

CSDM 5.0 Baseline Assessment Report

100X Technology Inc Demo02 · Prepared by Echelon · 2026-04-24

Instance	https://100xtechnologyincdemo02.service-now.com
Version	Zurich (glide-zurich-07-01-2025__patch7)
Scope	Full CSDM 5.0 Baseline — 12 Maturity Dimensions
Overall Rating	Walk with critical Crawl gaps in 3 dimensions

Executive Summary

This instance is solidly in the **Walk** stage across most CSDM 5.0 maturity dimensions, with pockets of Crawl-level gaps that create disproportionate risk. The CMDB tooling suite is well-installed (CMDB Workspace v7.5.6, Get-Well Dashboards, Data Foundation Model), but the tooling has not been operationalized: **zero principal classes are flagged, zero recommended fields are configured, and zero `managed_by_group` entries are set** on any of the 794 CI class info records.

The most critical gap is **Freshness**: 3,024 of 3,416 health results (88.5%) are Staleness violations, driven by 1,767 software package CIs not re-discovered in over 60 days — root-caused to a single unvalidated MID Server.

Quick wins: validate the MID Server, flag principal classes in CI Class Manager, and configure the first Data Manager Certification policy for Business Applications.

Maturity Dimensions

1. [Correctness](#)

2. [Completeness](#)

3. [Compliance](#)

4. [Freshness](#)

5. [Ownership & Governance](#)

6. [Lifecycle Management](#)

7. [Source Authority & Reconciliation](#)

8. [Integration Hygiene](#)

9. [Relationship Modeling](#)

10. [Operational Adoption](#)

11. [Attestation & Certification](#)

12. [Workspace & Tooling Adoption](#)

[Data Manager Best Practices](#)

[Appendix: Installed Tooling](#)

[Priority Remediation Roadmap](#)

Overall Maturity Vector

#	Dimension	Rating	Evidence Highlight
1	Correctness	Walk	321 duplicate flags (10%+ of non-spkg CIs); zero orphan rules configured
2	Completeness	Walk	16 required-field violations; zero recommended fields defined
3	Compliance	Walk	14 cert_audit records active; 3 scheduled, rest On Demand with stale last-run dates
4	Freshness	Crawl	3,024 staleness violations; single OOB 60-day rule; only 11 CIs have last_discovered set
5	Ownership & Governance	Crawl	0 of 794 cmdb_class_info records have managed_by_group set; 0 principal classes flagged
6	Lifecycle Management	Walk	CSDM migration NOT activated; 86/3,189 CIs have life_cycle_stage ; BA data partially populated
7	Source Authority & Reconciliation	Walk	2 active reconciliation definitions; 0 multisource records; 0 sys_object_source entries
8	Integration Hygiene	Crawl	1 MID Server (unvalidated, no status); Integration Commons installed; no SGC Central
9	Relationship Modeling	Walk	572 relationship records; 1,439 suggested rel rules (OOB); CSDM core service relationships need verification
10	Operational Adoption	Walk	Incidents: 39/132 (30%) CI-bound; Changes: 78/150 (52%) CI-bound; Service-binding: 12/132 (9%)
11	Attestation & Certification	Crawl	0 certification policies configured; 0 sn_cmdb_ws_dm_certification_policy records
12	Workspace & Tooling Adoption	Walk	CMDB Workspace v7.5.6 installed; Get-Well Dashboards installed; no Now Assist for CMDB; no SGC Central

CSDM Domain Population

CSDM Domain	Primary Table	Record Count	Notes
Business Applications	cmdb_ci_business_app	86	80/86 have owned_by ; 0/86 have managed_by_group ; all 86 have life_cycle_stage
Business Capabilities	cmdb_ci_business_capability	156	Present in health results; not in CSDM v5 core layers
Business Services	cmdb_ci_service_business	28	Operational status populated; owned_by populated on sampled records
Service Offerings	service_offering	54	Populated via OOB catalog integration
Application Services / Service Instances	cmdb_ci_service_auto	0	Zero records — critical CSDM gap
Discovered Services	cmdb_ci_service_discovered	0	Zero records — Discovery not populating services
Technical Services	cmdb_ci_service_technical	0	Zero records — upper-tier CSDM mapping absent
Legacy Services	cmdb_ci_service	44	Pre-CSDM service records not yet migrated to typed tables
Servers	cmdb_ci_server	41 (direct)	858 total computers; 816 are cmdb_ci_computer class
Computers	cmdb_ci_computer	858	Largest infrastructure class

CSDM Domain	Primary Table	Record Count	Notes
Software Packages	<code>cmdb_ci_spkg</code>	1,767	Dominant CI class; major staleness issue
Network Gear	<code>cmdb_ci_netgear</code>	5	<code>cmdb_ci_network_gear</code> table not present; <code>cmdb_ci_netgear</code> used
IP Routers/Switches	<code>cmdb_ci_ip_router</code> + <code>ip_switch</code>	4	Minimal network CI population
Total CMDB CI footprint	<code>cmdb_ci</code> (all classes)	3,189	—

Service layer gap: `cmdb_ci_service_auto`, `cmdb_ci_service_discovered`, and `cmdb_ci_service_technical` all return 0 records. The 44 `cmdb_ci_service` records are pre-CSDM "Service" class entries that should be reclassified into typed service classes. This is the most significant CSDM structural gap in this instance.

1. Correctness Walk

EVIDENCE

- `cmdb_health_result` metric=Duplicate: **321 records** flagged as duplicates across the CMDB
- `cmdb_health_result` metric=Staleness: **3,024 records** (overwhelmingly software packages)
- `cmdb_health_orphan_rule` : **0 records** — no orphan rules configured
- `cmdb_health_result` total: 3,416 health results across 4 active metric types (Staleness, Duplicate, Audit, Required)
- Duplicate flags are concentrated in: `cmdb_ci_computer` (1,107 health entries total across all metrics), `cmdb_ci_spkg` (1,767 staleness entries), `cmdb_ci_service` (75 entries)
- Health metric "Duplicate" (`sys_id` `e0172b...`) produced 321 violation records

RATING RATIONALE

321 duplicate records represent ~10% of the non-spkg CMDB footprint ($3,189 - 1,767 = 1,422$ non-spkg CIs). This places the instance at the boundary between Crawl (>10%) and Walk (<5%). The absence of any orphan rules means orphan detection is entirely disabled. The OOB Duplicate metric is active and producing results, which earns a Walk floor, but the orphan gap prevents advancing to Run.

TOP RECOMMENDATIONS

- 1 Configure orphan rules on principal CSDM classes** (`cmdb_ci_business_app` , `cmdb_ci_service_business` , `cmdb_ci_computer`): Navigate to CMDB Health → Orphan Rules and create at minimum a "no relationship" rule for Business Applications and Servers. This surfaces CIs that exist in the CMDB with no upstream or downstream relationship, which is a data quality proxy for ghost records.
- 2 Remediate the 321 duplicate flags:** Use the CMDB Workspace (`sn_cmdb_ws` , already installed) to review and merge or tombstone duplicate CI records — prioritize `cmdb_ci_computer` and `cmdb_ci_server` classes where duplicates have operational impact.

2. Completeness Walk

EVIDENCE

- `cmdb_recommended_fields` : **0 records** — no recommended fields defined for any CI class
- `cmdb_health_result` metric=Required: **16 violation records** — some required fields are tracked and failing
- `cmdb_health_result` metric=Recommended: Not present in active health results (0 records) — no recommended field scoring is running
- Required field violations span: `cmdb_ci_business_app` (101 health entries total), `cmdb_ci_computer` (1,107 entries), `cmdb_ci_service` (75 entries)
- `sys_dictionary` `mandatory=true` on `cmdb_ci-hierarchy` fields is the source of the 16 required-field violations

RATING RATIONALE

The instance has the OOB required-field tracking running (producing 16 violations), which meets the Walk floor. However, zero recommended fields have been defined, meaning completeness scoring does not extend beyond what ServiceNow mandates by default. No class has a curated recommended field list, preventing the "Recommended fields" health metric from producing any results.

TOP RECOMMENDATIONS

- 1 Define recommended fields for Business Applications first:** In CI Class Manager (accessible via CMDB Workspace), select `cmdb_ci_business_app` and configure a recommended field set: at minimum `owned_by` , `short_description` , `install_status` , `managed_by_group` . This immediately produces completeness scoring visible in the Get-Well Dashboards.
- 2 Resolve the 16 required-field violations:** Query `cmdb_health_result` where metric = Required and review which specific CIs have missing mandatory fields. Many are likely stale discovery records where fields were not populated on initial write.

3. Compliance Walk

EVIDENCE

- `cert_audit` : **14 records, all active=true**
- Scheduled cadence breakdown:
 - **Daily:** 2 records (UPS Audit 7-day, UPS Audit 3-day) — last run 2026-04-24 (current)
 - **Weekly:** 1 record (Critical Business Services on `cmdb_ci_service`) — last run 2026-04-20
 - **Monthly:** 2 records (Data Center Zone Consumption on `cmdb_ci_zone` , Database Audit on `cmdb_ci_database`) — last run 2026-04-03 and 2026-04-15
 - **On Demand:** 9 records — including Server Minimum Statistics (last run 2013-04-08), Windows Computer Minimum Statistics (last run 2013-04-08), Unix Server Baseline (last run 2013-04-08)
- `cmdb_health_result` `metric=Audit`: **55 violations flagged**
- `cmdb_data_management_policy` (CMDB Data Manager Policies): 3 records — all are Dependent CI policies (Retire, Deletion, Archival), none are Certification policies
- `sn_cmdb_ws_dm_certification_policy` : 0 records — no CMDB Workspace certification policies

RATING RATIONALE

The presence of 14 active `cert_audit` records with 5 operating on automated schedules meets the Walk threshold. However, 9 of 14 audits are On Demand only and have stale last-run dates back to 2013–2015, making them dormant in practice. Coverage is biased toward infrastructure (UPS, servers, computers) and largely absent for CSDM service-layer classes (no audit on `cmdb_ci_business_app` , `cmdb_ci_service_auto` , etc.).

TOP RECOMMENDATIONS

- 1 Add `cert_audit` records for CSDM core classes:** Create compliance audits for `cmdb_ci_business_app` and `cmdb_ci_service_business` on at least a monthly schedule. The Critical Business Services audit on `cmdb_ci_service` exists but targets the legacy base class — duplicate it for the typed CSDM service classes.
- 2 Convert dormant On-Demand audits to scheduled runs:** Update the 9 On-Demand audits (Server Minimum Statistics, Windows Computer Minimum Statistics, Unix Server Baseline, etc.) to run at least monthly. Their 2013 last-run dates indicate they are effectively inactive.

4. Freshness Crawl

EVIDENCE

- `cmdb_health_staleness_rule` : **1 record** — the OOB base rule: `applies_to=cmdb_ci` , `duration=60 Days`
- `cmdb_health_result` `metric=Staleness`: **3,024 violation records** — 88.5% of all health results
- Staleness violations by class: `cmdb_ci_spkg` = 1,767 (58.5% of all violations), `cmdb_ci_computer` = dominant remainder
- CIs with `last_discovered` populated: **11 of 3,189 (0.3%)**
- `cmdb_datasource_staleness` table: not queried separately; MID Server is unvalidated
- Lifecycle properties: `csdm.lifecycle.migration.activated` = `false` , `csdm.lifecycle.bulk_population.active` = `false`
- MID Server: 1 record, status empty, `validated=false`

RATING RATIONALE

A single OOB 60-day staleness rule with no per-class tuning, combined with 3,024 violations and only 11 CIs having any `last_discovered` value, places this firmly in **Crawl**. The Walk threshold requires at minimum the OOB rule to be actively respected (CIs being discovered and refreshed within 60 days), which is not happening given the volume of violations. The unvalidated MID Server is a root-cause indicator: Discovery is not running effectively.

TOP RECOMMENDATIONS

- 1 Validate and activate the MID Server:** The single MID Server ("Test MID") has no status and is not validated. Navigate to MID Server → Validate and complete the validation process. Without a working MID Server, Discovery cannot refresh CIs and staleness violations will continue to grow.
- 2 Create per-class staleness rules for operational classes:** After the MID Server is operational, add class-specific staleness rules for `cmdb_ci_computer` (30-day), `cmdb_ci_server` (30-day), and `cmdb_ci_business_app` (90-day via manual attestation cycle). The 1,767 software package staleness entries should be addressed by either running software discovery or removing the `cmdb_ci_spkg` class from the staleness rule scope if spkg inventory is not actively maintained.

5. Ownership & Governance Crawl

EVIDENCE

- `cmdb_class_info` total records: **794** (matches expected Zurich class count)
- `cmdb_class_info` where `managed_by_group` is populated: **0 of 794 (0%)**
- `cmdb_class_info` where `principal_class` = `true` : **0 of 794 (0%)**
- `cmdb_ci_business_app` where `owned_by` populated: 80 of 86 (93%) — individual CI-level ownership exists
- `cmdb_ci_business_app` where `managed_by_group` populated: 0 of 86 (0%)

- `cmdb_ci_service_business` : Sample shows `owned_by` populated (Roger Seid on sampled records)

RATING RATIONALE

Zero class-level `managed_by_group` assignments and zero principal class flags are the defining evidence for **Crawl** on this dimension. While individual BA records have `owned_by` populated (93%), this is CI-level ownership by individual users — not the class-level group governance that the CSDM maturity model measures. No TSO cascade is possible without `managed_by_group` on `cmdb_class_info`. The absence of principal class flags means the CMDB Workspace dashboards are also operating without class governance context.

TOP RECOMMENDATIONS

- 1 **Flag principal classes on `cmdb_class_info`** : Using CMDB Workspace → CI Class Manager, flag the following as principal classes: `cmdb_ci_business_app`, `cmdb_ci_service_business`, `cmdb_ci_computer`, `cmdb_ci_server`, `cmdb_ci_linux_server`, `cmdb_ci_win_server`. This is a configuration-only change with no data risk.
- 2 **Assign `managed_by_group` on principal class records**: After flagging principal classes, assign the appropriate ownership group to each. For example, set the IT department group on `cmdb_ci_computer` and `cmdb_ci_server`, and the Application team group on `cmdb_ci_business_app`. This unlocks TSO cascade to all CIs in those classes within 24 hours per the CSDM Data Sync scheduled job.

HOW `MANAGED_BY_GROUP` TASK ROUTING WORKS

Assigning Data Manager tasks to a CI's `managed_by_group` is the **correct and intended mechanism** — it is how ServiceNow designed the routing to work, not a custom workaround.

- When a policy run generates tasks (e.g., attestation tasks), those tasks are assigned to the group in the **`managed_by_group` field on the CI itself** — not a separate group defined on the policy.
- If the policy has **"Needs Review" enabled**, the assigned group receives email notifications and must approve before the subflow continues to dependent CIs.
- If **"Needs Review" is disabled**, tasks execute without manual intervention (fully automated mode).
- In CMDB Workspace's Home view, users see **only** the tasks for CIs where they are members of the Managed by Group — this is intentional and supports distributed ownership.

Critical prerequisite: `managed_by_group` must be populated on the target CIs. CIs with an empty `managed_by_group` will generate tasks with no assignee — this is the most common reason attestation tasks fall through the cracks and go unactioned on this instance (0 of 794 class info records are set).

6. Lifecycle Management

Walk

EVIDENCE

- `csdm.lifecycle.migration.activated` : **false** — CSDM lifecycle migration not activated
- `csdm.lifecycle.bulk_population.active` : **false** — bulk population not in progress
- `csdm.lifecycle.sync.between.ci.and.asset.activated` : **false** — CI↔Asset lifecycle sync disabled
- `cmdb_ci` where `life_cycle_stage` is populated: **86 of 3,189 (2.7%)**
- `cmdb_ci_business_app` where `life_cycle_stage` populated: **86 of 86 (100%)** — all BAs have lifecycle stage
- Sample lifecycle stages on BAs: Operational, Deploy, Design, End of Life — multi-stage vocabulary in use
- `cmdb_data_management_policy` : 3 policies — Dependent CI Retire, Dependent CI Deletion, Dependent CI Archival (all on `cmdb_ci` table)
- Legacy `install_status` field: still the primary lifecycle field for non-BA classes

RATING RATIONALE

The instance earns a Walk rating because CSDM lifecycle fields are in active use on Business Applications (100% coverage) and the three Dependent CI Data Manager policies represent a working Retire/Delete/Archive workflow. However, the migration is not activated at the platform level (both activation properties are `false`), meaning `life_cycle_stage` is not propagating to infrastructure classes. Only 86 of 3,189 CIs (2.7%) have `life_cycle_stage` — these are all Business Applications. All other classes remain on the legacy `install_status`-only model.

TOP RECOMMENDATIONS

- 1 **Activate CSDM lifecycle migration:** Set `csdm.lifecycle.migration.activated = true` in System Properties. This enables the CSDM lifecycle field mapping and unlocks the bulk population job. Review the pre-activation checklist (backup/snapshot recommended) before toggling, as this initiates a platform-wide data transformation.
- 2 **Add Data Manager Retire policies for principal infrastructure classes:** Currently, the 3 DM policies all target the base `cmdb_ci` table. Create class-specific Retire policies for `cmdb_ci_computer` and `cmdb_ci_server` that trigger the Dependent CI cascade when those classes reach End of Life stage, rather than relying on the broad `cmdb_ci` catch-all.

7. Source Authority & Reconciliation

Walk

EVIDENCE

- `cmdb_reconciliation_definition` : **2 active records**
 - "Computer rule by CredentiallessDiscovery" (source: CredentiallessDiscovery)
 - "Cloud Gateway rule by ServiceWatch" (source: ServiceWatch)
- `cmdb_multisource_data` : **0 records** — multisource CMDB 360 not in use
- `sys_object_source` : **0 records** — no source registrations
- `cmdb_datasource_staleness` : not populated (staleness by datasource not configured)
- Only 2 discovery sources are registered with reconciliation rules; all other potential sources (ServiceNow Discovery, manual imports, SGC connectors) have no explicit reconciliation definitions

RATING RATIONALE

Two active reconciliation definitions cover two specific discovery sources (CredentiallessDiscovery and ServiceWatch), satisfying the Walk threshold of "static rules per class." However, the coverage is minimal — only 2 of potentially many integration sources have reconciliation rules. The absence of `cmdb_multisource_data` records confirms that Multisource CMDB 360 (the Run/Fly capability) has not been activated. The 0 `sys_object_source` records indicate that source registration, which underpins IRE's multi-source resolution, is not configured.

TOP RECOMMENDATIONS

- 1 Register all active integration sources in `sys_object_source`** : For each data source writing to the CMDB (Discovery, ServiceWatch, manual CSV imports, any ServiceGraph Connectors), create a `sys_object_source` registration record. This is the prerequisite for IRE to apply correct precedence rules during reconciliation.
- 2 Add reconciliation definitions for the primary Discovery source:** The current rules cover CredentiallessDiscovery and ServiceWatch but there is no rule for the standard ServiceNow Discovery source. Add a reconciliation definition for `discovery_source=ServiceNow` on at minimum `cmdb_ci_computer` and `cmdb_ci_server` to prevent Discovery writes from overwriting authoritative data from other sources without precedence logic.

8. Integration Hygiene Crawl

EVIDENCE

- `ecc_agent` (MID Servers): **1 record** — name="Test MID", status=empty, `validated=false`
- `sys_store_app` for Integration Commons: `sn_cmdb_int_util` v2.22.0 installed (Integration Commons for CMDB)
- No SGC Central (`sn_sgc_central`) installed

- No CMDB-specific SGC connector apps found (no `sn_cmdb_int_*` scoped apps beyond `sn_cmdb_int_util`)
- No Now Assist for CMDB (`sn_cmdb_gen_ai`) installed
- MID Server is not validated — Discovery cannot execute probes or process ECC queue
- Only 11 CIs have `last_discovered` populated, confirming Discovery is not actively running

RATING RATIONALE

A single unvalidated MID Server with no active status, combined with zero SGC connectors and no evidence of Discovery-sourced CI updates (only 11 `last_discovered` values), places this at **Crawl**. The Walk threshold requires IRE to be used for Discovery at minimum, but Discovery itself is not operational. Integration Commons for CMDB is installed (v2.22.0) as a prerequisite framework, but no connectors are consuming it.

TOP RECOMMENDATIONS

- 1 Validate the MID Server and run a Discovery test:** Navigate to MID Server → "Test MID" and complete the validation workflow. Then configure at least one IP range in Discovery Schedules and run a quick discovery against a known subnet. This is the foundational prerequisite for all integration hygiene improvement.
- 2 Evaluate SGC Central for integration governance:** Install `sn_sgc_central` (Service Graph Connector Central) from the ServiceNow Store to gain a governance dashboard over all CMDB integration sources. Given that Integration Commons is already installed (v2.22.0), SGC Central is the natural next step to manage source authority and connector health visibility.

9. Relationship Modeling Walk

EVIDENCE

- `cmdb_rel_ci` (actual CI-to-CI relationships): **572 records total**
- `cmdb_rel_type_suggest` (suggested relationship rules): **1,439 records** — OOB suggested relationships covering all major CI classes
- Sample suggested relationships include: infrastructure hosting chains (VM → Host), containment (Namespace → Cluster), networking (UPS → Powers → Rack/Server), and cloud hierarchies (AWS ECS → ECS Service)
- `cmdb_ci_service_auto` : 0 records — no Application Service maps populated; BA↔SI↔TSO CSDM relationship chain cannot exist
- `cmdb_ci_service_technical` : 0 records — no Technical Service Offerings mapped
- `cmdb_ci_business_app` relationship coverage: 101 health results for BAs (including some relationship metrics), 86 BA records with ~1.2 health results each suggesting partial relationship coverage

- `cmdb_rel_ci` count of 572 against 3,189 CIs = **0.18 relationships per CI average — very sparse**

RATING RATIONALE

The 1,439 OOB suggested relationship rules satisfy Walk-level relationship modeling configuration. However, the 572 actual relationship records for 3,189 CIs (0.18 relationships/CI) is extremely sparse, and the complete absence of `cmdb_ci_service_auto` and `cmdb_ci_service_technical` means the core CSDM relationship chain (Business Application → Service Instance → Technical Service Offering) cannot be modeled. This gap prevents advancing beyond Walk.

TOP RECOMMENDATIONS

- 1 Populate `cmdb_ci_service_auto` (Application Services) as the bridge layer:** CSDM v5 requires Service Instances to connect Business Applications to infrastructure. Create Application Service records in `cmdb_ci_service_auto` for the top 10 Business Applications by business criticality, then manually establish the BA → runs on → Application Service and Application Service → uses → Server/Computer relationships. This unlocks impact analysis from business to infrastructure.
- 2 Run Relationship Compliance Health checks:** The `cmdb_health_metric` table contains `Rel_hosting`, `Rel_containment`, and `Rel_suggested` metrics. Execute these health metrics from CMDB Health → Health Metrics to understand which existing 572 relationships violate the 1,439 suggested relationship rules, and prioritize cleanup of those violations.

10. Operational Adoption Walk

EVIDENCE

- **Incidents:** 39 of 132 total = **29.5% CI-bound** (`cmdb_ci` field populated)
- **Changes:** 78 of 150 total = **52.0% CI-bound** (`cmdb_ci` field populated)
- **Service-binding** (incidents with `business_service` populated): 12 of 132 = **9.1%**
- Combined task-CI binding rate (incidents + changes): 117 of 282 = 41.5%
- *Note: This is a demo/dev instance with low ticket volumes (132 incidents, 150 changes total); binding rates on production instances may differ significantly.*

RATING RATIONALE

At 30% incident CI-binding and 52% change CI-binding, the instance sits comfortably in the Walk band (>20% incidents, principal classes enforced). The low service-binding rate (9.1%) is more concerning — this means almost no incidents are associated with a Business Service, which limits CMDB-driven service health dashboards. The change CI-binding at 52% is the strongest signal, suggesting ITSM process discipline exists for changes.

TOP RECOMMENDATIONS

1 Drive incident service-binding above 30%: The current 9.1% service-binding rate is below Walk threshold expectations. Configure a mandatory field or UI policy on the Incident form to prompt agents to populate `business_service` when a CI is selected. The `cmdb_ci_service_business` records (28 records) provide the lookup pool — ensure incident forms auto-suggest the business service associated with the selected CI.

2 Enforce CI selection on all incidents via Process/UI policy: 70.5% of incidents lack a CI reference. Implement a soft enforcement (advisory message) using a catalog client script or business rule that prompts the agent if an incident is being resolved without a CI linked. Avoid hard mandatory fields initially to minimize process friction; move to mandatory after 60-day adoption measurement.

11. Attestation & Certification Crawl

EVIDENCE

- `sn_cmdb_ws_dm_certification_policy` (CMDB Workspace Certification Policies): **0 records**
- `cmdb_data_management_policy` total: 3 records — all are Dependent CI type (Retire, Delete, Archive), none are Certification type
- `cert_audit` (legacy certification audits): 14 active, but these are field-level data audits, not CSDM ownership attestation workflows
- No certification schedules (`sn_cmdb_ws_dm_certification_attribute_status`): table exists (as child of `cmdb_data_management_policy`) but is empty
- CMDB Workspace v7.5.6 is installed, which includes the Data Manager Certification UI — it is simply unconfigured

DATA MANAGER POLICY TYPES

Data Manager provides six policy types out of the box. These are **not pre-configured templates** — you create instances of each type scoped to specific CI classes.

Policy Type	Purpose	When to Use
Attestation	Verify a CI still exists and its data is accurate	Highest-value starting point for most orgs
Certification	Certify specific attribute values are correct	After attestation is established
Retire	Transition CIs to a retired lifecycle state	Requires a retirement definition per CI class
Archive	Move CI and related non-CMDB data to archive	After retirement, for data reduction

Policy Type	Purpose	When to Use
Delete	Permanently remove CIs	End-of-life, after archival
Delete CMDB Related Entries	Clean up records in related tables	Housekeeping alongside delete

***Lifecycle chain:** Retire → Archive → Delete is the recommended sequential lifecycle chain — do not jump straight to delete. Each of Retire / Archive / Delete requires an **active retirement definition** for the targeted CI class. The base `cmdb_ci` class ships with a retirement definition by default and cannot be deleted.*

RECOMMENDED ROLLOUT ORDER (CRAWL → WALK → RUN)

Do not enable all policy types for all CI classes at once. ServiceNow's best practices guidance prescribes a scoped, phased approach:

Phase	Scope	Policy Types
Crawl	1–2 high-value CI classes (e.g., servers, laptops)	Attestation policies only
Walk	Expand attestation; add more CI classes	Add Certification policies
Run	Full CI class coverage	Retire / Archive / Delete lifecycle policies

Attestation is **non-destructive** and **human-reviewed**. It forces the organization to confront data quality issues before any lifecycle actions are taken. Starting with Delete or Retire policies before attestation is in place is the most common mistake — it leads to accidental CI removal and eroded trust in the CMDB.

Begin with one critical service or application to prove value and refine your process before onboarding every CI class. — [Get Started with Data Manager](#)

RATING RATIONALE

Zero certification policies configured is a clear **Crawl** rating. The legacy `cert_audit` records (14) provide some data-quality audit coverage but do not constitute attestation in the CSDM sense — attestation requires CI owners to periodically confirm the accuracy of CI data, typically via Data Manager Certification workflows. The installed CMDB Workspace provides the UI for certification, but it has not been activated.

TOP RECOMMENDATIONS

- 1 Create the first Data Manager Certification policy for Business Applications:** In CMDB Workspace → Data Manager, create a Certification policy targeting `cmbd_ci_business_app` with a quarterly schedule. Assign to the group responsible for application portfolio management. This is the highest-value certification target given that BAs are the CSDM anchor class and already have 93% individual ownership (`owned_by`) populated.
- 2 Define certification scope and attribute list before launch:** Before activating the certification policy, use the CMDB Workspace to select which fields will be attested (recommended: `name` , `owned_by` , `install_status` , `life_cycle_stage` , `managed_by_group`). Starting with 5–6 fields per class prevents attestor fatigue and establishes a repeatable pattern for expanding to other classes.

12. Workspace & Tooling Adoption Walk

EVIDENCE

- CMDB Workspace (`sn_cmbd_ws`): **v7.5.6** — installed and active
- CMDB and CSDM Data Foundations Dashboards (`sn_getwell`): **v4.1.2** — installed and active
- Data Foundation Model (`sn_cmbd_foundation`): **v1.7.2** — installed and active
- CMDB CI Class Models (`sn_cmbd_ci_class`): **v1.83.0** — installed and active
- CMDB Agent Workspace (`sn_cmbd_workspace`): **v1.0.0** — installed and active
- Integration Commons for CMDB (`sn_cmbd_int_util`): **v2.22.0** — installed and active
- CMDB Page Templates (`sn_cmbd_pg_tmplts`): **v3.2.4** — installed and active
- CMDB NLQ Search (`sn_cmbd_nlq_search`): **v2.3.3** — installed and active
- SGC Central (`sn_sgc_central`): **NOT installed**
- Now Assist for CMDB (`sn_cmbd_gen_ai`): **NOT installed** (Now Assist for ITSM is installed, `sn_itsm_gen_ai` v12.2.7)
- CMDB Success Advisor (`sn_cmbd_advisor`): **NOT installed**
- Now Assist platform apps: Now Assist Core v28.7.12 installed; broad Now Assist footprint exists for ITSM, Voice, Code, etc. — but not for CMDB specifically

RATING RATIONALE

With CMDB Workspace, Get-Well Dashboards, Data Foundation Model, CI Class Models, and Integration Commons all installed and active, this instance has a strong Walk-level tooling foundation. The key gaps preventing Run/Fly are: no SGC Central (integration governance), no Now Assist for CMDB (AI-assisted enrichment), and no CMDB Success Advisor (prescribed outcome tracking). The tooling is installed but its operational usage is limited — principal classes are not flagged, recommended fields are not configured, and the workspace capabilities for certification and data management remain unconfigured.

TOP RECOMMENDATIONS

- 1 Operationalize CMDB Workspace by configuring principal classes and recommended fields:** The Workspace is installed (v7.5.6) but the configuration that powers its dashboards and CI Class Manager views is absent (0 principal classes, 0 recommended fields). These configurations take less than an hour and immediately surface health scoring in the Get-Well Dashboards.
- 2 Install SGC Central and evaluate Now Assist for CMDB:** `sn_sgc_central` enables governance over all integration sources feeding the CMDB — critical given the unvalidated MID Server and absent reconciliation coverage. `sn_cmdb_gen_ai` (Now Assist for CMDB) is the path to Fly-level tooling and can be trialed in the existing Zurich instance given that Now Assist Core (v28.7.12) is already installed.

Appendix: Installed CMDB Tooling Suite

Application	Scope	Version	Status	Notes
CMDB Workspace	<code>sn_cmdb_ws</code>	7.5.6	Active	Core workspace — installed but unconfigured (no principal classes)
CMDB and CSDM Data Foundations Dashboards (Get-Well)	<code>sn_getwell</code>	4.1.2	Active	Health dashboards installed; limited value without recommended fields
Data Foundation Model	<code>sn_cmdb_foundation</code>	1.7.2	Active	Foundation schema models installed
CMDB CI Class Models	<code>sn_cmdb_ci_class</code>	1.83.0	Active	Class model definitions; current version
CMDB Agent Workspace	<code>sn_cmdb_workspace</code>	1.0.0	Active	Agent-facing CMDB workspace
Integration Commons for CMDB	<code>sn_cmdb_int_util</code>	2.22.0	Active	Prerequisite for SGC connectors; no connectors installed yet

Application	Scope	Version	Status	Notes
CMDB Page Templates	sn_cmdb_pg_tmplts	3.2.4	Active	UI page templates for CMDB views
CMDB NLQ Search	sn_cmdb_nlq_search	2.3.3	Active	Natural language query for CMDB
SGC Central	sn_sgc_central	—	NOT installed	Integration governance dashboard — recommended next install
Now Assist for CMDB	sn_cmdb_gen_ai	—	NOT installed	AI enrichment — Now Assist Core (v28.7.12) is present as prerequisite
CMDB Success Advisor	sn_cmdb_advisor	—	NOT installed	Prescribed outcome tracking
MID Server ("Test MID")	—	—	Unvalidated, no status	Root cause of Freshness/Discovery failure

Data Manager: Prerequisites & Best Practices

Before scheduling or running any Data Manager policy, the following prerequisites must be completed. Skipping these steps is the primary cause of policy runs generating unassigned tasks, failing silently, or producing unintended CI state changes.

Prerequisites Checklist

1. **Populate `managed_by_group`** on all CIs you plan to target — this drives task assignment and email routing. On this instance, 0 of 794 `cmdb_class_info` records have this set (see [Dimension 5](#)).
2. **Create retirement definitions** for each CI class targeted by Retire, Archive, or Delete policies. The base `cmdb_ci` class definition ships OOB but class-specific definitions do not.
3. **Configure condition filters** on each policy to scope which CIs are included per run — never run against all CIs globally on first execution.
4. **Test in non-production** before enabling scheduled policy runs in production.

5. **Assign CMDB governance roles** (`cmdb_admin` , `cmdb_manager`) to the teams responsible for reviewing and approving policy tasks.

Quick-Reference: Best Practices Summary

Principle	Guidance
First policy type to implement	Attestation
Task assignment mechanism	<code>managed_by_group</code> field on the CI
Lifecycle action order	Retire → Archive → Delete (never skip)
Retirement definition required	Yes, for Retire / Archive / Delete policies
Rollout approach	Crawl (1–2 CI classes) → Walk → Run
"Needs Review" setting	Enable for human approval; disable for full automation
Most common failure point	<code>managed_by_group</code> unpopulated on target CIs

Reference: [Working with CMDB Data Manager](#) / [Create a Policy](#) / [Retirement Definitions](#)

Priority Remediation Roadmap

Based on evidence gathered, the following sequenced actions would advance the most dimensions simultaneously:

Priority	Action	Dimensions Advanced	Effort
1	Validate MID Server and run Discovery	Freshness, Integration Hygiene, Correctness	Low — configuration only
2	Flag principal classes on <code>cmdb_class_info</code> (6 core classes)	Ownership, Completeness, Workspace Adoption	Very Low — UI toggle
3	Set <code>managed_by_group</code> on principal class <code>cmdb_class_info</code> records	Ownership, Governance	Low — 6 records to update

Priority	Action	Dimensions Advanced	Effort
4	Configure recommended fields for <code>cmdb_ci_business_app</code>	Completeness, Workspace Adoption	Low — CI Class Manager UI
5	Activate CSDM lifecycle migration (with pre-activation snapshot)	Lifecycle Management	Medium — property toggle + validation
6	Create first Data Manager Certification policy (Business Applications, quarterly)	Attestation & Certification	Medium — policy configuration
7	Create <code>cmdb_ci_service_auto</code> records for top Business Applications	Relationship Modeling	Medium — data modeling work
8	Add orphan rules for <code>cmdb_ci_business_app</code> and <code>cmdb_ci_server</code>	Correctness	Low — 2-3 rule records
9	Register integration sources in <code>sys_object_source</code> and add reconciliation definitions	Source Authority	Medium — requires source inventory
10	Install SGC Central and evaluate Now Assist for CMDB	Workspace Adoption, Integration Hygiene	Medium — store app installs