

Discovery and the Configuration Management Database (CMDB)

OCTOBER 2008

Marv Waschke, Alan Kasper,
Randal Locke and Charles Jimerson

CA SERVICE MANAGEMENT

Table of Contents

Executive Summary

SECTION 1: CHALLENGE 2

Obtaining an ROI from a CMDB is a Challenge

Define the CMDB Value and Focus

Time-to-Value

SECTION 2: OPPORTUNITY 3

Discovery Tools Can Help to Create CMDB ROI if Used Correctly

How Discovery Works to Support a CMDB

Step-by-Step CMDB Population and Maintenance

CMDB Maintenance and Discovery

SECTION 3: BENEFITS 11

A Quick Return on Your CMDB Investment

SECTION 4: CONCLUSIONS 12

SECTION 5: GLOSSARY 12

SECTION 6: ABOUT THE AUTHORS 14

Executive Summary

Challenge

A Configuration Management Database (CMDB) can address many problems:

- Intentional changes with unintended consequences, like upgrading a file server in the Kansas sales office that brings down the New York City data center.
- Delayed incident resolution.
- Poor estimates of the impact of incidents, such as a panic call at 1:00 a.m. when a 4-terabyte (TB) storage unit goes down, only to discover that it was a spare on the test LAN.
- Outlay for hardware that does not deliver an improvement in service performance.
- Poor governance and security decisions, such as locking down an insignificant print server while leaving payroll record storage wide open.
- Projects that take far longer than predicted and do not deliver at predicted capacity.

However, simply installing a CMDB will not solve these problems. For a return on a CMDB investment, the CMDB must be populated with the right information and maintained to ensure accuracy. This involves performing a complex series of tasks that must be undertaken with care and intelligence. The rewards are immense, but there are also many pitfalls.

Opportunity

In a nutshell, a CMDB is a dynamic blueprint for IT services and infrastructure. When a building engineer addresses a problem or plans a renovation, they reach for the building blueprint to figure out what is wrong and the consequences of damages. They study the blueprint to decide the best way to proceed. IT engineers must do the same thing, but without a CMDB IT staff have only fragments of data on which to make critical decisions. Some pieces are in the network management system, others in the asset management system, still others are floating around the department in people's heads or in notebooks on shelves. It is no surprise that there are problems! An efficiently and accurately populated CMDB provides the direction needed for optimal reaction to adverse events and planning for success. The tools are there to populate and maintain a CMDB, but they must be used correctly.

Benefits

The CMDB is a strategic blueprint for planning and controlling IT strategy, design, implementation, and operations. With an effective CMDB, IT departments plan better, make fewer mistakes in implementation, exercise more precise control of configuration, execute changes with less risk, and respond more quickly and effectively to incidents and problems in operations.

Obtaining an ROI from a CMDB is a challenge

Define the CMDB Value and Focus

A Configuration Management Database (CMDB) is a tool for managing operational risk and improving overall business response and cost by tracking the linkages between business services and the supporting IT infrastructure. A CMDB contains configuration items (CIs) which are the components of the IT system, their properties, and relationships between CIs. Without the knowledge contained in the CMDB, organizations have a limited understanding of the effect of a given IT component on their critical services. This results in inaccurate estimates of the impacts and potential consequences of a change to the infrastructure. When an outage occurs, operators often have to wait and see what happens instead of taking immediate remedial action. This unpredictable and uncontrollable lag between occurrence and remediation may result in substantial costs in downtime and performance. By understanding the relationships between services and infrastructure, a CMDB makes it possible to reduce downtime or performance degradation. This is one of the reasons for implementing a CMDB.¹

A CMDB is the focal point of IT operational information. In addition to recording the configuration of a service, a Configuration Management System (CMS), used in concert with Change Management, manages the configuration of the IT infrastructure and constitutes the single source of truth for both actual and authorized configurations. An authorized configuration is usually the cumulative product of decisions by a Change Advisory Board (CAB) and describes the configuration of IT systems that has been reviewed and endorsed by the CAB. A CMDB or federation of CMDBs is the data repository for a CMS. Authorized configurations should be carefully protected from contamination with unauthorized data in a CMDB. Without an authorized CMDB, determining the CAB approved configuration requires laborious and error-prone examination of cumulative change orders. A CMDB collects together CAB decisions in a single repository. Reviewing changes collectively, the CAB can see the big picture while making their decisions, and the IT department can properly assign value, priority, and risks to IT efforts.

A CMDB primarily supports other functional areas and their tools, such as Service Desk or Capacity Planning. In this role, the value of the CMDB is directly proportional to the quality of its content and its ability to improve the execution of the services and functions that consume its information. A CMDB must be populated correctly and it must be surrounded by processes that keep it accurate and up to date. Without a solid initial population and continued support through sound processes, having a CMDB could be worse than not having one at all. Inaccurate information is sometimes more harmful than ignorance.

The tools that are used to identify what is in the IT environment, CIs, their properties and their relationships, are called discovery tools. The effective use of discovery tools is a key to a successful CMDB implementation.

¹ **Note:** Section 5, References, contains a glossary of the terms used throughout this paper.

Time-to-Value

In many cases it is best to start with just a few critical applications and model them to the level where changes have a history of producing problems. Gaining value from a CMDB requires an upfront investment in time and resources. The key to ensuring success is to start with small, high value projects that leverage the unique value of the CMDB, and then branch out to include other areas. This helps ensure that the return on investment will begin early and continue to grow. CMDB projects fail when the upfront costs are high and positive returns are slow to appear. This paper shows you strategies that can help to ensure early returns on your investment.

The first step is to decide what data you need in the CMDB to support your high value projects. This seems self-evident, but probably the most frequent and far-reaching startup error is to skip this step and to populate the CMDB with an automated discovery of every component in the infrastructure. This approach allows laptops, desktops and other components that may not impact critical business services to find their way into the CMDB as well as information that is so broad and so deep that there is no hope of maintaining it all. At best, these irrelevant components are clutter; in many cases, the clutter obscures important facts about the critical components. At worst, maintaining this irrelevant data consumes time and money and focuses resources on the wrong things.

This is not to say that laptops or desktops never belong in a CMDB. If an important service depends on a particular desktop, then by all means include it. However, when seeking early ROI while establishing a CMDB practice, a selective CMDB of highly critical CIs will usually deliver value with less investment than a less selective CMDB with broader content. ITIL defines a CI as: Any Component that needs to be managed in order to deliver an IT Service. Information about each CI is recorded in a Configuration Record within the Configuration Management System and is maintained throughout its Lifecycle by Configuration Management. CIs are under the control of Change Management. CIs typically include IT Services, hardware, software, buildings, people, and formal documentation such as Process documentation and SLAs.² Keep this precept in mind to simplify the project and optimize the return on your CMDB investment.

SECTION 2: OPPORTUNITY

Discovery Tools Can Help to Create CMDB ROI if Used Correctly

How Discovery Works to Support a CMDB

Many CMDB administrators are challenged to maintain accurate and up-to-date information on CIs and their relationships within their environment. To meet the challenge, they must understand the use of the various discovery tools that can help populate and maintain information in the CMDB efficiently. The following paragraphs lay out a step-by-step methodology for using discovery tools to populate and maintain your CMDB while building steady ROI.

The strategy begins with a focus on IT business services. The relationships between CIs and the services they support are at least as important as individual CI details. Discovery and automation assists collecting CI and relationship information in an organized manner. Relationship information is made available by providing a logical model of the infrastructure and relating that logical model to IT and business services.

² ITIL® V3 Glossary v3.1.24, 11 May 2007, http://www.best-management-practice.com/gempdf/ITIL_Glossary_V3_1_24.pdf

DATA SILOS AND MDRS An IT silo is an independent source of information on a particular aspect of the IT environment. For example, an accounting application that tracks IT assets but does not relate these assets to the resources managed by the network management system is a silo. A CMDB is an aggregator of data and an integrator of IT silos. When speaking of CMDBs, these silos are more commonly known as Management Data Repositories or MDRs. These MDRs are often sources of discovered data. Depending on the discovery tool, these MDRs often contain information about servers, network components and other relevant data. The data can be very detailed or less detailed depending on the tool used for the discovery process. For example, a discovery based on a ping sweep will usually be less detailed than a discovery that relies on several mechanisms, such as interrogation or installed agents. The data contained in isolated silo MDRs often has value, but the value has to be extracted with attention to its relevance and accuracy.

It is emphasized here that a CMDB is selective; most often, only a subset of all discovered IT asset information should be imported into the CMDB. Start the process of identifying the right subset by defining the problem being solved and the data required to support the solution. This in part will define what to discover. Note that in most cases you will not need or want to import all data available from an MDR if the CMDB user or consuming application can easily reach out to the MDR for further information on a CI via federation.

Step-by-Step CMDB Population and Maintenance

The information contained in the CMDB should be a model or representation of IT Services within the IT Infrastructure. However, not all elements in the environment should be or can be discovered. Establish ground rules by resisting the temptation to populate the CMDB with all data within the enterprise. Only those CIs and relationships that satisfy required business requirements should be discovered and imported into the CMDB. The CI and relationship discovery and population process is represented in Figure A and discussed in the subsequent paragraphs:

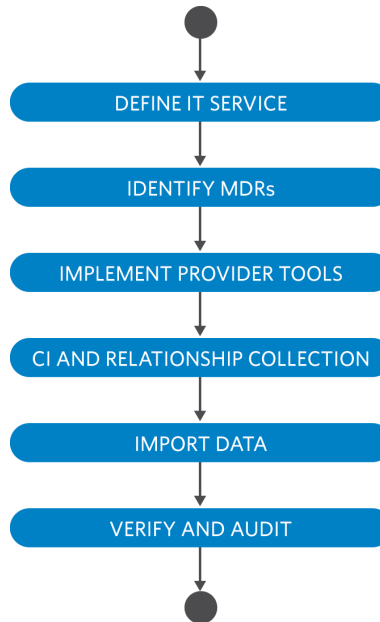
DEFINE IT SERVICES The first step in the process is the identification of IT Services. An IT Service can be thought of as a set of related components provided in support of one or more business processes. An IT Service can represent any process or a group of processes that you have identified as being supported by a group of managed resources. In fact, any conceivable activity supported by managed infrastructure resources can be modeled as an IT Service. On a more practical note, a service might typically represent a web-based retail transaction service, an application server service, a printing service, an email service, a routing service, or a source control service. These are but a few familiar examples of the many IT-based services that you can model as IT Services. Your choice of what to model as a service should be based on the priorities of your business.

Businesses that have a Business Continuity Plan (BCP) have already done much of the analysis needed for maximizing the return on a CMDB. Even enterprises that don't have a formal BCP, often have developed similar plans for disaster survival that identify the critical functions and resources for the enterprise.

FIGURE A

In building your CMDB be very focused. Seed the CMDB with a single or small number of critical applications or business services so that your implementation is manageable.

PROCESS FOR POPULATING A CMDB



Examine the business continuity plan to determine which business services and infrastructure components are critical to the survival of your business. Begin seeding the CMDB with a single or small number of critical applications or business services. Remember, this is about criticality, not numbers. It is more important to identify the most critical CIs and services than it is to put every device and service into the CMDB.

If you don't have a BCP or other survival plan, begin by identifying a list of critical services, critical applications, and key supporting infrastructure devices. Most organizations are aware of those applications that are critical to sustaining their business. Look in particular to those applications that are focused on sales, manufacturing, or managing the supply chain. Look for those applications that link the company with customers and other service providers. These are what keep the business viable and help to ensure that the CMDB is focused on those activities that add value. You may find it helpful to read some of the extensive literature on BCP if you are having trouble identifying what is critical.

Before creating an IT Service, consider what the service should represent, what resources impact the viability of the service, and what level of service viability (or service health) can be inferred from the condition of the resources that support the IT Service. Not all CIs affect a service in the same way. For example, the server that hosts a database engine has a more significant effect on the health of the service than a workstation that hosts an administrative interface for the database. The significance of a CI should be reflected in the type of relationship the CI has to the service and should be configured into any automated systems management tools that monitor the service. Keep in mind that CI to service relationships are used to trace impact and root cause. Choose relationships to record in the CMDB that are most applicable to these uses.

IDENTIFY MANAGEMENT DATA REPOSITORIES The next stage in the discovery process is identifying appropriate Management Data Repositories (MDRs.) Most IT organizations have no shortage of MDRs. Identify the MDRs that may be considered an authoritative source for sets of data. A backup MDR may contain database information, while another may contain information about systems and applications, and others may contain information about the network infrastructure.

MDRs vary greatly in sophistication. Some MDRs are mini-CMDBs that break out services and relate them to CIs. Others are less sophisticated and you have to define the services in the CMDB first and then link them to CIs federated from the MDR.

The choice of MDRs to use and how to use them should be driven by the services chosen in the previous step. Do not make the mistake of using an MDR only because it is there. A federated MDR must contribute to the value of the CMDB.

When new MDRs are identified, establishing CI types is often a concern. Discovered CIs must be classified into types. The types native to an MDR will not necessarily correspond to types in the CMDB. Type mapping is one of the first activities performed when a new MDR is federated. This is performed before the initial data load of the CMDB or when a new type of CI is identified and needs to be included in the CMDB. Usually, the MDR type is transformed into a CMDB in the transfer from the MDR to the CMDB. In most cases, MDR types can be mapped to existing CMDB types, but occasionally a new type will have to be added to the CMDB.

IMPLEMENT PROVIDER TOOLS TO COLLECT CI AND RELATIONSHIP INFORMATION

Implement tools to collect CI and Relationship information from the MDR providers and populate the CMDB. Automated processes to load and update the CMDB should be developed to reduce costs and human error. Note that discovery tools, inventory tools, enterprise management tools and network management tools can all be federated to the CMDB.

If possible, take advantage of industry specifications, such as the CMDBf CMDB Federation specification (<http://cmdbf.org>) for implementing provider tools. This is a vendor neutral specification for linking MDRs and federating CMDBs using Web services. It was developed by a consortium of IT management vendors (CA, BMC, Fujitsu, HP, IBM, and Microsoft). By following vendor neutral specifications, you avoid the work of implementing a completely different interface for each MDR. As the specification becomes widely accepted, more commercial MDRs and CMDBs will ship with CMDBf services, and MDR federation will become easier and less complex.

Analyze the data in the discovery tool to verify that it is finding the items defined in the business continuity plan. It is these components that will be maintained in the CMDB through Change and Configuration Management.

COLLECT CI AND RELATIONSHIP INFORMATION

Once the CI and relationship information collection tools are in place and configured, they can be used to collect information for the initial population of the CMDB and for comparing the actual configuration information with data contained in the CMDB. Some collection tools are configured so that they know how to find and interpret information. Keep in mind that being able to uniquely identify a CI is critical to the process. Systems, for example, should have a fully resolved Server/Host Name and not just an IP Address for discovery and collection processes. Make sure you understand the assumptions of what makes a CI unique.

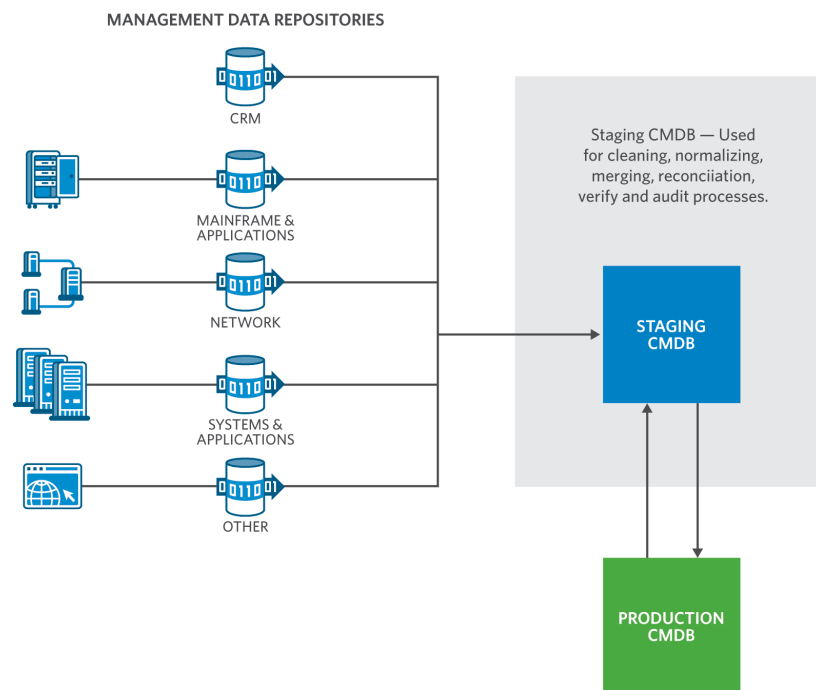
IMPORT CI AND RELATIONSHIP DATA

Early in the process it is valuable to perform a minimalist discovery. For example, perform a ping sweep of the components identified in the BCP that are defined as tier 1 critical applications, associated databases, components, etc. This step defines a portion of both the CI attributes and relationships.

FIGURE B

Export or federate the data from the discovery tool to populate the CMDB with only those CIs relating to that Business Service.

MULTIPLE SOURCES I.E. MDRS CAN CONTRIBUTE DATA TO THE CMDB



Baseline these components in the discovery tool. Different tools do this in different ways. The simplest baseline is a simple saved file. Some tools will build baselines automatically.

Export or federate the data from the discovery tool to populate the CMDB with only those CIs relating to that Business Service. Process one critical service at a time, concentrating on quality and value rather than the number of services.

Federate with other sources to find additional attribute information as required. With this step you round out the CI records in the CMDB for each critical service or application.

A variety of methods are available for mapping CI and relationship information for import to the CMDB's schema. Generally, a process is run to automatically generate an XML file to create CIs and relationship and MDR provider information. The MDR provider information is useful in supporting the external launch in context (i.e., reach out) of a CI from the CMDB to the federated source. Once the information is processed and prepared, the XML file is imported into the CMDB.

VERIFY AND AUDIT CI AND RELATIONSHIP INFORMATION

A series of reviews and audits that verify the physical existence of CIs and relationship information check that they are correctly recorded in the Configuration Management System and are necessary to ensure data integrity.

The content of the CMDB should be checked against the IT environment periodically. This is accomplished by an audit and verify process. CMDB audits are most often performed under the supervision of the configuration manager using a divide and conquer strategy. Most CIs are assigned to specific managers in the CMDB. CI managers are responsible for auditing and verifying their own CIs. Typically, CI managers will pull a report of their CIs and review each one against the real environment. Ideally, the real environment will be discovered and compared automatically with the authorized configuration. It is also good practice for CI managers to pull daily or weekly reports of changed CIs that are assigned to them and audit the changed CIs for accuracy. A CMDB that logs changes is of great value in these audits. The negative effects of corrupt data cannot be underestimated and a site should always be on the lookout. Vigilance is rewarded with smooth and successful operations.

CMDB Maintenance and Discovery

The population of a CMDB never ends; new hardware is acquired, old hardware is retired and redeployed, new services are rolled out and old services are transformed. As this happens, segments of the CMDB are populated with the new CIs and relationships. In addition to continuous population, the CMDB must also be maintained as CI configurations change and relationships are added and modified. Sometimes maintenance and population are hard to distinguish, and there is no real need to make fine distinctions. Note that population generally only involves new CIs and relationships while maintenance is concerned with verifying and updating existing CIs and relationships.

CMDB maintenance is important because information is the single most important resource that an organization has to manage. To maintain the accuracy of data that is aggregated in the CMDB, appropriate tooling and data collection methods must be in place. This ensures that data within the CMDB is not adulterated with corrupt data or allowed to become stale. The usefulness of the CMDB is dependent on the quality of the data that is stored within it.

Automated discovery is often the only pragmatic method of maintaining the integrity of data represented in the CMDB for a large organization. This may seem counter to ITIL Change Management practice where the CMDB is only changed through change-order, but it is often an effective way to maintain the CMDB. This is especially valuable when a site has a preapproved change order practice in which the CAB delegates control of some areas of configuration to designated managers and technicians. In this case, automated discovery may be the best way to true up the records in the CMDB with the de facto authorized configuration.

In addition, a CMDB practice may record as is or observed configuration data in the CMDB and distinguish it from authorized baseline data. This is a substantial improvement over a system that records only the results of change orders. For example, in troubleshooting problems on the service desk, comparing the results of an automated discovery to the authorized baseline can be a powerful tool.

Automated discovery should be used to achieve greater efficiency in maintaining the integrity of data in the CMDB, but auditing processes must be involved to ensure that changed information was in fact authorized to be changed. The CMDB serves as a point of integration between Change, Configuration, Incident, Capacity and Availability Management processes. The CI and Relationship discovery process greatly improves time-to-value when implementing the CMDB. Automate when it makes sense to do so to eliminate human error and increase efficiency.

COMBINING AUTOMATED AND MANUAL SERVICE DISCOVERY

Manual discovery is discovery that involves a substantial amount of human intervention. It is appropriate for capturing data that cannot be obtained from automated scans. A frequent example of an undiscoverable CI is a business service. Although automated discovery can use pattern matching, packet sniffing, and other sophisticated queries to infer the existence of services, truly unique services are seldom discovered solely through automation. Critical services are sometimes so well-known that discovery is unnecessary and entering them by hand is easier than setting up a tool to discover them.

Often, automation and manual discovery are most effective when combined. For example, an accounting service may be well-known and some of its supporting CIs may be known, but a discovery tool could be set up to discover every CI of a certain pattern that shows up in the netstat (see glossary) output of the well-known supporting CIs. This discovery could reveal additional supporting CIs that you may not have been previously aware of. The previously unknown relationships discovered using this approach, which combines the manual discovery of the accounting service with automated discovery of its dependencies, can yield extraordinary ROI when they are used to evaluate previously unpredictable impacts and untraceable root causes.

Automation is also very useful for discovering repeated instances of similar services, such as Domain Name Service (DNS) servers. For example, following an acquisition or merger, IT departments may literally lose track of where their DNS servers are located. Problems with DNS security underscore the importance of knowing the exact locations of these servers are so they can be patched appropriately. An automated discovery and maintenance in the CMDB can fill this need effectively.

CMDB MAINTENANCE VIA AUTOMATED DISCOVERY

The process for automating discovery is similar to manual population. Automated discoveries, collections and mapping provide an efficient method for populating the CMDB with up-to-date configuration information — including relationships between CIs. Automated discovery is especially useful where large numbers of CIs with similar roles and characteristics are to be managed.

The usefulness of the CMDB is dependent on the quality of the data that is stored within it. A CMDB often contains data about managed resources such as computer systems and application software, process artifacts such as incident and change records, and the relationships that exist between them. Avoid populating the CMDB with thousands or tens of thousands of CIs without relationship information; this may provide little value as it is the relationship information that enables one of the major benefits derived from having a CMDB.

Automated Discovery should also be used to achieve greater efficiency in maintaining the integrity of data in the CMDB. This holds true as long as auditing processes are involved to ensure that the information that changes was in fact authorized to change. The CMDB serves as a point of integration between Change, Configuration, Incident, Capacity and Availability Management processes. The CI and Relationship discovery process greatly improves time-to-value when implementing the CMDB. Automate when it makes sense to do so to eliminate human error and increase efficiency.

DATA CLEANSING

Automated Discovery and MDR federation are critical tools in maintaining the CMDB, but the quality of data found in MDRs varies widely. Data that is not high quality must be scrubbed, or cleansed, before it is entered into the CMDB. The old saying “Garbage in, garbage out” applies to CMDBs as well. The subject of data quality is complex and has been studied extensively. A few characteristics of high quality CMDB data are listed below:

- **Normalized** This can be as simple as consistent use of spelling and terms. For example, the printer manufacturer should not be entered as HP in one place and Hewlett-Packard somewhere else. Similarly, if MAC addresses are expressed as 00-90-96-B9-5D-AC in one place, they should not be expressed 009096B95DAC in another.
- **Rationalized** Are CIs classified correctly, for example, are all routers correctly classified as routers or are some recorded as general network devices?
- **Complete** Are all the required fields filled in? Some fields will be required by the system; others will be required by the way you use the system.
- **Integral** Are there dangling references that need to be completed before the data can be entered?
- **Correct** The input should not contain information that is inaccurate.
- **Relevant** Should this data go into the CMDB?
- Are there duplicates? Most CMDBs can handle duplicate records. However, duplicate data coming from an MDR is suspicious. Why are the duplicates there? Is there a problem with the data in the MDR?

When beginning to load the CMDB automatically, you must pay careful attention to data cleansing before loading into the CMDB. Often, data cleansing can be automated with scripts, and some Extract Transfer and Load (ETL) tools have features that can help with cleansing. Whatever tools you choose, you should always be prepared to deal with these data issues.

The process followed in data preparation and cleansing often involves a staging area. The staging area can be as simple as a collection of flat files that are reviewed before proceeding to loading of the CMDB. In other cases, an entire staging database may be an effective strategy. In this strategy, a set of staging tables are set up in a relational database and the data loaded, examined, and groomed before loading into the CMDB. Using a relational database for this allows you to use database tools for identifying anomalies and generating cleaned data. You may go as far as creating an entire shadow CMDB.

When adding large amounts of new data to a CMDB, release management discipline may be appropriate, moving from development to QA to GA databases with formal releases and test methodology. This level of formality may seem overly elaborate and time-consuming, but the contribution of a CMDB to the functionality and stability of IT can be critical; in that case, a high level of discipline is easily worth the effort.

As you become familiar with your data sources, you may be able to decrease the level of scrutiny. However, you will save much time and trouble if you inspect data carefully and fix any quality issues before loading. Generally, you will have the best results if you can fix data quality problems at the source rather than, at the last minute, filtering before entering into the CMDB. Yet, this is not always possible.

Using the high value approach outlined here, the CMDB quickly moves into production and becomes a critical resource. This makes data cleansing particularly important. When a new data source is tapped, you must be careful not to introduce corrupt data into the production CMDB. Although CMDBs are generally designed to identify some data issues, no system can catch all problems. This is a case where prevention is far better than a cure.

SECTION 3: BENEFITS

A Quick Return on Your CMDB Investment

Using the approach outlined above and summarized below a CMDB consumer will focus on developing a CMDB that addresses the most critical business issues in an organization. This approach will minimize both the cost and time of getting started and also reduce the long term costs of keeping the CMDB maintained by focusing on just those items that produce the greatest return.

Step by Step Summary:

- Define Business Services (Note: Managing IT by service and not technology is the message of ITIL.)
 - Identify IT Functions within Business Services
 - Identify Systems that support the IT Functions
 - Identify IT Services that are transparent to the User – Applications, Systems, Infrastructure, etc.
 - Identify IT Services that are not transparent to the User – Backups, Databases, etc.
 - Map CIs to the IT Service
- Identify and define MDRs and which attributes or relationships can be leveraged
- Create appropriate CI map for data collection

- Discover and filter data
- Load data into the CMDB
- Reconcile data by verifying and auditing CIs and their relationships

SECTION 4

Conclusions

A CMDB forms the foundation for many processes that contribute to running, growing, and transforming the business. When running the business, a CMDB can reduce costs by helping to contain the effects of incidents, improve the process of change, and provide governance and predictability to operations. When growing and transforming the business, the CMDB is the basis for effective planning. However, without the right, timely and accurate data, or with too much data, these benefits are lost. Automated discovery tools are critically important for maintaining high quality information in the CMDB if used correctly. However, without careful planning and methodical application, these tools can degrade the quality of data as quickly and easily as they can maintain it. Following the guidance in this paper, IT departments will be able to populate and maintain their CMDB efficiently and avoid the mistakes that prevent them from realizing the value of their CMDB investment.

Using this approach, you will minimize the time required to represent critical applications and business services as interrelated CIs in the CMDB and deliver value to your organization. This minimalist and business-focused approach can provide you with a high return on investment in a short period of time.

SECTION 5

Glossary

Baseline is an authorized reference configuration. Baselines may be based on a snapshot of the configuration at a given point in time, or they can be manually defined.

Business Continuity Planning (BCP) is an interdisciplinary concept used to create and validate a practiced logistical plan for how an organization will recover and restore partially or completely interrupted critical function(s) within a predetermined time after a disaster or extended disruption.

CMDB (Configuration Management Database) is a database used to store Configuration Records throughout their lifecycle.

CMDBf refers to the Configuration Management Database Federation Working Group, a consortium of IT management software vendors (BMC, CA, Fujitsu, HP, IBM, and Microsoft) formed in 2006 to develop a specification for sharing data between MDRs and federating CMDBs. The specification was published in August of 2007. The Distributed Management Task Force (DMTF) has accepted the specification for development into a DMTF standard.

Configuration Management System (CMS) is a set of tools and databases that are used to manage an IT Configuration data. The CMS also includes information about incidents, problems, known errors, changes and releases. It also may contain data about employees, suppliers, locations, business units, customers and users. The CMS includes tools for collecting, storing, managing, updating, and presenting data about all configuration items and their relationships.

Management Data Repository (MDR) is a source of CI management data. Usually, MDRs are software tools for managing some aspect of a CI, although they can be simple files, spreadsheets, or databases.

Configuration item (CI) is any component that needs to be managed to deliver an IT Service. Information about each CI is recorded in a Configuration Record within the Configuration Management System and is maintained throughout its lifecycle by Configuration Management. CIs are under the control of Change Management. CIs typically include IT services, hardware, software, buildings, people, and formal documentation such as process documentation and SLAs.

Discovery is the process of identifying the CIs and relationships in an IT system. Usually, discovery refers to using automated tools, although discovery can be manual.

Federation is a special form of data integration in which data remains intact in the data source. In a federation, data copying is minimized.

Installed agents are a form of discovery in which agents are installed on computer systems, and the agents monitor configuration changes and the applications and services installed on the computer. The agent then communicates with the discovery tool.

Interrogation is a form of discovery in which the discovery tool queries functions and services that are part of the operating system or other software and are not installed with the discovery tool.

Netstat (network statistics) is a command-line tool that displays network connections (both incoming and outgoing), routing tables, and a number of network interface statistics. It is available on Unix, Unix-like, and Windows NT-based operating systems. Remote execution of netstat by a discovery tool is a form of interrogation. The discovery tool uses netstat output to infer CI properties and especially relationships.

Packet sniffing (also known as a network sniffing, network analysis or protocol analysis) is a discovery technique that uses computer software or hardware that can intercept and log traffic passing over a digital network or part of a network. A packet sniffing discovery tool uses data acquired in this way to infer the existence of CIs and relationships.

Pattern matching is a discovery technique that uses predefined patterns, sometimes called blueprints, to examine data obtained from interrogation, installed agents or other means, and identify CIs and relationships by matching the patterns.

Ping sweep is a technique used to determine which of a range of IP addresses map to live hosts. It sends ICMP (Internet Control Message Protocol) messages requesting that the target node echo a message back to the sender. It is perhaps the simplest way to discover what is attached to a network, although it provides little information other than existence of a live host. Ping sweeps are typically supplemented with other discovery methods to provide more information.

Relationships in a CMDB refer to a logical or physical relationship between two or more CIs that is recorded in the CMDB. ServerA hosts ApplicationB is a typical relationship. Relationships are used to infer impacts and root causes.

Snapshot refers to the current state of a configuration as captured by a discovery tool.

SECTION 6

About the Authors

Alan Kasper
Principal Product Manager

Mr. Kasper has over 40 years experience in IT solutions and nearly 20 years experience in IT Service Management. Historically at CA, Mr. Kasper was the Product Manager for both CA's Service Support suite including CA Service Desk and the CA CMDB (configuration management) solutions. He joined CA as part of an acquisition in 1995. Currently he is instrumental in defining functionality and market strategies relating to CA's overall Service Management solutions including its CA CMDB. Prior to coming to CA he was a co-founder (Vice President of Sales and Marketing) of Networx Inc., which was acquired by Legent in 1993 that developed Paradigm, which has evolved into a major part of CA's current Service Support family of products. Before that Mr. Kasper worked for The Boeing Company wherein he held Program Management positions relating to systems and network integration.

Marv Waschke
Senior Advisor, Product
Management

Marv Waschke is Vice President of Development and Senior Advisor for Product Management at CA. Part of the Service Management business unit, Marv managed development of CA's service desk product and was active in the design and engineering of the CA CMDB product. He was an author and editor of the CMDB Federation specification (found on cmdbf.org), and an author of an early version of Service Modeling Language specification, and the Common Model Library white paper (at cml.project.org). He currently sits on the DMTF CMDB Federation Work Group. Recently, Marv has been active in developing the CA Unified Service Model and integration platform.

Marv's opinions on IT service management and standards appear regularly in his blog: "Iterating on IT Service."

Randal Locke
Director of Technical Sales

Mr. Locke has more than 20 years of experience in IT Service Management. He has been instrumental in the development and delivery of ITIL solutions for large Clients in the Defense, Services, Manufacturing, Financial industries and within the Federal Government. He has performed these Service Management consulting services in North America and Europe.

Areas of expertise include Service Management/ITIL Consulting, Service Desk process and procedures, Incident, Problem and Change Management, Business Impact Analysis, Asset Management, Network and Systems Management and Business Process Reengineering.

Mr. Locke is an ITIL Service Manager. Mr. Locke has also co-authored an ITIL Strategy book for international publication by Van Haren called "Service Management Process Maps." He is also currently a member of the Help Desk Institute (HDI) and the IT Service Management Forum (itSMF) and is a Certified Help Desk Director by HDI/STI Knowledge.

Charles Jimerson
Principal Services Architect

Charles Jimerson is a Principal Services Architect for CA's Global Practice and has over 25 years of experience in the information technology industry in a variety of technology management and technical roles. Charles's implementation experience includes roles in corporate, government, and financial organizations. During the past 10 years, Charles has been designing and implementing solutions based on an assortment of technology and integrations through CA's Enterprise IT Management (EITM) vision and capability solutions. Proficient in multiple programming languages and operating systems, Charles also maintains the Master Certified IT Architect Certification from the Open Group, the Certified Information Systems Security Professional (CISSP) certification with (ISC)_, and the Certified Information Systems Auditor (CISA) certification with ISACA.

CA (NASDAQ: CA), one of the world's leading independent, enterprise management software companies, unifies and simplifies complex information technology (IT) management across the enterprise for greater business results. With our Enterprise IT Management vision, solutions and expertise, we help customers effectively govern, manage and secure IT.

MP332711008