

**Oracle Utilities Integration Cloud  
Controller**

Interface Guide

Release 18.1

**E91268-01**

May 2018

Oracle Utilities Integration Cloud Controller Release Release 18.1 Overview Guide

Copyright © 2012, 2018 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

---

---

# Contents

## Chapter 1

<b>Oracle Utilities Integration Cloud Controller Overview .....</b>	<b>1-1</b>
The Oracle Utilities Integration Cloud Controller .....	1-2
Interface Format .....	1-3
About Integration Cloud Controller Resource Paths .....	1-3

## Chapter 2

<b>Application API.....</b>	<b>2-1</b>
Application Service - REST API.....	2-2
GET - /application!1.0/integration/apps.....	2-2

## Chapter 3

<b>Credentials 12.2.1+ API.....</b>	<b>3-1</b>
Credential Service - REST API .....	3-2
GET - /credential!1.0/integration/credential .....	3-2
POST - /credential!1.0/integration/credential .....	3-3
PUT - /credential!1.0/integration/credential.....	3-4
DELETE - /credential!1.0/integration/credential?key=<key>&map=<map> .....	3-5
GET - /credential!1.0/integration/credential/<map>/<key>.....	3-6

## Chapter 4

<b>Credentials 12.1.3 API .....</b>	<b>4-1</b>
Credentials 12.1.3 - REST API.....	4-2
POST - /credential12.1!1.0/integration/credential .....	4-2
PUT - /credential12.1!1.0/integration/credential.....	4-3
DELETE - /credential12.1!1.0/integration/credential?key=<key>&map=<map>.....	4-4

## Chapter 5

<b>Configuration Properties API .....</b>	<b>5-1</b>
Configuration Service - REST API.....	5-2
GET - /config!1.0/integration/config?appId=<appId>&filename=<config_file_name> .....	5-2
POST - /config!1.0/integration/config .....	5-5

## Chapter 6

<b>DVM API .....</b>	<b>6-1</b>
DVM Service - REST API .....	6-2
GET - /dvm!1.0/integration/dvm?appId=<appId>&dvmName=<dvmName> .....	6-2
POST - /dvm!1.0/integration/dvm.....	6-4
GET - /dvm!1.0/integration/dvm/list?appId=<appId>.....	6-5
POST - /dvm!1.0/integration/dvm/list .....	6-8

## Chapter 7

<b>Internal API .....</b>	<b>7-1</b>
Internal Properties - REST API.....	7-2
GET - /internal!1.0/integration/props.....	7-2
GET - /internal!1.0/integration/props/{key} .....	7-3

---

POST - /internal1.0/integration/props .....	7-4
DELETE - /internal1.0/integration/props/{key} .....	7-5

# Chapter 1

---

## Oracle Utilities Integration Cloud Controller Overview

This guide provides an overview of the Oracle Utilities Integration Cloud Controller interface, including:

- [The Oracle Utilities Integration Cloud Controller](#)
- [Interface Format](#)

# The Oracle Utilities Integration Cloud Controller

This document describes the Oracle Utilities Integration Cloud Controller. The Integration Cloud Controller is used for managing operational requirements of Oracle Utilities integration applications in cloud implementations.

In a traditional integration application, integrations are managed through local scripts, WebLogic console, Enterprise manager, or the SOA composer application. In a cloud environment customers do not have direct access to scripts or web consoles and will require tools to manage integration applications.

The Oracle Utilities Integration Cloud controller application enables customers to manage Oracle Utilities integration applications in the Cloud environment using REST services

---

# Interface Format

The APIs in this document are presented in the following format:

API Name

Table summarizing methods available in the API.

Each method available is described by the following:

- Name and resource path
- Description of the method
- Full resource path
- Input parameters
- Output (in both XML and JSON formats)

## About Integration Cloud Controller Resource Paths

The resource paths for all Integration Cloud Controller API methods begin with the following stem:

```
http://<host>:<port>/soa-infra/resources/ICC
```

This stem is followed by details based on the specific API and method. For example, the resource path for the GET method in the Application API would append

```
"/application!1.0/integration/apps"
```

to the stem.

The full resource path would be:

```
http://<host>:<port>/soa-infra/resources/ICC/application!1.0/  
integration/apps
```

# Chapter 2

---

## Application API

The Application API can be used to query and view which integration applications are installed in the cloud integration environment.

This API provides the ability to query and get a list of integration applications installed on the environment. This provides users with Application Ids which can be then used for subsequent API calls to specific integration applications.



# Application Service - REST API

Resource Path	Method	Description
http://<host>:<port>/soa-infra/resources/ICC/application!1.0/integration/apps	GET	Fetch list of installed integration applications for the cloud environment

## GET - /application!1.0/integration/apps

### Description

This service fetches a list of installed integration applications for the cloud environment on the WebLogic server. The Id of the integration application can be used for subsequent API calls.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/application!1.0/integration/apps

### Input

None

### Output

XML Output:

```
<applicationList>
  <application>
    <id>CCB2-MDM2</id>
    <name>Customer Care and Billing integration to Meter Data
Management</name>
    <description>Integration between CCB and MDM</description>
  </application>
  <application>
    <id>OUWAMERP</id>
    <name>Work and Asset Management integration to ERP</name>
    <description>Integration between WAM and ERP</description>
  </application>
</applicationList>
```

## JSON Output:

```
{
  "application": [
    {
      "id": "CCB2-MDM2",
      "name": "Customer Care and Billing integration to Meter
Data Management",
      "description": "Integration between CCB and MDM"
    },
    {
      "id": "OUWAMERP",
      "name": "Work and Asset Management integration to ERP",
      "description": "Integration between WAM and ERP"
    }
  ]
}
```

# Chapter 3

---

## Credentials 12.2.1+ API

Credentials (CSF keys) are used by integration applications for authentication with the edge applications. The Credentials API provides the ability to create, update or delete credentials.

Weblogic Security Credentials are used by integration web service security policies for authentication with edge applications.

**Note:** The Credentials 12.2.1+ API is supported only on WebLogic 12.2.1+ server. This version has the OWSM REST API available which is internally being used.

# Credential Service - REST API

Resource Path	Method	Description
http://<host>:<port>/soa-infra/resources/ICC/credential!1.0/integration/credential	GET	Fetch list of credentials available on the WebLogic server
http://<host>:<port>/soa-infra/resources/ICC/credential!1.0/integration/credential	POST	Create credential
http://<host>:<port>/soa-infra/resources/ICC/credential!1.0/integration/credential	PUT	Update credential
http://<host>:<port>/soa-infra/resources/ICC/credential!1.0/integration/credential?key=<key>&map=<map>	DELETE	Delete credential
http://<host>:<port>/soa-infra/resources/ICC/credential!1.0/integration/credential/<map>/<key>	GET	Fetch the credential if it exists on the server

## GET - /credential!1.0/integration/credential

### Description

This service fetches a list of all CSF keys available on the WebLogic server. The CSF key and CSF map name are returned by the service.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/credential!1.0/integration/credential

### Input

None

### Output

XML Output:

```
<json>
  <oracle.wsm.security>OU_CCB_01</oracle.wsm.security>
  <oracle.wsm.security>OU_MWM_01</oracle.wsm.security>
  <oracle.wsm.security>OU_WAM_01</oracle.wsm.security>
  <oracle.wsm.security>FW_PIP_01</oracle.wsm.security>
  <oracle.wsm.security>demoCredential</oracle.wsm.security>
  <oracle.wsm.security>OUWAM_01</oracle.wsm.security>
  <oracle.wsm.security>CCB2-MDM2_MDM2</oracle.wsm.security>
  <CSF_MAP_NAME>CSF_KEY_NAME</CSF_MAP_NAME>
  <ICC>testCred1</ICC>
</json>
```

JSON Output:

```
{
  "oracle.wsm.security": [
    "OU_CCB_01",
    "OU_MWM_01",
    "OU_WAM_01",
    "FW_PIP_01",
    "demoCredential",
    "OUWAM_01",
    "CCB2-MDM2_MDM2"
  ],
  "CSF_MAP_NAME": "CSF_KEY_NAME",
  "ICC": "testCred1"
}
```

## POST - /credential!1.0/integration/credential

### Description

This service will create a new CSF key on the WebLogic server.

### Resource Path

<http://<host>:<port>/soa-infra/resources/ICC/credential!1.0/integration/credential>

### Input

- Key (CSF key name) - required
- map (CSF map name) - required
- username - required
- password (Password for the username) - required

XML Input:

```
<credential>
  <username>newuser</username>
  <password>password123</password>
  <key>demoCredential</key>
  <map>oracle.wsm.security</map>
</credential>
```

JSON Input:

```
{
  "credential": {
    "username": "testuser",
    "password": "password1234",
    "key": "demoCredential",
    "map": "oracle.wsm.security"
  }
}
```

## Output

XML Output:

```
<json>
  <iccResponse>
    <message>Credential added successfully.</message>
    <status>200</status>
    <code>ICCRresponse</code>
  </iccResponse>
</json>
```

JSON Output:

```
{
  "iccResponse": {
    "message": "Credential added successfully.",
    "status": "200",
    "code": "ICCRresponse"
  }
}
```

## PUT - /credential!1.0/integration/credential

### Description

This service will update an existing CSF key on the WebLogic server.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/credential!1.0/integration/credential

### Input

- Key (CSF key name) - required
- map (CSF map name) - required
- username - required
- password (Password for the username) - required

XML Input:

```
<credential>
  <username>newuser</username>
  <password>password123</password>
  <key>demoCredential</key>
  <map>oracle.wsm.security</map>
</credential>
```

JSON Input:

```
{
  "credential": {
    "username": "testuser",
    "password": "password1234",
    "key": "demoCredential",
    "map": "oracle.wsm.security"
  }
}
```

```
    }
  }
```

## Output

XML Output:

```
<json>
  <iccResponse>
    <message>Credential updated successfully.</message>
    <status>200</status>
    <code>ICCResponse</code>
  </iccResponse>
</json>
```

JSON Output:

```
{
  "iccResponse": {
    "message": "Credential updated successfully.",
    "status": "200",
    "code": "ICCResponse"
  }
}
```

## DELETE - /credential!1.0/integration/credential?key=<key>&map=<map>

### Description

This service will delete an existing CSF key on the WebLogic server.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/credential!1.0/integration/credential?key=<key>&map=<map>

### Input

- Key (CSF key name) - required - Query parameter
- Map (CSF map name) - required- Query parameter

### Output

XML Output:

```
<json>
  <iccResponse>
    <message>Credential deleted successfully.</message>
    <status>200</status>
    <code>ICCResponse</code>
  </iccResponse>
</json>
```

JSON Output:

```
{
  "iccResponse": {
    "message": "Credential deleted successfully.",
    "status": "200",
    "code": "ICCResponse"
  }
}
```

## GET - /credential!1.0/integration/credential/<map>/<key>

### Description

This service fetches an existing CSF key on the WebLogic server.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/credential1.0/integration/credential/<map>/<key>

### Input

- Key (CSF key name) - required - URI parameter
- Map (CSF map name) - required- URL parameter

### Output

XML Output:

```
<json>
  <credential>
    <map>oracle.wsm.security</map>
    <key>demoCredential1</key>
  </credential>
</json>
```

JSON Output:

```
{
  "credential": {
    "map": "oracle.wsm.security",
    "key": "demoCredential1"
  }
}
```



# Chapter 4

---

## Credentials 12.1.3 API

Credentials (CSF keys) are used by integration applications for authentication with the edge applications. The Credentials API provides the ability to create, update or delete credentials.

Weblogic Security Credentials are used by integration web service security policies for authentication with edge applications.

**Note:** The Credentials 12.1.3 API is supported only on WebLogic 12.1.3 server. This version uses WLST to manage the credentials.

## Credentials 12.1.3 - REST API

Resource Path	Method	Description
http://<host>:<port>/soa-infra/resources/ICC/credential12.111.0/integration/credential	POST	Create credential
http://<host>:<port>/soa-infra/resources/ICC/credential12.111.0/integration/credential	PUT	Update credential
http://<host>:<port>/soa-infra/resources/ICC/credential12.111.0/integration/credential?key=<key>&map=<map>	DELETE	Delete credential

### POST - /credential12.111.0/integration/credential

#### Description

This service will create a new CSF key on the WebLogic server.

#### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/credential12.111.0/integration/credential

#### Input

- Key (CSF key name) - required
- map (CSF map name) - required
- username - required
- password (Password for the username) - required

XML Input:

```
<credential xmlns="http://utilities.oracle.com/icc/credential">
  <username>newuser</username>
  <password>password123</password>
  <key>demoCredential</key>
  <map>oracle.wsm.security</map>
</credential>
```

JSON Input:

```
{
  "username": "newuser",
  "password": "password123",
  "key": "demoCredential113",
  "map": "oracle.wsm.security"
}
```

## Output

XML Output:

```
<iccResponse
  xmlns="http://utilities.oracle.com/icc/response">
  <status>200</status>
  <code>ICCRresponse</code>
  <message>Credential added successfully.</message>
  <description/>
</iccResponse>
```

JSON Output:

```
{
  "status": 200,
  "code": "ICCRresponse",
  "message": "Credential added successfully.",
  "description": null
}
```

## PUT - /credential12.1!1.0/integration/credential

### Description

This service will update an existing CSF key on the WebLogic server.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/credential12.1!1.0/integration/credential

### Input

- Key (CSF key name) - required
- map (CSF map name) - required
- username - required
- password (Password for the username) - required

XML Input:

```
<credential xmlns="http://utilities.oracle.com/icc/credential">
  <username>newuser</username>
  <password>password123</password>
  <key>demoCredential</key>
  <map>oracle.wsm.security</map>
</credential>
```

JSON Input:

```
{
  "username": "newuser",
  "password": "password123",
  "key": "demoCredential113",
  "map": "oracle.wsm.security"
}
```

## Output

XML Output:

```
<iccResponse
  xmlns="http://utilities.oracle.com/icc/response">
  <status>200</status>
  <code>ICCRresponse</code>
  <message>Credential updated successfully.</message>
  <description/>
</iccResponse>
```

JSON Output:

```
{
  "status": 200,
  "code": "ICCRresponse",
  "message": "Credential updated successfully.",
  "description": null
}
```

## DELETE - /credential12.1!1.0/integration/ credential?key=<key>&map=<map>

### Description

This service will delete an existing CSF key on the WebLogic server.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/credential12.1!1.0/integration/  
credential?key=<key>&map=<map>

### Input

- Key (CSF key name) - required - Query parameter
- Map (CSF map name) - required- Query parameter

### Output

XML Output:

```
<iccResponse
  xmlns="http://utilities.oracle.com/icc/response">
  <status>200</status>
  <code>ICCRresponse</code>
  <message>Credential deleted successfully.</message>
  <description/>
</iccResponse>
```

JSON Output:

```
{
  "status": 200,
  "code": "ICCRresponse",
  "message": "Credential deleted successfully.",
  "description": null
}
```

# Chapter 5

---

## Configuration Properties API

Integration applications use a configuration properties file for setting things like default values and end point URL's for edge application services.

The Configuration Properties API provides the ability to:

- Query the integration configuration properties file
- Update the integration configuration properties in Metadata Services (MDS)

**Note:** The SOA server needs to be restarted after the Configuration Properties is updated for the new values to take effect.

# Configuration Service - REST API

Resource Path	Method	Description
http://<host>:<port>/soa-infra/resources/ICC/config!1.0/integration/config?appId=<appId>&filename=<config_file_name>	GET	Read Configuration file for the particular integration from MDS, write to Controller Home directory and send as response
http://<host>:<port>/soa-infra/resources/ICC/config!1.0/integration/config	POST	Write Configuration properties in service input to the Controller Home directory and then move it to MDS

## GET - /config!1.0/integration/config?appId=<appId>&filename=<config\_file\_name>

### Description

This service fetches the integration configuration properties file available for the particular integration application. The configuration file is stored in the MDS database and this service will retrieve it from the MDS schema, write it to the controller home and then send as an output to the caller.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/config!1.0/integration/config?appId=<appId>&filename=<config\_file\_name>

### Input

- appId (integration application Id) - required - Query parameter
- filename (config file name) - optional - Query parameter. If not passed the filename will be ConfigurationProperties.xml

### Output

XML Output:

```
<config appId="cloud" filename="ConfigurationProperties.xml"
xmlns="http://xmlns.oracle.com/aia/core/config/V1">

  <AIAConfiguration version="1.0">
    <SystemConfiguration>
      <ModuleConfiguration moduleName="cloud testing">
        <Property name="CCB2.Generic.MessageCategory">11017</Property>
        <Property
name="CCB2.Generic.BusinessException.MessageNumber">11001</
Property>
        <Property name="CCB2.Generic.DVMEException.MessageNumber">11401</
Property>
      </ModuleConfiguration>
    </SystemConfiguration>
  </AIAConfiguration>
</config>
```

```
<Property name="DVM.OUCCB2_OUMDM2_MO.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_ContactType.ThrowException">false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_Country.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_DisconnectLocation.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_MeasurementCycle.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_MeasurementCycleRoute.ThrowException">fals
e</Property>
<Property
name="DVM.OUCCB2_OUMDM2_SPSourceStatus.ThrowException">false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_SPStatus.ThrowException">false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_SPTType.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_LifeSupportSensitiveLoad.ThrowException">f
alse</Property>
<Property name="DVM.OUCCB2_OUMDM2_USType.ThrowException">false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_SASStatus.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_DeviceType.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_MeterStatus.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_Manufacturer.ThrowException">false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_Model.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_DeviceConfigType.ThrowException">false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_HowToUse.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_ReadOutType.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_ConsumptiveSubtractive.ThrowException">fal
se</Property>
<Property
name="DVM.OUCCB2_OUMDM2_MCSERVICEType.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_NegativeConsumption.ThrowException">false<
/Property>
<Property
name="DVM.OUCCB2_OUMDM2_BillCondition.ThrowException">false</
Property>
```



```

<Property
name="DVM.OUCCB2_OUMDM2_AllowEstimate.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_BillingOption.ThrowException">false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_BillMode.ThrowException">false</
Prope
.
.
.
</config>

```

## JSON Output:

```

{
  "@appId": "cloud",
  "@filename": "ConfigurationProperties.xml",
  "AIAConfiguration": {
    "@version": 1,
    "SystemConfiguration": {
      "ModuleConfiguration": [
        {
          "@moduleName": "cloud testing",
          "Property": [
            {
              "@name": "CCB2.Generic.MessageCategory",
              "$": "11017"
            },
            {
              "@name":
"CCB2.Generic.BusinessException.MessageNumber",
              "$": "11001"
            },
            {
              "@name":
"CCB2.Generic.DVMException.MessageNumber",
              "$": "11401"
            },
            {
              "@name": "DVM.OUCCB2_OUMDM2_MO.ThrowException",
              "$": "false"
            },
            {
              "@name":
"DVM.OUCCB2_OUMDM2_ContactType.ThrowException",
              "$": "false"
            }
          ]
        }
      ]
    }
  }
}

```

## POST - /config!1.0/integration/config

### Description

This service will update the integration configuration properties file for the particular integration application. The configuration file is written to the controller home directory and then updated in the MDS database.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/config!1.0/integration/config

### Input

- appId (integration application Id) - required - part of the payload
- filename (config file name) - required - part of the payload
- Configuration File data - required - payload as shown below

XML Input:

```
<config appId="cloud" filename="ConfigurationProperties.xml"
xmlns="http://xmlns.oracle.com/aia/core/config/V1">

    <AIAConfiguration version="1.0">
<SystemConfiguration>
<ModuleConfiguration moduleName="cloud testing">
<Property name="CCB2.Generic.MessageCategory">11017</Property>
<Property
name="CCB2.GenericBusinessException.MessageNumber">11001</
Property>
<Property name="CCB2.GenericDVMException.MessageNumber">11401</
Property>
<Property name="DVM.OUCCB2_OUMDM2_MO.ThrowException">>false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_ContactType.ThrowException">>false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_Country.ThrowException">>false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_DisconnectLocation.ThrowException">>false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_MeasurementCycle.ThrowException">>false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_MeasurementCycleRoute.ThrowException">fals
e</Property>
<Property
name="DVM.OUCCB2_OUMDM2_SPSourceStatus.ThrowException">>false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_SPStatus.ThrowException">>false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_SPTType.ThrowException">>false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_LifeSupportSensitiveLoad.ThrowException">f
alse</Property>
```

```
<Property name="DVM.OUCCB2_OUMDM2_UStype.ThrowException">false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_SASStatus.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_DeviceType.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_MeterStatus.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_Manufacturer.ThrowException">false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_Model.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_DeviceConfigType.ThrowException">false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_HowToUse.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_ReadOutType.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_ConsumptiveSubtractive.ThrowException">fal
se</Property>
<Property
name="DVM.OUCCB2_OUMDM2_MCSERVICEType.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_NegativeConsumption.ThrowException">false<
/Property>
<Property
name="DVM.OUCCB2_OUMDM2_BillCondition.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_AllowEstimate.ThrowException">false</
Property>
<Property
name="DVM.OUCCB2_OUMDM2_BillingOption.ThrowException">false</
Property>
<Property name="DVM.OUCCB2_OUMDM2_BillMode.ThrowException">false</
Prope
.
.
.
</config>
```

## JSON Input:

```

{
  "@appId": "cloud",
  "@filename": "ConfigurationProperties.xml",
  "AIAConfiguration": {
    "@version": 1,
    "SystemConfiguration": {
      "ModuleConfiguration": [
        {
          "@moduleName": "cloud testing",
          "Property": [
            {
              "@name": "CCB2.Generic.MessageCategory",
              "$": "11017"
            },
            {
              "@name":
"CCB2.GenericBusinessException.MessageNumber",
              "$": "11001"
            },
            {
              "@name":
"CCB2.GenericDVMException.MessageNumber",
              "$": "11401"
            },
            {
              "@name": "DVM.OUCCB2_OUMDM2_MO.ThrowException",
              "$": "false"
            },
            {
              "@name":
"DVM.OUCCB2_OUMDM2_ContactType.ThrowException",
              "$": "false"
            }
          ]
        }
      ]
    }
  }
}

```

**Output**

## XML Output:

```

<iccResponse
  xmlns="http://utilities.oracle.com/icc/response">
  <status>200</status>
  <code>ICCResponse</code>
  <message>Configuration Properties file has been updated
successfully.</message>
  <description/>
</iccResponse>

```

## JSON Output:

```
{
  "status": 200,
  "code": "ICCRresponse",
  "message": "Configuration Properties file has been updated
successfully.",
  "description": null
}
```

# Chapter 6

---

## DVM API

Integration applications use Domain Value Map (DVM) files to map values between different systems. These are XML files used to map key-value pairs across edge applications.

The DVM API provides the ability to

- Query configured DVM values
- Update DVM values as needed

# DVM Service - REST API

Resource Path	Method	Description
http://<host>:<port>/soa-infra/resources/ICC/dvm!1.0/integration/dvm?appId=<appId>&dvmName=<dvmName>	GET	Read DVM file for the particular integration from MDS, write to Controller Home directory and send as response.
http://<host>:<port>/soa-infra/resources/ICC/dvm!1.0/integration/dvm	POST	Write DVM file in service input to the Controller Home directory and then move it to MDS
http://<host>:<port>/soa-infra/resources/ICC/dvm!1.0/integration/dvm/list?appId=<appId>	GET	Read the all the DVMs for an integration application from MDS, write to Controller Home directory and send as consolidated response document
http://<host>:<port>/soa-infra/resources/ICC/dvm!1.0/integration/dvm/list	POST	Write all the DVMs for an integration application in the input to the Controller Home directory and then move them to MDS

## GET - /dvm!1.0/integration/dvm?appId=<appId>&dvmName=<dvmName>

### Description

This service fetches the DVM file available for the particular integration application. DVM file are stored in the MDS database and this service retrieves it from the MDS schema, writes it to the Controller Home and then sends it as an output to the caller.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/dvm!1.0/integration/dvm?appId=<appId>&dvmName=<dvmName>

### Input

- appId (integration application Id) - required - Query parameter
- dvmName (DVM file name) - required - Query parameter

### Output

XML Output:

```
<dvmData appId="cloud" dvmName="cloudDVM.dvm"
  xmlns="http://xmlns.oracle.com/dvm">
  <dvm name="OUCCB2_OUMDM2_AllowEstimate">
    <description> Mapping between CCB Allow Estimate field
and MDM Allow Estimate field.
```

```

</description>
  <columns>
    <column name="OUCCB2_AllowEstimate"/>
    <column name="OUMDM2_AllowEstimate"/>
  </columns>
  <rows>
    <row>
      <cell>C1DN</cell>
      <cell>No</cell>
    </row>
    <row>
      <cell>C1YS</cell>
      <cell>Yes</cell>
    </row>
  </rows>
</dvm>
</dvmData>

```

## JSON Output:

```

{
  "@appId": "cloud",
  "@dvmName": "cloudDVM.dvm",
  "dvm": {
    "@name": "OUCCB2_OUMDM2_AllowEstimate",
    "description": "      Mapping between CCB Allow Estimate field
and MDM Allow Estimate field.\n  ",
    "columns": {
      "column": [
        {
          "@name": "OUCCB2_AllowEstimate"
        },
        {
          "@name": "OUMDM2_AllowEstimate"
        }
      ]
    },
    "rows": {
      "row": [
        {
          "cell": [
            "C1DN",
            "No"
          ]
        },
        {
          "cell": [
            "C1YS",
            "Yes"
          ]
        }
      ]
    }
  }
}

```



## POST - /dvm!1.0/integration/dvm

### Description

This service updates the integration DVM file for the particular integration application. The DVM file is written to the Controller Home directory and then updated in the MDS database.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/dvm!1.0/integration/dvm

### Input

- appId (integration application Id) - required - Part of the payload
- dvmName (DVM file name) - required - Part of the payload
- DVM data - required - payload as shown below

XML Input:

```
<dvmData appId="cloud" dvmName="cloudDVM.dvm"
  xmlns="http://xmlns.oracle.com/dvm">
  <dvm name="cloudDVM">
    <description>    Mapping between CCB Allow Estimate field
and MDM Allow Estimate field.
  </description>
    <columns>
      <column name="OUCCB2_Flag"/>
      <column name="OUMDM2_Flag"/>
    </columns>
    <rows>
      <row>
        <cell>C1DN</cell>
        <cell>No</cell>
      </row>
      <row>
        <cell>C1YS</cell>
        <cell>Yes</cell>
      </row>
    </rows>
  </dvm>
</dvmData>
```

JSON Input:

```
{
  "@appId": "cloud",
  "@dvmName": "cloudDVM.dvm",
  "dvm": {
    "@name": "cloudDVM",
    "description": "    Mapping between CCB Allow Estimate field
and MDM Allow Estimate field.\n    ",
    "columns": {
      "column": [
        {
          "@name": "OUCCB2_Flag"
        }
      ],
    }
  }
}
```

```

        {
            "@name": "OUMDM2_Flag"
        }
    ],
    },
    "rows": {
        "row": [
            {
                "cell": [
                    "C1DN",
                    "No"
                ]
            },
            {
                "cell": [
                    "C1YS",
                    "Yes"
                ]
            }
        ]
    }
}

```

## Output

XML Output:

```

<iccResponse
  xmlns="http://utilities.oracle.com/icc/response">
  <status>200</status>
  <code>ICCSuccess</code>
  <message>DVM data has been updated successfully.</message>
  <description/>
</iccResponse>

```

JSON Output:

```

{
  "status": 200,
  "code": "ICCSuccess",
  "message": "DVM data has been updated successfully.",
  "description": null
}

```

## GET - /dvm!1.0/integration/dvm/list?appld=<appld>

### Description

This service fetches all the DVM files available for the particular integration application. DVM files are stored in the MDS database and this service will retrieve them from the MDS schema, write them to the Controller Home and then send as consolidated output document to the caller.

## Resource Path

http://<host>:<port>/soa-infra/resources/ICC/dvm!1.0/integration/dvm/list?appId=<appId>

## Input

- appId (integration application Id) - required - Query parameter

## Output

XML Output:

```

<dvmList appId="cloud"
  xmlns="http://xmlns.oracle.com/dvm">
  <dvmData dvmName="cloudDVM.dvm" appId="cloud">
    <dvm name="cloudDVM">
      <description> Mapping between CCB Allow Estimate field
and MDM Allow Estimate field.
    </description>
    <columns>
      <column name="OUCCB2_Flag"/>
      <column name="OUMDM2_Flag"/>
    </columns>
    <rows>
      <row>
        <cell>C1DN</cell>
        <cell>No</cell>
      </row>
      <row>
        <cell>C1YS</cell>
        <cell>Yes</cell>
      </row>
    </rows>
    </dvm>
  </dvmData>
  <dvmData dvmName="cloudDVM2.dvm" appId="cloud">
    <dvm name="cloudDVM2">
      <description> Mapping between CCB Country and MDM
Country.
    </description>
    <columns>
      <column name="OUCCB2_Country"/>
      <column name="OUMDM2_Country"/>
    </columns>
    <rows>
      <row>
        <cell>US</cell>
        <cell>USA</cell>
      </row>
      <row>
        <cell>IN</cell>
        <cell>INDIA</cell>
      </row>
    </rows>
    </dvm>
  </dvmData>
</dvmList>

```

## JSON Output:

```

{
  "@appId": "cloud",
  "dvmData": [
    {
      "@dvmName": "cloudDVM.dvm",
      "@appId": "cloud",
      "dvm": {
        "@name": "cloudDVM",
        "description": " Mapping between CCB Allow Estimate
field and MDM Allow Estimate field.\n ",
        "columns": {
          "column": [
            {
              "@name": "OUCCB2_Flag"
            },
            {
              "@name": "OUMDM2_Flag"
            }
          ]
        },
        "rows": {
          "row": [
            {
              "cell": [
                "C1DN",
                "No"
              ]
            },
            {
              "cell": [
                "C1YS",
                "Yes"
              ]
            }
          ]
        }
      }
    },
    {
      "@dvmName": "cloudDVM2.dvm",
      "@appId": "cloud",
      "dvm": {
        "@name": "cloudDVM2",
        "description": " Mapping between CCB Country and MDM
Country.\n ",
        "columns": {
          "column": [
            {
              "@name": "OUCCB2_Country"
            },
            {
              "@name": "OUMDM2_Country"
            }
          ]
        },
        "rows": {
          "row": [
            {
              "cell": [
                "US",

```



```

    </dvm>
  </dvmData>
  <dvmData dvmName="cloudDVM2.dvm" appId="cloud">
    <dvm name="cloudDVM2">
      <description> Mapping between CCB Country and MDM
Country.
    </description>
      <columns>
        <column name="OUCCB2_Country"/>
        <column name="OUMDM2_Country"/>
      </columns>
      <rows>
        <row>
          <cell>US</cell>
          <cell>USA</cell>
        </row>
        <row>
          <cell>IN</cell>
          <cell>INDIA</cell>
        </row>
      </rows>
    </dvm>
  </dvmData>
</dvmList>

```

#### JSON Input:

```

{
  "@appId": "cloud",
  "dvmData": [
    {
      "@dvmName": "cloudDVM.dvm",
      "@appId": "cloud",
      "dvm": {
        "@name": "cloudDVM",
        "description": " Mapping between CCB Allow Estimate
field and MDM Allow Estimate field.\n ",
        "columns": {
          "column": [
            {
              "@name": "OUCCB2_Flag"
            },
            {
              "@name": "OUMDM2_Flag"
            }
          ]
        },
        "rows": {
          "row": [
            {
              "cell": [
                "C1DN",
                "No"
              ]
            },
            {
              "cell": [
                "C1YS",
                "Yes"
              ]
            }
          ]
        }
      }
    }
  ]
}

```

```

    ]
  }
},
{
  "@dvmName": "cloudDVM2.dvm",
  "@appId": "cloud",
  "dvm": {
    "@name": "cloudDVM2",
    "description": " Mapping between CCB Country and MDM
Country.\n ",
    "columns": {
      "column": [
        {
          "@name": "OUCCB2_Country"
        },
        {
          "@name": "OUMDM2_Country"
        }
      ]
    },
    "rows": {
      "row": [
        {
          "cell": [
            "US",
            "USA"
          ]
        },
        {
          "cell": [
            "IN",
            "INDIA"
          ]
        }
      ]
    }
  }
}
]
}
}

```

## Output

XML Output:

```

<result>
  <status>Succeeded</status>
</result>

```

JSON Output:

```

{
  "status": 200,
  "code": "ICCSuccess",
  "message": "DVM data has been updated successfully.",
  "description": null
}

```

# Chapter 7

---

## Internal API

The Internal is used by system administrators and provides the ability to:

- Query the Configuration Properties table
- Add, update or delete properties in the Configuration Properties table



# Internal Properties - REST API

Resource Path	Method	Description
http://<host>:<port>/soa-infra/resources/ICC/internal!1.0/integration/props	GET	Read all the properties in the Configuration Properties table.
http://<host>:<port>/soa-infra/resources/ICC/internal!1.0/integration/props/{key}	GET	Read a single property in the Configuration Properties table based on a supplied key.
http://<host>:<port>/soa-infra/resources/ICC/internal!1.0/integration/props	POST	Add or update property in the Configuration Properties table.
http://<host>:<port>/soa-infra/resources/ICC/internal!1.0/integration/props/{key}	DELETE	Delete a property in the Configuration Properties table based on a supplied key.

## GET - /internal!1.0/integration/props

### Description

This service fetches all the properties in the Configuration Properties table.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/internal!1.0/integration/props

### Input

None

### Output

XML Output:

```
<propertyList
  xmlns:tns="http://utilities.oracle.com/icc"
  xmlns="http://utilities.oracle.com/icc">
  <tns:property>
    <tns:key>owsm.credential.service.endpoint</tns:key>
    <tns:value>http://den01bvb.us.oracle.com:7888/idaas/
platform/admin/v1/credential</tns:value>
    <tns:type>INT</tns:type>
    <tns:version>1</tns:version>
  </tns:property>
  <tns:property>
    <tns:key>icc.wlst.credentials.map</tns:key>
    <tns:value>ICC</tns:value>
    <tns:type>INT</tns:type>
    <tns:version>1</tns:version>
  </tns:property>
  <tns:property>
    <tns:key>icc.wlst.credentials.admin.key</tns:key>
```

```

        <tns:value>adminWlstKey</tns:value>
        <tns:type>INT</tns:type>
        <tns:version>1</tns:version>
    </tns:property>
    .
    .
    .
</propertyList >

```

**JSON Output:**

```

{
  "property": [
    {
      "key": "owsm.credential.service.endpoint",
      "value": "http://den01bvb.us.oracle.com:7888/idaas/
platform/admin/v1/credential",
      "type": "INT",
      "version": 1
    },
    {
      "key": "icc.wlst.credentials.map",
      "value": "ICC",
      "type": "INT",
      "version": 1
    },
    {
      "key": "icc.wlst.credentials.admin.key",
      "value": "adminWlstKey",
      "type": "INT",
      "version": 1
    },
    .
    .
    .
    . ]
}

```

**GET - /internal!1.0/integration/props/{key}****Description**

This service fetches all the properties from the Configuration Properties table based on a key passed as a parameter.

**Resource Path**

http://<host>:<port>/soa-infra/resources/ICC/internal!1.0/integration/props/{key}

**Input**

- Property Key - required - URI parameter

## Output

XML Output:

```
<property
  xmlns:tns="http://utilities.oracle.com/icc"
  xmlns="http://utilities.oracle.com/icc">
  <tns:key>oracle.home</tns:key>
  <tns:value>E:/Oracle/soa</tns:value>
  <tns:type>INT</tns:type>
  <tns:version>1</tns:version>
</property>
```

JSON Output:

```
{
  "key": "oracle.home",
  "value": "E:/Oracle/soa",
  "type": "INT",
  "version": 1
}
```

## POST - /internal!1.0/integration/props

### Description

This service adds or updates a property in the Configuration Properties table.

### Resource Path

<http://<host>:<port>/soa-infra/resources/ICC/internal!1.0/integration/props>

### Input

- key - required - Part of payload
- value - required - Part of payload
- type - required - INT or EXT (Internal or External) - Part of payload
- version- required - Part of payload

XML Input:

```
<property xmlns="http://utilities.oracle.com/icc">
  <key>new.test.property</key>
  <value>Test property</value>
  <type>INT</type>
  <version>1</version>
</property>
```

JSON Input:

```
{
  "key": "new.test.property",
  "value": "Test property",
  "type": "INT",
  "version": 1
}
```

## Output

XML Output:

```
<iccResponse
  xmlns="http://utilities.oracle.com/icc/response">
  <status>200</status>
  <code/>
  <message>ICC Property added/updated succesfully.</message>
  <description/>
</iccResponse>
```

JSON Output:

```
{
  "status": 200,
  "code": null,
  "message": "ICC Property added/updated succesfully.",
  "description": null
}
```

## DELETE - /internal!1.0/integration/props/{key}

### Description

This service deletes a property from the Configuration Properties table based on a key passed as a parameter.

### Resource Path

http://<host>:<port>/soa-infra/resources/ICC/internal!1.0/integration/props/{key}

### Input

- Property Key - required - URI parameter

## Output

XML Output:

```
<iccResponse
  xmlns="http://utilities.oracle.com/icc/response">
  <status>200</status>
  <code/>
  <message>ICC Property deleted successfully.</message>
  <description/>
</iccResponse>
```

JSON Output:

```
{
  "status": 200,
  "code": null,
  "message": "ICC Property deleted successfully.",
  "description": null
}
```