



**Document Number: DSP2027**

**Date: 2012-09-12**

**Version: 1.0.1**

# **Cloud Infrastructure Management Interface (CIMI) Primer**

**Document Type: White Paper**

**Document Status: DMTF Informational**

**Document Language: en-US**

**Copyright Notice**

Copyright © 2012 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit <http://www.dmtf.org/about/policies/disclosures.php>.

## **Contents**

<b>Contents</b> .....	<b>3</b>
<b>Overview</b> .....	<b>6</b>
<b>Scenario 1 : Creating a New Machine</b> .....	<b>7</b>
Step 1 : Retrieve the CEP.....	7
Step 2 : Retrieve the list of Machine Images .....	7
Step 3 : Retrieve the list of Machine Configurations .....	8
Step 4 : Create a new Credential Resource .....	9
Step 5 : Create a new Machine .....	10
Step 6 : Query new Machine .....	11
Step 7 : Start a Machine .....	12
Step 8 : Stop a Machine .....	12
Step 9 : Update a Machine's attributes.....	12
<b>Scenario 2 : Adding a New Volume to a Machine</b> .....	<b>14</b>
Step 1 : Obtain the Machine URL .....	14
Step 2 : Retrieve the CEP.....	14
Step 3 : Get the list of VolumeConfigurations to determine which to use .....	14
Step 4 : Create a new Volume.....	15
Step 5 : Retrieve the Volume Information.....	15
Step 6 : Retrieve the Machine's volume collection .....	16
Step 7 : Connect the new Volume to a Machine .....	17
Step 8 : Query the Machine's volume collection to verify the update .....	18
<b>Scenario 3 : Defining and Using Machine Templates</b> .....	<b>19</b>
Step 1 : Retrieve the CEP.....	19
Step 2 : Create a new Machine Template .....	19
Step 3 : Create a new Machine by using a Machine Template .....	20
<b>Scenario 4 : Creating a New Machine from an existing Volume</b> .....	<b>21</b>
Step 1 : Retrieve the CEP.....	21
Step 2 : Get the list of Volumes to determine which to use .....	21
Step 3 : Create a new Machine .....	22
Step 4 : Query new Machine .....	22
<b>Scenario 5 : Defining and Using System Templates</b> .....	<b>24</b>
Step 1 : Retrieve the CEP.....	24
Step 2 : Create a new System Template.....	24
Step 3 : Create a new System by using a System Template .....	25
Step 4 : Query the new System.....	26
<b>Scenario 6 : Editing System Templates</b> .....	<b>27</b>
Step 1 : Edit an existing System Template.....	27
Step 2 : Create a new System using a System Template .....	29
Step 3 : Query the new System .....	29
<b>Change log:</b> .....	<b>30</b>

## Foreword

This document contains scenarios that describe common uses of the CIMI protocol.

### Acknowledgments

The authors wish to acknowledge the following people.

#### Editors:

- Davis, Doug - IBM
- Pilz, Gilbert - Oracle
- Zhang, Aaron - Huawei

#### Contributors:

- Ali, Ghazanfar - ZTE Corporation
- Andreou, Marios - Red Hat
- Bankston, Keith - Microsoft Corporation
- Bumpus, Winston - VMware Inc.
- Burkhart, Nathan - Microsoft Corporation
- Carlson, Mark - Oracle
- Carter, Steve - Novell
- Chu, Junsheng - ZTE Corporation
- Cohen, Josh - Microsoft Corporation
- Coleman, Derek - Hewlett-Packard Company
- Crandall, John - Brocade Communications Systems
- Davis, Doug - IBM
- Davis, Jim - WBEM Solutions
- de la Iglesia, Fernando - Telefónica
- Dempo, Hiroshi - NEC Corporation
- Durand, Jacques - Fujitsu
- Edery, Yigal - Microsoft Corporation
- Ericson, George - EMC
- Evans, Colleen - Microsoft Corporation
- Floeren, Norbert - Ericsson AB
- Freund, Robert - Hitachi, Ltd.
- Galán, Fermín - Telefónica
- Gopalan, Krishnan - Microsoft Corporation
- Iwasa, Kazunori - Fujitsu
- Johnson, Mark - IBM
- Khasnabish, Bhumi - ZTE Corporation
- Kowalski, Vincent - BMC Software
- Krishnaswamy, Ruby - France Telecom Group
- Lamers, Lawrence - VMware Inc.
- Lipton, Paul - CA Technologies
- Livingston, James - NEC Corporation
- Lubsey, Vince - Virtustream Inc.
- Lutterkort, David - Red Hat
- Maciel, Fred - Hitachi, Ltd.
- Maier, Andreas - IBM
- Malhotra, Ashok - Oracle
- Mischkinsky, Jeff - Oracle
- Molina, Jesus - Fujitsu
- Moscovich, Efraim - CA Technologies

- Murray, Bryan - Hewlett-Packard Company
- Neely, Steven - Cisco
- Ogawa, Ryuichi - NEC Corporation
- Parchem, John - Microsoft Corporation
- Pardikar, Shishir - Citrix Systems Inc.
- Peñalvo, Miguel - Telefónica
- Pilz, Gilbert - Oracle
- Polo, Alvaro - Telefónica
- Ronco, Enrico - Telecom Italia
- Rossini, Federico - Telecom Italia
- Rutkowski, Matthew - IBM
- Rutt, Tom - Fujitsu
- Shah, Hemal - Broadcom
- Shah, Nihar - Microsoft Corporation
- Sill, Alan - Texas Tech University
- Song, Zhexuan - Huawei
- Waschke, Marvin - CA Technologies
- Wells, Eric - Hitachi, Ltd.
- Wheeler, Jeff - Huawei
- Wiggers, Maarten - Fujitsu
- Winkler, Steve - SAP AG
- Yu, Jack - Oracle
- Zhang, Aaron - Huawei
- Zhang, HengLiang - Huawei

**Overview**

For the sake of simplicity, in each of the following scenarios the Cloud Provider only supports the minimum features needed to demonstrate the features highlighted by each scenario. Therefore, the results of the query to the Cloud Entry Point (CEP) to retrieve the list of supported features will be customized for each scenario. Additionally the URI of the Cloud Entry Point is assumed to be <http://example.com/CEP>.

## Scenario 1 : Creating a New Machine

This scenario will create a new Machine. The new Machine's configuration will be based on existing configurations and images offered by the provider. However, a new Credential resource (userid & password) will be created.

### Step 1 : Retrieve the CEP

The CEP will provide the links to the set of resources that are available in this Cloud. You retrieve the CEP to discover the URL to each collection:

```
GET / HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
  "id": "http://example.com/CEP",
  "baseURI": "http://example.com/",
  "resourceMetadata": { "href": "http://example.com/resourceMetadata" },
  "machines": { "href": "http://example.com/machines" },
  "machineConfigs": { "href": "http://example.com/machineConfigs" },
  "machineImages": { "href": "http://example.com/machineImages" },
  "credentials": { "href": "http://example.com/credentials" }
}
```

### Step 2 : Retrieve the list of Machine Images

Before you can create a new Machine, first decide what kind of operating system and/or software you want to have pre-installed. The Machine Images collection is the set of Machine Images that this Cloud offers - note that some may be predefined by the Cloud while some may be user-created:

```
GET /machineImages HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImageCollection",
  "id": "http://example.com/machineImages",
  "machineImages": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
      "id": "http://example.com/images/WinXP-SP2",
      "name": "WinXP SP2",
      "description": "Windows XP with Service Pack 2",
      "created": "2012-01-01T12:00:00Z",
      "updated": "2012-01-01T12:00:00Z",
      "imageLocation": { "href": "http://example.com/data/8934322" }
    },
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
      "id": "http://example.com/images/Win7",
      "name": "Windows 7",
      "description": "Windows 7",
      "created": "2012-01-01T12:00:00Z",
      "updated": "2012-01-01T12:00:00Z",
      "imageLocation": { "href": "http://example.com/data/8934344" }
    },
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
      "id": "http://example.com/images/Linux-SUSE",
```

```

    "name": "Linux SUSE",
    "description": "Linux SUSE",
    "created": "2012-01-01T12:00:00Z",
    "updated": "2012-01-01T12:00:00Z",
    "imageLocation": { "href": "http://example.com/data/8934311" }
  }
]
}

```

### Step 2.1 : Choose a Machine Image

Next examine each Machine Image to find one that meets your needs. The first one is acceptable, so you will use it later.

It is worth noting that if you knew you wanted to use the first item in the list and only wanted to see that one resource in the previous query then the following could have been done instead:

```
GET /machineImages?$last=1 HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImageCollection",
  "id": "http://example.com/machineImages",
  "machineImages": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
      "id": "http://example.com/images/WinXP-SP2",
      "name": "WinXP SP2",
      "description": "Windows XP with Service Pack 2",
      "created": "2012-01-01T12:00:00Z",
      "updated": "2012-01-01T12:00:00Z",
      "imageLocation": { "href": "http://example.com/data/8934322" }
    }
  ]
}

```

Note that you didn't need to specify \$first=1 in this case since "1" is its default value.

### Step 3 : Retrieve the list of Machine Configurations

Next you decide what kind of virtual hardware you want to install your Machine Image onto. As with determining the kind of Machine Image you want, first ask for the list of available Machine Configurations:

```
GET /machineConfigs HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI":
  "http://schemas.dmtf.org/cimi/1/MachineConfigurationCollection",
  "id": "http://example.com/machineConfigs",
  "machineConfigurations": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
      "id": "http://example.com/configs/tiny",
      "name": "tiny",
      "description": "a teenie tiny one",
      "created": "2012-01-01T12:00:00Z",
      "updated": "2012-01-01T12:00:00Z",
      "cpu": 1,

```



```

    "memory": 4000000,
    "disks" : [
      { "capacity": 50000000 }
    ]
  },

  { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
    "id": "http://example.com/configs/small",
    "name": "small",
    "description": "a small sized one",
    "created": "2012-01-01T12:00:00Z",
    "updated": "2012-01-01T12:00:00Z",
    "cpu": 1,
    "memory": 8000000,
    "disks" : [
      { "capacity": 500000000 }
    ]
  },

  { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
    "id": "http://example.com/configs/medium",
    "name": "medium",
    "description": "a medium one",
    "created": "2012-01-01T12:00:00Z",
    "updated": "2012-01-01T12:00:00Z",
    "cpu": 1,
    "memory": 16000000,
    "disks" : [
      { "capacity": 1000000000 },
      { "capacity": 1000000000 },
    ]
  }
]
}

```

### Step 3.1 : Choose a Machine Configuration

Next examine the returned list and pick a Machine Configuration that suits your needs. The first one is acceptable, so you will use it later.

### Step 4 : Create a new Credential Resource

You want to use your own userid and password for this new Machine, so you need to create a new Credential resource. This process is done by using the POST operation, but first you need to retrieve the Credential collection so that you know where to POST a new Credential resource to:

```
GET /credentials HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialCollection",
  "id": "http://example.com/credentials",
  "operations": [ { "rel": "add", "href": "http://example.com/credentials" } ]
}

```

Notice at this point there are no Credential resources in the environment. Before you can create a new Credential resource, you must first discover this Cloud provider's extension attributes for the Credential resource. By default the CIMI specification does not define how the initial user of a new Machine is

specified; rather it is left open for each Cloud provider to determine how this information should be provided. Clients can discover this information by querying the Credential resource metadata resource. To examine this resource, first look through the ResourceMetadata collection for this Provider's description of the Credential's resource. Start by retrieving the ResourceMetadata collection from the URI referenced in the Cloud Entry Point:

```
GET /resourceMetadata HTTP/1.1
```

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadataCollection",
  "id": "http://example.com/resourceMetadata",
  "resourceMetadatas": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadata",
      "id": "http://example.com/resources/Credential",
      "typeURI": "http://schemas.dmtf.org/cimi/1/Credential",
      "name": "Credential",
      "attributes": [
        { "name": "userID", "namespace": "http://example.com",
          "type": "string", "required": "true" },
        { "name": "password", "namespace": "http://example.com",
          "type": "string", "required": "true" }
      ]
    }
  ]
}
```

Now iterate over the list of resourceMetadata entries in the collection for the one whose "typeURI" is "http://schemas.dmtf.org/cimi/1/Credential". After you find it, you can now examine the extensions this Provider has added to the Credential resource and the above indicates that the Credential resource has been extended and must include two attributes called "userID" and "password". Both are of type "string".

Now create a new Credential resource by using the POST operation:

```
POST /credentials HTTP/1.1
```

```
Content-Type: application/json
```

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialCreate",
  "name": "Default",
  "description": "My Default User",
  "credentialTemplate": {
    "userID": "JoeSmith",
    "password": "letmein"
  }
}
```

```
HTTP/1.1 201 Created
```

```
Location: http://example.com/creds/12345
```

Note: While the "userID" and "password" attributes were discovered via the Credential ResourceMetadata, the "name" and "description" attributes are part of the common set of attributes available on all resources. In a future scenario you will see how the client knew that "userID" and "password" were the proper attribute names for this image type and Cloud provider.

### Step 5 : Create a new Machine

Retrieve the Machines collection so that you know where to POST a new Machine to:

```
GET /machines HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCollection",
  "id": "http://example.com/machines",
  "operations": [ { "rel": "add", "href": "http://example.com/machines" } ]
}

```

Now create a new one:

```

POST /machines HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate",
  "name": "myMachine1",
  "description": "My very first machine",
  "machineTemplate": {
    "machineConfig": { "href": " http://example.com/configs/tiny" },
    "machineImage": { "href": " http://example.com/images/WinXP-SP2" },
    "credential": { "href": "http://example.com/creds/12345" }
  }
}

```

```

HTTP/1.1 201 Created
Location: http://example.com/machines/843752

```

### Step 6 : Query new Machine

Retrieve the Machine to get the full representation of the new Machine:

```
GET /machines/843752 HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
  "id": "http://example.com/machines/843752",
  "name": "myMachine1",
  "description": "My very first machine",
  "created": "2012-08-15T12:15:00Z",
  "updated": "2012-08-15T12:15:00Z",
  "state": "STOPPED",
  "cpu": 1,
  "memory": 4000000,
  "disks" : { "href": "http://example.com/machines/843752/disks",
  "networkInterfaces": { "href": "http://example.com/machines/843752/NIs",
  "operations": [
    { "rel": "edit", "href": "http://example.com/machines/843752" },
    { "rel": "delete", "href": "http://example.com/machines/843752" },
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start",
      "href": "http://example.com/machines/843752" }
  ]
}
}

```

Notice the "state" attribute on the Machine is "STOPPED", which is the initial state of a new machine.

**Step 7 : Start a Machine**

The presence of the "start" operation in the "operations" array of the Machine representation indicates not only which URI to POST the "start" operation to, but that you are able to do it at this time.

```
POST /machines/843752 HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/start"
}
```

```
HTTP/1.1 204 No Content
```

**Step 7.1 : Query a Machine to verify it is started**

Query the Machine again to verify that it is started:

```
GET /machines/843752 HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
  "id": "http://example.com/machines/843752",
  "name": "myMachine1",
  "description": "My very first machine",
  "created": "2012-08-15T12:15:00Z",
  "updated": "2012-08-15T12:15:00Z",
  "state": "STARTED",
  "cpu": 1,
  "memory": 4000000,
  "disks" : { "href": "http://example.com/machines/843752/disks",
  "networkInterfaces": { "href": "http://example.com/machines/843752/NIs",
  "operations": [
    { "rel": "edit", "href": "http://example.com/machines/843752" },
    { "rel": "delete", "href": "http://example.com/machines/843752" },
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop",
      "href": "http://example.com/machines/843752" }
  ]
}
```

Notice the "state" attribute on the Machine is "STARTED" and that the "operations" array no longer indicates that the "start" operation is available; rather the "stop" operation is available now instead.

**Step 8 : Stop a Machine**

Using the "stop" operation's URL, you can now ask for the Machine to be stopped:

```
POST /machines/843752 HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/stop"
}
```

```
HTTP/1.1 204 No Content
```

**Step 9 : Update a Machine's attributes**

Using PUT operation on the "edit" operation's URL, you can update some of the attribute of the Machine, for example the "name" and "description":

```
PUT /machines/843752?$select=name,description HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
  "name" : "Cool Demo #1"
}
```

```
HTTP/1.1 200 OK
```

```
{ "name" : "Cool Demo #1" }
```

Notice that URL of the "edit" operation has been modified to indicate which attributes are being updated; only those attributes will be touched. Because the URL includes the "description" attribute but the HTTP request body does not, that attribute is erased.

## Scenario 2 : Adding a New Volume to a Machine

This scenario will create a new Volume and connect it to an existing Machine.

### Step 1 : Obtain the Machine URL

Machine:

```
http://example.com/machines/843752
```

### Step 2 : Retrieve the CEP

The CEP will provide the links to the set of resources that are available in this Cloud. Retrieve the CEP to discover the URL to each collection:

```
GET / HTTP/1.1 HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
  "id": "http://example.com/CEP",
  "baseURI": "http://example.com/",
  "machines": { "href": "http://example.com/machines" },
  "machineConfigs": { "href": "http://example.com/machineConfigs" },
  "machineImages": { "href": "http://example.com/machineImages" },
  "credentials": { "href": "http://example.com/credentials" },
  "volumes": { "href": "http://example.com/volumes" },
  "volumeConfigs": { "href": "http://example.com/vConfigs" }
}
```

### Step 3 : Get the list of VolumeConfigurations to determine which to use

When you create a new Volume, you need to decide what kind of Volume to create, e.g., its size, format, etc. The Volume Configurations collection is the set of predefined Volume Configurations that this Cloud offers:

```
GET /vConfigs HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI":
  "http://schemas.dmtf.org/cimi/1/VolumeConfigurationCollection",
  "id": "http://example.com/volumeConfigs",
  "volumeConfigurations": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
      "id": "http://example.com/vConfigs/small",
      "name": "Small",
      "description": "A pretty small one",
      "created": "2012-08-15T12:15:00Z",
      "updated": "2012-08-15T12:15:00Z",
      "type": "http://schemas.dmtf.org/cimi/1/mapped",
      "format": "NTFS",
      "capacity": 60000000
    },
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
      "id": "http://example.com/vConfigs/medium",
      "name": "Small",
      "description": "A medium sized one",
      "created": "2012-08-15T12:15:00Z",
```

```

    "updated": "2012-08-15T12:15:00Z",
    "type": "http://schemas.dmtf.org/cimi/1/mapped",
    "format": "NTFS",
    "capacity": 500000000
  },
  {
    "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
    "id": "http://example.com/vConfigs/large",
    "name": "Large",
    "description": "A large one",
    "created": "2012-08-15T12:15:00Z",
    "updated": "2012-08-15T12:15:00Z",
    "type": "http://schemas.dmtf.org/cimi/1/mapped",
    "format": "NTFS",
    "capacity": 1000000000
  }
]
}

```

### Step 3.1 : Choose a Volume Configuration

Next examine each Volume Configuration to find the one that meets your needs. The first one is acceptable, so you will use it later.

### Step 4 : Create a new Volume

Retrieve the Volumes collection so that you know where to POST a new Volume to:

```
GET /volumes HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeCollection",
  "id": "http://example.com/volumes",
  "operations": [ { "rel": "add", "href": "http://example.com/volumes" } ]
}

```

Now create a new one:

```

POST /volumes HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeCreate",
  "name": "myVolume1",
  "description": "My first new volume",
  "volumeTemplate": {
    "volumeConfig": { "href": "http://example.com/vConfigs/small" }
  }
}

```

```

HTTP/1.1 201 Created
Location: http://example.com/volumes/35782

```

### Step 5 : Retrieve the Volume Information

To verify that the Volume you created and connected to the Machine is what you wanted, follow the reference that was returned from the previous step:

```
GET /volumes/35782 HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Volume",
  "id": "http://example.com/volumes/35782",
  "name": "myVolume1",
  "created": "2012-09-15T12:15:00Z",
  "updated": "2012-09-15T12:15:00Z",
  "description": "My first new volume",
  "type": "http://schemas.dmtf.org/cimi/1/mapped",
  "capacity": 60000000,
  "operations": [
    { "rel": "edit", "href": "http://example.com/volumes/35782" },
    { "rel": "delete", "href": "http://example.com/volumes/35782" }
  ]
}

```

### Step 6 : Retrieve the Machine's volume collection

Before you can connect this new Volume to your Machine, you first need to retrieve the Machine's Volume collection to know where to send your request to. First retrieve the Machine to get the reference to the collection:

```
GET /machines/843752 HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
  "id": "http://example.com/machines/843752",
  "name": "myMachine1",
  "description": "My very first machine",
  "created": "2012-08-15T12:15:00Z",
  "updated": "2012-08-15T12:15:00Z",
  "state": "STARTED",
  "cpu": 1,
  "memory": 4000000,
  "disks": { "href": "http://example.com/machines/843752/disks",
  "volumes": { "href": "http://example.com/machines/843752/volumes" },
  "networkInterfaces": { "href": "http://example.com/machines/843752/NIs" },
  "operations": [
    { "rel": "edit", "href": "http://example.com/machines/843752" },
    { "rel": "delete", "href": "http://example.com/machines/843752" },
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop",
      "href": "http://example.com/machines/843752" }
  ]
}

```

Note that in the previous scenario, the "volumes" attribute was not present due to the limited scope of that scenario; however, now the "volumes" attribute appears because the scenario (and features of our sample Provider) are expanded to include support for Volumes.

Now retrieve the Volume collection:

```
GET /machines/843752/volumes HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumeCollection",

```



```

    "id": "http://example.com/machines/843752/volumes",
    "operations": [
      { "rel": "add", "href": "http://example.com/machines/843752/volumes" }
    ]
  }
}

```

Note that there are no Volumes currently connected to this Machine.

Alternatively, as an optimization, this collection could have been retrieved at the same time as the original Machine by using the \$expand query parameter:

```
GET /machines/843752?$expand=volumes HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
  "id": "http://example.com/machines/843752",
  "name": "myMachine1",
  "description": "My very first machine",
  "created": "2012-08-15T12:15:00Z",
  "updated": "2012-08-15T12:15:00Z",
  "state": "STARTED",
  "cpu": 1,
  "memory": 4000000,
  "disks" : { "href": "http://example.com/machines/843752/disks",
  "volumes": {
    "href": "http://example.com/machines/843752/volumes",
    "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumeCollection",
    "id": "http://example.com/machines/843752/volumes",
    "operations": [
      { "rel": "add", "href": "http://example.com/machines/843752/volumes" }
    ]
  }
},
  "networkInterfaces": { "href": "http://example.com/machines/843752/NIs" },
  "operations": [
    { "rel": "edit", "href": "http://example.com/machines/843752" },
    { "rel": "delete", "href": "http://example.com/machines/843752" },
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop",
      "href": "http://example.com/machines/843752" }
  ]
}

```

### Step 7 : Connect the new Volume to a Machine

You connect the Volume to the Machine by using the "add" operation on the Volume collection and pass in a new MachineVolume resource:

```

POST /machines/843752/volumes HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolume",
  "initialLocation": "V",
  "volume": { "href": "http://example.com/volumes/35782" }
}

```

```

HTTP/1.1 201 Created
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolume",

```

```

    "id": "http://example.com/machines/843752/volumes/1",
    "initialLocation": "V",
    "volume": { "href": "http://example.com/volumes/35782" },
    "operations": [
      { "rel": "edit", "href": "http://example.com/machines/843752/volumes/1"},
      { "rel": "delete", "href": "http://example.com/machines/843752/volumes/1"}
    ]
  }
}

```

### Step 8 : Query the Machine's volume collection to verify the update

Retrieve the Machine's volume collection to get the complete list of Volumes and use the list to verify the update was successful:

```
GET /machines/843752/volumes HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumeCollection",
  "id": "http://example.com/machines/843752/volumes",
  "machineVolumes": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumes",
      "id": "http://example.com/machines/843752/volumes/1",
      "initialLocation": "V",
      "volume": { "href": "http://example.com/volumes/35782" },
      "operations": [
        { "rel": "edit", "href": "http://example.com/machines/843752/volumes/1"},
        { "rel": "delete", "href": "http://example.com/machines/843752/volumes/1"}
      ]
    },
    { "rel": "add", "href": "http://example.com/machines/843752/volumes" }
  ]
}

```

## Scenario 3 : Defining and Using Machine Templates

This scenario will create a new Machine Template that will then be used to create a new Machine. Machine Templates are convenience resources that allow for well-defined descriptions (configuration, image, etc.) of a Machine to be persisted such that it can be reused later. This feature is particularly useful when the user of the new Machine may not be technically savvy enough to know all of the details necessary to create the Machine. Commonly, Machine Templates are created for demos, or complex configurations, where a particular Machine Image must be used on a particular Machine Configuration. Machine Templates allow this information to be persisted and easily reused.

For convenience, reuse the configuration information already obtained in the previous scenarios.

### Step 1 : Retrieve the CEP

The CEP will provide the links to the set of resources that are available in this Cloud. Retrieve the CEP to discover the URL to each collection:

```
GET / HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
  "id": "http://example.com/CEP",
  "baseURI": "http://example.com/",
  "machines": { "href": "http://example.com/machines" },
  "machineTemplates": { "href": "http://example.com/machineTemplates" },
  "machineConfigs": { "href": "http://example.com/machineConfigs" },
  "machineImages": { "href": "http://example.com/machineImages" },
  "credentials": { "href": "http://example.com/credentials" }
}
```

### Step 2 : Create a new Machine Template

From the previous scenarios, you already have the MachineConfiguration, MachineImage, and Credential resources that you will reuse for this MachineTemplate:

MachineImage:

```
http://example.com/images/WinXP-SP2
```

MachineConfiguration:

```
http://example.com/configs/tiny
```

Credential:

```
http://example.com/creds/12345
```

Before you can create the new MachineTemplate, you first need to determine the URL to which the POST is sent. This location is obtained from the MachineTemplate collection URL that was returned as part of the CEP:

```
GET /machineTemplates HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplateCollection",
  "id": "http://example.com/machineTemplates",
  "operations": [
    { "rel": "add", "href": "http://example.com/machineTemplates" }
  ]
}
```

```
}
```

Note that there are no MachineTemplates in the environment right now.

Now create the new MachineTemplate resource:

```
POST /machineTemplates HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplate",
  "name": "Demo1",
  "description": "My first demo",
  "machineConfig": { "href": "http://example.com/configs/tiny" },
  "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
  "credential": { "href": "http://example.com/creds/12345" }
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/machineTemplates/82754
```

### **Step 3 : Create a new Machine by using a Machine Template**

---

Now create a new Machine by using this Machine Template:

```
POST /machines HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate",
  "name": "myMachine2",
  "description": "My second machine",
  "machineTemplate": { "href": "http://example.com/machineTemplates/82754" }
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/machines/843799
```

## **Scenario 4 : Creating a New Machine from an existing Volume**

This scenario will create a new Machine that boots from an existing Volume. This simple example assumes that the user knows that there is an existing Volume with the bootable property equal to true.

### **Step 1 : Retrieve the CEP**

The CEP will provide the links to the set of resources that are available in this Cloud. Retrieve the CEP to discover the URL to each collection:

```
GET / HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceType": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
  "id": "http://example.com/CEP",
  "baseURI": "http://example.com/",
  "resourceMetadata": { "href": "http://example.com/resourceMetadata" },
  "machines": { "href": "http://example.com/machines" },
  "machineConfigs": { "href": "http://example.com/machineConfigs" },
  "machineImages": { "href": "http://example.com/machineImages" },
  "credentials": { "href": "http://example.com/credentials" },
  "volumes": { "href": "http://example.com/volumes" },
  "volumeConfigs": { "href": "http://example.com/vConfigs" }
}
```

### **Step 2 : Get the list of Volumes to determine which to use**

When you create a new Machine from a Volume, you need to decide which Volume to use. The Volume collection is the set of existing Volumes that this Cloud offers:

```
GET /volumes HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceType": "http://schemas.dmtf.org/cimi/1/VolumeCollection",
  "id": "http://example.com/volumes",
  "volumes": [
    { "resourceType": "http://schemas.dmtf.org/cimi/1/Volume",
      "id": "http://example.com/volumes/vol1",
      "name": "Win7-Bootable",
      "created": "2012-08-15T12:15:00Z",
      "updated": "2012-08-15T12:15:00Z",
      "description": "A bootable volume running Windows 7",
      "state": "AVAILABLE",
      "capacity": 60000000,
      "bootable": true
    },
    { "resourceType": "http://schemas.dmtf.org/cimi/1/Volume",
      "id": "http://example.com/volumes/vol2",
      "name": "Generic Volume",
      "created": "2012-08-15T12:15:00Z",
      "updated": "2012-08-15T12:15:00Z",
      "description": "A generic volume for Windows",
      "state": "AVAILABLE",
      "capacity": 60000000,
      "bootable": true
    }
  ]
}
```

```

    }
  ]
}

```

### Step 2.1 : Choose a Volume

Next examine each Volume to find the one that meets your needs. The first one is acceptable, so you will use it later.

### Step 3 : Create a new Machine

Retrieve the Machines collection so you know where to POST a new Machine to:

```
GET /machines HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceType": "http://schemas.dmtf.org/cimi/1/MachineCollection",
  "id": "http://example.com/machines",
  "operations": [
    { "rel": "add", "href": "http://example.com/machines" }
  ]
}

```

Now create a new one, connecting it to the bootable Volume:

```

POST /machines HTTP/1.1
Content-Type: application/json

{ "resourceType": "http://schemas.dmtf.org/cimi/1/MachineCreate",
  "name": "myMachine2",
  "description": "My second machine",
  "machineTemplate": {
    "volumes": [
      { "initialLocation": "V",
        "href": "http://example.com/volumes/vol1" }
    ]
  }
}

```

Note that the MachineTemplate in this case does not specify a MachineImage or MachineConfiguration to use. In this example, for simplicity, you can assume that the Provider has default values for those.

```

HTTP/1.1 201 Created
Location: http://example.com/machines/852108

```

### Step 4 : Query new Machine

Retrieve the Machine to get the full representation of the new Machine:

```
GET /machines/852108 HTTP/1.1
```

```

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceType": "http://schemas.dmtf.org/cimi/1/Machine",
  "id": "http://example.com/machines/852108",
  "name": "myMachine2",
  "description": "My second machine",
  "created": "2012-03-26T10:04:00Z",
  "updated": "2012-03-26T10:04:00Z",
}

```

```
"state": "STOPPED",
"cpu": "1",
"memory": 4000000,
"disks" : { "href": "http://example.com/machines/852108/disks",
"volumes": { "href": "http://example.com/machines/852108/volumes" },
"networkInterfaces": { "href": "http://example.com/machines/852108/NIs",
"operations": [
  { "rel": "edit", "href": "http://example.com/machines/852108" },
  { "rel": "delete", "href": "http://example.com/machines/852108" },
  { "rel": "http://schemas.dmtf.org/cimi/1/action/start",
    "href": "http://example.com/machines/852108" }
]
}
```

Notice the "state" attribute on the Machine is "STOPPED", which is the initial state of a new machine.

## Scenario 5 : Defining and Using System Templates

This scenario will create a new System Template that will then be used to create a new System. System Templates are convenience resources that allow for well-defined descriptions (configuration, image, etc.) of a System to be persisted such that it can be reused later.

### Step 1 : Retrieve the CEP

The CEP will provide the links to the set of resources that are available in this Cloud. Retrieve the CEP to discover the URL to each collection:

```
GET / HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
  "id": "http://example.com/CEP",
  "baseURI": "http://example.com/",
  "systemTemplates": { "href": "http://example.com/systemTemplates" },
  "machineTemplates": { "href": "http://example.com/machineTemplates" },
  "credentialTemplates": { "href": "http://example.com/credentialTemplates" },
  "volumeTemplates": { "href": "http://example.com/volumeTemplates" }
}
```

### Step 2 : Create a new System Template

You will define a SystemTemplate that when instantiated creates a Machine, a Volume connected to the Machine, and a Credential resource. To achieve this configuration, you will include in the SystemTemplate definition a MachineTemplate by value, a VolumeTemplate by reference, and a CredentialTemplate by reference. The VolumeTemplate and CredentialTemplate resources are already available:

VolumeTemplate:

```
http://example.com/volumeTemplates/95839
```

CredentialTemplate:

```
http://example.com/credentialTemplates/72000
```

Note: Alternatively, you could have included the VolumeTemplate and CredentialTemplate by value in the MachineTemplate definition below. However, it is beneficial to immediately see in the SystemTemplate which resources are involved and generally, automatic creation of the credential is more secure.

Before you can create the new SystemTemplate, you first need to determine the URL to which the POST is sent. This location is obtained from the SystemTemplate collection URL that was returned as part of the CEP:

```
GET /systemTemplates HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplateCollection",
  "id": "http://example.com/systemTemplates",
  "operations": [
    { "rel": "add", "href": "http://example.com/systemTemplates" }
  ]
}
```



Now create the new System Template resource:

```
POST /systemTemplates HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
  "name": "System Demo1",
  "description": "My first system template demo",
  "componentDescriptors": [
    { "name": "MyMachine",
      "type": "http://schemas.dmtf.org/cimi/1/Machine",
      "machineTemplate":
        { "name" : "Machine in system demo",
          "description" : "Machine in system",
          "machineConfig": { "href": "http://example.com/configs/tiny" },
          "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
          "credential": { "href": "#MyCredential" },
          "volumes": [
            { "initialLocation": "/vol",
              "href": "#MyVolume"
            }
          ]
        }
    },
    { "name": "MyCredential",
      "type": "http://schemas.dmtf.org/cimi/1/Credential",
      "credentialTemplate":
        { "href": "http://example.com/credentialTemplates/72000" }
    },
    { "name": "MyVolume",
      "type": "http://schemas.dmtf.org/cimi/1/Volume",
      "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839" }
    }
  ]
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/systemTemplates/48920
```

### Step 3 : Create a new System by using a System Template

Now create a new System by using this System Template:

```
POST /systems HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCreate",
  "name": "MySystem1",
  "description": "My first system",
  "systemTemplate": { "href": "http://example.com/systemTemplates/48920" }
}
```

```
HTTP/1.1 201 Created
Location: http://example.com/systems/78342
```

Note that, alternatively, the Provider could have decided to return a reference to a Job resource instead of waiting until the System is completely created. Instead of the above 201 response, this could have resulted in the following response:

```
HTTP/1.1 202 Accepted
```

```
CIMI-Job-URI: http://example.com/Jobs/90001

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Job",
  "id": "http://example.com/Jobs/90001",
  "name": "SystemCreationJob",
  "created": "2012-03-15T12:15:00Z",
  "updated": "2012-03-15T12:15:00Z",
  "targetResource": { "href": "http://example.com/systems" },
  "affectedResources": {
    "href": " http://example.com/systems/110001",
  },
  "action": "add",
  "status": "RUNNING",
  "progress": 30,
  "timeOfStatusChange": "2012-03-15T12:15:00Z",
  "isCancellable": "true",
  "nestedJobs": [
    { "href": "http://example.com/Jobs/90002" },
    { "href": "http://example.com/Jobs/90003" }
  ]
}
```

According to this response, the Provider chose to create two nested Jobs and the "affectedResources" attribute includes a reference to the newly created System. Periodic retrieval of the Job's representation will allow the Consumer to determine when the Job is completed, i.e., it is completed when the "progress" attribute has a value of 100.

#### Step 4 : Query the new System

Retrieve the System to get the full representation of the new System:

```
GET /systems/87342
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
  "id": "http://example.com/systems/78342",
  "name": "MySystem1",
  "description": "My first system",
  "created": "2012-08-15T12:15:00Z",
  "updated": "2012-08-15T12:15:00Z",
  "state": "STOPPED",
  "machines": { "href": "http://example.com/systems/87432/machines" },
  "credentials": { "href": "http://example.com/systems/87342/creds" },
  "volumes": { "href": "http://example.com/systems/87342/vols" },
  "operations" : [
    {"rel": "edit", "href": "http://example.com/systems/78342" }
  ]
}
```

## Scenario 6 : Editing System Templates

In this scenario you will add a second Machine to an existing System Template.

### Step 1 : Edit an existing System Template

You will edit the System Template created in a previous scenario and add another machine that shares its credential and volume resources:

SystemTemplate:

```
http://example.com/systemTemplates/48920
```

Retrieve the existing SystemTemplate definition:

```
GET /systemTemplates/48920 HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
  "name": "System Demo1",
  "description": "My first system template demo",
  "created": "2012-08-15T12:15:00Z",
  "updated": "2012-08-15T12:15:00Z",
  "componentDescriptors": [
    { "name": "MyMachine",
      "type": "http://schemas.dmtf.org/cimi/1/Machine",
      "machineTemplate":
        { "name" : "Machine in system demo",
          "description" : "Machine in system",
          "machineConfig": { "href": "http://example.com/configs/tiny" },
          "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
          "credential": { "href": "#MyCredential" },
          "volumes": [
            { "initialLocation": "/vol",
              "href": "#MyVolume"
            }
          ]
        }
    },
    { "name": "MyCredential",
      "type": "http://schemas.dmtf.org/cimi/1/Credential",
      "credentialTemplate":
        { "href": "http://example.com/credentialTemplates/72000" }
    },
    { "name": "MyVolume",
      "type": "http://schemas.dmtf.org/cimi/1/Volume",
      "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839" }
    }
  ],
  "operations" : [
    { "rel": "edit", "href": "http://example.com/systemTemplates/48920" }
  ]
}
```

Now update the SystemTemplate resource:

```
PUT /systemTemplates/48920 HTTP/1.1
```

```
Content-Type: application/json
```

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
  "name": "System Demo1",
  "description": "My first system template demo",
  "componentDescriptors": [
    { "name": "MyMachine",
      "type": "http://schemas.dmtf.org/cimi/1/Machine",
      "machineTemplate":
        { "name" : "Machine in system demo",
          "description" : "Machine in system",
          "machineConfig": { "href": "http://example.com/configs/tiny" },
          "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
          "credential": { "href": "#MyCredential" },
          "volumes": [
            { "initialLocation": "/vol",
              "href": "#MyVolume",
            }
          ]
        }
    },
    { "name": "MyCredential",
      "type": "http://schemas.dmtf.org/cimi/1/Credential",
      "credentialTemplate":
        { "href": "http://example.com/credentialTemplates/72000" }
    },
    { "name": "MyVolume",
      "type": "http://schemas.dmtf.org/cimi/1/Volume",
      "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839" }
    }
  ]
}
```

HTTP/1.1 200 OK

Content-Type: application/json

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
  "name": "System Demo1",
  "description": "My first system template demo",
  "created": "2012-09-15T12:15:00Z",
  "updated": "2012-09-15T12:15:00Z",
  "componentDescriptors": [
    { "name": "MyMachine",
      "type": "http://schemas.dmtf.org/cimi/1/Machine",
      "machineTemplate":
        { "name" : "Machine in system demo",
          "description" : "Machine in system",
          "machineConfig": { "href": "http://example.com/configs/tiny" },
          "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
          "credential": { "href": "#MyCredential" },
          "volumes": [
            { "initialLocation": "/vol",
              "href": "#MyVolume"
            }
          ]
        }
    },
    { "name": "MyCredential",
      "type": "http://schemas.dmtf.org/cimi/1/Credential",
      "credentialTemplate":
        { "href": "http://example.com/credentialTemplates/72000" }
    },
    { "name": "MyVolume",
      "type": "http://schemas.dmtf.org/cimi/1/Volume",
      "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839" }
    }
  ]
}
```

```

    },
    { "name": "MyCredential",
      "type": "http://schemas.dmtf.org/cimi/1/Credential",
      "credentialTemplate":
        { "href": "http://example.com/credentialTemplates/72000" }
    },
    { "name": "MyVolume",
      "type": "http://schemas.dmtf.org/cimi/1/Volume",
      "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839" }
    }
  ],
  "operations": [
    { "rel": "edit", "href": "http://example.com/systemTemplates/48920" }
  ]
}

```

### Step 2 : Create a new System using a System Template

Now create a new System using this System Template by sending a POST to the same URL that you used in the previous scenario to create a new System:

```

POST /systems HTTP/1.1
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCreate",
  "name": "MySystem2",
  "description": "My second system",
  "systemTemplate": { "href": "http://example.com/systemTemplates/48920" }
}

```

```

HTTP/1.1 201 Created
Location: http://example.com/systems/78343

```

### Step 3 : Query the new System

Retrieve the System to get the full representation of the new System:

```

GET /systems/87343

HTTP/1.1 200 OK
Content-Type: application/json

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
  "id": "http://example.com/systems/78342",
  "name": "MySystem2",
  "description": "My second system",
  "created": "2012-10-15T12:15:00Z",
  "updated": "2012-10-15T12:15:00Z",
  "state": "STOPPED",
  "machines": { "href": "http://example.com/systems/87432/machines" },
  "credentials": { "href": "http://example.com/systems/87342/creds" },
  "volumes": { "href": "http://example.com/systems/87342/vols" },
  "operations" : [
    { "rel": "edit", "href": "http://example.com/systems/78342" }
  ]
}

```

**Change log:**

<b>Version</b>	<b>Date</b>	<b>Description</b>
1.0.0	2012-08-28	Version 1.0
1.0.1	2012-09-12	Errata