Configuration Management Database:

Improving Configuration Item Data Quality

serviceNow®
INTRODUCTION
Within the IT industry, a Configuration Management Database (CMDB) is the central repository of information describing all of the IT infrastructure components used to deliver services to your business. Your configuration management capability should serve to ensure the as-built solutions meet their functional, operational, and physical requirements, the correct versions/configurations are used and match their documentation.

A Configuration Item, or CI, is the fundamental structural management unit of the CMDB and it serves as the basis for the language for effective communication within your IT community. Everything IT supports is ultimately expressed in terms of CIs: incidents are expressed in terms of degraded CI(s), problems are expressed in terms of which CIs are root causes and are impacted, changes are expressed in terms of which CIs are changing or are potentially impact / at risk for a specific change, etc.

Thus, the quality of CI data will be an essential tenet of your ability to communicate effectively. As you work to define your CIs, you can initially expect to capture CI data from your current data sources. Most organizations find their current data may or may not be clear, concise, or complete – it is almost always of inconsistent quality. However, at some point during their initial implementation of a CMDB, most organizations decide to load their CMDB with the best data available and embark on an ongoing effort to steadily improve data quality over time. The Configuration Management team should be relentless in challenging the IT organization to improve CI data quality.

This white paper outlines a best practice progression of phases to address and improve CI data quality in your CMDB. The processes around how the CIs are created, maintained, managed, and eventually removed from the CMDB are also described in this white paper to illustrate common practices. This paper also references the most common use cases for creating a CMDB.
DATA QUALITY OF CONFIGURATION ITEM DATA

As you build your CMDB, there are some distinct phases a typical organization passes through on the journey to improve CI data quality. The table below provides a high level description of each phase.

### Service Aware CMDB

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Define</td>
<td>Identify the types of configuration items that an organization needs for a successful deployment of use cases served by the CMDB. Common standard types include desktops, servers, network devices, software components, business applications, and more.</td>
</tr>
<tr>
<td>2. Create/Discover</td>
<td>Insert the individual instances of the configuration items as records into the CMDB. This may involve processes such as manual entry, automated discovery, or integration of data from external sources.</td>
</tr>
<tr>
<td>3. Configure/Enrich</td>
<td>Normalize and update any individual attributes of configuration items that do not conform to the standard format set by the organization.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>4. Assess/Map</strong></td>
<td>Add relationships. Each configuration item exists in the infrastructure as a physical item or the CI represents an abstract component in the organization such as a business service. Thus, the CI must have relationships that are identified either to other sibling CIs or mapped to one or more of the organization's business CIs.</td>
</tr>
<tr>
<td><strong>5. Reconciliation</strong></td>
<td>The CMDB manager must designate a single source of truth for each attribute of each CI type and assign these data sources accordingly. It is not uncommon for information that populates configuration items to be gathered by multiple data sources. These data sources often provide information for the same attributes for the same CIs. Reconciliation prevents attribute values from being overwritten by unauthorized sources.</td>
</tr>
<tr>
<td><strong>6. Data Certification</strong></td>
<td>Implement an independent certification process on either all, or a selected set of records to ensure quality control. Even after configuration item instances are inserted into the CMDB properly according to all the automated and manual processes, flaws in the processes might result in inaccurate data.</td>
</tr>
<tr>
<td><strong>7. Monitor</strong></td>
<td>Now that accurate and fully certified CIs exist in the CMDB, implement monitoring processes that run on a regular basis to interrogate the state of the CIs in the production environment of the data center. If there is a disruption in the functionality of the CI, the CMDB can be leveraged to determine the impact to other CIs and the business in order to maintain business continuity.</td>
</tr>
<tr>
<td><strong>8. Move</strong></td>
<td>Move hardware and update the CMDB to reflect changes. As a data center evolves and modernizes, and as companies change through growth and mergers, hardware sometimes must change location to meet the demands of the changing business. The CMDB provides the information needed to make these moves successful without disruption to the business.</td>
</tr>
<tr>
<td><strong>9. Retire</strong></td>
<td>Archive or remove altogether CIs that over time no longer meet any business purpose for an organization in the CMDB. The CI and its attributes and relationships are purged from the CMDB as it is no longer needed for any impact, asset, configuration, or compliance reports.</td>
</tr>
</tbody>
</table>

**1. Define**

**Strategy and Goals**

The first step in a successful CMDB implementation involves defining the requirements for the processes and use cases for which the CMDB will support your business. The goal of this step in the evolution of CI data is to ensure that your final CMDB will contain the data and information the other IT service management and compliance processes require so they can be as efficient as possible. This requires an understanding of the types of technologies that exist in your organization as well as your organization's business goals, strategies, and challenges.
An organization’s strategy and goals dictate the strategic IT projects and business services required. The IT projects will dictate the types of technologies that the organization will need to maintain. The technologies will determine the types of configuration items that the CMDB needs to track.

A corporation’s strategy is not just the one described in the letter sent by the CEO to the shareholders. It is also the technical strategy laid out by the Chief Technology Officer (CTO) that helps an organization manage technology resources in the most efficient manner. This includes the applications that will be developed or acquired, virtualization needs, cloud strategy, storage needs, disaster recovery, network strategy, hardware standardization, security needs, and globalization needs. Lastly, the compliance and regulatory requirements must be understood, as documentation to meet these requirements will use information from the CMDB and will dictate the information that is stored in the CMDB.

**CMDB Standards**

The CMDB can display views of data, providing you with the information you need to make wise IT business decisions and efficiently perform IT service management processes. The CMDB manager must plan ahead as to the types of required reports and dashboards that are unique to the organization, as well as reports that are defined by the industry.

These are the standards set for the organization. Once standards and guidelines are defined, it will be easier to make decisions about CMDB behavior. Data standards will make managing a growing organization much easier.

**Company IT Standards Represented in the CMDB**

Acme Inc. has a new strategy and is offering a new product line and a business unit called Acme Express. In order for their sales and marketing department to support this new product line, they need a new Customer Relationship Management (CRM) application to track product inventory, customers, opportunities, and sales.

The Chief Information Officer (CIO) and CTO must deliver the CRM application as soon as possible so the Acme Express business unit can operate. They follow an established standard in Acme Inc. and develop the CRM application using IBM WebSphere with an Oracle database running on Linux servers, since the company expertise already exists in WebSphere and DBAs already have expertise in Oracle. The CMDB must allow for CI types of WebSphere application server, Oracle database, and Linux servers. The CMDB manager must be aware of how WebSphere Application Server is architected with the Oracle DB.

![Figure 1: Example CRM implementation](servicenow.com)
2. CREATE/UPDATE CONFIGURATION ITEMS

In this phase, the individual instances of defined configuration item types are inserted as records into the CMDB. In addition, the relationships between different configuration items are established. There are a variety of methods used to populate the CMDB with CI information. Automated discovery tools can make the task of populating data faster and more reliable. However, how these tools are applied is very important - automation must be thought out extensively in order to ensure that the goals of the CMDB are met. Otherwise, it can be a time-consuming and expensive task.

Populating Configuration Items into the CMDB

A configuration item could consist of any hardware or software in an IT shop. It can also be a non-technical resource such as a user, a department, an application, a data center location, or anything else that relates to technology resources and that needs to be tracked in the CMDB. It is most important to an organization to understand the business use cases for why they need to track configuration items. The business use cases will dictate which configuration items need to be created and stored in the CMDB.

Once the use cases are established for the CMDB and the necessary configuration items are identified, the correct method for populating the CMDB with these CIs needs to be established. Automated processes that can discover configuration items are the most efficient, repeatable method for populating the CMDB. When business application configuration items need to be tracked, then a business service mapping approach must be employed. Otherwise, an infrastructure discovery customized as necessary to collect the correct configuration items should be used to populate the CMDB. Some CI attributes related to business data such as the department owner for a business application or the asset ID for a computer system, are populated manually. For most organizations, it is probably necessary to use a combination of most of these methods to populate the CMDB.

Below are some typical use cases with the corresponding configuration item types that are needed and method for populating the CMDB.

<table>
<thead>
<tr>
<th>Business Use Case</th>
<th>Configuration Item Type</th>
<th>Method for Populating the CMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Management</td>
<td>Business Application Names with related configuration files, software application components with related servers, network devices, storage devices, virtualization information</td>
<td>Business Service Mapping</td>
</tr>
<tr>
<td>Event Management</td>
<td>Business Application Names with related configuration information, software application components, server machines, network devices, storage devices, virtualization information</td>
<td>Business Service Mapping</td>
</tr>
<tr>
<td>Incident Management</td>
<td>Business Application Names with related configuration files, software application components with related servers, network devices, storage devices, virtualization information</td>
<td>Business Service Mapping</td>
</tr>
<tr>
<td>Asset Management</td>
<td>Servers, desktops network devices, and installed software applications with financial value</td>
<td>Infrastructure Discovery, Manual Entry</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Configuration Management</td>
<td>Servers, desktops, network devices, installed software components, and software patch information</td>
<td>Infrastructure Discovery</td>
</tr>
<tr>
<td>Compliance Reports</td>
<td>Servers, desktops, network information, installed software, related business application information or any information needed to meet required compliance reports</td>
<td>Infrastructure Discovery</td>
</tr>
</tbody>
</table>

### Configuration Item Identification

One important consideration in populating the CMDB is uniquely identifying CIs. Often two distinct CIs will have data that is similar. It is important to understand how technological components are deployed in your organization and what bits of information can be gathered for these components that can be used to uniquely identify different CIs. For example, if host names are repeated among different computers, and the host name attribute is used as the unique identifier for a server host, then only one CI will be created upon discovery for multiple hosts. Alternatively, if hosts are identified by an attribute that might change from time to time, such as the primary IP Address, then multiple CI entries will be created in the CMDB when only one host exists. In either case, the CMDB will contain inaccurate information.

The CI identifier must be one or more attribute that is static for the life of the configuration item and that is unique within all configuration items on the network. This sometimes requires a complex CI identification method to be built into either the discovery mechanism, or built into the CMDB.

### Automated Business Service Mapping

One of the most important use cases for the CMDB is tracking the impact different hardware and software has on individual business services/applications. This is important for Change Management and for regular monitoring of applications to prevent outages. One of the most common reasons for an outage in a business application is changes made to the infrastructure on which a business application depends. By understanding the relationships between the infrastructure and the business applications, changes can be made in such a way that prevents negative impact.

The fastest method for discovering the relationship between infrastructure and business services is using automated business service mapping. This involves scanning the business application from an entry point and tracing the application flow from one server to the next until all the dependent servers and software components are mapped. During this scan, configuration information is read to understand application dependencies.
Automated Infrastructure Discovery

The other automated approach to populating the CMDB involves horizontal network discovery. This involves an automated process of scanning all IP Addresses in the network of an organization. From each IP Address, the automated process will log into the device and collect hardware and software information. This effectively gathers an accurate inventory of each device in the organization.

Figure 2: Service map showing infrastructure that supports the Credit Check business service

Figure 3: Infrastructure inventory and dependencies
“Having a single system of record and common platform was critical to the communication, automation, and planning that was required for DevOps.”

Integration with External Data Sources
External data sources often contain information that needs to be populated into the CMDB. These are usually established internal databases that are already maintained by the organization, or a vendor-supplied database with configuration information that is relevant to the CMDB. If the information in the external source is static, then a one-time import of the data to the CMDB is all that is necessary. More often the data will continuously change, for example if the data source is an application maintained by the organization in which records are regularly updated. The organization may depend on this application so a regular feed will have to copy the relevant configuration data over to the CMDB.

Manual Entry
Manually entering information about CIs is time-consuming and can be error-prone which makes manual entry for CIs very unattractive. Even in a small organization, too many changes take place and it cannot be depended on for very long. Yet, many organizations start out using this method to establish their CMDB before automated processes are created.

There is some data that cannot be gathered automatically such as business and organizational information. Information about people cannot be gathered by a scan of the network and must be entered manually.

Attributes that Must be Gathered Manually

<table>
<thead>
<tr>
<th>Data Center Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floor</td>
</tr>
<tr>
<td>Rack-Mountable?</td>
</tr>
<tr>
<td>Department Owner</td>
</tr>
<tr>
<td>Asset Tag</td>
</tr>
<tr>
<td>Business Unit</td>
</tr>
<tr>
<td>Organizational Unit</td>
</tr>
</tbody>
</table>

Figure 4: Examples of manual-entry attributes

Information entered manually does not necessarily have to be entered by the CMDB manager. Processes can be established that require employees to enter information about data center resources for which they are responsible. For example, an employee or a department owner can fill out an asset request through an automated service catalog and have many of the manual fields required for the CMDB for that configuration item already populated.
3. CONFIGURE/ENRICH

Once configuration items and their individual relationships are populated in the CMDB, individual attributes may have inconsistent data formatting. For example, consider the following attributes populated by two different discovery methods and the normalization process used to update the attributes:

While the values for Memory and Disk Space are technically the same in each CI, the values are stored in different formats and this may cause confusion in reporting. Also, if the information is not normalized to a common format, then a subsequent discovery that identifies the same technical value but in a different format, will wrongly detect that value as different when it is not. This will be reported as a change in the CI when no change exists.

After the configuration item is populated, a normalization process must be invoked on the CIs to ensure consistent formatting.
4. ASSESS/MAP

One of the most valuable use cases for the CMDB involves business service mapping. This involves identifying the individual configuration items that are devices (servers and network devices) and storing a relationship between each device and the business application that depends on them. By mapping the business applications, the CMDB turns into an effective tool to support your business and not just a technical tool to support incidents, problems, changes, etc.

Creating such mapping is best done with an automated business service mapping method that automatically identifies the business application servers and their dependencies. Such a method was described above under the “Automated Business Service Mapping” section.

This mapping can also be done manually in the CMDB, but this is risky as the information may change. A business application owner may install the application on new servers or reconfigure a new server component without notifying the CMDB manager. This will cause the information in the CMDB to be inaccurate with regards to business service mapping. Any decision based on this inaccurate mapping data can cause problems for application stability and business continuity. For this reason, automated discovery with accurate business service mapping is the best approach for populating the CMDB.

![Figure 8: Examples of CRM and HR business service maps](image_url)

5. RECONCILIATION

Information that populates configuration items may frequently come from multiple data sources. These data sources may provide information for the same attributes for the same CIs as other discovery sources. This might cause the information in the CMDB to be inconsistent. To prevent attribute values from being overwritten by unauthorized sources, the CMDB manager must designate a single source of truth for each attribute of each CI type and assign these data sources accordingly. There should be only a single source of truth for any attribute of any CI, otherwise the configuration item will be marked as changed on a regular basis when no changes were actually made.

![Figure 9: Attribute values from multiple data sources reconciled to a single source of truth for a CI](image_url)
6. DATA CERTIFICATION

Data certification manages scheduled and on-demand validations of CMDB data. Information is added to the CMDB by automated discovery, importing from third-party tools, or manual entry. For regulatory needs and for final quality control of the CMDB, the CI data needs to be checked for accuracy. The person or team responsible for certification can define what sample of information will be audited for certification and will set a certification schedule. The schedule then generates a checklist for verifying the data. Individuals assigned to certification tasks will check the data rigorously following all standards of measurement to ensure the accuracy of the data.

Data certification does not need to be performed against all data in the CMDB. Only a subset of the data is required for a valid certification but the different data sets must sample different types of data so it validates all the automation mechanisms in place. Here are examples of some certification samples:

- Query a set of CIs that has been updated by two different discovery mechanisms and ensure that the proper sources of attribute data are applied to the different attributes for which they are assigned.
- Query a set of CIs that have the following types of dependency relationships and ensure that the relationships have been created:
  - Host to Host dependency
  - Installed Software running on a Host
  - Host is part of a Business Service
  - Virtual host has a relationship to the physical host
  - Host Application Server has a relationship to the dependent Database Server
  - Query a set of CIs with attributes that are manually entered by business owners and ensure that the users entered the data correctly.

7. MONITOR

At this point, the CMDB is populated regularly and processes are in place to normalize the data and certify that the data is accurate. Configuration items can now be created and run through their life cycle to serve the needs of the organization.

Organizations now need to regularly monitor CIs and their relationships to identify ongoing problems that can cause inaccuracies in reports. Regular monitoring can alert the CMDB manager of these issues and allow for repair processes to adjust the CMDB data as necessary. Here are examples of some of the problems with CIs that might be detected:

- **Stale CIs:** A CI becomes stale when its existence has not been verified for a determined period of time. This could be one or two weeks for a host or network device. It could be as little as a few days for a network dependency connection. A CI may become stale in the CMDB because it has been physically removed from the network and thus can no longer be found by automated discovery processes. The CI may still exist, but it may be damaged or intentionally taken off the network. In any case, the CMDB manager must investigate the situation with all concerned parties and initiate a fix either to the CMDB or to the hosts on the network.

- **Duplicate CIs:** If the identification rules for CIs do not ensure that each CI is uniquely identified with parameters that do not change, then duplicates of the same CI will appear in the CMDB. This data must be cleaned up and the identification rules and processes modified as necessary in order to ensure that this problem does not occur again.
• **Overloaded CIs:** An opposite problem of duplicate CIs exists when different CIs are identified as the same CI and one CI is created when several CIs should have been created. This can indicate a problem with the identification rules and processes and must be handled as soon as possible to fix the CI data in the CMDB.

• **Unidentified/Orphaned CIs:** CIs can appear in the CMDB that are not identified. They are seen on the network as an IP Address but no other information is available for them. In addition, these orphan CIs have no relationships with other CIs. This is usually an issue with access rights for automated discovery processes. The discovery process is not able to log into the newly appearing host to identify its existence and to inventory it as required for all the processes established in the CMDB.

### 8. MOVE

As a data center evolves and modernizes, and as companies change through growth and mergers, hardware sometimes must change location to meet the demands of the changing business. The CMDB provides the information needed to make these moves successful without disruption to the business. After the move, the CMDB must be updated with the new information.

The CMDB provides invaluable data on configuration items and their interdependencies as preparation for any move. As a practical matter, devices in the CMDB must be tagged as part of move groups. These move groups identify the collection of devices that will be temporarily de-activated from the network, packaged up and then transferred to the new location where they will be reconnected to the network. This process could take several days depending on the complexity of the move. The CMDB must allow for devices to remain in limbo during the process. Automated discovery processes will still run and try to connect to the devices to rediscover their information.

To prevent CIs from wrongly aging out as they are going through their relocation, move processes must identify each CI that is disconnected from the network because of their transfer. These processes must prevent any aging mechanism from removing the CIs from the CMDB since they are not in an unknown connection state.

Once the devices reconnect to the network, automated discovery can continue to update their attributes. Attributes related to location, need to be updated. Other attributes such as dependency connections may also change as the CI now exists in a completely new location and its network topology is different. There will probably be different network devices supporting the servers and possibly other dependent resources such as storage devices and virtualization.
9. RETIRE

Every individual CI will someday be retired. For a CI to be “retired,” the following criteria must be met:

- If the CI is a physical device, the physical CI must no longer exist in the network infrastructure of the data center. This must be verified through an automated process and possibly also through a certification process.
- A sufficient aging time period must have passed. This time period could be different depending on the CI type. A host CI may have as much as 2-3 weeks before confirmation procedures dictate that the CI be retired. A dependency connection may only get a few days.
- The CI must be marked as retired but its information must still be retained for reporting purposes. Often, there are regulatory requirements which dictate the length of time CMDB records must be retained. There should either be a separate report that lists all CIs that have been retired during a time interval or a change report shows that this CI has been retired and no longer exists in the topology of a business service. In either case, documenting the retired CIs, as a change to the environment is as important as documenting new CIs added to the environment.

CONCLUSION

- Once a high quality of CI data is achieved, we can concentrate on the overall value-add of the CMDB within an IT organization. IT processes can now leverage the benefits of efficiency and control of a fully populated and service-aware CMDB. IT organizations can therefore realize the following benefits:
  - Improve ITIL processes such as Incident Management, Change Management, and Problem Management by providing an impact analysis of configuration items to business applications.
  - Reduce the mean time to recovery (MTTR) of outages of applications because application topology and all their dependencies are understood.
  - Improve the effectiveness of monitoring tools by associating business impact with detected problems on infrastructure components.
  - Allow for effective strategic planning of the data center. Hardware can now be moved, consolidated, and optimized without disruption to the business that depends on it.
  - Better manage and control costs of IT assets since there is an accurate understanding of what assets are deployed and used.
  - Provide data for compliance and audit reports that are required of IT organizations.
  - Better control of configuration changes needed for on-going maintenance of software and hardware.

Once the processes to maintain the CMDB are effectively established and the benefits realized, the on-going maintenance effort will be minimal. The value of clear, concise, and correct information describing the CIs in the CMDB will fundamentally change the way IT communicates to itself and the rest of your organization.