



# File Access Manager Access Fulfillment Using a Script

Version: 8.2 Revised: July 01, 2021

---

## Copyright and Trademark Notices.

Copyright © 2021 SailPoint Technologies, Inc. All Rights Reserved.

All logos, text, content, including underlying HTML code, designs, and graphics used and/or depicted on these written materials or in this Internet website are protected under United States and international copyright and trademark laws and treaties, and may not be used or reproduced without the prior express written permission of SailPoint Technologies, Inc.

“SailPoint,” “SailPoint & Design,” “SailPoint Technologies & Design,” “Identity Cube,” “Identity IQ,” “IdentityAI,” “IdentityNow,” “SailPoint Predictive Identity” and “SecurityIQ” are registered trademarks of SailPoint Technologies, Inc. None of the foregoing marks may be used without the prior express written permission of SailPoint Technologies, Inc. All other trademarks shown herein are owned by the respective companies or persons indicated.

SailPoint Technologies, Inc. makes no warranty of any kind with regard to this manual or the information included therein, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SailPoint Technologies shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Patents Notice. <https://www.sailpoint.com/patents>

Restricted Rights Legend. All rights are reserved. No part of this document may be published, distributed, reproduced, publicly displayed, used to create derivative works, or translated to another language, without the prior written consent of SailPoint Technologies. The information contained in this document is subject to change without notice.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c)(1) and (c)(2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Regulatory/Export Compliance. The export and re-export of this software is controlled for export purposes by the U.S. Government. By accepting this software and/or documentation, licensee agrees to comply with all U.S. and foreign export laws and regulations as they relate to software and related documentation. Licensee will not export or re-export outside the United States software or documentation, whether directly or indirectly, to any Prohibited Party and will not cause, approve or otherwise intentionally facilitate others in so doing. A Prohibited Party includes: a party in a U.S. embargoed country or country the United States has named as a supporter of international terrorism; a party involved in proliferation; a party identified by the U.S. Government as a Denied Party; a party named on the U.S. Department of Commerce’s Entity List in Supplement No. 4 to 15 C.F.R. § 744; a party prohibited from participation in export or re-export transactions by a U.S. Government General Order; a party listed by the U.S. Government’s Office of Foreign Assets Control as ineligible to participate in transactions subject to U.S. jurisdiction; or any party that licensee knows or has reason to know has violated or plans to violate U.S. or foreign export laws or regulations. Licensee shall ensure that each of its software users complies with U.S. and foreign export laws and regulations as they relate to software and related documentation.

## **Access Fulfillment for Unmanaged Business Resources in File Access Manager**

IdentityIQ File Access Manager supports automatic access fulfillment for unmanaged business resources, using a user script.

The stages of approval remain as they are for managed business resources.

See the chapter on fulfillment in the IdentityIQ File Access Manager Administrator Guide.

# Configuration

Manual and automated fulfillment options on the access request template

Fulfillment field	Managed BRs	Unmanaged BRs
None	No action	No action
Fulfill Access Request	Fulfillment processed automatically by the system	Manual fulfillment process. The user performing the fulfillment has to mark the task as done.
Execute Custom Script	Fulfillment processed automatically by the system	Fulfillment processed automatically, calling the custom script for each BR.

## How to Set Up Fulfillment Using a Script

- Open the Access Request Template  
*Access Requests >> Configuration >> Manage Access Requests Templates*
- Double click an existing template to edit it, or click **New** to create a new template.
- In the **Fulfillment** field, select **Execute Custom Script**.

**Access Request Template**

Choose the review process, and the application and identity collectors to use it in access request processes

Name:

Review Process:  [\(Create a new Review Process\)](#)

Objects: Filter:  Filter:

Available Chosen

Set maximum duration for access requests handling

Duration:  Days

Fulfillment:

None

None

Fulfill Access Requests

Execute Custom Script

This will assume the script, named "Custom-Fulfillment.ps1" is in the required folder.

4. In the Access Certification campaign management:  
*Compliance > Access Certification > Campaign Management > Manage Access Requests Templates*
5. Edit an existing campaign, or click **+New Campaign**.
6. When you get to the **Summary** tab, open the **Fulfillment** option, by clicking **Edit**.
7. Select **Fulfill Permissions Revoke Requests**.
8. In the **Fulfillment Options**, select **Execute Custom Script**.

Fulfillment Process

None  Fulfill Permissions Revoke Requests

You can update the review process list in the Administrative Client and click the Refresh button [Refresh](#)

Access revoke request should be reviewed

**Fulfillment Options:**

Manual Fulfillment Review Process

Manual Fulfillment Review Process  with one-step review process for manual fulfillment.

Execute Custom Script

All

**Review Process** ⓘ

By Data Owner

**Type of Account**

User Account

**Default Reviewer(s) \***

Search for a user

For managed BRs, this campaign will automatically revoke the permissions from users selected

Users from BRs that are not managed that were selected to revoke permission will be processed using this user script, as described above.

### Script location

The user script has to be stored in the folder %SAILPOINT\_HOME%\%SAILPOINT\_APP\_NAME%\ScheduledTaskHandler

There is a sample script in that folder that comes with the installation package.

## Script sample and input / output variables

```
<#
.SYNOPSIS
Custom-Fulfillment.ps1 - Changes a file or folder ACL.

.DESCRIPTION
This script will modify the security descriptor of a specified item, such as a file or a
folder, to match the values that have been supplied.

.INPUTS
This script gets a list of parameters as described below
.OUTPUTS
The script will return a string or integer value:
    Success = 0
    Error != 0

.NOTES
Written by: SailPoint Technologies
#>
# Main
param (
    [bool]    $isRollback,          # Determines whether this is a rollback action or not
    [string]$actionType,          # The action type performed (AddPermission, RemovePer-
mission, AddUserToGroup, RemoveUserFromGroup)
    [string]$requestedBy,        # The user that created the access request
    [string]$applicationName,    # The request application name
    [string]$applicationType,    # The request application type (e.g. FILES MINI-FILTER)
    [string]$resourceFullPath,   # The full path of the resource (will be empty in case
the request is performed on a group)
    [string]$permissionType,    # The type of permission to add/remove
    [string]$accessRequestID,    # The access request ID
    [string]$campaignName,      # The campaign name (will be empty if the request
hasn't been created from a campaign)
    [string]$filterView,        # The campaign filter view type (FineGrained, User-
sAndRoles, Users, FineGrainedWithEveryone)
    # User fields in which the action is performed on (will be empty if the action is not
performed on a user)
    [string]$user,              # The user name
    [string]$userFullName,      # The user full name
    [string]$userUID,           # The user unique identifier
    [string]$userDisplayName,   # The user display name
    [string]$userPrincipalName, # The user principal name
    [string]$userType,          # The user entity type name
    [string]$userField1,        # The user enrichment fields (1 - 32)
    [string]$userField2,
    # (This goes on for a while)
    [string]$userField31,
    [string]$userField32,
    # Group fields in which the action is performed on (will be empty if the action is
not performed on a group)
    [string]$group,             # The group name
    [string]$groupUID,          # The group unique identifier
    [string]$groupType,         # The group type
    [string]$groupDomain,      # The group domain
```

```
    [string]$groupField1,    # The group enrichment fields (1 - 10)
    [string]$groupField2,
    [string]$groupField3,
# and so on...
    [string]$groupField9,
    [string]$groupField10
)
#####
# Start writing your code from here #
#####
if (($actionType -eq 'AddPermission') -and ($user -ne '')) {
    if ($isRollback -eq $false) {
        # Adds permission to the specified user
    }
    else {
        # Handle rollback for the current action
    }
}
if ($success) {
    return 0; # Success
}
else {
    return 1; # Failure
}
```

## Returned Values

The following codes are returned from the script:

**0**

Success

**1-7**

Error values