==========

==========

CATEGORY: WindowsProcesses

==========

==========

Process operations directly affect system stability, performance, diagnostics, and security posture. These SOPs ensure every process-related action performed through RDAM Script Wizard is **controlled**, **auditable**, and aligned with enterprise operational and security standards.

SOP 1 – List Running Processes

Script Name: List Running Processes **Category:** WindowsProcesses **Version:** 1.0 **Approved By:** IT Operations / Engineering

1. Purpose

This script retrieves all running processes, supporting diagnostics, performance analysis, and troubleshooting.

2. Scope

- Windows servers and workstations
- All user and system processes

3. Definitions

- **Process:** An executing program instance.

4. Preconditions

- Operator must have permission to query process information.

5. Required Inputs

- Optional: Process name filter

6. Procedure Steps

1. Input Collection

- Wizard prompts for optional filter.

2. Process Enumeration
    - Retrieve all running processes.
3. Attribute Retrieval
    - Extract:
        - Process name
        - PID
        - CPU usage
        - Memory usage
        - User account
4. Output Formatting
    - Present structured process list.
5. Logging
    - Log filter, operator, timestamp.

7. Expected Output

- List of running processes with key attributes.

8. Post-Execution Validation

- Operator may verify via Task Manager or `Get-Process`.

9. Error Handling

- Access denied
- Invalid filter

10. Security Considerations

- Process data may reveal sensitive application activity.

11. Audit Logging Requirements

- Operator ID
- Filter used
- Timestamp

12. Organizational Benefit Statement

This script provides a consistent, auditable method for enumerating processes, supporting diagnostics and performance analysis.

SOP 2 – Get Process Details

Script Name: Get Process Details **Category:** WindowsProcesses**

1. Purpose

This script retrieves detailed information about a specific process, supporting troubleshooting, diagnostics, and forensic analysis.

2. Scope

- Windows servers and workstations
- All running processes

3. Definitions

- **Process Details:** Includes path, owner, CPU, memory, and handles.

4. Preconditions

- Operator must have permission to query process details.
- Process must exist.

5. Required Inputs

- Process name or PID

6. Procedure Steps

1. Input Collection
    - Wizard prompts for process name or PID.
2. Process Resolution
    - Identify matching process.
3. Attribute Retrieval
    - Extract:
        - Executable path
        - User account

        - CPU and memory usage
        - Handle count
        - Start time
4. Output Formatting
    - Present structured process details.
5. Logging
    - Log process identifier, operator, timestamp.

7. Expected Output

- Detailed process information.

8. Post-Execution Validation

- Operator may verify via Task Manager.

9. Error Handling

- Process not found
- Access denied

10. Security Considerations

- Process paths may reveal sensitive application locations.

11. Audit Logging Requirements

- Operator ID
- Process name/PID
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for retrieving process details, supporting diagnostics and forensic workflows.

SOP 3 – Stop Process

Script Name: Stop Process **Category:** WindowsProcesses**

1. Purpose

This script terminates a running process, supporting troubleshooting, incident response, and resource recovery.

2. Scope

- Windows servers and workstations
- All user and system processes (with restrictions)

3. Definitions

- **Terminate:** Forcefully or gracefully stop a process.

4. Preconditions

- Operator must have administrative rights.
- Termination must be authorized.
- Process must exist.

5. Required Inputs

- Process name or PID

6. Procedure Steps

1. Input Collection
    - Wizard prompts for process name or PID.
2. Validation
    - Confirm process exists.
    - Prevent termination of critical system processes.
3. Termination Operation
    - Stop process gracefully if possible.
    - Force terminate if required and authorized.
4. Post-Termination Verification
    - Confirm process no longer running.
5. Logging
    - Log process identifier, operator, timestamp.

7. Expected Output

- Confirmation of process termination.

8. Post-Execution Validation

- Operator may verify via Task Manager.

9. Error Handling

- Access denied
- Process not found
- Critical process protection

10. Security Considerations

- Terminating processes may disrupt applications or services.

11. Audit Logging Requirements

- Operator ID
- Process name/PID
- Timestamp

12. Organizational Benefit Statement

This script ensures process termination is performed safely and with full accountability, supporting troubleshooting and incident response.

SOP 4 – Start Process

Script Name: Start Process **Category:** WindowsProcesses**

1. Purpose

This script starts a new process, supporting application launches, troubleshooting, and automation workflows.

2. Scope

- Windows servers and workstations
- Executable files and scripts

3. Definitions

- **Start Process:** Launching an executable or script.

4. Preconditions

- Operator must have permission to start processes.
- File path must exist.

5. Required Inputs

- Executable path
- Optional: Arguments

6. Procedure Steps

1. Input Collection
    - Wizard prompts for path and arguments.
2. Validation
    - Confirm file exists.
    - Confirm operator authorization.
3. Start Operation
    - Launch process with optional arguments.
4. Post-Start Verification
    - Confirm process is running.
5. Logging
    - Log executable path, operator, timestamp.

7. Expected Output

- Confirmation of process start.

8. Post-Execution Validation

- Operator may verify via Task Manager.

9. Error Handling

- File not found
- Access denied

- Invalid arguments

10. Security Considerations

- Starting processes may introduce security risks.

11. Audit Logging Requirements

- Operator ID
- Executable path
- Timestamp

12. Organizational Benefit Statement

This script ensures process startup is performed safely and consistently, supporting automation and troubleshooting.

SOP 5 – Get Process Modules

Script Name: Get Process Modules **Category:** WindowsProcesses**

1. Purpose

This script retrieves the modules (DLLs) loaded by a process, supporting troubleshooting, diagnostics, and forensic analysis.

2. Scope

- Windows servers and workstations
- All processes with accessible module lists

3. Definitions

- **Module:** A DLL or component loaded by a process.

4. Preconditions

- Operator must have permission to query process modules.
- Process must exist.

5. Required Inputs

- Process name or PID

6. Procedure Steps

1. Input Collection
    - Wizard prompts for process identifier.
2. Process Resolution
    - Identify matching process.
3. Module Enumeration
    - Retrieve list of loaded modules.
    - Extract:
        - Module name
        - File path
        - Memory address
4. Output Formatting
    - Present structured module list.
5. Logging
    - Log process identifier, operator, timestamp.

7. Expected Output

- List of modules loaded by the process.

8. Post-Execution Validation

- Operator may verify via Process Explorer.

9. Error Handling

- Access denied
- Process not found

10. Security Considerations

- Module paths may reveal sensitive application components.

11. Audit Logging Requirements

- Operator ID
- Process name/PID
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for retrieving process modules, supporting diagnostics and forensic workflows.

SOP 6 – Get Process Threads

Script Name: Get Process Threads **Category:** WindowsProcesses**

1. Purpose

This script retrieves thread information for a process, supporting diagnostics, performance analysis, and troubleshooting.

2. Scope

- Windows servers and workstations
- All processes with accessible thread lists

3. Definitions

- **Thread:** A unit of execution within a process.

4. Preconditions

- Operator must have permission to query process threads.
- Process must exist.

5. Required Inputs

- Process name or PID

6. Procedure Steps

1. Input Collection
    - Wizard prompts for process identifier.
2. Process Resolution
    - Identify matching process.
3. Thread Enumeration
    - Retrieve thread list.
    - Extract:
        - Thread ID

        - Start time
        - CPU usage
        - State
4. Output Formatting
    - Present structured thread list.
5. Logging
    - Log process identifier, operator, timestamp.

7. Expected Output

- Thread information for the specified process.

8. Post-Execution Validation

- Operator may verify via Process Explorer.

9. Error Handling

- Access denied
- Process not found

10. Security Considerations

- Thread data may reveal sensitive application behavior.

11. Audit Logging Requirements

- Operator ID
- Process name/PID
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for retrieving thread information, supporting diagnostics and performance analysis.