CA Catalyst

CA Configuration Automation® Connector Guide

r12.8 SP02

CA Technologies Product References

This document references the following CA Technologies products:

- CA Configuration Automation®
- CA Catalyst
- CA IT Process Automation Manager (CA IT PAM)
- CA NSM
- CA Service Desk Manager (CA SDM)
- CA Spectrum® Service Assurance (CA Spectrum SA)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Overview

This section contains the following topics:

About This Guide

This guide describes how to install and configure the CA Catalyst connector for CA Configuration Automation.

In general, CA Catalyst connectors expose product data to consuming products such as CA CMDB, CA Spectrum Service Assurance (CA Spectrum SA), and CA IT Process Automation Manager (CA IT PAM) for visualization, analysis, and management in a unique, broader context.

This guide contains information specific to the CA Configuration Automation connector. For general information about CA Catalyst connectors and the CA Catalyst infrastructure, information that applies to all connectors, and information about custom connector integrations, see the *Connector Guide* distributed with CA Catalyst.

Terminology

The following list contains concepts and terms that may be useful if you are integrating a CA Catalyst connector with CA Configuration Automation or other consuming products for the first time:

Connectors

Connectors are the links from products that consume connector data to external products, referred to in this document as domain managers. Each connector retrieves information from its domain manager and transmits the information through the connector framework to the consuming product for visualization and analysis. Connectors can also enact inbound operations on data in the source domain manager, such as object creation. CA Catalyst connectors use a unified connector framework to enable integration with multiple consuming products.

USM

The *Unified Service Model* (USM) is a schema of common object types and properties to which data from all connectors is converted. The USM schema enables analysis of data from all domain managers in a common interface with identical formatting.

Configuration Items (CIs)

Configuration items (CIs) are representations of IT elements managed by a domain manager. Each CI belongs to a *type* (defined in the USM schema), such as ComputerSystem, Database, Process, Relationship, and so on. Services are composed of CIs, and you define relationships between CIs in services.

Connectors transform managed objects from integrated products to adhere to the USM schema and import the objects into the consuming products as CIs.

Services

Services represent discrete business functions that can contain configuration items managed by multiple domain managers. For example, a payroll service may contain an Active Directory database managed by Microsoft SCOM, a user store managed by a security product, batch jobs managed by a mainframe product, a router managed by a network product, applications managed by an application management product, and so on.

Alerts

Alerts are the CA Catalyst mechanism for reporting fault conditions and service degradation. *Infrastructure alerts* are fault conditions originally reported by one of the domain managers (such as a CA NSM event or CA Spectrum alarm). An alert is associated with a corresponding CI, and associated alert severities determine CI condition and, ultimately, service impact. *Service alerts* are conditions generated by CA Catalyst based on analysis of a modeled service. Service alerts result when the condition of one or more CIs combines to impact the overall quality or risk level associated with the service.

Outbound from connector operations

Outbound from connector operations are operations that a connector invokes to import data from domain managers into consuming products such as CA Catalyst and CA CMDB. All connectors support outbound from connector operations.

Inbound to connector operations

Inbound to connector operations invoke changes in the domain manager data store as a result of changes to the imported data in the consuming product. For example, CI reconciliation in CA Catalyst can change the values of CI properties. Connectors that support inbound operations can then enact that change in the source domain manager so that its data matches the reconciled data. Or if a CI is deleted in a domain manager that CA Catalyst defines as a source of truth, connectors that support inbound operations can delete the CI in other domain managers with a record of that CI.

Inbound operations are not currently supported by the CCA connector.

CA Configuration Automation Connector

CA Configuration Automation monitors software configuration changes on targeted virtual and physical servers in your enterprise. It uses CA Network Discovery Gateway (NDG) to discover servers and software throughout your enterprise. These discovered servers and their software components can be managed individually, or organized into services and managed using the CA Configuration Automation UI. It enables you to automatically or manually run a number of operations that ensure configuration changes to any component being managed are known to the members of your data center who are tasked with monitoring these servers and services.

The following four operations can be configured to send alerts using the CCA connector:

- Server Change Detection
- Server Rule Compliance
- Service Change Detection
- Service Rule Compliance

To configure CA Configuration Automation to send alerts based on the results of the four aforementioned operations, the following configuration steps must be performed in CA Configuration Automation:

- The catalyst.events.enabled property must be set to True as described in View and Edit CA Configuration Automation Properties section of the *CA Configuration Automation Product Guide* or the online help.

  Note: If the catalyst.events.enabled property is *not* set to True, the following occurs:

  - No events are sent to the connector.
  - CI synchronization from CA Configuration Automation to the connector is not enabled, and the events for CI addition and deletion are not generated.

- When creating a management profile for servers or services, click the Change Detection Adhoc check box on the Management Options page to specify that when the management profile is run manually, an alert is sent using the connector.
- When creating a management profile for servers or services, click the Change Detection Scheduled check box on the Management Options page to specify that when the management profile is run as a scheduled job, an alert is sent using the connector.

Note: If you restart the CA Configuration Automation Server, you must restart the Catalyst Container hosting the CCA connector.

Integration Scenarios

The CCA connector interfaces with the CCA Server to expose server and software component configuration data for use by products that leverage the CA Catalyst infrastructure. Integrating CA Configuration Automation with consuming products provides the following benefits:

Configuration Management

The CCA connector provides CIs for servers, properties, and relationships to consuming products. CA Configuration Automation initially discovers a wide variety of server information using a series of network scans. Then it obtains details of the servers, server properties, software components, and their relationships. This information is critical for other CA and third-party products to understand the inventory of their network and how servers are related in the enterprise.

The CCA connector also sends alerts detailing results from CA Configuration Automation Rule Compliance and Change Detection operations. These alerts are associated with the respective CIs when the operations are run on CA Configuration Automation. The synchronizing provided by the connector enables configuration changes made on CA Configuration Automation to also be made on the consuming product.

Business Service Management

Exposing CA Configuration Automation service and CI data to CA Catalyst lets you evaluate the data in a different, broader business service context. CA Catalyst collects data from multiple managed products and lets you monitor imported services or model services from imported CIs. Service models may contain CIs managed by any number of products, and they give operations a clearer view of business impact when the following alerts occur:

- Configuration deviations in Change Detection Summaries of CA Configuration Automation-managed services and servers.
- Compliance rule deviations in Rule Compliance Summaries of CA Configuration Automation-managed services and servers.

Additionally, CIs imported from CA Configuration Automation can be grouped into services using service-modeling functionality. These CIs receive CA Configuration Automation Rule Compliance and Change Detection alerts when these operations are run on CA Configuration Automation.

Note: CA Configuration Automation users can schedule or manually run Rule Compliance and Change Detection operations. Both these operations can send alerts to CA Catalyst. These alerts also include launch in context URLs to view the Rule Compliance and Change Detections operations if scheduled. These URLs display the results only for the specified time based on the CA Configuration Automation configuration property.

Chapter 2: Installation

This chapter contains information about installing the CCA connector.

This section contains the following topics:

Operating Environment Support

The CCA connector supports the following releases:

- CA Catalyst r3.2
- CA Configuration Automation r12.8
- CA Configuration Automation r12.8 SP01

The CCA connector supports installation on the following operating systems:

- Microsoft Windows Server 2008 (32-bit and 64-bit) Release 2 Standard, Enterprise, and Datacenter with the latest service packs
- Microsoft Windows Server 2012 Standard edition
- Red Hat Enterprise Linux 5.0, and 6.0

Installation Considerations

Consider the following points while planning the CCA connector installation:

- The required CA Configuration Automation Java SDK is installed with the CCA connector. It provides the integration with the CA Configuration Automation.
- UNC shares are not supported by the CCA connector installation program.
- The CCA connector installation program requires you to provide the following CCA Server information:
  - Host name or IP address
  - Authentication credentials (user name and password)
  - Port number

Install the CA Configuration Automation Connector

You can install the CA Configuration Automation connector locally in the following locations:

- On the CCA Server
- On the Catalyst Server
- On a Linux or Windows system that is in the same domain as the CCA Server

Important! Verify that container installer version 3.2.0.0 or later is installed on the computer before you install the connector.

Follow these steps:

1. Double-click one of the following files from the connector package:
   - Connector_CCA.exe (Windows)
   - Connector_CCA.bin (Linux)

   If the computer meets all prerequisites, the installer Introduction page opens. If no container is installed on the computer, the Missing Prerequisites page opens.
2. Install the container if necessary:
   a. On the Missing Prerequisites page, click Install Now.

      The installer locates the container in the following default location:

      <installer launch dir>/Container folder

      If the installer finds no containers at the default location, a file selection dialog opens.

   b. Select the install.exe or install.bin file of the container, and then proceed with the installation.

      Important! For a Linux installation, verify that the install.bin file has execute permissions so it can start the container installer.
3. Click Next on the installer Introduction page.
4. On the License Agreement page, accept the agreement and then click Next.

   The Install Folder page opens, listing the available containers and the corresponding node names.
5. Click a container.

   If the product cannot resolve the corresponding node name, the Catalyst Container Server Configuration page opens.

6. Enter the name of the container node and then click Next.

   If a connector exists in the selected container and the connector version is lower than the installer version, the product prompts you to upgrade. Otherwise, the version conflict dialog indicates that the product is upgrading the higher version.

   Note: The product allows only one CA Configuration Automation connector in each container.

7. Click Upgrade to verify the upgrade process.

   If no connector instance exists on the selected container, the installation continues as a new installation. Complete the following fields on the Catalyst Server Configuration page:

   Host name

   Defines the Catalyst Server name.

   HTTP Port

   Defines the HTTP port on which the Catalyst Server listens.

8. Complete the following fields on the CCA Connector Configuration page:

   CCA Server Host Name

   Defines the name of the CCA Server that the connector monitors for alarms and updates.

   CCA Server Port

   Defines the port on which the CCA Server listens.

   Default: 8080

   CCA Server User

   Defines a user that can access the CCA Server.

   Default: ccaadmin

   CCA Server Password

   Defines the password that is associated with the specified user.

   Verify CCA Server Password

   Confirms that you entered the password accurately.

   CCA Notification Listener Port

   Defines the port that receives events from the product.

   Default: 7071

   HTTPS

   Specifies whether the target CCA Server is HTTPS-enabled.

X.509 Certificate Authentication

Specifies whether the target CCA Server is client authentication-enabled.

Certificate Path

Defines the path to the certificate associated with the CCA Server User.

Certificate Password

Defines the password that is associated with the certificate file.

9. Click Next.

   If the CCA Server provided has an existing registered connector, the product prompts you to select a different server.

   Note: The CCA Server supports only one connector instance.

10. On the Database Server screen, complete the following fields:

    Note: The database details on the Database Server and the Database Configuration pages must correspond to the CA Configuration Automation server details on the CCA Connector Configuration page.

    Database Type

    Defines the type of database that the CCA Server uses.

    Server Name

    Defines the name of the computer on which the CCA Database resides.

    Port Number

    Defines the port on which the CCA Database host listens.

    Default: 1433

    Instance Name (Optional)

    Defines the CCA Database instance name.

11. Click Next.

12. Complete the following fields on the Database Configuration page:

    Database Name

    Defines the CCA Database name.

    Default: cca

    Database User

    Defines the CCA database administrator user name.

    Default: cca

Database User Password

Defines the password that is associated with the specified CCA database administrator user.

Retype Password

Verifies that you entered the password accurately.

13. Click Next.
14. Complete the following fields on the Change Detection Alert Metric and Threshold Levels page:

Alert Metric

Specifies one of the following metrics, which (when it is combined with the threshold values) determines the alert severity:

CountChange

Defines that a Change Detection operation bases the alert severity on the number of changes from the source server to the target server.

CountSource

Defines that a Change Detection operation bases the alert severity on the number of changes to the source server.

CountTarget

Defines that a Change Detection operation bases the alert severity on the number of changes to the target server.

CountTotal

Defines that a Change Detection operation bases the alert severity on the total number of changes.

Information Threshold

Defines the minimum number of changes that the specified Change Detection metric requires to assign an Information severity level to an alert.

Default: 0

Note: Increase the threshold value with each severity level. Set the Information value as the lowest, the Minor value as the next lowest, and so on.

Minor Threshold

Defines the minimum number of changes that the specified Change Detection metric requires to assign a Minor severity level to an alert.

Default: 5

Major Threshold

Defines the minimum number of changes that the specified Change Detection metric requires to assign a Major severity level to an alert.

Default: 10

Critical Threshold

Defines the minimum number of changes that the specified Change Detection metric requires to assign a Critical severity level to an alert.

Default: 20

Fatal Threshold

Defines the minimum number of changes that the specified Change Detection metric requires to assign a Fatal severity level to an alert.

Default: 30

15. Click Next.
16. Review your selections on the Install Summary page and then click Install.

    The product installs the connector and integrates with the appropriate CA Configuration Automation and CA Catalyst instances.
17. Click Done on the Installation Complete page.

    The installation process creates a log file. If the installation summary page reports errors, review the following file to troubleshoot the installation:

    ```
    %CATALYST_HOME%\CCA Connector Uninstall folder \ Logs\
    CA_Configuration_Automation_Connector_Install%timestamp%.log
    ```

    Note: Restart the CCA server after the installation is complete.

Chapter 3: Configuration

This chapter describes how to configure the CCA connector after installation.

Configure the CA Configuration Automation Connector

After installation, you can change the connector properties you defined during installation and edit other properties to refine connector behavior or adjust to changes in the integrated product.

To configure the CCA connector

1. Log on to the Catalyst host server that contains the registry service, then navigate to the following directory:
\topology\physical*<CCA_connector_host_server>*\modules\configuration.
2. Open the CCAConnector_<CCAServerHost>_<CCA Server port>.xml file in an XML editor, change any of the following configuration parameters, and then save the file:

host

Specifies the CCA Server host from which the connector is collecting CIs and alerts.

port

Specifies the port on which CA Configuration Automation is running on the target CA Configuration Automation host.

username

Specifies a valid user name that can access CA Configuration Automation using the SDK.

password

Specifies the encrypted password for the specified username.

secure

Specifies whether the SDK uses HTTPS secure communication. Set the value to true to enable secure communication.

retry_count

Specifies the number of times to retry if there is a connection failure.

postfix

Specifies the end point on the server where the SDK webservice is accessible.

Default: /cacca/services/SDKService

cd_alert_metric

Specifies one of the following metrics that determines the severity level of the change detection alert:

CountChange

Specifies the number of changes that occurred from source to target.

CountSource

Specifies the number of changes that have occurred only in the source.

CountTarget

Specifies the number of changes that have occurred only in the target.

CountTotal

Specifies the total number of changes determined by the change detection operation.

cd_fatal_threshold

Specifies the minimum number of changes for the specified Change Detection metric required to assign a Fatal severity level to an alert.

cd_critical_threshold

Specifies the minimum number of changes for the specified Change Detection metric required to assign a Critical severity level to an alert.

cd_major_threshold

Specifies the minimum number of changes for the specified Change Detection metric required to assign a Major severity level to an alert.

cd_minor_threshold

Specifies the minimum number of changes for the specified Change Detection metric required to assign a Minor severity level to an alert.

cd_information_threshold

Specifies the minimum number of changes for the specified Change Detection metric required to assign an Information severity level to an alert.

notification_listen_port

Specifies the port that receives events from CA Configuration Automation.

delete_thread_interval

Specifies the interval for the timer thread. When the events are not turned on, the deletion of CI's in CA Configuration Automation needs to be synchronized with CA Catalyst. This option is not present by default and needs to be manually added if required.

Default (when added):15 minutes.

client_auth_cert_file

Specifies the path to the certificate for the user that is configured to connect to the CCA Server when client authentication is enabled on the CCA Server. This needs to be the certificate for the user configured in the username parameter earlier in this step.

client_auth_cert_password

Specifies the password for the certificate file configured in the client_auth_cert_file parameter.

client_auth_keystore_type

Specifies the keystore type for the certificate file configured in client_auth_cert_file. The default is PKCS12 (which is the format for the certificate downloaded from CCA server). The only other type supported is JKS which represents the java keystore type.

The property changes are saved.

Important! Do not change any other properties in the CCAConnector_<CCAServerHost>_<CCA Server port>.xml file unless you edit values in the Launch in Context Details parameter (see step 3).

3. (Optional) Change the host, port, or both values in the Launch in Context Details parameter if you changed either of these values in the file.

   Note: Do not change any other values associated with the Launch in Context Details functionality.

   The property changes are saved.

4. Stop and restart the connector.

   The connector restarts.

   Important! Do not perform rapid start and stop operations on the connector. Each stop and start sends the corresponding command to the connector. Rapid start and stop operations can cause these commands to queue on the connector and cause the connector to start and stop repeatedly until all commands in the queue are processed.

Chapter 4: Connector and CA Catalyst Interaction

This chapter describes how the CCA connector interacts with CA Catalyst and how CA Configuration Automation entities are mapped to the USM schema.

Note: When the CCA connector or CCA Server is restarted, the Compliance Status of the various Compliance CIs is reset to Unknown. This status is updated to Compliant or Non-Compliant after the next manual or scheduled compliance operation (that is, Change Detection or Rule Compliance) on the CCA Server.

This section contains the following topics:

Outbound from Connector Operations

The CCA connector can invoke outbound from connector operations to import the following CA Configuration Automation data into consuming products:

Automatic CI, service, and relationship synchronization

Imports CA Configuration Automation CIs and service definitions and continually synchronizes updates to CI, services, and relationships in CA Configuration Automation.

CI and relationship updates

Uses the CcaSiloConnector class libraries to monitor addition and deletion of CA Configuration Automation.

Alerts

Uses the CcaSiloConnector class libraries to monitor additions and modifications to CA Configuration Automation alerts. Specifically, the connector uses Operations Manager Connector Framework to synchronize alert data between CA Configuration Automation and CA Catalyst.

CI types and classes

Imports the following object types as the indicated classes:

- Services
- Servers
- Components

- Compliance Status (results of Rule Compliance and Change Detection Operations)
- Relationships (hierarchical relationships from CA Network Discovery Gateway (NDG) and virtualization relations)

USM Data Mapping

When connectors import services and CIs, they normalize the classes, properties, relationships, and severities to adhere to the USM schema. This section lists the CA Configuration Automation classes, severities, and relationships and their USM mapping after the import.

Note: For more information about CI property mapping, see the CCA connector policy file located at \topology\physical*<CCA_connector_host_server>*\modules\policy in the registry service.

Severity Mapping

The following table shows how the CCA connector maps CA Configuration Automation alert severities to USM severities:

CA Configuration Automation Severity	USM Severity
Information	Minor
Warning	Major
Error	Critical

Relationship Mapping

The following table shows how the CCA connector maps CA Configuration Automation component property values to USM component relationships:

CA Configuration Automation Component	USM Component Relationship
comp_uuid	Source_uuid
Srvr_uuid	Target_uuid
relationship	HasAccessTo/HasRequirementFor

The following table shows how the CCA connector maps CA Configuration Automation component property values to USM virtualization relationships:

CA Configuration Automation Component	USM Virtualization Relationship
source_uuid (Server)	Source_uuid
target_uuid (Server)	Target_uuid
relationship	IsManagedBy/IsHostedBy

Rule Compliance Mapping

The following table shows how the CCA connector maps CA Configuration Automation Rule Compliance results to USM ComplianceStatus:

CA Configuration Automation RuleComplianceSummary	USM ComplianceStatus
Server or service uuid concatenated with ComplianceStatus	MdrElementId
Update the URL to launch CA Configuration Automation in context to view the summary	UrlParams

Change Detection Mapping

The following table shows how the CCA connector maps CA Configuration Automation Change Detection results to USM ComplianceStatus:

CA Configuration Automation ChangeDetectionSummary	USM ComplianceStatus
Server or service uuid concatenated with ComplianceStatus	MdrElementId
Update the URL to launch CA Configuration Automation in context to view the summary	UrlParams

Background Process Mapping

The following tables show how the CCA connector maps CA Configuration Automation background processes to USM processes:

Services	[acm_os_svc]	Property Name
service display name	display_name	NamedAliases
service key name	svc_name	ProcessName/ProductName
service logon as	logon_as	usm-core2: LogOnAs
service path	path	usm-core2: ExecutablePath
service startup	startup	StartupType

Open Ports	[acm_open_ports]	Property Name
port	port	AccessedViaTcpPort/usm-core2:AccessedViaUdpPort
protocol	protocol	TCP/UDP
process name	name	ProcessName/ProductName
process path	path	usm-core2:ExecutablePath

Communication Relationships	[acm_comm_relshps]	Property Name
application name 1	column not in table, but returned as a result of accessing view	ProductName
application name 2	column not in table, but returned as a result of accessing view	ProductName

Netstat Communication Relationships	[acm_netstat_relshps]	Property Name
process name 1	process_name_1	ProcessName
executable path 1	exec_path_1	usm-core2:ExecutablePath
process name 2	process_name_2	ProcessName
executable path 2	exec_path_2	usm-core2:ExecutablePath

Binary Relationship Mapping

The following tables show how the CCA connector maps CA Configuration Automation binary relationships to USM relationships:

Network Details from BP (Windows)	[acm_param]	Property Name
cluster status	if "name"="Cluster Status", consider "value" column	usm-core2:MemberStatus
clustered	if "name"="Clustered", consider "value" column	Semantic= "HasMember" ComputerSystem

Communication Relationships	[acm_comm_relshps]	Property Name
IPv4 address 1	ipv4_addr_1	usm-core2:SourceIPV4 Address
IPv6 address 1	ipv6_addr_1	usm-core2:SourceIPV6 Address
port 1	port_1	usm-core2:SourceTransport LayerPort
protocol	protocol	usm-core2:TransportLayer
relationship type	relshp_typ	Semantic: IsConnectedTo,HasAccess To, IsConnectedTo,HasAccess To
IPv4 address 2	ipv4_addr_2	usm-core2:TargetIPV4 Address
IPv6 address 2	ipv6_addr_2	usm-core2:TargetIPV6 Address
port 2	port_2	usm-core2:TargetTransport LayerPort

CTA Communication Relationships	[acm_ta_relshps]	Property Name
attribute name	attr_name	usm-core2: ExtensionNameValuePairs

CTA Communication Relationships	[acm_ta_relshps]	Property Name
attribute value	attr_value	usm-core2: ExtensionNameValuePairs

Relationship Traffic Summary	[acm_traffic_summary]	Property Name
start time	start_tm	usm-core2: ExtensionNameValuePairs
stop time	stop_tm	usm-core2: ExtensionNameValuePairs
packet count	packet_cnt	usm-core2: ExtensionNameValuePairs

Communication Relationships - Configuration	[acm_relshp]	Property Name
component uuid	comp_uuid	SourceMdrElementID
relationship type	relshp_typ	Context = HasAccessTo/ HasRequirementFor

Containment Relationships	[acm_srvr_inst/acm_comp]	Property Name
parent	acm_svc.svc_name, acm_srvr_inst.srvr_name, acm_bp.name	SourceMdrElementID
child	acm_srvr_inst.srvr_name, acm_bp.name	TargetMdrElementID
relationship type	relshp_typ	Semantic:IsComposedOf

Virtual Relationships	[acm_srvr_relshps]	Property Name
parent	parent_srvr_uuid	SourceMdrElementID
child	child_srvr_uuid	TargetMdrElementID
relationship type	relshp_typ	Context = IsManagerFor/IsHostFor

Storage Relationships	[acm_srvr_stor_rel]	Property Name
parent	lun_name	SourceMdrElementID
child	srvr_name	TargetMdrElementID
relationship type	relshp_typ	Semantic:IsConnectedTo

Storage Relationships	[acm_srvr_stor_rel]	Property Name
parent	disk_name	SourceMdrElementID
child	lun_name	TargetMdrElementID
relationship type	relshp_typ	Semantic:IsComposedOf

Cluster Mapping

The following tables show how the CCA connector maps CA Configuration Automation clusters to USM:

Network Details from Blueprint (Windows)	[acm_param]	Property Name
cluster name	if "name" = "Cluster Name", consider "value" column	GroupName

Cluster	[acm_cluster]	Property Name
cluster name	cluster_name	GroupName/PrimaryDns Name
cluster IPv4 address	ipv4_addr	PrimaryIPV4Address
cluster IPv6address	ipv6_addr	PrimaryIPV6Address

Computer System Mapping

The following tables show how the CCA connector maps CA Configuration Automation computer systems to USM:

Server	[acm_srvr]	Property Name
server_uuid	srvr_uuid	MdrElementID

business process	business_ process	usm-core2: BusinessRelevance
modification time	modification_tm	LastModTimeStamp
status	status	AdministrativeStatus (New, managed, unmanged)
launch-in-context URL	srvr_name	UrlParams

Server	**[acm_srvr_inst]**	**Property Name**
host name	srvr_name	PrimaryDnsName
manufacturer	manufacturer	Vendor
model	mdl	Model
serial number	serial_number	PhysSerialNumber
windows domain name	windows_domain_name	usm-core2:PrimaryDnsDomain
NIS domain name	nis_domain_name	usm-core2:PrimaryNisDomain
BIOS name	bios_name	usm-core2:BiosName
BIOS manufacturer	bios_manufacturer	usm-core2:BiosVendor
BIOS serial number	bios_serial_number	BiosSystemID
SNMP system name	snmp_sys_name	SysName
SNMP description	snmp_descr	Description
IPv4 address	ipv4_addr	PrimaryIPV4Address
OS name	os_name	usm-core2:OSName
OS version	os_ver	PrimaryOSVersion
mac address	mac_addr	PrimaryMacAddress
BIOS firmware version	bios_firmware_ver	usm-core2:BiosVersion
architecture	architecture	ComputerSystem ProcessorType
CPU speed	cpu_speed	ProcessorSpeedInGHz
number of CPUs	number_of_cpus	NumberOfCores
created by	created_by	CreationUserName
creation time	creation_tm	CreationTimestamp
OS family	os_family	PrimaryOSType
refresh time	rfrsh_tm	LastModTimestamp

Server	[acm_srvr_inst]	Property Name
discovered time	discvd_tm	usm-core2: DiscoveryTimestamp
IPv6 address	ipv6_addr	PrimaryIPV6Address
original host name	orig_host_name	NamedAliases

Network Details from Blueprint (all operating systems)	[acm_param]	Property Name
domain	if "name"="Domain", consider "value" column	usm-core2:PrimaryDns Domain
IPv6 address	if "name"="IPv6 Address", consider "value" column	PrimaryIPV6Address
netmask	if "name"="Netmask", consider "value" column	usm-core2:IPV4NetMask or IPV6NetMask
number of NICs	if "name"="Number of NICs", consider "value" column	usm-core2:NumberOf InterfaceCards
primary IP address	if "name"="Primary IP Address", consider "value" column	PrimaryIPV4Address

Network Details from Blueprint (windows)	[acm_param]	Property Name
number of HBAs	if "name"="Number of HBAs", consider "value" column	usm-core2: NumberOfHbas
primary HBA worldwide name	if "name"="Primary HBA Worldwide Name", consider "value" column	usm-core2: PrimaryWWName

Network Details from Blueprint (Linux)	[acm_param]	Property Name
number of HBAs	if "name"="Number of HBAs", consider "value" column	usm-core2: NumberOfHbas

Network Details from Blueprint (Linux)	[acm_param]	Property Name
primary HBA worldwide name	if "name"="Primary HBA Worldwide Name", consider "value" column	usm-core2: PrimaryWWName

Network Details from Blueprint (HP-UX)	[acm_param]	Property Name
number of HBAs	if "name"="Number of HBAs", consider "value" column	usm-core2: NumberOfHbas
primary HBA worldwide name	if "name"="Primary HBA Worldwide Name", consider "value" column	usm-core2: PrimaryWWName

Network Details from Blueprint (AIX)	[acm_param]	Property Name
number of HBAs	if "name"="Number of HBAs", consider "value" column	usm-core2: NumberOfHbas
primary HBA worldwide name	if "name"="Primary HBA Worldwide Name", consider "value" column	usm-core2: PrimaryWWName

Hardware Details from Blueprint (all operating systems)	[acm_param]	Property Name
BIOS-Firmware date	if "name"="BIOS-Firmware Date", consider "value" column	usm-core2:BiosDate
BIOS-Firmware version	if "name"="BIOS-Firmware version", consider "value" column	usm-core2:BiosVersion
CPU quantity	if "name"="CPU Quantity", consider "value" column	NumberOfCores
CPU speed	if "name"="cpu speed", consider "value" column	ProcessorSpeedInGHz
CPU type	if "name"="CPU Type", consider "value" column	ProcessorType

Hardware Details from Blueprint (all operating systems)	[acm_param]	Property Name
host name	if "name"="host name", consider "value" column	PrimaryDnsName
manufacturer	if "name"="manufacturer", consider "value" column	Vendor
model	if "name"="Model", consider "value" column	Model
patch level	if "name"="Patch Level", consider "value" column	usm-core2:OSPatchLevel
physical memory	if "name"="physical memory", consider "value" column	MemoryInGB
serial number	if "name"="serial number", consider "value" column	PhysSerialNumber
system GUID	if "name"="System GUID", consider "value" column	ComputerSystem BiosSystemID
total disk size	if "name"="Total Disk size", consider "value" column	StorageInGB

Storage Details from Blueprint (all operating systems)	[acm_param]	Property Name
number of logical drives	if "name"="Number of Logical Drives", consider "value" column	usm-core2:NumberOfDisk Partitions
number of physical drives	if "name"="Number of Physical Drives", consider "value" column	usm-core2:NumberOfPhysi cal Drives
total disk size	if "name"="Total Disk size", consider "value" column	StorageInGB

Memory	[acm_srvr_inst]	Property Name
memory total slots	mem_tot_slots	usm-core2:NumberOf MemorySlots

Server Fiber Channel Worldwide Node Names	acm_srvr_fc_wwnn	Property Name
Worldwide Node Name	wwnn	usm-core2:PrimaryWWName

File Mapping

The following tables show how the CCA connector maps CA Configuration Automation file systems to USM:

Storage Details from Blueprint (all operating systems)	[acm_cfg_param]	Property Name
size (per logical Drive)	if "name"="Size ", consider "value" column	usm-core2:MaxSizeInMB

Filesystems	[acm_file_sys]	Property Name
filesystem name	name	FilePathUrl,[FileType= "Volume"]
filesystem size	file_sys_size	usm-core2:MaxSizeInMB
filesystem mountlocation	mount_location	usm-core2:MountedFileSystem

Hypervisor Manager Mapping

The following tables show how the CCA connector maps CA Configuration Automation hypervisor managers to USM:

Virtualization	[acm_srvr_inst]	Property Name
containing server virtualization type	ve_host_typ	usm-core2:VirtualizationEnvironment

InterfaceCard Mapping

The following tables show how the CCA connector maps CA Configuration Automation storage processors to USM:

Storage Processor	acm_stor_processor	Property Name
Name	name	DeviceSysName
IP Address	ip_addr	DeviceIPV4Address
DNS name for the IP Address	dns_name	DeviceDnsName

Location Mapping

The following tables show how the CCA connector maps CA Configuration Automation locations to USM:

Server	[acm_srvr_inst]	Property Name
SNMP location	snmp_location	LocationName

Server	[acm_srvr]	Property Name
location	location	LocationName

Service	[acm_svc]	Property Name
location	location	LocationName

Media Drive Mapping

The following tables show how the CCA connector maps CA Configuration Automation media drives to USM:

Storage Details from BP(For all OS)	[acm_cfg_param]	Property Name
description (per Physical Drive)	if "name"="Description ", consider "value" column	Description

Storage Details from BP(For all OS)	[acm_cfg_param]	Property Name
size (per Physical Drive)	if "name"="Size ", consider "value" column	CapacityInMB

Physical Disks	[acm_physical_disk]	Property Name
physical disk index	disk_ix	ContainingIndex
physical disk name	disk_name	Label
physical disk size	disk_size	CapacityInMB
physical disk interface type	intf_typ	usm-core2: DriveInterfaceType
physical disk media type	media_typ	DriveType
physical disk model	mdl	Model

CD/DVD Drives	[acm_cd_dvd_drive]	Property Name
CD DVD drive description	descr	Description,[DriveType= "OpticalDrive-DVD"]
CD DVD drive dev id	device_id	usm-core2: OSDriveName
CD DVD drive media type	media_typ	TypeName/DriveType

Tape Drives	[acm_tape_drive]	Property Name
tape drive desc	descr	Description,[DriveType= "TapeDrive"]
tape drive dev type	device_typ	TypeName/DriveType
tape drive manufacturer	manufacturer	Vendor

Memory Mapping

The following table shows how the CCA connector maps CA Configuration Automation memory to USM:

Memory	[acm_srvr_inst]	Property Name
memory capacity	mem_cap	SizeInMB
memory type	mem_typ	MemoryType/Model
memory speed	mem_speed	usm-core2: SpeedInGHz
memory slots in use	mem_slots_in_use	usm-core2: SlotsInUse
physical memory	physical_mem	MemoryInGB

Operating System Mapping

The following tables show how the CCA connector maps CA Configuration Automation operating systems to USM:

Hardware Details from Blueprint (all operating systems)	[acm_param]	Property Name
virtual memory	if "name"="virtual memory", consider "value" column	usm-core2:VirtualMemory InGB

Operating System	[acm_srvr_inst]	Property Name
OS name	os_name	NamedAliases
OS detail	os_detail	Description
OS type	os_typ	OSType
OS version major	os_ver_major	MajorVersion
OS version minor	os_ver_minor	MinorVersion
OS version build	os_ver_build	BuildNumber
patch level	os_patch_level	usm-core2: OSPatchLevel
OS_ver	os_ver	Version
OS kernel	os_kernel	usm-core2: OSKernelVersion

Memory	[acm_srvr_inst]	Property Name
memory virtual	virt_mem	usm-core2:VirtualMemory InGB

Person Mapping

The following tables show how the CCA connector maps CA Configuration Automation users (persons) to USM:

Server	[acm_srvr_inst]	Property Name
SNMP contact	snmp_contact	UserName

Server	[acm_srvr]	Property Name
business owner	business_owner	UserName
IT owner	it_owner	UserName

Service	[acm_svc]	Property Name
business owner	business_owner	UserName
IT owner	it_owner	UserName

Port Mapping

The following tables show how the CCA connector maps CA Configuration Automation ports to USM:

Network Details from Blueprint (Windows)	[acm_cfg_param]	Property Name
description	if "name"="Description ", consider "value" column	Description
IP Address	if "name"="IP Address ", consider "value" column	PrimaryIPV4Address, PrimaryIPV6Address or OtherIPAddresses
physical address	if "name"="Physical address ", consider "value" column	PrimaryMacAddress or OtherMacAddresses

Network Details from Blueprint (Windows)	[acm_cfg_param]	Property Name
speed	if "name"="Speed", consider "value" column	NomSpeedInBitsPerSec

Network Interface Cards	[acm_srvr_nic]	Property Name
mac address	mac_addr	PrimaryMacAddress
interface index	net_adapter	IfIndex
interface speed	speed	NomSpeedInBitsPerSec
interface duplexity	duplex	usm-core2:IsFullDuplex
interface negotiation	aneg	usm-core2:DuplexIs Negotiated
DNS domain	dns_domain	usm-core2: PrimaryDnsDomain

Network Interface Cards	[acm_intf_ipv4_addr]	Property Name
IPv4 address	ipv4_addr	PrimaryIPV4Address

Network Interface Cards	[acm_intf_ipv6_addr]	Property Name
IPv6 address(es)	ipv6_addr	PrimaryIPV6Address or OtherIPAddresses

Processor Mapping

The following table shows how the CCA connector maps CA Configuration Automation processors to USM:

Processors	[acm_srvr_inst]	Property Name
processor name	processor_name	NamedAliases
processor architecture	processor_architecture	ProcessorType
processor description	processor_descr	Description
processor manufacturer	processor_manufacturer	Vendor
processor max clock speed	processor_max_clock_ speed	SpeedInGHZ

Processors	[acm_srvr_inst]	Property Name
processor l2 cache size	processor_l2_cache_size	usm-core2: L2CacheInMB
processor l2 cache speed	processor_l2_cache_speed	usm-core2: L2CacheSpeedInGHz
processor logical cnt	processor_logical_cnt	usm-core2: NumberOfCores

Provisioned Software Mapping

The following tables show how the CCA connector maps CA Configuration Automation provisioned software to USM:

Component	[acm_comp]	Property Name
component uuid	comp_uuid	MdrElementID
component version	comp_ver	Version
component qualifier	comp_qual	SoftwarePathUrl
refresh time	rfrsh_tm	usm-core2:LastRefreshTimestamp
creation time	creation_tm	CreationTimestamp
missing	missing	AdministrativeStatus = "Missing-InSubseqDiscover"

Component	[acm_bp]	Property Name
component Name	name	usm-core2: BlueprintName

Component	[acm_param]	Property Name
product name	if "name" = "Product name", consider "value" column	Product name
vendor	if "name"="Vendor", consider "value" column	Vendor
category	[acm_bp].cat	SoftwareCategories

Component	[acm_param]	Property Name
launch-in-context URL	[acm_srvr].srvr_name, [acm_bp].name, [acm_comp].comp_qual	UrlParams

Installed Applications	[acm_os_appl]	Property Name
application name	name	ProductName
application publisher	publisher	Vendor
application version	ver	Version
application arch	architecture	ProcessorEnvironments
application install date	install_date	usm-core2:InstallDate
application install location	install_location	SoftwarePathUrl
is patch?	is_patch	ReleaseType

Router Mapping

The following table shows how the CCA connector maps CA Configuration Automation routers to USM:

Server	[acm_srvr_inst]	Property Name
is router?	is_router	If set ComputerSystem is mapped as Router

Service Mapping

The following table shows how the CCA connector maps CA Configuration Automation services to USM:

Services	[acm_svc]	Property Name
service uuid	svc_uuid	MdrElementID
service name	svc_name	ServiceName
description	descr	Description
created by	created_by	CreationUserName
creation time	creation_tm	CreationTimestamp

Services	[acm_svc]	Property Name
business process	business_process	usm-core2:BusinessRelevance
modification time	modification_tm	LastModTimestamp
launch-in-context URL	svc_name	UrlParams
launch-in-context Visualization URL	svc_name	usm-core2:ExtensionNameValuePairs

StorageArray Mapping

The following tables show how the CCA connector maps CA Configuration Automation storage systems to USM:

Storage System	acm_stor	Property Name
Name	name	NamedAliases
serial number	serial_number	PhysSerialNumber
manufacturer	manufacturer	Vendor
Model/Type	mdl_typ	Model
Storage capacity	storage_cap	HardDriveCapacityInGB

Storage Fiber Channel Worldwide names	acm_stor_fc_wwn_inst	Property Name
Worldwide Node Name	wwnn	usm-core2:PrimaryWWName
Worldwide Port Name	wwpn	usm-core2:OtherWWNames

StorageVolume Mapping

The following tables show how the CCA connector maps CA Configuration Automation storage Logical Unit Numbers to USM:

Storage LUN	acm_stor_lun_ex	Property Name
Network Address Authority	net_addr_authority	PhysSerialNumber

Storage LUN	acm_stor_lun_ex	Property Name
Storage capacity	storage_cap	PhysSerialNumber
Name	name	Label

Disk Partition Mapping

The following table shows how the CCA connector maps CA Configuration Automation disk partitions to USM:

Logical Partitions	[acm_disk_partition]	Property Name
logical partition name	partition_name	usm-core2:NamedAliases
drive designation	drive_designation	usm-core2:OSDriveName
logical partition index	partition_ix	usm-core2:ContainingIndex
logical partition filesystem	filesystem	usm-core2:PartitionType
logical partition is bootable	is_boot_partition	usm-core2:IsBootable
logical partition is primary	is_pri_partition	usm-core2:IsPrimary
logical partition size	partition_size	usm-core2:CapacityInMB

IPConfig Mapping

The following tables show how the CCA connector maps CA Configuration Automation IPConfig values to USM:

Server	[acm_srvr_inst]	Property Name
SNMP ip forwarding	snmp_ip_fwding	usm-core2:DoesIPForwarding
domain server IPv4 address	domain_srvr_ipv4_addr	usm-core2:PrimaryDnsServer IPV4Address
domain server IPv6 address	domain_srvr_ipv6_addr	usm-core2:PrimaryDnsServer IPV6Address
domain server name	domain_server_name	PrimaryDnsServer

Network Details from Blueprint (Windows)	[acm_cfg_param]	Property Name
broadcast address	if "name"="Broadcast Address ", consider "value" column	BroadcastIPV4Address or BroadcastIPV6Address
DNS	if "name"="DNS ", consider "value" column	DnsServerAddresses (IPv4 or IPv6)
gateway	if "name"="Gateway ", consider "value" column	GatewayIPV4Address or GatewayIPV6Address
primary wins	if "name"="Primary Wins", consider "value" column	PrimaryWins IPV4Address or Primary WinsIPV6Address
secondary wins	if "name"="Secondary Wins ", consider "value" column	OtherWinsAddresses
subnet mask	if "name"="Subnet Mask ", consider "value" column	usm-core2: IPV4NetMask or IPV6NetMask

Network Interface Cards	[acm_srvr_nic]	Property Name
default IPv4 gateway	default_gateway_ipv4_addr	GatewayIPV4Address
IPv4 DHCP server	dhcp_srvr_ipv4_addr	DhcpServerIPV4Address
default IPv6 gateway	default_gateway_ipv6_addr	GatewayIPV6Address IPv6
IPv6 DHCP server	dhcp_srvr_ipv6_addr	DhcpServerIPV6Address

Network Interface Cards	[acm_intf_ipv4_addr]	Property Name
subnet mask	subnet_mask	usm-core2:IPConfig usm-core2: IPV4NetMask or IPV6NetMask
IPv4 subnet	ipv4 subnet	usm-core2: IPV4Subnet

Network Interface Cards	[acm_intf_ipv6_addr]	Property Name
IPv6 subnet(s)	ipv6_subnet	usm-core2: IPV6Subnet or OtherIPSubnets (IPV4 or IPV6)

Network Interface Cards	[acm_intf_dns_srvr]	Property Name
DNS server IPv4(s)	srvr_ipv4_addr	PrimaryDnsServer IPV4Address and OtherDnsServer Addresses
DNS server IPv6(s)	srvr_ipv6_addr	PrimaryDnsServer IPV6Address and OtherDnsServer Addresses

Network Interface Cards	[acm_intf_wins_srvr]	Property Name
wins server IPv4(s)	srvr_ipv4_addr	PrimaryWinsServer IPV4Address and OtherWinsServer Addresses
wins server IPv6(s)	srvr_ipv6_addr	PrimaryWinsServer IPV6Address and OtherWinsServer Addresses

Storage iSCSI Initiator	acm_stor_iscsi_initr_inst	Property Name
Initiator Identifier	initr_id	Extension Name Value Pairs
Initiator IP Address	[acm_stor_iscsi_initr_ip].initr_ip_addr	Static IPV4 Address

Virtualization Manager Mapping

The following table shows how the CCA connector maps CA Configuration Automation virtualization managers to USM:

Server	[acm_srvr_inst]	Property Name
manager virtualization type	virt_mgmt_typ	ProcessDistinguishingID/ProductName

Virtual System Mapping

The following table shows how the CCA connector maps CA Configuration Automation virtual systems to USM:

Virtualization	[acm_srvr_inst]	Property Name
guest server virtualization type	ve_guest_typ	usm-core2:VirtualizationEnvironment
guest server logical name	ve_guest_logical_name	ComputerName
guest server startup mode	ve_guest_startup	usm-core2:IsAutomaticallyStarted

Binary Relationship Scope Mapping

The following table shows how the CCA connector maps CA Configuration Automation relationships to USM:

Parent	Relationship Type	Child Type	Scope	Comments
service	HasMember	ComputerSystem	Service Scope	
	HasDetail	ComplianceStatus	Service Scope	
	HasContact	Person	Service Scope	
	IsResidentOf	Location	Service Scope	
computer system	IsManagedBy	ComputerSystem	No Scope	Virtualization Relationship
	HasDetail	ComplianceStatus	Service Scope if Compliance status is Service level otherwise ComputerSystem Scope	
	HasContact	Person	No Scope	
	HasDetail	IPConfig	ComputerSystem Scope	
	IsHostFor	BackgroundProcess	ComputerSystem Scope	
	IsConnectedTo	BackgroundProcess	No Scope	Communication relationship
	IsConnectedTo	ComputerSystem	No Scope	Communication relationship
	IsResidentOf	ComputerSystem	No Scope	

Parent	Relationship Type	Child Type	Scope	Comments
provisioned software	HasAccessTo	ComputerSystem	Service Scope if Component discovered using a Service otherwise No Scope	Configuration relationship 'uses'
	HasRequirementFor	ComputerSystem	Service Scope if Component discovered using a Service otherwise No Scope	Configuration relationship 'communicates with'
	HasAccessTo	BackgroundProcess	Service Scope if Component discovered using a Service otherwise No Scope	Configuration relationship 'uses'
	HasRequirementFor	BackgroundProcess	Service Scope if Component discovered using a Service otherwise No Scope	Configuration relationship 'communicates with'
	IsAffectedBy	ProvisionedSoftware	Service Scope if Component discovered using a Service otherwise No Scope	Nested Component
	IsHostedBy	ComputerSystem	Service Scope if Component discovered using a Service otherwise No Scope	
port	HasDetail	IPConfig	ComputerSystem Scope	
	IsPartOf	ComputerSystem	ComputerSystem Scope	
cluster	HasMember	ComputerSystem	No Scope	
background process	IsConnectedTo	BackgroundProcess	No Scope	Communication relationship
	IsConnectedTo	ComputerSystem	No Scope	Communication relationship
mediadrive	IsPartOf	ComputerSystem	ComputerSystem Scope	
operatingsystem	IsAffectedBy	ComputerSystem	ComputerSystem Scope	
virtualsystem	IsHostedBy	ComputerSystem	ComputerSystem Scope	Virtualization Relationship
memory	IsPartOf	ComputerSystem	ComputerSystem Scope	
processor	IsPartOf	ComputerSystem	ComputerSystem Scope	

Custom Mappings

The CCA connector uses connector policy to map product data to adhere to the USM schema. Each connector includes default policy in an XML file in the following directory:

CATALYST_HOME\resources\Core\Catalogpolicy

CATALYST_HOME

Defines the CA Catalyst installation directory.

The default policy establishes all mappings included in this section. You can customize connector policy to add mappings based on product changes or customizations, or to support entities not supported by the default policy. For example, if you have added classes to the product that the default policy does not support, you can edit the connector policy to map these classes to USM types.

Note: For more information about writing and customizing connector policy, see the *CA Catalyst Connector Guide*.

View Catalyst Jobs

The Catalyst Integration tab displays the catalyst jobs, and the Export Summary tab lists the successfully exported or failed CIs. You can view the export summary details of the executed catalyst jobs from the Management or Administrator link.

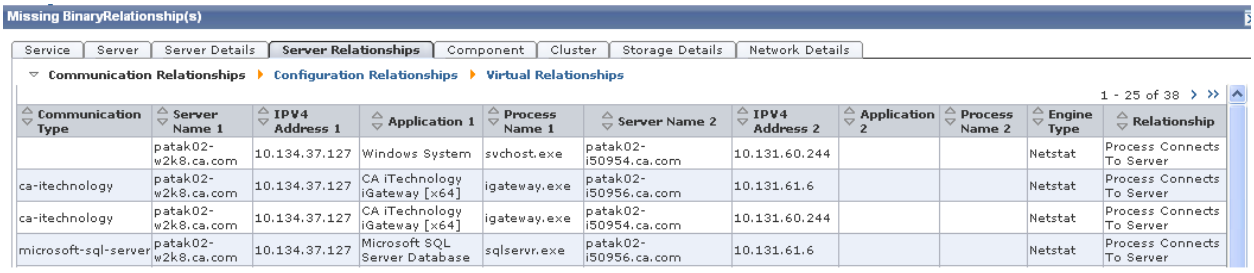
Follow these steps:

1. From the Management link:
   a. Click Management link, Management Profiles.
   b. Select a management profile, Enable Integration check box to view the Catalyst Export summary tab.
   c. Select the Catalyst Export summary tab.
2. From the Administrator link:
   a. Click the Administrator link, Catalyst Integration tab, and then the Jobs link.
   b. Select a job from the available catalyst jobs executed catalyst.
   c. Select the Export Summary tab.

   The Profile details (using the Management profile) or the Job details (using the Administrator link) page lists the total count of the exported CIs, successfully exported CIs, and failed CIs as part of the Catalyst job.
3. Click a link in listed in the Catalyst CI count (successful) column of a USM type to view the USM web view.

   Note: The USM web view is successfully launched only when you configure the catalyst.server.name and catalyst.server.port properties.
4. Click a link in the Catalyst CI count (failed) column of a particular USM type.

   The source information of all the CIs is listed failed during the export to CA Catalyst when the job was run as shown in the following illustration:



Communication Type	Server Name 1	IPv4 Address 1	Application 1	Process Name 1	Server Name 2	IPv4 Address 2	Application 2	Process Name 2	Engine Type	Relationship
	patak02-w2k8.ca.com	10.134.37.127	Windows System	svchost.exe	patak02-i50954.ca.com	10.131.60.244			Netstat	Process Connects To Server
ca-itechnology	patak02-w2k8.ca.com	10.134.37.127	CA iTechnology iGateway [x64]	igateway.exe	patak02-i50956.ca.com	10.131.61.6			Netstat	Process Connects To Server
ca-itechnology	patak02-w2k8.ca.com	10.134.37.127	CA iTechnology iGateway [x64]	igateway.exe	patak02-i50954.ca.com	10.131.60.244			Netstat	Process Connects To Server
microsoft-sql-server	patak02-w2k8.ca.com	10.134.37.127	Microsoft SQL Server Database	sqlservr.exe	patak02-i50956.ca.com	10.131.61.6			Netstat	Process Connects To Server

Note: For more information about the failed CIs and how to resolve the errors, see the Troubleshooting (see page 53) section.

Chapter 5: Uninstall the CA Configuration Automation Connector

You can uninstall the CCA connector when it is no longer required.

Follow these steps:

1. (Windows) Select Start, Programs, CA, Catalyst, Uninstall_CA Configuration Automation Connector for <CCA Server Host>_<CCA Server Port> on the connector system.

   (Linux) Run Uninstall CA Configuration Automation Connector for <CCA Server Host>_<CCA Server Port> from the <CCA Connector installation directory>/ Uninstall_CA Configuration Automation Connector for <CCA Server Host>_<CCA Server Port>

   The Uninstall CCA Connector dialog opens.

2. Select the Restart Container checkbox to restart the container service and click Uninstall.

   The CCA Connector is uninstalled. The Uninstall Complete page lists the uninstallation results, including errors that may have occurred.

   Note: If the Restart Container checkbox is not selected, we recommend you to restart the container after the CCA Connector is uninstalled.

3. Verify that the uninstall program removed the CCA connector OSGi bundle com.ca.cca.catalyst.connector_<installed version>.jar from the <Catalyst Home>\ container\system\com\ca\catalyst\3.0.0\bundles directory. Manually delete it if it is still there.

   Note: Restart the CA Catalyst Container service on the CCA connector node, if the Restart Container checkbox was not enabled in Step 2.

4. Click done. An uninstall log file is created. Review the log files if the uninstallation screen shows any errors.

   ```
   %CATALYST_HOME%\CCA Connector Uninstall folder \ Logs\
   CA_Configuration_Automation_Connector_UnInstall%timestamp%.log
   ```

   If you choose to uninstall the Catalyst container, perform the following tasks:

   - Invoke the CA Catalyst Container Uninstall from <Catalyst Home>\Uninstall_CA Catalyst Container\Uninstall CA Catalyst Container.
   - Delete the complete <connector node name> directory from the Catalyst Registry UI using the Actions menu. The Catalyst Registry UI is at the path /topology/physical/ location.

Chapter 6: Troubleshooting

This troubleshooting section describes common issues that you encounter while working on the CA Catalyst Container. It also suggests actions that you can take to resolve these issues. For more information about diagnostics and troubleshooting the CA Catalyst container, see the *CA Catalyst Implementation Guide.*

This section contains the following topics:

CA Catalyst Container not Compatible with Windows 2012 Server

The CA Catalyst Container is not compatible with Windows 2012. To install the CA Catalyst Container on Windows 2012, use the Windows 7 or Windows 2008 compatibility mode, and point it to CA Catalyst Server installed on a supported version of Windows.

To make the CA Catalyst Container compatible with Windows 2012 server, follow these steps:

1. Right click the installer exe file, and then select Properties.
2. Click the Compatibility tab, and then click the **Run this program in Compatibility mode** option.
3. Select the Operating system that the installer supports. For example, Windows 7.
4. Select **Run this program as an administrator** option to get the administrator privileges, and then click Ok.
5. Run the catalyst_install.bat, or setup.exe file through CA Setup Launcher.

   The CA Catalyst installation or setup now runs as the system is in the compatibility mode.

Graphical Mode Installation Fails on RHEL6 Machines

Symptom:

When I install CCA Connector on the RHEL6 machine the following error occurs:

```
Graphical installers are not supported by the vm
```

Solution:

This error may occur due to the following reasons:

- The file /usr/lib/libXtst.so is not installed on the system.
- The $DISPLAY environment variable is not properly set.
- The necessary X Windows libraries for running the GUI installer are not available.

To install the CCA Connector on the RHEL machine, do any *one* of the following:

- Install the libXtst.so library that is available in libXtst-1.2.1-2.el6.i686.rpm package.
- Set the $DISPLAY environment variable to a valid display.

Note: If you are logged in to the 64-bit RHEL machine, run the following command to install the missing 32-bit dependencies:

```
yum install xulrunner.i686
```

Verify the CA Configuration Automation Connector Installation

Symptom:

I want to verify that the CA Configuration Automation Connector is installed properly.

Solution:

Verify the Catalyst_CCAConnector_InstallDebug.log or CatalystInstallDebug.log files for any errors. For Windows, the log files are located in the %TEMP% directory. For LINUX, the log files are located in / directory. If either of the log file has errors, contact CA Technologies Technical Support for further assistance. If there are no errors in either of the log file, complete the following steps:

1. Log in to Registry UI using the following URL:

   https://registryserver:port/registry/carbon/admin/login.jsp

2. Expand the folder in the Browse section to locate the following directory:

   \topology\physical\<CA Configuration Automation Connector NODE>

3. Verify that the connector-modules.xml and startup.properties files are present in the directory.

   If either of the file is not available in the directory, then the installation of the CA Configuration Automation Connector has failed.

4. Uninstall, and reinstall the CA Configuration Automation Connector.

Verify the CA Configuration Automation Connector Status

Symptom:

I want to verify that the CA Configuration Automation Connector starts properly.

Solution:

Follow these steps:

1. Log in to CA Catalyst Administration UI using the following URL:

   http://<CA Catalyst-Server>:port/adminui

2. Click to expand the CA Configuration Automation Container node in the Catalyst Node panel.

   The available CA Configuration Automation Connectors, and status of the CA Configuration Automation Connector is displayed. If the status is marked as RUNNING, then the CA Configuration Automation Connector has started properly.

   Note: If the CA Catalyst Connector has just started, wait for the Connector nodes to move to the Running state before you perform any operations.

Errors with Exception Trace in ccaConnector.log File

The ccaConnector.log file shows errors with exception traces. Review the exceptions that appear in the ccaConnector.log file, and act as per that to resolve the exception. The log file in Windows is at the following location:

```
C:\Program Files (x86)\CA\Catalyst\<ContainerID>\container\data\log
```

The log file in LINUX is at the following location:

```
/opt/CA/Catalyst/<ContainerID>/container/data/log
```

The possible exceptions that may appear in the ccaConnector.log file are as follows:

- CA Configuration Automation Server Connection Exception
- Database Exception
- JVM Port Binding Exception

CA Configuration Automation Server Connection Exception

Symptom:

When I start the CA Catalyst Connector application, the CA Configuration Automation Server Connection exception appears in the ccaConnector.log file.

Solution:

Follow these steps:

1. Verify that the CCA Server is online, and that you can contact the server from the CA Configuration Automation server.
2. Verify that no firewall is blocking access to the necessary CCA Server port from the CA Configuration Automation Connector server.

   Important! If you restart the CCA Server, restart the CA Catalyst container service on which the CA Configuration Automation Connector is hosted.
3. If you changed the CCA Server password after the connector was installed, use the Registry UI to update the password parameter in CCAConnector_*<CCA Server Host>*_*<CCA Server Port>*.xml as follows:

   a. Open the following directory on the command prompt:

      ```
      %CATALYST_HOME%\tools\encrypt
      ```

      Note: Verify that the directory includes the java.exe.

   b. Run the encrypter.bat <New Password> command.

   c. Copy the encrypted string output from the encrypter.bat <New Password> command.

d. Use the following URL to log in to the Registry UI:

   `https://registryserver:port/registry/carbon/admin/login.jsp`

e. Browse to the following directory:

   `\topology\physical\<CCA Connector Server>\modules\configuration\CCAConnector_<CCA Server Host>_<CCA Server Port>.xml`

f. In the Content panel, click Edit as Text.

g. Replace the password property value with the value you copied from the encrypter.bat command.

h. Click Save Content.

i. Verify the solution:

   - Restart the CA Catalyst Container service.
   - Start the CA Catalyst Connector application.
   - Verify that the CA Configuration Automation Server Connection exception does not appear in the ccaConnector.log file.

Database Exception

Symptom:

When I launch the CA Catalyst Connector application, the Database exception appears in the ccaConnector.log file.

Solution:

Follow these steps:

1. Verify that the database server is online, and you can ping the server from the CA Configuration Automation Connector server.
2. Verify that there is no firewall that is blocking the access to the necessary database port from the CA Configuration Automation Connector server.
3. Update the db.password parameter in the CCAConnector.xml file from the Registry UI. Update the password parameter if you have changed the password since the installation of the CA Configuration Automation Connector. Follow these steps to change the password:
   a. Open the following directory on the command prompt:

      %CATALYST_HOME%\tools\encrypt

      Note: Ensure that the java.exe is present in the path.
   b. Run the encrypter.bat <New Password> file.
   c. Copy the encrypted string output from the encrypter.bat <New Password> command.
   d. Log in to Registry UI using the following URL:

      https://registryserver:port/registry/carbon/admin/login.jsp
   e. Browse to the following directory:

      \topology\physical\<CCA Connector Server>\modules\configuration\CCAConnector_<CCA Server Host>_<CCA Server Port>.xml
   f. Click Edit as Text in the Content panel.
   g. Replace the db.password property value with the value copied from the encryptor utility.
   h. Click Save Content.
   i. Restart the CA Catalyst Container service, and reverify.

JVM Port Binding Exception

Symptom:

When I launch the CA Catalyst Connector application, the JVM Port Binding exception appears in the ccaConnector.log file.

Solution:

Follow these steps:

1. Verify if any process is using port that is used for notification_listen_port parameter during the CA Configuration Automation Connector installation.
2. Update the parameter in the CCAConnector.xml file from the Registry UI if a port conflict exists after the CA Configuration Automation Connector installation.
   a. Log in to Registry UI using the following URL:

      https://registryserver:port/registry/carbon/admin/login.jsp

   b. Expand the folder in the Browse section to locate the following directory:

      \topology\physical\<CCA Connector Server>\modules\configuration\CCAConnector_<CCA Server Host>_<CCA Server Port>.xml. Click Edit as Text.

   c. Replace the value for notification_listen_port with the value that does not conflict with other ports on the CA Configuration Automation server.

   d. Click Save Content.

   e. Restart the CA Catalyst Container service.

Verify that the Data Sent to CA Catalyst Server From CA Configuration Automation Connector is Successful

Symptom:

I want to verify that the Data Sent to CA Catalyst Server from the CA Configuration Automation Connector is Successful.

Solution:

Follow these steps:

1. Open the Catalyst USM web view using the following URL:

   http://catalystserver:8080/ca-rest/browse/type?mdr=all

2. Select CA Configuration Automation from Data Source, and check if the CIs are listed on the page.

   If the CIs are not listed in the CA Configuration Automation Data Source, ensure the following actions:

   a. Ensure that you restart the Catalyst Container Service of the CA Configuration Automation Connector when you start the CCA Server.

   b. Ensure that the time settings on the CA Catalyst server and the CA Configuration Automation Connector server nodes are same to avoid the connectivity issues.

   c. Ensure that the problem.log file does not have errors in the CCA Catalyst Container and CA Catalyst Server Container. The log file in Windows is at the following location:

      `C:\Program Files\CA\Catalyst\<ContainerID>\container\data\log`

      The Log file in LINUX is at the following location:

      `/opt/CA/Catalyst/<ContainerID>/container/data/log`

      If errors exist, perform Step f and verify the export.

   d. If you used the Catalyst Job to import the data from CA Configuration Automation Connector, verify that the job is complete.

      This verification is applicable for the Management Profile based integration. If the job is complete, the Log tab in the CCA Server displays the Job started and Job finished messages.

   e. Check either the Export Summary Tab on the Catalyst Job, or Management profile Job that is used to export the CIs. If the Catalyst CI Count (Failed) column is non-zero, then click the link to view the failed CI details.

   f. Identify the root cause for the export failure.

      - Match the Check Sum Table with the projections of the CA Configuration Automation Connector.

- Clean the database when:
  - The CIs are not exported correctly from CA Configuration Automation to CA Catalyst.
  - The checksum entries do not match any CIs in CA Catalyst.

Match the Check Sum Table

Match the Check Sum table with the projections of the CA Configuration Automation Connector that the CA Catalyst Persistence Store maintains for every CI that is sent to the CA Catalyst server.

Note: The CA Configuration Automation Connector maintains the Check Sum table for every CI that is sent to the CA Catalyst server.

Follow these steps:

1. Run the following queries from the CA Configuration Automation database to find the failed CIs.

   Note: Adapt the queries with the corresponding database names, passwords, and the other details. The database may also need to configure cross machine database access for CA Configuration Automation and CA Catalyst databases.

   - Use the following query in the SQL database to identify the failed CIs:

     ```
     select ci_id = CASE CHARINDEX( '|', ci_typ)
     WHEN '0' THEN ci_id
              ELSE (ci_id + substring( ci_typ, CHARINDEX( '|', ci_typ) + 1,
     len(ci_typ)))
     END collate sql_latin1_general_cp1_ci_as from acm_catlst_ci_cksum
     EXCEPT
     select c_mdrelementid from [catalystdb].[dbo].t_ci_detail where
     c_mdrproduct = 'CA:00033'
     ```

   - Use the following query in the ORACLE database to identify the failed CIs:

     ```
     CREATE PUBLIC DATABASE LINK catalystdb CONNECT TO catalystadmin IDENTIFIED
     BY <Password> USING '//<DBServer>/<DB SID>';
     select CASE instr( ci_typ, '|')
     HEN 0 THEN cast( ci_id as NVARCHAR2(36) )
           ELSE concat( cast( ci_id as NVARCHAR2(36) ) , substr( ci_typ, instr(
     ci_typ, '|') + 1, length(ci_typ)))
     END as c_mdrelementid
     from acm_catlst_ci_cksum
     minus
     select c_mdrelementid from t_ci_detail@catalystdb where c_mdrproduct =
     'CA:00033'
     ```

2. Verify the invalidCIs.log file in the following directory for the missing IDs that are identified in Step 1.

   %CATALYST_HOME%\<ContainerID>\container\data\log

   Ensure that the source data for all the attributes is in expected format per USM, else contact CA Technologies Technical Support for further assistance. If the identified ID is a BinaryRelationship, ensure that the source and target CIs are exported as a part of the job successfully.

To verify whether the source and target CIs are exported, search for SourceMDRElementID and TargetMDRElementID for the relationship CI in the USM web view.

Clean the Database When CI's are not Correctly Exported from CCA to Catalyst

Purge data from the CA Catalyst database to clean the database when export of large CIs failed during the export process.

Follow these steps:

1. Stop the CA Catalyst Container CatalystConnector service on all nodes in your environment.
2. (Optional) Stop the CA Catalyst Administrator and CA Catalyst Registry services.
3. Run the following SQL command sequence from an appropriate tool such as Microsoft SQL Server Management Studio on CA Catalyst database:

   ```
   delete from t_ci_detail
   delete from t_ci_timestamp
   delete from t_notebooks_timestamp
   delete from t_tags
   delete from t_rest_access
   delete from t_connector_init_status
   ```

   More tables exist corresponding to the various USM types. The data from the tables are deleted when the data from t_ci_detail is deleted.
4. Run the following SQL command sequence. Run the command from an appropriate tool such as Microsoft SQL Server Management Studio to clean the CA Configuration Automation tables for republishing the data:

   ```
   delete from acm_catlst_ci_cksum
   delete from acm_catlst_ci_summary
   ```
5. Delete the following files from each Catalyst node. Assuming that the CA_CATALYST_HOME is C:\Program Files\CA\Catalyst.

   ```
   C:\Program Files\CA\Catalyst\<ContainerID>\container\data
   C:\Program Files\CA\Catalyst\<ContainerID>\container\CatalystDataStore
   C:\Program Files\CA\Catalyst\<ContainerID>\nls-store
   C:\Program Files\CA\Catalyst\<ContainerID>\solr\data
   ```

   The files serve different purposes. For example, the CatalystDataStore folder is used at connector startup as part of delta processing. It determines the changes that occurred while the connector is down.

   Note: Modify the file names if the CA_CATALYST_HOME points to a different directory on the server.
6. (Optional) Start the CA Catalyst Administrator and CA Catalyst Registry Services if the service stopped earlier.

7. Start the CA Catalyst Container CatalystConnector service on all nodes in your environment.
8. Rerun the Catalyst jobs or management profiles to export data, and then verify that they were exported successfully.

Index

C

D

I

P

T

U