1. ==

CATEGORY: SystemInfo

==

System information operations support diagnostics, inventory, compliance, and troubleshooting. These SOPs ensure every system-information action performed through RDAM Script Wizard is **controlled**, **auditable**, and aligned with **enterprise operational standards**.

SOP 1 – Get System Information Summary

Script Name: Get System Information Summary **Category:** SystemInfo **Version:** 1.0 **Approved By:** IT Operations / Engineering

1. Purpose

This script retrieves a consolidated summary of key system information, supporting troubleshooting, inventory, and compliance validation.

2. Scope

- Windows servers and workstations
- Local or remote systems (if supported)

3. Definitions

- **System Summary:** High-level overview of hardware, OS, and environment.

4. Preconditions

- Operator must have permission to query system information.

5. Required Inputs

- None

6. Procedure Steps

1. Collect System Metadata

    - Computer name
    - Manufacturer
    - Model
    - Serial number (if available)

2. Collect OS Metadata
    - OS name
    - Version
    - Build number
    - Install date
    - System uptime
3. Collect Hardware Summary
    - CPU model and count
    - Total memory
    - Disk summary
4. Output Formatting
    - Present structured summary.
5. Logging
    - Log operator and timestamp.

7. Expected Output

- Consolidated system information summary.

8. Post-Execution Validation

- Operator may verify using `systeminfo`.

9. Error Handling

- Access denied
- WMI failure
- Missing components

10. Security Considerations

- System metadata may reveal sensitive asset information.

11. Audit Logging Requirements

- Operator ID
- Timestamp

12. Organizational Benefit Statement

This script provides a consistent, auditable method for retrieving system summaries, supporting troubleshooting and inventory workflows.

SOP 2 – Get Hardware Details

Script Name: Get Hardware Details **Category:** SystemInfo

1. Purpose

This script retrieves detailed hardware information, supporting diagnostics, capacity planning, and asset management.

2. Scope

- Windows servers and workstations
- CPU, memory, disk, and motherboard details

3. Definitions

- **Hardware Inventory:** Detailed enumeration of physical components.

4. Preconditions

- Operator must have permission to query hardware information.

5. Required Inputs

- None

6. Procedure Steps

1. CPU Details
    - Model
    - Core count
    - Logical processors
    - Clock speed
2. Memory Details

    - Total RAM
    - Memory modules (size, type, speed)
3. Disk Details
    - Physical disks
    - Capacity
    - Media type (SSD/HDD)
4. Motherboard Details
    - Manufacturer
    - Model
    - Serial number (if available)
5. Output Formatting
    - Present structured hardware inventory.
6. Logging
    - Log operator and timestamp.

7. Expected Output

- Detailed hardware inventory.

8. Post-Execution Validation

- Operator may verify using Device Manager or `Get-CimInstance`.

9. Error Handling

- Access denied
- WMI failure
- Missing hardware data

10. Security Considerations

- Hardware details may reveal sensitive asset identifiers.

11. Audit Logging Requirements

- Operator ID
- Timestamp

12. Organizational Benefit Statement

This script provides a reliable, auditable method for retrieving hardware details, supporting diagnostics and asset management.

SOP 3 – Get Operating System Details

Script Name: Get Operating System Details **Category:** SystemInfo

1. Purpose

This script retrieves detailed OS information, supporting troubleshooting, compliance, and lifecycle management.

2. Scope

- Windows servers and workstations
- OS version, build, edition, and configuration

3. Definitions

- **OS Build:** Specific release version of Windows.

4. Preconditions

- Operator must have permission to query OS information.

5. Required Inputs

- None

6. Procedure Steps

1. Retrieve OS Metadata
    - Name
    - Edition
    - Version
    - Build number
    - Install date
    - System root
2. Retrieve Licensing Metadata
    - Activation status

    - Product key channel (if available)
3. Output Formatting
    - Present structured OS details.
4. Logging
    - Log operator and timestamp.

7. Expected Output

- Detailed OS configuration and metadata.

8. Post-Execution Validation

- Operator may verify using `winver` or `systeminfo`.

9. Error Handling

- Access denied
- WMI failure

10. Security Considerations

- Licensing data may be sensitive.

11. Audit Logging Requirements

- Operator ID
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for retrieving OS details, supporting compliance and lifecycle planning.

SOP 4 – Get BIOS / Firmware Information

Script Name: Get BIOS / Firmware Information **Category:** SystemInfo

1. Purpose

This script retrieves BIOS or UEFI firmware information, supporting diagnostics, compliance, and hardware lifecycle management.

2. Scope

- Windows servers and workstations

- BIOS/UEFI firmware

3. Definitions

- **Firmware:** Low-level system software controlling hardware initialization.

4. Preconditions

- Operator must have permission to query firmware information.

5. Required Inputs

- None

6. Procedure Steps

1. Retrieve Firmware Metadata
    - Manufacturer
    - Version
    - Release date
    - BIOS mode (Legacy/UEFI)
2. Output Formatting
    - Present structured firmware details.
3. Logging
    - Log operator and timestamp.

7. Expected Output

- BIOS/UEFI firmware information.

8. Post-Execution Validation

- Operator may verify via `msinfo32`.

9. Error Handling

- Access denied
- Firmware data unavailable

10. Security Considerations

- Firmware version may reveal vulnerability exposure.

11. Audit Logging Requirements

- Operator ID
- Timestamp

12. Organizational Benefit Statement

This script provides a consistent, auditable method for retrieving firmware details, supporting diagnostics and lifecycle management.

SOP 5 – Get Environment Variables

Script Name: Get Environment Variables **Category:** SystemInfo

1. Purpose

This script retrieves system and user environment variables, supporting troubleshooting, configuration validation, and application diagnostics.

2. Scope

- Windows servers and workstations
- System-level and user-level variables

3. Definitions

- **Environment Variable:** Key-value pair used by applications and the OS.

4. Preconditions

- Operator must have permission to query environment variables.

5. Required Inputs

- Optional: Variable name filter

6. Procedure Steps

1. Input Collection
    - Wizard prompts for optional filter.
2. Variable Enumeration
    - Retrieve system variables.
    - Retrieve user variables.
3. Filtering

    - Apply name filter if provided.
4. Output Formatting
    - Present structured variable list.
5. Logging
    - Log filter, operator, timestamp.

7. Expected Output

- List of environment variables and values.

8. Post-Execution Validation

- Operator may verify using `set` or `Get-ChildItem Env:`.

9. Error Handling

- Access denied
- Invalid filter

10. Security Considerations

- Variables may contain sensitive paths or tokens.

11. Audit Logging Requirements

- Operator ID
- Filter used
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for retrieving environment variables, supporting troubleshooting and configuration validation.