Document Number: DSP1081

Date: 2012-08-21

Version: 1.0.0

Virtual System Migration Profile

Document Type: Specification

Document Status: DMTF Standard

Document Language: en-US

CONTENTS

Figures

Tables

Foreword

This profile - the *Virtual System Migration Profile* (DSP1081) - was prepared by the System Virtualization, Partitioning and Clustering Working Group of the DMTF.

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

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

- Editor:
  - Michael Johanssen – IBM
- Participants from the DMTF System Virtualization, Partitioning and Clustering Working Group:
  - Gareth Bestor – IBM
  - Jim Fehlig – Novell
  - Mark Hapner – Sun Microsystems, Inc.
  - Ron Goering – IBM
  - Steve Hand – Symantec Corporation
  - Daniel Hiltgen – VMware Inc.
  - Michael Johanssen – IBM
  - Larry Lamers – VMware Inc.
  - Andreas Maier - IBM
  - Aaron Merkin – IBM
  - John Parchem – Microsoft Corporation
  - Shishir Pardikar – Citrix Systems Inc.
  - Nihar Shah – Microsoft Corporation
  - David Simpson – IBM

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 shall be instantiated and manipulated to represent and manage the components described in this document. The target audience for this specification is implementers who are writing CIM-based providers or consumers of management interfaces that represent the components described in this document.

Document conventions

Typographical conventions

The following typographical conventions are used in this document:

- Document titles are marked in *italics*.
- ABNF rules are in `monospaced` font.

213

Virtual System Migration Profile

1 Scope

This profile is a component DMTF management profile that extends the management capabilities of the referencing profile by adding the support to manage the migration of virtual systems. The support includes functionality to initiate and control migration operations, and feasibility checks for a potential migration operation.

2 Normative references

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

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

DMTF DSP0200, *CIM Operations over HTTP 1.3*
http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

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

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

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

DMTF DSP1042, *System Virtualization Profile 1.0*
http://www.dmtf.org/standards/published_documents/DSP1042_1.0.pdf

DMTF DSP1052, *Computer System Profile 1.0*
http://www.dmtf.org/standards/published_documents/DSP1052_1.0.pdf

DMTF DSP1054, *Indications Profile 1.1*
http://www.dmtf.org/standards/published_documents/DSP1054_1.1.pdf

DMTF DSP1057, *Virtual System Profile 1.0*
http://www.dmtf.org/standards/published_documents/DSP1057_1.0.pdf

DMTF DSP1103, *Job Control Profile 1.0*
http://www.dmtf.org/standards/published_documents/DSP1103_1.0.pdf

DMTF DSP8026, *System Virtualization Message Registry* 1.0
http://schemas.dmtf.orgs/wbem/messageregistry/1/dsp8026_1.0.xml

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

3 Terms and definitions

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

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

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

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

The terms defined in DSP0004, DSP0223, and DSP1001 apply to this document. For the purposes of this document, the terms and definitions given in DMTF DSP1033 and DMTF DSP1001 also apply. The following additional terms are used in this document.

3.1
conditional
indicates requirements strictly to be followed in order to conform to the document and from which no deviation is permitted when the specified conditions are met

3.2
mandatory
indicates requirements strictly to be followed in order to conform to the document and from which no deviation is permitted

3.3
optional
indicates a course of action permissible within the limits of the document

3.4
referencing profile
indicates a profile that owns the definition of this class and can include a reference to this profile in its “Related Profiles” table

3.5
unspecified
indicates that this profile does not define any constraints for the referenced CIM element

3.6
implementation
set of CIM providers that realize the classes specified by this profile

3.7
client
application that exploits facilities specified by this profile

3.8
source virtualization platform
the source of the virtual system being acted upon

3.9
subject virtual system
the virtual system that is being acted upon

3.10
target virtualization platform
the destination of the virtual system being acted upon

3.11
this profile
this DMTF management profile – the *Virtual System Migration Profile*

3.12
virtual system migration
process of moving a virtual system from a source virtualization platform to a target virtualization platform; for details, see 6.2.

3.13
virtual system migration service
service that provides virtual system migration facilities as defined by this standard; for details, see 7.1.

3.14
virtual computer system
virtual system
concept of virtualization as applied to a computer system

Other common industry terms are *virtual machine*, *hosted computer*, *child partition*, *logical partition*, *domain*, *guest*, or *container*.

3.15
virtual system state
state of a virtual system; for details, see .

3.16
virtualization platform
virtualizing infrastructure provided by a host system enabling the provisioning and deployment of virtual systems

3.17
virtual system migration task
task that performs a particular virtual system migration

3.18
virtual system migration type
type of virtual system migration; for details, see 6.2.

4 Symbols and abbreviated terms

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

4.1
CIM
Common Information Model

4.2
CIMOM
CIM object manager

4.3
RASD
CIM_ResourceAllocationSettingData

4.4
SLP
Service Location Protocol

4.5
VS
virtual system

4.6
VSSD
CIM_VirtualSystemSettingData

4.7
VS_MIGRATION_METHOD_CALL
the event that marks either the begin or the completion of a virtual system migration method call

4.8
VS_MIGRATION_JOB_CREATE
the event that marks a the creation of an instance of the CIM_ConcreteJob class representing a virtual system migration task

4.9
VS_MIGRATION_JOB_CHANGE
the event that marks a change an instance of the CIM_ConcreteJob class representing a virtual system migration task

4.10
VS_MIGRATION_JOB_DELETE
the event that marks the deletion of an instance of the CIM_ConcreteJob class representing a virtual system migration task

5 Synopsis

Profile Name: *Virtual System Migration Profile*

Version: 1.0.0

Organization: DMTF

CIM Schema Version: 2.33

Central Class: CIM_VirtualSystemMigrationService

Scoping Class: CIM_System

371 This profile is a component profile that defines the minimum object model needed to provide for the
372 migration of virtual systems.

373 Table 1 lists DMTF management profiles that this profile depends on.

374 **Table 1 – Related profiles**

Profile Name	Organization	Version	Relationship	Description
Profile Registration	DMTF	*1.0*	Mandatory	The profile that specifies registered profiles.
Indications	DMTF	*1.0*	Conditional [1]	The profile that specifies indications.
Job Control	DMTF	*1.0*	Optional	The profile that specifies job control.

[1] Condition: The implementation of DMTF DSP1054 (*Indications Profile*) is required in the scope of the referencing profile if the indications defined in this profile are implemented.

375 # 6 Description

376 The *Virtual System Migration Profile* specifies the modeling of virtual system migration.

377 ## 6.1 Virtual system migration class schema

378 Figure 1 shows the class schema of this profile. It outlines the elements that are owned or specialized by
379 this profile, as well as the dependency relationships between elements of this profile and other profiles.
380 For simplicity in diagrams the prefix CIM_ has been removed from class and association names.
381

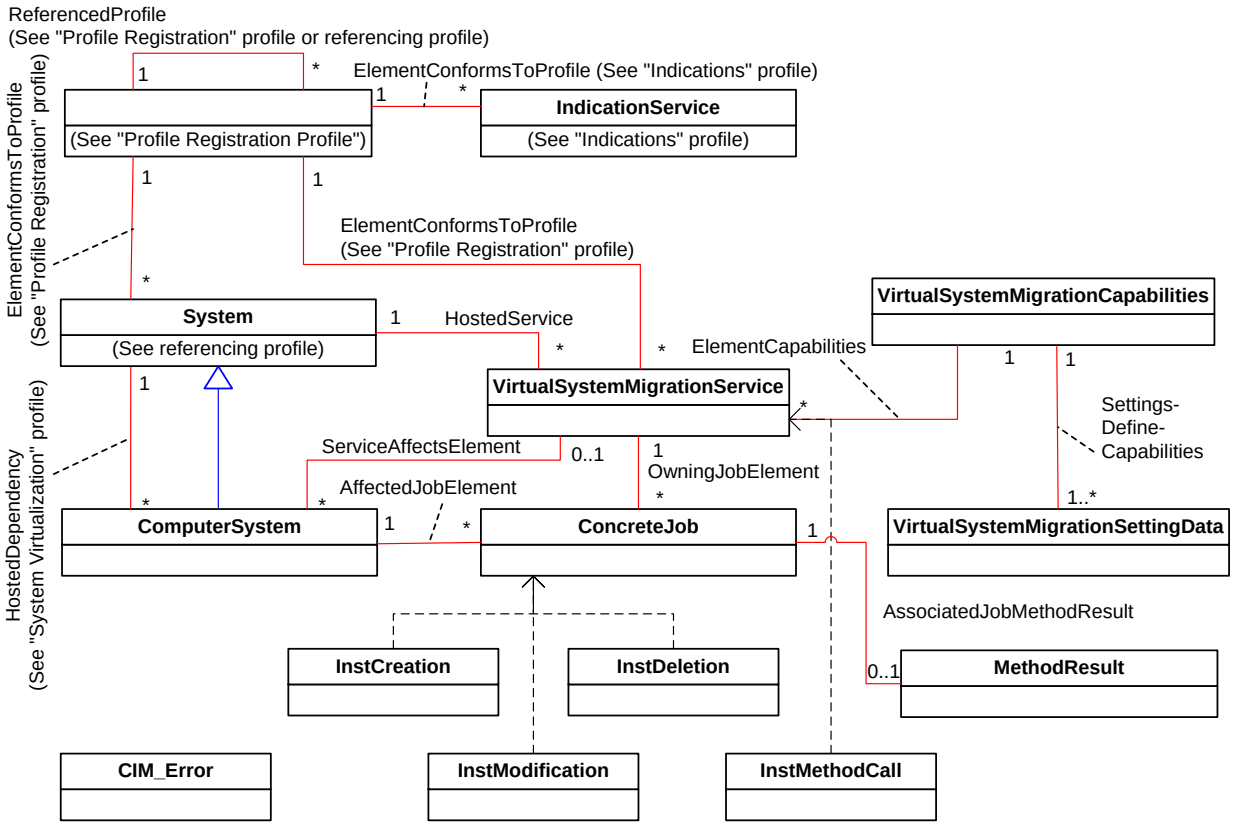

Figure 1 – Virtual System Migration Profile: Profile class diagram

This profile specifies the use of the following classes and associations:

- the CIM_VirtualSystemMigrationService class modeling the virtual system migration service
- the CIM_HostedService association modeling the hosting dependency between an instance of the CIM_VirtualSystemMigrationService class representing a virtual system migration service and the instance of the CIM_System class representing a host system
- the CIM_ServiceAffectsElement association modeling the relationship between an instance of the CIM_VirtualSystemMigrationService class representing a virtual system migration service and an instance of the CIM_ComputerSystem class representing a virtual system
- the CIM_VirtualSystemMigrationCapabilities class modeling the capabilities of a virtual system migration service
- the CIM_VirtualSystemMigrationSettingData class
  - modeling transient parameterization information in invocations of methods of the CIM_VirtualSystemMigrationService class
  - modelling capabilities expressing defaults and limitations applicable to the use of class instances as parameter
- the CIM_SettingsDefineCapabilities association modeling the relationship between virtual system migration capabilities and defaults or limitations that apply to migration methods

- the CIM_ElementCapabilities association modeling the relationship between an instance of the CIM_VirtualSystemMigrationService class representing a virtual system migration service and and instance of the the CIM_VirtualSystemMigrationCapabilities class representing its capabilities
- the CIM_ConcreteJob class modeling a virtual system migration task, the CIM_OwningJobElement association modeling the relationship between a virtual system migration service and a virtual system migration task, and the CIM_AffectedJobElement association modeling the relationship between a virtual system migration task and the affected virtual system

In general, any mention of a class in this document means the class itself or its subclasses. For example, a statement such as “an instance of the CIM_VirtualSystemMigrationSettingData class” implies an instance of the CIM_VirtualSystemMigrationSettingData class or of a subclass of the CIM_VirtualSystemMigrationSettingData class.

6.2 Virtual system migration concepts

Virtual system migration is the process of moving a virtual system from a source virtualization platform to a target virtualization platform, such that after migration the virtual system can be used at the target virtualization platform just as it was used at the source virtualization platform.

Three types of migration are distinguished:

- Static migration
- Dynamic migration
- Live migration

The implementation of a particular virtual system migration type depends on both the source and the target virtualization platform. Consequently the set of implemented virtual system migration types is not a static property of either source or target virtualization platform; instead the implementation of a particular virtual system migration type needs to be determined dynamically by means of check methods that are modeled as part of the virtual system migration service.

6.2.1 Static migration

Static migration applies primarily to the virtual system definition. Subject virtual systems are considered to be in the in the “Defined” virtual system state, that is, they are not instantiated and not performing work.

Static migration comprises activities such as

- validating that source and target virtualization platform are compatible
- validating that external connections, such as access to network and storage resources defined for the virtual system at the source virtualization platform can be identically redefined at the target virtualization platform
- redefining the virtual system at the target virtualization platform, including definition of external connections
- moving persistent resources like the content of locally defined virtual disks from the source virtualization platform to the target virtualization platform
- removing the virtual system from the source virtualization platform

6.2.2 Dynamic migration

Dynamic migration applies to both the virtual system definition and to the virtual system instance. Subject virtual systems may be in the “Defined”, “Active”, “Paused” or “Suspended” virtual system state. The "Active" virtual system state implies that the virtual system is instantiated and actively performs tasks; but

while the virtual system may be active when a dynamic migration process is initiated, the virtual system (including all of its virtual resources) is prevented from performing any activities for the complete duration of the migration process.

Dynamic migration comprises of the activities required for static migration, but in addition requires activities affecting the virtual system instance, like

- re-instantiating the virtual system at the target virtualization platform
- establishing external connections that were in effect for the virtual system at the source virtualization platform identically at the target virtualization platform
- moving volatile resources like the content of virtual memory from the source virtualization platform to the target virtualization platform

6.2.3 Live migration

Live migration applies to virtual system definition and virtual system instance. Subject virtual systems may be in the “Defined”, “Active”, “Paused” or “Suspended” virtual system states, that is, they may be instantiated and may continue performing work while the migration process is in progress.

Live migration comprises of activities required for static and for dynamic migration. It requires closer coordination between source and target virtualization platform such that state changes resulting from activities performed by the virtual system while the migration is in progress are captured. The integrity of the virtual system, its components, and its external connections is maintained over the whole migration process. External clients of software executed within the virtual system do not notice the ongoing migration process aside from performance impacts.

7 Implementation

This clause details the requirements related to classes and their properties for implementations of this profile. The CIM Schema descriptions for any referenced element and its sub-elements apply.

The list of all methods covered by this profile is in clause 8. The list of all properties covered by this profile is in clause 10.

In references to CIM Schema properties that enumerate values, the numeric value is normative and the descriptive text following it in parenthesis is informational. For example, in the statement “If an instance of the CIM_VirtualSystemMigrationCapabilities class contains the value 3 (MigrateVirtualSystemToSystem) in an element of the SynchronousMethodsSupported[ ] array property”, the “value 3” is normative text and “(MigrateVirtualSystemToSystem)” is descriptive text.

7.1 Virtual system migration service

The CIM_VirtualSystemMigrationService class is used to model virtual system migration services. Each virtual system migration service shall be represented by one instance of the CIM_VirtualSystemMigrationService class. A virtual system migration service is required to provide functionality for synchronous or asynchronous virtual system migration or both, such that this functionality can be exposed by the methods of the CIM_VirtualSystemMigrationService class.

7.2 Virtual system migration capabilities

This subclause details the requirements related to representing the capabilities of a virtual system migration service.

7.2.1 General

The CIM_VirtualSystemMigrationCapabilities class is used to model the capabilities of virtual migration services. The capabilities of a virtual system migration service shall be represented by an instance of the CIM_VirtualSystemMigrationCapabilities class. That instance shall be associated to the instance of the CIM_VirtualSystemMigrationService class representing the virtual system migration service (see 7.1) through an instance of the CIM_ElementCapabilities association.

7.2.2 Advertisement of method implementations

Values defined in the ValueMap qualifier of the SynchronousMethodsSupported[ ] and AsynchronousMethodsSupported[ ] array properties of the CIM_VirtualSystemMigrationCapabilities class shall designate corresponding methods of the CIM_VirtualSystemMigrationService class, as follows:

- The value 2 (MigrateVirtualSystemToHostSupported) shall designate the MigrateVirtualSystemToHost() method
- The value 3 (MigrateVirtualSystemToSystemSupported) shall designate MigrateVirtualSystemToSystem() method
- The value 4 (CheckVirtualSystemIsMigratableToHostSupported) shall designate the CheckVirtualSystemIsMigratableToHost () method
- The value 5 (CheckVirtualSystemIsMigratableToSystemSupported) shall designate the CheckVirtualSystemIsMigratableToSystem() method

The following rules apply to the use of these values for elements of the SynchronousMethodsSupported[ ] or AsynchronousMethodsSupported[ ] array properties in an instance of the CIM_VirtualSystemMigrationCapabilities class that represents the capabilities of a virtual system migration service:

- If a particular method is not implemented, the corresponding qualifier value shall not be present as a value for an element in either array
- If a particular method is implemented with synchronous behavior, exactly one element of the SynchronousMethodsSupported[ ] array property shall have the value that corresponds to that method as defined in this subclause
- If a particular method is implemented with asynchronous behavior, exactly one element of the AsynchronousMethodsSupported[ ] array property shall have the value that corresponds to that method as defined in this subclause

NOTE The normative text requires that methods implementing both synchronous and asynchronous behavior are indicated through both array properties.

NOTE A client discovers the fact that asynchronous execution of methods is implemented for one or more methods by checking the value of the AsynchronousMethodsSupported[ ] array property in the instance of the CIM_VirtualSystemMigrationCapabilities class representing the capabilities of a virtual system migration service is not NULL, and contains at least one non-NULL element.

7.3 Method parameterization

Instances of the CIM_VirtualSystemMigrationSettingData class shall be supported as a value of the MigrationSettingData parameter of methods of the CIM_VirtualSystemMigrationService class. If no instance or an incomplete instance is provided by the client, default values shall be applied as specified in 7.4.

7.3.1 CIM_VirtualSystemMigrationSettingData.MigrationType (Parameter)

If specified, the value of the MigrationType property shall be used to determine the requested migration type. If not specified, the respective default value shall be used to determine the migration type; see 7.4.3.

7.3.2 CIM_VirtualSystemMigrationSettingData.Priority (Parameter)

If specified, the value of the Priority property shall be used to determine the requested migration priority. If not specified, the respective default value shall be used to determine the migration priority; see 7.4.4.

7.3.3 CIM_VirtualSystemMigrationSettingData.Bandwidth (Parameter)

If specified, the value of the Bandwidth property shall be used to determine the requested bandwidth for the migration process. If not specified, the respective default value shall be used to determine the bandwidth; see 7.4.5.

7.3.4 CIM_VirtualSystemMigrationSettingData.BandwidthUnit (Parameter)

If specified, the value of the BandwidthUnit property shall be used to determine the unit of the requested bandwidth for the migration process. If not specified, the respective default value shall be used to determine the bandwidth unit; see 7.4.6.

7.3.5 CIM_VirtualSystemMigrationSettingData.TransportType (Parameter)

If specified, the value of the TransportType property shall be used to determine the requested transport type for the migration process. If not specified, the respective default value shall be used to determine the transport type; see 7.4.7.

7.3.6 CIM_VirtualSystemMigrationSettingData.OtherTransportType (Parameter)

The implementation of the OtherTransportType property is conditional.

Condition: The TransportType property is implemented, and the value of 1 (Other) is supported.

If the value of the TransportType property is 1 (Other), and the value of the OtherTransportType is specified, the value of the OtherTransportType property shall be used to determine the requested transport type for the migration process. If not specified, the respective default value shall be used to determine the transport type; see 7.4.6 and 7.4.8.

7.4 Migration settings representing capabilities

This subclause specifies the use of the CIM_VirtualSystemMigrationSettingData class and the CIM_SettingsDefineCapabilities association for the representation of default migration settings, and for the representation of valid migration settings that are accepted by methods of the related virtual system migration service.

7.4.1 Default migration settings

An instance of the CIM_VirtualSystemMigrationCapabilities class representing capabilities of a virtual system migration service shall be associated to an instance of the CIM_VirtualSystemMigrationSettingData class through an instance of the CIM_SettingsDefineCapabilities association.

Properties in the instance of the CIM_SettingsDefineCapabilities association shall be set as follows:

- The value of the PropertyPolicy property shall be 0 (Independent)
- The value of the ValueRole property shall be 0 (Default)
- The value of the ValueRange property shall be 0 (Point)

The instance of the CIM_VirtualSystemMigrationSettingData class shall convey information about the default migration settings that apply if one of the migration methods of the virtual system migration service is called and no value is provided for the MigrationSettingData parameter or if an instance of the

CIM_VirtualSystemMigrationSettingData class is provided as value of the MigrationSettingData parameter, but not all properties were assigned values.

7.4.2 Admissible migration settings

An implementation may provide instances of the CIM_VirtualSystemMigrationSettingData class that describe admissible values or ranges of admissible values for methods of a virtual system migration service.

If such instances are provided by an implementation, they shall be associated to the instance of the CIM_VirtualSystemMigrationCapabilities class that describes the capabilities of the respective virtual system migration service (see 7.2) through instances of the CIM_SettingsDefineCapabilities association where properties shall be set as follows:

- The value of the PropertyPolicy property
  - shall match 0 (Independent) if the properties of the referenced instance of the CIM_VirtualSystemMigrationSettingData class apply without correlation, that is, each property value applies regardless of other property values in the referenced instance
  - shall match 1 (Correlated) if the properties of the referenced instance of the CIM_VirtualSystemMigrationSettingData class apply with correlation, that is, each value applies only in correlation to other property values in the referenced instance
- The value of the ValueRole property shall match 3 (Supported)
- The value of the ValueRange property shall match one of the following:
  - 0 (Point) if the referenced instance of the CIM_VirtualSystemMigrationSettingData class represents one particular admissible setting. A particular instance of the CIM_VirtualSystemMigrationCapabilities class may have zero or more instances of the CIM_VirtualSystemMigrationSettingData class associated this way, expressing a set of admissible values. If such instances are provided, methods of the related virtual system migration service shall accept instances of the CIM_VirtualSystemMigrationSettingData class that match these instances.
  - 1 (Minimums) if the numeric values in the referenced instance of the CIM_VirtualSystemMigrationSettingData class represent minimum values; in this case, NULL values and values of non-numeric properties shall be ignored. A particular instance of the CIM_VirtualSystemMigrationCapabilities class may have at most one instance of the CIM_VirtualSystemMigrationSettingData class associated this way, expressing the admissible minimum. If such instance is provided, methods of the related virtual system migration service shall accept instances of the CIM_VirtualSystemMigrationSettingData class where the numeric values are above the minimum values as expressed by the referenced instance.
  - 2 (Maximums) if the numeric values in the referenced instance of the CIM_VirtualSystemMigrationSettingData class represent maximum values; in this case, NULL values and values of non-numeric properties shall be ignored. A particular instance of the CIM_VirtualSystemMigrationCapabilities class may have at most one instance of the CIM_VirtualSystemMigrationSettingData class associated this way, expressing the admissible maximum. If such instance is provided, methods of the related virtual system migration service shall accept instances of the CIM_VirtualSystemMigrationSettingData class where the numeric values are below the maximum values as expressed by the referenced instance.
  - 3 (Increments) if the numeric values in the referenced instance of the CIM_VirtualSystemMigrationSettingData class represent an increment; in this case, NULL values and values of non-numeric properties shall be ignored. A particular instance of the CIM_VirtualSystemMigrationCapabilities class may have at most one instance of the CIM_VirtualSystemMigrationSettingData class associated this way, expressing the

admissible increment. If such instance is provided, a minimum and a maximum shall be provided as specified in this subclause, and the increment shall apply between the minimum and the maximum. If such instance is provided, methods of the related virtual system migration service shall accept instances of the CIM_VirtualSystemMigrationSettingData class where the numeric values are within the range specified by the minimum and the maximum, and within that range match multiples of the increment.

7.4.3 CIM_VirtualSystemMigrationSettingData.MigrationType (Capabilities)

The value of the MigrationType property shall convey the default migration type or an admissible migration type applicable to migration processes initiated or checked through methods of a virtual system migration service.

7.4.4 CIM_VirtualSystemMigrationSettingData.Priority (Capabilities)

The value of the Priority property shall be used to convey the default priority or an admissible priority applicable to migration processes initiated or checked through methods of a virtual system migration service.

7.4.5 CIM_VirtualSystemMigrationSettingData.Bandwidth (Capabilities)

The implementation of the Bandwidth property for the representation of the default bandwidth or an admissible bandwidth is optional.

If implemented, the value of the Bandwidth property shall convey the default bandwidth or an admissible bandwidth applicable to migration processes initiated or checked through methods of a virtual system migration service.

7.4.6 CIM_VirtualSystemMigrationSettingData.BandwidthUnit (Capabilities)

The implementation of the Bandwidth property for the representation of the default bandwidth or an admissible bandwidth unit is conditional.

Condition: The BandwidthUnit property shall be implemented if the Bandwidth property for the representation of the default bandwidth or of an admissible bandwidth is implemented; see 7.4.5.

If implemented, the value of the BandwidthUnit property shall convey the default bandwidth unit applicable to the value of the default bandwidth and for the bandwidth property in migration request if the request does not specify a bandwidth unit.

7.4.7 CIM_VirtualSystemMigrationSettingData.TransportType (Capabilities)

The implementation of the TransportType property for the representation of the default transport type or an admissible transport type is optional.

If implemented, the value of the TransportType property shall convey the default transport type applicable to migration processes initiated or checked through methods of a virtual system migration service.

7.4.8 CIM_VirtualSystemMigrationSettingData.OtherTransportType (Capabilities)

The implementation of the OtherTransportType property for the representation of the default transport type or an admissible transport type is conditional.

Condition: The TransportType property is implemented, and the value 1 (Other) is supported.

If the value of the TransportType property is 1 (Other), the value of the OtherTransportType property shall be used to convey the default transport type applicable to migration processes initiated or checked through methods of a virtual system migration service.

7.5 Virtual system correlation

This subclause details requirement with respect to the correlation of the instances of the CIM_ComputerSystem class representing the source and the migrated virtual system.

7.5.1 General

The implementation of virtual system correlation is optional.

The new virtual system that is implicitly created as part of a virtual system migration process shall be logically identical to the original virtual system. The logical identity relationship shall be represented through values of the OtherIdentifyingInfo[ ] and IdentifyingDescriptions[ ] array properties such that at least one value pair exposed by these properties in the instance of the CIM_ComputerSystem class representing of the source virtual system is present in the instance of the CIM_ComputerSystem class representing the target virtual system.

7.5.2 CIM_ComputerSystem.OtherIdentifyingInfo[ ]

Condition: The CIM_VirtualSystemMigrationService.MigrateVirtualSystemToHost() method is implemented.

The value of the OtherIdentifyingInfo[ ] array property in the instance of the CIM_ComputerSystem representing the implicitly created migrated virtual system shall contain at least one element from the value of the OtherIdentifyingInfo[ ] array property in the instance of the CIM_ComputerSystem class representing the source virtual system.

7.5.3 CIM_ComputerSystem.IdentifyingDescriptions[ ]

Condition: If CIM_ComputerSystem.OtherIdentifyingInfo is implement then CIM_ComputerSystem.IdentifyingDescription shall be implemented.

The values of elements in the value of the IdentifyingDescriptions[ ] array property that have the same array index as those elements of the value of the OtherIdentifyingInfo[ ] array property that match the specification in 7.5.2 shall be identical in the instance of the CIM_ComputerSystem representing the implicitly created migrated virtual system and the instance of the CIM_ComputerSystem class representing the source virtual system.

7.5.4 Example

For example, if in the instance of the CIM_ComputerSystem class representing a source virtual system

- the value of the OtherIdentifyingInfo[ ] array property is { "CIM:GUID", "CIM:MAC", "CIM:Tag" }
- the value of the IdentifyingDescriptions array property is { "01234…", "0123456789AB", "CustomTag" }

then these values compose three value pairs that each provides a unique correlatable identification of the source virtual system. The requirements specified in 7.5.2 and 7.5.3 require that in the instance of the CIM_ComputerSystem class representing the migrated virtual system, the values of the OtherIdentifyingInfo[ ] array property and the IdentifyingDescriptions[ ] array property exhibit at least one of the value pairs defined in the instance of the CIM_ComputerSystem class representing the source virtual system. For example, if in the instance of the CIM_ComputerSystem class representing the migrated virtual system

- the value of the OtherIdentifyingInfo[ ] array property is { "CIM:Tag" }
- the value of the IdentifyingDescriptions array property is { "CustomTag" }

then the requirement would be complied with.

7.6 Indications

The implementation of indications is optional.

DMTF DSP1054 (*Indications Profile*) defines indications as the means to communicate events. The handling of events related to virtual system migration are defined in DMTF DSP1103..

8 Methods

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

The CIM Schema descriptions for any referenced method and its parameters apply.

8.1 Extrinsic methods

This subclause details specifications for extrinsic methods of the CIM_VirtualSystemMigrationService class.

8.1.1 CIM_VirtualSystemMigrationService.MigrateVirtualSystemToHost()

The implementation of the MigrateVirtualSystemToHost() method is optional; if implemented, the requirements defined in 7.2.2 apply.

Condition: The implementation of the MigrateVirtualSystemToHost() method is required if the MigrateVirtualSystemToSystem() method is not implemented; see 8.1.2.

Table 2 specifies detailed requirements for the MigrateVirtualSystemToHost() method.

Table 2 – CIM_VirtualSystemMigrationService.MigrateVirtualSystemToHost() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	ComputerSystem	CIM_ComputerSystem REF	Reference to an instance of the CIM_ComputerSystem class representing the virtual system to be migrated.
IN	DestinationHost	string	Address of the destination host; for the format see CIM Schema description.
IN	MigrationSettingData	string	Embedded instance of the CIM_VirtualSystemMigrationSettingData class.
IN	NewSystemSettingData	string	Embedded instance of the CIM_VirtualSystemSettingData class that replaces or adds property values for the virtual system after it is migrated.
IN	NewResourceSettingData	string[ ]	Embedded instances of the CIM_ResourceAllocationSettingData class that replace or add property values for virtual resources.
OUT	Job	CIM_ConcreteJob REF	A reference to the job that performs the task (NULL if the task is completed on return).

The method shall either return a return code or an exception. For return code values, see the CIM schema description.

719 The implementation of standard messages is optional. Table 3 specifies the optional standard messages
720 for the MigrateVirtualSystemToHost() method.

721 **Table 3 – MigrateVirtualSystemToHost(): Standard messages**

(Return Code) MessageID	Message Text
(6) DMTF SVPC0003	The virtual system named <Virtual_System_EN> does not exist.
(1) DMTF SVPC0004	The virtual system named <Virtual_System_Name> is in the <Virtual_System_State> state, but the requested operation requires one of the following virtual system states: <Required_Virtual_System_States>.
(1) DMTF SVPC0005	The virtual system named <Virtual_System_Name> has the virtual system type <Virtual_System_Type> state, but the requested operation requires one of the following virtual system types: <Required_Virtual_System_Types>.
(1) DMTF SVPC0006	The virtual system named <Virtual_System_Name> <Virtual_System_Config_Error> that prevent(s) the requested operation.
(1) DMTF SVPC0101	The file <File_Name_Name> <Unexpected_File_State>.
(1) DMTF SVPC0102	The file <File_Name_Name> <Failed_File_Operation>.
(4096) DMTF SVPC8001	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to the host system named <Target_Host_Name> has been initiated.
(1) DMTF SVPC8003	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to the host system named <Target_Host_Name> has failed.
(0) DMTF SVPC8004	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to the host system named <Target_Host_Name> has successfully completed.
(1) DMTF SVPC8020	The target host named <Target_Host_Name><Target_Host_State>.
(1) DMTF SVPC8021	The target virtual system named <Virtual_System_Name> already exists.
(1) DMTF SVPC8022	The target virtual system named <Virtual_System_Name> <Failed_Activity>.
(1) DMTF SVPC8023	The target virtual system named <Virtual_System_Name> <Configuration_Error>.
(1) DMTF SVPC8030	The resource requirements of the virtual system named <Virtual_System_Name> for the <Resource_Type> cannot be satisfied at the target host system named <Target_Host_Name>.
(1) DMTF SVPC8040	The <Migration_Operation>() parameter <Parameter_Name> at index <Parameter_Index> is <Parameter_Error>; expected parameter value is <Parameter_Spec>.
(1) DMTF SVPC8041	The migration of the virtual system named <Virtual_System_Name> to the target host system named <Target_Host_Name> failed because a <Component_Error> <Component_RC> occurred.
(1) DMTF SVPC8042	The migration of the virtual system named <Virtual_System_Name> to the target host system named <Target_Host_Name> failed because a timeout occurred.

722 ### 8.1.2 CIM_VirtualSystemMigrationService.MigrateVirtualSystemToSystem()

723 The implementation of the MigrateVirtualSystemToSystem() method is optional; if implemented, the
724 requirements defined in 7.2.2 apply.

725 Table 4 specifies detailed requirements for the MigrateVirtualSystemToSystem() method.

726 **Table 4 – CIM_VirtualSystemMigrationService.MigrateVirtualSystemToHost() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	ComputerSystem	CIM_ComputerSystem REF	Reference to an instance of the CIM_ComputerSystem class representing the virtual system to be migrated.
IN	DestinationSystem	CIM_System REF	Reference to an instance of the CIM_System class representing the destination host.
IN	MigrationSettingData	String	Embedded instance of the CIM_VirtualSystemMigrationSettingData class.
IN	NewSystemSettingData	String	Embedded instance of the CIM_VirtualSystemSettingData class that replaces or adds property values for the virtual system after it is migrated.
IN	NewResourceSettingData	string[ ]	Embedded instances of the CIM_ResourceAllocationSettingData class that replace or add property values for virtual resources.
OUT	NewComputerSystem	CIM_ComputerSystem REF	Reference to an instance of the CIM_ComputerSystem class representing the virtual system after it has been migrated.
OUT	Job	CIM_ConcreteJob REF	A reference to the job that performs the task (NULL if the task is completed on return).

727 The method shall either return a return code or an exception. For return code values, see the CIM
728 schema description.

729 The implementation of standard messages is optional. Table 5 specifies the optional standard messages
730 for the MigrateVirtualSystemToSystem() method.

731 **Table 5 – MigrateVirtualSystemToSystem(): Standard messages**

(Return Code) MessageID	Message Text
(6) DMTF SVPC0003	The virtual system named <Virtual_System_Name> does not exist.
(1) DMTF SVPC0004	The virtual system named <Virtual_System_Name> is in the <Virtual_System_State> state, but the requested operation requires one of the following virtual system states: <Required_Virtual_System_States>.
(1) DMTF SVPC0005	The virtual system named <Virtual_System_Name> has the virtual system type <Virtual_System_Type> state, but the requested operation requires one of the following virtual system types: <Required_Virtual_System_Types>.
(1) DMTF SVPC0006	The virtual system named <Virtual_System_Name> <Virtual_System_Config_Error> that prevent(s) the requested operation.
(1) DMTF SVPC0101	The file <File_Name_Name> <Unexpected_File_State>.
(1) DMTF SVPC0102	The file <File_Name_Name> <Failed_File_Operation>.
(4096) DMTF SVPC8001	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to the host system named <Target_Host_Name> has been initiated.
(1) DMTF SVPC8003	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to

(Return Code) MessageID	Message Text
	the host system named <Target_Host_Name> has failed.
(0) DMTF SVPC8004	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to the host system named <Target_Host_Name> has successfully completed.
(1) DMTF SVPC8020	The target host named <Target_Host_Name><Target_Host_State>.
(1) DMTF SVPC8021	The target virtual system named <Virtual_System_Name> already exists.
(1) DMTF SVPC8022	The target virtual system named <Virtual_System_Name> <Failed_Activity>.
(1) DMTF SVPC8023	The target virtual system named <Virtual_System_Name> <Configuration_Error>.
(1) DMTF SVPC8030	The resource requirements of the virtual system named <Virtual_System_Name> for the <Resource_Type> cannot be satisfied at the target host system named <Target_Host_Name>.
(1) DMTF SVPC8040	The <Migration_Operation>() parameter <Parameter_Name> at index <Parameter_Index> is <Parameter_Error>; expected parameter value is <Parameter_Spec>.
(1) DMTF SVPC8041	The migration of the virtual system named <Virtual_System_Name> to the target host system named <Target_Host_Name> failed because a <Component_Error> <Component_RC> occurred.
(1) DMTF SVPC8042	The migration of the virtual system named <Virtual_System_Name> to the target host system named <Target_Host_Name> failed because a timeout occurred.

8.1.3 CIM_VirtualSystemMigrationService.CheckVirtualSystemIsMigratableToHost()

The implementation of the CheckVirtualSystemIsMigratableToHost() method is conditional; if implemented, the requirements defined in 7.2.2 apply.

Condition: The implementation of the CheckVirtualSystemIsMigratableToHost() method is required if the CheckVirtualSystemIsMigratableToSystem() method is not implemented; see 8.1.4.

The method shall be implemented with synchronous behavior only.

Table 6 specifies detailed requirements for the CheckVirtualSystemIsMigratableToHost() method.

Table 6 – CIM_VirtualSystemMigrationService. CheckVirtualSystemIsMigratableToHost () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	ComputerSystem	CIM_ComputerSystem REF	Reference to an instance of the CIM_ComputerSystem class representing the virtual system to be migrated.
IN	DestinationHost	string	Address of the destination host; for the format see CIM Schema description.
IN	MigrationSettingData	string	Embedded instance of the CIM_VirtualSystemMigrationSettingData class.
IN	NewSystemSettingData	string	Embedded instance of the CIM_VirtualSystemSettingData class that replaces or adds property values for the virtual system after it is migrated.

Qualifiers	Name	Type	Description/Values
IN	NewResourceSettingData	string[ ]	Embedded instances of the CIM_ResourceAllocationSettingData class that replace or add property values for virtual resources.
OUT	IsMigratable	boolean	Method result indicating whether the source virtual system is migratable to the target host.

741 The method shall either return a return code or an exception. For return code values, see the CIM
742 schema description.

743 The implementation of standard messages is optional. Table 7 specifies the optional standard messages
744 for the CheckVirtualSystemIsMigratableToHost() method.

745 **Table 7 – CheckVirtualSystemIsMigratableToHost(): Standard messages**

(Return Code) MessageID	Message Text
(6) DMTF SVPC0003	The virtual system named <Virtual_System_Name> does not exist.
(1) DMTF SVPC0004	The virtual system named <Virtual_System_Name> is in the <Virtual_System_State> state, but the requested operation requires one of the following virtual system states: <Required_Virtual_System_States>.
(1) DMTF SVPC0005	The virtual system named <Virtual_System_Name> has the virtual system type <Virtual_System_Type> state, but the requested operation requires one of the following virtual system types: <Required_Virtual_System_Types>.
(1) DMTF SVPC0006	The virtual system named <Virtual_System_Name> <Virtual_System_Config_Error> that prevent(s) the requested operation.
(1) DMTF SVPC8020	The target host named <Target_Host_Name><Target_Host_State>.
(1) DMTF SVPC8021	The target virtual system named <Virtual_System_Name> already exists.
(1) DMTF SVPC8022	The target virtual system named <Virtual_System_Name> <Failed_Activity>.
(1) DMTF SVPC8023	The target virtual system named <Virtual_System_Name> <Configuration_Error>.
(1) DMTF SVPC8030	The resource requirements of the virtual system named <Virtual_System_Name> for the <Resource_Type> cannot be satisfied at the target host system named <Target_Host_Name>.

746 ### 8.1.4 CIM_VirtualSystemMigrationService.CheckVirtualSystemIsMigratableToSystem()

747 The implementation of the CheckVirtualSystemIsMigratableToSystem() method is optional; if
748 implemented, the requirements defined in 7.2.2 apply.

749 The method shall be implemented with synchronous behavior only.

750 Table 8 specifies detailed requirements for the CheckVirtualSystemIsMigratableToSystem() method.

751 **Table 8 – CIM_VirtualSystemMigrationService.CheckVirtualSystemIsMigratableToSystem()**
752 **Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	ComputerSystem	CIM_ComputerSystem REF	Reference to an instance of the CIM_ComputerSystem class representing the virtual system to be migrated.

Qualifiers	Name	Type	Description/Values
IN	DestinationSystem	CIM_System REF	Reference to an instance of the CIM_System class representing the destination host.
IN	MigrationSettingData	string	Embedded instance of the CIM_VirtualSystemMigrationSettingData class.
IN	NewSystemSettingData	string	Embedded instance of the CIM_VirtualSystemSettingData class that replaces or adds property values for the virtual system after it is migrated.
IN	NewResourceSettingData	string[ ]	Embedded instances of the CIM_ResourceAllocationSettingData class that replace or add property values for virtual resources.
OUT	IsMigratable	boolean	Method result indicating whether the source virtual system is migratable to the target host.

753 The method shall either return a return code or an exception. For return code values, see the CIM
754 schema description.

755 Support of standard messages is optional. Table 9 specifies the optional standard messages for the
756 CheckVirtualSystemIsMigratableToSystem() method.

757 **Table 9 – CheckVirtualSystemIsMigratableToSystem(): Standard messages**

(Return Code) MessageID	Message Text
(6) DMTF SVPC0003	The virtual system named <Virtual_System_Name> does not exist.
(1) DMTF SVPC0004	The virtual system named <Virtual_System_Name> is in the <Virtual_System_State> state, but the requested operation requires one of the following virtual system states: <Required_Virtual_System_States>.
(1) DMTF SVPC0005	The virtual system named <Virtual_System_Name> has the virtual system type <Virtual_System_Type> state, but the requested operation requires one of the following virtual system types: <Required_Virtual_System_Types>.
(1) DMTF SVPC0006	The virtual system named <Virtual_System_Name> <Virtual_System_Config_Error> that prevent(s) the requested operation.
(1) DMTF SVPC8020	The target host named <Target_Host_Name><Target_Host_State>.
(1) DMTF SVPC8021	The target virtual system named <Virtual_System_Name> already exists.
(1) DMTF SVPC8022	The target virtual system named <Virtual_System_Name> <Failed_Activity>.
(1) DMTF SVPC8023	The target virtual system named <Virtual_System_Name> <Configuration_Error>.
(1) DMTF SVPC8030	The resource requirements of the virtual system named <Virtual_System_Name> for the <Resource_Type> cannot be satisfied at the target host system named <Target_Host_Name>.

758 ## 8.2 Profile conventions for operations

759 Support for operations for each profile class (including associations) is specified in the following
760 subclauses. Each subclause includes either a statement “All operations in the default list in subclause 8.2
761 are supported as described by DMTF DSP0200 or a table listing all of the operations that are not

762 supported by this profile or where the profile requires behavior other than that described by DMTF
763 DSP0200.

764 The default list of operations is as follows:

765 • GetInstance
766 • Associators
767 • AssociatorNames
768 • References
769 • ReferenceNames
770 • EnumerateInstances
771 • EnumerateInstanceNames

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

774 This profile defines methods in terms of DMTF DSP0200.

775 ### 8.2.1 CIM_AffectedJobElement

776 Table 10 lists operations that either have special requirements beyond those from DMTF DSP0200 or
777 shall not be implemented.

778 **Table 10 – Operations: CIM_AffectedJobElement**

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

779 ### 8.2.2 CIM_AssociatedJobMethodResult

780 Table 11 lists operations that either have special requirements beyond those from DMTF DSP0200 or
781 shall not be implemented.

782 **Table 11 – Operations: CIM_AssociatedJobMethodResult**

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

783 ### 8.2.3 CIM_ConcreteJob

784 All operations in the default list in subclause 8.2 are supported as described by DMTF DSP0200.

8.2.4 CIM_ComputerSystem

All operations in the default list in subclause 8.2 are supported as described by DMTF DSP0200.

8.2.5 CIM_ElementCapabilities

Table 12 lists operations that either have special requirements beyond those from DMTF DSP0200 or shall not be implemented.

Table 12 – Operations: CIM_ElementCapabilities

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

8.2.6 CIM_HostedService

Table 13 lists operations that either have special requirements beyond those from DMTF DSP0200 or shall not be implemented.

Table 13 – Operations: CIM_HostedService

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

8.2.7 CIM_MethodResult

All operations in the default list in subclause 8.2 are supported as described by DMTF DSP0200.

8.2.8 CIM_OwningJobElement

Table 14 lists operations that either have special requirements beyond those from DMTF DSP0200 or shall not be implemented.

Table 14 – Operations: CIM_OwningJobElement

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

8.2.9 CIM_RegisteredProfile

All operations in the default list in subclause 8.2 are supported as described by DMTF DSP0200.

8.2.10 CIM_ServiceAffectsElement

Table 15 lists operations that either have special requirements beyond those from DMTF DSP0200 or shall not be implemented.

Table 15 – Operations: CIM_ServiceAffectsElement

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

8.2.11 CIM_SettingsDefineCapabilities

Table 16 lists operations that either have special requirements beyond those from DMTF DSP0200 or shall not be implemented.

Table 16 – Operations: CIM_SettingsDefineCapabilities

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

8.2.12 CIM_VirtualSystemMigrationCapabilities

All operations in the default list in subclause 8.2 are supported as described by DMTF DSP0200.

8.2.13 CIM_VirtualSystemMigrationService

All operations in the default list in subclause 8.2 are supported as described by DMTF DSP0200.

8.2.14 CIM_VirtualSystemMigrationSettingData

All operations in the default list in subclause 8.2 are supported as described by DMTF DSP0200.

9 Use-cases

The following use-cases and object diagrams illustrate use of this profile. They are for informational purposes only and do not introduce behavioral requirements for implementations of the profile.

9.1 Detection and inspection

This set of use-cases describes how to determine whether a conformant migration service is available for a particular virtual system, and whether the virtual system is migratable to a particular target host system.

NOTE Use-cases describing the discovery of profile implementations and the detection of instances of the central class are described in DMTF DSP1033.

9.1.1 Determine availability of migration services for a virtual system

Assumption: The client knows a reference to an instance of the CIM_ComputerSystem class that represents a virtual system.

- The client resolves the CIM_ServiceAffectsElement association to find the instance of the CIM_VirtualSystemMigrationService class that represents the virtual system migration service that is responsible for the virtual system, invoking the intrinsic AssociatorNames() CIM operation with parameter values set as follows:
  - the value of the ObjectName parameter refers to the instance of the CIM_ComputerSystem class that represents the virtual system
  - the value of the AssocClass parameter is set to “CIM_ServiceAffectsElement”
  - the value of the ResultClass parameter is set to “CIM_VirtualSystemMigrationService”

  The result is a set of references containing one element referencing the CIM_VirtualSystemMigrationService class representing virtual system migration service.

Result: The client knows virtual system migration services that are responsible for the virtual system; see DMTF DSP1033 for recipes that describe the detection of profile conformant elements, i.e. a virtual system migration service.

9.1.2 Determine the capabilities of a virtual system migration service

This use-case describes how to determine the capabilities of a virtual system migration service.

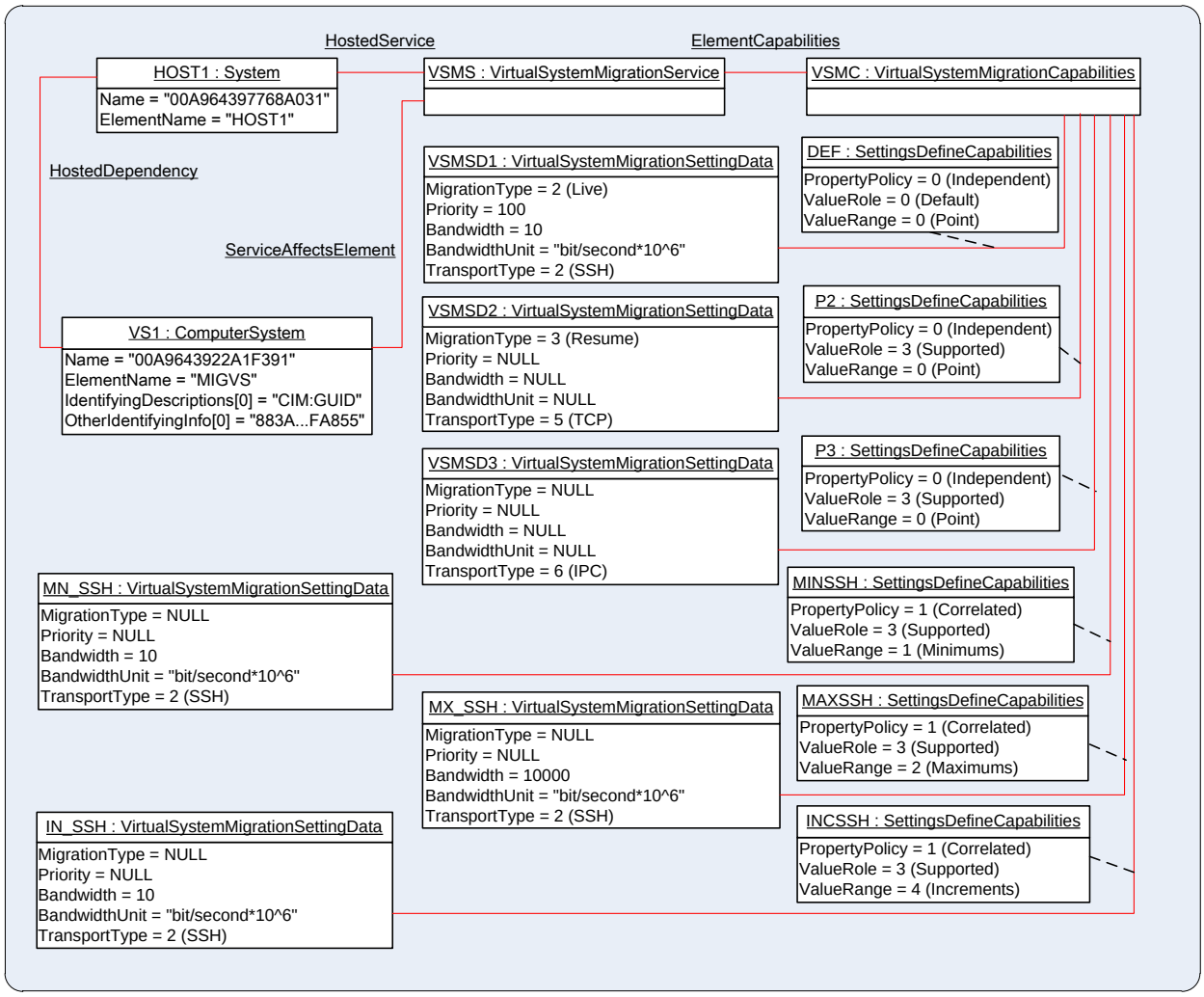

Figure 2 – Instance diagram: Virtual system migration capabilities

Assumption: The client knows a reference to an instance of the CIM_VirtualSystemMigrationService class representing a virtual system migration service.

1) The client resolves the CIM_ElementCapabilities association to find the instance of the CIM_VirtualSystemMigrationCapabilities class that represents the capabilities of the virtual system migration service, invoking the intrinsic AssociatorNames() CIM operation with parameter values set as follows:

   – the value of the ObjectName parameter refers to the instance of the CIM_VirtualSystemMigrationService class that represents the virtual system migration service. In Figure 2, that input instance is the instance VSMS.

   – the value of the AssocClass parameter is set to “CIM_ElementCapabilities”

   – the value of the ResultClass parameter is set to “CIM_VirtualSystemMigrationCapabilities”

   The result is a set of references containing one element referencing the CIM_VirtualSystemMigrationCapabilities class representing the capabilities of the virtual system migration service. (Figure 2: The result is a reference to the instance VSMC.)

2) The client obtains the set of instances of the CIM_SettingsDefineCapabilities association that associate the instance of the CIM_VirtualSystemMigrationCapabilities obtained in step 1) with instances of the CIM_VirtualSystemMigrationSettingData class that describe various details of the capabilities of the virtual system migration service, invoking the intrinsic References() operation with parameter values set as follows:

   – the value of the ObjectName parameter refers to the instance of the CIM_VirtualSystemMigrationCapabilities obtained in step 1) class that represents the virtual system migration service. (Figure 2: A reference to the instance VSMC).

   – the value of the ResultClass parameter is set to “CIM_SettingsDefineCapabilities”

   The result is a set of instances of the CIM_SettingsDefineCapabilities association that all reference the input instance. (Figure 2: Association instances DEF, P2, P3, MINSSH, MAXSSH and INCSSH).

3) For each association instance obtained in step 2), the client obtains the referenced instance of the CIM_VirtualSystemMigrationSettingData class that is references by the association instance, invoking the intrinsic GetInstance() operation with parameter values set as follows:

   – the value of the InstanceName parameter is set to the value of the PartComponent property taken from respective instance of the CIM_SettingsDefineCapabilities association as obtained in step 2).

4) For each pair of instances of the CIM_SettingsDefineCapabilities association obtained in step 2) and of the instance of the CIM_VirtualSystemMigrationSettingData class obtained in step 3) (Figure 2: Pairs DEF/VSMD1, P2/VSMD2, P3/VSMD3, MINSSH/MN_SSH, MAXSSH/MX_SSH and INCSSH/IN_SSH), the client inspects the properties values in the instance of the CIM_SettingsDefineState association, as follows:

   – If the value of the ValueRole property is 0 (Default) (Figure 2: Instance DEF), then the reference instance of the CIM_VirtualSystemMigrationSettingData class represents the default settings (Figure 2: Instance VSMDS1, indicating that live migration is used as a default migration operation, along with a relative priority of 100, a bandwidth of 10 Megabit per second and that the default transport protocol is "ssh".).

   – If the value of the ValueRole property is 3 (Supported) and the value of the ValueRange property is 0 (Point) (Figure 2: Instances P2 and P3), then the referenced instance of the CIM_VirtualSystemMigrationSettingData class represents supported settings (Figure 2: Instances VSMDS2 and VSMDS3).

   – If the value of the ValueRole property is 3 (Supported) and the value of the ValueRange property is 1 (Minimums) (Figure 2: Instance MINSSH), then the referenced instance of the CIM_VirtualSystemMigrationSettingData class represents minimum supported settings (Figure 2: Instance MN_SSH indicating a minimum bandwidth of 10 megabit per second for the TransportType 2 (SSH). Note that in Figure 2 the association instance MINSSH has set the value 1 (Correlated) for the PropertyPolicy property, indicating that all non-null values are correlated; thus the minimum setting only applies to the TransportType 2 (SSH), but not to other transport types).

   – If the value of the ValueRole property is 3 (Supported) and the value of the ValueRange property is 2 (Maximums) (Figure 2: Instance MAXSSH), then the referenced instance of the CIM_VirtualSystemMigrationSettingData class represents maximum supported settings (Figure 2: Instance MX_SSH, indicating a maximum bandwidth of 10000 megabit per second for the TransportType 2 (SSH).

   – If the value of the ValueRole property is 3 (Supported) and the value of the ValueRange property is 3 (Increments) (Figure 2: Instance INCSSH), then the referenced instance of the CIM_VirtualSystemMigrationSettingData class represents supported increment for settings (Figure 2: Instance IN_SSH, indicating that the admissible increment for bandwidth is 10 megabit per second).

Result: The client knows the capabilities of the virtual system migration service, in particular, which virtual system migration types and which transport types are supported, and what are the limitations the apply to the Bandwidth property and the Priority property if used as input instances for methods of the virtual system migration service.

9.1.3 Determine life migratability of a virtual system to a target virtualization platform

Assumption: The client knows all of the following:

- a reference to an instance of the CIM_ComputerSystem class that represents the virtual system
- a reference to an instance of the CIM_VirtualSystemMigrationService class representing the responsible virtual system migration service; see 9.1.1
- the IP address of the target virtualization platform

The sequence of activities is as follows:

1) The client invokes the extrinsic CheckVirtualSystemIsMigratableToHost() method on the instance of the CIM_VirtualSystemMigrationService, with parameter values set as follows:
   - the value of the ComputerSystem parameter refers to the instance of the CIM_Computer-System class that represents the virtual system to be migrated
   - the value of the DestinationHost parameter is set to the IP address of the target virtualization platform
   - the value of the MigrationSettingData parameter contains an embedded instance of the CIM_VirtualSystemMigrationSettingData class, with property values set as follows:
     - the value of the MigrationType property is set to 2 (Live)
     - the value of the Priority property is not set, requesting a default priority
   - the value of the NewSystemSettingData parameter is set as required
   - the value of the NewResourceSettingData[ ] array parameter is set as required to modify virtual resource allocation that existed in the source virtual system or to add new resources into the migrated virtual system
2) The implementation performs the requested check operation synchronously; the value of output parameters is set as follows:
   - The value of the return code is 0 (Completed with No Errors)
   - The value of the IsMigratable property is set to true.

Result: The client knows that a migration of the virtual system to the target virtualization platform is potentially possible. If the value of the IsMigratable property had been false, that would indicate that a migration as requested would not be possible.

NOTE The successful execution of the CheckVirtualSystemIsMigratableToHost() method does not ensure that a subsequent migration operation is successful as conditions such as resource availability may change substantially in short periods of time.

9.2 Migration operations

This clause lists use-cases describing migration operations of virtual systems.

9.2.1 Live migration

Assumption: All of the following:

- The client knows a reference to the instance of the CIM_ComputerSystem class that represents the source virtual system

- The client knows a reference to the instance of the CIM_VirtualSystemMigrationService class that is responsible for the source virtual system; see 9.1.1
- The client knows a reference to the instance of the CIM_System class that represents the target host system
- The source virtual system is in a virtual system state that is acceptable by the selected type of migration operation
- Optionally, the client may pre-check whether the source virtual system is migratable to the target virtualization platform; see 9.1.3. Note though that the migration may still fail.

The sequence of activities is as follows:

1) The client invokes the extrinsic MigrateVirtualSystemToSystem() method on the instance of the CIM_VirtualSystemMigrationService, with parameter values set as follows:
   – the value of the ComputerSystem parameter refers to the instance of the CIM_Computer-System class that represents the virtual system to be migrated
   – the value of the DestinationSystem parameter refers to the instance of the CIM_System class that represents the target virtualization platform
   – the value of the MigrationSettingData parameter contains an embedded instance of the CIM_VirtualSystemMigrationSettingData class, with property values set as follows:
      – the value of the MigrationType property is set to 2 (Live)
      – the value of the Priority property is not set, requesting a default priority
   – the value of the NewSystemSettingData parameter is not set
   – the value of the NewResourceSettingData[ ] array parameter is not set
2) The implementation initiates the requested operation as an asynchronous task; the value of output parameters is set as follows:
   – The value of the return code is 4096 (Method Parameters Checked – Job Started)
   – The value of the NewComputerSystem parameter is NULL
   – The value of the Job parameter refers to the instance of the CIM_ConcreteJob class that represents the ongoing migration operation.
3) The client tracks the state of the ongoing migration operation by repeatedly obtaining the instance of the CIM_ConcreteJob class, invoking the intrinsic GetInstance() CIM operation with parameter values set as follows:
   – The value of the InstanceName parameter refers to the instance of the CIM_ConcreteJob class using the reference returned in step 2)

   The result is the instance of the CIM_ConcreteJob class.
4) The client checks the value of the value of the JobState property in the instance of the CIM_ConcreteJob class.
   – If the value is one of 2 (New), 3 (Starting), 4 (Running) or 5 (Suspended), the client waits a certain amount of time and then continues repeating step 3).
   – If the value is 7 (Completed), the client continues with step 5).
   – If the value is one of 8 (Terminated), 9 (Killed) or 10 (Exception), the client continues with step 6).
   – The interpretation of other values of the JobState property is undefined by this profile.

5) The client resolves the CIM_AffectedJobElement association to find the instance of the CIM_ComputerSystem class that represents the migrated virtual system, invoking the intrinsic Associators() CIM operation with parameter values set as follows:

   – The value of the ObjectName parameter refers to the instance of the CIM_ConcreteJob class that represents the completed migration operation.

   – The value of the AssocClass parameter is set to “CIM_AffectedJobElement”.

   – The value of the ResultClass parameter is set to “CIM_ComputerSystem”.

   The result of this step is a set of instances of the CIM_ComputerSystem that represent the source virtual system and the target virtual system of the migration operation. From that set, the client drops the instance that represents the source virtual system, leaving the instance that represents the target virtual system.

   The use-case is completed in this case.

6) The client invokes the extrinsic GetError() method on the instance of the CIM_ConcreteJob class. The method has no input parameters.

   On return, the value of the Error parameters is an embedded instance of the CIM_Error class that conveys details about the error.

Result: The migration operation is completed. If the operation completed successfully, the client knows a reference to the instance of the CIM_ComputerSystem class representing the migrated virtual system. If the operation failed, the client knows an instance of the CIM_Error class conveying details about the failure.

Figure 3 shows the situation before and after a successful migration operation.

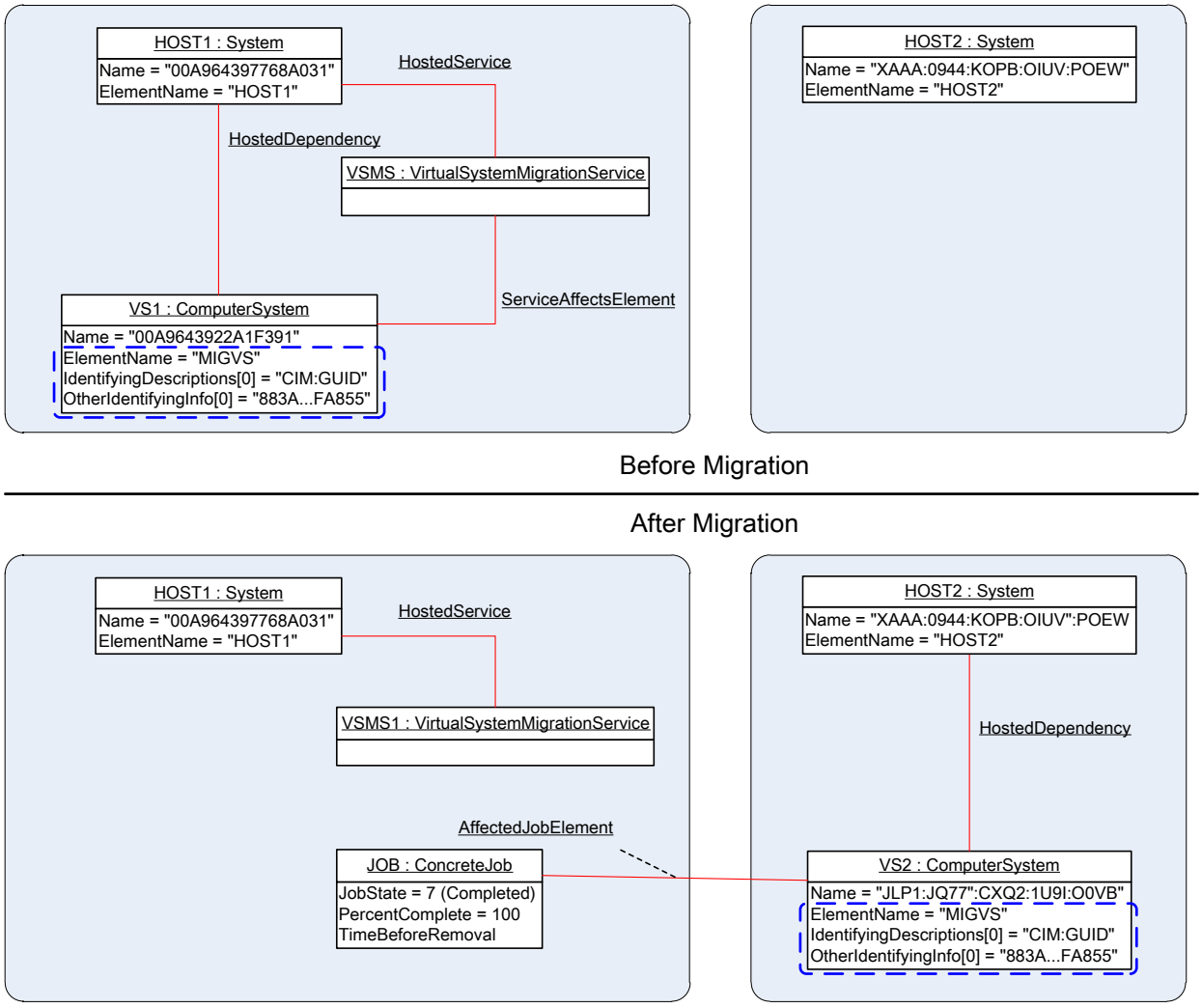

Figure 3 – Instance diagram: Virtual system migration

Note The structure of the values of the Name key property in the instances of the CIM_ComputerSystem class is implementation dependent. This profile does not specify structure and content of any key properties. However, if implemented in accordance with DMTF DSP1052:*1.0.0* (*Computer System Profile*) values of the IdentifyingDescriptions[ ] and the OtherIdentifyingInfo[ ] array properties reflect the fact that the migrated virtual system is in fact logically identical to the source virtual system by showing identical values for respective types of correlation properties as defined by DMTF DSP1052: *1.0.0*. For example, in Figure 3 the logical identity is established by means of identical GUIDs by indicating with the value “CIM:GUID” for IdentifyingDescriptions[0] that GUIDs are used for identification, and providing identical values for the OtherIdentifyingInfo[0] for both the source and the target virtual system.

9.2.2 Static migration with request for additional resources

Assumption: All of the following:

- The client knows a reference to the instance of the CIM_ComputerSystem class that represents the source virtual system
- The source virtual system has 2 GB of memory defined; this shall be increased to 5 GB as part of the migration operation
- The client knows a reference to the instance of the CIM_VirtualSystemMigrationService class that is responsible for the source virtual system; see 9.1.1

- The client knows a reference to the instance of the CIM_System class that represents the target host system
- The source virtual system is migratable to the target virtualization platform; see 9.1.3

The sequence of activities is as follows:

1) The client invokes the extrinsic MigrateVirtualSystemToSystem() method on the instance of the CIM_VirtualSystemMigrationService, with parameter values set as follows:
   - the value of the ComputerSystem parameter refers to the instance of the CIM_Computer-System class that represents the virtual system to be migrated
   - the value of the DestinationSystem parameter refers to the instance of the CIM_System class that represents the target host system
   - the value of the MigrationSettingData parameter contains an embedded instance of the CIM_VirtualSystemMigrationSettingData class, with property values set as follows:
     - the value of the MigrationType property is set to 4 (Restart)
     - the value of the Priority property is not set, requesting a default priority
   - the value of the NewSystemSettingData parameter is not set
   - the value of the NewResourceSettingData[ ] array parameter is set as follows:
     - exactly one array element containing a copy of the instance of the CIM_ResourceAllocationSettingData class that described the memory allocation with the definition of the source virtual system; see DMTF DSP1057 (*Virtual System Profile*) how to obtain that instance
     - The value of the AllocationUnits property has a value of “bytes*10^9”, indicating that memory is allocated in units of Gigabyte (with 1 GB equal to 10^9 byte)
     - The value of the Reservation property shall be set to 5, indicating that the amount of memory in the virtual system definition shall be increased to 5 GB

     Note Other combinations of values of the AllocationUnits and the Reservation property may yield the same result.
2) The implementation performs the requested operation as a synchronous task; the value of output parameters is set as follows:
   - The value of the return code is 0 (Completed with No Error)
   - The value of the NewComputerSystem parameter refers to the instance of the CIM_ComputerSystem class representing the migrated virtual system.
   - The value of the Job parameter is NULL.

Result: The migration operation is completed, the client knows a reference to the instance of the CIM_ComputerSystem class representing the migrated virtual system. The definition for the virtual systems memory size is increased to the requested value.

10 CIM elements

Table 17 lists CIM elements that are defined or specialized for this profile. Each CIM element shall be implemented as described in Table 21. The CIM schema descriptions for any referenced element and its sub-elements apply.

Clauses 7 (Implementation) and 8 (Methods) may impose additional requirements on these elements.

1071 **Table 17 – CIM Elements: Virtual System Migration profile**

Element	Requirement	Notes
Classes		
CIM_AffectedJobElement	Conditional	See 10.1
CIM_AssociatedJobMethodResult	Conditional	See 10.2
CIM_ConcreteJob	Conditional	See 10.3
CIM_ComputerSystem	Conditional	See 10.4
CIM_ElementCapabilities	Mandatory	See 10.5
CIM_Error	Conditional	See 10.6
CIM_HostedService	Mandatory	See 10.7
CIM_MethodResult	Conditional	See 10.8
CIM_OwningJobElement	Conditional	See 10.9
CIM_RegisteredProfile	Mandatory	See 10.10
CIM_ServiceAffectsElement	Mandatory	See 10.11
CIM_SettingsDefineCapabilities	Mandatory	See 10.12
CIM_VirtualSystemMigrationCapabilities	Mandatory	See 10.13
CIM_VirtualSystemMigrationService	Mandatory	See 10.14
CIM_VirtualSystemMigrationSettingData (Parameter)	Mandatory	See 10.15
CIM_VirtualSystemMigrationSettingData (Capabilities)	Mandatory	See 10.16
Indications		
Select * CIM_InstCreation Where SourceInstance ISA CIM_ConcreteJob	Conditional	Query Language: CQL Lifecycle indication representing the VS_MIGRATION_JOB_CREATE event See 10.17
Select * CIM_InstDeletion Where SourceInstance ISA CIM_ConcreteJob	Conditional	Query Language: CQL. Lifecycle indication representing the VS_MIGRATION_JOB_DELETE event See 10.18
Select * CIM_InstMethodCall Where SourceInstance ISA CIM_VirtualSystemMigrationService	Conditional	Query Language: CQL. Lifecycle indication representing the VS_MIGRATION_METHOD_CALL event. See 10.19
Select * CIM_InstModification Where SourceInstance ISA CIM_ConcreteJob AND PreviousInstance ISA CIM_ConcreteJob AND SourceInstance <> PreviousInstance	Conditional	Query Language: CQL. Lifecycle indication representing the VS_MIGRATION_JOB_CHANGE event. See 10.20

1072 ## 10.1 CIM_AffectedJobElement

1073 The implementation of the CIM_AffectedJobElement association is conditional.

1074 Condition: The CIM_AffectedJobElement association shall be implemented if one or more of the methods
1075 of the CIM_VirtualSystemMigrationService class is implemented with asynchronous execution behavior;
1076 see 8.1.1, 8.1.2, 8.1.3 and 8.1.4.

1077 The CIM_AffectedJobElement association shall be instantiated between the instance of the
1078 CIM_ConcreteJob class representing a virtual system migration task and instances of the
1079 CIM_ComputerSystem class representing the source and the target virtual systems that are affected by
1080 the virtual system migration task.

1081 Table 18 lists the requirements for elements of this association. These requirements are in addition to
1082 those specified in the CIM Schema and – if implemented – in (*Job Control Profile*).

1083 **Table 18 – Association: CIM_AffectedJobElement**

Elements	Requirement	Notes
AffectedElement	Mandatory	**Key:** Value shall reference the instance of the CIM_ ComputerSystem class **Cardinality:** 1..2
AffectingElement	Mandatory	**Key:** Value shall reference the instance of the CIM_ConcreteJob class **Cardinality:** 1
ElementEffects[ ]	Mandatory	Value shall have exactly one element that has the value 1 (Other).
OtherElementEffectsDescription[ ]	Mandatory	Value shall have exactly one element that has the value “Virtual System Migration”.

1084 ## 10.2 CIM_AssociatedJobMethodResult

1085 The implementation of the CIM_AssociatedJobMethodResult association is conditional.

1086 Condition: The CIM_AssociatedJobMethodResult association shall be implemented if one or more of the
1087 methods of the CIM_VirtualSystemMigrationService class is implemented with asynchronous execution
1088 behavior; see 8.1.1, 8.1.2, 8.1.3 and 8.1.4.

1089 The CIM_AssociatedJobMethodResult association shall be instantiated between the instance of the
1090 CIM_ConcreteJob class representing a virtual system migration task and the instance of the
1091 CIM_MethodResult class representing the related method invocation.

1092 Table 19 lists the requirements for elements of this association. These requirements are in addition to
1093 those specified in the CIM Schema, and – if implemented - in (*Job Control Profile*).

1094 **Table 19 – Association: CIM_AssociatedJobMethodResult**

Elements	Requirement	Notes
Job	Mandatory	**Key:** Value shall reference the instance of the CIM_ Concrete job class **Cardinality:** 1
JobParameters	Mandatory	**Key:** Value shall reference the instance of the CIM_MethodResult class **Cardinality:** 1

1095 ## 10.3 CIM_ConcreteJob

1096 The implementation of the CIM_ConcreteJob class is conditional.

1097 Condition: The CIM_ConcreteJob class shall be implemented if one or more of the methods of the
1098 CIM_VirtualSystemMigrationService class is implemented with asynchronous execution behavior; see
1099 8.1.1, 8.1.2, 8.1.3 and 8.1.4.

1100 Instances of the CIM_ConcreteJob class shall represent asynchronous virtual system migration tasks.

1101 Table 20 lists the requirements for elements of this class. These requirements are in addition to those
1102 specified in the CIM Schema and - if implemented - in (*Job Control Profile*).

1103 **Table 20 – Class: CIM_ConcreteJob**

Element	Requirement	Description
JobState	Mandatory	See CIM Schema
TimeOfLastStateChange	Mandatory	See CIM Schema

1104 10.4 CIM_ComputerSystem

1105 The implementation of the CIM_ComputerSystem class is conditional.

1106 Condition: The implementation of the CIM_ComputerSystem class shall be further constrained as
1107 specified in this clause beyond the requirements specified in DMTF DSP1057:*1.0.0* (*Virtual System*
1108 *Profile*), Subclause 10.2, and in DMTF DSP1052:*1.0.0* (*Computer System Profile*), Subclause 10.1 if the
1109 optional array properties OtherIdentifyingInfo[ ] and IdentifyingDescriptions[ ] array properties as specified
1110 by the DMTF DSP1052:*1.0.0*, Subclause 10.1 for instances of the CIM_ComputerSystem class
1111 representing virtual systems are implemented.

1112 Instances of the CIM_ComputerSystem class shall be used for the representation of the source and the
1113 target virtual system of a virtual system migration process.

1114 Table 21 lists the requirements for elements of this class.

1115 **Table 21 – Class: CIM_ComputerSystem**

Element	Requirement	Description
ElementName	Optional	See 7.5
IdentifyingDescriptions[ ]	Conditional	See 7.5.2
OtherIdentifyingInfo[ ]	Conditional	See 7.5.3

1116 10.5 CIM_ElementCapabilities

1117 The CIM_ElementCapabilities association shall be instantiated between an instance of the
1118 CIM_VirtualSystemMigrationService class representing a virtual system migration service and an instance
1119 of the CIM_VirtualSystemMigrationCapabilities class representing the capabilities of that virtual system
1120 migration service.

1121 Table 22 lists the requirements for elements of this association.

1122 **Table 22 – Association: CIM_ElementCapabilities**

Element	Requirement	Notes
ManagedElement	Mandatory	**Key:** Value shall reference the instance of the CIM_VirtualSystemMigrationService class **Cardinality:** *
Capabilities	Mandatory	**Key:** Value shall reference the instance of the CIM_VirtualSystemMigrationCapabilities class **Cardinality:** *

1123 ## 10.6 CIM_Error

1124 The implementation of the CIM_Error class is conditional.

1125 Condition: The CIM_Error class shall be implemented if standard messages are implemented; see 8.1.1,
1126 8.1.2, 8.1.3, 8.1.4 and subclauses of 8.2.

1127 Instances of the CIM_Error class shall be used to convey detailed error information if the execution of
1128 extrinsic methods or of intrinsic generic operations fails.

1129 Table 23 lists the requirements for elements of this class.

1130 **Table 23 – Class: CIM_Error**

Element	Requirement	Notes
MessageID	Mandatory	Value shall identify the standard message conveyed through the instance of the CIM_Error class.
Message	Mandatory	Value shall contain the formatted standard message.
MessageArguments	Mandatory	Value shall contain the dynamic content of the message.
ErrorSource	Conditional	Value shall identify the primary entity that caused the error condition.

1131 ## 10.7 CIM_HostedService

1132 The CIM_HostedService association shall be instantiated between the instance of the CIM_System class
1133 representing a host system and the instance of the CIM_VirtualSystemMigrationService class
1134 representing a virtual system migration service.

1135 Table 24 lists the requirements for elements of this association.

1136 **Table 24 – Association: CIM_HostedService**

Elements	Requirement	Notes
Antecedent	Mandatory	**Key:** Value shall reference the instance of the CIM_ System class **Cardinality:** 1
Dependent	Mandatory	**Key:** Value shall reference the instance of the CIM_VirtualSystemMigrationService class **Cardinality:** 1

1137 ## 10.8 CIM_MethodResult

1138 The implementation of the CIM_MethodResult class is conditional.

1139 Condition: The CIM_MethodResult class shall be implemented if one or more of the methods of the
1140 CIM_VirtualSystemMigrationService class is implemented with asynchronous execution behavior; see
1141 8.1.1, 8.1.2, 8.1.3 and 8.1.4.

1142 The CIM_MethodResult class shall represent invocations of methods of the
1143 CIM_VirtualSystemMigrationService that are executed asynchronously.

1144 Table 25 lists the requirements for elements of this class. These requirements are in addition to those
1145 specified in the CIM Schema, and - if implemented – in (*Job Control Profile*).

1146 **Table 25 – Class: CIM_MethodResult**

Element	Requirement	Notes
InstanceID	Mandatory	**Key**
PreCallIndication	Mandatory	Value shall be an embedded instance of the CIM_InstMethodCall indication representing pre-execution values
PostCallIndication	Conditional	If method execution is complete, value shall be an embedded instance of the CIM_InstMethodCall indication representing post-execution values

1147 ## 10.9 CIM_OwningJobElement

1148 The implementation of the CIM_OwningJobElement association is conditional.

1149 Condition: The CIM_OwningJobElement association shall be implemented if one or more of the methods
1150 of the CIM_VirtualSystemMigrationService class is implemented with asynchronous execution behavior;
1151 see 8.1.1, 8.1.2, 8.1.3 and 8.1.4.

1152 The CIM_OwningJobElement association shall be instantiated between the instance of the
1153 CIM_VirtualSystemMigrationService class representing a virtual system migration service and the
1154 instance of the CIM_ConcreteJob class representing a virtual system migration task that is owned by the
1155 service.

1156 Table 26 lists requirements for elements of this association. These requirements are in addition to those
1157 specified in the CIM Schema and - if implemented - in (*Job Control Profile*).

1158 **Table 26 – Association: CIM_OwningJobElement**

Elements	Requirement	Notes
OwningElement	Mandatory	**Key:** Value shall reference the instance of the CIM_VirtualSystemMigrationService class **Cardinality:** 1
OwnedElement	Mandatory	**Key:** Value shall reference the instance of the CIM_ConcreteJob **Cardinality:** *

1159 ## 10.10 CIM_RegisteredProfile

1160 Table 27 lists the requirements for elements of this class. These requirements are in addition to those
1161 specified by DMTF DSP1033:*1.0.0* (*Profile Registration Profile*).

1162 **Table 27 – Class: CIM_RegisteredProfile**

Elements	Requirement	Notes
RegisteredOrganization	Mandatory	Value shall be set to 2 (DMTF)
RegisteredName	Mandatory	Value shall be set to "Virtual System Migration"
RegisteredVersion	Mandatory	Value shall be set to the version of this profile: "1.0.0".

1163 ## 10.11 CIM_ServiceAffectsElement

1164 The CIM_ServiceAffectsElement association shall be instantiated between instances of the
1165 CIM_ComputerSystem class representing a virtual system and an instance of the
1166 CIM_VirtualSystemMigrationService class representing a virtual system migration service that is capable
1167 of managing migration operations for the virtual system.

1168 Table 28 lists the requirements for elements of this association.

1169 **Table 28 – Association: CIM_ServiceAffectsElement**

Element	Requirement	Notes
AffectedElement	Mandatory	**Key:** Value shall reference the instance of the CIM_ComputerSystem class representing the virtual system **Cardinality:** *
AffectingElement	Mandatory	**Key:** Value shall reference the instance of the CIM_VirtualSystemMigrationService class representing the managing virtual system migration service **Cardinality:** 1
ElementEffects[ ]	Mandatory	Value shall have exactly one element with the value set to 5 (Manages)
AssignedSequence	Mandatory	Value shall reflect the requested priority of a requested (pending) or ongoing migration operation

10.12 CIM_SettingsDefineCapabilities

The CIM_SettingsDefineCapabilities association shall be instantiated between an instance of the CIM_VirtualSystemMigrationCapabilities class representing capabilities of a virtual system migration service and an instance of the CIM_VirtualSystemMigrationSettingData class representing default values for operations of the service.

Table 29 lists the requirements for elements of this association.

Table 29 – Association: CIM_SettingsDefineCapabilities

Element	Requirement	Notes
GroupComponent	Mandatory	**Key:** Value shall reference the instance of the CIM_VirtualSystemMigrationCapabilities class representing capabilities of a virtual system migration service. **Cardinality:** 1
PartComponent	Mandatory	**Key:** Value shall reference the instance of the CIM_VirtualSystemMigrationSettingData class representing default migration settings **Cardinality:** 1..*
PropertyPolicy	Mandatory	See 7.4.1 for default migration settings and 7.4.2 for admissible migration settings.
ValueRole	Mandatory	See 7.4.1 for default migration settings and 7.4.2 for admissible migration settings.
ValueRange	Mandatory	See 7.4.1 for default migration settings and 7.4.2 for admissible migration settings.

10.13 CIM_VirtualSystemMigrationCapabilities

The CIM_VirtualSystemMigrationCapabilities class represents capabilities of one or more virtual system migration services.

Table 30 lists the requirements for elements of this class.

Table 30 – Class: CIM_VirtualSystemMigrationCapabilities

Elements	Requirement	Notes
InstanceID	Mandatory	**Key**
AsynchronousMethodsSupported[ ]	Mandatory	See 7.2.2
SynchronousMethodsSupported[ ]	Mandatory	See 7.2.2
DestinationHostsFormatsSupported[ ]	Conditional	Required if the optional CheckVirtualSystemIsMigratableToHost() method (see 8.1.2) and/or the optional MigrateVirtualSystemToHost() method (see 8.1.4) are implemented.

10.14 CIM_VirtualSystemMigrationService

The CIM_VirtualSystemMigrationService class models a virtual system migration service.

Instances of the CIM_VirtualSystemMigrationService shall represent virtual system migration services.

1185 Table 31 lists the requirements for elements of this class.

1186 **Table 31 – Class: CIM_VirtualSystemMigrationService**

Elements	Requirement	Notes
CreationClassName	Mandatory	**Key**
Name	Mandatory	**Key**
SystemCreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
MigrateVirtualSystemToHost()	Conditional	See 8.1.1
MigrateVirtualSystemToSystem()	Optional	See 8.1.2
CheckVirtualSystemIsMigratableToHost()	Conditional	See 8.1.3
CheckVirtualSystemIsMigratableToSystem()	Optional	See 8.1.4

1187

1188 10.15 CIM_VirtualSystemMigrationSettingData (Parameter)

1189 Instances of the CIM_VirtualSystemMigrationSettingData class shall represent the parameterization of
1190 operations of the CIM_VirtualSystemMigrationService class.

1191 Table 32 lists the requirements for elements of this class if used as a parameter of methods of the
1192 CIM_VirtualSystemMigrationService class.

1193 **Table 32 – Class: CIM_VirtualSystemMigrationSettingData (Parameter)**

Elements	Requirement	Notes
InstanceID	Mandatory	**Key:** Shall be NULL
MigrationType	Optional	See 7.3.1
Priority	Optional	See 7.3.2
Bandwidth	Optional	See 7.3.3
BandwidthUnit	Optional	See 7.3.4
TransportType	Optional	See 7.3.5
OtherTransportType	Conditional	See 7.3.6

1194 10.16 CIM_VirtualSystemMigrationSettingData (Capabilities)

1195 Instances of the CIM_VirtualSystemMigrationSettingData class shall represent default values that apply
1196 to method invocations of the CIM_VirtualSystemMigrationService class. Instances of the
1197 CIM_VirtualSystemMigrationSettingData class may represent admissible values for method invocations of
1198 the CIM_VirtualSystemMigrationService class if referenced by instances of the
1199 CIM_SettingsDefineCapabilities association where the value of the ValueRole property is either 0
1200 (Default) or 3 (Supported).

1201 Table 33 contains the requirements for instances of this class if used as default value expressing
1202 capabilities of the CIM_VirtualSystemMigrationService class.

1203 **Table 33 – Class: CIM_VirtualSystemMigrationSettingData (Capabilities)**

Elements	Requirement	Notes
InstanceID	Mandatory	**Key**
MigrationType	Mandatory	See 7.4.3
Priority	Mandatory	See 7.4.4
Bandwidth	Optional	See 7.4.5
BandwidthUnit	Conditional	See 7.4.6
TransportType	Optional	See 7.4.7
OtherTransportType	Conditional	See 7.4.8

1204

1205 10.17 CIM_InstCreation

1206 The implementation of the CIM_InstCreation indication is conditional.

1207 Condition: The CIM_InstCreation indication shall be implemented if indications (see 7.6) are implemented
1208 reporting the VS_MIGRATION_JOB_CREATE event.

1209 Table 34 contains the requirements for this case. These requirements are in addition to those specified in
1210 the CIM Schema and in the *Indications Profile*.

1211 **Table 34 – Indication: CIM_InstCreation**

Elements	Requirement	Notes
IndicationIdentifier	Mandatory	Value shall contain a unique identification of the indication instance; for the format see CIM Schema definition.
CorrelatedIndications[ ]	Conditional	See 7.6
IndicationTime	Mandatory	Value shall contain the time of the reported VS_MIGRATION_JOB_CREATE event
PerceivedSeverity	Mandatory	Value shall be 2 (Information)
IndicationFilterName	Mandatory	See 7.6
SourceInstance	Mandatory	See 7.6
SourceInstanceModelPath	Mandatory	Value shall refer to the instance of the CIM_ConcreteJob class copied into the SourceInstance parameter
SourceInstanceHost	Optional	Value should contain the host name or IP address of the source host system

1212 10.18 CIM_InstDeletion

1213 The implementation of the CIM_InstDeletion indication is conditional.

1214 Condition: The CIM_InstDeletion indication shall be implemented if any indications (see 7.6) are
1215 implemented reporting the VS_MIGRATION_JOB_DELETE event.

1216 Table 35 contains the requirements for this case. These requirements are in addition to those specified in
1217 the CIM Schema and in DMTF DSP1054 (*Indications Profile*).

1218 **Table 35 – Indication: CIM_InstDeletion**

Elements	Requirement	Notes
IndicationIdentifier	Mandatory	Value shall contain a unique identification of the indication instance; for the format see CIM Schema definition.
CorrelatedIndications[ ]	Conditional	See 7.6
IndicationTime	Mandatory	Value shall contain the time of the reported VS_MIGRATION_JOB_DELETE event
PerceivedSeverity	Mandatory	Value shall be 2 (Information)
IndicationFilterName	Mandatory	See 7.6
SourceInstance	Mandatory	See 7.6
SourceInstanceModelPath	Mandatory	Value shall refer to the instance of the CIM_ConcreteJob class copied into the SourceInstance parameter
SourceInstanceHost	Optional	Value should contain the host name or IP address of the source host system

1219

1220 10.19 CIM_InstMethodCall

1221 The implementation of the CIM_InstMethodCall indication is conditional.

1222 Condition: The CIM_InstMethodCall indication shall be implemented if any indications (see 7.6) are
1223 implemented reporting the VS_MIGRATION_METHOD_CALL event.

1224 Table 36 contains the requirements for this case. These requirements are in addition to those specified in
1225 the CIM Schema and in DMTF DSP1054 (*Indications Profile*).

1226 **Table 36 – Indication: CIM_InstMethodCall**

Elements	Requirement	Notes
IndicationIdentifier	Mandatory	Value shall contain a unique identification of the indication instance; for the format see CIM Schema definition.
CorrelatedIndications[ ]	Mandatory	Unspecified
IndicationTime	Mandatory	Value shall contain the time of the reported VS_MIGRATION_JOB_CHANGE event
PerceivedSeverity	Mandatory	Value shall be 2 (Information)
IndicationFilterName	Mandatory	See 7.6
SourceInstance	Mandatory	See 7.6
SourceInstanceModelPath	Mandatory	Value shall refer to the instance of the CIM_VirtualSystemMigrationService class copied into the SourceInstance parameter
SourceInstanceHost	Optional	Value should contain the host name or IP address of the source host system
MethodName	Mandatory	See 7.6
MethodParameters	Mandatory	See 7.6
ReturnValue	Conditional	Condition: Value of the PreCall property is FALSE - For value definition see CIM Schema
ReturnValueType	Conditional	Condition: Value of the PreCall property is FALSE - For value definition see CIM Schema
Error[ ]	Optional	See CIM Schema
PreCall	Mandatory	See CIM Schema

1227 ## 10.20 CIM_InstModification

1228 The implementation of the CIM_InstModification indication is conditional.

1229 Condition: The CIM_InstModification indication shall be implemented if any indications (see 7.6) are
1230 implemented reporting the VS_MIGRATION_JOB_CHANGE event.

1231 Table 37 contains the requirements for this case. These requirements are in addition to those specified in
1232 the CIM Schema and in DMTF DSP1054 (*Indications Profile*).

1233 **Table 37 – Indication: CIM_InstModification**

Elements	Requirement	Notes
IndicationIdentifier	Mandatory	Value shall contain a unique identification of the indication instance; for the format see CIM Schema definition.
CorrelatedIndications[ ]	Mandatory	Unspecified
IndicationTime	Mandatory	Value shall contain the time of the reported VS_MIGRATION_JOB_CHANGE event
PerceivedSeverity	Mandatory	Value shall be 2 (Information)
IndicationFilterName	Mandatory	See 7.6

Elements	Requirement	Notes
SourceInstance	Mandatory	See 7.6
SourceInstanceModelPath	Mandatory	Value shall refer to the instance of the CIM_ConcreteJob class copied into the SourceInstance parameter
SourceInstanceHost	Optional	Value should contain the host name or IP address of the source host system
PreviousInstance	Optional	See 7.6

1234

ANNEX A
(informative)

Change log

Version	Date	Description
1.0.0	2012-08-21	