Document Number: DSP1025

Date: 2006-11-02

Version: 1.0.0a

Software Update Profile

Document Type: Specification

Document Status: Preliminary Standard

Document Language: E

CONTENTS

Figures

Tables

Foreword

The *Software Update Profile* (DSP1025) was prepared by the Server Management Working Group of the DMTF.

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

Introduction

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

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

136 # Software Update Profile

137 ## 1 Scope

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

141 ## 2 Normative References

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

145 ### 2.1 Approved References

146 DMTF DSP0004, *CIM Infrastructure Specification 2.3.0*

147 DMTF DSP0200, *CIM Operations over HTTP 1.2.0*

148 DMTF DSP1000, *Management Profile Specification Template*

149 DMTF DSP1001, *Management Profile Specification Usage Guide*

150 ### 2.2 References under Development

151 DMTF DSP1033, *Profile Registration Profile*

152 DMTF DSP1023, *Software Inventory Profile*

153 ### 2.3 Other References

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

156 Unified Modeling Language (UML) from the Open Management Group (OMG), http://www.uml.org

157 DMTF DSP0215, *Server Management Managed Element Addressing Specification* (SM ME Addressing)

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

159 ## 3 Terms and Definitions

160 For the purposes of this document, the following terms and definitions apply. For the purposes of this
161 document, the terms and definitions given in DSP1033, DSP1001, and DSP1023 also apply.

162 ### 3.1
163 **can**
164 used for statements of possibility and capability, whether material, physical, or causal

3.2
cannot
used for statements of possibility and capability, whether material, physical, or causal

3.3
conditional
indicates requirements to be followed strictly to conform to the document when the specified conditions are met

3.4
mandatory
indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.5
may
indicates a course of action permissible within the limits of the document

3.6
need not
indicates a course of action permissible within the limits of the document

3.7
optional
indicates a course of action permissible within the limits of the document

3.8
referencing profile
indicates a profile that owns the definition of this class and can include a reference to this profile in its “Referenced Profiles” table

3.9
shall
indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.10
shall not
indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.11
should
indicates that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

3.12
should not
indicates that a certain possibility or course of action is deprecated but not prohibited

3.13
unspecified
indicates that this profile does not define any constraints for the referenced CIM element or operation

3.14
Software Installation Service
a component that can be used to perform an installation or update of software on a managed element

4 Symbols and Abbreviated Terms

None

5 Synopsis

Profile Name: Software Update

Version: 1.0.0

Organization: DMTF

CIM schema version: 2.14

Central Class: CIM_SoftwareInstallationService

Scoping Class: CIM_System

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

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

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

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

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

Table 1 – Referenced Profiles

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

6 Description

The *Software Update Profile* provides the ability to perform installation or update of software on managed elements in the scope of a managed system. *Installation* implies the first-time installation of the software on the managed element, and *update* implies that the managed element has a version of the software already installed on it. The profile also defines the relationship between a managed element and the installation service that represents the availability of software installation and update functionality for a managed element.

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

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

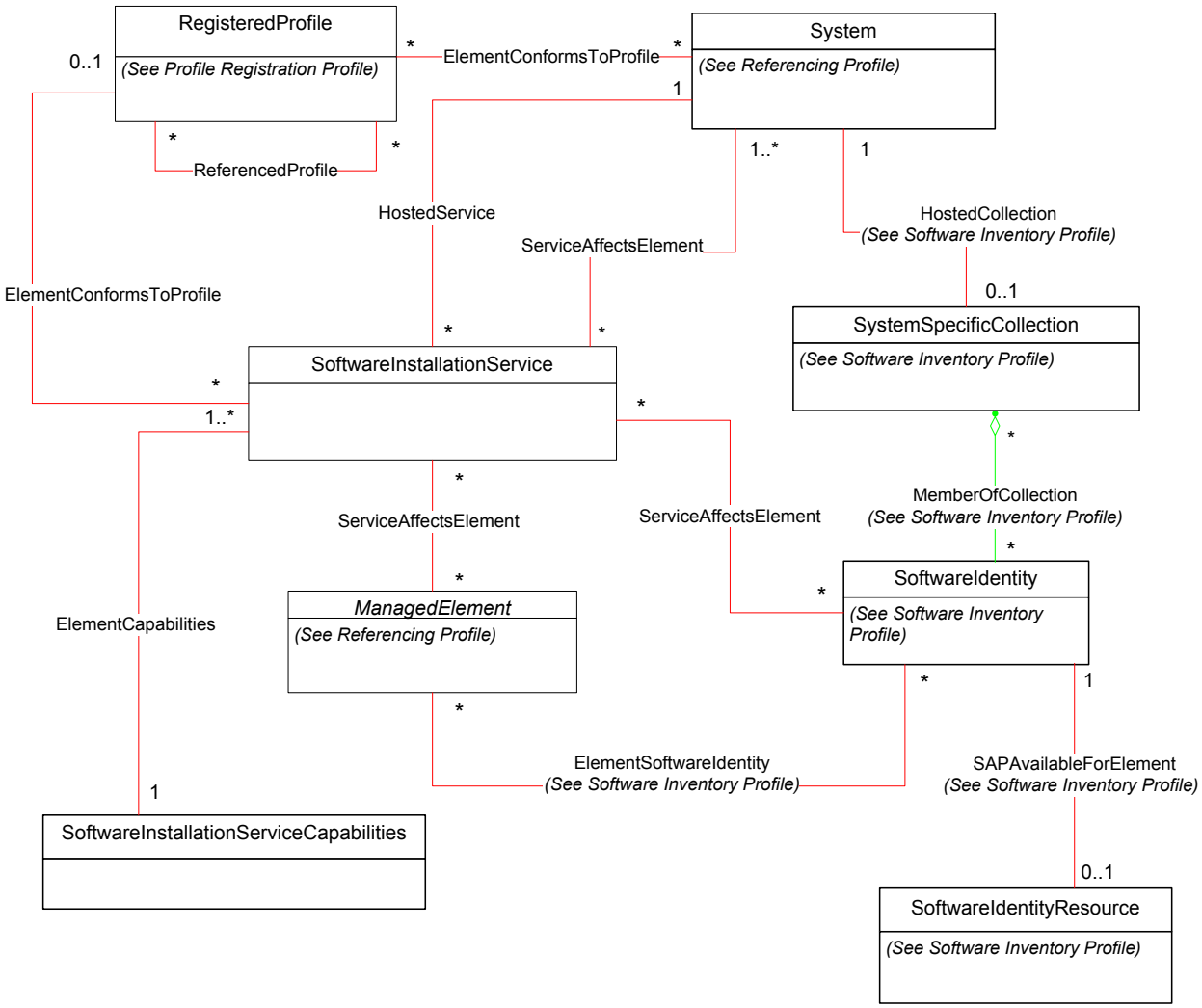


Figure 1 – Class Diagram: *Software Update Profile*

7 Implementation

This section details the requirements related to the arrangement of instances and their properties for implementations of this profile. Required methods are described in section 8 (“Methods”), and properties are described in section 10 (“CIM Elements”).

7.1 CIM_SoftwareInstallationService

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

7.2 CIM_SoftwareInstallationServiceCapabilities

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

7.2.1 CIM_SoftwareInstallationServiceCapabilities.SupportedURISchemes[ ]

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

7.3 Advertising Compatibility with a Software Identity (Optional)

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

7.3.1 Using Target Types

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

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

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

7.3.2 Using Extended Resource Type

The CIM_SoftwareIdentity.ExtendedResourceType property shall represent a single format for an installer that is capable of installing or updating the Software Identity. The minimum version of the installer format required for compatibility shall be represented using the following properties of the Software Identity:

- MinExtendedResourceTypeMajorVersion
- MinExtendedResourceTypeMinorVersion
- MinExtendedResourceTypeRevisionNumber
- MinExtendedResourceTypeBuildNumber

The installer formats supported by the instance of CIM_SoftwareInstallationService shall be represented using the SupportedExtendedResourceTypes array property of the associated CIM_SoftwareInstallationServiceCapabilities instance. For each installer format, the supported versions

shall be represented using the following array properties of the associated CIM_SoftwareInstallationServiceCapabilities instance at the corresponding index:

- SupportedExtendedResourceTypesMajorVersions
- SupportedExtendedResourceTypesMinorVersions
- SupportedExtendedResourceTypesRevisionNumbers
- SupportedExtendedResourceTypesBuildNumbers

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

7.3.3 CIM_ServiceAffectsElement

When an instance of CIM_SoftwareInstallationService is compatible with a Software Identity that is available for installation, an instance of CIM_ServiceAffectsElement shall associate the CIM_SoftwareInstallationService instance with the Software Identity.

7.4 Representing the Relationship between the Managed Element and the Software Installation Service

When an instance of CIM_SoftwareInstallationService is capable of installing or updating software on a managed element, an instance of CIM_ServiceAffectsElement may associate the CIM_SoftwareInstallationService instance with the CIM_ManagedElement instance. When an instance of CIM_SoftwareInstallationService is capable of installing or updating software on an instance of CIM_ComputerSystem or a managed element scoped to the CIM_ComputerSystem instance, an instance of CIM_ServiceAffectsElement shall associate the CIM_SoftwareInstallationService with the CIM_ComputerSystem instance.

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

The location of a Software Identity may be advertised. This behavior is optional. When this behavior is implemented, it shall be done according to the implementation requirements of the *Software Inventory Profile*.

7.6 Version Comparison Algorithm

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

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

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

   Else the instance with the higher value of the property that represents the major version shall be the instance with the higher version.

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

   Else the instance with the higher value of the property that represents the minor version shall be the instance with the higher version.

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

   Else the instance with the higher value of the property that represents the revision number shall be the instance with the higher version.

4) If the properties that represent the build number of the two instances are equal, the two instances shall have equal versions.

   Else the instance with the higher value of the property that represents the build number shall be the instance with the higher version.

8 Methods

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

8.1 CIM_SoftwareInstallationService.CheckSoftwareIdentity( )

The CIM_SoftwareInstallationService.CheckSoftwareIdentity( ) method allows a client application to determine whether a Software Identity can be installed or updated on a managed element. It also allows the client to determine some other characteristics of the installation, such as whether installation will require a reboot. When the Target parameter and the Collection parameter are both non-Null, the method shall return the value 2 (Error Occurred). When the Target parameter and the Collection parameters are Null, the method shall return the value 2 (Error Occurred).

Detailed requirements of the CheckSoftwareIdentity( ) method are specified in Table 2 and Table 3.

No standard messages are defined.

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

Table 3 – CIM_SoftwareInstallationService.CheckSoftwareIdentity( ) Method: Parameters

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

8.1.1 Source

The Source parameter is a reference to the Software Identity that represents the software to be checked for installation or update on a managed element. The method shall return the value 2 (Error Occurred) when this parameter is Null.

8.1.2 Target

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

When this parameter is non-Null and the method can determine that the Software Identity can be installed on the managed element represented by the Target parameter, the method shall return the value 0. When this parameter is non-Null and the method can determine that the Software Identity cannot be installed on the managed element represented by the Target parameter, the method shall return the value 2 (Error Occurred).

8.1.3 Collection

The Collection parameter is a reference to the instance of CIM_SystemSpecificCollection that represents the collection to which the Software Identity will be added. When this parameter is non-Null and the CanAddToCollection property of the associated instance of CIM_SoftwareInstallationServiceCapabilities is FALSE, the method shall return the value 2 (Error Occurred).

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

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

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

8.2 CIM_SoftwareInstallationService.InstallFromSoftwareIdentity()

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

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

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

When the Target parameter is non-Null and the Collection parameter is Null, the method will install or update the Software Identity on the managed element. When the Collection parameter is non-Null and the Target parameter is Null, the method will add the Software Identity to the collection.

Detailed requirements of the InstallFromSoftwareIdentity( ) method are specified in Table 4 and Table 5.

No standard messages are defined.

Table 4 – CIM_SoftwareInstallationService.InstallFromSoftwareIdentity( ) 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

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.

8.2.1 Job

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

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

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

8.2.2 InstallOptions

The InstallOptions array parameter is used to input installation options to the InstallFromSoftwareIdentity( ) method, which allows the client to control the installation procedure. When this parameter is Null, the installation options used are implementation specific. The method shall return the value 2 (Error Occurred) when this parameter contains an installation option that is not listed in the SupportedInstallOptions property of the associated instance of CIM_SoftwareInstallationServiceCapabilities.

8.2.3 InstallOptionsValues

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

If an installation option in the InstallOptions array parameter requires a value, and a Null value is specified in the InstallOptionsValues array parameter at the corresponding index, the method shall return the value 2 (Error Occurred).

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

8.2.4 Source

The Source parameter is a reference to the Software Identity that represents the software to be installed or updated on a managed element. The method shall return the value 2 (Error Occurred) when this parameter is Null.

8.2.5 Target

The Target parameter is a reference to the instance of CIM_ManagedElement that represents a managed element on which the Software Identity is intended to be installed or updated. If the Target parameter is a reference to the Scoping Instance and the software is applicable to a single managed element in its scope, including itself, the method shall install the software on the managed element. If the Target parameter is a reference to the Scoping Instance and the software is applicable to more than one managed element in its scope, the method may install the software on one, all, or none of the managed elements. The behavior is implementation specific.

When this parameter references an instance of CIM_SoftwareIdentity that represents a Software Bundle, the method shall return the value 0 only if all the aggregated instances of Software Identity were successfully installed. If at least one instance of Software Identity was not installed successfully, the method shall return the value 2 (Error Occurred).

When this parameter is non-Null and the method can install or update the Software Identity on the managed element represented by the Target parameter, the method shall return the value 0. When this parameter is non-Null and the method cannot install or update the Software Identity on the managed element represented by the Target parameter, the method shall return the value 2 (Error Occurred).

8.2.6 Collection

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

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

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

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

8.3 CIM_SoftwareInstallationService.InstallFromByteStream( )

The CIM_SoftwareInstallationService.InstallFromByteStream( ) method allows a client application to download or copy a series of bytes that contain a software image to a managed element. When this method is supported, at least one of SupportedAsynchronousActions property or SupportedSynchronousActions property of the associated instance of CIM_SoftwareInstallationServiceCapabilities shall contain the value 4 (Install From ByteStream).Table 7

466 No standard messages are defined.

467 **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

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

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

469 ### 8.3.1 Job

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

472 The method shall not return the Job output parameter when the SupportedAsynchronousActions property
473 of the associated instance of CIM_SoftwareInstallationServiceCapabilities does not contain the value 4
474 (Install From Byte Stream).

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

477 ### 8.3.2 InstallOptions

478 The InstallOptions array parameter is used to input installation options to the
479 InstallFromSoftwareIdentity( ) method, which allows the client to control the installation procedure. When
480 this parameter is Null, the installation options used are implementation specific and no error shall be
481 returned. The method shall return the value 2 (Error Occurred) when this parameter contains an
482 installation option that is not listed in the SupportedInstallOptions property of the associated instance of
483 CIM_SoftwareInstallationServiceCapabilities.

484 ### 8.3.3 InstallOptionsValues

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

487 If an installation option in the InstallOptions array parameter requires a value, and a Null value is specified
488 in the InstallOptionsValues array parameter at the corresponding index, the method shall return the value
489 2 (Error Occurred).

490 If an installation option in the InstallOptions array parameter is required not to have a value, and a non-
491 Null value is specified in the InstallOptionsValues array parameter at the corresponding index, the method
492 shall return the value 2 (Error Occurred).

8.3.4 Image

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

8.3.5 Target

The Target parameter is a reference to the instance of CIM_ManagedElement that represents a managed element on which the Software Identity is intended to be installed or updated. If the Target parameter is a reference to the Scoping Instance and the software is applicable to a single managed element in its scope, including itself, the method shall install the software on the managed element. If the Target parameter is a reference to the Scoping Instance and the software is applicable to more than one managed element in its scope, the method may install the software on one, all, or none of the managed elements. The behavior is implementation specific.

When this parameter is Null, the method shall return the value 2 (Error Occurred).

8.4 CIM_SoftwareInstallationService.InstallFromURI( )

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

Detailed requirements of the InstallFromURI( ) method are specified in Table 8 and Table 9.

No standard messages are defined.

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

Table 9 – CIM_SoftwareInstallationService.InstallFromURI( ) Method: Parameters

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

8.4.1 Job

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

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

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

8.4.2 InstallOptions

The InstallOptions array parameter is used to input installation options to the InstallFromSoftwareIdentity( ) method, which allows the client to control the installation procedure. When this parameter is Null, the installation options used are implementation specific. The method shall return the value 2 (Error Occurred) when this parameter contains an installation option that is not listed in the SupportedInstallOptions property of the associated instance of CIM_SoftwareInstallationServiceCapabilities.

8.4.3 InstallOptionsValues

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

If an installation option in the InstallOptions array parameter requires a value, and a Null value is specified in the InstallOptionsValues array parameter at the corresponding index, the method shall return the value 2 (Error Occurred).

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

8.4.4 URI

The URI parameter is used to specify the URI information of the software to be installed on the managed element. When the URI is Null or not well formed according to RFC 2396, the InstallFromURI( ) method shall return the value 2 (Error Occurred). When the URI scheme of this parameter is not present in the SupportedURISchemes property of the associated instance of CIM_SoftwareInstallationServiceCapabilities, the method shall return the value 2 (Error Occurred).

8.4.5 Target

The Target parameter is a reference to the instance of CIM_ManagedElement that represents a managed element on which the Software Identity is intended to be installed or updated. If the Target parameter is a reference to the Scoping Instance and the software is applicable to a single managed element in its scope, including itself, the method shall install the software on the managed element. If the Target parameter is a reference to the Scoping Instance and the software is applicable to more than one managed element in its scope, the method may install the software on one, all, or none of the managed elements. The behavior is implementation specific.

When this parameter is Null, the method shall return the value 2 (Error Occurred).

8.5 Profile Conventions for Operations

Support for operations for each profile class (including associations) is specified in the following subclauses. Each subclause includes either a statement “All operations in the default list in section 8.5 are supported as described by DSP0200 version 1.2” or a table listing all of the operations that are not supported by this profile or where the profile requires behavior other than that described by DSP0200 version 1.2.

559 The default list of operations is as follows:

560 • GetInstance
561 • Associators
562 • AssociatorNames
563 • References
564 • ReferenceNames
565 • EnumerateInstances
566 • EnumerateInstanceNames

567 A compliant implementation shall support all of the operations in the default list for each class, unless the
568 "Requirement" column states something other than *Mandatory*.

569 8.6 CIM_SoftwareInstallationService

570 All operations in the default list in section 8.5 are supported as described by DSP0200 version 1.2.

571 8.7 CIM_HostedService

572 Table 10 lists operations that either have special requirements beyond those from DSP0200 version 1.2
573 or shall not be supported.

574 **Table 10 – Operations: CIM_HostedService**

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

575 8.8 CIM_SoftwareInstallationServiceCapabilities

576 All operations in the default list in section 8.5 are supported as described by DSP0200 version 1.2.

577 8.9 CIM_ElementCapabilities

578 Table 11 lists operations that either have special requirements beyond those from DSP0200 version 1.2
579 or shall not be supported.

580 **Table 11 – Operations: CIM_ElementCapabilities**

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

8.10 CIM_ServiceAffectsElement

Table 12 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 12 – Operations: CIM_ServiceAffectsElement

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

9 Use Cases

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

9.1 Object Diagrams

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

9.1.1 Registered Profile

Figure 2 represents a possible instantiation of the *Software Update Profile*. The Central Instance, swinst1, has a CIM_HostedService association to the Scoping Instance, system1. Profile registration information is represented by profile1. Following the CIM_ElementConformsToProfile association from the Central Instance to profile1, the client can retrieve information such as the version of the current *Software Update Profile* implementation.

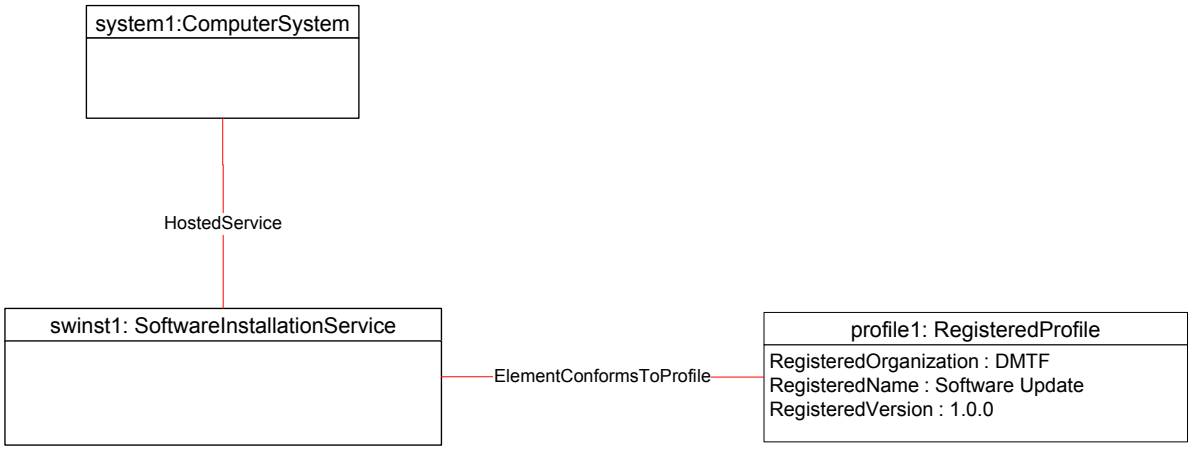


Figure 2 – Registered Profile

9.1.2 Representing Available Software, Managed Elements, Software Installation Services, and Their Relationships

Figure 3 represents a possible instantiation of the *Software Update Profile*. The optional behavior of "Representing Available Software" from the *Software Inventory Profile* has been implemented. The managed system, system1, hosts a collection, "Available Software" and an installation service, swinst1. The firmware image applicable to the Network PCIController (pcictrl1) is represented by the Software Identity (swid2), which is a member of the "Available Software" collection. A CIM_ElementSoftwareIdentity association is shown between the pcictrl1 and swid2.

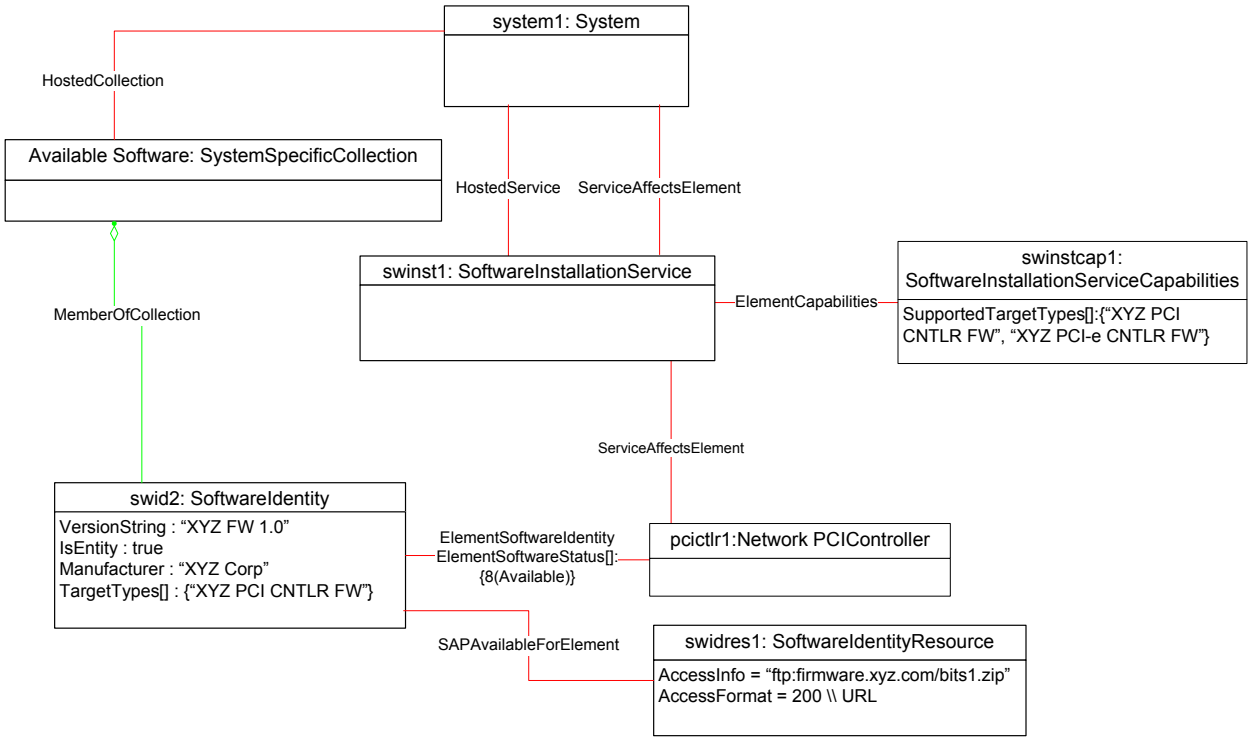


Figure 3 – Representing Available Software

The capabilities of swinst1 are represented by the instance of CIM_SoftwareInstallationServiceCapabilities (swinstcap1). The TargetTypes property on swid2 has a value that matches one of the values in the SupportedTargetTypes property of swinstcap1; therefore, swid2 is compatible with swinst1, and swid2 can be installed or updated using swinst1.

The CIM_ServiceAffectsElement association between pcictrl1 and swinst1 indicates that swinst1 can provide a software installation or update service to pcictrl1. The CIM_ServiceAffectsElement association between system1 and swinst1 indicates that swinst1 can provide a software installation or update service to system1 and or components installed in system1.

9.1.3 Representing a Software Identity with Installation Dependencies

Figure 4 represents a possible instantiation of the *Software Update Profile*. The optional behavior of "Representing Installation Dependencies" from the *Software Inventory Profile* has been implemented. The Software Identity, swid1, is a member of the "Available Software" collection and has Installation Dependencies on other Software Identities, swid2 and swid3. A copy of swid2 is available, and so the IsEntity property of swid2 is TRUE. A copy of swid3 is not available or installed, and so the IsEntity property of swid3 is FALSE.

swid2 and then swid3 need to be installed before swid1 can be installed. The object diagram does not show the instances of CIM_SoftwareInstallationService that are compatible with swid1 and swid2.

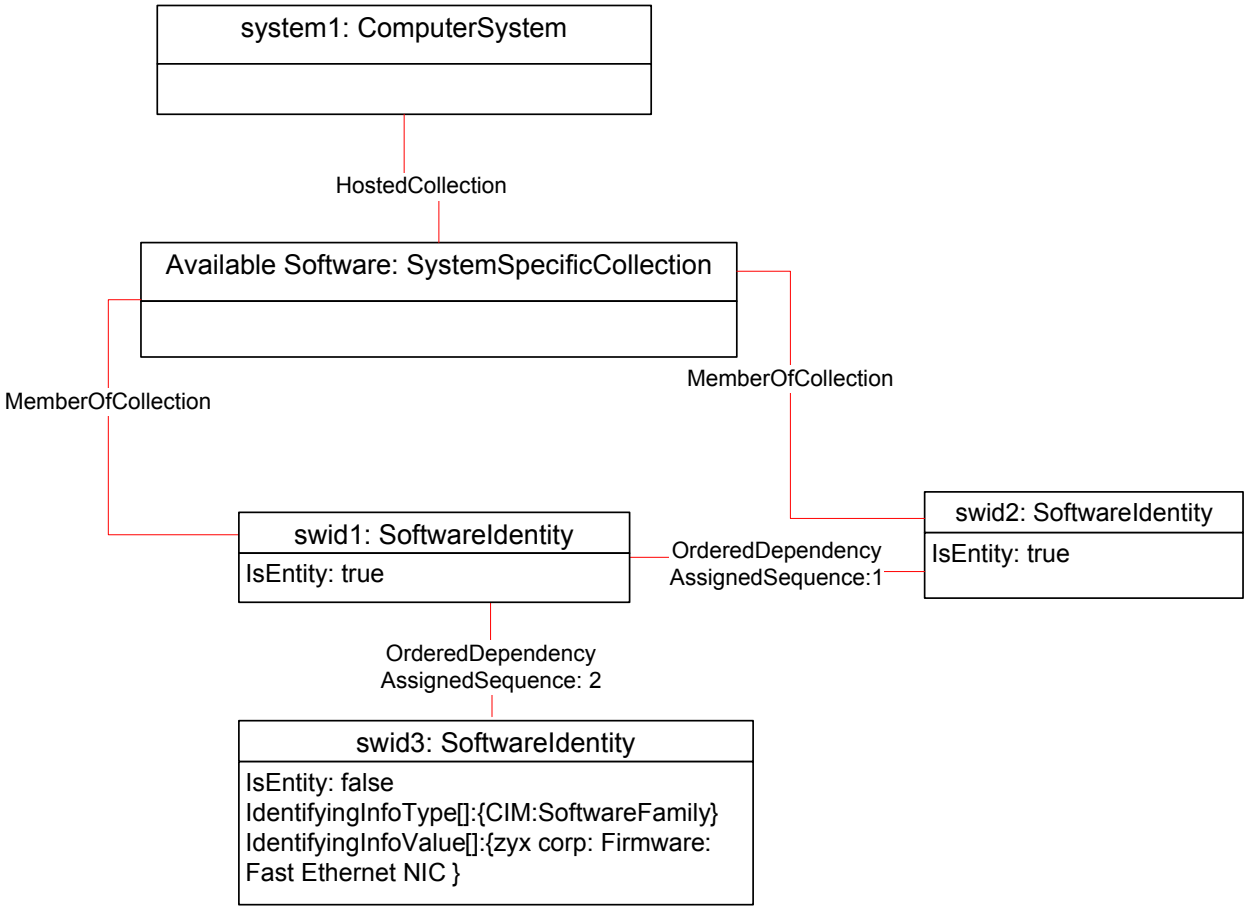


Figure 4 – Representing Installation Dependencies

9.1.4 Representing a Software Identity with an Installation Dependency That Is Installed

Figure 5 represents a possible instantiation of the *Software Update Profile*. The optional behavior of "Representing Installation Dependencies" from the *Software Inventory Profile* has been implemented. The Software Identity, swid1, is a member of the "Available Software" collection and has Installation Dependencies on another Software Identity, swid3.

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

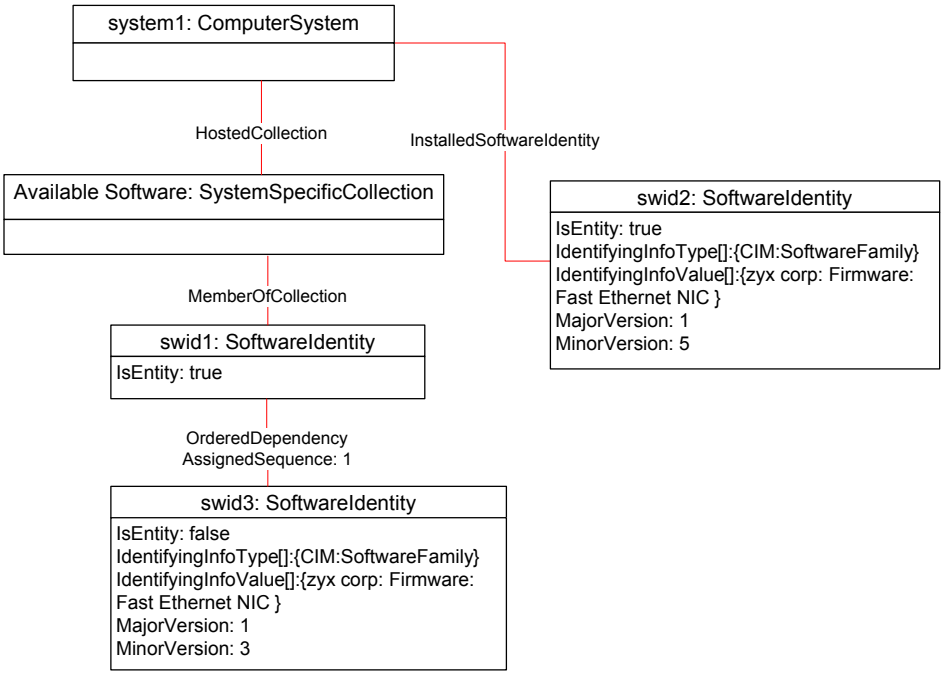


Figure 5 – Representing Installation Dependencies That Are Installed

9.1.5 Representing Software Bundles

Figure 6 represents a possible instantiation of the *Software Update Profile*. 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 to install swbun1.

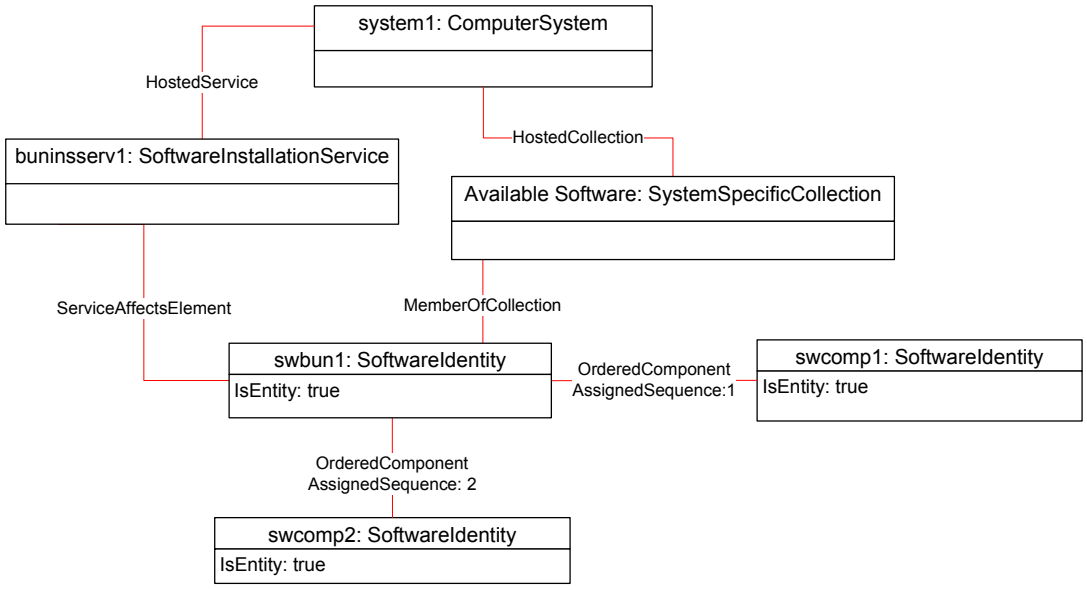
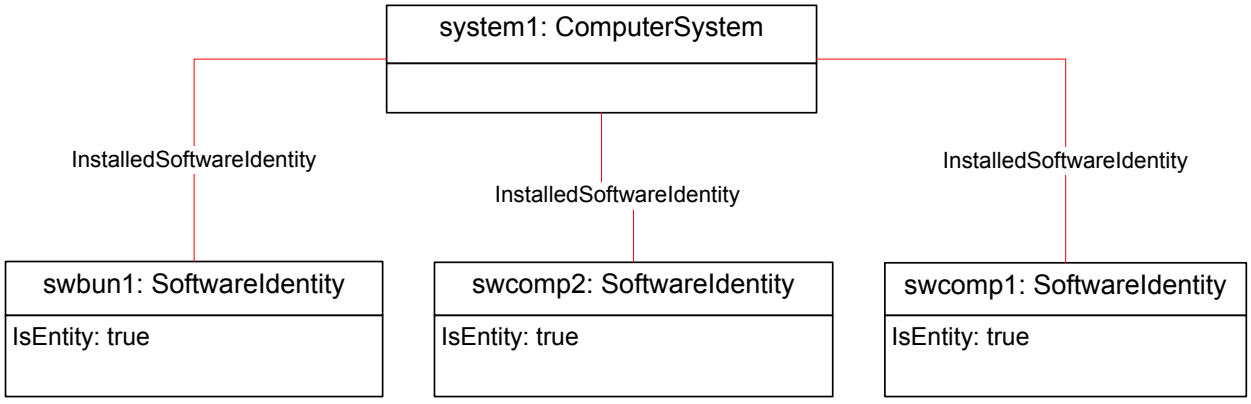


Figure 6 – Representing a Software Bundle

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



649

650 **Figure 7 – Representing a Software Bundle That Is Installed**

651 ### 9.1.6 Representing a Software Bundle That Is Not a Direct Target of Installation

652 Figure 8 represents a possible instantiation of the *Software Update Profile*. The optional behavior of
653 "Representing a Software Bundle" from the *Software Inventory Profile* has been implemented. The
654 Software Bundle, swbun2, has the aggregated instances of swcomp3 and swcomp4. The Software
655 Installation Service, insserv3, is compatible with swcomp3 and can be used for installing it. The Software
656 Installation Service, insserv4, is compatible with swcomp4 and can be used for installing it. swbun2
657 cannot be a direct target of installation as there is no compatible Software Installation Service.

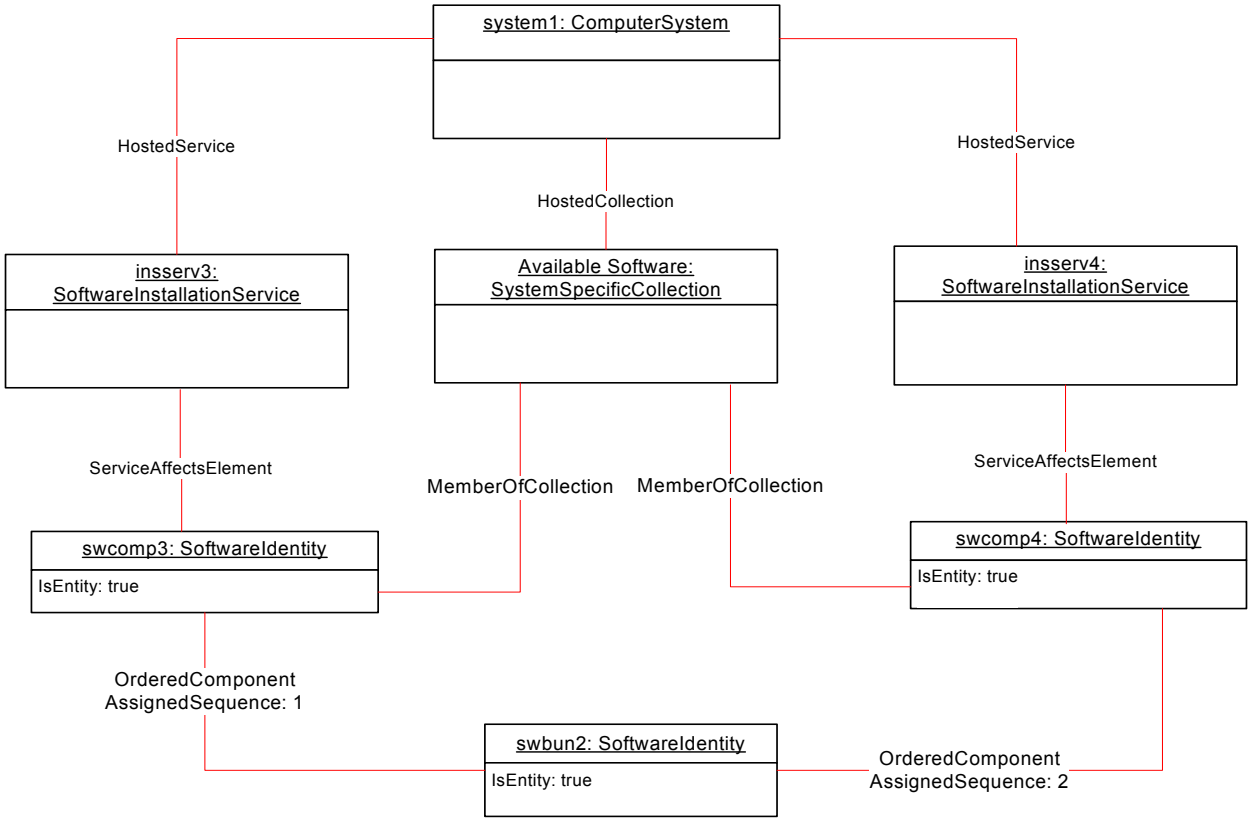


Figure 8 – Representing a Bundle That Is Not a Direct Target of Installation

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

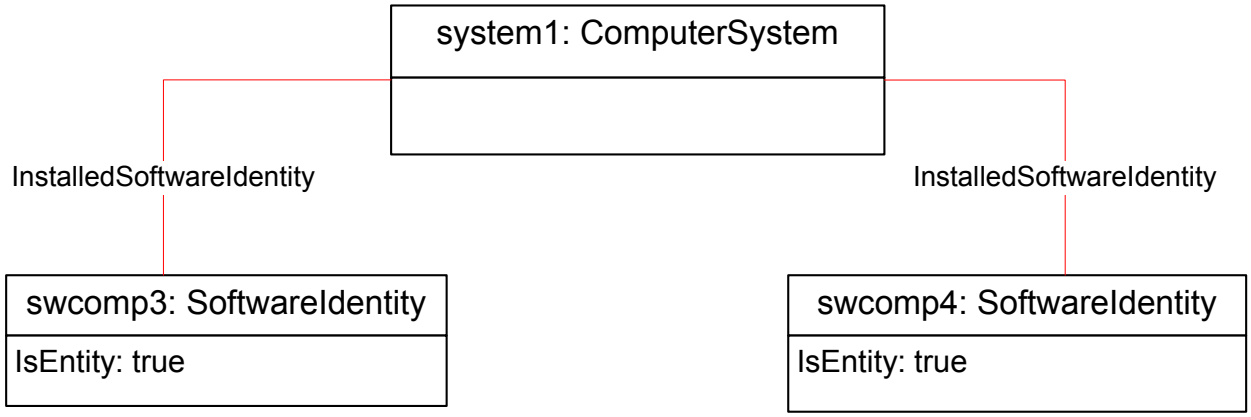


Figure 9 – Installed Components of a Bundle Which Is Not a Direct Target of Installation

9.2 Find the Software Installation Services Compatible with a Software Identity

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

1) For the given Software Identity, select the CIM_SoftwareInstallationService instances that are associated with it through the CIM_ServiceAffectsElement association.

2) Select the instances of CIM_SoftwareInstallationService with at least one value in the SupportedTargetTypes property of the associated CIM_SoftwareInstallationServiceCapabilities instance that is equal to at least one value in the TargetTypes property of the given Software Identity.

3) Select the instances of CIM_SoftwareInstallationService with at least one value in the SupportedExtendedResourceTypes property that is equal to the ExtendedResourceType property of the given Software Identity, and for which the supported version of the installer format is equal to or higher than the version of the installer format supported by the Software Identity (see section 7.3.2).

The instances of CIM_SoftwareInstallationService from steps 1, 2, and 3 represent the Software Installation Services that are compatible with the Software Identity.

9.3 Determine Whether Installing a Software Identity Requires a Reboot

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

1) Find the Software Installation Service compatible with the Software Identity by following the steps in section 9.2.

2) Invoke the CheckSoftwareIdentity( ) method on the CIM_SoftwareInstallationService instance with the given Software Identity as the Source parameter. After successful execution of the method, if the InstallCharacteristics parameter contains the value 7 (No Reboot Required), no reboot is required after installing the Software identity. If the parameter contains the value 6 (Manual Reboot Required), a reboot has to be performed to complete the installation. If the parameter contains a value other than 6 or 7, no information about the requirements for the reboot can be determined.

9.4 Find Software Available for Installation on a Managed Element When CIM_ElementSoftwareIdentity Exists

Assuming that the Software Identities compatible with a managed element are associated with the managed element through an instance of CIM_ElementSoftwareIdentity, a client can find the Software Identities available for installation that are applicable to a managed element as follows:

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

9.5 Find Software Available for Installation on a Managed Element When CIM_ElementSoftwareIdentity Does Not Exist

When the Software Identities compatible with a managed element are not associated with the managed element through an instance of CIM_ElementSoftwareIdentity, a client can find the Software Identities available for installation that are applicable to a managed element by using the following steps:

1) Starting at the Scoping Instance, find all the Available Software by following the steps in section 9.5 of the *Software Inventory Profile*.

2) Find the instances of CIM_SoftwareInstallationService that can provide installation or update service to the managed element by following the steps in section 9.7.

3) For each Software Identity identified in step 1, find the compatible Software Installation Services by following the steps in section 9.2.

4) For each Software Installation Service that is also in the set of Software Installation Services found in step 2, invoke the CheckSoftwareIdentity( ) method using the appropriate parameters.

If the method returns the value 0, the Software Identity can be installed on the managed element.

9.6 Find Software Available for Installation on a Component

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

1) Starting at the instance of CIM_ComputerSystem that represents the system to which the component belongs, find all the Available Software following the steps in section 9.5 of the *Software Inventory Profile*.

2) Select the Software Identities from step 1 where the property values match the required values for the component.

9.7 Find Software Installation Services That Can Install or Update Software on a Managed Element

A client can find the Software Installation Services that can install or update software on a managed element by using the following steps:

1) Starting from the managed element, select the instances of CIM_SoftwareInstallationService that are associated through the CIM_ServiceAffectsElement association.

2) Select the instances of CIM_SoftwareInstallationService that are associated with the Scoping Instance through the CIM_ServiceAffectsElement association.

The instances of CIM_SoftwareInstallationService from steps 1 and 2 represent the Software Installation Services that could provide installation or update service to the managed element.

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

A client can install or update software on a managed element with a Software Identity by using the following steps:

1) Find all the Software Identities that are applicable to the managed element by following the steps in section 9.4 and section 9.5. Select the Software Identity of interest.

2) Find the instances of CIM_SoftwareInstallationService that can provide installation or update service to the managed element by following the steps in section 9.7.

3) For the Software Identity from step 1, find the compatible Software Installation Services by following the steps in section 9.2.

4) For each Software Installation Service that is also in the set of Software Installation Services found in step 2, invoke the CheckSoftwareIdentity( ) method with the appropriate parameters.

5) If the method returns the value 0, invoke the InstallFromSoftwareIdentity( ) method on the instance of CIM_SoftwareInstallationService with the appropriate parameters.

6) Else If the Software Identity from step 1 is referenced by an instance of CIM_SAPAvailableForElement, and if either the SupportedAsynchronousActions property or the SupportedSynchronousActions property of the associated instance of CIM_SoftwareInstallationServiceCapabilities contains the value 5 (Install From URI) then,

    a) Starting from the Software Identity, select the instance of CIM_SoftwareIdentityResource through the CIM_SAPAvailableForElement association.

752 b) Extract the URI information by using the instance of CIM_SoftwareIdentityResource, and
753 invoke the InstallFromURI( ) method with the appropriate parameters.

754 9.9 Install from Software Identity When the Managed Element Is Not Modeled

755 A client can install or update software represented as a Software Identity on a component that is not
756 modeled as a managed element by using the following steps:

757 1) Find all the Software Identities that are applicable to the component by following the steps in
758 section 9.6. Select the Software Identity of interest.

759 2) Find the instances of CIM_SoftwareInstallationService that can provide installation or update
760 service to the instance of CIM_ComputerSystem that represents the system to which the
761 component belongs, following the steps in section 9.7.

762 3) For the Software Identity from step 1, find the compatible Software Installation Services by
763 following the steps in section 9.1.6.

764 4) For each Software Installation Service from step 3, which is also in the set of Software
765 Installation Services found in step 2,

766 o Invoke the InstallFromSoftwareIdentity( ) method on the instance of
767 CIM_SoftwareInstallationService with the Target parameter as the Scoping Instance.

768 o If the method returns the value 0, the Software Identity was successfully installed.

769 5) If the Software Identity from step 1 was not installed and if the Software Identity is referenced by
770 an instance of CIM_SAPAvailableForElement.and if either the SupportedAsynchronousActions
771 property or the SupportedSynchronousActions property of the associated instance of
772 CIM_SoftwareInstallationServiceCapabilities contains the value 5 (Install From URI) then

773 a) Starting from the Software Identity, select the instance of CIM_SoftwareIdentityResource
774 through the CIM_SAPAvailableForElement association.

775 b) Extract the URI information by using the instance of CIM_SoftwareIdentityResource, and
776 invoke the InstallFromURI( ) method with the appropriate parameters. If the method returns
777 the value 0, the Software Identity was successfully installed.

778 9.10 Install or Update Software on a Managed Element Using a URI

779 A client can install or update software on a managed element by using a URI that identifies the software
780 by using the following steps:

781 1) Find the instances of CIM_SoftwareInstallationService that can install or update software on the
782 managed element by following the steps in section 9.7.

783 2) Select an instance of CIM_SoftwareInstallationService with an associated instance of
784 CIM_SoftwareInstallationServiceCapabilities in which either the SupportedAsynchronousActions
785 property or the SupportedSynchronousActions property has a value equal to 5 (Install From
786 URI) and the SupportedURISchemes property contains the URI scheme of the URI.

787 3) Invoke the InstallFromURI( ) method on the instance of CIM_SoftwareInstallationService from
788 step 2 using the appropriate parameters.

789 9.11 Install from URI When the Managed Element Is Not Modeled

790 A client can install or update software on a component that is not modeled as a managed element by
791 using a URI that identifies the software by using the following steps:

792 1) Find the instances of CIM_SoftwareInstallationService that can provide installation or update
793 service to the instance of CIM_ComputerSystem that represents the system to which the
794 component belongs, following the steps in section 9.7.

2) Select an instance of CIM_SoftwareInstallationService with an associated instance of CIM_SoftwareInstallationServiceCapabilities in which either the SupportedAsynchronousActions property or the SupportedSynchronousActions property has a value equal to 5 (Install From URI) and the SupportedURISchemes property contains the URI scheme of the URI.

3) Invoke the InstallFromURI( ) method on the instance of CIM_SoftwareInstallationService from step 2 using the appropriate parameters.

9.12 Update Software on a Managed Element Using a Byte Stream

A client can install or update software on a managed element by transferring the image as a byte array by using the following steps:

1) Find the instances of CIM_SoftwareInstallationService that can install or update software on the managed element by following the steps in section 9.7.

2) Select an instance of CIM_SoftwareInstallationService with an associated instance of CIM_SoftwareInstallationServiceCapabilities in which either the SupportedAsynchronousActions property or the SupportedSynchronousActions property has a value equal to 4 (Install From Byte Stream).

3) Invoke the InstallFromByteStream( ) method on the instance of CIM_SoftwareInstallationService from step 2 using the appropriate parameters.

10 CIM Elements

Table 13 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 13. Sections 7 (“Implementation”) and 8 (“Methods”) may impose additional requirements on these elements.

Table 13 – CIM Elements: Software Update Profile

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

10.1 CIM_HostedService

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

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 *

10.2 CIM_SoftwareInstallationService

CIM_SoftwareInstallationService is used to represent a component that can be used to perform an installation or update of software on a managed element. Table 15 contains the requirements for elements of this class.

Table 15 – Class: CIM_SoftwareInstallationService

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

10.3 CIM_ElementCapabilities

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

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

10.4 CIM_SoftwareInstallationServiceCapabilities

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

Table 17 – Class: CIM_SoftwareInstallationServiceCapabilities

Elements	Requirement	Notes
InstanceID	Mandatory	Key
SupportedTargetTypes[ ]	Optional	See section 7.3.1.
SupportedExtendedResourceTypes[ ]	Optional	See section 7.3.2.
SupportedExtendedResourceTypesMajorVersions[ ]	Optional	See section 7.3.2.
SupportedExtendedResourceTypesMinorVersions[ ]	Optional	See section 7.3.2.
SupportedExtendedResourceTypesRevisionNumbers[ ]	Optional	See section 7.3.2.
SupportedExtendedResourceTypesBuildNumbers[ ]	Optional	See section 7.3.2.
SupportedInstallOptions[ ]	Mandatory	None
SupportedURISchemes[ ]	Conditional	See section 7.2.1.

10.5 CIM_ServiceAffectsElement—CIM_SoftwareIdentity Reference

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

Table 18 – Class: CIM_ServiceAffectsElement—CIM_SoftwareIdentity Reference

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 *

10.6 CIM_ServiceAffectsElement—CIM_ManagedElement Reference

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

Table 19 – Class: CIM_ServiceAffectsElement—CIM_ManagedElement Reference

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 *

10.7 CIM_SoftwareIdentity

CIM_SoftwareIdentity is defined by the *Software Inventory Profile*. The requirements denoted in Table 20 are in addition to those mandated by the *Software Inventory Profile*.

Table 20 – Class: CIM_SoftwareIdentity

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

10.8 CIM_RegisteredProfile

CIM_RegisteredProfile is defined by the *Profile Registration Profile*. The requirements denoted in Table 21 are in addition to those mandated by the *Profile Registration Profile*.

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).

ANNEX A
(informative)

Change Log

Version	Date	Author	Description

ANNEX B
(informative)

Acknowledgements

The authors wish to acknowledge the following people.

Editor:

- RadhaKrishna R. Dasari – Dell

Contributors:

- RadhaKrishna R. Dasari – Dell
- Jon Hass – Dell
- Khachatur Papanyan – Dell
- Marshal Savage – Dell
- Sudhir Shetty – Dell
- Jeff Hilland – HP
- Christina Shaw – HP
- Aaron Merkin – IBM
- Jeff Lynch – IBM
- Perry Vincent – Intel
- John Leung – Intel