

CA GREEN BOOKS

CA CMDB Integrations Green Book



LEGAL NOTICE

This publication is based on current information and resource allocations as of its date of publication and is subject to change or withdrawal by CA at any time without notice. The information in this publication could include typographical errors or technical inaccuracies. CA may make modifications to any CA product, software program, method or procedure described in this publication at any time without notice.

Any reference in this publication to non-CA products and non-CA websites are provided for convenience only and shall not serve as CA's endorsement of such products or websites. Your use of such products, websites, and any information regarding such products or any materials provided with such products or at such websites shall be at your own risk.

Notwithstanding anything in this publication to the contrary, this publication shall not (i) constitute product documentation or specifications under any existing or future written license agreement or services agreement relating to any CA software product, or be subject to any warranty set forth in any such written agreement; (ii) serve to affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (iii) serve to amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this publication remain at CA's sole discretion.

The information in this publication is based upon CA's experiences with the referenced software products in a variety of development and customer environments. Past performance of the software products in such development and customer environments is not indicative of the future performance of such software products in identical, similar or different environments. CA does not warrant that the software products will operate as specifically set forth in this publication. CA will support only the referenced products in accordance with (i) the documentation and specifications provided with the referenced product, and (ii) CA's then-current maintenance and support policy for the referenced product.

Certain information in this publication may outline CA's general product direction. All information in this publication is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this document "AS IS" without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill or lost data, even if CA is expressly advised of the possibility of such damages.

COPYRIGHT LICENSE AND NOTICE:

This publication may contain sample application programming code and/or language which illustrate programming techniques on various operating systems. Notwithstanding anything to the contrary contained in this publication, such sample code does not constitute licensed products or software under any CA license or services agreement. You may copy, modify and use this sample code for the purposes of performing the installation methods and routines described in this document. These samples have not been tested. CA does not make, and you may not rely on, any promise, express or implied, of reliability, serviceability or function of the sample code.

Copyright © 2011 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. Microsoft product screen shots reprinted with permission from Microsoft Corporation.

TITLE AND PUBLICATION DATE:

CA CMDB Integrations Green Book v1.1
Publication Date: April 15, 2011

ACKNOWLEDGEMENTS

Principal Authors and Technical Editors

Enrico Boverino
Christy Druzynski
Rich Lankester
Randal Locke
Jerry Maldonado
Terry Pisauro
Brian Poissant
Kim Rasmussen
Vincent Scovetta
John Sorensen
Amy Spada
Dietmar Werner

The principal authors and CA would like to thank the following contributors:

Tunji Abiri
Anand Chauhan
Naveen Desham
Craig S. Guess
Ramy W Hassanein
Taylor Howe
Trevor Jones
Kishan G. Silva Ratnayake
Mohammed Rafiuddin
Robert Steiskal
Barry Stern

As well as the following teams:

CA Services
Development
Engineering Services
Marketing
QA
Support
Technical Sales
Technical Information

PRODUCT REFERENCES

This document references the following CA Technologies products:

- CA CMDB
- CA CMDB Connector for z/OS
- CA Service Desk Manager
- CA Cohesion[®] Application Configuration Manager
- CA IT Asset Manager
- CA NSM
- CA Service Catalog
- CA Service Accounting
- CA SPECTRUM
- CA IT Client Management

ITIL[®] is a registered trademark and a registered community trademark of the UK Office of Government and Commerce (OGC) and is registered in the U.S. Patent and Trademark Office.

FEEDBACK

Please email us at greenbooks@ca.com to share your feedback on this publication. Please include the title of this publication in the subject of your email response. For technical assistance with a CA Technologies product, please contact us at <http://ca.com/support>. For assistance with support specific to Japanese operating systems, please contact CA Technologies at <http://www.casupport.jp>.

DOCUMENTATION CHANGES

The following is a list of new chapters and new or revised topics in the April 15, 2011 update to this Green Book:

- Added note to Chapter 4: “Integrating with CA Cohesion” to clarify that, when removing the Model information from the cmdb_mapping.xml file you must also comment out the Manufacturer information. Failure to do so will cause an error.

The topics in this document apply primarily to CA CMDB r12.0 and r12.1. For information on later releases, see the CA Green Books page on Support Online for other resources.

Contents

Chapter 1: Introduction	11
Why Use a CMDB?	12
Before You Begin.....	14
Where to go for more information	14
Who Should Read This Book?.....	15
Chapter 2: General Integration Considerations	17
Understanding Configuration Items.....	17
What is an IT Service?	17
Where do CIs come from?.....	22
Loading the CI Data	23
Asset Federation and Federation Adapters.....	24
Common Object Registration API (CORA)	26
Master Asset Data Model.....	27
CMDB Tools.....	35
CA CMDB Visualizer.....	38
CMDBf Standard Support.....	38
Where to go for more information	40
Chapter 3: Using CA CMDB with CA Service Desk Manager	41
Overview and Value of the Integration	41
Integration Points and Functionality	42
Integration Points from CA Service Desk	42
Integration Points from CA CMDB.....	42
How the Integration Works	43
Example of the CA CMDB Integration	44
Business Challenge	44
CA Approach.....	44
Integration Walkthrough.....	44
Configuring the Solution	57
Reference Documentation	57
Chapter 4: Integrating with CA Cohesion ACM	59
Overview and Value of the Integration	60
How the CA Cohesion ACM Integration Works	60
Prerequisites	60
Launch CA Cohesion in Context to a CI in CA CMDB	61
Exporting CIs from CA Cohesion to the CA CMDB.....	64
Running the CA CMDB Export Report.....	64
To run or save a CA CMDB Export report	65
Tips for Modifying the CMDB Attribute Mapping Section.....	68
Tips for Modifying the CMDB Class Mapping Section	72
Managed Hardware Reconciliation across Multiple Domains.....	74
Reconcile Discovered Servers.....	76
Reconcile Relationships.....	78

Launch CA Cohesion in Context for a CI in CA CMDB.....	82
Important SSL Considerations.....	84
Create Certificate.....	84
Add Certificate to Java Trusted Key Store.....	84
Pass the URL to GRLoader.....	85
Reference Documentation.....	85
Chapter 5: Integrating with CA SPECTRUM	87
Overview and Value of the Integration.....	87
How the CA SPECTRUM Integration Works.....	89
Before you Begin.....	89
Installing the Integration.....	92
Prerequisites.....	92
CMDB Mapping File Modifications.....	93
CA SPECTRUM Customizations.....	94
Create Services in CA SPECTRUM First.....	94
Use Fully Qualified Domain Names (FQDN).....	95
CA SPECTRUM Integration Script.....	95
CA Service Desk Manager and/or CA CMDB Customizations.....	97
Defining the SPECTRUM MDR.....	97
Use GRLoader to Copy Data from Remote MDRs.....	100
Reference Documentation.....	102
Chapter 6: Integrating with CA Network and Systems Management (NSM)	105
Overview and Value of the Integration.....	105
CA NSM and CA CMDB Integration.....	106
CA NSM and Change Impact Analyzer.....	106
Steps to Configure Change Impact Analyzer.....	107
Modify pdm_startup.....	108
Create a New CIA Repository.....	109
CIA Classes.....	109
CIA Relationships.....	110
Importing Data from CA NSM.....	112
Synchronizing CA NSM and CA CMDB CI Status.....	114
Reference Documentation.....	116
Chapter 7: Other Integrations	117
Integrating with CA Service Catalog.....	117
Integration Points and Functionality.....	119
Integration Value.....	120
How the Integration Works.....	121
Business Challenge.....	122
Reference Documentation.....	123
Integrating with CA Mainframe Solutions.....	124
Integration Points and Functionality.....	124
Business Challenges.....	125
How the Integration Works.....	126
Health Check.....	129

Reference Documentation	129
Chapter 8: CA CMDB Architecture and Failover Considerations	131
Components	131
CA Service Desk Manager Solution.....	131
CA CMDB 12.1	131
Distributed Implementation	132
Database Server	133
Primary Server	133
Secondary Server.....	135
LDAP Server	136
Cohesion Server	136
Scalability	137
High Availability.....	137
Windows Environment with Microsoft Clustering	138
UNIX or Linux Environment	141
Reference Documentation	142
Glossary	143
Index	147

Chapter 1: Introduction

This document looks at the different products that CA CMDB can integrate with and how those integrations can be leveraged in solutions that address specific business challenges, such as Incident and Problem Management, Change and Configuration Management and the other Service Management processes. It is not intended as a replacement for existing product documentation, but, rather as a supplement to it.

Each chapter clearly identifies:

- What business objective the integration can meet – in other words, why would you want to use these products together
- How the integration works
- How to perform the integration – including prerequisites and additional configuration requirements
- Where to go for more information – additional documentation or other resources that can provide further details regarding the integration

Additionally, some chapters may include the following:

- Useful extensions – additional integrations or modifications that can provide further business value
- Troubleshooting – tips for diagnosing and resolving common errors

Why Use a CMDB?

A Configuration Management Database (CMDB) is part of a Configuration Management System (CMS). Its primary use is to help all of the other processes become more effective and efficient by providing a consistent set of managed information regarding Configuration Items (CIs) in the environment. When information regarding those CIs is provided by multiple sources, a CMDB enables those source applications to share a common service context which enables greater coordination across different technical teams within IT. For example, service desk technicians and operations engineering teams can collaborate on root cause analysis by sharing the same service definition and configuration across CA Service Desk and CA SPECTRUM.

One of the reasons for implementing a CMDB is to help the Change Management team understand the impact of a pending change prior to actually making the change. From an Incident Management perspective, understanding what HAS changed can also help speed up the troubleshooting process. Understanding which CIs actually link to services in the environment is also critical to managing true impact of an outage or change in the environment.

The CA CMDB provides a solid foundation for enabling organizations to improve their ITIL Processes, including minimizing the adverse impact of changes to production services within the infrastructure. The CA CMDB provides a powerful set of default content to enable organizations to “Start Smart” and begin getting value from a solution in a very short timeframe. This default content includes a pre-defined collection of Families and Classes which can be used to further classify Configuration Items (CIs). Even more powerful, is that each Family definition includes a set of standard attributes based upon the kind of CI involved. This helps provide consistency in tracking the appropriate attributes for CIs. This base set of attributes can also be extended based upon the specific needs of the customer.

Managing the relationships between CIs is the most critical task when creating a Configuration Management System (“CMS”) and thus the creation and management of a Configuration Management Database (“CMDB”). The CA CMDB provides many different default relationship types which can begin to provide the foundation for success in managing the CIs in the environment. Understanding these relationships is the key to understanding the potential impact or risk associated with a pending change. It also enables the Incident Management process to understand which resources are affected by an outage, as well as the potential impact of that outage, by leveraging information and relationships from within the CMDB itself.

Service Asset and Configuration Management (“SACM”) is the ITIL process which manages the CMS and the CMDB. This process determines the breadth and depth or scope of control for the CMS. Once this is established you must place strict change control over the CIs within the CMDB to ensure that valid information is being leveraged for decisions within the organization. The CA CMDB provides a mechanism to assist with the management of necessary CI information federated from sources called “Management Data Repositories (MDRs)” or trusted sources of information. These MDRs provide the necessary attributes and relationships which are needed to manage CIs. The CMDB Federation (“CMDBF”) working group is helping to determine integration

specifications for leveraging cross CMDB integration points so that one vendor's CMDB information can be populated to another CMDB if necessary.

Visualization is also critical to the ease of consumption and understanding of CMDB information. Having the ability to visually navigate the inter-relationships between CIs and, more importantly, the type of relationship, is crucial to resolving and preventing incidents from occurring based upon a change in the environment. The CA CMDB provides a very robust visualization mechanism that allows you to drill down into more detailed information for a particular CI, as well as to set up filters to easily direct focus based upon a particular need. For example, by default the CA CMDB provides a filter for "Root Cause Analysis" which looks at related CIs to see what could have caused an outage to a particular service or CI. The "Impact Analysis" filter, on the other hand, analyzes these relationships to determine how an outage in one or more CIs might potentially impact a particular CI or group of CIs which are related to it. Each user also has the ability to create and save custom filters. For example, a "Business Services" filter could be used to identify services that may be affected by an outage to a particular CI and a filter for "Organizations" or "Locations" could be used to identify CIs that could be impacted by an outage to a particular CI. Additional CMDB Visualizer capabilities include the ability to create new CIs, relationships and even launch in context from the Visualizer back into the MDRs that are providing information to the CMDB. This can significantly reduce the amount of time it takes to resolve an incident by giving your analysts convenient access to all available information.

Managing the ever changing attributes of a CI can be a very daunting task. Therefore, CA CMDB has the ability to track all attribute and relationship changes that occur to any CI as well as incorporate the authorized changes that are associated to the CI in a single place within the CI Detail screen. Remember that a CMDB is set to manage the "should-be" state not the "as-is" state of a CI. The Versioning tab allows the analyst to see what attribute or relationship has changed as well as who made the change and when, including those changes that may be automated from MDRs. This assists with the Audit and Verification stages of the process within SACM.

Another powerful feature of the CA CMDB is that it is part of the CA Service Desk Manager solution set which includes the CA Service Desk for managing incidents, problems and changes, and the release and deployment management processes within your environment. Each function within the CA Service Desk Manager solution set has the ability to see what is occurring in the other functions. Thus, if you look at the CI detail screen for a particular CI you have the ability to see its associated incidents, problems or changes.

Finally, with the CA CMDB, if a component is being tracked, you can choose whether what is being tracked is actually a CI, an Asset, or both. This helps the Incident Management team see all Assets and CIs and helps the Change Management team only manage changes associated with CIs, thus increasing the effectiveness of each functional group.

Before You Begin

A well thought out implementation plan is the key to achieving your objectives for deploying CA CMDB. This means that, prior to installing the actual product and any of its integrated solutions, you need to identify what business services you need to support, which CIs are needed to provide those business services and where/how they are maintained. You will also need to identify and implement the appropriate processes regarding how the CI information is kept current and how it is used. Although product documentation can include some general recommendations along with the procedural details, it is highly recommended that you engage CA Services to assist in developing a design that is best suited for your specific environment and business objectives.

For information regarding the terminology and acronyms used in this document consult the Glossary provided.

Where to go for more information

For additional information consult the following documents which are provided with the CA CMDB product:

- *Administrator Guide* which identifies how to:
 - manage the CMDB servers
 - define the business structure and infrastructure
 - implement security policies
 - set up users
 - establish a support structure
 - use the MDR (Management Data Repository) Launcher
 - control system behavior through options and environment variables
 - create objects through text API
 - configure the web interface
 - manage the database
 - and also include a command reference, field description and LDAP result codes appendices
- *Implementation Guide* which documents the following:
 - Installation of CMDB, secondary server, and optional features
 - Data population
 - MDR Launcher – including Cohesion integration
 - NSM integration

- Implementing change impact analyzer
- Portal installation
- *Technical Reference Guide* which documents the following:
 - Default Families and Classes
 - Using GRLoader
 - Using ADT to create a Federation Adapter
 - Using CMDBf web services

Further details can be found in the following documents which are available through the CA Support Website:

- *Incident and Problem Management Green Book*. This guide currently includes chapters on the CA CMDB + CA Service Desk integration – both general and CA specific examples.

Who Should Read This Book?

This book provides the IT consultant, architect, or systems manager with guidance on and examples of how CA CMDB can be integrated with different solutions to effectively resolve key business challenges, such as change management and root cause analysis.

Implementation success is based on a combination of people, process, and technology; and this wide-ranging book provides process, technical, and architectural best practices. Information on key topics includes support models and best practices, use of common fields and architecture.

Readers of the more technical areas in this book will benefit strongly from some prior familiarity with the primary products. Therefore, readers are encouraged to make use of the standard product documentation and to attend the relevant CA Education courses. For more information on available courses, see the Education link on the www.ca.com website.

Chapter 2: General Integration Considerations

This chapter identifies the basic concepts behind integration with the CA CMDB – such as:

- how information is imported into the CA CMDB from other products
- how asset information is reconciled through the Common Object Registration API (CORA)
- how to launch back into the source product’s user interface in the context of the CI through the MDR Launcher

Understanding Configuration Items

The most fundamental element of the CA CMDB is the *configuration item (CI)* and the most powerful function of the CMDB is its ability to clearly demonstrate the relationships between the CIs that comprise a business or IT service. Understanding this enables you to both drill down to identify the root cause for a problem and zoom out to identify the broader potential impact of a change to any particular CI.

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 can be or even should be discovered. One of the most important ground rules in designing an efficient CA CMDB is to resist the temptation to populate the CMDB with all data within the enterprise. Only those CIs and relationships that satisfy specific business requirements and that you are willing to manage and maintain should be included in the CMDB.

The following steps represent a best practice for building a CMDB.

1. Define IT Services
2. Identify the MDRs which provide source CI details for those IT Services
3. Implement the appropriate provider tools
4. Collect CI and relationship details
5. Import data into the CA CMDB
6. Verify and audit this information.

What is an IT Service?

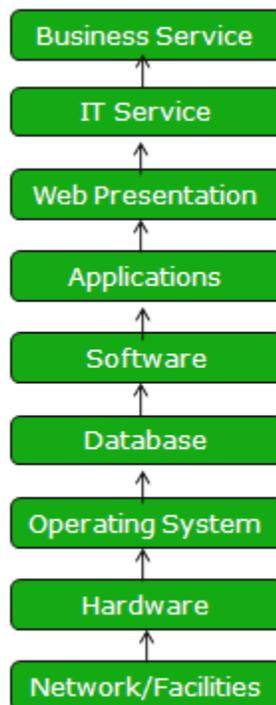
Before you start identifying CIs you need to carefully consider what comprises an IT Service. An IT Service can be thought of as “a set of related components provided in support of one or more business processes.” An IT Service can represent any process or a group of processes that you have identified 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. For example, this may include any of the following:

- a web-based retail transaction service
- an application server service
- a printing service
- an email service
- a routing service
- a source control service

There are many other examples of IT-based services that you can model as IT Services, however, your choice of what to model as a service should be based on the priorities of your business.

IT services can be modeled using a hierarchical approach consisting of multiple layers, each representing a collection of conceptually similar CIs that provide services to the layer above it and receive services from the layer below it. For example:



Here you can see that a Business Service relies on an IT Service which, in turn, uses a web presentation, which itself requires several software applications and so on. In this way you can see that the loss of a particular layer can impact the layers above it. For example, if there is a problem with the network, this may impact the hardware it supports, the operating system that runs on that hardware, including the database that is used by a software program and, eventually impact the ability to provide the business service on the top layer of this structure.

Other layers might include:

- Organizations
- Customers
- Locations
- Service Level Agreement (SLA)
- Application Development
- Virtual Environment
- High Availability
- Security

However, keep in mind that, for each additional layer you use you should be prepared to manage additional CI types, attributes and relationships in the CMDB as well. Your objectives for using the CMDB, as well as the type of consumers of the CMDB data will determine which layers become necessary and what can be captured as CI attributes for each layer.

Once you have identified the layers, the next step is to decide what types of CIs belong to each layer. CA CMDB classifies CIs with *Families* and *Classes* and provides several default definitions for both. These defaults can either be used as provided or modified as needed. With this information in mind you can begin constructing your CMDB.

CI Families

CI families are usually used to categorize a particular CI in a very general sense. Consider the following list of Families and Classes included in the CA CMDB r12.1 release:

- Cluster
- Cluster.Resource
- Cluster.Resource Group
- Computer
- Contact
- Contract
- Document
- Enterprise Service
- Enterprise Transaction
- Facilities.Air Conditioning
- Facilities.Fire Control
- Facilities.Furnishings
- Facilities.Other
- Facilities.Uninterruptible Power Supply
- Hardware
- Hardware.Logical Partition
- Hardware.Mainframe
- Hardware.Monitor
- Hardware.Other
- Hardware.Printer
- Hardware.Server
- Hardware.Storage
- Hardware.Virtual Machine
- Hardware.Workstation
- Investment.Idea
- Investment.Other
- Investment.Project
- Location
- Network.Bridge
- Network.Controller
- Network.Frontend
- Network.Hub
- Network.Network Interface Card
- Network.Other
- Network.Peripheral
- Network.Port
- Network.Router
- Network.Switch
- Organization
- Other
- Projects
- SAN.Interface
- SAN.Switch
- Security
- Services
- Service
- Service Level Agreement
- Software
- Software. Application
- Software.Application Server
- Software.Bespoke
- Software.COTS
- Software.Database
- Software.In-House
- Software.Operation System
- Telecom.Circuit
- Telecom.Other
- Telecom.Radio
- Telecom.Voice
- Telecom.Wireless

CA CMDB r12.1 Families and Classes

CI Classes

CI classes provide a more specific classification and are used to further categorize Configuration Items within a single CI Family. For example, the “Hardware.Storage” family includes the following classes:

- CD-Rom Drive
- Disk Array
- DVD
- File System
- Hard Drive
- Network Attached Storage
- Optical
- Other Hardware Storage
- Silo
- Storage Area Network
- Tape Array
- Tape Library
- Virtual Tape System
- Zip Drive

For a complete list of families and classes supported by the CA CMDB, consult the *CA CMDB Technical Reference Guide*.

CI Attributes

Each CI Family is associated with a specific set of attributes which enable a clear identification of the information that is relevant to that type of CI. For example, a Service Level Agreement would have attributes such as “Expiry Date” and “Owner”.

Attributes are stored in extension tables linked to the Family.

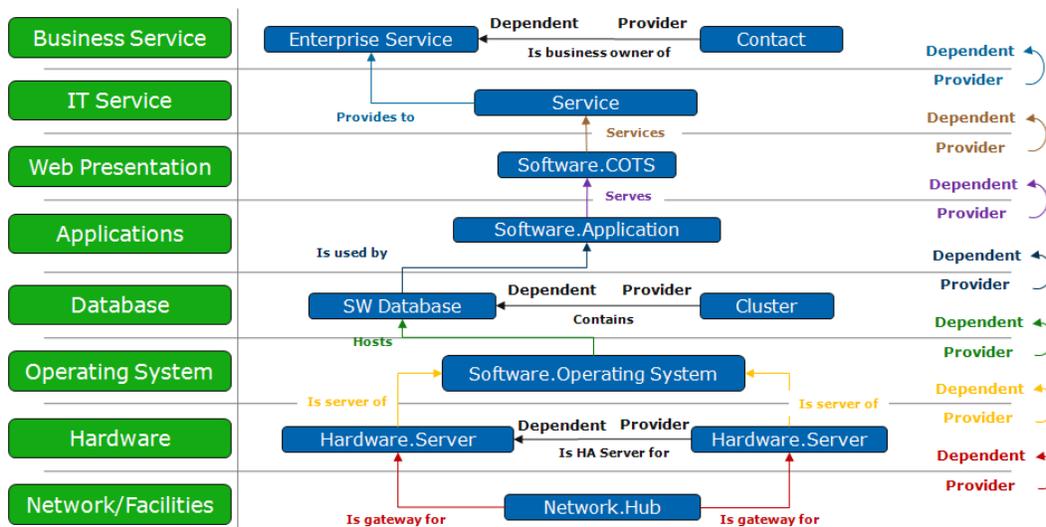
Determining which attributes to capture and maintain in the CMDB is done during the modeling of the Service and that determination should be based on the objectives of the CMDB initiatives.

CI Relationships

The next step is to understand the dependencies between each layer of the Service Model as well as the *relationships* between CIs in each layer.

CA CMDB includes an extensive list of pre-defined Relationship Types that can be used to facilitate the design of the CMDB. Relationship Types represent the specific classification of a relationship between two CIs. Each Relationship can be labeled to represent the logical roles of the two CIs in the organization’s Service Model where each CI contributes to the Service, by **providing** a “service” to a **dependent CI**.

Following is an example of an IT Service model that includes the CA CMDB CIs, Families and Relationship Types that exist between Layers and within each Layer:



Note: For a full list of the default Families, Attributes, Classes and Relationship Types included with the CA CMDB consult the *CA CMDB Technical Reference Guide*.

Where do CIs come from?

After you have identified your IT Services and the CIs that comprise them you need to identify the appropriate *Management Data Repositories* (MDRs.) An MDR is simply any source which provides data on a CI. Most IT organizations have no shortage of MDRs and it is important to identify which of those MDRs represents the authoritative source of information for your CIs, potentially down to the CI attribute and relationship level per CI type. Every CI in the CMDB can be associated with one or more MDRs.

For example, a backup MDR may contain database information, while another MDR may contain information about systems and applications, and still others may contain information about the network infrastructure. Some MDRs function as “mini-CMDBs” that delineate services and relate them to CIs. Other MDRs are less sophisticated and may require you to define the services in the CMDB first, then link them to CIs federated from the MDR.

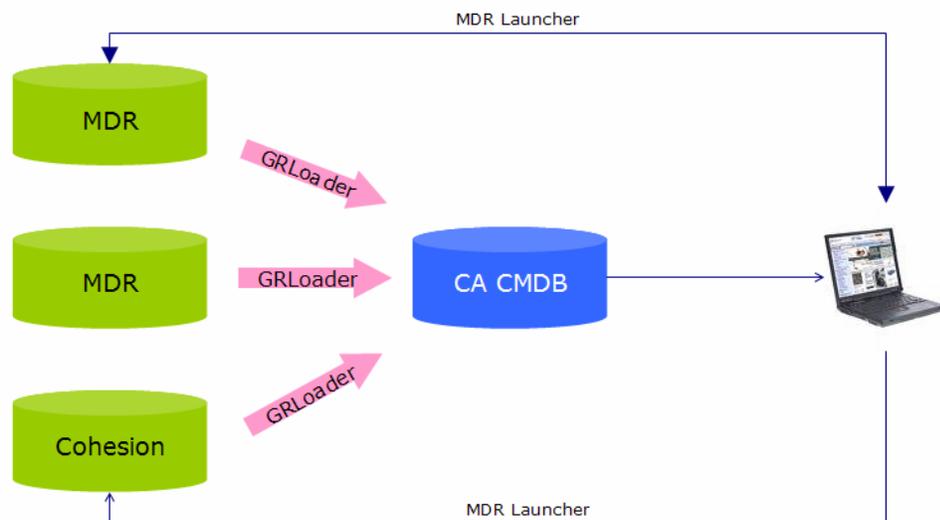
Deciding which MDRs to use, as well as how to use them, should be driven by the IT services you want to manage and the quality of the data that it offers. Do not make the mistake of using an MDR simply because it is there. A federated MDR must contribute value to the CMDB – not just volume.

When new MDRs are identified, establishing CI types can be a concern. Although discovered CIs must be classified into types, the “object types” used by the MDR may not necessarily correspond to CI families and classes used by the CMDB. Therefore, type mapping is one of the first activities performed when a new MDR is federated.

Type mapping is done 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 CI type used in the MDR is transformed into a CA CMDB CI type during transfer from the MDR to the CMDB. In most cases, types used in the MDRs can be mapped to existing CMDB CI types as defined above.

Loading the CI Data

CI data is loaded into the CA CMDB through the *General Resource Loader* (GRLoader) utility.



Once the data is in the CMDB, you can then use the *MDR Launcher* to access the source MDR from within the CA CMDB web interface. This enables you to obtain further information regarding that CI within the context of the source program.

Asset Federation and Federation Adapters

The same CI may be known to different systems by different names much the same way you may be known to different groups through your nickname, driver's license card, employee ID, or health insurance number. Although each of these IDs points back to the same individual – you – their relevance is specific to a particular context. For example, your friends are likely to refer to you by your nickname while your doctor's billing office requires your health insurance details and may even apply a file number that is specific to their billing programs.

Similarly, although a CI may be known to several different MDRs, for each of those MDRs it is known by a single identifier. This identifier is known as the *federated asset ID*. The process of associating a CI with one or more MDRs is known as *mapping the CI*. CI mapping can be performed in either of two ways:

- Through the CA CMDB Administrative user interface
- Through the GRLoader utility

One of the main functions of CA CMDB is to aggregate data regarding CIs of interest related to one IT configuration in a single place. Almost all IT organizations have data about the configurations in separate MDRs. For example, an organization may have a network and systems management system MDR, an identity management system MDR and/or an application performance management system MDR. CA CMDB can import CI information from all of these sources and preserve the link from the imported data back to its source MDR.

An *MDR provider* is a CA CMDB object that allows CIs to be associated with an external MDR. MDR providers define the callback mechanism used to launch an MDR's web-based User Interface in context. CIs which have been imported from a configured MDR are automatically mapped back to their source. An MDR also can be configured manually to map a CI to a federated web-based application. You only need to know the CI's federated asset ID (by which the MDR knows the CI).

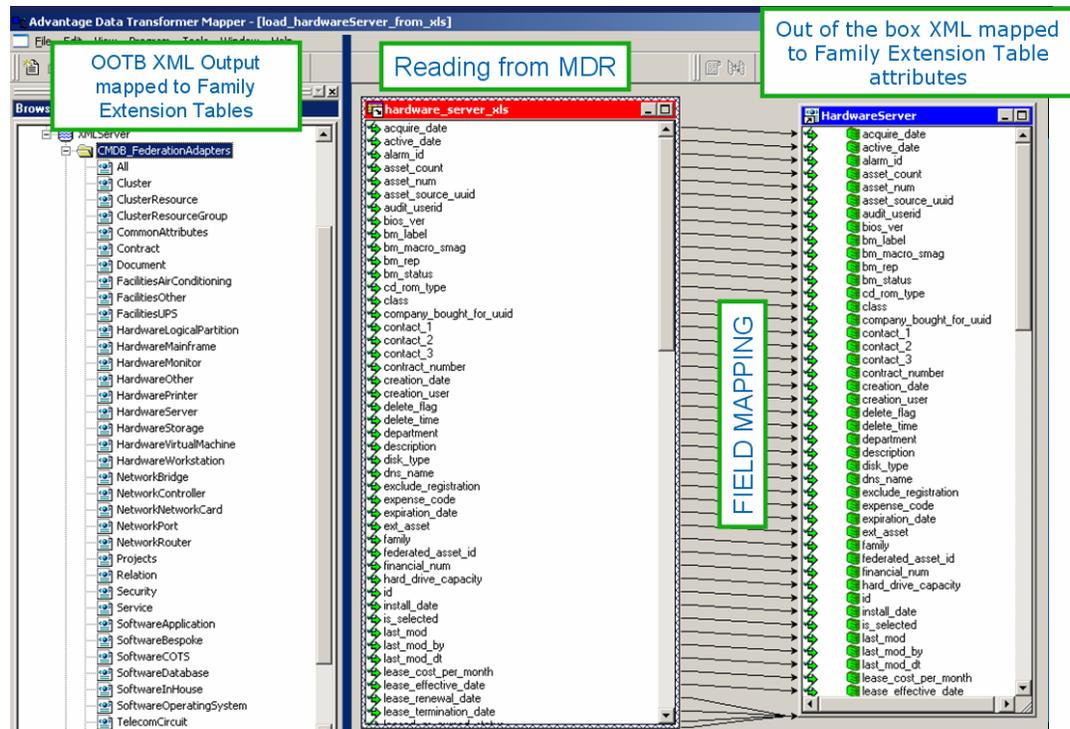
Federation Adapters are programs that provide a convenient way for you to include CIs from practically any data source into CA CMDB. With CA CMDB we include several Federation Adapters created using Advantage Data Transformer (ADT) programs. There are two kinds of Federation Adapters: default Federation Adapters provided with CA CMDB and Custom Federation Adapters which may be customized to meet special needs. Default Federation Adapters include the following:

- **Load_ci_from_xls** -- Load any family of CIs from an Excel spreadsheet.
- **Load_generic_template** -- Load any family of CIs from any data source.
- **Load_hardware_server** -- Load hardware server CIs into the MDB.
- **Load_relations_from_xls** -- Load a table of CI relationships from an Excel spreadsheet.
- **Load_SMS_from_view** -- Load Microsoft SMS data from a view created at the source database.

- **Load_UAM_from_view** -- Load Unicenter Asset Management Data from a view created at the source MDB.

The Federation Adapters expose the default CA CMDB (MDB) data schema through XML files (Outputs) that can be used to map the information extracted from any MDR, based on the desired Family.

Here you can see an example of the XML Output installed with CA CMDB and how it is aligned with the default Families and Extension Tables which defines the Attributes for each CI.



ADT Federations Adapter showing mapping between an MDR and CA CMDB Family

On the right side of this example you can see one of the pre-configured Federation Adapters used to load Server CI information from a CSV formatted file mapped to the XML Output of the Hardware.Server Family. If you create new Families or add Attributes to existing Family extension tables, you also need to adapt the Federation Adapter's XML output files to expose the new fields.

The results of this operation will be an XML file that will be used by GRLoader to load/update CI information in the Management Database (MDB) that is used by CA CMDB.

The same approach and logic apply to mapping Relationships between CIs. Details on using Federation Adapters and on creating custom adapters is provided in the *CA CMDB Implementation Guide*.

Common Object Registration API (CORA)

CA CMDB data is stored in the Management Database (CA MDB) where it can be shared with the other CA Products that connect to and are registered to that MDB. Many CA r11 and CA r12.x products utilize the common MDB schema to store and manage their data. CIs can be added to the MDB through a variety of sources including discovery tools, such as CA IT Client Manager (CA ITCM) and CA Network and Systems Management (NSM), or ownership tools, such as CA Service Desk Manager or CA IT Asset Manager. Even though a CI can be discovered by multiple products, the MDB asset schema is designed to reconcile the fact that a CI coming from different sources is actually the same CI. This is accomplished through CI registration using the Common Object Registration API – or “CORA”.

As the interface through which these CIs are registered and as the only source for updating these tables, the CORA ensures that CI data flows consistently, thereby supporting the data and referential integrity of the MDB's master CI data model.

The MDB asset schema provides a set of common asset tables for hardware and software CIs that allows for a cross-product view of CIs and meets the following requirements:

- Supports discovery of an asset by multiple sources (for example, CA NSM and CA IT Client Manager)
- Supports discovery of multiple values of an identifying CI Attributes (for example, multiple DNS names and/or MAC addresses)
- Enables multiple “virtual” assets to be discovered and reconciled to one physical CI (for example, VMware images or dual-boot scenarios)

Each of these products using the MDB leverages CORA to register a CI when it comes into view using the product's particular operations. For example, when CA NSM or CA ITCM “discovery” occurs, each discovered CI is registered. Similarly, when a CA Service Desk Manager or CA IT Asset Manager user enters CI information using the data entry forms of those products, that CI is also registered.

Consider the following example: After an initial first-level discovery through CA NSM Continuous Discovery, a server CI is registered with the identifying properties known to CA NSM, typically DNS name and MAC address. Later, CA IT Client Manager performs an asset scan for that same server and also registers the CI, also using the DNS name and MAC address, plus additional identifying attributes. Since the DNS name/MAC address pair matches the previously registered asset, the information held by CA NSM is now effectively joined to the information held and managed by CA IT Client Manager.

Master Asset Data Model

The MDB master asset data model consists of the following 3 levels of references:

- The **source level**, which consists of the `ca_asset_source` table, is used to track CIs as they enter the system from different data sources, whether input manually or through discovery.
- The **logical level**, which consists of the `ca_logical_asset` and `ca_logical_asset_property` tables, is used to store virtual CIs. The logical CI level acts as a middle layer that exists between the data source and the physical level to accommodate CIs embedded in other CIs such as VMware sessions or dual-boot scenarios.
- Finally, the **physical level**, which consists of the `ca_asset` table, stores the identifiers that define the object as a distinct, physical CI.

After CORA is given a set of registration identifiers from the calling application, it performs one of the following actions:

- Returns the source reference if the registration identifiers match an existing CI, thus preventing duplicate CIs from being registered.
- Inserts a new physical, logical, logical property, or source record into the database depending on where the mismatch occurs. This step also prevents duplication of data by inserting records only at the appropriate levels. For instance, if there are no physical CIs that can be identified by the registration identifiers, a new physical CI is created. However, if a physical CI can be identified by the registration identifiers, but not a logical CI, then a new logical CI is created and linked to the existing physical CI.
- Updates an existing identifier(s) in the database with one of the registration identifiers. In this scenario, a single physical CI can be identified by the registration identifiers and one or more identifiers need to be updated.
- Merges two physical or logical CIs together. This occurs when CORA receives information indicating that two or more physical CIs are, in fact, the same CI. The existing physical CIs are merged together to form one CI and information for each CI is stored in `ca_logical_asset_property` table.

For r11.x or 12.x when a product registers a CI and CORA generates a Universally Unique Identifier (UUID) that matches an existing CI, CORA also automatically links (reconciles) Owned and Discovered information for that CI.

To determine which CORA version is being used by the product, execute the following command

```
coraver
```



Asset Matching Logic

When a CI is registered, CORA generates the asset UUID (ca_asset) by applying black-box logic to the following six properties:

- Serial Number
- Asset Tag (appearing as Alt Asset ID)
- Host Name
- MAC Address
- DNS Name
- Asset Label (Name)

CORA applies the following weighting system to these properties to determine if a match exists. Since certain properties are considered “more important” than others, CORA recognizes a duplicate based on those values alone.

- Serial Number is the most highly weighted field. Two CIs with the same serial number are always matched by CORA unless the Asset Tag or Host Names are different.
- Alt. Asset ID, which for hardware CIs represents the Asset Tag attribute, is the second most highly weighted field. Serial Number and Alt. Asset ID appear at the highest level of the CI registration schema in ca_asset. If Serial Number and Asset Tag match, CORA can create a new CI only if the Host Name is unique.
- Host Name appears in the middle level (ca_logical_asset). If Serial Number and Alt. Asset ID are blank the Host Name takes precedence over DNS Name and MAC Address values. Although more than one DNS/MAC pair can be specified for the same Host Name, it is still considered to be the same CI.
- DNS Name and MAC Address are weighted the same. CORA will recognize the same CI if DNS Name or MAC Address match and will create a new CI when they do not.
- Finally, although Asset Label (Name) is required to create a CI, you can have multiple CIs with the same name as long as all the other CORA fields are empty.

The following matrix demonstrates the logic used by CORA:

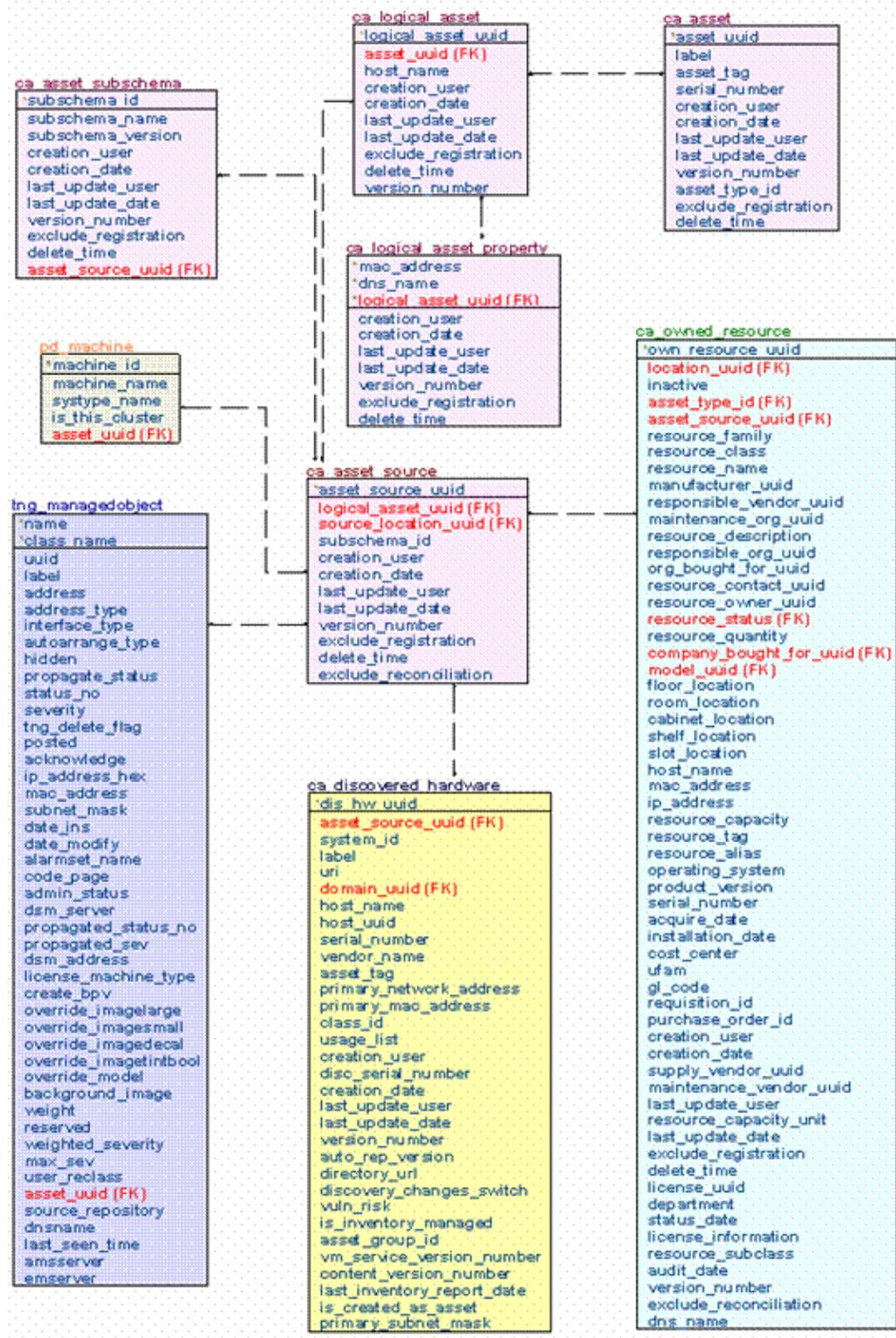
Serial Number	Alt Asset ID	Host Name	DNS Name	MAC Address	Asset Label	CORA Results
Unique	Unique	Unique	Unique	Unique	Unique	New Asset Created
Unique	Duplicate	Duplicate	Duplicate	Duplicate	Duplicate	New Asset Created
Duplicate	Unique	Duplicate	Duplicate	Duplicate	Duplicate	New Asset Created
Duplicate	Duplicate	Unique	Duplicate	Duplicate	Duplicate	New Asset Created
Duplicate	Duplicate	Duplicate	Unique	Unique	Unique	Recognized as Duplicate Asset
Null	Null	Null	Unique	Duplicate	Duplicate	New Asset Created
Null	Null	Null	Null	Unique	Duplicate	New Asset Created
Null	Null	Null	Null	Null	Unique	New Asset Created
Null	Null	Null	Null	Null	Duplicate	Recognized as Duplicate Asset
Null	Null	Null	Unique	Unique	Unique	New Asset Created
Null	Null	Null	Duplicate	Unique	Unique	New Asset Created
Null	Null	Null	Unique	Duplicate	Unique	New Asset Created

Discovered and Owned Assets:

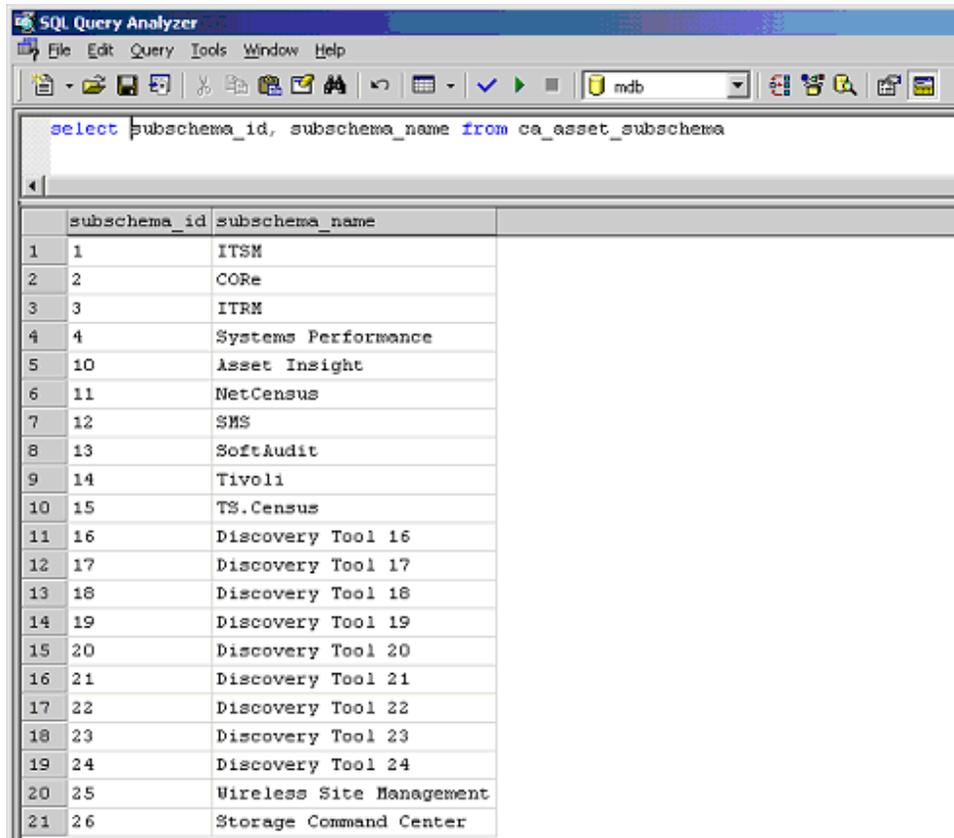
Attributes for each CI are divided into “Discovered” and “Owned” in order to facilitate reconciliation and verification capabilities. The primary tables used to identify data sources are:

- CA_DISCOVERED_HARDWARE
- TNG_MANAGEDOBJECT
- PD_MACHINE
- CA_OWNED_RESOURCE

To understand how these tables relate to one another, consider the following graphic.



The ca_asset_source table contains the subschema_id column which identifies the origin of the CI. The subschema_id values are maintained in ca_asset_subschema as shown with the following query:



The screenshot shows the SQL Query Analyzer interface. The query entered is: `select subschema_id, subschema_name from ca_asset_subschema`. The results are displayed in a table with the following data:

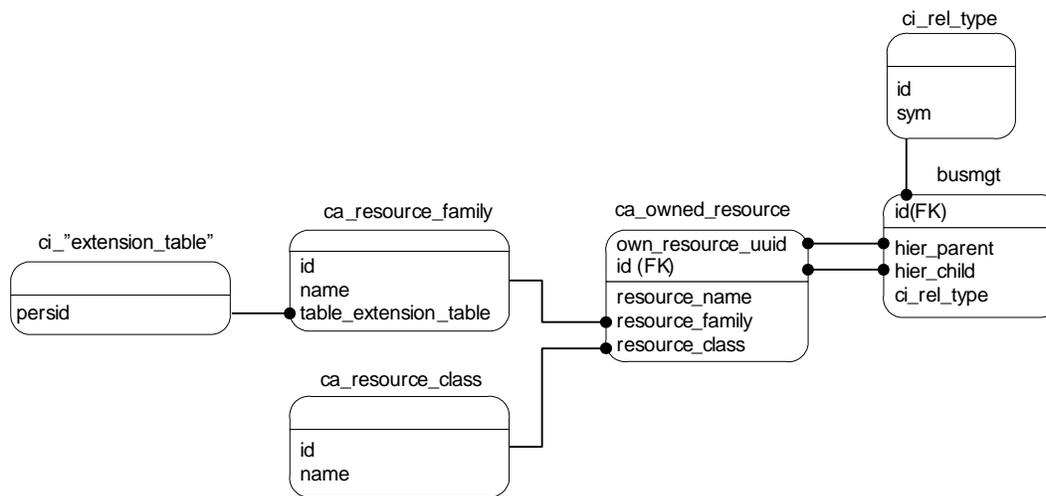
	subschema_id	subschema_name
1	1	ITSM
2	2	CORe
3	3	ITRM
4	4	Systems Performance
5	10	Asset Insight
6	11	NetCensus
7	12	SMS
8	13	SoftAudit
9	14	Tivoli
10	15	TS.Census
11	16	Discovery Tool 16
12	17	Discovery Tool 17
13	18	Discovery Tool 18
14	19	Discovery Tool 19
15	20	Discovery Tool 20
16	21	Discovery Tool 21
17	22	Discovery Tool 22
18	23	Discovery Tool 23
19	24	Discovery Tool 24
20	25	Wireless Site Management
21	26	Storage Command Center

The subschema_name column contains shortcut descriptions that refer to CA products or Third Party products. Those of immediate interest and their associated primary tables are:

- “ITSM” (IT Service Management) objects, which include “Owned” sources such as CA IT Asset Manager, CA Service Desk Manager, and CA CMDB (ca_owned_resource), have a subschema_id of “1”
- “CORE” (CA NSM WorldView Repository - Common Object Repository) or CA NSM (tng_managedobject) objects have a subschema_id of “2”
- “ITRM” (IT Resource Management) or CA ITCM objects (ca_discovered_hardware) have a subschema_id of “3”
- Systems Performance Management objects (pd_machine) have a subschema_id of “4”

If a CI is registered in the MDB by different products, CORA only registers that CI once then links the information from the different data sources. As a result the ca_asset table has a single unique entry for each CI.

Here you can see the CA CMDB specific tables that are introduced to populate the CA CMDB Content of Families, Classes, Attributes and Relationships Types:



The following screenshots provide a walkthrough of the queries executed after a sample CI is registered by CA IT Asset Manager/CA Service Desk Manager, CA NSM, and CA ITCM. Note that the order in which the products register the CI is not relevant to the process.

First, from Machine name into ca_asset and then, from ca_asset into ca_logical_asset (using asset_uuid):

```

Query
select * from ca_logical_asset where asset_uuid in (
select asset_uuid from ca_asset where label like '%Server1%')
    
```

	logical_asset_uuid	asset_uuid	host_name
1	0xECE60606A217C74D9E8D7COD836901D8	0xE7FC0779EE22424389C28A30FD23D607	Server1-topgun

The ca_logical_asset_property shows the logical instances of the same CI. For instance, if the same CI is registered by CORA with different DNS Names and/or MAC addresses but the same Host name, CORA recognizes it as the same CI and stores two logical instances in this table:

```

Query
select * from ca_logical_asset_property where logical_asset_uuid in(
select logical_asset_uuid from ca_logical_asset where asset_uuid in (
select asset_uuid from ca_asset where label like '%Server1%'))
w
    
```

	dns_name	mac_address	logical_asset_uuid	creation_user	creation_date
1	Server1-TOPGUN	000C295ACD6A	0xECE60606A217C74D9E8D7COD836901D8	NULL	1160141977
2	Server1-topgun.ca.com	000C295ACD6A	0xECE60606A217C74D9E8D7COD836901D8	NULL	1160141840



Note: In this example the DNS Name input by CA IT Asset Manager did not use the fully qualified name as it was discovered by CA NSM. It is only an example.

Then, from ca_logical_asset to ca_asset_source (using logical_asset_uid):

```

select * from ca_asset_source where logical_asset_uid in (
select logical_asset_uid from ca_logical_asset where asset_uid in (
select asset_uid from ca_asset where label like '%Server1%'))

```

	asset_source_uid	logical_asset_uid	source_location_uid	subschema_id
1	0xD93E456B08FE604C9D8461E13E94DC55	0xECE60606A217C74D9E8D7C0D836901D8	NULL	1
2	0xED4DABA4834CDD4FBAB89D8BB31FDE01	0xECE60606A217C74D9E8D7C0D836901D8	NULL	2
3	0xEE19F52014604C4D8ED5C211C10CFCCA	0xECE60606A217C74D9E8D7C0D836901D8	NULL	3

Note: Here you can see the different data sources from the subschema_id value.

Then, from ca_asset_source into CA ITCM ca_discovered_hardware (using asset_source_uid):

```

select * from ca_discovered_hardware where asset_source_uid in (
select asset_source_uid from ca_asset_source where logical_asset_uid in (
select logical_asset_uid from ca_logical_asset where asset_uid in (
select asset_uid from ca_asset where label like '%Server1%'))))

```

	dis hw uuid	host_name	domain uuid	label	serial number
1	0xEE19F52014604C4D8ED5C211C10CFCCA	Server1-topgun	0x041D21D1F5607047819DA5EAD0162A67	Server1-topgun	56dd501

Then, from ca_asset_source into CA IT Asset Manager/CA Service Desk ca_owned_resource (using asset_source_uid):

```

select * from ca_owned_resource where asset_source_uid in (
select asset_source_uid from ca_asset_source where logical_asset_uid in (
select logical_asset_uid from ca_logical_asset where asset_uid in (
select asset_uid from ca_asset where label like '%Server1%'))))

```

	own_resource_uid	inactive	asset_type_id	resource_name	resource_description
1	0x8CA49B2CEDD9FA4D9D3D5345002B3FEC0	0	1	Server1-topgun	NULL

Then, from ca_asset_source into CA NSM tng_manageobject (using asset_source_uid):

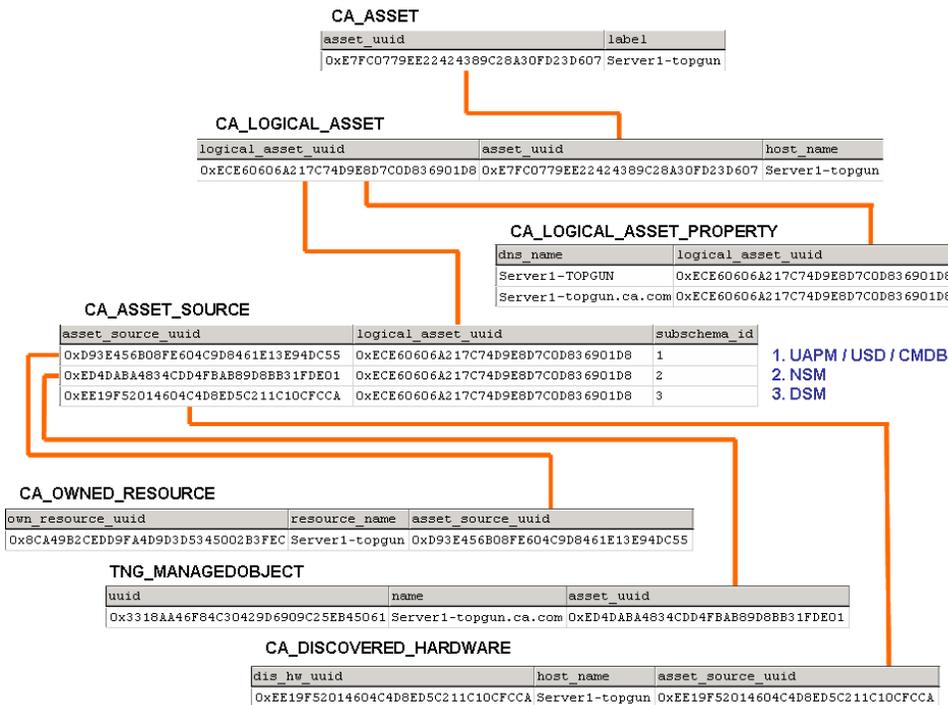
```

select uuid, label, address, asset_uid from tng_managedobject where asset_uid in (
select asset_source_uid from ca_asset_source where logical_asset_uid in (
select logical_asset_uid from ca_logical_asset where asset_uid in (
select asset_uid from ca_asset where label like '%Server1%'))))

```

	uuid	label	address	asset_uid
1	0x3318AA46F84C30429D6909C25EB45061	Server1-topgun.ca.com	130.119.20.221	0xED4DABA4834CDD4FBAB89D8BB31FDE01

These links can be summarized in the following picture:



Note that, in this example, CA ITCM is represented as “DSM”, for “Desktop and Server Management” – the name previously given to the CA IT Client Manager (ITCM) product.

Note about CI and Asset Classes

Although “class” is a mandatory field for CI registration, it is not a field that is used by CORA. The concept of “class”, in fact, is interpreted differently by different products. For example:

- In CA Service Desk Manager, the concept of “family” is used to identify the highest level of definition for a CI and each family can consist of one or more “classes” to allow for a more granular categorization of CIs. Further, each family has an extension table that defines the attributes that are visible in the CI Detail page. When CA CMDB is implemented, it includes over 50 families and over 140 classes that are each stored in the MDB and shared between CA Service Desk Manager and CA IT Asset Manager.
- When the MDB used by CA CMDB is shared with CA IT Asset Manager, those CMDB families are shared and are known to CA IT Asset Manager as “asset types” for “models” and “assets”. In other words, for CA IT Asset Manager:
 - CMDB Families = CA IT Asset Manager Asset Types
 - CMDB Classes = CA IT Asset Manager Classes
- In CA IT Asset Manager, the Asset Type is stored in the family_id field of the ca_model_def table and in the resource_family field of the ca_owned_resource table of the MDB.



- CA ITCM, on the other hand, does not use families and classes to register discovered assets. However, when CA Service Desk Manager is also installed and integrated with CA ITCM, if CA ITCM initiates the creation of a Service Desk ticket and, as such, the registration of a discovered asset as “owned,” too, (in order for Service Desk to link the ticket to it) it uses the default “Hardware” family and “Discovered Hardware” class.

Note: If you need to modify the family and classes used by ITCM, we recommend using the default ADT Federation Adapter for ITCM (**Load_UAM_from_view**)

- CA CMDB content creates new families and classes. However, these classes are not the same classes that are used by CA NSM to classify discovered objects. In fact, only a small number of CA NSM classes match CA CMDB classes. Procedures for mapping CA Service Desk/CA CMDB classes to CA NSM classes are provided in the *CA CMDB Administrator Guide*.

Note: Since multiple CA Service Desk/CA CMDB classes can be mapped to the same CA NSM Class, the `pdm_nsmimp` can not use the CA NSM class to determine which CA Service Desk/CA CMDB Class to use when creating the asset because it doesn't know how to select the correct one if multiple ones are mapped. Also in this circumstance we recommend the use of ADT and Federation Adapters to create the required mapping.

The full schema for the MDB is viewable through the Implementation Best Practices page (formerly the “r11 Implementation CD”) which is available on <http://ca.com/support>. The direct link to the MDB Schema Viewer is:

<https://support.ca.com/phpdocs/0/common/impcd/r11/MDBMain/schema/viewer/index.htm>

CMDB Tools

As previously noted, the MDR Launcher is a mechanism that enables you to view the source MDR’s browser based interface in order to obtain additional information regarding a particular CI. This can be particularly useful since the MDR typically contains more detailed information regarding the CI that it manages. In general, the CMDB is not intended to be a data warehouse for *all* CI attributes – rather, the intent should be to consolidate the *most important* attributes that require central management. In other words, the CMDB should include only those attributes that are directly relevant to the intended purpose of the CA CMDB, such as Change Management. All other attributes can be accessed through the MDR Launcher.

Some ways in which you can leverage the MDR link for CIs include:

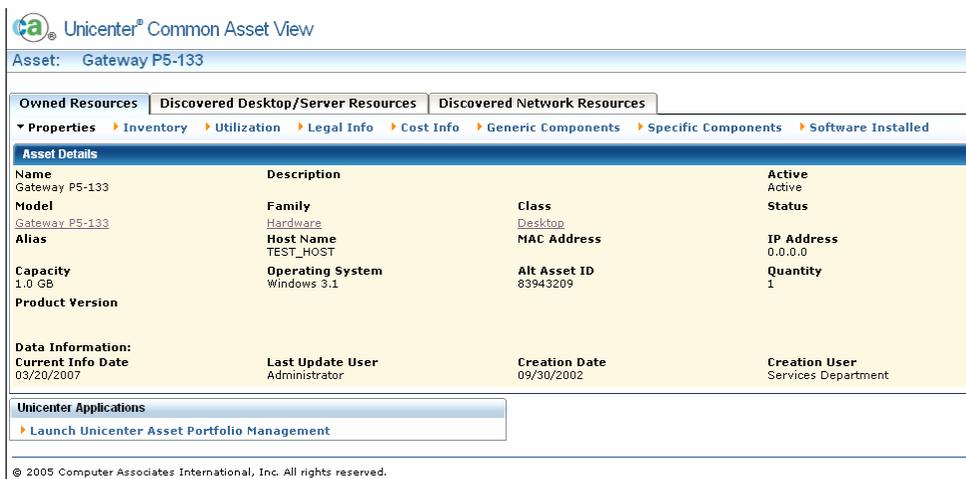
- Verify a hardware change by launching CA Cohesion from the CA CMDB hardware.server detail screen
- Obtain diagnostic and incident reporting details for a particular CI by launching its vendor web page from the CI Detail screen
- Identify contract details by launching the contracts management system (such as CA IT Asset Manager) from the Contract CI detail screen



- Review SLA details by launching CA Service Catalog from the SLA’s CI screen
- Launch CA Remote Control from the server’s CI to take over that server in order to diagnose and correct a problem.

Another useful CA CMDB tool is the *Common Asset Viewer* (formerly known as the Asset Maintenance System). The Common Asset Viewer is a collection of browser based view-only screens available to various CA applications so that they can view details on any asset in the MDB. The Common Asset Viewer provides a common interface which can be used for viewing owned and discovered asset information.

Note: The MDB must be shared between these applications as the Common Asset Viewer does not support applications residing in multiple MDBs.



In r11.x, and r12.x, the Common Asset Viewer is embedded with CA Service Desk Manager, CA CMDB, CA IT Asset Manager, and CA ITCM applications. In CA Service Desk and CA CMDB, when looking at a CI, Common Asset Viewer provides a common interface through which the consolidated asset details relating to the CI can be viewed. It also enables navigation from the asset data to other CA asset-related applications and allows users to see data that is stored about the CI in these other applications.

Common Asset Viewer contains three tabs: one for Owned asset information, one for Discovered asset information, and one for Network asset information. The asset data contained in these tabs are typically read from and maintained by the following CA applications:

Common Asset Viewer Tab	Typical Source for Data Information and Maintenance
Owned Resources	CA IT Asset Manager; CA Service Desk Manager; CA CMDB
Discovered Desktop/Server Resources	CA IT Client Manager (in particular, the CA Asset Management component)

Common Asset Viewer Tab	Typical Source for Data Information and Maintenance
Discovered Network Resources	CA Network and Systems Management

Common Asset Viewer is accessed through a URL and can be configured to display any combination of these three Common Asset Viewer tabs by specifying the correct parameter when invoking its URL. Users can also configure how Common Asset Viewer appears and whether it will include links to CA Service Desk, CA Asset Portfolio Manager, or CA ITCM applications. For example, you can begin in CA Service Desk and launch into Common Asset Viewer to view details on an asset by forwarding URL information from one application to the other.

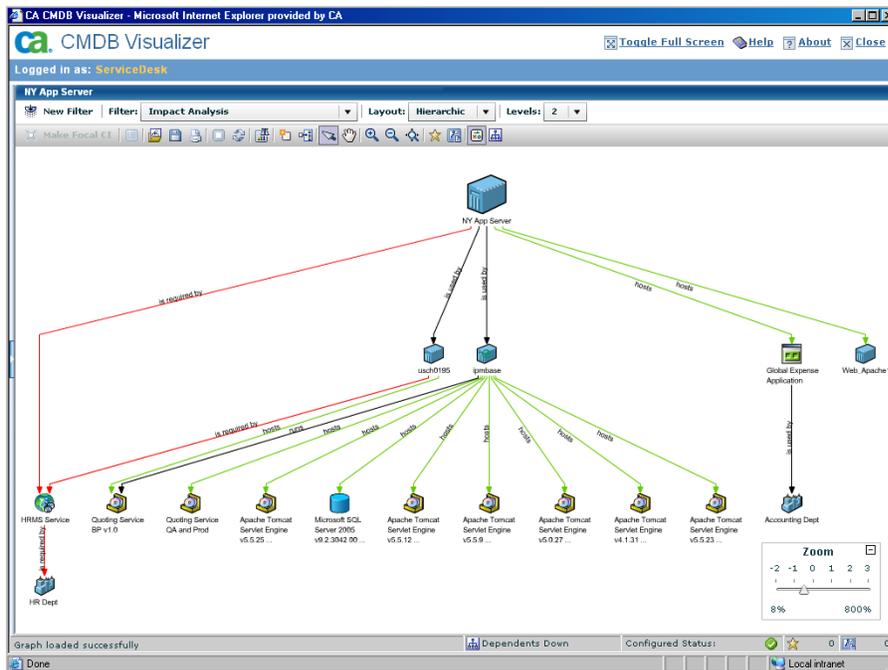
Each application that embeds Common Asset Viewer installs it into its own application directory. The Common Asset Viewer directory is *not* shared by different applications. However, as long as all Common Asset Viewer instances point to the same MDB, they will all display the same data.

Common Asset Viewer configuration information is stored in the AMS.properties file located in the Common Asset Viewer installation directory. Although this file can be edited manually, in most cases, it should be edited by using the AMConfig.java program included with Common Asset Viewer. Limited instructions for using this program are included at the beginning of the AMS.properties file itself.



CA CMDB Visualizer

CA CMDB also includes a *Visualizer* component which graphically displays the relationships between CIs and can be used to quickly pinpoint both change impact and root cause analysis. You can also launch in context directly from the Visualizer using the MDR links described above in order to drill into details about the CI in focus directly from there. Here you can see an example of an IT Service displayed through the CA CMDB Visualizer:



Additional examples demonstrating how to navigate through the Visualizer for a faster identification of possible root cause of Incidents and Problems and Impact of Changes are provided throughout this document.

CMDBf Standard Support

Resolution of incidents and problems reported to the service desk often involves combining data from many different sources. A support analyst may need to see hardware or software details, contract information, recent changes, and other data that resides in disparate management data repositories to resolve a reported failure. In today's environment part of the required data may reside in a CMDB, while the rest may be in repositories maintained by discovery tools, network management applications, asset management systems, change management applications, and so on. The challenge is to provide an easy means to access this data, regardless of where it resides.

CA, along with BMC, Fujitsu, HP, IBM, and Microsoft, is a founding member of the CMDB Federation Working Group (CMDBf). This group of major CMDB players recognized the need for standards to address inter-operation of CMDBs and other management data sources across vendor lines. The CMDBf has been working since February, 2006, to develop a set of specifications and supporting materials that enable the creation of a federated CMDB that spans multiple authoritative data sources, regardless of the origin of the data source.

In this effort, a CMDB is defined as a data repository that contains Configuration Items (CIs) that have been authorized using a configuration management/change management process. A CMDB contains a subset of the universe of attributes that describe a particular CI, and also contains information about the relationships between and among CIs.

Along with the CMDB itself, the CMDBf has defined two more pieces of the federation picture:

- A management data repository (MDR) is a definitive data source that has additional information about the CIs in the CMDB that may be of interest to a group of users or another application. Examples are discovery applications, network management applications, asset management systems, and so on.
- Transaction artifacts (TAs) are process outputs like incidents, problems, change orders, alerts, and so on.

A Federated CMDB may encompass any combination of CMDBs, Management Data Repositories, and Transaction Artifacts. A typical example of a federation scenario could include a CMDB, a network discovery tool, a network management tool, and a service desk. In the real world, this could be a total CA solution, or could be a mix of CA and other third party products.

Federation is the system that ties all of these data sources together into a virtual database. To participate, an MDR must register with the Federation system to identify how to connect with it, what capabilities it provides, and what data it would like to consume.

CA Service Desk typically provides transaction artifacts (incidents and problems) to applications like CA Service Metric Analysis and CA Service Accounting, and it consumes data from the CA Service Catalog, Discovery applications, Network Management applications, and so on.

The overall Federation specification being defined by CMDBf is being completed in stages and includes:

- Mechanisms for administering a collection of federated CMDB data sources.
- Document formats for exchanging meta-data about and instances of resources, process artifacts, and relationships.
- Mechanisms for supporting notification of changes to data in a participating federated CMDB data source.
- Mechanisms for synchronizing data between participating federated data sources.
- Provisions for appropriate security.



Where to go for more information

For more details on the direction CMDBf is taking and the phases in which the specification will be created, the White Paper describing the Federated CMDB Vision can be found at <http://www.cmdbf.org>.

Where to go for more information

In addition to the CA CMDB product documentation further details can be found in the following documents which are available through the CA Support Website:

- *Incident and Problem Management Green Book*. This guide currently includes chapters on the CA CMDB + CA Service Desk integration – both general and CA specific examples.
- For CORA troubleshooting tips, consult Technical Document TEC484950 which provides a detailed CORA Troubleshooting guide.
- “MDB, the Assets and CORA” doc, which is available at the following link (with portions included in several Green Books).

https://support.ca.com/phpdocs/0/common/impcd/r11/MDBMain/Doc/CORA_MDB_and_Assets_SC.pdf

Chapter 3: Using CA CMDB with CA Service Desk Manager

CA Service Desk and CA CMDB, which were previously distributed as standalone products, were consolidated, along with CA Service Desk Knowledge Tools and CA SupportBridge, into the CA Service Desk Manager r12 product. ITIL compatible, PinkVERIFY certified, and built on a proven, scalable architecture, CA Service Desk Manager aligns IT processes with your business goals while providing superior service for employees, customers and partners.

Although integration between these components is included as part of the CA Service Desk Manager installation, this chapter focuses on how to make the most of the integration points between these components. The following key topics are presented:

- Overview and value of the integration
- How the CA Service Desk integration works
- Using the integration
- Reference Documentation

Overview and Value of the Integration

As parts of the CA Service Desk Manager product, CA CMDB and CA Service Desk integrate by default. By installing and configuring CA CMDB in conjunction with CA Service Desk, the integration will immediately take advantage of the complementary functionality, providing the following value:

- The Change Management and associated Workflow capabilities provided by CA Service Desk will assist your organization in evaluating, prioritizing (through impact and risk analysis), approving, planning, testing, documenting, scheduling, and implementing authorized changes to CIs within the CA CMDB.
- The Incident Management capabilities in CA Service Desk will assist your organization in tracking and resolving any disruptions in business services resulting from outages within the infrastructure as well as tracking outages associated with authorized or unauthorized changes made to CIs within the CA CMDB.
- The Problem Management capabilities in CA Service Desk will assist your organization in eliminating recurring incidents affecting CIs in the CA CMDB, and minimizing the impact of incidents that cannot be prevented.
- The Knowledge Management capabilities in CA Service Desk will allow organizations to gather, analyze, store, and share knowledge and known error information for CIs in the CA CMDB.



- The Request Management capabilities in CA Service Desk will assist your organization in granting end users access to business services modeled in the CA CMDB.

Integration Points and Functionality

When integrated with CA CMDB, CA Service Desk adds full access to support for requests, incidents, problems, change orders, knowledge, and other process artifacts for a particular CI.

Integration Points from CA Service Desk

As part of the CA Service Desk Manager solution, CA CMDB provides the following benefits to CA Service Desk:

- Ability to launch the CA CMDB Visualizer in context from the CI Detail window, providing a graphical display of CIs and their relationships supporting the service desk analyst activities. The CMDB Visualizer view can also be filtered based on pre-defined and customizable filters.
- Access to Advantage Data Transformer (ADT), Cohesion ACM, the GRLoader tool and the Universal Federation Adapter, which enables the import of data from third-party sources (named Management Data Repositories, or MDRs) using either the CA Service Desk or CA CMDB interface.
- Ability to initiate CMDB MDR Launcher from the CI Detail window. The MDR Launcher is configurable with no programming required, and provides access to the management data repositories containing federated information about a displayed CI.
- Addition of Standard CI field and Versioning tab in the CI Detail window which automatically keeps a record of any attribute changes to the CI and can be used as a basis for attribute comparison between other snapshots, user-defined milestones, and standard CIs. Also includes the ability to export attribute data to a CSV file.
- Adjusted Attributes tab in CI Detail Window includes dynamic, extended CI Attributes based upon the CI Family.
- Additional Administration features and new content for Families, Classes, Manufacturers, Model Definitions, Relationship Types, Stored Queries, Access Types, and Roles, as well as the CMDBf Web Services API

Integration Points from CA CMDB

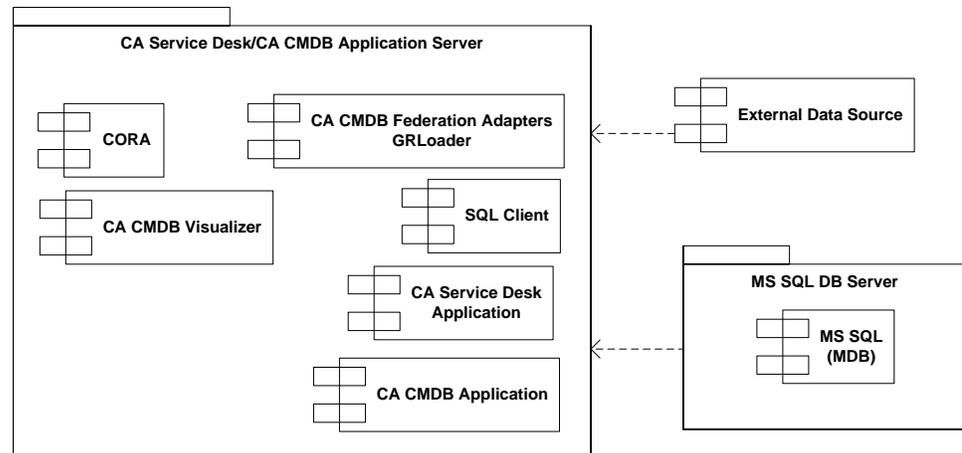
Integration with CA Service Desk provides the CA CMDB with full access to request, incident, change, problem, knowledge, and workflow, including but not limited to:

- Administration and new MDB content for requests, incidents, change orders, problems, stored queries, access types, roles, reports and more
- Ability to associate a CI and a subsequent workflow to a request, incident, change order, or problem.

- New Request, Incident, Change, and Problem tabs in the CI Detail Window to keep a record of all tickets opened against a particular CI.
- New tabs in initial login screen for Knowledge, Knowledge Schedule, Change Order Schedule, and Reports.

How the Integration Works

Consider the following component placement diagram:



Although the CA CMDB integrates with CA Service Desk, by default, as part of the CA Service Desk Manager product, the Service Desk component must be installed prior to installing CA CMDB. When CA CMDB is installed and configured into a pre-existing CA Service Desk installation, the additional integration functionality will be immediately available.

Data from external data sources called MDRs, such as other CA Technology, 3rd party database tools, or Excel spreadsheets, can be imported into the CA CMDB using the GRLoader import tool leveraging the Universal Federation Adapter or even leveraging web services with the CMDBf specification which is provided within the application.

The CA CMDB Visualizer provides a consolidated view of CIs and presents a graphical representation of the relationships between them.

Example of the CA CMDB Integration

Business Challenge

Mark C., the Change Manager at a large financial company, wants to better manage the changes made to business critical CIs in order to reduce the impact those changes have on the business services they support. Changes to these CIs could include anything from modification and reconfigurations to deployment and decommissioning. Although the CA CMDB serves as a repository of CIs and their relationships, what he is looking for is a way for IT Staff users to submit Requests for Change, and to ensure these changes are followed through to execution and implementation with minimal impact to the business.

CA Approach

Establishing an effective Change Management process can help streamline the procedures as well as provide a higher level of consistency of the Change Process within the organization.

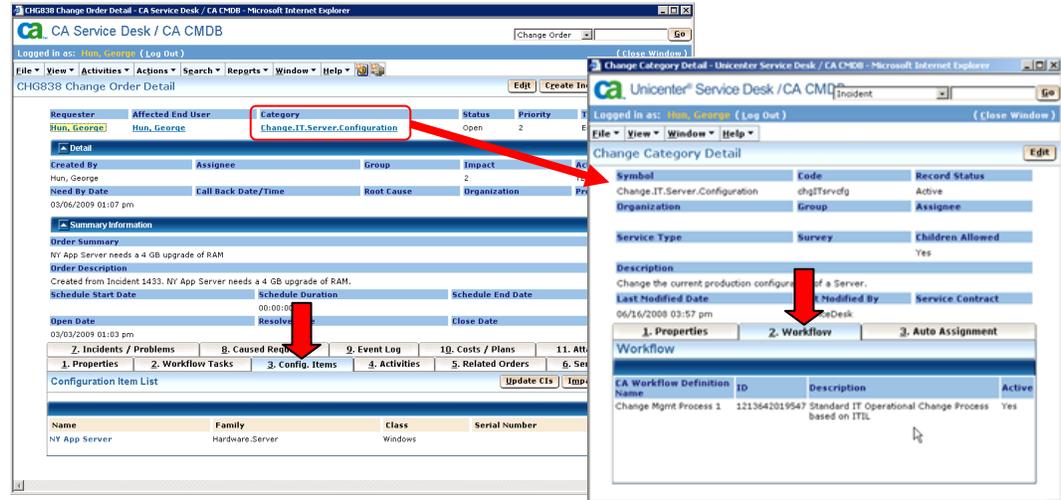
Effective Change Management which properly leverages information contained in a CMDB can provide the following:

- Reduction in outages due to change
- Decreased cost of change
- Reduced time to implement change
- Decreased risk of unsuccessful change
- Minimized operational disruptions

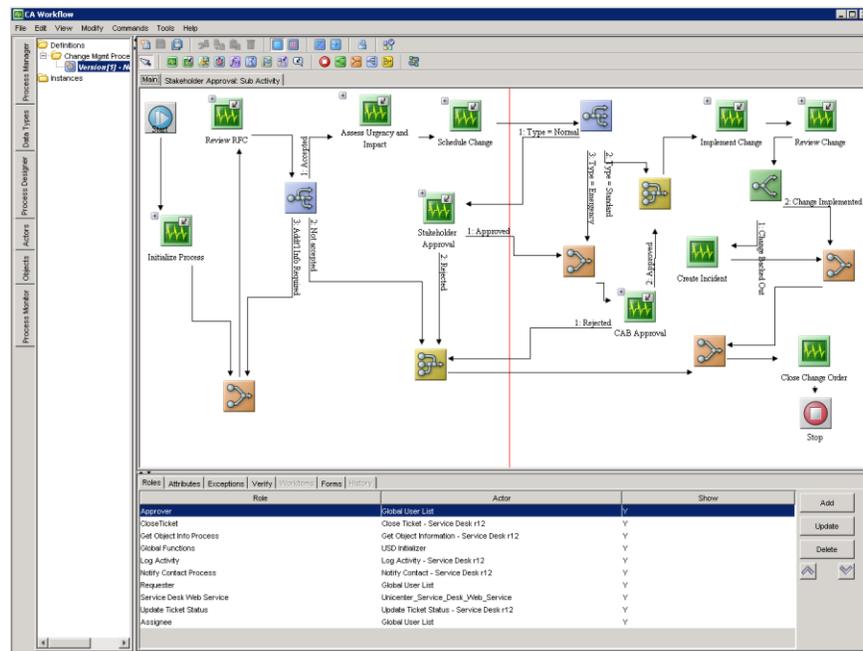
CA advises Mark's company to implement CA Service Desk with the CA CMDB.

Integration Walkthrough

Level Two Analyst, George H., has just created a Change Order based upon an Incident that was recently opened with the service desk, to resolve an outage of the company's Global Expenses application. This Incident record was generated by a CA Wily Event, which detected an error associated with an Expenses transaction. A reboot of the server hosting the Global Expenses application, 'NY App Server', temporarily resolves the problem, however after performing research; it appears that the root cause is the insufficient amount of RAM on this server. To prevent any additional outages to this business critical application, George needs to upgrade the RAM on the NY App Server. George opens a Request for Change (RFC or Change Order) to start the process of implementing this Change and he assigns the Change Order to the category of 'Change.IT.Server.Configuration'. This Change Category has an IT Change Management process tied to it.



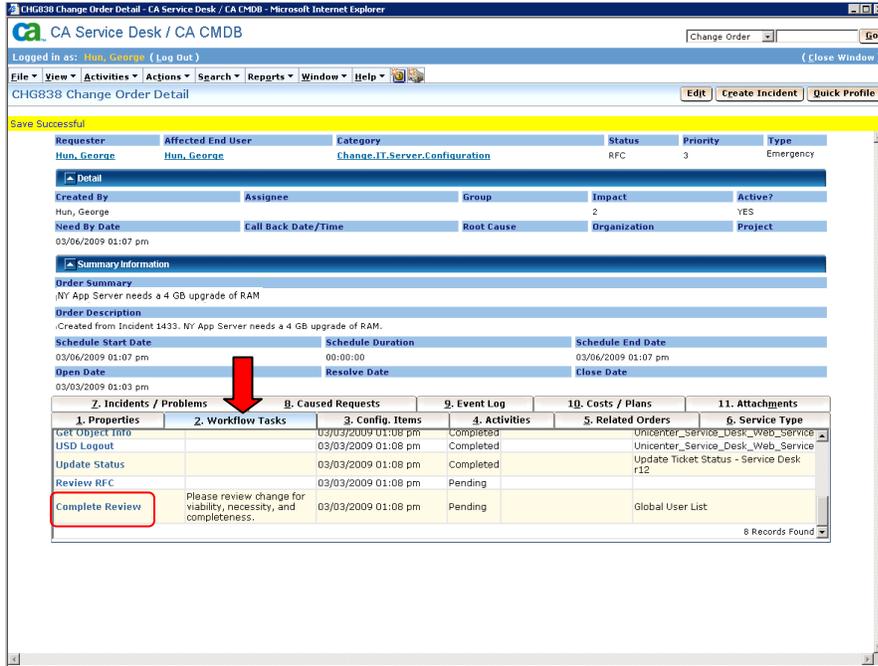
Upon submission of the RFC, the following Change Management process (defined in the CA Workflow IDE) is instantiated behind the scenes.



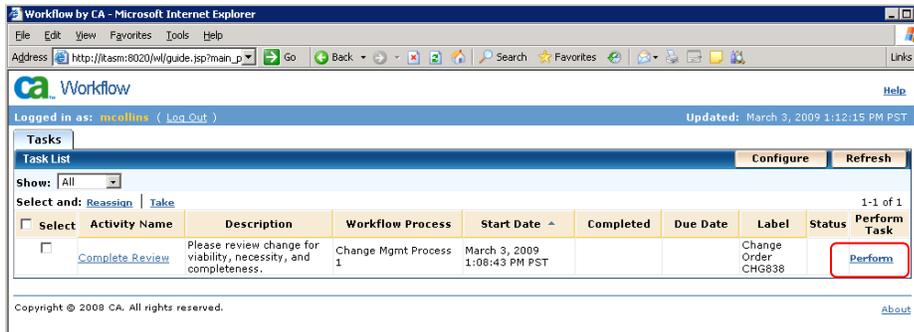
This automated process flow assesses urgency and impact, obtains approval from the Change Advisory Board (CAB) to implement this change, schedules the implementation, executes this change, and then validates that the change was implemented successfully. If the change was not successfully implemented, an incident is opened and the change is backed out.

The first pending task in the process is for the Change Manager, Mark C., to Review the RFC in order to validate all necessary parameters on the RFC.

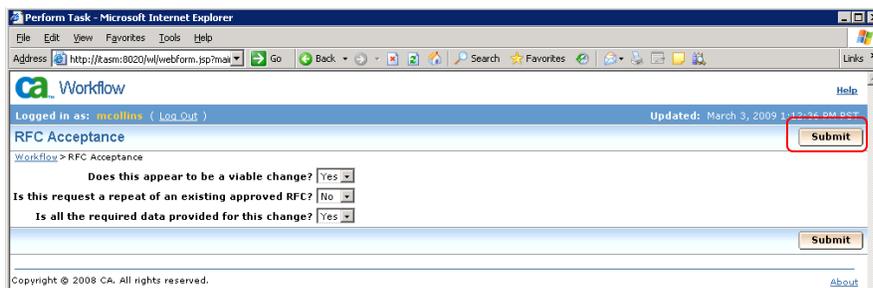
Example of the CA CMDB Integration



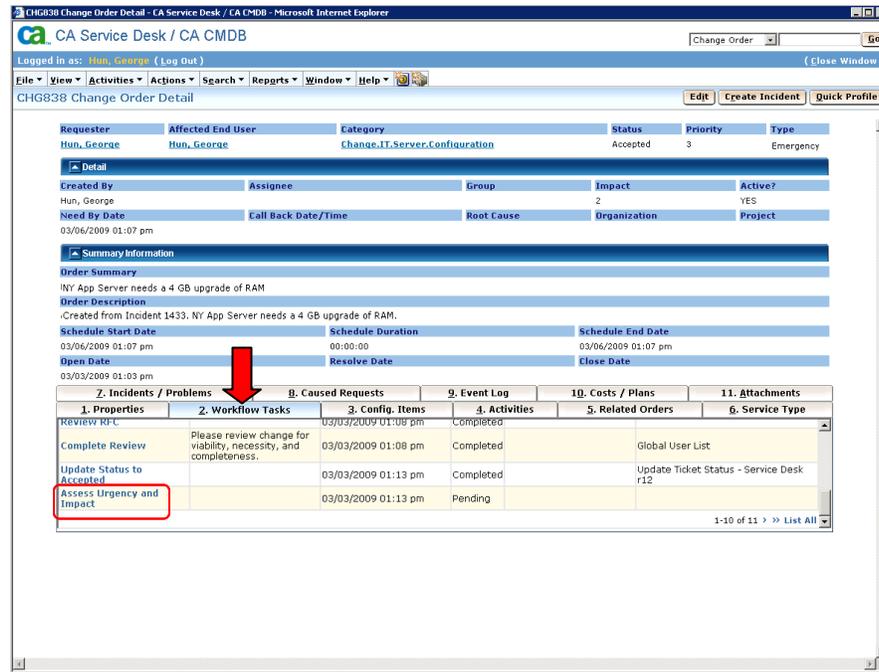
When Mark logs into his Worklist to check for any tasks that he may have to complete, he sees 'Complete Review' of Change Order# 838, and clicks 'Perform' to complete this task:



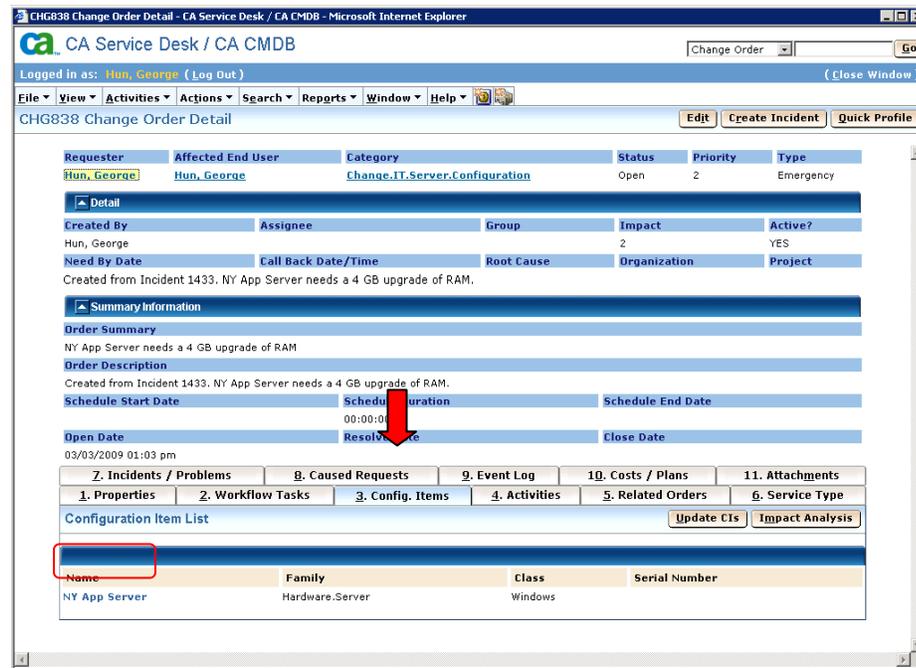
Mark is presented with an RFC (Request for Change) Acceptance Form. Since this is a viable change, this is not a repeat of an approved RFC, and all of the required information is documented correctly, Mark completes the form and clicks 'Submit':



The next step in the process is for Mark to perform an urgency and impact analysis on the 'NY App Server' using the CA CMDB Visualizer to determine what other CIs in the organization could be impacted if this Server is brought down:

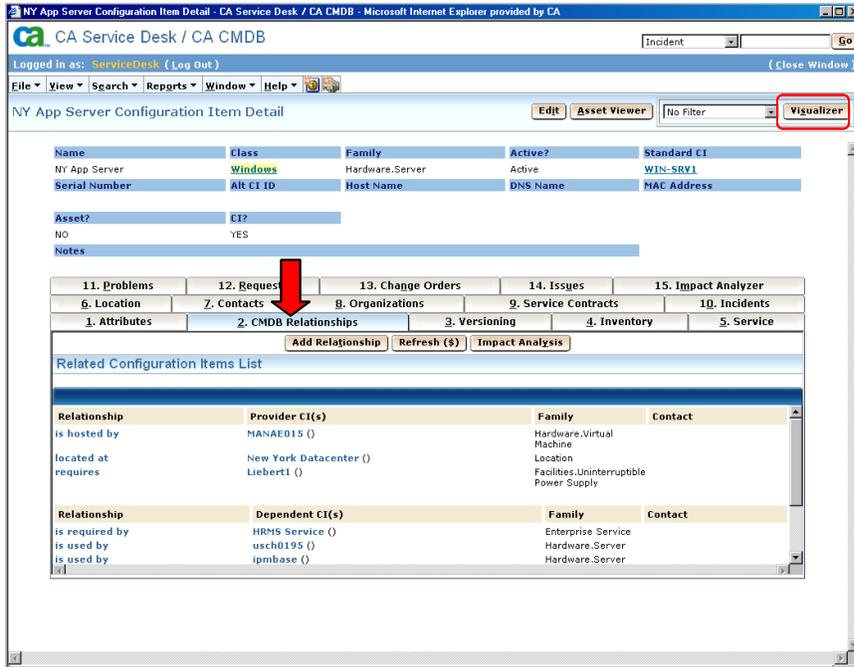


To do this Mark clicks on the 'Configuration Items' Tab in the Change Order, and goes into CI detail for 'NY App Server':

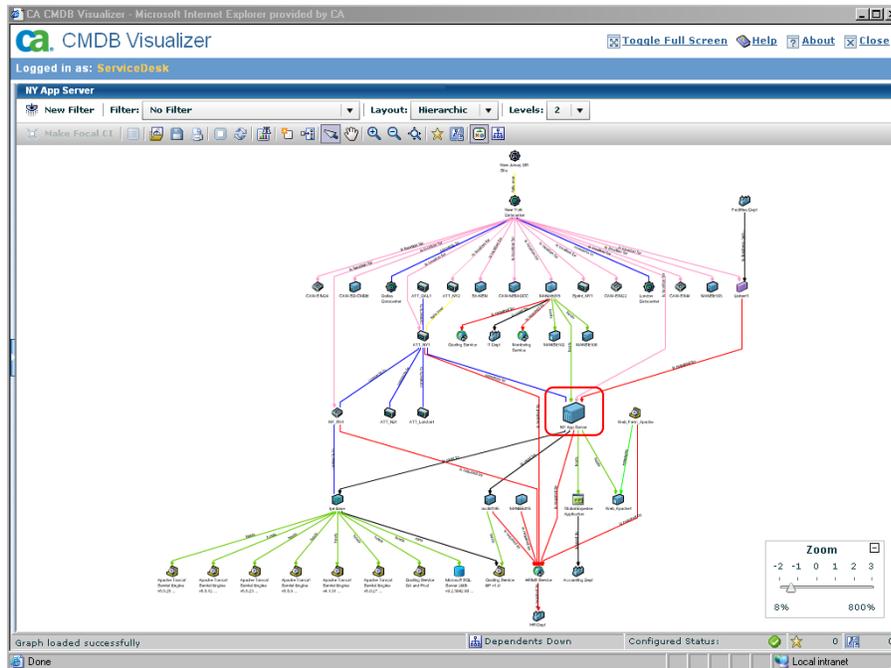


Example of the CA CMDB Integration

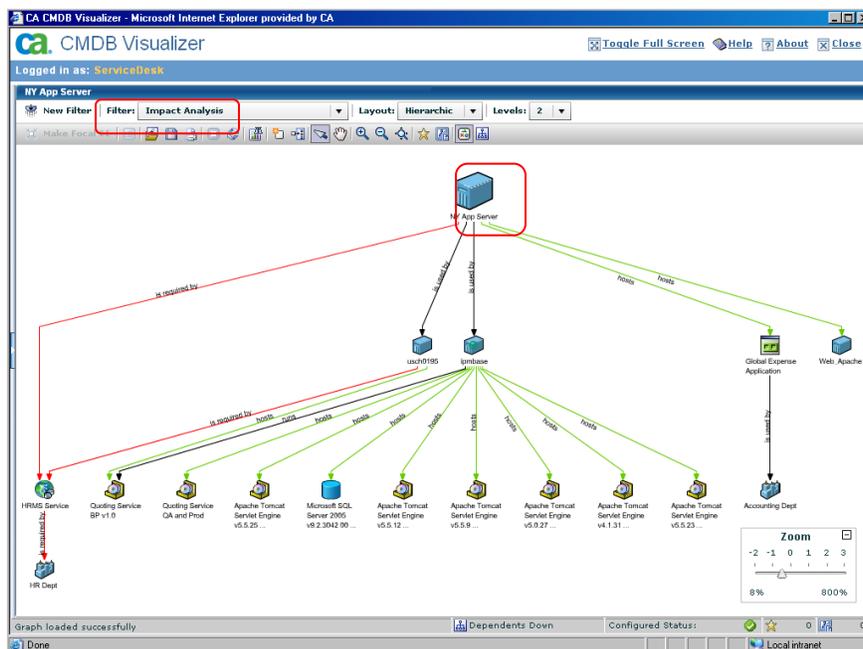
From within the CI Detail Screen, he can view all related CIs and their relationship types in the 'CMDB Relationships' Tab. He can also click the 'Visualizer' button to display a full graphical view of the relationships of this CI:



Through the Visualizer component, he can see all the CIs which have relationships to the server hosting the Global Expenses application, 'NY App Server' - including providers, dependents, and peer-to-peer relationships. The Visualizer also graphically displays the relationship types:

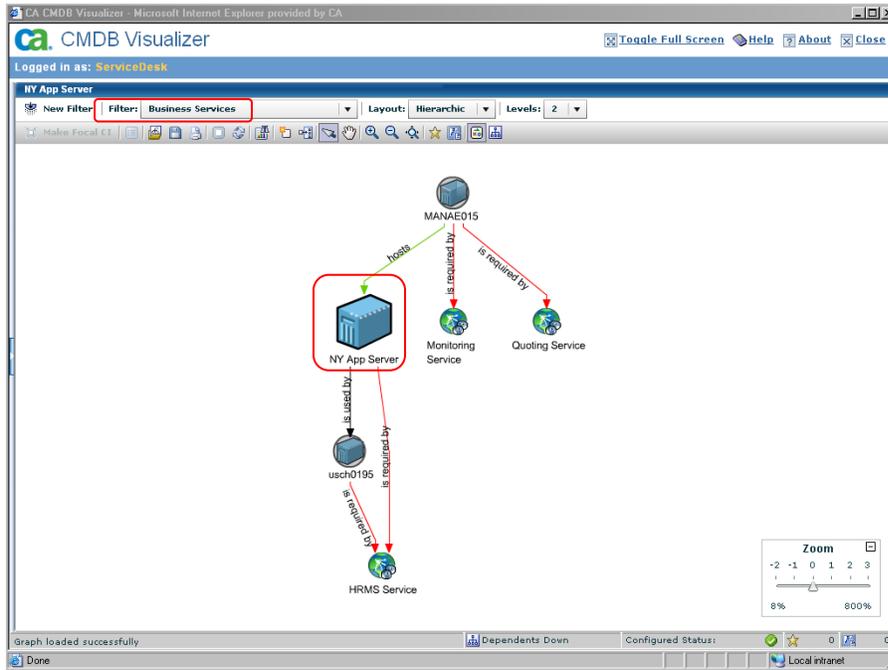


By selecting the 'Impact Analysis' Filter, Mark can display all CIs that are dependent upon 'NY App Server'. This information can help him determine the potential impact associated with this change, as well as how to appropriately schedule the change in order to minimize the impact for end users.

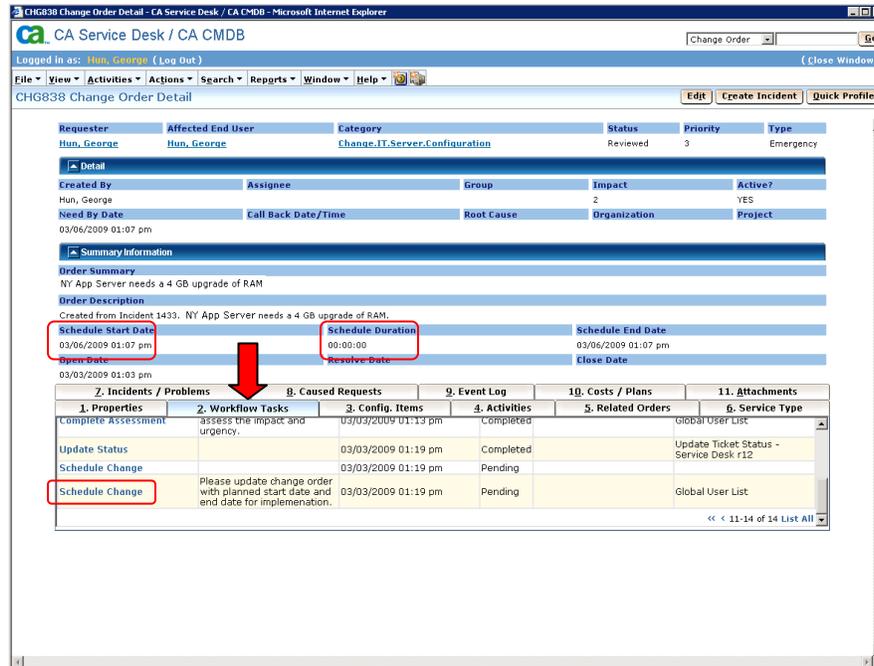


Example of the CA CMDB Integration

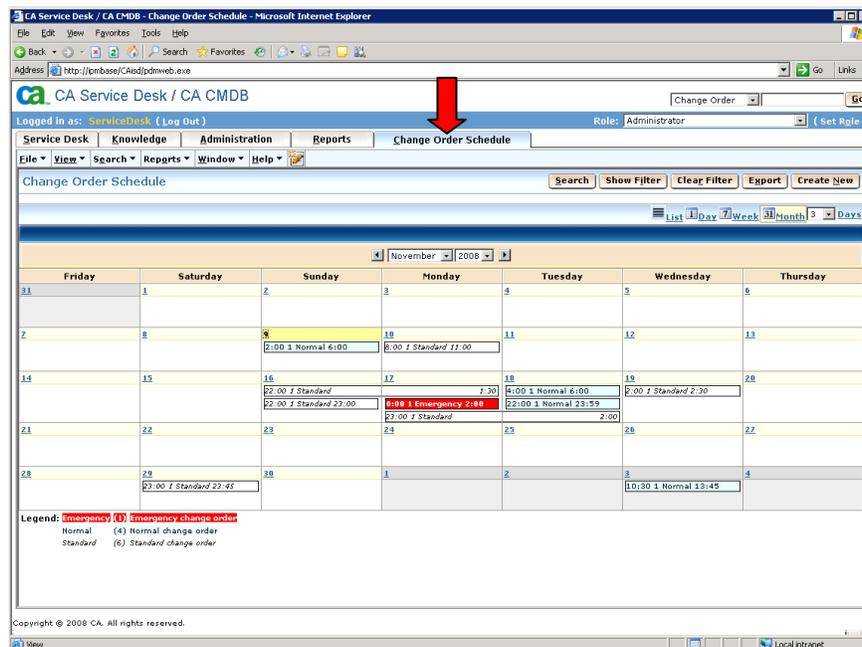
Mark can also create his own filter. In this case, Mark created a 'Business Services' filter, which will show all business services that will be impacted if this server is brought down. As he can see in this example, the 'NY App Server' is required by the 'HRMS Service' provided to the HR department. This means that the HR department will also be impacted by any changes made to 'NY App Server'.



The next step in the process is to schedule the Change, so Mark sets the 'Schedule Start Date' and 'Scheduled Duration' of the potential Change during the next Change Window:

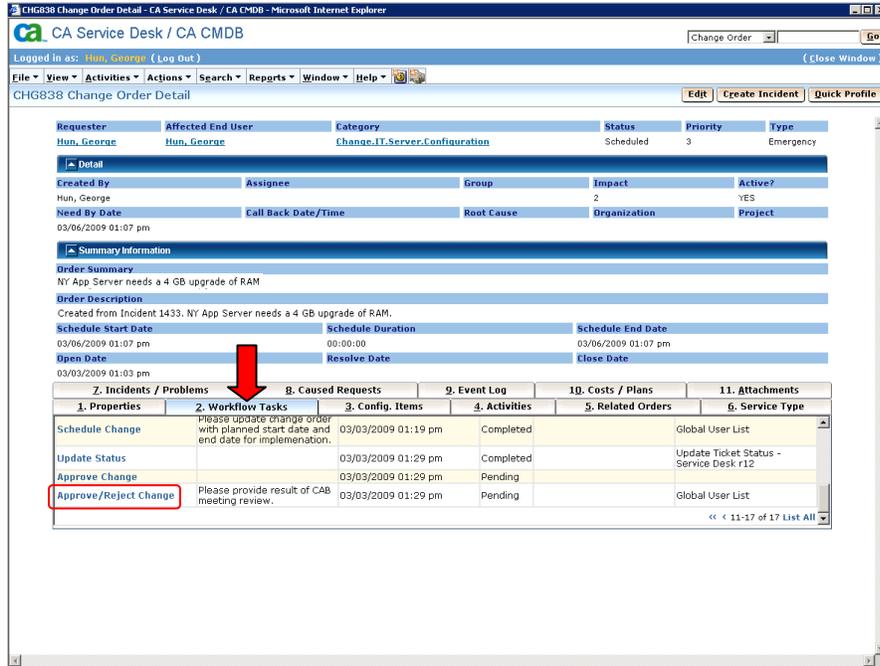


Once these dates are set, the RFC is saved and added to the Change Order Schedule, which is accessible through the Service Desk User Interface:



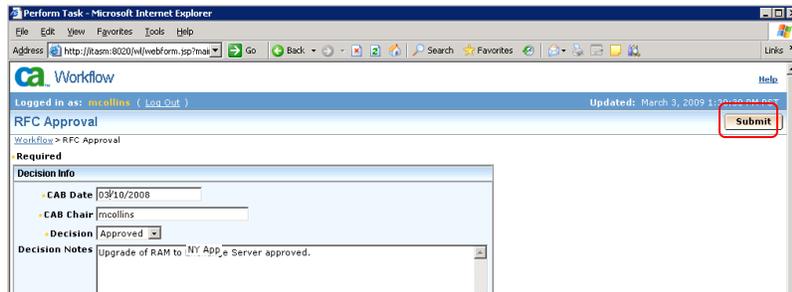
Since this change is tied to a business critical service, the necessary approvals will have to be obtained from the CAB in order to implement it

Example of the CA CMDB Integration



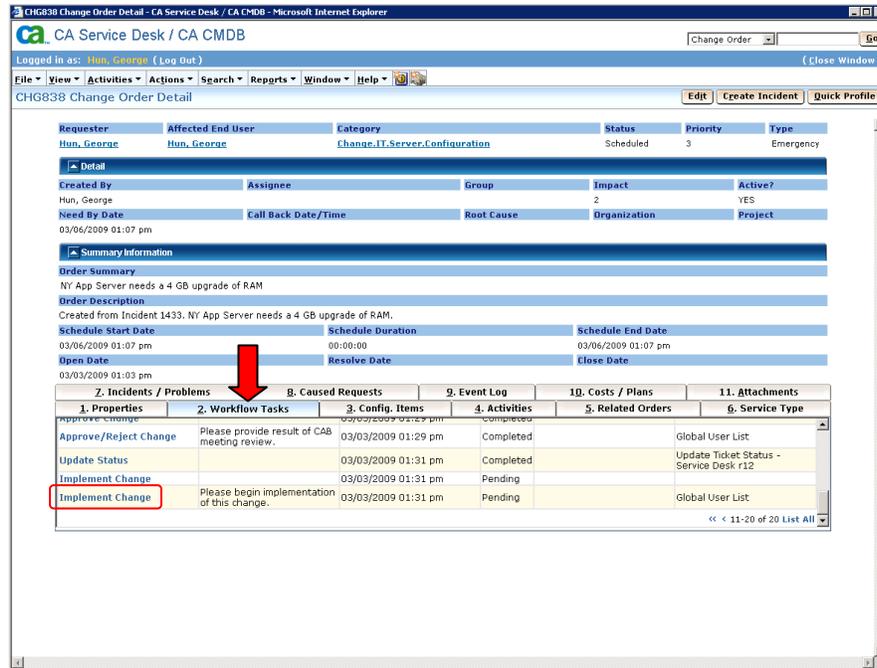
As the Change Manager, Mark sees that he has an 'Approve/Reject' pending task in his Worklist for Change Order# 838:

Mark fills out the CAB information and approval decision. Since this change is required for end users to continue submitting their expenses, the CAB approves the Change:

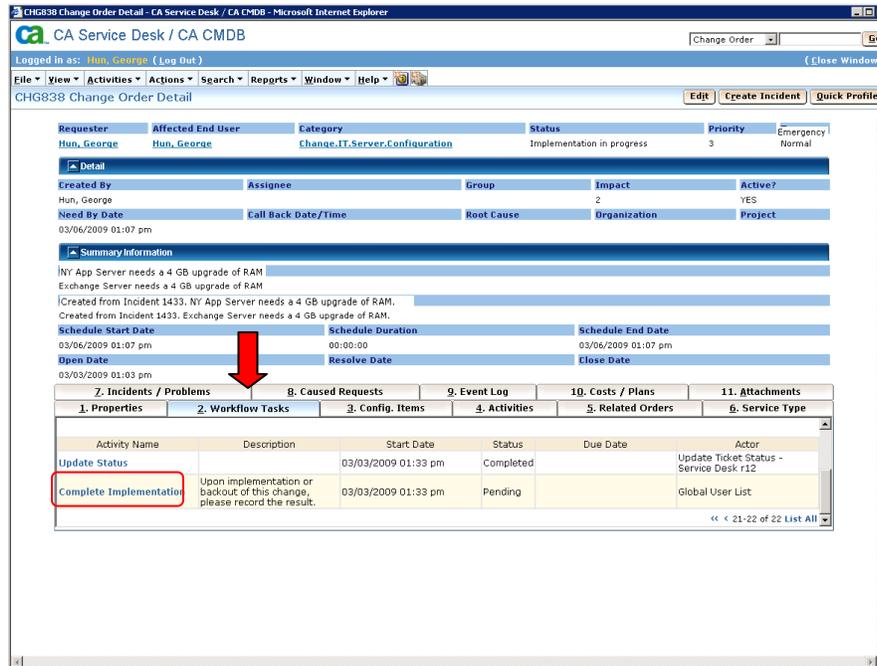


Once it is approved, the Implementer, George, is assigned a task to implement the Change to upgrade the RAM on 'NY App Server' at the time previously scheduled:



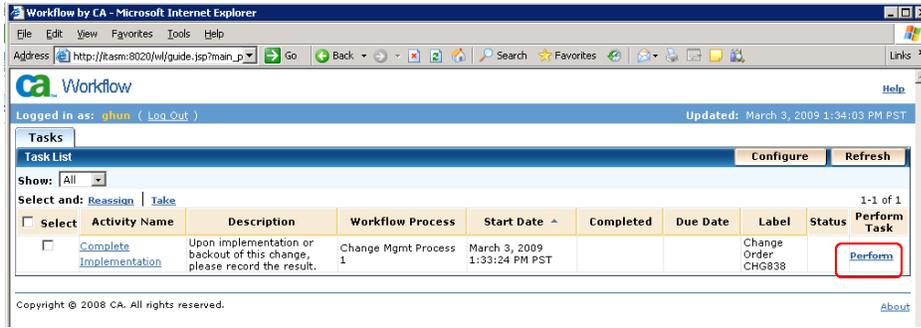


Once the 'NY App Server' has been upgraded to 8 GB RAM, George is assigned another task to sign off on the completion of the Change.



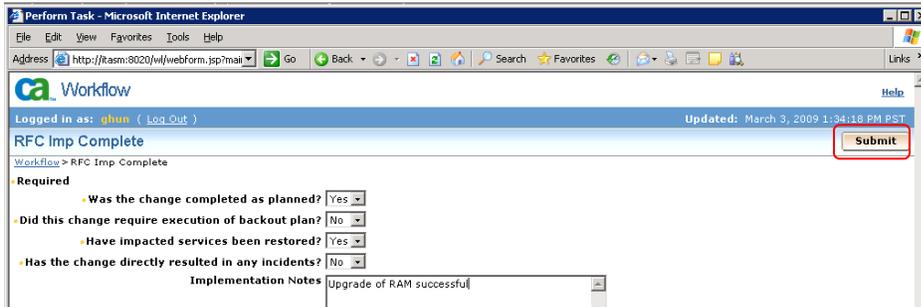
George logs into his Worklist and sees a pending task in his queue to 'Complete Implementation':

Example of the CA CMDB Integration

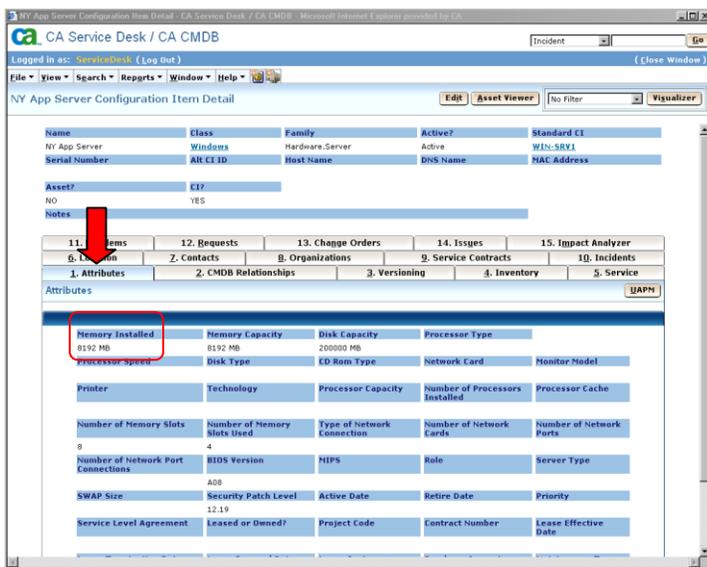


He completes a form indicating that the Change was successful. This change can be validated manually or through a discovery tool, such as CA Cohesion ACM, which can then populate the corresponding CA CMDB attributes through Web Services.

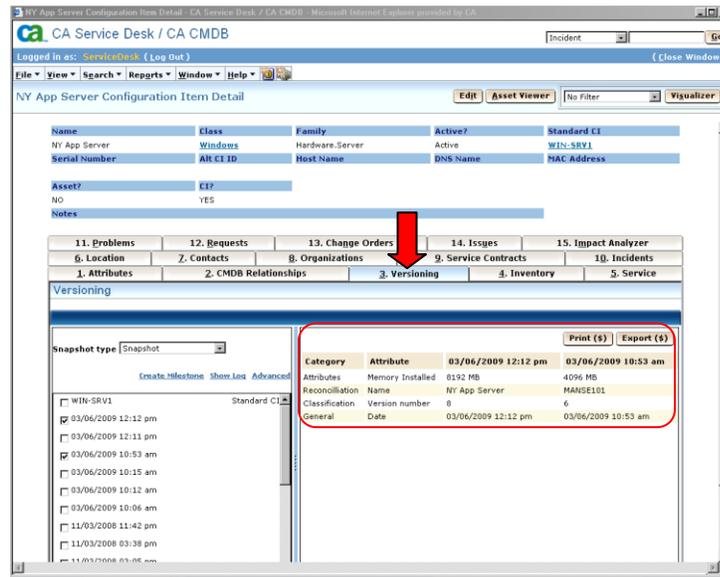
Note: If the Change was not successful, George would indicate that here as well, and the Workflow would have subsequently automatically opened an Incident based off of this Change Order.



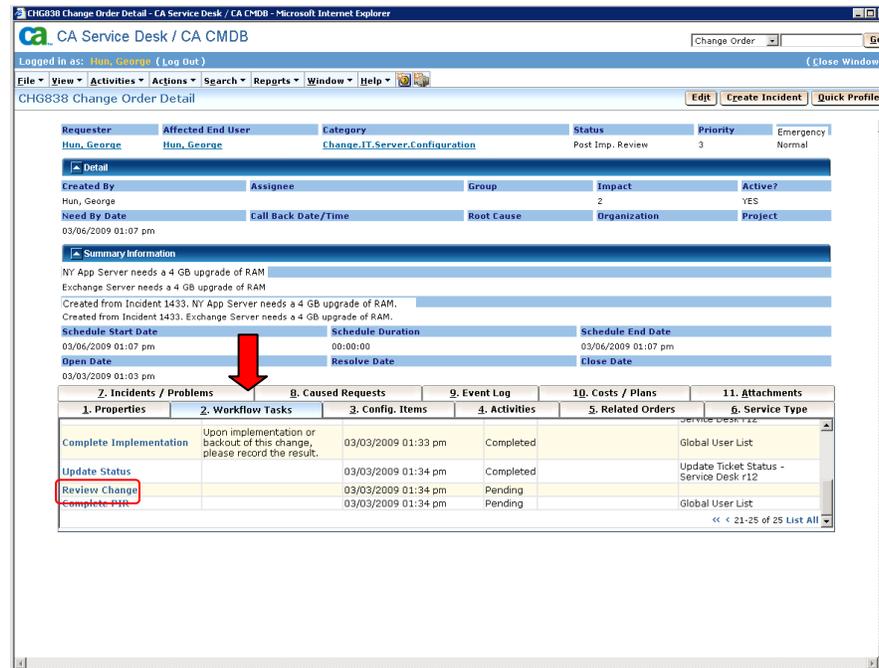
The Change is implemented and the RAM upgrade is successful. This updated RAM value is now visible in the Attributes tab within the CI detail for the 'NY App Server.'



It can also be viewed within the Versioning tab:

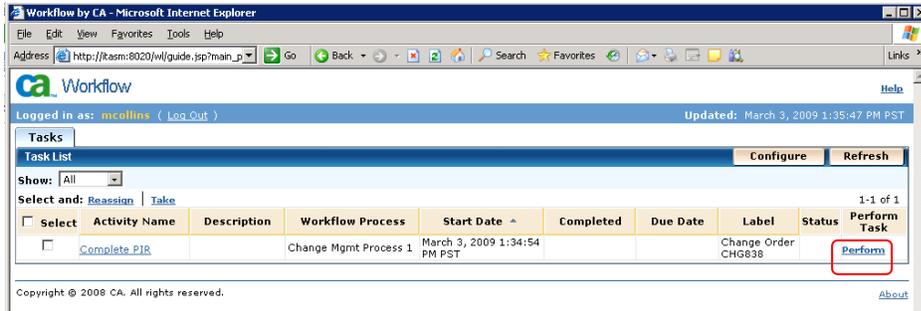


With the Change successfully completed and the RAM on the 'NY App Server' upgraded from 4GB to 8GB, Mark, as Change Manager, is assigned a task to complete 'Post Implementation Review' to ensure that the objective for the Change has been met.

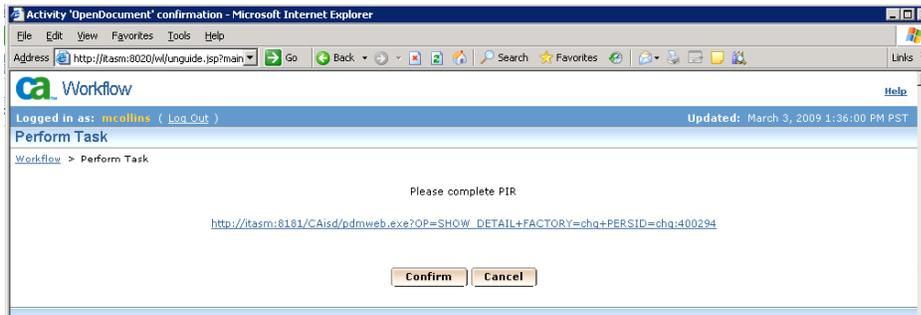


Mark logs into his Worklist, sees this final task for Change Order #838 and clicks Perform:

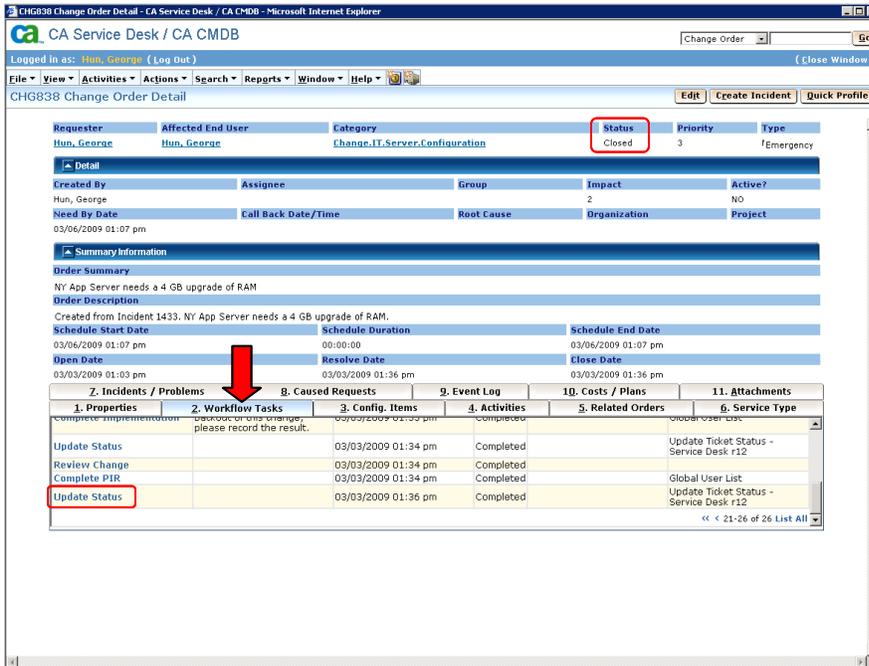
Example of the CA CMDB Integration



After completing the Post Implementation Review Mark then clicks Confirm:



The final task in the Workflow is to automatically close Change Order# 838:



Configuring the Solution

The process of configuring this solution consists of the following tasks

- **Task 1:** Install and configure CA Service Desk 12.0. Note that when integrating CA Service Desk with CA CMDB, CA Service Desk must be installed prior to installing CA CMDB.
- **Task 2:** Install and configure CA CMDB 12.0. Note that CA CMDB must be installed on the CA Service Desk Primary Server.
- **Task 3:** Install and configure CA CMDB Visualizer.
- **Task 3:** Install and configure CA EEM 8.3.
- **Task 4:** Install and configure CA Workflow 1.1.5 SP4
- **Task 5:** Document, from a business perspective, and with all stakeholders involved, the process flow required for your organization to successfully manage and implement planned IT changes.
- **Task 6:** Map your business process to a technical process definition.
- **Task 7:** Create a technical process definition using the CA Workflow IDE. .
- **Task 8:** Assign your CA Workflow Process Definition to a Service Desk Change Category.
- **Task 9:** Open a new Service Desk Change Order, assign it to your new Change Category, and associate the CIs that need to undergo change.
- **Task 10:** Thoroughly test and validate your process definition.

Note that it is recommended to test your CA Workflow Process Definition in a test environment prior to the moving the process into production.

Reference Documentation

Information on this integration can be found in the following sources:

For CA Service Desk r12 and CA CMDB r12:

- Chapter 3: "Planning" in the *CA Service Desk Implementation Guide* as well as the corresponding Chapter 2: "Planning" in the *CA CMDB Implementation Guide*
- Chapter 2: "Managing Servers," Chapter 4: "Implementing Policy," and Chapter 8: "Managing Configuration Items" in the *CA Service Desk Administration Guide*



Reference Documentation

See also Chapter 2:"Upgrading" in the *CA Service Desk Implementation Guide* as well as the CA Service Desk r12 Upgrade Information Page for important details on upgrading integrated CA Service Desk and CA CMDB implementations to r12.

The following documents apply primarily to pre-r12 releases but some details may apply to r12 as well:

- *Incident and Problem Management Green Book* – Chapter 4 "Using the CMDB"
- *CA Unicenter Service Desk Integrations Green Book*- Chapter 14 "Change and Configuration Management - Integrating with CA CMDB"

Chapter 4: Integrating with CA Cohesion ACM

CA Cohesion Application Configuration Manager (CA Cohesion ACM) is included, along with CA CMDB in the suite of products that comprise CA Software Delivery Manager r12. It is used to automatically populate and maintain the CMDB with accurate CI attribute and relationship information – “as-is’ or “last known state”.

Using CA Cohesion ACM, you can:

- Discover the servers in your enterprise
- Find out what operating systems, databases, and software application components are installed on those servers
- Quickly access complex data, information, and configuration settings from within those components
- Determine the relationships and dependencies between the servers in your enterprise
- Detect server and service configuration changes and differences
- Take and retain snapshots (point-in-time copies) of your services
- the Ensure corporate software component and configuration policy compliance to standards and best practices
- Enact change on a collection of software component attributes within a service
- Troubleshoot and improve the mean time to repair your servers and services

This chapter discusses how CA CMDB r12.1 and CA Cohesion ACM r5.0 can be configured to work together. The following key topics are presented:

- Overview and value of the integration
- How the CA Cohesion ACM integration works
- How to use the integration
- Reference Documentation

Overview and Value of the Integration

When CA Cohesion ACM is integrated with CA CMDB the following functions are available:

- Import of CI detail and CI relationships into the CA CMDB from CA Cohesion ACM
- Ability to launch CA Cohesion details to view details of discovered CI attributes through the MDR Launcher
- Refresh of CI details which is useful in confirming a change to the infrastructure

Cohesion's ability to detect changes from baseline or Gold Standard configurations across applications and servers supports Change and Configuration Management efforts.

How the CA Cohesion ACM Integration Works

Integration between these products is provided by default. As a defined MDR, CA Cohesion performs discovery and provides data which is used to define CIs and their attributes in the CMDB.

CA CMDB uses both the agent and agent-less discovery techniques of CA Cohesion ACM to automatically discover servers, software components and the relationships. CI data and relationship details can be exported to the CMDB repository at scheduled intervals through an XML formatted flat file. This file is then fed into the GRLoader program in order to populate the CMDB. A federation link is maintained between the CI that is created in the CMDB and the source discovery application (CA Cohesion, in this case) enabling you to launch back into the Cohesion ACM UI from the CMDB UI in the context of a particular CI. From there you can view additional CI details that are only stored in CA Cohesion ACM.

Prerequisites

This chapter assumes that both CA CMDB r12 or r12.1 and CA Cohesion ACM 5.0 SP1 have already been installed and are working in the environment. For help installing or configuring either application refer to the individual product documentation.

A CA Cohesion discovery is a prerequisite before modifying the `cmdb_mapping.xml` file and importing CI data into CA CMDB. Once a CA Cohesion network discovery management profile and discovery profile have been completed, the xml file can be modified to import and manage those CIs in CA CMDB.

Review Appendix C in the *CA Cohesion Application Configuration Manager Implementation Guide* and make a backup copy of the default `cmdb_mapping.xml` file. By default, the file is located in the following locations:

- `\Program Files\CA\Cohesion\Server\server\webapps\cohesion\WEB-INF\classes` (Windows)
- `/opt/CA/Cohesion/Server/server/webapps/cohesion/WEB-INF/classes` (UNIX or Linux)

Review Chapter 7 in the *CA Cohesion Application Configuration Manager Implementation Guide*. Follow the instruction to configure CA Cohesion ACM user authentication in the section titled Integrating CA Cohesion ACM with Other Applications.

For CA Cohesion ACM 5.0 Windows servers, it is recommended that the following patches available on <http://support.ca.com> are applied:

- RO05522 (Windows): WIN-ADD SUPPORT FOR CA CMDB R12
- RO05523 (Windows): CUMULATIVE FOR COHSN 5.0
- RO06094 (Windows): TIMEZONE PATCH 5.0

Launch CA Cohesion in Context to a CI in CA CMDB

One of the integration features between CA CMDB r12.1 and CA Cohesion ACM 5.0 is the ability to launch the CA Cohesion interface from a CI in CA CMDB in order to view additional details on that CI.

Note: NICs and File systems do not support launch in context back to CA Cohesion from CA CMDB.

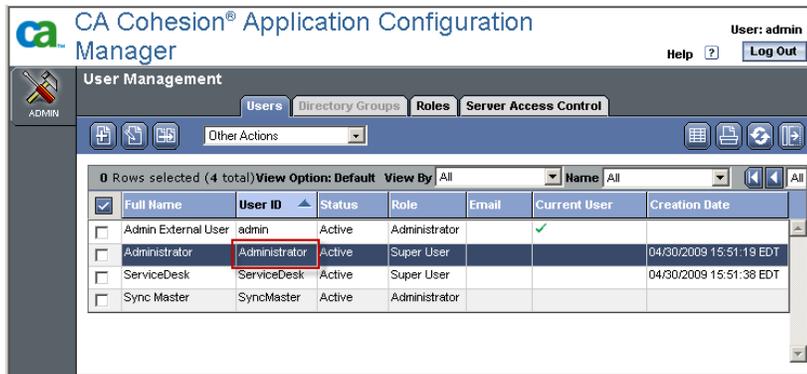
Before launching in context you need to define CA Cohesion as an MDR provider.

1. Log into the CA CMDB Administrative UI and select the Administration tab
2. Drill down to the MDR List link under CA CMDB → MDR Management
3. Click the Create New button
4. Enter the following details:
 - **Button Name:** < button label that will appear on the CI Detail page> For example, Cohesion
 - **MDR Name:** <this must match the com.cendura.installation.name parameter in the cendura.properties file>
 - **MDR Class:** Cohesion
 - **Owner:** Administrator
 - **Hostname:** <hostname of the Cohesion server>
 - **Port:** <the Cohesion Port >Ex. 8091
 - **Path:** <keep the default>
 - **Parameters:** <keep the default>



Launch CA Cohesion in Context to a CI in CA CMDB

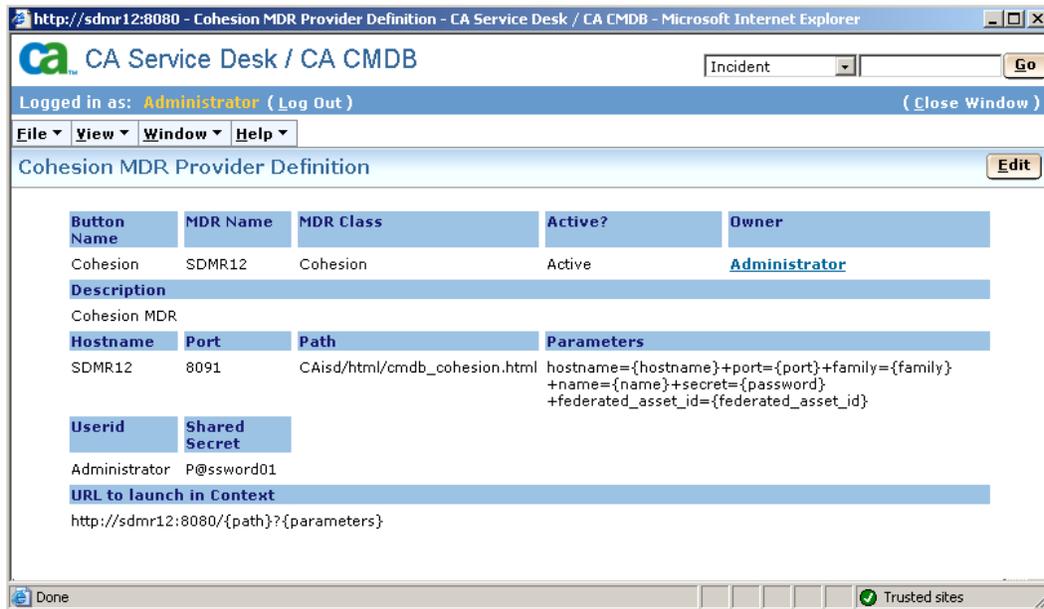
Userid: < CA Cohesion user ID with at least the Specialist role > Ex. Administrator



- **Shared Secret:** < this is the secret password you defined for the com.cendura.security.oneclickauth.secret parameter in the cendura.properties file >

```
# -- Configure One-Click Authentication --
com.cendura.security.oneclickauth.secret=unicenter
com.cendura.security.oneclickauth.scheme=
com.cendura.security.oneclickauth.user=cmdbadmin
```

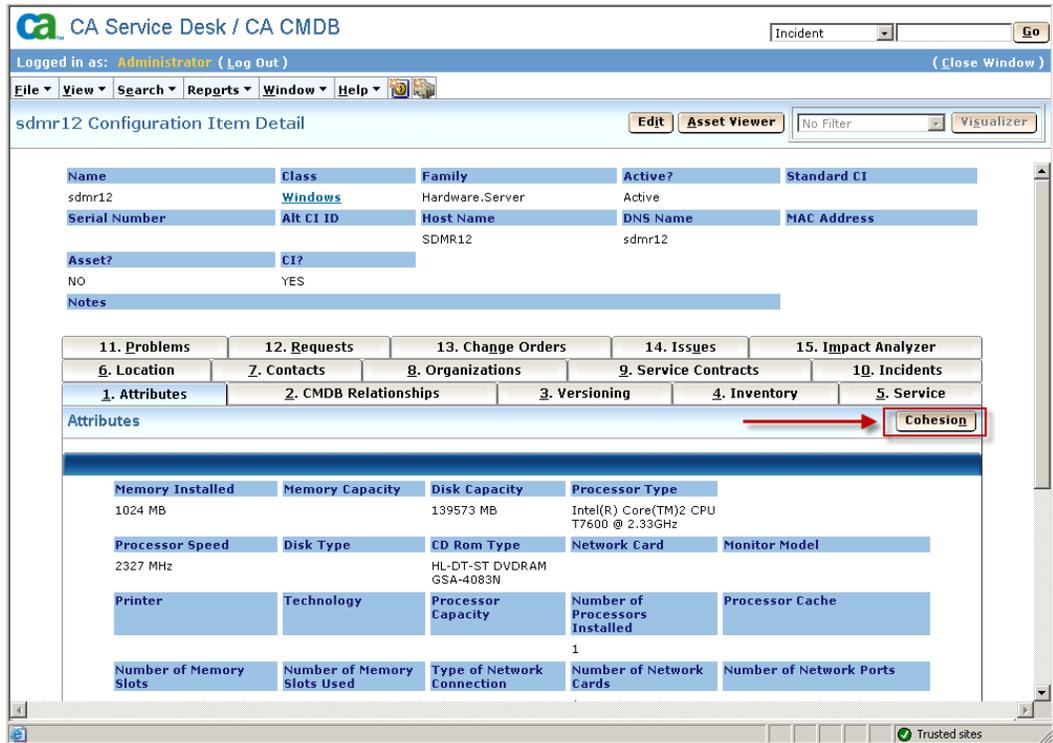
- URL to Launch in Context: <keep the default>



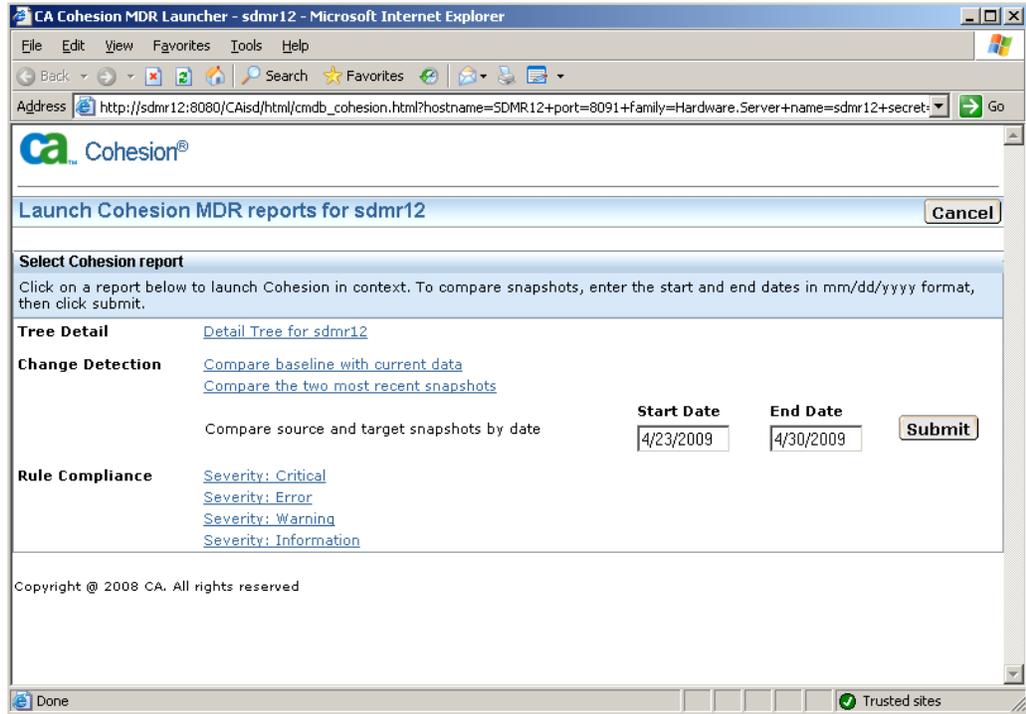
For additional information on the MDR fields, refer to the CA CMDB on line help.

Now that the Cohesion MDR is defined, you can import CIs from CA Cohesion ACM into CA CMDB using the CA CMDB Export report. The CIs that are updated or imported from CA Cohesion will have a "Cohesion" button on their CI detail form under the Attributes tab in CA CMDB.

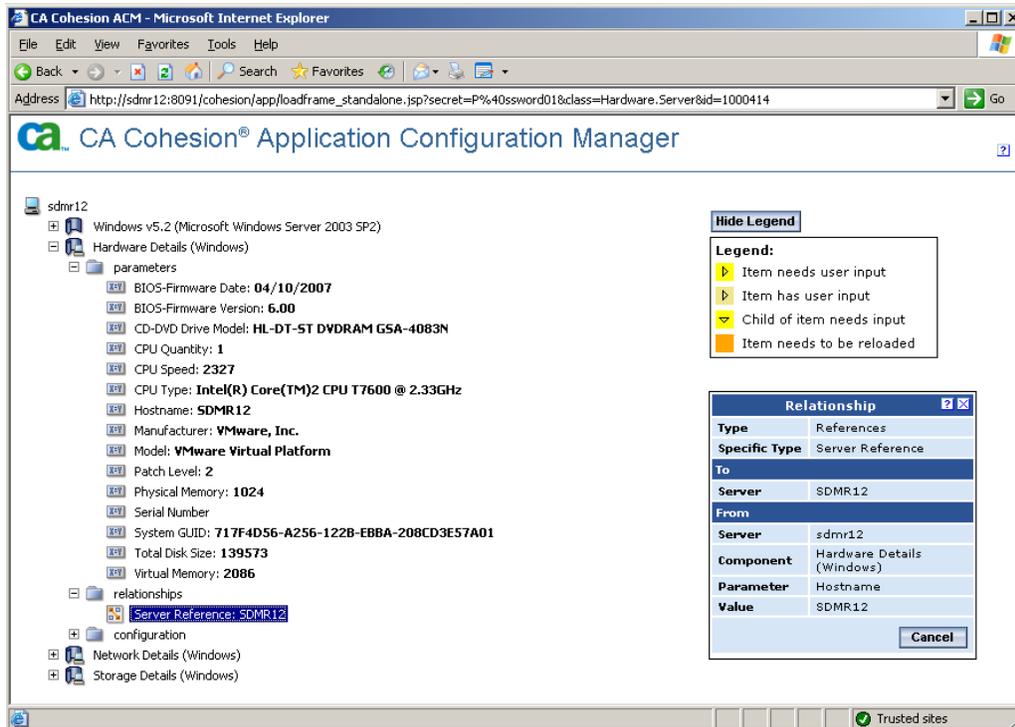




Clicking on the Cohesion button will launch the Cohesion MDR report for that CI.



Selecting the “Detail Tree” link on the report for the CI will launch additional details from CA Cohesion ACM.



Follow the next section, Exporting CIs from CA Cohesion to the CA CMDB, to run the CA CMDB Export report.

Exporting CIs from CA Cohesion to the CA CMDB

Running the CA CMDB Export Report

In addition to generating an HTML or XML format report (like the other report templates do), the CA CMDB Export report uses the reporting interface to export Configuration Items (CIs) in a xml format that “grloader” can read to import the CIs to an instance of the CA configuration Management Database (CMDB).

This report requires that you set a number of CA CMDB specific report options in addition to performing the general report generation steps.

You should also consider the following points before performing the export operation:

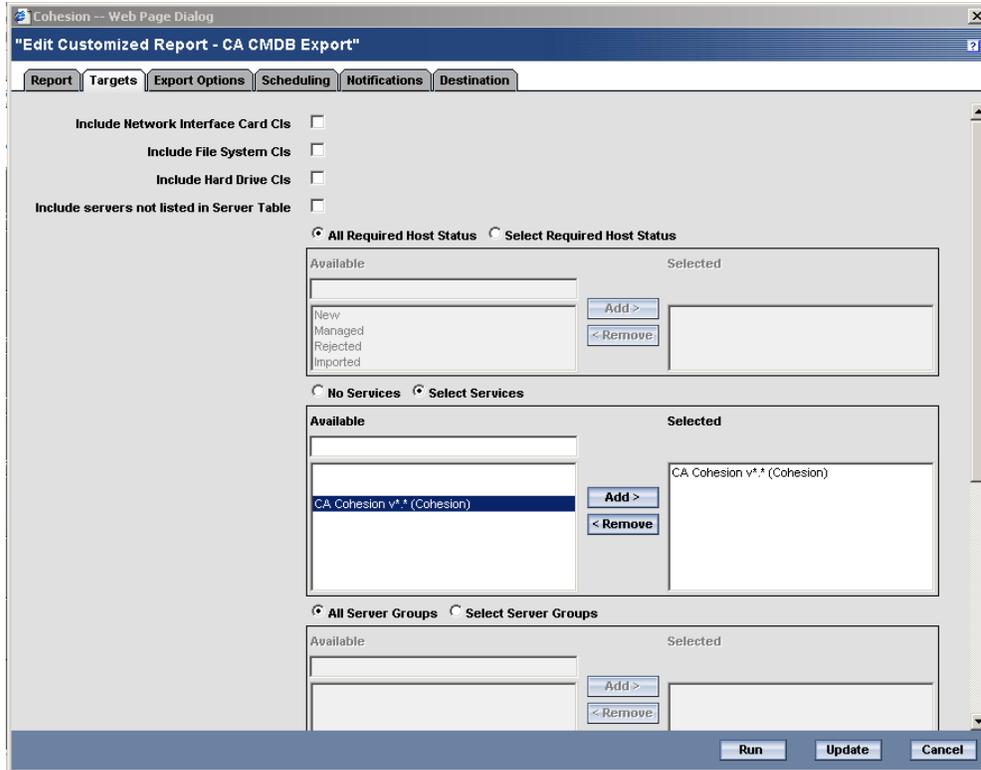
- Servers discovered by a Network Discovery operation are stored in the server table using the fully qualified server name (for example, factotum.ca.com).

- The CMDB Export Report uses the short name derived from the HostName parameter that is obtained only after a server's state is set to Managed and a Server Discovery Profile is run. For example, if the Network Discovery operation discovered a server called factotum.ca.com, the CMDB Export Report would use factotum as the managed server name that is mapped to the CMDB CI name after a Server Discovery Profile is run.
- If you run the CMDB Export Report against servers that are not managed and have never had a Discovery Profile run, it will not export these servers because they do not have short name (that is, a HostName value).
- If you run the CMDB Export Report against managed servers that have not had a Discovery Profile run at least once, it will not export these servers.

To run or save a CA CMDB Export report

The following steps are performed while logged into the CA Cohesion ACM web interface.

1. Click the Reports navigation icon to display the Report Templates page.
Note: If you have created custom reports, the Custom Reports page is displayed instead. Click the Report Templates tab.
2. In the Template Name column, click CA CMDB Export.
The Report tab of the Run or Save Report dialog box is displayed.
3. Edit any of the default values for the Report Name, Report Description, or Format fields.
4. Click the Targets tab:



5. Ensure the checkboxes are checked next to the types of CIs you want exported.

Check the Include Servers not listed in Server Table checkbox if you want CIs from servers not listed in the Server table to be exported. CA Cohesion ACM can have information about servers that are not listed in the Server table. For example, after discovering CA Cohesion ACM software, it will know the server on which CA Cohesion ACM and the CA Cohesion ACM database are hosted.

Select which of the following options are to be included in the export:

- Host status
- Services
- Server Groups
- Servers
- Component Blueprints

6. Click the Export Options tab:

7. Perform the following actions on this dialog:

- a. Ensure the Run Export checkbox is checked if you are exporting CIs. If the checkbox is not selected, the configuration items are not exported to CA CMDB, instead, the generated XML is displayed within the report output. This report can be used to preview or debug the data transferred between CA Cohesion ACM and the CMDB.
- b. Enter your CMDB user name, CMDB password, and retype your password in the corresponding fields.

Checking this option is the same as specifying `-u -p` in the Other Options field

- c. In the Server URL field, enter the CMDB Server URL and port. For example:

```
http://<CMDB_server_name>:8080
```

Checking this option is the same as specifying `-s` in the Other Options field.

- d. Click the Preload Data checkbox to preload several tables into memory

This Option is used to improve the performance of large export jobs (more than 50 entries). Checking this box also increases memory usage, so it may impact other processes.

Checking this option is the same as specifying `-P` in the Other Options field .

- e. Click the Check Input XML Data Only checkbox to prevent any database updates.

This allows you to validate the input before actually loading the data into the database. Checking this option is the same as specifying `-c` in the Other Options field (step h).

- f. Click the Update CIs checkbox is checked to allow updates to existing CIs.
Checking this option is the same as specifying `-a` in the Other Options field.
 - g. Click the Insert New CIs checkbox to allow new CIs to be accepted by CMDB.
Checking this option is the same as specifying `-n` in the Other Options field.
Optionally, you can use the Other Options field to define the export using a command line-like interface to enter the following required and optional GRLoader information:
 - h. Select a level from the Trace Level drop-down list to set the output verbosity for the error file.
Checking this option is the same as specifying `-T` in the Other Options field.
8. Click the Save button to save the report
 9. Click the Customized Reports tab and select the CA CMDB Report that was just saved
 10. Click the Run Highlighted Reports Icon to run the report

Tips for Modifying the CMDB Attribute Mapping Section

CA Cohesion ACM installs a file called *cmdb_mapping.xml* that enables CA CMDB users to customize the mapping of Cohesion CIs and their attributes so they match the structure of CMDB CIs when exported to CA CMDB.

The contents of the *cmdb_mapping.xml* file are used by the Cohesion CA CMDB Export report (the mechanism by which CA Cohesion ACM exports CI data to CA CMDB) to determine which Cohesion CIs, attributes, and relationships are exported and how to map them to CA CMDB families, classes, attributes, and relationships.

When the *cmdb_mapping.xml* file is used as provided without any modifications, the results will most likely include duplicate CIs. This chapter provides some tips and tricks for modifying the xml file to produce more desirable results.

The *cmdb_mapping.xml* file contains three mapping types: Attribute Mapping, Class Mapping, and Relationship Mapping. For information on attribute mapping, refer to Appendix C in the *CA Cohesion Application Configuration Manager Implementation Guide*.

Model Attribute

The CA Cohesion CI Server has several attributes that can be exported into the CA CMDB. Most of the attributes that will be exported are of a STRING type in CA CMDB. Others, like the model attribute, reference other CA CMDB objects and will require a lookup of the record before the CI can be created in the CA CMDB. When the CA CMDB Export report is run, CA Cohesion attempts to associate the model name discovered in Cohesion with the server CI and pass that information to CA CMDB. The Model record must exist in the *ca_model_def* table of the MDB for CA CMDB to access. If it does not exist, errors will be generated in the CA CMDB Export report, similar to the following:

```
<!--ERROR: Error setting attr 'model' on object
'nr:10B22E3C4236664D825BF7A72BE416DB' to value
'1330D45A5897D040B52391F821071FE4' NOT FOUND
1330D45A5897D040B52391F821071FE4-->
```

One other thing to note is that, if the model data does not exist in the CA CMDB, the server CI getting passed from CA Cohesion will not be created in the CA CMDB. If the servers are not created, then the relationships around the CIs will also fail, resulting in more errors in the export results. To avoid generating any errors before the model information has been created in CA CMDB, you can comment out the model attribute mapping line in the server CI mapping section. Comments are denoted by `<!--text here -->`.

1. Open the `cmdb_mapping.xml` file on the CA Cohesion server.
2. Search for the server CI attribute mapping section

```
<!--attribute mapping for server CI -- >
```

3. Comment out the model attribute mapping line:

```
<!--<attributeMapping CohesionCI="server" CohesionAttr="model"
CMDBAttr="model" CMDBFamily="*" />-->
```

Important! If you remove the Model attribute from the `cmdb_mapping.xml` file you must also comment out the Manufacturer attribute. If this is not done the Manufacturer value will be exported without the Model value and this will result in the following error when you try to load the file through GRLoader:

```
<!--Error setting attr 'manufacturer' on object
'nr:408CBF3D78187F4D889E58DE17055CEDED' to value
'45606AC2F4BFE84F8F1D438E1BFF2B4F' AHD05206: Not allowed to modify
this field -- >
```

Software Components

CA Cohesion associates each managed server with the software discovered on it. Note in the following example that “Microsoft Cluster Server” is listed twice - once for each server it was discovered on.

CA Cohesion® Application Configuration Manager

Servers

Server Management | Software | Discovery Profiles | Management Profiles | Access Profiles | Groups | Snapshots

0 Rows selected (111 total) View Option: Default View By: All

Component Name	Version	Qualifier	Server Name	Component State
<input checked="" type="checkbox"/> Microsoft Cluster Server	5.2	C:\WINDOWS	lodvm03ee32n2	Managed
<input type="checkbox"/> Microsoft Cluster Server	5.2	C:\WINDOWS	lodvm03ee32n1.vndom1.local	Managed
<input type="checkbox"/> Microsoft SQL Server - Client	9.00.1399.06	C:\Program Files\Microsoft SQL Server\80\Tools	lodvm03ee32n2	Managed
<input type="checkbox"/> Microsoft SQL Server - Client	9.00.3042.00	C:\Program Files\Microsoft SQL Server\80\Tools	lodvm03ee32n1.vndom1.local	Managed
<input type="checkbox"/> Microsoft SQL Server - Datafiles	9.00.1399.06	F:\Microsoft SQL Server\MSSQL1\MSSQL	lodvm03ee32n2	Managed
<input type="checkbox"/> Microsoft SQL Server - Datafiles	9.2.3042.00	F:\Microsoft SQL Server\MSSQL1\MSSQL	lodvm03ee32n1.vndom1.local	Managed
<input type="checkbox"/> Microsoft SQL Server 2005	9.2.3042.00	C:\Program Files\Microsoft SQL Server\MSSQL1\MSSQL	lodvm03ee32n1.vndom1.local	Managed
<input type="checkbox"/> Microsoft SQL Server 2005	9.00.1399.06	C:\Program Files\Microsoft SQL Server\MSSQL1\MSSQL	lodvm03ee32n2	Managed
<input type="checkbox"/> MySQL (Windows)		C:\FastESP\rdms	lodvm03ee32n2	Managed
<input type="checkbox"/> Windows	5.2	Microsoft Windows Server 2003 R2 SP2	lodvm03ee32n1.vndom1.local	Managed
<input type="checkbox"/> Windows	5.2	Microsoft Windows Server 2003 R2 SP2	lodvm03ee32n2	Managed

This information can be imported into the CA CMDB as software CIs and will be associated through a relationship to the server it was discovered on. Using the default `cmdb_mapping.xml` file, the software CI being imported into CA CMDB will be created once each time it is discovered on a server. If you are looking for a 1:1 ratio of software per server, the results may match what is desired.

The following example shows the result of software discovered on the two servers that were selected for import in the CA CMDB Export report. Note that, since the Microsoft Cluster Server software CI was discovered on two servers, it was imported into the CA CMDB twice - once for `lodvm03ee32n2` and once for `lodvm03ee32n1.vndom1.local`. The Configuration Item list in CA CMDB does not show that the “server” attribute is different for each CI record. Therefore, the software CI will appear as a duplicate record.

CA Service Desk / CA CMDB

Logged in as: Administrator (Log Out)

Service Desk | Knowledge | Administration | Reports | Change Order Schedule

File | View | Reports | Window | Help

Administration

- Archive and Purge
- Attachments Library
- CA CMDB
 - CI Classes
 - CI Families
 - CI List**
 - CI Models
 - CI Relationship List
 - CI Relationship Types
 - CI Service Status
- MDR Management
- Events and Macros
- Knowledge
- Notifications
- Options Manager
- Security and Role Management
- Service Desk
- Web Services Policy

Configuration Item List

Name	Class	Family	Serial Number
Microsoft Cluster Server v5.2 (C:\WINDOWS)	COTS	Software.COTS	
Microsoft Cluster Server v5.2 (C:\WINDOWS)	COTS	Software.COTS	
Microsoft SQL Server - Client v9.00.1399.06 (C:\Program Files\Microsoft SQL Server\80\Tools)	SQL	Software.Database	
Microsoft SQL Server - Client v9.00.3042.00 (C:\Program Files\Microsoft SQL Server\80\Tools)	SQL	Software.Database	
Microsoft SQL Server - Datafiles v9.00.1399.06 (F:\Microsoft SQL Server\MSSQL1\MSSQL)	SQL	Software.Database	
Microsoft SQL Server - Datafiles v9.2.3042.00 (F:\Microsoft SQL Server\MSSQL1\MSSQL)	SQL	Software.Database	
MySQL (Windows) (C:\FastESP\rdms)	Other Software Database	Software.Database	
MySQL Datafiles (C:\FastESP\data\rdms)	Other Software Database	Software.Database	
MySQL Datafiles (C:\FastESP\rdms\data)	Other Software Database	Software.Database	



Importing duplicate software CIs may not be desirable and can add complications to reporting. One option to import each software CI once and keep the necessary relationships intact is to comment out the `system_name` attribute line.

1. Open the `cmdb_mapping.xml` file on the CA Cohesion server.
2. Search for the attribute mapping section for software components.

```
<!--attribute mapping for Software Component CI -->
```

3. Comment out the `system_name` attribute mapping line.

```
<!-- <attributeMapping CohesionCI="component" CohesionAttr="system_name"
CMDBAttr="systems_name" CMDBFamily="*" />-->
```

Note: This will only work if the software installed on each server has the same install path.

Another option is to use the `NameQualifier` parameter in the Component Blueprint to remove the path indicated in the CI name. If defined it is displayed after the name in the discovered service tree view. Refer to the *CA Cohesion Application Configuration Manager Product Guide*, Chapter 5, section "Component Blueprint Building Reference Files", under the Category Descriptions sub section for instructions on modifying this parameter.

4. Comment out the server attribute mapping line:

```
<!-- <attributeMapping CohesionCI="component" CohesionAttr="host"
CMDBAttr="server" CMDBFamily="*" />-->
```

This will avoid the Server attribute value, shown in the following screen shot, from getting updated each time the software CI is discovered on a new server.

The screenshot displays the CA CMDB interface for a Microsoft SQL Server 2005 configuration item. The 'Attributes' section is expanded, showing a table with the following data:

Database ID	Portfolio	Environment	Type	Version	Server	Support Type	Support Start Date	Support End Date	Priority	Service Level Agreement	Leased or Owned?	Purchase Amount	Project Code	Contract Number
12334			Relational Databases	9.2.3042 00	w2k3base1									

Tips for Modifying the CMDB Class Mapping Section

The class mapping section of the `cmdb_mapping.xml` file includes a list of default Cohesion components and their default mappings to CA CMDB Classes. Any component not listed in this section will be assigned a default class value. For example, if a discovered software component is not listed in the xml file it will be given a default value of `Software.COTS` as defined in the following line:

```
<!--mapping to define the CMDB Class to map to -->
...
<classMapping CohesionCI="component" ComponentName="*"
CMDBFamily="Software.COTS" CMDBClass="COTS" / >
```

Note: When you specify or change a CMDB class mapping in the xml file, that class and family must already exist in CA CMDB or else the export will fail.

If there are software components that are not already listed in the xml file, or that need to be re-classified, the class mapping section can be modified. For example, to avoid classifying a discovered instance of Microsoft Active Directory as `Software.COTS` by default, add the following to the xml file.

1. Open the `cmdb_mapping.xml` file on the CA Cohesion server.
2. Find the section that defines the CMDB Class mapping section.

```
<!--mapping to define the CMDB Class to map to -- >
```

3. Add the following line:

```
<classMapping CohesionCI="component" ComponentName="Active
Directory Service" CMDBFamily="Security" CMDBClass="Application
Security" />
```

Additionally, to ensure the proper classification of discovered software CIs, Microsoft Cluster and Java Web Application, add the following two lines to the same class mapping section. Otherwise classification will default to `Software.COTS`.

1. Open the `cmdb_mapping.xml` file on the CA Cohesion server.
2. Find the section that defines the CMDB Class mapping section.

```
<!--mapping to define the CMDB Class to map to -- >
```

3. Add the following lines:

```
<classMapping CohesionCI="component" ComponentName="Microsoft
Cluster Server" CMDBFamily="Cluster" CMDBClass="Cluster" />
<classMapping CohesionCI="component" ComponentName="Java Web Application"
CMDBFamily="Software.Application.Server" CMDBClass="Application Server" />
```

In the following example you can see that the Microsoft Cluster Server v5.2 is listed once and is classified correctly with a class of Cluster, whereas before the component would have had two records, each being classified as `Software.COTS`.

CA Service Desk / CA CMDB

Logged in as: Administrator (Log Out)

Service Desk Knowledge Administration Reports Change Order Schedule

File View Reports Window Help

Administration
 Archive and Purge
 Attachments Library
 CA CMDB
 CI Classes
 CI Families
CI List
 CI Models
 CI Relationship List
 CI Relationship Types
 CI Service Status
 MDR Management
 Events and Macros

Configuration Item List

Search Show Filter Clear Filter Discover

Loq43 (C:\Program Files\CA\SC\Mdb\Windows)	COTS	Software.COTS
Microsoft Cluster Server v5.2 (C:\WINDOWS)	Cluster	Cluster
Microsoft SQL Server - Client v9.00.1399.06 (C:\Program Files\Microsoft SQL Server\80\Tools)	SQL	Software.Database
Microsoft SQL Server - Client v9.00.3042.00 (C:\Program Files\Microsoft SQL Server\80\Tools)	SQL	Software.Database
Microsoft SQL Server - Datafiles v9.00.1399.06 (F:\Microsoft SQL Server\MSSQL_1\MSSQL)	SQL	Software.Database
Microsoft SQL Server - Datafiles v9.2.3042.00 (F:\Microsoft SQL Server\MSSQL_1\MSSQL)	SQL	Software.Database

The relationships with the two cluster nodes are brought over correctly. We can see this when we click on the Microsoft Cluster Server v5.2 link and look at the CMDB Relationships tab.

CA Service Desk / CA CMDB

Logged in as: Administrator (Log Out)

Microsoft Cluster Server v5.2 (C:\WINDOWS) Configuration Item Detail

Edit Asset Viewer No Filter Visualizer

Name	Class	Family	Active?	Standard CI
Microsoft Cluster Server v5.2 (C:\WINDOWS)	Cluster	Cluster	Active	

Serial Number	Alt CI ID	Host Name	DNS Name	MAC Address

Asset? NO CI? YES

Notes
 Discover cluster server software on windows 2000/2003

11. Problems 12. Requests 13. Change Orders 14. Issues 15. Impact Analyzer
 6. Location Z. Contacts 8. Organizations 9. Service Contracts 10. Incidents
 1. Attributes 2. CMDB Relationships 3. Versioning 4. Inventory 5. Service

Add Relationship Refresh (\$) Impact Analysis

Related Configuration Items List

Relationship	Provider CI(s)	Family	Contact
is hosted by	LODVM03EE32N1 ()	Hardware.Server	
is hosted by	LODVM03EE32N2 ()	Hardware.Server	

Relationship Dependent CI(s)
 There are no dependent configuration items

Relationship	Peer CI(s)	Family	Contact
communicates with	LODVM03EE32N2 ()	Hardware.Server	
communicates with	LODVM03EE32N1 ()	Hardware.Server	

Note: Additional information on the attributes in the class mapping section can be found in Appendix C of the *CA Cohesion Application Configuration Manager Implementation Guide*.

Managed Hardware Reconciliation across Multiple Domains

Reconciling managed hardware discovered through Cohesion can be a challenge when the discovery crosses multiple domains. This reconciliation may become even more of a challenge when the hardware itself is discovered across multiple domains. When the default configurations are used, the discovery performed with CA Cohesion will create one server CI plus an additional CI for each Network Interface Card (NIC), Hard Drive, and File System component existing on the server. This is due to the fact that the actual server CI record is discovered through CA Cohesion with a Fully Qualified Domain Name (FQDN) in each domain. If nothing is done to reconcile the records, the data will be added into the CA CMDB for each instance of the discovered Name.

For example, the exchsvr01 server listed in the table below has four NIC cards, five hard drives and IPs in two domains. As a result, discovery identifies two servers, eight NIC cards, and 10 hard drives.

Discovered FQDN	Desired Name	Description
exchsvr01.dom1.ca.com	exchsvr01	Discovered by Cohesion in dom1.ca.com domain
exchsvr01.dom2.ca.com		Discovered by Cohesion in dom2.ca.com domain
exchsvr01.dom1.ca.com DISK-0	exchsvr01 DISK-0	Hard Drive
exchsvr01.dom2.ca.com DISK-0		Hard Drive
exchsvr01.dom1.ca.com DISK-1	exchsvr01 DISK-1	Hard Drive
exchsvr01.dom2.ca.com DISK-1		Hard Drive
exchsvr01.dom1.ca.com DISK-2	exchsvr01 DISK-2	Hard Drive
exchsvr01.dom2.ca.com DISK-2		Hard Drive
exchsvr01.dom1.ca.com DISK-3	exchsvr01 DISK-3	Hard Drive
exchsvr01.dom2.ca.com DISK-3		Hard Drive
exchsvr01.dom1.ca.com DISK-4	exchsvr01 DISK-4	Hard Drive
exchsvr01.dom2.ca.com DISK-4		Hard Drive
exchsvr01.dom1.ca.com DISK-5	exchsvr01 DISK-5	Hard Drive
exchsvr01.dom2.ca.com DISK-5		Hard Drive
exchsvr01.dom1.ca.com NetworkAdaptor-0	exchsvr01 NetworkAdaptor-0	Network Interface Card
exchsvr01.dom2.ca.com NetworkAdaptor-0		Network Interface Card
exchsvr01.dom1.ca.com NetworkAdaptor-1	exchsvr01 NetworkAdaptor-1	Network Interface Card
exchsvr01.dom2.ca.com NetworkAdaptor-1		Network Interface Card
exchsvr01.dom1.ca.com NetworkAdaptor-2	exchsvr01 NetworkAdaptor-2	Network Interface Card
exchsvr01.dom2.ca.com NetworkAdaptor-2		Network Interface Card
exchsvr01.dom1.ca.com NetworkAdaptor-3	exchsvr01 NetworkAdaptor-3	Network Interface Card
exchsvr01.dom2.ca.com NetworkAdaptor-3		Network Interface Card

While this result may be acceptable for managing discovered data in the Cohesion application, it may not be desirable as data input into the CA CMDB where the server CI needs to exist as one record with relationships to the four NICs and five Hard Drives. We can correct the mapping of Cohesion to CA CMDB data by doing the following:

1. Using the blueprint parameter in Cohesion to look up a server's Host Name so that the Host Name in Cohesion can be mapped to the Name in CA CMDB through Cohesion XML mapping file.
2. Removing the fully qualified domain name from the Hard Drive.

3. Adding the MAC address to the NIC blueprint and passing it into CA CMDB through the Cohesion XML mapping.

It is important to note that this situation only occurs when the NICs are registered with IP addresses that exist in multiple domains. If the IP addresses are in the same domain, the relationship between the server and its NICs will be properly passed into the CA CMDB.

Directions to implement the solution are as follows.

Reconcile Discovered Servers

The first step is to reconcile the two server records using the Hostname. The two discovered servers will be passed from CA Cohesion into CA CMDB by mapping the “Discovered Host Name” field to the CA CMDB “Name” field. By default, the Cohesion Blueprint uses the unqualified hostname as a parameter in Cohesion, and this parameter will be used as a mapping value in the Cohesion XML mapping file.

Step 1. Verify the Computer Name (short name) parameter in the Hardware Detail (Windows) Blueprint

1. Launch the Cohesion ACM UI and select the Blueprints tab
2. Select Hardware Details (Windows) and click the Edit/View button
3. Expand out the parameters folder and, under the directives folder, select Hostname
4. Close the window if the default settings are adequate.

Step 2. View the `cmdb_mapping.xml` file

1. On the Cohesion Server, browse to the `\Classes` folder under the Cohesion install directory. The default path is
2. Locate the `cmdb_mapping.xml` file in this directory and open it using an XML editing tool.
3. In the first section, `<!-- attribute mapping for server CI -->`, locate the mapping for “name”. For example:

```
<attributeMapping CohesionCI="server" CohesionAttr="Hostname" CMDBAttr="name"
CMDBFamily="*" />
```

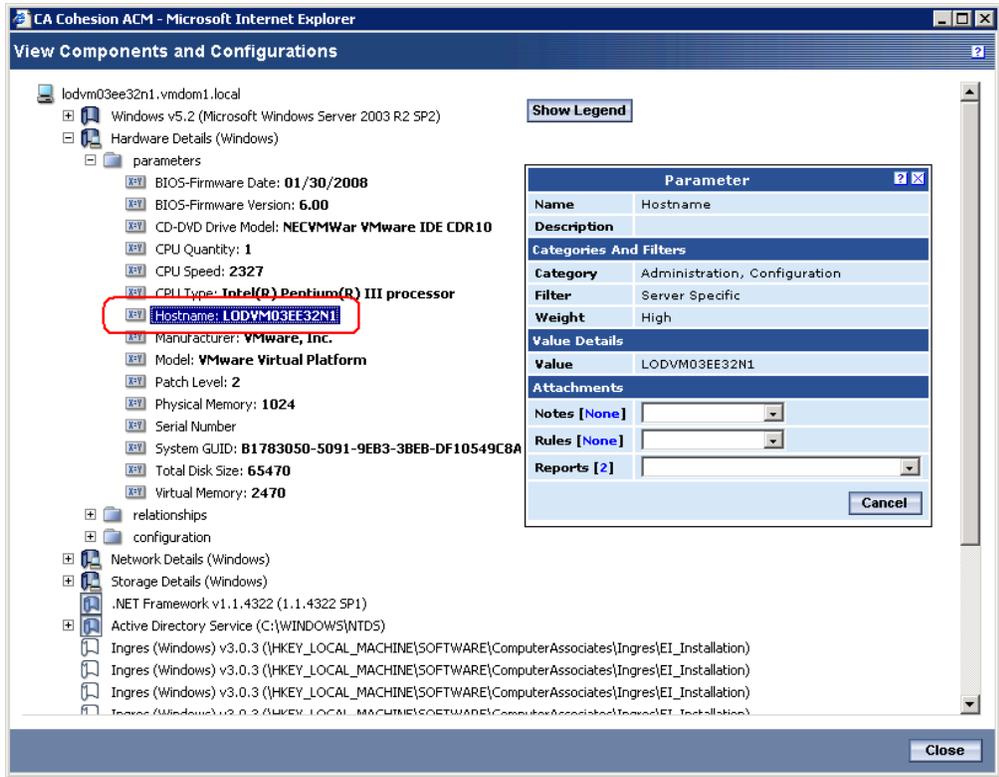
Note that the attribute `CMDBAttr="name"` is mapped to `CohesionAttr="Hostname"`

```

-<Mappings>
<!-- attribute mapping for server CI -->
<attributeMapping CohesionCI="server" CohesionAttr="ip_address" CMDBAttr="alarm_id" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="domainname" CMDBAttr="dns_name" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="host_id" CMDBAttr="federated_asset_id" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="mac_address" CMDBAttr="mac_address" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="make" CMDBAttr="manufacturer" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="model" CMDBAttr="model" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="serialno" CMDBAttr="serial_number" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="system_name" CMDBAttr="system_name" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="hostname" CMDBAttr="name" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="biosver" CMDBAttr="bios_ver" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="dvddrive" CMDBAttr="cd_rom_type" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="locstorcap" CMDBAttr="hard_drive_capacity" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="pmem" CMDBAttr="phys_mem" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="cpuspeed" CMDBAttr="proc_speed" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="arch" CMDBAttr="proc_type" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="ospatchlvl" CMDBAttr="security_patch_level" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="platform_name" CMDBAttr="server_type" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="vmem" CMDBAttr="swap_size" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="cpucount" CMDBAttr="number_proc_inst" CMDBFamily="*" />
<attributeMapping CohesionCI="server" CohesionAttr="niccount" CMDBAttr="number_net_card" CMDBFamily="*" />
<!-- attribute mapping for virtual CI -->
    
```

Step 3. Verify collection of Hostname parameter

1. Run a Discovery profile and/or Management Profile if you have not already done so.
2. Review the Hardware Detail for one or more servers and verify that the Hostname parameter in the Hardware Detail component shows the computer’s short name.



Reconcile Relationships

The next step is to modify the XML mapping file for proper reconciliation. By adding the MAC Address attribute in this file to CIs classified with the Network.Network Interface Card family, this enables the NIC cards to be properly reconciled through the CORA using MAC Address when it is passed into the CA CMDB.

To do this, use an XML editor to open the `cmdb_mapping.xml` file and search the file for the NIC class section. Add the following line to the NIC CI attribute mapping section:

```
<!--attribute mapping for NIC CI ->

<attributeMapping CohesionCI="server" CohesionAttr="mac_address"
CMDBAttr="mac_address" CMDBFamily="*" />
```

If you do not have the CA Cohesion 5.0 cumulative patch RO05523 applied, modify the following line to use "Hostname" for the CohesionAttr instead of "name":

```
<attributemapping CohesionCI="nic" CohesionAttr="Hostname"
CMDBAttr="name" CMDBFamily="*" />
```

Note that the CA Cohesion ACM cumulative patch RO05523 delivers an updated `cmdb_mapping.xml` file with the CohesionAttr="uname" instead of "name" for the CohesionAttr value.

Reconcile NIC Relationships

This step will add the physical address (`mac_address`) for each NIC found on a server. Note that CORA only supports a single NIC per server, so if a server has multiple NICs CORA will create a new CI for every NIC.

Add the Physical Address mapping line below to the NIC CI class mapping section to map the Cohesion NIC Physical Address to the CMDB Network Interface Card `mac_address` attribute.

```
<!-- attribute mapping for NIC CI -->
<attributeMapping CohesionCI="nic" CohesionAttr="Physical Address"
CMDBAttr="mac_address" CMDBFamily="*" />
```

Modify the Server CI class mapping section

For CA Cohesion ACM 5.0 installs **without** patch RO05523 applied, comment out the `mac_address` mapping line from the Server CI class mapping sections of the mapping file. This will prevent Cohesion from adding the MAC address entries to the Hardware.Server class CI and help CORA reconcile each entry once – instead of creating multiple entries for each MAC Address reconciled against.

1. Search for the Server CI class mapping section in the xml file:

```
<!--Attribute mapping for server CI -- >
```

2. Comment out the following line:

```
<!-- <attributeMapping CohesionCI="server"
CohesionAttr="mac_address" CMDBAttr="mac_address" CMDBFamily="*" />
-- >
```

For CA Cohesion ACM 5.0 installs **with** patch RO05523 applied, this step is unnecessary as the updated cmdb_mapping.xml file delivered with this patch does not have the mac_address attribute line under the server mapping section.

Modify the Virtual CI class mapping section

Comment out the mac_address mapping line from the Virtual CI class mapping sections of the mapping file. This will prevent Cohesion from adding the MAC address entries to the Hardware.Server class CI and will help CORA reconcile each entry once instead of creating multiple entries for each MAC Address it reconciles against.

1. Search for the Virtual CI class mapping section in the xml file:

```
<!--Attribute mapping for virtual CI -- >
```

2. Comment out the following line:

```
<!-- <attributeMapping CohesionCI="virtual"
CohesionAttr="mac_address" CMDBAttr="mac_address" CMDBFamily="*" />
-- >
```

Modify the Hard drive CI class mapping section

Update the xml file to use "Hostname" for the CohesionAttr instead of "name". If you use "name", each hard drive will be "discovered" in each domain. Using "Hostname" will ensure that the Hard Drive information is only discovered once.

1. Search for the hard drive CI attribute mapping section in the xml file:

```
<!--Attribute mapping for Hard drive CI -- >
```

2. Modify the following line to use "Hostname" instead of "name".

```
<!-- <attributeMapping CohesionCI="Hard Drive"
CohesionAttr="Hostname" CMDBAttr="name" CMDBFamily="*" /> -- >
```

Modify the File System CI class mapping section

If you do not have the CA Cohesion 5.0 cumulative patch RO05523 applied, modify the following line to use "Hostname" for the CohesionAttr instead of "name". If you use name, each file system will be "discovered" in each domain. Using "Hostname" will ensure that the File System information is only discovered once no matter how many domains the server exists in.

1. Search for the File System CI attribute mapping section in the xml file:



```
<!--Attribute mapping for File System CI -- >
```

2. Modify the following line to use "Hostname" instead of "name".

```
<!-- <attributeMapping CohesionCI="File System"  
CohesionAttr="Hostname" CMDBAttr="name" CMDBFamily="*" /> -- >
```

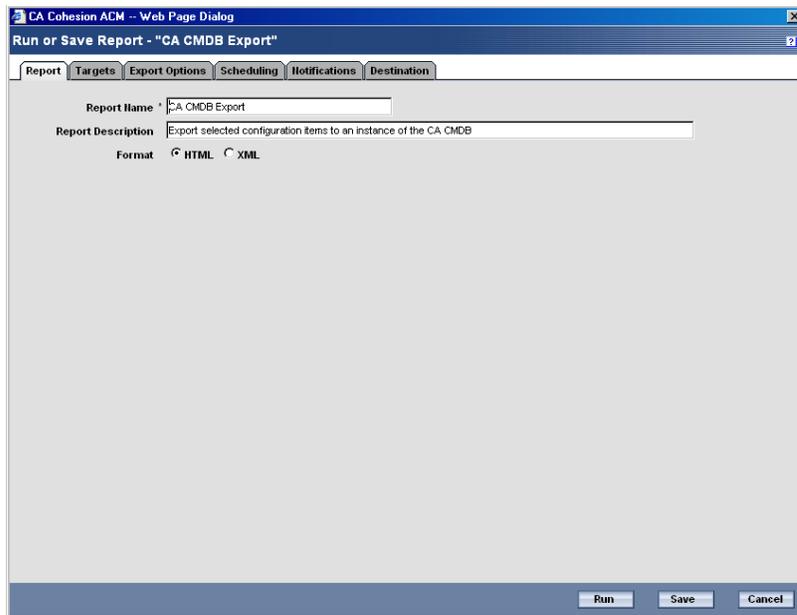
For CA Cohesion ACM 5.0 installs **with** patch RO05523 applied, the CohesionAttr value uses "uname" instead of "name".

Notes for Managing CIs with CA CMDB and CA Cohesion

The instructions in the previous section are provided to help customers modify the cmdb_mapping.xml file in order to get desirable results for CI data being imported into the CA CMDB from CA Cohesion. Not all customers have a business requirement to manage NIC's and Hard Drives as separate CIs.

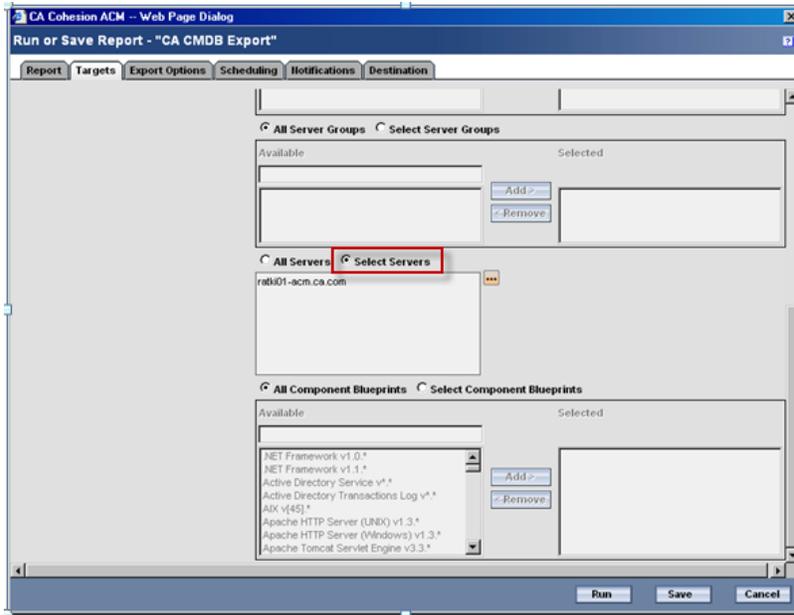
Use the CA CMDB Export report to export the discovered CIs from CA Cohesion into the CA CMDB.

3. Log into CA Cohesion and launch the UI
4. Go to Report templates tab and click on CACMDB Export report.
5. In the Report tab, specify the name of the report and the format.

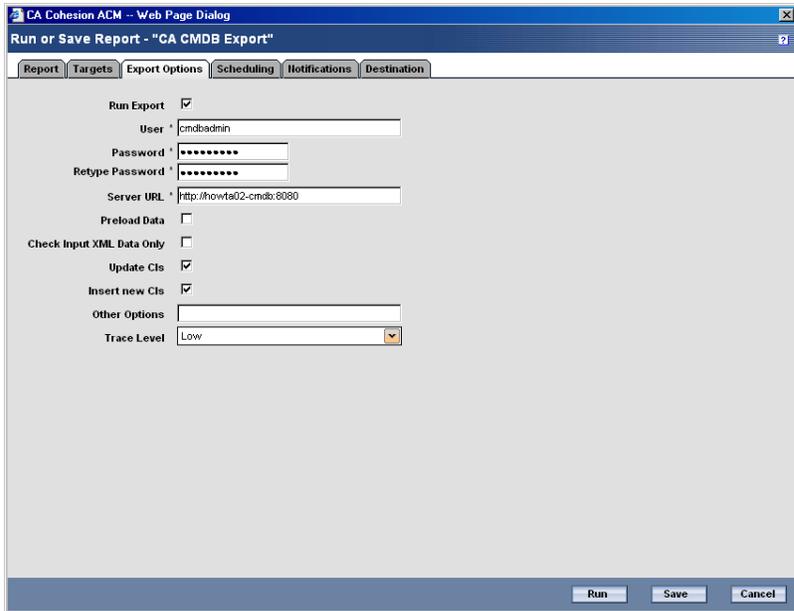


The report name can be anything. The Format selecting indicates whether you want the results displayed in HTML or XML.

6. Click the Targets tab and specify which objects you want to export from CA Cohesion. In the following example only one server from CA Cohesion is going to be exported into CA CMDB:



7. Click the Export Options tab and specify the connection details to the CA CMDB server:



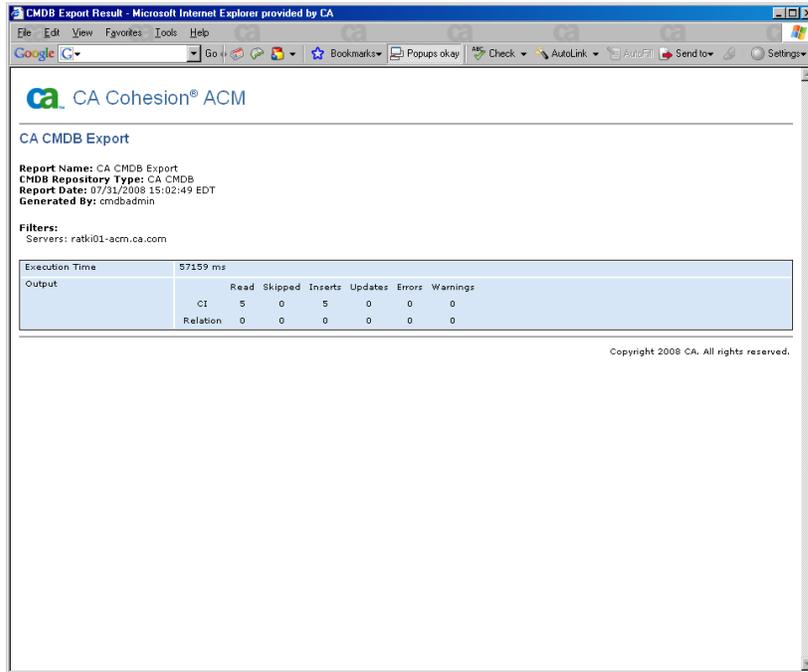
The User ID provided in the **User** field must identify a user in CA CMDB who has the rights to create CIs.

In the **Server URL** field specify the CA CMDB host URL.

8. Check the **Run Export** option – if this is not checked, CIs will not be created on the CMDB server.
9. Optionally, click the **Scheduling** tab to provide a schedule for when this export report should run.
10. Click **Save** to save the report.



11. Click Run to run the report. Once the report runs, you should see an export report similar to the following, showing you the details of how the export executed:



CA Cohesion® ACM

CA CMDB Export

Report Name: CA CMDB Export
CMDB Repository Type: CA CMDB
Report Date: 07/31/2008 15:02:49 EDT
Generated By: cmdbadmin

Filters:
Servers: ratki01-acm.ca.com

Execution Time	57159 ms						
Output	Read	Skipped	Inserts	Updates	Errors	Warnings	
CI	5	0	5	0	0	0	
Relation	0	0	0	0	0	0	

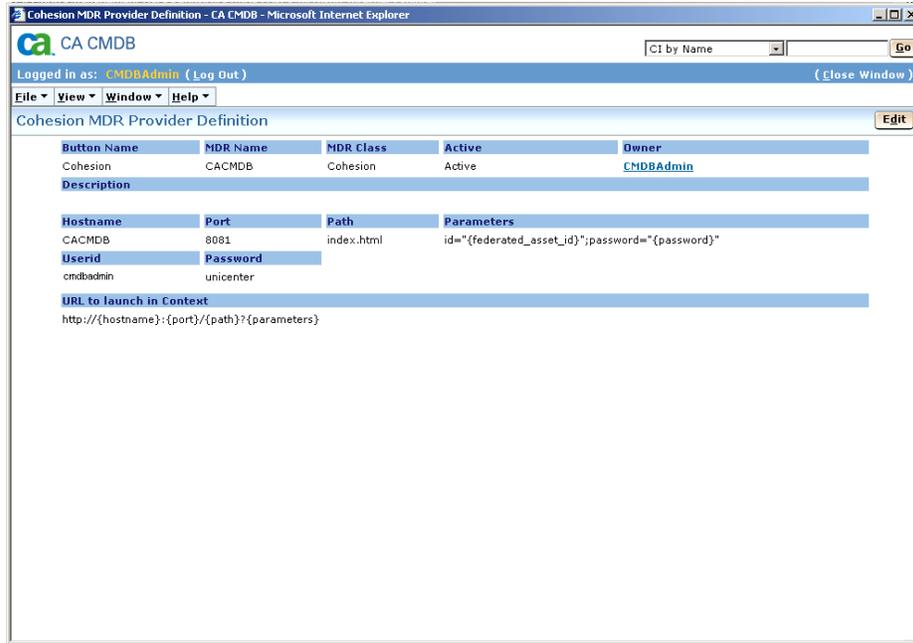
Copyright 2008 CA. All rights reserved.

Launch CA Cohesion in Context for a CI in CA CMDB

Only data that is relevant to supporting a service should be entered into the CMDB. In most cases this does not include NICs, File Systems, or Hard Drives. One of the integration features between CA CMDB and CA Cohesion is the ability to launch the CA Cohesion UI in context from a CI definition in the CA CMDB, enabling you to access that additional level of CI detail.

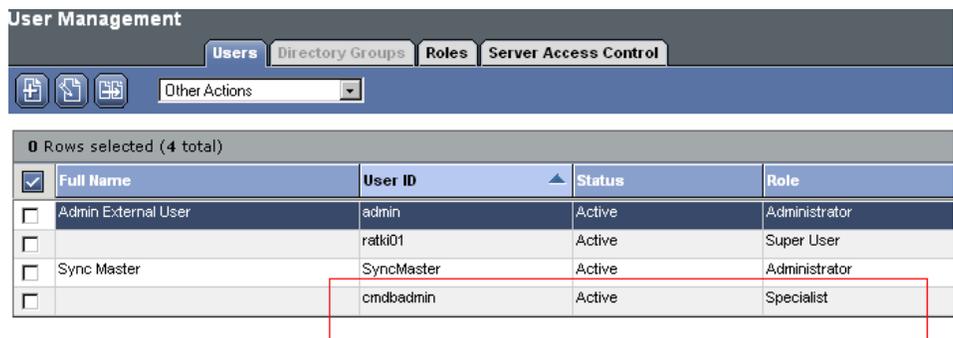
To enable launch in context you need to define Cohesion as an MDR. To do this:

1. Launch the CA CMDB Administrative UI and click on the MDR Launcher.
2. Select MDR Data Providers.
3. Click Create New



Provide the following details for the Cohesion MDR:

- **MDR Name:** This has to be the name of the Cohesion server. Ensure that there is a level of change control to manage the CIs going forward.
- **MDR Class:** This will always be “CA Cohesion” for this type of integration.
- **Hostname:** This is the host of the MDR, which will be the CA Cohesion server you are connecting to.
- **Userid** – This is a CA Cohesion user ID with at least the Specialist role:



Note: After the initial load of CI data, customers should have at least Specialist rights. The reason for using a different user ID is to prevent CA Cohesion from kicking you out of the session you are already logged on as.

- **Password** – this is the secret password you have defined in the cendura.properties file (see below).

```
# -- Configure One-Click Authentication --
com.cendura.security.oneclickauth.secret=unicenter
```



```
com.cendura.security.oneclickauth.scheme=  
com.cendura.security.oneclickauth.user=cmdbadmin
```

Path, Parameters and URL are not required for CA Cohesion.

You will now be able to launch CA Cohesion in context to a CI from the CA CMDB UI by clicking on the Cohesion button that appears on the Attributes tab for that CI.

Important SSL Considerations

When CA CMDB, or CA CMDB with CA Service Desk Manager, is configured to use the Secure Sockets Layer (SSL), you need to configure CA Cohesion so that it can successfully communicate with the secured instance of CMDB. Otherwise, when you attempt to run a report from CA Cohesion in order to export CIs, GRLoader will be unable to log on to the SSL enabled CMDB Server.

To enable GRLoader to work correctly, Java needs to be able to authenticate the server certificate for the Web Services. This means that you will have to create a certificate, add it to the Java's cacerts (trusted key store), then pass the URL of the https server to GRLoader.

Create Certificate

To create the certificate, first open a command prompt and change directories to the JRE install location of the CMDB Server directory. By default this is the following:

```
C:\Program Files\CA\SharedComponents\JRE\1.4.2_06
```

Execute the following command:

```
bin\keytool -export -alias <insert alias here> -keystore  
<storename> -rfc -file <insert .cer filename> -storepass  
<password>
```

For example:

```
bin\keytool -export -alias tomcat -keystore .keystore -rfc -file  
tomcat.cer -storepass changeit
```

The next step is to configure java to use the certificate.

Add Certificate to Java Trusted Key Store

In order for GRloader to communicate to the https server, java needs to be configured to use the certificate you created in the previous step. GRLoader uses the following copy of java on Cohesion Server:

```
C:\Program Files\CA\Cohesion\Server\jsdk\jre
```

However, you need to run the following command on the cacerts file in the following directory:

```
C:\Program Files\CA\Cohesion\Server\jsdk\jre\lib\security
```

Therefore, you will need to import the .cer (certificate file) file you just created on the CMDB Server to cacerts directory on each of Cohesion Server that will run reports against it. To do this, first open a command prompt and change directories to the JRE install location. By default, this is the following:

```
C:\Program Files\CA\Cohesion\Server\jsdk\jre
```

Then, enter the following command:

```
bin\keytool -import -alias <insert alias> -file <insert .cer filename> -keystore <storename> -storepass <password>
```

For example:

```
bin\keytool -import -alias tomcat -file tomcat.cer -keystore lib\security\cacerts -storepass changeit
```

Pass the URL to GRLoader

Finally, in order to run reports from Cohesion Server against the https CMDB server you must modify the 'Server URL' attribute in export option tab to the following:

```
<https server url:port>
```

For example:

```
https://localhost:8443
```

For additional details on configuring GRLoader to work in an SSL environment, consult technical document TEC428625 which is available on support.ca.com.

Reference Documentation

For more information about this integration, consult the following sources:

- Integrating Cohesion with CA CMDB

Note: For additional information about the Cohesion-CA CMDB integration, see the CA Cohesion ACM Implementation Guide.
- Importing CIs from a Cohesion MDR

Note: For information on how to import CIs from a Cohesion MDR, see the online help that is available from Cohesion ACM Reports Report Templates tab.
- Launch-in-Context for Cohesion MDRs



For launch-in-context integration to work best with CA Cohesion ACM, we recommend that you use the CA CMDB Administration tab to define the Cohesion MDR before running the Cohesion CMDB report.

Note: Because CA Cohesion ACM does not support a unique Federated Asset ID for NIC or File System CIs, Cohesion does not support MDR Launcher for NIC or File System CIs. Therefore, a Cohesion-based NIC or a File System CI does not display an MDR launch button even when it was imported successfully.

- Additional Blueprints

Additional Blueprints can be found at the following link:

https://support.ca.com/phpdocs/0/common/impcd/r11/Cohesion/Cohesion_Frame.htm

This includes “Light” Component Blueprints which increase performance and scalability by only discovering the data that is useful to CA CMDB customers. In general, all files, registry entries, and all configuration data that does not contain relationships are removed from the Light Component Blueprints. They also retain the discovery parameters and the data (CIs, attributes, and relationships) that is exported to CA CMDB.

- MAC Address Normalization

See Chapter 3 in the *CA CMDB Technical Reference Guide* for information on MAC address behavior and GRLoader parameters that can be used to enable or disable MAC normalization.

Chapter 5: Integrating with CA SPECTRUM

This chapter discusses how CA CMDB and CA SPECTRUM can be configured to work together. The following key topics are presented:

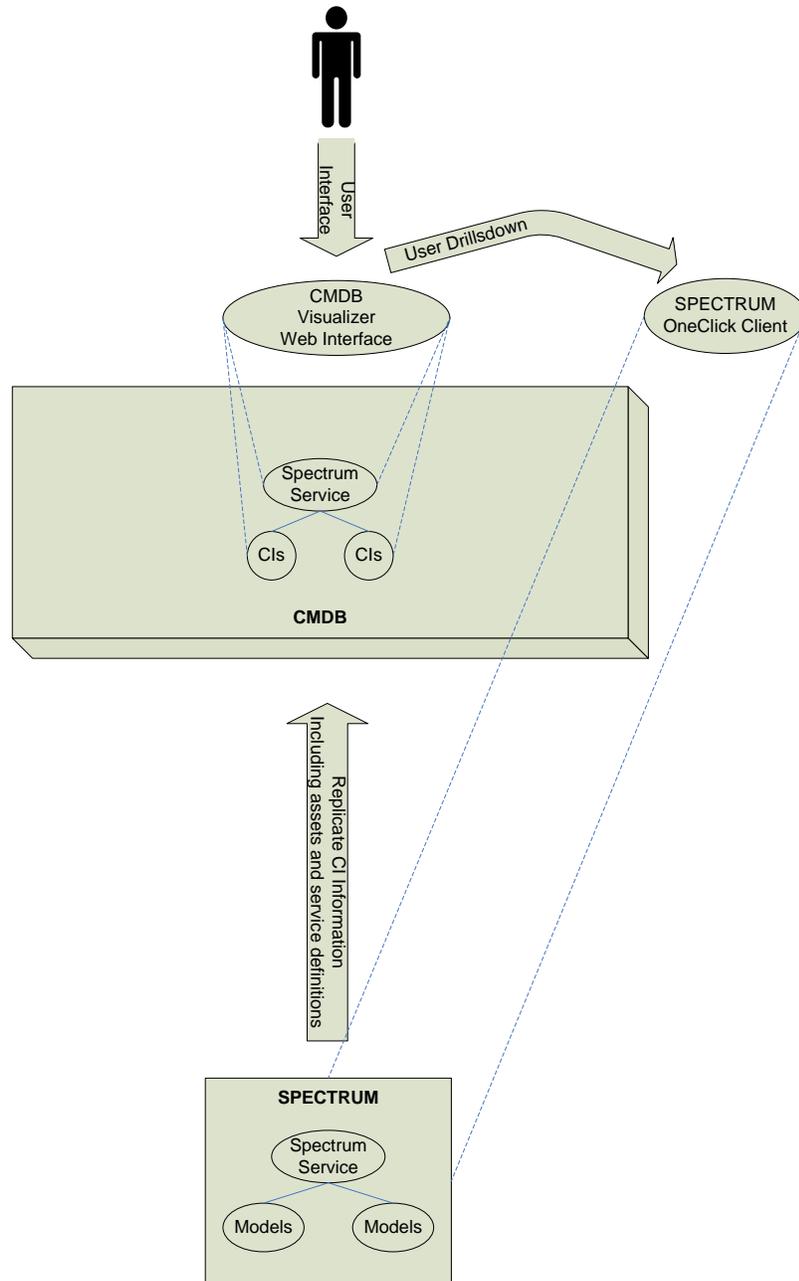
- Overview and value of the integration
- How the CA SPECTRUM integration works
- How to install the integration
- Reference Documentation

Note: The CA SPECTRUM to CA CMDB integration requires a license to use the CA SPECTRUM Modeling Gateway Toolkit. With CA SPECTRUM 9.0 this is provided as part of the Standard and Premium Suites, however, if you are licensed for either the Remote Operations or Foundation Suites, it will need to be purchased as an additional module.

Overview and Value of the Integration

Beginning with CA SPECTRUM r8.1 SP1, CA provides uni-directional integration from CA SPECTRUM to CA CMDB that enables you to populate the CMDB with the relevant CA SPECTRUM discovered network resources that are crucial to helping you to make your business services available for your end-users. Defining CA SPECTRUM as a CA CMDB MDR gives you a broader view of the resources used in your environment, and, particularly when coupled with information provided by other MDRs, supports root cause analysis and change impact analysis scenarios.

The true value of this integration lies in its ability to identify the business services supported by and the interrelationships between the business critical network resources managed and monitored by CA SPECTRUM. This knowledge can reduce mean time to repair by supporting root cause analysis, and can help you minimize the future disruption of business services by providing a clearer picture of the potential business impact of changes to the CIs that support those services.



This chapter summarizes the integration between the two components and describes CA’s current best practice recommendations for designing and implementing the integration. It also discusses how other potential integrations, such as CA Network and Systems Management (CA NSM) or CA Cohesion ACM, may provide information about the same CIs that are exported from CA SPECTRUM. For this reason, proper reconciliation is needed to prevent creation of duplicate CIs in the CA CMDB. You are also strongly advised to engage CA Services for additional guidance in deploying this integration.

As with all other chapters in this guide, this chapter assumes a new implementation and a working knowledge of both solutions.

How the CA SPECTRUM Integration Works

Although these two solutions can stand alone, this integration was designed to elevate and extend the value of each solution. The joint solution offers more sophisticated root cause analysis and change impact analysis capabilities to both CA SPECTRUM customers and to CA Service Desk Manager or CA CMDB customers.

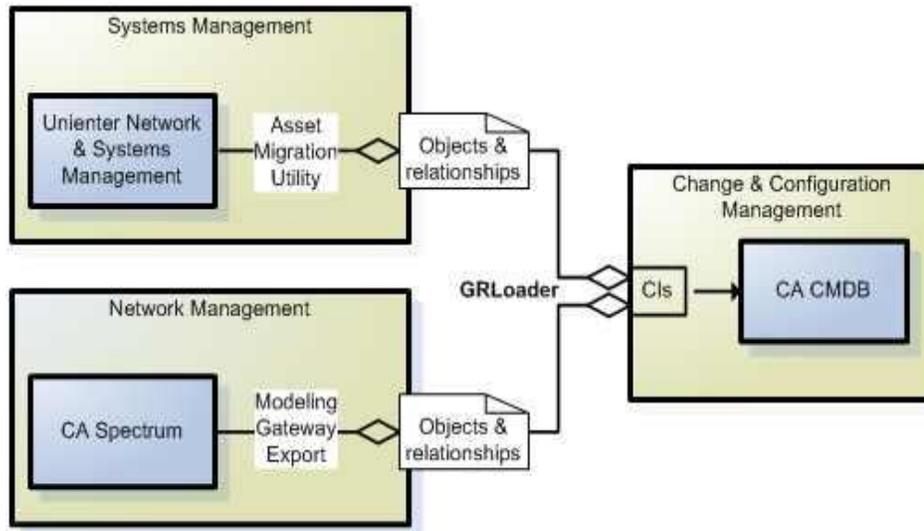
For CA SPECTRUM r8.1, support for CMDB integration (uni-directional) requires application of the appropriate Patches and Service Pack. For CA SPECTRUM r9.0, CMDB integration is included as part of the standard installation media. In addition to enabling CIs, along with their attributes and relationships, to be exported to (GRLoader) XML format and then directly loaded into the CMDB, when CA SPECTRUM r9.0 is also integrated with CA Service Desk, this allows you to correctly link any incidents created in CA Service Desk to the relevant CI in the CMDB.

Note: For more information on the integration between CA Service Desk and CA SPECTRUM consult the *CA Service Desk SPECTRUM Integration Guide (5178)*.

Before you Begin

Before you implement the integration between CA SPECTRUM and CA CMDB there are some important considerations that need to be carefully evaluated:

- Are there different requirements between the initial CI load situation and the subsequent CA SPECTRUM export\CA CMDB import, and how frequently must these updates be made?
- Which types of CIs will be exported from CA SPECTRUM and imported into the CMDB?
The default export configuration limits the exported CI types to CA SPECTRUM device models and service models
- Which CI attributes and relationships per CI type are going to be exported from CA SPECTRUM and imported into the CMDB?
- Which other sources, typically referred to as Management Data Repositories (MDRs), are going to contribute information regarding the same CIs exported from CA SPECTRUM?



- How is proper reconciliation of a CI object ensured when attributes of the same object are imported from other MDRs?

For example, if CA SPECTRUM service objects are imported into the CMDB as CIs of the CMDB Service family, it will not reconcile with any existing CIs or if the same CI has other MDRs with the default configuration. In these situations, it is recommended that you modify the CA SPECTRUM export prior to loading the CIs into the CMDB.

- How are updates of attributes controlled when the same attributes may potentially be imported from different sources?

For example, should you be able to update the exact same attributes from different sources, and if yes, which source should take precedence?

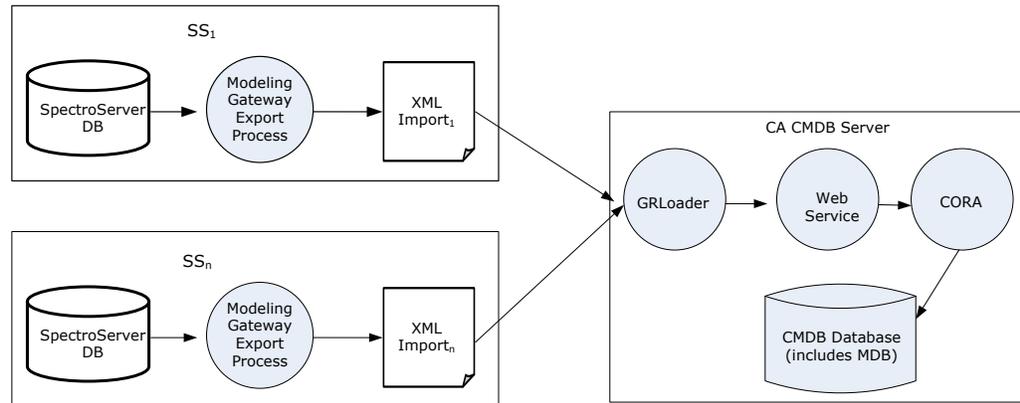
- How are updates of relationships between CIs controlled when relationships are imported from multiple sources?

- What are the MDR relationships and launch capabilities when importing from different sources?

- How has the CA SPECTRUM architecture been deployed? For example:

- A single SpectroSERVER propagating data into a single CA CMDB
- Multiple SpectroSERVERs in a single CA SPECTRUM cluster propagating data into a single CA CMDB. **(Note:** Some CIs will overlap on the SpectroSERVERs.)
- Multiple stand-alone SpectroSERVERs (non-distributed) propagating data into a single CA CMDB. **(Note:** Some CIs may overlap on the SpectroSERVERs.)

Consider the following process flow:



In this example you can see the following progression:

1. A scheduled job that is external to CA SPECTRUM (such as a Windows Scheduler or Cron job) is used to periodically run the SPECTRUM Modeling Gateway Export utility with a full export of all models and relationships available for export.
2. This creates an XML file containing all models and relationships available for export into the CMDB.

Note: This does not include any “delta” information, such as what has been added, changes or deleted.

3. GRLoader is then used to load the converted file into the CMDB. Typically, GRLoader is called by the same script used to call the Modeling Gateway Export (in other words, through the script run by the scheduled job or through a scheduled job on the CA CMDB Server).

Note: GRLoader must be run from the CA CMDB server in the initial release.

4. Once the data is submitted via GRLoader, the CMDB runs the imported data through the CORA reconciliation process.

Note that multiple SpectroSERVERS (as depicted in the example) can run the export utility. As a result, the process must account for data coming from more than one SpectroServer – in other words, more than one MDR for CA SPECTRUM.

With these considerations in mind, the CA SPECTRUM – CA CMDB integration has to be properly designed and the CA SPECTRUM default mapping file may have to be changed to match the requirements of the integration. Execution of the CA SPECTRUM Modeling Gateway export and the CA CMDB GRLoader import processes will also have to be scheduled based on the CMDB update frequency requirements.

The following CA SPECTRUM export\CA CMDB import options are available by default:

- "Device-to-Device". This includes:
 - Devices that are connected through ports
 - Devices that are directly connected (no ports)
 - Devices that are connected through a fanout
 - Devices that are connected through a WA_Link (wide area link)
- "Service-to-Device". This includes:
 - Devices that are monitored directly by a service
 - Devices with a port that is monitored directly by a service
 - Devices inside a resource monitor that are monitored by a service
 - Devices with a port inside a resource monitor that is monitored by a service
 - Devices dynamically inside a global collection that is monitored by a service
 - Devices statically inside a global collection that is monitored by a service
- "Service-to-Service". This includes:
 - Services that are monitored directly by a service
 - Services inside a resource monitor that are monitored by a service
 - Services dynamically inside a global collection that is monitored by a service
 - Services statically inside a global collection that is monitored by a service

Installing the Integration

This section provides information on the steps required to implement the integration.

Prerequisites

The prerequisites must be met in order to support the integration between CA SPECTRUM and CA CMDB:

- The CA SPECTRUM Modeling Gateway toolkit has been installed

Although it is included in CA SPECTRUM r9.0, the CA CMDB integration components are not installed with CA SPECTRUM 8.1. To obtain the Modeling Gateway Export, you must contact CA Support and open a ticket requesting the Modeling Gateway Export utility. Make sure to note that you need the utility for CA SPECTRUM 8.1 and integration with the CA CMDB. Once the components are downloaded, they must be copied into the appropriate areas on the SpectroSERVER machine:

```
modelinggateway81.jar to <$SPECROOT>/lib
```

modelinggatewayresource.xml to <\$SPECROOT>/SS-Tools (rename the file to .modelinggatewayresource.xml)

.cmdbresource.xml to <\$SPECROOT>/SS-Tools

.cmdbresource.dtd to <\$SPECROOT>/SS-Tools

(UNIX only) modelinggateway to <\$SPECROOT>/SS-Tools

(Windows only) modelinggateway.bat to <\$SPECROOT>/SS-Tools

- The CA SPECTRUM – CA CMDB mapping file has been modified according to the requirements noted earlier (see following section for further details)
- CA Service Desk Manager or CA CMDB has been installed, including its GRLoader executable
- CA SPECTRUM OneClick client (Java-based client) must be installed on users' workstations. This also requires installation of JRE, which can be downloaded from the CA SPECTRUM OneClick server via a link on the OneClick server web page.

CMDB Mapping File Modifications

Some modifications will need to be made to the default mapping file. For example, the default mapping causes some exported Cisco routers from CA SPECTRUM to be classified as 'Network Switch' CIs in the CMDB, if CA SPECTRUM can distinguish between routers and switches in the infrastructure it may be necessary to change the mapping file. Here you can see an example of the mapping file:

```

28
29 <!-- This file defines the mappings between the SPECTRUM models and
30      the CMDB classes. -->
31 <!DOCTYPE CMDBImportExportResourceFile SYSTEM "../cmdbresource.dtd">
32 <CMDBImportExportResourceFile>
33
34 <!-- Defines the relationship types written to the CMDB. The three types of
35      relationships relationships
36      that are recognized in the integration are as follows:
37      1) Device-to-Device: One device connects to another device (bilateral)
38      2) Service-to-Device: A service has a device as a monitored resource
39      3) Service-to-Service: A service has a service as a monitored resource -->
40 <CMDBRelationshipMappings>
41
42 <!-- represents a device connecting to another device -->
43 <CMDBRelationshipMapping>
44   <SPECTRUMRelationship>Device-to-Device</SPECTRUMRelationship>
45   <CMDBRelationship>connects to</CMDBRelationship>
46 </CMDBRelationshipMapping>
47
48 <!-- represents a service with a device resource -->
49 <CMDBRelationshipMapping>
50   <SPECTRUMRelationship>Service-to-Device</SPECTRUMRelationship>

```

Note: The mapping file is <\$SPECROOT>/SS-Tools/.cmdbresource.xml

The default mapping also causes servers exported from CA SPECTRUM to be classified as "Other Operating System". As a result, you may have to change the mapping file if servers are included in the scope of CIs that have CA SPECTRUM as an MDR.

The following sections discuss how to configure and use the integration, based on best practices.

CA SPECTRUM Customizations

The following CA SPECTRUM changes and customizations are recommended:

- Create Services in CA SPECTRUM first
- Use fully qualified domain names (FQDN)
- Point to CA Service Desk URL (if CA CMDB is deployed with CA Service Desk)
- Use CA SPECTRUM Integration Script

Create Services in CA SPECTRUM First

Services should be created in CA SPECTRUM first, prior to integrating with CA CMDB. Before creating a service, make sure to carefully consider what the service represents, what resources (SPECTRUM models) impact the viability of the service, and what level of service viability (or service health) should be inferred from the condition of the resources that support the service.

A CA SPECTRUM service model, or service, is an abstract entity. It can represent any process or group of processes identified as being supported by a group of resources managed in CA SPECTRUM. In fact, any conceivable activity supported by infrastructure resources managed in CA SPECTRUM can be modeled as a 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 can be modeled as services in CA SPECTRUM.

Enabling management of resources as services provides a better understanding of fault impacts. Consider a managed resource such as a router. It can be accurately but narrowly defined as “a device that forwards data from one network to another”. From a service perspective, however, it is an indispensable component that supports internet business activities. If the router’s performance is compromised, activities that are dependent on it are likely to be compromised as well. Consider also the other resources that operate in concert with the router to support those activities. How does their operational status affect those activities? Consolidating these resources as a service enables you to monitor how the resources collectively support the business activities that infrastructure end-users depend on. It also helps identify compromised or at-risk services that can be addressed before they become unavailable.

In addition to the resources that constitute a service, a service contains a policy that infers a *service’s viability* (its service health) based on the collective status of its resources. A policy monitors a single common attribute for all service resources. It consists of an *attribute map*, which associates resource attribute values to service health values, and a *rule set*, which calculates the service’s health value based on the attribute map associations.

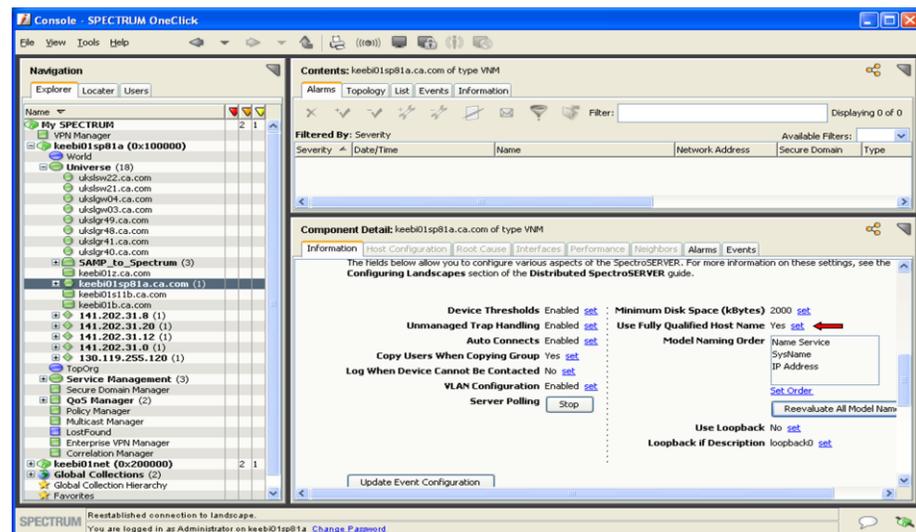
A service can also contain one or more *resource monitors* instead of a single policy. A resource monitor monitors a single resource attribute value common to all or to just a subset of the resources for a service. Typically, a set of resource monitors are defined and deployed to monitor multiple resource attributes for a service.

A service's viability, which identifies its ability to support the business process it represents, can be measured by a Service Level Agreement (SLA). This is represented by a CA SPECTRUM model that indicates when a service is or is not compliant with service obligations guaranteed by the service provider. A CA SPECTRUM model can also represent a service associated with one or more service customers, who depend on the service and are typically a party to the SLA contract. This enables customers to monitor their services from the customer models created for them.

For more information on creating and managing services in CA SPECTRUM refer to Chapter 2, "Creating and Managing Services" in the *CA SPECTRUM Service Manager User Guide*.

Use Fully Qualified Domain Names (FQDN)

Configuring SPECTRUM to use Fully Qualified Domain Names (FQDN) can improve the likelihood of matching resources and, therefore, of correctly reconciling CIs imported from CA SPECTRUM with those imported from other MDRs, such as CA Network and Systems Management (NSM) and CA Cohesion.



CA SPECTRUM Integration Script

The spectocmdb.pl script is used to automate the export of CA SPECTRUM model information from a SpectroSERVER and import the data into the CA CMDB. Although spectocmdb.pl can be installed in any directory the best practice is to install it under `$SPECROOT\custom\spectocmdb`. This will ensure that the script will be protected/retained during subsequent CA SPECTRUM patches or upgrades. This directory must contain the following two subdirectories:



- The 'log' directory contains the log files created by the script when it is run. It can be used for troubleshooting/monitoring.
- The 'export' directory contains the export files created by the CA SPECTRUM Modeling Gateway export.

Note: Both of these directories are maintained as part of the script to make sure they do not grow out of control.

The spectocmdb.pl script can be executed with a complete configuration file or a complete set of qualifiers. To run with a completed configuration file, execute the following:

```
perl spectocmdb.pl
```

Note that the configuration file must be located in the same directory as the script and named "spectocmdb.cfg". The configuration file parameters are NOT case-sensitive. The parameters are:

- GrloaderDir
- VNM
- CmdbUser
- CmdbPassword
- CmdbServer
- CmdbServerPort
- CmdbServerProtocol
- MaxLogSize
- NumRetentionDays

To run without a configuration file, execute as follows:

```
perl spectocmdb.pl -grloaderdir
c:/win32app/spectrum/custom/grloader -javabin
c:/progra~1/java/jre1.5.0_12/bin/java -vnm ss01 -cmdbuser
Administrator -password password01 -cmdbserver cmdb01 -
cmdbserverport 8080 -cmdbserverprotocol http -maxlogsize 1048576
-numretentiondays 14
```

The command-line parameters override the configuration file settings and are NOT case sensitive. The command-line parameters are:

Parameter	Description
-grloaderdir	Full directory path to where GRLoader has been installed.
-vnm	Name of SpectroSERVER that models should be exported from.
-cmdbuser	Username used to login to CA CMDB server with GRLoader.



-cmdbpassword	The password for the username used to login to the CA CMDB server with GRLoader.
-cmdbserver	The hostname or IP address of the server the CA CMDB server is running on.
-cmdbserverport	The web server port that GRLoader should connect to. This is typically 8080.
-cmdbserverprotocol	This is either 'http' or 'https'. This is the protocol used to connect to the CA CMDB server.
-maxlogsize	The maximum size in bytes that the log file for the script can grow to. The current log file and one previous log file will be retained.
-numretentiondays	The number of days to retain the exports that have been extracted from SPECTRUM.
-help	This prints out the abbreviated help screen.
-man	Prints out the full manual information. (this output)
-debug	This turns on debugging for the SPECTRUM Modeling Gateway export and the GRLoader import (trave level 10).

The Perl script was tested with the Perl version distributed as part of CA SPECTRUM 8.1 SP1 (Perl v5.8.7 built for cygwin-thread-multi-64int).

CA Service Desk Manager and/or CA CMDB Customizations

The following CA CMDB\CA Service Desk Manager actions are recommended to support this integration:

- Correctly configure MDR definition for each CA SPECTRUM object and relationship data source (as shown in the next section)
- Use GRLoader to load data from remote MDRs
- CA Service Desk Integration modifications for CA SPECTRUM r8.1. Information on these modifications can be found in the *CA SPECTRUM and CA Service Desk Integration Guide*.

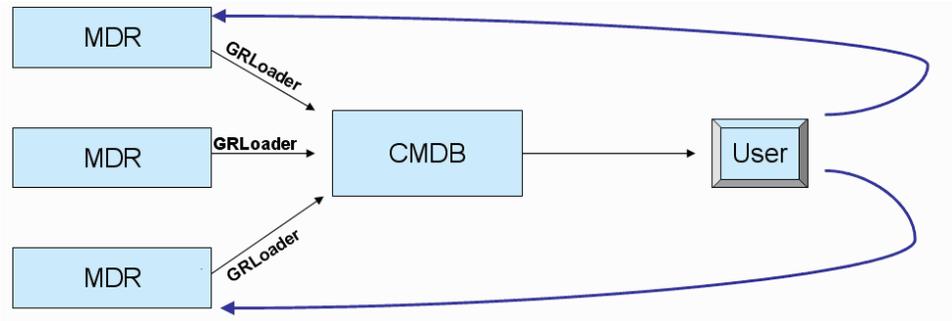
Defining the SPECTRUM MDR

Before you can import managed object and relationship information from CA SPECTRUM into the CA CMDB you need to first define SPECTRUM as an MDR Provider. A separate MDR definition is created for each CA SPECTRUM object and relationship data source. Environments with several SpectroSERVER's require separate MDR definitions. If the SpectroSERVER's are all members of a single CA SPECTRUM distributed SpectroSERVER environment, only two MDRs must be defined for CA SPECTRUM:

- CA SPECTRUM "OneClick" MDR
- CA SPECTRUM "Service Dashboard" MDR



GRLoader provides facilities for importing and loading CIs and also for associating CIs with their origins. In addition, by using the MDR Launcher capability when viewing a CI in the CA CMDB, you can navigate seamlessly back into the system from which the CI originated, as shown in the following diagram.



To enable launch of CA SPECTRUM OneClick from the CA CMDB MDR Launcher, do the following:

1. Click the CA CMDB Administration tab.
2. Expand the MDR Launcher section and click MDR providers.
3. In the middle window, click Create New. Provide information for the following fields:
 - **Button Name:** OneClick
 - **MDR Name:** OneClick
 - **MDR Class:** SPECTRUM
 - **Active:** Active
 - **Owner:** CMDBAdmin
 - **Description:** Context launch for CA SPECTRUM OneClick.
 - **Hostname:** <your CA SPECTRUM OneClick server host>
 - **Port:** <your CA SPECTRUM OneClick server port>
 - **Path:** /spectrum/oneclick.jnlp
 - **Parameters:** topology={federated_asset_id}
 - **URL to launch in Context:** http://{hostname}:{port}/{path}?{parameters}
4. Click Save.
This will generate a message indicating that the MDR was created.
5. Click Close.

Here you can see an example of the OneClick MDR provider definition:

Button Name	MDR Name	MDR Class	Active	Owner
OneClick	OneClick	SPECTRUM	Active	CMDBAdmin

Description
Context launch for SPECTRUM OneClick

Hostname	Port	Path	Parameters
smida18-pc	80	/spectrum/oneclick.jnlp	topology={federated_asset_id}

Userid **Password**

URL to launch in Context
http://{hostname}:{port}/{path}?{parameters}

To enable launch of the CA SPECTRUM Service Dashboard from the CMDB MDR do the following:

1. Create a new MDR (see step 1 of the OneClick launch section).
2. Provide information for the following fields:
 - **Button Name:** SLM Dashboard
 - **MDR Name:** ServiceManager
 - **MDR Class:** SPECTRUM
 - **Active:** active
 - **Owner:** CMDBAdmin
 - **Description:** Context launch for Service Manager CI.
 - **Hostname:** <your CA SPECTRUM OneClick server host>
 - **Port:** <your CA SPECTRUM OneClick server port>
 - **Path:** /spectrum/oneclick.jnlp
 - **Parameters:** mh={federated_asset_id}
 - **URL to launch in Context** (no spaces): http://{hostname}:{port}/{path}?-app=com.aprisma.spectrum.slm.SLMDashBoardConsole@slm?{parameters}
3. Click Save.
This will generate a message indicating that the MDR was created.
4. Click Close.

Here you can see an example of the Dashboard MDR provider definition:

CA CMDB

Logged in as: CMDBAdmin (Log Out) (Close Window)

File View Window Help

SLM Dashboard MDR Provider Definition Edit

Button Name	MDR Name	MDR Class	Active	Owner
SLM Dashboard	ServiceManager	SPECTRUM	Active	CMDBAdmin

Description
Context launch for Service Manager CI

Hostname	Port	Path	Parameters
smida18-pc	80	/spectrum/oneclick.jnlp	mh={federated_asset_id}

Userid Password

URL to launch in Context
http://{hostname}:{port}/{path}?-app=com.aprima.spectrum.slm.SLMDashBoardConsole@slm?{parameters}

Use GRLoader to Copy Data from Remote MDRs

An MDR is considered “remote” if it is not installed on the same system as the CA CMDB.

GRLoader can be used to copy data from a remote MDR to the CMDB in either of two ways:

- Copy the XML data from the remote system which runs the MDR to the system running CA CMDB and execute GRLoader on the CA CMDB system.
- Execute GRLoader on the remote MDR system itself.

To prepare to execute GRLoader from a remote system do the following:

1. Create \$SPECROOT\custom\grloader directory for GRLoader to reside in.
2. Copy the contents of the %NX_ROOT%\java\lib directory from the CA CMDB system to a directory on the remote system that you want to run it on. This will be called the %ROOT% system.
3. Create a file called NX.ENV in the %ROOT% directory @NX_LOG=path_which_will_contain log files
4. Create directory %ROOT%\site\cfg
5. Create directory %ROOT%\log

For the SpectroSERVER(s) a best practice is to install GRLoader in \$SPECROOT\custom\GRLoader on the SpectroSERVER. This will protect the GRLoader install during SPECTRUM upgrades or patches since the \$SPECROOT/custom directory is protected/retained during either of these processes.

Note: When GRLoader is first run it converts the contents of the <GRLOADERDIR>\NX.ENV file into the <GRLOADER>\site\cfg\GRLoader.properties file. If you ever update the NX.ENV file (for example, because you moved GRLoader to a different directory) delete or rename the <GRLOADER>\site\cfg\GRLoader.properties file and let GRLoader create the new/updated GRLoader.properties file.

To run GRLoader from the above system, execute the following command:

```
java -cp %ROOT% -jar %ROOT%/GRLoader.jar -N %ROOT% -u [userid] -s [server] -i [other GRLoader options]
```

where %ROOT% is the fully qualified path containing the files copied

CA Service Desk Issues and Incidents

By default, the integration between CA SPECTRUM 8.1 and CA Service Desk generates issues instead of incidents and there is no CI linked to the issue being raised, except within the issue description. If the integration is supposed to create incidents this can be changed in the CA Service Desk Policy Type as the integration is merely based on the SPECTRUM alarm cause or the SPECTRUM model handle. For CA SPECTRUM r9.0 and later, however, the integration has been extended. Incidents are created by default and the proper CI is linked to the incident that is raised. If a CI does not already exist, it will be created. For more information about the additional features included in CA SPECTRUM r9.0, see the product documentation.

The following example depicts are Cisco imported as a CI, along with all of its attributes entirely populated by the SPECTRUM – CMDB integration.

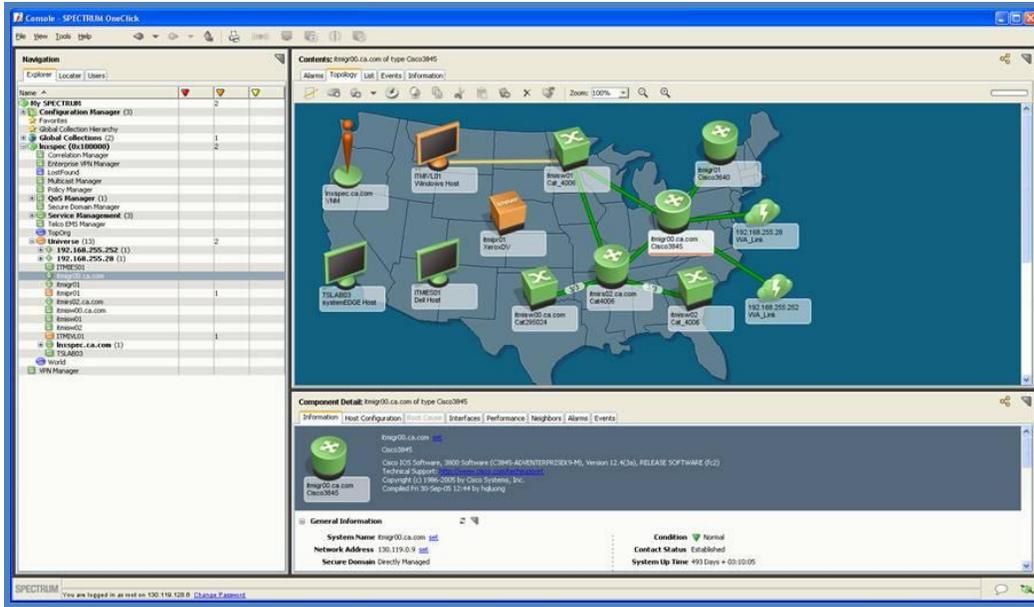
The screenshot shows the 'Configuration Item Detail' page for a Cisco Network Switch. The page includes a navigation menu with 15 numbered items: 1. Attributes, 2. CMDB Relationships, 3. Inventory, 4. Log, 5. Service, 6. Location, 7. Contacts, 8. Organizations, 9. Service Contracts, 10. Incidents, 11. Problems, 12. Requests, 13. Change Orders, 14. Issues, and 15. Impact Analyzer. The 'Attributes' section is expanded, showing a table of network-related attributes.

Name	Class	Family	Active?	
ukslsw21.ca.com	Network Switch	Network.Switch	Active	
Serial Number	Alt CI ID	Host Name	DNS Name	MAC Address
fox09280371		ukslsw21.ca.com		001562A39F7F

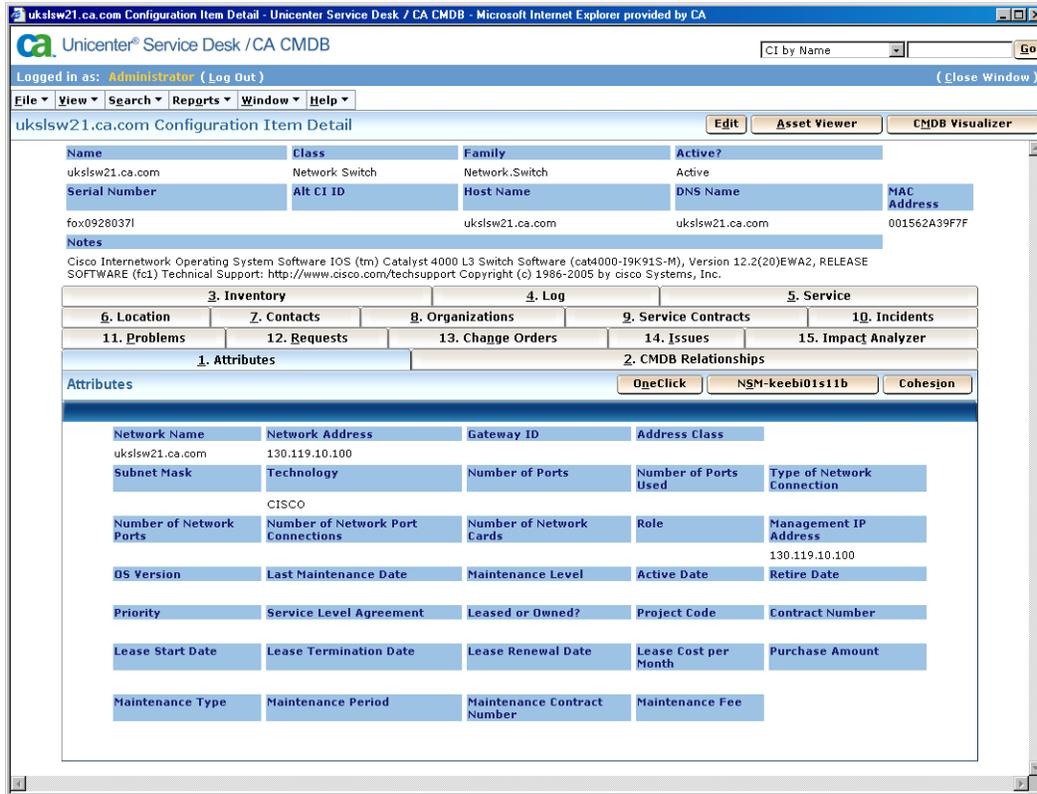
Network Name	Network Address	Gateway ID	Address Class	
	130.119.10.100			
Subnet Mask	Technology	Number of Ports	Number of Ports Used	Type of Network Connection
Number of Network Ports	Number of Network Port Connections	Number of Network Cards	Role	Management IP Address
OS Version	Last Maintenance Date	Maintenance Level	Active Date	Retire Date
Priority	Service Level Agreement	Leased or Owned?	Project Code	Contract Number
Lease Start Date	Lease Termination Date	Lease Renewal Date	Lease Cost per Month	Purchase Amount
Maintenance Type	Maintenance Period	Maintenance Contract Number	Maintenance Fee	

Clicking on the OneClick button launches the next screen into SPECTRUM OneClick in context of this specific CI.





In the next example you can see how the same Cisco switch CI was also discovered by CA Cohesion and CA NSM. As a result there are three MDR launch buttons – one for CA NSM, one for CA Cohesion and one for CA SPECTRUM – along with enriched attribute information:



For additional information on this integration, consult the CA SPECTRUM and CA Service Desk Integration Guide (document 5178). You are also strongly encouraged to engage CA Service when implementing this integration.

Chapter 6: Integrating with CA NSM

This chapter discusses how CA CMDB and CA NSM can be configured to work together. The following key topics are presented:

- Overview and value of the integration
- CA NSM and CA CMDB Integration
- CA NSM and Change Impact Analyzer
- Reference Documentation

Overview and Value of the Integration

The integration between CA NSM and CA CMDB provides the following benefits:

- Search for CA NSM discovered assets from CA CMDB
- Synchronize CI status information from CA CMDB to CA NSM
- Export CI data and relationships to CA NSM WorldView from CA CMDB
- Populate the CA CMDB with discovered IT infrastructure data and hierarchical relationships from CA NSM
- Enable impact and root cause analysis of CI states in CA NSM WorldView
- Monitor event console messages from CA NSM and post announcements to the CA CMDB scoreboard

In a best practice scenario, CA CMDB and CA NSM would be integrated using the steps documented in the *Systems Management Integrations Green Book*. This document details the integration between CA CMDB and CA NSM through CA SPECTRUM and is available through the CA Support <http://support.ca.com> website.

This chapter also includes information about using the Change Impact Analyzer (CIA) component. This optional component incorporates features from both products to visually demonstrate the impact of changes throughout the environment through CA NSM WorldView map.

Further benefits are available in conjunction with additional product integrations. For example, when CA Service Desk is also installed, the integration provides automatic request and incident creation. The use of CA Cohesion ACM is recommended as the best practice approach for automating the discovery of servers and software and their relationships. This integration leverages the CORA API to ensure that the CIs discovered by CA Cohesion are reconciled and mapped with the appropriate relationships.

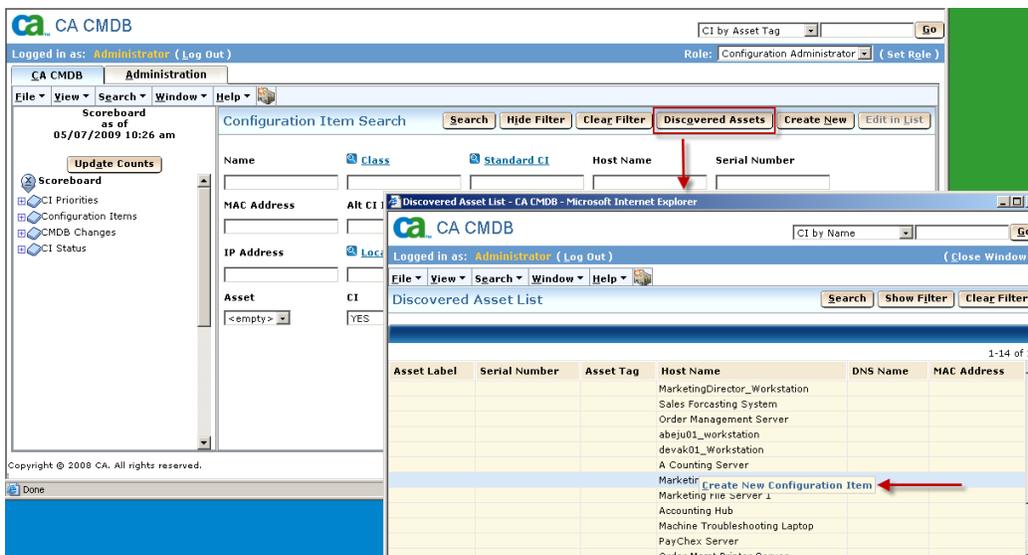


Even without the additional integration with CA Service Desk or CA Cohesion ACM the integration directly between CA CMDB and CA NSM can still add value. This chapter explains how the direct integration works, when to use it, and how it can be configured.

CA NSM and CA CMDB Integration

When CA CMDB and CA NSM share the same MDB, a search can be done from CA CMDB for Discovered Assets in NSM. Assets that have been discovered in CA NSM can be imported into the CA CMDB on an individual, as needed basis.

From any CI search list, click the Discovered Assets button to pull up a list of discovered assets in CA NSM. Right click on the asset to pull up the “Create New Configuration Item” link. Selecting this will launch the Create New Configuration Item window in CA CMDB.



You will need to enter a class for the new CI and any additional attribute information available. By default, the “Asset?” flag is set to No and the “CI?” flag is set to Yes.

CA NSM and Change Impact Analyzer

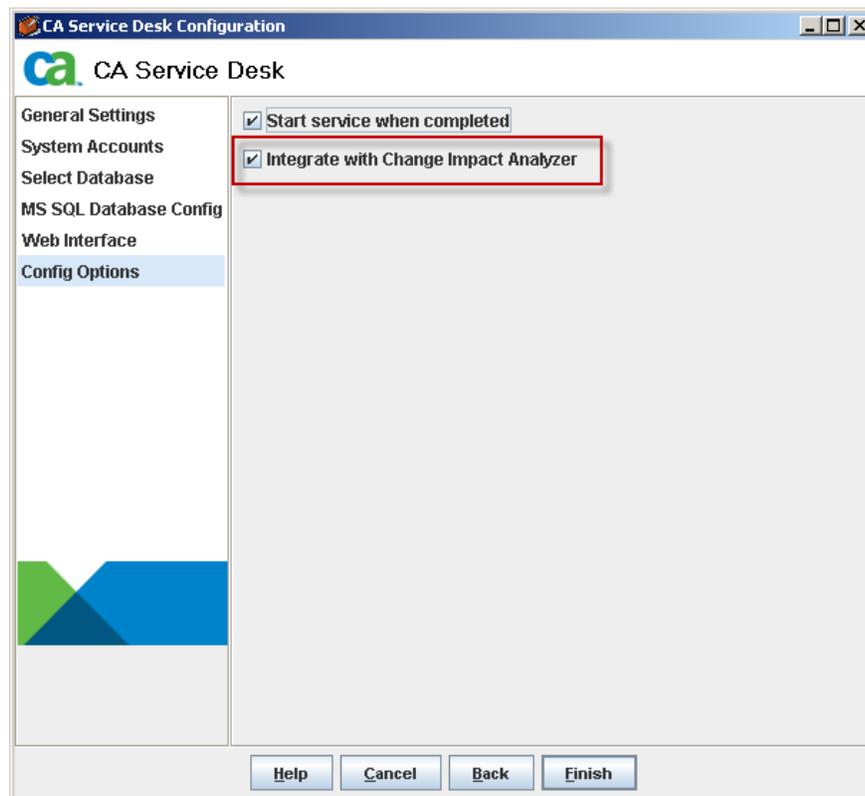
The Change Impact Analyzer (CIA) component, an optional feature of the CA CMDB, utilizes features from CA CMDB and CA NSM to enable you to import discovered data and relationships existing in CA NSM into the CA CMDB, and introduce changes to the CA NSM WorldView map and to see the impact that changes will have on your organization.

Note that the discovery of CA CMDB relationship classification is not leveraged by the CIA, but the containment hierarchy and the basic configuration information of the CIs can be imported.

Steps to Configure Change Impact Analyzer

The discovered asset search described above is a feature available by default without and does not require any additional configuring. In order to synchronize asset status, create CA CMDB announcements and import bulk asset data from CA NSM, however, the CIA feature must be configured.

If CA CMDB and CA NSM are installed on the same server, the option for integrating with Change Impact Analyzer is selected by default during the initial installation and configuration of CA CMDB. This option must have been selected before moving forward with configuring CIA. If it was not previously selected you must re-run the CA CMDB configuration and select that option.



If CA NSM and CA CMDB are installed on different servers, a CA CMDB Secondary server must be installed on the CA NSM server so that this option can be selected. Refer to the instructions in the CA CMDB Implementation Guide for integrating CA NSM and CA CMDB.

After the above option has been selected, CIA must be configured so that the additional integration features can be used. To finish configuring CIA, the `pdm_uspsnm_nxd` daemon must be running. This is done by modifying the `pdm_start` up file as noted in the following steps.

Modify pdm_startup

To configure the pdm_startup file, do the following:

1. Open a DOS Shell (Command Prompt).
2. In the DOS Shell enter “nxcd” and press return.

This will always bring you to the base directory of the CA CMDB installation regardless of operating system.

3. Change Directory to samples\pdmconf
4. Execute the following:

```
pdm_perl pdm_edit.pl
```

5. In response to the prompt asking if you are editing a pdm_startup for NT, enter “Y” and press Enter
6. Instructions for modifying your pdm_startup are displayed, and you are then asked to enter Y to continue. Enter “Y” and press Enter
7. Select 9 from the list of options and hit Enter
8. Select A to Add host, and then hit Enter
9. You are now prompted to enter a hostname. The default is [primary]. Hit Enter to take the default.

Note: If CMDB and NSM are on different machines, enter the hostname of secondary server. CIA or pdm_uspnsn_nxd daemon resides on the secondary server which is in NSM Machine.

10. You should see the following line afterwards:

```
Change Impact Analyzer Host: primary
```

11. Press the Enter Key to return to the menu.
12. Press X to Save and quit the script, and to have your pdm_startup.rmt file created in the directory
13. Back up the current pdm_startup.tpl file and replace it with the pdm_startup.rmt file you just created. To do this:
 - a. Rename the existing \$NX_ROOT/pdmconf/pdm_statup.tpl file to pdm_startup.tpl.orig
 - b. Copy \$NX_ROOT/samples/pdmconf/pdm_startup.rmt to \$NX_ROOT/pdmconf and rename it to pdm_startup.tpl
14. Run the configuration for CA CMDB by executing pdm_configure from the DOS Shell, setting all passwords to the values used in the installation, and insuring that the “Load Default Data” is NOT Selected.
15. Verify that the Change Impact daemon has started by running “pdm_status” from a command prompt. You should see an entry for Change Impact (pdm_uspnsn_nxd).

Create a New CIA Repository

To create a new CIA repository, do the following:

1. Start the CA CMDB Client and sign-in using an ID with CMDB Administrator access
2. Create a new CIA Repository pointing to CA NSM. To do this:
 - For a **standalone CA CMDB installation**, click on the Administration Tab, select Change Impact Analyzer, and select Repository Configuration
 - For a **combined CA Service Desk/CA CMDB installation**, click on the Administration Tab, select Administration, Service Desk, Application Data, configuration Items (Assets), Change Impact Analyzer, and select Maintain Repository Configuration.
3. Click Create New to add a new Repository record for CIA
4. Fill in the following fields:
 - **Symbol:** The unique identifier for this repository. You should assign a symbol that makes the repository easily recognizable from the list.
 - **Status:** Indicates whether the repository is active or inactive.
 - **Description:** A detailed description of the repository. You can use this field to describe the purpose of the repository or the relationships assigned to the repository.
 - **Repository:** The name of the corresponding NSM WorldView repository. This name must be the same as an existing NSM virtual repository.
5. Click Save

The repository is created. The Configuration Items tab lists the configuration items that reference this repository in their Assigned Repositories list. The Relationships tab lists the Change Impact Analyzer relationships that reference this repository on their Assigned Repositories list.

6. Select the NSM Credentials tab to set the login credentials used by the Change Impact Analyzer daemon when connecting to this repository. To do this:
 - a. Enter the User ID. This should be "sa" or "nsmadmin"
 - b. Enter the Password and enter it again in the Verify Password field.
 - c. Click Change Repository Login to store the information in the database.
 - d. Click Test Repository Login to verify the login information signs in to the NSM WorldView repository successfully.

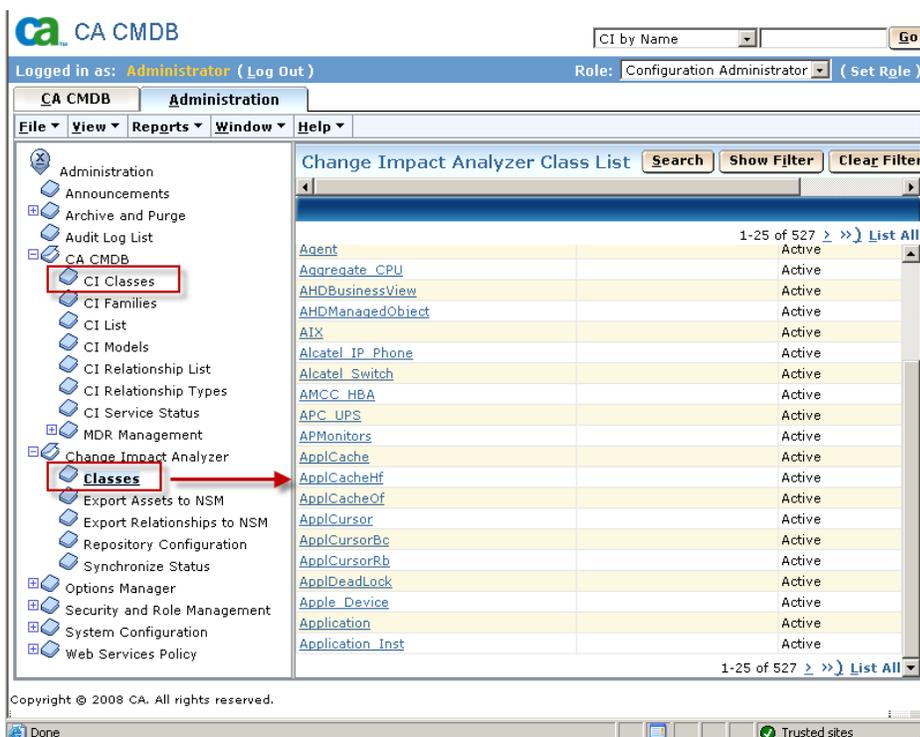
CIA Classes

When CIA is configured, two different lists of classes are seen in the CA CMDB Administration tab.



- **Configuration item classes** are general categories of hardware, software, and services; for example, workstation, printer, and services. These configuration item classes can then be grouped into broader categories called configuration item families.
- **Change Impact Analyzer classes** represent the WorldView Managed Object classes found in each participating WorldView Repository. There should be an entry in this table for every WorldView class. To implement the mapping strategy, each Configuration Item Class is mapped to a Change Impact Analyzer Class. By using a command line utility during initial configuration, the table of Change Impact Analyzer Classes is built automatically from all the WorldView Classes contained in the WorldView Repositories. Another command line utility maps Asset Classes to Change Impact Analyzer Classes.

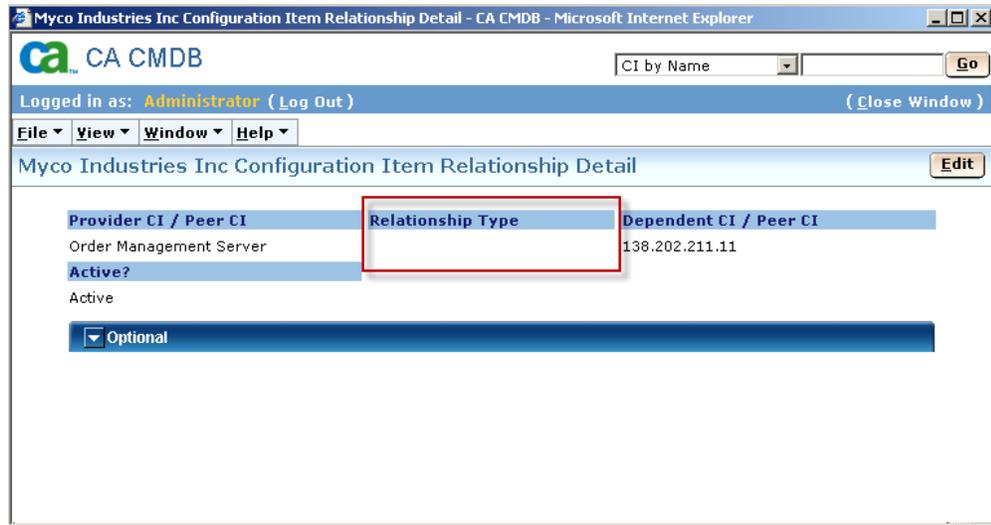
When a new CI is created from CA NSM, the CI Class is used for classification.



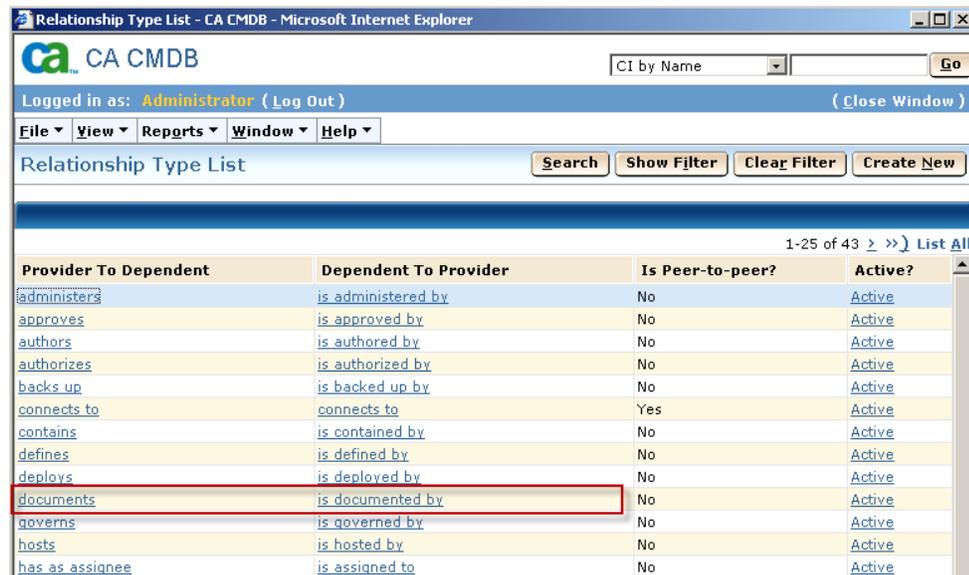
CIA Relationships

By default, CA CMDB provides 280 relationship types which are used to define and manage relationships between CIs. When CA NSM assets and relationships are imported into the CA CMDB, the CA CMDB relationship types are not used. The imported relationships are hierarchical only.

The following screen shot shows a relationship that was imported from CA NSM, but no Relationship Type is available.



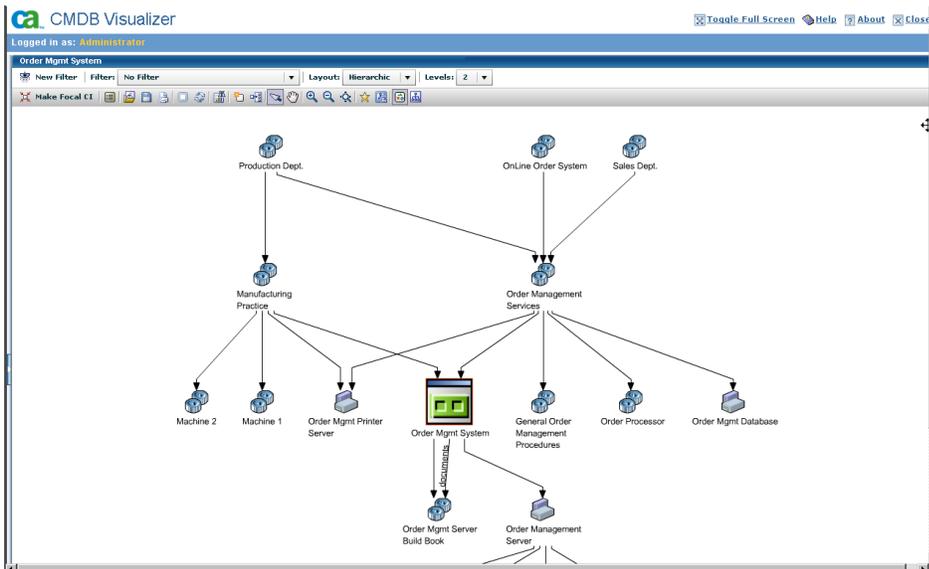
The Relationship Type field can be manually updated by clicking the Edit button or by using the “Edit in List” feature on the CI search screen, but the relationship direction will be backwards. For example, look at the following Order Mgmt System relationship with the Order Mgmt Server Build Book. The best available Relationship Type is documents/is documented by.



When that relationship type is selected, it is obvious that the order is backwards and that it should read Order Mgmt System “is documented by” Order Mgmt Server Build Book.



This is also seen when using the Visualizer. The line between CIs, which normally displays the relationship type, is missing unless the relationship was manually added as in the previous example.



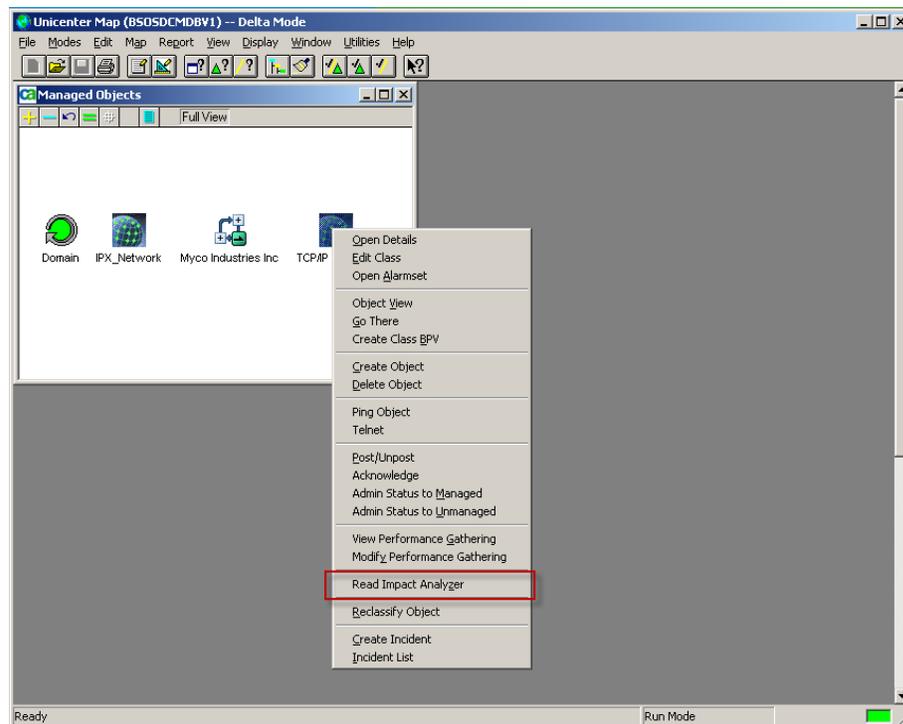
Importing Data from CA NSM

To import an initial load of asset data from CA NSM into the CA CMDB follow the steps below. Note that maintaining the data moving forward is the responsibility of Change and Configuration Management. Strict change control should prohibit the random updating or importing of additional assets without going through the change review process.

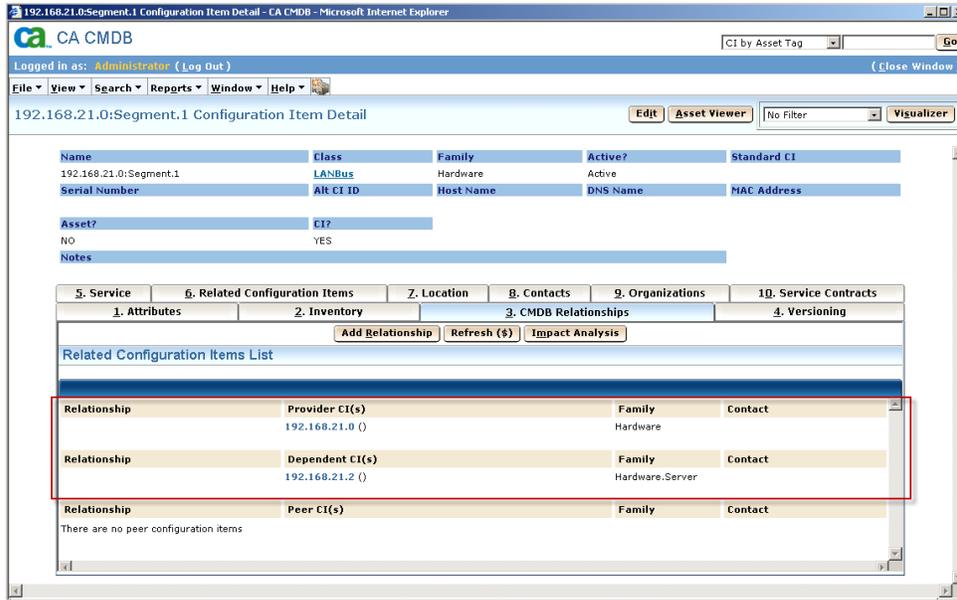
To import data from CA NSM to CA CMDB:

1. Confirm that the CA CMDB services are running.
2. Open the CA NSM 2D-Map.
3. Double-click on the TCP Network.

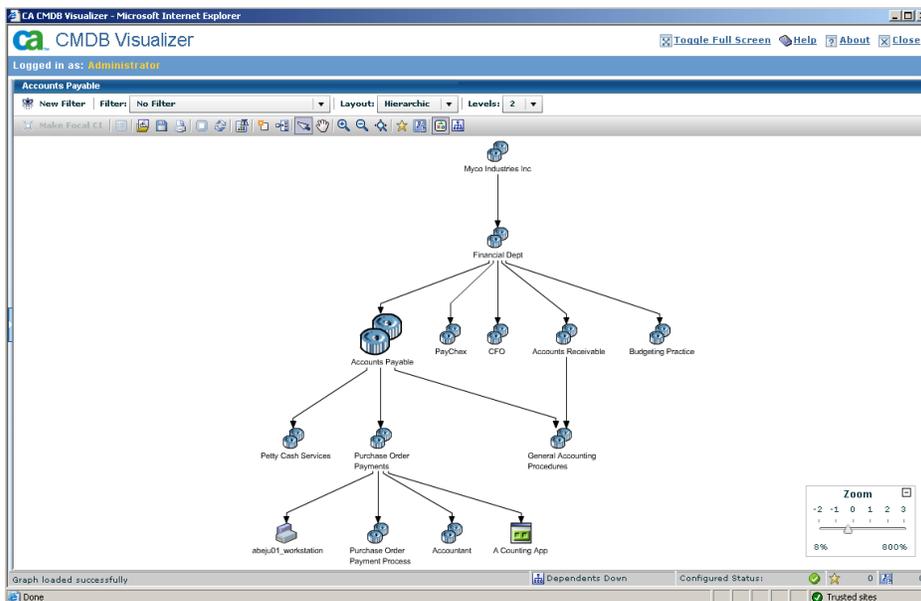
Right mouse click on one of the networks discovered during the CA NSM Installation and Configuration phase and select Read Impact Analyzer.



4. Select the NSM MDB, and click OK.
5. The discovered CIs and their physical relationships will be populated into the CA CMDB. Confirm this by opening a CA CMDB Web Client, and review the list of CIs.
6. Open one of the CIs and note that some of its information is populated.
7. Note that the CMDB Relationships tab has a provider CI and a dependent CI, but no Relationship types exist.



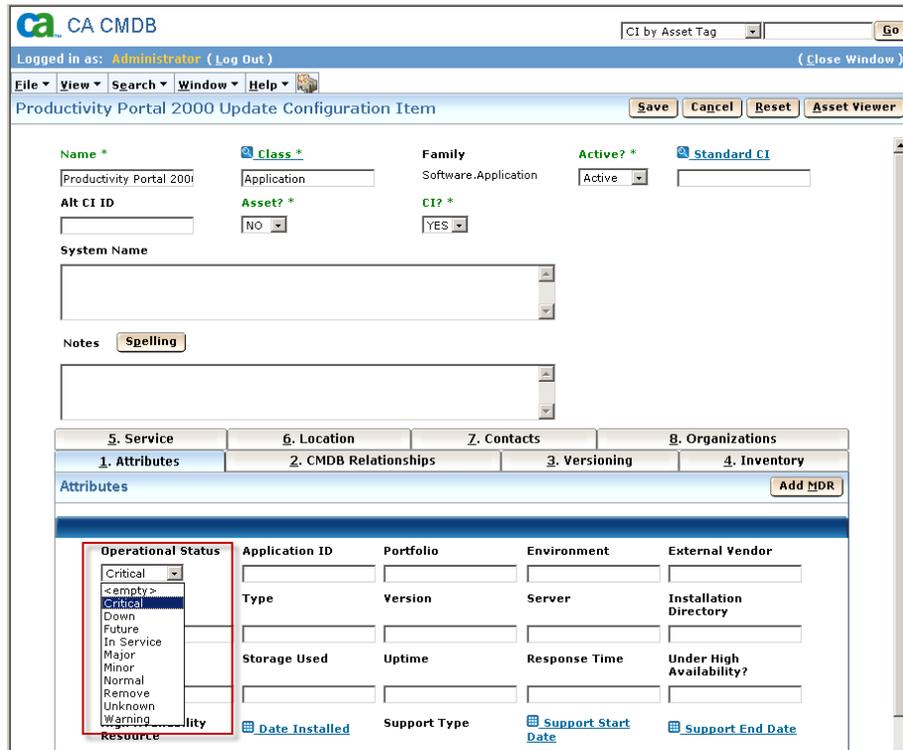
8. Walk the relationships by clicking the Visualizer button on the CI detail form.



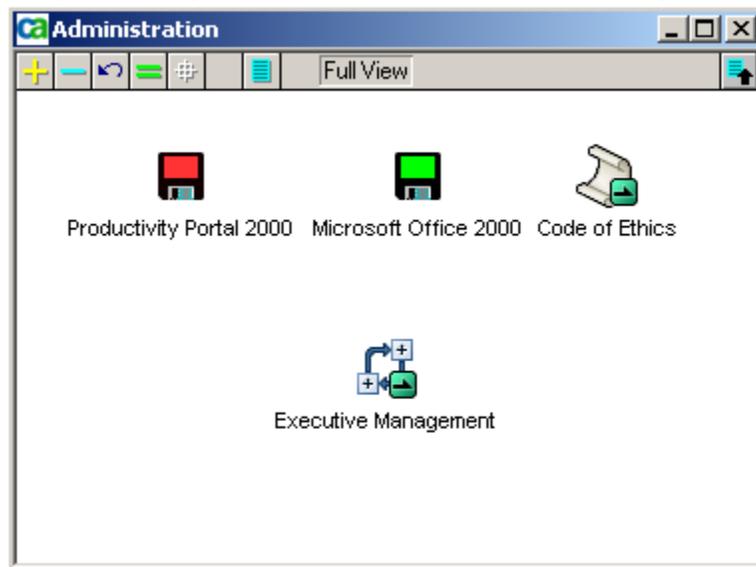
Synchronizing CA NSM and CA CMDB CI Status

The operational status of a CI in CA CMDB can be set and then synchronized with the object's status in CA NSM. To do this:

1. View a selected CI in CA NSM Worldview and verify that its icon status is green
2. Search for that CI in the CA CMDB and change its operational status to Critical



3. On the CA CMDB Administration tab, click on Change Impact Analyzer and then Synchronize Status
4. Select the Synchronize Status button
5. Verify that the object's color in CA NSM WorldView has changed to red



Reference Documentation

See Chapter 12: "Integrating with Other Products" in the *CA CMDB Implementation Guide*. for more information on integration with CA NSM and for instructions on how to define services in the CA CMDB and import them to CA NSM,

For information on implementing Change Impact Analyzer, refer to Chapter 12: "Integrating with Other Products" in the *CA CMDB Implementation Guide* as well as Chapter 4 "Using the CA CMDB" in the *Incident and Problem Management Green Book*.

For information on best practices for deployment of CA NSM r11.2 using System Management Pack 2.0, CA SPECTRUM, CA Service Desk Manager, and other CA products, refer to the *Systems Management Integrations (2009 Edition)* Green Book available on CA Support's web site.

<https://www.ca.com/greenbooks>

The CA Systems Management Pack (formerly "Service Availability Management Pack" or "SAM Pack") uses CA NSM BPVs to define collections of IT components and functions that, together, comprise an IT Service. WorldView and Event Management policies are then used to assess the business impact of events generated by those IT Service components and issue the appropriate notification and response.

Chapter 7: Other Integrations

CA CMDB integrates with many other solutions – both CA and non-CA – allowing you to import and reconcile collected attribute and relationship details for the CIs that support your critical IT business services. By providing a broader view of your IT resources and a clearer understanding of how they support your business, CA CMDB helps you to enforce compliance standards, implement the necessary change controls and perform effective root cause analysis to maintain SLAs, increase client satisfaction and reduce downtime for repairs.

Some of these integrations occur automatically and with minimal additional integration effort. CA IT Client Manager and CA IT Asset Manager, for example, can integrate with CA CMDB through a shared MDB. Others require additional configuration. For example, CA Wily Customer Experience Manager (CA Wily CEM) provides an integration package that enables you to import CA Wily CEM configuration components (such as business processes and business transactions) into the CA CMDB to help report on SLAs and help quantify the business impact of performance on the IT infrastructure.

This chapter provides additional insights on integrating CA CMDB with the following:

- CA Service Catalog
- CA Mainframe Solutions

For additional information on integrating with CA IT Asset Manager, consult Chapter 8: “Managing Configuration Items” in the *CA CMDB Administration Guide*. An overview of these enhancements is provided in the “CA APM and CA CMDB Enhanced Integration Overview” which is available from the CA APM product home page on <http://support.ca.com>.

For additional information on integrating with CA IT Client Manager, consult Chapter 9: “Data Population” in the *CA CMDB Implementation Guide* and Chapter 3: “Business Structure” in the *CA CMDB Administration Guide*.

Information on how to install and use the integration pack can be found in the *CA Wily Introscope Integration Pack for CA CMDB Implementation Guide* which is available, along with the Integration pack, from the support.wilytech.com support site.

Integrating with CA Service Catalog

CA Service Catalog manages the business of IT as a service provider, from initial service request through fulfillment. It simplifies requests for your IT services and resources by providing a single point of contact from which to browse available service offerings, request new services and review the status and history of existing requests. CA Service Catalog enables you to track specific fulfillment times for services rendered as well as to designate thresholds for taking



actions based on workflow. CA Service Catalog provides a central library of service offerings designed to simplify users' access to the complete portfolio of IT service offerings. A full featured service builder allows the specification of service offerings including associated cost and service levels.

The CMDB provides a logical image of the IT environment and the services delivered. The components that are required in order to deliver IT services and the services themselves are managed as Configuration Items (CIs) with their status, their configuration parameters and – most importantly - with the relationships between each other and the services delivered.

The CMDB has never had a concept of being a monolithic, all knowing repository of everything that can be of potential interest for IT management. Configuration Management best practices suggest focusing on critical services, the related Configuration Items and on the information that is relevant for delivering quality services to the business.

ITIL V3 takes these concepts further by introducing the Configuration Management System (CMS). In the Configuration Management System the Integrated CMDB is linked with other sources of configuration information , such as multiple physical (special purpose) CMDBs, Asset Management Systems, media libraries, various systems management tools and even Enterprise Applications like Human Resource Management, Supply Chain Management and many other sources of configuration data.

In the CA architecture, the CMS will leverage a service oriented architecture and federation techniques in order to access configuration information maintained in MDRs and present them in a unified and consistent manner. The data are structured and integrated according to CA's Unified Service Model (USM) with IT and Business Services being the pivotal elements of this data model. The USM is a service-centric information model at the heart of CA's EITM architecture. It unites information from diverse domain managers to create a 360 degree view of a service – including the technology, assets, people, projects and processes supporting any given service and the relationships among these components. The USM does this by leveraging the service definition that is maintained within the CMDB and the rich data sources housed in CA's Capability Solutions, such as Service Catalog. The USM consists of:

- Service definitions. The definition of a service describes the Configuration Items (CI's) that support a given service, the inter-relationships between CIs supporting the service and key CI and service attributes.
- Federation APIs. Definitions of how CA Capability Solutions and third party solutions federate instantiated service data with the CMDB through web services.

- **Key Indicators.** The ability to define additional key indicators, extending the shared data model to more effectively measure the performance of any specific service. The types of data that can aggregate to a service include infrastructure availability and performance data, end user application response time data and capital and operating expenses. Examples of the types of service-level key indicators supported include SLA/OLA compliance and year-to-date budget-to-actuals

CA Service Catalog is an important contributor to the USM and a key building block of the Configuration Management Systems with CA CMDB as core element.

CA Service Catalog is integrated with CA CMDB as an MDR which contributes business and IT services CIs and their relationships to the CMDB. These CIs can then be further related to the supporting infrastructure, applications, and underpinning services and can be enhanced through additional configuration data supplied by other MDRs or through the change and configuration management processes.

This configuration data enables Incident, Problem and Change Management to identify the root cause of service degradations or to assess the impact of incidents and changes on the subscribing lines of business. Service related configuration information will benefit all service management processes across the entire service life cycle, for example:

- Service Design and Capacity Management
- Security Assessments
- Financial management

Integration Points and Functionality

The key to this integration is identifying which information and which relationships are relevant for IT Service Management. The Configuration Management processes— primarily Configuration Identification and Configuration Control - must provide guidance as to how to populate the CMDB with service CIs and relationships.

Integration Points from CA Service Catalog

CA Service Catalog Service offerings can be associated with existing CIs in the CMDB. Although most organizations prefer to utilize configuration management practices to regulate creation of CIs, if a CI for a service offering does not yet exist, it can also be created by CA Service Catalog. A default status can be configured as “active” or “inactive” depending on how such CIs should be further managed. Associations can also be removed if they are no longer relevant. Integration with the CMDB Visualizer provides CA Service Catalog with a graphical analysis of a CI’s dependencies, including details provided by other MDRs integrated with the CA CMDB, supporting both root cause analysis and change impact analysis. CA Service Catalog also gains access to the CMDB integration report for reviewing service associations.



Integration points from CA CMDB

Since CA Service Catalog is integrated into the CMDB as an MDR, this enables you to launch the Service Catalog user interface from CA CMDB, providing access to the Service Catalog Impact Analysis page. The Impact Analysis page provides subscription information for a CI and its associated service offerings.

Integration Value

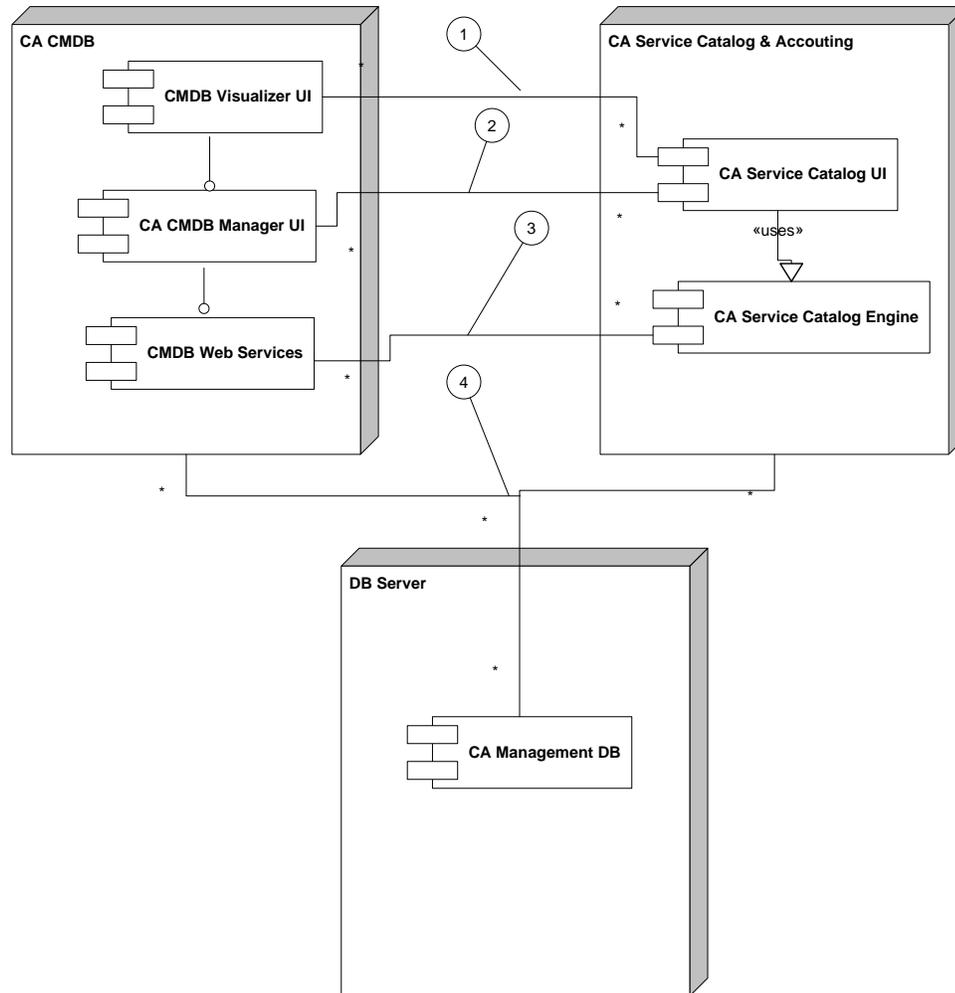
CA CMDB and CA Service Catalog support the concepts of ITIL V3 Service Asset and Configuration Management with the Configuration Management System at the heart of the Service Knowledge Management System.

All Service Management processes across the entire service management life cycle (Strategy → Design → Transition → Operation) will benefit from configuration information that has been federated from CA Service Catalog to the CMDB. CA's Unified Service Model integrates this information with configuration data from other contributors and presents it in a unified way to all consumers.

Integrating CA Service Catalog as an MDR reduces the amount of data to be replicated. At the same time this approach reduces the complexity of the CMDB data model and simplifies the configuration management tasks. The example provided in the Business Challenge section on page 122 demonstrates how all service management functions can benefit from this approach.

How the Integration Works

The following UML diagram outlines the CA Service Catalog – CA CMDB Integration.



Here you can see how:

1. CA CMDB Visualizer can be launched from the CA Service Catalog in order to show the service model and dependencies. The CA CMDB Manager UI can be called from the Visualizer in order to view further configuration details.
2. The CA Service Catalog UI can be launched directly from the CA CMDB Manager UI (or through the Visualizer's context menu) in order to perform Service Catalog Impact Analysis.
3. CA Service Catalog service offerings can be associated with CIs in the CA CMDB. If a corresponding CI does not already exist it can be created from CA Service Catalog.
4. Both CA CMDB and CA Service Catalog share the same MDB.

Full details on how to install the integration can be found in Chapter 6: "Integrating with CA CMDB" in the *CA Service Accounting and CA Service Catalog Integration Guide (r12)*.

Business Challenge

If Incident Management detects a degradation or imminent outage of a service it wants to understand which business units and, in some cases, even which individual users are affected in order to appropriately prioritize the incident and notify those users. Change Management is also interested in this type of information in order to do more precise risk assessments. Other Service Management processes, such as Capacity Management or Security Management, will benefit as well from having the relationship between services and subscribing users in the CMDB.

The Configuration Manager understands the business value but is concerned about creating hundreds - or possibly even thousands - of relationships for a single service CI. Although the CA CMDB would technically be able to handle large numbers of relationships, it would be challenging for Configuration Management to ensure that this data is correct and up to date at all times. The information overload would also make the CMDB less user friendly for the analysts.

CA Approach

CA Service Catalog automatically maintains a record of all subscriptions to service offerings. The best practice recommendation is not to replicate hundreds and thousands of these subscriptions as relationships in the CMDB. Instead, CA proposes integrating CA Service Catalog as an MDR. With that integration all relevant service offerings will be represented as CIs in the CMDB while subscription details will continue to be kept in CA Service Catalog only.

The CA Service Catalog UI can be launched, in context, from the CA CMDB for a service CI providing quick access to the Service Catalog Impact Analysis tool in CA Service Catalog. This tool provides a dashboard style view of subscriptions aligned by business unit. It also allows you to drill down to view a list of individual users subscribed to the designated service offering.

When prioritizing an incident the Incident Management process analyst can easily determine which business units are affected by selecting the service CI in the incident ticket. Due to the seamless integration with CA CMDB the analyst has immediate access to the configuration data.

In the following example, you can see how the CA Service Catalog UI can be accessed as an MDR through the Configuration Item Detail – Attributes tab.

CA CMDB

Logged in as: CMDBAdmin (Log Out) (Close Window)

File View Search Reports Window Help

Exchange Service Configuration Item Detail Edit Asset Viewer No Filter Visualizer

Name	Class	Family	Active?	Standard CI
Exchange Service	Business Service	Enterprise Service	Active	

Alt CI ID: 10145

System Name

Notes

5. Service 6. Location 7. Contacts 8. Organizations

1. Attributes 2. CMDB Relationships 3. Versioning 4. Inventory

Attributes Service Catalog

Operational Status

Portfolio Service Category Service Version Service Manager

Next, you can see how the Service Catalog UI is launched in context for the selected service CI and the CI Associated Service Details are displayed.

From the summary view the analyst can drill down to see a list of accounts or business units that are affected, and even further to view a list of individual subscribers.

CA Service Management Profile Help

Welcome: Administrator, Service Delivery (Log Out) Business Unit: ca.com Role: Service Delivery Administrator

Request Details

Impact Analysis Home > Request Count

CI Associated Request Details 1 - 4 of 4

ID	Name	Requested By	Requested For	Priority	Date Created	Date Modified	Status
10028	Create Email Account	spadmin	Sarah	3	9/17/2008 19:50:45	9/17/2008 19:50:59	Submitted
10025	Create Email Account	spadmin	spadmin	3	9/17/2008 19:45:8	9/17/2008 19:45:16	Submitted
10026	Create Email Account	spadmin	Joe	3	9/17/2008 19:49:31	9/17/2008 19:49:54	Submitted
10027	Create Email Account	spadmin	Jim	3	9/17/2008 19:50:13	9/17/2008 19:50:26	Submitted

Copyright © 2008 CA. All rights reserved. About

Reference Documentation

For more information on integration between CA CMDB and CA Service Catalog consult the following resources:

<https://support.ca.com/cadocs/j0/j027581e.pdf>



Integrating with CA Mainframe Solutions

With the introduction of the CA CMDB Connector for z/OS, an extension off of CA CMDB, you can now automatically discover components of your mainframe environment and visually understand how changes to this environment can impact business critical applications. Configuration items (CIs) and associated relationships can be depicted in a manner that is easy to understand. Standard federated drill down capabilities provide a more detailed view of specific mainframe CIs. The CA CMDB Connector for z/OS captures the important mainframe elements ranging from LPARs, address spaces, jobs, CA IDMS, CA Datacom, DB2, CICS, MQSeries, and hardware (both tape and disk).

Many organizations are sometimes faced with the daunting challenge of making sense of their complete environment. Difficulties or inaccuracies in the mapping of current mainframe resources can lead to outages or poor application performance. The output of the CA CMDB Connector for z/OS is consumed by CA CMDB affording the capabilities of a comprehensive and accurate view of the resources consumed.

The CA CMDB Connector for z/OS provides the ITIL structure and management for Change, Incident/Problem, and other IT Service Management disciplines by enabling a clear view of mainframe CIs and their interrelationships with mainframe and distributed resources.

Integration Points and Functionality

The CMDB Connector for z/OS provides an intuitive web-based interface (as well as 3270, if needed) that is designed to discover (or rediscover) mainframe CIs and associated relationships for inclusion in your IT Services. For example, you can include:

- Processor Complex, LPAR and Operating System level
- Address spaces and jobs
- CA-IDMS™ Subsystems
- CA-Datacom® Subsystems
- IBM DB2 Subsystems
- IBM IMS Subsystems
- IBM CICS Regions
- IBM MQSeries Subsystems
- Hardware (tapes and DASD)

The benefits of this integration include the following:

- Regular discovery of mainframe resources enables you to audit and reconcile changes to the configuration of your mainframe environment.

- Use of common tools to populate CA CMDB with both mainframe and distributed resources simplifies training requirements and administrative overhead.
- Ability to view mainframe resources through CMDB Visualizer provides a better understanding of the relationships and dependencies across mainframe and distributed components.
- Launch button that provides federated mainframe CI data directly from mainframe.
- Ability to pinpoint IT Service CI relationships as well as “what has changed” supports service desk functions in determining root cause analysis and reducing mean time to repair.
- Ability to identify which CIs can be impacted by either planned changes or unexpected outages affecting related CIs can help you be proactive in scheduling updates or redirecting resources in order to minimize extensive downtime.

Business Challenges

Here you see the many business challenges that this integration helps meet:

- **Changes to hardware\software can impact the health of business critical applications**
By providing discovery of detailed resource information and their respective relationships, this integration enables you to visualize the resources and understand the changes that can impact the continuous available of these resources.
- **Administrators are updating system resources and devices daily**
By tracking system resource changes through CA Service Desk Manager (which includes CA CMDB), this integration keeps your support team better informed and can help them understand which changes have impacted services. Support personnel can then concentrate their efforts on solving the problem rather than having to first discover the configuration.
- **Difficult to gauge impact to SLAs**
This integration provides the ability to drill down into system resources and device information to hasten problem analysis as well as to provide a proactive means for pinpointing future problems.
- **Manual tracking of changes to system configurations can result in inaccurate, out-of-date information and is ineffective as a means of curtailing problems**
Discovery/re-discovery services through the CA CMDB Connector for z/OS can be regularly scheduled. Automating rediscovery and providing visual accessibility into detailed systems resources enables you to view the current CI and the relationships within the CMDB.
- **Inconsistencies with internal application programming**
This integration can utilize the compare function across identical LPARs. Changes implemented on one LPAR can be checked with a compare facility against other like LPARs to ensure changes have been rolled out across all of the required systems.



■ **Changes to systems resources across heterogeneous environments**

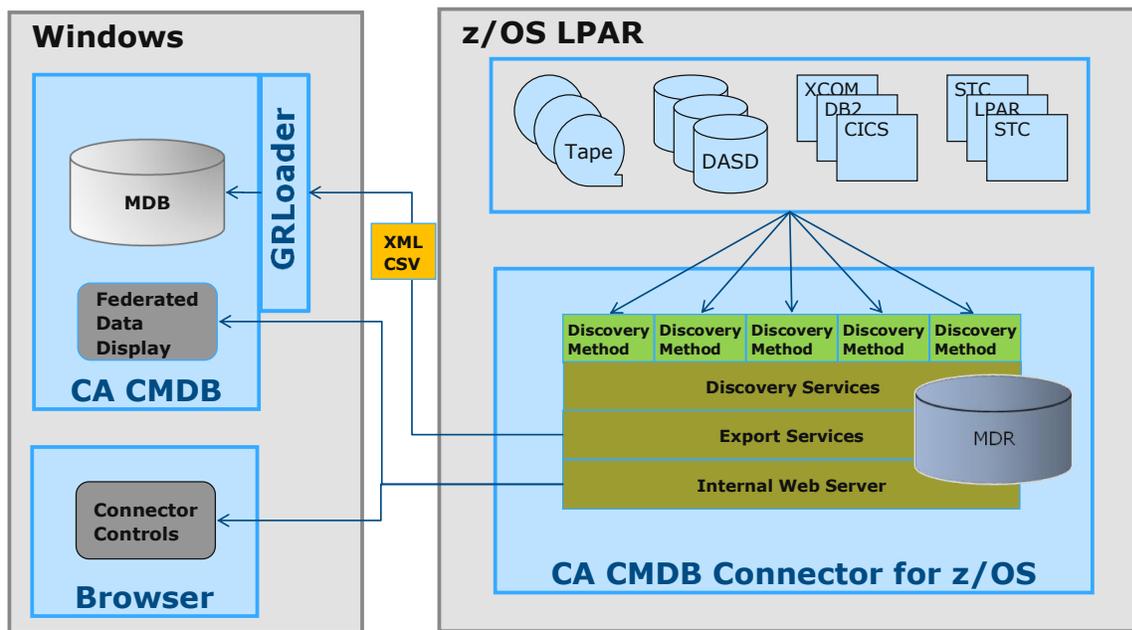
This integration is able to obtain CI information and analyze changes to system resources at the enterprise level – extending to the mainframe and distributed platforms. By saving critical infrastructure information, resource and relationship information in a centralized CMDB repository, this hastens problem analysis and identification of problem resources.

■ **Increase understanding of current resource configuration**

The CMDB is able to visually represent the resource’s environment and relationships. This enables you to proactively manage changes to the environment and clearly articulate intelligent decisions based on current resource knowledge.

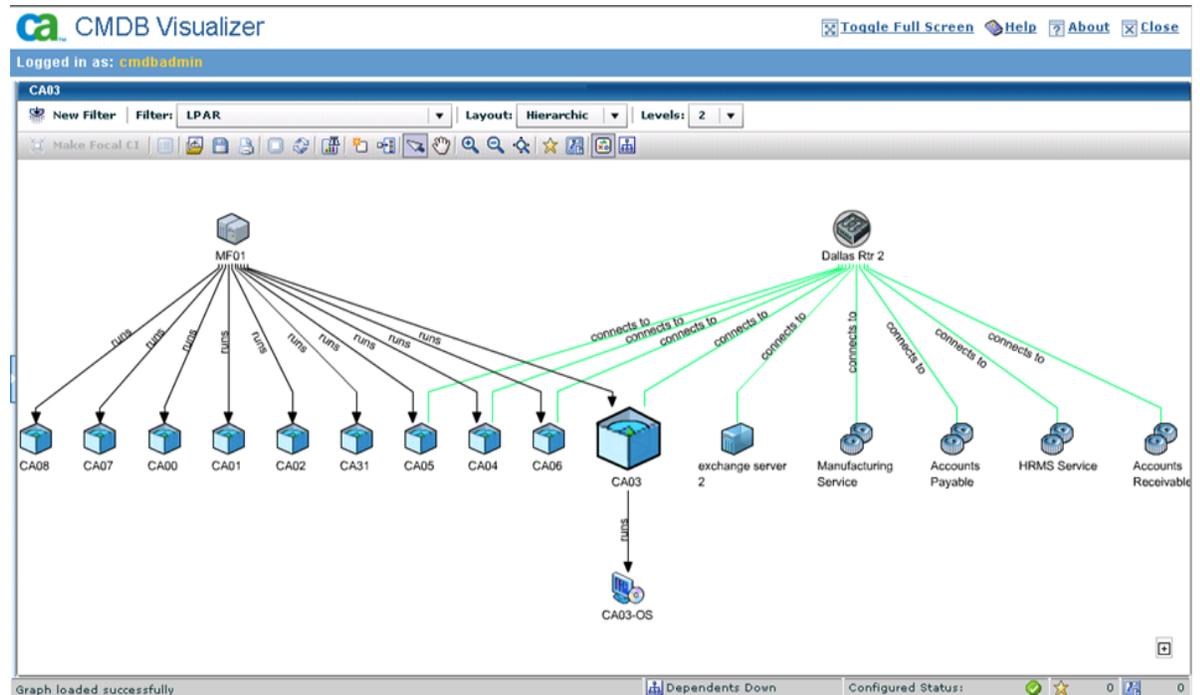
How the Integration Works

The following graphic depicts a basic overview of how the CA CMDB-Mainframe integration works:



CA CMDB Connector for z/OS is an application program that runs as an MVS started task within a given z/OS system LPAR (Logical Partition). The application acts as an MDR, providing XML structured data to CA CMDB which identify CIs and the associated relationships within the z/OS system. The Connector also acts as a federated data server, storing and presenting additional mainframe information in response to a push-button request from CA CMDB. The Connector includes a set of prebuilt methods to perform discovery of specific z/OS components. Discovery requires the setup of appropriate interfaces - for example, CA-IDMS, Customer Information Control System (CICS), DB2 and WebSphere MQ. The Connector needs to be installed on each LPAR where mainframe CI discovery is required.

Discovered CIs and relationships are exported to XML files and can then be imported into the CA CMDB repository. From there you can use CA CMDB to visualize mainframe and distributed CIs. For example, here you can see an integrated view:



The product has the following central components:

- **Discovery Services and Methods**

Discovers CIs on the z/OS system based on the discovery methods used. A discovery method is a REXX procedure that discovers z/OS components, builds the CIs and relationships for those components, and stores the CIs and relationships in the MDR. In addition to several prebuilt methods (for example, a method to discover address spaces and started tasks) you can work with CA Services to create custom methods as needed.

- **Data Exporter**

Exports discovered CIs from the MDR to a file in XML format where it can be loaded into the CA CMDB repository using the GRLoader utility. The Exporter can also format the data in comma-separated value (CSV) format, which enables you to view the data using, for example, a spreadsheet application. With a CSV file, you can use ADT to filter and manipulate the data before you import the data.

Keep in mind that not all attributes of a discovered CI are exported. However, if the MDR is defined as a data provider to CA CMDB, users browsing a z/OS CI in the CA CMDB repository can retrieve these additional attributes about the CI.

- **User Interfaces**



Enables users to interact with the product. This includes:

- **Web Interface (WebCenter)** which enables CA CMDB analysts to discover and export z/OS CIs to the CA CMDB repository.
- **3270 Interface** which enables z/OS systems programmers to administer the region. This interface also provides the functions that are available from the web interface.

The installation and setup of the product is described in the *Product Guide* and typically consists of the following phases:

1. Discover the resources on a z/OS system using the discovery methods.
2. Rediscover resources over a period to ensure that the MDR is complete and current.
3. Perform initial population of CA CMDB using a full MDR export.
4. Review the data, and establish a standard for updating CA CMDB using MDR exports by date range.
5. Automate the discovery and export process.
6. Automate the transfer of the MDR export file from the z/OS system to the CA CMDB server and the execution of GRLoader to import the file.
7. Roll out the automation to production.

Depending on the number of CIs and relationships discovered, initial importation of a full MDR to CA CMDB can take several hours. Subsequent updates are usually small. GRLoader includes optional parameters to optimize the process. The following are several tips to assist in this process:

- To improve performance for large files by preloading data, specify the -P parameter.
- To improve performance for large files by checking XML only (not the data), specify the -C parameter.
- To permit updates, specify the -a parameter.
- To get help, specify the -h parameter.

Automating the Process

Once you have successfully discovered CIs and imported them into the CA CMDB, you should establish the appropriate processes to maintain the integrity of the z/OS CIs in the CMDB by conducting regular discovery and export of changed or new MDR records. The method you choose to update the CMDB will depend on what change control standards you have established at your site. For example, a simple approach would be to export the MDR to XML by date range and use the GRLoader to perform updates on a regularly scheduled basis. You should also consider automating the process once you are satisfied with the results.

Automation enables you to keep the MDR up to date, for example, by executing the discovery process regularly using a scheduler. You can automate the discovery and export processes using the DISCOVER and EXPORT region commands. Your automation application can issue these commands using the MODIFY system command. Automation can continue after the export process by transferring the exported XML file to the CA CMDB server by using, for example, CA XCOM Data Transport. On receipt of the file, the server can automate the execution of GRLoader.

For more details consult the *CA CMDB Connector for z/OS Product Guide*. Additional details regarding GRLoader are provided in Chapter 2.

Health Check

The CA CMDB Connector for z/OS provides five separate health check points:

- Initialization Health Check - Ensure product region has initialized properly.
- ACB Health Check - Ensure the regions primary access method control block (ACB) is open and ready for communication.
- MDR Health Check - Warn of MDR approaching EXTENT limit.
Ensure the MDR file (used to hold discovered CIs) is within an acceptable threshold concerning the number of extents available.
- AOM Status Check - AOM is required to issue system commands for discoveries.
Ensure that the Automated Operations Management (interface to z/OS) is operational for its discovery processes.
- Web Interface Stack Check - The web interface is available.
Ensures the port has been configured (WebCenter Parameter Group) or port is not started (TCP/IP may not have been started).

Reference Documentation

For additional information on using the CA CMDB Connector for z/OS, consult the *Product Guide*.



Chapter 8: CA CMDB Architecture and Failover Considerations

This chapter provides a general discussion of the CA CMDB architecture, including high availability considerations.

Components

This section describes the software included with CA Service Desk Manager and breaks down the smaller software components included to support a CA CMDB implementation.

CA Service Desk Manager Solution

CA CMDB is included with the CA Service Desk Manager Solution. CA Service Desk Manager is the industry's most complete ITIL® service desk product, integrating Incident, Problem, Change, CMDB, Application Dependency Mapping, Knowledge and Support Automation into a single product to deliver fast time to value with low total cost of ownership. The CA products included in the shipment of CA Service Desk Manager R12.1 are:

- CA Service Desk r12.1
- CA Knowledge Tools r12.1
- CA CMDB r12.1
- CA Cohesion ACM 5.0
- CA Support Automation 6.0 SP1
- CA Business Intelligence 2.1
- CA Business Intelligence Fix Pack 4.5
- FAST ESP 5.1.3
- CA Workflow 1.1.15 SP5

CA CMDB 12.1

CA CMDB R12 is packaged with CA Service Desk Manager 12.1 but has its own installation media. The components available for installation from the CA CMDB 12.1 media are:

- CA MDB 1.5
- CA CMDB r12.1
- CA CMDB Visualizer r12
- CA Advantage Data Transformer (CA ADT) 2.2



Distributed Implementation

- CA CMDB Federation Adapters
- CA Embedded Entitlements Manager (CA EEM) 8.4

CA Cohesion ACM 5.0 also has separate media for its installation.

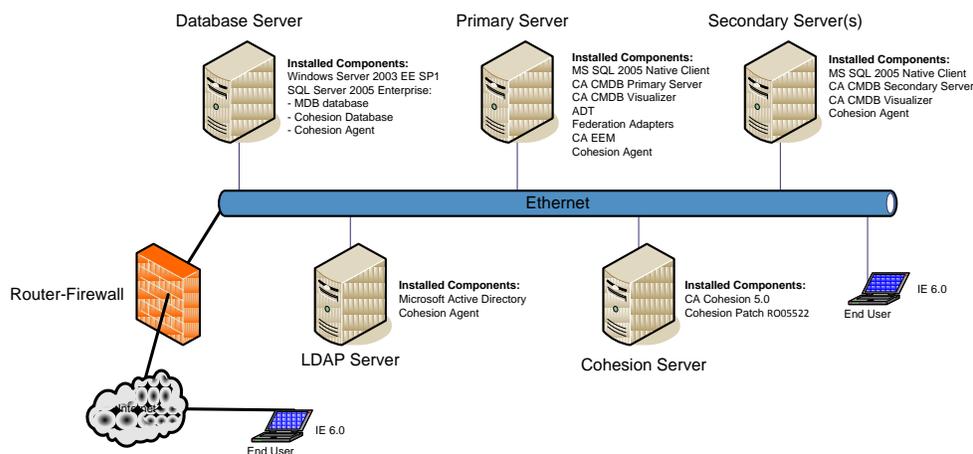
Distributed Implementation

Architecting a CA CMDB solution takes knowledge of the components to be deployed and a good understanding of how each component will be used in the overall solution. In a complete Service Management solution, CA Service Desk and CA CMDB are architected to be installed together, but there might be an occasion when Configuration Management is the sole business focus and only CA CMDB is included in the solution plan. In this case, CA CMDB can be deployed stand alone without CA Service Desk.

For information on implementing and installing the CA CMDB r12 components, refer to the *CA CMDB Implementation Guide*. Actual recommendations for CA CMDB deployments and component installation will vary based on budgets, hardware available, business requirements, and system usage.

The following diagram represents a high level CA CMDB distributed implementation with Cohesion.

CA CMDB r12 Network Diagram



The five servers in the example architecture are discussed below.

Database Server

The Database Server will host the Management Database (MDB) which is the common operational store or repository that CA products use to store data.

The recommended best practice is to host the MDB on its own server separate from the CA CMDB.

If there is an existing database server in the production environment that has adequate available resources, it can be used to store the MDB and the Cohesion database. In the above example, Microsoft SQL Server was the database selected for the architecture. The selection of a database vendor is a decision that should be made by the owner of the solution and based on available resources such as database knowledge, administrators, management, backup plans, and future roadmaps.

In the example architecture above, the following components are installed on the database server:

- Database: Microsoft SQL Server 2005
- Remote MDB database
- Cohesion database

A typical database server will have at least a dual processor with 4GB RAM or quad processors with 8GB of RAM, depending on the type, size, and requirements of the architecture.

This server can be a single point of failure if there is no failover plan architected. A database backup plan should be in place for the MDB, such as nightly data backups or high availability configuration through XOSoft or clustering.

For additional information on the CA MDB, refer to the *CA Management Database r1.5 Overview Guide*.

Primary Server

This is typically a new server that will be introduced into the environment.

The following components will be installed on this server:

- **MS SQL 2005 Native Client** – To allow connection to the database server.
- **CA CMDB** – CA CMDB Primary Server Application. The Primary Server is made up of three main components.



- Object Manager (domsrvr)—the distributed object manager server. It is the most important process in the CA CMDB product. When you install a primary server, two Object Managers are installed by default: one for client connections and one dedicated to Web Screen Painter (WSP). When you install a secondary server, you can configure additional Object Managers to run. For scalability, it is recommended that you configure multiple domsrvrs/webengine pairs either on your primary server or on secondary servers as deemed appropriate to handle your expected end user load.
 - Web Engine (webengine) — The Web Engine provides the web server used to support access to CA CMDB via a browser. It connects to web browsers through a pdmweb cgi running on a Microsoft IIS or Apache Tomcat web server. There must be a web engine for WSP on the primary server so WSP Schema Designer can write schema files. Web engines are the true client of an Object Manager for user client web browsers. Web engines cache html web forms for connected users. You can manipulate the cache using the pdm_webcache utility and see web client connection statistics using the pdm_webstat utility.
 - Web Director — an optional process that provides simple load balancing among multiple web engines. The Web Director selects the Web Engine with the least amount of active users and redirects the user(s) to the desired web engine. Subsequent requests are handled by the web engine directly without involving the webdirector. It is also possible to use the webdirector to provide enhanced security for login while allowing most user interactions to use a higher performance standard connection. The system administrator can configure the webdirector to direct login requests to a specific webengine that uses the SSL (secure socket layer) protocol. Once a user has been authenticated, subsequent requests are redirected to a different webengine using a standard protocol. Web Director is a specially configured pdmweb cgi.
- **CA CMDB Visualizer** – Visualizer application to graphically depict CI relationships.
 - **CA Advantage Data Transformer (ADT)** – Tool for mapping data from one source into the CA CMDB.
 - **Federation Adapters** – Adapters to assist in mapping and loading data into the CA CMDB.
 - **CA Embedded Entitlements Manager (CA EEM)** – Embedded Entitlements Manager is a central repository of user information. CA CMDB only uses Embedded Entitlements Manager for authentication and typically it is not included in a standalone implementation if there is already an existing LDAP directory in the environment that can be pointed to for user authentication. This component might be included if there are multiple shared LDAP directories in the environment that hold user information instead of just one. The top directory can be associated with CA EEM so that the users belonging to the shared directory trees below can be authenticated into CA CMDB. For a CA CMDB installation, the location of the CA EEM installation does not matter.

EEM 8.4 stores all of its data in the CA Directory, its internal storage system, as opposed to in Ingres in earlier releases. While, EEM doesn't use the MDB, collocation of the EEM server with an application data store / MDB server may make sense for a small, lightly loaded system.

For a small single product installation of CA CMDB, it is acceptable to co-locate the EEM server and the CA CMDB application to reduce communication overhead. If a backup system or HA solution is required for CA CMDB an alternate EEM server can be collocated with the backup/alternate application server.

For a large single product installation of CA CMDB, if the peak load of authentication is significant, then it may be optimal to separate the EEM server from the application. For example if peak conditions result in a heavily loaded CA CMDB Primary server, then either the server / processor capacity should be increased, or the EEM server should be on a separate server.

- **Cohesion Agent** – A Cohesion Agent is installed on each server in the environment.

Determining the optimal distribution of these components depends on the environment use, supported number of users, available hardware and budget, as well as scalability, availability, and failover requirements.

A typical primary server will have at least a dual processor with 4GB RAM or quad processors with 8GB of RAM, depending on the type, size, and requirements of the architecture.

This server can be a single point of failure for the CA CMDB solution if there is no failover plan architected.

Secondary Server

CA CMDB may be configured with secondary servers to boost system performance. Secondary servers can be added as remote servers (closer to remote clients) to improve performance for these remote clients. The secondary server is a component for scaling large production architectures but is typically not part of a development environment.

The following components will be installed on this server:

- **CA CMDB** – CA CMDB Secondary Server Application. The Secondary Server is made up of the same three main components as the Primary Server. Determining the optimal distribution of these components depends on the environment use, supported number of users, available hardware and budget, as well as scalability, availability, and failover requirements.
 - Object Manager (domsrvr)
 - Web Engine (webengine)
 - Web Director



- **CA CMDB Visualizer** – Secondary Visualizer application to graphically depict CI relationships. It is suggested to have a Visualizer installed along with every CA CMDB server, but it is not necessary. The Visualizer is launched primarily in context from CA CMDB, so installing a Visualizer on a secondary server keeps the load balanced across the servers. If there are no secondary Visualizers installed the request will go to the primary Visualizer on the primary CA CMDB server, thus potentially overloading the primary server.
- **Cohesion Agent** – A Cohesion Agent is either installed on each server in the environment or execution of the Cohesion Blueprints is requested through ssh from the Cohesion Server.

The Secondary Server in the example above runs one or more object manager/web engine pairs. Each pair will support between 150-250 concurrent user connections, depending on the type of user connected. Each pair requires one dedicated GB of RAM. A typical secondary server will have at least a dual processor with 4GB RAM supporting two object manager/web engine pairs or quad processors with 8GB of RAM supporting 6 object manager/web engine pairs, depending on how large the architecture needs to scale.

There can be multiple secondary servers in an environment which will be placed either close to the Primary Server or local to the end users. The existing network will determine what configuration will result in the greatest web interface performance.

Failover for Web Engines can be configured using the `pdm_edit` script provided so that end users will be automatically redirected to a different Web Engine if the one they are currently connected to or the hardware that houses that Web Engine fails.

LDAP Server

This server will be pre-existing in the architecture and houses a user directory service such as Microsoft Active Directory. If all users that require access to the CA CMDB application exist in the directory, EEM typically won't be a component in the architecture.

Cohesion Server

CA Cohesion ACM is a standards-based suite of products that lets you manage your enterprise's distributed hardware and software services from a centralized browser-based window.

One or more Cohesion servers will be introduced into the environment so that server hardware and software components can be discovered, related, and imported into the CA CMDB. The Cohesion Server can normally handle approximately 500 application servers, with the number increasing to 1,000+ as the number of discovered applications and the size of the servers decrease. At a minimum, each Cohesion server should have dual processors with 4GB of RAM allocated to it.

Scalability

CA CMDB is a highly scalable solution. It has the flexibility to add secondary servers to support additional connections as you grow, and the ability to distribute the server and its components.

The first thing that must be decided when designing the CMDB solution is the type of implementation. The most scalable architecture is the Distributed Architecture described in the previous section. In these designs, the primary server is the application server and all web services are handled by one or more web servers.

When there is a requirement to scale to support additional users as a result of company growth, secondary servers can easily be added. Should there be an acquisition or new site or location that would need to access the CMDB, a secondary server could be added locally to support those users.

The Web Director component plays an important role in an environment with multiple secondary servers.

The rule of thumb for how many concurrent connections can be supported per web engine is approximately 250 to 400 users, depending on the server load. The actual results are heavily dependent on the hardware, overall system load, and what the users are doing on the system. For example, queries and modifications or creations of objects by the user would utilize heavier bandwidth than a user who is just looking at a single object in read-only mode.

Whenever a secondary server is added, it is important to include an Object Manager and Web Engine pair. It is also possible to add additional Object Manager and Web Engine pairs to a single server if server hardware is sufficient. Each pair requires one dedicated CPU with a recommended additional 1 GB of RAM.

High Availability

The architecture above can be modified to provide a level of high availability for the MDB and CA CMDB application. Typically the cost of supporting a highly available architecture increases as the required up time increases. This is due to the additional hardware costs needed for redundancy and the cost of resources to support the solution.

High availability should be maintained on several levels:

- On the first level, a clean Uninterruptible Power System (UPS) will provide continuous service.
- A diesel generator added to the UPS will maintain power during blackouts.
- The servers themselves should have dual power supply, dual network interface controller (NIC), and RAID Level 5 or better for disk redundancy.



High Availability

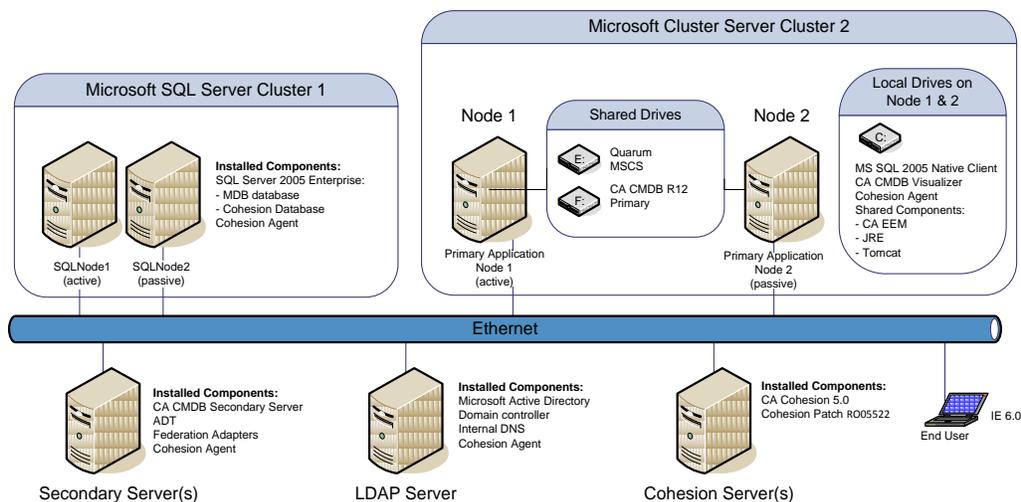
The existence of all this redundancy in the server components will not necessarily prevent the failure of the server over time and subsequent downtime, but it will extend the life of the server before a failure occurs.

The CA CMDB is a web-based application. The MDB stores the data when an object is generated. This environment will include a primary CA CMDB application server, database server, and possibly a few secondary web servers. The primary CA CMDB application server is not cluster aware, but is cluster tolerant. Therefore, to achieve high availability on the next level, each server must have its own redundancy.

Windows Environment with Microsoft Clustering

The diagram below depicts an example architecture where both the MDB and CA CMDB application were designed for high availability using Microsoft Clustering. The same theory will apply to other clustering tools.

CA CMDB r12 MSCS Network Diagram



NOTE: All Cluster Servers are Windows Server 2003 EE SP1

This architecture may be a good solution for organizations that need as close to 24x7 availability of the CMDB as possible.

Following is an example of how each Windows server is designed for high availability.

MICROSOFT SQL DATABASE SERVER

The database uses Microsoft SQL Clustering to maintain high availability. When using MS SQL 2005, the cluster must have two separate nodes. The configuration of the SQL cluster creates an active/passive mode. Fiber channels using dual paths attach the two cluster nodes to the storage file system. The storage file system is usually a Storage Area Network (SAN) that provides

redundancy and shared storage. If an end user is in the process of creating or saving a CI record, the record will be successfully saved without losing the data entered in the New Configuration Item form. The failover of the database is transparent to the end user aside from a small delay on the form of about 30-60 seconds that will be seen when saving the record. The end user remains logged into the CA CMDB application and does not need to log in again.

PRIMARY APPLICATION SERVER

The primary CA CMDB application server can be replicated to the backup primary server by using Microsoft Windows 2003 clustering. The cluster servers need to be in an active/passive mode configuration. As noted earlier, the primary server is not cluster aware, but is cluster tolerant. This means that the primary servers will not automatically fail over to the backup node on the cluster. Scripts will need to be in place to identify the cluster name and to execute the failover. The clustered node that is now the live active primary application server is recycled and attached to the MS SQL database.

If a failover of the active CA CMDB application node occurs, the end user will be disconnected from the application and will be required to log in again. If the end user is in the process of creating or saving a CI record, the record will not be saved, and the end user will be presented with the log in screen once the CA Service Desk Server service on the backup node is on line.

SECONDARY WEB SERVERS

The secondary web servers are located in a single location or region where users can gain access to the CA CMDB application. The secondary web server has one connection that goes back to the primary application server. To create the HA environment with the web servers, a minimum of two servers should be placed in each location. The secondary web servers use the web director component. This component provides load balancing between the web servers and maintains a single URL between the web servers.

If one of the two Secondary Servers fails, users logged into a web engine on that server will lose connection. Any information that has not been checked in or saved at the time of the server failure will be lost. Users will be presented with an option to sign on again to one of the remaining available web engines on the other Secondary Server.

High Level Install Steps

The high level steps for implementing this architecture are:

1. Check hardware and software pre-requisites
2. Ensure that the SQL Cluster is active, working, and fails over as expected



High Availability

3. Install CA MDB on SQL Node 1
4. Install CA MDB on SQL Node 2
5. Optional: Install EEM on the SQL database server
6. Install CA CMDB Primary Server on Application Node 1
7. Install CA CMDB Primary Server on Application Node 2
8. Start the CA Service Desk Server service
9. Test MDB Failover
 - Log into the CA CMDB web interface.
 - Enter data for a new CI into the Configuration Item Edit screen and keep the form open without clicking save.
 - Move the SQL Group.
 - Go back to the CI edit screen and save the data on the form.
 - You may see a short delay of 30-60 seconds but the data should be saved.
10. Test CA CMDB Application Failover
 - Log into the CA CMDB web interface.
 - Enter data for a new CI into the Configuration Item Edit screen and keep the form open without clicking save.
 - Move the Cluster Group that includes the CA CMDB Application.
 - Verify that the CA Service Desk Server service successfully re-started on the failover node.
 - Go back to the CI edit screen and save the data on the form.
 - A message will be presented to the end user that the server has been restarted. Any data not saved will be lost.
11. Install CA CMDB on dedicated Secondary Server(s) and deploy Web Services
12. Configure Object Managers, Web Engines, and Web Directors on the active Primary Server Node and the Secondary Server(s).
13. Install CA CMDB Visualizer on the Primary Server's local drive on Node 1
14. Install CA CMDB Visualizer on the Primary Server's local drive on Node 2
15. Install CA CMDB Visualizer on Secondary Server(s)
16. Test Visualizer
 - Log into the Visualizer to test connection
 - Fail over the database and verify that the Visualizer launches post-failover
 - Fail over the CA CMDB Application and verify that the Visualizer launches post-failover

17. If required: Install EEM on the local drive of Application Node 1
18. If required: Install EEM on the local drive of Application Node 2
19. If required: Configure EEM Server failover between Application Node 1 and 2
20. Optional: Configure Security through external LDAP or EEM
21. Install ADT and Federation Adapters on Primary Server
22. Install Cohesion on dedicated Cohesion Server
23. Install Cohesion Patch on Cohesion Server
24. Install Cohesion Agents on all required servers
25. Configure integration with Cohesion and CA CMDB

UNIX or Linux Environment

The following is an example of how each server will be designed for high availability in a UNIX or Linux Environment.

PRIMARY APPLICATION SERVER

The primary CA CMDB application server in a UNIX and Linux platform can maintain an HA environment by doing the following:

- Build a hot standby primary server and have it available.
- When there is a failover to the standby server, do the following:
 - Have a script built to rename the standby server name from the production application server name.
 - Have the script to change the standby IP address from the production application server IP address.
 - Run pdm_configure.
 - Start the CA Service Desk Server service.
 - Test to make sure logins and application is running.
- Maintain strict change management on the production and backup server. Make sure all maintenance and patches are applied to both servers.
 - Test and validate the process of failover at least on an annual basis.

ORACLE DATABASE SERVER

Currently the CA CMDB application supports the 10g version of Oracle. A few options that maintain HA on the Oracle database are as follows:

- Oracle 10g Real Application Clustering (RAC)
 - Provides multiple nodes for high availability.



Reference Documentation

Provides fiber channels using dual paths to attach the two cluster nodes to the storage file system. The storage file system is usually a Storage Area Network (SAN) that provides redundancy and shared storage.

- CA XOssoft HA Option for Oracle

Provides asynchronous replication over LAN or WAN.

- Veritas Oracle cluster

Provides multiple nodes for HA.

Reference Documentation

For more information consult the *CA CMDB Release Notes*.

Glossary

ADT

Advantage Data Transformer. ADT is a rapid visual development environment for enterprise data transformation and integration. Using ADT, data can be moved from any ODBC, CSV, TEXT or XML source and mapped to any target repository. Additionally, it is possible to transform and realign the data as data movements take place. ADT is the underlying technology that facilitates all data imports into the CA CMDB system.

Asset Attributes

With the CA CMDB we associate a specific set of attributes to each CI Family. These attributes enable a clear identification of the information relevant to that type of CI. Attributes are stored in extension tables linked to the Asset Family.

Asset Class

CI classes provide a more specific classification and are used to further categorize Configuration Items within a single Asset Family. For example “Unisys” is a class within the Hardware.Server Asset Family.

Asset Family

CI families are usually used to categorize a particular CI in a very general sense. For example, Hardware.Server.

Asset Relationships

Asset Relationships represent the logical roles of the two CIs in the organization’s Service Model where each CI contributes to the Service, **providing** a “service” to a **dependent CI**.

CI

Configuration Item. The CI is the actual component being managed. This is determined by the CMS scope and may incorporate Services, Servers, Applications, Database, Network components, Process Documents, such as Service Level Agreements (SLAs), and other components that need to be managed.

CMDB

Configuration Management Database. A CMDB provides the repository of the Configuration Items (CIs) as well as their attributes, relationships and associated connectors to Incident, Problem and Change and Release Management components.

CA CMDB

The CA CMDB is a software solution providing the ability the manage configuration items within the organization and assist in the Service Asset and Configuration Management process.



CMDBf

CMDB federation working group comprised of several companies with the goal of setting a standard for managing information that is to be federated into a CMDB.

CMS

Configuration Management System. The CMS is composed of the CMDB(s) as well as the components used to manage this information and leverage this data.

CORA

Common Object Registration API. CORA reconciles assets from different management tools into a single representation of an asset. CA products which maintain asset data in the MDB are required to register the asset.

Federation Adapters

The ADT programs, included with CA CMDB, that pre-define sources and targets for data. The number of adapters included is increasing over time but currently include Unicenter Asset Management r4, Generic Spreadsheets, Generic ODBC sources and Microsoft SMS. Adapters can also be created to handle specific data import challenges within the deployment.

Federated Asset ID

Unique identifier used by the MDR to refer to a specific Configuration Item.

GRLoader

General Resource Loader.

ITIL

IT Infrastructure Library. The most widely accepted approach to Information Technology Service Management in the world. ITIL provides a cohesive set of best practices drawn from the public and private sectors internationally to define processes and interactions regarding service management.

MDB

Management Database. Common database shared by many CA solutions, including CA NSM, CA Service Desk Manager, CA Service Catalog and CA Asset Portfolio Management.

MDR

Management Data Repository. MDRs are trusted sources of information providing attribute and relationship data to the CA CMDB. The CA MDB is an example of an MDR.

MDR Launcher

CA CMDB component used to access the MDR from within the CA CMDB web interface. This enables you to obtain further information regarding that CI within the context of the source program.

SACM

Service Asset and Configuration Management process. The SACM process controls the management of the CMDB and its integration to other processes.

Visualizer

CA CMDB component which provides a graphical display of CI relationships.

Index

3

3270 interface • 128

A

access method control block (ACB) • 129

Alt Asset ID • 28

Asset Federation • 24

asset label • 28

asset matching logic • 28

asset tag • 28

assets

- Alt Asset ID • 28

- and CORA • 34

- classes • 34

- discovered • 106

- discovered vs. owned • 29

- matching logic matrix • 28

- owned vs discovered • 36

- reconciling cross-domain • 74

- serial number • 28

- weighting • 28

attributes

- mapping • 68, 94

- model • 68

- overview • 21

automating

- discovery • 125

- discovery and export • 129

B

browser based interface • 35

Business Process Views (BPVs) • 116

C

CA Advantage Data Transformer (ADT) • 24, 35, 42, 134

CA Asset Portfolio Management • 117

CA Business Intelligence • 131

CA Cohesion ACM • 42, 59, 131, 135, 136

CA Datacom • 124

CA IDMS • 124

CA IT Asset Manager • 26, 31, 34, 36, 117

CA IT Client Manager • 26, 31, 35, 36, 117

CA Knowledge Tools • 131

CA Network and Systems Management (NSM) • 26, 31, 35, 105

CA Service Availability Management Pack (SAMP) • 116

CA Service Catalog • 117

CA Service Desk • 101

CA Service Desk Knowledge Tools • 41

CA Service Desk Manager • 26, 31, 36, 39, 41, 97, 125, 131

CA SPECTRUM • 87

- integration script • 95

CA SPECTRUM Modeling Gateway Toolkit • 87

CA SPECTRUM OneClick • 98

CA Support Automation • 131

CA SupportBridge • 41

CA Systems Management Pack (SMP) • 116

CA Wily Customer Experience Manager (CEM) • 117

CA Wily Introscope • 117

CA Workflow • 131

Capacity Management • 119

Change and Configuration Management) • 112

Change Impact Analyzer (CIA) • 105, 106

Change Management • 12, 41, 119

CICS • 124

classes

- and CORA • 34

- CIA • 109

- default • 21

- mapping • 72, 78

- overview • 12

clustering • 138

CMDB Connector for z/OS • 124

CMDB Export Report • 65

CMDB Federation Working Group (CMDBf) • 12, 38

cmdb_mapping.xml file • 68

CohesionAttr attribute • 78, 79

Common Asset Viewer (CAV) • 36, 37

Common Object Registration API (CORA)

- asset matching logic • 28

- master asset data model • 27

- MDB • 31

- NICs • 78



- reconciliation matrix • 28
- version • 27
- Configuration Items (CI)
 - what is • 12
- Configuration Items (CIs)
 - associated MDRs • 22
 - attributes • 21
 - classes • 34
 - Common Asset Viewer • 36
 - create new • 106
 - duplicate • 71
 - exporting from CA Cohesion • 64
 - exporting from CA SPECTRUM • 92
 - exporting from mainframe systems • 127
 - families • 19
 - import through Cohesion • 60
 - launch in context • 35
 - loading data • 23
 - mapping • 24
 - reconciling through CORA • 27
 - relationships • 21
 - synchronize status • 114
 - type mapping • 23
 - understanding • 17
 - universal attributes • 39
 - virtual CI classes) • 79
- Configuration Items(CIs)
 - classes • 21
- Configuration Management System (CMS) • 12, 78, 118
- connectors • 124
- coraver command • 27
- create
 - services • 94
- custom
 - Federation Adapters • 24

D

- Data Exporter • 128
- DB2 • 124
- default
 - CI classes • 21
 - CI families • 19
 - Federation Adapters • 24
 - relationship types • 22
- Discovered Host Name field • 76
- Discovered vs. Owned Assets: • 29
- discovery • 125
 - cross-domain • 74

- Discovery Profile • 65
- DNS Name • 28
- domsrvr • 134

E

- Embedded Entitlements Manager (EEM) • 134
 - exporting CIs • 64
 - from CA SPECTRUM • 92
 - from mainframe systems • 127

F

- failover • 137
- families
 - attributes • 21
 - overview • 12
- Families
 - default • 19
 - in Service Desk Manager • 34
- FAST ESP • 131
- federated
 - CMDBs • 38
 - MDRs • 23
- Federation Adapters • 35, 134
 - custom • 24
 - default • 24
 - overview • 24
 - SMS • 24
 - UAM • 25
- federation APIs • 118
- filters • 13
- Fully Qualified Domain Names (FQDN) • 95

G

- General Resource Loader (GRLoader) • 25, 42, 60, 91, 98, 100
 - overview • 23

H

- health check • 129
- high availability • 137
- host name • 28

I

- impact analysis • 121

- importing
 - data from CA NSM • 112
- Incident Management • 12, 41, 119
- incidents vs. issues • 101
- integration script • 95
- IT Resource Management • 31
- IT Service Model • 22
- IT Services
 - creating • 94
 - modeling • 17
 - overview • 17
- ITIL • 12, 41, 118

K

- key indicators • 118
- Knowledge Management • 41

L

- launch in context • 35, 38, 61, 82
- LDAP • 134, 136
- LPARs • 124

M

- MAC address • 28, 79
- mainframe integration • 124
- Management Data Repositories (MDRs) • 43
 - automating update) • 129
 - defining CA Cohesion • 82
 - defining CA SPECTRUM • 97
 - defining Cohesion • 61
 - definition • 39
 - federated • 23
 - for mainframe • 127
 - overview • 12, 22
 - provider • 24
 - remote • 100
- Management Database (MDB) • 26, 31, 35, 106, 133
- mapping • 25
 - attributes • 68
 - CA SPECTRUM • 91
 - CIs • 24
 - class • 68
 - class) • 72
 - classes • 78
 - relationships • 68

- virtual CI classes • 79
- mapping file modifications • 93
- Master Asset Data Model • 27
- MDB Schema Viewer • 35
- MDR Launcher • 23, 35, 42, 98, 102
- model attribute • 68
- modifying
 - mapping file • 93
- MQSeries • 124
- Management Data Repositories (MDRs) • 102
- multiple NICs • 74

N

- Network Interface Cards (NICs) • 74, 78
- Note about Asset Classes • 34

P

- pdm_startup file • 108
- PinkVERIFY • 41
- Problem Management • 41

R

- reconciling
 - discovered servers • 76
 - hardware • 74
 - NIC relationships • 78
- relationships
 - CIA • 110
 - exporting • 68
 - mapping • 68
 - overview • 21
- remote
 - MDRs • 100
- Request Management • 42
- root cause analysis • 87, 105, 119
 - filter • 13
- rule set • 94

S

- scalability • 137
- serial number • 28
- servers
 - reconciling • 76
- service
 - definitions • 118



- Service Asset and Configuration Management (SACM) • 12
- service level agreements (SLAs) • 117, 125
- Service Management • 120
- service model • 94
- services
 - viability • 95
- SMS adapter • 24
- spectocmdb.pl script • 95
- synchronizing CI status • 114
- system_name attribute • 71
- systems performance management • 31

T

- Third party database tools • 43
- Transaction Artifacts (TAs) • 39
- type mapping • 23

U

- UAM Adapter • 25
- Unified Service Model (USM) • 118
- Universal Federation Adapter • 42
- Universally Unique Identifier (UUID) • 27
- updating CMDB • 91

V

- virtual CI class mapping • 79
- visualization
 - CMDB Visualizer • 38, 42, 112, 121
 - MDR Launcher • 23, 35, 42, 98
 - overview • 13
- Visualizer • 38, 42, 112, 121, 134

W

- Web Director • 134
- Web Engine • 134
- workflow • 41

X

- XML files • 25

Z

- z/OS • 124