Fetch

Summary

Fetch is one of the console tools in MikroTik RouterOS. It is used to copy files to/from a network device via HTTP, HTTPS, FTP or SFTP. It can also be used to send POST/GET requests and send any kind of data to a remote server. In HTTPS mode by default, no certificate checks are made, setting *check-certificate* to *yes* enables trust chain validation from the local certificate store (can be used only in HTTPS mode).

Properties

Property	Description
address (*string*; Default:)	IP address of the device to copy file from.
as-value (*set* | *not-set*; Default: **not-set**)	Store the output in a variable, should be used with the output property.
ascii (*yes* | *no*; Default: **no**)	Can be used with FTP and TFTP
certificate (*string*; Default:)	Certificate that should be used for host verification. Can be used only in HTTPS mode.
check-certificate (*yes* | *yes-without-crl* | *no*; Default: **no**)	Enables trust chain validation from local certificate store. *yes-without-crl, validates a certificate, not performing CRL check (certificate revocation list).* Can be used only in HTTPS mode.
dst-path (*string*; Default:)	Destination filename and path.
duration (*time;* Default:)	Time how long fetch should run.
host (*string*; Default:)	A domain name or virtual domain name (if used on a website, from which you want to copy information). For example, `address=wiki.mikrotik.com host=forum.mikrotik.com` In this example the resolved ip address is the same (66.228.113.27), but hosts are different.
http-auth-scheme (*basic|digest*; Default: **basic**)	HTTP authentication scheme
http-method (*delete|get|head|post|put|patch*; Default: **get**)	HTTP method to use
http-data (*string*; Default:)	The data, that is going to be sent, when using PUT or POST methods. Data limit is 64Kb.
http-header-field (*string*; Default: ***empty***)	List of all header fields and their values, in the form of `http-header-field=h1:fff,h2:yyy`
http-content-encoding (*deflate|gzip*; Default: ***empty***)	Encodes the payload using **gzip** or **deflate** compression and adds a corresponding Content-Encoding header. Usable for HTTP POST and PUT only.
keep-result (*yes* | *no*; Default: **yes**)	If yes, creates an input file.
mode (*ftp|http|https|sftp|tftp*; Default: **http**)	Choose the protocol of connection - http, https , ftp, sftp or tftp.

output (*none*|*file*|*user*|*user-with-headers*; Default: **file**)	Sets where to store the downloaded data. • `none` - do not store downloaded data • `file` - store downloaded data in a file • `user` - store downloaded data in the data variable (variable limit is 64Kb) • user-with-headers - store downloaded data and headers in the data variable (variable limit is 64Kb (20Kb for downloaded data, 44Kb for headers))
password (*string*; Default: **anonymous**)	Password, which is needed for authentication to the remote device.
port (*integer*; Default:)	Connection port.
src-address (ip address; Default:)	Source address that is used to establish connection. Can be used only HTTP/S and SFTP modes.
src-path (*string*; Default:)	Title of the remote file you need to copy.
upload (*yes* | *no*; Default: **no**)	Only (S)FTP modes support upload. If enabled then fetch will be used to upload files to a remote server. Requires *src-path* and *dst-path* parameters to be set.
url (*string*; Default:)	URL pointing to file. Can be used instead of **address** and **src-path** parameters.
user (*string*; Default: **anonymous**)	Username, which is needed for authentication to the remote device.

Configuration Examples

The following example shows how to copy the file with filename "conf.rsc" from a device with ip address 192.168.88.2 by FTP protocol and save it as file with filename "123.rsc". User and password are needed to login into the device.

```
[admin@MikroTik] /tool> fetch address=192.168.88.2 src-path=conf.rsc \
user=admin mode=ftp password=123 dst-path=123.rsc port=21 \
host="" keep-result=yes
```

Example to upload file to another router:

```
[admin@MikroTik] /tool> fetch address=192.168.88.2 src-path=conf.rsc \
user=admin mode=ftp password=123 dst-path=123.rsc upload=yes
```

Another file download example that demonstrates the usage of url property.

```
[admin@MikroTik] /> /tool fetch url="https://www.mikrotik.com/img/netaddresses2.pdf" mode=http
  status: finished

[admin@test_host] /> /file print
 # NAME                    TYPE                SIZE                CREATION-TIME
 ...
 5 netaddresses2.pdf       .pdf file           11547               jun/01/2010 11:59:51
```

Sending information to a remote host

It is possible to use an HTTP POST request to send information to a remote server, that is prepared to accept it. In the following example, we send geographic coordinates to a PHP page:

```
/tool/fetch http-method=post http-header-field="Content-Type:application/json" http-data="{\"lat\":\"56.12\",\"
lon\":\"25.12\"}" url="https://testserver.lv/index.php"
```

In this example, the data is uploaded as a file. Important note, since variable data comes from a file, a file can only be in size up to 4KB. This is a limitation of RouterOS variables.

```
/export file=export.rsc

:global data [/file get [/file find name=export.rsc] contents];
:global $url "https://prod-51.westeurope.logic.azure.com:443/workflows/blabla/triggers/manual/paths/invoke....";

/tool fetch mode=https http-method=put http-data=$data url=$url
```

Return value to a variable

It is possible to save the result of the fetch command to a variable. For example, it is possible to trigger a certain action based on the result that an HTTP page returns. You can find a very simple example below that disables **ether2** whenever a PHP page returns "0":

```
{
    :local result [/tool fetch url=https://10.0.0.1/disable_ether2.php as-value output=user];
    :if ($result->"status" = "finished") do={
        :if ($result->"data" = "0") do={
            /interface ethernet set ether2 disabled=yes;
        } else={
            /interface ethernet set ether2 disabled=no;
        }
    }
}
```