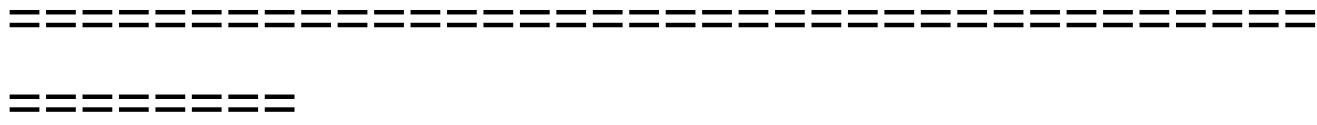


# CATEGORY: FileSystem



File system operations directly impact data integrity, storage management, compliance, and security. These SOPs ensure every file and folder action performed through RDAM Script Wizard is controlled, auditable, and aligned with enterprise standards.

## SOP 1 – Get File or Folder Details

**Script Name:** Get File or Folder Details **Category:** FileSystem **Version:** 1.0 **Approved By:** IT Operations / Security

### 1. Purpose

This script retrieves detailed metadata about a file or folder, supporting troubleshooting, inventory, compliance checks, and forensic analysis.

### 2. Scope

- Local or remote file systems (if supported)
- NTFS volumes
- Used by IT operations, security, and support teams

### 3. Definitions

- **Metadata:** Attributes such as size, timestamps, permissions, owner.
- **NTFS Attributes:** Read-only, hidden, system, archive, etc.

### 4. Preconditions

- Operator must have read access to the target path.
- Path must exist.
- System must support NTFS metadata retrieval.

## **5. Required Inputs**

- File or folder path

## **6. Procedure Steps**

### **1. Input Collection**

- Wizard prompts for path.
- Validate path format.

### **2. Existence Check**

- Confirm file or folder exists.
- If not found, abort and log.

### **3. Metadata Retrieval**

- Retrieve:
  - Name
  - Full path
  - Type (file/folder)
  - Size (if file)
  - Creation time
  - Last modified time
  - Last accessed time
  - Owner
  - Permissions (ACL summary)

### **4. Output Formatting**

- Present metadata in structured format.

### **5. Logging**

- Log path, operator, timestamp.

## **7. Expected Output**

- Detailed metadata for the file or folder.

## **8. Post-Execution Validation**

- Operator may verify using File Explorer or PowerShell.

## **9. Error Handling**

- Path not found
- Access denied
- Unsupported file system

## **10. Security Considerations**

- Metadata may reveal sensitive information; restrict access.
- Avoid inspecting paths outside approved scope.

## **11. Audit Logging Requirements**

- Operator ID
- Path
- Timestamp
- Success/Failure

## **12. Organizational Benefit Statement**

This script provides a consistent, auditable method for retrieving file system metadata, supporting troubleshooting, compliance, and forensic workflows.

# **SOP 2 – Copy File or Folder**

**Script Name:** Copy File or Folder **Category:** FileSystem

## **1. Purpose**

This script copies a file or folder from a source path to a destination path, supporting migrations, backups, and operational workflows.

## **2. Scope**

- Local or remote file systems
- NTFS volumes
- Used by operations, support, and engineering teams

## **3. Definitions**

- **Copy Operation:** Duplicate file/folder to new location.
- **Overwrite:** Replace existing files if allowed.

## **4. Preconditions**

- Operator must have read access to source and write access to destination.
- Source path must exist.
- Destination path must be valid.

## **5. Required Inputs**

- Source path
- Destination path
- Optional: Overwrite flag

## **6. Procedure Steps**

### **1. Input Collection**

- Wizard prompts for source, destination, overwrite flag.

### **2. Source Validation**

- Confirm source exists.
- Determine if file or folder.

### **3. Destination Validation**

- Confirm destination directory exists or can be created.
- If file exists and overwrite disabled, abort.

### **4. Copy Operation**

- Perform copy using appropriate API.
- Preserve timestamps if supported.

### **5. Post-Copy Verification**

- Confirm destination file/folder exists.
- Validate size and attributes match.

### **6. Logging**

- Log source, destination, operator, timestamp.

## **7. Expected Output**

- Confirmation of successful copy.

## **8. Post-Execution Validation**

- Operator may verify file integrity manually.

## **9. Error Handling**

- Source not found
- Access denied
- Destination invalid
- File in use

## **10. Security Considerations**

- Copying sensitive files must follow data handling policies.
- Avoid copying system files without authorization.

## **11. Audit Logging Requirements**

- Operator ID
- Source
- Destination
- Timestamp

## **12. Organizational Benefit Statement**

This script ensures file copies are performed safely and consistently, reducing risk of data loss or misplacement.

# **SOP 3 – Delete File or Folder**

**Script Name:** Delete File or Folder **Category:** FileSystem

## **1. Purpose**

This script deletes a file or folder from the file system, supporting cleanup, decommissioning, and storage management.

## **2. Scope**

- Local or remote file systems
- NTFS volumes
- Used by operations, security, and support teams

### **3. Definitions**

- **Deletion:** Permanent removal of file/folder.

### **4. Preconditions**

- Operator must have delete permissions.
- Path must exist.
- Deletion must be authorized.

### **5. Required Inputs**

- Path to delete
- Optional: Recursive flag (for folders)

### **6. Procedure Steps**

#### **1. Input Collection**

- Wizard prompts for path and recursive flag.

#### **2. Existence Check**

- Confirm file/folder exists.

#### **3. Safety Checks**

- Prevent deletion of system directories.
- Validate path is within approved scope.

#### **4. Deletion Operation**

- Delete file or folder.
- If folder and recursive enabled, delete all contents.

#### **5. Post-Deletion Verification**

- Confirm path no longer exists.

#### **6. Logging**

- Log path, operator, timestamp.

### **7. Expected Output**

- Confirmation of deletion.

### **8. Post-Execution Validation**

- Operator may verify manually.

## 9. Error Handling

- Access denied
- File in use
- Path not found

## 10. Security Considerations

- Deleting files may impact applications; ensure approvals.
- Avoid deleting logs or audit data without authorization.

## 11. Audit Logging Requirements

- Operator ID
- Path
- Timestamp

## 12. Organizational Benefit Statement

This script ensures file deletions are performed safely and with full accountability, reducing risk of accidental data loss.

# SOP 4 – Get Folder Size (Recursive)

**Script Name:** Get Folder Size (Recursive) **Category:** FileSystem

## 1. Purpose

This script calculates the total size of a folder and all its subfolders, supporting storage analysis, cleanup planning, and capacity management.

## 2. Scope

- Local or remote NTFS file systems
- Used by operations, engineering, and storage teams

## 3. Definitions

- **Recursive Enumeration:** Traversing all subfolders and files.
- **Folder Size:** Sum of all file sizes within folder tree.

## 4. Preconditions

- Operator must have read access to all subfolders.

- Path must exist.

## 5. Required Inputs

- Folder path

## 6. Procedure Steps

### 1. Input Collection

- Wizard prompts for folder path.

### 2. Existence Check

- Confirm folder exists.

### 3. Recursive Enumeration

- Traverse folder tree.
- Sum file sizes.

### 4. Output Formatting

- Present:
  - Total size
  - File count
  - Folder count

### 5. Logging

- Log path, operator, timestamp.

## 7. Expected Output

- Total folder size in bytes/MB/GB.

## 8. Post-Execution Validation

- Operator may compare with File Explorer.

## 9. Error Handling

- Access denied
- Path not found
- Very large folder (performance warning)

## 10. Security Considerations

- Avoid scanning sensitive directories without authorization.

## 11. Audit Logging Requirements

- Operator ID
- Path
- Timestamp

## 12. Organizational Benefit Statement

This script provides accurate folder size calculations, supporting storage planning and cleanup initiatives.

# SOP 5 – Copy Folder (Recursive)

**Script Name:** Copy Folder (Recursive) **Category:** FileSystem

## 1. Purpose

This script copies an entire folder tree, including all subfolders and files, supporting migrations, backups, and data reorganization.

## 2. Scope

- Local or remote NTFS file systems
- Used by operations and engineering teams

## 3. Definitions

- **Recursive Copy:** Copying all contents of a folder tree.

## 4. Preconditions

- Operator must have read access to source and write access to destination.
- Source folder must exist.

## 5. Required Inputs

- Source folder path
- Destination folder path
- Optional: Overwrite flag

## 6. Procedure Steps

### 1. Input Collection

- Wizard prompts for source, destination, overwrite flag.

## **2. Source Validation**

- Confirm folder exists.

## **3. Destination Validation**

- Create destination folder if needed.

## **4. Recursive Copy Operation**

- Copy all files and subfolders.
- Preserve timestamps if supported.

## **5. Post-Copy Verification**

- Confirm folder structure matches.
- Validate file counts and sizes.

## **6. Logging**

- Log source, destination, operator, timestamp.

## **7. Expected Output**

- Confirmation of successful recursive copy.

## **8. Post-Execution Validation**

- Operator may verify manually.

## **9. Error Handling**

- Access denied
- File in use
- Destination invalid

## **10. Security Considerations**

- Copying sensitive data must follow data handling policies.

## **11. Audit Logging Requirements**

- Operator ID
- Source
- Destination
- Timestamp

## **12. Organizational Benefit Statement**

This script ensures large folder copies are performed safely and consistently, reducing risk during migrations and reorganizations.

## **SOP 6 – Find Large Files**

**Script Name:** Find Large Files **Category:** FileSystem

### **1. Purpose**

This script identifies files exceeding a specified size threshold, supporting storage cleanup, capacity planning, and forensic review.

### **2. Scope**

- Local or remote NTFS file systems
- Used by operations, security, and engineering teams

### **3. Definitions**

- **Large File:** File exceeding defined size threshold (e.g., >100 MB).

### **4. Preconditions**

- Operator must have read access to target folder.
- Path must exist.

### **5. Required Inputs**

- Folder path
- Size threshold (MB/GB)

### **6. Procedure Steps**

#### **1. Input Collection**

- Wizard prompts for folder and size threshold.

#### **2. Folder Validation**

- Confirm folder exists.

#### **3. Recursive Enumeration**

- Traverse folder tree.
- Identify files exceeding threshold.

#### **4. Output Formatting**

- Present:
  - File path
  - Size
  - Last modified time

#### **5. Logging**

- Log folder, threshold, operator.

### **7. Expected Output**

- List of large files.

### **8. Post-Execution Validation**

- Operator may manually inspect files.

### **9. Error Handling**

- Access denied
- Invalid threshold
- Path not found

### **10. Security Considerations**

- Large files may contain sensitive data; handle appropriately.

### **11. Audit Logging Requirements**

- Operator ID
- Folder
- Threshold
- Timestamp

### **12. Organizational Benefit Statement**

This script helps identify storage hogs, improving capacity planning and reducing unnecessary disk usage.

## **SOP 7 – Remove Old Files**

**Script Name:** Remove Old Files **Category:** FileSystem

# **1. Purpose**

This script deletes files older than a specified age threshold, supporting cleanup, retention enforcement, and storage optimization.

## **2. Scope**

- Local or remote NTFS file systems
- Used by operations, compliance, and engineering teams

## **3. Definitions**

- **Old File:** File whose last modified time exceeds threshold (e.g., 30/60/90 days).

## **4. Preconditions**

- Operator must have delete permissions.
- Path must exist.
- Deletion must align with retention policies.

## **5. Required Inputs**

- Folder path
- Age threshold (days)
- Optional: Recursive flag

## **6. Procedure Steps**

### **1. Input Collection**

- Wizard prompts for folder, age threshold, recursive flag.

### **2. Folder Validation**

- Confirm folder exists.

### **3. Recursive Enumeration**

- Traverse folder tree.
- Identify files older than threshold.

### **4. Deletion Operation**

- Delete identified files.
- Skip files in use.

### **5. Post-Deletion Verification**

- Confirm files removed.

## 6. Logging

- Log folder, threshold, count of deleted files, operator.

## 7. Expected Output

- Summary of deleted files.

## 8. Post-Execution Validation

- Operator may verify manually.
- Compliance team may validate retention alignment.

## 9. Error Handling

- Access denied
- Path not found
- Invalid threshold

## 10. Security Considerations

- Deleting files may impact applications; ensure approvals.
- Must comply with data retention policies.

## 11. Audit Logging Requirements

- Operator ID
- Folder
- Threshold
- Deleted file count
- Timestamp

## 12. Organizational Benefit Statement

This script enforces data hygiene and storage efficiency by removing outdated files in a controlled, auditable manner.