Temporary Administrator Access on Windows with FileWave

Overview

This recipe uses a FileWave Fileset to add an existing dedicated local Windows account to the built-in Administrators group for a limited period. The default window is 10 minutes. A scheduled task running as Local System removes the membership when the window expires.

Use the dedicated account to answer a UAC credential prompt when approved support work requires administrator rights. The script does not create, enable, unlock, or set the password for the account.

> ⚠ Temporary local administrator access carries real risk. During the access window, an administrator can disable the cleanup task, change protected files, create another administrator, install persistent software, or interfere with management. Removing group membership also does not terminate processes that were already elevated or invalidate access tokens that Windows already issued. Use a dedicated support account, keep the window short, retain a separate recovery administrator, and investigate any failed cleanup immediately.

Requirements

- FileWave Central and enrolled Windows clients.
- An existing, enabled local Windows account with the same name on each target device. The example uses `Filewave`.
- The account is normally a standard user and is not used as a permanent administrator or as the user's daily sign-in account.
- 64-bit Windows with Windows PowerShell 5.1, the `Microsoft.PowerShell.LocalAccounts` module, and the `ScheduledTasks` module. Do not use this workflow on a domain controller.
- A separate protected administrator or recovery method that does not depend on this workflow.
- A separate managed method for provisioning and rotating the dedicated account password. This workflow does not need the password, so do not add it to this script, Fileset, or its Launch Arguments.
- Treat this as an approved, device-specific privilege lease. Do not assign the Fileset broadly or allow another workflow to add the same account to Administrators while a lease is active.
- A common account name must not mean a shared fleet-wide password. Prefer unique per-device credentials, protect retrieval, rotate them through a separate managed process, and retain an approval and audit record for each request.

Download

Item	Purpose
Grant-FileWave-Temporary-Admin.ps1	Activation script that grants or renews temporary administrator membership and registers cleanup as Local System.

How the script works

1. It relaunches in native 64-bit Windows PowerShell 5.1 when needed, verifies Local System identity, confirms the required commands are available, and resolves exactly one enabled local account.
2. It stores enforcement artifacts beneath the protected Local System profile instead of the user-writable `C:\ProgramData\FileWave` tree. The script rejects reparse points and hard-linked artifacts, applies protected ACLs owned by Local System, and permits local administrators read-only access.
3. It identifies the account and Administrators group by SID. If the account is already a direct administrator and no valid script-owned state exists, it treats that membership as permanent and leaves it unchanged.
4. It serializes all grants, renewals, and cleanup operations with one global workspace mutex. Each approved run receives a new lease ID and UTC expiration.
5. Before adding membership, it writes state and the generated cleanup script with create-new, write-through, atomic replacement; validates their bytes and ACLs; and registers the cleanup task as Local System.
6. It reads the scheduled task back and verifies its action, Local System principal, highest run level, expiration, startup and daily triggers, repetition settings, retry policy, battery behavior, and enabled state before changing membership.
7. It re-resolves the exact account SID immediately before the transition, adds membership only after cleanup enforcement is ready, and performs a final holistic readback of membership, lease state, artifacts, and task configuration.
8. The task starts at expiration, retries every five minutes for seven days, then continues daily and at system startup until cleanup succeeds. Cleanup verifies membership absence before removing enforcement artifacts.
9. If the workflow owns temporary membership but cannot secure or verify automatic cleanup, it attempts to remove that membership immediately and exits with code `1`. Failed removal preserves the task and evidence for retry and investigation.

Create the Fileset

1. Download `Grant-FileWave-Temporary-Admin.ps1`.
2. In FileWave Central, open Filesets and create an empty Fileset named Temporary Administrator Access - Windows.
3. Select the Fileset and open the Scripts view.
4. Import the PowerShell file as an Activation script.
5. Open the script properties and add these Launch Arguments in order:

Position	Value	Purpose
1	`-UserName`	Names the first parameter.
2	`Filewave`	The dedicated local account. Allowed characters are letters, numbers, periods, underscores, and hyphens, up to 20 characters.
3	`-DurationMinutes`	Names the duration parameter.
4	`10`	Temporary access duration. The script accepts 1 through 1,440 minutes; use the shortest operational window.

Info - Temporary Admin Access - Win : Grant-FileWave-Temporary-Admin.ps1

Grant-FileWave-Temporary-Admin.p

Release Control

Kind: File

Created: Fri Aug 7 2026 08:55 am

Modified: Fri Aug 7 2026 08:55 am

Permissions (macOS only) | ACLs | Verification | Executable | Flags

Execution Control

- [x] Execute once when activated
  - ( ) Interactive (ignored in non Windows™ clients)
  - (•) Non-interactive (background)
  - [x] Wait for executable to finish

Wait for: 5 Minutes

Launch Arguments | Environment Variables

-UserName
Filewave
-DurationMinutes
10

\+ -

The values of the command line arguments are set just before the script execution.
To use an inventory field value to set a command line argument value, use the syntax %FIELD_NAME%.
For instance: **foo-%asset_tag%**

Note: Log files will be collected for synchronous non-interactive scripts only

Apply