===
=========

CATEGORY: WindowsServicesAdvanced

===
=========

Advanced service operations directly affect system reliability, service chaining, recovery behavior, and security posture. These SOPs ensure every advanced service-related action performed through RDAM Script Wizard is **controlled**, **auditable**, and aligned with **enterprise operational and security standards**.

SOP 1 – Get Service Dependencies

Script Name: Get Service Dependencies **Category:** WindowsServicesAdvanced **Version:** 1.0 **Approved By:** IT Operations / Engineering

1. Purpose

This script retrieves the list of services that a specified service depends on, supporting troubleshooting, configuration validation, and service-startup analysis.

2. Scope

- Windows servers and workstations
- All service types

3. Definitions

- **Dependencies:** Services required for another service to start.

4. Preconditions

- Operator must have permission to query service configuration.
- Service must exist.

5. Required Inputs

- Service name

6. Procedure Steps

1. Input Collection
    - Wizard prompts for service name.
2. Service Resolution
    - Identify matching service.
3. Dependency Retrieval
    - Extract list of required services.
4. Output Formatting
    - Present structured dependency list.
5. Logging
    - Log service name, operator, timestamp.

7. Expected Output

- List of services required by the specified service.

8. Post-Execution Validation

- Operator may verify via `Get-Service -RequiredServices`.

9. Error Handling

- Service not found
- Access denied
- No dependencies

10. Security Considerations

- Dependency data may reveal system architecture.

11. Audit Logging Requirements

- Operator ID
- Service name
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for retrieving service dependencies, supporting troubleshooting and configuration management.

SOP 2 – Get Dependent Services

Script Name: Get Dependent Services **Category:** WindowsServicesAdvanced

1. Purpose

This script retrieves the list of services that depend on a specified service, supporting impact analysis, troubleshooting, and change-control planning.

2. Scope

- Windows servers and workstations
- All service types

3. Definitions

- **Dependent Services:** Services that require the specified service to run.

4. Preconditions

- Operator must have permission to query service configuration.
- Service must exist.

5. Required Inputs

- Service name

6. Procedure Steps

1. Input Collection
    - Wizard prompts for service name.
2. Service Resolution
    - Identify matching service.
3. Dependent Retrieval
    - Extract list of services that rely on the specified service.
4. Output Formatting
    - Present structured dependent-service list.

5. Logging
    - Log service name, operator, timestamp.

7. Expected Output

- List of services dependent on the specified service.

8. Post-Execution Validation

- Operator may verify via `Get-Service -DependentServices`.

9. Error Handling

- Service not found
- Access denied
- No dependent services

10. Security Considerations

- Dependent-service data may reveal system architecture.

11. Audit Logging Requirements

- Operator ID
- Service name
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for retrieving dependent services, supporting impact analysis and change management.

SOP 3 – Set Service Recovery Options

Script Name: Set Service Recovery Options **Category:** WindowsServicesAdvanced

1. Purpose

This script configures service recovery actions (restart, run program, reboot) to improve system resilience and reduce downtime.

2. Scope

- Windows servers and workstations
- Services capable of recovery configuration

3. Definitions

- **Recovery Action:** Automatic response to service failure.

4. Preconditions

- Operator must have administrative rights.
- Service must exist.
- Action must be authorized.

5. Required Inputs

- Service name
- First failure action
- Second failure action
- Subsequent failure action
- Optional: Reset fail count (days)
- Optional: Restart delay

6. Procedure Steps

1. Input Collection
    - Wizard prompts for recovery parameters.
2. Validation
    - Confirm service exists.
    - Validate action types.
3. Apply Recovery Configuration
    - Set failure actions.
    - Set reset period.
    - Set restart delay.
4. Post-Configuration Verification
    - Confirm recovery settings applied.
5. Logging
    - Log service name, actions, operator, timestamp.

7. Expected Output

- Confirmation of recovery configuration.

8. Post-Execution Validation

- Operator may verify via `sc qfailure`.

9. Error Handling

- Access denied
- Invalid action
- Service not found

10. Security Considerations

- Misconfigured recovery actions may cause service loops.

11. Audit Logging Requirements

- Operator ID
- Service name
- Recovery actions
- Timestamp

12. Organizational Benefit Statement

This script ensures recovery settings are applied safely and consistently, supporting system resilience and uptime.

SOP 4 – Get Service Failure Actions

Script Name: Get Service Failure Actions **Category:** WindowsServicesAdvanced

1. Purpose

This script retrieves the configured failure actions for a service, supporting troubleshooting, compliance, and operational analysis.

2. Scope

- Windows servers and workstations
- All services with recovery settings

3. Definitions

- **Failure Action:** Automatic response to service failure.

4. Preconditions

- Operator must have permission to query service configuration.

5. Required Inputs

- Service name

6. Procedure Steps

1. Input Collection
    - Wizard prompts for service name.
2. Service Resolution
    - Identify matching service.
3. Retrieve Failure Actions
    - Extract:
        - First failure action
        - Second failure action
        - Subsequent failure action
        - Reset period
        - Restart delay
4. Output Formatting
    - Present structured failure-action summary.
5. Logging
    - Log service name, operator, timestamp.

7. Expected Output

- Detailed failure-action configuration.

8. Post-Execution Validation

- Operator may verify via `sc qfailure`.

9. Error Handling

- Service not found
- Access denied
- No failure actions configured

10. Security Considerations

- Failure-action data may reveal operational behavior.

11. Audit Logging Requirements

- Operator ID
- Service name
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for retrieving failure actions, supporting troubleshooting and compliance.

SOP 5 – Get Service Logon Account

Script Name: Get Service Logon Account **Category:** WindowsServicesAdvanced

1. Purpose

This script retrieves the logon account used by a service, supporting troubleshooting, security reviews, and configuration validation.

2. Scope

- Windows servers and workstations
- Services running under system or custom accounts

3. Definitions

- **Logon Account:** Identity used to run the service.

4. Preconditions

- Operator must have permission to query service configuration.

5. Required Inputs

- Service name

6. Procedure Steps

1. Input Collection
    - Wizard prompts for service name.
2. Service Resolution
    - Identify matching service.
3. Retrieve Logon Account
    - Extract service logon identity.
4. Output Formatting
    - Present structured logon-account summary.
5. Logging
    - Log service name, operator, timestamp.

7. Expected Output

- Service logon account information.

8. Post-Execution Validation

- Operator may verify via `services.msc`.

9. Error Handling

- Service not found
- Access denied

10. Security Considerations

- Logon accounts may reveal privileged identities.

11. Audit Logging Requirements

- Operator ID
- Service name
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for retrieving service logon accounts, supporting security and configuration management.

SOP 6 – Set Service Logon Account

Script Name: Set Service Logon Account **Category:** WindowsServicesAdvanced

1. Purpose

This script updates the logon account for a service, supporting security hardening, account rotation, and configuration changes.

2. Scope

- Windows servers and workstations
- Services running under system or custom accounts

3. Definitions

- **Logon Account Update:** Changing the identity used to run a service.

4. Preconditions

- Operator must have administrative rights.
- New account must exist.
- Password must be valid.
- Action must be authorized.

5. Required Inputs

- Service name
- New logon account
- Password

6. Procedure Steps

1. Input Collection
    - Wizard prompts for service name, account, and password.
2. Validation
    - Confirm service exists.

   - Confirm account exists.

3. Update Operation
   - Apply new logon credentials.
4. Post-Update Verification
   - Confirm service configuration updated.
5. Logging
   - Log service name, account (not password), operator, timestamp.

7. Expected Output

- Confirmation of logon account update.

8. Post-Execution Validation

- Operator may verify via `services.msc`.

9. Error Handling

- Access denied
- Invalid credentials
- Service not found

10. Security Considerations

- Passwords must never be logged.
- Changing logon accounts may affect service permissions.

11. Audit Logging Requirements

- Operator ID
- Service name
- New account
- Timestamp

12. Organizational Benefit Statement

This script ensures service logon account changes are performed safely and with full accountability, supporting security and configuration management.