



# File security permissions

## ONTAP Automation

NetApp  
June 10, 2024

This PDF was generated from [https://docs.netapp.com/us-en/ontap-automation/workflows/wf\\_nas\\_fs\\_prepare.html](https://docs.netapp.com/us-en/ontap-automation/workflows/wf_nas_fs_prepare.html) on June 10, 2024. Always check docs.netapp.com for the latest.

# Table of Contents

- File security permissions ..... 1
  - Prepare to manage file security and audit policies ..... 1
  - Get the effective permissions for a file ..... 2
  - Get the auditing information for a file ..... 3
  - Apply new permissions to a file ..... 6
  - Update security descriptor information ..... 7
  - Delete an access control entry ..... 8

# File security permissions

## Prepare to manage file security and audit policies

You can manage the permissions and audit policies for files available through the SVMs within an ONTAP cluster.

### Overview

ONTAP uses System Access Control Lists (SACLs) and Discretionary Access Control Lists (DACLS) to assign permissions to file objects. Beginning with ONTAP 9.9.1, the REST API includes support for managing the SACL and DACL permissions. You can use the API to automate the administration of the file security permissions. In many cases you can use a single REST API call instead of multiple CLI commands or ONTAPI (ZAPI) calls.



For ONTAP releases prior to 9.9.1, you can automate the administration of the SACL and DACL permissions using the CLI passthrough feature. See [Migration considerations](#) and [Using the private CLI passthrough with the ONTAP REST API](#) for more information.

Several example workflows are available to illustrate how to manage the ONTAP file security services using the REST API. Before using the workflows and issuing any of the REST API calls, make sure to review [Prepare to use the workflows](#).

If you use Python, also see the script [file\\_security\\_permissions.py](#) for examples of how to automate some of the file security activities.

### ONTAP REST API versus ONTAP CLI commands

For many tasks, using the ONTAP REST API requires fewer calls than the equivalent ONTAP CLI commands or ONTAPI (ZAPI) calls. The table below includes a list of API calls and the equivalent the CLI commands needed for each task.

ONTAP REST API	ONTAP CLI
GET /protocols/file-security/effective-permissions/	<code>vserver security file-directory show-effective-permissions</code>
POST /protocols/file-security/permissions/	<ol style="list-style-type: none"><li><code>vserver security file-directory ntfs create</code></li><li><code>vserver security file-directory ntfs dacl add</code></li><li><code>vserver security file-directory ntfs sacl add</code></li><li><code>vserver security file-directory policy create</code></li><li><code>vserver security file-directory policy task add</code></li><li><code>vserver security file-directory apply</code></li></ol>
PATCH /protocols/file-security/permissions/	<code>vserver security file-directory ntfs modify</code>

ONTAP REST API	ONTAP CLI
DELETE /protocols/file-security/permissions/	<ol style="list-style-type: none"> <li>1. vserver security file-directory ntfs dacl remove</li> <li>2. vserver security file-directory ntfs sacl remove</li> </ol>

#### Related information

- [Python script illustrating file permissions](#)
- [Simplified management of file-security permissions with ONTAP REST APIs](#)
- [Using the private CLI passthrough with the ONTAP REST API](#)

## Get the effective permissions for a file

You can retrieve the current effective permissions for a specific file or folder.

#### HTTP method and endpoint

This REST API call uses the following method and endpoint.

HTTP method	Path
GET	/api/protocols/file-security/effective-permissions/{svm.uuid}/{path}

#### Processing type

Synchronous

#### Additional input parameters for curl examples

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl example in this step.

Parameter	Type	Required	Description
\$SVM_ID	Path	Yes	This is the UUID of the SVM containing the file.
\$FILE_PATH	Path	Yes	This is the path to the file or folder.

#### Curl example

```
curl --request GET \
--location "https://$FQDN_IP/api/protocols/file-security/effective-
permissions/$SVM_ID/$FILE_PATH" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

## JSON output example

```
{
  "svm": {
    "uuid": "cf5f271a-1beb-11ea-8fad-005056bb645e",
    "name": "vs1"
  },
  "user": "administrator",
  "type": "windows",
  "path": "/",
  "share": {
    "path": "/"
  },
  "file_permission": [
    "read",
    "write",
    "append",
    "read_ea",
    "write_ea",
    "execute",
    "delete_child",
    "read_attributes",
    "write_attributes",
    "delete",
    "read_control",
    "write_dac",
    "write_owner",
    "synchronize",
    "system_security"
  ],
  "share_permission": [
    "read",
    "read_ea",
    "execute",
    "read_attributes",
    "read_control",
    "synchronize"
  ]
}
```

## Get the auditing information for a file

You can retrieve the auditing information for a specific file or folder.

### HTTP method and endpoint

This REST API call uses the following method and endpoint.

HTTP method	Path
GET	/api/protocols/file-security/permissions/{svm.uuid}/{path}

### Processing type

Synchronous

### Additional input parameters for curl examples

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl example in this step.

Parameter	Type	Required	Description
\$SVM_ID	Path	Yes	This is the UUID of the SVM containing the file.
\$FILE_PATH	Path	Yes	This is the path to the file or folder.

### Curl example

```
curl --request GET \
--location "https://$FQDN_IP/api/protocols/file-
security/permissions/$SVM_ID/$FILE_PATH" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

### JSON output example

```
{
  "svm": {
    "uuid": "9479099d-5b9f-11eb-9c4e-0050568e8682",
    "name": "vs1"
  },
  "path": "/parent",
  "owner": "BUILTIN\\Administrators",
  "group": "BUILTIN\\Administrators",
  "control_flags": "0x8014",
  "acls": [
    {
      "user": "BUILTIN\\Administrators",
      "access": "access_allow",
      "apply_to": {
        "files": true,
        "sub_folders": true,
        "this_folder": true
      },
      "advanced_rights": {
        "append_data": true,

```

```

    "delete": true,
    "delete_child": true,
    "execute_file": true,
    "full_control": true,
    "read_attr": true,
    "read_data": true,
    "read_ea": true,
    "read_perm": true,
    "write_attr": true,
    "write_data": true,
    "write_ea": true,
    "write_owner": true,
    "synchronize": true,
    "write_perm": true
  },
  "access_control": "file_directory"
},
{
  "user": "BUILTIN\\Users",
  "access": "access_allow",
  "apply_to": {
    "files": true,
    "sub_folders": true,
    "this_folder": true
  },
  "advanced_rights": {
    "append_data": true,
    "delete": true,
    "delete_child": true,
    "execute_file": true,
    "full_control": true,
    "read_attr": true,
    "read_data": true,
    "read_ea": true,
    "read_perm": true,
    "write_attr": true,
    "write_data": true,
    "write_ea": true,
    "write_owner": true,
    "synchronize": true,
    "write_perm": true
  },
  "access_control": "file_directory"
}
],
"inode": 64,

```

```
"security_style": "mixed",
"effective_style": "ntfs",
"dos_attributes": "10",
"text_dos_attr": "----D---",
"user_id": "0",
"group_id": "0",
"mode_bits": 777,
"text_mode_bits": "rwxrwxrwx"
}
```

## Apply new permissions to a file

You can apply a new security descriptor to a specific file or folder.

### Step 1: Apply the new permissions

#### HTTP method and endpoint

This REST API call uses the following method and endpoint.

HTTP method	Path
POST	/api/protocols/file-security/permissions/{svm.uuid}/{path}

#### Processing type

Asynchronous

#### Additional input parameters for curl examples

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl example in this step.

Parameter	Type	Required	Description
\$SVM_ID	Path	Yes	This is the UUID of the SVM containing the file.
\$FILE_PATH	Path	Yes	This is the path to the file or folder.



## Curl example

```
curl --request POST --location "https://$FQDN_IP/api/protocols/file-security/permissions/$SVM_ID/$FILE_PATH?return_timeout=0" --include --header "Accept */*" --header "Authorization: Basic $BASIC_AUTH" --data '{ \acl\": [ { \access\": \access_allow\", \advanced_rights\": { \append_data\": true, \delete\": true, \delete_child\": true, \execute_file\": true, \full_control\": true, \read_attr\": true, \read_data\": true, \read_ea\": true, \read_perm\": true, \write_attr\": true, \write_data\": true, \write_ea\": true, \write_owner\": true, \write_perm\": true }, \apply_to\": { \files\": true, \sub_folders\": true, \this_folder\": true }, \user\": \administrator\" } ], \control_flags\": \32788\", \group\": \S-1-5-21-2233347455-2266964949-1780268902-69700\", \ignore_paths\": [ \parent/child2\" ], \owner\": \S-1-5-21-2233347455-2266964949-1780268902-69304\", \propagation_mode\": \propagate\''
```

## JSON output example

```
{
  "job": {
    "uuid": "3015c294-5bbc-11eb-9c4e-0050568e8682",
    "_links": {
      "self": {
        "href": "/api/cluster/jobs/3015c294-5bbc-11eb-9c4e-0050568e8682"
      }
    }
  }
}
```

## Step 2: Retrieve the status of the job

Perform the workflow [Get job instance](#) and confirm the `state` value is `success`.

## Update security descriptor information

You can update a specific security descriptor to a specific file or folder, including the primary owner, group, or control flags.

### Step 1: Update the security descriptor

#### HTTP method and endpoint

This REST API call uses the following method and endpoint.

HTTP method	Path
PATCH	/api/protocols/file-security/permissions/{svm.uuid}/{path}

### Processing type

Asynchronous

### Additional input parameters for curl examples

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl example in this step.

Parameter	Type	Required	Description
\$SVM_ID	Path	Yes	This is the UUID of the SVM containing the file.
\$FILE_PATH	Path	Yes	This is the path to the file or folder.

### Curl example

```
curl --request POST --location "https://$FQDN_IP/api/protocols/file-security/permissions/$SVM_ID/$FILE_PATH?return_timeout=0" --include --header "Accept */*" --header "Authorization: Basic $BASIC_AUTH" --data '{ \"control_flags\": \"32788\", \"group\": \"everyone\", \"owner\": \"user1\"}'
```

### JSON output example

```
{
  "job": {
    "uuid": "6f89e612-5bbd-11eb-9c4e-0050568e8682",
    "_links": {
      "self": {
        "href": "/api/cluster/jobs/6f89e612-5bbd-11eb-9c4e-0050568e8682"
      }
    }
  }
}
```

## Step 2: Retrieve the status of the job

Perform the workflow [Get job instance](#) and confirm the state value is success.

## Delete an access control entry

You can delete an existing Access Control Entry (ACE) from a specific file or folder. The change propagates to any child objects.

## Step 1: Delete the ACE

### HTTP method and endpoint

This REST API call uses the following method and endpoint.

HTTP method	Path
DELETE	/api/protocols/file-security/permissions/{svm.uuid}/{path}

### Processing type

Asynchronous

### Additional input parameters for curl examples

In addition to the parameters common with all REST API calls, the following parameters are also used in the curl example in this step.

Parameter	Type	Required	Description
\$SVM_ID	Path	Yes	This is the UUID of the SVM containing the file.
\$FILE_PATH	Path	Yes	This is the path to the file or folder.

### Curl example

```
curl --request DELETE --location "https://$FQDN_IP/api/protocols/file-security/permissions/$SVM_ID/$FILE_PATH?return_timeout=0" --include --header "Accept */*" --header "Authorization: Basic $BASIC_AUTH" --data '{ \"access\": \"access_allow\", \"apply_to\": { \"files\": true, \"sub_folders\": true, \"this_folder\": true }, \"ignore_paths\": [ \"/parent/child2\" ], \"propagation_mode\": \"propagate\"}'
```

### JSON output example

```
{
  "job": {
    "uuid": "3015c294-5bbc-11eb-9c4e-0050568e8682",
    "_links": {
      "self": {
        "href": "/api/cluster/jobs/3015c294-5bbc-11eb-9c4e-0050568e8682"
      }
    }
  }
}
```

## Step 2: Retrieve the status of the job

Perform the workflow [Get job instance](#) and confirm the state value is success.

## Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

## Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.