



1

2

3

4

Document Number: DSP1023

Date: 2008-09-04

Version: 1.0.0

5 **Software Inventory Profile**

6 **Document Type: Specification**

7 **Document Status: Final Standard**

8 **Document Language: E**

9 Copyright Notice

10 Copyright © 2008 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 for uses consistent with this purpose, provided that correct attribution is given. As DMTF
14 specifications may be revised from time to time, the particular version and release date should always be
15 noted.

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

CONTENTS

30	Foreword	6
31	Introduction	7
32	1 Scope	9
33	2 Normative References.....	9
34	2.1 Approved References	9
35	2.2 References under Development	9
36	2.3 Other References.....	9
37	3 Terms and Definitions	9
38	4 Symbols and Abbreviated Terms	11
39	5 Synopsis	12
40	6 Description	12
41	7 Implementation.....	14
42	7.1 Representing Software	14
43	7.2 Representing Installed Software.....	14
44	7.3 Representing Version Information of Software.....	14
45	7.4 Representing Relationships between Software Identity and Managed Element.....	14
46	7.5 Finding the Scoping Instance of the CIM_System Class.....	16
47	7.6 Representing Available Software.....	16
48	7.7 Representing a Software Bundle	17
49	7.8 Identifying a Software Identity.....	18
50	7.9 Representing Installation Dependencies	19
51	7.10 Version Comparison Using the MajorVersion, MinorVersion, RevisionNumber, and	
52	BuildNumber Properties.....	20
53	8 Methods.....	20
54	8.1 Profile Conventions for Operations.....	20
55	8.2 CIM_SoftwareIdentity.....	21
56	8.3 CIM_InstalledSoftwareIdentity	21
57	8.4 CIM_ElementSoftwareIdentity	21
58	8.5 CIM_SystemSpecificCollection	22
59	8.6 CIM_HostedCollection	22
60	8.7 CIM_MemberOfCollection	23
61	8.8 CIM_SoftwareIdentityResource	23
62	8.9 CIM_SAPAvailableForElement.....	23
63	8.10 CIM_HostedAccessPoint	23
64	8.11 CIM_OrderedComponent	23
65	8.12 CIM_OrderedDependency.....	24
66	9 Use Cases.....	24
67	9.1 Object Diagrams	24
68	9.2 Find All the Software Installed on All the Managed Elements within the Scope of a	
69	Managed System	34
70	9.3 Find All the Software Installed on a Managed Element.....	34
71	9.4 Find All the Software That Is Compatible with a Managed Element but Has Not Been	
72	Installed.....	35
73	9.5 Find All the Software That Is Available for Installation on Any Managed Element within	
74	the Scope of a Managed System.....	35
75	9.6 For a Given NIC, Find the Driver That Is Running in the Operating System.....	35
76	9.7 Set a Particular Software Image on a Hardware Managed Element to Run After the Next	
77	Reset or Reboot.....	35
78	9.8 Set a Particular Software Image on a Hardware Managed Element to Run After the Next	
79	Reset or Reboot But Not After a Subsequent Reset or Reboot	36
80	9.9 Find and Set a Driver to Run After the Next Reset or Reboot for a NIC	36

81	9.10 Find the Most Recent Firmware Available for a NIC	36
82	9.11 Find the Most Recent Firmware Installed on a NIC.....	36
83	9.12 Find the Software Families of Which a Software Identity Is a Member	37
84	9.13 Determine Whether a Dependency of a Software Identity Is Satisfied.....	37
85	10 CIM Elements.....	37
86	10.1 CIM_SoftwareIdentity.....	38
87	10.2 CIM_InstalledSoftwareIdentity	38
88	10.3 CIM_ElementSoftwareIdentity	39
89	10.4 CIM_SystemSpecificCollection	39
90	10.5 CIM_HostedCollection	39
91	10.6 CIM_MemberOfCollection	40
92	10.7 CIM_SoftwareIdentityResource	40
93	10.8 CIM_SAPAvailableForElement.....	41
94	10.9 CIM_HostedAccessPoint	41
95	10.10 CIM_OrderedComponent	41
96	10.11 CIM_OrderedDependency.....	42
97	10.12 CIM_RegisteredProfile.....	42
98	ANNEX A (informative) Change Log.....	43
99	ANNEX B (informative) Acknowledgments	44

100

101 Figures

102	Figure 1 – Class Diagram: Software Inventory Profile.....	13
103	Figure 2 – Registered Profile	24
104	Figure 3 – Object Diagram Showing Installed Software	25
105	Figure 4 – Object Diagram Showing an Installed Driver.....	26
106	Figure 5 – Object Diagram Showing Installed BIOS.....	26
107	Figure 6 – Object Diagram Showing Installed Software	27
108	Figure 7 – Object Diagram Showing Multiple Installed Software on a Managed Element	28
109	Figure 8 – Object Diagram with No Instantiation of Managed Element.....	28
110	Figure 9 – Object Diagram Showing Available Firmware	29
111	Figure 10 – Object Diagram Showing an Available Driver.....	30
112	Figure 11 – Object Diagram Showing a Firmware Image and Its Location	31
113	Figure 12 – Object Diagram Showing a Software Bundle	32
114	Figure 13 – Object Diagram Showing Available Software That Is Part of a Software Bundle.....	33
115	Figure 14 – Object Diagram Showing Installed and Available Software	34

116

117 Tables

118	Table 1 – Referenced Profiles	12
119	Table 2 – Relationships Between Enumeration Values of ElementSoftwareStatus.....	15
120	Table 3 – Operations: CIM_InstalledSoftwareIdentity	21
121	Table 4 – Operations: CIM_ElementSoftwareIdentity.....	21
122	Table 5 – Operations: CIM_HostedCollection.....	22
123	Table 6 – Operations: CIM_MemberOfCollection.....	23
124	Table 7 – Operations: CIM_SAPAvailableForElement	23
125	Table 8 – Operations: CIM_HostedAccessPoint	23
126	Table 9 – Operations: CIM_OrderedComponent.....	24
127	Table 10 – Operations: CIM_OrderedDependency	24
128	Table 11 – CIM Elements: Software Inventory Profile	37

129 Table 12 – Class: CIM_SoftwareIdentity..... 38
130 Table 13 – Class: CIM_InstalledSoftwareIdentity 38
131 Table 14 – Class: CIM_ElementSoftwareIdentity 39
132 Table 15 – Class: CIM_SystemSpecificCollection 39
133 Table 16 – Class: CIM_HostedCollection 39
134 Table 17 – Class: CIM_MemberOfCollection..... 40
135 Table 18 – Class: CIM_SoftwareIdentityResource 40
136 Table 19 – Class: CIM_SAPAvailableForElement..... 41
137 Table 20 – Class: CIM_HostedAccessPoint 41
138 Table 21 – Class: CIM_OrderedComponent..... 41
139 Table 22 – Class: CIM_OrderedDependency 42
140 Table 23 – Class: CIM_RegisteredProfile..... 42
141

142

Foreword

143 The *Software Inventory Profile* (DSP1023) was prepared by the Server Management Working Group.

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

146

Introduction

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

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

154

Software Inventory Profile

1 Scope

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

2 Normative References

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

2.1 Approved References

164 DMTF [DSP0200](#), *CIM Operations over HTTP 1.2.0*

165 DMTF [DSP0004](#), *CIM Infrastructure Specification 2.3.0*

166 DMTF [DSP1000](#), *Management Profile Specification Template 1.0.0*

167 DMTF [DSP1001](#), *Management Profile Specification Usage Guide 1.0.0*

168 DMTF [DSP1033](#), *Profile Registration Profile 1.0.0*

2.2 References under Development

170 DMTF [DSP1025](#), *Software Update Profile 1.0.0*

2.3 Other References

172 ISO/IEC Directives, Part 2, [Rules for the structure and drafting of International Standards](#)

173 OMG, [Unified Modeling Language \(UML\) from the Open Management Group \(OMG\)](#)

3 Terms and Definitions

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

3.1

can

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

3.2

cannot

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

- 183 **3.3**
184 **conditional**
185 indicates requirements to be followed strictly to conform to the document when the specified conditions
186 are met
- 187 **3.4**
188 **mandatory**
189 indicates requirements to be followed strictly to conform to the document and from which no deviation is
190 permitted
- 191 **3.5**
192 **may**
193 indicates a course of action permissible within the limits of the document
- 194 **3.6**
195 **need not**
196 indicates a course of action permissible within the limits of the document
- 197 **3.7**
198 **optional**
199 indicates a course of action permissible within the limits of the document
- 200 **3.8**
201 **referencing profile**
202 indicates a profile that owns the definition of this class and can include a reference to this profile in its
203 "Referenced Profiles" table
- 204 **3.9**
205 **shall**
206 indicates requirements to be followed strictly to conform to the document and from which no deviation is
207 permitted
- 208 **3.10**
209 **shall not**
210 indicates requirements to be followed strictly to conform to the document and from which no deviation is
211 permitted.
- 212 **3.11**
213 **should**
214 indicates that among several possibilities, one is recommended as particularly suitable, without
215 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 216 **3.12**
217 **should not**
218 indicates that a certain possibility or course of action is deprecated but not prohibited
- 219 **3.13**
220 **unspecified**
221 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 222 **3.14**
223 **Software Identity**
224 an instance of CIM_SoftwareIdentity that represents and contains the identifying property values of a
225 software image

226 **3.15**

227 **Installed Software**

228 software that is installed on any managed element in the scope of a system

229 **3.16**

230 **Available Software**

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

234 **3.17**

235 **Software Bundle**

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

238 **3.18**

239 **Managed Element**

240 an instance of CIM_ManagedElement that represents a managed element in the scope of a system

241 **3.19**

242 **Software Family**

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

245 **3.20**

246 **Installation Dependency**

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

248 **4 Symbols and Abbreviated Terms**

249 **Experimental Maturity Level**

250

251 Some of the content considered for inclusion in *Software Inventory Profile* has yet to receive sufficient
252 review to satisfy the adoption requirements set forth by the Technical Committee within the DMTF. This
253 content is presented here as an aid to implementers who are interested in likely future developments
254 within this specification. The content marked experimental may change as implementation experience is
255 gained. There is a high likelihood that it will be included in an upcoming revision of the specification. Until
256 that time, it is purely informational, and is clearly marked within the text.

257 A sample of the typographical convention for experimental content is included here:

258 **EXPERIMENTAL**

259 Experimental content appears here

260 **EXPERIMENTAL**

261 The following symbols and abbreviations are used in this document.

262 None

263 5 Synopsis

264 **Profile Name:** Software Inventory

265 **Version:** 1.0.0

266 **Organization:** DMTF

267 **CIM Schema version:** 2.20

268 **Central Class:** CIM_SoftwareIdentity

269 **Scoping Class:** CIM_System

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

276 CIM_SoftwareIdentity shall be the Central Class of this profile. The instance of CIM_SoftwareIdentity shall
 277 be the Central Instance of this profile.

278 CIM_System shall be the Scoping Class of this profile. The instance of CIM_System shall be the Scoping
 279 Instance of this profile and shall be selected using the algorithm described in section 7.5.

280 References to CIM_System may be interpreted as references to subclasses of CIM_System such as
 281 CIM_ComputerSystem. Table 1 identifies profiles on which this profile has a dependency.

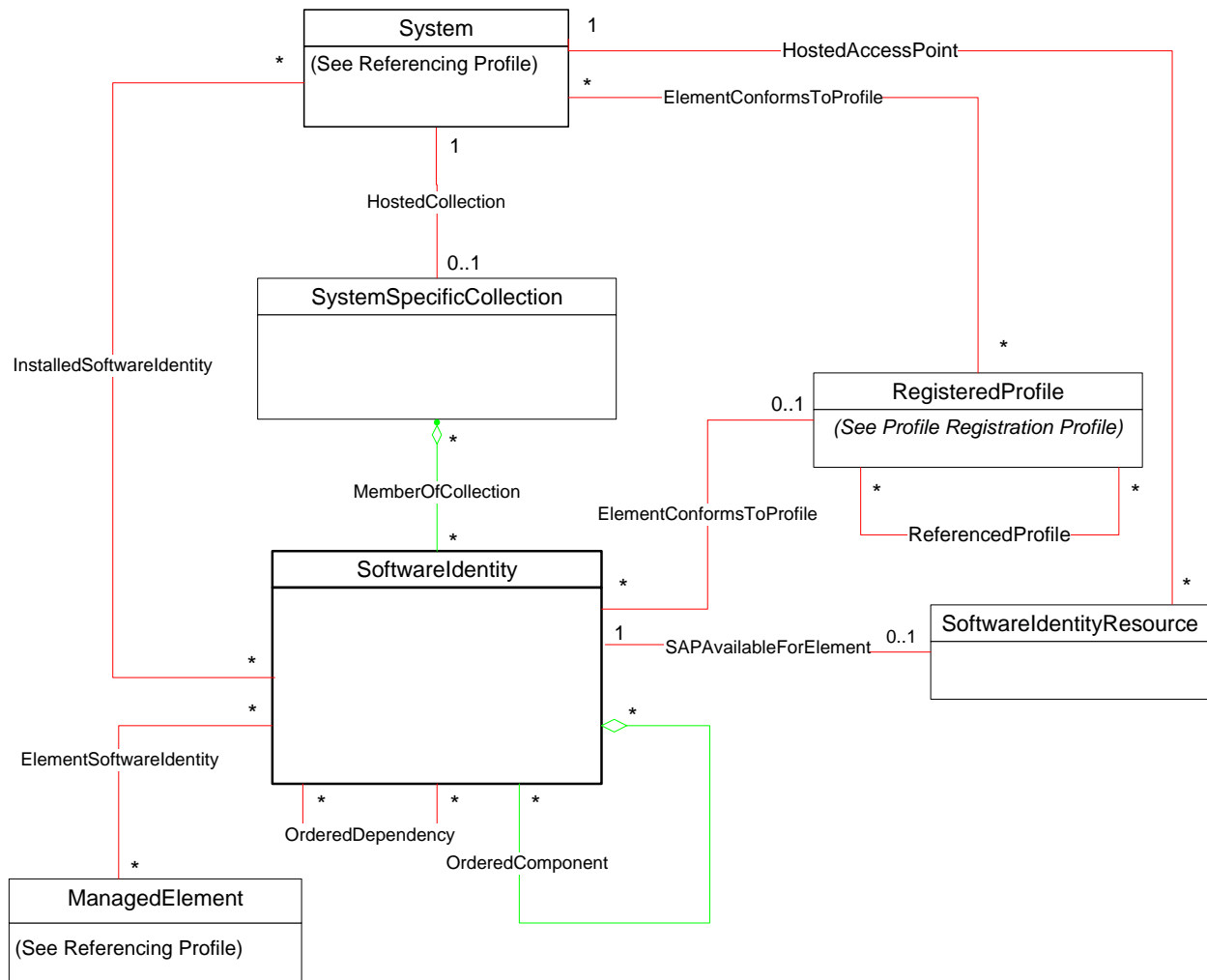
282

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Description
Profile Registration	DMTF	1.0	Mandatory

283 6 Description

284 The *Software Inventory Profile* provides the ability to perform an inventory of installed BIOS, firmware,
 285 drivers, and related software such as providers and instrumentation software. This profile also describes
 286 the CIM schema elements required to represent the software that can be installed on a managed system.
 287 It also provides information about what software is associated with particular Managed Elements such as
 288 devices. Figure 1 represents the class schema of the *Software Inventory Profile* and shows the elements
 289 of the *Software Inventory Profile*, as well as the dependent relationships between the elements of
 290 *Software Inventory Profile* and the referencing profiles. For simplicity, the prefix *CIM_* has been removed
 291 from the names of the classes.



292

293

Figure 1 – Class Diagram: Software Inventory Profile

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

- 295 • the software that is installed on any Managed Element in the scope of the managed system (see
- 296 section 7.2) so that the user of the profile can inventory the installed software for the managed
- 297 system
- 298 • the software that is available for installation on any Managed Element in the scope of the
- 299 managed system (see section 7.6) for providing the user of the profile the capability to view all the
- 300 software that is available for any Managed Element within the scope of the managed system

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

303 7 Implementation

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

307 7.1 Representing Software

308 The implementation shall model Installed Software (see section 7.2), Available Software (see section
309 7.6), or both, as a part of this profile.

310 7.2 Representing Installed Software

311 When an implementation models Installed Software, each Installed Software image modeled by the
312 implementation shall be represented by exactly one instance of CIM_SoftwareIdentity. The IsEntity
313 property of the instance of CIM_SoftwareIdentity shall have the value true.

314 7.2.1 CIM_InstalledSoftwareIdentity Instance

315 The Software Identity that represents an Installed Software shall be associated to the Scoping Instance
316 using exactly one instance of CIM_InstalledSoftwareIdentity.

317 7.3 Representing Version Information of Software

318 When the version information is not represented using the VersionString property, it shall be represented
319 using the MajorVersion, MinorVersion, RevisionNumber, and BuildNumber properties. These properties
320 are conditional and shall be implemented when the VersionString property is Null. When MinorVersion
321 has a non-Null value, MajorVersion shall have a non-Null value. When RevisionNumber has a non-Null
322 value, MinorVersion shall have a non-Null value. When BuildNumber has a non-Null value,
323 RevisionNumber shall have a non-Null value. The algorithm for comparing versions of two instances of
324 CIM_SoftwareIdentity using these properties is described in section 7.10.

325 7.4 Representing Relationships between Software Identity and Managed 326 Element

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

330 7.4.1 CIM_ElementSoftwareIdentity Instance

331 When a Managed Element is represented, the relationships between the Managed Element and the
332 compatible Software Identity shall be represented using an instance of CIM_ElementSoftwareIdentity.

333 When the Managed Element is not represented with an instance, the relationship between the compatible
334 Software Identity and the Managed Element may be still represented by associating the Software Identity
335 to the Scoping Instance through an instance of CIM_ElementSoftwareIdentity.

336 7.4.1.1 CIM_ElementSoftwareIdentity.ElementSoftwareStatus

337 The CIM_ElementSoftwareIdentity.ElementSoftwareStatus property shall represent the relationships of
338 the software, represented by the Software Identity, to the Managed Element, through one or more
339 enumeration values.

340 If the relationship between the Software Identity and the Managed Element is unknown, then the
341 CIM_ElementSoftwareIdentity.ElementSoftwareStatus property shall contain no enumeration values.

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

343 **7.4.1.1.1 CIM_ElementSoftwareIdentity.ElementSoftwareStatus Enumeration Relationships**

344 The relationships between the ElementSoftwareStatus property enumeration values on a single instance
 345 of CIM_ElementSoftwareIdentity are described in Table 2. When the ElementSoftwareStatus property of
 346 an instance of CIM_ElementSoftwareIdentity has the value specified in the "Enumeration Value" column
 347 of Table 2, the ElementSoftwareStatus property of the same instance shall also have other enumeration
 348 values specified in the corresponding row of the "Mandatory Pairing With" column of Table 2.

349 When the ElementSoftwareStatus property of an instance of CIM_ElementSoftwareIdentity has the value
 350 specified in the "Enumeration Value" column of Table 2, the ElementSoftwareStatus property of the same
 351 instance may also have other enumeration values specified in the corresponding row of the "May Be
 352 Used With" column of Table 2.

353 When the ElementSoftwareStatus property of an instance of CIM_ElementSoftwareIdentity has the value
 354 specified in the "Enumeration Value" column of Table 2, the ElementSoftwareStatus property of the same
 355 instance shall not have other enumeration values specified in the corresponding row of the "Shall Not Be
 356 Used With" column of Table 2.

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

361 **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)

362 When a Software Identity that is associated with a Managed Element through an instance of
 363 CIM_ElementSoftwareIdentity with the ElementSoftwareStatus property containing the value 3 (Next) or 7
 364 (SingleUse) fails to run, the system shall automatically attempt to use the Software Identity that is
 365 associated with the same Managed Element through an instance of CIM_ElementSoftwareIdentity with
 366 the ElementSoftwareStatus property containing the value 4 (FallBack), and no client action shall be
 367 required.

368 **7.4.2 ElementSoftwareIdentity for Software That Is Intended for a Managed Element But** 369 **Does Not Run or Get installed on It**

370 When an instance of CIM_ElementSoftwareIdentity is used to represent the relationship between a
371 Software Identity and a Managed Element such that the Software Identity will work with or can operate
372 the Managed Element but is installed and runs on a different Managed Element, the only value that the
373 ElementSoftwareIdentity.ElementSoftwareStatus property shall have is 9 (Supports).

374 **7.5 Finding the Scoping Instance of the CIM_System Class**

375 The following algorithm shall be used for locating the Scoping Instance of the CIM_System class from any
376 instance of CIM_SoftwareIdentity:

- 377 1) If the selected instance is referenced by an instance of CIM_InstalledSoftwareIdentity, the Scoping
378 Instance shall be the instance of CIM_System that is associated through the instance of
379 CIM_InstalledSoftwareIdentity.
- 380 2) Otherwise, if the selected instance is referenced by the instance of CIM_MemberOfCollection, select
381 the instance of CIM_SystemSpecificCollection that is associated through the instance of
382 CIM_MemberOfCollection. The Scoping Instance of the profile shall be the instance of CIM_System
383 that is associated with the selected instance of CIM_SystemSpecificCollection through the instance
384 of CIM_HostedCollection.

385 **7.6 Representing Available Software**

386 When an implementation represents the Installed Software with Available Software, each Available
387 Software image modeled by the implementation shall be represented by a Software Identity. The IsEntity
388 property of the instance of CIM_SoftwareIdentity shall have the value true. The following subsections are
389 applicable when Available Software is represented.

390 **7.6.1 CIM_SystemSpecificCollection Instance**

391 An implementation shall instantiate a single instance of CIM_SystemSpecificCollection, which is a
392 collection of all the Available Software. The ElementName property of this instance of
393 CIM_SystemSpecificCollection shall have a value of "Available Software".

394 **7.6.2 CIM_HostedCollection Instance**

395 The instance of CIM_SystemSpecificCollection shall be associated to the Scoping Instance by exactly
396 one instance of CIM_HostedCollection.

397 **7.6.3 CIM_MemberOfCollection Instance**

398 For each Software Identity that represents an Available Software, exactly one instance of
399 CIM_MemberOfCollection shall associate the Software Identity to the CIM_SystemSpecificCollection
400 instance.

401 **EXPERIMENTAL**

402 **7.6.4 Advertising the Location Information of a Software Identity**

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

405 7.6.4.1 CIM_SoftwareIdentityResource Instance

406 The location of a Software Identity shall be represented by an instance of CIM_SoftwareIdentityResource.
407 This could be used as an input to the software installation service. See the "Implementation" section of
408 the *Software Update Profile*.

409 7.6.4.2 CIM_SAPAvailableForElement Instance

410 An instance of CIM_SAPAvailableForElement shall be used to associate a Software Identity with a
411 CIM_SoftwareIdentityResource instance that represents the location information of the Software Identity.

412 7.6.4.3 CIM_HostedAccessPoint

413 An instance of CIM_HostedAccessPoint shall be used to associate a CIM_SoftwareIdentityResource
414 instance and the CIM_System or CIM_ComputerSystem instance that represents the Scoping Instance of
415 the Available Software whose location information is advertised by the CIM_SoftwareIdentityResource
416 instance.

417 EXPERIMENTAL

418 7.6.5 Identifying Target Operating Systems

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

422 7.6.5.1 CIM_SoftwareIdentity.TargetOSTypes[]

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

427 7.6.5.2 CIM_SoftwareIdentity.TargetOperatingSystems[]

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

430 EXPERIMENTAL

431 7.7 Representing a Software Bundle

432 A Software Bundle may be modeled. This behavior is optional. A Software Bundle shall be represented
433 using a Software Identity. The Software Identity shall have a value of 13 (Software Bundle) in the
434 Classifications[] property. Each software image in the Software Bundle shall be represented by a
435 Software Identity that shall be associated to the Software Identity that represents the Software Bundle,
436 using a single instance of CIM_OrderedComponent.

437 7.7.1 CIM_OrderedComponent.GroupComponent

438 The instance of CIM_SoftwareIdentity that represents the Software Bundle shall be the value of the
439 GroupComponent property.

440 7.7.2 CIM_OrderedComponent.PartComponent

441 The instance of CIM_SoftwareIdentity that represents the individual software image that is a part of the
442 Software Bundle shall be the value of the PartComponent property.

443 7.7.3 CIM_OrderedComponent.AssignedSequence

444 The AssignedSequence property indicates the order in which the Software Identity referenced by the
445 CIM_OrderedComponent instance shall be installed during the installation of the bundle. The Software
446 Identity with the lowest value of AssignedSequence on the associated CIM_OrderedComponent instance
447 shall be installed first and the highest shall be installed last. An AssignedSequence value of zero shall
448 indicate no ordering requirement. Equivalent values of the AssignedSequence property shall indicate no
449 ordering preference.

450 7.8 Identifying a Software Identity

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

453 7.8.1 General Use of IdentityInfoType and IdentityInfoValue Properties

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

457 7.8.2 Using IdentityInfoType and IdentityInfoValue to Model a Software Family

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

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

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

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

473 <CIM_SoftwareIdentity.Classifications[]>:<CIM_SoftwareIdentity.TargetOSTypes[]>:<
474 Information of the Hardware/ Hardware family supported by the Software Identity>

475 <CIM_SoftwareIdentity.Classifications[]> is one of the numeric values contained in the
476 Classifications property, and <CIM_SoftwareIdentity.TargetOSTypes[]> is one of the
477 values contained in the TargetOSTypes property of the instance of CIM_SoftwareIdentity.

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

481 <CIM_SoftwareIdentity.Classifications[]>:< Information of the Hardware/ Hardware family
482 supported by the Software>

483 <CIM_SoftwareIdentity.Classifications[]> is one of the numeric values contained in the
484 Classifications property of the instance of CIM_SoftwareIdentity.

485 **7.8.2.1 Determining Common Software Family Membership**

486 Two instances of CIM_SoftwareIdentity shall belong to the same Software Family when at least one of
487 the Software Families modeled for the first CIM_SoftwareIdentity instance matches at least one of the
488 Software Families modeled for the second CIM_SoftwareIdentity instance.

489 **EXPERIMENTAL**

490 **EXPERIMENTAL**

491 **7.9 Representing Installation Dependencies**

492 Software on which a Software Identity is dependent may be modeled. This behavior is optional. When
493 information about the dependency is known but a copy of the software is not modeled, the dependency
494 shall be modeled using an instance of CIM_SoftwareIdentity and the IsEntity property shall have the
495 value false. When information about the dependency is known and a copy of the software is modeled, the
496 dependency shall be modeled using an instance of CIM_SoftwareIdentity and the IsEntity property shall
497 have the value true.

498 **7.9.1 CIM_OrderedDependency**

499 When a Software Identity that is a member of the Available Software collection has installation
500 dependencies on software that is represented by an instance of CIM_SoftwareIdentity, the
501 instrumentation shall instantiate an instance of the CIM_OrderedDependency association between the
502 Software Identity and each Installation Dependency, represented by an instance of CIM_SoftwareIdentity,
503 to arrange the Installation Dependencies in a hierarchical order.

504 **7.9.1.1 CIM_OrderedDependency.Antecedent**

505 The instance of CIM_SoftwareIdentity that represents the Installation Dependency shall be the value of
506 the Antecedent property.

507 **7.9.1.2 CIM_OrderedDependency.Dependent**

508 The instance of CIM_SoftwareIdentity for which the Installation Dependencies are represented shall be
509 the value of the Dependent property.

510 **7.9.1.3 CIM_OrderedDependency.AssignedSequence**

511 The AssignedSequence property indicates the order or sequence in which the Installation Dependencies
512 shall be resolved during the installation of the Software Identity. The Installation Dependency with the
513 lowest value of AssignedSequence on the associated CIM_OrderedComponent instance shall be
514 installed first and the highest shall be installed last. An AssignedSequence value of zero shall indicate no
515 ordering requirement.

516 **EXPERIMENTAL**

517 7.10 Version Comparison Using the MajorVersion, MinorVersion, 518 RevisionNumber, and BuildNumber Properties

519 The following algorithm shall be used to indicate that a CIM_SoftwareIdentity instance has a higher
520 version than the other instance of CIM_SoftwareIdentity when two instances of CIM_SoftwareIdentity are
521 compared.

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

- 525 1) If the MajorVersion properties of the two instances are equal, go to step 2.
526 Otherwise, the instance with the higher value of the MajorVersion property shall be the instance with
527 the higher version.
- 528 2) If the MinorVersion properties of the two instances are equal, go to step 3.
529 Otherwise, the instance with the higher value of the MinorVersion property shall be the instance with
530 the higher version.
- 531 3) If the RevisionNumber properties of the two instances are equal, go to step 4.
532 Otherwise, the instance with the higher value of the RevisionNumber property shall be the instance
533 with the higher version.
- 534 4) If the BuildNumber properties of the two instances are equal, the two instances shall have equal
535 value.
536 Otherwise, the instance with the higher value of the BuildNumber property shall be the instance with
537 the higher version.

538 8 Methods

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

541 8.1 Profile Conventions for Operations

542 Support for operations for each profile class (including associations) is specified in the following
543 subclauses. Each of these subclauses includes either the statement “All operations in the default list in
544 section 8.1 are supported as described by [DSP0200 version 1.2](#)” or a table listing all the operations that
545 are not supported by this profile or where the profile requires behavior other than that described by
546 [DSP0200 version 1.2](#).

547 The default list of operations is as follows:

- 548 • GetInstance
- 549 • EnumerateInstances
- 550 • EnumerateInstanceNames
- 551 • Associators
- 552 • AssociatorNames
- 553 • References
- 554 • ReferenceNames

555 A compliant implementation shall support all of the operations in the default list for each class, unless the
556 “Requirement” column text states something other than *Mandatory*.

557 **8.2 CIM_SoftwareIdentity**

558 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

559 **8.3 CIM_InstalledSoftwareIdentity**

560 Table 3 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or
561 shall not be supported.

562 **Table 3 – Operations: CIM_InstalledSoftwareIdentity**

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

563 **8.4 CIM_ElementSoftwareIdentity**

564 Table 4 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or
565 shall not be supported.

566 **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

567 **8.4.1 CIM_ElementSoftwareIdentity – ModifyInstance**

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

- 569 • When the ModifyInstance operation is used to set the ElementSoftwareStatus property to contain the
570 value 3 (Next):
 - 571 1) find all the other instances of CIM_ElementSoftwareIdentity that
 - 572 a) reference the same instance of CIM_ManagedElement as the target instance of
573 CIM_ElementSoftwareIdentity and
 - 574 b) reference an instance of CIM_SoftwareIdentity that belongs to the same Software Family
575 as the instance of CIM_SoftwareIdentity that is referenced by the target instance of
576 CIM_ElementSoftwareIdentity.
 - 577 2) For each of the CIM_ElementSoftwareIdentity instances found, remove the value 3 (Next) from
578 the ElementSoftwareStatus property if present.
- 579 • The implementation shall not allow the ModifyInstance operation to add the value 2 (Current) to and
580 remove the value 2 (Current) from the ElementSoftwareStatus property.
- 581 • When the ModifyInstance operation is used to set the ElementSoftwareStatus property to contain the
582 value 4 (FallBack):

- 583 1) Find all the other instances of CIM_ElementSoftwareIdentity that
- 584 a) reference the same instance of CIM_ManagedElement as the target instance of
- 585 CIM_ElementSoftwareIdentity and
- 586 b) reference an instance of CIM_SoftwareIdentity that belongs to the same Software Family
- 587 as the instance of CIM_SoftwareIdentity that is referenced by the target instance of
- 588 CIM_ElementSoftwareIdentity.
- 589 2) For each of the CIM_ElementSoftwareIdentity instances found, remove the value 4 (Fallback)
- 590 from the ElementSoftwareStatus property if present.
- 591 • The implementation shall not allow the ModifyInstance operation to add or remove the value 5
- 592 (Default) from the ElementSoftwareStatus property.
- 593 • The implementation shall not allow the ModifyInstance operation to add or remove the value 6
- 594 (Installed) from the ElementSoftwareStatus property.
- 595 • When the ModifyInstance operation is used to set the ElementSoftwareStatus property to contain the
- 596 value 7 (SingleUse):
- 597 1) Find all the other instances of CIM_ElementSoftwareIdentity that
- 598 a) reference the same instance of CIM_ManagedElement as the target instance of
- 599 CIM_ElementSoftwareIdentity and
- 600 b) reference an instance of CIM_SoftwareIdentity that belongs to the same Software Family
- 601 as the instance of CIM_SoftwareIdentity that is referenced by the target instance of
- 602 CIM_ElementSoftwareIdentity.
- 603 2) For each of the CIM_ElementSoftwareIdentity instances found, remove the value 7 (SingleUse)
- 604 from the ElementSoftwareStatus property if present.
- 605 • The implementation shall not allow the ModifyInstance operation to remove the value 8 (Available)
- 606 from the ElementSoftwareStatus property. The implementation shall allow adding the value
- 607 8 (Available) to the ElementSoftwareStatus property only if the associated Software Identity is
- 608 associated with the CIM_SystemSpecificCollection that has the ElementName property equal to
- 609 "Available Software" through an instance of CIM_MemberOfCollection.
- 610 • The implementation shall not allow the ModifyInstance operation to add or remove the value
- 611 9 (Supports) from the ElementSoftwareStatus property.

612 8.5 CIM_SystemSpecificCollection

613 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

614 8.6 CIM_HostedCollection

615 Table 5 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or

616 shall not be supported.

617 **Table 5 – Operations: CIM_HostedCollection**

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

618 **8.7 CIM_MemberOfCollection**

619 Table 6 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or
620 shall not be supported.

621 **Table 6 – Operations: CIM_MemberOfCollection**

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

622 **8.8 CIM_SoftwareIdentityResource**

623 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

624 **8.9 CIM_SAPAvailableForElement**

625 Table 7 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or
626 shall not be supported.

627 **Table 7 – Operations: CIM_SAPAvailableForElement**

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

628 **8.10 CIM_HostedAccessPoint**

629 Table 8 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or
630 shall not be supported.

631 **Table 8 – Operations: CIM_HostedAccessPoint**

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

632 **8.11 CIM_OrderedComponent**

633 Table 9 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or
634 shall not be supported.

635

Table 9 – Operations: CIM_OrderedComponent

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

636 **8.12 CIM_OrderedDependency**

637 Table 10 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)
 638 or shall not be supported.

639

Table 10 – Operations: CIM_OrderedDependency

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

640 **9 Use Cases**

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

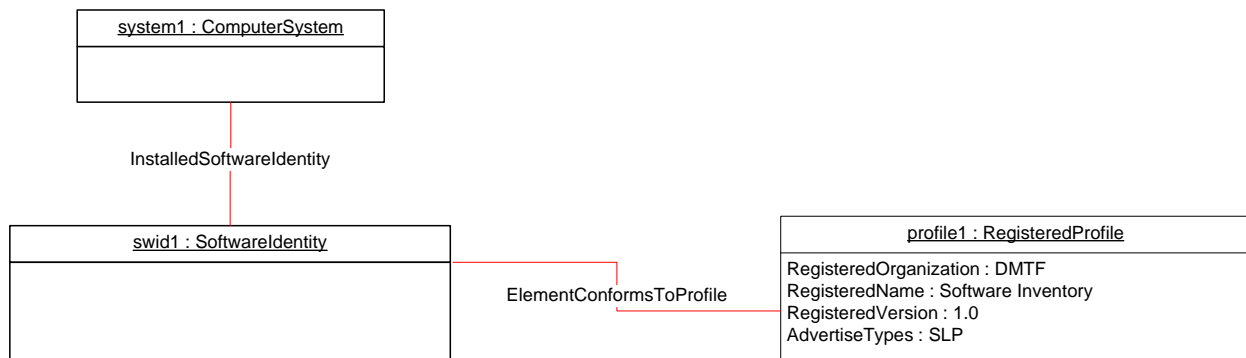
642 **9.1 Object Diagrams**

643 This section contains object diagrams for the *Software Inventory Profile*. For simplicity, the prefix *CIM_*
 644 has been removed from the names of the classes in the diagrams.

645 **9.1.1 Registered Profile**

646 Figure 2 represents a possible instantiation of the *Software Inventory Profile*. In this instantiation, the
 647 Central Instance, swid1, has an InstalledSoftwareIdentity association to the Scoping Instance, system1.
 648 Profile registration information is represented with the profile1 instance. Following the
 649 CIM_ElementConformsToProfile association from the Central Instance to profile1, the client can retrieve
 650 information such as the version of the current *Software Inventory Profile* implementation.

651



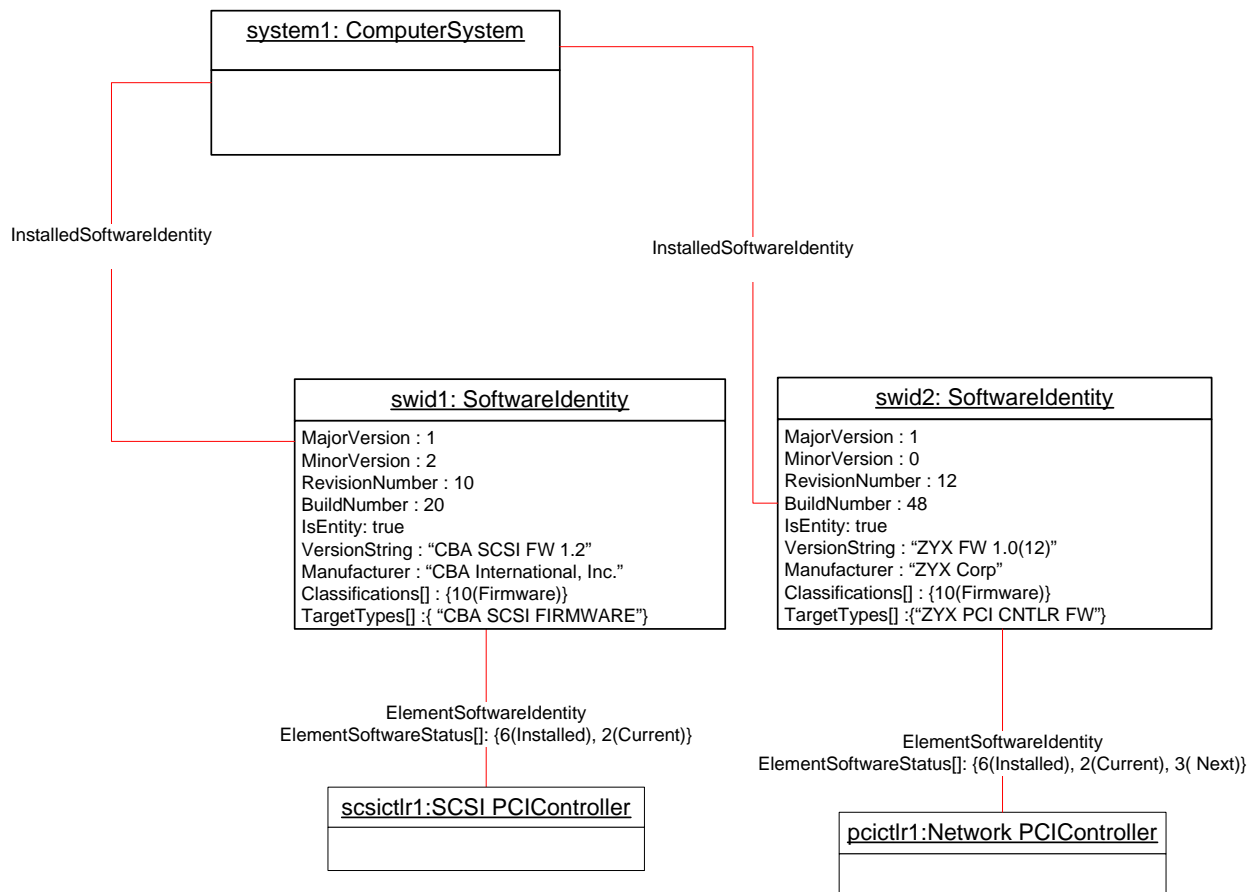
652

Figure 2 – Registered Profile

653 **9.1.2 Representing Installed Firmware**

654 Figure 3 represents a possible instantiation of the *Software Inventory Profile*. In the object diagram,
 655 Software Identity swid1 is shown as installed on the SCSI PCI Controller, scsictlr1, and currently running
 656 on it. The ElementSoftwareStatus property on the ElementSoftwareIdentity association instance between
 657 swid1 and scsictlr1 does not have the value 3 (Next) because it is not the firmware that will run after the
 658 next reboot of the system.

659 Software Identity swid2 is shown as installed on the Network PCI Controller, pcictlr1, and currently
 660 running on it. swid2 would also run on the next reset or reboot of pcictlr1. The object diagram does not
 661 show the CIM_SystemDevice association between system1 and scsictlr1, and system1 and pcictlr1, but
 662 both scsictlr1 and pcictlr1 are scoped to system1 and so the CIM_InstalledSoftwareIdentity association
 663 is shown between system1 and swid1, and system1 and swid2.

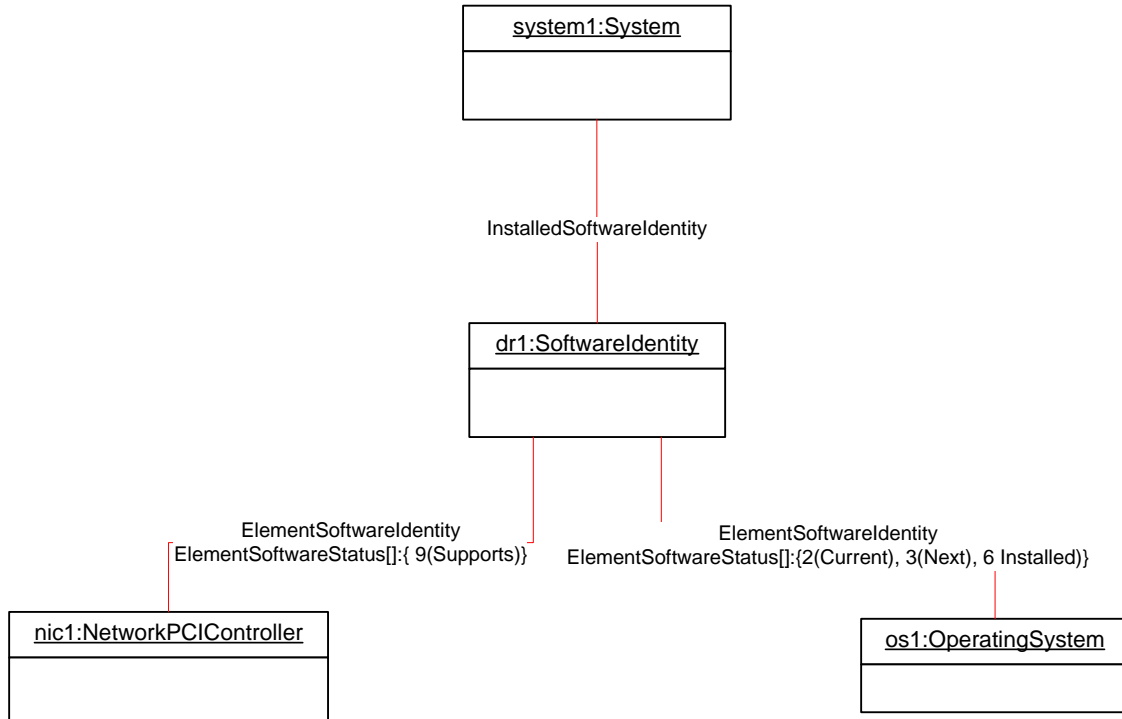


664

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

666 **9.1.3 Representing an Installed Driver**

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



671

672

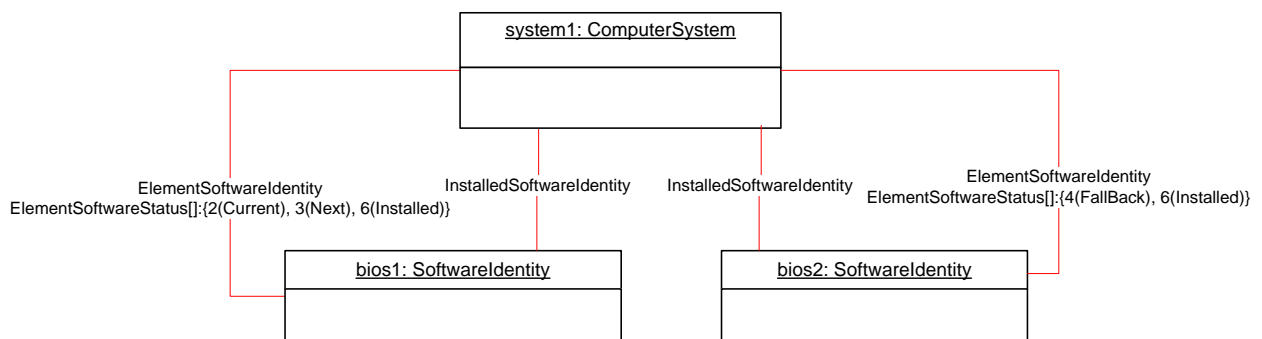
Figure 4 – Object Diagram Showing an Installed Driver

673 **9.1.4 Representing BIOS Installed on a System**

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

677 bios1 is for the system, system1, and so the CIM_ElementSoftwareIdentity association is used to
 678 associate them with the ElementSoftwareStatus property having the values 2 (Current), 3 (Next), and 6
 679 (Installed).

680 bios2 is the backup for bios1 and is also for system, system1, and so the CIM_ElementSoftwareIdentity
 681 association is used to associate them with the ElementSoftwareStatus property having the values
 682 4 (FallBack) and 6 (Installed).



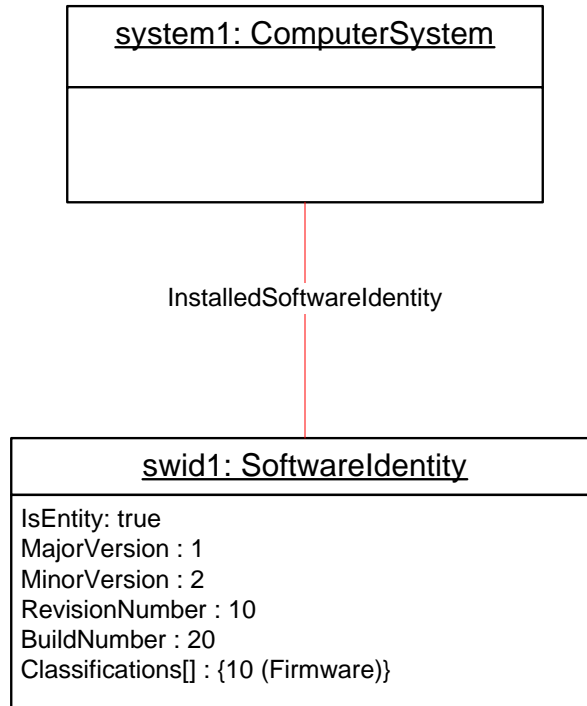
683

684

Figure 5 – Object Diagram Showing Installed BIOS

685 **9.1.5 Representing Installed Software without Any Association to the Managed Element**

686 Figure 6 represents a possible instantiation of the *Software Inventory Profile*. The firmware represented
 687 by swid1 is installed on some Managed Element in the scope of system1 but the Managed Element is not
 688 modeled by the instrumentation and since the CIM_ElementSoftwareIdentity association is not
 689 instantiated between system1 and swid1, the relationship the Managed Element and swid1 is not known.



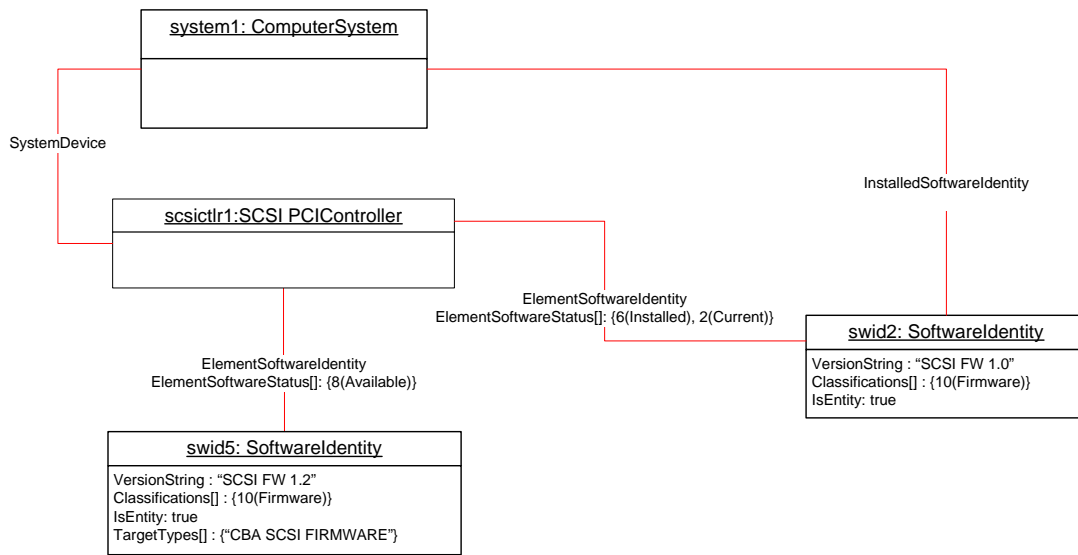
690

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

692 **9.1.6 Representing More Than One Executable Software Identity on a Managed Element**

693 Figure 7 represents a possible instantiation of the *Software Inventory Profile*. In the object diagram,
 694 Software Identity swid2 is shown as installed on the SCSI PCI Controller, scsictrl1, and is currently
 695 running on it. The ElementSoftwareStatus property on the CIM_ElementSoftwareIdentity instance that
 696 associates swid2 and scsictrl1 has the values 2 (Current) and 6 (Installed).

697 Software Identity swid3 is the manufacturer shipped version and is installed on scsictrl1 but is not
 698 currently running. The CIM_ElementSoftwareIdentity instance that associates swid3 and scsictrl1 conveys
 699 this relationship by the ElementSoftwareStatus property having the values 5 (Default) and 6 (Installed).



700

701

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

702

9.1.7 Representing Available and Installed Firmware without Managed Element

703

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.

704

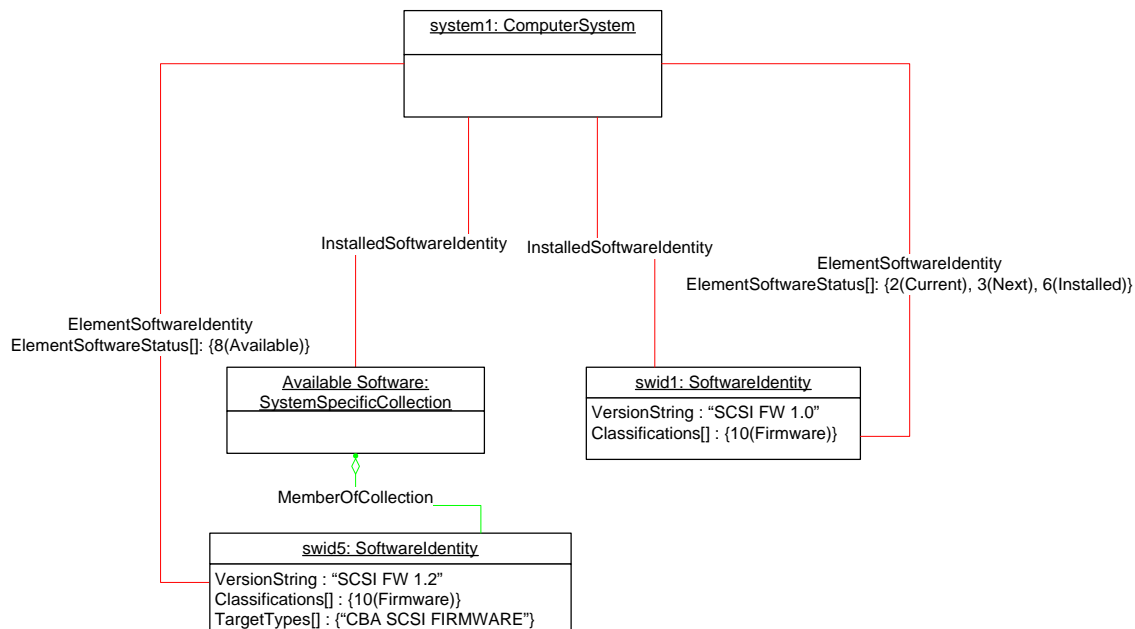
705

706

707

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

708



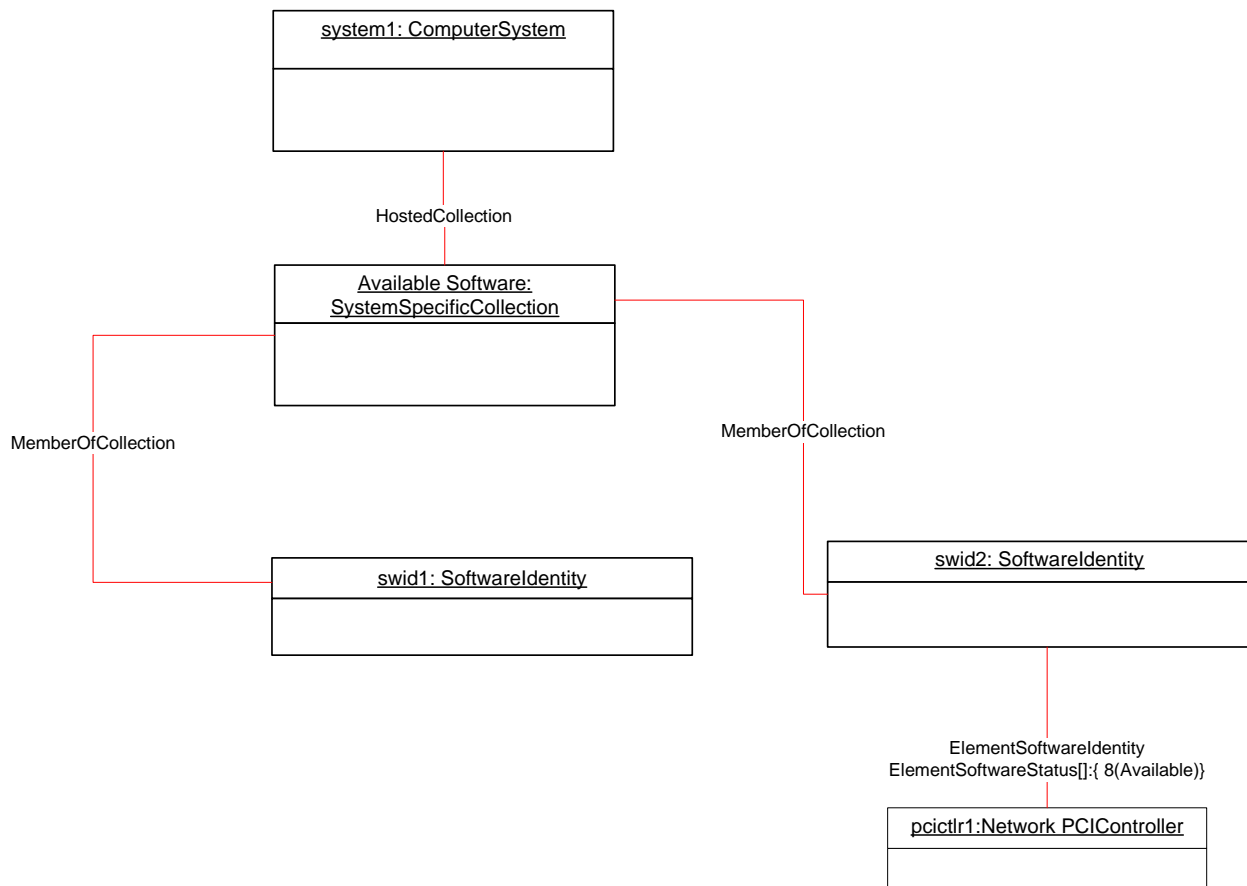
709

710

Figure 8 – Object Diagram with No Instantiation of Managed Element

711 **9.1.8 Representing Available Firmware**

712 Figure 9 represents a possible instantiation of the *Software Inventory Profile*. In the object diagram,
 713 Software Identity swid2 is shown as available for installation on the Network PCI Controller, pcictrl1,
 714 using the CIM_ElementSoftwareIdentity association. Software Identity swid1 is an Available Software but
 715 the compatible Managed Element is not modeled and no CIM_ElementSoftwareIdentity instance
 716 references swid1. pcictrl1 is scoped to system1, but the object diagram does not show the
 717 CIM_SystemDevice association between system1 and pcictrl1, and so the CIM_MemberOfCollection
 718 association is shown between an “Available Software” collection and swid2.



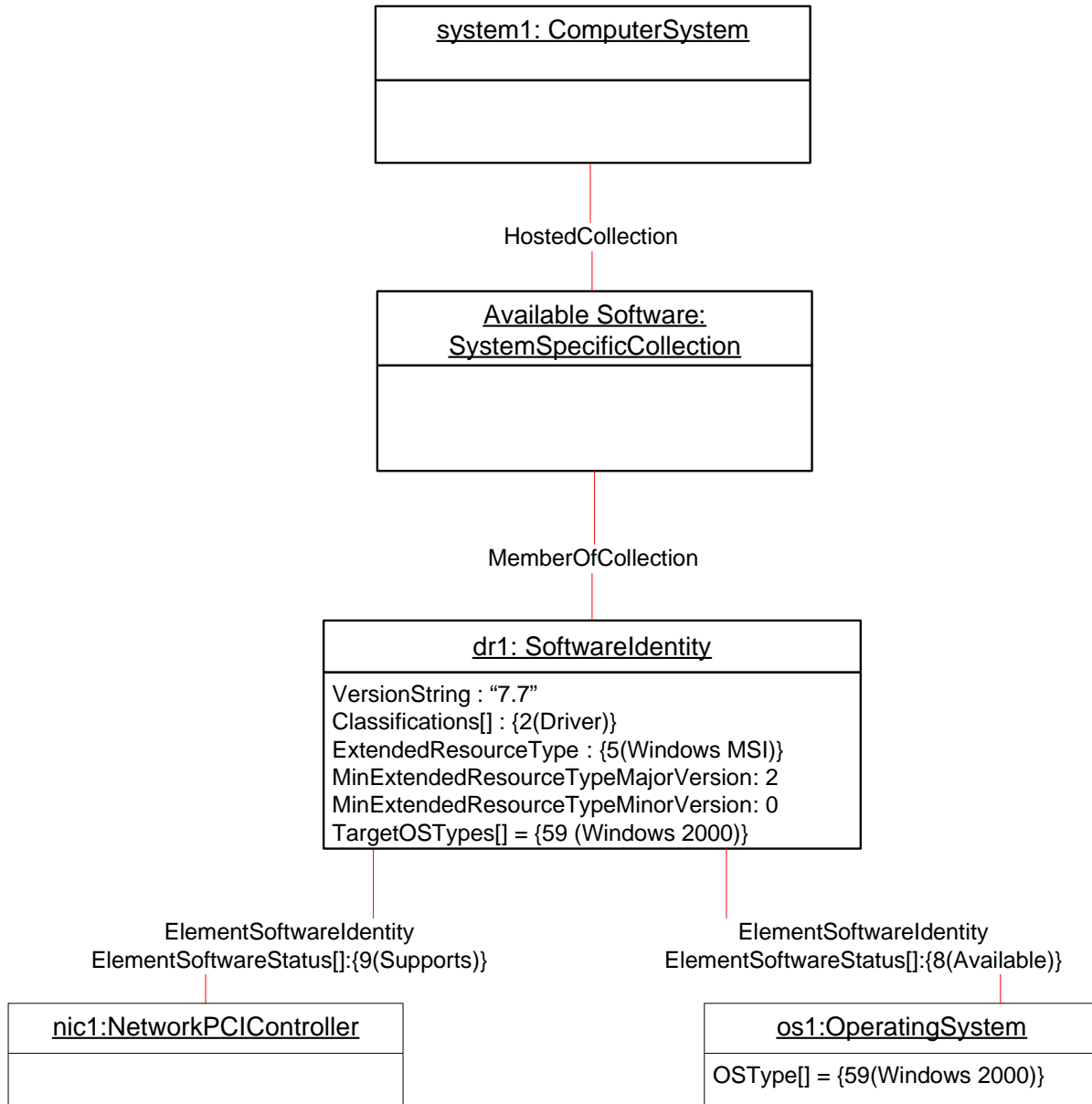
719

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

721 **9.1.9 Representing an Available Driver and Its Relationship to the Operating System**

722 Figure 10 represents a possible instantiation of the *Software Inventory Profile*: an available driver. In this
 723 instantiation, the driver, dr1, is applicable to the NIC, nic1. The ElementSoftwareStatus property of the
 724 CIM_ElementSoftwareIdentity association instance between dr1 and nic1 has the value 9 (Supports),
 725 indicating that dr1 is applicable to nic1. The object diagram also represents the driver’s relationship to the
 726 operating system, os1, with the ElementSoftwareIdentity association instance having the
 727 ElementSoftwareStatus property with the value 8 (Available), indicating that dr1 is applicable to os1 and
 728 is available for installation. The relationship between system1 and os1 is not shown.

729



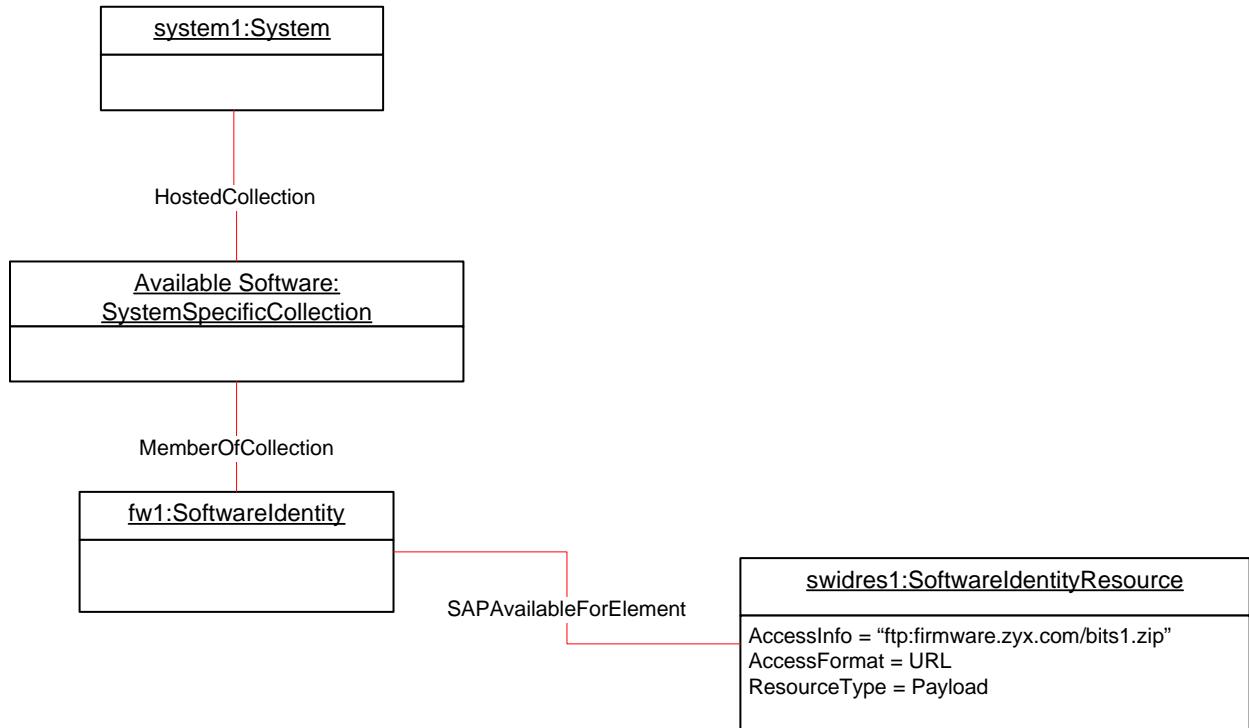
730

731

Figure 10 – Object Diagram Showing an Available Driver

732 **9.1.10 Representing Available Software and Its Location Information**

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



736

737

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

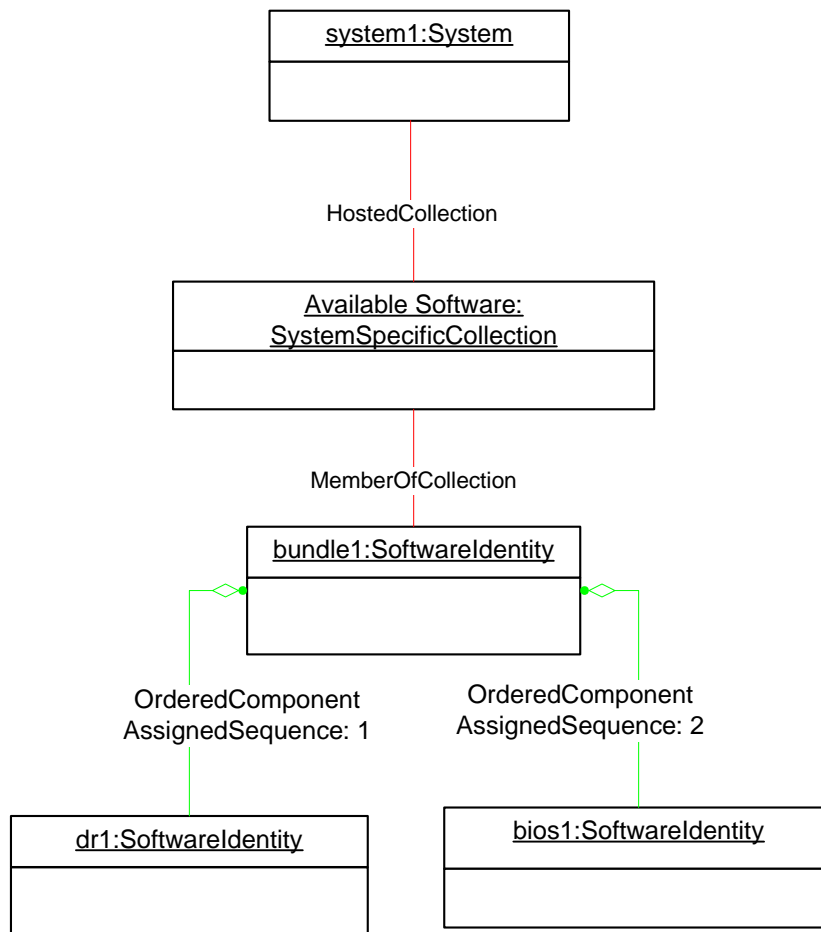
738 **EXPERIMENTAL**

739 **9.1.11 Representing a Software Bundle**

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

- 742 • dr1 with the Assigned sequence of 1, indicating that dr1 will be the first to be installed while installing the bundle
- 743 • bios1 with the Assigned sequence of 2, indicating that bios1 will be the second to be installed while installing the
 744 bundle

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



747

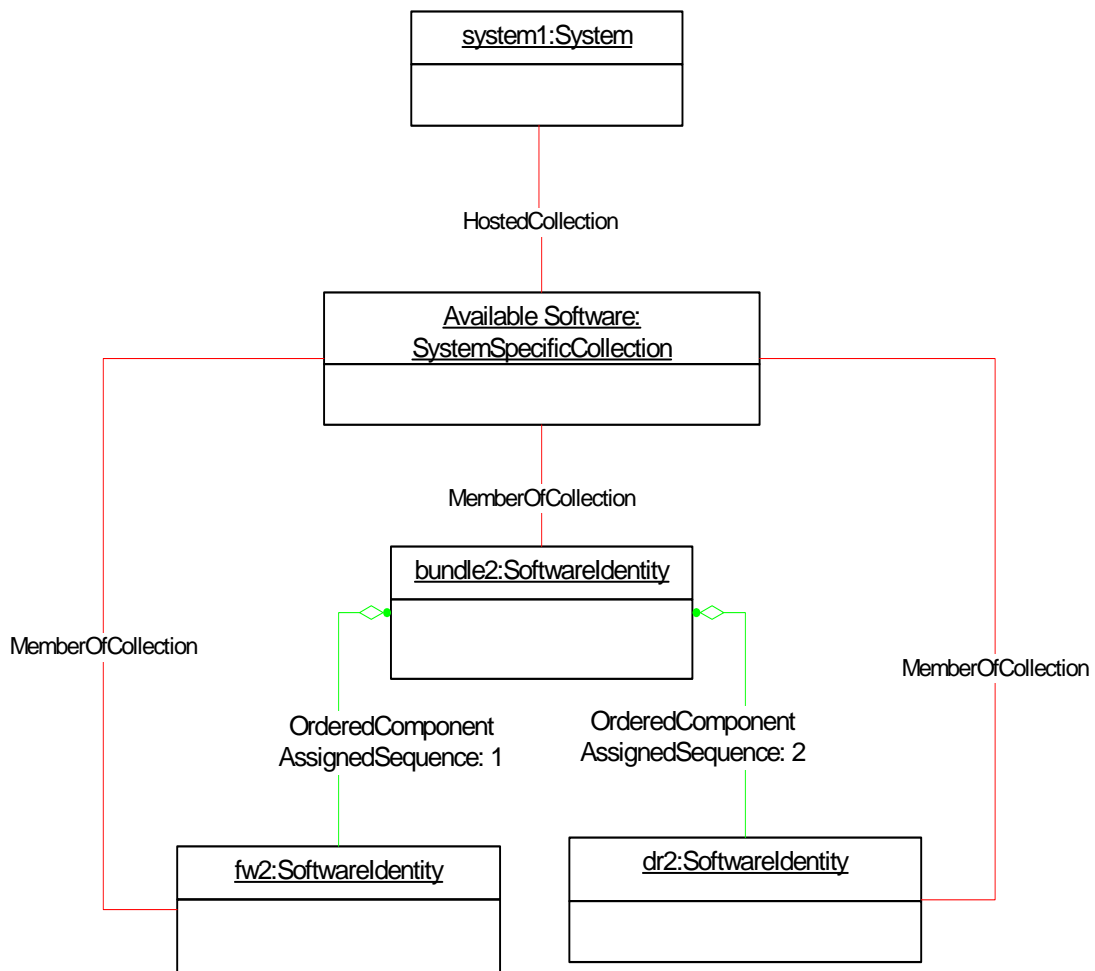
748 **Figure 12 – Object Diagram Showing a Software Bundle**

749 **EXPERIMENTAL**

750 **EXPERIMENTAL**

751 **9.1.12 Representing Software That Is Part of a Software Bundle and Available**

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



756

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

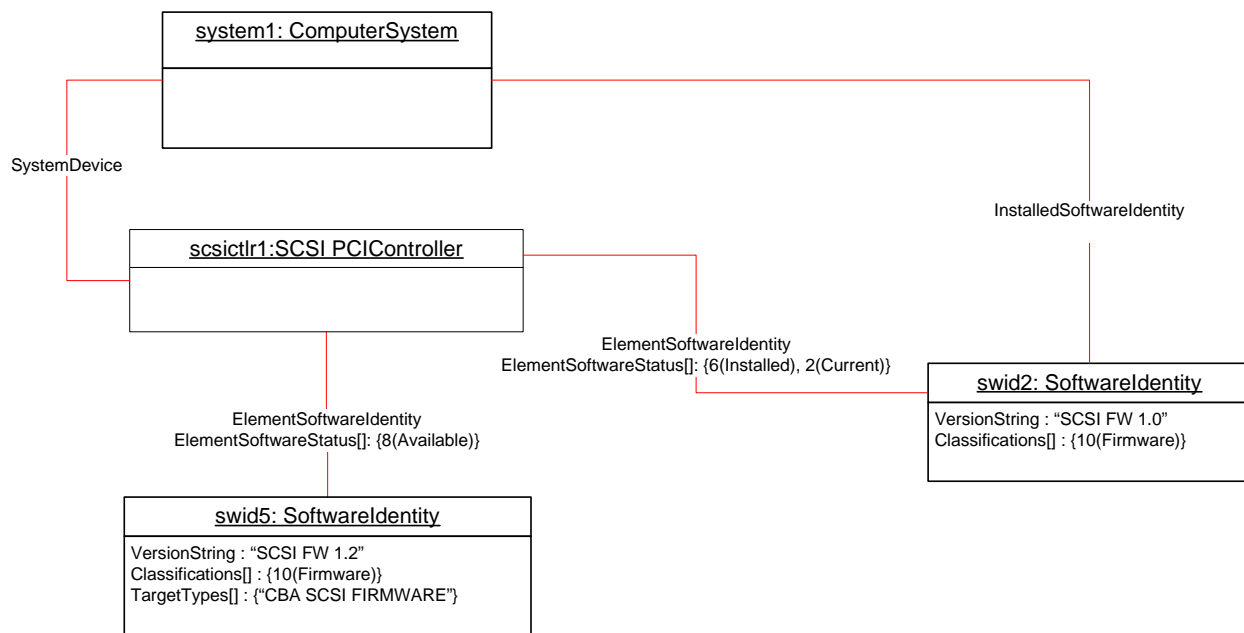
758 **EXPERIMENTAL**

759 **EXPERIMENTAL**760 **9.1.13 Representing Installed and Available Software**

761 Figure 14 represents a possible instantiation of the *Software Inventory Profile*. In the object diagram,
 762 Software Identity swid2 is shown as installed on the SCSI PCI Controller, scsictlr1, and is currently
 763 running on it. The ElementSoftwareStatus property on the CIM_ElementSoftwareIdentity instance
 764 associating swid2 and scsictlr1 has the values 2 (Current) and 6 (Installed).

765 Software Identity swid5 is shown as Available Software for scsictlr1, and so the ElementSoftwareStatus
 766 property on the CIM_ElementSoftwareIdentity instance that associates swid5 and scsictlr1 has the value
 767 8 (Available).

768



769

770 **Figure 14 – Object Diagram Showing Installed and Available Software**771 **EXPERIMENTAL**772 **9.2 Find All the Software Installed on All the Managed Elements within the**
773 **Scope of a Managed System**

774 For the instance of CIM_System that represents the given managed system, select all the instances of
 775 CIM_SoftwareIdentity that are associated through instances of CIM_InstalledSoftwareIdentity. The
 776 resulting instances represent the software installed on all the Managed Elements in the scope of the
 777 managed system.

778 **9.3 Find All the Software Installed on a Managed Element**

779 For the given instance of CIM_ManagedElement, select the instance of CIM_SoftwareIdentity that is
 780 associated through an instance of CIM_ElementSoftwareIdentity such that the ElementSoftwareStatus
 781 property contains the value 6 (Installed).

782 **9.4 Find All the Software That Is Compatible with a Managed Element but Has** 783 **Not Been Installed**

784 For the given instance of CIM_ManagedElement, using the CIM_ElementSoftwareIdentity association,
785 select the associated instances of CIM_SoftwareIdentity that are not associated to the scoping
786 CIM_System or CIM_ComputerSystem instance through the CIM_InstalledSoftwareIdentity association. If
787 the given instance of CIM_ManagedElement does not have any associating
788 CIM_ElementSoftwareIdentity instances referencing it, the compatible software will not be determinable
789 and no instances of CIM_SoftwareIdentity will be returned.

790 **9.5 Find All the Software That Is Available for Installation on Any Managed** 791 **Element within the Scope of a Managed System**

792 For the instance of CIM_ComputerSystem that represents the given managed system, select the instance
793 of CIM_SystemSpecificCollection with ElementName value of "Available Software" that is associated
794 through and instance of CIM_HostedCollection. Select all the instances of CIM_SoftwareIdentity that are
795 associated through an instance of CIM_MemberOfCollection.

796 **9.6 For a Given NIC, Find the Driver That Is Running in the Operating System**

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

- 798 1) For the instance of CIM_ManagedElement that represents the NIC, select the instances of the
799 CIM_ElementSoftwareIdentity association with the ElementSoftwareStatus property containing the
800 value 9 (Supports).
- 801 2) Select the instances of CIM_SoftwareIdentity that the instances of CIM_ElementSoftwareIdentity
802 from step 1 reference.
- 803 3) From the given instance of CIM_ManagedElement that represents the NIC, select the instance of
804 CIM_ComputerSystem that is associated through an instance of CIM_SystemDevice.
- 805 4) From the CIM_ComputerSystem instance from step 3, select the instance of CIM_OperatingSystem
806 that is associated through an instance of CIM_RunningOS.
- 807 5) Select the instances of CIM_ElementSoftwareIdentity that reference the instance of
808 CIM_OperatingSystem from step 4 and contain the value 2 (Current) in the ElementSoftwareStatus
809 property.
- 810 6) Select the instance of CIM_SoftwareIdentity that is referenced by at least one instance of
811 CIM_ElementSoftwareIdentity from step 2 and at least one instance of CIM_ElementSoftwareIdentity
812 from step 5.

813 **9.7 Set a Particular Software Image on a Hardware Managed Element to Run** 814 **After the Next Reset or Reboot**

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

- 817 1) Select the CIM_ElementSoftwareIdentity association instance that associates the Managed Element
818 instance that represents the device with the Software Identity instance that represents the software
819 image.
- 820 2) Set the value of the ElementSoftwareStatus property on the ElementSoftwareIdentity association to
821 3 (Next).

822 **9.8 Set a Particular Software Image on a Hardware Managed Element to Run** 823 **After the Next Reset or Reboot But Not After a Subsequent Reset or Reboot**

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

- 826 1) Select the CIM_ElementSoftwareIdentity association instance that associates the Managed Element
827 instance that represents the device with the Software Identity instance that represents the software
828 image.
- 829 2) Set the value of the ElementSoftwareStatus property on the ElementSoftwareIdentity association to
830 7 (SingleUse).

831 **9.9 Find and Set a Driver to Run After the Next Reset or Reboot for a NIC**

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

- 833 1) For the instance of CIM_ManagedElement that represents the NIC, select the instances of
834 CIM_ElementSoftwareIdentity association with the ElementSoftwareStatus property containing the
835 value 9 (Supports).
- 836 2) Select the instance of CIM_SoftwareIdentity that the instances of CIM_ElementSoftwareIdentity from
837 step 1 reference.
- 838 3) Identify the CIM_SoftwareIdentity instance that corresponds to the driver.
- 839 4) From the given instance of CIM_ManagedElement that represents the NIC, select the instance of
840 CIM_ComputerSystem that is associated through an instance of CIM_SystemDevice.
- 841 5) From the CIM_ComputerSystem instance from step 4, select the instance of CIM_OperatingSystem
842 that is associated through an instance of CIM_RunningOS.
- 843 6) Select the instances of CIM_ElementSoftwareIdentity that reference the instance of
844 CIM_OperatingSystem from step 5 and contain the value 6 (Installed) in the ElementSoftwareStatus
845 property.
- 846 7) Select the instance of CIM_ElementSoftwareIdentity that associates the instance of
847 CIM_ManagedElement and the instance of CIM_SoftwareIdentity from step 3. Set the value of the
848 ElementSoftwareStatus property of this instance to 3 (Next).

849 **9.10 Find the Most Recent Firmware Available for a NIC**

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

- 851 1) For the given instance of CIM_ManagedElement that represents the NIC, select the instances of
852 CIM_SoftwareIdentity that are associated through instances of CIM_ElementSoftwareIdentity with
853 the ElementSoftwareStatus property containing the value 8 (Available) with the Classifications[]
854 property on the CIM_SoftwareIdentity instance containing the value 10 (Firmware).
- 855 2) From the instances returned, select the instance of CIM_SoftwareIdentity with the highest version.
856 (See section 7.10 for the version comparison algorithm.)

857 **9.11 Find the Most Recent Firmware Installed on a NIC**

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

- 859 1) For the given instance of CIM_ManagedElement that represents the NIC, select the instances of
860 CIM_SoftwareIdentity that are associated through instances of CIM_ElementSoftwareIdentity with
861 the ElementSoftwareStatus property containing the value 6 (Installed) with the Classifications[]
862 property on the CIM_SoftwareIdentity instance containing the value 10 (Firmware).

- 863 2) From the instances returned, select the instance of CIM_SoftwareIdentity with the highest version.
 864 (See section 7.10 for the version comparison algorithm.)

865 **9.12 Find the Software Families of Which a Software Identity Is a Member**

866 For the given instance of CIM_SoftwareIdentity, select all the values in the IdentityInfoValue[] property
 867 array that have a value at the corresponding index in the IdentityInfoType[] property array equal to
 868 "CIM:SoftwareFamily". Each of the selected values represents a Software Family of which the Software
 869 Identity is a member.

870 **9.13 Determine Whether a Dependency of a Software Identity Is Satisfied**

871 Given an instance of CIM_SoftwareIdentity that represents an Installation Dependency for a Software
 872 Identity, a client can determine if the dependency is resolved as follows:

- 873 1) From the Scoping Instance, select all the instances of CIM_SoftwareIdentity that are associated
 874 through instances of CIM_InstalledSoftwareIdentity.
- 875 2) For each Software Identity from step 1, determine all the Software Families to which it belongs by
 876 using the algorithm in section 9.12.
- 877 3) For the instance of CIM_SoftwareIdentity that represents the dependency, determine the Software
 878 Families by using the algorithm in section 9.12.
- 879 4) Select the instance of CIM_SoftwareIdentity from step 1 such that at least one Software Family to
 880 which it belongs (from step 2) is equal to at least one Software Family to which the dependency
 881 belongs (from step 3).

882 The dependency is satisfied if the version of the selected Software Identity is greater than or equal to the
 883 version of the dependency represented by an instance of CIM_SoftwareIdentity. (See section 7.10 for the
 884 version comparison algorithm.)

885 **10 CIM Elements**

886 Table 11 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
 887 implemented as described in Table 11. Sections 7 ("Implementation") and 8 ("Methods") may impose
 888 additional requirements on these elements.

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

Element Name	Requirement	Description
Classes		
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. EXPERIMENTAL
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. EXPERIMENTAL

Element Name	Requirement	Description
CIM_OrderedDependency	Optional	See sections 7.9.1 and 10.11. EXPERIMENTAL
CIM_RegisteredProfile	Mandatory	See section 10.12.
Indications		
None defined in this profile		

890 10.1 CIM_SoftwareIdentity

891 CIM_SoftwareIdentity is used to represent either Installed Software or Available Software. Table 12
892 contains the requirements for elements of this class.

893 **Table 12 – Class: CIM_SoftwareIdentity**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
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.
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.

894 10.2 CIM_InstalledSoftwareIdentity

895 CIM_InstalledSoftwareIdentity is used to associate an instance of CIM_System and an instance of
896 CIM_SoftwareIdentity. CIM_InstalledSoftwareIdentity is conditional and shall be implemented when
897 Installed Software is modeled. Table 13 contains the requirements for elements of this class.

898 **Table 13 – Class: CIM_InstalledSoftwareIdentity**

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

899 **10.3 CIM_ElementSoftwareIdentity**

900 CIM_ElementSoftwareIdentity is used to associate an instance of CIM_ManagedElement and an instance
 901 of CIM_SoftwareIdentity when the instance of CIM_ManagedElement is instrumented. Table 14 contains
 902 the requirements for elements of this class.

903 **Table 14 – Class: CIM_ElementSoftwareIdentity**

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

904 **10.4 CIM_SystemSpecificCollection**

905 CIM_SystemSpecificCollection is used to represent a collection of Available Software. Table 15 contains
 906 the requirements for elements of this class.

907 **Table 15 – Class: CIM_SystemSpecificCollection**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
ElementName	Mandatory	See section 7.6.1.

908 **10.5 CIM_HostedCollection**

909 CIM_HostedCollection is used to associate CIM_System and CIM_SystemSpecificCollection.
 910 CIM_HostedCollection is conditional and shall be implemented when an instance of
 911 CIM_SystemSpecificCollection is instrumented. Table 16 contains the requirements for elements of this
 912 class.

913 **Table 16 – Class: CIM_HostedCollection**

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

914 **10.6 CIM_MemberOfCollection**

915 CIM_MemberOfCollection is used to associate an instance of CIM_SystemSpecificCollection and an
 916 instance of CIM_SoftwareIdentity. CIM_MemberOfCollection is conditional and shall be implemented
 917 when an instance of CIM_SystemSpecificCollection is instrumented. Table 17 contains the requirements
 918 for elements of this class.

919 **Table 17 – Class: CIM_MemberOfCollection**

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

920 **EXPERIMENTAL**921 **10.7 CIM_SoftwareIdentityResource**

922 CIM_SoftwareIdentityResource is used to represent the location of a Software Identity, which could be
 923 used as input to the software installation service (see the *Software Update Profile*). Table 18 contains the
 924 requirements for elements of this class.

925 **Table 18 – Class: CIM_SoftwareIdentityResource**

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
Name	Mandatory	Key
InfoFormat	Mandatory	
AccessInfo	Mandatory	
ResourceType	Optional	

926 **EXPERIMENTAL**

927 **10.8 CIM_SAPAvailableForElement**

928 CIM_SAPAvailableForElement is used to associate CIM_SoftwareIdentityResource and
 929 CIM_SoftwareIdentity. CIM_SAPAvailableForElement is conditional and shall be implemented when the
 930 location information of CIM_SoftwareIdentity is represented. Table 19 contains the requirements for
 931 elements of this class.

932 **Table 19 – Class: CIM_SAPAvailableForElement**

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

933 **10.9 CIM_HostedAccessPoint**

934 CIM_HostedAccessPoint is used to associate CIM_System and CIM_SoftwareIdentityResource when an
 935 instance of CIM_SoftwareIdentityResource is instrumented. Table 20 contains the requirements for
 936 elements of this class.

937 **Table 20 – Class: CIM_HostedAccessPoint**

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

938 **EXPERIMENTAL**

939 **10.10 CIM_OrderedComponent**

940 CIM_OrderedComponent is used to associate an instance of CIM_SoftwareIdentity that represents a
 941 Software Bundle and an instance of CIM_SoftwareIdentity that represents one of the discrete software
 942 images contained in the Software Bundle. Table 21 contains the requirements for elements of this class.

943 **Table 21 – Class: CIM_OrderedComponent**

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

944 **EXPERIMENTAL**

945 **EXPERIMENTAL**

946 **10.11 CIM_OrderedDependency**

947 CIM_OrderedDependency is used to associate an instance of CIM_SoftwareIdentity that represents an
 948 Installation Dependency and an instance of CIM_SoftwareIdentity for which the Installation Dependencies
 949 are represented. Table 22 contains the requirements for elements of this class.

950 **Table 22 – Class: CIM_OrderedDependency**

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

951 **EXPERIMENTAL**

952 **10.12 CIM_RegisteredProfile**

953 The CIM_RegisteredProfile class is defined by the *Profile Registration Profile*. The requirements denoted
 954 in Table 23 are in addition to those mandated by the *Profile Registration Profile*.

955 **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).

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

959
960
961
962**ANNEX A
(informative)****Change Log**

Version	Date	Description
1.0.0a	2006/10/19	Preliminary Standard version.
1.0.0b	2007/03/12	Preliminary Standard refresh. Minor errata incorporation.
1.0.0	2007/11/21	Final Standard
1.0.0	2008/07/21	Section 7.7, 7.9, 9.1.11, 9.1.12, 9.1.13, 10.7. 10.10, 10.11 marked EXPERIMENTAL in order to go FINAL
1.0.0	2008/07/28	Address David Hines ballot comments http://www.dmtf.org/apps/org/workgroup/platformprofiles/download.php/39476/DSP1023_comments_Hines_2008-07-28.xls

963

964
965
966
967

ANNEX B (informative)

Acknowledgments

968 The authors wish to acknowledge the following people.

969 **Editor:**

- 970 • RadhaKrishna R. Dasari – Dell, Inc.**Contributors:**
971 • RadhaKrishna R. Dasari – Dell, Inc.Jon Hass – Dell, Inc.
972 • Khachatur Papanyan – Dell Inc.
973 • Aaron Merkin – IBM
974 • Jeff Hilland – Hewlett-Packard Corporation
975 • Christina Shaw – Hewlett-Packard Corporation
976 • Michael Tehranian – Sun Microsystems
977 • Perry G. Vincent – Intel Corporation
978 • John Leung – Intel Corporation