
MMD API Documentation

Release v3.0.3

uinnova

April 30, 2016

1	Introduce	1
2	Contents	3
2.1	API Concepts	3
2.2	Command Line API	3
2.3	Web Service API	6
2.4	JDBC API	9
2.5	Message Queue API	10

Introduce

This documentation provides guidelines on interact with uDCV thought API, the following API topic are included:

- *API Concepts*
- *Command line API*
- *Web Service API*
- *JDCB API*
- *Message Queue API*

Contents

2.1 API Concepts

uDCV provides 4 different API interfaces, which is command line, web service, JDBC, and message queue respectively.

2.2 Command Line API

2.2.1 Summary

uDCV come with a handy command line tools named uDCV-cli, which capable of send alarms, performance data, even add/update/delete asset records by a single command. This howto guide shows you how to use this powerful tool.

2.2.2 Install

- download uDCV-cli from uinnova website and unzip the tarball to uDCV server
https://www.dropbox.com/s/45w7h9as09vf4gg/uDCV_cli.zip?dl=0
- that is it !

2.2.3 Send Alarm

Usage

Open a windows command line, run `event.bat` with follow arguments:

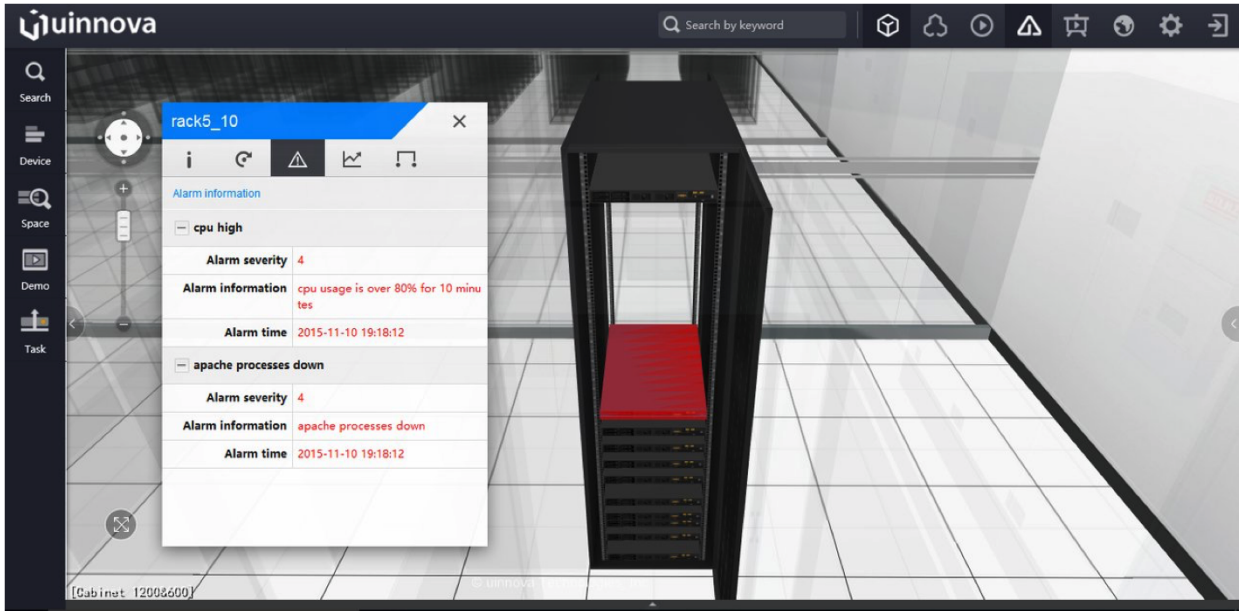
Table 2.1: Usage

Arguments	Value
arg1	Scene name
arg2	—
arg3	Device ID
arg4	Event Title
arg5	OPEN/CLOSED
arg6	Severity (1 ~6) 1 = highest
arg7	Event Body
arg8	Occurrence
arg9	Modify time
arg10	—
arg11	—

Example

```
event.bat ecc _ rack5_10 "cpu high" OPEN 4 "cpu usage is over 80% for 10
minutes" 1447211892000 1447211892000 _ _ event.bat ecc _ rack18_11 "apache
processes down" OPEN 2 "apache processes down" 1447211892000 1447211892000 _ _
```

Screen Capture



2.2.4 Send Performance Data

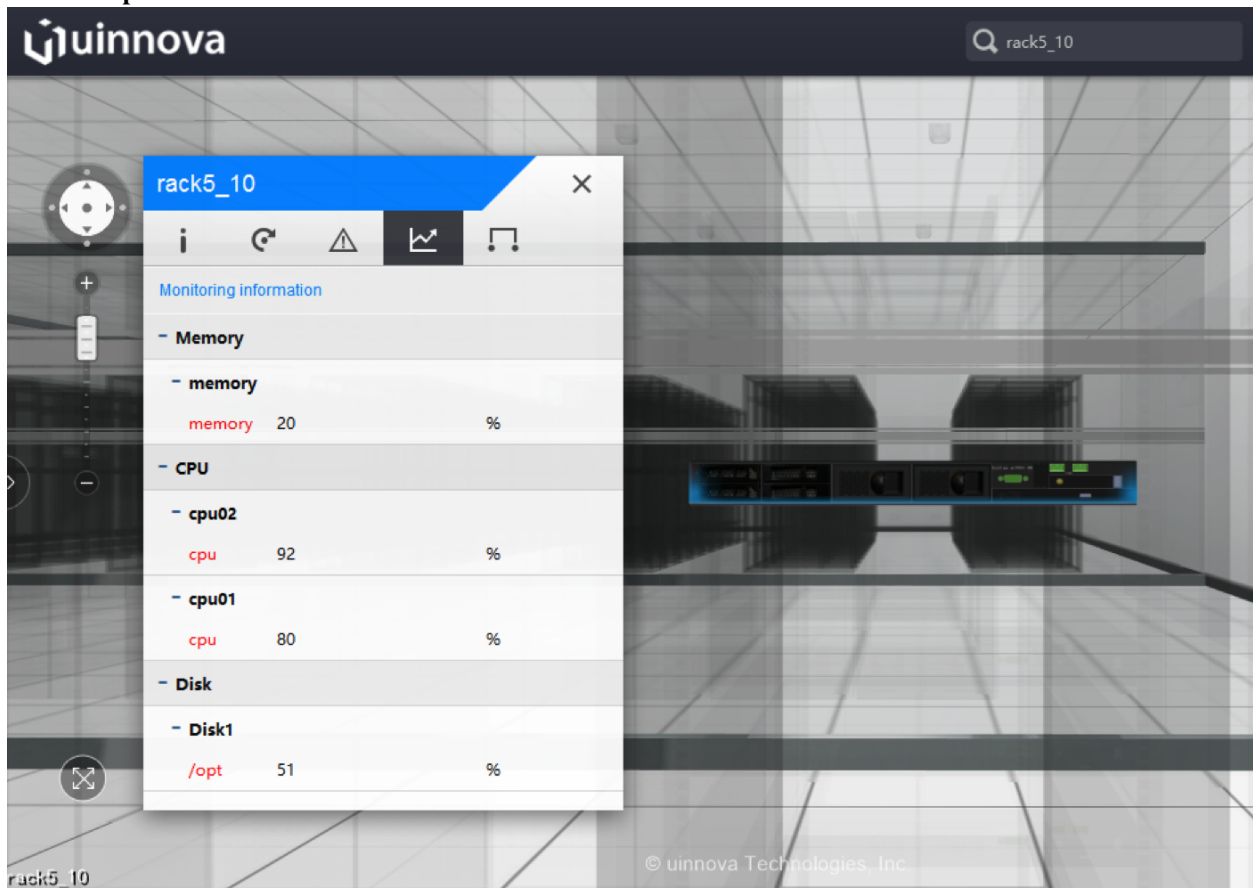
Usage

Table 2.2: Usage

Arguments	Value
arg1	Scene name
arg2	–
arg3	Device ID
arg4	Index group
arg5	instance
arg6	Index name
arg7	[numerical state]
arg8	unit
arg9	value
arg10	Occurrence

Example

```
perf.bat ecc _ rack5_10 CPU cpu01 cpu numerical % 80 1447211892000
perf.bat ecc _ rack5_10 Memory memory memory numerical % 20 1447211892000
perf.bat ecc _ rack5_10 Disk Disk1 /opt numerical % 51 1447211892000
```

Screen Capture

2.2.5 Add Asset to Cabinet

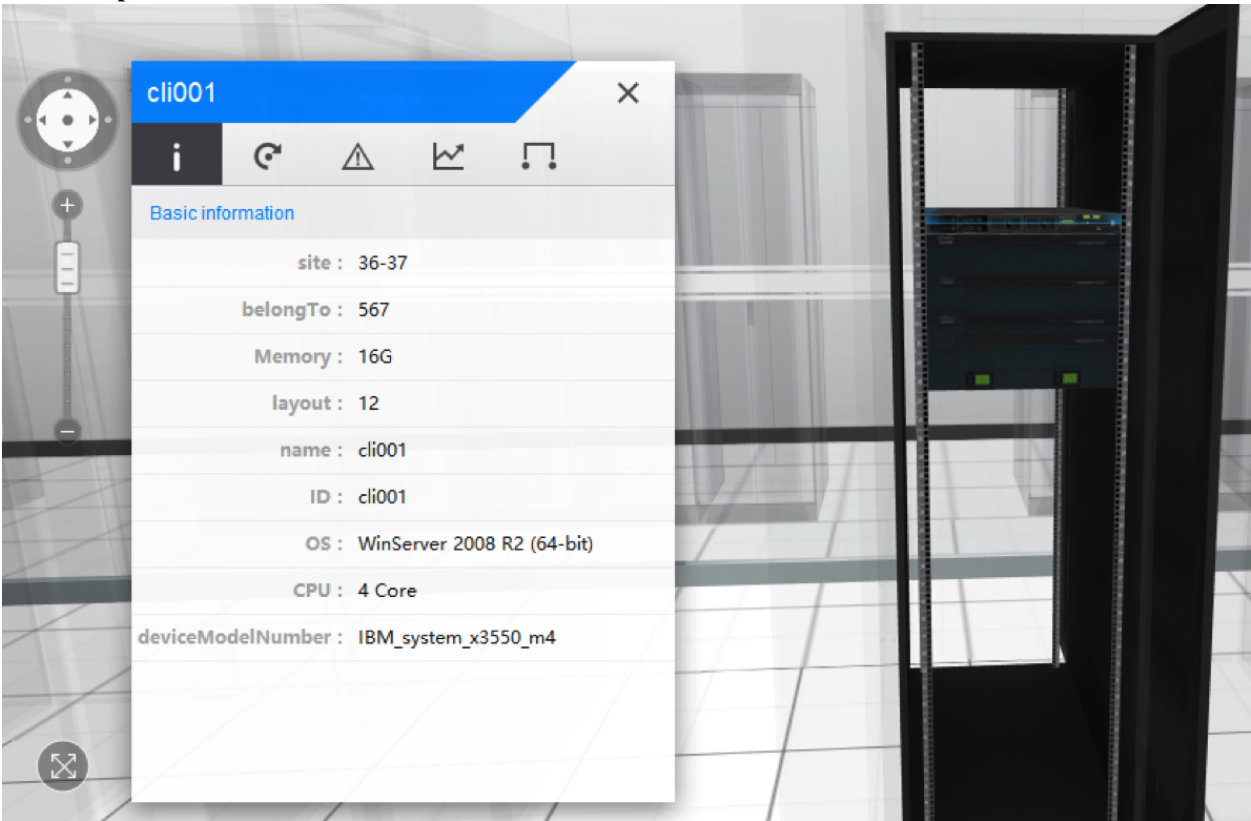
Table 2.3: Usage

Arguments	Value
arg1	Add, update or delete [addl updl del]
arg2	scene
arg3	classID such as rackDevice
arg4	Asset ID
arg5	Asset properties in JSON format, see example below

Example

```
asset.bat upd ecc rackDevice cli001 '{"ID':'cli001','name':'cli001','belongTo':'567','CPU':'4 Core',
```

Screen Capture



2.3 Web Service API

2.3.1 Summary

uDCV provides web service API to manipulate assets, performance and event data. A typical web service invoke in Java as following :

```

1 import org.apache.cxf.endpoint.Client;
2 import org.apache.cxf.jaxws.endpoint.dynamic.JaxWsDynamicClientFactory;
3
4 /**
5  * @Title: callService
6  * @Description:
7  * @returnClient
8  */
9
10 public static Client callService() {
11     JaxWsDynamicClientFactory dcf = JaxWsDynamicClientFactory.newInstance();
12     Client client = dcf.createClient("http://localhost:8080/uinv_dev/services/DataStream?wsdl");
13     return client;
14 }
15
16 public static void main(String[] args) {
17     String pushData = "[{"scene":"demo","key":"","id":"ds003Server","app":"seltImport","inst":"_","p
18     Object[] res = null;
19     try {
20         res = client.invoke("pushMonitor", pushData);
21     } catch (Exception e) {
22         e.printStackTrace();
23     }
24     return (String) res[0];
25 }

```

2.3.2 Performance Service

Table 2.4: Usage

Arguments	Value
URL	http://localhost:8080/uinv_dev/services/DataStream?wsdl
Function Name	pushMonitor
Input Parameter	performance data in JSON format

Example

```

1 public static void main(String[] args) {
2     String pushData = "[{"scene":"demo","key":"","id":"ds003Server","app":"seltImport","inst":"_","p
3     Object[] res = null;
4     try {
5         res = client.invoke("pushMonitor", pushData);
6     } catch (Exception e) {
7         e.printStackTrace();
8     }
9     return (String) res[0];
10 }

```

2.3.3 Monitoring Service

Table 2.5: Usage

Arguments	Value
URL	http://localhost:8080/uinv_dev/services/DataStream?wsdl
Function Name	pushAlarm
Input Parameter	alarm data in JSON format

Example

```
1 String pushData= "[{"scene":"demo","key":"","id":"d03server","title":"alarm title","status":"OPEN","s
2
3 Object[] res = null;
4
5 try {
6     res = client.invoke("pushAlarm", pushData);
7 } catch (Exception e) {
8     e.printStackTrace();
9 }
10     return (String) res[0];
11 }
```

2.3.4 Asset Service

Add Asset

Table 2.6: addRackEquipment

Arguments	Value
URL	http://localhost:8080/uinv_dev/services/DataStream?wsdl
Function Name	addRackEquipment
Input Parameter	data in JSON format

```
1 public static void main(String[] args) {
2     String pushData= "{\"type\":\"rackDevice\",\"BizID\":\"FX10023100234\",\"Name\" : \"FX1002310023
3     Object[] res = null;
4     try {
5         res = client.invoke("addRackEquipment", pushData);
6     } catch (Exception e) {
7         e.printStackTrace();
8     }
9     return (String) res[0];
10 }
```

Update Asset

Table 2.7: updateRackEquipment

Arguments	Value
URL	http://localhost:8080/uinv_dev/services/DataStream?wsdl
Function Name	updateRackEquipment
Input Parameter	data in JSON format

```

1 public static void main(String[] args) {
2     String pushData= "{\"type\":\"rackDevice\",\"BizID\":\"FX10023100234\",\"Name\" : \"FX10023100234\"}";
3     Object[] res = null;
4     try {
5         res = client.invoke("updateRackEquipment", pushData);
6     } catch (Exception e) {
7         e.printStackTrace();
8     }
9     return (String) res[0];
10 }

```

Delete Asset

Table 2.8: deleteRackEquipment

Arguments	Value
URL	http://localhost:8080/uinv_dev/services/DataStream?wsdl
Function Name	deleteRackEquipment
Input Parameter	data in JSON format

```

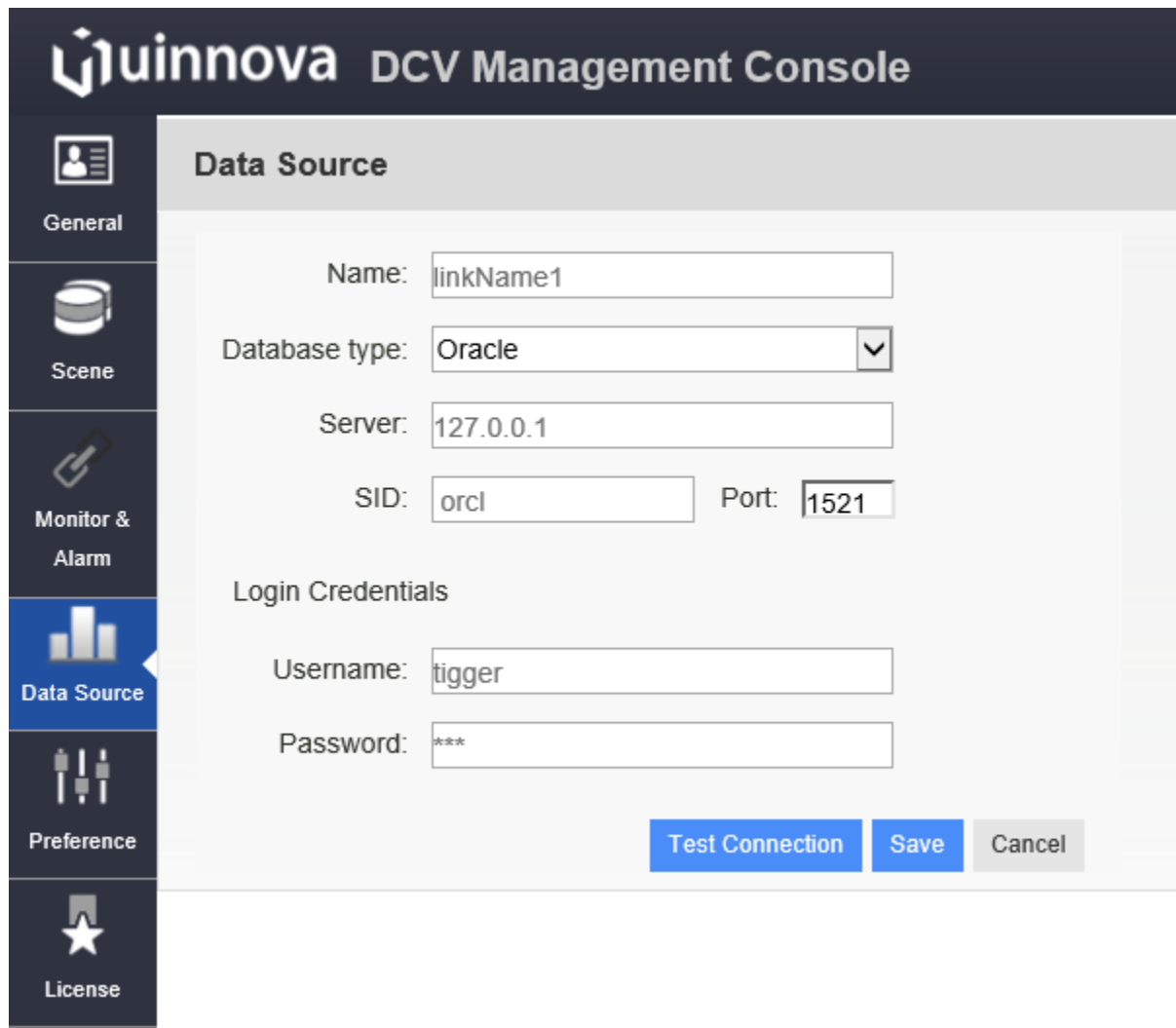
1 public static void main(String[] args) {
2     String pushData= "[\"FX10023100234\",\"FX10023100244\"]";
3     Object[] res = null;
4     try {
5         res = client.invoke("deleteRackEquipment", pushData);
6     } catch (Exception e) {
7         e.printStackTrace();
8     }
9     return (String) res[0];
10 }

```

2.4 JDBC API

uDCV can connect and fetch data from external database using JDBC, to setup database connection, configure field mapping and schedule data fetch job, please go to management console --> data source menu in left-hand side navigation bar.

Define Datasource



The screenshot shows the iQuinova DCV Management Console interface. On the left is a vertical sidebar with icons and labels for 'General', 'Scene', 'Monitor & Alarm', 'Data Source' (which is highlighted in blue), 'Preference', and 'License'. The main area is titled 'Data Source' and contains the following fields:

- Name:
- Database type: (with a dropdown arrow)
- Server:
- SID: Port:
- Login Credentials section:
 - Username:
 - Password:

At the bottom right of the form are three buttons: 'Test Connection' (blue), 'Save' (blue), and 'Cancel' (gray).

2.5 Message Queue API

2.5.1 Summary

uDCV leverage Apache Active MQ <http://activemq.apache.org> , as Message API, which:

Table 2.9: MQ

Queue Name	Usage
asset	Asset Data
perf	Performance Data
event	Event Data

Note:

- The data format of Message body is JSON.
- Default MQ listening port is 61616

2.5.2 asset

Table 2.10: **asset**

Name	Value	Type
<code>_operation_</code>	addlupdlidel	String
<code>_sc_</code>	scene name	String
<code>_pool_</code>	catalog	String
<code>_id_</code>	primary key	String
<code>_data_</code>	asset properties	JSON

Example

```

1  [
2    {
3      "_operation_": "upd",
4      "_sc_": "t1",
5      "_pool_": "rackDevice",
6      "_id_": "dtz",
7      "_data_": {
8        "ID": "dtz",
9        "name": "VM-DB2",
10       "belongTo": "P311-A1-01",
11       "CPU": "4",
12       "Memory": "4096111M",
13       "OS": "WinServer 2008 R2 (64-bit)",
14       "site": "23-25",
15       "layout": "12",
16       "deviceType": "IBMSystem"
17     }
18   }
19 ]

```

2.5.3 perf

Table 2.11: **perf**

Name	Value	Type
scene	scene name	String
id	CI id	String
app	Performance group name	String
inst	Performance instance name	String
param	Performance name	String
type	numericalstatus	String
unit	Performance unit	String
val	Performance value	String or Double

Example

```

1  [
2    {
3      "scene": "demo",
4      "key": "",
5      "id": "ds003Server",
6      "app": "selfImport",
7      "inst": "_",

```

```
8     "param": "temperature",
9     "type": "numerical",
10    "unit": "%",
11    "val": "58",
12    "time": 1431405196437
13  },
14  {
15    "scene": "demo",
16    "key": "",
17    "id": "ds002Server",
18    "app": "seltImport",
19    "inst": "_",
20    "param": "temperature",
21    "type": "status",
22    "unit": "%",
23    "val": "xxx58",
24    "time": 1431405196437
25  }
26 ]
```

2.5.4 event

Table 2.12: perf

Name	Value	Type
scene	scene name	String
id	CI id	String
title	Event title	String
status	open/close	String
severity	1 2 3 4 5 6	String
msg	Event body	String
time	Event occurrence time	Long
modifyTime	Event modify time	Long
arg1	Extended field (optional)	String
arg2	Extended field (optional)	String

Example

```
1 [
2   {
3     "scene": "demo",
4     "key": "",
5     "id": "d03server",
6     "title": "CPU High",
7     "status": "OPEN",
8     "severity": "1",
9     "msg": "CPU usage is over 90% over 5 minutes",
10    "time": 1431405196437,
11    "modifyTime": 1431405196437,
12    "arg1": "",
13    "arg2": ""
14  },
15  {
16    "scene": "demo",
17    "key": "",
18    "id": "d03server",
```



```
19     "title": "Disk Full",
20     "status": "CLOSED",
21     "severity": "1",
22     "msg": "Disk usage is over 70%",
23     "time": 1431405196437,
24     "modifyTime": 1431405196437,
25     "arg1": "",
26     "arg2": ""
27 }
28 ]
```