

SERVICENOW

CIS-HAM

The Implementation-First Guide

HARDWARE ASSET MANAGEMENT PROFESSIONAL



R. PATHAK

IT Operations Engineer · 10+ years in IT operations and IT asset management

UNOFFICIAL STUDY GUIDE · Not affiliated with or endorsed by ServiceNow, Inc.

Copyright and Legal Notice

ServiceNow CIS-HAM: The Implementation-First Guide to Hardware Asset Management

First edition, 2026.

Copyright (c) 2026 Rahul Pathak. All rights reserved.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the author, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law.

Published by the author. Written and bylined as R. Pathak.

Trademarks and Affiliation

ServiceNow, the ServiceNow logo, Now, Now Platform, and the names of ServiceNow products and services referenced in this book are the registered or unregistered trademarks of ServiceNow, Inc. in the United States and other countries. All other trademarks, product names, and company names mentioned are the property of their respective owners.

This book is an independent publication. It is **not** authorized, sponsored, endorsed, licensed, or otherwise affiliated with ServiceNow, Inc. References to ServiceNow products, certifications, and documentation are made for identification and educational purposes only, under nominative fair use. The author and this publication have no association with ServiceNow's certification program, its examination content, or its examination committee.

About the Certification

Certified Implementation Specialist - Hardware Asset Management (CIS-HAM) is a certification administered by ServiceNow, Inc. This book is a study and implementation guide written independently of that program. It does not contain, reproduce, or paraphrase live examination questions, and it is not an "exam dump." The practice questions in this book were written by the author to illustrate the reasoning patterns the exam objectives describe. Passing the practice questions in this book does not guarantee a passing result on the certification exam.

Exam format details, domain weightings, prerequisites, and fees change between ServiceNow release cycles. Always confirm current details against the official CIS-HAM exam blueprint on ServiceNow's learning portal before scheduling your exam.

Disclaimer

The information in this book is provided on an "as is" basis, for general educational and professional development purposes. While the author has made good-faith efforts to ensure the material is accurate and current at the time of writing, no warranty, express or implied, is made as to its completeness, accuracy, or fitness for any particular purpose.

Platform behavior changes. ServiceNow releases new platform versions on a regular cadence. Table names, field names, role names, module paths, and out-of-box behaviors described here reflect the author's understanding at the time of writing and may differ in your instance, in your ServiceNow release, or after future updates. Always verify configuration details against the official ServiceNow product documentation for your specific release, and test in a sub-production instance before making changes to production.

Not professional advice. Nothing in this book constitutes legal, financial, accounting, tax, or regulatory advice. Depreciation methods, data-retention periods, data-sanitization standards, and disposal-documentation requirements vary by jurisdiction, industry, and organization. Consult your own qualified finance, legal, and compliance advisors before adopting any practice described here.

Illustrative scenarios. The "Real-World Scenario" boxes throughout this book are composite, illustrative narratives assembled from patterns that recur across hardware asset management implementations. They are not reports of specific, verifiable incidents at named organizations, and the figures in them are illustrative rather than audited. They are included to show the shape of a problem, not to document a particular company's experience.

Diagrams. The conceptual diagrams in this book were produced with AI-assisted image generation tools and are stylized illustrations of concepts. They are not screenshots of the ServiceNow product and should not be read as depicting an actual product interface.

To the fullest extent permitted by law, the author disclaims any liability for loss or damage arising directly or indirectly from the use or application of any information in this book.

How to Use This Book

This is an implementation manual for ServiceNow Hardware Asset Management, written by a practitioner, and it doubles as preparation for the CIS-HAM certification. The order of those two things matters. Everything here is built around how a HAM deployment actually gets stood up and kept honest, because someone who understands why a deployment is configured a particular way can reason their way through an unfamiliar scenario, and is also useful on a real project the following Monday. Memorized answers do neither.

Who this is for

- Implementation consultants and architects preparing for the CIS-HAM certification
- IT asset managers and analysts moving from spreadsheets into a managed ServiceNow HAM deployment
- ITSM and platform administrators who own, or are inheriting, a hardware asset estate
- Anyone who already holds the ServiceNow Certified System Administrator credential and is choosing a specialization

Prior familiarity with core ServiceNow platform concepts - tables, records, roles, and flows - is assumed. Deep prior HAM experience is not.

How the book is organized

The five parts follow the order a real deployment happens in. You start by establishing why the thing exists and what the lifecycle looks like, because that governs every later decision. You then build a data foundation, because nothing downstream is trustworthy without it. Only then do you run day-to-day operations, put a usable interface in front of people, and finally make the financial numbers defensible. Chapter 3 explains why inverting that order is one of the most common ways a HAM project stalls.

Three recurring elements appear throughout:

- **Exam Tip boxes** identify a reasoning pattern the exam objectives target, and the wrong-answer trap that usually accompanies it
- **Real-World Scenario boxes** show the shape of a problem as it presents itself on an actual project, in composite, illustrative form
- **Check Your Understanding** questions close every chapter; the answer key with explanations is in Appendix B

A note on verification

Technical identifiers such as table and field names are given throughout to make the material usable on a real implementation. Treat them as a starting point for verification in your own instance rather than as a specification. Where this book and your instance disagree, your instance and the official ServiceNow documentation for your release are correct.

Contents

Front matter — Copyright and Legal Notice · Disclaimer · How to Use This Book

Part I — Why Hardware Asset Management Exists

- [Chapter 1: What Hardware Asset Management Actually Solves](#)
- [Chapter 2: The Hardware Asset Lifecycle, End to End](#)
- [Chapter 3: Personas, Roles, and Implementation Best Practices](#)

Part II — Building a Data Foundation You Can Trust

- [Chapter 4: Assets vs. Configuration Items](#)
- [Chapter 5: Hardware Discovery and Trustworthy Data](#)
- [Chapter 6: The Content Service and Hardware Model Normalization](#)
- [Chapter 7: Normalization Statuses and Manual Correction](#)

Part III — Running the Estate Day to Day

- [Chapter 8: Procurement and Sourcing in HAM](#)
- [Chapter 9: Stockrooms, Stock Rules, and Inventory Operations](#)
- [Chapter 10: Asset Tasks, Flows, and Automation](#)
- [Chapter 11: Consumables, Contracts, and Retirement/Disposal](#)

Part IV — Putting It in Front of People

- [Chapter 12: The Hardware Asset Workspace](#)
- [Chapter 13: Requests, Procurement Integration, and Data Certification](#)
- [Chapter 14: Reporting and Dashboards for Hardware Assets](#)

Part V — Making the Numbers Defensible

- [Chapter 15: Hardware Financials — Rate Cards, Depreciation, and TCO](#)

Appendices

- [Appendix A: The HAM Data Model, and Where It Goes Wrong](#)
- [Appendix B: Answer Key and Explanations](#)
- [Appendix C: Verifying This Material Against Official Sources](#)
- [Appendix D: Preparing for the Exam](#)

PART I — WHY HARDWARE ASSET MANAGEMENT EXISTS

Chapter 1: What Hardware Asset Management Actually Solves

Every organization that issues a laptop, a monitor, or a phone to an employee is already doing hardware asset management. The question is whether they're doing it on purpose. Most aren't. Spreadsheets go stale the day after they're updated. Nobody can say with confidence how many laptops are sitting unassigned in a stockroom, which ones are still under warranty, or which ones should have been wiped and disposed of eighteen months ago. Hardware Asset Management (HAM) in ServiceNow exists to close that gap: to give an organization a single, trustworthy, continuously updated record of every physical IT asset it owns, from the moment it's ordered to the moment it's destroyed or resold.

This matters for reasons that go well beyond tidiness. Audit and compliance teams need to prove what hardware exists and where it is. Security teams need to know which devices are unaccounted for, because an unaccounted-for laptop is a potential data breach. Finance needs accurate depreciation schedules and needs to stop paying maintenance contracts on hardware that was decommissioned two years ago. Procurement needs to know what's already sitting in a stockroom before ordering more. None of this works if the underlying data is unreliable, and unreliable hardware data is the default state for most organizations before a real ITAM implementation. In ServiceNow, this system of record lives on the `alm_hardware` table, and everything in this book eventually traces back to keeping that table honest.

REAL-WORLD SCENARIO: *A global bank failed a regulatory audit when internal auditors discovered roughly 450 unassigned laptops sitting untracked in a regional stockroom while procurement was simultaneously issuing purchase orders for 300 new ones. The root cause was regional Excel workarounds updated manually once a quarter. The fine and remediation cost ran into seven figures. The fix wasn't exotic: pre-created "On Order" asset records tied to the purchase order, barcode-scanned dock receiving, and real-time minimum/maximum stock rules, but none of it existed before the incident forced the issue.*

Why HAM Is Its Own Application

It would be reasonable to ask why hardware assets need a dedicated application when the CMDB already tracks configuration items. The answer is that a CMDB record and an asset record answer different questions. A CI answers "what is this thing and what does it do in the environment": its function, its relationships, its role in a service. An asset record answers "who owns this, what did it cost, what state is it in commercially, and what should happen to it next." A laptop can be a CI the moment it's discovered on the network, but it's an asset from the moment a purchase order is cut for it: long before it has any relationship to a business service. HAM is built to manage that full

commercial and physical lifecycle, independent of whether the item is currently represented in the CMDB at all.

The Business Drivers, in Practice

In real implementations, the drivers that get a HAM project funded tend to cluster around a handful of recurring pains:

- Software and hardware audits that turn into weeks-long fire drills because nobody can produce accurate counts on demand
- Warranty and support contracts being paid on assets that were already retired
- New hires waiting days for equipment because nobody knew what was actually available in stock
- Security incidents involving devices that had left the building without anyone noticing
- Finance unable to reconcile the fixed-asset register against what IT says it actually has

A CIS-HAM implementer's job is to design a system that eliminates these specific pains, not to install HAM as a generic best practice. Every configuration decision in the chapters that follow should be traceable back to one of these business outcomes.

How HAM Fits Alongside SAM and the Wider ITAM Suite

ServiceNow's IT Asset Management suite splits hardware and software into separate applications, HAM and Software Asset Management (SAM), because the two disciplines diverge quickly once you get past the shared fundamentals. Hardware has physical location, condition, and depreciation, tracked on `alm_hardware`. Software has license entitlements, consumption, and compliance position, tracked on tables like `samp_license_entitlement`. Both plug into the same underlying Asset and CI data model and the same procurement and contract structures, which is why an implementer who's already comfortable with SAM will recognize a lot of the scaffolding here. The differences show up in the domain-specific processes: stockrooms and physical movement for HAM, entitlement reconciliation for SAM.

EXAM TIP: A recurring exam pattern is a scenario that describes a business pain (a failed audit, an under-utilized stockroom, an unreconciled fixed-asset register) and asks which HAM capability addresses it. Learn the pains in this chapter as a checklist, not as background reading: several questions map a scenario directly onto one of these drivers.

Key Takeaways

- HAM exists to make hardware data trustworthy enough to drive audit, security, and financial decisions, not just to have an inventory list.

- An asset record and a CI record answer different questions; HAM manages the commercial and physical lifecycle, independent of a CI relationship.
- Real implementations are funded by specific, nameable pains — know them, because exam scenarios are built around them.
- HAM and SAM share a data foundation but diverge in domain-specific process: physical movement and depreciation versus entitlement and license compliance.

Check Your Understanding

1.1 An organization discovers it is still paying vendor maintenance fees on a rack of servers that were physically decommissioned two years ago. Which HAM capability most directly prevents this?

- **A.** Contract records linked to asset records, with retirement triggering a contract review
- **B.** Hardware model normalization against the Content Service
- **C.** Minimum-quantity stock rules on the stockroom holding the servers
- **D.** A quarterly data certification campaign sent to end users

1.2 Which statement best explains why Hardware Asset Management exists as its own application rather than living entirely inside the CMDB?

- **A.** The CMDB cannot store serial numbers or model information
- **B.** HAM is intended to replace the CMDB for organizations that own physical hardware
- **C.** Asset records answer commercial and lifecycle questions, while CI records answer operational and relationship questions
- **D.** The CMDB is reserved for software, while HAM covers everything physical

1.3 Which of the following is NOT a typical business driver used to justify funding a HAM implementation?

- **A.** Audits that become multi-week manual reconciliation exercises
- **B.** Standardizing which scripting language the platform team uses for customizations
- **C.** New hires waiting days for equipment that was already sitting in stock
- **D.** Devices leaving the organization without a documented data-sanitization record

Answers and explanations for these questions are in Appendix B.

Chapter 2: The Hardware Asset Lifecycle, End to End

Everything in HAM configuration exists to support one thing: moving an asset correctly through its lifecycle, and always being able to say, accurately, which stage it's in right now. Get the lifecycle model right first, because every stockroom rule, every task, every report in later chapters is built on top of it.



Figure 2.1: The ServiceNow hardware lifecycle — On Order through Retired/Disposed

The Lifecycle Stages

ServiceNow models the hardware lifecycle on the asset record's `install_status` field (the top-level state), with a companion substatus field carrying the finer-grained detail. In practice, implementers work with a lifecycle that looks like this:

Top-level state (<code>install_status</code>)	Substate (<code>substatus</code>)	Operational definition
On Order	N/A	PO created; asset pre-logged; item not yet received at the dock
In Stock	Available, Pending Deployment, Reserved	Received into a stockroom; available for allocation or staged for a request
Assigned / In Use	N/A	Allocated to an active user, department, or location
In Transit	Pending Transfer, In Route	Moving between stockrooms via a tracked Transfer Order
In Repair	Pending Repair, Out for Service	Sent to an internal repair desk or vendor for service
Retired	Pending Disposal, Disposed, Sold	Permanently off-ramped,

	Donated	sanitized, and certified
--	---------	--------------------------

The critical implementation discipline is that every transition between these stages should happen through a defined process, a task, a flow, a stockroom transaction. It should never happen through someone manually editing `install_status` or `substatus` directly on the form. The moment manual edits become normal, the lifecycle data stops being trustworthy, and trustworthy data is the entire point of the application.

REAL-WORLD SCENARIO: *A hospital system saw its receiving dock become an invisible bottleneck. Hardware sat unboxed for weeks before anyone logged it, so the asset system showed a clean picture that had nothing to do with reality. Adding a specific In Stock substate for "received but not yet deployed," paired with barcode scanning at the point of unloading, meant dock staff updated the record in the same motion as unloading the container — no separate data-entry step to skip under pressure.*

Designing State Transitions Around Real Operations

A common implementation mistake is copying the out-of-box state model without asking whether it matches how the organization actually moves hardware. If your stockroom team physically stages assets for a week before deployment, "in stock" needs a substate that captures "staged, not yet available" or your in-stock counts will be wrong every time someone checks. If assets frequently move between regional stockrooms before final assignment, "in transit" needs to be a first-class, trackable state, not an assumption that transfers happen instantly.

Before configuring anything, walk the physical process with the stockroom and deployment teams. Ask what actually happens to a laptop between the loading dock and a new hire's desk, and where records currently break down. The state model you build should make that specific breakdown impossible, or at least immediately visible.

Ownership Changes and the Asset Record

Lifecycle state and ownership are related but distinct. An asset can change state (in stock → assigned) without changing legal or financial ownership, and it can change custodial ownership (reassigned from one employee to another) without ever leaving the "assigned" state. HAM implementations need to track both: the operational state (what stage of its life the asset is in) and the assignment (who currently has it, and who is accountable for it). Confusing these two concepts is one of the most common design errors in early implementations, and it shows up later as reports that can't distinguish "nobody has this laptop right now" from "we have no idea who has this laptop."

Retirement Is a Process, Not an Event

Retirement deserves special attention because it's the stage implementers most often under-design. A retired asset isn't simply deleted or hidden: it needs a documented disposal method, evidence of data wipe or destruction (for compliance and security), and often a certificate of destruction from a third-party vendor attached to the record. Skipping this design work is what produces the audit finding "we cannot demonstrate that decommissioned hardware containing sensitive data was properly destroyed": one of the most damaging findings a security or compliance audit can produce.

EXAM TIP: *Expect scenario questions that describe an asset moving between two states and ask which substate or transaction correctly represents it. Particularly around the difference between a stockroom transfer, a reassignment, and a retirement. Read carefully for whether the scenario is describing a change in physical location, a change in custodian, or a permanent removal from service; each has a distinct, correct answer.*

Key Takeaways

- The lifecycle is a defined sequence of states: on order, in stock, assigned, in transit, in repair, retired, and every transition should happen through a process, never a manual field edit.
- State models should be designed around how the organization actually moves hardware, not copied out of the box.
- Lifecycle state and ownership/assignment are distinct concepts and need to be tracked separately.
- Retirement requires a documented, auditable disposal process — it's a compliance control, not a cleanup task.

Check Your Understanding

2.1 Which field on the hardware asset record carries the top-level lifecycle state?

- **A.** operational_status
- **B.** substatus
- **C.** asset_state
- **D.** install_status

2.2 A laptop is being shipped from the Chicago stockroom to the Austin stockroom, and will be in transit for three days. Which approach correctly represents this?

- **A.** Edit the stockroom field on the asset record directly to Austin
- **B.** Retire the Chicago record and create a new asset record in Austin
- **C.** Raise a transfer order, which moves the asset into the In Transit state until it is received
- **D.** Leave the record unchanged until the asset physically arrives in Austin

2.3 An employee changes roles and hands their assigned laptop to the colleague taking over. The device stays deployed and in daily use. What should change on the asset record?

- **A.** assigned_to changes to the new custodian, while install_status remains Assigned / In Use
- **B.** install_status moves to In Transit, and assigned_to is cleared
- **C.** install_status moves to In Stock until the new custodian is confirmed
- **D.** Nothing changes, because the asset never physically left the building

Answers and explanations for these questions are in Appendix B.

This is a free sample

You've reached the end of the free sample. It covers the front matter and the first two chapters, so you can judge the writing, the depth, and whether the implementation-first approach fits how you learn.

The full guide continues with thirteen more chapters across all five areas of the discipline: the asset and configuration item boundary, discovery, and the full model normalization pipeline; procurement, stockrooms, stock rules, transfer orders, asset tasks, and compliant disposal; the Hardware Asset Workspace, requests, data certification, and stakeholder reporting; and rate cards, expense lines, depreciation, and total cost of ownership. It closes with forty-five review questions and worked answers, an appendix on the HAM data model and what implementers get wrong about each table, and a chapter-by-chapter exam-preparation plan.

The complete book is available on this page in PDF and EPUB, DRM-free.

Independent and unofficial. Not affiliated with, endorsed by, or sponsored by ServiceNow, Inc.