



1
2
3
4

Document Number: DSP1025

Date: 2009-06-16

Version: 1.0.0

5 **Software Update Profile**

6 **Document Type: Specification**
7 **Document Status: DMTF Standard**
8 **Document Language: E**
9

10 Copyright Notice

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

12 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
13 management and interoperability. Members and non-members may reproduce DMTF specifications and
14 documents, provided that correct attribution is given. As DMTF specifications may be revised from time to
15 time, the particular version and release date should always be 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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.

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

CONTENTS

33	Foreword	5
34	Introduction	6
35	1 Scope	7
36	2 Normative References.....	7
37	2.1 Approved References	7
38	2.2 Other References.....	7
39	3 Terms and Definitions.....	7
40	4 Symbols and Abbreviated Terms.....	9
41	5 Synopsis	9
42	6 Description	9
43	7 Implementation Requirements	10
44	7.1 CIM_SoftwareInstallationService Instance	11
45	7.2 CIM_SoftwareInstallationServiceCapabilities Instance	11
46	7.3 Advertising Compatibility with a Software Identity (Optional)	11
47	7.4 Representing Relationship between Managed Element and Software Installation Service.....	12
48	7.5 Advertising the Location Information of a Software Identity (Optional)	12
49	7.6 Version Comparison Algorithm	12
50	8 Methods.....	13
51	8.1 CIM_SoftwareInstallationService.CheckSoftwareIdentity()	13
52	8.2 CIM_SoftwareInstallationService.InstallFromSoftwareIdentity()	14
53	8.3 CIM_SoftwareInstallationService.InstallFromByteStream().....	16
54	8.4 CIM_SoftwareInstallationService.InstallFromURI()	18
55	8.5 Profile Conventions for Operations.....	19
56	8.6 CIM_SoftwareInstallationService Operations	20
57	8.7 CIM_HostedService Operations	20
58	8.8 CIM_SoftwareInstallationServiceCapabilities Operations	20
59	8.9 CIM_ElementCapabilities Operations.....	20
60	8.10 CIM_ServiceAffectsElement	21
61	9 Use Cases	21
62	9.1 Object Diagrams	21
63	9.2 Find the Software Installation Services compatible with a Software Identity.....	27
64	9.3 Determine Whether Installing a Software Identity Requires a Reboot	27
65	9.4 Find Software Available for Installation on a Managed Element when	
66	CIM_ElementSoftwareIdentity Exists.....	28
67	9.5 Find Software Available for Installation on a Managed Element when	
68	CIM_ElementSoftwareIdentity Does Not Exist	28
69	9.6 Find Software Available for Installation on a Component.....	28
70	9.7 Find Software Installation Services that Can Install or Update Software on a Managed	
71	Element.....	28
72	9.8 Install or Update Software on a Managed Element Using Software Identity.....	29
73	9.9 Install from Software Identity when the Managed Element is not modeled.....	29
74	9.10 Install or Update a Software on a Managed Element Using a URI	30
75	9.11 Install from URI When the Managed Element Is Not Modeled.....	30
76	9.12 Update Software on a Managed Element Using a Byte Stream	30
77	10 CIM Elements	31
78	10.1 CIM_HostedService	31
79	10.2 CIM_SoftwareInstallationService.....	31
80	10.3 CIM_ElementCapabilities	32
81	10.4 CIM_SoftwareInstallationCapabilities	32
82	10.5 CIM_ServiceAffectsElement – CIM_SoftwareIdentity Reference.....	33
83	10.6 CIM_ServiceAffectsElement – CIM_ManagedElement Reference	33

84	10.7	CIM_SoftwareIdentity.....	33
85	10.8	CIM_RegisteredProfile.....	34
86	ANNEX A (Informative)	Change Log.....	35
87			

88 Figures

89	Figure 1 – Class Diagram: Software Update Profile	10
90	Figure 2 – Registered Profile	21
91	Figure 3 – Software Update Profile: Object Diagram.....	22
92	Figure 4 – Software Update Profile: Object Diagram.....	23
93	Figure 5 – Software Update Profile: Object Diagram.....	24
94	Figure 6 – Software Update Profile: Object Diagram.....	25
95	Figure 7 – Software Update Profile: Object Diagram.....	25
96	Figure 8 – Software Update Profile: Object Diagram.....	26
97	Figure 9 – Software Update Profile: Object Diagram.....	27
98		

99 Tables

100	Table 1 – Referenced Profiles	9
101	Table 2 – CIM_SoftwareInstallationService.CheckSoftwareIdentity() Method: Return Code Values.....	13
102	Table 3 – CIM_SoftwareInstallationService.CheckSoftwareIdentity() Method: Parameters	13
103	Table 4 – CIM_SoftwareInstallationService.InstallFromSoftwareIdentity() Method: Return Code Values	14
104	Table 5 – CIM_SoftwareInstallationService.InstallFromSoftwareIdentity() Method: Parameters.....	15
105	Table 6 – CIM_SoftwareInstallationService.InstallFromByteStream() Method: Return Code Values.....	17
106	Table 7 – CIM_SoftwareInstallationService.InstallFromByteStream() Method: Parameters.....	17
107	Table 8 – CIM_SoftwareInstallationService.InstallFromURI() Method: Return Code Values.....	18
108	Table 9 – CIM_SoftwareInstallationService.InstallFromURI() Method: Parameters.....	18
109	Table 10 – Operations: CIM_HostedService	20
110	Table 11 – CIM_ElementCapabilities Operations	20
111	Table 12 – CIM_ServiceAffectsElement Operations	21
112	Table 13 – CIM Elements: Software Update Profile	31
113	Table 14 – Class: CIM_HostedService	31
114	Table 15 – Class: CIM_SoftwareInstallationService	32
115	Table 16 – Class: CIM_ElementCapabilities.....	32
116	Table 17 – Class: CIM_SoftwareInstallationCapabilities	32
117	Table 18 – Class: CIM_ServiceAffectsElement	33
118	Table 19 – Class: CIM_ServiceAffectsElement	33
119	Table 20 – Class: CIM_SoftwareIdentity.....	33
120	Table 21 – Class: CIM_RegisteredProfile.....	34
121		

122

Foreword

123 The *Software Update Profile* (DSP1025) was prepared by the Server Management Working Group and
124 the Physical Platform Profiles Working Group of the DMTF.

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

127 **Acknowledgments**

128 The authors wish to acknowledge the following people.

129 **Editor:**

- 130 • RadhaKrishna R. Dasari – Dell

131 **Contributors:**

- 132 • RadhaKrishna R. Dasari – Dell
- 133 • Jon Hass – Dell
- 134 • Khachatur Papanyan – Dell
- 135 • Marshal Savage – Dell
- 136 • Sudhir Shetty - Dell
- 137 • Jeff Hilland – HP
- 138 • Christina Shaw – HP
- 139 • Aaron Merkin – IBM
- 140 • Jeff Lynch – IBM
- 141 • Perry Vincent – Intel
- 142 • John Leung – Intel

143

144

Introduction

145 The information in this specification should be sufficient for a provider or consumer of this data to identify
146 unambiguously the classes, properties, methods, and values that must be instantiated and manipulated to
147 support the installation and update of BIOS, Firmware, Drivers and related software on a managed
148 element within a managed system, using the DMTF Common Information Model (CIM) core and extended
149 model definitions.

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

152

153

Software Update Profile

154 1 Scope

155 The *Software Update Profile* describes the classes, associations, properties, and methods used to
156 support the installation and update of BIOS, firmware, drivers and related software on a managed
157 element within a managed system.

158 2 Normative References

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

162 2.1 Approved References

163 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,
164 http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf

165 DMTF DSP0200, *CIM Operations over HTTP 1.2*,
166 http://www.dmtf.org/standards/published_documents/DSP0200_1.2.pdf

167 DMTF DSP0215, *Server Management Managed Element Addressing Specification 1.0*,
168 http://www.dmtf.org/standards/published_documents/DSP0215_1.0.pdf

169 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
170 http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf

171 DMTF DSP1023, *Software Inventory Profile 1.0*,
172 http://www.dmtf.org/standards/published_documents/DSP1023_1.0.pdf

173 DMTF DSP1033, *Profile Registration Profile 1.0*,
174 http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf

175 2.2 Other References

176 IETF RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax*, <http://www.ietf.org/rfc/rfc2396.txt>

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

179 3 Terms and Definitions

180 For the purposes of this document, the following terms and definitions apply. For the purposes of this
181 document, the terms and definitions given in [Software Inventory Profile](#) also apply.

182 3.1

183 can

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

- 185 **3.2**
186 **cannot**
187 used for statements of possibility and capability, whether material, physical or causal
- 188 **3.3**
189 **conditional**
190 indicates requirements to be followed strictly in order to conform to the document when the specified
191 conditions are met
- 192 **3.4**
193 **mandatory**
194 indicates requirements to be followed strictly in order to conform to the document and from which no
195 deviation is permitted
- 196 **3.5**
197 **may**
198 indicates a course of action permissible within the limits of the document
- 199 **3.6**
200 **need not**
201 indicates a course of action permissible within the limits of the document
- 202 **3.7**
203 **optional**
204 indicates a course of action permissible within the limits of the document
- 205 **3.8**
206 **referencing profile**
207 indicates a profile that owns the definition of this class and can include a reference to this profile in its
208 "Related Profiles" table
- 209 **3.9**
210 **shall**
211 indicates requirements to be followed strictly in order to conform to the document and from which no
212 deviation is permitted
- 213 **3.10**
214 **shall not**
215 indicates requirements to be followed strictly in order to conform to the document and from which no
216 deviation is permitted
- 217 **3.11**
218 **should**
219 indicates that among several possibilities, one is recommended as particularly suitable, without
220 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 221 **3.12**
222 **should not**
223 indicates that a certain possibility or course of action is deprecated but not prohibited
- 224 **3.13**
225 **unspecified**
226 indicates that this profile does not define any constraints for the referenced CIM element or operation

227 **3.14**228 **Software Installation Service**

229 A component that can be used to perform an installation or update of software on a managed element

230 **4 Symbols and Abbreviated Terms**

231 None

232 **5 Synopsis**233 **Profile Name:** Software Update234 **Version:** 1.0.0235 **Organization:** DMTF236 **CIM Schema Version:** 2.22237 **Central Class:** CIM_SoftwareInstallationService238 **Scoping Class:** CIM_System

239 The *Software Update Profile* describes the classes and properties used to support the installation and
 240 update of BIOS, firmware, drivers and related software on a managed element within a managed system.

241 CIM_SoftwareInstallationService shall be the Central Class of this profile. An instance of
 242 CIM_SoftwareInstallationService shall be the Central Instance of this profile.

243 CIM_System shall be the Scoping Class of this profile. The instance of CIM_System shall be the Scoping
 244 Instance of this profile.

245 References to CIM_System may be interpreted as references to subclasses of CIM_System such as
 246 CIM_ComputerSystem.

247 Table 1 lists profiles upon which this profile has a dependency.

248

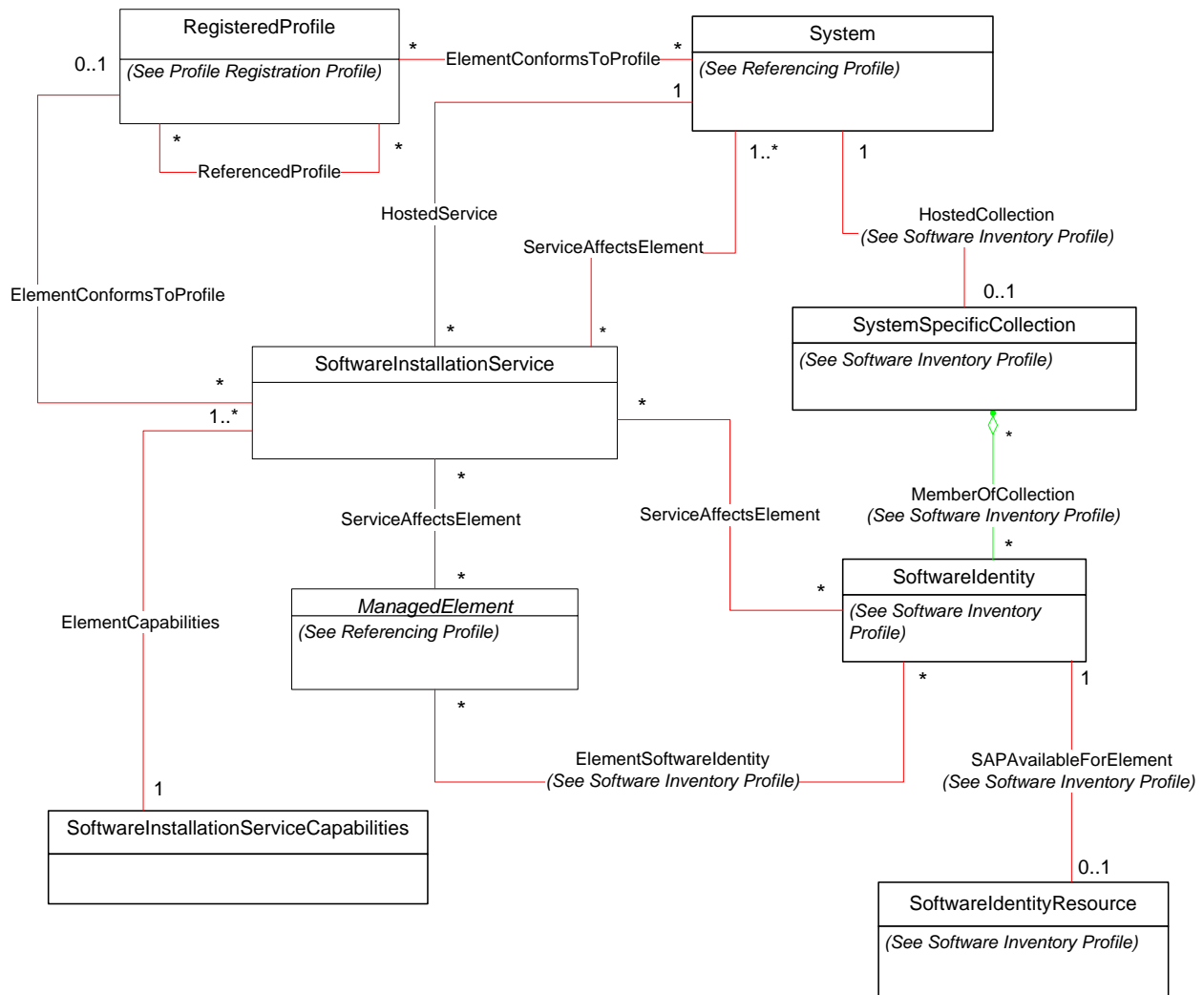
Table 1 – Referenced Profiles

Profile Name	Organization	Version	Description
Profile Registration	DMTF	1.0	Mandatory
Software Inventory	DMTF	1.0	Optional

249 **6 Description**

250 The *Software Update Profile* provides the ability to perform installation or update of software on Managed
 251 Elements in the scope of a managed system. The profile also defines relationship between a managed
 252 element and the installation service that represents the availability of software installation and update
 253 functionality for a managed element.

254 Figure 1 represents the class schema of the *Software Update Profile* and shows the elements of the
 255 profile along with the dependent relationships between the elements of the profile and the referencing
 256 profiles. For simplicity, the prefix CIM_ has been removed from the name of the classes.



257

258

Figure 1 – Class Diagram: Software Update Profile

259 This profile supports installation and update of software on a managed element. Installation of software
 260 implies the first time installation of the software on the managed element and update of software implies
 261 that the managed element has a version of the software already installed on it.

262 The CIM_SoftwareInstallationService provides the ability to perform installation or update of software.
 263 The CIM_SoftwareInstallationServiceCapabilities define the capabilities of
 264 CIM_SoftwareInstallationService such as the list of the methods supported, the types of software that it is
 265 capable of installing and the supported installation options such as install, update, repair, forced
 266 installation and silent mode installation.

267 7 Implementation Requirements

268 This section describes the implementation requirements of the *Software Update Profile*. The list of all
 269 required methods can be found in Section 8 and properties in Section 9.2.

270 **7.1 CIM_SoftwareInstallationService Instance**

271 Each Software Installation Service shall be represented using exactly one instance of
272 CIM_SoftwareInstallationService. An instance of CIM_SoftwareInstallationService shall support at least
273 one of InstallFromSoftwareIdentity(), InstallFromByteStream() or InstallFromURI() methods.

274 **7.2 CIM_SoftwareInstallationServiceCapabilities Instance**

275 The capabilities of a Software Installation Service shall be represented by an instance of
276 CIM_SoftwareInstallationServiceCapabilities. Each instance of CIM_SoftwareInstallationService shall be
277 associated with exactly one instance of CIM_SoftwareInstallationServiceCapabilities through
278 CIM_ElementCapabilities association. An instance of CIM_SoftwareInstallationServiceCapabilities may
279 be associated with one or more instances of CIM_SoftwareInstallationService through
280 CIM_ElementCapabilities association.

281 **7.2.1 CIM_SoftwareInstallationServiceCapabilities.SupportedURISchemes[]**

282 When the SupportedAsynchronousActions property or SupportedSynchronousActions property contains
283 the value 5 (Install From URI), this property shall list the URI schemes that are supported by the
284 associated instance of CIM_SoftwareInstallationService.

285 **7.3 Advertising Compatibility with a Software Identity (Optional)**

286 The following sections describe mechanisms to advertise compatibility between a Software Identity and
287 an instance of CIM_SoftwareInstallationService that can install or update the Software Identity. The
288 behavior described in each of the following sections is optional and should be implemented.

289 **7.3.1 Using Target Types**

290 The CIM_SoftwareIdentity.TargetTypes array property shall contain one or more strings that are used to
291 advertise the compatibility with a Software Installation Service.

292 The CIM_SoftwareInstallationService.SupportedTargetTypes array property shall contain one or more
293 strings that are used to advertise the compatibility with a Software Identity.

294 An instance of CIM_SoftwareInstallationService that is compatible to a Software Identity shall have at
295 least one of the values in the SupportedTargetTypes property of the associated instance of
296 CIM_SoftwareInstallationServiceCapabilities equal to at least one of the values in the TargetTypes array
297 property of the Software Identity.

298 **7.3.2 Using ExtendedResourceType**

299 The CIM_SoftwareIdentity.ExtendedResourceType property shall represent a single format for an installer
300 that is capable of installing or updating the Software Identity. The minimum version of the installer format
301 required for compatibility shall be represented using the MinExtendedResourceTypeMajorVersion,
302 MinExtendedResourceTypeMinorVersion, MinExtendedResourceTypeRevisionNumber,
303 MinExtendedResourceTypeBuildNumber properties of the Software Identity.

304 The installer formats supported by the instance of CIM_SoftwareInstallationService shall be represented
305 using the SupportedExtendedResourceTypes array property of the associated
306 CIM_SoftwareInstallationServiceCapabilities instance. For each installer format, the supported versions
307 shall be represented using the SupportedExtendedResourceTypesMajorVersions,
308 SupportedExtendedResourceTypesMinorVersions,
309 SupportedExtendedResourceTypesRevisionNumbers, SupportedExtendedResourceTypesBuildNumbers
310 array properties of the associated CIM_SoftwareInstallationServiceCapabilities instance at the
311 corresponding index.

312 An instance of CIM_SoftwareInstallationService that is compatible to a Software Identity shall have at
313 least one of the values in the SupportedExtendedResourceTypes property of the associated instance of
314 CIM_SoftwareInstallationServiceCapabilities equal to the ExtendedResourceType property of the
315 Software Identity and the version of the installer format supported by the instance of
316 CIM_SoftwareInstallationService shall be equal to or higher than the minimum version of the installer
317 format required by the Software Identity. The version comparison algorithm is described in section 7.6.

318 **7.3.3 CIM_ServiceAffectsElement**

319 When an instance of CIM_SoftwareInstallationService is compatible with a Software Identity that is
320 available for installation, there shall be an instance of CIM_ServiceAffectsElement that associates the
321 CIM_SoftwareInstallationService instance with the Software Identity.

322 **7.4 Representing Relationship between Managed Element and Software** 323 **Installation Service**

324 When an instance of CIM_SoftwareInstallationService is capable of installing or updating software on a
325 Managed Element, there may be an instance of CIM_ServiceAffectsElement that associates the
326 CIM_SoftwareInstallationService with the CIM_ManagedElement instance. When an instance of
327 CIM_SoftwareInstallationService is capable of installing or updating software on an instance of
328 CIM_ComputerSystem or a ManagedElement scoped to the CIM_ComputerSystem instance, there shall
329 be an instance of CIM_ServiceAffectsElement that associates the CIM_SoftwareInstallationService with
330 the CIM_ComputerSystem instance.

331 **7.5 Advertising the Location Information of a Software Identity (Optional)**

332 The location of a Software Identity may be advertised. This is optional behavior. When this optional
333 behavior is implemented, it shall be done according to the Implementation Requirements of the [Software](#)
334 [Inventory Profile](#).

335 **7.6 Version Comparison Algorithm**

336 The following algorithm shall be used to compare the minimum version of the installer format supported
337 by a Software Identity with the installer format version supported by an instance of
338 CIM_SoftwareInstallationService when the version information is represented as major version, minor
339 version, revision number, and build number components using separate properties.

340 When comparing two properties in each step described below, if only one of the properties is null then the
341 instance which has a non-null property shall be the instance with higher version. When both properties
342 are null, the two instances shall be considered as having equal value.

343 1) If the properties representing the major version of the two instances are equal, go to step 2.

344 Else the instance with the higher value of the property representing the major version shall be
345 the instance with higher version.

346 2) If the properties representing the minor version of the two instances are equal, go to step 3.

347 Else the instance with the higher value of the property representing the minor version shall be
348 the instance with higher version.

349 3) If the properties representing the revision number of the two instances are equal, go to step 4.

350 Else the instance with the higher value of the property representing the revision number shall be
351 the instance with higher version.

352 4) If the properties representing the build number of the two instances are equal then the two
 353 instances shall have equal version.

354 Else the instance with the higher value of the property representing the BuildNumber property
 355 shall be the instance with higher version.

356 8 Methods

357 This section details the requirements for supporting intrinsic operations and extrinsic methods for the CIM
 358 elements defined by this profile.

359 8.1 CIM_SoftwareInstallationService.CheckSoftwareIdentity()

360 The CIM_SoftwareInstallationService.CheckSoftwareIdentity() method allows a client application to
 361 determine whether a Software Identity can be installed or updated on a Managed Element. It also allows
 362 the client to determine some other characteristics of the installation, such as whether install will require a
 363 reboot. When the Target parameter and the Collection parameter are both non-NULL, the method shall
 364 return 2 (Error Occurred). When the Target parameter and the Collection parameters are NULL, the
 365 method shall return 2 (Error Occurred).

366 No standard messages are defined.

367 **Table 2 – CIM_SoftwareInstallationService.CheckSoftwareIdentity() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.

368 **Table 3 – CIM_SoftwareInstallationService.CheckSoftwareIdentity() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	Source	CIM_SoftwareIdentity REF	See 8.1.1.
IN	Target	CIM_ManagedElement REF	See 8.1.2.
IN	Collection	CIM_Collection REF	See 8.1.3.
OUT	InstallCharacteristics	uint16[]	An array describing the characteristics of installation/update of the Software Identity on the Managed Element

369 8.1.1 Source

370 The Source parameter is a reference to the Software Identity that represents the software to be checked
 371 for installation or update on a Managed Element. The method shall return 2 (Error Occurred) when this
 372 parameter is NULL.

373 8.1.2 Target

374 The Target parameter is a reference to the instance of CIM_ManagedElement that represents a managed
 375 element on which the Software Identity is intended to be installed or updated. When the Software Identity
 376 cannot be installed on the managed element represented by this parameter, the method shall return
 377 2 (Error Occurred).

378 When this parameter is non-NULL and the method can determine that the Software Identity can be
 379 installed on the Managed Element represented by the Target parameter, the method shall return 0. When
 380 this parameter is non-NULL and the method can determine that the Software Identity cannot be installed
 381 on the Managed Element represented by the Target parameter, the method shall return 2 (Error
 382 Occurred).

383 8.1.3 Collection

384 The Collection parameter is a reference to the instance of CIM_SystemSpecificCollection that represents
 385 the collection to which the Software Identity will be added. When this parameter is not NULL and the
 386 CanAddToCollection property of the associated instance of CIM_SoftwareInstallationServiceCapabilities
 387 is FALSE, the method shall return 2 (Error Occurred).

388 When this parameter is non-NULL and the method can determine that the Software Identity can be added
 389 to the collection, the method shall return 0. When this parameter is non-NULL and the method can
 390 determine that the Software Identity cannot be added to the collection, the method shall return 2 (Error
 391 Occurred).

392 When this parameter is a reference to a collection whose Scoping Instance does not have a
 393 CIM_ServiceAffectsElement association to the CIM_SoftwareInstallationService upon which the method
 394 was invoked, the method shall return 2 (Error Occurred).

395 When this parameter is not a reference to an instance of CIM_SystemSpecificCollection implemented as
 396 defined in the [Software Inventory Profile](#), the method shall return 2 (Error Occurred).

397 8.2 CIM_SoftwareInstallationService.InstallFromSoftwareIdentity()

398 The CIM_SoftwareInstallationService.InstallFromSoftwareIdentity() method allows a client application to
 399 install or update a Software Identity on a Managed Element and provides some installation options for the
 400 client to control the installation procedure. When this method is supported, at least one of
 401 SupportedAsynchronousActions property or SupportedSynchronousActions property of the associated
 402 instance of CIM_SoftwareInstallationServiceCapabilities shall contain the value 3 (Install From Software
 403 Identity).

404 When the method is used to install or update a software for which Installation Dependencies are
 405 advertised and the Dependencies are not satisfied, the method shall return 2 (Error Occurred).

406 When the Target and the Collection parameters are both non-NULL, the method shall return 2. When the
 407 Target and the Collection parameters are NULL, the method shall return 2 (Error Occurred).

408 When the Target parameter is non-NULL and the Collection parameter is NULL, the method will install or
 409 update the Software Identity on the Managed Element. When the Collection parameter is non-NULL and
 410 the Target parameter is NULL, the method will add the Software Identity to the collection.

411 No standard messages are defined.

412 **Table 4 – CIM_SoftwareInstallationService.InstallFromSoftwareIdentity() Method: Return Code**
 413 **Values**

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob.

414 **Table 5 – CIM_SoftwareInstallationService.InstallFromSoftwareIdentity() Method: Parameters**

Qualifiers	Name	Type	Description/Values
OUT	Job	CIM_ConcreteJob REF	See section 8.2.1.
IN	InstallOptions	uint16[]	See section 8.2.2.
IN	InstallOptionsValues	string[]	See section 8.2.3
IN	Source	CIM_SoftwareIdentity REF	See section 8.2.4.
IN	Target	CIM_ManagedElement REF	See section 8.2.5.
IN	Collection	CIM_Collection REF	See section 8.2.6.

415 **8.2.1 Job**

416 The Job parameter is a reference to the instance of CIM_ConcreteJob that represents the job or task that
417 may be started by the invocation of the InstallFromSoftwareIdentity() method.

418 The method shall not return the Job output parameter when SupportedAsynchronousActions property of
419 the associated instance of CIM_SoftwareInstallationServiceCapabilities does not contain the value 3
420 (Install From Software Identity).

421 The method may return the Job output parameter and a return code value of 4096 when the parameters
422 for the method have been validated and a job has been spawned to complete the installation/update.

423 **8.2.2 InstallOptions**

424 The InstallOptions array parameter is used to input the desired installation options to the
425 InstallFromSoftwareIdentity() method allowing the client to control the installation procedure. When this
426 parameter is NULL, the installation options used are implementation specific. The method shall return
427 2 (Error Occurred) when this parameter contains an installation option that is not listed in the
428 SupportedInstallOptions property of the associated instance of
429 CIM_SoftwareInstallationServiceCapabilities.

430 **8.2.3 InstallOptionsValues**

431 The InstallOptionsValues array parameter is used when any installation option needs to be input as a
432 key-value pair with this parameter containing the value part.

433 If an install option in the InstallOptions array parameter requires a value, and there is a NULL value
434 specified in the InstallOptionsValues array parameter at the corresponding index, the method shall return
435 2 (Error Occurred).

436 If an install option in the InstallOptions array parameter is required not to have a value, and a non-NULL
437 value is specified in the InstallOptionsValues array parameter at the corresponding index, the method
438 shall return 2 (Error Occurred).

439 **8.2.4 Source**

440 The Source parameter is a reference to the Software Identity that represents the software to be installed
441 or updated on a Managed Element. The method shall return 2 (Error Occurred) when this parameter is
442 NULL.

443 8.2.5 Target

444 The Target parameter is a reference to the instance of CIM_ManagedElement that represents a managed
445 element on which the Software Identity is intended to be installed or updated. If the Target parameter is a
446 reference to the Scoping Instance and

447 1) If the Software is applicable to a single managed element in its scope, including itself, the
448 method shall install the software on the managed element.

449 2) If the Software is applicable to more than one managed element in its scope, the method may
450 install the software on one, all or none of the managed elements. The behavior is
451 implementation specific.

452 When this parameter references an Instance of CIM_SoftwareIdentity representing a Software Bundle,
453 the method shall return 0 only if all the aggregated instances of Software Identity were successfully
454 installed. If at least one SWID was not installed successfully, the method shall return 2 (Error Occurred).

455 When this parameter is non-NULL and the method can install or update the Software Identity on the
456 Managed Element represented by the Target parameter, the method shall return 0. When this parameter
457 is non-NULL and the method cannot install or update the Software Identity on the Managed Element
458 represented by the Target parameter, the method shall return 2 (Error Occurred).

459 8.2.6 Collection

460 The Collection parameter is a reference to the instance of CIM_SystemSpecificCollection that represents
461 the collection of Available Software to which the Software Identity referenced by the Source parameter
462 will be added. When this parameter is not NULL and the CanAddToCollection property of the associated
463 instance of CIM_SoftwareInstallationServiceCapabilities is FALSE, the method shall return 2 (Error
464 Occurred).

465 When this parameter is non-NULL and the method can successfully add to the collection, the method
466 shall return 0. When this parameter is non-NULL and the method cannot add the Software Identity to the
467 collection, the method shall return 2 (Error Occurred).

468 When this parameter is a reference to a collection whose Scoping Instance does not have a
469 CIM_ServiceAffectsElement association to the CIM_SoftwareInstallationService upon which the method
470 was invoked, the method shall return 2 (Error Occurred).

471 When this parameter is not a reference to an instance of CIM_SystemSpecificCollection implemented as
472 defined in the [Software Inventory Profile](#) (Version 1.0), the method shall return 2 (Error Occurred).

473 8.3 CIM_SoftwareInstallationService.InstallFromByteStream()

474 CIM_SoftwareInstallationService.InstallFromByteStream() method allows a client application to download
475 or copy a series of bytes containing a software image to a Managed Element. When this method is
476 supported, at least one of SupportedAsynchronousActions property or SupportedSynchronousActions
477 property of the associated instance of CIM_SoftwareInstallationServiceCapabilities shall contain the value
478 4 (Install From ByteStream).

479 No standard messages are defined.

480 **Table 6 – CIM_SoftwareInstallationService.InstallFromByteStream() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob.

481 **Table 7 – CIM_SoftwareInstallationService.InstallFromByteStream() Method: Parameters**

Qualifiers	Name	Type	Description/Values
OUT	Job	CIM_ConcreteJob REF	See 8.3.1.
IN	InstallOptions	uint16[]	See 8.3.2
IN	InstallOptionsValues	string[]	See 8.3.3
IN	Image	uint8	See 8.3.4.
IN	Target	CIM_ManagedElement REF	See 8.3.5.

482 **8.3.1 Job**

483 The Job parameter is a reference to the instance of CIM_ConcreteJob that represents the job or task that
484 may be started by the invocation of the InstallFromByteStream() method.

485 The method shall not return the Job output parameter when SupportedAsynchronousActions property of
486 the associated instance of CIM_SoftwareInstallationServiceCapabilities does not contain the value
487 4 (Install From ByteStream).

488 The method may return the Job output parameter and a return code value of 4096 when the parameters
489 for the method have been validated and a job has been spawned to complete the installation/update.

490 **8.3.2 InstallOptions**

491 The InstallOptions array parameter is used to input the desired installation options to the
492 InstallFromSoftwareIdentity() method allowing the client to control the installation procedure. When this
493 parameter is NULL, the installation options used are implementation specific and no error shall be
494 returned. The method shall return 2 (Error Occurred) when this parameter contains an installation option
495 that is not listed in the SupportedInstallOptions property of the associated instance of
496 CIM_SoftwareInstallationServiceCapabilities.

497 **8.3.3 InstallOptionsValues**

498 The InstallOptionsValues array parameter is used when any installation option needs to be input as a
499 key-value pair with this parameter containing the value part.

500 If an install option in the InstallOptions array parameter requires a value, and there is a NULL value
501 specified in the InstallOptionsValues array parameter at the corresponding index, the method shall return
502 2 (Error Occurred).

503 If an install option in the InstallOptions array parameter is required not to have a value, and a non-NULL
504 value is specified in the InstallOptionsValues array parameter at the corresponding index, the method
505 shall return 2 (Error Occurred).

506 **8.3.4 Image**

507 The Image parameter is used to input the array of bytes that contain the installation image. When this
508 parameter is NULL, the method shall return 2 (Error Occurred).

509 **8.3.5 Target**

510 The Target parameter is a reference to the instance of CIM_ManagedElement that represents a managed
511 element on which the Software Identity is intended to be installed or updated. If the Target parameter is a
512 reference to the Scoping Instance and

- 513 1) If the Software is applicable to a single managed element in its scope, including itself, the
514 method shall install the software on the managed element.
- 515 2) If the Software is applicable to more than one managed element in its scope, the method may
516 install the software on one, all or none of the managed elements. The behavior is
517 implementation specific.

518 When this parameter is NULL, the method shall return 2 (Error Occurred).

519 **8.4 CIM_SoftwareInstallationService.InstallFromURI()**

520 CIM_SoftwareInstallationService.InstallFromURI() method allows a client application to install or update
521 software on a Managed Element from a URI. When this method is supported, at least one of
522 SupportedAsynchronousActions property or SupportedSynchronousActions property of the associated
523 instance of CIM_SoftwareInstallationServiceCapabilities shall contain the value 5 (Install From URI).

524 No standard messages are defined.

525 **Table 8 – CIM_SoftwareInstallationService.InstallFromURI() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob.

526 **Table 9 – CIM_SoftwareInstallationService.InstallFromURI() Method: Parameters**

Qualifiers	Name	Type	Description/Values
OUT	Job	CIM_ConcreteJob REF	See 8.4.1.
IN	InstallOptions	uint16[]	See 8.4.2
IN	InstallOptionsValues	string[]	See 8.4.3
IN	URI	string	See 8.4.4.
IN	Target	CIM_ManagedElement REF	See 8.4.5.

527 **8.4.1 Job**

528 The Job parameter is a reference to the instance of CIM_ConcreteJob that represents the job or task that
529 may be started by the invocation of the InstallFromURI() method.

530 The method shall not return the Job output parameter when SupportedAsynchronousActions property of
531 the associated instance of CIM_SoftwareInstallationServiceCapabilities does not contain the value
532 5 (Install From URI).

533 The method may return the Job output parameter and a return code value of 4096 when the parameters
534 for the method have been validated and a job has been spawned to complete the installation/update.

535 **8.4.2 InstallOptions**

536 The InstallOptions array parameter is used to input the desired installation options to the
537 InstallFromSoftwareIdentity() method allowing the client to control the installation procedure. When this
538 parameter is NULL, the installation options used are implementation specific. The method shall return
539 2 (Error Occurred) when this parameter contains an installation option that is not listed in the
540 SupportedInstallOptions property of the associated instance of
541 CIM_SoftwareInstallationServiceCapabilities.

542 **8.4.3 InstallOptionsValues**

543 The InstallOptionsValues array parameter is used when any installation option needs to be input as a
544 key-value pair with this parameter containing the value part.

545 If an install option in the InstallOptions array parameter requires a value, and there is a NULL value
546 specified in the InstallOptionsValues array parameter at the corresponding index, the method shall return
547 2 (Error Occurred).

548 If an install option in the InstallOptions array parameter is required not to have a value, and a non-NULL
549 value is specified in the InstallOptionsValues array parameter at the corresponding index, the method
550 shall return 2 (Error Occurred).

551 **8.4.4 URI**

552 The URI parameter is used to specify the URI information of the software to be installed on the Managed
553 Element. When the URI is NULL or not well-formed according to [RFC 2396](#), the InstallFromURI() method
554 shall return 2 (Error Occurred). When the URI scheme of this parameter is not present in the
555 SupportedURISchemes[] property of the associated instance of
556 CIM_SoftwareInstallationServiceCapabilities, the method shall return 2 (Error Occurred).

557 **8.4.5 Target**

558 The Target parameter is a reference to the instance of CIM_ManagedElement that represents a managed
559 element on which the Software Identity is intended to be installed or updated. If the Target parameter is a
560 reference to the Scoping Instance and

- 561 1) If the Software is applicable to a single managed element in its scope, including itself, the
562 method shall install the software on the managed element.
- 563 2) If the Software is applicable to more than one managed element in its scope, the method may
564 install the software on one, all or none of the managed elements. The behavior is
565 implementation specific.

566 When this parameter is NULL, the method shall return 2 (Error Occurred).

567 **8.5 Profile Conventions for Operations**

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

570 The default list of operations is as follows:

- 571 • GetInstance
- 572 • Associators

- 573 • AssociatorNames
- 574 • References
- 575 • ReferenceNames
- 576 • EnumerateInstances
- 577 • EnumerateInstanceNames

578 8.6 CIM_SoftwareInstallationService Operations

579 All operations in the default list in 8.5 shall be implemented as defined in [DSP0200](#).

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

581 8.7 CIM_HostedService Operations

582 Table 10 lists implementation requirements for operations. If implemented, these operations shall be
 583 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 10, all operations
 584 in the default list in 8.5 shall be implemented as defined in [DSP0200](#).

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

586 **Table 10 – Operations: CIM_HostedService**

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

587 8.8 CIM_SoftwareInstallationServiceCapabilities Operations

588 All operations in the default list in 8.5 shall be implemented as defined in [DSP0200](#).

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

590 8.9 CIM_ElementCapabilities Operations

591 Table 11 lists implementation requirements for operations. If implemented, these operations shall be
 592 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 11, all operations
 593 in the default list in 8.5 shall be implemented as defined in [DSP0200](#).

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

595 **Table 11 – CIM_ElementCapabilities Operations**

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

596 **8.10 CIM_ServiceAffectsElement**

597 Table 12 lists implementation requirements for operations. If implemented, these operations shall be
 598 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 12, all operations
 599 in the default list in 8.5 shall be implemented as defined in [DSP0200](#).

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

601 **Table 12 – CIM_ServiceAffectsElement Operations**

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

602 **9 Use Cases**

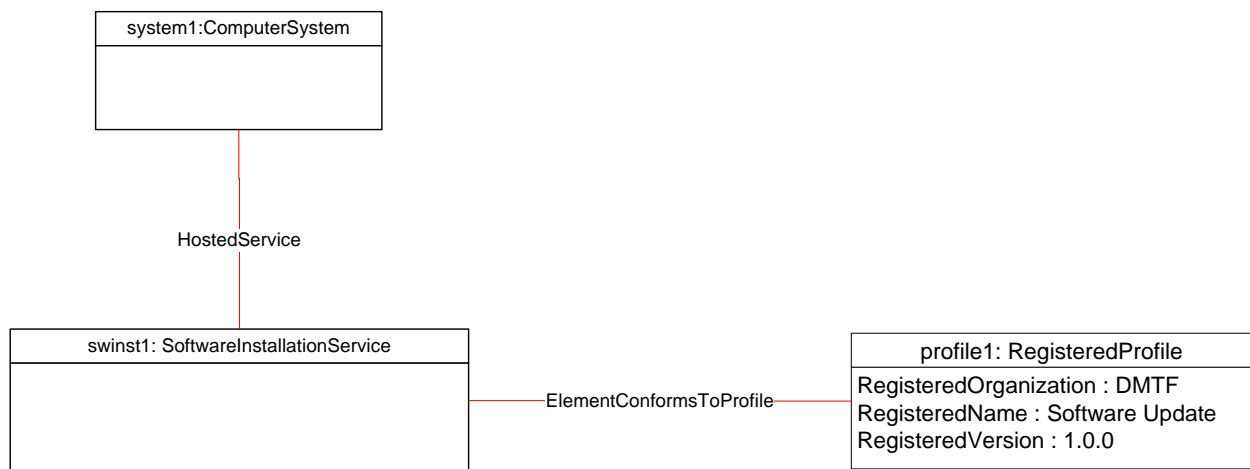
603 This section contains Object Diagrams and Use Cases for the *Software Update Profile*.

604 **9.1 Object Diagrams**

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

607 **9.1.1 Registered Profile**

608 Figure 2 represents a possible instantiation of the *Software Update Profile*. In this instantiation, the central
 609 instance, swinst1 has a CIM_HostedService association to the Scoping Instance, system1. Profile
 610 registration information is represented by profile1. Following the CIM_ElementConformsToProfile
 611 association from the central instance to profile1, the client can retrieve information such as the version of
 612 the current *Software Update Profile* implementation.



613

614

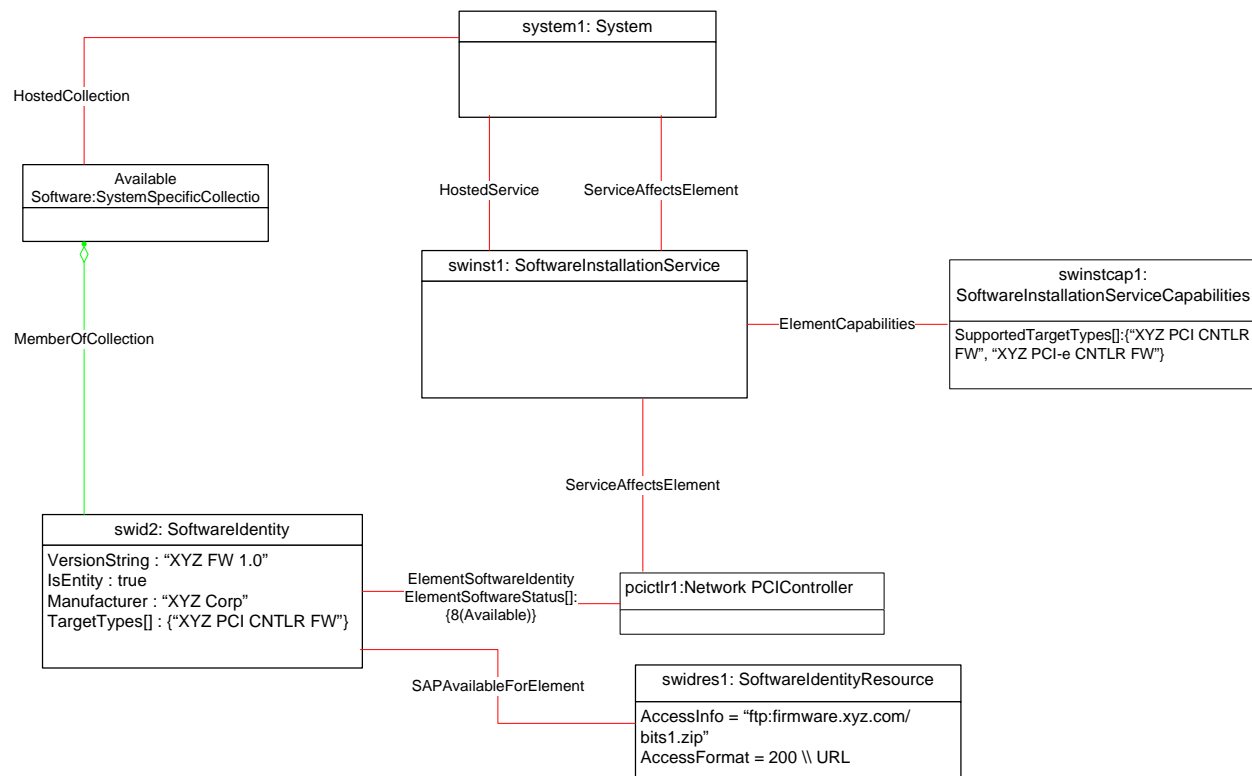
Figure 2 – Registered Profile

615 9.1.2 Representing Available Software, Managed Element, Software Installation Service 616 and their Relationships

617 Figure 3 represents a possible instantiation of the *Software Update Profile*. In this instantiation, the
618 optional behavior of “Representing Available Software” from the [Software Inventory Profile](#) has been
619 implemented. The managed system, system1 hosts a collection, “Available Software” and an installation
620 service, swinst1. The firmware image applicable to the Network PCIController (pcictrl1) is represented by
621 the Software Identity (swid2) which is a member of the “Available Software” collection. A
622 CIM_ElementSoftwareIdentity association is shown between the pcictrl1 and swid2.

623 The capabilities of swinst1 are represented by the instance of
624 CIM_SoftwareInstallationServiceCapabilities (swinstcap1). The TargetTypes[] property on swid2 has a
625 value that matches one of the values in SupportedTargetTypes[] property of swinstcap1 and so swid2 is
626 compatible with swinst1. Thus, swid2 can be installed or updated using swinst1.

627 The CIM_ServiceAffectsElement association between pcictrl1 and swinst1 indicates that the swinst1 can
628 provide a software install or update service to pcictrl1. The CIM_ServiceAffectsElement association
629 between system1 and swinst1 indicates that the swinst1 can provide a software install or update service
630 to system1 and or components installed in system1.



631

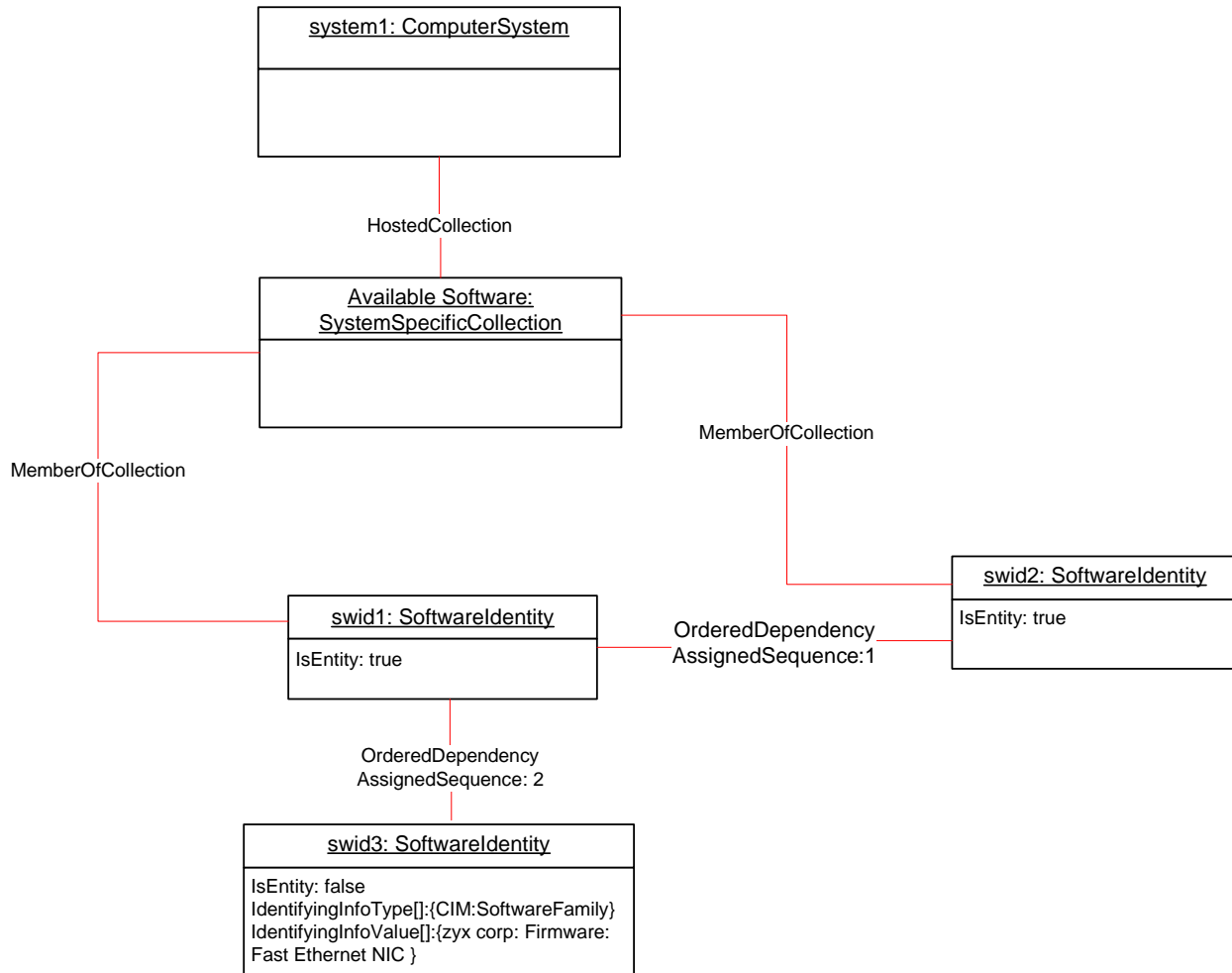
632 **Figure 3 – Software Update Profile: Object Diagram**

633 9.1.3 Representing a Software Identity with Installation Dependencies

634 Figure 4 represents a possible instantiation of the *Software Update Profile*. In this instantiation, the
635 optional behavior of “Representing Installation Dependencies” from the [Software Inventory Profile](#) has
636 been implemented. The Software Identity, swid1, is a member of the “Available Software” collection and
637 has Installation Dependencies on other Software Identities swid2 and swid3. A copy of swid2 is available

638 and so the IsEntity property of swid2 is true. A copy of the swid3 is not available or installed, and so the
 639 IsEntity property of swid 3 is false.

640 swid2 followed by swid3 need to be installed before installing swid1. The object diagram does not show
 641 the instances of CIM_SoftwareInstallationService that are compatible with swid1 and swid2.



642

643 **Figure 4 – Software Update Profile: Object Diagram**

644 **9.1.4 Representing a Software Identity with an Installation Dependency which Is**
 645 **Installed**

646 Figure 5 represents a possible instantiation of the *Software Update Profile*. In this instantiation, the
 647 optional behavior of “Representing Installation Dependencies” from the [Software Inventory Profile](#) has
 648 been implemented. The Software Identity, swid1, is a member of the “Available Software” collection and
 649 has Installation Dependencies on another Software Identity, swid3.

650 swid2, which is installed on the system, belongs to the same Software Family as swid3 and has a higher
 651 version, and so the Installation Dependency of swid1 is satisfied.

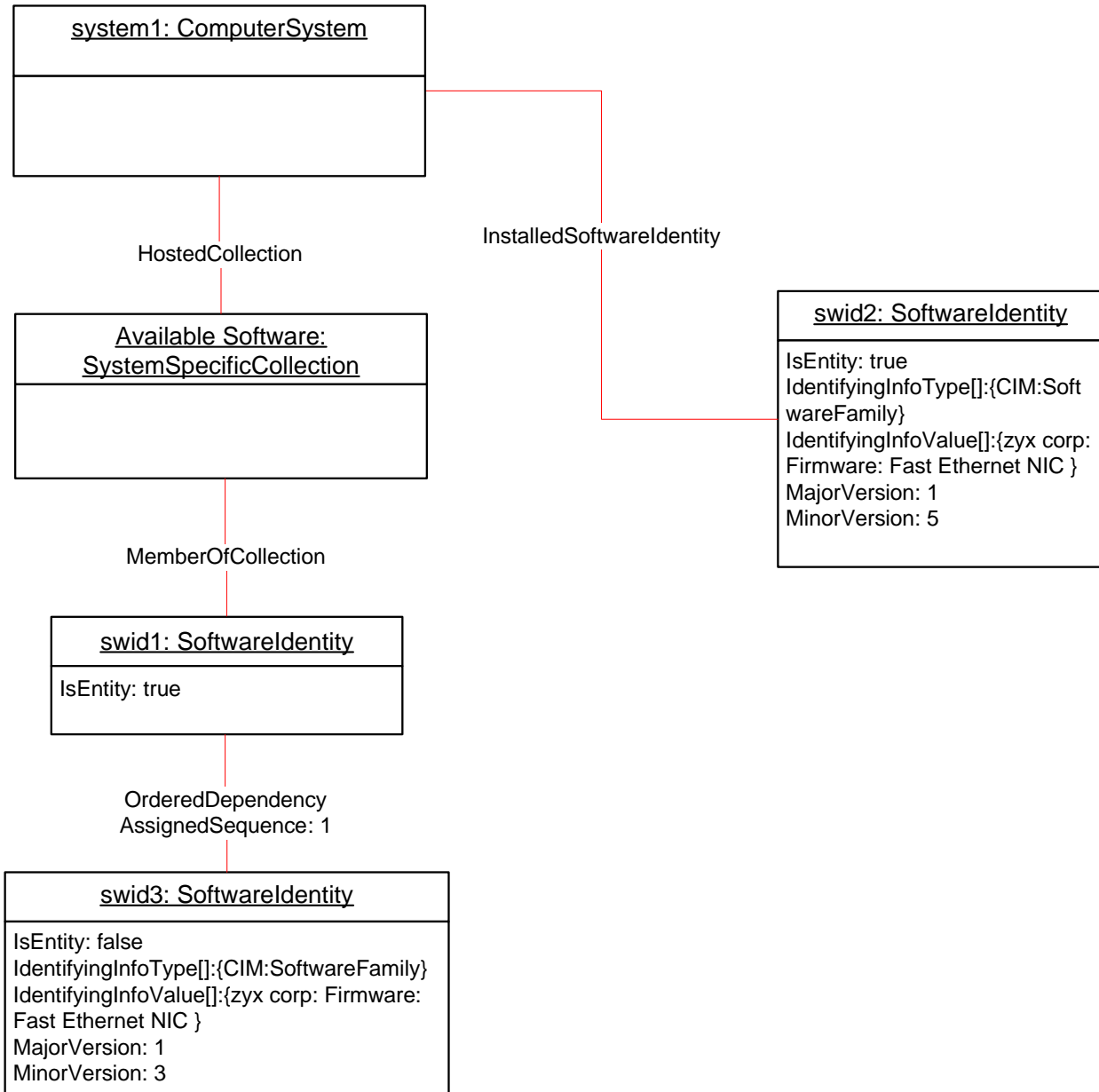
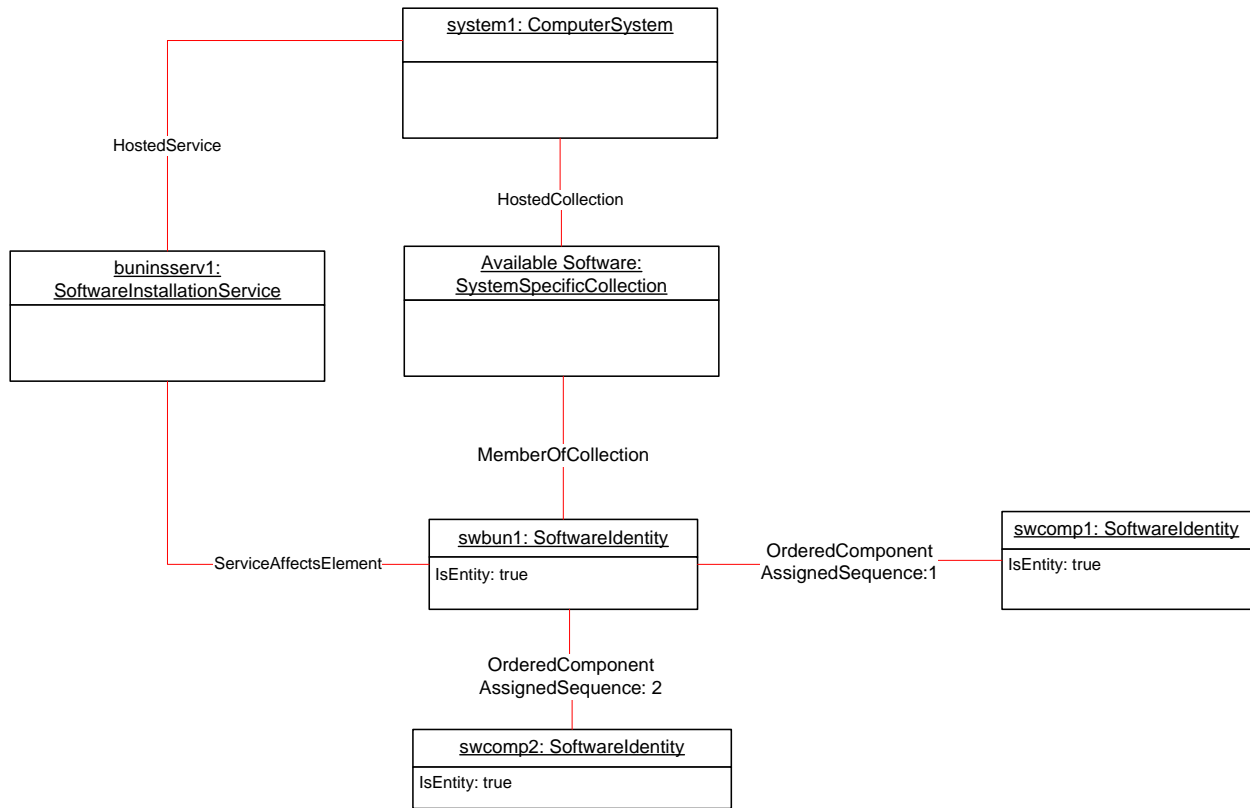


Figure 5 – Software Update Profile: Object Diagram

9.1.5 Representing Software Bundles

Figure 6 represents a possible instantiation of the *Software Update Profile*. In this instantiation, the optional behavior of “Representing a Software Bundle” from the *Software Inventory Profile* has been implemented. The Software Bundle, `swbun1`, is a member of the “Available Software” collection and has the aggregated instances of `swcomp1` and `swcomp2`. The Software Installation Service, `buninsserv1`, is compatible with `swbun1` which is indicated by the `CIM_ServiceAffectsElement` association between the Software Bundle and the Software Installation Service. `buninsserv1` can be used for installing `swbun1`.



662

663

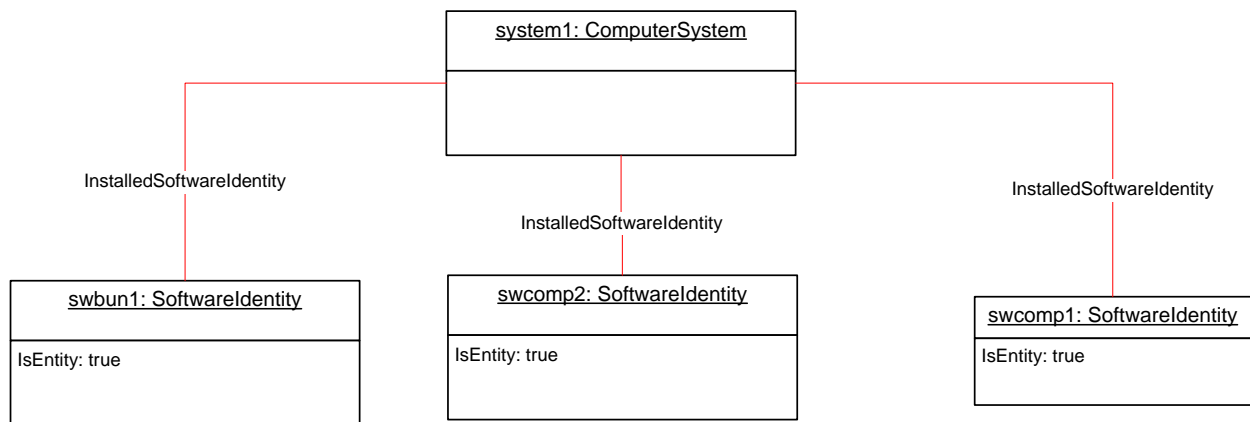
Figure 6 – Software Update Profile: Object Diagram

664

Figure 7 represents the result of installing swbun1. In this instantiation, swbun1, swcomp1 and swcomp2 are shown as Installed Software for the system. In this example, the Software Bundle is a software package which is tracked separately from the contained software components.

665

666



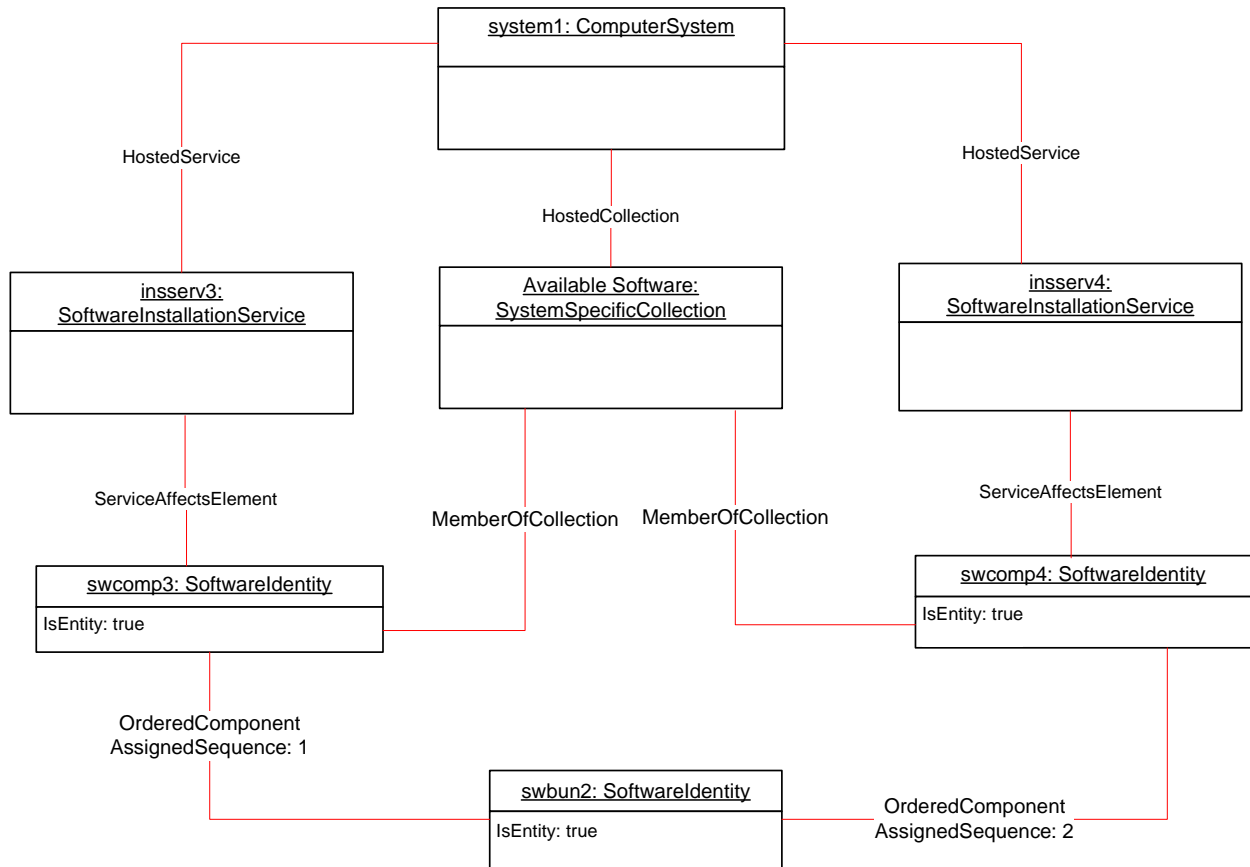
667

668

Figure 7 – Software Update Profile: Object Diagram

669 **9.1.6 Representing Software Bundles**

670 Figure 8 represents a possible instantiation of the *Software Update Profile*. In this instantiation, the
 671 optional behavior of “Representing a Software Bundle” from the [Software Inventory Profile](#) has been
 672 implemented. The Software Bundle, swbun2 has the aggregated instances of swcomp3 and swcomp4.
 673 The Software Installation Service, insserv3, is compatible with swcomp3 and can be used for installing it.
 674 The Software Installation Service, insserv4, is compatible with swcomp4 and can be used for installing it.
 675 swbun2 cannot be a direct target of installation as there is no compatible Software Installation Service.

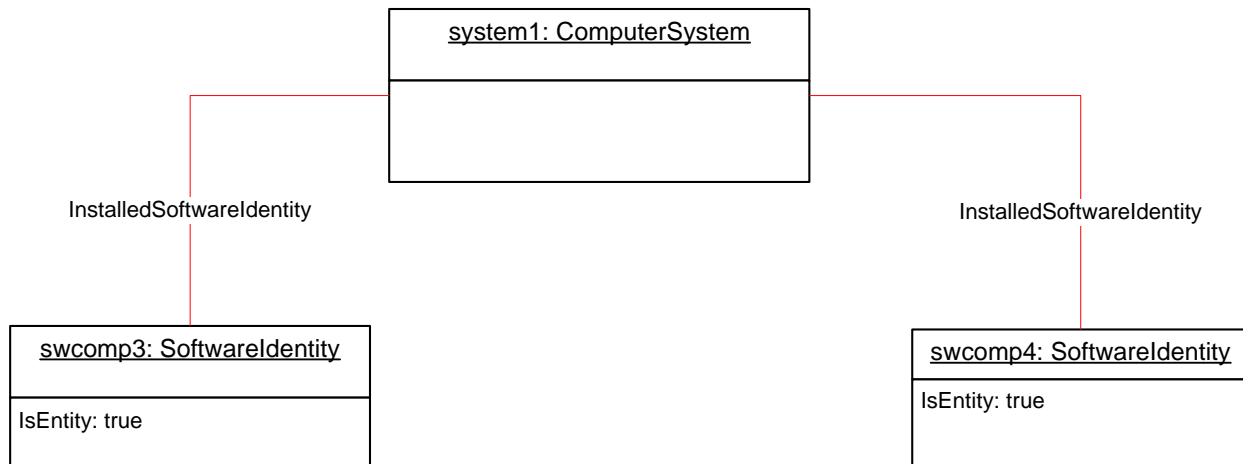


676

677

Figure 8 – Software Update Profile: Object Diagram

678 Figure 9 represents the result of installing swcomp3 and swcomp4. In this instantiation swcomp3 and
 679 swcomp4 are shown as Installed Software for the system. swbun2 was not the target of installation and
 680 therefore is not shown as Installed Software.



681

682

Figure 9 – Software Update Profile: Object Diagram

683 **9.2 Find the Software Installation Services compatible with a Software Identity**

684 A client can determine the Software Installation Services compatible with a Software Identity as follows:

- 685 1) For the given Software Identity, select the CIM_SoftwareInstallationService instances that are
 686 associated to the Software Identity through the CIM_ServiceAffectsElement association.
- 687 2) Select the instances of CIM_SoftwareInstallationService with at least one value in the
 688 SupportedTargetTypes property of the associated CIM_SoftwareInstallationServiceCapabilities
 689 instance equal to at least one value in the TargetTypes property of the given Software Identity.
- 690 3) Select the instances of CIM_SoftwareInstallationService with at least one value in the
 691 SupportedExtendedResourceTypes property equal to the ExtendedResourceType property of
 692 the given Software Identity and the version of the installer format supported by
 693 CIM_SoftwareInstallationService instance is equal to or higher than the version of the installer
 694 format supported by the Software Identity (see 7.3.2).
- 695 4) The instances of CIM_SoftwareInstallationService from steps 1, 2 and 3 represent the Software
 696 Installation Services that are compatible with the Software Identity.

697 **9.3 Determine Whether Installing a Software Identity Requires a Reboot**

698 A client can determine whether installing a Software Identity requires a reboot using the following steps:

- 699 1) Find the Software Installation Service compatible with the Software Identity by following the
 700 steps in 9.2
- 701 2) Invoke the CheckSoftwareIdentity() method on the CIM_SoftwareInstallationService instance
 702 with the given Software Identity as the Source parameter. After successful execution of the
 703 method, if the InstallCharacteristics parameter contains the value 7 (No Reboot Required), then
 704 no reboot is required after installing the Software identity. If it contains the value 6 (Manual
 705 Reboot Required), then a reboot has to be performed to complete the installation. If it does not
 706 contain 6 or 7 then no information about the requirements for the reboot can be determined.

707 **9.4 Find Software Available for Installation on a Managed Element when** 708 **CIM_ElementSoftwareIdentity Exists**

709 Assuming that the Software Identities compatible to a Managed Element are associated to the Managed
710 Element through CIM_ElementSoftwareIdentity, a client can find the Software Identities available for
711 installation that are applicable to a Managed Element by using the following step:

- 712 1) Select the instances of Software Identity that are associated to the instance of
713 CIM_ManagedElement through an instance of CIM_ElementSoftwareIdentity with the
714 ElementSoftwareStatus property containing the value 8 (Available).

715 **9.5 Find Software Available for Installation on a Managed Element when** 716 **CIM_ElementSoftwareIdentity Does Not Exist**

717 When the Software Identities compatible to a Managed Element are not associated to the Managed
718 Element through CIM_ElementSoftwareIdentity, a client can find the Software Identities available for
719 installation that are applicable to a Managed Element by using the following steps:

- 720 1) Starting at the Scoping Instance, find all the Available Software following steps described in
721 section 9.5 of the [Software Inventory Profile](#).
- 722 2) Find the instances of CIM_SoftwareInstallationService that can provide installation or update
723 service to the Managed Element following the steps described in section 9.7.
- 724 3) For each Software Identity from step 1, find the compatible Software Installation Services
725 following the steps described in section 9.2.
- 726 4) For each Software Installation Service which is also in the set of Software Installation Services
727 found in step 2, invoke the CheckSoftwareIdentity() method using the appropriate parameters.
- 728 5) If the method returns 0, the Software Identity can be installed on the Managed Element.

729 **9.6 Find Software Available for Installation on a Component**

730 Given a priori knowledge of the values of the properties of an instance of Software Identity when the
731 instance of Software Identity is applicable to the component of interest, a client can find the Software
732 Identities available for installation that are applicable to the component using the following steps:

- 733 1) Starting at the instance of CIM_ComputerSystem representing the system to which the
734 component belongs, find all the Available Software following the steps described in section 9.5
735 of the *Software Inventory Profile*.
- 736 2) Select the Software Identities from step 1 where the property values match the required values
737 for the component.

738 **9.7 Find Software Installation Services that Can Install or Update Software on a** 739 **Managed Element**

740 A client can find the Software Installation Services that can install or update software on a Managed
741 Element by using the following steps:

- 742 1) Starting from the Managed Element, select the instances of CIM_SoftwareInstallationService
743 that are associated through CIM_ServiceAffectsElement association.
- 744 2) Select the instances of CIM_SoftwareInstallationService that are associated to the Scoping
745 Instance through CIM_ServiceAffectsElement association.
- 746 3) The instances of CIM_SoftwareInstallationService from steps 1 and 2 represent the Software
747 Installation Services that could provide installation or update service to the Managed Element.

748 **9.8 Install or Update Software on a Managed Element Using Software Identity**

749 A client can install or update software on a Managed Element with a Software Identity by using the
750 following steps:

- 751 1) Find all the Software Identities that are applicable to the Managed Element following the steps
752 described in section 9.4 and section 9.5. Select the Software Identity of interest.
- 753 2) Find the instances of CIM_SoftwareInstallationService that can provide installation or update
754 service to the Managed Element following the steps described in section 9.7.
- 755 3) For the Software Identity from step 1, find the compatible Software Installation Services
756 following the steps described in section 9.2.
- 757 4) For each Software Installation Service which is also in the set of Software Installation Services
758 found in step 2, invoke the CheckSoftwareIdentity() method using the appropriate parameters.
- 759 5) If the method returns 0, invoke the InstallFromSoftwareIdentity() method on the instance of
760 CIM_SoftwareInstallationService with the appropriate parameters.
- 761 6) Else if the Software Identity from step 1 is referenced by an instance of
762 CIM_SAPAvailableForElement.
 - 763 a) Check if at least one of SupportedAsynchronousActions property or
764 SupportedSynchronousActions property of the associated instance of
765 CIM_SoftwareInstallationServiceCapabilities contains the value 5(Install From URI).
 - 766 • Starting from the Software Identity, select the instance of
767 CIM_SoftwareIdentityResource through the CIM_SAPAvailableForElement
768 association.
 - 769 • Extract the URI information using the instance of CIM_SoftwareIdentityResource and
770 invoke the InstallFromURI() method with the appropriate parameters.

771 **9.9 Install from Software Identity when the Managed Element is not modeled**

772 A client can install or update software represented as a Software Identity on a component which is not
773 modeled as a Managed Element by using the following steps:

- 774 1) Find all the Software Identities that are applicable to the component following the steps
775 described in section 9.6. Select the Software Identity of interest.
- 776 2) Find the instances of CIM_SoftwareInstallationService that can provide installation or update
777 service to the instance of CIM_ComputerSystem representing the system to which the
778 component belongs, following the steps described in section 9.7.
- 779 3) For the Software Identity from step 1, find the compatible Software Installation Services
780 following the steps described in section 9.1.6.
- 781 4) For each Software Installation Service from step 3, which is also in the set of Software
782 Installation Services found in step 2:
 - 783 a) Invoke the InstallFromSoftwareIdentity() method on the instance of
784 CIM_SoftwareInstallationService with the Target parameter as the Scoping Instance.
 - 785 b) If the method returns 0, then the Software Identity was successfully installed.

- 786 5) If the Software Identity was not installed, check if the Software Identity from step 1 is referenced
787 by an instance of CIM_SAPAvailableForElement
- 788 a) Check if at least one of SupportedAsynchronousActions property or
789 SupportedSynchronousActions property of the associated instance of
790 CIM_SoftwareInstallationServiceCapabilities contains the value 5 (Install From URI).
- 791 • Starting from the Software Identity, select the instance of
792 CIM_SoftwareIdentityResource through the CIM_SAPAvailableForElement
793 association.
- 794 • Extract the URI information using the instance of CIM_SoftwareIdentityResource and
795 invoke the InstallFromURI() method with the appropriate parameters.
- 796 • If the method returns 0, then the Software Identity was successfully installed.

797 9.10 Install or Update a Software on a Managed Element Using a URI

798 A client can install or update software on a Managed Element using a URI that identifies the software by
799 using the following steps:

- 800 1) Find the instances of CIM_SoftwareInstallationService that can install or update software on the
801 Managed Element using the steps described in section 9.7.
- 802 2) Select an instance of CIM_SoftwareInstallationService with the associated instance of
803 CIM_SoftwareInstallationServiceCapabilities having at least one of the values in the
804 SupportedAsynchronousActions property or SupportedSynchronousActions property equal to 5
805 (Install From URI) and the SupportedURISchemes property containing the URI scheme of the
806 URI.
- 807 3) Invoke the InstallFromURI() method on the instance of CIM_SoftwareInstallationService from
808 step 2 using the appropriate parameters.

809 9.11 Install from URI When the Managed Element Is Not Modeled

810 A client can install or update software on a component which is not modeled as a Managed Element
811 using a URI that identifies the software by using the following steps:

- 812 1) Find the instances of CIM_SoftwareInstallationService that can provide installation or update
813 service to the instance of CIM_ComputerSystem representing the system to which the
814 component belongs, following the steps described in section 9.7.
- 815 2) Select an instance of CIM_SoftwareInstallationService with the associated instance of
816 CIM_SoftwareInstallationServiceCapabilities having at least one of the values in the
817 SupportedAsynchronousActions property or SupportedSynchronousActions property equal to
818 5(Install From URI) and the SupportedURISchemes property containing the URI scheme of the
819 URI.
- 820 3) Invoke the InstallFromURI() method on the instance of CIM_SoftwareInstallationService from
821 step 2 using the appropriate parameters.

822 9.12 Update Software on a Managed Element Using a Byte Stream

823 A client can install or update software on a Managed Element by transferring the image as a byte array by
824 using the following steps:

- 825 1) Find the instances of CIM_SoftwareInstallationService that can install or update software on the
826 Managed Element using the steps described in section 9.7.

- 827 2) Select an instance of CIM_SoftwareInstallationService with the associated instance of
- 828 CIM_SoftwareInstallationServiceCapabilities having at least one of the values in the
- 829 SupportedAsynchronousActions property or SupportedSynchronousActions property equal to
- 830 4(Install From ByteStream).
- 831 3) Invoke the InstallFromByteStream() method on the instance of CIM_SoftwareInstallationService
- 832 from step 2 using the appropriate parameters.

833 **10 CIM Elements**

834 **Table 13 – CIM Elements: Software Update Profile**

Element Name	Requirement	Description
Classes		
CIM_HostedService	Mandatory	See 10.1.
CIM_SoftwareInstallationService	Mandatory	See 7.1 and 10.2.
CIM_ElementCapabilities	Mandatory	See 10.3.
CIM_SoftwareInstallationCapabilities	Mandatory	See 7.2 and 10.4.
CIM_ServiceAffectsElement	Optional	See 7.3.3, 10.5 and 10.6.
CIM_SoftwareIdentity	Optional	See 7.5 and 10.7.
CIM_RegisteredProfile	Mandatory	See 10.8.
Indications		
None defined in this profile		

835 **10.1 CIM_HostedService**

836 CIM_HostedService associates the CIM_ComputerSystem instance with the
 837 CIM_SoftwareInstallationService instance that it hosts. Table 14 contains the requirements for elements
 838 of this class.

839 **Table 14 – Class: CIM_HostedService**

Elements	Requirement	Notes
Antecedent	Mandatory	Key: This property shall be a reference to the instance of CIM_ComputerSystem. Cardinality 1
Dependent	Mandatory	Key: This property shall be a reference to the instance of CIM_SoftwareInstallationService. Cardinality *

840 **10.2 CIM_SoftwareInstallationService**

841 CIM_SoftwareInstallationService is used to represent a Software Installation Service. Table 15 contains
 842 the requirements for elements of this class.

843

Table 15 – Class: CIM_SoftwareInstallationService

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
Name	Mandatory	Key
CheckSoftwareIdentity()	Optional	See 8.1.
InstallFromSoftwareIdentity()	Optional	See 8.2.
InstallFromByteStream()	Optional	See 8.3.
InstallFromURI()	Optional	See 8.4.

844 10.3 CIM_ElementCapabilities

845 CIM_ElementCapabilities associates the CIM_SoftwareInstallationService instance that represents the
 846 service responsible for performing software installations/updates with the
 847 CIM_SoftwareInstallationServiceCapabilities instance that represents the capabilities of the Software
 848 Installation Service. Table 16 contains the requirements for elements of this class.

849

Table 16 – Class: CIM_ElementCapabilities

Elements	Requirement	Notes
ManagedElement	Mandatory	Key: This property shall be a reference to the instance of CIM_SoftwareInstallationService. Cardinality 1..*
Capabilities	Mandatory	Key: This property shall be a reference to the instance of CIM_SoftwareInstallationServiceCapabilities. Cardinality 1

850 10.4 CIM_SoftwareInstallationCapabilities

851 CIM_SoftwareInstallationServiceCapabilities represents the capabilities of a Software Installation Service.
 852 Table 17 contains the requirements for elements of this class.

853

Table 17 – Class: CIM_SoftwareInstallationCapabilities

Elements	Requirement	Notes
InstanceID	Mandatory	Key
SupportedTargetTypes[]	Optional	See 7.3.1.
SupportedExtendedResourceTypes[]	Optional	See 7.3.2.
SupportedExtendedResourceTypesMajorVersions[]	Optional	See 7.3.2.
SupportedExtendedResourceTypesMinorVersions[]	Optional	See 7.3.2.
SupportedExtendedResourceTypesRevisionNumbers[]	Optional	See 7.3.2.
SupportedExtendedResourceTypesBuildNumbers[]	Optional	See 7.3.2.
SupportedInstallOptions[]	Mandatory	
SupportedURISchemes[]	Conditional	See 7.2.1

854 **10.5 CIM_ServiceAffectsElement – CIM_SoftwareIdentity Reference**

855 CIM_ServiceAffectsElement associates the instance of CIM_SoftwareInstallationService to the instance
 856 of CIM_SoftwareIdentity. Table 18 contains the requirements for elements of this class.

857 **Table 18 – Class: CIM_ServiceAffectsElement**

Elements	Requirement	Notes
AffectedElement	Mandatory	Key: This property shall be a reference to the instance of CIM_SoftwareIdentity. Cardinality *
AffectingElement	Mandatory	Key: This property shall be a reference to the instance of CIM_SoftwareInstallationService. Cardinality *

858 **10.6 CIM_ServiceAffectsElement – CIM_ManagedElement Reference**

859 CIM_ServiceAffectsElement associates the instance of CIM_SoftwareInstallationService to the instance
 860 of CIM_ManagedElement. Table 19 contains the requirements for elements of this class.

861 **Table 19 – Class: CIM_ServiceAffectsElement**

Elements	Requirement	Notes
AffectedElement	Mandatory	Key: This property shall be a reference to the instance of CIM_ManagedElement. Cardinality *
AffectingElement	Mandatory	Key: This property shall be a reference to the instance of CIM_SoftwareInstallationService. Cardinality *

862 **10.7 CIM_SoftwareIdentity**

863 CIM_SoftwareIdentity is defined by the *DSP1023*. The requirements denoted in Table 20 are in addition
 864 to those mandated by the *DSP1023*.

865 **Table 20 – Class: CIM_SoftwareIdentity**

Elements	Requirement	Notes
TargetTypes[]	Optional	See 7.3.1.
ExtendedResourceType	Optional	See 7.3.2.
MinExtendedResourceTypeMajorVersion	Optional	See 7.3.2.
MinExtendedResourceTypeMinorVersion	Optional	See 7.3.2.
MinExtendedResourceTypeRevisionNumber	Optional	See 7.3.2.
MinExtendedResourceTypeBuildNumber	Optional	See 7.3.2.

866 **10.8 CIM_RegisteredProfile**

867 CIM_RegisteredProfile is defined by the *DSP1033*. The requirements denoted in
868 Table 21 are in addition to those mandated by the *DSP1033*.

869 **Table 21 – Class: CIM_RegisteredProfile**

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

870
871
872
873
874

ANNEX A (Informative)

Change Log

Version	Date	Author	Description
1.0.0	2009-06-16		DMTF Standard Release

875