**Examples of Success Goals and Metrics**

| **Goal Name** | **Goal Type** | **Metric** | **Target Value** | **Deadline** |
| --- | --- | --- | --- | --- |
| Reduce Unused SaaS Licenses | Cost Savings | % Reduction in unused licenses | 20% | Q1 2025 |
| Ensure Compliance for Adobe | Compliance | % Compliance for Adobe licenses | 100% | End of 2024 |
| Optimize License Allocations | Optimization | % Reduction in over-allocated licenses | 25% | Next 6 months |

**1. Software License Optimization**

* **Right-Sizing Licenses**: /**Usage Insights**/**Automated License Management**:

**2. Improved Compliance**

* **License Compliance**/**Audit Readiness**/**Policy Enforcement**

**3. Cost Savings**

* **Reduced Overhead Costs**/**Reclamation of Unused Software**/**Vendor Contract Insights**:

**4. Centralized Management**

* **Single Source of Truth**/**Integrated with CMDB**/**Global Software Library**

**5. Automation and Efficiency**

* **Discovery Integration**/**License Reconciliation**/**Workflow Automation**

**6. Risk Reduction**

* **Avoid Over- or Under-Licensing**/**Vendor Risk Mitigation**/**End-of-Life and Vulnerability Tracking**

**7. Enhanced Reporting and Insights**

* **Dashboards and Reports**/**Forecasting**/**Audit Reports**

**8. Integration with IT Processes**

* **ServiceNow Ecosystem**/**Request Management**/**Software Delivery Automation**:

**9. Scalability and Flexibility**

* **Supports Multiple Vendors/Cloud and On-Premise**
* By leveraging SAM Professional, organizations gain better control over their software assets, ensure compliance, and optimize costs while reducing manual effort and risks associated with improper software management.

Installing and configuring **ServiceNow Software Asset Management (SAM) Professional** involves several steps, including activating the plugin, configuring integration with discovery tools, setting up licenses, and aligning processes. Below is a step-by-step guide:

**1. Prerequisites**

Before you begin, ensure the following:

* **Licenses**: You have purchased the SAM Professional subscription.
* **Permissions**: Ensure you have the **admin role** or appropriate permissions.
* **Discovery Tools**: Set up ServiceNow Discovery or other supported discovery tools for automatic software detection.
* **CMDB**: Ensure your CMDB (Configuration Management Database) is populated and accurate.

**2. Install the SAM Professional Plugin**

**3. Configure Software Asset Management**

**A. Enable Content Service**

**B. Set Up Discovery**

**C. Configure SAM Roles**

**4. Import Software Entitlements**

**5. Set Up License Metrics**

**6. Configure Normalization and Reconciliation**

**7. Set Up SAM Dashboards**

**8. Automate License Reclamation**

**9. Integrate with Other Tools**

* **Service Catalog**:
  + Integrate SAM with the ServiceNow catalog to manage software requests and approvals.
* **IT Operations Management (ITOM)**:
  + Leverage ITOM for better discovery and asset correlation.
* **Procurement**:
  + Link SAM with Procurement to manage software purchases and renewals.

| **Scenario** | **Recommended Tool** |
| --- | --- |
| Network-wide discovery and mapping | ServiceNow Discovery |
| Detailed endpoint software inventory | SCCM |
| Real-time updates on managed endpoints | SCCM |
| Agentless software detection | ServiceNow Discovery |
| Microsoft-heavy IT environments | SCCM |
| Broader IT asset and dependency management | ServiceNow Discovery |
| Integrated SAM for license compliance | Both |

**1. Understand the Data Flow**

* **SCCM**: Collects detailed hardware and software data from Windows devices where the SCCM agent is installed.
* **ServiceNow Discovery**: Performs agentless scans to identify hardware, software, and dependencies across network devices, servers, cloud resources, and non-Windows endpoints.
* **Reconciliation**: ServiceNow’s **Reconciliation Rules** ensure the CMDB avoids duplication and prioritizes data from SCCM or Discovery based on device type or attribute.

**2. Prerequisites**

* **SCCM Connector**: Ensure you have the **SCCM Integration for ServiceNow** plugin installed and configured.
* **ServiceNow Discovery**: Ensure Discovery is set up and running with valid credentials for non-Windows devices (e.g., SSH for Linux, SNMP for network devices).
* **ServiceNow ITOM Visibility or SAM Professional**: Required for normalization and reconciliation.

**3. Configure SCCM Integration in ServiceNow**

**A. Install the SCCM Integration Plugin**

1. Navigate to **System Definition > Plugins**.
2. Search for and activate **Integration - Microsoft SCCM**.
3. Install any dependent plugins (e.g., IntegrationHub or MID Server).

**B. Configure the MID Server**

1. Set up a **MID Server** in the environment to act as a bridge between SCCM and ServiceNow.
2. Validate the MID Server connectivity.

**C. Establish SCCM Data Import**

1. Go to **Integration > SCCM > Setup** in ServiceNow.
2. Provide the necessary SCCM server details, including credentials, endpoint URLs, and ports.
3. Schedule regular imports for SCCM data (e.g., daily or weekly).

**4. Configure ServiceNow Discovery**

**A. Define IP Ranges**

1. Navigate to **Discovery > Discovery Schedule**.
2. Configure **IP Ranges** to exclude Windows devices already managed by SCCM and include other devices like Linux, network devices, and cloud resources.

**B. Configure Credentials**

1. Add credentials for non-Windows devices (e.g., Linux/Unix, SNMP for network devices, and cloud integration credentials).
2. Test credentials to ensure Discovery can access the devices.

**C. Run Discovery**

1. Schedule Discovery scans to run periodically or trigger them manually as needed.
2. Monitor scan results to ensure data is collected for non-Windows devices.

**5. Normalize and Reconcile Data**

**A. Use ServiceNow Software Asset Management (SAM)**

1. **Normalization**:
   * Normalize software data from both SCCM and Discovery using ServiceNow’s **Content Service**.
   * Resolve discrepancies in software names, versions, and publishers.
2. **License Compliance**:
   * Map normalized software to licenses for tracking compliance.

**B. Reconciliation Rules**

1. Navigate to **CMDB Reconciliation > Reconciliation Definitions**.
2. Create or modify reconciliation rules to:
   * **Prioritize SCCM data** for Windows devices.
   * Use **Discovery data** for non-Windows devices.
   * Define **Attribute-Specific Rules**:
     + Example: Use SCCM for attributes like installed software and operating system details on Windows devices.
     + Use Discovery for attributes like IP addresses or relationships for non-Windows devices.

**6. Automate Updates**

* **SCCM Data Import Schedule**:
  + Automate SCCM imports to keep Windows CI data up-to-date.
* **Discovery Scan Schedule**:
  + Schedule Discovery scans for non-Windows devices to run at regular intervals.

**7. Monitor and Validate**

**A. CMDB Completeness**

* Use the **CMDB Health Dashboard** to check for data accuracy, completeness, and compliance.
* Identify duplicate CIs and resolve conflicts using reconciliation rules.

**B. Integration Logs**

* Review logs for SCCM imports and Discovery scans to ensure successful data synchronization.

**C. Test Sample Data**

* Validate a few devices to confirm:
  + Windows devices have SCCM-populated data in the CMDB.
  + Non-Windows devices have Discovery-populated data.

**8. Benefits of This Configuration**

* **Granular Software Detection for Windows**: SCCM provides detailed inventory for managed Windows endpoints.
* **Broad Coverage for Non-Windows Devices**: Discovery fills gaps for Linux, macOS, network devices, and cloud resources.
* **Single Source of Truth**: CMDB combines data from both tools while avoiding duplication or conflicts.
* **Improved SAM**: Use normalized data for accurate license compliance and optimization.

**Best Practices**

1. **Exclude Overlapping IP Ranges**: Ensure Discovery avoids scanning devices managed by SCCM to prevent conflicts.
2. **Define Ownership**: Assign clear ownership for maintaining SCCM integration and Discovery schedules.
3. **Audit CMDB Regularly**: Periodically review CMDB health and refine reconciliation rules to address discrepancies.
4. **Optimize SAM Models**: Regularly update software models and entitlements for license compliance.

By setting up SCCM for Windows devices and Discovery for others, and reconciling both data sources in the CMDB, you can achieve comprehensive IT visibility while leveraging the strengths of both tools. Let me know if you need more specific configuration details!Bottom of Form

To use **ServiceNow Discovery** for software discovery, you need to ensure that the required **ports are open** between the **MID Server** and the endpoints, and the correct **credentials are configured**. Here’s a detailed breakdown:

**1. Ports to Open for Discovery**

Discovery requires access to various protocols to collect data from endpoints. The specific ports depend on the device types and operating systems being discovered.

**A. Windows Devices**

* **WMI (Windows Management Instrumentation)**:
  + Ports:
    - TCP 135 (RPC Endpoint Mapper)
    - Dynamic RPC range (TCP ports 49152–65535 by default for Windows Server 2008 and later).
  + Note: Ensure that the MID Server can communicate with the dynamic ports. You may need to restrict the range for security.
* **SMB (Server Message Block)**:
  + Ports:
    - TCP 445 (File and Printer Sharing for WMI-based software detection).

**B. Linux/Unix Devices**

* **SSH (Secure Shell)**:
  + Port: TCP 22.

**C. Network Devices**

* **SNMP (Simple Network Management Protocol)**:
  + Ports:
    - UDP 161 (SNMP data collection).
    - UDP 162 (optional, for SNMP traps).

**D. Database Servers**

* **Databases** (if you need software discovery for database servers):
  + Common Ports:
    - TCP 1433 (Microsoft SQL Server).
    - TCP 3306 (MySQL).
    - TCP 1521 (Oracle DB).
    - TCP 5432 (PostgreSQL).
  + Other ports depend on the database type and configuration.

**E. Cloud and Virtualization**

* For discovering cloud or virtualized environments:
  + **AWS, Azure, or GCP APIs**: No ports needed; access is via the internet using the respective API credentials.
  + **VMware vCenter**: TCP 443 (HTTPS).

**2. Credentials to Configure**

Credentials are critical for Discovery to access endpoint data. You’ll need to configure appropriate credentials in **ServiceNow Credentials Management**.

**A. Windows Devices**

* **Type**: Windows Credentials.
* **Requirements**:
  + A domain account with administrative privileges on the target Windows machines.
  + The account should have the necessary permissions to access WMI and SMB (e.g., member of the local Administrators group).
  + Format: domain\username or username.

**B. Linux/Unix Devices**

* **Type**: SSH Credentials.
* **Requirements**:
  + A user account with sufficient privileges to execute commands (e.g., root or an account with sudo access).
  + Use a key-based authentication or password-based authentication:
    - Key-based: Upload the private key to ServiceNow.
    - Password-based: Provide a username and password.

**C. Network Devices**

* **Type**: SNMP Credentials.
* **Requirements**:
  + SNMP community string for SNMP v2.
  + For SNMP v3, provide:
    - Username.
    - Authentication protocol and password.
    - Privacy protocol and password.

**D. Databases**

* **Type**: Database Credentials.
* **Requirements**:
  + A database user account with sufficient privileges to read schema and configuration information.
  + Example: For SQL Server, the account should have the VIEW SERVER STATE and VIEW DATABASE STATE permissions.

**E. Cloud and Virtualization**

* **AWS, Azure, GCP**: API Key or IAM credentials.
* **VMware vCenter**: A user account with Read-Only or higher permissions.

**3. Additional Configuration**

**A. MID Server Firewall Rules**

* Ensure the MID Server can communicate with all endpoints over the required ports.
* Allow MID Server to access ServiceNow instance over HTTPS (TCP 443).

**B. Restrict Port Ranges (Optional)**

* For WMI, restrict the dynamic port range (default: 49152–65535) using the **DCOM Port Range** configuration.

**C. Test Credentials**

* Use the **Test Credential** feature in ServiceNow to verify connectivity and permissions.

**D. Discovery Patterns**

* Ensure appropriate **patterns** are configured in Discovery to detect software on endpoints.

**4. Best Practices**

* **Limit Access**: Use least-privileged accounts wherever possible.
* **Secure Credentials**: Store credentials securely using ServiceNow's built-in encryption.
* **Network Segmentation**: Ensure the MID Server is located in the same network segment as the endpoints for better connectivity.
* **Credential Rotation**: Periodically update passwords or keys for security.

By opening the correct ports and configuring the appropriate credentials, Discovery can effectively identify and detect software installed on your endpoints and update the CMDB accordingly. Let me know if you need help with specific configurations!

To get **SSO applications, groups, and users** into **ServiceNow Software Asset Management (SAM)**, you can leverage the integration between your **Identity Provider (IdP)** (e.g., Okta, Azure AD, Ping Identity) and ServiceNow. Here's a step-by-step guide:

**1. Prerequisites**

1. **ServiceNow Modules**:
   * SAM Professional.
   * IntegrationHub (for API-based integrations).
   * Optional: CMDB, if you'd like to link applications with CIs.
2. **Access to SSO Provider**:
   * Admin credentials to the IdP (e.g., Okta, Azure AD).
3. **Supported IdP**:
   * Ensure your IdP supports user, group, and application exports via REST API, SCIM, or other protocols.
4. **ServiceNow MID Server** (if required):
   * If your IdP is on-premises, configure a MID Server for communication between ServiceNow and the IdP.

**2. Integrate SSO with ServiceNow**

**A. Use Out-of-the-Box Integrations**

1. **Okta Integration**:
   * Navigate to **IntegrationHub > Integrations** in ServiceNow.
   * Look for "Okta Integration" and configure:
     + API URL.
     + Client ID and Client Secret.
     + Synchronization schedules for applications, users, and groups.
2. **Azure AD Integration**:
   * Use the **Integration - Microsoft Azure AD** plugin.
   * Configure the connection with:
     + Azure Tenant ID.
     + Client ID and Secret.
     + Permissions to read applications, groups, and users.
3. **Ping Identity, OneLogin, etc.**:
   * Use a generic REST API or SCIM connector.

**B. Custom Integration Using REST or SCIM**

If your IdP does not have an out-of-the-box connector, follow these steps:

1. **Set Up REST API Integration**:
   * Obtain the API endpoint, client ID, and client secret from the IdP.
   * Use ServiceNow’s **REST API Integration** or **IntegrationHub** to connect to the IdP.
2. **Create Data Import Jobs**:
   * Fetch:
     + Applications: API endpoint for apps (e.g., /api/v1/apps for Okta).
     + Groups: API endpoint for groups (e.g., /api/v1/groups).
     + Users: API endpoint for users (e.g., /api/v1/users).
   * Parse and map the data to ServiceNow tables.

**3. Map SSO Data in ServiceNow**

Once integration is set up, map the imported data to ServiceNow tables:

**A. SSO Applications**

* **Target Table**: cmdb\_ci\_spa (for SaaS apps).
* Fields to map:
  + Application Name.
  + App Type (e.g., SaaS, on-prem).
  + Description.
  + Active Users.

**B. Groups**

* **Target Table**: sys\_user\_group.
* Fields to map:
  + Group Name.
  + Description.
  + Associated Users.

**C. Users**

* **Target Table**: sys\_user.
* Fields to map:
  + Username.
  + Email Address.
  + Assigned Groups.

**4. Set Up Data Synchronization**

* Schedule regular data imports to keep the information up-to-date:
  + Synchronize users, groups, and applications daily or weekly.
  + Use ServiceNow's **Scheduled Jobs** for automation.

**5. Manage SSO Data in SAM**

Once the data is in ServiceNow, SAM can utilize it for license tracking and optimization:

1. **Link Applications to Licenses**:
   * Use the **SAM License Workbench** to connect SSO applications to entitlements.
2. **Track Usage**:
   * Monitor active vs. inactive users for each SaaS application.
   * Identify overlicensed or underused applications.
3. **Group-Level Management**:
   * Assign licenses at the group level using imported SSO groups.
4. **Reclamation**:
   * Automate license reclamation for inactive users or unused groups.

**6. Reporting and Dashboards**

* Use ServiceNow reporting to create dashboards:
  + **SSO Applications**: Usage, active users, and compliance.
  + **SSO Groups**: Assigned vs. used licenses.
  + **SSO Users**: Active, inactive, or deprovisioned users.

**Example: Integration with Okta**

1. Configure Okta integration in ServiceNow with API credentials.
2. Import data:
   * Applications: /api/v1/apps.
   * Groups: /api/v1/groups.
   * Users: /api/v1/users.
3. Map the data to ServiceNow tables (cmdb\_ci\_spa, sys\_user\_group, and sys\_user).
4. Schedule synchronization jobs.
5. Link applications in SAM with imported users and groups.

By importing and mapping SSO data into ServiceNow, you can ensure your SAM processes cover all SaaS applications, users, and groups, enabling better license management, compliance, and cost optimization. Let me know if you'd like help with a specific IdP or configuration!

**Software Spend Detection in ServiceNow SAM**

**Software Spend Detection** is a feature in **ServiceNow Software Asset Management (SAM)** that helps identify software purchases and expenditures by analyzing financial data, such as purchase orders, invoices, and expenses. This allows organizations to reconcile their software spend with entitlements, ensuring proper license tracking, cost optimization, and compliance.

**How Software Spend Detection Works**

1. **Financial Data Import**:
   * Data related to software purchases (e.g., purchase orders, invoices) is imported into ServiceNow from your financial system or manually entered.
2. **Data Normalization**:
   * The system normalizes software-related spending using the **ServiceNow Content Library**.
   * This includes matching vendor names, product names, and costs to known software publishers and products.
3. **Spend Categorization**:
   * Identifies whether the spend is for SaaS, on-premises software, hardware, or services.
   * Classifies spend by product category and links it to the appropriate license records.
4. **Reconciliation**:
   * Matches spend data with software entitlements in SAM.
   * Highlights discrepancies, such as unused or duplicate licenses and overspending.
5. **Insights and Optimization**:
   * Dashboards and reports provide insights into spending trends, unused licenses, and potential savings opportunities.

**Steps to Configure Software Spend Detection**

**1. Enable the Feature**

1. Navigate to **SAM > Spend Detection** in the ServiceNow instance.
2. Ensure the **Software Spend Detection plugin** is activated (requires SAM Professional).
   * If not activated, reach out to your system admin or ServiceNow support.

**2. Configure Data Sources**

You need to bring financial data into ServiceNow from systems such as ERP, procurement, or expense management platforms.

**A. Import Financial Data**

1. Use the **Import Sets** or **IntegrationHub** to import:
   * Purchase orders.
   * Invoice data.
   * Expense reports.
2. **Mapping the Data**:
   * Map vendor and product details from the source data to the respective ServiceNow fields.
   * Ensure fields like Vendor, Product, Amount, Purchase Date, and Category are populated.

**B. Connect Financial Systems (Optional)**

For automatic data imports:

* **Supported Systems**: SAP, Oracle Financials, Coupa, etc.
* Use IntegrationHub or REST API to configure connections.

**3. Normalize Spend Data**

1. **ServiceNow Content Library**:
   * Use the built-in library to normalize vendor and product names.
   * This maps financial data to known software products and publishers.
2. **Manual Adjustments**:
   * If spend data is not automatically recognized, manually map unmatched vendors or products to their corresponding entries in the Content Library.

**4. Link Spend to Software Entitlements**

1. Navigate to **SAM > Licensing > License Workbench**.
2. Use the spend data to:
   * Create or update entitlements for on-premises software or SaaS subscriptions.
   * Track cost per license or subscription.

**5. Configure Scheduled Jobs**

1. Automate the periodic import of financial data.
2. Navigate to **System Definition > Scheduled Jobs**.
3. Create a job to:
   * Sync spend data from financial systems.
   * Normalize and categorize spend automatically.

**6. Reporting and Dashboards**

1. Use the out-of-the-box reports and dashboards in **SAM > Spend Detection**:
   * Analyze total spend by vendor, product, and category.
   * View SaaS vs. on-premises software spending.
   * Identify unused or over-licensed software.
2. Customize reports to include:
   * Cost trends over time.
   * Cost-per-user or cost-per-license metrics.

**Best Practices**

1. **Regular Data Import**: Schedule imports to ensure spend data is current.
2. **Vendor Management**: Maintain accurate vendor information in the CMDB or Vendor table.
3. **Spend Optimization**:
   * Use spend insights to negotiate better deals with vendors.
   * Automate license reclamation for unused software.
4. **Data Accuracy**:
   * Ensure imported financial data is clean and well-structured.
   * Use the ServiceNow Content Library for normalization.

**Benefits of Software Spend Detection**

1. **Cost Optimization**:
   * Detect and reduce overspending.
   * Optimize license allocation.
2. **Improved Compliance**:
   * Ensure software spending aligns with license entitlements.
3. **Enhanced Visibility**:
   * Gain insights into software costs across the organization.
4. **Better Decision-Making**:
   * Use spend data to plan budgets and negotiate contracts.

**Success Goals in ServiceNow Software Asset Management (SAM)**

**Success Goals** in ServiceNow SAM are objectives that help you measure and track the success of your software asset management initiatives. These goals align with organizational priorities, providing actionable insights to improve software governance, reduce costs, and ensure compliance. Success goals are tied to key performance indicators (KPIs) and are often visualized using dashboards.

**Purpose of Success Goals in SAM**

1. **Measurable Outcomes**:
   * Define clear, quantifiable objectives for SAM processes.
   * Track progress toward achieving these objectives.
2. **Improved Governance**:
   * Align SAM activities with organizational goals like cost savings, compliance, or software optimization.
3. **Stakeholder Engagement**:
   * Provide visibility into SAM performance for executives and stakeholders.
4. **Actionable Insights**:
   * Identify areas for improvement in license usage, compliance, and spend optimization.

**How to Create Success Goals in SAM**

Follow these steps to create and implement success goals in ServiceNow SAM:

**1. Identify Key Areas for Improvement**

Determine the primary focus of your SAM initiative. Typical areas include:

* **Compliance**: Ensure all software usage aligns with licensing agreements.
* **Cost Savings**: Reduce unnecessary software spend.
* **Optimization**: Optimize license allocations and usage.
* **Risk Reduction**: Mitigate risks related to audits or over-licensing.

**2. Define Specific Success Goals**

Based on the focus areas, define clear and actionable goals. Use the **SMART** criteria:

* **Specific**: Clearly state what you aim to achieve.
* **Measurable**: Define metrics to track progress.
* **Achievable**: Ensure goals are realistic.
* **Relevant**: Align with business objectives.
* **Time-Bound**: Set deadlines for achievement.

**Examples**:

* Reduce unused software licenses by 20% within the next quarter.
* Achieve 100% compliance for top 10 critical applications by year-end.
* Decrease SaaS spend by 15% within six months.

**3. Set Up Success Goals in ServiceNow**

1. **Navigate to SAM Module**:
   * Go to **SAM > Planning > Success Goals**.
2. **Create a New Goal**:
   * Click on **New** to create a goal.
3. **Provide Details**:
   * **Goal Name**: A descriptive name for the goal (e.g., "Reduce Unused Licenses").
   * **Goal Type**: Select the category (e.g., Cost Savings, Compliance, Optimization).
   * **Target Metric**: Specify the KPI or metric to measure (e.g., "Number of unused licenses reduced").
   * **Target Value**: Define the expected outcome (e.g., "Reduce by 20%").
   * **Deadline**: Set a timeline for achieving the goal.
   * **Owner**: Assign responsibility for tracking and completing the goal.
4. **Link Related Data**:
   * Attach **tasks**, **licenses**, or **applications** relevant to the goal.
   * Link to **dashboards** or **reports** for ongoing tracking.

**4. Track Progress Toward Goals**

1. **Use Dashboards**:
   * View progress in real-time on SAM dashboards.
   * Access visual insights into spend, compliance, and optimization metrics.
2. **Analyze Reports**:
   * Run reports to evaluate progress against success goals.
   * Identify areas where additional focus or resources are required.
3. **Automate Updates**:
   * Use **scheduled jobs** to refresh data and update progress automatically.

**5. Adjust and Iterate**

1. **Review Regularly**:
   * Assess progress toward goals during periodic reviews (monthly, quarterly).
   * Address roadblocks or challenges affecting goal achievement.
2. **Adjust Goals if Needed**:
   * Revise target values or deadlines based on changes in business priorities or SAM maturity.

**Examples of Success Goals and Metrics**

| **Goal Name** | **Goal Type** | **Metric** | **Target Value** | **Deadline** |
| --- | --- | --- | --- | --- |
| Reduce Unused SaaS Licenses | Cost Savings | % Reduction in unused licenses | 20% | Q1 2025 |
| Ensure Compliance for Adobe | Compliance | % Compliance for Adobe licenses | 100% | End of 2024 |
| Optimize License Allocations | Optimization | % Reduction in over-allocated licenses | 25% | Next 6 months |

**Benefits of Success Goals in SAM**

1. **Visibility**: Clear goals provide a shared understanding of priorities and progress.
2. **Cost Control**: Focused initiatives help reduce unnecessary software spend.
3. **Compliance**: Helps organizations stay audit-ready by monitoring license usage.
4. **Data-Driven Decisions**: Goals tied to KPIs support better strategic planning.

By creating and tracking success goals in ServiceNow SAM, you can ensure your organization remains compliant, cost-effective, and optimized in its software asset management practices. Let me know if you'd like help setting up specific goals!

Detecting software and licenses on **mobile devices** in ServiceNow **Software Asset Management (SAM)** involves leveraging **ServiceNow Discovery**, **Integration with Mobile Device Management (MDM) tools**, and **SaaS license tracking**. Here's how you can achieve this:

**1. Prerequisites**

1. **SAM Professional**: Ensure that the SAM Professional module is installed and configured.
2. **IntegrationHub**: To integrate with third-party MDM tools.
3. **MDM Solution**: Use a compatible MDM solution like **Microsoft Intune**, **VMware Workspace ONE**, or **MobileIron**.
4. **Discovery Configuration** (optional): For mobile devices connected to corporate networks.
5. **Content Library**: Ensure the ServiceNow Content Library is updated for software normalization.

**2. Use Cases for Mobile Software and Licenses Detection**

* **Installed Applications**: Detect apps installed on mobile devices.
* **SaaS Licenses**: Track licenses for SaaS applications accessed via mobile devices.
* **License Compliance**: Ensure compliance with corporate app licensing agreements.
* **Cost Tracking**: Monitor app subscription costs, such as mobile versions of Microsoft 365, Adobe, or Salesforce.

**3. Methods to Detect Software and Licenses on Mobile Devices**

**A. Integration with Mobile Device Management (MDM) Tools**

The most effective way to track software and licenses on mobile devices is by integrating ServiceNow with an MDM tool.

1. **Supported MDM Tools**:
   * Microsoft Intune.
   * VMware Workspace ONE.
   * MobileIron.
   * Jamf Pro (for Apple devices).
2. **Steps to Integrate MDM with ServiceNow**:
   * Go to **IntegrationHub > Integrations**.
   * Select the prebuilt integration for your MDM tool (e.g., **Intune Integration**).
   * Configure API credentials (client ID, client secret, tenant ID, etc.).
   * Import device data (device name, owner, OS, installed apps, and versions).
3. **Imported Data Includes**:
   * Device details: Device type, OS, version, and owner.
   * Installed applications: App names, versions, and associated licenses.
   * Usage statistics (if supported by the MDM).
4. **Map Data in SAM**:
   * Link applications detected via MDM to SAM software models.
   * Normalize app data using the **Content Library**.

**B. Using ServiceNow Discovery**

If an MDM tool is unavailable, you can use **Discovery** to detect software on mobile devices connected to your network.

1. **Network Scan**:
   * Configure **ServiceNow Discovery** to scan mobile devices connected to corporate networks.
   * Ensure credentials (e.g., SSH, SNMP) are configured for supported mobile OSes (limited to enterprise-managed devices).
2. **Limitations**:
   * Discovery is less effective for mobile devices not consistently connected to the corporate network.
   * It may not detect all installed applications.

**C. SaaS License Tracking**

For SaaS applications accessed on mobile devices:

1. Use **SaaS License Management** in SAM.
   * Integrate with SaaS vendors like Microsoft 365, Google Workspace, or Adobe.
   * Track usage and license assignments for apps accessed via mobile devices.
2. **Steps**:
   * Navigate to **SAM > SaaS License Management > Applications**.
   * Add integrations for SaaS providers.
   * Track user activity and licenses through API integration.

**4. Track and Manage Licenses**

Once software and licenses are detected, ensure they are properly managed:

1. **Normalize Data**:
   * Use the ServiceNow Content Library to normalize app names and versions.
2. **Reconcile Licenses**:
   * Match detected applications to licenses in SAM.
   * Identify unused licenses or over-allocated licenses.
3. **Compliance Tracking**:
   * Use the License Workbench to track compliance.
   * Address non-compliance issues by purchasing additional licenses or reclaiming unused ones.
4. **Cost Optimization**:
   * Identify underutilized mobile apps and reassign or retire licenses.

**5. Reporting and Dashboards**

1. **Mobile Software Inventory**:
   * Generate reports on detected applications, their versions, and installation counts.
2. **License Compliance**:
   * Use dashboards to monitor compliance for mobile-specific licenses.
3. **Usage Trends**:
   * Analyze which mobile apps are actively used to optimize licensing.

**6. Best Practices**

* **Standardize App Deployment**:
  + Use the MDM tool to enforce standard app deployment policies.
* **Restrict Unauthorized Apps**:
  + Detect and block unauthorized apps from accessing corporate data.
* **Regular Data Sync**:
  + Schedule regular imports from MDM tools or Discovery scans.
* **Reclaim Licenses**:
  + Use SAM automation to reclaim unused licenses.

**Example: Integrating Microsoft Intune with ServiceNow SAM**

1. Configure **Microsoft Intune Integration**:
   * Use the prebuilt Intune integration in ServiceNow.
   * Import device and application data.
2. Normalize Applications:
   * Map imported app data to SAM software models using the Content Library.
3. Track Licenses:
   * Use SAM to reconcile Intune-managed applications with licenses.
4. Generate Reports:
   * Create a dashboard showing installed applications, license usage, and compliance.

By integrating SAM with MDM tools and leveraging SaaS license tracking, you can achieve comprehensive visibility into mobile software usage, ensuring compliance and optimizing costs effectively. Let me know if you need more details on a specific integration!

Integrating **ServiceNow Software Asset Management (SAM)** with the **Service Catalog** streamlines the process of software provisioning and un-provisioning by automating requests, approvals, and license management. This integration ensures that software requests are fulfilled efficiently while maintaining compliance and optimizing costs.

**Benefits of Integration**

1. **Streamlined Provisioning**:
   * Automate the assignment of software licenses when users request software.
2. **Un-Provisioning**:
   * Automatically reclaim licenses when users return or no longer need software.
3. **Compliance**:
   * Ensure that license allocation adheres to available entitlements.
4. **Cost Optimization**:
   * Reduce license costs by reclaiming unused licenses efficiently.
5. **Self-Service**:
   * Allow employees to request software directly via the Service Catalog.

**Steps to Integrate SAM with the Service Catalog**

**1. Configure SAM for License Management**

Before integrating with the Service Catalog, ensure SAM is set up correctly:

1. Navigate to **SAM > Administration > License Workbench**.
2. Import and normalize software entitlements (licenses).
3. Configure license metrics (per-user, per-device, etc.).
4. Set up reclamation rules to reclaim unused licenses (e.g., based on inactivity or returned software).

**2. Create Service Catalog Items for Software Requests**

Create catalog items that users can request for software provisioning and un-provisioning.

**A. Provisioning Software**

1. **Navigate to**:
   * Go to **Service Catalog > Maintain Items**.
2. **Create a New Catalog Item**:
   * Name: Provide a clear name for the software (e.g., "Microsoft Office 365").
   * Category: Assign it to a relevant category (e.g., "Software Requests").
3. **Configure the Request Form**:
   * Add fields like:
     + User information (auto-filled from the user profile).
     + Software name (auto-filled for specific software items).
     + Justification for the request.
     + Cost information (optional).
4. **Workflow Association**:
   * Attach an **approval workflow** to the catalog item:
     + Example: Manager approval for high-cost software.
   * Add automation to check for available licenses in SAM.
5. **Backend Automation**:
   * Use **Flow Designer** or **Orchestration** to automate provisioning:
     + Assign the requested license to the user in SAM.
     + Update the CMDB with the software assignment.
     + Notify the user once provisioning is complete.

**B. Un-Provisioning Software**

1. **Create an "Un-Provisioning" Catalog Item**:
   * Similar to the provisioning catalog item, but for returning or un-assigning software.
2. **Workflow Automation**:
   * Use **Flow Designer** to automate:
     + Removing the license from the user in SAM.
     + Reclaiming the license for future use.
     + Updating the CMDB.

**3. Automate License Checks**

Ensure SAM validates license availability during the provisioning process:

1. **License Availability Check**:
   * Configure SAM to check for available entitlements before approving a request.
2. **Integration with Request Workflow**:
   * Add a script in the request workflow to:
     + Query SAM for license availability.
     + Deny the request if no licenses are available.
     + Notify SAM admins to purchase additional licenses if required.

**4. Use SAM Reclamation Rules**

Automate un-provisioning for unused licenses:

1. Navigate to **SAM > Reclamation Rules**.
2. Configure rules to identify unused software:
   * Inactivity (e.g., no usage for 30 days).
   * User returns software voluntarily.
3. Link Reclamation to the Service Catalog:
   * Automatically trigger an un-provisioning request when reclamation rules detect unused licenses.

**5. Reporting and Notifications**

1. **Track Requests**:
   * Use the Service Catalog Request module to monitor open and completed software requests.
2. **License Utilization Reports**:
   * Use SAM dashboards to view provisioned vs. available licenses.
3. **Notifications**:
   * Notify users and stakeholders at key stages:
     + Request submission.
     + License provisioning.
     + License un-provisioning.

**Example Workflow for Provisioning Microsoft Office 365**

1. **Request Submission**:
   * A user submits a request for "Microsoft Office 365" via the Service Catalog.
2. **Approval**:
   * Workflow routes the request to the user's manager for approval.
3. **License Check**:
   * SAM checks if a license is available:
     + If yes: Proceed to provisioning.
     + If no: Notify the SAM admin to purchase additional licenses.
4. **Provisioning**:
   * Automate license assignment using Flow Designer.
   * Update the CMDB to reflect the new software assignment.
5. **Completion**:
   * Notify the user that the software is ready to use.

**Best Practices**

1. **Maintain Accurate License Data**:
   * Regularly update license entitlements in SAM to avoid delays in provisioning.
2. **Leverage Automation**:
   * Use Flow Designer and Orchestration to reduce manual effort in software provisioning and reclamation.
3. **Implement Reclamation Rules**:
   * Reclaim unused licenses to minimize waste.
4. **Monitor Performance**:
   * Use SAM and Service Catalog dashboards to track provisioning times, license utilization, and request volumes.

This integration provides a seamless software management process, enhances user experience, and ensures compliance and cost efficiency. Let me know if you'd like further details or examples!

Creating software models in **ServiceNow Software Asset Management (SAM)** is a crucial step to manage software effectively, ensuring proper license compliance, optimization, and cost control. A software model serves as a representation of a specific software product, enabling tracking, normalization, and reconciliation of software data within the platform.

**Steps to Create a Software Model**

**1. Prerequisites**

Before creating software models, ensure:

1. **SAM Professional** is installed and configured.
2. **Content Library** is up-to-date for normalization.
3. Discovery or data import processes are configured to detect software installations.

**2. Navigate to the Software Models Section**

1. Go to **SAM > Licensing > Software Models**.
2. Click **New** to create a software model.

**3. Define the Software Model**

On the software model form, configure the following fields:

**Basic Information**

1. **Name**:
   * Enter a descriptive name for the software model (e.g., "Microsoft Office 365").
2. **Publisher**:
   * Select the software publisher (e.g., "Microsoft").
   * Ensure the publisher is normalized; otherwise, SAM will not recognize it during reconciliation.
3. **Product**:
   * Specify the software product name (e.g., "Office 365 E3").
4. **Version** (optional):
   * Add the version of the software, if applicable (e.g., "2023").

**Software Category**

1. **Category**:
   * Choose the appropriate category (e.g., "Office Suites").
   * Categories are used for grouping and normalization.

**License Metrics**

1. **Metric Group**:
   * Select the license metric used to track compliance (e.g., Per User, Per Device, Per Core, etc.).
2. **Metric**:
   * Define the metric specific to the software (e.g., "Per User").
3. **Licensable**:
   * Mark the model as **licensable** if it requires a license.

**Normalization**

1. **Normalization Status**:
   * Ensure the model is set to **Compliant** (this requires the content library).
   * Use the **Normalize** button to standardize the name, version, and category.

**Entitlements and Costs**

1. **Linked Entitlements**:
   * Link entitlements (licenses) to the software model to track usage and compliance.
2. **Cost Details**:
   * Add cost information (e.g., cost per license, maintenance cost) for financial tracking.

**Discovery Map (if applicable)**

1. **Discovery Map**:
   * Map the software model to discovered installations detected by ServiceNow Discovery.
   * This links the software model to specific discovery results, ensuring accurate data.

**4. Save and Normalize the Software Model**

1. After entering all the details, click **Save**.
2. Use the **Normalize** function:
   * This compares the model with the ServiceNow Content Library to ensure proper categorization and recognition.

**5. Link Licenses (Entitlements) to the Software Model**

To manage compliance and track usage, link entitlements to the software model:

1. Navigate to **SAM > Licensing > Entitlements**.
2. Create a new entitlement or edit an existing one.
3. Associate the entitlement with the software model:
   * Select the software model in the **Software Model** field.
4. Save the entitlement.

**6. Use the License Workbench**

1. Navigate to **SAM > License Workbench**.
2. Select the newly created software model.
3. Review the following:
   * **Compliance Status**: Check if license usage is within the purchased entitlements.
   * **Installations**: View detected installations linked to the model.
   * **Optimization Opportunities**: Identify underutilized licenses or cost-saving options.

**7. Verify with Discovery or Data Imports**

1. If using **Discovery**, ensure that discovered software matches the software model.
2. For manual imports, map imported data to the software model to ensure proper tracking.

**Best Practices for Creating Software Models**

1. **Normalize Regularly**:
   * Regularly update the Content Library and normalize software models to ensure accuracy.
2. **Group Software by Categories**:
   * Use software categories to organize software models for easier management.
3. **Reconcile Data**:
   * Frequently reconcile installations, entitlements, and licenses to maintain compliance.
4. **Use Automation**:
   * Leverage SAM automation to track usage, reclaim unused licenses, and detect over-licensed software.
5. **Maintain Version Control**:
   * Create separate models for different versions of the same software, if necessary.

**Example: Creating a Software Model for Adobe Acrobat Pro**

1. **Name**: Adobe Acrobat Pro
2. **Publisher**: Adobe
3. **Product**: Acrobat
4. **Version**: DC 2023
5. **Category**: PDF Software
6. **Metric**: Per User
7. **Normalization**: Use the Normalize function.
8. **Entitlements**: Link to entitlements for Adobe Acrobat licenses.

This ensures accurate tracking and management of the software while aligning with organizational compliance and cost goals. Let me know if you'd like assistance with a specific example or further configuration!

does discovery creates these software models automatically during the discovery or do we need to create them before the discovery? if the software model is not found for the detected software during the discovery what it will do?

**ChatGPT said:**

ChatGPT

Here’s a detailed explanation of how **ServiceNow Discovery** interacts with software models in **Software Asset Management (SAM)**:

**1. Does Discovery Create Software Models Automatically?**

No, **ServiceNow Discovery** does not create software models automatically during discovery. Instead:

* Discovery **detects installed software** on devices and updates the **[Installed Software]** (cmdb\_sam\_sw\_install) table in the CMDB.
* The detected software is then **normalized** by comparing it with the **ServiceNow Content Library**, which includes recognized software publishers, products, and versions.

For a software model to exist:

* You need to either create it manually or ensure that the normalization process matches the discovered software to an existing software model in SAM.

**2. Do You Need to Create Software Models Before Discovery?**

Not necessarily. If the ServiceNow **Content Library** has information about the discovered software, a software model will be created during the **normalization process**.

Here’s how it works:

* When discovery detects new software, the normalization process attempts to match it with known software in the Content Library.
* If a match is found, the system automatically creates a normalized software model in SAM.

However, you may need to create software models manually in these cases:

1. **Custom Software**:
   * If the software is proprietary or not included in the Content Library.
   * You must manually create the software model and map it to the discovered software.
2. **Non-Normalized Software**:
   * If normalization fails because the software is not recognized by the Content Library.

**3. What Happens if No Software Model is Found During Discovery?**

If a software model is not found during the normalization process:

1. **The Installed Software Entry Remains Unnormalized**:
   * The discovered software is listed as unnormalized in the **Installed Software** (cmdb\_sam\_sw\_install) table.
   * It won’t be associated with any software model.
2. **Impact on SAM**:
   * SAM won’t track license compliance or usage for the unnormalized software.
   * It remains in the system as "Unnormalized Software" until resolved.
3. **Resolution**:
   * **Manual Normalization**: You can manually normalize the software entry by creating a new software model or updating the mapping.
   * **Update Content Library**: If the software is widely used, you can request ServiceNow to add it to the Content Library.

**4. Normalization Process Overview**

The normalization process matches discovered software to known software in the ServiceNow Content Library. The key steps include:

1. **Discovery Detects Software**:
   * Discovery populates the **Installed Software** table.
2. **Normalization Runs**:
   * Normalization attempts to map discovered software to a publisher, product, and version in the Content Library.
3. **Outcomes**:
   * **Match Found**: A software model is created or linked automatically.
   * **No Match**: The entry is flagged as "Unnormalized" and requires manual intervention.

**5. Best Practices to Handle Unnormalized Software**

1. **Regularly Update the Content Library**:
   * Ensure your Content Library is current by scheduling updates.
   * This improves the chances of matching discovered software to existing software models.
2. **Manually Normalize Unrecognized Software**:
   * Navigate to **SAM > Normalize Software Installations** to resolve unnormalized entries.
3. **Create Custom Software Models**:
   * For proprietary or uncommon software, manually create software models and associate them with the unnormalized installations.
4. **Review Installed Software**:
   * Regularly review unnormalized software in the Installed Software table to ensure accurate tracking.

**Summary of the Discovery and Software Model Relationship**

| **Scenario** | **Outcome** |
| --- | --- |
| Software matches the Content Library | Software is normalized, and a software model is created/linked automatically. |
| Software does not match the Content Library | Entry remains unnormalized, requiring manual intervention or a custom software model. |

If you need guidance on creating custom software models or resolving unnormalized software entries, let me know!

Configuring and using **patterns** for specific software discovery in ServiceNow **Software Asset Management (SAM)** involves tailoring the **Discovery** process to accurately identify and manage software products. Patterns are reusable, modular, and scalable methods for gathering data about software installations.

Here’s how to configure and use patterns for specific software discovery:

**1. Understanding Patterns in Discovery**

* **Patterns** are used by the **ServiceNow Discovery** module to collect data from devices, applications, or services.
* For software discovery, patterns are used to:
  + Detect installed software.
  + Retrieve specific details (version, edition, etc.).
  + Populate the CMDB and integrate with SAM for compliance tracking.

**2. When to Use Custom Patterns for Software Discovery**

Use patterns for specific software discovery if:

* The software is not being detected by default Discovery.
* You need detailed information about the software beyond what is available through basic Discovery probes and sensors.
* The software installation has unique indicators, such as specific files, processes, or registry entries.

**3. Steps to Configure and Use Patterns for Software Discovery**

**Step 1: Access the Pattern Designer**

1. Navigate to **Discovery > Patterns**.
2. Click **Create New** to define a custom pattern for your specific software.

**Step 2: Define the Basic Details**

1. **Name**:
   * Provide a meaningful name for the pattern (e.g., "Adobe Acrobat Pro Discovery").
2. **Type**:
   * Choose **Application** as the pattern type.
3. **Category**:
   * Select **Software** as the category.

**Step 3: Identify the Software**

Define the **entry point** to identify the software installation on a device. The entry point could be:

1. **Processes**:
   * Look for a specific running process associated with the software (e.g., acrobat.exe).
2. **Files and Folders**:
   * Search for specific files or folders (e.g., C:\Program Files\Adobe\Acrobat).
3. **Registry Keys** (Windows-specific):
   * Check the Windows registry for specific keys related to the software (e.g., HKLM\Software\Adobe\Acrobat).
4. **Services**:
   * Identify the software by detecting related services.

**Step 4: Create Pattern Steps**

Add steps to the pattern to locate and gather details about the software:

1. **Identification Section**:
   * Add steps to identify the software's presence (e.g., match process names, folder paths, or registry entries).
2. **Data Collection Section**:
   * Collect detailed information about the software, such as:
     + Version
     + Edition
     + Installation path
     + Publisher
3. **Validation Section** (Optional):
   * Include additional checks to validate the software discovery.

**Step 5: Test the Pattern**

1. Deploy the pattern on a test device using the **Pattern Debugger**.
2. Verify that the pattern correctly detects the software and collects the necessary details.
3. Troubleshoot any errors by adjusting pattern steps.

**Step 6: Publish and Activate the Pattern**

1. Once the pattern is tested and validated, click **Publish** to make it available for Discovery.
2. Activate the pattern so it can be used in live Discovery scans.

**4. Using the Pattern in Discovery**

1. Ensure the **MID Server** has access to the target devices.
2. Configure Discovery to use the pattern:
   * Assign the custom pattern to the relevant **Discovery Schedule**.
3. Run a **Discovery Scan** to detect the software.

**5. Linking Detected Software to SAM**

After Discovery detects the software:

1. The discovered software is listed in the **Installed Software** table (cmdb\_sam\_sw\_install).
2. The software is normalized using the ServiceNow **Content Library**.
3. A corresponding **Software Model** is created or linked in SAM.
4. Link entitlements to manage licenses and track compliance.

**6. Monitor and Refine**

1. Review the results of the Discovery scan to ensure the software is detected accurately.
2. Update the pattern as needed to improve detection:
   * Add new identification methods.
   * Refine data collection steps.
3. Schedule regular scans to keep the CMDB updated.

**Example: Custom Pattern for Adobe Acrobat Pro**

1. **Entry Point**:
   * Process: acrobat.exe
   * Registry Key: HKLM\Software\Adobe\Acrobat
2. **Steps**:
   * Identify the running process.
   * Verify the installation directory.
   * Collect details from the registry (e.g., version, license key).
3. **Output**:
   * Populate the Installed Software table with detailed information about Adobe Acrobat Pro.

**Benefits of Using Patterns for Software Discovery**

1. **Enhanced Accuracy**:
   * Tailor detection for specific software with unique characteristics.
2. **Detailed Information**:
   * Collect version, edition, and installation details critical for SAM.
3. **Compliance Management**:
   * Integrate with SAM to ensure accurate tracking of software usage and licenses.
4. **Cost Optimization**:
   * Detect underutilized software for reclamation or cost reduction.

Bottom of Form