



1

2

3

4

**Document Number: DSP1023**

**Date: 2009-06-17**

**Version: 1.0.1**

5 **Software Inventory Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

## 9 Copyright Notice

10 Copyright © 2008, 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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25 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

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

31

# CONTENTS

33	Foreword .....	7
34	Introduction .....	8
35	1 Scope .....	9
36	2 Normative References.....	9
37	2.1 Approved References .....	9
38	2.2 Other References.....	9
39	3 Terms and Definitions .....	9
40	4 Symbols and Abbreviated Terms.....	11
41	5 Synopsis.....	11
42	6 Description .....	12
43	7 Implementation.....	14
44	7.1 Representing Software .....	14
45	7.2 Representing Installed Software.....	14
46	7.3 Representing Version Information of Software.....	14
47	7.4 Representing Relationships between Software Identity and Managed Element.....	14
48	7.5 Finding the Scoping Instance of the CIM_System Class.....	16
49	7.6 Representing Available Software.....	16
50	7.7 Representing a Software Bundle .....	17
51	7.8 Identifying a Software Identity.....	18
52	7.9 Representing Installation Dependencies .....	19
53	7.10 Version Comparison Using the MajorVersion, MinorVersion, RevisionNumber, and	
54	BuildNumber Properties.....	19
55	8 Methods.....	20
56	8.1 Profile Conventions for Operations.....	20
57	8.2 CIM_SoftwareIdentity.....	20
58	8.3 CIM_InstalledSoftwareIdentity .....	20
59	8.4 CIM_ElementSoftwareIdentity .....	21
60	8.5 CIM_SystemSpecificCollection .....	22
61	8.6 CIM_HostedCollection .....	22
62	8.7 CIM_MemberOfCollection .....	22
63	8.8 CIM_SoftwareIdentityResource .....	23
64	8.9 CIM_SAPAvailableForElement.....	23
65	8.10 CIM_HostedAccessPoint .....	23
66	8.11 CIM_OrderedComponent .....	24
67	8.12 CIM_OrderedDependency.....	24
68	9 Use Cases.....	24
69	9.1 Object Diagrams .....	24
70	9.2 Find All the Software Installed on All the Managed Elements within the Scope of a	
71	Managed System .....	35
72	9.3 Find All the Software Installed on a Managed Element.....	35
73	9.4 Find All the Software That Is Compatible with a Managed Element but Has Not Been	
74	Installed.....	35
75	9.5 Find All the Software That Is Available for Installation on Any Managed Element within	
76	the Scope of a Managed System.....	35
77	9.6 For a Given NIC, Find the Driver That Is Running in the Operating System.....	36
78	9.7 Set a Particular Software Image on a Hardware Managed Element to Run After the Next	
79	Reset or Reboot.....	36
80	9.8 Set a Particular Software Image on a Hardware Managed Element to Run After the Next	
81	Reset or Reboot but Not After a Subsequent Reset or Reboot.....	36
82	9.9 Find and Set a Driver to Run After the Next Reset or Reboot for a NIC .....	36
83	9.10 Find the Most Recent Firmware Available for a NIC .....	37

84	9.11 Find the Most Recent Firmware Installed on a NIC.....	37
85	9.12 Find the Software Families of Which a Software Identity Is a Member .....	37
86	9.13 Determine Whether a Dependency of a Software Identity Is Satisfied.....	37
87	10 CIM Elements.....	38
88	10.1 CIM_SoftwareIdentity.....	38
89	10.2 CIM_InstalledSoftwareIdentity .....	39
90	10.3 CIM_ElementSoftwareIdentity .....	39
91	10.4 CIM_SystemSpecificCollection.....	40
92	10.5 CIM_HostedCollection .....	40
93	10.6 CIM_MemberOfCollection .....	40
94	10.7 CIM_SoftwareIdentityResource .....	41
95	10.8 CIM_SAPAvailableForElement.....	41
96	10.9 CIM_HostedAccessPoint.....	41
97	10.10 CIM_OrderedComponent .....	42
98	10.11 CIM_OrderedDependency.....	42
99	10.12 CIM_RegisteredProfile.....	42
100	ANNEX A (informative) Change Log.....	43

101

## 102 Figures

103	Figure 1 – Class Diagram: Software Inventory Profile.....	13
104	Figure 2 – Registered Profile .....	25
105	Figure 3 – Object Diagram Showing Installed Software .....	26
106	Figure 4 – Object Diagram Showing an Installed Driver.....	27
107	Figure 5 – Object Diagram Showing Installed BIOS.....	27
108	Figure 6 – Object Diagram Showing Installed Software .....	28
109	Figure 7 – Object Diagram Showing Multiple Installed Software on a Managed Element .....	29
110	Figure 8 – Object Diagram with No Instantiation of Managed Element.....	29
111	Figure 9 – Object Diagram Showing Available Firmware .....	30
112	Figure 10 – Object Diagram Showing an Available Driver.....	31
113	Figure 11 – Object Diagram Showing a Firmware Image and Its Location .....	32
114	Figure 12 – Object Diagram Showing a Software Bundle .....	33
115	Figure 13 – Object Diagram Showing Available Software That Is Part of a Software Bundle.....	34
116	Figure 14 – Object Diagram Showing Installed and Available Software .....	35

117

## 118 Tables

119	Table 1 – Related Profiles.....	12
120	Table 2 – Relationships Between Enumeration Values of ElementSoftwareStatus.....	15
121	Table 3 – Operations: CIM_InstalledSoftwareIdentity .....	20
122	Table 4 – Operations: CIM_ElementSoftwareIdentity.....	21
123	Table 5 – Operations: CIM_HostedCollection.....	22
124	Table 6 – Operations: CIM_MemberOfCollection.....	23
125	Table 7 – Operations: CIM_SAPAvailableForElement .....	23
126	Table 8 – Operations: CIM_HostedAccessPoint .....	23
127	Table 9 – Operations: CIM_OrderedComponent.....	24
128	Table 10 – Operations: CIM_OrderedDependency .....	24
129	Table 11 – CIM Elements: Software Inventory Profile.....	38
130	Table 12 – Class: CIM_SoftwareIdentity.....	38

131 Table 13 – Class: CIM\_InstalledSoftwareIdentity ..... 39

132 Table 14 – Class: CIM\_ElementSoftwareIdentity ..... 39

133 Table 15 – Class: CIM\_SystemSpecificCollection ..... 40

134 Table 16 – Class: CIM\_HostedCollection ..... 40

135 Table 17 – Class: CIM\_MemberOfCollection..... 40

136 Table 18 – Class: CIM\_SoftwareIdentityResource ..... 41

137 Table 19 – Class: CIM\_SAPAvailableForElement..... 41

138 Table 20 – Class: CIM\_HostedAccessPoint ..... 41

139 Table 21 – Class: CIM\_OrderedComponent..... 42

140 Table 22 – Class: CIM\_OrderedDependency ..... 42

141 Table 23 – Class: CIM\_RegisteredProfile..... 42

142



144

## Foreword

145 The *Software Inventory Profile* (DSP1023) was prepared by the Physical Platform Profiles Working Group  
146 and the Server Management Working Group.

147 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
148 management and interoperability.

149

## 150 **Acknowledgments**

151 The authors wish to acknowledge the following people.

### 152 **Editor:**

153 • RadhaKrishna R. Dasari – Dell, Inc. **Contributors:**

154 • Jon Hass – Dell, Inc.

155 • Khachatur Papanyan – Dell Inc.

156 • Aaron Merkin – IBM

157 • Jeff Hilland – Hewlett-Packard Corporation

158 • Christina Shaw – Hewlett-Packard Corporation

159 • Michael Tehranian – Sun Microsystems

160 • Perry G. Vincent – Intel Corporation

161 • John Leung – Intel Corporation

162 • Hemal Shah – Broadcom

163 • Larry Lamers - VMware

164

165

## Introduction

166 The information in this specification should be sufficient for a provider or consumer of this data to identify  
167 unambiguously the classes, properties, methods, and values that are instantiated and manipulated to  
168 identify and query the inventory of installed BIOS, firmware, drivers, and related software in a managed  
169 system. This profile also describes the Common Information Model (CIM) schema elements required to  
170 represent the software that can be installed on a managed system.

171 The target audience for this specification is implementers who are writing CIM-based providers or  
172 consumers of management interfaces that represent the component described in this document.



173

# Software Inventory Profile

## 174 1 Scope

175 The *Software Inventory Profile* describes the CIM schema elements required to provide an inventory of  
176 installed BIOS, firmware, drivers, and related software in a managed system. This profile also describes  
177 the CIM schema elements required to represent the software that can be installed on a managed system.

## 178 2 Normative References

179 The following referenced documents are indispensable for the application of this document. For dated  
180 references, only the edition cited applies. For undated references, the latest edition of the referenced  
181 document (including any amendments) applies.

### 182 2.1 Approved References

183 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,  
184 [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.5.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf)

185 DMTF DSP0200, *CIM Operations over HTTP 1.3*,  
186 [http://www.dmtf.org/standards/published\\_documents/DSP0200\\_1.3.pdf](http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf)

187 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,  
188 [http://www.dmtf.org/standards/published\\_documents/DSP1001\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf)

189 DMTF DSP1033, *Profile Registration Profile 1.0*,  
190 [http://www.dmtf.org/standards/published\\_documents/DSP1033\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf)

### 191 2.2 Other References

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

## 194 3 Terms and Definitions

195 For the purposes of this document, the following terms and definitions apply. For the purposes of this  
196 document, the terms and definitions given in [DSP1033](#) and [DSP1001](#) also apply.

### 197 3.1

#### 198 **can**

199 used for statements of possibility and capability, whether material, physical, or causal

### 200 3.2

#### 201 **cannot**

202 used for statements of possibility and capability, whether material, physical, or causal

### 203 3.3

#### 204 **conditional**

205 indicates requirements to be followed strictly to conform to the document when the specified conditions  
206 are met

- 207 **3.4**  
208 **mandatory**  
209 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
210 permitted
- 211 **3.5**  
212 **may**  
213 indicates a course of action permissible within the limits of the document
- 214 **3.6**  
215 **need not**  
216 indicates a course of action permissible within the limits of the document
- 217 **3.7**  
218 **optional**  
219 indicates a course of action permissible within the limits of the document
- 220 **3.8**  
221 **referencing profile**  
222 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
223 "Referenced Profiles" table
- 224 **3.9**  
225 **shall**  
226 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
227 permitted
- 228 **3.10**  
229 **shall not**  
230 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
231 permitted.
- 232 **3.11**  
233 **should**  
234 indicates that among several possibilities, one is recommended as particularly suitable, without  
235 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 236 **3.12**  
237 **should not**  
238 indicates that a certain possibility or course of action is deprecated but not prohibited
- 239 **3.13**  
240 **unspecified**  
241 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 242 **3.14**  
243 **Software Identity**  
244 an instance of CIM\_SoftwareIdentity that represents and contains the identifying property values of a  
245 software image
- 246 **3.15**  
247 **Installed Software**  
248 software that is installed on any managed element in the scope of a system

249 **3.16**250 **Available Software**

251 software that the management infrastructure has determined is available, either locally or at a remote  
252 location, for installation on the managed system and may be appropriate to install without any assertion  
253 about the ability to perform the installation through the management infrastructure

254 **3.17**255 **Software Bundle**

256 a software image that consists of one or more discrete software images that can be installed individually  
257 or together

258 **3.18**259 **Managed Element**

260 an instance of CIM\_ManagedElement that represents a managed element in the scope of a system

261 **3.19**262 **Software Family**

263 a group of software in which each member software could be installed in the place of the other on a  
264 Managed Element and offer similar functionality to a Managed Element

265 **3.20**266 **Installation Dependency**

267 a software image that needs to be installed before installing the target Software Identity

268 **4 Symbols and Abbreviated Terms**269 **4.1**270 **CIM**

271 Common Information Model

272 **4.2**273 **CIMOM**

274 CIM object manager

275 **5 Synopsis**

276 **Profile Name:** Software Inventory

277 **Version:** 1.0.1

278 **Organization:** DMTF

279 **CIM Schema Version:** 2.22

280 **Central Class:** CIM\_SoftwareIdentity

281 **Scoping Class:** CIM\_System

282 The *Software Inventory Profile* describes the classes and properties used to provide an inventory of  
283 installed BIOS, firmware, drivers, and related software in a managed system. This profile also describes  
284 the classes and properties required to represent the software that can be installed on a managed system.  
285 The profile defines the use of a Software Identity for representing the software image known to the  
286 managed system. The profile also defines the relationship between a Managed Element and the Software  
287 Identity that is applicable to that Managed Element.

288 CIM\_SoftwareIdentity shall be the Central Class of this profile. The instance of CIM\_SoftwareIdentity shall  
289 be the Central Instance of this profile.

290 CIM\_System shall be the Scoping Class of this profile. The instance of CIM\_System shall be the Scoping  
291 Instance of this profile and shall be selected using the algorithm described in section 7.5.

292 References to CIM\_System may be interpreted as references to subclasses of CIM\_System such as  
293 CIM\_ComputerSystem. Table 1 identifies profiles on which this profile has a dependency.

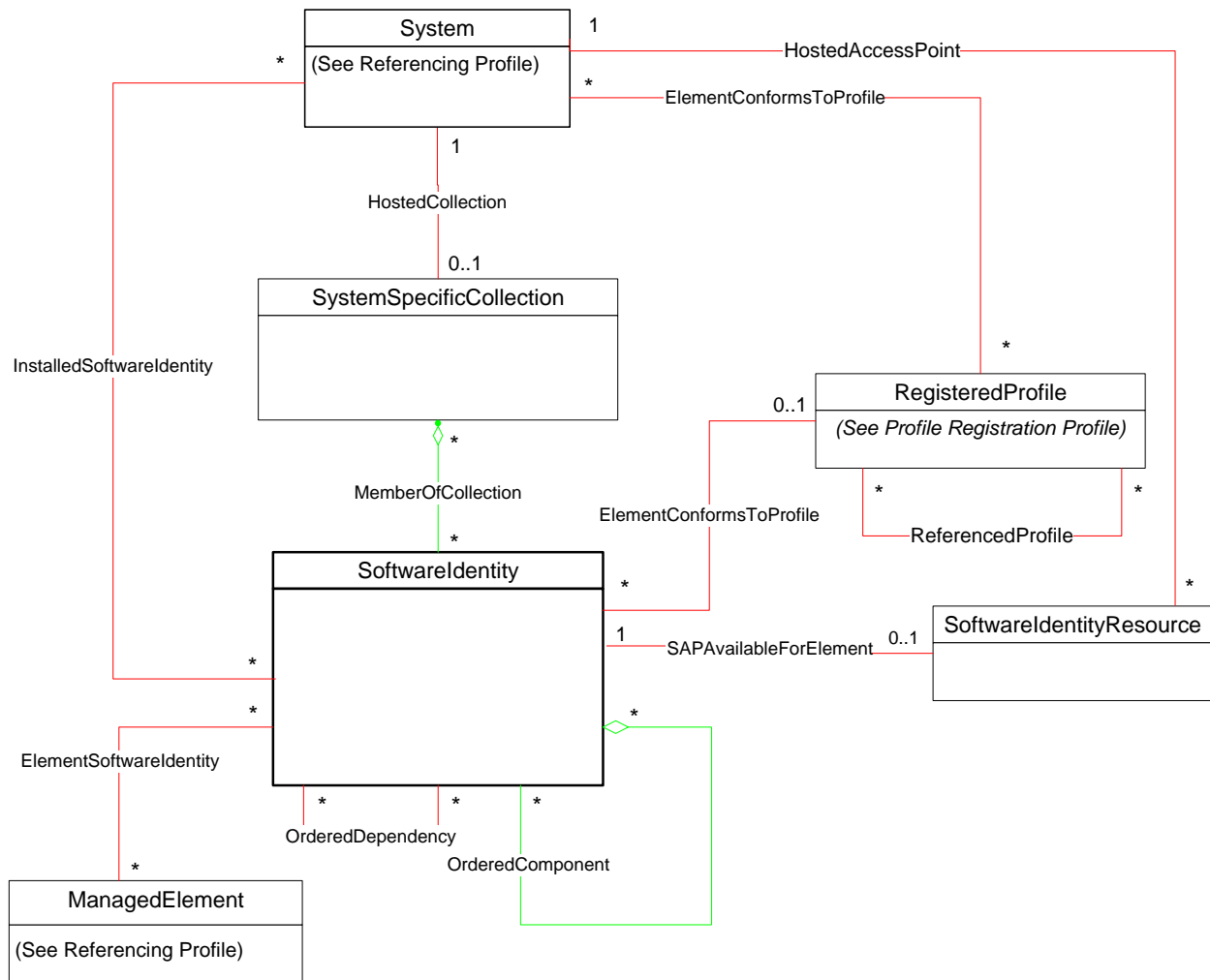
294

**Table 1 – Related Profiles**

Profile Name	Organization	Version	Requirement	Description
Profile Registration	DMTF	1.0	Mandatory	The profile that specifies registered profiles

## 295 **6 Description**

296 The *Software Inventory Profile* provides the ability to perform an inventory of installed BIOS, firmware,  
297 drivers, and related software such as providers and instrumentation software. This profile also describes  
298 the CIM schema elements required to represent the software that can be installed on a managed system.  
299 It also provides information about what software is associated with particular Managed Elements such as  
300 devices. Figure 1 represents the class schema of the *Software Inventory Profile* and shows the elements  
301 of the *Software Inventory Profile*, as well as the dependent relationships between the elements of  
302 *Software Inventory Profile* and the referencing profiles. For simplicity, the prefix *CIM\_* has been removed  
303 from the names of the classes.



304

305

**Figure 1 – Class Diagram: Software Inventory Profile**

306

The *Software Inventory Profile* can be used to represent the following software:

307

308

309

- the software that is installed on any Managed Element in the scope of the managed system (see section 7.2) so that the user of the profile can inventory the installed software for the managed system

310

311

312

- the software that is available for installation on any Managed Element in the scope of the managed system (see section 7.6) for providing the user of the profile the capability to view all the software that is available for any Managed Element within the scope of the managed system

313

314

For Available or Installed Software, the relationship between a Managed Element and the software that is compatible with the Managed Element (see section 7.4) can be modeled.

## 315 7 Implementation

316 This section describes the implementation requirements of the *Software Inventory Profile*. Required  
317 methods are described in section 8 (“Methods”), and properties are described in section 10 (“CIM  
318 Elements”).

### 319 7.1 Representing Software

320 The implementation shall model Installed Software (see section 7.2), Available Software (see  
321 section 7.6), or both, as a part of this profile.

### 322 7.2 Representing Installed Software

323 When an implementation models Installed Software, each Installed Software image modeled by the  
324 implementation shall be represented by exactly one instance of CIM\_SoftwareIdentity. The IsEntity  
325 property of the instance of CIM\_SoftwareIdentity shall have the value true.

#### 326 7.2.1 CIM\_InstalledSoftwareIdentity Instance

327 The Software Identity that represents an Installed Software shall be associated to the Scoping Instance  
328 using exactly one instance of CIM\_InstalledSoftwareIdentity.

### 329 7.3 Representing Version Information of Software

330 When the version information is not represented using the VersionString property, it shall be represented  
331 using the MajorVersion, MinorVersion, RevisionNumber, and BuildNumber properties. These properties  
332 are conditional and shall be implemented when the VersionString property is Null. When MinorVersion  
333 has a non-Null value, MajorVersion shall have a non-Null value. When RevisionNumber has a non-Null  
334 value, MinorVersion shall have a non-Null value. When BuildNumber has a non-Null value,  
335 RevisionNumber shall have a non-Null value. The algorithm for comparing versions of two instances of  
336 CIM\_SoftwareIdentity using these properties is described in section 7.10.

### 337 7.4 Representing Relationships between Software Identity and Managed 338 Element

339 The relationships between the software and the Managed Element may be modeled. This behavior is  
340 optional. When this behavior is implemented, the requirements specified in the following sections shall be  
341 met.

#### 342 7.4.1 CIM\_ElementSoftwareIdentity Instance

343 When a Managed Element is represented, the relationships between the Managed Element and the  
344 compatible Software Identity shall be represented using an instance of CIM\_ElementSoftwareIdentity.

345 When the Managed Element is not represented with an instance, the relationship between the compatible  
346 Software Identity and the Managed Element may be still represented by associating the Software Identity  
347 to the Scoping Instance through an instance of CIM\_ElementSoftwareIdentity.

##### 348 7.4.1.1 CIM\_ElementSoftwareIdentity.ElementSoftwareStatus

349 The CIM\_ElementSoftwareIdentity.ElementSoftwareStatus property shall represent the relationships of  
350 the software, represented by the Software Identity, to the Managed Element, through one or more  
351 enumeration values.

352 If the relationship between the Software Identity and the Managed Element is unknown, then the  
353 CIM\_ElementSoftwareIdentity.ElementSoftwareStatus property shall contain no enumeration values.

354 NOTE: The ElementSoftwareStatus property does not convey the current status of the Managed Element itself.

355 **7.4.1.1.1 CIM\_ElementSoftwareIdentity.ElementSoftwareStatus Enumeration Relationships**

356 The relationships between the ElementSoftwareStatus property enumeration values on a single instance  
 357 of CIM\_ElementSoftwareIdentity are described in Table 2. When the ElementSoftwareStatus property of  
 358 an instance of CIM\_ElementSoftwareIdentity has the value specified in the “Enumeration Value” column  
 359 of Table 2, the ElementSoftwareStatus property of the same instance shall also have other enumeration  
 360 values specified in the corresponding row of the “Mandatory Pairing With” column of Table 2.

361 When the ElementSoftwareStatus property of an instance of CIM\_ElementSoftwareIdentity has the value  
 362 specified in the “Enumeration Value” column of Table 2, the ElementSoftwareStatus property of the same  
 363 instance may also have other enumeration values specified in the corresponding row of the “May Be  
 364 Used With” column of Table 2.

365 When the ElementSoftwareStatus property of an instance of CIM\_ElementSoftwareIdentity has the value  
 366 specified in the “Enumeration Value” column of Table 2, the ElementSoftwareStatus property of the same  
 367 instance shall not have other enumeration values specified in the corresponding row of the “Shall Not Be  
 368 Used With” column of Table 2.

369 NOTE: The "May Be Used With," "Mandatory Pairing With," and "Shall Not Be Used With" columns express the  
 370 relationship of a contained value to the value in the "Enumeration Value" column. They do not express the  
 371 relationship between two values contained in the column itself. Therefore, the occurrence of two values together in  
 372 the "May Be Used With" column has no bearing on whether the two values may be used together.

373 **Table 2 – Relationships Between Enumeration Values of ElementSoftwareStatus**

Enumeration Value	Mandatory Pairing With	May Be Used With	Shall Not Be Used With
2 (Current)		3 (Next), 4 (FallBack), 5 (Default), 6 (Installed), 7 (SingleUse), 8 (Available)	9 (Supports)
3 (Next)	6 (Installed)	2 (Current), 4 (FallBack), 5 (Default)	7 (SingleUse), 8 (Available), 9 (Supports)
4 (FallBack)	6 (Installed)	2 (Current), 3 (Next), 5 (Default), 7 (SingleUse)	8 (Available), 9 (Supports)
5 (Default)		2 (Current), 3 (Next), 4 (FallBack), 6 (Installed), 7 (SingleUse), 8 (Available), 9 (Supports)	
6 (Installed)		2 (Current), 3 (Next), 4 (FallBack), 5 (Default), 7 (SingleUse)	8 (Available), 9 (Supports)
7 (SingleUse)	6 (Installed)	5 (Default), 2 (Current), 4 (FallBack)	3 (Next), 8 (Available), 9 (Supports)
8 (Available)		2 (Current), 5 (Default)	3 (Next), 4 (FallBack), 6 (Installed), 7 (SingleUse), 9 (Supports)
9 (Supports)		5 (Default)	2 (Current), 3 (Next), 4 (FallBack), 6 (Installed), 7 (SingleUse), 8 (Available)

374 When a Software Identity that is associated with a Managed Element through an instance of  
 375 CIM\_ElementSoftwareIdentity with the ElementSoftwareStatus property containing the value 3 (Next) or 7  
 376 (SingleUse) fails to run, the system shall automatically attempt to use the Software Identity that is  
 377 associated with the same Managed Element through an instance of CIM\_ElementSoftwareIdentity with  
 378 the ElementSoftwareStatus property containing the value 4 (FallBack), and no client action shall be  
 379 required.

#### 380 **7.4.2 ElementSoftwareIdentity for Software That Is Intended for a Managed Element But** 381 **Does Not Run or Get installed on It**

382 When an instance of CIM\_ElementSoftwareIdentity is used to represent the relationship between a  
383 Software Identity and a Managed Element such that the Software Identity will work with or can operate  
384 the Managed Element but is installed and runs on a different Managed Element, the only value that the  
385 ElementSoftwareIdentity.ElementSoftwareStatus property shall have is 9 (Supports).

### 386 **7.5 Finding the Scoping Instance of the CIM\_System Class**

387 The following algorithm shall be used for locating the Scoping Instance of the CIM\_System class from any  
388 instance of CIM\_SoftwareIdentity:

- 389 1) If the selected instance is referenced by an instance of CIM\_InstalledSoftwareIdentity, the  
390 Scoping Instance shall be the instance of CIM\_System that is associated through the instance  
391 of CIM\_InstalledSoftwareIdentity.
- 392 2) Otherwise, if the selected instance is referenced by the instance of CIM\_MemberOfCollection,  
393 select the instance of CIM\_SystemSpecificCollection that is associated through the instance of  
394 CIM\_MemberOfCollection. The Scoping Instance of the profile shall be the instance of  
395 CIM\_System that is associated with the selected instance of CIM\_SystemSpecificCollection  
396 through the instance of CIM\_HostedCollection.

### 397 **7.6 Representing Available Software**

398 When an implementation represents the Installed Software with Available Software, each Available  
399 Software image modeled by the implementation shall be represented by a Software Identity. The IsEntity  
400 property of the instance of CIM\_SoftwareIdentity shall have the value true. The following subsections are  
401 applicable when Available Software is represented.

#### 402 **7.6.1 CIM\_SystemSpecificCollection Instance**

403 An implementation shall instantiate a single instance of CIM\_SystemSpecificCollection, which is a  
404 collection of all the Available Software. The ElementName property of this instance of  
405 CIM\_SystemSpecificCollection shall have a value of "Available Software".

#### 406 **7.6.2 CIM\_HostedCollection Instance**

407 The instance of CIM\_SystemSpecificCollection shall be associated to the Scoping Instance by exactly  
408 one instance of CIM\_HostedCollection.

#### 409 **7.6.3 CIM\_MemberOfCollection Instance**

410 For each Software Identity that represents an Available Software, exactly one instance of  
411 CIM\_MemberOfCollection shall associate the Software Identity to the CIM\_SystemSpecificCollection  
412 instance.

#### 413 **7.6.4 Advertising the Location Information of a Software Identity**

414 The location of Available Software may be modeled. This behavior is optional. When this behavior is  
415 implemented, the requirements specified in the following sections shall be met.

##### 416 **7.6.4.1 CIM\_SoftwareIdentityResource Instance**

417 The location of a Software Identity shall be represented by an instance of CIM\_SoftwareIdentityResource.  
418 This could be used as an input to the software installation service.



#### 419 **7.6.4.2 CIM\_SAPAvailableForElement Instance**

420 An instance of CIM\_SAPAvailableForElement shall be used to associate a Software Identity with a  
421 CIM\_SoftwareIdentityResource instance that represents the location information of the Software Identity.

#### 422 **7.6.4.3 CIM\_HostedAccessPoint**

423 An instance of CIM\_HostedAccessPoint shall be used to associate a CIM\_SoftwareIdentityResource  
424 instance and the CIM\_System or CIM\_ComputerSystem instance that represents the Scoping Instance of  
425 the Available Software whose location information is advertised by the CIM\_SoftwareIdentityResource  
426 instance.

### 427 **7.6.5 Identifying Target Operating Systems**

428 The operating systems supported by a Software Identity may be modeled. This behavior is optional.  
429 When this behavior is implemented, the target operating systems of a Software Identity shall be  
430 represented by using one or all of the methods described in the following sections.

#### 431 **7.6.5.1 CIM\_SoftwareIdentity.TargetOSTypes[]**

432 The TargetOSTypes[] array property shall be used to list the operating systems that are supported by the  
433 Software Identity. An empty array shall indicate that the supported operating systems are unknown. A  
434 value of 66 (Not Applicable) shall indicate that the operating system is irrelevant when determining the  
435 compatibility of the Software Identity.

#### 436 **7.6.5.2 CIM\_SoftwareIdentity.TargetOperatingSystems[]**

437 This TargetOperatingSystems[] property shall be used to represent the operating systems supported by  
438 the Software Identity that are not listed in the TargetOSTypes[] property array values.

## 439 **7.7 Representing a Software Bundle**

440 A Software Bundle may be modeled. This behavior is optional. A Software Bundle shall be represented  
441 using a Software Identity. The Software Identity shall have a value of 13 (Software Bundle) in the  
442 Classifications[] property. Each software image in the Software Bundle shall be represented by a  
443 Software Identity that shall be associated to the Software Identity that represents the Software Bundle,  
444 using a single instance of CIM\_OrderedComponent.

#### 445 **7.7.1 CIM\_OrderedComponent.GroupComponent**

446 The instance of CIM\_SoftwareIdentity that represents the Software Bundle shall be the value of the  
447 GroupComponent property.

#### 448 **7.7.2 CIM\_OrderedComponent.PartComponent**

449 The instance of CIM\_SoftwareIdentity that represents the individual software image that is a part of the  
450 Software Bundle shall be the value of the PartComponent property.

#### 451 **7.7.3 CIM\_OrderedComponent.AssignedSequence**

452 The AssignedSequence property indicates the order in which the Software Identity referenced by the  
453 CIM\_OrderedComponent instance shall be installed during the installation of the bundle. The Software  
454 Identity with the lowest value of AssignedSequence on the associated CIM\_OrderedComponent instance  
455 shall be installed first and the highest shall be installed last. An AssignedSequence value of zero shall  
456 indicate no ordering requirement. Equivalent values of the AssignedSequence property shall indicate no  
457 ordering preference.

## 458 7.8 Identifying a Software Identity

459 This section describes the use of the IdentityInfoType[] and IdentityInfoValue[] array properties to identify  
460 a Software Identity.

### 461 7.8.1 General Use of IdentityInfoType and IdentityInfoValue Properties

462 The IdentityInfoValue[] array property contains values that provide additional information to identify a  
463 Software Identity. The corresponding element in the IdentityInfoType[] array property shall indicate the  
464 type of information stored in the IdentityInfoValue[] array.

### 465 7.8.2 Using IdentityInfoType and IdentityInfoValue to Model a Software Family

466 Software Family is an application-specific invariant identifier that is consistent among versions of a  
467 Software Identity. Software Family may be used to correlate instances of the same software across  
468 namespaces or management infrastructures, regardless of version.

469 A Software Identity may belong to multiple Software Families. Each Software Family of the Software  
470 Identity shall be represented as follows:

- 471 • The IdentityInfoType[] array property shall have the value of "CIM:SoftwareFamily".
- 472 • The corresponding element in the IdentifyingInfoValue[] array property shall be of the format  
473 "<OrgID> : <LocalID>". <OrgID> shall include a copyrighted, trademarked, or otherwise unique  
474 name that is owned by the business entity creating or defining the Software Identity and LocalID  
475 is a unique value that is consistent among different versions of the software. The algorithm used  
476 to guarantee uniqueness of the LocalID is implementation specific. Two possible algorithms are  
477 as follows:

- 478 1) Following is an example algorithm that may be used to generate the LocalID of a Software  
479 Identity for which the supported operating systems can be determined by the  
480 instrumentation:

481 <CIM\_SoftwareIdentity.Classifications[]>:<CIM\_SoftwareIdentity.TargetOSTypes[]>:  
482 < Information of the Hardware/ Hardware family supported by the Software Identity>

483 <CIM\_SoftwareIdentity.Classifications[]> is one of the numeric values contained in the  
484 Classifications property, and <CIM\_SoftwareIdentity.TargetOSTypes[]> is one of the  
485 values contained in the TargetOSTypes property of the instance of CIM\_SoftwareIdentity.

- 486 2) Following is an example algorithm that may be used to generate the LocalID of a Software  
487 Identity for which the supported operating systems cannot be determined by the  
488 instrumentation:

489 <CIM\_SoftwareIdentity.Classifications[]>:< Information of the Hardware/ Hardware family  
490 supported by the Software>

491 <CIM\_SoftwareIdentity.Classifications[]> is one of the numeric values contained in the  
492 Classifications property of the instance of CIM\_SoftwareIdentity.

#### 493 7.8.2.1 Determining Common Software Family Membership

494 Two instances of CIM\_SoftwareIdentity shall belong to the same Software Family when at least one of  
495 the Software Families modeled for the first CIM\_SoftwareIdentity instance matches at least one of the  
496 Software Families modeled for the second CIM\_SoftwareIdentity instance.

## 497 **7.9 Representing Installation Dependencies**

498 Software on which a Software Identity is dependent may be modeled. This behavior is optional. When  
499 information about the dependency is known but a copy of the software is not modeled, the dependency  
500 shall be modeled using an instance of CIM\_SoftwareIdentity and the IsEntity property shall have the  
501 value false. When information about the dependency is known and a copy of the software is modeled, the  
502 dependency shall be modeled using an instance of CIM\_SoftwareIdentity and the IsEntity property shall  
503 have the value true.

### 504 **7.9.1 CIM\_OrderedDependency**

505 When a Software Identity that is a member of the Available Software collection has installation  
506 dependencies on software that is represented by an instance of CIM\_SoftwareIdentity, the  
507 instrumentation shall instantiate an instance of the CIM\_OrderedDependency association between the  
508 Software Identity and each Installation Dependency, represented by an instance of CIM\_SoftwareIdentity,  
509 to arrange the Installation Dependencies in a hierarchical order.

#### 510 **7.9.1.1 CIM\_OrderedDependency.Antecedent**

511 The instance of CIM\_SoftwareIdentity that represents the Installation Dependency shall be the value of  
512 the Antecedent property.

#### 513 **7.9.1.2 CIM\_OrderedDependency.Dependent**

514 The instance of CIM\_SoftwareIdentity for which the Installation Dependencies are represented shall be  
515 the value of the Dependent property.

#### 516 **7.9.1.3 CIM\_OrderedDependency.AssignedSequence**

517 The AssignedSequence property indicates the order or sequence in which the Installation Dependencies  
518 shall be resolved during the installation of the Software Identity. The Installation Dependency with the  
519 lowest value of AssignedSequence on the associated CIM\_OrderedComponent instance shall be  
520 installed first and the highest shall be installed last. An AssignedSequence value of zero shall indicate no  
521 ordering requirement.

## 522 **7.10 Version Comparison Using the MajorVersion, MinorVersion, 523 RevisionNumber, and BuildNumber Properties**

524 The following algorithm shall be used to indicate that a CIM\_SoftwareIdentity instance has a higher  
525 version than the other instance of CIM\_SoftwareIdentity when two instances of CIM\_SoftwareIdentity are  
526 compared.

527 When comparing two properties in each of the following steps, if only one of the properties is Null, the  
528 instance that has a non-Null property shall be the instance with the higher version. When both properties  
529 are Null, the two instances shall be considered as having equal value.

530 1) If the MajorVersion properties of the two instances are equal, go to step 2.

531 Otherwise, the instance with the higher value of the MajorVersion property shall be the instance  
532 with the higher version.

533 2) If the MinorVersion properties of the two instances are equal, go to step 3.

534 Otherwise, the instance with the higher value of the MinorVersion property shall be the instance  
535 with the higher version.

536 3) If the RevisionNumber properties of the two instances are equal, go to step 4.

537 Otherwise, the instance with the higher value of the RevisionNumber property shall be the  
538 instance with the higher version.

539 4) If the BuildNumber properties of the two instances are equal, the two instances shall have equal  
540 value.

541 Otherwise, the instance with the higher value of the BuildNumber property shall be the instance  
542 with the higher version.

## 543 8 Methods

544 This section details the requirements for supporting intrinsic operations for the CIM elements defined by  
545 this profile. The *Software Inventory Profile* does not define any extrinsic methods.

### 546 8.1 Profile Conventions for Operations

547 For each profile class (including associations), the implementation requirements for operations, including  
548 those in the following default list, are specified in class-specific subclauses of this clause.

549 The default list of operations is as follows:

- 550 • GetInstance
- 551 • Associators
- 552 • AssociatorNames
- 553 • References
- 554 • ReferenceNames
- 555 • EnumerateInstances
- 556 • EnumerateInstanceNames

### 557 8.2 CIM\_SoftwareIdentity

558 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

559 NOTE: Related profiles may define additional requirements on operations for the profile class.

### 560 8.3 CIM\_InstalledSoftwareIdentity

561 Table 3 lists implementation requirements for operations. If implemented, these operations shall be  
562 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 3, all operations in  
563 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

564 NOTE: Related profiles may define additional requirements on operations for the profile class.

565 **Table 3 – Operations: CIM\_InstalledSoftwareIdentity**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

566 **8.4 CIM\_ElementSoftwareIdentity**

567 Table 4 lists implementation requirements for operations. If implemented, these operations shall be  
 568 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 4, all operations in  
 569 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

570 NOTE: Related profiles may define additional requirements on operations for the profile class.

571 **Table 4 – Operations: CIM\_ElementSoftwareIdentity**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.4.1.	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

572 **8.4.1 CIM\_ElementSoftwareIdentity – ModifyInstance**

573 The following rules shall dictate the behavior of the ModifyInstance operation:

- 574 • When the ModifyInstance operation is used to set the ElementSoftwareStatus property to  
 575 contain the value 3 (Next):
  - 576 1) Find all the other instances of CIM\_ElementSoftwareIdentity that
    - 577 a) reference the same instance of CIM\_ManagedElement as the target instance of  
 578 CIM\_ElementSoftwareIdentity and
    - 579 b) reference an instance of CIM\_SoftwareIdentity that belongs to the same Software  
 580 Family as the instance of CIM\_SoftwareIdentity that is referenced by the target  
 581 instance of CIM\_ElementSoftwareIdentity.
  - 582 2) For each of the CIM\_ElementSoftwareIdentity instances found, remove the value 3 (Next)  
 583 from the ElementSoftwareStatus property if present.
- 584 • The implementation shall not allow the ModifyInstance operation to add the value 2 (Current) to  
 585 and remove the value 2 (Current) from the ElementSoftwareStatus property.
- 586 • When the ModifyInstance operation is used to set the ElementSoftwareStatus property to  
 587 contain the value 4 (FallBack):
  - 588 1) Find all the other instances of CIM\_ElementSoftwareIdentity that
    - 589 a) reference the same instance of CIM\_ManagedElement as the target instance of  
 590 CIM\_ElementSoftwareIdentity and
    - 591 b) reference an instance of CIM\_SoftwareIdentity that belongs to the same Software  
 592 Family as the instance of CIM\_SoftwareIdentity that is referenced by the target  
 593 instance of CIM\_ElementSoftwareIdentity.
    - 594 c) For each of the CIM\_ElementSoftwareIdentity instances found, remove the value 4  
 595 (FallBack) from the ElementSoftwareStatus property if present.
  - 596 • The implementation shall not allow the ModifyInstance operation to add or remove the value 5  
 597 (Default) from the ElementSoftwareStatus property.
  - 598 • The implementation shall not allow the ModifyInstance operation to add or remove the value 6  
 599 (Installed) from the ElementSoftwareStatus property.

- 600       • When the ModifyInstance operation is used to set the ElementSoftwareStatus property to  
601       contain the value 7 (SingleUse):
- 602       1) Find all the other instances of CIM\_ElementSoftwareIdentity that
- 603           a) reference the same instance of CIM\_ManagedElement as the target instance of  
604           CIM\_ElementSoftwareIdentity and
- 605           b) reference an instance of CIM\_SoftwareIdentity that belongs to the same Software  
606           Family as the instance of CIM\_SoftwareIdentity that is referenced by the target  
607           instance of CIM\_ElementSoftwareIdentity.
- 608       2) For each of the CIM\_ElementSoftwareIdentity instances found, remove the value 7  
609       (SingleUse) from the ElementSoftwareStatus property if present.
- 610       • The implementation shall not allow the ModifyInstance operation to remove the value 8  
611       (Available) from the ElementSoftwareStatus property. The implementation shall allow adding  
612       the value 8 (Available) to the ElementSoftwareStatus property only if the associated Software  
613       Identity is associated with the CIM\_SystemSpecificCollection that has the ElementName  
614       property equal to "Available Software" through an instance of CIM\_MemberOfCollection.
- 615       • The implementation shall not allow the ModifyInstance operation to add or remove the value  
616       9 (Supports) from the ElementSoftwareStatus property.

## 617 8.5 CIM\_SystemSpecificCollection

618 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

619 NOTE: Related profiles may define additional requirements on operations for the profile class.

## 620 8.6 CIM\_HostedCollection

621 Table 5 lists implementation requirements for operations. If implemented, these operations shall be  
622 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 5, all operations in  
623 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

624 NOTE: Related profiles may define additional requirements on operations for the profile class.

625 **Table 5 – Operations: CIM\_HostedCollection**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

## 626 8.7 CIM\_MemberOfCollection

627 Table 6 lists implementation requirements for operations. If implemented, these operations shall be  
628 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 6, all operations in  
629 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

630 NOTE: Related profiles may define additional requirements on operations for the profile class.

631

**Table 6 – Operations: CIM\_MemberOfCollection**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

**632 8.8 CIM\_SoftwareIdentityResource**

633 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

634 NOTE: Related profiles may define additional requirements on operations for the profile class.

**635 8.9 CIM\_SAPAvailableForElement**

636 Table 7 lists implementation requirements for operations. If implemented, these operations shall be  
 637 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 7, all operations in  
 638 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

639 NOTE: Related profiles may define additional requirements on operations for the profile class.

640

**Table 7 – Operations: CIM\_SAPAvailableForElement**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

**641 8.10 CIM\_HostedAccessPoint**

642 Table 8 lists implementation requirements for operations. If implemented, these operations shall be  
 643 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 8, all operations in  
 644 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

645 NOTE: Related profiles may define additional requirements on operations for the profile class.

646

**Table 8 – Operations: CIM\_HostedAccessPoint**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

## 647 8.11 CIM\_OrderedComponent

648 Table 9 lists implementation requirements for operations. If implemented, these operations shall be  
 649 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 9, all operations in  
 650 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

651 NOTE: Related profiles may define additional requirements on operations for the profile class.

652 **Table 9 – Operations: CIM\_OrderedComponent**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

## 653 8.12 CIM\_OrderedDependency

654 Table 10 lists implementation requirements for operations. If implemented, these operations shall be  
 655 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 10, all operations  
 656 in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

657 NOTE: Related profiles may define additional requirements on operations for the profile class.

658 **Table 10 – Operations: CIM\_OrderedDependency**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

## 659 9 Use Cases

660 This section contains object diagrams and use cases for the *Software Inventory Profile*.

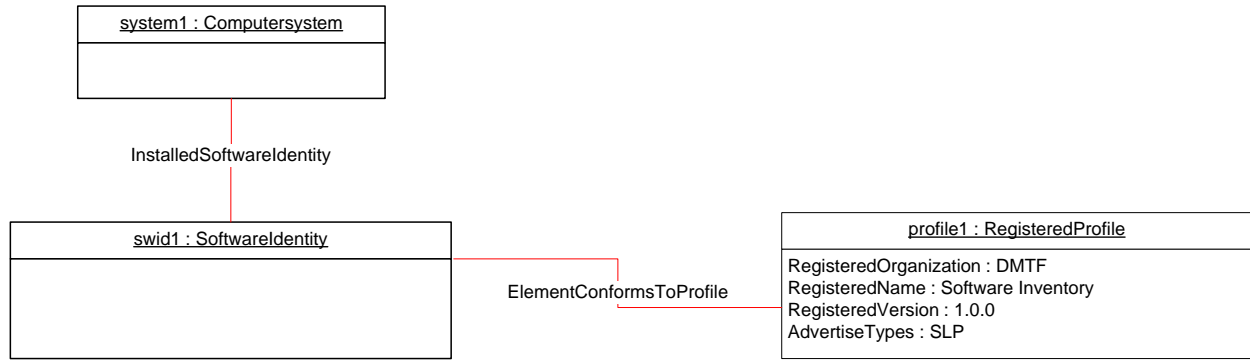
### 661 9.1 Object Diagrams

662 This section contains object diagrams for the *Software Inventory Profile*. For simplicity, the prefix *CIM\_*  
 663 has been removed from the names of the classes in the diagrams.



664 **9.1.1 Registered Profile**

665 Figure 2 represents a possible instantiation of the *Software Inventory Profile*. In this instantiation, the  
 666 Central Instance, swid1, has an InstalledSoftwareIdentity association to the Scoping Instance, system1.  
 667 Profile registration information is represented with the profile1 instance. Following the  
 668 CIM\_ElementConformsToProfile association from the Central Instance to profile1, the client can retrieve  
 669 information such as the version of the current *Software Inventory Profile* implementation.



670

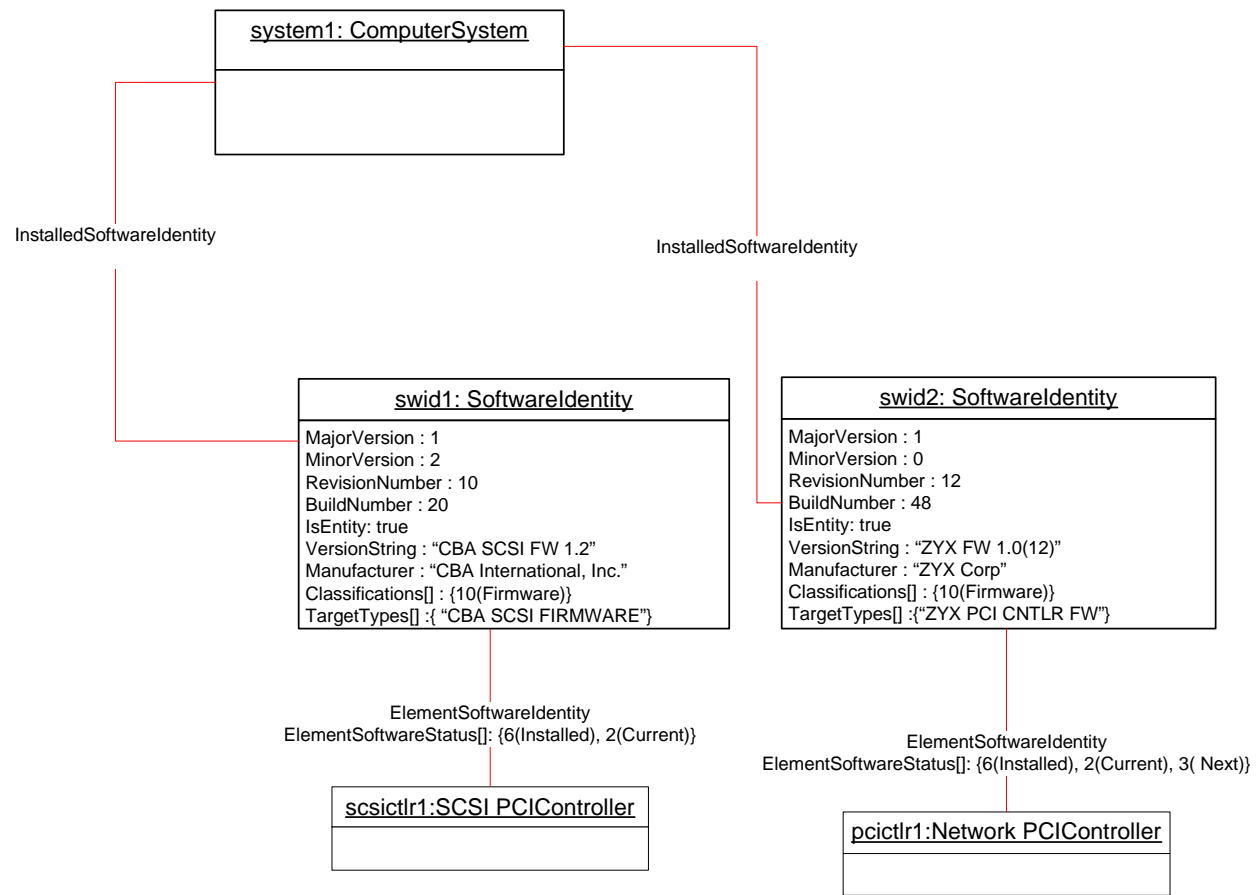
671

**Figure 2 – Registered Profile**

672 **9.1.2 Representing Installed Firmware**

673 Figure 3 represents a possible instantiation of the *Software Inventory Profile*. In the object diagram,  
 674 Software Identity swid1 is shown as installed on the SCSI PCI Controller, scsictrl1, and currently running  
 675 on it. The ElementSoftwareStatus property on the ElementSoftwareIdentity association instance between  
 676 swid1 and scsictrl1 does not have the value 3 (Next) because it is not the firmware that will run after the  
 677 next reboot of the system.

678 Software Identity swid2 is shown as installed on the Network PCI Controller, pcictrl1, and currently  
 679 running on it. swid2 would also run on the next reset or reboot of pcictrl1. The object diagram does not  
 680 show the CIM\_SystemDevice association between system1 and scsictrl1, and system1 and pcictrl1, but  
 681 both scsictrl1 and pcictrl1 are scoped to system1 and so the CIM\_InstalledSoftwareIdentity association  
 682 is shown between system1 and swid1, and system1 and swid2.



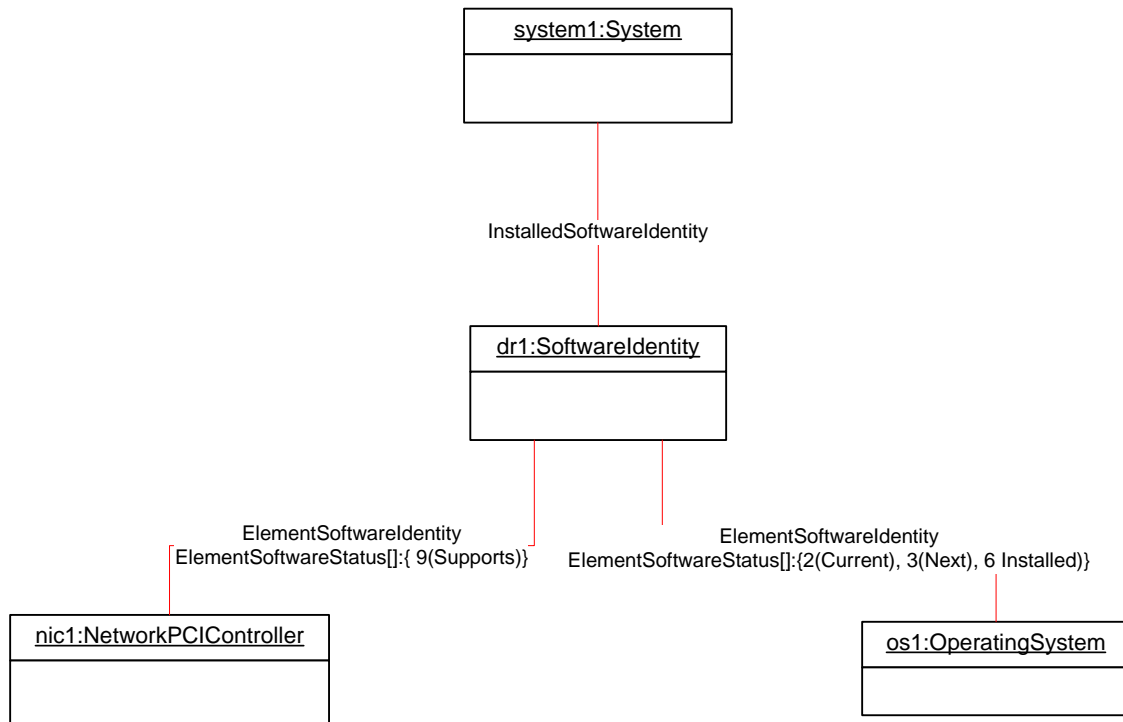
683

684

**Figure 3 – Object Diagram Showing Installed Software**

685 **9.1.3 Representing an Installed Driver**

686 Figure 4 represents a possible instantiation of the *Software Inventory Profile*. It shows how to model an  
 687 installed driver. In this instantiation, the driver, dr1, is applicable to the NIC, nic1. The  
 688 ElementSoftwareStatus value "Supports" indicates that dr1 is applicable to nic1. The driver is installed in  
 689 the OS, os1, and is the driver for nic1 that is currently running in os1.



690

691

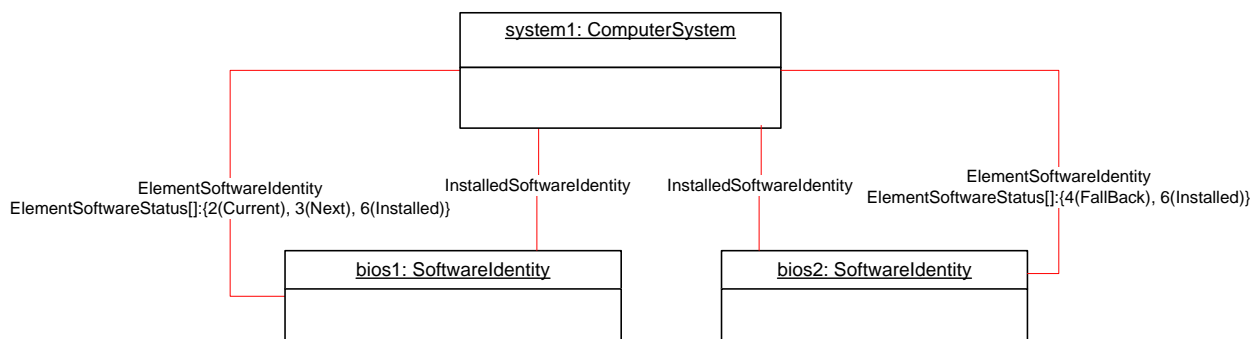
**Figure 4 – Object Diagram Showing an Installed Driver**

692 **9.1.4 Representing BIOS Installed on a System**

693 Figure 5 represents a possible instantiation of the *Software Inventory Profile*. Both bios1 and bios2 are  
 694 associated with system1 through an instance of InstalledSoftwareIdentity because both of them are  
 695 installed on a component of the system, which happens to be the system itself.

696 bios1 is for the system, system1, and so the CIM\_ElementSoftwareIdentity association is used to  
 697 associate them with the ElementSoftwareStatus property having the values 2 (Current), 3 (Next), and 6  
 698 (Installed).

699 bios2 is the backup for bios1 and is also for system, system1, and so the CIM\_ElementSoftwareIdentity  
 700 association is used to associate them with the ElementSoftwareStatus property having the values  
 701 4 (FallBack) and 6 (Installed).



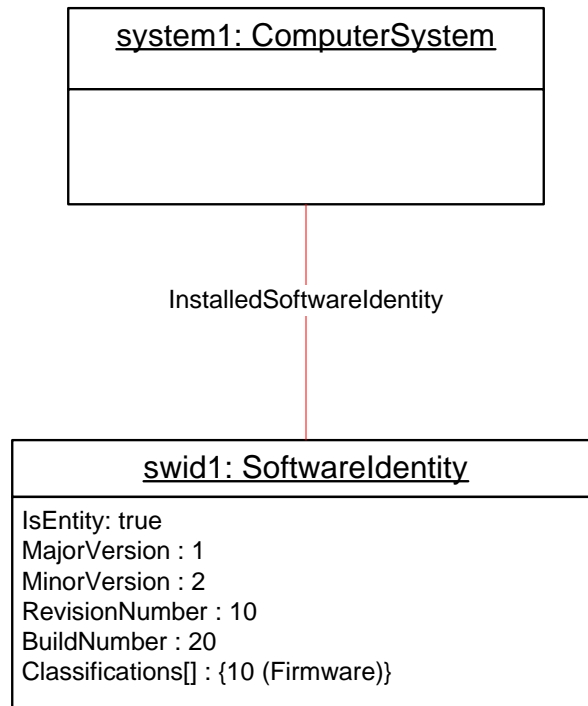
702

703

**Figure 5 – Object Diagram Showing Installed BIOS**

### 704 9.1.5 Representing Installed Software without Any Association to the Managed Element

705 Figure 6 represents a possible instantiation of the *Software Inventory Profile*. The firmware represented  
 706 by swid1 is installed on some Managed Element in the scope of system1 but the Managed Element is not  
 707 modeled by the instrumentation and since the CIM\_ElementSoftwareIdentity association is not  
 708 instantiated between system1 and swid1, the relationship the Managed Element and swid1 is not known.



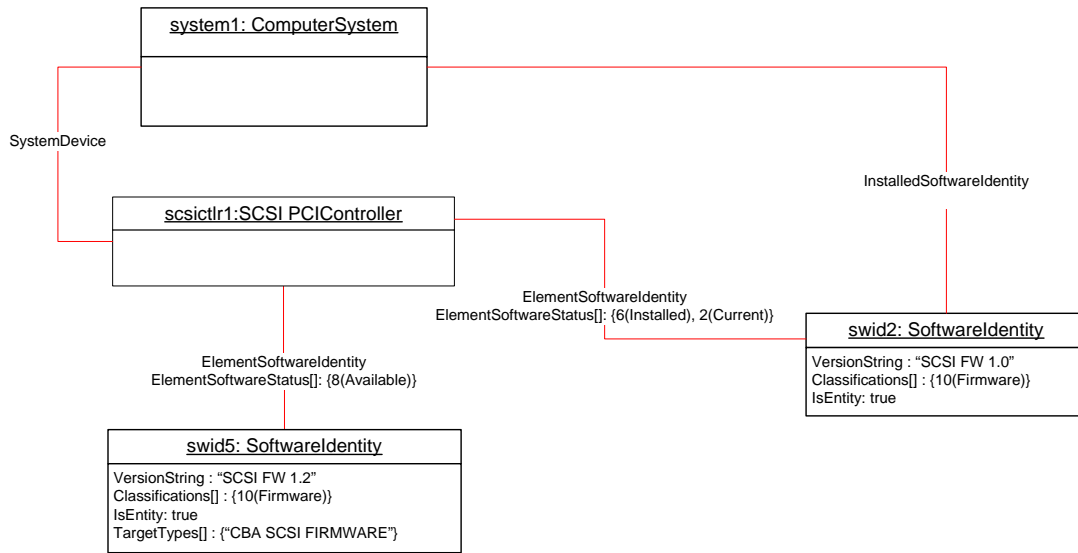
709

710 **Figure 6 – Object Diagram Showing Installed Software**

### 711 9.1.6 Representing More Than One Executable Software Identity on a Managed Element

712 Figure 7 represents a possible instantiation of the *Software Inventory Profile*. In the object diagram,  
 713 Software Identity swid2 is shown as installed on the SCSI PCI Controller, scsictrl1, and is currently  
 714 running on it. The ElementSoftwareStatus property on the CIM\_ElementSoftwareIdentity instance that  
 715 associates swid2 and scsictrl1 has the values 2 (Current) and 6 (Installed).

716 Software Identity swid3 is the manufacturer shipped version and is installed on scsictrl1 but is not  
 717 currently running. The CIM\_ElementSoftwareIdentity instance that associates swid3 and scsictrl1 conveys  
 718 this relationship by the ElementSoftwareStatus property having the values 5 (Default) and 6 (Installed).



719

720

**Figure 7 – Object Diagram Showing Multiple Installed Software on a Managed Element**

721

**9.1.7 Representing Available and Installed Firmware without Managed Element**

722

Figure 8 represents a possible instantiation of the *Software Inventory Profile*. The object diagram is an alternative instantiation of Figure 7 where the SCSI PCI Controller, scsictlr1, is not instantiated. Thus swid2 and swid5 are associated through the CIM\_ElementSoftwareIdentity associations to the Scoping Instance, system1.

723

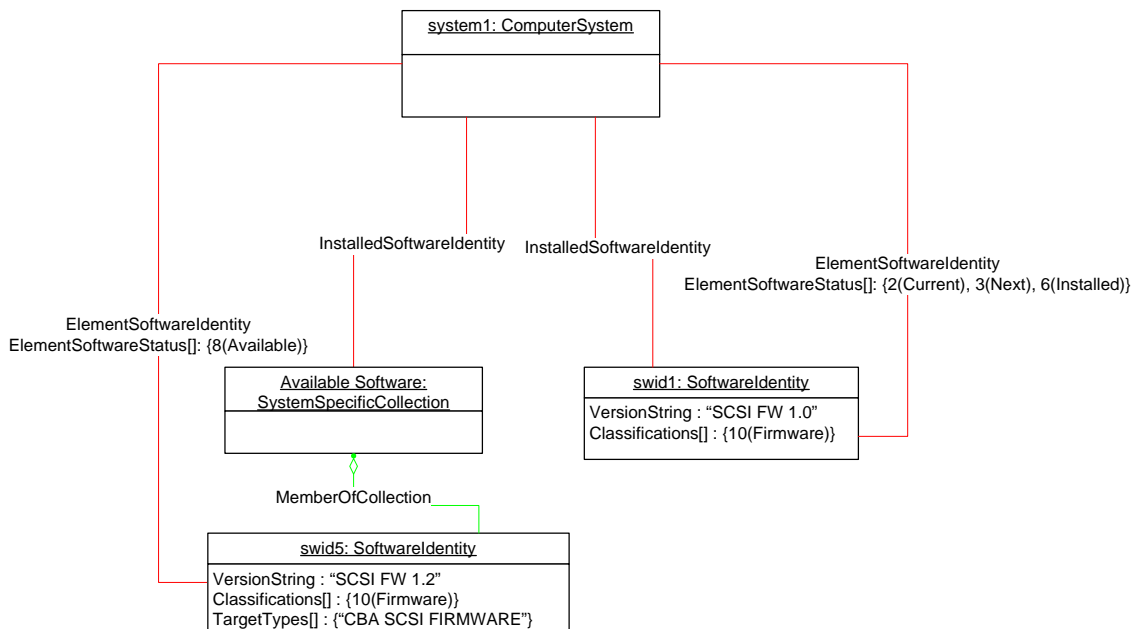
724

725

726

The ElementSoftwareStatus property on these associations still represents the relationship between the SCSI PCI Controller and swid2 and swid5.

727



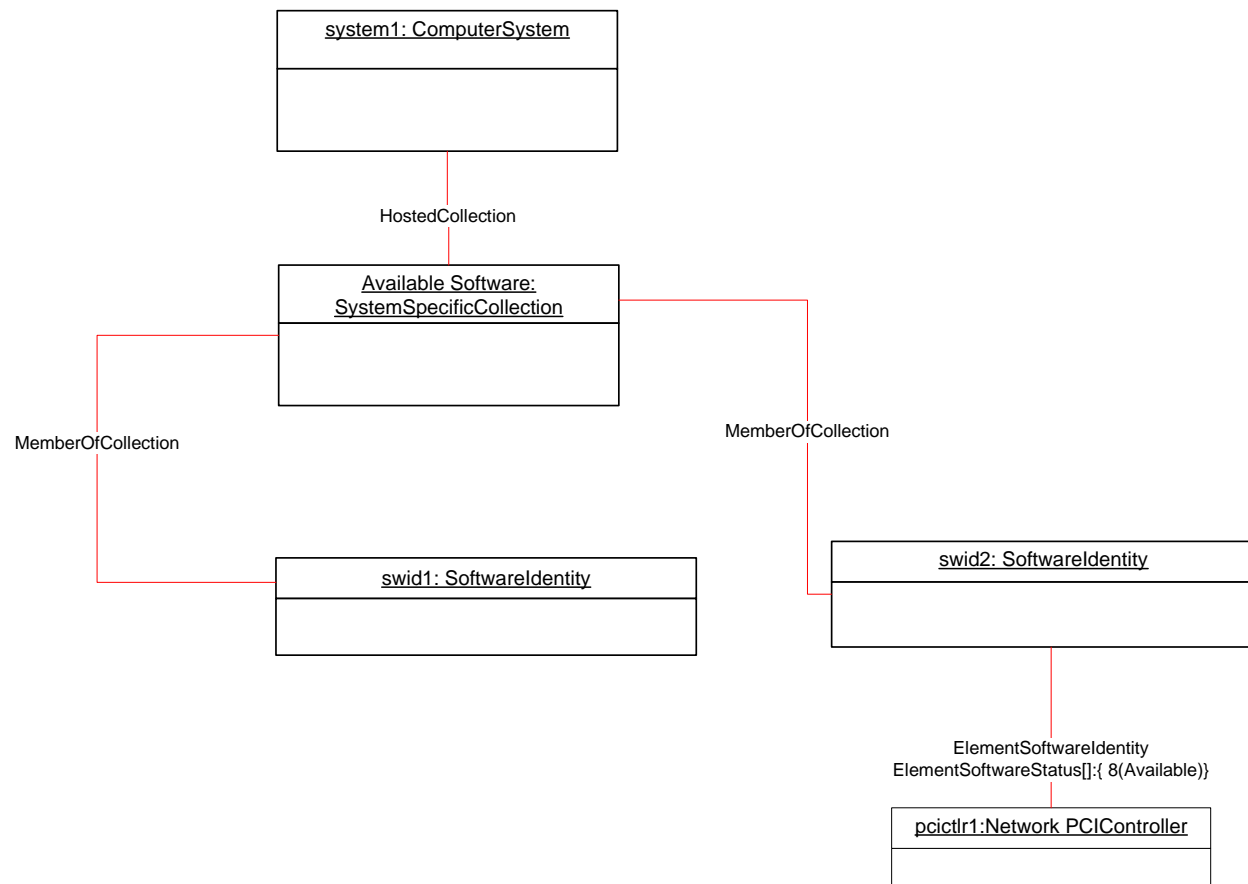
728

729

**Figure 8 – Object Diagram with No Instantiation of Managed Element**

### 730 9.1.8 Representing Available Firmware

731 Figure 9 represents a possible instantiation of the *Software Inventory Profile*. In the object diagram,  
 732 Software Identity swid2 is shown as available for installation on the Network PCI Controller, pcictrl1,  
 733 using the CIM\_ElementSoftwareIdentity association. Software Identity swid1 is an Available Software but  
 734 the compatible Managed Element is not modeled and no CIM\_ElementSoftwareIdentity instance  
 735 references swid1. pcictrl1 is scoped to system1, but the object diagram does not show the  
 736 CIM\_SystemDevice association between system1 and pcictrl1, and so the CIM\_MemberOfCollection  
 737 association is shown between an “Available Software” collection and swid2.



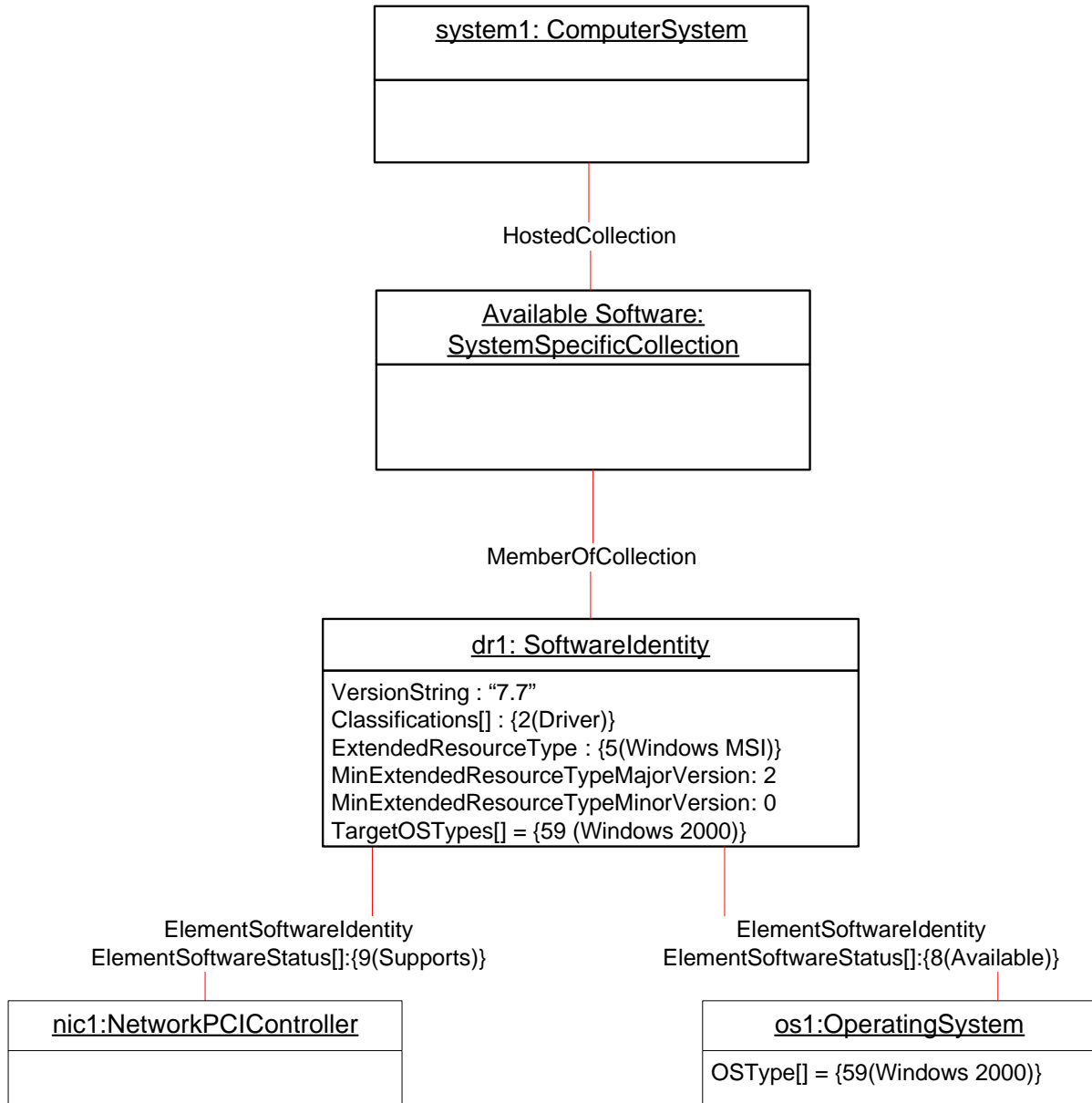
738

739 **Figure 9 – Object Diagram Showing Available Firmware**

### 740 9.1.9 Representing an Available Driver and Its Relationship to the Operating System

741 Figure 10 represents a possible instantiation of the *Software Inventory Profile*: an available driver. In this  
 742 instantiation, the driver, dr1, is applicable to the NIC, nic1. The ElementSoftwareStatus property of the  
 743 CIM\_ElementSoftwareIdentity association instance between dr1 and nic1 has the value 9 (Supports),  
 744 indicating that dr1 is applicable to nic1. The object diagram also represents the driver’s relationship to the  
 745 operating system, os1, with the ElementSoftwareIdentity association instance having the  
 746 ElementSoftwareStatus property with the value 8 (Available), indicating that dr1 is applicable to os1 and  
 747 is available for installation. The relationship between system1 and os1 is not shown.

748



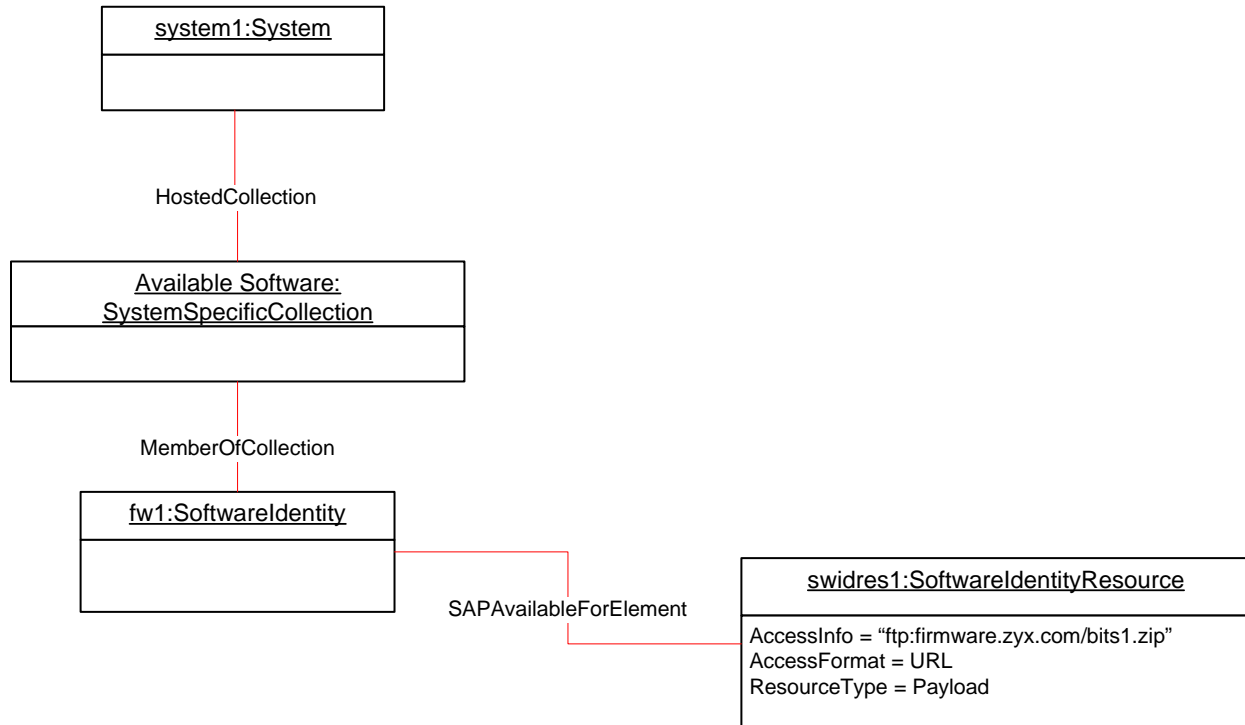
749

750

Figure 10 – Object Diagram Showing an Available Driver

### 751 9.1.10 Representing Available Software and Its Location Information

752 Figure 11 represents a possible instantiation of the *Software Inventory Profile*: an Available Software and  
 753 its location information. In this instantiation, the firmware, fw1, is available to the system and its location  
 754 information is modeled by swidres1.



755

756 **Figure 11 – Object Diagram Showing a Firmware Image and Its Location**

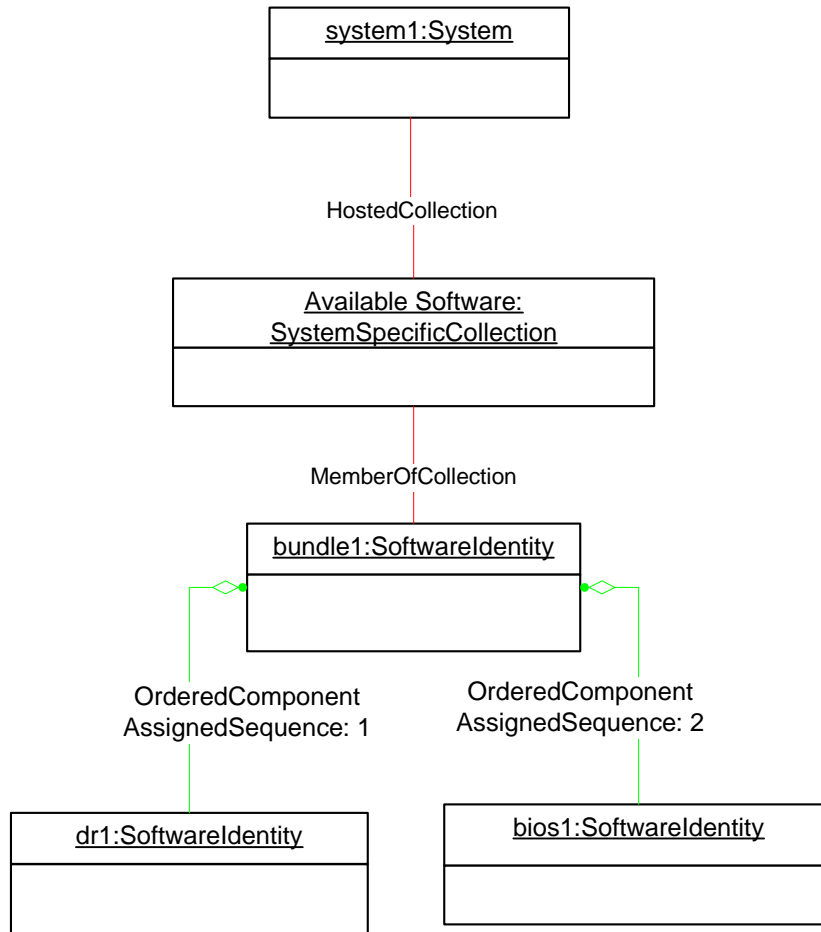
### 757 9.1.11 Representing a Software Bundle

758 Figure 12 represents a possible instantiation of the *Software Inventory Profile*: a Software Bundle. In the  
 759 diagram, the Software Bundle, bundle1, consists of two Software Identities:

- 760
- 761 • dr1 with the Assigned sequence of 1, indicating that dr1 will be the first to be installed while installing the bundle
  - 762 • bios1 with the Assigned sequence of 2, indicating that bios1 will be the second to be installed while installing the bundle
- 763



764 After bundle1 has been installed, instrumentation will create associations relating to dr1 as shown in  
765 Figure 4 and associations relating to bios1 as shown in Figure 5.



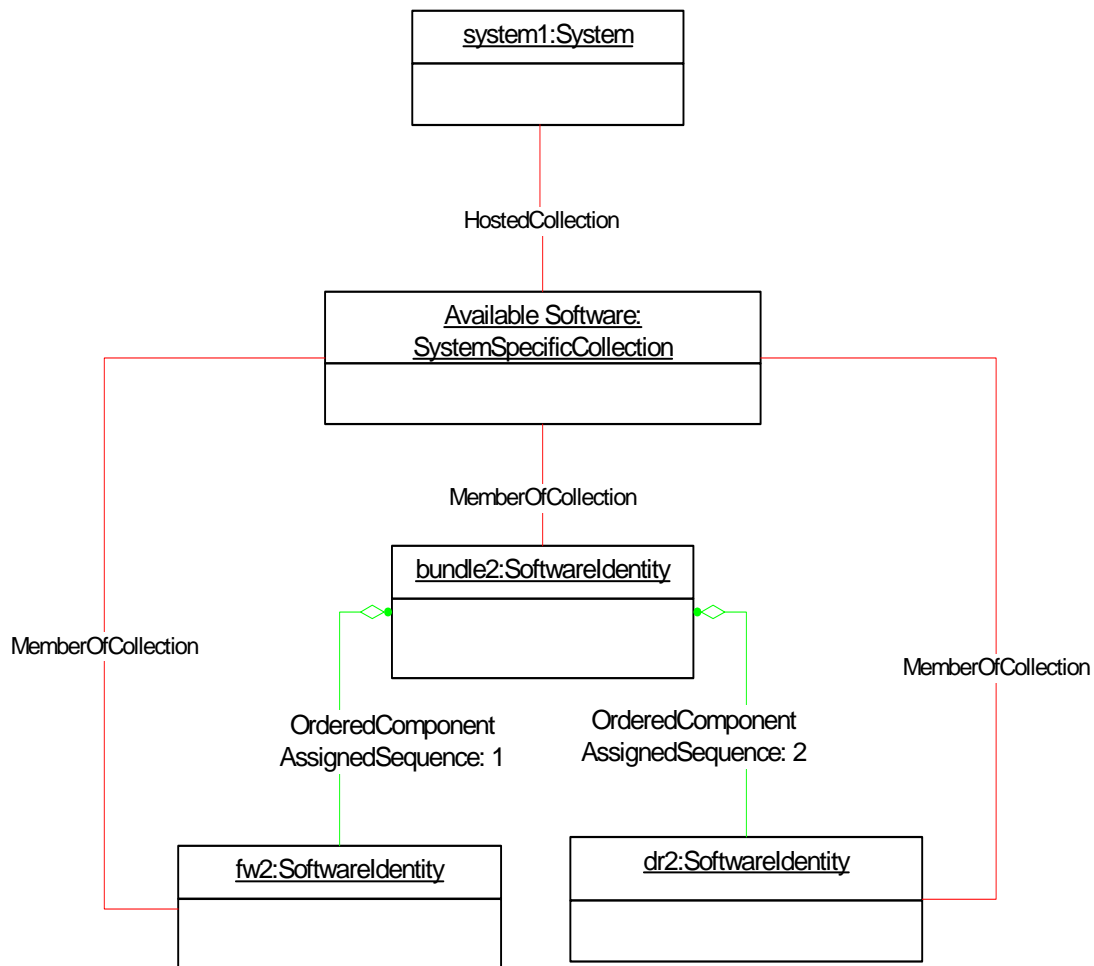
766

767

**Figure 12 – Object Diagram Showing a Software Bundle**

### 768 9.1.12 Representing Software That Is Part of a Software Bundle and Available

769 Figure 13 represents a possible instantiation of the *Software Inventory Profile*. In the diagram, the  
 770 Software Bundle, bundle2, consists of two Software Identities, fw2 and dr2, both of which are members of  
 771 the “Available Software” collection. So, fw2 and dr2 could be installed either individually or as a part of  
 772 installing bundle2.



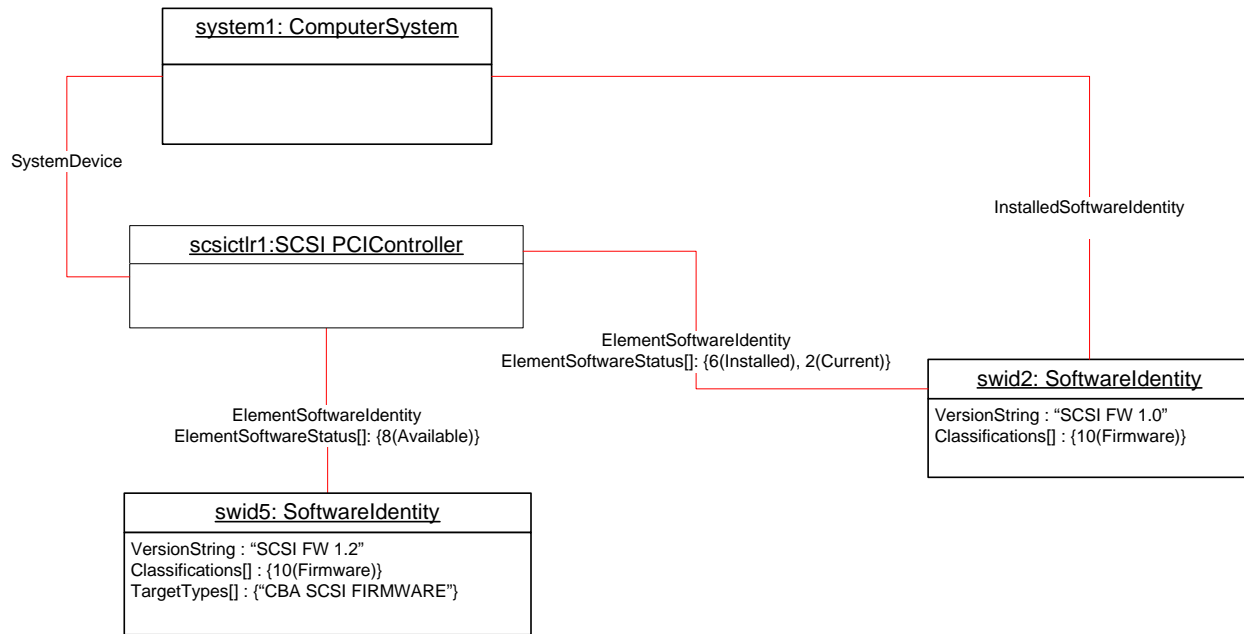
773

774 **Figure 13 – Object Diagram Showing Available Software That Is Part of a Software Bundle**

### 775 9.1.13 Representing Installed and Available Software

776 Figure 14 represents a possible instantiation of the *Software Inventory Profile*. In the object diagram,  
 777 Software Identity swid2 is shown as installed on the SCSI PCI Controller, scsictrl1, and is currently  
 778 running on it. The ElementSoftwareStatus property on the CIM\_ElementSoftwareIdentity instance  
 779 associating swid2 and scsictrl1 has the values 2 (Current) and 6 (Installed).

780 Software Identity swid5 is shown as Available Software for scsictrl1, and so the ElementSoftwareStatus  
 781 property on the CIM\_ElementSoftwareIdentity instance that associates swid5 and scsictrl1 has the value  
 782 8 (Available).



783

784

Figure 14 – Object Diagram Showing Installed and Available Software

785 **9.2 Find All the Software Installed on All the Managed Elements within the**  
 786 **Scope of a Managed System**

787 For the instance of CIM\_System that represents the given managed system, select all the instances of  
 788 CIM\_SoftwareIdentity that are associated through instances of CIM\_InstalledSoftwareIdentity. The  
 789 resulting instances represent the software installed on all the Managed Elements in the scope of the  
 790 managed system.

791 **9.3 Find All the Software Installed on a Managed Element**

792 For the given instance of CIM\_ManagedElement, select the instance of CIM\_SoftwareIdentity that is  
 793 associated through an instance of CIM\_ElementSoftwareIdentity such that the ElementSoftwareStatus  
 794 property contains the value 6 (Installed).

795 **9.4 Find All the Software That Is Compatible with a Managed Element but Has**  
 796 **Not Been Installed**

797 For the given instance of CIM\_ManagedElement, using the CIM\_ElementSoftwareIdentity association,  
 798 select the associated instances of CIM\_SoftwareIdentity that are not associated to the scoping  
 799 CIM\_System or CIM\_ComputerSystem instance through the CIM\_InstalledSoftwareIdentity association. If  
 800 the given instance of CIM\_ManagedElement does not have any associating  
 801 CIM\_ElementSoftwareIdentity instances referencing it, the compatible software will not be determinable  
 802 and no instances of CIM\_SoftwareIdentity will be returned.

803 **9.5 Find All the Software That Is Available for Installation on Any Managed**  
 804 **Element within the Scope of a Managed System**

805 For the instance of CIM\_ComputerSystem that represents the given managed system, select the instance  
 806 of CIM\_SystemSpecificCollection with ElementName value of "Available Software" that is associated  
 807 through and instance of CIM\_HostedCollection. Select all the instances of CIM\_SoftwareIdentity that are  
 808 associated through an instance of CIM\_MemberOfCollection.

## 809 **9.6 For a Given NIC, Find the Driver That Is Running in the Operating System**

810 The client can find the driver that is currently running by using the following steps:

- 811 1) For the instance of CIM\_ManagedElement that represents the NIC, select the instances of the  
812 CIM\_ElementSoftwareIdentity association with the ElementSoftwareStatus property containing  
813 the value 9 (Supports).
- 814 2) Select the instances of CIM\_SoftwareIdentity that the instances of  
815 CIM\_ElementSoftwareIdentity from step 1 reference.
- 816 3) From the given instance of CIM\_ManagedElement that represents the NIC, select the instance  
817 of CIM\_ComputerSystem that is associated through an instance of CIM\_SystemDevice.
- 818 4) From the CIM\_ComputerSystem instance from step 3, select the instance of  
819 CIM\_OperatingSystem that is associated through an instance of CIM\_RunningOS.
- 820 5) Select the instances of CIM\_ElementSoftwareIdentity that reference the instance of  
821 CIM\_OperatingSystem from step 4 and contain the value 2 (Current) in the  
822 ElementSoftwareStatus property.
- 823 6) Select the instance of CIM\_SoftwareIdentity that is referenced by at least one instance of  
824 CIM\_ElementSoftwareIdentity from step 2 and at least one instance of  
825 CIM\_ElementSoftwareIdentity from step 5.

## 826 **9.7 Set a Particular Software Image on a Hardware Managed Element to Run** 827 **After the Next Reset or Reboot**

828 The client can set a particular software image on a hardware managed element to run after the next reset  
829 or reboot by using the following steps:

- 830 1) Select the CIM\_ElementSoftwareIdentity association instance that associates the Managed  
831 Element instance that represents the device with the Software Identity instance that represents  
832 the software image.
- 833 2) Set the value of the ElementSoftwareStatus property on the ElementSoftwareIdentity  
834 association to 3 (Next).

## 835 **9.8 Set a Particular Software Image on a Hardware Managed Element to Run** 836 **After the Next Reset or Reboot but Not After a Subsequent Reset or Reboot**

837 The client can set a particular software image on a hardware managed element to run after the next reset  
838 or reboot but not after a subsequent reset or reboot by using the following steps:

- 839 1) Select the CIM\_ElementSoftwareIdentity association instance that associates the Managed  
840 Element instance that represents the device with the Software Identity instance that represents  
841 the software image.
- 842 2) Set the value of the ElementSoftwareStatus property on the ElementSoftwareIdentity  
843 association to 7 (SingleUse).

## 844 **9.9 Find and Set a Driver to Run After the Next Reset or Reboot for a NIC**

845 A client can set a driver to run on the next reset or reboot by using the following steps:

- 846 1) For the instance of CIM\_ManagedElement that represents the NIC, select the instances of  
847 CIM\_ElementSoftwareIdentity association with the ElementSoftwareStatus property containing  
848 the value 9 (Supports).
- 849 2) Select the instance of CIM\_SoftwareIdentity that the instances of CIM\_ElementSoftwareIdentity  
850 from step 1 reference.
- 851 3) Identify the CIM\_SoftwareIdentity instance that corresponds to the driver.

- 852 4) From the given instance of CIM\_ManagedElement that represents the NIC, select the instance  
853 of CIM\_ComputerSystem that is associated through an instance of CIM\_SystemDevice.
- 854 5) From the CIM\_ComputerSystem instance from step 4, select the instance of  
855 CIM\_OperatingSystem that is associated through an instance of CIM\_RunningOS.
- 856 6) Select the instances of CIM\_ElementSoftwareIdentity that reference the instance of  
857 CIM\_OperatingSystem from step 5 and contain the value 6 (Installed) in the  
858 ElementSoftwareStatus property.
- 859 7) Select the instance of CIM\_ElementSoftwareIdentity that associates the instance of  
860 CIM\_ManagedElement and the instance of CIM\_SoftwareIdentity from step 3. Set the value of  
861 the ElementSoftwareStatus property of this instance to 3 (Next).

## 862 **9.10 Find the Most Recent Firmware Available for a NIC**

863 A client can find the most recent firmware available for a NIC by using the following steps:

- 864 1) For the given instance of CIM\_ManagedElement that represents the NIC, select the instances  
865 of CIM\_SoftwareIdentity that are associated through instances of CIM\_ElementSoftwareIdentity  
866 with the ElementSoftwareStatus property containing the value 8 (Available) with the  
867 Classifications[] property on the CIM\_SoftwareIdentity instance containing the value 10  
868 (Firmware).
- 869 2) From the instances returned, select the instance of CIM\_SoftwareIdentity with the highest  
870 version. (See section 7.10 for the version comparison algorithm.)

## 871 **9.11 Find the Most Recent Firmware Installed on a NIC**

872 A client can find the most recent firmware installed on a NIC by using the following steps:

- 873 1) For the given instance of CIM\_ManagedElement that represents the NIC, select the instances  
874 of CIM\_SoftwareIdentity that are associated through instances of CIM\_ElementSoftwareIdentity  
875 with the ElementSoftwareStatus property containing the value 6 (Installed) with the  
876 Classifications[] property on the CIM\_SoftwareIdentity instance containing the value 10  
877 (Firmware).
- 878 2) From the instances returned, select the instance of CIM\_SoftwareIdentity with the highest  
879 version. (See section 7.10 for the version comparison algorithm.)

## 880 **9.12 Find the Software Families of Which a Software Identity Is a Member**

881 For the given instance of CIM\_SoftwareIdentity, select all the values in the IdentityInfoValue[] property  
882 array that have a value at the corresponding index in the IdentityInfoType[] property array equal to  
883 "CIM:SoftwareFamily". Each of the selected values represents a Software Family of which the Software  
884 Identity is a member.

## 885 **9.13 Determine Whether a Dependency of a Software Identity Is Satisfied**

886 Given an instance of CIM\_SoftwareIdentity that represents an Installation Dependency for a Software  
887 Identity, a client can determine if the dependency is resolved as follows:

- 888 1) From the Scoping Instance, select all the instances of CIM\_SoftwareIdentity that are associated  
889 through instances of CIM\_InstalledSoftwareIdentity.
- 890 2) For each Software Identity from step 1, determine all the Software Families to which it belongs  
891 by using the algorithm in section 9.12.
- 892 3) For the instance of CIM\_SoftwareIdentity that represents the dependency, determine the  
893 Software Families by using the algorithm in section 9.12.

- 894 4) Select the instance of CIM\_SoftwareIdentity from step 1 such that at least one Software Family  
 895 to which it belongs (from step 2) is equal to at least one Software Family to which the  
 896 dependency belongs (from step 3).

897 The dependency is satisfied if the version of the selected Software Identity is greater than or equal to the  
 898 version of the dependency represented by an instance of CIM\_SoftwareIdentity. (See section 7.10 for the  
 899 version comparison algorithm.)

## 900 10 CIM Elements

901 Table 11 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be  
 902 implemented as described in Table 11. Sections 7 (“Implementation”) and 8 (“Methods”) may impose  
 903 additional requirements on these elements.

904 **Table 11 – CIM Elements: Software Inventory Profile**

Element Name	Requirement	Description
<b>Classes</b>		
CIM_SoftwareIdentity	Mandatory	See sections 7.2, 7.6, and 10.1.
CIM_InstalledSoftwareIdentity	Conditional	See sections 7.2.1 and 10.2.
CIM_ElementSoftwareIdentity	Optional	See sections 7.4 and 10.3.
CIM_SystemSpecificCollection	Optional	See sections 7.6.1 and 10.4.
CIM_HostedCollection	Conditional	See sections 7.6.2 and 10.5.
CIM_MemberOfCollection	Conditional	See sections 7.6.3 and 10.6.
CIM_SoftwareIdentityResource	Optional	See sections 7.6.4.1 and 10.7.
CIM_SAPAvailableForElement	Conditional	See sections 7.6.4.2 and 10.8.
CIM_HostedAccessPoint	Optional	See sections 7.6.4.3 and 10.9.
CIM_OrderedComponent	Optional	See sections 7.7 and 10.10.
CIM_OrderedDependency	Optional	See sections 7.9.1 and 10.11.
CIM_RegisteredProfile	Mandatory	See section 10.12.
<b>Indications</b>		
None defined in this profile		

### 905 10.1 CIM\_SoftwareIdentity

906 CIM\_SoftwareIdentity is used to represent either Installed Software or Available Software. Table 12  
 907 contains the requirements for elements of this class.

908 **Table 12 – Class: CIM\_SoftwareIdentity**

Elements	Requirement	Notes
InstanceID	Mandatory	<b>Key</b>
IsEntity	Mandatory	See sections 7.2, 7.6, and 7.9.
VersionString	Optional	
MajorVersion	Conditional	See section 7.3.
MinorVersion	Conditional	See section 7.3.
RevisionNumber	Conditional	See section 7.3.

Elements	Requirement	Notes
BuildNumber	Conditional	See section 7.3.
TargetOSTypes[]	Optional	See section 7.6.5.
TargetOperatingSystems[]	Optional	See section 7.6.5.
IdentityInfoType[]	Optional	See section 7.8.2.
IdentityInfoValue[]	Optional	See section 7.8.2.
Classifications[]	Optional	See sections 7.7 and 7.8.2.

909 **10.2 CIM\_InstalledSoftwareIdentity**

910 CIM\_InstalledSoftwareIdentity is used to associate an instance of CIM\_System and an instance of  
 911 CIM\_SoftwareIdentity. CIM\_InstalledSoftwareIdentity is conditional and shall be implemented when  
 912 Installed Software is modeled. Table 13 contains the requirements for elements of this class.

913 **Table 13 – Class: CIM\_InstalledSoftwareIdentity**

Elements	Requirement	Notes
System	Mandatory	<b>Key:</b> This property is a reference to the Scoping Instance. Cardinality *
InstalledSoftware	Mandatory	<b>Key:</b> This property is a reference to the Software Identity that represents Installed Software. Cardinality *

914 **10.3 CIM\_ElementSoftwareIdentity**

915 CIM\_ElementSoftwareIdentity is used to associate an instance of CIM\_ManagedElement and an instance of  
 916 CIM\_SoftwareIdentity when the instance of CIM\_ManagedElement is instrumented. Table 14 contains  
 917 the requirements for elements of this class.

918 **Table 14 – Class: CIM\_ElementSoftwareIdentity**

Elements	Requirement	Notes
Antecedent	Mandatory	<b>Key:</b> This property is a reference to the Software Identity. Cardinality *
Dependent	Mandatory	<b>Key:</b> This property is a reference to the instance of CIM_ManagedElement. Cardinality *
ElementSoftwareStatus	Mandatory	See section 7.4.1.1.1.

## 919 10.4 CIM\_SystemSpecificCollection

920 CIM\_SystemSpecificCollection is used to represent a collection of Available Software. Table 15 contains  
921 the requirements for elements of this class.

922 **Table 15 – Class: CIM\_SystemSpecificCollection**

Elements	Requirement	Notes
InstanceID	Mandatory	<b>Key</b>
ElementName	Mandatory	See section 7.6.1.

## 923 10.5 CIM\_HostedCollection

924 CIM\_HostedCollection is used to associate CIM\_System and CIM\_SystemSpecificCollection.  
925 CIM\_HostedCollection is conditional and shall be implemented when an instance of  
926 CIM\_SystemSpecificCollection is instrumented. Table 16 contains the requirements for elements of this  
927 class.

928 **Table 16 – Class: CIM\_HostedCollection**

Elements	Requirement	Notes
OwningElement	Mandatory	<b>Key:</b> This property is a reference to the Scoping Instance. Cardinality 1
OwnedElement	Mandatory	<b>Key:</b> This property is a reference to the collection of Available Software. Cardinality 0..1

## 929 10.6 CIM\_MemberOfCollection

930 CIM\_MemberOfCollection is used to associate an instance of CIM\_SystemSpecificCollection and an  
931 instance of CIM\_SoftwareIdentity. CIM\_MemberOfCollection is conditional and shall be implemented  
932 when an instance of CIM\_SystemSpecificCollection is instrumented. Table 17 contains the requirements  
933 for elements of this class.

934 **Table 17 – Class: CIM\_MemberOfCollection**

Elements	Requirement	Notes
Collection	Mandatory	<b>Key:</b> This property is a reference to the collection of Available Software. Cardinality *
Member	Mandatory	<b>Key:</b> This property is a reference to the instance of CIM_SoftwareIdentity that represents an Available Software. Cardinality *



935 **10.7 CIM\_SoftwareIdentityResource**

936 CIM\_SoftwareIdentityResource is used to represent the location of a Software Identity, which could be  
 937 used as input to the software installation service (see the [Software Update Profile](#)). Table 18 contains the  
 938 requirements for elements of this class.

939 **Table 18 – Class: CIM\_SoftwareIdentityResource**

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	<b>Key</b>
SystemName	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
Name	Mandatory	<b>Key</b>
InfoFormat	Mandatory	
AccessInfo	Mandatory	
ResourceType	Optional	

940 **10.8 CIM\_SAPAvailableForElement**

941 CIM\_SAPAvailableForElement is used to associate CIM\_SoftwareIdentityResource and  
 942 CIM\_SoftwareIdentity. CIM\_SAPAvailableForElement is conditional and shall be implemented when the  
 943 location information of CIM\_SoftwareIdentity is represented. Table 19 contains the requirements for  
 944 elements of this class.

945 **Table 19 – Class: CIM\_SAPAvailableForElement**

Elements	Requirement	Notes
AvailableSAP	Mandatory	<b>Key:</b> This property is a reference to the CIM_SoftwareIdentityResource instance. Cardinality 1
ManagedElement	Mandatory	<b>Key:</b> This property is a reference to the Software Identity. Cardinality 0..1

946 **10.9 CIM\_HostedAccessPoint**

947 CIM\_HostedAccessPoint is used to associate CIM\_System and CIM\_SoftwareIdentityResource when an  
 948 instance of CIM\_SoftwareIdentityResource is instrumented. Table 20 contains the requirements for  
 949 elements of this class.

950 **Table 20 – Class: CIM\_HostedAccessPoint**

Elements	Requirement	Notes
Antecedent	Mandatory	<b>Key:</b> This property is a reference to the scoping CIM_System instance. Cardinality 1
Dependent	Mandatory	<b>Key:</b> This property is a reference to instance of CIM_SoftwareIdentityResource. Cardinality *

951 **10.10 CIM\_OrderedComponent**

952 CIM\_OrderedComponent is used to associate an instance of CIM\_SoftwareIdentity that represents a  
 953 Software Bundle and an instance of CIM\_SoftwareIdentity that represents one of the discrete software  
 954 images contained in the Software Bundle. Table 21 contains the requirements for elements of this class.

955 **Table 21 – Class: CIM\_OrderedComponent**

Elements	Requirement	Notes
GroupComponent	Mandatory	<b>Key:</b> See section 7.7.1. Cardinality *
PartComponent	Mandatory	<b>Key:</b> See section 7.7.2. Cardinality *
AssignedSequence	Mandatory	See section 7.7.3.

956 **10.11 CIM\_OrderedDependency**

957 CIM\_OrderedDependency is used to associate an instance of CIM\_SoftwareIdentity that represents an  
 958 Installation Dependency and an instance of CIM\_SoftwareIdentity for which the Installation Dependencies  
 959 are represented. Table 22 contains the requirements for elements of this class.

960 **Table 22 – Class: CIM\_OrderedDependency**

Elements	Requirement	Notes
Antecedent	Mandatory	<b>Key:</b> See section 7.9.1.1. Cardinality *
Dependent	Mandatory	<b>Key:</b> See section 7.9.1.2. Cardinality *
AssignedSequence	Mandatory	See section 7.9.1.3.

961 **10.12 CIM\_RegisteredProfile**

962 The CIM\_RegisteredProfile class is defined by the [Profile Registration Profile](#). The requirements denoted  
 963 in Table 23 are in addition to those mandated by the [Profile Registration Profile](#).

964 **Table 23 – Class: CIM\_RegisteredProfile**

Elements	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of "Software Inventory".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

965 NOTE: Previous versions of this document included the suffix "Profile" for the RegisteredName value. If  
 966 implementations querying for the RegisteredName value find the suffix "Profile", they should ignore the suffix, with  
 967 any surrounding white spaces, before any comparison is done with the value as specified in this document.

968  
969  
970  
971

## **ANNEX A (informative)**

### **Change Log**

<b>Version</b>	<b>Date</b>	<b>Description</b>
1.0.0	2007/11/21	Final Standard
1.0.1	2009/06/17	DMTF Standard Release

972