UPAS Function Card: PM Module (Patch Management)

This Function Card is designed to guide you through the core setup of the UPAS PM (Patch Management) module. The document is divided into two main parts: Part 1, "Basic Framework Setup," covers the necessary prerequisites for all functions. Part 2 provides several "Feature Implementation Examples" to walk you through the most common security policies.

Core Architecture: The 3-Tier Model

Before you begin, it is crucial to understand the UPAS 3-tier architecture, as its operational model differs from the common 2-tier (Console-Agent) products on the market.

- **Console:** The central hub for policy creation and data aggregation. Administrators issue commands from here.
- **Sensor:** A regional data center deployed within the client's network. It is responsible for high-frequency communication with and data collection from the Agents under its jurisdiction, as well as executing commands from the Console.
- **Agent:** Installed on each endpoint computer, responsible for executing commands from the Sensor and reporting its local status.

Operational Model: Policies from the Console are first synchronized to the Sensor, which then distributes them to its assigned Agents. This distributed architecture significantly reduces the load on the Console and allows for more flexible regional management. Therefore, all policy applications are completed within the **Agent > Sensor > Console** synchronization cycle.

Setup Logic

1. Define Scope → 2. Create Source Groups → 3. Set Scan Schedule → 4. Apply Policies

The setup logic is sequential: Define "who" to manage, "categorize" them into groups, set "how often" to check them, and finally, "apply policies" to them.

Four Steps for Basic Setup

Step 1: Define Management Scope - Who to Manage

1. **Path:** Go to Settings > Agent Settings > Agent Installation configuration.
2. **Purpose:** To define rules that determine which devices should be actively managed by UPAS.
3. **Method:** Use criteria such as "Device attribute," "Computer Name," to precisely filter the group of devices you wish to manage.
4. **Result:** Only devices that meet the criteria defined here will be considered "managed," making them subject to subsequent policies.

Step 2: Create & Manage Source Groups - Categorize Managed Devices

1. **Path:** Go to System > MAC Management > click the **gear icon (⚙)>** Group.
2. **Purpose:** To segment managed devices into different logical groups based on various attributes or needs.
3. **Method:** Create parent and child groups based on criteria like "IP Subnet," "Operating System," or "Department Name."
4. **Result:** A structured asset hierarchy (e.g., "Finance Dept - Windows PCs") is established, preparing for targeted policy application.

Step 3: Set Global Scan Schedule - How Often to Check

1. **Path:** Go to System Management > Patch Management (PM).
2. **Purpose:** To set a global, default scanning frequency.
3. **Method:** Click the **gear icon (⚙)** in the top-left corner and go to Inspection policy settings to configure how often Agents should report their status.(Frequency of inspections)
4. **Result:** A regular check-in rhythm is established, ensuring device statuses are updated periodically.

Step 4: Apply Policies to Groups - What to Check

1. **Path:** Go to **System > PM > PM settings > Group management policies tab.**
2. **Purpose:** To apply specific check rules to the device groups you created in Step 2.
3. **Method:** Select a **source group**, and in the policy tabs below, enable **"Monitoring"** or **"Blocking"** and choose the corresponding rule.
4. **Result:** Devices within the selected group will now be checked against your specified policies according to the schedule.

Part 1: Completion Check

Verification Item	**Status**
The **Basic Framework Setup** (Steps 1-4) has been fully completed.	☐

Part 2: Feature Implementation (1) - Windows Update Check

Use Case

Proactively audit all Windows computers within the organization to ensure they have the latest critical security updates installed, maintaining security compliance and defending against the latest threats.

Configuration Flow

Step 1: Configure KB Update Source and Schedule

1. **Path:** Go to System > PM, click the **gear icon (⚙)**, and navigate to the Windows Update Settings tab.
2. **Purpose:** To configure rules for the Console to synchronize with the Microsoft Update Catalog.
3. **Method:** Click the **"Update Settings"** button in the top-right corner to configure

the source, update time, scan range, and severity levels.

4. **Result:** The UPAS Console will now periodically and automatically fetch the latest Windows update information to be used as a baseline for checks.

Step 2: Create a "Required KB List" Rule

1. **Path:** Go to System > PM > click the **gear icon (⚙)** > Windows Update Settings > Patch Check Items.
2. **Purpose:** To create a check rule containing specific KB numbers.
3. **Method:** Click **"Add a Policy,"** select the desired updates to switch on Inspection from the KB list, and save the rule with a descriptive name (e.g., "Critical Security Updates - Aug 2025").
4. **Result:** A new check rule is created. You can now apply this rule to a target group (e.g., "Windows PCs") via the "Group Management Policy" page.

Part 2: Completion Check

Verification Item	Status
The **Windows Update Check** feature has been configured and verified successfully.	☐

Part 3: Feature Implementation (2) - Antivirus Check & Auto-Remediation

Use Case

Ensure all company computers have the designated antivirus software installed. For computers where it is missing, the system should automatically perform a silent installation without IT intervention, achieving automated security compliance.

Configuration Flow

Step 1: Create an "Antivirus Check" Rule

1. **Path:** Go to System > PM, click the **gear icon (⚙)**, and navigate to the Antivirus update Settings tab.
2. **Purpose:** To create a check and remediation rule for antivirus software.
3. **Method:** Click **"Add a Policy,"** find your company's standard antivirus software in the list, and select the check conditions.
4. **Result:** A basic check rule is created and can be applied to target groups for monitoring.

Step 2: Configure " Auto-patching " (Optional)

1. **Path:** Return to the rule you created in Step 1.
2. **Purpose:** To define an automatic action for the system to take when a device is non-compliant.
3. **Method:** For the designated antivirus software, **enable the " Auto-patching" switch**. Click the **" Uploading patch files "** tab to upload a silent installer package and any necessary command-line parameters.
4. **Result:** When a managed device is detected without the specified antivirus, the system will automatically install it.

Part 3: Completion Check

Verification Item	Status
The **Antivirus Check & Auto-patching** feature has been configured and verified successfully.	☐

Part 4: Feature Implementation (3) - Required Software Check & Auto-Install

Use Case

Ensure all managed computers have essential office software installed (e.g., 7-Zip, VPN client). For new employees or re-imaged computers, the system automatically deploys standard software to maintain a unified corporate software environment.

Configuration Flow

Step 1: Create a " Permit **Software " Rule**

1. **Path:** Go to System > PM, click the **gear icon (⚙)**, and navigate to the Permit Software Settings tab.
2. **Purpose:** To create a checklist rule for required software.
3. **Method:** Click **"Add a Policy,"** and within the rule page, click **"Add Permit Software"** and fill in the software details.
4. **Result:** A check rule containing specific required software is created and can be applied to target groups for monitoring.

Step 2: Configure "Auto-patching" (Optional)

1. **Path:** Return to the rule you created in Step 1.
2. **Purpose:** To define an automatic installation action for when a required application is missing.
3. **Method:** For the software item, **enable the "Auto-patching" switch**. Go to the **"Upload patch file for the permitted software installation."** page to manage the installer and silent installation parameters.
4. **Result:** When a managed device is detected to be missing a required application, the system will automatically deploy it.

Part 4: Completion Check

Verification Item	**Status**
The **Permit Software Check & Auto-Install** feature has been configured and verified successfully.	☐

Part 5: Feature Implementation (4) - Prohibited Software Policy

Use Case

Prevent employees from running non-work-related or potentially risky software (e.g., games, P2P clients, messaging apps like LINE) on company computers to improve productivity and reduce security risks.

Configuration Flow

Step 1: Create a "Prohibited Software" Rule

1. **Path:** Go to System > PM, click the **gear icon (⚙)**, and navigate to the Prohibited Software Settings tab.
2. **Purpose:** To create a list of software that is forbidden to run.
3. **Method:**
    - Click **"Add a Policy"** and give it a descriptive name (e.g., "Prohibit Entertainment & Chat Apps").
    - Within the rule page, click **"Add"**.
    - Fill in the **Software Name** (e.g., LINE) and its corresponding **Process Name** (e.g., LINE.exe).
4. **Result:** A rule containing specific prohibited software is created. You can now apply this rule to target groups.

💡 Core Concept: Prohibiting "Execution," Not "Installation"

The UPAS Prohibited Software feature works by **detecting and terminating the specified process (thread)**, rather than preventing the software from being installed. This means a user can still install the application, but the moment they try to run it, the process will be terminated by the Agent.

Important Note: Monitoring vs. Blocking Mode

When applying this rule to a group, you can choose between two modes:

- **Monitoring Mode:**
  - **Effect:** The system **only logs** the user's attempt to run the prohibited software and generates an event. It **does not** actually stop the application from running.
  - **Use:** Ideal for the initial policy rollout phase to assess the extent of non-compliant behavior without impacting users.
- **Blocking Mode:**
  - **Effect:** The system will **actively terminate** the prohibited software process when the user attempts to run it.
  - **Use:** Suitable for environments that require strict enforcement of security policies.

⚠ Risk Warning

Use this feature with caution! Never add critical system processes (e.g., explorer.exe, svchost.exe) to the prohibited list, as this could cause the operating system to crash or become unstable. Always verify that the process names you enter are correct and safe before enabling blocking mode.

Part 5: Completion Check

Verification Item	Status
The **Prohibited Software Policy** feature has been configured and verified successfully.	☐