PINMAN

How Pinman Works

A practical model for understanding, comparing, and managing the configuration of a Windows workstation.

TRACK | UNDERSTAND | CONTROL

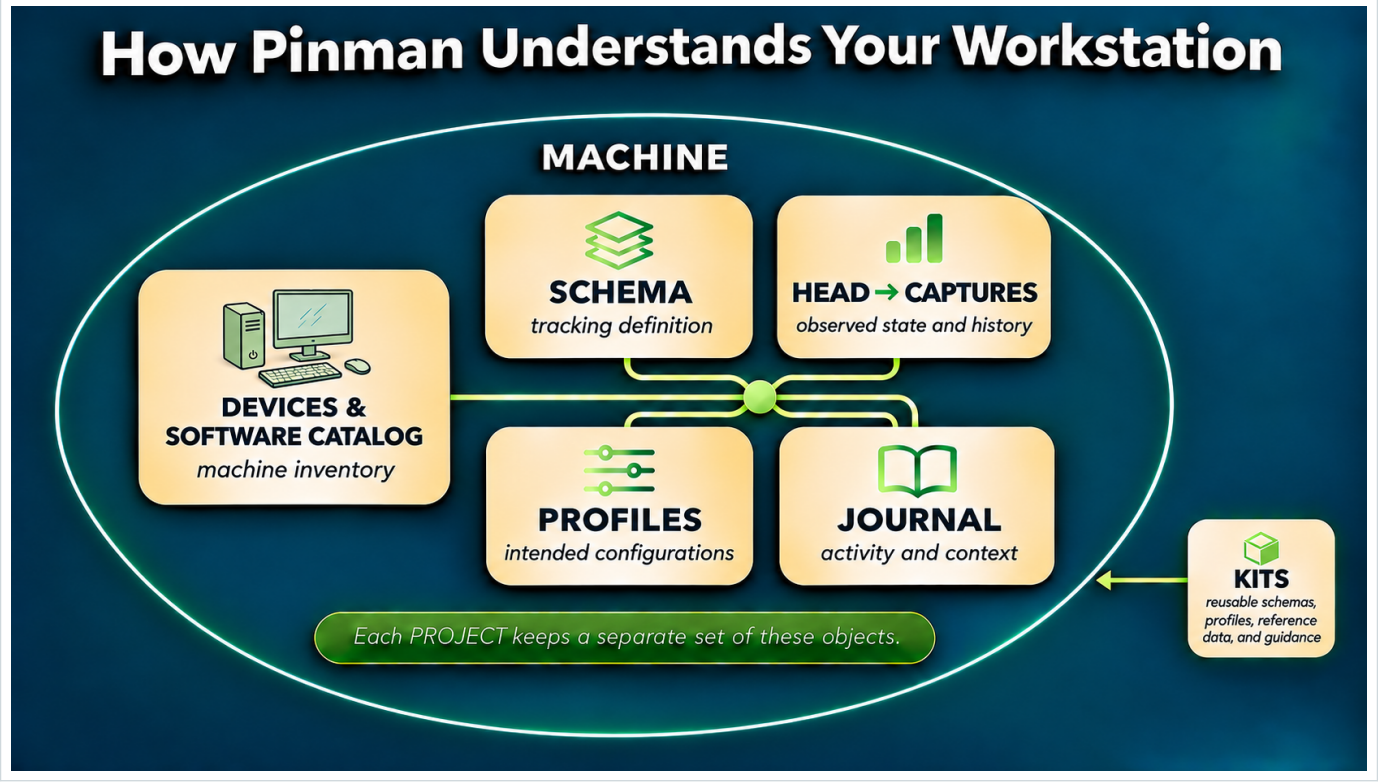
THE CORE IDEA

Pinman brings together configuration scattered across Windows, applications, and connected hardware, so you can see what changed, describe what you want, and make controlled changes.

THE BIG PICTURE

One model for a scattered system

Windows spreads workstation configuration across files, the Registry, system settings, devices, services, and software-specific tools. Pinman brings the parts you care about together.



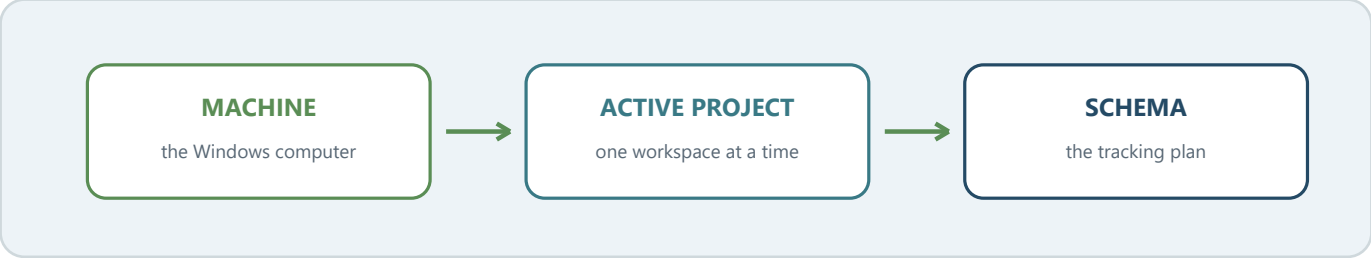
A Project keeps a separate Schema, Head and Captures, Profiles, Journal, and machine inventory. Kits provide reusable content for that model.

Schema Defines what Pinman tracks.	**Head + Captures** Record what Pinman observes.	**Profiles** Describe configurations you want.
Journal Connects activity with its context.	**Inventory** Describes devices and software.	

DEFINE THE WORK

Machine, Project, and Schema

Pinman separates the computer itself, the workspace you are using, and the plan that says which configuration matters.



Machine and Project

A **Machine** is the Windows computer Pinman manages. Your work on it is organized in a **Project**. Each Project has its own Schema, Captures, Profiles, Journal, device model, Software Catalog, and related data.

A Machine can have several Projects, with one active at a time. Switching Projects changes the Pinman workspace; it does not switch computers or change the Machine. Separate Projects are useful for tutorials, experiments, and distinct bodies of configuration work.

Schema

The **Schema** is the tracking plan. Its tree identifies exact configuration locations and controlled patterns for groups of similar configuration.

Source	What it can identify
`fs:`	Directories, files, and settings inside supported file formats
`winreg:`	Windows Registry keys and values
`winrt:`	Display, audio, device, and other Windows runtime settings

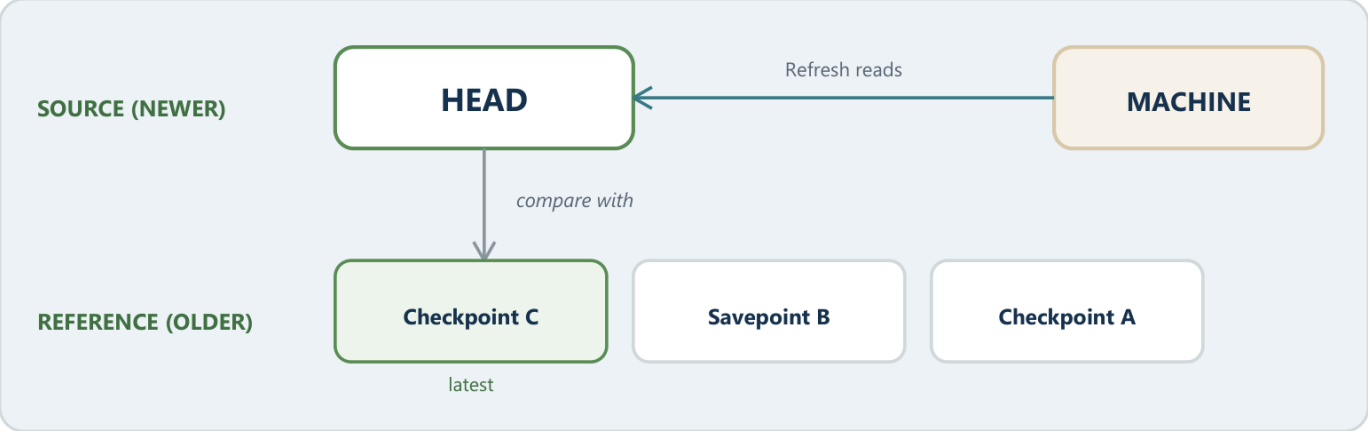
> The Schema defines what Pinman tracks. It does not contain observed values. Editing it changes the tracking plan without reading or changing the Machine.

OBSERVE AND CONTROL

Head, Captures, and Profiles

Pinman distinguishes its current working record, frozen history, and the configurations you intend to use.

The **Head** is Pinman's current understanding of the Machine. **Refresh** reads all or part of the selected configuration and updates the Head. Different Items can have different observation times, so the UI shows freshness where it matters.



CAPTURES	PROFILES
A **Capture** freezes recorded state at a point in time. Compare it with the Head or another Capture to understand change. A Checkpoint keeps Item records; a Savepoint also backs up applicable files.	A **Profile** describes a configuration you want to use. Check it against the Head, preview the changes, and Apply supported values. Pinman records previous values so supported changes can be undone.

Turn on read-only mode when you want to inspect without making changes. You can still Refresh, compare Captures, check Profiles, and preview what an Apply would do.

INTERPRET THE WORK

Change, intent, and context

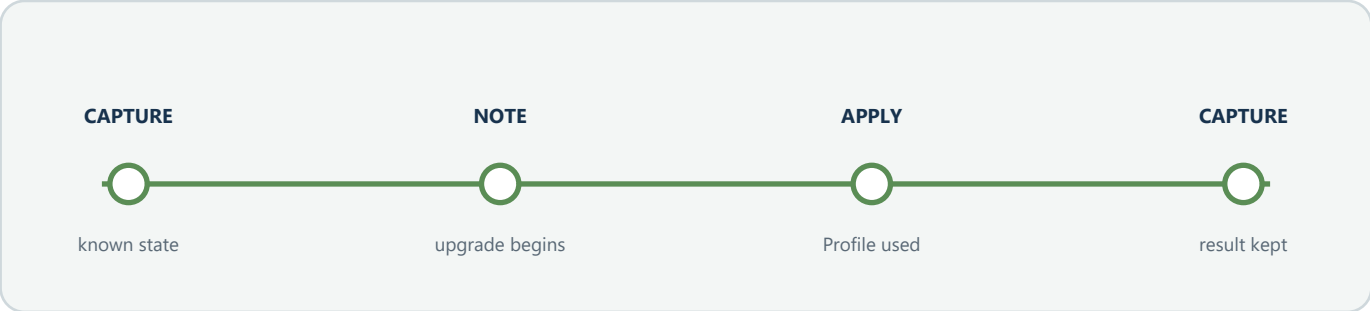
Pinman separates three related questions: what changed, whether the current record matches your intent, and what was happening at the time.

CAPTURE COMPARISON	PROFILE DRIFT
What changed since an earlier point? Choose a newer Source and an older Reference. Most comparisons use the Head and a Capture, but two Captures can also be compared.	**Does the Head match this Profile?** The Profile's targets define the intended configuration. Drift identifies current values that would need to change to match it.

Comparing recorded state does not Refresh the Head. A drift check also uses the Head as Pinman currently knows it. Refresh first when you need the comparison or drift result to reflect the Machine now.

Journal

The **Journal** connects technical records with a timeline. Pinman adds entries for important activity such as Captures and Profile applies, and you can add notes about the work surrounding them.



> The Journal turns configuration evidence into a useful story: an upgrade, a troubleshooting session, a deliberate settings change, or an unexpected problem.

ADD STABLE CONTEXT

Devices, software, and reusable Kits

Configuration is easier to understand when it is connected to the hardware and software it belongs to.

Model	What it contributes
Device	Windows identifiers, properties, presence, and a stable record of connected hardware
Box	A physical machine or section of hardware containing assigned Devices
Role	A purpose such as “presentation display” that a Profile can use instead of a device identifier
Software Catalog	Stable identities for known applications, drivers, utilities, services, runtimes, and Windows tools

A replacement Device can take over the same role, allowing a Profile to keep working without being rewritten. Software Catalog entries can connect tracked Items to the software that owns or uses them even when its configuration is spread across several Windows sources.

Detection adds machine-specific software state such as presence, version, and installation path. It checks known catalog entries; the Schema still determines what Pinman tracks.

Kits

> A **Kit** packages reusable Pinman content for a kind of Machine, software, or workflow. It can include a Schema, Software Catalog definitions, Profiles, reference data, and documentation released together under one version.

Install a Kit to add its content to the Kit library, then import it into a Project instead of building everything from scratch. Projects using the same Kit still keep their own Captures and history.

- Start from a maintained tracking model instead of an empty Schema.
- Share Profiles and reference material across a team or community.
- Update related content together under an explicit Kit version.

PUT IT TOGETHER

A practical Pinman workflow

You can begin with a small amount of tracked configuration and add richer models as the work calls for them.

1. **Choose the model**
   Start a Project and choose or build its Schema, possibly from a Kit.
2. **Keep a reference**
   Create a Checkpoint or Savepoint for a useful known state.
3. **Let work happen**
   Configure, update, test, troubleshoot, and use the Machine.
4. **Read and compare**
   Refresh the Head and compare it with the Reference you choose.
5. **Describe intent**
   Create Profiles for configurations you want to check or reuse.
6. **Keep the story**
   Use Captures, applies, and Journal notes to understand the result.

Start small. Track a few settings and compare Captures. Add Profiles, hardware roles, Software Catalog entries, or Kits when they solve a problem you actually have.

Continue exploring

Understanding Captures	pinmantech.com/docs/concepts/captures
What Pinman can change	pinmantech.com/docs/concepts/what-pinman-changes
The Basics tutorial	pinmantech.com/docs/tutorial/basics
Installing Kits	pinmantech.com/docs/getting-started/kits

Read the living documentation at pinmantech.com/docs/how-it-works.