1

2 **Document Number: DSP1103**

3 **Date: 2012-05-11**

4 **Version: 1.0.0**

5 # Job Control Profile

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: en-US**

9

CONTENTS

67   **Figures**

70   **Tables**

91

92   Foreword

93   The *Job Control Profile* (DSP1103) was prepared by the DMTF WBEM Infrastructure Modeling Working
94   Group.

95   DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
96   management and interoperability. For information about the DMTF, see http://www.dmtf.org.

97   **Acknowledgments**

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

99   • Jim Davis – WBEM Solutions, Inc.

100   • George Ericson – EMC

101   • Paul Ferdinand – WBEM Solutions

102   • Michael Johanssen – IBM

103   • Larry Lamers – VMware

104   • Andreas Maier – IBM

105   • Jim Marshall – WBEM Solutions

106   • Karl Schopmeyer – Inova Development

107   • Mike Walker – DMTF Fellows

108   The DMTF also acknowledges the contributions of the Storage Network Industry Association (SNIA).

109

Introduction

The information in this specification should be sufficient for a provider or consumer of this data to unambiguously identify the classes, properties, methods, and values that shall be instantiated to create, manage and monitor Jobs using the DMTF Common Information Model (CIM) Schema.

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*.

- Important terms that are used for the first time are marked in *italics*.

- ABNF rules are in `monospaced font`.

ABNF usage conventions

Format definitions in this document are specified using ABNF (see [RFC5234](#)), with the following deviations:

- Literal strings are to be interpreted as case-sensitive Unicode characters, as opposed to the definition in [RFC5234](#) that interprets literal strings as case-insensitive US-ASCII characters.

Experimental material

Experimental material has yet to receive sufficient review to satisfy the adoption requirements set forth by the DMTF. Experimental material is included in this document as an aid to implementers who are interested in likely future developments. Experimental material may change as implementation experience is gained. It is likely that experimental material will be included in an upcoming revision of the specification. Until that time, experimental material is purely informational.

The following typographical convention indicates experimental material:

EXPERIMENTAL

Experimental material appears here.

EXPERIMENTAL

In places where this typographical convention cannot be used (for example, tables or figures), the "EXPERIMENTAL" label is used alone

139   **Job Control Profile**

1   Scope

141   The *Job Control Profile* defines the CIM elements that are used to report and manage information
142   regarding a "job." (A "job" is a task that takes some time to execute.) This profile provides the ability to
143   monitor the status of a job from creation through completion.

2   Normative references

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

149   DMTF DSP0004, *CIM Infrastructure Specification 2.6,*
150   http://dmtf.org/sites/default/files/standards/documents/DSP0004_2.6.pdf

151   DMTF DSP0200, *CIM Operations over HTTP 1.3,*
152   http://dmtf.org/sites/default/files/standards/documents/DSP0200_1.3.pdf

153   DMTF DSP0207, *WBEM URI Mapping 1.0,*
154   http://dmtf.org/sites/default/files/standards/documents/DSP0207_1.0.pdf

155   DMTF DSP1001, *Management Profile Specification Usage Guide 1.0,*
156   http://dmtf.org/sites/default/files/standards/documents/DSP1001_1.0.pdf

157   DMTF DSP1033, *Profile Registration Profile 1.0,*
158   http://www.dmtf.org/sites/default/files/standards/documents/DSP1033_1.0.0.pdf

159   IETF RFC3986, Uniform Resource Identifier (URI): Generic Syntax, Jan. 2005,
160   http://www.ietf.org/rfc/rfc3986.txt

161   IETF RFC5234, Augmented BNF for Syntax Specifications: ABNF, Jan. 2008,
162   http://www.ietf.org/rfc/rfc5234.txt

163   ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,
164   http://isotc.iso.org

3   Terms and definitions

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

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

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

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

179 The terms defined in DSP0004, DSP0200, and DSP1001 apply to this document. The following additional
180 terms are used in this document.

181 **3.1**
182 **job**
183 a task that takes some time to execute

184 **3.2**
185 **organization**
186 consortium, standards group, or company creating a DMTF profile specification

4   Symbols and abbreviated terms

188 The abbreviations defined in DSP0004, DSP0200, and DSP1001 apply to this document. The following
189 additional abbreviations are used in this document.

190 **4.1**
191 **CQL**
192 CIM Query Language

193 **4.2**
194 **QoS**
195 Quality of service

196 **4.3**
197 **URI**
198 Uniform Resource Identifier

199 **4.4**
200 **WBEM**
201 Web Based Enterprise Management

5   Synopsis

203 **Profile name:** Job Control

204 **Version:** 1.0.0

205 **Organization:** DMTF

206 **CIM schema version:** 2.31

207 **Central Class:** CIM_ConcreteJob

208 **Scoping Class:** CIM_ManagedElement

209 The *Job Control Profile* is a component profile that extends the management capability of the referencing
210 profiles by adding the ability to manage jobs for long running tasks.

211 The Central Instance of this profile shall be an instance of CIM_ConcreteJob. The Scoping Instance shall
212 be the instance of CIM_ManagedElement (the central instance of the referencing profile) with which the
213 Central Instance (the instance of CIM_ConcreteJob) is associated through CIM_HostedDependency.

214 Table 1 identifies profiles on which this profile has a dependency.

215   **Table 1 – Referenced profiles**

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration (DSP1033)	DMTF	1.0	Mandatory	

216 6   Description (Informative)

217 The *Job Control Profile* describes the necessary properties and methods to describe the jobs supported
218 by managed elements and how a client can track the progress and/or manage a job. The execution of a
219 method could happen very quickly or could take a substantial amount of time to execute. The *Job Control*
220 *Profile* is designed to handle the case when a method takes a long time to execute and a client has the
221 need to monitor the execution.

222 Figure 1 represents the UML class diagram for the *Job Control Profile*. For simplicity, the *CIM_* prefix has
223 been removed from the names of the classes in Figure 1.
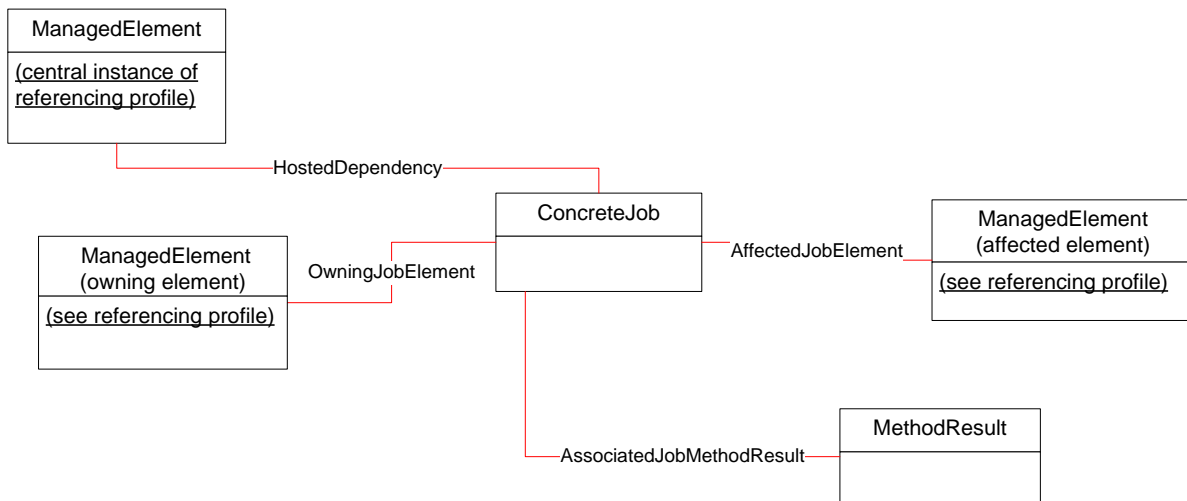
224
225

226   **Figure 1 – Job Control Profile: Class diagram**

227 When a client executes a method that supports the *Job Control Profile*, a reference to an instance of
228 ConcreteJob is returned as an output parameter and the return value for the method is set to "Method
229 parameters checked - job started". The ConcreteJob instance allows the progress of the method to be
230 checked, and instance Indications can be used to subscribe for job completion. The associations
231 OwningJobElement and AffectedJobElement are used to indicate the element whose method created the
232 job by side effect and the element being affected by the job. The job itself may create, modify, and/or
233 delete many elements during its execution. The nature of this effect is the creation or deletion of the
234 instances or associations or the modification of instance properties. These elements are said to be
235 *affected* by the job. The elements linked by AffectedJobElement may change throughout the lifecycle of

236 the ConcreteJob instance; in addition, the job may be associated to more than one Input and/or Output
237 elements or other elements affected by side effect. Input and Output elements are those referenced by
238 method parameters of the same type (input and output parameters, respectively).

239 Referencing profiles shall document each ManagedElement that will be associated to ConcreteJob
240 through AffectedJobElement. A referencing profile may add additional requirements or constraints.

241 7 Implementation

242 This clause details the requirements related to the arrangement of instances and their properties for
243 implementations of this profile. Methods are listed in clause 8 ("Methods"), and properties are listed in
244 clause 10 ("CIM Elements").

245 7.1 CIM_ConcreteJob

246 The following subclauses describe implementation requirements for CIM_ConcreteJob.

247 7.1.1 CIM_ConcreteJob.OperationalStatus

248 The OperationalStatus property is used to communicate the status of the job. As such, it is critical that
249 implementations are consistent in how this property is set. See Table 4 to see the requirements for how
250 JobState and OperationalStatus values work together. The OperationalStatus value shall be a value from
251 Table 2.

252 **Table 2 – OperationalStatus value descriptions**

OperationalStatus	Description
2 (OK), 17 (Completed)	The job completed normally.
6 (Error), 17 (Completed)	The job did not complete normally and an error occurred.
10 (Stopped)	The job was stopped (with a clean and orderly stop).
6 (Error)	The job was killed or is shutting down.
2 (OK)	The job is running or suspended.
15 (Dormant)	A new job has not yet started.
2 (OK), 8 (Starting)	The job is starting.
2 (OK), 9 (Stopping)	The job is in the process of stopping.

253 7.1.2 CIM_ConcreteJob.JobState

254 The JobState property is used to communicate job-specific state. Table 3 describes the values. The
255 JobState value shall be one of the values from Table 3.

256 **Table 3 – JobState value descriptions**

JobState	Description
2 (New)	The job was created but has not yet started.
3 (Starting)	The job has started.
4 (Running)	The job is currently executing.

JobState	Description
5 (Suspended)	The job has been suspended. The job may be suspended for many reasons, for example, if it has been usurped by a higher priority or if a client has suspended it.
6 (Shutting Down)	The job is completing its work, has been terminated, or has been killed. The job may be cleaning up after having completed only some of its work.
7 (Completed)	The job has completed normally; its work has been completed successfully.
8 (Terminated)	The job has been terminated.
9 (Killed)	The job has been aborted. The job may not clean up after itself.
10 (Exception)	The job failed and is in some abnormal state. The client may fetch the error conditions from the job.

257 Table 4 maps the permitted value combinations for the OperationalStatus and JobState properties on
258 ConcreteJob.

259 **Table 4 – OperationalStatus to JobState mapping**

OperationalStatus	JobState	Description
2 (OK), 17 (Completed)	7 (Completed)	Completed normally
6 (Error), 17 (Completed)	10 (Exception)	Completed abnormally
10 (Stopped)	8 (Terminated)	Terminated
6 (Error)	9 (Killed)	Aborted/Killed
2 (OK)	4 (Running)	Executing
15 (Dormant)	2 (New)	Created but not yet executing
2 (OK), 8 (Starting)	3 (Starting)	Starting up
2 (OK)	5 (Suspended)	Suspended
2 (OK), 9 (Stopping)	6 (Shutting Down)	Terminated and potentially cleaning up
6 (Error)	6 (Shutting Down)	Killed and is aborting

260 ### 7.1.3   CIM_ConcreteJob.PercentComplete

261 The PercentComplete property value represents the percentage of the job that has completed at the time
262 that the instance is requested. The percentage should reflect the amount of work accomplished. 0 percent
263 means that the job has not started, and 100 percent complete means the job has completed successfully.
264 In cases where an implementation cannot determine the percentage of the job that has completed, the
265 implementation should set the value to 50 percent complete while the job is running.

266 ### 7.1.4   CIM_ConcreteJob.DeleteOnCompletion

267 The DeleteOnCompletion property value indicates whether or not the job should be automatically deleted
268 upon completion. If this property is set to FALSE and the job completes, then the DeleteInstance( )
269 operation (see 8.2.1) shall be used to delete the job. If the job is set to delete on completion and the job
270 does not complete with an OperationalStatus value of 2 (OK) and 17 (Completed), the job shall remain for
271 some period of time (see 7.1.5) to allow the client to get the last error (see 8.2.2).

272 **7.1.5 CIM_ConcreteJob.TimeBeforeRemoval**

273 The job shall report how long it will remain after it has finished executing, fails on its own, is terminated, or
274 is killed. The TimeBeforeRemoval property reports a datetime offset. The TimeBeforeRemoval and
275 DeleteOnCompletion properties are related. If the value of the DeleteOnCompletion property is FALSE,
276 then the job shall remain until is it explicitly deleted. If the value of the DeleteOnCompletion property is
277 TRUE, then the job shall exist for the length of time specified in the TimeBeforeRemoval property. An
278 implementation may not support the setting of the DeleteOnCompletion property because it does not
279 support the client modifying the job instance. The amount of time specified in the TimeBeforeRemoval
280 should be five or more minutes. This amount of time allows a client to recognize that the job has failed
281 and retrieve the Error.

282 **7.2 CIM_AffectedJobElement**

283 The rules for the AffectedJobElement associations for a given job shall be defined by the referencing
284 profile.

285 **7.3 CIM_MethodResult**

286 Jobs are produced as a side effect of the invocation of an extrinsic method that takes some time to
287 execute. Reporting the resulting job is the purpose of this profile. The CIM_MethodResult class is used to
288 report the extrinsic method called and the parameters passed to the method. In this way, third-party
289 observers of a WBEM Server can tell what the job is and what it is doing. A CIM_MethodResult instance
290 contains the CIM_InstMethodCall indications that have been or would have been produced as the result
291 of the extrinsic method invocation. That is, the instance shall contain the indications whether or not there
292 was an appropriate indication subscription.

293 A client may get the CIM_InstMethodCall indication produced when the method was called from the
294 PreCallIndication property. This indication, an instance of CIM_InstMethodCall, contains the input
295 parameters provided by the client that called the method.

296 After the method execution completes, a client may get the CIM_InstMethodCall indication produced from
297 the PostCallIndication property. This indication contains the input parameters provided by the client that
298 called the method.

299 # 8 Methods

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

302 **8.1 Profile conventions for operations**

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

305 The default list of operations is as follows:

306 • GetInstance

307 • Associators

308 • AssociatorNames

309 • References

310 • ReferenceNames
311

312   • EnumerateInstances

313   • EnumerateInstanceNames

8.2   CIM_ConcreteJob

315   All operations in the default list in 8.1 shall be implemented as defined in DSP0200.

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

317   The following additional methods shall be implemented.

8.2.1   DeleteInstance

319   If the DeleteOnCompletion value is FALSE, the instance can only be deleted with the intrinsic method
320   DeleteInstance.

8.2.2   GetErrors( )

322   The GetErrors( ) method is used to get the error information if a job does not complete successfully. If the
323   value of the JobState property is 7 (Completed) and the value of the OperationalStatus property is 2 (OK)
324   or 17 (Completed), then no instance of CIM_Error is returned (that is, the value of the Errors parameter
325   will be null). If the value of the JobState property is 10 (Exception), then the GetErrors( ) method shall
326   return one or more instances of CIM_Error related to the execution of the procedure or method invoked
327   by the job. If the value of the OperationalStatus property is not 2 (OK) or 17 (Completed), then the
328   GetErrors( ) method shall return CIM_Error instances related to the running of the job.

329   The return code values and parameters for the GetErrors( ) method are specified in Table 5 and Table 6,
330   respectively.

331   **Table 5 – GetErrors( ) method: Return code values**

Value	Description
0 (Success)	The method executed successfully.
2 (Unspecified Error)	The method encountered an unspecified error during execution.
3 (Timeout)	The method timed out trying to get the error information.
4 (Failed)	The method failed.
6 (Access Denied)	Access to the execution of this method was denied.

332   **Table 6 – GetErrors( ) method: Parameters**

Qualifiers	Name	Type	Description
OUT, EmbeddedInstance(CIM_Error)	Errors	String[]	The actual CIM_Error instances explaining the failure. If the job completed successfully this value will be null. See the Referencing profile for the list of valid standard messages.

8.2.3   RequestStateChange( )

334   A job may be suspended, terminated, or killed. Suspending a job means that the job will not be executing
335   and will be suspended until it is resumed. Terminating a job means to request that the job stop executing
336   and clean up its state prior to completing. Killing a job means to request that the job abort executing,
337   usually meaning that there is little or no cleanup of the job state.

338 The RequestStateChange( ) method shall be supported. The RequestStateChange( ) method shall
339 support the RequestedState parameter with a value of 4 (Terminate) or 5 (Kill). Support for suspending
340 and resuming jobs is optional.

341 The return code values and parameters for the RequestStateChange( ) method are specified in Table 7
342 and Table 8, respectively.

343 **Table 7 – RequestStateChange( ) method: Return code values**

Value	Description
0 (Completed with No Error)	The method executed successfully.
2 (Unknown/Unspecified Error)	A failure occurred for some vendor-specific reason.
3 (Cannot complete within Timeout Period)	The requested amount of time is less than the time the requested state transition takes.
4 (Failed)	The method failed.
5 (Invalid Parameters)	One or more parameters are invalid.
6 (In Use)	Another client has requested a state change that has not completed.
4096 (Method Parameters Checked – Transition Started)	The method parameters were validated and the transition for a job has been started.
4097 (Invalid State Transition)	The state change requested is invalid for the current state.
4098 (Use of Timeout Parameter Not Supported)	This implementation does not support the TimeoutPeriod parameter. A client may pass a NULL value for the TimeoutPeriod and try again. There is no mechanism to determine what state changes are supported by a particular implementation
4099 (Busy)	A state change is underway in the job; as such, the state cannot be changed. An implementation may use this return code to indicate the job cannot be suspended, killed, or terminated at all or in the current phase of execution.

344 **Table 8 – RequestStateChange( ) method: Parameters**

Qualifiers	Name	Type	Description
IN	RequestedState	uint16	For the purposes of this profile, the valid states that can be requested are Start, Suspend, Terminate, and Kill. Terminate and Kill are mandatory. Support for Start and Suspend is optional.
IN	TimeoutPeriod	datetime	A timeout period that specifies the maximum amount of time that the client expects the transition to the new state to take. The interval format must be used to specify the TimeoutPeriod. A value of 0 or a null parameter indicates that the client has no time requirements for the transition. If this property does not contain 0 or null and the implementation does not support this parameter, a return code of 4098 (Use of Timeout Parameter Not Supported) shall be returned.

345 8.3 CIM_MethodResult

346 All operations in the default list in 8.1 shall be implemented as defined in DSP0200.

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

348   **8.4   CIM_OwningJobElement**

349   Table 9 lists implementation requirements for operations. If implemented, these operations shall be
350   implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 9, all operations in
351   the default list in 8.1 shall be implemented as defined in DSP0200.

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

353   **Table 9 – Operations: CIM_OwningJobElement**

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

354   **8.5   CIM_AffectedJobElement**

355   Table 10 lists implementation requirements for operations. If implemented, these operations shall be
356   implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 10, all operations
357   in the default list in 8.1 shall be implemented as defined in DSP0200.

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

359   **Table 10 – Operations: CIM_AffectedJobElement**

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

360   **8.6   CIM_AssociatedJobMethodResult**

361   Table 11 lists implementation requirements for operations. If implemented, these operations shall be
362   implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 11, all operations
363   in the default list in 8.1 shall be implemented as defined in DSP0200.

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

365   **Table 11 – Operations: CIM_AssociatedJobMethodResult**

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

366   8.7   CIM_HostedDependency

367   Table 12 lists implementation requirements for operations. If implemented, these operations shall be
368   implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 12, all operations
369   in the default list in 8.1 shall be implemented as defined in DSP0200

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

371   **Table 12 – Operations: CIM_HostedDependency**

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

372   9   Use cases (Informative)

373   This clause provides informative use cases and object diagrams.

374   The following use cases assume that you reviewed the object path of the job as an output parameter from
375   a method that was invoked.

376   9.1   Terminate/Kill a job

377   The termination of a job I defined as stopping the job cleanly, saving data, preserving the state and
378   shutting down all underlying processes in an orderly manner. The following steps describe the termination
379   of a job. Kill a job is defined as the termination of a job without the requirement to save data or preserve
380   the state.

381   1)   Using the object path of the job, invoke the RequestStateChange() method, with the
382   RequestedState parameter value set to 4 (terminate).

383   2)   If the return value of the method is 0 (Completed with No Error), then the job was terminated
384   successfully.

385   To Kill a job, use the same steps above, setting the RequestedState parameter value to 5 (Kill).

386   9.2   Suspend and resume a job

387   A job may be suspended and resumed using the following steps.

388   Suspend

389   1)   Invoke the requestStateChange() method with the RequestedState parameter value set to 3
390   (Suspend).

391   2)   If the return value of the method is 0 (Completed with No Error), then the job was suspended.

392   Resume

393   1)   Invoke the requestStateChange() method with the RequestedState parameter value set to 2
394   (Start).

395   2)   If the return value of the method is 0 (Completed with No Error), then the job was started.

396

10 CIM Elements

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

401   **Table 13 – CIM Elements: Job Control Profile**

Element name	Requirement	Description
Classes		
CIM_ConcreteJob	Mandatory	See 10.1.
CIM_MethodResult	Optional	See 10.2.
CIM_OwningJobElement	Mandatory	See 10.3.
CIM_AffectedJobElement	Optional	See 10.4.
CIM_AssociatedJobMethodResult	Optional	See 10.5.
CIM_HostedDependency	Mandatory	See 10.6.
CIM_RegisteredProfile	Mandatory	See 10.7.
Indications		
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ConcreteJob	Mandatory	Query Language: DMTF:CQL Creation of a job
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND SourceInstance.CIM_ConcreteJob::JobState<> PreviousInstance.CIM_ConcreteJob::JobState	Mandatory	Query Language: DMTF:CQL Change in job State
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus [*] = 17 AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus [*] = 6	Mandatory	Query Language: DMTF:CQL Job Completed with an Exception/Error
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus [*] = 17 AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus [*] = 2	Mandatory	Query Language: DMTF:CQL Job Completed Successfully

Element name	Requirement	Description
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND SourceInstance.CIM_ConcreteJob::PercentComplete <> PreviousInstance.CIM_ConcreteJob::PercentComplete	Mandatory	Query Language: DMTF:CQL Change in PercentComplete

402   10.1 CIM_ConcreteJob

403   CIM_ConcreteJob represents a job and allows the job to be monitored. Table 14 contains the
404   requirements for elements of this class.

405   **Table 14 – Class: CIM_ConcreteJob**

Elements	Requirement	Notes
InstanceID	Mandatory	**Key**
Name	Mandatory	
OperationalStatus	Mandatory	See 7.1.1.
JobState	Mandatory	See 7.1.2.
PercentComplete	Mandatory	See 7.1.3.
DeleteOnCompletion	Mandatory	See 7.1.4.
ErrorCode	Optional	A vendor-specific error code. If not null/uninitialized, this value shall be set to zero if the job completed without error.
ErrorDescription	Conditional	Mandatory when ErrorCode value is non-null.
TimeBeforeRemoval	Mandatory	See 7.1.5.
GetErrors( )	Mandatory	See 8.2.2.
RequestStateChange( )	Conditional	See 8.2.3.

406   10.2 CIM_MethodResult

407   CIM_MethodResult provides the pre and post views of execution of a method after the job is complete.
408   Table 15 contains the requirements for elements of this class.

409   **Table 15 – Class: CIM_MethodResult**

Elements	Requirement	Notes
InstanceID	Mandatory	**Key**
PreCallIndication	Mandatory	See 7.3.
PostCallIndication	Mandatory	See 7.3.

410   10.3 CIM_OwningJobElement

411   CIM_OwningJobElement represents an association between a job and the ManagedElement responsible
412   for the creation of the job. Table 16 contains the requirements for elements of this class.

413 **Table 16 – Class: CIM_OwningJobElement**

Elements	Requirement	Notes
OwningElement	Mandatory	Key: Shall reference the ManagedElement that owned the method that created the Job.
OwnedElement	Mandatory	Key: Shall reference an instance of CIM_ConcreteJob

414 **10.4 CIM_AffectedJobElement**

415 CIM_AffectedJobElement represents an association between a job and the ManagedElement(s) that may
416 be affected by its execution. Table 17 contains the requirements for elements of this class.

417 **Table 17 – Class: CIM_AffectedJobElement**

Elements	Requirement	Notes
AffectedElement	Mandatory	Key: Shall reference the instance of CIM_ManagedElement that is affected by the execution of the job.
AffectingElement	Mandatory	Key: Shall reference the instance of CIM_ConcreteJob.
ElementEffects[]	Optional	If implemented, shall include only the following values: 1 (other), 2 (Exclusive Use), 3 (Performance Impact), 4 (Element Integrity), 5 (Create)
OtherElementEffectsDescriptions[]	Conditional	Mandatory when ElementEffects contains the value 1 (Other).

418 **10.5 CIM_AssociatedJobMethodResult**

419 CIM_AssociatedJobMethodResult associates the CIM_MethodResult for a specific CIM_ConcreteJob.
420 Table 18 contains the requirements for elements of this class.

421 **Table 18 – Class: CIM_AssociatedJobMethodResult**

Elements	Requirement	Notes
Job	Mandatory	Key: Shall reference the instance of CIM_ConcreteJob
JobParameters	Mandatory	Key: Shall reference the instance of CIM_MethodResult that represents the results for the CIM_ConcreteJob instance specified in Job

422 **10.6 CIM_HostedDependency**

423 CIM_HostedDependency associates CIM_ConcreteJob instances with the central instance of the
424 referencing profile. Table 19 contains the requirements for the elements of this class.

425 **Table 19 – Class: CIM_HostedDependency**

Elements	Requirement	Notes
Antecedent	Mandatory	Key: Shall reference the central instance of the referencing profile.
Dependent	Mandatory	Key: Shall reference an instance of CIM_ConcreteJob.

426 ## 10.7 CIM_RegisteredProfile

427 The requirements denoted in Table 20 are in addition to those mandated by the *Profile Registration*
428 *Profile* (DSP1033).

429 **Table 20 – Class: CIM_RegisteredProfile**

Properties	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of "Job Control".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

430 <div align="center">**ANNEX A**</div>
431 <div align="center">**(informative)**</div>
432
433
434 **Change log**

Version	Date	Author	Description
1.0.0a	2011-04-05	Jim Davis	Work In Progress (based on submission by SNIA)
1.0.0b	2012-02-23	Jim Davis	DMTF Standard
1.0.0	2012-05-11		DMTF Standard

435