Document Number: DSP0264

Version: 1.0.0

Date: 2012-12-14

Cloud Infrastructure Management Interface - Common Information Model (CIMI-CIM)

A CIM Representation of the CIMI Model

Document Type: Specification

Document Status: DMTF Standard

Document Language: en-US

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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

Tables

Foreword

This document is a deliverable from the DMTF Cloud Management Working Group. It defines a CIM representation for the Cloud Infrastructure Management Interface [CIMI] logical model. See the CIMI specification [DSP0263] for more information. This document assumes that the reader is familiar with the concepts defined in the *CIM Infrastructure Specification 2.7* (DSP0004).

Acknowledgments

The DMTF acknowledges the following individuals for their contributions to this document:

Editors:

- Bankston, J. Keith – Microsoft Corporation
- Burkhart, Nathan - Microsoft Corporation
- Cohen, Josh - Microsoft Corporation
- Davis, Jim – WS, Inc.
- Ericson, George – EMC

Contributors:

- Ali, Ghazanfar - ZTE Corporation
- Andreou, Marios - Red Hat
- Bankston, J. 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, Bhumip - 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 - Open Grid Forum
- 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

150

Cloud Infrastructure Management Interface - Common Information Model (CIMI-CIM)

1 Scope

This document makes use of the common meta-model used by CIM, the Common Information Model to describe the CIMI logical model. This is defined in DSP004, *CIM Infrastructure Specification 2.7.*

1.1 Typographical conventions

This specification uses the following conventions in the descriptive text:

- Any name that is usable as a type (embedded structures as well as atopic types such as “integer”, “string”) are in italic.
- Resource names and class names are in fixed-width font in the body text. In headers, the default font for the clause header will be used.
- Attribute names and qualifiers are in regular font, with the first character capitalized.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For references without a date or version, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

DMTF DSP0004, *Common Information Model (CIM) Infrastructure version 2.7*, http://dmtf.org/sites/default/files/standards/documents/DSP0004_2.7.0.pdf

DMTF DSP0223, *Generic Operations 1.0*, http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf

DMTF DSP0263, *Cloud Infrastructure Management Interface (CIMI) Model and REST Interface over HTTP, An Interface for Managing Cloud Infrastructure version 1.0.0*, http://dmtf.org/sites/default/files/standards/documents/DSP0263_1.0.0.pdf

DMTF DSP1001, *Management Profile Specification Usage Guide 1.1,* http://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf

ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards,* http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype

3 Terms and definitions

In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause.

The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"), "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Annex H. The terms in parenthesis are alternatives for the preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that ISO/IEC Directives, Part 2, Annex H specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning.

The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 5.

The terms "normative" and "informative" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do not contain normative content. Notes and examples are always informative elements.

The terms defined in DSP0004, DSP0223, and DSP1001 apply to this document. The following additional terms are used in this document.

3.1

CIM (Common Information Model)

CIM (Common Information Model) defined by DSP0004 as:

1. The name of the metamodel used to define schemas (e.g., the CIM schema or extension schemas).
2. The name of the schema published by the DMTF (i.e., the CIM schema).

This specification describes the translation between the CIM metamodel and CIMI Resources.

3.2

CIM Schema

The schema published by the DMTF that defines the Common Information Model. It is divided into a core model and a common model. Extension schemas are defined outside of the DMTF and are not considered part of the CIM schema.

3.3

MOF (Managed Object Format)

A DMTF-defined language used to create metamodel conformant representations of model elements. The Managed Object Format (MOF) is an Interface Definition Language (IDL) based on ISO/IEC 14750:1999. CIM Infrastructure specification (DSP0004) ANNEX A provides a complete description of the MOF language.

3.4

Ordinary class

A class that is neither an association class nor an indication class.

4 CIMI CIM translation

Transformation of the CIMI CIM into CIM metamodel conformant representations enables access of the services defined by CIMI in CIM-based environments. Such environments encompass a broad range of supported operating systems, languages, platforms, protocols, and other technologies.

This specification describes transformations in a manner that enables any CIM metamodel conformant representation. This document will utilize MOF for examples of such transformations.

4.1 CIM formal model

CIM representations of model resources are independent of access protocol and implementation technologies.

The use of CIM representations enables CIMI resources to be managed together with other key cloud foundation resources, such as storage, virtual machines, hardware, and operating systems that also use CIM representations.

A conformant CIMI CIM Service provider shall provide CIM representations of CIMI resources that are consistent with the formal definitions of the CIMI model according to the transformations described in this specification.

The DMTF provides MOF representations of CIMI resources that are transformed according to this specification.

NOTE Although some of the CIMI CIM classes correspond to existing CIM schema, for example `CIMI_Job`, no attempt has been made to derive from the CIM schema.

4.2 Translation rules

The following clauses define normative rules for translating between the CIMI resources as defined in the Cloud Infrastructure Management Interface (DSP0263) and their representation in CIM. Though all examples are represented using MOF format, this is only one of the formats that is used to represent CIM class definitions.

4.2.1 Common resource attributes

CIMI CIM ordinary classes inherit from a class named `CIMI_BaseElement`. This class defines the common attributes that are shared by all CIMI resources as described in DSP0263, clause 5.7.1.

The class definition for `CIMI_BaseElement` shall contain a property for each attribute defined in DSP0263, clause 5.7.1. These properties shall be derived using the Attribute translation rules defined in clause 4.2.1 except as noted below.

The *Id* attribute shall be a property of type *string*. The *Id* property shall have the Key qualifier. This property shall be the key property for all instances of CIMI ordinary classes.

See clause 5 for a non-normative reference of the MOF representation of `CIMI_BaseElement`.

4.2.2 Resource metadata

Resource metadata defined in DSP0263, clause 5.11 shall be defined in CIM following the rules defined below in clause 4.2.3. For the purposes of this document, resource metadata is information about provider-defined constraints, capabilities, or features. Resource metadata shall be represented in the same way as any other resource.

4.2.3 Resource translation rules

The rules described in this clause produce an ordinary class definition and some number of auxiliary structure and association definitions for each resource defined in DSP0263. The CIM classes represented by the MOF files in clause 5 conform to these rules.

Each CIMI resource is translated first to a CIM class definition. This will result in the definition of that class and some number other auxiliary structure, class, and association class definitions.

4.2.3.1 Ordinary class definitions

The schema name for ordinary class definitions shall be “CIMI” and the class name for each resource shall be the `name` of the resource as defined in DSP0263 and separated by an underscore, "_". For example, the CIMI resource `Machine` would translate to class named `CIMI_Machine`.

Each ordinary class shall inherit from `CIMI_BaseElement`, which defines the common attributes as specified in DSP0263, clause 5.7.1.

The following CIM qualifiers (Table 1) apply to each ordinary class definition.

Table 1: Qualifiers for ordinary classes

CIM Qualifier	Value
Description	The description qualifier shall be specified with the text following the heading of the clause that defines the resource in the CIMI specification.
UMLPackagePath	The UMLPackagePath qualifier shall be specified according to the following ABNF: `"CIMI:" resourceName` resourceName is the name of the corresponding CIMI resource.
Version	The version qualifier shall be specified with the value of the CIMI specification version.

Each attribute of a CIMI resource is translated into either a property or an association class definition. The following list defines the rules for how to translate the attribute:

1) If the attribute is a reference or a collection, a CIM association class is created as specified in 4.2.3.2.

2) If the attribute is a simple type, a CIM property is created with a primitive type as specified in 4.2.3.4.

3) If the attribute is a Map, a well-known structure named `CIMI_Map` is used; see 4.2.3.5. The property name shall be the same name as the CIMI attribute name and the data type shall be *string*. If the CIMI attribute is an array, the property shall be an array. Table 2 specifies the structure property qualifiers.

4) If the attribute is a structure, a CIM structure is created as specified in 4.2.3.3. The property name shall be the same name as the CIMI attribute name and the data type shall be *string*. If the CIMI attribute is an array, the property shall be an array. Table 2 specifies the structure property qualifiers.

Table 2: Structure property qualifiers

CIM Qualifier	Value
Description	The Description qualifier shall be specified with the text provided in the description of the attribute.
EmbeddedInstance	The EmbeddedInstance qualifier shall be specified with the name of the structure (e.g., `CIMI_Volume`).
Read	The Read qualifier shall be specified with value False if the Consumer Constraints listed in the description specifies "write-only".
Required	The Required qualifier shall be specified with no value if the Provider Constraints listed in the description specifies support mandatory.
Write	The Write qualifier shall be specified if the Consumer Constraints listed in the description specifies "read-write" or "write-only".

Each operation in DSP0263 that is not an intrinsic operation shall be included as a method in the CIM class definition. The following specifies how to map a method:

- Method Name - The method name in CIM shall be the link URL as defined in DSP0263 with the prefix “http://www.dmtf.org/cimi/action/” removed. For example, the operation supported by the Machine resource that is defined in DSP0263 with the link “http://www.dmtf.org/cimi/action/start” is defined in CIM with a method named start.
- Return Value – The return value shall be of type *uint32*.
- Input Parameters – If the method includes any input parameters, the name of the input parameter will be the same as the parameter name specified in the CIMI. The data type shall map the same as for Simple Properties; see 4.2.3.4. Table 3 specifies the rules for qualifiers for IN qualified parameters.

Table 3: IN parameter qualifiers

CIM Qualifier	Value
Description	The Description qualifier shall be specified with the text provided in the description of the parameter.
IN	The In qualifier shall be specified with a value of True.
Required	The Required qualifier shall be specified with no value if the parameter is specified as mandatory. This qualifier shall not be specified if the parameter is optional.
Units	The Units qualifier shall be specified if the description defines the value as a programmable unit listed in DSP0004 (e.g., KiloBytes, Percent, Seconds, …)
Values	The Values qualifier shall be specified if the parameter type is *string* and the description includes the phrase, "Allowable values include:" The qualifier value is the array of strings specified by the values listed in the description.

- Output Parameters – If the method includes any output parameters, the name of the input parameter will be the same as the parameter name specified in the CIMI. The data type shall map the same as for Simple Properties; see 4.2.3.4. Table 4 specifies the rules for qualifiers for OUT qualified parameters.

302 **Table 4: OUT parameter qualifiers**

CIM Qualifier	Value
Description	The Description qualifier shall be specified with the text provided in the description of the parameter.
IN	The IN qualifier shall be specified with a value of False.
OUT	The OUT qualifier shall be specified with a value of True.
Required	The Required qualifier shall be specified with no value if the parameter is specified as mandatory. This qualifier shall not be specified if the parameter is optional.
Units	The Units qualifier shall be specified if the description defines the value as a programmable unit listed in DSP0004 (e.g., KiloBytes, Percent, Seconds, …)
Values	The Values qualifier shall be specified if the parameter type is *string* and the description includes the phrase, "Allowable values include:" The qualifier value is the array of strings specified by the values listed in the description.

303

304 #### 4.2.3.2 Association class definitions

305 If the attribute of the CIMI resource (excluding structures) is a reference or a collection, an association
306 class shall be created. The association class name shall be the concatenation of “CIMI”, an underscore,
307 "_", the name of the resource as defined in DSP0263 and the corresponding CIMI attribute name with an
308 initial capital letter – for example, the association with the class name of `CIMI_MachineNetwork`. Table
309 5 specifies the rules for qualifiers for association classes.

310 **Table 5: Qualifiers for association classes**

CIM Qualifier	Value
Association	The Association qualifier shall be specified first and with no value.
Description	The Description qualifier shall be specified. The value should be the text “<classname> <attributename> association”, for example “CIM_Machine eventLog association”.
UMLPackagePath	The UMLPackagePath qualifier shall be specified with the value according to the following ABNF: `"CIMI:" resourceName referenceName` Where `resourceName` is the name of the CIMI resource that defines the CIMI attribute that is translated into the association class and `referenceName` is the name of the CIMI attribute that caused creation of this association. The `referenceName` is specified with an initial capital letter.
Version	The Version qualifier shall be specified with the value of the version of the CIMI specification.

311 The association shall include two reference properties. The first is a reference to the CIM class
312 representing the CIMI resource that included the reference or collection property. The description shall be
313 "The <classname>", where <classname> is the classname; for example, `CIMI_Machine`. The second
314 shall be a reference to the CIM class corresponding to the referenced or collected CIMI resource. The
315 description shall be the description of the original CIMI attribute. Table 6 specifies the rules for qualifiers
316 that apply to reference properties.

317 **Table 6: Reference property qualifiers**

CIM Qualifier	Value
Key	The Key qualifier shall be specified as the first qualifier with no arguments.
Description	The Description qualifier shall be specified with the text provided in the description of the attribute.
Min	The Min qualifier shall be specified if the minimum number of referenced instances is not 0.
Max	The Max qualifier shall be specified with a value if the maximum number of referenced instances is not unlimited. If the CIMI attribute is not an array, the Max qualifier shall be specified with a value of one (1).

318 #### 4.2.3.3 Structure definitions

319 A structure will be created if the attribute of a CIMI resource is a structure. The structure class name shall
320 be the concatenation of “CIMI”, an underscore, "_", the name of the resource as defined in DSP0263 and
321 the corresponding CIMI attribute name with an initial capital letter; for example,
322 `CIMI_MachineConfigurationDisks`.

323 The following CIM qualifiers (Table 7) apply to each structure definition.

324 **Table 7: Qualifiers for structures**

CIM Qualifier	Value
Indication	The Indication qualifier shall be specified with no arguments first.
Structure	The Structure qualifier shall be specified with no arguments second.
Description	The Description qualifier shall be specified with the text following the CIMI attribute that references this structure.
UMLPackagePath	The UMLPackagePath qualifier shall be specified with the value according to the following ABNF: `"CIMI:" resourceName` Where resourceName is the name of the corresponding CIMI resource.
Version	The Version qualifier shall be specified with the version of the CIMI specification.

325 For each attribute of the CIMI structure, a property shall be created. The following rules apply:

326 1) If the attribute has a simple type, it translates to a CIM property with a primitive type; see 4.2.3.4.

327 2) If the attribute is a ref, it translates the same as if it were a URI; see 4.2.3.4.

328 3) If the attribute is a `map`, a well-known structure named `CIMI_Map` is used; see 4.2.3.5. The
329 property name shall be the same name as the CIMI attribute name and the data type shall be
330 *string*. If the CIMI attribute is an array, the property shall be an array. Table 2 specifies the
331 structure property qualifiers.

332 4) If the attribute is a Structure, a CIM structure is created as specified in 4.2.3.3. The property
333 name shall be the same name as the CIMI attribute name and the data type shall be *string*. If the
334 CIMI attribute is an array, the property shall be an array. Table 2 specifies the structure property
335 qualifiers.

336 #### 4.2.3.4 Simple properties

337 The CIMI model defines a set of data types in (clause 5.5 of DSP0263).

338 Table 8 defines the translation between CIMI and CIM primitive types.

339 **Table 8: Primitive type mapping**

CIMI	MOF
boolean	*boolean*
dateTime	*datetime*
duration	*datetime*
integer	*uint8* *sint8* *uint16* *sint16* *uint32* *sint32* *uint64* *sint64*
string	*string*
byte[]	*uint8[]*
URI	*string*

340 The property name of a CIMI attribute with a primitive type shall be the same as the CIMI attribute name.
341 The property type shall be the CIM primitive type from Table 8. There are multiple mappings for the CIMI
342 *integer* type. The modeler may exercise judgment. However, if there is any doubt, *sint64* should be
343 chosen. If the CIMI specification attribute is an array, the CIM property shall be an array. Table 9 defines
344 qualifiers that apply to simple properties.

345 **Table 9: Simple property qualifiers**

CIM Qualifier	Value
Description	The Description qualifier shall be specified with the text provided in the description of the attribute.
Read	The Read qualifier shall be specified with value False if the Consumer Constraints listed in the description specifies "write-only".
Reference	The Reference qualifier shall be specified if the CIMI type is *URI*.
Required	The Required qualifier shall be specified with no value if the Provider Constraints listed in the description specifies support mandatory.
Units	The Units qualifier shall be specified if the description defines the value as a programmable unit listed in DSP0004 (e.g., KiloBytes, Percent, Seconds, …)
Values	The Values qualifier shall be specified if the attribute type is *string* and the description includes the phrase, "Allowable values include:" The qualifier value is the array of strings specified by the highlighted values listed in the description.
Write	The Write qualifier shall be specified with no value if the Consumer Constraints listed in the description specifies "read-write" or "write-only".

346 #### 4.2.3.5 Map

347 CIMI defines a *map* of key/value pairs. The following structure is used to represent a *map*.

```
[Indication, Structure, Version("1.0.0"),
    Description("CIMI Map"),
    UMLPackagePath ( "CIMI::Map" )]
CIMI_Map {

      [Description("The key.")]
   string Key;

      [Description("The value.")]
   string Value;
}
```

5 CIMI CIM MOF representation examples

The following clauses shows examples of CIMI entities represented as CIM MOF classes.

The normative CIM metamodel representations are published by the DMTF. The representations are published in MOF and other formats.

The following non-normative copies of the MOF files are provided for illustration. Where any differences occur between the published MOF files and the copies below, the published MOF files shall be considered authoritative.

The Cloud Infrastructure Management Interface classes are defined in a schema with the prefix CIMI and derived from a common root class `CIMI_BaseElement`, which does not derive from any DMTF standard CIM schema class.

5.1 Ordinary class

5.1.1 CIMI_BaseElement

Defined in: CIMI_BaseElement.mof

```
   [Abstract, Version ( "1.0.0" ),
    UMLPackagePath ( "CIMI::BaseElement" ),
    Description ( "Common properties for all CMWG classes" )]
class CIMI_BaseElement {

      [Key, Description (
          "The unique self-reference to this resource; assigned upon"
          "resource creation. This attribute value shall be unique in the"
          "Provider's cloud."
          )]
   string id;

      [Required, Write, Description (
          "The human readable name of this resource; assigned by the "
          "creator as a part of the resource creation input." )]
   string name;

      [Required, Write, Description (
```

```
          "The human readable description of this resource; assigned "
          "by the creator as a part of the resource creation input." )]
   string description;

      [Description (
          "The timestamp when this resource was created. The format "
          "should be unambiguous, and the value is immutable")]
   datetime created;

       [Description (
          "The time at which the last explicit attribute update "
        "was made on the resource. Note, while operations such "
          "as \"stop\" do implicitly modify the \"state\" attribute "
        "it does not change the \"updated_time\"." )]
   datetime updated;
};
```

5.1.2 CIMI_Machine

Defined in: CIMI_Machine.mof

```
   [Version("1.0.0"), Description(
      "An instantiated compute resource that encapsulates both CPU and Memory."),
    UMLPackagePath ( "CIMI::Machine" )]
Class CIMI_Machine : CIMI_BaseElement {

      [Required, Description(
          "The operational state of the Machine.\n"
          "Allowable values include:\n"
          "CREATING: The Machine is in the process of being created. "
          "Allowable action when in this state is: delete.\n"
          "STARTING: The Machine is in the process of being started. "
          "Allowable actions when in this state are: start, restart, "
          "stop, and delete.\n"
          "STARTED: The Machine is available and ready for use. Allowable actions "
          "when in this state are: stop, restart, pause, suspend, capture, "
          "and delete.\n"
          "STOPPING: The Machine is in the process of being stopped. Allowable "
          "actions when in this state are: start, restart, stop, and delete. "
          "STOPPED: This value is the virtual equivalent of powering off a physical "
          "Machine. There is no saved CPU or memory state. Allowable actions when "
          "in this state are: start, restart, capture, and delete.\n"
          "PAUSING: The Machine in the process of being PAUSED. Allowable actions "
          "when in this state are: start, restart, and delete.\n"
          "PAUSED: In this state the Machine and its virtual resources remain "
          "instantiated and resources remain allocated, similar to the STARTED "
          "state, but the Machine and its virtual resources are not enabled to "
          "perform tasks. Allowable actions when in this state are: start, restart, "
          "capture, and delete.\n"
          "SUSPENDING: The Machine is in the process of being suspended. Allowable "
```

```
          "actions when in this state are: start, restart, and delete.\n"
          "SUSPENDED: In this state the Machine and its virtual resources are stored "
          "on non-volatile storage. The Machine and its resources are not enabled to "
          "perform tasks. Allowable actions when in this state are: start, restart, "
          "capture, and delete.\n"
          "DELETING: The Machine is in the process of being deleted. Allowable "
          "action when in this state is: delete.\n"
          "ERROR: The Provider has detected an error in the Machine. Allowable "
          "actions when in this state are: start, restart, stop, and delete.\n"
          "PAUSED and SUSPENDED states are optional and Providers may choose to "
          "support them or not.\n"
          "Providers may define additional values.")
        Values{"CIMI_CREATING","CIMI_STARTING","CIMI_STOPPING","CIMI_STOPPED",
               "CIMI_PAUSING","CIMI_PAUSED","CIMI_SUSPENDING",
               "CIMI_SUSPENDED","CIMI_DELETING","CIMI_ERROR",
               "CIMI_PAUSED","CIMI_SUSPENDED"}]
   String state;

      [Description("The amount of CPU that this Machine has.")]
   Uint32 cpu;

      [Required,
       Description(
          "The size of the memory (RAM) allocated to this Machine.\n\n"
          "When this value is increased, it implies that the Machine is allocated "
          "more RAM, and vice versa when the value is decreased.")]
   Uint64 memory;

      [Description(
          "The CPU architecture that will be supported by Machines created by using "
          "this configuration.\n"
          "Allowable values include: 68000, Alpha, ARM, Itanium, MIPS, PA_RISC, "
          "POWER, PowerPC, x86, x86_64, z/Architecture, SPARC. Providers may define "
          "additional values."),
        Values{"CIMI_68000", "CIMI_Alpha", "CIMI_ARM", "CIMI_Itanium", "CIMI_MIPS",
               "CIMI_PA_RISC", "CIMI_POWER", "CIMI_PowerPC", "CIMI_x86",
               "CIMI_x86_64", "CIMI_z/Architecture", "CIMI_SPARC"}]
   String cpuArch;
};
```

5.1.3 CIMI_Disk

Defined in: CIMI_Disk.mof

```
   [Version("1.0.0"), Description(
      "The size of the memory (RAM) allocated to this Machine. "
      "When this value is increased, it implies that the Machine is allocated more "
      "RAM, and vice versa when the value is decreased. "
```

```
       "This attribute has the following sub-attributes that serve to describe it:")]
CIMI_Disk {

       [Required, Description(
          "The initial capacity, in kilobytes, of the disk. "),
        Units ( "KiloBytes" )]
   String capacity;

       [Description(
          "Operating System specific location(path) in its namespace where this disk "
          "will first appear. Note, once deployed Consumers might move where this "
          "Disk is located.\n"
          "Support of this attribute indicates that the Provider can report this "
          "information back to the Consumer.")]
   String initialLocation;
};
```

5.2 Association

5.2.1 CIMI_MachineEventLog

Defined in: CIMI_MachineEventLog.mof

```
   [Association, Version("1.0.0"),
    Description("CIMI_Machine eventLog association"),
    UMLPackagePath ( "CIMI::Machine" )]
CIMI_MachineEventLog {

       [Key, Description("The CIMI_Machine")]
   CIMI_Machine  REF machine;

       [Key, MAX(1), Description(
          "A reference to the EventLog of this Machine.")]
   CIMI_EventLog REF eventLog;
};
```

5.2.2 CIMI_MachineLatestSnapshot

Defined in: CIMI_MachineLatestSnapshot.mof

```
   [Association, Version("1.0.0"),
    Description("CIM_Machine latestSnapshot association"),
    UMLPackagePath ( "CIMI::Machine" )]
CIMI_MachineLatestSnapshot {

      [Key, Description("The CIMI_Machine")]
   CIMI_Machine  REF machine;

      [Key, Max(1), Description(
         "A reference to the SNAPSHOT representing the latest state captured for "
         "this Machine (either most recent Snapshot or the last Snapshot reverted "
         "to).")]
   CIMI_MachineImage REF latestSnapshot;
};
```

5.3 Structure

5.3.1 CIMI_MachineTemplateVolumes

Defined in: CIMI_MachineTemplateVolumes.mof

```
   [Indication, Structure, Version("1.0.0"),
    Description(
      "CIMI_Machine NetworkInterfaces association."),
    UMLPackagePath ( "CIMI::MachineTemplateVolumes" )]
CIMI_MachineTemplateVolumes {

      [Description(
         "An Operating System specific location(path) in its namespace where "
         "the Volume will appear. Support of this attribute indicates that the "
         "Provider allows for Consumers to choose where the Volume will appear.")]
   String initialLocation;

      [Required, Reference, Description(
         "Reference to the Volume that will be connected.")]
   String volume;

```

ANNEX A
(informative)
Change log

Version	Date	Description
1.0.0	2012-12-13	

Bibliography

DMTF DSP-IS0102, Distributed Management Task Force, Inc., *Architecture for Managing Clouds White Paper 1.0*, http://dmtf.org/sites/default/files/standards/documents/DSP-IS0102_1.0.0.pdf

DMTF DSP-IS0103, Distributed Management Task Force, Inc., *Use Cases and Interactions for Managing Clouds 1.0.0*, http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0103_1.0.0.pdf