEKLY** - Specifies that the task runs every `<n>` weeks. You can use any value between 1 - 52 weeks. By default, this is 1 week.
	• **MONTHLY** - Specifies that the task runs every `<n>` months. You can use any of the following values:
	  ◦ A number between 1 - 12 months
	  ◦ **LASTDAY** - To run the task on the last day of the month
	  ◦ **FIRST, SECOND, THIRD, or FOURTH along with the** `/d <day>` **parameter** - Specifies the particular week and day to run the task. For example, on the third Wednesday of the month.
	• **ONCE** - Specifies that the task runs once.
	• **ONSTART** - Specifies that the task runs at startup.
	• **ONLOGON** - Specifies that the task runs when the user specified by the **/ru** parameter logs on.
	• **ONIDLE** - Specifies that the task runs after the system is idle for the number of minutes specified by the **/i** parameter

/d DAY[,DAY...]	Specifies how often the task runs within its schedule type. The valid options include: • **WEEKLY** - Specifies that the task runs weekly by providing a value between 1-52 weeks. Optionally, you can also add a specific day of the week by adding a value of MON - SUN or a range of [MON - SUN...]). • **MONTHLY** - Specifies that the task runs weekly each month by providing a value of FIRST, SECOND, THIRD, FOURTH, LAST. Optionally, you can also add a specific day of the week by adding a value of MON - SUN or by providing a number between 1 - 12 months. If you use this option, you

	can also add a specific day of the month, by providing a number between 1-31. **NOTE:** The date value of 1 - 31 is valid only without the **/mo** parameter, or if the **/mo** parameter is monthly (1 - 12). The default is day 1 (the first day of the month).
/m MONTH[,MONTH…]	Specifies a month or months of the year during which the scheduled task should run. The valid options include JAN - DEC and `*` (every month). The **/m** parameter is valid only with a MONTHLY schedule. It's required when the LASTDAY modifier is used. Otherwise, it's optional and the default value is `*` (every month).
/i `<Idletime>`	Specifies how many minutes the computer is idle before the task starts. A valid value is a whole number from 1 to 999. This parameter is valid only with an ONIDLE schedule, and then it's required.
/st `<Starttime>`	Specifies the start time for the task, using the 24-hour time format, HH:mm. The default value is the current time on the local computer. The **/st** parameter is valid with MINUTE, HOURLY, DAILY, WEEKLY, MONTHLY, and ONCE schedules. It's required for a ONCE schedule.
/ri `<interval>`	Specifies the repetition interval for the scheduled task, in minutes. This isn't applicable for schedule types: MINUTE, HOURLY, ONSTART, ONLOGON,ONIDLE, and ONEVENT. Valid range is 1 - 599940 (599940 minutes = 9999 hours). If either the **/et** or **/du** parameters are specified, the default is **10 minutes**.
/et `<endtime>`	Specifies the time of day that a minute or hourly task schedule ends in <HH:MM> 24-hour format. After the specified end time, schtasks does not start the task again until the start time recurs. By default, task schedules have no end time. This parameter is optional and valid only with a MINUTE or HOURLY schedule.
/du `<duration>`	Specifies a maximum length of time for a minute or hourly schedule in <HHHH:MM> 24-hour format. After the specified time elapses, schtasks does not start the task again until the start time recurs. By default, task schedules have no maximum duration. This parameter is optional and valid only with a MINUTE or HOURLY schedule.
/k	Stops the program that the task runs at the time specified by **/et** or **/du**. Without **/k**, schtasks doesn't start the program again after it reaches the time specified by **/et** or **/du** nor does it stop the program if it's still running. This parameter is optional and valid only with a MINUTE or HOURLY schedule.
/sd `<Startdate>`	Specifies the date on which the task schedule starts. The default value is the current date on the local computer. The format for **Startdate** varies with the locale selected for the local computer in **Regional and Language Options**. Only one format is valid for each locale. The valid date formats include (be sure to choose the format most similar to the format selected for **Short date** in **Regional and Language Options** on the local computer): • `<MM>//` - Specifies to use month-first formats, such as English (United States) and Spanish (Panama). • `<DD>//` - Specifies to use day-first formats, such as Bulgarian and Dutch (Netherlands). • `<YYYY>//` - Specifies to use for year-first formats, such as Swedish and French (Canada).
/ed `<Enddate>`	Specifies the date on which the schedule ends. This parameter is optional. It isn't valid in a ONCE, ONSTART, ONLOGON, ONIDLE, or ONEVENT schedule. By default, schedules have no ending date. The default value is the current date on the local computer. The format for **Enddate** varies with the locale selected for the local computer in **Regional and Language Options**. Only one format is valid for each locale. The valid date formats include (be sure to choose the format most similar to the format selected for **Short date** in **Regional and Language Options** on the local computer): • `<MM>//` - Specifies to use month-first formats, such as English (United States) and Spanish (Panama). • `<DD>//` - Specifies to use day-first formats, such as Bulgarian and Dutch (Netherlands).

	• `<YYYY>//` - Specifies to use for year-first formats, such as Swedish and French (Canada).
/ec `<channelname>`	Specifies the event channel name triggered by the ONEVENT schedule type that matches a system event log criteria.
/it	Specifies to run the scheduled task only when the run as user (the user account under which the task runs) is logged on to the computer. This parameter has no effect on tasks that run with system permissions or tasks that already have the interactive-only property set. You can't use a change command to remove the interactive-only property from a task. By default, run as user is the current user of the local computer when the task is scheduled or the account specified by the **/u** parameter, if one is used. However, if the command includes the **/ru** parameter, then the run as user is the account specified by the **/ru** parameter.
/np	No password is stored. The task runs non-interactively as the given user. Only local resources are available.
/z	Specifies to delete the task upon the completion of its schedule.
/xml `<xmlfile>`	Creates a task specified in the XML file. Can be combined with the **/ru** and **/rp** parameters, or with the **/rp** parameter by itself if the XML file already contains the user account information.
/v1	Creates a task visible to pre-Vista operating systems. This is not compatible with the **/XML** parameter.
/f	Specifies to create the task and suppress warnings if the specified task already exists.
/rl `<level>`	Specifies the Run Level for the job. Acceptable values are **LIMITED** (scheduled tasks will be ran with the least level of privileges, such as Standard User accounts) and **HIGHEST** (scheduled tasks will be ran with the highest level of privileges, such as Superuser accounts). The default value is **Limited**.
/delay `<delaytime>`	Specifies the wait time to delay running the task after it's triggered in mmmm:ss format. This is only valid for the ONSTART, ONLOGON, and ONEVENT schedule types.
/hresult	Specifies the process exit code to be in HRESULT format.
/?	Displays help at the command prompt.

To schedule a task to run every `<n>` minutes

In a minute schedule, the **/sc minute** parameter is required. The **/mo** (modifier) parameter is optional and specifies the number of minutes between each run of the task. The default value for **/mo** is *1* (every minute). The **/et** (end time) and **/du** (duration) parameters are optional and can be used with or without the **/k** (end task) parameter.

Examples

- To schedule a security script, *Sec.vbs*, to run every 20 minutes, type:

  <div style="text-align:right">⎘ Copy</div>

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

  Because this example doesn't 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.

- To schedule a security script, *Sec.vbs*, to run on the local computer every 100 minutes between 5:00 P.M. and 7:59 A.M. each day, type:

  Copy
  ```
  schtasks /create /tn "Security Script" /tr sec.vbs /sc minute /mo 100 /st 1
  ```

  This example uses the **/sc** parameter to specify a minute schedule and the **/mo** parameter to specify an interval of 100 minutes. It uses the **/st** and **/et** parameters to specify the start time and end time of each day's schedule. It also uses the **/k** parameter to stop the script if it's still running at 7:59 A.M. Without **/k**, schtasks wouldn't start the script after 7:59 A.M., but if the instance started at 6:20 A.M. was still running, it wouldn't stop it.

To schedule a task to run every `<n>` hours

In an hourly schedule, the **/sc hourly** parameter is required. The **/mo** (modifier) parameter is optional and specifies the number of hours between each run of the task. The default value for **/mo** is *1* (every hour). The **/k** (end task) parameter is optional and can be used with either **/et** (end at the specified time) or **/du** (end after the specified interval).

Examples

- To schedule the MyApp program to run every five hours, beginning on the first day of March 2002, type:

  Copy
  ```
  schtasks /create /sc hourly /mo 5 /sd 03/01/2002 /tn MyApp /tr c:\apps\myap
  ```

  In this example, the local computer uses the **English (Zimbabwe)** option in **Regional and Language Options**, so the format for the start date is MM/DD/YYYY (03/01/2002).

- To schedule the MyApp program to run hourly, beginning at five minutes past midnight, type:

  Copy
  ```
  schtasks /create /sc hourly /st 00:05 /tn MyApp /tr c:\apps\myapp.exe
  ```

- To schedule the MyApp program to run every 3 hours, for 10 hours total, type:

  Copy
  ```
  schtasks /create /tn MyApp /tr MyApp.exe /sc hourly /mo 3 /st 00:00 /du 001
  ```

  In this example, the task runs at 12:00 A.M., 3:00 A.M., 6:00 A.M., and 9:00 A.M. Because the duration is 10 hours, the task isn't run again at 12:00 P.M. Instead, it starts again at 12:00 A.M. the next day. Also, because the program runs for just a few minutes, the **/k** parameter, which stops the program if it's still running when the duration expires, isn't necessary.

To schedule a task to run every `<n>` days

In a daily schedule, the **/sc daily** parameter is required. The **/mo** (modifier) parameter is optional and specifies the number of days between each run of the task. The default value for **/mo** is *1* (every day).

Examples

- To schedule the MyApp program to run once a day, every day, at 8:00 A.M. until December 31, 2021, type:

  📋 Copy

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc daily /st 08:00 /ed 31
  ```

  In this example, the local computer system is set to the **English (United Kingdom)** option in **Regional and Language Options**, so the format for the end date is DD/MM/YYYY (31/12/2021). Additionally, because this example doesn't include the **/mo** parameter, the default interval of *1* is used to run the command every day.

- To schedule the MyApp program to run every twelve days at 1:00 P.M. (13:00) beginning on December 31, 2021, type:

  📋 Copy

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc daily /mo 12 /sd 12/31
  ```

  In this example, the system is set to the **English (Zimbabwe)** option in **Regional and Language Options**, so the format for the end date is MM/DD/YYYY (12/31/2021).

- To schedule a security script, *Sec.vbs*, to run every 70 days, type:

  📋 Copy

  ```
  schtasks /create /tn "Security Script" /tr sec.vbs /sc daily /mo 70 /it
  ```

  In this example, the **/it** parameter is used to specify that the task runs only when the user under whose account the task runs is logged onto the computer. Because the task runs with the permissions of a specific user account, this task only runs when that user is logged on.

  > ⓘ **Note**
  >
  > To identify tasks with the interactive-only (**/it**) property, use a verbose query (**/query /v**). In a verbose query display of a task with /it, the **Logon Mode** field has a value of Interactive only.

To schedule a task to run every `<n>` weeks

In a weekly schedule, the **/sc weekly** parameter is required. The **/mo** (modifier) parameter is optional and specifies the number of weeks between each run of the task. The default value for **/mo** is *1* (every week).

Weekly schedules also have an optional **/d** parameter to schedule the task to run on specified days of the week, or on all days (). The default is *MON (Monday)*. The every day () option is equivalent to scheduling a daily task.

Examples

- To schedule the MyApp program to run on a remote computer every six weeks, type:

  📋 Copy

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc weekly /mo 6 /s Server
  ```

Because this example leaves out the **/d** parameter, the task runs on Mondays. This example also uses the **/s** parameter to specify the remote computer and the **/u** parameter to run the command with the permissions of the user's Administrator account. Additionally, because the **/p** parameter is left out, SchTasks.exe prompts the user for the Administrator account password, and because the command is run remotely, all paths in the command, including the path to MyApp.exe, refer to paths on the remote computer.

- To schedule a task to run every other Friday, type:

  📋 Copy

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

  This example 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, leave out the **/mo** parameter or set it to *1*.

To schedule a task to run every `<n>` months

In this schedule type, the **/sc monthly** parameter is required. The **/mo** (modifier) parameter, which specifies the number of months between each run of the task, is optional and the default is *1* (every month). This schedule type also has an optional **/d** parameter to schedule the task to run on a specified date of the month. The default is *1* (the first day of the month).

Examples

- To schedule the MyApp program to run on the first day of every month, type:

  📋 Copy

  ```
  schtasks /create /tn MyApp /tr myapp.exe /sc monthly
  ```

  The default value for both the **/mo** (modifier) parameter and the **/d** (day) parameter is *1*, so you don't need to use either of those parameters for this example.

- To schedule the MyApp program to run every three months, type:

  📋 Copy

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

  This example uses the **/mo** parameter to specify an interval of 3 months.

- To schedule the MyApp program to run every other month on the 21st day of the month at midnight for a year, from July 2, 2002 to June 30, 2003, type:

  📋 Copy

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc monthly /mo 2 /d 21 /s
  ```

  This example uses the **/mo** parameter to specify the monthly interval (every two months), the **/d** parameter to specify the date, the **/st** parameter to specify the time, and the **/sd** and **/ed** parameters to specify the start date and end date, respectively. Also in this example, the local computer is set to the **English (South Africa)** option in **Regional and Language Options**, so the dates are specified in the local format, YYYY/MM/DD.

To schedule a task to run on a specific day of the week

The day of the week schedule is a variation of the weekly schedule. In a weekly schedule, the **/sc weekly** parameter is required. The **/mo** (modifier) parameter is optional and specifies the number of weeks between each run of the task. The default value for **/mo** is *1* (every week). The **/d** parameter, which is optional, schedules the task to run on specified days of the week, or on all days (*). The default is *MON (Monday)*. The every day option `(/d *)` is equivalent to scheduling a daily task.

Examples

- To schedule the MyApp program to run every week on Wednesday, type:

  <div>Copy</div>

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc weekly /d WED
  ```

  This example uses the **/d** parameter to specify the day of the week. Because the command leaves out the **/mo** parameter, the task runs every week.

- To schedule a task to run on Monday and Friday of every eighth week, type:

  <div>Copy</div>

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc weekly /mo 8 /d MON,FR
  ```

  This example uses the **/d** parameter to specify the days and the **/mo** parameter to specify the eight-week interval.

To schedule a task to run on a specific week of the month

In this schedule type, the **/sc monthly** parameter, the **/mo** (modifier) parameter, and the **/d** (day) parameter are required. The **/mo** (modifier) parameter specifies the week on which the task runs. The **/d** parameter specifies the day of the week. You can specify only one day of the week for this schedule type. This schedule also has an optional **/m** (month) parameter that lets you schedule the task for particular months or every month (*). The default for the **/m** parameter is every month (*).

Examples

- To schedule the MyApp program to run on the second Sunday of every month, type:

  <div>Copy</div>

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc monthly /mo SECOND /d
  ```

  This example uses the **/mo** parameter to specify the second week of the month and the **/d** parameter to specify the day.

- To schedule the MyApp program to run on the first Monday in March and September, type:

  <div>Copy</div>

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc monthly /mo FIRST /d M
  ```

This example uses the **/mo** parameter to specify the first week of the month and the **/d** parameter to specify the day. It uses the **/m** parameter to specify the month, separating the month arguments with a comma.

To schedule a task to run on a specific day each month

In this schedule type, the **/sc monthly** parameter and the **/d** (day) parameter are required. The **/d** parameter specifies a date of the month (1 - 31), not a day of the week, and you can specify only one day in the schedule. The **/m** (month) parameter is optional, with the default being every month (), while the **/mo** (modifier) parameter isn't valid with this schedule type.

Schtasks.exe won't let you schedule a task for a date that's not in a month specified by the **/m** parameter. For example, trying to schedule the 31st day of February. However, if you don't use the **/m** parameter, and schedule a task for a date that doesn't appear in every month, then the task won't run in the shorter months. To schedule a task for the last day of the month, use the last day schedule type.

Examples

- To schedule the MyApp program to run on the first day of every month, type:

  <div align="right">📋 Copy</div>

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc monthly
  ```

  Because the default modifier is *none* (no modifier), this command uses the default day of *1*, and the default month of *every month*, without requiring any additional parameters.

- To schedule the MyApp program to run on May 15 and June 15 at 3:00 P.M. (15:00), type:

  <div align="right">📋 Copy</div>

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc monthly /d 15 /m MAY,J
  ```

  This example 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.

To schedule a task to run on the last day of a month

In the last day schedule type, the **/sc monthly** parameter, the **/mo LASTDAY** (modifier) parameter, and the **/m** (month) parameter are required. The **/d** (day) parameter isn't valid.

Examples

- To schedule the MyApp program to run on the last day of every month, type:

  <div align="right">📋 Copy</div>

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc monthly /mo lastday /m
  ```

  This example uses the **/mo** parameter to specify the last day and the **/m** parameter with the wildcard character (*) to indicate that the program runs every month.

- To schedule the MyApp program to run on the last day of February and the last day of March at 6:00 P.M., type:

⎙ Copy

```
schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc monthly /mo lastday /m
```

This example uses the **/mo** parameter to specify the last day, the **/m** parameter to specify the months, and the **/st** parameter to specify the start time.

To schedule to run once

In the run-once schedule type, the **/sc once** parameter is required. The **/st** parameter, which specifies the time that the task runs, is required. The **/sd** parameter, which specifies the date that the task runs, is optional, while the **/mo** (modifier) and **/ed** (end date) parameters aren't valid.

Schtasks won't let you schedule a task to run once if the date and time specified are in the past, based on the time of the local computer. To schedule a task that runs once on a remote computer in a different time zone, you must schedule it before that date and time occurs on the local computer.

Example

- To schedule the MyApp program to run at midnight on January 1, 2003, type:

⎙ Copy

```
schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc once /sd 01/01/2003 /s
```

This example uses the **/sc** parameter to specify the schedule type and the **/sd** and **/st** parameters to specify the date and time. Also in this example, the local computer uses the **English (United States)** option in **Regional and Language Options**, the format for the start date is MM/DD/YYYY.

To schedule a task to run every time the system starts

In the on-start schedule type, the **/sc onstart** parameter is required. The **/sd** (start date) parameter is optional and the default is the current date.

Example

- To schedule the MyApp program to run every time the system starts, type:

⎙ Copy

```
schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc onstart
```

In this example, the local computer uses the **English (United States)** option in **Regional and Language Options**, the format for the start date is MM/DD/YYYY.

To schedule a task to run when a user logs on

The on logon schedule type schedules a task that runs whenever any user logs on to the computer. In the on logon schedule type, the **/sc onlogon** parameter is required. The **/sd** (start date) parameter is optional and the default is the current date.

Example

- To schedule a task that runs when a user logs on to a remote computer, type:

  Copy

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

  This example 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.

To schedule a task to run when the system is idle

The on idle schedule type schedules a task that runs whenever there is no user activity during the time specified by the **/i** parameter. In the on idle schedule type, the **/sc onidle** parameter and the **/i** parameter are required. The **/sd** (start date) is optional and the default is the current date.

Example

- To schedule the MyApp program to run whenever the computer is idle, type:

  Copy

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

  This example uses the required **/i** parameter to specify that the computer must remain idle for ten minutes before the task starts.

To schedule a task to run now

Schtasks doesn't have a Run Now option, but you can simulate that option by creating a task that runs once and starts in a few minutes.

Example

- To schedule a task to run once, on November 13, 2020 at 2:18 P.M. local time, type:

  Copy

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc once /st 14:18 /sd 11/
  ```

  In this example, the local computer uses the **English (United States)** option in **Regional and Language Options**, so the format for the start date is MM/DD/YYYY.

To schedule a task that runs with different permissions

You can schedule tasks of all types to run with permissions of an alternate account on both the local and a remote computer. In addition to the parameters required for the particular schedule type, the **/ru** parameter is required and the **/rp** parameter is optional.

Examples

- To run the MyApp program on the local computer, type:

  Copy

  ```
  schtasks /create /tn MyApp /tr myapp.exe /sc weekly /d TUE /ru Admin06
  ```

  This example uses the **/ru** parameter to specify that the task should run with the permissions of the user's Administrator account (*Admin06*). Also in this example, the task is scheduled to run every Tuesday, but you can use any schedule type for a task run with alternate permissions.

  In response, SchTasks.exe prompts for the run as password for the *Admin06* account, and then displays a success message:

  Copy

  ```
  Please enter the run as password for Admin06: ********
  SUCCESS: The scheduled task MyApp has successfully been created.
  ```

- To run the MyApp program on the *Marketing* computer every four days, type:

  Copy

  ```
  schtasks /create /tn MyApp /tr myapp.exe /sc daily /mo 4 /s Marketing /u Ma
  ```

  This example uses the **/sc** parameter to specify a daily schedule, and the **/mo** parameter to specify an interval of four days. Additionally, this example uses the **/s** parameter to provide the name of the remote computer and the **/u** parameter to specify an account with permission to schedule a task on the remote computer (*Admin01 on the Marketing computer*). Finally, this example uses the **/ru** parameter to specify that the task should run with the permissions of the user's non-Administrator account (*User01* in the *Reskits* domain). Without the **/ru** parameter, the task would run with the permissions of the account specified by **/u**.

  When running this example, Schtasks first requests the password of the user named by the **/u** parameter (to run the command) and then requests the password of the user named by the **/ru** parameter (to run the task). After authenticating the passwords, schtasks displays a message indicating that the task is scheduled:

  Copy

  ```
  Type the password for Marketing\Admin01:********
  Please enter the run as password for Reskits\User01: ********
  SUCCESS: The scheduled task MyApp has successfully been created.
  ```

- To run schedule the *AdminCheck.exe* program to run on the Public computer every Friday at 4:00 A.M.,, but only if the administrator of the computer is logged on, type:

  Copy

  ```
  schtasks /create /tn "Check Admin" /tr AdminCheck.exe /sc weekly /d FRI /st
  ```

  This example uses the **/sc** parameter to specify a weekly schedule, the **/d** parameter to specify the day, and the **/st** parameter to specify the start time. It also uses the **/s** parameter to provide the name of the remote computer, the **/u** parameter to specify an account with permission to schedule a task on the remote computer, the **/ru** parameter to configure the task to run with the permissions of the administrator of the Public computer (*Public\Admin01*), and the **/it** parameter to indicate that the task runs only when the *Public\Admin01* account is logged on.

> ⓘ **Note**
>
> To identify tasks with the interactive-only (**/it**) property, use a verbose query (`/query /v`). In a verbose query display of a task with **/it**, the **Logon Mode** field has a value of **Interactive only**.

To schedule a task that runs with system permissions

Tasks of all types can run with permissions of the **System** account on both the local and a remote computer. In addition to the parameters required for the particular schedule type, the **/ru system** (or **/ru**) parameter is required, while the **/rp** parameter isn't valid.

> ⓘ **Important**
>
> The **System** account doesn't have interactive logon rights. Users can't see or interact with programs or tasks run with system permissions. The **/ru** parameter determines the permissions under which the task runs, not the permissions used to schedule the task. Only Administrators can schedule tasks, regardless of the value of the **/ru** parameter.
>
> To identify tasks that run with system permissions, use a verbose query (`/query /v`). In a verbose query display of a system-run task, the **Run As User** field has a value of **NT AUTHORITY\SYSTEM** and the **Logon Mode** field has a value of **Background only**.

Examples

- To schedule the MyApp program to run on the local computer with permissions of the **System** account, type:

  ```
  schtasks /create /tn MyApp /tr c:\apps\myapp.exe /sc monthly /d 15 /ru Syst
  ```

  In this example, the task is scheduled to run on the fifteenth day of every month, but you can use any schedule type for a task run with system permissions. Additionally, this example uses the **/ru System** parameter to specify the system security context. Because system tasks don't use a password, the **/rp** parameter is left out.

  In response, SchTasks.exe displays an informational message and a success message, without prompting for a password:

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

- To schedule the MyApp program to run on the *Finance01* computer every morning at 4:00 A.M., using system permissions, type:

  ```
  schtasks /create /tn MyApp /tr myapp.exe /sc daily /st 04:00 /s Finance01 /
  ```

  This example uses the **/tn** parameter to name the task and the **/tr** parameter to specify the remote copy of the MyApp program, the **/sc** parameter to specify a daily schedule, but leaves out the **/mo** parameter because *1* (every day) is the default. This example also uses the **/st** parameter to specify the start time, which is also the time the task will run

each day, the **/s** parameter to provide the name of the remote computer, the **/u** parameter to specify an account with permission to schedule a task on the remote computer, and the **/ru** parameter to specify that the task should run under the System account. Without the **/ru** parameter, the task would run using the permissions of the account specified by the **/u** parameter.

Schtasks.exe requests the password of the user named by the **/u** parameter and, after authenticating the password, displays a message indicating that the task is created and that it will run with permissions of the **System** account:

⎙ Copy

```
Type the password for Admin01:**********

INFO: The Schedule Task MyApp will be created under user name (NT AUTHORITY
SYSTEM).
SUCCESS: The scheduled task MyApp has successfully been created.
```

To schedule 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.

1. Using a text editor, such as Notepad, create a batch file that includes the name and fully qualified path to the .exe file required to start the Event Viewer (Eventvwr.exe) and System Monitor (Perfmon.exe) programs.

   ⎙ Copy

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

2. Save the file as *MyApps.bat*, open schtasks.exe, and then create a task to run *MyApps.bat* by typing:

   ⎙ Copy

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

   This command creates the Monitor task, which runs whenever anyone logs on. It uses the **/tn** parameter to name the task, the **/tr** parameter to run MyApps.bat, the **/sc** parameter to indicate the OnLogon schedule type and the **/ru** parameter to run the task with the permissions of the user's Administrator account.

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

To schedule a task that runs on a remote computer

To schedule a task to run on a remote computer, you must add the task to the remote computer's schedule. Tasks of all types can be scheduled on a remote computer, but the following conditions must be met:

- You must have permission to schedule the task. As such, you must be logged on to the local computer with an account that is a member of the Administrators group on the remote computer, or you must use the **/u** parameter to provide the credentials of an Administrator of the remote computer.

- You can use the **/u** parameter only when the local and remote computers are in the same domain or the local computer is in a domain that the remote computer domain trusts. Otherwise, the remote computer cannot authenticate the user account specified and it cannot verify that the account is a member of the Administrators group.

- The task must have sufficient permission to run on the remote computer. The permissions required vary with the task. By default, the task runs with the permission of the current user of the local computer or, if the **/u** parameter is used, the task runs with the permission of the account specified by the **/u** parameter. However, you can use the **/ru** parameter to run the task with permissions of a different user account or with system permissions.

Examples

- To schedule the MyApp program (as an administrator) to run on the *SRV01* remote computer every ten days starting immediately,type:

           ⧉ Copy

  ```
  schtasks /create /s SRV01 /tn MyApp /tr c:\program files\corpapps\myapp.exe
  ```

  This example uses the **/s** parameter to provide the name of the remote computer. Because the local current user is an Administrator of the remote computer, the **/u** parameter, which provides alternate permissions for scheduling the task, isn't necessary.

  > ⓘ **Note**
  >
  > When scheduling tasks on a remote computer, all parameters refer to the remote computer. Therefore, the file specified by the **/tr** parameter refers to the copy of MyApp.exe on the remote computer.

- To schedule the MyApp program (as a user) to run on the *SRV06* remote computer every three hours, type:

           ⧉ Copy

  ```
  schtasks /create /s SRV06 /tn MyApp /tr c:\program files\corpapps\myapp.exe
  ```

  Because Administrator permissions are required to schedule a task, the command uses the **/u** and **/p** parameters to provide the credentials of the user's Administrator account (*Admin01* in the *Reskits* domain). By default, these permissions are also used to run the task. However, because the task does not need Administrator permissions to run, the command includes the **/u** and **/rp** parameters to override the default and run the task with permission of the user's non-Administrator account on the remote computer.

- To schedule the MyApp program (as a user) to run on the *SRV02* remote computer on the last day of every month.

           ⧉ Copy

  ```
  schtasks /create /s SRV02 /tn MyApp /tr c:\program files\corpapps\myapp.exe
  ```

  Because the local current user (*user03*) isn't an Administrator of the remote computer, the command uses the **/u** parameter to provide the credentials of the user's Administrator account (*Admin01* in the *Reskits* domain). The Administrator account permissions will be used to schedule the task and to run the task.

  Because the command did not include the **/p** (password) parameter, schtasks prompts for the password. Then it displays a success message and, in this case, a warning:

<div style="text-align: right">📋 Copy</div>

```
Type the password for reskits\admin01:********

SUCCESS: The scheduled task MyApp has successfully been created.
WARNING: The scheduled task MyApp has been created, but may not run because
```

This warning indicates that the remote domain could not authenticate the account specified by the **/u** parameter. In this case, the remote domain could not authenticate the user account because the local computer isn't a member of a domain that the remote computer domain trusts. When this occurs, the task job appears in the list of scheduled tasks, but the task is actually empty and it won't run.

The following display from a verbose query exposes the problem with the task. In the display, note that the value of **Next Run Time** is **Never** and that the value of **Run As User** is **Could not be retrieved from the task scheduler database**.

Had this computer been a member of the same domain or a trusted domain, the task would have been successfully scheduled and would have run as specified.

<div style="text-align: right">📋 Copy</div>

```
HostName: SRV44
TaskName: MyApp
Next Run Time: Never
Status:
Logon mode: Interactive/Background
Last Run Time: Never
Last Result: 0
Creator: user03
Schedule: At 3:52 PM on day 31 of every month, start
starting 12/14/2001
Task To Run: c:\program files\corpapps\myapp.exe
Start In: myapp.exe
Comment: N/A
Scheduled Task State: Disabled
Scheduled Type: Monthly
Start Time: 3:52:00 PM
Start Date: 12/14/2001
End Date: N/A
Days: 31
Months: JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NO
V,DEC
Run As User: Could not be retrieved from the task sched
uler database
Delete Task If Not Rescheduled: Enabled
Stop Task If Runs X Hours and X Mins: 72:0
Repeat: Every: Disabled
Repeat: Until: Time: Disabled
Repeat: Until: Duration: Disabled
Repeat: Stop If Still Running: Disabled
Idle Time: Disabled
Power Management: Disabled
```

Remarks

- To run the **/create** command with the permissions of a different user, use the **/u** parameter. The **/u** parameter is valid only for scheduling tasks on remote computers.

- To view more `schtasks /create` examples, type `schtasks /create /?` at a command prompt.

- To schedule a task that runs with permissions of a different user, use the **/ru** parameter. The **/ru** parameter is valid for tasks on local and remote computers.

- To use the **/u** parameter, the local computer must be in the same domain as the remote computer or it must be in a domain that the remote computer domain trusts. Otherwise, either the task isn't created, or the task job is empty and the task doesn't run.

- Schtasks always prompts for a password unless you provide one, even when you schedule a task on the local computer using the current user account. This is normal behavior for schtasks.

- Schtasks doesn't verify program file locations or user account passwords. If you don't enter the correct file location or the correct password for the user account, the task is created, but it won't run. Also, if the password for an account changes or expires, and you don't change the password saved in the task, then the task won't run.

- The **System** account doesn't have interactive logon rights. Users don't see and can't 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.

Related links

- [Command-Line Syntax Key](#)

- [schtasks change command](#)

- [schtasks delete command](#)

- [schtasks end command](#)

- [schtasks query command](#)

- [schtasks run command](#)

Feedback

Was this page helpful? 👍 Yes   👎 No