CA IT Client Manager

Asset Management Administration Guide

Release 12.8

CA Technologies Product References

This documentation set references to the following CA products:

- CA Advantage® Data Transport® (CA Data Transport)
- CA Asset Intelligence
- CA Asset Portfolio Management (CA APM)
- CA Common Services™
- CA Desktop Migration Manager (CA DMM)
- CA Embedded Entitlements Manager (CA EEM)
- CA Network and Systems Management (CA NSM)
- CA Patch Manager
- CA Process Automation
- CA Business Intelligence
- CA Service Desk Manager
- CA WorldView™
- CleverPath™ Reporter

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Welcome to CA ITCM Asset Management

CA IT Client Manager provides sophisticated IT asset management functionality specifically designed to manage heterogeneous environments. It enables the IT departments to reduce service costs by enforcing standards and policies on their hardware and software asset configurations.

This guide provides technical assistance for the administrator to perform the following tasks:

- Configure the common configuration policies for asset management specific settings
- Create software signatures, export software signatures, and import software signatures
- Schedule jobs and collect tasks
- Create queries
- Define policies
- Create and configure collection modules
- Configure files for collection
- Configure Asset Collector, Agent Bridge, and Asset Converter to collect inventory data from various sources
- Integrate Intel AMT with CA ITCM
- Run inventory on a virtual host and collect inventory information (when a virtual host inventory collect task has been configured and enabled for that virtual host)

Overview and Features

Asset management provides the following system management functions that help you to manage assets in the most complex IT environments. The robust management database (MDB) can identify and deploy the agents to all the IT assets in your network environment using the built-in deployment wizard.

After the agents are deployed, you can perform the following tasks on the computer assets:

Hardware Inventory

Detects and reports detailed inventory information such as serial numbers of the hard disks, CPU information, RAM, internal and peripheral disk drives, OS versions and service packs, network settings, and power settings.

Hardware inventory is collected based on the inventory detection modules configured on the agents.

These modules let you do the following:

- Collect the System Management BIOS (SMBIOS) inventory by default.
- Customize and create new modules for discovering specific hardware and network information about the assets.
- Scan the target computers for compliance with the FDCC checklists.
- Support Windows Management Instrumentation (WMI) standard for collecting inventory and configuration information.
- Track changes to the assets and maintains a history with the date and time of change and the previous and current values.

Software Inventory

Detects the software applications, regular or virtual, installed on the managed assets and reports detailed and accurate information. Asset management can collect the software inventory using one of the following methods:

Software Signature

Scans the agent computers using the signature database. The signature database with predefined signatures packaged with asset management gives you (IT administrators) a quick overview of which applications are installed where, right after installation. You can also create software signatures for the software specific to your business and download new and updated signatures over the internet from a central CA maintained database. By default, these updated signatures are automatically downloaded once a day by the Default Software Contents Download job.

Intellisigs

An Intellisig is a script that detects software installed on agent computers. Intellisigs provide the most flexible and accurate way to detect software compared to the heuristic and software signature scanning. Intellisigs extract the software definition information from a defined source that the software manufacturer provides. The source can be a text file, database, registry, or binary file. For example, dsmver –f <output file> is the command that outputs version information about CA ITCM. Unless the manufacturer changes the way version information is stored for the product, Intellisigs can detect the future versions of the product also.

Heuristic Scanning

Scans the Windows Add or Remove Program database, MSI database, and shortcuts in the desktop and Start menu for the installed applications. On Linux/UNIX agents, heuristic scanning is performed on the PIF, PKG, and RPM databases.

Polite Scan

Specifies the process priorities for the signature scan and the asset management agent. This helps in minimizing the hard drive load during the signature scan.

Note: Polite scan can be configured on Linux or UNIX systems only. On Windows agents, by default, polite scan is optimized to minimum impact with fast execution and minimum CPU time.

Software Usage Monitoring

Software Usage Monitoring allows control over the usage of a specific application. The software usage agent can do the following:

- Monitor usage patterns and reallocate any unused software eliminating the need for purchasing a new license for a user who needs it.
- Categorize applications into suites and ensure accurate license tracking.
- Manage applications even if the assets are offline by logging usage information locally.
- Prevent the users from accessing the application when the number of licenses exceed the given license count.
- Put users in a queue when the number of licenses exceed and allow access when a user closes the application.

Network Inventory

The general inventory module also scans the network configuration such as the TCP/IPv4, TCP/IPv6, WINS, and DNS configuration on the managed assets.

Note: The IPv6 inventory does not include any information about gateways on Windows XP and Windows 2003 Server. It provides gateways information only on Windows Vista and later versions of Windows.

Queries

You can query the database for specific information that you need. You can query the job or collect task status and search for inventories or assets matching specified criteria. You can also create a policy based on a query.

Policy-based Management

Asset management manages the assets connected to it by enforcing certain policies. These policies let you focus on critical data when certain threshold values are exceeded. You can create query-based and event-based policies and specify the action that needs to be taken when a policy is violated or the event has occurred. You can take advantage of the predefined event-based policies as well.

Legacy Agent Support

The AM Agent Bridge provides support for all legacy Unicenter Asset Management 4.0 agents.

Asset Collector

The Asset Collector collects the hardware and software inventory information from well-formed inventory files. You can also track the origin and trustworthiness of this inventory. Using the Asset Collector, you can create inventory information for various sources such as proprietary devices that are not currently supported by CA IT Client Manager, Mainframe computers, and so on.

CA Asset Converter for Microsoft SCCM

The CA Asset Converter for Microsoft® SCCM is a component that connects to a Microsoft System Center Configuration Manager (SCCM) and extracts information, including hardware and software inventory, about the computers stored in its database.

Virtual Host Inventory

The virtual host inventory feature collects and reports inventory data on virtual hosts from a CA ITCM agent on a remote machine.

Non Resident Inventory

The Non Resident Inventory (NRI) solution provides a simple way to inventory a system without having to deploy the CA ITCM agent to it.

Robust Reporting

Asset management uses the DSM Reporter for creating various asset-related reports. These reports help in taking management decisions. Using the Query Designer, you can evaluate the data and specify the fields to be shown in the reports. You can print these reports or export to another application such as a CSV file or HTML file.

You can generate the reports manually at any time or schedule them to be generated at a specific time using the DSM Reporter GUI. The system engine generates the scheduled reports at the specified time.

More information:

Collect Tasks (see page 149)
Custom Software Signatures (see page 45)
Configure Polite Scan After Installation (see page 45)

Benefits of Using Asset Management

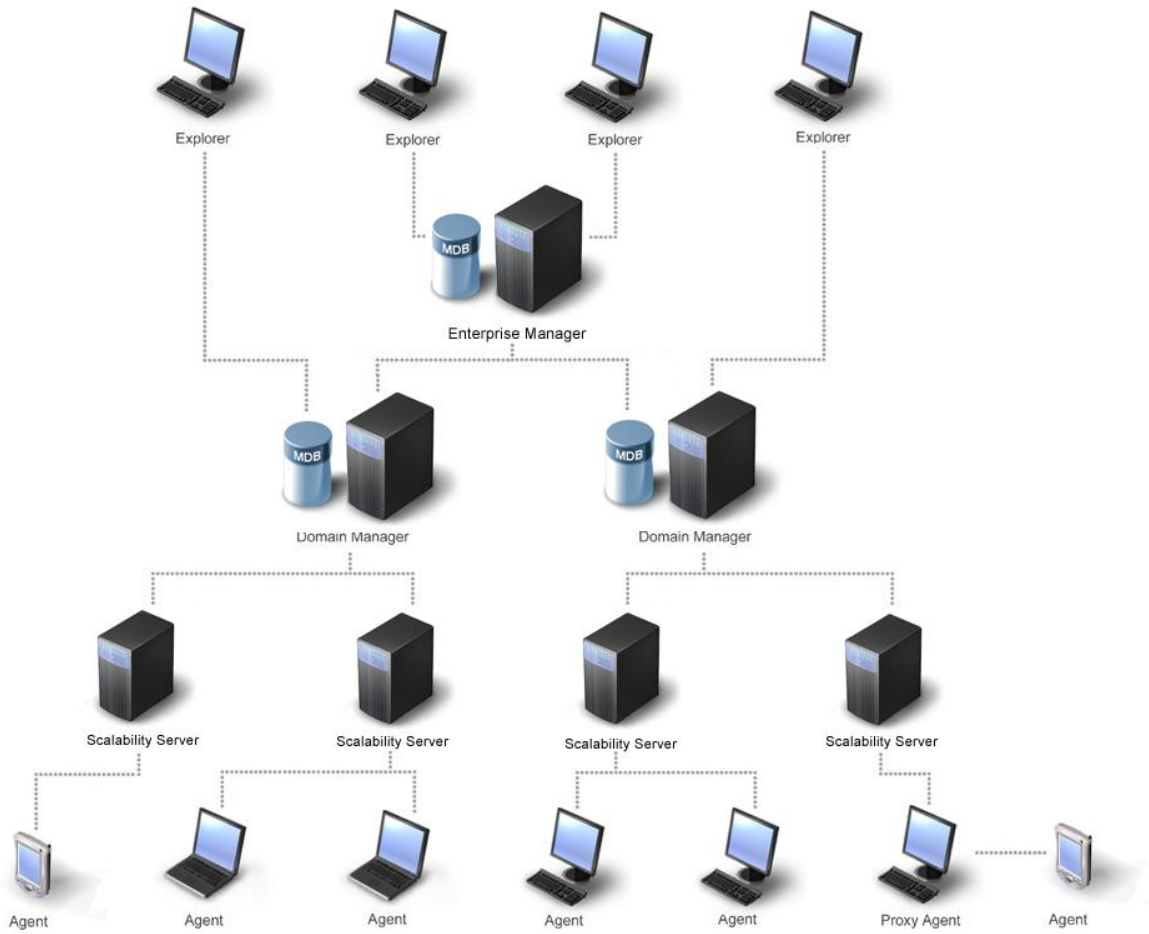
Asset management provides a variety of management tools that let you monitor, automate, and maintain your systems management functions in a number of ways and thus reduces costs. These tools also provide instant knowledge of what assets are deployed, and delivers full-featured asset tracking capabilities through automated discovery, hardware and software inventory, configuration management, software usage monitoring, software license management, and extensive cross-platform reporting.

Chapter 2: Architecture

This chapter presents an overview of the architecture of asset management, including brief descriptions of its various components and some features.

The following graphic shows the DSM architecture:

DSM Architecture



The scalable multi-tier architecture of CA ITCM can accommodate any size organization, large or small. The architecture is robust, as all asset information is managed at each domain manager, and if replicated to the enterprise manager, the assets can be managed from the enterprise manager according to the standard replication rule.

This section contains the following topics:

Enterprise Manager

The enterprise manager lets you synchronize and concentrate data from all of its connected domains. In distributed installations, it lets the headquarters maintain an overview of the subsidiaries. The enterprise manager helps you perform the following tasks:

- Access multiple domains and their data from the enterprise manager. For example, you can access the inventory, user information, jobs, and alarm definitions of all the connected domains.
- Perform consolidated actions that apply to either selected or all of the connected domains. For example, you can configure policies that you want to apply on selected or all the domains.
- Prepare reports that include the data from the domain managers that are connected to the enterprise manager.

Note: You can have only one enterprise manager in an environment.

Enterprise Manager Components

The enterprise manager contains components that manage, control, collect, distribute, process, maintain, and hold data used by asset management. The following are the basic components of the enterprise manager:

Enterprise DSM Explorer

Provides an enterprise management interface for scheduling requests and actions, making queries, and establishing policies.

Enterprise DSM Reporter

Provides a reporting tool to create reports using replicated data or targeted domains.

Enterprise Engine

Includes an enterprise engine that performs the Contents Download job and maintains query-based groups or policies at the enterprise level.

Enterprise Database

Includes a database for storing information about the assets (computers and users) that are collected by the engine from the connected domains which is viewable from the Enterprise DSM Explorer.

Enterprise DSM Explorer Access

From the enterprise explorer, you can access all domain managers:

- From an overview window, you can access the enterprise manager and all the domain managers.
- The enterprise explorer can open windows for working with a particular domain manager. You can drag-and-drop objects between domain manager and the enterprise manager.
- The enterprise manager and several domain managers can be open at the same time. Once a manager has been opened for a particular work session on the DSM Explorer, it need not be opened again.
- The enterprise explorer setup window lets you configure how to connect to each domain database, and to the enterprise manager.

All agents and users attached to asset management are controlled in one of the following ways:

- Directly by the domain DSM Explorer
- Indirectly by the enterprise DSM Explorer

Enterprise Engine

The enterprise engine performs the following functions:

- Performs the Contents Download job
- Maintains query-based groups or policies at the enterprise level
- Sends query results to the DSM Explorer
- Schedules report templates

Note: You can run the engine on the same computer as the DSM Explorer. However, we recommend that you run the engine on a separate computer to improve the response time when making queries and processing inventory data that is updated from multiple domains.

Domain Manager

The domain manager runs on a server class computer on the network and performs the following tasks:

- Maintains configuration information of the managed assets in the domain.
- Configures the security for asset management users who can connect to the domain manager and the extent of permissions they have on each object.
- Communicates with the scalability server to push the jobs and collect tasks scheduled on the assets.
- Stores the hardware and software inventory collected from each agent computer.
- Maintains the software usage information.
- Triggers the engine to collect data stored in the scalability server.
- Queries the database and applies the policies.
- Provides a powerful reporting tool.
- Provides a simple way to inventory a system without having an agent installed on the system (Non Resident Inventory).

Domain Manager Components

The asset management domain consists of the following components, which are used to organize and manage information for the domain manager:

Group

Denotes the assets defined as a collection, primarily for configuration and scheduling. A managed asset can belong to one or more groups.

Dynamic Group

Includes the assets that fulfill the defined criteria and are found dynamically by making queries of the domain database. For any query, one or more dynamic groups can be defined directly from the DSM Explorer.

Scalability Sector

Denotes the domain space for system transactions. You can create this server for each network operating system, protocol, topology, or segment.

Asset

Denotes the smallest component. It can be a single computer or a user.

Storage Areas

By default, asset management uses two storage areas to store inventory data. These areas are the domain database and the scalability server.

Domain Database

Stores the data collected by the asset management agent. The engine retrieves the data from the scalability server and updates the domain database. The DSM Explorer uses the domain database for storage and retrieval of inventory and configuration data from the assets. The database can be located locally or on a remote system.

Scalability Server

Stores the data collected by the asset management agent temporarily in a hard drive. It is a temporary storage area linking the domain database and the agents. It is used for storing management transactions scheduled on computers and users, and to hold status data on the results of such transactions executed by the agent. To meet the requirements of your network, you can have a scalability server based on network operating systems, transport protocols, or the geography of your enterprise.

Engine

The engine in the domain manager performs the following tasks:

- Performs the data processing tasks for the domain manager. For example, the engine can update the domain database with the status of actions, or make queries against the domain database to find data.
- Performs the Contents Download job of software definitions.
- Performs the replication (pull and push) of the objects between the domain it serves and the enterprise. This is applicable if domain is a member of an enterprise.

The engine is a CAF plug-in and is started automatically at startup time. The engine is generally scheduled to run always.

Scalability Server

The scalability server provides a point of buffering and resilience between end systems (agents) and the manager and performs the following functions:

- Communicates with the domain manager to get the jobs and tasks scheduled for the managed assets.
- Provides the job and scheduling configurations to the assets.
- Stores the inventory information collected from the agents.

There may be a large number of agents to be managed, too many for a single manager to handle and also expensive as each agent will need a database client to provide data to the database. In distributed environments, the scalability server is inserted to handle huge amounts of agents in geographical segmentations. This avoids overloading the network with traffic to and from the manager as the servers act as intermediaries and relay information to and from the manager.

The scalability server has a collection of directories and files known as sectors, in which agent data files are stored. When the agent collects data, it is stored on the scalability server for processing by the DSM engine. The engine collects the information from the scalability servers and updates the domain database.

In distributed environments, the scalability server is inserted to handle huge numbers of agents in geographical segmentations to avoid overloading the network with traffic to and from the manager.

Agent

The agent resides on the managed asset and performs the tasks scheduled for a computer or user. The agent can perform many tasks, such as hardware and software inventory scanning, software distribution jobs, and remote control activities.

Primarily, the agent performs the following functions:

- Checks the scalability server for jobs and collect tasks scheduled on the computer where the agent resides.
- Runs the jobs and collect tasks.
- Saves the collected data in the scalability server's collect area.

You can deploy the agents by running either the product installation wizard (from DVD or a share) or the Deployment wizard. The agent must be deployed on the computers that need to be managed by the asset management system. You can use the deployment wizard to deploy the agents based on any of the following criteria:

- Computers in a domain
- Computers within an IP address range
- Specific computers
- Computers in an LDAP directory (including Active Directory).

After the agent is deployed, the asset appears in the list of managed assets. Computers in this list are ready to run the asset jobs and collect tasks. The information collected can be extended through the use of the Management Information Format (MIF).

More information:

Management Information Format (.MIF) Files (see page 219)
Non Resident Inventory (see page 353)

Types of Agents

There are four types of asset management agents:

- Windows agents
- Linux agents
- Solaris, HP-UX, AIX, and MAC OS X agents
- Docked Device Proxy agents for Palm and Windows CE Devices

Windows Asset Management Agents

The Windows agents contain control programs for computers running under various operating systems. For the supported operated systems, see Certification Matrix.

Note: A control program starts whenever you start a managed asset or log on to the asset.

The agent software has two parts; one starts when the managed computer starts the network operating system to get system-related data and the other starts when a user logs on to the network to get-user related data.

Note: For the most current list of supported platforms, see the *CA IT Client Manager Readme*.

Linux/UNIX Agents

The Linux/UNIX agents contain control programs for computers running on Linux and UNIX operating systems.

Note: A control program starts whenever you start a managed asset or log on to the asset.

The agent software has two parts; one starts when the managed computer starts the network operating system to get system-related data and the other starts when a user logs on to the network to get user-related data.

Docked Device Proxy Agents for Palm and Windows CE Devices

With CA ITCM asset management, you can perform inventory, distribute software, and schedule various tasks using mobile devices. These jobs can be executed when the mobile device user temporarily returns to the user's computer to exchange or synchronize information between the mobile device and the computer. Typically, the mobile device will be hooked up through a "cradle" connected to the serial port of the computer.

The platform supported by asset management agent software is the Microsoft Windows CE/Pocket PC series of devices running under the Windows CE operating system.

Note: The Microsoft Activesync component must be installed before enabling docking device support on a host PC. Also, the Microsoft Activesync component must be made aware of the changes to the PATH environment variable done during the installation of CA ITCM. After the CA ITCM installation, you must log off and log in again to set the required environment variable. For more information about Activesync and Palmsync integrations and their supported versions, see the *Implementation Guide*.

How to Start the Asset Management Agent

When the asset management agent is started, it runs the collect tasks and asset jobs configured on the agent computer.

You can start the asset management agent in *one* of the following ways:

- Automatically – In the case of Express Installation, the agent is configured to run once a day at 12:00 a.m. You can change this interval by performing a custom installation, and specifying the time interval or the specific hours when you need the agent to run.
- Manually – Use one of the following methods:
  - Right-clicking the computer asset in the DSM Explorer and selecting Asset Jobs, Activate Job Check.
  - Running "caf start amagent" on the local computer with sufficient privileges.

Note: The asset management agent starts the scan at the scheduled time and collects the information with Local System Account (Windows) or root (Linux/UNIX). When a user logs on, the agent starts collecting the user-related data such as network connections and user template inventory.

Agent Working Directory

On Windows computers, the agent's working directory is available in the following paths:

- ...\Agent\units\00000001 folder has the data collected by the agent with the Local System Account.
- ...\Agent\units\00000003 and all the folders with odd numbers have the data collected by the agent from a user's login. Asset management creates a folder for each user on the computer.
- ...\Agent\units\00000002 and all the folders with even numbers are used by software delivery

On Linux and UNIX computers, the agent's working directory is typically available in /opt/CA/DSM/Agent/AM/data/transfer folder. However, the install path may be customized.

DSM Explorer

The DSM Explorer is the graphical user interface (GUI) for asset management that organizes the objects into appropriate categories and displays them in a tree structure. From the DSM Explorer, you can perform the following tasks:

- View and manage information in the database
- Query the database
- Schedule and view jobs to be executed on a computer, user, or group of computers or users, such as:
  - Hardware and software inventory
  - Software usage and configuration control
  - Message display
  - Forced program execution
  - Software upgrade
- Set up policies
- Perform auditing functions
- Control computers, users, and agents
- Create software definitions

It also has an embedded Online Tutorial that guides you through various procedures.

Asset Management Objects in the DSM Explorer

You can view and configure the asset management objects from many places in the DSM Explorer window.

Asset management lets you create and configure the objects at the following levels:

Group Level

Objects created at the group level are automatically linked to the member assets. If you disable or enable an object at the group level, it is enabled or disabled for all the member assets also. Objects disabled at the group level can be enabled only at the group level.

Computer Level

Objects created at the computer level are linked to the respective computers.

User Level

Objects created at the user level are linked to the respective users.

Domain Level

Objects created at the domain level can be linked to any asset or group. At the domain level, you can see all the objects created in the domain including those at the group and asset level.

Asset Management Security Objects Classes

Asset management security object classes deal with the access rights to the asset management objects. All asset management users need appropriate access rights to these object classes to login and perform various functions in the DSM Explorer:

- Asset Group
- Computer
- Health Monitoring Alert
- Security Profile
- Class Permissions
- Security Area
- Database Credentials
- User Account
- User Profile
- External Asset
- Software Category
- Software Definition
- Software Group
- Software Package
- Procedure Group
- Procedure
- OS Installation Image
- Software Job Container
- Software Job
- Asset Job

- Common Query
- Policy - Query Based
- Policy - Event Based
- Control Panel Access
- Engine
- Engine Task
- Inventory Task
- Software Discovery Task
- Software Usage Task
- Template Task
- Virtual Host Inventory Task
- Deployment Job
- Domain Group
- Domain
- Scalability Server Group
- Scalability Server
- Manager
- Report Template
- Report Scheduling
- Policy - Configuration Computer
- Policy - Software Based
- Configured Directory

Note: For more information about common security and the object classes, see the *DSM Explorer Help* and the *Implementation Guide*.

Common Application Framework (CAF)

Each of the DSM components uses the common application framework (CAF). CAF is a cross-platform service controller that provides a single point of control for all DSM components.

CAF dynamically provides DSM services as required using an extensible plug-in model. Each CAF plug-in is a program that provides agent, scalability server, or manager functionality. A CAF plug-in can also be an extension of CAF itself and provide some common service, for example, registration with scalability servers or system event detection.

Normally, CAF starts all plug-ins automatically at boot time. CAF can also start and stop plug-ins on demand from the command line and at particular times and regular intervals using its scheduler. For a description of how to specify scheduled jobs that run in CAF, see the "CAF Scheduled Jobs" appendix.

CAF is also able to query plug-ins for status information and route messages from other plug-ins.

Important! On Windows, CAF is installed by default to log on as the Local System account. If for security reasons you need to change this Net Logon property, you must do this after the installation using the Computer Management, Services console from the Windows control panel. However, changing to an account with reduced privileges may result in unexpected behavior or reduced functionality of CA IT Client Manager.

Note: For more information about CAF and its plug-ins, see the *Implementation Guide.*

CA Message Queuing (CAM)

The CA Message Queuing Service (CAM) is a component used by CA products as a lightweight messaging service providing high performance, low resource utilization, reliable, scalable application-to-application messaging. The asset management agent uses CAM for communicating with the scalability server. By default, CAM uses UDP port 4104 for communication.

Comstore.xml

The comstore.xml file stores all the configuration data for all the CA ITCM components. Each agent has a copy of this file with its own configuration data after installation.

You can modify these configurations using the common configuration policy settings. After any updates, comstore.xml is converted to comstore.enc. Subsequent CCNF operations update comstore.enc only.

To decrypt the comstore, use the following commend:

```
ccnfcmda -cmd GetConfigStore -fi filename.xml
```

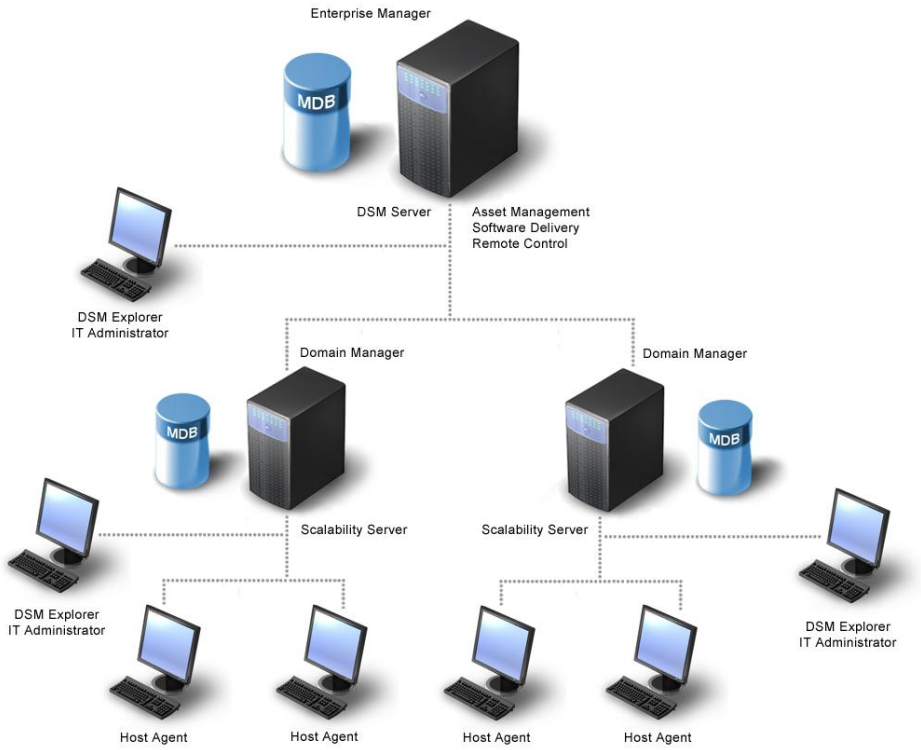
Important! Entering incorrect values in the comstore.xml will have adverse impact on the functioning of the CA ITCM components.

Chapter 3: Implementation

This chapter discusses the implementation and practical considerations before installing asset management in large organizations. A large organization in this case is considered a site with more than 500 computers. Asset management lets you manage large computer networks by distributing the storage and processing of data across different components and servers.

CA ITCM has a common architecture for all the plug-ins (asset management, software delivery, and remote control). This chapter gives you an insight into the asset management specific implementation considerations of this architecture.

The following graphic shows the CA IT Client Manager architecture:



Note: For more details about installation prerequisites and installation steps, see the *Implementation Guide*.

Selection of Servers

Because asset management lets you distribute its software and database components across multiple servers, you must select which servers in your network are suitable to use for implementation.

Note: For more information about the server configurations, see the readme.

Selection Criteria

There are two criteria for selecting the servers for asset management implementation:

- The servers should be capable of supporting many concurrent users without seriously degrading server performance.
- The servers should have enough disk space for storage of asset management specific files and database information.

Impact on Network Performance

By configuring the asset management system with more than one scalability server, you can minimize the asset management traffic that must traverse WAN segments in a mixed LAN/WAN environment by always having at least one scalability server available on the local LAN segment, seen from each agent.

Engines

Several engines can be introduced into the asset management system to facilitate the collection of data from many scalability servers into the domain database. Such engines can be set up with different configurations to match the precise need for having up-to-date data in the domain database.

Agents

The asset management agent can be installed in either of the following ways:

- Using the Deploy CA ITCM Agents or Scalability Server Wizard
- Using the CA ITCM Installation DVD

Each agent must be configured to connect to the relevant scalability server. The agent stores data locally in the agent working directory. It compares the detected computer or user data with the previous scan results stored locally, and transfers only the difference to the scalability server (typically less than 20 KB).

More information:

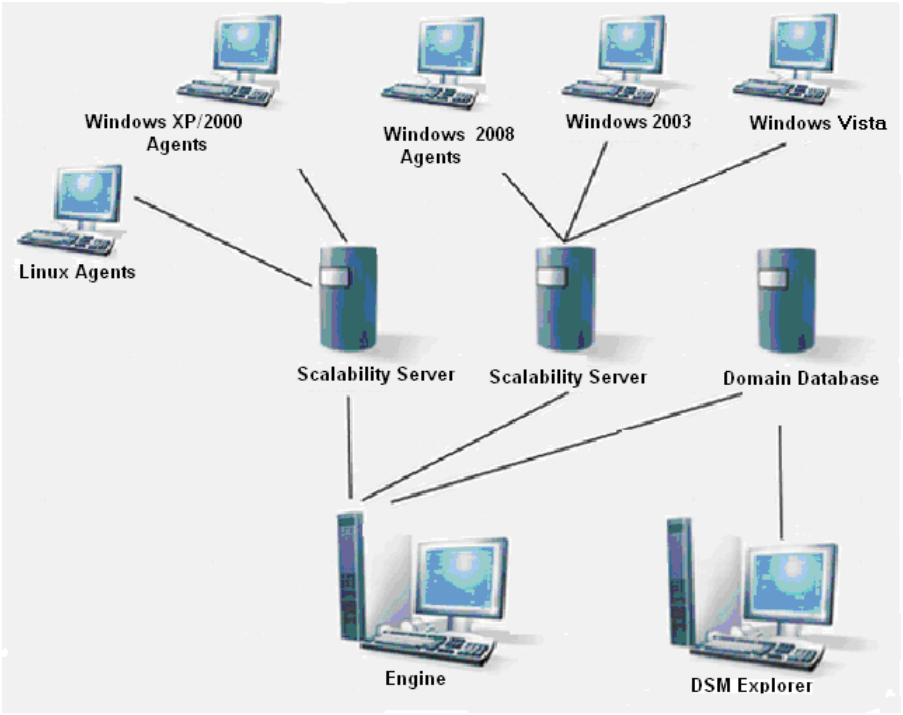
Agent (see page 27)
Agent Working Directory (see page 29)

Storage Areas

To avoid unnecessary bottlenecks that can arise from directly sending the data from the agent to the domain database, you can divide storage into two different areas.

Instead of connecting to the Domain database, each managed asset connects to a scalability server and copies the data to it in a compressed ASCII text format. The engine then picks up the ASCII text file from the scalability server's collect area and processes it so that it can be put into the database structure of the domain database.

The following graphic shows how the DSM Explorer, the engine, and agents are related to the scalability server and the domain database:



The previous graphic illustrates the connections of agents and the engine to the scalability server:

- Agents connect to the scalability server to access management transactions and to report hardware and software data and their status information.
- The engine connects to the scalability server to update data to and from the domain database and to get status information on scheduled and completed actions.

Practical Considerations before Installation

You should be familiar with the preceding sections before reading the following information on practical considerations in reference to installation, use, and maintenance of an asset management system in a multiple-site or large-site environment. New terms included in this section are as follows:

Main site—The site with the enterprise and domain database.

Site—A remote site connected to the main site through a WAN link.

Planning the Installation

Before you install asset management, consider these questions:

- How many sites are there?
- What are the server names and types at the sites, both remote and main?
- What are the speeds of the different WAN links?
- Which protocols are transmitted through the WAN links?
- Will sufficient bandwidth be available for users during office hours or should asset management be configured to use the WAN links only at night?
- Where will the DSM Explorer users be placed—at the main site only or also at remote sites?

Selecting the Best Possible Main Site

The optimum choice for the main site in a large organization is the site where most of your DSM Explorer users will be placed. The domain database should reside at this location.

Select a site where most support and management activities will be carried out. Frequently, this is also the site with most of the asset management agents. The reason for this recommendation is that most database access takes place from the DSM Explorer to the domain database and from the engines to the domain database. (The engines are usually placed in a room designed for technical equipment, and close to the IS department.)

Number of Scalability Servers

Before you decide on the number of scalability servers, you should consider the following:

- Number of computers and users (assets) per scalability server
- Number of sites
- Number of files that the server can handle per directory (dependent on network operating system)
- How licensing is done on the servers (dependent on network operating system)

Computers Per Scalability Server

You can easily handle 1000 computers per scalability server (if you are using only asset management). The asset management agent uses CA Message Queuing (CAM) to communicate with the scalability server.

Scalability Servers Per Site

You will need at least one scalability server per site, as agents should never connect through WAN links. If agents are connected through the WAN link, it will cause unnecessary delay when running the agent.

Files Per Directory

Some network operating systems ensure high performance on the server only if you do not exceed a certain number of files in a directory.

There will not be more files in one directory than the number of computers and users in a given server. The sole exception is the collect directory, but as the number of files in the directory is temporary, this does not pose a concern.

Licensing on Servers

Some servers are licensed by connection. In an environment of this kind, you may want to have at least one scalability server per login server, to ensure that you do not exceed the number of licenses on any server at any time.

Scenario

A medium-sized fresh-food manufacturer has a few hundred employees, with four (4) major offices spread across North America. Three (3) of their offices are in the United States (Boston, New York, and Portland, Oregon) and one is in the Canadian city of Vancouver. About 75 to 100 employees work in each of these offices. The company also has a smaller satellite office in Austin, Texas, that joined the company through a recent acquisition. About 20 to 30 people work in the Austin office.

The main branch is located in New York, and it also serves as the company's headquarters. The enterprise manager is located here, enabling central management of all aspects of the business and monitoring of all the branches.

Although the four offices are geographically distant, they fall into two distinct regions. The Boston and New York offices are both in the Northeast region of America, while the Portland and Vancouver offices are both in the Northwest. So, a decision was made to place one scalability server and one domain manager in each region. The two scalability servers and domain managers manage the day-to-day tasks of both of their respective branches.

Each region would have its own domain administrator, who would deploy the asset management agent on the computers in their network, create security profiles and specify who can access the DSM Explorer and to what extent and oversee the effective utilization of the IT assets. The administrator is also responsible for troubleshooting problems using the DSM Explorer on the regional domain manager.

The administrator or any other user with appropriate rights would do the following asset management specific configurations to suite your business requirement:

- Create and configure the collection modules to collect the additional hardware inventory.
- Create additional software signatures to be discovered by asset management.
- Enforce policies on the all the connected agents and have complete control over the agents. For example, the administrator can create a policy to check the successful execution of a job on all the agents and report any failure through email or by CA Service Desk Manager tickets.

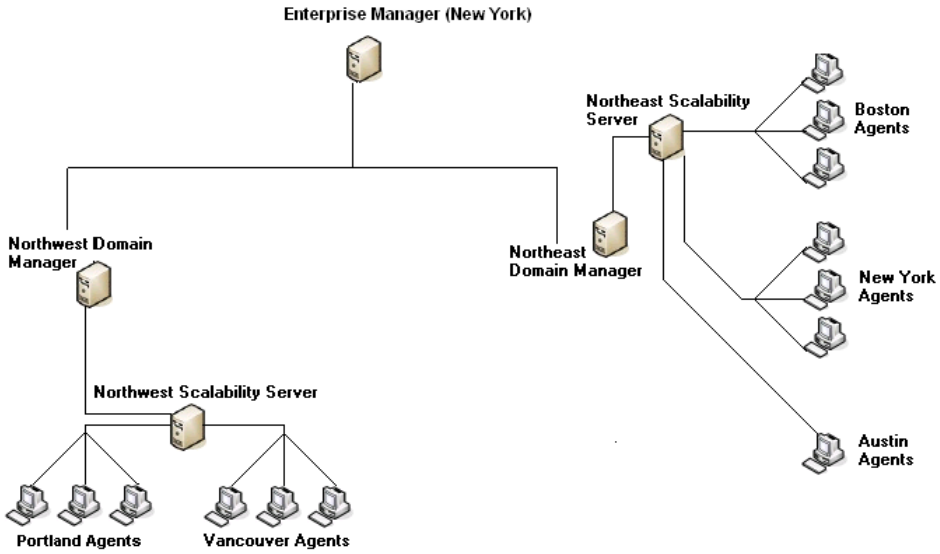
The only remaining question is, what do you do about Austin? It does not fall neatly into either region. Should those employees use the scalability server on the west coast, or the east coast? Or should Austin have its own scalability server, domain manager, and domain administrator?

The deciding factor is the number of users who will be connecting to the scalability server. Since there are only about 30 employees in the Austin office, it probably does not need to have its own scalability server. Instead, the Austin employees can send their data to the scalability server that has the fewest users.

If the needs of the employees increase, another scalability server can be added, either in Austin, or in one of the two main regions. Since CA ITCM is completely scalable, additional scalability servers and domain managers can be added as needed. The scalable multi-tier architecture of CA ITCM can accommodate any size organization, large or small. The architecture is robust, as all asset inventory information is managed at each domain manager location.

The entire architecture is controlled by the enterprise manager that replicates the data from and to the domain managers. It can push jobs and policies that must be run on all the computers in the domain.

The following graphic depicts how the asset management architecture might be implemented in the fresh-food manufacturing company:



Note: For more detailed information about implementing asset management, see the *Implementation Guide*.

Chapter 4: Configuring Asset Management

You can configure various settings for asset management to optimize and improve the performance of the asset management agent. These settings are maintained and controlled by the domain manager which you can configure using the DSM Explorer window.

This chapter discusses the various asset management objects that you can configure as per your business requirement to enhance the agent functionality.

- Grouping Assets
- Configuring the Common Configuration Policy for asset management-specific settings
  - Configure Integration to emailing System
  - Configure Integration to Service Desk
- Collection Modules
  - Inventory Detection Modules
  - Inventory Template Modules
- Creating Custom Software Signatures
- Configure files for backup
- Configure and schedule collect tasks

This section contains the following topics:

Grouping Assets

An asset group lets you logically group your computers and users into a group. All Computers, All User Accounts, and All User Profiles are the default groups for computers, users, and user profiles, respectively. By grouping your assets into an asset group, you can do the following:

- Query the member of the group.
- Push certain jobs or tasks specific to the group.
- View the computers in the group that had violated policies.
- Specify group level permissions. For example, if you have various departments in your organization, you can probably group the assets based on the department code and give full permissions to the Department Manager for the group.

An asset group can be one of the following:

Static Groups

Contains members who are manually added to or removed from the group.

Dynamic Groups

Contains members who are dynamically added or removed based on the query result. Dynamic groups are based on a query.

Polite Scan

Polite scan lets you set process priorities for your software scan and the asset management agent. Polite scan minimizes the hard drive load during the file scan and thus reduces the I/O time consumed by the software scanner.

For Windows agents, polite scan is implemented for the best performance.

For Linux/UNIX Agents, polite scan can be configured either during installation or by modifying the comstore.xml file after installation, as described in the *Configure Polite Scan after Installation* section.

Note: Do not use an editor to modify the comstore.xml file.

Configure Polite Scan After Installation

You must configure polite scan to specify the priorities for the signature scan and the asset management agent. You can do this either during installation or after installation. Based on the priority set, the agent can run at any priority level.

Note: Polite scan can be configured only on UNIX/Linux agents.

To configure polite scan after installation

1. Execute the following command:
   `caf stop`

   The caf process stops.

2. Set the priority level of the agent using the following command:

   `ccnfcmda -cmd SetParameterValue -ps /itrm/uam/prio_value -pn Client -v Value`

   The value can range between -20 and 19:

   -20

   Indicates that the agent will run with high priority.

   19

   Indicates that the agent will run with low priority.

   0

   Indicates that the agent will run with default or normal priority.

3. Start the caf process using the following command:
   `caf start`

   The agent runs with the configured priority.

Custom Software Signatures

A *software signature* defines the attributes of a software application, such as the main executable file name, other associated files, size range, version range, creation, and modification dates of the software. All these attributes of a software signature uniquely identify a software application. Software signatures in asset management are created as software definitions. You can create software definitions for a product, release, patch, suite, suite component, or virtual application image. By default, asset management provides predefined software signatures covering the most widely used software in the IT industry.

Note: You cannot modify the predefined or CA Provided signatures.

Other than these predefined software signatures, you can also create new software signatures for discovering the licensed software you use or you can add new releases or patches to existing product definitions. Creating the software definitions at the domain level displays the discovered software information at the domain level only. However, if you create it at the enterprise level, the discovered software information is shown and is usable at both the levels—domain level and enterprise level.

You can create the following type of signatures:

Product

Defines the basis for a software definition. It is only a container component. It comprises of releases, suites. A product by itself does not have any meaning without any of these components. For example, Microsoft Outlook 2000 is a product. A product definition has only the version information.

Release

Includes a specific release of software. Release definitions can be created only for the products. For example, Microsoft Outlook 2000 SP2 is a release.

Patch

Includes the fixes for a release. Patches can be created only for releases. For example, Q303833 is a patch for Microsoft Outlook 2000 SP2 release.

Suite

Includes a software suite that comprises various individual and integrated products. For example, Microsoft Office. A suite definition can be created only for the products.

Suite Component

Includes the individual product in the suite such as Microsoft Word. This definition can be created only for the suites.

Virtual Application Images

Defines the basis for a virtual application image software definition. As CA cannot provide software definitions for recognizing virtual application images in the Content Download service, the DSM administrator must create these virtual application image definitions.

When the agent runs, it scans the computer for the specifications given in the software signature and recognizes the software only if all the specifications are met.

More information:

Create a Product (see page 48)
Create a Release (see page 49)
Create a Patch (see page 49)
Create a Suite (see page 50)
Create a Suite Component (see page 51)
Virtual Application Images (see page 110)
Signature Scanning for Virtual Applications (see page 112)

How to Create Custom Software Signatures

You can create a custom signature when a predefined signature is not available for software, for example, software uniquely used by your enterprise. You can also add new software releases and patches to an existing product.

Creation of custom signatures includes the following steps:

1. Create a product definition for the signature.
2. Add a release or suite to the product.
3. Add a patch to the release.
4. Add a suite component to the suite.

More information:

Create a Product (see page 48)
Create a Release (see page 49)
Create a Patch (see page 49)
Create a Suite (see page 50)
Create a Suite Component (see page 51)

Create a Product

Products are the basis for a definition. You must create a product before you can add releases or suites to it.

To create a new product

1. Navigate to the Domain, Software, Definitions, Categories folder in the DSM Explorer.

   The existing categories in the domain appear.

2. Select an existing category or create a new one by clicking New Category in the Tasks section.

   The existing definitions in the category appear.

3. Right-click the category and select New Product.

   The Create New Product dialog appears.

4. Enter the product name and version, and click OK.

   A message appears asking whether you want to create a release for the product.

   a. Click Yes.

      The Create New Release dialog appears where you can specify the details of a new release.

   b. Click No.

      The product is added to the category.

      Note: The product by itself does not have any meaning. You must create either a release or patch to give the specifications for the signature.

More information:

Custom Software Signatures (see page 45)

Create a Release

A release includes the software definition. It defines the parameters for uniquely identifying the release. Release definitions are used by the signature scanner while scanning the agent computers.

To create a new release

1. Navigate to the Domain, Software, Definitions, Categories folder in the DSM Explorer.

   The existing categories in the domain appear.

2. Select an existing category or create a new one by clicking New Category in the Tasks section.

   The existing definitions in the category appear.

3. Right-click the product to which you want to add the release and select New Release.

   The Create New Release dialog appears

4. Specify the details of the new release in the General, Recognition, and Exclude Options tabs, and click OK.

   A new release is created under the product.

Create a Patch

If you want to track the patch-level inventory data, you need to create the patches.

To create a new patch

1. Navigate to the Domain, Software, Definitions, Categories folder in the DSM Explorer.

   The existing categories in the domain appear.

2. Select an existing category or create a new one by clicking New Category in the Tasks section.

   The existing definitions in the category appear.

3. Double-click the product.

   The existing releases and suites in the product are displayed.

4. Right-click a release to which you want to add the patch and select New Patch.

   The Create New Patch dialog appears.

5. Specify the details of a new patch in the General, Recognition, and Exclude Options tabs and click OK.

   A new patch is created under the release.

More information:

Custom Software Signatures (see page 45)

Create a Suite

Suites include the suite definitions. When the signature scanner scans the agent computers, it returns both suite and suite component inventory. For example, you can view the number of Microsoft Office installations in your domain and also view the number of Microsoft Word installations.

To create a new suite

1. Navigate to the Domain, Software, Definitions, Categories folder in the DSM Explorer.

   The existing categories in the domain appear.

2. Select an existing category or create a new one by clicking New Category in the Tasks section.

   The existing definitions in the category appear.

3. Right-click the product to which you want to add the suite and select New Suite.

   The Create New Suite dialog appears.

4. Specify the details of a new suite in the General, Recognition, and Exclude Options tabs and click OK.

   A new suite is created under the product.

More information:

Custom Software Signatures (see page 45)

Create a Suite Component

A suite component is an individual product in the suite such as Microsoft Word. To get the suite component level inventory, you need to create and add the suite components to the suites.

To create a new suite component

1. Navigate to the Domain, Software, Definitions, Categories folder in the DSM Explorer.

   The existing categories in the domain appear.

2. Select an existing category or create a new one by clicking New Category in the Tasks section.

   The existing definitions in the category appear.

3. Double-click the product.

   The existing releases and suites in the product are displayed.

4. Right-click the suite to which you want to add the suite component and select New Suite Component.

   The Create New Suite Component dialog appears.

5. Specify the details of the new component in the General, Recognition, and Exclude Options tabs and click OK.

   A new suite component is created under the suite.

More information:

Custom Software Signatures (see page 45)

Organize Products by Categories

You can categorize your software definitions and organize them for easy retrieval based on the definition types, manufacturer, and category.

To organize products by categories

1. Navigate to Software, Definitions, Categories under the domain manager.

   The existing categories are displayed in the right pane.

2. Click New Category in the Tasks Section.

   The New Software Category dialog appears.

   Note: You can also use the existing categories.

3. Specify the category name and other details in this dialog and click OK.

   A new category is added.

4. Right-click the category and select Organize Category.

   The Organize Category dialog appears.

5. Filter the software definitions by specifying conditions, add them to the Selected Definitions, and click OK.

   The selected definitions are added to the category.

Add Product Components

You can organize the product by adding the following related components:

- Suites and releases to a product
- Patches to a release
- Suite components to a suite

The components created under a parent definition are automatically added to that definition. If you want to add these components to other definitions as well, you can add the product components accordingly.

To add the product components

1. Right-click a product, release, or suite and select Organize Product.

   The Organize Product dialog appears.

2. Filter the related components you want to add to the software definition, add them to the Selected Definitions, and click OK.

   The selected components are organized under the definition.

Enable or Disable Scan for a Definition

By default, all the software definitions are enabled for signature scanning. You can disable scan for a definition if you do not want the signature scanner to scan that release or patch.

To enable or disable scan for a definition

Method 1:

1. Navigate to Domain, Software, Definitions, Categories, All Definitions and select the required definition.

   The details about the selected definition appear.

2. Verify the Enable for Discovery column. This column can have either of the following values:

   Yes

   Indicates that the scanning is enabled for the definition.

   No

   Indicates that the scanning is disabled for the definition.

3. Right-click the definition and select Enable Scan or Disable Scan as applicable.

   A confirmation message appears.

   a. Click Yes.

      This enables or disables the scan for all the linked definitions. For example, for a product, clicking Yes disables or enables the scan for the linked releases, suites, and patches.

   b. Click No.

      This enables or disables the scan only for the selected definition.

Method 2:

1. Navigate to Domain, Computers and Users, All Computers, *Computer,* Software, Discovered.

   The software discovered in the selected computer appears.

2. Right-click the software you no longer want to scan, and select Disable Scan.

   The selected software immediately disappears from the list of discovered software, and the Enable for Discovery column under the All Definitions folder is set to No.

Note: Using Method 2, you can only disable the scan for a definition. To enable the scan, follow the steps in Method 1.

Intellisigs

Intellisigs—Software Detection through Scripts

An *Intellisig* is a script that detects software installed on agent computers. Intellisigs provide the most flexible and accurate way to detect software compared to the heuristic and software signature scanning. Intellisigs extract the software definition information from a defined source that the software manufacturer provides. The source can be a text file, database, registry, or binary file. For example, dsmver –f <output file> is the command that outputs version information about CA ITCM. Unless the manufacturer changes the way version information is stored for the product, Intellisigs can detect the future versions of the product also. While software signature scanning requires signatures for every version of a product, release, and patch, a single Intellisig can detect releases and patches of multiple software products. For example, you can create an Intellisig to detect all versions of Microsoft Office products, including past and future versions.

Note: You can use Intellisigs to detect software on various platforms.

The following Intellisig types are available:

CA Intellisigs

Includes Intellisigs that CA provides. Similar to the CA-provided software signatures, the CA-provided Intellisigs are downloaded as part of the Content Download job.

Custom Intellisigs

Includes Intellisigs that you create, based on your requirement. You create these Intellisigs and import them using DSM Explorer.

Note: Intellisigs do not support software usage (metering), both offline and online.

DSM Reporter does not support Intellisigs reports.

More information:

Intellisig—An Overview (see page 55)
What is an Intellisig? (see page 56)

Intellisig—An Overview

A traditional signature only looks for the specific files of specific version and size that are part of a release or patch. It is a fixed rule that cannot adapt itself to changes in the software being detected. An Intellisig is implemented in code such that it can embody as much intelligence and flexibility as required. This intelligence and flexibility allows an Intellisig to detect many more items than a single signature can. For example, a single Intellisig can detect all Microsoft Office products of all versions, as opposed to a large set of signatures, one for each possible item that could be in Microsoft Office.

Dmscript is a portable scripting language and processor that is delivered with CA ITCM. The dmscript language has been extended with functions to deliver the output in a standard simple format.

To write an Intellisig, perform the following tasks:

1. Analyze the Software Identification Information
2. Code the Intellisig.
3. Test the Intellisig.
4. Release the Intellisig for production use.

Following are the outputs of an Intellisig:

- A number of product definitions, for example, Microsoft Windows. You do not have to create products in the MDB before they can be recognized. An Intellisig creates the definitions as it runs. This feature allows new products to be discovered.
- A number of release definitions of the products, for example, Windows 7 Ultimate 32bit ENU.
- A number of patch definitions applying to the releases. These fixes are published and applied to installations. For example, KB123456.
- One or more instances of an installed release or patch.

This output appears in DSM Explorer under the Discovered Software and Software Definitions nodes.

What is an Intellisig?

An Intellisig is composed of the following components:

- A script written in dmscript
- Any number of ancillary files containing data, operating system dependent scripts such as shell scripts, or other dmscripts
- Metadata describing the Intellisig, name, version, timeout, and triggers

These components are stored in the MDB and are distributed to scalability servers and agents. An Intellisig runs under the control of dmscript, which is launched by the software scanner. The scanner, in turn, is launched by the CA Asset Management agent.

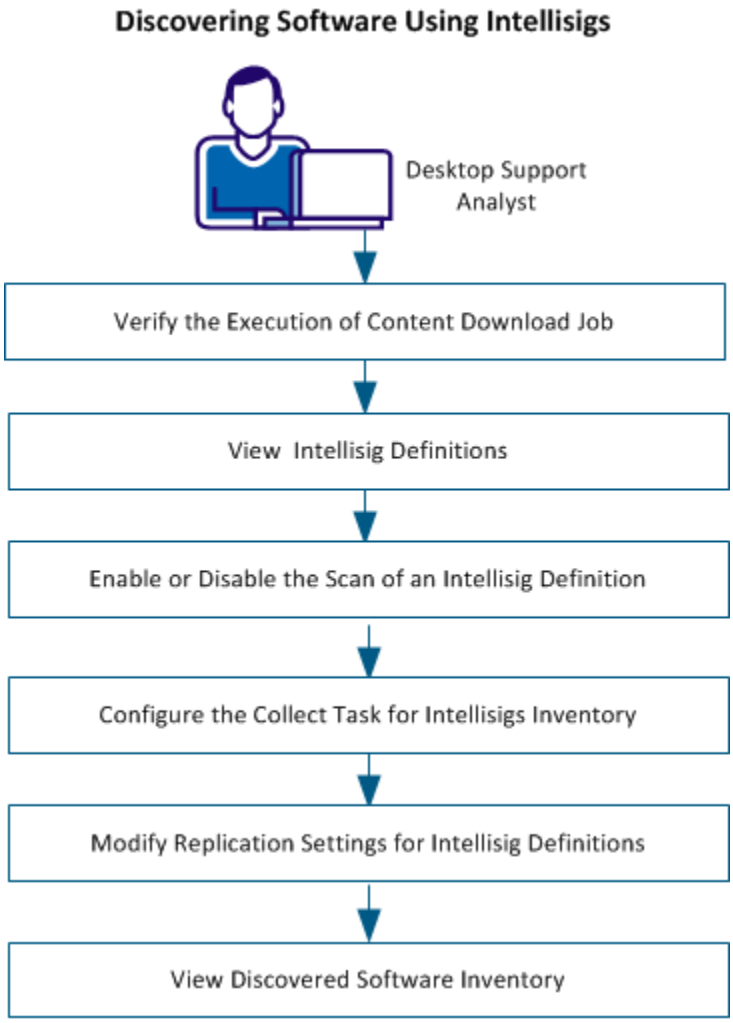
The output of the script is an XML file called <Intellisig-uuid>.xml. The format of this file follows the ISO19970-2 standard for software ID tags. It is an international standard that provides a means for software products to store information about themselves. This information is then read by products such as CA ITCM and converted to the regular CA Asset Management inventory format by the software scanner.

The domain manager sends the current set of software Intellisigs to the scalability servers, which, in turn, send them to the agents. When the CA Asset Management agent runs, it checks if it is configured with a software collection task. If configured, it starts the software scanner, which in turn, runs each available Intellisig. For each Intellisig, any available triggers are evaluated to check if the script should be run. If the trigger is true, dmscript is run with the appropriate arguments. When complete, the agent collects the output XML file, converts it to the inventory format, and uploads it to the server and back to the domain manager. After it is stored in the MDB, the results are available in DSM Explorer. The agent performs traditional signature and heuristic scans too if configured to do so.

To see working Intellisigs, use DSM explorer to view the contents of the standard Intellisigs that are provided by CA, and are downloaded from the content server. You can use these Intellisigs as models for developing your own Intellisigs.

Discovering Software Using Intellisigs

As a desktop support analyst, you want to discover software on target computers so that you get an inventory of software installed on the target computers.



Verify the Execution of Content Download Job

The Default Software Contents Download job downloads the Intellisigs that are provided by CA from the CA website. Verify that this job has been executed at least once from the time you installed this release. The successful completion of the job helps ensure that the CA-provided Intellisigs are downloaded to your domain manager.

Follow these steps:

1. Navigate to Control Panel, Engines, All Engines, System Engine.

   The engine log and Task List is displayed on the right pane.

2. Verify that the Status of the Default Software Contents Download job displays OK and it was Last Executed after you installed this release.

   If either of these conditions is false, you can either run the job immediately or wait till the next scheduled time of the job. If both the conditions are satisfied, you can view the downloaded Intellisig definitions.

   Note: If you want to create and use custom Intellisigs, see Creating Custom Intellisigs (see page 65).

View Intellisig Definitions

An Intellisig includes a master definition and several detected definitions. A master definition is the first-level object that contains the Intellisig details such as name, creation details, target platform, and the script details. The detected definitions include the product, release, and patch definitions that are created only after the Intellisig detects them on a target computer. After you download Intellisigs, you can only view the master Intellisig definitions and their details. The detected definitions are available only after the Intellisigs collect task detects the software on a target computer. To view the Intellisig definitions, click Software, Definitions, Intellisigs in DSM Explorer.

Enable or Disable the Scan of an Intellisig Definition

By default, all the Intellisigs are enabled for software scanning. If you temporarily do not want the software scanner to scan an Intellisig, you can disable the scan of the Intellisig. You can enable it later when you want the scanner to discover it again.

Note: You can disable only at the Intellisig level, not at the product, release, or patch definition level.

Follow these steps:

1. Navigate to Software, Definitions, Intellisigs.

   All the Intellisigs are displayed in the right pane.

2. Right-click the Intellisig that you want to disable and select Disable Scan. Click Yes to confirm.

   When the collect task runs next time, the disabled Intellisigs are not sent to the agent. After the collect task runs, discovered records from previous scan if any, are hidden on the All Computers, computer name, Software, Discovered node. However, the detected product, release, or patch definitions are not removed from Software, Definitions, Intellisigs

Configure the Collect Task for Software Inventory

To collect software inventory using Intellsigs, you must configure the collect task to include Intellisig scanning.

Follow these steps:

1. Navigate to All Computers, *group or computer name*, Configuration.

   Note: Collect tasks configured at the group-level are applicable for all the computers in the group.

2. Right-click Collect Tasks and select New.
3. Select Software Discovery in the Select new Collect Task type dialog and click OK.
4. Specify a name for the collect task and click the Methods tab.
5. Select the Method as Signature Scanning and click Configure Scanner.
6. Select Intellisig scan and click OK.

   Note: You can select both Signature scan and Intellisig scan as part of the same collect task. Such a selection is applicable when you have some definitions as signatures and some as Intellisigs. When the same software is detected by both the scanners, the software is listed twice in the discovered software list. However, the list provides the discovery source that specifies whether the software was collected by the signature or Intellisig scan.

   The collect task is configured. When the collect task runs next time, the agent runs the Intellisig scripts, and collects Intellisigs inventory.

Important! CA Patch Manager needs signature scan to be enabled. If you are using CA Patch Manager for applying security patches on your target computers, verify that you have configured signature scanning also on target computers. Configuring only Intellisig prevents patch management from working properly.

View Discovered Software

You can view the discovered software to see the software discovered by Intellisigs on agent computers. The discovered Intellisigs are available from the following locations in DSM Explorer:

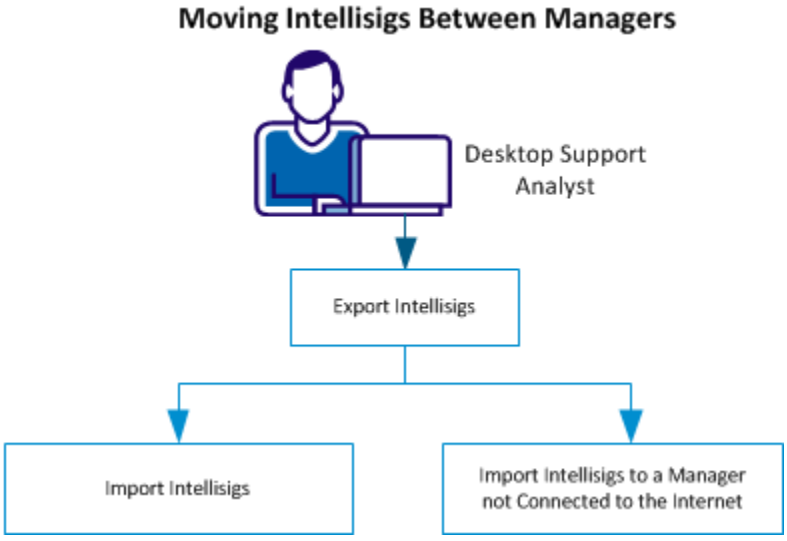
- Computers and Users, All Computers, *Computer Name*, Software, Discovered
- Software, Definitions, Intellisigs, *Intellisig Name, Detected Product, Detected Release,* Discovered Installations
- Software, Definitions, Intellisigs, *Intellisig Name, Detected Product, Detected Release, Detected Patch,* Discovered Installations
- Software, Definitions, Categories, *Detected Product, Detected Release,* Discovered Installations

- Software, Definitions, Categories, *Detected Product, Detected Release, Detected Patch,* Discovered Installations
- Software, Definitions, Manufacturers, *Detected Product, Detected Release,* Discovered Installations
- Software, Definitions, Manufacturers, *Detected Product, Detected Release, Detected Patch,* Discovered Installations

Note: For more information about software definitions, categories, and manufactures, see the DSM Explorer *online help.*

Moving Intellisigs Between Managers

As a desktop support analyst, you can move custom Intellisigs between managers so that you do not have to create them manually in each manager. You can export the Intellisigs from the source manager and import them into the target manager.



Export Intellisigs

Export the Intellisigs to either modify the Intellisig or move the Intellisig from a domain manager to enterprise manager or to another domain manager. You can export all the Intellisigs, specific Intellisigs, or specific versions of an Intellisig. You can also use the CLI for exporting Intellisigs. For more information about the CLI, see intellisigcmd export—Export Intellisigs (see page 107).

Note: You can export only at the Intellisig level, not at the product, release or patch definitions level.

Follow these steps:

1. Navigate to Software, Definitions and do one of the following depending on whether you want to export all the Intellisigs, specific Intellisigs, or specific versions of an Intellisig:

   All Intellisigs

   - Right-click the Intellisigs node and select Export.

   Specific Intellisigs

   - Click the Intellisigs node and select the Intellisigs that you want to export in the right pane. Right-click the selection and select Export.

   Specific Versions of an Intellisig

   - Click the Intellisigs node and right-click the Intellisig and select Properties. Click the Versions tab, select the versions that you want to export, and click Export on the Intellisig Properties dialog.

   The Intellisig Export dialog lists the Intellisigs you selected.

2. Specify whether you want to export the Intellisig as a compressed file or uncompressed folder and click Browse to modify the location if required.

   Note: We recommend using uncompressed format while modifying the Intellisigs and compressed format while moving the Intellisigs between managers.

3. Click OK.

   The selected Intellisigs are exported and placed in the location you specified.

More information:

Modifying a Custom Intellisig (see page 91)
Moving Intellisigs Between Managers (see page 61)

Import Intellisigs

Import the Intellisigs to either update the Intellisig after modifications or move the Intellisig from a domain manager to enterprise manager or to another domain manager. You can also use the CLI for exporting Intellisigs. For more information about the CLI, see intellisigcmd import—Import Intellisigs (see page 108).

Follow these steps:

1. Navigate to Software, Definitions and right-click the Intellisigs node and select Import.

   The Intellisigs Import dialog opens.

2. Click Browse and select the zip or xml file of the Intellisig that you want to import.

3. Select the Import Mode to specify whether you want to replace the existing definition, merge only new changes, or merge all the changes.

4. (Optional) Clear the Update active Intellisig version during the import option, if you do not want to modify active Intellisig versions. Click OK.

   The import process begins.

Note: The intellisig in zip format must be exported by corresponding zip (from the export zip options of Intellisig only). Do not use any third-party compression (zip) utility.

Import Intellisigs to a Manager not connected to the Internet

You can import Intellisigs to a manager even if it is not connected to the internet, provided that you have one manager that is connected to the Internet where you download the Intellisigs from the CA content server.

For more information, see content utility (see page 115) under the Intellisig feature.

Follow these steps:

1. Run the command-line tool contentutility.exe.

   Note: You can run the tool either by providing an .xml parameter file or by specifying the parameters directly on the command line. When you run the tool with no parameters, it creates an xml file in the CA\DSM\bin directory called content_utility.xml. You can specify a combination of CA-provided, custom-created, ca_intellisig, and custom_intellisig parameters to determine which software definitions are exported or imported.

2. Specify values for the following parameters:

   ca_intellisig

   Specifies whether to include CA-defined Intellisigs in the export or import.

   custom_intellisig

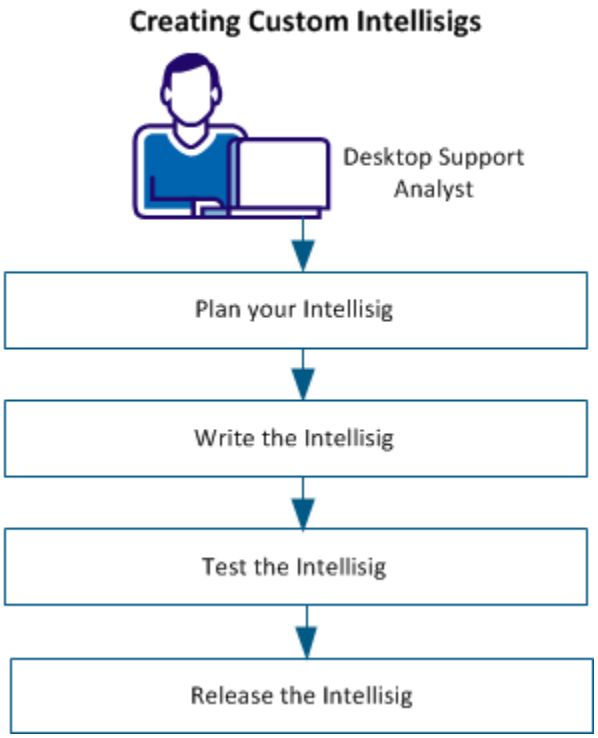
   Specifies whether to include custom-defined Intellisigs in the export or import.

   intellisig_detail

   Specifies whether to include scripts and triggers that are associated with an Intellisig during an export or import. If set to no, the master definition is exported or imported but the details are omitted.

Creating Custom Intellisigs

As a developer, you can create custom Intellisigs for any software that you use for which a CA-provided Intellisig is not available.



Plan your Intellisig

Intellisig Design Considerations

- Intellisigs provide a mechanism to detect current and future releases of products. As an author, focus on ways of inspecting the system in such a way that once written, the script is future proof.
- We recommend that you keep the number of Intellisig definitions low, each covering more than just a single product. A good practice is to create one Intellisig for each vendor and to create an Intellisig per vendor and per software category for large vendors. For example: IBM – Database Software, Microsoft - Database Software, Microsoft – Office Software, Apache – Office Software, and CA – Management Software. Verify that you keep the category names synchronized for all Intellisigs from a usability point of view.
- An Intellisig contains a main script and can contain a number of data files, which themselves can be scripts. You can keep the design clean and operating at the vendor or category level by using the data files as script-holders for product detection. The main script, which is vendor-specific, invokes all the other (data) scripts. These scripts, in turn, detect individual products. To extend the capability of an Intellisig, create a version of the vendor-specific Intellisig, adding new or updated existing scripts rather than adding a complete new Intellisig. Versions are embedded chains within a single Intellisig, hence, they do not clutter the user interface. Keep the list of the top-level Intellisigs as short as possible.
- Software that is detected heuristically or using traditional signatures uses a one-to-one relationship between software products and releases. With Intellisigs, you can control this behavior because you create both the detected products and releases. The best practice is to create a one-to-many relationship between detected products and releases for the following three reasons:

  1. A product can have multiple releases.
  2. The amount of data that is needed when using one-to-one relationships is high as compared to one-to-many.
  3. Most importantly, CA DSM feeds CA Software Compliance Manager (SCM) with detection information. SCM works around DSM products, not releases. Take this into consideration and create detected products that are based on how they are licensed. This way, you can help ensure that an SCM-licensable product can be matched to a single DSM-detected product. In addition, you can achieve a granular detection at the release (and patch) level and a meaningful detection at the product level.

- Intellisig-created products, releases, and patches (detected software definitions) are identified by name and version label as well as their location within the detection hierarchy of an Intellisig. An empty version label is allowed and treated as a separate version (the empty version). Intellisig-detected software definitions are distinct from CA-provided, custom, and heuristic software definitions even if they have the same name and version. In addition, different Intellisigs can create different definitions with the same name and version.

- Use the DMscript functions and provide software names and versions for each.
- A manufacturer can be provided when creating a detected definition. This information can be provided either by UUID or by name. The manufacturer UUID must be that of an existing manufacturer. If it does not exist, the detected definition has an empty manufacturer. The manufacturer name can be either that of an existing manufacturer or that of a new one. For the latter, the manufacturer is created and the detected definition is assigned to it. An existing manufacturer can be either CA-provided, custom-created, or created by heuristic software discovery.
- A detected definition can also be assigned to a category. This information can be provided either by UUID or by name. If provided by UUID, the category must exist. If provided by name, the category is created if it does not exist. An existing category can be either CA-provided or custom-created.

A best practice for Intellisig behavior is given below, using three different products: Microsoft Windows, Microsoft Office, and Microsoft SQL Server.

Name a Product

Verify that the product reflects the licensable entity. The best practice for naming is as follows (optional in square brackets):

- Name: "<Manufacturer> <Product> [<Edition>] [<Architecture>] [<Language>]" – When applicable, add the edition. If the licensing depends on architecture or language, include it in the name. Verify to include this information only if necessary, to minimize the number of products created.
- Version Label: "<major>[.<minor>]" – Distinguishes a version from the other versions

Example:

- Name: "Microsoft Windows 7 Ultimate"

  Version Label: "6.1"
  Use both the major and minor versions to distinguish the version from Windows Vista (6.0) and Windows 8 (6.2). Remove 7 from the name if necessary.
- Name: "Microsoft Office 2010 Professional Plus"

  Version Label: "14"
  Use only the major version because it distinguishes two releases of Office from each other. Remove 2010 from the name if necessary.
- Name: "Microsoft SQL Server 2008 R2 Enterprise"

  Version Label: "10.5"
  Use both the major and minor version to distinguish the version from other versions. Remove 2008 R2 from the name if necessary.

Provide the following optional properties if available, to be inserted into dedicated columns in the database: VersionNumber, Language, Bitness, Architecture, Manufacturer, Category and Description.

Name a Release

Verify that the release captures as much information about the detected software as possible and that it is linked to a product. The best practice for naming is as follows (optional in square brackets):

- Name: "<Manufacturer> <Product>[<Edition>] [<Architecture>] [<Language>]"
- Version Label: "<major>.<minor>.<minor'>.<minor''> [<release/service pack>]" – Include as many details as possible. Each Architecture and Language has a separate release record but all are linked to the same product.

Example:

- Name: "Microsoft Windows 7 Ultimate x64 en-us"

  Version Label: "6.1.7601 Service Pack 1 Build 7601"
- Name: 'Microsoft Office 2010 Professional Plus x64 en-us"

  Version Label: "14.0.6112.5000 Service Pack 1"
- Name: "Microsoft SQL Server 2008 R2 Enterprise x64 en-us"

  Version Label: "10.50.1617.0 Service Pack 1"

Provide the following optional properties if available, to be inserted into dedicated columns in the database: VersionNumber, Language, Bitness, Architecture, Manufacturer, Category and Description.

Name an Instance of a Release

The release instance is the actual detection record that links a release to a computer and forms part of the software inventory. A computer can have multiple instances of the same release installed. An installation can have additional properties which are specific to the instance.

Example:

- Microsoft Windows 7 Ultimate instance:

  InstallPath: C:\Windows

  Origin=Forward Inc

  TrustLevel=5

- Microsoft Office 2010 Professional Plus instance:

  InstallPath: C:\Program Files\Office

  ProductGUID: FBD367D1-642F-47CF-B79B-9BE48FB34007

  CustomData: Product-ID=02257-210-8656854-49625

  Origin=Forward Inc

  TrustLevel=5

- Microsoft SQL Server 2008 R2 Enterprise instance 1:

  InstallPath: C:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER

  Label: MSSQLSERVER

  CustomData: CPU=2/8/16;RAM=32GB;

  Origin=Forward Inc

  TrustLevel=5

- Microsoft SQL Server 2008 R2 Enterprise instance 2:

  InstallPath: C:\Program Files\Microsoft SQL Server\MSSQL10_50.SQLEXPRESS

  Label: EXPRESS

  CustomData: CPU=2/8/16;RAM=32GB;

  Origin=Forward Inc

  TrustLevel=5

Provide the following optional properties if available, to be inserted into dedicated columns in the database: Label, InstallPath, SerialNumber, ProductGUID, LastAccessed, Origin, TrustLevel and CustomData.

The CustomData property is used to collect instance-specific information that affects licensing. Its size is limited to 255 characters. This property can hold information about the number of processors, cores or threads, and memory. CA ITCM does not use the collected custom data but it can be used in field-developed solutions.

Name a Patch

Verify that the patch captures as much information about the detected software as possible and that it is linked to a release. The patches that are detected by using Intellisigs are not used by the DSM Patch Manager.

Example:

Name: "KB971033 x64 en-us"
Version label: ""

Provide the following optional properties if available, to be inserted into dedicated columns in the database:

- VersionNumber Language
- Bitness
- Architecture
- Manufacturer
- Category and
- Description

Name an Instance of a Patch

The patch instance is the actual detection record that links a patch to a computer and forms part of the software inventory. A computer can have multiple instances of the same release installed. Each of those release instances can either have or not have the patch installed. Therefore, create one patch instance for each release instance. A patch can have additional properties that are specific to the instance.

Provide the following optional properties if available, to be inserted into dedicated columns in the database: Label, InstallPath, SerialNumber, ProductGUID, LastAccessed, Origin, TrustLevel and CustomData.

Example:

KB971033 x64 en-us instance:

Origin=Forward Inc
TrustLevel=5

Sample Intellisigs and the Folder Structure

Two sample Intellisigs are provided in *DSM path*\Intellisigs folder.

- For Windows:00000000-A0A0-AAAA-0000-AAAA0000AAAA
- For Unix: 00000000-B0B0-BBBB-0000-BBBB0000BBBB

Sample.xml contains the metadata to import these samples into your MDB. Run the following command in the Intellisigs directory:

```
intellisigcmd import file=sample.xml
```

Each sample Intellisig has the following folder structure:

*uuid of intellisig**version id**scriptfile*.dms

Note: Place any additional Intellisig data files in the same location.

More information:

Export Intellisigs (see page 62)
Import Intellisigs (see page 63)

Analyze the Software Identification Information

As with traditional signatures, determine the product structure because this has to reflect in the script and its output. Determine the following:

- Product naming, for example, Microsoft Office
- Determine how releases of the product are structured and named.
  - Verify if there are 32-bit and 64-bit releases.
  - Verify if there are multiple languages.
  - Verify if there are different releases for different platforms, for example, Office for Windows and Office for Mac.
- Verify if the patches are individual binaries or installable packages registered with the OS.
- Verify if the product provides an ISO19770-2 SWID file that can be read to provide the information required.

After determining the product structure and the naming convention, decide on the granularity of the Intellisig. This means deciding on how many separate Intellisigs you need to cover the product. The product may be large, for example, Microsoft Office. If the script required to detect each product is large, plan on multiple scripts per sub-product. These scripts are easier to maintain than a large script that performs all the tasks. If the product is small, for example 7-zip, a single script suffices.

Finally, determine the method of detection. An Intellisig must examine resources on the system to detect what is installed. This means looking for specific files, registry keys, configuration files, and settings. An Intellisig can be as intelligent as required. At one level, it can emulate a traditional signature and look for specific information. At another level, it can analyze the computer and detect all possible products from all manufacturers, even if those products are unknown to the Intellisig (similar to a heuristic scan). An ideal Intellisig can operate between these two levels. At a minimum, verify that any version and configuration of a product, past and future, can be detected (as opposed to specific existing versions).

Write your Intellisig

A custom Intellisig requires a metadata file and a script. The files need to be in a specific folder structure. For a sample Intellisig script and folder structure, see Sample Intellisigs and the Folder Structure (see page 71).

Write the Metadata File

The metadata file is a .xml file that contains information about the Intellisig. The domain manager and software scanning agent use this file to manage it as a unit. This file also contains definitions of the triggers. The agent evaluates these rules to determine whether to run the script to minimize the runtime of the agent.

Example:

```
<eso_fingerprints version="1.0.0">
    <preferences>
        <ignore dllcache="true" />
        <ignore spuninstall="true" />
        <ignore internetcache="true" />
        <ignore cookies="true" />
        <ignore nortontrash="true" />
        <ignore sysbackup="true" />
        <ignore SDLib="true" />
    </preferences>
    <technologies dateupdateint="1320340565" dateupdate="05/12/11 11:28:05" >
        <technology id="11111111-B1B1-BBBB-1111-BBBB11110000"
                        name="Trigger Test Unix Single File And" version="1.0.0"
                    descr="Intellisig only run if file trigger detected
(trigger_file.txt)"
                    swtype="17" srctype="6" iscript="intellisig-trigger-test.dms"
os="Unix">
              <additional_files>
                  <data name="Trigger_Test_Unix.txt"/>
                  <data name="Trigger_Test2_Unix.txt"/>
              </additional_files>
              <group type="and">
                  <file name="trigger_file.txt" path="*" />
              </group>

    </technology>
    </technologies>
</eso_fingerprints>
```

ID

Specifies the UUID of the intellisig. This ID can be generated using the Intellisig command line tool, intellisigcmd.exe genuuid.

iscript

Specifies the name of the Intellisig script file.

Additional_files

Specifies a list of optional files that the script requires.

Group

Defines an optional trigger.

Write the Script

A basic understanding of writing programs using dmscript is a prerequisite for writing the Intellisig script.

Note: For more information about the dmscript language, see *ITCM_DMSScriptingLanguage_Ref_ENU.pdf*.

An Intellisig script performs the following tasks:

- Initializes, collects, and checks arguments.
- Creates the output file.
- Searches for the intended products, releases, patches, and installed instances and collects the required items of information about each.
- Invokes the appropriate dmscript functions to copy this information to the output file in the correct order.
- Reports any errors
- Cleans up and terminates

On Windows, you can use the dmsedit utility to develop the script. This utility is an integrated editor and debugger for developing and running scripts. This method provides all the regular debug features of breakpoints, stepping through code, and examining variables.

Important! Do not use the print statement on Windows because the script is not interactive. On Windows, the result appears in an unscrollable dialog and appears only for 10 seconds. On UNIX, the output is stdout.

Use a text editor and a command line window to develop the script. You can use the dmstrace function to output diagnostic information to the trace log.

Write Scripts that Detect Multiple Products

The dmscript language supports the #include directive which lets you include scripts to execute functions in them. In line with the best practices for creating Intellisigs, we recommend the following approach to write scripts:

1. Write a script for each product you intend to detect and add each as an ancillary file to the Intellisig definition. These scripts perform the detection and registration of the detected product, release, patch as well as their discovered instances. Verify that these scripts do not perform any Intellisig initialization or finalization code.
2. Write a main script that performs the Intellisig initiation and finalization and add it to the Intellisig definition. Use the #include directive in the main script to include the ancillary files and run the exposed detection function in each as part of the main script execution.

Example:

The ancillary file windows.dms defines the function DetectWindows(…).

The ancillary file office.dms defines the function DetectOffice(…).

The ancillary file sql.dms defines the function DetectSql(…).

The main script main.dms appears as follows:

```
#include windows.dms
#include office.dms
#include sql.dms
```

DoInitialization(…)

Specifies the REM function defined in main.dms that parses args and calls OpenDetectedSoftwareOutputFiles and performs other initialization tasks.

DetectWindows(…)

Specifies the REM function defined in windows.dms.

DetectOffice(…)

Specifies the REM function defined in office.dms.

DetectSql(…)

Specifies the REM function defined in sql.dms.

DoFinalization(…)

Specifies the REM function defined in main.dms that calls CloseDetectedSoftwareOutputFiles and performs other cleanup tasks.

Write Help for the Custom Intellisig

You can write help for the Intellisigs that you create and provide additional details in a separate web page. The details help the Intellisig users to know more about the Intellisig. The help is displayed on the Intellisig properties page in DSM Explorer.

Follow these steps:

1. View the help file for one of the CA-provided Intellisigs.
   a. Open DSM Explorer and navigate to Software, Definitions.
   b. Right-click a CA-provided Intellisig and click Properties.
   c. Click the Details tab on the Intellisig Properties dialog.
   d. Click the ? button.

      A webpage displays additional information about the Intellisig.
2. Create a .html help file for your custom Intellisig based on a CA-provided Intellisig.
3. Save the help file with the Intellisig name as the file name.
4. Host the help files on a web server.

   After you host the help files, configure the URL in the default configuration policy.
5. Navigate to Control Panel, Configuration, Configuration Policy. Unseal Default Configuration Policy.
6. Navigate to DSM, Administration Console.
7. Double-click the customIntellisigsURL parameter and specify the value as c:\webinfo\%name%.html in the Setting Properties dialog.

   The help file is associated with the corresponding Intellisig based on the HTML file name and the URL you configured. When you test the Intellisig, you can test the help also.

   Note: The caIntellisigsURL parameter specifies the help file details for CA-provided Intellisigs.
8. Repeat step 1 for your custom Intellisig to view the corresponding help file.

Initialization

The scanner launches dmscript in the same directory as the script file with the following arguments:

```
dmscript <scriptfile> -I <uuid of Intellisig> -v <version of Intellisig> -n <name of Intellisig>  -t <trigger_info>
```

trigger_info

Specifies an optional string that contains information about the trigger that runs the Intellisig. A script can use this string as hints in its discovery process, if necessary.

A script must collect and check that these arguments are present and if missing, report an error and exit. The values of arguments are available from the argv function.

Example:

```
dim sUuid as string
dim sISVersion as String
dim sISName as String
dim X as Integer
for X=0 to argc()
    if ( argv(X)="-i") then
        sUuid = argv(X+1)
    endif
    if ( argv(X)="-v") then
        sISVersion=argv(X+1)
    endif
    if ( argv(X)="-n") then
        sISName = argv(X+1)
    endif
next X
if sUuid = "" then
    LogDetectedSoftwareError ("00404","Param1=missing argument uuid to script ")
    exit
endif
```

Trace and Report Errors

Dmscript provides a means of outputting tracing information to the log file, TRC_DMSCRIPTINTERPRETER_0.log. The log file is useful for debugging during development and troubleshooting after the release of the Intellisig.

Use the LogDetectedSoftwareError function to report errors back to the manager. This function writes errors in a standard format to a file called <uuid>.err. This file is automatically uploaded to the domain manager. Errors are also copied to the trace log file. DSM Explorer displays the messages in the job status dialog after converting the messages to the local language.

Note: Though this function lets you specify English text directly, use localized error messages. If required, edit the statmod.<lang> file on the manager or explorer to add your own messages.

Create the Output File

Open the output file using the OpenDetectedSoftwareOutputFiles function.

Example:

```
if OpenDetectedSoftwareOutputFiles (sUuid,sISVersion,sISName) <> CASWDETECT_OK then
    exit
endif
```

This function creates a file called <uuid>.xml, which contains the results.

Search for Items

Searching for items depends on the details of the product. You can use the following functions:

Detect Files and Directories

To detect if a file exists, use the ExistFile function. To detect a directory, use the ExistDirectory function.

Example:

```
ProgramFiles = EnvGetString(“ProgramFiles”)
if ExistDirectory (ProgramFiles + “\Microsoft Office”) then
…found MS Office…
if ExistFile (ProgramFiles + “\Microsoft Office\Office14\EXCEL.EXE") then
… found excel…
```

Read the Registry

Use the following functions:

RegOpenKey

Reads the registry

RegQueryValue

Reads a value

RegEnumKeys

Enumerates keys

RegEnumVariable

Enumerates variables

RegCloseKey

Closes a key

Note: dmscript is a 32-bit program, so Windows applies registry redirection to certain keys such as HKLM\SOFTWARE. To read the parts of the registry that is reserved for 64-bit programs, call setmode64(1). This action turns redirection off. This function also applies to certain parts of the file system such as \windows\system32.

Read the Environment

To read the content of an environment variable on Windows or UNIX, use the EnvGetString function.

Example:

```
path = EnvGetString("PATH")
…now search the path for a target directory..
```

You can also enumerate all environment variables using the EnvGetFirst and EnvGetNext functions.

Read Configuration Files

Use the following functions:

- To read an .ini style file, use the ReadIniEntry function.
- To read a plain text file, open it using the OpenFile function then read it using ReadFile.You can use the new GetToken function to parse lines of text.
- To read an XML or binary file, write a custom DLL or external program and call it from the script. You can read the output of the program using the OpenFile function.

Note: The custom binary cannot be included in the Intellsig definition.

Run Programs

It is possible that dmscript cannot accomplish some tasks or there can be tasks in which it is simpler to run an existing shell script or external program. In such cases, use the Execute function.

Example:

To get the system version on Windows, use the following command:

```
Execute ("cmd.exe /c ver > dms.tmp", True, 0)
```

The output is available in dms.tmp.

Read the list of Installed Programs

On Windows, run one of the following commands, from Microsoft, SysInternals:

```
wmic product
psinfo -accepteula -s
```

Redirect the output to a file and then read it.

Detect Running Programs

Dmscript cannot detect running programs directly but can spawn system commands which can detect these programs and then read their output.

Example:

```
(Unix)"
execute (“ps –ef | grep myprogram > /tmp/dms.tmp”)
(Windows)
execute ("cmd.exe /c tasklist /nh /fi ""imagename eq powerpnt.exe"" > dms.tmp",true,0)
…now read the contents of dms.tmp…
```

Create the Output

Products, releases, patches and instances exist in a hierarchy that you must follow in a script:

- A product has a number of releases.
- A release has a number of patches.
- A release has a number of instances (or installed copies of the software).
- A patch has a number of instances (or installed copies of the patch).

This means that you must specify a product first, followed by its releases, patches and instances. When creating the subsidiary items, the parent name and version must be specified too. Dmscript remembers the hierarchy being created and checks that the parents of items exist and are valid. If not, an error is returned. There are a number of specific functions for creating products. These functions have the following general format:

```
CreateDetectedSoftwareXXXX (name, version, [parentname, parentversion,…]
OptionalProperties)
```

Create a Product

To create a product, use the following code:

```
CreateDetectedSoftwareProduct (ProductName as String,

                               ProductVersionLabel as String,

                               OptionalProperties as String) as integer
```

Example:

```
CreateDetectedSoftwareProduct("Microsoft Windows 7 Ultimate", “6.1”,
"VersionNumber=6.1 | Language=en-us | Bitness=64 | Architecture=x64 |
Manufacturer=Microsoft Corporation | Category=Operating Systems | Description=The
Microsoft Windows 7 Product | ")
```

In this example, the values are defined but in a real script they are determined as part of the detection process.

Create a Release

To create a release of a product, use the following code:

```
CreateDetectedSoftwareRelease (ProductName as String,
                               ProductVersionLabel as String,
                               ReleaseName as String,
                               ReleaseVersionLabel as String,
                               OptionalProperties as String) as integer
```

Example:

```
CreateDetectedSoftwareRelease("Microsoft Windows 7 Ultimate", "6.1", "Microsoft
Windows 7 Ultimate x64 64 en-us", "6.1.7600", "VersionNumber=6.1.7600 |Language=en-us
| Bitness=64 |Architecture=x64 | Manufacturer=Microsoft Corporation |
Category=Operating Systems | Description=The Microsoft Windows 7 Release | ")
```

Create a Patch

To create a patch, use the following code:

```
CreateDetectedSoftwarePatch (ProductName as String,
                             ProductVersionLabel as String,
                             ReleaseName as String,
                             ReleaseVersionLabel as String,
                             PatchName as String,
                             PatchVersionLabel as String,
                             OptionalProperties as String) as integer
```

Example:

```
CreateDetectedSoftwarePatch("Microsoft Windows 7 Ultimate", "6.1", "Microsoft
Windows 7 Ultimate x64 64 en-us", "6.1.7600", "KB971033 x64 64 en-us", "Language=en-us
| Bitness=64 | Architecture=x64 | Manufacturer=Microsoft Corporation |
Category=Operating Systems | Description=The Microsoft Windows 7 Activation Checker
Update | ")
```

Create an Instance

To create an installed instance of a release, use the following code:

```
CreateDetectedSoftwareReleaseInstance (ProductName as String,
                                       ProductVersionLabel as String,
                                       ReleaseName as String,
                                       ReleaseVersionLabel as String,
                                       OptionalProperties as String) as integer
```

To create an installed instance of a patch:

```
CreateDetectedSoftwarePatchInstance (ProductName as String,
                                     ProductVersionLabel as String,
                                     ReleaseName as String,
                                     ReleaseVersionLabel as String,
                                     PatchName as String,
                                     PatchVersionLabel as String,
                                     OptionalProperties as String) as integer
```

Example:

```
CreateDetectedSoftwareReleaseInstance("Microsoft Windows 7 Ultimate", "6.1",
"Microsoft Windows 7 Ultimate x64 64 en-us", "6.1.7600", "Origin=Forward Inc |
TrustLevel=5 | InstallPath=C:\Windows | SerialNumber=1234-567-890414-86668 |
LastAccessed=2011-11-29T12:30 | ")

CreateDetectedSoftwarePatchInstance("Microsoft Windows 7 Ultimate", "6.1",
"Microsoft Windows 7 Ultimate x64 64 en-us", "6.1.7600", "KB971033 x64 64 en-us", "",
"Origin=Forward Inc | TrustLevel=5 | ")
```

Note: Replace the value in the Origin parameter to reflect your organization name.

Close the Output File

After releasing the resources your script uses, call CloseDetectedSoftwareOutputFiles.

Test the Intellisig

Test your script offline on a small number of computers before you release it. Testing lets you correct the script before you apply it on all the computers in your organization. You can run a script from the command line and then examine the following files for the results:

- Compilation errors, if any, are generated in the trace log file. You can also view these errors in dmsedit and stdout on Unix.
- Any errors that are raised by the script are stored in <uuid>.err and also the trace log file.
- The discovered software results are stored in <uuid>.xml.

When testing, you do not need to specify a UUID on the command line. Any string works and is used to name the output files. The UUID is required only by the agent, which searches for files that are named using this string. It is recommended that you use a UUID.

Verify that you have available the computers, operating systems, and software installed that you want to detect. If not, simulate the installed software by creating the appropriate file, registry, and configuration entries that would normally be present. You can obtain trial versions from the vendors.

Verify that you test the script on all the target operating systems.

For the second level of testing, create a small DSM environment using spare computers or virtual machines. These computers must have an MDB, a domain manager, and a scalability server on one computer and the Asset Management agent installed on one of the target operating computers. Import the Intellisig using DSM explorer and run an Intellisig-based software detection task. Check that the expected results appear.

Release the Intellisig

An Intellisig that CA ITCM produces is delivered to customers using the content download job. If the Intellisig is custom created, use DSM Explorer to export it from the test manager and then import it into the production enterprise or domain manager.

More information:

Export Intellisigs (see page 62)
Import Intellisigs (see page 63)

Intellisig Triggers

You can restrict the running of an Intellisig script to occur only when specific conditions or triggers are met. This restriction enables efficient Intellisigs to be designed. Each Intellisig need not search a file system for specific files. You can instruct the scanner to scan for specific files, registry entries, services or installed packages, and the Intellisig script runs if these conditions are met. The scanner only performs the file system search once, irrespective of the number of file criteria specified.

Note: These triggers are defined in the same way as the traditional signature definitions. Experience in creating traditional signatures using these criteria is a prerequisite.

Trigger Criteria Types

The types of triggers that can be associated with an Intellisig are as follows:

File

Supports checking for file presence, specifying criteria for search paths, filename content patterns, file creation and modification date ranges, file size ranges, MD5 hash values, and permissions.

Registry

Supports checking for registry entries on specific keys, pattern matching of values, and 32-bit and 64-bit hive checks.

Package

Checks for an installed package, specifying version, and release. On Windows, this option checks the Add/Remove programs database. On UNIX, it checks the installation database of the platform.

Service

Searches for an installed service or daemon.

Sysinfo

Supports checking attributes such as the platform, processor, OS release, OS name, and OS version.

Logical Hierarchy of Triggers

Multiple trigger criteria can be specified on an Intellisig, and the triggers can be grouped with logical AND, OR, and NOT operators. The logical operators are defined as groups with a type. The type being either *and*, *or*, or *not*, which defines how the results from items that are contained within the group are combined. A group (except *not* groups) can contain any number of additional trigger criteria, including more groups. A group of type *not* can only contain a single item, although this single item could be another group.

Excluded Directories

You can specify to exclude directories when defining an Intellisig trigger. The directories listed in the exclude directories are excluded when the signature scanner searches the file system for the files specified in file trigger criteria.

XML Format

When writing a trigger, specify criteria using XML.

Include a top-level group in each trigger. Each group contains a type. The type is either *and*, *or*, or *not*. The type defines the logical operation that is applied to the criteria contained in the group.

```
<group type=”and”>
…<other citeria>…
</group>
```

File Criteria

File criteria are defined with the <file> tag. The file tag supports the following attributes:

name

Specifies the file to search for

path

Specifies the path the search

match

Scans the specified file for a match of the specified pattern

md5

Compares the MD5 hash of the file with the specified value

minversion

Compares the version of the specified file with the value provided

maxversion

Compares the version of the specified file with the value provided

mincreation

Compares the file creation date of the specified file with the value provided

maxcreation

Compares the file creation date of the specified file with the value provided

minmodified

Compares the file modification date of the specified file with the value provided

maxmodified

Compares the file modification date of the specified file with the value provided

minfilesize

Compares the file size of the specified file with the size value specified

maxfilesize

Compares the file size of the specified file with the size value specified

permsmustexclude

Excludes permissions

permsmustinclude

Includes permissions

rootowner

Specifies the root owner of the file

daclallow

Allows ACL

dacldeny

Denies ACL

arch

Specifies the architecture of a binary file on the Windows platform

Example:

```
<group type="and">
<file name="msword.exe" path="*" />
</group>
```

Registry Criteria

Registry criteria are specified with the <registry> tag. The registry tag supports the following attributes:

name

Specifies the registry Key to search

match

Specifies the pattern to match against

daclallow

Allows ACL

dacldeny

Denies ACL

arch

Specifies the behavior on a 64-bit machine. Supported values are 32, or 64, and any. On a 64-bit Windows computer, specifying 64 only searches the 64-bit hive for the specified key. Specifying 32 only searches the 32-bit hive on a 64-bit computer. This value is ignored on a 32-bit computer. If the architecture is not specified, the 32 behavior is the default behavior.

Example:

```
<group type=”and”>
<registry name="HKEY_LOCAL_MACHINE\SOFTWARE\Adobe\Adobe Acrobat\6.0\Language\UI"
match="ENU" />
</group>
```

Package Criteria

Package criteria are specified with the <package> tag. The package tag supports the following attributes:

name

Specifies the name of the installed product to search for

version

Specifies the version of the installed product to search for

Example:

```
<group type="and">
<package name="DameWare Mini Remote Control" version="6.0.*" />
```

Service Criteria

Service criteria are specified with the <service> tag. The service tag supports the following attributes:

name

Specifies the name of the installed service to search for

path

Specifies the (UNIX only) path to the installed service

Example:

```
<group type="and">
<service name="SNMP"/>
</group>
```

SysInfo Criteria

Sysinfo criteria is specified with the <sysinfo> tag. The sysinfo tag supports the following attributes:

platform

Specifies whether the platform is x86 or x64 on Windows

processor

Identifies the processor

osrelease

Identifies the OS Release

osname

Identifies the OS Name

osversion

Identifies the OS Version

Example:

```
<group type=”and”>
<sysinfo osname="Windows" />
</group>
```

Trigger Parameter when an Intellisig is Launched

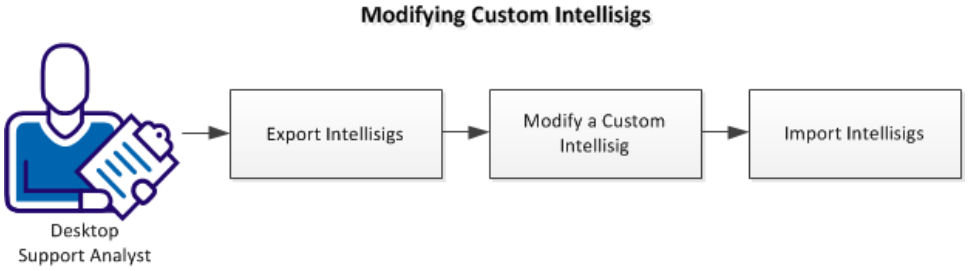
When a trigger is satisfied, and the Intellisig script is executed, a string is passed to the script defining the criteria that contributed to the triggering of the Intellisig. The trigger is passed to the Intellisig script using the *–t* flag. The format of the trigger string that is passed to the script depends on the criteria types used in the trigger definition. The following table provides examples of trigger strings with different trigger criteria types used. Where multiple triggers are contributing to the running of an Intellisig script, the individual criteria are specified with a separating | character. An Intellisig script must be able to parse the trigger string to leverage the processing that has been done in the trigger evaluation.

File	Trigger Example	-t parameter examples
file	<group type=”and”> <file name=”file1.test” path=”*” /> </group> <group type=”not”> <file name=”file1.test” path=”*” /> </group>	Positive trigger: -t “file:c:\file1.test” NOT trigger: -t “!file:file1.test” For the NOT trigger a full path cannot be provided.
registry	<group type="and”> <registry name="HKEY_LOCAL_MACHINE\SOFTWARE\iSigTest\T1" match="V1"/> </group> <group type="not”> <registry name="HKEY_LOCAL_MACHINE\SOFTWARE\iSigTest\T1" match="V1"/> </group>	Positive trigger: -t "registry:HKEY_LOCAL_MACHINE\SOFTWARE\iSigTest\T1" NOT trigger: -t "!registry:HKEY_LOCAL_MACHINE\SOFTWARE\iSigTest\T1"
service	<group type="and"> <service name="DNS Client"/> </group> <group type="not"> <service name="DNS Client"/> </group>	Positive trigger: -t "service:DNS Client" NOT trigger -t "!service:DNS"

File	Trigger Example	-t parameter examples
sysinfo	<group type="and"> <sysinfo osname="windows" platform="x86"/> </group> <group type="not"> <sysinfo osname="windows" platform="x86"/> </group>	Positive trigger: -t "sysinfo:osname=Windows;platform=x86" NOT trigger: -t "!sysinfo:osname=Linux;platform=x86"
package	<group type="and"> <package name="CA DSM Explorer"/> </group> <group type="not"> <package name="CA DSM Explorer"/> </group>	Positive trigger: -t "package:CA DSM Explorer" NOT trigger: -t "!package:CA DSM Explorer EXTRA123"

Modifying a Custom Intellisig

As a developer of the Intellisigs, you can modify the custom Intellisigs to update the script, triggers, or data files in the Intellisig.



Export Intellisigs

Export the Intellisigs to either modify the Intellisig or move the Intellisig from a domain manager to enterprise manager or to another domain manager. You can export all the Intellisigs, specific Intellisigs, or specific versions of an Intellisig. You can also use the CLI for exporting Intellisigs. For more information about the CLI, see intellisigcmd export—Export Intellisigs (see page 107).

Note: You can export only at the Intellisig level, not at the product, release or patch definitions level.

Follow these steps:

1. Navigate to Software, Definitions and do one of the following depending on whether you want to export all the Intellisigs, specific Intellisigs, or specific versions of an Intellisig:

   All Intellisigs

   - Right-click the Intellisigs node and select Export.

   Specific Intellisigs

   - Click the Intellisigs node and select the Intellisigs that you want to export in the right pane. Right-click the selection and select Export.

   Specific Versions of an Intellisig

   - Click the Intellisigs node and right-click the Intellisig and select Properties. Click the Versions tab, select the versions that you want to export, and click Export on the Intellisig Properties dialog.

   The Intellisig Export dialog lists the Intellisigs you selected.

2. Specify whether you want to export the Intellisig as a compressed file or uncompressed folder and click Browse to modify the location if required.

   Note: We recommend using uncompressed format while modifying the Intellisigs and compressed format while moving the Intellisigs between managers.

3. Click OK.

   The selected Intellisigs are exported and placed in the location you specified.

More information:

Modifying a Custom Intellisig (see page 91)
Moving Intellisigs Between Managers (see page 61)

Modify a Custom Intellisig

You can modify an Intellisig to update the script, triggers, or data files.

Follow these steps:

1. Verify that you have exported the Intellisig.
2. Open the exported Intellisig archive file or folder and update the files as per your requirement. For more information about customizing Intellisigs, see Creating Custom Intellisigs (see page 65).
3. Save the modified files and compress them if you want to import them as zip files.

Import Intellisigs

Import the Intellisigs to either update the Intellisig after modifications or move the Intellisig from a domain manager to enterprise manager or to another domain manager. You can also use the CLI for exporting Intellisigs. For more information about the CLI, see intellisigcmd import—Import Intellisigs (see page 108).

Follow these steps:

1. Navigate to Software, Definitions and right-click the Intellisigs node and select Import.

   The Intellisigs Import dialog opens.
2. Click Browse and select the zip or xml file of the Intellisig that you want to import.
3. Select the Import Mode to specify whether you want to replace the existing definition, merge only new changes, or merge all the changes.
4. (Optional) Clear the Update active Intellisig version during the import option, if you do not want to modify active Intellisig versions. Click OK.

   The import process begins.

Note: The intellisig in zip format must be exported by corresponding zip (from the export zip options of Intellisig only). Do not use any third-party compression (zip) utility.

Reconciling Intellisig Data with other CA Products

Detected Intellisig software definitions and their discovered software instances are synchronized from a CA ITCM source MDB to a target MDB, to allow integration with other CA products.

Synchronize Intellisig Definitions with a Remote Database

Detected Intellisig software definitions and their discovered software instances are added to the synchronization configuration when the synchronization task is created, if the target MDB supports Intellisigs. If the synchronization task already exists, the configuration are updated when the task next runs.

This is applicable for both CA-provided and custom Intellisigs.

The master Intellisig definition of the detected software definitions are created during the first phase of synchronization when you run the content utility.

Detected software definitions are matched with any existing definitions at the target (originating from other MDB sources).

Links of detected software definitions to manufacturers and categories are replicated to the target MDB with the manufacturers and categories created if required. Any previous category membership created by Intellisig are deleted. But any category membership created manually are preserved.

On deleting the synchronization job and selecting cleanup, the discovered software that originated from the source MDB are deleted. If this is the last source MDB, the detected software definitions and master Intellisigs without details are also deleted.

Replicating Intellisigs

You can create your custom Intellisig definitions on the enterprise manager and can replicate these definitions. Replication helps ensure that you do not need to define the definitions separately on each domain manager.

(Optional) Modify Replication Settings for Intellisig Definitions

You can modify the default replication settings that are defined in the replication task.

Follow these steps:

1. Open DSM Explorer. Navigate to Control Panel, Configuration, Configuration Policy, *Policy name* and right-click the policy and select Unseal.
2. Navigate to Manager, Engines under the configuration policy.

3. Specify values for the following configuration policy parameters:

   Ignore replicated Intellisig if they are locally defined

   Controls the behavior of the replication if the Intellisig definition was previously created locally at a domain manager. If you set this parameter to true, the master definition and details from the enterprise manager are not replicated. Otherwise, the master definition at the domain manager is updated by the definition from the enterprise manager. Its domain_uuid is changed to that of the source. The existing definition at the domain manager is updated. Any existing details of the master at the domain manager are deleted. The Intellisig details from the enterprise manager are replicated to the domain manager.

   Default: False

   Convert replicated Intellisigs on unlinking

   Overrides the default behavior that deletes all replicated master Intellisig definitions and their details at the domain manager on unlinking. If you set this parameter to true, on unlinking, any replicated master Intellisig definition and details are converted at the domain manager as belonging to the domain manager. Namely, their domain_uuid is changed from that of the enterprise manager to that of the domain manager.

   Default: False

   Leave replicated software definitions

   Specifies whether the replicated heuristic and Intellisig detected software definitions must be retained at the enterprise manager when the domain managers have removed them. If set to False, they are deleted at the enterprise manager when the last domain manager reports them as deleted.

   Default: False

   Leave unreferenced software definitions on collection

   Specifies if heuristic software definitions must be left undeleted at the domain manager, on an engine collect, when their last discovered software instance is deleted.

   Default: False

4. Save and the seal the policy. Apply the policy on computers that host the DSM engine, which runs the replication tasks.

More information:

Considerations for Using Intellisigs in an Enterprise Manager Setup (see page 96)

Considerations for Using Intellisigs in an Enterprise Manager Setup

An enterprise manager replicates detected software definitions and discovered software records when a domain manager is linked to the enterprise manager. After the replication process, you can view, query, report, and synchronize the information from a single location.

The following considerations apply when you use Intellisigs in an enterprise manager setup:

- Custom-created Intellisig definitions and their details are replicated from the enterprise manager to a domain manager.
- CA-defined Intellisig definitions and their details are not replicated from the enterprise manager to a domain manager.
- Products, releases, and patches that are detected by Intellisigs are replicated from a domain manager to the enterprise manager.
- If an Intellisig definition does not exist at the enterprise manager, it is created.

- Manufacturers and categories of the definitions that are detected by Intellisigs are created as follows:
  - For a detected software definition, an Intellisig script detects the manufacturer of the definition and links it to a category. These relationships are collected in the MDB of the domain and replicated to the enterprise manager. This process is simple if the replication finds a matching manufacturer or category in the enterprise manager with the same UUID as at the domain manager.
  - If a manufacturer or category of the same UUID does not exist at the enterprise manager and the manufacturer or category at the domain manager is provided by CA, a manufacturer or category of the same name and UUID are created at the enterprise manager with the source type as CA.
  - If a manufacturer or category of the same UUID does not exist at the enterprise manager and the manufacturer or category at the domain manager is not provided by CA, the replication searches for a manufacturer of category of the same name.
  - If a manufacturer or category of the same name is thus found in the enterprise manager, it is used to link with the detected definition.
  - If a manufacturer or category of the same name is not found, a new manufacturer or category is created and used to link with the detected definition.
  - The source type of a new manufacturer or category is either CA Intellisig or custom Intellisig, depending on the source type of the Intellisig. They are not replicated to any domains.
  - If categories and manufacturers of a detected software definition that were previously reported by Intellisigs have changed, they are unlinked from the detected software definitions on the domain manager. Categories and manufacturers that are linked manually remain unchanged.
- Discovered software releases and patches that are detected by Intellisigs are replicated from a domain manager to the enterprise manager.
- Heuristic and detected Intellisig definitions are deleted when no discovered software records are linked to the enterprise manager or domain manager, provided the Leave replicated software definitions configuration policy is set to is false. For more information about the policy, see Modify Replication of Intellisig Definitions from Enterprise Manager (see page 94)
- Any software definition category membership is replicated from the enterprise manager to the domain manager, provided that the software exists at the domain manager.

DMScript Extensions

Creating Software Definitions and Detection Records

When you create custom Intellisigs, call the DMScript functions within your Intellisig script for reporting software records that are detected on the agent computer. DMScript provides built-in functions that write detected software records to an output file.

Note: DMScript is a scripting language that provides a common means of executing commands on agents. For more information about DMScript, see the *Desktop Management Scripting Language* guide in the CA IT Client Manager bookshelf.

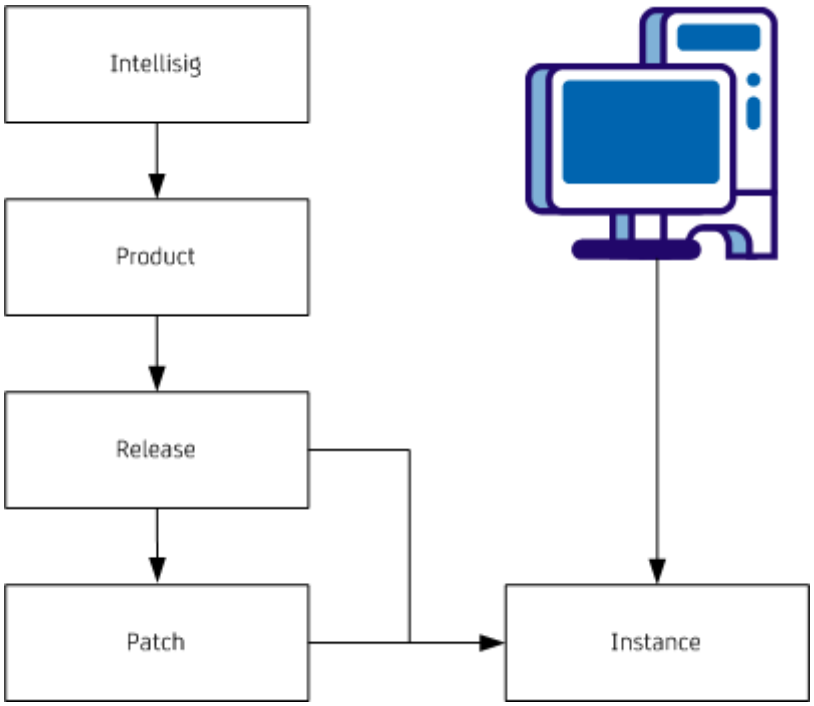
The Intellisig script must call the functions in the following order:

1. OpenDetectedSoftwareOutputFiles
2. CreateDetectedSoftwareProduct
3. CreateDetectedSoftwareRelease
4. CreateDetectedSoftwarePatch
5. CreateDetectedSoftwareReleaseInstance
6. CloseDetectedSoftwareOutputFiles

For more information about Intellisigs functions, see Desktop Management Scripting Language guide.

Hierarchy of Intellisig Objects

Understanding the hierarchy of Intellisig objects is important when you create custom Intellisigs. The objects must exist in a particular hierarchy. Each object has a parent, which must be created beforehand. The hierarchy is as follows:



- An Intellisig contains information to detect a number of products.
- A product, for example, Microsoft Office, has a number of releases such as Office 2003, office 2010, and so on.
- A release has a number of patches that contains fixes.
- A computer has a number of instances of a release or patch installed.
- Products, releases, and patches are created as software definitions and instances are created as discovered software.
- When an object is created, DMscript checks for the availability of the parent object. For example, if a release is created, then DMScript checks whether the parent product exists. This check helps ensure that proper relationships are created between the related product, release, patch, and instances.

Fixed and Variable Parameters

The functions for creating detected software output files provide a way for future extensions by accepting fixed parameters and variable parameters. While fixed parameters are mandatory, variable parameters are optional. Variable parameters typically include the properties that are associated with the Intellisig and provide a means for adding additional properties. The functions can be expanded in future to add variable parameters without impacting existing scripts.

The following guidelines are applicable for specifying and using fixed and variable parameters:

- Fixed parameters are mandatory and require values explicitly.
- Variable parameters are optional and specified in the OptionalProperties parameter. The OptionalProperties parameter is a string and can contain any number of property name and value pairs using the following format:

  ```
  property=value | property=value |…
  ```

- Use the | character to separate the property name and value pairs. Spaces before and after the | character are not significant.
- Unknown properties in the OptionalProperties generate warnings but do not stop the script. This behavior lets the agents of varying releases to process Intellisig scripts sent from the domain manager. If a script uses a property that is unknown to a particular version of DMScript, the property is ignored but a warning is written to the DMScript trace log.
- Unknown values in the optional parameters raise a warning but the value is passed on to the engine to interpret or ignore. For example, the value of the bitness parameter can only be 32 or 64 but if you pass 16, the script does not fail.

Supported Architectures

Following is the list of architecture names that you can pass to the DMscript functions that write the detected records to the software detection output file:

- AMD64
- ARM
- Alpha
- Big Endian MIPS
- HP Precision
- IA64
- IBM S/390
- Little Endian MIPS
- Motorola 680x0
- PA_RISC
- PowerPC
- SPARC
- s390x
- x64
- x86

Error Codes and Optional Properties

The LogDetectedSoftwareError method accepts the error messages in a specific format. The error messages are localizable strings that are generated when there is an Intellisig execution error at the agent. The error messages are then sent to the domain manager and displayed as a status comment against the software inventory collect task in DSM Explorer.

Example: LogDetectedSoftwareError

```
LogDetectedSoftwareError("ISE:00400","PARAM5=Microsoft Windows 7
Ultimate x64 64 en-us|PARAM6=VersionNumber=6.1.7600
|VersionLabel=6.1.7600 |Language=en-us |Bitness=64
|Architecture=x64 |Manufacturer=Microsoft
Corporation|Category=Operating Systems |Description=The Microsoft
Windows 7 Product|PARAM7=SWDETECT_BADARGS");
```

The Intellisig scanner automatically raises the following error messages to report issues related to launching an Intellisig:

Error Code	Error Text
ISE:00302	The intellisig named %1$t, version %2$t, UUID %3$t, is missing the mandatory parameter %4$t.
ISE:00303	Intellisig named %1$t, version %2$t, UUID %3$t, script file %4$t did not produce an output file.
ISE:00304	The execution of Intellisig named %1$t, version %2$t, UUID %3$t, script file %4$t has exceeded the allowable timeout (%5$t seconds).
ISE:00305	Intellisig named %1$t, version %2$t, UUID %3$t, script file %4$t could not be found.
ISE:00306	Intellisig named %1$t, version %2$t, uuid %3$t, script file %4$t, caused an internal error while attempting to start an intellisig
ISE:00307	The dmscript intepreter reported an error parsing the Intellisig named %1$t, version %2$t, uuid %3$t, script file %4$t
ISE:00405	Intellisig %1$t version %2$t, UUID %3$t, script %4$t, %5$t was called to create "%6$t" with an unknown parent "%7$t".
ISE:00406	Intellisig %1$t version %2$t, UUID %3$t, script %4$t, %5$t was called to create "%6$t" with a parent "%7$t" of the wrong type.
ISE:00407	Intellisig %1$t version %2$t, UUID %3$t, script %4$t, %5$t was called to create an instance with a parent "%6$t"of the wrong type.
ISE:00411	Intellisig %1$t version %2$t, UUID %3$t, script %4$t, %5$t was called with mandatory property "%6$t" set to blank.
ISE:00412	Intellisig %1$t version %2$t, UUID %3$t, script %4$t, the lastaccessed property was specified with an invalid value: %5$t: should be in the format: yyyy-mm-dd:hh:mm.

You can raise the following error messages to handle issues related to creating an Intellisig:

Error Code	Error Text	Function Syntax
ISE:00400	Intellisig %1$t version %2$t, UUID %3$t, script %4$t, Failed to create software product. The parameters were ProductName:%5$t OptionalProperties : %6$t. Return Code : %7$t	LogDetectedSoftwareError ("ISE:00400", "PARAM5=ProductName|PARAM6=OptionalProperties|PARAM7=ReturnCode")
ISE:00401	Intellisig %1$t version %2$t, UUID %1$t, script %4$t, Failed to create Software Release. The parameters were ProducName:%5$t ReleaseNameProductRelease:%6$t OptionalProperties %7$t. Return Code : %8$t	LogDetectedSoftwareError ("ISE:00401", "PARAM5=ProductName|PARAM6=ReleaseName|PARAM7=OptionalProperties|PARAM8=ReturnCode")
ISE:00402	Intellisig %1$t version %2$t, UUID %3$t, script %4$t, Failed to create Software Patch. The parameters were ReleaseName:%5$t PatchName:%6$t OptionalProperties %7$t. Return Code: %8$t	LogDetectedSoftwareError ("ISE:00402", "PARAM5=ReleaseName|PARAM6=PatchName|PARAM7=OptionalProperties|PARAM8=ReturnCode")
ISE:00403	Intellisig %1$t version %2$t, UUID %3$t, script %4$t, Failed to create Software Instance. The parameters were PatchName:%5$t OptionalProperties: %6$t. Return Code : %7$t	LogDetectedSoftwareError ("ISE:00403", "PARAM5=PatchName|PARAM6=OptionalProperties|PARAM7=ReturnCode")

Error Code	Error Text	Function Syntax
ISE:00404	Intellisig %1$t version %2$t, UUID %3$t, script %4$t, produced the following error : %5$t **Note:** You can specify any custom error text in PARAM5.	LogDetectedSoftwareError ("ISE:00404", "PARAM5="An unexpected error has occured")

Additional Information on Intellisigs

Certificate used for Integrity Checking of Intellisigs

CA ITCM maintains the integrity of an Intellisig by associating an encrypted hash with each script of the Intellisig in the database. The agent verifies the hashes and then executes each Intellisig. If a hash fails the verification, the agent declines to run the Intellisig and sends a message to the GUI to indicate this verification error as well as the details of the specific Intellisigs that failed the integrity checks.

Note: The hash is encrypted using the certificate x509cert://dsm r11/cn=manager signer,o=computer associates, c=us tagged with ManagerSigner. If you use custom x509 certificates, Intellisigs automatically use the custom certificate for encrypting the hash. Verify that have imported and distributed the custom certificates as per the instructions in the CA IT Client Manager Implementation Guide.

Detect Software on a 64-bit OS Using Both 32-bit and 64-bit Registry

You can specify the architecture value of a registry when creating custom traditional signatures in DSM Explorer for both registry and file entries. The specified value checks that a binary is built for the specified Windows architecture.

Follow these steps:

1. Navigate to DSM Explorer, Software, Definitions, *software category*, Release of *software category*, Properties, Recognition, Advanced, Registry.
2. Select any one of the following values in the Architecture drop-down box for a 64-bit OS:

   32

   Triggers a search in the 32-bit registry hives.

   64

   Triggers a search in the 64-bit registry hives.

Note: If you change the OS group type to UNIX, the Architecture drop-down box is disabled.

Set Default Timeout for Intellisigs

The default timeout of Intellisigs helps ensure that faulty Intellisigs do not run indefinitely on a computer. You can define a default timeout value for all the Intellisig scripts that are executed on a computer or for a specific Intellisig. You can apply a longer default time to agents where the load is high, for example, database servers, and where Intellisigs require extra time to complete. For desktop computers, where the load is low, you can configure a lower timeout value.

Follow these steps:

1. Navigate to DSM, Agent, Asset Management.
2. Define a value for the following parameter:

   IntellisigDefaultExecutionTimeout

   Specifies the default timeout value in minutes.

   Default: 5

Note: When you create an Intellisig, set its timeout value to default. This action helps ensure that the agent applies the configured default timeout. If the Intellisig has a timeout value greater than zero, the specified value is enforced when the Intellisig is run.

intellisigcmd - Command Line Tool

intellisigcmd is a command line tool for Intellisigs. This tool has the following format:

```
intellisigcmd <cmd> param1=value1 param2=value2 ... [<DB_Credentials>]
```

cmd

Specifies the import, export, or genuuid command.

DB_credentials

Specifies the database credentials of the MDB. By default, the credentials are retrieved from the comstore.

Use the following sample format to specify the DB credentials:

Example: SQLServer DB Credentials format

```
dbvendor=mssql dbhost=myhost dbname=mdb dbuser=ca_itrm dbpassword=mypwd
dbinstance=inst
```

Example: Oracle DB Credentials format

```
dbvendor=oracle dbhost=myhost dbname=orcl dbuser=ca_itrm dbpassword=mypwd
dbinstance=1521
```

intellisigcmd export—Export Intellisigs

The intellisigcmd export command lets you export Intellisigs.You can either use the DSM Explorer or use the command to export Intellisigs. For more information about using DSM Explorer, see Export Intellisigs (see page 62)

This command has the following format:

```
intellisigcmd export file=<export name> [type=xml|zip] [platform=all|windows|unix]
```

export name

Specifies the name of the Intellisig XML or zip file that you want to export. If you do not provide the file extension, and the type is xml, the command creates a folder with the supplied name.

type

Specifies whether you want to export an XML or zip file. If you do not include the type parameter, the command assumes the export type depending on the export file extension.

Valid values: xml, zip

platform

Specifies the platform to determine which Intellisigs are exported.

Valid values: all, windows, unix

Default: all

intellisigcmd import—Import Intellisigs

The intellisigcmd import command lets you import Intellisigs.You can either use the DSM Explorer or use the command to import Intellisigs. For more information about using DSM Explorer, see Import Intellisigs (see page 63).

This command has the following format:

```
intellisigcmd import [file=<import source>] [type=xml|zip]
[mode=replace|mergenew|mergeall] [updateactive=yes|no] [delete=yes|no]
```

import source

Specifies the name of the XML or zip file to which you want to import the Intellisig. If you do not provide the file extension, the command assumes the file extension depending on the type.

Note: If you want to import to an XML file, verify that the supporting directories exist in the same folder as the XML file.

type

Specifies whether you want to import as XML or zip file. If you do not include this parameter, the command assumes the import type depending on the import file extension.

Valid values: xml, zip

mode

Specifies the import mode. Following import modes are supported:

Default: mergenew

replace

Replaces existing definitions with the definition being imported. Existing definitions are lost.

mergenew

Appends new Intellisig versions to the definitions on the manager. Existing definitions are not modified.

mergeall

Appends new Intellisig versions and updates the existing definitions that are included in the import file. Intellig versions that are not defined in the import files are not modified.

updateactive

Specifies whether active Intellisig versions can be updated during the import.

Valid values: Yes, Y, true, 1 or No, N, false, 0

Default: Yes

delete

Specifies whether you want to delete Intellisigs before the import. If you do not include the delete switch, none of the Intellisigs are deleted before import.

Default: No

intellisigcmd genuuid—Generate UUIDs

The intellisigcmd genuuid command lets you generate unique UUIDs which you can use when creating custom Intellisigs.

This command has the following format:

```
intellisigcmd genuuid [num=<count>]
```

num

Specifies the number of UUIDs to be generated. If you do not specify this parameter, a single UUID is generated. Otherwise, <count> UUIDs are generated.
Valid Values: 1 to 1000

Virtual Application Images

In this release of CA ITCM, asset management supports discovery and inventory of applications virtualized with Microsoft App-V (formerly SoftGrid) and VMware ThinApp.

An application is virtualized by capturing the application data using the Microsoft App-V Sequencer or VMware ThinApp Setup Capture applications. The application is condensed into a *virtual application image*. More than one application can be virtualized into the same image, for example, in the case of suite products or applications that are tightly integrated.

Note: The new software signatures pushed down by the CA ITCM scalability server for detecting virtual application images are specially tagged, such that they are ignored by legacy agents. Therefore, agents will process the software signatures and inventory virtual applications correctly.

Unlike a regular application, which is always installed locally, a *virtual application* may be installed in a number of ways:

Streamed

In this mode, the virtual application image is not copied to the end-user's computer. The virtual applications within the image are advertised to the user, for example, on the user's Start menu or desktop. When the user opens the streamed virtual application, the required application data is streamed over the network on-demand.

Standalone

In this mode, the virtual application image is staged locally on the user's hard disk. The virtual applications within the image are advertised to the user. When the user opens the standalone virtual application, the required application data is streamed directly from the local hard disk.

Staged

In this mode, the virtual application image is installed locally on the user's hard disk, but the virtual applications within the image are not advertised to the end user. This type of installation is typically found on a server for other streamed installations. The applications within the virtual application image are not available to the local users of the computer.

Asset management can discover and inventory all three types of virtual application installation. In addition, the virtual application images themselves are reported. Virtual application images can be streamed or staged.

Asset management can discover virtual applications regardless of whether they have been deployed using CA ITCM's software delivery functionality, proprietary streaming solutions, or ad hoc distribution methods.

Note: See the *Software Delivery Administration Guide* and the *Software Delivery Help* for information about creating, packaging, and deploying virtual application images.

Streaming Applications with VMware ThinApp

VMware ThinApp does not use a proprietary streaming server for network distribution of virtual applications. Instead, applications are streamed by opening them directly from network shares on an ad hoc basis.

Asset management treats ThinApp virtual applications as being streamed only if a shortcut to the application on the share is available on the user's desktop or Start menu. Merely placing ThinApp virtual applications on a share that could be accessed by a specific computer does not enable detection of the streamed application.

Support for detecting streamed ThinApp applications is not enabled by default, but can be enabled using the Signature Scanner configuration in the Collect Tasks section under Control Panel in the DSM Explorer.

Detection of ThinApp Packages in Non-Domain Environments

The detection of streaming ThinApp packages is intermittent in a non-domain environment. Unless the per-user scan is run at a time that a user is authenticated with the server share, the scan cannot detect the contents of the share.

Typically, the scan runs immediately after logon (in which case it is quite likely the user has not been authenticated with the scalability server yet), and also whenever the full signature scan is run, which may occur only infrequently and may not occur when the user is logged on.

Therefore, CA strongly recommends using Domain/Active Directory environments for streaming ThinApp applications and deploying ThinApp virtual application packages.

Discovery Strategies

As with inventory of regular applications, asset management has two strategies to discover and inventory virtual applications: The signature scanner and the heuristic scanner.

More infomration:

Software Discovery (see page 159)

Signature Scanning for Virtual Applications

The signature scanner supports application virtualization by discovering the virtual application images, for example, a ThinApp executable or App-V .sft file, available on the agent machines.

In the MDB, virtual application images are associated with software releases and patches that the image is known to contain. This information is used to provide a full inventory of both the virtual application images and virtual applications contained within them.

Because the creation of virtual application images is performed by the end user, the CA Content team cannot provide software definitions for recognizing virtual application images. Therefore, the DSM administrator must create these virtual application image definitions. This process is automatic when a virtual application package is being registered with the software delivery library. Recognition information for virtual application image definitions can also be automatically discovered when manually creating a custom definition.

Association of a virtual application image with its contents is an important task that must be performed manually. When a virtual application image definition is created, it is placed in an unsealed state, and the content of the image is undefined. The DSM administrator must associate the image with the regular releases that it is known to contain by copying and pasting from the existing CA provided or custom-created software definitions. Wherever possible, a CA provided software definition should be used, as this will allow you to query for instances of a particular application regardless of the installation type.

Once the known software has been associated with the virtual application image, the image definition can be sealed. This action instantly updates the software inventory to reflect the new association between image and releases. Any computer with a discovered instance of the image will now also display the discovered virtual releases in its inventory.

The virtual application image definition can be unsealed again if the contents needs to be modified. When unsealed, the existing associations between image and releases remain active until the image definition is re-sealed or the changes are discarded.

Example - Using ThinApp to Create a Virtual Application Image and Software Definition

The DSM administrator uses ThinApp to create a virtual application image of Microsoft Office.

If the virtual application is to be deployed using CA ITCM's software delivery functionality, the Virtual Application Package Registration Wizard is used. (For more information about this wizard, see the Software Delivery section of the *DSM Explorer Help*.)

After registration, the administrator is prompted to create a virtual application image definition. If the software delivery library is not used, the administrator could use the Create New Virtual Application Image Definition dialog to register the virtual application image. (For more information about this dialog, see the Software Definitions section of the *DSM Explorer Help*.)

The DSM administrator chooses a suitable name for the image definition, such as “Microsoft-TA."

Next, the administrator browses the Software Definitions tree node to find the CA provided definitions for Microsoft Word, Microsoft Excel, and so on. Using copy-and-paste, these existing release definitions are associated with the newly created virtual application image definition. Once all of the known contents have been associated with the virtual application image definition, the DSM administrator seals the image.

The software inventory for a machine with the package provisioned would now contain a staged virtual application image of “Microsoft-TA,” and standalone virtual releases of Microsoft Word, Microsoft Excel, and so on. The virtual releases are special instances of existing releases that can be queried and reported on alongside regular instances of the same applications.

Scanning ThinApp Patches and Upgrades

ThinApp patches and upgrades work by not replacing the original .exe but by adding a new main container. When the original is run, it searches for any newer containers and uses the newest one instead, completely disregarding its own contents.

For example, given the following ThinApp packages:

- winrar.exe (the original ThinApp package)
- winrar.001 (an upgrade ThinApp package)

  If present, when winrar.exe is run this will be run instead.
- winrar.002 (another upgrade ThinApp package)

  This would be chosen over winrar.exe and winrar.001, and so on.

Since the original ThinApp package will always run the latest upgrade (unless the upgrade packages are removed from that directory), the signature scan will only detect the latest version of the ThinApp package.

Discovery of Unknown Virtual Application Images

Because the virtual applications are packaged by the end user, it is expected that the vast majority of virtual applications in an environment will be known to the DSM administrator, and therefore, appropriate software definitions can be created. However, it is possible that images could be missed, or introduced from a foreign source.

The signature scanner is capable of reporting virtual application images that are discovered but do not match any known software definitions. This capability is disabled by default, and must be enabled in the signature scanner configuration, which is accessible from the Collect Tasks node in the DSM Explorer.

Unknown virtual application images are reported in the *agent*, Software, Unknown subnode under each computer listed in the DSM Explorer. The virtualization technology of the package is reported. If the discovered unknown virtual application image is determined by the DSM administrator to be known, a virtual application image definition for the image can be automatically created. If not, the DSM administrator can take corrective action to investigate the unknown package further.

Note: For more information about unknown virtual application images and creating software definitions for them, see the Asset Management section of the *DSM Explorer Help*.

Heuristic Scanning for Virtual Applications

The heuristic scanner can be configured to identify virtual application images and the virtual applications contained within them. The heuristic scanner automatically associates the virtual applications with the virtual application images that contain them.

You can choose which virtual application technologies to detect in the heuristic scanner configuration dialog. Microsoft App-V applications are detected by querying the App-V Client, if installed. ThinApp Virtual Applications are detected during the Start menu and Desktop scan. Therefore, the Start menu and Desktop scans must be enabled, if ThinApp heuristic detection is enabled.

Limitations of the Heuristic Scanner

The heuristic scanner takes the software it finds at *face value*, and reports it as such. Therefore, it is possible to deceive the scanner by intentionally modifying the external appearance of a virtual application image.

The heuristic scanner creates new software definitions for the software it discovers. These instances of software cannot be correlated with regular instances of software in reports and queries.

The heuristic scanner cannot detect streamed ThinApp virtual applications.

For these reasons, CA recommends continuing to use the signature scanner to inventory virtual application images, unless you already use the heuristic scanner as the primary source of software inventory in your enterprise.

Content Utility

The Content Utility tool lets you share software definitions between independent CA ITCM installations. The new and updated software definitions provided by CA Technologies are first downloaded to an online domain (or enterprise) manager that has access to the Internet, and then can be imported into all the offline domain (or enterprise) managers.

The exported definitions are saved as text files that can be shared by installations that are either not part of the same network or run in an offline environment.

You can export the software definitions to files on a shared network location or a portable media, such as a USB device, and then import them into the offline installations.

The utility works with both CA Technologies provided and custom-created software definitions and is available as a command line utility.

Important! This utility is currently supported with SQL and Oracle databases and can run on or later releases of CA ITCM Release 12.8 manager systems. You can export and import the contents from/to CA ITCM managers Release 12.8 Release or later.

Run the Content Utility

The Content Utility is a command line tool that performs the actions described in the configuration file, content_utility.xml.

To run the Content Utility, specify the following command in the command prompt:

```
Contentutility.exe
```

Configure the Content Utility

When the Content Utility runs for the first time, it generates a configuration file named content_utility.xml in the *Installation path*\CA\DSM\bin folder. You must configure this file to set the following parameters:

- Specify whether you want to export or import or both. If you choose both, the import starts only after the export is complete.
- Specify whether to include CA-defined Intellisigs in the export or import.
- Specify whether you want to process the CA Provided or Custom Created definitions or both.
- Specify the export and import server names.

The content_utility.xml looks like this after the first run:

```
<contentutility_configuration>
        <general>
                <datadirectory>default</datadirectory>
        </general>
        <export>
                <manager>
                        <hostname>machine001.ca.com</hostname>
                        <enabled>yes</enabled>
                        <ca_provided>no</ca_provided>
                        <custom_created>yes</custom_created>
                        <ca_intellisig>no</ca_intellisig>
                        <custom_intellisig>yes</custom_intellisig>
                        <intellisig_detail>yes</intellisig_detail>
                </manager>
        </export>
        <import>
                <manager>
                        <hostname>machine101.ca.com</hostname>
                        <enabled>yes</enabled>
                        <ca_provided>no</ca_provided>
                        <custom_created>yes</custom_created>
                        <ca_intellisig>no</ca_intellisig>
                        <custom_intellisig>yes</custom_intellisig>
                        <intellisig_detail>yes</intellisig_detail>
                </manager>
        </import>
</contentutility_configuration>
```

General Section

This section has the following tag:

datadirectory

Defines the location of the export and import files. The default path is C:\Documents and Settings\All Users\Application Data\CA\software_definitions.

Export Section

In this section, you can specify the following parameters for exporting the files:

hostname

Defines the name of the DSM manager where you want to export the files.

ca_provided

Defines whether the CA Provided definitions must be exported.

custom_created

Defines whether the custom software definitions must be exported.

enabled

Specifies whether files are to be exported.

ca_intellisig

Specifies whether to export Intellisigs or not.

Import Section

In this section, you can specify the following parameters for importing the files:

hostname

Defines the name of the DSM manager where you want to import the files.

ca_provided

Defines whether the CA Provided definitions must be imported.

custom_created

Defines whether the custom software definitions must be imported.

enabled

Specifies whether files are to be imported.

ca_intellisig

Specifies whether to import Intellisigs or not.

Content Utility Log Files

The content utility generates a set of log files from which you can determine the status or result of the export or import. The following log files are located under ...\DSM\logs:

ContentUtility.log

Contains the information generated by the content utility since the time it started.

ContentUtility[hostname].log

Contains the actions performed on the host system.

.out and .err files

Contains the information generated during import that is overwritten with the last import data. These files are available in the data directory for each processed table.

Collection Modules

Asset management ships with two types of collection modules—Inventory Detection Modules and Inventory Template Modules.

The Inventory Detection Modules contain the specifications for collecting the hardware inventory information. The Inventory Template Modules define the information to be collected from a .MIF file.

These modules are available under Control Panel, Configuration, Collection Modules in the DSM Explorer.

Note: The templates scheduled for a user are executed when a particular user logs in to the asset and those scheduled for a computer are executed with the Local System account (Windows) or root (Linux/UNIX) depending on their scheduled settings.

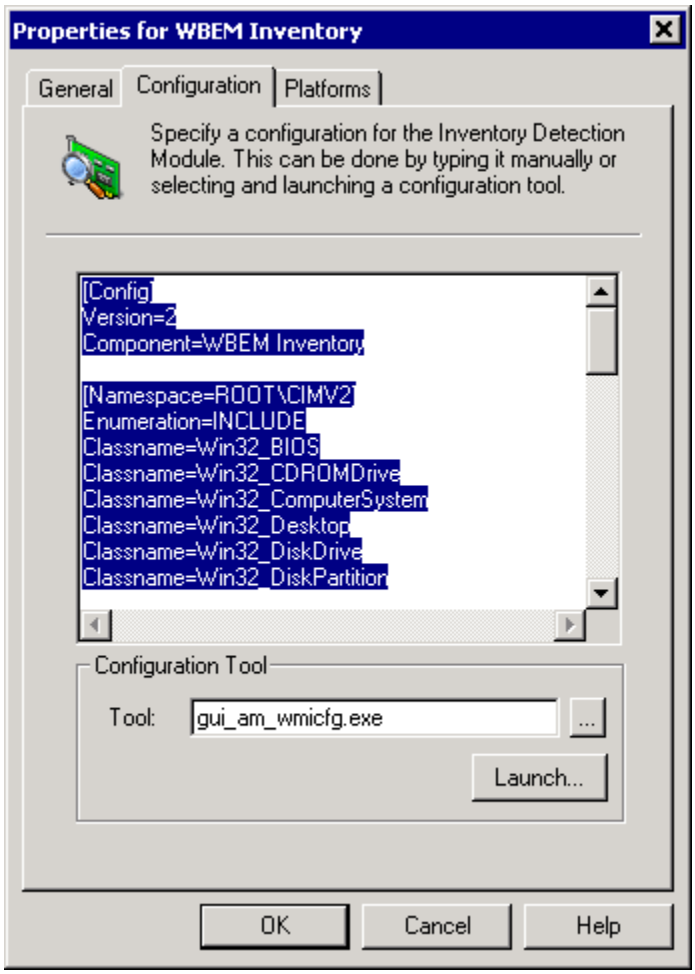
Inventory Detection Modules

Asset management provides a set of predefined detection modules that collect basic and advanced inventory information such as, general, performance, printer, WBEM, user account, and user environment inventories.

The specifications for collecting the inventory information can be any of the following:

- An executable file that has the code for collecting the values. This file can be different based on the operating system of the agent. Asset management takes care of invoking the appropriate file based on the operating system of the asset.
- A configuration tool that you can launch and select the items you want to inventory.
- Manually enter the configuration information.

The following illustration provides the sample configuration information for the inventory detection module:



Note: For more details about creating new inventory detection modules, see the *DSM Explorer Help*.

More information:

Hardware Inventory (see page 151)

Configure Inventory Detection Module

By configuring the existing inventory detection modules, you can collect additional inventory information. A best example is configuring the WBEM Inventory module. This module collects the hardware inventory information from the Windows assets. You can configure or create a new module based on this module and collect additional inventory information.

Important! You should be careful while enabling additional inventory data. Configuring a lot of information can cause heavy load on the asset and considerably reduce the network performance as the collected, transferred, and imported data can be huge.

To configure the WBEM inventory module

1. Double-click the WBEM inventory module.

   The Properties for WBEM Inventory dialog appears.

2. Click Launch in the Configuration tab.

   The WMI Configuration window appears.

3. Select the inventory items you want to add to the module and click OK.

   The inventory detection module is configured with the additional inventory items. You can add WBEM module to any of the collect tasks so that the assets running the collect task returns the additional inventory items.

More information:

Create, Configure, and Schedule Hardware Inventory Collect Task (see page 153)

View or Modify Inventory Detection Modules

By default, the detection modules define the executables from which the inventory information must be collected. If you have any source for collecting such inventory, you can modify the executable file name associated with a detection module. You can also launch the configuration tool to collect additional inventory information and specify the platforms on which the module is supposed to run.

To view or modify an inventory detection module, double-click the module. The Properties for *Detection Module* dialog appears. Modify the settings in this dialog as you require.

Note: For more details about this dialog, see the *DSM Explorer Help.*

Inventory Template Modules

Inventory Template Modules define specific inventory information you want to collect from the MIF files. You can configure these modules to prompt the logged-in user for input. The agent takes care of populating the information to a MIF file.

Asset management provides a default template, User Template, that collects company and user information. The User Template is linked to all the users in the All User Accounts group. By default, this template does not prompt the user for input. You must configure the template module to do so.

This template uses the NCUSER.MIF file to store and retrieve the company and user information. This file is available under C:\Program Files\CA\DSM\Agent\units\00000003\uam folder.

You can also create new templates to collect information for specific devices such as printers, terminals, bridges, routers, and so on. The template prompts the user for input at the scheduled time for the attributes being collected. You can create templates using the built-in Template Editor and configure it for the required users and computers.

Note: The templates configured for a user are executed when the particular user logs in (from anywhere in the domain) and those configured for a computer are executed using the Local System account. The templates are available only for Windows agents.

More information:

Create, Configure, and Schedule Template Inventory (see page 171)

Configure User Template Inventory

You can configure the User Template inventory module or any other user-defined module to specify the attributes you want to collect and configure when to prompt the user for input.

To configure the User Template inventory module

1. Navigate to Collection Modules, Inventory Template Modules.

   The existing templates in the domain are displayed.

2. Right-click the User Template module and click Properties.

   The Properties for User Template dialog appears.

3. Click Template Editor.

   The Template Editor dialog appears displaying the attributes being collected in a tree structure.

   Note: Once the values for these attributes are collected, the tree structure is replicated under the Inventory, Additional folder of the user.

4. Select any attribute and click Edit attribute.

   The Edit attribute dialog appears.

5. Select Prompt End User or Prompt End User and Force Answer in the Access field and click OK.

6. Repeat Steps 4 and 5 for the attributes you want to prompt for value.

   When the agent runs next time it prompts the user for input.

Note: For more information about creating, viewing and modifying the template modules, see the Asset Management section of the *DSM Explorer Help*.

Create New Template

You can create templates to collect user-defined inventory information.

To create a new template

1. Navigate to the Control Panel, Configuration, Collection Modules, Inventory Template Modules folder.

   The existing template modules are displayed in the right pane.

2. Click New in the Tasks section.

   The Create new Template dialog appears.

3. Enter the name of the module and click Template Editor.

   The Create Template dialog appears.

4. Enter the name of the template, specify how often you want to prompt the user for inputs, and click OK.

   The Template Editor dialog appears.

5. Add groups and create attributes under each group.

   Note: Ensure that you select the appropriate value in the Access field of each attribute.

   These groups and attributes appear will have same structure when collected from the agent computers.

6. Click OK.

   A new template is created with the given specifications.

You can enable the new template in any of the existing template inventory collect tasks or create a new collect task.

Note: Once collected the data entered for this module is available under the Computer and User Groups, All Computers, *Computer,* Inventory, Additional, *Template* folder.

Create New Template: General Tab

Use the General tab to specify the general information about the template:

This tab contains the following fields:

Name

Defines the name of the new template.

Description

Describes the template briefly.

Template Editor

Opens the Create Template dialog (see page 124) where you can specify whether you want to prompt the user for input.

Create New Template: Advanced Tab

Use the Advanced tab to specify the template file name.

Note: Typically, the default values are sufficient and need not be changed.

This tab contains the following field:

File Name

Specifies the name of the template file, where the collected template information is saved. A template file describes a component's manageable attributes in .MIF file format. This file will be present in the local asset management agent directory (typically, C:\Program Files\CA\DSM\Agent\units\00000001 or \00000003) after the asset management agent software has been executed on an agent computer.

Create Template Dialog

The Create Template dialog lets you specify how frequently you want the user to update the information.

This dialog contains the following fields:

Name

Displays the Component Name. This could be a printer, network component, or something entirely different (for example, Borrowed Equipment or Office Materials).

Note: The component name is the group header in the template editor.

After the template has been executed, the name you have entered in this field appears under the related computer's Inventory/Additional folder.

Description

Describes the header components.

Attribute Updating

Schedules all the attributes that prompt the user for information.

Prompt Questions

Specifies how often you would like this information to be updated by the user.

Default: Never

The available options follow:

Never

Displays the input box without displaying the text message. When you do not want to show any text message, select this option.

Once

Displays the input box and the text message only the first time the template collect task is executed.

Once a Month

Displays the input box and the text message on the specified day of the month.

Note: The Template collect task should be enabled for that day.

Periodically

Displays the input box and the text message as scheduled. You can specify the number of days after which the module should collect the information.

Note: The Template collect task should be enabled for that day.

Always

Displays the input box and the text message every time the template collect task is executed.

Show Text Before Prompting

Defines the text to be shown before you prompt the user to update the information.

OK

Opens the Template Editor (see page 126) where you can create a structure for your template.

Template Editor Dialog

The Template Editor dialog lets you add groups and attributes for the template you are creating. A group name can be asset numbers and attribute names. A group can be computer, monitor, printer, table, and chair. The values which the end user enters for these attributes are the respective asset numbers for computer, monitor, table, and so on.

This dialog contains the following fields:

New Group

Displays the Create new group dialog. Specify a name for the new group and a short description and click OK when you have completed your entry. A group is created with the given name. Click this button to create any number of groups and sub-groups.

After the template has been executed, the name you enter here appears under Inventory/Additional/*Template name*/*group name.*

Delete Group

Deletes an existing group. Select a group and click Delete Group. Confirm the deletion when prompted.

Edit Group

Opens the Edit Group dialog where you can edit the properties of the group.

New Attribute

Adds a new attribute to the template. Select a group in the tree structure of the left pane, to place your attribute in the appropriate group. See Create New Attribute (see page 127).

Delete Attribute

Deletes an existing attribute. Select an attribute and click Delete Attribute. Confirm the deletion when prompted.

Edit Attribute

Opens the Edit Attribute window, where you can edit the information for the attribute.

Create New Attribute Dialog

Use the Create New Attribute dialog to create new attributes to be added to a template group.

This dialog contains the following fields:

Attribute Name

Defines the name of the attribute.

After the template has been executed, the name you enter here appears under Inventory/Additional/*template name/group name/attribute name* (in the right pane only).The value entered for this attribute also appears.

Description

Describes the attribute briefly.

Type

Specifies the data type compliant with the data you plan to enter in this field.

Integer

Indicates an Integer data type, such as 64 bits.

String

Indicates a String data type, such as 256 characters (maximum length).

User Defined

Lets you define a number of user-defined values for the attribute in the User Defined Type section. When the user is prompted for a value in the agent computer, the values specified in this field appear in a drop-down list.

Access

Specifies the access type for this new attribute, such as Prompt End User or Make available.

Read-Only

Indicates that the value is read-only. The value is predefined at the time of creating the attribute. The user is not prompted for any value. This is the standard DMI-specification access type for template file Attributes.

Make Available

Makes the value editable.

Prompt End User

Prompts user on the agent computer and lets the user enter a value for the attribute.

Prompt End User and Force Answer

Prompts the user for a value and forces an answer. When you select this access type, the asset management agent displays an input box that lets the user to enter a value for the attribute, and proceeds only when a value is entered.

User Defined Type

Activates the Add and Remove buttons in this section when you select User Defined in the Type field.

Value

Specifies a list of values for the user defined data type. Click the Add and Remove buttons to add and remove the values. When you are prompted to enter a value, select from the values that appear in a drop-down list. You can click the sort button next to the Value field to sort the values in ascending or descending order.

If you have selected Integer or String as the type, you can specify the default value in this field. When prompted, you can retain the default value or change it. For example, if you know that you want to keep all your new components in one group, you can enter this group as a default value and it will appear in the template file automatically.

View or Modify Existing Inventory Templates

You can modify a template to add, edit, or remove groups and attributes.

To view or modify a template

1. Navigate to the Control Panel, Configuration, Collection Modules, Inventory Template Modules folder.

   The existing templates are displayed in the right pane.

2. Double-click the template you want to view or edit.

   The Properties dialog for the selected template appears.

3. Click Template Editor.

   The Template Editor dialog (see page 126) appears displaying the group and attributes defined in the template.

4. Click Edit Group or Edit Attribute to modify the groups or attributes of the template respectively, and click OK.

   The changes to the template module are saved.

Configure a Template for an Asset

To collect the inventory configured on a template inventory, you must configure it for the asset. Use one of the following methods:

Method 1:

1. Navigate to the Control Panel, Configuration, Collection Modules and then click Inventory Template Modules.

   All the available templates are shown in the right pane.

2. Link the required template module by dragging it to the asset or group you want to collect the information from.

   The selected template module is configured on the asset. The configured template is placed in the Computer User Groups, All Computers, *Asset Name*, Configuration, Collect Tasks folder.

Method 2:

1. Navigate to the required asset and click Configuration folder.
2. Right-click Collect Tasks and click New from the context menu.

   The Select new collect task type dialog appears.

3. Select Template Inventory and click OK.
4. Click the Templates tab and select the template or templates that need to be executed on the asset and click OK.

   The selected template module is configured on the asset.

Device Compliance Scanner (DCS)

Device Compliance Scanner (DCS) performs an automated evaluation of the target computers based on checklists that the National Institute for Standards and Technology (NIST) has created using the Secure Content Automation Protocol (SCAP).

The FDCC checklists released by NIST at the time of release of this product are by default installed when you install DCS. The scanner can also perform a compliance check on any other valid SCAP data stream, including any older or newer FDCC checklists. You can configure the scanner to perform a compliance check on additional or custom checklists.

Checklists Bundled with This Release

The following checklists are bundled with this release:

- Windows XP checklist
- Windows Vista checklist
- Windows XP Firewall checklist
- Windows Vista Firewall checklist
- IE7 checklist

For more information about these checklists, go to http://nvd.nist.gov/fdcc/index.cfm.

Note: If the checklists are valid SCAP data streams, the scanner can also process additional checklists.

How Checklists Are Distributed

When DCS scans an agent computer, it requires the SCAP checklists to be present on the agent computer. The following process explains how the checklists are distributed automatically to the agent computers and the actions to take for the automatic distribution of the checklists:

1. When DCS is installed on the domain manager, the CA ITCM installer copies the bundled FDCC checklists to the *ITCM_installpath*\SCAP_Checklists directory on the domain manager.

   Note: If you have custom or updated checklists, manually copy them to a new directory under SCAP_Checklists directory.

2. The DSM engine runs the Default SCAP Checklist Processing Job to perform the following tasks:

   - Monitor the SCAP_Checklists directory in the domain manager for new or updated checklists
   - Package the new or updated checklists in compressed archive files, digitally sign them to prevent data tampering, and save them under the \Documents and Settings\All Users\Application Data\CA\scap_checklists directory.
   - Update the MDB with the list of the new and updated checklists.
   - Create or update inventory detection modules for new or updated checklists respectively.

3. The DSM engines run the engine collect task to push the compressed archive files of the new or updated checklists to the scalability servers.
4. The agent runs the hardware inventory collect task that is configured to scan the checklists, pulls the required compressed archive files of the new or updated checklists from the scalability server, and stores them on the agent computer.
5. The agent verifies the signature on the compressed archive files. If it is unable to verify the signature, a log entry is added to the TRC_AMAGENT*.log file.

   If the signature verification failed because of a change in the DSM basic host identity certificate, redistribute the checklist files.

Note: To distribute the checklist files to the scalability servers, you must set the Distribute SCAP checklists to Scalability Servers configuration policy to True. This policy, which is set to False by default, is under Configuration Policy, Default Computer Policy, DSM, Manager, Engines in the DSM Explorer tree.

Basic Host Identity Certificate for Signing the Compressed Checklists

The digital signature of the compressed checklist files is created using the DSM basic host identity certificate, also referred to as dsmcommon. The generated signature is sent with the compressed checklist file to the scalability server, from where the asset management agent retrieves the checklist files when running a DCS scan. The agent then verifies the signature on the compressed checklist files and proceeds with the scan only if the signature verification is successful.

Redistribute the Checklists When the Certificate Changes

If the basic host identity certificate changes after the checklist has been signed and distributed, the verification of the signature on the agent will fail and the configured DCS inventory module will not run. To resolve this problem, alter the version of the checklist so that it will be redistributed with a newly generated signature to the scalability server and the Asset Management agent computer.

To redistribute the checklists when the certificate changes

1. Open the *checklist*_xccdf.xml file on the domain manager and locate the <version> tag.
2. Change the version number to enable the redistribution of the checklist.

   Note: Specify an earlier version number as this reduces the chances of a version number conflict when a new checklist is released.
3. Save the XCCDF file.
4. Open the DSM Explorer and run the Default SCAP Checklist Processing Job so that the modified checklist is compressed and signed.

   The checklist is now ready for redistribution to the scalability server.

How DCS Works

DCS is implemented as an Asset Management inventory detection module. You can configure this inventory detection module as part of a hardware inventory collect task. The following process helps you understand how the scanner works and the actions you must take for the working of the scanner:

1. CA ITCM automatically creates inventory detection modules for all the checklists placed under *ITCM_Installpath*\SCAP_Checklists folder.
2. Configure one or more hardware inventory collect tasks to schedule the scan and collect the results from the FDCC inventory detection modules. You can create a new collect task or modify the existing one to schedule the scan.
3. When the collect task runs at the agent computer, the scanner starts the scan based on the checklists available on the agent computer. Each checklist has an SCAP data stream. An SCAP data stream consists of the following files:
   - An eXtensible Configuration Checklist Description Format (XCCDF) file that defines a set of rules
   - One or more Open Vulnerability and Assessment Language (OVAL) files that specify how to check for compliance, using the rules defined in the XCCDF file
   - (Optional) A Common Platform Enumeration (CPE) dictionary file that specifies how to check whether the target computer has the required operating environment or applications. For example, if the checklist is for Windows XP, the CPE dictionary file specifies how to check whether the target computer has Windows XP.
4. The scanner parses the rules in the XCCDF file and invokes an OVAL interpreter to evaluate the OVAL definitions referenced in the SCAP data stream.
5. The interpreter produces OVAL result files that contain the values for each OVAL definition.
6. The scanner then reads the result files and determines the outcome of compliance check for each rule in the checklist and produces the following files:
   - XCCDF compliant test result file in the XML format
   - Asset Management inventory file

   Note: All the result files are stored in a subdirectory under the asset management agent's working directory.
7. The information in the inventory file is stored in the management database (MDB), and the results of the scan are displayed in the DSM Explorer and Web Console. You can create queries and reports based on this inventory information just as you do with any other inventory data.

Collection of Result Files from the Agent Computer

The scanner stores the XCCDF and OVAL result files on the agent computer by default. You can configure the FDCC inventory detection modules to enable the collection of result files from the agent computer to the scalability server. When the engine runs the collect task next time, it collects the result files from the scalability server and stores them on the domain manager. Storing the result files on the domain manager helps you manage them centrally and retrieve the files quickly when required.

Note: The result files are signed with a digital signature to prevent data tampering between the agent and the manager. If the manager is unable to verify the signature, an event is raised and logged in the default event log.

More information:

Modify the Result File Location (see page 135)

Configure the Scanner

The following sections describe the steps to configure the scanner for inventory.

Modify Windows Firewall Settings

You must modify certain firewall settings on FDCC-compliant Windows XP and Windows 7 computers to ensure that CA ITCM functions properly.

Follow these steps:

1. Click Start, Run, and enter gpedit.msc at the Run prompt.
2. In the Local Group Policy Editor window, locate the policies on the following path: Computer Configuration, Administrative Templates, Network, Network Connections, Windows Firewall, Standard Profile.
3. Change the Windows Firewall: Do not allow exceptions policy setting to Disabled.
   - Change the Windows Firewall: Allow local port exceptions policy setting to Enabled. Add the following ports to the exception ports on the firewall:
     - TCP port 4105
     - UDP port 4104
     - TCP port 4728

   The internal communications mechanisms of the CA ITCM product use the ports described here. CA ITCM cannot operate unless these ports can be accessed. Without access agents are unable to contact their manager or report inventory or status. Also, control messages cannot be passed from the manager to the agent. Communications over these ports is securely encrypted and managed by the CA ITCM product; CA ITCM Release 12.8 uses FIPS-compliant encryption.

Configure the Collection of Test Result Files

Typically, the FDCC inventory detection modules do not require further configuration, other than the configuration to collect test result files. The XCCDF and OVAL test result files are stored in a subdirectory under the Asset Management agent's working directory. To collect these files after the scan and store them centrally in the domain manager, configure the DCS inventory detection modules to enable the automatic collection of the result files.

Note: To configure other parameters in the inventory detection module, see the description of each parameter in the Creating Inventory Detection Modules for Additional Checklists section.

To configure the collection of test result files

1. Navigate to Control Panel, Configuration, Inventory Detection Modules.

   The new DCS inventory detection modules appear with the other inventory detection modules.

2. Double-click the inventory detection module you want to configure.

   The Properties for *Module Name* dialog appears.

3. Click the Launch button on the Configuration tab.

   The SCAP Configuration dialog appears with the default configuration.

4. Select the following check boxes in the General tab:

   - Collect XCCDF Result File
   - Collect OVAL Result Files

   Note: The OVAL test result files are often around 10MB. If you do not have specific reasons for storing them on the domain manager, you can collect only the XCCDF result files.

5. Click OK.

   When the collect task runs again, the engine collects the test result files and stores it on the domain manager.

   Note: The result files are signed with a digital signature to prevent data tampering between the agent and the manager. If the manager is unable to verify the signature, an event is raised and logged in the default event log.

More information:

Modify the Result File Location (see page 135)

Modify the Result File Location

When you configure the collection of SCAP result files from the agent, the result files are, by default, stored under the *ITCM_installpath*\SCAP_Result_Files directory in the domain manager. You can modify the result file location if necessary.

To modify the result file location, change the configuration policy setting SCAP Result File Location under Default Computer Policy, DSM, Manager, Asset Management. When the collect task runs next time, the engine will collect the test result files and store them in the directory specified.

Configure Hardware Inventory Collect Tasks to Collect DCS Inventory

To schedule the FDCC checklist scan and collect the test results, configure a hardware inventory collect task.

Note: If you have multiple hardware inventory collect tasks, decide whether you want to schedule the checklist scan on all of them or only on a selected few. For example, if you have grouped all your Windows Vista computers and created a specific collect task for the group, you can configure the collect task for WinVista, VistaFirewall, and IE7 checklists. However, even if you configure the checklists on all computers, the scanner will scan only those computers that meet the OS requirement.

To configure the hardware inventory collect task

1. In the DSM Explorer, navigate to Control Panel, Configuration, Collect Tasks, Hardware Inventory.

   The existing hardware inventory collect tasks appear.

2. Right-click the collect task that you want to configure and select Properties.

   The Properties for *Collect Task Name* dialog appears.

3. Click the Detection Modules tab, select the DCS inventory detection modules, and click OK.

   The changes are saved. When the collect task runs next time, it will collect the scan results for the configured checklists.

(Optional) Create Inventory Detection Modules for SCAP Inventory

CA ITCM automatically creates inventory detection modules for all the FDCC checklists placed under the SCAP_Checklists folder. When you copy a new version of an existing checklist to a new folder under SCAP_Checklists folder, the existing inventory detection module for the checklist is updated with the information from the latest version. In rare circumstances, you may want to use different versions of a checklist for scanning. In this case, you need to manually create inventory detection modules for the versions that are not currently configured for use. For example, if an inventory detection module is configured to scan the highest version always, you can create inventory detection modules for any older checklist version if you want to use the same for scanning.

To create inventory detection modules

1. In the DSM Explorer, navigate to Control Panel, Configuration, Collection Modules, Inventory Detection Modules.

   The existing detection modules appear in the right pane.

2. Right-click Inventory Detection Modules folder and click New from the Context menu.

   The Create New Inventory Module dialog appears.

3. In the General tab, specify the inventory module name. Specify a name that represents the checklist name.

4. In the Configuration tab, specify the configuration for SCAP inventory detection. Following is the sample configuration for FDCC IE7 checklist:

   ```
   [SCAP]
   SCAPPath=FDCC-Major-Version-1.2.1.0\ie7
   XCCDFFile=fdcc-ie7-xccdf.xml
   XCCDFID=fdcc-ie-7
   XCCDFVersion=v1.2.1.0
   XCCDFVersionOptions=v1.2.1.0Profileoptions=`v1.2.1.0;Federal Desktop Core
   Configuration version 1.2.1.0; CPEDictionary=fdcc-ie7-cpe-dictionary.xml
   CollectXCCDFResultFile=false
   CollectOVALResultFiles=false
   OvaldiPath=ovaldi-ca
   InvComponent=$SCAP$FDCC IE7
   ```

   Note: For more information about the parameters and their descriptions, see the appendix SCAP Configuration Parameters.

5. In the Platforms tab, select Windows 32 bit, click Win32 generic, and then enter **amiscap.exe** in the text field next to the option button.

6. Click OK.

   The inventory detection module for the configured checklist is created and appears under the Inventory Detection Modules folder. Configure one or more hardware inventory collect tasks to include the new inventory detection modules.

Additional SCAP Data Streams

In addition to the checklists bundled with this release, the scanner can scan any valid SCAP data stream. The additional SCAP data stream can be a new or an updated FDCC checklist, a custom checklist, or an SCAP data from any source.

How to Configure Additional SCAP Data Streams

CA ITCM can distribute additional SCAP data streams to the target agent computer automatically. Configuring additional SCAP data streams for automatic distribution involves the following tasks:

1. Copying the SCAP Data stream to the domain manager
2. Configuring the Hardware Inventory Collect Task (see page 135)

Copy the SCAP Data Stream to the Domain Manager

The checklist files (SCAP data stream) that you want DCS to scan must be available in a specific directory in the domain manager. The DSM engine checks this directory for new or updated checklists when it runs the Default SCAP Checklist Processing Job.

Copy the SCAP data stream to a new directory under the *ITCM_installpath*\SCAP_Checklists directory in the domain manager.

Note: You must place all the files belonging to an SCAP data stream or checklist in a directory under the SCAP_Checklists directory.

Export the SCAP Configuration

You can export the configuration information from the SCAP Configuration dialog to a .CFG file. You can use this file to import the configuration information into the SCAP Configuration dialog when creating inventory detection modules for the custom or additional SCAP data streams.

To export the SCAP configuration

1. Double-click the DCS inventory detection module, the SCAP configuration of which you want to export.

   The Properties for Module Name dialog appears.

2. In the Configuration tab, ensure that it uses the gui_am_scapcfg.exe configuration tool.

3. Click Launch.

   The SCAP Configuration dialog appears with the existing configuration.

4. From the System Menu icon in the top-left corner of the dialog, select Export Configuration.

   The Save As dialog appears.

5. Specify the file name and click Save.

   The configuration is exported to a file in the location you specified.

Import an SCAP Configuration

When you are creating inventory detection modules, you can import information from an SCAP configuration file into the SCAP Configuration dialog. Importing the SCAP configuration fills in the configuration details in the respective fields in the SCAP Configuration dialog.

To import an SCAP Configuration

1. In the SCAP Configuration dialog of the new inventory module, select Import Configuration from the System Menu icon in the top-left corner of the dialog.

   The Select File to Import dialog appears.

2. Select a valid SCAP configuration file, and click Open.

   The configuration information is imported into the respective fields in the SCAP Configuration dialog.

   Following is the content of a sample SCAP Configuration file:

```
[SCAP]
SCAPPath=FDCC-Major-Version-1.2.1.0\ie7
XCCDFFile=fdcc-ie7-xccdf.xml
XCCDFID=fdcc-ie-7
CPEDictionary=fdcc-ie7-cpe-dictionary.xml
InvComponent=$SCAP$FDCC IE7
CollectXCCDFResultFile=false
CollectOVALResultFiles=false
OvaldiPath=ovaldi-ca
```

Working with the Scanned Results

The following sections describe what you can do with the scan results.

Results Reported by the Scanner

After the compliance check, the scanner reports the following results for each rule in the XCCDF file:

Pass

Indicates that the computer has passed the compliance check for the selected rule.

Fail

Indicates that the computer has failed the compliance check for the selected rule.

Error

Indicates that there was an error while performing the compliance check for the selected rule.

Not Checked

Indicates that the rule does not contain a check defined, making it impossible for the scanner to perform a compliance check for it.

Unknown

Indicates that the characteristics being evaluated cannot be found or the characteristics can be found but collected object flag is "not collected".

Not Applicable

Indicates that the rule is not applicable to the operating environment installed on the agent computer.

View Scan Results

You can view the scan results to see if an agent computer passed or failed the compliance check. The results display against each rule in the XCCDF file. The DSM Explorer and the Web Console present the scan results in an easy-to-read format. You can also open the XCCDF and OVAL test result files to view the results of the scan.

To view the scan results from the result files

Navigate to the following directory to view the XCCDF and OVAL test result files:

- *agent working directory*\SCAP_Result_Files on the agent computer
- *ITCM_installpath*\SCAP_Result_Files on the domain manager if you have configured the collection of test result files

  Note: The paths mentioned above are the default locations of the test result files.

To view the scan results from the GUI

1. Navigate to Computers and Users, All Computers, Computer Name, Inventory, SCAP, *Inventory Component Name*.

   Note: *Inventory Component Name* is the value you specified for the Inventory Node Name field in the SCAP Configuration dialog when you created the inventory detection module.

   The scan results for the selected inventory component are displayed in various sub-nodes. For more information about the scan results, see the *DSM Explorer Help*.

Queries and Reports

You can create queries or reports based on the results produced by DCS, just as you do with any other inventory data. For more information on queries and reports, see the *DSM Explorer* online help and *DSM Reporter* online help.

Predefined Report Templates

The DSM Reporter provides the following CA ITCM Release 12.8 predefined report templates for DCS scan results:

SCAP Scan Summary

Reports the summary of the scan results for each computer in the domain manager.

Flat Score

Reports flat score results for each computer in the domain manager.

Rule Results Overview

Reports the scan results for all the rules in a checklist for a particular computer. This report invokes a runtime query that lets you filter the computers for which you want to view the rule results overview.

Patch Results Overview

Reports the scan result for all the patches in a checklist for a particular computer. This report invokes a runtime query that lets you filter the computers for which you want to view the patch results overview.

SCAP Input Files Information

Reports the details of the input files (SCAP data stream) used in a particular computer for DCS scan. This report invokes a runtime query that lets you filter the computers for which you want to view the input files information.

SCAP Output Files Information

Reports the details of the result files produced by DCS scan on a particular computer. This report invokes a runtime query that lets you filter the computers for which you want to view the output files information.

DCS Log Files

DCS logs are added to the following log files on the agent computer:

TRC_UAM_*.log

Contains the logs related to compression and decompression of the checklist files, creation and verification of the signatures for the checklist files, and the actual checklist processing.

TRC_AMAGENT*.log

Contains the logs related to compression and decompression of the checklist files, creation and verification of the signatures for the checklist files, and the actual checklist processing.

TRC_AMRAPI*.log

Contains the logs related to the transfer of checklist files and result files to and from the agent.

On the Manager, DCS scanner logs are added to the following file:

TRC_AMSCAP_FTPLUGIN*.log

Contains the logs related to the transfer of XCCDF result files, OVAL result files, and compressed checklist files to and from the DSM engine.

The log files are available under *ITCM_installpath*\logs directory. Apart from these logs, the scanner also saves the output of running the OVAL interpreter for each OVAL file in an *ovalfilename*-ovaldi-stdout.txt file under the checklist output directory on the agent computer.

File Collection

Asset management lets you centrally manage system configuration files on your network. It can backup the configuration files on the managed assets and maintain various versions of the configuration files in the database. This helps you in pushing back to a previous version of the file.

Note: File Collection is available only for Windows agents.

You can configure the name and number of backup copies of configuration files, such as AUTOEXEC.BAT, CONFIG.SYS, WIN.INI, SYSTEM.INI, or NET.CFG. You can also define specific sets of backup files at the group level, so that you can accommodate the needs of different network groups and users without having to spend hours configuring backups on a user-by-user basis. Whenever the asset management agent runs, it checks whether the configuration file is modified. If yes, it updates the DSM Explorer with the previous and current file contents.

Configuring file collection at the computer level is more efficient than at the user level due to the following reasons:

- The agent uses the credentials of Local System Account for collecting the files. These user accounts have full access rights to all the files. Contrarily, if you configure it at the user level, the user's credentials are used to collect the files. If the user does not have permissions for accessing the file, the file collection fails.
- Configuring file collection at the user level is not of much use as the file system is maintained by the computer and copies of the file do not exist for each user.
- If you configure file collection at the user level, you will see multiple copies of the same file for each user. This can hit the database performance severely.

Note: You should be careful with regard to the number of configuration files and the maximum revisions you want to backup because these will have an impact on the database, and performance of the asset and the domain manager.

More information:

Access the File Collection Folder in the DSM Explorer (see page 144)

Access the File Collection Folder in the DSM Explorer

You can access the file collection folder to configure files for backup and to view the revisions of the backed up files.

To access the File Collection folder in the DSM Explorer

1. Navigate to the following paths in the DSM Explorer depending on the level at which you want to view or configure the file collection:

Group Level

Computer Group

Domain, Computers and Users, All Computers, Group Details.

User Account Group

Domain, Computers and Users, All User Accounts, Group Details.

User Created Group

Domain, Computers and Users, *Group*, Group Details.

Note: If you do not want a specific asset to have this configuration, you can disable it at the asset level.

Computer Level

Domain, Computers and Users, All Computers, *Computer.*

User Level

Domain, Computers and Users, All User Accounts, *UserAccount.*

Domain Level

Domain, Control Panel, Configuration, File Collection.

2. Navigate to the Software, File Collection folder except at the domain level.

   The files configured for backup are displayed in the right pane. Files configured at the group level are automatically configured for back up on the member assets. You can, however, disable such file collection if you do not want it to be collected from a specific asset.

Note: Configuring file collection at the user level is not of much use because the file is maintained at the computer level and copies of the file do not exist for each user.

View Files Configured for Backup

Configuration file information for an asset displays the following:

- The files configured for backup
- The date and time when the file was last modified
- The number of revisions available in the database
- Maximum number of revisions, where revision 1 indicates the current version.

To view the files configured for backup, navigate to the File Collection folder at the group or asset level. Double-click a configuration file to view the backup revisions available.

Configure a File for Backup

To backup a file on the agent computer, you first have to configure it in the domain manager. Once configured, the asset management agent backs up this file whenever it is modified.

Note: File scan cannot retrieve files from directories for which the user is denied access to.

To configure a file for backup

1. Navigate to the Computers and Users, All Computers, *Computer*, Software, File Collection folder.

   The files configured for backup appear in the right pane.

2. Click Configure in the Tasks section.

   The Select files to collect dialog appears.

3. Select the files you want to configure and click OK.

   The selected files are marked for backup and are displayed in the right pane.

Note: If the agent is configured to File scan a network drive on a Windows agent computer, the agent must be installed to run under the same user account that maps the network drive.

More information:

Add new file collection definition (see page 146)

Add new file collection definition

If the file you want to back up or collect is not available in the list of files defined for backup, you can add a new file collection definition.

To add a new file that you want to back up

1. Navigate to the Control Panel, Configuration, File Collection folder.

   The existing file collection definitions appear.

2. Click New in the Tasks section.

   The Add File Collection Definition dialog opens.

3. Fill in the required information, and click OK.

   The selected file is added to the list of files.

Modify the File Collection Definition

You can modify a file collection definition to change any of its settings. To modify a file collection definition, right-click the configuration file and select Properties. The Configuration File Properties dialog appears. Change the required settings and click OK.

View Configuration File Backup Versions and Push-back

The asset management agent collects the configuration files from the asset if the files were modified since the agent's last run, and records it as a revision in the DSM Explorer. You can push-back any of the revisions to revert to the previous versions.

To view configuration file backup versions

1. Navigate to File Collection at the asset level.

   The right pane displays the files configured for backup.

2. Double-click a configuration file

   The right pane displays the backup revisions available for the file. Revision number 1 indicates the current version of the file.

3. Right-click a revision and select Edit and push-back

   The Edit contents to push back to Agent window appears where you can modify the contents if necessary.

4. Click OK.

   The selected revision is pushed to the asset.

How Configuration File Backups Are Managed

Asset management uses the following process to manage the configuration file backups:

1. When a file is initially configured for backup, the agent collects the contents of this file and saves it in the agent's working directory on the asset.
2. Every time the computer or user logs into the network, the agent checks the changes between the last saved file and current version of the file. As this check is performed locally on the asset, this is faster.
3. If there is a difference, the agent repeats Step 1.

Define an Event Policy

You can keep track of the changes made to the configuration files by defining an event based policy. You can configure this policy, for example, to display a message to the user or send an email to an authorized person when a user tries to modify the configuration file.

To define an event policy

Method 1

1. Navigate to Control Panel, Configuration, File Collection (see page 144).

   The right pane displays all the files configured for backup.

2. Right-click a configuration file in the right pane and select Define Event Policy.

   The Policy Designer dialog appears.

3. Select the file to be monitored in the File Collection section and add the actions you want to take when the file is modified and click OK.

   The specified actions are taken when a user modifies the selected configuration file.

Method 2

1. Navigate to Policies, Event Based, Asset File Collection.

   The right pane displays the policies in the asset file collection category.

2. Right-click Asset File Collection and select New.

   The Policy Designer dialog appears.

3. Select the file to be monitored in the File Collection section and add the actions you want to take when the file is modified and click OK.

   The specified actions are taken when a user modifies the selected configuration file.

Note: For more details about the Policy Designer dialog, see the *DSM Explorer Help*.

Collect Tasks

A collect task collects data from the assets. Collect tasks can be created at the asset, group, and domain level. Tasks created at the asset level are linked to the respective assets, and those at the group level are linked to all the member assets in the group. At the domain level, you can view all the tasks in the domain, create new tasks, and link them to any group or asset. You can create and link any number of collect tasks to a particular asset or group.

Asset management provides the following predefined collect tasks that are linked to the All Computers group and all its member assets:

Inventory Configuration

Collects the hardware inventory information from the assets. By default, it collects the information given in the General, Microsoft License, Protection, and Performance inventory modules and displays it under the Inventory folder of the asset in the DSM Explorer.

Software Inventory Configuration

Collects the software inventory information from the assets. By default, it performs a signature scanning on all the assets and displays it under the Software, Discovered folder of the assets in the DSM Explorer. After the installation of CA ITCM, the asset management agent starts collecting the software inventory information on the assets based on the defined software signatures.

User Template Configuration

Collects the company and user specific information. This task is automatically configured for all the assets in the All Computers group. By default, it collects the information given in the User Template module and displays it under the Inventory folder of the asset in the DSM Explorer.

Apart from these predefined collect tasks, you can also create new collect tasks of the following types:

- File Scan
- Hardware Inventory
- Software Discovery
- Software Usage
- Template Inventory
- Virtual Host Inventory

Note: For more information about each of these collect tasks, see the Asset Management section of the *DSM Explorer Help*.

Access the Collect Tasks Folder in the DSM Explorer

You can create, configure, link, disable, check the status, and schedule the collect tasks in the Collect Tasks folder.

To access the Collect Tasks folder

1. Navigate to the following paths in the DSM Explorer depending on the level at which you want to view or configure the asset jobs:

   Group Level

   Computer Group

   Domain, Computers and Users, All Computers, Group Details.

   User Account Group

   Domain, Computers and Users, All User Accounts, Group Details.

   User Created Group

   Domain, Computers and Users, *Group*, Group Details.

   Computer Level

   Domain, Computers and Users, All Computers, *Computer*

   User Level

   Domain, Computers and Users, All User Accounts, *UserAccount.*

   Domain Level

   Domain, Control Panel

2. From the above paths, navigate to the Configuration, Collect Tasks folder.

   The collect tasks created at the group level are automatically linked to the member assets of the group. You can, however, disable the execution of such collect tasks at the asset level. Double-click a collect task to view the member assets on which the task is linked and their respective task status.

Note: You can link the tasks created at all these levels to any managed asset or group with the exception of the virtual host inventory collect task. A virtual host inventory collect task can be created only for an asset or agent that has the AM remote agent (AMRemoteAgent) installed, and it can be linked only to other agents of the same OS class. For example, a virtual host inventory collect task created for a Windows agent can be linked only to other Windows agents.

More information:

Collect Tasks (see page 149)

Engine Collect Task

Engine collect tasks collect the information from the scalability servers and update the database. The engine collect tasks are available under the Control Panel, Engines, Engine Tasks, Collect Tasks folder. You can configure an engine collect task to perform the following functions:

Audit

Collects the audit data generated by the Software Usage collect task for which you have enabled auditing.

Backup

Collects the files configured under File Collection.

Content Distribution

Distributes new content from the domain manager to the scalability server so that target computers can retrieve the content.

Hardware Inventory

Collects the inventory information generated by the Hardware Inventory and Template Inventory collect tasks.

Software Discovery

Collects the inventory information generated by the Software Discovery and File Scan collect tasks.

Software Usage

Collects the information generated by the Software Usage collect tasks.

Status

Collects the status time stamps from the Asset Jobs and Collect Tasks.

Hardware Inventory

The hardware inventory collect task gets information about the hardware deployed on each of the computers in your enterprise. Effective asset management is not only critical for controlling equipment costs and ensuring that users have access to the necessary equipment; it is also an aid in performing LAN and WAN troubleshooting.

The hardware inventory collect task can collect both physical data such as the type of hard disk and RAM, and logical data such as the partitions in the hard disk. Asset management provides a set of predefined inventory detection modules that collect hardware inventory from various sources. You can modify these modules to collect additional inventories or create new modules.

Note: For more details about the inventory detection modules, see the Asset Management section of the *DSM Explorer Help*.

More information:

Inventory Detection Modules (see page 119)

How the Asset Management Agent Collects Hardware Inventory

To configure and schedule hardware inventory, you must first understand the actions taken by the asset management agent to collect the inventory from the agent computers. Understanding this process will help you to troubleshoot the inventory collection process, and also schedule hardware inventory based on how often you want to collect the inventory and the impact of collecting the inventory at such intervals.

The agent performs the following actions to collect the hardware inventory:

- At the scheduled time, the agent executes each collect task configured for the asset.
- Runs the appropriate executable for collecting the information under each detection module and saves it in a specific file, for example, the general inventory information is saved as ig40.inv on the agent.
- Saves the full scan files and compares these files with the previous files. It then sends the delta values to the scalability server. It does not store the delta values locally. The information sent to the scalability server is stored as i## file and if previous delta values have not been collected, the version number of the file is incremented like i01, i02, and so on. For more information on how the delta values are collected, see the Collecting Delta Values (see page 157) section explained below.
- The agent first saves this file in the Agent Working Directory (see page 29) and then puts it in the scalability server's collect area (...\ServerDB\Sector\Collect or …/Server/Sector/COLLECT in case of Linux/UNIX platforms). The agent repeats this step every time it collects the data.

  Note: If the agent runs more than once before the engine collects the data, it creates an inventory file for each run. For example, the general inventory information will be collected with the extension i01, i02 and so on. So, 24 files will be created starting from i00 to i23.
- When the engine triggers the engine collect task, it collects all the files from the scalability server, processes them, and stores the content in the database. Once processed, the engine deletes the files from the scalability server's collect area.
- New or modified inventory module configurations, including their scheduling settings, are updated on the scalability server through the collect task. The agent accepts the new or updated inventory information.

Three objects are involved in this process—agent, collect task, and engine collect task. Each of these objects has independent scheduling options. When the scheduling for any of these objects differs, you should consider the following:

- The execution of the collect task depends on the agent's run.
- Collect tasks have their own scheduling options. By default, tasks are scheduled to run always.
- The update to the database depends on the scheduling of the engine collect task.

The following example explains this scenario. The following scheduling options are specified for the three objects:

- Agent - every one hour
- Inventory collect task - Always run Job
- Engine collect task - Once a day

In this case, the agent runs after every one hour and immediately executes the collect task. However, the engine can collect the information only once a day. So, the database is updated only once a day though the agent collects the information 24 times a day. Note that this is an undesirable scenario. To have an optimum implementation, you can schedule the collect task and engine collect task to execute at the same time. This ensures that all the collected data is updated in the database.

Create, Configure, and Schedule Hardware Inventory Collect Task

The default Inventory Configuration collect task collects the general and performance inventories. If you want to collect additional inventory or restrict the inventory collected by the default collect task for a particular group or asset, you can create a new collect task, configure the required modules, and schedule the task on the assets or groups.

To create, configure, and schedule a hardware inventory collect task

1. Navigate to the Collect Tasks folder at the asset or group level and click New.

   The Select New Collect Task Type dialog appears.

2. Select the collect task type and click OK.

   The Schedule new Collect Task dialog appears.

3. Enter a unique name for the collect task and click Set Scheduling.

   The Scheduling Options dialog appears.

4. Specify the scheduling options and click OK.

   The Scheduling Options dialog closes and the Schedule new Collect Task dialog appears.

5. Click the Detection Modules tab and select the required inventory detection modules.

   The collect task is configured with the selected modules.

   Note: For more information about what each inventory detection module collects, see the Asset Management section of the *DSM Explorer Help*.

More information:

Access the Collect Tasks Folder in the DSM Explorer (see page 150)
How the Asset Management Agent Collects Hardware Inventory (see page 152)

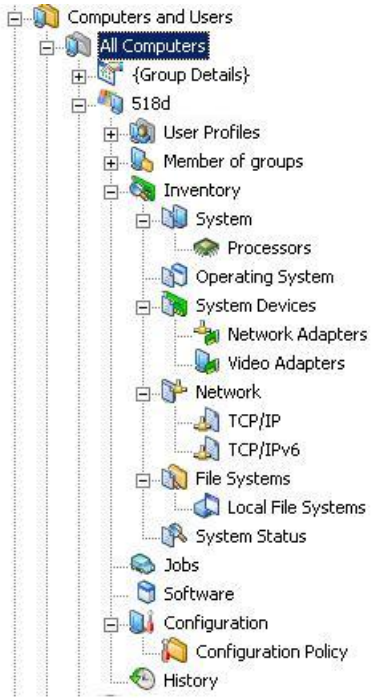
Computer Inventory

The hardware inventory collect task retrieves the hardware inventory information and displays it under the Inventory folder (such as System, Operating System, System Devices, Network, File Systems, External Devices, and so on) in the DSM Explorer. The amount of information available under the Inventory folder varies depending on whether asset management is installed or not. Thus, the structure of the hardware inventory information depends on the modules that are used for collecting the information. Based on these modules, the collected information can be categorized under the following categories:

Basic Hardware Inventory

Includes the basic hardware inventory collected by the common agent. As only the basic hardware inventory module is used in this case, only a subset of the information is collected and shown. The common agent is automatically deployed on the computers in the DSM domain. This inventory is collected regardless of whether asset management is installed or not.

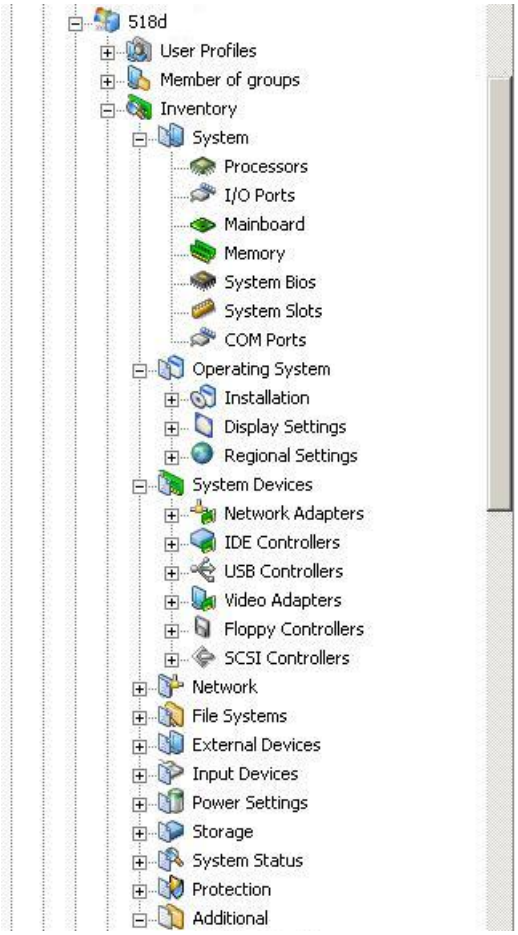
For example, the following screenshot shows the information available if the basic hardware inventory module is used:



Detection Modules Inventory

Includes the inventory information that is collected based on the configured detection modules. These inventories are collected only if you have installed asset management. By default, the General Inventory, Microsoft License Information, Performance Inventory, and Protection Inventory options are selected.

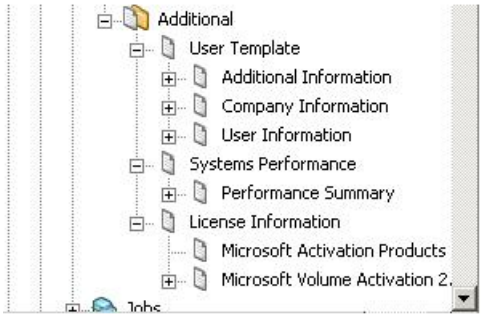
For example, the following screenshot shows the information available if the asset management inventory is used:



Additional Inventory

Includes the information collected through various modules other than the General Inventory module. This folder also includes the system performance and template inventories. These inventories are collected only if you have installed asset management.

For example, the following screenshot shows the additional inventory information available for other optional modules:



More information:

View Hardware Inventory Information (see page 159)

Collecting Delta Values

When collecting the hardware and software inventory information from the computers, the asset management agent reports only the differential (delta) values by comparing the previous scan results and the current scan results. The agent first collects the entire inventory and then compares the results with the previous scan results (available in the bak directory). It creates a DIF file and saves the difference. This DIF file is placed in the collect area of the scalability server which is then processed by the engine for updating the database. This reduces network traffic and improves the engine performance as there are fewer records to be moved over the network and processed by the engine.

By default, the agent sends only the delta values. However, you can instruct the agent to send all the collected values to the scalability server. This is called re-collecting the inventory

Note: The agent does not send delta values in case of signature scanning.

Recollect Inventory

You can configure the agent to send full inventory instead of sending delta values. This default behavior of the agent can be modified by the following methods:

Method 1:

To do this from the domain manager

1. Navigate to the Domain, Computers and Users, All Computers folder.

   The right pane displays all the computers connected to the domain.

2. Right-click the *computer* on which you want to recollect the inventory, and select Asset Jobs, Activate Job Check.

   The Asset Job Check dialog appears.

3. Select the Re-Collect option with either or both of the other rescan options, and click OK.

   The agent runs on the target computer and collects all the inventory information.

Method 2:

To do this from the agent

1. Specify the following command in the command line:

   ```
   caf start amagent args /RESCAN_SOFTWARE /RESCAN_INVENTORY /COLLECT
   ```

   The agent runs on the target computer, collects both software and hardware inventory information and saves the same in the scalability server's collect area.

   Note: If you want to collect either of them and not both, remove the rescan argument for the inventory you do not want to recollect. For example, to recollect only the software inventory, specify the following command:

   ```
   caf start amagent args /RESCAN_SOFTWARE /COLLECT
   ```

View Hardware Inventory Information

The hardware inventory information lets you know what hardware resources you own, and where they are in your organization.

To view the hardware inventory information

1. Navigate to the following folders depending on the level at which you want to view the information:

   Computer Level

   Domain, Computers and Users, All Computers, *Computer*

   User Level

   Domain, Computers and Users, All User Accounts, *UserAccount.*

2. Navigate to the Inventory folder.

   At the computer level, you can view the computer related inventory such as CPU, RAM and at the user level, you can view the information collected from the user such as, template data.

Software Discovery

The software discovery collect task collects the software installed on the assets. It lets you keep track of any unlicensed software installations and take corrective actions. It can also detect software packages on both Linux and Windows-based assets by scanning the file system and comparing software signatures as well as by querying operating system-specific package registration information.

The software discovery collect task can collect the software information using one of the following methods:

- Heuristic scanning
- Software signature (default)

Heuristic Scanning

Scans local software databases like Registry and MSI on the agent computer. You can also choose to scan the desktop and Start menu for application shortcuts.

Note: Choosing desktop and Start menu shortcuts is space- and time-consuming and may not be accurate, as you may have shortcuts on the desktop and Start menu even after you uninstall the product.

The Linux/UNIX systems do not require any heuristic scan configuration. By default, heuristic scanning on the Linux/UNIX platforms scans the RPM, PKG, and PIF databases.

Note: By default, heuristic scanning results are not replicated on the enterprise manager. But, you can configure them through the DSM Explorer. To enable or disable the replication, navigate to the Software, Discovered node and select Replicate Heuristic to Enterprise from the context menu of the Discovered node. The modified configuration is applicable to all the computers in this group. For more information, see the *Implementation Guide*.

Software Signature

Scans the hard disk of the agent computer for the executable files and then matches the files with the signatures available in the database. Software signatures are specified as software definitions in the DSM Explorer. The Default Software Contents Download job under the Engine Tasks downloads the latest signatures once a day from the CA Online Content Service. In addition, you can create new software definitions and enable them for scanning. Creating the software definitions at the domain level displays the discovered software information at the domain level only. However, if you create it at the enterprise level, the discovered software information is shown both at the domain and enterprise level.

Intellisigs

Extracts the software definition information from a defined source that the software manufacturer provides. The source can be a text file, database, registry, or binary file. For example, dsmver –f <output file> is the command that outputs version information about CA ITCM. Unless the manufacturer changes the way version information is stored for the product, Intellisigs can detect the future versions of the product also.

More information:

Enable or Disable Scan for a Definition (see page 53)
Predefined Engine Tasks (see page 414)
Custom Software Signatures (see page 45)
How to Create Custom Software Signatures (see page 47)
Signature Scanning for Virtual Applications (see page 112)
Heuristic Scanning for Virtual Applications (see page 115)

How the Asset Management Agent Collects Software Inventory

The asset management agent uses different processes to collect the software inventory information depending on the method of scanning. Understanding this process helps you to troubleshoot the software inventory collection process and also schedule and configure software inventory at an interval that best suits your organization.

Heuristic Scanning

- At the scheduled time, the asset management agent scans the selected sources for the installed software on Windows agents. On Linux/UNIX, it searches the RPM, PKG, and PIF databases.

Signature Scanning

- At the scheduled time, the asset management agent scans the hard disk of the asset for the executable files and then matches the files with the signatures available in the database.

The agent performs the following actions for both the methods after collecting installed software information:

- Creates the amapp.dat file if it uses heuristic software inventory scan. This file contains the software detected by heuristic scanning. However, if it uses software signature scanning, it creates the amsoft.xml file. This file contains the software detected by signature scanning.
- Compares the results collected from both the scans with the previous scan results, and sends the delta values to the scalability server where delta values are stored in w## files.
- When the engine triggers the engine collect task, it collects all the files from the scalability server, processes them, and stores the content in the database. Once processed, the engine deletes the files from the scalability server's collect area.

Three objects are involved in this process - agent, inventory collect task, and engine collect task. Each of these objects has independent scheduling options. When the scheduling for any of these objects differs, you should consider the following:

- The execution of the inventory collect task depends on the agent's run.
- Inventory collect tasks have their own scheduling options. By default, tasks are scheduled to run always.
- The update to the database depends on the scheduling of the engine collect task. Conversely, the engine collect task also updates the information from the database to the scalability server: new or modified agent collection module configurations and job scheduling options. Software signatures for the Software Signature scans are also updated by the engine collect task.

The following example explains this scenario. The following scheduling options are specified for the three objects:

- Agent - Every one hour
- Inventory collect task - Always run Job
- Engine collect task - Once a day

In this case, the agent runs after every one hour and immediately executes the collect task. However, the engine can collect the information only once a day. So, the database is updated only once a day though the agent collects the information 24 times a day. Note that this is an undesirable scenario. To have an optimum implementation, you can schedule the collect task and engine collect task to execute at the same time. This ensures that all the collected data is updated in the database.

More information:

View the Discovered Software (see page 165)
Access the Collect Tasks Folder in the DSM Explorer (see page 150)
Check the Status of a Collect Task (see page 192)
Create, Configure, and Schedule Software Discovery Collect Task (see page 162)

Create, Configure, and Schedule Software Discovery Collect Task

The default Software Inventory Configuration collect task is configured to perform the signature scanning. If you want to collect heuristic scanning results or the unknown software information for a particular group or asset, you can create a new collect task, configure it, and schedule the task on the assets or groups.

To create, configure, and schedule a software discovery collect task

1. Navigate to the Collect Tasks folder at the asset or group level and click New.

   The Select New Collect Task Type dialog appears.

2. Select the collect task type and click OK.

   The Schedule New Collect Task dialog appears.

3. Enter a unique name for the collect task and click Set Scheduling.

   The Scheduling Options dialog appears.

4. Specify the scheduling options and click OK.

   The Scheduling Options dialog closes and the Schedule new Collect Task dialog appears.

5. In the Methods tab, select *one* of the following:

   Heuristic Scanning

   Enables heuristic scanning. Click Configure Scanner to include or exclude the sources to be scanned. Valid options are:

   - Add/Remove Programs database
   - MSI (Microsoft Software Installer)
   - Shortcuts - Desktop
   - Shortcuts - Start menu
   - VMware ThinApp Virtual Application Images
   - Microsoft App-V Virtual Application Images

   Signature Scanning

   Enables signature scanning. Click Configure Scanner to include or exclude the sources to be scanned. Valid options are:

   - Intellisig Scan
   - Signature Scan
   - Report Unknown Executable Files
   - Report Unknown Virtual Application Images
   - Report Streamed VMware ThinApp Virtual Applications

   The asset management agent performs different actions to collect the information under these two methods.

More information:

Access the Collect Tasks Folder in the DSM Explorer (see page 150)
How the Asset Management Agent Collects Software Inventory (see page 161)

Export Discovered Software Information

The Discovered Software report displays additional details such as the installation path and the number of installations. You can export this report to a csv file and save it in your hard disk.

To view and export the discovered software information

1. Navigate to the Discovered folder in the DSM Explorer.

   The discovered software are displayed in the right pane. This information is available only if the software inventory collect task runs successfully.

2. Select the discovered software for which you want to view the report and right-click the selection.

   The context menu appears.

3. Click Details.

   The Software Report for *Computer* dialog appears.

4. Click Export.

5. Specify the csv file name and path in the Enter output file dialog and click OK.

   The report is exported to a csv file.

More information:

View the Discovered Software (see page 165)
Check the Status of a Collect Task (see page 192)

View the Discovered Software

The software inventory information lets you know the software installed on the assets.

To view the software inventory information

1. Navigate to the following folders depending on the level at which you want to view the information:

   Group Level

   Computer Group

   Domain, Computers and Users, All Computers, Group Details.

   User Account Group

   Domain, Computers and Users, All User Accounts, Group Details.

   User Created Group

   Domain, Computers and Users, *Group*, Group Details.

   Computer Level

   Domain, Computers and Users, All Computers, *Computer.*

   User Level

   Domain, Computers and Users, All User Accounts, *UserAccount.*

2. Navigate to the Software, Discovered folder.

   The discovered software are displayed in the right pane.

More information:

Software Discovery (see page 159)

Collecting Unknown Software

By default, a software discovery collect task performs a signature scan and retrieves only the registered applications—software that matches the signatures. However, you can also configure it to retrieve unknown applications, regular and virtual—software information that does not match any signatures.

The asset management agent collects either all the executable files on the hard disk of the asset or, if enabled, all of the virtual packages that do not match any of the software definitions and displays them appropriately in the following folders:

- All the registered applications are displayed under the Domain, Computers and Users, All Computers, {Group Details}, *Computer,* Software, Discovered folder.

- All the unknown regular and virtual applications are displayed under the Domain, Computers and Users, All Computers, {Group Details}, *Computer,* Software, Unknown folder.

Note: The collection of unknown software information is supported only with signature scanning.

Register Unknown Applications

You can register unknown applications to keep track of unlicensed and illegal software installations on assets, and to register any legal software so that when the agent runs next time, the software is detected as a registered application.

To register an unknown application

1. Navigate to the Unknown folder in the DSM Explorer.
2. Right-click the application you want to register and select Register.

   The Create new Release dialog appears.
3. Verify the details in the General, Recognition, and Exclude Options tab and click OK.

   The application is registered as a software signature and is listed under the Domain, Software, Definitions, Categories, All Definitions folder.

Collecting Agent Inventory Dates

The information collected by the asset management agent includes the delivery date and creation date of the software signature file used by the asset management software scanner. The System Status pane displays these inventory items for an asset. The names of the items are:

Software Signature File Creation Date

Identifies the date and time when the signature file was created by the engine and was ready for distribution to the scalability servers.

Software Signature File Delivery Date

Identifies the date and time when the signature file was delivered to the target computer.

You can include these inventory items as part of query definitions in CA ITCM.

Note: The signature creation date is not available until the engine has created a signature that has been downloaded by the agent. This action does not occur if content distribution has been disabled in the collect task.

Delete Heuristic Software Definitions and Discovered Inventory

You can delete heuristic software definitions and their corresponding discovered inventory.

To delete heuristic software definitions and discovered inventory

1. Navigate to the following folder for each domain manager in DSM Explorer: Software, Definitions, Categories, All Definitions (or other category node).

   The right pane displays all software definitions.

2. Select the definitions to be deleted and right-click.

   A context menu appears.

3. Click Delete.

   CA ITCM deletes the definitions and corresponding inventory items.

   Note: You can also disable all heuristic scanning if you no longer want to collect heuristic software data. To do so, disable the heuristic scan collect tasks on the enterprise and domain managers. For more information on how to disable heuristic scan collect tasks, see the Disable a Collect Task, Unlink a Collect Task from an Asset, and Delete a Collect Task procedures.

File Scan

The file scan collect task collects the individual files and directory structures on the hard disk of the assets and displays it in the File Explorer of DSM Explorer.

Create, Configure, and Schedule a File Scan

File scan collect tasks collect the folder and drive information from the agent computer's hard disk and displays it under File Explorer

To create, configure, and schedule a file scan

1. Navigate to the Collect Tasks folder at the asset or group level and click New.

   The Select New Collect Task Type dialog appears.

2. Select the collect task type and click OK.

   The Schedule New Collect Task dialog appears.

3. Enter a unique name for the collect task and click Set Scheduling.

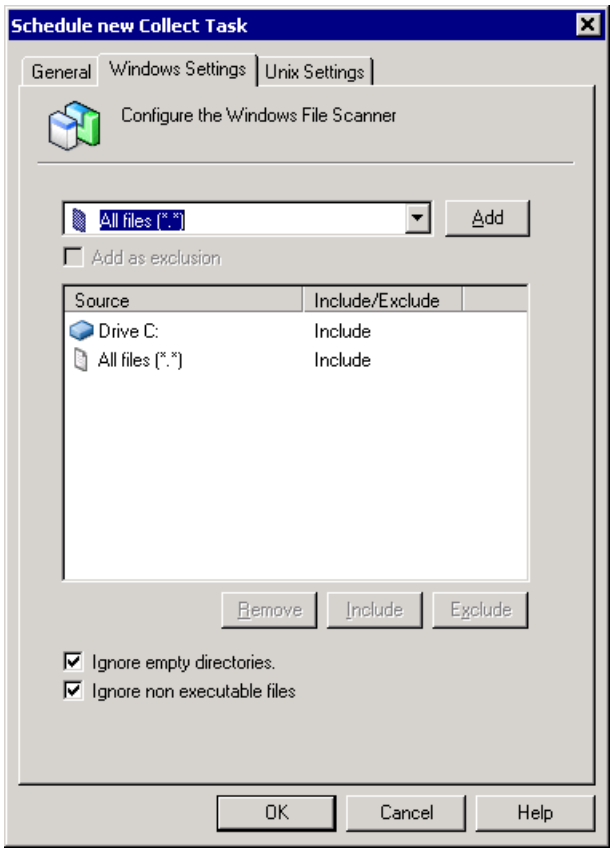
   The Scheduling Options dialog appears.

4. Specify the scheduling options and click OK.

   The Scheduling Options dialog closes and the Schedule new Collect Task dialog appears.

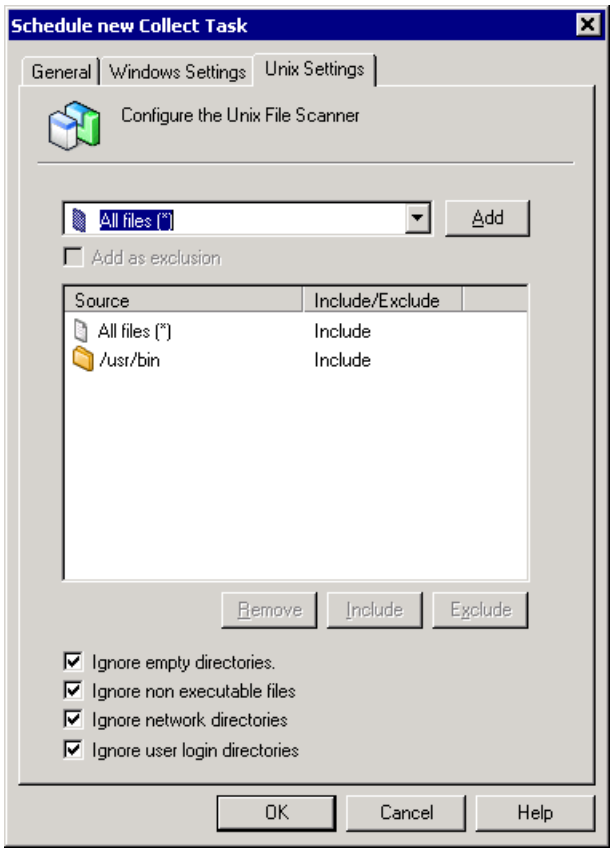
5. Select the Windows Settings tab and select the file types, folders, or drives you want to scan on a windows agent.

   The task in the following illustration is configured to collect all the executable files (.exe, .dll, .bat) in Drive C ignoring the empty directories.



6. Select the Unix Settings tab and select the file types, folders, or drives you want to scan on a Unix/Linux agents and click OK.

   The task in the following illustration is configured to collect all the executable files, ignoring empty, network, and user login directories.



   Note: File scan cannot retrieve files from directories for which the user is denied access to. On Windows, if the asset management agent is configured to file scan a network drive, the agent must be installed to run under the same user account that maps the network drive.

More information:

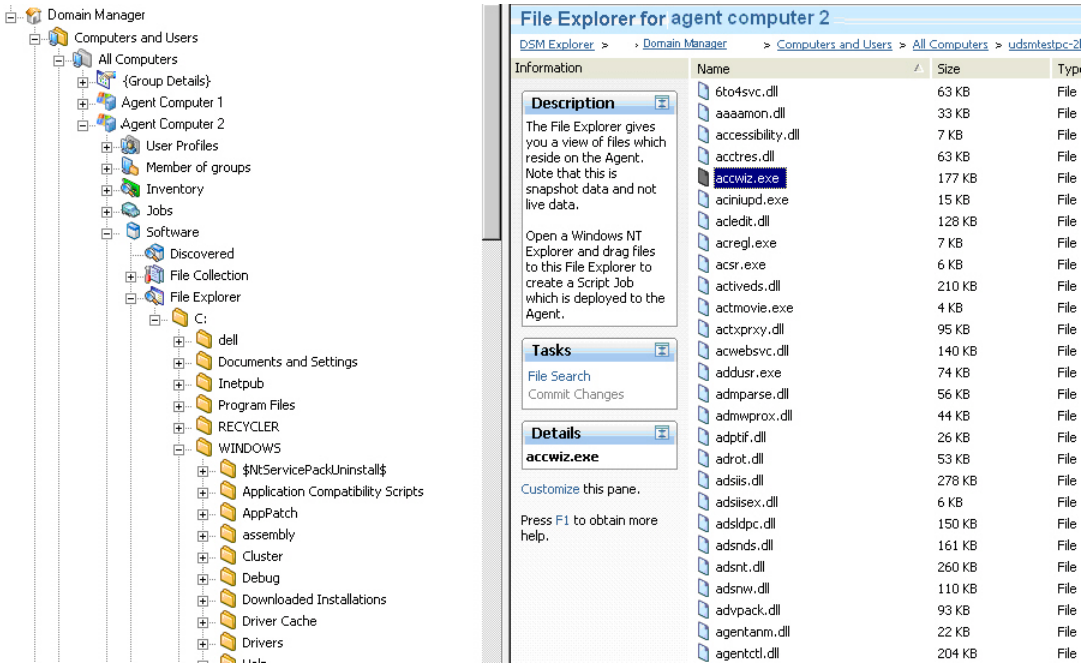
Check the Status of a Collect Task (see page 192)

File Explorer

The File Explorer displays all the files and directories collected by the file scan collect task. The look and feel of the File Explorer is similar to the Windows Explorer. You can create files and directories, copy files from the network share on the domain computer to the File Explorer of the asset, and delete files and directories. These actions are not performed on the asset immediately; instead, an auto script job is created and executed during the next run of the agent.

The File Explorer displays only the filename, directories, size, date, and attributes; you cannot see the contents of files on the remote computer.

The following illustration displays the File Explorer for an agent computer:



Note: This is snapshot data and not live data. Any changes made in the asset will take effect the next time the asset management agent runs.

Template Inventory

The template inventory collect task collects the data configured in the inventory template modules. Templates are used to collect information that is not part of the hardware or software inventories. You can create and modify templates in the Control Panel, Configuration, Collection Modules, Inventory Template Modules folder.

At the scheduled time, the agent prompts the user for input and transfers the data to the scalability server from where the engine imports to the database. This information is displayed under All User Accounts, *User*, Inventory, Additional folder.

The default User Template module is linked to the All Computers group. However, it does not prompt the user for input. You have to configure the module for prompting the user.

Note: For more information about creating and configuring the inventory template modules, see the *DSM Explorer Help*.

Create, Configure, and Schedule Template Inventory

You can create and configure a template inventory collect task to collect the inventory information configured in the selected template detection modules.

To create and configure a template inventory collect task

1. Navigate to the Collect Tasks folder at the asset or group level and click New.

   The Select New Collect Task Type dialog appears.

2. Select the collect task type and click OK.

   The Schedule New Collect Task dialog appears.

3. Enter a unique name for the collect task and click Set Scheduling.

   The Scheduling Options dialog appears.

4. Specify the scheduling options and click OK.

   The Scheduling Options dialog closes and the Schedule new Collect Task dialog appears.

5. Select the Template Modules tab, select the template modules you want to collect the inventory from and click OK.

   The template inventory collect task is saved with the above settings.

More information:

Check the Status of a Collect Task (see page 192)
Inventory Template Modules (see page 122)

Software Usage

The software usage collect task offers the software licensing administration feature which can be used to do the following:

- Monitor which software packages and applications installed in your organization are being used by what assets
- Measure the usage frequency of any application, whether it is executed from a server or local disk
- Police the number of copies of software products that are running on the computers in your enterprise to comply with license agreements and to save upgrades to new software as the users are easily identified
- Detect any application that has been started and stopped on the computer (software audit)
- Prevent the use of unauthorized software, including prohibiting users from installing software on their own and ensuring that a company uses only the admissible number of licenses
- Instead of defining what software cannot be used, a system administrator can define which applications are allowed and no other applications will then be executable on the desktop.

  Important! This configuration must be done very carefully as the user will not be able to run any other application than the specified ones. You must test your configuration in a test environment before applying it on a live environment.

You can select the applications you want to monitor, specify the number of available licenses per application, and indicate the action to be taken if the specified number is exceeded. You can specify the following actions:

- No action (Monitor only)
- Show a warning
- Prevent the application from starting

You can monitor the whole site, a particular group of assets, or a single asset. You can set up the software usage collect task to distinguish between the software packages installed on local drives and on network drives.

Though you can configure software usage collect task both at the computer and user level, configuring it at the computer level lets the software usage agent run as a service using the Local System account. This ensures that the agent is running always, regardless of which user has logged in.

Note: The software usage feature is available for all the software signature definitions defined in the database. However, for virtual applications, software usage and auditing are not supported.

Enable Usage Monitoring

You must enable usage monitoring to track the software licenses allocated to, and used by the computers in the DSM domain. By default, usage monitoring is disabled.

You can enable software usage monitoring in two ways:

- By adding the software definitions to the software usage collect task.
- By enabling the software usage monitoring for the discovered software in the Discovered Software pane.

To enable software usage monitoring in the Discovered Software pane, right-click the software you want to monitor in the Discovered Software pane of the asset or group, and select Enable Usage Monitoring. A new software usage collect task is created and linked to the asset or group. In this case, the Online Usage is disabled by default. If a software usage collect task already exists, the new definitions are added to the same task.

Software Usage Components

The software usage feature uses the following components:

Software Usage Server

The software usage server resides on the scalability server. It performs the following functions:

- Tracks and saves the online utilization data of the monitored applications.
- Keeps track of the audit information and the licenses available for each application.
- When an agent request for access to an application, the server checks the available licenses for the requested application and grants the access if a license is available and reduces the number of available license by 1. If not, it instructs the agent to perform the actions specified in the software usage collect task.
- Maintains a queue. When the number of licenses exceed, the server puts further requests in the queue and manages this queue by releasing the licenses once the existing users close the application.

Note: The software usage server can distinguish different versions of the same application, for example, Microsoft Word 97 and Microsoft Word 2000.

Software Usage Agent

The software usage agent resides on all the assets managed by asset management and is started whenever the asset management agent starts.

- For online software usage, the agent performs the following functions:

  Note: Online software usage is not supported on UNIX/Linux agents. In order to prevent malfunction of the UNIX software usage agent, the Online Metering option needs to be disabled for UNIX/Linux software usage tasks. This can be verified and adjusted using the DSM Explorer user interface. Navigate as follows: Computers and Users, All Computers, Group Details, Configuration, Collect Tasks, Software Usage. Open the Properties dialog for the Software Usage tasks, and on the Online tab, ensure that the Enable Online Software Usage option is disabled.

  - Monitors the application usage, the start and stop time of the monitored applications and whether they are started and stopped normally. The agent ignores multiple instances on one computer.
  - When a user tries to open an application that is enabled for software usage, the software usage agent takes the request to the software usage server.
  - If a license is available, the server notifies the agent which in turn, allows the user to work on the application. If not, either displays a warning message or prevents the user from accessing the application.
  - Notifies a previously prevented user about the availability of the application when the license count is below the maximum.

- For offline software usage, the agent performs the following functions:
  - Monitors the application usage, the start and stop time of the monitored applications and whether they are started and stopped normally. The agent ignores multiple instances on one computer.
  - Saves the data in the asset management agent's working directory, just like any other collected data.

Note: The software definitions must be unique. If you create two or more identical software definitions and configure the software usage agent to monitor them, only the first definition added to the software usage collect task is reported as started.

Full Utilization Data

From the DSM Explorer, you can create queries on the software usage data. You can query both online and offline software usage data. You can create advanced queries, reports, or graphs such as usage statistics or alarms on thresholds - based on this data.

Offline Utilization

In the case of offline utilization, the start and stop of the monitored applications are logged locally on the agent's working directory on the computer asset. When the asset management agent runs next time, the engine updates this information in the database. You can view the list of applications from the Offline Utilization View window.

You can also create reports based on the offline utilization data.

Note: If the offline utilization data is not displayed in the DSM Explorer, stop the running application and then start the asset management agent.

More information:

Stop and Start the Engine (see page 413)

Online Utilization

In the case of online utilization, the software usage agent continuously monitors when an application is started or stopped and sends this information to the software usage server that resides on the scalability server. Hence, at any point of time you can see the actual number of used licenses in the Online Utilization window in both tabular and graphical form. This information is stored in the file system of the scalability server for a maximum of three days, but not in the database. You can however, explicitly import the data into the database. The software usage agent communicates directly with the software usage server to know whether a license is available for the requested application or not.

Online utilization is helpful when you want to prevent the use of the application when the number of licenses exceed and inform the user who is trying to access the application. You can configure online utilization with the following options:

No action (Monitor only)

Produces only the usage start and stop entries.

Display warning message

Produces the usage start and stop entries, checks the licenses in use, and displays a warning message when the number of licenses exceed.

Prevent execution, and display prevent message

Produces the usage start and stop entries, checks the licenses in use, and prevents the execution of the application when the number of licenses exceed. You can put these users in queue and allow access when licenses are available.

Note: Online utilization is supported only on Windows agents.

Import Online Utilization Data into the Database

The online utilization data is stored in the file system of the scalability server for a maximum of three days only. If you want this data for future reports, you can import the data from the scalability server and export it to the database.

To import online utilization data into the database

1. Right-click the asset or group from which you want to import the utilization data, and select View.

   The Offline Utilization window appears.

2. Click the Data menu and select Online Utilization.

   The Online Utilization window appears.

3. Click the Data menu again and select Import to DB.

   Note: The Import to DB option is available only if you are in the Online Utilization window.

   The online utilization data in the scalability server is imported and saved in the database.

Queuing

When all the available licenses for an application are being used, you have the option of entering a queue. This queue works in the FIFO (First-In First-Out) principle: the first user in the queue obtains the next available license, and so on. This option is valid only if you have selected the prevent execution option.

Queued users will receive a notification when a license for their application becomes available. If they choose to use the application at that time, the application will be launched when they click Yes in the dialog. If the user did not respond within a time limit, the license will be released and given to the next user in queue.

Detecting Inactivity

You can configure the software usage agent to close any application which is inactive for a defined period of time, to free up used licenses. This can be configured in the software usage collect task.

Note: Monitoring application inactivity is valid only when you want to track online software usage and when you configure software usage for users. Only software usage for users can monitor the inactivity of the application but only if the application and the software usage agent run under the same account. With software usage for computers, you cannot monitor the application inactivity.

More information:

Create, Configure, and Schedule a Software Usage Collect Task (see page 181)

Auditing

Auditing registers any application that has been started and stopped on the computer without specifying the applications to monitor. The Software Usage feature includes the function of auditing executable files, DLL files, and Windows Services.

Note: UNIX Agents support auditing on executables only.

The Audit functionality is designed to run with the software usage agent. You must add one application, and then enable Auditing in the software usage collect task.

Note: Enabling Audit for all computers will generate a huge amount of network and database traffic. It must be used mainly by single assets or asset groups, for detecting new or illegal applications.

Time-stamp Synchronizing

A date and time synchronization scheme, independent of the network operating system, is built into the software usage agent to increase the correctness of the timestamps on the data received. This feature is only available when using online utilization. It uses time on the software usage server.

View Utilization Information

The DSM Explorer presents the usage information in various formats that helps you analyze the data easily.

To view the utilization information

1. Navigate to Domain, Computers and Users, All Computers/All User Accounts, *Computer/User Account*, Software, Usage folder.

   The right pane displays all the applications configured for software usage monitoring.

2. Select an application in the right pane and click View in the Tasks section.

   The Online/Offline Utilization window appears.

   In the above illustration, note the following:

   - The dotted horizontal line shows when the approved number of licenses (configurable) was exceeded.
   - From the menu bar, choose Data to find the view you want to activate. The views are Graph, Statistics, and Detail.
   - Statistics presents detailed information on the assets using the selected application in the specified time range.
   - A report on the selected application is given in the list for the time period selected in the graph.

How to Mark Time Intervals on X-axis of Graph View

While in the Graph View, you can view data for a specific time for closer inspection:

1. To view data for a specific time, select a specific point in the graph area with the left mouse button. This will display an x-axis (vertical line).
2. To view specific data at the selected point, click and hold the left mouse button and drag the cursor to mark the time interval; then release the mouse button.
3. Click the Zoom icon to view the data in the marked time interval.

Software Usage Limitations

Software Usage for Windows is subject to the following limitations:

- No usage information for DOS-based applications.
- Unicode is not supported.
- Software usage is supported on Terminal Server Console sessions, but not over Terminal Services sessions. When several instances of an application are running simultaneously, software usage counts only the first instance of the application.
- Software usage running on a Terminal Server requires that the Terminal Services service is running.

Network Time

Most networks have a utility program that can set the time of the computer to that of the network (File Server). In some cases, it may be a good idea to run this utility in the login script. This should ensure that the time of all computers is set correctly—provided that the time of the network (File Server) is correct.

Software Usage Server

The software usage server is responsible for the following actions:

- Managing licensing information as all software usage agents report to the server.
- Managing queued users and assigning licenses to them, as they become available.
- Maintaining license queuing information on applications configured for online software usage monitoring.
- Reporting license information, by request, to the DSM Explorer.

This server can run as a service on any platform that supports Scalability Server.

Note: When the software usage configuration is changed, the software usage server is updated by the new configuration. The server then resets any information about currently used licenses and queues. As a result, when an agent requests for an application next time, the license count is shown as 1.

Access Levels for Software Usage and License

The access levels of the assets define their privileges with regard to the assignment of free licenses.

Power Unit

Includes the assets that will always be able to launch the application.

VIP Unit

Includes the assets that are always preferred with the assignment of free licenses.

Ordinary Users

Includes the assets that have the lowest priority with regard to the assignment of free licenses. The assets that are not added as Power or VIP Units have the ordinary access.

Assign Access Levels to an Asset

The software usage server allocates the licenses based on the access level of the assets. You can assign appropriate access levels to the computers and users based on the priority that you set, for example, you can assign Power Unit access to those computers or users whose access to the application is very critical. You can assign different access levels to different applications.

To assign access levels to an asset

1. Right-click the computer or user for which you want to assign the access level and select Software Rights.

   The Software Rights for *asset* dialog appears.

2. Select the application for which you want to assign the rights, select the access level you want to assign to the asset, and click Add.

   The application is added under the asset name with the given access level.

3. Repeat Step 2 to assign the asset access levels to more applications and click OK.

   The access levels are assigned to the asset.

How Licensing Information Is Shared Across the Enterprise

Offline Utilization enables several software usage servers to share license information with each other across the enterprise. To configure the software usage collect task to enable this feature, you must first understand how the licensing information is shared by these servers:

1. The Scalability Servers have the offline utilization data of all the connected assets in a file system.
2. This file is replicated by each domain replication job.
3. The engine collect job updates the usage information on the database
4. Whenever an asset requests access to an application, the software usage agent checks the database for availability of license and allows access to the application accordingly.

Create, Configure, and Schedule a Software Usage Collect Task

The software usage collect task collects the software usage information from the agent computers. You can configure a software usage collect task to limit the software usage based on the available licenses, enable auditing, set the inactivity parameter, and specify the warning and prevent messages.

To create, configure, and schedule a software usage collect task

1. Navigate to the Collect Tasks folder at the asset or group level and click New.

   The Select New Collect Task Type dialog appears.

2. Select the collect task type and click OK.

   The Schedule New Collect Task dialog appears.

3. Enter a unique name for the collect task and click Set Scheduling.

   The Scheduling Options dialog appears.

4. Specify the scheduling options and click OK.

   The Scheduling Options dialog closes and the Schedule new Collect Task dialog appears.

5. Select the Applications tab and click Add.

   The Add Application for Monitoring dialog appears.

6. Select the application you want to monitor, specify the number of available licenses, define the inactivity period, and select the action to be executed when the license is exceeded and click OK.

   The Add Application for Monitoring dialog closes and the Schedule new Collect Task dialog appears.

7. Enable or disable online software usage in the Online tab, select the operating systems the agent should run on in the Platforms tab, enable or disable auditing in the Auditing tab, specify the prevent and warning messages in the Advanced tab, and click OK.

   The software usage collect task is created with the given configurations and scheduling options.

More information:

Detecting Inactivity (see page 176)

View Software Usage Information

Software usage displays a list of monitored applications that have been used by the assets and its start and stop time.

To view software usage information

1. Navigate to the following paths in the DSM Explorer depending on the level at which you want to view the information:

   Group Level

   Computer Group

   Domain, Computers and Users, All Computers, Group Details.

   User Account Group

   Domain, Computers and Users, All User Accounts, Group Details.

   User Created Group

   Domain, Computers and Users, *Group*, Group Details.

   Computer Level

   Domain, Computers and Users, All Computers, *Computer*

   User Level

   Domain, Computers and Users, All User Accounts, *UserAccount.*

2. Navigate to the Software, Usage folder.

   The software usage information available in the selected group, computer, or user is displayed in the right pane.

More information:

Software Usage (see page 172)

Virtual Host Inventory

CA ITCM supports *platform virtualization*, which is the encapsulation of computers or operating systems wherein their physical characteristics are hidden from users and they emulate the computing platform at runtime.

We use the following terms to describe the support for virtual host inventory in this document:

- A *guest* is the generic term for a virtual machine such as a VMware VM, Solaris domain and [non-global] zone, AIX logical partition, HP-UX nPar and virtual partition.
- One or more guests are handled by a *virtual host*, that is, software in the form of an application or a hypervisor. Examples are ESX Server and Sun System Controller.
- A *remote agent* is an additional part of an CA ITCM asset management agent that accesses a virtual host from another computer. This is typically done because CA ITCM software cannot be installed on the virtual host.

The virtual host inventory collect task can be configured in the DSM Explorer for those Windows, Linux, and UNIX computers and agents that have the AM remote agent installed, allowing asset management to gather information about the virtual hosts in your enterprise. Depending on the capabilities of the virtual host, it can collect data such as host type, name of the virtual machine, serial number, and memory size.

Note: The virtual host inventory collect task can only be created for a specific asset. Unlike the other types of collect tasks, the virtual host inventory collect task cannot be created for or linked to a user account. Navigate to the relevant asset to create this collect task.

More information:

Virtual Hosts (see page 322)

How to Create Virtual Host Inventory Collect Tasks

You can create and configure virtual host inventory collect tasks to detect and collect detailed information from virtualization servers about one or more virtual hosts.

To create a new virtual host inventory collect task, the DSM administrator uses the Select New Collect Task type dialog and performs the following steps:

1. Create the new collect task, choosing the Virtual Host Inventory option, and set its scheduling options.
2. Configure remote inventory collection (see page 185).
3. Add virtualization servers (see page 185).

Configure Remote Inventory Collection

Use the Virtualization Servers tab page to add, remove, and view the properties of virtualization servers you want to configure for remote inventory collection.

To add a virtualization server to the collect task

1. On the Virtualization Servers tab page, click Add.

   The Add the Virtualization Server to Configure dialog appears.

2. Select a Virtualization Server type and enter the appropriate details. (See Add Virtualization Servers for more detailed information.)

   Note: You can add up to five virtualization servers per collect task. If the same agent must collect data from more than five virtualization servers, you can define additional collect tasks for the same agent.

3. Click OK.

   You are returned to the Virtualization Servers tab page.

4. If the information in the list view is correct, click OK.

   You are returned to the Select New Collect Task type dialog.

5. Click OK.

   The RVI configuration information is transferred to the specified agent computer at the next scheduled runtime.

Add Virtualization Servers

Use the Add the Virtualization Server to the configure dialog to select a virtualization server type and add one or more virtualization servers.

Follow these steps:

1. Select a virtualization server type from the Type drop-down list. Valid types are:
   - HP
   - IBM
   - Sun
   - VMware vSphere Hypervisor (ESX/ESXi)
   - VMware vCenter Server
   - Citrix XenServer
   - Microsoft Hyper-V
2. Enter the virtualization server name or IP address in the Host Name field.
3. Enter the Web Service user name in the Username field.

4. Depending on the specified virtualization server type and the OS type of the agent computer, enter the following information.

   Note: Required fields are listed in the table that follows this procedure.

   Password

   Specifies the Web Service password of the specified user.

   Note: Regarding passwords and SSH keys, only one or the other is required for authentication. We recommend using SSH keys because they are more secure than plain text passwords.

   Host Serial

   (IBM HMC) Specifies the host serial number.

   Web Service URL

   Specifies the URL of the Web Service.

   VMware vSphere Hypervisor (ESX/ESXi) and VMware vCenter Server

   The URL takes the following form:

   `https://`*`ESXHostFQDNservername`*`/sdk`

   Citrix XenServer

   The URL takes the following form:

   `https://`*`XenMasterFQDNservername`*

   The variable in each instance represents the fully qualified host name of the server. Alternatively, an IP address can be given instead of the host name.

   The default behavior for this field is for it to be auto-populated unless the RVI Edit URL configuration policy has been changed. For more information about this policy, see Asset Management (Administration Console) Policy Group.

   SSH key location

   Specifies the location of the Secure Shell (SSH) *private* key. Provide an SSH key location for a Linux/UNIX host only if the key is not in the default location.

   Note: An empty passphrase must be provided when generating the SSH key.

5. Verify that the Enabled for remote inventory collection option is selected (default setting).

6. (VMware vCenter Server and Citrix XenServer) Click the Filter tab to specify filtering options before going to the next step.

7. Click OK.

   You are returned to the Virtualization Servers tab page.

8. Click OK.

   The new virtual host inventory collect task is added to the Collect Tasks folder.

More information:

Asset Management (Administration Console) Policy Group (see page 203)

The following table indicates which fields are required based on the specified virtualization server type and the OS type of the agent computer:

Target/Remote Agent Host	Windows Agent Host	Linux/UNIX Agent Host
HP Integrity Virtual Host	Password or SSH key location	(Optional) SSH key location
IBM Hardware Management Console (HMC)	Host Serial Password or SSH key location	Host Serial (Optional) SSH key location
Sun System Controller	Password or SSH key location	Password or SSH key location
VMware vSphere Hypervisor (ESX, ESXi)	Password Web Service URL	Password Web Service URL
VMware vCenter Server	Password Web Service URL	Password Web Service URL
Citrix XenServer	Password Web Service URL	Password Web Service URL
Microsoft Hyper-V	Password	Host Name

Viewing Remote Agents and Virtual Hosts

Virtualization inventory appears in the DSM Explorer under the All Computers node with new and updated nodes as follows:

Release 12.5 Nodes	Release 12.8 Nodes
Configuration	Virtualization/VMware ESX Configuration
Supported Features	Virtualization/VMware ESX Configuration/Supported Features
Virtualization/Datastores	Virtualization/VMware ESX Datastores
Virtualization/Service Console	Virtualization/VMware ESX Service Console **Note:** This node is available only for ESX, not ESXi.
N/A	Virtualization/VMware Virtual Machines

Release 12.5 Nodes	Release 12.8 Nodes
N/A	Virtualization/Citrix XenServer Configuration
N/A	Virtualization/Citrix XenServer Virtual Machines
N/A	Virtualization/Citrix XenServer Storage Repositories
N/A	Virtualization/Microsoft Hyper-V Configuration
N/A	Virtualization/Microsoft Hyper-V Virtual Machines

In CA IT Client Manager, you can search for and view all discovered AM remote agents that collect inventory for virtualization servers, virtual hosts, and guests. Using the Query Designer, the query results are listed under All Computers from where you can configure an agent for collection.

More information:

Relationships Between a Virtual Host and Guests (see page 351)
Viewing Virtual Host Inventory (see page 349)

View All Remote Agents

When a remote agent is installed in CA IT Client Manager, it is registered as a new agent component named AM remote agent, which represents the collecting entity. The AM remote agent appears in the Agent components list when viewing the agent properties of a computer.

Note: The AM virtual agent (AMVirtualAgent) now represents the collected entity in this release. The AM virtual agent serves to indicate that the virtual host contains no real agents, but rather only a virtual agent.

You can create a custom computer query based on agent components to view all remote agents. The query results list all agents that have the AM remote agent installed.

Follow these steps:

1. Right-click Queries in the DSM Explorer and select New from the context menu.

   The Select Target dialog appears.

2. Select Computers and click OK.

   The Query Designer dialog appears.

3. Click General Information, Computer.

   The Select Field dialog appears.

4. Select Agent Component from the list view, and then select AMRemoteAgent from the Component drop-down list in the Arguments group.

5. (Optional) Select the Version option and then select a version number from the corresponding drop-down list.
6. Click OK.

   You are returned to the Query Designer.
7. (Optional) Click Preview to run the query.

   The results are displayed in the Query Preview pane.
8. Click Close.

   You are returned to the Query Designer.
9. Click OK.

   The Save Query dialog appears.
10. Enter a name for your query and Click OK to save it.

    The query is created with the specified name and added to the Queries, Computers subfolder.
11. (Optional) Right-click on a discovered agent in the Query pane and select Navigate To from the context menu.

    You are taken to the agent node under All Computers where you can opt to configure the agent for remote virtual host inventory collection.

View All Virtual Hosts

Virtual hosts appear as normal nodes under All Computers along with all the other computers.

If you want to view only virtual hosts, you can create a custom computer query based on the class ID of the computer.

To view only virtual hosts

1. Right-click Queries in the DSM Explorer and select New from the context menu.

   The Select Target dialog appears.
2. Select Computers and click OK.

   The Query Designer dialog appears.
3. Click General Information, Computer.

   The Select Field dialog appears.
4. Select Classification, Class or subclass from the list view.

5. Select the appropriate class from the Classification drop-down list in the Arguments group box. Relevant options are as follows:

Class	Subclass
Unclassified	
	RTOS
	Sun RTOS
Virtual Host Machine	
	HP Chassis
	IBM Server
Hypervisor	
	VMware ESX
	VMware ESX Server 3.5
	VMware ESX Server 4.0
	Sun eXtended System Control Facility (XSCF)

6. Click OK.

   You are returned to the Query Designer.

7. Click OK.

   The Save Query dialog appears.

8. Enter a name for your query and Click OK to save it.
9. The query is created with the specified name and added to the Queries, Computers subfolder.
10. Select Run Query from the Tasks portlet.

    The results are displayed under the All Computers pane.

Note: Since there is no agent component installed for virtual hosts, no jobs or configuration can be applied to them. Hence, the asset context menu is disabled.

Related Computers/Related Host Computers View

You can view the relationships between computers, including the relationships between virtualization platforms. Selecting a server, for example, a VMware ESX server, under All Computers displays the managed guest machines as related computers in the Related Computers pane.

Likewise, selecting a managed guest, for example, a VMware guest computer, under All Computers displays its related host server under the Related Host Computers pane.

Note: Only managed guests, that is, those with a DSM agent installed on them, have nodes under All Computers. Unmanaged guests do not appear in the Related Computers pane.

Link Existing Collect Tasks

If you want to perform the same collect task on various assets, you can create the collect task once and link it to all other assets. You can link the collect task created at the domain level, group level, and asset level.

Note: A collect task created at the domain level can be dragged and dropped on to the target assets.

To link an existing collect task to an asset

1. Navigate to Computers and Users, All Computers, *Computer*, Jobs, Asset Jobs.

   All the collect tasks linked to the computer appear.

2. Click Link Existing Collect Task in the Tasks section.

   The Select Collect Tasks dialog appears.

3. Select the collect tasks to be linked and click OK.

   The selected collect tasks are linked to the asset.

Note: A virtual host inventory collect task can be created only for an asset or agent that has the AM remote agent (AMRemoteAgent) installed, and it can be linked only to other agents of the same OS class. For example, a virtual host inventory collect task created for a Windows agent can be linked only to other Windows agents.

Unlink a Collect Task from an Asset

You can unlink a collect task that you no longer want to execute on an asset or an asset group.

To unlink a collect task, right-click the collect task and select Unlink from the context menu.

Note: Unlinking a collect task from the asset does not delete the collect task from the domain.

Check the Status of a Collect Task

The inventory information displayed in the DSM Explorer depends on whether the collect task has run successfully or not.

To check the status of a collect task at the asset level

1. Navigate to the Collect Tasks folder.
2. Right-click a collect task and select Status.

   The Module Status dialog appears with the status details of the collect task.

   Note: For more details about the status and status comments, press F1.

More Information

Access the Collect Tasks Folder in the DSM Explorer (see page 150)

Disable a Collect Task

When you create a collect task, you enable scanning automatically. You can disable scanning on agents, groups, and user accounts by disabling the collect task.

To disable a collect task

1. Navigate to the following folder in DSM Explorer: Computers and Users, All Computers.
2. Open the Group Details folder to disable the collect task on all agents on the domain, or open an agent folder to disable the collect task for an individual agent.

3. Open the Configuration folder and click Collect Tasks.

   The right pane displays the collect tasks that apply to the group or agent.

4. Click a collect task to select it and right-click.

   A context menu appears.

5. Select Disable.

   The collect task is disabled for the individual agent (if you selected an agent) or for all agents in the group (if you selected Group Details).

Delete a Collect Task

You can delete a collect task that you no longer use.

To delete a collect task

1. Navigate to the Control Panel, Configuration, Collect Tasks folder.

   The various collect task types are displayed in the right pane.

2. Double-click the collect task type of the task.

   All the collect tasks available under the type are displayed.

3. Right-click the task that you want to delete and select Delete.

   The Confirm Delete message appears.

4. Click Yes.

   The task is unlinked from all the linked assets and groups and deleted from the database.

Configuring the Collect Task to Control Content Distribution

You can control when content is distributed from the domain manager down to the scalability server and managed agents. An engine collect task performs this distribution of content. You can configure this collect task to distribute content automatically to the scalability server if it detects new content on the domain manager. You can also configure the engine collect task to exclude automatic content distribution.

Configure Engine Collect Task to Exclude Automatic Distribution

You can configure the functions that a collect task performs from the CA ITCM DSM Explorer GUI.

To configure an engine collect task to exclude automatic distribution

1. Launch the DSM Explorer and navigate to the Engine Tasks, Collect node using the following path:

   Domain, Control Panel, Engines, Engine Tasks, Collect

   The collect tasks that have been configured on the selected domain manager appear.

2. Right-click the collect task you want to configure and select the Properties option.

   The Properties dialog displays the properties of the collect task.

3. Click the Collect tab.

4. Select Content Distribution in the Selected Tasks list and click the left arrow button.

   Content Distribution moves to the Available Tasks list and is excluded from the collect task.

5. (Optional) Move tasks from the Available Tasks list to the Selected Tasks list by selecting them and clicking the right arrow button.

   The Selected Tasks list displays the functions to be included in the collect task.

6. (Optional) Move tasks from the Selected Tasks list to the Available Tasks list by selecting them and clicking the left arrow button.

   The Available Tasks list displays the functions to be excluded from the collect task.

7. Click OK to confirm the changes.

8. Select Yes when prompted to re-initialize the status of all engines linked to this task.

   The collect task is configured and the Content Distribution function is not included.

On-Demand Content Distribution

If you have configured your engine collect task to exclude content distribution, there may be times when you do want to distribute the current content from the engine to the scalability server. To make this distribution possible, you can use the CADSMCMD command line utility to request content distribution from the MDB to the scalability server.

To perform an on-demand distribution of content, use the following CADSMCMD command options:

```
cadsmcmd scalabilityServer action=collectSector name=<servername>
CONTENTDISTRIBUTION
```

This command forces a content distribution to occur from the MDB to the specified scalability server, regardless of the default collect job configuration.

Note: For more information, see the *CLI Reference Guide*.

Platform Virtualization

Remote Virtualization Inventory (RVI)

The AM remote agent collects the following RVI data from the virtual hosts:

- Hardware inventory of the virtualization server that hosts virtual machines or manages such systems
- Software patch information for the virtualization server (if available)
- Virtualization information, such as data centers, clusters, and resource pools
- Hardware information for each virtual machine (VM), such as the guest name, UUID, power state

This information is collected for a VM regardless of its state—powered on, suspended, powered off. However, certain information is only available when the VM is powered on and the virtualization tools have been installed.

Note: The RVI agent cannot collect certain BIOS inventory attributes from Linux virtual machines that are running on Microsoft Hyper-V and XenServer servers. Furthermore, there is no support for inventory collection from Microsoft System Center Virtual Machine Manager (SCVMM).

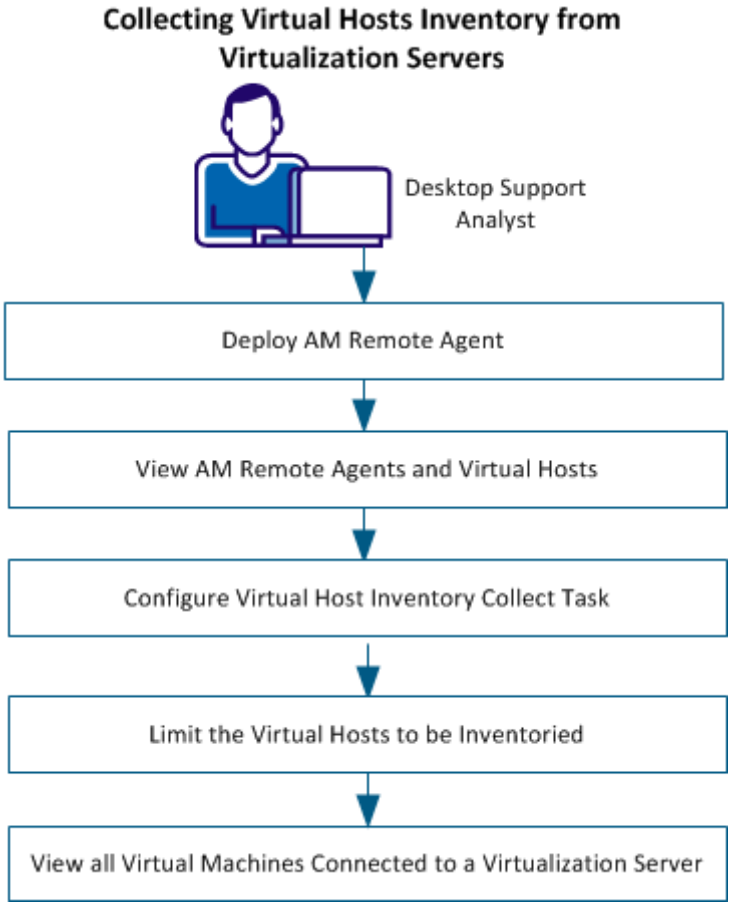
More information:

Configure Virtual Host Inventory Collect Task (see page 198)

Collecting Virtual Hosts Inventory from Virtualization Servers

As a Desktop Support Analyst, you can collect the RVI (Remote Virtualization Inventory) data from multiple virtual hosts. A virtual host is a virtualization server that hosts numerous virtual machines. Collecting the inventory on virtual hosts identifies the relationship between the virtual host and the related virtual machines. When you register a virtual machine, the domain manager does not identify it as a virtual machine.

The inventory collection process identifies the virtual machines as virtual by adding additional properties to the computer. Additionally, the RVI data includes hardware and software patch information about the server. The data also includes the virtualization information and hardware information of each of the virtual machines.



Deploy AM Remote Agent

To collect RVI data from virtual hosts, deploy the AM remote agent on a physical or virtual computer. You can deploy the AM remote agent on multiple computers depending on the number of hosts and virtual machines you want to manage.

Follow these steps:

1. Open DSM Explorer and navigate to Control Panel, Deployment, Infrastructure Deployment Wizard.
2. Follow the instructions in the wizard for deploying the agent and select one of the CA DSM Agent AM RVI plug-in packages.
3. Complete the wizard to begin deployment.

   The deployment begins on the target computer.
4. Click Control Panel, Deployment, Deployment Job Status.

   The status of the deployment job is displayed.

Download the RVI Inventory Collection DLL

To collect RVI inventory from the AM Remote agents for XenServer, you must download the inventory collection DLL manually from a third-party FTP download site.

Follow these steps:

1. Open http://opensrcd.ca.com/ips/07400_4 and do one of the following depending on the AM remote agent platform:

   Windows:

   Copy \WindowsProductFiles_x86\libxenserver.dll to *DSM_install_path*\bin on the AM remote agent computer.

   Linux:

   Copy \LinuxProductFiles_x86\libxenserver.so to *target*/opt/CA/SharedComponents/lib on the AM remote agent computer.

   The AM remote agent is configured to collect RVI inventory.

View AM Remote Agents and Virtual Hosts

Only AM remote agents can collect virtual host inventory data. You must know the AM remote agents and virtual hosts in your environment so that you can decide the virtual hosts you want to assign to each AM remote agent. For example, you have nine virtual hosts and want to assign three hosts per AM remote agent. You can view all the AM remote agents and hosts available and can decide the hosts that you want to assign to each AM remote agent.

Follow these steps:

1. Right-click the Queries node in DSM Explorer and select New.

   The Select Target dialog opens.

2. Select Computers and click OK.

   The Query Designer dialog opens.

3. Navigate to Insert Argument, General Information from the Query and select Computer.

   The Select Field dialog opens.

4. Do the following steps depending on whether you want to view remote agents or virtual hosts:

   - **View All Remote Agents**

     a. Select Agent Component from the list view, and then select AMRemoteAgent from the Component drop-down list in the Arguments group.

     b. (Optional) Select the version number from the Version drop-down list.

     Note: Alternatively, you can use the Assets with the AM Remote Agent Component query to view all remote AM agents.

   - **View All Virtual Hosts**

     a. Navigate to Classification, Class or Subclass from the list view.

     b. Select the appropriate server name from the Classification drop-down list in the Arguments group.

     Note: If you do not see the server name in the list, click <more..> at the end of the list. Increase the Result limit to a higher number if you still do not see the server name in the Search possible value dialog.

5. Click OK.

   You are returned to the Query Designer dialog.

6. Click Preview.

   The query returns the list of remote agents or virtual hosts that are based on your criteria.

Configure Virtual Host Inventory Collect Task

You can configure the virtual host inventory collect task to collect RVI data from virtualization servers.

Note: You must have installed Tools on the Hypervisors to collect the complete guest operating system inventory. Also, note that RVI data cannot be collected from Linux virtual machines that are running on Microsoft Hyper-V servers.

Follow these steps:

1. Open the DSM Explorer and navigate to Computers and Users, All Computers, *AM remote agent*, Configuration, Collect Tasks.
2. Right-click Collect Tasks and select New.

   The Select new Collect Task type dialog opens.
3. Select Virtual Host Inventory and click OK.

   Note: The Virtual Host Inventory type is available only for assets that have the AMRemoteAgent installed.

   The Schedule New Collect Task dialog opens.
4. Enter the name of the collect task in the General tab and click Set Scheduling. Specify the scheduling options for the collect task and click OK.
5. Click Add from the Virtualization Servers tab.

   The Add the Virtual Host to Configure dialog opens.

   Note: You can add only five virtualization servers to a collect task. If the same agent must collect data from more than five virtualization servers, you can define additional collect tasks for the same agent.

6. Select the virtualization server type, enter the necessary information in the following fields for the selected virtualization server type, and click OK.

   Note: Some of the fields are disabled, if the information is not applicable for the selected server type.

   Host Name

   Specifies the FQDN of the virtual host you want to inventory.

   User Name

   Specifies the user name of any user on the virtual host. The AM remote agent uses this credential for logging in to the system and collecting inventory.

   Password

   Specifies the password of the specified user.

   Note: You can either use the password and SSH key for authentication. We recommend using SSH keys because they are more secure than plain text passwords.

   Host Serial

   (IBM HMC) Specifies the host serial number.

   Web Service URL

   Specifies the URL of the Web Service.

   The URL takes the following form for VMware vSphere Hypervisor (ESX/ESXi) and VMware vCenter Server:

   `https://ESXHostFQDNservername/sdk`

   The URL takes the following form Citrix XenServer

   `https://XenMasterFQDNservername`

   The variable in each instance represents the fully qualified host name of the server. Alternatively, an IP address can be given instead of the host name.

   By default, this field is auto-populated unless you have changed the RVI Edit URL configuration policy. For more information about this policy, see Asset Management (Administration Console) Policy Group.

   SSH key location

   Specifies the location of the Secure Shell (SSH) private key. Provide an SSH key location for a Linux/UNIX host only if the key is not in the default location.

   Note: Provide an empty passphrase when generating the SSH key.

   Enabled for remote inventory collection

   Specifies whether remote inventory collection is enabled. Verify that this option is selected.

7. (Optional) Click the Filter tab and select the hosts belonging to a specific data center or cluster. Perform this step only when you want to limit the hosts from which the inventory is collected. For more information, see Limit the Virtual Hosts to be Inventoried (see page 201).

   Note: The Filter tab is displayed only for VMware vCenter Server and Citrix XenServer.

   The filter is added to the collect task.

8. Click OK.

   The collect task runs at the agent computer at the next scheduled runtime and collects the RVI data from the selected servers.

   Note: You can link the virtual host inventory collect task only to other agents of the same OS class. For example, a virtual host inventory collect task that is created for a Windows agent can be linked only to other Windows agents.

More information:

Remote Virtualization Inventory (RVI) (see page 195)

Limit the Virtual Hosts to be Inventoried

By default, the collect task inventories all the virtual hosts of a selected virtualization server. However, if you do not want to inventory all the hosts, you can limit the inventory collection to fewer virtual hosts. You can set the filters either while creating a collect task or later by modifying an existing collect task.

Note: You can set the filters only for VMware vCenter Server and Citrix XenServer.

Follow these steps:

1. Open the DSM Explorer and navigate to Computers and Users, All Computers, *AM remote agent*, Configuration, Collect Tasks.

   A list of configured and scheduled inventory collection tasks are displayed.

2. Right-click the task for which you want to set the filters and select Properties.

   The Properties dialog opens.

3. Select the host from the Virtualization Servers tab and click Properties.

   The Update Virtual Host Configuration Settings dialog opens.

4. Click the Filter tab and specify the filter at the data center, cluster, and host level for VMware vCenter Server. You can specify the filter only at the host level for Citrix XenServer.

   Note: You can use the asterisk (*) and percent (%) wild characters for all names used in the filter.

5. Click OK.

   The filters are set to the collect task. The collect task collects RVI data only from the selected hosts.

View all Virtual Machines Connected to a Virtualization Server

After the AM remote agent collects inventory, you can view the relationships between the hosts and virtual machines. Select the virtualization server under All Computers. The homepage tab on the right pane displays the managed guests in the Related Computers view under the Environment portlet.

For example, if the VMware ESX - 790-1g2-i2duo node is selected, a VMware managed guest is displayed in the Related Computers view.

Similarly, selecting the same managed VMware guest under All Computers displays its related host server, VMware ESX - 790-1g2-i2duo, in the Related Host Computers view on the home page of the asset.

Note: Only managed guests, that is, those guests with a DSM agent installed on them, are displayed in the Related computers view.

Configuration Policies

Asset Management (Agent) Policy Group

A new policy subgroup, Asset Management, is added under the Agent policy group for remote virtual inventory and the display of legacy VMware ESX virtualization inventory.

You can modify the policy parameter values by double-clicking a policy to display the Setting Properties dialog.

The Asset Management policy group contains the following policy:

RVI Legacy Inventory

Specifies whether VMware ESX virtualization inventory appears in the DSM Explorer in its legacy Release 12.8 location under All Computers and in its new location.

If set to True, legacy VMware ESX virtualization inventory is displayed in both places in the DSM Explorer tree. If False, legacy VMware ESX virtualization inventory is displayed only in the new Release 12.8 location.

The following table indicates legacy and current locations for VMware ESX inventory:

Release 12.5 Nodes	Current Nodes
Configuration	Virtualization/VMware ESX Configuration
Supported Features	Virtualization/VMware ESX Configuration/Supported Features
Virtualization/Datastores	Virtualization/VMware ESX Datastores
Virtualization/Service Console	Virtualization/VMware ESX Service Console **Note:** This node is available only for VMware ESX, not VMware ESXi.

Default: False, <locally managed>

Asset Management (Administration Console) Policy Group

A new policy subgroup, Asset Management, is added under the Administration Console policy group for specifying the Web Service URL when creating a virtual host inventory collect task.

You can modify the policy parameter values by double-clicking a policy to display the Setting Properties dialog.

The Asset Management policy group contains the following policy:

RVI Edit URL

Specifies whether the Web Service URL can be manually entered or edited in the virtual host inventory collect task. The Web Service URL field is automatically populated in the collect task, that is based on the host name regardless of the value set for this policy.

If True, the Web Service URL field is enabled and you can edit the URL.

Note: If you make subsequent changes to the Host Name field, update the Web Service URL field.

If False, you cannot edit the Web Service URL and the field is disabled.

Default: False, <locally managed>

More information:

Add Virtualization Servers (see page 185)

Queries and Reports

Queries

When creating a query in the Query Designer, you now have additional arguments for reporting on virtualization servers and virtual hosts. You can also use the following predefined queries:

- Asset is a Virtual Machine - Citrix XenServer
- Asset is a Virtual Machine - Microsoft Hyper-V
- Asset is a Virtual Machine - VMware ESX

Reports

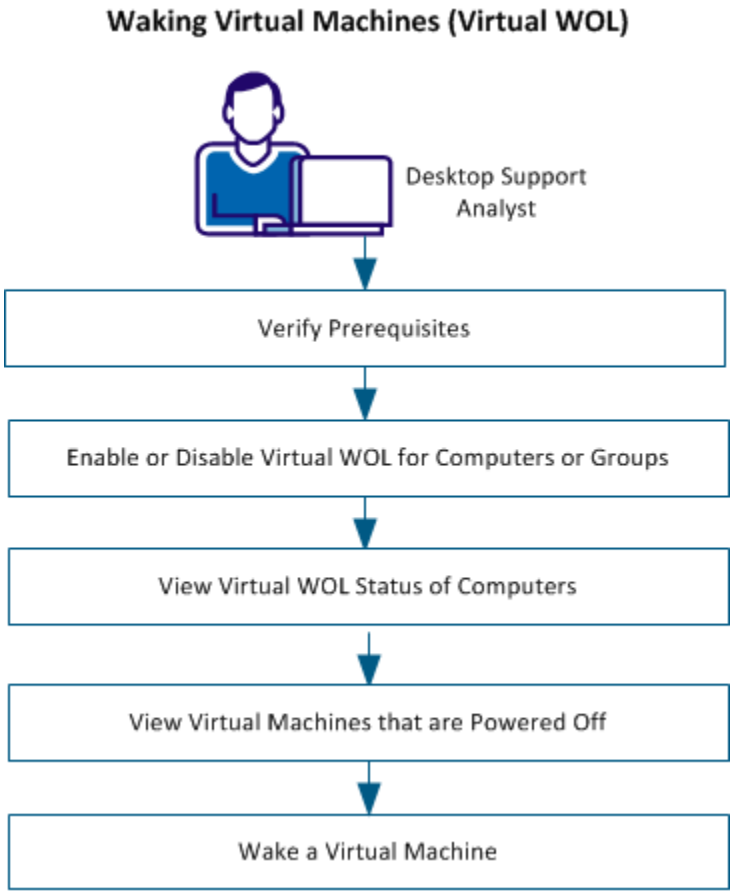
The following reports are added to DSM Reporter for the extended platform virtualization support:

- All Citrix XenServer Hosts and Citrix XenServer Virtual Machines
- All Hyper-V Hosts and Hyper-V Virtual Machines

Once a report template is run and inventory is collected, these reports list all virtualization servers and their corresponding virtual machines.

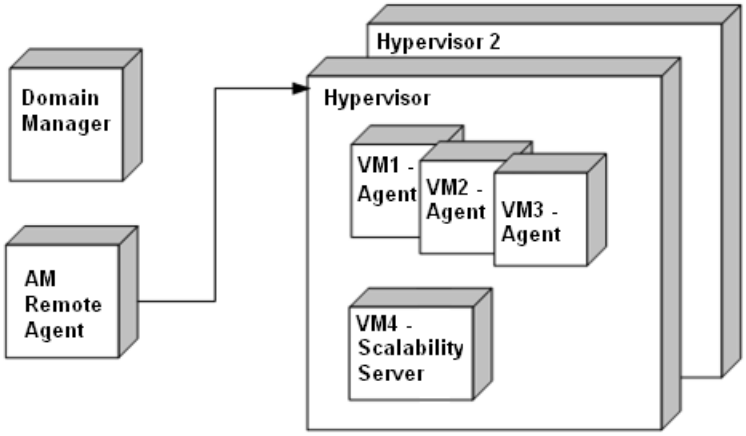
Waking Virtual Machines (Virtual WOL)

As a Desktop Support Analyst, you can power up a single or group of virtual machines that are in suspended or sleep mode from a remote computer. CA ITCM supports virtual WOL (vWOL) to wake up virtual machines.



Virtual Environment Scenario

A typical virtual environment includes a domain manager, an AM remote agent computer, and various virtualization server types containing DSM agent and scalability virtual machines as shown in the following illustration:



The DSM domain manager manages *all* computers, and the AM remote agent computer collects details about the VMs for both RVI and virtual WOL operations.

In this scenario, you can perform the virtual WOL using the following methods:

- Send a software delivery job to a virtual machine

  The job container is sent to the appropriate scalability server, which wakes up a VM using the WOL functionality of CAF API.

- Use the DSM Explorer on the domain manager to wake a particular VM

  For more information, see Wake a Virtual Machine Using DSM Explorer (see page 210).

- Use remote control to connect to a VM by waking it first
- Remote control wakes up a VM when you click Power Up on the Connection Settings, Advanced tab. Also, you can connect to a sleep mode VM. You are prompted with a message whether to wake up and connect to the VM.
- Send a request to the OSIM boot server to wake a virtual machine as part of the OS installation job

  You can configure the Wake-On-LAN option in the Setup OS Installation dialog during the activation of OS installation.

How CA ITCM Wakes a Virtual Machine

The method for waking a virtual machine differs from the method that is used to wake a physical computer. Each platform virtualization vendor provides their own SDK for WOL operations. As the AM remote agent already uses these SDKs to collect inventory, the agent can perform virtual WOLs using the new generic SendWOL method. CA ITCM uses the following process to wake virtual machines.

- When a virtual machine registers with the DSM domain manager, the domain manager does not initially know that it is a virtual machine. The WOL protocol type is changed from "nw" to "vm" only when the AM remote agent next runs and sends the required relationship information between the virtual machine and its host. This action also disables WOL on the virtual machine (the default setting for virtual machines).
- The AM remote agent gathers the following inventory information about VMs from their virtualization servers and reports this information to the DSM domain manager and MDB:

  Note: The RVI agent cannot collect certain BIOS inventory attributes from Linux virtual machines that are running on Microsoft Hyper-V and XenServer servers, Virtual WOL is not supported on such Linux virtual machines.

  - FQDN of the parent virtualization server of the VM for waking up.
  - Platform virtualization type (VMware ESXi, Microsoft Hyper-V, and so on)
  - Login credentials for the virtualization server
  - Vendor-specific identifying string for the VM

    Note: Typically, the string is a virtual GUID but it could be a system ID or serial number.
- The AM remote agent stores this information in a simple flat file named rvidb.txt under the folder CA\DSM\Agent\units\00000001\uam\rvi. This file can be accessed to add and retrieve information about a virtualization server given its URL. To avoid unlimited growth of this file, all aged virtualization servers and their respective lists of VMs are deleted automatically after one year.
- When the domain manager receives a vWOL request, it delegates the request to a scalability server. The scalability server in turn delegates it to the AM remote agent computer. The AM remote agent computer then wakes the virtual machine.

Verify Prerequisites

The following prerequisites are required for virtual WOL to work:

- Verify that the AM remote agent (RVI agent) is installed on a physical computer or on a virtual machine. Virtual WOLs are always performed from the computer that runs the AM remote agent.

  Note: The RVI agent cannot collect certain BIOS inventory attributes from Linux virtual machines that are running on Microsoft Hyper-V and XenServer servers, Virtual WOL is not supported on such Linux virtual machines.

- Verify that the scalability server is able to communicate to the AM remote agent computer. Virtual WOL requests are routed from the scalability server to the AM remote agent.
- Verify that the AM remote agent has discovered and collected virtualization data from the VMs that you want to wake. The AM remote agent collects virtualization data through the inventory collection task. Until the inventory collection happens, a WOL operation assumes that the target computer is physical and not virtual. The inventory collection takes time depending on the number of VMs to be discovered and registered.

Note: Virtual WOL is not supported on guests running on the free version of VMware ESXi as the free version restricts API access to the host.

Enable or Disable Virtual WOL for Computers or Groups

Virtual WOL is enabled by default on all the virtual machines. When a virtualization server hosts numerous VMs, but lacks the resources to run them at the same time, you can disable virtual WOL on its VMs. You can enable it back when you want to power up computers remotely.

Follow these steps:

1. Right-click the computer or group and select Power up, Disable.

   The virtual WOL is disabled on the computer. The computer or group is added to the *do not wake* group.

2. Right-click the computer or group and verify that the Power up, Power up is dimmed.

   This action verifies that the virtual WOL is disabled.

View Virtual WOL Status of Computers

You can view the virtual WOL status of a computer to know whether you can power it up using CA ITCM. You can view the status in one of the following nodes in DSM Explorer:

- Power on enabled under the Agent Status portlet on the homepage tab of the virtual machine
- *Can be powered up* option on the Computer Properties dialog

  Note: This option is available only for DSM components that can perform a bulk wakeup, such as software delivery and DTS. The *Can be powered up* option is not applicable for remote control.
- Power Up option under Computers, General, Computer in the Add Argument dialog in Query Designer for computer-based queries

View Virtual Machines that are Powered Off

You must know whether a virtual machine is powered on or off before you wake it up. You only want to wake up those that are powered off. To view the power on status of virtual machines on a particular virtualization server, navigate to All Computers, *Virtualization server*, Inventory, Virtualization, *Server* Virtual Machines. View the State column on the right pane.

Wake a Virtual Machine Using the CLI

You can wake a virtual machine using the CLI. For example, when you want to wake multiple virtual machines using a batch script. The following command wakes a virtual machine:

`caf sendwol [user` *`username`* `password` *`password`*`] vm vGUID` *`guid`* `vsvr` *`url`* `rviagent` *`fqdn`*

username and password

Specifies the credentials of an administrative user on the AM remote agent computer. When you use the same computer credential that you are executing the CLI, you can skip the parameters.

vm

Specifies that virtual WOL must be performed.

vGUID

Specifies the ID of the virtual machine as known to the host. The ID can be a virtual GUID, system ID, or serial number depending on the platform virtualization server type.

vsvr

Specifies the URL of the virtual server host of the current virtual machine.

rviagent

Specifies the FQDN of the AM remote agent that discovered the virtual machine.

Virtual WOL Example

This example performs virtual WOL on a virtual machine:

```
caf sendwol user root password secret vm vsvr https://virt.acme.com/sdk vguid 123456
rviagent rvi3.acme.com
```

Note: For more detailed information about CAF commands, type caf <command> /? at the command prompt.

Wake a Virtual Machine Using DSM Explorer

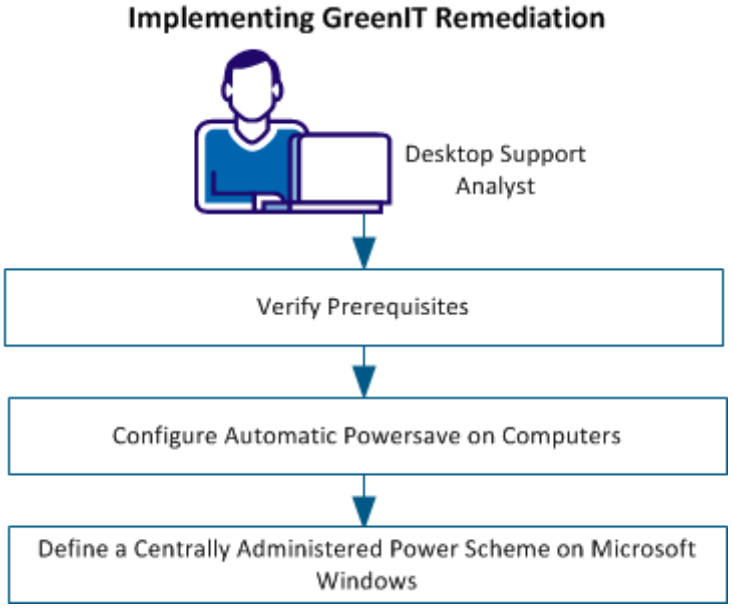
You can wake a virtual machine from the DSM Explorer. Right-click the virtual machines that you want to wake and select the Power up option. This action sends the virtual WOL request for the computer. Connect to the virtual machine to see whether the virtual machine is woken up.

GreenIT Remediation

Implementing GreenIT Remediation

As a desktop support analyst, you can configure the power settings and power schemes in your organization to ensure optimal power consumption. Using CA ITCM to define and push a configuration policy to computers to automatically power down during a specified time (for example, after work hours, over the weekend, or on a holiday. Also, you can monitor the current power scheme and define a standard Windows power scheme implementation across your organization.

The following illustration shows the tasks that you complete to implement GreenIT remediation:



Perform the following tasks to implement GreenIT remediation:

- Verify Prerequisites (see page 212)
- Configure Automatic Powersave on Computers (see page 212)
- Define a Centrally Administered Power Scheme on Microsoft Windows (see page 214)

Verify Prerequisites

Review the following prerequisites before you implement GreenIT remediation:

- Verify that a DSM Agent is installed on the target computers. You can install a DSM agent as part of CA ITCM, CA IT Inventory Manager, or ecoDesktop.
- Verify that you have a license for ecoDesktop.

Important! To use the GreenIT Remediation features, you need a separate license for ecoDesktop, available at an additional cost.

Configure Automatic Powersave on Computers

You can configure automatic Powersave on specific computers or a group of computers. Automatic Powersave helps save energy by powering down the computers automatically during specified time such as after work hours, weekends, and holidays.

Follow these steps:

1. Open DSM Explorer and navigate to Control Panel, Configuration, *configuration policy*.

   Configuration policy name

   Specifies the name of the policy that is applied to the computers or group for which you are defining the automatic Powersave settings.

2. Right-click the configuration policy and click Unseal.
3. Navigate to DSM, Common Components, CAF, GreenIT Powersave.
4. Set GreenIT: Powersave to Enabled.
5. Navigate to DSM, Common Components, CAF, Shutdown, Power Management Calendar, Time Ranges.

   The Power Management Calendar dialog appears.

6. Specify the required values for the given parameters.

   Hours

   Specifies the hours for the automatic Powersave action. Unlike with the Software Delivery shutdown feature, for automatic Powersave, only the start time of the time range is used to schedule a Powersave job. This means that the start time of the time range is the time when the automatic Powersave action is triggered. Even if specified, the end time of the time range does not affect the GreenIT Automatic Powersave feature.

   In the Type field, select one of the following values:

   Normal Weekdays

   Specifies the weekdays on which the Powersave action is executed.

   Special Dates

   Specifies dates in the *yyyy-mm-dd* format. Zeros in the date indicate that the setting is applicable for every year, month, or day.
   Example
   If the specified date is 2012-00-08, the job is executed on eighth day of every month of 2012. If the specified date is 00-08-00, the job is executed on all days of every August.

   Date Ranges

   Includes each day between the start date and end date.

   Example
   If the specified date range is 2012-05-01 to 2012-05-11, the job is executed on all days from 1st to 11th May 2012.

   The criterion for the automatic Powersave action is defined.

7. Copy this configuration policy to the target computers or computer groups.

At the scheduled time, CA ITCM displays a start timer message for the automatic shutdown, hibernate, sleep, or turnoff. Based on users actions, such as Shutdown now, Hibernate now, Sleep now, or Turn off Displaynow, Defer, or Cancel, computers are powered down. If there is no response from the user, the scheduled action is performed automatically at the end-of-countdown timeout.

Example:

If you specify the automatic Powersave time from 6.00pm to 9.00am every day. The users that are logged on to the computers are alerted at 6.00pm about the scheduled powerdown for the computer. They can click any one of the following options:

Shutdown now or Hibernate now or Sleep now or Turn off Display now

Applies the Powersave action immediately. On a computer resume from the Powersave state (such as Shutdown, Hibernate, Turn off Display, or Sleep), Powersave is initiated again according to the next defined Powersave schedule.

Defer

Defers the Powersave action for some time. You can specify the time to defer and the maximum number of times users can click Defer, after which the Powersave action is applied.

Cancel

Cancels the Powersave action. The timer and pop-up dialog are canceled. Powersave is initiated again according to the next defined Powersave schedule.

If there is no response from the user at the end of a countdown timeout, the Powersave action is applied automatically.

Define a Centrally Administered Power Scheme on Microsoft Windows

You can define a standard power scheme for your organization to ensure that the overall power consumption is minimal. Define different power schemes and assign the schemes to different groups of users or computers.

Follow these steps:

1. Navigate to Jobs, Asset Jobs, right-click Asset Jobs.

   The New Job wizard opens.

2. Select GreenIT Remediation as the job type. Click Next.

3. Enter a name and description for the new job. Click Next.

   Note: You cannot modify the name of a scheme while creating a Green IT Remediation job. The scheme takes the same name as the GreenIT Asset Management job. To modify the scheme name, modify the name of the GreenIT Remediation job.

4. Specify values for the following parameters:

   Monitor Time-out (minutes)

   Specifies the computer idle timeout after which the monitor is powered down.

   Hard Disk Time-out (minutes)

   Specifies the computer idle timeout after which the hard disk is powered down.

   Standby/Sleep Time-out (minutes)

   Specifies the computer idle timeout after which a computer on a standby or sleep mode is powered down.

   Hibernate Time-out (minutes)

   Specifies the computer idle timeout after which a computer on the hibernate mode is powered down.

   Wake Armed

   Specifies whether to enable the wake armed devices that are configured to wake the computer from any sleep state.

   Hibernate

   Specifies the computer idle timeout after which a computer is set to hibernate.

5. Click the Set Scheduling button and specify the required settings to change the current scheduling options.

6. Click Finish.

   The Green IT Remediation job category appears as a subfolder under the Jobs folder. This folder lists all the Green IT Remediation jobs defined in the domain. The Green IT Remediation jobs are displayed with a new icon.

7. Link or drag-and-drop this GreenIT Remediation job to the target computers or computer groups.

The new Green IT Remediation Asset Job distributes a power scheme for Green IT Remediation. The power scheme that is thus distributed to computers is applied to all users registered on the computer using a query group. This distribution standardizes the power settings across the organization and ensures optimal power consumption.

Inventory Updates for GreenIT Remediation

The following table lists the new fields added to different nodes of the inventory:

Inventory Node	Field	Description
General Inventory, Power Settings, Power Policy	AC - Processor Throttle DC - Processor Throttle	CPU throttling or processor throttling is the process when the CPU attempts to prevent damage due to overheating. If the CPU temperature exceeds the specified limits, the system throttles down the CPU for cool down. This process takes place when the computer is idle or when no critical tasks are executed on it. The CPU switches to a lower frequency for optimal consumption of energy and to ensure minimal noise from the fans.
General Inventory, Power Settings, Power Usage Times	Total Log Time Off Time Off Time % On Time On Time % Suspend Time Suspend Time % Verify Time	These fields comprise data that various Inventory Detection and Collection modules collect. **Total Log Time:** Specifies the total time duration over which analysis is performed. It is allowed for the last 30 days or the duration since the last CA ITCM installation, whichever is less. **Suspend Time:** Includes the sleep and hibernate times. **Verify Time:** Specifies the actual time that is considered for the analysis. It is the sum of On Time, Off time, and Suspend time.

Inventory Node	Field	Description
General Inventory, Power Settings, System Sleep States	*System Sleep States*	CA ITCM supports multiple sleep modes that correspond to the power states defined in the Advanced Configuration and Power Interface (ACPI) specification.
General Inventory, Power Settings, System Wake Armed Devices	*System Wake Armed Devices*	Devices that are currently configured by the user to wake a computer from any sleep mode.
General Inventory, Power Settings, System Wake Programmable Devices	*System Wake Programmable Devices*	Devices that a user can configure to wake a computer from a sleep mode.

Impact of GreenIT Remediation on Software Delivery

The Power Management Calendar in Software Delivery enables an administrator to define a time window when specified hardware is set to powerdown. GreenIT Powersave utilizes the same calendar for defining the Powersave settings. When both the features are enabled, the time schedules that are configured for one feature are applied to the other feature.

When a GreenIT Powersave action such as Suspend or Shutdown is initiated, the software delivery agent does not run as long as the Powersave dialog appears to the user. After Powersave is complete, the software delivery agent uses the WOL feature to start the computer and run software delivery jobs.

Note: Software delivery takes place even while the Turn off display dialog appears to the user or when the Turn off Display operation is in progress.

When the GreenIT Powersave job initiates during the software delivery, automatic Powersave actions wait until the completion of software delivery or until the timeout expiry. This behavior does not hold true for powesave action Turn off display.

Use the following configuration parameters to define whether CA ITCM-initiated Powersave and user-initiated power actions are allowed when a software delivery job is in progress:

CAF: Allow Powersave action

Specifies whether to allow any CA ITCM-initiated Powersave action (excluding turn off display) when a software delivery job is in progress. This parameter does not control the actions that a user initiates from the Start menu. This parameter also specifies whether to prevent sleep or hibernate actions that are initiated by Windows power schema while a software delivery job is in progress.

Reboot: Allow application logoff and reboot

Specifies whether to allow any logoff, reboot, or shutdown actions (including user-initiated actions but excluding sleep, hibernate and turn off display actions) when a software delivery job is in progress.

Both the parameters are set to False by default. In this state, these parameters do not allow either CA ITCM-initiated Powersave actions (such as shutdown, hibernate, or sleep) or a user-initiated shutdown action, when a software delivery job is in progress. We recommend that you set both parameters to the same value to ensure consistent behavior. You can set both the parameters to True and software delivery jobs that are either in progress or are waiting for an input from the user are aborted, and a reboot, logoff, or shutdown is performed.

You can configure the postponement timeout for CA ITCM-initiated Powersave actions by using the following configuration parameter available at DSM, Common Components, CAF, General:

CAF: Maximum power action postponement timeout

Specifies the maximum time value for which CAF waits for the plug-in to complete before implementing a Powersave action.
The default value is set to 7200 sec. If set to 0, postponement timeout is not allowed.

Alternatively, you can postpone CA ITCM-initiated Powersave actions when a software delivery job is in progress.

Follow these steps:

1. Navigate to DSM, Common Components, CAF, General, Postpone CAF power action.
2. Set the CAF Plugin: Software Delivery Agent parameter to Enabled.

Note: For Asset Management, you can set the CAF Plugin: Asset Management Agent parameter to Enabled. Enabling this parameter postpones the CA ITCM-initiated Powersave actions when an asset management job is in progress.

Chapter 5: Customizing Asset Management

You can customize asset jobs, queries, policies, and so on to suit your business needs and add more control over your environment.

This section contains the following topics:

Management Information Format (.MIF) Files

The Management Information Format (.MIF) files are used to describe a hardware or software component data. MIF files are used by asset management system to report the system configuration information. In asset management each scheduled inventory, templates, configuration files and external asset definitions consists of a mif file.

The .MIF file is a text file, which is installed and presented to the Service Layer for inclusion in the MIF database. You can modify a .MIF file with a text editor, although component providers are encouraged to automate this process. The Distributed Management Task Force (DMTF) has designed the specifications for .MIF files in the Desktop Management Interface (DMI) specification.

Asset management uses the .MIF files to register and store information locally on the agent working directories (see page 29).

Usage of MIF Files in Asset Management

Asset management uses .MIF files to register and store information in the agent working directories. These .MIF files are built according to DMI specification. In asset management, certain extensions to the .MIF files are made to be able to prompt the user to fill in information for these .MIF files. An example is to let the user choose certain values. Every .MIF file placed in the local computer and local user directory is processed by the agent. The information in all these .MIF files is sent to the scalability server (and to the domain database by the engine).

Asset management uses the MIF files in the following cases:

- To collect template inventories (Windows agents only)
- To collect external asset information (External Asset Recognition)

More information:

Agent Working Directory (see page 29)

How MIF files Are used in Template Inventory

Template inventory modules let you collect customized inventory information from a MIF file; information which is not collected by any of the inventory detection modules. Asset management uses the following process to store and retrieve the information from the MIF files. By understanding these actions, you can choose how to best configure the MIF files as per your requirements:

- The asset management agent creates a MIF file for each template inventory when executing a template inventory collect task to store the collected inventory information.
- The agent software processes every MIF file placed in the local computer or user directory. Asset management uses a compression algorithm to reduce the size of these .MIF files and thus speeds up the file transmission. The compressed files have the same file name as the MIF files but have the extension .MNV and are stored in the BAK directory of the computer or user.

  Note: The agent processes only the .MIF files that fulfill the DMI specification.
- If the agent runs for the first time or if the BAK directory is deleted, the agent sends the full inventory to the scalability server. You can also manually send a notification for collecting full inventory. In all the other cases, only differential value is sent. The BAK directory contains the information with which the agent compares the current inventory values with the previous scan values.
- The differences between current and previously collected inventory values are sent to the scalability server which is available under ...\ServerDB\SECTOR\COLLECT.
- The engine collects this information from the collect area and updates the domain database accordingly.

More information:

Create, Configure, and Schedule Template Inventory (see page 171)
Inventory Template Modules (see page 122)
Recollect Inventory (see page 158)

How MIF Files Are Used in External Asset Recognition

Asset management requires the following objects for recognizing external assets:

- An external asset definition
- A MIF file that has the asset information.

Asset management can recognize the external assets in your enterprise. An external asset is any asset other than computers and users for example, a printer. Understanding this process, helps you in configuring the MIF files for recognizing the external assets.

- The external asset definition contains a recognition item to identify the appropriate MIF file that resides in the scalability server's collect area.
- The Engine collects each MIF file in the collect area (...\ServerDB\SECTOR\COLLECT on windows and or / ServerDB/SECTOR/COLLECT on Linux).
- Checks whether the recognition item in any of the external asset definition matches the values in the MIF file
  - If a match is found, the engine collects the information configured in the external asset definition, creates a new external asset with the given name and deletes the file from the collect area.
  - If not, the engine just deletes the MIF file.

    Note: On the Linux or UNIX scalability server, the MIF file must have the file extension in uppercase, for example, my_printer.MIF; otherwise, the MIF file will not be processed.
- Creates the All External Assets node if it does not exist and creates a new external asset entry under it.

Create External Asset Definition

External asset definitions define how to recognize an external asset. Once defined, the engine can recognize the asset and collect inventory from the asset.

To configure the external assets

1. Navigate to the Domain, Control Panel, Configuration, External Asset Recognition folder.

   The existing external asset definitions are displayed.

2. Click New.

   The Create new External Asset Class Definition dialog appears.

   Note: For more details about the fields in this dialog, press F1.

3. Click Open MIF Browser.

   The Select MIF Item dialog appears with the items in the selected MIF file.

4. Drag the relevant items to the Create new External Asset Class Definition dialog and click OK.

   The external asset definition is created under the External Asset Recognition folder.

5. Copy the MIF file that has the external asset information to the collect area (...\ServerDB\SECTOR\COLLECT on Windows or / ServerDB/SECTOR/COLLECT on Linux).

   Note: On Linux or UNIX scalability server, the MIF file must have the file extension in uppercase, for example, my_printer.MIF; otherwise, the MIF file will not be processed.

   The engine collects this file and processes it. When it finds the recognition item in the MIF file, it creates a new external asset and places it under the All External Assets folder.

Jobs

A job directs agents or engines to perform specific actions. You can, for example, use jobs to run a command or script, collect the inventory information, and pop a message on the asset. You can schedule these jobs to run at a specific time. Asset management categorizes jobs into the following:

- Asset Jobs; those that are performed by the agent on the assets. Hence, an asset job is executed only if it is linked to an asset.
- Engine Tasks; those that are performed by the engine. Hence, an engine job is executed only if it is linked to an engine.

Asset Jobs

An asset job directs agents to perform specific actions on a computer or when a user logs in. You can, for example, use asset jobs to run a script or command, and display a message.

Jobs can be created at the asset, group, and domain level. Jobs created at the asset level are linked to the respective assets, and those at the group level are linked to all the member assets in the group. At the domain level, you can view all the jobs in the domain, create new jobs, and link it any group or asset. You can create and link any number of jobs to a particular asset or group.

Append the following path to the Domain name in the address bar of the DSM Explorer window. In the following paths,

- *Computer* refers to the name of the computer in the domain.
- *User* refers to the name of the user in the domain.

Asset/User Level

Assets–Domain/Computers and Users/All Computers/*Computer*/Jobs/Asset Jobs

Users–Domain/Computers and Users/All User Accounts/*user*/Jobs/Asset Jobs

Group Level

All Computers Group–Domain/Computers and Users/All Computers/{Group Details}/Jobs/Asset Jobs
All Users Group–Domain/Computers and Users/All User Accounts/{Group Details}/Jobs/Asset Jobs
User Defined Groups–Domain/Computers and Users/*Group Name*/{Group Details}/Jobs/Asset Jobs

Domain Level

Domain/Jobs

An existing job can be linked to any group or computer. Hence, the folder under which a job is created is irrelevant.

Navigate to the asset jobs folder to see the available jobs and their scheduling options.

Note: For more information about job types and creating and editing jobs, see the Asset Management section of the *DSM Explorer Help.*

More information:

Access the Asset Jobs Folder in the DSM Explorer (see page 224)

Access the Asset Jobs Folder in the DSM Explorer

You can create, configure, link, and schedule the assets jobs in the Asset Jobs folder.

To access the Asset Jobs folder in the DSM Explorer

1. Navigate to the following paths in the DSM Explorer depending on the level at which you want to view or configure the asset jobs:

   Group Level

   Computer Group

   Domain, Computers and Users, All Computers, Group Details.

   User Account Group

   Domain, Computers and Users, All User Accounts, Group Details.

   User Created Group

   Domain, Computers and Users, *Group*, Group Details.

   Computer Level

   Domain, Computers and Users, All Computers, *Computer*

   User Level

   Domain, Computers and Users, All User Accounts, *UserAccount.*

   Domain Level

   Domain

2. From the above paths, navigate to the Jobs, Asset Jobs folder.

   Jobs created at the group level are automatically linked to the member assets of the group. You can, however, disable the execution of such jobs at the asset level. Jobs linked to user are executed when the user logs in.

Note: You can link the asset jobs created at all these levels to any managed asset or group.

More information:

Jobs (see page 222)
Enable or Disable Jobs (see page 231)

Job Types

You must select a job type when you create a new job. Each type of job requires different parameters for executing the job. The following asset jobs are available:

- Message Job (Windows only)
- Command Job
- Synchronization Job (Windows only)
- External Utility Job
- Secure Configuration Files Job (Windows only)
- Script Job

Message Job

Message jobs can display a message on the asset or when a user logs in. You can see the job status to know when the message was displayed.

When a message job is assigned to a user and the same user logs on to more than one session, the message job can be displayed in any one of the user sessions.

Note: This job is applicable only for Windows agents.

More information:

View Job Status (see page 230)

Command Job

Command Jobs enable you to execute a command file (for example, a batch file or a UNIX shell script) on the selected asset or group. All normal native command-language commands can be used in this file.

Synchronization Job

Use the Directory Synchronization Job to synchronize two directories on an asset. Synchronization involves comparing the contents of the target directory and the source directory, and making the content of the former identical to the latter.

Note: This job can be linked only to the computer assets and is applicable only for Windows agents.

In Step 3 of the New Job wizard, specify the directories:

Source

Indicates the directory to be compared against (the image).

Target

Indicates the directory to be examined for differences (reflection of the image).

Note: The use of UNC path is supported. Syntax: \\servername\sharename

In Step 4 of the New Job wizard, specify the following:

- Whether subdirectories should also be examined for differences.
- Whether files on the Target that are nonexistent on the Source should be deleted.
- Whether attributes should be ignored when comparing, to prevent making Target file attributes equal to Source file attributes.
- Whether attributes should be ignored when copying, in which case the job will try to overwrite Target files that are write protected.
- Whether to cancel the job, when the size difference between the two directories exceeds the specified size in KB. This parameter is provided to prevent accidental copying of a large number of KB over the network.

Example: Printer File Synchronization

This is an example of how to update the printer files of users that are on the local hard disk of the users. To make it easier, you need to update only, say, N:\PRINTERS with the latest printer files. You can then use directory synchronization to equate the directories C:\WINDOWS\PRINTERS and N:\PRINTERS. The printer files located on the hard disk are now made equal to the printer files on the network drive N. The directory is thereafter examined for differences each time the computer starts on the network.

```
Source: N:\PRINTERS

Target: C:\WINDOWS\PRINTERS

Enabled: Include Subdirectories; Delete unknown files; Ignore file attributes on
compare; Ignore file attributes on copy.
```

Example: Portable Computer File Update

If there are portable computers on your network, you can synchronize a subdirectory on the portable computer's hard disk with a personal network drive on the server. This permits automatic transfer to the network of portable computer files that were created while the portable computer was not connected to the network.

The Directory Synchronization Job ensures absolute conformity between two directories. It ensures that the attributes are also equal. Files on the target that are not found on the source will be deleted, if so specified. Directories not found on the target will be created.

In some cases, you do not want all files and subdirectories in a directory to be synchronized. (For example, in an Agent Software directory, you do not want the PROTOCOL.INI and the HOST files to be equal throughout your network.) In this case, you should specify the files and directories, which should not be synchronized.

Example: File Exclusion

This is an example of how to exclude files from synchronization. Assume that the local C:\PW directory is synchronized with a default network directory F:\PWDEFAUL. As a system manager, you know that the PROTOCOL.ini file and all the *.TPL files of that directory must be excluded from synchronization, because those files contain unique values for each individual computer. (The directory BAK is also excluded).

```
Source: F:\PWDEFAUL

Target: C:\PW

Enabled: Include Subdirectories; Delete Files on Target if Unknown on Source.

Specified: Files not to be synchronized: *.TPL, PROTOCOL.INI, (DIR) BAK
```

External Utility Job

External utility jobs execute an external utility. This type of job can be used to execute a script written for a management tool other than asset management.

Script Job

Script jobs can execute scripts on assets. You can either write the script directly or open a saved script from a file.

Secure Configuration Files Job

A Secure Configuration Files Job ensures that the specified system or configuration file is always equal to the content (secure file) specified in this job.

When the agent runs, it checks whether the configuration file is modified. If yes, it copies the secure file to the asset and overwrites the modified file.

Note: This job is applicable only for Windows agents. On UNIX or Linux, you can create a DSM script to copy files onto a share.

Change Job Execution Order

Jobs are executed in the same order in which it was created. However, you can change this order if the execution of one job depends on the execution of another.

Note: You can do this only at the group and asset level.

To change the job execution order

1. Navigate to the Asset Jobs folder.
2. Right-click the job for which you want to change the order and select Reorder.

   A sub menu appears.
3. Select any of the following menu items:

   Move Up

   Moves the job one level up. If you want to move ahead more than one job, click this menu item till you reach the position.

   Move Down

   Moves the job one level down. If you want to move down more than one job, click this menu item till you reach the position.

   Move First

   Moves the job to the first position.

   Most Last

   Moves the job to the last position.

   Set Position

   Moves the job to the specified position. You can use this option for example, when you want to move ahead or below of more than 5 jobs.

   Clear Job Order

   Clears the existing order of the jobs and reorders them alphabetically.

   The jobs are reordered based on the selected menu item.

Scheduling Jobs

Jobs can be scheduled to execute once, a specified number of times, or continuously. When you set up a job, for example, you can schedule the action to execute in different ways:

- Repeatedly
- At a specific date and time, or after a period of time
- Based on day-of-the-week dependencies
- Based on the execution of another job, or the existence of a file, or an end-user decision to execute the action now or later

By default, jobs are scheduled to run always. This means, the jobs are executed whenever the asset management agent starts on the asset. In other words, the execution of asset jobs is dependent on the agent run. For example, if the asset management agent is scheduled to run once a day and the asset job is scheduled to run only on Thursday, the job will be executed when the agent starts on Thursday.

You can specify the scheduling options in two ways:

- At the time of creating the job, the New Job Wizard has a Set Scheduling button that lets you specify the scheduling options.
- After creating the job, you can right-click the job and select Scheduling from the context menu.

In both the cases, the Scheduling Options dialog appears where you can define the following:

Scheduling

Specifies the frequency of a job.

Conditions

Specifies various conditions for the job. You can schedule the job to run within a time and date range or on specific days in the week.

Dependency

Specifies that the execution of a job is dependent on the successful completion of another job or the existence of another file. You can specify the maximum number of simultaneous executions of an action per scalability server and thus distribute the network load over a period of time using the job concurrency limit.

Pre-Job

Specifies a condition that should be checked before running the job:

- Enable prompts for the end-users allowing them to cancel execution of a job.
- Enable a force execution after a certain number of executions.
- Map a network drive, which may be necessary for the job to run.

Post-Job

Specifies the actions to be performed after the job is completed. You can schedule a restart of the computer or the automatic deletion of a job after completion.

Miscellaneous

Specifies that the job can be run without any user interaction.

Note: For more details about Scheduling and Scheduling Options dialog, see the *DSM Explorer Help*.

Reinitializing Job

Once you make any modifications to a job, the Reinitialize Job message box appears. You can choose any of the following to reinitialize the status of the job to Waiting:

Yes

Reinitializes the status of the job on all the assets connected to the job.

No

Reinitializes the status of the job on selected asset or group.

View Job Status

The job status lets you know whether the job was successfully completed or is waiting or completed with errors.

To view the job status

1. Navigate to Asset Jobs (see page 224).

   The right pane displays all the asset jobs linked to the selected asset or group.

2. Right-click the job and select Status.

   The Action Status dialog appears displaying more details about the status such as the status text and the date and time of last execution of the job and so on.

Enable or Disable Jobs

You can temporarily disable the jobs that you do not want to execute on an asset or group. The jobs disabled at the group level, can be enabled only at the group level. Similarly, jobs disabled at the asset level can be enabled only at the asset level.

To enable or disable jobs

1. Navigate to Asset Jobs (see page 224).

   The right pane displays all the asset jobs linked to the selected asset or group.

2. Right-click the job and select Disable or Enable.

   The job is disabled or enabled and the status changes accordingly.

Queries

Querying is a crucial advantage of using a database. You can query the database to search for assets that meets the specified criteria. You can use a query for creating dynamic groups. The members of the dynamic group change according to the results of the query. You can also create a policy based on a query and apply the policy on the resulting objects. Asset management also provides predefined queries that perform basic and advanced querying of the database. You can use the Query Designer tool to create complex queries easily.

Create a New Query

You can create a query when you need to get a list of assets that meet certain criteria. You can use these queries to create dynamic groups and query-based policies.

To create a new query

1. Right-click Queries in the DSM Explorer and select New from the context menu.

   The Select Target dialog appears.

2. Select the object on which you want to create the query and click OK.

   The Query Designer dialog appears.

3. Insert the arguments required to create the new query and click OK.

   The query is created with the given name and displayed under the appropriate folder in the Queries folder.

Example: Create a Query to Display All the Assets That Have Completed the CA eTrust Virus Scanner - Force Signature Update Job Successfully

1. Right-click Queries and select New from the context menu.

   The Select Target dialog appears.

2. Select Computers and click OK.

   The Query Designer dialog appears.

3. Click Jobs, Asset Jobs

   The Select Field dialog appears.

4. Select Job Status, Current Status.

5. In the Job field, select CA eTrust Virus Scanner - Force Signature Update, and in the Status field, click OK.

6. Click OK, and enter a unique name for the query.

   The new query is created under the Queries, Computers folder.

7. Select the query, and click Quick Preview.

   All the assets that have successfully completed the CA eTrust Virus Scanner - Force Signature Update job appear.

Query Designer

The query designer gives you an easy way to query the database. You do not need to know the database fields if you are using this tool.

The Query Designer tool lets you create or modify a query. Using this tool, you can query information about computer and users, jobs, policies, software, inventory and external directories. You can also query the database fields and create advanced queries. The left pane of the Query Designer dialog lists these items: Insert Argument with all the available arguments, Boolean Operators, Remove Argument, a preview function and a view SQL statement option.

Note: Although the user can now be registered with the Asset Collector, as can an associated trust and origin, the Query Designer does not allow user queries to be created with trust level and origin as conditions.

The right pane of the Query Designer dialog shows the arguments as you add them to the new query.

Insert Argument

General Information

Lists discovered hardware and software properties.

Inventory

Lists discovered inventory.

Software

Queries Software Packages, Discovered Software, Software Usage and File Contents.

Jobs

Queries Asset Jobs and Collect Tasks.

Policies

Queries the Query Based and Event Based policies.

Directory Query

Browses an organizational asset to query any member, or select any user to query its direct reports.

Note: To create a directory query, a directory object must be available under Control Panel, Directory Integration, Configured Directories folder.

Advanced Argument

Queries the asset management tables in the database for a specific query. Select the arguments in the Add Advanced Argument dialog.

Link Query

Opens the Select Query dialog where you can link an existing query and add the query as an argument.

Boolean Operators

Specifies the following operators that Query Designer uses to test for the validity of some condition and return a Boolean value of TRUE or FALSE.

AND

Combines two Boolean expressions and returns TRUE when both of the expressions are TRUE.

OR

Combines two conditions and returns TRUE when either expression is TRUE. When more than one logical operator is used in a statement, OR operators are evaluated after AND operators.

NOT

Reverses the value of any Boolean expression.

( )

Changes the order of evaluation.

Remove Argument

Removes the argument selected on the right pane from the Query Designer.

Preview

Runs the query and opens the Query Preview dialog to display the query result.

View SQL

Shows the SQL Query statement in the SQL View dialog. You cannot modify the SQL statement on this dialog.

Run a Query

After a query is designed, you can run the query to view the results.

To run a query

1. Select the query you want to execute.
2. Click Run Query in the Tasks section.

   The query is executed and the results are displayed in the Query Preview dialog.

Note: The query execution times out after every 5 minutes with a message that allows you to either continue or cancel the query execution. At any time during the execution, you can press the Esc key to cancel the execution.

Note: Queries submitted to the engine run automatically.

Preview a Query

You can preview the query results even while you are designing the query in the Query Designer dialog. This helps you to know if the query is producing the desired output. You can also view the SQL version of the query by clicking the View SQL icon.

To preview the query results in the Query Designer dialog, create a new query in the Query Designer dialog and click Preview in the left pane. The Query Preview dialog appears showing the results of the query.

Note: The query execution times out after every 5 minutes with a message that allows you to either continue or cancel the query execution. At any time during the execution, you can press the Esc key to cancel the execution.

If you want to view the results of a query that has been applied to a dynamic group or a policy, or if you want to query your domain for the latest information, you must complete the following:

- Submit the query to an engine.
- Ensure that the engine has processed the query.

Important! The Query Designer gives you both flexibility and responsibility. An erroneous query can cause the engine to crash. Always test the query using the Preview option before applying it to a dynamic group or a policy!

Save a Query Result

You can save the resulting records of a query for future reference.

To save a query result

1. Right-click the query and select Run Query.

   The Query Preview dialog appears with the result.

2. Click Save and specify the name of the query result.

   The query result is saved with the given name and is displayed under the selected query.

Delete a Query

You can delete a query that you no longer use.

To delete an existing query

1. Right-click the query and select Delete from the context menu.

   A confirmation dialog appears.

2. Click Yes.

   The deletion is confirmed and the query is deleted from the database.

Import or Export Query Definition

You can import a saved query and execute it in the DSM Explorer. You must save the query in a .cmsobj file. To import a query definition, navigate to the Queries folder, right-click the folder and select Import Definition.

To export a query created in the DSM Explorer to a .cmsobj file, navigate to the Queries folder and right-click the query you want to export and select Export Definition.

Note: Exporting queries in .qry format is not a valid option in CA ITCM r12.5 and above. However, importing query definitions in the old .qry format is still supported. One way to convert from .qry to .cmsobj format is to import queries in the old format and export them in .cmsobj format. A batch script would be useful if you have many query definitions to convert.

Policies

Policies help the administrators to focus on critical data when certain threshold values are exceeded or events occur. You can use policies to set alarms or warning and specify the actions to be taken when a policy is violated. For example, you can create a policy to raise an alarm when a duplicate network address found in the domain.

The Asset Management Policy System is based on user-defined queries and on predefined events.

Query Based Policies

Includes the policies based on a query that specifies which criteria must be met to invoke the policy. As the query checks only the delta values sent by agents, continuous querying of the database is not necessary. The query based policies are available under the Policies, Query Based folder.

Event Based policies

Includes the policies that are invoked when a specific event is triggered. Asset management provides a set of predefined event based policy categories. You cannot create or delete the policies under these categories. However, you can modify these policies for example, to add additional actions. The event based policies are available under the Policies, Event Based folder.

Policy Definition

You can define the policies in the Policy Designer dialog. When you define the policy, consider the following factors:

- Linked Query
- Actions to perform
- Event that invokes the policy
- Header and description of policy

Create a Query Based Policy

Query based policies are evaluated whenever the associated query is run.

To create a query based policy

1. Navigate to the Policies folder.
2. Click New in the Tasks section.

   The Policy Designer dialog appears.
3. Specify the necessary details in the following sections:

   General

   Specifies general information about the policy such as name, severity, and the dependant query.

   Evaluation

   Specifies the options for evaluating the policy. You can either evaluate the policy when Collect task processing is done or designate the engine for policy evaluation at the specified interval.

   Service Desk

   Enables the CA Service Desk Manager integration. You can select the problem types and choose to a raise service ticket whenever the policy is violated.

   Note: This section is enabled only if you have set the Service Desk integration to True in the Default Computer Policy. By default, it is set to False.

   Add Action

   Specifies the action type to be triggered by the policy.

Note: For more details about the Policy Designer dialog, see the *DSM Explorer Help*.

More information:

Configure Integration to Service Desk (see page 250)

Create an Event Based Policy

You can define new event based policies under the following categories:

- Asset File Collection
- Asset Inventory
- Asset Software
- Asset Software Usage

Note: For more details about these categories, see the Event Based Policy topic in the Asset Management section of the *DSM Explorer Help.*

To create an event based policy

1. Navigate to the Policies, Event Based folder.
2. Right-click any of the folder specified above list and select New.

   The Policy Designer dialog appears.
3. Specify the details in the following sections:

   General

   Defines the name, assigns severity, and enables history tracking for the policy.

   Service Desk

   Enables the CA Service Desk Manager integration. You can select the problem types and choose to a raise service ticket whenever the policy is violated.

   Note: This section is enabled only if you have set the Service Desk integration to True in the Default Computer Policy. By default, it is set to False.

   Add Action

   Specifies the action type to be triggered by the policy.

   Based on the event category selected, any of the following section is displayed in this dialog:

   Asset File Collection: File Collection

   Specifies the file to be monitored. Policies in this category are triggered when the selected file is modified.

   Asset Inventory: Inventory Item

   Specifies the inventory item to be monitored by the policy. Policies in this category are triggered when the value of the selected inventory item changes.

   Asset Software: Application

   Specifies the software application to be monitored and the events to be tracked by the policy. This type of policy is triggered when the selected event occurs on the selected application.

Asset Software Usage: Application

Specifies the application to be monitored for software usage. You can select the various usage events to be tracked such as, Execution was prevented, Started normally, stopped abnormally and so on. This type of policy is triggered when the selected usage event occurs on the selected application.

The policy is created with the given specifications.

Note: For more details about the Policy Designer dialog, see the *DSM Explorer Help*.

More information:

Define an Event Policy (see page 148)

Adding Actions to the Policy

Policy actions define the actions to be taken when a policy is violated. For example, you may want a message to be sent to any users when they reach a certain threshold in storage space. You can also execute the policy actions when the asset no longer violates the policy.

Note: For more details about policy actions, see the Asset Management section of the *DSM Explorer Help*.

Notification

Displays an Alarm on the DSM Explorer. You can construct message texts as with Event Log.

Message

Creates a Display Message Job on the agent. You can construct message texts as with Event Log.

Schedule Job

Schedules a specified job to be executed on a violating asset. The job runs the next time the agent runs.

Launch Application

Runs a selected application on the engine when an asset triggers a policy.

Add to Group

Adds the asset to a specified domain group.

Generate Text File

Appends text to the TXT file containing the message text generated by the engine when the policy occurs. You can construct message texts as with Event Log.

Send Mail

Sends an MAPI email to the specified recipients. You can construct message texts as with Event Log.

Note: The mail client must be running to make any email policies work.

Delete Asset

Removes the asset from the System.

System Event Log

Logs policy notification in the System Event Log. You can construct a message text to appear in the policy notification. For example, $NAME$::$TYPE$ will display the specific Asset Name, and Asset Type as found in the database, in the Message Text. This text is contained in the text file of a notification, which is triggered by the Engine when the policy occurs.

Unicenter Event Log

Logs the policy notification in the Unicenter Event Console. You can access the event console from Tools, Unicenter, Event Console menu in the DSM Explorer.

Note: The Engine generates the log entry on the computer on which it is installed.

AMT Command

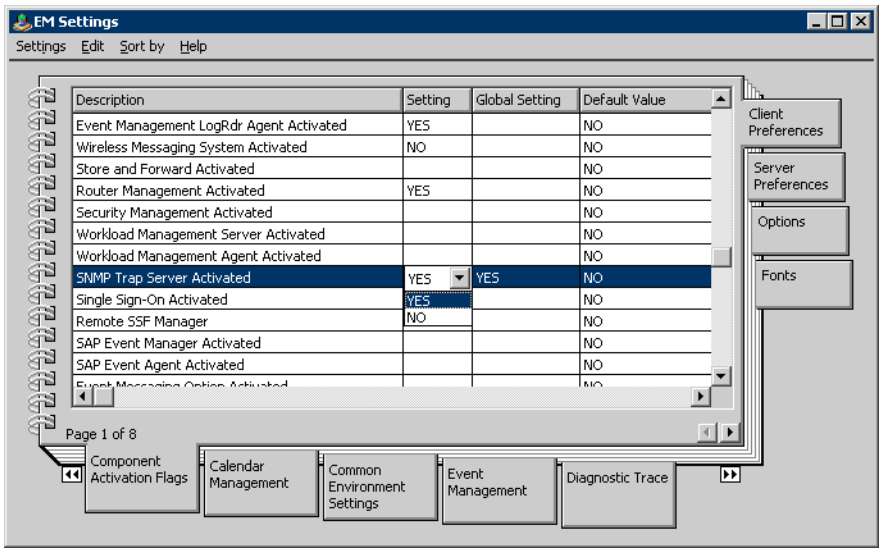
Runs the Intel AMT remote commands on the Intel AMT assets when the policy is violated. These commands let you control the Intel AMT assets.

SNMP Trap

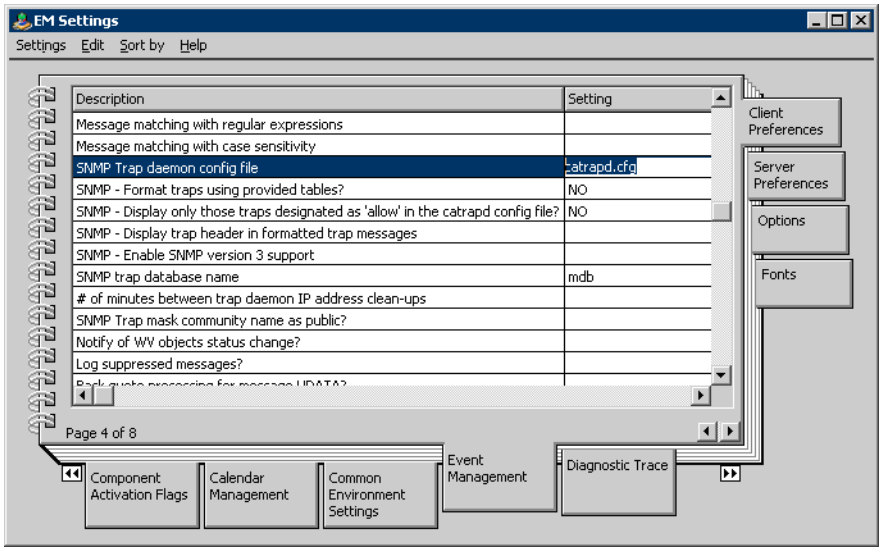
Triggers an SNMP Trap. You can construct message texts as with Event Log. SNMP traps are used in combination with Unicenter Enterprise Management enabling integration between the CA products.

Note: You must have installed the SNMP functionality on the asset which is typically a Windows component. Additionally, you need to enable the following settings for CCS event management (or Unicenter NSM):

- Make sure that you have enabled the SNMP Trap Server Activated option in Unicenter Enterprise Management as specified in the following illustration. Restart the computer in order to make this work properly.



- If you are using a configuration file, make sure that you enable the event manager to use the specified file as mentioned in the following illustration.



- In the configuration file, make sure that you enable the OID used by CAITCM SNMPTRAP.EXE utility.

```
# CAITCM SNMP traps

*:*:*:*:1.3.6.1.4.1.791.*                allow            # Enable AM
Policy traps
```

Policy Evaluation

Policy evaluation determines when and how the policy will be triggered.

Query based policies get evaluated in the following situations:

- Manually—you can right-click a policy and select Evaluate Now.
- Automatically—you can specify the evaluation option in Evaluation section of the Policy Designer dialog. You can either evaluate the policy when the collect task processing is done or designate the engine to perform the policy evaluation at the specified interval.

Event based policies get evaluated whenever the specified event occurs.

Note: For more information about Policy Designer dialog, see the Asset Management section of the *DSM Explorer Help*.

Predefined Policies

Asset management provides the following predefined query based policies. By default, these policies do not have any actions associated with it. You can define the actions you want to perform when an asset violates or no longer violates the policy.

Assets with Virtual Application Images for which the Definition is Unsealed

Evaluates the computers with discovered virtual application images for which the virtual application image definitions are unsealed.

Asset low on resources (Performance rated 75% used)

Evaluates the computers with the daytime system performance over 75%.

Users with AM Job Status Error

Evaluates whenever the asset jobs fail in a user login with the status as Error.

Assets with AM Job Status Error

Evaluates whenever the asset jobs fail on the computers with the status as Error.

Memory decreased

Evaluates whenever the memory on the assets has decreased from the previous scan value.

Protection CA Anti-Spyware, CA Anti-Virus, and CA Intrusion Prevention Status

Evaluates whenever the related signature download is older than 14 days or the status of the related module is Not Operational.

The predefined policies for the event based policies are available in the following categories. As with the query based policies, you can also define actions for these event based policies.

Note: You cannot create or delete the policies in these categories.

Asset Job Linking

Evaluates whenever the assets link or unlink a job.

Asset Jobs

Evaluates whenever the asset jobs are created, modified, or deleted.

Assets

Evaluates whenever the assets are created, modified, changes the file ID, sector or asset DNA, or registers relation.

Collect Tasks

Evaluates whenever the collect tasks are created, modified, or deleted.

File Collection Definition

Evaluates whenever the file collection are created, modified, or deleted.

Inventory Detection Modules

Evaluates whenever the inventory detection modules are created, modified, or deleted.

Inventory Template Modules

Evaluates whenever the inventory template modules are created, modified, or deleted.

Policies

Evaluates whenever the policies are created, modified, or deleted.

Queries

Evaluates whenever the queries are created, modified, or deleted.

Query Result

Evaluates whenever the query results are created or deleted.

Software Definition

Evaluates whenever the software definitions are created, modified, or deleted.

Policy Severity

The type of policy indicates the severity of a policy. Depending on the significance or severity of the policy, you can assign a severity in the General section of the Policy Designer dialog. When a computer violates the policy, the icon to the left of the computer in the DSM Explorer changes based on the severity of the policy.

Alarm

Indicates a critical situation and requires immediate action. The computer icon changes to Red when a computer violates a policy with this severity.

Warning

Indicates a potential problem. This type of policy is identical for advanced notification. The computer icon changes to Yellow when a computer violates a policy with this severity.

Message

Indicates that it is only for information. This is the default type. The computer icon changes to Blue when a computer violates a policy with this severity.

Event Log

An event log records policy notifications. You can construct a message text with built-in variables to appear in the policy notification.

For example, $NAME$, $TYPE$ displays the name of the asset that has violated the policy and its type.

This text is contained in the text file of a notification, which is triggered by the Engine when the policy occurs.

Note: For more information about the notification actions, see the Asset Management section of the *DSM Explorer Help*.

Example: Handle Duplicate Network Addresses

There are two stages to creating a policy set off by duplicate network addresses.

- Setup of a query that catches nonunique network addresses
- Setup of a policy that acts on that query

To set up a query

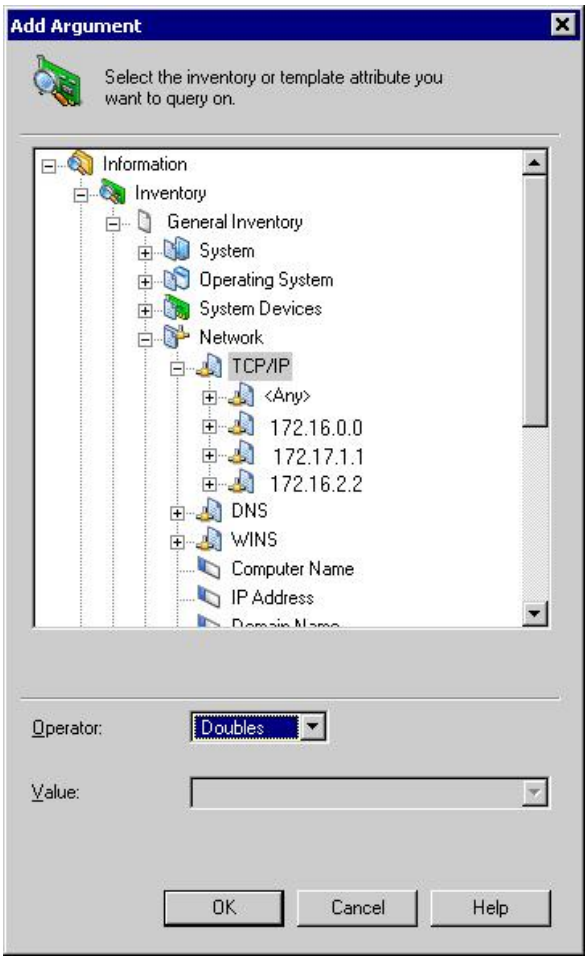
1. Navigate to Queries.
2. Right-click the Queries folder and select New.

   The Select Target dialog appears.
3. Select Computers and click OK.

   The Query Designer dialog appears.
4. Select Inventory, Discovered Inventory

   The Add Argument dialog appears.

5. Navigate through the tree view as per the illustration below and click OK.



6. Enter the Query a name, and click OK.

You have to create a policy based on the above query.

To setup a policy

1. Navigate to Policies.
2. Right-click Query Based Policy and select New.

   The Policy Designer dialog appears.
3. Select the above query from the list of Queries.
4. Add the required actions to the policy and click OK.

This policy is triggered whenever a user tries to specify a duplicate IP address in the network and performs the actions specified.

Example: Track Installation and Uninstallation of Applications on the Agent Computers

You can create an event based policy to trigger an action when applications are installed or uninstalled on the asset management agent computers.

Note: If you want to track both installation and uninstallation, you must create two different policies.

Note: The Asset software policy gets triggered only if the software is installed or uninstalled using software jobs and the software has a related software definition in CA ITCM.

To track the installation or uninstallation of applications

1. Navigate to Policies, Event Based, Asset Software folder.
2. Click New in the Tasks section.

   The Policy Designer dialog appears.
3. Click Application.

   The Software pane appears.
4. Select the event (installation or uninstallation) on which the actions must be performed.
5. Select one of the following options:

   Track all applications

   Tracks the installation or uninstallation of all the applications.

   Track specific application

   Tracks the installation or uninstallation of a specific application. You can select the application you want to track from the drop-down list.
6. Add the actions that you want to perform when the policy is triggered.
7. Click OK.

   The policy is saved.

Integration with CA Service Desk Manager

Asset management integrates with CA Service Desk Manager to automatically raise service desk tickets whenever a policy is violated. You can also create ad hoc service desk tickets in the context of a managed computer asset. The DSM administrator is enabled to launch the CA Service Desk Manager web browser, so that tickets associated with software policies can be viewed immediately. Similarly, the Service Desk Analyst is enabled to launch the DSM Explorer and the Web Console to locate the violated policies.

As a prerequisite for enabling the integration, you must have enabled the Service Desk Integration in the common configuration policy.

The following features are implemented:

Problem Types

Asset management populates the problem types from the CA Service Desk Manager database and displays in the Service Desk section of the Policy Designer dialog. You can select any of these problem types when creating a policy. Asset management assigns this problem type when creating a ticket in CA Service Desk Manager.

Duplicate Ticket Handling

CA Service Desk Manager has a duplicate ticket handling policy to decide whether to open a new issue for each policy violation or just append to the activity log of the same issue. You can also configure a combination of both so that a new issue is opened for example, every day.

Closure of Tickets

Asset management does not support automatic closure of service desk tickets. You have to close the tickets from the Service Desk GUI.

Note: For more information about CA Service Desk Manager integration, see the *Implementation Guide*.

More information:

Configure Integration to Service Desk (see page 250)
Create a Service Desk Ticket (see page 253)

Configure Integration to Service Desk

Asset management integrates with CA Service Desk Manager for automatically raising service tickets when an asset violates a policy. To configure this integration to Service Desk, follow these steps:

1. Navigate to Domain, Control Panel, Configuration, Configuration Policy, Default Configuration Policy, DSM, Service Desk Integration, default.

   The right pane displays the policy settings relating to service desk integration.

2. Double-click the following policy settings to change the values:

   Note: You can change these values only if you unseal the Default Configuration Policy by right-clicking the policy and selecting Un-Seal.

   Enabled

   Indicates whether CA Service Desk Manager integration is enabled.

   Default: False

   Identifier Field

   Defines a reserved field in MDB for querying all tickets being created by a specified asset or software distribution.

   Note: Instead of the default, you may want to reserve one of the string# fields that CA Service Desk Manager provides for user-specific purposes.

   Default: summary

   Logon Password

   Specifies the logon password for CA Service Desk Manager Web Services.

   Logon Service Aware Policy

   Specifies the name of the policy to log into if Type of Logon to Service Desk is set to notmanaged. If left blank, the default Service Desk policy is used.

   Note: A PKCS#12 certificate includes a policy description that will always override this value.

   Default: MANAGED_ASSET_EVENTS

   Logon User Name

   Defines the user account for the logon to CA Service Desk Manager Web Services.

   Default: System_MA_User

   Service Desk Endpoint

   Specifies the URL to CA Service Desk Manager Web Services. Replace *myhost* with the appropriate server address for your installation. Note that port 8080 is the default port.

   Default: http//*myhost*:8080/axis/services/USD_R11_WebService.

Throttling

Indicates whether networking and CPU throttling are enabled.

Default: False

Timeout

Specifies the timeout interval for calls to CA Service Desk Manager Web Services. Valid values are: 0 = infinite, positive integers = seconds, and negative integers = milliseconds.

Example: A value of 200 means a call will time out after 200 seconds, but a value of -200 means it will time out after 200 milliseconds.

Default: 120

Type of Logon to Service Desk

Indicates how logon to CA Service Desk Manager Web Services is controlled. Valid values are: managed (by PKCS#12 certificate) and notmanaged (by user account and password).

Default: managed

The integration to the Service Desk is configured as per the settings specified in the above fields.

Note: You must restart the DSM Explorer GUI for the integration to take effect.

Important! This section is intended for the administrators only. Entering incorrect information may result in the product not working properly. Hence, it is recommended that you create a new policy and modify the same.

Configure Service Desk Integration for a Policy

Asset management can automatically raise a CA Service Desk Manager ticket when a managed asset violates a policy. When a policy violation occurs, the CA Service Desk Manager can do one of the following:

- Create a new ticket for the policy violation.
- Append to the activity log if a ticket is already created for the same policy violation.

Note: CA Service Desk Manager makes the decision based on the duplicate ticket handling policy. If you want to create a new ticket each time a policy is violated, contact the CA Service Desk Manager Administrator.

To configure Service Desk Integration for a policy

1. Right-click the policy and select Properties.

   The Policy Designer dialog appears.

2. Click Service Desk in the left pane

   The Service Desk section appears.

   Note: The Create Service Desk ticket option is available only if you have enabled the CA Service Desk integration. For more information, see Service Desk Integration Policy Group under Configuration Policy in the *DSM Explorer Help*.

3. Select the Enable Service Desk Integration check box and the problem type to be assigned in CA Service Desk Manager for the policy violation; specify the comment to be included as the description in the service desk issue, and click OK.

   The policy is saved. This policy will create a ticket when any managed asset violates the policy or will append to the activity log.

More information:

Configure Integration to Service Desk (see page 250)

Create a Service Desk Ticket

You can manually create a service desk ticket in the context of computer assets from the DSM Explorer. For example, you can create a service desk ticket for a computer that has either violated policies or has failed jobs.

To create a service desk ticket

1. Right-click a computer asset, and select Create Service Desk ticket from the context menu.

   The CA Service Desk Create New Ticket window appears with the selected asset name in the Asset field.

   Note: The Create Service Desk ticket option is available only if you have enabled the CA Service Desk integration. For more information, see Service Desk Integration Policy Group under Configuration Policy in the *DSM Explorer Help*.

2. Specify the other required fields in CA Service Desk and click OK.

   A service desk ticket is raised.

Note: For more information about creating tickets in CA Service Desk, see the *CA Service Desk Implementation Guide*.

Create Service Desk Ticket for a Failed Job

Asset management integrates with CA Service Desk Manager only through policies. So, if you want to automatically raise a ticket for a failed job, you must do the following:

- Create query that retrieves the failed jobs
- Create a policy based on the query. This policy can be configured to create a ticket for a failed job.

To create a query to retrieve the failed jobs

1. Right-click the Queries folder and select New.

   The Select Target dialog appears.

2. Select Computers and click OK.

   The Query Designer dialog appears.

3. In the Insert Argument section, select Jobs, Asset Jobs.

   The Select Field dialog appears.

4. Expand the tree and select Job Status.

5. Select the job to be queried in the Job field, select the Status as Error and click OK.

   The Select Field dialog closes.

6. Click OK in the Query Designer dialog.

   The Save Query dialog appears.

7. Enter the name of the query and click OK.

   The query is saved with the given name and is displayed under the Queries, Computers folder.

To create a policy based on the above query

1. Right-click the Policies, Query Based folder and select New.

   The Policy Designer dialog appears.

2. Specify the policy name, policy severity and select the above query from the list of queries in the General section.

3. Click the Service Desk section and enable the service desk integration, select the problem type and specify the description or comment and click OK.

   The policy is saved. This policy checks whether the selected job is failed on any managed computer. If yes, it creates a service desk ticket for the policy violation.

Duplicate Ticketing Policy

CA Service Desk Manager implements duplicate handling based on the policy that each ticket relates to. The CA Service Desk Manager administrator can configure the following behavior based on the problem type:

Ignore the Duplicate Ticket

Ignores and discards the duplicate tickets when more than one ticket is created for the same policy.

Create a new ticket

Creates a new ticket each time a policy is violated

Append the existing ticket with a log comment

Appends the log comment in the first ticket with the description in the second one. When a policy is first violated, CA Service Desk Manager creates a ticket for the policy violation and for the subsequent occurrences it just appends a log comment to the first ticket.

Create a child ticket

Creates a child ticket if a ticket for the policy violation is already available.

Note: Each time a managed asset violates or no longer violates a policy, a service desk ticket is created for that asset. However, if a policy is re-evaluated and the violation state of the asset is unchanged, no new ticket is created.

Enable or Disable a Policy

You can disable a policy when you do not want the policy to get triggered for a while and enable it when you want to apply the policy again.

To disable or enable a policy

1. Navigate to the Policies folder.
2. Select the appropriate folder - Query Based or Event Based under it.

   The policies available under the category are displayed.
3. Right-click the policy you want to disable or enable and click Disable or Enable.

   The policy is disabled or enabled accordingly.

Delete a Policy

When you no longer want to trigger a policy, you can delete it from the database.

To delete a policy

1. Navigate to the Policies folder.
2. Select the appropriate folder - Query Based or Event Based under it.

   Lists all the policies available under the category.
3. Right-click the policy you want to delete and select Delete.

   A confirmation message appears.
4. Click OK.

   The policy is deleted from the database.

Asset Collector

Using the Asset Collector, you can create inventory information for the following sources, for example:

- Proprietary devices that are not currently supported by CA IT Client Manager
- Mainframe computers
- Computers with operating systems that are not currently supported by CA IT Client Manager
- Computers on which you do not want to install the agent
- Users

The Asset Collector allows you to capture details of assets, users and their associated inventory that cannot currently be accomplished with the DSM infrastructure. You can also track the origin and trustworthiness of this inventory, giving you greater control and management over your assets.

The traditional approach of collecting inventory data from a device has been to create a custom DSM agent that runs on specific target hardware. To date, our DSM agents collect asset information from traditional sources. The Asset Collector is a component that collects the hardware and software inventory information from well-formed inventory files. Due to the simple and flexible architecture of the Asset Collector, you are able to create inventory information for any device or user. This produces a significant reduction in cost.

Note: In CA ITCM Release 12.8, the Asset Collector is enhanced to enable processing of assets and inventory from various tenants or organizations. For more information, see the *Asset Collector Release Notes* available on the CA support site (http://ca.com/support).

Prerequisites

The Asset Collector CAF plug-in is automatically installed with a scalability server on a Windows platform.

However, the asset collector plug-in is not enabled or running by default.

To enable the plug-in you have to run the following command:

```
caf enable assetcollector
```

The plug-in is then visible in a caf status.

Start the asset collector plug-in with the following command:

```
caf start assetcollector
```

How the Asset Collector Works

The following steps are performed to collect inventory information from the inventory files:

1. You place all of the inventory files in a particular folder and configure the relevant common configuration policy to specify the location of the folder.

   If not specified using the configuration policy, the following folders are created:

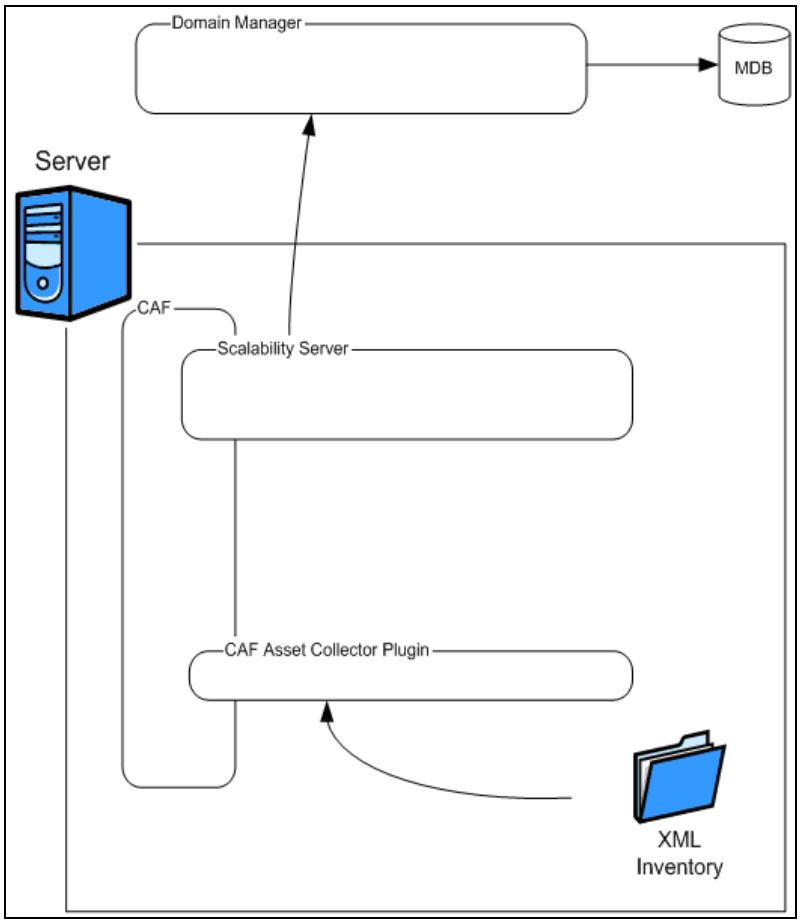
   - A collection folder is created at the *DSM Install Path*\AssetCollectorCollect
   - An output folder is created at *DSM Install Path*\AssetCollectorOutput
   - A Delta working folder will be created at *DSM Install Path*\AssetCollectorBAK

   Note: We recommend that this folder is well protected using your operating system security, so that only authorized users can create, modify, or delete the inventory files.

2. The Asset Collector monitors the folders that you configured in the configuration policy for new inventory files.
3. When a new inventory file is detected, the file is parsed to extract the asset, and/or user inventory information.
4. If the inventory information is already processed, the Asset Collector compares the previous and the current versions and sends only the differential inventory to the scalability server; otherwise, full inventory is sent to the scalability server.

The scalability server processes the information and forwards it to its DSM manager for storage in the Management Database (MDB).

The following graphic illustrates how the Asset Collector works:



Inventory File Types

The Asset Collector processes files with the following extensions:

.xiu = xml inventory unsigned files.

.xis = xml inventory signed files.

Both of these file types are standard text files containing XML.

More information:

invSign—Sign, Verify, or Unsign an Inventory File (see page 285)

Differencing

Differencing in the Asset Collector is the process of establishing the differences in separate versions of the same file.

When an inventory file is provided (if differencing is enabled), the previous version of the file that was processed is compared to the current inventory. Only the differences are delivered to the scalability server. This enables an optimized modification of the data to be performed in the MDB.

Origin and Trust Level

Origin and Trust Level are new fields introduced with the Asset Collector. You use them to define the source and trustworthiness of an asset. You can define these fields in the inventory file or within the Asset Collector's configuration and then view the new columns in the GUI.

Origin

Defines the source from which the asset\user is collected and the information is displayed in the DSM Explorer. The source can be a CA ITCM agent or an Asset Collector. For the Asset Collector, you can specify the origin of a collected asset\user either in the inventory file, or in the Asset Collector configuration policy. Typical values for origin can be User Keyed, SMS Extract, and Legacy. Assets collected by the CA ITCM agents have the origin as CA.

Trust Level

Defines the trustworthiness of an asset\user and displays the information in the DSM Explorer. Trust level is displayed as a range of 1 to 5 stars, 5 being the most trustworthy. For Asset Collector, you can configure the trust level in the inventory file or in the Asset Collector configuration policy. As the inventory information collected by the Asset Collector is initiated by a manual process, the reliability of the information may be low. Consequently, the information collected by the Asset Collector has the trust level of 3 by default. The information collected by the CA ITCM agents has the trust level of 5.

More information:

Configure the Asset Collector (see page 262)
Configure the Asset Collector for a Typical Environment (see page 264)

Schema Changes for Origin and Trust Level

To store the origin and trust level information of the assets/users, the following schema changes have been made to asset tables:

Table Name	Column Name	Column Datatype	Column Null Option	Column Comment
inv_root_map	trustlevel	integer	not null	Defines the trust level for a component discovered on an asset/user. **Default:** 5
inv_root_map	origin	nvarchar(64)	not null	Defines the origin information for a component discovered on an asset/user.
ca_agent	trustlevel	integer	not null	Defines the trust level of an asset/user. **Default:** 5
ca_agent	origin	nvarchar(64)	not null	Defines the origin of an asset/user.
ca_discovered_hardware	external_host_key	nvarchar(64)	not null	Defines the key generated or passed by the Asset Collector.
ca_discovered_software	trustlevel	integer		Defines the trust level of discovered software of an asset/user. **Default:** 5
ca_discovered_software	origin	nvarchar(64)	not null	Defines the origin of discovered software of an asset/user.

Reconciliation

Reconciliation is the process that is applied if an agent is installed onto an asset that was being reported through the Asset Collector. The Asset Collector information is overwritten with the information from the agent. Any subsequent inventory from the Asset Collector is ignored.

The Asset Collector cannot be used to append inventory information to a currently reported asset.

Note: For reconciliation to work the asset registered by the Asset Collector has to be using the same Media Access Control (MAC) address and host name as the computer.

Reconciliation also occurs for user objects that are created using the Asset Collector. The User is registered using a URI as its key. If a user with the same URI is registered via a DSM Agent, it overwrites the collected information, and any subsequent Asset Collector inventory for this URI is ignored.

Host Keys

Before an agent registers an asset with the MDB, a UUID is required. The UUID is typically created on the agent computer and stored on the agent computer for future use.

However, because the inventory file is supplied at the server, the Asset Collector uses a unique host key (*host_key*) in place of the UUID. This can be defined in the inventory file (*host_key*) or if it is not specified, it is automatically created for the asset. To facilitate the reconciliation process when an agent is installed on the asset, the collected asset data must include at least the MAC address and host name of the asset being registered.

When the Asset Collector generates a host key for an asset, it writes this key back into the inventory file of the asset. The Asset Collector also maintains a list of host names, MAC address pairs, and their associated host keys, which allows the same key to be used if the inventory file is again delivered without a host key.

The list of host keys is written to storage so that it is available the next time the Asset Collector is started. This list can be purged of outdated entries at a configurable interval using a configuration policy, and the age of the data that will be purged is also configurable.

User URIs

User Objects also have a unique identifier. The user's unique identifier is the URI field that is specified in the inventory file. This means that if the URI in an inventory file is changed and the inventory file resubmitted to the Asset Collector, a new user object will be created. You will have to manually delete the original user object.

Asset Collector Configuration

You can configure the Asset Collector using configuration policies. A new Asset Collector subnode and policy group folder have been added to the DSM Explorer tree under Control Panel, Configuration, Configuration Policy, Default Computer Policy, DSM, Scalability Server.

Note: For detailed information about Asset Collector configuration policies, see the Configuration Policy section of the *DSM Explorer Help*.

Note: The Asset Collector collection, output, and delta working folders should not be configured to point to the same folders. If they do point to the same folders when the server starts, the Asset Collector service closes and raises a system event.

Configure the Asset Collector

You can customize, or configure, the Asset Collector for your specific environment.

To configure the Asset Collector

1. From the DSM Explorer, navigate to the Control Panel, Configuration, Configuration Policy node.
2. Right click on Configuration Policy and select New Policy.

   The New Configuration Policy dialog appears.
3. Enter a Name and Description and click OK.

   The new configuration policy appears in the tree view.
4. Click on the new configuration policy node you have just created to expand it.
5. Click on DSM, Scalability Server, Asset Collector.

   All of the Asset Collector policies appear on the Asset Collector pane.
6. Double-click the appropriate setting, for example, Collection Folders.

   The Setting Properties dialog appears.
7. Set the value for the setting.

8. Repeat Steps 3 and 4 as necessary.
9. Click OK.
10. Right-click on the policy node and select Seal from the menu.
11. Drag-and-drop the policy onto the Scalability Server in the All Computers folder.

    This applies the policy to the Scalability Server you want to configure with this policy.

The Asset Collector is configured for your environment.

Note: Before you can modify a policy, you must unseal it. For more information, see the Configuration Policy section of the *DSM Explorer Help*.

More information:

Origin and Trust Level (see page 259)

Configuration Scenario

The following scenario is typical for generating inventory from a manually created inventory file. It includes the following configuration options:

- Define the collection folder.
- Specify the origin and trust level.
- Define the working and output folders.
- Indicate whether the collection folder's *child folders* are scanned for new inventory files.
- Specify the name of the server.
- Indicate whether digitally signed files only are processed.

More information:

Origin and Trust Level (see page 259)
Digitally Signed Inventory Files (see page 280)

Configure the Asset Collector for a Typical Environment

Use the following procedure to configure the Asset Collector for a typical environment.

To configure the Asset Collector for a typical environment

1. Navigate to the Control Panel, Configuration, Configuration Policy node.
2. Right-click on Configuration Policy and select New Policy.

   The New Configuration Policy dialog appears.
3. Enter a Name and Description and click OK.

   The new configuration policy appears in the tree view.
4. Click DSM, Scalability Server, Asset Collector.

   All of the Asset Collector policies appear on the Asset Collector pane.
5. For Steps 4-11, double-click the relevant policy, modify or accept the default value in the corresponding Setting Properties dialog as specified, and click OK.
6. Leave the value for the Collection Folders policy blank.

   By default, the AssetCollectorCollect collection folder is created in the DSM installation directory.
7. Accept the default value, XML File, for the Default Origin policy.

   This value is left unchanged in this scenario, as the inventory that will be generated is from a manually created inventory file.

   Note: If, however, you had an automated task creating the inventory files, you could change this value to reflect this. For example, suppose you have a tool that lets you create inventory files based on handheld devices in your enterprise. If this tool delivers its inventory file output to the AssetCollectorCollect collection folder, then you could specify the default origin as "Handheld Inventory Collector." The default origin and trust level are only used when the inventory file does not contain an entry for origin or trust.
8. Accept the default value, 3, for the Default Trust Level policy.

   Because the source of the inventory is a manually created inventory file, the inventory entered could become outdated easily. A trust level of 5 is not used because of the reasons stated (data could become invalid).

   Note: It is up to the DSM administrator to determine what trust level is used.
9. Leave the value for the Delta Working Folder policy blank.

   By default, the AssetCollectorBAK working folder is created in the DSM installation directory.
10. Leave the value for the Output Folder policy blank.

    By default, the AssetCollectorOutput output folder is created in the DSM installation directory.

11. Set the Recursive policy to True.

    A value of True specifies that the selected collection folder can be scanned for subfolders with new inventory files.

    Note: This setting has no effect if the collection folder has no subfolders.

12. Accept the default server, *localhost*, for the Server Name policy.

    This policy specifies the name of the server to which inventory files are uploaded and registered. In most cases, the Asset Collector is installed on a scalability server, so that the inventory can be delivered to the local machine.

13. Set the Signed Files Only policy to False.

    A value of False indicates that all files, digitally signed and unsigned, are processed. A value of True indicates that the Asset Collector only processes files that have valid and trusted digital signatures.

14. Right-click on the policy node and select Seal from the menu.

    Note: For more information about sealing and unsealing policies, see the Configuration Policy section of the *DSM Explorer Help*.

15. Drag-and-drop the policy onto the target computer in the All Computers folder.

    This applies the policy to the target computer you want to configure with this policy.

    The Asset Collector is configured for this scenario.

More information:

Origin and Trust Level (see page 259)
Digitally Signed Inventory Files (see page 280)

Tenancy Collection

Tenancy lets you manage the asset information collected from various sources within the same MDB. The collected asset information is imported into the MDB in such a way that the tenancy membership of the asset is maintained and managed within the same MDB.

CA ITCM is not multi-tenant capable. CA ITCM, however, can collect external inventory files and store any tenant information which can be used by other multi-tenant capable CA Products.

Asset Collector uses the collection folders to receive inventory files. You can configure the collection folders to associate tenants with individual collection folders.

The tenancies are defined in the *ca_tenant* MDB table. Defining a tenancy on a collection folder lets the engine populate a new column named *tenant_id* on the *ca_asset* table in the MDB. The column *tenant_number* from the ca_tenant table is used to configure the Asset Collector.

Note: CA ITCM cannot populate the ca_tenant table, although other CA products such as CA Service Desk Manager or CA IT Asset Manager can populate it. So, when you define tenants with CA Service Desk Manager or some other CA product, specify a tenancy number for each tenant. Asset Collector uses this tenancy number to differentiate between tenants.

Enable the Database for Tenancy Collection

By default, the CA ITCM database is not configured to perform tenancy collection. You must enable a number of database triggers for the MDB to maintain the tenancy columns in the database.

Execute the following statement to enable the triggers:

For Oracle database:

```
execute sp_enableTenantTriggers(1);

commit;
```

Note: You must execute the command as *mdbadmin* and not the ca_itrm user on Oracle.

For Microsoft SQL Server database:

```
exec sp_enableTenantTriggers 1
```

Note: You must execute the command in the MDB namespace.

You can change your session to MDB namespace by executing the following command:

```
use mdb;
```

Configure Tenancy Collection

You can manage the asset information collected from various sources within the same MDB by configuring the Asset Collector configuration folders.

You can configure the Asset Collector configuration folders by specifying:

- Tenancy for each collection folder
- Collection folders without specifying tenants
- Multiple collection folders for a single tenant

Note: You cannot configure one collection folder for multiple tenants.

To configure collection folders for tenancy collection

1. Open DSM Explorer, navigate to the Control Panel, Configuration, Configuration Policy node.
2. Right-click a sealed policy and select Un-Seal.

   The policy is unsealed.
3. Expand the unsealed policy, and navigate to DSM, Scalability Server, Asset Collector.

   The Asset Collector policies appear on the Asset Collector pane.
4. Double-click Collection Folders.

   The Modify Setting dialog appears.
5. For each row, define a collection folder and click OK.

   Note: The tenancy number column is optional. A value specified in this column has to match with an entry in the ca_tenancy column of the ca_tenancy table in the MDB.
6. Right-click the policy node and select Seal from the menu.

   The policy is sealed.
7. Drag-and-drop the policy onto the scalability server in the All Computers folder.

   The policy is applied to the scalability server.

Rules for Processing Inventory Files

You can specify rules for processing the inventory files collected from multiple tenants. The rules can be based on one of the two attributes (trust level and collection time) of the inventory files.

There are two modes of operation:

Trust Enabled Mode (TRUE)

Processes the inventory file based on the trust level.

An inventory file with trust level equal to or greater than the trust level of the previous inventory file propagates to the scalability server.

Trust Disabled Mode (FALSE)

Processes the inventory file based on the collect time.

An inventory file with collect time higher than the collect time of the previous inventory file propagates to the scalability server.

To prevent resubmission of inventory records taken on the same day, define the s*ame day window* configuration in seconds so that any inventory file having a collection time within the same day window is not processed.

If you set the same day window to zero, the check is not performed, and all inventories with a subsequent collection time are processed.

Note: If you do not want to define processing rules for every tenant, you can define default-processing rules for the following configurations:

- Every tenant that does not have any rule configured
- Inventory files without a tenancy number in the collection folder

Configure Rules for Processing Inventory Files

You can specify rules for processing the inventory files collected from multiple tenants. To accept or reject an inventory file, configure rules based on trust level or collection time.

To configure rules for processing inventory files

1. Open DSM Explorer, navigate to the Control Panel, Configuration, Configuration Policy node.
2. Right-click a sealed policy and select Un-Seal.

   The policy is unsealed.
3. Expand the unsealed policy, and navigate to DSM, Scalability Server, Asset Collector.

   The Asset Collector policies appear on the Asset Collector pane.
4. Double-click Processing Rules.

   The Modify Setting dialog appears.
5. Complete the fields in the dialog.

   The following fields are not self-explanatory:

   Tenant Number

   Specifies the tenant number.

   The tenant number here must match a tenant number defined in the collection folders table, and hence the tenant_number column of the ca_tenant table.

   Trust Mode

   Specifies if trust mode or collection time is used to process the inventory files. Set the value to TRUE to use trust level or FALSE to use collection time to process the inventory files.

   Same Day Window

   Specifies the collection time window in seconds. Any inventory file that has a collection time within the same day window is not processed.

   To disable same day window, set the value to zero.

   Note: You must define only one set of processing rules for each tenant.

   Click OK.
6. Right-click the policy node and select Seal from the menu.

   The policy is sealed.
7. Drag-and-drop the policy onto the scalability server in the All Computers folder.

   The policy is applied to the scalability server.

Note: If you do not want to define processing rules for each tenant, or if you have configuration folders without a tenant number, define the following processing rules in the Asset Collector configuration section:

- Processing Rules: Default Trust Mode
- Processing Rules: Default Same Day Window

Defining the default rules results in the same behavior as the tenant processing rules, but are applied when the tenant specified does not have a rule defined, or the asset submitted does not have a tenant associated.

Map Origin to Trust Level

Using the Origin to Trust level mapping, you can define a trust level for an asset from a particular origin. Defining a trust level is useful when you are collecting inventory information from multiple origins. The mapping is used when the collected asset file does not have a trust level defined.

To configure origin to trust level mapping

1. Open DSM Explorer, navigate to the Control Panel, Configuration, Configuration Policy node.
2. Right-click a sealed policy and select Un-Seal.

   The policy is unsealed.
3. Expand the unsealed policy, and navigate to DSM, Scalability Server, Asset Collector.

   The Asset Collector policies appear on the Asset Collector pane.
4. Double-click Origin to Trust Mapping.

   The Modify Setting dialog appears.
5. For each origin, define a trust level.

   Note: Do not define multiple trust levels for the same origin. However, you can use the same trust level for multiple origins.

   Click OK.
6. Right-click the policy node and select Seal from the menu.

   The policy is sealed.
7. Drag-and-drop the policy onto the scalability server in the All Computers folder.

   The policy is applied to the scalability server.

Configure Tenant Number

You can specify a tenant number if you have other CA products like CA Service Desk Manager which require tenant classification for a CA ITCM agent. You can associate a tenant with a CA ITCM agent by specifying a *tenant number* in the configuration policy of the scalability server.

A scalability server can support only a single tenant. If you want to collect agents for different tenants, you must use a different scalability server for each tenant.

The tenant number is defined in the ca_tenant table by CA Service Desk Manager or CA IT Asset Manager.

You must configure the tenant number that will be used for assets that are registered through a scalability server and is applied only when tenant number is not provided by the Asset Collector.

To configure a tenant number on the scalability server

1. Open DSM Explorer, navigate to the Control Panel, Configuration, Configuration Policy node.
2. Right-click a sealed policy and select Un-Seal.

   The policy is unsealed.
3. Expand the unsealed policy, and navigate to DSM, Scalability Server, Common Server.

   The Common Server policies appear on the Common Server pane.
4. Double-click Tenant Number.

   The Setting Properties dialog appears.
5. Enter a tenant number and click OK.

   Note: The tenant number that is applied must match a tenant number in the ca_tenant table.
6. Right-click the policy node and select Seal from the menu.

   The policy is sealed.
7. Drag-and-drop the policy onto the scalability server in the All Computers folder.

   The policy is applied to the scalability server.

Reject Inventory Files Having a Future Collect Time

You can configure the Asset Collector to accept only the inventory files that have a valid time stamp in the *xml inventory unsigned files (.xiu)*, and reject inventory files that have a collect time in the future. You can configure same day tolerance to define *future date* to assist processing of inventory files from different time zones.

To configure rejection of inventory files having a future collect time

1. Open DSM Explorer, navigate to the Control Panel, Configuration, Configuration Policy node.
2. Right-click a sealed policy and select Un-Seal.

   The policy is unsealed.
3. Expand the unsealed policy, and navigate to DSM, Scalability Server, Asset Collector.

   The Asset Collector policies appear on the Asset Collector pane.
4. Modify the following configuration parameters:

   Collect Time: Allow inventory without a collect time

   Specifies whether inventory files without a collect time are allowed.

   Set the value to TRUE to allow inventory files without a collect time in the xml.

   Collect Time: Future Date Tolerance

   Defines the tolerance in seconds that is applied to the current time to define a future date.

   Any inventory file having a collect time in the future will be checked against the future data.

   Collect Time: Reject Future Files

   Specifies whether to reject inventory files with collection time exceeding future date.

   Set the value to TRUE to reject files that have collection time beyond the future date tolerance.
5. Right-click the policy node and select Seal from the menu.

   The policy is sealed.
6. Drag-and-drop the policy onto the scalability server in the All Computers folder.

   The policy is applied to the scalability server.

Configure Post-Processing Actions

When the Asset Collector processes inventory files, the following results are possible:

- Inventory file is accepted
- Inventory file is rejected
- Inventory file contains an error

Asset Collector lets you define post-processing actions on above events.

If the inventory file is rejected or it contains an error, you can configure Asset Collector to delete the file, copy the file to the output folder, or rename the file with a .error extension.

If the inventory file is accepted, you can configure Asset Collector to delete the file or copy the file to the output folder.

To configure post processing actions

1. Open DSM Explorer, navigate to the Control Panel, Configuration, Configuration Policy node.
2. Right-click a sealed policy and select Un-Seal.

   The policy is unsealed.
3. Expand the unsealed policy, and navigate to DSM, Scalability Server, Asset Collector.

   The Asset Collector policies appear on the Asset Collector pane.
4. Modify the following configuration parameters:

   Inventory File Rejected

   Specifies the action on the rejected inventory file.

   Set the value of the parameter to 0, 1, or 2 to delete the file, move the file to the output folder, or rename the file with a .error extension respectively.

   Inventory File Processed

   Specifies the action on the processed inventory file.

   Set the value of the parameter to 0 or 1 to delete the file, or to move the file to the output folder respectively.

   Inventory File Error

   Specifies the action on the inventory file that contains an error.

   Set the value of the parameter to 0, 1, or 2 to delete the file, move the file to the output folder, or rename the file with a .error extension respectively.
5. Right-click the policy node and select Seal from the menu.

   The policy is sealed.

6. Drag-and-drop the policy onto the scalability server in the All Computers folder.

   The policy is applied to the scalability server.

Configure Asset Collector MDB Audits

You can generate audit information in the MDB for enhanced traceability and reporting.

You can configure the Asset Collector to generate audit records. These records are written into the *CA_AC_AUDIT_LOG* table in the MDB.

Configure Asset Collector Auditing

Asset Collector maintains an internal cache of audit events and sends them to the scalability server when certain thresholds in terms of size or age are reached.

You can customize the threshold values to match your environment.

To configure Asset Collector auditing

1. Open DSM Explorer, navigate to the Control Panel, Configuration, Configuration Policy node.
2. Right-click a sealed policy and select Un-Seal.

   The policy is unsealed.
3. Expand the unsealed policy.

   The policy expands.
4. Navigate to DSM, Scalability Server, Asset Collector.

   The Asset Collector policies appear on the Asset Collector pane.
5. Modify the following configuration parameters to match your environment:

   Audit Log: Max Age

   Defines the maximum age in seconds that the audit log queue must reach before it sends the audit log to the scalability server for inclusion in the MDB.

   Audit Log: Wait Period

   Defines the polling period in seconds. The polling period is used to check the audit log queue and age by the audit component.

   Audit Log: Max Queue Size

   Defines the maximum number of items allowed in the audit log queue before it is sent to the scalability server for inclusion in the MDB.
6. Right-click the policy node and select Seal from the menu.

   The policy is sealed.
7. Drag-and-drop the policy onto the scalability server in the All Computers folder.

   The policy is applied to the scalability server.

Configure Asset Collector Auditing Events

You can configure the events that generate an audit record from the configuration section of the Asset Collector.

To configure Asset Collector auditing events

1. Open DSM Explorer, navigate to the Control Panel, Configuration, Configuration Policy node.
2. Right-click a sealed policy and select Un-Seal.

   The policy is unsealed.
3. Expand the unsealed policy.

   The policy expands.
4. Navigate to DSM, Scalability Server, Asset Collector, Events.

   The configurable audit events appear.
5. Configure the audit events to match your requirement as follows:

   Audit Accepted Assets

   Specifies if an audit record is created for each successfully processed inventory file.

   Audit Reject Collect Time Future Time

   Specifies if an audit record is created when an inventory file is rejected because the collect time specified in the file appears to be a time in the future.

   Audit Reject Collect Time Older

   Specifies if an audit record is created when an inventory file is rejected because the collect time in the inventory file is older than a previous submission for the same asset.

   Audit Reject Collect Time Same Day

   Specifies if an audit record is created when an inventory file is rejected because its collection time falls within the same day window of a previously processed asset.

   Audit Reject Missing Values

   Specifies if an audit record is created when an inventory file is rejected because a key value in the file is missing.
6. Right-click the policy node and select Seal from the menu.

   The policy is sealed.
7. Drag-and-drop the policy onto the scalability server in the All Computers folder.

   The policy is applied to the scalability server.

Asset Collector Collection Audit Table

The Asset Collector collection audit items are written to the CA_AC_AUDIT_LOG table. This table has the following columns:

Name	Description
Asset Name	Defines the host name of the asset.
MAC Address	Defines the MAC address if available.
Scalability Server	Defines the scalability server that the Asset Collector reports to.
Origin	Defines the origin of the asset.
Tenant Number	Specifies the tenant identifier of the asset.
State	Specifies if the asset is accepted (0) or rejected (1).
Event Code	Specifies the Event Code for asset rejection.
Details	Indicates the reason for asset rejection.

The Event Code column of the CA_AC_AUDIT_LOG table shows the following possible event codes:

Event Code	Reason	Description
0	Not applicable	Specifies that the asset is accepted. The value is set to zero for asset accepted events.
1	Older collection time	Specifies that the asset is rejected because of a lower collection time than the last accepted inventory file for the same asset.
2	Lower trust level	Specifies that the asset is rejected because of a lower trust level than the last accepted inventory file for the same asset having trust mode enabled.
3	Same day as last submission	Specifies that the asset is rejected because its collection time falls within the same day tolerance of the last accepted inventory file for the same asset.
4	Future collect time	Specifies that the asset is rejected because its collection time represents a time in the future.
5	Missing values	Specifies that the asset is rejected because some key data fields are missing.

Audit Table Management

You can manage the size of the CA_AC_AUDIT_LOG table by purging the old records. You can purge old records by configuring values in the configuration section of Asset Collector.

To configure purging of old records

1. Open DSM Explorer, navigate to the Control Panel, Configuration, Configuration Policy node.
2. Right-click a sealed policy and select Un-Seal.

   The policy is unsealed.
3. Expand the unsealed policy, and navigate to DSM, Scalability Server, Asset Collector.

   The Asset Collector policies appear on the Asset Collector pane.
4. Modify the following parameters:

   Audit Purge Interval

   Specifies the time in days before the audit records are purged.

   To prevent purging, set the value to zero.

   Audit Purge Max Age

   Specifies the age in days, after which the audit records are purged.
5. Right-click the policy node and select Seal from the menu.

   The policy is sealed.
6. Drag-and-drop the policy onto the domain manager in the All Computers folder.

   The policy is applied to the domain manager.

Security

This section contains security-related topics that you need to consider while using the Asset Collector.

Protect the Collection Folders

The Asset Collector collects the inventory files from the folders configured in the Collection Folders policy. We recommend that you secure these folders using operating system security so that only authorized users have the rights to create, modify, or delete the files from the folders.

Digitally Signed Inventory Files

The Asset Collector can use a Public Key Infrastructure (PKI) signing process to ensure that the creator of the file is trustworthy.

More information:

Configure the Asset Collector for a Typical Environment (see page 264)

How the Digitally Signed Inventory Files Are Verified

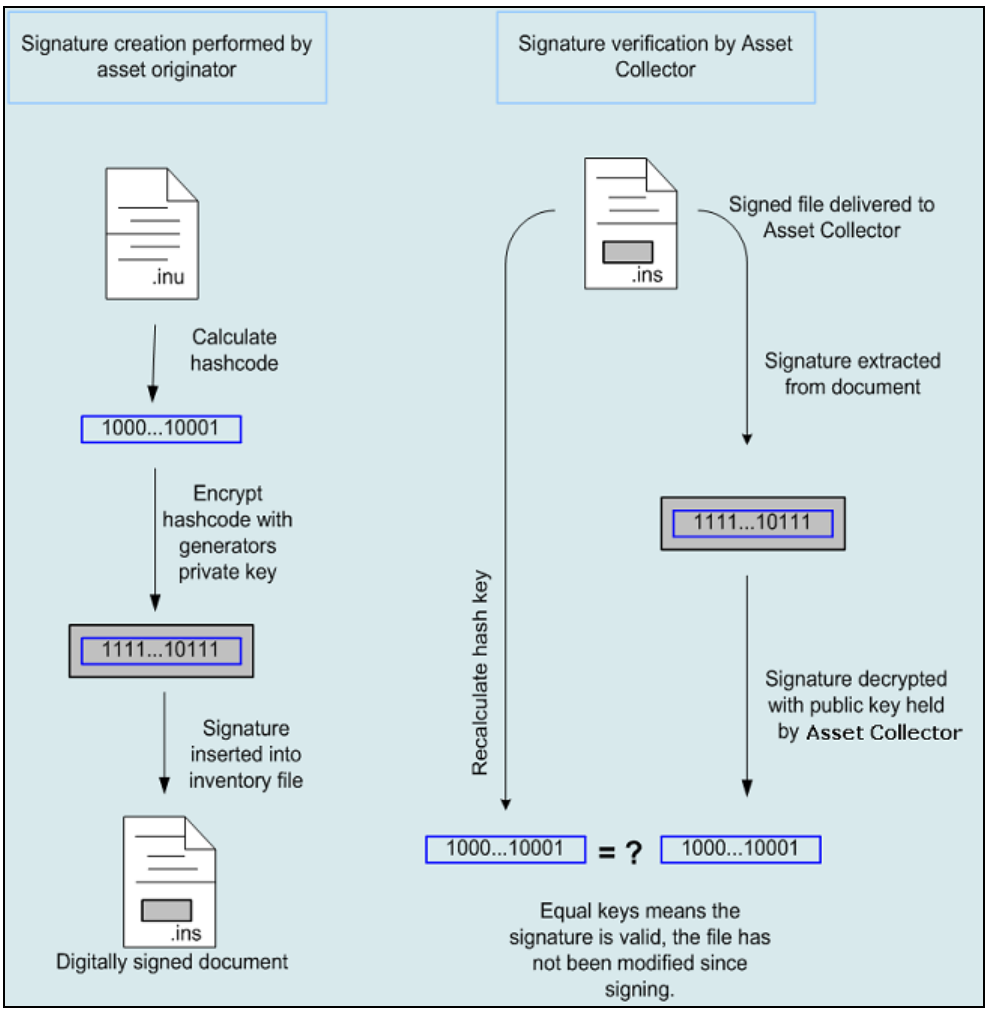
The following process explains how the digitally signed inventory files are handled.

1. The Administrator creates certificates and installs certificates on a machine where inventory files will be signed and public keys on the Asset Collector where the signature will be verified.
2. The inventory file is signed using the invsign.exe tool.

   For more information, see invSign—Sign, Verify, or Unsign an Inventory File (see page 285).
3. The inventory file is delivered to the Asset Collector Collection folder where the asset collector will validate the signature using the previously installed public key.

Important! Do not modify a signed file as it corrupts the binary information that makes up the signature.

The following illustration depicts this process:



cacertutil create—Create a Certificate

The cacertutil create command creates a certificate.

This command has the following format:

```
cacertutil create -o:filename –op:passphrase –s:subject -od:filename
```

-o:*filename*

Specify the output filename of the certificate.

-op:*passphrase*

Specify a passphrase to encrypt the certificate.

-s:*subject*

Specify the subject name to whom the certificate is issued.

-od:*filename*

Create a DER encoded file with just the public certificate.

Example: Create a Certificate

This example creates two files AssetCollectorCert.p12 and AssetCollectorCert.der. The .der file contains only the public part of the certificate and must be copied on to the Asset Collector computer where the signature is to be verified.

```
cacertutil create –o:c:\AssetCollectorCert.p12 -op:password
-s:"CN=AssetCollector,O=Computer Associates,c=US" -od:c:\AssetCollectorCert.der
```

Note: For more information about cacertutil commands, see the Command Line Reference Guide.

cacertutil import—Import a Certificate

The cacertutil import command imports a certificate into the CA ITCM comstore.

Important! This has to be done on the computer where the administrator plans to perform the signing of the inventory files and must be done before the signing.

This command has the following format:

Note: For more information about cacertutil commands, see the Command Line Reference Guide.

```
cacertutil import –i:filename -ip:passphrase  -t: tag
```

-i:*filename*

Specify the filename of the certificate to be imported.

important parameters:

-ip:*passphrase*

Specify the passphrase and store encrypted in comstore.

-t:*tag*

Specify a tag name.

Note: You must use the same tag name specified in the –t parameter on the computers that verify and sign the inventory files.

Example: Import a Certificate

```
cacertutil import –i:C:\AssetCollectorCert.p12 –ip:password –t:AssetCollector
```

cacertutil import—Import the Public Key

The cacertutil import command imports the public key on the Asset Collector server to the CA ITCM comstore.

This command has the following format:

```
cacertutil import -i :filename -t:tag -it:certificatetype
```

-i:*filename*

Specify the filename of the certificate to be imported.

important parameters:

-it:*type*

Specify the type of certificate to be imported. Valid options are X509V3 or PKCS#12.

Default is PKCS#12.

-t:*tag*

Specify a tag name.

Note: For more information about cacertutil commands, see the Command Line Reference Guide.

Example: Import the Public Key

```
cacertutil import –i:C:\AssetCollectorCert.der –t:AssetCollector –it:x509v3
```

invSign—Sign, Verify, or Unsign an Inventory File

The invsign command is used to sign, verify, or unsign an inventory file.

Important! The invsign command accepts files with the following extensions: .xiu files for signing, and .xis files for unsigning. The .xiu file is renamed to .xis after successfully signing and the .xis file is renamed to .xiu on a successful unsign.

This command has the following format:

`invsign` *`command  xiu or xis file`* [*`certificate tag`*]

Command

Includes one of the following commands:

Sign

Signs the inventory file with the given certificate tag. The signature contains a binary stream of data that is appended to end of the inventory file.

Important! Opening a signed inventory file adds unexpected characters at the end of the file. Therefore, do not open or attempt to modify a signed inventory file as it may invalidate the signature and cause the file to be rejected by the Asset Collector.

Note: The private keys must have already been imported into the certificate store using the tag name given with the sign command. Also, to sign an inventory file, a certificate must be installed in comstore of the computer where you are signing.

Verify

Verifies the signature in the inventory file.

Note: The public keys must be available under the same tag name as provided during the signing process.

Unsign

Removes the signature from the inventory file.

Inventory File

Defines the name of the inventory file that you want to sign, verify, or unsign.

Certificate Tag

(Mandatory for sign command). Defines the certificate tag that you want to assign while signing the inventory file.

Note: A certificate tag is an identifier for a unique certificate. Each certificate is assigned a certificate tag.

The Certificate tag is required only with the sign command.

Example: invSign Command

```
invsign sign Server1.xiu AssetCollector

invSign verify Server1.xis

invSign unsign Server1.xis
```

You can specify multiple files for signing, un-signing and verifying with the invsign command line.

The format of a multi-file sign is:

```
Invsign sign file1.xiu file2.xiu file3.xiu [certificate tag]
```

You can also use wild cards:

```
Invsign sign f*.xiu [certificate tag]
```

Or you can use a combination of both:

```
Invsign sign f*.xiu newComputer.xiu [certificate tag]
```

Important! You can specify multiple files and wild cards with unsign and verify commands also. With these commands the *CertificateName* should not be supplied.

More information:

Inventory File Types (see page 258)

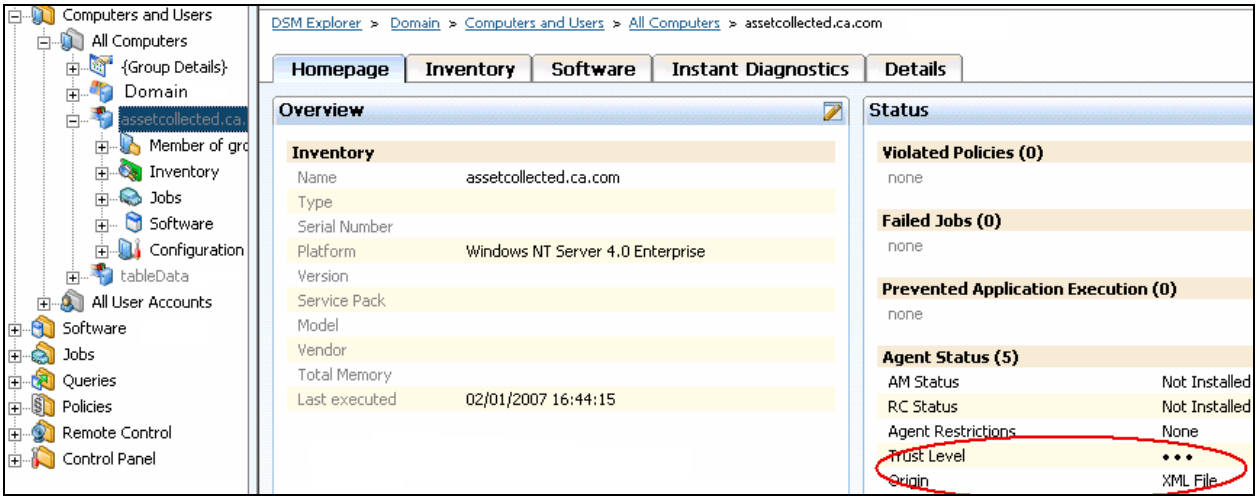
Viewing the Origin and Trust Level

You can view the origin and trust level of an asset, software inventory and hardware inventory, and user.

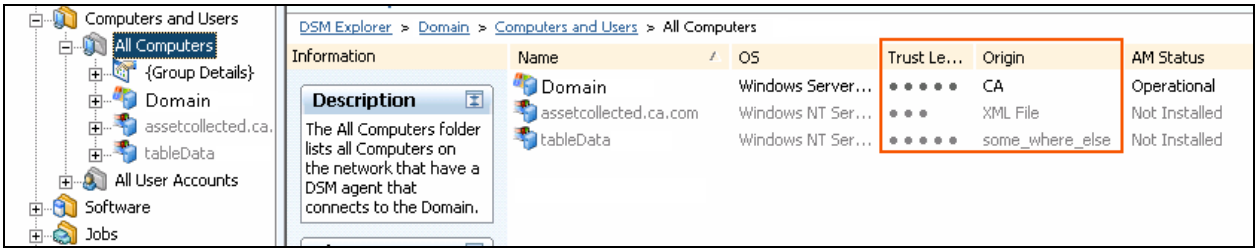
Origin and Trust Level of an Asset

You can view the origin and trust level of an asset from the following locations in the DSM Explorer:

- The home page of the asset; the Trust Level and Origin appear in the Status pane, as shown in the following illustrations:



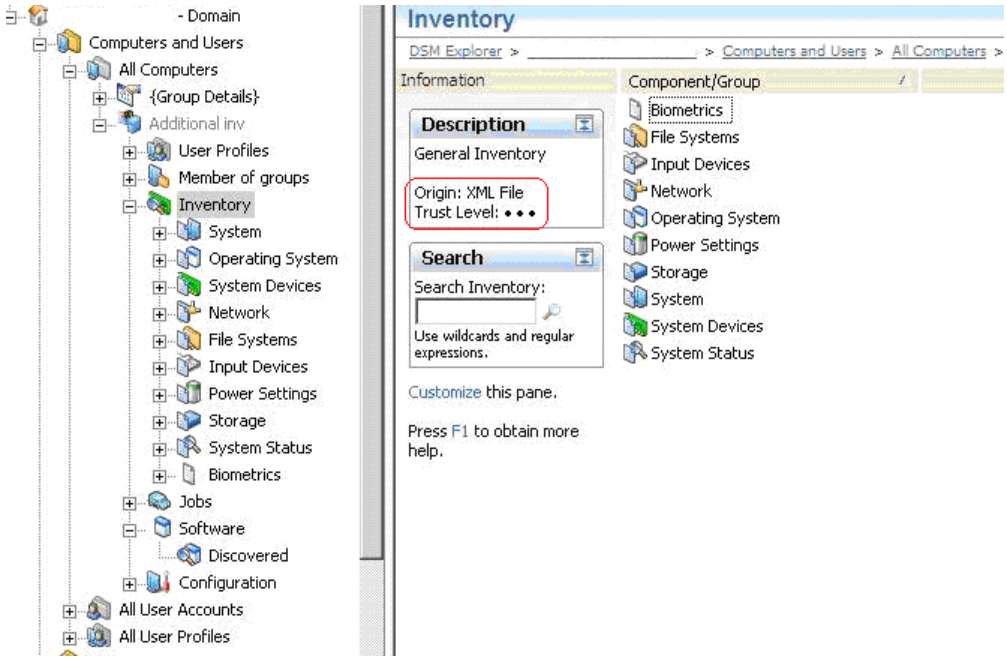
- The computer group pane



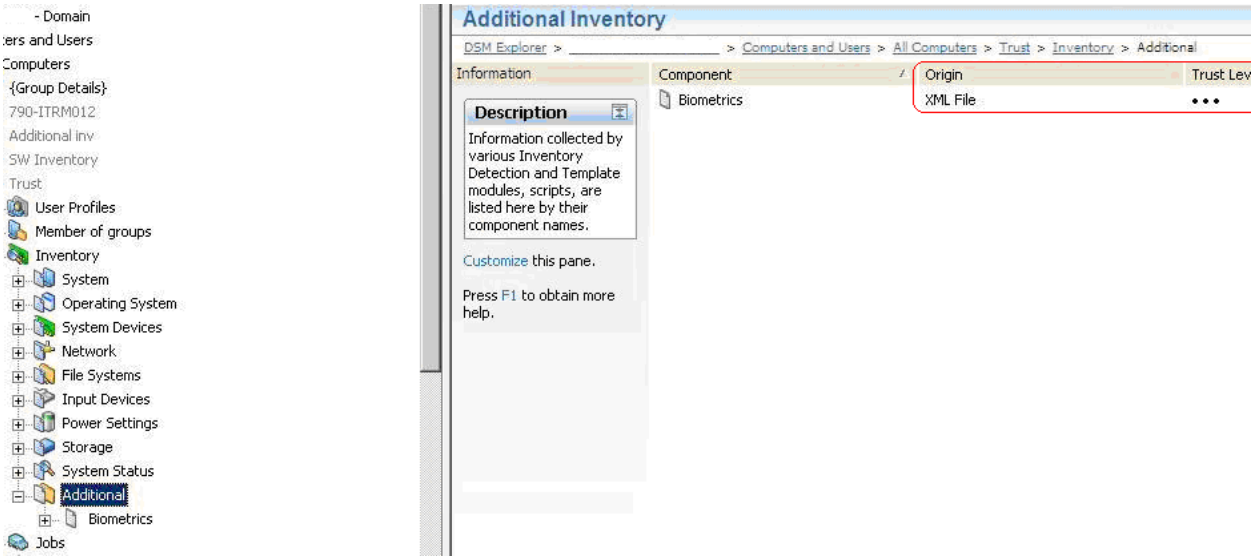
Origin and Trust Level of the Hardware Inventory

You can view the origin and trust level of the hardware inventory from the following locations in the DSM Explorer:

- The Information pane of the Inventory folder. You can select any inventory item to see its origin and trust level in the information pane, as shown in the following illustration:

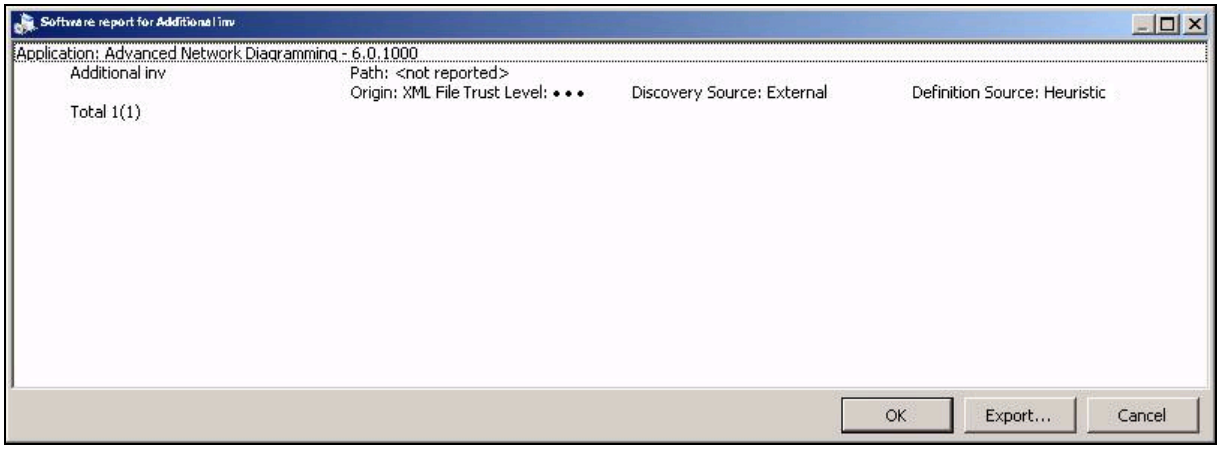


- The Origin and Trust Level columns in the right pane of the additional inventory folders. These columns are hidden by default. You can enable them by right-clicking the column header and selecting Origin and Trust Level, as shown in the following illustration:



Origin and Trust Level of the Software Inventory

You can view the origin and trust level of the software inventory by right-clicking the software in the Discovered Software pane and selecting Details. This opens the Software report displaying the origin and trust level of the selected software, as shown in the following illustration:

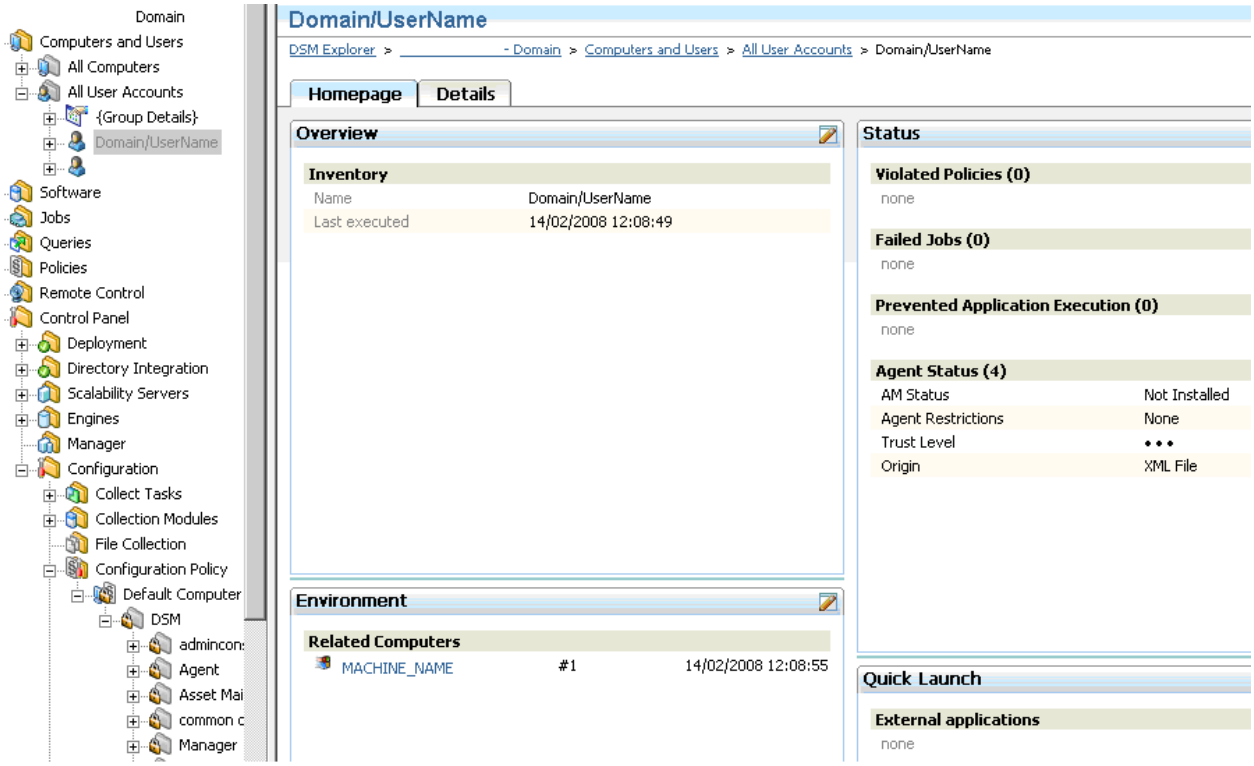


Note: You can also select multiple software items in the Discovered Software pane.

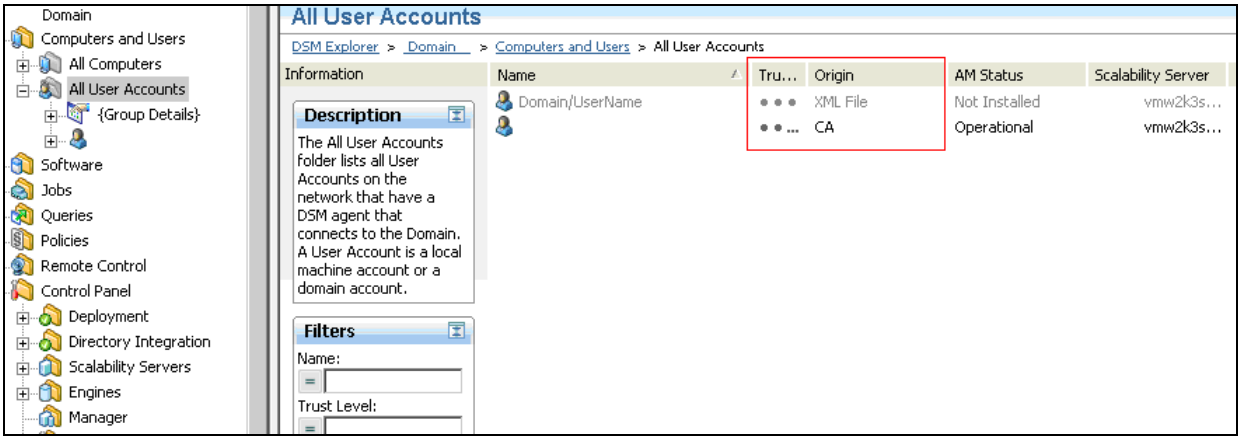
Origin and Trust Level of a User

You can view the origin and trust level of a user from the following locations in the DSM Explorer:

- The home page of the user; the Trust Level and Origin appear in the Status pane, as shown in the following illustrations:



- The All User Accounts pane.



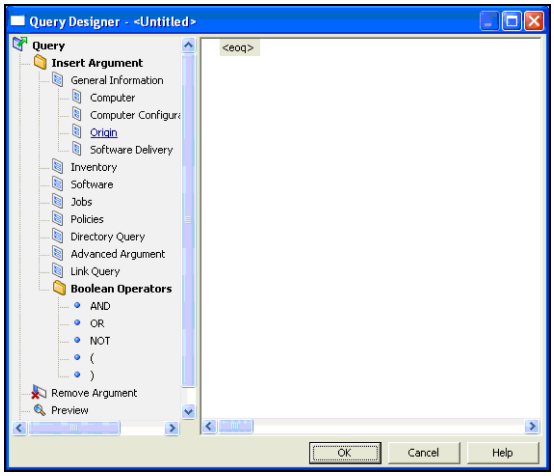
Query Designer Dialog

The Query Designer dialog lets you query the origin and trust level of an asset.

Clicking Origin opens the Select Field dialog where you specify the origin and trust level parameters for querying.

Note: You cannot create queries with User Trust Level and Origin in this release.

The following illustration shows the Origin category under Insert Argument, General Information.



Creating the Inventory File

You can create the inventory file to be processed by the Asset Collector.

Inventory File Format

The Asset Collector attempts to process any file with the extension .xiu and .xis files.

- .xiu = xml inventory unsigned files.
- .xis = xml inventory signed files.

Important! These files are still text files containing xml.

The invsign tool will also only accept .xiu files for signing, and .xis files for unsigning.

The inventory file has a general header section, hardware section, and software section, as shown in the following example:

```
<asset translator="ACBsFmt" version="1_0">
<asset>
        <general>    </general>
        <hardware>   </hardware>
        <software>   </software>
        <user>
                <general>    </general>
                <hardware>   </hardware>
                <software>   </software>
        </user>
</asset>
```

As shown in this layout, the xiu file can also contain an optional <user> section, which itself can contain hardware and software sections.

Important! The first line tells the Asset Collector how to parse this file and must not be omitted.

General Section

The general section contains the entries for the inventory item. The items allowed in the header section are determined by the type of item being described. Some header item should only be provided in an <asset> <general> section, others only in <user> <general> sections. There are also some attributes that can be provided in either a user or asset general section.

A user can be delivered alone or with a computer. If a user is delivered with a computer, the user is linked to the computer. This relationship is visible in the DSM Explorer.

The following table lists the general section tags:

Name	Description	Header
vendor	string optional	asset
serial_number	string optional	asset

Name	Description	Header
asset_tag	string optional	asset
host_name	not fully qualified, required	asset
host_key	string optional key will be generated if not supplied	asset
class_id	Platform name as a string or a number **Note:** See the detailed list of platform names in the "Platform Names and IDs" appendix.	asset
Network_id	Optional **Note:** Supports multiple network interfaces in the registration section of the XIU file.	asset
default_address	required	asset
default_mac	required	asset
default_hostname	required	asset
default_subnet_mask	optional	asset
collect_time	time in seconds, optional	asset
trustlevel	integer between 1 and 5, optional	asset
origin	string optional	asset
fileid	Max length 36	user
URI	required	user
user_name	required	user
domain_name	N/A	user
previous_uri	N/A	user
r_usage_list	N/A	user
r_name	N/A	user
r_proc_os_id	N/A	user

Name	Description	Header
r_ip_address	N/A	user
user_item_1	N/A	common
user_item_2	N/A	common
user_item_3	N/A	common
user_item_4	N/A	common
r_computer	N/A	common
collect_time	N/A	common
trustlevel	Int 1 to 5	common
origin	String 64 char max	common

Note: If the MAC address is not available, you can specify the UUID of the asset.

Example: User Only

The format for a user only in the file is as follows:

```
<asset>
    <user>
        <general>     </general>
        <hardware>    </hardware>
        <software>    </software>
    </user>
</asset>
```

The hardware and software section are optional in both the asset and user sections of the file.

Example: User and Asset

The format for a user delivered with an asset is as follows:

```
<asset>
    <general>     </general>
    <hardware>    </hardware>
    <software>    </software>
    <user>
        <general>       </general>
        <hardware></hardware>
        <software>      </software>
    </user>
</asset>
```

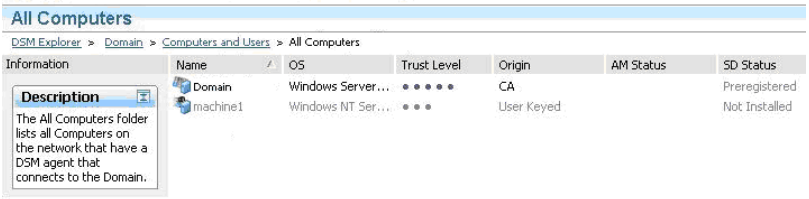
Example: General section

The following is a code example for the General Section.

```
<?xml version="1.0" encoding="utf-8" ?>
<asset translator="ACBsFmt" version="1_0">
    <general>
        <host_name>machine1</host_name>
        <default_hostname>machine1.domain.com</default_hostname>
        <default_address>150.120.13.119</default_address>
        <default_mac>00:0D:56:CB:C1:ED</default_mac>
        <class_id>21</class_id>
        <network_id>
                        <mac_address>mac</mac_address>
                        <ip_address>ip</ip_address>
                        <dns_name>name</dns_name>
                        <subnet_mask></subnet_mask>

                </network_id>
        <host_key>70669be2-1fed-4201-9777-d7137486dcab</host_key>
        <origin>User Keyed</origin>
    </general>

</asset>
```

Note: Default_hostname, default_address, default_subnet_mask are the required entries. The network_id section is optional. You can include multiple network_id sections. The default_hostname, default_subnet_mask, default_mac, are used in the default_XXX fields, in the common_asset_report. Additional network_id sections are populated into the registration message.

The following illustration shows how the general section inventory looks when viewed in the DSM Explorer.



In the preceding illustration, note the following:

- Machine1 is unavailable because it has been reported through the Asset Collector, and so is not a fully functional agent.
- As the inventory file in the preceding example did not include the *trustlevel* tag, the default trust level 3 is assigned to machine1.
- The Origin column of machine1 displays User Keyed as given in the XML file in the preceding example.

Hardware Section

The Hardware section contains the entries for General Inventory and Additional Inventory. The General Inventory section can have only those inventory groups and attributes that are collected by the CA ITCM agents. If you want to add any user-defined groups and attributes, you must create them outside the GeneralInventory group tag. The user-defined groups and attributes appear under the Inventory, Additional folder in the DSM Explorer.

The hardware section has the following tags:

group name

Defines the group name under which the inventory items are created.

attribute name

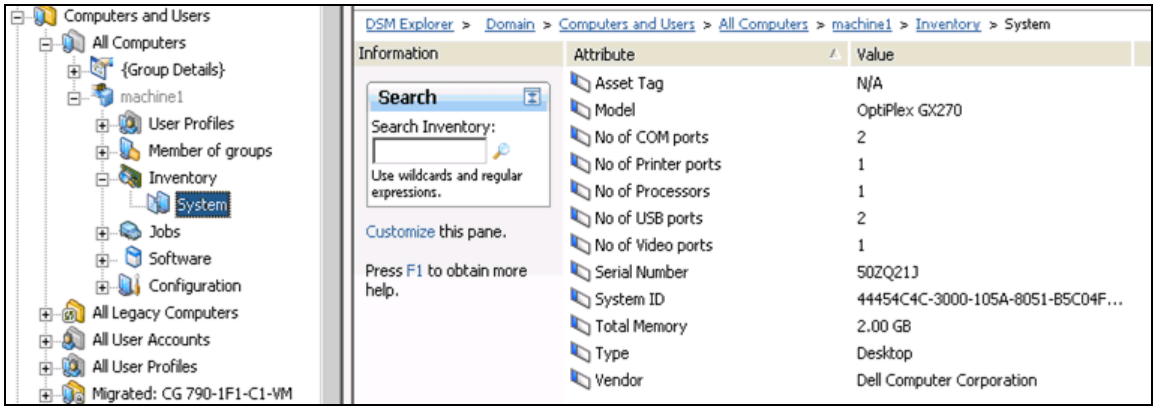
Defines the name, type, and subtype of the inventory item.

Example: Hardware Section (Attribute and Value)

The following code is an example of the hardware section.

```
<hardware>
<group name="GeneralInventory">
    <group name="System">
        <attribute name="Total Memory" type="int64"
        subtype="byte">2147483648</attribute>
        <attribute name="Model" type="string">OptiPlex GX270</attribute>
        <attribute name="Type" type="string">Desktop</attribute>
        <attribute name="No of Processors" type="int32"
        subtype="normal">1</attribute>
        <attribute name="No of COM ports" type="int32"
        subtype="normal">2</attribute>
        <attribute name="No of Printer ports" type="int32"
        subtype="normal">1</attribute>
        <attribute name="No of USB ports" type="int32"
        subtype="normal">2</attribute>
        <attribute name="No of Video ports" type="int32"
        subtype="normal">1</attribute>
        <attribute name="Vendor" type="string">Dell Computer
        Corporation</attribute>
        <attribute name="Serial Number" type="string">50ZQ21J</attribute>
        <attribute name="Asset Tag" type="string">N/A</attribute>
        <attribute name="System ID"
        type="string">44454C4C-3000-105A-8051-B5C04F32314A</attribute>
    </group>
</group>

</hardware>
```

The following illustration shows the GUI results of the above code.

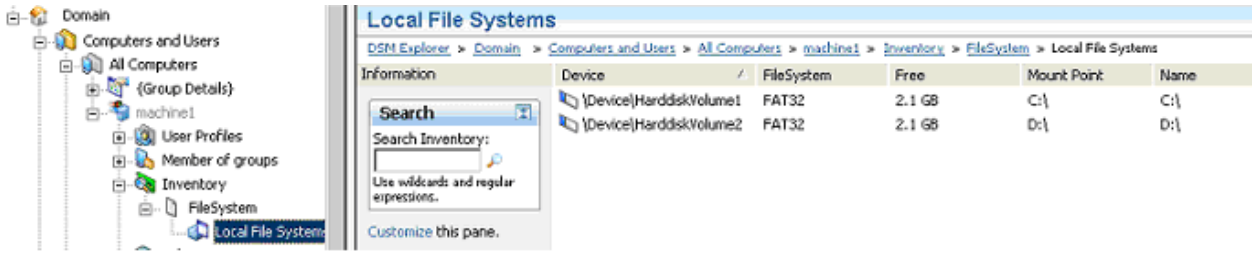


Example: Hardware Inventory File (Table Format)

The following code shows an example of the hardware inventory file.

```
<hardware>
    <group name="GeneralInventory">
    <group name="FileSystem">
    <group name="LocalFileSystem" tableid="0001">
    <attribute name="Device" type="string">\Device\HarddiskVolume1</attribute>
        <attribute name="FileSystem" type="string">FAT32</attribute>
        <attribute name="Free" type="int64" subtype="G">2230124544</attribute>
        <attribute name="Mount Point" type="string">C:\</attribute>
        <attribute name="Name" type="string">C:\</attribute>
        <attribute name="Serial Number" type="string">5872-D416</attribute>
        <attribute name="Size" type="int64" subtype="G">3784294400</attribute>
        <attribute name="Type" type="string">Fixed</attribute>
        <attribute name="Used" type="int64" subtype="G">1554169856</attribute>
        <attribute name="Used %" type="int32">20</attribute>
        <attribute name="Volume Label" type="string">C</attribute>
    </group>

    <group name="LocalFileSystem" tableid="0002">
        <attribute name="Device" type="string">\Device\HarddiskVolume2</attribute>
        <attribute name="FileSystem" type="string">FAT32</attribute>
        <attribute name="Free" type="int64" subtype="G">2230124544</attribute>
        <attribute name="Mount Point" type="string">D:\</attribute>
        <attribute name="Name" type="string">D:\</attribute>
        <attribute name="Serial Number" type="string">5872-D423</attribute>
        <attribute name="Size" type="int64" subtype="G">43784794400</attribute>
        <attribute name="Type" type="string">Fixed</attribute>
        <attribute name="Used" type="int64" subtype="G">41554269856</attribute>
        <attribute name="Used %" type="int32">89</attribute>
        <attribute name="Volume Label" type="string">D</attribute>
    </group>
    </group>
    </group>
</hardware>
```

The following illustration shows the GUI results of the above code.



General Inventory Groups and Attributes

Within Asset Inventory, each group in the inventory file is represented by a group tag. The top-most group name can either be GeneralInventory or AdditionalInformation. If you want a particular inventory group to be under the GeneralInventory group, you must use the group names given in the following list. Groups created outside the GeneralInventory group are added under the Additional node and will not have any icons associated with them in the DSM Explorer. User inventory can use any string for group names, although note that User inventory groups are all represented by the same icon.

Following is the list of valid group names for use in an Assets Inventory under the GeneralInventory group:

- GeneralInventory
- System
- Mainboard
- Processors
- OperatingSystem
- Memory
- Keyboard
- SoundAdapters
- Monitor
- DisplayAdapter
- NetworkAdapters
- Network
- SystemBios
- IOPorts
- Country
- Identification
- TCPIP
- Controllers
- Disk
- HostInformation
- CronQueue
- Status
- FileSystems
- FixedDrives

- SwapDevices
- LocalFileSystem
- SystemSlots
- SystemDevices
- FloppyControllers
- DisplaySettings
- Installation
- RegionalSettings
- SystemUpdates
- Network
- IDEControllers
- USBControllers
- SCSIControllers
- IOPorts
- VideoAdapters
- SystemBios
- DNS
- WINS
- Storage
- FixedDrives
- FloppyDrives
- CDROMDrives
- Partitions
- LogicalVolumes
- InputDevices
- PointingDevices
- PowerSettings
- PowerPolicy
- ACAdapter
- SystemStatus

User Defined Group

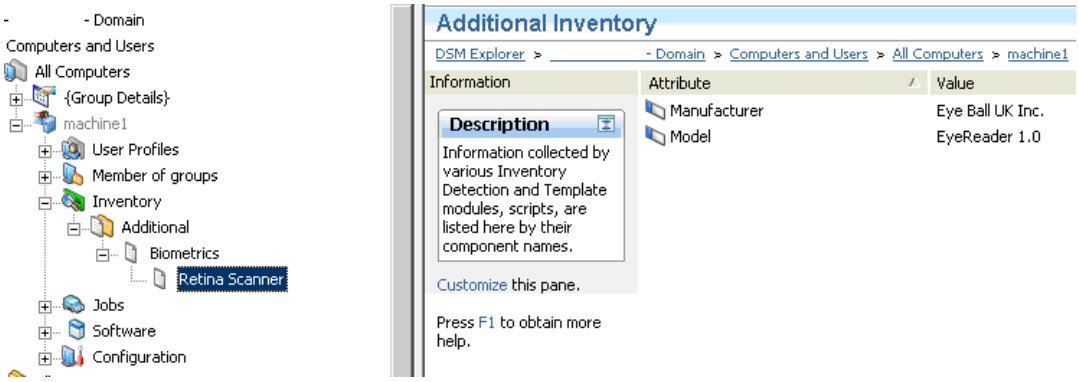
This section lets you create a user-defined group containing various attributes, under the Additional folder of the machine inventory:

Example: User-Defined Group

The following code shows an example of a user-defined group.

```
<hardware>

        <group name="Biometrics">

        <group name="Retina Scanner">
                <attribute name="Manufacturer" type="string">Eye Ball UK
    Inc.</attribute>
                <attribute name="Model" type="string">EyeReader 1.0</attribute>

        </group>

        </group>
</hardware>
```

The following illustration shows how the user-defined hardware inventory file section looks when viewed in the DSM Explorer.



Types and Subtypes

The following table includes the valid types and subtypes that you can specify for the attributes:

Note: The subtypes determine how a value is displayed in the GUI.

Type	Subtype	Description
Boolean		Boolean values can be displayed based on the following subtypes:
	TrueFalse	True or False
	YesNo	Yes or No
	OnOff	On or Off
	SupportedUnsupported	Supported or Not Supported
	ActiveNotactive	Active or Not Active
	OkError	OK or Error
	PresentNotpresent	Present or Not Present
int32 & int64		Numerical values can be displayed based on the following subtypes:
	separation	Thousands separated, that is, 1,000,000
	normal	No separation
	K	Number is divided by 1024 before display
	M	Number is divided by 1024^2 before display
	G	Number is divided by 1024^3 before display
	T	Number is divided by 1024^4 before display
	kilo	Divided by 1000 before display
	mega	Divided by 1e6 before display
	giga	Divided by 1e9 before display
	milli	Multiplied by 1e3 before display
	micro	Multiplied by 1e6 before display
	nano	Multiplied by 1e9 before display

Type	Subtype	Description
	hex	Numbers shown as hex
	time	Displayed as a date-time
	timeInterval	Displayed as a duration
	bytes	GUI decides to display as KB, MB, GB or TB
Float		Float values can be displayed based on the following subtypes:
	Auto	Auto format
	placesXX	Show to XX decimal places
String		No subtypes for string

Add New Groups

You can add groups under the GeneralInventory and AdditionalInventory tags in the Hardware section and group the related inventory attributes.

The syntax for a creating a group and an attribute is as follows:

```
<group name="user defined name">
    <attribute name="attribute name"
        type="type"
        subtype="subtype">
        value
    </attribute>
</group>
```

Note: Multiple attributes can be contained in a group, and a group can contain multiple groups.

Software Section

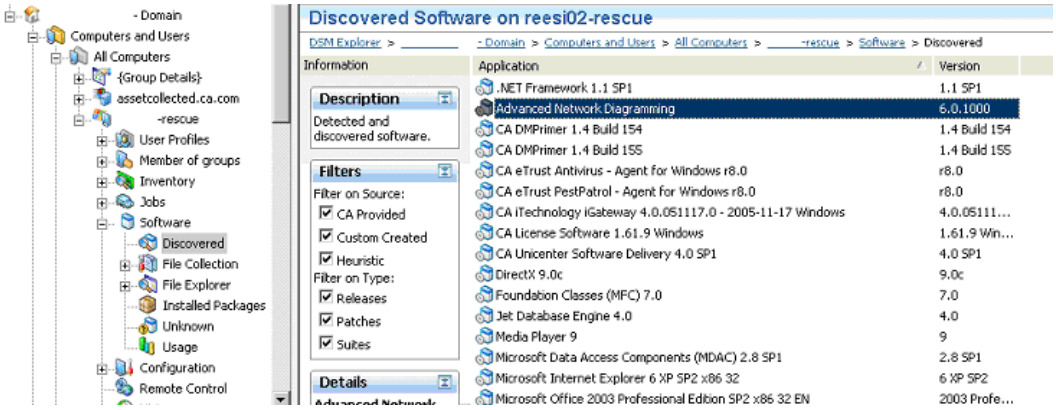
The Software section contains the entries for software inventory.

The software inventory file follows a simpler format, as groups are not required.

Note: The values GUID, Path and Filename are not mandatory.

```
<software>
    <package name="Advanced Network Diagramming">
        <attribute name="Ver">6.0.1000</attribute>
        <attribute name="Pub">Visio Corporation</attribute>
        <attribute name="Method">msi</attribute>
        <attribute name="GUID">{325C4969-4808-4A87-9547-F58620C444CA}</attribute>
        <attribute name="Path">C:\Program Files\ANP</attribute>
        <attribute name="Filename">ANP.exe</attribute>
    </package>

</software>
```

The following illustration shows how the software inventory file section looks when viewed in the DSM Explorer.



Sample XIU File

The Asset Collector includes a sample XIU file with the valid group and attributes tags. You can modify this XIU file to include your inventory values.

The sample XIU file, SampleInventoryFile.xiu, is available in the *installation path*\bin folder.

Restrictions and Limitations

This section describes the restrictions and limitations in the current version of the Asset Collector.

Supported Platforms

The Asset Collector is supported on all the same platforms as the CA ITCM Release 12.8 scalability server, with the exception of Linux\UNIX.

DSM Reporter

User trust level and origin is not available in the DSM Reporter in this release.

Query Design

It is not possible to create queries with user trust level and origin in this release.

WAC

Query and search using user trust level and origin is not available.

Agent Bridge

The Agent Bridge is a CA ITCM enhancement that provides backwards compatibility for extended and legacy agents based on the current version of CA ITCM. You can gather information obtained from mobile agents and then distribute jobs, modules, templates, and configurations to those agents.

Note: The Agent Bridge is supported on Microsoft Windows only.

Note: The agent platforms supported by the Agent Bridge in this release are Windows Mobile 6.1 (ARM-based, including StrongARM, XScale), Windows Mobile 6 (Classic, Standard, Professional) (ARM-based, including StrongARM, XScale, TI OMAP), and Windows Mobile 5 (ARM-based, including StrongARM, XScale).

Major Components

The Agent Bridge consists of the following components:

- Common legacy server component and its process components

  Handles communication for the Agent Bridge, configuration, and agent registration.

- Asset Management components

  Provide server support for extended and legacy asset management agents.

- Software Delivery components

  Provide server support for extended and legacy software delivery agents.

Note: Asset management and software delivery legacy agents installed on the same machine should point to the same CA ITCM Release 12.8 scalability server with Agent Bridge enabled.

UUID Generator for Agent Bridge

A UUID generator allows Agent Bridge to generate an agent UUID for a legacy agent if one has not been reported. Additionally, a new configuration policy has been added to control the UUID generation behavior.

Note: You should always use the latest agents possible for your operating environment. To check the certification status of a legacy agent, see the appropriate Compatibility Matrix available on the CA Support Online web site, http://support.ca.com.

Limitations

As the current release of CA IT Client Manager is a new generation of asset management, software delivery, and remote control, there are certain new features and functions that were not in the previous versions of these components and, therefore, will not work with legacy agents. Agent Bridge support does *not* include the following CA ITCM Release 12.8 DSM features:

- Asset job checks
- Performance module extraction
- Configuration policies

  Note: The Configuration Policy tree node is suppressed for legacy agents, and the Activate Job Check option is likewise disabled for legacy agents.
- Instant diagnostics
- DMDeploy functionality
- Software file scan based on signatures

Due to the changes in technology in this release of CA ITCM, there are also some existing Unicenter Asset Management 4.0 and Unicenter Software Delivery 4.0 features that are *not* supported on legacy agents in this release of CA ITCM using Agent Bridge. These unsupported features are as follows:

- Online metering for asset management legacy agents

  Note: Offline metering, however, is supported.
- Software file scan based on application definitions
- Docking devices based on Unicenter Asset Management 4.0 and Unicenter Software Delivery 4.0 legacy agents
- Software Catalog for software delivery legacy agents
- Auto-installations of software delivery legacy agents

Supporting Windows Mobile agents, the Agent Bridge reads formatted legacy data files and translates them into CA ITCM Release 12.8 formatted data inventory files to be collected by the DSM engine. It translates as much data as possible; and as a result, duplicate subnodes like General Inventory may sometimes be displayed in the DSM Explorer due to slight differences between various operating environments and different versions of legacy agents. Also, certain properties like Serial Number, Asset Tag, and Disc Serial Number will be empty under the Computer Properties node for some computers because these assets do not get collected in legacy agents.

Known Issues

Currently, the following issues have been identified for Agent Bridge.

Activate Job Check

The Activate Job Check on the software delivery legacy agent might fail in the following situation:

1. During package deployment the option, Jobs will be triggered by scalability server, is not selected, and
2. Once the job container is created and is in the Active state, you select either the Activate Job Check option from the DSM Explorer or run sdacmd jobcheck on the agent machine.

The result is that the status of the job remains in the Active state.

The workaround for this problem is to run sdjexec.exe on a Unicenter Software Delivery 4.0 agent machine. The status of the job will then be displayed as successful in the DSM Explorer.

Windows Mobile 5.0 Devices

When a Unicenter Asset Management 4.0 SP1 C2 agent is installed first, followed by a Unicenter Software Delivery 4.0 SP1 C3 agent, on a Windows Mobile 5.0 device pointing to Agent Bridge, the version number of the Unicenter Asset Management agent shown in the DSM Explorer might be incorrect.

Also, when a Unicenter Software Delivery 4.0 SP1 C3 agent is installed first, followed by a Unicenter Asset Management 4.0 SP1 C2 agent, on a Windows Mobile 5.0 device pointing to Agent Bridge, the Unicenter Asset Management 4.0 SP1 C2 agent might not get reported.

Note: See the "Troubleshooting" chapter for more diagnostic tips.

Agent Registration

The CA ITCM agent registers with the MDB by providing the following information:

- Host name
- Host Universal Unique Identifier (UUID)
- Operating system class ID
- IP address
- MAC address
- Usage list string

With Agent Bridge, the host UUID is provided by the legacy agent, and the legacy agent is registered to the CA ITCM system using the Agent Bridge. The agent registration method used by CA ITCM and Agent Bridge maps the legacy information to that in CA ITCM. For example, an asset management 8-byte file ID is mapped to a CA ITCM formatted, 32-byte alphanumeric host UUID. The host UUID that is generated for a legacy agent is common to both asset management and asset management agents, and is stored in the common usage file.

If a legacy agent does not provide a unique host UUID and the UUID generator is not enabled, it is not registered by the Agent Bridge, and the agent is treated as an invalid agent.

If the UUID generator is enabled, the correct procedures for registering asset management and software delivery legacy agents are as follows:

To register legacy agents, some of which may not report their own UUIDs

- If both the asset management and software delivery agents do not report host UUIDs, then you can register the asset management and software delivery agents in any order. However, you need to give time between the registrations of the two agents, typically, at least five (5) minutes apart.
- If the asset management agent does not report a host UUID but the software delivery agent does, then you should always register the software delivery agent first. Then after five (5) minutes register the asset management agent.
- If the software delivery agent does not report a host UUID but the asset management agent does, then you should always register the asset management agent first. Then after five (5) minutes register the software delivery agent.

In general, the majority of Unicenter Asset Management 4.0 agents and Unicenter Software Delivery 4.0 agents report host UUIDs when registering with the server.

Asset Management Agents

The registration of all legacy Unicenter Asset Management 4.0 agents is supported by Agent Bridge in CA ITCM.

Mobile PDA agents require the application of patch T434066 before they can work with Agent Bridge. This patch can also be obtained from CA Online Support at http://ca.com/support.

Software Delivery Agents

The registration of all legacy Unicenter Software Delivery 4.0 agents is supported by the agent registration method used by CA ITCM and Agent Bridge.

Migration Notes

When migrating legacy agents to the current release of CA ITCM using Agent Bridge, keep in mind the following limitations:

- After a migrated agent has performed a "reinstall after crash" (RAC), a new migration for the same agent creates a new unit rather than updating the existing agent.
- If you already have a migrated software delivery legacy agent in your CA ITCM Release 12.8 environment, then installing a Unicenter Asset Management 4.0 legacy agent or CA ITCM Release 12.8 asset management agent afterwards results in the status of the migrated legacy software delivery agent being overwritten. The SD status will be changed from "Locked by Migration" to "Not Installed." Therefore, do not use this method to migrate and register legacy agent data. Instead, use the DSM migration tool.

Note: For more information about the DSM migration tool, see the *Implementation Guide*.

AM Agent Bridge

The Agent Bridge legacy sector server support for asset management agents (AM Agent Bridge) is a self-contained module, amLrss.exe, that is part of the CA ITCM asset management scalability server. It is configurable by policy, and can be started and stopped as part of the CA ITCM Common Application Framework (CAF) service.

AM Agent Bridge also includes the following modules that are part of the asset management scalability server:

- Mobile sector server (amss.exe)

  Controls data communication between the legacy sector server and mobile PDA agents. The process is started upon the start up of the amLrss legacy sector server, and it is stopped if amLrss is stopped.

- Bridge translation module (amBridge.exe)

  Translates the data between the legacy sector server database and the CA ITCM Release 12.8 sector database. The process is started by amLrss at a scheduled interval, which is configurable by policy.

Supported Operating Environments and Asset Management Agents

The CA ITCM media includes an agent for Windows Mobile 5.0, 6.0, and 6.1. This agent requires the use of the legacy Agent Bridge. No other agents are supported with the legacy Agent Bridge.

Note: You should always use the latest asset management agent possible for your operating environment. To check for the latest version, see the asset management Compatibility Matrix available on the CA Support Online web site, http://support.ca.com.

Local and Domain Users

If the Enable Legacy User Support configuration policy is enabled, Agent Bridge handles legacy user accounts the same way as they were handled in Unicenter Asset Management 4.0, that is, there is no difference between local users and domain users, and no domain information about the user's account is passed to CA ITCM Release 12.8. However, the prefix LEGACY_USER_ACCOUNT is used as the Domain name for all asset management legacy users' accounts.

Configuration

You can configure the AM Agent Bridge for legacy agents using configuration policies. The Asset Management policy group folder and policies can be found the Control Panel, Configuration, Configuration Policy, Default Computer Policy, DSM, Scalability Server subnode.

Additionally, the Backwards Compatibility folder and policy can be found under the Configuration Policy, Default Computer Policy, DSM, Scalability Server\Common server subnode.

Note: For detailed information about AM Agent Bridge configuration policies, see the Configuration Policy section of the *DSM Explorer Help*.

Event Messages for AM Agent Bridge

Following are the AM Agent Bridge event messages:

- "The configuration of AM Agent Bridge has changed."

  Note: Whenever a configuration policy is modified by the user, event messages are sent to both the OS event log and the DSM event log.
- "AM Agent Bridge has started successfully."
- "AM Agent Bridge has stopped successfully."
- "AM Agent Bridge failed to start."
- "AM Agent Bridge failed to stop."

How to Collect Legacy Data

Software and hardware inventory data is collected using inventory modules and collect tasks.

The following is the approach for collecting the inventory data of the legacy agents involved:

- Unicenter Asset Management 4.0 Extended and Legacy Agents

  For Unicenter Asset Management 4.0 extended and legacy agents, existing CA ITCM Release 12.8 inventory modules in the DSM Explorer can be used.

Inventory Detection Modules

The contents of the legacy agent inventory files can be collected using the existing CA ITCM Release 12.8 engine inventory detection modules. The results are displayed in the DSM Explorer under the All Computers, *computername*, Inventory, Additional node. The Additional node is *dynamic* and may contain any or all the following subnodes depending on the type of legacy agent registered: General Inventory, Hardware, Serial Number and Model (WMI), Services, Software Inventory, Software Delivery Information, User Inventory, User Information, User Template, and WBEM inventory.

The Agent Bridge reads Unicenter Asset Management 4.0 formatted data files and translates them into CA ITCM Release 12.8 formatted data inventory files to be collected by the DSM engine.

Using Existing Inventory Modules

The following table lists the detection modules that are listed for both CA ITCM Release 12.8 and Unicenter Asset Management 4.0 systems. The CA ITCM Release 12.8 module names are translated to Unicenter Asset Management 4.0 module names using an Agent Bridge translation module:

Inventory Module Name	Purpose	OS Name	CA ITCM Release 12.8 Name	UAM 4.0 Name
General Inventory Advanced (4.0)		Windows 95/98/Me	amadvinv9x.exe	IG40W95.EXE
		Windows CE	N/A	IG40WCE.EXE

By default, the CA ITCM Release 12.8 General Inventory module is translated to Unicenter Asset Management 4.0 formatting.

There is a new inventory module defined in CA ITCM Release 12.8 that is *not* available for legacy agents:

Inventory Module Name	Purpose	OS Name	CA ITCM Release 12.8 Name	UAM 4.0 Name
Performance Inventory	OS-specific	Windows XP/2003/Vista/2008	pcmtouam1.exe	N/A
		UNIX	pcmtouam	N/A

Creating Inventory Modules

Inventory detection modules for Unicenter Asset Management 4.0 legacy agents can be created using the DSM Explorer and distributed to the legacy agents.

In the following table, the inventory modules are specific only to the legacy agents; they are not applicable for CA ITCM Release 12.8 DSM agents.

Inventory Module Name	Purpose	OS Name	UAM 4.0/3.2 Name
General Inventory (3.2 Agents)	OS-specific	Windows 95/98/Me	IGENW95.EXE
		Windows CE	IGENWCE.EXE

Offline Metering

Offline metering, or *offline software usage* as it is called in CA ITCM Release 12.8, can be reported by the Agent Bridge, but only for legacy agents. This feature allows CA ITCM to monitor applications with existing software packages and applications or user-defined software.

For user-defined software, the offline metering feature is the same as that in Unicenter Asset Management 4.0, wherein you define a new software package, including version and module information, and usage is passed forwards to the legacy agent to be monitored. The metering inventory file is then collected by the Agent Bridge, and the usage of the user-defined software can be viewed in the DSM Explorer.

For existing software packages and applications that are based on signature scan information in CA ITCM Release 12.8, the legacy agent monitors software without matching the version information. This means that if you select a software package with a specific version number from the software list in the DSM Explorer, software usage detected by the legacy agent will not exactly match the versions as defined by the Software Usage module that you have selected.

For example, if you select Adobe Acrobat 5.0 x86 32 EN for offline software usage monitoring, the legacy agent returns the usage data of *any version* of Adobe Acrobat regardless of the version numbers provided. This is because the legacy agent lacks version detection functionality.

SD Agent Bridge

For information on the software delivery agents (SD Agent Bridge), see the *Software Delivery Administration Guide*.

Integrating Intel AMT with CA ITCM

The Intel® Active Management Technology (Intel AMT) functionality for CA ITCM lets you manage Intel AMT-enabled assets from within the DSM Explorer user interface. It deals with the integration and usage of the Intel AMT assets in the DSM Explorer. Using this functionality, you can perform the following tasks:

- Provide access to the Intel AMT assets by creating integration points in the DSM Explorer user interface.
- Provide policy-based management for the Intel AMT assets.
- Use various Intel AMT commands to control and perform different actions on the Intel AMT assets.

Configure the Intel AMT Asset

Before running any Intel AMT command on the Intel AMT asset, you need to configure the Intel AMT asset. Configuring the Intel AMT asset requires you to provision it from the DSM domain manager so that the asset is ready to run the Intel AMT commands.

Note: If you do not provision the asset from the DSM domain manager, the asset will not run any Intel AMT command.

A provisioning toolkit is installed with the DSM domain manager in C:\Program Files\CA\SC\ConstantAccess\caprovision.exe. Use the caprovision.exe file from the DSM domain manager to provision the Intel AMT asset.

Run caprovision.exe

Run caprovision.exe (caprovision provision *hostname adminusername adminpasswd userid userpasswd*) before you perform any action on the Intel AMT asset.

To run the caprovision.exe file, enter the following command:

`caprovision -c command -h hostname -os hostOSName -u adminUser -p adminPassword [-usern userName] [-userP userPassword] [-certn certificateName] [-certP certificatePassword] [-i]`

-c *command*

Defines the operation to be performed of the Intel AMT asset. Supported operations/commands are provision, unprovision, getprovisioneddata, isdeviceprovisioned. Mandatory only if -i option is not used. The provision option enables the device to work with various commands; that is, it sets the user, password, and tag in the AMT device memory. The unprovision option removes the device from the provision state; that is, it removes the user, password, and tag from the device memory. The getprovisioneddata option lists the provision data for the named device. The isdeviceprovisioned option tells you if a named device has been provisioned.

-h *hostname*

Defines the host name of the Intel AMT asset. Mandatory only if -i option is not used.

-u *adminUser*

Defines the user name for the administrator. Mandatory only if -i option is not used.

-p *adminPassword*

Defines the password for the administrator. Mandatory only if -i option is not used.

-os *hostOSName*

Defines the name of the operating system on the Intel AMT asset. Mandatory only if -i option is not used.

-usern *userName*

Defines the name for the new user. User name is randomly generated if this option is not used and -i option is absent.

-userp *userPassword*

Defines the password for the new user. Password is randomly generated automatically if this option is not used and -i option is absent.

-i

Defines the interactive mode. It also prompts the user for the parameters which were not provided at the command line.

Validate the Intel AMT Asset Configuration

After configuring the Intel AMT asset, you can validate whether the asset has been configured properly.

To validate the Intel AMT asset configuration

1. Browse the Intel AMT asset on port number 16992 using the following URL:

   `http://<hostname>:16992`

   The login page appears.

2. Provide valid user credentials.

   If the user credentials are valid and the asset is configured properly, the Index page with details of the Intel AMT asset appears.

You can now use the DSM Explorer to perform various operations on the Intel AMT asset.

Intel AMT Status

In CA ITCM, the Intel AMT status is represented in different states for an Intel AMT asset. The following table includes all of the options available for the Intel AMT status:

State	Description
Unknown	CA ITCM currently does not know about the Intel AMT status for this asset. You must click the refresh icon to query for the actual state.

State	Description
Not Supported	Asset does not have the Intel AMT chipset, and also does not support Intel AMT.
Not Provisioned	Asset has the Intel AMT chipset; but, currently it is not provisioned to be managed through Intel AMT.
Not Enabled	Asset has the Intel AMT chipset, but management through Intel AMT is not enabled.
Enabled	Asset has the Intel AMT chipset, and is enabled to be managed through Intel AMT.
Quarantined	Asset has the Intel AMT chipset, and is currently quarantined. All network traffic except DSM is blocked.

View the Status of an Intel AMT Asset

View the status of an Intel AMT asset in the Agent Status section under the Status portlet. You can take appropriate actions after analyzing the status of the asset.

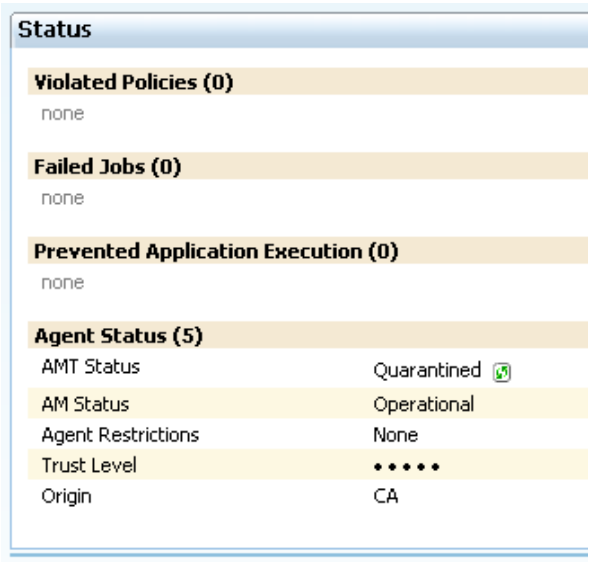
To view the status of an Intel AMT asset

1. Navigate to Domain, Computers and Users, All Computers, *Computer*.

   The Homepage tab appears.

2. Go to the Agent Status section under the Status portlet.

3. View the AMT Status value to know the status of the Intel AMT asset, as shown in the following illustration:



Note: By default, the status is Unknown. You must click the refresh icon displayed next to the AMT Status value to display the current status of the Intel AMT asset.

Browse an Intel AMT Asset

You can browse an Intel AMT asset and view the inventory information. However, only a remote (read-only) status of the asset is enabled.

To browse an Intel AMT asset

1. Navigate to Computers and Users, All Computers, *Computer*.

   The Homepage tab appears.

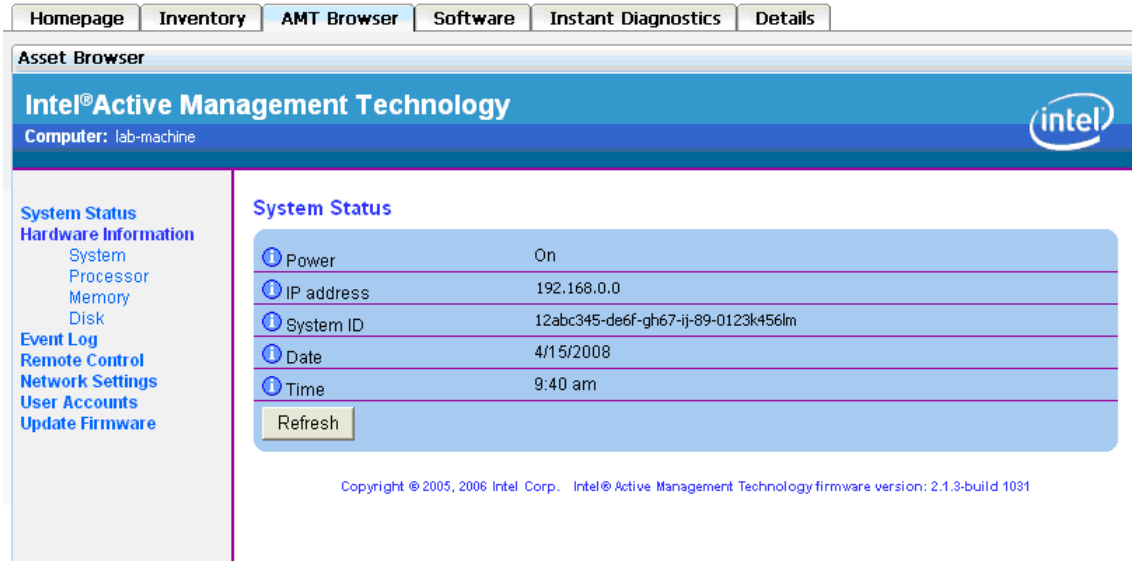
2. Click the AMT Browser tab.

   Note: This tab is available only if Intel AMT is enabled on the asset.

   The Log On page appears.

3. Click the Log On button and enter the user credentials.

   The System Status page appears displaying information such as IP address, System ID, and so on.



Note: Click any link in the left pane to view the corresponding information.

Put an Intel AMT Asset in Quarantine

When an Intel AMT-enabled asset enters the quarantine mode, a filter is applied to enforce a policy for the incoming and outgoing network traffic on the Intel AMT asset. Any Intel AMT asset that enters the quarantine mode will have only AMT and DSM ports open; therefore, no network access is allowed for any user, application, or internet application. However, DSM administrators can execute agents and perform DSM-related management of the Intel AMT asset.

To put an Intel AMT asset in quarantine

1. Navigate to Domain, Computers and Users, All computers, *Computer*.

   The Homepage tab appears.

2. Go to the Quick Launch portlet.

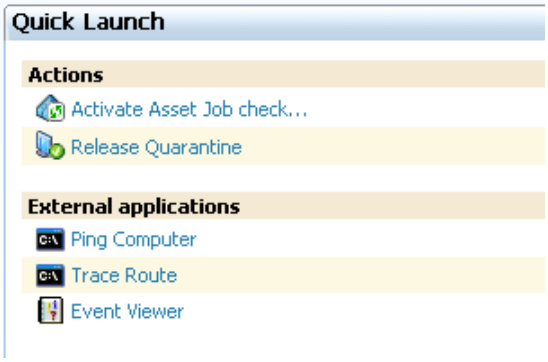
3. Click the Quarantine Asset link under the Actions section.

   Note: The Quarantine Asset link is available only if Intel AMT is enabled on the asset.

   A confirmation message appears.

4. Click Yes.

   The Intel AMT asset is successfully put under quarantine, as shown in the following illustration:

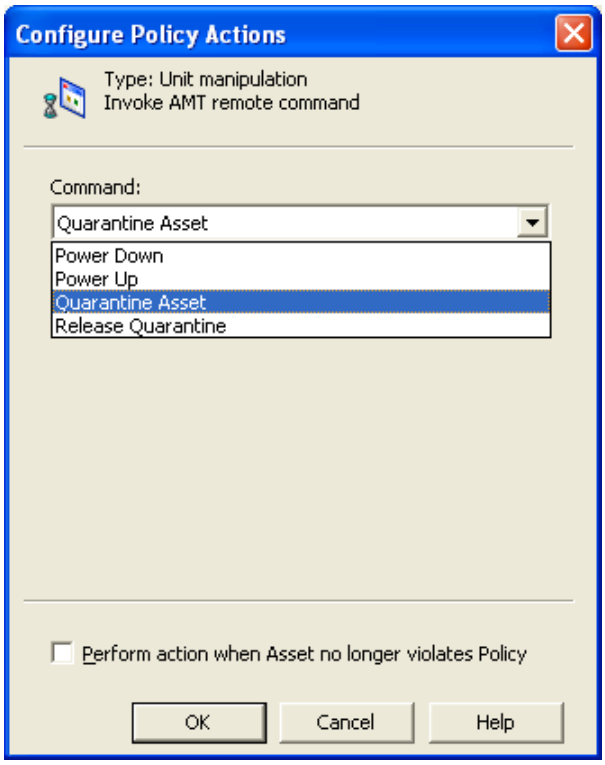


After the Intel AMT asset is successfully put under quarantine, the Quarantine Asset link changes to the Release Quarantine link. Click the Release Quarantine link whenever you want to release the Intel AMT asset from quarantine.

Use AMT Command Policy Action

You can create a policy to manage Intel AMT assets. While creating a policy, use the AMT Command policy action to run Intel AMT commands on the Intel AMT assets. Whenever this policy is violated, specified command is executed on the Intel AMT asset.

The following illustration shows the available Intel AMT remote commands:



Note: For more information on the AMT Command policy action, see the Asset Management section of the *DSM Explorer Help*.

Virtual Hosts

CA ITCM supports *platform virtualization*, which is the encapsulation of computers or operating systems wherein their physical characteristics are hidden from users and they emulate the computing platform at runtime.

In the previous release, support was provided by the Partitioned UNIX Server agent. In this release, the functionality of the Partitioned UNIX Server agent has been fully integrated with the new AM remote agent, and this agent can now be configured exclusively in the DSM Explorer to collect information for virtual hosts.

The AM remote agent runs on virtual host computers and performs the following tasks:

- Running the virtual host inventory collect task. This collect task creates an inventory file that spawns the inventory tree of the DSM Explorer.
- Registering the virtual host computers as asset management units.
- Maintaining the relationship of each virtual host computer to all the managed guest computers it contains.

The collected information shows up in the DSM Explorer inventory tree and reflects the specific properties of a virtual host. HP, IBM, Sun, and VMware ESX servers are presented as computer units. A computer unit shows up for each of the managed guest computers.

The virtual host and its associated managed guest computers are reported as related computers, and you can easily navigate between both.

The additional AM remote agent can be installed on any AIX, HP, Linux, Solaris SPARC and x86, and Windows agent machine.

Each managed guest machine is shown as a separate computer in the DSM Explorer tree view.

More information:

How to Create Virtual Host Inventory Collect Tasks (see page 184)
Viewing Remote Agents and Virtual Hosts (see page 187)
Virtual Host Inventory (see page 184)

Supported Virtualization Server Platforms

The inventory on virtual hosts supports the following UNIX platforms:

Note: The AM remote agent runs only on Windows or Linux agents.

AIX

Following is the list for platform virtualization support on AIX:

- IBM System p5 570 (AIX 5.3)
- IBM System p5 595 (AIX 5.3)
- IBM eServer pSeries p650 (AIX 5.3)
- IBM eServer pSeries p670 (AIX 5.3)
- IBM eServer pSeries p690 (AIX 5.3)

HP-UX

Following is the list for platform virtualization support on HP-UX:

- HP 9000 rp3440 (HP-UX 11.11, 11.23)
- HP 9000 rp4400 (HP-UX 11.11, 11.23)
- HP 9000 rp7410 (HP-UX 11.11, 11.23)
- HP 9000 rp7420 (HP-UX 11.11, 11.23)
- HP Integrity rx8620 (HP-UX 11.23)
- HP Integrity rx2660 (HP-UX 11.23)
- HP 9000 Superdome SD32 (HP-UX 11.11, 11.23)
- HP 9000 Superdome SD32A (HP-UX 11.11, 11.23)
- HP 9000 Superdome SD64 (HP-UX 11.11, 11.23)

Following is the list for platform virtualization support on multi-core HP-UX:

- HP 9000 rp3440
- HP 9000 rp4440
- HP Integrity rx2620
- HP Integrity rx2660

Solaris

Following is the list for platform virtualization support on Solaris:

- SUN Enterprise E10K (Solaris 8)
- SUN Fire 15K (Solaris 8, 9, 10)
- SUN Fire 4800 (Solaris 8, 9, 10)
- SUN Fire 6800 (Solaris 8, 9, 10)
- SUN Fire 6900 (Solaris 8, 9, 10)
- Sun Enterprise M5000 (Solaris 8, 9, 10)

Following is the list for platform virtualization support on multi-core Solaris:

- Sun Fire V490

VMware ESX

Following is the list for platform virtualization support on VMware ESX:

- VMware ESX 3.5
- VMware ESX 4.0
- VMware ESX/ESXi 4.0, 4.1 and ESXi 5.1
- VMware vCenter Server 4.0, 4.1 and 5.1

Citrix XenServer

Following is the list for platform virtualization support on Citrix:

- Citrix XenServer 5.0, 5.5, 5.6, 5.6 FP1, 5.6 SP2, 6.0, and 6.1

Microsoft Hyper-V Server

Following is the list for platform virtualization support on Windows platforms:

- Microsoft Hyper-V Server 2008 R2 – Standard, Enterprise and Datacenter editions
- Microsoft Hyper-V Server 2008 R2 SP1 – Standard, Enterprise and Datacenter editions
- Microsoft Hyper-V Server 2012 – Standard, Enterprise and Datacenter editions
- Stand alone Hyper-V 2008 R2

Note: For the most current list of supported platforms, see the *CA IT Client Manager Readme*.

Content Update from the CA Website

The implementation of the AM remote agent requires an up-to-date content of the MDB database from the CA website. A computer with a DSM domain manager and DSM Explorer must have access to the Internet to download the content. From the DSM Explorer, set the Software Contents Download configuration policy (Domain, Control Panel, Configuration, Configuration Policy, Default Computer Policy, DSM, Manager, Software Contents Download). The Contents Download Port policy is set to 443 (new installation) or 5250 (upgrades).

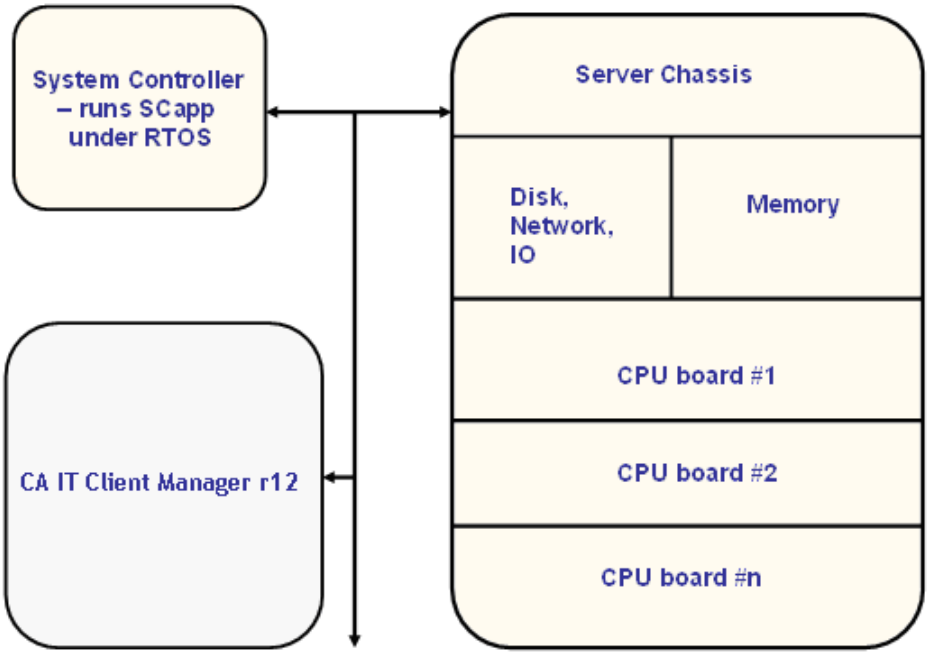
Note: For more information, see the Software Contents Download Policy Group section in the *DSM Explorer Help*. Alternatively, you can use the utility, ContentUtility.exe, to download the content. For more information on the utility, see the "Configuring Asset Management" chapter in this guide.

Solaris SPARC

On Solaris SPARC, the module produces partition information on Sun Enterprise/Sun Fire systems that support domains.

A domain is a part of a large hardware configuration usually comprising several distinct computers using a combination of CPU boards, memory, disk arrays, and other available resources. Thus, a domain can be considered as a separate encapsulated environment operating as a distinct computer. The system controller (SC) and the eXtended System Controller Facility (XSCF) manage the allocation of hardware resources for each domain. It also records and maintains the configurations and status of the domains hosted on a Sun server rack.

The following illustration represents the hardware arrangement that the AM remote agent supports for the Sun domains:



Information Collected for Each System Controller

The following table includes the information collected for each system controller:

The domains that it is hosting, which includes	The host server that it is controlling, which includes	The system controller itself, which includes
The host name for each domain	The amount of installed memory	The host name and IP address of the system controller
The IP address for each domain		The amount of memory available to it
The state (active, inactive) for each domain		The operating system revision level
The MAC address (except XSCF)		
The host UUID		

Installation

The AM remote agent is an additional part of the asset management agent installation on a UNIX server that can be partitioned.

You must install this agent on the Solaris domain and direct it to the system controller through SSH.

Important! Configure the system controller on the same network as the rest of the CA IT Client Manager installation.

Prerequisites for Accessing the M3000/M4000/M5000/M8000/M9000 eXtended System Control Facility

The SSH service on the eXtended System Control Facility (XSCF) is disabled by default. You must perform the following prerequisites for accessing the XSCF:

- Add an XSCF user account
- Create a password for an XSCF user
- Assign privileges to an XSCF user
- Enable SSH
- Reboot the XSCF
- Generate a host public key for the SSH service
- Install an SSH userpublic key

Add an XSCF User Account

To add an XSCF user account

1. Log in to the XSCF console with useradm privileges.
2. Use the "adduser username" command to add a user:

   ```
   XSCF> adduser uamuser
   ```

3. Use the "showuser" command to verify that the "uamuser" has been added.

Create a Password for an XSCF User

To create a password

1. Log in to the XSCF console with useradm privileges.
2. Use the “password username” command to set the password for the user:

   ```
   XSCF> password uamuser
   ```

3. Enter the password when prompted and then re-enter the password.

Assign Privileges to an XSCF User

To assign privileges

1. Log in to the XSCF console with useradm privileges.
2. Use the “setprivileges user privileges” command to assign privileges for the user.

   ```
   XSCF> setprivileges uamuser platadm
   ```

3. Use the "showuser -p" command to confirm the privilege of the "uamuser".

Note: The platadm privilege is required to display the version of the installed XSCF Firmware.

Enable SSH

To enable SSH, log in to the XSCF console with platadm privileges and use the “setssh”command:

```
XSCF> setssh –c enable
```

Reboot the XSCF

Rebooting the XSCF is required for the changes in the above SSH setting to take effect.

To reboot the XSCF, enter the following command from the XSCF console:

```
XSCF> rebootxscf
```

Generate a Host Public Key for the SSH Service

A host public key is required for the SSH service.

To generate a host public key

1. Log in to the XSCF console with platadm privileges.
2. Type the setssh command as follows:

   ```
   XSCF> setssh –c genhostkey
   ```

3. Generate the host key by resetting the XSCF. (See Reboot the XSCF.)

Install an SSH User Public Key

The SSH keys must be generated on the AM remote agent system and installed on the XSCF. You must perform the following prerequisites for accessing the XSCF:

1. Use the ssh-keygen command to set up the SSH keys on the AM remote agent machine. Once the SSH keys are set up, the agent uses SSH and accesses the XSCF to collect the relevant inventory information.

   Note: The SSH package is not installed by default on Solaris 8 or later; you can download it from the http://www.openssh.org/ site.

2. As a root user, run ssh-keygen on the agent machine.

   This generates two keys—a private key and a public key. These keys are stored in ~/.ssh/ id_rsa and ~/.ssh/id_rsa.pub.

3. Use setssh to add the contents of the public key file (~/.ssh/id_rsa.pub) to the intended asset management user, uamuser, on the XSCF. Then copy-and-paste the user public key, which was made in Step 2. After pressing the Enter key, press the “Ctrl” and “D” keys to complete the installation.

   ```
   XSCF> setssh –c addpubkey –u uamuser
   ```

4. Use the showssh command to confirm the uamuser public key has been installed on the XSCF:

   ```
   XSCF> showssh –c pubkey –u uamuser
   ```

   Now you can use SSH to log in to the system controller without being prompted for a password.

Prerequisites for Accessing the E4800/4900/6800/6900 System Controller

The SSH service on the system controller is disabled by default. You must perform the following prerequisites for accessing the system controller:

- Enable SSH
- Enable remote access to the system controller
- Reboot the system controller
- Validate all modifications

Enable Remote Access to the System Controller

To enable remote access to the system controller, select "ssh" as follows:

```
Connection type (ssh, telnet, none) [none]: ssh
```

Enable SSH

Use the "setupplatform -p network" command to enable SSH from the platform shell as follows:

```
schostname:SC> setupplatform -p network

Network Configuration

Is the system controller on a network? [yes]:

Use DHCP or static network settings? [static]:

Hostname [schostname]:

IP Address [xx.x.xx.xx]:

Netmask [xxx.xxx.xxx.x]:

Gateway [xx.x.xx.x]:

DNS Domain [xxxx.xxx.xxx]:

Primary DNS Server [xxx.xxx.xxx.xx]:

Secondary DNS Server [xxx.xxx.xx.x]:
```

Reboot the System Controller

Rebooting the system controller is required for the changes in the above network settings to take effect.

```
Idle connection timeout (in minutes; 0 means no timeout) [0]:
```

To reboot the system controller, enter the following command from the platform shell:

```
schostname:SC> reboot -y
```

Validate All Modifications

After rebooting the system controller, use the "showplatform" command to validate that all the modifications are implemented.

Prerequisites for Accessing the E10K/12K/15K System Controller

The SSH keys must be generated on both the system controller and its domains. You must perform the following prerequisites for accessing the system controller:

1. Use the "ssh-keygen" command to set up the SSH keys on the system controller and the domain systems. Once the SSH keys are set up, the agent uses SSH and accesses the system controller to collect the relevant inventory information.

   Note: The SSH package is not installed by default on Solaris 8 or later; you can download it from the http://www.openssh.org/ site.

2. As a root user, run ssh-keygen on the domain.

   This generates two keys—a private key and a public key. These keys are stored in ~/.ssh/ id_rsa and ~/.ssh/id_rsa.pub.

3. Add the contents of the public key file (~/.ssh/id_rsa.pub) into the intended asset management user, ~/.ssh/authorized_keys, on the system controller.

   Now you can use SSH to log in to the system controller without being prompted for a password.

Accessing the System Controller

The system controller does not allow the installation of any third-party software on it. Thus, you need to access the system controller remotely to find out the domains it is managing, their configuration, or their current status. To facilitate remote access to the system controller, SSH must be enabled on the system controller. The asset management agent uses SSH to access the system controller and then collects the inventory information.

Test the Virtual Host Inventory Manually

You can test the virtual host inventory manually and verify that the correct information has been collected.

Note: Running the inventory collection module manually is for testing or diagnostic purposes. In some situations, diagnostic messages are displayed on the screen, which does not affect the inventory collection and is not harmful.

To test the virtual host inventory manually

1. Create a virtual host inventory collect task using the DSM Explorer.
2. Enter the following command in the Solaris domain where the asset management agent is installed:

   ```
   caf start amagent
   ```

   Note: Starting the amagent will automatically start the AM remote agent and the registration process.

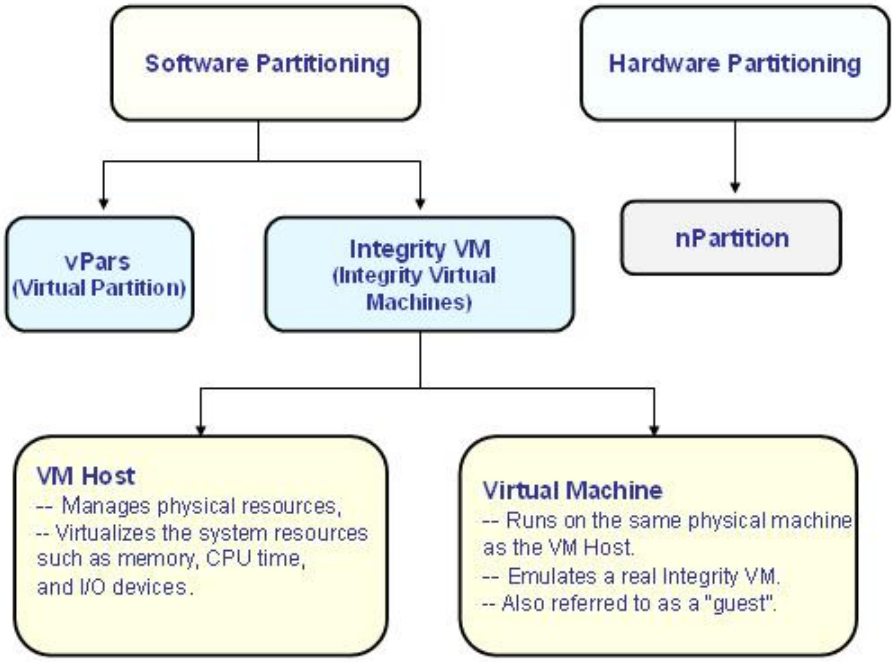
   The system controller inventory is written to the following location:

   ```
   $CA_ITRM_BASEDIR/Agent/AM/data/transfer_vm/12amvminvux.inv
   ```
3. Check the information in the inventory file against the output from various commands specific to the system controller.

   Note: For more information, see the *Sun Fire Midrange System Controller Command Reference Manual* (http://docs.sun.com/source/) and the *Sun SPARC Enterprise M3000/M4000/M5000/M8000/M9000 Servers Administration Guide* (http://dlc.sun.com/pdf).

HP-UX

On HP systems, partitioning is covered under two major categories—software partitioning and hardware partitioning. The following illustration shows the partitioning under both software partitioning and hardware partitioning:



- nPartitions (nPars), which is a hard partitioning, enables the users to create partitions at the hardware level. Each nPartition runs its own OS image, and applications running in one partition are completely isolated from the applications running in other hard partitions. Thus, each partition works as if it is a separate physical server.
- Virtual Partitions (vPars), which is a soft partitioning, enables the users to create multiple virtual systems within a server or nPartition. Each virtual system is isolated from the other virtual system and has its own OS instance, resources, applications, and users.
- HP Integrity Virtual Machines (Integrity VM), which is a soft partitioning, provides the ability to create multiple virtual machines on HP Itanium server or nPartition. Each virtual machine runs its own separate "guest" operating system instance, applications, and users in a fully isolated environment. The Itanium server allocates its physical resources to different virtual machines, which it hosts, as and when required.

HP Integrity Virtual Machines

The AM remote agent collects and reports information for HP Itanium server running HP-UX 11i v2 (May 2005) or later. The virtual machines functionality is provided by application-level software, and each virtual machine can run guests with different operating systems such as HP-UX 11i v2, HP-UX 11i v3, Windows Server 2003, and Red Hat 4 in version 3.0. The following information is collected for HP Itanium server:

- The virtual machines that it is hosting, which includes:
  - The host name for each virtual machine.
  - The IP address for each virtual machine.
  - The operating system installed on each virtual machine.
  - The state (active, inactive) for each virtual machine.
  - The VM guest's serial number and UUID.
- The operating system revision level.
- The physical devices attached to it such as CD, DVD.
- The amount of memory available to it.
- The disk space used out of the total available.
- The host name and IP address of the host.
- The network hardware (card type and configuration, MAC address).

Installation

The AM remote agent is installed in addition to the standard asset management agent.

For HP Integrity Virtual Machines, you need to install this agent on the agent machine (AIX, HP, Linux, Solais, or Windows) and direct it to the VM Host through SSH.

Prerequisites for Accessing the Integrity VM Host (BAR)

The SSH keys must be generated on both the VM Host and the VM Guest systems. You must perform the following prerequisites for accessing the VM Host:

- The HP-UX Distributed Systems Administration Utilities (DSAU) tools can be used to set up the SSH keys on the VM Host and the VM Guest systems, which is installed by default on HP-UX 11.23 (0512 release) or later.
- The “/opt/dsau/bin/csshsetup $VMHostName” command is used to set up the SSH keys between VM Host and the VM Guest systems. Once the SSH keys are set up, the agent uses SSH and accesses the VM Host to collect the relevant inventory information.

Accessing the Integrity VM Host

It is not recommended that you install any application software on the VM Host. You need to access the VM Host remotely to find out the virtual machines it is managing, their configuration, or their current status. To facilitate remote access to the VM Host, you must generate the SSH keys on both the VM Host and the VM Guest systems. The asset management agent uses SSH to access the VM Host to collect the relevant inventory information.

Configuration

To configure the Integrity VM Host, create a virtual host inventory collect task using the DSM Explorer.

Test the Virtual Host Inventory Manually

You can test the virtual host inventory manually and verify that the correct information has been collected.

Note: Running the inventory collection module manually is for testing or diagnostic purposes. In some situations, diagnostic messages are displayed on the screen, which does not affect the inventory collection and is not harmful.

To test the virtual host inventory manually

1. Create a virtual host inventory collect task using the DSM Explorer.
2. Enter the following command in the VM Guest system where the asset management agent is installed:

   ```
   caf start amagent
   ```

   Note: Starting the amagent will automatically start the AM remote agent and the registration process.

   The VM Host inventory is written to the following location:

   ```
   $CA_ITRM_BASEDIR/Agent/AM/data/transfer_vm/12amvminvux.inv
   ```
3. Check the information in the inventory file against the output from various commands specific to the VM Host.

   Note: For more information, see the *HP Integrity Virtual Machines: Installation, Configuration, and Administration* (http://docs.hp.com/).

HP nPartitions and Virtual Partitions

The AM remote agent produces nPar and vPar details on HP servers running HP-UX 11i or later. The agent can access the nPar and vPar systems, obtain and register their inventory information.

Information Collected for HP Chassis and nPartitions

For HP Chassis and nPartitions, the following information is collected for the Chassis and its nPartitions:

- The name of the server
- The serial number of the server
- The model name of the server
- The vendor of the server
- The Hard or Soft partitions information
- The amount of memory available
- The total number of processor of the server

The following information is not collected for the Chassis and its nPartitions:

- The MAC address of the server
- The IP address of the server
- Detailed processor information
- The storage information of the server

Relationship Between nPar and vPar Systems

In the HP-UX nPar system, there can be multiple nPars. Each nPar identifies the other nPar by its partition ID and partition name. All of these nPars share the same serial number.

The nPar does not have any information about:

- The host name of the other nPars
- Any vPar system it owns

Note: One nPar system can have multiple vPars.

Similarly, for the vPar system, each vPar identifies the other vPar by its partition name. All of these vPars share the same serial number with their parent nPar.

The vPar knows about:

- The nPar it belongs to
- The other vPars this nPar owns, and their partition name

However, the vPar does not have information about the host name of the other vPars. The other vPar systems identify themselves with the partition name.

Displaying the nPar and vPar Relationship in the DSM Explorer GUI

The relationship between nPar and vPar is displayed generically as Related Computers in the DSM Explorer user interface. Displaying the relationship as Related Computers ignores the parent-child pattern and provides a simple way of representing the relationship.

Installation

You must install the AM remote agent on every nPar and vPar to collect and report the inventory information. This installation does not require the configuration of the virtual host inventory collect task.

Note: It is recommended that you install the agent on every nPar and vPar only if you have sufficient resources available on the nPar and vPar systems.

Test the Chassis Inventory Module Manually

You can test the Chassis inventory module manually and verify that the correct information has been collected.

Note: Running the inventory collection module manually is for testing or diagnostic purposes. In some situations, diagnostic messages are displayed on the screen, which does not affect the inventory collection and is not harmful.

To test the Chassis inventory module manually

1. Enter the following commands in the nPar (or vPar) system where the asset management agent is installed:

   ```
   cd $CA_ITRM_BASEDIR/Agent/AM/plugin
   ./amvminvhp 0
   ```

   The Chassis inventory is written to the following location:

   ```
   $CA_ITRM_BASEDIR/Agent/AM/data/transfer_vm/amvminvux.inv
   ```

2. Check the information in the inventory file against the output from various commands specific to nPar and vPar.

After the Chassis inventory file is created, you can manually run the AM remote agent using one of the following methods:

- Method 1:

  ```
  cd $CA_ITRM_BASEDIR/Agent/AM/plugin

  ./amVMAgent register
  ```

- Method 2:

  ```
  caf start amagent
  ```

  Note: Starting the amagent will automatically start the AM remote agent and the registration process.

Test the nPar Inventory Module Manually

You can test the nPar inventory module manually and verify that the correct information has been collected.

Note: Running the inventory collection module manually is for testing or diagnostic purposes. In some situations, diagnostic messages are displayed on the screen, which does not affect the inventory collection and is not harmful.

To test the nPar inventory module manually

1. Enter the following commands in the nPar (or vPar) system where the asset management agent is installed:

   ```
   cd $CA_ITRM_BASEDIR/Agent/AM/plugin

   ./amvminvhp 1
   ```

   The nPar inventory is written to the following location:

   ```
   $CA_ITRM_BASEDIR/Agent/AM/data/transfer_vm/amvminvux.inv
   ```

2. Check the information in the inventory file against the output from various commands specific to nPar and vPar.

After the nPar inventory file is created, you can manually run the AM remote agent using one of the following methods:

- Method 1:

  ```
  cd $CA_ITRM_BASEDIR/Agent/AM/plugin

  ./amVMAgent register
  ```

- Method 2:

  ```
  caf start amagent
  ```

  Note: Starting the amagent will automatically start the AM remote agent and the registration process.

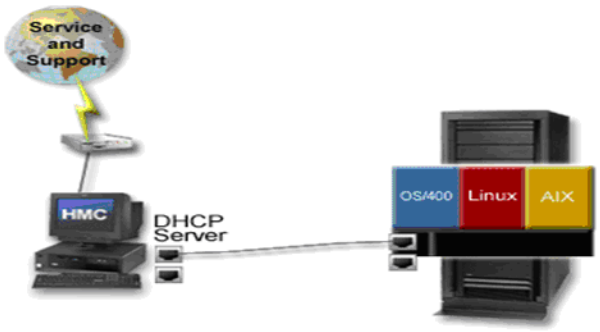
  Note: Collecting the nPar inventory after the Chassis inventory collection overwrites the Chassis information in $CA_ITRM_BASEDIR/Agent/AM/data/transfer_vm/amvminvux.inv.

AIX

On AIX, the module produces logical partition (LPAR) information running AIX 5.3 or above on IBM eServer pSeries systems and IBM System p5 systems, where the IBM Hardware Management Console (HMC) is used to configure and manage the LPARs and the server system.

A logical partition is the division of a computer's processors, memory, and storage into multiple sets of resources, such that a set of resources can be operated independently with its own operating system instance and applications. The number of logical partitions that can be created depends on the system's processor model and resources available.

The Hardware Management Console (HMC) provides a standard user interface for configuring, managing, and monitoring the logical partitions on the IBM eServer pSeries systems and IBM System p5 systems. It consists of a 32-bit Intel-based desktop PC running AIX or Linux with a DVD-RAM drive. The following illustration shows the relationship between the LPAR, its server, and the HMC:



Note: The versions of the HMC supported by the AM remote agent are v2.0 to v7.0.

Information Collected for Each IBM Server

The following information is collected for IBM server:

- The logical partitions that it is hosting, which includes:
  - The partition name for each logical partition
  - The partition ID for each logical partition
  - The state (active, inactive) for each logical partition
  - The server name that the logical partition resides on
- The number of IO devices
- The number of processors
- The amount of memory available
- The serial number
- The name of the server
- The model name of the server
- The vendor of the server

The following information is not collected for IBM server:

- The MAC address of the server
- The IP address of the server
- Detailed processor information about the server
- The storage information about the server

Installation

For the IBM server, the AM remote agent is installed in addition to the asset management agent.

You need to install this agent on the agent machine (AIX, HP, Linux, Solaris, or Windows) and direct it to the HMC through Secure Shell (SSH).

IBM Hardware Management Console (HMC) Configuration Script

The agent uses a trusted SSH key relationship and a special "uamuser" user account to connect to an IBM HMC. This avoids the need to divulge the HMC administrator's password to the agent.

For UNIX and Linux agents, there is a shell-script called UAMhmcConfig that will guide you through the steps to set up the agent. If you are accessing an IBM HMC from a Windows agent, the configuration steps you need to perform are described later in this section.

Prerequisites

There are two prerequisites that must be met to enable the UAMhmcConfig script and the asset management agent to run remote commands on the HMC:

- The "Enable/Disable remote command execution" option on the HMC must be enabled. You can check this by using the HMC system management environment and examining the settings under HMC Maintenance, System Configuration.
- SSH must be installed on the agent machine where the script is to be run.

  Note: SSH is supplied with the AIX operating system but not installed by default.

  Note: For Windows agents, the Plink (PuTTY Link) utility is installed with the agent, providing an SSH command line interface. See the Configuring Plink on a Windows Remote Agent Host (see page 352) section for details. The commands for configuring the Windows agent described later in this section may also work with any other SSH command tool that you might already have.

Linux and UNIX Agent Configuration

The UAMhmcConfig script will set up the necessary requirements for the inventory process to remotely access the HMC. An Asset Management user account (uamuser) will be created on the HMC and the agent uses this account to collect the inventory information. The script will also create SSH keys to enable remote command access to the HMC without password prompting.

The UAMhmcConfig script that can be found in the default installation directory (usually /opt/CA/DSM/Agent/AM/scripts), must be run under the root user account before the asset management agent is to be run. The script only needs to be run once in a single agent machine with the AM remote agent installed to enable HMC access for the agent. However, running the UAMhmcConfig script in more than one agent machine ensures that the overall server inventory is still likely to be produced, even if the agent machine in which the script has been run suddenly becomes unavailable.

If the UAMhmcConfig script is not run before the asset management agent is invoked, the agent will not be able to report on the overall server inventory, such as total resource usage and free resources available on the server.

Once the script is started, you are asked for the following information:

- The host name of the HMC that is used to manage the server.
- The name and password of an existing user account on the HMC, which can be used to create the uamuser account. This existing account must at least have the role of "User Administrator".
- The password for the uamuser account that will be created.

The script checks if the uamuser account already exists on the HMC and only creates it if required. During this check, a message might appear that states that the authenticity of the HMC cannot be established and asks you if you want to continue connecting. This usually means that the host name of the HMC you entered does not have an entry in the $HOME/.ssh/known_hosts file. Answering yes to this question automatically updates the known_hosts file, so you will not see this message in the future.

Once the uamuser account is available, the asset management agent public/private key pair is created. The key files reside in $HOME/.ssh and are named uamkey and uamkey.pub for the private and public keys respectively. To enable the asset management agent to run remote commands under the uamuser account, on the HMC, the public key needs to be placed in the uamuser authorized_keys2 file. The UAMhmcConfig script picks the uamuser authorized_keys2 file, places the asset management agent public key in the file and copies the authorized_keys2 file back again. During this procedure, you are prompted for the uamuser password, which you entered during the account creation phase.

Note: When configuring the virtual host inventory collect task, at the Add Virtual Hosts (see page 185) step enter “$ROOT_HOME/.ssh/uamkey” in the SSH key location field for the IBM HMC.

Uninstallation

Before uninstalling the asset management agent, you must call the script UAMhmcDeconfig (usually under the $CA_ITRM_BASEDIR/Agent/AM/scripts path) to undo the configuration on the HMC. If this is not done before the uninstallation, the “uamuser” user is left in the HMC computer. While on the agent computer, the SSH key for this uamuser is left in $ROOT_HOME/.ssh/uamkey.

Windows Agent Configuration

Configuration of a Windows agent on IBM HMC requires performing the following steps:

- Create a uamuser account on the HMC.
- Establish a trusted key relationship between the Windows agent and HMC using the uamuser account.

Creating the trusted key relationship requires changing an HMC file by editing it on the Windows agent host. An SCP utility is necessary to transfer the file between the hosts. A suitable utility to do this is the PSCP client, which is another utility from PuTTY/Plink. PSCP can be obtained from the PuTTY/Plink website, www.chiark.greenend.org.uk.

To create a 'uamuser' account

1. Log in to the HMC as an administrator (or use an SSH command run as an administrative user).
2. Add a user called "uamuser" to the HMC using *one* of the following commands:

   For HMC version 4 or greater, enter:

   ```
   mkhmcusr -u uamuser -a hmcoperator -d UAM user
   ```

   For older HMC versions, enter:

   ```
   mkhmcusr -u uamuser -a sysadmin -d UAM user
   ```

   To use Plink to perform this step, enter:

   ```
   "C:\Program Files\CA\DSM\Agent\units\00000001\uam\plink" –ssh
   hmc-admin-user@hmc-host mkhmcusr -u uamuser -a hmcoperator -d UAM user
   ```

   The *hmc-admin-user* variable represents the name of the HMC administrative user with rights to create new users, and *hmc-host* represents the host name or IP address of the HMC to which you need the Windows agent to connect.

To establish a trusted key relationship between the Windows agent and HMC

1. Generate a public/private SSH key pair, for example, by using PuTTYgen or another SSH tool.
2. Use an empty passphrase for the private key.
3. Install the public key into the authorized key file for the uamuser on the HMC. To do this:

   a. Copy the key file from the HMC server to the Windows asset management agent host to edit it using an SCP client. For example, with PSCP the command line is as follows:

   ```
   pscp –scp uamuser@hmc-host:/home/uamuser/.ssh/authorized_keys2
   C:\Temp\authorized_keys2
   ```

   b. Edit the C:\Temp\authorized_keys2 file (to continue our example). Append to this file a line containing the public key generated above.

Important! Keep the file in Unix format; do not convert to DOS.

c. Upload the modified file to the HMC server, using the following command, for example:

```
pscp –scp C:\Temp\authorized_keys2
uamuser@hmc-host:/home/uamuser/.ssh/authorized_keys2
```

4. Save the private key in a safe location (for example, “C:\Program Files\CA\DSM\Agent\uamkey.txt”).
5. Specify this key file in the collect task dialog where it prompts for the “SSH Key Location”.

Uninstallation

Before removing the Windows agent, or if you wish to revoke the trusted relationship between the Windows agent and an IBM HMC, perform the following steps:

- Revoke the trusted key relationship between the Windows agent and the IBM HMC.
- Remove the uamuser user account on the IBM HMC.

To revoke the trusted key relationship

1. Log in to the HMC as an administrative user.
2. Edit the /home/uamuser/.ssh/authorized_keys2 file. Remove the public key inserted during configuration.
3. Save the file.
4. Remove the private key file stored on the Windows agent computer that was created during the configuration steps described earlier.

To remove the uamuser user account

1. Log in to the HMC as an administrative user.
2. Run the following command:

   ```
   rmhmcusr -u uamuser
   ```

   To use Plink to perform this command from the Windows computer, run the following command:

   ```
   "C:\Program Files\CA\DSM\Agent\units\00000001\uam\plink" –ssh
   hmc-admin-user@hmc-host rmhmcusr -u uamuser
   ```

Test the Virtual Host Inventory Manually

You can test the virtual host inventory manually and verify that the correct information has been collected.

Note: Running the inventory collection module manually is for testing or diagnostic purposes. In some situations, diagnostic messages are displayed on the screen, which does not affect the inventory collection and is not harmful.

To test the virtual host inventory manually

1. Create a virtual host inventory collect task using the DSM Explorer.
2. Enter the following command in the agent machine where the asset management agent is installed:

   ```
   caf start amagent
   ```

   Note: Starting the amagent will automatically start the AM remote agent and the registration process.

   The HMC inventory is written to the following location:

   ```
   $CA_ITRM_BASEDIR/Agent/AM/data/transfer_vm/12amvminvux.inv
   ```
3. Check the information in the inventory file against the output from various commands specific to the HMC.

VMware ESX

VMware ESX is a data center virtualization platform. The VMware ESX server does not run on top of a third-party operating system, but contains its own kernel called the vmkernel. Multiple virtual machines can run on a single VMware ESX server. A virtual machine is a tightly isolated software container that can run its own operating systems and applications as if it were a physical computer. A virtual machine behaves exactly like a physical computer and contains it own virtual (that is, software-based) CPU, RAM, hard disk and network interface card (NIC).

VMware ESX Server 3.5 and 4.0 are supported by the AM remote agent.

Collection of Inventory from VMware ESX Servers

CA ITCM provides two ways to collect inventory from VMware ESX servers.

- The first method is to install an asset management agent on the VMware ESX service console.

  For VMware ESX 3.5, the special VMware ESX agent on the supplemental CD must be installed.

  For VMware ESX 4.0, the Release 12.8 agent may be installed.

  This agent will collect the properties of the VMware service console. The disk, memory, and other system properties that are collected will describe the state of the VMware service console and its allocated resources. Note that this will be a small subset of the whole machine's resources. For backwards compatibility, this agent will report the inventory using the host name reported by the VMware service console.

- The second method is to install and configure the asset management Remote Virtualization Inventory add-on to the asset management agent, a new feature in CA ITCM Release 12.8, which will collect the properties of the whole computer. The disk, memory, and other system properties reported by this agent will describe the state of the whole computer's hardware. In order to distinguish this inventory from the data returned by an agent that might have been installed on the VMware service console, this agent will report the inventory using the service console's host name prefixed with "VMware ESX – ".

If both methods of inventory collection for the same computer are used, the DSM Explorer will show the relationship between the inventory collected for the VMware service console and for the entire machine by using the Related Computers pane.

Information Collected for VMware ESX Servers

The information collected by the AM remote agent is similar to that collected by the local agent; however, there are differences in the type of information that can be collected by the AM remote agent, and this is dependent on the capabilities of the VMware programming interface.

The AM remote agent will report VMware ESX supported features as well as configuration information. This information is not reported by the local agent. The AM remote agent inventory collection, however, does not currently report information such as the host input devices.

Note: The collection of some Virtual Machine properties by the AM remote agent requires that VMware tools be installed and running on the Virtual Machines. The status of VMware tools can be determined by the Tools Status column for Virtual Machine properties. The properties that require VMware tools are Host Name, IP Address, Tools Version, Guest Family, Total Disk Space, Total Free Space, and Label.

VMware ESX Security and Authentication

CA ITCM's AM remote agent only supports VMware ESX servers running in secure HTTP mode. VMware recommends that secure HTTP (HTTPS, the default configuration) be used for production deployment. For this reason, the AM remote agent connects only to ESX hosts where the default, recommended configuration of secure HTTP is used.

Secure Sockets Layer (SSL) is used as the connection protocol between the DSM agent and the ESX host machine. However, because the identity of an ESX host machine is not known in advance, there is no authentication between the DSM agent and ESX host. Certificates will not be used to authenticate an ESX host.

Configuration

To configure inventory collection for VMware ESX hosts, create a virtual host inventory collect task using the DSM Explorer.

VMware ESX Parameters

To access the VMware ESX Web Service, you must supply the logon credentials for an ESX host machine. The credentials must be for a user of a VMware ESX host who has the VMware system role of Administrator or Read-only.

Hostname

Specifies the name of the ESX host for which asset management inventory is to be collected. An IP address can also be supplied.

Web Service Username

Specifies the name of a VMware ESX user who has the VMware system role of Administrator or Read-only.

Web Service Password

Requires the password corresponding to the specified Web Service Username.

Web Service URL

Indicates the Web Service URL, which takes the following form:

`https://ESXHostFQDNservername/sdk`

ESXHostFQDNservername represents the fully qualified host name of the ESX server. Alternatively, an IP address can be given instead of the host name.

ESX inventory collection is done using the Web Service (SOAP) engine. By default, the Web Service runs on port 443 as a secure web service that can be accessed using SSL over HTTPS.

Installation

The AM remote agent is installed as an additional package to the asset management agent. The AM remote agent package is available for Windows or Linux.

Test the Virtual Host Inventory Manually

You can test the virtual host inventory manually and verify that the correct information has been collected.

Note: Running the inventory collection module manually is for testing or diagnostic purposes. In some situations, diagnostic messages are displayed on the screen, which does not affect the inventory collection and is not harmful.

To test the virtual host inventory manually

1. Create a virtual host inventory collect task using the DSM Explorer.
2. Enter the following command on the Windows or Linux computer where the AM remote agent is installed:

   ```
   caf start amagent
   ```

   Note: Starting the amagent will automatically start the AM remote agent and the registration process.

   The ESX host inventory is written to the following location:

   ```
   $CA_ITRM_BASEDIR/Agent/AM/data/transfer_vm/12amvminvux.inv
   ```
3. Check the information in the inventory file against that in the VMware Virtual Infrastructure Client or Managed Object Browser.

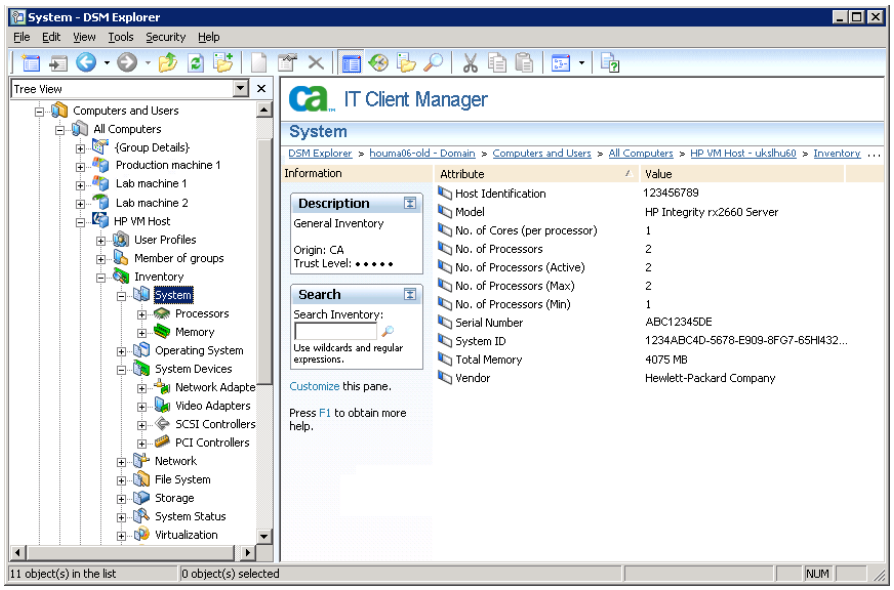
Viewing Virtual Host Inventory

Collected inventory information for virtual host computers is displayed like any other inventory information in the tree view of the DSM Explorer.

The inventory reflects the specific properties of a virtual host, and Sun, IBM, HP, and VMware ESX servers are presented as computer units.

A computer unit appears for each of the managed guest computers. The virtual host unit and its associated guests are reported as related computers, and you can easily navigate between both.

The following illustration shows the inventory information for a virtual host:



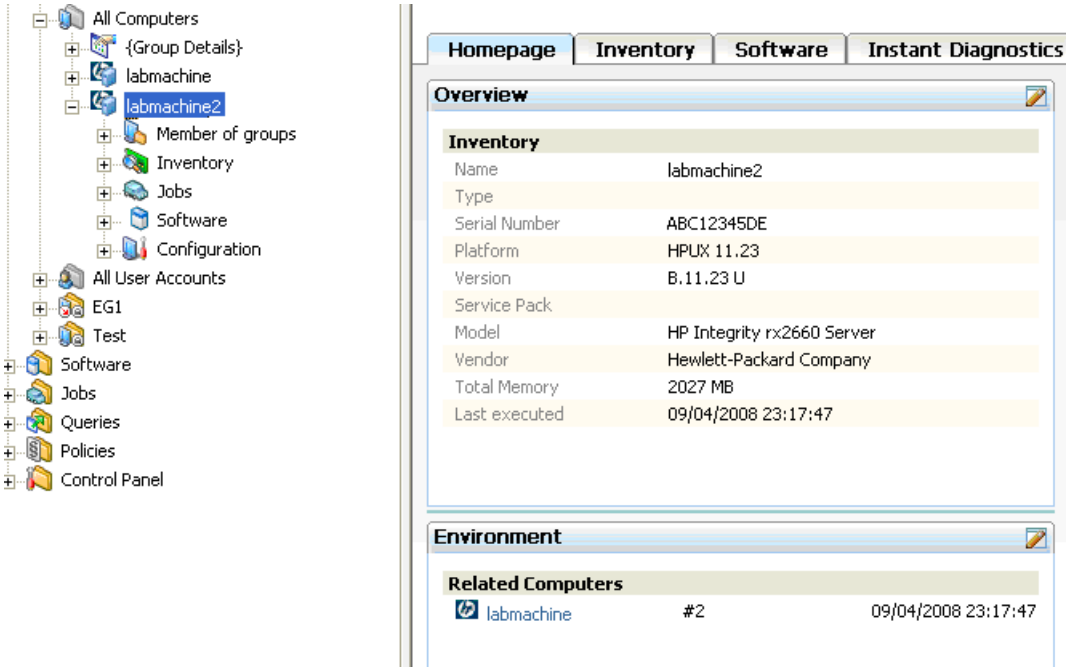
Note: The number of processors displayed in the DSM Explorer (under the Inventory, System, Processors node) for the inventory data of the system controller computer includes only the powered up processors. The powered off processors are not counted in this case.

More information:

Viewing Remote Agents and Virtual Hosts (see page 187)

Relationships Between a Virtual Host and Guests

Virtual host and guest computers register as asset management units and are displayed in the tree view of the DSM Explorer. The following illustration shows the relationship between a virtual host and its managed guest computers:



In the Homepage tab, click the unit hyperlink available under Related Computers to see the virtual host with which the managed guest has relations.

Creation of the Relationship Between a Virtual Host and Its Guests

The relationship between a virtual host and its guests is created when the virtual host is registered to the domain manager. The domain manager creates the relationship between the virtual host and all of its known guests (such as HP-UX nPars and vPars, IBM LPAR, and Sun domains) in the domain. The relationship between guests that are registered after the virtual host has been registered is not created until the next virtual host registration process.

Note: For HP-UX nPar, HP-UX vPar, and IBM LPAR, there is no parent-child relationship between the virtual host and its guest system.

How the AM Remote Agent Is Displayed in the Computer Properties Dialog

As the AM remote agent is not actually installed on the HMC, the system controller, and the VM Host system, it is displayed as "Asset Management Virtual Agent" on the Agent tab of the Computer Properties dialog instead of the more standard "Asset Management Agent." The Agent tab also displays other agent-related information such as agent name, registration, and so on.

To view agent information

1. Right-click the computer and select Properties.

   The Computer Properties dialog opens.

2. Click the Agent tab.

   For an AM remote agent, the Components column displays Asset Management Virtual Agent as its description.

Configuring Plink on a Windows Remote Agent Host

The CA ITCM AM remote agent installs and uses Plink (PuTTY Link) as an SSH client to communicate with most types of remote hosts. However, the first time that Plink connects as a given user to a new target it displays the host key and, as a security measure, interactively asks if the host is legitimate. Obviously, the AM remote agent does not support such interactivity with a user, so you must first reassure Plink before gathering inventory.

Since Plink saves the remote host's key in the registry, you can generate such a registry entry for a convenient login user and then save the same key under the LocalSystem account that the asset management agent runs as.

To generate a registry entry and save the host key under the LocalSystem account

1. If it has not been done already, generate a public and private SSH key pair using PuTTYgen, for example, and install the public key on the remote target.
2. As a normal login user, run Plink in a command window to the remote host(s) for which inventory is to be gathered, using something like the following:

   ```
   "C:\Program Files\CA\DSM\Agent\units\00000001\uam\plink" –ssh –i
   "private_key_file" root@target ls
   ```

   The variable *private_key_file* is the absolute path of the file, and *target* is the remote host name. Adjust the Plink path if CA ITCM is installed at some other location.

3. If PuTTY says that the server's host key is not cached in the registry and asks if you want to add the key, respond affirmatively. The command (ls in this example) should now run successfully. If not, check the SSH key files and repeat if necessary.

4. Search for the cached key in the registry using regedit or some other registry editor. Look under HKEY_CURRENT_USER\Software\SimonTatham\putty\SshHostKeys. Export this key to a temporary file.
5. Edit the export file by replacing all occurrences of HKEY_CURRENT_USER by the LocalSystem SID, HKEY_USERS\S-1-5-18. Save the edited file.
6. In regedit, import the edited file.
7. Check that the HKEY_USERS\S-1-5-18\Software\SimonTatham\putty\SshHostKeys key contains an entry for the remote host.

Plink must be configured for each remote host for which the AM remote agent gathers inventory; so connect to all remote hosts in order to generate host keys, and then run the above export-import procedure once.

If you suspect that Plink may not be configured correctly for the LocalSystem account, run the asset management agent (“caf start amagent”), wait for all AM processes to finish, and then examine the trace file, TRC_UAM_amvminvux_*.log. Search for “The server's host key is not cached in the registry.” If you find such an error, Plink is not correctly configured. Also, if you see the "Access denied" error in the trace file, this usually means a problem with the credentials used to access the remote host, for example, an incorrect password or SSH key.

Note: For detailed information about Plink, refer to the PuTTY/Plink manual on the following website: http://www.chiark.greenend.org.uk/~sgtatham/putty/docs.html

Non Resident Inventory

The Non Resident Inventory (NRI) solution provides a simple way to inventory a Windows, Linux, or UNIX computer without having to deploy the CA IT Client Manager agent (DSM agent) to it.

The NRI solution is based on existing CA ITCM inventory components, and it provides the same robust discovery capability that the installed agent-based discovery does, but works on computer systems without any CA software installed prior to or after the inventory scan has been performed.

Three different levels of discovery capabilities are provided, with each level accessible through the installed NRI web-site that you should visit in order to inventory your systems. The NRI agent can, however, also be executed through login scripts.

The NRI solution is customizable and offers you the capability of creating your own levels or packages to suit your needs.

More Information:

Configuring the NRI Web Service (see page 362)
Customizing the Web Page (see page 362)
Launching NRI Elective Inventory (see page 355)

Requirements

Non Resident Inventory requires the following components to be already installed:

- Web Administration Console (WAC)
- DSM Web Service
- Scalability Server
- CA Asset Collector

Launching NRI Elective Inventory

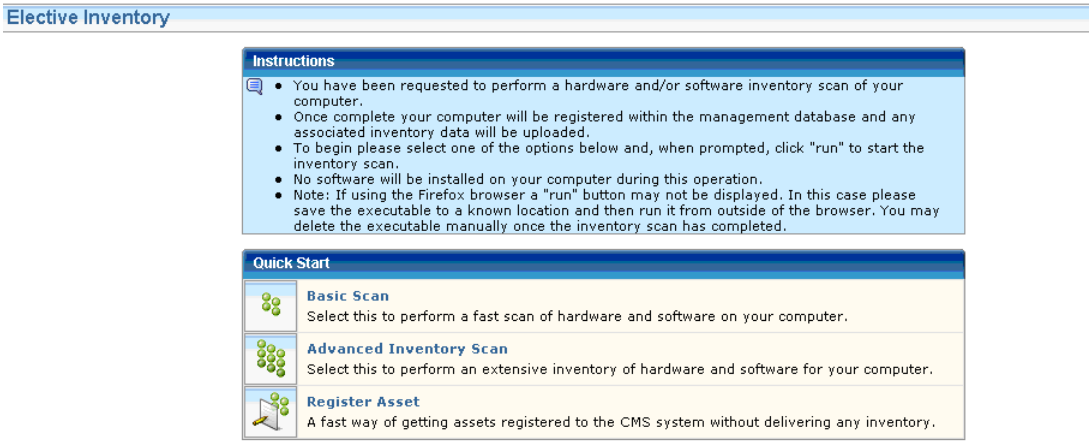
The NRI Elective Inventory web page is immediately accessible to all users who have access to the hosting web server.

Important! The received inventory folder and the Asset Collector folders must be fully accessible by the Internet Guest Account, which must be set up manually by the user. See the Configuring the NRI Web Service section to better understand which folders to modify.

To navigate to the NRI web page, enter the following URL in a supported web-browser. For more information about supported web browsers, see Certification Matrix

```
http://webserver-hostname/wac/jsp/ei/ei.jsp
```

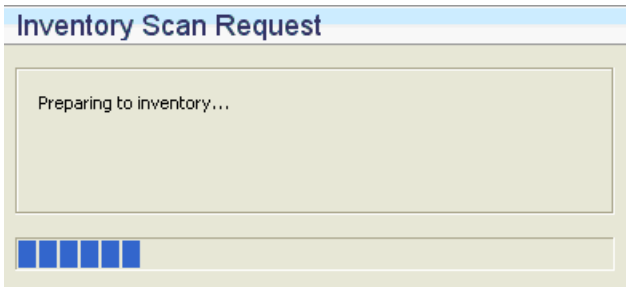
The NRI web page is based on the CA ITCM WAC system and offers a simple view of the defined inventory levels or packages, as shown in the following illustration:



To start the inventory, select one of the offered packages from the Quick Start portlet by simply clicking the image or the link. (See NRI Agent Packages for more information about what each package collects and does.)

Each link points to a named version of the NRI Primer executable. When the web browser asks if you want to “run” or “save” the file, select Run. The NRI Primer is downloaded and executed.

The following illustration shows the progress of the inventory scan request:



The time used to download the necessary detection modules and perform the actual inventory depends on the current load of the NRI hosting web server. For the Advanced Inventory Scan option especially, the number of files on the installed hard-drives can affect the time.

The scan can take anything from one (1) minute to more than 30 minutes to complete. The scan process will utilize a significant part of the computer's resources but can be used normally throughout the scan.

A message dialog declares when the scan is done.

Launch NRI from Linux or UNIX Computer

You cannot launch NRI on Linux or UNIX from the NRI website. NRI on these operating environments is designed to run with minimum requirements. The distribution of the NRI agent and analysis of the collected result is done manually.

To launch NRI from a Linux or UNIX computer

1. From the command shell, go to:

   <dvd_root>/*ProductFiles_x86*/nriagent/nriagent.tar

   Note: The nriagent.tar file is located under the platform specific folder. You can also copy it to any shared location and launch the NRI.

2. Extract the file to any location using the command:

   ```
   tar –xf nriagent.tar
   ```

   The files are extracted to the folder *nriagent*.

3. To register a computer, run the script:

   ```
   ./cmnriagent –script register.ini
   ```

   An inventory file is created to register the computer.

4. NRI allows you to perform two types of inventory:

   - Basic hardware inventory and heuristic software scan
   - Full hardware inventory and software signature scan.

   To perform a basic hardware inventory and heuristic software scan, run the script:

   ```
   ./cmnriagent –script basic.ini
   ```

   To perform a full hardware inventory and software signature scan, run the script:

   ```
   ./cmnriagent –script adv.ini
   ```

   Note: To use additional inventory modules, create a customized .ini file and copy it to the nriagent folder.

   The inventory starts and an inventory report is created in the *nriagent* folder and is named after the generated host UUID on the computer, for example, 12FDBEBA-572D-4408-BFC8-E7922AD4A998.xiu.

5. Copy the inventory report to one of the AssetCollectorCollect folders belonging to a running Asset Collector.

   The Asset Collector detects the new inventory file and extracts the asset inventory information.

NRI Agent Packages

A predefined set of NRI agent packages is created when a scan is requested. A package contains the NRI agent executable, needed libraries, and inventory detection modules with their needed libraries. All files are placed into one folder, meaning that various packages share common libraries. The default package location is ...\DSM\Web Console\webapps\wac\jsp\ei\wl\windows\packages, for example, C:\Program Files\CA\DSM\Web Console\webapps\wac\jsp\ei\wl\windows\packages.

The following three packages are created:

- Basic Scan

  Executes the basic inventory scanner and the heuristic scanner.

- Advanced Scan

  Executes the general inventory scanner and software signature scanner.

- Register Asset

  Registers assets simply and quickly. No hardware inventory or software information is reported.

Reported Information

Although the NRI agent uses the standard DSM agent components, there could potentially be less inventory reported by the NRI agent than by the DSM agent. The reason is that in some of the use cases mentioned above the NRI agent is likely to be executed by a user with a standard domain user account, and this will typically have less privilege than the Local System account (or optionally an administrator account) under which the regular DSM agent runs. Based upon the privileges under which it is being executed, the NRI agent collects as much information as it can.

The NRI agent uses three techniques to obtain the hardware inventory information:

- Calling standard Windows API functions
- Reading registry keys
- Installing and running a device driver to collect SMBIOS inventory data. The most detailed hardware inventory information is collected via this SMBIOS mechanism. Once the SMBIOS data is collected, the device driver is removed leaving no lasting impact on the configuration of the inventoried system. As installing a device driver requires privilege, the detailed SMBIOS data cannot be collected unless the NRI agent is run as a privileged user.
- The NRI agent collects three different memory values:
  - OS Reported Memory

    This value is equal to the amount of memory that is reported by the operating system to be available after devices such as video adapters have taken their share.
  - Physical Memory

    This value is equal to the memory mounted on the main board.
  - Total Memory

    This value equals the upper value of the two above values.

Note: Physical Memory is not reported if the NRI agent is executed in the context of a restricted user.

If the NRI agent is executed through a Remote Desktop connection, some values can be reported that are different to the actual values.

For heuristic-based software scanning, the NRI agent obtains details of the software installed by examining the:

- Windows MSI database to obtain details of installed MSI packages
- Windows Add/Remove Software list

For signature-based software scanning, the detection of installed software relies upon software content signatures being available to the NRI agent. As supplied, the NRI agent is shipped with the latest available content signatures; however, the DSM administrator can update this with newer content definitions by the simple replacement of a single signature content file.

If the NRI agent is executed by a restricted user, the amount of reported software inventory can vary from what is reported by a user with administrative rights. For heuristic software, only applications available to the user are reported. For a signature scan, some signatures can be prohibited from checking if an application is installed due to file access restrictions.

The default packages return a trust level of 5, if the user is an administrator. If the user is restricted, the trust level is returned as 3. During reporting on the collected inventory it is important to note the reported trust level.

If an asset has previously been inventoried by a full DSM agent, then the NRI inventory will overwrite that inventory if the inventory has been collected by an administrator (trust level 5). NRI inventory generated by a non- administrative user will not overwrite inventory that was collected by a DSM agent.

For a list of reported inventory information that can be expected, see List of Inventory Items Reported (see page 473).

Update the Signature Database

The Advanced Scan package offers software detection through the CA ITCM software signature scanner. The signature scanner uses a signature database to recognize the installed software. During installation of the Non Resident Inventory solution a version of the signature database file is also installed.

The CA ITCM system automatically downloads new signatures to the core CA ITCM system for use with the real CA ITCM agent and beyond, custom created ones might be added as well. However, the Non Resident Inventory solution is not included in these automatic updates, so you have to perform these updates manually.

To manually update the signature database used by the Advanced Scan package

1. Open a file explorer on the server hosting the solution.
2. Navigate to the ...DSM\ServerDB\SECTOR\SSFW folder.

   The folder contains a list of W0000xxx.zml files.
3. Locate the .ZML file with the highest number, for example, W0000084.zml.
4. Copy the file to Web Console\webapps\wac\jsp\ei\wl\windows\packages.
5. Navigate to Web Console\webapps\wac\jsp\ei\wl\windows\packages.
6. Remove file named w0000001.zml.
7. Rename the copied file to w0000001.zml.

Limit the Number of Primer Scripts

There are two options for limiting the number of offered packages from the Elective Inventory web page. The web page is dynamically built based on the available primer scripts

Primer scripts are placed in the ...DSM\Web Console\webapps\wac\jsp\ei\wl\scripts folder on the web server hosting the NRI solution. The file names reveal the purpose of each script.

Note: For more information, see the NRI Primer Script File section.

To remove any option from the web page

1. Remove the file from the scripts folder or change the extension of the file to something other than ".wlscr".
2. After removing or renaming the file, restart the CA ITCM Tomcat service from a command prompt, entering the following commands:

   ```
   caf stop tomcat
   ```

   ```
   caf start tomcat
   ```

   Important! The Web Admin Console, as well as the NRI web page, becomes inaccessible, and currently logged-on users' sessions will expire while the Tomcat service is stopped.

Alternatively, it is possible to limit the listed scripts to a single script by specifying the name of the script file as part of URL, for example,

http://webserver-hostname/wac/jsp/ei/ei.jsp?script=2adv

This example limits the list to only the Advanced Inventory Scan package.

Customizing the Web Page

The NRI Elective Inventory web site can be customized to adhere to your organization's standards.

The main page for the NRI solution is “ei.jsp”, which is located in the ...\DSM\Web Console\webapps\wac\jsp\ei folder.

Important! Do not change or overwrite any of the server side scripting tags in the file. Server-side code is encapsulated in a “<%” starting tag, and a “%>” ending tag.

Images currently used by the web page are located in the ...\DSM\Web Console\webapps\wac\jsp\ei\images folder. The CA logo file is called “circleca.png” and can be substituted with a logo matching your organization. Alternatively, a new logo file can be placed in this folder. You must change the reference in the “ei.jsp” file to point to the new logo.

Labels displayed on the page are externalized to the file named “i18n_ei.properties” located in the ...\DSM\Web Console\webapps\wac\WEB-INF\classes\com\ca\wac\ei\i18n folder.

For each change to take effect a restart of the Tomcat service is necessary. Run the following commands to restart the Tomcat service:

```
caf stop tomcat
caf start tomcat
```

Note: Before making changes to the web site ensure to back up your files.

Configuring the NRI Web Service

The NRI Primer uses the new NRI Web Service to download and upload files from the hosting web server. The NRI Web Service is configurable through a CA ITCM common configuration policy that can be modified and applied from the DSM Explorer. (See the Configuration Policy section of the *DSM Explorer Help* for additional information about configuration policies.)

The policies specific to the NRI Web Service are found under the Control Panel, Configuration, Configuration Policy, Default Computer Policy, DSM, Web Services, Non Resident Inventory subnode in the DSM Explorer tree.

Non Resident Inventory Policy Group

The Non Resident Inventory policy group lets you view or edit the existing configuration policy parameters that control the behavior of the NRI Web Service. You can modify policy parameter values by double-clicking a policy to display the Setting Properties dialog.

Copy files to Asset Collector

Specifies whether or not the NRI Web Service should copy the received inventory reports to the local Asset Collector. If enabled, the files are placed in the first configured collection folder of the Asset Collector. Valid values are as follows: 1 = enabled and 0 = disabled.

Default: 1

Important! If the collection folder is different from the default collection folder (that is, AssetCollectorCollect relative to the DSMROOT folder, c:\program files\ca\dsm), then this folder must be fully accessible by the Internet Guest Account, which must be set up manually by the user.

Delete files after copy to Asset Collector

Specifies whether or not the NRI Web Service removes the files from the received inventory folder after it has copied them to the Asset Collector folder. This policy is only acknowledged if the Copy files to Asset Collector policy is set to 1. Valid values are as follows: 1 = enabled and 0 = disabled.

Default: 0

Enabled

Indicates whether or not the NRI Web Service should accept a file download or inventory upload request. Valid values are as follows: 1 = enabled and 0 = disabled.

Default: 1

Folder where received files are placed

Specifies where the NRI Web Service should place the received inventory reports. Unless the folder is fully qualified, it is placed relative to the DSMROOT folder, for example, c:\program files\ca\dsm.

Default: NRIReceivedReports

Important! If the default value is changed to a different folder, then this folder must be fully accessible by the Internet Guest Account, which must be set up manually by the user.

Running the NRI Agent from a Login Script

As an alternative to using the Elective Inventory web page, you can execute the NRI agent directly, for example, using a login script or even a USB memory stick.

The NRI agent executable, cmNriAgent.exe, is located in the ...\DSM\Web Console\webapps\wac\jsp\ei\wl\windows folder along with all the necessary inventory modules and supporting libraries.

In the same folder there are also three script files corresponding to the three primer scripts and options offered on the Elective Inventory page. To execute the NRI agent using one of the scripts, run the following command from a command prompt:

```
cmNriAgent.exe –script .\basic.ini
```

This command executes the NRI agent and creates an inventory report named after the generated host UUID on the computer, for example, 12FDBEBA-572D-4408-BFC8-E7922AD4A998.xiu.

This file can then be copied manually at any time to one of the collection folders belonging to a running Asset Collector.

Note: For more information, see the CA Asset Collector section.

The NRI agent can also be configured to direct the output directly to the Asset Collector folder by using the “-o” argument on the command line. For more information about the capabilities of the NRI agent, see the next section.

It is equally easy to run the NRI agent using a USB memory stick. Simply copy all the files from ...\DSM\Web Console\webapps\wac\jsp\ei\wl\windows to the USB stick. Then call the NRI agent as follows, preferably specifying that the output be sent to a subfolder on the memory stick:

```
cmNriAgent.exe –script .\basic.ini –o .\reports
```

Note: The NRI agent expects the output folder to exist and will fail if one does not.

The NRI Agent

The core component of the Non Resident Inventory solution is the new NRI agent. The agent knows the internal workings of the existing CA ITCM inventory detection modules, and is able to convert their output to an Asset Collector recognized XML file.

The NRI agent is executed manually and is controlled from the command line or through a script file or a combination of both.

Command Line

The following table lists the command line switches that the NRI agent recognizes:

Command	Description
-o “<output path>”	Specifies the location where the agent should put the inventory XML file.
-origin “wished origin”	Specifies the origin of the asset. See the Asset Collector documentation for more information on origin.
-trustlevel <trustlevel>	Specifies the trust level of the asset. The trust level can be a number between 1 and 5. See the Asset Collector documentation for more information on trust level.
-file “[set the File Name variable]”	The output file is by default named using the generated host UUID. This can be overridden through this parameter.
-nohw	Do not collect hardware inventory.
-nosw	Do not collect software inventory.
-script <script file>	Location of script file.

Script File

The agent can be executed and instructed on what to do through a script file. The script exposes some of the options that are offered on the command line. If options are specified on the command line as well as in the script file, the command line specification takes precedence.

Note: If you want to modify the default configuration script files, they are located in the ...\DSM\Web Console\webapps\wac\jsp\ei\wl\scripts folder. Their respective file names are 1basic.wlscr, 2adv.wlscr, and 3register.wlscr.

The script file must always have a General section:

```
[General]
OutputFolder=.\
OutputFile=
Origin=CA Elective Inventory
Trustlevel=5
TrustlevelRestrictedUser=3
KeepHostkey=Yes
ProcessPriority=0-3
```

The following table includes the parameters available in the General section:

Parameter	Description
OutputFolder	Specifies the location where the agent should put the inventory XML file.
OutputFile	Specifies the output file. By default, the output file is named using the generated host UUID. This can be overridden through this parameter.
Origin	Specifies the origin of the asset. See the Asset Collector documentation for more information on origin.
Trustlevel	Specifies the trust level of the asset. The trust level can be a number between 1 and 5. See the Asset Collector documentation for more information on trust level.
TrustLevelRestrictedUser	Specifies the trust level of the user if the NRI agent is executed by a restricted user. Restricted users do not have the same access to information as administrators, hence the reported information cannot be considered as trustworthy. If this item is not specified the “trustlevel” value is used.

Parameter	Description
KeepHostkey (optional)	If specified and the value is Yes, the agent leaves the generated hostkey in the registry. If the value is No, the key is removed. Default is Yes. The hostkey should be kept if there are plans to push the real CA ITCM agents later. The CA ITCM agent then reuses the hostkey, making it easier to consolidate it with the elective scanned agent.
ProcessPriority (optional)	This can be used to control the priority of the sub-processes launched by the CA ITCM agent. The default value is 0 which is “idle”, meaning that sub-processes only use the CPU which is not consumed by other processes. Possible values are: 0 = Idle (default) 1 = low 2 = normal 3 = high **Note:** The actual prioritization is controlled by the OS.

Script File: Inventory Section

To specify which inventory detection modules to execute, an Inventory section must be added to the script file. If the section is omitted, the agent executes the basic inventory scanner by default.

Example: Inventory Section

```
[Inventory]
Method=0/1/2

ScanRestrictedUser=0/1
AdditionalModules=1
Module1Binary=vistainv.exe
Module1Commandline=-o inv.out
Module1Outputfile=inv.out
Module1Cleanup=inv.out,tmp.bak
TemplateModules=1
Template1=user_information.tpl
```

The following table includes the parameters available in the Inventory section:

Parameter	Description
Method	Specifies one of the default inventory modules. The value can be one of the following: 0 = None of the default modules are executed. 1 = Runs the basic inventory module. Creates about 4KB of information. 2 = Runs the full general inventory module. Creates about 30KB of information.
ScanRestrictedUser	Indicates if hardware inventory scanning should be performed for restricted users. 0 = No. Restricted users are not scanned. 1 = Yes (default)
AdditionalModules	Specifies the number of additional modules to execute. The agent will iterate from 1 to the number of modules specified and read the module information from the ModuleXBinary, ModuleXCommandLine, ModuleXOutputfile, and ModuleXCleanup items.
ModuleXBinary	Specifies the file name for inventory module.
ModuleXCommandline	Specifies the command line to be used to execute the inventory module.

Parameter	Description
ModuleXOutputfile	Specifies the name of the file in which the inventory module stores the information.
ModuleXCleanup	Specifies a comma-separated list of files that the agent should delete after module execution. Files can be specified with a path.
TemplateModules	Specifies the number of inventory template modules to prompt the user for. **Note:** In order to run template inventory through NRI Elective Inventory, you must add the following files to the download section of the NRI Primer script file: amtplw32.exe, amtext.enu and amm2iw32.exe.
TemplateX	Specifies the file name of template file. The template file is in MIF format, and can be created either by the DSM Explorer or by manually launching the gui_am_tpledit.exe from the CA ITCM installation.

Script File: Software Section

Control of software inventory scanning is done in the Software section of the script file.

Example: Software Section

```
[Software]
Method=0/1/2

ScanRestrictedUser=0/1
```

The following table includes the parameters available in the Software section:

Parameter	Description
Method	Specifies which software scanning method to use. The value can be one of the following: 0 = No software scan is performed. 1 = Runs the heuristic scanner which scans the MSI and Add/Remove database. 2 = Runs the Signature Scanner which uses the CA provided signature database to recognize software.
ScanRestrictedUser	Indicates if software inventory scanning should be performed for restricted users. 0 = No. Restricted users are not scanned. 1 = Yes (default)

The NRI Primer

The NRI Primer is the binary that is linked to and from the Web Console web pages which host the NRI solution. The Web Launcher can, however, be executed directly from a disk or network share if desired.

For the NRI Primer supported operating systems, see Certification Matrix

The NRI solution's main responsibilities are to execute the NRI agent and to facilitate a way to communicate between the inventoried computer and the CA ITCM domain manager over the intranet or Internet. The communication is done using a new set of web service methods.

Script File

The Web Launcher is controlled through a script file. The script file includes information about which files to download and execute, as well as information on how to download and upload files and to which server. The script file is in INI file format and contains three sections: General, Advanced, and Platforms.

More information:

Script File: General Section (see page 370)
Script File: Advanced Section (see page 371)
Script File: Download Primer <Platform> Section (see page 372)
Script File: Download <Platform> Section (see page 372)

Script File: General Section

The General section contains information which can be used by the Web Console to form the link and to determine if the script contains support for the operating system of the hosting web browser.

Example: General Section

```
[General]
Title=Basic Inventory Scan
Description=Select this to perform a basic and fast inventory of your computer.
Image=nri_basic_scan.gif
```

The following table includes the parameters available in the General section:

Parameter	Description
Title	This is solely for display purposes and can be used to form the link in the WAC.

Parameter	Description
Description	A general description.
Image	Specifies the file name for the image used in the Web Console page. Image is picked from the common "images" folder of the Web Console installation. If no image is specified, the nricustom.gif is used.

Script File: Advanced Section

The Advanced section currently contains only the following item.

Example: Advanced Section

```
[Advanced]
KeepWorkArea=0/1
```

The following table includes the parameters available in the Advanced section:

Parameter	Description
KeepWorkArea	Default value is 0. If the value is set to 1, the NRI Primer does not remove all downloaded files. Generated inventory files are not deleted as well.

Script File: Platforms Section

The Platforms section lists which platforms the script supports. There are currently four main platforms defined, as listed below. If the value is Yes, the script is supported for the language. If the value is No or a platform is not listed, then the platform is not supported.

Example: Platforms Section

```
[Platforms]
Windows=Yes
Linux=Yes
Unix=No
Mac=No
```

Script File: Download Primer <Platform> Section

For each supported platform the script file may contain a Download Primer <*Platform*> section. This section lists some primer-specific files that must be downloaded for the primer to work.

Example: Download Primer <*Platform*> Section

```
[Download_Primer_Windows]
Files=2
File1=def_inf.exe
File2=zlib1.dll
```

The following table includes the parameters available in the Download Primer <*Platform*> section:

Parameter	Description
Files	Specifies the number of files to download.
FileX	Specifies the name of the file to download.

Note: The files listed in the example above are the files necessary to decompress files that have been compressed.

Script File: Download <Platform> Section

For each supported platform the script file contains a Download <*Platform*> section. This section describes for the Web Launcher what and how to download files for the specified platform. The following is an example for the Windows platform.

Example: Download <*Platform*> Section

```
[Download_Windows]
Method=webservice
Executable=cmNriAgent.exe
CommandLine=-script .\adv.ini
Files=2
File1=basic_heuristic.exe
File2=w0000001.zml,w0000001.xml
```

The following table includes the parameters available in the Download <*Platform*> section:

Parameter	Description
Method	Defines the method to use to download the files. Currently only "webservice" is supported.

Parameter	Description
Executable	Defines the name of the executable the primer should execute when all files are downloaded.
CommandLine	Specifies the command line to use when executing *Executable*.
Files	Indicates the number of files to download.
FileX	Specifies the name of the file to download. Specify uncompressed name after the comma. **Note:** For more information, see Compressing and Uncompressing Files for Download (see page 373).

More information:

Script File: General Section (see page 370)
Script File: Advanced Section (see page 371)

Compressing and Uncompressing Files for Download

Some files will be fairly large and could take a considerable time to download. For this reason it is possible to compress and uncompress files. To compress files, the def_inf.exe tool must be used. To have the NRI Primer uncompress files after downloading, the compressed name and uncompressed name must both be specified on the "FileX" line, separated by a comma.

Example

```
File2=w0000001.zml,w0000001.xml
```

In this example, the file "w0000001.zml" is downloaded by the NRI Primer and then expanded to a file named "w0000001.xml". Both files are removed when cleaning the work area.

Configure the Additional Inventory Modules for the NRI Agent

This section provides information on how you can configure your own NRI inventory collection modules.

Prerequisites

Before you start configuring the NRI inventory collection modules, make sure that you complete the following tasks:

- Understand the structure and usage of the script file related to the Non Resident Inventory.

  Note: For more information, see the Script File section.

- Prepare the binaries that you plan to use for the creation of the inventory collection module for the NRI agent. The output format of the inventory files generated by these modules must be asset management compliant.

The complete procedure has been explained with the help of different examples. First example explains how you can configure a template inventory collection module for the NRI agent, second example explains how you can configure a user information collection module for the NRI agent, and third example explains how you can configure a .wlscr file to include a localized language. Each example provides information on how to achieve that task.

Create the Template Inventory Collection Module

You can create the template inventory collection module for the NRI agent. The steps involved in this process are as follows:

1. Creating the template file for the NRI agent.
2. Creating the INI file.
3. Creating the .wlscr script file.
4. Copying the files to the appropriate folders.
5. Restarting the Tomcat service.
6. Accessing the NRI web page.
7. Running the new module.

Create a Template File for the NRI Agent

Create a template file that is to be used in the NRI agent. For this example, the content of the template file is taken from asset management and the format of the file is the same as the AM template file. The name of the template file in this example is template.tpl.

The template.tpl file is created with the following content:

```
//

// A MIF File Created with the Asset Management system

//

Start Component

 Name = "User Template"

 UpdateType = "Never"

 UpdateMsg = ""

 LastUpdated = ""

 Description = "Basic Template for Users"

 Start Group

  Name = "User Information"

  Id = 1

  Class = "NETCON|User Information|1.0"

  Description = ""

  Start Attribute

   Name = "User Name"

   Id = 1

   Storage = Specific

   Type = String(32)

   Access = Write-Only
```

```
End Attribute
Start Attribute
 Name = "First Name"
 Id = 2
 Storage = Specific
 Type = String(64)
 Access = Write-Only
End Attribute
Start Attribute
 Name = "Last Name"
 Id = 3
 Storage = Specific
 Type = String(64)
 Access = Write-Only
End Attribute
Start Attribute
 Name = "Address"
 Id = 4
 Storage = Specific
 Type = String(128)
 Access = Write-Only
End Attribute
Start Attribute
 Name = "ZipCode"
 Id = 5
 Storage = Specific
 Type = String(32)
 Access = Write-Only
```

```
 End Attribute
 Start Attribute
  Name = "City"
  Id = 6
  Storage = Specific
  Type = String(128)
  Access = Write-Only
 End Attribute
 Start Attribute
  Name = "Country"
  Id = 7
  Storage = Specific
  Type = String(64)
  Access = Write-Only
 End Attribute
 Start Attribute
  Name = "Home Phone"
  Id = 8
  Storage = Specific
  Type = String(32)
  Access = Write-Only
 End Attribute
End Group
Start Group
 Name = "Company Information"
 Id = 2
 Class = "NETCON|Company Information|1.0"
```

```
 Description = ""
 Start Attribute
  Name = "Department"
  Id = 1
  Storage = Specific
  Type = String(64)
  Access = Write-Only
 End Attribute
 Start Attribute
  Name = "Phone"
  Id = 2
  Storage = Specific
  Type = String(32)
  Access = Write-Only
 End Attribute
End Group
Start Group
 Name = "Additional Information"
 Id = 3
 Class = "NE   TCON|Additional Information|1.0"
 Description = ""
 Start Attribute
```

```
   Name = "Additional Text"

   Id = 1

   Storage = Specific

   Type = String(256)

   Access = Write-Only

  End Attribute

  Start Attribute

   Name = "Default Workstation"

   Id = 2

   Storage = Specific

   Type = String(64)

   Access = Write-Only

  End Attribute

 End Group

End Component
```

Create an INI File

For this example, the INI file (tpl.ini) has been created by copying the content of the basic.ini file. The basic.ini is available in the folder ..\DSM\Web Console\webapps\wac\jsp\ei\wl\windows\packages.

The content of the tpl.ini file is as follows:

```
[General]

OutputFolder=.\

OutputFile=

Origin=CA NRI (Tpl Scan)

Trustlevel=5

TrustlevelRestrictedUser=3

KeepHostkey=Yes

[Inventory]

Method=1
```

```
ScanRestrictedUser=1

AdditionalModules=0

TemplateModules=1

Template1=template.tpl

[Software]

Method=1

ScanRestrictedUser=1
```

The value for the option "TemplateModules" has been changed to 1 (this value was 0 in the basic.ini file.) The value 1 implies that there is only one template file.

The "TemplateModules" option is followed by the option Template1. This option represents the name of the template file. So we have the line as:

```
Template1=template.tpl
```

If you have more than one template to add, the content of the INI file will include additional options as follows:

```
TemplateModules=3

Template1=template.tpl

Template2=mytemplate2.tpl

Template3=mytemplate3.tpl
```

Create a .wlscr Script File

After creating the tpl.ini file, create the .wlscr script file for this template inventory collection module. The name of the .wlscr file for this example is tpl.wlscr.

See the content of the tpl.wlscr file below to note that you need to change the following options based on your requirements:

- Title
- Description

- CommandLine parameters with -origin description
- Number of files to be downloaded
- File name list - You must have the list of the binaries name to put in this script file so that the primer knows which file to download to the target computer. In this example, 5 new files have been added:
  - tpl.ini
  - template.tpl
  - amm2iw32.exe
  - amtplw32.exe
  - amtext.enu

Important! The difficulty in this process is to obtain the number of binaries that can be used to perform this task and the command line that is used to run this module. This requires the expertise of a user who understands the asset management agent functionality so as to define which module to produce which inventory data. The above binaries are distributed with the asset management agent installation.

The content of the .wlscr file is as follows:

```
[General]

Title=Template Test

Description=Select this to perform a fast scan of hardware and software on your computer.

Image=nri_basic_scan.gif

[Advanced]

KeepWorkArea=0

[Platforms]

Windows=Yes

Linux=No

Unix=No

Mac=No

[Download_Windows]
```

```
Method=webservice

Executable=cmNriAgent.exe

CommandLine=-script .\tpl.ini -origin "CA Elective Inventory (Basic Scan+Template)"

Files=20

File1=cmNriAgent.exe

File2=tpl.ini

File3=cainf.dll

File4=cawinexf.dll

File5=cfUtilities.dll

File6=msvcp71.dll

File7=msvcr71.dll

File8=cfTrace.dll

File9=cfTrace.ini

File10=amappw32.exe

File11=cfbasichwwnt.exe

File12=cfXMLParser.dll

File13=cfOSServices.dll

File14=cfig40wnt.sys

File15=cfigw32.sys

File16=cfigw64.sys

File17=template.tpl

File18=amm2iw32.exe

File19=amtplw32.exe

File20=amtext.enu
```

After all the files are created, go to the folder ..\DSM\Web Console\webapps\wac\jsp\ei\wl\scripts. Count the existing number of script files in that folder and increment the count by 1. Include this new number to the name of the tpl.wlscr file. For example, if the folder contains 3 script files, you must name the next file as 4tpl.wlsc.

Copy the Files to the Appropriate Folder

Copy the other files to the ..\DSM\Web Console\webapps\wac\jsp\ei\wl\windows\packages folder.

In this example, you need to copy the following files to the folder:

- tpl.ini
- template.tpl
- amm2iw32.exe
- amtplw32.exe
- amtext.enu

Restart the Tomcat Service

Restart the Tomcat service by using the following commands:

```
caf stop tomcat
```

```
caf start tomcat
```

Access the NRI Web Page

Enter the following URL to navigate to the NRI web page:

```
http://<webserver-hostname>/wac/jsp/ei/ei.jsp
```

You can see that the new inventory collection module is added to the list.

Run the New Module

Click the new module to run it.The template information is collected and added to the inventory data set.

Create the Additional Inventory Collection Module

In this example, a user account inventory collection module is created for the NRI agent.

The procedure to create the additional inventory collection module is very similar to the creation of the template inventory collection module.

Important! Before you start, make sure that you have obtained all necessary binaries that are to be used for this inventory module. You also need to find out the execution of the command line of these modules if you run them from the command prompt. You also need to know the parameters to pass, the output files that are generated, and so on.

The steps involved in this process are as follows:

1. Creation of the INI file
2. Creation of the .wlscr file
3. Follow Steps 4-7 as explained in the Create the Template Inventory Collection Module section.

Create an INI file

The binary "amusernt.exe" is used with the additional DLLs. Using the basic inventory file list along with this additional binary will make this inventory binary work. The command line to call this binary is as follows:

```
amusernt.ext invdata.inv
```

Based on the above information, the ini file of this additional inventory collection module is created. The file is named as addmod.ini in this example.

See the content of the file below to notice that you need to change the following options in the ini file based on your requirements:

- Origin
- AdditionalModules: enter the value as 1. That means you have one additional module to run. If you have more than one additional module to add, enter the appropriate number.
- With the AdditionalModules field as non zero value, you need to add the following additional fields based on the basic.ini file:
  - Module1Bianry=
  - Module1Commandline=
  - Module1Outputfile=
  - Module1cleanup=

Note: If the value in the AdditionalModules field is 2, the second set of module information would be as follows:

- Module2Binary=xxxx.exe
- Module2Commonline=.\xxxx xxxxx
- Module2Outputfile=.\xxxx.txt
- Module2Cleanup=.\xxxx.txt

The content of the file is as follows:

```
[General]

OutputFolder=.\

OutputFile=

Origin=CA NRI (User Account Scan)

Trustlevel=5

TrustlevelRestrictedUser=3

KeepHostkey=Yes

[Inventory]

Method=1

ScanRestrictedUser=1

AdditionalModules=1

Module1Binary=amusersnt.exe

Module1Commandline=.\users.inv

Module1Outputfile=.\users.inv

Module1cleanup=.\users.inv

TemplateModules=0

[Software]

Method=1

ScanRestrictedUser=1
```

Creation of the .wlscr File

The following is an example of the .wlscr file to be used in this example. The file has been named as 5addmod.wlscr. You can change the Title and Description of the file as described below. Additionally, you also need to look at the highlighted fields to see if these fields require any change:

```
[General]

Title= Additional Module Test

Description=Select this to perform a fast scan of hardware and software on your computer plus user account inventory

Image=nri_basic_scan.gif

[Platforms]

Windows=Yes

Linux=No

Unix=No

Mac=No

[Download_Windows]

Method=webservice

Executable=cmNriAgent.exe

CommandLine=-script .\addmod.ini -origin "CA Elective Inventory (Basic Scan+User module)"

Files=17

File1=cmNriAgent.exe

File2=addmod.ini

File3=cainf.dll

File4=cawinexf.dll

File5=cfUtilities.dll

File6=msvcp71.dll

File7=msvcr71.dll

File8=cfTrace.dll

File9=cfTrace.ini
```

```
File10=amappw32.exe

File11=cfbasichwwnt.exe

File12=cfXMLParser.dll

File13=cfOSServices.dll

File14=cfig40wnt.sys

File15=cfigw32.sys

File16=cfigw64.sys

File17=amusersnt.exe
```

Follow Steps 4-7 (already explained in the above section) to complete the process.

Configure the Localized .wlscr Files

You can modify your own version of the .wlscr file to have a localized language to be included in the file. The NRI inventory options are then displayed in the localized language when the NRI link is accessed.

To configure the localized .wlscr files

1. Open the .wlscr file, for example, 5addmod.wlscr.

   The file opens.

2. Modify the section [General] in this file to add your own language. For example, the German language name and the description has been added to the 5addmod.wlscr file as follows:

   ```
   [General]

   Title= Additional Module Test

   Description=Select this to perform a fast scan of hardware and software on your computer plus user account inventory

   Image=nri_basic_scan.gif

   Title_DE=xxxxxx

   Description_DE=xxxxx……..
   ```

   The information is displayed if you have installed the DSM server on a computer with German operating system.

   Note: The following are the language tokens that are to be used for other localized languages:

   - _ENU: (default) No token is needed
   - _DE: German

- _FR: French
- _JPN: Japanese
- _ES: Spanish
- _IT: Italian
- _KOR: Korean
- _PT: Portuguese

3. Save the file and close it.

   All the settings are saved.

4. Restart the Tomcat service by using the following commands:

   ```
   Caf stop tomcat
   ```

   ```
   Caf start tomcat
   ```

   The Tomcat service is started.

5. Enter the URL to access the NRI web page.

   Your own version of the localized language for the NRI inventory options is displayed.

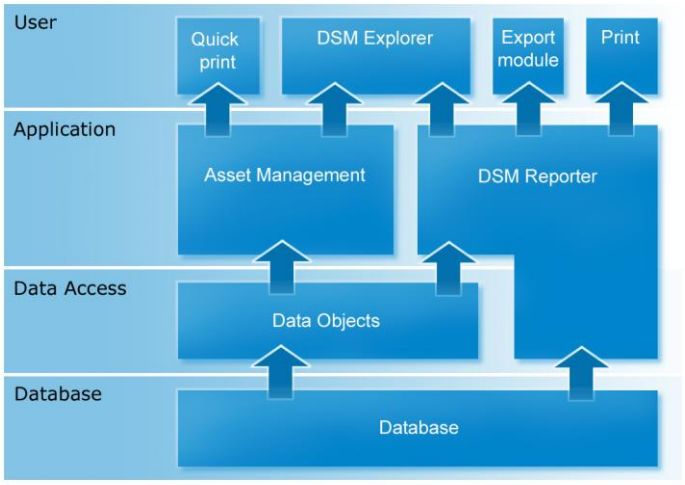
DSM Reporter

The DSM Reporter tool extracts inventory information from the database. Using the Query Designer, the user is able to evaluate this data and specify which fields are provided on the reports. The information extracted can be filtered, sorted, summarized, and viewed immediately in the GUI, and can be utilized for report printouts or exporting to other applications (for example as CSV file or HTML file). Reports are displayed hierarchically in an Explorer-like tree view, and a number of predefined result sets can be kept for each report as auto history. In order to prevent a result set from getting deleted when obsolete, the user can choose to save it.

Report generation can be user initiated or scheduled. Scheduling of reports is done in the Reporter GUI. The Engine will then initiate the report generation itself, according to the scheduling properties.

Note: When you launch the DSM Reporter for the first time after installation, make sure that you give enough time to import all the report templates into the database. However, if anything goes wrong while importing the report templates, as a workaround, open the registry editor and delete the subkey in HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\Unicenter ITRM\Reporter\Library. After deleting the subkey, launch the DSM Reporter again.

The following graphic shows how the DSM Reporter is embedded in the asset management environment:



Note: For more information about the predefined reports and DSM Reporter, see the *DSM Reporter Help*.

Features

Using the Reporter, you can create reports based on the following categories:

Computers

Includes reports for computer specific information such as manufacturer information, service level, summary, user relation, virtual machine reports and so on.

DSM Infrastructure

Includes reports for the engine information and information on all managed assets in the domain.

Hardware

Includes reports for the hardware information.

Network

Includes reports for the network information.

Policies

Include reports for the policy related information.

Unmanaged Assets

Includes reports for the unmanaged assets.

Users

Includes reports for the user specific information.

Using these types of data, you can create sophisticated reports with little effort. You can create reports based solely on, for example Software Inventory or you can mix data of all seven kinds. Thus, for example, you could find all computers that have the Netscape Navigator program installed, and then have the report display the names of the people who use these computers. The Reporter lets you query the database for the various data gathered by the agent from each computer in the domain.

The Reporter features include for example:

- Integrated explorer-like handling of Report Templates and Results, featuring user definable unlimited folder and template structure
- Results can be viewed immediately in the GUI
- Results can be exported to files in various formats
- Reporting on general asset properties, software properties, properties for executed jobs and modules
- Reporting on current inventory values as well as most recent, previously collected values
- Reporting on non-specific logical inventory groups, like All Processors and All Local Drives

- Point-and-click filter editor with color syntax hi-lighting. Default filters for Report Templates as well as individual filtering of each result set
- Creation of SQL scripts for running a report
- Printing of HTML reports that can be customized

Start the DSM Reporter

You can start the DSM Reporter in one of the following ways:

- From Windows, click Start, Programs, CA, IT Client Manager, DSM Reporter
- From the DSM Explorer menu bar, click Tools, DSM Reporter

Using the Reporter window you can view reports, browse for report templates and scheduled reports, and create new report templates. Click either of the items in the Information window in the right pane or expand the appropriate Contents item.

Selecting a node in the tree view displays corresponding information or sub items in the Reporter pane. Using property sheets, you can add new items and edit the existing ones. On the other hand, all essential information is displayed directly in the DSM Explorer, so that opening property sheets or dialogs is generally not necessary for viewing data. HTML views provide the user with a quick overview, even for complex report definitions.

Specify Reporter Preferences

Reporter preferences define the settings for administrators and all users. You can customize the reporter preferences to present the report the way you want it.

To specify reporter preferences

1. Click Tools, Preferences from the menu bar.

   The Preferences window appears.

2. Click the administrator tab and specify the settings for the administrator:

   Information Section

   Defines whether you want to display the tips in the report's information pane or use old style message boxes. Tips are marked with a yellow bulb icon.

Run Report Section

Defines the user's personal preferences regarding the appropriate actions to take place after running a report.

Results Section

Defines the report viewing settings.

3. Click the Global Settings tab and specify settings that affect all users of the DSM Reporter and can only be changed by the Administrator.

Auto Results Section

Specifies the number of auto-history result sets to be kept under Results in the Contents tree. Older results are deleted permanently from the database. With respect to the user interface, the administrator even may choose to delete folders that still contain files.

Scheduled results to keep by default Section,

Specifies the number of scheduled results to be kept by default.

Restrictions Section

Specifies that only administrators can reset statistics and that only administrators can edit result count for scheduling.

User Interface Section

Allows the deletion of non-empty folders, performance of a consistency check and so on.

Enterprise Section

Lets you include the domain name in the new and imported templates when the reporter runs in at the enterprise manager.

Note: For more information about the Preferences dialog, see the *DSM Reporter Help*.

Working with the Reporter

The basic tree structure of the DSM Reporter consists primarily of folders and report templates. The structure is similar to a file structure. You can create and organize folders and report templates, to suit your needs. The structure also holds items representing primary properties for report templates and scheduled reports, and items representing existing result sets.

Create New Report Template

Report templates contain the definitions for a report. Once created, you can either run these reports or schedule it to run at specific times.

To create a new report template

1. Right-click the appropriate folder under the Report Templates folder and select New Report Template.

   The Report Template Type dialog appears.

2. Select the type of the report template and click OK.

   Based on the selected report type, the New *report type* dialog appears.

   Note: For more information about the Report Template Type dialog and the New *report type* dialog, see the *DSM Reporter Help*.

Run a Report

To run a report immediately and add to the report's Results set, right-click the report and select Run Report.

Export Report Templates

You can also export report templates from one domain to another domain in an enterprise to reuse the report templates.

To export reports from the selected folder (including sub folders)

1. Right-click any folder under the Report Templates folder and select Export All Report Templates.

   The Export Report Template dialog appears.

2. Specify the destination folder and select the Export templates from subfolders recursively checkbox and click OK.

   The report templates available in the selected folder and all its subfolders are exported to the destination folder.

Note: You can also export a particular report to a file, by right-clicking the report and selecting Export, Report Definition.

Import Report Templates

You can reuse the report templates created in a domain by exporting and importing it in some other domain.

To import a report template

1. Right-click the Report Templates folder and select Import Report Template

   A search dialog is opened.

2. Browse for the appropriate report definition file (.rep) and click Open.

   The Template Report is shown immediately under the Report Templates folder and in the information pane as well.

Schedule a Report

You can schedule a report to run at a specific time.

To schedule a report

1. Right-click the report and select Schedule Report.

   The Properties dialog of the report appears.

2. Click Set Scheduling in the Scheduling tab.

   The Scheduling Properties dialog appears.

3. Specify the scheduling options and click OK.

   The report is scheduled to run at the specified time.

Exporting SQL Scripts

The Reporter generates the SQL script for the purpose of exporting it, and the option is only available if all fields and options of the report template support it. The fields that require local processing cannot be expressed in SQL, and templates containing such fields cannot be exported as SQL scripts.

You can export the SQL scripts to the clipboard or to a file. Right-click the report and select Export, Export SQL script to clipboard or Export SQL script to a file respectively.

Report Output Options

The basic format of a reporter result set consists of two tables in the database: one containing rows of objects, for example, assets, and columns corresponding to the fields included in the report, and one supplying meta data for the result set, such as the full names of included inventory items and more. The later is referred to as the “map-table,” whereas the first mentioned is considered the actual result table.

Viewing Results in the GUI

Any result set can be viewed entirely in the GUI simply by selecting it in the tree view. The Information pane shows all assets matching the criteria of the report definition. The Information column views common result information and provides the ability of filtering the result (can be toggled on or off to compare the full result set with the subset defined by the filter) or viewing the legend of a field.

Preview and Print Results

Result sets can be previewed in HTML view and also printed. The print result is based on the HTML generation, which can be customized by the user. Customization includes table generation properties, sorting and more. Furthermore, the user can access and modify the Cascading Style Sheet, which defines most default formatting options for the HTML generation.

Note: You need to run the report to preview or print its result.

To preview and print the results, right-click the result and select Print. The HTML report is displayed as a preview in a separate window. This window has the following buttons.

Print

Opens the Microsoft Windows Print dialog.

Page Setup

Opens the Microsoft Windows Page Setup dialog for specifying the page properties.

Options

Gives access to any available layout and customization option for the result.

Close

Closes the current window.

Exporting Results

Built-in export modules for various CSV and HTML formats make it easy to export the results for use in external tools. All exports can be customized from the GUI, and individual customizations are saved with each scheduled report, to make publishing of scheduled reports very flexible. To export the results, right-click the result and select Export, the required export format.

CA Asset Converter for Microsoft SCCM

The CA Asset Converter for Microsoft® SCCM is a component that connects to a Microsoft System Center Configuration Manager (SCCM) and extracts information, including hardware and software inventory, about the computers stored in its database.

CA Asset Converter adapts this information into a form resembling as closely as possible the information that would have been collected by a native DSM asset management agent. The information is then written into a collection of XML files in the format that is required by CA Asset Collector. Once these files have been picked up by CA Asset Collector, the computers from the Microsoft SCCM database appear in the CA MDB as discovered hardware assets alongside those found by the native DSM agents. The associated hardware and software inventory information is also stored and structured in the same way.

CA Asset Converter can be configured to collect only hardware inventory, only software inventory, or no inventory information at all. In the latter case, only the computer names and a few basic attributes will be extracted.

The CA Asset Converter for Microsoft SCCM engine task is available on Windows only and is certified against Microsoft Systems Center Configuration Manager (SCCM).

This engine task is created, configured, assigned, scheduled, and run in the same way as existing DSM engine tasks.

Important! Licensing Information. CA Asset Converter for Microsoft SCCM is a separately licensed component. Contact your CA representative to purchase the required license before using this component. Such unlicensed product component may be used for a 30-day trial period. After this period has expired, you must purchase a license to continue using this component. Failure to do so will violate the license agreement.

Note: The CA Asset Converter for Microsoft SCCM is installed as part of the installation of the DSM engine on a Windows system. However, you can configure it to deliver its output to an instance of the Asset Collector other than the one with which it is installed.

More information:

Mapping of Microsoft SCCM Hardware Asset Inventory to CA ITCM Asset Management Hardware Inventory (see page 533)
Mapping of Microsoft SCCM Software Asset Inventory to CA ITCM Asset Management Software Inventory (see page 537)

Create a CA Asset Converter for Microsoft SCCM Engine Task

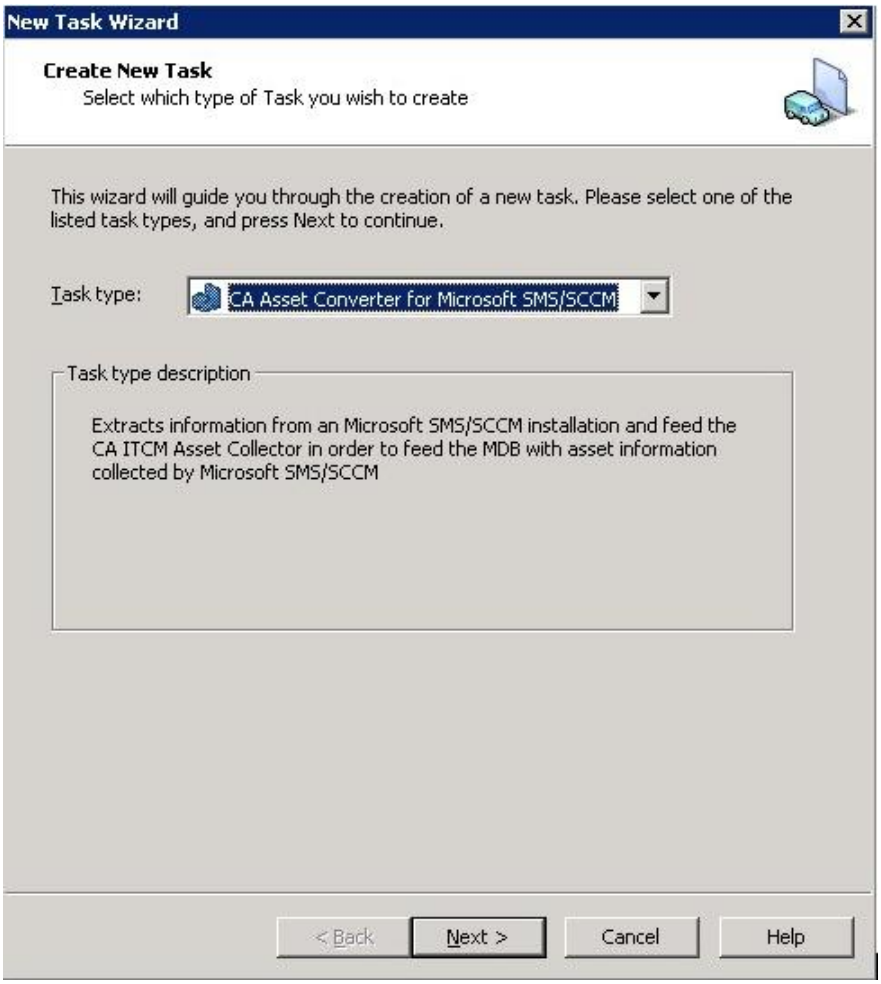
To configure the CA Asset Converter for Microsoft SCCM, you need to first create an engine task.

Note: A CA Asset Converter for Microsoft SCCM engine task cannot be assigned to an engine that is running on the same computer as the Microsoft SCCM site server from which it is meant to extract data.

To create a CA Asset Converter for Microsoft SCCM engine task

1. Open the DSM Explorer and navigate to Control Panel, Engines, Engine Tasks.
2. Select New from the Tasks area of the information pane.

   The New Task Wizard for engine tasks appears.
3. Select CA Asset Converter for Microsoft SCCM from the Task type drop-down list, as shown in the following illustration:



4. Click Next.

The remaining screens are identical to the Properties dialog of an existing CA Asset Converter for Microsoft SCCM task and are described in the next section.

Task Properties

A CA Asset Converter for Microsoft SCCM task is configured either during its creation in the New Task Wizard or, for an existing task, through its Properties dialog. In either case, the configuration GUI is the same except that you navigate between the pages using the Next and Back buttons whereas the Properties dialog has tabs.

Description

The first page, which is common to all engine tasks of any type, allows you to specify a name and description for the current task. The name must be unique and nonempty. The description is optional.

Configure the SCCM Server Option

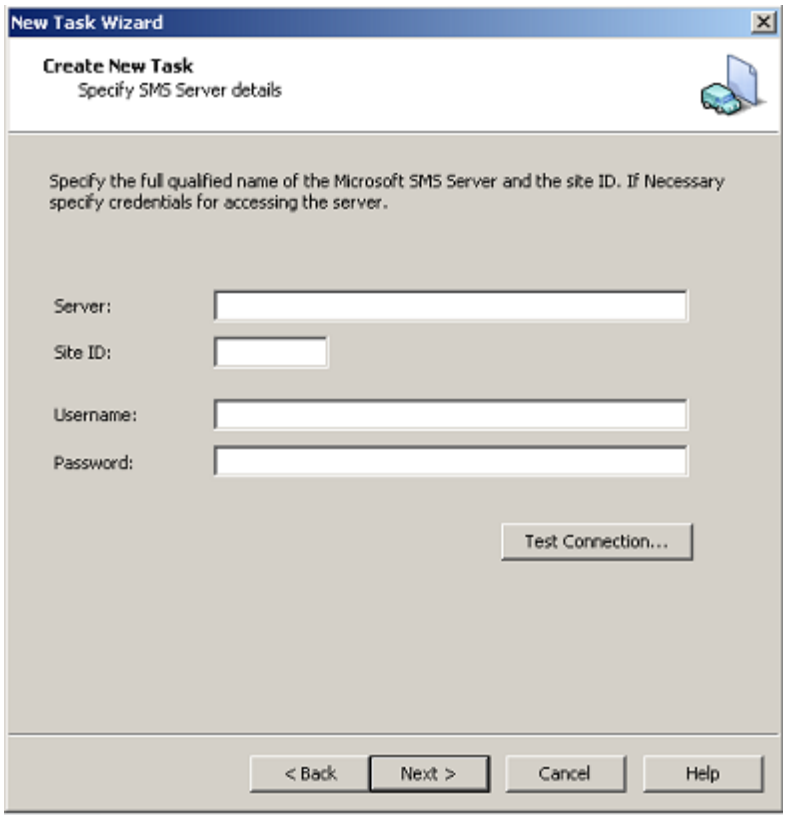
The SCCM Server tab is used to provide information about the SCCM Site server that will be the target for the extraction.

Use this procedure to configure the SCCM server option.

To configure the SCCM Server option

1. In the Server field, enter the host name or address of the SCCM Site server from which to extract information.

   The following illustration shows all the SCCM server options:



2. In the Site ID field enter the site ID of the SCCM installation to contact.

   Each SCCM Site installation has a three-letter site ID. It is possible to install several SCCM sites on the same server as long as they have different site IDs, and, in such cases, this field must be used to specify which site the Asset Converter task connects to.

   When only one SCCM site is present on the server, this field should be left empty, as the Asset Converter automatically chooses the only available site ID.

3. In the Username field, enter the domainname/username to use when connecting to the SCCM Site server. If left empty, the current user context of the engine performing the task is used.

4. In the Password field, enter the password to use when connecting to the SCCM Site server. If left empty, the current user context of the engine performing the task is used.

   The user specified must be able to access (remotely) the SCCM WMI classes on the server machine.

5. Click the Test Connection button to determine if the server details currently specified allow a connection to an SCCM site to be established.
6. Click Next.

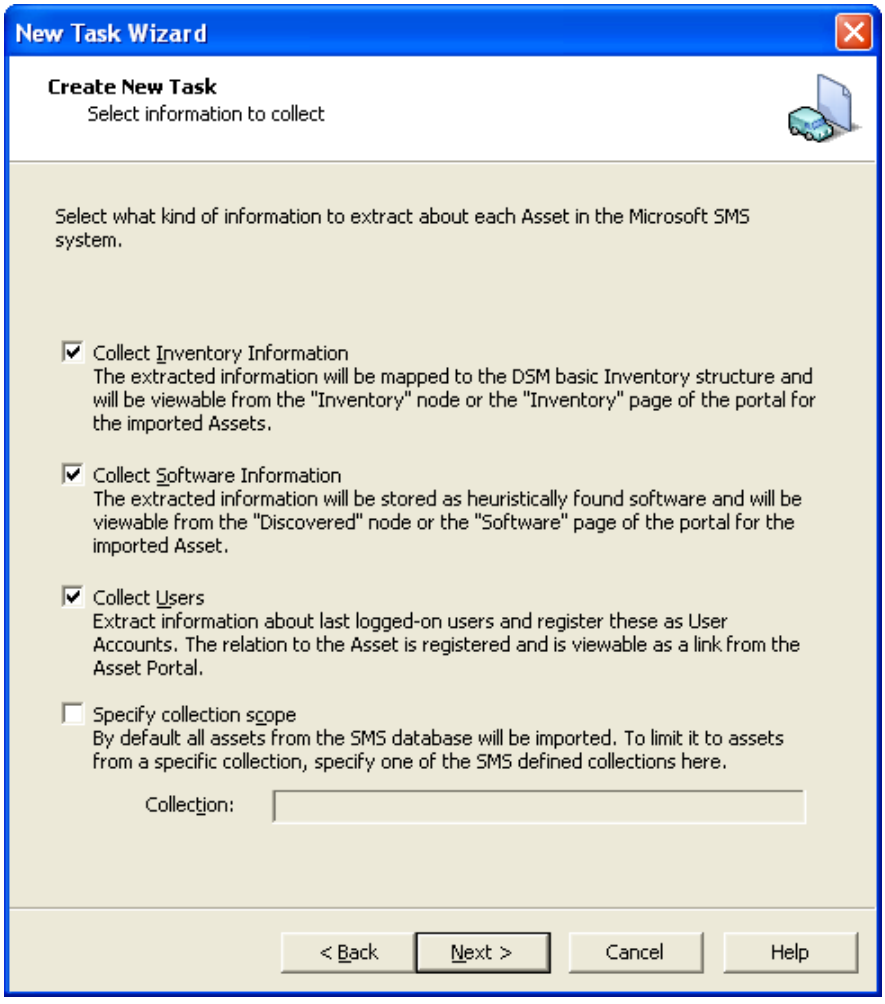
A successful result here does not guarantee the same outcome from the task once it is performed by an engine, as it could then be running on a different machine and in a different user context. The test performed when you click on the Test Connection button in the Create New Task dialog is performed on the machine running the GUI and in the context of the user running the GUI. When the actual SCCM Converter task is run it is done on the engine machine and in the context of the user account used by the engine (the local system account by default). This means that a successful test does not guarantee a successful connection when the task is actually run. You can eliminate the potential differences by making sure you include the domain name in the Username field of the configuration (in this format: domainname\username). If this is not done it will default to the domain of the current user which for a default engine amounts to the local users of the engine machine. If possible the test should also be performed on a machine which has the exact same access to the SCCM installation as the engine machine meant to perform the conversion.

Configure the Collect Options

The Collect Options tab contains options to independently enable or disable the collection of hardware or software information.

To configure the Collect Options

1. Choose and select from the following options, Collect Inventory Information, Collect Software Information, Collect Users, and Specify collection scope, for example:



2. Click Next.

By default this information is collected from all computers in the SCCM installation with an SCCM client installed.

It is possible to limit the scope of the extraction to one specified collection of computers from the SCCM site. To do this, select the Specify collection scope check box and enter the name of the SCCM collection to extract in the Collection field.

Configure the Delivery Location

The Delivery Location tab specifies where the XML files produced by the CA Asset Converter for Microsoft SCCM task are to be placed. This would typically be the collection folder of an Asset Collector installation.

Use the following procedure to configure the Delivery Location.

To Configure the Delivery Location

1. Select an option from the following options:

   Use Local Asset Collector

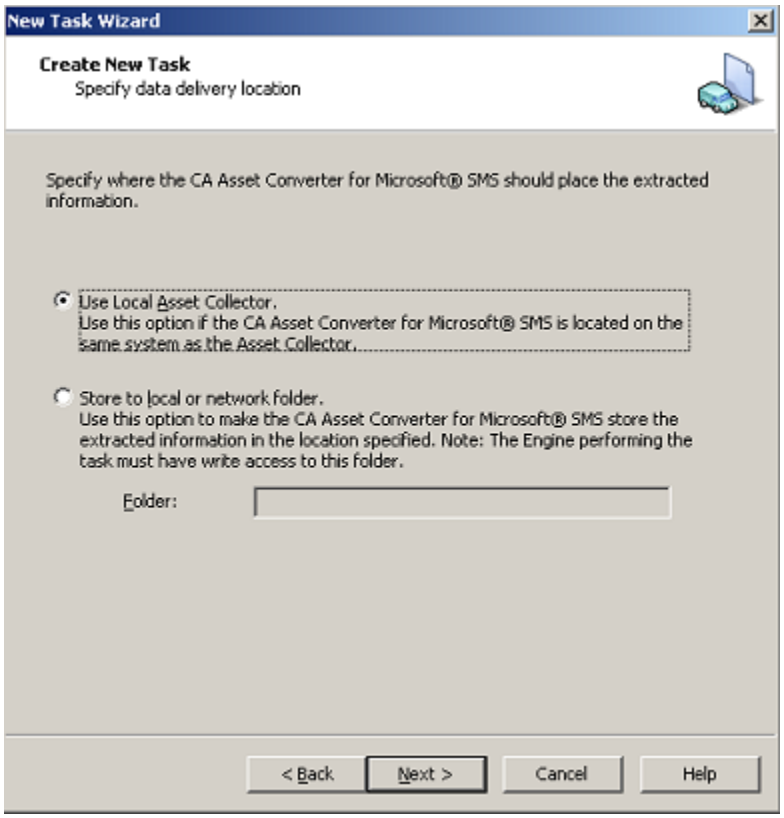
   Specifies that the files are written to the collection folder of the Asset Collector running on the same computer as the CA Asset Converter for Microsoft SCCM task. This is the default setting.

   Note: If the computer has no Asset Collector installed, the CA Asset Converter for Microsoft SCCM task will fail.

Store to local or network folder

Specifies that the files are placed in the specified local or network folder.

For example, in the following illustration, the Use Local Asset Collector option is selected:



2. Click Next.

Scheduling

The CA Asset Converter for Microsoft SCCM task uses the same scheduling dialog as all other engine tasks. As a full extraction of SCCM data may be time consuming and may create load on the SCCM database server, it is recommended that the task be scheduled at a time when the server is most likely to be idle.

Task Management

You can run the CA Asset Converter for Microsoft SCCM task and review its status.

Run a Task

Once the task has been created and configured, it must then be linked to an engine in order to be run.

To run a CA Asset Converter for Microsoft SCCM task

There are several ways of doing this:

- Drag and drop the SCCM task onto the appropriate engine in the tree view of the DSM Explorer.
- Navigate to the engine in the tree view and click Link Existing Task in the information pane. Then select the CA Asset Converter for Microsoft SCCM task in the dialog that appears.
- Navigate to the engine in the tree view and click Add New Task. The Create New Task wizard appears, allowing the creation, configuration, and linking of the task all to be accomplished in one step.

Once linked, the engine will perform the task the next time it comes up in its task list and the scheduling configuration allows it to be run.

Task Status

The status of the task is shown in the engine's task list. It can have the following values:

Waiting

The engine has not yet run the task.

Active

The task is currently running.

OK

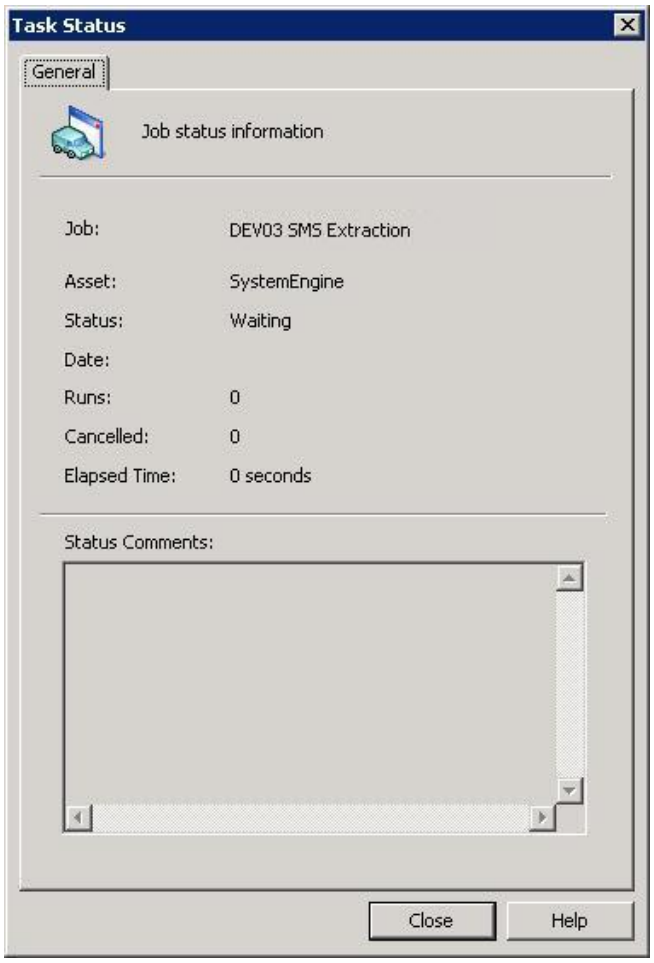
The task has been run and an “OK” status was returned.

ERROR

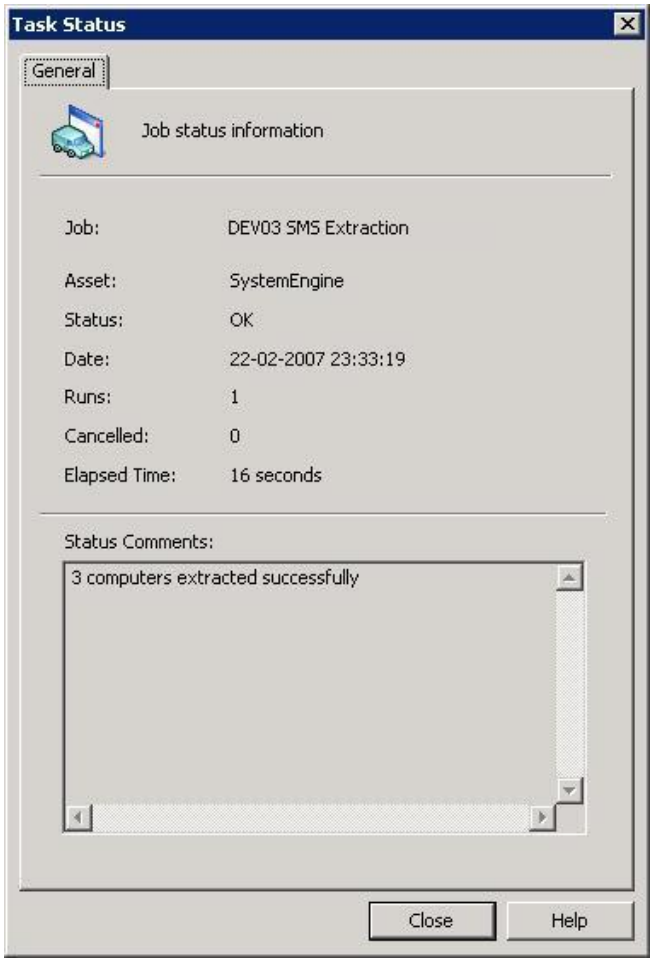
The task has been run and an “ERROR” status was returned.

The task list also shows the time when (if ever) the task was last executed.

Example Showing a CA Asset Converter for Microsoft SCCM Task with Waiting Status



Example Showing a CA Asset Converter for Microsoft SCCM Task with OK Status


Task Status
General
Job status information
Job:
DEV03 SMS Extraction
Asset:
SystemEngine
Status:
OK
Date:
22-02-2007 23:33:19
Runs:
1
Cancelled:
0
Elapsed Time:
16 seconds
Status Comments:
3 computers extracted successfully
Close
Help

SCCM Converted Data in the CA MDB

Once an XML file produced by the CA Asset Converter for Microsoft SCCM has been picked up by an Asset Collector, the computer it represents will appear in the DSM GUI alongside the other discovered hardware assets.

The following illustration shows that the Agent Status section has no DSM asset management agent installed on the computer and that the origin of the computer is Microsoft SCCM.

Diagnostics and Tracing

The common DSM tracing system is used to record detailed information while a CA Asset Converter for Microsoft SCCM task is being run. If the appropriate configuration step has been performed, the trace file used is TRC_SMSEX.log. Otherwise, it is directed to the TRC_UAM.log file where it may be interspersed with information from other DSM components.

Command Line Utility am_sms_ex.exe

The executable performing the actual SCCM extraction work, *am_sms_ex.exe*, may be launched manually for testing or debugging purposes.

The available command line parameters are:

/s:<*servername*>

Specifies the name or address of the SCCM Site server from which to extract. Defaults to localhost when not supplied.

/u:<*username*>

Specifies the user name to use when connecting to the server. Defaults to current user.

/pp:<*password*>

Specifies the password to use (in plain text). Only used when a user name has also been given.

/c:<*site id*>

Identifies the site ID of the SCCM installation to contact. Only required if the server contains more than one SCCM installation.

/d:<*output directory*>

Indicates the directory in which to place any XML files that are produced.

/collect_sw

Extracts software information.

/collect_hw

Extracts hardware information.

/collect_users

Extracts the last logged on user for each computer as a related asset.

Chapter 6: Using the System Engine

The system engine is the central component in the architecture and acts as a primary communication point between the servers and the database. The system engine is available at two levels and performs different functions at each level:

System Engine - Enterprise Manager

The system engine in the enterprise manager performs the data processing. This engine can do the following:

- Schedule report templates to get data from the enterprise database or domain databases.
- Execute queries, SQL Scripts and external utilities.
- Execute the Default Software Contents Download Job for updating the enterprise manager's Software Definition list.

System Engine - Domain Manager

The system engine in the domain manager performs the following tasks:

- Schedule report templates to get data from the domain database
- Data collection
- Data distribution
- Data replication from and to the enterprise manager
- Execute queries, SQL Scripts and external utilities
- Execute the Default Software Contents Download Job for updating the domain manager's Software Definition list.
- Execute the Default Directory Synchronization Job

This engine usually runs on a dedicated Windows computer. However, it can also run on the domain manager. Running the engine on a dedicated computer improves the response time of the domain manager when queries are made, and when inventory updated from many computers is processed.

This section contains the following topics:

Create an Engine Instance

You can create multiple instances of engines on the same computer. On the domain manager computer, the *main* engine is called SystemEngine; while on a remotely installed system, it is called *computername_Engine*. You can right-click any such engine and create its new instance. The new instance will reside on the same computer.

Having additional engine instances is useful in the cases where:

- The re-scheduling interval of tasks is too large due to the number of scheduled engine tasks.
- The tasks that might take a long time to finish and would exclude other tasks from being processed in a reasonable time period.

To create a new instance of the engine

1. Right-click System Engine under the Control Panel folder, and select Create new instance.

   The Create new Engine instance dialog opens.

2. Enter a unique name for the engine in the text field, and click OK.

   The newly created engine appears under the All Engines folder.

   Note: The new instance runs on the same computer as the selected engine. To distribute engines on the other computers, you need to install a new engine on the chosen computer using the CA ITCM installer.

Configure Engine

You can configure an engine to have optimum utilization and to improve the performance of the engine.

To configure the engine

1. Navigate to the Control Panel, Engines, All Engines folder.

   The existing engines in the domain are displayed under the folder.

2. Right-click the engine and select Properties.

   The Properties for *Engine_Name* dialog appears.

3. Click the Advanced tab and specify the following settings:

   Number of files the engine will collect during one collect cycle

   Specifies the number of files you want the engine to collect during one collect cycle. By default this number is set to 10000. This means, the engine will collect 10000 files during one collect cycle and then let the engine tasks in queue to execute before it starts the next cycle. The next collect cycle begins after the interval specified in the field below.

   Default: 10000

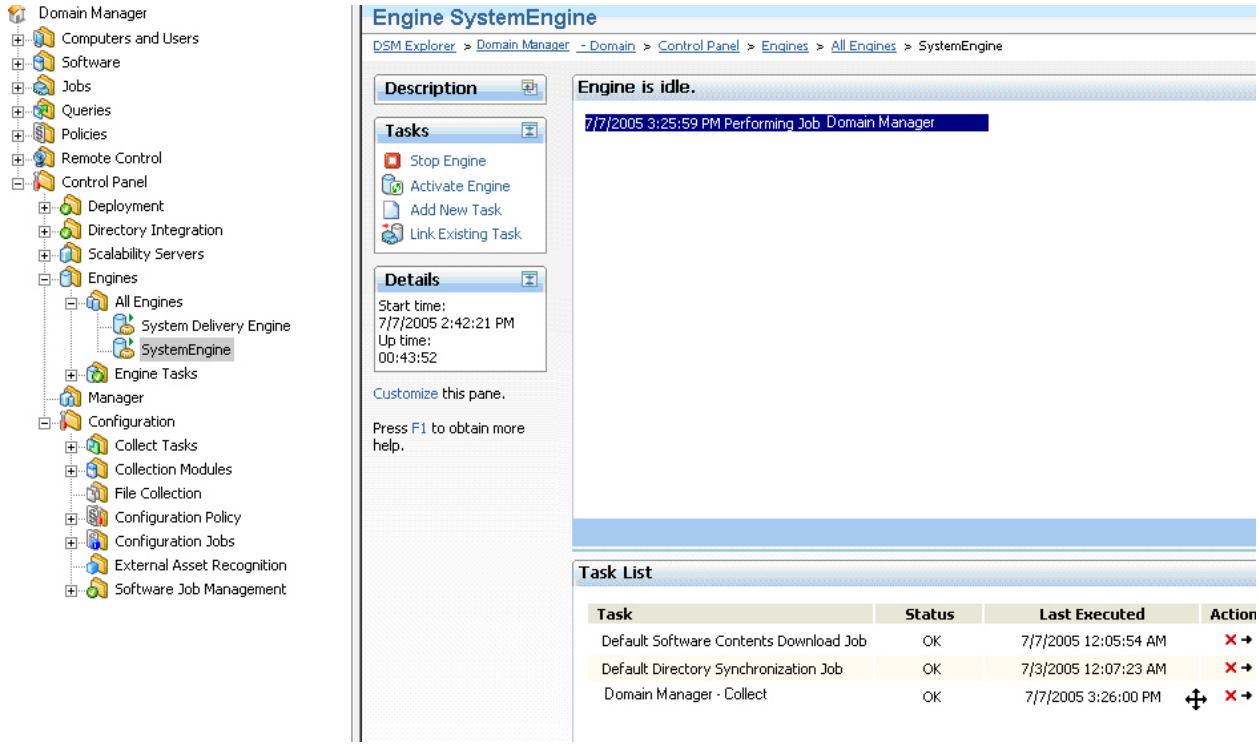
   Time Interval in seconds between Engine Tasks

   Specifies the time interval in seconds between the engine task runs. By default this interval is set to 60 seconds. This means, the engine collects the data from the scalability server every 1 minute. If you have any engine task that runs for a long time, change this interval to accommodate the engine task execution.

   Default: 60

Engine Log

The engine or engine instance logs all its actions in a log file. You can view this log file to learn when the engine collected data from the scalability server or to troubleshoot any engine-related problems. A brief overview of this log is available in the SystemEngine pane:



The complete engine log is available in C:\Program Files\CA\DSM\Logs\SystemEngine.log for the system engine and <engine name>.log, respectively, for other engines.

Stop and Start the Engine

The domain manager runs the engine automatically at the startup time. You can however, stop and start the engine at any time to perform all the tasks in the task list again.

To stop and start the engine

1. Navigate to the Control Panel, Engines, All Engines, SystemEngine folder.

   The right pane displays the engine status and the tasks list.

2. Click Stop Engine in the Tasks section.

   The engine is stopped. If the engine is running any task at this time, it completes the task before stopping the engine.

3. Click Start Engine in the Tasks section.

   The engine starts and performs the tasks in the task list.

   Note: Only the users who have administrative privileges on the computer that is running the engine can start or stop the engine remotely.

Job Performance

The system engine executes the following jobs on the domain manager:

- Predefined tasks
- Engine tasks

Predefined Engine Tasks

The predefined tasks are linked to the engine and run immediately after starting the engine. The following predefined tasks are available:

Default Software Contents Download Job

Downloads new software signatures from the CA Online Content Service to the database. After the software content download is complete, you must be sure to restart the DSM Explorer and the AM manager, respectively. Otherwise, the downloaded software will not be reflected in the DSM Explorer GUI.

To restart the AM manager, use the following command:

```
caf stop ammanager
caf start ammanager
```

Important! Do not run CAF Stop while the software signatures are being downloaded. The download process updates many important tables in the database which needs to be consistent. So, if you kill the process, data can become inconsistent. The download process is a java executable, so check for java.exe in the process list and check that the job 'Default Software Contents Download Job' is not running by navigating to the System Engine in the Control Panel.

Default Directory Synchronization Job

Executes the directory synchronization job configured under Control Panel, Directory Integration, Directory Synchronization folder.

Default SCAP Checklist Processing Job

Processes SCAP checklists located in the *ITCM_installpath*\SCAP_Checklists folder. If there are new or updated checklists, this job creates a compressed archive of the checklist files, and creates or updates inventory detection modules for the new or updated checklists.

Engine Collect Task

Executes the engine collect task to collect the inventory information such as hardware, software, usage, and so on.

Note: Modifying, unlinking, or deleting the predefined engine tasks can impact some of the functionalities in the CA ITCM system.

More information:

Engine Collect Task (see page 151)

Engine Tasks

You can link the engine tasks to the engine. Once linked, the engine adds the tasks to the task list and performs them in an order.

The following engine tasks are available:

Collect Task

Instructs the engine to collect the inventory information from the scalability server.

Query Task

Instructs the engine to run a query and return the results to the DSM Explorer.

SQL Script Task

Instructs the engine to run an SQL Script.

External Utility Task

Instructs the engine to run an external utility.

Note: Make sure that the external utility does not stop responding, for example, waiting for any user interaction. Such a situation would cause the engine to stop processing all other tasks linked to it.

Database Synchronization Task

Instructs the engine to synchronize CA ITCM assets and inventory data that are collected in a Microsoft SQL server (on Windows) of the DSM domain or enterprise manager with the accordant data in the remote SQL server (on Windows) or Oracle DBMS (on Solaris).

CA Asset Converter for Microsoft SCCM task

Instructs the engine to connect to a System Center Configuration Manager (SCCM) and extract information, including hardware and software inventory, about the computers stored in its database.

CA Repository Extraction

Extracts package metadata from the repository and lets you view the Debian package details on DSM Web Console. For more information please see *Extract Package Metadata from the Repository* in the Software Delivery Administration Guide.

Scheduled Report Task

Instructs the engine to run a scheduled report.

Note: For more details about each engine task, refer to the Asset Management section of the *DSM Explorer Help*.

Replication Job

When a domain manager is connected to an enterprise manager, the ITCM automatically creates the upload replication and the download replication tasks. The ITCM links the two replication tasks to the system engine in the domain manager. The engine runs the download replication job immediately, followed by the upload replication job to replicate the data in both the directions between the domain manager and the enterprise manager. The upload replication task occurs in two stages:

- Delete the data from the Enterprise Manager (upload-delete stage).
- Update the data into the Enterprise Manager (upload-update stage).

Similarly the download replication task involves deleting data from the Domain Manager (the download-delete stage) and updating data into the Domain Manager (the download-update stage).

The tasks have the following names by default:

- <host-name> Upload Replication
- <host-name> Download Replication

Notes:

- The replication tasks are created on the Domain Manager only.
- Before you run the replication job, ensure that the domain manager and the enterprise manager are at the same patch level.

By default, the replication job is scheduled to run always. You can change the job properties to schedule it differently.

Note: If the domain data is not replicated on the enterprise manager, verify the status of the replication and engine collect tasks at the domain manager. When the status of both jobs is OK, run the following commands at the domain manager:

```
caf stop engine
caf start engine
```

Chapter 7: Diagnostics and Troubleshooting

The following sections describe different troubleshooting actions that can help you solve problems when using asset management.

Log File Collection Tool dsminfo

CA Technologies provides the dsmInfo tool, which collects diagnostic information from systems that have CA ITCM installed. The data collected is compressed into a single file that contains log files, system information, directory structures, and registry and environment information. This diagnostic tool is available in the CA ITCM product installation media under the DiagnosticTools folder.

If a problem with CA ITCM is reproducible, then run the following command to change the trace level to DETAIL:

```
cftrace -c set -l DETAIL
```

Reproduce the problem and collect the diagnostic information with the dsmInfo tool.

Notes:

For more information about this tool, see the DSMInfoReadMe.txt file available under the DiagnosticTools folder in the product installation media.

The dsmInfo tool produces ".7z" files by default. These files provide better compression than zip files, so uploading to CA Technologies is easier.

Troubleshooting the Errors Reported

Following are some of the ways to resolve the errors reported by the scan:

- In the DSM Explorer, navigate to *Computer Name,* Inventory, SCAP, *Checklist Name,* Status. The Status attribute in the right pane displays the reason why the scan resulted in an error. This attribute can reveal errors such as, benchmark not being applicable to the operating environment, errors in XCCDF file, or OVAL file.
- You can investigate the rule errors by examining the output from the OVAL interpreter, which contains the results of each OVAL definition used by the checklist. To investigate the rule errors, do one of the following:
  - View the *MachineName-fdcc-checklistname*-oval-ovaldi-stdout.txt file under the *agent working directory*\SCAP_Result_Files directory.
  - View the *MachineName-fdcc-checklistname*-oval-ovaldi-stdout.txt file under the *ITCM_installpath*\SCAP_Result_Files*checklistname**version_number* directory on the domain manager if you have configured the collection of OVAL test result files.
- You can set the trace level to detail using the following command and investigate the Asset Management log files for any errors generated by the scan:

  ```
  cftrace -c set -f UAM -l DETAIL
  ```

  Check the log files generated by the scanner for more details.

AM Manager Crashes When MDB Goes Down

The AM manager can crash if the database or database listener is not active.

Duplicate Entries Created by Asset Collector

Symptom:

When I reload the same asset with a different MAC address, the Asset Collector creates duplicate entries for agents. The same behavior applies to the NRI assets.

Solution:

Duplicates entries are created because data is processed based on external_host_key and host_uuid. A new verification has been provided for the serial number to help ensure that duplicate entries are not created.

Do the following:

Run the following command to create the parameter and set the parameter value to 1:

```
ccnfcmda -cmd setparametervalue -ps /itrm/manager/engine -pn
AssetMatchOnUniqueSerialNumber –v  1
```

This change allows a unique search for the serial number, in addition to the existing search for the hostuuid (hostname and MAC).

Error Event Logged

Symptom:

Create and and register OSIM images work successfully, but sdmpcworker logs event "File not found".

Solution:

You may ignore this log entry: The Microsoft loader tries to load fonts even when they are not required. The Microsoft loader then works without this font file, but the boot server logs the attempt to read the file with tftp.

The Inventory Attribute Hostname Shows Incorrect Values for Citrix XenServer Virtual Machines

Symptom:

When I collect the virtualization inventory for a Citrix XenServer platform using RVI Agent plug-in, the inventory attribute hostname under the node Virtualization, Citrix XenServer Virtual Machines shows incorrect values for some of the guest Virtual Machines.

Solution:

This problem happens if reverse IP lookup in the DNS table returns an incorrect hostname value or no value at all. To get the correct inventory data, ensure that IP and Hostnames are updated correctly in the DNS Server.

No New Host UUID for Microsoft Hyper-V Clone

When using VM to VM cloning in Microsoft Hyper-V, it is not possible for the DSM agent to identify that it has been cloned. Therefore, it cannot generate a new host UUID. The reason is that the UUID source is Microsoft Windows Management Instrumentation (WMI) and not BIOS, the default source for CA ITCM. For Microsoft Hyper-V, the UUID is accessible only from the hypervisor WMI and not the guest.

Note: This problem does not occur when cloning from a template.

To ensure that the clones are represented as new virtual machines, follow these steps:

1. Stop CAF.
2. Delete the HOSTUUID key from the registry:
   HKLM\SOFTWARE\ComputerAssociates\HostUUID
3. Change the host name and reboot.

Performance Inventory is not Available on Kubuntu and Debian

Performance Lite Agent is not available for the Debian and Kubuntu platforms. Hence, performance inventory is not collected on these platforms.

Asset Management does not Collect Inventory Data on VM of Citrix XenServer

The Asset Management agent does not collect inventory data for the following subinventory nodes when the agent runs on a virtual machine of Citrix XenServer:

- Inventory, System, Memory
- Inventory, System, I/O Ports
- Inventory, System, System Slots
- Inventory, System, COM Ports
- Inventory, Power Settings, AC Adapter
- Inventory, Power Settings, Battery

vDisk Record Is Not Found in the DSM Explorer

The vDisk record is not found in the DSM explorer, due to the following reasons:

- Connecting to the provisioning server fails to fetch the vDisk information.
- The system times on the vDisk and the provisioning server are not in sync.
- Connecting to the provisioning server via the proxy is not supported.

In all the cases, CAF shuts down.

To fix, perform the following steps:

Follow these steps:

1. Log in to the vDisk with administrator privileges.
2. Verify the personality data provisioning services information.
3. Troubleshoot the connection to the provisioning server.

   On success, CAF starts and the vDisk registration happens.

Processor Throttle Displays a Different Value in CA ITCM

The inventory item Processor Throttle displays a different value in CA ITCM when compared to the output of the powercfg.exe utility. This value is read using a 64-bit version of powerprof.dll.

This value matches the output of the powercfg.exe utility available in the ..\windows\SysWOW64 folder on the Windows platform. The 32-bit version of the same powercfg.exe utility available at the ..\ windows\system32 folder displays a different output.

Citrix XenServer Guest Virtual Machines Are not Shown in Virtualization Inventory

Symptom:

When I collect the virtualization inventory for a Citrix XenServer platform using RVI Agent plug-in, some of the guest Virtual Machines are not shown under Inventory, Virtualization, Citrix XenServer Virtual Machines.

Solution:

A solution to this problem is being worked-out. For more information, see support.ca.com to check if the solution is available.

Tenant ID Not Updated When an Asset Moves to New Tenant

When an asset moves to a new tenant, the tenant ID is not updated. The workaround is to delete the asset first and then reregister it with the new tenant.

Content Download Fails

Symptom:

If I download content when a CIC process runs in the background and another instance is launched by the engine, a detached CIC instance occurs.

Solution:

Do the following:

1. Terminate the wrapper.exe process.
2. Terminate java.exe that is used by CIC.
3. Delete anchor files and loc files if any from C:\Program Files (x86)\CA\SC\CIC\bin.
4. Initiate the content download job again from Engine tasks.

Software Job History Behavior after the Uninstall Job

When the time period specified in the Job history cleanup time setting elapses, the software job history of an agent is cleaned up. The cleanup task runs daily at the time specified in the configuration policy Start Time. Based on the Job history cleanup time setting, the cleanup task checks every day for install jobs that are older than the number of hours specified in the configuration policy. The default period is 4320 hours or 180 days.

The cleanup task attempts to remove all the procedures of a package together. This means that the task searches for the uninstall job history of the package before deleting the install job history. So, even if the install job history is more than 180 days, the task cannot remove the job history until the uninstall job history is available. After you deploy the uninstall procedure, the software job history contains the uninstall job too. When the cleanup task runs at the next scheduled time, it looks for the install jobs that are older than the time specified in the configuration policy and have the corresponding uninstall job in the history. When a match is found, the complete job history of the package (including the install, activate, and uninstall procedures) is deleted from the software job history of the agent. This results in the removal of the history soon after uninstallation.

With the Release 12.8, the comstore setting is changed such that the software job history is deleted when the uninstall job exceeds the time specified in the configuration policy. For example, if you have set the Job history cleanup time parameter to 10 days, the complete job history of the package upto ten days, from the time the uninstall procedure was installed on the computer, is deleted.

Erratic Communication Between CAM Servers

Symptom:

You are experiencing intermittent or ongoing problems with the distribution or collection of data in your asset management environment and have verified that the affected components are active and able to communicate with each other. One example of this problem may be the failure of an otherwise fully functioning asset management agent to return inventory information.

Solution:

If you have determined that the product components are properly running and active, and have ensured that two-way communication is possible, use the camping command to determine if this communication problem is due to the level of network protocol support in the environment.

Always try to use User Datagram Protocol (UDP) as the preferred/default connection protocol. The only exceptions to this rule are when it is determined that one of the following has occurred:

- There is a restriction on the use of UDP within your network
- There is a restriction on the maximum size of UDP packets within your network, which in turn impacts CAM's ability to send messages greater than this size. To determine whether there is a packet size restriction, use the camping -s option, progressively specifying larger and larger UDP packets until such time as you either verify that there is no realistic limit or you determine that there is a limit.

For example:

```
camping -s 64 <remote_machine_ip_address>
camping -s 128 <remote_machine_ip_address>
camping -s 256 <remote_machine_ip_address>
camping -s 512 <remote_machine_ip_address>
camping -s 1024 <remote_machine_ip_address>
camping -s 2048 <remote_machine_ip_address>
camping -s 4096 <remote_machine_ip_address>
camping -s 8192 <remote_machine_ip_address>
```

If you determine that there is a size restriction, specify smaller increments to try to establish the exact maximum size that can be supported.

Next, establish whether the maximum size you have determined actually is reached by CAM. You can do this by examining the CAM trace logs covering a typical transfer (or series of transfers). Look for lines like the following in the CAM trace logs:

```
send_message(10.1.1.14) called

Seq 41, Fd (Sec 0), from uxkul01/CAIFTRANS, to NTAOC04/CAIFTRANS, len 8198, data ><,
created 43434, life 60, notify: yes, src 10.7.1.1, dst 10.1.1.14
```

In this example, len 8198 in the second line refers to the size of the message packet that CAM is trying to send (in this case 8198 bytes). You can then establish the largest packet size that CAM has by looking for this field in all the messages that have been sent by CAM. If this exceeds the size of the maximum UDP packet (as determined in the previous step through the use of the camping -s switch), consider increasing or removing the UDP packet size restriction.

If the asset management inventory collection failed, it is possible that the size of that inventory collection (which, for an initial collection, may be large) may have exceeded the maximum size for UDP packets in your network.

In some cases, an intermediate router has a smaller MTU size restriction than either the originating or destination point. If that intermediate router fails to send the proper ICMP response to the sending host, indicating that condition, it is known as a black hole router. If you experience problems with dropped packets that are similar in size, this may be your problem.

Check the Microsoft knowledge base for more information on the following topics:

- How to Troubleshoot Black Hole Router Issues (159211)
- Default MTU Size for Different Network Topology (140375)

File Transfer Time Out

Symptom:

After you have selected the most appropriate network protocol, you still find that transfers are timing out.

Solution:

Transfers may be timing out for two reasons:

- The remote machine to which the transfer is taking place is too busy to be able to service all its incoming connections within the time allotted for the transfer. In this situation, consider staggering the transfers to the remote machine so that it has time to service them all in the allotted time.
- The connection bandwidth is not high enough to allow for the complete transfer of the file/data within the time allotted for the transfer. You can establish whether this is the case by looking in the CAM trace logs at how many messages CAM is able to send to the remote machine over the period of a second. Using the packet size, as established previously, together with the number of messages processed every second, you can then extrapolate this to establish an approximate maximum file/data size that CAM is capable of transmitting within the allotted time frame. If this is smaller than the size of the file/data that needs to be sent then the bandwidth is not sufficient, given the time allowed. In this situation, you need to either improve the connection bandwidth or approach the support personnel for the application that initiated the transfer to establish whether/how the transfer time can be extended.

Unable to Deploy Asset Management Agent to Windows XP SP2 Computers

Symptom:

When I deploy the asset management agent on Windows XP SP2 computers, it results in delivery failure and shows the status as no primer transport.

Solution:

The local security policy and the firewall on the Windows XP SP2 computers prevent access to the system. Before deploying the asset management agent, you must change the local security policy and specify the firewall settings.

To change the local security policy

1. Open Windows Control Panel, Administrative Tools, Local Security Policy.

   The Local Security Settings window appears.

2. Navigate to Local Policies, Security Options and double-click Network Access: Sharing and security model for local users. Change the security setting to Classic - local users authenticate as themselves and click OK.

   The Local Security Policy is changed.

You must open the UDP ports for CAM communication, by changing the firewall settings.

To change the firewall settings

1. Open Windows Control Panel, Windows Firewall

   The Windows Firewall dialog appears.

2. Select File and Printer Sharing in the Exceptions tab, add one UDP port for 4104 and one UDP/TCP port for 7 and click OK.

Existing Query on a Recreated Group Fails

Symptom:

When I delete a group and recreate it with the same name, the queries dependent on the group does not produce any result.

Solution:

When you create a group, a new UUID is assigned to the group. The Query Designer uses this UUID (not the group name) to query the database. Hence, if you delete a group and recreate it with the same name, the UUIDs of both groups differ, due to which the query fails. You can edit the query and select the group name again to view the correct results.

Note: This symptom is also applicable if you are querying the scalability server for the linked assets. If you delete a server and register it again, the queries dependent on the server does not produce any result. Follow the same solution.

Software Contents Download Job Fails

The Default Software Contents Download job can fail in the following circumstances:

- Missing proxy server details (see page 429)
- Network problem–Unknown error occurred attempting evmUpdate1_0 .RC:-6 (see page 429)
- Unicenter Patch Management Installation (see page 430)

Missing Proxy Server Details

Symptom:

If the DSM installation resides behind a proxy server, the automatic software contents download job may fail due to missing proxy customization.

Solution:

The proxy server details must be updated in the Common Configuration Policy before the software content download job connects to the server.

1. Navigate to Control Panel, Configuration, Configuration Policy in the DSM Explorer

   The existing configuration policies appear.

2. Select the policy you want to modify and navigate to DSM, Manager, Software Contents Download.

   The Software Content Download configurations appear.

3. Add the proxy details such as the server name, port, user and password.

4. Seal the configuration.

   Contents download can connects through the proxy server and complete the software download.

Unknown error occurred attempting evmUpdate1_0 .RC:-6

Symptom:

The Default Software Contents Download job may fail due to network problems. This is indicated by the entry–Unknown error occurred attempting evmUpdate1_0 .RC:-6 in the cic.log file.

Note: The cic.log file is available in the CIC's log directory (Program Files\CA\SC\CIC\logs.)

Solution:

The subsequent downloads will be successful and the download should complete without any error.

Unicenter Patch Management Installation

Symptom

Software content download job fails.

Solution

Before installing CA ITCM, ensure that Unicenter Patch Management is not already installed.

No VM Host or System Controller Displayed in the DSM Explorer

Symptom:

I cannot find the 12amvminvux.inv file under the $CA_ITRM_BASEDIR/Agent/AM/data/transfer_vm directory (UNIX) after amagent has completed. (The relevant Windows directory is ..\Agent\units\00000001\uam\transfer_vm.)

Solution:

You must enable the specific virtual host in the virtual host inventory collect task.

No Related Computers Displayed for the Virtual Host

Symptom:

I do not see related computers for the virtual host although amagent has completed successfully.

Solution:

You must run amagent again to upload the latest inventory to the DSM manager.

Lack of Discovered Software

The generation of a zero-sized Windows signature file under All Users\Application Data\CA\eso_fingerprints can cause the signature scan to fail. Such a zero-sized file is possibly due to a conflict of concurrent processes. The workaround is to get the signature files regenerated by either deleting the zero-sized file or adding a new software definition.

Number of Processors Reported in Dual-Core HP-UX Computers

Symptom:

I find that the number of processors reported under Inventory, System is not the same as reported by the HP-UX System Administration Manager (SAM).

Solution:

If the computer is running with Dual-Core processors, SAM shows two processors for each physical processor. In such a case, the attribute “Number of Cores (per processor)” under Inventory, System must be taken into account.

Number of Processors Reported in SUN E4800/4900/6800/6900 System Controller

Symptom:

I find that the number of processors reported under Inventory, System is not the same as reported by the "showboards -p cpu" command.

Solution:

If the "Description" for the system board is "No board power", the number of processors on this board is not counted.

Installed and Displayed Versions of amVMAgent Are Not the Same

Symptom:

When I compare the installed version of amVMAgent with the version that is displayed in the DSM Explorer user interface, I find that the versions are not the same.

Solution:

If you use amVMAgent (of different version) to do the registration manually by placing the inventory file amvminvux.inv into the $CA_ITRM_BASEDIR/Agent/AM/data/transfer_vm folder, the version number of the virtual agent that is registered using this method will be the same as amVMAgent on that computer.

The Platform Is Displayed as "Unclassified" in the Overview Page

Symptom:

The platform is displayed as "Unclassified" in the Overview page.

Solution:

Enable the default engine job "Default Software Contents Download Job" in the DSM Explorer GUI (Control Panel, Engines, System Engine) and check that this job is enabled and no error has occurred. If the job execution is displayed as an error, make sure that this system is able to connect to the internet to allow the update to be downloaded from the CA website.

No Relationship Between a Virtual Host and Its Guests

Symptom:

I registered a virtual host and later on I registered a new guest system (such as HP-UX vPar). After registration of the guest system, I do not see the relationship between this guest system and its related virtual host. What went wrong?

Solution:

By default, the relationship between a virtual host and its guest system is created only when the virtual host is registered. If any guest system is registered afterwards, no relationship will be established.

You can manually start the asset management agent on the agent machine with the configured virtual host inventory collect task, or wait until the next scheduled asset management agent process.

An Inventory File Has an Error Tag

Symptom:

One of my inventory files has an error tag.

Solution:

If there is a problem processing an inventory file, the file will be renamed, appending ".error" to the end of the filename. For details of what has caused the error, you should consult the application event log.

Load Distribution

Symptom:

I am not sure how to distribute the load.

Solution:

If you have multiple Scalability Servers with Asset Collectors, you can distribute the load by evenly distributing your inventory files between the available collectors.

Finding Errors

Symptom:

I am not sure where to look for errors.

Solution:

If an error is detected in an inventory file, the file has the extension .error appended to its filename. You can look into the application events console to determine what the error detected is.

Some examples of errors are:

- Badly formatted xml
- String values that are too long
- Integer values that contain strings
- Integer values that are outside set limits

Limits to How Many Inventory Files the Asset Collector Can Process

Symptom:

I do not know the how many inventory files the Asset Collector can process.

Solution:

The limit to the number of files that can be processed is only limited by the available disk space. The hostuuid cache grows with each new asset provided. Storage space is the only limiting factor.

Trust Level and Origin Fields Do Not Appear in the DSM Explorer

Symptom:

I cannot see the Trust and Origin fields in the DSM Explorer.

Solution:

There are three possible reasons for this:

- Your DSM Explorer is connecting to a manager that does not have the Asset Collector changes installed. In this scenario the DSM Explorer does not display these columns
- Your DSM Explorer installation has not been updated to include the Asset Collector changes
- Both of the above

Trust and Origin Fields in the DSM Explorer Do Not Appear When I Connect to Another Domain

Symptom:

I can only see the Trust and Origin fields in the DSM Explorer when I connect to one domain but not another.

Solution:

This is probably caused by only one of the domains having the Asset Collector changes applied. Remember that the MDB changes and components changes (installation) have to be applied separately.

Backup

Symptom:

I want to backup for a disaster recovery.

Solution:

Add the Host UUID \ Mac Address cache file to your current disaster recovery plan. The name of this file is specified in your active configuration policy. The default name is machostcache.xml.

Asset Collection from Floppy Drives

Symptom:

I want to specify a floppy drive as a collection point for the Asset Collector, but file notification change mechanism works when I insert a floppy disk.

Solution:

To force the Asset Collector to recollect from a floppy disk, stop and start the Asset Collector CAF plugin with the following command:

```
caf stop assetcollector
caf start assetcollector
```

Collect from Network Drives

Symptom:

I cannot collect from network drives.

Solution:

By default the asset collector runs as the Local System account user. You will not have access to the network drives of your computer. You have to configure the Asset collector CAF plugin to run as a different user.

To enable the Asset Collector to collect from network drives

1. Configure the AssetCollector to allow running as a different user.
   ```
   caf setprop assetcollector setcreds 1
   ```
2. Specify the credentials to use.
   ```
   caf setcreds assetcollector user <username> password <password>
   ```
3. Stop and start asset collector to start using new user.
   ```
   caf stop assetcollector
   caf start assetcollector
   ```

To reverse this configuration, use the following command:

```
caf setcreds assetcollector user “” password “”
```

Unprocessed Files

Symptom:

The files in the collection folder are not being processed. They are not being moved to the output folder as expected.

Solution:

Check that the asset collector is running. This can be done by issuing a caf status, or caf status assetcollector command.

If the Asset Collector is not running, start it with caf start assetcollector.

Check that the collection folder the asset collector is monitoring is the folder where you are placing your file. From the DSM explorer you can request a configuration report from a computer to see the configuration policy in use.

It could be that the services that the Asset Collector communicates with to register and upload inventory is not running. The asset collector relies on the Common Server plugin for registration, and the asset management server for inventory upload. You should see messages regarding failure to register or upload in your event log, if this is the case. Once the services are restarted, the Asset Collector should upload the inventory on it's next attempt, this could be up to 5 minutes after the dependent services are started.

You may have configured the Asset Collector to report inventory to an invalid scalability server. Check this in the active configuration.

Many Errors in the Application Event Log

Symptom:

I am seeing lots of Error events generated in the Application Event Log when the Asset Collector cannot register an asset or upload inventory data.

Solution:

When the Asset Collector cannot register or upload an inventory file, the file is queued to be retried. The Asset Collector will attempt to retry the registration after 5 seconds. If this registration cannot be performed, the time delay is increased to 10 seconds, and then 20 seconds, this repeated to a maximum delay of 5 minutes. An event will be raised on each of these failed attempts. So when the caf plug-ins that the asset collector relies on return to service it could be 5 minutes before any inventory files are processed by the Asset Collector.

SCCM Converter Engine Task Fails

Symptom:

An SCCM Converter engine task fails with a *Could not connect to SCCM site* message even though doing a Test connection in the Create New Task dialog reports a success.

Solution:

The test performed when you click on the Test Connection button in the Create New Task dialog is performed on the machine running the GUI and in the context of the user running the GUI. When the actual SCCM Converter task is run it is done on the engine machine and in the context of the user account used by the engine (the local system account by default). This means that a successful test does not guarantee a successful connection when the task is actually run. You can eliminate the potential differences by making sure you include the domain name in the Username field of the configuration (in this format: domainname\username).

If this is not done it will default to the domain of the current user which for a default engine amounts to the local users of the engine machine. If possible the test should also be performed on a machine which has the exact same access to the SCCM installation as the engine machine meant to perform the conversion.

Missing Inventory Information

Symptom:

Assets in the computer groups are showing but the detailed inventory information is missing.

Solution:

If the scalability server the Asset Collector is configured to send inventory to does not have an asset management server installed, then the inventory information will not be available for the registered assets. You will see the Asset in your computer groups, but the detailed inventory information will be missing. Only use scalability servers that have the asset management server plug-in available and running.

How Do I Know Agent Bridge Is Running?

Symptom:

I need to verify that Agent Bridge is indeed running. How do I do this?

Solution:

To verify that the AM Agent Bridge is running, open the Windows Task Manager and check the Processes tab for two processes: amLrss.exe and amss.exe. If these are listed, the AM Agent Bridge is enabled and running.

As for the SD Agent Bridge, which is fully integrated with the software delivery scalability server, if the software delivery scalability server is running, SD Agent Bridge is running.

Cannot Register My Legacy Agent

Symptom:

I cannot register my legacy agent with CA ITCM after enabling the Agent Bridge.

Solution:

After enabling your Agent Bridge configuration policies, you have to wait for several minutes before you can register your legacy agent.

USD Agent Overwritten by UAM Agent on My PDA

Symptom:

I have a PDA device with a USD 4.0 agent installed and pointed to Agent Bridge. It was registered successfully with Unicenter DSM. Later on, I installed a UAM 4.0 agent to the device as well, pointed it to the same Agent Bridge, and registered it. However, my USD agent then got overwritten by my UAM agent with a different name. What happened to my USD agent?

Solution:

Existing UAM and USD 4.0 PDA device agents have the ability to be given a specific agent name during the installation of the UAM and USD PDA device agent software. This name is used for the registration of the PDA device. If the UAM agent is given a name that differs from the USD name, then this causes the UAM and USD PDA agents to overwrite each other with their own names during the registration process.

To solve the problem, both UAM and USD agents must be given the same name during the installation of the agent software on the device, so that both UAM and USD agents share the same agent name in the Unicenter DSM system.

Why Do I Get Duplicate Agent Names?

Symptom:

I registered a Unicenter Asset Management 3.2 agent first, then immediately registered a Unicenter Software Delivery 4.0 agent. After a few minutes I saw two agents displayed in the DSM Explorer with the same name, one for Unicenter Asset Management and one for Unicenter Software Delivery.

Solution:

With the UUID generator enabled, the Unicenter Asset Management agent has generated a host UUID via Agent Bridge, but the Unicenter Software Delivery agent has received a host UUID from the agent computer itself. If both agents are registered within a very short time frame, then there is the possibility that duplicate agent names would be generated, one for Unicenter Asset Management and one for Unicenter Software Delivery.

In this situation, you have to manually remove the Unicenter Asset Management agent by deleting it from DSM Explorer. To avoid this problem from happening again, follow these registration rules:

To register legacy agents, some of which may not report their own UUIDs

- If both the Unicenter Asset Management and Unicenter Software Delivery agents do not report host UUIDs, then you can register the Unicenter Asset Management and Unicenter Software Delivery agents in any order. However, you need to give time between the registrations of the two agents, typically, at least five (5) minutes apart.
- If the Unicenter Asset Management agent does not report a host UUID but the Unicenter Software Delivery agent does, then you should always register the Unicenter Software Delivery agent first. Then after five (5) minutes register the Unicenter Asset Management agent.
- If the Unicenter Software Delivery agent does not report a host UUID but the Unicenter Asset Management agent does, then you should always register the Unicenter Asset Management agent first. Then after five (5) minutes register the Unicenter Software Delivery agent.

In general, the majority of Unicenter Asset Management 4.0 agents and Unicenter Software Delivery 4.0 agents report host UUIDs when registering with the server.

How Can I Display the Agent Bridge Log Files?

Symptom:

I cannot see the log files for the Agent Bridge components in the ..DSM\logs directory. How can I display these log files?

Solution:

After you run the command line script, AgentBridge.dms, you should restart the CAF service to ensure that all trace file entries are registered properly to the system. If after starting CAF, you still cannot see the Agent Bridge log files, contact CA Support Online at http://support.ca.com for further assistance.

Unable to Locate Log File

Symptom:

I am trying to locate the amss.log file without success. Where can I find this log file?

Solution:

The name of the log file for the AM Agent Bridge mobile sector server module, amss.exe, is TRC_AMSS.LOG, not amss.log. It can be found in the ..DSM\logs directory.

Note that there are two other new log files for the AM Agent Bridge:

- TRC_AMBRIDGE.LOG

  Log file for the AM Agent Bridge translation module, amBridge.exe.

- TRC_AMLRSS.LOG

  Log file for the AM Agent Bridge sector server module, amLrss.exe.

Only One Hardware Module Is Executed

Symptom:

I only see that one of my hardware inventory modules has been executed. All of the others reflect a "waiting" state. What went wrong with Agent Bridge?

Solution:

On Microsoft Windows, only the very first hardware inventory module is executed by an agent and the rest are ignored. This behavior is by design in Unicenter Asset Management 4.0 agents. To work around this limitation, place all of the hardware inventory collect tasks into one module and assign it to the computer. In this case, all inventory tasks will be executed by an agent.

Missing Solaris Non-Global Zones, Resource Pools, and Processor Sets Inventory

Symptom:

I cannot see any inventory related to Solaris non-global zones.

Solution:

You must install the asset management agent on the global zone. There is no need to configure a virtual host inventory collect task, because information about the non-global zones, resource pools, and processor sets is collected by default by the agent.

The non-global zones inventory is reported under Virtualization, Zones in the DSM Explorer. The resource pools inventory is reported under Virtualization, Resource Pools, Resource Pools, and the processor sets information is reported under Virtualization, Resource Pools, Processor Sets.

Script Compilation Errors in an Intellisig

Symptom:

A particular Intellisig always shows script compilation errors.

Solution:

Verify that the version of dmscript.exe on the target computer is not below R12.5.1010. An updated version is required because Intellisig scripts use new features of dmscript that were not available before R12.5.1010.

Intellisig fails to Execute even when Collect Task Runs Successfully

Symptom:

A particular Intellisig does not execute even though the Collect Task runs successfully.

Solution:

Follow these steps:

1. Navigate to Software, Definitions, Intellisigs.
2. Verify that the Intellisig you want to execute has Enable for Discovery set to Yes. If not, right-click and select Enable Scan.
3. Right-click on the Intellisig, select Properties and view the Versions tab. Verify that the correct Intellisig version has Active set to Yes. If no versions are active, the Intellisig does not run.

Check Execution of Intellisgs

Symptom:

I want to check if my Intellisigs are executing correctly.

Solution:

If there are any errors generated during the running of software inventory, you can view them in the System Status portal on the DSM Explorer main page.

Follow these steps:

1. Navigate to Computer, Configuration, Collect Tasks.

   Software Inventory Configuration is displayed as one of the collection tasks, if software inventory has been configured on that asset.
2. Right-click the task and select Status to see detailed error information about the software inventory execution.

You can view the state of software inventory execution on individual computers.

Execution of Intellisigs on Target Computers Times Out

Symptom:

The execution of some Intellisigs on some target computers always times out.

Solution:

Intellisigs are designed to run as background processes, to help ensure that that there is minimum impact on the performance of the target computer when the scripts are running. Because of this reason, it is possible that some long running scripts on highly loaded computers take a longer time to execute. In such cases, Intellisigs exceed the available execution timeout.

You can control the available execution timeout using configuration policies. You can find the configuration policy value that controls the default Intellisig execution timeout value at the following location:

Control Panel, Configuration, Configuration Policy, *policy name*, DSM, Agent, Asset Management, Intellisig Default Execution Timeout.

With CA ITCM Configuration policies, you can define the timeout value to a different period in different policies. You can apply different policies to different computers, or groups of computers. In this way, you can modify the configuration value only on the policies that are active on the computers where the Intellisig timeout is detected.

Review of Intellisig Scripts that Run on Agents

Symptom:

I want to review the Intellisig scripts that run on agents.

Solution:

Follow these steps:

1. Navigate to DSM explorer, Software, Definitions, Intellisigs.
2. Right-click on the Intellisig you want to review.
3. Click the Version tab and then select the Intellisig version you want to review.

   Note: An Intellisig can have a number of versions, but only one version is active at a time. It is the active version that is executed on your agent computers.
4. Select the version you want to review and click the Properties button.

   A dialog displays detailed information of the Intellisig.
5. Click the Script tab to view the name of the Intellisig script.
6. Click the Preview button to view the contents of the file.

Confirm Export of Intellisig Definitions by the Domain Manager

Symptom:

I want to confirm that the latest Intellisig definitions have been exported by the Manager.

Solution:

Intellisig definitions are exported by the CA ITCM engine to XML and DAT files in the folder %ALLUSERSPROFILE%\Application Data\CA\eso_fingerprints. Intellisig definitions for Windows are available in Xnnnnnnn.xml and Xnnnnnnn.dat files, and for UNIX in Mnnnnnnn.xml and Mnnnnnnn.dat, where nnnnnnn is a number which increments whenever there is a change to the definitions.

The XML files contains a list of the Intellisig definitions, the name of the main script file, and any triggers and additional files.

The DAT files contain the main script and any additional files. It is in a compressed format and can be uncompressed for example, by using 7-Zip. You can compare the contents of the XML and DAT files with the Intellisig definitions that you see in the DSM Explorer.

Modification of Engine-Managed Policy Parameter does not Change its Behavior

Symptom:

I have changed an engine-managed policy parameter but the behavior has not changed.

Solution:

Verify that the machine on which the engine is running has received the updated configuration policy. You can do this from the DSM Explorer by navigating to All Computers, Computer, Configuration Policy, Request Configuration Report, View Configuration report. Verify that the policy parameter has the required value.

After the policy parameter has changed on the computer, the CA ITCM engine on that computer gets notified of the change and re-reads the parameter. You can check this in the detailed engine log, by looking for CONFIGURATION_CHANGED. To help ensure that the new configuration is retrieved, you can restart the engine process.

Note: Verify that the Convert replicated Intellisgs on unlinking parameter is effective on the computer where the operation is initiated. This computer is usually the enterprise manager.

Replicated Intellisig Definition does not Replicate after Manual Deletion

Symptom:

I have manually deleted a replicated Intellisig definition on my Domain Manager. The Intellisig definition did not replicate again from the enterprise manager.

Solution:

Replication works on a delta mechanism depending on what has been created, updated, or deleted from an MDB. If you manually delete a replicated object (for example, a replicated Intellisig definition at a domain manager), it is not re-replicated unless there is a change to it at the enterprise manager.

Appendix A: Management Information Format (MIF) Files Reference

The following sections discuss the various conventions, definitions, and statements involved in using .MIF files.

This section contains the following topics:

Lexical Conventions

The MIF uses either the International Standards Organization document ISO 8859-1 (Latin Alphabet No. 1) or the Unicode 1.1 Specification for its character sets. If a Unicode MIF is provided, the first octet of the .MIF file must be 0xFE (hexadecimal) and the second must be 0xFF. Otherwise, the Service Layer treats the file as an ISO8859-1 MIF.

There are four classes of tokens: keywords, integer constants, strings (literals), and separators. Two keywords, “start” and “end,” are scope keywords, which are only useful when followed by another keyword. Blanks, tabs, new lines, carriage returns, and comments (collectively known as "white space") as described following are ignored except when they serve as separate tokens. White space is required to separate adjacent keywords and constants.

MIF is case-insensitive in all situations, except for literal strings where characters surrounded by double quotes are case-sensitive.

Literal strings separated by white space are concatenated and stored as one literal string.

Comments

Comments can be placed throughout the file, and are ignored. The start of a comment is denoted by two consecutive forward slashes (//). The comment continues through the end of the line.

Keywords

A .MIF file structure uses the following keywords:

Component, Group, Attribute, Table, Path, enum, Name, Description, ID, Type, Class, Key, Value, Access, Storage, Language, Start, End, Unsupported, Counter, Counter64, Gauge, Octetstring, Displaystring, String, Integer, Int, Date, Integer64, Int64, Win16, Win32, DOS, Macros, Os2, UNIX, Read-only, Readwrite, Write-only, Direct-interface, Common, Specific

Data Types

Data types describe the storage requirements and some semantics. MIF supports the data types listed in the following table:

Data Type	Description
integer or int	32-bit signed integer; no semantics known.
integer64 or int64	signed integer; no semantics known.
gauge 32-bit	unsigned integer that can decrease or increase.
counter 32-bit	unsigned integer that never decreases.
counter64	64-bit unsigned integer that never decreases.
string (n) or displaystring(n)	Displayable string of n octets.
octetstring(n)	String of n octets, not necessarily displayable.
date	28-octet displayable string.

Counter

A counter increases to its maximum value (232-1 or 264-1) and rolls over to zero at its maximum value. An automobile odometer is an example of a counter.

Gauge

A gauge can increase or decrease, but when it reaches its maximum value (which is 232-1) it continues to report the maximum value, until the value decreases below the maximum. An automobile speedometer is an example of a gauge.

String

For the string types, the value n represents the maximum number of octets in the string. The actual number of octets in use can be shorter than this maximum value. Strings are stored with their length in the first four octets (which are not counted in n); they do not need to be zero-terminated as in the C/C++ programming languages. String lengths represent the number of octets in the string, not the number of characters.

Date

A Date in the MIF file should have the following format:

```
yyyymmddHHMMSS.uuuuuu+ooo
```

where,

yyyy is the year, mm is the month number, dd is the day of the month, HHMMSS are the hours, minutes, and seconds, respectively, uuuuuu is the number of microseconds, and +ooo is the offset from UTC in minutes. If east of UTC, the number is preceded by a plus (+) sign; if west, by a minus (-) sign. While this is only 25 octets the date is stored as a 28-octet field for memory alignment reasons, and the last three octets are zero ("\0").

For example, Friday 6 February 2004 at 1:30:15 PM EDT would be represented as: 20040206133015.000000-300

Values must be zero-padded, if necessary, like 06 in the example previously shown. If a value is not supplied for a field, each character in the field must be replaced with asterisk (*) characters.

Constants

Integer values can be specified as in the C/C++ programming languages. The following table provides information on the syntax and base:

Syntax	Base
Nnn	decimal
Onnn	octal

Oxnnn or OXnnn	hexadecimal

n is a digit in the proper base.

Note: MIF does not support floating-point values.

Literals

Literals (strings) are character sequences surrounded by double quotes. Adjacent double-quote characters (besides white space) indicate multi-part literals, which are treated as one string.

For example:

"This is an example" " of a multi-part" " literal string."

The literal Escape character is the backslash (\). It is used as in the C/C++ programming languages, to enter the characters listed in the following table:

Sequence	Character
\a	Alert (ring terminal bell)
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\\	Backslash
\"	Double quote
\xhh	Bit pattern, hexadecimal
\ooo	Bit pattern, octal

If the character following a backslash is not one of the sequences previously shown, the backslash is ignored.

For the octal bit pattern, ooo can be one, two, or three octal digits (from \0 to \377) when the MIF is specified in ISO8859-1 format, and from one to six octal digits (from \0 to \177777) when the MIF is in Unicode format.

For the hexadecimal bit pattern, hh can be one or two hex digits (from \x0 to \xff) when the MIF is specified in ISO8859-1 format, and from one to four hexdigits (from \x0 to \xffff) when the MIF is in Unicode format.

Block Scope

The keywords "start" and "end" delimit the scope of a definition block. An associated keyword must follow both start and end. The keywords and their scope are listed in the following table:

Block	Within	Description
Component	.MIF file	Defines a component. All other blocks exist within this scope. There can be only one component definition per .MIF file.
Path	Component	Associates a symbolic string with operating system specific path names. Zero or more path definitions may exist in the MIF, usually at the top of the file before any groups.
Group	Component	Defines a collection of attributes, sometimes used as a template row for a table. At least one group is required per .MIF file (the ComponentID group, defined following).
Attribute	Group	Defines a unit of managed data. All attributes "exist" in the scope of a group definition. A group must contain at least one attribute.
Table	Component	Defines one or more instances of a group using a previously defined group. Optional.
Enum	Component or Attribute	Defines a list of integer-to-string mappings. Named enumerations can be defined at the component level, while unnamed enumerations can be defined in the scope of an attribute definition. Optional, but while many enum definitions can exist at the component level, only one can be defined per attribute.

Following is an example of a .MIF file structure. For readability, only one of each block is given. Each level is indented for readability:

```
start component
    start path
    end path
    start enum
    end enum
    start group
            start attribute
                    start enum
                    end enum
            end attribute
    end group
    start table
    end table
end component
```

Language Statement

The language statement describes the native (human) language of the .MIF file. This optional statement, if provided, appears before the start component statement. The syntax is as follows, with language string being a text string that identifies the language, dialect as territory, and character encoding:

```
language = "language string"
```

The format of the language string is as follows, where language-code is one of the two-letter codes defined in ISO 639, territory-code is one of the two letter codes defined in ISO 3166, and encoding is either ISO8859-1 or Unicode.

```
language-code|territory-code|encoding
```

For example, the following language string indicates French Canadian, with ISO8859-1 (8-bit) encoding:

```
"fr|CA|iso8859-1"
```

If any fields are not supplied, they are omitted, but the two vertical bars must appear in the string. The default language string is

```
"en|US|iso8859-1".
```

The encoding field is ignored in the .MIF file, because the first two bytes of the file determine the encoding. However, the encoding field is used when communicating through .MIF files.

Since users can edit the .MIF file to translate the literal strings to a different language, they can change the language string also. The language statement can appear only once per .MIF file.

Samples of the codes defined in the two ISO standards are found at the end of this chapter.

Note: .MIF files that have been localized, that is, translated, should translate only literal strings. For example, names, descriptions and enumeration literals, and any comments in the MIF. Do not localize class strings or language names. Keywords must not be localized.

Common Statements

The following three statements can be used in the scope of most definitions, as noted. Definition-specific statements are described when the definition is described.

Name Statement

The name statement assigns a relatively short string to the definition. This name is displayed to users and must be less than 256 characters. A sample of the syntax is:

```
name = "name string"
```

In the name statement, the .MIF file provider defines the name string. However, users can edit the .MIF file and change the name. The name statement can appear only once per definition. Names are not required to be unique except for enumeration and path names, which must be unique among other enum (and path) names in a component.

Description Statement

The optional description statement gives more information about the element being defined. It provides a description of the element and displays it to the users. A sample of the syntax is:

```
description = "description string"
```

In the description statement, the .MIF file provider defines the description string. However, users can edit the .MIF file and change the description. The description statement is used in the component, group, and attribute definitions. The description statement can appear only once per definition.

ID Statement

The ID statement assigns a unique numeric identifier for the definition. Each type of definition that is required to have an ID must have a unique ID in its scope. IDs are used for naming items at the API level, and for mapping to network management protocols. The syntax is:

```
id = n
```

In the ID statement, the .MIF file provider defines n. The value of n must be a non-zero 32-bit unsigned integer and must be unique in the scope of the containing definition. For example, all attributes in a group must have different IDs, but it is not necessary for attribute IDs to be unique across groups. Since components and management applications use these IDs for communication, users cannot change them.

The id statement is required in the attribute and table definitions. It is optional in the group definition. It is not used in the component, path, and enum definitions. The Service Layer assigns IDs to the components at installation time. The id statement can appear only once per definition.

Component Definition

The component definition has the following syntax:

```
start component

        name = "component name"

        [description = "description string"]

        (component definition goes here)

end component
```

Only one component definition can appear in a .MIF file.

Path Definition

Path definitions relocate the files used for active management of the component.

A path definition has the following statements in the order given below:

- start path
- name statement defining a symbolic name

  Note: The symbolic name can be assigned a value later in the attribute definition indicating whether it should be retrieved or set by invoking the callable function.
- number of lines equating operating system identifiers to the path of the callable program
- end path

The operating system identifiers are DOS, MacOS, OS2, UNIX, win16, and win32. This is case-insensitive. When the code provides the component instrumentation that connects to the Service Layer (SL), use the keyword direct-interface. Contrarily, if the SL starts the code at request time, specify the path name of the callable program.

Example: Path Definition

```
start path

        name = "Performance Info Instrumentation Code"

        win16 = "C:\\someplace\\wincode.dll"

        os2 = direct-interface

        dos = "C:\\someplace\\doscode.com"

        unix = "/someplace/unixcode"

end path
```

Many path definitions may appear in the component definition; one for each callable function. The path name must be unique among all other path names in a component definition.

See the sample MIF (at the end of this chapter) for examples on the use of the symbols defined in the path definition.

Enum Definition

Enumerated lists let strings be associated with signed 32-bit integers. They are defined in the component scope or in the scope of individual attributes. Component instrumentation uses these enumerations to pass integers through the DMI, so that management applications can display the corresponding text string in the user's native language.

The syntax of enumerated lists is:

```
start enum

        name = "enum name"

        vvv = "string literal for vvv"

        [xxx = "string literal for xxx"]

end enum
```

enum name

Specifies a unique enumeration list name in this component.

Integer values vvv and xxx

Specifies the lowest and highest numbers. This can be listed in any order. It is not necessary to have every number, which is represented between the lowest and highest, listed. However, each value must be unique in this enumeration definition.

Many enum definitions can appear in the component definition; one for each enumeration list. Enumerations do not have ID or description statements.

Group Definition

A group is a collection of one or more attributes. Groups let component providers arrange attributes into logical sets. Groups can also be used to represent arrays (tables) of attributes. The use of groups lets logical subsets in a component be standardized across vendors.

The syntax of a group definition is:

```
start group

        name = "group name"

        class = "class string"

        [id = nnn]

        [description = "description string"]

        [key = nnn[,mm]...]

        (attribute definitions go here)

end group
```

The ID statement, if provided, must have a value unique among other groups in the component. Specifying a group ID without a key means that this group definition defines a group. If both id and key are provided, the group definition represents a table managed by the component instrumentation code. Any subsequent table definitions (defined following) in the component definition cannot use this group definition as a template.

If the key statement is provided and the id statement is not provided, the group definition represents a template row in a to-be-defined table, and the value statements (defined following) refer to default values in the row. A table definition may follow to populate the table based on the template. For more information, see the section on table definition. The following table describes the possibilities:

Key?	ID?	Result
No	No	Error
No	Yes	Scalar group. Not table. Id is group's ID.
Yes	No	Template. Table definitions may follow.
Yes	Yes	Table. Id is table's ID. Can be used as template later.

Many groups can be defined in the component.

Class Statement

The mandatory class statement identifies the source of the group and the group version. All groups using the same class string must share the same attribute definitions in the group, including attribute type, access, storage (defined following) and IDs. The attribute name, description, and value may be different, however. Class statement assists management applications in determining the semantics of the group's attributes.

The class statement syntax is:

```
class = "class string"
```

By convention, class string is encoded as

```
"defining body|specific name|version"
```

Defining Body

Specifies the name of the organization defining the group. For example,"DMTF", "IEEE", "Acme Computer"

Specific Name

Indicates the contents of the group. For example, "Server Stats", "Toaster Controls".

Version

Indicates the version of the group definition. For example, "1", "A", "FIRST ONE EVER".

Essentially, the class string is an opaque string, and any convention can be used. However, since applications and Service Layers may rely on this convention for obtaining information through the List Component command, component providers are encouraged to use this convention.

It is an error to specify the same class string for two groups, if the group definitions are different. Management applications can count on identical group definitions for identical class strings.

Note: "DMTF|Sample|1.0" is not the same as "DMTF | Sample | 1.0" as one has spaces around the vertical bars and the other does not.

Implementations that provide a subset of the attributes defined by a class can just omit the definitions for the unsupported attributes. A better method is to use the unsupported keyword in the attribute definition (defined following). Management applications must be sufficiently robust to deal with subsets of a class.

Only one class statement is allowed per group.

Key Statement

Key statements define attribute IDs that are used as the index into the table. When the attributes in a group define a row in a table, the group definition must contain a key statement. Attributes acting as keys can be of any data type. Keys always identify no more than one instance of a group (row of a table).

The key statement syntax is:

```
key = n[,m]
```

n

Specifies the attribute ID that acts as the key for this table. If multiple attributes are used to index a table, they should be specified as comma-separated integers. When management applications send requests or component instrumentations send results, key values must be sent in the order that they are listed in the key statement.

Only one key statement is allowed per group.

Attribute Definition

An attribute is an item of data related to a component. Attributes are defined in the scope of a group. The syntax of the attribute definition is:

```
start attribute

        name = "attribute name"

        id = nnn

        [description = "description string"]

        type = datatype

        [access = method]

        [storage = storagetype]

        [value = [v | * "name" | "enum string"]]

end attribute
```

The required id statement must have a value that is unique among all other attributes in the group. Groups must have at least one attribute definition. Many attribute definitions can appear in the group definition.

Type Statement

The mandatory type statement in the attribute definition describes the storage and semantic characteristics of the attribute being defined. The syntax is:

```
type = datatype
```

Datatype is usually one of the data types previously defined.

A data type may be an enumeration; stored and treated as a signed 32-bit integer. Enumerations that have been previously defined (at the component level) can be referenced by name, as if they were a type, for example: type = "Color." Enumerations may also be constructed "in line":

```
type = start enum

        (enum definition

        end enum
```

In this case, the enumeration does not need a name, since it cannot be referred to outside the scope of this attribute definition. Any name given is ignored.

Only one type statement can appear in the attribute definition.

Access Statement

The optional access statement determines whether the attribute value can be read or written.

The syntax is:

```
access = method
```

Method

Specifies whether the access is read-only, read-write, or write-only. If the access statement is not specified, the default access is read-only. Attributes marked as keys cannot be write-only. Only one access statement can appear in the attribute definition.

Storage Statement

The optional storage statement provides a hint to management applications to assist in optimizing storage requirements.

The syntax is:

```
storage = where
```

Where

Specifies the storage area. It can be common or specific. Common signifies that the value of this attribute is typically limited to a small set of possibilities. An example of common can be the clock speed of a CPU. Specific signifies that the value of this attribute is probably not a good candidate for optimization, because there can be many different values. An example of a specific attribute would be a component's serial number.

If the storage statement is not specified, the default storage is specific. Only one storage statement can appear in the attribute definition.

Value Statement

The value statement provides a value or value access mechanism.

The syntax is:

```
value = v

value = "enumeration value"

value = * "Name"

value = unsupported
```

v

Specifies that the attribute is read-only, which never change. For example, the manufacturer of a component, or for read-write attributes, which the Service Layer can handle, as opposed to the component instrumentation. Specifying v for write-only attributes is prohibited. The value v must be specified in the correct data type for the attribute; for example, dates and literal strings must be specified in double quotes.

"enumeration value"

Specifies that the value is an enumeration string which the Service Layer maps to an integer. This value is enclosed within double quotes. The mapping must have been previously defined in an enum definition in this component or attribute definition, and the attribute's type must be an enumeration. Specifying an enumeration value for write-only attributes is prohibited.

Note: Specifying an integer for an enumeration is acceptable.

*** "Name"**

Indicates the symbolic name of the component instrumentation code to invoke to read or write the attribute at runtime. The symbolic name must have been previously defined in a path definition in this component definition. The value * "Name" has a name with an asterisk (*) before it and surrounded by double quotes.

The value unsupported (reserved keyword) can be put in to tell the Service Layer that this attribute is not supported by this component.

The value statement is required except when table templates are defined, when it is optional. If a value is provided in a template, it becomes the default value when populating the table. If it is not provided, there is no default value.

Example: Example of a group with two attributes

```
Start Group

        Name = "Software Template"

        Class = "DMTF|Software Example|1"
```

```
        Key = 1 // key on Product Name

        Start Attribute

                ID = 1

                Name = "Product Name"

                Description = "The name of the product"

                Storage = Common

                Type = String(64)

        End Attribute

        Start Attribute

                ID = 2

                Name = "Product Version"

                Description = "The product's version number"

                Type = String(32)

                Value = ""

        End Attribute

End Group
```

In this example, the group is acting as a template, as there is no group ID but a key is specified. The default value for the version is an empty string. There is no default for the product name.

Populating Tables

An array of group instances construct a table. The instances are rows of the table. Often, defining the group with a key is sufficient for defining the table, as the values of the attributes in each row are provided by the component. However, sometimes it is useful to provide the table's values in the .MIF file itself to define values in an attribute definition.

The table population mechanism separates the definition of the group from the data in the group. It uses a previously defined group as a template to store values into the MIF database.

The syntax to populate tables is as follows:

```
start table

        name = "table name"

        id = nnn

        class = "class string"

        { v1[,v2 ...] }

        [ { vn[,vm ...] } ]

end table
```

A name statement must describe this table. The required id statement specifies an integer value unique across all other groups and tables in this component. The required class statement identifies the previously defined group that is being used as a template.

In a table row, the values are provided as previously shown, separated by commas and surrounded by the curly braces "{" and "}". The list of values is provided left-to-right in attribute-ID order; the value of the attribute with the lowest ID appearing first. If a value in the list is omitted, the corresponding attribute value, if defined in the template, is used as the "default" value. Omitting an attribute's value when no default value is provided in the template is prohibited. Rows with too few commas are treated as rows with the requisite number of trailing commas, so the values specified in the template are used for the remaining attributes in the row.

Example: Populating a table using the group previously defined

```
start table

        Name = "Software Table"

        Class = "DMTF|Software Example|1"

        Id = 42
```

```
        {"Circus", "4.0a"}

        {"Disk Blaster", "2.0c"}

        {"Oleo", "3.0"}

        {"Presenter", "1.2"}
end table
```

In this example, the resulting table has four rows. The value statements in the group definition are used as default values during row population and not as a row themselves.

To populate rows without providing unique values for the combination of attributes that comprise the key is incorrect. Service Layers must reject a MIF that does not provide unique keys during row population.

A table definition must come after the group definition to which it refers. The group must have been specified with a key statement, and without an ID statement. More than one table can be created from a single template, but each table must have a different ID.

ComponentID Group

Every .MIF file must contain a standard group with ID 1. This group offers baselevel identification of the component and represents the minimum amount of information, which a component vendor should provide. An attribute, which is not supported or has no meaning for a given component should give the keyword unsupported as its value.

The ComponentID. "DMTF|ComponentID|1.0" is the class string. The attributes in the group are all read-only. Their definitions are mentioned in the following table:

Integer Name	ID	Type	Storage	Description
"Manufacturer"	1	String(64)	Common	Organization producing this component.
"Product"	2	String(64)	Common	Name of this component or product.
"Version"	3	String(64)	Specific	Version string for this component.
"Serial Number"	4	String(64)	Specific	Serial number for this component.
"Installation"	5	Date	Specific	Time and date of the last install of the component on this system.
"Verify"	6	Integer	Common	Verification level for this component.

Asking for the value of the "Verify" attribute causes the component instrumentation to perform checks to verify that the component is still in the system and working properly. It should return one of the values mentioned in the following table:

Value	Meaning
0	Error occurred, check status code
1	Component does not exist
2	Verify not supported
4	Component exists, functionality untested
5	Component exists, functionality unknown
6	Component exists, functionality no good
7	Component exists, functionality good

.MIF File Example

The following is a .MIF file example:

```
Start Component

// A double slash indicates that the line contains comments,

// and should be ignored...

// First we name and describe the component...

  Name = "AM Windows 95 PC information"

  Description = "This MIF file contains basic hardware inventory for a AM Windows 95
Computer"

  // Then we list the groups...

     Start Group

        // First we name the group....

        // Every group within the component must be

        // assigned a unique ID and a class...

        // and optionally a description

           Name = "Processors"

           Id = 1

           Class = "AM|Processors|001"

           Description = "Describes the processors present in the system"

           // Then we list the groups attributes...

              Start Attribute

                // First we name the attribute....

                // Every attribute within the group must be

                // assigned a unique ID and optionally a description

                        Name = "Main processor type"

                        Id = 1

                        Description = "The type of processor currently in the system"

                        // Specify the attribute type (A string)...
```

```
                Type = String(64)

                // ...and the attribute value...

                Value = "Pentium"

            // End of attribute...

            End Attribute

            // Describe a new attribute within the group...

            Start Attribute

                Name = "Main processor speed"

                Id = 2

                Description = "Maximum speed (in MHz) of this processor"

                Type = Integer

                Value = 90

            End Attribute

            Start Attribute

                Name = "Main processor vendor ID"

                Id = 3

                Description = "This is the Vendor ID returned by the
processor in response to the CPUID instruction"

                Type = String(64)

                Value = "GenuineIntel"

            End Attribute

            // Continue until no more attributes in the group...

            // End of group...

    End Group

    // Describe a new group within the component...

    Start Group

       Name = "Operating system"

       Id = 2

       Class = "AM|OSsystem|001"
```

```
        Description = "Describes the operating system running on this system"
        Start Attribute
                Name = "Operating system"
                Id = 1
                Description = "Specifies the operating system"
                Type = String(64)
                Value = "Windows 95"
        End Attribute
        Start Attribute
                Name = "Operating system Major version"
                Id = 2
                Description = "Specifies the major version of the operating system"
                Type = Integer
                Value = 4
        End Attribute
        Start Attribute
                Name = "Operating system Minor version"
                Id = 3
                Description = "Specifies the minor version of the operating system"
                Type = Integer
                Value = 0
        End Attribute
    End Group
// Continue until no more groups in component..
// End of Component...
End Component
```

Asset Management Compressed Text File

The following is the asset management compressed text file for the above .MIF file.

000101[Asset Management Windows 95 PC information]

000201[Asset Management Windows 95 PC information|Processors]

00010500Main processor type|Pentium

00020201Main processor speed|90

00030500Main processor vendor ID|GenuineIntel

000301[Asset Management Windows 95 PC information|Operating system]

00010500Operating system|Windows 95

00020201Operating system Major version|4

00030201Operating system Minor version|0

Appendix B: Inventory Matrix

This section contains the following topics:

List of Inventory Items Reported

The Basic Scan reports a subset of the inventory reported by the Advanced Scan, and the Advanced Scan can be expected to report the items listed in the following table, logically grouped, at a minimum.

The following legends should be considered if the NRI agent is executed by restricted users:

* Item which is not reported by restricted users.

** Item which will be more elaborate or informative when reported by an administrator user.

*** Item with a value that may be influenced through remote desktop connection.

Inventory Items	Basic Scan	Advanced Scan
\System	x	x
Total Memory**	x	x
Physical Memory*	x	x
OS Reported Memory	x	x
Model*	x	x
Type*	x	x
No. of Processors	x	x
No. of COM ports*		x
No. of Printer Ports		x
No. of USB Ports*		x
No. of Video Ports*		x
Wake-up Type*		x
Vendor*	x	x
Serial Number*	x	x

Inventory Items	Basic Scan	Advanced Scan
Asset Tag	x	x
System ID*	x	x
\ System \ System Bios*		x
Vendor		x
SM Bios Version		x
Version		x
ROM Size		x
Date		x
\System \ I/O Ports*		x
Internal Designator		x
Internal Connector Type		x
External Designator		x
External Connector Type		x
External Port Type		x
\ System \ Processors	x	x
VendorID	x	x
Vendor	x	x
Model	x	x
Speed	x	x
Max. Speed*		x
Type		x
Serial Number		x
Asset Tag*		x
Socket Designation*		x
L2 Cache*		x
Hyper-Threading		x
64-bit Support		x

Inventory Items	Basic Scan	Advanced Scan
Core Count	x	x
\ System \ System Slots		x
Name		x
Bus Type		x
Current Usage		x
Speed		x
Bus Width		x
Characteristics		x
\ System \ COM Ports		x
Model		x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		x
Location		x
Status Text		x
Status		x
\ System \ Memory		x
Memory Bank		x
Size		x
Type		x
Serial Number		x
Location		x
Memory Form		x
Use		x
Frequency		x

Inventory Items	Basic Scan	Advanced Scan
\ System \ Mainboard		x
Vendor		x
Version		x
No. of Memory Banks		x
No. of Used Memory Banks		x
No. of Free Memory Banks		x
BIOS ROM Size		x
No. of CPU Sockets		x
No. of Used CPU Sockets		x
Model		x
No. of Free CPU Sockets		x
Serial Number		x
\ System Devices	x	x
No. of Sound Adapters		x
No. of Video Adapters		x
No. of Network Adapters	x	x
No. of IDE Controllers		x
No. of USB Controllers		x
No. of Floppy Controllers		x
No. of SCSI Controllers		x
No. of PCI Controllers		x
\ System Devices \ Sound Adapters		x
Model		x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		x
Location		x

Inventory Items	Basic Scan	Advanced Scan
Status Text		x
Status		x
\ System Devices \ Video Adaptors	x	x
Model	x	x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		x
Location		x
Status Text		x
Status		x
Chip Type		x
Bios String		x
Memory Size		x
X Resolution		x
Y Resolution		x
Refresh Frequency***		x
Color Depth		x
\ System Devices \ Network Adapters	x	x
Model	x	x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		x
Location		x
Status Text		x
Status		x
MAC Address	x	x

Inventory Items	Basic Scan	Advanced Scan
Speed		x
Description	x	x
Adapter Type		x
Device Name		x
\ System Devices \ IDE Controllers		x
Model		x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		x
Location		x
Status Text		x
Status		x
\ System Devices \ USB Controllers		x
Model		x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		x
Location		x
Status Text		x
Status		x
\ System Devices \ Floppy Controllers		x
Model		x
Vendor		x
Driver Provider		x
Driver		x

Inventory Items	Basic Scan	Advanced Scan
Driver Version		x
Location		x
Status Text		x
Status		x
\ System Devices \ SCSI Controllers		x
Model		x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		x
Location		x
Status Text		x
Status		x
\ System Devices \ PCI Controllers		x
Vendor		x
Model		x
Revision		x
\ Operating System	x	x
Version	x	x
Vendor		
Release		
Service Pack	x	x
Operating System	x	x
Language	x	x
Local Language		x
Virtual Machine	x	x
Windows Update Count		x

Inventory Items	Basic Scan	Advanced Scan
Platform Type		x
\ Operating System \ Regional Settings		x
Country Code		x
Country Name		x
Currency Style		x
Currency Symbol		x
Data Separator		x
Date Format		x
Date Separator		x
Decimal Separator		x
Language		x
Measurement System		x
Significant Digits in Currency		x
Thousands Separator		x
Time Format		x
Time Separator		x
\ Operating System \ Display Settings		x
Adapter Name		x
Resolution		x
Horizontal Resolution		x
Vertical Resolution		x
Display Frequency***		x
Color Depth		x
\ Operating System \ Installation		x
Version		x
Major Version		x
Minor Version		x

Inventory Items	Basic Scan	Advanced Scan
Service Pack		x
Build Number		x
Name		x
Vendor		x
Product ID		x
Product Key		x
Organization		x
Licensee		x
System Path		x
Install Date		x
Language		x
Hyper V Enable		x
Hyper V Capable		x
\ Operating System \ System Updates		x
Update		x
Description		x
Installed By		x
Installed Date		x
Type		x
Uninstall Program		x
\ Operating System \ Cron Queue		x
UserName		x
Entry Index		x
Minute		x
Hour		x
Day of Month		x
Month		x
Day of Week		x

Inventory Items	Basic Scan	Advanced Scan
Command		x
\ Operating System \ Installed Patches		x
Name		x
\ Operating System \ Swap Devices		x
Size		x
Free		x
Device		x
\ Network		
Computer Name	x	x
Computer Description		x
IP Address	x	x
Host Name	x	x
Domain Name	x	x
IPv4 Enabled	x	x
IPv6 Enabled	x	x
\ Network \ TCP/IP	x	x
IP Address	x	x
Subnet Mask		x
MAC Address		x
Domain Name Server		x
Network Adaptor	x	x
Driver		x
Driver Version		x
Driver Provider		x
Host Name	x	x
Domain	x	x

Inventory Items	Basic Scan	Advanced Scan
Default Gateway		x
DHCP		x
DHCP Server		x
Lease Obtained		x
Lease Expires		x
WINS*		x
Primary WINS Server*		x
Secondary WINS Server*		x
Device Name		x
\ Network \ DNS		x
Server		x
Sequence		x
Domain		x
Server Name		x
Host		x
\ Network \ WINS		x
Network Adapter		x
Secondary WINS Server		x
Primary WINS Server		x
\ Network \TCP/IPv6 \ IPv6 Addresses	x	x
IPv6 Address	x	x
Zone ID		x
MAC Address		x
Domain	x	x
DHCP		x
Device Name		x
Prefix Origin		x

Inventory Items	Basic Scan	Advanced Scan
Suffix Origin		x
Valid Lifetime		x
Preferred Lifetime		x
Lease Lifetime		x
DAD State		x
Type		x
Network Adapter	x	x
Primary DNS Server		x
DHCPv6 Server		x
Tunnel Type		x
Default Gateway		x
WINS Server		x
Host Name		x
Prefix Length		x
\ Network \ TCP/IPv6 \ DNS		x
Server		x
Zone ID		x
Sequence		x
\ Network \ TCP/IPv6 \ Gateways		x
Address		x
Zone ID **Note:** The IPv6 inventory does not include any information about gateways on Windows XP and Windows 2003 Server. It provides gateways information only on Windows Vista and later versions of Windows.		x
\ Storage	x	x
No. of Fixed Drives	x	x
No. of Floppy Drives		x

Inventory Items	Basic Scan	Advanced Scan
No. of CD-ROM Drives		x
Total Disk Space		x
\ Storage \ Fixed Drives		x
Model		x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		
Location**		x
Status Text		x
Status		x
Serial Number*		x
Firmware Version*		x
Target ID*		x
Interface		x
Device		x
Size*		x
Cache Size*		x
Cylinder*		x
Tracks Per Cylinder*		x
Sectors per Track*		x
Bytes Per Sector*		x
\ Storage \ Floppy Drives		x
Model		x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		x

Inventory Items	Basic Scan	Advanced Scan
Location		x
Status Text		x
Status		x
Device		x
\ Storage \ CD-ROM Drives		x
Model		x
Vendor**		x
Driver Provider		x
Driver		x
Driver Version		x
Location		x
Status Text		x
Status		x
Serial Number*		x
Firmware Revision*		x
Target ID*		x
Interface		x
Device		x
DVD Drive*		x
\ File System	x	x
No. of Logical Volumes		x
No. of Local File Systems		x
No. of Partitions		x
\ File System \ Local File Systems	x	x
Name	x	x
Size	x	x
Used	x	x

Inventory Items	Basic Scan	Advanced Scan
Free	x	x
Used %	x	x
Type	x	x
File System		x
Volume Label		x
Mount Point		x
Device		x
Serial Number		x
\ File System \ Logical Volumes		x
Name		x
Volume Size		x
Volume Used		x
Volume Free		x
Volume Used %		x
File System		x
Drive Model		x
Drive Device		x
Volume Device		x
\ File System \ Partitions		x
Bootable Device		x
Drive Device		x
File System		x
Partition Number		x
Partition Type		x
Recognized Partition		x
Size		x
Start Offset (Byte)		x
Volume Device		x

Inventory Items	Basic Scan	Advanced Scan
\ Input Devices		
Keyboard		x
Pointing Device		x
\ Input Devices \ Keyboard		x
Model		x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		x
Location		x
Status Text		x
Status		x
\ Input Devices \ Pointing Devices		x
Model		x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		x
Location		x
Status Text		x
Status		x
No. of Buttons		x
Left Handed		x
Type		x
Wheel***		x
Button Swapped		x

Inventory Items	Basic Scan	Advanced Scan
\ External Devices		x
No. of Monitors		x
No. of Printers		x
Default Printer		x
\ External Devices \ Monitor		x
Model		x
Vendor		x
Driver Provider		x
Driver		x
Driver Version		x
Location		x
Status Text		x
Status		x
Max. Resolution		x
Serial Number		x
Production Year		x
Production Week		x
\ External Devices \ Local Printer		x
Model		x
Location		x
Comment		x
Default Printer		x
Port		x
\ Power Settings		x
Support Hibernation		x
UPS Present		x

Inventory Items	Basic Scan	Advanced Scan
\ Power Settings \AC Adapter		x
AC Power		x
Driver		x
Driver Provider		x
Driver Version		x
Location		x
Model		x
Status		x
Status Text		x
Vendor		x
\ Power Settings \ Power Policy		x
AC – Hard Disk Timeout		x
AC – Hibernate Timeout		x
AC – Monitor Timeout		x
AC – Sleep Timeout		x
Power Scheme Name		x
Support Hibernation		x
\ Power Settings \ Power Policy \ AC Policy		x
Display Dimming		x
Dynamic Throttle		x
Fan Throttle Tolerance %		x
Forced Throttle %		x
Hard Disk Spindown Timeout		x
Idle Sensitivity %		x
Idle Timeout		x
Max Sleep		x
Min Sleep		x
Min Throttle %		x

Inventory Items	Basic Scan	Advanced Scan
Optimize For Power		x
Reduced Latency Sleep		x
Suspend to Hibernate timeout		x
Video Timeout		x
Win Logon Flags		x
Broadcast Capacity Resolution (only shown if a battery is present)		x
Lid Closed (only shown if a Lid is present, that is, it is a laptop)		x
On Lid Open (only shown if a Lid is present)		x
\ Power Settings \ Power Policy \ AC Policy \ Power Button Pressed **(Also for \ Power Settings \ Power Policy \ AC Policy \ Sleep Button Pressed, \ Power Settings \ Power Policy \ AC Policy \ On Enter Idle State, and \ Power Settings \ Power Policy \ AC Policy \ Over Throttled)**		x
Action		x
EventCode : Clears a user power button press		x
EventCode : Response to a user power button press		x
EventCode : Shutdown/Off		x
EventCode : Specifies a program to be executed		x
EventCode : User notified using sound		x
EventCode : User notified using the UI		x
Flags : Disable all wake events		x
Flags : Force critical suspension		x
Flags : Lightest first		x
Flags : Lock console		x
Flags : Override apps		x
Flags : Query allowed		x
Flags : UI allowed		x

Inventory Items	Basic Scan	Advanced Scan
\ System Status	x	x
Boot Time		x
Hardware Scan	x	x
Up Time		x
DSM agent user account	x	x
DSM agent user privileged	x	x
No. of Current users		x
FIPS Mode	x	
Software Signature File Creation Date		x
Software Signature File Delivery Date		x
\ System Status\ Agent Software		x
DSM agent user privileged		x
DSM agent user account		x
Vendor		x
Version		x
Build		x
Basedir		x
\ System Status\ Agent Software \ Inventory Module		x
Description		x
File Size		x
Version		x
File Date		x
Build		x
File		x
\ System Status\ Logged in users		x
User Identification		x

Inventory Items	Basic Scan	Advanced Scan
Login at		x
Login from		x
User Name		x

Appendix C: Inventory File Properties

DCS creates an inventory file for each inventory detection module that you configured in the collect task that is linked to the agent computer. The inventory file is available in the agent's working directory. It contains one component (top-level group), the name of which is taken from the inventory detection module configuration.

The following sections describe the tables, groups, and the attributes in the inventory file.

This section contains the following topics:

Status (Group)

The Status group contains the following information about the overall status of the compliance check:

Attribute	Type	Description
Check completed	boolean	Specifies if the check completed is successful or not
Status	string	Specifies the reasons if the check is not completed

Status/Input Files (Table)

The Status/Input Files table contains the following information about the SCAP data stream files that are processed:

Attribute	Type	Description
Name	string	Specifies the file name of the input SCAP data stream
Type	string	Specifies if the input file is an XCCDF or OVAL file
Location	string	Specifies the location of the input file
Timestamp	string	Specifies the date and time when the input file was created
Size	int64	Specifies the size of the input file in bytes

Status/Output Files (Table)

The Status/Output Files contains the following information about the output files that the scanner produces:

Note: This table does not include the inventory file.

Attribute	Type	Description
Name	string	Specifies the output file name
Type	string	Specifies if the output file is an XCCDF or OVAL file
Location	string	Specifies the location of the output file
Timestamp	string	Specifies the date and time when the output file was created
Size	int64	Specifies the size of the output file in bytes

General (Group)

The General group contains the following information about the benchmark and the test:

Attribute	Type	Description
ID	string	Specifies a unique ID for the test result reported in a particular inventory file
Profile	string	Optional. Specifies the ID of the profile applied during the benchmark application
Start time	string	Specifies the time when the benchmark application or compliance check started
End time	string	Specifies when the check ended. Both start and end times are given as strings in the format used for XCCDF files
Benchmark Ref	int64	Specifies a reference (filename or link) to the used XCCDF file
Title	string	Optional
Remark	string	Optional
Organization	string	Optional. Specifies the organization that is responsible for the results
Organization_2, Organization_3...	string	Optional. Specifies additional organizations. If the additional organizations are hierarchical then they must appear in high-to-low order

General/Identity (Optional Group)

The General/Identity optional group contains the following information about the user account used for the test:

Attribute	Type	Description
Name	string	Specifies the name of the identity (user account) of the benchmark
Privileged	bool	Specifies if the identity was privileged (administrator)
Authenticated	bool	Specifies if the identity was authenticated

Target (Group)

The Target group contains the following information about the target computer where scanner performed the benchmark test:

Attribute	Type	Description
Name	string	Specifies the name of the target computer on which the benchmark test was applied
Address	string	Optional. Specifies the network address of the target computer
Address 2, 3,...	string	Optional. Specifies additional network addresses

Target/Facts (Optional Table)

The Target/Facts optional table contains the following information about the target computer:

Attribute	Type	Description
Name	string	Specifies the name of the fact (a URL)
Type	string	Specifies the "number", "string" or "boolean"
Value	string	Specifies the value

Set Values (Table)

The Set Values table contains the following information about the values used for the value objects during the test:

Attribute	Type	Description
ID	string	Specifies the ID of the used value
Value	string	Specifies the used value

Rule Results/<rule id> (Group)

The Rule Results/*rule id* group contains the following information about the groups created for each rule that is selected during the test:

Attribute	Type	Description
Idref	string	Specifies the name of the fact (a URL)
Role	string	
Time	string	Specifies the time when the result was generated
Severity	string	
Version	string	
Weight	double	Specifies the weight of the result in the weighted scoring models
Result	string	Specifies the result such as, "pass", "fail", "error", "unknown", "notchecked" or "notapplicable"

Rule Results/<rule id>/Idents (Optional Table)

The Rule Results/*rule id*/Idents optional table contains a table of globally meaningful identifiers for the rule such as CCE IDs with the following information:

Attribute	Type	Description
System	string	Specifies a URL identifying the naming system
Name	string	Specifies the name of the identifier in the naming system

Scores (Table)

The Scores table contains the following information about the test scores according to the models specified in the benchmark:

Attribute	Type	Description
Model	string	Specifies the URL of the scoring model
Score	double	Specifies the achieved score
Maximum	double	Specifies the possible maximum score

Appendix D: Platform Names and IDs

The following table includes the platforms that can be specified in the *class_id* tag in the *general* section of the inventory file. The class IDs can be specified as a numeric value or using the string value.

Note: For the latest list of supported operating systems, refer to the Certification Matrix.

Class ID (Platform ID)	Platform Name
0	Unclassified
2	Windows
3	Windows Server 2003
4	Windows Server 2003 Web Edition
5	Windows Server 2003 Enterprise Edition
6	Windows Server 2003 Datacenter Edition
7	Windows 2000
8	Windows 2000 Professional
9	Windows 2000 Advanced Server
10	Windows 2000 DC Server
11	Windows 2000 Server
12	Windows 9x
13	Windows 95
14	Windows 98
15	Windows 98 SE
16	Windows ME
17	Windows NT
18	Windows NT Workstation 3.51
19	Windows NT workstation 4.0
20	Windows NT Server 4.0
21	Windows NT Server 4.0 Enterprise
22	Windows XP
23	Windows XP Home

24	Windows XP Professional
25	Windows XP Media Center Edition
26	Windows XP Tablet PC Edition
27	Windows XP Professional x64 Edition
28	Windows 3x
29	Windows 3.0
30	Windows 3.1
31	Windows 3.11
32	Windows CE
33	Windows CE 2.00
34	Windows CE 2.01
35	Windows CE 2.11
36	Windows CE 3.00
37	Windows Mobile 2003
38	Palm OS
39	Symbian
40	Unix
41	DG_UX
42	Tru64
43	HP Tru64 4
44	DEC 3.0
45	DEC 3.2
46	DEC 4.0
47	HP Tru64 5
48	FUJIUxp
49	HPUnix
50	HPUX 9
51	HPUX 10
52	HPUX 11
53	ICLUnix
54	Linux

74	NCRUnix
75	NCR SV
76	NCR SV 2x
77	NCR SV 3x
78	NCR SST
79	NCR SST S4
80	NCR SST S4I
81	AIX
82	AIX 3.2
83	AIX 4.1
84	AIX 4.2
85	AIX 4.3
86	AIX 5.0
87	AIX 5.1
88	AIX 5.2
89	Unixware
90	Unixware 2.0
91	Unixware 2.1
92	Unixware 7.x
93	SCO OpenServer 3.2
94	SCO OpenServer 5.2
95	Sinix
96	Sinix 5.42
97	Sinix 5.43
98	Sinix 5.44
99	Sinix 5.45
100	Irix
101	Irix 6.2
102	Irix 6.3
103	Irix 6.4
104	Irix 6.5

105	Dynix
106	Dynix 4.2.x
107	Dynix 4.4.x
108	Solaris
109	Solaris 2.3
110	Solaris 2.4
111	Solaris 2.5
112	Solaris 2.6
113	Solaris 7
114	Solaris 8
115	Solaris 9
116	SunOS 4.1.4
117	OS/2
118	AS400
119	OS390
120	DOS
121	Macintosh
122	Macintosh 10.0
123	Macintosh 10.1
124	Macintosh 10.2
125	Macintosh 10.3
126	Macintosh 10.4
127	Netware
128	OpenVMS
129	OpenVMS 5.5
130	OpenVMS 6.0
131	OpenVMS 6.1
132	OpenVMS 6.2
133	OpenVMS 7.0
134	OpenVMS 7.1
135	OpenVMS 7.2

136	Windows NT Server 3.51
137	Windows XP Embedded
138	Windows Server 2003 Standard Edition
139	Windows Small Business Server 2003
140	Windows Server 2003 Enterprise x64 Edition
141	Windows Server 2003 Enterprise 64-Bit Itanium Edition
201	Caldera/SCO Linux
202	Caldera Linux 3.0
203	Caldera Linux 3.1
204	SCO Linux 4.0
205	Connectiva
206	Connectiva Linux 8
207	Connectiva Linux 9
208	Connectiva Linux 10
209	Debian GNU/Linux
210	Debian GNU/Linux 2.0
211	Debian GNU/Linux 2.1
212	Debian GNU/Linux 2.2
213	Debian GNU/Linux 3.0
214	Fedora
215	Fedora Core 1
216	Fedora Core 2
217	Miracle Linux
218	Miracle Linux 2.1
219	Miracle Linux 3.0
220	Red Flag Linux
221	Red Flag Linux 3.0
222	Red Flag Linux 3.2
223	Red Flag Linux 4.0
224	Red Flag Linux 4.1
225	RedHat Linux

226	Red Hat Linux 7.0
227	Red Hat Linux 7.1
228	Red Hat Linux 7.2
229	Red Hat Linux 7.3
230	Red Hat Linux 8.0
231	Red Hat Enterprise Linux ES 2.1
232	Red Hat Enterprise Linux AS 2.1
233	Red Hat Enterprise Linux WS 2.1
234	Red Hat Enterprise Linux ES 3
235	Red Hat Enterprise Linux AS 3
236	Red Hat Enterprise Linux WS 3
237	Red Hat Enterprise Linux ES 4.0
238	Red Hat Enterprise Linux AS 4.0
239	Red Hat Enterprise Linux WS 4.0
240	Red Hat Desktop
241	Sun Java Desktop System
242	Sun Java Desktop System 2003
243	Sun Java Desktop System 2
244	SuSE Linux
245	SuSE Linux 7.0
246	SuSE Linux 7.1
247	SuSE Linux 7.3
248	SuSE Linux 8.0
249	SuSE Linux 8.1
250	SuSE Linux 8.2
251	SuSE Linux Professional 9
252	SuSE Linux Professional 9.1
253	SuSE Linux Personal 9
254	SuSE Linux Personal 9.1
255	SuSE Linux Enterprise Server 7
256	SuSE Linux Enterprise Server 8

257	SuSE Linux Enterprise Server 9
258	SuSE Standard Server 8
259	SuSE Desktop 1
260	SuSE Desktop 2
261	Turbo Linux
262	TurboLinux Server 8
263	TurboLinux Workstation 8
264	TurboLinux Enterprise 8
265	TurboLinux 10
266	Xandros Desktop OS
267	Xandros Desktop 1.0
268	Xandros Desktop 1.1
269	Xandros Desktop 2.0
270	OpenVMS 7.3
271	Unixware 7.1.4
272	SCO OpenServer
273	SCO OpenServer 5.7
274	SCO Unix
275	Nokia AdminSuite
276	Nokia AdminSuite 1.0
277	OpenVMS 8.0
278	OpenVMS 8.2
279	Solaris 10
280	AIX 5.3
281	NCR MP-RAS 3.02
282	HPUX 11.10
283	HPUX 11.22
284	HPUX 11.23
285	HPUX 11.00
286	HP Tru64 5.1
287	HP Tru64 5.0

288	HP Tru64 4.0F
289	HP Tru64 4.0G
290	HPUX 10.00
291	HPUX 10.10
292	HPUX 10.20
293	HPUX 10.30
294	HPUX 11.20
296	AIX 5
297	Windows Vista
298	Windows Vista Business
299	Windows Vista Enterprise
300	Windows Vista Home Premium
301	Windows Vista Home Basic
302	Windows Vista Ultimate
303	Windows Vista Business x64 Edition
304	Windows Vista Enterprise x64 Edition
305	Windows Vista Home Premium x64 Edition
306	Windows Vista Home Basic x64 Edition
307	Windows Vista Ultimate x64 Edition
308	Windows Mobile
309	Windows Mobile 5.0
2001	_3COM
2002	Acer
2003	Adaptec
2004	Alcatel
2005	Altos
2006	Apple
2007	Asante
2008	ATT
2009	Avaya
2010	Banyan

2011	Bay
2012	Belkin
2013	Breeze
2014	Brocade
2015	Cabletron
2016	Chaparral
2017	Ciprico
2018	Crossroads
2019	Chipcom
2020	Cisco
2021	Compaq
2022	DEC
2023	Dell
2024	Digital
2025	EMC
2026	Emulex
2027	Epson
2028	Ericcson
2029	EthAirNet
2030	Exabyte
2031	Extreme
2032	Fore
2033	Foundry
2034	Fujitsu
2035	Gadzoox
2036	Gateway
2037	Gator
2038	Hitachi
2039	ICL
2040	Intel
2041	Intergraph

2042	IPX
2043	HP
2044	IBM
2045	JNI
2046	KarlNet
2047	LSILogic
2048	Linksys
2049	McDATA
2050	Micom
2051	Motorola
2052	MultiNet
2053	NBase
2054	NCD
2055	NCR
2056	NEC
2057	NetApps
2058	NetBotz
2059	NetGear
2060	NetJet
2061	NetQue
2062	NetWorth
2063	Nokia
2064	Nortel
2065	Novell
2066	Orinoco
2067	Palm
2068	Pathlight
2069	Qlogic
2070	RoamAbout
2071	SCO
2072	Samsung

2073	Siemens
2074	Silicon Graphics
2075	SMC
2076	StorageTek
2077	Sun
2078	Symbol
2079	SynOptics
2080	Tandberg
2081	Tandem
2082	Tektronix
2083	Telebit
2084	Toshiba
2085	Troika
2086	Unisys
2087	Vitalink
2088	Vixel
2089	Xerox
2090	Xiotech
2091	Xircom
2092	Xylan
2093	Xylogics
2094	Xyplex
2095	Wellfleet
2096	Brocade 2400
2097	Brocade 3200
2098	Brocade 3500
2099	Brocade 12000
2100	Dell Axim
2101	Dell Latitude
2102	Dell Optiplex
2103	EMC Clariion

2104	EMC Symmetrix
2105	Siemens optiPoint
2106	Dell Optiplex GL110
2107	Dell Optiplex GLX270
2108	Bay Bridge
2109	Bay Hub
2110	Bay Switch
2111	HP Printer
2112	HP Switch
2113	HP Bridge
2114	HP Computer
2115	HP Server
2116	_3COM Switch
2117	HP PC
2118	Brocade Switch
2119	Alcatel IP Phone
2120	Dell Switch
2121	Cisco Router
2122	Cisco Switch
2123	Enterasys
2124	Enterasys Router
2125	Enterasys Switch
2126	IBM Router
2127	IBM Switch
2128	IBM MSS
2129	IBM 8265
2130	IBM 8271
2131	IBM 8371
2132	Foundry Switch
2133	Dell Computer
2134	Cisco Fabric Switch

2135	Adaptec HBA
2136	AMCC
2137	AMCC HBA
2138	Chaparral Bridge
2139	CNT
2140	CNT Switch
2141	Crossroads Bridge
2142	Emulex HBA
2143	Emulex Hub
2144	Gadzoox Switch
2145	Gadzoox Hub
2146	HP HBA
2147	IBM HBA
2148	LSILogic HBA
2149	McDATA Director
2150	McDATA Switch
2151	McDATA EFC Manager
2152	Pathlight Bridge
2153	Qlogic HBA
2154	Qlogic Switch
2155	Vixel Switch
2156	Vixel Hub
2157	Cisco Access Point
2158	Intel Access Point
2159	Symbol Access Point
2160	Compaq Access Point
2161	Belkin Access Point
2162	NetGear Access Point
2163	Linksys Access Point
2164	RoamAbout Access Point
2165	KarlNet Access Point

2166	SMC Access Point
2167	Orinoco Access Point
2168	BreezeCOM Access Point
2169	EthAirNet Access Point
2170	Avaya Access Point
2171	Ericcson Access Point
2172	_3COM Access Point
2173	Nortel Access Point
2174	Ascend
2175	Kyocera
2176	Lexmark
2177	Sato
2178	Ascend Router
2179	Epson Printer
2180	Kyocera Printer
2181	Lexmark Printer
2182	Sato Printer
3001	Computer
3002	Networking Device
3003	Printer
3004	Storage Device
3005	Telecom Device
3006	Desktop
3007	Laptop
3008	Mainframe
3009	Minicomputer
3010	Server
3011	Bridge
3012	Hub
3013	Network Adapter
3014	Modem

3015	Router
3016	Switch
3017	Bubble Jet Printer
3018	Dot Matrix Printer
3019	Laser Printer
3020	Line Printer
3021	Microfiche
3022	Plotter
3023	Tape Subsystem
3024	Disk Subsystem
3025	Phone
3026	Telco Hub
3027	Telco Circuit
3028	Telco Router
3029	Network Interface Card
3030	Host Bus Adapter
3031	iSCSI Adapter
3032	Wireless Network Card
3033	SAN Switch
3034	Ethernet Switch
3035	Inkjet
3036	Tape Drive
3037	Tape Enclosure
3038	Disk Drive
3039	Disk Enclosure
3040	IP Phone
3041	Optical Disk Subsystem
3042	WORM Enclosure
3044	Tape
3045	Tape Subsystem Component
3046	Disk Subsystem Component

3047	Optical Disk Subsystem Component
3048	Robotic Arm
3049	Optical Disk
3050	CD-ROM Drive
3051	DVD Drive
4001	Virtual Machine
4002	Virtual Node
4003	VMware Session
4004	MS Virtual Server
4005	Cluster Node
4006	Logical Partition
4007	HP Hard Partition
5001	x86
5002	Itanium (64-bit)
5003	Windows Server
5004	Windows Workstation
5005	Mobile
5006	Windows Server 64-Bit
5007	Windows NT Server 3.x
5008	Windows NT Server 4.0 or later
5009	Windows 9x Editions
5010	Windows NT Workstation 4.0 or later
5011	Windows CE or later
5013	Windows Server 2003 64-Bit or later
5014	Windows NT Server 4.0 Editions
5015	Windows 2000 Server or later
5016	Windows NT Workstation 4.0 Editions
5017	Windows 2000 Workstation or later
5018	Windows CE 2x
5019	Windows CE 3x
5020	Windows Server 2003 64-Bit

5021	Windows 2000 Server Editions
5022	Windows Server 2003 or later
5023	Windows 2000 Workstation Editions
5024	Windows XP or later
5025	Windows Server 2003 Editions
5026	Windows Server 2003 (x64)
5027	Windows XP Editions
5028	Windows XP (64-bit)
5029	Macintosh 10.x or later
5030	HP Tru64 DEC
5031	HP Tru64 4.0x or later
5032	HP Tru64 4.0x
5033	HP Tru64 5.0 or later
5034	HP Tru64 5.0x
5035	HP Tru64 5.1 or later
5036	HP Tru64 5.1x
5037	HPUX 10.x or later
5038	HPUX 10.x
5039	HPUX 11.x or later
5040	HPUX 11.x
5042	AIX 5.x or later
5043	AIX 5.x
5044	NCR SST S4 or later
5045	NCR SST S4x
5046	NCR MP-RAS 3.x or later
5047	NCR MP-RAS 3.x
5048	Netware 5.1 SP1
5049	Netware 6.5 SP3
5050	Netware 5.1 SP3
5051	Netware 6.5
5052	Netware 5.1

5053	Netware 5.1 SP7
5054	Netware 5.1 SP6
5055	Netware 6.5 SP1.1
5056	Netware 6.5 SP2
5057	Netware 5.1 SP2
5058	Netware 5.1 SP4
5059	Netware 5.1 SP8
5060	Netware 5.1 SP5
5061	Red Hat Linux 9.0
5062	ONS
5063	IOS
5064	OSF/1
5065	Netware 6.5 SP4
5066	Netware 6.5 SP5
5067	z/OS
5068	Windows Vista Editions
5069	Windows Vista (64-bit)
5082	SuSE Linux Enterprise Server 10
5083	SuSE Linux Enterprise Server 10.1
5084	SuSE Linux Enterprise Desktop 10
5085	SuSE Linux Enterprise Desktop 10.1
5088	Windows Server 2003 Standard x64 Edition
5089	Windows Server 2003 Datacenter x64 Edition
5090	Unixware 7.1.3
5091	Macintosh 10.5
5092	Red Hat Enterprise Linux Server 5
5093	Red Hat Enterprise Linux Desktop 5
5094	Red Hat Enterprise Linux 5 Advanced Platform
5095	Solaris Domain Partition
5096	Sun Servers
5097	Sun Fire E25K

5098	Sun Fire E20K
5099	Sun Fire 15K
5100	Sun Fire 12K
5101	Sun Enterprise 10000
5102	Sun Enterprise 6500
5103	Sun Enterprise 5500
5104	Sun Enterprise 4500
5105	Sun Enterprise 3500
5106	Windows Embedded for POS
5107	Windows Embedded for Point of Service Version 1
5108	Windows Embedded
5109	Windows Embedded for Point of Service
5110	Windows Embedded for Point of Service 1.0
5111	HPUX 11.11
5112	HPUX 11.31
5114	RTOS
5115	Sun RTOS
5116	Red Hat Enterprise Linux Server 5.1
5117	Red Hat Enterprise Linux Desktop 5.1
5118	Red Hat Enterprise Linux 5.1 Advanced Platform
5119	Windows Server 2008
5120	Windows Server 2008 Standard Edition
5121	Windows Server 2008 Standard x64 Edition
5122	Windows Server 2008 Enterprise Edition
5123	Windows Server 2008 Enterprise x64 Edition
5124	Windows Server 2008 Datacenter Edition
5125	Windows Server 2008 Datacenter x64 Edition
5126	Windows Web Server 2008
5127	Windows Web Server 2008 x64 Edition
5128	Windows Storage Server 2008
5129	Windows Storage Server 2008 x64 Edition

5130	Windows Small Business Server 2008
5131	Windows Essential Business Server 2008
5132	Windows Server 2008 IA-64 Edition
5133	Windows Server 2008 IA-64 or later
5134	Windows Server 2008 or later
5135	Windows Server 2008 Editions (32bit)
5136	Windows Server 2008 (x64)
5137	Windows Mobile 6.0
5138	AIX 6
5139	AIX 6.1
5140	AIX 6.x or later
5141	AIX 6.x
5142	RIM
5143	RIM OS
5144	Hypervisor
5145	Microsoft Hyper-V
5146	VMware ESX
5147	VMware ESXi
5148	Windows Server 2008 Hyper-V
5149	VMWare ESX Server 3.5
5150	VMWare ESXi 3.5
5153	Virtual Host Machine
5154	HP Chassis
5155	IBM Server
5156	Windows 7
5157	Windows 7 Enterprise
5158	Windows 7 Enterprise x64 Edition
5159	Windows 7 Home Premium
5160	Windows 7 Home Premium x64 Edition
5161	Windows 7 Professional
5162	Windows 7 Professional x64 Edition

5163	Windows 7 Starter
5164	Windows 7 Home Basic
5165	Windows 7 Home Basic x64 Edition
5166	Windows 7 Ultimate
5167	Windows 7 Ultimate x64 Edition
5168	Macintosh 10.6
5169	SUSE Linux Enterprise Server 11
5188	Windows Server 2008 R2
5189	Windows Server 2008 R2 Datacenter x64 Edition
5190	Windows Server 2008 R2 Enterprise x64 Edition
5191	Windows Server 2008 R2 IA-64 Edition
5192	Windows Server 2008 R2 Standard x64 Edition
5193	Windows Server 2008 Server Core Standard Edition
5194	Windows Server 2008 Server Core Standard x64 Edition
5195	Windows Server 2008 R2 Server Core Standard x64 Edition
5196	Windows Server 2008 Server Core Enterprise Edition
5197	Windows Server 2008 Server Core Enterprise x64 Edition
5198	Windows Server 2008 R2 Server Core Enterprise x64 Edition
5199	Windows Server 2008 Server Core Datacenter Edition
5200	Windows Server 2008 Server Core Datacenter x64 Edition
5201	Windows Server 2008 R2 Server Core Datacenter x64 Edition
5202	Windows Web Server 2008 R2 x64 Edition
5203	Windows Web Server 2008 Server Core Edition
5204	Windows Web Server 2008 Server Core x64 Edition
5205	SuSE Linux Enterprise Desktop 11
5206	Windows Web Server 2008 R2 Server Core x64 Edition
5210	Windows Server 2008 Server Core Installation
5211	Windows Server 2008 (x64) Server Core Installation
5212	Windows Server 2008 R2 or later
5213	Windows Server 2008 R2 Server Core Installation
5214	Windows Vista or later

5215	Windows 7 or later
5216	Windows 7 Editions
5217	Windows Vista (x64) or later
5218	Windows 7 (x64) or later
5219	Windows 7 (64-bit)
5220	openSUSE
5221	openSUSE 11.0
5222	openSUSE 11.1
5223	Sun eXtended System Control Facility (XSCF)
5224	Windows Mobile 6.1
5225	Unknown
5226	VMWare ESX Server 4.0
5229	Windows Embedded POSReady 2009
5230	Windows Storage Server 2008 Standard Edition
5231	Windows Storage Server 2008 Basic Edition
5232	Windows Storage Server 2008 Enterprise Edition
5233	Windows Storage Server 2008 Workgroup Edition
5234	Windows Storage Server 2008 Standard X64 Edition
5235	Windows Storage Server 2008 Basic X64 Edition
5236	Windows Storage Server 2008 Enterprise X64 Edition
5237	Windows Storage Server 2008 Workgroup X64 Edition
5238	VMWare ESX Server 4.1
5239	Windows Mobile 6.5
5240	VMware ESXi 4.0
5241	VMware ESXi 4.1
5242	Citrix XenServer
5243	Citrix XenServer 4.0
5244	Citrix XenServer 4.1
5245	Citrix XenServer 5.0
5246	Citrix XenServer 5.5
5247	Citrix XenServer 5.6

5248	Windows Server 2008 R2 Hyper-V
5249	Hyper-V Server 2008
5250	Hyper-V Server 2008 R2
5251	Red Hat Enterprise Linux 5.4 Advanced Platform
5252	Red Hat Enterprise Linux Desktop 5.4
5253	Red Hat Enterprise Linux Server 5.4
5254	Windows Embedded Standard 7 Edition
5255	Windows Embedded Standard 7 X64 Edition
5256	AIX 7
5257	AIX 7.1
5258	Red Hat Enterprise Linux Server 6.0 x86
5259	Red Hat Enterprise Linux Server 6.0 x86_64
5260	Debian GNU/Linux 5.0
5261	Debian GNU/Linux 6.0
5262	Debian GNU/Linux 7.0
5263	Ubuntu Linux
5264	Ubuntu Linux 10.04
5265	Ubuntu Linux 10.10
5266	Ubuntu Linux 11.04
5267	Ubuntu Linux 11.10
5268	Ubuntu Linux 12.04
5269	Macintosh 10.7
5270	Red Hat Enterprise Linux Server 6.1 x86
5271	Red Hat Enterprise Linux Server 6.1 x86_64
5272	openSUSE 11.2 x86
5273	openSUSE 11.2 x86_64
5274	openSUSE 11.3 x86
5275	openSUSE 11.3 x86_64
5276	openSUSE 11.4 x86
5277	openSUSE 11.4 x86_64
5278	Red Hat Enterprise Linux Server 6.2 x86

5279	Red Hat Enterprise Linux Server 6.2 x86_64
5280	Debian GNU/Linux 1.1
5281	Debian GNU/Linux 1.2
5282	Debian GNU/Linux 1.3
5283	Debian GNU/Linux 3.1
5284	Debian GNU/Linux 4.0
5285	Ubuntu Linux 4.10
5286	Ubuntu Linux 5.04
5287	Ubuntu Linux 5.10
5288	Ubuntu Linux 6.06
5289	Ubuntu Linux 6.10
5290	Ubuntu Linux 7.04
5291	Ubuntu Linux 7.10
5292	Ubuntu Linux 8.04
5293	Ubuntu Linux 8.10
5294	Ubuntu Linux 9.04
5295	Ubuntu Linux 9.10
5296	Windows CE 6.00
5297	Windows CE 5.00
5298	Windows CE .NET 4.0
5299	Windows CE .NET 4.1
5300	Windows CE .NET 4.2
5301	Oracle Linux
5302	Oracle Enterprise Linux Server 5 x86
5303	Oracle Enterprise Linux Server 5 x86_64
5304	Oracle Linux Server 6 x86
5305	Oracle Linux Server 6 x86_64
5307	Windows Storage Server 2008 R2
5308	Windows Storage Server 2008 R2 Essentials x64 Edition
5309	Red Hat Enterprise Linux Server 6.3 x86
5310	Red Hat Enterprise Linux Server 6.3 x86_64

5311	Windows Embedded POSReady 7
5312	Windows 8
5313	Windows 8 Pro x86
5314	Windows 8 Pro x64
5315	Windows 8 Enterprise x86
5316	Windows 8 Enterprise x64
5317	Windows Embedded standard 8 x86
5318	Windows Embedded standard 8 x64
5319	Windows 8 x86
5320	Windows 8 x64
5321	Windows Server 2012
5322	Windows Server 2012 Foundation x64
5323	Windows Server 2012 Essentials x64
5324	Windows Server 2012 Standard x64
5325	Windows Server 2012 Datacenter x64
5326	Windows 8 or later
5327	Windows 8 Editions
5328	Windows 8 (x64) or later
5329	Windows 8 (64-bit)
5330	Windows Server 2012 or later
5331	Windows Storage Server 2012
5332	Windows Storage Server 2012 Workgroup x64
5333	Windows Storage Server 2012 Standard x64
5334	Solaris 11
5335	AIX 7.x or later
5336	AIX 7.x
5337	Macintosh 10.8
5338	VMware ESXi 5.1
5339	Red Hat Enterprise Linux 5.5 Advanced Platform
5340	Red Hat Enterprise Linux 5.6 Advanced Platform
5341	Red Hat Enterprise Linux 5.7 Advanced Platform

5342	Red Hat Enterprise Linux 5.8 Advanced Platform
5343	Red Hat Enterprise Linux 5.9 Advanced Platform
5344	Red Hat Enterprise Linux Desktop 5.5
5345	Red Hat Enterprise Linux Desktop 5.6
5346	Red Hat Enterprise Linux Desktop 5.7
5347	Red Hat Enterprise Linux Desktop 5.8
5348	Red Hat Enterprise Linux Desktop 5.9
5349	Red Hat Enterprise Linux Server 5.5
5350	Red Hat Enterprise Linux Server 5.6
5351	Red Hat Enterprise Linux Server 5.7
5352	Red Hat Enterprise Linux Server 5.8
5353	Red Hat Enterprise Linux Server 5.9
5354	Red Hat Enterprise Linux Server 6.4 x86
5355	Red Hat Enterprise Linux Server 6.4 x86_64
5356	Hyper-V Server 2012
5357	Windows Server 2012 Hyper-V
5358	Android
5359	Android 2.2
5360	Android 2.3
5361	Android 3.0
5362	Android 4.0
5363	iOS 6.1.1
5364	iOS 6.1.3
5365	iOS 6.1
5366	iOS 6.0.1
5367	iOS 6.0.2
5368	iOS 5.1
5369	iOS 5
5370	BlackBerry OS
5371	BlackBerry OS 7.0
5372	BlackBerry OS 6.0

5373	BlackBerry OS 5.0
5374	Windows Phone
5375	Windows Phone 8.0
5376	Android 4.1
5377	Android 4.2
5378	BlackBerry 10.0
5379	BlackBerry 10.1
5380	iOS 6.0
5381	iOS 6.1.2
5382	iOS 6.1.4
5383	Macintosh 10.9
5384	Windows 8.1
5385	Windows 8.1 Enterprise x64
5386	Windows 8.1 Enterprise x86
5387	Windows 8.1 Pro x64
5388	Windows 8.1 Pro x86
5389	Windows 8.1 x64
5390	Windows 8.1 x86
5391	Windows Embedded 8.1 Standard x64
5392	Windows Embedded 8.1 Standard x86
5393	Windows Server 2012 R2
5394	Windows Server 2012 R2 Datacenter x64
5395	Windows Server 2012 R2 Essentials x64
5396	Windows Server 2012 R2 Standard x64
5397	openSUSE 12.1 x86
5398	openSUSE 12.1 x86_64
5399	openSUSE 12.2 x86
5400	openSUSE 12.2 x86_64
5401	openSUSE 12.3 x86
5402	openSUSE 12.3 x86_64
5403	Citrix XenServer 6.0

5404	Citrix XenServer 6.1
5405	Windows Server 2012 R2 Foundation x64
5406	Windows Server 2012 Server Core Datacenter x64
5407	Windows Server 2012 Server Core Standard x64
5408	Windows Server 2012 R2 Server Core Datacenter x64
5409	Windows Server 2012 R2 Server Core Standard x64
5410	Hyper-V Server 2012 R2
5411	Windows Server 2012 R2 Hyper-V
5412	Windows Storage Server 2012 R2
5413	Windows Storage Server 2012 R2 Standard x64
5414	Windows Storage Server 2012 R2 Workgroup x64
5415	Windows Server 2012 R2 or later
5416	Windows Server 2012 R2 Server Core Installation
5417	Windows Storage Server 2012 R2 or later
5418	Windows 8.1 or later
5419	Windows 8.1 Editions
5420	Windows 8.1 (x64) or later
5421	Windows 8.1 (64-bit)
5422	Windows Server 2012 Server Core Installation

Appendix E: SCAP Configuration Parameters

This section describes the SCAP configuration parameters that you need to specify while configuring an FDCC inventory detection module. You can specify the following parameters in the text field provided on the Configuration tab of the Create New Inventory Module dialog:

SCAPPath

Specifies the path to the SCAP data stream directory on the agent computer. This path must match the SCAP data stream directory on the domain manager. For example, for the IE7 checklist, specify FDCC-Major-Version-1.2.1.0\ie7. When the checklist is distributed to the agent computer, a similar directory structure is created under the *ITCM_installpath*\Agent\units\00000001\UAM\ SCAP_Content directory on the agent computer.

XCCDFFile

Specifies the name of the XCCDF file in the SCAP data stream that determines the compliance benchmark.

Note: This file must be present in the location specified in the SCAPPath parameter.

XCCDFID

Specifies the ID given against the Benchmark tag in the XCCDF file. For example, the benchmark ID for Windows XP checklist is FDCC-Windows-XP.

XCCDF Version

(Optional) Specifies the version of the checklist to use if there are several versions with the same XCCDFID.

XCCDFVersionOptions

Lists the versions of the checklist that are currently available for distribution. The DSM Engine uses this parameter to populate the XCCDF Version field in the SCAP Configuration dialog. This parameter is available for all the SCAP inventory modules that the DSM Engine creates.

Important! Do *not* include this parameter when you are manually creating an inventory module for a checklist. Adding or modifying this parameter can lead to invalid configuration settings and may result in empty scan results for the checklist.

ProfileOptions

Lists the valid profiles for each version of the checklist that are currently available for distribution. The DSM Engine uses this parameter to populate the XCCDF Profile field in the SCAP Configuration dialog. This parameter is available for all the SCAP inventory modules that the DSM Engine creates. However, when you create an inventory module for a checklist, do *not* include this parameter as this can display invalid profile names in the SCAP Configuration dialog.

Important! Do *not* include this parameter when you are manually creating an inventory module for a checklist. Adding or modifying this parameter can lead to invalid configuration settings and may result in empty scan results for the checklist.

CPEDictionary

(Optional) Defines the name of the CPE dictionary file. If the SCAP data stream contains a dictionary file, specify the file name against this parameter; otherwise, you can omit this parameter.

Note: This file must be present in the location specified in the SCAPPath parameter.

InvComponent

Defines the component name to use in the inventory file produced by the scanner. This value is used as the top-level group name in the inventory file and hence also appears as the inventory component name under the Inventory, SCAP category in the DSM Explorer.

CollectXCCDFResultFile

Configures the collection of the XCCDF result file for the checklist. Select this option to let the scanner securely copy the XCCDF result file from the Asset Management agent's working directory to the domain manager after every checklist scan.

Default: false

CollectOVALResultFiles

Configures the collection of OVAL result files for the checklist from the Asset Management agent's working directory to the domain manager.

Default: false

XCCDFProfile

Specifies the title of the XCCDF profile to be applied for the compliance check. Leaving the value of this parameter empty applies no profile, and thus uses all the settings inthe XCCDF file.

OutputPath

Defines the directory in which the OVAL and XCCDF result files are to be placed. You can either specify an absolute path or a path relative to the SCAP_Result_Files directory, which is under the Asset Management agent's working directory. If this field is empty, the files are stored under the default path, which is *agent working directory*\SCAP_Result_Files\Data Stream Path.

Note: The user account that runs the scan must have write access to the directory specified in this field.

OvaldiPath

Defines the directory on the agent computer that contains the Ovaldi interpreter. You can specify either an absolute path or a path relative to the bin directory of the agent installation. The OVAL interpreter shipped with this release of CA ITCM is installed under the *ITCM_installpath*\bin\ovaldi-CA directory. If your SCAP data stream requires an OVAL interpreter other than the one shipped with this release, ensure to distribute the OVAL interpreter to all the agent computers and specify the path in this field.

Default: *ITCM_installpath*\ovaldi-CA

Organization

(Optional) Defines the name of the organization that you want the *<organization>* tag to contain in the XCCDF result file. Specify the organization name and click Add to List.

Note: You can add any number organizations and move them in the order that you want. The values are hierarchical with the highest level appearing first.

Important! Do *not* include this parameter when you are manually creating an inventory module for a checklist. Adding or modifying this parameter can lead to invalid configuration settings and may result in empty scan results for the checklist.

Appendix F: CA Asset Converter for Microsoft SCCM Inventory Mapping

This section contains the following topics:

Mapping of Microsoft SCCM Hardware Asset Inventory to CA ITCM Asset Management Hardware Inventory

The following table delineates the hardware inventory data collected by the CA Asset Converter for Microsoft® SCCM and its corresponding attribute mapping in CA ITCM asset management.

AM Inventory Groups	AM Inventory Attributes	SCCM Inventory Attributes
$System$	Model	SMS_G_System_COMPUTER_SYSTEM.Model
	Total Memory	SMS_G_System_X86_PC_MEMORY.TotalPhysicalMemory
	Vendor	SMS_G_System_COMPUTER_SYSTEM.Manufacturer
	Type	SMS_System_COMPUTER_SYSTEM.SystemRole
	Asset Tag	SMS_System_COMPUTER_SYSTEM.SMBIOSAssetTag
	Domain	SMS_System_COMPUTER_SYSTEM.ResourceDomainORWorkgroup
	Last Logon User Domain	SMS_System_COMPUTER_SYSTEM.LastLogonUserDomain
	Last Logon User Name	SMS_System_COMPUTER_SYSTEM.LastLogonUserName
	Originating SMS Site Server	From job engine configuration

AM Inventory Groups	AM Inventory Attributes	SCCM Inventory Attributes
	Serial Number	SMS_G_System_SYSTEM_ENCLOSURE.Serial Number
$System$|$System Bios$	Version	Version SMS_G_System_PC_BIOS.Version
	Vendor	SMS_G_System_PC_BIOS.Manufacturer
$System$|$Processors$	VendorID	SMS_G_System_PROCESSOR.Manufacturer
	Model	(based on) SMS_G_System_PROCESSOR.ProcessorId
	Speed	SMS_G_System_PROCESSOR.CurrentClockSpeed
	Max Speed	SMS_G_System_PROCESSOR.MaxClockSpeed
	Socket Designation	SMS_G_System_PROCESSOR.SocketDesignation
$Network$	Computer Name	SMS_R_System.Name
	Domain Name	SMS_R_System.ResourceDomainORWorkgroup
	IP Address	SMS_G_System_NETWORK_ADAPTER_CONFIGURATION.IPAddress
$Network$|TCP/IP	IP Address	SMS_G_System_NETWORK_ADAPTER_CONFIGURATION.IPAddress
	Subnet Mask	SMS_G_System_NETWORK_ADAPTER_CONFIGURATION.IPSubnet
	Default Gateway	SMS_G_System_NETWORK_ADAPTER_CONFIGURATION.DefaultIPGateway
	DHCP	SMS_G_System_NETWORK_ADAPTER_CONFIGURATION.DHCPEnabled
	DHCP Server	SMS_G_System_NETWORK_ADAPTER_CONFIGURATION.DHCPServer
$System Devices$|$Sound Adapter$	Model	SMS_G_System_SOUND_DEVICE.Name
	Vendor	SMS_G_System_SOUND_DEVICE.Manufacturer

AM Inventory Groups	AM Inventory Attributes	SCCM Inventory Attributes
	Status	SMS_G_System_SOUND_DEVICE.Status
$System Devices$|$Video Adapters$	Model	SMS_G_System_VIDEO_CONTROLLER.Name
	Driver	SMS_G_System_VIDEO_CONTROLLER.InstalledDisplayDrivers
	Driver Version	SMS_G_System_VIDEO_CONTROLLER.DriverVersion
	Chip Type	SMS_G_System_VIDEO_CONTROLLER.VideoProcessor
	X Resolution	SMS_G_System_VIDEO_CONTROLLER.CurrentHorizontalResolution
	Y Resolution	SMS_G_System_VIDEO_CONTROLLER.CurrentVerticalResolution
	Refresh Frequency	SMS_G_System_VIDEO_CONTROLLER.CurrentRefreshRate
	Memory Size	SMS_G_System_VIDEO_CONTROLLER.AdapterRAM
$System Devices$$Network Adapters$	Model	SMS_G_System_NETWORK_ADAPTER.Name
	Vendor	SMS_G_System_NETWORK_ADAPTER.Manufacturer
	MAC Address	SMS_G_System_NETWORK_ADAPTER.MACAddress
	Description	SMS_G_System_NETWORK_ADAPTER.Description
	Adapter Type	SMS_G_System_NETWORK_ADAPTER.AdapterType
$System Devices$|$IDE Controllers$	Model	SMS_G_System_IDE_CONTROLLER.Name
	Vendor	SMS_G_System_IDE_CONTROLLER.Manufacturer
	Status	SMS_G_System_IDE_CONTROLLER.Status

AM Inventory Groups	AM Inventory Attributes	SCCM Inventory Attributes
$System Devices$|$USB Controllers$	Model	SMS_G_System_USB_CONTROLLER.Name
$Operating System$	Operating System	SMS_G_System_OPERATING_SYSTEM.Caption
	Service Pack	SMS_G_System_OPERATING_SYSTEM.CSDVersion
	Version	(based on) SMS_G_System_OPERATING_SYSTEM.Version
$Operating System$|$Installation$	Name	SMS_G_System_OPERATING_SYSTEM.Caption
	Vendor	SMS_G_System_OPERATING_SYSTEM.Manufacturer
	System Path	SMS_G_System_OPERATING_SYSTEM.WindowsDirectory
	Install Date	SMS_G_System_OPERATING_SYSTEM.InstallDate
	Version	(based on) SMS_G_System_OPERATING_SYSTEM.Version
	Major Version	(based on) SMS_G_System_OPERATING_SYSTEM.Version
	Minor Version	(based on) SMS_G_System_OPERATING_SYSTEM.Version
	Build Number	(based on) SMS_G_System_OPERATING_SYSTEM.Version
$File Systems$|$Local File Systems$	Name	SMS_G_System_LOGICAL_DISK.Name
	Size	SMS_G_System_LOGICAL_DISK.Size
	Used	Calculated from Size and Free
	Free	SMS_G_System_LOGICAL_DISK.FreeSpace

AM Inventory Groups	AM Inventory Attributes	SCCM Inventory Attributes
	Used %	Calculated from Size and Free
	Type	SMS_G_System_LOGICAL_DISK.DriveType
	File System	SMS_G_System_LOGICAL_DISK.FileSystem
	Volume Label	SMS_G_System_LOGICAL_DISK.VolumeName
	Mount Point	SMS_G_System_LOGICAL_DISK.Name
$Storage$	Total Disk Space	Calculated from fixed drives (SMS_G_SYSTEM_DISK) Size
$Storage$|$Fixed Drives$	Device	SMS_G_System_DISK.DeviceID
	Size	SMS_G_System_DISK.Size

More information:

CA Asset Converter for Microsoft SCCM (see page 396)

Mapping of Microsoft SCCM Software Asset Inventory to CA ITCM Asset Management Software Inventory

The following table delineates the software inventory data collected by the CA Asset Converter for Microsoft® SCCM and its corresponding attribute mapping in CA ITCM asset management.

AM Inventory Groups	AM Inventory Attributes	SCCM Inventory Attributes
		The software information is all based on instances of the WMI class SMS_G_System_ADD_REMOVE_PROGRAMS.
Discovered Software	Name	SMS_G_System_ADD_REMOVE_PROGRAMS.DisplayName
	Version	SMS_G_System_ADD_REMOVE_PROGRAMS.Version

AM Inventory Groups	AM Inventory Attributes	SCCM Inventory Attributes
	Publisher	SMS_G_System_ADD_REMOVE_PROGRAMS.Publisher

More information:

CA Asset Converter for Microsoft SCCM (see page 396)

Appendix G: Implementation of SCAP Standards

The Device Compliance Scanner (DCS) in CA ITCM scans target computers for compliance with the FDCC checklist. This chapter describes the implementation of SCAP standards.

SCAP

The DCS is built around Security Content Automation Protocol (SCAP). SCAP is a suite of selected open standards that enumerate software flaws, security-related configuration issues, and product names. The suite also measures systems to determine the presence of vulnerabilities, and provides mechanisms to rank (score) the results of these measurements to evaluate the impact of the discovered security issues.

CA ITCM implements compliance checking of any SCAP 1.1 data stream written in the XML formats leveraged by the SCAP standard: XCCDF, CCE, CVE, CPE, CVSS, and OVAL. DCS is implemented as an asset management inventory module. DCS is distributed to all the agents, which then performs the compliance check at the scheduled time and produces the output files required by the specifications. The DCS scanner uses the XCCDF and OVAL assessment protocols to determine what items to check and how to check them. The scanner also uses the CPE, CCE, CVSS, and CVE reference protocols to verify that all rules are accurately and appropriately reflected in the system. The DCS scanner reports the results to the central management database for inspection, reporting, and querying. The result files are generated for each file in the input SCAP data stream and are stored on the agent computer and domain manager (if configured) for verification.

CVE

The Common Vulnerabilities and Exposures (CVE) standard is a list or dictionary that provides standard identifiers for publicly known information security vulnerabilities and software flaws. The compliance check results produced by CA ITCM include the relevant CVE ID references in the output for every rule checked, provided such references are included in the checklist definition itself. The CVE information is stored in the patch result XML file generated by the scanner and is available in the agent's working directory for inspection and verification.

In SCAP data streams, OVAL content meant for the detection of applications, patches, or vulnerabilities can contain CVE ID references identifying the exact element in the CVE list. The FDCC checklists for Windows XP and IE7 contain separate OVAL files dedicated to this purpose and include CVE IDs.

When processing these SCAP data streams, the generated OVAL result files also include the CVE ID references for each OVAL definition. Additionally, the inventory data presented for the target computer in the DSM Explorer contain a Detailed patch results group where every OVAL definition meant for detecting patches or vulnerabilities has its own subgroup. This subgroup contains a CVE References table wherever the OVAL definition has such references defined in the SCAP data stream itself. Each CVE reference contains the CVE URL and NVD URL. The DSM Explorer allows browsing directly to these URLs.

CCE

Common Configuration Enumeration (CCE) provides unique identifiers to system configuration issues to facilitate fast and accurate correlation of configuration data across multiple information sources and tools. For example, you can use CCE Identifiers to associate checks in configuration assessment tools with statements in configuration best-practice documents.

In an SCAP data stream, references to CCE IDs can be present in either the XCCDF file or in OVAL files. In the XCCDF file, the CCE reference takes the form of <ident> tags listing CCE IDs associated with each rule in the list. If the CCE IDs are present in the XCCDF file, DCS includes these references for each rule result. This information is available both in the generated XCCDF result file and in the inventory data sent to the database. In the DSM Explorer, the CCE reference information is available under Inventory, SCAP, Checklist Name, Rule Results, Rule Name, Idents.

The OVAL files can contain CCE IDs associated with each OVAL definition that are contained in <reference> tags. If such references are present, they are included in the OVAL result files generated while processing the OVAL definitions.

CCE references and results are also available with the set of result files under the name <machine>-<checklist>-xccdf-CCE-result.txt.

All the FDCC checklists packaged with CA ITCM include CCE ID references both in the XCCDF files and the OVAL files. The name and location of the output files can be viewed from the DSM Explorer under the Inventory, SCAP, Checklist Inventory Component, Status group.

CPE

Common Platform Enumeration (CPE) is a structured naming scheme for information technology systems, platforms, and packages. Based upon the generic syntax for Uniform Resource Identifiers (URI), CPE includes a formal name format, a language for describing complex platforms, a method for checking names against a system, and a description format for binding text and tests to a name.

An SCAP data stream can optionally include a CPE dictionary that maps CPE names to OVAL definitions that test for the presence of the OS or application identified by that CPE name. DCS uses this dictionary when the XCCDF file from the data stream contains <*platform*> tags, which indicate that the XCCDF file requires the presence of the specified CPE name. All the packaged FDCC checklists contain CPE dictionary files and their reference in the XCCDF files. The XCCDF results files contain the CPE names in the <*platform*> tags to indicate a successful platform test for the entire checklist. The name and location of the output files can be viewed from the DSM Explorer under the Inventory, SCAP, Checklist Inventory Component, Status group.

CVSS

The Common Vulnerability Scoring System (CVSS) standard provides an open framework for communicating the characteristics and impacts of IT vulnerabilities. Its quantitative model helps ensure repeatable accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the scores. Thus, CVSS is well suited as a standard measurement system for industries, organizations, and governments that need accurate and consistent vulnerability impact scores. Two common uses of CVSS are prioritization of vulnerability remediation activities and calculation of the severity of vulnerabilities discovered on computers. The National Vulnerability Database (NVD) provides CVSS scores for almost all known vulnerabilities.

For every patch or vulnerability, CVE ID references are provided in the DSM Explorer. The DSM Explorer also provides detailed patch results that contain the CVE URL and the NVD URL. The user can use the URL to visit the National Institute of Standards and Technology (NIST) web page for the corresponding CVE ID's entry in the NVD. This database entry includes the CVSS score and additional information about the vulnerability. The CVE reference details are available under the Inventory, SCAP, Checklist Inventory Component, Detailed patch results group.

XCCDF

eXtensible Configuration Checklist Description Format (XCCDF) is a specification language for writing security checklists, benchmarks, and related kinds of documents. An XCCDF document represents a structured collection of security configuration rules for some set of target systems. The specification is designed to support information interchange, document generation, organizational and situational customization, automated compliance testing, and compliance scoring.

DCS reads the XCCDF file and scans the target computers based on the rules given in the XCCDF file. The scanner generates an inventory file that contains the results for each rule in the XCCDF file. The XCCDF output file is stored in the agent's working directory on each agent computer. The results for each rule and the final scores are displayed in the DSM Explorer under Inventory, SCAP, Checklist Inventory Component, Rule Results.

You can view the name and location of the XCCDF files and generated result files in the DSM Explorer under Inventory, SCAP, Checklist Inventory Component, Status group.

OVAL

CA ITCM implements the Open Vulnerability and Assessment Language (OVAL) standard. OVAL is an international, information security, community standard used to promote open and publicly available security content. Its goal is to standardize the transfer of this information across the entire spectrum of security tools and services.

Checklist rule definitions in XCCDF files typically use references to OVAL definitions in OVAL files as the way to indicate how to check a target computer for compliance with the rule. Similarly, CPE names listed in the CPE dictionary also use references to OVAL definitions to specify how to check for the presence of a piece of software indicated by the name. All of the bundled DCS SCAP data streams contain at least one OVAL file for each of these purposes.

For each evaluated OVAL file, the OVAL interpreter produces an OVAL results file in the agent's working directory. You can view the name and location of all the OVAL files and generated result files in the DSM Explorer under Inventory, SCAP, Checklist Inventory Component, Status group.

Glossary

application

An *application* is a piece of software, for example, Microsoft Word.

application virtualization

Application virtualization is the encapsulation of an application, separating it from the underlying operating system on which it is executed. At runtime the application is tricked into acting as if it were directly interfacing with the original operating system and all the resources managed by it, but in reality it is not.

centrally managed environment

A *centrally managed environment* is one where the remote control domain manager controls the host settings through computer policies, global address book (GAB) items, licensing of the host agent on the domain, and user permissions. This is the default setting for CA IT Client Manager.

centrally managed host environment

A *centrally managed host environment* is one where either a remote control enterprise or domain manager is responsible for the configuration of the hosts and the authentication of viewer connections. It also manages the address book that users use to find hosts.

Common Configuration Enumeration (CCE)

Common Configuration Enumeration (CCE) is one of the SCAP standards. It contains Standard identifiers and dictionary for system configuration issues related to security. A rule definition in an SCAP data stream can contain references to one or more CCE identifiers, indicating that the rule is a representation of a specific CCE configuration guidance statement or configuration control. For more information, go to http://cce.mitre.org/.

Common Platform Enumeration (CPE)

Common Platform Enumeration (CPE) is one of the SCAP standards. It contains standard identifiers and dictionary for platform or product naming. For example, some elements in XCCDF files can be restricted to only apply to certain platforms and this is done using CPE identifiers. For more information, go to http://cpe.mitre.org/.

Common Vulnerabilities and Exposures (CVE)

Common Vulnerabilities and Exposures (CVE) is a dictionary of common names (that is, CVE Identifiers) for publicly known information security vulnerabilities. These identifiers make it easier to share data across separate network security databases and tools. CVE is one of the components used in SCAP. See http://cve.mitre.org/ for details.

Common Vulnerability Scoring System (CVSS)

Common Vulnerability Scoring System (CVSS) is one of the SCAP standards. It contains standards for conveying and scoring the impact of vulnerabilities. For more information, go to http://www.first.org/cvss/index.html.

configuration view

A *configuration view* is a customized Windows-only user interface that lets you edit configuration policies that are related to specific components or functionality. Configuration views summarize the relevant policies for a component or function independent of where they are actually located in the hierarchy and the DSM Explorer tree.

connectors

connectors are the links from products that consume connector data to external products, or *domain managers*. Each connector retrieves information from its domain manager and transmits the information through the connector framework to the consuming product for visualization and analysis. Connectors can also enact inbound operations on data in the source domain manager, such as object creation. connectors use a unified connector framework to enable integration with multiple consuming products.

desktop recompose

Desktop recompose is the process of assigning a new golden template to the virtual desktop. Operating systems and applications have to be maintained during their lifetime to fix problems resolved by hot fixes or service packs or to provide new features by new versions. For linked clones, this means the master image, or golden template, has to be updated. Once the updates are completed, the linked clone is recomposed and becomes active. During the recompose operation the related linked clones are linked to this new golden template and are refreshed.

desktop refresh

Desktop refresh is the process of resetting the virtual desktop to its original state. Linked clones track changes to the virtual machine with the clone. To control the storage allocations with the clone, VMware View offers the refresh operation that resets the clone to its baseline and releases all deltas provided for tracking changes. This means that all information stored to the system drive since the creation of clone or its last refresh or recompose is lost. Unlike desktop recompose, the same golden template continues to be used as before the refresh operation.

eXtensible Configuration Checklist Description Format (XCCDF)

eXtensible Configuration Checklist Description Format (XCCDF) is a specification language for writing security checklists, benchmarks, and related kinds of documents. An XCCDF document represents a structured collection of security configuration rules for some set of target computers. The specification is designed to support information interchange, document generation, organizational and situational tailoring, automated compliance testing, and compliance scoring. For more information, go to http://nvd.nist.gov/xccdf.cfm.

Federal Information Processing Standard (FIPS)

Federal Information Processing Standard (FIPS) is a security standard that is issued and approved by NIST. It specifies the security requirements that must be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information.

FIPS-certified cryptography module

FIPS-certified cryptography module refer to RSA CryptoC BSAFE module, which is FIPS 140-2 certified.

FIPS-Compliant Cryptography

FIPS-compliant cryptography refers to the use of FIPS 140-2 certified modules, FIPS-approved, and FIPS-allowed techniques and algorithms for cryptography.

FIPS-only

FIPS-only is a mode of operation for CA ITCM wherein only FIPS-compliant cryptography is allowed. In this mode, CA ITCM is not backward compatible with the previous releases of CA ITCM.

FIPS-preferred

FIPS-preferred is a mode of operation for CA ITCM wherein bulk of cryptographic operations are FIPS-compliant, leaving few encryptions in legacy format. In this mode, CA ITCM is backward-compatible with the previous releases of CA ITCM.

golden template

In CA ITCM terminology, the *golden template* is the virtual machine from which virtual desktops are cloned.

guest

A *guest* in generic platform virtualization terminology is the virtual machine and the guest operating system.

guest operating system

The *guest operating system* is the operating system running inside a virtual machine.

health monitoring

Health Monitoring (HM) functionality lets you configure alerts, set threshold values, and monitor the overall health of the CA ITCM infrastructure.

host

A *host* in generic platform virtualization terminology is the physical machine, the host operating system, and the hypervisor.

host cluster

The *host cluster* is the aggregate computing and memory resources of a group of hosts sharing some or all of the same network and storage.

host operating system

The *host operating system* is the operating system running on a physical machine.

hosted virtual environment

A *hosted virtual environment* is the virtualization software that runs on top of a host operating system, that is, the physical machine, host OS, and the hypervisor.

hypervisor

The *hypervisor* is the virtualization software layer simulating physical hardware on behalf of the guest operating system. This term is synonymous with Virtual Machine Monitor (VMM).

instance software state database

The *instance software state database* is a part of the software state database that contains the history of all software jobs executed by the agent running on a non-golden template system, that is, any clones of the golden template.

linked clones

In VMware View, *linked clones* of a master or golden image only refer to the master or golden image but do not include it. Changes to the system during user sessions are not stored to the master image but are kept in delta files with the clone.

location awareness

Location Awareness lets DSM Agent on a computer detect the location of the computer.

master target device

In Citrix XenDesktop, a *master target device* is the base desktop with the OS and required set of applications from which a vDisk is generated.

master vDisk

In Citrix XenDesktop, a *master vDisk* is the initial vDisk generated from the golden template machine.

MITRE

The *MITRE Corporation* is a not-for-profit organization chartered to work in the public interest. MITRE offers the interpreters, source code, schemas, and data files at no cost so that individuals and organizations can build and expand upon them. Ovaldi is one such interpreter that is freely available.

National Institute of Standards and Technology (NIST)

National Institute of Standards and Technology (NIST) is a non-regulatory federal agency within the U.S. Department of Commerce. NIST's mission is to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life. The United States (U.S.) National Vulnerability Database (NVD), operated by the NIST, provides a repository and data feeds of content that utilize the SCAP standards. It is also the repository for certain official SCAP standards data. Thus, NIST defines open standards within the SCAP context and defines the mappings between the SCAP enumeration standards.

native virtual environment

A *native virtual environment* is the virtualization software that runs directly on the physical machine, becoming or acting as a host operating system (often minimal), that is, the physical machine and the hypervisor. A synonymous term is "bare metal environment."

non-linked clones

In VMware View, *non-linked clones,* or full clones, are full copies of a master or golden image. The clone includes a copy of the image and all changes to the system during user sessions are stored to this copy.

nonpersistent clones

Nonpersistent clones are virtual desktops from the nonpersistent pool of VMware View user data that are transient out-of-the-box. Once a user logs off, the clone is refreshed and all user data at the system disk are lost.

nonpersistent linked clone virtual desktop

A *nonpersistent linked clone virtual desktop* is a virtual machine that is refreshed or recomposed every time the user logs on, with no persistence for custom installed applications, personalization, and so on.

offline patching

Offline Patching lets you export the patch content and patch files remotely and import to the CA ITCM environment using CA Patch Manager without accessing Internet.

Offline RAC

Offline RAC is a reinstall after crash (RAC) task that is driven by the agent rather than by the manager. Virtual desktops are *recomposed* frequently, that is, whenever the golden template is updated and the disk is reset, any changes to the virtual desktop since the previous reset are effectively voided. For virtual desktops, the agent and not the manager is responsible for the creation of the RAC job container. When the disk reset occurs, the agent initiates an Offline RAC to restore any software that has been deployed to the agent.

Open Vulnerability and Assessment Language (OVAL)

Open Vulnerability and Assessment Language (OVAL) is one of the SCAP standards. It contains standard XML for testing procedures for security related software flaws, configuration issues, and patches as well as for reporting the results of the tests. All the rule checks in the checklists take the form of references to OVAL definitions contained in OVAL files from the SCAP data stream. For more information, go to http://oval.mitre.org/.

Ovaldi

Ovaldi is an OVAL Interpreter developed by the MITRE Corporation. It is a freely available reference implementation created to show how information can be collected from a computer for testing to evaluate and carry out the OVAL definitions for that platform, and to report the results of the tests. The interpreter demonstrates the usability of OVAL Definitions and ensures correct syntax and adherence to the OVAL Schemas.

package format

The *package format* is a property of a software package. Formats include regular and virtual.

package type

The *package type* is a property of a software package. Current types include Generic, MSI, SXP, PIF, and PKG. Package type is not used or altered for the purpose of supporting virtual application packages.

partition

A *partition* is an isolated instance of a host operating system. Partitions do not usually use guest operating systems because they all share the host's operating system.

partitioned virtual environment

A *partitioned virtual environment* is one where multiple instances of the host operating system can run in isolation on the same physical machine. This is not strictly a virtualization technology, but is used to solve the same type of problems.

persistent clones

Persistent clones are virtual desktops from the persistent pool that survive as they are after the user has logged off until they are refreshed or recomposed. VMware View offers out-of the box separate devices for system and user data with the persistent clones. Information stored to the user data device survives any refresh or recompose action while changes to the system disk are lost.

persistent linked clone virtual desktop

A *persistent linked clone virtual desktop* is a virtual machine that is dedicated to a specific user, and the user can request specific software to be added, customize settings, and so on. At each logon the user's customized environment is restored. This persists until the virtual desktop is refreshed or recomposed. At that point, all the software products installed on system drive are lost.

persistent non-linked clone virtual desktop

A *persistent non-linked clone virtual desktop* is a virtual machine that is dedicated to a specific user and is presented to that user at each logon with their custom installed applications, user settings, data, and so on.

platform virtualization

Platform virtualization is the encapsulation of computers or operating systems, hiding their physical characteristics from users and emulating the computing platform at runtime.

provisioned application

A *provisioned application* is an application (regular or virtual) that has been made available for execution on a target computer. The application need not be "installed" locally in order to treat it as provisioned.

regular application

A *regular application* is application software that has not been virtualized and can be installed and executed in a traditional fashion. When talking about releases, patches, and suites, regular applications are implied.

Replication

Replication is an engine task to perform the data replication from Domain Manager to Enterprise Manager and Enterprise Manager to Domain Manager.

sandbox

A sandbox is an application runtime environment that isolates the application from the computer's operating system and resources and also from other applications on the computer. The degree of isolation is usually set to allow the application some access to the operating system resources, such as the documents folder.

scalability server

A *scalability server* is the central server to enable geographical scalability for management tasks. It is a distributed process that is the primary interface for agents.

SCAP data stream

SCAP data stream consists of security checklist data represented in automated XML formats, vulnerability and product name related enumerations, and mappings between the enumerations. An SCAP data stream consists of the XML following files:

- An XCCDF file
- One or more OVAL files
- (Optional) A CPE dictionary file

schema map

A *schema map* is a mapping of the attribute names associated with data objects, such as users, computers, and groups, used in an external directory to those attribute names used by corresponding CA ITCM objects. The fixed and standard set of DSM attribute names is used for querying directories and for formulating complex queries and reports.

Security Content Automation Protocol (SCAP)

The *Security Content Automation Protocol (SCAP)*, pronounced "S Cap", is a method for using the standards such as XCCDF, CCE, CVE, CVSS, CPE, and OVAL to enable automated vulnerability management, measurement, and policy compliance evaluation (e.g., FISMA compliance). More specifically, SCAP is a suite of selected open standards that enumerate software flaws, security related configuration issues, and product names; measure systems to determine the presence of vulnerabilities; and provide mechanisms to rank (score) the results of these measurements in order to evaluate the impact of the discovered security issues. SCAP defines how these standards are combined. The National Vulnerability Database provides a repository and data feeds of content that use the SCAP standards. For more information, go to http://nvd.nist.gov/.

software signature

A *software signature* defines the attributes of a software application, such as the main executable file name, other associated files, size range, version range, creation, and modification dates of the software. All these attributes of a software signature uniquely identify a software application. Software signatures in asset management are created as software definitions. You can create software definitions for a product, release, patch, suite, suite component, or virtual application image. By default, asset management provides predefined software signatures covering the most widely used software in the IT industry.

software type

The *software type* is a property of a software definition. Current types include suite, product, release, patch, and virtual application image.

staged virtual application image

A *staged virtual application image* is a virtual application image that has been discovered in the file system of a computer.

stand-alone environment

A *stand-alone environment* is one where the users of the host and viewer computers locally manage all settings, properties, and licensing of the CA ITCM remote control component. It is set by a Standalone Agent installation. To install it manually, the RC agent setup needs to be called directly.

standalone virtual application

A *standalone virtual application* is a virtual application that has been provisioned in a way where the virtual application image used as the source resides on the system to which it has been provisioned.

streamed virtual application

A *streamed virtual application* is a virtual application that has been provisioned in a way where the virtual application image used as the source resides on a remote system that is different from the system to which it has been provisioned.

streamed virtual application image

A *streamed virtual application image* is a virtual application image that has been discovered to be accessible through the network from a computer. Discovery of streamed virtual application images will usually only be possible if the virtual applications residing inside of the image have been provisioned.

vDisk

In Citrix XenDesktop, a *vDisk*, or virtual disk, is basically an image file with the OS and the required set of applications.

virtual application (VA)

A *virtual applicatio*n is software that has been virtualized.

virtual application image

A *virtual application image* contains one or more virtual applications stored inside a file, possibly with a set of supporting metadata files.

virtual application image definition

A *virtual application image definition* describes the "footprint" for discovering a virtual application image. To discover an image containing one or more included virtual applications (stored inside), regular software signatures must be associated with the virtual application image definition.

virtual application package (VAP)

A virtual application image packaged inside of one or more software delivery packages is referred to as a v*irtual application package*. These packages are used to provision computers with virtual applications.

virtual application staging package

A *virtual application staging package* is a virtual application package used to stage the virtual application image.

virtual application standalone package

A *virtual application standalone package* is a virtual application package used to provision a virtual application in standalone mode.

virtual application streaming package

A *virtual application streaming package* is a virtual application package used to provision a virtual application in streaming mode.

virtual disk

A *virtual disk* is a set of files that forms a file system that appears as a physical disk to the guest operating system.

virtual image

A *virtual image* is a file or set of of files containing the complete definition of a virtual machine, including its hardware specifications and virtual disks. It is the host's file system representation of a guest. A virtual image can be online or offline depending on the running state of the virtual machine it captures.

virtual machine (VM)

A *virtual machine* is an isolated virtualized environment simulating a physical machine. The virtual machine does by definition not include the guest operating system.

virtual patch

A *virtual patch* is the virtual equivalent of a regular patch and has the same basic meaning. The term is used when reporting software inventory for virtual applications (not virtual application images).

virtual release

A *virtual release* is the virtual equivalent of the regular release and has the same basic meaning. The term is used when reporting software inventory for virtual applications (not virtual application images). Note that a provisioned virtual application can use either a staged or streamed virtual application image as source. The virtual applications contained within the virtual application image can themselves be seen as staged but not yet provisioned.

XCCDF profile

An *XCCDF profile* is a policy that is applied to the target computer or compared to the configuration of the target computer. The XCCDF file for each SCAP data stream defines the list of profiles supported. The XCCDF file must have at least one XCCDF profile, which specifies the rules to be used for checking a particular type of system. You can create separate XCCDF profiles for each applicable operational environment in which a system may be deployed.

Index

R

S

T

U

V

W