==

CATEGORY: WMI

==

Windows Management Instrumentation (WMI) operations provide deep access to system configuration, hardware, software, and operational data. These SOPs ensure every WMI-related action performed through RDAM Script Wizard is **controlled**, **auditable**, and aligned with **enterprise operational and security standards**.

SOP 1 – Run WMI Query

Script Name: Run WMI Query **Category:** WMI **Version:** 1.0 **Approved By:** IT Operations / Engineering

1. Purpose

This script executes a WMI query (WQL) against a specified namespace, supporting diagnostics, inventory, and troubleshooting.

2. Scope

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

3. Definitions

- **WQL:** WMI Query Language, similar to SQL.
- **Namespace:** Logical grouping of WMI classes.

4. Preconditions

- Operator must have permission to query WMI.
- Namespace and class must exist.

5. Required Inputs

- Namespace
- WQL query

6. Procedure Steps

1. Input Collection
    - Wizard prompts for namespace and WQL query.
2. Validation
    - Confirm namespace exists.
    - Validate WQL syntax (basic checks).
3. Query Execution
    - Execute WQL query.
    - Retrieve results.
4. Output Formatting
    - Present structured query results.
5. Logging
    - Log namespace, query, operator, timestamp.

7. Expected Output

- WMI query results.

8. Post-Execution Validation

- Operator may verify using `Get-WmiObject` or `Get-CimInstance`.

9. Error Handling

- Invalid namespace
- Invalid WQL
- Access denied
- No results

10. Security Considerations

- WMI queries may expose sensitive system data.

11. Audit Logging Requirements

- Operator ID
- Namespace
- Query
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for executing WMI queries, supporting diagnostics and inventory workflows.

SOP 2 – List WMI Namespaces

Script Name: List WMI Namespaces **Category:** WMI

1. Purpose

This script enumerates available WMI namespaces, supporting discovery, troubleshooting, and administrative workflows.

2. Scope

- Windows servers and workstations
- Root and nested namespaces

3. Definitions

- **Namespace:** Container for WMI classes.

4. Preconditions

- Operator must have permission to query WMI.

5. Required Inputs

- None

6. Procedure Steps

1. Namespace Enumeration
    - Retrieve root namespaces.
    - Recursively enumerate child namespaces.
2. Output Formatting

    - Present structured namespace list.

3. Logging
    - Log operator and timestamp.

7. Expected Output

- List of WMI namespaces.

8. Post-Execution Validation

- Operator may verify via `wbemtest`.

9. Error Handling

- Access denied
- Namespace enumeration failure

10. Security Considerations

- Namespace structure may reveal system capabilities.

11. Audit Logging Requirements

- Operator ID
- Timestamp

12. Organizational Benefit Statement

This script provides a reliable, auditable method for discovering WMI namespaces, supporting troubleshooting and configuration analysis.

SOP 3 – List WMI Classes in Namespace

Script Name: List WMI Classes in Namespace **Category:** WMI

1. Purpose

This script retrieves all WMI classes within a specified namespace, supporting discovery, diagnostics, and administrative workflows.

2. Scope

- Windows servers and workstations
- Any WMI namespace

3. Definitions

- **WMI Class:** A schema defining system data or operations.

4. Preconditions

- Operator must have permission to query the namespace.
- Namespace must exist.

5. Required Inputs

- Namespace
- Optional: Class name filter

6. Procedure Steps

1. Input Collection
    - Wizard prompts for namespace and optional filter.
2. Validation
    - Confirm namespace exists.
3. Class Enumeration
    - Retrieve all classes in namespace.
    - Apply filter if provided.
4. Output Formatting
    - Present structured class list.
5. Logging
    - Log namespace, filter, operator, timestamp.

7. Expected Output

- List of WMI classes.

8. Post-Execution Validation

- Operator may verify via `wbemtest`.

9. Error Handling

- Namespace not found
- Access denied

- Invalid filter

10. Security Considerations

- Class names may reveal system architecture.

11. Audit Logging Requirements

- Operator ID
- Namespace
- Filter
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for enumerating WMI classes, supporting diagnostics and discovery.

SOP 4 – Get WMI Class Properties

Script Name: Get WMI Class Properties **Category:** WMI

1. Purpose

This script retrieves the properties of a WMI class, supporting schema discovery, troubleshooting, and development.

2. Scope

- Windows servers and workstations
- Any WMI class

3. Definitions

- **Property:** A data field defined in a WMI class.

4. Preconditions

- Operator must have permission to query WMI.
- Namespace and class must exist.

5. Required Inputs

- Namespace
- Class name

6. Procedure Steps

1. Input Collection
    - Wizard prompts for namespace and class name.
2. Validation
    - Confirm namespace exists.
    - Confirm class exists.
3. Property Enumeration
    - Retrieve class properties.
    - Extract:
        - Property name
        - Data type
        - Qualifiers
4. Output Formatting
    - Present structured property list.
5. Logging
    - Log namespace, class name, operator, timestamp.

7. Expected Output

- List of class properties and metadata.

8. Post-Execution Validation

- Operator may verify via `wbemtest`.

9. Error Handling

- Class not found
- Access denied
- Namespace invalid

10. Security Considerations

- Property metadata may reveal sensitive system structure.

11. Audit Logging Requirements

- Operator ID
- Namespace
- Class name
- Timestamp

12. Organizational Benefit Statement

This script provides a consistent, auditable method for retrieving WMI class properties, supporting development and troubleshooting.

SOP 5 – Invoke WMI Method

Script Name: Invoke WMI Method **Category:** WMI

1. Purpose

This script invokes a method on a WMI class or instance, supporting administrative automation, diagnostics, and advanced system operations.

2. Scope

- Windows servers and workstations
- Any WMI class with invokable methods

3. Definitions

- **WMI Method:** An operation defined on a WMI class.

4. Preconditions

- Operator must have administrative rights (for most methods).
- Namespace and class must exist.
- Method must exist and be safe to invoke.

5. Required Inputs

- Namespace
- Class name
- Method name
- Optional: Method parameters

6. Procedure Steps

1. Input Collection
   - Wizard prompts for namespace, class, method, and parameters.
2. Validation
   - Confirm class exists.
   - Confirm method exists.
   - Validate parameters.
3. Invocation
   - Execute method.
   - Capture return value and output parameters.
4. Output Formatting
   - Present structured method results.
5. Logging
   - Log namespace, class, method, operator, timestamp.

7. Expected Output

- Method execution results.

8. Post-Execution Validation

- Operator may verify via PowerShell or `wbemtest`.

9. Error Handling

- Method not found
- Invalid parameters
- Access denied
- Invocation failure

10. Security Considerations

- Some WMI methods can modify system state; ensure approvals.

11. Audit Logging Requirements

- Operator ID

- Namespace
- Class name
- Method name
- Timestamp

12. Organizational Benefit Statement

This script ensures WMI method invocation is performed safely and with full accountability, supporting automation and advanced system operations.