



1  
2  
3  
4  
5

**Document Identifier: DSP2027**

**Date: 2014-07-31**

**Version: 1.1.0**

6 **Cloud Infrastructure Management Interface**  
7 **(CIMI) Primer**

8 **Document Type: White Paper**

9 **Document Status: DMTF Informational**

10 **Document Language: en-US**

11 Copyright notice

12 Copyright © 2012-2014 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

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

33			
34			
35	Foreword .....		5
36	Introduction.....		7
37	1 Creating a new Machine.....		9
38	1.1 Retrieve the CEP .....		9
39	1.2 Retrieve the list of Machine Images.....		9
40	1.3 Choose a Machine Image .....		10
41	1.4 Retrieve the list of Machine Configurations .....		10
42	1.5 Choose a Machine Configuration .....		11
43	1.6 Create a new Credential Resource.....		11
44	1.7 Create a new Machine .....		13
45	1.8 Query new Machine .....		13
46	1.9 Start a Machine .....		14
47	1.10 Query a Machine.....		14
48	1.11 Stop a Machine .....		14
49	1.12 Update a Machine's attributes .....		15
50	2 Adding a New Volume to a Machine .....		15
51	2.1 Obtain the Machine URL .....		15
52	2.2 Retrieve the CEP .....		15
53	2.3 Get the list of VolumeConfigurations .....		16
54	2.4 Choose a Volume Configuration.....		16
55	2.5 Create a new Volume .....		16
56	2.6 Retrieve the Volume information .....		17
57	2.7 Retrieve the Machine's Volume collection .....		17
58	2.8 Connect the new Volume to a Machine .....		19
59	2.9 Query the Machine's volume collection .....		19
60	3 Defining and using Machine Templates .....		20
61	3.1 Retrieve the CEP .....		20
62	3.2 Create a new Machine Template.....		20
63	3.3 Create a new Machine by using a Machine Template.....		21
64	4 Creating a new Machine from an existing Volume.....		21
65	4.1 Retrieve the CEP .....		21
66	4.2 Get the list of Volumes.....		22
67	4.3 Choose a Volume .....		22
68	4.4 Create a new Machine .....		22
69	4.5 Query new Machine .....		23
70	5 Defining and using System Templates.....		24
71	5.1 Retrieve the CEP .....		24
72	5.2 Create a new System Template .....		24
73	5.3 Create a new System by using a System Template.....		25
74	5.4 Query the new System.....		26
75	6 Editing System Templates.....		27
76	6.1 Edit an existing System Template .....		27
77	6.2 Create a new System using a System Template.....		29
78	6.3 Query the new System.....		30
79	7 Creating a Public Facing Network.....		30
80	7.1 Retrieve the Cloud Entry Point (CEP).....		30
81	7.2 Verify provider examples for network configurations.....		31
82	7.3 Retrieve a list of network port configurations.....		32
83	7.4 Creating a PUBLIC accessible network.....		32
84	7.5 Verify that a PUBLIC network has been created .....		33

85        7.6    Create a network port ..... 34  
86        7.7    Create a Machine attached to the public network ..... 34  
87    8    Provider responses and return values ..... 36  
88        8.1    Unrecognized attributes ..... 36  
89        8.2    Unreasonable requests..... 36  
90    ANNEX A (informative) Change log ..... 37  
91

92

**Foreword**

93 The *Cloud Infrastructure Management Interface (CIMI) Primer* (DSP2027) was prepared by the Cloud  
 94 Management Working Group of the DMTF. This document contains scenarios that describe common uses  
 95 of the CIMI protocol.

96 This specification has been developed as a result of joint work with many individuals and teams,  
 97 including:

98 Dies Köper Fujitsu (Editor)  
 99 Larry Lamers VMware (Editor)

**Contributors:**

101	Hemal Shah	Broadcom
102	John Crandall	Brocade Communications Systems
103	Paul Lipton	CA Technologies
104	Efraim Moscovich	CA Technologies
105	Mark Carlson	DMTF Fellow
106	Marvin Waschke	DMTF Fellow
107	Steven Neely	Cisco
108	Shishir Pardikar	Citrix Systems Inc.
109	George Ericson	EMC
110	Norbert Floeren	Ericsson AB
111	Arturo Martin de Nicolas	Ericsson AB
112	Ruby Krishnaswamy	France Telecom Group
113	Jacques Durand	Fujitsu
114	Kazunori Iwasa	Fujitsu
115	Jesus Molina	Fujitsu
116	Tom Rutt	Fujitsu
117	Maarten Wiggers	Fujitsu
118	Bryan Murray	Hewlett
119	Derek Coleman	Hewlett Packard Company
120	Robert Freund	Hitachi, Ltd.
121	Fred Maciel	Hitachi, Ltd.
122	Eric Wells	Hitachi, Ltd.
123	Zhexuan Song	Huawei
124	Jeff Wheeler	Huawei
125	Aaron Zhang	Huawei
126	HengLiang Zhang	Huawei
127	Doug Davis	IBM
128	Mark Johnson	IBM
129	Andreas Maier	IBM
130	Matthew Rutkowski	IBM
131	Keith Bankston	Microsoft Corporation
132	Nathan Burkhart	Microsoft Corporation
133	Josh Cohen	Microsoft Corporation
134	Yigal Edery	Microsoft Corporation
135	Colleen Evans	Microsoft Corporation
136	Krishnan Gopalan	Microsoft Corporation
137	John Parchem	Microsoft Corporation
138	Nihar Shah	Microsoft Corporation
139	Hiroshi Dempo	NEC Corporation
140	James Livingston	NEC Corporation
141	Ryuichi Ogawa	NEC Corporation
142	Steve Carter	Novell
143	Ashok Malhotra	Oracle

144	Gilbert Pilz	Oracle
145	Jack Yu	Oracle
146	Marios Andreou	Red Hat
147	David Lutterkort	Red Hat
148	Steve Winkler	SAP AG
149	Enrico Ronco	Telecom Italia
150	Federico Rossini	Telecom Italia
151	Fernando de la Iglesia	Telefónica
152	Fermín Galán	Telefónica
153	Miguel Peñalvo	Telefónica
154	Alvaro Polo	Telefónica
155	Alan Sill	Texas Tech University
156	Vince Lubsey	Virtustream Inc.
157	Winston Bumpus	VMware Inc.
158	Jim Davis	WBEM Solutions
159	Ghazanfar Ali	ZTE Corporation
160	Junsheng Chu	ZTE Corporation
161	Bhumip Khasnabish	ZTE Corporation

162

## Introduction

163 For the sake of simplicity, in each of the following scenarios, the Cloud provider only supports the  
164 minimum features needed to demonstrate the features highlighted by each scenario. Therefore, the  
165 results of the query to the Cloud Entry Point (CEP) to retrieve the list of supported features are  
166 customized for each scenario. Additionally, the URI of the Cloud Entry Point is assumed to be  
167 `http://example.com/CEP` and all resources are assumed to be available in the `example.com` domain and  
168 accessible with the same protocol (HTTP). In the HTTP request-response examples in this document, the  
169 creation of the connection and HTTP headers not mandated by the CIMI specification are omitted for  
170 brevity.  
171





# 173 Cloud Infrastructure Management Interface (CIMI) Primer

## 174 1 Creating a new Machine

175 This scenario creates a new Machine. The new Machine's configuration is based on existing  
176 configurations and images offered by the provider. However, a new Credential resource (userid and  
177 password) is created.

### 178 1.1 Retrieve the CEP

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

```
181 GET /CEP HTTP/1.1
```

```
182 HTTP/1.1 200 OK  
183 Content-Type: application/json  
184  
185 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",  
186   "id": "http://example.com/CEP",  
187   "baseURI": "http://example.com/",  
188   "resourceMetadata": { "href": "http://example.com/resourceMetadata" },  
189   "machines": { "href": "http://example.com/machines" },  
190   "machineConfigs": { "href": "http://example.com/machineConfigs" },  
191   "machineImages": { "href": "http://example.com/machineImages" },  
192   "credentials": { "href": "http://example.com/credentials" }  
193 }  
194 }
```

### 195 1.2 Retrieve the list of Machine Images

196 Before you can create a new Machine, first decide what kind of operating system or software you want to  
197 have preinstalled. The Machine Images collection is the set of Machine Images that this Cloud offers -  
198 note that some Machine Images may be predefined by the Cloud while some may be user created. The  
199 URI path comes from the data returned in the query to CEP for the `machineImages` key.

200 To retrieve the list of Machine Images, use the following syntax:

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

```
202 HTTP/1.1 200 OK  
203 Content-Type: application/json  
204  
205 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImageCollection",  
206   "id": "http://example.com/machineImages",  
207   "count": 3,  
208   "machineImages": [  
209     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",  
210       "id": "http://example.com/images/WinXP-SP2",  
211       "name": "WinXP SP2",  
212       "description": "Windows XP with Service Pack 2",  
213       "created": "2012-01-01T12:00:00Z",  
214       "updated": "2012-01-01T12:00:00Z",  
215       "imageLocation": "http://example.com/data/8934322"  
216     },  
217     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",  
218       "id": "http://example.com/images/Win7",  
219       "name": "Windows 7",  
220     },  
221     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",  
222       "id": "http://example.com/images/Win7",  
223       "name": "Windows 7",  
224     }  
225   ]  
226 }
```

```

222     "description": "Windows 7",
223     "created": "2012-01-01T12:00:00Z",
224     "updated": "2012-01-01T12:00:00Z",
225     "imageLocation": "http://example.com/data/8934344"
226   },
227
228   { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
229     "id": "http://example.com/images/Linux-SUSE",
230     "name": "Linux SUSE",
231     "description": "Linux SUSE",
232     "created": "2012-01-01T12:00:00Z",
233     "updated": "2012-01-01T12:00:00Z",
234     "imageLocation": "http://example.com/data/8934311"
235   }
236 ]
237 }

```

### 238 1.3 Choose a Machine Image

239 Next examine each Machine Image to find one that meets your needs. The first one is acceptable, so it  
 240 will be used later.

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

```
243 GET /machineImages?$first=1&$last=1 HTTP/1.1
```

```

244
245 HTTP/1.1 200 OK
246 Content-Type: application/json
247
248 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImageCollection",
249   "id": "http://example.com/machineImages",
250   "count": 3,
251   "machineImages": [
252     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
253       "id": "http://example.com/images/WinXP-SP2",
254       "name": "WinXP SP2",
255       "description": "Windows XP with Service Pack 2",
256       "created": "2012-01-01T12:00:00Z",
257       "updated": "2012-01-01T12:00:00Z",
258       "imageLocation": "http://example.com/data/8934322"
259     }
260   ]
261 }

```

262 Note that you do not need to specify `$first=1` in this case because "1" is its default value. The first  
 263 machineImage is returned.

### 264 1.4 Retrieve the list of Machine Configurations

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

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

```

268
269 HTTP/1.1 200 OK
270 Content-Type: application/json
271
272 { "resourceURI":
273   "http://schemas.dmtf.org/cimi/1/MachineConfigurationCollection",
274   "id": "http://example.com/machineConfigs",
275   "count": 3,
276   "machineConfigurations": [

```

```

277 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
278   "id": "http://example.com/machineConfigs/tiny",
279   "name": "tiny",
280   "description": "a teenie tiny one",
281   "created": "2012-01-01T12:00:00Z",
282   "updated": "2012-01-01T12:00:00Z",
283   "cpu": 1,
284   "memory": 4000000,
285   "disks" : [
286     { "capacity": 50000000 }
287   ]
288 },
289
290 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
291   "id": "http://example.com/machineConfigs/small",
292   "name": "small",
293   "description": "a small sized one",
294   "created": "2012-01-01T12:00:00Z",
295   "updated": "2012-01-01T12:00:00Z",
296   "cpu": 1,
297   "memory": 8000000,
298   "disks" : [
299     { "capacity": 500000000 }
300   ]
301 },
302
303 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
304   "id": "http://example.com/machineConfigs/medium",
305   "name": "medium",
306   "description": "a medium one",
307   "created": "2012-01-01T12:00:00Z",
308   "updated": "2012-01-01T12:00:00Z",
309   "cpu": 1,
310   "memory": 16000000,
311   "disks" : [
312     { "capacity": 1000000000 },
313     { "capacity": 1000000000 }
314   ]
315 }
316 ]
317 }

```

## 318 1.5 Choose a Machine Configuration

319 Next examine the returned list and select a Machine Configuration that suits your needs. The first one is  
 320 acceptable, so it will be used later. It is identified by the `id`  
 321 `"http://example.com/machineConfigs/tiny"`.

## 322 1.6 Create a new Credential Resource

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

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

328 The response is:

```

329 HTTP/1.1 200 OK
330 Content-Type: application/json
331
332 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialCollection",

```

```

333     "id": "http://example.com/credentials",
334     "operations": [ { "rel": "add", "href": "http://example.com/credentials" } ]
335 }
336

```

337 Notice at this point that there are no Credential resources in the environment. Before you can create a  
 338 new Credential resource, you must first discover this Cloud provider's extension attributes for the  
 339 Credential resource. By default the CIMI specification does not define how the initial user of a new  
 340 Machine is specified; rather it is left open for each Cloud provider to determine how this information  
 341 should be provided. Clients can discover this information by querying the Credential resource metadata  
 342 resource. To examine this resource, first look through the `ResourceMetadata` collection for this  
 343 provider's description of the Credential's resource. Start by retrieving the `ResourceMetadata` collection  
 344 from the URI referenced in the CEP:

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

```

346 HTTP/1.1 200 OK
347 Content-Type: application/json
348
349 { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadataCollection",
350   "id": "http://example.com/resourceMetadata",
351   "count": 1,
352   "resourceMetadatas": [
353     { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadata",
354       "id": "http://example.com/resources/Credential",
355       "typeURI": "http://schemas.dmtf.org/cimi/1/Credential",
356       "name": "Credential",
357       "attributes": [
358         { "name": "userName", "namespace": "http://example.com",
359           "type": "string", "required": "true" },
360         { "name": "password", "namespace": "http://example.com",
361           "type": "string", "required": "true" }
362       ]
363     }
364   ]
365 }
366

```

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

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

```

372 POST /credentials HTTP/1.1
373 Content-Type: application/json
374
375 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialCreate",
376   "name": "Default",
377   "description": "My Default User",
378   "credentialTemplate": {
379     "userName": "JoeSmith",
380     "password": "letmein"
381   }
382 }
383

```

```

384 HTTP/1.1 201 Created
385 Location: http://example.com/creds/12345

```

386 Note While the "userID" and "password" attributes were discovered via the Credential ResourceMetadata, the  
 387 "name" and "description" attributes are part of the common set of attributes available on all resources. In a future  
 388 scenario it is shown how the client knew that "userID" and "password" were the proper attribute names for this image  
 389 type and Cloud provider.

## 390 1.7 Create a new Machine

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

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

```
393
394 HTTP/1.1 200 OK
395 Content-Type: application/json
396
397 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCollection",
398   "id": "http://example.com/machines",
399   "count": 0,
400   "operations": [ { "rel": "add", "href": "http://example.com/machines" } ]
401 }
```

402 If you only want to know the available operations, issue the following command.

```
403 GET /machines?$select=operations HTTP/1.1
```

```
404
405 HTTP/1.1 200 OK
406 Content-Type: application/json
407
408 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCollection",
409   "operations": [ { "rel": "add", "href": "http://example.com/machines" } ]
410 }
```

411 Now create a new one:

```
412 POST /machines HTTP/1.1
413 Content-Type: application/json
414
415 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate",
416   "name": "myMachine1",
417   "description": "My very first machine",
418   "machineTemplate": {
419     "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
420     "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
421     "credential": { "href": "http://example.com/creds/12345" }
422   }
423 }
```

```
424 HTTP/1.1 201 Created
```

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

427 The response returns a unique machine reference "<http://example.com/machines/843752>" that is used in  
428 the following subclause.

## 429 1.8 Query new Machine

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

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

```
432
433 HTTP/1.1 200 OK
434 Content-Type: application/json
435
436 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
437   "id": "http://example.com/machines/843752",
438   "name": "myMachine1",
439   "description": "My very first machine",
440   "created": "2012-08-15T12:15:00Z",
441   "updated": "2012-08-15T12:15:00Z",
442   "state": "STOPPED",
443   "cpu": 1,
```

```

444     "memory": 4000000,
445     "disks" : { "href": "http://example.com/machines/843752/disks" },
446     "networkInterfaces": { "href": "http://example.com/machines/843752/NIs" },
447     "operations": [
448       { "rel": "edit", "href": "http://example.com/machines/843752" },
449       { "rel": "delete", "href": "http://example.com/machines/843752" },
450       { "rel": "http://schemas.dmtf.org/cimi/1/action/start",
451         "href": "http://example.com/machines/843752" }
452     ]
453 }

```

454 Notice the "state" attribute on the Machine is "STOPPED" because that is the initial state of a new  
455 machine.

## 456 1.9 Start a Machine

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

```

459 POST /machines/843752 HTTP/1.1
460 Content-Type: application/json
461
462 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
463   "action": "http://schemas.dmtf.org/cimi/1/action/start"
464 }

```

```

465
466 HTTP/1.1 204 No Content

```

## 467 1.10 Query a Machine

468 Query the Machine again to verify that it is started:

```

469 GET /machines/843752 HTTP/1.1
470
471 HTTP/1.1 200 OK
472 Content-Type: application/json
473
474 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
475   "id": "http://example.com/machines/843752",
476   "name": "myMachine1",
477   "description": "My very first machine",
478   "created": "2012-08-15T12:15:00Z",
479   "updated": "2012-08-15T12:15:00Z",
480   "state": "STARTED",
481   "cpu": 1,
482   "memory": 4000000,
483   "disks" : { "href": "http://example.com/machines/843752/disks",
484   "networkInterfaces": { "href": "http://example.com/machines/843752/NIs",
485   "operations": [
486     { "rel": "edit", "href": "http://example.com/machines/843752" },
487     { "rel": "delete", "href": "http://example.com/machines/843752" },
488     { "rel": "http://schemas.dmtf.org/cimi/1/action/stop",
489       "href": "http://example.com/machines/843752" }
490   ]
491 }

```

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

## 494 1.11 Stop a Machine

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

```
496 POST /machines/843752 HTTP/1.1
497 Content-Type: application/json
498
499 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
500   "action": "http://schemas.dmtf.org/cimi/1/action/stop"
501 }
502
503 HTTP/1.1 204 No Content
```

## 504 1.12 Update a Machine's attributes

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

```
507 PUT /machines/843752?$select=name,description HTTP/1.1
508 Content-Type: application/json
509
510 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
511   "name" : "Cool Demo #1"
512 }
513
514 HTTP/1.1 200 OK
515
516 { "name" : "Cool Demo #1" }
```

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

## 520 2 Adding a New Volume to a Machine

521 This scenario creates a new Volume and connects it to an existing Machine.

### 522 2.1 Obtain the Machine URL

523 Machine:

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

### 525 2.2 Retrieve the CEP

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

```
528 GET /CEP HTTP/1.1
529
530 HTTP/1.1 200 OK
531 Content-Type: application/json
532
533 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
534   "id": "http://example.com/CEP",
535   "baseURI": "http://example.com/",
536   "machines": { "href": "http://example.com/machines" },
537   "machineConfigs": { "href": "http://example.com/machineConfigs" },
538   "machineImages": { "href": "http://example.com/machineImages" },
539   "credentials": { "href": "http://example.com/credentials" },
540   "volumes": { "href": "http://example.com/volumes" },
541   "volumeConfigs": { "href": "http://example.com/volumeConfigs" }
542 }
```

## 543 2.3 Get the list of VolumeConfigurations

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

```
547 GET /volumeConfigs HTTP/1.1
```

```
548 HTTP/1.1 200 OK
549 Content-Type: application/json
550
551 { "resourceURI":
552   "http://schemas.dmtf.org/cimi/1/VolumeConfigurationCollection",
553   "id": "http://example.com/volumeConfigs",
554   "volumeConfigurations": [
555     { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
556       "id": "http://example.com/volumeConfigs/small",
557       "name": "Small",
558       "description": "A pretty small one",
559       "created": "2012-08-15T12:15:00Z",
560       "updated": "2012-08-15T12:15:00Z",
561       "type": "http://schemas.dmtf.org/cimi/1/mapped",
562       "format": "NTFS",
563       "capacity": 60000000
564     },
565     { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
566       "id": "http://example.com/volumeConfigs/medium",
567       "name": "Medium",
568       "description": "A medium sized one",
569       "created": "2012-08-15T12:15:00Z",
570       "updated": "2012-08-15T12:15:00Z",
571       "type": "http://schemas.dmtf.org/cimi/1/mapped",
572       "format": "NTFS",
573       "capacity": 500000000
574     },
575     { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
576       "id": "http://example.com/volumeConfigs/large",
577       "name": "Large",
578       "description": "A large one",
579       "created": "2012-08-15T12:15:00Z",
580       "updated": "2012-08-15T12:15:00Z",
581       "type": "http://schemas.dmtf.org/cimi/1/mapped",
582       "format": "NTFS",
583       "capacity": 1000000000
584     }
585   ]
586 }
587 ]
588 ]
589 }
```

## 590 2.4 Choose a Volume Configuration

591 Next examine each Volume Configuration to find the one that meets your needs. The first one is  
592 acceptable, so it will be used later.

## 593 2.5 Create a new Volume

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

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

```
596
597 HTTP/1.1 200 OK
```



```
598 Content-Type: application/json
599
600 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeCollection",
601   "id": "http://example.com/volumes",
602   "operations": [ { "rel": "add", "href": "http://example.com/volumes" } ]
603 }
```

604 **Now create a new Volume:**

```
605 POST /volumes HTTP/1.1
606 Content-Type: application/json
607
608 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeCreate",
609   "name": "myVolume1",
610   "description": "My first new volume",
611   "volumeTemplate": {
612     "volumeConfig": { "href": "http://example.com/volumeConfigs/small" }
613   }
614 }
615
616 HTTP/1.1 201 Created
617 Location: http://example.com/volumes/35782
```

## 618 2.6 Retrieve the Volume information

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

```
621 GET /volumes/35782 HTTP/1.1
622
623 HTTP/1.1 200 OK
624 Content-Type: application/json
625
626 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Volume",
627   "id": "http://example.com/volumes/35782",
628   "name": "myVolume1",
629   "created": "2012-09-15T12:15:00Z",
630   "updated": "2012-09-15T12:15:00Z",
631   "description": "My first new volume",
632   "type": "http://schemas.dmtf.org/cimi/1/mapped",
633   "capacity": 60000000,
634   "operations": [
635     { "rel": "edit", "href": "http://example.com/volumes/35782" },
636     { "rel": "delete", "href": "http://example.com/volumes/35782" }
637   ]
638 }
```

## 639 2.7 Retrieve the Machine's Volume collection

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

```
643 GET /machines/843752 HTTP/1.1
644
645 HTTP/1.1 200 OK
646 Content-Type: application/json
647
648 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
649   "id": "http://example.com/machines/843752",
650   "name": "myMachine1",
651   "description": "My very first machine",
652   "created": "2012-08-15T12:15:00Z",
653   "updated": "2012-08-15T12:15:00Z",
```

```

654     "state": "STARTED",
655     "cpu": 1,
656     "memory": 4000000,
657     "disks": { "href": "http://example.com/machines/843752/disks",
658     "volumes": { "href": "http://example.com/machines/843752/volumes" },
659     "networkInterfaces": { "href": "http://example.com/machines/843752/NIs" },
660     "operations": [
661       { "rel": "edit", "href": "http://example.com/machines/843752" },
662       { "rel": "delete", "href": "http://example.com/machines/843752" },
663       { "rel": "http://schemas.dmtf.org/cimi/1/action/stop",
664         "href": "http://example.com/machines/843752" }
665     ]
666 }

```

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

670 Now retrieve the Volume collection:

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

```

672 HTTP/1.1 200 OK
673 Content-Type: application/json
674 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumeCollection",
675   "id": "http://example.com/machines/843752/volumes",
676   "operations": [
677     { "rel": "add", "href": "http://example.com/machines/843752/volumes" }
678   ]
679 }
680 }
681 }

```

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

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

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

```

686 HTTP/1.1 200 OK
687 Content-Type: application/json
688 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
689   "id": "http://example.com/machines/843752",
690   "name": "myMachine1",
691   "description": "My very first machine",
692   "created": "2012-08-15T12:15:00Z",
693   "updated": "2012-08-15T12:15:00Z",
694   "state": "STARTED",
695   "cpu": 1,
696   "memory": 4000000,
697   "disks": { "href": "http://example.com/machines/843752/disks",
698     "volumes": {
699       "href": "http://example.com/machines/843752/volumes",
700       "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumeCollection",
701       "id": "http://example.com/machines/843752/volumes",
702       "operations": [
703         { "rel": "add", "href": "http://example.com/machines/843752/volumes" }
704       ]
705     },
706   "networkInterfaces": { "href": "http://example.com/machines/843752/NIs" },
707   "operations": [
708     { "rel": "edit", "href": "http://example.com/machines/843752" },
709     { "rel": "delete", "href": "http://example.com/machines/843752" },
710   ]
711 }

```

```

712     { "rel": "http://schemas.dmtf.org/cimi/1/action/stop",
713       "href": "http://example.com/machines/843752" }
714   ]
715 }

```

## 716 2.8 Connect the new Volume to a Machine

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

```

719 POST /machines/843752/volumes HTTP/1.1
720 Content-Type: application/json
721
722 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolume",
723   "initialLocation": "V",
724   "volume": { "href": "http://example.com/volumes/35782" } }
725 }

```

```

726
727 HTTP/1.1 201 Created
728 Content-Type: application/json
729
730 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolume",
731   "id": "http://example.com/machines/843752/volumes/1",
732   "initialLocation": "V",
733   "volume": { "href": "http://example.com/volumes/35782" },
734   "operations": [
735     { "rel": "edit", "href": "http://example.com/machines/843752/volumes/1"},
736     { "rel": "delete", "href": "http://example.com/machines/843752/volumes/1"}
737   ]
738 }

```

## 739 2.9 Query the Machine's volume collection

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

```

742 GET /machines/843752/volumes HTTP/1.1
743
744 HTTP/1.1 200 OK
745 Content-Type: application/json
746
747 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumeCollection",
748   "id": "http://example.com/machines/843752/volumes",
749   "machineVolumes": [
750     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumes",
751       "id": "http://example.com/machines/843752/volumes/1",
752       "initialLocation": "V",
753       "volume": { "href": "http://example.com/volumes/35782" },
754       "operations": [
755         { "rel": "edit","href": "http://example.com/machines/843752/volumes/1"},
756         { "rel": "delete","href": "http://example.com/machines/843752/volumes/1"}
757       ]
758     },
759     "operations": [
760       { "rel": "add", "href": "http://example.com/machines/843752/volumes" }
761     ]
762 }

```

## 763 3 Defining and using Machine Templates

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

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

### 772 3.1 Retrieve the CEP

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

```
775 GET /CEP HTTP/1.1
```

```
776  
777 HTTP/1.1 200 OK  
778 Content-Type: application/json  
779  
780 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",  
781   "id": "http://example.com/CEP",  
782   "baseURI": "http://example.com/",  
783   "machines": { "href": "http://example.com/machines" },  
784   "machineTemplates": { "href": "http://example.com/machineTemplates" },  
785   "machineConfigs": { "href": "http://example.com/machineConfigs" },  
786   "machineImages": { "href": "http://example.com/machineImages" },  
787   "credentials": { "href": "http://example.com/credentials" }  
788 }
```

### 789 3.2 Create a new Machine Template

790 From the previous scenarios, you already have the MachineConfiguration, MachineImage, and Credential  
791 resources that are reused for this MachineTemplate:

792 MachineConfiguration:

```
793 http://example.com/machineConfigs/tiny
```

794 MachineImage:

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

796 Credential:

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

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

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

```
802
803 HTTP/1.1 200 OK
804 Content-Type: application/json
805
806 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplateCollection",
807   "id": "http://example.com/machineTemplates",
808   "operations": [
809     { "rel": "add", "href": "http://example.com/machineTemplates" }
810   ]
811 }
```

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

813 Now create the new MachineTemplate resource:

```
814 POST /machineTemplates HTTP/1.1
815 Content-Type: application/json
816
817 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplate",
818   "name": "Demo1",
819   "description": "My first demo",
820   "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
821   "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
822   "credential": { "href": "http://example.com/creds/12345" }
823 }
```

```
824
825 HTTP/1.1 201 Created
826 Location: http://example.com/machineTemplates/82754
```

### 827 3.3 Create a new Machine by using a Machine Template

828 Now create a new Machine by using this Machine Template:

```
829 POST /machines HTTP/1.1
830 Content-Type: application/json
831
832 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate",
833   "name": "myMachine2",
834   "description": "My second machine",
835   "machineTemplate": { "href": "http://example.com/machineTemplates/82754" }
836 }
```

```
837
838 HTTP/1.1 201 Created
839 Location: http://example.com/machines/843799
```

## 840 4 Creating a new Machine from an existing Volume

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

### 843 4.1 Retrieve the CEP

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

```
846 GET /CEP HTTP/1.1
847
848 HTTP/1.1 200 OK
849 Content-Type: application/json
850
851 { "resourceType": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
852   "id": "http://example.com/CEP",
```

```

853   "baseURI": "http://example.com/",
854   "resourceMetadata": { "href": "http://example.com/resourceMetadata" },
855   "machines": { "href": "http://example.com/machines" },
856   "machineConfigs": { "href": "http://example.com/machineConfigs" },
857   "machineImages": { "href": "http://example.com/machineImages" },
858   "credentials": { "href": "http://example.com/credentials" },
859   "volumes": { "href": "http://example.com/volumes" },
860   "volumeConfigs": { "href": "http://example.com/volumeConfigs" }
861 }

```

## 862 4.2 Get the list of Volumes

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

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

```

866 HTTP/1.1 200 OK
867 Content-Type: application/json
868
869 { "resourceType": "http://schemas.dmtf.org/cimi/1/VolumeCollection",
870   "id": "http://example.com/volumes",
871   "volumes": [
872     { "resourceType": "http://schemas.dmtf.org/cimi/1/Volume",
873       "id": "http://example.com/volumes/vol1",
874       "name": "Win7-Bootable",
875       "created": "2012-08-15T12:15:00Z",
876       "updated": "2012-08-15T12:15:00Z",
877       "description": "A bootable volume running Windows 7",
878       "state": "AVAILABLE",
879       "capacity": 60000000,
880       "bootable": true
881     },
882     { "resourceType": "http://schemas.dmtf.org/cimi/1/Volume",
883       "id": "http://example.com/volumes/vol2",
884       "name": "Generic Volume",
885       "created": "2012-08-15T12:15:00Z",
886       "updated": "2012-08-15T12:15:00Z",
887       "description": "A generic volume for Windows",
888       "state": "AVAILABLE",
889       "capacity": 60000000,
890       "bootable": true
891     }
892   ]
893 }
894
895

```

## 896 4.3 Choose a Volume

897 Next examine each Volume to find the one that meets your needs. The first one is acceptable, so it will be  
 898 used later.

## 899 4.4 Create a new Machine

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

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

```

902
903 HTTP/1.1 200 OK
904 Content-Type: application/json
905

```

```
906 { "resourceType": "http://schemas.dmtf.org/cimi/1/MachineCollection",
907     "id": "http://example.com/machines",
908     "operations": [
909         { "rel": "add", "href": "http://example.com/machines" }
910     ]
911 }
```

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

```
914 POST /machines HTTP/1.1
915 Content-Type: application/json
916
917 { "resourceType": "http://schemas.dmtf.org/cimi/1/MachineCreate",
918     "name": "myMachine2",
919     "description": "My second machine",
920     "machineTemplate": {
921         "volumes": [
922             { "initialLocation": "V",
923               "href": "http://example.com/volumes/voll" }
924         ]
925     }
926 }
```

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

```
929
930 HTTP/1.1 201 Created
931 Location: http://example.com/machines/852108
```

## 932 4.5 Query new Machine

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

```
934 GET /machines/852108 HTTP/1.1
935
936 HTTP/1.1 200 OK
937 Content-Type: application/json
938
939 { "resourceType": "http://schemas.dmtf.org/cimi/1/Machine",
940     "id": "http://example.com/machines/852108",
941     "name": "myMachine2",
942     "description": "My second machine",
943     "created": "2012-03-26T10:04:00Z",
944     "updated": "2012-03-26T10:04:00Z",
945     "state": "STOPPED",
946     "cpu": "1",
947     "memory": 4000000,
948     "disks" : { "href": "http://example.com/machines/852108/disks",
949     "volumes": { "href": "http://example.com/machines/852108/volumes" },
950     "networkInterfaces": { "href": "http://example.com/machines/852108/NIs",
951     "operations": [
952         { "rel": "edit", "href": "http://example.com/machines/852108" },
```

```
953     { "rel": "delete", "href": "http://example.com/machines/852108" },
954     { "rel": "http://schemas.dmtf.org/cimi/1/action/start",
955       "href": "http://example.com/machines/852108" }
956   ]
957 }
```

958  
959 Notice the "state" attribute on the Machine is "STOPPED", which is the initial state of a new machine.

## 960 5 Defining and using System Templates

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

### 964 5.1 Retrieve the CEP

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

```
967 GET /CEP HTTP/1.1
```

```
968  
969 HTTP/1.1 200 OK
```

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

```
971  
972 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
973   "id": "http://example.com/CEP",
974   "baseURI": "http://example.com/",
975   "systemTemplates": { "href": "http://example.com/systemTemplates" },
976   "machineTemplates": { "href": "http://example.com/machineTemplates" },
977   "credentialTemplates": { "href": "http://example.com/credentialTemplates" },
978   "volumeTemplates": { "href": "http://example.com/volumeTemplates" }
979 }
```

### 980 5.2 Create a new System Template

981 A SystemTemplate is defined so that when instantiated the result is a Machine is created, a Volume is  
982 connected to the Machine, and a Credential resource exists. To achieve this configuration, the following  
983 are included: a SystemTemplate definition, a MachineTemplate by value, a VolumeTemplate by  
984 reference, and a CredentialTemplate by reference. The VolumeTemplate and CredentialTemplate  
985 resources are already available:

986 VolumeTemplate:

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

988 CredentialTemplate:

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

990 Note Alternatively, the VolumeTemplate and CredentialTemplate may be included by value in the  
991 MachineTemplate definition below. However, it is beneficial to immediately see in the SystemTemplate the resources  
992 that are involved and in general, automatic creation of the credential is more secure.

993 Before creating a SystemTemplate, the URL to which the POST is sent needs to be determined. This  
994 location is obtained from the SystemTemplate collection URL that was returned as part of the CEP.

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

```
996
```



```
997 HTTP/1.1 200 OK
998 Content-Type: application/json
999 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplateCollection",
1000   "id": "http://example.com/systemTemplates",
1001   "operations": [
1002     { "rel": "add", "href": "http://example.com/systemTemplates" }
1003   ]
1004 }
```

1005 **Now create the new System Template resource:**

```
1006 POST /systemTemplates HTTP/1.1
1007 Content-Type: application/json
1008
1009 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
1010   "name": "System Demo1",
1011   "description": "My first system template demo",
1012   "componentDescriptors": [
1013     { "name": "MyMachine",
1014       "type": "http://schemas.dmtf.org/cimi/1/Machine",
1015       "machineTemplate":
1016         { "name" : "Machine in system demo",
1017           "description" : "Machine in system",
1018           "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
1019           "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
1020           "credential": { "href": "#MyCredential" },
1021           "volumes": [
1022             { "initialLocation": "/vol",
1023               "href": "#MyVolume"
1024             }
1025           ]
1026         }
1027     },
1028     { "name": "MyCredential",
1029       "type": "http://schemas.dmtf.org/cimi/1/Credential",
1030       "credentialTemplate":
1031         { "href": "http://example.com/credentialTemplates/72000" }
1032     },
1033     { "name": "MyVolume",
1034       "type": "http://schemas.dmtf.org/cimi/1/Volume",
1035       "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839" }
1036     }
1037   ]
1038 }
```

```
1040 HTTP/1.1 201 Created
1041 Location: http://example.com/systemTemplates/48920
```

### 1042 **5.3 Create a new System by using a System Template**

1043 Now create a new System by using this System Template:

```
1044 POST /systems HTTP/1.1
1045 Content-Type: application/json
1046
1047 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCreate",
1048   "name": "MySystem1",
1049   "description": "My first system",
1050   "systemTemplate": { "href": "http://example.com/systemTemplates/48920" }
1051 }
1052
1053 HTTP/1.1 201 Created
1054 Location: http://example.com/systems/78342
```

1055 Note that, alternatively, the provider could have decided to return a reference to a Job resource instead of  
1056 waiting until the System is completely created. Instead of the above 201 response, this type of request  
1057 could have resulted in the following response:

```
1058 HTTP/1.1 202 Accepted
1059 CIMI-Job-URI: http://example.com/Jobs/90001
1060
1061 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Job",
1062   "id": "http://example.com/Jobs/90001",
1063   "name": "SystemCreationJob",
1064   "created": "2012-03-15T12:15:00Z",
1065   "updated": "2012-03-15T12:15:00Z",
1066   "targetResource": { "href": "http://example.com/systems" },
1067   "affectedResources": {
1068     "href": "http://example.com/systems/110001",
1069   },
1070   "action": "add",
1071   "status": "RUNNING",
1072   "progress": 30,
1073   "timeOfStatusChange": "2012-03-15T12:15:00Z",
1074   "isCancellable": "true",
1075   "nestedJobs": [
1076     { "href": "http://example.com/Jobs/90002" },
1077     { "href": "http://example.com/Jobs/90003" }
1078   ]
1079 }
```

1080 According to this response, the provider chose to create two nested Jobs and the "affectedResources"  
1081 attribute includes a reference to the newly created System. Periodic retrieval of the Job's representation  
1082 allows the Consumer to determine when the Job is completed, i.e., it is completed when the "progress"  
1083 attribute has a value of 100.

## 1084 5.4 Query the new System

1085 Retrieve the System to get the full representation of the new System:

```
1086 GET /systems/87342
1087
1088 HTTP/1.1 200 OK
1089 Content-Type: application/json
1090
```

```

1091 { "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
1092     "id": "http://example.com/systems/78342",
1093     "name": "MySystem1",
1094     "description": "My first system",
1095     "created": "2012-08-15T12:15:00Z",
1096     "updated": "2012-08-15T12:15:00Z",
1097     "state": "STOPPED",
1098     "machines": { "href": "http://example.com/systems/87432/machines" },
1099     "credentials": { "href": "http://example.com/systems/87342/creds" },
1100     "volumes": { "href": "http://example.com/systems/87342/vols" },
1101     "operations" : [
1102         { "rel": "edit", "href": "http://example.com/systems/78342" }
1103     ]
1104 }

```

## 1105 6 Editing System Templates

1106 In this scenario a second Machine is added to an existing System Template.

### 1107 6.1 Edit an existing System Template

1108 Edit the System Template created in a previous scenario and add another machine that shares its  
 1109 credential and volume resources:

1110 SystemTemplate:

1111 <http://example.com/systemTemplates/48920>

1112 Retrieve the existing SystemTemplate definition:

1113 GET /systemTemplates/48920 HTTP/1.1

```

1114
1115 HTTP/1.1 200 OK
1116 Content-Type: application/json
1117 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
1118     "name": "System Demo1",
1119     "description": "My first system template demo",
1120     "created": "2012-08-15T12:15:00Z",
1121     "updated": "2012-08-15T12:15:00Z",
1122     "componentDescriptors": [
1123         { "name": "MyMachine",
1124           "type": "http://schemas.dmtf.org/cimi/1/Machine",
1125           "machineTemplate":
1126             { "name" : "Machine in system demo",
1127               "description" : "Machine in system",
1128               "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
1129               "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
1130               "credential": { "href": "#MyCredential" },
1131               "volumes": [
1132                 { "initialLocation": "/vol",
1133                   "href": "#MyVolume"
1134               }

```

```

1135     ]
1136   }
1137 },
1138 { "name": "MyCredential",
1139   "type": "http://schemas.dmtf.org/cimi/1/Credential",
1140   "credentialTemplate":
1141     { "href": "http://example.com/credentialTemplates/72000" }
1142 },
1143 { "name": "MyVolume",
1144   "type": "http://schemas.dmtf.org/cimi/1/Volume",
1145   "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839"}
1146 }
1147 ],
1148 "operations" : [
1149   { "rel": "edit", "href": "http://example.com/systemTemplates/48920" }
1150 ]
1151 }

```

1152 **Now update the SystemTemplate resource:**

```

1153 PUT /systemTemplates/48920 HTTP/1.1
1154 Content-Type: application/json
1155
1156 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
1157   "name": "System Demo1",
1158   "description": "My first system template demo",
1159   "componentDescriptors": [
1160     { "name": "MyMachine",
1161       "type": "http://schemas.dmtf.org/cimi/1/Machine",
1162       "machineTemplate":
1163         { "name" : "Machine in system demo",
1164           "description" : "Machine in system",
1165           "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
1166           "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
1167           "credential": { "href": "#MyCredential" },
1168           "volumes": [
1169             { "initialLocation": "/vol",
1170               "href": "#MyVolume",
1171             }
1172           ]
1173         },
1174       "quantity": 2
1175     },
1176     { "name": "MyCredential",
1177       "type": "http://schemas.dmtf.org/cimi/1/Credential",
1178       "credentialTemplate":
1179         { "href": "http://example.com/credentialTemplates/72000" }
1180     },
1181     { "name": "MyVolume",
1182       "type": "http://schemas.dmtf.org/cimi/1/Volume",

```

```

1183     "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839"}
1184   }
1185 ]
1186 }
1187
1188 HTTP/1.1 200 OK
1189 Content-Type: application/json
1190
1191 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
1192   "name": "System Demo1",
1193   "description": "My first system template demo",
1194   "created": "2012-09-15T12:15:00Z",
1195   "updated": "2012-09-15T12:15:00Z",
1196   "componentDescriptors": [
1197     { "name": "MyMachine",
1198       "type": "http://schemas.dmtf.org/cimi/1/Machine",
1199       "machineTemplate":
1200         { "name" : "Machine in system demo",
1201           "description" : "Machine in system",
1202           "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
1203           "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
1204           "credential": { "href": "#MyCredential" },
1205           "volumes": [
1206             { "initialLocation": "/vol",
1207               "href": "#MyVolume"
1208             }
1209           ]
1210         },
1211     { "name": "MyCredential",
1212       "type": "http://schemas.dmtf.org/cimi/1/Credential",
1213       "credentialTemplate":
1214         { "href": "http://example.com/credentialTemplates/72000" }
1215     },
1216     { "name": "MyVolume",
1217       "type": "http://schemas.dmtf.org/cimi/1/Volume",
1218       "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839"}
1219     }
1220   ],
1221   "operations": [
1222     { "rel": "edit", "href": "http://example.com/systemTemplates/48920" }
1223   ]
1224 }
1225
1226 }

```

## 1227 6.2 Create a new System using a System Template

1228 Now create a new System using this System Template by sending a POST to the same URL that you  
 1229 used in the previous scenario to create a new System:

```
1230 POST /systems HTTP/1.1
```

```
1231 Content-Type: application/json
1232
1233 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCreate",
1234   "name": "MySystem2",
1235   "description": "My second system",
1236   "systemTemplate": { "href": "http://example.com/systemTemplates/48920" }
1237 }
1238
1239 HTTP/1.1 201 Created
1240 Location: http://example.com/systems/78343
```

### 1241 6.3 Query the new System

1242 Retrieve the System to get the full representation of the new System:

```
1243 GET /systems/87343
1244
1245 HTTP/1.1 200 OK
1246 Content-Type: application/json
1247
1248 { "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
1249   "id": "http://example.com/systems/78342",
1250   "name": "MySystem2",
1251   "description": "My second system",
1252   "created": "2012-10-15T12:15:00Z",
1253   "updated": "2012-10-15T12:15:00Z",
1254   "state": "STOPPED",
1255   "machines": { "href": "http://example.com/systems/87432/machines" },
1256   "credentials": { "href": "http://example.com/systems/87342/creds" },
1257   "volumes": { "href": "http://example.com/systems/87342/vols" },
1258   "operations" : [
1259     { "rel": "edit", "href": "http://example.com/systems/78342" }
1260   ]
1261 }
```

## 1262 7 Creating a Public Facing Network

1263 This clause is intended to provide guidance for basic network creation and deployment.

1264 This scenario is to describe the actions required to create a Public (Internet Facing Network). Due to  
1265 serialization, start with the network first unless the Provider offers a default network "starting point".

### 1266 7.1 Retrieve the Cloud Entry Point (CEP)

1267 The CEP provides the links to the network resources that are available.

```
1268 GET /CEP HTTP/1.1
1269
1270 HTTP/1.1 200 OK
1271 Content-Type: application/json
1272 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
1273   "id": "http://example.com/CEP",
1274   "baseURI": "http://example.com/",
```

```

1275 "resourceMetadata": { "href": "http://example.com/resourceMetadata" },
1276 "networks": { "href": "http://example.com/networks" },
1277 "networkConfigs": { "href": "http://example.com/networkConfigs" },
1278 "networkPorts": { "href": "http://example.com/networkPorts" },
1279 "networkPortConfigs":
1280   { "href": "http://example.com/networkPortConfigs" }
1281 }

```

1282 For additional details about beginning to use CIMI at the CEP, refer to the opening scenarios of this  
 1283 Primer.

## 1284 7.2 Verify provider examples for network configurations

```
1285 GET /networkConfigs HTTP/1.1
```

```

1286
1287 HTTP/1.1 200 OK
1288 Content-Type: application/json
1289
1290 { "resourceURI":
1291   "http://schemas.dmtf.org/cimi/1/NetworkConfigurationCollection",
1292   "id": "http://example.com/networkConfigs",
1293   "count": 3,
1294   "networkConfigurations": [
1295     { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkConfiguration",
1296       "id": "http://example.com/networkConfigs/InternetFacingNetwork",
1297       "name": "Public Access 1",
1298       "description": "internet reachable",
1299       "created": "2013-07-07T12:00:00Z",
1300       "updated": "2013-07-07T12:00:00Z",
1301       "networkType": "PUBLIC",
1302       "mtu": 1492,
1303       "classOfService": "GOLD"
1304     },
1305
1306     { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkConfiguration",
1307       "id": "http://example.com/networkConfigs/RED1",
1308       "name": "Private Red Network",
1309       "description": "Red Network",
1310       "created": "2013-07-07T12:00:00Z",
1311       "updated": "2013-07-07T12:00:00Z",
1312       "networkType": "PRIVATE",
1313       "mtu": 1500,
1314       "classOfService": "SILVER"
1315     },
1316
1317     { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkConfiguration",
1318       "id": "http://example.com/networkConfigs/BLUE2",
1319       "name": "Private Blue Network",
1320       "description": "Blue Network",
1321       "created": "2013-07-07T12:00:00Z",
1322       "updated": "2013-07-07T12:00:00Z",

```

```

1323     "networkType": "PRIVATE",
1324     "mtu": 1500,
1325     "classOfService": "BRONZE"
1326   }
1327 ]
1328 }

```

1329 Locate the Internet facing example network with a networkType "PUBLIC". Note the return information  
 1330 because this information is used later.

1331 Various MTU sizes are often supported within an end-to-end network infrastructure. These MTU sizes  
 1332 vary due to encryption, protocol requirements, and equipment manufacturer limitations.

### 1333 7.3 Retrieve a list of network port configurations

```
1334 GET /networkPortConfigs HTTP/1.1
```

1335

```
1336 HTTP/1.1 200 OK
```

```
1337 Content-Type: application/json
```

1338

```

1339 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortConfigurationCollection",
1340   "id": "http://example.com/networkPortConfigs",
1341   "count": 2,
1342   "networkPortConfigurations": [
1343     {"resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration",
1344      "id": "http://example.com/networkPortConfigs/GoldAccessPort",
1345      "name": "Gold Access Port",
1346      "description": "Enhanced Access Port",
1347      "created": "2013-07-07T12:00:00Z",
1348      "updated": "2013-07-07T12:00:00Z",
1349      "portType": "ACCESS",
1350      "classOfService": "GOLD"
1351     },
1352
1353     {"resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration",
1354      "id": "http://example.com/networkPortConfigs/BaseAccessPort",
1355      "name": "Base Access Port",
1356      "description": "AccessPort",
1357      "created": "2013-07-07T12:00:00Z",
1358      "updated": "2013-07-07T12:00:00Z",
1359      "portType": "ACCESS",
1360      "classOfService": "BRONZE"
1361     }
1362   ]
1363 }

```

1364 Locate the network configuration with the classOfService attribute of GOLD. This network  
 1365 configuration is used in a later example.

### 1366 7.4 Creating a PUBLIC accessible network

1367 Retrieve the CEP network collection so that you know to where to POST a new network:



1368 GET /networks HTTP/1.1

1369 HTTP/1.1 200 OK

1371 Content-Type: application/json

```
1372 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkCollection",
1373   "id": "http://example.com/networks",
1374   "count": 0,
1375   "operations": [ { "rel": "add", "href": "http://example.com/networks" } ]
1376 }
1377
```

1378

1379 POST /networks HTTP/1.1

1380 Content-Type: application/json

```
1381 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkCreate",
1382   "name": "PublicFacingNetwork",
1383   "description": "Public facing network",
1384   "networkTemplate": {
1385     "networkConfig":
1386       { "href": "http://example.com/networkConfigs/InternetFacingNetwork" }
1387   }
1388 }
1389
```

1390

1391 HTTP/1.1 201 Created

1392 Location: http://example.com/networks/255111

## 1393 7.5 Verify that a PUBLIC network has been created

1394 GET /networks/255111 HTTP/1.1

1395

1396 HTTP/1.1 200 OK

1397 Content-Type: application/json

```
1398 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Network",
1399   "id": "http://example.com/networks/255111",
1400   "name": "PublicFacingNetwork",
1401   "description": "Public facing network",
1402   "created": "2013-07-07T12:15:00Z",
1403   "updated": "2013-07-07T12:15:00Z",
1404   "state": "STARTED",
1405   "networkType": "PUBLIC",
1406   "mtu": 1492,
1407   "classOfService" : "GOLD",
1408   "networkPorts": { "href": "http://example.com/networks/255111/networkPorts" },
1409   "operations": [
1410     { "rel": "edit", "href": "http://example.com/networks/255111" },
1411     { "rel": "delete", "href": "http://example.com/networks/255111" },
1412     { "rel": "http://schemas.dmtf.org/cimi/1/action/start",
1413       "href": "http://example.com/networks/255111" }
1414   ]
1415 }
1416
```

1417 The networkPorts collection is an empty collection at this time.

## 1418 7.6 Create a network port

1419 Retrieve the CEP `networkPorts` collection so that you know to where to POST:

```
1420 GET /networkPorts HTTP/1.1
```

```
1421
1422 HTTP/1.1 200 OK
1423 Content-Type: application/json
1424
1425 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortCollection",
1426   "id": "http://example.com/networkPorts",
1427   "count": 0,
1428   "operations": [ { "rel": "add", "href": "http://example.com/networkPorts" } ]
1429 }
```

1430 Create a new network port. Note that the network must be indicated:

```
1431 POST /networkPorts HTTP/1.1
```

```
1432 Content-Type: application/json
```

```
1433
1434 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortCreate",
1435   "name": "publicNetworkPort1",
1436   "description": "Port 1 of the public network",
1437   "networkPortTemplate": {
1438     "network": { "href": "http://example.com/networks/255111" },
1439     "networkPortConfig": { "href":
1440 "http://example.com/networkPortConfigs/GoldAccessPort" }
1441   }
1442 }
```

```
1443
1444 HTTP/1.1 201 Created
```

```
1445 Location: http://example.com/networkPorts/885412
```

## 1446 7.7 Create a Machine attached to the public network

1447 The previous sections in the Primer have illustrated how to create a machine. It is not the intent to  
 1448 duplicate that material but to present here those properties and attributes that should be queried or  
 1449 configured or both to connect a machine to a network by using CIMI.

1450 Create a machine indicating a network to which to connect and a network port to use:

```
1451 POST /machines HTTP/1.1
```

```
1452 Content-Type: application/json
```

```
1453
1454 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate",
1455   "name": "myMachine1",
1456   "description": "My connected machine",
1457   "machineTemplate": {
1458     "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
1459     "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
1460     "credential": { "href": "http://example.com/creds/12345" },
1461     "networkInterfaces": [
1462       { "network": { "href": "http://example.com/networks/255111" },
1463         "networkPort": { "href": "http://example.com/networkPorts/885412" }
1464     ]
1465   }
1466 }
1467 }
```

```
1468
1469 HTTP/1.1 201 Created
```

```
1470 Location: http://example.com/machines/885412
```

1471 Query the created machine and the `networkInterfaces` collection within it by using a `$expand` query  
 1472 parameter:

```

1473 GET /machines/885412?$expand=networkInterfaces HTTP/1.1
1474
1475 HTTP/1.1 200 OK
1476 Content-Type: application/json
1477
1478 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
1479   "id": "http://example.com/machines/885412",
1480   "name": "myMachine1",
1481   "description": "My connected machine",
1482   "created": "2012-08-15T12:15:00Z",
1483   "updated": "2012-08-15T12:15:00Z",
1484   "state": "STOPPED",
1485   "cpu": 1,
1486   "memory": 4000000,
1487   "disks" : { "href": "http://example.com/machines/885412/disks"},
1488   "networkInterfaces": {
1489     "resourceURI":
1490 "http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceCollection",
1491     "id": "http://example.com/machines/885412/networkInterfaces",
1492     "href": "http://example.com/machines/885412/networkInterfaces",
1493     "count":1,
1494     "networkInterfaces": [
1495       {
1496         "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineNetworkInterface",
1497         "id": "http://example.com/machines/885412/networkInterfaces/1",
1498         "addresses": {"href":
1499 "http://example.com/machines/885412/networkInterfaces/1/addresses"},
1500         "network": {"href": "http://example.com/networks/255111"},
1501         "networkPort": {"href": "http://example.com/networkPorts/885412"},
1502         "state": "PASSIVE"
1503       }
1504     ]
1505   },
1506   "operations": [
1507     { "rel": "edit", "href": "http://example.com/machines/885412"},
1508     { "rel": "delete", "href": "http://example.com/machines/885412"},
1509     { "rel": "http://schemas.dmtf.org/cimi/1/action/start",
1510       "href": "http://example.com/machines/885412" }
1511   ]
1512 }

```

1513 Because we did not provide any address within the networkInterface section of the machine template  
 1514 when we created the machine, the provider must have allocated one. Query the address collection to  
 1515 verify it:

```

1516 GET /machines/885412/networkInterfaces/1/addresses HTTP/1.1
1517
1518 HTTP/1.1 200 OK
1519 Content-Type: application/json
1520
1521 { "resourceURI":
1522 "http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddressCollection",
1523   "id": "http://example.com/machines/885412/networkInterfaces/1/addresses",
1524   "count": 1,
1525   "machineNetworkInterfaceAddresses": [
1526     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddress",
1527       "id": "http://example.com/machines/885412/networkInterfaces/1/addresses/344277",
1528       "name": "Address 1003",
1529       "description": "Public address",
1530       "address": { "href": "http://example.com/addresses/53234" }
1531     }
1532   ]
1533 }

```

1534 Details about the allocated address can be retrieved by querying the referenced address:

1535

```
1536 GET /addresses/53234 HTTP/1.1
```

1537

```
1538 HTTP/1.1 200 OK
```

```
1539 Content-Type: application/json
```

1540

```
1541 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Address",
```

```
1542   "id": "http://example.com/addresses/53234",
```

```
1543   "name": "192.0.2.240 (Public)",
```

```
1544   "description": "Public address",
```

```
1545   "ip": "192.0.2.240",
```

```
1546   "allocation": "dynamic",
```

```
1547   "protocol": "IPv4",
```

```
1548   "network": {"href": "http://example.com/networks/255111"},
```

```
1549   "resource": {"href": "http://example.com/machines/885412/networkInterfaces/1"}
```

```
1550 }
```

1551 This step provides a subnet structure that maps to real-world use of Layer 3 Services. Among other  
1552 options, the Address can be provided by the provider or consumer. This option varies with each vendor,  
1553 provider, implementation, and network architecture in use.

## 1554 8 Provider responses and return values

### 1555 8.1 Unrecognized attributes

1556 A syntax error in the filter expression results in an error being generated. The provider returns a 400 'Bad  
1557 Request' response to a query with an unrecognized attribute name used in the `$filter` query  
1558 parameter.

1559 For example:

```
1560 GET /machines?$filter=aaa='bbb' HTTP/1.1
```

1561 The Machine resource has no "aaa" attribute: The recommended action is to return a 400 'Bad Request'  
1562 because the consumer did not follow `$filter`'s syntax, i.e., "aaa" is not a "resource attribute name".

### 1563 8.2 Unreasonable requests

1564 CIMI clients are expected to make reasonable requests. CIMI providers are not required to advertise  
1565 maximum attributes for all resources. A CIMI provider may set limits on the length of attribute values it  
1566 finds reasonable. It may reject a request it deems unreasonable. This is common practice in web-based  
1567 protocols today.

1568 These limits may not all be advertised in the ResourceMetadata, although it is recommended that CIMI  
1569 providers do so. A CIMI provider that receives a request that exceeds any of these limits, returns a  
1570 response with an appropriate standard HTTP status code, e.g., HTTP return code 413 'Request Entity  
1571 Too Large'.

**ANNEX A**  
(informative)**Change log**1572  
1573  
1574  
1575

1576

Version	Date	Description
1.0.0	2012-08-28	
1.0.1	2012-09-12	Errata
1.1.0	2014-07-31	DMTF Informational release

1577