



CISCO CONFIDENTIAL

## CHAPTER 9

# Integrating Applications with CMIC

---

Cisco Management Integration Center (CMIC) is a registry or repository for management services, enabling Cisco Management Servers to integrate with outside-world applications. CMIC acts as a service lookup for any application trying to find a particular service or set of services provided by other applications on in a network, thereby facilitating interaction amongst these application. CMIC takes care of configuring the server to integrate with applications and services residing outside the box.

CMIC Registry replaces the CMF Desktop XML Registry. It allows you to:

- Register services
- Unregister services
- Lookup services

The following topics describe how to use CMIC with your application:

- [Understanding CMIC](#)
- [Using CMIC Services](#)

For basic information on CMIC, see the “[About the Cisco Management Integration Center \(CMIC\) Component](#)” section on page 6-6.

For more information about CMIC, see:

- The *CMF Product Requirement Document CMF 3.0 PRD, EDCS-205209*
- The *CTM Functional Specification, EDCS-124878*
- The *PSU Functional Specification, EDCS-110450*
- The *CMIC Software Unit Design Specification, EDCS-137328*.

## Understanding CMIC

The CMIC registry provides a Java API for registration, unregistration, and lookup of management services. These APIs are exposed via the Common Services Transport Mechanism (for details, see [Chapter 31, “Using the Common Services Transport Mechanism”](#)) so that they can be accessed over a network via North Bound (NB) APIs. All NB APIs take the extra parameter `UserCredentials`, containing user information, so accesses can be authenticated.

CMIC makes use of an XML template, called the Management Service Template (MST), to define in a coherent, standardized way each management service supplied by an application or tool. Each MST defines the attributes of the service that are relevant to other applications and services, such as the service name, description, how to invoke the service, etc. The only difference between instances of the same service is the host, port, and protocol of the instance where the service resides. The template collects all

## CISCO CONFIDENTIAL

the common information, and the information that changes based on the instance is given during the CMIC service registration call. The MST file name serves as the service's Uniform Resource Name (URN); this URN is referred to in the API call for any subsequent registrations or unregistrations of the service.

The CMIC registry provides a user interface to register the following different kinds of application services:

- Third-party applications or custom tools. The UI guides the user through a series of steps that create a launch point for the application or tool from the Cisco Works Home page.
- Cisco products with pre-defined, certified MSTs. The user can select a template and specify the server on which the application is installed.
- Imported registrations from another server. For this kind of service, the UI first prompts the user for the name of the server hosting the service, along with the server's SSL port number. Once the user selects a server, the UI displays a list of applications registered with that server, and the user can then select applications that need to be registered with the local server. Imported applications having the requisite integration tags for the CiscoWorks Home page will be displayed on the local CiscoWorks Server.
- All CiscoWorks applications residing in the same server as CMIC can register during their installation by making an API call to CMIC registry. Applications not local to CW server (residing in different server) can register themselves either through the UI or using the NB bound register API exposed through CSTM.

## Using CMIC Services

The following topics describe how to use CMIC services with applications:

- [Registering Applications](#)
- [Querying an Application](#)
- [Calling an Application](#)
- [Integrating CMIC with CWHP, Device Center, and Setup Center](#)
- [About the CMIC APIs](#)
- [About the Management Service Template](#)
- [Component Interaction](#)
- [About CMIC Registry Dependencies](#)
- [Sample MST File](#)

## Registering Applications

Registering applications is the foremost pre-requisite to find any management service. All management services, which intend to be called by other services need to first register with CMIC.

All services willing to register need to define a Management Service Template. The template is XML based and has information about various parameters of a service like host, port, protocol, where the service resides, URL to invoke the service etc.

The users create an MST. Once the MST is created, the user is provided with a URN that identifies the MST. The name of the MST file is URN.xml

## CISCO CONFIDENTIAL

This URN is used by respective services in register API along with the host, port & protocol of the host on which the service resides. CMIC registry stores all such registered services in a repository.

CMIC registry stores all such registered services in records.db repository available at `NMSROOT/objects/data/cmfcmic/registry`.

You can unregister or register applications to records.db file using command line options. This section contains the following subsections:

- [Unregistering Applications Through Command Line](#)
- [Registering Applications Through Command Line](#)

### Unregistering Applications Through Command Line

To unregister an application from CMIC, enter the following command:

```
$NMSROOT\MDC\JRE\bin\java.exe -cp
$NMSROOT\MDC\tomcat\webapps\cwhp\WEB-INF\lib\cwhp.jar;
$NMSROOT\lib\classpath\cmic.jar;
$NMSROOT\MDC\tomcat\shared\lib\xerces.jar;
$NMSROOT\objects\log4j\1.1.3\log4j.jar;
$NMSROOT\MDC\tomcat\shared\lib\MICE.jar;
$NMSROOT\MDC\tomcat\shared\lib\NATIVE.jar;
$NMSROOT\lib\classpath;com.cisco.nm.cwcs.cwhp.applications.remove URN port protocol
hostname
```

where

- *NMSROOT* is the directory in which your product is installed
- *URN* is the name of the file representing the application without the .xml extension under mst-templates directory (`$NMSROOT\objects\data\cmfcmic\mst-templates`)
- *port* is the port number used for registering the applications
- *protocol* is the protocol used for registering the application
- *hostname* is the name of host on which the service resides. If the host name is not specified, the command line utility will consider the local host name as the default value. Verify the contents of the registered templates directory (`$NMSROOT\objects\data\cmfcmic\`)

#### Example

To remove the Device Troubleshooting link from CiscoWorks Home page, enter the following command:

```
$NMSROOT\MDC\JRE\bin\java.exe -cp
$NMSROOT\MDC\tomcat\webapps\cwhp\WEB-INF\lib\cwhp.jar;
$NMSROOT\lib\classpath\cmic.jar;$NMSROOT\MDC\tomcat\shared\lib\xerces.jar;
$NMSROOT\objects\log4j\1.1.3\log4j.jar;
$NMSROOT\MDC\tomcat\shared\lib\MICE.jar;
$NMSROOT\MDC\tomcat\shared\lib\NATIVE.jar;
$NMSROOT\lib\classpath;com.cisco.nm.cwcs.cwhp.applications.devicecenter.1.0 1741 http afspcwc001
```

### Registering Applications Through Command Line

To register an application with CMIC, enter the following command:

```
$NMSROOT\MDC\JRE\bin\java.exe -cp
$NMSROOT\MDC\tomcat\webapps\cwhp\WEB-INF\lib\cwhp.jar;
$NMSROOT\lib\classpath\cmic.jar;
$NMSROOT\MDC\tomcat\shared\lib\xerces.jar;
```

## CISCO CONFIDENTIAL

```
$NMSROOT\objects\log4j\1.1.3\log4j.jar;  
$NMSROOT\MDC\tomcat\shared\lib\MICE.jar;  
$NMSROOT\MDC\tomcat\shared\lib\NATIVE.jar;  
$NMSROOT\lib\classpath;com.cisco.nm.cwcs.cwhp.applications.add URN port protocol hostname
```

where

- *NMSROOT* is the directory in which your product is installed
- *URN* is the name of the file representing the application without the .xml extension under mst-templates directory (*\$NMSROOT\objects\data\cmf\cmic\mst-templates*)
- *port* is the port number used for registering the applications
- *protocol* is the protocol used for registering the application
- *hostname* is the name of host on which the service resides. If the host name is not specified, the command line utility will consider the local host name as the default value. Verify the contents of the registered templates directory (*\$NMSROOT\objects\data\cmf\cmic\*)

## Querying an Application

A query is made to CMIC to find a management service from the registered management services. This query is based on certain search parameters

To find a particular or set of services the “query” API is used. CMIC supports query on various parameters like Application Name, Vendor name, Application Category (Fault, Configuration, Accounting, Performance, Security), Task Name etc. A query is constructed by the service requester using one or more of the parameters and sent to CMIC registry. The query can be an “AND” or “OR” on the parameters chosen for query.

## Calling an Application

The query returns matched services to service requestor, the query results are processed and call to respective services are made.

CMIC registry returns a list of services matching the query to the requester. The Service requestor can process the result and launch the returned services or provide links to these services for the user to take appropriate action.

## Integrating CMIC with CWHP, Device Center, and Setup Center

CMIC MST files contain all the information needed to manage and display the tasks in the Cisco Works Home Page Application Panel.

CWHP uses CMIC to register and query installed applications, Cisco Works Resources, Cisco.com resources, Cisco Works Applications installed on other servers, 3rd party applications, Custom Tools. CMIC registry has data about all registered applications in the users network, a snap shot of registry would give CWHP all the information it needs to provide launch points to various applications with the help of queries.

To integrate with CWHP, Device center, and Setup Center, you need to create a CMIC MST template and tag URLs appropriately (with integration tags) as defined by CWHP and DC, then register the template with CMIC.

## CISCO CONFIDENTIAL

You need not create different templates for CWHP, Device Center and Setup Center. All tasks can be a part of single template and be registered with CMIC.

Registration is a one-time process and you do not need to register an application again after installation. The registration code can be part of post-installation script when the application is installed.

All templates must be present under *NMS-ROOT/object/data/cmfcmic/mst-templates*. Application teams can drop their template at this location and then mention the URN ( same as file name with out the .xml extension) during their registration with CMIC. You can have only one MST file for an application.

The following table describes the integration tags used by the CWHP:

**Table 9-1 CWHP Integration Tags**

CWHP Function	CMIC MST Integration Tag	Description
Cisco Works Applications installed on the same server	CWHP_APP_TASK	Use this tag for a task in the MST file to show the task in the relevant application panel of CWHP.
Cisco Works Resources	CWHP_CW_RSRC	Use this tag of a task in the MST file to show the task in the Cisco Works Resources panel of CWHP.
Cisco.com Resources	CWHP_CSCO_RSRC	Use this tag for a task in the MST file will show this task in the Cisco.com Resources panel of CWHP.
Cisco Works Applications installed on other server	CWHP_OTHR_APPS	Use this tag for a task in the MST file will show this task in the Cisco Works Other Servers panel of CWHP.
3rd Party Applications	CWHP_THRD_PRTY	Use this tag for a task in the MST file will show this task in the Third Party Applications panel of CWHP.
Custom Tools	CWHP_CSTM_TOOL	Use this tag for a task in the MST file will show this task in the Custom Tools panel of CWHP.
Device Center	DC_TOOLS	Device Troubleshooting Panel
Setup Center	systemSetup	LMS Setup Center Panel
	securitySetup	
	dataCollectionSettings	
	dataCollectionSchedule	
	dataPurge	

## Wrapper Java Code

During the application's post install, you need to call the CMIC register APIs to register the MST file with CMIC. This requires that you write a Java wrapper which will accept the MST file name. You can use the following Wrapper Java Code to register the MST file with CMIC:

```
$NMSROOT/lib/jre/bin/java -classpath
$NMSROOT/lib/classpath/cmfc.jar:$NMSROOT/MDC/tomcat/lib/apps/MICE.jar:$NMSROOT/MDC/tomcat/
lib/apps/NATIVE.jar:$NMSROOT/objects/log4j/1.1.3/log4j.jar:$NMSROOT/MDC/tomcat/lib/apps/xerces.jar com.cisco.nm.cmfc.mic.registry.CMICApplicationRegistry URN legacy action
[disable-option].
```

This code uses the arguments shown in [Table 9-2](#).

**CISCO CONFIDENTIAL****Table 9-2 Register and Unregister Arguments**

Arguments	Description
URN	URN of the application to be registered.
action	Use <i>register</i> to register the application. Use this argument in the application's post install code. Use <i>unregister</i> for unregistering the application. Use this argument in the application's post remove code.
legacy	The argument indicates whether the application is a legacy application or a new application. Use 0 for new applications running under the Tomcat servlet engine. This argument helps in deciding the port and protocol for the application.
disable option	(optional) This is used to specify whether the application should be hidden in the UI when displaying the list of applications. This would help prevent the user from unregistering the application through UI. <b>self</b> -- Do not show application link if the application is present. But show the link when it is imported from other servers <b>all</b> -- Do not show the application link anywhere. <b>none</b> -- Default option. When unspecified, shows the link irrespective of application location

To register RME4.0 using Tomcat with default display option, use the following code in post install:

```
$NMSROOT/lib/jre/bin/java -classpath
$NMSROOT/lib/classpath/cmhc.jar:$NMSROOT/MDC/tomcat/lib/apps/MICE.jar:$NMSROOT/MDC/tomcat/
lib/apps/NATIVE.jar:$NMSROOT/objects/log4j/1.1.3/log4j.jar:$NMSROOT/MDC/tomcat/lib/apps/xerces.jar com.cisco.nm.cmf.cmhc.registry.CMHCApplicationRegistry RME4.0 0 register.
```

To register RME4.0 using Tomcat with an option to display an application link, use the following code in post install:

```
$NMSROOT/lib/jre/bin/java -classpath
$NMSROOT/lib/classpath/cmhc.jar:$NMSROOT/MDC/tomcat/lib/apps/MICE.jar:$NMSROOT/MDC/tomcat/
lib/apps/NATIVE.jar:$NMSROOT/objects/log4j/1.1.3/log4j.jar:$NMSROOT/MDC/tomcat/lib/apps/xerces.jar com.cisco.nm.cmf.cmhc.registry.CMHCApplicationRegistry RME4.0 0 register all.
```

To unregister RME4.0 using Tomcat, use the following code in post remove:

```
$NMSROOT/lib/jre/bin/java -classpath
$NMSROOT/lib/classpath/cmhc.jar:$NMSROOT/MDC/tomcat/lib/apps/MICE.jar:$NMSROOT/MDC/tomcat/
lib/apps/NATIVE.jar:$NMSROOT/objects/log4j/1.1.3/log4j.jar:$NMSROOT/MDC/tomcat/lib/apps/xerces.jar com.cisco.nm.cmf.cmhc.registry.CMHCApplicationRegistry RME4.0 0 unregister.
```

## System Flow for CWHP using CMIC

- Each Cisco Works Product like RME, CM as well as Cisco Works Resources and Cisco.com resources installed on the same server where CWHP is installed, will call Register API of CMIC passing template URN, host, port, protocol.  
Cisco Products Installed on other servers, 3rd Party Applications, Custom home grown tools will be created by the end user through a CMIC Administration User Interface.
- Based on the template URN, CMIC Registry will fetch the template from MST Template Store and updates the MST file with the host, port, and protocol. After registration, the MST template is stored in two places in .xml format:
  - In Registered templates store in .xml format with host, port, protocol filled in.
  - In CMIC registry in serialized format with host, port, and protocol filled in.

**CISCO CONFIDENTIAL**

3. During startup CWHP loads a Servlet that searches CMIC Registry on all CWHP related Integration Tags.
4. CWHP Servlet store the CWHP related CMIC information in cache to improve CWHP performance.
5. User accesses CWHP page and this invokes CWHP Servlet which in turn calls CMIC to make sure it is in sync with CMIC and if not recreate the Cache.
6. CWHP Servlet then fetches the information from Cache
7. CWHP Servlet then invokes CWHP Security UIIAuthorize Interface and returns the tasks authorized for the user. The User see the CWHP page displayed with only authorized tasks for him.

## About the CMIC APIs

The following APIs are supported:

**Table 9-3** *CMIC APIs*

API	Description
register()	To register an application with CMIC.
unregister()	To unregister an application with CMIC.
searchRegistry()	To search for a specific or list of applications in registry.
getAllRegisteredApplications()	To get a list of all registered applications.
searchRegistryReturnTree()	To look for a specific or list of applications in registry and return in a hierarchical structure.
isRegistryUpdated()	To check whether registry is updated with reference to a time stamp.

Note: Please refer the [CMIC Java doc](#) for complete list of API signatures and definitions.

## About the Management Service Template

Currently Cisco runs a partner program called Cisco Management Connection, which allows third party Applications to add a launch point from CiscoWorks desktop. All such applications need to follow a certification process for their links to appear in the desktop. A similar program will be introduced to certify the MST's. This will be driven from CCO, where individual applications will be guided to create and update templates. The created template will be certified by a gatekeeper and made available to CMIC registry through PSU. All Cisco Application services will follow the same process to create and update their templates.

The information in Management Service Template is organized under various tags. The following table describes the tags and attributes used by the MST:

**CISCO CONFIDENTIAL****Table 9-4 CMIC MST Tags**

Tag	Description
APPLICATIONRECORD	All attributes common to the application such as application name, application version and description
VENDORINFO	Vendor details are captured under this tag.
TASKGROUP	Used for grouping of task information.
TASKINFO	An application can have multiple tasks associated with it. Each task can be associated with one or more 'TASKGROUP' (which has a GroupName, GroupURL and GroupURLWindowName attributes) in a nested structure, or you can leave the task without associating it to a TASKGROUP.
INTEGRATIONTAG	Each task can have multiple 'INTEGRATIONTAG' elements. Applications that want to have a closer integration with other applications, define an integration tag.
ATTRIBUTES	A task can be associated with one or more integration tags. If an application task requires few custom attributes, it can make use of the 'ATTRIBUTES' element to define the custom data.
WSDL	Each task has 'WSDL' tag, the information captured here would be what input a service takes, what output it results, the format of input & output. This is defined using the Web Services Description Language (WSDL).

For more information on MST tags and attributes of each tag see, CMIC Software Design Specification, EDCS-137328.

## Component Interaction

The table provides information on the components that participate in the interaction between various components.

**Table 9-5 CMIC Component Interaction**

Components	Description
CMIC User Interface	Handles all user related registrations and unregistration. The user can also browse through registry entries under various categories. The user interface authorizes and authenticates each user with the security services before displaying the UI.
CMIC API	Interface through which services are registered, unregistered and queried in the repository.
CSTM	All external API calls (NB API) over the network will be routed through Common Transport Mechanism (CTM). CTM is a service that accepts incoming binary requests; typically the interface will authorize all such calls and pass the request to corresponding service that handles the request (in this case CMIC). CMIC process the request API and sends the result back to CTM, which in turn sends it back to the caller.  For more details on CTM please refer to CTM functional specification.



**CISCO CONFIDENTIAL****Table 9-5** *CMIC Component Interaction*

Components	Description
Security Services	All authentication/authorization required by user interface and CTM interface will be fetched from security services.
PSU	Package Support Update is used to fetch MST template additions or updation to existing templates from CCO. This component is not required in the first phase.
Template Store	All downloaded MST templates from PSU will be stored here and is referred whenever there is a register/Unregister call by CMIC API.

## About CMIC Registry Dependencies

The CMIC registry depends on the services shown in Table.

**Table 9-6** *CMIC Dependencies*

Service	Description
Security Services	Authentication and Authorization of users, API calls received over network.
Common Services Transport Mechanism (CSTM)	Enabling API calls to be made over the network.
Desktop Services	Include UII, Web Server, and Tomcat.
Log and Trace	Logging service for recording various events and debugging.
Locking Service	For resource synchronization.
Backup and Restore	CMIC registry will provide a hook, which will do the backing up of all registered application data. The hook is called by the backup and restore service.

## Sample MST File

The following file is used by RME to register launch points with CMIC:

```
<?xml version="1.0" encoding="UTF-8"?>

<!--*****-->
<!--      Copyright (c) 2003 Cisco Systems, Inc.      -->
<!--      All rights reserved.      -->
<!--*****-->

<APPLICATIONRECORD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ws="http://schemas.xmlsoap.org/wsdl/" xsi:noNamespaceSchemaLocation="../cmic.xsd"
AppName="RME" AppVersion="4.0" TemplateVersion="1.0" AppDescription="Resource Manager
Essentials" IsCiscoCertified="false" IsCisco="false" Protocol="" Host="" Port="23"
ModifiedUser="Madan" ModifiedTime="12 " AppURL="/rme/goHome.do"
```

**CISCO CONFIDENTIAL**

```

AppURLWindowName="RME4.0" > <VENDORINFO VendorName="Cisco Systems" Address="170
West TasmanDr. San Jose, CA 95134 USA" Phone="(408)526-4000" Fax="(408)526-4000"
Email="tac@cisco.com" ContactURL="http://www.cisco.com"
SupportURL="http://www-tac.cisco.com"/>

<TASKGROUP GroupName="Resource Manager Essentials" GroupURL="/rme/goHome.do">

  <TASKINFO TaskName="Devices" TaskIdentity="a" TaskDescription="Device Management"
TaskCategory="C" TaskSubCategory="C/admin" SecurityTag="nm.cm.admin"
TaskURL="/rme/deviceMgmtDflt.do" SubmitMethod="POST" IsAPI="false">

  <INTEGRATIONTAG TagName="CWHP_TASK">

</INTEGRATIONTAG>

  </TASKINFO>

  <TASKINFO TaskName="Configuration" TaskIdentity="b" TaskDescription="Configuration
Management" TaskCategory="C" TaskSubCategory="C/admin" SecurityTag="nm.cm.admin"
TaskURL="/rme/configurationDefault.do" SubmitMethod="POST" IsAPI="false">

  <INTEGRATIONTAG TagName="CWHP_TASK">

</INTEGRATIONTAG>

  </TASKINFO>

  <TASKINFO TaskName="Image Management" TaskIdentity="c" TaskDescription="Image
Distribution" TaskCategory="C" TaskSubCategory="C/admin" SecurityTag="nm.cm.admin"
TaskURL="/rme/swimDefault.do" SubmitMethod="POST" IsAPI="true">

  <INTEGRATIONTAG TagName="CWHP_TASK">

</INTEGRATIONTAG>

  </TASKINFO>

  <TASKINFO TaskName="Job Management" TaskIdentity="d" TaskDescription="Job Browser"
TaskCategory="C" TaskSubCategory="C/admin" SecurityTag="nm.cm.admin"
TaskURL="/rme/JobMgmt.do" SubmitMethod="POST" IsAPI="false">

  <INTEGRATIONTAG TagName="CWHP_TASK">

</INTEGRATIONTAG>

  </TASKINFO>

  <TASKINFO TaskName="Reports" TaskIdentity="e" TaskDescription="Report Administration"
TaskCategory="C" TaskSubCategory="C/admin" SecurityTag="nm.cm.admin"
TaskURL="/rme/CriReportJob.do" SubmitMethod="POST" IsAPI="false">

  <INTEGRATIONTAG TagName="CWHP_TASK">

</INTEGRATIONTAG>

  </TASKINFO>

  <TASKINFO TaskName="Tools" TaskIdentity="f" TaskDescription="RME Tools"
TaskCategory="C" TaskSubCategory="C/admin" SecurityTag="nm.cm.admin"
TaskURL="/rme/toolsDefault.do" SubmitMethod="POST" IsAPI="false">

  <INTEGRATIONTAG TagName="CWHP_TASK">

</INTEGRATIONTAG>

  </TASKINFO>

```

**CISCO CONFIDENTIAL**

```
<TASKINFO TaskName="Admin" TaskIdentity="g" TaskDescription="Administration of RME
Server" TaskCategory="C" TaskSubCategory="C/admin" SecurityTag="nm.cm.admin"
TaskURL="/rme/adminDefault.do" SubmitMethod="POST" IsAPI="false">

  <INTEGRATIONTAG TagName="CWHP_TASK">

    </INTEGRATIONTAG>

  </TASKINFO>

</TASKGROUP>

</APPLICATIONRECORD>
```

***CISCO CONFIDENTIAL***